EFFICIENT INTEGRATION OF TRANSIENT CONSTRAINTS IN THE DESIGN OF AIRCRAFT DYNAMIC SYSTEMS

Léon L. Phan*, Dimitri N. Mavris*, Jean-Jacques Charrier, Elena García***

***Aerospace Systems Design Laboratory, Georgia Institute of Technology**
****Hispano-Suiza, Safran Group**

Keywords: *transient constraints, neural networks, visualization*

Abstract

This paper describes a design methodology for systems that are subject to transient constraints. The methodology builds on surrogate modeling of the envelopes of the transient signals, using artificial neural networks. It enables the designer to visually and interactively perform design space exploration and optimization while ensuring that the transient constraints are met, thus avoiding potentially costly design iterations. For this purpose, VisTRE, an interactive visualization environment, is developed.

The design methodology, along with the visualization environment VisTRE, is tested on the design and optimization of a 350VDC electrical network, where the generator, its controller, and an actuator motor, are designed in order to minimize the electrical losses without violating the transient power quality constraint.

1 Introduction: Transient Regimes

Throughout the aircraft mission, a wide range of dynamic systems experience transient regimes. For instance, these transients represent a high proportion of the operation of the thrust reverser system, due to its short mission duration.

These transients can have severe impacts on the operation of the aircraft. For example, the switching on of a high electrical load might cause a network voltage drop inducing a loss of power available to critical aircraft systems. With the increasing interest in more-electric aircraft architectures, the potential problems arising from network transient perturbations become even more essential to address.

1.1 Transient Regimes and Dynamic Constraints

The design and development of systems are thus affected by the transient regimes the systems are expected to encounter. In particular, transient regimes may induce a peculiar type of constraints on design responses: these constraints are herein called “**dynamic constraints**”, as their value varies with time. An example of dynamic constraint related to transient regimes is given in Fig. 1, which depicts the transient constraint on the voltage of a 350 VDC network, as defined in the standards developed by the “Power Optimised Aircraft” (POA) research project [1][2]. From the instant at which the transient perturbation occurs (t=0), the voltage level has to stay within the bounds shown in the figure.

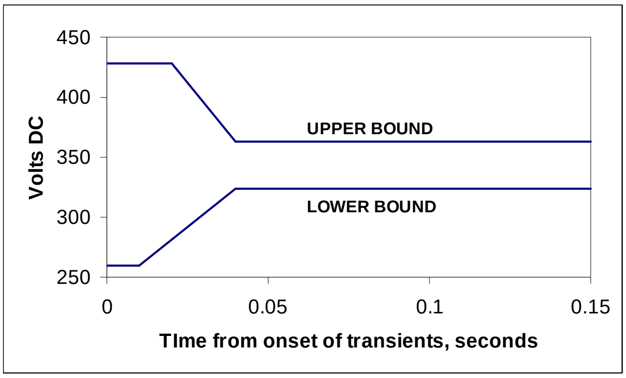

Fig. 1. Dynamic Transient Constraint for the Power Quality of a 350 VDC Network [1][2]

Dynamic signals are often multiscale in nature, as they may be composed of low frequency components (the trend) as well as high frequency components (the ripple). Thus, verifying that the signal meets the dynamic constraint amounts to verifying that its envelope stays within the bounds defined by the transient constraints. The envelope of a dynamic signal is composed of two signals, herein called the “sup-envelope” and the “inf-envelope”, as described in Fig. 2.

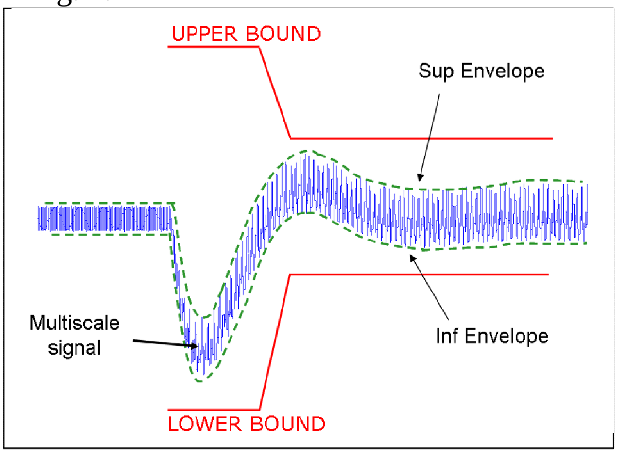

Fig. 2. Envelopes of the Transient Signal

1.2 Transient Constraints and Time-Domain Simulation Inefficiencies: a Challenge for the Development of Aircraft Systems

The verification of dynamic constraints generally requires high-fidelity Time-Domain Simulation (TDS) or testing on test rigs. Therefore, it intervenes in the later stages of the system development process, thus potentially causing costly design iterations and suboptimal designs. For example, testing on the electrical system of the International Space Station (ISS) revealed that power quality transient constraints were violated [3]. This triggered redesign activities aiming at validating the acceptance of less stringent constraints.

Time-domain simulation often consists of numerical resolutions of successive differential equations, which can make it time consuming. This becomes even more acute when the dynamic model under simulation exhibits behavior with multiple time scales, which is frequently the case for electrical networks that use high-frequency switching electronic devices. The inherent inefficiencies of time-domain simulation therefore pose a challenge for the efficient and thorough exploration of the design space of dynamic systems subject to transient constraints.

In order to avoid the design iterations due to the verification of transient constraints, there is a need for an innovative methodology that allows for the thorough design space exploration of the system, while eliminating the design cases that do not meet the dynamic transient constraints.

2 Research Objectives

The research objective of the research presented in this paper was to formulate a methodology that integrates the verification of transient dynamic constraints in the design of aircraft dynamic system. This methodology enables a *thorough* and *efficient* design space exploration for a dynamic system for which a time-domain simulation model has been formulated, as described in Fig. 3.

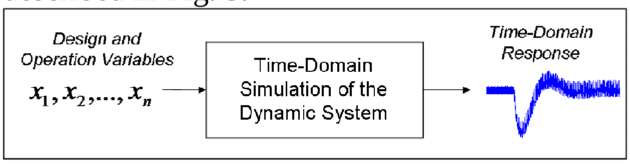

Fig. 3. Input Model of the Dynamic System

The application of the methodology yields a region of the design space (settings of design and operation variables) for which the transient constraints are satisfied and the other static responses (such as weight, cost, or electrical losses) are maximized. The capabilities of the design methodology are summarized in Table 1.

Table 1: Design Methodology Capabilities

- Determine design and operation regions $\{(x_1,\dots,x_n)\}$ that:
  - Optimize static responses (minimize weight, maximize efficiency, etc.)
  - Meets the transient constraints for time-domain responses (voltage power quality, transient torque constraints, etc.)
- Interactively visualize, explore, and query the design space:
  - Observe the transient behavior of time-domain signals
  - Perform “what-if” analysis

3 Design Methodology: Design Space Exploration

In order to perform a thorough design space exploration while accounting for dynamic transient constraints, a new methodology based on a data farming approach is developed. This methodology, which is based on the Filtered-Monte-Carlo methodology [4], consists of several steps that intervene after the time-domain simulation model of the system is created. A simplified overview of the methodology is provided in Fig. 4.

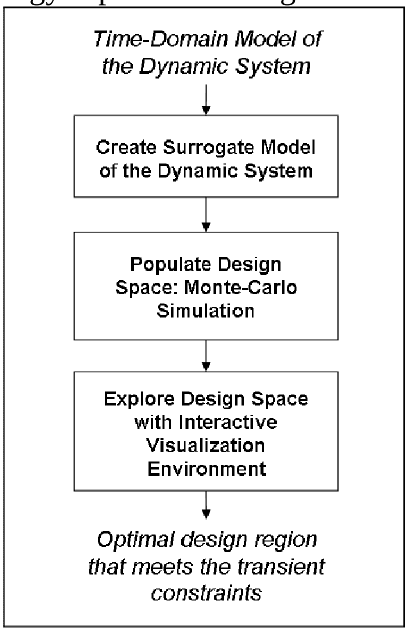

Fig. 4. Methodology Overview

As one can see on the figure, the methodology is composed of three main activities. First, surrogate models of the original time-domain simulation tool are created. Surrogate models are statistical approximations of analysis tools. As such, they are fast and efficient to run, and thus they enable to circumvent the run-time inefficiencies of time-domain simulation.

The second activity consists in using the surrogate models in order to compute the behavior of the system across the entire design space. For this purpose, a Monte-Carlo Simulation is performed, where uniform distributions are assigned to the design variables and design points are generated randomly.

Finally, the data points and behavior curves generated by the Monte-Carlo Simulation are imported and aggregated into a visualization environment that enables the efficient and thorough design space exploration. For this purpose, an innovative environment, called the Visual Transient Response Explorer (VisTRE), was developed.

Surrogate modeling and interactive visualization with VisTRE are the two major enablers of the design methodology. They are described in the following sections, which precede a more thorough description of the steps involved in the methodology.

3.1 Dynamic Surrogate Models

Because Time-Domain Simulation may be too inefficient to run for the implementation of a thorough design space exploration, surrogate models are used in order to approximate the Modeling and Simulation (M&S) tool described in Fig. 3. As explained previously, the transient dynamic constraints are verified on the envelope of the signal. Therefore, the desired surrogate model approximates the envelope of the signal rather than the signal itself, as illustrated in Fig. 5.

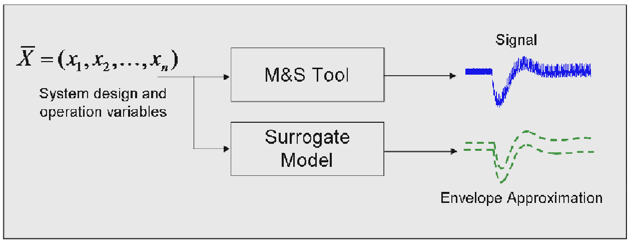

Fig. 5. Surrogate Modeling of the Envelope

There are several techniques for generating surrogate models, such as response surface modeling, Kriging networks, and Artificial Neural Networks (ANN's). ANN's are good at handling response nonlinearities, which are more and more present on electrical systems due to the increasing use of power electronics. Therefore, ANN's were selected as the approach for surrogate modeling.

ANN's, which mimic the way the brain functions, have become a popular choice for the regression of non-linear parameters, since it has been demonstrated that they can approximate any arbitrary continuous function [5][6]. They are structured as a collection of interconnected simple processing units called *perceptrons* or nodes, which act as a transfer function between an input variable and an output variable. For each connection between two perceptrons, there is an associated weight representing the strength of the connection. These perceptrons can be arranged in multiple ways to form more or less complex neural network architectures. The three-layered architecture, pictured in Fig. 6 and used in this paper, has been shown to exhibit the universal approximator property, if the perceptrons of the hidden layer correspond to the application of the logistic sigmoid function (given in Equation 1). Fig. 6 represents the approximation $\hat{y}$ of the time-domain signal (or its envelope) as a function of the design variables x_i and of time t. Because of the dependence on time, the resulting surrogate models are herein called "dynamic" surrogate models.

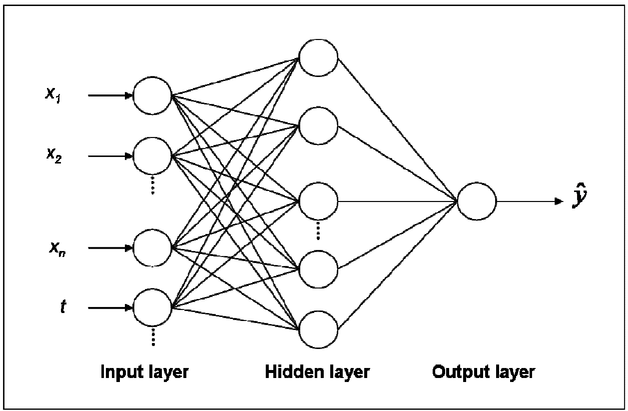

Fig. 6. Time-Domain Feedforward Artificial Neural Network

$$S(x) = \frac{1}{1+e^{-x}} \tag{1}$$

The output of each node is the result of the corresponding transfer function applied to a linear combination of the node inputs. Thus, the ANN pictured in Fig. 6 returns the calculated (predicted) value $\hat{y}$:

$$\hat{y} = c + \sum_{j=1}^{N_H}\left(d_j \cdot S\left(a_j + b_{oj}.t + \sum_{i=1}^{n}\left(b_{ij}x_i\right)\right)\right) \tag{2}$$

Where:
- a_j is the intercept term for the j^{th} hidden node
- b_{ij} is the coefficient for the i^{th} design variable
- x_i is the value of the i^{th} design variable
- t is the time
- n is the number of input variables
- c is an intercept term
- d_j is the coefficient for the j^{th} hidden node
- N_H is the number of hidden nodes
- S is the logistic sigmoid function

The process of fitting the responses is called learning, or training. Training the ANN is done by first selecting a number of nodes N_H in the hidden layer. Then the coefficients a, b, c and d are iteratively adjusted so that the resulting sum of squared errors between the predicted values and the actual values of the training set (observed values obtained after a "test campaign") is minimized. The resulting sum of squared errors (SSE) is given in Equation 3 below.

$$SSE \equiv \sum_i (y_i - \hat{y}_i)^2 \tag{3}$$

Where:
- y_i are the actual points of the training data set
- $\hat{y}_i$ are the predicted points calculated by the regression

Thus, the training of the ANN can be viewed as an optimization problem, where the objective function to be minimized is the error between the predicted and the actual response values, and where the variables of the optimization process are the coefficients a, b, c, d, and the number of hidden nodes N_H. At the end of the training process, the response y is approximated by an equation as defined in Equation 2.

3.2 The Visual Transient Response Explorer (VisTRE): An Interactive Visualization and Design Environment

The second major enabler that was developed for the design methodology is an innovative visualization environment called Visual Transient Response Explorer (VisTRE). VisTRE is an interactive tool developed in JMP, the statistical software package that links visualization tools to data tables.

In VisTRE, the design space is visualized and explored with two types of windows, as illustrated in Fig. 7. The first window, on the left-hand side of Fig. 7, is a multivariate scatterplot matrix, where the static parameters (variables and responses that do not depend on time) are visualized. The scatterplot essentially consists of an aggregation of plots of pairwise dimensions. In the scatterplot matrix, each row or column represents a parameter, whether it is an input parameter or a response, and these parameters are plotted against one another, thus forming cells. The example shown in Figure 31 is representative of a design problem with two design variables (X_1 and X_2) and two design responses (R_1 and R_2). Each point within a cell corresponds to an operating/design point, i.e. to a setting of the input parameters. One can see on the figure that in the cell plotting X_1 against X_2, the entire space is filled, which indicates that all the design regions are explored.

The scatterplot is interactive and the cells are linked to one another, so that any action of the user in one cell is instantly propagated to the other cells. Thus, when the user selects a point in a cell, the environment indicates the location of the corresponding design scenario in the other cells.

Another feature of the visualization environment is the ability for the user to add constraints on the parameters and instantly filter-out the design points that meet (or do not meet) the constraints. Feasibility regions can thus be highlighted and selected.

In VisTRE, the second type of visualization windows, shown on the right-hand side of Fig. 7, are overlay plots, where all the transient behavior curves are aggregated. In these overlays, each design case is no longer represented by a point in the cell, but by a behavior curve. Each cell corresponds to a particular transient event for a particular signal (voltage, current, motor speed), and plots the clustering of all the behavior curves that are taken by this signal when the design space is entirely sampled through a Monte-Carlo simulation.

The time-domain part (scatterplot) and the static cells are *interactively linked* so that the selection of a behavior curve in the time-domain cell triggers the selection of the corresponding points in the scatterplot cells, and vice versa. In each of these time-domain cells, the user can then add or modify dynamic constraints. In Fig. 7, dynamic constraints of the shape of the power quality constraints discussed in the introduction are shown as the red dashed lines.

The interactivity of the environment enables the quick investigation of several constraint scenarios and the visualization of how the design options are sensitive to the uncertainty on the constraints. The major capability of the VisTRE environment is the filtering of the design space: the user can instantaneously visualize the behavior curves that meet (or do not meet) the transient dynamic constraints, as illustrated in Fig. 8, where the design scenarios that violate the constraints are represented as red curves in the time-domain cells and as red points in the static scatterplot.

Finally, the user can instantaneously remove the undesirable points and curves in red so as to keep only those design scenarios that meet the dynamic constraints. From this point, the user can further refine the design space by filtering the remaining design points in the scatterplot and thereby keep the design cases that minimize or maximize the static responses (weight, cost, loss, efficiency, etc.).

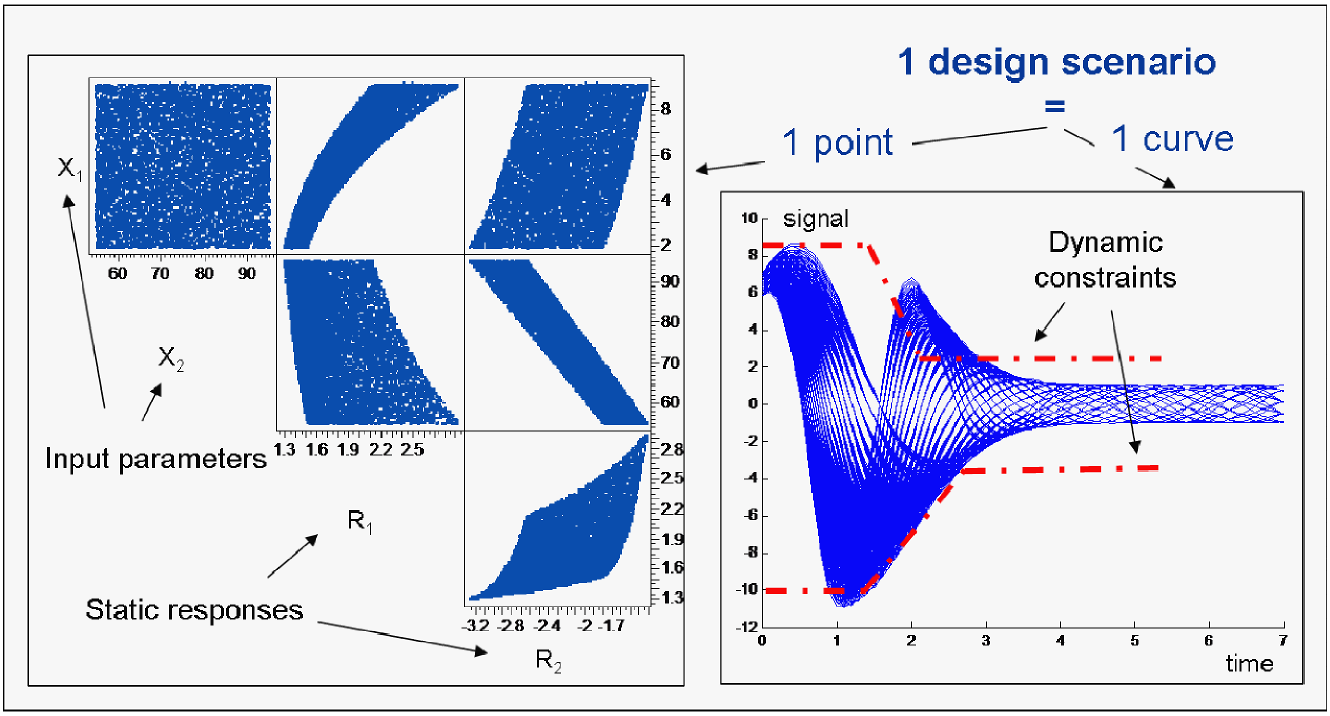

Fig. 7. Conceptual View of VisTRE: Unfiltered Design Space

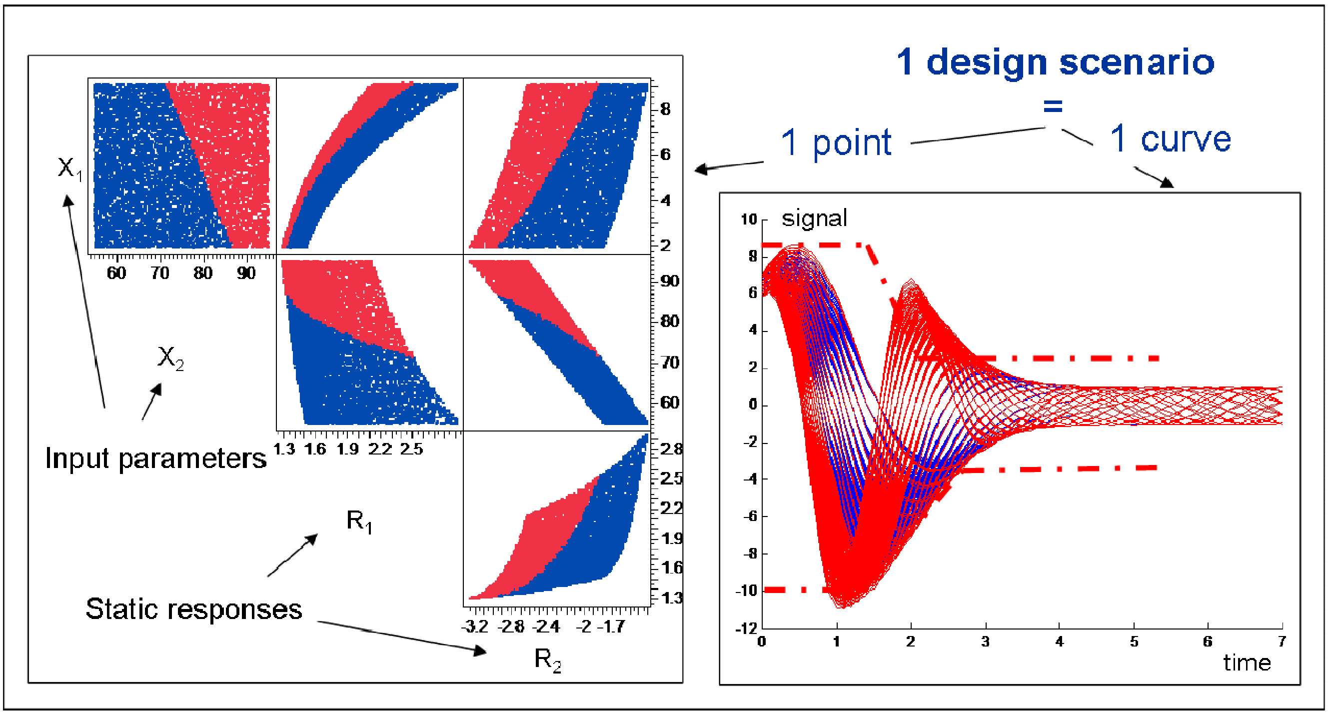

Fig. 8. Conceptual View of VisTRE: Filtered Design Space (Unfeasible Scenarios in Red)

3.3 Methodology: Detailed Description

In this section, a more thorough description is provided of all the elementary steps of the design methodology.

3.3.1 Step 1: Define the Problem

The first step aims at circumventing the inefficiency of the time-domain simulation model by creating a parametric surrogate model that approximates the dynamic behavior of tool. The output parameter is the time-behavior of the system response to a transient perturbation.

First, it is crucial to identify the parameters that will be considered during the design of the dynamic system. Most of these parameters will be directly derived from the requirements issued by the earlier design phases. These parameters can be grouped into two categories: the responses and the input variables.

The responses are the parameters that are being optimized or that are under constraints. For each response, the designer has to determine whether they are static or dynamic, and determine the constraints that regulate them.

The input variables are the controllable parameters that have an impact on the responses. For each response, the design team has to identify the factors influencing the response and their ranges of variation. For the dynamic responses, it is essential to determine the operation scenarios that might trigger transient behaviors, and determine the input variables that have an influence on the shape of these transient behaviors.

3.3.2 Step 2: Create the Modeling and Simulation Environment

The next step consists in setting up a Modeling and Simulation (M&S) environment. From the functional and physical specifications passed from the previous design phases, and in light of the parameters and constraints identified in Step 1, the design team now creates models capturing the behavior of the responses with respect to the input variables.

In general, at this phase, the design team has identified a physical decomposition that can potentially meet the design requirements. The system is thus composed of elementary components, for which simulation models are often already available in popular TDS software packages such as Simulink [7] or Dymola [8].

3.3.3 Step 3: Test Planning: Design of Experiments

Neural Networks are computed (trained) based on a set of observed signals (training set) generated by a test campaign. This step consists in creating the test plan for the test campaign, via a Design of Experiments (DoE).

DoE's are statistical techniques that allow for an efficient sampling of the design and operating space with a minimal number of tests [9]. Because of the way the variable settings are arranged, it is possible to isolate the effects of variables independently from one another. For a given number of input variables, there are several possible designs [9]. The choice of the design depends on the number of runs that can be afforded, and on where in the design and operating space the attention is focused. In the context of neural network training, Latin Hyper Cube (LHC) designs (or other space-filling designs) are generally favored, as they efficiently sample the entire input space without creating much bias of the resulting model towards a particular input variable. LHC's are a generalization of the well-known Latin Squares, in which two cells that belong to the same row or column cannot have the same value. In LHC's, the number of tests to be performed is selected in advance. A LHC design can be optimized in order to “spread” its points across the design space. Fig. 9 depicts an example of optimized LHC sampling in the case of two dimensions [10].

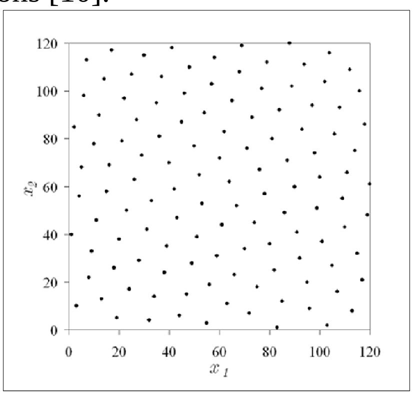

Fig. 9. Optimized Lating HyperCube Sampling [10]

The DoE produces a test table describing the runs that will be performed in order to produce data that will be used subsequently during training, in order to generate dynamic surrogate models. Once generated, these surrogate models will need to be validated. Therefore, the test plan produced in this step includes, besides the DoE cases (called training cases), a set of additional cases that will be run in order to obtain data points that will be used to validate the approximation capability of the dynamic surrogate models. These additional cases, called validation cases, are generated randomly.

3.3.4 Step 4: Test Campaign and Postprocessing

In Step 4, the test plan designed in Step 3 is carried out so as to obtain the transient behavior of the signals of interest for each test case, as well as the static responses that will be used to populate the static scatterplot of the Visual Transient Response Explorer (VisTRE), the interactive visualization environment defined in previous sections. The results of the test campaign are collected in matrices.

Then, data resulting from the test campaign needs to be processed in order to facilitate the training of the neural networks for the envelope signals. Thus, an envelope detection scheme, illustrated in Fig. 10, needs to be applied to each test case of the test campaign.

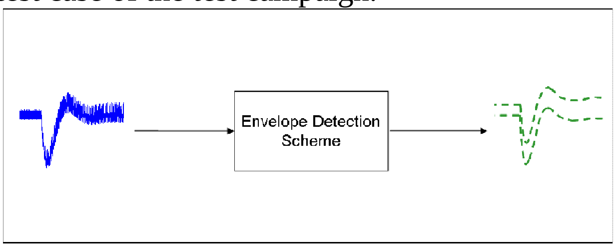

Fig. 10. Envelope Detection Scheme

The envelope detection scheme used in this paper is a two step process. First, the scales are separated in a processs similar to "denoising", by applying a technique called "Multi-Resolution Analysis" (MRA), which decomposes the signal into a low frequency signal (or "*trend*"), and a high frequency one (or "detail", or "*ripple*"), as illustrated in Fig. 11. MRA is based on wavelet theory, which enables to localize in time the frequency features of signals [11].

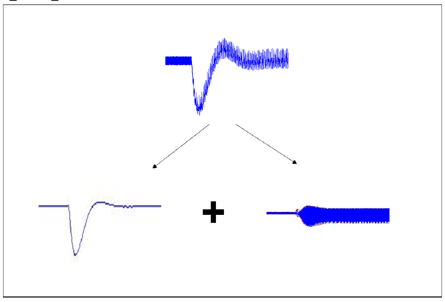

Fig. 11. Signal Denoising with Multi-Resolution Analysis

The envelope of the signal can then be viewed as the addition of the flatter envelope of the ripple and the trend signal. The ripple envelope, composed of two signals (upper ripple envelope and lower ripple envelope), is then computed using a sliding window method, depicted in Fig. 12.

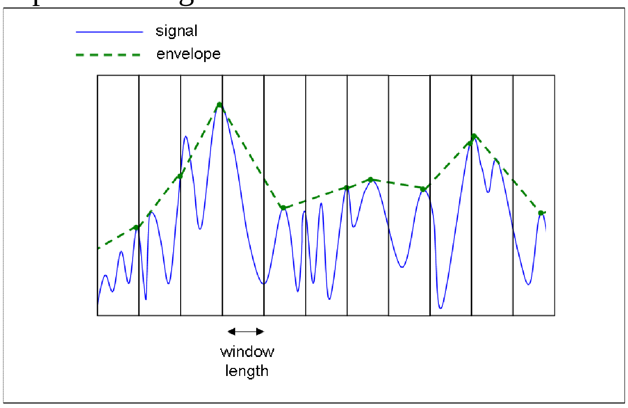

Fig. 12. Sliding Window Method

Thus, for each test case of the test campaign, the envelope detection scheme produces three time series per dynamic response: one for the trend of the signal, one for the sup-envelope of the ripple, and one for the inf-envelope of the ripple. These time series will be the input to the training activity of the sigmoid-based feedforward Artificial Neural Networks (ANN), as illustrated in Fig. 13.

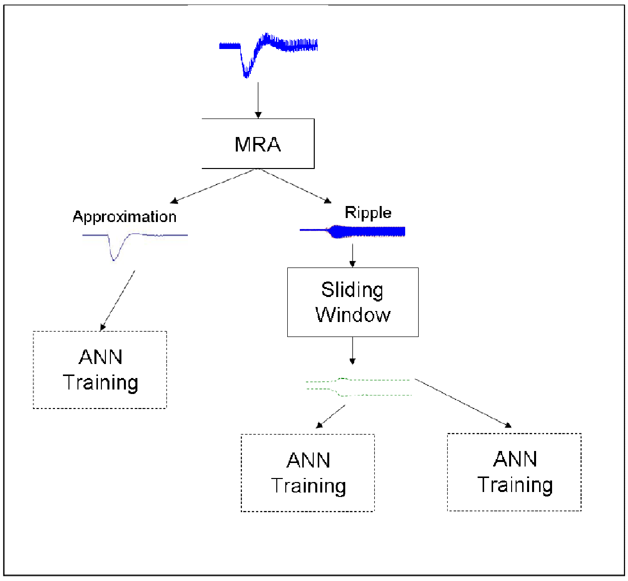

Fig. 13. ANN's for Surrogate Modeling of the Envelope

3.3.5 Step 5: Create Dynamic Surrogate Models of the Envelopes

In this step, the neural networks for the approximation of the signal envelopes are generated. The objective of the training (or "learning") process is to determine a neural network architecture that minimizes the error for the points corresponding to the training matrices (the outcomes of the test campaign). A neural network architecture is defined by the number of neurons in the hidden layer, and the weights a, b, c, d of the connections (or synapses) between neurons (cf. Fig. 6 and Equation 2).

First, a number of neurons in the hidden layer is assumed. Then, for this architecture, the synaptic weights a, b, c, and d are optimized in order to minimize the SSE-based error metric defined in Equation 3. The optimization process is carried out using the widely popular backpropagation algorithm [12].

3.3.6 Step 6: Monte-Carlo Simulation

The outcome of Step 5 is a set of dynamic surrogate models that approximate the transient behavior of the dynamic system. The design process can now leverage the efficiency of those surrogate models to generate the behavior curves needed for the thorough exploration of the design/operation space via the VisTRE environment.

In Step 6, these behavior curves are generated with a Monte-Carlo simulation., where uniform distributions are applied to the static input variables. The Monte-Carlo engine uses a random number generator to draw N_{MC} design/operation scenarios that sample the entire design/operation space. The number of random scenarios N_{MC} is chosen arbitrarily. It should be assigned a high enough value so that the generated design scenarios cover all regions of the design space. For each of these scenarios, the dynamic surrogate models are run, taking advantage of their efficiency. Thus, the outcome of Step 6 is a set of N_{MC} static data points (one set per static response) as well as a set of time-domain signals. For the time-domain cells, $3{*}N_{MC}$ behavior curves per dynamic response are generated: N_{MC} signals representing the trend of the transient response, N_{MC} signals representing the sup-envelope of the signal ripple, and N_{MC} signals representing the inf-envelope of the signal ripple.

3.3.7 Step 7: Populate Interactive Visualization Environment

The previous step generated $3{*}N_{MC}$ behavior curves per dynamic response (voltage, current, etc.). In this step, the $3{*}N_{MC}$ behavior curves are recombined to generate the desired N_{MC} envelopes, as depicted in Fig. 14.

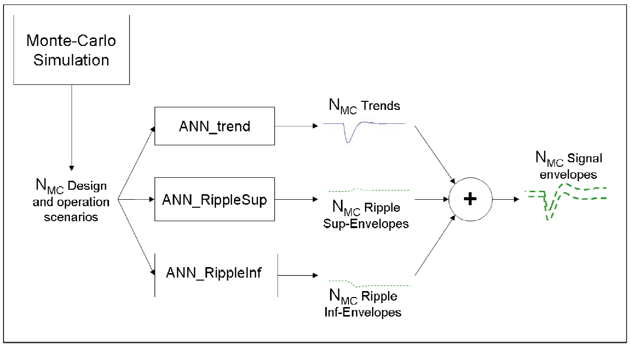

Fig. 14. Envelope Reconstruction after Monte-Carlo Simulation

Then, the data (design points, static responses, and dynamic envelopes), which now samples the entire design space, is imported into the interactive visualization environment VisTRE.

3.3.8 Step 8: Explore and Filter the Design/Operation Space

At this stage, the behavior curves and the static data points have been imported into the interactive visualization environment VisTRE. The user can now take advantage of the capabilities of the environment to explore the design/operation space, and visualize correlations between parameters.

In addition to the insight on the design/operation space that it may offer, the interactive visualization environment allows the designer to add static and dynamic constraints, and filter-out the static design points and dynamic behavior curves that do not meet these constraints, as previously explained (cf. Fig. 7 and Fig. 8). The result is therefore a set of design/operation points that meet the design and operation constraints. The designer can further restrict the design/operation space by selecting the design/operation points that correspond to optimal values of design criteria (e.g. minimal weight, minimal cost).

4 Application: Design of a 350VDC Electrical Network

In this section, the full methodology is implemented for the design of a simple 350VDC electrical network, composed of a controlled generator that feeds power to an actuator motor. The variations of mechanical torque demanded by the actuator to the motor induce variations of power demand from the motor, which in turn induce transient perturbations on the electrical network. Thus, power quality constraints will apply to the network voltage, which has to be controlled in order to be maintained at a level of 350VDC. The network is depicted in the Simulink block diagram of Fig. 15. During the simulation, the actuator motor torque experiences a step increase, which induces an undervoltage of the network voltage, of the same shape as that of Fig. 2. The network voltage is subject to the transient dynamic constraint pertaining to power quality given in Fig. 1.

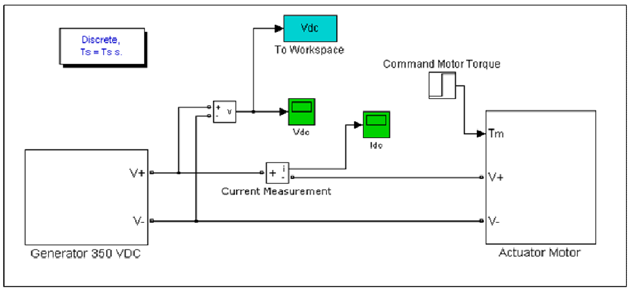

Fig. 15. Simulink Model of the Electrical Network

The candidate architecture for the generator system is comprised of several subsystems. First, a Permanent Magnet Generator (PMG), which converts mechanical power provided by a rotating shaft into a three-phase alternating electrical signal. The three-phase current is then converted into a DC voltage by an Insulated-Gate Bipolar Transistor (IGBT) rectifier. IGBT's are fast-switching power devices that are increasingly used in power electronics [13]. The generator system also includes a Generator Control Unit (GCU). A simple Proportional-Integral (PI) control strategy with two gains K_P and K_I was chosen.

The actuator motor system is composed of a Permanent Magnet Motor (PMM), fed by a three-phase electrical signal that is the output of an inverter, which converts DC current into alternating three-phase currents. The inverter implemented here is based on Metal-Oxide-Semiconductor Field-Effect Transistors (MOSFET), which are used to switch or amplify signals.

The goal of the design of the 350VDC electrical network is to specify the design characteristics of the generator and its controller, as well as those of the motor, in order to minimize the electrical losses of the generator. The electrical loss is represented by a simple model, which takes the average value of the instantaneous Joules losses induced by the phase currents i_A, i_B and i_C in the stator of the Permanent Magnet Generator.

A total of 17 design and operation variables were identified: they characterized the generator, the controller gains, the actuator, and the actuator motor step torque level. By applying the design methodology to the system, Artificial Neural Networks were created for the envelope (sup-envelope and inf-envelope) and the full design space was populated via a Monte-Carlo Simulation. As described in the methodology, the resulting data points and data curves were imported into the innovative visualization environment VisTRE.

The design space was then filtered in VisTRE, so that only the design regions verifying the transient constraints were kept. This is illustrated in Fig. 16, which shows the time-behavior of the inf-envelope of the network voltage for the remaining design cases. One can see that at any instant, the values taken by the voltage (in blue) fall above the lower bound transient constraint (in red).

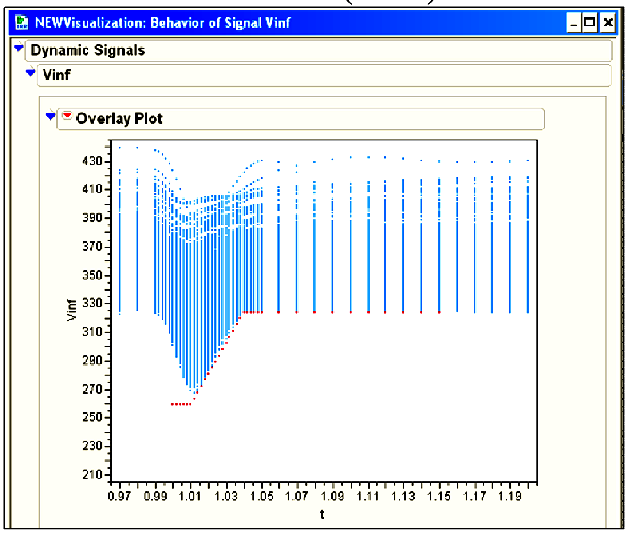

Fig. 16. VisTRE Snapshot: Behavior of the Inf-Envelope of the Network Voltage, after Filtering

Finally, the design space was further restricted to select the design points that minimized the electrical losses visualized in the scatterplot matrix. Thus, as a result of the application of the design methodology described in this paper, the electrical losses were minimized while ensuring that the transient constraints remained satisfied.

5 Conclusion

Transient regimes experienced by dynamic systems may have severe impacts on the operation of the aircraft. They are often regulated by dynamic constraints, requiring the dynamic signals (such as voltage or current) to remain within bounds whose values vary with time. The verification of these peculiar types of constraints, which generally requires high-fidelity time-domain simulation, intervenes late in the system development process, thus potentially causing costly design iterations.

The research objective of this work was to develop a methodology that integrates the verification of dynamic constraints in the early specification of dynamic systems. In order to circumvent the inefficiencies of time-domain simulation, multivariate dynamic surrogate models of the original time-domain simulation models are generated in order to emulate the transient behavior of the envelope of the time-domain responses in a quick and efficient manner. As an intermediate step, an envelope extraction method was therefore formulated, building on a wavelet-based Multi-Resolution Analysis scheme. The Dynamic surrogate models are created by training sigmoid-based Artificial Neural Networks that include time as an input variable.

The run-time efficiency of the resulting dynamic surrogate models enables the implementation of a data farming approach, in which the full design space is sampled through a Monte-Carlo Simulation. The Visual Transient Response Explorer (VisTRE), an interactive visualization environment enabling what-if analyses, was developed; the user can thereby instantaneously comprehend the transient response of the system (or its envelope) and its sensitivities to design and operation variables, as well as filter the design space to have it exhibit only the design scenarios verifying the dynamic transient constraints.

The methodology formulated herein was tested on the design and optimization of a 350VDC network, where a generator and its control system were concurrently designed in order to minimize the electrical losses, while ensuring that the transient undervoltage induced by peak demands in the consumption of a motor does not violate transient power quality constraints.

6 References

[1] Power Optimised Aircraft, contract G4RD-2001-00601 under the European Communities Framework Programme for Research.

[2] Felix M and Routex, J-Y. A Copper Bird for Aircraft Equipment Systems Integration and Electrical Network Characterization, *AIAA 2007-1392 ,45th AIAA Aerospace Sciences Meeting and Exhibit*, Reno NV, 2007.

[3] Aintablian H, Fassburg H, and Soendker E, International Space Station (ISS) Truss Orbital Replaceable Unit (ORU) Overvoltage Transient Resolution, *37th Intersociety Energy Conversion Engineering Conference*, Washington DC, 2002.

[4] Ender T, *A Top-Down, Hierarchical, System-of-Systems Approach to the Design of an Air Defense Weapon*, PhD thesis, Georgia Institute of Technology, 2006.

[5] Hornik K, Stinchcombe M, and White H, Multilayer Feedforward Networks are Universal Approximators, *Neural Networks*, Vol. 2, no. 2, pp 359-366, 1989.

[6] Phan L.L, Mavris D.N, Charrier J.-J, Thalin P, and Garcia E, Parametric Modeling of an Electrical Test Rig for Power Optimized Aircraft Architectures, *26th International Congress of the Aeronautical Sciences (ICAS) including 8th AIAA Aviation Technology, Integration, and Operations Conference (ATIO), Anchorage, Alaska*, 2008.

[7] MathWorks, Simulink, Version 7.3., *http://www.mathworks.com/products/simulink/*, 2009.

[8] Dassault Systemes, Dymola (Dynamic Modeling Laboratory), http://www.3ds.com/products/catia/portfolio/dymola, 2010.

[9] Montgomery DC. *Design and Analysis of Experiments*, 3rd edition, John Wiley & Sons, Inc., 1991.

[10] Bates S.J, Sienz J and Toropov V.V, Formulation of the Optimal Latin Hypercube Design of Experiments using a Permutation Genetic Algorithm, *45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference*, Palm Springs, California, April 2004.

[11] Mallat, S.G, A Theory for Multiresolution Signal Decomposition: the Wavelet Representation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 674-693, July 1989.

[12] Rojas R, *Neural Networks - A Systematic Introduction*, Springer-Verlag, 1996.

[13] Khanna, V.K, *The Insulated Gate Bipolar Transistor: IGBT Theory and Design*, Wiley-IEEE Press, 2003.

7 Acknowledgements

The authors wish to thank Nathalie Chantelat and Philippe Poret, of Hispano-Suiza (Safran Group), for their continuous support and their feedback.

8 Contact Author Email Address

The author may be contacted at the following email address:
leon.phan@supaero.org

9 Copyright Statement