Model and controller order reduction

Graduate course on Optimal and Robust Control

Zdeněk Hurák

Department of Control Engineering
Faculty of Electrical Engineering
Czech Technical University in Prague

May 18, 2021

1. Motivation and approaches for model and controller order reduction
2. Model order reduction
3. Modes
   - Reachability and observability directions
4. Balanced truncation
5. Other balancing methods
   - Balanced stochastic truncation
6. Hankel norm approximation
7. Frequency weighted approximation and controller order reduction
   - Frequency weighted approximation
   - Controller order reduction

Motivation

- Models for complex systems can be of very high order, especially when modelling spatially distributed systems (PDE) by spatial discretization (FEM, FD)

Motivation

- Models for complex systems can be of very high order, especially when modelling spatially distributed systems (PDE) by spatial discretization (FEM, FD)
- Modern model-based control design methods (LQG, $\mathcal{H}_2$, $\mathcal{H}_\infty$, μ-synthesis) give controllers of (about) the same order as the model of the plant

Motivation

- Models for complex systems can be of very high order, especially when modelling spatially distributed systems (PDE) by spatial discretization (FEM, FD)
- Modern model-based control design methods (LQG, $\mathcal{H}_2$, $\mathcal{H}_\infty$, μ-synthesis) give controllers of (about) the same order as the model of the plant
  - Reduce the order of the model first and then design a controller for the reduced-order model

Motivation

- Models for complex systems can be of very high order, especially when modelling spatially distributed systems (PDE) by spatial discretization (FEM, FD)
- Modern model-based control design methods (LQG, $\mathcal{H}_2$, $\mathcal{H}_\infty$, μ-synthesis) give controllers of (about) the same order as the model of the plant
  - Reduce the order of the model first and then design a controller for the reduced-order model
  - Design a controller for the full-order model and then approximate the controller with a reduced-order controller

Motivation

- Models for complex systems can be of very high order, especially when modelling spatially distributed systems (PDE) by spatial discretization (FEM, FD)
- Modern model-based control design methods (LQG, $\mathcal{H}_2$, $\mathcal{H}_\infty$, μ-synthesis) give controllers of (about) the same order as the model of the plant
  - Reduce the order of the model first and then design a controller for the reduced-order model
  - Design a controller for the full-order model and then approximate the controller with a reduced-order controller
    Controller-order reduction cannot be done just by applying some technique for model-order reduction to a full controller

Motivation

- Models for complex systems can be of very high order, especially when modelling spatially distributed systems (PDE) by spatial discretization (FEM, FD)
- Modern model-based control design methods (LQG, $\mathcal{H}_2$, $\mathcal{H}_\infty$, μ-synthesis) give controllers of (about) the same order as the model of the plant
  - Reduce the order of the model first and then design a controller for the reduced-order model
  - Design a controller for the full-order model and then approximate the controller with a reduced-order controller
    Controller-order reduction cannot be done just by applying some technique for model-order reduction to a full controller
  - (Design a low-order controller directly)

- $\boxed{\text{SVD methods}}$

- $\boxed{\text{SVD methods}}$
- Krylov subspace methods

- $\boxed{\text{SVD methods}}$
- Krylov subspace methods
- SVD-Krylov methods

- $\boxed{\text{SVD methods}}$
- Krylov subspace methods
- SVD-Krylov methods

- $\boxed{\text{SVD methods}}$
- Krylov subspace methods
- SVD-Krylov methods

Literature

1. A. C. Antoulas, Approximation of Large-Scale Dynamical Systems. Philadelphia: Society for Industrial and Applied Mathematic, 2005.

- $\boxed{\text{SVD methods}}$
- Krylov subspace methods
- SVD-Krylov methods

Literature

1. A. C. Antoulas, Approximation of Large-Scale Dynamical Systems. Philadelphia: Society for Industrial and Applied Mathematic, 2005.
2. G. Obinata and B. D. O. Anderson, Model Reduction for Control System Design, 2001 edition. New York: Springer, 2000.

- $\boxed{\text{SVD methods}}$
- Krylov subspace methods
- SVD-Krylov methods

Literature

1. A. C. Antoulas, Approximation of Large-Scale Dynamical Systems. Philadelphia: Society for Industrial and Applied Mathematic, 2005.
2. G. Obinata and B. D. O. Anderson, Model Reduction for Control System Design, 2001 edition. New York: Springer, 2000.

Useful in their own right (modeling and simulation).

Inspiration from SVD applied on matrices, images, ...

Inspiration from SVD applied on matrices, images, ...

```
load clown
size(X)

figure(1)
image(X)
colormap(map)
```

```
[U,S,V] = svd(X);
k = 10
Xred = U(:,1:k)*S(1:k,1:k)*V(:,1:k)';

figure(2)
image(Xred)
colormap(map)
```

```
s = diag(S)

figure(3)
plot(s,'.')
ylabel('Singular values')
```

Bounds on the error (Schmidt-Eckart-Young-Mirsky)

Bounds on the error (Schmidt-Eckart-Young-Mirsky)

$$\boxed{\min_{X, rank(X)=k} \|A - X\|_2 = \sigma_{k+1}(A)}$$

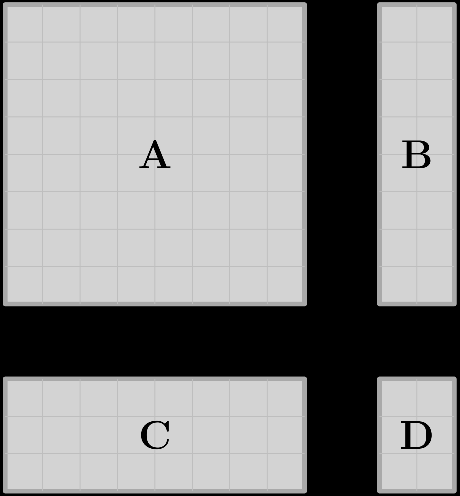
provided $\sigma_i > \sigma_{i+1}$.

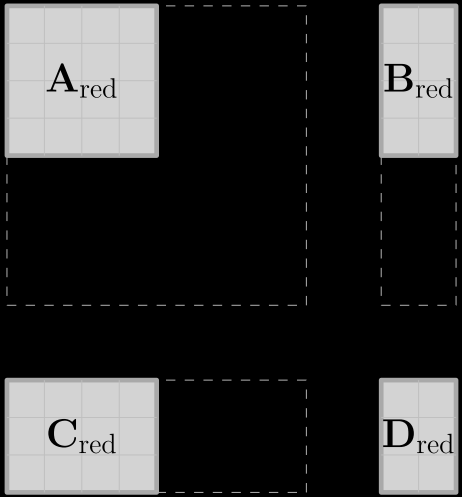
Bounds on the error (Schmidt-Eckart-Young-Mirsky)

$$\boxed{\min_{X, rank(X)=k} \|A - X\|_2 = \sigma_{k+1}(A)}$$

provided $\sigma_i > \sigma_{i+1}$.
A nonunique minimizer is obtained by truncating the dyadic decomposition

$$X_{\min} = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \ldots + \sigma_k u_k v_k^T$$


A
B
C
D



$\mathbf{A}_{\text{red}}$
$\mathbf{B}_{\text{red}}$
$\mathbf{C}_{\text{red}}$
$\mathbf{D}_{\text{red}}$

Full order model

$$\begin{aligned}
\dot{x}_1(t) &= A_{11}x_1(t) + A_{12}x_2(t) + B_1u(t) \\
\dot{x}_2(t) &= A_{21}x_1(t) + A_{22}x_2(t) + B_2u(t) \\
y(t) &= C_1x_1(t) + C_2x_2(t) + Du(t)
\end{aligned}$$

Full order model

$$\begin{aligned}\dot{x}_1(t) &= A_{11}x_1(t) + A_{12}x_2(t) + B_1u(t)\\ \dot{x}_2(t) &= A_{21}x_1(t) + A_{22}x_2(t) + B_2u(t)\\ y(t) &= C_1x_1(t) + C_2x_2(t) + Du(t)\end{aligned}$$

Reduced model (assuming $x_2 = 0$, relabelling x_1 as x)

$$\begin{aligned}\dot{x}(t) &= A_{11}x(t) + B_1u(t)\\ y(t) &= C_1x(t) + Du(t)\end{aligned}$$

Full order model

$$\begin{aligned}\dot{x}_1(t) &= A_{11}x_1(t) + A_{12}x_2(t) + B_1u(t)\\ \dot{x}_2(t) &= A_{21}x_1(t) + A_{22}x_2(t) + B_2u(t)\\ y(t) &= C_1x_1(t) + C_2x_2(t) + Du(t)\end{aligned}$$

Reduced model (assuming $x_2 = 0$, relabelling x_1 as x)

$$\begin{aligned}\dot{x}(t) &= A_{11}x(t) + B_1u(t)\\ y(t) &= C_1x(t) + Du(t)\end{aligned}$$

Matlab (Control System Toolbox): **modred(G,ix,'Truncate')**

Also *singular perturbation*. Assume

$$\dot{x}_2(t) = 0 = A_{21}x_1(t) + A_{22}x_2(t) + B_2u(t)$$

Also *singular perturbation*. Assume

$$\dot{\mathsf{x}}_2(t) = 0 = \mathsf{A}_{21}\mathsf{x}_1(t) + \mathsf{A}_{22}\mathsf{x}_2(t) + \mathsf{B}_2\mathsf{u}(t)$$

from which (assuming A_{22} is nonsingular), we express x_2 and substitute to the first state equation

Also *singular perturbation*. Assume

$$\dot{x}_2(t) = 0 = A_{21}x_1(t) + A_{22}x_2(t) + B_2u(t)$$

from which (assuming A_{22} is nonsingular), we express x_2 and substitute to the first state equation and get

$$\begin{aligned}\dot{x}_1(t) &= A_r x_1(t) + B_r u(t)\\ y(t) &= C_r x_1(t) + D_r u(t)\end{aligned}$$

where

$$\begin{aligned}A_r &= A_{11} - A_{12}A_{22}^{-1}A_{21}\\ B_r &= B_1 - A_{12}A_{22}^{-1}B_2\\ C_r &= C_1 - C_2A_{22}^{-1}A_{21}\\ D_r &= D - C_2A_{22}^{-1}B_2\end{aligned}$$

Also *singular perturbation*. Assume

$$\dot{x}_2(t) = 0 = A_{21}x_1(t) + A_{22}x_2(t) + B_2u(t)$$

from which (assuming A_{22} is nonsingular), we express x_2 and substitute to the first state equation and get

$$\begin{aligned} \dot{x}_1(t) &= A_r x_1(t) + B_r u(t) \\ y(t) &= C_r x_1(t) + D_r u(t) \end{aligned}$$

where

$$\begin{aligned} A_r &= A_{11} - A_{12}A_{22}^{-1}A_{21} \\ B_r &= B_1 - A_{12}A_{22}^{-1}B_2 \\ C_r &= C_1 - C_2A_{22}^{-1}A_{21} \\ D_r &= D - C_2A_{22}^{-1}B_2 \end{aligned}$$

Matlab: **modred(G,ix,'MatchdDC')**

$\boxed{\text{Both truncation and residualization depend on the chosen basis}}$

Modal truncation

Diagonal (modal) realization

Diagonal (modal) realization $\Leftrightarrow$ Each mode is projected into a respective state variable.

Diagonal (modal) realization $\Leftrightarrow$ Each mode is projected into a respective state variable.
The remaining modes are left intact (appreaciated in some domains where physical interpretation of modes is important: aerospace, ...)

Diagonal (modal) realization $\Leftrightarrow$ Each mode is projected into a respective state variable.
The remaining modes are left intact (appreaciated in some domains where physical interpretation of modes is important: aerospace, ...)
General heuristic rule: eliminating the fastest modes, hence truncating the corresponding state variables.

Diagonal (modal) realization $\Leftrightarrow$ Each mode is projected into a respective state variable.

The remaining modes are left intact (appreaciated in some domains where physical interpretation of modes is important: aerospace, ...)

General heuristic rule: eliminating the fastest modes, hence truncating the corresponding state variables.

```
w(1) = 0.56806689746895; zeta(1) = 0.00096819582773; k(1) = 0.01651378989774;
w(2) = 3.94093897440699; zeta(2) = 0.00100229920475; k(2) = 0.00257034576009;
w(3) = 10.58229653714164;zeta(3) = 0.00100167293203; k(3) = 0.00002188016252;
w(4) = 16.19234386986640;zeta(4) = 0.01000472824082; k(4) = 0.00027927762861;

G = ss(0);
for i=1:4
    Gi = k(i)*tf(w(i)^2,[1,2*zeta(i)*w(i),w(i)^2]);
    G = G + ss(Gi);
end
```

```
Gmt = modred(G,5:8,'Truncate')
```

Modal truncation caveats

Generally works fine, but sometimes fast modes can affect the input-output behaviour more than the low frequency ones

Generally works fine, but sometimes fast modes can affect the input-output behaviour more than the low frequency ones

```
k(1) = 0.01; k(2) = 0.005; k(3) = 0.007; k(4) = 0.004;

G = ss(0);
for i=1:4
    Gi = k(i)*tf(w(i)^2,[1,2*zeta(i)*w(i),w(i)^2]);
    G = G + ss(Gi);
end
```

```
Gmt = modred(G,5:8,'Truncate')          % Try 'MatchDC' option too
```

Recap: Eigendecomposition

Recap: Eigendecomposition

$$\mathsf{A}\mathsf{v} = \lambda \mathsf{v}$$

λ is an eigenvalue, v is an associated (right) eigenvector.

$$\mathsf{A}\mathsf{v} = \lambda \mathsf{v}$$

λ is an eigenvalue, v is an associated (right) eigenvector.

$$\mathsf{A}\mathsf{V} = \mathsf{V}\mathsf{D}$$

$$\mathsf{A}\mathsf{v} = \lambda \mathsf{v}$$

λ is an eigenvalue, v is an associated (right) eigenvector.

$$\mathsf{A}\mathsf{V} = \mathsf{V}\mathsf{D}, \qquad \mathsf{A} = \mathsf{V}\mathsf{D}\mathsf{V}^{-1}$$

$$\mathsf{Av} = \lambda \mathsf{v}$$

λ is an eigenvalue, v is an associated (right) eigenvector.

$$\mathsf{AV} = \mathsf{VD}, \qquad \mathsf{A} = \mathsf{VDV}^{-1}$$

Similarly the left version

$$\mathsf{w}^{\mathrm{T}}\mathsf{A} = \mathsf{w}^{\mathrm{T}}\lambda$$

$$\mathsf{A}\mathsf{v} = \lambda \mathsf{v}$$

λ is an eigenvalue, v is an associated (right) eigenvector.

$$\mathsf{AV} = \mathsf{VD}, \qquad \mathsf{A} = \mathsf{VDV}^{-1}$$

Similarly the left version

$$\mathsf{w}^{\mathrm{T}}\mathsf{A} = \mathsf{w}^{\mathrm{T}}\lambda$$

$$\mathsf{W}^{\mathrm{T}}\mathsf{A} = \mathsf{D}\mathsf{W}^{\mathrm{T}}, \qquad \mathsf{A}^{\mathrm{T}}\mathsf{W} = \mathsf{WD}$$

$$\mathsf{A}\mathsf{v} = \lambda \mathsf{v}$$

λ is an eigenvalue, v is an associated (right) eigenvector.

$$\mathsf{AV} = \mathsf{VD}, \qquad \mathsf{A} = \mathsf{VDV}^{-1}$$

Similarly the left version

$$\mathsf{w}^{\mathrm{T}}\mathsf{A} = \mathsf{w}^{\mathrm{T}}\lambda$$

$$\mathsf{W}^{\mathrm{T}}\mathsf{A} = \mathsf{DW}^{\mathrm{T}}, \qquad \mathsf{A}^{\mathrm{T}}\mathsf{W} = \mathsf{WD}$$

$$\mathsf{w}_i^{\mathrm{T}}\mathsf{v}_j = 0, \qquad \mathsf{w}_i^{\mathrm{T}}\mathsf{v}_i = 1, \qquad i \neq j.$$

Matlab: **eig()**

```
>> A = [1, 2, 3; 4, 5, 6; 7, 8, 9]
A =
     1     2     3
     4     5     6
     7     8     9
>> [V,D,W] = eig(A)
V =
   -0.2320   -0.7858    0.4082
   -0.5253   -0.0868   -0.8165
   -0.8187    0.6123    0.4082
D =
   16.1168         0         0
         0   -1.1168         0
         0         0   -0.0000
W =
   -0.4645   -0.8829    0.4082
   -0.5708   -0.2395   -0.8165
   -0.6770    0.4039    0.4082
```

$$e^{\mathsf{A}t} = \mathsf{A}_1 e^{\lambda_1 t} + \mathsf{A}_2 e^{\lambda_2 t} + \ldots + \mathsf{A}_n e^{\lambda_n t}$$

where $\lambda_1, \lambda_2, \ldots, \lambda_n$ are eigenvalues of A (for $\lambda_i \neq \lambda_j$)

$$e^{At} = A_1 e^{\lambda_1 t} + A_2 e^{\lambda_2 t} + \ldots + A_n e^{\lambda_n t}$$

where $\lambda_1, \lambda_2, \ldots, \lambda_n$ are eigenvalues of A (for $\lambda_i \neq \lambda_j$)
or

$$e^{At} = A_1 e^{\lambda_1 t} + A_2 t e^{\lambda_1 t} + A_3 e^{\lambda_3 t} + \ldots + A_n e^{\lambda_n t}$$

for $\lambda_1 = \lambda_2$,

$$e^{\mathsf{A}t} = \mathsf{A}_1 e^{\lambda_1 t} + \mathsf{A}_2 e^{\lambda_2 t} + \ldots + \mathsf{A}_n e^{\lambda_n t}$$

where $\lambda_1, \lambda_2, \ldots, \lambda_n$ are eigenvalues of A (for $\lambda_i \neq \lambda_j$)
or

$$e^{\mathsf{A}t} = \mathsf{A}_1 e^{\lambda_1 t} + \mathsf{A}_2 t e^{\lambda_1 t} + \mathsf{A}_3 e^{\lambda_3 t} + \ldots + \mathsf{A}_n e^{\lambda_n t}$$

for $\lambda_1 = \lambda_2$,

$$\mathsf{A}_i = \mathsf{v}_i \mathsf{w}_i^{\mathrm{T}}, \qquad i = 1, \ldots, n.$$

$$e^{\mathsf{A}t} = \mathsf{A}_1 e^{\lambda_1 t} + \mathsf{A}_2 e^{\lambda_2 t} + \ldots + \mathsf{A}_n e^{\lambda_n t}$$

where $\lambda_1, \lambda_2, \ldots, \lambda_n$ are eigenvalues of A (for $\lambda_i \neq \lambda_j$)
or

$$e^{\mathsf{A}t} = \mathsf{A}_1 e^{\lambda_1 t} + \mathsf{A}_2 t e^{\lambda_1 t} + \mathsf{A}_3 e^{\lambda_3 t} + \ldots + \mathsf{A}_n e^{\lambda_n t}$$

for $\lambda_1 = \lambda_2$,

$$\mathsf{A}_i = \mathsf{v}_i \mathsf{w}_i^{\mathrm{T}}, \qquad i = 1, \ldots, n.$$

$$\mathsf{x}(t) = e^{\mathsf{A}t}\mathsf{x}(0) = \mathsf{A}_1 e^{\lambda_1 t}\mathsf{x}(0) + \mathsf{A}_2 e^{\lambda_2 t}\mathsf{x}(0) + \ldots + \mathsf{A}_n e^{\lambda_n t}\mathsf{x}(0).$$

Solution of state equations, Matrix exponential, Modes

$$e^{\mathsf{A}t} = \mathsf{A}_1 e^{\lambda_1 t} + \mathsf{A}_2 e^{\lambda_2 t} + \ldots + \mathsf{A}_n e^{\lambda_n t}$$

where $\lambda_1, \lambda_2, \ldots, \lambda_n$ are eigenvalues of A (for $\lambda_i \neq \lambda_j$)

or

$$e^{\mathsf{A}t} = \mathsf{A}_1 e^{\lambda_1 t} + \mathsf{A}_2 t e^{\lambda_1 t} + \mathsf{A}_3 e^{\lambda_3 t} + \ldots + \mathsf{A}_n e^{\lambda_n t}$$

for $\lambda_1 = \lambda_2$,

$$\mathsf{A}_i = \mathsf{v}_i \mathsf{w}_i^{\mathrm{T}}, \qquad i = 1, \ldots, n.$$

$$\mathsf{x}(t) = e^{\mathsf{A}t}\mathsf{x}(0) = \mathsf{A}_1 e^{\lambda_1 t}\mathsf{x}(0) + \mathsf{A}_2 e^{\lambda_2 t}\mathsf{x}(0) + \ldots + \mathsf{A}_n e^{\lambda_n t}\mathsf{x}(0).$$

$$\boxed{\mathsf{x}(t) = \mathsf{v}_1 e^{\lambda_1 t} \underbrace{\mathsf{w}_1^{\mathrm{T}}\mathsf{x}(0)}_{\alpha_1} + \mathsf{v}_2 e^{\lambda_2 t} \underbrace{\mathsf{w}_2^{\mathrm{T}}\mathsf{x}(0)}_{\alpha_2} + \ldots + \mathsf{v}_n e^{\lambda_n t} \underbrace{\mathsf{w}_n^{\mathrm{T}}\mathsf{x}(0)}_{\alpha_n}.}$$

$$y(t) = Cx(t)$$

$$\mathsf{y}(t) = \mathsf{C}\mathsf{x}(t)$$

In the single-output case

$$\boxed{y(t) = \underbrace{\mathsf{C}\mathsf{v}_1\mathsf{w}_1^{\mathrm{T}}\mathsf{x}(0)}_{\gamma_1} e^{\lambda_1 t} + \underbrace{\mathsf{C}\mathsf{v}_2\mathsf{w}_2^{\mathrm{T}}\mathsf{x}(0)}_{\gamma_2} e^{\lambda_1 t} + \ldots + \underbrace{\mathsf{C}\mathsf{v}_n\mathsf{w}_n^{\mathrm{T}}\mathsf{x}(0)}_{\gamma_n} e^{\lambda_n t}.}$$

$$\mathrm{y}(t) = \mathrm{C}\mathrm{x}(t)$$

In the single-output case

$$\boxed{y(t) = \underbrace{\mathrm{C}\mathrm{v}_1\mathrm{w}_1^{\mathrm{T}}\mathrm{x}(0)}_{\gamma_1} e^{\lambda_1 t} + \underbrace{\mathrm{C}\mathrm{v}_2\mathrm{w}_2^{\mathrm{T}}\mathrm{x}(0)}_{\gamma_2} e^{\lambda_1 t} + \ldots + \underbrace{\mathrm{C}\mathrm{v}_n\mathrm{w}_n^{\mathrm{T}}\mathrm{x}(0)}_{\gamma_n} e^{\lambda_n t}.}$$

In Laplace domain

$$Y(s) = \frac{\gamma_1}{s-\lambda_1} + \frac{\gamma_2}{s-\lambda_2} + \ldots + \frac{\gamma_n}{s-\lambda_n}.$$

$$\mathsf{Q} = \int_0^\infty e^{\mathsf{A}^\mathsf{T} t}\mathsf{C}^\mathsf{T} \underbrace{\mathsf{C}e^{\mathsf{A}t}}_{\text{state-to-output map}} dt$$

$$\mathsf{Q} = \int_0^\infty e^{\mathsf{A}^\mathsf{T} t} \mathsf{C}^\mathsf{T} \underbrace{\mathsf{C} e^{\mathsf{A} t}}_{\text{state-to-output map}} dt$$

Q is matrix,

$$Q = \int_0^\infty e^{A^T t} C^T \underbrace{C e^{A t}}_{\text{state-to-output map}} dt$$

Q is matrix, $Q = Q^{\mathrm{T}}$,

$$\mathsf{Q} = \int_0^\infty e^{\mathsf{A}^\mathsf{T} t} \mathsf{C}^\mathsf{T} \underbrace{\mathsf{C} e^{\mathsf{A} t}}_{\text{state-to-output map}} dt$$

Q is matrix, $\mathsf{Q} = \mathsf{Q}^{\mathrm{T}}$, $\mathsf{Q} \geq 0$;

$$Q = \int_0^\infty e^{A^T t} C^T \underbrace{C e^{At}}_{\text{state-to-output map}} dt$$

Q is matrix, $Q = Q^T$, $Q \geq 0$; For (A, C) stable and observable $Q > 0$.

$$Q = \int_0^\infty e^{A^T t} C^T \underbrace{C e^{At}}_{\text{state-to-output map}} dt$$

Q is matrix, $Q = Q^T$, $Q \geq 0$; For (A, C) stable and observable $Q > 0$.
It gives "energy" observed at the output

$$\boxed{\|y\|_2^2 = x^T(0) Q x(0)}$$

$$Q = \int_0^\infty e^{A^T t} C^T \underbrace{C e^{At}}_{\text{state-to-output map}} dt$$

Q is matrix, $Q = Q^T$, $Q \geq 0$; For (A, C) stable and observable $Q > 0$.
It gives "energy" observed at the output

$$\boxed{\|y\|_2^2 = x^T(0) Q x(0)}$$

Directional analysis (through eigenvalue decomposition):

$$Q = U_Q \Sigma_Q U_Q^*, \qquad Q = U_Q \Sigma_Q U_Q^T \text{ for real eigenvectors}$$

Computation of the (infinite-time) observability gramian

Computation of the (infinite-time) observability gramian

Continuous-time Lyapunov matrix

$$A^T Q + QA + C^T C = 0$$

Computation of the (infinite-time) observability gramian

Continuous-time Lyapunov matrix

$$A^T Q + QA + C^T C = 0$$

Matlab: **lyap()**, **gram()**

Computation of the (infinite-time) observability gramian

Continuous-time Lyapunov matrix

$$A^T Q + QA + C^T C = 0$$

Matlab: **lyap()**, **gram()**
Discrete-time Lyapunov matrix

$$A^T QA + C^T C = Q$$

Computation of the (infinite-time) observability gramian

Continuous-time Lyapunov matrix

$$A^T Q + QA + C^T C = 0$$

Matlab: **lyap()**, **gram()**
Discrete-time Lyapunov matrix

$$A^T QA + C^T C = Q$$

Matlab: **dlyap()**, **gram()**

(Infinite-time) reachability gramian

(Infinite-time) reachability gramian

$$P = \int_0^\infty \underbrace{e^{At}B}_{\text{input-to-state map}} B^T e^{A^T t} dt$$

(Infinite-time) reachability gramian

$$P = \int_0^\infty \underbrace{e^{At}B}_{\text{input-to-state map}} B^T e^{A^T t} dt$$

$P = P^{\mathrm{T}} \in \mathbb{R}^{n\times n}$, $P \geq 0$;

$$\mathsf{P} = \int_0^\infty \underbrace{e^{\mathsf{A}t}\mathsf{B}}_{\text{input-to-state map}} \mathsf{B}^\mathsf{T} e^{\mathsf{A}^\mathsf{T} t} dt$$

$\mathsf{P} = \mathsf{P}^{\mathrm{T}} \in \mathbb{R}^{n \times n}$, $\mathsf{P} \geq 0$; For (A, B) stable and reachable/controllable $\mathsf{P} > 0$.

(Infinite-time) reachability gramian

$$\mathsf{P} = \int_0^\infty \underbrace{e^{\mathsf{A}t}\mathsf{B}}_{\text{input-to-state map}} \mathsf{B}^\mathsf{T} e^{\mathsf{A}^\mathsf{T} t} dt$$

$\mathsf{P} = \mathsf{P}^{\mathrm{T}} \in \mathbb{R}^{n\times n}$, $\mathsf{P} \geq 0$; For (A, B) stable and reachable/controllable $\mathsf{P} > 0$.

Recall

$$\mathsf{x}(t) = e^{\mathsf{A}t}\mathsf{x}(0) + \int_0^t e^{\mathsf{A}(t-\tau)}\mathsf{B}\mathsf{u}(\tau)d\tau$$

(Infinite-time) reachability gramian

$$\mathsf{P} = \int_0^\infty \underbrace{e^{\mathsf{A}t}\mathsf{B}}_{\text{input-to-state map}} \mathsf{B}^\mathsf{T} e^{\mathsf{A}^\mathsf{T} t} dt$$

$\mathsf{P} = \mathsf{P}^{\mathrm{T}} \in \mathbb{R}^{n \times n}$, $\mathsf{P} \geq 0$; For (A, B) stable and reachable/controllable $\mathsf{P} > 0$.
Recall

$$\mathsf{x}(t) = e^{\mathsf{A}t}\mathsf{x}(0) + \int_0^t e^{\mathsf{A}(t-\tau)}\mathsf{B}\mathsf{u}(\tau) d\tau$$

Minimum energy signal $\mathsf{u}_{\mathsf{min}}$ needed to bring system to a given state $\mathsf{x}(0) = \mathsf{x}_{\mathrm{r}}$

$$\mathsf{P} = \int_0^\infty \underbrace{e^{\mathsf{A}t}\mathsf{B}}_{\text{input-to-state map}} \mathsf{B}^\mathsf{T} e^{\mathsf{A}^\mathsf{T} t} dt$$

$\mathsf{P} = \mathsf{P}^{\mathrm{T}} \in \mathbb{R}^{n \times n}$, $\mathsf{P} \geq 0$; For (A, B) stable and reachable/controllable $\mathsf{P} > 0$.
Recall

$$\mathsf{x}(t) = e^{\mathsf{A}t}\mathsf{x}(0) + \int_0^t e^{\mathsf{A}(t-\tau)}\mathsf{B}\mathsf{u}(\tau) d\tau$$

Minimum energy signal $\mathsf{u}_{\mathsf{min}}$ needed to bring system to a given state $\mathsf{x}(0) = \mathsf{x}_\mathrm{r}$ has the energy

$$\boxed{\|\mathsf{u}_{\mathsf{min}}\|_2^2 = \mathsf{x}_\mathrm{r}^\mathsf{T} \mathsf{P}^{-1} \mathsf{x}_\mathrm{r}}$$

$$P = \int_0^\infty \underbrace{e^{At}B}_{\text{input-to-state map}} B^T e^{A^T t} dt$$

$P = P^T \in \mathbb{R}^{n \times n}$, $P \geq 0$; For (A, B) stable and reachable/controllable $P > 0$.
Recall

$$x(t) = e^{At}x(0) + \int_0^t e^{A(t-\tau)}Bu(\tau)d\tau$$

Minimum energy signal u_{min} needed to bring system to a given state $x(0) = x_r$ has the energy

$$\boxed{\|u_{min}\|_2^2 = x_r^T P^{-1} x_r}$$

→ BEARD, Randal W. Linear operator equations with applications in control and signal processing. IEEE Control Systems Magazine, 2002, 22.2: 69-79.

$$P = \int_0^\infty \underbrace{e^{At}B}_{\text{input-to-state map}} \quad B^T e^{A^T t} dt$$

$P = P^T \in \mathbb{R}^{n \times n}$, $P \geq 0$; For (A, B) stable and reachable/controllable $P > 0$.
Recall

$$x(t) = e^{At}x(0) + \int_0^t e^{A(t-\tau)}Bu(\tau)d\tau$$

Minimum energy signal u_{min} needed to bring system to a given state $x(0) = x_r$ has the energy

$$\boxed{\|u_{min}\|_2^2 = x_r^T P^{-1} x_r}$$

→ BEARD, Randal W. Linear operator equations with applications in control and signal processing. IEEE Control Systems Magazine, 2002, 22.2: 69-79.

Do the same directionality analysis using $P = U_P \Sigma_P U_P^*$

Computation of the (infinite) reachability gramian

Continuous-time Lyapunov equation

$$\mathsf{AP} + \mathsf{PA}^\mathsf{T} + \mathsf{BB}^\mathsf{T} = 0$$

Matlab: **lyap()**
System reachable iff $\mathsf{P} > 0$.
Discrete-time Lyapunov equation

$$\mathsf{APA}^\mathsf{T} + \mathsf{BB}^\mathsf{T} = \mathsf{P}$$

Matlab: **dlyap()**

Example in Matlab

1. Different modes exhibit themselves in different directions in the state space (they are differently projected into the state variables).

1. Different modes exhibit themselves in different directions in the state space (they are differently projected into the state variables).
2. Strongly observable modes can be weakly controllable and vice versa.

1. Different modes exhibit themselves in different directions in the state space (they are differently projected into the state variables).
2. Strongly observable modes can be weakly controllable and vice versa.

Simple truncation or residualization will not work well. So what?

1. Find a state-space realization where **direction** in which the system is easily controlled are **aligned** with the directions in which the states are easily observable.

1. Find a state-space realization where **direction** in which the system is easily controlled are **aligned** with the directions in which the states are easily observable. Aligning the principal axes of the reachability and observability ellipsoids.

1. Find a state-space realization where **direction** in which the system is easily controlled are **aligned** with the directions in which the states are easily observable. Aligning the principal axes of the reachability and observability ellipsoids. **Ballanced realization**.
2. Remove the state variables that are weakly involved in IO response.

Balanced realization, balanced truncation

1. Find a state-space realization where **direction** in which the system is easily controlled are **aligned** with the directions in which the states are easily observable. Aligning the principal axes of the reachability and observability ellipsoids. **Ballanced realization**.
2. Remove the state variables that are weakly involved in IO response. The ellipsoids aligned with the coordinate axes.

Balanced realization, balanced truncation

1. Find a state-space realization where **direction** in which the system is easily controlled are **aligned** with the directions in which the states are easily observable. Aligning the principal axes of the reachability and observability ellipsoids. **Ballanced realization**.
2. Remove the state variables that are weakly involved in IO response. The ellipsoids aligned with the coordinate axes. **Balanced truncation**, balanced residualization.

$\bar{x} = Tx$

$$\bar{\mathsf{x}} = \mathsf{T}\mathsf{x} \quad \Rightarrow \quad \mathsf{x} = \mathsf{T}^{-1}\bar{\mathsf{x}},$$

$$\bar{x} = Tx \quad \Rightarrow \quad x = T^{-1}\bar{x}, \quad A = T^{-1}\bar{A}T,\ B = T^{-1}\bar{B},\ C = \bar{C}T,\ D = \bar{D}$$

$$\bar{x} = Tx \quad \Rightarrow \quad x = T^{-1}\bar{x}, \quad A = T^{-1}\bar{A}T,\ B = T^{-1}\bar{B},\ C = \bar{C}T,\ D = \bar{D}$$

$$\bar{P} = TPT^{*}, \quad \bar{Q} = \left(T^{-1}\right)^{*} QT^{-1},$$

$$\bar{\mathsf{x}} = \mathsf{T}\mathsf{x} \quad \Rightarrow \quad \mathsf{x} = \mathsf{T}^{-1}\bar{\mathsf{x}}, \quad \mathsf{A} = \mathsf{T}^{-1}\bar{\mathsf{A}}\mathsf{T},\ \mathsf{B} = \mathsf{T}^{-1}\bar{\mathsf{B}},\ \mathsf{C} = \bar{\mathsf{C}}\mathsf{T},\ \mathsf{D} = \bar{\mathsf{D}}$$

$$\bar{\mathsf{P}} = \mathsf{T}\mathsf{P}\mathsf{T}^*, \quad \bar{\mathsf{Q}} = \left(\mathsf{T}^{-1}\right)^* \mathsf{Q}\mathsf{T}^{-1}, \quad \bar{\mathsf{P}}\bar{\mathsf{Q}} = \mathsf{T}\mathsf{P}\mathsf{Q}\mathsf{T}^{-1}$$

$$\bar{x} = Tx \quad \Rightarrow \quad x = T^{-1}\bar{x}, \quad A = T^{-1}\bar{A}T,\ B = T^{-1}\bar{B},\ C = \bar{C}T,\ D = \bar{D}$$

$$\bar{P} = TPT^*, \quad \bar{Q} = \left(T^{-1}\right)^* QT^{-1}, \quad \bar{P}\bar{Q} = TPQT^{-1}$$

Find **balancing** transformation T, $|T| \neq 0$ such that

$$\boxed{\bar{P} = \bar{Q}}$$

$$\bar{x} = Tx \quad \Rightarrow \quad x = T^{-1}\bar{x}, \quad A = T^{-1}\bar{A}T,\ B = T^{-1}\bar{B},\ C = \bar{C}T,\ D = \bar{D}$$

$$\bar{P} = TPT^*, \quad \bar{Q} = \left(T^{-1}\right)^* QT^{-1}, \quad \bar{P}\bar{Q} = TPQT^{-1}$$

Find **balancing** transformation T, $|T| \neq 0$ such that

$$\boxed{\bar{P} = \bar{Q}}$$

One easy possibility is **diagonal balancing**

$$\bar{P} = \bar{Q} = \mathrm{diag}(\ \underbrace{\sigma_1, \sigma_2, \ldots, \sigma_n}_{\text{Hankel singular values}}\ )$$

$$\bar{\mathsf{x}} = \mathsf{T}\mathsf{x} \quad \Rightarrow \quad \mathsf{x} = \mathsf{T}^{-1}\bar{\mathsf{x}}, \quad \mathsf{A} = \mathsf{T}^{-1}\bar{\mathsf{A}}\mathsf{T},\ \mathsf{B} = \mathsf{T}^{-1}\bar{\mathsf{B}},\ \mathsf{C} = \bar{\mathsf{C}}\mathsf{T},\ \mathsf{D} = \bar{\mathsf{D}}$$

$$\bar{\mathsf{P}} = \mathsf{T}\mathsf{P}\mathsf{T}^*, \quad \bar{\mathsf{Q}} = \left(\mathsf{T}^{-1}\right)^* \mathsf{Q}\mathsf{T}^{-1}, \quad \bar{\mathsf{P}}\bar{\mathsf{Q}} = \mathsf{T}\mathsf{P}\mathsf{Q}\mathsf{T}^{-1}$$

Find **balancing** transformation $\mathsf{T},\ |\mathsf{T}| \neq 0$ such that

$$\boxed{\bar{\mathsf{P}} = \bar{\mathsf{Q}}}$$

One easy possibility is **diagonal balancing**

$$\bar{\mathsf{P}} = \bar{\mathsf{Q}} = \mathrm{diag}(\underbrace{\sigma_1, \sigma_2, \ldots, \sigma_n}_{\text{Hankel singular values}})$$

Matlab: finding Hankel singular values with **hsvd()** (Control System Tbx) or **hankelsv()** (Robust Control Tbx)

Algorithm for balancing transformation

Two key computations:

Algorithm for balancing transformation

Two key computations:

Cholesky decomposition of P

$$\boxed{P = LL^*}$$

Algorithm for balancing transformation

Two key computations:

Cholesky decomposition of P

$$\boxed{P = LL^*}$$

Eigenvalue decomposition of L^*QL

$$\boxed{L^*QL = K\Sigma^2K^*}$$

Two key computations:

Cholesky decomposition of P

$$\boxed{P = LL^*}$$

Eigenvalue decomposition of L^*QL

$$\boxed{L^*QL = K\Sigma^2K^*}$$

With these, the product of the two gramians is

$$PQ = LL^*Q\underbrace{LL^{-1}}_{I}$$

Two key computations:

Cholesky decomposition of P

$$\boxed{P = LL^*}$$

Eigenvalue decomposition of L^*QL

$$\boxed{L^*QL = K\Sigma^2 K^*}$$

With these, the product of the two gramians is

$$PQ = LL^*Q\underbrace{LL^{-1}}_{I} = L(L^*QL)L^{-1}$$

Two key computations:

Cholesky decomposition of P

$$\boxed{P = LL^*}$$

Eigenvalue decomposition of L^*QL

$$\boxed{L^*QL = K\Sigma^2K^*}$$

With these, the product of the two gramians is

$$PQ = LL^*Q\underbrace{LL^{-1}}_{I} = L(L^*QL)L^{-1} = LK\Sigma^2K^*L^{-1}$$

Two key computations:

Cholesky decomposition of P

$$\boxed{P = LL^*}$$

Eigenvalue decomposition of L^*QL

$$\boxed{L^*QL = K\Sigma^2 K^*}$$

With these, the product of the two gramians is

$$PQ = LL^*Q\underbrace{LL^{-1}}_{I} = L(L^*QL)L^{-1} = LK\Sigma^2K^*L^{-1}$$

Diagonalizing similarity transformation $TPQT^{-1} = \bar{P}\bar{Q} = \Sigma^2$ but also $\bar{P} = \Sigma$ and $\bar{Q} = \Sigma$, is

$$\boxed{T = \Sigma^{1/2}K^*L^{-1}, \quad T^{-1} = LK\Sigma^{-1/2}}$$

Algorithm for balancing transformation

Two key computations:

Cholesky decomposition of P

$$\boxed{P = LL^*}$$

Eigenvalue decomposition of L^*QL

$$\boxed{L^*QL = K\Sigma^2K^*}$$

With these, the product of the two gramians is

$$PQ = LL^*Q\underbrace{LL^{-1}}_{I} = L(L^*QL)L^{-1} = LK\Sigma^2K^*L^{-1}$$

Diagonalizing similarity transformation $TPQT^{-1} = \bar{P}\bar{Q} = \Sigma^2$ but also $\bar{P} = \Sigma$ and $\bar{Q} = \Sigma$, is

$$\boxed{T = \Sigma^{1/2}K^*L^{-1}, \quad T^{-1} = LK\Sigma^{-1/2}}$$

Matlab (Control System Tbx): **balreal(), balred()**

Properties of balanced and truncated model

Reduced (truncated) system given by (A_{11}, B_1, C_1, D) is

1. stable

Reduced (truncated) system given by (A_{11}, B_1, C_1, D) is

1. stable
2. the approximation error is upper-bounded

$$\boxed{\|G - G_k\|_\infty \leq 2(\sigma_{k+1} + \sigma_{k+2} + \ldots + \sigma_q)}$$

(without multiplicities!)

Properties of balanced and truncated model

Reduced (truncated) system given by (A_{11}, B_1, C_1, D) is

1. stable
2. the approximation error is upper-bounded

$$\boxed{\|G - G_k\|_\infty \leq 2(\sigma_{k+1} + \sigma_{k+2} + \ldots + \sigma_q)}$$

(without multiplicities!)

However, **not minimizing the error!**

Matlab example

Need for more efficient/reliable balancing algorithms

Need for more efficient/reliable balancing algorithms

- large-scale systems

Need for more efficient/reliable balancing algorithms

- large-scale systems
- fast decay of eigenvalues of gramians (practically very low rank even for high order systems)

Need for more efficient/reliable balancing algorithms

- large-scale systems
- fast decay of eigenvalues of gramians (practically very low rank even for high order systems)

Try avoiding inverses (ill-conditioned, inefficient) and balancing the whole system.

Need for more efficient/reliable balancing algorithms

- large-scale systems
- fast decay of eigenvalues of gramians (practically very low rank even for high order systems)

Try avoiding inverses (ill-conditioned, inefficient) and balancing the whole system.
More in

1. ANTOULAS, Athanasios C. Approximation of large-scale dynamical systems. SIAM, 2005.
2. ANTOULAS, Athanasios C.; SORENSEN, Dan C. Approximation of large-scale dynamical systems: An overview. 2001. Download at `https://scholarship.rice.edu/bitstream/handle/1911/101964/TR01-01.pdf?sequence=1`

We need three matrix decompositions

Upper triangular Cholesky decomposition of P

$$P = UU^*$$

Lower triangular Cholesky decomposition of Q

$$Q = LL^*$$

Singular value decomposition of U*L

$$U^*L = W\Sigma V^*$$

Square root balancing (and truncating)

Product of gramians can be then expressed as

$$\begin{aligned}
\mathsf{PQ} &= \mathsf{UU^*LL^*} \\
&= \mathsf{U(U^*L)(L^*U)U^{-1}} \\
&= \mathsf{U(U^*L)(V\Sigma W^*)U^{-1}}
\end{aligned}$$

Diagonalizing similarity transformation is

$$\mathsf{T} = \Sigma^{-1/2}\mathsf{V^*L^*}, \quad \mathsf{T}_i = \underbrace{\mathsf{UW}\Sigma^{-1/2}}_{\mathsf{T}^{-1}}$$

Finding only the relevant part of the transformation matrix

$$\mathsf{T}_1 = \Sigma_1^{-1/2}\mathsf{V}_1^*\mathsf{L}^*, \quad \mathsf{T}_{1i} = \mathsf{UW}_1\Sigma_1^{-1/2}$$

Truncated system given by $(\mathsf{T}_1\mathsf{AT}_{1i}, \mathsf{T}_1\mathsf{B}, \mathsf{CT}_{1i}, \mathsf{D})$.
Matlab (Robust Control Tbx): **balancmr()**

Square root balancing up to scaling

Simply omit the scaling $\Sigma^{-1/2}$. Can improve conditioning. Scale the system instead.

$$T = V^* L^*, \quad T_i = UW$$

$$\left[\begin{array}{c|c} \Sigma^{-1/2} A \Sigma^{-1/2} & \Sigma^{-1/2} B \\ \hline C \Sigma^{-1/2} & D \end{array}\right]$$

Two extra matrix decompositions needed

QR decomposition of UW

$$\mathsf{UW} = \mathsf{X}\Phi$$

QR decomposition of LV

$$\mathsf{LV} = \mathsf{Y}\Psi$$

where X, Y are orthogonal, Φ, Ψ are upper triangular.

Balancing free square root algorithm (A.Varga, 1991)

Transformation diagonalizing up to a triangular transformation is given by

$$\mathsf{T} = (\mathsf{Y}^*\mathsf{X})^{-1}\mathsf{Y}^* = \mathsf{X}^*, \quad \mathsf{T}^{-1} = \mathsf{X}$$

Then

$$\left[\begin{array}{c|c} \mathsf{X}^*\mathsf{A}\mathsf{X} & \mathsf{X}^*\mathsf{B} \\ \hline \mathsf{C}\mathsf{X} & \mathsf{D} \end{array}\right]$$

is balanced up to an upper triangular matrix $\mathsf{K} = \Sigma^{1/2}\Phi$.
Next truncate QR factorizations of UW and LW

$$\mathsf{U}\mathsf{W}_1 = \mathsf{X}_k\Phi_k, \quad \mathsf{L}\mathsf{V}_1 = Y_k\Psi_k$$

Then

$$\mathsf{T}_1 = (\mathsf{Y}_k^*\mathsf{X}_k)^{-1}\mathsf{Y}_k^*, \quad \mathsf{T}_{1i} = \mathsf{X}_k$$

transforms $(1,1)$ subsystem to

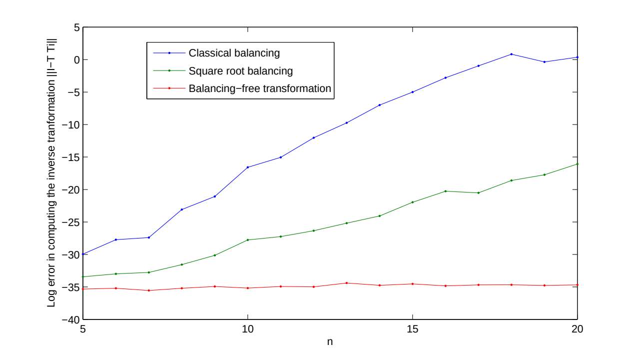
$$\left[\begin{array}{c|c} \mathsf{T}_1\mathsf{A}\mathsf{T}_{1i} & \mathsf{T}_1\mathsf{B} \\ \hline \mathsf{C}\mathsf{T}_{1i} & \mathsf{D} \end{array}\right]$$

Matlab: **balred()**

Numerical comparison of various ways of balancing

```
[A,B,C,D] = butter(n,1,'s');
```

The three algorithms for computing similarity transformation are compared w.r.t. $\|I - TT_i\|_2$



Nondiagonalizing (triangularizing) similarity transf. of PQ

Nondiagonalizing (triangularizing) similarity transf. of PQ

Schur decomposition of PQ in ascending order

$$\mathsf{V}_A^*\mathsf{PQV}_A = \mathsf{S}_A$$

Nondiagonalizing (triangularizing) similarity transf. of PQ

Schur decomposition of PQ in ascending order

$$\mathsf{V}_A^* \mathsf{P}\mathsf{Q}\mathsf{V}_A = \mathsf{S}_A$$

Schur decomposition of PQ in descending order

$$\mathsf{V}_D^* \mathsf{P}\mathsf{Q}\mathsf{V}_D = \mathsf{S}_D$$

Nondiagonalizing (triangularizing) similarity transf. of PQ

Schur decomposition of PQ in ascending order

$$\mathsf{V}_A^* \mathsf{PQV}_A = \mathsf{S}_A$$

Schur decomposition of PQ in descending order

$$\mathsf{V}_D^* \mathsf{PQV}_D = \mathsf{S}_D$$

Keep the columns of V_A and V_D corresponding to the k large eigs

$$\mathsf{V}_a = \mathsf{V}_A \begin{bmatrix} 0 \\ \mathsf{I}_k \end{bmatrix}, \quad \mathsf{V}_d = \mathsf{V}_D \begin{bmatrix} \mathsf{I}_k \\ 0 \end{bmatrix}$$

Nondiagonalizing (triangularizing) similarity transf. of PQ

Schur decomposition of PQ in ascending order

$$\mathsf{V}_A^*\mathsf{PQV}_A = \mathsf{S}_A$$

Schur decomposition of PQ in descending order

$$\mathsf{V}_D^*\mathsf{PQV}_D = \mathsf{S}_D$$

Keep the columns of V_A and V_D corresponding to the k large eigs

$$\mathsf{V}_a = \mathsf{V}_A \begin{bmatrix} 0 \\ \mathsf{I}_k \end{bmatrix}, \quad \mathsf{V}_d = \mathsf{V}_D \begin{bmatrix} \mathsf{I}_k \\ 0 \end{bmatrix}$$

Singular value decomposition of $\mathsf{V}_a^*\mathsf{V}_d$

$$\mathsf{V}_a^*\mathsf{V}_d = \mathsf{U}_L\mathsf{S}\mathsf{U}_R^*$$

Nondiagonalizing (triangularizing) similarity transf. of PQ

Schur decomposition of PQ in ascending order

$$\mathsf{V}_A^* \mathsf{PQV}_A = \mathsf{S}_A$$

Schur decomposition of PQ in descending order

$$\mathsf{V}_D^* \mathsf{PQV}_D = \mathsf{S}_D$$

Keep the columns of V_A and V_D corresponding to the k large eigs

$$\mathsf{V}_a = \mathsf{V}_A \begin{bmatrix} 0 \\ \mathsf{I}_k \end{bmatrix}, \quad \mathsf{V}_d = \mathsf{V}_D \begin{bmatrix} \mathsf{I}_k \\ 0 \end{bmatrix}$$

Singular value decomposition of $\mathsf{V}_a^* \mathsf{V}_d$

$$\mathsf{V}_a^* \mathsf{V}_d = \mathsf{U}_L \mathsf{S} \mathsf{U}_R^*$$

The transformation matrices are

$$\mathsf{T} = \mathsf{S}^{-1/2} \mathsf{U}_L^* \mathsf{V}_a^*, \quad \mathsf{T}_i = \mathsf{V}_d \mathsf{U}_R \mathsf{S}^{-1/2}$$

Nondiagonalizing (triangularizing) similarity transf. of PQ

Schur decomposition of PQ in ascending order

$$\mathsf{V}_A^* \mathsf{P}\mathsf{Q}\mathsf{V}_A = \mathsf{S}_A$$

Schur decomposition of PQ in descending order

$$\mathsf{V}_D^* \mathsf{P}\mathsf{Q}\mathsf{V}_D = \mathsf{S}_D$$

Keep the columns of V_A and V_D corresponding to the k large eigs

$$\mathsf{V}_a = \mathsf{V}_A \begin{bmatrix} 0 \\ \mathsf{I}_k \end{bmatrix}, \quad \mathsf{V}_d = \mathsf{V}_D \begin{bmatrix} \mathsf{I}_k \\ 0 \end{bmatrix}$$

Singular value decomposition of $\mathsf{V}_a^* \mathsf{V}_d$

$$\mathsf{V}_a^* \mathsf{V}_d = \mathsf{U}_L \mathsf{S} \mathsf{U}_R^*$$

The transformation matrices are

$$\mathsf{T} = \mathsf{S}^{-1/2} \mathsf{U}_L^* \mathsf{V}_a^*, \quad \mathsf{T}_i = \mathsf{V}_d \mathsf{U}_R \mathsf{S}^{-1/2}$$

Matlab: **schurmr()**

Different objects can be used for balancing. Instead of solution to Lyapunov eqs, solutions to some other equations can be used: Riccati equations?

Assumptions

- square systems
- stable
- D nonsingular

System transfer function

$$\mathrm{H}(s) = \mathrm{C}(s\mathrm{I} - \mathrm{A})^{-1}\mathrm{B} + \mathrm{D}$$

Reachability gramian P

$$AP + PA^T + BB^T = 0$$

Reachability gramian P

$$\mathsf{AP} + \mathsf{PA}^\mathsf{T} + \mathsf{BB}^\mathsf{T} = 0$$

Define $\bar{\mathsf{B}} = \mathsf{PC}^\mathsf{T} + \mathsf{BD}^\mathsf{T}$

Reachability gramian P

$$\mathsf{AP} + \mathsf{PA}^\mathsf{T} + \mathsf{BB}^\mathsf{T} = 0$$

Define $\bar{\mathsf{B}} = \mathsf{PC}^\mathsf{T} + \mathsf{BD}^\mathsf{T}$ and find stabilizing solution Q

$$\mathsf{A}^\mathsf{T}\mathsf{Q} + \mathsf{QA} + (\mathsf{C} - \bar{\mathsf{B}}^\mathsf{T}\mathsf{Q})^\mathsf{T}(\mathsf{DD}^\mathsf{T})^{-1}(\mathsf{C} - \bar{\mathsf{B}}^\mathsf{T}\mathsf{Q}) = 0$$

Reachability gramian P

$$AP + PA^T + BB^T = 0$$

Define $\bar{B} = PC^T + BD^T$ and find stabilizing solution Q

$$A^TQ + QA + (C - \bar{B}^TQ)^T(DD^T)^{-1}(C - \bar{B}^TQ) = 0$$

Finding a diagonalizing transformation for PQ and applying to the original system.

Balanced stochastic truncation

Reachability gramian P

$$\mathsf{AP} + \mathsf{PA}^\mathsf{T} + \mathsf{BB}^\mathsf{T} = 0$$

Define $\bar{\mathsf{B}} = \mathsf{PC}^\mathsf{T} + \mathsf{BD}^\mathsf{T}$ and find stabilizing solution Q

$$\mathsf{A}^\mathsf{T}\mathsf{Q} + \mathsf{QA} + (\mathsf{C} - \bar{\mathsf{B}}^\mathsf{T}\mathsf{Q})^\mathsf{T}(\mathsf{DD}^\mathsf{T})^{-1}(\mathsf{C} - \bar{\mathsf{B}}^\mathsf{T}\mathsf{Q}) = 0$$

Finding a diagonalizing transformation for PQ and applying to the original system.
Stochastic balancing guarantees multiplicative error bounds:

$$\sigma_{k+1} \leq \|\mathsf{H}^{-1}(\mathsf{H} - \mathsf{H}_k)\|_\infty \leq \prod_{i=k+1}^{n} \frac{1+\sigma_i}{1-\sigma_i} - 1$$

Reachability gramian P

$$\mathsf{AP} + \mathsf{PA}^\mathsf{T} + \mathsf{BB}^\mathsf{T} = 0$$

Define $\bar{\mathsf{B}} = \mathsf{PC}^\mathsf{T} + \mathsf{BD}^\mathsf{T}$ and find stabilizing solution Q

$$\mathsf{A}^\mathsf{T}\mathsf{Q} + \mathsf{QA} + (\mathsf{C} - \bar{\mathsf{B}}^\mathsf{T}\mathsf{Q})^\mathsf{T}(\mathsf{DD}^\mathsf{T})^{-1}(\mathsf{C} - \bar{\mathsf{B}}^\mathsf{T}\mathsf{Q}) = 0$$

Finding a diagonalizing transformation for PQ and applying to the original system.
Stochastic balancing guarantees multiplicative error bounds:

$$\sigma_{k+1} \leq \|\mathsf{H}^{-1}(\mathsf{H} - \mathsf{H}_k)\|_\infty \leq \prod_{i=k+1}^{n} \frac{1+\sigma_i}{1-\sigma_i} - 1$$

Matlab: **bstmr()**

So far, no **minimization** of the error. Only the existence of upper bounds.

So far, no **minimization** of the error. Only the existence of upper bounds. Minimization of $\mathcal{H}_2$ and $\mathcal{H}_\infty$ norms of errors overly difficult (nonconvex optimization).

So far, no **minimization** of the error. Only the existence of upper bounds. Minimization of $\mathcal{H}_2$ and $\mathcal{H}_\infty$ norms of errors overly difficult (nonconvex optimization). However, it is feasible to minimize Hankel norm.

So far, no **minimization** of the error. Only the existence of upper bounds. Minimization of $\mathcal{H}_2$ and $\mathcal{H}_\infty$ norms of errors overly difficult (nonconvex optimization). However, it is feasible to minimize Hankel norm.
Given a system G find a reduced-order G_r minizing the error as in

$$\min_{\mathsf{G}_r} \|\mathsf{G} - \mathsf{G}_r\|_H$$

Hankel operator Γ_G (for a given system G) maps past inputs into future outputs

$$\mathrm{y}(t) = \int_{-\infty}^{0} \underbrace{\mathrm{C}e^{\mathrm{A}(t-\tau)}\mathrm{B}}_{\mathrm{h}(t-\tau)} \mathrm{u}(\tau)d\tau, \quad t > 0$$

Hankel operator, Hankel norm

Hankel operator Γ_G (for a given system G) maps past inputs into future outputs

$$y(t) = \int_{-\infty}^{0} \underbrace{Ce^{A(t-\tau)}B}_{h(t-\tau)} u(\tau) d\tau, \quad t > 0$$

$\mathcal{L}_2$-induced norm of Hankel operator is

$$\|G\|_H = \sup_{u \in \mathcal{L}(-\infty,0]} \frac{\|y\|_{\mathcal{L}[0,\infty)}}{\|u\|_{\mathcal{L}(-\infty,0]}}$$

Hankel operator Γ_G (for a given system G) maps past inputs into future outputs

$$y(t) = \int_{-\infty}^{0} \underbrace{Ce^{A(t-\tau)}B}_{h(t-\tau)} u(\tau)d\tau, \quad t > 0$$

$\mathcal{L}_2$-induced norm of Hankel operator is

$$\|G\|_H = \sup_{u \in \mathcal{L}(-\infty,0]} \frac{\|y\|_{\mathcal{L}[0,\infty)}}{\|u\|_{\mathcal{L}(-\infty,0]}}$$

Apparently,

$$\|G\|_H \leq \|G\|_\infty$$

Hankel operator Γ_G (for a given system G) maps past inputs into future outputs

$$\mathrm{y}(t) = \int_{-\infty}^{0} \underbrace{\mathrm{C}e^{\mathrm{A}(t-\tau)}\mathrm{B}}_{\mathrm{h}(t-\tau)} \mathrm{u}(\tau) d\tau, \quad t > 0$$

$\mathcal{L}_2$-induced norm of Hankel operator is

$$\|\mathrm{G}\|_H = \sup_{\mathrm{u} \in \mathcal{L}(-\infty,0]} \frac{\|\mathrm{y}\|_{\mathcal{L}[0,\infty)}}{\|\mathrm{u}\|_{\mathcal{L}(-\infty,0]}}$$

Apparently,

$$\|\mathrm{G}\|_H \leq \|\mathrm{G}\|_\infty$$

$$\boxed{\|\mathrm{G}\|_H = \sqrt{\rho(\mathrm{QP})}}$$

Hankel operator Γ_G (for a given system G) maps past inputs into future outputs

$$y(t) = \int_{-\infty}^{0} \underbrace{C e^{A(t-\tau)} B}_{h(t-\tau)} u(\tau) d\tau, \quad t > 0$$

$\mathcal{L}_2$-induced norm of Hankel operator is

$$\|G\|_H = \sup_{u \in \mathcal{L}(-\infty,0]} \frac{\|y\|_{\mathcal{L}[0,\infty)}}{\|u\|_{\mathcal{L}(-\infty,0]}}$$

Apparently,

$$\|G\|_H \leq \|G\|_\infty$$

$$\boxed{\|G\|_H = \sqrt{\rho(QP)}}$$

Matlab: **hankelmr()**

Different focus at different frequencies. Find a reduced-order system such that

$$\|W_o(G - G_r)W_i\|_\infty$$

is small (either minimum possible or at least upper-bounded).

Different focus at different frequencies. Find a reduced-order system such that

$$\|W_o(G - G_r)W_i\|_\infty$$

is small (either minimum possible or at least upper-bounded).
For two-sided case: no bound, no stability guarantees (refutation of Enn's conjecture)

Make

$$\|(G - G_r)V\|_\infty$$

small or at least guaranteed.
With the state space models of G and V

$$G = \left[\begin{array}{c|c} A & B \\ \hline C & 0 \end{array}\right], \quad V = \left[\begin{array}{c|c} A_v & B_v \\ \hline C_v & D_v \end{array}\right]$$

Make

$$\|(G - G_r)V\|_\infty$$

small or at least guaranteed.
With the state space models of G and V

$$G = \left[\begin{array}{c|c} A & B \\ \hline C & 0 \end{array}\right], \quad V = \left[\begin{array}{c|c} A_v & B_v \\ \hline C_v & D_v \end{array}\right]$$

The state-space realization of GV is

$$GV = \left[\begin{array}{cc|c} A & BC_v & BD_v \\ 0 & A_v & B_v \\ \hline C & 0 & 0 \end{array}\right] = \left[\begin{array}{c|c} \bar{A} & \bar{B} \\ \hline \bar{C} & \bar{D} \end{array}\right]$$

(New) controllability gramian

$$\bar{P}\bar{A}^{T} + \bar{A}\bar{P} + \bar{B}\bar{B}^{T} = 0$$

(New) controllability gramian

$$\bar{P}\bar{A}^T + \bar{A}\bar{P} + \bar{B}\bar{B}^T = 0$$

Observability gramian stays intact

$$QA + A^TQ + C^TC = 0$$

(New) controllability gramian

$$\bar{P}\bar{A}^T + \bar{A}\bar{P} + \bar{B}\bar{B}^T = 0$$

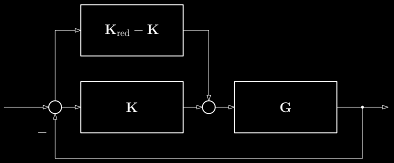
Observability gramian stays intact

$$QA + A^TQ + C^TC = 0$$

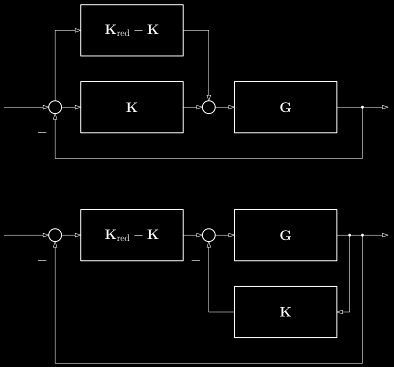
Then... Balancing the product PQ... Stability guaranteed. Some ugly upper bounds available...

Replace the original full-order K with the reduced-order K_{red}

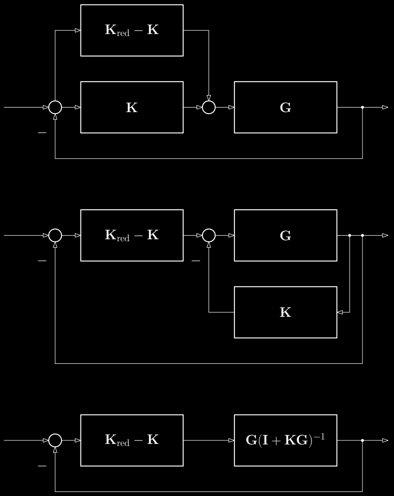
Replace the original full-order K with the reduced-order K_{red}



Replace the original full-order K with the reduced-order K_{red}



Replace the original full-order K with the reduced-order K_{red}



Sufficient conditions for K_{red} to be stabilizing

- K and K_{red} have the same number of unstable poles and $K - K_{\mathrm{red}}$ is bounded on the imaginary axis

Sufficient conditions for $\mathsf{K}_{\mathrm{red}}$ to be stabilizing

- K and $\mathsf{K}_{\mathrm{red}}$ have the same number of unstable poles and $\mathsf{K} - \mathsf{K}_{\mathrm{red}}$ is bounded on the imaginary axis
- Either

$$\|(\mathsf{K} - \mathsf{K}_{\mathrm{red}})\mathsf{G}(\mathsf{I} + \mathsf{KG})^{-1}\|_\infty < 1$$

Sufficient conditions for $\mathsf{K}_{\rm red}$ to be stabilizing

- K and $\mathsf{K}_{\rm red}$ have the same number of unstable poles and $\mathsf{K} - \mathsf{K}_{\rm red}$ is bounded on the imaginary axis
- Either

$$\|(\mathsf{K} - \mathsf{K}_{\rm red})\mathsf{G}(\mathsf{I} + \mathsf{KG})^{-1}\|_\infty < 1$$

or

$$\|(\mathsf{I} + \mathsf{GK})^{-1}\mathsf{G}(\mathsf{K} - \mathsf{K}_{\rm red})\|_\infty < 1$$

Sufficient conditions for $\mathsf{K}_{\mathrm{red}}$ to be stabilizing

- K and $\mathsf{K}_{\mathrm{red}}$ have the same number of unstable poles and $\mathsf{K} - \mathsf{K}_{\mathrm{red}}$ is bounded on the imaginary axis
- Either
$$\|(\mathsf{K} - \mathsf{K}_{\mathrm{red}})\mathsf{G}(\mathsf{I} + \mathsf{KG})^{-1}\|_\infty < 1$$
or
$$\|(\mathsf{I} + \mathsf{GK})^{-1}\mathsf{G}(\mathsf{K} - \mathsf{K}_{\mathrm{red}})\|_\infty < 1$$

Hence *frequency weighting matrix*

$$\mathsf{V}(s) = \mathsf{G}(\mathsf{I} + \mathsf{KG})^{-1} = (\mathsf{I} + \mathsf{GK})^{-1}\mathsf{G}$$