



Proper Generalized Decomposition Based Dynamic Data-Driven Control of Material Forming Processes

Chady Ghnatios, Françoise Masson, Antonio Huerta, Elías Cueto, Francisco Chinesta

► To cite this version:

Chady Ghnatios, Françoise Masson, Antonio Huerta, Elías Cueto, Francisco Chinesta. Proper Generalized Decomposition Based Dynamic Data-Driven Control of Material Forming Processes. 14th ESAFORM International Conference on Material Forming, 2011, Belfast, United Kingdom. 10.1063/1.3589673 . hal-01008703

HAL Id: hal-01008703

<https://hal.science/hal-01008703>

Submitted on 11 Mar 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Proper Generalized Decomposition Based Dynamic Data-Driven Control of Material Forming Processes

C. Ghnatios*, F. Masson*, A. Leygue[†], E. Cueto** and F. Chinesta*

**EADS Corporate Foundation International Chair*

GeM : Ecole centrale de Nantes, Université de Nantes

1 rue de la Noe, BP 92101, F-44321 Nantes cedex 3, France

Chady.Ghnatios@ec-nantes.fr; Francoise.Retat@ec-nantes.fr; Francisco.Chinesta@ec-nantes.fr

[†]GeM : Ecole centrale de Nantes, Université de Nantes

1 rue de la Noe, BP 92101, F-44321 Nantes cedex 3, France

Adrien.Leygue@ec-nantes.fr

***Campus Rio Ebro. - Edificio Agustin de Betancourt*

University of Zaragoza

Maria de Luna, 7, E-50018 Zaragoza, Spain

ecueto@unizar.es

Abstract. Dynamic Data-Driven Application Systems - DDDAS - appear as a new paradigm in the field of applied sciences and engineering, and in particular in simulation-based engineering sciences. By DDDAS we mean a set of techniques that allow the linkage of simulation tools with measurement devices for real-time control of systems and processes. DDDAS entails the ability to dynamically incorporate additional data into an executing application, and in reverse, the ability of an application to dynamically steer the measurement process. DDDAS needs for accurate and fast simulation tools using if possible offline computations to limit as much as possible the online computations. We could define efficient solvers by introducing all the sources of variability as extra coordinates in order to solve offline only once the model to obtain its most general solution, to be then considered for online purposes. However, such models result defined in highly multidimensional spaces suffering the so called curse of dimensionality. We proposed recently a technique, the Proper Generalized Decomposition - PGD-, able to circumvent the redoubtable curse of dimensionality. The marriage of DDDAS concepts and tools and PGD "offline" computations could open unimaginable possibilities in the field of dynamics data driven application systems. In this work we explore some possibilities in the context of process control, malfunctioning identification and system reconfiguration.

INTRODUCTION

Traditionally, Simulation-based Engineering Sciences - SBES- relied on the use of static data inputs to perform the simulations. However, a new paradigm in the field of Applied Sciences and Engineering has emerged in the last decade known as DDDAS. A dynamic data driven application system or DDDAS "entails the ability to dynamically incorporate additional data into an executing application, and in reverse, the ability of an application to dynamically steer the measurement process" [1]. While research on DDDAS should involve many applications and algorithms, our work focuses on the development of

mathematical and statistical algorithms for the simulation within the framework of such a system. In brief, we intend to incorporate a new generation of simulation techniques into the field, allowing to perform faster simulations, able to cope with uncertainty, multiscale phenomena, inverse problems and many other features. This new generation of simulation techniques has received the name of Proper Generalized Decompositions -PGD- and has received an increasing level of attention by the SBES community [2]. The PGD yields an approximate solution in the separated form :

$$u(x_1, \dots, x_d) \approx \sum_{i=1}^N F_i^1(x_1) \times \dots \times F_i^d(x_d) \quad (1)$$

Thus the complexity of the solution scales linearly with the dimension of the problem [3], and not exponentially as for classical algorithms. The iterative constructor assumes the first N functions products already computed, therefore, looking for the next term, i.e.:

$$u(x_1, \dots, x_d) \approx \sum_{i=1}^N F_i^1(x_1) \times \dots \times F_i^d(x_d) + F_{N+1}^1(x_1) \times \dots \times F_{N+1}^d(x_d) \quad (2)$$

Before solving the resulting non linear model related to the calculation of these functions, a model linearization is performed. The simplest choice consists in using an alternating directions fixed point algorithm. First of all, we proceed by assuming $F_{N+1}^2(x_2), \dots, F_{N+1}^d(x_d)$ given at the previous iteration of the non-linear solver and then computing $F_{N+1}^1(x_1)$. From the just updated $F_{N+1}^1(x_1)$ and previously used $F_{N+1}^3(x_3), \dots, F_{N+1}^d(x_d)$ we can update $F_{N+1}^2(x_2)$ and so on until reaching convergence [3]. For more information about the construction of the PGD, readers can refer to [3] [4].

In this paper, we propose an alternative approach in optimisation and real-time control using PGD, to our knowledge never explored. In fact, we propose to consider the unknown process parameters as new coordinates of the model. Thus, we obtain a very general solution for all possible parameters, inside a fixed domain.

Once the general solution computed, we can consider optimization. Various optimisation methods have been tested, here we show the Levenberg-Marquardt method, a fast and reliable minimization algorithm designed for non-linear least squares [5]. The same method is used while controlling the process. In this part of the work, we introduce the combination of "offline" heavy computation with "online" light control computations based on machine feedback. The work scheme is illustrated in figure (1).

OFFLINE COMPUTATIONS

In this section we are introducing the main ideas related to process optimization through the analysis of a quite simple thermal model. Despite the apparent simplicity, the strategy here described can be extended for addressing more complex scenarios. The steady state temperature field $u(\mathbf{x})$ in any point of the die $\mathbf{x} = (x, y) \in \Omega \subset \mathcal{R}^2$ can be obtained from the solution of the 2D heat transfer equation that involves advection and diffusion

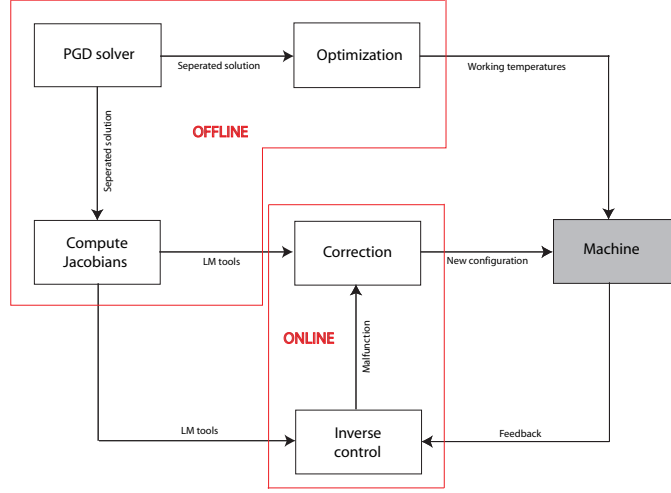


Figure 1. The global work scheme

mechanisms as well as an eventual source term Q . The velocity field is everywhere unidirectional, i.e. $\mathbf{v}^T = (v, 0)$ and then the heat transfer equation reduces to:

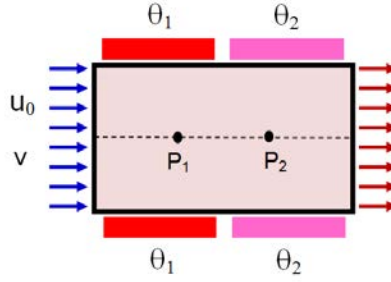


Figure 2. Material forming process illustration

$$\rho \cdot c \cdot \left(v \cdot \frac{\partial u}{\partial x} \right) = k \cdot \Delta u + Q \quad (3)$$

where k is the thermal conductivity, ρ is the density and c is the specific heat. The die is equipped with two heating devices as depicted in figure (2) whose flux constitute the process parameters to be optimized. For simplicity's sake we consider constant the internal heat generation Q , the velocity v and the inlet temperature u_0 , all of them assumed known. The prescribed boundary conditions write:

$$\begin{cases} u(x=0, y, \theta_1, \theta_2) = u_0 \\ \nabla u(x \in L_1, y=0 \text{ or } y=H, \theta_1, \theta_2) \mathbf{n} = \theta_1 \\ \nabla u(x \in L_2, y=0 \text{ or } y=H, \theta_1, \theta_2) \mathbf{n} = \theta_2 \end{cases} \quad (4)$$

\mathbf{n} being the unit outward vector defined on the domain boundary, L_1 and L_2 are the length of heater 1 and 2 respectively. A null heat flux is assumed in the remaining part of the domain boundary.

In the optimisation procedure, we consider only the coldest thermal history of an imaginary material particle traversing the die, thus the cost function is written as :

$$\mathcal{C}(\theta_1, \theta_2) = \frac{1}{2} \left(\int_0^L u \left(x, y = \frac{H}{2}, \theta_1, \theta_2 \right) dx - \beta \right)^2 = \frac{1}{2} (f(\mathbf{p}) - \beta)^2 \quad (5)$$

where L is the die length and β is defined as the optimal value able to give a certain material transformation. During optimization, having a separated representation of the solution enables to easily compute gradients and jacobians, thus permitting the use of minimization algorithms, such as Levenberg-Marquardt. This algorithm is briefly described below :

1. We define $\mathbf{p} = (\theta_1, \theta_2)$, thus cost function writes :

$$\mathcal{C}(\mathbf{p}) = \frac{1}{2} (f(\mathbf{p}) - \beta)^2 \quad (6)$$

2. Computation of the Jacobian at iteration k :

$$\mathbf{J}^k = \begin{pmatrix} \left. \frac{\partial f(\mathbf{p})}{\partial \theta_1} \right|_{\theta_1^k, \theta_2^k} & \left. \frac{\partial f(\mathbf{p})}{\partial \theta_2} \right|_{\theta_1^k, \theta_2^k} \end{pmatrix} \quad (7)$$

3. Compute the residual at iteration k :

$$r^k = f(\mathbf{p}^k) - \beta \quad (8)$$

4. Update the optimal values vector $\mathbf{p}^{k+1} = (\theta_1^{k+1}, \theta_2^{k+1})$ according to:

$$\mathbf{p}^{k+1} = \mathbf{p}^k - \left(\left(\mathbf{J}^k \right)^T \mathbf{J}^k + \lambda \mathbf{I} \right)^{-1} \cdot \left(\mathbf{J}^k \right)^T \cdot r^k \quad (9)$$

where \mathbf{I} is in the present case the 2×2 unit matrix.

5. If the cost function decreases very fast we can reduce the relaxation coefficient λ and then the Levenberg-Marquardt algorithm approaches the Newton one. On the other hand, if the cost function decreases slowly we can increase the value of λ to approach a classical gradient strategy. We consider a standard adaptation of such coefficient.

The solution given by this algorithm is shown in figure (3).

ONLINE COMPUTATIONS

We now consider two positions on the symmetry axis, $\mathbf{P}_1 = (1, 0.5)$ and $\mathbf{P}_2 = (2, 0.5)$ (see figure (2)). The resulting temperature at those points, when the heating devices runs optimally, i.e. when $\theta_1 = \theta_1^{op}$ and $\theta_2 = \theta_2^{op}$ are: $u^{op}(\mathbf{P}_1) = u(x=1, y=0.5, \theta_1^{op}=1030, \theta_2^{op}=686.9) = 264.2$ and $u^{op}(\mathbf{P}_2) = u(x=2, y=0.5, \theta_1^{op}=1030, \theta_2^{op}=686.9) =$

361.7. In order to control the process we could introduce a thermocouple at position \mathbf{P}_1 and another one at position \mathbf{P}_2 . As long as both "online" measurements \tilde{u}_1 and \tilde{u}_2 give values close enough to $u^{op}(\mathbf{P}_1)$ and $u^{op}(\mathbf{P}_2)$ respectively, the process can be considered working in optimal conditions, the one related to the optimal flux of both heating devices. In this section, we simulated a breakdown by assuming a malfunction of the second heating device, that instead of imposing the flux $\theta_2^{op} = 686.9$ only applies a fraction of this optimal value, in our numerical exercise we considered $\theta_2 = 0.4 \times \theta_2^{op} = 274.8$. If we particularize the general solution in the location of the thermocouples we obtain: $\tilde{u}(\mathbf{P}_1) = u(x = 1, y = 0.5, \theta_1^{op} = 1030, \theta_2 = 274.79) = 210.3$ and $\tilde{u}(\mathbf{P}_2) = u(x = 2, y = 0.5, \theta_1^{op} = 1030, \theta_2 = 274.7) = 266.9$.

Inverse analysis

The analysis for identifying the flux imposed by the heating devices from the sensors' measurements defines an inverse problem. We propose to proceed by minimizing "online" the following cost function:

$$\mathcal{C}(\theta_1, \theta_2) = \frac{1}{2} \sum_{i=1}^{i=2} (\tilde{u}(\mathbf{P}_i) - u(x_i, y = 0.5, \theta_1, \theta_2))^2 \quad (10)$$

where x_i ($i = 1, 2$) are the coordinates of the points at which the thermocouples are located.

Again, we apply the Levenberg-Marquardt algorithm for minimizing the cost function (10). In the present minimization problem we have $\mathbf{p} = (\theta_1, \theta_2)$. However in this case we define a vector $\mathbf{f}(\mathbf{p}) = \mathbf{f}(\theta_1, \theta_2)$ as:

$$\mathbf{f}(\mathbf{p}) = \begin{pmatrix} f_1(\mathbf{p}) \\ f_2(\mathbf{p}) \end{pmatrix} = \begin{pmatrix} u(x_1, y = 0.5, \theta_1, \theta_2) \\ u(x_2, y = 0.5, \theta_1, \theta_2) \end{pmatrix} \quad (11)$$

such that the cost function (10) can be rewritten as:

$$\mathcal{C}(\theta_1, \theta_2) = \frac{1}{2} (\tilde{\mathbf{u}} - \mathbf{f}(\mathbf{p}))^2 \quad (12)$$

After three iterations we obtain an estimation of the flux on both heating devices $\tilde{\theta}_1 = 1033.5$ and $\tilde{\theta}_2 = 269$ that agree with the scenario that we considered ($\theta_1 = 1030$, $\theta_2 = 274.7$). The inverse identification runs very fast and it only involves slight calculations, so it could be performed "online" and using very light computing devices, as for example a smartphone.

Process reconfiguration

Finally, we should reconfigure the process to his optimum, by selecting the new values of the flux of both heating devices θ_1^* and θ_2^* respectively. To ensure the optimization of the process, we will minimize again the cost function expressed in equation (5). There

are many possibilities that a control strategy could evaluate automatically before choosing the optimal reconfiguration, but we are considering here only a simple possibility. Our action consists of keeping the second heating device in its present state, i.e. $\theta_2^* = 146$ and looking for the optimal value of θ_1^* minimizing the cost function:

$$\mathcal{C}(\theta_1^*) = \int_0^L u\left(x, y = \frac{H}{2}, \theta_1^*, \theta_2 = 269\right) dx - \beta \quad (13)$$

We perform the minimization by using the Levenberg-Marquardt algorithm that in three iterations converges to the value $\theta_1^* = 2871,3$. Figure 3 depicts the resulting temperature field related to the new optimal process parameters $\theta_1^* = 2871,3$ and $\theta_2^* = 269$.

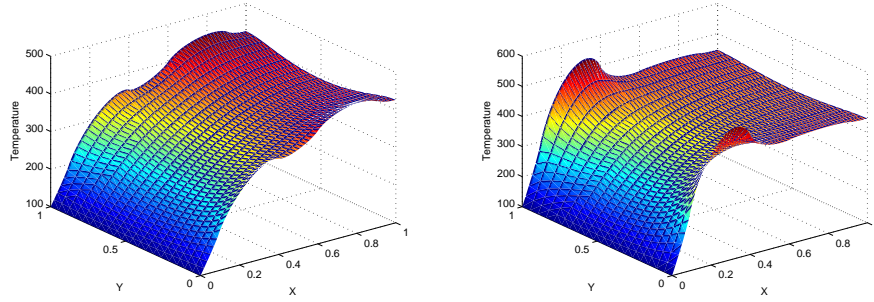


Figure 3. Multidimensional solution particularized for the optimal temperatures to the left and after reconfiguring the system to the right

CONCLUSIONS

This work presents a first attempt of applying dynamic data driven simulation for controlling industrial processes whose modelling involves complex linear or non-linear partial differential equations. In the numerical examples here addressed "offline" calculations need around two minutes of computing time (using matlab, a standard laptop and a non optimized simulation code) whereas all the "online" calculations were performed in 1.5 ms. The examples here addressed are too simple to be conclusive, but at least, they prove the pertinence of the proposed approach.

REFERENCES

1. V. authors, Dddas workshop 2006, Tech. rep., National Science Foundation, Arlington, VA, U.S.A. (2006).
2. H.-J. Bungartz, and M.Griebel, *Acta Numerica* **13**, 1–123 (2004).
3. F. Chinesta, A. Ammar, and E. Cueto, *Archives of Computational Methods in Engineering* **17**, 327–350 (2009).
4. E. Pruliere, F. Chinesta, and A. Ammar, *Mathematics And Computer Simualtion* **81**, 791–810 (2010).
5. D. Marquardt, *SIAM Journal on Applied Mathematics* **11**, 431–441 (1963).