LEARNING INTERPRETABLE LATENT DYNAMICS FOR A 2D AIRFOIL SYSTEM

Christian Donner[1*], **Natasa Tagasovska**[1], **Guosheng He**[2], **Karen Mulleners**[2], **Hideaki Shimazaki**[3], **Guillaume Obozinski**[1]
[1]Swiss Data Science Center, EPFL & ETH Zurich,
[2]Unsteady flow diagnostics laboratory, EPFL
[3]Center for Human Nature, AI, and Neuroscience, Hokkaido University
[*]Corresponding author: christian.donner@sdsc.ethz.ch

ABSTRACT

We develop a model, that is flexible enough to capture the complex unsteady aerodynamics of a 2D airfoil undergoing dynamic stall, but simple enough to be trained end-to-end with experimental data, to provide analytically tractable dynamics, and to propagate uncertainty. Provided with a specific motion trajectory, it outputs mean and uncertainty estimates for the pressure field and forces that act on the airfoil. To this end, the model learns a low-dimensional latent space, where the system dynamics are linear but parametrised by control variables, namely the angle of attack. Learning additionally a non-linear mapping from this latent space to the observed parameters, we show that the model provides qualitatively and physically correct predictions for trajectories of lift and pressure field outside the training distribution.

1 INTRODUCTION

Describing, forecasting, and in the best case understanding effects of unsteady turbulent flows on an airfoil is a major challenge in aerodynamics (Leishman, 2002; Jones, 2020; Raiola et al., 2020). Even in simplified configurations where we concentrate on the flow development in the central cross-sectional plane of a two-dimensional airfoil and ignore the three-dimensional flow development, the problem remains challenging. Nonlinear phenomena like *stall* (McCroskey, 1981), i.e., a sudden drop in lift force caused by the flow separation at locations along the airfoil, are a topic of active research. Stall typically occurs when the airfoil's angle of attack (AoA), i.e. the angle between the relative wind and the airfoil's principal axis or chordline, increases beyond a critical AoA. The problem becomes even more intricate when the airfoil is subjected to an unsteady change in the AoA and stall is delayed to higher AoA's than under static conditions. This is the so-called *dynamic* stall (Mulleners & Raffel, 2013).

Solving the Navier-Stokes equations even for a reduced turbulent system, such as the 2D airfoil, can be quite challenging, computationally expensive, and lack in accuracy (Speziale, 1998; Gharali & Johnson, 2013; Thé & Yu, 2017; Smith et al., 2020). Hence, efforts have been made to identify only a subset of important descriptive variables for the system to model the dynamics (Lidard et al., 2021; Ghoreyshi & Cummings, 2013) or first reduce the system's dimensionality (Gomez et al., 2019), to then describe the dynamics. In contrast, recent advances in machine learning make it nowadays possible to learn different flows end-to-end and use the learnt model to generate surrogate data (see e.g. Brenner et al. (2019); Brunton et al. (2020) and references therein). However, because the proposed models are quite flexible and rich, learning them requires a lot of data, they are less robust to distributional shifts, their extrapolation properties are limited, and it is difficult to propagate uncertainty through them.

Here, we attempt to design a model that is flexible enough to capture the aerodynamics of a moving 2D airfoil, but simple enough that (a) end-to-end model training is possible with a reasonable amount of training data, (b) its dynamics are interpretable, and (c) it provides uncertainty estimates. This payoff dictates our design decisions at every stage: We refrain from modelling the full flow around the airfoil, but rather model the pressure at the boundary layer (i.e. in direct vicinity of the airfoil)

and the forces, such as the lift and pitching moment, that are immediate resultants of the surface pressure distribution around the airfoil's surface. We resort to a model placed within the state-space modelling (SSM) framework (Durbin & Koopman, 2012; Kitagawa, 1996): Aeroynamics are modelled linearly in a low-dimensional latent space, but the dynamics depend nonlinearly on control variables, such as the AOA. This ensures that the learnt dynamics are simple, such that e.g. fixed points and stability can be determined analytically, and that the model retains a certain level of flexibility. Furthermore, we consider, that the low-dimensional space can be mapped to the observations by a function, that can account for nonlinear relations, but is tractable enough to propagate uncertainties from the latent-space to observational space and vice-versa (Ghahramani & Roweis, 1999; Deisenroth et al., 2009).

In the following section 2 we introduce formally the model and briefly describe how it can be inferred from training data. Then, we demonstrate the predictive power of the model and how it can provide understanding of the underlying system in section 3, followed by some closing remarks in section 4.

2 LATENT VARIABLE MODEL FOR HIGH-DIMENSIONAL TIME-SERIES

2D airfoil data While the proposed model is applicable to any multi-dimensional discrete time-series $\boldsymbol{y}_t \in \mathbb{R}^D$, we introduce it directly with the final application in mind. At each time point t, we observe the pressure coefficients measured at certain positions on the airfoil $\mathbf{c}_p(t) = (c_{p_u}(t, x_1), \ldots, c_{p_u}(t, x_d), c_{p_l}(t, x_{d+1}), \ldots, c_{p_l}(t, x_{D_p}))^\top$, where x_i is the i^{th} x-position of measurement on the wing and $c_{p_{l\backslash u}}$ denotes whether the pressure was measured on the lower\upper side of the wing. Furthermore, we observe the coefficient of lift $c_l(t)$ and pitching moment $c_m(t)$, which are are directly related to $\mathbf{c}_p(t)$. For example the coefficient of lift is given by

$$c_l(t) = g_l(\mathbf{c}_p(t)) = \frac{1}{x_{\text{te}} - x_{\text{le}}} \int_{x_{\text{le}}}^{x_{\text{te}}} \left(c_{p_{\text{l}}}(t, x) - c_{p_{\text{u}}}(t, x)\right) \mathrm{d}x, \tag{1}$$

where x_{le}, and x_{te} are the x-positions of the leading and trailing edge of the wing, and $x_{\text{te}} - x_{\text{le}}$ is the airfoil chord length (details in section B).

At any time t this results in observations being $\boldsymbol{y}_t = (\mathbf{c}_p(t), c_l(t), c_m(t))^\top$. Furthermore, we observe certain control variables $\boldsymbol{u}_t$, that change the dynamics of the system. In our example $\boldsymbol{u}_t = (\alpha(t), \frac{d\alpha}{\mathrm{d}t}(t))^\top$, where $\alpha(t)$ is the AoA. For details on the data we refer to He et al. (2020).

Modelling observations with a low-dimensional latent space A common approach to model multivariate sequences is within the SSM framework (Durbin & Koopman, 2012). In what follows we will present a tailored SSM for representing the dynamics of a 2D airfoil, that we call *DAirSSM*.

As in many SSM models, we model observations $\boldsymbol{y}_t$ following a Gaussian distribution with covariance matrix Q_y and whose mean is determined by some latent state-variable $\boldsymbol{z}_t \in \mathbb{R}^{D_z}$. However, in contrast to the frequently assumed linearity, in our framework the mean depends nonlinearly on $\boldsymbol{z}_t$. DAirSSM models the expected pressure coefficient vector $\boldsymbol{\mu}_{c_p}(\boldsymbol{z}_t) = \mathbb{E}\left[\mathbf{c}_p(t)|\boldsymbol{z}_t\right]$ under the form

$$\boldsymbol{\mu}_{c_p}(\boldsymbol{z}_t) = W\boldsymbol{\phi}(\boldsymbol{z}_t), \tag{2}$$

where the feature vector $\boldsymbol{\phi}(\boldsymbol{z}) = (1, \boldsymbol{z}, k_1(\boldsymbol{z}), k_2(\boldsymbol{z}), \ldots)^\top \in \mathbb{R}^{D_\phi}$, and the nonlinear features $k_j(\boldsymbol{z}) = \exp\left(-\frac{\|\boldsymbol{z}-\boldsymbol{r}_j\|_2^2}{2\nu^2}\right)$ are radial basis functions (RBFs). The centres $\boldsymbol{r}_j$ are positioned on a regular grid centred at the origin in the latent space. The length scale ν is considered to be fixed. With $\boldsymbol{\phi}(\boldsymbol{z}) = (1, \boldsymbol{z})^\top$ one would recover a linear SSM, also know as the Kalman filter (Kalman, 1960). Instead of learning a full rank weight matrix W we assume a low-rank matrix $W = W_{c_p} W_\phi^\top$, with $W_{c_p} \in \mathbb{R}^{D_p \times D_w}$ and $W_\phi \in \mathbb{R}^{D_\phi \times D_w}$ and the rank $D_w < D_p < D_\phi$. By doing so, we significantly reduce the number of parameters that need to be learnt. This choice is motivated by the intuition that, for the flow regimes considered, the pressure field can be approximated by a combination of a number of modes which are less than the number of measurement positions.

Specifically for our application, we also enforce physical constraints in the model (cf. equation 1 and section B). The model's mean of the observed quantities $c_l(t)$ and $c_m(t)$ can be defined by $\mu_{c_l}(\boldsymbol{z}_t) = \mathbb{E}\left[c_l(t)|\boldsymbol{z}_t\right] = g_l(\boldsymbol{\mu}_{c_p}(\boldsymbol{z}_t))$ and $\mu_{c_m}(\boldsymbol{z}_t) = \mathbb{E}\left[c_m(t)|\boldsymbol{z}_t\right] = g_m(\boldsymbol{\mu}_{c_p}(\boldsymbol{z}_t))$, where the

expectation is with respect to the conditional distribution of $\mathbf{c}_p(t)$ given $\boldsymbol{z}_t$. This way we incorporate the two quantities without adding new model parameters.

Interpretable state dynamics As for linear state-space models we model the (discretised) time evolution of the latent state $\boldsymbol{z}$ as being linear with additional Gaussian noise. But instead of having stationary dynamics, control variables $\boldsymbol{u}_t$ might alter the system. Hence, the state equations are

$$\boldsymbol{z}_t = A(\boldsymbol{u}_t)\boldsymbol{z}_{t-1} + \boldsymbol{b}(\boldsymbol{u}_t) + \boldsymbol{\epsilon}_t, \tag{3}$$

with $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, Q_z)$. The matrix and vector valued functions A and $\boldsymbol{b}$ are modelled by a neural network with one shared hidden layer (10 units), a $\tanh$ non-linearity, and a linear readout layer. Since equation 3 is linear, we are able to analytically investigate the asymptotic behaviour of the system for fixed control variables $\boldsymbol{u}$, i.e. the fixed points and determine stability of the system.

Inference For a training dataset $\mathcal{D} = \{(\boldsymbol{y}_t, \boldsymbol{u}_t)\}_{t=1}^T$, DAirSSM , defined by the Gaussian observation model $p(\boldsymbol{y}_t|\boldsymbol{z}_t, \boldsymbol{\theta})$ and equation 3 has the joint Markov likelihood

$$p(\boldsymbol{y}_{1:T}, \boldsymbol{z}_{0:T}|\boldsymbol{u}_{1:T}, \boldsymbol{\theta}, \boldsymbol{\zeta}, \boldsymbol{\mu}_0, \Sigma_0) = p(\boldsymbol{z}_0|\boldsymbol{\mu}_0, \Sigma_0) \prod_{t=1}^{T} p(\boldsymbol{y}_t|\boldsymbol{z}_t, \boldsymbol{\theta}) p(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}, \boldsymbol{u}_t, \boldsymbol{\zeta}), \tag{4}$$

where $\boldsymbol{\mu}_0, \Sigma_0$ are the mean and covariance of the initial state. Furthermore, we denote by $\boldsymbol{\theta} = \{W_{c_p}, W_\phi, Q_y\}$, and $\boldsymbol{\zeta}$ as the parameters of the aforementioned neural network for $A(\boldsymbol{u}), \boldsymbol{b}(\boldsymbol{u})$ in addition to the latent noise covariance Q_z. All these parameters need to be learnt. For notational convenience, fixed parameters (e.g. $\boldsymbol{r}, \nu$) are omitted in equation 4.

To invoke the expectation-maximisation procedure (Dempster et al., 1977) for learning the relevant model parameters, we derive the $\mathcal{Q}$-function of the inference problem

$$\mathcal{Q} \doteq \mathbb{E}_z \left[\ln p(\boldsymbol{y}_{1:T}, \boldsymbol{z}_{0:T}|\boldsymbol{u}_{1:T}, \boldsymbol{\theta}, \boldsymbol{\zeta}, \boldsymbol{\mu}_0, \Sigma_0)\right] \leq \ln p(\boldsymbol{y}_{1:T}, |\boldsymbol{u}_{1:T}, \boldsymbol{\theta}, \boldsymbol{\zeta}, \boldsymbol{\mu}_0, \Sigma_0), \tag{5}$$

where the expectation is with respect to the posterior density of $\boldsymbol{z}_{0:T}$ given fixed model parameters. The density over the latent trajectory $\boldsymbol{z}_{0:T}$ cannot be derived exactly for DAirSSM . Hence, in the E-step we resort to an approximate procedure similar to (Deisenroth et al., 2009): Using the forward-backward algorithm, the prediction and smoothing step can be computed exactly, because the state equation 3 is linear in $\boldsymbol{z}$. However, because of the nonlinear form of equation 2 we have to resort to a Gaussian approximation for the filter density. Given that we chose linear and RBF features $\phi(\boldsymbol{z})$ for the emission term of equation 2, as in Ghahramani & Roweis (1999), the moments of the filter density can be computed analytically, and we obtain our filter estimate by Gaussian moment matching (Deisenroth et al., 2009). In the M-step we maximise equation 5 with respect to the model parameters. For Q_y, Q_z, $\boldsymbol{\mu}_0$, and Σ_0 we obtain analytical solutions, while for the rest we use numerical gradient ascent procedures. We could also solve for W_{c_p} and W_ϕ analytically, which would require to solve for each of them alternately. In practice, it turned out to be beneficial to optimise them numerically but jointly.

3 RESULTS

We train DAirSSM on 4 experiments of a NACA0015 airfoil (see section 2, and He et al. (2020) for details). In each experiment, the airfoil was sinusoidally oscillated around a mean AoA $\alpha_0 = 20°$, with an amplitude $\alpha_1 = 8°$ such that $\alpha_t = \alpha_0 + \alpha_1 \sin(2\pi f t)$ for different pitching frequencies ($f = 4.8, 3.2, 1.6,$ and 0.8 Hz). Pressure coefficients $\mathbf{c}_p$ were then measured at 36 sites on the wing (20 on upper, 16 on the lower wing) during 78 pitching cycles. The model is trained with $D_z = 2$, $D_w = 5$, and $D_\phi = 28$ (25 RBF , 2 linear, and 1 constant feature). After training, we make predictions for $\mathbf{c}_p$ and c_l corresponding to a 5^{th} experiment with $f = 2.4$ Hz (figure 1 **A**–**D**). For the model predictions, only the AoA trajectory and its time derivative are provided. DAirSSM accurately predicts the measured pressure field, as well as the time evolution of the lift coefficient (figure 1 **C** and **D**). Uncertainty is predicted to be higher for $\mathbf{c}_p$ on the upper side of the wing, where in fact larger fluctuations are observed. Early in the pitching cycle figure 1 **C** shows that the uncertainty is overestimated. This is due to the fact that observation noise Q_y is assumed to be constant, which could most likely be addressed by making Q_y dependent on the control variables $\boldsymbol{u}$. Note, that for the mean prediction to correctly predict c_l the model needs to correctly predict $\mathbf{c}_p$, since we enforce the physical relation between the two. In section A we compare the predictive

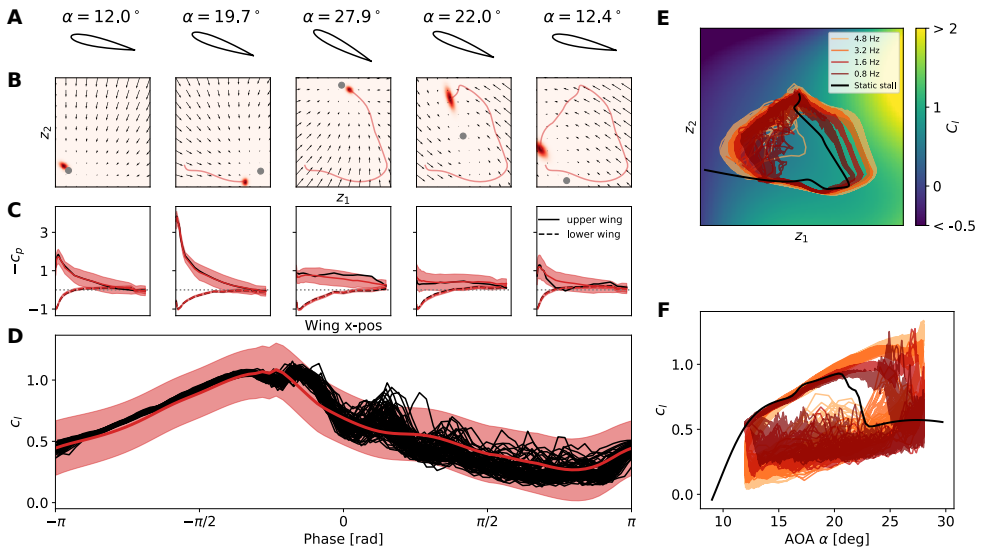

Figure 1: **Predictions of DAirSSM . A–D** depict results for a held-out experiment with pitching frequency $f = 2.4$ Hz. **A**: The airfoil with different AoAs α at different phases of a pitching cycle. **B**: The latent space of the model with the predicted density of the latent state (red colour map), the past mean trajectory (red line), and the predicted dynamics (black arrows) with the stable fixed point (grey dot). **C**: The measured (black lines) and mean predicted negative pressure (red lines) at sensor positions. Shaded areas denote the 99% confidence interval of the prediction. **D**: The coefficient of lift as a function of pitching cycle phase. Denoted are the measured trajectory for 78 cycles (black lines), and the model's mean prediction (red line), and its 99% confidence interval (red shaded area). **E–F** show that DAirSSM predicts correctly the phenomenon of static stall. **E**: The inferred mean trajectory of the training experiments in the latent space (legend indicates pitching frequencies). The black line indicates the (stable) fixed points for different stationary AoA ($\mathrm{d}\alpha/\mathrm{d}t = 0$) of the inferred dynamical system. The colourbar depicts the mapping from latent space to the lift c_l. **F**: c_l as function of AoA α. The trajectories are the training data, and the black line the mapped fixed points as in panel E.

performance of the full DAirSSM , and baselines that only incorporate the subset of the models' features. The predictive performance of the full DAirSSM is best on the held-out experiment.

Extrapolating model predictions As noted previously, large fluctuations are observed on the upper side of the wing, which is due to the aforementioned stall phenomenon, where the airflow detaches at a critical AoA resulting in a sudden drop of lift. It is known that this critical AoA decreases with decreasing pitching frequency and that it is lowest for a constant AoA ($\boldsymbol{u} = (\alpha, 0)^\top$). Even though there were no trajectories with constant AoA in the training data, DAirSSM correctly predicts a drop of c_l for smaller AoAs than in the training data (figure 1 **F**, cf. Fig. 8 in Mulleners & Raffel (2013)). Since the latent dynamics are linear (see e.g. figure 1 **B**) these predictions are analytic. Looking at the projections of the training data into the latent space (figure 1 **E**) we understand how the model is arriving at the prediction: in the latent space, DAirSSM extrapolates the training data along the direction of decreasing pitching frequencies.

4 Conclusion

We showed that DAirSSM can reliably predict and reconstruct resultants of a constant fluid flow around a 2D pitching airfoil system. The principal features of the models are (a) the control variables modifying linear latent dynamics through a neural network, and (b) the latent dynamics modifying non-linearly the observed variables via a weighted combination of linear and RBF functions. Nonetheless, the model still allowed for a training procedure that takes into account uncertainty of observations and the latent state. Using linear latent dynamics simplifies the inference, provides interpretability, and allowed us to extrapolate the model analytically to the static regime of the airfoil, which provides empirical validation of the learnt dynamics of DAirSSM given that the extrapolation captures correctly the static stall phenomenon.

ACKNOWLEDGMENTS

We thank Dr. Iordan Doytchinov and Dr. Qian Wang for feedback on the manuscript and vivid discussions. Furthermore, we acknowledge the *SenseDynamics* project to provide the framework and collaborations that made this work possible.

REFERENCES

M P Brenner, J D Eldredge, and J B Freund. Perspective on machine learning for advancing fluid mechanics. *Physical Review Fluids*, 4(10):100501, 2019. doi: 10.1103/physrevfluids.4.100501.

Steven L Brunton, Bernd R Noack, and Petros Koumoutsakos. Machine learning for fluid mechanics. *Annual Review of Fluid Mechanics*, 52:477–508, 2020.

Marc Peter Deisenroth, Marco F Huber, and Uwe D Hanebeck. Analytic moment-based gaussian process filtering. In *Proceedings of the 26th annual international conference on machine learning*, pp. 225–232, 2009.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1): 1–22, 1977.

James Durbin and Siem Jan Koopman. *Time series analysis by state space methods*. Oxford university press, 2012.

Zoubin Ghahramani and Sam T Roweis. Learning nonlinear dynamical systems using an em algorithm. 1999.

Kobra Gharali and David A Johnson. Dynamic stall simulation of a pitching airfoil under unsteady freestream velocity. *Journal of Fluids and Structures*, 42:228–244, 2013.

Mehdi Ghoreyshi and Russell M Cummings. Challenges in the aerodynamics modeling of an oscillating and translating airfoil at large incidence angles. *Aerospace Science and Technology*, 28(1): 176–190, 2013.

Daniel F Gomez, Francis Lagor, Phillip B Kirk, Andrew Lind, Anya R Jones, and Derek A Paley. Unsteady dmd-based flow field estimation from embedded pressure sensors in an actuated airfoil. In *AIAA Scitech 2019 Forum*, pp. 0346, 2019.

Guosheng He, Julien Deparday, Lars Siegel, Arne Henning, and Karen Mulleners. Stall delay and leading-edge suction for a pitching airfoil with trailing-edge flap. *AIAA Journal*, 58(12):5146–5155, 2020.

Anya R. Jones. Gust encounters of rigid wings: Taming the parameter space. *Physical Review Fluids*, 5(11):110513, 2020. doi: 10.1103/physrevfluids.5.110513.

Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. 1960.

Genshiro Kitagawa. Monte carlo filter and smoother for non-gaussian nonlinear state space models. *Journal of computational and graphical statistics*, 5(1):1–25, 1996.

J Gordon Leishman. Challenges in modelling the unsteady aerodynamics of wind turbines. *Wind Energy*, 5(2-3):85 132, 2002. ISSN 1099-1824. doi: 10.1002/we.62. URL `http://doi.wiley.com/10.1002/we.62`.

Justin M Lidard, Debdipta Goswami, David Snyder, Girguis Sedky, Anya R Jones, and Derek A Paley. Feedback control and parameter estimation for lift maximization of a pitching airfoil. *Journal of Guidance, Control, and Dynamics*, pp. 1–8, 2021.

William J McCroskey. The phenomenon of dynamic stall. Technical report, National Aeronuatics and Space Administration Moffett Field Ca Ames Research ..., 1981.

Karen Mulleners and Markus Raffel. Dynamic stall development. *Experiments in fluids*, 54(2):1–9, 2013.

Marco Raiola, Stefano Discetti, and Andrea Ianiro. Data-driven identification of unsteady-aerodynamics phenomena in flapping airfoils. *Experimental Thermal and Fluid Science*, pp. 110234, 2020.

Marilyn J Smith, Anya R. Jones, Fatma Ayancik, Karen Mulleners, and Jonathan W. Naughton. An Assessment of the State-of-the-Art from the 2019 ARO Dynamic Stall Workshop. In *AIAA Aviation forum*, virtual, 2020. doi: 10.2514/6.2020-2697.

Charles G Speziale. Turbulence modeling for time-dependent rans and vles: a review. *AIAA journal*, 36(2):173–184, 1998.

Jesse Thé and Hesheng Yu. A critical review on the simulations of wind turbine aerodynamics focusing on hybrid rans-les methods. *Energy*, 138:257–289, 2017.

A MODEL COMPARISON

model name	nonlinear $\boldsymbol{\mu}_y$	physically constrained	D_z	D_w	test log-likelihood ℓ_{test}
DAirSSM	no	yes	2	3	303910
DAirSSM	no	yes	4	5	478380
DSSM	yes	no	2	5	420910
DAirSSM	yes	yes	2	5	**578815**

Table 1: **DAirSSM Model validation**

Different models are trained on 4 pitching airfoil experiments, and the test log-likelihood is computed on one hold-out experiment (see section 3). The nonlinear DAirSSM presented in the main text is compared to linear models (first two rows) where $\boldsymbol{\phi}(\boldsymbol{z}) = (1, \boldsymbol{z})^\top$, but still with the physical constraints of DAirSSM . Note, that for these models we have the fixed relation $D_w = D_z + 1$. Furthermore, we train a nonlinear dynamical SSMs, similar to the full mode, but c_l and c_m are just treated as two additional observed variables, and the model is not physically informed. Note, that for the model predictions, the models were not provided with any observations, just with the trajectory of control variables $\boldsymbol{u}_t$ with $t = 1, \ldots, T$, i.e.

$$\ell_{\text{test}}(\boldsymbol{y}_{1:T}, \boldsymbol{u}_{1:T}) = \sum_{t=1}^{T} \ln \int p(\boldsymbol{y}_t|\boldsymbol{z}_t) p(\boldsymbol{z}_t|\boldsymbol{u}_{1:t}) \mathrm{d}\boldsymbol{z}_t. \tag{6}$$

We see, that the nonlinear DAirSSM performs best in predicting the dynamics of the hold-out experiment.

B PHYSICAL MODEL CONSTRAINTS

As c_l in equation 1 the *pitching moment coefficient* c_m is determined through the pressure field $\mathbf{c}_p$ by

$$c_m(t) = g_m(\mathbf{c}_p(t)) = \frac{1}{(x_{\text{te}} - x_{\text{le}})^2} \int_{x_{\text{le}}}^{x_{\text{te}}} \left(c_{p_{\text{u}}}(t, x) - c_{p_{\text{l}}}(t, x)\right) \times \left(x - \frac{x_{\text{te}} - x_{\text{le}}}{4}\right) \mathrm{d}x. \tag{7}$$

where $x_{\text{te}} - x_{\text{le}}$ is the airfoil chord length and the pressure coefficient c_p is defined as:

$$c_p = \frac{P - P_0}{0.5\rho \cdot U_0^2} \tag{8}$$

P is the pressure on the airfoil and P_0 is the static pressure of the incoming flow. ρ is the air density and U_0 is the incoming flow velocity relative to the airfoil.

The integrals are solved numerically with trapzoidal integration where the sensor positions are the integration points. For solving the expectations $\mu_{c_l}(\boldsymbol{z}_t), \mu_{c_m}(\boldsymbol{z}_t)$ we also integrate numerically, with the integration points at the sensor position.