Variability and Statistical Dynamic Analysis Flow for Linear Interconnect Networks

António Lucas Robalo Martins

Abstract—The analysis of interconnect networks under the influence of limitations in the lithographic process is of extreme importance in the design cycle of integrated circuits, as those variations can cause undesired behaviour in the final product. Power grids are extremely large interconnects, covering the circuit area and used to provide the required bias to all other circuit elements. Variations can impact the power grid local or globally, giving rise to the need to simulate the effects of process variation. Extensive work has been done trying to solve this problem efficiently, whose main difficulty lies on the size of the power grids, making the problem computationally intensive. To analyze the effects of process variation, statistical data of significance is obtained from analysis of the circuit behaviour, which unfortunately implies solving the circuit under different parameter settings. Several approaches to the variational analysis problem have been suggested, in particular a class of methods for static variational analysis that separates the problem in two stages: first, a parametrized model for the network is created only once, and then the model is solved for any number of parameter settings in a highly efficient fashion. In this work we propose a novel scheme for the dynamic analysis of power grid behaviour under the influence of variations, where a compressed parametrized model is used to represent the network. The proposed scheme is highly parallelizable, and achieves good accuracy, a reasonable speed-up versus common techniques and has low memory requirements due to high model compressibility.

Index Terms—Dynamic Analysis, Variational Analysis, Model Order Reduction, Parametrization.

I. INTRODUCTION

An important step in integrated circuit (IC) design is the design of the interconnect power grid (PG) networks, as they span across the entire chip area, providing circuit bias and carrying large currents. As the size of each component in an IC decreases, the interconnect lines that bring the circuit together also get smaller. Nowadays, the width of the interconnect lines is on the order of the tens of nanometers, with 14nm technologies being presented [1].

Technological evolution has led to the design of IC’s with smaller and smaller components, in an effort to cram more sophisticated designs in a single chip. As the size of each component in an IC decreases, it is natural that the power grid wires that bring the circuit together also get smaller. Nowadays, the width of the interconnect lines is on the order of the tens of nanometers, with 14nm technologies being presented [1].

The nanometer regime introduces new challenges, as the lithographic process described above failed to keep pace with Moore’s law and the printed features are now under the wavelength of the light used on the process. That creates an environment where printed features become highly susceptible to variations on the lithographic process, leading to increased variability on the final designs, affecting their behaviour and performance. This variations affect the power grids, e.g. variations on the power grid wiring width that may cause non-desired voltage and current fluctuations on the chip, that can lead to several problems. Therefore the need to analyze and simulate the network under the effect of those variations arises.

Nowadays, the analysis of design variability is very challenging [2], as it is a time-consuming task due to the large number of network variations that must be studied, leading to repetitive simulation of the network taking increased computational effort. Several cases must be studied: DC perturbations, where the circuit voltage bias is affected, and AC perturbations, changes in the transient response when the circuit is turned on. An example of a relevant analysis that is affected by the process variation is the electromigration analysis.

Electromigration (EM) refers to the unwanted movement of materials in a semiconductor, and this effect becomes especially relevant when working with high current densities, as it ends up happening in an IC, where currents circulate in very thin power grid wires. This phenomena can cause failures in the integrated circuit, making it crucial to simulate and analyze new devices before their manufacture so that EM problems can be found in the early stages of development, as the cost of prototyping and manufacture is elevated. This analysis can be done by determining the voltage of the power grid nodes (from which the branch currents can be obtained and the EM analysis done).

The importance of analyzing the power grid is real. This task, however, is not computationally simple. PG descriptions are very complete and the number of voltage nodes is extremely high, in the order of hundred million nodes. Solving the power grid becomes a resource-hungry task.

If the AC problem is considered, the power grid must be solved for several time steps, and when process variation is introduced all the analysis effort must be repeated for a large number of parameter settings that describe the effect of the variations, so that statistical information about how the power grid behaviour changes with process variation can be obtained. In the end, a time-variant variational analysis seems to require a considerable amount of solves of the original analysis problem.

However, one can exploit both the structure of the power grid representation and the physical proprieties of the grid, as the spatial distribution of voltages, to find a better solution for the PG analysis problem.
The present dissertation will focus on the analysis of power grids under the effect of perturbations caused by process variation, discussing state of the art methodologies and presenting a novel scheme to the efficient and memory-aware analysis of the dynamic behaviour of interconnect networks. The performance of the proposed scheme will be evaluated using a set of benchmark power grids, usually used by the community.

The end result of the present work is a novel scheme that efficiently estimates the effects of variability on dynamic interconnects. We start by defining a set of possible process variations and present their impact on the power grid behaviour. Then, we present the analysis scheme, accomplished in a two-stage approach: a set up stage where a parametrized model for the network is generated only once, and compressed on-the-fly; a highly efficient evaluation stage that can be repeated for any number of parameter settings.

The compression will allow the analysis of large networks that would otherwise be impossible to evaluate.

This scheme can be embedded in enhanced design cycles and EDA software, allowing the study of the impact on circuit performance of the variations, for example to estimate the distribution of the maximum resistor current (relevant for electromigration analysis), the maximum voltage drop (relevant for grid integrity analysis), or to obtain statistical information about voltage variations.

A paper describing this work entitled “Variability and Statistical Analysis Flow for Dynamic Linear Systems with Large Number of Inputs” was submitted to the DATE 2016 Conference [3] and is under review at this time.

The manuscript is structured as follows. Section II briefly reviews the PG problem and the basics of the variability modeling and analysis. Section III introduces the proposed methodology, and discusses computational implementations and practical considerations. Finally results for a set of industrial PG benchmarks are presented in Section IV, and conclusions drawn in Section V.

II. BACKGROUND AND PROBLEM DEFINITION

A. Variational Power Grid Model

A power grid model can be obtained through extraction and assumes that VDD and GND strips, as well as vias, are modeled resistively. The coupling resulting from the overlapping between metal strips in different levels is modeled through a coupling capacitance or a set of capacitances to ground. While simplified, this type of model is representative of what is used in commercial tools [4] and is sufficient for most analysis, including voltage and IR drop computation or electromigration. For analysis of the dynamic behavior of the grid, a set of equations can be obtained:

\[
G \vdots v(t) + C \frac{d}{dt} v(t) = i(t) \quad (1)
\]

where \( G, C \in \mathbb{R}^{n \times n} \) are respectively the conductance and capacitance matrices of the system, \( v(t) \in \mathbb{R}^n \) is the vector of grid node voltages through time, and \( i(t) \in \mathbb{R}^n \) the vector of bias currents at some of the grid nodes (to simplify, grid bias is assumed to have been converted to current sources through Norton equivalents). Although an RLC equivalent model accounting for inductive packaging connections can also be extracted and analyzed, we will assume a simplified RC model. Discretizing (1), for instance using the backward-Euler method with a fixed timestep \( h \), leads to

\[
Y \vdots v(t) = \left( G + \frac{1}{h} C \right) v(t) = i(t) + \frac{1}{h} C \vdots v(t-h) \quad (2)
\]

which can be solved in sequence for \( t = 0, \cdots, Mh = T \), \([0, T]\) the interval of interest. Solution of (2) is computationally expensive since the system matrix, \( Y \), is very large. However, it is also very sparse, and usually has a regular structure, which allows efficient direct and iterative methods to be applied for solving (2), albeit at some memory and computational cost, given the large matrix sizes [5]–[8]. Furthermore, for fixed-time-step the system matrix, \( Y \) in (2), is the same at all time points and thus can be factored once and the factors reused at every time point. Alternative approaches to solving the system in (2) exploit the connection with Random Walks [9], whereas others try to apply model order reduction [10]–[12], to compress the system matrix and speed up the solve.

For variability analysis, we consider possible variations on the network elements, resistors and capacitors. The value of the resistance depends on the resistivity \( \rho \), length \( L \) and section \( S \) of the wires, which for most planar structures is represented as the product of the width \( W \) and thickness \( T \), whereas the capacitance depends on the medium permittivity, \( \epsilon \), the area of the plates, \( A \), and its distance, \( d \):

\[
R = \rho L \frac{S}{W} \frac{1}{\epsilon} \quad C = \epsilon \frac{A}{d} \quad (3)
\]

The values of these electrical and geometrical parameters are subject to the effect of fluctuations, random and systematic deviations inherent to the fabrication process. We can represent \( R \) and \( C \) as functions of the parameters in the set \( p = [\Delta \rho, \Delta L, \Delta W, \Delta T, \Delta \epsilon, \Delta A, \Delta d] \),

\[
R(p) = \frac{(\rho+\Delta \rho)(L+\Delta L)}{(W+\Delta W)(T+\Delta T)} \quad C(p) = (\epsilon + \Delta \epsilon) \frac{A+\Delta A}{d+\Delta d} \quad (4)
\]

Different variability models can be considered here leading to different approximations. For instance if all variations can be considered independent and a low order approximation is deemed sufficient (which is reasonable for small parameter variations around the nominal value),

\[
R(p) \approx R_0 (1 + \Delta \rho + \Delta L + \sum_{k=1}^{O} (-\Delta W)^k + \sum_{k=1}^{O} (-\Delta T)^k) \quad C(p) \approx C_0 (1 + \Delta \epsilon + \Delta A + \sum_{k=1}^{O} (-\Delta d)^k)
\]

(5)

where \( R_0 \) and \( C_0 \) are the nominal values (no variation in the parameter set) and \( O \) is the truncated order of the approximation (usually a small order will suffice for near perfect approximation). Alternatively for inter-chip variability analysis, these variations can be assumed as localized in certain chip areas or regions and thus a different set of parameters can be considered for different areas within the chip. In any
The evaluation of $G$ where the dependent admittance (capacitance) matrix $G$ can be efficiently formed using the incidence of each element in the circuit. Alternatively, a representation such as (5) might be available for the admittance directly, from which $G$ can be computed through stamping (similarly for $C$).

A common representation is to generate a set of sensitivities of the admittance and capacitance matrices, $G(p)$ and $C(p)$, to an arbitrary order. For instance, the first order Taylor series can be written as a function of the nominal, $G_0$ and $C_0$, and the sensitivities, $G_k$ and $C_k$, to a set of $P$ parameters, $p_k$, as

$$G(p) = G_0 + \sum_{k=1}^{P} G_k p_k$$

$$C(p) = C_0 + \sum_{k=1}^{P} C_k p_k$$

This representation can be trivially extended to higher order.

### B. Estimating the Effect of Variations

Under a certain parameter setting, $p$, (2) can be written as

$$Y(p)v(t, p) = i(t) + \frac{1}{h} C(p)v(t-h, p)$$

where $Y(p) = G(p) + C(p)/h$, and we assumed that the parameter set values are time independent and the external stimuli does not depend upon the parameters. The generation of the time-evolution of the grid voltages due to a specific set of stimuli requires two steps: a) evaluate the system matrices, i.e. obtain $G(p)$ and $C(p)$ for the given parameter setting $p$, and b) solve the system (2) for $t = 0, \ldots, Mh$ with the evaluated matrix and obtain the time-evolution of the grid node voltages. The evaluation of $G(p)$ and $C(p)$ can be easily done, either by stamping the values of (5) for the given parameter, or by evaluating the low order approximation in (6).

This generates a new system valid for the current parameter set which can be solved, the solution recorded and the process repeated for the next parameter setting. For an illustration of the process, see 1.

Herein lies one of the basic difficulties with estimating the impact of parameter variability on grid behavior: for each parameter setting $p$, the matrices in (7) change; to generate the time-evolution of the voltage solution $v(t, p)$, implies a re-factorization of the matrix $Y(p)$ or a new solution process if an iterative method is used. This is very costly and in fact to compute the impact of multiple parameter settings is quickly overwhelming.

Some work has been devoted to the analysis of such networks under parameter variations [13], including combination with MOR methods [14], the extension of Random Walks for variational analysis [15], [16], incremental sparse methods [17], bounded effects and estimates for voltage drop variations for statistical methods [18], and the use of Hermite Polynomials for the generation of a variational model that allows a stochastic analysis [19]. Most of these existing methodologies for the variability analysis of PGs focus on the idea of accelerating the solution of this system for different parameter settings, either by incremental analysis and approximations, system parametrization and model reduction, or localized updates of the solution. For large parameter settings, for instance, to estimate the distribution of peak resistor current (relevant for electromigration analysis), peak voltage drop (relevant for grid integrity analysis) or any other required metric, more efficient methods are required.

### III. PROPOSED VARIABILITY ANALYSIS

Parameter variations are responsible for the time evolution of the node voltages drifting from the nominal. The estimation of these voltage perturbations is critical for safe and reliable PG design. Our analysis is based on two basic assumptions. We conjecture that the node voltages are smoothly time-varying and can be accurately recovered with low order approximations. Furthermore, the voltages for different time steps and parameter settings are correlated, both from a spatial as well as a time viewpoint. This correlation translates into the numerical concept of rank deficiency, which means that we can represent the voltages in the nodes for different time points and parameter settings as a linear combination of a small basis vectors:

$$v(mh, p) \approx Q \alpha(mh, p)$$

where $m$ corresponds to a discretized time step, $t = mh$, $p$ is the parameter vector, and $Q \in \mathbb{R}^{n \times q}$ is the compressed basis, with $q << n$ a relatively small number and $\alpha(m, p)$ a set of coefficients (an appropriate tolerance, $\mu$ is used at each timepoint to determine whether a new vector should be added to the basis). If no compression were used, the above expression would still be valid but now $Q \in \mathbb{R}^{n \times (P+1)}$ which is huge, rendering the method unusable.

Our approach involves two steps. The first step is the proposed model generation, where the terms for the representation in (8) are generated. The second is the analysis, in which the representation in (8) is used to evaluate the node voltages at different time points for as many different parameter settings as desired.

#### A. Model Generation – SET UP

The model generation further requires the separation of the time space and the parameter space. We will use a standard
time-domain simulation of the nominal model (zero variation), to generate the required basis. Following the approach outlined in (2), we can fix a time step $h$, and establish the backward-Euler regression formula to compute the voltage vector at the next time point,

$$Y_0v^{(m)}_0 = i^{(m)} + \frac{1}{h}C_0v^{(m-1)}_0$$  \hspace{1cm} (9)

where to simplify notation we now write $v^{(m)}_0$ to represent the nominal voltage vector at the $m$-th time point ($t = mh$) (similarly for $i^{(m)}$). In order to determine the effect of perturbations on the parameters, we handle each time point as a static model, and pursue a similar approach to the one proposed in [20] for the static case. We aim to generate a set of linear terms that approximate the effect of the parameter variations at such time step

$$v^{(m)}_p = v^{(0)}_0 + \sum_{k}^p p_k v^{(m)}_k$$  \hspace{1cm} (10)

Such an approximation is presumably sufficiently accurate if the range of variation of the parameter set $p$ is small compared to the nominal value, which is often the norm. Otherwise higher order terms can be added for the sake of accuracy, at the cost of rapidly increasing the computing requirements, especially when multiple parameters are involved. Similarly to [20], the coefficients $v^{(m)}_k$ can be efficiently computed by either exploding the voltage in terms of a low order Taylor series with respect to the parameters (an approach previously exploited in [21] for parametric Model Order Reduction), or by matching the perturbed response of the voltage vector at the maximum (or any other given) variation for each parameter. Both cases establish a recursion for each parameter that avoids parameter settings. We illustrate here how to compute these for a single parameter.

For a single parameter $p_1$, with maximum value $\hat{p}_1$, and a single term $v^{(m)}_1$, matching the response at the maximum variation leads to

$$v^{(m)}_0 + \hat{p}_1 v^{(m)}_1 = v^{(\hat{p}_1)}^{(m)}$$  \hspace{1cm} (11)

We can compute the value of $v^{(\hat{p}_1)}^{(m)}$ by applying backward-euler,

$$Y(\hat{p}_1)v^{(\hat{p}_1)}^{(m)} = i^{(m)} + \frac{1}{h}C(\hat{p}_1)v^{(\hat{p}_1)}^{(m-1)}$$  \hspace{1cm} (12)

We can approximate the matrices by their first order parametric representation $Y(\hat{p}_1) = Y_0 + \hat{p}_1 Y_1$, $C(\hat{p}_1) = C_0 + \hat{p}_1 C_1$, and do the same for the previous time point voltage, $v^{(\hat{p}_1)}^{(m-1)} = v^{(0)}_0 + \hat{p}_1 v^{(m-1)}_1$. After some linear algebra, we get

$$v^{(\hat{p}_1)}^{(m)} = (Y_0 + \hat{p}_1 Y_1)^{-1} \left( i^{(m)} + \frac{1}{h}C_0v^{(m-1)}_0 + \hat{p}_1 b \right)$$  \hspace{1cm} (13)

where $b = 1/h \left( C_1v^{(m-1)}_0 + C_0v^{(m-1)}_1 + \hat{p}_1 C_1 v^{(m-1)}_1 \right)$.

We can approximate the inverse term in (13) by a truncated series

$$(Y_0 + \hat{p}_1 Y_1)^{-1} = \left( I - \Delta^1 + \Delta^2 - \ldots \right) Y_0^{-1}$$  \hspace{1cm} (14)

where $\Delta = \hat{p}_1 Y_0^{-1} Y_1$. By substituting the previous results in (11), reordering the terms, and using the backward-euler equivalence in (9), we finally arrive at the following expression

$$v^{(m)}_1 = (I - \Delta + \Delta^2 - \ldots) Y_0^{-1} \left( b - Y_1 v^{(m)}_0 \right)$$  \hspace{1cm} (15)

A simplified approach is to expand the voltage $v^{(m)}$ and matrices $G(p)$ and $C(p)$ in (7) by their corresponding first order expansion (10) and (6), and to match the coefficients of the same order. This is equivalent to obtain the derivative of (7) with respect to the given parameter $p_1$ around its nominal value, which, neglecting high order terms, eventually give us the following approximation

$$v^{(m)}_1 = Y_0^{-1} \left( \frac{1}{h} (C_1 v^{(m-1)}_0 + C_0 v^{(m-1)}_1) - Y_1 v^{(m)}_0 \right)$$  \hspace{1cm} (16)

Note that the computation of the terms in the series in (15) and (16) can be done in a recursive fashion, where the solves only involve $Y_0$, and a series of matrix vector products that only depend on previously available data, namely the nominal voltage at the previous and current time points ($v^{(m)}_0$ and $v^{(m-1)}_0$), and the perturbation of the voltage at the previous time point ($v^{(m-1)}_1$). The same procedure can be followed for all the other parameters.

These computations are independent for each parameter, and thus embarrassingly parallelizable. Furthermore, if $Y_0$ has been factorized (and stored), we only need to re-use the factors to solve different right hand sides to generate all the $v^{(m)}_k$ terms.

Once the vectors for the current time point ($m$) have been generated, we can use these vectors to update the basis $Q$ with the dominant vectors. In order to do so, we need to orthogonalize the new vectors with respect to the vectors in the basis. After that, we can apply a rank-revealing QR (rrqr) factorization and truncation to keep the dominant vectors in the basis, and use them to update the $Q$ matrix, and the $R$ matrix, that we will need to keep track of the relation of the vectors to the time points and parameters. In Algorithm 1 the Set Up is summarized.

**Algorithm 1: Model Set Up**

<table>
<thead>
<tr>
<th>Data: $h, v^{(0)}_p, \tilde{v}_p \forall p \in {1, \ldots, P}$, $\tilde{v}^{(m)}_m \forall m \in {1, \ldots, M}$, $G_0, C_0, G_p, C_p \forall p \in {1, \ldots, P}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Result: $Q, R$</td>
</tr>
</tbody>
</table>

Factorize $Y_0 = (G_0 + C_0)(R)$

for $m = 1, \ldots, M$ do

- Generate the nominal response of the system $v^{(m)}_0$
- Apply the RRQR and update $Q, R$

for $p = 1, \ldots, P$ do

- Generate the perturbed response of the system $v^{(m)}_p$
- Apply the RRQR and update $Q, R$

end

end

---

| $\Delta = \hat{p}_1 Y_0^{-1} Y_1$, $b = 1/h \left( C_1 v^{(m-1)}_0 + C_0 v^{(m-1)}_1 + \hat{p}_1 C_1 v^{(m-1)}_1 \right)$ |

$$(Y_0 + \hat{p}_1 Y_1)^{-1} \left( i^{(m)} + \frac{1}{h}C_0v^{(m-1)}_0 + \hat{p}_1 b \right)$$

$$(Y_0 + \hat{p}_1 Y_1)^{-1} = \left( I - \Delta^1 + \Delta^2 - \ldots \right) Y_0^{-1}$$
on the $n$ circuit nodes for any of the $M$ time points and for any parameter setting. $R \in \mathbb{R}^{n \times M(P+1)}$ is the matrix that allows us to select and weight the vectors in $Q$ to actually generate the voltage vector.

The columns of $R$ can be subdivided in subsets of $P+1$ columns. Each of these subsets is related to a time point $m$, with the first column the one related to the nominal response, and each of the remaining $P$ columns related to a parameter in the parameter set $p$. If we want to evaluate the voltages for the time point $m$, for a given parameter setting $\hat{p}$, we just need to select the corresponding columns in $R$, and apply a set of matrix vector products

$$v^{(m)}(p = \hat{p}) = Q \left( R(m) \left[1 \right. \left. \hat{p} \right]^T \right)$$

(17)

where $R$ is the sub-block of $R$ that contains the columns that go from $m$ $(P+1)$ to $(m+1)$ $(P+1)$, and $\left[1 \ \hat{p} \right]$ is a row vector of $P+1$ with value 1 for the nominal position, and the value of the parameters for the given setting in the remaining positions.

An interesting advantage of this approach is that time-dependent parameter variations (e.g. to determine the effect of temperature variations whose value is known to vary with time) can be easily accommodated since the computation in the evaluate stage is done one timepoint at a time using the values of the parameters at each timepoint.

C. Computational Issues

Figure 2 presents a depiction of the proposed flow, in which the SET UP (dotted red box) and EVALUATION (dashed blue box) stages are clearly separated. The SET UP stage, akin to model generation, is done only once, and computes the terms $v_k (k = 1, \cdots, P)$ required for generating the approximation. The selection of the value $\hat{p}_k$ will determine the region of best accuracy. Zero value will give a series approximation with better accuracy at small perturbations. Fixing $\hat{p}_k$ at a given variation will give better accuracy around nominal and that given variation, but its accuracy may degrade for other values if the voltage response is not close to linear. We assume that the sensitivities of the conductance and capacitance matrices, $G_k$ and $C_k$, are given, or in cases where they are not available, we have access to an extractor and can generate them through a simple method such as differentiation, i.e. $G_k = (G(\hat{p}_k) - G(\hat{p}_0))/(\hat{p}_k - \hat{p}_0)$.

The complete flow requires factorizing the nominal pencil term, $Y_0 = G_0 + C_0/h$ (e.g. using Cholesky factorization) only once, and storing the factors. These factors are used to solve (i.e. back-solve using the factors) for the nominal voltage $v_0$, and then for the sensitivities of the linear approximation $v_k$ at each time step. For a linear approximation the total number of (back-)solves per time point is directly proportional to the total number of parameters, $P$. The generation of the different linear terms $v_k$ for each time step in the SET UP phase is perfectly independent once the factorization of the nominal matrix is done. Multiple cores or machines can therefore be used in any type of architecture with speedup limited only by the number of parameters of the problem. The factorizations can be done either as a block for each time step, or as rank-one update for each parameter. In any case they can be efficiently applied by using implicit Householder-based rank-revealing QR methods [22]. A relative threshold can be used to drop the columns of $Q$ and rows of $R$ associated with rank deficient vectors, and thus reduce memory requirements for the storage. Also, the analysis can be focused on a limited number ($r < n$) of relevant nodes (e.g. critical nodes or outputs) which could help to minimize memory requirements.

The EVALUATION stage, akin to model evaluation, computes the results of the variational analysis. Since the “coefficients” $v_k$ of the linear approximation have been pre-computed, the actual variational analysis is very fast, as it only requires a set of dense matrix vector products for each parameter setting. The values of each parameter setting are stored in a column vector, and left-multiplied by the columns of matrix $R$ corresponding to the desired time point, to generate the coefficient vector $\alpha(m,p)$ in (8) for the given parameter setting. This vector is used as a weighting for the basis $Q$ to reconstruct the voltage in the nodes. Efficient level 2 BLAS routines can be applied in this stage. Furthermore, different parameter settings can be divided between an arbitrary number of machines and computed concurrently both in shared and/or distributed memory architectures. The computation is fully/embarrassingly parallel. Additionally, the evaluation of the dense matrix vector products can be accelerated by using GPUs, since the basis remains constant for any time point and parameter settings: multiple parameter settings can be loaded to the GPU and computed in a vectorial fashion.

Interestingly, the proposed approach can benefit from complementary acceleration techniques to solve the time domain analysis of the nominal system with constant matrix $Y_0$, including $H$-matrix representations and fast solvers, Model Order Reduction (MOR), or multi-grid and iterative methods [5], [6], [8] to speed up the SET UP phase.

IV. EXPERIMENTS AND RESULTS

In this section we present results from application of the proposed approach to a set of realistic designs taken from
the IBM Power Grid Benchmarks for Transient Analysis [4]. Table I shows the characteristics of the examples used where n stands for the effective number of nodes (and therefore of capacitors) and r for the number of resistors, after processing the data and removing shorts. We have further divided the PG into R local regions, using topological and net information. This division tries to emulate locality of the perturbations. For each region we used 6 independent parameters from the set of parameters (resistivity, length, width, thickness of resistor wires and permittivity, plate width and plate distance of capacitors). P stands for the total number of independent parameters (6 times the number of regions). The parameters affect the resistors in the corresponding region, with a 3σ effect of ±10% for the resistivity, ±1% for the length, ±30% for the thickness and ±30% for the width. Capacitor values in each region are affected with a 3σ effect of ±10% for the permittivity, ±30% for each dimension in plate area and ±10% for plate separation). We use the original perturbed system, i.e. the stamp of the perturbed resistors and capacitors under the effect of some parameters, using a 3rd order approximation in (10), as the golden solution against which the accuracy of the methods is presented. We compare standard Taylor series approximation (12), with the proposed approach, using the same sensitivities as in (12), obtained by direct differentiation at the maximum perturbation for each parameter.

All our examples were run in a MATLAB environment, which means times are merely indicative, since it is a non-compiled language. Nonetheless, since all experiments were conducted in the same environment, the relative comparisons are a relatively fair indication. To quantify the results, we show the maximum absolute error (Eabs), the average absolute error (Aabs), and the average relative error (Arel), obtained from the simulation of a large number of parameter settings and considering all timepoints in the simulation:

\[
E_{\text{abs}} = \max |v^m(\bar{p}) - v^m_r|, \forall m = 1, \cdots, M \\
A_{\text{abs}} = \text{mean} \left( |v^m(\bar{p}) - v^m_r| \right), \forall m = 1, \cdots, M \\
A_{\text{rel}} = \text{mean} \left( |v^m(\bar{p}) - v^m_r| / |v^m_r| \right), \forall m = 1, \cdots, M
\]

(18)

Table II shows, for example ibmpg1t, the effect of decreasing the tolerance \( \mu \) (see the discussion around (8)) on our ability to accurately recover the time-domain representation of the network (errors shown are the maximum seen for a set of 1000 random parameter settings). The data shows that the representation is in fact quite compressible. For values of \( \mu \) of 10^{-2} and comparing our approach with a method based on an uncompressed basis (see the discussion around Eqn. (8)) we achieve savings of over 98% in memory while incurring on a very reasonable absolute and relative error. This proves our assumption that the time-domain waveforms are indeed highly correlated and the data is compressible along the time dimension. This can be further studied by changing the interval of the time simulation, as seen in Table III. Memory savings increase with the increase in simulation time.

Next, in Table IV, we show, for examples ibmpg1t and ibmpg2t, the effect of an increasing number of parameters in our ability to compress the representation. The data shows that for increasing number of parameters (P), more basis vectors (q) are required, but the rate of increasing is slower than the increase in the number of parameters, again showing considerable compressibility and the potential for considerable memory savings which will later translate into computational savings in the EVALUATION phase. This sub-linear increase in memory usage is highlighted in Fig. 3, where a logarithmic function is adjusted to the data.

Furthermore, for large number of parameters there are appreciable memory savings in compressing the representation. This again is in line with our assumptions of considerable correlation in the parameter space.

A third dimension where there is also potential for large compression is the space dimension. To achieve such compres-
The proposed approach requires more time in the set up stage, but the extra effort is rewarded during the evaluation stage, where even for a relatively small number of parameter settings (1000 in this case), we already have a large speed up: 9.4×, 1.7×, and 2.5× for ibmpg1t, ibmpg2t and ibmpg3t respectively. The speed up per iteration (without taking into account the Set Up time, and thus an indicator of the actual speed up for very large parameter settings) can be larger than 20 fold for the larger grid. This speed up is expected to increase for larger examples, since the cost of the solve required by TS is \(O(n^3)\), with \(\beta > 1\), whereas the cost of evaluating the proposed model, which can be done using highly efficient BLAS routines, is \(O(nq)\), with \(q\) the size of the basis.

In terms of accuracy, both approaches exhibit acceptable levels for the problem at hand, with the worst case error for all parameter settings and timepoints (\(E_{abs}\)) similar in all cases. In all cases the accuracy is quite reasonable, considering the simplicity of the approximation used, with very small relative errors shown in all cases.

The proposed flow can be efficiently used to extract and monitor statistical information of the performance of the system while subject to parameter variations. By performing a large number of parameter evaluations, the average and standard deviation values of a time domain simulation for multiple nodes can be analyzed. For example, Figure 5 shows the nominal (no variation) time domain evolution of the voltage of a node in ibmpg2t, along with the average and the 3σ limits of the information extracted from 1000 MC settings. It is clear that the impact of the variation, even on a relatively simple case, can be quite relevant. This is a valuable information for designers, that can be exploited in the design cycle to minimize failures due to process and parameter variations.

The proposed algorithm efficiency can be dramatically increased by the use of more efficient algorithms for matrix manipulations [23]. The use of the BLAS package [24], namely the level 2 BLAS routines that deal with matrix-vector operations, will speed up the solve stage of the algorithm.
There are several directions one can follow to enhance the model generation stage. Three possibilities would be: generating a smaller model, therefore reducing the evaluation time; generating a model in a faster fashion, while maintaining the approximation error low; generating a more complex model, increasing specification.

All seem as a probable goal for future work that is developed around the presented analysis flow. As new specifications for the model, adding packaging models leading to RLC analysis or introducing nonlinear devices seem interesting options. With that assumption in mind, we suggest a new approach to the generation of the parametrized model, based on support vector machines.

Support vector machines (SVM) are machine learning models used to analyze data and recognize patterns, and have seen application in numerous areas for the most different applications, i.e. object detection, cancer diagnosis and signal processing [26]. A SVM-based approach could potentially provide enhancements in the generation of a compact representation of the perturbed network, and also function as a model order reduction tool. This approach would try to reduce the size of the parametrized model that is generated by the Set Up. The main concern is to guarantee that the overhead caused by the Set Up is effectively being discounted in the Evaluation stage, and that the final balance of the computational cost is positive.

REFERENCES


The BLAS routines are a considered a standard for matrix operations e.g. dot products, matrix-vector multiplications, and matrix-matrix products [25]. BLAS implementations are highly optimized for speed, and often take into account the specificity of the hardware they run on. Those implementations are usually written in FORTRAN (as the original specification), in C, or directly in machine code, making them extremely efficient.

An experiment was prepared using example ibmpg2t, intending to compare the execution time of the SPARE approach while using the standard MATLAB matrix-vector operation algorithm and the BLAS routine, for the Evaluation stage. This, however, remains as a work in progress at this date, but a high speedup factor is expected.

V. CONCLUSIONS

We presented an efficient methodology to analyze the effect of parameter variations on the dynamic behavior of power grids and large RC interconnect networks. The approach relies on two main assumptions: the compressibility of the node responses and the smoothness of the variational dependence. Considerable compressibility has been shown in both the time domain and the parameter space with space compression also trivially within reach. For moderate parameter variations often a linear or low order approximation is sufficiently accurate, but the approach is quite general and can be computed to any order. Model generation is very efficient, as it only requires one matrix factorization of the nominal system matrix which is then reused to compute the different terms of the approximation. Once this model is generated, it allows a very fast evaluation of the effect of different parameter settings, requiring only dense matrix vector products at each timepoint for each parameter setting.

1) Future Work: There is still work to do in the development of efficient analysis methods for large power grids under the effect of variations, as this problem continues to be a great challenge due to its size. At the moment, we suggest a continuation of the present work by using two-stage schemes, as evaluating a parametrized model as suggested is a task that albeit being computationally heavy can be done very efficiently. This leaves the generation of the parametrized model as a working topic.

Fig. 5. TD evolution of the nominal voltage of one node in ibmpg2t, along with the average and 3σ limits obtained by the variability analysis.


