Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

Model separability indices for efficient dynamic simulation

A.V. Papadopoulos * **F. Casella** ** **A. Leva** **

* *Lund University, Department of Automatic Control, Lund, Sweden (e-mail: alessandro.papadopoulos@control.lth.se)*
** *Politecnico di Milano, Dipartimento di Elettronica, Informazione e Bioingegneria, Milano, Italy (e-mail: {francesco.casella,alberto.leva}@polimi.it)*

Abstract: This paper proposes a means to quantify the aptitude of a dynamic model to be partitioned into weakly coupled submodels. The proposal applies to both linear and nonlinear systems, is largely independent of their scale, and requires information that is easy to provide on the part of the analyst. The presented indices can be exploited in different manners, from easing numerical integration on monolithic solution platforms to tailoring distributed or co-simulation ones based on the particular characteristics of the model at hand. Examples are reported to show the usefulness and the practical viability of the proposal.

Keywords: Dynamic modelling, efficiency enhancement, numerical simulation, distributed simulation, nonlinear models, large-scale systems.

1. INTRODUCTION AND RELATED WORK

Efficient dynamic simulation is becoming more and more important for all engineering studies. Modern tools allow to construct and manage very complex models also on lightweight platforms, such as laptops. As such, however, the computational capabilities of the target platform often becomes the bottleneck, especially if simulation tools need bringing to the plant floor (e.g., to streamline and facilitate commissioning).

Different approaches can be taken to cope with the aforementioned complexity. The most common is the adoption of Model Order Reduction (MOR) techniques [Antoulas, 2005, Donida et al., 2010], which are widely studied but are well-established only in the linear case. When dealing with nonlinear systems only problem-specific extensions are available, essentially based on linearisation or Taylor expansion, bilinearisation, or functional Volterra series expansion, followed by a suitable projection (see, e.g., Innocent et al. [2003]).

Another interesting approach is, for example, the Transmission Line Modelling (TLM) presented in Sjölund et al. [2010]. TLM is based on modelling the propagation of a signal which is limited by the time it takes to travel across a medium. By utilizing this information it is possible to partition the system into independent blocks that may be simulated in parallel. This leads to improved simulation efficiency since it enables full performance of multi-core CPUs. This, how ever requires that the modeller explicitly introduces the transmission model, i.e., the decoupling part, by introducing some additional components, based on his/her intuition.

Schiela and Olsson [2000] propose an eigenvalue-based technique, more similar to the one related to this research, where a structural analysis is performed aimed at splitting the system into two subsystem so as to use a mixed-mode integration method to improve simulation efficiency. This is natural for linear systems, but the eigenvalue analysis hinders extensions to the nonlinear case.

This paper is part of a long-term research path aimed at endowing modelling and simulation tools with the capability of automatically introduce suitable approximation in the solution of complex nonlinear models, so as to achieve the needed simulation efficiency, and above all, in as transparent a manner as possible for the user. In the previous work Papadopoulos et al. [2013] we introduced an analysis technique to partition a system based on the time scale of its dynamics, and a way to exploit that partition by mixed-mode integration. Here we propose other technical means to exploit the same information, and we define some separability indices to quantify the keenness of a system to be partitioned.

2. BACKGROUND

In this section we summarise some background concepts and briefly review previous results [Papadopoulos et al., 2013], to contextualise the additional contributions of this paper in the overall research path.

The main idea is to enhance simulation efficiency by partitioning the dynamic model at hand, based on the time scale of the contained dynamics. Peculiar to our research is however the analysis technique used for that purpose, named *Cycle analysis* (CA), and described below. The so obtained system partition can be exploited by mixed-mode (i.e., implicit-explicit) integration, as suggested, e.g., in Schiela and Olsson [2000], or taken as the basis for further quantifications of the model "separability", which is the specific subject of this paper.

2.1 Cycle analysis

The main idea of CA is that in a causal ODE model, both each variable and each equation can be associated with a characteristic time scale. To this end, consider the state-space form of a continuous-time system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \tag{1}$$

⋆ A.V. Papadopoulos is member of the LCCC Linnaeus Center at Lund University.

that discretised with the Explicit Euler (EE) method with integration step h yields

$$\mathbf{x}_{k+1} = \mathbf{x}_k + h \cdot \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k). \tag{2}$$

Suppose now that (2) is at an asymptotically stable equilibrium. If a small perturbation is applied to a single state variable x_k, a transient occurs, and either the same perturbation affects the other state variables, without in turn re-affecting x_k, or the same perturbation, after some integration steps, re-affects x_k.

In the first case, no numerical instability can be introduced by the numerical integration process, but in the second case there is a "dependency cycle" among some state variables that may lead to unstable behaviour of the integration algorithm (intuitively, if along the dependency cycle the perturbation is amplified). CA detects the dependency cycles in the system, and defines conditions under which the perturbation cannot lead to numerical instability.

The first step in CA is to build the dependency digraph (or directed graph) $G = (N, E)$ associated with the model. Said digraph has a node $n \in N$ for each state, while the set of edges is formed as

$$E = I + h\frac{\partial \mathbf{f}}{\partial \mathbf{x}}$$

where h is the integration step and the $\partial \mathbf{f}/\partial \mathbf{x}$ is the Jacobian of the continuous-time system. In other words, the Jacobian of (2) is the adjacency matrix of the weighted digraph, and accounts for the propagation entity of the disturbance from the i-th variable to the j-th one.

The second CA step is to detect the set $\mathcal{C}$ of all cycles in the digraph. For every cycle $c \in \mathcal{C}$ detected in G, the *cycle gain* is then computed.

Definition 1. A *cycle gain* $\mu_c(h)$ of a cycle $c \in \mathcal{C}$ is

$$\mu_c(h) = \prod_{x_i, x_j \in c} e_{i,j} = \begin{cases} 1 + h\dfrac{\partial f_i}{\partial x_i} & \text{if } L = 1 \\ h^L \cdot \prod_{x_i, x_j \in c} \dfrac{\partial f_i}{\partial x_j} & \text{if } L > 1 \end{cases}$$

where $e_{i,j}$ are the edges involved in the cycles and L is the length of the cycle.

The cycle gain quantifies how much the disturbance given to a single state variable $x_i \in c$ is amplified along one cycle c. Apparently, by suitably constraining this quantity, we ensure that no numerical unstable behaviours can occur. Starting from the computed $\mu_c(h)$, inequalities of the form $|\mu_c(h)| \le \alpha$ can be set *for each cycle*, yielding

$$\begin{cases} 0 < h \le (1+\alpha)\left|\dfrac{\partial f_i}{\partial x_i}\right|^{-1} & \text{if } L = 1 \text{ and } \dfrac{\partial f_i}{\partial x_i} < 0 \\ 0 < h \le \sqrt[L]{\alpha} \cdot \left|\prod_{x_i, x_j \in c} \dfrac{\partial f_i}{\partial x_j}\right|^{-\frac{1}{L}} & \text{otherwise.} \end{cases} \tag{3}$$

where α is an upper bound on the allowed amplification of the disturbance entering the cycle. As a result, every cycle $c \in \mathcal{C}$ has been associated with a constraint on h, i.e., with an upper bound on the integration step which allows the perturbation not to be amplified along c.

Finally, each variable x_i is associated with the most restrictive constraint h_{x_i} on h among the set of cycles $\mathfrak{C}_{x_i} = \{c \in \mathcal{C} | x_i \in c\}$. Formally,

$$\begin{aligned} &\text{maximise} \quad h_{x_i} \\ &\text{subject to} \quad |\mu_c(h)| \le \alpha, \quad \forall c \in \mathfrak{C}_{x_i}. \end{aligned} \tag{4}$$

The result of CA is that each dynamic variable is associated with an upper bound of the integration step needed for a numerically stable integration, thus with a quantity related to its time scale. As a result, the variables can be ordered, for example, by increasing value of $\overline{h}_i$ and presented to the modeller, who can decide how to cut the model. For further details on CA, the interested reader is referred to Papadopoulos and Leva [2014].

2.2 Exploiting CA by integration method

CA can be exploited to improve simulation efficiency by clustering the dynamic variables by time scale. One can then use explicit integration methods for the slow ones, and implicit methods for the fast ones. This implies losing a precise representation of fast phenomena, but on the other hand it improves efficiency. To exemplify, consider the generic nonlinear ODE system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) \tag{5}$$

and assume it to be partitioned into two subsystem: one with slow dynamics, the other with fast dynamics. Following an approach similar to the one presented in Schiela and Olsson [2000], we can left-multiply the state vector by a projection matrix $P = \operatorname{diag}\{p_1, p_2, \ldots, p_n\}$, with $p_i \in \{0, 1\}$ to select the slow part, and by $\overline{P} = I - P$ to select the fast part. Therefore (5) can be written as

$$\begin{cases} \dot{\mathbf{x}}^S = P\dot{\mathbf{x}} = P\mathbf{f}\left(\mathbf{x}^S, \mathbf{x}^F\right) \\ \dot{\mathbf{x}}^F = \overline{P}\dot{\mathbf{x}} = \overline{P}\mathbf{f}\left(\mathbf{x}^S, \mathbf{x}^F\right) \end{cases} \tag{6}$$

where $\mathbf{x}^S$ represents the slow variables, and $\mathbf{x}^F$ the fast ones. In this work we use Explicit Euler (EE) for the slow part, and Implicit Euler (IE) for the fast part, but the presented concepts are totally general.

Using those methods, equation (6) can be expressed as

$$\begin{aligned} \mathbf{x}^S_{k+1} &= P\mathbf{x}_{k+1} = P\mathbf{x}_k + hP\mathbf{f}\left(\mathbf{x}^S_k, \mathbf{x}^F_k, t_k\right) \\ \mathbf{x}^F_{k+1} &= \overline{P}\mathbf{x}_{k+1} = \overline{P}\mathbf{x}_k + h\overline{P}\mathbf{f}\left(\mathbf{x}^S_{k+1}, \mathbf{x}^F_{k+1}, t_k\right), \end{aligned}$$

which in the linear case becomes

$$\begin{aligned} \mathbf{x}^S_{k+1} &= P\mathbf{x}_k + hPA\mathbf{x}_k \\ \mathbf{x}^F_{k+1} &= \overline{P}\mathbf{x}_k + h\overline{P}A\mathbf{x}_{k+1}. \end{aligned}$$

Composing those two equations, and solving for $\mathbf{x}_{k+1}$, we can obtain

$$\mathbf{x}_{k+1} = \left(I - h\left(I - P\right)A\right)^2 \left(I + hPA\right)\mathbf{x}_k.$$

Alternatively to the proposed techniques, other couples of Explicit-Implicit Runge-Kutta methods can be used, for example all the ones proposed in Schiela and Olsson [2000], Ascher et al. [1997]. In principle one could also use completely different methods, which however we do not discuss here owing to the predominant diffusion of Runge-Kutta ones.

Summarising, exploiting CA by integration method leads to join the best of implicit and explicit integration in a knowledgeable manner for the case under question. The resulting integration scheme is represented in Figure 1.

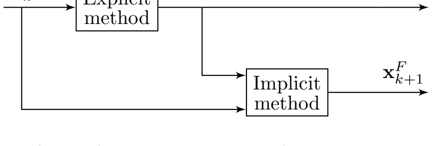

Fig. 1. Mixed-mode integration scheme.

3. EXPLOITING CA BY SIMULATION ARCHITECTURE

Broadly speaking, we can say that CA can be exploited along two fundamental axes. One – already treated – refers to the used integration methods, the other – which is the novel contribution of this work – to the adopted simulation architecture.

To enhance simulation efficiency, it is useful to identify which parts of a model can be simulated in parallel. To this end, the dependency digraph used for CA can be further exploited by detecting Parallelisable Cycle Sets (PCS), as briefly explained in this section.

A PCS is defined in the simplest manner as a set of cycles in the digraph that share a single node and have no other nodes in common. Extensions can be given considering *sets* of common nodes instead of a single one, or "weak" absence of other common nodes, for example based on the dominance of some cycle gains over others, but these are not necessary for the purpose of this section and cannot be treated in this work due to space limitations. The interested reader can refer to Fortunato [2010] for some details on how to formally define and detect PCS-like structures – usually defined as community in the network analysis theory – on the same digraph used here for CA.

The key idea motivating the search for PCS is that it is not infrequent to encounter situations in which fast parts of an overall model are made mutually dependent only by slower ones. This happens, for example, when several heat networks are connected to a large central energy storage. Another similar situation is when the presence of some controls eliminates high-frequency variabilities and thus confines the coupling of some parts of the model to low frequency only. This could be the case when branches of a grid are connected to a central strong node, which is tightly controlled. A possible example of PCS as seen on the model digraph is shown in Figure 2. If the model parameters actually make the PCS emerge, node C would be the common one, and the four subsystems corresponding to the cycles in the PCS would be composed of nodes {T}, {R}, {B}, and {L, LT, LB}.

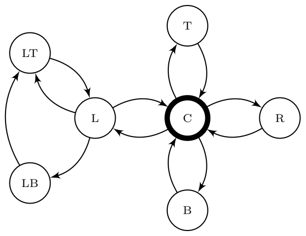

Fig. 2. An example of PCS as seen on the model digraph.

The usefulness of PCS comes by simply observing that they evidence situations like those just mentioned, and that in such cases the fast parts of the system can not only be dynamically decoupled from the slow ones, but also simulated in parallel. Furthermore, given the variety of the encountered time scales, the same model can give rise to different partitions into parallelisable models, depending on how the modeller chooses to split the time scales. Based on this idea, PCS can be exploited in at least two ways.

First, they can be detected on the entire digraph, i.e., before possibly selecting the time scale splits. Even in the case of a monolithic solution, and independently of the integration method, doing so provides an automatic selection of which parts of the system can be parallelised, e.g., by acting as discussed in Casella [2013].

Second, one can perform CA as described in the previous sections, and then detect PCS only for those parts for which implicit methods are to be used, so as to combine the improvements coming from DD with an efficiency enhancement of the most computationally intensive part of the simulation code.

From a more technological standpoint, one can then just employ parallel computing architectures, or even use the so obtained information to structure a co-simulation setup. In the latter case, the proposed technique provides more formally grounded an alternative to heuristics based e.g. on the minimisation of the number of signals exchanged among the co-simulation units [Kernighan and Lin, 1970, Hendrickson and Devine, 2000]. Of course such optimisations are not possible when the structure of the simulation setup is dictated by the used software tools, but in the last years formalisms and standards have been emerging to provide designers with more freedom in this respect, see e.g., Andersson et al. [2011], Blochwitz et al. [2012], Papadopoulos and Leva [2013].

4. SEPARABILITY INDICES

The result of CA is to associate each dynamic variable with an upper bound of the integration step, thus with a quantity related to its time-scale. The variables can then be ordered – and possibly clustered – by increasing value of $\overline{h}_{x_i}$. Based on this, some synthetic indices will now be defined, useful for deciding how to partition the original model in weakly coupled submodels. It will also be shown how such indices extend the idea of "stiffness", like CA was shown to evidence more decoupling-related information than eigenvalue analysis.

4.1 Measuring stiffness

To start, consider the classical stiffness indicator based on eigenvalues analysis, i.e., the *stiffness ratio.*

Definition 2. (Stiffness ratio). The *stiffness ratio* σ_R as defined in Cellier and Kofman [2006] is defined as the ratio between the absolute largest real part and the absolute smallest real part of any eigenvalue of the Jacobian of (1), i.e.,

$$\sigma_R = \frac{\max_i |\Re\{\lambda_i\}|}{\min_i |\Re\{\lambda_i\}|}.$$

It is worth saying that highly stiff systems are associated with high values of σ_R.

Apparently, the stiffness ratio is defined for a linear (or linearised) system, and indicates how much the smaller time scale differs from the larger one. It is thus a good index for understanding whether or not to use an integration method for stiff systems on the entire model, but gives no information about how many "clusters of time scales" are present in it, nor about which dynamic variable belongs to which cluster.

To exemplify, let us limit to the linear case, and consider Figure 3. In the left graph, the continuous-time eigenvalues of the system (indicated with crosses) are not equally spaced in the left-half-plane, and can be divided into two clusters: those that are close to the origin are associated with "slow dynamics", while the others are associated with "fast dynamics". The presence of the two different time scales is also evidenced by computing the stiffness ratio of Definition 2. Let us now consider the right graph of the same figure. In this case, the stiffness ratio is the same, since the closest and the farthest eigenvalues from the origin are the same, while the eigenvalues of the system are almost equally distributed in the left-half-plane. This feature of the system is strictly related to how much the system can be "separable" and is not evidenced in any way by the stiffness ratio.

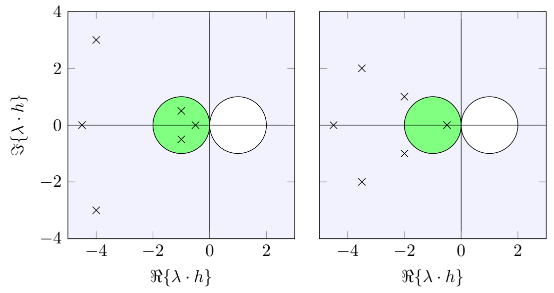

Fig. 3. Two linear systems with the same stiffness ratio.

Coming back to the CA approach, two different indices based it can be defined. One (the *stiffness index*, see Definition 3) quantifies the span of the time scales in the model, analogously to the one of Definition 2. The other (the *separability index*, see Definition 5) indicates to what extent the clusters of dynamic variables corresponding to those time scales the system can be computed in a decoupled manner. Both indices are function of α as indicated in (3), and being based on CA, they can be computed also for nonlinear systems.

Denote by $\mathcal{H}$ the set of integration steps h_{x_i} associated with each dynamic variable as in (4), and assume $\mathcal{H}$ ordered by ascending values of h, i.e., $\mathcal{H} = \{h_1 \leq h_2 \leq \ldots \leq h_N\}$.

Based on that, the following definitions can be given.

Definition 3. (Stiffness index). The *stiffness index* for a given α is the ratio between the minimum and the maximum integration step found with the cycle analysis, i.e.,

$$\sigma(\alpha) = \frac{h_{\max}(\alpha)}{h_{\min}(\alpha)}. \tag{7}$$

4.2 Measuring separability

Analogously to the stiffness ratio σ_R, also for the stiffness index highly stiff systems are associated with high values of σ.

Definition 4. (Separability term). The *separability term* for a given α, and for a given couple of variables x_i and x_j is

$$s_\alpha(i,j) = \frac{|h_i(\alpha) - h_j(\alpha)|}{\max_m (h_{m+1}(\alpha) - h_m(\alpha))}, \qquad h_i, h_j \in \mathcal{H}.$$

Definition 5. (Separability index). The *separability index* for a given α is the unity minus the ratio between the maximum and the average difference among two subsequent values of the time scales, i.e.,

$$\begin{aligned} s(\alpha) &= 1 - \frac{\frac{1}{N-1} \sum_{i=1}^{N-1} h_{i+1}(\alpha) - h_i(\alpha)}{\max_i (h_{i+1}(\alpha) - h_i(\alpha))} \\ &= 1 - \frac{1}{N-1} \sum_{i=1}^{N-1} s_\alpha(i+1,i). \end{aligned}$$

Apparently, high values of $s(\alpha) \in (0,1)$ indicate that the time scales involved in the system are different enough to be effectively separated.

In the following, the presented indices will be used to evaluate the level of stiffness and separability of the considered examples. Summarising, the stiffness ratio and index are comparable and synthetic descriptions of the separation between the maximum and the minimum model time scales, not suited however for understanding whether said model can be partitioned. The separability index is another synthetic one, but is specifically targeted at quantifying the possibility of such a separation. The separability term is a local index to a couple of adjacent time scales, and an analysis of its behaviour can easily suggest possible separation points.

4.3 Separability analysis

The proposed separability index (5) is a synthetic description of a structural property of the overall system, but additional information can be extracted from CA, providing also suggestions on how the system can be partitioned. However, CA – thus the computation of the proposed indices – usually requires the choice of a value of α, which is discussed in [Papadopoulos et al., 2013].

On the basis of those remarks, a *parametric separability analysis* can be performed:

(1) perform a parametric CA, and express the time scales associated with each dynamic variable as a function of $\alpha \in (0,1)$;
(2) for each value of $\alpha \in (0,1)$, order the time scales obtaining a set of values of the integration steps $\mathcal{H} = \{h_1 \leq h_2 \leq \ldots \leq h_N\}$;
(3) for each value of $\alpha \in (0,1)$, compute the separability terms $s_\alpha(i+1,i)$ for all $i = 1, \ldots, N-1$;
(4) plot the obtained $s_\alpha(i+1,i)$ as a function of α and $i = 1, \ldots, N-1$, possibly as a colormap.

The result of this kind of analysis is that whenever a couple of dynamic variables (identified in the set $\mathcal{H}$ with their indices i and $i+1$), the plot will highlight a peak—examples of those kind of plots are presented later on.

This kind of analysis provides information that is twofold and immediately interpretable by the modeller. First, considering only a single peak for simplicity, the variables on one side of the peak may be considered as coupled, and highly decoupled in terms of time scale from the variables on the other side of the peak. This can be exploited for designing a partition of the system, or multiple in the case of many peaks. In addition, since the variables are ordered by time scales, those with lower indices are associated with fast time scales, the others with slow ones.

4.4 Exploiting the partition

The information coming from the separability analysis can be exploited in different ways. A first possibility is to split the model into two submodels, and use suitable integration methods [Papadopoulos et al., 2013]. On the other hand, the identified time scales can be used to structure more complex (co-)simulation architectures, splitting the system into many subsystems.

In particular, if the time scales present in the considered model can be clustered into more than two sets, an iterative approach can be used so as to improve simulation efficiency, extending the mixed-mode integration method to a multi-rate mixed-mode integration. The simulation structure can be obtained as follows

(1) identify the time scales by means of the separability analysis proposed herein;

(2) according to the time scale of interest, the system can be split into two subsystems, one slow that will be simulated with an explicit method, one fast that will be integrated with implicit method(s);
(3) if the faster dynamics cannot be split into other subsystems according to the time scales, then the algorithm terminates;
(4) otherwise, the fast dynamics are split into two subsystems, one faster, and one slower, that will be integrated with a multi-rate implicit method;
(5) go to step (3).

This approach automatically builds the simulation architecture, exploiting the system structure without any added effort on the part of the modeller. The resulting faster partitions are smaller reducing the computational complexity of solving them with implicit algorithms.

5. SIMULATION EXAMPLES

5.1 Double-mass, triple spring-damper

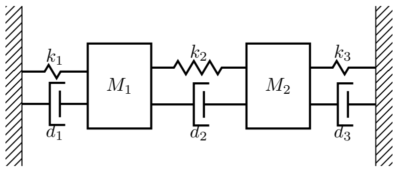

Fig. 4. Double-mass, triple spring-damper.

This example refers to a simple test problem, similar to that presented in González et al. [2011]. The considered system is composed of two masses and three parallel spring-damper elements, connected as shown in Fig. 4, and moving in a horizontal plane (i.e., gravity has no effect). Both elasticity and damping friction are assumed to be linear phenomena, so that the couplings between the dynamic variables can be easily determined by acting on the elastic constants k_i and the damping factors d_i. In particular, in the reported test, $M_1 = M_2 = 1\,\mathrm{kg}$, $k_1 = 500\,\mathrm{N\,m^{-1}}$, $d_1 = 5\,\mathrm{N\,s\,m^{-1}}$, $k_2 = 1\,\mathrm{N\,m^{-1}}$, $d_2 = 1\,\mathrm{N\,s\,m^{-1}}$, $k_3 = 5\,\mathrm{N\,m^{-1}}$, and $d_3 = 1\,\mathrm{N\,s\,m^{-1}}$.

Letting x_1 and x_2 the horizontal positions of the two masses represented in Fig. 4, the model can be written as

$$\begin{cases} M_1\ddot{x}_1 = -(d_1+d_2)\dot{x}_1 + d_2\dot{x}_2 - (k_1+k_2)x_1 + k_2x_2 \\ M_2\ddot{x}_2 = d_2\dot{x}_1 - (d_2+d_3)\dot{x}_2 + k_2x_1 - (k_2+k_3)x_2 \end{cases} \tag{8}$$

The separability analysis can be perform to understand if the model is suited to be partitioned. The result of the parametric analysis is shown in Fig. 5, where the numbers on the vertical axis index the variables ordered by increasing time scale.

The highest separability term is obtained between the 4-*th* and the 5-*th* variable, suggesting where to partition the model for the decoupled integration, however, another partition could be performed between the 6-*th* and the 7-*th* variable.

According to CA there are 17 cycles in the model digraph, and choosing $\alpha = 0.5$, the following constraints on the integration step are obtained.

$$\begin{array}{llll} \dot{x}_1: & h \le 0.032 & \dot{x}_2: & h \le 0.289 \\ x_1: & h \le 0.032 & x_2: & h \le 0.289 \\ \dot{y}_1: & h \le 0.045 & \dot{y}_2: & h \le 0.408 \\ y_1: & h \le 0.045 & y_2: & h \le 0.408 \end{array} \tag{9}$$

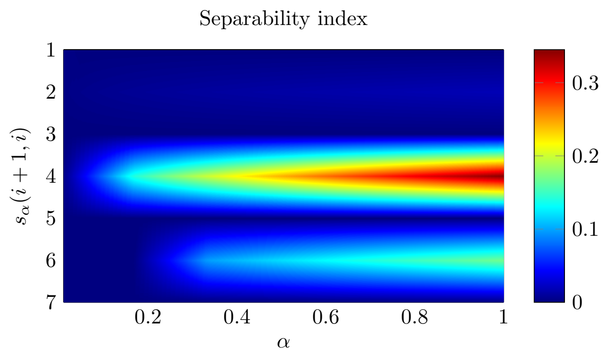

Fig. 5. Separability parametric analysis of the double-mass, triple spring-damper system.

Based on the aforementioned analysis, we can partition the model into two submodels by separating the first 4 variables – the set of the ones on the left, that are considered fast – from the other 4 — the set of the ones on the right that are considered slow. Notice that incidentally the analysis brought to an intuitive result, i.e., to separate the two set of equations associated to the two masses, without any *a priori* suggestion to the method of the physical structure of the system.

To complete the example, the proposed indices are here computed. In particular, model (8), and CA result (9) yield

$$\sigma(0.5) = 12.923, \quad s(0.5) = 0.779.$$

The stiffness index $\sigma(\alpha)$ indicates that the system is highly stiff. On the other hand, the separability index $s(\alpha)$ shows that the considered example has dynamics evolving with quite different time scales, thus making it effective to partition the model into subsystems. Notice that σ_R cannot be computed since there are purely imaginary eigenvalues in the system.

5.2 Counterflow heat exchanger

This example refers to a counterflow heat exchanger with two incompressible streams reported in Fig. 6. Both

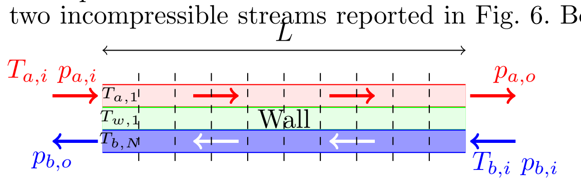

Fig. 6. Counterflow heat exchanger scheme.

streams and the interposed wall are spatially discretised with the finite volume approach, neglecting axial diffusion in the wall and also in the streams, as zero-flow operation is not considered for simplicity. Taking ten volumes for both streams and the wall, with the same spatial division (again, for simplicity) leads to a nonlinear dynamic system of order 30, having as boundary conditions the four pressures at the stream inlets and outlets, and the two temperatures at the inlets. More precisely, the system is given by

$$\begin{cases} c_a \dfrac{M_a}{N}\dot{T}_{a,i} = w_a c_a (T_{a,i-1} - T_{a,i}) + \dfrac{G_a}{N}(T_{w,i} - T_{a,i}) \\ c_w \dfrac{M_w}{N}\dot{T}_{w,i} = -\dfrac{G_a}{N}(T_{w,i} - T_{a,i}) - \dfrac{G_b}{N}(T_{w,i} - T_{b,N-i+1}) \\ c_b \dfrac{M_b}{N}\dot{T}_{b,i} = w_b c_b (T_{b,i-1} - T_{b,i}) + \dfrac{G_b}{N}(T_{w,N-i+1} - T_{b,i}) \end{cases} \tag{10}$$

where T stands for temperature, w for mass flowrate, c for (constant) specific heat, M for mass, and G for thermal conductance; the a, b and w subscripts denote respectively the two streams and the wall, while $i \in [1, N]$ ($i = 0$ for boundary conditions) is the volume index, counted for both streams from inlet to outlet, the wall being enumerated like stream a.

The parameter values used in the example are reported in Table 1.

Table 1. Parameter values of Model (10).

Parameters					
N	30	M_b	1 kg	c_w	$3500\,\mathrm{J\,kg^{-1}\,K^{-1}}$
$T_{a,\mathrm{in}}$	323.15 K	M_w	10 kg	G_a	$8000\,\mathrm{W\,K^{-1}}$
$T_{b,\mathrm{in}}$	288.15 K	c_a	$4200\,\mathrm{J\,kg^{-1}\,K^{-1}}$	G_b	$8000\,\mathrm{W\,K^{-1}}$
M_a	0.1 kg	c_b	$3500\,\mathrm{J\,kg^{-1}\,K^{-1}}$		

A parametric CA is performed so as to analyse the structure of the system, and Fig. 7 shows its result. In particular, 585 cycles are present in the system.

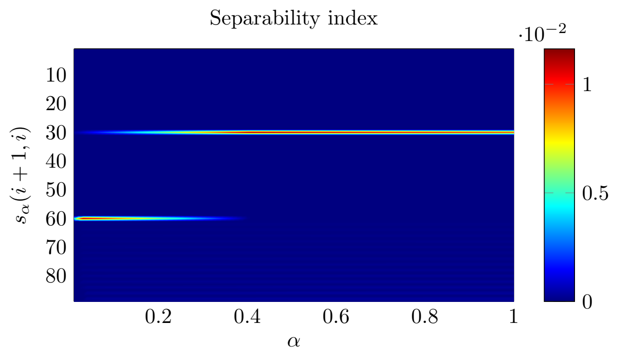

Fig. 7. Separability analysis of the heat exchanger (10).

The separability analysis highlights that there are at least a couple of points where the system can be separated. However, for $\alpha = 1.0$ there is only one point in which the system can be split, and it is between the 30-*th* and the 31-*st* variable, separating the faster state variables $T_{a,i}$, from the other ones. It is worth noticing that in this example, there is no neat physical separation between the dynamics, since they all belong to the same physical domain, and also to the same physical object. Separability analysis, however, can detect those structural properties of the system independently of its nature.

The synthetic indices are

$$\sigma(0.5) = 9.771, \quad s(0.5) = 0.986.$$

The stiffness $\sigma(\alpha)$ index shows that the considered system is sufficiently stiff, but the more interesting aspect is that the separability one shows that it is very suited for the partition, since its time scales are very well separated.

6. CONCLUSIONS

In this work we proposed a method for analysing structural properties of nonlinear dynamical systems, extending the results presented in Papadopoulos et al. [2013]. Synthetic indices quantifying the stiffness and the separability of the considered system are here proposed, that in conjunction with the parametric analysis can provide immediate information to the modeller for the construction of the simulation architecture. In addition, different ways to exploit the information coming from the parametric cycle analysis, both from the architectural viewpoint and from the numerical method one.

In the near future, further investigations are needed for extending and formally address the properties of the numerical simulation architectures that can be obtained by means of the proposed exploitations. Therefore, the integration of the proposed methods to off-the-shelf simulation tools is the main goal of the overall research, aimed at ease the modeller tasks in a way as transparent as possible. Finally, the exploitation of PCS for enhancing simulation performance of the fast parts is to be applied to some significant cases.

REFERENCES

C. Andersson, J. Åkesson, C. Führer, and M. Gäfvert. Import and export of Functional Mock-up Units in JModelica.org. In *Proc. 8th Int. Modelica Conf.*, pages 329–338, 2011.

A.C. Antoulas. *Approximation of large-scale dynamical systems*, volume 6. Society for Industrial Mathematics, 2005.

U.M. Ascher, S.J. Ruuth, and R.J. Spiteri. Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations. *Applied Numerical Mathematics*, 25(2–3):151–167, 1997.

T. Blochwitz, M. Otter, J. Åkesson, M. Arnold, C. Clauß, H. Elmqvist, M. Friedrich, A. Junghanns, J. Mauss, D. Neumerkel, H. Olsson, and A. Viel. Functional Mockup Interface 2.0: The standard for tool independent exchange of simulation models. In *Proc. 9th Int. Modelica Conf.*, pages 173–184, 2012.

F. Casella. A strategy for parallel simulation of declarative object-oriented models of generalized physical networks. In *Proc. 5th Int. Workshop on Equation-Based Object-Oriented Modeling Languages and Tools*, pages 45–51, 2013.

F. Cellier and E. Kofman. *Continuous system simulation.* Springer, London, UK, 2006.

F. Donida, F. Casella, and G. Ferretti. Model order reduction for object-oriented models: a control systems perspective. *Math. and Comp. Modelling of Dynamical Systems*, 16(3):269–284, 2010.

S. Fortunato. Community detection in graphs. *Physics Reports*, 486 (3):75–174, 2010.

F. González, M.Á. Naya, A. Luaces, and M. González. On the effect of multirate co-simulation techniques in the efficiency and accuracy of multibody system dynamics. *Multibody System Dynamics*, 25(4):461–483, 2011.

B. Hendrickson and K. Devine. Dynamic load balancing in computational mechanics. *Computer Methods in Applied Mechanics and Engineering*, 184(2–4):485–500, 2000.

M. Innocent, P. Wambacq, S. Donnay, H.A.C. Tilmans, W. Sansen, and H. De Man. An analytic volterra-series-based model for a mems variable capacitor. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Trans. on*, 22(2):124–131, 2003.

B.W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *Bell system technical j.*, 29:291–307, 1970.

A.V. Papadopoulos and A. Leva. Automating dynamic decoupling in object-oriented modelling and simulation tools. In *5th Int. Workshop on Equation-Based Object-Oriented Modeling Languages and Tools*, pages 37–44, 2013.

A.V. Papadopoulos and A. Leva. A model partitioning method based on dynamic decoupling for the efficient simulation of multibody systems. *Multibody System Dynamics*, 2014. (in press).

A.V. Papadopoulos, J. Åkesson, F. Casella, and A. Leva. Automatic partitioning and simulation of weakly coupled systems. In *Proc. 52nd IEEE Conference on Decision and Control*, pages 3172–3177, 2013.

A. Schiela and H. Olsson. Mixed-mode integration for real-time simulation. In *Modelica Workshop 2000 Proc.*, pages 69–75, 2000.

M. Sjölund, R. Braun, P. Fritzson, and Petter Krus. Towards efficient distributed simulation in modelica using transmission line modeling. In *3rd Int. workshop on Equation-Based Object-Oriented Modeling Languages and Tools*, pages 71–80, 2010.