WESTFÄLISCHE WILHELMS-UNIVERSITÄT MÜNSTER

APPLIED MATHEMATICS MÜNSTER

Model Reduction for Inverse Network Models

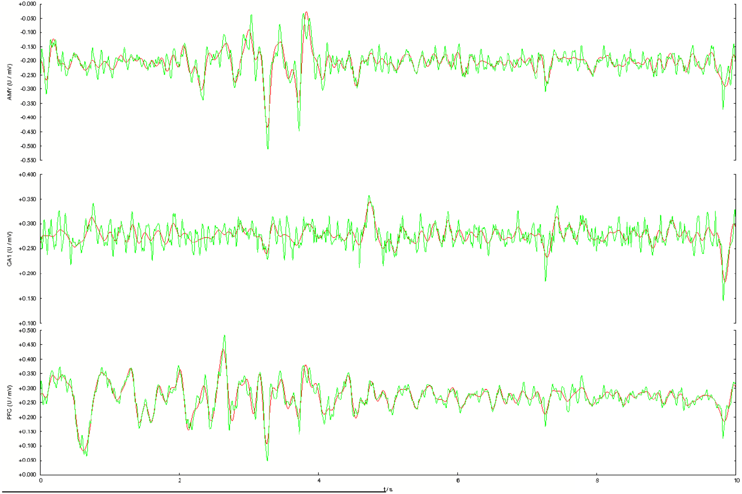
Christian Himpe (christian.himpe@uni-muenster.de)
Mario Ohlberger (mario.ohlberger@uni-muenster.de)

WWU Münster
Institute for Computational and Applied Mathematics

06.12.2013

Motivation

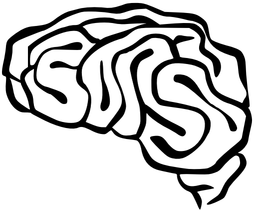
Intracranial EEG and tuned model output[1]:



[1] [Himpe'11]

Application

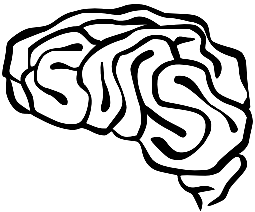
- Effective Connectivity
- Causal Connectivity
- Learning / Unlearning



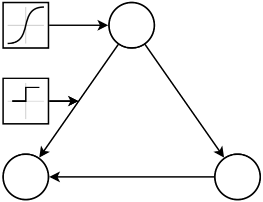
- Experiments (EEG/MEG, fMRI/fNIRS)
- Deduce Connectivity

Application

- Effective Connectivity
- Causal Connectivity
- Learning / Unlearning



- Network Model
- Dynamical System
- Control System



- Experiments (EEG/MEG, fMRI/fNIRS)
- Deduce Connectivity

- Large-Scale
- Inverse Problem

Control Systems

Linear Dynamical System

$$\begin{aligned} \dot{x}(t) &= Ax(t) \qquad x \in \mathbb{R}^N,\ A \in \mathbb{R}^{N \times N} \\ x(0) &= x_0 \end{aligned} \tag{1}$$

(1) A is the adjacency matrix of a weighted directed graph.

Control Systems

Linear Dynamical System

$$\dot{x}(t) = Ax(t) \qquad x \in \mathbb{R}^N,\ A \in \mathbb{R}^{N\times N} \qquad (1)$$
$$x(0) = x_0$$

Linear Control System

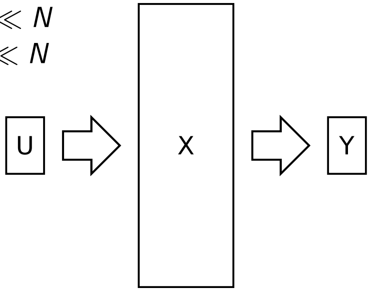
$$\dot{x}(t) = Ax(t) + Bu(t) \qquad u \in \mathbb{R}^M,\ B \in \mathbb{R}^{N\times M}$$
$$y(t) = Cx(t) \qquad x \in \mathbb{R}^N,\ A \in \mathbb{R}^{N\times N}$$
$$x(0) = x_0 \qquad y \in \mathbb{R}^O,\ C \in \mathbb{R}^{O\times N}$$

(1) A is the adjacency matrix of a weighted directed graph.

Model Reduction

Context:

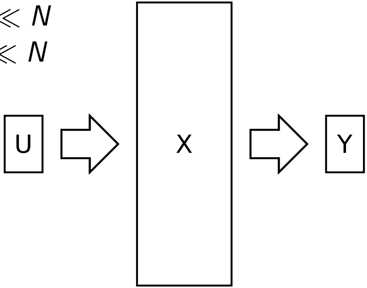
- $N = \dim(x) \gg 1$
- $M = \dim(u) \ll N$
- $O = \dim(y) \ll N$



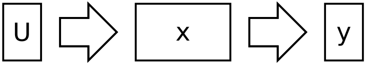
Model Reduction

Context:

- $N = \dim(x) \gg 1$
- $M = \dim(u) \ll N$
- $O = \dim(y) \ll N$



Reduced Order Model:



- $M \ll N$
- $\|Y - y\| \ll 1$

Linear Model Reduction

Aim:

1. Identifiy important and less important states.
2. Compute projection sorting states by importance.
3. Truncate neglectable states.

Input-To-Output map:

$$u \longmapsto x \longmapsto y$$

Linear Model Reduction

Aim:

1. Identifiy important and less important states.
2. Compute projection sorting states by importance.
3. Truncate neglectable states.

Input-To-Output map:

$$u \longmapsto x \longmapsto y$$

Input-To-State map:

$$u \longmapsto x$$

Controllability

State-To-Output map:

$$x \longmapsto y$$

Observability

Controllability & Observability

Impulse Response:

$$G(t) = Ce^{At}B,\ t > 0$$

Hankel Operator:

$$H = \int_0^\infty Ce^{At}Bdt$$

Controllability Operator:

$$C = \int_0^\infty e^{At}Bdt$$

Observability Operator:

$$O = \int_0^\infty Ce^{At}dt$$

Controllability Gramian:

$$W_C := CC^*$$

Observability Gramian:

$$W_O := O^*O$$

$$\sigma_i := \lambda(H) = \sqrt{\lambda(W_O W_C)}$$

Balanced Truncation[2]

Controllability Gramian W_C:

Lyapunov equation:

$$AW_C + W_C A^T = -BB^T$$

If $\mathrm{Re}(\lambda(A)) < 0$):

$$W_C = \int_0^\infty e^{At} BB^T e^{A^T t} dt$$

Observability Gramian W_O:

Lyapunov equation:

$$A^T W_O + W_O A = -C^T C$$

If $\mathrm{Re}(\lambda(A)) < 0$):

$$W_O = \int_0^\infty e^{A^T t} C^T C e^{At} dt$$

Balancing: Sort by least controllable AND least observable states.

$$\exists U, V : UV = \mathbb{1}, VW_C V^T = U^T W_O U = \begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \end{pmatrix}$$

[2] [Moore'81]

Direct Truncation[4] (Approximate Balancing)

Cross Gramian[3] W_X**:**

Sylvester equation:

$$AW_X + W_X A = -BC$$

If $\mathrm{Re}(\lambda(A)) < 0$):

$$W_X = \int_0^\infty e^{At} B C e^{At} dt$$

NO balancing required!

$$W_O W_C = W_X^2 \Rightarrow \sigma_i = |\lambda(W_X)| \Rightarrow W_X = UDV \approx U \begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \end{pmatrix} V$$

---

[3] Assume $\Sigma = \{A, B, C\}$ symmetric $\Leftarrow CA^{-1}B$ symmetric!

[4] Review in [Antoulas'05]

Truncation

1 $$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx \\ x(0) = x_0 \end{cases}$$

Truncation

1 $$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx \\ x(0) = x_0 \end{cases}$$

---

2 $$\begin{cases} \dot{x} = VAUx + VBu \\ y = CUx \\ x(0) = Vx_0 \end{cases}$$

Truncation

1
$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx \\ x(0) = x_0 \end{cases}$$

2
$$\begin{cases} \dot{x} = VAUx + VBu \\ y = CUx \\ x(0) = Vx_0 \end{cases}$$

3
$$\begin{cases} \dot{x} = \tilde{A}x + \tilde{B}u \\ y = \tilde{C}x \\ x(0) = \tilde{x}_0 \end{cases}$$

Truncation

1
$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx \\ x(0) = x_0 \end{cases}$$

2
$$\begin{cases} \dot{x} = VAUx + VBu \\ y = CUx \\ x(0) = Vx_0 \end{cases}$$

3
$$\begin{cases} \dot{x} = \tilde{A}x + \tilde{B}u \\ y = \tilde{C}x \\ x(0) = \tilde{x}_0 \end{cases}$$

4
$$\begin{cases} \dot{x} = \begin{pmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ \tilde{A}_{21} & \tilde{A}_{22} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} \tilde{B}_1 \\ \tilde{B}_2 \end{pmatrix} u \\ y = \begin{pmatrix} \tilde{C}_1 & \tilde{C}_2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \\ x(0) = \begin{pmatrix} \tilde{x}_{0,1} \\ \tilde{x}_{0,2} \end{pmatrix} \end{cases}$$

Truncation

1 $$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx \\ x(0) = x_0 \end{cases}$$

---

2 $$\begin{cases} \dot{x} = VAUx + VBu \\ y = CUx \\ x(0) = Vx_0 \end{cases}$$

---

3 $$\begin{cases} \dot{x} = \tilde{A}x + \tilde{B}u \\ y = \tilde{C}x \\ x(0) = \tilde{x}_0 \end{cases}$$

---

4 $$\begin{cases} \dot{x} = \begin{pmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ \tilde{A}_{21} & \tilde{A}_{22} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} \tilde{B}_1 \\ \tilde{B}_2 \end{pmatrix} u \\ y = \begin{pmatrix} \tilde{C}_1 & \tilde{C}_2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \\ x(0) = \begin{pmatrix} \tilde{x}_{0,1} \\ \tilde{x}_{0,2} \end{pmatrix} \end{cases}$$

---

5 $$\begin{cases} \dot{\tilde{x}} = \tilde{A}_{11}\tilde{x}_1 + \tilde{B}_1 u \\ \tilde{y} = \tilde{C}_1\tilde{x}_1 \\ \tilde{x}(0) = \tilde{x}_{0,1} \end{cases}$$

Nonlinear Control Systems

Linear Control System

$$\dot{x}(t) = Ax(t) + Bu(t) \qquad u \in \mathbb{R}^M,\ B \in \mathbb{R}^{N\times M}$$

$$y(t) = Cx(t) \qquad x \in \mathbb{R}^N,\ A \in \mathbb{R}^{N\times N}$$

$$x(0) = x_0 \qquad y \in \mathbb{R}^O,\ C \in \mathbb{R}^{O\times N}$$

General Control System

$$\dot{x}(t) = f(x(t), u(t)) \qquad u \in \mathbb{R}^M,$$

$$y(t) = g(x(t), u(t)) \qquad x \in \mathbb{R}^N,\ f : \mathbb{R}^N \times \mathbb{R}^M \to \mathbb{R}^N$$

$$x(0) = x_0 \qquad y \in \mathbb{R}^O,\ g : \mathbb{R}^N \times \mathbb{R}^M \to \mathbb{R}^O$$

What now?

Empirical Gramians[5]

- Empirical Controllability Gramian: $W_C = \langle \int_0^\infty x_U(t) x_U^*(t) dt \rangle_U$
- Empirical Observability Gramian: $W_O = \langle \int_0^\infty \rho(y_X^*(t) y_X(t)) dt \rangle_X$
- Empirical Cross Gramian: $W_X = \langle \int_0^\infty \varphi(x_U(t), y_X(t)) dt \rangle_{U \times X}$

with perturbation spaces

1. U for perturbing the input u,
2. X for perturbing the initial state x_0

assembled from

- rotations (orthogonal matrices),
- scaling (real numbers),
- for each input / state (unit normal vectors),

determined by the operating range of the underlying contol system.

---

5 [Lall'99],[Hahn'02],[Streif'06],[Himpe'13a]

Empirical Gramians II

Note:

- For linear systems, the "empirical" equal the "classic" gramians.
- Computation requires only basic matrix and vector operations.

General Projection Framework:

$$\dot{\tilde{x}}(t) = Vf(U\tilde{x}(t), u(t))$$
$$\tilde{y}(t) = g(U\tilde{x}(t), u(t))$$
$$\tilde{x}(0) = Vx_0$$

Thus, **Nonlinear Model Reduction** can also use

- Balanced Truncation,
- Direct Truncation.

The same tools as for linear model reduction.

Parameter Reduction

Parametrized Control System

$$\dot{x}(t) = f(x(t), u(t), \theta) \qquad u \in \mathbb{R}^M,\ \theta \in \mathbb{R}^P$$

$$y(t) = g(x(t), u(t), \theta) \qquad x \in \mathbb{R}^N,\ f : \mathbb{R}^N \times \mathbb{R}^M \times \mathbb{R}^P \to \mathbb{R}^N$$

$$x(0) = x_0 \qquad y \in \mathbb{R}^O,\ g : \mathbb{R}^N \times \mathbb{R}^M \times \mathbb{R}^P \to \mathbb{R}^O$$

Aim:

1. Identifiy important and less important parameters.
2. Compute projection sorting parameters by importance.
3. Truncate neglectable parameters.

Parameter Reduction

Parametrized Control System

$$\dot{x}(t) = f(x(t), u(t), \theta) \qquad u \in \mathbb{R}^M,\ \theta \in \mathbb{R}^P$$

$$y(t) = g(x(t), u(t), \theta) \qquad x \in \mathbb{R}^N,\ f : \mathbb{R}^N \times \mathbb{R}^M \times \mathbb{R}^P \to \mathbb{R}^N$$

$$x(0) = x_0 \qquad y \in \mathbb{R}^O,\ g : \mathbb{R}^N \times \mathbb{R}^M \times \mathbb{R}^P \to \mathbb{R}^O$$

Aim:

1. Identifiy important and less important parameters.
2. Compute projection sorting parameters by importance.
3. Truncate neglectable parameters.

1. Identifiy important and less important states.
2. Compute projection sorting states by importance.
3. Truncate neglectable states.

Parameter Reduction

Parametrized Control System

$$\dot{x}(t) = f(x(t), u(t), \theta) \qquad u \in \mathbb{R}^M,\ \theta \in \mathbb{R}^P$$

$$y(t) = g(x(t), u(t), \theta) \qquad x \in \mathbb{R}^N,\ f : \mathbb{R}^N \times \mathbb{R}^M \times \mathbb{R}^P \to \mathbb{R}^N$$

$$x(0) = x_0 \qquad y \in \mathbb{R}^O,\ g : \mathbb{R}^N \times \mathbb{R}^M \times \mathbb{R}^P \to \mathbb{R}^O$$

Aim:

1. Identifiy important and less important parameters.
2. Compute projection sorting parameters by importance.
3. Truncate neglectable parameters.

1. Identifiy important and less important states.
2. Compute projection sorting states by importance.
3. Truncate neglectable states.

Handle parameters as:

- constant inputs
- constant states

More Empirical Gramians[6]

Empirical Sensitivity Gramian (Controllability-Based) W_S :

$$\hat{u} = \begin{pmatrix} u \\ \theta \end{pmatrix} \rightarrow \dot{x} = f(x, \hat{u}) = f(x, u) + \sum_{k=1}^{P} f(x, \theta_k) \rightarrow W_S = \delta_{i,j} \operatorname{trace}(W_{C,i})$$

Empirical Identifiability Gramian (Observability-Based) W_I :

$$\hat{x} = \begin{pmatrix} x \\ \theta \end{pmatrix} \rightarrow \dot{\hat{x}} = f(\hat{x}, u) = \begin{pmatrix} f(x, u, \theta) \\ 0 \end{pmatrix}, \hat{x}(0) = \begin{pmatrix} x_0 \\ \theta \end{pmatrix} \Rightarrow W_I = S(\hat{W}_O)$$

Empirical Joint Gramian (Cross-Gramian-Based) W_J :

$$\hat{x} = \begin{pmatrix} x \\ \theta \end{pmatrix} \rightarrow \dot{\hat{x}} = f(\hat{x}, u) = \begin{pmatrix} f(x, u, \theta) \\ 0 \end{pmatrix}, \hat{x}(0) = \begin{pmatrix} x_0 \\ \theta \end{pmatrix} \Rightarrow W_{\dot{I}} = \bar{S}(W_J := \hat{W}_X)$$

[6] [Sun'06], [Geffen'08], [Himpe'13a]

Combined Reduction

Controllability-Based

1. Compute W_S
   $\rightarrow W_C$
2. Decompose W_S
3. Truncate θ
4. Compute W_O
5. Balance W_C, W_O
6. Decompose W_{CO}
7. Truncate x

Observability-Based

1. Compute W_I
   $\rightarrow W_O$
2. Decompose W_I
3. Truncate θ
4. Compute W_C
5. Balance W_C, W_O
6. Decompose W_{CO}
7. Truncate x

Cross-Gramian-Based

1. Compute $W_{\ddot{I}}$
   $\rightarrow W_X$
2. Decompose $W_{\ddot{I}}$
3. Truncate θ
4. Decompose W_X
5. Truncate x

emgr - Empirical Gramian Framework[7]

Gramians:

- Empirical Controllability Gramian
- Empirical Observability Gramian
- Empirical Cross Gramian
- Empirical Sensitivity Gramian
- Empirical Identifiability Gramian
- Empirical Joint Gramian

Features:

- Uniform Interface
- Compatible with MATLAB & OCTAVE
- Vectorized & Parallelizable
- Open-Source licensed

More info at: `http://gramian.de`

[7] see [Himpe'13]

Numerical Experiment

Hyperbolic Network Model[8]:

$$\begin{aligned} \dot{x}(t) &= A(\theta)\tanh(Kx(t)) + Bu(t) \\ y(t) &= Cx(t) \\ x(0) &= x_0 \end{aligned}$$

Dimensions:

- $\dim(u) = \{4, 5, 6, 7, 8\}$
- $\dim(y) = \dim(u)$
- $\dim(x) = \{16, 25, 36, 49, 64\},\ \dim(\tilde{x}) \overset{!}{=} \dim(u)$
- $\dim(\theta) = \dim(x)^2,\ \dim(\tilde{\theta}) \overset{!}{=} \dim(\tilde{x})^2$

[8] [Quan'01]

Setup

Synthetic Data:

1. Generate random network (θ) with ensured stability of $A(\theta)$.
2. Integrate System to obtain system output.
3. Add Gaussian noise to output.

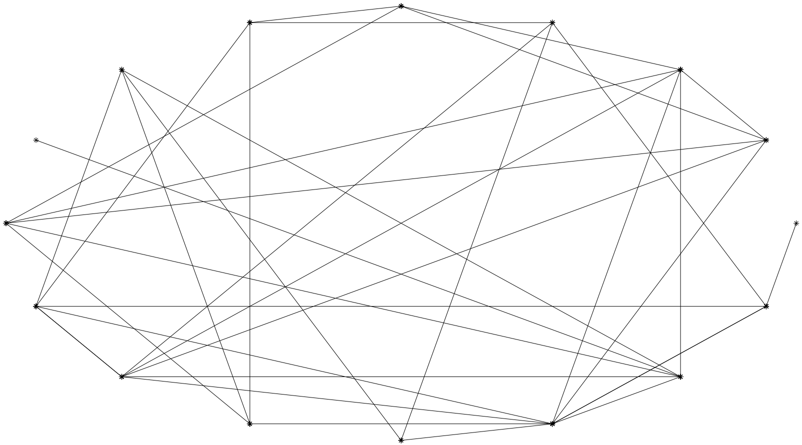
Inverse Problem[9]:

1. Offline Phase:
   Compute reduced order model using combined reduction.
2. Online Phase:
   Optimize reduced model using least squares.

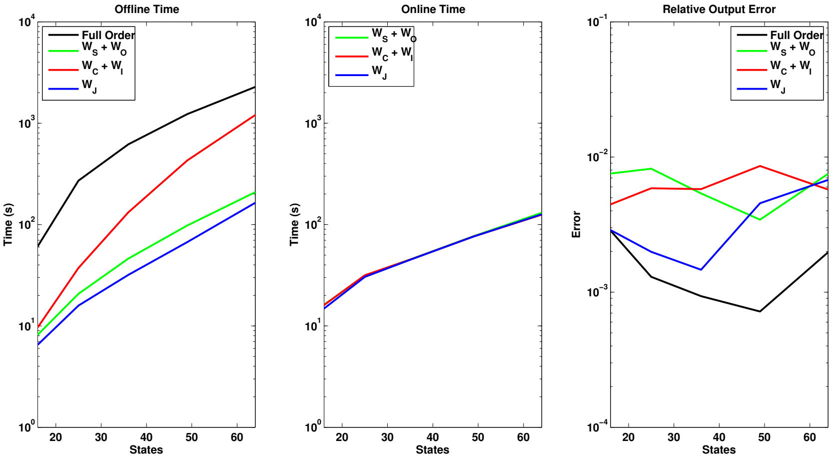
---

[9] from here on θ is unknown

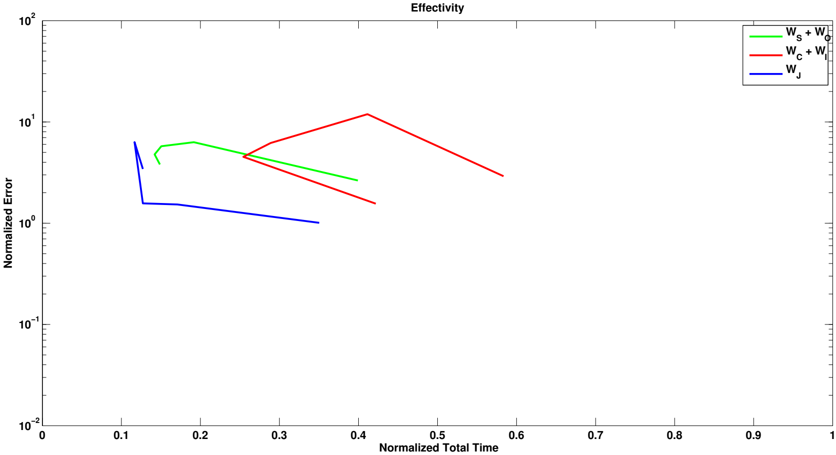
Network (N=16)



Numerical Results



Reduction Effectivity



tl;dl

- Large-Scale Inverse Problems: Model Reduction.
- Model Reduction for Control Systems: Gramian-Based.
- Nonlinear Model Order Reduction: Empirical Gramians.
- Combined Reduction: Reduction of States and Parameters.
- (Empirical) Cross Gramian: very efficient!

- Get the Source Code: `http://j.mp/zifmtn13` .

Thanks!