Complementary Deep - Reduced Order Model

Emmanuel Menier, Michele Alessandro Bucci, Mouadh Yagoubi, Marc Schoenauer

CD-ROM

Complementary Deep - Reduced Order Model

E. Menier[1,2] M.A. Bucci [2] M. Yagoubi[1] M. Schoenauer [2]

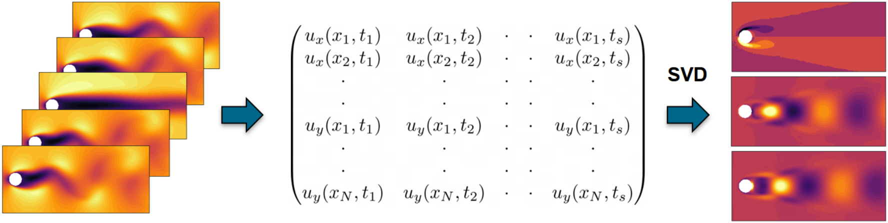
[1]IRT SystemX

[2]TAU Team, LISN / Inria

Euromech Workshop, June 2021

POD

- Proper Orthogonal Decomposition can be used to extract the principal modes (V) from a physical simulation :



- The solution of the problem can then be reduced to a linear combination of reduced number (m) of modes :

$$u(x,t) \approx V(x)\alpha(t)$$

Galerkin Projection

- An equation for the coefficients of this simplified form is obtained through Galerkin Projection of the original system :

$$\frac{\partial u(x,t)}{\partial t} = g(u(x,t)),\ V^T u = \alpha$$

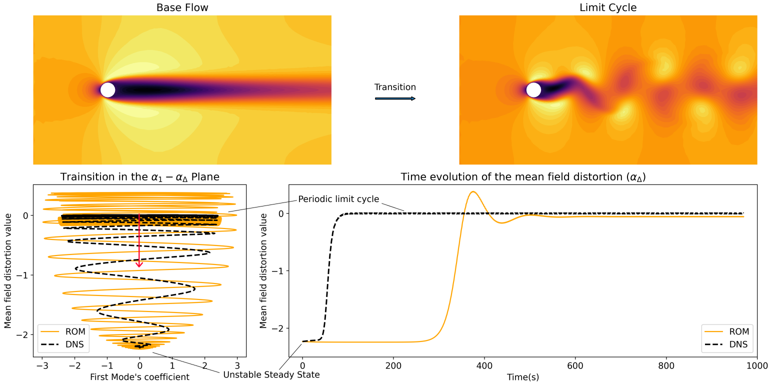
$$\implies \frac{d\alpha}{dt} = V^T g(u(x,t))$$

- The dynamics g() of the reduced model are computed from the simplified solution, which introduces error :

$$\frac{d\alpha}{dt} \approx V^T g(V\alpha) \neq V^T g(u(x,t))$$

ROM Errors

- This approximation error on the dynamics is well represented by Noack's[1] 3 equations ROM for the cylinder flow :



[1] B. R. Noack et al., "A hierarchy of low-dimensional models for the transient and post-transient cylinder wake", Journal of Fluid Mechanics **497**, 335–363 (2003).

ROM Correction

- To address classical POD-Galerkin model's shortcomings, we propose to add a correction term to their dynamics :

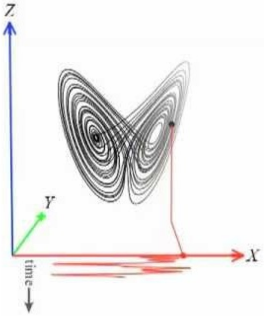
$$\frac{d\alpha}{dt} = V^T g(u) = V^T g(V\alpha) + f$$

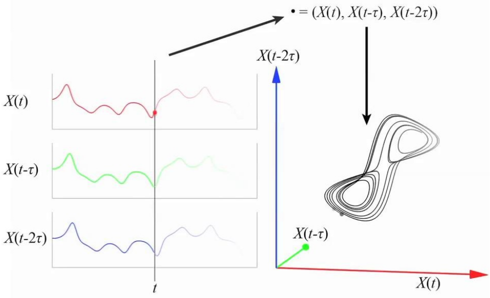
- This correction term depends on information outside of the POD basis :

$$g(u) = g(VV^T u + (I - VV^T)u) \approx g(V\alpha) + g'_{V\alpha}(I - VV^T)u$$

Taken's Theorem

- Following Taken's theorem, we can retrieve the information lost during the projection on the POD basis by considering the past states of the system :





Lorenz Visuals obtained from this link.

DDE

- Delay Differential Equations can be used to aggregate information from the past in a time continuous manner :

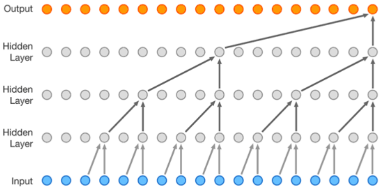
$$\frac{dx}{dt} = f(t, x, y), \;\; y(t) = \int_{-\infty}^{0} x(t + \tau) e^{\lambda\tau} d\tau$$

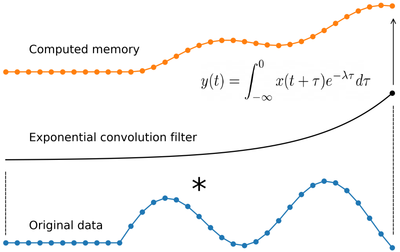
- These equations are solved as an augmented ODE system :

$$\begin{array}{c} \frac{dx}{dt} \\ \frac{dy}{dt} \end{array} = \begin{array}{c} f(t, x, y) \\ x - \lambda y \end{array}$$

Neural Correction Term

- This idea is related to other Deep Learning SOTA architectures[2]:





- A Neural Network is used to learn the ROM's correction from the memory y :

$$\begin{array}{ccccc} \frac{d\alpha}{dt} & = & g(V\alpha) & + & \mathcal{NN}(y) \\ \frac{dy}{dt} & & \alpha & - & \Lambda y \end{array}$$

[2] A. van den Oord et al., "Wavenet: A generative model for raw audio", CoRR **abs/1609.03499** (2016).

Wave Net visual obtained from this link.

Encoders

- Taking inspiration from previous work on DMD and Neural Networks[3], we propose to use encoder networks to expand the dimension of the memory :

$$\begin{array}{rcl} \frac{d\alpha}{dt} & = & g(V\alpha) \;+\; \mathcal{N}\mathcal{N}(Y) \\ \frac{dY}{dt} & & Enc(\alpha) \;-\; \Lambda Y \end{array}$$

- With this extension, the model resembles a continuous version of other modeling approaches based on Recurrent Networks[4].

[3] S. E. Otto and C. W. Rowley, "Linearly-recurrent autoencoder networks for learning dynamics", (2019).

[4] P. R. Vlachas et al., "Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks", 10.1098/rspa.2017.0844 (2018).

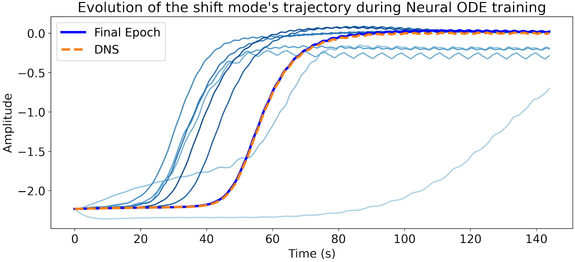
Training Data and Neural ODE

Training data for the true ROM trajectories can be obtained from the snapshot data :

$$\hat{\alpha}_{t_i} = V^T u_{DNS,t_i}$$

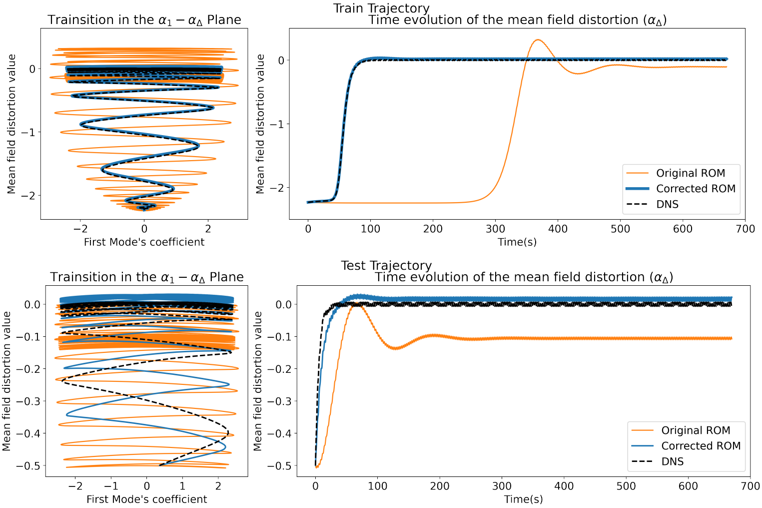
The model can then be trained through the Neural ODE[5] framework :

$$\mathcal{L}(\alpha) = \sum_{t_0}^{t_n} \|\alpha_{t_i} - \hat{\alpha}_{t_i}\|_2$$



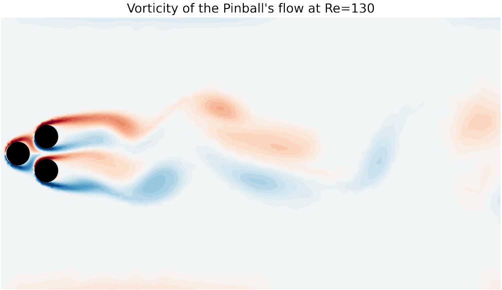
[5] R. T. Q. Chen et al., "Neural ordinary differential equations", (2019).

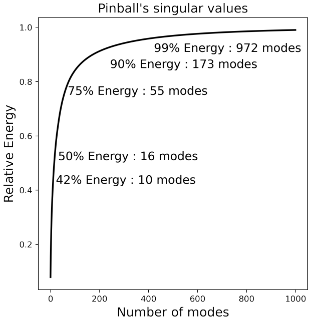
Cylinder Results



Chaotic Pinball

- We tested the approach on a more challenging case, the chaotic pinball[6]:



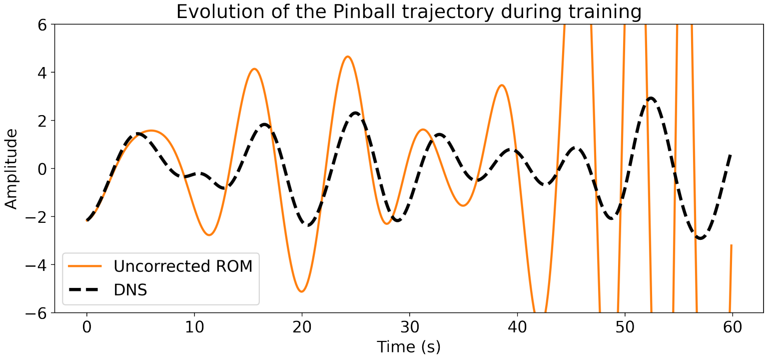


- Reminder : 8 modes are enough to capture 99 % of the cylinder flow's energy.

[6] N. Deng et al., "Low-order model for successive bifurcations of the fluidic pinball", Journal of Fluid Mechanics **884**, 10.1017/jfm.2019.959 (2019).

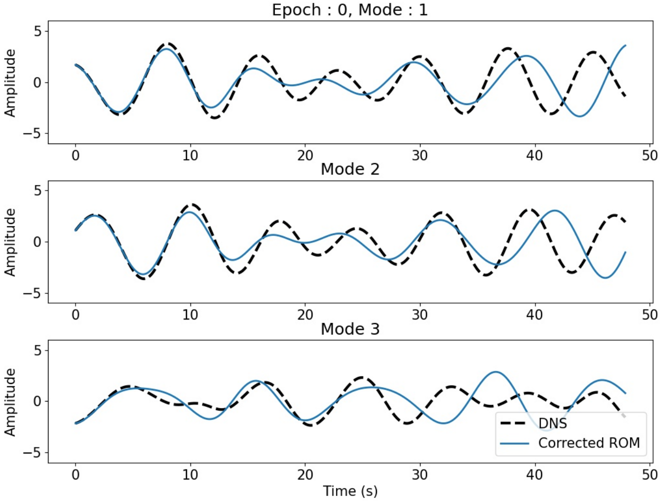
Pinball Training

- Taking a very reduced number of modes for the pinball flow yields a very unstable model.

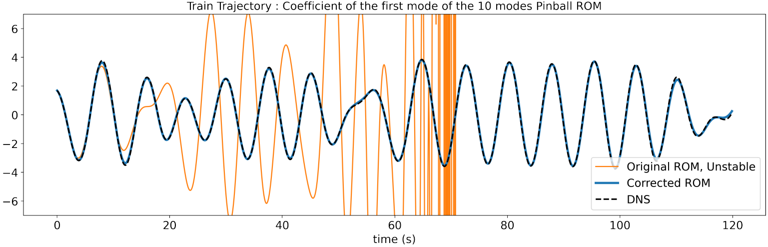


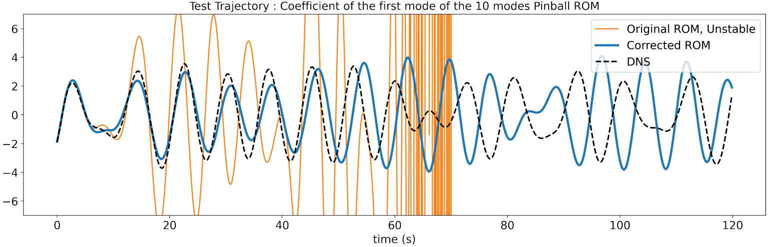
Pinball Training

- The ROM can then be corrected through our approach :



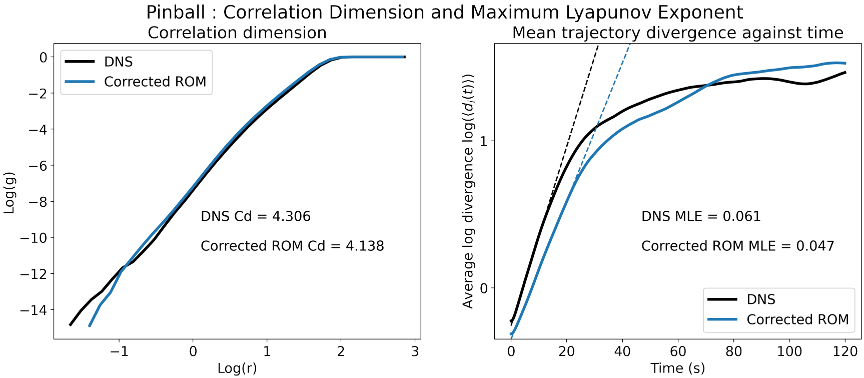
Pinball Results





Attractor Statistics

- The test trajectory presents statistics similar to the original attractor :



Summary & Future Work

Summary :

- A novel time-continuous network architecture was proposed.
- We showed that it can be used to learn an efficient correction for POD-Galerkin models.

Future Work :

- Complete work on the pinball case with more data and hyperparameter tuning.
- Derive energy constraints to give physical sense to the correction model.

References

B. R. Noack, K. Afanasiev, M. Morzyński, et al., "A hierarchy of low-dimensional models for the transient and post-transient cylinder wake", Journal of Fluid Mechanics **497**, 335–363 (2003).

A. van den Oord, S. Dieleman, H. Zen, et al., "Wavenet: A generative model for raw audio", CoRR **abs/1609.03499** (2016).

S. E. Otto and C. W. Rowley, "Linearly-recurrent autoencoder networks for learning dynamics", (2019).

P. R. Vlachas, W. Byeon, Z. Y. Wan, T. P. Sapsis, and P. Koumoutsakos, "Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks", 10.1098/rspa.2017.0844 (2018).

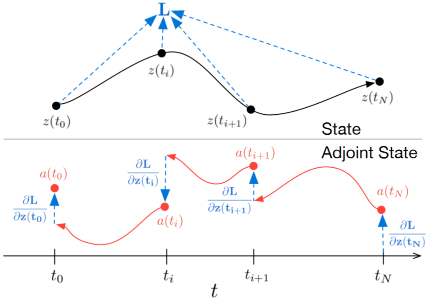
R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, "Neural ordinary differential equations", (2019).

N. Deng, B. R. Noack, M. Morzynski, et al., "Low-order model for successive bifurcations of the fluidic pinball", Journal of Fluid Mechanics **884**, 10.1017/jfm.2019.959 (2019).

Annex Slides

Neural ODE

- Neural ODE apply the adjoint backpropagation algorithm to Neural Networks.



$$\frac{dz(t)}{dt} = f(z, t; \theta)$$

$$a(t) = \frac{dL}{dz}\Big|_t$$

$$\frac{da(t)}{dt} = -a(t)\frac{\partial f(z(t),t;\theta)}{\partial z}$$

$$\frac{dL}{d\theta} = -\int_{t_0}^{t_1} a(t)\frac{\partial f(z(t),t;\theta)}{\partial \theta}$$

ROM NODE

- In our case, the Neural ODE corresponds to the complete ROM :

$$z = \begin{pmatrix} \alpha \\ Y \end{pmatrix}$$

$$f(z, t; \theta) = \begin{array}{ccc} \frac{1}{Re}\overline{K}\alpha - \alpha^T\overline{N}\alpha & + & \mathcal{N}\mathcal{N}(Y; \theta_1) \\ Enc(\alpha; \theta_2) & - & \Lambda Y \end{array}$$

- Because the model is fully expressed with differentiable objects, the required jacobians can be easily evaluated through automatic differentiation.

Cylinder Hyperparameters

ReLU activation	
Encoder Network	
Layer 1	3 *neurons*
Layer 2	11 *neurons*
Layer 3	19 *neurons*
Layer 4	27 *neurons*
Correction Network	
Layer 1	30 *neurons*
Layer 2	30 *neurons*
Layer 3	30 *neurons*
Layer 4	3 *neurons*

Pinball Hyperparameters

Swish activation	
Encoder Network	
Layer 1	10 *neurons*
Layer 2	36 *neurons*
Layer 3	63 *neurons*
Layer 4	90 *neurons*
Correction Network	
Layer 1	100 *neurons*
Layer 2	500 *neurons*
Layer 3	500 *neurons*
Layer 4	10 *neurons*

Pinball Training

- This model can then be progressively corrected and stabilised through our approach :