VI International Conference on Adaptive Modeling and Simulation
ADMOS 2013
J. P. Moitinho de Almeida, P. Díez, C. Tiago and N. Parés (Eds)

STOCHASTIC MODEL REDUCTION APPLIED TO INVERSE ANALYSIS

TOMASZ GARBOWSKI*

*Poznan University of Technology
Institute of Structural Engineering
ul. Piotrowo 5, 60-695, Poznan, Poland
e-mail: tomasz.garbowski@put.poznan.pl

Key words: Model approximation, Gaussian Processes, Inverse Analysis

Abstract. This article describes the use of Gaussian Processes in model reduction techniques with application to inverse problems. The work is focused on the proper construction of the model approximation, namely on training process based on limited number of learning samples. By making use of Active Learning criterion employed in training process a significant improvement in model prediction can be observed. An example of application of stochastic surrogate model for the paperboard characterization through biaxial tensile test and Digital Image Correlation measurements is also presented.

1 INTRODUCTION

The inverse analysis is a technique widely used for structure or material characterization, especially when unknown or uncertain parameters embedded in the model have to be determined. Knowing the structure responses such as displacements, accelerations, eigenmodes, etc. the backward computations can be performed to identify causes, e.g. constitutive constants, structural parameters. In the literature there are many examples of application of inverse analysis for material characterization [1, 2, 3], damage detection [4, 5], estimation of residual stresses [6, 7], to list just a few.

The inverse analysis often uses a numerical model (e.g. finite element or boundary element model, analytical model) for test simulation, which computes selected quantities to be compared to experientially measured ones. The inverse procedure, through iterative optimization algorithms, minimizes the discrepancy between experimentally measured $\mathbf{U}_{\text{EXP}}$ and numerically computed $\mathbf{U}_{\text{NUM}}$ quantities, so the minimization problem reads:

$$\arg\min_{\mathbf{x}\in\mathbb{R}^n} \left\|\mathbf{U}_{\text{NUM}}(\mathbf{x}) - \mathbf{U}_{\text{EXP}}\right\|_2^2 \tag{1}$$

where $\mathbf{x}$ is a set of sought parameters.

If the numerical model is complex and/or has to be computed many times, the iterative minimization procedure becomes very expensive, therefore, not attractive from practical point of view especially when the test has to be performed routinely 'in situ' (i.e. without a computer which can handle heavy computations). The alternative is to use a surrogate which approximates the behavior of the numerical model but is much simpler, thus less expensive. The surrogate is usually constructed as a 'black box' where for the approximation the following methods, among others, are commonly used: Radial Basis Functions (RBFs), Polynomials, Proper Orthogonal Decomposition (POD) combined with RBFs [8, 9], Artificial Neural Networks (ANNs) [10] or Gaussian Processes (GP) [11, 12].

2 MODEL REDUCTION THROUGH GAUSSIAN PROCESSES

2.1 Model approximation

In order to build a smooth and accurate analytical approximation one needs to construct a forward model to generate responses (i.e. training samples):

$$\mathbf{x}_n \Rightarrow \left\{ \begin{matrix} \text{Forward} \\ \text{Model} \end{matrix} \right\} \Rightarrow \mathbf{u}\left(\mathbf{x}_n\right) \tag{2}$$

where $\mathbf{x}_n = \left[x_1, x_2, \ldots, x_M\right]_n^T$ is an input vector (i.e. vector of sought parameters) and $\mathbf{u}\left(\mathbf{x}_n\right) = \mathbf{u}_n = \left[u_1\left(\mathbf{x}_n\right), u_2\left(\mathbf{x}_n\right), \ldots, u_k\left(\mathbf{x}_n\right)\right]^T$ is an output vector (i.e. vector of measurable quantities) for $n = 1, \ldots, N$. Both vectors $\{\mathbf{x}_n, \mathbf{u}_n\}$ represents a single training sample and form n-th column of $(M \times N)$ parameter matrix $\mathbf{X}$ and $(K \times N)$ snapshot matrix $\mathbf{U}$:

$$\mathbf{X} = \begin{bmatrix} x_{11} & \cdots & x_{1N} \\ \vdots & \ddots & \vdots \\ x_{M1} & \cdots & x_{MN} \end{bmatrix}, \qquad \mathbf{U} = \begin{bmatrix} u_{11} & \cdots & u_{1N} \\ \vdots & \ddots & \vdots \\ u_{K1} & \cdots & u_{KN} \end{bmatrix}, \tag{3}$$

Ideally would be to find method which needs the smallest possible number of 'training samples' and in the same time is precise and robust. The approximation method based on Gaussian Processes satisfies all these requirements: it gives very good results when the number of training examples is limited. Another important feature of GP is that it gives not only the approximation of the mean value of sought solution but also its standard deviation. This feature gives a possibility of automatic and systematic improvement of the solution, because in the locations with high values of standard deviation one can expect a weak approximation and therefore, it points out where, in the parameter space, an additional experimental or numerical data is necessary to improve the solution.

The GP employed for stochastic approximation is usually formulated within Bayesian framework, thus provides additional information about the magnitude of correlation between state variables and control variables. It is very important to know the relevance of input-output correlation because based on it one can exclude from the model the parameters which do not influence the measurable quantities. The reduction saves the

experimental efforts of finding parameters which appear to be irrelevant in particular simulation. Such information obtained 'for free' during the training process is similar to the results from sensitivity analysis often performed for model check.

The stochastic model reduction techniques based on GP have, however, one significant disadvantage, namely the Gaussian Processes are usually parameterized in terms of their covariance functions. This makes it difficult to deal with multiple outputs, because ensuring that the covariance matrix is positive definite is problematic. An alternative formulation is to treat Gaussian processes as white noise sources convolved with smoothing kernels, and to parameterize the kernel instead (see [13]). Using this approach, one can extend Gaussian Processes to handle multiple, coupled outputs.

When GP are applied for model reduction (here for the forward model approximation) within the inverse problem, the number of measurable quantities (e.g. measured displacement fields, velocities or accelerations in different moments of time and in various space locations) is often very large. In such situation the probable correlations between the state variables can be computed, and consequently used to reduce the number of model outputs by the application of POD, i.e.:

$$\mathbf{U} \Rightarrow \big\{\mathrm{POD}\big\} \Rightarrow \bar{\mathbf{U}} \tag{4}$$

where

$$\bar{\mathbf{U}} = \begin{bmatrix} u_{11} & \cdots & u_{1N} \\ \vdots & \ddots & \vdots \\ u_{L1} & \cdots & u_{LN} \end{bmatrix}, \tag{5}$$

with $L \ll K$. However, the problem size, even when reduced, does not necessarily make possible an efficient application of multi-output GP. An alternative is to formulate the GP approximation on modified training samples, which instead of output consisting of truncated measurable quantities $\{\mathbf{x}_n, \bar{\mathbf{u}}_n\}$, have an output containing a particular scalars $\{\mathbf{x}_n, t_n\}$. The scalar value output can be chosen for example as a 'squared distance' from the experimental measurements to their numerically computed counterparts, namely:

$$t_n = \|\mathbf{u}_n - \mathbf{u}_{\mathrm{EXP}}\|_2^2 . \tag{6}$$

By adopting a latter method the GP approximation can be constructed in two stages. First N training samples $\{\mathbf{x}_n, \mathbf{u}_n\}$ are computed through forward model (2). Later for each particular identification procedure an experimental data $\mathbf{u}_{\mathrm{EXP}}$ are used for output truncation through (6) to a single scalar (so the training data is now $\{\mathbf{X}, \mathbf{T}\}$, where $\mathbf{T} = [t_1, t_2, \ldots, t_N]$). Having a scalar output in each training sample the single-output GP approximation can be easily constructed. On this stage an additional retraining by an active learning method (see e.g. [11]) can be performed in order to improve the surrogate. Once the GP approximation is designed, any optimization algorithm can be used to find a function minimum according to formula (1). Because the surrogate in general approximates a multi-modal function (i.e. function with many local minima)

it seems reasonable to employ for function minimization some global search techniques such as genetic algorithms [14] or particle swarm methods [15]. Such algorithms require usually a big number of iterations in order to find a global minimum, however, when used on reduced model the computing time is not an issue anymore.

2.2 Linear regression model

In order to explain how to construct a model approximation by Gaussian Processes, first a Linear Regression (LR) model should be considered. LR is a linear function of model parameters $\mathbf{w}$ and nonlinear function of the input vector $\mathbf{x}$, and usually is defined as:

$$y\left(\mathbf{x},\mathbf{w}\right)=\sum_{j=1}^{M}w_j\phi_j\left(\mathbf{x}\right), \tag{7}$$

which simply is a linear combination of fixed, nonlinear basis functions $\phi_j(\mathbf{x})$ of the input variables (e.g. polynomial basis functions).

If we now take N given training patterns $(\mathbf{x}_n, t_n)$, $\mathbf{x}_n$ being the input vector, t_n the response for $n = 1 \ldots N$, then the parameters $\mathbf{w}$ of the linear model can be computed by, for example, penalized least squares method:

$$\mathbf{w}=\left(\mathbf{\Phi}^T\mathbf{\Phi}+\lambda\mathbf{I}\right)^{-1}\mathbf{\Phi}^T\mathbf{t}, \tag{8}$$

where $\mathbf{\Phi}$ is $N\times M$ design matrix with elements defined as $\phi_m(\mathbf{x}_n)$. The regularization parameter λ is called hyper-parameter and can be estimated using validation set or by applying Bayesian inference and maximizing evidence of dataset $p(\mathbf{t}|\lambda)$ w.r.t. λ (details are given in [12]).

2.3 Gaussian Process

Gaussian process model can be obtained by reformulation of the linear model in terms of dual representation. In this approach, linear model is trained by minimizing a regularized error, which is defined using $N\times N$ symmetric Gram matrix:

$$\mathbf{K}=\mathbf{\Phi}\mathbf{\Phi}^T=\phi(\mathbf{x})^T\phi(\mathbf{x}')=k(\mathbf{x},\mathbf{x}'), \tag{9}$$

where $k(\mathbf{x},\mathbf{x}')$ is a kernel function. The vector $\mathbf{k}_n = k(\mathbf{x}_n,\mathbf{x})$ represents n-th row or column of $\mathbf{K}$ matrix.

The prediction for a new input $\mathbf{x}^*$ can be computed by the formula:

$$\mathcal{GP}\left(\mathbf{x}^*|\mathbf{x},\mathbf{t},\lambda\right)=k(\mathbf{x},\mathbf{x}^*)^T(\mathbf{K}+\lambda\mathbf{I})^{-1}\mathbf{t}, \tag{10}$$

where $k(\mathbf{x},\mathbf{x}^*)$ is a covariance between a new input $\mathbf{x}^*$ and the other inputs, $\mathbf{t}=(t_1\ldots t_N)^T$ is a vector of training target values.

From the Bayesian point of view the dual representation of linear model leads to the Gaussian process, where the kernel function is interpreted as a covariance function of the

GP. Application of such regression model for prediction allows to compute the predictive distribution of the target variable $y(\mathbf{x}^*)$ for a new input vector $\mathbf{x}^*$. This requires evaluation of conditional distribution $p(y|\mathbf{t})$, which for the Gaussian processes is a Gaussian distribution with mean and covariance respectively given by:

$$\text{mean}\,(\mathbf{x}^*) = \mathbf{k}^T\mathbf{C}^{-1}\mathbf{t}, \tag{11}$$

$$\sigma^2(\mathbf{x}^*) = c - \mathbf{k}^T\mathbf{C}^{-1}\mathbf{k}, \tag{12}$$

where $\mathbf{C}$ is the $N\times N$ covariance matrix given by:

$$C(\mathbf{x},\mathbf{x}') = k(\mathbf{x},\mathbf{x}') + \beta^{-1}\mathbf{I}, \tag{13}$$

where β is the variance of the target distribution and $\mathbf{I}$ is an identity matrix. The covariance function $C(\mathbf{x},\mathbf{x}')$ defines the property that vectors $\mathbf{x}$ and $\mathbf{x}'$, which are close in input space, should give rise to highly correlated outputs $y(\mathbf{x})$ and $y(\mathbf{x}')$.

2.4 Covariance function

The covariance function can be any function that will generate a non-negative definite covariance matrix for any ordered set of (input) vectors $(\mathbf{x}_1,\ldots\mathbf{x}_N)$. A stationary, non-isotropic squared exponential covariance function $k\,(\mathbf{x},\mathbf{x}')$ is chosen here, and given by:

$$k\,(\mathbf{x},\mathbf{x}') = \nu\exp\left(-\frac{1}{2}\sum_{i=1}^{M} w_i\,(x_i - x_i')^2\right) + b, \tag{14}$$

where the term b represents a bias that controls the vertical offset of the Gaussian process, while ν controls the vertical scale of the process. The w_i parameters allow a different distance measure for each dimension. If w_i is small then the i-th input is down-weighted and have little effect on the input. These hyper-parameters play vary important role mainly because they have a direct link to model sensitivities with respect to input parameters thus provide a measure of importance of input parameters.

After defining the covariance function we can make predictions of the new input vectors but first it is necessary to learn the hyper-parameters

$$\boldsymbol{\theta} = [\nu,\omega_1,\ldots,\omega_M,b,\beta]\,. \tag{15}$$

In order to find those parameters one can search for the most probable set by maximizing the log likelihood function given by:

$$\ln p(\mathbf{t}|\boldsymbol{\theta}) = \frac{1}{2}\ln|\mathbf{C}| - \frac{1}{2}\mathbf{t}\mathbf{C}^{-1}\mathbf{t} - \frac{N}{2}\ln 2\pi, \tag{16}$$

using gradient-based optimization algorithms, such as a first-order batch Levenberg-Marquardt Algorithm (LMA) or Trust Region Algorithm (TRA), which provides fast convergence (see details in [16]).

2.5 Application of GP to inverse problems

Once the N training samples $\{\mathbf{X}, \mathbf{U}\}$ is generated through forward model (2) and later adjusted to the particular experimental data $\mathbf{u}_{\text{EXP}}$ by the formula (6) so the training set becomes $\{\mathbf{X}, \mathbf{T}\}$ one can learn hyper-parameters $\boldsymbol{\theta}$ of covariance matrix $C(\mathbf{x}, \mathbf{x}')$, thus construct GP approximation $\mathcal{GP}(\mathbf{X}, \mathbf{T}, \boldsymbol{\theta})$ of forward model.

From this point, by making use of equations (11) and (12), the prediction of GP output t^* for a new parameter vector $\mathbf{x}^*$ can be computed through conditional distribution

$$\mathcal{GP}(\mathbf{x}^*|\mathbf{X}, \mathbf{T}, \boldsymbol{\theta}) = \left\{\text{mean}(t^*) = \mathbf{k}^T\mathbf{C}^{-1}\mathbf{T}, \quad \text{covariance}(t^*) = c - \mathbf{k}^T\mathbf{C}^{-1}\mathbf{k}\right\}. \tag{17}$$

The mean value and covariance of model prediction are computed by simple matrix multiplications, however, if number of sampling points is large, the construction and inversion of covariance matrix have to be carefully designed in order to retain an efficiency.

2.6 Active learning

If one would like to improve an approximation adding new training samples, the specific training process, which automatically finds new locations of sampling points has to be employed. In order to optimize the retraining procedure of GP based surrogate model, the active learning criterion, that improve the global model fit, can be implemented. A new training points are sequentially added in the zones where the model predictions are poor and/or in the vicinity of the minimum of the approximated solution, meaning that the iterative retraining of the surrogate model is performed by adding a new training patterns, sampled in the new locations in the parameter space (selected by the algorithm itself).

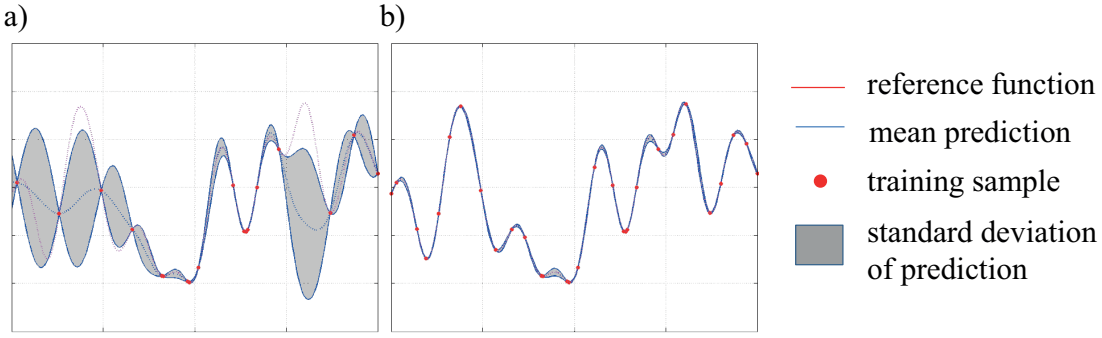

Figure 1: An example of 1D function approximation by GP and active learning algorithm: (a) 15 samples approximation, (b) 28 samples approximation

This approach is very efficient when the number of training samples is limited, so the algorithm starts building approximation with small selection of patterns (randomly or uniformly distributed) and improves the approximation by sampling the parameter space in a clever way (based on its confidence about the quality of the approximation). It stops

when the selected number of retraining samples is reached. In Fig. 1 the active training process is shown.

3 EXAMPLE

An example of a model characterization is used here to illustrate the application of above described model reduction techniques. The examples show the use of GP as numerical model's surrogate for characterization of paperboard parameters [17] through biaxial test and DIC measurements techniques combined with inverse analysis. From such test one can identify most of the in-plane parameters of paperboard (for the details see [10]).

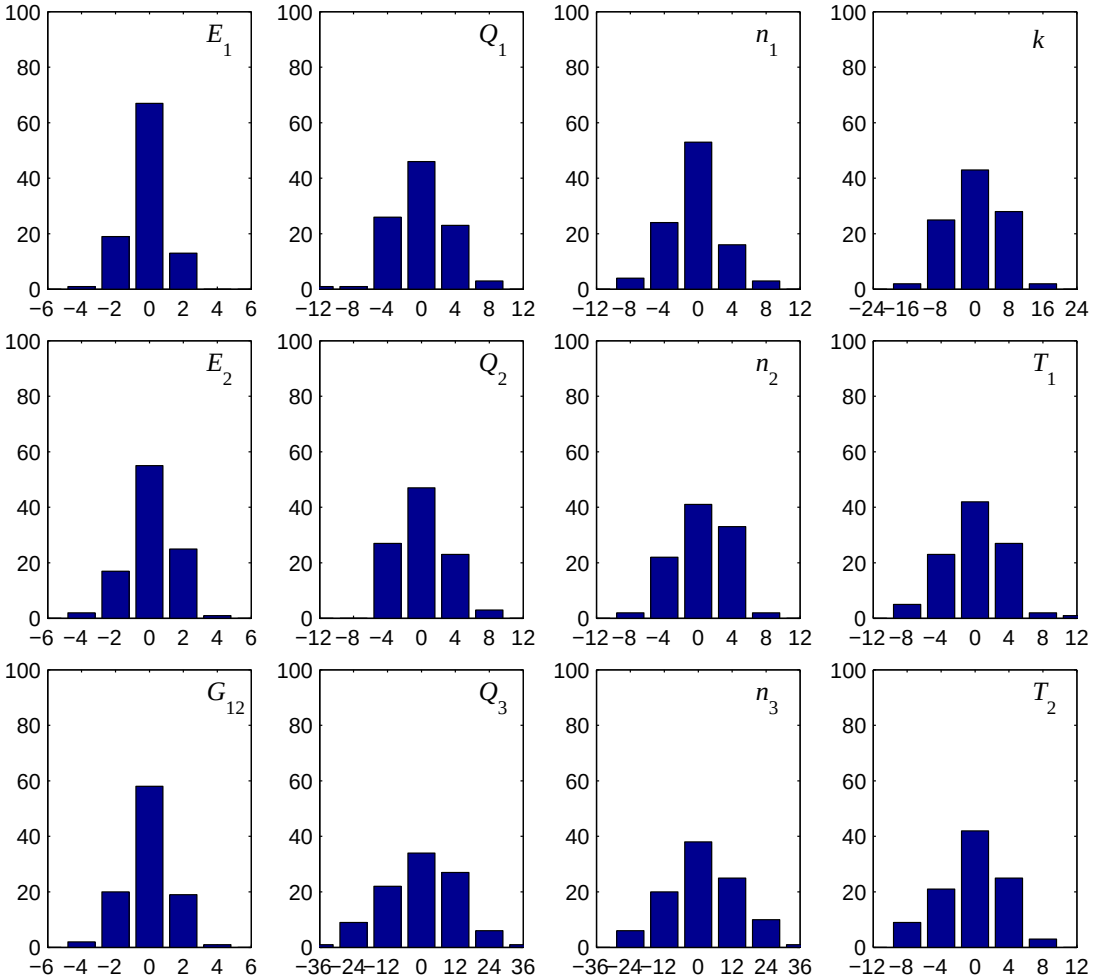

Figure 2: Performance of the GP designed for identification of 12 parameters in Xia *et al.* model with measurement noise ±0.5 µm: percentage of relative error in abscissae; in ordinates, percentages of results within each abscissae interval

Figure 2 shows the performance of GP trained and later retrained by Active Learning

algorithm for each (over one hundred) pseudo-experimental testing sample.

In the above examples the GP based approximation serves as a surrogate of complete Finite Element (FE) model, which in combination with iterative minimization routine, namely trust-region algorithm (TRA) and evolutionary-based techniques, i.e. particle swarm algorithm (PSA) gives very accurate results of model parameters identification. The results are compared to those obtained by making use of Artificial Neural Networks (details on application of ANNs are given in [10]). Both ANNs and GP approximations are constructed on 10.000 numerically generated samples additionally noised and truncated to DIC measurements precision. The sampling points are distributed in the 12-dimensional parameter space according to Optimal Latin Hypercube technique [18].

The GP model is further retrained by making use of 100 new testing data. Each testing sample is used to truncate the output matrix $\mathbf{U}$ through formula (6) to the vector $\mathbf{T}$. Active Learning criterion inserted in training algorithm automatically adds new training points (totally 1.000) in the argument space according to rules specified in previous section. In Tab. 1 the comparison of ANNs performance with GP trained by TRA and GP trained by PSA with active learning criterion is presented.

Table 1: Relative approximation error on 100 testing samples by (a) Artificial Neural Networks; (b) Gaussian Processes and Trust Region Algorithm and (c) GP combined with Active Learning and Particle Swarm Algorithm

Parameter	ANN	GP (TRA)	GP+AL (PSA)
E_1	1.29	1.49	1.11
E_2	1.55	1.61	1.25
G_{12}	1.66	1.81	1.15
Q_1	4.22	4.10	2.63
Q_2	4.05	3.92	2.89
Q_3	14.3	15.1	12.0
n_1	3.74	3.70	2.16
n_2	4.66	4.32	2.62
n_3	15.9	14.7	9.86
k	7.28	7.41	5.38
T_1	3.20	4.04	2.40
T_2	3.50	4.07	2.23

4 SUMMARY

The model reduction approach has the following important advantages w.r.t. the procedure based on evaluation of full numerical model: (a) it is much faster (the computational burden is moved to training phase) and (b) it does not require to use powerful computers for heavy and repetitive computations of numerical model, so once the model

is constructed it can be used 'in situ' on the portable computer.

The GP approximation model, which serves as a numerical model reduction, was used here in combination with inverse analysis to solve practical engineering identification problems. In the work the proper construction of the GP model was discussed, namely: (a) training process based on minimal number of training data, by making use of automatic samples selection through computed standard deviation of model prediction; (b) control parameters reduction based on input-output correlation (sensitivity-like analysis) and (c) state variables 'compaction' for single-output GP implementation. The successful application of stochastic model reduction techniques for the material model characterization problems was also presented and compared to other approximation method based on Artificial Neural Networks.

REFERENCES

[1] Bolzon, G., Maier, G. and Panico, M. Material model calibration by indentation, imprint mapping and inverse analysis. *International Journal of Solids and Structures*, (2004) **41**: 2957–2975.

[2] Ardito, R., Maier, G. and Massalongo, G. Diagnostic analysis of concrete dams based on seasonal hydrostatic loading. *Engineering Structures*, (2008) **30**: 3176–3185

[3] Maier, G., Buljak, V., Garbowski, T., Cocchetti, G. and Novati G. Mechanical characterization of materials and diagnosis of structures by inverse analyses: some innovative procedures and applications. *International Journal of Computational Methods* (2013) submitted.

[4] Knitter-Piatkowska, A., Pozorski, Z. and Garstecki, A. Application of discrete wavelet transformation in damage detection. Part I: Static and dynamic experiments. *Computers Assisted Mechanics and Engineering Sciences* (2006) **13**:21–38.

[5] Ziopaja, K., Pozorski, Z. and Garstecki, A. Damage detection using thermal experiments and wavelet transformation. *Inverse Problems in Science and Engineering* (2011) **19, 1**:127–153.

[6] Bocciarelli, M. and Maier, G. Indentation and imprint mapping method for identification of residual stresses. *Computational Materials Science*, (2007) **39**:381–392.

[7] Buljak, V. and Maier, G. Identification of residual stresses by instrumented elliptical indentation and inverse analysis. *Mechanics Research Communications*, (2012) **41**: 21–29

[8] Maier, G., Bolzon, G., Buljak, V., Garbowski, T. and Miller, B. *Chapter 24. Synergistic combinations of computational methods and experiments for structural diagnosis.* Computer Methods in Mechanics. Lectures of the CMM 2009, Eds. M. Kuczma, K. Wilmanski, Springer-Varlag Berlin Heidelberg, (2010) 453–476.

[9] Garbowski, T., Maier, G. and Novati, G. Diagnosis of concrete dams by flat-jack tests and inverse analyses based on proper orthogonal decomposition. *Journal of Mechanics of Materials and Structures*, (2011) **6, 1-4**:181–202.

[10] Garbowski, T. , Maier, G. and Novati, G. On calibration of orthotropic elastic-plastic constitutive models for paper foils by biaxial tests and inverse analyses. *Structural and Multidisciplinary Optimization*, (2012) **46, 1**:111–128.

[11] C.E. Rasmussen, C.K.I. Williams. *Gaussian Processes for Machine Learning*. MIT Pres, (2006).

[12] Bishop, C.M. *Pattern Recognition and Machine Learning*. Springer, (2007).

[13] Alvarez, M., Lawrence, N. Sparse Convolved Gaussian Processes for Multi-output Regression. *Twenty-Second Annual Conference on Neural Information Processing Systems (NIPS), Vancouver, Canada*, (2008).

[14] Burczynski, T., Kus, W., Dlugosz, A. and Orantek, P. Optimization and defect identification using distributed evolutionary algorithms *Engineering Applications of Artificial Intelligence*, (2004) **17, 4**: 337–344

[15] Kennedy, J. Particle swarm optimization. *Proceedings., IEEE International Conference on Neural Networks*, (1995) **4**: 1942–1948

[16] Nocedal, J., Wright, S. *Numerical Optimization*. Springer, (2006).

[17] Xia, Q.S., Boyce, M.C., Parks, D.M. A constitutive model for the anisotropic elastic-plastic deformation of paper and paperboard. *International Journal of Solids and Structures*, (2000) **61**:1–48.

[18] Park, J.-S. Optimal Latin-hypercube designs for computer experiments, *Journal of Statistical Planning and Inference*, (1994) **39, 1**: 95–111