

Model order reduction of deep structured state-space models: A system-theoretic approach

Marco Forgone, Manas Mejari, and Dario Piga

¹IDSIA Dalle Molle Institute for Artificial Intelligence SUPSI-USI, Lugano, Switzerland

8th Edition of the Workshop on Nonlinear System Identification
Benchmarks, Lugano, April 26 2024

Deep structured state-space models

Growing interest within the machine-learning community. They dominate **long-range** sequence learning where Transformers suffer the $\mathcal{O}(N^2)$ scaling.

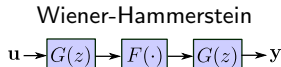
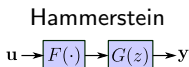
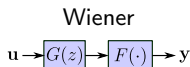
- Interconnection of **linear dynamical** systems with **static non-linearities** (and Normalization layers, skip connection, ...)
- The architecture should **ring a bell** to sysid researchers.

Deep structured state-space models

Growing interest within the machine-learning community. They dominate **long-range** sequence learning where Transformers suffer the $\mathcal{O}(N^2)$ scaling.

- Interconnection of **linear dynamical** systems with **static non-linearities** (and Normalization layers, skip connection, ...)
- The architecture should **ring a bell** to sysid researchers.

The classic **block-oriented** modeling framework.

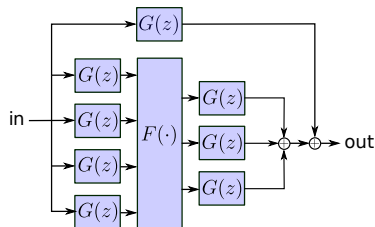


E.W. Bai, F. Giri Block-oriented nonlinear system identification. *Springer*, 2010

The dynoNet architecture

LTI blocks for deep learning also in our 2021 **dynoNet** architecture

dynoNet architecture



Python code

```
G1 = LinearMimo(1, 4, ...) # a SIMO tf
F = StaticNonLin(4, 3, ...) # a static NN
G2 = LinearMimo(3, 1, ...) # a MISO tf
G3 = LinearSiso(1, 1, ...) # a SISO tf
```

```
def model(in_data):
    y1 = G1(in_data)
    z1 = F(y1)
    y2 = G2(z1)
    out = y2 + G3(in_data)
```

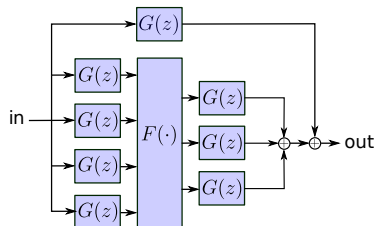


M. Forgiione and D.Piga. *dynoNet*: A Neural Network architecture for learning dynamical systems. *International Journal of Adaptive Control and Signal Processing*, 2021

The dynoNet architecture

LTI blocks for deep learning also in our 2021 **dynoNet** architecture

dynoNet architecture



Python code

```
G1 = LinearMimo(1, 4, ...) # a SIMO tf
F = StaticNonLin(4, 3, ...) # a static NN
G2 = LinearMimo(3, 1, ...) # a MISO tf
G3 = LinearSiso(1, 1, ...) # a SISO tf
```

```
def model(in_data):
    y1 = G1(in_data)
    z1 = F(y1)
    y2 = G2(z1)
    out = y2 + G3(in_data)
```



M. Forgiione and D.Piga. *dynoNet*: A Neural Network architecture for learning dynamical systems.
International Journal of Adaptive Control and Signal Processing, 2021

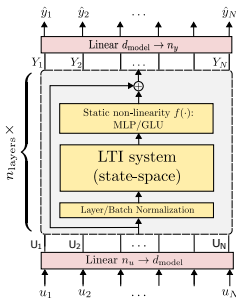
Differentiable transfer functions are now also in the **torchaudio** library.



<https://pytorch.org/audio/main/generated/torchaudio.functional.lfilter.html>

Deep Structured State-Space Model Architecture

- Architecture with normalization layers, skip connections, (dropout).
- State-space parameterization of the linear dynamical system



Focus on making the LTI system learning **fast and well-posed**.

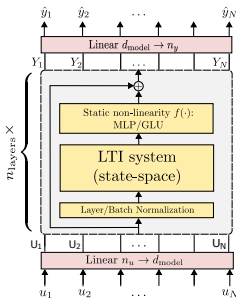


A. Gu, K. Goel, C. Ré. Efficiently Modeling Long Sequences with Structured State Spaces. *ICLR*, 2022

Our idea: bring in **model order reduction** to simplify these architectures!

Deep Structured State-Space Model Architecture

- Architecture with normalization layers, skip connections, (dropout).
- State-space parameterization of the linear dynamical system



Focus on making the LTI system learning **fast and well-posed**.




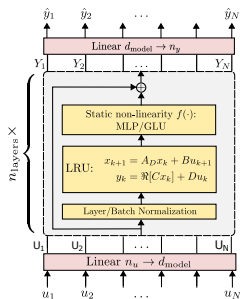
A. Gu, K. Goel, C. Ré. Efficiently Modeling Long Sequences with Structured State Spaces. *ICLR*, 2022

Our idea: bring in **model order reduction** to simplify these architectures!

Deep Linear Recurrent Unit

We consider the deep LRU architecture recently introduced by DeepMind:

 A. Orvieto et al. Resurrecting Recurrent Neural Networks for Long Sequences. *ICML*, 2023



- Discrete-time, MIMO LTI system
- Complex diagonal state-transition matrix A_D :


$$A_D = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_{n_x})$$

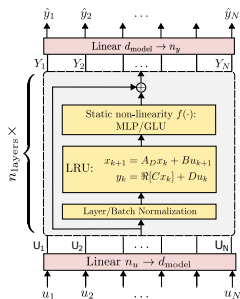
- Stable parameterization with $|\lambda_i| < 1$
- Implementation with either:
 - ▶ $\mathcal{O}(N)$ sequential ops (standard recursion)
 - ▶ N parallel jobs, $\mathcal{O}(\log N)$ ops each (parallel scan)

Our idea: further exploit the diagonal structure for model order reduction.

Deep Linear Recurrent Unit

We consider the deep LRU architecture recently introduced by DeepMind:

 A. Orvieto et al. Resurrecting Recurrent Neural Networks for Long Sequences. *ICML*, 2023



- Discrete-time, MIMO LTI system
- Complex diagonal state-transition matrix A_D :

$$A_D = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_{n_x})$$

- Stable parameterization with $|\lambda_i| < 1$
- Implementation with either:
 - ▶ $\mathcal{O}(N)$ sequential ops (standard recursion)
 - ▶ N parallel jobs, $\mathcal{O}(\log N)$ ops each (parallel scan)

Our idea: further exploit the **diagonal structure** for model order reduction.

Model Order Reduction

Consider a LTI state-space system partitioned as:

$$\begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1(k-1) \\ x_2(k-1) \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u(k),$$
$$y(k) = [C_1 \ C_2] \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + Du(k),$$

The states x_2 can be removed by:

- **Truncation** \Rightarrow keep (A_{11}, B_1, C_1, D)
- **Singular Perturbation** \Rightarrow solve for $x_2(k) = x_2(k-1)$

Reduction is often applied to systems in

- **Modal form** (A diagonal, states sorted from slowest to fastest)
- **Balanced form** (states sorted for decreasing Hankel singular values)

We tested all combinations: MT, MSP, BT, BSP.

Model Order Reduction

Consider a LTI state-space system partitioned as:

$$\begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1(k-1) \\ x_2(k-1) \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u(k),$$
$$y(k) = [C_1 \ C_2] \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + Du(k),$$

The states x_2 can be removed by:

- **Truncation** \Rightarrow keep (A_{11}, B_1, C_1, D)
- **Singular Perturbation** \Rightarrow solve for $x_2(k) = x_2(k-1)$

Reduction is often applied to systems in

- **Modal form** (A diagonal, states sorted from slowest to fastest)
- **Balanced form** (states sorted for decreasing Hankel singular values)

We tested all combinations: MT, MSP, BT, BSP.

Model Order Reduction

Consider a LTI state-space system partitioned as:

$$\begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1(k-1) \\ x_2(k-1) \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u(k),$$
$$y(k) = [C_1 \ C_2] \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + Du(k),$$

The states x_2 can be removed by:

- **Truncation** \Rightarrow keep (A_{11}, B_1, C_1, D)
- **Singular Perturbation** \Rightarrow solve for $x_2(k) = x_2(k-1)$

Reduction is often applied to systems in

- **Modal form** (A diagonal, states sorted from slowest to fastest)
- **Balanced form** (states sorted for decreasing Hankel singular values)

We tested all combinations: MT, MSP, BT, BSP.

Regularization for MOR

Regularization introduced to enhance the MOR:

$$\theta^* = \arg \min_{\theta \in \Theta} \frac{1}{N} \sum_{k=1}^N (y_k - \hat{y}_k(\theta))^2 + \gamma \mathcal{R}(\theta),$$

Modal ℓ_1

$$\mathcal{R}(\theta) = \sum_{j=1}^{n_x} |\lambda_j|$$

- Push some modes λ_j towards 0
- Tailored for modal reduction
MT/MSP

Hankel nuclear norm (HNN)

$$\mathcal{R}(\theta) = \sum_{j=1}^{n_x} \sigma_j$$

- Push some HSV σ_j towards 0
- Tailored for balanced reduction
BT/BSP

- Modal ℓ_1 efficient (λ_j are on the diagonal of A_D)
- HNN also efficient exploiting diagonal A_D structure...

Regularization for MOR

Regularization introduced to enhance the MOR:

$$\theta^* = \arg \min_{\theta \in \Theta} \frac{1}{N} \sum_{k=1}^N (y_k - \hat{y}_k(\theta))^2 + \gamma \mathcal{R}(\theta),$$

Modal ℓ_1

$$\mathcal{R}(\theta) = \sum_{j=1}^{n_x} |\lambda_j|$$

- Push some modes λ_j towards 0
- Tailored for modal reduction
MT/MSP

Hankel nuclear norm (HNN)

$$\mathcal{R}(\theta) = \sum_{j=1}^{n_x} \sigma_j$$

- Push some HSV σ_j towards 0
- Tailored for balanced reduction
BT/BSP

- Modal ℓ_1 efficient (λ_j are on the diagonal of A_D)
- HNN also efficient exploiting diagonal A_D structure...

Hankel nuclear norm minimization

- The HNN $\sum_{j=1}^{n_x} \sigma_j$ is a convex approximation to the McMillan degree of a linear system G (minimum realization order).
- The choice of HNN regularization is further motivated by the bound:

$$\|G - G_r\|_{\mathcal{H}_\infty} \leq 2 \sum_{j=r+1}^{n_x} \sigma_j,$$

valid when G_r is obtained with balanced reduction methods BT/BSP

If we push some HSVs of G towards zero, we can then find an (almost) equivalent reduced G_r with balanced reduction methods!

Hankel nuclear norm minimization

- The HNN $\sum_{j=1}^{n_x} \sigma_j$ is a convex approximation to the McMillan degree of a linear system G (minimum realization order).
- The choice of HNN regularization is further motivated by the bound:

$$\|G - G_r\|_{\mathcal{H}_\infty} \leq 2 \sum_{j=r+1}^{n_x} \sigma_j,$$

valid when G_r is obtained with balanced reduction methods BT/BSP

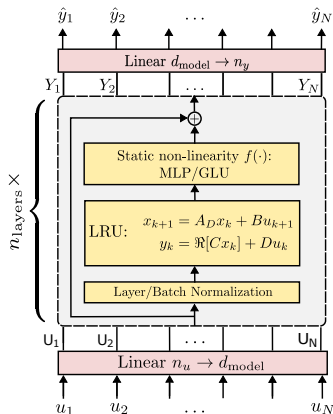
If we push some HSVs of G towards zero, we can then then find an (almost) equivalent reduced G_r with balanced reduction methods!


Example

Experiments on the F16 ground vibration benchmark.

Deep LRU with:

- $n_{\text{layers}} = 6$ layers
- $n_x = 100$ states per layer
- $d_{\text{model}} = 50$ units per layer
- Layer Normalization
- MLP non-linearity

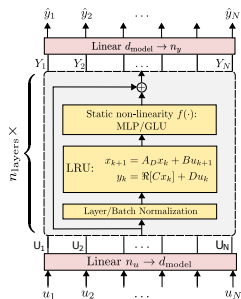


 <https://github.com/forgi86/lru-reduction>

Nominal results

Training repeated with:

- No regularization
- Modal ℓ_1 regularization
- Hankel norm regularization.



Results on test set FullMSine_Level16_Validation:

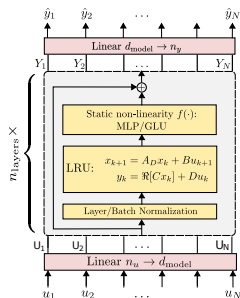
Regularization	Channel 1			Channel 2			Channel 3		
	<i>fit</i>	RMSE	NRMSE	<i>fit</i>	RMSE	NRMSE	<i>fit</i>	RMSE	NRMSE
No reg.	86.5	0.180	0.134	90.0	0.167	0.099	76.2	0.368	0.237
Modal ℓ_1	85.4	0.195	0.145	89.8	0.171	0.102	74.5	0.395	0.254
Hankel norm	85.8	0.190	0.142	89.0	0.185	0.110	75.5	0.379	0.245

In line with literature. Regularization has a large effect on the LTI blocks!

Nominal results

Training repeated with:

- No regularization
- Modal ℓ_1 regularization
- Hankel norm regularization.

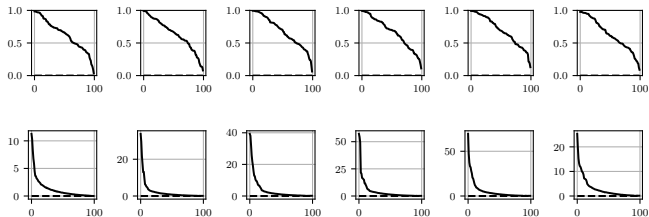


Results on test set FullMSine_Level16_Validation:

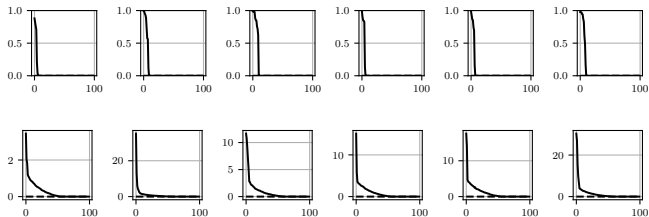
Regularization	Channel 1			Channel 2			Channel 3		
	fit	RMSE	NRMSE	fit	RMSE	NRMSE	fit	RMSE	NRMSE
No reg.	86.5	0.180	0.134	90.0	0.167	0.099	76.2	0.368	0.237
Modal ℓ_1	85.4	0.195	0.145	89.8	0.171	0.102	74.5	0.395	0.254
Hankel norm	85.8	0.190	0.142	89.0	0.185	0.110	75.5	0.379	0.245

In line with literature. Regularization has a large effect on the LTI blocks!

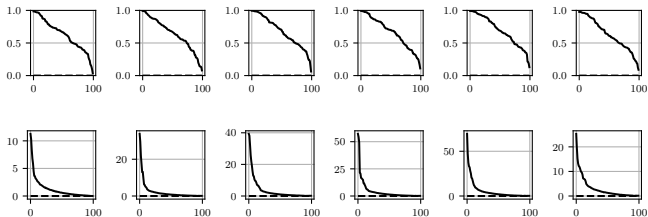
No regularization: eigenvalues magnitude (top) and HSV (bottom)



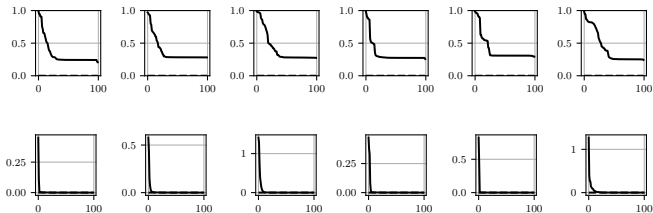
Modal ℓ_1 regularization: eigenvalues magnitude (top) and HSV (bottom)



No regularization: eigenvalues magnitude (top) and HSV (bottom)



HNN regularization: eigenvalues magnitude (top) and HSV (bottom)

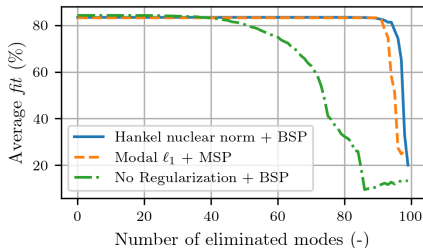


Model Order Reduction

Performance of all Regularization/Reduction combinations assessed.

Regularization Method	Reduction Method			
	BT	BSP	MT	MSP
No Regularization	28	43	3	35
Modal ℓ_1	56	73	0	91
Hankel nuclear norm	89	91	18	76

Table: Number of states eliminated s.t. performance decrease is less than 1%



- Best results with Hankel nuclear norm + balanced singular perturbation and modal ℓ_1 + modal singular perturbation
- Without regularization, MOR is significantly less effective

Conclusions & future research

Preliminary efforts to improve deep SSMs with System Theoretic tools.
MOR + tailor-made regularization to reduce state dimensionality.

Future research:

- More extensive simulations (e.g., effect of the regularization strength)
- Other model order reduction (e.g., Kyrilov-based) and regularizers
- Reduction also of layers and channels
- Application to other models where LTI blocks are key, e.g.
 - ▶ Koopman-based
 - ▶ dynoNet
 - ▶ Other deep SSMs

Conclusions & future research

Preliminary efforts to improve deep SSMs with System Theoretic tools.
MOR + tailor-made regularization to reduce state dimensionality.

Future research:

- More extensive simulations (e.g., effect of the regularization strength)
- Other model order reduction (e.g., Kyrilov-based) and regularizers
- Reduction also of layers and channels
- Application to other models where LTI blocks are key, e.g.
 - ▶ Koopman-based
 - ▶ dynoNet
 - ▶ Other deep SSMs

Thank you.
Questions?

`marco.forgione@idsia.ch`

Model Order Reduction

- Modal reduction is almost directly applicable to the LRU, which is already in modal (diagonal) form.
 - 1 Sort system (A_D, B, C, D) for decreasing eigenvalues magnitude
 - 2 Eliminate fastest modes
- Balanced reduction techniques require a **three-step** procedure:
 - 1 Balance (A_D, B, C, D) to a non-diagonal (A_b, B_b, C_d, D)
 - 2 Reduce (A_b, B_b, C_d, D) to a non-diagonal (A_r, B_r, C_r, D)
 - 3 Diagonalize (A_r, B_r, C_r, D) to fit the LRU structure