The Past Noise Forecasting Demo

Dmitri Kondrashov and Mickael Chekroun, UCLA

August 29, 2017

1 A simple model for testing PNF

Largely based on Supplementary Information (SI) of Chekroun, Kondrashov and Ghil (2011) PNAS paper [1]. In this SI, we have considered the periodically and stochastically forced version (in the Itô sense) of a Holling population dynamics model [3]:

$$\begin{cases} \mathrm{d}x_1 = \{(r + \sigma \mathrm{d}W_t)x_1(\alpha + x_1)(1 - x_1) - cx_1x_2 + a\sin(2\pi ft)\}\mathrm{d}t \\ \\ \mathrm{d}x_2 = \{-m\alpha x_2 + (c - m)x_1x_2\}\mathrm{d}t. \end{cases} \tag{1.1}$$

The parameters of this model are described hereafter.

Three key features of this model are of special interest for testing the PNF method: (i) it is nonlinear and stochastic, but contrarily to the model used in the Main Text of [1], it does not belong to the class of EMR models; (ii) it is forced by white noise, thus emphasizing that the PNF can work without memory effects in the stochastic forcing; and (iii) it can be subjected to an easily reproducible battery of numerical tests. The skill in the PNF forecasts is mainly due to the model's low-frequency variability (LFV) manifested by low-frequency oscillatory mode (LFM), and to its *pathwise linear response*; see [1, Main Text] for more details.

Variables x_2 and x_1 here represent predator and prey density, respectively. The stochastic term in the system (1.1) represents a random perturbation of the parameter r by white noise. The variable x_1 is also assumed to exhibit a seasonal variation — due, for instance, to migration effects — as modeled by the presence of the deterministic, additive forcing $a\sin(2\pi ft)$.

We integrate the system (1.1) from $t = 0$ to $t = T_f = 2000$ (in dimensionless units) by using a classical stochastic Euler-Maruyama scheme with step size $\Delta t = 0.1$, in the context of Itô calculus. The values of the parameters are $\sigma = 0.3$, $m = r = 1$, $c = 1.5$, $\alpha = 0.3$, $a = 0.05$, and $f = 0.25$, while the initial state is $(x_1(0), x_2(0)) = (0.5, 0.5)$. For these parameter values, when both periodic forcing and noise are absent, the model has only one globally stable equilibrium.

When turning on the periodic forcing, with amplitude $a = 0.05$ and frequency $f = 0.25$, the system exhibits only one periodic orbit of period 4, which is globally stable. In the presence of noise ($\sigma = 0.3$), a low-frequency mode of period equal to approximately 25 units — *i.e.*, a frequency $f' = 0.04$ — becomes dominant. In [1, Fig. S4A] it was noticed that — when the noise is turned off — this mode is rapidly damped and is visible only during the transient that leads up to the unique attracting periodic orbit. The model's LFV in the case $\sigma = 0.3$ can be therefore attributed to a damped non-normal mode that is sustained by the noise.

2 PNF Setup

First, a model is integrated forward and the path of the noise (here $\mathrm{d}W_t$ in Eq. (1.1)) that drove this model over previous finite-time windows is stored. The PNF method is then applied in two steps. First, noise samples — obtained as copies of this path — are selected from past time intervals that resemble the LFM phase prior to a particular forecast. Then, these noise samples — having same length as the forecast lead time — are used to drive the system into the future. In practice, the LFMs are characterized by the reconstructed components (RCs) determined by Singular Spectrum Analysis (SSA) [2]. The noise sample selection in [1] was based on finding short analogs of the LFM (as captured by the SSA RCs) in it's past history, which match best the LFM just preceding the start of the forecast. This conditioning of the noise forcing on the initial LFM phase improves forecast skill. Linear response theory — familiar from non-equilibrium statistical physics and dynamical systems theory — helps explain the PNF method's success under suitable circumstances.

Numerical results

As outlined above, the periodically and stochastically forced system (1.1) exhibits LFV with frequency $f' = 0.04 < f$, where $f = 0.25$ is the forcing frequency. It is easy to check numerically that this system also exhibits a pathwise linear response, in the sense described in [1, Main Text].

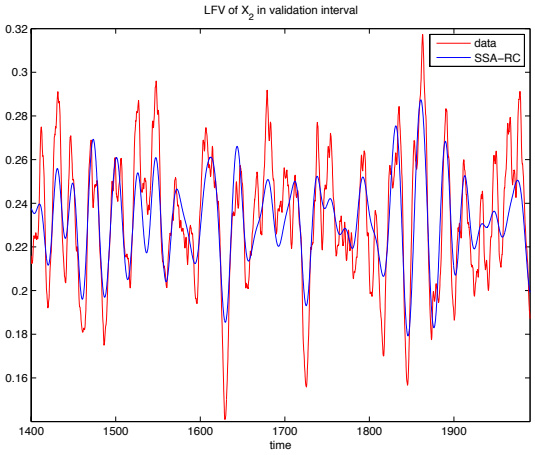

Figure 1: Time series of the component x_2 (red) and its SSA reconstruction given by RC_2 with the mean of x_2 added (blue); RC_2 has a dominant period of 25 (nondimensional units) and captures most of the variance in the prey population x_2. This mode is responsible for the LFV present in the signal.

We will be interested hereafter in the prediction of the *anomalies* x_1^{c} and x_2^{c} of the population variables x_1 and x_2 obtained by integrating the system (1.1), where the superscript $(\cdot)^{\mathrm{c}}$ refers to fact that these anomalies are *centered, i.e.* they have zero mean. Obviously, the PNF method performs similarly for centered or non-centered data: one just has to add the mean back in.

For this numerical demonstration we have applied certain modifications to the PNF procedure of [1] to make it more transparent, and in particular we have adopted ideas from [4] to determine an instantaneous phase $0 < \phi < 2\pi$ of narrow-band LFM components by applying the Hilbert Transform to the SSA reconstruction of the LFV of interest; here for the leading oscillatory pair of x_2^{c} associated with the period $\simeq 25$ as obtained by application of an SSA window

equal to 30. This phase can be uniquely determined for all data points in such an RC, and the noise samples are chosen from the past LFM phase values that are in the small neighborhood ϵ_ϕ to the one at the start time of the forecast. In the results presented here, we used adaptively refined value of ϵ_ϕ, resulting in a typical subset of noise samples of ≈ 30 members for $\epsilon_\phi \approx 0.005$. Another procedure that performed well in practical tests, is where one could get a larger subset of noise samples determined by phase proximity (with larger value of ϵ_ϕ), and then refine it by choosing only those that correspond to initial states in the raw x_2^c data closest to those at the start time of the forecast. Note that by increasing very substantially either subset, ultimately the ergodic limit is obtained of standard ensemble prediction driven by random ensemble of noise samples (ENS).

Skills are shown in Figures 2 and 3, while the validation interval and other technical details are given in the figure captions. The results clearly demonstrate the superiority — in terms of both correlation coefficient, corr, and root-mean-square error, rms, — of the PNF method in forecasting x_2^c against a large, brute-force ensemble of forecasts, for lead times longer than 4, *i.e.*, longer than the forcing period. The PNF method beats thus classical ensemble prediction (labeled as "ENS" in the figures below) for both x_2^c and x_1^c, but the gain in forecasting the former is much more substantial. This difference is clearly due to the fact that the snippets were selected based on the smoothed version of x_2^c alone.

MATLAB scripts (some functions require the Statistics Toolbox)

The following scripts are used to obtain Figures 1-3 in this demo:
run_pnf.m – main script to run example,
pnf_toy_generate.m generate driving noise and perform model integration,
pnftoy.m – advance model in Eq. (1.1) by one time step,
find_pnf.m – make PNF or pure random ensemble (ENS) model forecasts in validation interval; see **run_pnf.m** and **help find_pnf** for detailed info about input and output arguments.
ssapc.m – compute SSA Principal components (PCs),
ssarc.m – compute SSA Reconstructed components (RCs),
gluenoise2.m – generate large ensemble of noise segments ("snippets") from it's time series
center.m – centers (removes the mean) of a time series,
hilbssa.m – computes phase of SSA RCs using Hilbert-Transform,
phasyn.m –find indices of given phase in the time history of SSA reconstruction with specified accuracy (in radians)

- **NB2** The modeled time series has **N=20000** data points and is divided into two parts. The noise snippets are selected in "training" interval [**1 NS-lead**], while the validation interval for prediction is [**NS NE**]. The default parameters **NS=14000**; **NE= N - lead = 20000 - 100=19900**, see more info in **find_pnf.m** and **run_pnf.m**. Note that it will take $\approx$ 35min. to finish **run_pnf.m** with default parameters on MacBook Pro. To make computations faster decrease validation interval – for example make **NS=18000**.

- **NB1** The figures are shown for default parameters without PNF refinement by initial conditions – input parameter **iic=0** is set in **run_pnf.m** to run **find_pnf.m**. As an optional test set **iic=1** to do such refinement for much larger phase-based ensemble – set also $\epsilon = 0.3$ and **NPH= 1000**.

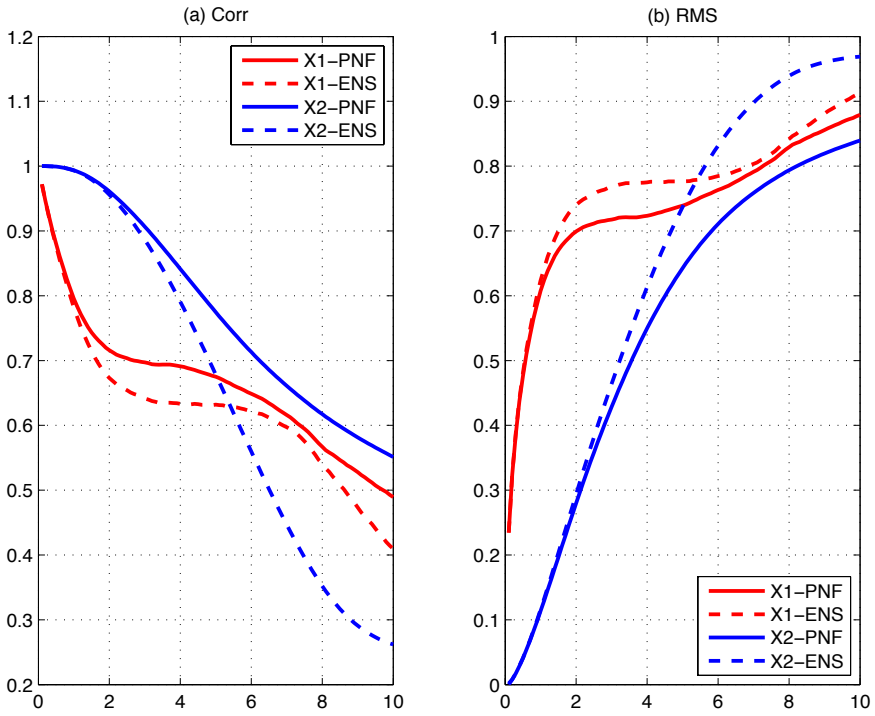

Figure 2: Skill comparison between ensemble predictions (ENS) and PNF predictions for modified Holling model in Eq. (1.1); the ensemble has $N = 100$ members: (a) corr, and (b) rms. The validation interval is $T_f' = 590$ units long, from 1400 to 1990, with 5900 values of t^* issued in steps of $\Delta t = 0.1$ throughout the interval, to make a 10-unit prediction out to the respective $t^* + T$. The snippets are selected in the interval that extends from $t = 0$ to $t = 1400$. The variables predicted are the anomalies x_1^c and x_2^c of the prey and predator densities, x_1 and x_2, respectively.

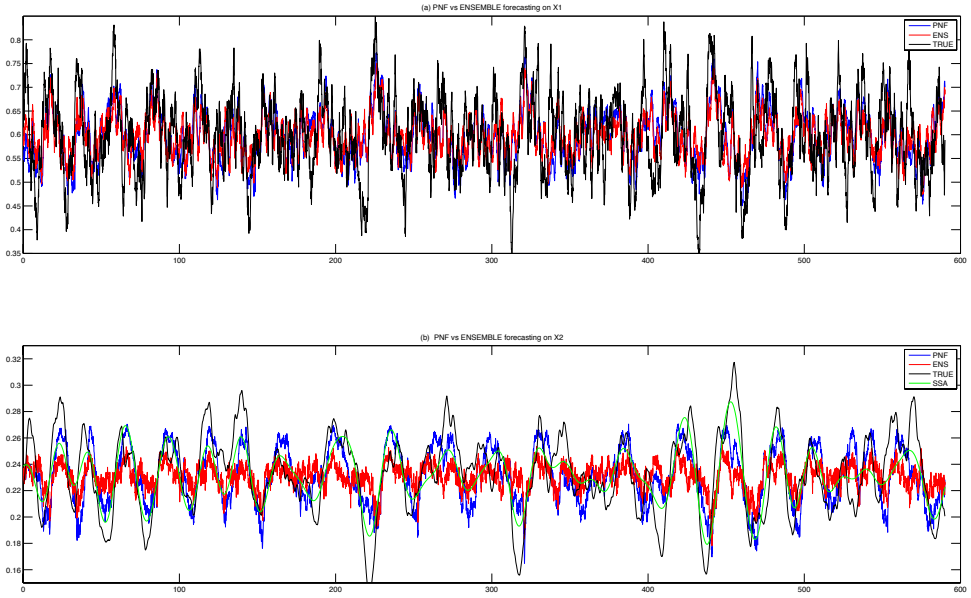

Figure 3: Comparison between ensemble predictions (ENS) and PNF predictions in the time domain: (a) for the x_1^c; and (b) the x_2^c. This plot shows clearly the superior skill of the PNF method — for both variables but especially for x_2^c — when compared with brute-force forecasting with a large ensemble. Note that the drastic improvement of PNF (blue) w.r.t. EMR (red) for x_2 is mostly coincident with energetic phases of LFM, captured by SSA RC (green). The lead time T in this plot is $T = 8$.

3 Empirical Model Reduction (EMR) and Prediction Demo

Matlab script **toy_emrfit.m** demonstrates how by applying general purpose Matlab routine **fitemrplsext.m** solely on a time series $\{x_2 : i = 1, ..., N\}$, a one-variable EMR model [5] in x_2 alone cane be derived to replace the original two-variable model of Eq. (1.1). If multiplicative periodic forcing is included into two-level qubic EMR model on its main level, almost perfect match of autocorrelation function of x_2 is obtained vs. the EMR model without periodic forcing, compare Figures 4 and 5. Note that parametric form of univariate EMR model for x_2 is very different from its *true* form in Eq. (1.1) as it doesn't explicitly account for interactions with periodically and stochastically forced "unresolved" variable x_1. The latter are parameterized by the multi-level EMR-estimated interactions.

Script **toy_emrpredict.m** demonstrates how to apply prediction and cross-validation EMR routine **fcstemrplsext.m**. In addition to EMR fitting parameters such as in **fitemrplsext.m**, input parameters of **fcstemrplsext.m** include specific predictive parameters such as maximum prediction lead time, end of the model training interval, the start and the end of the validation interval which may not necessarily fully overlap with the EMR training interval, i.e. out–of–sample. Output of **fcstemrplsext.m** includes EMR ensemble mean forecast, validation time series, as well as prediction skill in terms of anomaly correlation and normalized root-mean-squared error. See more about available options by **help fcstemrplsext**.

Figure 6 shows that best 2-level qubic EMR model handily beats in prediction linear single level (linear inverse or LIM) model, and that EMR prediction skill is actually almost the same as for the *true ideal* model Eq. (1.1) (compare it with blue dashed lines in Fig. 2a,b).

- **NB1**: When PLS is necessary for multivariate data input, some experimenting is required to find optimal data normalization **inorm** to obtain best predictive EMR model.
- **NB2**: Part of the output diagnostics by **fitemrplsext.m** is the total number of successful EMR simulations attempted to reach the necessary ensemble size as specified in the input parameter. Typically these numbers should be equal and this is another indication of quality of EMR model. Because predictions exceeding very large (by absolute magnitude) nonphysical values are simply discarded, the *optimal* EMR model will typically have none.
- **NB3**: For multi-level EMR model, predictions are initialized automatically back in the past so that random effect of stochastic forcing at the last level is reflected correctly in the forecast.

The following scripts are used to obtain figures for this demo:
toy_emrfit.m – main script to run EMR fit example,
autocorremr.m – compute and compare autocorrelations
toy_emrpredict.m – script to run example of out–of–sample prediction and cross-validation
fcstemrplsext.m – general purpose predictive and cross-validation EMR routine
fcstemrconsext.m – general purpose predictive and cross-validation EMR routine with conserving bilinear terms
model_proj.m –project the out–of–sample data onto multi-level EMR model to estimate the latent variables of addt'l levels and to properly initialize forecasts.

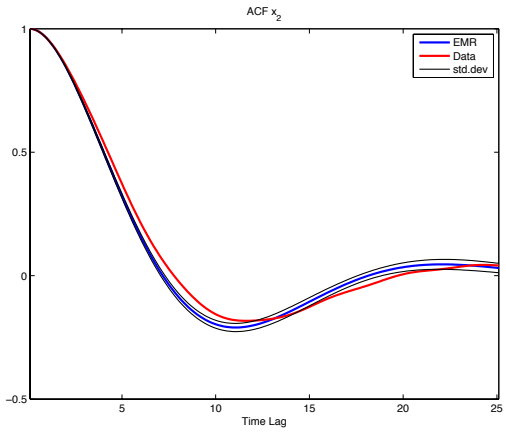

Figure 4: Autocorrelations of x_2: observations obtained by integrating from the full model of Eq. (1.1) –red; ensemble mean of EMR model simulations that doesn't include periodic forcing– blue, standard deviation of EMR ensemble – black; abscissa – time lag.

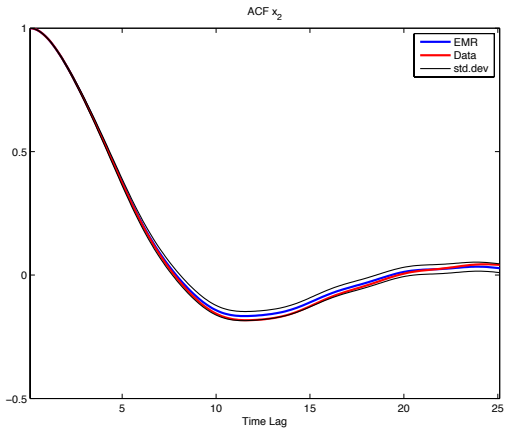

Figure 5: Same as Fig. 4 but when multiplicative periodic forcing is included into EMR model.

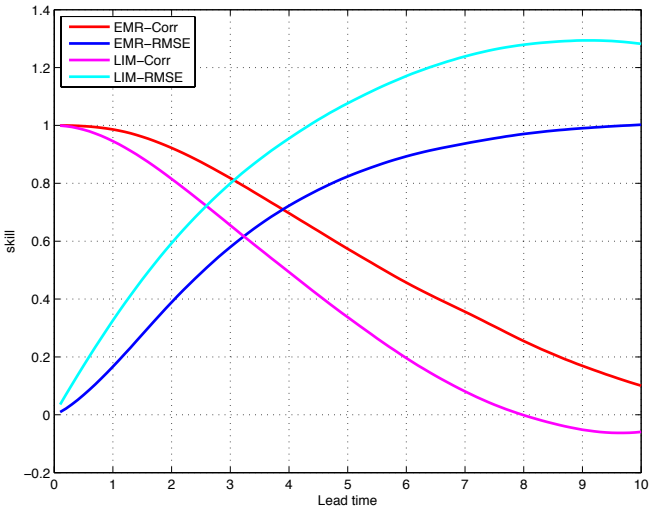

Figure 6: Prediction skill for reduced models obtained solely from x_2 time series; the best cubic 2-level EMR model handily beats LIM both in terms of anomaly correlation and RMS.

References

[1] Chekroun, M., D. Kondrashov, and M. Ghil (2011), Predicting stochastic systems by noise sampling, and application to the El Niño-Southern Oscillation, *Proc. Natl. Acad. Sci. USA*, *108*(29), 11,766–11,771, 10.1073/pnas.101575.

[2] Ghil, M., M. R. Allen, M. D. Dettinger, K. Ide, D. Kondrashov, M. E. Mann, A. W. Robertson, A. Saunders, Y. Tian, F. Varadi, and P. Yiou (2002), Advanced spectral methods for climatic time series, *Rev. Geophys.*, *40*.

[3] Holling, C. (1965), The functional response of predators to prey density and its role in mimicry and population regulation, *Mem. Entomol. Soc. Canada*, **45**, 5–60.

[4] Feliks, Y., M. Ghil, and A. W. Robertson (2010), Oscillatory climate modes in the eastern mediterranean and their synchronization with the north atlantic oscillation, *J. Clim.*, *23*, 4060–4079.

[5] Kondrashov, D., S. Kravtsov, A. W. Robertson and M. Ghil, 2005: A hierarchy of data-based ENSO models. *J. Climate*, **18**, 4425–4444.