



Overcoming the Obstacle of Time-dependent Model Output for Statistical Analysis by Nonlinear Methods

Girard Sylvain ^a, Gerrer Claire-Eleuthèriane ^{a*}

^a *Phimeca Engineering, 18 bvd de Reuilly, 75012 Paris, France.*

Received 08 September 2020; Revised 15 November 2020; Accepted 21 November 2020; Published 01 March 2021

Abstract

Modelica models represent static or dynamic systems. Their outputs can be scalar (numbers) or time-dependent (time series). The most advanced mathematical methods for the analysis of numerical models cannot cope with functional outputs. This paper aims to show an efficient method to reduce a time-dependent output to a few numbers. 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 adapt and implement an existing method called the *auto-associative model*, invented by Stéphane Girard, to overcome this shortcoming. The auto-associative model generalizes PCA as it projects the data on a nonlinear (instead of linear) basis. It also provides physically interpretable data representations. The difference in efficiency between both methods is illustrated in a case study of the well-known bouncing ball model. We perform output reduction and reconstruction using both methods to compare the completeness of information kept throughout the dimension reduction process.

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 [1] with advanced mathematical methods. Sensitivity analysis, model emulation, Bayesian inference and the like can now be performed routinely using scripting languages such as Python [2]. A substantial hurdle remains: 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 century-old statistical learning method [3] has been applied in virtually all fields where data is available. It is easy to implement, 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 in 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-dimensional representations produced by AAM can be leveraged to get insights about the modelled physical phenomena.

1.1. 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 often dynamic, namely their outputs are functions of time. Discretised time

* Corresponding author: gerrer@phimeca.com

<http://dx.doi.org/10.28991/HIJ-2021-02-01-01>

➤ This is an open access article under the CC-BY license (<https://creativecommons.org/licenses/by/4.0/>).

© Authors retain all copyrights.

functions are high dimensional vectors which considerably obstructs the analysis. First, it is subjected to the “curse of dimensionality” [4], 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 lose their discriminating power, especially when the component variables are correlated, which is especially 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.

The great majority of mathematical methods involved in computer experiments apply to models with scalar outputs. For instance, sensitivity analysis [5] 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” [6] 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 [7].

A common expedient to enable analysing functional outputs is to project them on a function basis [8]. When there is no obvious candidate, principal component analysis allows to automatically build an adapted basis.

2. Linear Dimension Reduction with Principal Component Analysis

The geometric approach to PCA provides the most intuitive understanding of the method. The discretisation in d steps of N realisations of a functional model output can be seen as a point cloud of N points in R . 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 [8]. 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.

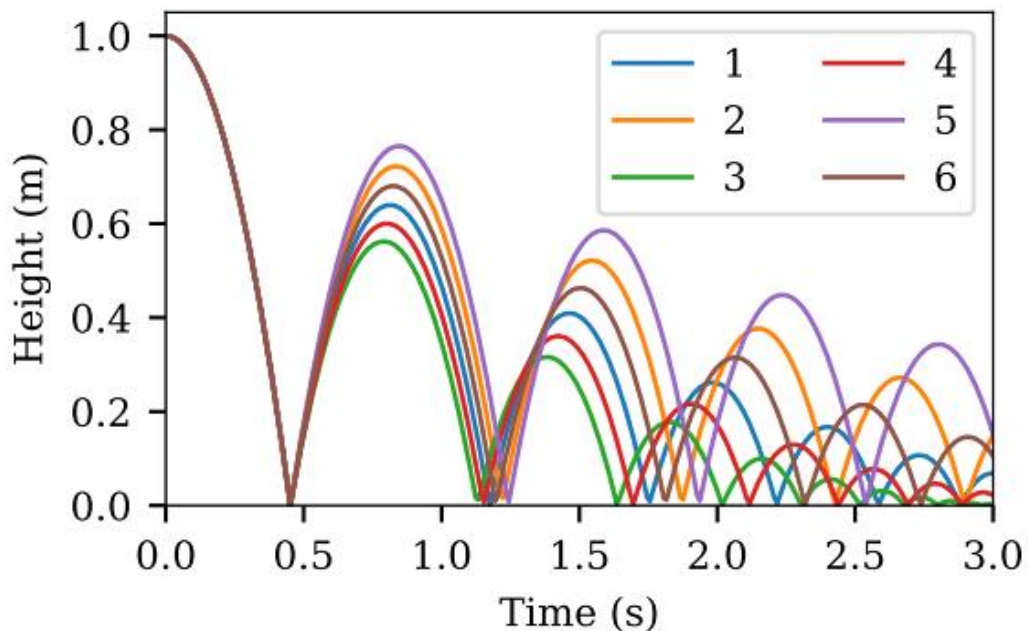


Figure 1. The first 6 trajectories of the training set when only the coefficient of restitution varies

The principal directions of the data set are sequentially built, to be mutually orthogonal. The set of the principal directions form a new basis in the space 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.

2.1. 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): 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 [18].

The trajectories were simulated with coefficient of restitution sampled between 0.7 and 0.9. following a Sobol sequence [9] to avoid redundancies. We used an FMU generated with OpenModelica, and the OtFMI Python module [10] 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 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 standard and relative reconstruction errors are outsize. 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 error globally increase as time goes by, despite the lessening of average height.

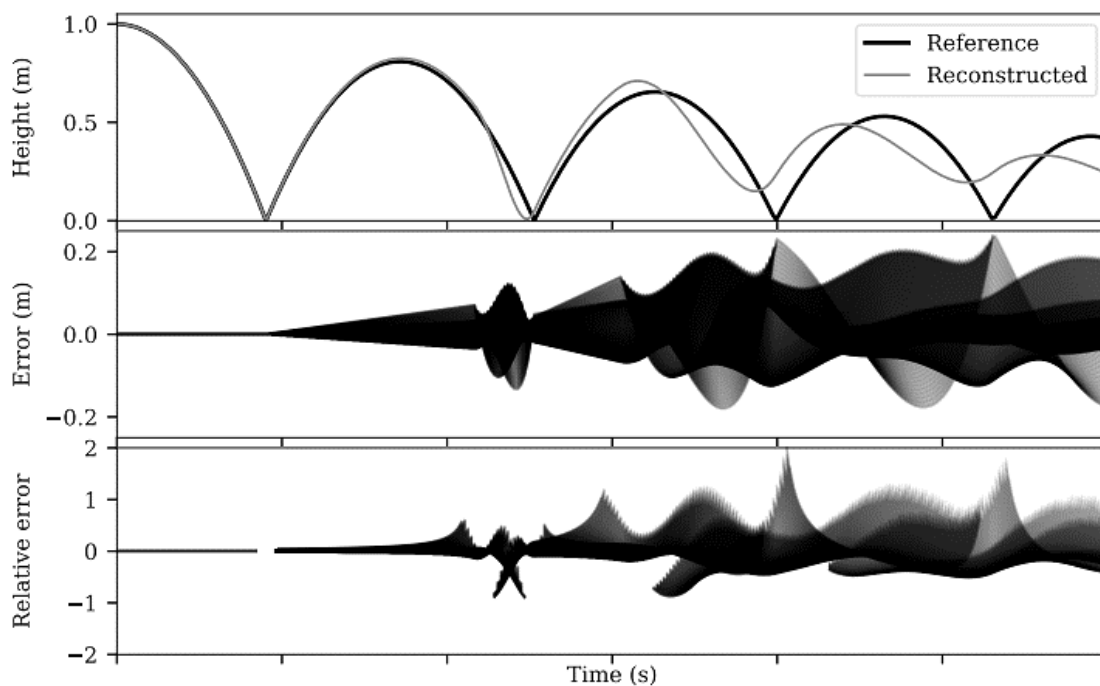


Figure 2. Reconstruction performance of PCA with a single principal component when only the coefficient of restitution varies. Top: comparison between original and reconstructed trajectories producing the worst RMSE. Middle: absolute reconstruction errors. Bottom: relative reconstruction errors. Intervals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

2.2. Time Delays: A Major Stumbling Block for PCA

What happens here is that the point cloud of trajectories has a *linear* dimension much greater than 1. It is a one-dimensional manifold extending in multiple directions in 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 %.

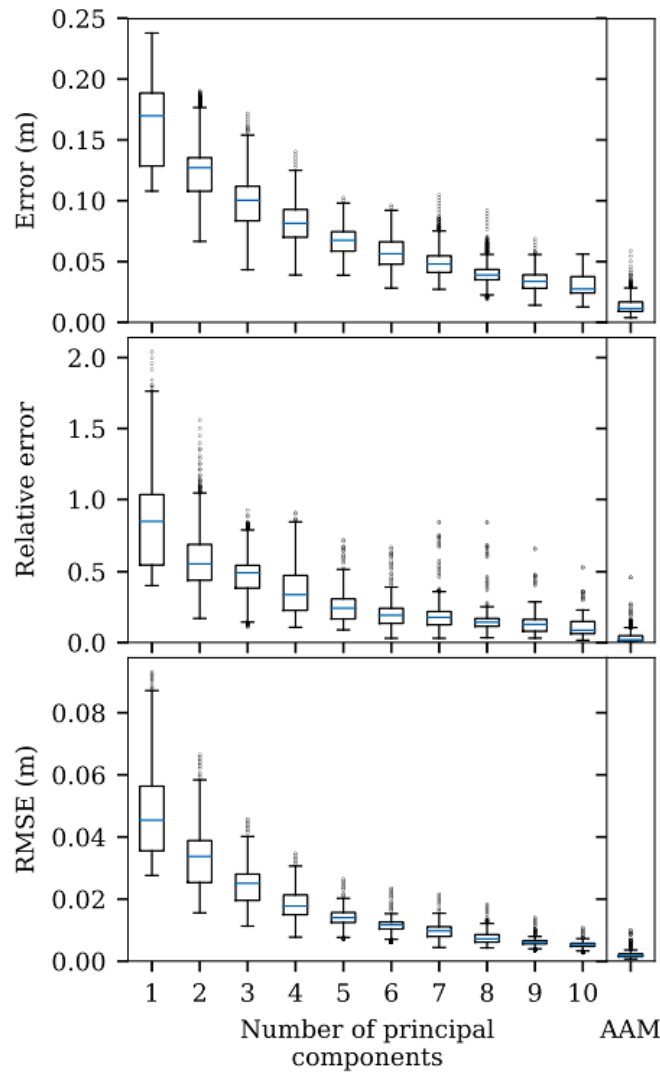


Figure 3. Distributions of time maximum absolute error, relative error and RMSE between test trajectories and reconstructions with increasing number of principal components (left), and reconstructions by a one dimensional AAM (right) when only the coefficient of restitution varies. Blue lines indicate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

PCA attempts to catch the main temporal dynamics of functional outputs by linear combinations of the discretised values. Time shifts are nonlinear relationships involving time and an input variable. Fukunaga and Olsen (1971) [11] 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 span 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. Nonlinear dimension reduction techniques are required to handle such situations [12].

3. Auto-associative Models for Nonlinear Dimension Reduction

The auto-associative model (AAM) proposed by Girard and Iovleff (2008) [13] approximates point clouds by implicitly defined manifolds, instead of cubes like PCA does. It handles nonlinearity 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:

- *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) [13] that best preserves nearest neighbours.
- *Projection* – The point cloud is projected onto the computed direction. The resulting coordinates will be called scores, by analogy with PCA terminology.

- *Regression function estimation* – The regression function linking scores to the data points is estimated, here by spline regression.
- *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.

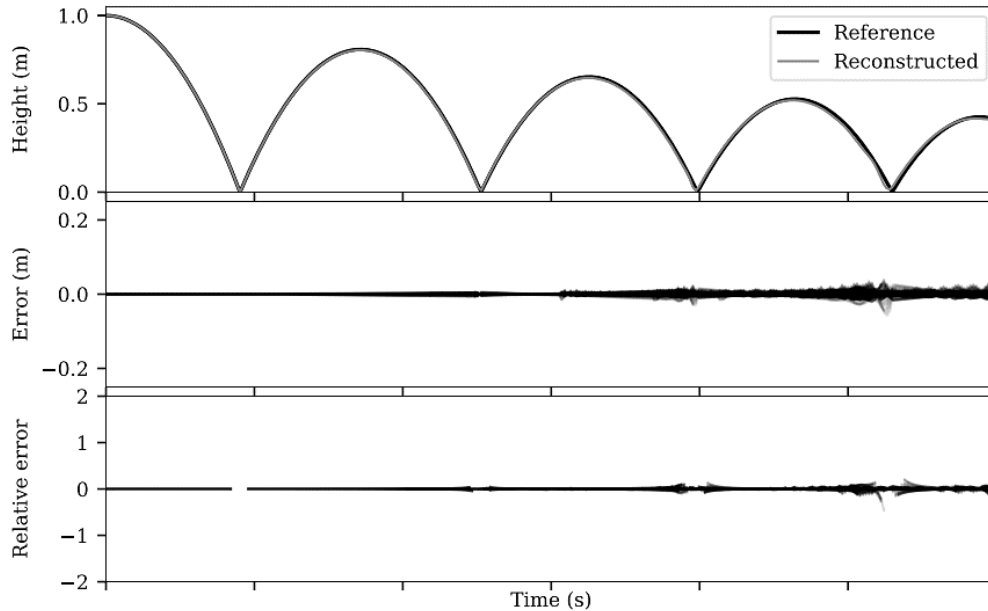


Figure 4. Reconstruction performance of a dimension 1 AAM when only the coefficient of restitution varies; Same graphical convention as in Figure 2

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.

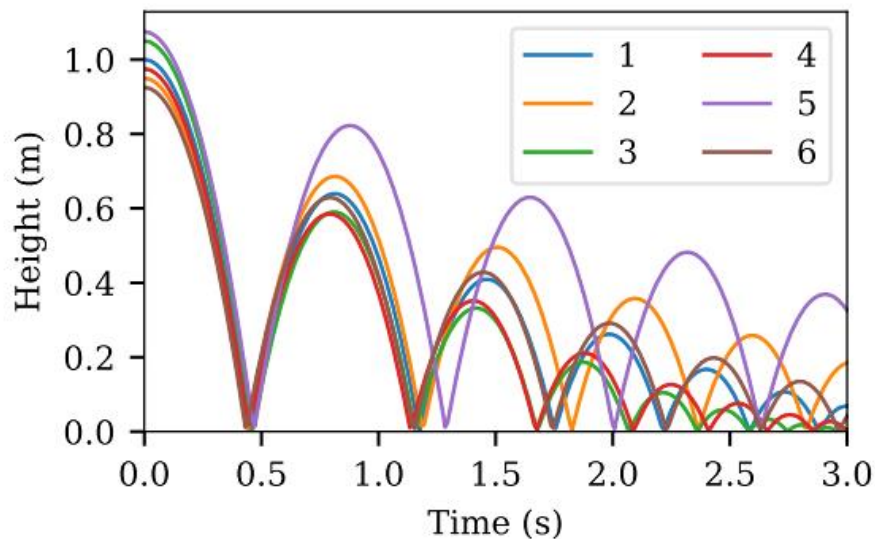


Figure 4. The first 6 trajectories of the training set with both the coefficient of restitution and initial height varying

4. 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 automatized for instance using cross validation.

Figure 4 illustrates the very good performance of the method. The worst reconstruction on the same test set 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 many 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 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, between 0.7 to 0.9 and between 0.9 to 1.1 m respectively.

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 Figures 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.

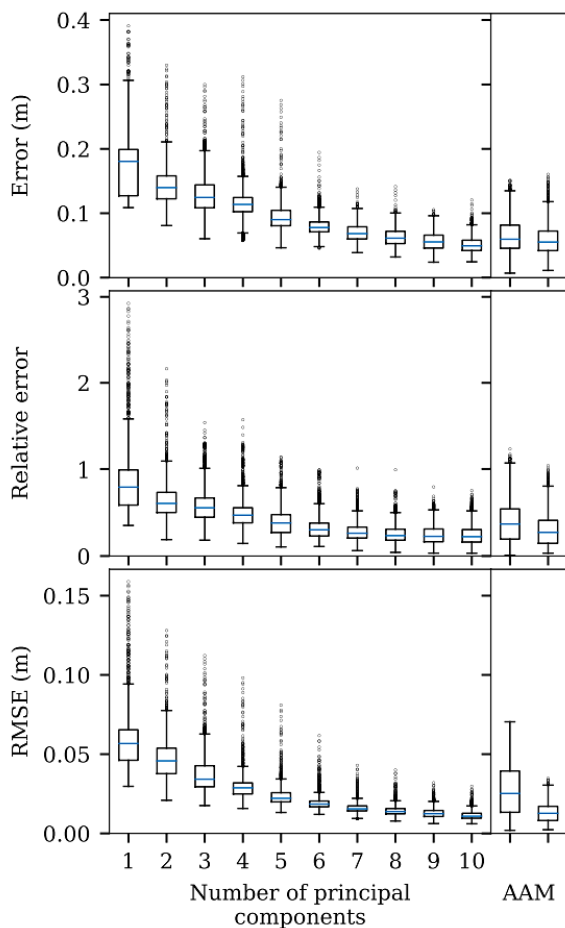


Figure 5. Distributions of time maximum absolute error, relative error and RMSE between test trajectories and reconstructions with increasing number of principal components (left), and reconstructions by a 1 and 2 dimensional AAM (right) with both the coefficient of restitution and initial height varying. Same graphical convention as in Figure 3.

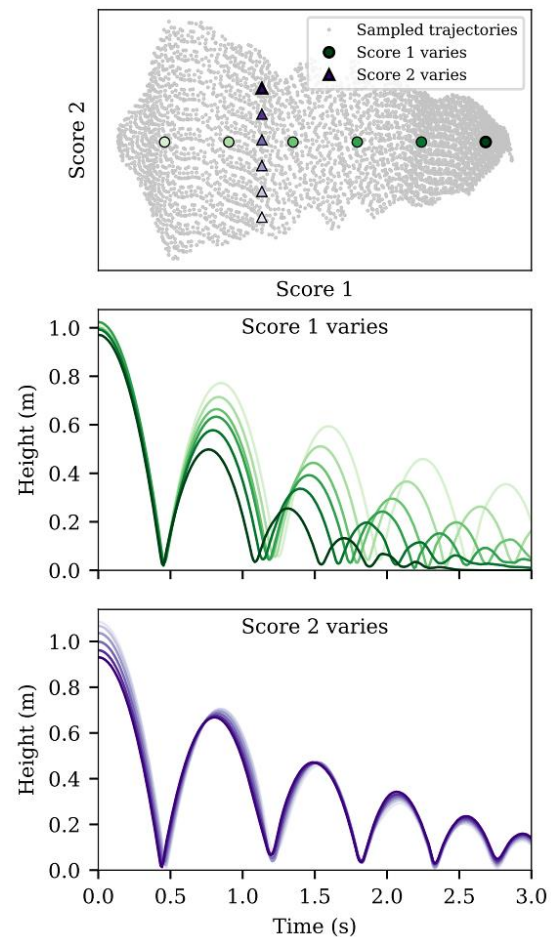


Figure 6. One dimensional cross-sections of the AAM projection space. Top: grey dots locate train and test trajectories in the AAM projection space; circles (resp. triangles) are located on arbitrary lines to illustrate the effect of varying the 1st (resp. 2nd) projection score. Middle (resp. bottom): reconstructed trajectories corresponding to the circle (resp. triangles) of same tint in the top plot.

5. 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) [14] along with the Jansen (1999) and Saltelli et al. (2010) estimators advocated by Saltelli et al. (2010) [19, 20].

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 a 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 second 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 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.

6. Conclusion and Perspectives

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

7. Funding and Acknowledgements

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.

8. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

9. References

- [1] Fritzson, P. (2020). Modelica: Equation-Based, Object-Oriented Modelling of Physical Systems. Foundations of Multi-Paradigm Modelling for Cyber-Physical Systems, 45–96. doi:10.1007/978-3-030-43946-0_3.
- [2] Girard, S., & Yalamos, T. (2017). A probabilistic take on system modeling with modelica and python. Available online: <https://tinyurl.com/proba-system-model>. (accessed on January 2021).
- [3] Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. Journal of Educational Psychology, 24(6), 417–441. doi:10.1037/h0071325.
- [4] Houle, M. E. (2015). Inlierness, outlierness, hubness and discriminability: an extreme-value-theoretic foundation. National Institute of Informatics Technical Report NII-2015-002E, Tokyo, Japan.

- [5] Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., ... Tarantola, S. (2007). *Global Sensitivity Analysis. The Primer*, Wiley Online Library. doi:10.1002/9780470725184.
- [6] Girard, S. (2014). *Physical and Statistical Models for Steam Generator Clogging Diagnosis*. Springer Briefs in Applied Sciences and Technology. doi:10.1007/978-3-319-09321-5.
- [7] Roustant, O., Ginsbourger, D., & Deville, Y. (2012). DiceKriging, DiceOptim: TwoR Packages for the Analysis of Computer Experiments by Kriging-Based Metamodeling and Optimization. *Journal of Statistical Software*, 51(1). doi:10.18637/jss.v051.i01.
- [8] Espadoto, M., Martins, R. M., Kerren, A., Hirata, N. S. T., & Telea, A. C. (2019). Towards a Quantitative Survey of Dimension Reduction Techniques. *IEEE Transactions on Visualization and Computer Graphics*, 1–1. doi:10.1109/tvcg.2019.2944182.
- [9] Sobol', I. M. (1979). On the Systematic Search in a Hypercube. *SIAM Journal on Numerical Analysis*, 16(5), 790–793. doi:10.1137/0716058.
- [10] Girard, S. (2017). *Otfmi: simulate FMU from OpenTURNS: User documentation*. Technical Report RT-PMFRE-00997-003, Phimeca. Available online: <https://github.com/openturns/otfmi>. (accessed on January 2021).
- [11] Fukunaga, K., & Olsen, D. R. (1971). An Algorithm for Finding Intrinsic Dimensionality of Data. *IEEE Transactions on Computers*, C-20(2), 176–183. doi:10.1109/t-c.1971.223208.
- [12] Gerrer, C.-E., & Girard, S. (2019). Non Linear Dimension Reduction of Dynamic Model Output. *Proceedings of the 13th International Modelica Conference, Regensburg, Germany, March 4–6, 2019*. doi:10.3384/ecp19157189.
- [13] Girard, S., & Iovleff, S. (2008). Auto-Associative Models, Nonlinear Principal Component Analysis, Manifolds and Projection Pursuit. *Principal Manifolds for Data Visualization and Dimension Reduction*, 202–218. doi:10.1007/978-3-540-73750-6_8.
- [14] Sobol', I. M. (2001). Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. *Mathematics and Computers in Simulation*, 55(1-3), 271 – 280. doi:10.1016/s0378-4754(00)00270-6.
- [15] Campbell, K., McKay, M. D., & Williams, B. J. (2006). Sensitivity analysis when model outputs are functions. *Reliability Engineering & System Safety*, 91(10-11), 1468–1472. doi:10.1016/j.res.2005.11.049.
- [16] Girard, S., Yalamas, T., & Baudin, M. (2018, October). Apprentissage statistique et modélisation 0D/1D des systèmes: application au vieillissement des batteries. Available online: <https://hal.archives-ouvertes.fr/hal-02064908/> (accessed on January 2021).
- [17] Jolliffe, I. T., & Cadima, J. (2016). Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065), 20150202. doi:10.1098/rsta.2015.0202.
- [18] M. M. Tiller, (2015) *Modelica by Example*, Xogeny. Available online: <http://book.xogeny.com/behavior/discrete/bouncing>.
- [19] Jansen, M. J. W. (1999). Analysis of variance designs for model output. *Computer Physics Communications*, 117(1-2), 35–43. doi:10.1016/s0010-4655(98)00154-4.
- [20] Saltelli, A., Annoni, P., Azzini, I., Campolongo, F., Ratto, M., & Tarantola, S. (2010). Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index. *Computer Physics Communications*, 181(2), 259–270. doi:10.1016/j.cpc.2009.09.018.