

# Non Linear Dimension Reduction of Dynamic Model Output

Claire-Eleuthèriane Gerrer<sup>1</sup> Sylvain Girard<sup>1</sup>

<sup>1</sup>Phimeca Engineering, France, {gerrer, girard}@phimeca.com

## Abstract

Most advanced mathematical methods for the analysis of numerical model cannot cope with functional outputs of dynamic Modelica models. Principal component analysis is a well established method for dimension reduction, and can be used to tackle this issue. It relies however on a linear hypothesis that limits its applicability. We illustrate on a case study how the non linear method of auto-associative model overcomes this shortcoming and provides physically interpretable data representations.

*Keywords:* dimension reduction, functional data analysis, FMI, OtFMI, principal component analysis, auto-associative model, sensitivity analysis.

## 1 Introduction

The advent of the *functional mock-up interface* (FMI) and the emergence of associated tools considerably facilitated the analysis of Modelica models with advanced mathematical methods. Sensitivity analysis, model emulation, Bayesian inference and the like can now be performed routinely using scripting language such as Python (Girard and Yalamas; Girard et al., 2018) except for a substantial hurdle: many Modelica models are *dynamic*, and functional outputs are difficult to handle. Dimension reduction is a means to sidestep this difficulty. Principal component analysis (PCA) is by far the most prominent method for dimension reduction. This almost one century old statistical learning method (Hotelling, 1933) has been applied in virtually all fields where data are available. It is easy to implement and to understand, and relatively robust. It relies however on the hypothesis that the variables at hand can be aggregated into linear combinations, which unfortunately is not true for many dynamic model outputs. We illustrate this issue on a simple case study, and show how an alternative approach of more general applicability, the auto-associative model (AAM), allows to overcome it. Finally, we show how low dimension representations produced by AAM can be leveraged to get insights about the modelled physical phenomena.

## 2 Why reducing the dimension of dynamic model?

A computer experiment is the analysis of the output of a model obtained by varying its inputs according to a design of experiment. Modelica models are more often

than not *dynamic*, namely their outputs are functions of time. Discretised time functions are high dimensional vectors which considerably obstructs the analysis. First, it is subjected to the “curse of dimensionality” (Houle, 2015), namely a variety of undesirable consequences of increasing the dimension. For instance, the volume of a cube increases exponentially with its dimension, and sample size required to densely fill it become quickly prohibitive. Distances in large dimension spaces loose their discriminating power, especially when the component variables are correlated, which is specially true for discretised time functions. Indeed, it is not straightforward to compare curves as it is with numbers. In statistical analysis, modelling the joint distribution of a set of more than 4 dependent variables, for instance using kernel estimation, is generally intractable.

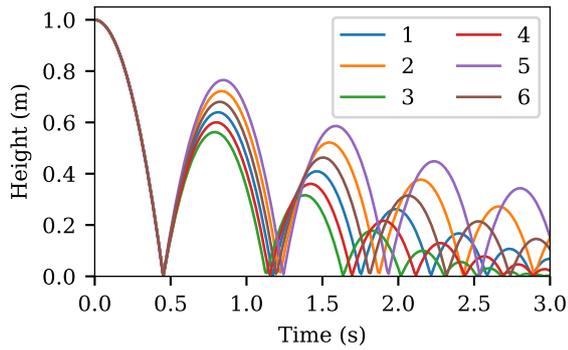
Actually, the great majority of mathematical methods involved in computer experiments apply to models with scalar outputs. For instance, sensitivity analysis (Saltelli et al., 2008) aims at measuring the relative influence of the inputs on an output. Applying sensitivity analysis to each of them individually yields sensitivity indices that are functions of time: the output values at each chosen time step can be considered as distinct output variables. This approach to sensitivity analysis, sometimes deemed “*sequential*” (Girard, 2014, chapter 7) has its merits but is difficult to interpret.

Model emulation (also known as meta-modelling or surrogate modelling) is another technique that cannot cope with high dimensional outputs. It consists in substituting a CPU inexpensive mathematical approximation for a numerical model in order to achieve large sample size required for instance by some optimisation techniques, or for Bayesian parameter estimation, or to enable instantaneous interaction with the model. Kriging is an example of method for emulating numerical models (Roustant et al., 2012).

A common expedient to enable analysing functional outputs is to project them on a function basis (Campbell et al., 2006). When there is no obvious candidate, principal component analysis allows to automatically build an adapted basis.

## 3 Linear dimension reduction with principal component analysis

The geometric approach to PCA provides the most intuitive understanding of the method. The discretisation in  $d$



**Figure 1.** The first 6 trajectories of the training set when only the coefficient of restitution varies.

steps of  $N$  realisations of a functional model output can be seen as a point cloud of  $N$  points in  $\mathbb{R}^d$ . PCA then finds the axes along which data spread the most. These axes, called *principal directions*, have the property to minimise the distances between the points of the point cloud and their projection on the axes (Jolliffe and Cadima, 2016).

Each principal direction defines a linear combination of the initial  $d$  variables called *principal components*. The projection of the data points along the principal directions are called *scores*.

The principal directions of the data set are sequentially built, so as to be mutually orthogonal. The set of the principal directions form a new basis in the space  $\mathbb{R}^d$ . The  $k$  first principal directions,  $k \in \{1, \dots, d\}$ , form a basis of the linear subspace of dimension  $k$  that best contains the scatter plot. Thus, PCA finds the linear subspace of given dimension (or hypercube, because the span of the data is usually limited) that best contains the point cloud.

### 3.1 PCA of the bouncing ball model

We applied PCA to a set of 128 trajectories of the famous bouncing ball model adapted by Tiller (2015)<sup>1</sup>: a ball is dropped from a given height and bounce back touching the ground with a fraction of the velocity it acquired during the fall determined by a fixed *coefficient of restitution*.

The trajectories were simulated with coefficient of restitution sampled between 0.7 and 0.9. following a Sobol’ sequence (Sobol’, 1979) so as to avoid redundancies. We used an FMU generated with OpenModelica, and the OtFMI Python module<sup>2</sup> (Girard, 2017) to carry out the simulations. Figure 1 displays the first 6 trajectories of this training set. All trajectories coincide before the first bounce at 0.45 s and increasingly deviate from one another at each subsequent bounce.

We simulated the next 896 ( $= 1024 - 128$ ) trajectories of the Sobol’ sequence to serve as a test sample for evaluating the performance of PCA. They were discretised at 300 evenly spaced time steps. Because the model has a

<sup>1</sup>The adapted bouncing ball model is available at <http://book.xogeny.com/behavior/discrete/bouncing/>.

<sup>2</sup><https://github.com/openturns/otfmi>.

single input, the intrinsic dimension of the set of trajectories, namely the smallest number of parameters required to fully parametrise it, is 1. The test trajectories were projected onto the first principal direction and compared to their original counterpart. The top panel of Figure 2 compares the worst reconstructed trajectory to the original. Here “worst” understands as resulting in the biggest root mean squared error (RMSE) between reconstruction and original. It must be noted however that the first 28 test trajectories sorted by decreasing RMSE are very similar to one another, as well as to those sorted by decreasing absolute error or relative error. Beyond that rank, the absolute error ranking diverges substantially from the two others. One principal component is clearly insufficient to capture the diversity of the trajectories: the reconstructed trajectory does not even touch the ground after the second bounce. Indeed, the middle and bottom panel show that the absolute and relative reconstruction errors with a single principal component are outside. As expected, the error is null before the first bounces. It then displays a complex oscillatory pattern, ensuing from both the physical phenomenon and the sampling scheme. Interestingly, the absolute error globally increase as time goes by, despite the lessening of average height.

### 3.2 Time delays, a major stumbling block for PCA

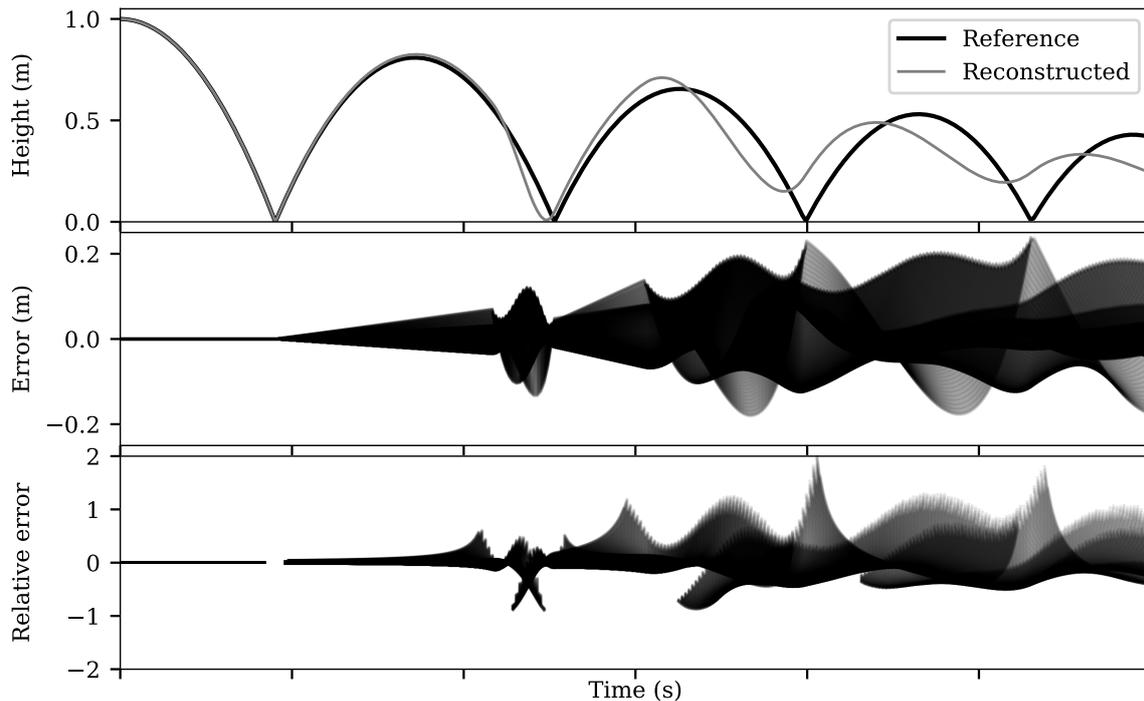
What happens here is that the the point cloud of trajectories has a *linear* dimension much greater than 1. It is a one dimensional manifold extending in multiple directions in  $\mathbb{R}^{300}$ . As such, it cannot be “enclosed” in a line. Figure 3 illustrates the result of increasing the number of retained principal components (left panel). The decrease in all three error measures (absolute, relative and RMSE) is rather slow. For instance, a reduction to dimension 4 still results in a substantial number of relative errors greater than 50 %.

PCA attempts to catch the main temporal dynamics of functional outputs by linear combinations of the discretised values. Time shifts are non linear relationships involving time and an input variable. Fukunaga and Olsen (1971) illustrated this issue by considering a model whose output is a bump of fixed shape (they use a Gaussian bell curve) centred at variable time instants. In that case the principal directions spans the same vector space as the collection of bumps centred at each time step. Hence, the exact linear dimension grows with refinement of the time resolution of the discretisation. Non linear dimension reduction techniques are required to handle such situations.

## 4 Auto-associative models for non linear dimension reduction

The auto-associative model (AAM) proposed by Girard and Iovleff (2008)<sup>3</sup> approximates point clouds by implic-

<sup>3</sup>Stéphane Girard, not Sylvain.



**Figure 2.** Reconstruction performance of PCA with a single principal component when only the coefficient of restitution varies. Top: comparison between original and reconstructed trajectories producing the worst RMSE. Middle: absolute reconstruction errors. Bottom: relative reconstruction errors. Intervals where the height was below 0.1 m were discarded. Lines are set to 0.1 opacity; darker tint indicate superposition of a large number of lines.

itly defined manifolds, instead of cubes like PCA does. It handles non linearity and can generally achieve reduction to dimension equal or close to the intrinsic dimension while preserving the fidelity of the reconstruction.

The algorithm for building AAM has 4 steps that are repeated until reconstruction is good enough:

1. *Direction computation* – A direction is computed by maximising an *index*, namely a function of the coordinates of the projection of the data points onto that direction. We used the index suggested by Girard and Iovleff (2008) that best preserves nearest neighbours.
2. *Projection* – The point cloud is projected onto the computed direction. The resulting coordinates will be called scores, by analogy with PCA terminology.
3. *Regression function estimation* – The regression function linking scores to the data points is estimated, here by spline regression.
4. *Update* – The point cloud from the current iteration is replaced by the *residuals*, namely the difference between data points and the output of the regression function estimated in step 3.

The algorithm terminates when the residuals are small enough. The final dimension is equal to the number of iterations.

PCA is a special case of auto-associative models where the regression functions are postulated to be linear. Its index is the variance of the projection of the point cloud. Its maximisation is equivalent to minimising the mean squared error between projections and data points. In that respect, it is a *global* index, contrary to the index we used for AAM based on nearest neighbour preservation, a local property.

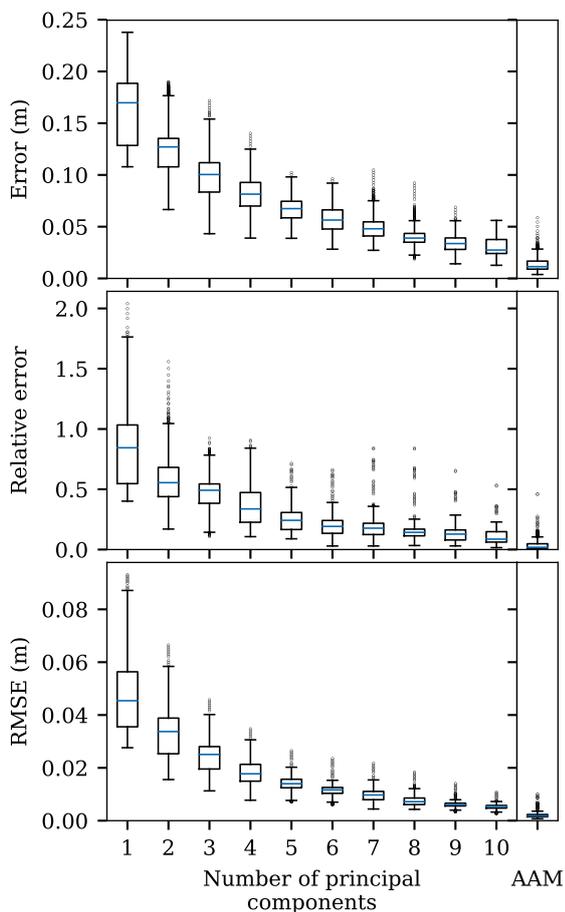
#### 4.1 AAM of the bouncing ball model

We fitted an AAM of dimension 1 on the same training set of 128 trajectories as before. We used a basis of 28 splines for the regression estimation. The number of splines was tuned manually, but this could be automatised for instance using cross validation.

Figure 4 illustrates the very good performance of the method. The worst reconstruction on the same test set as before is almost a perfect match, except for a tiny time delay and a blunting of the cusp at the last bounces. More than 90 % of the reconstructions have relative error below 10 % throughout the simulation, and more than 99 % of them have a maximal absolute error below 0.037 m.

Figure 3 shows that AAM performs better than PCA even if we keep a large number of principal components. In particular, the maximum absolute error of AAM is significantly smaller to that of PCA with 10 components.

Even better results were obtained in another similar experiment where the initial height, instead of the coefficient



**Figure 3.** Distributions of time maximum absolute error, relative error and RMSE between test trajectories and reconstructions with increasing number of principal components (left), and reconstructions by a one dimensional AAM (right) when only the coefficient of restitution varies. Blue lines indicate the medians. Boxes span the interval between first,  $Q_1$ , and third,  $Q_3$ , quartiles. Whiskers reach the last data point above (resp. below)  $Q_1 - 1.5 \times (Q_3 - Q_1)$  (resp.  $Q_3 + 1.5 \times (Q_3 - Q_1)$ ). Dots are points outside the reach of the whiskers.

of restitution, varied (results not shown). In a third experiment, we simulated 512 training trajectories with both the coefficient of restitution and initial height varying, respectively between 0.7 to 0.9 and between 0.9 m to 1.1 m. Figure 5 shows the first 6 trajectories of this training set, whose size was augmented to  $4096 - 512 = 3584$  trajectories. The effect of the two input variables combine: the higher the starting height, the higher the velocity at the first bounce. This is exemplified by the 5th trajectory (violet line) resulting from both high coefficient of restitution (0.875) and initial height (1.075 m): its second bounce is substantially away from the group of other trajectories (compare Figure 1). From visual inspection of the trajectories in Figure 1 and 5, we infer that the intrinsic dimension of the 2 input model is most likely equal to 2 because the two inputs have different effects on the output.

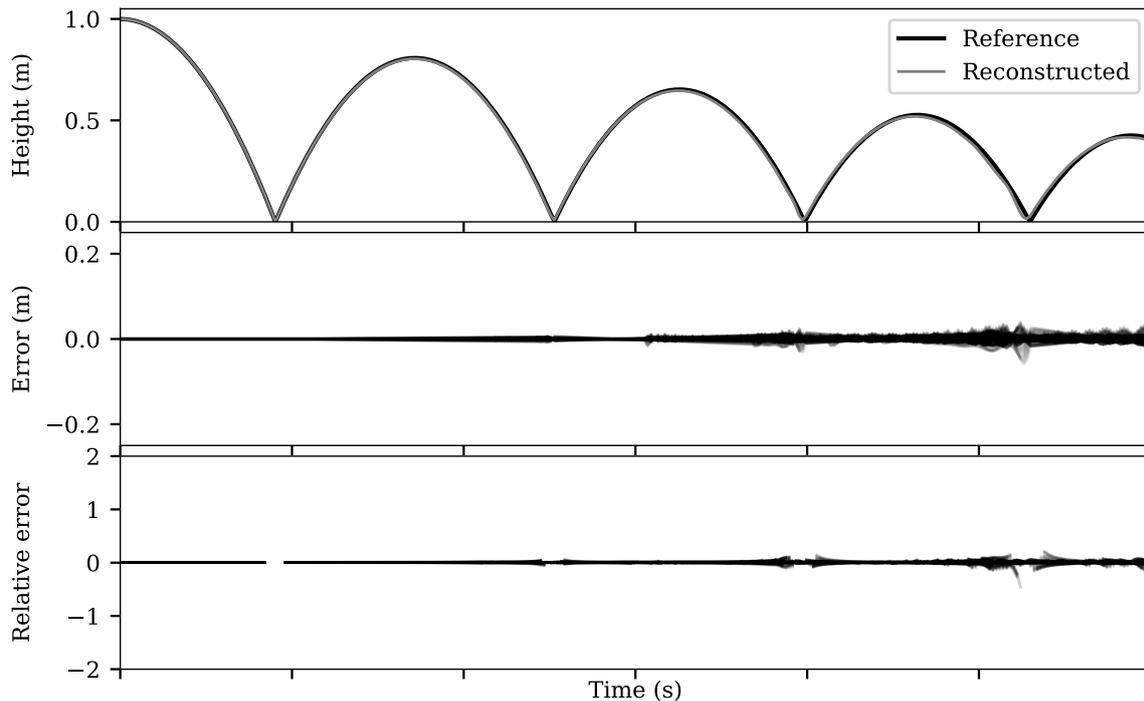
Figure 6 compares the performance of PCA with increasing number of principal components with that of AAM of dimension 1 and 2. Errors in reconstruction by PCA are globally much higher than in the single input experiment. Their distributions are leptokurtic (more “peaked”) and positively skewed: there are a lot of important errors far away from the median and spanning a large interval. AAM performs not as good as in the single input experiment but is still much better PCA with 2 components, and roughly equivalent to PCA with 5 components.

## 4.2 Sensitivity analysis in AAM projection space

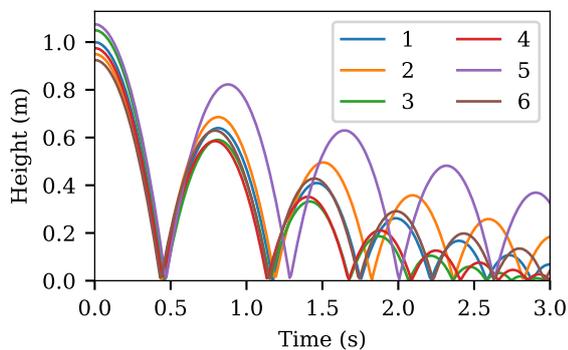
The gain in performance between the dimension 1 and 2 AAM, visible in Figure 6 (right panel), supports our guess that the intrinsic dimension of the model is 2. We confirmed that fact by analysing the sensitivity of the AAM scores to the coefficient of restitution and initial height. We computed first order and total Sobol’ indices with the Monte Carlo algorithm proposed by Sobol’ (2001) along with the “Jansen 1999” and “Saltelli 2010” estimators advocated by Saltelli et al. (2010).

The first AAM score is almost exclusively dependent on the coefficient of restitution (first order index: 94.2 %), with negligible interaction (second order joint index: 0.7 %). The second AAM score is dominated by the initial height (first order index: 78.5 %), with substantial contribution of the coefficient of restitution (first order index: 9.2 %), and interaction between the two (second order joint index: 12.2 %).

In order to interpret the physical meaning of these results, we reconstructed trajectories corresponding to locations evenly distributed along lines in the AAM projection plan. These “cross-sections” of the AAM plan space are shown in Figure 7. They illustrate what it means to have, say, “an average AAM first score and an high AAM second score”. The first score mostly controls the bouncing instants. As a matter of fact, the middle plot of Figure 7 is pretty similar to Figure 1 showing the effect of the coefficient of restitution alone, which is coherent with the result of the sensitivity analysis stated above. The sec-



**Figure 4.** Reconstruction performance of a dimension 1 AAM when only the coefficient of restitution varies. Same graphical convention as in figure 2.



**Figure 5.** The first 6 trajectories of the training set with both the coefficient of restitution and initial height varying.

ond score affects the height of the peaks while keeping bouncing instants constant. It is similar to the effect of varying initial height alone (not shown), except that the latter alters bouncing instants. AAM actually automatically decomposed the influence of the input into a “time delay and damping” component, and a “height only” component. This level of legibility cannot be achieved with PCA whenever the linearity hypothesis does not hold.

It should be noted that the procedure detailed above is fully automatic. We treated the model as a black box, and did not take advantage of any insight about its physical or mathematical properties. This is particularly alluring as it forebodes routine usage by non specialists, and possible inclusion into graphical Modelica tools.

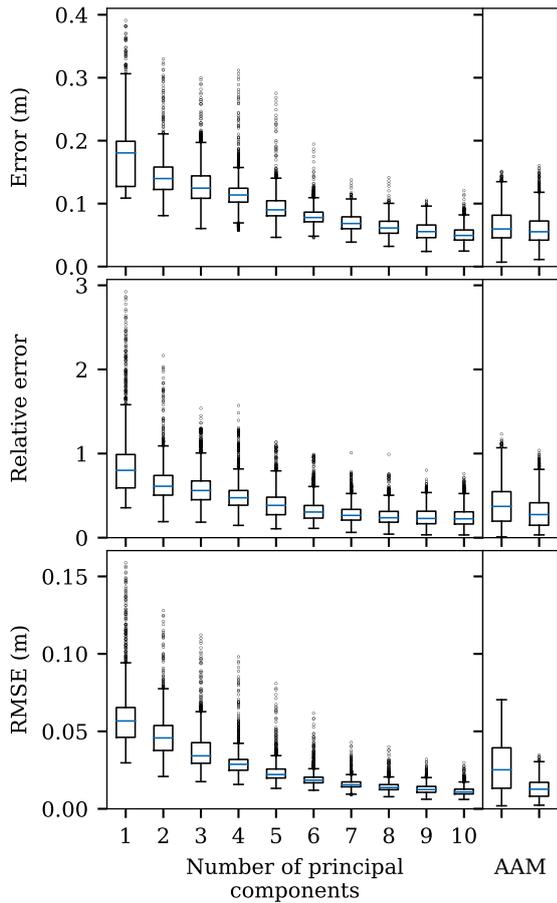
## 5 Conclusion and perspectives

Recent enrichments of the Modelica technological ecosystem enable straightforward implementation of advanced computer experiments with Modelica models. There remain however a major hurdle to overcome, namely adapting the panoply of mature mathematical methods to dynamic models with functional outputs. We showed on an example that linear dimension reduction with PCA may fall short of this objective, even for rather simple models. The recently developed non linear approach of AAM seems a very promising candidate to supplement, or even replace it altogether. It achieved very satisfying results on the presented case study and other more realistic ones not shown here. It is only little more complicated from the theoretical viewpoint, and almost as easy to use as PCA. “Degrees of freedom” in the algorithm are kept at a minimum, thus avoiding the need for elusive tuning skills.

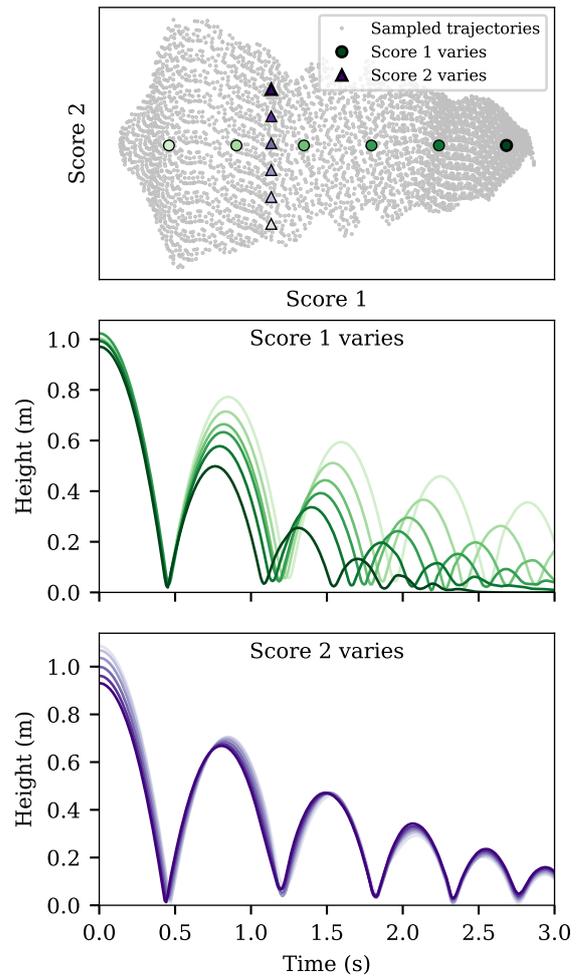
Our implementation of the regression estimation is rather elementary. Hence, there is room for further performance enhancement. On the theoretical side, the question of how to define relevant metrics in the space of AAM scores is of great interest for sensitivity analysis or supervised importance sampling.

## Acknowledgement

This work was partially supported by the Paris region through the FUI research project “Modeliscale”, a collaboration with Dassault Systèmes, Inria, EDF, Engie, CEA INES, DPS, Eurobios and Phimeca Engineering.



**Figure 6.** Distributions of time maximum absolute error, relative error and RMSE between test trajectories and reconstructions with increasing number of principal components (left), and reconstructions by a 1 and 2 dimensional AAM (right) with both the coefficient of restitution and initial height varying. Same graphical convention as in Figure 3.



**Figure 7.** One dimensional cross-sections of the AAM projection space. Top: grey dots locate train and test trajectories in the AAM projection space; circles (resp. triangles) are located on arbitrary lines to illustrate the effect of varying the 1st (resp. 2nd) projection score. Middle (resp. bottom): reconstructed trajectories corresponding to the circle (resp. triangles) of same tint in the top plot.

## References

- Katherine Campbell, Michael D. McKay, and Brian J. Williams. Sensitivity analysis when model outputs are functions. *Reliability Engineering & System Safety*, 91(10-11):1468–1472, 2006.
- Keinosuke Fukunaga and David R. Olsen. An algorithm for finding intrinsic dimensionality of data. *IEEE Transactions on Computers*, C-20(2):176–183, feb 1971. doi:10.1109/t-c.1971.223208. URL <https://doi.org/10.1109/t-c.1971.223208>.
- Stéphane Girard and Serge Iovleff. Auto-associative models, nonlinear principal component analysis, manifolds and projection pursuit. In *Lecture Notes in Computational Science and Engineering*, pages 202–218. Springer Berlin Heidelberg, 2008. doi:10.1007/978-3-540-73750-6\_8. URL [https://doi.org/10.1007/978-3-540-73750-6\\_8](https://doi.org/10.1007/978-3-540-73750-6_8).
- Sylvain Girard. *Physical and Statistical Models for Steam Generator Clogging Diagnosis*. Springer International Publishing, 2014. doi:10.1007/978-3-319-09321-5. URL <https://doi.org/10.1007/978-3-319-09321-5>.
- Sylvain Girard. Otfmi: simulate FMU from OpenTURNS: User documentation. Technical Report RT-PMFRE-00997-003, Phimeca, 2017.
- Sylvain Girard and Thierry Yalamas. A probabilistic take on system modeling with modelica and python. URL <https://tinyurl.com/proba-system-model>.
- Sylvain Girard, Thierry Yalamas, and Michael Baudin. Statistical learning and 0D/1D modelling: application to battery ageing. In *Lambda Mu 21 proceedings*. Institut de maîtrise des risques (IMdR), 2018.
- Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.
- Michael E Houle. Inlierness, outlierness, hubness and discriminability: an extreme-value-theoretic foundation. *Technical Report NII-2015-002E*, 2015.
- Ian T. Jolliffe and Jorge Cadima. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202, mar 2016. doi:10.1098/rsta.2015.0202. URL <https://doi.org/10.1098/rsta.2015.0202>.
- Olivier Roustant, David Ginsbourger, and Yves Deville. DiceKriging, DiceOptim: two R packages for the analysis of computer experiments by kriging-based metamodeling and optimization. *Journal of Statistical Software*, 51(1):1–55, 2012.
- Andrea Saltelli, Marco Ratto, Terry Andres, Francesca Campolongo, Jessica Cariboni, Debora Gatelli, Michaela Saisana, and Stefano Tarantola. *Global sensitivity analysis: the primer*. Wiley Online Library, 2008.
- Andrea Saltelli, Paola Annoni, Ivano Azzini, Francesca Campolongo, Marco Ratto, and Stefano Tarantola. Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index. *Computer Physics Communications*, 181(2):259–270, 2010.
- Il’ya Meerovich Sobol’. On the systematic search in a hypercube. *SIAM Journal on Numerical Analysis*, 16(5):790–793, 1979. doi:10.1137/0716058. URL <http://dx.doi.org/10.1137/0716058>.
- Il’ya Meerovich Sobol’. Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. *Mathematics and Computers in Simulation*, 55:271–280, 2001.
- Michael M. Tiller. *Modelica by Example*. Xogeny, 2015. URL <http://book.xogeny.com/>.

