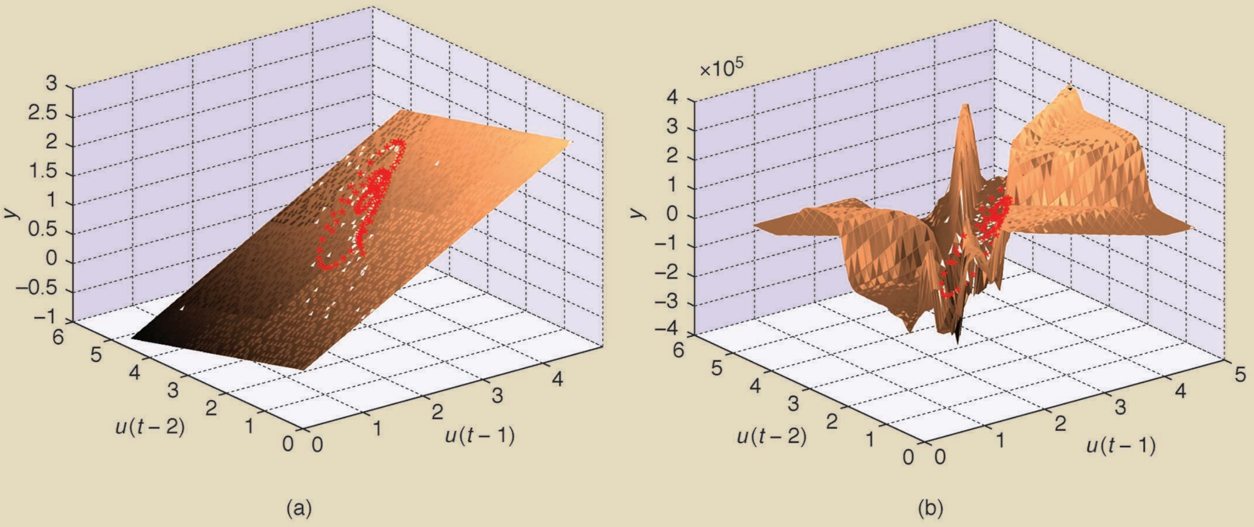
Nonlinear System Identification

A USER-ORIENTED ROAD MAP



(a)

(b)

JOHAN SCHOUKENS and LENNART LJUNG

IMAGE LICENSED BY INGRAM PUBLISHING

Nonlinear system identification is an extremely broad topic, since every system that is not linear is nonlinear. That makes it impossible to give a full overview of all aspects of the field. For this reason, the selection of topics and the organization of the discussion are strongly colored by the personal journey of the authors in this nonlinear universe.

The identification of linear dynamic systems started in the late 1950s. Zadeh [1] prioritized the need for a well-developed system identification framework at the very outset, followed by early overviews of the field [2]. A series of books established the field [3]–[7]. Linear system identification presented many successes, and data-driven modeling became an enabling factor in modern design methods. Nonlinear system identification [8]–[29] began when linear system identification [6], [7], [30] failed to address users' questions. The real world is nonlinear and time varying, and, in some applications, these aspects cannot be ignored (see Figure 1). Therefore, linear models are imprecise or do not reproduce essential aspects of system behavior. This article is focused on nonlinear system identification. Overviews of time-varying system identification are given in [31] and the references therein.

Nonlinear behavior appears in many engineering problems. In mechanical engineering, nonlinear stiffness, damping, and interconnections influence ground vibration tests of airplanes and satellites, resulting in resonance frequencies and dampings that vary with the excitation level (see Figure 2, [32], and [33]). In telecommunications, power amplifiers are pushed into a nonlinear operation regime to improve power efficiency. Distillation columns exhibit nonlinear dynamic behavior. Many biological systems (for example, the eyes, ears, and sense of touch) first apply a nonlinear compression (known as the *Weber–Fechner law*) to cover the very large dynamic range of the inputs.

Digital Object Identifier 10.1109/MCS.2019.2938121
Date of current version: 13 November 2019

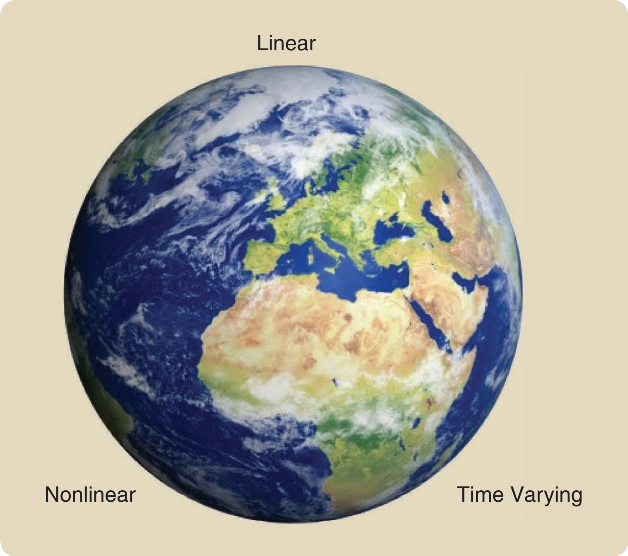


FIGURE 1 Linear system identification is successfully applied to a wide variety of problems in many different fields. However, the ever-increasing demand for higher performance and efficiency pushes systems in a nonlinear operational direction, so that nonlinear models are required for their design and control. The real world is nonlinear and time varying, and these aspects cannot be ignored. Data-driven nonlinear model building has applications in traditional industrial and emerging new high-technology applications, among others, from the mechanical and electrical fields to the electronic, telecommunication, and automotive. Biomechanical and biomedical applications can also take full advantage of a nonlinear modeling framework. Good nonlinear models provide designers with intuitive insights that can guide them toward better solutions for tomorrow's products.

Furthermore, the human brain is governed by nonlinear relations between the neurons.

The need for nonlinear system identification extends far beyond the control application field. Nonlinear models are instrumental in achieving a basic understanding of problems in which researchers still struggle with comprehending how a system works (for instance, brain activity modeling and chemical reactions). In these applications, high accuracy is not always needed, and qualitative results can be very helpful in isolating the dominant terms. Therefore, structural model errors (that is, deficiencies in the chosen model structure) become more important than noise disturbances, and the system identification tools should be properly tuned to deal with dominating structural model errors. This shows that there are many different reasons to move from linear to nonlinear models. Here we have to note the motivation of the researchers who developed nonlinear identification tools; without understanding the factors driving the researcher/scientist, it is often difficult to understand and appreciate the choices they made.

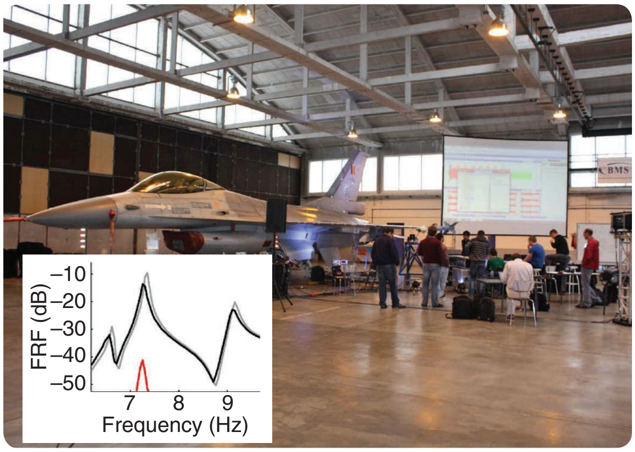


FIGURE 2 The variation of the resonance frequency and damping of a torsion mode of the wing of a fighter aircraft for varying excitation levels. The right wing is excited at the tip with a shaker. The frequency response function (FRF) from the input force to the tip acceleration is measured at a small excitation level (gray line) and a medium one (black line). The red line gives the level of the nonlinear distortions. The nonlinearity is due to friction and gaps in the bolted connection between the wing and the missile. In the inset, the FRF of the best linear approximation (BLA) $G_{\mathrm{BLA}}(f)$ is shown (gray and black line) for two different excitation levels in the frequency band of interest (see "Linear Models of Nonlinear Systems") [32], [33].

Since nonlinear system identification is an expansive topic that covers a diversity of fields, it is not possible to cover all technical details in full. Thus, as discussed in "Summary," the focus of this article is on the general concepts, with the aim of emphasizing the link between the different user choices within a unifying framework. This is also reflected in the outline of the article, which is organized into a number of sections that cover the main flow of

Summary

The goal of this article is twofold. First, nonlinear system identification is introduced to a wide audience, guiding practicing engineers and newcomers in the field to a sound solution of their data-driven modeling problems for nonlinear dynamic systems. In addition, the article provides a broad perspective on the topic for researchers who are already familiar with linear system identification theory, showing the similarities and differences between linear and nonlinear problems. The existing literature will be referred to for detailed mathematical explanations and formal proofs. Here, the focus is the basic philosophy, giving an intuitive understanding of the problems and the solutions by providing a guided tour of the wide range of user choices in nonlinear system identification. To reach that goal, guidelines and many examples are provided. The recommended reading mode is to first read the main text and then, if needed, consult the sidebars for more details.

the nonlinear system identification process, supported by sidebars to provide additional information and background information on some topics.

The article starts with a short discussion on the lead actors in (nonlinear) system identification. These are still the same as those identified in the early 1960s by Zadeh [1]: the data, the model, and the matching criterion. These are discussed from a nonlinear identification perspective in the section "The Lead Actors in Nonlinear System Identification." Next, the section "Why is Nonlinear System Identification So Involved?" clarifies why the complexity of the modeling process grows very fast when moving from linear to nonlinear system identification. The goal of the nonlinear modeling process strongly affects the required user effort, which is discussed in the "Goal of the Nonlinear System Identification Process" section. Nonlinear system identification is much more involved than linear identification. It should be a well-informed decision to move toward nonlinear methods. This is discussed in the section "Linear or Nonlinear System Identification: A User's Decision." There are many more nonlinear model structures than there are for linear systems. Making a proper choice from this wide range of possibilities is one of the major difficulties for newcomers in the field. Guidance to make a proper choice is given from a systems behavior perspective (for example, hysteresis, chaos, and fading memory) and from a user's point of view (physical or black-box model, a model that is linear in the parameters). This is discussed in the section "The Palette of Nonlinear Models" and "External or Internal Nonlinear Dynamics," respectively. Black-box nonlinear models are more difficult to assess and to understand than physical models ("Black-Box Model Complexity"). It is often hard to obtain intuitive insight in the modeled behavior because the number of terms in the model becomes very large. Eventually, attention is paid to experiment design. A number of extensive sidebars provide more detailed background information and highlight important aspects of the nonlinear identification process so that the main flow of the article remains clear.

Many aspects are illustrated by examples, and supporting software is made publicly available. Guidelines for the user to pinpoint the main lessons and conclusions are given at the end of most sections.

THE LEAD ACTORS IN NONLINEAR SYSTEM IDENTIFICATION

Any identification procedure to build models using observed input–output data is characterized by three main components: the data, a set of candidate models, and an estimation method. The process of gaining confidence in the estimated model, that is, the validation procedure, should also be added to these components. These four elements are briefly presented in this section, offering a highly structured, unified view on data-driven modeling. More comprehensive treatments will follow in forthcoming sections.

The Data

The input–output data used to select a model constitute the fundamental information source. Selecting the signals to be measured, deciding how the input should be configured, and collecting the data with appropriate sampling procedures will have a major impact on the quality of the resulting model. This is the task of *experiment design*. It is important to realize that no model can be a perfect description of the true system under investigation. Any model will be an approximation of the truth, and it will be affected by the aspects of the system that are excited during the experiment.

It is important to design the experiment to cover the intended use of the model. The input power spectrum (for example, white or colored noise) and the amplitude distribution (for example, uniform, Gaussian, or binary) of the excitation should be properly set. Of course, the classical linear identification rules also remain valid (persistency of excitation and maximum Fisher information) [6], [30]. However, these should be balanced against the other requirements to identify a well-behaving approximation model (see also "Impact of Structural Model Errors").

User Guideline

Make sure that the experiment covers the domain of interest and brings out all essential system features of interest.

The Model Structure: Set of Candidate Models

For linear identification, the choice of model sets is quite easy to grasp: a state-space structure of a certain dimension or transfer functions of certain orders. In contrast, the selection of a model set for nonlinear identification is a major problem and offers a very rich range of possibilities. It is driven by the user's preferences and directed by the system behavior. In fact, aspects of model properties and considerations of model choices dominate this article. An overview of the user's choices along possible and useful nonlinear model structures is given in the section "The Palette of Nonlinear Models," arranged by the amount of prior physical knowledge about the system that is incorporated into the model. In addition, the system's behavior imposes whether the nonlinearity should be captured in a dynamic closed loop or not. This may impact the behavioral patterns of the model and is further discussed in "External or Internal Nonlinear Dynamics."

An important distinction is whether disturbance sources enter before the nonlinearity or not. This does not only affect the behavior of the nonlinear system [34]–[37]. Dedicated identification algorithms will also be needed [38]–[40]. Proper stochastic treatment of the model becomes more

Impact of Structural Model Errors

Any estimated model has some error. It is important to distinguish between two error sources. *Structural model errors* are the type of errors from deficiencies in the model structure. The model is simply not capable of producing correct model outputs. Even with an infinite amount of perfect estimation data, the model output will have errors. *Random model errors* are the disturbances present in the estimation data that will affect the model, such that even when there are no structural model errors, the model output will have errors. These concepts are defined more precisely in the following. For simplicity, only the case of *output error models* (or *simulation models*) will be treated [see "Simulation Errors and Prediction Errors" (S21): $y(t) = y_0(t) + v(t)$]. The term $v(t)$ models the disturbances, which, for simplicity in this sidebar, are assumed to be additive and independent of the input.

For a simulation model, the predictions of $y(t)$ will depend only on past inputs

$$\hat{y}(t \mid \mathcal{S}) = y_0(t) = h_0(u^{t-1}) \quad \text{prediction with the true system,} \tag{S1}$$

$$\hat{y}(t \mid \mathcal{M}(\theta)) = h(u^{t-1}, \theta) \quad \text{prediction with a model for parameter value } \theta. \tag{S2}$$

Clearly, the output $y(t)$ contains an *innovation* $\nu(t) = y(t) - \hat{y}(t \mid \mathcal{S})$ that cannot be predicted by any model, even with the true system. Model errors y_ε will always be in addition to ν. Consider two cases:

1) *The true system is in the model class* ($\mathcal{S} \in \mathcal{M}$). There exists a θ_0 such that $h_0(u^{t-1}) = h(u^{t-1}, \theta_0)$. Normally, the estimate $\hat{\theta}_N \to \theta_0$ as $N \to \infty$ [6] and the model parameter error $\theta_0 - \hat{\theta}_N$ will be a random error caused by data disturbances $v(t)$. Therefore, the model error $h(u^{t-1}, \theta_0) - h(u^{t-1}, \hat{\theta}_N)$ will be defined by the random error in $\theta_0 - \hat{\theta}_N$, and

$$y(t) - h(u^{t-1}, \hat{\theta}_N) = \nu(t) + [h(u^{t-1}, \theta_0) - h(u^{t-1}, \hat{\theta}_N)]. \tag{S3}$$

2) *The true system is not in the model class* ($\mathcal{S} \notin \mathcal{M}$). Define

$$\theta_u^* = \lim_{N \to \infty} \hat{\theta}_N. \tag{S4}$$

The subscript u emphasizes that θ_u^* will depend on the statistical properties of the input u. The output model error $y(t) - \hat{y}(t \mid \mathcal{M}(\hat{\theta}_N))$ can now be decomposed as

$$\begin{aligned} y(t) - h(u^{t-1}, \hat{\theta}_N)) &= \nu(t) && \text{innovation} \\ &+ [h_0(u^{t-1}) - h(u^{t-1}, \theta_u^*)] && \text{structural model error} \\ &+ [h(u^{t-1}, \theta_u^*) - h(u^{t-1}, \hat{\theta}_N))] && \text{random model error.} \end{aligned} \tag{S5}$$

Denote the structural model error in (S5) as $y_\varepsilon(t) = h_\varepsilon(u^{t-1})$. Observe that this error depends upon the input.

REMARKS

1) Under mild ergodicity properties of the estimation of data disturbances, the parameter θ_u^* in (S4) obeys

$$\theta_u^* = \mathrm{argmin}_\theta E \,\| y(t) - h(u^{t-1}, \theta) \|^2, \tag{S6}$$

which stresses that θ_u^* is the best model available in the set $\mathcal{M}$ for the chosen input.

2) For some applications, it may be of interest to consider a family of different input properties and configurations $u \in \mathcal{X}$. If these configurations are equipped with a probability measure, the best model for the family $\mathcal{X}$, θ_{BAM}, can be defined as

$$\theta_{\mathrm{BAM}} = E_{u \in \mathcal{X}} \theta_u^*. \tag{S7}$$

EXPERIMENT DESIGN

A system describes that part of reality of importance to the user. The main idea of system theory is to make this description independent of the actual inputs that are applied, creating a clear split between the system characteristics and the signals on which the system acts. System identification theory implies that the excitation signal does not affect the system, and, hence, the model should not depend upon it. In the presence of structural model errors and approximate modeling, this paradigm no longer holds. The approximate model is valid only around a given working point in a restricted input domain where the structural model error $h_\varepsilon(u(t))$ is acceptably small. The approximate model depends on the working point and the class of inputs, so a major advantage of the system's theoretic framework is lost. Nevertheless, this is the best that can be done if a complete model class that includes the true system is too complex.

Example: Static Nonlinear System u(t) = arctan(u(t)) Approximated by $y_s(t) = au(t)$

The dependency of the model on the excitation class is illustrated in Figure S1, where the true system is given by $h_0(u(t)) = \arctan(u(t))$, and the simplified model is $y_s(t) = au(t)$. The disturbing noise $v(t)$ is set to zero to focus completely on the impact of the excitation class on the approximate model. A clear dependency of θ_u^* on the input distribution can be observed.

The results of this simple illustration are generally valid. Whenever a complex system is approximated with a simplified model, the results will depend on the actual applied inputs. For that reason, the experiment should be designed such that it covers the input domain of interest. It should be tuned to the problem to be solved so that the structural model errors $y_\varepsilon(t)$ remain acceptably small for the user. This will set the actual subdomain that must be covered. This is illustrated on the Duffing oscillator example in Figures 8 and S22. Within this restricted domain, it still remains important to select signals that are sufficiently rich to collect as much information as possible within the tolerable cost of the experiment.

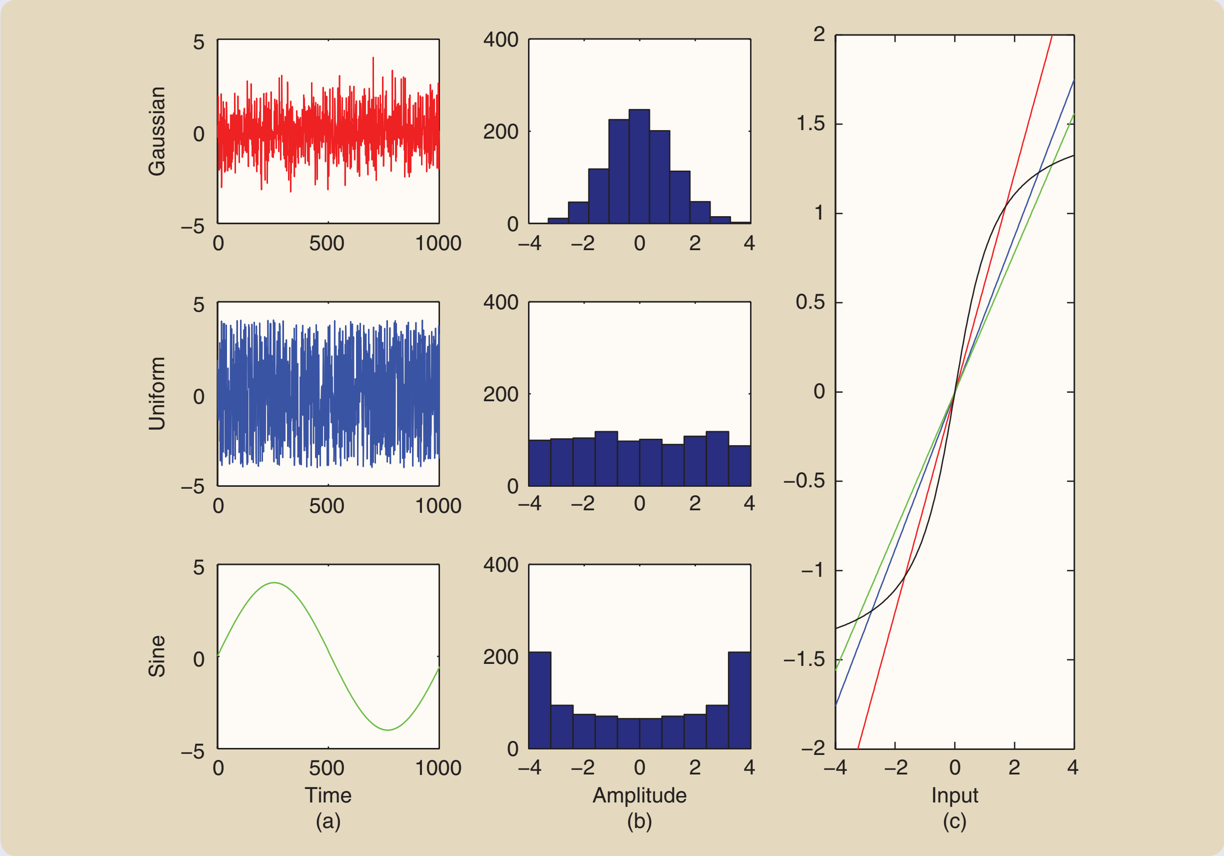


FIGURE S1 A linear approximation of a nonlinear system [79]. (a) The nonlinear system $y = \arctan(u)$ is approximated with a linear model $y = au$ for three excitation signals with a different amplitude distribution: Gaussian (red), uniform (blue), and sine (green) excitations are applied. All signals are scaled to have the same peak value. (b) The histograms for 1024 samples for each of the excitation signals. (c) The approximate linear models depend strongly on the distribution of the excitation signal. Since most of the probability mass of a Gaussian distribution is around the origin (see the Gaussian histogram), the Gaussian excitation results in the best fit in that domain (red). A sine mostly excites the extreme values (see the histogram of the sine excitation), and it results in a fit that better approximates the nonlinear function for these extreme values (green). This is at a cost of larger approximation errors around the origin. The behavior of the uniform distribution is between these two extreme distributions, and this is also true for the corresponding fit (blue).

Example: Design of an Excitation for the Wet-Clutch Setup

A simulation model of the wet-clutch setup in Figure S2 was used during an iterative control design for the system. The goal was to obtain a fast but smooth engagement of the clutch. To reach that goal, spiky signals with a large amplitude are used to obtain a short filling phase, followed by a small excitation to obtain a smooth engagement. The design of rich excitation signals that mimic this behavior is discussed in Figure S3.

CHOICE OF THE COST FUNCTION

No Structural Model Errors Present

In linear system identification, the classical choice for the cost function is [6], [7]

$$V_N(\theta) = \frac{1}{N}\sum_{t=1}^{N}\frac{1}{2}\varepsilon^2(t,\theta),$$

$$\varepsilon(t,\theta) = H^{-1}(q,\theta)[y(t) - G(q,\theta)u(t)], \tag{S8}$$

with G the plant model and H the noise model. This can also be written in the frequency domain [6] (neglecting the begin and end effects that create leakage errors [69]) as

$$V_N(\theta) = \frac{1}{N}\sum_{k=1}^{N}\frac{1}{2}\frac{|Y(k) - G(k,\theta)U(k)|^2}{|H(k,\theta)|^2}. \tag{S9}$$

$U(k)$, $Y(k)$ are the discrete Fourier transform of u, y evaluated at the frequency $f_k = kf_s/N$, and $G(k,\theta)$ and $H(k,\theta)$ are the plant and noise transfer functions, respectively, evaluated at f_k. In the prediction error framework, the parametric plant and noise models are simultaneously estimated by minimizing the

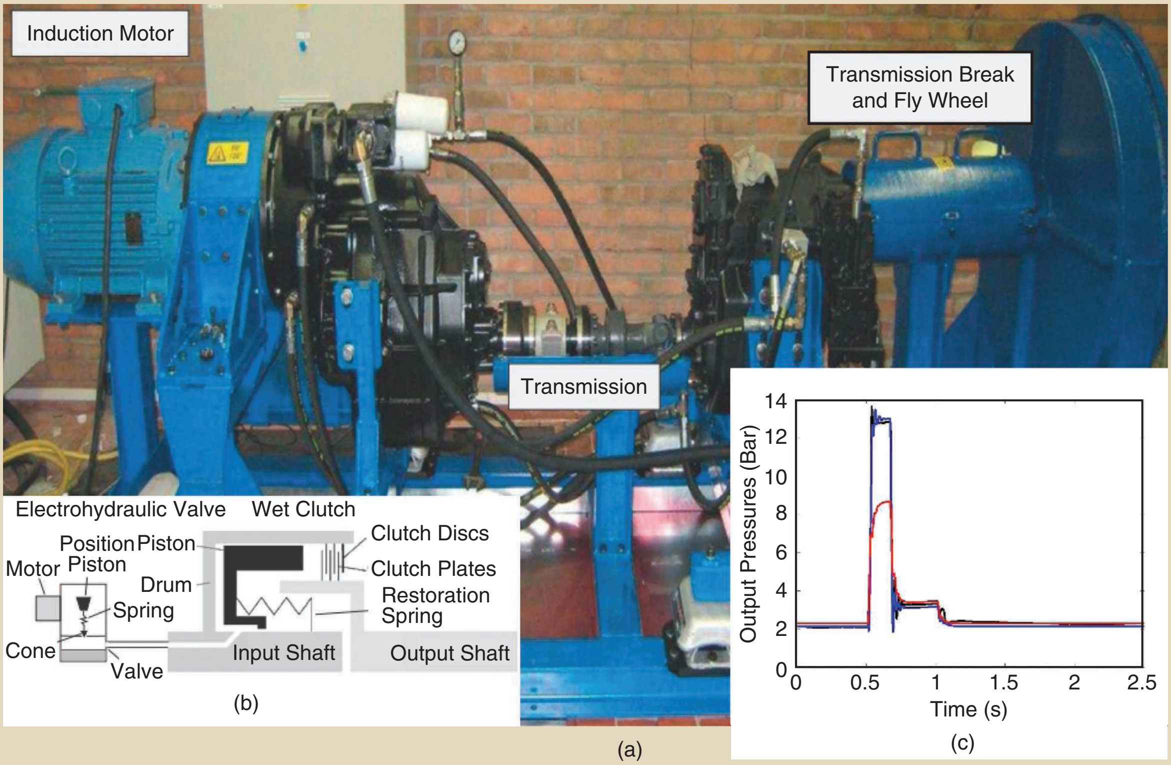


FIGURE S2 (a) A wet-clutch setup [20]. (b) A cross-sectional schematic of the electrohydraulic valve and wet clutch. A wet-clutch device transmits torque from an input axis to an output axis via fluid friction prior to the engagement of the friction plates. Such devices are commonly used in automatic transmissions for off-highway vehicles and agricultural machines to transfer torque from the engine to the load. An electrohydraulic proportional valve regulates the pressure inside the clutch, which causes the engagement of the piston with the friction plates. A model describing the relation between the current applied to the motor of the electrohydraulic valve and the resulting pressure during the filling stage of the clutch is built for a smooth engagement. (c) A linear model (red line) fails to model the true oil pressure (black line) for the spiky input, while a nonlinear state-space model (blue line) matches very well with the real data [20]. The example is further discussed in Figure S3. (Source: W.D. Widanage et al. [20]; used with permission.)

cost function $V_N(\theta)$ with respect to θ. It is shown [6] that this is the maximum likelihood formulation of the identification problem if the disturbing noise v is Gaussian distributed.

An alternative is to replace the parametric noise model by a nonparametric measurement of the noise variance $\sigma_v^2(k) = |H(k,\theta)|^2$. The variance $\sigma_v^2(k)$ is directly estimated from the data in a nonparametric preprocessing step using periodic excitations [30], [69] (see "Nonparametric Noise and Distortion Analysis Using Periodic Excitations") or starting from arbitrary excitations using the recently developed nonparametric estimation methods [88]. Observe that, in this approach, the estimation of the noise model does not depend upon the plant model, such that a too-simple plant model will not affect the noise model. The cost function is then

$$V_N(\theta) = \frac{1}{N}\sum_{k=1}^{N}\frac{1}{2}\frac{|Y(k) - G(k,\theta)U(k)|^2}{\sigma_v^2(k)}. \tag{S10}$$

In the absence of structural model errors, there is a full equivalence between both approaches [30], [70], and the differences are mostly on the implementation side [71]. This picture changes completely if structural model errors are present.

This discussion can be directly generalized to nonlinear systems by replacing the linear model $G(q,\theta)u(t)$ with the nonlinear model $h(u,\theta)$. A further generalization would be, for example, to include a nonlinear noise model to address process noise (see "Process Noise in Nonlinear System Identification").

Structural Model Errors Present

In the presence of structural model errors, the parametric noise model $|H(k,\theta)|^2$ will account for the power in both the disturbing noise $v(t)$ and the structural model errors $y_\varepsilon(t)$. Moreover, the maximum likelihood motivation is no longer valid because the structural model errors are not independent of the input u.

The nonparametric noise analysis based on periodic excitations will still estimate $\sigma_v^2(k) = |H(k,\theta)|^2$ while, for the advanced nonparametric methods [72]–[74], a combination of the disturbing noise variance and the mean squared structural model errors will be retrieved.

This raises the question of what is the best choice for the weighting function in (S8)–(S10) in the presence of structural model errors. It makes no sense to weight the structural model errors with the noise variance if the structural model errors dominate the noise. The weighting function should reflect in what frequency band larger structural model errors can be tolerated and where small structural model errors are needed. This is done by replacing the noise-variance-based weighting $H^{-1}(q,\theta)$ in (S8) or $1/\sigma_v^2(k)$ in (S10) with a predefined frequency weighting $L^{-1}(q)$ or $1/w^2(k)$ chosen by the user that reflects the most acceptable behavior of the structural model errors for the application in mind

$$\begin{aligned} V_N(\theta) &= \frac{1}{N}\sum_{t=1}^{N}\frac{1}{2}\varepsilon^2(t,\theta), \\ \varepsilon(t,\theta) &= L^{-1}(q)\,[y(t)-G(q,\theta)u(t)]. \end{aligned} \qquad \text{(S11)}$$

Observe that, in contrast to (S8), the weighting filter $L^{-1}(q)$ no longer depends on θ. In the frequency domain, the cost function (S11) is

$$\hat{\theta}_N = \text{argmin}_\theta \sum_{f\in F}\frac{|Y(f)-\hat{Y}(f|\theta)\|^2}{|L(f)|^{\,2}}. \qquad \text{(S12)}$$

By taking the sum over only the frequencies of interest $f\in F$, the fit is focused on the frequency band of interest.

COVARIANCE MATRIX EXPRESSIONS

The covariance matrix of the parameter estimates C_θ for the linear-least-squares problem $y = K(u)\theta + w$, where $y,\ w\in\mathbb{R}^{N\times 1}$, $\theta\in\mathbb{R}^{n_\theta\times 1}$, and $K(u)\in\mathbb{R}^{N\times n_\theta}$, is given by [6], [7], [30]

$$C_\theta = E_{u,w}\left[\left(\frac{1}{N}K(u)^TK(u)\right)^{-1}\frac{1}{N}K(u)^Tww^TK(u)\frac{1}{N}(K(u)^TK(u))^{-1}\right]. \qquad \text{(S13)}$$

No Structural Model Errors $y_\epsilon(t)$ Present: w(t) = v(t) and $y_\epsilon(t)$ = 0

In this case, $w(t)$ is independent of u, and (S13) converges for $N\to\infty$ to

$$C_\theta = E_u[K(u)^TK(u)]^{-1}]\,E_{u,w}[K(u)^Tww^TK(u)]\,E_u[K(u)^TK(u)]^{-1}]. \qquad \text{(S14)}$$

Making use of the independency of w and u, it follows that w is independent of $K(u)$, so that

$$\begin{aligned} E_{u,w}[K(u)^Tww^TK(u)] &= E_u[K(u)^TE_w[ww^T]K(u)] \\ &= E_u[K(u)^TC_wK(u)], \end{aligned} \qquad \text{(S15)}$$

and the covariance matrix becomes

$$C_\theta = E_u[K(u)^TK(u)]^{-1}]\,E_u[K(u)^TC_wK(u)]\,E_u[K(u)^TK(u)]^{-1}]. \qquad \text{(S16)}$$

Since w is white noise, $C_w = \sigma_w^2 I$, and

$$C_\theta = \sigma_w^2E_u[K(u)^TK(u)]^{-1}. \qquad \text{(S17)}$$

Structural Model Errors $y_\epsilon(t)$ Present: w(t) = v(t) + $y_\epsilon(t)$

In the structural model error case, $K(u)$ and w both depend upon the input u, so the independence is lost and $E_{u,w}[K(u)^Tw^TwK(u)] \neq E_u[K(u)^TE_w[w^Tw]K(u)]$. In that case, higher-order moments of u show up in the calculation of $E_{u,w}[K(u)^Tw^TwK(u)]$, and the general expression (S14) should be used. Since the dependency of w on u is not known (the structural model error is unknown), it becomes impossible to obtain closed-form expressions for the covariance matrix C_θ. This creates a huge problem because the classical but simplified expression (S17) underestimates the variability, providing the user with a far too optimistic approximation of the uncertainty of the estimates [81]. Estimating the variability directly from a set of repeated experiments with varying inputs is a pragmatic solution to this problem but provides no closed-form expressions [81].

Example 1: Linear Approximation of $y = u^n$

It is shown in [75] that the underestimation of the variance is maximal for static nonlinear systems. These can be approximated arbitrarily well in a mean-square sense using a polynomial representation. For that reason, the study of static nonlinear systems $y_0(t) = u(t)^n$ is highly informative. For such a system excited with Gaussian noise, the best linear approximation (BLA) G_{BLA} is constant (see the section "More on the Best Linear Approximation G_{BLA}" in "Linear Models of Nonlinear Systems") [30], [34], [82], [85], [86], and, hence, $G_{\text{BLA}}(q,\theta) = a_{\text{BLA}}$, which is different from zero only when n is odd. It is also possible to explicitly calculate the ratio between the full nonlinear-induced variance (with structural model errors present and input-dependent residuals) and the classical variance (with no structural model errors present and input-independent residuals) of $\hat{a}_{\text{BLA}}$ [75]

$$\sigma^2_{\text{dependant errors}}\Big/\sigma^2_{\text{independent errors}} = 2n+1. \qquad \text{(S18)}$$

This shows that the underestimation of the variance by (S17) grows with the nonlinear degree n.

Example 2: The Forced Duffing Oscillator

A linear approximating model is estimated to the forced Duffing oscillator from the experimental data (tail part) discussed in "Simulation Errors and Prediction Errors." The tail is split into 10 subrecords with a length of 8692 points, and each of these is used to identify a second-order, discrete-time plant model and a sixth-order noise model using the Box–Jenkins

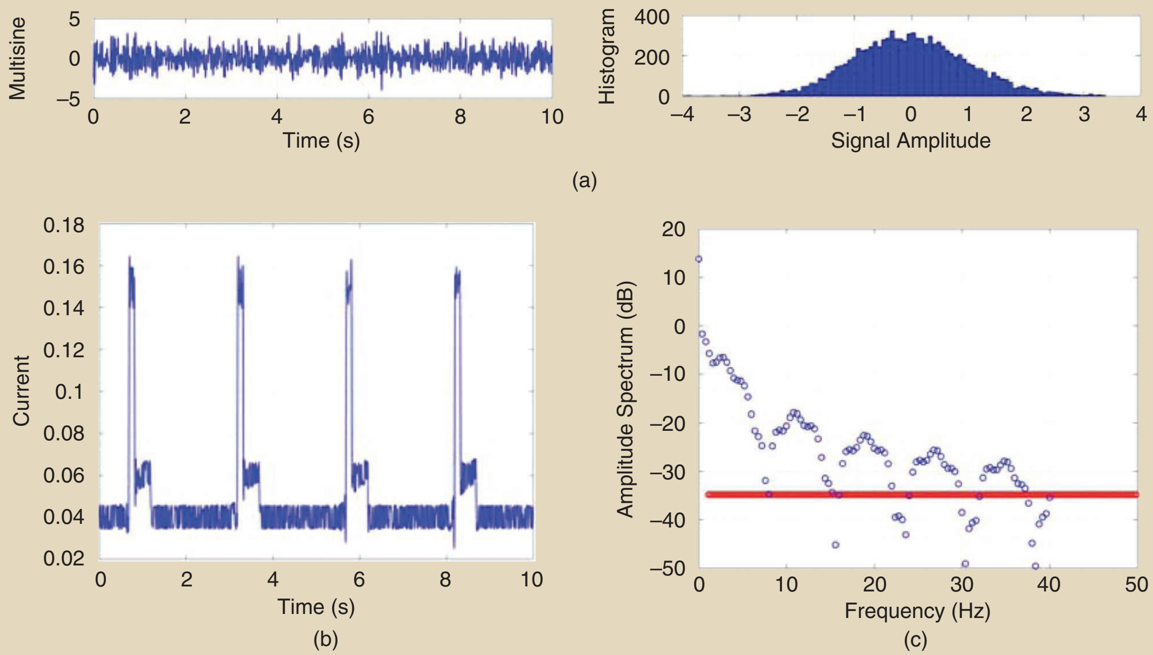


FIGURE S3 The design of excitation signals for the wet-clutch setup in Figure S2 [20]. (a) A random-phase multisine has a Gaussian amplitude distribution. This does not fit with the spiky signals typically applied to the wet clutch as shown in Figure S2. The nonlinear state-space models identified with these signals failed to simulate the system for the spiky signals that were applied during the iterative learning control design. The models even became unstable. For that reason, (b) a band-limited periodic signal was designed that consists of the sum of two signals. The first signal is a band-limited approximation of the spiky signal [see the blue spectrum in (c)]. A multisine with a flat and small-amplitude spectrum [red in (c)] is added to it. The red components excite the dynamics around the spiky profile. This results in a rich excitation that mimics the future use of the model very well. This guarantees that the structural model errors will be small for the intended application. (Source: W.D. Widanage et al. [20]; used with permission.)

model structure of the prediction-error method [6], [7]. The estimated second-order plant transfer function is shown in Figure S4. The estimation procedure resulted in the plant and noise model. From this information, in the structural model, error-free system identification approach, it is possible to obtain an estimate of the variance on the results. In Figure S4, the estimated standard deviation of the transfer function is compared with the sample standard deviation calculated from the repeated estimates on the 10 subrecords. Both curves look very similar, but the model-based estimated value (green) underestimates the actual observed standard deviation (red) by 50% or more. This is because the system identification framework fails to precisely estimate the uncertainty in the presence of nonlinear structural model errors. Note that whenever structural model errors are present, the confidence bounds are wrong.

Example 3: Wind Tunnel Experiment

In Figure S5, the underestimation of the variability in the presence of nonlinear model errors is illustrated in a wind tunnel experiment. In this experiment [93], the transfer function of the BLA [see "Linear Models of Nonlinear Systems" (S30)] is measured from the forced displacement (a random-phase multisine) [30] at the root of a wing mounted in a wind tunnel [Figure S5(a) and (b)] to the acceleration of the tip of the wing [Figure S5(b)]. The simplified variance analysis (neglecting the dependency of *w* on the input) and the actual observed variance obtained from different realizations of the experiment are shown. The simplified analysis underestimates the actual observed standard deviation by 11 dB (approximately a factor of three).

OPTIMAL STRATEGY TO GENERATE SIMPLIFIED MODELS

A key issue in system identification is how to cope with high system complexity. Sometimes structural model errors are unavoidable because overly complex models are needed to include the system in the model class. In other situations, structural model errors are deliberately created because simple models are needed in the next phase of the control design process. In that case, the experiment can be designed such that some complex system behavior is concealed, and the simple model still performs well on the domain of interest [76].

Identifying simplified models leads to structural model errors and notes all of the questions discussed before. This also raises the question of the best strategy: 1) first identify the most complex model affordable (reduce the structural model errors as much as possible), followed by a model reduction step, or 2) identify a too-simple model, dealing directly with the structural model errors.

In [76], it is shown using the maximum likelihood invariance principle that the first strategy is the best choice to follow (if it is affordable) because it results in an asymptotically efficient estimate (smallest variance) consistent (retrieves the "true" value if the number of data goes to infinity). Moreover, it allows the user to separate the identification and model reduction steps by using the maximum likelihood framework in the first step (resulting in efficient estimates with the lowest uncertainty), followed by a model reduction step using an application-oriented cost function. This two-step approach also allows the user to make a proper characterization of the reliability of the simplified model. For this reason, it is the best strategy, whenever it is affordable. However, this first option is often not affordable. The only solution is to directly identify a simple model in a setting with structural model errors. The following user guidelines help address that situation.

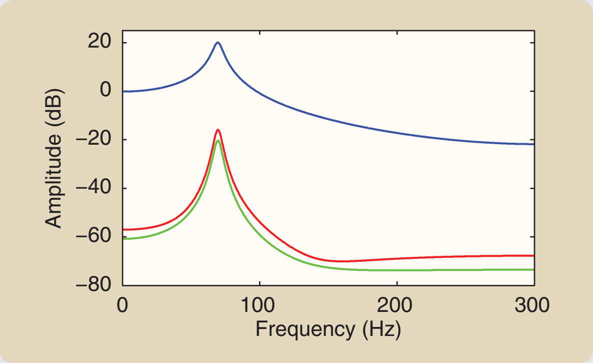


FIGURE S4 A study of the estimated standard deviation in the presence of nonlinear distortions [79]. The amplitude of the estimated transfer function model is shown in blue. The green line is the theoretic standard deviation of the estimated plant model, calculated under the assumption that there are no structural model errors (w is independent of u). The red line is the actual observed standard deviation of the estimated plant model, estimated from the variations of the estimated plant model over the 10 subrecords. It can be seen that the observed standard deviation is underestimated by 4 dB by the simplified theoretical analysis. This leads to too-small error bounds.

USER GUIDELINES: HOW TO DEAL WITH STRUCTURAL MODEL ERRORS

- *Experiment design*: The experiment should be as close as possible to the future applications so that the structural model errors are guaranteed to be small under these conditions. For example, the best simplified model can be completely different for a Gaussian or a sine excitation. The same holds true for varying power spectra, amplitude ranges, and operating points.
- *Choice of the cost function*: When structural model errors dominate, the maximum likelihood paradigm that proposes a cost function based on the disturbing noise properties is no longer the natural choice. The cost function should reflect the user's needs and express where smaller structural model errors are needed and larger structural model errors can be tolerated. This can be done by using, for example, a relative error in the cost or by proposing a user-defined weighting function.
- *Frequency weighting*: The frequency weighting of the errors should no longer reflect the disturbing noise variance when the structural model errors y_ϵ dominate the disturbing noise v. Instead, the weighting should be chosen to ensure that the structural model errors remain small in the frequency band of interest.
- *Did we lose 50 years of time?* From the discussions in this sidebar, it might appear that the past efforts on the development of a system identification framework are in vain when structural model errors exist. However, this is certainly not true. The lessons learned from the classical system identification approach should not be forgotten; the clearly structured picture that is provided in classical textbooks [6], [7], [30] is still valid at full power and provides a road map of how to organize the system identification process.

Improper data handling can overemphasize small noise disturbances, making the identification process again vulnerable to noise disturbances that are far below the structural model error level. The user is still strongly advised to make a clear split between the experiment design, the model class selection, the choice of the cost function, and the choice of the numerical procedures used to minimize the cost function. The consistency analysis developed in the structural model error-free framework still provides insight into the convergence of the algorithms in the presence of structural model errors. The major open issue that is still unsolved is how to generate reliable uncertainty bounds in the presence of structural model errors.

cumbersome and may require advanced tools when the process noise enters before the nonlinearity [41]–[56]. This is discussed in "Process Noise in Nonlinear System Identification" and "Identifying Nonlinear Dynamical Systems in the Presence of Process Noise."

In any case, a model should be capable of producing a *model output* $\hat{y}(t)$ for the output at time t, based on previous input–output measurements. This could be computed as a formal prediction of the output, or it can be based on other considerations. The set of candidate models

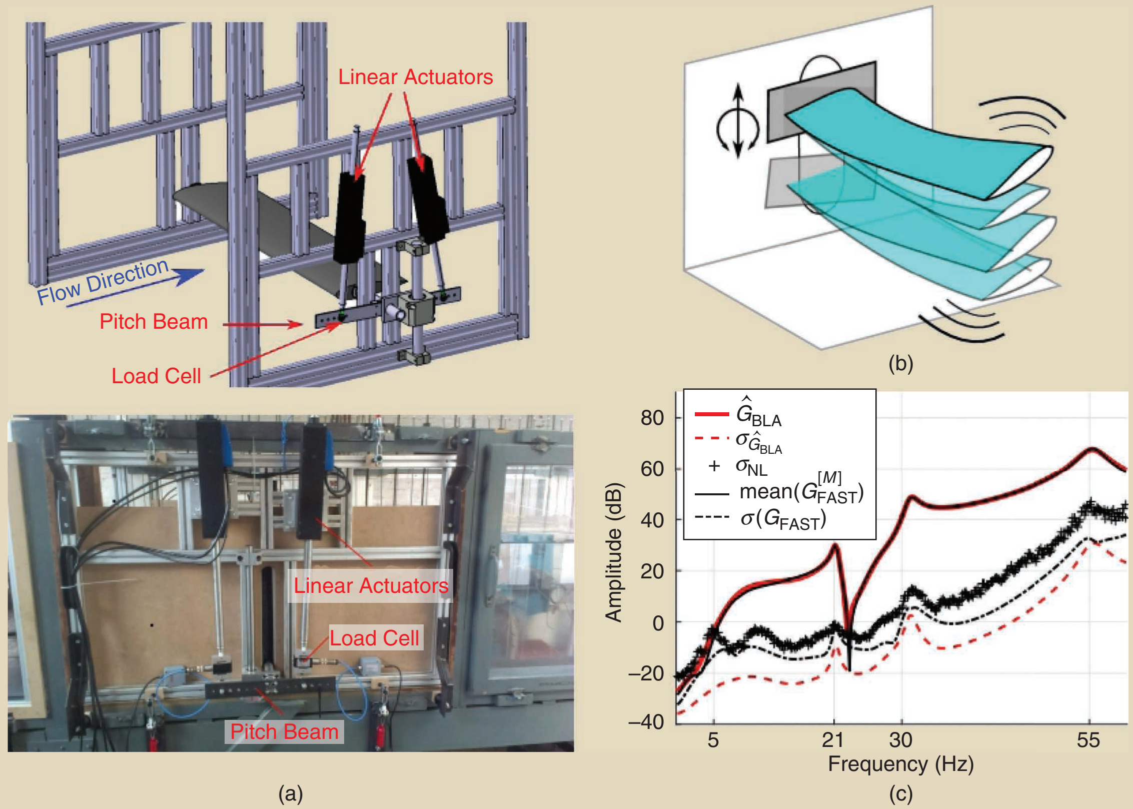


FIGURE S5 An illustration in a wind tunnel experiment [92], [93] of the underestimated standard deviation on parametric models in the presence of nonlinear model errors. In this experiment [93], (a), (b) the transfer function is measured from the forced displacement (a random phase multisine) [30] at the root of a wing mounted in a wind tunnel to (b) the acceleration of the tip of the wing. The wind speed is 8 m/s, and the angle of attack is 17.5°. The nonlinear distortion level (+), obtained from a nonparametric analysis, is 20 dB above the noise floor (not shown on this figure). The measurements are repeated for eight realizations of the random-phase multisine input (bottom). The best linear approximation (BLA) for this nonlinear system is estimated. (c) The solid red line shows the parametric transfer function model of the BLA obtained by processing all realizations together. The BLA is also estimated for each individual realization. The black solid line that coincides with the red line shows the mean value of these individual estimates. The theoretical standard deviation (broken red line) obtained under the assumption that the errors are independent of the input, underestimates the actual observed standard deviation by 11 dB (about a factor of three). This is a typical result when structural model errors dominate the noise. [Sources: (a) J. Ertveldt et al. [92] (b) and (c): J. Ertveldt [93]; used with permission.]

- *Uncertainty bounds*: The covariance matrix generated by the structural model error-free system identification framework is wrong in the presence of structural model errors. This is still like that if the noise model is tuned to also include the structural model errors. The covariance matrix underestimates the true value because it does not account for the dependency between the structural model errors and the input. This is well in line with the rule of thumb: do not pay any attention to the model's uncertainty bounds if the validation test fails. At this moment, there is no theoretical framework available that can provide more reliable bounds. The user is advised to repeat the experiment for different excitation signals (for example, different realizations of a random excitation) and to study the variability of the results under these conditions (see Figure S4) [81].

is typically parameterized by a parameter vector θ, and the notation

$$\hat{y}(t \,|\, \theta) \tag{1}$$

will be used for the model output corresponding to the model parameter θ. A common and easy case is that $\hat{y}(t|\theta)$ is linear in θ (while possibly nonlinear in the data). This case is referred to as a *linear-in-the-parameters model*.

User Guideline

The choice of the model structure is directed by behavior and structural aspects.

- *Behavior aspects*: These are imposed by the system. The question is can the selected model reproduce the observed macroscopic behavior, such as shifting resonances and hysteresis?
- *Structural aspects*: These are a user choice set by the level of physical insight that the user desires to inject into the model, ranging from white-box physical models to black-box models.
- *Structural model errors*: Such artifacts in the model result if the chosen model structure is not rich enough to contain a true description of the system.

The Estimation Method

With a given data set and model set, the identification task is to *select that model (θ) that best describes the observed data*. Most estimation methods are based on a criterion of fit between the observed output $y(t)$ and the model output $\hat{y}(t\,|\,\theta)$, which can be conceptually written as

$$\hat{\theta}_N = \operatorname{argmin}_\theta \sum_{t=1}^{N} \| y(t) - \hat{y}(t\,|\,\theta) \|^2. \tag{2}$$

If the model output is computed as a one-step-ahead prediction based on the model set and data available at time $t-1$ and the prediction error is Gaussian, then the model estimate $\hat{\theta}_N$ will be the maximum likelihood estimate. However, the conceptual method (2) can be interpreted in more pragmatic terms without making a statistical motivation (see "Impact of Structural Model Errors").

The cost function (2) can be extended with a regularization term to include prior knowledge or impose a desired behavior like smoothness or exponential decay on the solution [see also (S43) and dedicated tools can be used to impose additional structure on the solution, as discussed in "Black-Box Model Complexity"] [26], [57]–[68].

Simulation Errors and Prediction Errors

There are two basic uses of a model: *simulation* and *prediction*. Simulation means that the model is subject to an input sequence $U^t = [u(1), u(2), \ldots, u(t)]$, and its response to that input is computed. Such applications are important to evaluate the system's behavior under new situations without having to perform actual experiments. Prediction means that an observed input–output sequence $Z^t = [u(1),\ y(1),\ u(2),\ y(2), \ldots,$ $u(t),\ y(t)]$ is given, and a prediction of the next output $\hat{y}(t+1\,|\,t)$ is sought [or outputs k steps ahead of $\hat{y}(t+k\,|\,t)$], in which case, tentative future inputs $u(t+1), \ldots,\ u(t+k)$ should be supplied]. Such applications are important for system prediction but primarily for control design (control of a system subject to disturbances can be seen as control of the predicted output). Model predictive control [22] is commonly used in industry and provides an estimated model and a certain control/prediction horizon to select the control action that optimizes the predicted output.

In (9), a model is essentially a predictor for the output. The model is a *simulation model* if the prediction depends only on past inputs, so it will be focused on the simulation task. It is a *prediction model* if the prediction also depends on past outputs.

Note that a prediction model $h(Z^t)$ (omit the parameter vector θ for simplicity) can also be used as a simulation model (simulating future outputs without access to past outputs), simply by replacing, recursively, the outputs in the predictor by predicted outputs. Let

$$Z_0^t = [u(1), y_0(1), u(2), y_0(2), \ldots, u(t), y_0(t)], \tag{S19}$$

where $y_0(s) = h(Z_0^{s-1}),\ s = 1, 2\ldots$, which is depicted in Figure S6.

The *k-step-ahead predictor* $\hat{y}(t+k\,|\,k)$ is an intermediary entity between one-step-ahead prediction and simulation. For nonlinear models, a formal calculation of the k-step-ahead predictor as a conditional expectation is at least as difficult as finding the optimal one-step-ahead predictor. However, a pragmatic and approximate way to compute a k-step-ahead predictor from any one-step-ahead prediction formula, optimal or not, is as follows, mimicking (S19).

Let $\hat{y}(t\,|\,t-1) = f((y-1),\ u(t-1),\ y(t-2),\ u(t-2),\ y(t-3),$ $u(t-3), \ldots\ldots)$ be the expression for the one-step-ahead predictor. Form the expression for the k-step-ahead predictor at time t by recursively performing (in t and k)

$$\begin{aligned}
\hat{y}(t\,|\,t-2) &= f(\hat{y}(t-1|\,t-2), u(t-1), y(t-2), u(t-2),\\
&\qquad y(t-3), u(t-3), \ldots),\\
\hat{y}(t\,|\,t-3) &= f(\hat{y}(t-1|\,t-3), u(t-1), \hat{y}(t-2\,|\,t-3), u(t-2),\\
&\qquad y(t-3), u(t-3), \ldots),\\
&\ \ \vdots\\
\hat{y}(t\,|\,t-k) &= f(\hat{y}(t-1|\,t-k), u(t-1), \hat{y}(t-2\,|\,t-k), u(t-2),\\
&\qquad \hat{y}(t-3\,|\,t-k), u(t-3), \ldots),
\end{aligned} \tag{S20}$$

where $\hat{y}(t-1|\,t-k)$ has first been computed. If f is the optimal one-step-ahead predictor and a linear function of its arguments, this gives the optimal k-step-ahead predictor. Otherwise, it is an approximate method for producing outputs k steps ahead while still providing an idea of the model's properties over longer horizons.

User Guidelines

The criterion of fit can be based on a statistically grounded choice if the noise disturbances dominate over the structural model errors. If the latter dominate, a weighting function can be selected that reduces the impact of structural model errors in the domain of interest (for instance, using a user-selected frequency weighting), at a cost of obtaining larger errors outside the domain of interest.

Model Validation

When a model has been estimated, the questions Does it solve our problem? and/or Is it in conflict with either the data or prior knowledge? are asked. This is the essential procedure of model validation, which is further discussed subsequently in the section with this title.

A key element is that model validation typically involves a validation data set, an independent data set from the system that was not used to estimate the model. Often, the decision is that the model is not good enough, so some choices must be revised. Typically, other model sets must be tested, or the conclusion might be that the data were not informative enough, so the experiment design must be reworked. This is why identification often is seen as an iterative problem with a so-called identification loop (for example, see [6, Fig. 17.1]).

User Guideline

Validating a model is a rather subjective and pragmatic problem. Check on a rich validation data set that covers the intended use of the model if the estimated model meets the user expectations.

WHY IS NONLINEAR SYSTEM IDENTIFICATION SO INVOLVED?

Nonlinear system identification can be much more involved than linear identification. Three aspects contribute to this observation:

1) Nonlinear models live on a complex manifold in a high-dimensional space, while linear models live on a simple hyperplane that is much easier to characterize.
2) Structural model errors are often unavoidable in nonlinear system identification, and this affects the

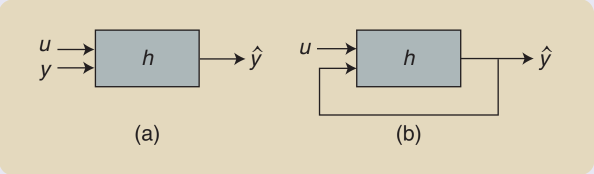


FIGURE S6 (a) A prediction model starts from the measured inputs u^{t-1} and outputs y^{t-1} to estimate the output $y(t)$. (b) In a simulation model, the measured output y^{t-1} is replaced by the estimated output $\hat{y}^{t-1}$.

What can be said about the error created by simulation and prediction models, respectively? Let h_0^p be the true prediction description of a system (based on the correct statistical properties). Let $h_0^s(U^t)$ be the simulation model created from this using (S19). Then $\hat{y}_0(t) = h_0^s(U^t)$ will be the noise-free output response to the input U^t, and

$$v(t) = y(t) - \hat{y}_0(t) \tag{S21}$$

will be the true output error disturbance.

If v is white noise, the true model structure is a simulation model, and $\hat{y}_0(t)$ are the optimal predictions. Otherwise, the disturbance $v(t)$ has a smooth or correlated behavior so that its future values can be partly predicted from its past. This property improves the quality of the prediction. The past values of $v(t)$ can be estimated as the difference between the past model and measured output values $\hat{v}(t) = y(t) - \hat{y}_0(t)$. This is the intrinsic idea used in the development of optimal prediction models. Whether it is worthwhile to further develop an improved predictor or not depends on the nature of the disturbance $v(t)$.

- $v(t)$ *is dominated by measurement or sensor noise*: Sensor or measurement noises are not related to the process of interest. The main goal is the elimination of this disturbance so that $y_0(t)$ is the signal of interest. Hence, a simulation model is the natural choice.
- $v(t)$ *is dominated by process noise*: Process noise is an intrinsic part of the system output. It models that part of the system output due to inputs that are not known to the user. In control applications, good predictions are important, so the noise disturbance must also be included in the model. Moreover, the past outputs are available to determine the next control action, which turns the prediction model into a natural tool for these applications.
- $v(t)$ *is dominated by structural model errors*: If the model set is not rich enough to capture the true system, structural model errors appear (see "Impact of Structural Model Errors"), which can also be represented as a disturbance.

A predictor can use past outputs to be informed about structural errors and find a better predictor than a simulation model. Consider a simple process $y(t) = y(t-1) + u(t)$ that ignores a structural error in the true system: $y(t) = y(t-1) + u(t) + C$ (C ignored in the model). The predictor $y(t) = y(t-1) + u(t)$ will have an error C, while the simulated output cannot avoid an error Ct.

This is one of the reasons why prediction is an easier task than simulation and why it is more demanding to obtain a small simulation error than a small prediction error.

three major choices (experiment design, model selection, and choice of the cost function).

3) Process noise entering before the nonlinearity requires new numerical tools to solve the optimization problem.

From Hyperplane to Manifold

A linear dynamic system is described by a linear relation between the lagged inputs and outputs, as with a simple two-tabs finite impulse response (FIR) model

Nonparametric Noise and Distortion Analysis Using Periodic Excitations

Here, tools will be presented that allow the user to detect and analyze the presence of nonlinear distortions during the initial tests. Using a well-designed periodic excitation [see (26) in the section “Experiment Design”], the frequency response function of the best linear approximation (see “Linear Models of Nonlinear Systems”), the power spectrum of the disturbing noise, and the level of the nonlinear distortions will be obtained from a nonparametric analysis without any user interaction. The user can set the desired frequency resolution and the desired power spectrum of the excitation signal. The phase will be chosen randomly on $[0, 2\pi)$. This article only gives a brief introduction. See [30] and [150] for a more extensive discussion and [90] for measurements under nonlinear closed-loop conditions.

THE RESPONSE OF A NONLINEAR SYSTEM TO A PERIODIC EXCITATION

A linear time-invariant system cannot transfer power from one frequency to another, while a nonlinear system does. Consider a nonlinear system $y = u^{\alpha}$, excited at the frequencies $\pm\omega_k, k = 1, \ldots, F$. The frequencies at the output are given by making all possible combinations of α frequencies, including repeated frequencies, selected from the set of $2F$ excited frequencies (see also Figure S7 for an illustration of a sine excitation with frequencies $\{-1, 1\}$) [30]:

$$\sum_{i=1}^{\alpha} \omega_{k_i}, \quad \text{with } \omega_{k_i} \in \{-\omega_F, \ldots, -\omega_1, \omega_1, \ldots, \omega_F\}. \tag{S22}$$

For example, for the quadratic kernel ($\alpha = 2$) excited by $u(t) = \sin(\omega t)$, with $\omega = 1$, it follows that $\omega_{k_i} \in \{-1, 1\}$. Making all combinations of two frequencies in this set results in the output frequencies: $\{1+1=2,\ 1+(-1)=0,\ -1+1=0,\ -1+(-1)=-2\}$. Using Volterra series models (S68) [28], this result can be generalized to dynamic fading-memory systems that include discontinuous nonlinear systems [30], [123], [151]. Chaotic systems are excluded from this study because these have no periodic output for a periodic input.

In the next section, the nonlinear distortions will be detected using a multisine excitation where some amplitudes U_k in (26) are set to zero for a well-selected set of frequencies.

DETECTION AND CHARACTERIZATION OF THE NONLINEAR DISTORTIONS

Sine Test

The simplest nonlinearity test is a sine test. As shown in Figure S7, nonlinear operations like x^2 or x^3 create harmonic components (S22) that can be detected at the output of the system and reveal the nonlinear behavior of the system. This result can be generalized to dynamic nonlinear systems. The sine test method is very popular in, for example, mechanical engineering. To speed up the measurement, the sine is replaced by a sweeping sine [23], [79]. A sine test is not very robust because the higher harmonics that indicate the presence of nonlinear behavior can be amplified or attenuated by the linear dynamics of the system. In highly resonating or bandpass systems, the presence of nonlinearities can be strongly underestimated. More robust tests were developed using multisine excitations (26).

Multisine Test

Using well-designed multisines, the nonlinear sine test is made more robust (for a more reliable estimate of the nonlinear level) and sped up [79]. The basic idea, illustrated in Figure S8, is quite simple and starts from a multisine (26) that excites a well-selected set of odd frequencies [which correspond to odd values of k in (26)]. This excitation signal is applied to the nonlinear system under test. Even nonlinearities appear at the even frequencies because an even number of odd frequencies are added together. Odd nonlinearities are present only at the odd frequencies because an odd number of odd frequencies

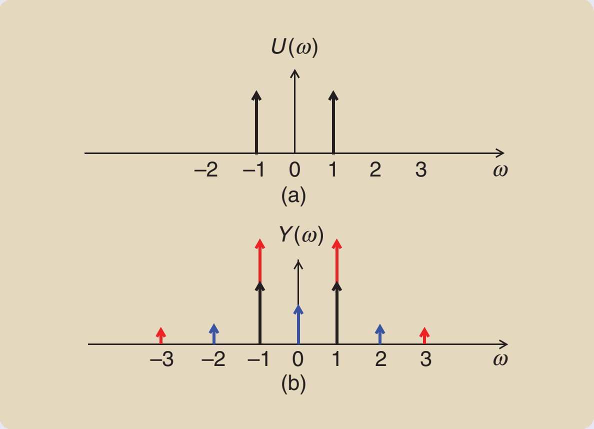


FIGURE S7 The detection of a nonlinear behavior using a sine excitation. (a) The input spectrum shows two spikes at the positive and negative excitation frequency. (b) The output spectrum shows the linear (black), quadratic (blue), and cubic (red) contributions of the output. At frequencies 2 and 3, the presence of even and odd nonlinearities is detected.

$$\hat{y}(t) = a_1 u(t-1) + a_2 u(t-2), \tag{3}$$

which is shown in Figure 3(a). The output is confined to a hyperplane in the 3D space. For a nonlinear FIR (NFIR) model, the relation can become arbitrary complex

$$\hat{y}(t) = h(u(t-1), u(t-2)), \tag{4}$$

as shown in Figure 3(b). The task is to estimate this surface. It is clear that the linear hyperplane can be characterized with only a few points, while it is impossible to tell anything

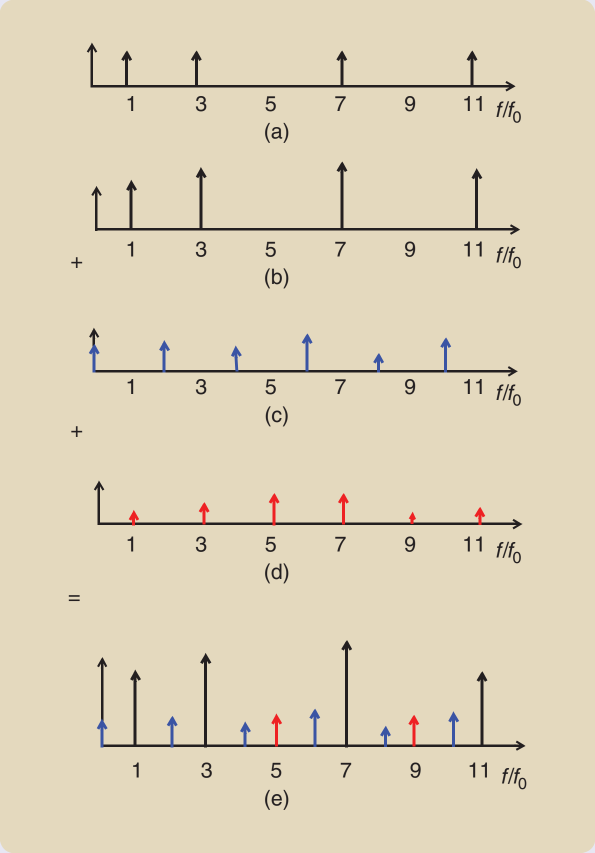


FIGURE S8 The design of a multisine excitation for a nonlinear analysis [79]. (a) The selection of the excited frequencies at the input. The output contributions: (b) linear, (c) even, and (d) odd. (e) The total output. For simplicity, only the positive frequencies are shown.

are added together. At the odd frequencies that are not excited at the input, the odd nonlinear distortions become visible at the output because the linear part of the model does not contribute to the output at these frequencies (for example, frequencies 5 and 9 in Figure S8). By using a different color for each contribution, it is easy to recognize these in an amplitude spectrum plot of the output signal.

DISTURBING NOISE CHARACTERIZATION

In the next step, the disturbing noise analysis is completed. By analyzing the variations of the periodic input and output signals over the measurements of the repeated periods, the sample mean and the sample (co)variance of the input and output disturbing noise is calculated, as a function of the frequency. Although the disturbing noise varies from one period to the other, the nonlinear distortions do not. Thus, they remain exactly the same. This results in the following simple procedure. Consider the periodic signal $u(t)$ in Figure S9, which is measured over P periods. For each subrecord, corresponding to a period, the discrete Fourier transform (DFT) is calculated using the fast Fourier transform algorithm, resulting in the DFT spectra of each period $U^{[l]}(k)$, $Y^{[l]}(k)$, for $l = 1, \ldots, P$. The sample means $\hat{U}(k)$, $\hat{Y}(k)$ and noise (co)variances $\hat{\sigma}^2_{\hat{U}}(k)$, $\hat{\sigma}^2_{\hat{Y}}(k)$, $\hat{\sigma}^2_{\hat{Y}U}(k)$ at frequency k are then given by

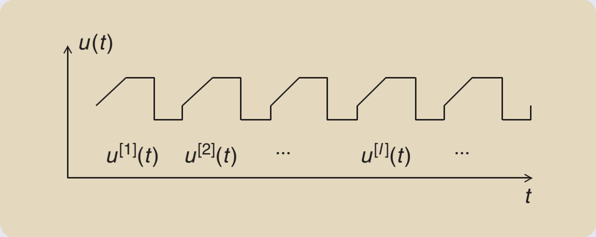


FIGURE S9 The calculation of the sample mean and variance of a periodic signal [30].

$$\hat{U}(k) = \frac{1}{P}\sum_{l=1}^{P} U^{[l]}(k) \quad \hat{Y}(k) = \frac{1}{P}\sum_{l=1}^{P} Y^{[l]}(k)$$

and

$$\begin{aligned}
\hat{\sigma}^2_{\hat{U}}(k) &= \frac{1}{P-1}\sum_{l=1}^{P} |U^{[l]}(k) - \hat{U}(k)|^2, \\
\hat{\sigma}^2_{\hat{Y}}(k) &= \frac{1}{P-1}\sum_{l=1}^{P} |Y^{[l]}(k) - \hat{Y}(k)|^2, \\
\hat{\sigma}^2_{\hat{Y}U}(k) &= \frac{1}{P-1}\sum_{l=1}^{P} (Y(k) - \hat{Y}(k))(U(k) - \hat{U}(k))^H, \qquad \text{(S23)}
\end{aligned}$$

where $(.)^H$ denotes the complex conjugate. The variance of the estimated mean values $\hat{U}(k)$ and $\hat{Y}(k)$ is $\hat{\sigma}^2_{\hat{U}}(k)/P$ and $\hat{\sigma}^2_{\hat{Y}}(k)/P$, respectively. Adding all this information in one figure results in a full nonparametric analysis of the system with information about the system (the frequency response function), the even and odd nonlinear distortions, and the power spectrum of the disturbing noise, as shown in Figure 4 for the forced Duffing oscillator.

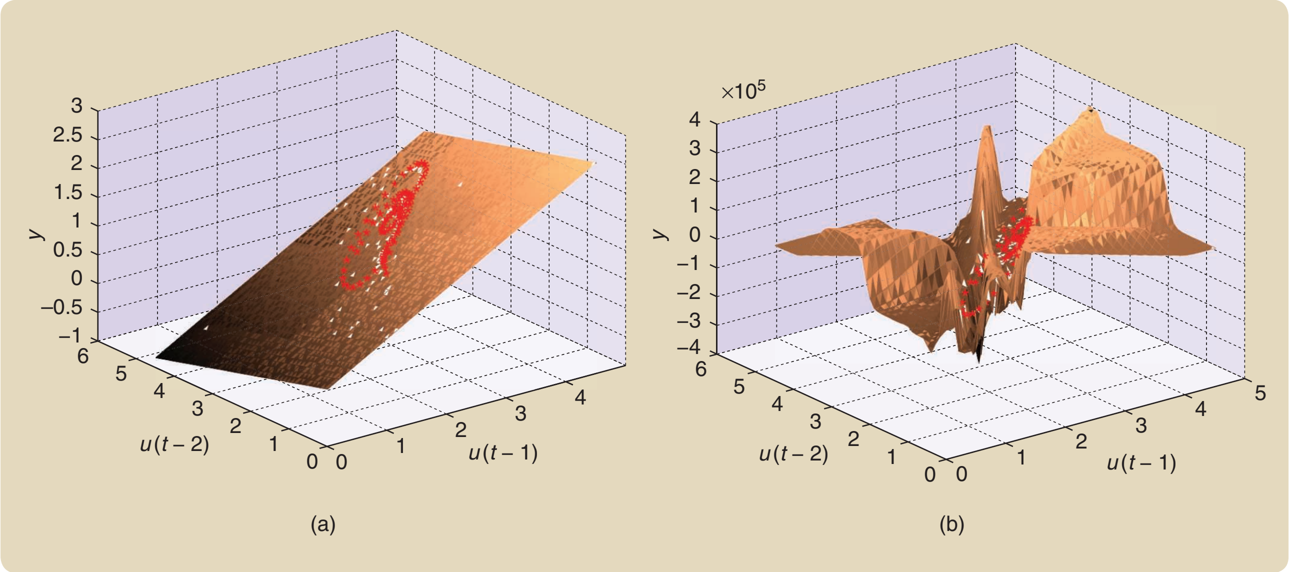


FIGURE 3 (a) The hyperplane $\hat{y} = a_1 u(t-1) + a_2 u(t-2)$ that characterizes a linear system is far less complex than (b) nonlinear models that live on a manifold $\hat{y} = h(u(t-1), u(t-2))$ in the input–output space. This affects the whole identification process from experiment design and selection of the model structure to the choice and minimization of the cost function and generating initial values. The complexity of the problems to be solved grows quickly with increasing dimensionality, for example, the number of delayed inputs and outputs.

about the complex manifold outside the domain where the function is sampled.

This immediately reveals a number of issues in nonlinear system identification that are less pronounced (or not even present) in linear system identification. First, experiment design will be extremely important because it should be guaranteed that the full domain of interest is covered. Extrapolation of the model should be avoided at all costs, unless there is physical insight that provides a natural description of the manifold. Second, finding the parameters that describe the manifold often results in a highly nonlinear optimization problem. Good initial values are needed to ensure that the global minimum is reached. In practice, this is often impossible, and the user must be satisfied with a good local minimum. Third, because the manifold can be very complex, it is often not possible to propose a model structure flexible enough to reproduce it exactly. This leads to the presence of structural model errors. These affect the whole identification process.

User Guideline

In summary, nonlinear systems are intrinsically more involved and complex than linear systems. This affects the experiment design and the model selection of the system identification process.

System Identification in the Presence of Structural Model Errors

As explained in the previous section, it is hard to avoid structural model errors in nonlinear system identification. In "Impact of Structural Model Errors," a formal definition is given of structural and random model errors, followed by a discussion of how to deal with structural model errors that dominate the noise disturbances. Special attention is paid to the user choice of how to shape the structural model errors, and consideration is given to the impact of structural model errors on the variance estimate of the model [69]–[75].

User Guideline

In summary, be aware that there is a high risk for dominating structural model errors over the noise disturbances in nonlinear system identification, which makes some classical choices and results of the linear identification theory less relevant. Proper actions are needed to address this new situation.

Impact of Process Noise on the System Identification Problem

The output of a nonlinear system depends not only on the known or measured inputs; the system can also be affected by signals that are not known to the user. These unknown inputs $w(t)$ are called *process noise*. While the measurement noise $v(t)$ does not affect the evolution of the system, process noise does, as is clearly seen in the state-space representation of a nonlinear system

$$\begin{aligned} x(t+1) &= f(x(t), u(t), w(t)), \\ y(t) &= h(x(t), u(t)) + v(t). \end{aligned} \tag{5}$$

Process noise can have a structural impact on the behavior of a system because it also affects the system's internal signals and not only the measurements. To proceed, we can consider, without loss of generality, the simple static nonlinear system

$$\begin{aligned} y(t) &= (u(t)+w(t))^2 + v(t) \\ &= u(t)^2 + (w(t)^2 + 2w(t)u(t)) + v(t). \end{aligned} \tag{6}$$

The effect of the process noise $w(t)$ at the output is $w(t)^2 + 2w(t)u(t)$. It depends on the input and does not have typical noise properties (zero mean and stationarity). Its mean value is different from zero, and the variance depends on $u(t)$. Moreover, it is shown in (S34) that the apparent gain of the system can also change for odd nonlinearities.

This simple example illustrates that the presence of process noise significantly increases the complexity of the system identification problem. If the process noise enters before the nonlinearity, its effect on the output is affected by the nonlinear operations, so that it is no longer realistic to use the Gaussian framework to formulate the cost function. Unknown distributions are faced that depend on the input and model parameters. For that reason, the output error framework that assumes that all noise enters at the output of the system must be abandoned, and a generalized multivariate probabilistic framework is needed. The complexity of the methods to solve these problems goes far beyond the linear system identification methods, and a completely new set of tools is required [41]–[56]. It is important to detect the presence of process noise that passes through the nonlinearity and select the proper tools when needed. This is discussed in detail in "Process Noise in Nonlinear System Identification" and "Identifying Nonlinear Dynamical Systems in the Presence of Process Noise."

User Guideline

Check if process noise passes through the nonlinearity and select the proper tools, as needed.

GOAL OF THE NONLINEAR SYSTEM IDENTIFICATION PROCESS

The goal of the modeling effort strongly affects the complexity of the system identification process. The issues to be addressed are simulation or prediction models, physical models or black-box models, and application-driven models. All of these aspects are briefly discussed next.

Models for Simulation or Models for Control?

Prediction Model

In layman's terms, a prediction model estimates the output of the system one step ahead at time $t+1$, using the measured input up to time $t+1$ and the measured outputs up to time t. Prediction models are central to modern control applications, where essentially the predicted output is controlled [22].

Simulation Model

The alternative is that the measured outputs are not used at all, but the output is calculated from inputs only. This is called a simulation model, and it can be used to simulate the behavior of the system for new inputs. These models are useful to test what happens in new situations, design systems and controllers, and mimic physical systems.

It is much harder to obtain a good nonlinear simulation than a prediction model. Simulation models can become unstable, and it is harder to obtain small structural model errors than it is for one-step-ahead prediction models. Even a very simple linear model can often provide a good one-step-ahead prediction if the sample frequency is high enough. A detailed discussion is given in "Simulation Errors and Prediction Errors." Figure S21 shows the results of linear and nonlinear simulation and prediction models for the forced Duffing oscillator.

User Guideline

Do not spend effort to obtain a complex simulation model if a simple prediction model can do the job. However, keep in mind that a good prediction can fail completely to generate a reliable simulation.

Physical or Black-Box Models?

A physical model is built on a deep insight into the internal behavior of the system. Alternatively, the simple use of physical insight can be used to guide the model process. Eventually, in both cases, the model depends upon a number of parameters that can be obtained from dedicated measurements (like the friction coefficient of a wheel on the road). These models are highly preferred in the industry, but they can be very expensive to construct and difficult to use in real-time control. For that reason, such models are often not affordable.

Black-box models become very attractive when a physical model is too expensive to develop. They describe the input–output behavior of a system and are tuned directly from experimental data. Such models are simple to use and can be applied in real-time computations. Of course, there are many possibilities between both extremes. These are discussed in "The Palette of Nonlinear Models" section.

User Guideline

Select the model level that best balances the need for physical insight/behavior insight and the expenses to build and use the model.

Models Constrained for Particular Applications

The ideal model should cover all possible applications, providing good output simulations for all possible excitations. Of course, this is an unattainable ideal, and a more restricted goal should be defined. This is done while keeping the application in mind: the model should cover those situations and signals that are important for the actual application and not more than that [76]. If a system will be mainly driven by low-frequency sine excitations, then no effort should be spent to also develop the model for wideband random-noise excitations. This can again significantly reduce the modeling effort.

User Guideline

Carefully select the domain and application of interest and focus the modeling effort on it.

Process Noise in Nonlinear System Identification

Process noise $w(t)$, as defined in (5), affects not only the measurements but also the system's internal signals. In this sidebar, the impact of process noise is studied in more detail. The process noise $w(t) = H_w(q)e_w(t)$ is considered to be a zero-mean, white or colored noise variable.

PROCESS NOISE IN LINEAR SYSTEMS

It is well known that, for linear systems, the effects of process noise can be collected at the output as an additive noise term $H_w(q)w(t)$, where it is combined with the measurement noise $v(t) = H_v(q)e(t)$ into one noise term

$$\begin{aligned} y(t) &= y_0(t) + H_w(q)e_w(t) + H_v(q)e_v(t) \\ &= y_0(t) + H(q)\tilde{e}(t) \\ &= y_0(t) + \tilde{v}(t). \end{aligned} \tag{S24}$$

The disturbances $e_w(t)$, $e_v(t)$ are mutually independently distributed and independent of the input $u(t)$. Under these conditions, the combined effect of the process and the measurement noise result in a zero-mean distributed output disturbance that is independent of the input.

Process Noise in Nonlinear Systems

In nonlinear systems, process noise can have a structural impact on the identified models, as was illustrated in (6) and (S36). Zero-mean process noise can change, as in the linear gain of a nonlinear system, and the apparent disturbances at the output can become nonstationary and depend upon the input, as discussed in more detail later in this sidebar.

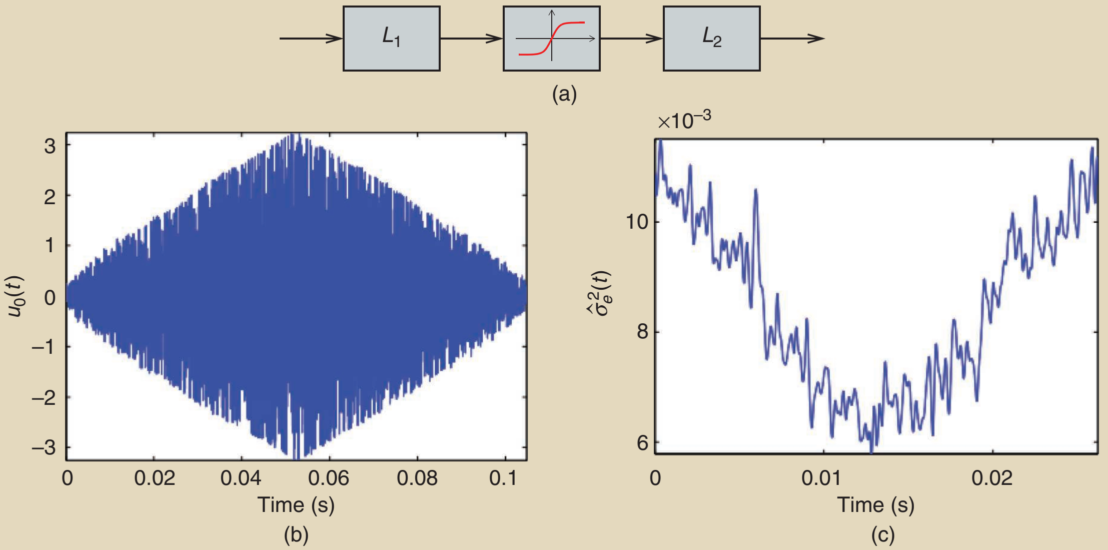


FIGURE S10 The detection of process noise [38]. (a) A Wiener–Hammerstein system with a saturating nonlinearity sandwiched between two low-pass linear filters L_1 and L_2 receives (b) a nonstationary input that is periodically repeated and applied to it. The process noise enters before the nonlinearity. This system was studied and discussed in detail at the nonlinear benchmark workshop [39]. The presence of the process noise is revealed through its nonstationary behavior at the output. (c) The smoothed variance of the output noise varies over time. Observe that it becomes small where the excitation is large and large where the excitation is small, pointing to process noise that is injected before a saturating nonlinearity.

LINEAR OR NONLINEAR SYSTEM IDENTIFICATION: A USER'S DECISION

As discussed previously, nonlinear system identification is much more involved than the identification of a linear system. The experiment design is more tedious, the model selection much more involved, and the parameter estimation more difficult. For that reason, moving from the well-established linear identification tools toward the more advanced nonlinear identification methods is an important decision that significantly affects the cost of the identification effort (time, money, and experimental resources), and thus the decision-making process should be well informed. Is a nonlinear model needed to reach the required model quality? Is the quality of the data good enough to improve the results of a linear identification approach? How much can be gained if a linear model is replaced by a nonlinear

PROCESS NOISE: CURSE OR BLESSING?

Curse

In most applications, process noise is extremely disturbing. It is more difficult to control a system in the presence of process noise. The system identification methods also become much more involved, as explained later. For this reason, process noise is mostly considered as a highly annoying effect.

Blessing

In some applications, process noise is used to reduce or linearize the averaged effect of abrupt nonlinearities, which is called dithering. The desired effects of dithering are augmenting the linearity of the open- or closed-loop system, increasing the robustness and asymptotic stability [34]–[36]. Dithering can be used to reduce the effect of Coulomb friction, dead zones in hydraulic valves, and hysteresis effects. However, this results in an increased wear due to the rapid motions [36]. Dithering is also employed in digital instrumentation and data acquisition systems to improve their measurement-related characteristics. A typical example is the use of dithering in an analog-to-digital converter to improve the resolution, dynamic range, and spectral purity below the quantization level [37].

Detection of the Presence of Process Noise

The dependence of the process noise output $y_p(t)$ on the input $u(t)$ can be used to detect it, even in the presence of measurement noise. In [38], periodic nonstationary input signals are used. Assuming that a periodic input results in a periodic output, the process and measurement noise $\tilde{v}(t)$ can be estimated as the nonperiodic part of the output. Next, the variance $\sigma_{\tilde{v}}(t)$ is estimated. Assuming that the measurement noise is stationary, the presence of the process noise is revealed by a time-varying variance, as illustrated in Figure S10.

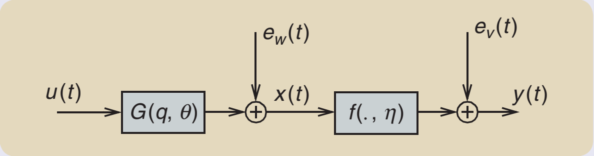


FIGURE S11 A Wiener system with white process noise $e_w(t)$ and measurement noise $e_v(t)$. The signals $u(t)$ and $y(t)$ are available to identify the system, and the nuisance signal is not measured.

Impact of Process Noise on the System Identification Problem

The presence of process noise increases the complexity of the system identification problem significantly. The nonlinear operations change the distribution of the process noise $w(t)$ and make a least-squares problem formulation at the output of the system quite inefficient and even strongly biased. For that reason, the output error framework, which assumes that all noise enters at the output of the system, has to be abandoned. Instead, the procedure should start from the joint (Gaussian) distribution of $w(t)$, $v(t)$. The subsequent discussion focuses on the identification of a Wiener system to set the ideas. Following the ideas in [40], the likelihood function can be easily generated using the intermediate nuisance variable $x(t)$ in Figure S11 from this probability density function:

$$p_y(\theta,\eta) = \int_{x\in\mathbb{R}} p_v(y - f(x,\eta))\, p_w(x - G(q,\theta)u)\, dx. \tag{S25}$$

The calculation of this integral becomes very difficult because it is not possible to eliminate $x(t)$, $t = 1, \ldots, N$ analytically with N data points. Dedicated numerical methods are developed to address these high-dimensional integrals. See (S38) in "Identifying Nonlinear Dynamical Systems in the Presence of Process Noise" for an initial introduction.

SUMMARY

- *Process noise*: output of a nonlinear system may no longer be Gaussian distributed and independent of the input.
- *Detection*: When applying nonstationary periodic excitations, it is possible to detect the presence of process noise.
- *Identification*: If no process noise is detected, the simpler output error formulation can be used to identify the nonlinear model. If the process noise is large, more advanced identification approaches are needed to guarantee consistent and efficient estimates.

Linear Models of Nonlinear Systems

Nonlinear models are clearly much more versatile and complicated than linear models. A general linear model can be represented and fully characterized by a *transfer function* $G(s)$, which is the Laplace transform of the model's impulse response $g(\tau)$

$$G(s)=\int_{0^-}^{\infty} g(\tau)e^{-s\tau}d\tau. \tag{S26}$$

In discrete time, the transfer function $G(z)$ is the Z-transform of the impulse response. Because of the simplicity of linear models, it is tempting and common to complete linear approximations of nonlinear systems. Several linear models can possibly capture different aspects of the nonlinear system. Two ways of defining linear approximations are discussed here.

LINEARIZATION AROUND AN EQUILIBRIUM

Consider a general nonlinear state-space model

$$\dot{x}(t)=f(x(t),u(t)), \tag{S27a}$$
$$y(t)=h(x(t)). \tag{S27b}$$

Suppose the input is constant, $u(t)=u^*$, and there is a corresponding state equilibrium x^*; $f(x^*,u^*)=0$. Define the deviations

$$\Delta u(t)=u(t)-u^*, \tag{S28a}$$
$$\Delta x(t)=x(t)-x^*, \tag{S28b}$$
$$\Delta y(t)=y(t)-y^*=y(t)-h(x^*). \tag{S28c}$$

By expanding the nonlinear functions f and h in a Taylor series around x^*, u^*, and y^* and neglecting terms of higher order, we obtain a linear state-space equation

$$\Delta\dot{x}(t)=A\Delta x(t)+B\Delta u(t), \tag{S29a}$$
$$\Delta y(t)=C\Delta x(t), \tag{S29b}$$
$$A=\frac{\partial}{\partial x}f(x,u)|_{x^*,u^*},\quad B=\frac{\partial}{\partial u}f(x,u)|_{x^*,u^*}, \tag{S29c}$$
$$C=\frac{\partial}{\partial x}h(x,u)|_{x^*,u^*}. \tag{S29d}$$

In the vicinity of the equilibrium, the nonlinear system (S27) can be approximated by the linear transfer function $G(s)=C(sI-A)^{-1}B$. This is a well-known and commonly used linearization. Corresponding expressions apply in discrete time.

STOCHASTIC LINEARIZATION

Suppose the nonlinear system (S27) is excited by an input u with a spectrum $\Phi_u(\omega)$, and the cross spectrum between output and input $\Phi_{yu}(\omega)$ is well defined. The *second-order properties* of the input and output signals (that is, the covariance functions and spectra) are thus well defined. For this input, define the linear model [83]

$$G_{\mathrm{BLA}}(\omega)=\Phi_{yu}(\omega)[\Phi_u(\omega)]^{-1}, \tag{S30}$$

which has the same second-order properties as the nonlinear system. By considering second-order signal properties (for this input), the nonlinear system (S27) thus cannot be distinguished from the linear model G_{BLA}, which are second-order equivalents [84].

This also means that, if a linear model is estimated with standard linear identification methods (that use only the second-order properties of the signals), the estimate will converge to G_{BLA}. The linear second-order equivalent is also called the *best linear approximation* (*BLA*). Note that the BLA of a nonlinear system will depend on the input signal spectrum. The definition of spectra does not require a stochastic setting. It is sufficient that the following limits exist (in discrete time):

$$R_{yu}(\tau)=\lim_{N\to\infty}\frac{1}{N}\sum_{t=1}^{N}y(t)u(t-\tau), \tag{S31a}$$
$$\Phi_{yu}(\omega)=\sum_{\tau=-\infty}^{\infty}R_{yu}(\tau)e^{-i\omega\tau}, \tag{S31b}$$

for the input–output signals [83] and correspondingly for Φ_u.

MORE ON THE BEST LINEAR APPROXIMATION GBLA

This section further analyzes the BLA model (see [79] for more details). The BLA G_{BLA} in (S30), represented by its impulse response $g_{\mathrm{BLA}}(t)$, or its frequency response function (FRF) $G_{\mathrm{BLA}}(\omega)$, is [30], [84], [88], the solution of

$$G_{\mathrm{BLA}}=\mathrm{argmin}_G E\{|y_0(t)-G(q)u(t)|^2\}, \tag{S32}$$

with q the shift operator for a discrete-time model. Similar expressions can be given for continuous-time models. All expected values $E\{\}$ are with respect to the random input $u(t)$. In most applications, the dc value of the input and output signal should be removed to obtain a model that is valid around a given set point.

The transfer function $G_{\mathrm{BLA}}(k)$ depends on the characteristics of the input signal. Changing the power spectrum or the amplitude distribution (for example, replacing a Gaussian by a uniform distribution) of the excitation will change the BLA [94], [95].

The Nonlinear Noise Source $y_s(t)$

The difference between the output of the nonlinear system and that of the BLA $y_s(t)=y(t)-G_{\mathrm{BLA}}(q)u(t)$ is called the stochastic nonlinear contribution or nonlinear noise. Although this name might be misleading (the error is deterministic for a given input signal), a stochastic contribution is preferred because it looks very similar to a noise disturbance for a random excitation [30], [87]. Because $y_s(t)$ is the residual of a least-squares fit, it is uncorrelated with the input. However, in general, it is still dependent on the input. The properties of G_{BLA} and Y_s are well known for Gaussian and Rieman-equivalent excitations [88]. Extensions to more general distributions are studied in [84]–[86], [94], and [95].

A New Paradigm

Combining both results, the output of the nonlinear system can be written as

$$\begin{aligned} y(t) &= y_0(t) + v(t), \\ y_0(t) &= G_{\mathrm{BLA}}(q)u(t) + y_S(t), \end{aligned} \tag{S33}$$

where $y_s(t)$ is uncorrelated but dependent on the input $u(t)$.

Experimental Illustration on the Duffing Oscillator

In this example, measurements on the forced Duffing oscillator are shown (see [79] for more details). The FRF is measured for four different excitation levels and shown in Figure S12. For each excitation level, the FRF is averaged over 50 realizations of the input signal to obtain a smoother result. Two observations can be made: 1) the resonance frequency shifts to the right for increasing excitation levels and 2) the measurements become noisier. Both effects are completely due to the nonlinear distortions.

BLA of a Static Nonlinearity (Simple Example)

The BLA of a static nonlinearity excited with a Gaussian excitation is a constant [82]. For example, the BLA of $y = u^3$ is

$$\begin{aligned} y_0(t) &= u(t)^3 \quad \text{with } u \sim N(0, \sigma_u^2), \\ G_{\mathrm{BLA}} &= E\{y_0(t)u(t)\}/E\{u(t)^2\} = 3\sigma_u^2. \end{aligned} \tag{S34}$$

Observe that

$$y_s(t) = y_0(t) - G_{\mathrm{BLA}}u(t) = u(t)^3 - 3\sigma_u^2 u(t) \tag{S35}$$

is uncorrelated with but dependent upon u. This is a generally valid observation.

Impact of Process Noise on the BLA

Process noise $w(t)$ coming into the system before the nonlinearity creates mixing terms with the input signal so that, at the output of the system, the process noise contributions are no longer independent of the input. This will also affect the BLA, and a generalization of the framework is needed. See [89] for a full discussion and an extension of the simple example (S34). Assuming that the process noise $w(t)$ is independent of the input $u(t)$, the BLA of $y(t) = (u+w)^3$ with $u \sim N(0, \sigma_u^2)$, $w \sim N(0, \sigma_w^2)$ becomes

$$G_{\mathrm{BLA}} = E\{y(t)u(t)\}/E\{u(t)^2\} = 3\sigma_u^2 + 3\sigma_w^2. \tag{S36}$$

This shows that the averaged behavior of a nonlinear system is strongly affected by the presence of process noise entering the system before the nonlinearity.

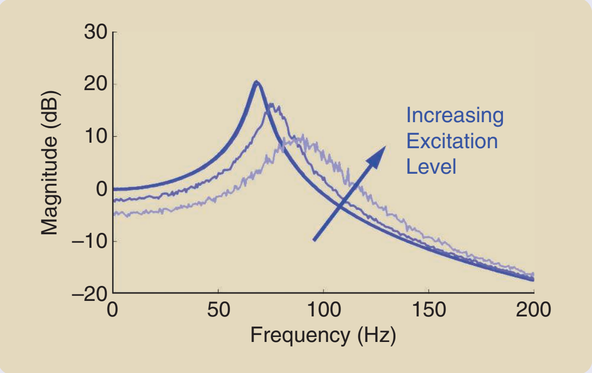


FIGURE S12 The measured frequency response function of the best linear approximation of the forced Duffing oscillator [79]. Observe that there is a systematic shift, and the disturbances grow with the excitation level. The shift is due to the systematic nonlinear contributions that create a shift in the dynamics of G_{BLA}. The increased noisy behavior is due to the stochastic nonlinearities y_s that grow with the excitation level.

SUMMARY

- *BLA* G_{BLA}: It is possible to identify a simplified representation (for example, the linear model G_{BLA}) for a nonlinear system.
- *Stochastic nonlinearities* y_s: For a random excitation, the structural nonlinear model errors (called stochastic nonlinearities y_s) look like noise. The function $y_s(t)$ is uncorrelated but not independent of the input $u(t)$. For an untrained user, it is quite hard to distinguish process or measurement noise from y_s (the nonlinear model errors). Different actions must properly address the two effects (see [79]).
- G_{BLA} *depends on the nature of the input*: The best simplified approximation (G_{BLA}) depends on the nature of the excitation. Changing the power spectrum or input distribution can change G_{BLA}.
- *Structure detection*: The variations of G_{BLA} for changing experimental conditions provide insight on the structure of the nonlinear system. For example, moving poles (resonances) are possible only for nonlinear closed-loop systems. See [191] for further details.
- *Process noise*: Unmeasured random inputs process noise into the system before the nonlinearity can systematically affect the output. The BLA, with respect to the known input, depends on the process noise properties.

one? Often, additional information is needed to address these questions.

If it is possible to apply periodic excitation signals, a full nonparametric analysis can be made that requires no user interaction, while the experimental cost with respect to a linear study remains almost the same (see "Nonparametric Noise and Distortion Analysis Using Periodic Excitations"). On the basis of the results, the user can detect the presence

Identifying Nonlinear Dynamical Systems in the Presence of Process Noise

A rather general formulation to be used when identifying nonlinear dynamical systems is arguably provided by the nonlinear state-space (NLSS) model that represents a system with input signal u_t and output signal y_t in terms of a latent Markovian state x_t

$$x_{t+1} = f(x_t, u_t; \theta) + w_t(\theta), \tag{S37a}$$

$$y_t = h(x_t, u_t; \theta) + v_t(\theta). \tag{S37b}$$

Here, the nonlinear functions $f(\cdot)$ and $h(\cdot)$ represent the dynamics and measurements, respectively. The variables w_t and v_t describe the process noise and measurement noise, respectively. Finally, the unknown static model parameters are denoted by θ, and the initial state is given by $x_0 \sim p(x_0)$ for some distribution $p(\cdot)$.

The problem to be solved is the identification of the unknown parameters θ in (S37) based on observed inputs $u^T = \{u_t\}_{t=1}^T$ and the corresponding outputs $y^T = \{y_t\}_{t=1}^T$. The maximum likelihood formulation can be written as

$$\hat{\theta} = \text{argmax}_\theta p(y^T; \theta), \tag{S38}$$

where the nature of the intractability of the likelihood is revealed by

$$p(y^T; \theta) = \prod_{t=1}^{T} p(y_t | y^{t-1}, \theta) = \prod_{t=1}^{T} \int p(y_t | x_t, \theta) p(x_t | y^{t-1}, \theta) dx_t. \tag{S39}$$

To stress that the process noise enters this integral, the following alternative integral can be considered:

$$p(y^T; \theta) = \prod_{t=1}^{T} \int p(y_t | x_t, \theta) p(x_t | x_{t-1}) p(x_{t-1} | y^{t-1}, \theta) dx_{t-1:t}, \tag{S40}$$

where it is clear that the process noise enters the integral via the term $p(x_t | x_{t-1}) = p_{w_t}(x_t - f(x_{t-1}, u_{t-1}; \theta))$.

More specifically, the challenge is that the predictive state distribution $p(x_t | y^{t-1}, \theta)$ cannot be explicitly computed, and approximations must be made. The sequential Monte Carlo (SMC) methods [41], namely, particle filters and smoothers, can be used to compute this distribution arbitrarily well. These methods were introduced in the beginning of the 1990s [42], yet it is still a research area with new results emerging. The idea behind these methods is to maintain an empirical distribution

$$\hat{p}(x_t | y_{1:t-1}) = \sum_{i=1}^{N} W_t^i \delta_{x_t^i}(x_t), \tag{S41}$$

comprised of samples $\{x_t^i\}_{i=1}^N$ (sometimes referred to as *particles*) and their corresponding weights $\{W_t^i\}_{i=1}^N$. Here, $\delta_{x_t^i}(x_t)$ denotes the Dirac delta. The SMC methods describe how to update these weights over time in such a way that the estimate (S41) converges to the true underlying weights as the number of samples $N \rightarrow \infty$. The theoretical basis underpinning these methods is by now quite extensive. An entry point into this literature is [41] and [43]. What is perhaps most relevant for the present discussion is that the SMC method is capable of producing *unbiased* estimators of the likelihood [43], [44]. The likelihood estimate is obtained by inserting (S41) into (S39).

Based on this, noisy unbiased estimates of the cost function in the maximum likelihood problem can be computed. As with any optimization problem—especially when faced with a stochastic optimization problem, as here—it is easier to solve if there are also gradients available. The SMC method can be used for this as well [45], [46]. Driven by deep learning, the state of the art for solving stochastic optimization problems is evolving rapidly [47].

As an alternative to this solution, the expectation maximization (EM) method [48] can be used to solve (S38) in the presence of process noise (see, for example, [49] and [50]). EM is an iterative algorithm based on a current iterate θ_k that computes an approximation of the so-called intermediate quantity

$$Q(\theta, \theta_k) = \int \log p(x^T, y^T; \theta) p(x^T | y^T, \theta_k) dx_{0:T}, \tag{S42}$$

which is then maximized to find the next iterate θ_{k+1}. This procedure is then repeated until convergence, and it is guaranteed to stop at a stationary point of the likelihood surface. The SMC method is key as well, in that it allows for the approximation of the smoothing distribution $p(x^T | y^T, \theta_k)$ in (S42) arbitrarily well.

The Bayesian approach is also highly interesting for nonlinear system identification in general [51] and when there is process noise present. The slight variation to the preceding is that the unknown parameters are assumed to be random variables, and rather than computing a point estimator like (S38), the posterior distribution $p(\theta | y^T)$ is computed. The breakthrough came in 2010 with the introduction of the so-called particle Markov chain Monte Carlo methods [52].

Tutorial overviews of how to identify NLSS models using SMC are provided by [53] and [54]. Concrete system identification examples where there is significant process noise available are provided in [49], [55], and [56].

of nonlinearities, quantify their level, and find out if they are even or odd nonlinearities. With this information, the user can make a well-informed decision on what approach to use and how much can be gained by switching from linear to nonlinear modeling.

Detection, Separation, and Characterization of the Nonlinear Distortions and Disturbing Noise

In this article, only a basic introduction to nonlinear distortion analysis is given. A detailed theoretical analysis is provided in [30], and illustrations on practical examples (fighter aircraft, diesel engine [77], and industrial robot [78]) are discussed in [79]. In this article, the ideas are illustrated on the forced Duffing oscillator, which is examined in full detail in "Extensive Case Study: The Forced Duffing Oscillator." A detailed introduction to nonlinear distortion analysis is given in "Nonparametric Noise and Distortion Analysis Using Periodic Excitations."

Nonlinear Distortion Analysis

The basic idea is very simple and starts from a periodic input signal with period $T = 1/f_0$. Only a well-selected set of odd frequencies (odd means that f is an odd multiple of f_0) is excited. All other frequencies have zero amplitude (see Figure S8). This excitation signal is applied to the nonlinear system under test. Effects from even nonlinearities (the simplest even nonlinearity is $y = u^2$) appear at the even frequencies, while odd nonlinearities (like $y = u^3$) are present at only the odd frequencies (see [79] and the references therein). At the odd frequencies not excited at the input, the odd nonlinear distortions become visible at the output because the linear part of the model does not contribute to the output at these frequencies. These contributions are each plotted with a different color and become easy to recognize in an amplitude spectrum plot of the output signal. This is illustrated in Figure 4. The forced Duffing oscillator (see Figure S18) is excited at different excitation levels, and the output is plotted for the excited frequencies, even and odd nonlinearities, and disturbing noise level. For small excitation levels, the nonlinear distortions are at the 10% level (–20 dB below the output), while for the high excitation levels, the nonlinear distortions dominate the output.

Linear or Nonlinear Model?

On the basis of this information, the user can make a well-informed decision. For example, a linear model can be used if it is known that only small excitations will be applied and if 10% errors can be tolerated. However, for the large excitation levels, this is no longer an option because the nonlinearities are too large. In that case, a nonlinear model is needed.

Noise Floor

In Figure 4, the nonlinear distortions are more than 40 dB, or a factor of 100, above the noise level (see "Nonparametric Noise and Distortion Analysis Using Periodic Excitations"), also called the noise floor of the measurements [80]. The noise floor is the maximum power level over a given frequency band in the frequency domain of components that are not due to the applied signal, harmonics, or spurious signals (these are signals that are out of the control of the user, for example, a disturbance picked up from the mains). The noise floor is estimated by analyzing the variations

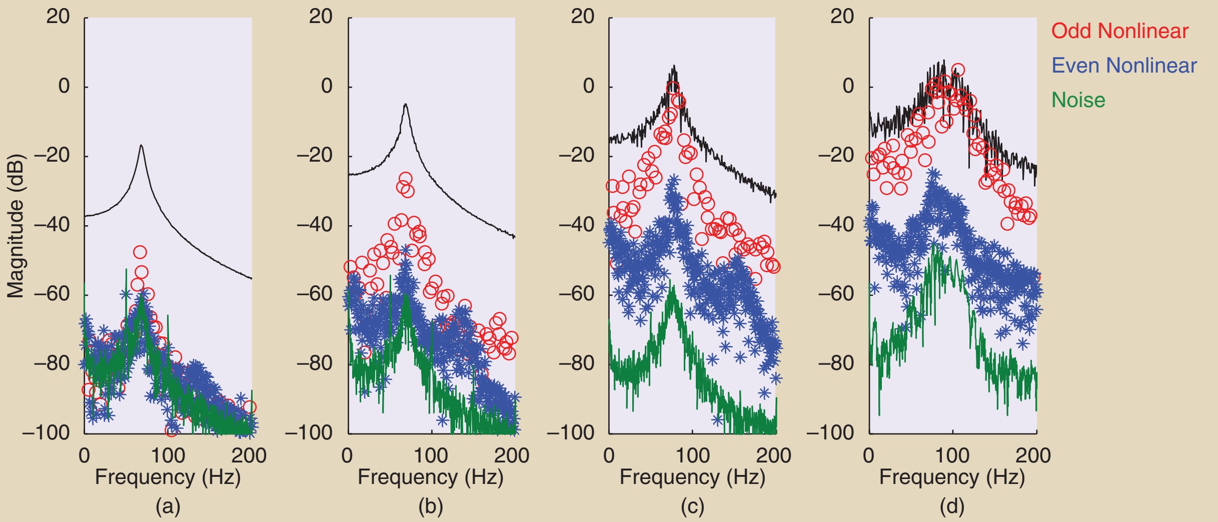


FIGURE 4 A nonparametric analysis of the nonlinear distortions on the forced Duffing oscillator (see Figure S18) [79]. The system is excited at a well-selected set of frequencies. The nonlinearities become visible at the unexcited frequencies. Black dots: output at the excited frequencies; red circles: odd nonlinearities; blue stars: even nonlinearities; green line: disturbing noise level. The excitation level is growing from (a) to (d). Observe that the level of the nonlinear distortions expands with the excitation level, and the disturbing noise level remains almost constant.

Black-Box Model Complexity

Black-box modeling is a very flexible method that requires little or no physical insight of the user. This comes with the cost of an exploding number of model parameters for a growing complexity, leading to an increased risk of overfitting [6], [7], [30]. Regularization and data-driven structure retrieval tools are developed to keep this number of growing parameters or their effect on the modeled output under control. Both approaches are discussed next.

REGULARIZATION

The easiest approach to obtain a simplified model is to set the least significant parameters equal to zero in a model-pruning step using manual trial-and-error methods. Regularization techniques replace manual tuning by automatic procedures [58], [67]. The basic idea is to add an additional term $R(\theta)$ to the cost function (S11), imposing an extra constraint on the parameters

$$V_N(\theta) = \frac{1}{N}\sum_{t=1}^{N}\frac{1}{2}\epsilon^2(t,\theta) + R(\theta). \tag{S43}$$

The choice of R sets the behavior of the regularized solution, leading to sparse or smooth solutions.

Sparse Models

Letting $R(\theta) = \lambda|\theta|$ gives LASSO [57], a method that sets as many parameters as possible equal to zero, which leads to sparse models.

Smooth Models

A quadratic regularization $R(\theta) = \theta^T P_\theta^{-1}\theta$ leads to a milder regularization than LASSO. The regularization term pulls the estimates toward zero, resulting in a reduced output variance at the cost of an increased bias. An optimal bias/variance balance is made that minimizes the mean square error of the output, as illustrated in Figure S13.

The success of this approach strongly depends on a proper choice of the regularization matrix P that reflects the additional user knowledge or user desires (also called prior information), which is added on top of the data. It can be based on physical insight or imposed on the user's desire for a smooth solution [58]. Observe that, in this approach, the number of parameters is not reduced, but, instead, their freedom to vary independently is restricted, leading to a smaller effective number of parameters. Imposing smooth and exponentially decaying solutions is illustrated for nonparametric Volterra models (36) in the section "Example 5(a): Black-Box Volterra Model of the Brain" [56].

DATA-DRIVEN STRUCTURE RETRIEVAL

An alternative approach to reduce the number of model parameters is to impose more structure on the model. In black-box modeling, the structural information should be retrieved from the data rather than using physical information. The explosive growth of the number of parameters in nonlinear black-box modeling is due to the parameterization of the multivariate nonlinear function (8) present in every nonlinear model. Retrieving more efficient representations of the function $F \in R^{n_q \times n_p}$ is key to reducing the number of parameters and controlling the model flexibility. Recently, decoupling methods were developed that allow the multivariate function F to be written as a combination of linear transformations and a well-selected set of single-input, single-output (SISO) nonlinear functions, as shown in Figure S14. In "Decoupling of Multivariate Polynomials," a tensor-based decoupling approach is explained in more detail. A decoupled representation offers major advantages:

- The combinatorial growth of the number of parameters is reduced to a linear growth as a function of the complexity. For example, if F is a multivariate polynomial of degree d, the number of parameters drops from $n_q O(n_p^d)$ to $n_q O(rd)$, with r the number of internal single-input, single-output branches.
- The decoupled representation of F is easier to interpret, giving more intuitive access to the nonlinear behavior of the system. For example, it is much simpler to plot a set of SISO nonlinear functions than to make a graphical representation of a high-dimensional multivariate nonlinear function because only slices of the multivariate function can be shown.

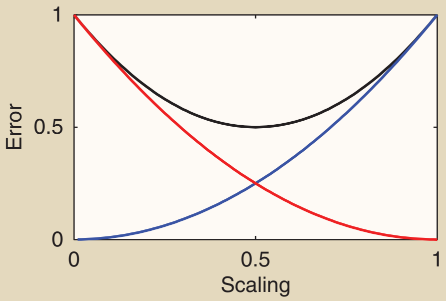


FIGURE S13 The bias and variance tradeoff of a scaled estimator. The evolution of the total mean square error (MSE) (black), squared bias (red), and variance error (blue) as a function of the scaling factor. An error e can always be written as the sum $e = b + v$ of its mean value $b = E\{e\}$ (called the *bias*) and the remaining part $v = e - b$, with variance σ^2. The total MSE is $e_{\mathrm{MS}} = b^2 + \sigma^2$. Depending on the preference, either the bias b or the MSE e_{MS} should be as small as possible. It is always possible to scale an unbiased estimator (no bias present) toward zero, such that e_{MS} drops. This is illustrated on a simple scalar example. Assume that $\hat{\theta}$ is an unbiased estimate of the true parameter $\theta_0 = 1$, with variance $\sigma^2 = 1$. Consider next the scaled estimator $\tilde{\theta} = \lambda\hat{\theta}$. The bias of $\tilde{\theta}$ is $b = (1-\lambda)$, and the variance $\tilde{\theta}$ is $\sigma_{\tilde{\theta}}^2 = \lambda^2$. The MSE becomes $e_{\mathrm{MS}} = (1-\lambda)^2 + \lambda^2$. Black: MSE; red: bias error; blue: variance error.

- Tuning the number of branches in the decoupled representation not only is a tool to reduce the flexibility of the model but also employed to make a user-selected balance between structural model errors and model complexity, which opens the possibility for model complexity reduction.

Applications

The decoupling approach can be applied to a variety of problems. Equation (S51) in "Extensive Case Study: The Forced Duffing Oscillator," gives a detailed illustration of the decoupling approach. An early application of the decoupling strategy [59] proposed a tensor-based decoupling method to decouple the Volterra kernels of different degrees separately. Next, this idea was further refined and applied to the identification of parallel Wiener and parallel Wiener–Hammerstein block-oriented models [60]–[62]. The full decoupling method [63] described in this sidebar is used in [65] to decouple a nonlinear state-space model for the Bouc–Wen hysteresis problem. Recently, the ideas were further generalized to apply the decoupling idea to the pruning of polynomial nonlinear autoregressive exogenous (NARX) models [26]. Eventually, the decoupling approaches were also directly applied to the design of decoupled controllers [68].

Decoupling of Multivariate Polynomials

Representations that increase the structural insight and reduce the number of model parameters are most welcome. The approach briefly described here is discussed in full detail in [63]. The starting point is a set of multivariate basis functions, for example, polynomials, that must be unraveled into a simplified structure. The cross-links among input variables are decoupled into single-variable functions by considering linear transformations at the input and output of the static nonlinearity. These transformations reveal internal variables between which univariate relations hold. Consider the variables $\mathbf{p}$ and $\mathbf{q}$ and the multivariate vector function $\mathbf{f}$ with proper dimensions:

$$\mathbf{q} = \mathbf{f}(\mathbf{p}) = W\mathbf{g}(V^{\top}\mathbf{p}). \tag{S44}$$

The entry i of $\mathbf{g}$ is a univariate function $\mathbf{g}_i = g_i(x_i),\ i = 1, \ldots, r$, with $\mathbf{x} = V^{\top}\mathbf{p}$. The model can be given a physical/intuitive interpretation while, at the same time, the number of parameters decreases. A graphical representation is given in Figure S14. Each univariate function is a branch, and r is the number of branches used in the decoupled presentation. In [64], an exact decomposition method is proposed to obtain a decoupled representation.

Example: Exact Decomposition of a Multivariate Polynomial

Consider the polynomials $f_1(p_1, p_2)$ and $f_2(p_1, p_2)$ of total degree $d = 3$, given as

$$\begin{aligned} q_1 = f_1(p_1, p_2) &= 54p_1^3 - 54p_1^2p_2 + 8p_1^2 + 18p_1p_2^2 \\ &\quad + 16p_1p_2 - 2p_2^3 + 8p_2^2 + 8p_2 + 1, \end{aligned} \tag{S45a}$$

$$\begin{aligned} q_2 = f_2(p_1, p_2) &= -27p_1^3 + 27p_1^2p_2 - 24p_1^2 - 9p_1p_2^2 - 48p_1p_2 \\ &\quad -15p_1 + p_2^3 - 24p_2^2 - 19p_2 - 3, \end{aligned} \tag{S45b}$$

which can be represented under the following decoupled structure:

$$\begin{bmatrix} q_1 \\ q_2 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ -3 & -1 \end{bmatrix} \begin{bmatrix} 2x_1^2 - 3x_1 + 1 \\ x_2^3 - x_2 \end{bmatrix}, \quad \text{with} \quad \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -2 & -2 \\ 3 & -1 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \end{bmatrix},$$

revealing the internal univariate polynomials and the linear transformations at the input and output of the structure.

Uniqueness

It is shown that the decoupled models are not always unique [63]. Hence, decoupling will not lead to the physical underlying representation, but it will still nevertheless provide increased intuitive insight into the behavior of the system.

Exact and Approximate Decomposition

In an exact decoupling of an arbitrary multivariate function, the number of branches r might become very large. A truncated decoupling with a reduced number of branches can be used to approximate the multivariate function. This problem is studied in [66]. The approximation error is tuned using a weighted least-squares criterion. The number of branches r can be used as a handle to balance the complexity of the model against the level of the tolerated structural model errors. This is illustrated on the forced Duffing oscillator in Figure S15

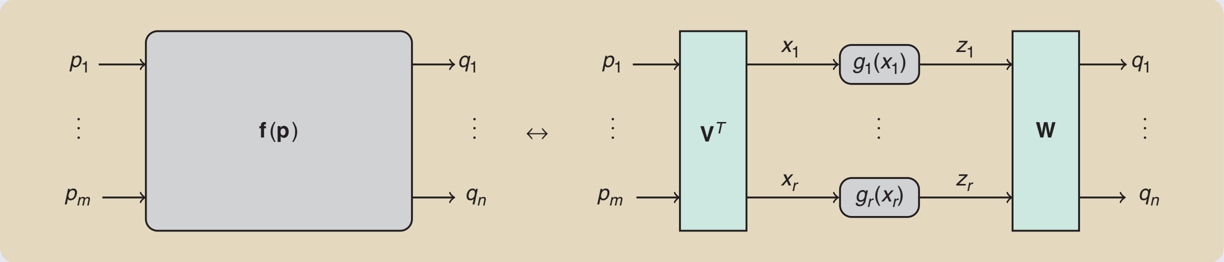


FIGURE S14 The multivariate nonlinear function $q = f(p)$ is replaced by a decoupled representation [63].

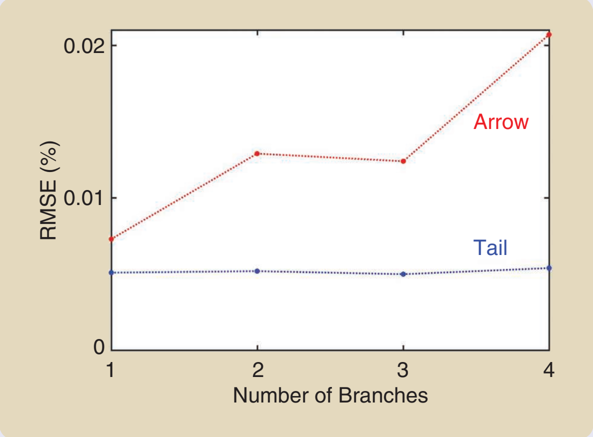


FIGURE S15 The evolution of the root-mean-square error (RMSE) on the tail and arrow data as a function of the number of branches.

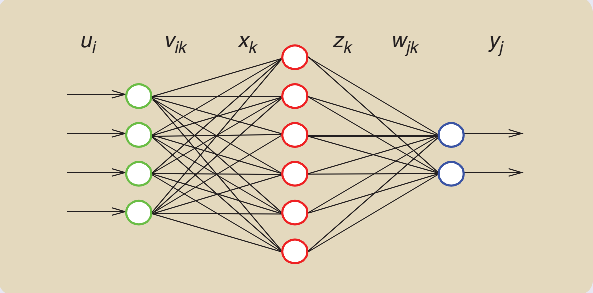


FIGURE S17 A neural network model. The inputs u_i (green) are linearly combined with the weights v_{ik} to generate the inputs x_k for the neurons (red). The output z_k of each neuron is calculated by a univariate nonlinear function on its input x_k, and it is linearly combined with the weights w_{kj} to generate the outputs y_j (blue). This is exactly the same structure as was used in the decoupling approach. The major difference between the neural network and the decoupling method is that, in the latter, the nonlinear functions vary from one node to the other, while in the neural network, they are all equal. This additional flexibility results in a faster convergence, so that less branches are needed.

and the Bouc–Wen model in Figure S16, which are reported in detail in [65].

Relation With Neural Networks

The structure of the decoupled representation in Figure S14 is very similar to that of a neural network, as shown in Figure S17. In a neural network, the univariate functions belong to the same family (for example, sigmoids or hinge functions). It is known that neural nets are universal approximators [10], [11], [13] converging as $O(1/\sqrt{r})$, with r the number of neurons in the hidden layer. In the decoupling method, the univariate functions follow from the decoupling and are tuned to the specific problem. This additional degree of freedom will reduce the number of branches needed in the decoupled representation of the multivariate function.

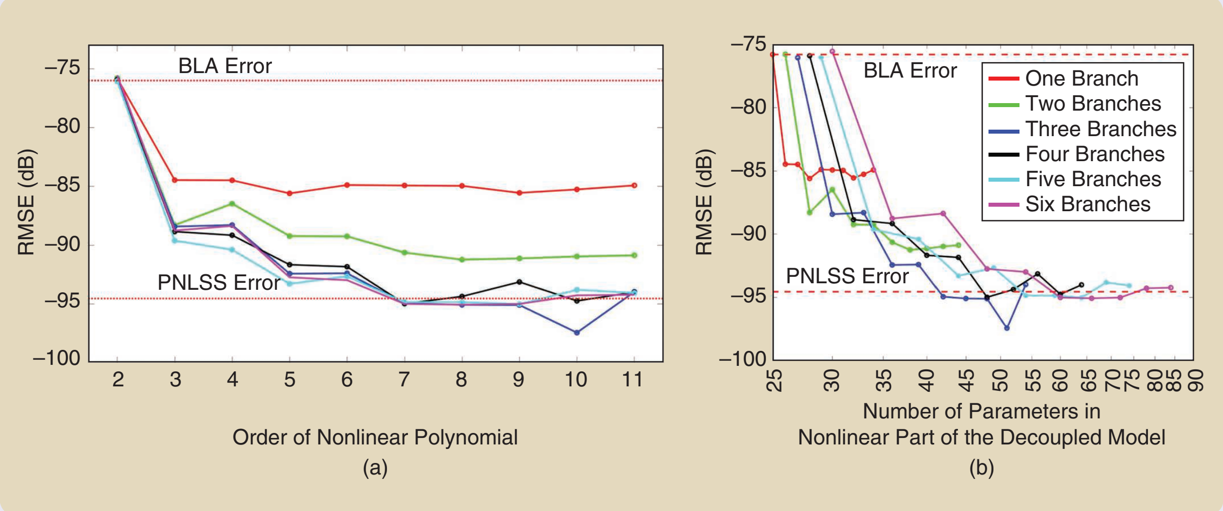


FIGURE S16 An illustration of the approximate decoupling on the Bouc–Wen hysteresis problem [65]. The multivariate function of the full polynomial nonlinear state-space (PNLSS) model (three states, multivariate polynomial of degree seven on the states and the input, 364 parameters) is replaced by a decoupled representation with an increasing number of branches (one to six). Next, the degree of the univariate polynomials is varied between three and 11. The decoupled model is tuned to the data using a least-squares cost function. In the figure, the root-mean-square error (RMSE) on a validation set is plotted for a varying number of branches, ranging from one to six. (a) The error is plotted as a function of the degree of the univariate polynomial. (b) The error is plotted as a function of the number of parameters, which depends on the degree of the univariate polynomials. The best results are obtained for a model with four branches and a polynomial of degree 11. The MSE of the best linear approximation (BLA) is –76 dB, and the error of the best nonlinear models is 10 times smaller. The RMSE of the coupled PNLSS model is indicated with a*.

over the periodic repetitions of the output. A noise floor more than 40 dB below the signals of interest shows that the quality of the data is very high. The distance between the nonlinear distortion levels and the noise floor is a measure of the potential improvement that can be obtained by using a nonlinear model. In the case of Figure 4, a gain of a factor of 100 is possible for the high excitation levels if a good nonlinear model can be obtained.

User Guideline

Make a nonparametric distortion analysis whenever it is possible to apply periodic excitations. Use this information to decide if a linear or nonlinear approach is needed and check how much can be gained by turning toward nonlinear system identification.

Linear Modeling in the Presence of Nonlinear Distortions

Deciding to use the linear identification approach also implies the presence of structural model errors in the results. This may be an acceptable solution, but the user should fully understand the impact of the structural model errors on the validity of the results. This is discussed in “Impact of Structural Model Errors.” The major conclusions are that 1) the experiment should be tuned to the application (the same class of excitation signals) and 2) no reliable theoretical uncertainty bounds can be provided. These should be obtained from repeated experiments with different excitation signals generated from the relevant class of input signals [81]. The problem of linear modeling in the presence of nonlinear distortions is discussed in “Linear Models of Nonlinear Systems” and [30], [72], and [82]–[95]. The classical linear identification methods will lead to consistent estimates of the best linear approximation (BLA).

For random excitations, the nonlinear effects at the output will look very similar to noise, and it is very difficult for an inexperienced user to recognize their presence. Their power spectrum can be estimated using a parametric noise model simultaneously identified with the plant model (as in utilizing a Box–Jenkins model [6]). The noise model can be employed in a control design to make the disturbance analysis. However, uncertainty bounds calculated from it are no longer valid because the errors are not independent of the input. Again, there is no theoretical framework available today to solve that problem. The variance of the estimates should be obtained by repeating the experiment multiple times with a random-varying excitation, as explained in “Impact of Structural Model Errors.”

User Guideline

In summary, linear models can be very useful, even in the presence of strong nonlinearities, because they are much easier to address. Make sure to understand the impact of the nonlinear distortions on the BLA.

Nonlinear System Identification

A nonlinear system identification approach is justified if the preprocessing step indicates nonlinear distortion levels that are too high, well above the noise floor of the data. It is this problem that will be further addressed in this article, in terms of what identification methods to use, how to select a model class, and how to design the experiments. Again, the user will have many options, starting from simple nonlinear models that are good enough to solve the problem to complex models that include the fine details deeply hidden in the data.

User Guideline

Select nonlinear system identification only if there is enough evidence that linear models will not solve the problem.

THE PALETTE OF NONLINEAR MODELS

The Multitude of Nonlinear Models

A major challenge in dealing with nonlinear system identification is that a great many nonlinear model structures have been suggested. A user can be confused when becoming familiar with the different choices available and how to choose a structure that suits a particular situation. A clear perspective on this wide variety of choices is needed to make a well-informed choice. Nonlinear model structures can be ranked along two axes that are directed, respectively, by the user’s preference and system behavior.

1) *User’s preference*: A first classification of the models is made in terms of how much prior knowledge about the system is used [96]. Based on the familiar concept of black-box models for general flexible structures with no physical insights, the user can then select from a palette of models of different shades of gray to delineate many approaches to common nonlinear models.
2) *System behavior*: An alternative to using varying degrees of structural physical insights is to use behavioral aspects of the system. Does it, for example, show behaviors like chaos, shifting resonance frequencies and varying damping, or hysteresis, as discussed by Pearson [97]? Then, the main decision is whether to include the nonlinearity in a feedback loop. This selection is not a free user choice; it is imposed by the system behavior. A detailed ranking along this line is discussed in “Static Nonlinearities” and “External or Internal Nonlinear Dynamics.”

The user should combine both aspects in the final selection of the model. Note that there are many other dimensions in which the different approaches can be classified, namely, universality, computational effort, and suitability to address unstable systems. These aspects are not further elaborated in this article.

Extensive Case Study: The Forced Duffing Oscillator

Throughout this article, many results are illustrated on the forced Duffing oscillator, sometimes called the Silverbox in nonlinear benchmark studies. This sidebar describes the setup and experiments used throughout this article. A detailed description of the experiments is given in [79].

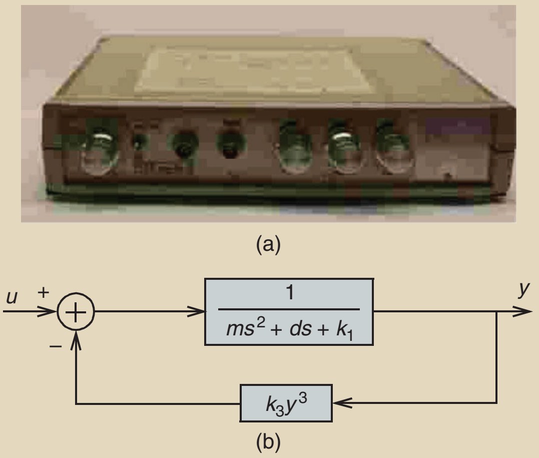
SYSTEM, EXPERIMENTAL SETUP, AND EXPERIMENTS

The system is an electronic circuit that mimics a nonlinear mechanical system with a cubic hardening spring, as shown in Figure S18. This class of nonlinear systems has a rich behavior, including regular and chaotic motions and the generation of subharmonics [8], [27].

The experimental setup consists of a generator and two data acquisition cards that are synchronized to avoid leakage errors in the spectral analysis. The generator starts from the zero-order-hold (ZOH) reconstruction [106] u_{ZOH} of a discrete-time sequence $u_d(k)$ passed through a low-pass generator filter G_{gen} to eliminate the higher harmonics of the ZOH reconstruction $u(t) = G_{\mathrm{gen}} u_{\mathrm{ZOH}}$. The sampling frequency is $f_s = 10\,\mathrm{MHz}/2^{14} \approx 610\,\mathrm{Hz}$. The data acquisition cards are alias protected (the signals are passed through a low-pass filter before sampling) and sample the input and output at a rate f_s. High-impedance buffers are used to eliminate the interaction between the plant and the measurement setup.

The experiments are shown in Figure S19 and consist of three parts using a tail, arrow, and sweeping sine input $u(t)$ [Figure S19(a), (b), and (c), respectively]. The output [Figure S19(d)–(f)] of the circuit corresponds to the displacement $y(t)$. The following observations can be made:

- The tail [Figure S19(a)] and swept sine excitation [Figure S19(c)] have approximately the same root-mean-square (RMS) value. The arrow signal [Figure S19(b)], with a maximum amplitude that is twice that of the other signals, will be used to verify the extrapolation capabilities of the models that are identified on the other excitations. The swept sine alike excitation is a Schroeder multisine [30] that has a dominant odd behavior because the amplitude of the excited odd frequencies is approximately 30 dB above the level of the excited even frequencies [see Figure S19(i) and (l)].
- All input signals have the same bandwidth [see Figure S19(g)–(i)].
- The output [Figure S19(f)] for the sweeping sine [Figure S19(c)] becomes very large around the resonance frequency, even if the input level remains constant. This results in an internal extrapolation for models that are identified on the tail [Figure S19(a)].
- In the input spectrum [Figure S19(g)], it can be seen that there are spurious components at the odd multiples of the mains frequency (50 Hz). The signal is picked up by the circuit and acts as process noise.
- The nonparametric distortion analysis in Figure 4 shows that the signal-to-noise ratio (SNR) of these measurements is 40–60 dB (with the measurement and process noise at the 1–0.1% level).


(a)

(b)

FIGURE S18 (a) A forced Duffing oscillator [79] is a system in which the electronic circuit mimics a nonlinear mechanical system with a hardening spring. The system is excited with an input $u(t)$ (the applied force to the mechanical system). The output of the system corresponds to the displacement $y(t)$. (b) The schematic representation of the system as a second-order system with a nonlinear feedback.

LINEAR SIMULATION AND PREDICTION OF THE FORCED DUFFING OSCILLATOR

The behavior of simulation and prediction errors (see "Simulation Errors and Prediction Errors") is illustrated on the forced Duffing oscillator. Because the SNR of these measurements is very high (noise disturbances well below 1%), the simulation and prediction errors in the subsequent study are completely dominated by structural model errors. The simulation and prediction errors will be shown for linear Box–Jenkins (BJ) models, autoregressive exogenous (ARX) models [6], [7], and a nonlinear ARX (NARX) model.

Linear Simulation and Prediction of the Duffing Oscillator

In the first step, a linear model

$$y(t) = y_0(t) + v(t) = G(q,\theta)u(t) + H(q,\theta)e(t), \tag{S46}$$

with

$$\begin{aligned} G(q,\theta) &= \frac{B(q,\theta)}{A(q,\theta)} = \frac{b_0 + b_1 q^{-1} + \cdots + b_{n_b} q^{-n_b}}{1 + a_1 q^{-1} + \cdots + a_{n_a} q^{-n_a}}, \\ H(q,\theta) &= \frac{C(q,\theta)}{D(q,\theta)} = \frac{1 + c_1 q^{-1} + \cdots + c_{n_c} q^{-n_c}}{1 + d_1 q^{-1} + \cdots + d_{n_d} q^{-n_d}}, \end{aligned} \tag{S47}$$

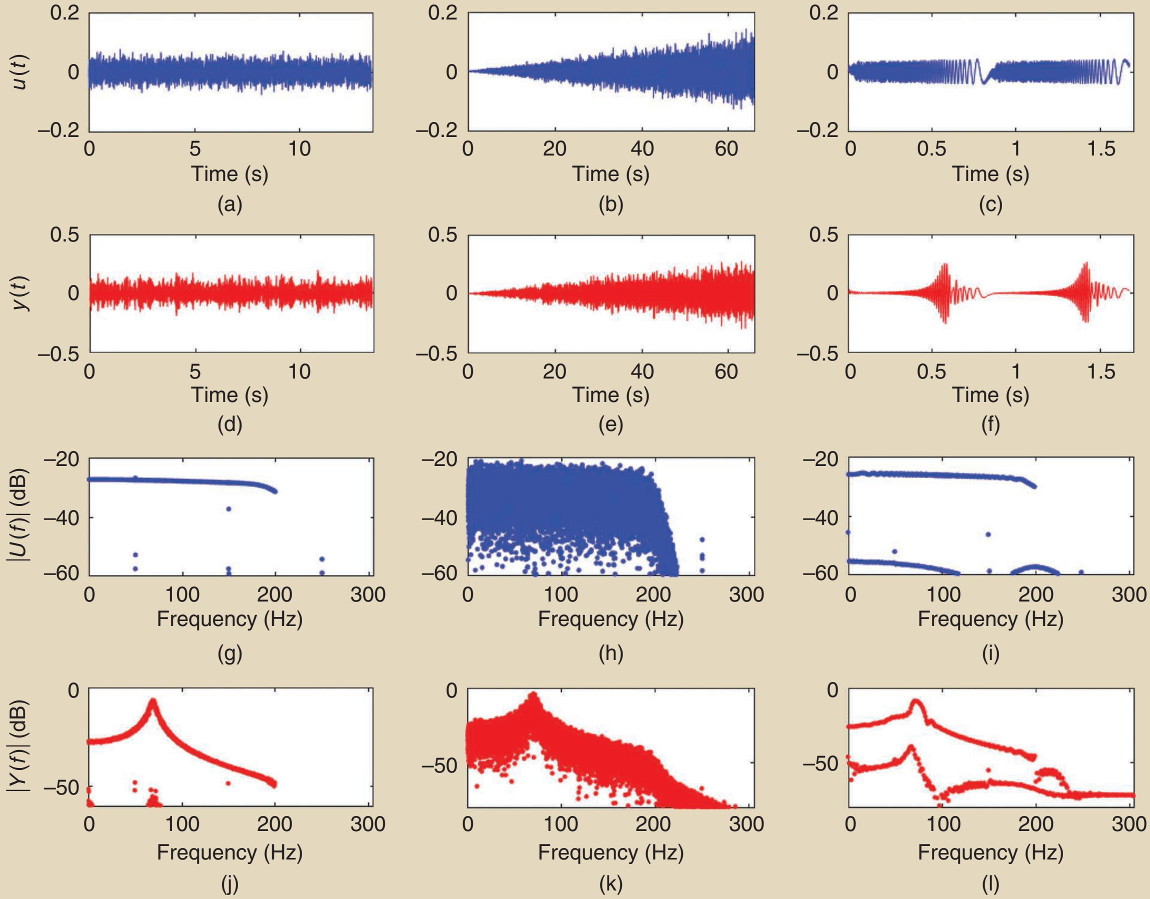


FIGURE S19 Results from the Duffing oscillator for (a), (d), (g), and (j): a random-phase multisine (26) experiment; (b), (e), (h), and (k): a growing noise experiment; and (c), (f), (i), and (l): a sweeping sine alike experiment. The input signals are shown in blue, and the output signals in red. The two top lines show the time-domain signals, and the two lines at the bottom show the amplitude spectra.

is estimated on the tail experiment using the prediction error framework [6], [7]. The order of the BJ model is $n_a = n_b = 2$ and $n_c = n_d = 6$, and the plant and noise models G and H are independently parameterized. For the ARX model, the order of the plant model is also $n_a = n_b = 2$, while in this case, the noise model is $C = 1, D(q,\theta) = A(q,\theta)$. The latter fits with the assumption that disturbances originate mainly at the input of the system, so that it shares its dominant dynamics with the input signal. The BJ and ARX simulation errors are shown in Figure S20 and compared to the 95% amplitude levels calculated from the estimated noise models. The BJ noise model describes the disturbances very well, while the ARX model underestimates the errors around the resonance frequency.

These results are used to simulate $\hat{y}_0(t) = G(q,\theta)u(t)$ [see Figure S21(g)–(i)] and predict $\hat{y}(t) = H^{-1}(q,\theta)G(q,\theta)u(t) + (1 - H^{-1}(q,\theta))y(t)$ [see Figure S21(j)–(l)], the output for the three experiments [6], [7]. The simulation errors of both the BJ and ARX models are on top of each other in these figures. However, the better BJ noise model results in a significantly smaller prediction error for the BJ models, compared to that of the ARX models. By increasing the model orders n_a, n_b for the ARX model, these results could be improved. However, to be able to compare the BJ and ARX models of the same complexity, the discussion is continued with the overly simple ARX model.

User Guideline

In summary, linear prediction can provide good results in the presence of nonlinear distortions. This is much harder for linear simulation. The quality of the noise model has a strong impact on the quality of the predictions. A changing nature of the excitation signal results in a changing behavior of the residuals due to structural model errors. This can turn a good prediction model into a poor one.

NONLINEAR SIMULATION AND PREDICTION OF THE FORCED DUFFING OSCILLATOR USING A POLYNOMIAL NONLINEAR AUTOREGRESSIVE EXOGENOUS MODEL AND A POLYNOMIAL NONLINEAR STATE-SPACE MODEL

A polynomial NARX (22) and a polynomial nonlinear state-space model (20) are estimated on one of the realizations of the random-phase multisines in the tail.

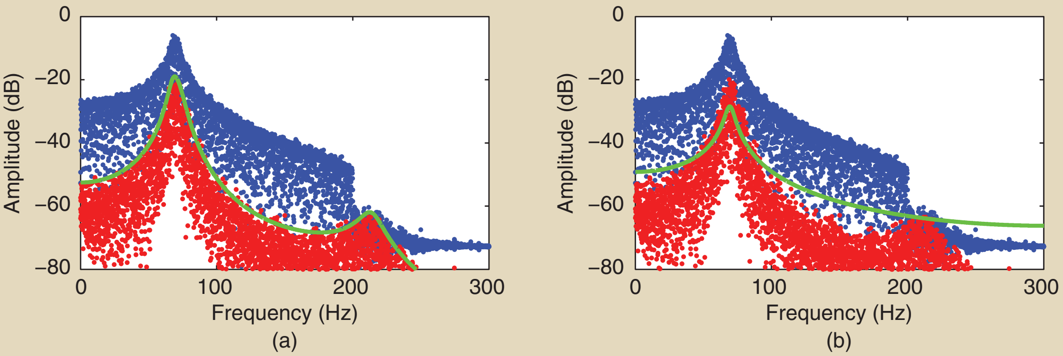


FIGURE S20 The linear modeling of the forced Duffing oscillator [79]. (a) The output of the forced Duffing oscillator is simulated using an estimated Box–Jenkins (BJ) model (the plant model has two poles and two zeros, and the noise model has six poles and six zeros) or (b) an autoregressive exogenous (ARX) model (a plant model order with two poles and two zeros). The amplitude of the discrete Fourier transform of the measured output and the simulation error are shown. The blue dots are the measured output, and the red dots are the simulation error. The green line is the 95% error level that is calculated from the estimated noise model. Observe that the simulation error for both models is very similar. The BJ noise model describes the disturbances very well, while the ARX model underestimates the errors around the resonance frequency.

Polynomial Nonlinear Autoregressive Exogenous Model and Polynomial Nonlinear State-Space Model

The polynomial NARX model is of order $n_a = 2, n_b = 2$ and polynomial expansion of degree three, with arguments $R = [u(t), u(t-1), u(t-2), y(t-1), y(t-2))]^T$. The selection of the nonlinear degree and arguments follow the results in [26] that were obtained on a trial-and-error basis

$$\begin{aligned} y_0(t) &= h(u(t), \ldots, u(t-n), y(t-1), \ldots, y(t-n_a)) \\ &= \sum_{\text{all combinations}} c_{p,m-p}(k_1, k_2), \ldots, k_m) \prod_{i=1}^{p} y(t-k_i) \prod_{i=p+1}^{p} u(t-k_i). \end{aligned} \tag{S48}$$

It might also be possible to obtain better results by replacing the polynomials with other basis functions. It is important to observe that this model is linear in the parameters $c_{p,m-p}$, which will reduce the identification to a problem linear in the parameters for a quadratic cost function.

The polynomial nonlinear state space is of order two (two states) and polynomial degree three, with arguments $R = [x_1(t),\ x_2(t),\ u(t)]^T$

$$\begin{aligned} \begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} &= A \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + Bu(k) + \begin{bmatrix} \mathbf{f_1(x_1(k), x_2(k), u(k))} \\ \mathbf{f_2(x_1(k), x_2(k), u(k))} \end{bmatrix}, \\ y(k) &= C \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + Du(k). \end{aligned} \tag{S49}$$

Only the state transition equation has a nonlinear term, and adding a nonlinear term to the output equation did not improve the results.

Cost Function

Polynomial NARX models are linear in the parameters, and, hence, the cost function (given by the squared equation errors) is minimized by solving a linear set of equations so that no initialization problem must be solved. This is the major advantage of polynomial NARX models compared with many of the other models presented in this article. Because the noise enters nonlinearly into the model, a bias will appear. However, as long as the SNR is decent (for instance, better than 20 dB), this will not be the major issue. For that reason, polynomial NARX models became one of the most popular methods, and they are widely applied in many different fields.

For the nonlinear state-space model, the cost function is nonlinear in the parameters (nonconvex), and a numerical method is used to minimize the cost function:

$$\begin{aligned} e(t) &= y(t) - h(u(t), \ldots, u(t-n), y(t-1), \ldots, y(t-n_a)), \\ V &= \frac{1}{N}\sum_{t=1}^{N} e(t)^2, \end{aligned} \tag{S50}$$

where $u(t)$ and $y(t)$ are the measured values.

Results

The results for the polynomial NARX and polynomial nonlinear state-space (PNLSS) models are shown and discussed in Figure S22. For smaller output levels, the two models are quite comparable; but for larger outputs, the errors of the polynomial NARX model are two to three times larger than those of the PNLSS model.

Deep-Learning Nonlinear Autoregressive Exogenous Model

A deep-learning neural network with two layers and 27 nodes in each layer is identified on a section of the tail data (see "Machine Learning and System Identification"). The NARX

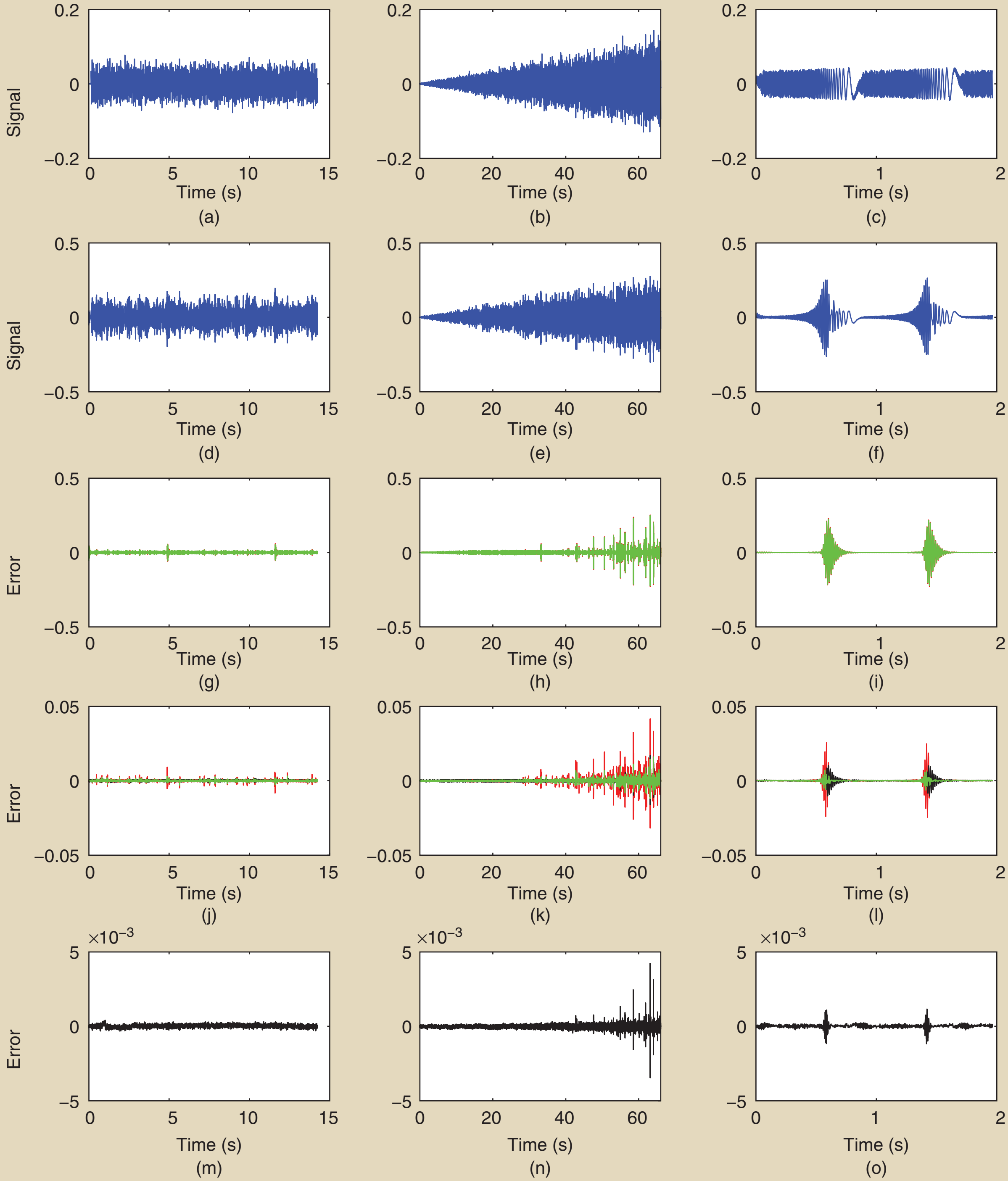


FIGURE S21 A study with the Duffing oscillator for the (a) and (d) tail experiment; (b) and (e) arrow experiment; and (c) and (f) sweeping sine experiment. (g)–(i) The Box–Jenkins (BJ) (green) and autoregressive exogenous (ARX) (red) simulation errors are almost equal to each other. (j)–(l) However, the prediction errors for the BJ model (green) are well below those of the ARX model (red), especially for larger amplitudes of the output. This is mainly due to the better BJ noise model, as shown in Figure S20. The simulations for the nonlinear ARX (NARX) model (black) are also shown. (m)–(o) The polynomial NARX simulation (black) has the same quality as the linear predictions in (j) and (k) (lines on top of each other), while it is in between the ARX and BJ linear predictions in (l). The polynomial NARX prediction errors are a factor of 10 smaller. The simulation error is shown in Figure S22.

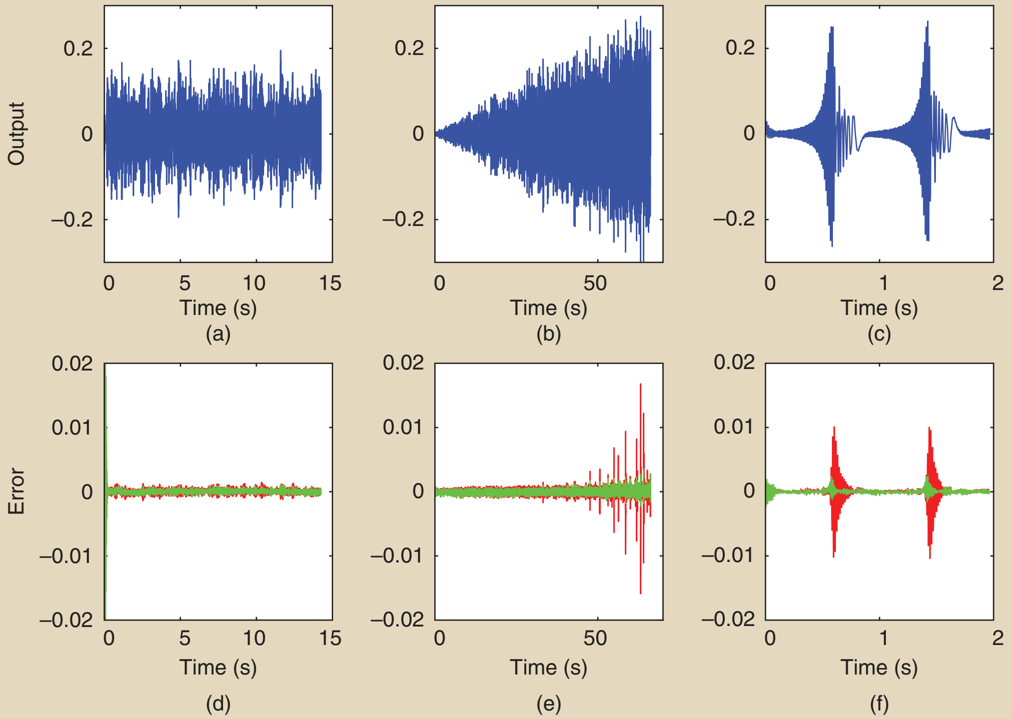


FIGURE S22 Modeling the forced Duffing oscillator for the (a) tail, (b) arrow, and (c) sweeping sine experiments using a discrete-time polynomial nonlinear autoregressive exogenous (NARX) model and polynomial nonlinear state-space model. Both are identified on a section of the tail data. The simulation error is shown for the polynomial NARX (red) and the polynomial nonlinear state space (green) on validation data in the (d) tail, (e) arrow, and (f) sweep. For smaller output levels, both models are quite comparable. However, for larger outputs, the errors of the polynomial NARX model are two to three times larger. The large initial green spike in (d) is due to transient effects because the initial states were not estimated.

model has four delayed inputs and outputs. The model is validated on the sweep data, and the results are shown in Figure S23. The simulation errors for the PNLSS and the deep-learning-based NARX models perform equally well on this test, illustrating the high potential of using deep-learning methods within the nonlinear system identification framework.

User Guideline

In summary, NARX models with an expansion in known basis functions (like the polynomial NARX model) are highly attractive because these are quite simple to use: no initialization and no problems with local minima. The PNLSS model results in slightly better results because it was better tuned. The nonlinear prediction models perform significantly better than the linear ones, even if they are not optimally tuned. The polynomial regression functions can be replaced with more convenient alternatives from machine learning (Gaussian bells and hinge functions), as discussed in “Static Nonlinearities.” The polynomial NARX method can be combined with pruning and structure-revealing methods to simplify the models.

PRUNING, DATA-DRIVEN STRUCTURE RETRIEVAL, AND MODEL REDUCTION

Polynomial Nonlinear Autoregressive Exogenous Model Pruning

The major drawback of polynomial NARX models is the combinatorial growth of the number of parameters with the number of regressors (in this example, five) and the degree (in this example, three). No further pruning of the model was made in the results presented. It is well known that pruning can improve the model quality, especially if the number of regressors grows [19]. In [19], different strategies are presented to gradually include the dominating terms for the growing complexity of the polynomial NARX model.

Recently, a top-down approach was proposed [26] using the decoupling strategy presented in the section “Decoupling of Multivariate Polynomials.” First, a full model (including all regressor combinations) is identified, resulting in a single-output multivariate polynomial that is next decoupled as a sum of univariate polynomials P_k that act on linear combinations of the regressors. In [26], the polynomial NARX model for the forced Duffing oscillator is decoupled with four polynomials of degree three: $y(t) = \Sigma_{k=1}^{4} P_k(v_k^T R)$. This method offers a systematic approach to model pruning, especially for a large number of regressors.

Nonlinear State-Space Model Reduction and Model Data-Driven Structure Retrieval

Decoupling

Similar to the polynomial NARX model, it is also possible to decouple the multivariate nonlinear vector function $[f_1, f_2]^t$ in (S49) using the decoupling strategy presented in "Decoupling of Multivariate Polynomials." Four internal single-input, single-output (SISO) branches were needed in the decoupled representation (see Figure S14), which are shown in Figure S24. In a second step, the degree of the polynomials was increased from three to five, and the cost function was minimized for this decoupled model. This reduced the RMS error (RMSE) of the decoupled model to 0.4% (or fit = 99.6), while the original full third-degree model had an RMSE of 0.49% (or fit 99.51) (see Table S1).

Model Reduction

In the following step, the number of branches of the decoupled model was reduced. In each reduction step, a new cost function minimization was performed. The RMSEs of the reduced models on the tail and arrow validation data are shown in Figure S15. Although it could be expected that the errors would start to grow when the model is simplified, they remain constant on the tail data and drop significantly on the arrow data. This indicates that the true system can be described with only one nonlinear branch in the decoupled model (see Figure S14). The reduced errors on the tail data indicate a better generalization (extrapolation) capability of the reduced model. The models are estimated on the tail data that cover a smaller domain than the arrow and sweep data, as is shown in Figure 8. The evolution of the RMSE on the tail validation data, number of linear parameters n_{θ_L}, and the number of nonlinear parameters $n_{\theta_{NL}}$ as a function of the model complexity during the model reduction process is given in Table S1.

Data-Driven Structure Retrieval

The final nonlinear state-space model is

$$\begin{aligned}
\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} &= A\begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + Bu(k) + \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} g(p), \\
y(k) &= C\begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + Du(k), \\
p &= \theta_3 x_1(k) + \theta_4 x_2(k) + \theta_5 u(k), \qquad \text{(S51)}
\end{aligned}$$

where $g(p)$ is a SISO polynomial of degree five, with argument p that is a linear combination of the states x_1, x_2 and the input u (with parameters $\theta_3, \theta_4, \theta_5$). The coefficients θ_1, θ_2 scale the polynomial output $g(p)$. The validation results on the arrow data are given in Figure S25. Observe that the full and the final reduced model have almost the same quality.

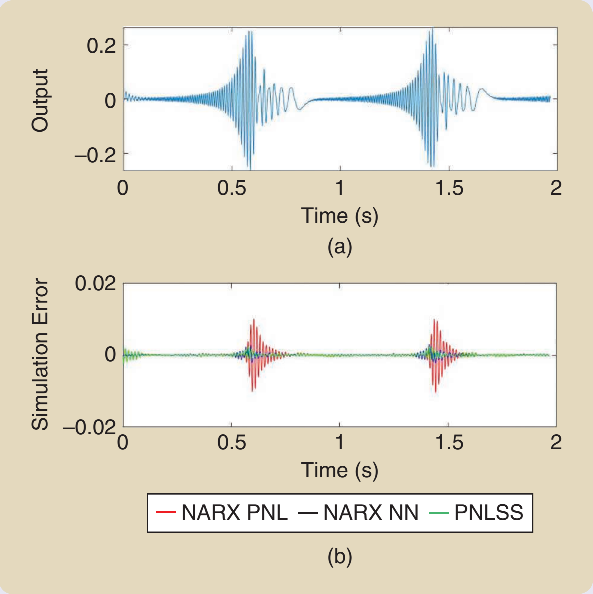


FIGURE S23 Modeling the forced Duffing oscillator using deep learning. A discrete-time nonlinear autoregressive exogenous (NARX) model is identified using a two-layer neural network with 27 hidden nodes in each layer. The model is identified on a section of the tail data and validated on the sweeping sine data. (a) The measured output. (b) The simulation for the polynomial NARX (red), the polynomial nonlinear state-space (PNLSS) model (green), and the deep learning-based NARX model (NARX NN, deep blue). It can be seen that the deep-learning NARX model and the PNLSS model perform equally well.

Because the forced Duffing oscillator is a lab setup, this model can be compared to the physical model. In this case, the data-driven retrieved model structure and the physical model structure are identical. This is a highly motivating result for the black-box identification framework. However, this conclusion cannot be generalized because it is not obvious that the simpler model coincides with the physical model. Moreover, the structure of many nonlinear systems is unidentifiable from input–output data because of indistinguishability problems; the same data can be represented by multiple models that cannot be distinguished from each other without additional information (as with the measurement of an internal signal). In the case of the forced Duffing oscillator, it is also possible to represent the same input–output behavior by a system with a nonlinearity in the forward path instead of the feedback path [127].

An alternative approach to obtain an extremely structured model for the forced Duffing oscillator in a black-box modeling framework is presented in [25]. It is based on a data-based mechanistic modeling approach, as explained in [21], where the objective is to obtain a model that can be interpreted in the mechanistic terms most appropriate to the nature of the dynamic system. The nonlinear nature

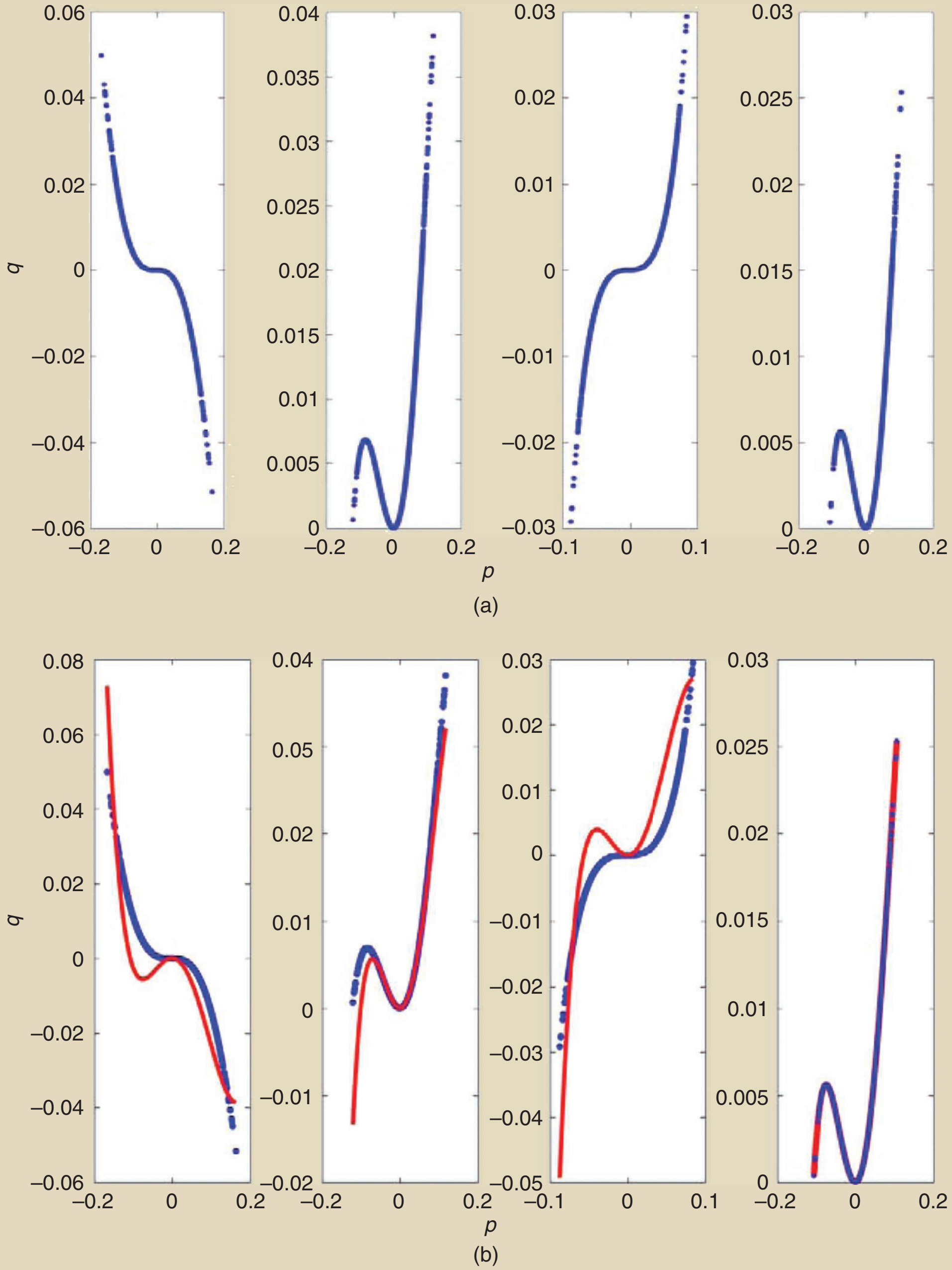


FIGURE S24 (a) The multivariate nonlinear polynomial vector function [the bold term in (S49)] is decoupled using four simple-input, simple-output branches. (b) These functions are approximated by a single function after proper scaling of the input p and the output q in each of the branches.

of the system is captured using state-dependent parameters. The basic idea is to start with a linear model and check for data-dependent variations of some of the parameters in the model. For the Duffing oscillator, this resulted in a transfer function model that is very similar to the final model (S51).

General Structure of Nonlinear Models

In general, the measured system input and output at time t will be denoted by $u(t)$ and $y(t)$. All measured data up to time t will be denoted by

$$Z^t = \{u(s), y(s)\} \quad \text{for all} \quad s \leq t. \tag{7}$$

The models can be expressed in discrete or continuous time. Most real-life systems evolve in continuous time, but discrete-time models are often preferred to simplify the numerical simulations. Linear systems can be perfectly represented by discrete-time models for zero-order-hold excitations [4], [6], [7]. However, this approach cannot

TABLE S1 The evolution of the root-mean-square error (RMSE) on the tail validation data, the number of linear parameters n_{θ_L}, and the number of nonlinear parameters $n_{\theta\text{NL}}$ as a function of the model complexity during the model reduction process.

	Best Linear Approximation	Nonlinear State-Space Model	Decoupled	Equal Branches	Single Branch
RMSE	12%	0.49%	0.4%	0.4%	0.4%
n_{θ_L}	5	5	5	5	5
$n_{\theta\text{NL}}$	0	30	12	6	3

User Guideline

In summary, the decoupling method offers a data-driven systematic approach to model pruning. Tuning the number of branches offers a data-driven model reduction approach. The simplified models can significantly improve the intuitive insight in the nonlinear model behavior. The underlying physical model structure can sometimes be retrieved, although no guarantee can be given that a simple model coincides with the physical model.

CONCLUSIONS OF THE CASE STUDY ON THE FORCED DUFFING OSCILLATOR

In this case study, structural model errors dominate the noise disturbances. The following general observations can be made. The prediction errors for a given plant model are smaller than the simulation errors. This was expected because prediction methods explicitly use the output measurements to reduce the simulation error. The prediction method decreases the impact of the structural model errors, compared with the simulation method. This holds true as well for the linear simulation/prediction models and the nonlinear models. The quality of the prediction depends strongly on the quality of the noise model. The better BJ noise model also results in better predictions than those of the polynomial NARX model. A nonlinear model better captures the system behavior than a linear model. This results in smaller simulation and prediction errors. The decoupling methods are a powerful tool for pruning, data-driven structure retrieval, and model reduction.

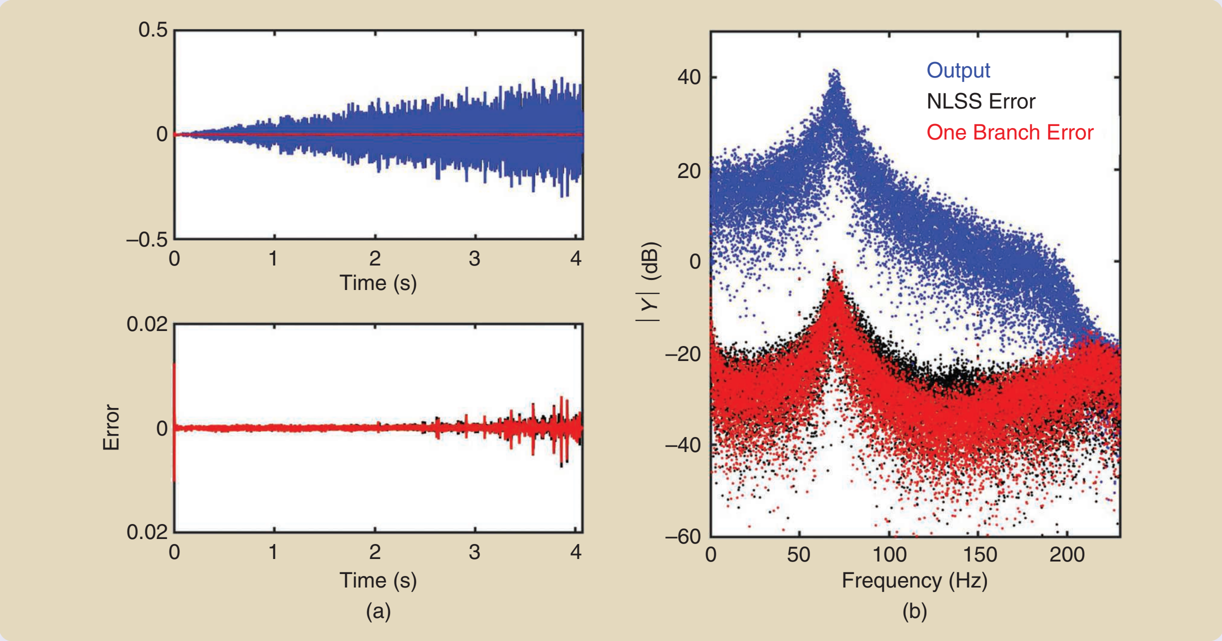


FIGURE S25 The validation of the final model (S51) on the arrow data in the (a) time and (b) frequency domains. The measured output is shown in blue, the validation of the final model in red, and the full model in black. NLSS: nonlinear state space.

be generalized to nonlinear systems because the ZOH nature of the signals is lost inside the nonlinear system. Continuous-time modeling of nonlinear systems has been intensively studied in the literature [19], [98]–[105]. In "Approximating a Continuous-Time Nonlinear State-Space Model With a Discrete-Time Model," the discrete-time approximation is studied within the band-limited setup [106], [107]. The discretization error can be made arbitrarily small by increasing the sampling frequency and/or the model complexity, as shown in that sidebar. Both models will be considered in this section. In the discrete-time case, the time variable will be enumerated by time instances $t = 1, 2, 3, \ldots$ (a constant sampling interval of one time unit).

The general structure of all of the nonlinear models can be in the form

$$q(t) = F(p(t)), \tag{8}$$

with p and q vectors that are built on the signals turning around in the model (namely, inputs, outputs, states, and internal variables). The function F is a static nonlinearity. The properties and parameterization of static nonlinearities are discussed in more detail in "Static Nonlinearities" and "External or Internal Nonlinear Dynamics," which also illustrate how it can be used in various nonlinear dynamic models.

The model is an expression that allows the computation of the next output $y(t)$ based on previous observations:

$$\hat{y}(t \mid \theta, Z^{t-}). \tag{9}$$

This model output will depend on a parameter vector θ used to parameterize the model class. The notation Z^{t-} denotes that $y(t)$ is excluded. In discrete time, this means Z^{t-1}. Note that, if a direct term is needed in the model, the regressors Z^{t-1} can be extended with $u(t)$.

Machine Learning and System Identification

Machine learning has become a central topic and buzzword in today's decision and estimation world. It clearly has links to nonlinear system identification. This article provides a simple presentation of the leading ideas in machine learning and how they relate to the topic of the current survey.

THE BASIC PROBLEM IN MACHINE LEARNING

The core of all machine learning can be said to be *function estimation*: measure noisy observations $y(t_i), x(t_i)$ of a function between two spaces $\mathcal{X}$ to $\mathcal{Y}$

$$y(t) = F(x(t)) + \text{noise}, \tag{S52}$$

and infer the function F. This is like a classical curve-fitting problem. However, the spaces may be of a highly general nature: $\mathcal{X}$ could be signals, images, or texts, and $\mathcal{Y}$ could be numerical values or classifications (like "text is offensive"). It is the broad range of possible spaces that allows the problem formulation to be applied to a wide variety of tasks, which is the reason for the current interest in the field. For standard numerical spaces $\mathcal{X}$ and $\mathcal{Y}$, the formulation (S52) remains an example of standard *nonlinear regression*.

Most applications of machine learning have a *black-box* view of F, that is, no physical knowledge about the function is employed but only flexible mappings are used when estimating it. This is similar to the situation in "Static Nonlinearities."

With a totally free description of F, it is clear that some constraining technique must be applied when fitting it to the observed $x(t_i), y(t_i)$ to avoid a perfect (and useless) fit. In machine learning, two basic techniques are used. The first is regularization, which means that a penalty is added to the model fit to account for deviations in F from expected smoothness properties. This can be compared with (S43). This can also be interpreted as the Gaussian processes approach, described in connection with Figure S26. The second is to use a flexible parametric model class for F, which is aligned with the general philosophy used in this article (see the section "The Lead Actors in Nonlinear System Identification"). The two approaches will be discussed in more detail in the following.

APPROACH 1: REGULARIZATION, GAUSSIAN PROCESSES, AND KERNEL METHODS

The basic approach for estimating the function F in (S52) is

$$\min_{F \in \mathcal{F}} \sum_{i=1}^{N} (y(t_i) - F((x(t_i))^2. \tag{S53}$$

However, this formulation is ill posed for a general function class $\mathcal{F}$ since the fit can always be perfect. This allows for regularization techniques [137], which add a regularizer J to the criterion:

$$\min_{F \in \mathcal{F}} \left(\sum_{i=1}^{N} (y(t_i) - F((x(t_i))^2 + \gamma J(F)) \right). \tag{S54}$$

There are many ways to define the regularizer. A common and elegant way is to assume that $\mathcal{F}$ is a Hilbert space with norm $\| F \|_{\mathcal{F}}^2$ and use

$$\min_{F \in \mathcal{F}} \left(\sum_{i=1}^{N} (y(t_i) - F((x(t_i))^2 + \gamma \| F \|_{\mathcal{F}}^2 \right). \tag{S55}$$

These techniques were developed in 1970–1990 for general function estimation, giving rise to spline approximations [138], [139]. The penalty norm became known as the *kernel* for the problem—hence the term *kernel techniques* for the approach, see [140].

A way to define the kernel/regularization is the Gaussian process approach. Suppose that F is the realization of a Gaussian process, and assume that its prior distribution is that it is zero mean, with covariance function $R(t, s) = \mathbf{E}F(t)F(s)$. Using this as the scalar product in the Hilbert space $\mathcal{F}$, the estimate in (S55) will be the posterior probability density function of the random process F, given the prior and the observations $y(t_i),\ x(t_i),\ i = 1, \ldots, N$. This was used to create a general theory for estimating linear systems, described in [58]. F is the impulse response of the linear system, and the kernel R is chosen to describe the smoothness and decay of the impulse response (using some hyperparameters that are estimated by empirical Bayes). Similar techniques have been used for the identification of nonlinear models [141], [142].

Following the discussion in "Simulation Errors and Prediction Errors," the model output $\hat{y}(t|\theta)$ will be a simulation output if it depends only on past inputs, and the corresponding model is a simulation or output error model. If the model output also depends on past outputs, then it is a predicted output, and the corresponding model a prediction model. This term does not necessarily imply that it is based on correct probabilistic treatment of the stochastic signals involved in the model. In the list of models that follows, it will be indicated how they comply with the general structures (8) and (9).

Summary Model Structures: User Guidelines

- *The user-selected model set*: This should reflect both the observed behavior aspects and the available physical insight.
- *Behavior aspects*: As an example, the presence of chaos or shifting resonances requires a closed loop around the nonlinearity. These aspects lead to a selection imposed by the system.
- *Physical insight*: This varies from physical models (white models) on one end to black-box models on the other that use only the available data without any physical insight.

APPROACH 2: FLEXIBLE PARAMETRIC NONLINEAR MODEL STRUCTURES

The other approach is to constrain the model by restricting the minimization in (S53) to a parameterized model set $\mathcal{F} = \{F(x,\theta) \mid \theta \in D_{\mathcal{M}}\}$. In the black-box situation, it is then a matter of finding such a set capable of allowing arbitrarily good approximations of any reasonable function F. In machine learning, the dominating black-box parameterization is the *neural net* (S64), (S69), (S70). The activation function is often chosen as a rectified linear unit, (S65c), which makes the parameterization a piecewise linear approximation over an adaptive grid of the function F. Deep networks (S66c) are common in current research, leading to deep learning.

DEEP LEARNING AS A SYSTEM IDENTIFICATION PROCESS

Machine learning and deep learning are closely related to system identification. To emphasize this, a session with the System Identification Toolbox [173] (see "Software Support") for the estimation of the Duffing oscillator models is shown in the Matlab code in Listing S1. See also Figures S19 and S20.

LISTING S1 The Matlab Code for Deep Learning

```
load SNLS80mV.mat
fs = 1e7/2^14;
edat = iddata((V2(30550:38500)', V1(30550:
  38500)', 1/fs)
load Schroeder80mV.mat
ySchroeder = V2(10585:10585 + 1023);
uSchroeder = V1(10585:10585 + 1023);
vdat=iddata(ySchroeder',uSchroeder',1/fs);
mbj=bj(edat,[4 4 2 2 0]); %linear model
compare(vdat,mbj),shg % shows the fit to
  simulating validation data
resid(vdat,mbj),shg
% Neural network with one layer and 40 nodes:
net=cascadeforwardnet(40]); N2 =
  neuralnet (net); %two layers
Deep network with two layers with 27 nodes each:
net=cascadeforwardnet([27, 27]);
  N2 = neuralnet(net);
mN1 = nlarx(edat,[4 4 0],N1);
mN2 = nlarx(edat,[4 4 0],N2);
[YH, FIT] = compare(vdat,mbj,mN1,mN2),shg
  %simulation fit
resid(vdat,mN1,mN2)
```

Sections of data are selected for estimation and validation; `edat` corresponds to Figure S22(a) and `vdat` is Figure S22(c). Then, a linear Box–Jenkins model is estimated and evaluated by simulation on validation data (compared with Figure S21), including computing the fit measure (32). The model is tested for validation using residual analysis. See "Linear Validation" under the "Model Validation" section.

Neural networks and deep learning are attempted with a deep net (S66c) with two layers. The System Identification Toolbox can use network models from the Deep Learning Toolbox in Matlab and incorporate them in the `idnlarx` object.

The numbers *FIT*-computed with the `compare` command show that the simulation fit for the linear Box–Jenkins model is 37.68%, while the deep network gives a fit of 99.26%, and the one-layer neural network returns 98.09%. The average norm of the simulation error for model `mN2` is

```
norm(vdat.y-YH{3}.y)/sqrt(length(vdat.y)) =
4.18e-4 (mV).
```

This validation simulation is shown in Figure S23 as *NARX NN*. It is compared with other nonlinear models on the same data and shows that the deep neural network model compares quite well with the best of the other models. The software syntax shows also that deep networks can be estimated, evaluated, and used much like the conventional linear models. The actual minimization of the criterion may be more cumbersome and require user interaction.

Static Nonlinearities

A static nonlinearity (no memory present) is the basic building block in nonlinear models. Consider the mapping

$$z = H(x); \quad z \in R^p, x \in R^m. \tag{S56}$$

If z and x are time-varying signals in the model, the mapping is applied for each t, $z(t) = H(x(t))$, so the mapping is *static*. Such a mapping can occur in many contexts in a nonlinear model and is indeed what constitutes the nonlinear behavior [see also (8)]. In a state-space model, H can be mapping from the state at time t to the state at the next time instant (20). In a regression model like (22), H can be the mapping from the regressors to the model output. In block-oriented models (Figure 6), the static nonlinearity block is a fundamental component. In this sidebar, the parameterization of static nonlinearities $H(x, \theta)$ will be discussed.

SINGLE-INPUT, SINGLE-OUTPUT NONLINEARITIES

Consider the simple single-input, single-output (SISO) case $p = m = 1$, which includes all of the essential ideas. This is the case of a 1D curve.

Break-Point-Based: Piecewise Constant and Piecewise Linear

A very simple idea to parameterize a curve is to define its values at a number of *break points* $\{x_1, \ldots, x_M\}$. So

$$\theta = \{x_k, z_k = H(x_k), k = 1, \ldots M\} \tag{S57}$$

would be the parameterization of the curve. An interpolation rule must then be applied to define $H(x)$ at the intermediate points. Standard options include a piecewise constant rule [the value $H(x)$ equals $H(x_k)$, where x_k is the break point immediately to the left of x] and a piecewise linear rule [the value $H(x)$ is linearly interpolated from its two neighboring break points]. Clearly, more sophisticated interpolation rules could also be used.

Basis Function Expansions

A common way to parameterize functions is to choose a system of *basis functions*, $\{\rho_k(x), k = 1, \ldots, M\}$ and parameterize the curve as

$$H(x, \theta) = \sum_{k=1}^{M} \theta_k \rho_k(x, \theta), \tag{S58}$$

where the basis functions ρ may or may not depend on θ. Many basis function expansions are possible, and a few will now be reviewed.

Custom Regressors

With some physical insights (or semiphysical modeling), the user can identify specially chosen basis functions $\rho_k^c(x)$ that reflect typical nonlinearities for the application in question. For example, using $\rho(x) = \sqrt{x}$ if $z = H(x)$ describes the output in a free-level flow system, where x is the level in the flow system. This gives the expansion

$$H(x, \theta) = \sum_{k=1}^{M} \theta_k \rho_k^c(x). \tag{S59}$$

Polynomial Expansion

The most common "black-box" expansion is the polynomial expansion $\rho_k(x) = x^k$ or the Taylor series

$$H(x, \theta) = \sum_{k=1}^{M} \theta_k x^k. \tag{S60}$$

Such polynomial expansions are used in modeling contexts where x consists of delayed inputs, also known as a *Volterra model* [see (S68)].

Linear Regressions

Note that, if the basis functions ρ_k are known [do not depend on the parameter θ, as in (S59) and (S60)], then (S58) is a linear regression, so it is easy to estimate θ from the measurement of z and x.

Local Basis Functions

It is also possible to construct local basis functions (pulses) that are nonzero only over certain intervals:

$$\rho_k(x, \gamma_k, \beta_k) = \begin{cases} 1 \text{ if } \gamma_k \le x < \gamma_k + 1/\beta_k, \\ 0 \text{ else.} \end{cases} \tag{S61}$$

Then, if $\gamma_{k+1} = \gamma_k + 1/\beta_k$, the expansion

$$H(x, \theta) = \sum_{k=1}^{M} \alpha_k \rho_k(x, \gamma_k, \beta_k) \tag{S62}$$

will be a piecewise constant function like (S57), with break points $\{\gamma_k\}$. It can approximate any reasonable function arbitrarily well with sufficiently large M. Note that

$$\rho_k(x, \gamma_k, \beta_k) = \kappa(\beta_k(x - \gamma_k)), \tag{S63}$$

where $\kappa(\cdot)$ is the unit pulse indicator, which is one for $0 \le x < 1$ and zero elsewhere.

Neural Networks

Inspired by the approximation capability of (S62) and (S63), a "mother function" $\kappa(x)$, also known as an *activation function*, can be selected. When translated by γ_k and dilated by β_k, this results in the expansion

$$H(x, \theta) = \sum_{k=1}^{M} \alpha_k \kappa(\beta_k(x - \gamma_k)). \tag{S64}$$

Common activation functions are the Gaussian bell (a soft pulse)

$$\kappa(x) = e^{-x^2} \tag{S65a}$$

and the sigmoid function (a soft step)

$$\kappa(x) = \frac{1}{1+e^{-x}}. \tag{S65b}$$

Another common choice of the activation function is $\kappa(x) =$ ReLU(x) (rectified linear unit):

$$\kappa(x) = \begin{cases} 0 & \text{if } x < 0, \\ x & \text{if } x \geq 0, \end{cases} \tag{S65c}$$

which makes H piecewise linear and continuous in x. The expansion (S64) represents the simplest *neural networks*.

Deep Neural Networks

The basis function expansion (S64) can be seen as a mapping from the regression variables x [if x is an n-dimensional column vector, the terms can be interpreted as (S69) or (S70)] to M new regressors:

$$x_k^{(2)} = \kappa(\beta_k(x-\gamma_k)) \quad k = 1, \ldots, M. \tag{S66a}$$

The mapping can be written compactly as

$$x^{(2)} = G^{(1)}(x, W^{(1)}), \tag{S66b}$$

where $W^{(1)}$ contains all of the coefficients in β_k and γ_k, and $G^{(1)}$ is an element-wise mapping from n regressors to M new regressors. These new regressors can, in turn, be subjected to a new mapping

$$x^{(0)} = x, \tag{S66c}$$

$$x^{(\ell)} = G^{(\ell)}(x^{(\ell-1)}, W^{(\ell)}), \quad \ell = 1, \ldots, L, \tag{S66d}$$

$$H(x,\theta) = W^{(L)} x^{(L)} - b, \tag{S66e}$$

where the last step gives the output of the model as an affine combination of the last step's regressors. The parameter θ contains all of the parameters in $\{W^{(\ell)}\ \ell = 1, \ldots, L\}$ and b. The model (S66c) is a deep neural network with L (hidden) layers, each with $\dim(x^{(\ell)})$ regressors or hidden units. There is a considerable current interest in deep networks as models. While one layer, as in (S64), is sufficient to create a universal approximator of any reasonable function, deep networks have more powerful approximation capabilities in practice [135].

Trees

A useful mapping $z = H(x)$ can be defined via (binary) trees. Such trees are popular mappings and used in, for example, decision and classification trees [136]. Here, a regression tree will be defined. The value x will be subjected to a sequence of binary questions at the different nodes. All of the binary answers will be collected and, based on those, a value $z = H(x)$ will be assigned.

At the root node, for a scalar x, the basic question is $x \geq c_0$?, for some number c_0. Depending on the answer, a new question will be asked at the next level $x \geq c_1^r$?, $r = 1, 2$. It continues so that, at level k, there are 2^k nodes with similar questions $x \geq c_k^r$?, $r = 1, \ldots, 2^k$. For a tree with depth n, the final level is $k = n$ with 2^n nodes, called *leaves*. The leaves correspond to a partition of the x axis into 2^n parts. Each leaf has a value d_k for $H(x)$ at the x-value that leads to that point. Some branches may be cut before level n. The leaves are then defined at a lower level. The tree defines a function $H(x)$, which is piecewise constant—all values of x that belong to the same partition give the same value d_k. A tree is an alternative to piecewise constant functions (S57) where the break points are defined in a contrived way by the tree partitioning. For a regression tree, it is customary to add an interpolation step to compute the function value:

$$H(x) = d_k + xL_k, \tag{S67}$$

where d_k is the value produced by the leaf of the tree and L_k is a scalar that is also provided by the leaf in question. The tree $H(x)$ becomes a *piecewise linear function* of x.

Gaussian Processes

Gaussian processes offer a nonparametric approach to model $z = H(x)$ in (S56) that can be used in a Bayesian framework [131]. This static model can then be used to form dynamic models, as in $f(x(t), u(t), \theta)$ and $h(x(t), \theta)$ in (20), or nonlinear autoregressive exogenous models and other examples in this article. The idea is that the function to be estimated is embedded in a stochastic framework, so that $H(x)$ is a realization of a Gaussian stochastic process. There is a prior (before any measurements) distribution with a zero mean, large variance, and a covariance function that describes the assumed properties of the function H (that is, smoothness, possible exponential decay, and periodicity).

When $z_k = H(x_k)$, $k = 1, \ldots N$ have been measured, the posterior distribution $H^p(x|z)$ can be formed for any x. The mean of that function will be the estimate of the function H and is formed by interpolation and extrapolation of the measurements $[z_k, x_k]$ using the probabilistic relationships in the prior distribution. The reliability of the estimate can also be assessed from the posterior variance. This is, of course, a reflection of the prior assigned distributions.

If all variables are jointly Gaussian, all of this can be computed by simple and efficient linear algebraic expressions. The idea is depicted in Figure S26. The framework can be seen as a generalization of the regularization approach [187] [see (24) and the discussion in "Black-Box Model Complexity"]. See [132] for a full introduction to the topic and [131] for a recent tutorial to apply Gaussian processes to the modeling of dynamic systems.

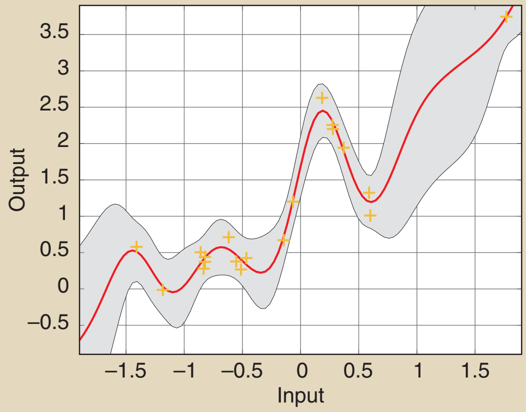


FIGURE S26 The yellow crosses are the observed x and y values. The red line is the estimate of the f function, and the shaded region shows the credibility, which is the standard deviation around the mean f.

MULTIPLE-INPUT, SINGLE-OUTPUT NONLINEARITIES

When the nonlinearity is multiple input ($m > 1$), x will be an m-vector. Most of the ideas for SISO models carry over to the multiple-input, single-output (MISO) case. Some specific aspects will be noted here.

Polynomial Expansion

In the polynomial expansion (S60), the term $\theta_k x^k$ should be interpreted as the sum of all those terms that can be created from exponentials of the components $x_i^{r_i}$ whose exponents sum to k. Each term x^k expands into several new terms. Each of these terms requires its own parameter, so the number of parameters in (S60) rapidly increases with m and M. A common special case of polynomial expansion is the Volterra model.

Volterra Models

A Volterra system [28] is a nonlinear finite impulse response system with a multivariate polynomial nonlinearity. Let the regressors be $\varphi(t) = [u(t), u(t-1), u(t-m+1)]$. Then, the model is

$$\hat{y}(t|\theta) = \sum_{\alpha=1}^{M} y_0^{\alpha}(t),$$

$$y_0^{\alpha}(t) = \sum_{\tau_1=0}^{m-1} \dots \sum_{\tau_M=0}^{m-1} g_\alpha(\tau_1, \dots, \tau_M) u(t-\tau_1) u(t-\tau_2) \dots u(t-\tau_M). \tag{S68}$$

The kernel $g_\alpha(\tau_1, \dots, \tau_\alpha)$ is the multidimensional impulse response of degree α and corresponds to the model parameters θ_k.

The number of terms in the Volterra models is $O(m^M)$, and it becomes very large when the memory length grows. For that reason, Volterra models were usually only applied to problems with short memory length m and moderate polynomial orders M. Only recently could more complex problems be handled [56], [183], using the regularization framework [187] to reduce the impact of the exploding number of parameters (see “Black-Box Model Complexity”).

Neural Networks

For the neural network model (S64), it is customary to keep the activation function $\kappa(x)$ with a scalar argument and reinterpret the argument with vector x in one of two ways. One approach is ridge construction, in which β_k is an m-vector, where

$$\kappa(\beta_k(x-\gamma_k)) \text{ is interpreted as } \kappa(\beta_k^T x - \gamma_k). \tag{S69}$$

This means that the basis function assumes the same value for all x in the hyperplane $\beta_k^T x = \text{constant}$, thus creating a ridge structure for the function values. With a sigmoid activation function, this leads to the celebrated *one hidden layer feedforward sigmoid neural net*. A second approach is radial construction, in which γ_k is an m-vector of translation and β_k is a positive definite matrix of dilation coefficients (or a scaled version of the identity matrix). Intepret

$$\kappa(\beta_k(x-\gamma_k)) \text{ as } \kappa(\|x-\gamma_k\|_{\beta_k}), \tag{S70}$$

where $\|x\|_{\beta_k}^2 = x^T \beta_k x$ is a quadratic norm defined by β_k. This means that the contribution of the basis function depends only on the distance between x and a given center point, thus providing a certain radial symmetry to the function. The Gaussian activation function gives the well-known radial basis neural network.

Trees

With an m-vector x, the questions asked at the nodes will be $[1 \;\; x^T]^T C_k^r > 0$? for $m+1$ vectors C_k^r. The linear interpolation term L_k in (S67) will also be an m-vector provided by the leaf in question.

MULTIPLE-INPUT, MULTIPLE-OUTPUT NONLINEARITIES

Turning to the multiple-output case $p > 1$, there is a common and simple solution: treat it as p independent MISO cases. If models are built from data, the actual models for the different outputs may appear in different shapes and different model orders. Gray-box thinking and/or experimental evidence may suggest that the same parameters may be used in different output channels to obtain a more efficient representation. This is noted in “Decoupling of Multivariate Polynomials,” see Figure S14. However, this does not change the essential features of multiple-input, multiple output representation of static nonlinearities captured when MISO models are constructed.

- » *White-box models*: In general, these are dedicated (a new model for each new problem), expensive, and compact and provide deep physical insight.
- » *Black-box models*: Such models offer a generic methodology, are sensitive to an explosion of the number of parameters, and provide little intuitive insight.

The following sections offer a unified view on the large variety of model classes. It may be convenient to extend the metaphor of white- and black-box models to a sequence of various shades of gray, indicating the amount of physical insight that will be used:

- » White models are physical models, and the values of the physical parameters are estimated.
- » Smoke-gray models are semiphysical models based on a natural selection and transformation of the measured variables using simple physical insights.
- » Steel-gray models are linearization-based models that depend on the working point and the nature of the excitation.
- » Slate-gray models are block-oriented models that are well suited to inject structural insight.
- » Black models are universal approximators, namely, Volterra, nonlinear autoregressive exogenous (NARX) models, nonlinear state-space models, and neural networks.
- » Pit-black models involve a nonparametric smoothing approach.

Snow-White Models

A model of a dynamical system is a collection of mathematical expressions that relate signals y and u and parameters and variables that characterize the system behavior. Important signals are the system's output signal(s) $y(t)$ that are of primary interest to be modeled; the input signal(s) $u(t)$ to the system, which are measurable signals that affect the outputs and may or may not be manipulated by the user; disturbance signals $w(t)$, which are unmeasurable signals that affect the outputs and typically described by random processes; and auxiliary signals $x(t)$ used in the model description. A model is then a collection of equations involving y, u, w, and x. To be a useful model, it must be possible to infer something about y from the other measured variables.

Deterministic Models

If no disturbances w are present, then it should be possible to compute $y(t)$ from previous values of $u(s); s \leq t$. That means that the model is deterministic or a simulation model or an output-error model.

Stochastic Models

If there are stochastic disturbances w present, the outputs also become stochastic variables. A specific value of $y(t)$ cannot be assigned based on the other variables, but a stochastic characterization of it must be used instead, that is, its probability density function (pdf) or mean value $\hat{y}(t)$. Since w is not observed, its values up to time t have to be inferred from the values of the observations $y(s)$ and $u(s), s < t$. The model values of $y(t)$ will then be conditioned on these variables, and the conditional mean $\hat{y}(t)$ will be a prediction based on past values.

To address these computability questions, it is necessary to be more specific about the structure of the collections of equations involving the model variables.

Deterministic Differential Algebraic Equation Models

For simpler notation, we introduce the vector $z(t) = [y(t), x(t)]^T$ for the outputs and auxiliary variables. Assume that there are no disturbances. When writing the equations that correspond to the physical knowledge of the system, differential equations are used in addition to algebraic relations among the variables. The equations can be written as

$$F(z(t), \dot{z}(t), u(t)) = 0, \tag{10}$$

where $\dot{z}$ is the time derivative of z. It is sufficient to consider first-order derivatives, since higher-order ones can be rewritten with the aid of extra x variables. Also, if $\dot{u}$ were to appear in (10), u can be included in z, and the basic form can still be applied. This is a differential algebraic equation (DAE) model. There is a clear conceptual relation between (10) and the general structure (8).

It is a normal case that (10) can be solved for z for a given u, and most software packages [108] for modeling systems contain DAE solvers. There is extensive literature [109], [110] that addresses this problem of the solvability of (10).

Deterministic State-Space Models

If $\dot{z}$ can be solved explicitly from (10), it follows that

$$\dot{x}(t) = f(x(t), u(t)), \tag{11a}$$
$$y(t) = h(x(t), u(t)), \tag{11b}$$

which is a standard, nonlinear state-space description for the relationship of u to y. If an initial state $x(0)$ is given, this equation has as a unique solution y under general and mild conditions. So, $y(t)$ will be a function of past $u(s)$:

$$y(t) = y(t \mid u^t); \quad u^t = \{u(s), s < t\}. \tag{12}$$

Stochastic State-Space Models

With disturbances present, it is customary and necessary to represent these as white noise w filtered through certain filters and assume that

$$x(t+1) = f(x(t), u(t), w(t)), \tag{13a}$$
$$y(t) = h(x(t), u(t)) + v(t), \tag{13b}$$

where $w(t)$ is a sequence of independent random variables with pdf $g_w(\cdot)$, and $v(t)$ is a sequence of independent random variables with pdf $g_v(\cdot)$. To avoid intricate issues

External or Internal Nonlinear Dynamics

DUAL REPRESENTATION OF LINEAR SYSTEMS

Infinite Impulse Response (IIR) Models

A linear system can be modeled by the recurrent representation

$$\begin{aligned} y(t) &= b_0u(t)+b_1u(t-1)+\cdots+b_{n_b}u(t-n_b) \\ &\quad -a_1y(t-1)-\cdots-a_{n_a}y(t-n_a) \\ &= h_{\mathrm{IIR}}(u(t),\ldots,u(t-n_b),y(t-1),\ldots,y(t-n_a)). \qquad \text{(S71)} \end{aligned}$$

The system (S71) can have an infinite memory (its impulse response has an infinite length) and, for that reason, is called an *IIR model*.

Finite Impulse Response (FIR) Models

The equivalent impulse response representation of (S71) is $y(t)=g(t)*u(t)=\Sigma_{k=0}^{\infty}g(k)u(t-k)$. Truncating this infinitely long impulse response to a finite length n leads to the FIR model

$$y(t)=g(t)*u(t)=\sum_{k=0}^{n}g(k)u(t-k)=h_{\mathrm{FIR}}(u(t),\ldots,u(t-n)). \qquad \text{(S72)}$$

Dual Representation of Linear Systems

From the previous discussion, it follows that linear systems can be represented either by an IIR model $y(t)=h_{\mathrm{IIR}}(u(t),\ldots,u(t-n_b),y(t-1),\ldots,y(t-n_a))$ or an FIR model $y(t)=h_{\mathrm{FIR}}(u(t),\ldots,u(t-n))$ (where n can grow to infinity).

From a behavior perspective, there is no difference between these representations. However, a structural difference is how the memory (dynamics) is created in both representations. The FIR model h_{IIR} makes no use of an internal memory; the dynamic behavior is obtained using delayed inputs. This is referred to as a model with external memory. The IIR model h_{IIR} includes delayed outputs to create an internal memory.

NONLINEAR FINITE IMPULSE RESPONSE AND INFINITE IMPULSE RESPONSE MODELS

Internal or External Dynamics

By choosing h in (S71) and (S72) to be nonlinear, the linear FIR-IIR classification can be generalized toward nonlinear IIR (NIIR) systems $y(t)=h_{\mathrm{NIIR}}(u(t),\ldots,u(t-n_b),y(t-1),\ldots,y(t-n_a))$ with internal dynamics (S72) or nonlinear FIR (NFIR) systems $y(t)=h_{\mathrm{NFIR}}(u(t),\ldots,u(t-n_b))$ with external dynamics (S71) [11], [15]. For notational simplicity, the h_{NFIR} and h_{NIIR} will both be written as h. The difference between these models follows from the arguments of the function.

For linear systems, it is a user's choice to select the FIR or IIR representation as there is a full equivalence between both. This is no longer the case for nonlinear systems. Not all NIIR systems can be modeled with an NFIR model, as explained further along. The choice between NFIR and NIIR systems/models affects the structure, behavior, and stability properties. In addition, the numerical methods for addressing these systems are strongly dependent upon it.

Structural Aspects

The NFIR/NIIR nature of a system is uniquely linked to its topology. For NFIR systems, there can be no dynamic closed loop around the nonlinearity, while, for NIIR systems, the nonlinearity is captured in a dynamic closed loop. This property is used in the structural detection framework [191]. It starts from the best linear approximation (BLA) (S30) that is identified for a varying input offset or amplitude [91]. The poles of the BLA remain fixed for NFIR systems, while they move for NIIR systems.

Remark 1

The class of systems included in the NIIR representation can be further generalized along with the nonlinear state-space models that have a nonlinear feedback over some of the states, such that internal dynamic nonlinear closed loops are created.

Behavior Aspects

Some typical nonlinear behaviors like chaos, bifurcations, jumps, moving resonances, and autonomous oscillations (see Figure S27) can be generated only by NIIR systems. NFIR models cover fading memory systems whose behavior is more closely related to that of a linear system. Fading memory systems [123] forget the past inputs asymptotically over time. In other words, a nonlinear system that has no fading memory requires an NIIR model.

Fading Memory Systems

NFIR systems are a subset of fading memory systems. All stable NFIR systems have a fading memory, unless the nonlinearity would be discontinuous. The simplest example of such an exception is a static discontinuous system. An NIIR system can have a fading memory behavior on a restricted input domain [for example, the Duffing oscillator in the section

"Linear Simulation and Prediction of the Forced Duffing Oscillator" (S46)] and, on that input domain, NFIR models can be used to approximate the NIIR system. This is illustrated on the Duffing oscillator in [29].

However, the output of nonlinear NIIR systems can also show bifurcations and jump phenomena (small variations at the input can result in sudden qualitative or topological changes in the output), even for systems with smooth nonlinearities [8], [16], [17], [27]. The output can become chaotic, or autonomous oscillations can appear. An NFIR model cannot model these phenomena, illustrated in Figure S27.

Stability

While the stability of NFIR systems can be guaranteed under very general conditions, it is much harder to analyze the stability of NIIR systems. Although general theories exist to analyze and impose stability on NIIR systems [14], the user must often complete extensive simulations to check stability [15].

Numerical Aspects

The numerical optimization aspects of nonlinear systems/models are strongly affected by moving from NFIR to NIIR. Calculating the derivative of the output of an NIIR system with respect to the model parameters means computing the output of another nonlinear model [9]. Table S2 gives an overview of the NFIR and NIIR systems considered in this article. See [97] for a more extensive table of different models and behaviors.

SUMMARY: USER GUIDELINES

- *Internal or external dynamics models*: The choice between an external dynamics NFIR model (with no nonlinear closed loop) or an internal dynamics NIIR model (with a nonlinear closed loop present) strongly influences the complexity of the identification problem.
- *Initial tests or insight*: Initial tests or insight can help to choose between the two model classes. This avoids wasted time and effort in trying to fit the wrong model structure to the data.
- *External dynamics model class*: Only systems with a fading memory behavior can be modeled (with a small error) with an external dynamics model. Typical models are NFIR, Volterra, and open-loop block-oriented models (Wiener, Hammerstein, Wiener–Hammerstein, and Hammerstein–Wiener). These models can be used only if the poles of the BLA do not move for varying experimental conditions.

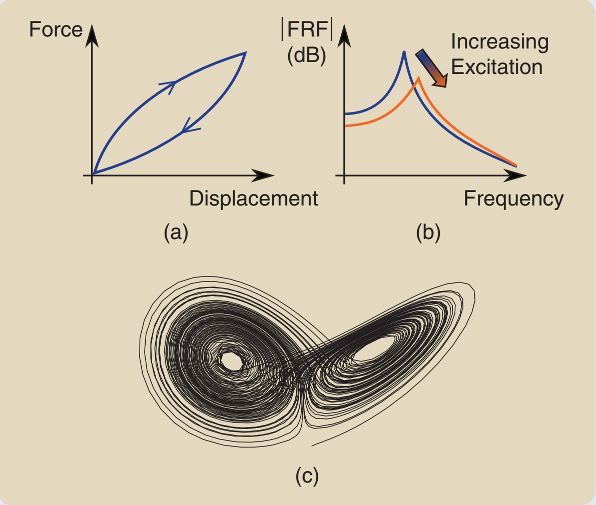


FIGURE S27 (a) Hysteresis, (b) moving resonance frequency, and (c) chaotic behavior are typical nonlinear phenomena that can be modeled only by systems with a nonlinear feedback loop.

TABLE S2 Systems/models with external or internal nonlinear dynamics.

External Nonlinear Memory No Nonlinear Output Feedback	Internal Nonlinear Memory Nonlinear Output Feedback
Nonlinear finite impulse response	Nonlinear autoregressive exogenous
Volterra	
Open loop, block oriented	Closed loop, block oriented
Nonlinear state-space lower triangular	Nonlinear state-space full

- *Internal dynamics model class*: Strong nonlinear behaviors like moving resonances, jump phenomena (bifurcations), hysteresis, and autonomous oscillations can be modeled only with internal dynamics (NIIR) models. External dynamics (NFIR) models are not able to represent such phenomena. Typical models are nonlinear autoregressive exogenous (NARX) models, block-oriented models with feedback, and nonlinear state-space models. Whenever the poles of the BLA move for varying experimental conditions, this class of models is preferred.

Approximating a Continuous-Time Nonlinear State-Space Model With a Discrete-Time Model

Although real-world systems evolve in continuous time, discrete-time models are preferred in many control applications because most simulations and many controllers are implemented digitally [99]. However, without special care, the discrete-time approximation can have a completely different behavior than the original continuous-time system [100], [101]. To provide a better understanding of these problems, the discretization of linear systems is first considered, and next the discrete-time approximation of continuous-time nonlinear systems is discussed.

DISCRETIZATION OF LINEAR SYSTEMS

Zero-Order-Hold Discretization

In linear system identification, it is well known that a continuous-time system that is excited by a zero-order-hold (ZOH) input (also called *piecewise constant excitation* [6]) can be replaced by a discrete-time model that gives an exact description of the discrete-time, input–output relations [6], [7], [106]. Generalizing this result to nonlinear systems is not always possible because the ZOH character can be lost inside the system. For example, the output of the feedback in nonlinear closed-loop systems will no longer be ZOH. For that reason, solutions that do not rely explicitly on the ZOH nature are needed.

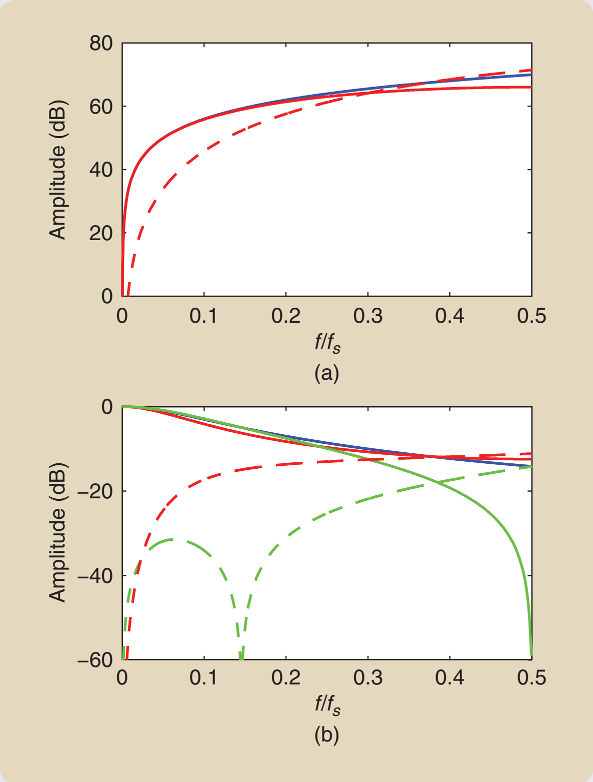


FIGURE S28 The approximation of a continuous-time system by a discrete-time system. (a) A discrete-time approximation of a differentiator: $G(s)=s$ (blue) and $G(z)=(1-z^{-1})f_s$ (solid red). Observe that the error (broken line) grows fast with the frequency to 100% at $f/f_s \approx 0.3$. (b) The approximation of a continuous-time first-order system $G(s)$ (blue) by a discrete-time system $G_1(z)$ obtained by a finite difference transformation $s \to (1-z^{-1})f_s$ (red) and a first-order discrete-time system $G_2(z)$ (green) that is fitted in a least-square sense in the bandwidth $[0\,0.25]f_s$. Observe that the error of the finite difference solution is very small, around $f=0$, but grows very fast.

Alternative Discretizations

Replacing the continuous-time differentiation by a finite difference is intuitively very appealing:

$$\frac{dx(t)}{dt} \approx \frac{x(t)-x(t-T_s)}{T_s} = (x(t)-x(t-T_s))f_s \qquad \text{(S73)}$$

or in the Laplace- and Z-domain

$$s \to (1-z^{-1})f_s. \qquad \text{(S74)}$$

In the frequency domain, $s=j2\pi f$ and $z^{-1}=e^{-j2\pi f/f_s} \approx 1-j2\pi f/f_s+O((f/f_s)^2)$, and (S74) becomes

$$s \to (1-z^{-1})f_s \approx j2\pi f + O((f/f_s)^2). \qquad \text{(S75)}$$

This simple solution works well only if the sample frequency f_s is much larger than the frequency band of interest. In Figure S28(a), it can be seen that the relative error grows to 100% for $f > 0.3f_s$. In Figure S28(b), the transformation (S74) is applied to a continuous-time first-order system

$$G(s)=\frac{1}{1+\tau s} \to G_1(z)=\frac{1}{1+\tau f_s-\tau f_s z^{-1}}. \qquad \text{(S76)}$$

Observe that the error is very small around the origin but grows very fast to 100% for $f > 0.3f_s$.

A very popular approach in the system identification community is to identify directly, in the frequency band of interest, a discrete model $G(Z)$ for the continuous-time system $G(s)$. This is also illustrated in Figure S28(b). A first-order discrete-time model $G_2(z)$ is obtained by a least-squares fit in the frequency band $[0,0.25]f_s$. Observe that the error of $G_2(z)$ is much smaller than that of $G_1(z)$. See the signal processing literature for more information on the classical solutions to this problem (for example, impulse invariant transformation, bilinear transformation) [107].

DISCRETIZATION OF NONLINEAR SYSTEMS

Finding good discrete-time approximations for continuous-time nonlinear systems has been studied extensively [19], [98], [102], [103]. As was done for linear systems, the discussion starts with dedicated methods for ZOH excitations, followed by an approach to address the more general class of lowpass excitations.

ZOH Framework

An exhaustive overview of the discretization problem is made in [99] and the references therein. The authors look for an approximate discrete-time representation, operating in a ZOH framework. Some of their major conclusions are as follows:

- The popular forward Euler method should be used with extreme care because it results in relative errors that grow fast with f/f_s, leading to the need for a very high oversampling. This confirms the observation made in Figure S28.
- The dynamics of sampled data models can be different from those of the continuous-time system; numerical sampling zeros are created.
- A truncated Taylor series (TTS) approximation model is proposed to estimate a continuous-time nonlinear state-space (NLSS) equation by a discrete-time equivalent. The global fixed-time truncation error (the error when integrating over a fixed time interval) drops to zero, proportional to the inverse of the sampling frequency f_s as an $O(\Delta = 1/f_s)$.

Low-Pass Framework

An alternative approach for the approximation of a continuous-time NLSS model is presented in [104]. Consider

$$\begin{aligned}\frac{dx(t)}{dt} &= f(x(t), u(t)),\\ y(t) &= h(x(t), u(t)),\end{aligned} \tag{S77}$$

which is excited with an input signal $u(t)$ that is a low-pass signal of degree d_u [the square root of the power spectrum is an $O(1/f^{d_u})$ for $f > f_c$]. Observe that a ZOH excitation is a low-pass signal with $d_u =$ 1. Similar to the identification approach for the linear first-order example in the previous section, a discrete-time NLSS model

$$\begin{aligned}x_d(k+1) &= F_d(x_d(k), u_d(k)),\\ y_d(k) &= G_d(x_d(k), u_d(t))\end{aligned} \tag{S78}$$

is identified directly from the data. The discrete-time signals $x_d(k)$ and $y_d(k)$ equal the sampled continuous-time signals $x(t)$ and $y(t)$ within an error

$$\begin{aligned}x_d(k) &= x(kT_s) + O(1/f_s^{d_x-1.5}) + \varepsilon_l,\\ y_d(k) &= y(kT_s) + O(1/f_s^{d_x-1.5}) + \varepsilon_l.\end{aligned} \tag{S79}$$

$d_x = \min(d_u, d_F) + 1$, with d_F a characteristic of the nonlinear system, as explained in Figure S29 [104]. For a static discontinuous nonlinear function, $d_F = 1$ and $d_F = n + 1$ if the nth derivative exists in the domain of interest [104], [105]. The error ε_l is a user choice set by the choice of the complexity of the discrete-time NLSS model. The discretization error is dominated by the aliasing effect, which is due to the sampling of the input, output, and internal signals. The error in the direct identification approach is smaller than that of the TTS ZOH approach. This is because the discrete-time model is tailored to the continuous-time data in the least-squares minimization step. This result provides a strong theoretical foundation for the popular and successful practice to directly identify explicit discrete-time models of continuous-time nonlinear systems.

ILLUSTRATION: DISCRETIZATION OF THE DUFFING OSCILLATOR

The results of the previous section are illustrated on the Duffing oscillator setup (Figure S18) described in “Experimental Setup: The Forced Duffing Oscillator.” In this case, the ZOH output $x_{\mathrm{ZOH}}(t)$ of the generator filter is filtered by a fourth-order

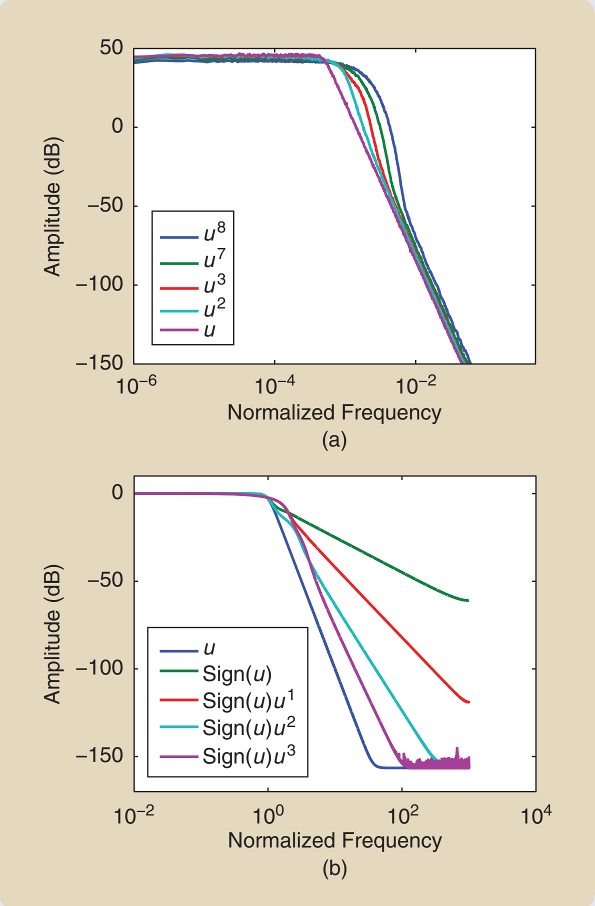


FIGURE S29 A study of the amplitude spectrum of a low-pass signal of relative degree $d_u = 5$ that passed through a nonlinear system [104]. The amplitude spectrum of the output normalized to a root mean square value of one is shown, and the dc value is not shown. (a) The results for a static nonlinear system $y = p^n$, $n = 1, 2, 3, 7, 8$. Observe that all of the signals have the same relative degree, independent of n. In this case, the relative degree of the output is set by the relative degree of the input. (b) The results for a static nonlinear system $y = \mathrm{sign}(u)u^n$ with $n = 1, 2, 3$. In this case, the relative degree is set by the nonlinear system: $d_F = n + 1$ for $d_F \le d_u$.

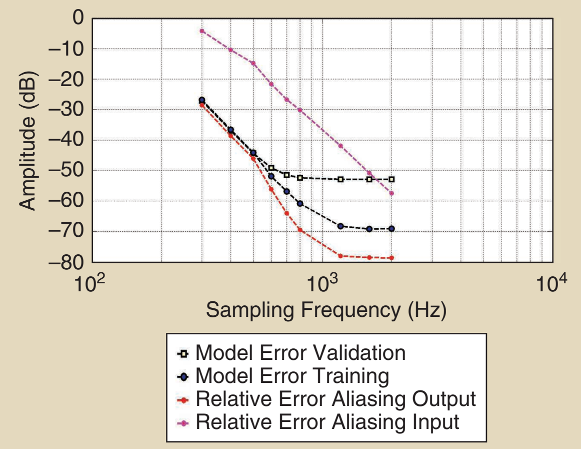


FIGURE S30 A discretization of the Duffing oscillator [104]. An experimental verification on the Duffing oscillator shown in Figure S18 is made. A discrete-time model nonlinear state-space model is identified, and the root mean square value of the simulation error on the estimation set and a validation set is plotted as a function of the sample frequency. It is also compared to the relative alias error of the input and output. The alias error drops with about 70 dB/decade, which is in perfect agreement with the presence of the fourth-order filter in the generator path. It is also seen that the error on the modeled output follows the slope of the aliased power of the input and output, as was expected from the theory. At lower error levels, structural model errors dominate. These could not be further reduced by increasing the complexity of the model.

low-pass filter with a cutoff frequency of 200 Hz. A wideband zero-mean excitation with a bandwidth of $f_s/2$ is used as the excitation signal in the frequency band [0–39,062 Hz]. With these settings, the signal $u_c(t)$ has a flat amplitude spectrum up to 200 Hz, to be compared with the bandwidth of the second-order system that is below 100 Hz.

The continuous-time input and output are sampled at a high sampling frequency $f_s = 78{,}125$ Hz. Next, the data are subsampled at different rates [104]. For each subsampled data set, a discrete-time nonlinear state-space model is identified on the measurements, using a discrete-time nonlinear polynomial state-space model and employing the methods described in [188] and "Extensive Case Study: The Forced Duffing Oscillator." The model has two states, and the degree of the internal multivariate polynomial is three (it depends on both the states and the input). Higher orders and degrees were tested, but this did not significantly improve the results. The results are shown in Figure S30. It shows that the errors drop as an $O(1/f_s^4)$ until the noise floor or the structural model error floor is reached. The drop rate is proportional to the drop of the alias errors and much faster than $1/f_s$.

SUMMARY

Different options are available to approximate a continuous-time system/model by a discrete-time model.

- *Data-driven approach*: A nonlinear discrete-time model can be identified directly from the experimental data. The approximation error is dominated by the alias error that is $O(1/f_s^{d_x-1.5})$. The model complexity of the discrete-time model can be higher than that of the continuous-time counterpart to keep the discretization error below a user-defined error level.
- *Model-based approach*: If a continuous-time model is available, it is possible to turn it into a discrete-time one. Two options are available for the user.
  - o *TTS approximate model*: The continuous-time equations are transformed into a set of discrete-time equations using a Taylor series approximation. The major advantage is that there is a close connection with the original, physical equations. The main drawbacks are 1) the restriction to ZOH excitations and 2) the approximation error is $O(f_s^{-1})$, which drops slowly with the sample frequency.
  - o *Simulation-identification approach*: The continuous-time model is used to create a rich data set used as the input for a direct fit of a discrete-time model. The major advantages are 1) a compact model is obtained with a user-controllable balanced quality/complexity and 2) by a proper design of the data set, the user can focus the model on the intended application. The major disadvantage is the loss of physical interpretability of the model.

with continuous-time white noises, only the discrete-time case is considered. Using the state-space forms (13), the links with the general predictor model structure (9) become clear.

The problem of finding the conditional distribution or conditional mean $\hat{y}(t|t-1)$, given past input–output signals, is the well-known nonlinear prediction or filtering problem. If f and h are linear and v and w are Gaussian variables, this is solved by the familiar Kalman filter [111]. In the general case, the filtering problem has no closed-form solution. However, there has been recent progress on a numerical solution in terms of particle filters [49], [112]. A simplistic but mostly erroneous way to address the nonlinear filtering problem is to assume that all disturbances can be collected as white noise added to the output. Then, the model will correspond to an output error model like (11) (see also "Process Noise in Nonlinear System Identification").

User Guideline

In summary, the work to construct a snow-white model, sometimes called first principles modeling or mechanistic modeling, is typically time consuming and laborious, and

Software Support

The algorithms needed for nonlinear model definition, estimation, and analysis are typically quite complex and need sophisticated computer software support. Several program packages for such support are publicly available, some for free download and others for license agreements. Some of these packages will be briefly described.

THE SYSTEM IDENTIFICATION TOOLBOX FOR USE WITH MATLAB

The System Identification Toolbox [173] is a commercially available program package, distributed by MathWorks Inc. It contains implementations of several of the methods/models discussed in this article, as follows:

- `idnlarx` is a model object that handles the nonlinear autoregressive exogenous model (22) with several possible nonlinearities.
- `idhw` handles the Hammerstein–Wiener block-oriented model in Figure 6.
- `idnlgrey` treats nonlinear state-space gray-box models (20), where *f* and *h* must be programmed by the user in Matlab or C-code.

Static nonlinearities (S56) can be chosen, for example, as

- `pwlinear`: (S57)
- `poly1d`: (S60)
- `sigmoidnet`: (S64)
- `treepartition`: (S67)
- `customnet`: (S59)

for use in `idnlarx` and `idnlhw`. All model validation and evaluation commands, namely, `compare` (for cross validation); `resid` (for linear residual analysis); and `sim`, `predict`, and `forecast` (for simulation and forecasting) are available, analogous to linear models, and can be used to compare linear and nonlinear models in the same command.

NONPARAMETRIC NOISE ANALYSIS

The freely available frequency-domain identification toolbox FDIDENT can be used to make the nonparametric noise and distortion analysis (see "Nonparametric Noise and Distortion Analysis Using Periodic Excitations" and http://home.mit.bme.hu/~kollar/fdident/). This toolbox also includes the tools to design the random-phase multisines (26) and perform the nonparametric nonlinear analysis. In [151], all of the procedures for the noise and distortion analysis presented in this article are discussed in full detail, and the related Matlab software can be freely downloaded from booksupport.wiley.com.

NONLINEAR STATE-SPACE MODEL

A package to identify a polynomial nonlinear state-space model (see "Extensive Case Study: The Forced Duffing Oscillator") is freely available at http://homepages.vub.ac.be/~jschouk/.

NONLINEAR BENCHMARK WEBSITE

The website http://www.nonlinearbenchmark.org has many experimental data sets that are well documented. For each benchmark, a detailed description of the setup and the experiments is given. In addition, a list of references is provided to publications that process these data. The data for "Extensive Case Study: The Forced Duffing Oscillator" are available on this website.

it does not involve any system identification. It is supported by software like the object-oriented languages Modelica [113] and Simscape [114].

Off-White Models

Snow-white modeling includes one or several physical constants whose numerical values are not known. If the values cannot be established by separate measurements, they must be included as a parameter θ in the model. In the deterministic state-space case, the model then takes the form

$$\dot{x}(t) = f(x(t), u(t), \theta), \tag{14a}$$

$$\hat{y}(t|\theta) = h(x(t), u(t), \theta), \tag{14b}$$

where $\hat{y}(t|\theta)$ is the output corresponding to the specific parameter value θ. In the stochastic case, the symbol is the predicted output time t based on the model with parameter θ. Since its value is unknown, the model is no longer snow white but has become an *off-white* model. Such models are also known as *gray-box models* [115], but, as follows from the ensuing discussion, there are several shades of gray.

Building an off-white model requires the same effort as a snow-white model, which might be significant. Also, it may not be possible to retrieve all of the physical parameters from an identification experiment.

The off-white model (14) is a clear case of the general predictor structure (9). In addition, it complies with (8) by using F to solve the underlying nonlinear differential equation. An application of off-white model identification to a cascaded tanks system is given in (33).

User Guideline

In summary, off-white model identification requires substantial modeling. It is important to realize that any deficiencies in the physical model may cause corruption in the physical parameter estimates.

Smoke-Gray Models: Semiphysical Modeling

Semiphysical modeling is physical modeling with a more leisurely attitude regarding the physics. It could use

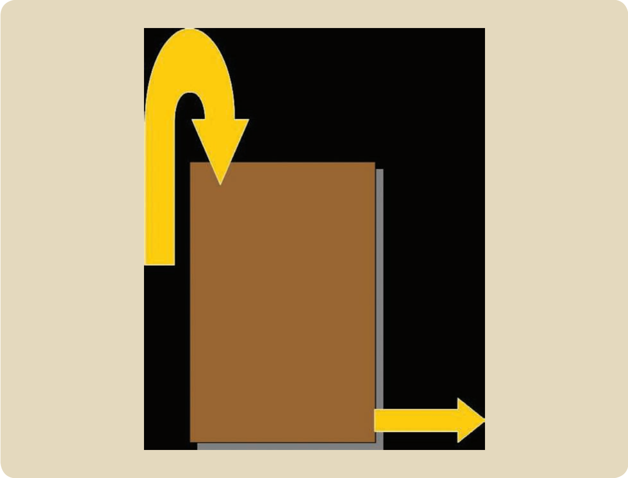


FIGURE 5 A schematic picture of a buffer vessel with inflow and outflow.

qualitative reasoning rather than formal equations. Take, for example, a voltage-controlled dc motor, with input applied voltage u and output motor shaft angle y. A known load disturbance torque L also acts on the shaft. Well-known physical laws demonstrate that the applied voltage to the rotor circuit is split between the internal resistance in the rotor winding and the back electromotive force resulting from the motion of the winding in the magnetic field. The latter is proportional to the rotational speed ω. The torque T from the magnetic field is proportional to the current in the winding, and the resulting torque on the shaft is T-L minus the frictional torque (which is typically proportional to ω). The resulting torque will (by Newton's law of motion) be proportional to the rotor acceleration $\dot{\omega}$. This means that the voltage u will be a linear combination of $\omega, \dot{\omega}$, and L. Since ω is the derivative of y, it follows that

$$x = \begin{bmatrix} y \\ \omega \end{bmatrix}, \tag{15a}$$

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & \alpha \end{bmatrix} x + \begin{bmatrix} 0 \\ \beta \end{bmatrix} u + \begin{bmatrix} 0 \\ 1 \end{bmatrix} L. \tag{15b}$$

This is, of course, the same model that would have been obtained with careful white modeling, with the difference that α and β would have been expressed in physical constants of the motor (that is, internal resistance, friction coefficient, rotor moment of inertia, and magnetic field characteristics). That is not really essential, since the only physical constants that can be retrieved from input–output data are the combined expressions of α and β.

Another useful form of semiphysical modeling is finding *nonlinear transformations of the measured data*, so that the transformed data have a better chance to describe the system in a linear relationship. To give a trivial example, consider a process where water is heated by an immersion heater. The input is the voltage applied to the heater, and the output is the temperature of the water. Any attempt to build a linear model from voltage to temperature will fail. It is clear that the power of the heater is the driving stimulus for the temperature. Thus, let the squared voltage be the input to a linear model generating water temperature at the output. Despite the trivial nature of this example, it is good to note as a template for data preprocessing. Many identification attempts have failed because of the lack of adequate semiphysical modeling. See [6, ex. 5.1 and pp. 533–536] for more examples.

Nonlinear recalibration of the time scale is in the family of nonlinear transformations of measured data. Many systems have a natural time maker. This could be a rotational system, where the angle of the rotation is a natural time unit, or a flow system, where the accumulated amount of transported substance provides a natural time flow (see the case in the section "Example 2: Smoke-Gray Model (Semiphysical Model): An Industrial Buffer Flow System" and Figure 5).

The most common and important application of semiphysical modeling is to *concatenate known submodels*. A simple example is the dc case cited previously, which is a concatenation of a dc motor model from u to ω and an integrator. Having libraries of basic element models from which more complex models are built using simple physics and logic is now the most common modeling approach in many application areas. Modeling languages like Modelica are based on this principle, and extensive Modelica libraries exist for most applications.

User Guideline

It is always important to consider the physics of the system to be identified, even if a complete off-white model is not constructed. Such semiphysical modeling can provide insights into important nonlinear transformations that can be essential components in the model.

Steel-Gray Models: Linearization-Based Models

Linear Models of Nonlinear Systems in Identification

It is well known that a nonlinear model like (14) can be linearized around a stationary point x^* [see "Linear Models of Nonlinear Systems" and (S32)]. Using linearized models may be the most common way to address nonlinear systems in practice. Identifying a linearized model is normally not done by going through the linearizing differentiations (since the nonlinear models are typically unknown). Instead, one simply postulates a linear model structure and applies normal linear system identification. If the excitation keeps the system in the close vicinity of the stationary point x^*, the identified model will be close to the stationary point linearization. In general, the identified model will be a stochastic linearization, or BLA (S33), reflecting the input and output signal spectra of the identification data. In any case, the usefulness of the model may be limited, since it describes only the system in the vicinity of x^*.

Local Linear Models

An obvious remedy to the local nature of the linearized model is to work with several linearized models and connect them in some way to address the global behavior of the system. The many ways to do this have led to an extensive literature on local linear models [15]. Several other terms are used as well.

The basic idea is to divide the state space into regions within which a linear model is used to describe the system. We use a collection of measurable regime points $P = \{p_i, 1 = 1, \dots, d\}$ that define the centers of the regions. Each point is like a stationary point x^* of a state-space model but does not need not to be formally defined. Each of these points is associated with a linear model, generically characterized by its output prediction $\hat{y}_i(t \mid \theta, Z^{t-1})$. The type of linear model is arbitrary, and some examples are given subsequently. When the system is at the regime point p_i, there is a clear linear model for predicting the output. At other points $p(t)$, the prediction is interpolated from the values in P by

$$\hat{y}(t \mid \theta, Z^{t-1}) = \sum_{i=1}^{d} \rho(p(t), p_i)\hat{y}_i(t \mid \theta, Z^{t-1}), \tag{16}$$

where ρ is a weighting or *validity function* that describes the validity of model i at another value p of the regime variable. This is the archetype of a local linear model. This is explicitly of the form of (9). In terms of (8), the nonlinearity hides in the weights ρ and the shifts between linear models that they represent. For a concrete case, the following items must be specified:

» What are the measurable regime points?

» How is the collection P determined?

» What type of linear models $y_i(t \mid \theta, Z,^{t-1})$ are used?

» How does one choose the validity function $\rho(p, p_i)$?

Note that regime points are often naturally defined by the application and may typically be part of the state vector. They correspond to operating conditions that are known to provide well-defined behavior. Typical cases could be fluid levels in flow systems applications and speed and altitude in flight applications. The regime points can also be determined from data, as in a tree-based construction in the local linear model tree (LOLIMOT) [15]. Furthermore, there is a wide range of linear model types available. The simplest one is the ARX model, which predicts the output as a linear combination of past inputs and outputs [for instance, $\hat{y}(t \mid \theta) = -a_1 y(t-1) + b_1 u(t-1)$ in a first-order case]. Associating the rth regime point model with parameters using superscript (r), the complete model (16) is

$$\hat{y}(t \mid \theta, Z^{t-1}) = \sum_{r=1}^{d} \rho(p(t), p_r)[b_1^{(r)} u(t-1) - a_1^{(r)} y(t-1)], \tag{17}$$

which is a linear regression with parameters $\theta = [a_1^{(r)}, b_1^{(r)}, r = 1, \dots, d]$. There is vast literature on multiple and local linear models [15], [116], [117], where the latter use fuzzy sets for the model interpolations.

Linear Parameter-Varying Models

A related concept is that of *linear parameter-varying* (*LPV*) models. In state-space form, they can be described as

$$x(t+1) = A(p(t))x(t) + B(p(t))u(t), \tag{18a}$$

$$y(t) = C(p(t))x(t), \tag{18b}$$

correspondingly, in continuous time. Disturbances can also be added. The function $p(t)$ is a measured regime variable. Formally, (18) is not a nonlinear system but rather a linear, time-varying system. However, if $p(t)$ in some way depends on the state x, this can be a handy way of addressing a nonlinear system. There is an extensive literature on dealing with and identifying LPV models [118]–[120]. If $p(t)$ assumes only a finite number of different values, (18) is really a collection of local linear models, and several identification schemes can be based on the ideas in the previous section and/or on handling time-varying linear systems. An important difficulty for LPV systems is keeping track of the state-space basis when $p(t)$ is changing [121].

User Guideline

In summary, using linearization is a standard tool to handle nonlinear physical systems. Many possibilities exist to merge linearized pieces into a good nonlinear model.

Slate-Gray Models: Block-Oriented Models

A common and useful family of models is obtained by the concatenation blocks of two types: linear dynamic models $y = G(s)u$ and static nonlinearities $y(t) = f(z(t))$. Many such combinations have direct physical interpretations (see Figure 6), as with the Wiener model [a linear model followed by a static nonlinearity $y = f(G(s)u)$], which describes a linear plant with a nonlinear output sensor, and the Hammerstein model [a static nonlinearity followed by a linear system $y = G(s)f(u)$], which depicts a linear model controlled via a nonlinear saturating actuator.

Such *block-oriented* models have a flavor of the smoke-gray, semiphysical models, constructed by simple engineering insights of a qualitative nature. However, beyond that, block-oriented models possess interesting approximation properties. It is known, for example, [122], that the parallel Wiener model in Figure 6, with sufficiently many linear branches, can approximate fading memory systems $u \rightarrow y$ arbitrarily well. The convergence rate can be improved by switching to parallel Wiener–Hammerstein models [123]. Therefore, block-oriented models can be used for many nonlinear systems in appropriate configurations without any physical interpretation and achieve good modeling results. This is in the spirit of black-box models (see the next section), which motivates giving block-oriented models a darker shade of gray than smoke gray.

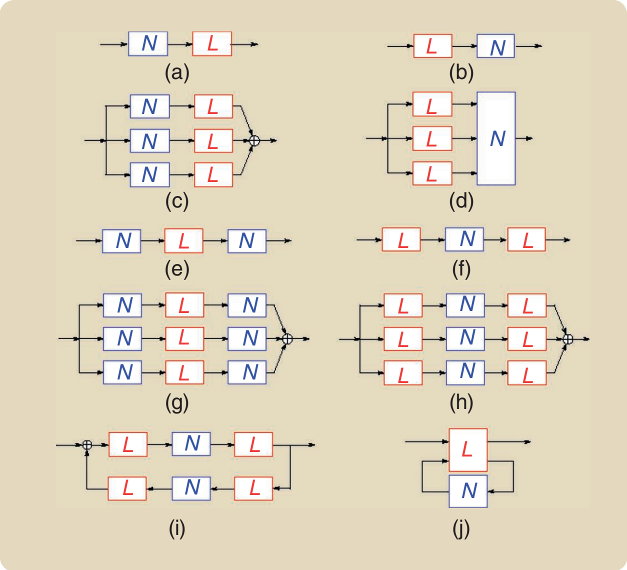


FIGURE 6 Examples of block-oriented models [189]. The linear blocks *L* capture all dynamics while the static nonlinear blocks *N* are used to model the nonlinearity. (a) The Hammerstein model can include the actuator nonlinearities, and (b) sensor nonlinearities are covered by the Wiener model. The (c) parallel Hammerstein and (d) parallel Wiener models are also shown. (e) The combined effects are covered by the Hammerstein–Wiener model. (f) The Wiener–Hammerstein model describes a static nonlinearity with an input- and output-matching network. (g) The parallel Hammerstein-Wiener model is shown. (h) The parallel Wiener–Hammerstein structures are universal approximators for fading memory systems [122], [123]. (i) and (j) The feedback structures are a generalized representation of the Duffing oscillator studied in "Simulation Errors and Prediction Errors." Note that, in the parallel structures, either a multiple-input, multiple-output nonlinear block (for example, in the parallel Wiener model) or a set of single-input, single-output nonlinear blocks (as in the parallel Wiener–Hammerstein model) can be used.

The block-oriented models clearly fit the general structure (8), with linear blocks in combination with static nonlinearities. To find the predictor (9), it is necessary to follow the signal flow through the linear and nonlinear parts. If there is process noise, a correct calculation of the predictor may require the use of advanced statistical techniques, namely, particle filters [55] (see also "Identifying Nonlinear Dynamical Systems in the Presence of Process Noise").

The estimation of a block-oriented model follows the general minimization of fit between the model's predicted output and measured output. However, before that can be done, considerable work may be required to determine the structure and initial parameter values. The concept of the BLA [see "Linear Models of Nonlinear Systems" and (S33)] is useful in this context. A comprehensive account of estimation techniques for block-oriented models is given in [18] and [124].

The Hammerstein, Wiener, Hammerstein–Wiener, and Wiener–Hammerstein models (including the parallel structures) are all examples of nonlinear systems with external nonlinear dynamics (see "External or Internal Nonlinear Dynamics"). The nonlinear blocks are not captured in a dynamic feedback loop. The Wiener–Hammerstein feedback and the linear fractional representation feedback systems are both nonlinear systems with internal nonlinear dynamics because the nonlinear block N is part of a dynamic closed loop. Addressing multiple branches [125] and feedback structures [126] is extremely challenging.

User Guideline

In summary, block-oriented models form a powerful and intuitive tool to handle nonlinear systems. It is often useful to try a simple Wiener or Hammerstein model to see if nonlinear model components show significant improvements over linear models.

Black Models: Universal Approximators

So far, the starting point has been some kind of physical or behavioral aspect of the system when constructing the model. However, the model—the predictor—is a mapping from past input–output data to the space where the output lives. Lacking insights into the system, the focus is on building general flexible mappings that are universal and effective approximators of any reasonable predictor function, which is the idea behind black-(box) models.

A General Structure of the Mapping

A general way to generate very flexible mappings from Z^{t-1} to $\hat{y}$ is to construct a state x from past input–output data, and let the predictor be a general function of this state

$$x(t+1) = f(x(t), u(t), y(t), \theta), \tag{19a}$$

$$\hat{y}(t\,|\,\theta) = h(x(t), \theta), \tag{19b}$$

where f and/or h are flexible, static, nonlinear functions of their arguments $x(t)$, $u(t)$, and $y(t)$. The possibilities of parameterizations to reach flexibility are discussed in general terms in "Static Nonlinearities." The model (19) is of the general predictor form (9) and links to the structure (8) through the nonlinear maps f and h. However, working with both f and h may be too general, so two cases will be discussed further.

Unknown State Transition Function: Nonlinear State-Space Models

Perhaps the most natural general black-box approach is to postulate a general state-space model like (14) in discrete time (see also "Approximating a Continuous-Time Nonlinear State-Space Model With a Discrete-Time Model"):

$$x(t+1) = f(x(t), u(t), \theta), \tag{20a}$$

$$\hat{y}(t\,|\,\theta) = h(x(t), \theta). \tag{20b}$$

If the state transition function f is not known, it could be parameterized with θ, that is, a flexible basis function expansion (S58). If a polynomial expansion (S60) is used, a PNLSS results, see also (S49). Similar expansions can be applied to h. In some cases, state x is measurable and h would be known. In such instances, [127] has applied a Gaussian process model to f, which corresponds to a basis expansion in terms of eigenfunctions associated with the kernel (the covariance function for the Gaussian process). See also Figure S26.

Nonlinear State-Space Models With Internal and External Dynamics

In "External or Internal Nonlinear Dynamics," the character of the nonlinear dynamics is discussed. This reflects whether the signals are fed around a nonlinearity or not. For a nonlinear state space, that property depends on the structure of f.

External Dynamics

If f is lower triangular,

$$x_j(t+1) = f(x_{1,\ldots,j-1}, u(t), \theta), \tag{21}$$

the states can be solved for explicitly from u and θ. It is then possible to write (20) in the form $\hat{y}(t \mid \theta) = h(u(t), \ldots, u(t-n_b))$, which is a system with external nonlinear dynamics (an NFIR model). Observe that in (21), the state $x_1(t)$ does not depend upon other states, only $u(t)$.

Internal Dynamics

If the nonlinear state space cannot be written in the lower triangular form (21), it is, in general, not possible to solve the equations explicitly as a function of $u(t)$, and the function (20) becomes a system with internal nonlinear dynamics. For these systems, there will be at least one state $x_i(t)$ that is a nonlinear function of the past values of some of the other states.

Two other aspects that should be carefully considered are stability and the impact of the initial states.

Unstable Models

Nonlinear state-space models with internal dynamics can become unstable. Even if the physical system is stable, the approximate model can become unstable during the optimization process. Using a modified formulation of the optimization problem, it is possible to address these unstable models [128]. The topic is also discussed in the section "Closed-Loop Data and Open-Loop Unstable Systems."

Unknown Initial States

The response of a nonlinear state-space model depends strongly on the initial states $x(0)$. These are usually unknown. The simplest solution is to omit the first part of the response. However, this is not affordable if the available record lengths are small compared to the transient time of the system. A better solution is to consider the initial states as additional parameters that are estimated simultaneously with the model parameters. For validation, it is important to realize that the initial state parameters need to be reestimated on the validation set to avoid the initial errors on the validation data dominating the errors.

Nonlinear Autoregressive Exogenous Models

A special case of (19) is the NARX model, where the state is constructed as a finite number of past inputs and outputs

$$x(t) = \varphi(t) = [y(t-1), \ldots, y(t-n_a), u(t-1), \ldots u(t-n_b)]^T \tag{22a}$$

$$\hat{y}(t \mid \theta) = h(\varphi(t), \theta). \tag{22b}$$

If h is a linear function, this predictor is the familiar simple ARX structure for a linear model. But, as indicated, a general nonlinear static function h can be expressed, for example, in the basis function expansion (S58). This structure is therefore known as a *NARX model*. If $n_a = 0$, it is an NFIR. NARX models are a common class of nonlinear models and can describe a large domain of nonlinear systems [19], [129]. However, they are not as general as the nonlinear state-space models discussed before. For example, the nonlinear system $y(t) = x(t)^2$, with $x(t) = G(q)u(t)$ and $G(q) = B(q)/A(q)$, cannot be represented in an input–output presentation (since the even-nonlinearity x^2 cannot be inverted).

NARX models come in many different shapes, depending on how h is parameterized (see "Static Nonlinearities"). They include Volterra systems (S68), neural networks (S64), Gaussian processes (Figure S26) [130], [131], and custom-made, semiphysical models (S59). The case where the nonlinear function h is written as a linear combination of known basis functions (S58), [11], [12] simplifies the identification problem to a linear regression. No iterative optimization procedure is needed [6], [30]. This is one reason why NARX models are quite popular and have been successfully applied to many industrial problems.

The number of terms M of a NARX model with basis expansion (S58) may grow very fast with the memory length. Special model-pruning methods have been developed to keep only the most dominant terms in the model [19]. More advanced methods to make a local variable selection are discussed in [132] and [133]. In these representations, a local model structure is determined in an arbitrary user-selected number of operating points, linking this approach also to the local model structures that were discussed before in the class of steel-gray models. More information about black-box models can be found in "Static Nonlinearities" and [134]–[141].

User Guideline

Lacking physical insights, it may be necessary to use black-box model structures. Many flexible and useful structures exist. However, they all have a strong *curve-fitting* flavor and may not recognize any intrinsic system features. They basically reflect the properties of the estimation data, which must be chosen with great care.

Pit-Black Models: Nonparametric Smoothing

So far, models have been described in parametric and analytic terms. However, there is also another possibility. A *pit-black model* takes a geometrical view of the observed data set and the model construction. This approach is outside the scope of the current survey. However, a few basic facts can be provided to make the model discussion more complete.

The model is a relation between the predictor and the output (that is, between all past observations and the observed next output). Denote by $\varphi(t)$ an n-dimensional vector of relevant representation of past observations [that is, the state in (19)]. Then, the model is a relation

$$y(t) = g(\varphi(t)) + e(t). \tag{23}$$

Here, g acts like the predictor in the previous sections, so it corresponds to the general model structure (9). However, instead of focusing on estimating g by some parameterization, the whole data record can be viewed geometrically. Assume that $y(t)$ is scalar. Then, each value pair $y(t), \varphi(t)$ is a point in an $(n+1)$-dimensional space Z. The data set is what might be thought of as a point cloud in this space. From that perspective, the modeling task is to find a surface g in Z that describes this cloud as accurately as possible (see Figure 7).

This can be accomplished by various ways to smooth the raw data cloud. The basic assumption is that the model surface is smooth, that is, $g(\varphi_1)$ and $g(\varphi_2)$ are close if φ_1 and φ_2 are. This means that, if the observed points $[y_i, \varphi_i]$, $i = 1, \ldots, N$ are available, an estimate $\hat{g}(\varphi_*)$ can be constructed for any point φ_* from observed y_i at neighboring points:

$$\hat{g}(\varphi_*) = \sum_{i=1}^{N} y_i w(|\varphi_* - \varphi_i|), \tag{24}$$

where the kernel w weights the value of the observation y by its distance to the sought regressor point φ_*. Such smoothing kernels and nonparametric estimates $\hat{g}(\varphi_*)$ are discussed extensively in the statistical literature [142], [143]. In the control literature, they have been considered under the name *just-in-time models*, since the estimate at point φ_* is constructed from raw measured data only when it is requested.

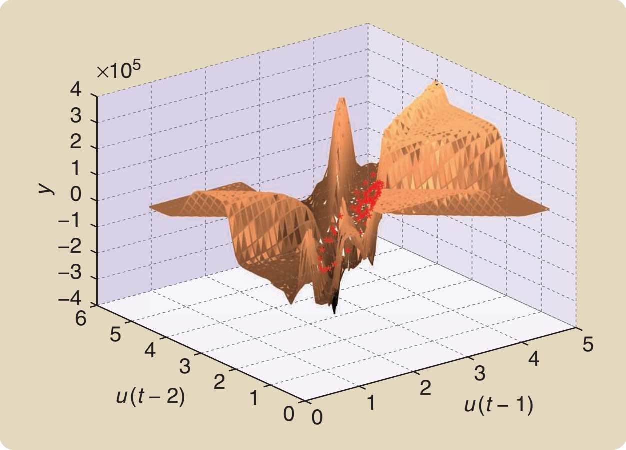


FIGURE 7 The cloud of observed data (red dots) and the model as a surface in the data space.

A very simple and common approach is to let w be zero, except for the $\varphi(t)$ that is closest to φ_* in the database. That makes the estimate $\hat{g}$ equal to its nearest neighbor. Many other kernels have been suggested and studied. A commonly used one is a parabola bottom, turned upside down, known the *Epanechikov kernel* [144],

$$w(x) = C(1 - x^2)_+ \quad x = \| \varphi_* - \varphi(t) \|, \tag{25}$$

where $(\cdot)_+ = \max[\cdot, 0]$, and C is a normalization constant. Another approach to selecting the weights w in (24) is *direct weight optimization* [145]. Here, w is selected so that an upper bound of the error in the estimate $\hat{g}(\varphi_*) - g(\varphi_*)$ is minimized using a quadratic programming technique.

Manifold Learning

An additional strategy for this geometrical construction of linear models is to find lower-dimensional manifolds in the cloud where there is a clear concentration of points. A linear model version of this idea can be used for illustration. In the data cloud setup, a model is a hyperplane in the data space. Many techniques are well known to fit the hyperplane to data, including principal component analysis, for finding essential linear subspace descriptions. This means that the modeling can be concentrated to the selected subspace with lower-dimensional models. The nonlinear counterpart is to find a lower-dimensional manifold $\Upsilon = r(Z)$ and express a model in terms of the lower-dimensional image of the variables φ under this mapping. Finding such a manifold is a challenging problem that has been discussed in an extensive literature under *manifold learning*. See [146] (Isomap) and [147] (local linear embedding).

User Guideline

In summary, pit-black models constructing model predictors directly from data (not explicitly employing a parameterized model) are an interesting option for nonlinear identification that has not been used that much in the control community.

EXPERIMENT DESIGN

The experimental data are the fundamental information source for the data-driven modeling process. Practical concerns (easy access to a nonparametric noise and nonlinear distortion analysis to guide the model selection process) and theoretical issues (maximizing the information in the data with respect to the selected model structure) should be addressed during the design of the experiment. This leads directly to the following guidelines:

- *Practical concerns*: Use periodic excitations whenever possible because these give direct access to a nonparametric distortion analysis without any user interaction (see "Nonparametric Noise and Distortion Analysis Using Periodic Excitations").
- *Theoretical concerns*: Design the amplitude distribution and power spectrum of the excitation to maximize the information with respect to the parameters that must be estimated in the selected model structure. Note that this is still no guarantee that the full domain of interest is covered (see, for example, Figure 8).
- *Warning*: Because structural model errors often dominate the noise-induced errors, it is necessary to select excitation signals that reflect the later use of the model to keep the structural model errors small in the domain of interest (see "Impact of Structural Model Errors").

Design of Periodic Excitation Signals

A simple periodic excitation is $u(t) = U_1 \cos(2\pi f_0 t)$. The period of this signal is $T = 1/f_0$. In most experiments, $u(t)$ is generated and processed in discrete time $t = lT_s$, with $l = 1, \ldots, N$ and $T_s = 1/f_s$ the inverse sample frequency. The sample frequency and period length are matched to each other by choosing $T = NT_s$, so that N samples fit exactly in one period of the signal. This relates also the frequency f_0 to the sample frequency f_s by $f_0 = f_s/N$.

More general periodic signals are represented by their Fourier series as the sum of harmonically related sines and cosines, with frequencies at integer multiples of f_0:

$$u(t) = \sum_{k=1}^{F} U_k \cos(2\pi k f_0 t + \varphi_k). \tag{26}$$

Such a signal is called a *multisine* [148], and it has a period $T = 1/f_0$ and a frequency resolution $f_0 = f_s/N = 1/T$. The amplitude spectrum U_k, phases φ_k, the number of excited frequencies F, and the frequency resolution f_0 are user choices that define the periodic signal. These can be set by the following (see also "Nonparametric Noise and Distortion Analysis Using Periodic Excitations").

User Guidelines to Design a Multisine [30], [149], [150]

- *Spectral resolution* $f_0 = f_s/N$: This should be chosen high enough so that no sharp resonances are missed [151].
- *Period length* N: This is set by $T = 1/f_0 = N/f_s$. A higher-frequency resolution requires a longer measurement time.
- *Amplitude spectrum* U_k, $k = 1, \ldots, F$: This should be chosen such that the frequency band of interest is covered.
- *Phases*: Use random phases that are mutually independent for $k \neq l$, and $E\{e^{j\varphi_k}\} = 0$. For example, select φ_k from a uniform distribution on the interval $[0, 2\pi)$. For this choice, it follows from the central limit theorem that $u(t)$ is asymptotically Gaussian distributed for $F \to \infty$ [30].
- *Signal amplitude*: This should be scaled such that it also covers the input amplitude range of interest.
- *Length of the experiment*: At least one and preferably a few periods (for example, three) should be measured.

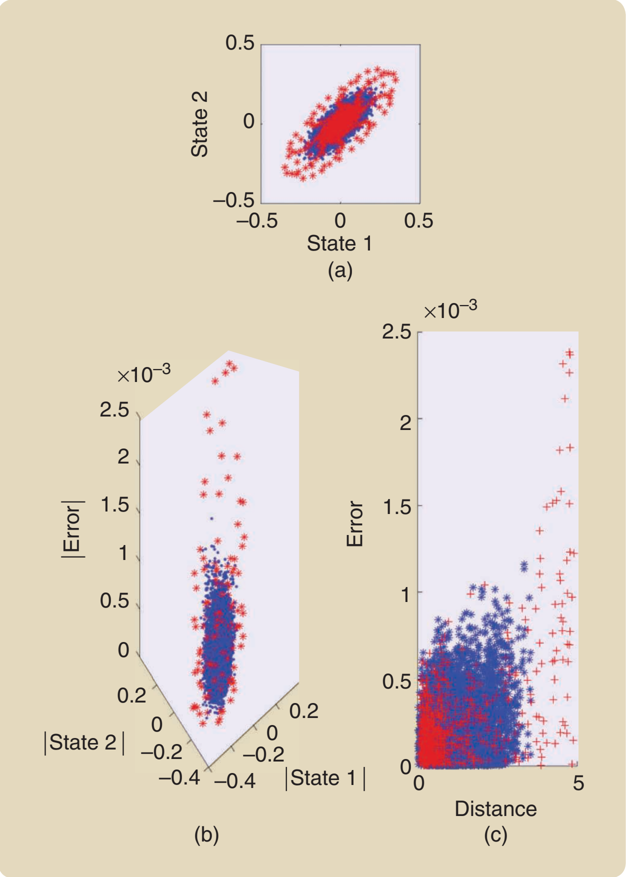


FIGURE 8 The model validation on the forced Duffing oscillator. A nonlinear state-space model is identified (see Figure S22) using a part of the tail data described in "Extensive Case Study: The Forced Duffing Oscillator." Next, the models are validated on another data set from the tail part (blue) and the sweeping part (red). The input signals in both data sets have almost the same maximum amplitude and power spectrum, as shown in Figure S19. (a) A plot of the state trajectory for both validation experiments. Observe that the domain covered by the sweeping signal is much larger than that of the tail signal. (b) The absolute values of the simulation error for both experiments are plotted as a function of the states. The errors on the states outside the domain covered by the tail signal are much larger. This is clearly visible in (c), where the absolute error is plotted as a function of the normalized distance $s^t C_s^{-1} s$ with s the states and C_s the covariance matrix of the tail (blue) states. Observe that the red errors in (b) and (c) for the large values of the distance are much larger than the blue ones. This shows that a nonlinear model that is validated only on the blue data fails to explain the red data that cover a larger domain, illustrating the risk of extrapolation in nonlinear system identification.

- *Experimental repetition*: If possible, repeat the experiment with a new realization of the random-phase multisine, and average the results over the multiple realizations for improved estimates of the nonlinear distortion levels. The additional data sets are also valuable for model validation on different but very similar excitations and the generation of more reliable uncertainty bounds in the presence of structural model errors (see also "Impact of Structural Model Errors").

Note that, by optimizing the phases, it is possible to create randomized signals with a user-controlled amplitude distribution and power spectrum [152]. For example, signals with a uniform amplitude distribution that excite a specified frequency band can be generated.

Experiment Design: Most Informative Experiment

The goal of the experiment design can be formalized as a procedure to obtain the minimum required information needed to reach the modeling goals at the lowest experimental cost (time, power consumption, and disturbance of the process). Originally, optimal experiment design was completely focused on maximizing the information content of the experiment [153], quantified by the Fisher information matrix M [6], [7], [30] that is directly linked to the smallest possible covariance matrix of the parameter estimates $P_\theta = M^{-1}$. A scalar measure [for example, the determinant $\det(M)$ or the trace $\mathrm{tr}(M)$] is used to quantify the information in the experiment.

Although an optimal experiment design is not always created for each experiment, it is quite useful to complete the exercise on a number of typical problems to be studied because it provides intuitive insight into what makes a good experiment. On the basis of that experience, the quality of the experiments can be significantly boosted, even without explicitly designing an optimal input.

For models that are linear in the parameters, the information matrix M does not depend on the actual parameter variables. This does not hold true if the output of the model is a nonlinear function of the parameters. In that case, $M(\theta_0)$ depends explicitly on the true but unknown parameters θ_0. Often, the true parameters θ_0 are replaced by an estimate $\hat{\theta}$ obtained from an initial experiment.

The more structured a model is, the higher the gain that can be obtained by a specific optimally designed experiment. The optimal experiment for a first-order linear system, described by its nonparametric impulse response representation $g(k), k = 1, \ldots, n$, is a white-noise excitation. If the same system is represented by its transfer function model $G(s) = (1/(1+\tau s))$, the optimal experiment is a single sine excitation at a frequency $f = 1/\tau$ [154].

Optimal Input Design for Linear Systems

The optimal input design for linear systems is now fully understood. In the basic problem, an excitation is designed that maximizes the determinant of the information matrix $\det(M)$. The optimal design minimizes the normalized variance of the estimated transfer function and is retrieved by solving a convex optimization problem. Since the problem depends only on the second-order properties of the input signal, its solution is given by an optimal power spectrum of the excitation signal [153]–[155]. The actual shape of the signal (the amplitude distribution) does not affect the information. However, practical constraints can have a strong impact on it, leading, for instance, to signals ranging from filtered white noise (having a Gaussian distribution) to binary excitations with a user-imposed power spectrum.

It became clear that this simple problem statement does not meet the full user needs. The optimal experiment design should be plant friendly (not all excitations are acceptable for the operators) [156]. Moreover, the uncertainty on the estimated model should be tuned to result in a control design that meets the global goals of the project. These initiated a search of application-oriented input design [76], [157]–[160], resulting in a design that pushes the uncertainty in a direction where it does not hurt the quality of the application. The uncertainty ellipsoid of the model is matched to the contour plots of the application cost.

Optimal Input Design for Nonlinear Systems

While the optimal input design for linear systems is well understood, it is much harder to provide general guidance on the optimal input design for nonlinear systems. Although it is possible to numerically retrieve the Fisher information matrix for nonlinear systems, it is difficult to interpret these equations and translate them into an optimal input [161]. This is because of the dependence of the Fisher information matrix M on the higher-order moments of the input for nonlinear systems. This leads to the design of the multivariate probability distribution of the input for all moments and all lags [161], [162], which is a highly nonconvex problem.

In the case of a static nonlinear system linear in the parameters, the problem becomes convex again. The solution depends only on the amplitude distribution of the excitation and results in a signal concentrated around a discrete set of excitation levels. The order of the samples does not influence the solution, so that the power spectrum is completely free.

The optimal input design for a nonparametric impulse response estimation (resulting in a white-noise excitation) can be generalized to nonlinear systems using the nonparametric Volterra representation. It leads to a design that combines the properties of the linear impulse response design (white-noise excitation) with that of optimal input design for static nonlinear systems (a discrete set of amplitude levels) [163]. This idea was also the starting point for a numerical design, leading to numerical procedures that perform a brute-force search for discrete-level signals [164], [165].

Solving the full optimal experiment design problem for nonlinear systems is tackled today using brute-force numerical optimization methods. The solutions can be generalized and simplified using a proper normalization of the problem. The choice of the signal constraint (the power constraint at the input or output and the amplitude constraint at the input, output, or an intermediate signal) is the most important. A natural choice is to restrict the amplitude range at the input of the nonlinear subsystem [166]. In some problems, the interest is in the identification of a single parameter in a nonlinear model. Its variance can be reduced using a well-designed feedback law that can be applied in real time [167].

Closed-Loop Data and Open-Loop Unstable Systems

To have a reasonable experiment, it may be preferable or necessary to generate the input by output feedback. Using closed-loop data for identification is not a problem [168]. The model convergence properties (as the number of observations tend to infinity) for prediction error methods are essentially the same for linear and nonlinear systems and for open- and closed-loop data [169, Lemma 4.1]: the estimated model will converge to that member in the model set that gives the smallest prediction error, under the conditions present during the experiment. The latter qualification may have unwanted effects, since the output may be partly explained via the regulator signals. A generic way to avoid that is to inject output-independent reference and excitation signals with the input. Closed-loop experiments should be specifically examined in this respect.

When an open-loop unstable system is identified, these issues become important. A common example is when the plant is an unstable aircraft [170]. The model is

$$x(t+1) = f(x(t), u(t), \theta), \tag{27}$$

$$y(t) = h(x(t)). \tag{28}$$

If the true value of θ corresponds to an unstable model, this expression cannot be used. In that case, an observer must be employed to create a stable prediction error:

$$\hat{x}(t+1) = f(\hat{x}(t), u(t), \theta) + K(t, \theta)\varepsilon(t, \theta), \tag{29}$$

$$\hat{y}(t \mid \theta) = h(\hat{x}(t)), \tag{30}$$

$$\varepsilon(t, \theta) = y(t) - \hat{y}(t \mid \theta). \tag{31}$$

Finding the optimal predictor is highly nontrivial and, in most cases, does not correspond to an additive term K. Two ad hoc solutions have been tested with reasonable success (for an aircraft application) in [170] and [171]: 1) compute K as an extended Kalman filter, and 2) parameterize $K(t,\theta)$ as a constant matrix K, and let it be part of the θ-vector. (This is known as the *parameterized observer approach*.)

User Guideline

In summary, an optimal input design based solely on the Fisher information matrix M and its related variance expressions should be applied only if there are no dominant structural model errors present. For linear systems, optimal input design is a well-developed field, and the nature of the solutions is fully understood, even if numerical procedures are needed to calculate the optimal input. Optimal input design is closely linked to the intended application. For nonlinear systems, numerical procedures are available for some nonlinear model structures. A full understanding of the important aspects of a good solution is still lacking.

MODEL VALIDATION

Model validation addresses the question "Does the model solve our problem?" and/or "Is it in conflict with either the data or prior knowledge?" A number of linear and nonlinear validation tools are discussed next.

Cross Validation

One of the most common and pragmatic tools for model validation is *cross validation*, which checks how well the model is able to reproduce the behavior of new data sets—*validation data*—that were not used to estimate the model. One way is to use the input of the validation data to produce a *simulated model output* $\hat{y}_s(t)$ and compare how well this model output reproduces the output $y(t)$ of the validation data. The comparison could simply be a subjective, ocular inspection of the plots to see if essential aspects of the system for the intended application are adequately reproduced.

The comparison can also be done by computing numerical measures of the fit between the two signals. These are naturally based on the distance between $y(t)$ and $\hat{y}_s(t)$. A common numerical measure is the *fit*, used in the System Identification Toolbox [172].

$$\text{fit} = 100\left(1 - \frac{\sqrt{\left\| \sum y(t) - \hat{y}_s(t) \right\|^2}}{\sqrt{\sum \left\| y(t) - \text{mean}(y(t)) \right\|^2}}\right)(\text{in }\%). \tag{32}$$

The fit determines the percentage of how much of the variation of the output is correctly reproduced by the model.

For models that contain integration or are used for control design, it may be more revealing to evaluate the model's prediction capability. The k-step-ahead predicted output for validation data $\hat{y}_p(t \mid t-k)$ is computed as described in (S20). Loosely, it means that $\hat{y}_p(t \mid t-k)$ is the model's prediction of $y(t)$ based on all relevant past inputs and all outputs up to time $t - k$. Compare this result with (S20). The prediction can then be compared with the measured validation output by inspecting the plots or by the fit criterion (32). In addition to these simple simulation and prediction applications, cross validation can be used in several sophisticated ways, which are discussed in the statistics literature [173].

Nonparametric Validation for Periodic Excitations

Using periodic excitations gives access to a nonparametric noise and distortion analysis (see, for example, Figure 4 in "Detection, Separation, and Characterization of the

Nonlinear Distortions and Disturbing Noise"). Adding the residues $y(t)-\hat{y}(t)$ to such a figure shows how well the model captured the nonlinear contributions (how far the validation errors are below the nonlinear distortion levels) and if the errors drop to the noise floor.

Actions

- *The error level is above the noise floor*: Structural model errors are detected. The user should decide if these errors are acceptable or not.
- *Nonparametric analysis of the errors*: If the special periodic excitations noted in "Nonparametric Noise and Distortion Analysis Using Periodic Excitations" are used in the validation test, it is possible to determine the nature of the dominant errors (even or odd nonlinearities are missed in the model), giving indications of how the model can be improved.

Linear Validation Tools

Linear validation tools check the whiteness of the residuals (autocorrelation test) and verify if no linear relations between the input and the residuals are left (cross correlation). These are both second-order moment tests that reveal only a subset of the possible problems (the higher-order moments are not tested). However, it still provides valuable information.

Actions

- *Cross correlation detected*: A more flexible linear part of the model can reduce the linear dependency at a low cost.
- *Autocorrelation detected*: The residuals are still colored. Using a linear noise model, it is possible to reduce the prediction errors. This can improve the efficiency of the estimation procedure.

Nonlinear Validation

Higher-Order Moment Tests

A full validation of a nonlinear model requires the higher-order moments to also be white: no higher-order cross correlations should exist. In practice, these tests are often not made for the following reasons:

1) Moments of order n are $(n-1)$-dimensional objects. Visually evaluating these at all lags becomes cumbersome and time consuming.
2) The required experiment length to estimate the higher-order moments with a given precision grows extremely fast with the order n, making such a test often unfeasible. For some dedicated problems, higher-order tests are proposed to detect the presence of nonlinearities [174].

Change the Nature of the Input

The behavior of a nonlinear system strongly depends on the nature of the excitation signal, even if the maximum input amplitude and power spectrum remain the same. For that reason, it is a strong requirement for a model to cover a wide range of input signals, which can be tested during the validation. In Figure S21, it is shown on the Duffing oscillator that the nonlinear NARX model that was tuned for the tail data failed to give a good simulation on the sweep data.

Action

Verify the quality of the model on all relevant classes of excitation signals.

Checking the Domain

Changing the nature of the excitation also changes the domain on which the internal nonlinear function (8) present in each nonlinear model is evaluated. The best guarantee to obtain a valid model is by ensuring that the complete domain of interest is covered during the estimation and validation. Although this is not always possible, it might be helpful to check the covered domain by plotting the phase plane trajectory for the estimated model for different excitations. This is illustrated on the Duffing oscillator example in Figures 8 and S23.

Check the Uncertainty Bounds

As explained in the discussion of uncertainty bounds in "Impact of Structural Model Errors" and illustrated in Figure S5, it is very hard to provide reliable uncertainty bounds in the presence of structural model errors that dominate the noise disturbances. The actual observed variability of the model is larger than the theoretically expected one. For that reason, it is indispensable to verify the validity of the theoretically calculated uncertainty bounds.

Action

Estimate the selected model structure with fixed complexity for different realizations of the excitation, and verify if the actual observed standard deviation agrees with the theoretical one.

User Guidelines

- Validating a model is a rather subjective and pragmatic problem. Check on a rich validation data set that covers the intended use of the model if the estimated model meets the user expectations.
- The final modeling goal should be kept in mind during the model validation. In many problems, structural model errors below a user-defined level can be tolerated, even if these errors are clearly detected in the model validation step.
- Check the theoretically obtained uncertainty bounds. In the presence of structural model errors, these may underestimate the actual variability.
- Make sure that the validation tests cover the full domain of interest.

EXAMPLES OF NONLINEAR SYSTEM IDENTIFICATION: FROM WHITE- TO BLACK-BOX MODELS

In the next series of examples, the use of different levels of physical insight into the nonlinear system identification process is illustrated.

Example 1: Off-White Model—Tank System

The System

The cascaded tanks system is a benchmark system that was set up at the University of Uppsala, Sweden [175]. It is a fluid-level control system consisting of two tanks with free outlets fed by a pump, described in detail in Figure 9.

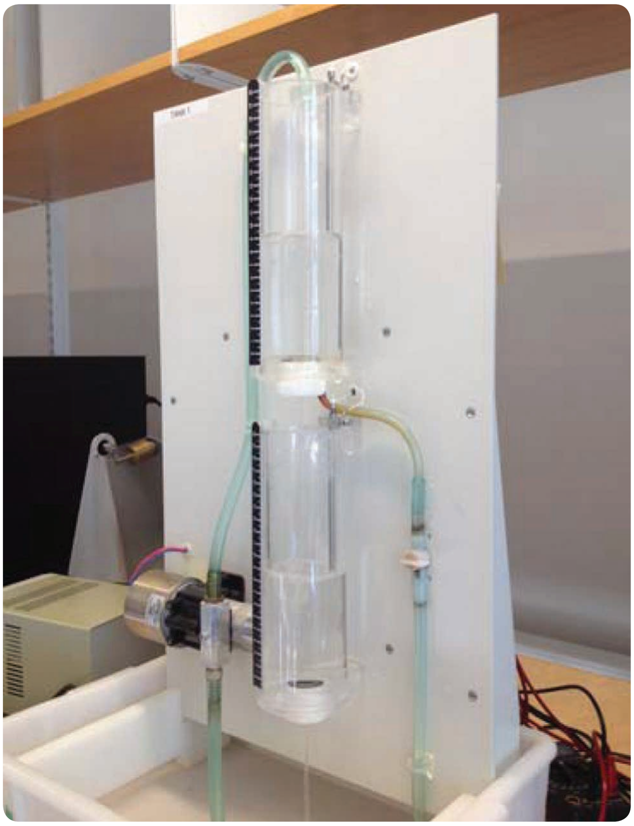


FIGURE 9 The cascaded tanks system [175]. The input signal controls a water pump that delivers the water from a reservoir into the upper water tank, where it flows through a small opening into the lower tank and finally through a small opening from the lower tank back into the reservoir. When the amplitude of the input signal is too large, an overflow can occur in the upper tank, with a delay also in the lower tank. When the upper tank overflows, part of the water moves into the lower tank, and the rest flows directly into the reservoir. This effect is partly stochastic. Hence, it acts as an input-dependent process noise source. The overflow saturation nonlinear behavior of the lower tank is clearly visible in the output signals that saturate at level 10 (see Figure 10). The input is the pump voltage, and the output is the water level of the lower tank.

A Physical Model

When no overflow occurs, a model can be constructed based on Bernoulli's principle and the conservation of mass:

$$\begin{aligned}\dot{x}_1(t) &= -k_1\sqrt{x_1(t)} + k_4 u(t) + w_1(t),\\ \dot{x}_2(t) &= k_2\sqrt{x_1(t)} - k_3\sqrt{x_2(t)} + w_2(t),\\ y(t) &= x_2(t) + e(t),\end{aligned} \tag{33}$$

where $u(t)$ is the input signal; $x_1(t)$ and $x_2(t)$ are the states of the system; $w_1(t), w_2(t)$, and $e(t)$ are additive noise sources; and $k_1, \ldots, k_4$ are constants depending on the system properties.

The relation between the water flowing from the upper tank to the lower tank and the water flowing from the lower tank into the reservoir are weakly nonlinear functions if there is no overflow (33), while in the presence of overflow, hard nonlinearities need to be identified. To model the overflow, $x_1(t)$ and $x_2(t)$ are constrained to their maximum value, and an additional term $w_3(t)$ is added to the second equation in (33) for $x_1(t) > x_{1\max}$ [176]. In [176], it is also proposed to add additional terms $k_5 x_1(t)$ to $\dot{x}_1(t)$ and $k_6 x_2(t)$ to $\dot{x}_2(t)$ to include the losses in the fluid flow. The losses are proportional to the velocity of the fluid squared and, therefore, proportional to the height of the fluid in each tank. Ultimately, this additional flexibility in the model is also used to accommodate other imperfections of the model, leading to an improvement that goes far below the expected impact of the loss terms in this system [176].

The Data

The input signals are ZOH-multisine signals that are 1024 points long and excite the frequency range from 0 to 0.0144 Hz, both for the estimation and test cases (see Figure 10). The lowest frequencies have a higher amplitude than the higher frequencies. The sample period T_s is 4 s, and the period length is 4096 s. Two similar data sets were collected, one for estimation and one for test (validation). The water level was measured using capacitive water-level sensors, and the measured output signals had a signal-to-noise ratio (SNR) close to 40 dB. The water-level sensors are considered part of the system. They are not calibrated and can introduce an extra source of nonlinear behavior.

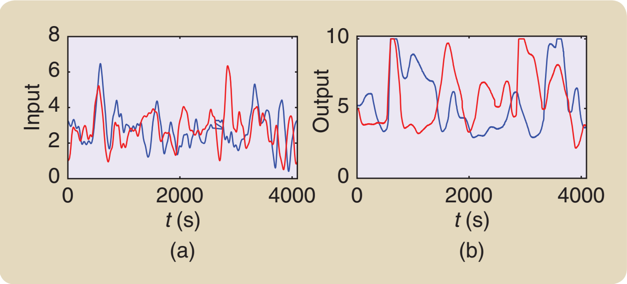


FIGURE 10 (a) Input and (b) output signals of the estimation data (blue) and test data (red) [175]. Observe the saturation in the output level (second tank) when the output $y(t) = 10$.

Note that the system was not in a steady state during the measurements. The system states have an unknown initial value at the start of the measurements. This unknown state is the same for both the estimation and the test data records, and it also must be estimated.

Cost Function

The cost function used to match the model to the data is

$$V = \frac{1}{N}\sum_{t=1}^{N}(y(t) - \hat{y}_{\mathrm{mod}}(t\,|\,\theta))^2, \tag{34}$$

with $y_{\mathrm{mod}}(t\,|\,\theta)$ a model output computed from (33) by ignoring the noises w and e.

Results

In [176], the parameters of the simple and extended physical model are directly estimated using well-selected numerical optimization procedures (see Figure 11). The fit (32) of the simulated output equals 95.77% on the estimation data and 94.07% on the test data. For the extended model, which also includes the loss terms, the fit increased to 98.98% on the estimation data and 98.22% on the test data.

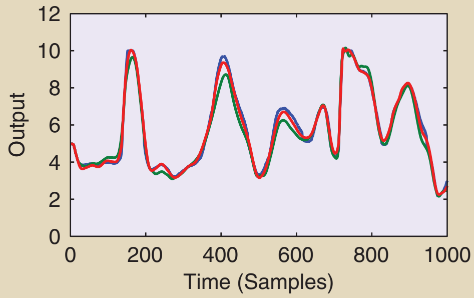


FIGURE 11 The cascaded tanks are modeled using the simple physical model (33) and the extended model (33) plus additional terms $k_5 x_1(t)$ to $\dot{x}_1(t)$ and $k_6 x_2(t)$ to $\dot{x}_2(t)$. Both models are validated on the test data [176]. The extended physical model (red) simulates the measured output (blue) much better (fit = 98.98%) than the simple model (33) (green) with fit = 94.07%. Observe that the saturations starting at $t = 150$ and $t = 750$ are well retrieved by the extended model.

As a reference, a black-box Gaussian-process NARX model (see the "Nonlinear Autoregressive Exogenous Models" section) with 15 lags for the input and output was estimated, resulting in fit = 95.38% for the simulation error and fit = 99.943% for the one-step-ahead prediction error. This illustrates once more that it is much easier to obtain a small prediction error than a small simulation error.

Example 2: Smoke-Gray Model (Semiphysical Model)—An Industrial Buffer Flow System

The System

This is an example of the usefulness of recalibration of the time scale [177], as explained in the section "Smoke-Gray Models: Semiphysical Modeling" (see Figure 5). The process is a buffer vessel in a pulp factory, in Skutskär, Sweden. The pulp spends 48 h in the different stages of the process—cooking, washing, and bleaching. It passes through several buffers to allow for a smooth, continuous treatment. It is important to know the residence time in the buffers for proper bookkeeping. The so-called κ number is a property of the pulp, measuring its lignin content. The buffer vessel is schematically depicted in Figure 5. The problem is to find a model for the dynamics of the buffer vessel that can be used to evaluate the residence time in the vessel and to, in a sense, time-mark the pulp as it passes through the several vessels.

The Data

In a particular buffer, the κ-number of the outflow (output y) was measured, along with the input u, the κ-number of the inflow (see Figure 12). The vessel level and flow were also measured (Figure 13).

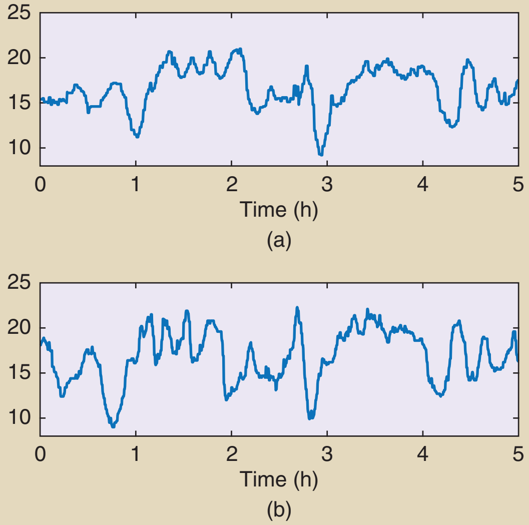


FIGURE 12 (a) The output κ-number y. (b) The input κ-number u.

First Model Attempt: Linear Model Based on Raw Data

The data were first detrended (by subtracting means from the inputs and outputs). A linear process model was then estimated with the input–output data u and y, using the first half of the data. That gives the model $G(s) = (0.818/(1+676s))e^{-480s}$. This model was simulated with the input, and the model output is compared with the measured output in Figure 14. This linear model is quite bad, as the simulated output differs quite substantially from the measured output.

Semiphysical Modeling Application

The physics behind the flow system must be considered. How does the flow and buffer level affect the buffer dynamics? If there is no mixing in the vessel (plug flow), the vessel is just a pure time delay for the pulp flow. The delay time is given by the vessel volume/pulp flow (dimension time).

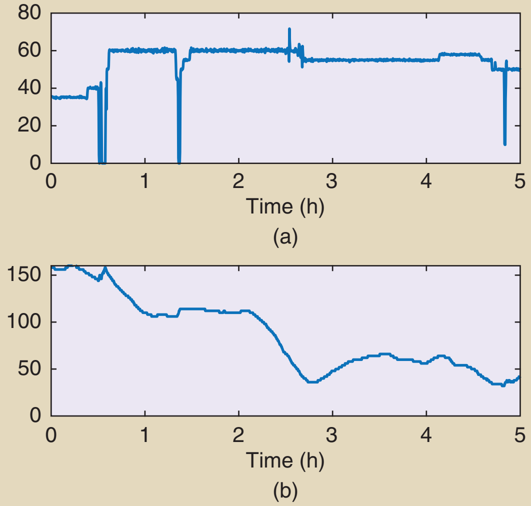


FIGURE 13 (a) The buffer flow. (b) The buffer level.

If there is perfect mixing in the tank, the system is a textbook first-order system with a gain of one and a time constant equal to the volume/flow. If volume and flow are changing, the system is nonlinear. The natural time variable is really volume/flow, which has been measured. The observed data can be resampled according to this natural time variable.

Time Scale Recalibration

A nonlinear transformation was applied to the raw data by resampling it to the natural time variable. The code in Matlab is shown in listing 1. The resampled inputs and outputs are indicated in Figure 15.

Second Model Attempt: Linear Model Based on Resampled Data

Building a linear process model from the first half of the resampled and detrended inputs and outputs gives the model $G(s) = (0.8116/(1+110.28s))e^{-369.58s}$. Simulating that model and comparing with the measured output (for resampled data) gives a much better fit, as shown in Figure 16, analogous to Figure 14. The semiphysical model is linear in the resampled data but nonlinear in the original raw data in Figures 12 and 13. It gives a sufficiently good description of the buffer to allow proper time-marking of the pulp before and after.

Example 3: Steel-Gray Model (Linearization-Based Model)—High-Pressure Fuel Supply System

The team of Oliver Nelles (University of Siegen, Germany) developed a local linear modeling-based approach and used it to identify a high-pressure fuel supply (HPFS) system used in a common-rail direct fuel injection for diesel engines. See [15], [178], and [179] for a detailed description of the project and the general methodology. This section is fully based on these references.

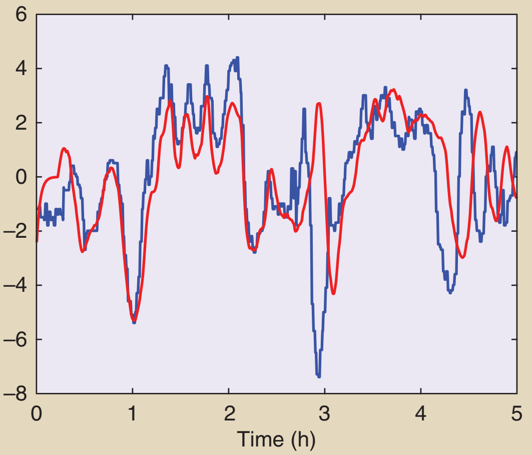


FIGURE 14 A linear model output (red) compared with the measured output (blue). The fit (32) for this model is 21.1%. The first half of the data was used for estimation.

Listing 1: The Matlab code for resampling data

```
z = [y,u];
pf = flow./level+sqrt(eps); %the eps-
     term to keep time running %even
     when there is no flow.
t = 1:length(z);
newt = interp1(cumsum(pf),t,[pf(1):
       sum(pf)]);
newz = interp1(t,z, newt);}
```

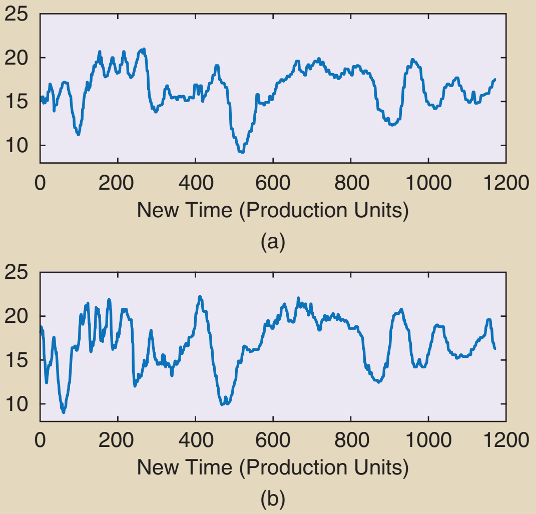


FIGURE 15 (a) The resampled output κ-number and (b) input κ-number.

The System

The main components of an HPFS system are the high-pressure rail, the high-pressure fuel pump, and the engine control unit (ECU). The pump is actuated by the crankshaft of the engine. A demand control valve in the pump controls the delivered volume per stroke. A pressure-relief valve is also included in the pump but should never open, if possible. Hence, the maximum pressure should be limited during the whole measurement process. The pump transports the fuel to the rail, which contains the pressure sensor. From there, it is injected into the combustion chambers. The system has three inputs and one output. The engine speed n_{mot} affects the number of strokes per minute of the pump and the engine's fuel consumption. The fuel pump actuation M_{SV} gives the fuel volume, which is transported with every stroke of the pump. It is applied by opening and closing the demand control valve accordingly during one stroke of the pump. The injection time t_{inj} is a variable calculated by the ECU, which sums the opening times of the single injectors and is, thus, related to the discharge of fuel from the rail.

During the measurement procedure, the injection time is not varied manually but set by the ECU. The permissible times depend on many factors, and a wrong choice could extinguish the combustion or even damage components. The engine load would have a major influence on the HPFS system via the injection time, but it is omitted to prevent the necessity of a vehicle test bench.

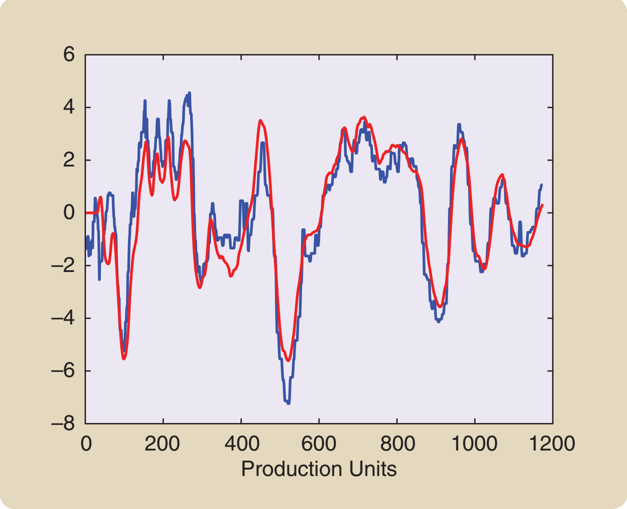


FIGURE 16 The simulated model output (red) and measured output (blue) for detrended resampled data. The fit (32) for this model is 60.4%. The first half of the data was used for estimation.

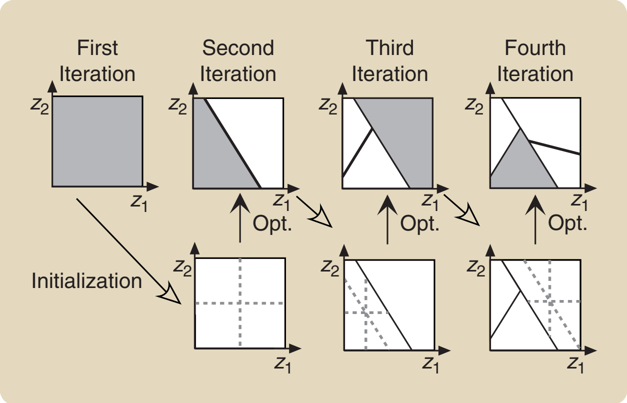


FIGURE 17 The first iterations of the hierarchical local model tree using a 2D *z*-input space [180].

The Model

A nonlinear model constructed using the delayed variables

$$u(t), \ldots, u(t-n_b), y(t-1), \ldots, y(t-n_a) \tag{35}$$

will be used to model the pressure $y(t)$ of the rail as a function of three inputs $u = [u_1, u_2, u_3]$, with $u_1(t) = n_{\mathrm{mot}}$, $u_2(t) = M_{\mathrm{SV}}$, and $u_3(t) = t_{\mathrm{inj}}$. The local linear modeling method (16) is selected to represent and identify the nonlinear model [15] (see also the section "Steel-Gray Models: Linearization-Based Models"). These are specified by the choice of 1) the regime points, 2) the validity functions, and 3) the local linear models.

The *regime points* $p(k)$ (also called *z-variables* in [178] and [179]), which are the entries of the validity functions $w(p(t), p_i)$ in (16), are reduced to one time-delayed process, with inputs and output in this application: $p(k)=[u_1(k-1), u_2(k-1), u_3(k-1), y(k-1)]$. This choice considers that the actual operating point is defined by the level of the actual process inputs and output. The first and higher derivatives of the model inputs and output are assumed to be insignificant to describe the operating point.

The *validity functions* $w(p(t), p_i)$ in this contribution are constructed using the hierarchical local model tree (HILOMOT) algorithm [180]. This incremental growing-tree construction algorithm divides the input space with axes-oblique splits. The validity functions are generated by sigmoid splitting functions that are linked in a hierarchical, multiplicative way. See [180] for more details.

The procedure of the HILOMOT algorithm can be explained with the help of Figure 17. Starting with a global affine model, in each iteration, an additional local affine model is generated. The local model with the worst local error measure (the gray areas in Figure 17) is split into two submodels, such that the spatial resolution is adjusted in an adaptive way.

The local linear models $\hat{y}_i(t\,|\,\theta, Z^{t-1})$ are ARX models of order three: $Z^{t-1} = [u_1(t-1), u_1(t-2), u_1(t-3), \ldots, u_3(t-1), u_3(t-2), u_3(t-3), y(t-1), y(t-2), y(t-3)]$.

The Data

Experiment Design: Two main aspects drive the experiment design. The experiments should be rich enough to obtain a good estimate of the local linear models. Even more important, the experiments should be designed such that the regime points cover the full space of interest [15], [178], [179]. The excitation signal should also comply with the constraints imposed by the process (that is, restrictions on the amplitude level and signal gradients). In [179], a procedure is described that searches for randomized steplike sequences that meet all of these constraints. The step time is set by the dominant time constant of the system, while the step sequences are designed to assure that the regime points are well distributed over the full operational space of the system. Detailed information, including the design of excitations for multiple input systems, is given in [179].

The Data: The optimized experiment is called *OptiMized Nonlinear InPUt Signal* (*OMNIPUS*), and it will be compared to a second experiment, in which the excitation is a combination of ramp and chirp sequences, proposed in [181]. The measurement time for each signal is limited to 10 min. The sampling frequency $f_s = 100\text{ Hz}$ results in a signal length of $N = 60{,}000$ samples.

Cost Function

The models are estimated by minimizing the squared errors (two) between the measured $y(t)$ and modeled output $\hat{y}(t\,|\,\theta)$. No weighting is applied.

Results

Two local linear model networks (LMNs) are identified, with the first being estimated on the OMNIPUS data and the second on the ramp–chirp data. Both models are used to simulate the measured plant output on a test data set that consists of a new realization of a ramp–chirp and OMNIPUS excitation. The simulated output of both models on the test data are shown and discussed in Figure 18. The error for

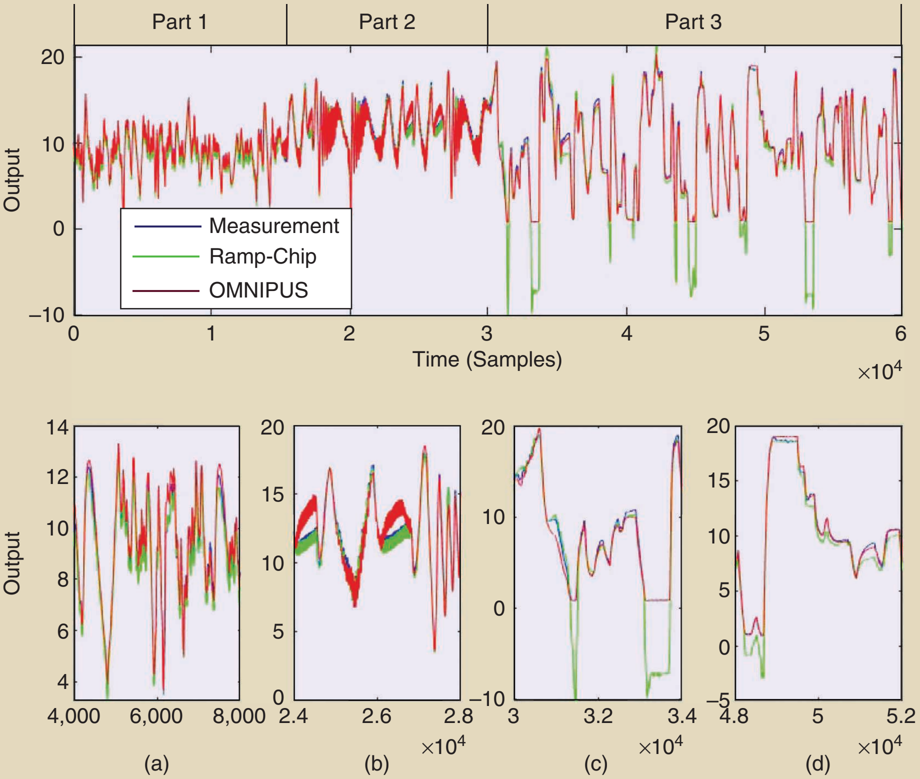


FIGURE 18 Two local linear model networks are identified. The first is estimated on an OptiMized Nonlinear InPUt Signal (OMNIPUS) excitation and the second on a ramp–chirp excitation. Both models are used to simulate the measured plant output on a test data set consisting of a new realization of a ramp–chirp and OMNIPUS excitation. The simulated output of the ramp–chirp (green) and the OMNIPUS hierarchical local model tree (red) model [178] on the test data are shown and compared with the measured output (blue) given as a reference. The test signal is designed such that the model behavior outside the training domain can be analyzed. Part 1 consists of ramp test sequences, part 2 of chirp test sequences, and part 3 of the OMNIPUS test sequence. The ramp–chirp model was trained on the ramp–chirp training data (similar to the signals in parts 1 and 2). Thus, the second half of the test signal (part 3) is outside the training domain of the ramp–chirp model. Alternately, a model was trained on the OMNIPUS training data (part 3). Thus, the first half of the test signal (parts 1 and 2) is outside the training domain. This effect can be seen in (b) and (c). In (a), good model fit for both models on the ramp–chirp sequence is obtained. In (b), a plant model mismatch of the model that was trained on the OMNIPUS data applied to the ramp–chirp sequence is visible, and in (c), the plant model mismatch of the model that was trained on the ramp–chirp data and applied to the OMNIPUS sequence can be observed. In (d), a good model fit for both models on the OMNIPUS sequence is obtained.

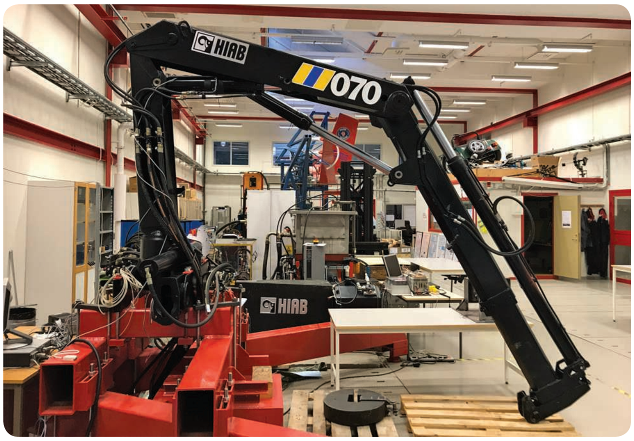


FIGURE 19 The tree-harvesting hydraulic crane.

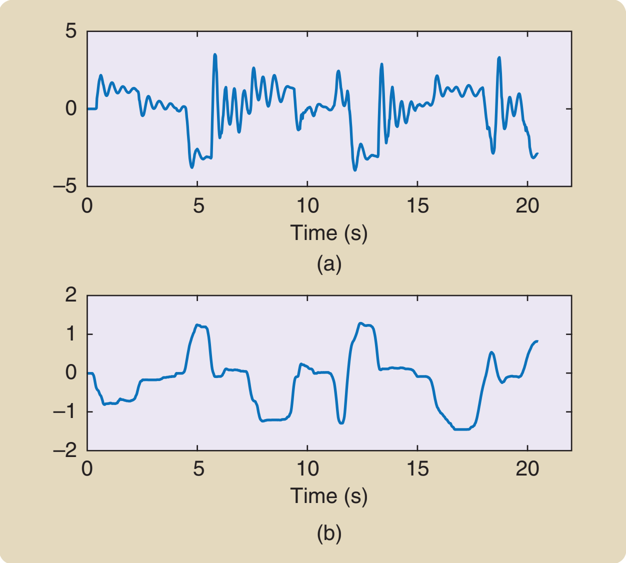


FIGURE 20 The data from the hydraulic crane. (a) Output (position of the tip). (b) Input (hydraulic pressure).

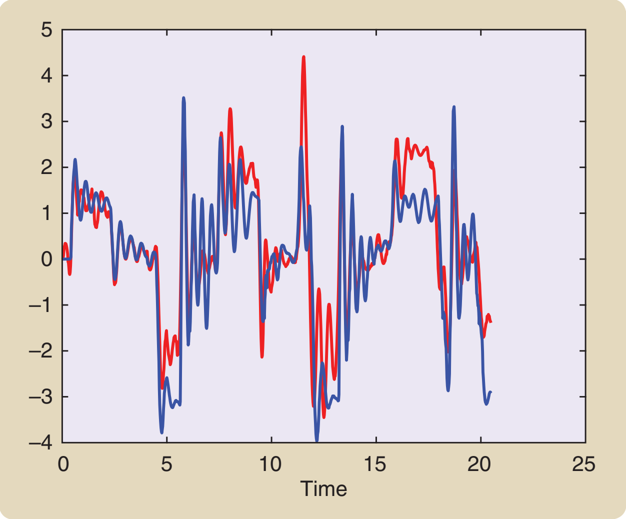


FIGURE 21 A comparison between the best linear model's simulated output (red) and the measured output (blue), with a fit of 42%. The first half of the data was used for estimation.

both local linear models is, in most cases, small [see Figure 18(a) and (b)]. However, there are also some significant mismatches [for example, Figure 18(c)]. The nonlinear behavior of the process is not well identified by the model identified on the ramp–chirp LMN. This major mismatch between process and model indicates a poor modeling, most likely because informative data in this area of operation are missing. Figure 18(b) shows a minor mismatch between the process and the OMNIPUS LMN.

Example 4: Slate-Gray Model (Block-Oriented Model)—Hydraulic Crane

The Process

When handling logs in tree harvesting, huge hydraulic cranes do all of the lifting and moving. The cranes can be thought of as industrial robots controlled by hydraulic pressure, and the controlled output is the position of the gripper at the top of the crane (see Figure 19). The crane shows oscillatory behavior at the gripper, and, to design a good regulator, a model must be developed.

The Data

Some collected data from a particular crane are shown in Figure 20 (see [182]). The dynamics are clearly resonant.

A Linear Model

To find a model, a linear model of the crane is first built using the data. The first half of the data sequence in Figure 20 was used to estimate the model, and the second half was used to evaluate the fit between the model's simulated output and the measured output. After some experimentation, the best model was obtained as a fifth-order linear state-space model. The comparison between model and measured outputs is shown in Figure 21. This best fit for a linear model (42%) is not very impressive and not good enough for control design.

Nonlinear Models: A Hammerstein Model

The lack of success with linear models may indicate that there are nonlinear effects in the system. A simple test to see if nonlinearities can improve the fit is to try a Hammerstein model (compare with Figure 6). A Hammerstein model with a fifth-order linear system—preceded by a nonlinear (piecewise linear) static nonlinearity estimated from the first half of the data record—gave the model fit depicted in Figure 22. The estimated nonlinearity at the input is shown in Figure 23.

The improvement from 42 to 72% fit with the input nonlinearity is quite impressive. In retrospect, a semiphysical explanation can be given. Most of the resonance dynamics is due to the mechanical construction of the crane. The transformation from the hydraulic input pressure to actual forces on the mechanical parts of the crane is more complicated, and, in that way, a model with an unknown nonlinear static transformation of the input acting on a linear system becomes physically feasible. Note that the estimated input nonlinearity is a saturation.

Example 5(a): Black-Box Volterra Model of the Brain

In collaboration with the Department of Biomechanical Engineering at the Delft University of Technology, The Netherlands, a regularized Volterra model has been identified for a part of the human sensorimotor system, as explained in Figure 24. Detailed information on this experiment is reported in [183] and [184].

The System

In the experiment, the relation between the wrist joint motion (input u) and the electroencephalography (EEG) signal measured at the sculp (output y) is modeled. The experiment setup used to obtain the EEG data evoked as a reaction to the wrist perturbation is depicted in Figure 24.

Volterra Models

Regularized Volterra models are a nonparametric representation of the system, using multidimensional impulse responses (S68) called Volterra kernels. Because the number of parameters grows extremely fast with the degree of the kernel, an additional constraint will be imposed on it to express that the kernel should be smooth in some directions, and it decays exponentially to zero along these directions. This is illustrated in Figure 25 for a second-degree kernel. By using constrained models, it is still possible to identify the kernels from relatively short data sets. This turned nonparametric Volterra modeling into a handy tool [56].

The choice of the maximum degree in (S68) is critical. Choosing it too high results in a fast-growing number of parameters to be estimated, and choosing it too low creates large structural model errors. A prior nonparametric analysis [79] showed that a linear model can capture only 10% [in terms of variance accounted for (VAF), corresponding to 5% in terms of fit (32)] of the characteristics of the wrist joint–brain system, while more than 70% VAF (or 45% fit) is attributed to even nonlinear behavior [184]. On the basis of these results, a second-degree Volterra model [see also (S68)] will be estimated:

$$\hat{y}(t\,|\,\theta)=\sum_{\alpha=0}^{2} y_0^{\alpha}(t), \tag{36}$$

including a dc offset ($\alpha=0$), a linear kernel ($\alpha=1$), and a quadratic kernel ($\alpha=2$). The output $\hat{y}(t)$ is calculated by direct evaluation of (S68).

The memory length of a Volterra model (S68) is set by the maximum length of the multidimensional kernels g_α, which is, in (S68), $m-1$. In this example, $m=33$ corresponds to a memory length of approximately 130 ms at a sampling rate of 256 Hz (see [163] and [184] for more details). This choice resulted in the smallest structural model errors.

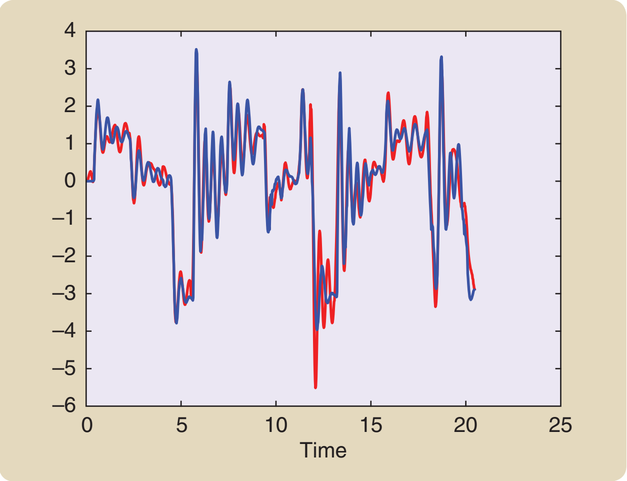


FIGURE 22 A comparison between the best linear model's simulated output (red) and the measured output (blue) for a Hammerstein model, with a fit of 72%. The first half of the data was used for estimation.

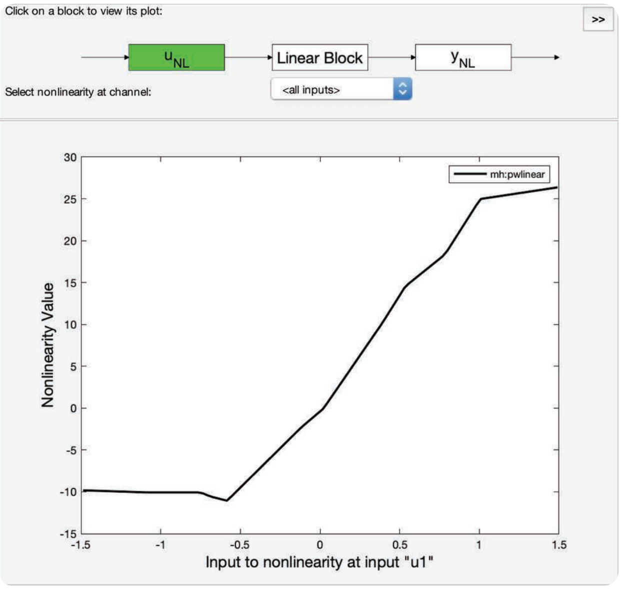


FIGURE 23 A screenshot of the estimated piecewise linear input nonlinearity (10 break points allowed).

The Data

Experiment Design: The perturbation signals used were random-phase multisines with a period of 1 s, resulting in a fundamental frequency $f_0 = 1$ Hz. Only odd harmonics of the fundamental frequency were excited, namely 1, 3, 5, 7, 9, 11, 13, 15, 19, and 23 Hz (odd random-phase multisines). Exciting the nonlinear system using different phase realizations of a multisine signal (that is, the same amplitude per frequency yet new random phases) allows for using different data for estimation and validation when modeling. Seven different multisine realizations were generated. After transient removal, 210 periods are available for each of the seven realizations. The aforementioned choices for the duration of the excitation resulted in a total duration of

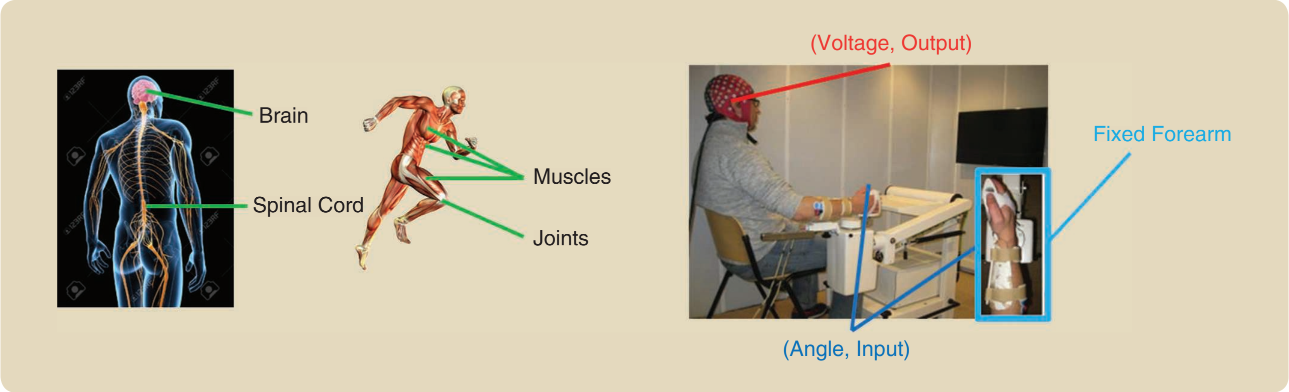


FIGURE 24 Modeling a sensorimotor system [183], [184]. (a) The sensorimotor system refers to all of the mechanisms in the human body that contribute to human motion. The brain, spinal cord, muscles, and joints between the bones constitute the basic parts of the sensorimotor system. In this experiment, the relation between the wrist joint motion and the electroencephalography (EEG) signals measured at a selected position on the skull is modeled. (b) The right forearm of the subject is strapped into an armrest and the right hand strapped to the handle, requiring no hand force to hold the handle. The imposed wrist motion includes circular motion of the wrist around a fixed reference (zero angle). The angle of the wrist constitutes the input to the system, while the EEG signal in the brain, as a reaction to this motion, is the system output. The measured output is contaminated with measurement noise, while the input signal is assumed to be exactly known.

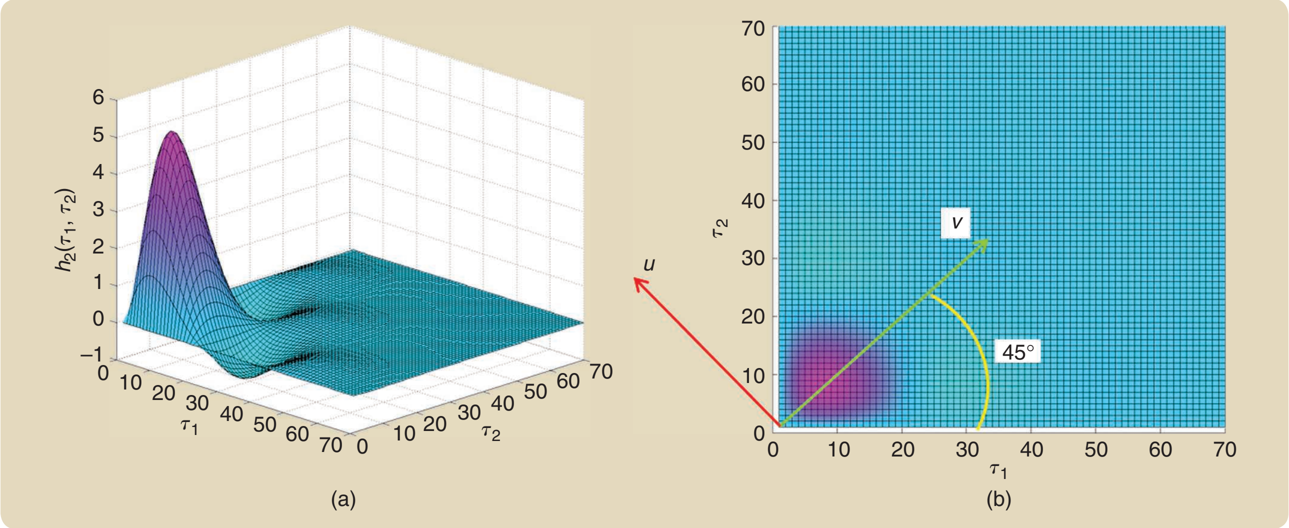


FIGURE 25 (a) The regularized estimation of a Volterra kernel [163], [185]. (b) An example of a second-degree Volterra kernel with imposed smoothness and exponential decay in the u and v directions. The smoothness and decay are set by hyperparameters tuned during the optimization step by adding a regularization term to the cost function. This approach is generalized to n-dimensional kernels, using n regularization directions. (Source: Birpoutsoukis [163]; used with permission.)

clear perturbation equal to 24.5 min per subject. Including the extra time intervals corresponding to the preparation of the equipment, preparation of the subject, and the pauses necessary for the safety and convenience of the subject, the total experiment time results to be more than 2 h!

Preprocessing of the Data: The EEG measured signals were high-pass filtered with a cutoff frequency of 1 Hz. The 50-Hz disturbance from the mains was removed. In a second step, the data were averaged over the periods, resulting in an SNR of approximately 20 dB. For all of the subjects, the recorded output signals were shifted in time to impose a time delay of 20 ms, corresponding to the traveling time of the signals from the wrist to the brain. As long as this delay is not too large, its choice is not critical because the Volterra kernels can also accommodate delays, at a cost of a growing memory length of the kernels. This results in a fast increase of the number of parameters to be estimated (the values of the Volterra kernels).

Cost Function

The model parameters are estimated by minimizing a regularized least-squares cost function [56] using a Bayesian perspective [185] consisting of the sum of the squared differences between the averaged measured output and the modeled output $\hat{y}(t \mid \theta)$ (36), plus a regularization term [see also (S43)]:

$$V = \frac{1}{N}\sum_{t=1}^{N}(y(t)-\hat{y}(t\,|\,\theta))^2 + [\theta_1^T \;\; \theta_2^T]\begin{bmatrix}P_1^{-1} & 0\\ 0 & P_2^{-1}\end{bmatrix}\begin{bmatrix}\theta_1\\ \theta_1\end{bmatrix}. \quad (37)$$

The parameters $\theta = [\theta_0 \;\; \theta_1^T \;\; \theta_2^T]$, with $\theta_i \in \mathbb{R}^{n_{\theta_i}}$, are the parameters of the Volterra kernel of degree i. For $m = 33$, $n_{\theta_1} = 33$ for the linear kernel g_1, and $n_{\theta_2} = m(m+1)/2 = 561$ for a symmetric quadratic kernel g_2 [28]. The regularization matrix is block diagonal in which each block P_i accounts for the regularization of the Volterra kernel of degree i. Because of this choice and the selection of an odd excitation (only odd-frequency components are present in the multisine), the identification of the linear part is completely decoupled from the even nonlinear part (the zero- and second-degree kernels). The hyperparameters in the regularization matrices P_i are tuned using a marginalized maximum likelihood estimator [185], which leads to a nonlinear numerical optimization in the hyperparameters (the model parameters $[\theta_0, \theta_1^T, \theta_2^T]$ are eliminated in the marginalizing step). Once the hyperparameters are fixed, the remaining problem is linear in the model parameters $[\theta_0, \theta_1^T, \theta_2^T]$ and directly worked out by solving the linear least-squares problem (37).

Results

The results are shown in Figure 26. Only the second-degree kernel is depicted. The linear part of the model captured approximately 10% of the power of the output signal y (formally defined as VAF in [184]), resulting in a 5% fit, while more than 70% of the output power was attributed to even nonlinear behavior in the nonparametric analysis [184], leading to a 45% fit. The Volterra series, combined with the regularization technique described previously, was used to model the nonlinear system behavior.

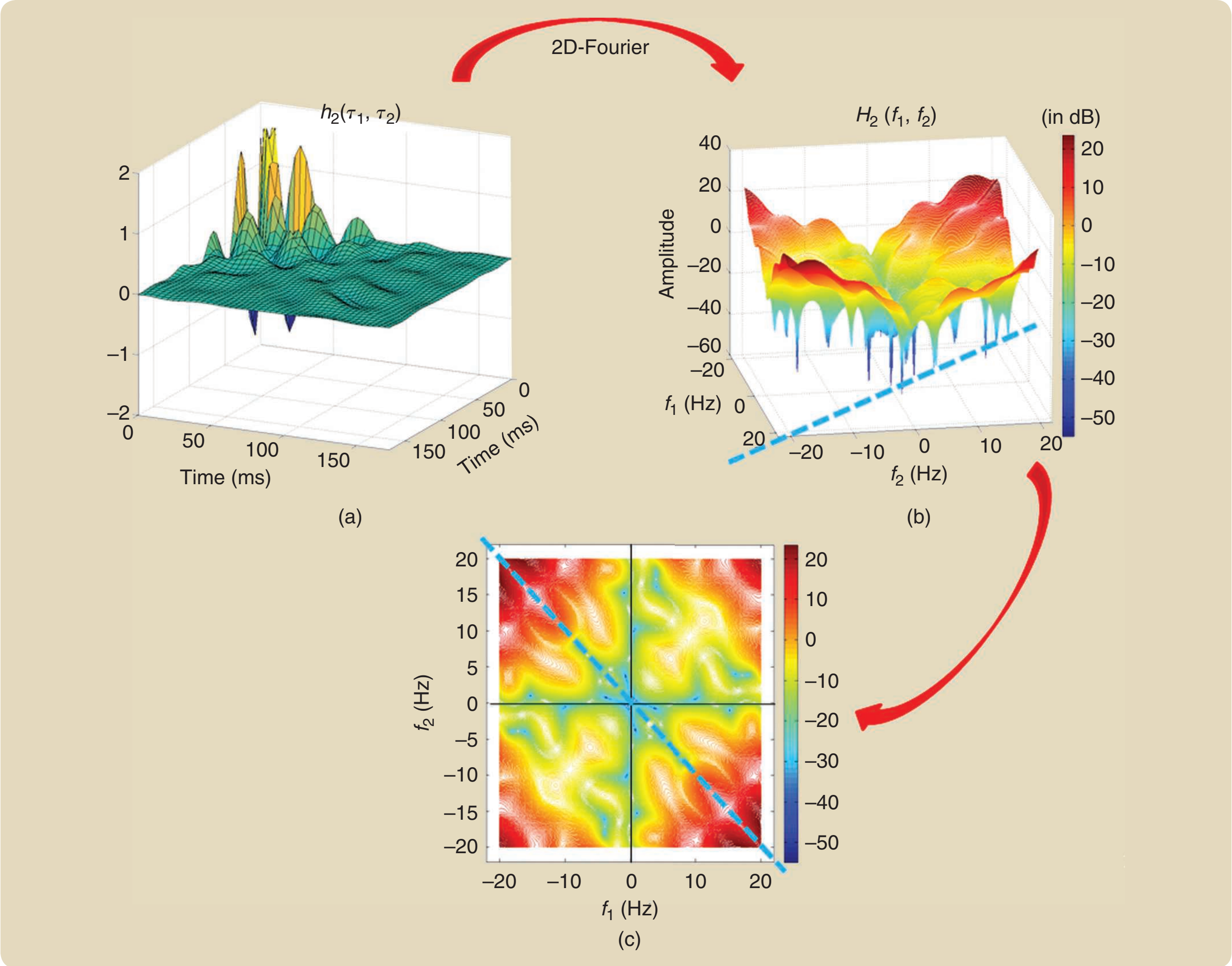


FIGURE 26 The estimation of the wrist angle in the electroencephalography system [163]. (a) The estimated second-order Volterra kernel. (b) A 2D frequency response function (FRF) obtained by applying the 2D Fourier transform to the 2D impulse response (Volterra kernel). (c) An *xy* view of the 2D FRF. From these results, it can be concluded that the system behaves as a high-pass system (the lower frequencies are strongly attenuated) that transfers the intermediate high-frequency power, in the upper-left and bottom-right corners of the *xy* view, down to the low frequencies at the output. (Source: Birpoutsoukis [163]; used with permission.)

The selection of a model set for nonlinear identification is a major problem and offers a very rich range of possibilities.

The obtained models were able to achieve, on average, a 46% VAF, corresponding to a 26% fit on validation data sets across different participants. Since a 10% VAF of the output is due to noise, it indicates that approximately a 44% VAF remains unmodeled. In this case, a fourth-order kernel would be needed to further improve the results. However, even with regularization, this remains an unattractive problem. Richer and longer experiments would be needed, and that is not feasible. Although the second-degree Volterra model still suffers from large structural model errors, it provides already useful insights into the wrist–brain system.

In [163], a new experiment was made using a richer excitation, so that the risk for overfitting is further reduced. Instead of exciting only the odd frequencies, the even frequencies were also excited. At that moment, the estimation of the linear part can no longer be decoupled from the nonlinear part of the model. With this rich data set, the VAF increases to almost 60% (or a 36% fit). However, the qualitative conclusions remained the same.

Example 5(b): Nonlinear State-Space Black-Box Model of a Lithium-Ion Battery

Lithium-ion batteries are attracting significant and growing interest because their high energy and high power density render them an excellent option for energy storage, particularly in hybrid and electric vehicles. In this section, a nonlinear state-space model is proposed for the operating points at the cusp of the linear and nonlinear regimes of the battery's short-term electrical operation. This point is selected on the basis of a nonparametric distortion analysis as a function of the state of charge (SOC) and temperature. More detailed information is available in [186] for a fixed SOC (10%) and temperature model (25 °C). In [187], a nonlinear state-space model is developed that covers a varying temperature (from 5 to 40 °C) and SOC (2–10%). For higher SOC values, it follows from the nonparametric distortion analysis that a linear model can be used.

The System and the Experimental Setup

A high energy density lithium-ion polymer battery (EIG-ePLB-C020, lithium nickel cobalt manganese) with a nominal voltage of 3.65 V, nominal capacity of 20 Ah, and ac impedance (1 kHz) <3 mΩ (along with the PEC battery tester SBT0550, with 24 channels) is used for the data acquisition. The tests are performed on a preconditioned battery inside a temperature-controlled chamber at 25 °C [186], [187].

Nonlinear State-Space Model

A discrete-time nonlinear state-space model (S78) is selected to approximate the continuous-time system (see "Approximating a Continuous-Time Nonlinear State-Space Model With a Discrete-Time Model"). The nonlinear state space model in (19) is rewritten as

$$\begin{aligned} x(k+1) &= \tilde{F}(x(k), u(k)) = Ax(k) + Bu(k) + F(x(k), u(k)), \\ y(k) &= \tilde{G}(x(k), u(t)) = Cx(k) + Du(k) + G(x(k), u(t)). \end{aligned} \tag{38}$$

Although the split between the linear part and the nonlinear terms $F(x(k), u(k))$ and $G(x(k), u(t))$ in (38) is not unique, it is convenient to write the equations like that because the initialization procedure that will be presented further on provides initial estimates for $A, B, C,$ and D that are obtained from the BLA of the nonlinear system.

The nonlinear terms $F(x(k), u(k))$ and $G(x(k), u(t))$ are multivariate nonlinear functions. These will be written as a linear combination of nonlinear basis functions. The whole palette of possibilities can be used, ranging from polynomials and hinge functions to Gaussian processes. In this example, a polynomial representation will be used.

The Data

An odd random-phase multisine signal is used as an input excitation signal. The band of excitation is kept between 1 and 5 Hz because the dynamic range of interest of the battery for hybrid and electric vehicle applications is covered well within this band of excitation. It also takes into consideration the limitations of the battery tester in terms of the sampling frequency. The excitation signal has a period of 5000 samples, and the sample frequency f_s is set to 50 Hz, resulting in a frequency resolution of $f_0 = 50/5000 = 0.01$ Hz. The input is zero mean with a root-mean-square value of 20 A. A detailed description of the whole measurement procedure (for example, charging and discharging) is given in [186].

Cost Function

The cost function is formulated in the frequency domain on the difference between $Y(k)$ and $\hat{Y}(k,\theta)$, which are the discrete Fourier coefficients of the measured and modeled output,

$$V = \frac{1}{F}\sum_{k\in B}^{N} \frac{|Y(k) - \hat{Y}(k|\theta)|^2}{W(k)}, \tag{39}$$

Any model will be an approximation of the truth, and it will be affected by the aspects of the system that are excited during the experiment.

where B is the set of frequencies of interest and $W(k)$ a user-selected weighting. Because structural model errors dominate, the weighting $W(k)$ is set to one (see "Impact of Structural Model Errors").

Results

Nonparametric Distortion Analysis: At the start of the identification process, a nonlinear distortion analysis (see "Nonparametric Noise and Distortion Analysis Using Periodic Excitations") provides insight into the quality of the data and the possible gain that can be made with a nonlinear model. In this case, Figure 27 shows that the SNR of the data is very high (above 50 dB). The nonlinear distortions are very low for a high SOC, while they increase above 10% for a low SOC (10%). More results for other temperatures, SOCs, and amplitude ranges are available in [187]. On the basis of these results, it was decided to focus the nonlinear modeling effort on the low SOC range.

Parametric Nonlinear State-Space Model: The nonlinear state-space model is determined in two steps [188]. In the initialization step, the BLA (S30) is ascertained using one of the classical linear identification methods. The noise weighting $W(k)$ in (39) during this step is set by the nonlinear distortions. These results are used to find initial values for the A, B, C, and D matrices. In the second step, the nonlinear terms are added to the model. These are linear in the parameters and initialized at zero. Eventually, a nonlinear numerical optimization is used to minimize the cost function with respect to all of the parameters (see [186] for more details). In this and the next steps, $W(k) = 1$ because model errors dominate.

In this case, a polynomial nonlinear state-space model (38) is used with three internal states, and F is a multivariate polynomial of degree three. A major problem that often shows up in many applications is the appearance of instabilities. Even if the identified model remained stable on the test data, the model still frequently becomes unstable on the validation data. This is mainly due to the polynomial basis functions that were used in this example. Outside the state-space domain covered by the test data, the polynomials have the tendency to grow very fast, resulting in model instabilities. The results are discussed in Figure 28. The nonlinear model outperforms the linear one by a factor of 10 on the validation data. However, the errors are still far above the noise floor.

For nonlinear state-space models that include output feedback, it is also possible to replace the states $x(k)$ by a well-selected set of outputs $y(k)$ as the input to the multivariate nonlinear functions, leading back to the full expression (19). Although such a representation can always be reduced to that in (38), it can be much more compact. A typical application is in vibrating mechanical systems, where the nonlinearities often depend on the displacement of the structure close to the nonlinearities, which can be selected to be the states of the nonlinear equations [24]. This choice also simplifies the initialization of the identification problem because, in these applications, the states are directly measurable outputs.

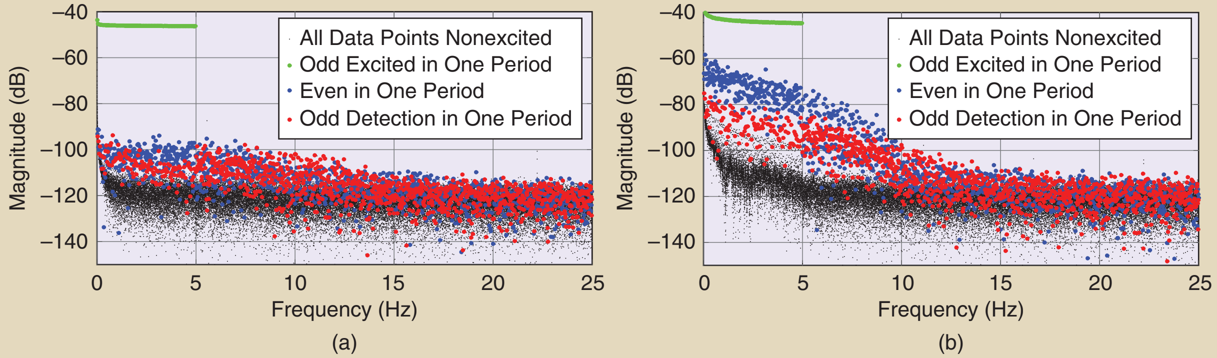


FIGURE 27 (a) A nonlinear distortion analysis [186], [187] for a state of charge (SOC) of 90% and (b) an SOC of 10% at a temperature of 25 °C. The signal-to-noise ratio of the measurements is very good. The output at the excited frequencies (green) is more than 60 dB above the noise floor (black). At a 90% SOC, the nonlinear distortions are at −50 dB (less than 1%). At a 10% SOC, the even nonlinear distortions (blue) are well above the odd distortions (red) at 10% (20 dB) of the output. (Source: Relan [187]; used with permission.)

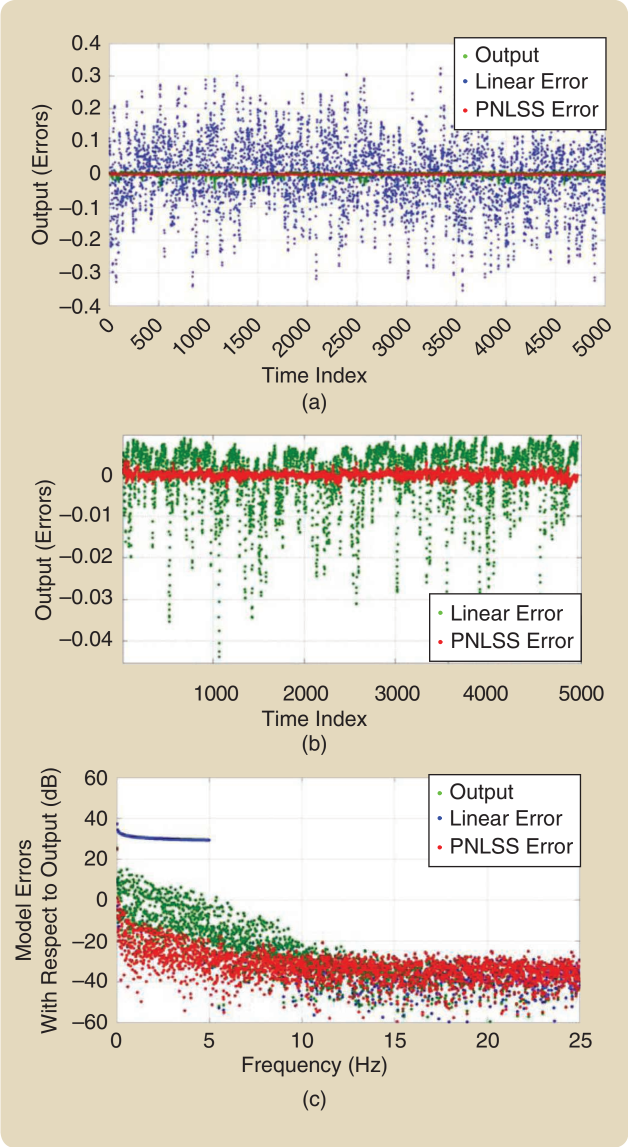


FIGURE 28 The nonlinear simulation of a battery [186], [187]. (a) The simulation error of the best linear (green) and nonlinear (red) state-space models on a validation set. (b) A zoom of the results in (a). Observe that the error is strongly asymmetrically distributed, which is well in agreement with the presence of dominating even nonlinearities. This results in large spikes for the linear model that are completely removed by the nonlinear model. (c) The same results are shown in the frequency domain. Although the errors of the nonlinear state space model are 20 dB below those of the linear model, they are still far above the noise floor that was measured in the nonparametric noise analysis (see Figure 27). In this case, the structural model errors dominate. PNLSS: polynomial nonlinear state-space model.

CONCLUSIONS

In this article, an extensive overview of the nonlinear system identification process is given. Nonlinear system identification is a very rich topic with many different aspects. The selection of the topics and the organization of the discussion are strongly influenced by the authors' experiences. The discussion is aligned along the following topics: 1) whether a nonlinear modeling approach is needed or if it is still possible to address users' questions using a linear design; 2) the lead actors of the system identification process: experiment design, model structure selection, choice of the cost function, and validation; and 3) an illustration of the wide variety of methods on a series of experiments. The sidebars provide more detailed technical background information on some issues.

The main conclusions are formulated as a set of guidelines and summaries. More detailed guidelines are provided throughout the article.

- *Linear or nonlinear system identification?* The need for nonlinear modeling starts where linear modeling fails to solve the problems. Nonlinear system identification is more involved than linear because models require higher flexibility, and the presence of structural model errors is difficult to avoid. Tools are available to check the level of nonlinearity in a nonparametric preprocessing step, which enables a well-informed decision to be made.
- *The main actors of the nonlinear system identification process*
  - *Experiment design*
    - Use periodic excitations whenever possible.
    - Cover the domain of interest to keep structural model errors under control. Covering the amplitude range and the frequency band of interest are necessary but not sufficient conditions to guarantee that no internal extrapolation will appear in later use of the model.
    - The experiment should be informative to keep the noise-induced uncertainty low (make the Fisher information matrix large).
  - *Selection of a model structure*: The choice of the model class is driven by: 1) the user preference (white-box or black-box models), 2) the system behavior (open-loop or closed-loop nonlinear system), and 3) models for simulation or prediction.
    - *User choice*: Decide how much physical insight will be injected. The palette provides an overview of models ranging from white- to black-box modeling.
    - *System behavior*: Fading memory (with nonlinear open-loop models, the nonlinearity is not captured in a dynamic loop) is much easier to identify than nonlinear closed-loop models, where the nonlinearity is part of a dynamic closed loop. However, addressing complex behaviors like shifting resonances, changing damping, or even chaotic behavior requires nonlinear closed-loop model structures.
    - The model complexity and effort are strongly affected by the later use of the model (prediction or simulation). The development of a good prediction model is less demanding than obtaining a good simulation model.

The development of a good prediction model is less demanding than obtaining a good simulation model.

- *Choice of the cost function*
  - Do structural model errors dominate?
    - *Yes*: The choice of the cost function is set by user criteria to shape the model errors.
    - *No*: The choice of the cost function is set by the noise properties.
  - Regularization is a very powerful tool that can help to keep the model complexity under control. It can be used in black-box models to either create sparse models or to impose smooth solutions.
- *Validation:* Does the model meet the user needs? Are the errors small enough in the domain of interest? Is there information left in the data?
  - Test the model on new data and check the internal domain.
  - Check the linear validation criteria (a whiteness test of the output residuals and a cross-correlation test between the input and output residuals).
  - Nonlinear validation criteria (for example, higher-order moment tests)

» *Uncertainty bounds*: If structural model errors are present (the model does not pass the validation test, but it is good enough for the intended application), no reliable theoretical uncertainty bounds are available. The variability of the model should be obtained from repeated experiments with varying excitations.

ACKNOWLEDGMENTS

This work was supported in part by the Flanders Fund for Scientific Research (FWO), the Vrije Universiteit Brussel (VUB), Brussels, Belgium, and the European Research Council Advanced Grant SNLSID, under contract 320378. The authors thank Julien Ertveldt (VUB) for the results on the wind tunnel experiments, including Figure S5; Dhammika Widanalage and Julian Stouv (VUB) for the wet-clutch results shown in Figure S2; Gaëtan Kerschen and Jean-Philippe Noël (University of Liège, Belgium) and Bart Peeters (Siemens Product Life-cycle Management Belgium) for the results on the ground vibration test shown in Figure 2; Torbjön Wigren (University Uppsala, Sweden), Maarten Schoukens (VUB), and Keith Worden and his team (University of Sheffield, United Kingdom) for the results on the cascaded water tanks, including Figure 11; Oliver Nelles (University of Siegen, Germany) and his team for the high-pressure fuel supply system example, including Figures 17 and 18; Martijn Vlaar and his team (Technical University Delft, The Netherlands) and Georgios Birpoutsoukis (VUB) for the results on the brain modeling (Figure 24); Rishi Relan (VUB) for the results on the battery model in Figures 27 and 28; Erliang Zhang (Zhengzhou University, China) and Maarten Schoukens (VUB) for the results on process noise detection in Figure S10; Alireza Fakhrizadeh Esfahani (VUB) and Jean-Philippe Noël (University of Liège) for the results on the Bouc–Wen hysteresis problem (Figure S16); and Thomas B. Schön (University Uppsala) for the contribution to "Identifying Nonlinear Dynamical Systems in the Presence of Process Noise." Ulla Saleneck suggested the names for the different shades of gray.

AUTHOR INFORMATION

Johan Schoukens (johan.schoukens@vub.be) received the master's degree in electrical engineering in 1980 and the Ph.D. degree in engineering sciences in 1985, both from the Vrije Universiteit Brussel (VUB), Brussels, Belgium. In 1991, he received the degree of Geaggregeerde voor het Hoger Onderwijs from the VUB and, in 2014, the doctor of science degree from the University of Warwick, Coventry, United Kingdom. From 1981 to 2000, he was a researcher at the Belgian National Fund for Scientific Research, Department of Electrical Engineering, VUB. From 2000 to 2018, he was a professor in electrical engineering, and since 2018 he has been an emeritus professor in the Department of Engineering Technology (INDI), VUB, and the Department of Electrical Engineering, Technical University Eindhoven, The Netherlands. From 2009 to 2016, he was a visiting professor at the Katholieke Universiteit Leuven, Belgium. His main research interests include system identification, signal processing, and measurement techniques. He is a Fellow of the IEEE and a member of the Royal Flemish Academy of Belgium for Sciences and the Arts. He received the 2002 Andrew R. Chi IEEE Transactions on Instrumentation and Measurement Best Paper Award, the 2002 IEEE Instrumentation and Measurement Society Distinguished Service Award, and the 2007 Belgian Francqui Chair at the Université Libre de Bruxelles, Brussels, Belgium. In 2011, he received a doctor honoris causa degree from the Budapest University of Technology and Economics, Hungary. Since 2013, he has been an honorary professor at the University of Warwick.

Lennart Ljung received the Ph.D. degree in automatic control from Lund Institute of Technology, Sweden, in 1974. Since 1976, he has been a professor and chair of automatic control at Linköping University, Sweden. He has held visiting positions at Stanford University, California, and the Massachusetts Institute of Technology and has written several books on system identification and estimation. He is a Fellow of the IEEE and IFAC, and an IFAC advisor. He is a member of the Royal Swedish Academy of Sciences, the Royal Swedish Academy of Engineering Sciences, and Academia Europaea and an honorary member of the Hungarian Academy of Engineering, an honorary professor of the Chinese Academy of Mathematics and Systems Science, and a foreign member of the U.S. National Academy of Engineering. He has received honorary doctorates from the Baltic State Technical University, St. Petersburg, Russia; Uppsala University, Sweden; the Technical University of Troyes, France; the Catholic University of Leuven, Belgium; and Helsinki University of Technology, Finland. In 2002, he was awarded the IFAC Giorgio Quazza Medal and, in 2017, the IFAC Nathaniel B. Nichols Medal. In 2003, he was presented with the Hendrik W. Bode Lecture Prize from the IEEE Control Systems Society and the IEEE Control Systems Award in 2007. He was honored with the Great Gold Medal from the Royal Swedish Academy of Engineering Sciences in 2018.

REFERENCES

[1] L. A. Zadeh, "From circuit theory to system theory," *Proc. IRE*, vol. 50, no. 5, pp. 856–865, 1962.

[2] K. J. Åström and P. Eykhoff, "System identification: A survey," *Automatica*, vol. 7, no. 2, pp. 123–167, 1971.

[3] G. E. P. Box and G. M. Jenkins, *Time Series Analysis, Forecasting and Control*. San Francisco, CA: Holden-Day, 1970.

[4] K. J. Åström, *Introduction to Stochastic Control Theory*. New York: Academic, 1970.

[5] P. Eykhoff, *System Identification and State Estimation*. New York: Wiley, 1974.

[6] L. Ljung, *System Identification: Theory for the User* (2nd ed., 1999). Englewood Cliffs, NJ: Prentice Hall, 1987.

[7] T. Söderström and P. Stoica, *System Identification*. Englewood Cliffs, NJ: Prentice Hall, 1989.

[8] J. M. T. Thompson and H. B. Stewart, *Nonlinear Dynamics and Chaos*. New York: Wiley, 1986.

[9] K. Narendra and K. Parthasarty, "Identification and control of dynamical systems using neural networks," *IEEE Trans. Neural Netw.*, vol. 1, no. 1, pp. 4–27, 1990.

[10] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Netw.*, vol. 2, no. 5, pp. 359–366, 1989.

[11] J. Sjöberg et al., "Nonlinear black-box modeling in system identification: A unified overview," *Automatica*, vol. 31, no. 12, pp. 1691–1724, 1995.

[12] A. Juditsky et al., "Nonlinear black-box models in system identification: Mathematical foundations," *Automatica*, vol. 31, no. 12, pp. 1725–1750, 1995.

[13] J. A. K. Suykens, J. P. L. Vandewalle, and B. L. R. De Moor, *Artificial Neural Networks for Modeling and Control of Non-Linear Systems*. Norwell, MA: Kluwer, 1996.

[14] J. A. K. Suykens, J. Vandewalle, and B. L. R. De Moor, "NL theory: Checking and imposing stability of recurrent neural networks for nonlinear modeling," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2682–2691, 1997.

[15] O. Nelles, *Nonlinear System Identification*. New York: Springer-Verlag, 2001.

[16] K. Worden and G. R. Tomlinson, *Nonlinearity in Structural Dynamics: Detection, Identification and Modelling*. Bristol, UK: IOP Publishing, 2001.

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

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

[19] S. A. Billings, *Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains*. Hoboken, NJ: Wiley, 2013.

[20] W. D. Widanage, J. Stoev, A. Van Mulders, J. Schoukens, and G. Pinte, "Nonlinear system-identification of the filling phase of a wet-clutch system," *Control Eng. Practice*, vol. 19, no. 12, pp. 1506–1516, 2011.

[21] P. C. Young, *Recursive Estimation and Time-Series Analysis: An Introduction for the Student and Practitioner*. New York: Springer-Verlag, 2011.

[22] D. Mayne, "Model predictive control: Recent developments and future promise," *Automatica*, vol. 50, no. 12, pp. 2967–2986, 2014.

[23] J. P. Noël, L. Renson, and G. Kerschen, "Complex dynamics of a nonlinear aerospace structure: Experimental identification and modal interactions," *J. Sound Vib.*, vol. 333, no. 12, pp. 2588–2607, 2014.

[24] J. P. Noël, J. Schoukens, and G. Kerschen, "Grey-box nonlinear state-space modelling for mechanical vibrations identification," in *Preprints of the 17th IFAC Symp. System Identification*, Beijing, China, Oct. 19–21, 2015, pp. 817–822.

[25] P. C. Young and A. Janot, "Efficient parameterisation of nonlinear system models: A comment on Noël and Schoukens (2018)," *Int. J. Control*, pp. 1–5, Sept. 2018. doi: 10.1080/00207179.2018.1521008.

[26] D. T. Westwick, G. Hollander, K. Karami, and J. Schoukens, "Using decoupling methods to reduce polynomial NARX models," *18th IFAC Symp. System Identification, SYSID 2018*, IFAC-PapersOnLine, vol. 51, no. 15, pp. 796–801, 2018.

[27] Y. Ueda, "Survey of regular and chaotic phenomena in the forced duffing oscillator," *Chaos Solitons Fractals*, vol. 1, no. 3, pp. 199–231, 1991.

[28] M. Schetzen, *The Volterra and Wiener Theories of Nonlinear Systems*. New York: Wiley, 1980.

[29] J. Schoukens, J. G. Nemeth, P. Crama, Y. Rolain, and R. Pintelon, "Fast approximate identification of nonlinear systems," *Automatica*, vol. 39, no. 7, pp. 1267–1274, 2003.

[30] R. Pintelon and J. Schoukens, *System Identification: A Frequency Domain Approach*, 2nd ed. Hoboken, NJ: Wiley-IEEE Press, 2012.

[31] R. Pintelon, E. Louarroudi, and J. Lataire, "Nonparametric estimation of time-variant dynamics," in *Proc. 17th IFAC Symp. System Identification*, Beijing, China, 2015.

[32] M. Vaes et al., "Nonlinear ground vibration identification of an F-16 aircraft. Part I: Fast nonparametric analysis of distortions in FRF measurements," in *Proc. 16th Int. Forum on Aeroelasticity and Structural Dynamics (IFASD),* Saint Petersburg, Russia, 2015, paper no. IFASD-2015–101.

[33] T. Dossogne, J. P. Noël, G. Grappasonni, G. Kerschen, B. Peeters, and J. Schoukens, "Nonlinear ground vibration identification of an F-16 aircraft. Part 2: Understanding nonlinear behaviour in aerospace structures using sine-sweep testing," in *Proc. 16th Int. Forum on Aeroelasticity and Structural Dynamics (IFASD)*, Saint Petersburg, Russia, 2015, paper no. IFAS-2015–35.

[34] A. Gelb and W. E. Vander Velde, *Multiple-Input Describing Functions and Nonlinear System Design*. New York: McGraw-Hill, 1968.

[35] F. Arfaei Malekzadeh, R. Mahmoudi, and A. van Roermund, *Analog Dithering Techniques for Wireless Transmitters*. New York: Springer-Verlag, 2013.

[36] D. P. Atherton, *Nonlinear Control Engineering*. New York: Van Nostrand Reinhold, 1982.

[37] P. Carbone and D. Petri, "Performance of stochastic and deterministic dithered quantizers," *IEEE Trans. Instrum. Meas.*, vol. 49, no. 2, pp. 337–340, 2000.

[38] E. Zhang, M. Schoukens, and J. Schoukens, "Structure detection of Wiener–Hammerstein systems with process noise," *IEEE Trans. Instrum. Meas.*, vol. 66, no. 3, pp. 569–576, 2017.

[39] M. Schoukens and J. P. Noël, "Wiener–Hammerstein benchmark with process noise," in *Proc. 2nd Workshop on Nonlinear System Identification Benchmarks,* Brussels, 2016. [Online]. Available: http://www.nonlinearbenchmark.org

[40] A. Hagenblad, L. Ljung, and A. Wills, "Maximum likelihood identification of Wiener models," *Automatica*, vol. 44, no. 11, pp. 2697–2705, 2008.

[41] A. Doucet and A. M. Johansen, "A tutorial on particle filtering and smoothing: Fifteen years later," in *Nonlinear Filtering Handbook*, D. Crisan and B. Rozovsky, Eds. New York: Oxford Univ. Press, 2011.

[42] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *Proc. Inst. Elect. Eng. Radar Signal Process.*, vol. 140, no. 2, pp. 107–113, 1993.

[43] P. Del Moral, *Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications*. New York: Springer-Verlag, 2004.

[44] M. K. Pitt, R. dos Santos Silva, R. Giordani, and R. Kohn, "On some properties of Markov chain Monte Carlo simulation methods based on the particle filter," *J. Econometrics*, vol. 171, no. 2, pp. 134–151, 2012.

[45] G. Poyiadjis, A. Doucet, and S. S. Singh, “Particle approximations of the score and observed information matrix in state space models with application to parameter estimation,” *Biometrika*, vol. 98, no. 1, pp. 65–80, 2011.

[46] A. Doucet, P. E. Jacob, and S. Rubenthaler, “Derivative-free estimation of the score vector and observed information matrix with application to state-space models,” Tech. Rep. arXiv:1304.5768, July 2015.

[47] L. Bottou, F. E. Curtis, and J. Nocedal, “Optimization methods for large-scale machine learning,” *SIAM Rev.*, vol. 60, no. 2, pp. 223–311, 2018.

[48] A. Dempster, N. Laird, and D. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *J. Roy. Stat. Soc. B*, vol. 39, no. 1, pp. 1–38, 1977.

[49] T. B. Schön, A. Wills, and B. Ninness, “System identification of nonlinear state-space models,” *Automatica*, vol. 47, no. 1, pp. 39–49, Jan. 2011.

[50] J. Olsson, R. Douc, O. Cappé, and E. Moulines, “Sequential Monte Carlo smoothing with application to parameter estimation in nonlinear state-space models,” *Bernoulli*, vol. 14, no. 1, pp. 155–179, 2008.

[51] V. Peterka, “Bayesian system identification,” *Automatica*, vol. 17, no. 1, pp. 41–53, 1981.

[52] C. Andrieu, A. Doucet, and R. Holenstein, “Particle Markov chain Monte Carlo methods,” *J. Roy. Stat. Soc. B*, vol. 72, no. 3, pp. 269–342, 2010.

[53] T. B. Schön et al., “Sequential Monte Carlo methods for system identification,” in *Proc. 17th IFAC Symp. System Identification (SYSID)*, Beijing, China, Oct. 2015, pp. 775–786.

[54] N. Kantas, A. Doucet, S. S. Singh, J. M. Maciejowski, and N. Chopin, “On particle methods for parameter estimation in state-space models,” *Stat. Sci.*, vol. 30, no. 3, pp. 328–351, 2015.

[55] A. Wills, T. B. Schön, L. Ljung, and B. Ninness, “Identification of Hammerstein–Wiener models,” *Automatica*, vol. 49, no. 1, pp. 70–81, 2013.

[56] F. Lindsten, T. B. Schön, and M. I. Jordan, “Bayesian semiparametric wiener system identification,” *Automatica*, vol. 49, no. 7, pp. 2053–2063, 2013.

[57] R. Tibsharani, “Regression shrinkage and selection via the lasso,” *J. Roy. Stat. Soc. B*, vol. 58, no. 1, pp. 267–288, 1996.

[58] G. Pillonetto, F. Dinuzzo, T. Chen, G. De Nicolao, and L. Ljung, “Kernel methods in system identification, machine learning and function estimation: A survey,” *Automatica*, vol. 50, no. 3, pp. 657–682, 2014.

[59] G. Favier and T. Bouilloc, “Parametric complexity reduction of Volterra models using tensor decompositions,” in *Proc. 17th European Signal Processing Conf. (EUSIPCO)*, Glasgow, Scotland, Aug. 2009, pp. 2288–2292.

[60] M. Schoukens and Y. Rolain, “Crossterm elimination in parallel Wiener systems using a linear input transformation,” *IEEE Trans. Instrum. Meas.*, vol. 61, no. 3, pp. 845–847, 2012.

[61] K. Tiels and J. Schoukens, “From coupled to decoupled polynomial representations in parallel Wiener–Hammerstein models,” in *Proc. 52nd IEEE Conf. Decision and Control (CDC)*, Florence, Italy, Dec. 2013, pp. 4937–4942.

[62] M. Schoukens, K. Tiels, L. Ishteva, and J. Schoukens, “Identification of parallel Wiener–Hammerstein systems with a decoupled static nonlinearity,” in *Proc. 19th World Congr. Int. Federation of Automatic Control*, Cape Town, South Africa, Aug. 2014, pp. 505–510.

[63] P. Dreesen, M. Schoukens, K. Tiels, and J. Schoukens, “Decoupling static nonlinearities in a parallel Wiener–Hammerstein system: A first-order approach,” in *Proc. IEEE Int. Instrumentation and Measurement Technology Conf. (I2MTC)*, Pisa, Italy, May 2015.

[64] P. Dreesen, M. Ishteva, and J. Schoukens, “Decoupling multivariate polynomials using first-order information and tensor decompositions,” *Siam J. Matrix Anal. Appl.*, vol. 36, no. 2, pp. 864–879, 2015.

[65] A. Fakhrizadeh Esfahani, P. Dreesen, K. Tiels, J. Noël, and J. Schoukens, “Parameter reduction in nonlinear state-space identification of hysteresis,” *Mech. Syst. Signal Process.*, vol. 104, pp. 884–895, May 2018. doi: 10.1016/j.ymssp.2017.10.017.

[66] G. Hollander, P. Dreesen, M. Ishteva, and J. Schoukens, “Approximate decoupling of multivariate polynomials using weighted tensor decomposition,” *Numer. Linear Algebra Appl.*, vol. 25, no. 2, 2018. doi: 10.1002/nla.2135.

[67] P. Sliwinski, A. Marconato, P. Wachel, and G. Birpoutsoukis, “Nonlinear system modelling based on constrained Volterra series estimates,” *IET Control Theory Appl.*, vol. 11, no. 15, pp. 2623–2629, 2017.

[68] J. Stoev, J. Ertveldt, T. Oomen, and J. Schoukens, “Tensor methods for MIMO decoupling and control design using frequency response functions,” *Mechatronics*, vol. 45, pp. 71–81, Aug. 2017. doi: 10.1016/j.mechatronics.2017.05.009.

[69] J. Schoukens, K. Godfrey, and M. Schoukens, “Nonparametric data driven modeling of linear systems: Estimating the frequency response and impulse response function,” *IEEE Control Syst. Mag.*, vol. 38, no. 4, pp. 49–88, 2018.

[70] J. Schoukens, R. Pintelon, and Y. Rolain, “Study of conditional ML estimators in time and frequency-domain system identification,” *Automatica*, vol. 35, no. 1, pp. 91–100, 1999.

[71] J. Schoukens, Y. Rolain, G. Vandersteen, and R. Pintelon, “User friendly Box–Jenkins identification using nonparametric noise models,” in *Proc. 50th IEEE Conf. Decision and Control and European Control Conf. (CDC-ECC)*, Orlando, FL, USA, Dec. 12–15, 2011. doi: 10.1109/CDC.2011.6160204.

[72] J. Schoukens, G. Vandersteen, K. Barbe, and R. Pintelon, “Nonparametric preprocessing in system identification: A powerful tool,” *Eur. J. Control*, vol. 15, no. 3–4, pp. 260–274, 2009.

[73] R. Pintelon, J. Schoukens, G. Vandersteen, and K. Barbe, “Estimation of nonparametric noise and FRF models for multivariable systems. Part I: Theory,” *Mech. Syst. Signal Process.*, vol. 24, no. 3, pp. 573–595, 2010.

[74] R. Pintelon, J. Schoukens, G. Vandersteen, and K. Barbe, “Estimation of nonparametric noise and FRF models for multivariable systems. Part II: Extensions, applications,” *Mech. Syst. Signal Process.*, vol. 24, no. 3, pp. 596–616, 2010.

[75] J. Schoukens and R. Pintelon, “Study of the variance of parametric estimates of the best linear approximation of nonlinear systems,” *IEEE Trans. Instrum. Meas.*, vol. 59, no. 12, pp. 3159–3167, 2010.

[76] H. Hjalmarsson, “System identification of complex and structured systems,” *Eur. J. Control*, vol. 15, no. 3–4, pp. 275–310, 2009.

[77] C. Criens, T. van Keulen, F. Willems, and M. Steinbuch, “A control oriented multivariable identification procedure for turbocharged diesel engines,” *Int. J. Powertrains*, vol. 5, no. 2, pp. 95–119, 2016.

[78] E. Wernholt and S. Gunnarsson, “Estimation of nonlinear effects in frequency domain identification of industrial robots,” *IEEE Trans. Instrum. Meas.*, vol. 57, no. 4, pp. 856–863, 2008.

[79] J. Schoukens, M. Vaes, and R. Pintelon, “Linear system identification in a nonlinear setting: Nonparametric analysis of the nonlinear distortions and their impact on the best linear approximation,” *IEEE Control Syst. Mag.*, vol. 36, no. 3, pp. 38–69, 2016.

[80] *IEEE Standard for Digitizing Waveform Recorders*, IEEE Standard 1057, 1994.

[81] H. Hjalmarsson and L. Ljung, “Estimating model variance in the case of undermodeling,” *IEEE Trans. Autom. Control*, vol. 37, no. 7, pp. 1004–1008, 1992.

[82] J. J. Bussgang, “Cross-correlation functions of amplitude-distorted Gaussian signals,” MIT Lab. Electronics, Tech. Rep. 216, 1952.

[83] L. Ljung, “Estimating linear time-invariant models of nonlinear time-varying systems,” *Eur. J. Control*, vol. 7, no. 2–3, pp. 203–219, 2001.

[84] M. Enqvist, “Linear models of nonlinear systems,” Ph.D. thesis, Inst. Technol., Linköping Univ., Sweden, 2005.

[85] M. Enqvist and L. Ljung, “Linear approximations of nonlinear FIR systems for separable input processes,” *Automatica*, vol. 41, no. 3, pp. 459–473, 2005.

[86] M. Enqvist, “Separability of scalar random multisine signals,” *Automatica*, vol. 47, no. 9, pp. 1860–1867, 2011.

[87] J. Schoukens, T. Dobrowiecki, and R. Pintelon, “Parametric and non-parametric identification of linear systems in the presence of nonlinear distortions: A frequency domain approach,” *IEEE Trans. Autom. Control*, vol. 43, no. 2, pp. 176–190, 1998.

[88] J. Schoukens, J. Lataire, R. Pintelon, G. Vandersteen, and T. Dobrowiecki, “Robustness issues of the best linear approximation of a nonlinear system,” *IEEE Trans. Instrum. Meas.*, vol. 58, no. 5, pp. 1737–1745, 2009.

[89] M. Schoukens, R. Pintelon, T. Dobrowiecki, and J. Schoukens, “Extending the best linear approximation framework to the process noise case,” arXiv Preprint, arXiv:1804.07510, 2018.

[90] R. Pintelon and J. Schoukens, “FRF measurement of nonlinear systems operating in closed loop,” *IEEE Trans. Instrum. Meas.*, vol. 62, no. 5, pp. 1334–1345, May 2013.

[91] A. Fakhrizadeh Esfahani, J. Schoukens, and L. Vanbeylen, “Using the best linear approximation with varying excitation signals for nonlinear system characterization,” *IEEE Trans. Instrum. Meas.*, vol. 65, no. 5, pp. 1271–1280, May 2016.

[92] J. Ertveldt, J. Schoukens, R. Pintelon, and S. Vanlanduit, “Experiments on the active Aeroelastic Test Bench (AATB) for the identification

of unsteady aerodynamics," in *Proc. Int. Forum on Aeroelasticity and Structural Dynamics (IFASD)*, Saint-Petersburg, Russia, 2015, paper. no. IFASD-2015−050.

[93] J. Ertveldt, "Application of frequency-domain system identification to wind tunnel experiments on the Active Aeroelastic Test Bench," Ph.D. thesis, Vrije Univ. Brussel, 2017.

[94] H. K. Wong, J. Schoukens, and K. Godfrey, "Analysis of best linear approximation of a Wiener–Hammerstein system for arbitrary amplitude distributions," *IEEE Trans. Instrum. Meas.*, vol. 61, no. 3, pp. 645–654, 2012.

[95] T. P. Dobrowiecki, "On Bussgang's theorem, separability, and colored random phase multisines," *IEEE Trans. Instrum. Meas.*, pp. 1–9, 2019. doi: 10.1109/TIM.2019.2906333.

[96] L. Ljung, "Perspectives on system identification," *Annu. Rev. Control*, vol. 34, no. 1, pp. 1–12, 2010.

[97] R. K. Pearson, "Selecting nonlinear model structures for computer control," *J. Process Control*, vol. 13, no. 1, pp. 1–26, 2003.

[98] G. P. Rao and H. Unbehauen, "Identification of continuous-time systems," *Proc. Inst. Elect. Eng. Control Theory Appl.*, vol. 153, no. 2, pp. 185–220, 2006.

[99] G. C. Goodwin, J. C. Agüero, M. E. C. Garrido, M. E. Salgado, and J. I. Yuz, "Sampling and sampled-data models: The interface between the continuous world and digital algorithms," *IEEE Control Syst. Mag.*, vol. 33, no. 5, pp. 34–53, 2013.

[100] R. K. Pearson and U. Kotta, "Nonlinear discrete-time models: State-space vs. I/O representations," *J. Process Control*, vol. 14, no. 5, pp. 533–538, 2004.

[101] R. K. Pearson, "Nonlinear empirical modeling techniques," *Comput. Chem. Eng.*, vol. 30, no. 10–12, pp. 1514–1528, 2006.

[102] H. Garnier, "Direct continuous-time approaches to system identification. Overview and benefits for practical applications," *Eur. J. Control*, vol. 24, pp. 50–62, 2015. doi: 10.1016/j.ejcon.2015.04.003.

[103] B. Zhang and S. A. Billings, "Identification of continuous-time nonlinear systems: The nonlinear difference equation with moving average noise (NDEMA) framework," *Mech. Syst. Signal Process.*, vol. 60, pp. 810–835, Aug. 2015. doi: 10.1016/j.ymssp.2015.01.009.

[104] J. Schoukens, R. Relan, and M. Schoukens, "Discrete time approximation of continuous time nonlinear state space models," in *Proc. 20th World Congress Int. Federation Automatic Control*, IFAC-PapersOnLine, vol. 50, no. 1, pp. 8339–8346, 2017.

[105] R. F. Baum, "The correlation function of Gaussian noise passed through nonlinear devices," *IEEE Trans. Inf. Theory*, vol. 15, no. 4, pp. 448–456, 1982.

[106] J. Schoukens, R. Pintelon, and H. Van hamme, "Identification of linear dynamic systems using piecewise-constant excitations: Use, misuse and alternatives," *Automatica*, vol. 30, no. 7, pp. 1153–1169, 1994.

[107] A. V. Oppenheim and A. S. Willsky, *Signals and Systems*. London: Prentice Hall, 1997.

[108] M. Tiller, *Introduction to Physical Modeling with Modelica*. Boston: Kluwer Academic Publishers, 2001.

[109] K. E. Brenan, S. L. Campbell, and L. R. Petzold, *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*. Amsterdam, The Netherlands: North-Holland, 1989.

[110] L. Ljung and T. Glad, *Modeling and Identification of Dynamic Systems*. Lund, Sweden: Studentlitteratur, 2016.

[111] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*. Englewood Cliffs, NJ: Prentice Hall, 1979.

[112] A. Doucet and V. B. Tadić, "Parameter estimation in general state-space models using particle methods," *Ann. Inst. Stat. Math.*, vol. 55, no. 2, pp. 409–422, 2003.

[113] P. Fritzson, *Principles of Object-Oriented Modeling and Simulation with Modelica 3.3: A Cyber-Physical Approach*, 2nd ed. Hoboken, NJ: Wiley-IEEE Press, 2014.

[114] Mathworks, "Simscape." Accessed on: 2019. [Online]. Available: www.mathworks.com/products/simscape

[115] T. Bohlin, *Practical Grey-Box Process Identification*. London: Springer-Verlag, 2006.

[116] R. Murray-Smith and T. A. Johansen, Eds., *Multiple Model Approaches to Modeling and Control*. London: Taylor and Francis, 1997.

[117] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its application to modeling and control," *IEEE Trans. Syst., Man, Cybern.*, vol. 15, no. 1, pp. 116–132, 1985.

[118] L. Lee and K. Poolla, "Identification of linear parameter-varying systems using non-linear programming," *ASME J. Dyn. Syst. Meas. Control*, vol. 121, no. 1, pp. 71–78, 1999.

[119] R. Toth, *Modeling and Identification of Linear Paramter-Varying Systems*. New York: Springer-Verlag, 2010.

[120] R. Toth, P. C. S. Heuberger, and P. M. J. Van den Hof, "Prediction-error identification of LPV systems: Present and beyond," in *Control of Linear Parameter Varying Systems with Applications*, ch. 2, J. J. Mohammadpour and C. W. Scherer, Eds. New York: Springer-Verlag, 2012, pp. 27–58.

[121] Q. Zhang and L. Ljung, "From structurally independent local LTI models to LPV model," *Automatica*, vol. 84, pp. 232–235, Oct. 2017. doi: 10.1016/j.automatica.2017.06.006.

[122] S. Boyd and L. O. Chua, "Fading memory and the problem of approximating nonlinear operators with Volterra series," *IEEE Trans. Circuits Syst.*, vol. 32, no. 11, pp. 1150–1161, 1985.

[123] G. Palm, "On representation and approximation of nonlinear systems. Part II: Discrete time," *Biol. Cybern.*, vol. 34, no. 1, pp. 49–52, 1979.

[124] M. Schoukens and K. Tiels, "Identification of block-oriented nonlinear systems starting from linear approximations: A survey," *Automatica*, vol. 85, pp. 272–292, Nov. 2017. doi: 10.1016/j.automatica.2017.06.044.

[125] M. Schoukens, A. Marconato, R. Pintelon, G. Vandersteen, and Y. Rolain, "Parametric identification of parallel Wiener-Hammerstein systems," *Automatica*, vol. 51, no. 1, pp. 111–122, 2015.

[126] J. Schoukens, L. Gommé, W. Van Moer, and Y. Rolain, "Identification of a block-structured nonlinear feedback system, applied to a microwave crystal detector," *IEEE Trans. Instrum. Meas.*, vol. 57, no. 8, pp. 1734–1740, 2008.

[127] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA: MIT Press, 2006.

[128] A. Van Mulders, J. Schoukens, M. Volckaert, and M. Diehl, "Two nonlinear optimization methods for black box identification compared," *Automatica*, vol. 46, no. 10, pp. 1675–1681, 2010.

[129] I. J. Leontaritis and S. A. Billings, "Input output parametric models for non-linear systems. Part II: Stochastic non-linear systems," *Int. J. Control*, vol. 41, no. 2, pp. 329–344, 1985.

[130] M. Liu, G. Chowdhary, B. Castra da Silva, S.-Y. Liu, and J. P. How, "Gaussian processes for learning and control: A tutorial with examples," *IEEE Control Syst.*, vol. 38, no. 5, pp. 53–86, Oct. 2018.

[131] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA: MIT Press, 2006.

[132] W. X. Zhao, H.-F. Chen, and E.-W. Bai, "Kernel-based local order estimation of nonlinear nonparametric systems," *Automatica*, vol. 51, pp. 243–254, Jan. 2015. doi: 10.1016/j.automatica.2014.10.069.

[133] W. X. Zhao, H.-F. Chen, E.-W. Bai, and K. Li, "Local variable selection of nonlinear nonparametric systems by first order expansion," *Syst. Control Lett.*, vol. 111, pp. 1–8, Jan. 2018.

[134] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org

[135] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Belmont, CA: Wadsworth, 1984.

[136] A. Tikhonov and V. Arsenin, *Solutions of Ill-Posed Problems*. Washington, DC: Winston/Wiley, 1977.

[137] G. Wahba, *Spline Models for Observational Data*. Philadelphia, PA: SIAM, 1990.

[138] G. Kimeldorf and G. Wahba, "A correspondance between Bayesian estimation of stochastic processes and smoothing by splines," *Ann. Math. Stat.*, vol. 41, no. 2, pp. 495–502, 1971.

[139] R. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*. Cambridge, MA: MIT Press, 2001.

[140] A. Chiuso and G. Pillonetto, "A Bayesian approach to sparse dynamic network identification," *Automatica*, vol. 48, no. 8, pp. 1107–1111, Apr. 2012.

[141] W. Pan, Y. Yuan, L. Ljung, J. Gonsalves, and G.-B. Stan, "Identification of nonlinear state-space systems from heterogeneous datasets," *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 2, pp. 737–747, 2018.

[142] J. Fan and I. Gijbels, *Local Polynomial Modelling and its Applications: Monographs on Statistics and Applied Probability*. London, U.K.: Chapman & Hall, 1996.

[143] E. Nadaraya, "On estimating regression," *Theory Prob. Appl.*, vol. 9, no. 1, pp. 141–142, 1964.

[144] V. A. Epanechnikov, "Non-parametric estimation of a multivariate probability density," *Theory Prob. Appl.*, vol. 14, no. 1, pp. 4–20, 1969.

[145] J. Roll, A. Nazin, and L. Ljung, “Non-linear system identification via direct weight optimization,” *Automatica*, vol. 41, no. 3, pp. 475–490, Mar. 2005.

[146] J. B. Tenenbaum, V. de Silva, and J. C. Langford, “A global geometric framework for nonlinear dimensionality reduction,” *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[147] S. T. Roweis and L. K. Saul, “Nonlinear dimension reduction by locally linear embedding,” *Science*, vol. 290, no. 5500, pp. 2323–2326, Dec. 2000.

[148] J. Schoukens, R. Pintelon, E. Vanderouderaa, and J. Renneboog, “Survey of excitation signals for FFT based signal analyzers,” *IEEE Trans. Instrum. Meas.*, vol. 37, no. 3, pp. 342–352, 1988.

[149] J. Schoukens, M. Vaes, and R. Pintelon, “Linear system identification in a nonlinear setting: Nonparametric analysis of the nonlinear distortions and their impact on the best linear approximation,” *IEEE Control Syst.*, vol. 36, pp. 38–69, June 2016.

[150] J. Schoukens, R. Pintelon, and Y. Rolain, *Mastering System Identification in 100 Exercises*. Hoboken, NJ: Wiley, 2012.

[151] E. Geerardyn, Y. Rolain, R. Pintelon, and J. Schoukens, “Design of quasi-logarithmic multisine excitations for robust broad frequency band measurements,” *IEEE Trans. Instrum. Meas.*, vol. 62, no. 5, pp. 1364–1372, 2013.

[152] J. Schoukens and T. Dobrowiecki, “Design of broadband excitation signals with a user imposed power spectrum and amplitude distribution,” in *Proc. IEEE Instrumentation and Measurement Technology Conf.*, St. Paul, MN, USA, 1998, pp. 1002–1005.

[153] R. K. Mehra, “Optimal input signals for parameter estimation in dynamic systems: Survey and new results,” *IEEE Trans. Autom. Control*, vol. 19, no. 6, pp. 753–768, 1974.

[154] G. C. Goodwin and R. L. Payne, *Dynamic System Identification: Experiment Design and Data Analysis*. San Francisco, CA: Academic, 1977.

[155] L. Pronzato, “Optimal experimental design and some related control problems,” *Automatica*, vol. 44, no. 2, pp. 303–325, 2008.

[156] D. E. Rivera, H. Le, M. W. Braun, and H. D. Mittelman, “Plant-friendly system identification: A challenge for the process industries,” *IFAC Proc. Vol.*, vol. 36, no. 16, pp. 891–896, 2003.

[157] M. Gevers, “Identification for control: From the early achievements to the revival of experiment design,” *Eur. J. Control*, vol. 11, no. 4–5, pp. 335–352, 2005.

[158] X. Bombois and G. Scorletti, “Design of least costly identification experiments: The main philosophy accompanied by illustrative examples,” *J. Eur. Syst. Autom.*, vol. 46, no. 6–7, pp. 587–610, 2012.

[159] M. Forgione, X. Bombois, and P. M. J. Van den Hof, “Data-driven model improvement for model-based control,” *Automatica*, vol. 52, pp. 118–124, Feb. 2015. doi: 10.1016/j.automatica.2014.11.006.

[160] M. Annergren, C. A. A. Larsson, H. Hjalmarsson, X. Bombois, and B. Wahlberg, “Application-oriented input design in system identification: Optimal input design for control,” *IEEE Control Syst. Mag.*, vol. 37, no. 2, pp. 31–56, 2017.

[161] K. Mahata, J. Schoukens, and A. De Cock, “Information matrix and D-optimal design with Gaussian inputs for Wiener model identification,” *Automatica*, vol. 69, pp. 65–77, July 2016. doi: 10.1016/j.automatica.2016.02.026.

[162] M. Gevers, M. Caenepeel, and J. Schoukens, “Experiment design for the idenification of a simple wiener system,” in *Proc. 51st IEEE Conf. Decision and Control*, Maui, HI, 2012, pp. 7333–7338.

[163] G. Birpoutsoukis, “Volterra series estimation in the presence of prior knowledge,” Ph.D. thesis, Vrije Univ. Brussel, 2018.

[164] M. Forgione, X. Bombois, P. M. J. Van den Hof, and H. Hjalmarsson, “Experiment design for parameter estimation in nonlinear systems based on multilevel excitation,” in *Proc. 13th European Control Conf. (ECC)*, Strasbourg, France, 2014, pp. 25–30.

[165] A. De Cock, M. Gevers, and J. Schoukens, “D-optimal input design for nonlinear FIR-type systems: A dispersion-based approach,” *Automatica*, vol. 73, pp. 88–100, Nov. 2016. doi: 10.1016/j.automatica.2016.04.052.

[166] A. De Cock, “D-Optimal input design for the identification of structured nonlinear systems,” Ph.D. thesis, Vrije Univ. Brussel, 2017.

[167] H. A. Chianeh, J. D. Stigter, and K. J. Keesman, “Optimal input design for parameter estimation in a single and double tank system through direct control of parametric output sensitivities,” *J. Process Control*, vol. 21, no. 1, pp. 111–118, 2011.

[168] I. Gustavsson, L. Ljung, and T. Söderström, “Identification of processes in closed loop: Identification and accuracy aspects,” *Automatica*, vol. 13, no. 1, pp. 59–77, 1977.

[169] L. Ljung, “Convergence analysis of parametric identfication methods,” *IEEE Trans. Autom. Control*, vol. 23, no. 5, pp. 770–783, Oct. 1978.

[170] R. Larsson, “System identification on flight test data,” Ph.D. thesis, Linkoping Univ., 2019.

[171] R. Larsson, Z. Sjanic, M. Enqvist, and L. Ljung, “Direct prediction-error identification of unstable nonlinear systems applied to flight test data,” in *Proc. 18th IFAC Symp. System Identification*, Saint-Malo, France, July 2009, pp. 1444–1149.

[172] L. Ljung, *System Identification Toolbox for Use with Matlab,* 9th ed. Natick, MA: The MathWorks, 2018.

[173] N. Draper and H. Smith, *Applied Regression Analysis*. New York: Wiley, 1981.

[174] M. Enqvist, J. Schoukens, and R. Pintelon, “Detection of unmodeled nonlinearities using correlation methods,” in *Proc. IEEE Instrumentation and Measurement Technology Conf.*, Warsaw, Poland, May 1–3, 2007.

[175] M. Schoukens, P. Mattson, T. Wigren, and J.P. Noël, “Cascaded tanks benchmark combining soft and hard nonlinearities,” in *Proc. Workshop Nonlinear System Identification Benchmarks*, Brussels, Belgium, Apr. 2016, pp. 20–23.

[176] T. J. Rogers, G. R. Holmes, E. J. Cross, and K. Worden, “On a grey box modelling framework for nonlinear system identification,” in *Special Topics in Structural Dynamics*, vol. 6 (Conference Proceedings of the Society for Experimental Mechanics Series), N. Dervilis, Ed. Cham, Switzerland: Springer, 2017.

[177] T. Andersson and P. Pucar, “Estimation of residence time in continuous flow systems with dynamics,” *J. Process Control*, vol. 5, no. 1, pp. 9–17, Feb. 1995.

[178] T.O. Heinz, M. Schillinger, B. Hartmann, and O. Nelles, “Excitation signal design for nonlinear dynamic systems,” in *Automation Technology: Methods and Applications of Control, Regulation, and Information Technology*, K. Röpke and C. Gühmann, Eds., Berlin: Walter de Gruyter GmbH, 2017, pp. 191–208. doi: 10.1515/auto-2018-0027.

[179] T. O. Heinz and O. Nelles, “Excitation signal design for nonlinear dynamic systems with multiple inputs: A data distribution approach,” *Automatisierungstechnik*, vol. 66, no. 9, pp. 714–724, 2018.

[180] O. Nelles, “Axes-oblique partitioning strategies for local model networks,” in *Proc. 2006 IEEE Conf. Computer Aided Control System Design, 2006 IEEE Int. Conf. Control Applications, 2006 IEEE Int. Symp. Intelligent Control*, IEEE, Oct. 2006, pp. 2378–2383.

[181] N. Tietze, “Model-based calibration of engine control units using Gaussian process regression,” Ph.D. thesis, Technischen Univ. Darmstadt, 2015.

[182] S. Gunnarsson and P. Krus, “Adaptive compensation of deformations in mechanical structures controlled by hydraulic actuators,” in *Proc. 11th IFAC World Congr.*, 1990, pp. 51–55.

[183] M. Vlaar, T. Solis-Escalante, A. Vardy, F. Van der Helm, and A. Schouten, “Quantifying nonlinear contributions to cortical responses evoked by continuous wrist manipulation,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 25, no. 5, pp. 481–491, 2016.

[184] M. P. Vlaar et al., “Modeling the nonlinear cortical response in EEG evoked by wrist joint manipulation,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 26, no. 1, pp. 205–215, 2018.

[185] G. Birpoutsoukis, A. Marconato, J. Lataire, and J. Schoukens, “Regularized nonparametric Volterra kernel estimation,” *Automatica*, vol. 82, pp. 72–82, Aug. 2017. doi: 10.1016/j.automatica.2017.04.014.

[186] R. Relan, Y. Firouz, J. M. Timmermans, and J. Schoukens, “Data-driven nonlinear identification of li-ion battery based on a frequency domain nonparametric analysis,” *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 5, pp. 1825–1832, 2017.

[187] R. Relan, “Data-driven discrete-time identification of continuous time nonlinear systems and nonlinear modelling of li-ion batteries,” Ph.D. thesis, Vrije Univ. Brussel, 2017.

[188] J. Paduart, L. Lauwers, J. Swevers, K. Smolders, J. Schoukens, and R. Pintelon, “Identification of nonlinear systems using polynomial nonlinear state space models,” *Automatica*, vol. 46, no. 4, pp. 647–656, 2010.

[189] J. Schoukens et al., “Structure discrimination in block-oriented models using linear approximations: A theoretic framework,” *Automatica*, vol. 53, pp. 225–234, Mar. 2015. doi: 10.1016/j.automatica.2014.12.045.