I. Troch, F. Breitenecker, eds.   ISBN 978-3-901608-35-3

MODEL ORDER REDUCTION FOR OBJECT-ORIENTED MODELS: A CONTROL SYSTEMS PERSPECTIVE

F. Casella, F. Donida, G. Ferretti
Politecnico di Milano, Italy

Corresponding author: G. Ferretti, Politecnico di Milano, Dipartimento di Elettronica e Informazione
Piazza Leonardo da Vinci, 32, 20133 Milano, Italy, ferretti@elet.polimi.it

Abstract. In recent years several concepts and tools for modelling and simulation of nonlinear heterogeneous and multi-domain systems have been developed, speeding up to a great extent the process of an accurate analysis by simulation of increasingly complex technological systems. In parallel, for dimensioning and, above all, for control system design purposes, the need of models of reduced complexity has emerged, together with the need of tools capable of extracting, from the overall dynamic model, reduced models, representing the dominant behaviour. The paper reviews recent results in nonlinear model order reduction, originally developed with reference to computer aided analysis and design of electronic circuits, and puts them in a control system design oriented perspective, pointing out interesting future research directions.

1 Introduction

Equation-based, object-oriented modelling techniques, languages, and tools emerged during the '90s, and are now well established in the modelling and simulation field, in particular to support the design of modern integrated systems, which require multi-physics modelling capabilities. Object-oriented models are usually non-linear, and mostly used for simulation purposes. There is a strong interest in object-oriented modelling for control system analys and design, but so far the use of such models to support control-related activites has been mostly limited to the verification of the control system performance in closed loop by simulation, rather than giving direct support to controller design [4]. For this specific purpose, the equations that correspond to object-oriented models built by aggregation of reusable components are usually too complex, and not formulated in a way that can be directly employed in the controller design.

In the field of electronic circuit analysis and design, a research interest has recently emerged in Model Order Reduction (MOR) techniques for nonlinear circuits. The main motivation behind this interest is to support hardware design activities. On one hand, as pointed out by [8] and [34], the design of modern mixed-signal (analog and digital) integrated circuits represent a challenge for modelling and simulation software, due to the ever-increasing complexity of the circuits allowed by the progress of integrated circuit manufacturing technology. In order to master the complexity, it is necessary to provide good macro-models of basic functional units, which are then used at a higher level for the design of the integrated system. These macro-models are usually developed manually by skilled designers, but this activity is very labour-intensive, and might lead to neglect parasitic and secondary phenomena which have a significant impact on the overall system performance, expecially as the features of the integrated circuits get smaller and smaller. This motivates the development of techniques to extract these macro-models automatically from finer-detail models of the hardware. On the other hand, as noted by [35], MOR techniques can help to highlight the influence of key design parameters of the hardware on the dynamic performance of the system, e.g., the frequency and damping of poles in the transfer function of the linearized model, thus providing valuable information for the optimization of the system design.

Similar issues also arise in the field of control system design, even though with some significant differences. The aim of this paper is then to review some of these recently developed MOR techniques, putting them in a control system perspective, and motivating the need for further research and extensions in order to make them more effective for control applications.

The paper is structured as follows. Section 2 discusses how object-oriented models could be used for direct support of control system design, with emphasis on the specific requirements of control-oriented models. In Section 3, several recently developed MOR techniques are reviewed, while Section 4 points out open issues and future research work which is needed to make these techniques more effective for control system design support. Section 5 concludes the paper with final remarks.

2 From object-oriented modelling to control system design

2.1 Object-oriented modelling of physical systems

The object-oriented approach to system-level, heterogeneous physical system modelling follows an a-causal, equation-based approach. Basic components, corresponding to elementary physical objects, or to specific phenomena going on in a physical object, are built by directly writing the differential and algebraic equations describing them as on the paper. The equations refer to both the internal variables and to the connector (or port) variables. Connectors represent the boundary conditions of the component, and usually contain flow/effort pairs of variables, such as current and voltage, with no implied causality (i.e., neither input nor output). The basic components can then be connected together through their ports to form aggregate models; the connection of one or more ports corresponds to adding connection equations to the system, stating that the sum of the flow variables is zero, and that the effort variables are equal. Where appropriate, it is also possible to define causal input and output connectors, e.g. for signal processing system. The complete system model, obtained by the connection of many basic components, thus corresponds to a system of differential-algebraic equations (DAEs), including the component equations and the connection equations, which is usually called the *flattened model*.

The Modelica language [21, 12] is the result of a cooperative effort by tool vendors, end users, and researchers, towards the development of standard, tool-independent object-oriented language for the modelling of physical and technological systems, with a strong emphasis on the development of libraries of reusable components, in order to reduce the model development time and effort. During the last decade, component libraries have been developed covering many fields of engineering, including:

- mechatronic systems;
- automotive systems;
- electrical and electronic circuits;
- electrical machinery;
- power hydraulics;
- thermo-fluid systems (cooling and air conditioning, heating, power generation);
- aircraft and space systems.

Many of these libraries, as well as their applications to specific projects, are documented in the proceedings of the International Modelica Conference series, which are available online on the Modelica Association webpage [23]. The availability of an ever-growing base of reusable modelling knowledge promotes model-based system design, by making it easier and faster.

Object-oriented modelling is particularly attractive to support the study of the control of integrated systems, which combine different technologies to achieve a certain goal. For example, the study of an aircraft landing gear requires the integration of the aircraft body model, of the multibody model of the gear links, of the power hydraulics actuators or of the electrically driven actuators, and of the control system, together in the same system model. Since object-oriented modelling techniques are based on completely general models described by DAEs, there is no particular problem in describing the dynamics of such integrated systems within a unified framework.

2.2 Mathematical features of typical object-oriented models

Modelica allows to describe hybrid systems, whose behaviour can be modelled by continuous time equations, combined with event-driven or discrete-time equations. This very general class of systems is too large for the scope of the following discussion, which will just focus on continuous-time, time-invariant models, which contain no discrete variable and no explicit reference to the time variable. The overall model is assumed to have one or more exogenous inputs, collected in the vector u and corresponding to the control signals and to the external disturbances acting on the system, as well as one or more output variables, collected in the vector y and corresponding to the sensor signals.

As noted above, the flattened system model corresponds to a set of nonlinear DAEs, which can be formulated as

$$F_0(x_0, \dot{x}_0, v_0, u, p_0) = 0, \tag{1}$$

where $F_0(\cdot)$ is a vector-valued function, x_0 is the vector of the dynamic variables, which also appear under derivative sign, v_0 the vector of the algebraic variables, which are not differentiated in the original model, and p the vector of the system parameters, which are constant throughout the time evolution of the system. The vector y of the system ouputs contains a subset of the elements of x_0, v_0.

It is often the case that the DAEs of an object-oriented system are of index greater than one; this happens every time some of the equations in (1) are purely algebraic constraints among dynamic variables. For example, the connection of two 3D rigid body models via a rotational joint in a closed kinematic loop, or the connection of two 1D rotational inertia models via an ideal gear introduce algebraic constraints among the two body coordinates, which appear differentiated in the equations of motion. In this case, it is standard practice in object-oriented tools to transform (1) into an index-one system, by means of Pantelides' algorithm and of the dummy derivative algorithm [22], which add new equations and variables by differentiating the algebraic constraints and substituting some of the derivatives with dummy algebraic variables. The same algorithm can also be used to change the state variables, when this is appropriate for some reason. The end result is a new set of index-one DAEs

$$F(x, \dot{x}, v, u, p) = 0, \tag{2}$$

where x is the new vector of the differentiated variables (which can now be called the system's states, as they can be assigned arbitrary initial values) and v is the new vector of algebraic variables. The outputs y are now selected from the elements of x, v.

Unfortunately, this model formulation is not directly suitable for the design of control systems, due to a number of reasons.

First of all, if the model has been built by connecting reusable components from an existing library, it will usually be much more detailed than needed for the effective synthesis of a control law. In other words, it will describe a lot of dynamic behaviour which is irrelevant or marginal for the purpose of control design, and that should therefore be eliminated from the model:

- fast dynamics in a frequency range well beyond the control system bandwith;
- dynamics which is unreachable and/or unobservable from the selected inputs and outputs;
- second-order effects with negligible impact on the input-output behaviour;
- description of the nonlinear system behaviour over a range of operating points which is wider than needed by the specified operating modes of the closed-loop system.

Second, the model is not formulated in standard state-space form, which is required by most control design techniques. Furthermore, the model typically has too many state variables: the complexity of typical control system design algorithm is $O(n^2), O(n^3)$, or more, n being the number of states, and most advanced design techniques will also run into numerical problems when fed a very high-dimensional model. Finally, solving the system equations might be too time-consuming, expecially when the model contains large algebraic loops and/or nonlinear function with large computation time (such as fluid property models in thermo-fluid systems).

It is therefore necessary to reduce the complexity of the model by eliminating all irrelevant details, and to cast the model in the form required by the control system synthesis algorithms.

2.3 Requirements of models for control system design

Generally speaking, there are different scenarios for the use of reduced-order models in control related activities. In the first scenario, all the parameters of the plant are known and fixed: the model can then be used for control design, and for robustness analysis against nonlinear effects. In all the other cases, a model which keeps the dependency from some key parameters is required.

In the second scenario, some of the parameters are uncertain, because first-principle modelling is difficult or impossible, but there are experimental data available, or simulation data from some more accurate reference model: in this case the model might be used for grey-box parameter identification ([17, 7]).

In the third scenario some parameters are design parameters of the process, which should be chosen in order to obtain a dynamics which is most favourable for control. Ideally, the model should provide a direct relationship between the design parameters and the dynamic features of the plant, like the damping or natural frequency of oscillation modes, or the presence of non-minimum phase behaviour, like in the case of right-half-plane zeros in the transfer function of the linearized model. If a simple, explicit relationship cannot be obtained, some kind of sensitivity analysis can also be very helpful to guide the design process.

Most control design techniques and algorithms require models in state-space form, i.e.

$$\begin{cases} \dot{x}(t) = f(x(t), u(t), p) \\ y(t) = g(x(t), u(t), p) \end{cases} \tag{3}$$

Ideally, explicit analytical expressions should be available for the two right-hand-side functions. If this is not possible, some numerical procedure should be provided to compute the functions, as well as their Jacobians with respect to all their inputs. For the design of nonlinear Model Predictive Controllers (MPC, see [26] for a comprehensive review), a discrete-time version of the state-space equations is usually required

$$\begin{cases} x(k+1) = f(x(k), u(k), p) \\ y(k) = g(x(k), u(k), p). \end{cases} \tag{4}$$

It is possible to obtain equations (4) from (3) by means of standard one-step numerical integration methods, such as forward Euler or Runge-Kutta. If explicit methods are used, one has to pay attention to numerical stability in the case of stiff systems; on the other hand, it is hoped that the model-order reduction procedure has already gotten rid of the fast states of the original model prior to the discretization step.

An increasing body of control-theoretic methods is being developed for special forms of the state-space equations (3). A first popular one is the so-called Linear Fractional Representation (LFR), or Linear Fractional Transformation (LFT), where nonlinearities and uncertain or design parameters are 'pulled out' of the model, which is then represented as the feedback connection between a linear, time invariant state-space model and two feedback blocks containing the nonlinear functions $\Theta(\omega)$ and the uncertain parameters δ_j (see Fig. 1)

$$\begin{cases} \dot{x} = Ax + B_1 w + B_2 \zeta + B_3 u \\ z = C_1 x + D_{11} w + D_{12} \zeta + D_{13} u \\ \omega = C_2 x + D_{21} w + D_{22} \zeta + D_{23} u \\ y = C_3 x + D_{31} w + D_{32} \zeta + D_{33} u \\ w = diag\left\{\delta_1 I_{r_1}, \ldots, \delta_q I_{r_q}\right\} z \\ \zeta = \Theta(\omega). \end{cases} \tag{5}$$

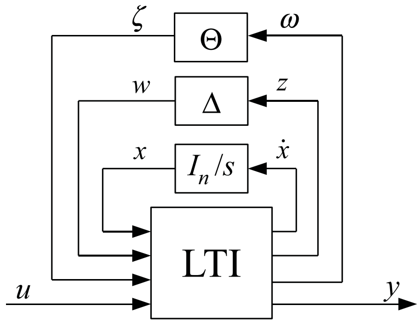

Figure 1: Block diagram of the considered LFT representation.

This formalism can be very useful both for robust control analysis [39] and for grey box parameter identification [14, 7]. Tools and algorithms are now being developed which allow to perform the transformation from (1) [6] or (3) [20] to (5) automatically.

Another popular formalism is that of Linear Parameter Varying systems, which is the basis for many gain scheduling control design techniques (see the survey papers [15, 30]). The system model must be formulated as

$$\begin{cases} \dot{x} = A(\rho(t))x(t) + B(\rho(t))u(t) \\ y = C(\rho(t))x(t) + D(\rho(t))u(t). \end{cases}$$

The system matrices depend on the time-varying vector of scheduling variables $\rho(t)$, which is assumed to be measurable, and thus known to the controller. The vector ρ may include some of the system states, as well as known functions of the system inputs, in which case the model is known as *quasi-LPV*. There might be different assumptions about the dependency of the system matrices from ρ, depending on the specific control design techniques. The most common ones are:

- Affine parameter dependence (LPV-A): $A(\rho) = A_0 + A_1\rho_1 + A_2\rho_2 + \ldots$, and similarly for B, C, and D. An extension to polynomial dependence can also be considered.
- Input-affine dependence (LPV-I): as above, but only for matrices B and D, while A and C are constant.
- LFT parameter dependence (LPV-LFT): the dependence on the scheduling parameters is described by a linear fractional transformation, such as in (5), with $\delta_j = \rho_j$.

The reduced-order model should eventually be cast in the above-mentioned formulations, with the minimum possible complexity which is compatible with a good performance of the closed-loop control systems. This can be either checked empirically *a posteriori* via closed-loop simulation with the full model, or formally proven *a priori* via robust control analysis techniques, in case it is possible to quantify the approximation inherent in the MOR phase in a way which is compatible with the robustness analysis technique.

For all possible kinds of controller, minimum complexity means first and foremost the lowest possible number of state variables, in order to keep the complexity of the controller design phases and of the implemented control law at a reasonable level. It might also mean that only the nonlinearities that are effectively influencing the system behaviour over the expected operating range are considered, in order to avoid unnecessarily conservative and/or complex designs. For this reason, local approximation techniques, which only try to approximate the system behaviour around specified reference trajectories, are definitely useful.

For most design techniques, where the bulk of the computation is done off-line during the controller synthesis phase, it is not essential that the reduced-order model is also fast to compute, as the design phase is done once and for all, and not in real time. A situation where, e.g., the computation of $f(x,u)$ might require the iterative solution of implicit equations might still be acceptable. On the other hand, models for MPC control require to compute the right-hand-side of the state-space equations many time at each time step by nonlinear optimization algorithms that must run in real time; in this case it is absolutely essential that the speed of computation is as fast as possible.

2.4 Strategies for control-oriented MOR of object-oriented models

The starting point for the generation of a reduced-order, control-oriented model from an object-oriented model is the set of index-1 DAEs (2) obtained from the O-O compiler. This model can be brought into standard state-space form (3) by solving it for the unknowns $\dot{x}, y$ through a suitable combination of symbolic and numeric techniques, which are routinely implemented in O-O tools to produce the simulation code. Two different strategies can then be followed to obtain a reduced-order model. The first option is to first bring the full DAE model into state-space form, and then apply MOR techniques for ODE models; the other option is to apply MOR techniques for DAE models to the full DAE model, and then bring the reduced-order DAE in state-space form at the end of the process. Techniques to support both strategies will be reviewed in the next section.

3 Model order reduction techniques for non linear systems

3.1 Trajectory PieceWise-Linear (TPWL) approximation

The non linear dynamical system considered in the Trajectory PieceWise-Linear (TPWL) approximation is given in the state-space form [31]:

$$\begin{cases} \dot{x} = f(x) + Bu, \\ y = Cx, \end{cases} \tag{6}$$

where $x \in R^n$ is the state vector, $f : R^n \longrightarrow R^n$ is a nonlinear vector-valued function, B in an $n \times m$ input matrix, $u \in R^m$ is the system input, C in a $p \times n$ output matrix and $y \in R^p$ is the output.

The TPWL technique is based on a quasi-piecewise-linear approximation of the function f, around s linearization points (states) x_i ($i = 0, \dots, (s-1)$):

$$f(x) \approx \sum_{i=0}^{s-1} \tilde{w}_i(x)(f(x_i) + A_i(x - x_i)), \tag{7}$$

where A_i are the Jacobians of f evaluated at states x_i and $\tilde{w}_i(x)$'s are state-dependent weights ($\sum_{i=0}^{s-1} \tilde{w}_i(x) = 1$, for all x). Linear model order reduction can be then applied to each linearized system:

$$\begin{cases} \dot{x} = f(x_i) + A_i(x - x_i) + Bu, \\ y = Cx, \end{cases} \tag{8}$$

through some classical techniques, based on the construction of an orthonormal basis $V = [v_1, \dots, v_q]$, with $q \ll n$, spanning the most significant dynamics. Thus, assuming $x = Vz$, where $z \in R^q$ is the reduced state-space vector, one obtains:

$$\begin{cases} \dot{z} = \left(\sum_{i=0}^{s-1} w_i(z)[V^T f(x_i) + V^T A_i(Vz - x_i)]\right) + V^T Bu, \\ y = CVz, \end{cases} \tag{9}$$

where $w_i(z)$'s are weights dependent on the reduced order state z ($\sum_{i=0}^{s-1} w_i(z) = 1$, for all z). These weights are computed based on the distances between the current state and the linearization points (see [32]), with a rapid transition from one linearization point to another, so that a single linearized model is actually dominant for most of the state space (in this sense (9) is a piecewise-linear model). For example, in [2] the following weighting scheme is proposed:

$$w_i(z) = \frac{\exp\left[\frac{-\beta d_i^2}{m^2}\right]}{\sum_j \exp\left[\frac{-\beta d_j^2}{m^2}\right]} \tag{10}$$

where β is a positive constant, $d_i = \|z - z_i\|$, with z_i being the projection of the linearization state x_i, and $m = \min_i(d_i)$.

As far as the choice of a suitable projection basis V is concerned, various techniques may be considered. For example, a set of Krylov subspaces could be determined for different linearized model and then merged together, or the projection could be computed according to a Truncated Balanced Realization (TBR) techinque, or a hybrid combination of both approaches ([24],[36],[38]). In this last case, a singular value decomposition on the projection matrix V is performed, and only vectors corresponding to the largest singular values are selected as the columns for a new reduced projection matrix. An alternative technique could be also considered to generate the projection matrix: the Proper Orthogonal Decomposition (POD) [27]. This technique is based on a snapshot of the system solution, obtained for example by simulation: $\bar{X} = [\ x(t_1) \quad x(t_2) \quad \dots \quad x(t_N)\ ]$ and computes a basis V minimizing the overall projection error $\|\bar{X} - VV^T\bar{X}\|$. The solution is hence given by the singular value decomposition (SVD) on the data collection $V = \mathrm{svd}(\bar{X})$, where V is made up by the dominant singular vectors.

The choice of the linearization points is also crucial. In order to limit the computational burden, while spanning the n-dimensional state space of the original model (6), the authors propose to generate a collection of models selected from a single training trajectory, corresponding to some relevant training input, by performing a simulation of the full order nonlinear system. Of course, the reduced-order model is accurate as long as its trajectories are not too far from the training one.

3.2 Piecewise-Linear parameterized model order reduction

In [31] the authors also propose a method for estimating *a posteriori* the approximation error, based on the estimation of the norm of the Hessian of function f, moreover, some results on the stability and passivity of the reduced-order model are derived. If the original nonlinear system is L_p-stable and the input signal is bounded, the difference δx between the state of the full nonlinear model and the state of the piecewise-linear approximated model is bounded, provided that all matrices VV^TA_i are symmetric and strictly stable or all matrices VV^TA_i are strictly diagonally dominant, with all diagonal elements being negative. Moreover, if 0 is an exponentially stable equilibrium point for system $\dot{x} = f(x)$, there exists a stability-preserving weighting procedure.

In [2] the TPWL technique is extended by considering a nonlinear dependence of system (6) from some parameters:

$$\begin{cases} \dot{x} = f(x,p) + B(p)u, \\ y = Cx, \end{cases} \tag{11}$$

where $p \in R^k$ is a vector of parameters then, by polynomial fitting or Taylor series approximation, the dependence of the parameters is extracted from the nonlinear functions:

$$\begin{aligned} f(x,p) &\approx \textstyle\sum_{j=0}^{k} g_j(p) f_j(x) \\ B(p)u &\approx \textstyle\sum_{j=0}^{k} g_j(p) B_j(x) u \end{aligned}$$

For example, a first-order Taylor series expansion on $f(x,p)$ in the neighborhood of a nominal value of parameters $\bar{p}$ yields

$$\begin{aligned} f_0(x) &= f(x,\bar{p}) - \textstyle\sum_{j=1}^{k} \bar{p}_j \left.\frac{\partial f(x,p)}{\partial p_j}\right|_{\bar{p}} \\ f_j(x) &= \left.\frac{\partial f(x,p)}{\partial p_j}\right|_{\bar{p}} \qquad (j \neq 0) \end{aligned}$$

By introducing a new set of parameters $\tilde{p}_i = g_i(p)$ is finally possible to obtain

$$\dot{x} = \sum_{j=0}^{k} \tilde{p}_j \left[f_j(x) + B_j u \right] \tag{12}$$

The TPWL technique may be now applied to (12) similarly to (7), while considering the parameter dependence of the system, obtaining

$$\begin{aligned} \dot{x} &= \textstyle\sum_{i=0}^{s-1} \sum_{j=0}^{k} w_i(x) \tilde{p}_j \left[A_{ij} x + k_{ij} + B_j u \right] \\ A_{ij} &= \left.\frac{\partial f_j(x)}{\partial x}\right|_{x_i} \\ k_{ij} &= f_j(x_i) - \left.\frac{\partial f_j(x)}{\partial x}\right|_{x_i} x_i \end{aligned}$$

Again, standard projection techniques can be applied to each linearized system in (3.2) obtaining

$$\begin{cases} \dot{z} = \sum_{i=0}^{s-1} \sum_{j=0}^{k} w_i(z) \tilde{p}_j \left[V^T A_{ij} V z + V^T k_{ij} + V^T B_j u \right] \\ y = CVz \end{cases}$$

(note that the constant vector k_{ij} may be treated as a second input vector B_2 with a constant input $u_2(t) = 1$).

Authors suggest in [2] to select the linearization points with additional trajectories with respect to the training inputs considered in the standard TPWL, relevant to a set of nominal points in the parameter space. In particular, training should be performed in regions where the system is more sensitive to each parameter (authors also show how to estimate the sensitivity of trajectories to parameter variations).

They also suggest to simulate a linearized model, instead of the full nonlinear system, in order to evaluate the training trajectories: once the current simulated state leaves the neighborhood of a linearization point, a new linearized model is created at the current state. It must be also pointed out that the additional trajectories relevant to parameter-space training increase the cost of construction the model but not its simulation, since the weighting functions are typically nonzero for few models at any given time.

3.3 Model order reduction based on PieceWise Polynomial (PWP) representation

Very recently, a new method has been proposed to overcome a limitation of TPWL methods [8], namely the fact that these methods result accurate for large-signal transient analyses, in that they capture strong nonlinearities well in a wide range, but may fail for small-signal distortion analysis. The reason is that when considering a trajectory close to a linearization point, the TPWL model reduces to a linear time-invariant system (LTI) and no distorsion can be captured. Essentially, the new method: PieceWise Polynomials (PWP), is based on a higher order (tensor) polynomial approximation rather than a simple linear one as in TPWL, capturing small signal distorsion well around the expansion point.

Again, s points x_i are considered in the state space ($i = 0,\ldots,(s-1)$) but, instead of a first-order expansion as in the TPWL, a higher-order expansion is performed, for the sake of simplicity a quadratic expansion will be considered:

$$\begin{cases} \dot{x} = f(x_i) + A_i(x - x_i) + A_{2,i}(x - x_i) \otimes (x - x_i) + B_i u, \\ y = Cx, \end{cases}$$

where matrix $A_{2,i}$ collect the second-order derivatives of $f(x)$ and the symbol $\otimes$ represents the Kronecker tensor product.

A projection basis V_i can be constructed for each expansion point by applying a weakly nonlinear MOR technique, and a uniform projection base $V \in R^{n\times q}$ is then generated through SVD on the collection of all basis $V = \mathrm{svd}\begin{bmatrix} V_0 & V_1 & \ldots & V_{s-1} \end{bmatrix}$. The final reduced-order PWP model is again obtained by a weighted combination of models:

$$\begin{cases} \dot{z} = \sum_{i=0}^{s-1} w_i(z)\left(\hat{f}(x_i) + \hat{A}_i(z - z_i) + \hat{A}_{2,i}(z - z_i) \otimes (z - z_i) + \hat{B}_i u\right) \\ y = C\left[\sum_{i=0}^{s-1} w_i(z)(x_i + V(z - z_i))\right] \end{cases}$$

where $z_i = V^T x_i$, $\hat{f}(x_i) = V^T f(x_i)$, $\hat{A}_i = V^T A_i V$, and $\hat{A}_{2,i} = V^T A_{2,i} V \otimes V$.

Some comments are now in order about the weakly nonlinear MOR techniques. Consider a polynomial expansion of $f(x)$ around a dc operating point:

$$\dot{x} = A_1 x + A_2 x \otimes x + A_3 x \otimes x \otimes x + \ldots + Bu \tag{13}$$

where A_i is the i-th order derivative. According to the Volterra theory [29] the solution of (13) is the sum of different responses:

$$x(t) = \sum_{i=1}^{\infty} x_i(t)$$

where each response $x_i(t)$ can be recursively computed by solving the same LTI with different inputs. For example, the first- through the third-order responses are the solutions of the following linear systems:

$$\dot{x}_1 = A_1 x_1 + Bu \tag{14}$$

$$\dot{x}_2 = A_1 x_2 + A_2 x_1 \otimes x_1 \tag{15}$$

$$\dot{x}_3 = A_1 x_3 + A_2(x_1 \otimes x_2 + x_2 \otimes x_1) + A_3 x_1 \otimes x_1 \otimes x_1 \tag{16}$$

The problem has been therefore recasted as a reduction of a series of LTI.

The simplest way to deal with the reduction of this form is to separately reduce each LTI [28], for example by Krylov subspace computation, by plugging the response of each system as an input to the following LTI in the series. For example, assume that the first-order LTI (14) is reduced by the projection base V_1, then the response can be approximated as $x_1(t) \approx V_1 z_1(t)$ and plugged into the second-order LTI (15) having $u_2(t) = z_1(t) \otimes z_1(t)$ as input

$$\dot{x}_2 = A_1 x_2 + Bu_2$$

for which another projection base V_2 can be determined, and so on. However, in this way the dimension of the projection base rapidly increases, resulting in inefficiently large reduce models. A more compact projection base V could be computed via SVD on the collection of separated basis [25], i.e. $V = \mathrm{svd}(\begin{bmatrix} V_1 & V_2 & \ldots \end{bmatrix})$ but the improvement is in general scarcely appreciable. The most efficient way to approach the problem is the Nonlinear Model Order Reduction Method (NORM)–Momentwise Projection [16], which obtains a very compact reduced model without loss of accuracy by removing some redundancies among the Krylov subspaces of each LTI system in the series. Anyway, in [8] an alternative to NORM is presented, namely the MPI method, which exploits the intrinsic correlation among the inputs to the LTI in series.

The PWP method is also endowed with an heuristic algorithm to choose expansion points in order to cover a wide range of the state space with limited expansion regions. Essentially, a new state is added to the expansion point set when the relative error between the current evaluation of $f(x)$ and its linearization exceeds a predefined error tolerance along the training trajectory. Another improvement in efficiency is obtained by merging regions from different trajectories, in order to maximize the state-space coverage. Thus, new points on new trajectories are added to the base set of expansion points only if the distance between the new points and the points already in the base set is greater that some predefined tolerances.

Finally, some limitations of the choice (10) for the weighting functions were pointed out in [8], namely the fact that sometimes the error is large when the current state is on the border of the space covered by the expansion points. Authors then propose the following expression for the weighting function:

$$w_i(z) = \left[\frac{d_{\min}}{d_i(z)} \exp\left(-\frac{d_i(z) - d_{\min}}{D_{\min}} \right) \right]^p$$

where $d_i(z) = |z - z_i|_2^2$, $d_{\min} = \min(d_i(z))$, $D_{\min}$ is the minimum distance among the base set points z_i, while parameter p is used to make the transition smoother or shaper when switching from one region to another. The whole weight function is finally normalized to satisfy $\sum_{i=0}^{s-1} w_i(z) = 1$.

3.4 Simulation-free nonlinear model order reduction

A rather different approach to nonlinear model order reduction, thus not based on linearization nor on simulation, has been proposed in [33]. The key idea of the method is to separate the original nonlinear model into a linear and a nonlinear part, and to consider the nonlinearities of the resulting system as additional inputs to the linear part.

The method considers a general nonlinear time-invariant system:

$$\begin{cases} \dot{x} = f(x,u) \\ y = Cx \end{cases}$$

which can be rewritten as

$$\begin{cases} \dot{x} = Ax + Bu + Fg(x,u) \\ y = Cx \end{cases} \tag{17}$$

where the vector $g(x,u)$ collects the nonlinear parts of the element of $f(x,u)$, in such a way that every nonlinear term appears only once and is free from any possible constant factor. In turn, by defining a new input matrix $B^* = \begin{bmatrix} B & F \end{bmatrix}$ and a new input $u^* = \begin{bmatrix} u & g(x,u) \end{bmatrix}^T$ model (17) can be rewritten in the form

$$\begin{cases} \dot{x} = Ax + B^* u^* \\ y = Cx \end{cases} \tag{18}$$

If u^* is actually considered as an exogenous input, which is not since it depends on the state x, the order of system (18) can be reduced through any MOR technique for LTI systems:

$$\begin{cases} \dot{z} = \hat{A}z + \hat{B}^* u^* \\ y = \hat{C}z \end{cases} \tag{19}$$

Finally, if an approximation $\hat{x}$ of the state vector is considered in $g(x,u)$, just derived from (19), thus $\hat{x} = Vz$ or, more generally, $\hat{x} = W_1 z + W_2 u$, the overall reduced order model can be defined by a feedback connection of (19) and $g(\hat{x},u)$:

$$\begin{cases} \dot{z} = \hat{A}z + \hat{B}u + \hat{F}g(Vz,u) \\ y = \hat{C}z \end{cases}$$

or

$$\begin{cases} \dot{z} = \hat{A}z + \hat{B}u + \hat{F}g(W_1 z + W_2 u, u) \\ y = \hat{C}z \end{cases}$$

Authors in [33] suggest the adoption of Eitelberg's order reduction method [9], which they extend in order to find suitable matrices W_1 and W_2. As in the original method, the solution is found by minimizing the error between the step responses of the full system and the reduced-order system, after defining a set of dominant state variables and imposing a null steady state error for the reduced linear model. Actually, a sub-optimal solution is computed, minimizing the time-dependent term of the error to find W_1 and vanishing the time-independent term by a suitable choice of W_2.

3.5 Analytical model order reduction based on symbolic analysis

An even more different approach is described in [37, 35], where the dominant system behavior is extracted by automated derivation of approximated symbolic formulas, by means of mixed symbolic (computer-algebra) and numerical strategies. One major advantage of this approach is that it can be applied to a very general nonlinear differential-algebraic (DAE) system of equations in the form (2).

A key aspect of the method is that a *symbolic* formulation of the DAE system is considered, in other words, a complete class of models with arbitrary parameter values. Another advantage is that the accuracy of the reduced-order model can be predefined and automatically checked, and the result of the simplification maintains a physical interpretation. The method is original from the field of the analog electronic circuits design, but it can be extended to general equation-based, acausal modelling environments.

The equation-based approximation process can be sketched as follows:

1. A general symbolic DAE system and a list of numerical reference values are first generated.
2. Based on the said reference values, the system of equations is evaluated and numerically solved for some variables of interest.
3. The results are compared with a simulation, to ensure that the equations are correct and the symbolic model reduction can start.
4. Since the model reduction consists in a sequence of symbolical simplification steps, it is fundamental to predict the influence on the error with respect to an objective function a simplification step causes. This prediction is performed by a *ranking* algorithm that orders simplification steps according to the error generated [13]. One way is to iteratively applying each simplification step to the original system and carrying out one single Newton step starting from the reference solution, the corresponding deviations set up the ranking order.
5. The next step applies the simplification (or a cluster of simplifications having the same order of magnitude in the ranking), the accumulated error is calculated and compared with the given error bound. If the error bound is exceeded the algorithm terminates returning the approximated system, otherwise the next simplification steps from the ranking list are carried out.

Several simplification strategies may be applied: algebraic simplification (variable elimination and decoupling of blocks), branch simplification (deletion of branches of piecewise-defined solution), switch simplification (fixing of values of switch variables), term substitution (replacement of variables with mean values obtained in reference simulations), term deletion (removal of terms in equations).

One important point to remark is the fact that simplifications may increase the index of the DAE system (2), thus preventing the numerical solution with standard solvers [1]. As a consequence, an index observer must be integrated in the simplification algorithm, in order to monitor and avoid index increase.

4 Extensions and future research directions

The MOR methods described in this survey are generally applied to the design of integrated micro-electro-mechanical systems (MEMS) and electronic circuits. Within this application domain the objective is to obtain reduced models to support the integrated system design through virtual prototyping techniques. This kind of models can contain strongly nonlinear components, but the external inputs to the system - typically current and voltage generators - appear as linear terms. This consideration results into a simplification of the mathematical formulation of the dynamic model, that can be represented as a ODE system which is linear in the inputs (6). When taking into account generic models of multi-domain dynamic systems, this formulation might not be general enough. It would be then interesting to extend the TPWL techniques to the more generic nonlinear systems (3), which could be obtained after index reduction and reformulation as ODEs.

Another issue with existing TPWL techniques is that they usually employ a Krylov-based approach for the reduction of the linearized models, which requires those models to have no poles in the origin. This is not a problem with typical MEMS and electronic circuits, which always have finite DC gains, but might be an issue with generic object-oriented models that contain pure integrators. In this case, different methods for the local approximation of the linearized systems should be investigated.

The methods described in Section 3.5, although initially developed for electronic circuits, have already found some applications in other domains, as shown in [35]. It would be interesting to test those methods with a wider range of multi-physics model, obtained from object-oriented models written in Modelica. It might as well be interesting to embed index reduction techniques in these methods, in order to allow an even more aggressive order reduction.

It will also be necessary to investigate how well different MOR techniques can match the specific formalisms required by the different control design techniques, both from a theoretical point of view and through testing on specific applications.

Another promising research direction could be to explicitly consider the information that is available on the topological representation of the dynamic system before obtaining the abstract mathematical representation (as DAE or ODE). Several works are present in literature considering mechanical applications. In particular, an "Energy-Based Model Reduction" is proposed in [18, 19], using the bond graph approach. Even if this could lead to some advantages, the limitation of the proposed methods is quite restrictive, since it is only possible to use bond graph connectors when creating a model. An improvement in the direction of supporting more general object-oriented models could be to extend these techniques considering also more general connectors.

Obtaining effective results in this field would require the integration of different tools (Modelica compilers, symbolic manipulation tool, numerical tools), possibly developed within different communities. More in general, a closer interaction between research groups working in different areas, like electronic circuit design, modelling and simulation, and control theory, would be helpful in order to obtain better results for everyone.

5 Conclusions

Object-oriented modelling methodologies, and in particular the Modelica language, are emerging as a tool for system design, but their use is currently mostly limited to simulation activities. On the other hand, the control system (and thus its design) is a key part of all modern integrated systems.

The models which are required for the control system synthesis have rather stringent requirements in terms of simplicity and of specific mathematical structures, which are discussed in this paper. In particular, MOR is essential to obtain models which are simple enough to be directly used for that purpose. This activity is nowadays performed manually, and there is thus a clear need of methods and tools to perform this task automatically, starting from object-oriented descriptions of the plant.

Several MOR techniques, which have been recently introduced by the electronic circuit design community, have been reviewed in this paper. Much work remains to be done in order to adapt and possibly extend them in order to work satisfactorily when dealing with generic object-oriented models; some possible research lines have then been proposed in this direction.

6 References

[1] Ascher U. M., and Petzold L. R.: *Computer Methods for Ordinary Diferential Equations and Differential-Algebraic Equations*. SIAM, Philadelphia, (1998).

[2] Bond B. N., and Daniel L.: *A Piecewise-Linear Moment-Matching Approach to Parameterized Model-Order Reduction for Highly Nonlinear Systems*. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 26, 12, (2007), 2216–2129.

[3] Casella F., Donida F., and Lovera M.: *Beyond Simulation: Computer Aided Control System Design Using Equation-Based Object Oriented Modelling for the Next Decade*. In: Proc. 2nd International Workshop on Equation-Based Object-Oriented Languages and Tools, July 8, 2008, Paphos, Cyprus, 35–45.

[4] Casella F., Donida F., and Lovera M. "Beyond simulation: Computer aided control system design using equation-based object oriented modelling for the next decade," in *Proceedings 2nd International Workshop on Equation-Based Object-Oriented Languages and Tools*, Paphos, Cyprus, Jul 8 2008. Available: http://www.ep.liu.se/ecp/029/005/

[5] Casella F., Donida F., and Lovera M.: *Automatic Generation of LFTs from Object-oriented Modelica Models*. In: Proc. 2008 IEEE Multi-conference on Systems and Control, September 3-5, 2008, San Antonio, Texas, USA, ThCC.11.

[6] Casella F., Donida F., and Lovera M. Automatic Generation of LFTs from object-oriented non-linear models with uncertain parameters. In *Proceedings of MATHMOD 2009*, Vienna, Austria, Feb 11–13, 2009.

[7] Casella F. and Lovera M. LPV/LFT modelling and identification: overview, synergies and a case study. In *Proceedings of the 2nd IEEE Multi-conference on Systems and Control*, San Antonio, USA, Sep 3–5, 2008.

[8] Dong N., and Roychowdhury J.: *General-Purpose Nonlinear Model-Order Reduction Using Piecewise-Polynomial Representations*. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 27, 2, (2008), 249–264.

[9] Eitelberg E.: *Model Reduction by Minimizing the Weighted Equation Error*. International Journal of Control, 34, (1981), 1113–1123.

[10] Feng L.: *Review of Model Order Reduction Methods for Numerical Simulation of Nonlinear Circuits*. Applied Mathematics and Computation, 167, (2005), 576–591.

[11] Feng L.: *Parameter Independent Model Order Reduction*. Mathematics and Computers in Simulation, 68, (2005), 221–234.

[12] Fritzson P., *Principles of Object-Oriented Modeling and Simulation with Modelica 2.1*. Wiley, 2003.

[13] Halfmann T., and Wichmann T.: *Symbolic Methods in Industrial Analog Circuit Design*. in: Anile A. M., Ali G., Mascali G. Eds., Scientific Computing in Electrical Engineering, Springer, (2006).

[14] Hsu K., Vincent T., Wolodkin G., Rangan S., and Poolla K. An LFT approach to parameter estimation. *Automatica*, 44(12):3087–3092, 2008.

[15] Leith D. J. and Leithead W. E., "Survey of gain-scheduling analysis and design," *International Journal of Control*, vol. 73, no. 11, pp. 1001–1025, 2000.

[16] Li P., and Pileggi L. T.: *NORM: Compact Model Order Reduction of Weakly Nonlinear Systems*. In: Proc. IEEE Design Automation Conference, 2003, 472–477.

[17] Ljung L. Perspectives on system identification. In *2008 IFAC World Congress, Soul, South Korea*, 2008.

[18] Loucas L. S. and Stein J. L. *Energy-based model reduction of linear systems*. In Proc. of the 4th International Conference on Bond Graph Modeling and Simulation ICBGM'99, San Francisco, CA, USA, Jan 17–20, 1999.

[19] Loucas L. S. and Yildir U.B.: *Modelling and reduction techniques for studies of integrated hybrid vehicle systems*. Mathematical and Computer Modelling of Dynamical Systems, 12, 2-3, pp. 203-218, 2006.

[20] Magni J.F. User manual of the Linear Fractional Representation Toolbox, version 2.0. Technical Report TR 5/10403.01F DCSD, ONERA, Feb 2006. URL: http://www.cert.fr/dcsd/idco/perso/Magni/download/lfrt_manual_v20.pdf.

[21] Mattsson S. E., Elmqvist H., and Otter M.. Physical system modeling with Modelica. *Control Engineering Practice*, 6(4):501–510, 1998.

[22] Mattsson S. E. and Söderlind G. Index reduction in differential-algebraic equations using dummy derivatives. *SIAM Journal on Scientific Computing*, 14(3):677–692, 1993.

[23] The Modelica Association, Modelica Home Page, `http://www.modelica.org/`.

[24] Muscato, G.: *Parametric Generalized Singular Perturbation Approximation for Model Order Reduction*. IEEE Transactions on Automatic Control, 45, 2, (2000), 339–343.

[25] Phillips J.: *Projection Frameworks for Model Reduction of Weakly Nonlinear Systems*. In: Proc. IEEE Design Automation Conference, June 2000, 184–189.

[26] Qin S. J. and Badgwell T. A. A survey of industrial model predictive control technology. *Control Engineering Practice*, 11:733–764, 2003.

[27] Rathinam, M., and Petzold, L.: *A New Look at Proper Orthogonal Decomposition*. SIAM Journal of Numerical Analysis, 41, 5, (2003), 1893–1925.

[28] Roychowdhury J.: *Reduced-Order Modelling of Linear Time Varying Systems*. In: Proc. International Conference on Computed-Aided Design, November 1998, 92–95.

[29] Rugh, W.: *Nonlinear System Theory–The Volterra Wiener Approach*. Baltimore, MD: Johns Hopkins University Press, (1981).

[30] Rugh, W. and Shamma J.: "Research on gain scheduling," *Automatica*, vol. 36, no. 10, pp. 1401–1425, 2000.

[31] Rewieński M., and White J.: *Model Order Reduction for Nonlinear Dynamical Systems Based on Trajectory Piecewise-Linear Approximations*. Linear Algebra and its Applications, 415, (2006), 426–454.

[32] Rewieński M., and White J.: *A Trajectory Piecewise-Linear Approach to Model Order Reduction and Fast Simulation of Nonlinear Circuits and Micromachined Devices*. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 22, 2, (2002), 155–170.

[33] Salimbahrami B., and Lohmann B.: *A Simulation-free Nonlinear Model Order-reduction Approach and Comparison Study*. Mathematical and Computer Modeling of Dynamical Systems, 10, 3–4, (2004), 317–329.

[34] Schwarz P., Bastian J., Clauss C., Haase J., Köhler A., Otte G., and Schneider P.: *A Tool-Box Approach to Computer-Aided Generation of Reduced-Order Models*. In: Proc. EUROSIM 2007, September 9–13, 2007, Ljubljana, Slovenia.

[35] Sommer R., Halfmann T., and Broz J.: *Automated Behavioral Modeling and Analytical Model-Order Reduction by Application of Symbolic Circuit Analysis for Multi-Physical Systems*. Simulation Modelling Practice and Theory, 16, (2008), 1024–1039.

[36] Shi, G., and Shi, C. J. R.: *Model-Order Reduction by Dominant Subspace Projection: Error Bound, Subspace Computation, and Circuit Applications*. IEEE Transactions on Circuits and Systems, 52, 5, (2005), 975–993.

[37] Wichmann R., Popp R., Hartong W., Hedrich L. *On the Simplification of Nonlinear DAE Systems in Analog Circuit Design*. In: Proc. of the Second Workshop on Computer Algebra in Scientific Computing CASC '99, Munich, Germany, May 31–Jun 4, 1999.
Reduced-Order Modelling of Linear Time Varying Systems. In: Proc. International Conference on Computed-Aided Design, November 1998, 92–95.

[38] Wong, N., Balakrishnan, V., Koh, C., and Ng, T.: *Two Algorithms for Fast and Accurate Passivity-Preserving Model Order Reduction*. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 25, 10, (2006), 975–993.

[39] Zhou K., Doyle J. U., and Glover K. *Robust and optimal control*. Prentice-Hall, New Jersey, 1996.