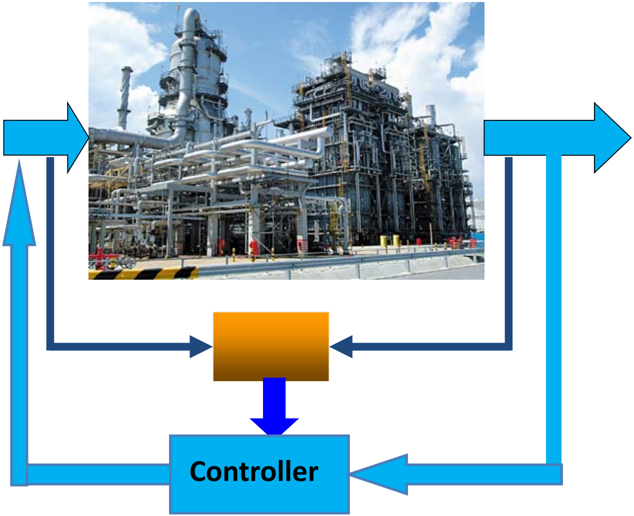
ACTIVE APPLICATION ORIENTED LEARNING OF COMPLEX DYNAMICAL SYSTEMS WITH APPLICATION TO MPC

Håkan Hjalmarsson

ACCESS Linnaeus Center
AdBIOPRO - Center for Advanced Bioproduction
School of Electrical Engineering and Computer Science
KTH - Royal Insititute of Technology, Stockholm

DDCLS'18
May 26, 2018

The problem



Ogunnaike 1996:

... obtaining the process model is the single most time consuming task in the application of model based control

Ogunnaike 1996:

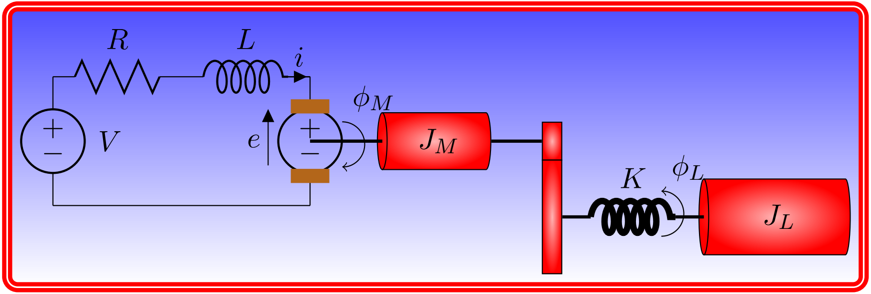
... obtaining the process model is the single most time consuming task in the application of model based control

I'm afraid this still describes state-of-the art....

Outline

Outline

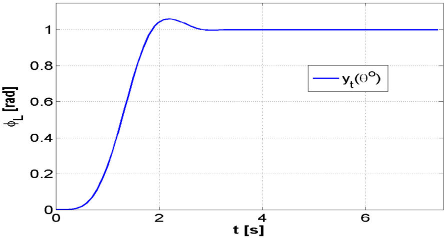
An application example: MPC of a DC-motor



- Input: Voltage V
- Output: Angle ϕ_L
- Model parameters θ: Resistance R, Moment of inertia J_L, Elasticity K, ...
- True parameters: θ_o

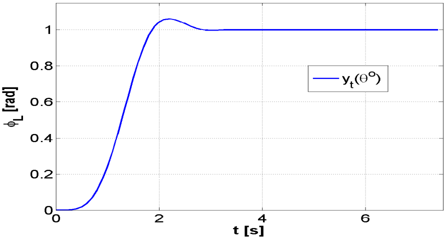
An application example: MPC of a DC-motor

- Ideal response: $y_t(\theta_o)$ - true parameters used in MPC



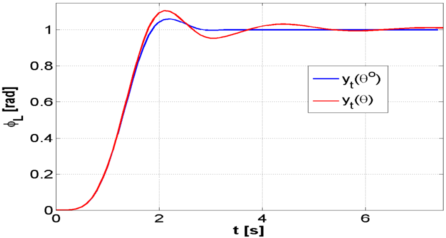
An application example: MPC of a DC-motor

- Ideal response: $y_t(\theta_o)$ - true parameters used in MPC
- Actual response: $y_t(\theta)$ – parameter θ used in MPC



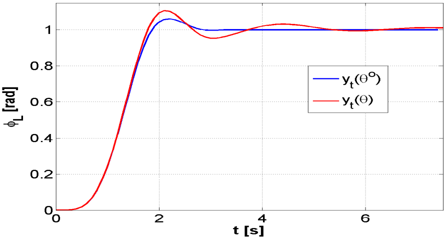
An application example: MPC of a DC-motor

- Ideal response: $y_t(\theta_o)$ - true parameters used in MPC
- Actual response: $y_t(\theta)$ – parameter θ used in MPC



An application example: MPC of a DC-motor

- Ideal response: $y_t(\theta_o)$ - true parameters used in MPC
- Actual response: $y_t(\theta)$ – parameter θ used in MPC

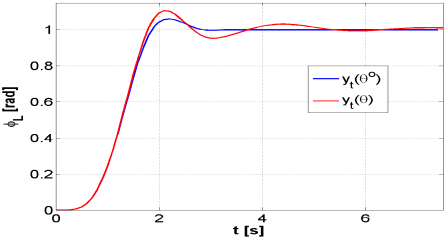


Performance degradation /Set of acceptable models

$$V_{\text{app}}(\theta) = \frac{1}{N}\sum_{t=1}^{N}(y_t(\theta_o) - y_t(\theta))^2$$

An application example: MPC of a DC-motor

- Ideal response: $y_t(\theta_o)$ - true parameters used in MPC
- Actual response: $y_t(\theta)$ – parameter θ used in MPC



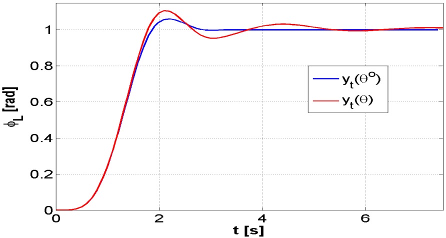
Performance degradation /Set of acceptable models

$$V_{\text{app}}(\theta) = \frac{1}{N}\sum_{t=1}^{N}(y_t(\theta_o) - y_t(\theta))^2$$

$$\mathcal{E}_{\text{app}} = \left\{\theta : V_{\text{app}}(\theta) \leq \frac{1}{\gamma}\right\} \qquad (\gamma = \text{accuracy})$$

An application example: MPC of a DC-motor

- Ideal response: $y_t(\theta_o)$ - true parameters used in MPC
- Actual response: $y_t(\theta)$ – parameter θ used in MPC

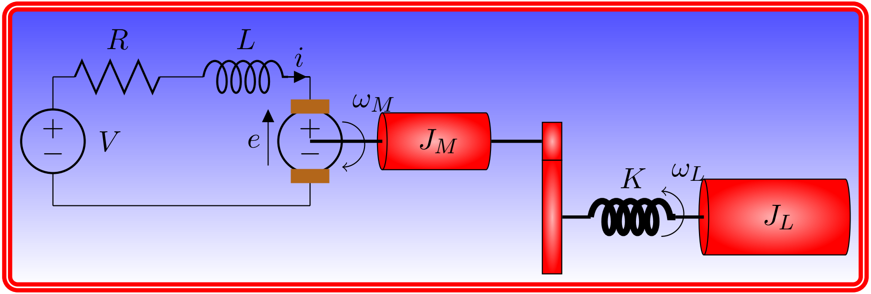


Performance degradation /Set of acceptable models

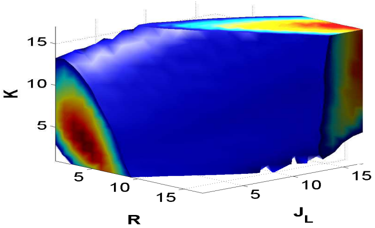
$$V_{\text{app}}(\theta) = \frac{1}{N}\sum_{t=1}^{N}(y_t(\theta_o) - y_t(\theta))^2$$

$$\mathcal{E}_{\text{app}} = \left\{\theta : V_{\text{app}}(\theta) \leq \frac{1}{\gamma}\right\} \qquad (\gamma = \text{accuracy})$$

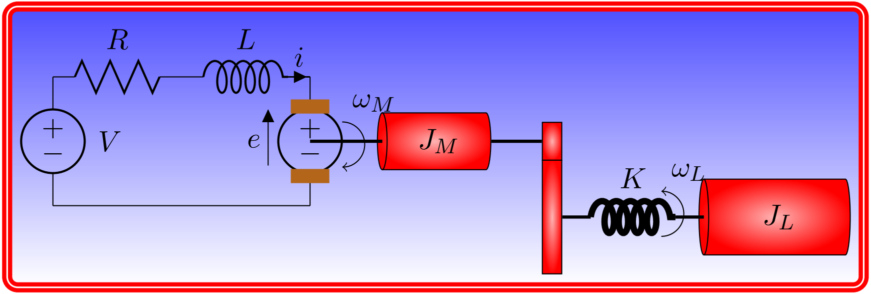
An application example: MPC of a DC-motor



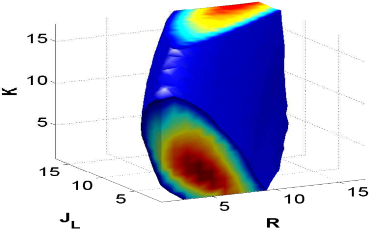
Set of acceptable models $\mathcal{E}_{\text{app}}$:



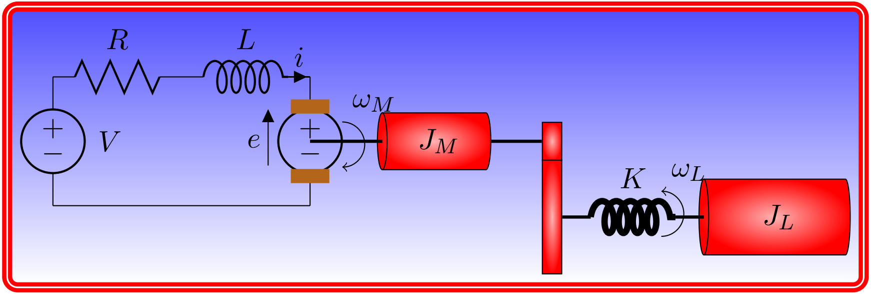
An application example: MPC of a DC-motor



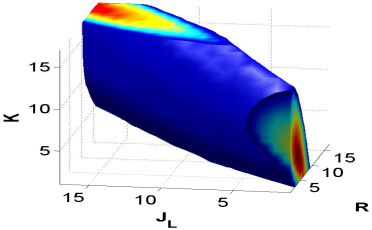
Set of acceptable models $\mathcal{E}_{\text{app}}$:



An application example: MPC of a DC-motor



Set of acceptable models $\mathcal{E}_{\text{app}}$:



Summary of concepts

- Performance degradation for application: $V_{\mathsf{app}}(\theta)$

Summary of concepts

- Performance degradation for application: $V_{\mathsf{app}}(\theta)$
- Set of acceptable models: $\mathcal{E}_{\mathsf{app}} = \left\{\theta : V_{\mathsf{app}}(\theta) \leq \frac{1}{\gamma}\right\}$

Summary of concepts

- Performance degradation for application: $V_{\mathsf{app}}(\theta)$
- Set of acceptable models: $\mathcal{E}_{\mathsf{app}} = \left\{\theta : V_{\mathsf{app}}(\theta) \leq \frac{1}{\gamma}\right\}$
- Identification: Produce $\hat{\theta}_N \in \mathcal{E}_{\mathsf{app}} \subset \mathbb{R}^n$ ($N =$ sample size)

Summary of concepts

- Performance degradation for application: $V_{\text{app}}(\theta)$
- Set of acceptable models: $\mathcal{E}_{\text{app}} = \left\{\theta : V_{\text{app}}(\theta) \leq \frac{1}{\gamma}\right\}$
- Identification: Produce $\hat{\theta}_N \in \mathcal{E}_{\text{app}} \subset \mathbb{R}^n$ (N = sample size)
- Experiment design objective:

Summary of concepts

- Performance degradation for application: $V_{\mathsf{app}}(\theta)$
- Set of acceptable models: $\mathcal{E}_{\mathsf{app}} = \left\{\theta : V_{\mathsf{app}}(\theta) \leq \frac{1}{\gamma}\right\}$
- Identification: Produce $\hat{\theta}_N \in \mathcal{E}_{\mathsf{app}} \subset \mathbb{R}^n$ ($N =$ sample size)
- Experiment design objective: *Experimental cost*

Summary of concepts

- Performance degradation for application: $V_{\text{app}}(\theta)$
- Set of acceptable models: $\mathcal{E}_{\text{app}} = \left\{\theta : V_{\text{app}}(\theta) \leq \frac{1}{\gamma}\right\}$
- Identification: Produce $\hat{\theta}_N \in \mathcal{E}_{\text{app}} \subset \mathbb{R}^n$ ($N =$ sample size)
- Experiment design objective: *Experimental cost*
- Least-costly identification

Summary of concepts

- Performance degradation for application: $V_{\text{app}}(\theta)$
- Set of acceptable models: $\mathcal{E}_{\text{app}} = \left\{\theta : V_{\text{app}}(\theta) \leq \frac{1}{\gamma}\right\}$
- Identification: Produce $\hat{\theta}_N \in \mathcal{E}_{\text{app}} \subset \mathbb{R}^n$ ($N =$ sample size)
- Experiment design objective: *Experimental cost*
- Least-costly identification
- For example: Experimental cost $=$ input energy

Summary of concepts

- Performance degradation for application: $V_{\text{app}}(\theta)$
- Set of acceptable models: $\mathcal{E}_{\text{app}} = \left\{\theta : V_{\text{app}}(\theta) \leq \frac{1}{\gamma}\right\}$
- Identification: Produce $\hat{\theta}_N \in \mathcal{E}_{\text{app}} \subset \mathbb{R}^n$ (N = sample size)
- Experiment design objective: *Experimental cost*
- Least-costly identification
- For example: Experimental cost = input energy

Application oriented experiment design

$$\min N\mathrm{E}[u_t^2]$$
$$\text{s.t. } \hat{\theta}_N \in \mathcal{E}_{\text{app}} \subset \mathbb{R}^n$$

An optimal experiment design problem

Summary of concepts

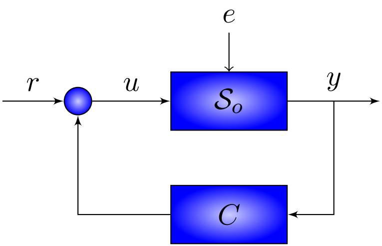
- Performance degradation for application: $V_{\text{app}}(\theta)$
- Set of acceptable models: $\mathcal{E}_{\text{app}} = \left\{\theta : V_{\text{app}}(\theta) \leq \frac{1}{\gamma}\right\}$
- Identification: Produce $\hat{\theta}_N \in \mathcal{E}_{\text{app}} \subset \mathbb{R}^n$ (N = sample size)
- Experiment design objective: *Experimental cost*
- Least-costly identification
- For example: Experimental cost = input energy

Application oriented experiment design

$$\min N\mathrm{E}[u_t^2]$$
$$\text{s.t. } \hat{\theta}_N \in \mathcal{E}_{\text{app}} \subset \mathbb{R}^n$$

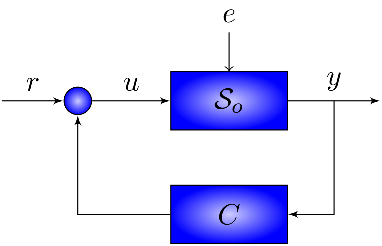
An optimal experiment design problem

System identification



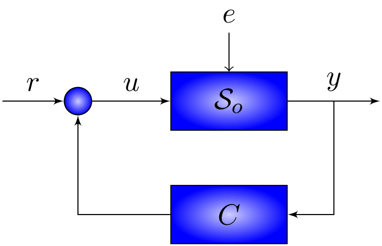
- Random innovations with variance λ_e

System identification



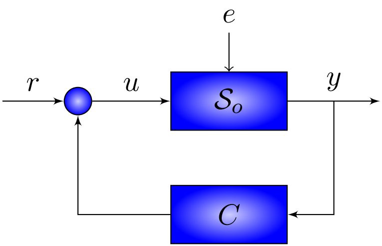
- Random innovations with variance λ_e
- Stationary signals

System identification



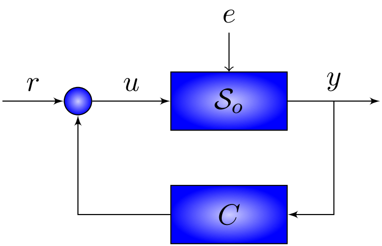
- Random innovations with variance λ_e
- Stationary signals
- True system in the model set: $\mathcal{S}_o \Leftrightarrow \theta_o$ (to be relaxed later)

System identification



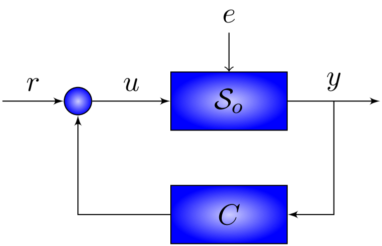
- Random innovations with variance λ_e
- Stationary signals
- True system in the model set: $\mathcal{S}_o \Leftrightarrow \theta_o$ (to be relaxed later)
- Prediction error identification:

System identification



- Random innovations with variance λ_e
- Stationary signals
- True system in the model set: $\mathcal{S}_o \Leftrightarrow \theta_o$ (to be relaxed later)
- Prediction error identification:
  - Prediction error: $\varepsilon_t(\theta) = y_t - \hat{y}_t(\theta)$

System identification



- Random innovations with variance λ_e
- Stationary signals
- True system in the model set: $\mathcal{S}_o \Leftrightarrow \theta_o$ (to be relaxed later)
- Prediction error identification:
  - Prediction error: $\varepsilon_t(\theta) = y_t - \hat{y}_t(\theta)$
  - $\hat{\theta}_N = \arg\min \sum_{t=1}^{N} \varepsilon_t^2(\theta)$

Application oriented experiment design

- $V_{\mathsf{id}}(\theta) = \mathrm{E}[\varepsilon_t^2(\theta)] - \lambda_e$

Application oriented experiment design

- $V_{\mathsf{id}}(\theta) = \mathrm{E}[\varepsilon_t^2(\theta)] - \lambda_e$
- High accuracy γ (implies large sample size N)

Application oriented experiment design

- $V_{\mathsf{id}}(\theta) = \mathrm{E}[\varepsilon_t^2(\theta)] - \lambda_e$
- High accuracy γ (implies large sample size N)
- $\sqrt{N}\left(\hat{\theta}_N - \theta_o\right) \sim \mathrm{As}\mathcal{N}\left(0, 2\lambda_e V_{\mathsf{id}}''(\theta_o)^{-1}\right)$

Application oriented experiment design

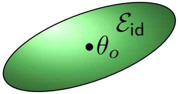
- $V_{\mathsf{id}}(\theta) = \mathrm{E}[\varepsilon_t^2(\theta)] - \lambda_e$
- High accuracy γ (implies large sample size N)
- $\sqrt{N}\left(\hat{\theta}_N - \theta_o\right) \sim \mathrm{As}\mathcal{N}\left(0, 2\lambda_e V_{\mathsf{id}}''(\theta_o)^{-1}\right)$
- Cannot guarantee $\hat{\theta}_N \in \mathcal{E}_{\mathsf{app}}$

Application oriented experiment design

- $V_{\mathsf{id}}(\theta) = \mathrm{E}[\varepsilon_t^2(\theta)] - \lambda_e$
- High accuracy γ (implies large sample size N)
- $\sqrt{N}\left(\hat{\theta}_N - \theta_o\right) \sim \mathrm{As}\mathcal{N}\left(0, 2\lambda_e V_{\mathsf{id}}''(\theta_o)^{-1}\right)$
- Cannot guarantee $\hat{\theta}_N \in \mathcal{E}_{\mathsf{app}}$
- Relaxation: Use standard (asymptotic) confidence ellipsoid $\mathcal{E}_{\mathsf{id}}$

Application oriented experiment design

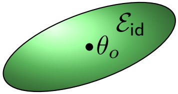
- $V_{\mathsf{id}}(\theta) = \mathrm{E}[\varepsilon_t^2(\theta)] - \lambda_e$
- High accuracy γ (implies large sample size N)
- $\sqrt{N}\left(\hat{\theta}_N - \theta_o\right) \sim \mathrm{As}\mathcal{N}\left(0, 2\lambda_e V''_{\mathsf{id}}(\theta_o)^{-1}\right)$
- Cannot guarantee $\hat{\theta}_N \in \mathcal{E}_{\mathsf{app}}$
- Relaxation: Use standard (asymptotic) confidence ellipsoid $\mathcal{E}_{\mathsf{id}}$



Application oriented experiment design

- $V_{\mathsf{id}}(\theta) = \mathrm{E}[\varepsilon_t^2(\theta)] - \lambda_e$
- High accuracy γ (implies large sample size N)
- $\sqrt{N}\left(\hat{\theta}_N - \theta_o\right) \sim \mathrm{As}\mathcal{N}\left(0, 2\lambda_e V''_{\mathsf{id}}(\theta_o)^{-1}\right)$
- Cannot guarantee $\hat{\theta}_N \in \mathcal{E}_{\mathsf{app}}$
- Relaxation: Use standard (asymptotic) confidence ellipsoid $\mathcal{E}_{\mathsf{id}}$

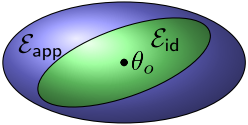
$\mathbf{Probability}(\hat{\theta}_N \in \mathcal{E}_{\mathsf{id}}) \approx \alpha$ (e.g. 99%)



Application oriented experiment design

- $V_{\mathsf{id}}(\theta) = \mathrm{E}[\varepsilon_t^2(\theta)] - \lambda_e$
- High accuracy γ (implies large sample size N)
- $\sqrt{N}\left(\hat{\theta}_N - \theta_o\right) \sim \mathrm{As}\mathcal{N}\left(0, 2\lambda_e V_{\mathsf{id}}''(\theta_o)^{-1}\right)$
- Cannot guarantee $\hat{\theta}_N \in \mathcal{E}_{\mathsf{app}}$
- Relaxation: Use standard (asymptotic) confidence ellipsoid $\mathcal{E}_{\mathsf{id}}$

$\mathbf{Probability}(\hat{\theta}_N \in \mathcal{E}_{\mathsf{id}}) \approx \alpha$ (e.g. 99%)



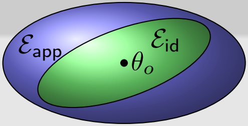
Application oriented experiment design

- $V_{\mathsf{id}}(\theta) = \mathrm{E}[\varepsilon_t^2(\theta)] - \lambda_e$
- High accuracy γ (implies large sample size N)
- $\sqrt{N}\left(\hat{\theta}_N - \theta_o\right) \sim \mathrm{As}\mathcal{N}\left(0, 2\lambda_e V''_{\mathsf{id}}(\theta_o)^{-1}\right)$
- Cannot guarantee $\hat{\theta}_N \in \mathcal{E}_{\mathsf{app}}$
- Relaxation: Use standard (asymptotic) confidence ellipsoid $\mathcal{E}_{\mathsf{id}}$

$\mathbf{Probability}(\hat{\theta}_N \in \mathcal{E}_{\mathsf{id}}) \approx \alpha$ (e.g. 99%)

Application oriented experiment design

$$\min N\mathrm{E}[u_t^2]$$
$$\text{s.t. } \mathcal{E}_{\mathsf{id}} \subseteq \mathcal{E}_{\mathsf{app}} \subset \mathbb{R}^n$$



An alternative formulation

Application oriented experiment design

$$\min N\mathrm{E}[u_t^2]$$

$$\text{s.t. } \mathcal{E}_{\mathsf{id}} \subseteq \mathcal{E}_{\mathsf{app}} \subset \mathbb{R}^n$$

An alternative formulation

Application oriented experiment design

$$\min N\mathrm{E}[u_t^2]$$

$$\text{s.t. } \mathcal{E}_{\mathsf{id}} \subseteq \mathcal{E}_{\mathsf{app}} \subset \mathbb{R}^n$$

can approximately be formulated as

Application oriented experiment design

$$\min N\mathrm{E}[u_t^2]$$

$$\text{s.t. } NV_{\mathsf{id}}(\theta) \geq \lambda_e \gamma n V_{\mathsf{app}}(\theta), \ \forall \theta \in \mathcal{E}_{\mathsf{app}}$$

Outline

Output error models

True system: $y_t = G_o(q)u_t + e_t, \quad$ open loop

Output error models

True system: $y_t = G_o(q)u_t + e_t, \quad$ open loop

Model: $y_t = G(q, \theta)u_t + e_t$

Output error models

True system: $y_t = G_o(q)u_t + e_t, \quad$ open loop

Model: $y_t = G(q,\theta)u_t + e_t$

PE: $\varepsilon_t(\theta) = y_t - G(q,\theta)u_t$

Output error models

True system: $y_t = G_o(q)u_t + e_t, \quad$ open loop

Model: $y_t = G(q,\theta)u_t + e_t$

PE: $\varepsilon_t(\theta) = y_t - G(q,\theta)u_t = (G_o(q) - G(q,\theta))u_t + e_t$

Output error models

True system: $y_t = G_o(q)u_t + e_t, \quad$ open loop

Model: $y_t = G(q,\theta)u_t + e_t$

PE: $\varepsilon_t(\theta) = y_t - G(q,\theta)u_t = (G_o(q) - G(q,\theta))u_t + e_t$

$$V_{\mathsf{id}}(\theta) = \mathrm{E}[\varepsilon_t^2(\theta)] - \lambda_e$$

Output error models

True system: $y_t = G_o(q)u_t + e_t, \quad$ open loop

Model: $y_t = G(q,\theta)u_t + e_t$

PE: $\varepsilon_t(\theta) = y_t - G(q,\theta)u_t = (G_o(q) - G(q,\theta))u_t + e_t$

$$
\begin{aligned}
V_{\mathsf{id}}(\theta) &= \mathrm{E}[\varepsilon_t^2(\theta)] - \lambda_e \\
&= \mathrm{E}[((G_o(q) - G(q,\theta))u_t)^2] + \mathrm{E}[e_t^2] - \lambda_e
\end{aligned}
$$

Output error models

True system: $y_t = G_o(q)u_t + e_t, \quad$ open loop

Model: $y_t = G(q,\theta)u_t + e_t$

PE: $\varepsilon_t(\theta) = y_t - G(q,\theta)u_t = (G_o(q) - G(q,\theta))u_t + e_t$

$$
\begin{aligned}
V_{\mathsf{id}}(\theta) &= \mathrm{E}[\varepsilon_t^2(\theta)] - \lambda_e \\
&= \mathrm{E}[((G_o(q) - G(q,\theta))u_t)^2] + \mathrm{E}[e_t^2] - \lambda_e \\
&= \frac{1}{2\pi}\int_{-\pi}^{\pi} \Phi_u^{\mathsf{id}}(e^{j\omega}) \left| G_o(e^{j\omega}) - G(e^{j\omega},\theta) \right|^2 \mathrm{d}\omega
\end{aligned}
$$

Output error models

Application oriented experiment design

$$\min N\mathrm{E}[u_t^2]$$

s.t. $$\underbrace{NV_{\mathsf{id}}(\theta)}_{\frac{1}{2\pi}\int_{-\pi}^{\pi} N\Phi_u^{\mathsf{id}}(e^{j\omega})\,|G_o(e^{j\omega})-G(e^{j\omega},\theta)|^2\,\mathrm{d}\omega} \geq \lambda_e \gamma n V_{\mathsf{app}}(\theta), \ \forall\theta\in\mathcal{E}_{\mathsf{app}}$$

Output error models

Application oriented experiment design

$$\min NE[u_t^2] = \frac{1}{2\pi}\int_{-\pi}^{\pi} N\Phi_u^{\mathsf{id}}(e^{j\omega})\,\mathrm{d}\omega$$

s.t. $$\underbrace{NV_{\mathsf{id}}(\theta)}_{\frac{1}{2\pi}\int_{-\pi}^{\pi} N\Phi_u^{\mathsf{id}}(e^{j\omega})\,|G_o(e^{j\omega})-G(e^{j\omega},\theta)|^2\,\mathrm{d}\omega} \geq \lambda_e \gamma n V_{\mathsf{app}}(\theta),\ \forall\theta \in \mathcal{E}_{\mathsf{app}}$$

Output error models

Application oriented experiment design

$$\min N\mathrm{E}[u_t^2] = \frac{1}{2\pi}\int_{-\pi}^{\pi} N\Phi_u^{\mathsf{id}}(e^{j\omega})\,\mathrm{d}\omega$$

$$\text{s.t.} \quad \underbrace{NV_{\mathsf{id}}(\theta)}_{\frac{1}{2\pi}\int_{-\pi}^{\pi} N\Phi_u^{\mathsf{id}}(e^{j\omega})\,|G_o(e^{j\omega})-G(e^{j\omega},\theta)|^2\,\mathrm{d}\omega} \geq \lambda_e \gamma n V_{\mathsf{app}}(\theta),\ \forall\theta \in \mathcal{E}_{\mathsf{app}}$$

- Minimization with respect to energy density spectrum $N\Phi_u^{\mathsf{id}}$

Output error models

Application oriented experiment design

$$\min N\mathrm{E}[u_t^2] = \frac{1}{2\pi}\int_{-\pi}^{\pi} N\Phi_u^{\mathsf{id}}(e^{j\omega})\,\mathrm{d}\omega$$

$$\text{s.t.} \quad \underbrace{NV_{\mathsf{id}}(\theta)}_{\frac{1}{2\pi}\int_{-\pi}^{\pi} N\Phi_u^{\mathsf{id}}(e^{j\omega})\,|G_o(e^{j\omega})-G(e^{j\omega},\theta)|^2\,\mathrm{d}\omega} \geq \lambda_e\gamma n V_{\mathsf{app}}(\theta),\ \forall\theta\in\mathcal{E}_{\mathsf{app}}$$

- Minimization with respect to energy density spectrum $N\Phi_u^{\mathsf{id}}$
- Optimization tries to achieve

$$NV_{\mathsf{id}}(\theta) = \lambda_e\,\gamma\,nV_{\mathsf{app}}(\theta),\ \forall\theta\in\mathcal{E}_{\mathsf{app}}$$

Output error models

Application oriented experiment design

$$\min N\mathrm{E}[u_t^2] = \frac{1}{2\pi}\int_{-\pi}^{\pi} N\Phi_u^{\mathsf{id}}(e^{j\omega})\,\mathrm{d}\omega$$

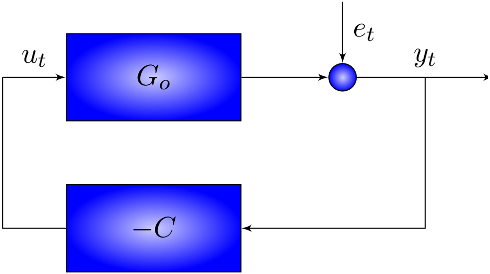
$$\text{s.t.} \quad \underbrace{NV_{\mathsf{id}}(\theta)}_{\frac{1}{2\pi}\int_{-\pi}^{\pi} N\Phi_u^{\mathsf{id}}(e^{j\omega})\,|G_o(e^{j\omega})-G(e^{j\omega},\theta)|^2\,\mathrm{d}\omega} \geq \lambda_e \gamma n V_{\mathsf{app}}(\theta),\ \forall\theta\in\mathcal{E}_{\mathsf{app}}$$

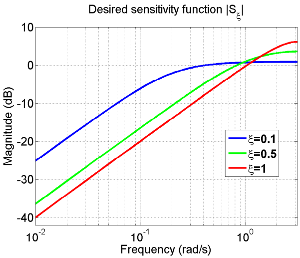
- Minimization with respect to energy density spectrum $N\Phi_u^{\mathsf{id}}$
- Optimization tries to achieve

$$NV_{\mathsf{id}}(\theta) = \lambda_e\,\gamma\,nV_{\mathsf{app}}(\theta),\ \forall\theta\in\mathcal{E}_{\mathsf{app}}$$

Identification cost matched to performance degradation

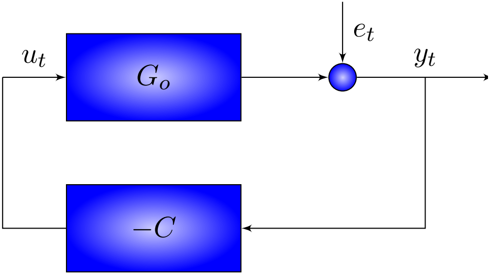
Model Reference Control

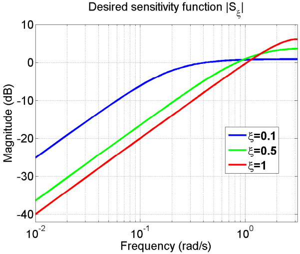




- Controller $C = C(G)$, G output error model

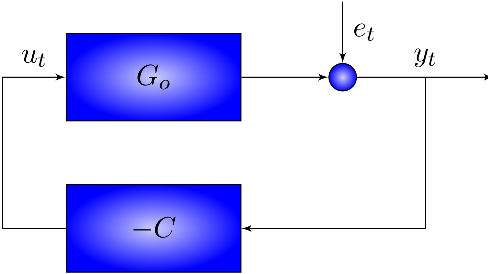
Model Reference Control

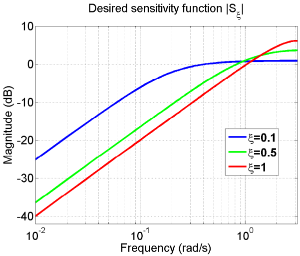




- Controller $C = C(G)$, G output error model
- Desired sensitivity function: S_ξ

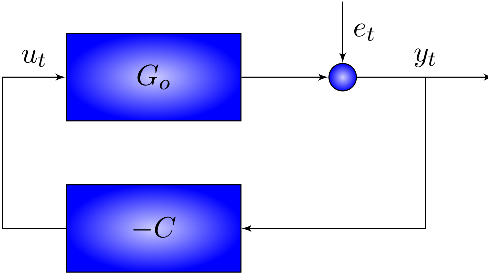
Model Reference Control

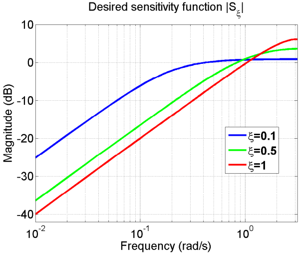




- Controller $C = C(G)$, G output error model
- Desired sensitivity function: S_ξ
- Achieved sensitivity function: $S(G) = \frac{1}{1+C(G)G_o}$

Model Reference Control





- Controller $C = C(G)$, G output error model
- Desired sensitivity function: S_ξ
- Achieved sensitivity function: $S(G) = \frac{1}{1+C(G)G_o}$
- Performance degradation: $V_{\text{app}}(G) := \left\| \frac{S(G)-S_\xi}{S_\xi} \right\|_2^2$

Model Reference Control

$$\min N\mathrm{E}[u^2(t)]$$
$$\text{s.t. } NV_{\mathsf{id}}(\theta) \geq \gamma\lambda_e n\, V_{\mathsf{app}}(\theta)$$

Model Reference Control

$$\min N\mathrm{E}[u^2(t)]$$
$$\text{s.t. } NV_{\mathsf{id}}(\theta) \geq \gamma\lambda_e n\, V_{\mathsf{app}}(\theta)$$

- Matching condition: $NV_{\mathsf{id}}(\theta) = \gamma\lambda_e n V_{\mathsf{app}}(\theta)$

Model Reference Control

$$\min N\mathrm{E}[u^2(t)]$$
$$\text{s.t. } NV_{\mathsf{id}}(\theta) \geq \gamma\lambda_e n\, V_{\mathsf{app}}(\theta)$$

- Matching condition: $NV_{\mathsf{id}}(\theta) = \gamma\lambda_e n V_{\mathsf{app}}(\theta)$

$\Rightarrow$ $N\Phi_u^{\mathsf{id}} = \gamma\lambda_e n\Phi_u^{\mathsf{desired}}$

Model Reference Control

$$\min N\mathrm{E}[u^2(t)]$$
$$\text{s.t. } NV_{\mathsf{id}}(\theta) \geq \gamma\lambda_e n \, V_{\mathsf{app}}(\theta)$$

- Matching condition: $NV_{\mathsf{id}}(\theta) = \gamma\lambda_e n V_{\mathsf{app}}(\theta)$

$\Rightarrow$ $N\Phi_u^{\mathsf{id}} = \gamma\lambda_e n \Phi_u^{\mathsf{desired}}$

- *Experimental conditions during identification should be a scaled version of the desired operating conditions!*

Outline

Static gain estimation

$$y_t = \sum_{t=1}^{n} \theta_k u_{t-k} + e_t$$

Static gain estimation

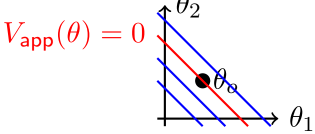
$$y_t = \sum_{t=1}^{n} \theta_k u_{t-k} + e_t$$

Performance degradation: $V_{\mathsf{app}}(\theta) = (\sum \theta_k - \sum \theta_k^o)^2$

Static gain estimation

$$y_t = \sum_{t=1}^{n} \theta_k u_{t-k} + e_t$$

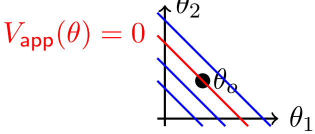
Performance degradation: $V_{\mathsf{app}}(\theta) = (\sum \theta_k - \sum \theta_k^o)^2$



Static gain estimation

$$y_t = \sum_{t=1}^{n} \theta_k u_{t-k} + e_t$$

Performance degradation: $V_{\mathsf{app}}(\theta) = (\sum \theta_k - \sum \theta_k^o)^2$



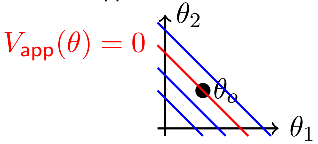
- Optimal input: $u_t = u$ (constant)

Static gain estimation

Model order:	low	true	high
Accuracy:		*good*	

$$y_t = \sum_{t=1}^{n} \theta_k u_{t-k} + e_t$$

Performance degradation: $V_{\mathsf{app}}(\theta) = (\sum \theta_k - \sum \theta_k^o)^2$



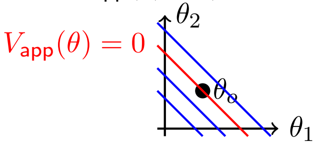
- Optimal input: $u_t = u$ (constant)

Static gain estimation

Model order:	low	true	high
Accuracy:		*good*	

$$y_t = \sum_{t=1}^{n} \theta_k u_{t-k} + e_t$$

Performance degradation: $V_{\mathsf{app}}(\theta) = (\sum \theta_k - \sum \theta_k^o)^2$



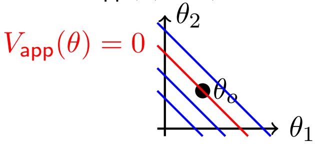
- Optimal input: $u_t = u$ (constant) $\Rightarrow$ $y_t = \left(\sum_k \theta_k^o\right)\ u + e_t$

Static gain estimation

Model order:	low	true	high
Accuracy:		*good*	

$$y_t = \sum_{t=1}^{n} \theta_k u_{t-k} + e_t$$

Performance degradation: $V_{\mathsf{app}}(\theta) = (\sum \theta_k - \sum \theta_k^o)^2$



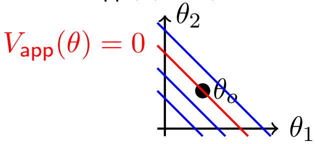
- Optimal input: $u_t = u$ (constant) $\Rightarrow$ $y_t = \left(\sum_k \theta_k^o\right)\ u + e_t$
- Property of interest visible

Static gain estimation

Model order:	low	true	high
Accuracy:		*good*	

$$y_t = \sum_{t=1}^{n} \theta_k u_{t-k} + e_t$$

Performance degradation: $V_{\text{app}}(\theta) = (\sum \theta_k - \sum \theta_k^o)^2$



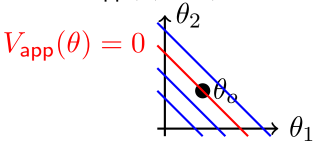
- Optimal input: $u_t = u$ (constant) $\Rightarrow$ $y_t = (\sum_k \theta_k^o)\ u + e_t$
- Property of interest visible
- No other system property visible (due to min energy crit.)

Static gain estimation

Model order:	low	true	high
Accuracy:		*good*	

$$y_t = \sum_{t=1}^{n} \theta_k u_{t-k} + e_t$$

Performance degradation: $V_{\mathsf{app}}(\theta) = (\sum \theta_k - \sum \theta_k^o)^2$



- Optimal input: $u_t = u$ (constant) $\Rightarrow$ $y_t = \left(\sum_k \theta_k^o\right)\ u + e_t$
- Property of interest visible
- No other system property visible (due to min energy crit.)

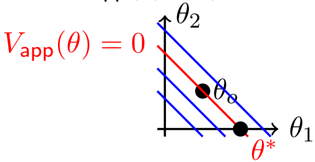
$\Rightarrow$ Accuracy does not decrease if model underparametrized

Static gain estimation

Model order:	low	true	high
Accuracy:		*good*	

$$y_t = \sum_{t=1}^{n} \theta_k u_{t-k} + e_t$$

Performance degradation: $V_{\mathsf{app}}(\theta) = (\sum \theta_k - \sum \theta_k^o)^2$



- Optimal input: $u_t = u$ (constant) $\Rightarrow$ $y_t = \left(\sum_k \theta_k^o\right) \; u + e_t$
- Property of interest visible
- No other system property visible (due to min energy crit.)

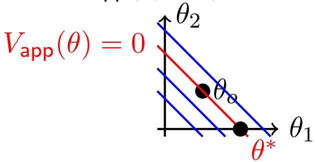
$\Rightarrow$ Accuracy does not decrease if model underparametrized

Static gain estimation

Model order:	low	true	high
Accuracy:	*good*	*good*	

$$y_t = \sum_{t=1}^{n} \theta_k u_{t-k} + e_t$$

Performance degradation: $V_{\text{app}}(\theta) = (\sum \theta_k - \sum \theta_k^o)^2$



- Optimal input: $u_t = u$ (constant) $\Rightarrow$ $y_t = (\sum_k \theta_k^o)\; u + e_t$
- Property of interest visible
- No other system property visible (due to min energy crit.)

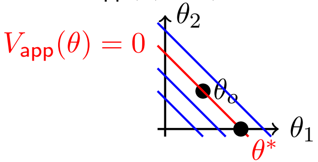
$\Rightarrow$ Accuracy does not decrease if model underparametrized

Static gain estimation

Model order:	low	true	high
Accuracy:	*good*	*good*	*good*

$$y_t = \sum_{t=1}^{n} \theta_k u_{t-k} + e_t$$

Performance degradation: $V_{\text{app}}(\theta) = (\sum \theta_k - \sum \theta_k^o)^2$



- Optimal input: $u_t = u$ (constant) $\Rightarrow$ $y_t = (\sum_k \theta_k^o)\ u + e_t$
- Property of interest visible
- No other system property visible (due to min energy crit.)

$\Rightarrow$ Accuracy does not decrease if model underparametrized

$\Rightarrow$ Accuracy does not decrease if model overparametrized

Application oriented experiment design: Summary

Application oriented experiment design

Aims at achieving

$$N\,V_{\mathsf{id}}(\theta) = \lambda_e\,\gamma\,n\,V_{\mathsf{app}}(\theta)$$

using minimum energy

Application oriented experiment design: Summary

Application oriented experiment design

Aims at achieving

$$N\, V_{\mathsf{id}}(\theta) = \lambda_e\, \gamma\, n\, V_{\mathsf{app}}(\theta)$$

using minimum energy

- Typically leads to similar operating conditions during identification as the ones desired during the application

Application oriented experiment design: Summary

Application oriented experiment design

Aims at achieving

$$N\, V_{\mathsf{id}}(\theta) = \lambda_e\, \gamma\, n\, V_{\mathsf{app}}(\theta)$$

using minimum energy

- Typically leads to similar operating conditions during identification as the ones desired during the application
- To achieve this requires *parsimonious excitation*:

Application oriented experiment design: Summary

Application oriented experiment design

Aims at achieving

$$N\, V_{\mathsf{id}}(\theta) = \lambda_e\, \gamma\, n\, V_{\mathsf{app}}(\theta)$$

using minimum energy

- Typically leads to similar operating conditions during identification as the ones desired during the application
- To achieve this requires *parsimonious excitation*:
  i) System properties important to the application should be visible in the data

Application oriented experiment design: Summary

Application oriented experiment design

Aims at achieving

$$N\, V_{\mathsf{id}}(\theta) = \lambda_e\, \gamma\, n\, V_{\mathsf{app}}(\theta)$$

using minimum energy

- Typically leads to similar operating conditions during identification as the ones desired during the application
- To achieve this requires *parsimonious excitation*:
  - i) System properties important to the application should be visible in the data
  - ii) System properties not important to the application should not be visible in the data, unless necessary for i).

(The let sleeping dogs lie paradigm)

Application oriented experiment design: Summary

Application oriented experiment design

Aims at achieving

$$N\, V_{\mathsf{id}}(\theta) = \lambda_e\, \gamma\, n\, V_{\mathsf{app}}(\theta)$$

using minimum energy

- Typically leads to similar operating conditions during identification as the ones desired during the application
- To achieve this requires *parsimonious excitation*:
  i) System properties important to the application should be visible in the data
  ii) System properties not important to the application should not be visible in the data, unless necessary for i).
  (The let sleeping dogs lie paradigm)
- As a result, the entire system may not have to be identified!

Application oriented experiment design: Summary

Application oriented experiment design

Aims at achieving

$$N\, V_{\mathsf{id}}(\theta) = \lambda_e\, \gamma\, n\, V_{\mathsf{app}}(\theta)$$

using minimum energy

- Typically leads to similar operating conditions during identification as the ones desired during the application
- To achieve this requires *parsimonious excitation*:
  i) System properties important to the application should be visible in the data
  ii) System properties not important to the application should not be visible in the data, unless necessary for i).
  (The let sleeping dogs lie paradigm)
- As a result, the entire system may not have to be identified!
  - Choice of model structure less critical

Outline

Computations: The Information Application Inequality

The design constraint can be written as

$$\mathcal{E}_{\mathsf{id}} \quad \subseteq \quad \mathcal{E}_{\mathsf{app}},$$

Computations: The Information Application Inequality

The design constraint can be written as

$$
\begin{aligned}
\mathcal{E}_{\mathsf{id}} &\subseteq \mathcal{E}_{\mathsf{app}}, \text{ or} \\
N\, V_{\mathsf{id}}(\theta) &\geq \lambda_e \gamma n\, V_{\mathsf{app}}(\theta),\ \forall \theta \in \mathcal{E}_{\mathsf{app}}
\end{aligned}
$$

Computations: The Information Application Inequality

The design constraint can be written as

$$\begin{aligned} \mathcal{E}_{\mathsf{id}} &\subseteq \mathcal{E}_{\mathsf{app}}, \text{ or} \\ N\, V_{\mathsf{id}}(\theta) &\geq \lambda_e \gamma n\, V_{\mathsf{app}}(\theta),\ \forall \theta \in \mathcal{E}_{\mathsf{app}} \end{aligned}$$

but also as

$$\mathcal{I}_1^N(\theta_o) \succeq \frac{\gamma}{2n} V''_{\mathsf{app}}(\theta_o)$$

Computations: The Information Application Inequality

The design constraint can be written as

$$\begin{aligned} \mathcal{E}_{\mathsf{id}} &\subseteq \mathcal{E}_{\mathsf{app}}, \text{ or} \\ N\, V_{\mathsf{id}}(\theta) &\geq \lambda_e \gamma n\, V_{\mathsf{app}}(\theta),\ \forall \theta \in \mathcal{E}_{\mathsf{app}} \end{aligned}$$

but also as

$$\mathcal{I}_1^N(\theta_o) \succeq \frac{\gamma}{2n} V''_{\mathsf{app}}(\theta_o)$$

where $\mathcal{I}_1^N(\theta_o)$ is the Fisher information

Computations: The Information Application Inequality

The design constraint can be written as

$$\begin{aligned} \mathcal{E}_{\mathsf{id}} &\subseteq \mathcal{E}_{\mathsf{app}}, \text{ or} \\ N\, V_{\mathsf{id}}(\theta) &\geq \lambda_e \gamma n\, V_{\mathsf{app}}(\theta),\ \forall \theta \in \mathcal{E}_{\mathsf{app}} \end{aligned}$$

but also as

$$\mathcal{I}_1^N(\theta_o) \succeq \frac{\gamma}{2n} V''_{\mathsf{app}}(\theta_o)$$

where $\mathcal{I}_1^N(\theta_o)$ is the Fisher information $= \frac{N V''_{\mathsf{id}}(\theta_o)}{2\lambda_e}$

Computations: The Information Application Inequality

The design constraint can be written as

$$\begin{aligned} \mathcal{E}_{\mathsf{id}} &\subseteq \mathcal{E}_{\mathsf{app}}, \text{ or} \\ N\, V_{\mathsf{id}}(\theta) &\geq \lambda_e \gamma n\, V_{\mathsf{app}}(\theta),\ \forall \theta \in \mathcal{E}_{\mathsf{app}} \end{aligned}$$

but also as

$$\mathcal{I}_1^N(\theta_o) \succeq \frac{\gamma}{2n} V''_{\mathsf{app}}(\theta_o)$$

where $\mathcal{I}_1^N(\theta_o)$ is the Fisher information $= \frac{N V''_{\mathsf{id}}(\theta_o)}{2\lambda_e}$

The Information Application Inequality

Computations: The Information Application Inequality

The design constraint can be written as

$$\begin{aligned} \mathcal{E}_{\mathsf{id}} &\subseteq \mathcal{E}_{\mathsf{app}}, \text{ or} \\ N\, V_{\mathsf{id}}(\theta) &\geq \lambda_e \gamma n\, V_{\mathsf{app}}(\theta),\ \forall \theta \in \mathcal{E}_{\mathsf{app}} \end{aligned}$$

but also as

$$\mathcal{I}_1^N(\theta_o) \succeq \frac{\gamma}{2n} V''_{\mathsf{app}}(\theta_o)$$

where $\mathcal{I}_1^N(\theta_o)$ is the Fisher information $= \frac{N V''_{\mathsf{id}}(\theta_o)}{2\lambda_e}$

The Information Application Inequality

Recall: V_{id} linear in the input spectrum

Computations: The Information Application Inequality

The design constraint can be written as

$$\begin{aligned} \mathcal{E}_{\mathsf{id}} &\subseteq \mathcal{E}_{\mathsf{app}}, \text{ or} \\ N\, V_{\mathsf{id}}(\theta) &\geq \lambda_e \gamma n\, V_{\mathsf{app}}(\theta),\ \forall \theta \in \mathcal{E}_{\mathsf{app}} \end{aligned}$$

but also as

$$\mathcal{I}_1^N(\theta_o) \succeq \frac{\gamma}{2n} V''_{\mathsf{app}}(\theta_o)$$

where $\mathcal{I}_1^N(\theta_o)$ is the Fisher information $= \frac{N V''_{\mathsf{id}}(\theta_o)}{2\lambda_e}$

The Information Application Inequality

Recall: V_{id} linear in the input spectrum

Information Application Inequality is an LMI in the input spectrum

Computations

$$
\begin{aligned}
\min \quad & N\mathrm{E}[u_t^2] \\
\text{s.t.} \quad & \mathcal{I}_1^N \succeq \frac{\gamma n}{2} V''_{\mathsf{app}}
\end{aligned}
$$

Computations

$$\begin{aligned} \min_{\text{Input spectrum}} \quad & N\mathrm{E}[u_t^2] \\ \text{s.t.} \quad & \mathcal{I}_1^N \succeq \frac{\gamma n}{2} V''_{\text{app}} \end{aligned}$$

Computations

$$\begin{aligned} \min_{\text{Input spectrum}} \quad & N\mathrm{E}[u_t^2] \\ \text{s.t.} \quad & \mathcal{I}_1^N \succeq \frac{\gamma n}{2} V''_{\text{app}} \end{aligned}$$

- Semi-Definite Program (SDP) in input spectrum Φ_u^{id}

Computations

$$\min_{\text{Input spectrum}} \quad N\mathrm{E}[u_t^2]$$
$$\text{s.t.} \quad \mathcal{I}_1^N \succeq \frac{\gamma n}{2} V''_{\text{app}}$$

- Semi-Definite Program (SDP) in input spectrum Φ_u^{id}
- Optimal experiment design $\Rightarrow$ Input spectrum Φ_u^{id}

Computations

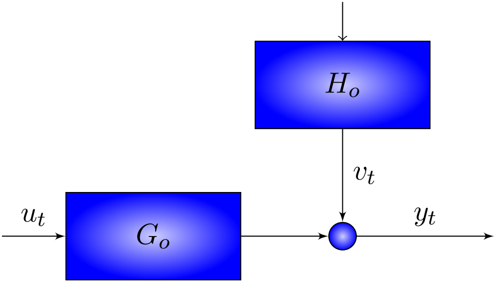
$$\begin{aligned} \min_{\text{Input spectrum}} \quad & N\mathrm{E}[u_t^2] \\ \text{s.t.} \quad & \mathcal{I}_1^N \succeq \frac{\gamma n}{2} V''_{\text{app}} \end{aligned}$$

- Semi-Definite Program (SDP) in input spectrum Φ_u^{id}
- Optimal experiment design $\Rightarrow$ Input spectrum Φ_u^{id}
- Spectral factorization: $\Phi_u^{\text{id}}(e^{j\omega}) = |R_u(e^{j\omega})|^2$

Computations

$$\min_{\text{Input spectrum}} \quad N\mathrm{E}[u_t^2]$$
$$\text{s.t.} \quad \mathcal{I}_1^N \succeq \frac{\gamma n}{2} V''_{\text{app}}$$

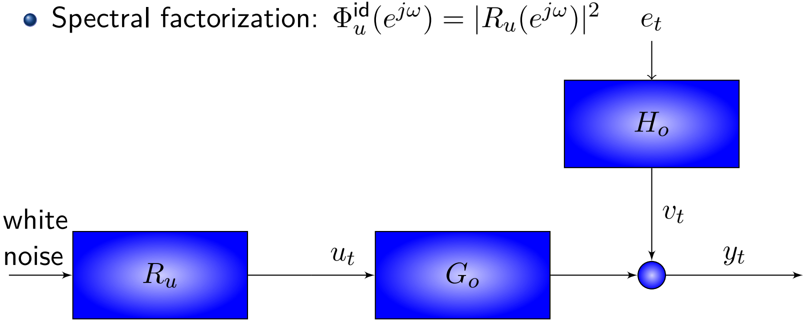
- Semi-Definite Program (SDP) in input spectrum Φ_u^{id}
- Optimal experiment design $\Rightarrow$ Input spectrum Φ_u^{id}
- Spectral factorization: $\Phi_u^{\text{id}}(e^{j\omega}) = |R_u(e^{j\omega})|^2$



Computations

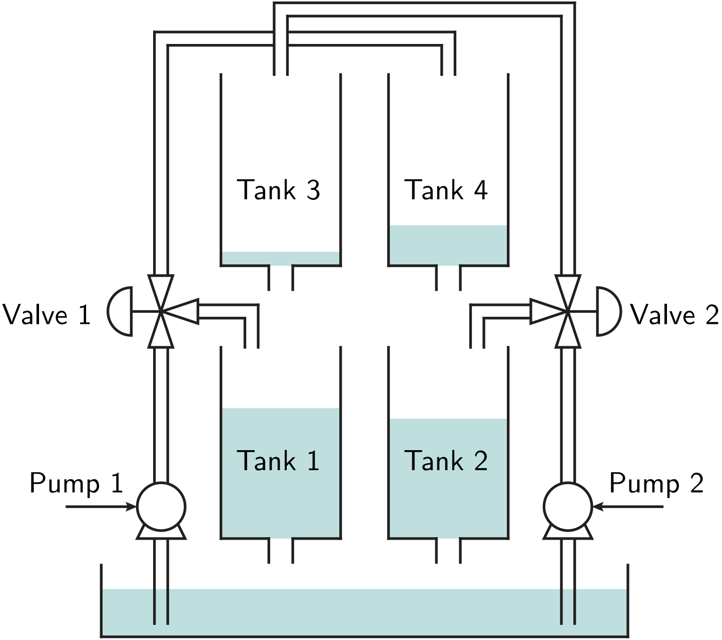
$$\min_{\text{Input spectrum}} \quad N\mathrm{E}[u_t^2]$$
$$\text{s.t.} \quad \mathcal{I}_1^N \succeq \frac{\gamma n}{2} V''_{\text{app}}$$

- Semi-Definite Program (SDP) in input spectrum Φ_u^{id}
- Optimal experiment design $\Rightarrow$ Input spectrum Φ_u^{id}
- Spectral factorization: $\Phi_u^{\text{id}}(e^{j\omega}) = |R_u(e^{j\omega})|^2$

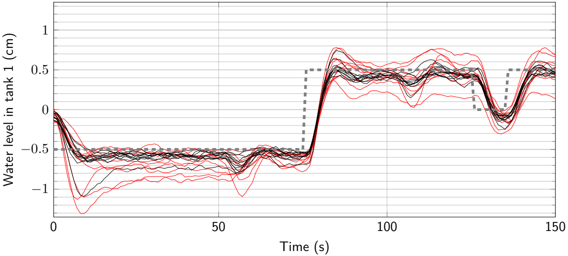


Outline

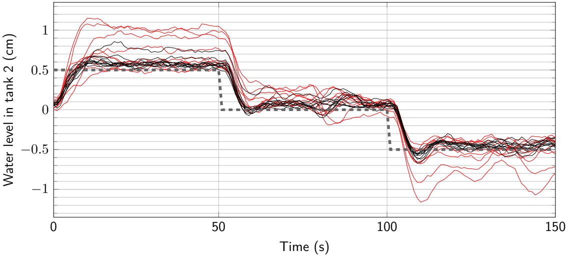
Experimental results: Water tank process



Experimental results: Water tank process



(a)



(b)

MPC: Black: based on AOID-model. Red: based on white noise excitation

Computations

Application oriented experiment design

$$\min N \mathrm{E}[u^2(t)]$$

$$\text{s.t. } \mathcal{I}_1^N(\theta_o) \succeq \frac{\gamma n}{2} V''_{\mathsf{app}}(\theta_o)$$

Computations

Application oriented experiment design

$$\min N \mathrm{E}[u^2(t)]$$
$$\text{s.t. } \mathcal{I}_1^N(\theta_o) \succeq \frac{\gamma n}{2} V''_{\mathsf{app}}(\theta_o)$$

Computations

Application oriented experiment design

$$\min N \mathrm{E}[u^2(t)]$$
$$\text{s.t. } \mathcal{I}_1^N(\theta_o) \succeq \frac{\gamma n}{2} V''_{\mathsf{app}}(\theta_o)$$

- Optimization problem depends on the unknown system!

Computations

Application oriented experiment design

$$\min N \mathrm{E}[u^2(t)]$$
$$\text{s.t. } \mathcal{I}_1^N(\theta_o) \succeq \frac{\gamma n}{2} V''_{\mathsf{app}}(\theta_o)$$

- Optimization problem depends on the unknown system!
- Solutions:

Computations

Application oriented experiment design

$$\min N \mathrm{E}[u^2(t)]$$
$$\text{s.t. } \mathcal{I}_1^N(\theta_o) \succeq \frac{\gamma n}{2} V''_{\mathsf{app}}(\theta_o)$$

- Optimization problem depends on the unknown system!
- Solutions:
  - Robust experiment design
    (e.g. Rojas, Welsh, Goodwin, Feuer 2007)

Computations

Application oriented experiment design

$$\min N \mathrm{E}[u^2(t)]$$
$$\text{s.t. } \mathcal{I}_1^N(\theta_o) \succeq \frac{\gamma n}{2} V''_{\mathsf{app}}(\theta_o)$$

- Optimization problem depends on the unknown system!
- Solutions:
  - Robust experiment design
    (e.g. Rojas, Welsh, Goodwin, Feuer 2007)
  - Adaptive (or sequential) experiment design

Computations

Application oriented experiment design

$$\min N\mathrm{E}[u^2(t)]$$
$$\text{s.t. } \mathcal{I}_1^N(\theta_o) \succeq \frac{\gamma n}{2} V''_{\mathsf{app}}(\theta_o)$$

- Optimization problem depends on the unknown system!
- Solutions:
  - Robust experiment design
    (e.g. Rojas, Welsh, Goodwin, Feuer 2007)
  - Adaptive (or sequential) experiment design
    $\Rightarrow$ Active application oriented learning

Outline

Application oriented experiment design

Output error models

The impact of optimal experiments on the identification problem

Computing the optimal input

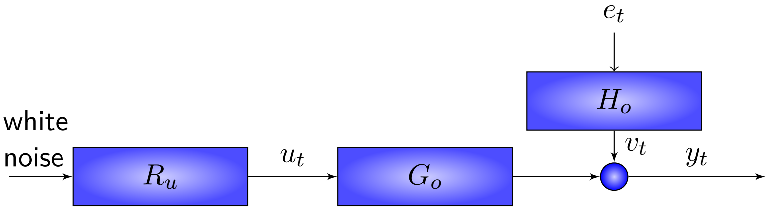
Experimental results

Active application oriented learning

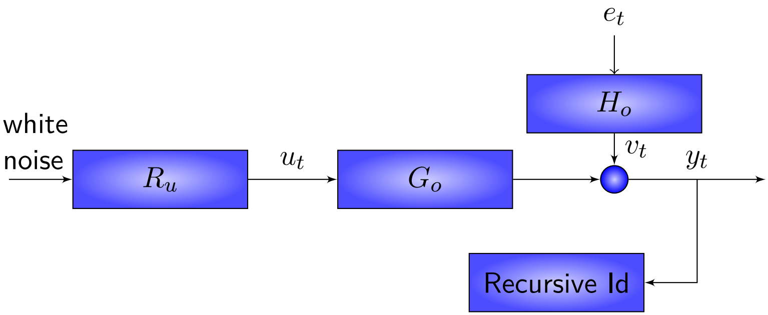
Application oriented dual control

Summary

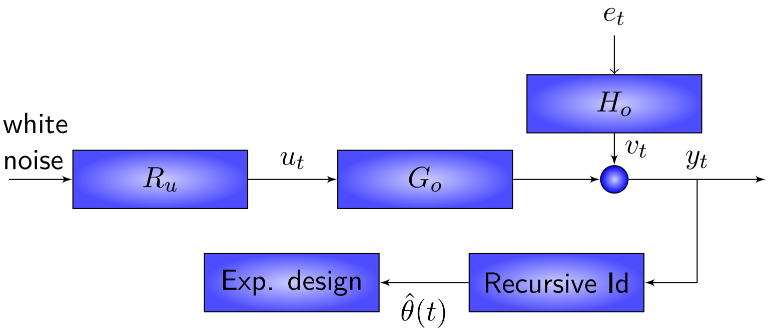
Active application oriented learning



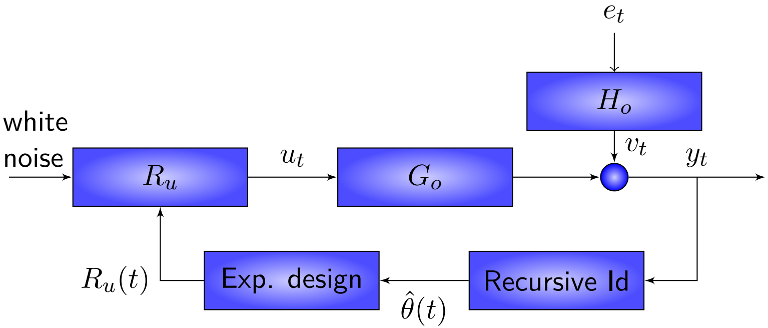
Active application oriented learning



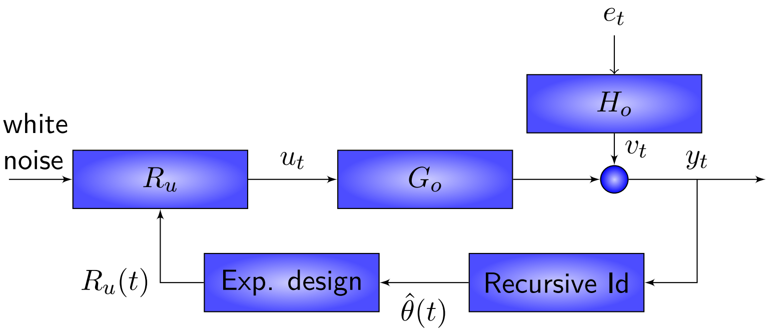
Active application oriented learning



Active application oriented learning

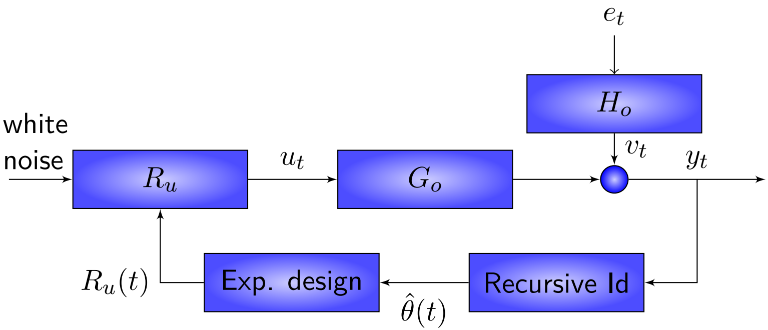


Active application oriented learning



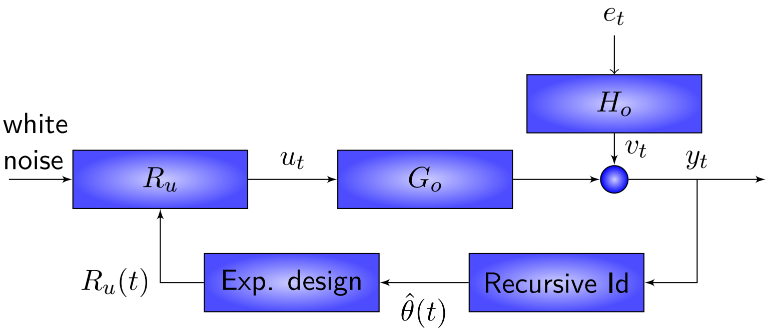
- An adaptive feedback system

Active application oriented learning



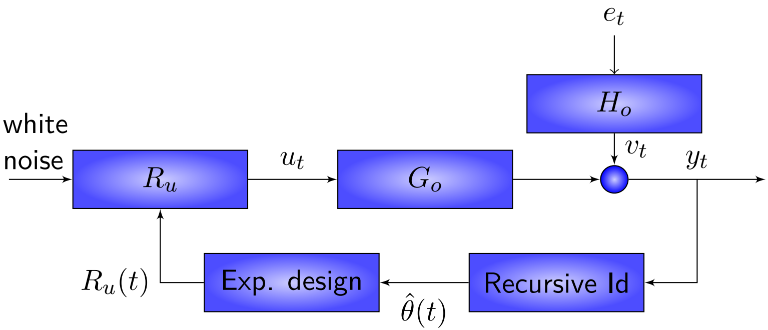
- An adaptive feedback system
- But measured signal not fed back directly

Active application oriented learning



- An adaptive feedback system
- But measured signal not fed back directly
- Exp. design limits input power $\Rightarrow$ Stability when G_o stable

Active application oriented learning



- An adaptive feedback system
- But measured signal not fed back directly
- Exp. design limits input power $\Rightarrow$ Stability when G_o stable

Key questions:

- Convergence?
- Accuracy?

Active application oriented learning

Key questions:

- Convergence?
- Accuracy?

Active application oriented learning

Key questions:

- Convergence?
- Accuracy?

Theorem

- *True linear time-invariant system in the model set*
- *System stable*

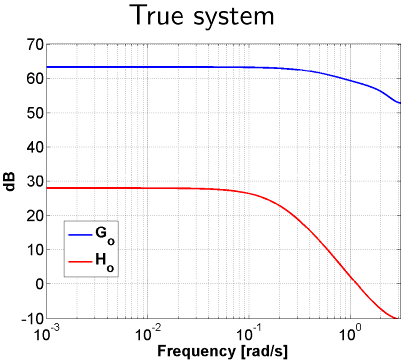
$\Rightarrow$ $\hat{\theta}(t)$ *has the same asymptotic accuracy as the off-line estimate that uses data collected under the optimal experimental conditions (using knowledge of* θ_o*)*

Active application oriented learning

What happens when true system is not in the model set?

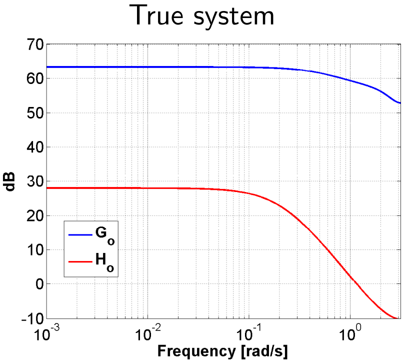
Example: Non-minimum phase zero estimation

True system: $y_t = \dfrac{(q-3)(q-0.1)(q-0.2)(q+0.3)}{q^4(q-0.5)} u_t + \dfrac{q}{q-0.8} e_t^o$



Example: Non-minimum phase zero estimation

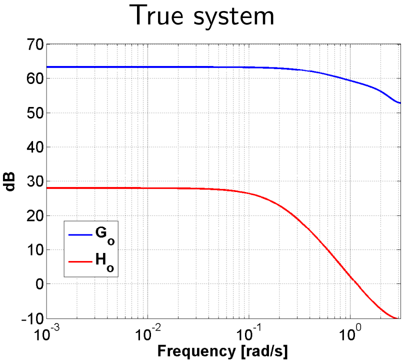
$$\text{True system: } y_t = \frac{(q-3)(q-0.1)(q-0.2)(q+0.3)}{q^4(q-0.5)}u_t + \frac{q}{q-0.8}e_t^o$$



Example: Non-minimum phase zero estimation

$$\text{True system: } y_t = \frac{(q-3)(q-0.1)(q-0.2)(q+0.3)}{q^4(q-0.5)}u_t + \frac{q}{q-0.8}e_t^o$$

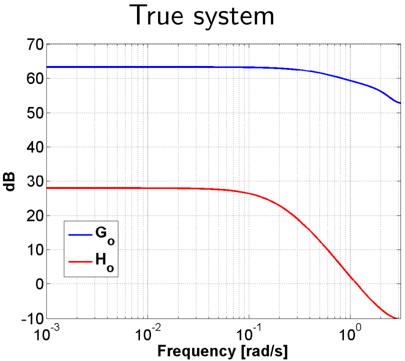
$$\text{Model: } y_t = \frac{\theta_1 q + \theta_2}{q^2}u_t + e_t$$

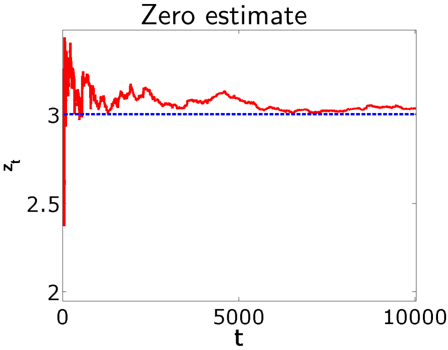


Example: Non-minimum phase zero estimation

$$\text{True system: } y_t = \frac{(q-3)(q-0.1)(q-0.2)(q+0.3)}{q^4(q-0.5)}u_t + \frac{q}{q-0.8}e_t^o$$

$$\text{Model: } y_t = \frac{\theta_1 q + \theta_2}{q^2}u_t + e_t$$





Outline

Application oriented dual control

$$\min_{\text{Input spectrum}} \quad N\mathrm{E}[u_t^2], \quad \text{s.t. } \mathcal{I}_1^N(\theta_o) \succeq \frac{\gamma n}{2} V''_{\mathsf{app}}(\theta_o)$$

Application oriented dual control

$$\min_{\text{Input spectrum}} \quad N\mathrm{E}[u_t^2], \quad \text{s.t. } \mathcal{I}_1^N(\theta_o) \succeq \frac{\gamma n}{2} V''_{\mathsf{app}}(\theta_o)$$

Idéa: Replace cost function with control objective.

Application oriented dual control

$$\min_{\text{Input spectrum}} \quad N\mathrm{E}[u_t^2], \quad \text{s.t. } \mathcal{I}_1^N(\theta_o) \succeq \frac{\gamma n}{2} V''_{\mathsf{app}}(\theta_o)$$

Idéa: Replace cost function with control objective.

Policy (control rule): $\pi = (\pi_1, \pi_2, \ldots)$

Application oriented dual control

$$\min_{\text{Input spectrum}} \quad N\mathrm{E}[u_t^2], \quad \text{s.t. } \mathcal{I}_1^N(\theta_o) \succeq \frac{\gamma n}{2} V''_{\mathsf{app}}(\theta_o)$$

Idéa: Replace cost function with control objective.

Policy (control rule): $\pi = (\pi_1, \pi_2, \ldots)$

Initial state distribution: $\beta(x)$

Application oriented dual control

$$\min_{\text{Input spectrum}} \quad N\mathrm{E}[u_t^2], \quad \text{s.t. } \mathcal{I}_1^N(\theta_o) \succeq \frac{\gamma n}{2} V''_{\mathsf{app}}(\theta_o)$$

Idéa: Replace cost function with control objective.

Policy (control rule): $\pi = (\pi_1, \pi_2, \ldots)$

Initial state distribution: $\beta(x)$

Instantaneous cost: $c(x_t, u_t)$

Application oriented dual control

$$\min_{\text{Input spectrum}} \quad N\mathrm{E}[u_t^2], \quad \text{s.t. } \mathcal{I}_1^N(\theta_o) \succeq \frac{\gamma n}{2} V''_{\mathsf{app}}(\theta_o)$$

Idéa: Replace cost function with control objective.

Policy (control rule): $\pi = (\pi_1, \pi_2, \ldots)$

Initial state distribution: $\beta(x)$

Instantaneous cost: $c(x_t, u_t)$ e.g. $y_t^T Q y_t + u_t S u_t$

Application oriented dual control

$$\min_{\text{Input spectrum}} \quad N\mathrm{E}[u_t^2], \quad \text{s.t. } \mathcal{I}_1^N(\theta_o) \succeq \frac{\gamma n}{2} V''_{\mathsf{app}}(\theta_o)$$

Idéa: Replace cost function with control objective.

Policy (control rule): $\pi = (\pi_1, \pi_2, \ldots)$

Initial state distribution: $\beta(x)$

Instantaneous cost: $c(x_t, u_t)$ e.g. $y_t^T Q y_t + u_t S u_t$

Expected average cost: $C_\beta(\pi, N) = \frac{1}{N} \sum_{t=1}^{N} \mathbb{E}_\beta^\pi \{c(x_t, u_t)\}$

Application oriented dual control

$$\min_{\text{Input spectrum}} \quad N\mathrm{E}[u_t^2], \quad \text{s.t. } \mathcal{I}_1^N(\theta_o) \succeq \frac{\gamma n}{2} V''_{\text{app}}(\theta_o)$$

Idéa: Replace cost function with control objective.

Policy (control rule): $\pi = (\pi_1, \pi_2, \ldots)$

Initial state distribution: $\beta(x)$

Instantaneous cost: $c(x_t, u_t)$ e.g. $y_t^T Q y_t + u_t S u_t$

Expected average cost: $C_\beta(\pi, N) = \frac{1}{N} \sum_{t=1}^{N} \mathbb{E}_\beta^\pi \{c(x_t, u_t)\}$

Constraints: $x_t \in \mathcal{X} \subseteq \mathbf{R}^n$, $y_t \in \mathcal{Y} \subseteq R^p$, $u_t \in \mathcal{U} \subset \mathbf{R}^m$

Application oriented dual control

$$\min_{\text{Input spectrum}} \quad N\mathrm{E}[u_t^2], \quad \text{s.t. } \mathcal{I}_1^N(\theta_o) \succeq \frac{\gamma n}{2} V''_{\text{app}}(\theta_o)$$

Idéa: Replace cost function with control objective.

Policy (control rule): $\pi = (\pi_1, \pi_2, \ldots)$

Initial state distribution: $\beta(x)$

Instantaneous cost: $c(x_t, u_t)$ e.g. $y_t^T Q y_t + u_t S u_t$

Expected average cost: $C_\beta(\pi, N) = \frac{1}{N} \sum_{t=1}^{N} \mathbb{E}_\beta^\pi \{c(x_t, u_t)\}$

Constraints: $x_t \in \mathcal{X} \subseteq \mathbf{R}^n$, $y_t \in \mathcal{Y} \subseteq R^p$, $u_t \in \mathcal{U} \subset \mathbf{R}^m$

$$\begin{aligned} \min_{\pi} \quad & C_\beta(\pi, N) \\ \text{s.t.} \quad & \mathcal{I}_1^N(\theta_o) \succeq \frac{\gamma n}{2} V''_{\text{app}}(\theta_o) \quad \text{Reward} \\ & x_t \in \mathcal{X} \subseteq \mathbf{R}^n,\ y_t \in \mathcal{Y} \subseteq R^p,\ u_t \in \mathcal{U} \subset \mathbf{R}^m \end{aligned}$$

Application oriented dual control

$$
\begin{aligned}
\min_{\pi} \quad & C_\beta(\pi, N) \\
\text{s.t.} \quad & \mathcal{I}_1^N(\theta_o) \succeq \frac{\gamma n}{2} V''_{\mathsf{app}}(\theta_o) \\
& x_t \in \mathcal{X} \subseteq \mathbf{R}^n,\ y_t \in \mathcal{Y} \subseteq R^p,\ u_t \in \mathcal{U} \subset \mathbf{R}^m
\end{aligned}
$$

Application oriented dual control

$$
\begin{aligned}
\min_{\pi} \quad & C_\beta(\pi, N) \\
\text{s.t.} \quad & \mathcal{I}_1^N(\theta_o) \succeq \frac{\gamma n}{2} V''_{\mathsf{app}}(\theta_o) \\
& x_t \in \mathcal{X} \subseteq \mathbf{R}^n,\ y_t \in \mathcal{Y} \subseteq R^p,\ u_t \in \mathcal{U} \subset \mathbf{R}^m
\end{aligned}
$$

We will look at two approaches two solve this problem:

Application oriented dual control

$$
\begin{aligned}
\min_{\pi} \quad & C_\beta(\pi, N) \\
\text{s.t.} \quad & \mathcal{I}_1^N(\theta_o) \succeq \frac{\gamma n}{2} V''_{\mathsf{app}}(\theta_o) \\
& x_t \in \mathcal{X} \subseteq \mathbf{R}^n,\ y_t \in \mathcal{Y} \subseteq R^p,\ u_t \in \mathcal{U} \subset \mathbf{R}^m
\end{aligned}
$$

We will look at two approaches two solve this problem:

- Markov Decision Process (MDP) formulation

Application oriented dual control

$$\begin{aligned} \min_{\pi} \quad & C_\beta(\pi, N) \\ \text{s.t.} \quad & \mathcal{I}_1^N(\theta_o) \succeq \frac{\gamma n}{2} V''_{\mathsf{app}}(\theta_o) \\ & x_t \in \mathcal{X} \subseteq \mathbf{R}^n,\ y_t \in \mathcal{Y} \subseteq R^p,\ u_t \in \mathcal{U} \subset \mathbf{R}^m \end{aligned}$$

We will look at two approaches two solve this problem:

- Markov Decision Process (MDP) formulation
- Receeding horizon formulation: MPC-X

Markov Decision Process formulation

Infinite horizon cost: $C_\beta(\pi) = \limsup_{N\to} C_\beta(\pi, N)$

Markov Decision Process formulation

Infinite horizon cost: $C_\beta(\pi) = \limsup_{N\to} C_\beta(\pi, N)$

Reward: $\mathcal{I}_\beta(\pi, \theta) \triangleq \lim_{N\to\infty} \frac{1}{N}\mathcal{I}_1^N(\theta)$

Markov Decision Process formulation

Infinite horizon cost: $C_\beta(\pi) = \limsup_{N\rightarrow} C_\beta(\pi, N)$

Reward: $\mathcal{I}_\beta(\pi, \theta) \triangleq \lim_{N\rightarrow\infty} \frac{1}{N}\mathcal{I}_1^N(\theta)$ (per sample information)

Markov Decision Process formulation

Infinite horizon cost: $C_\beta(\pi) = \limsup_{N\to} C_\beta(\pi, N)$

Reward: $\mathcal{I}_\beta(\pi,\theta) \triangleq \lim_{N\to\infty} \frac{1}{N}\mathcal{I}_1^N(\theta)$ (per sample information)

Theory based on discretized state and action spaces

Markov Decision Process formulation

Infinite horizon cost: $C_\beta(\pi) = \limsup_{N\rightarrow} C_\beta(\pi, N)$

Reward: $\mathcal{I}_\beta(\pi,\theta) \triangleq \lim_{N\rightarrow\infty} \frac{1}{N}\mathcal{I}_1^N(\theta)$ (per sample information)

Theory based on discretized state and action spaces

$\Rightarrow$ State x & input u take only finite number of values

Markov Decision Process formulation

Infinite horizon cost: $C_\beta(\pi) = \limsup_{N\to} C_\beta(\pi, N)$

Reward: $\mathcal{I}_\beta(\pi,\theta) \triangleq \lim_{N\to\infty} \frac{1}{N}\mathcal{I}_1^N(\theta)$ (per sample information)

Theory based on discretized state and action spaces

$\Rightarrow$ State x & input u take only finite number of values

$\Rightarrow$ Need to approximate state-space description

Markov Decision Process formulation

Infinite horizon cost: $C_\beta(\pi) = \limsup_{N\to} C_\beta(\pi, N)$

Reward: $\mathcal{I}_\beta(\pi, \theta) \triangleq \lim_{N\to\infty} \frac{1}{N}\mathcal{I}_1^N(\theta)$ (per sample information)

Theory based on discretized state and action spaces

$\Rightarrow$ State x & input u take only finite number of values

$\Rightarrow$ Need to approximate state-space description

$\Rightarrow$ Transition probabilities: $p_{x\bar{x}}(u) = \mathbb{P}\{x_{t+1} = \bar{x}\,|x_t = x, u_t = u\}$

Markov Decision Process formulation

Infinite horizon cost: $C_\beta(\pi) = \limsup_{N\to} C_\beta(\pi, N)$

Reward: $\mathcal{I}_\beta(\pi,\theta) \triangleq \lim_{N\to\infty} \frac{1}{N}\mathcal{I}_1^N(\theta)$ (per sample information)

Theory based on discretized state and action spaces

$\Rightarrow$ State x & input u take only finite number of values

$\Rightarrow$ Need to approximate state-space description

$\Rightarrow$ Transition probabilities: $p_{x\bar{x}}(u) = \mathbb{P}\{x_{t+1} = \bar{x}\,|x_t = x, u_t = u\}$

Can be computed based on geometry of discretization and knowledge of distributions of disturbances

Markov Decision Process formulation

Infinite horizon cost: $C_\beta(\pi) = \limsup_{N\to} C_\beta(\pi, N)$

Reward: $\mathcal{I}_\beta(\pi, \theta) \triangleq \lim_{N\to\infty} \frac{1}{N}\mathcal{I}_1^N(\theta)$ (per sample information)

Theory based on discretized state and action spaces

$\Rightarrow$ State x & input u take only finite number of values

$\Rightarrow$ Need to approximate state-space description

$\Rightarrow$ Transition probabilities: $p_{x\bar{x}}(u) = \mathbb{P}\{x_{t+1} = \bar{x} \,|x_t = x, u_t = u\}$

Can be computed based on geometry of discretization and knowledge of distributions of disturbances

Policy: $\pi_t(x, u) = \mathbb{P}\{u_t = u \,|x_t = x\}$

Markov Decision Process formulation: Implementation

$$
\begin{aligned}
\min_{\pi} \quad & C_\beta(\pi) \\
\text{s.t.} \quad & N\mathcal{I}_\beta(\pi, \theta_o) \succeq \frac{\gamma n}{2} V''_{\mathsf{app}}(\theta_o) \\
& x_t \in \mathcal{X} \subseteq \mathbf{R}^n,\ y_t \in \mathcal{Y} \subseteq R^p,\ u_t \in \mathcal{U} \subset \mathbf{R}^m
\end{aligned}
$$

Markov Decision Process formulation: Implementation

$$
\begin{aligned}
\min_{\pi} \quad & C_\beta(\pi) \\
\text{s.t.} \quad & \color{red}{N}\mathcal{I}_\beta(\pi, \theta_o) \succeq \frac{\gamma n}{2} V''_{\mathsf{app}}(\theta_o) \\
& x_t \in \mathcal{X} \subseteq \mathbf{R}^n,\ y_t \in \mathcal{Y} \subseteq R^p,\ u_t \in \mathcal{U} \subset \mathbf{R}^m
\end{aligned}
$$

Solution?

Markov Decision Process formulation: Implementation

$$\begin{aligned} \min_{\pi} \quad & C_\beta(\pi) \\ \text{s.t.} \quad & N\mathcal{I}_\beta(\pi, \theta_o) \succeq \frac{\gamma n}{2} V''_{\mathsf{app}}(\theta_o) \\ & x_t \in \mathcal{X} \subseteq \mathbf{R}^n,\ y_t \in \mathcal{Y} \subseteq R^p,\ u_t \in \mathcal{U} \subset \mathbf{R}^m \end{aligned}$$

Solution?

Define z_{xu} as the probability of being in state x and taking action u

Markov Decision Process formulation: Implementation

$$\begin{aligned} \min_{\pi} \quad & C_\beta(\pi) \\ \text{s.t.} \quad & \color{red}{N}\color{black}\mathcal{I}_\beta(\pi, \theta_o) \succeq \frac{\gamma n}{2} V''_{\mathsf{app}}(\theta_o) \\ & x_t \in \mathcal{X} \subseteq \mathbf{R}^n,\ y_t \in \mathcal{Y} \subseteq R^p,\ u_t \in \mathcal{U} \subset \mathbf{R}^m \end{aligned}$$

Solution?

Define z_{xu} as the probability of being in state x and taking action u

Occupancy measure

Markov Decision Process formulation: Implementation

$$
\begin{aligned}
\min_{\pi} \quad & C_\beta(\pi) \\
\text{s.t.} \quad & N\mathcal{I}_\beta(\pi, \theta_o) \succeq \frac{\gamma n}{2} V''_{\mathsf{app}}(\theta_o) \\
& x_t \in \mathcal{X} \subseteq \mathbf{R}^n,\ y_t \in \mathcal{Y} \subseteq R^p,\ u_t \in \mathcal{U} \subset \mathbf{R}^m
\end{aligned}
$$

Solution?

Define z_{xu} as the probability of being in state x and taking action u

Occupancy measure

MDP problem is a semi-definite program in $\{z_{xu}\}$.

Markov Decision Process formulation: Simulation study

$$\begin{cases} x_{t+1} = -\theta_1 x_t + \theta_2 u_t - \theta_1 v_t, \\ \quad y_t = x_t + v_t, \end{cases}$$

Markov Decision Process formulation: Simulation study

$$\begin{cases} x_{t+1} = -\theta_1 x_t + \theta_2 u_t - \theta_1 v_t, \\ \quad y_t = x_t + v_t, \end{cases}$$

v_t: Gaussian white noise with variance 1×10^{-3}

Markov Decision Process formulation: Simulation study

$$\begin{cases} x_{t+1} = -\theta_1 x_t + \theta_2 u_t - \theta_1 v_t, \\ y_t = x_t + v_t, \end{cases}$$

v_t: Gaussian white noise with variance 1×10^{-3}

$\theta_o = [0.5, 0.5]^T$

Markov Decision Process formulation: Simulation study

$$\begin{cases} x_{t+1} = -\theta_1 x_t + \theta_2 u_t - \theta_1 v_t, \\ y_t = x_t + v_t, \end{cases}$$

v_t: Gaussian white noise with variance 1×10^{-3}

$\theta_o = [0.5, 0.5]^T$

x split in 51 regions.

u split in 21 regions.

Markov Decision Process formulation: Simulation study

$$\begin{cases} x_{t+1} = -\theta_1 x_t + \theta_2 u_t - \theta_1 v_t, \\ \quad y_t = x_t + v_t, \end{cases}$$

v_t: Gaussian white noise with variance 1×10^{-3}

$\theta_o = [0.5, 0.5]^T$

x split in 51 regions.

u split in 21 regions.

$c_t(x, u) = 2y_t^2 + u_t^2$

Markov Decision Process formulation: Simulation study

$$\begin{cases} x_{t+1} = -\theta_1 x_t + \theta_2 u_t - \theta_1 v_t, \\ y_t = x_t + v_t, \end{cases}$$

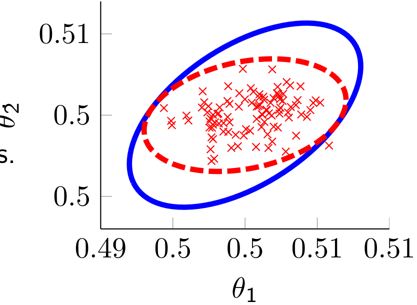
v_t: Gaussian white noise with variance 1×10^{-3}

$\theta_o = [0.5, 0.5]^T$

x split in 51 regions

u split in 21 regions.

$c_t(x, u) = 2y_t^2 + u_t^2$



Markov Decision Process formulation: Simulation study

$$\begin{cases} x_{t+1} = -\theta_1 x_t + \theta_2 u_t - \theta_1 v_t, \\ \quad y_t = x_t + v_t, \end{cases}$$

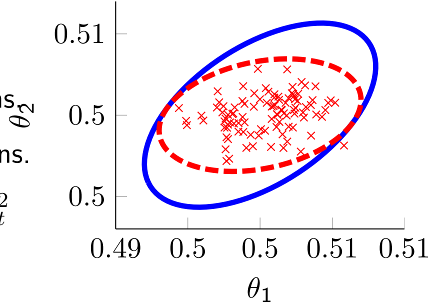
v_t: Gaussian white noise with variance 1×10^{-3}

$\theta_o = [0.5, 0.5]^T$

x split in 51 regions

u split in 21 regions.

$c_t(x, u) = 2y_t^2 + u_t^2$



Set of acceptable models: Blue solid ellipse.

Markov Decision Process formulation: Simulation study

$$\begin{cases} x_{t+1} = -\theta_1 x_t + \theta_2 u_t - \theta_1 v_t, \\ y_t = x_t + v_t, \end{cases}$$

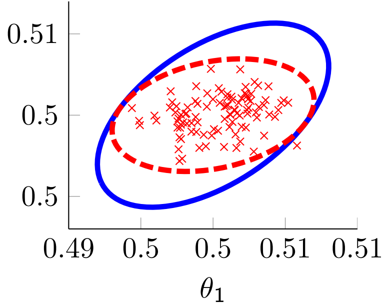
v_t: Gaussian white noise with variance 1×10^{-3}

$\theta_o = [0.5, 0.5]^T$

x split in 51 regions

u split in 21 regions.

$c_t(x,u) = 2y_t^2 + u_t^2$



Set of acceptable models: Blue solid ellipse.

Desired confidence ellipsoid: Red dashed ellipse

Markov Decision Process formulation: Simulation study

$$\begin{cases} x_{t+1} = -\theta_1 x_t + \theta_2 u_t - \theta_1 v_t, \\ y_t = x_t + v_t, \end{cases}$$

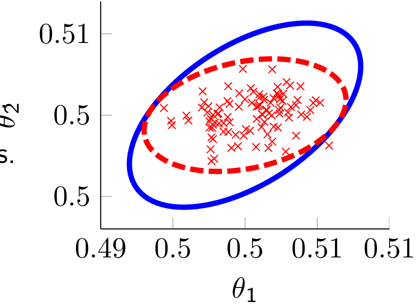
v_t: Gaussian white noise with variance 1×10^{-3}

$\theta_o = [0.5, 0.5]^T$

x split in 51 regions

u split in 21 regions.

$c_t(x, u) = 2y_t^2 + u_t^2$



Set of acceptable models: Blue solid ellipse.

Desired confidence ellipsoid: Red dashed ellipse

Crosses: 100 Monte Carlo simulations using the MDP controller

Markov Decision Process formulation: Summary

- Elegant and powerful formulation

Markov Decision Process formulation: Summary

- Elegant and powerful formulation
- Leads to a semi-definite program

Markov Decision Process formulation: Summary

- Elegant and powerful formulation
- Leads to a semi-definite program
- but suffers from the curse of dimensionality due to discretization of state-space

Receeding horizon formulation

Receeding horizon formulation

Cost at time t:

$$C_t = \sum_{k=1}^{F} c_t(x_k, u_k) = \sum_{k=1}^{F} \|y_{k+1} - r_{t+k+1}\|_Q^2 + \sum_{k=1}^{F} \|u_k\|_S^2$$

Receeding horizon formulation

Cost at time t:

$$C_t = \sum_{k=1}^{F} c_t(x_k, u_k) = \sum_{k=1}^{F} \|y_{k+1} - r_{t+k+1}\|_Q^2 + \sum_{k=1}^{F} \|u_k\|_S^2$$

$$\underset{\{u_k\}_{k=1}^{F}}{\text{minimize}} \quad C_t$$

Receeding horizon formulation

Cost at time t:

$$C_t = \sum_{k=1}^{F} c_t(x_k, u_k) = \sum_{k=1}^{F} \|y_{k+1} - r_{t+k+1}\|_Q^2 + \sum_{k=1}^{F} \|u_k\|_S^2$$

$$\begin{aligned} \underset{\{u_k\}_{k=1}^{F}}{\text{minimize}} \quad & C_t \\ \text{subject to} \quad & x_1 = \hat{x}_t, \end{aligned}$$

Receeding horizon formulation

Cost at time t:

$$C_t = \sum_{k=1}^{F} c_t(x_k, u_k) = \sum_{k=1}^{F} \|y_{k+1} - r_{t+k+1}\|_Q^2 + \sum_{k=1}^{F} \|u_k\|_S^2$$

$$\begin{aligned} \underset{\{u_k\}_{k=1}^F}{\text{minimize}} \quad & C_t \\ \text{subject to} \quad & x_1 = \hat{x}_t, \\ & x_{k+1} = A(\theta_o)x_k + B(\theta_o)u_k, \end{aligned}$$

Receeding horizon formulation

Cost at time t:

$$C_t = \sum_{k=1}^{F} c_t(x_k, u_k) = \sum_{k=1}^{F} \|y_{k+1} - r_{t+k+1}\|_Q^2 + \sum_{k=1}^{F} \|u_k\|_S^2$$

$$\begin{aligned} \underset{\{u_k\}_{k=1}^{F}}{\text{minimize}} \quad & C_t \\ \text{subject to} \quad & x_1 = \hat{x}_t, \\ & x_{k+1} = A(\theta_o)x_k + B(\theta_o)u_k,\ y_k = C(\theta_o)x_k \end{aligned}$$

Receeding horizon formulation

Cost at time t:

$$C_t = \sum_{k=1}^{F} c_t(x_k, u_k) = \sum_{k=1}^{F} \|y_{k+1} - r_{t+k+1}\|_Q^2 + \sum_{k=1}^{F} \|u_k\|_S^2$$

$$\begin{array}{ll} \underset{\{u_k\}_{k=1}^{F}}{\text{minimize}} & C_t \\ \text{subject to} & x_1 = \hat{x}_t, \\ & x_{k+1} = A(\theta_o)x_k + B(\theta_o)u_k,\ y_k = C(\theta_o)x_k \\ & u_k \in \mathcal{U},\ x_k \in \mathcal{X},\ y_k \in \mathcal{Y} \end{array}$$

Receeding horizon formulation

Cost at time t:

$$C_t = \sum_{k=1}^{F} c_t(x_k, u_k) = \sum_{k=1}^{F} \|y_{k+1} - r_{t+k+1}\|_Q^2 + \sum_{k=1}^{F} \|u_k\|_S^2$$

$$\begin{aligned}
\underset{\{u_k\}_{k=1}^{F}}{\text{minimize}} \quad & C_t \\
\text{subject to} \quad & x_1 = \hat{x}_t, \\
& x_{k+1} = A(\theta_o)x_k + B(\theta_o)u_k,\ y_k = C(\theta_o)x_k \\
& u_k \in \mathcal{U},\ x_k \in \mathcal{X},\ y_k \in \mathcal{Y} \\
& \mathcal{I}_1^{t+N_I}(\theta_o) \succeq {\color{red}\kappa_t} \frac{\gamma n}{2} V''_{\mathsf{app}}(\theta_0)
\end{aligned}$$

Receeding horizon formulation

Cost at time t:

$$C_t = \sum_{k=1}^{F} c_t(x_k, u_k) = \sum_{k=1}^{F} \|y_{k+1} - r_{t+k+1}\|_Q^2 + \sum_{k=1}^{F} \|u_k\|_S^2$$

$$\begin{aligned} \underset{\{u_k\}_{k=1}^F}{\text{minimize}} \quad & C_t \\ \text{subject to} \quad & x_1 = \hat{x}_t, \\ & x_{k+1} = A(\theta_o)x_k + B(\theta_o)u_k,\ y_k = C(\theta_o)x_k \\ & u_k \in \mathcal{U},\ x_k \in \mathcal{X},\ y_k \in \mathcal{Y} \\ & \mathcal{I}_1^{t+N_I}(\theta_o) \succeq \kappa_t \frac{\gamma n}{2} V''_{\text{app}}(\theta_0) \end{aligned}$$

Scaling κ_t monotonically increasing from 0 to 1 at $t = N - N_I$

Receeding horizon formulation

Cost at time t:

$$C_t = \sum_{k=1}^{F} c_t(x_k, u_k) = \sum_{k=1}^{F} \|y_{k+1} - r_{t+k+1}\|_Q^2 + \sum_{k=1}^{F} \|u_k\|_S^2$$

$$\begin{aligned}
\underset{\{u_k\}_{k=1}^F}{\text{minimize}} \quad & C_t \\
\text{subject to} \quad & x_1 = \hat{x}_t, \\
& x_{k+1} = A(\theta_o)x_k + B(\theta_o)u_k,\ y_k = C(\theta_o)x_k \\
& u_k \in \mathcal{U},\ x_k \in \mathcal{X},\ y_k \in \mathcal{Y} \\
& \mathcal{I}_1^{t+N_I}(\theta_o) \succeq \kappa_t \frac{\gamma n}{2} V''_{\text{app}}(\theta_0)
\end{aligned}$$

Scaling κ_t monotonically increasing from 0 to 1 at $t = N - N_I$

Major problems:

- θ_o unknown

Receeding horizon formulation

Cost at time t:

$$C_t = \sum_{k=1}^{F} c_t(x_k, u_k) = \sum_{k=1}^{F} \|y_{k+1} - r_{t+k+1}\|_Q^2 + \sum_{k=1}^{F} \|u_k\|_S^2$$

$$\begin{aligned}
\underset{\{u_k\}_{k=1}^F}{\text{minimize}} \quad & C_t \\
\text{subject to} \quad & x_1 = \hat{x}_t, \\
& x_{k+1} = A(\theta_o)x_k + B(\theta_o)u_k,\ y_k = C(\theta_o)x_k \\
& u_k \in \mathcal{U},\ x_k \in \mathcal{X},\ y_k \in \mathcal{Y} \\
& \mathcal{I}_1^{t+N_I}(\theta_o) \succeq \kappa_t \frac{\gamma n}{2} V''_{\text{app}}(\theta_0)
\end{aligned}$$

Scaling κ_t monotonically increasing from 0 to 1 at $t = N - N_I$

Major problems:

- θ_o unknown
- *Data not stationary & feedback*

Receeding horizon formulation

Cost at time t:

$$C_t = \sum_{k=1}^{F} c_t(x_k, u_k) = \sum_{k=1}^{F} \|y_{k+1} - r_{t+k+1}\|_Q^2 + \sum_{k=1}^{F} \|u_k\|_S^2$$

$$\begin{aligned}
\underset{\{u_k\}_{k=1}^F}{\text{minimize}} \quad & C_t \\
\text{subject to} \quad & x_1 = \hat{x}_t, \\
& x_{k+1} = A(\theta_o)x_k + B(\theta_o)u_k,\ y_k = C(\theta_o)x_k \\
& u_k \in \mathcal{U},\ x_k \in \mathcal{X},\ y_k \in \mathcal{Y} \\
& \mathcal{I}_1^{t+N_I}(\theta_o) \succeq \kappa_t \frac{\gamma n}{2} V''_{\text{app}}(\theta_0)
\end{aligned}$$

Scaling κ_t monotonically increasing from 0 to 1 at $t = N - N_I$

Major problems:

- θ_o unknown
- *Data not stationary & feedback*

  $\Rightarrow$ *Cannot use spectrum as design variable*

Receeding horizon formulation: Implementation

Approximations:

Receeding horizon formulation: Implementation

Approximations:

- Initial estimate $\hat{\theta}$ replaces θ_o

Receeding horizon formulation: Implementation

Approximations:

- Initial estimate $\hat{\theta}$ replaces θ_o
- $I_1^N(\hat{\theta})$ sample approximation of $\mathcal{I}_1^N(\theta_o)$

Receeding horizon formulation: Implementation

Approximations:

- Initial estimate $\hat{\theta}$ replaces θ_o
- $I_1^N(\hat{\theta})$ sample approximation of $\mathcal{I}_1^N(\theta_o)$

Quadratic in design variables $\bar{u} = \left[u_1, \ldots, u_F\right]^T$

Receeding horizon formulation: Implementation

Approximations:

- Initial estimate $\hat{\theta}$ replaces θ_o
- $I_1^N(\hat{\theta})$ sample approximation of $\mathcal{I}_1^N(\theta_o)$

Quadratic in design variables $\bar{u} = \left[u_1, \ldots, u_F\right]^T$

Lifting:

Receeding horizon formulation: Implementation

Approximations:

- Initial estimate $\hat{\theta}$ replaces θ_o
- $I_1^N(\hat{\theta})$ sample approximation of $\mathcal{I}_1^N(\theta_o)$

Quadratic in design variables $\bar{u} = \left[u_1, \ldots, u_F\right]^T$

Lifting: Introduce $U = \bar{u}\bar{u}^T$

Receeding horizon formulation: Implementation

Approximations:

- Initial estimate $\hat{\theta}$ replaces θ_o
- $I_1^N(\hat{\theta})$ sample approximation of $\mathcal{I}_1^N(\theta_o)$

Quadratic in design variables $\bar{u} = \begin{bmatrix} u_1, \ldots, u_F \end{bmatrix}^T$

Lifting: Introduce $U = \bar{u}\bar{u}^T \Leftrightarrow$

$$\begin{bmatrix} U & \bar{u} \\ \bar{u}^T & 1 \end{bmatrix} \succeq 0, \quad \mathrm{rank} \begin{bmatrix} U & \bar{u} \\ \bar{u}^T & 1 \end{bmatrix} = 1$$

Receeding horizon formulation: Implementation

Approximations:

- Initial estimate $\hat{\theta}$ replaces θ_o
- $I_1^N(\hat{\theta})$ sample approximation of $\mathcal{I}_1^N(\theta_o)$

Quadratic in design variables $\bar{u} = \left[u_1, \ldots, u_F\right]^T$

Lifting: Introduce $U = \bar{u}\bar{u}^T \Leftrightarrow$

$$\begin{bmatrix} U & \bar{u} \\ \bar{u}^T & 1 \end{bmatrix} \succeq 0, \quad \operatorname{rank} \begin{bmatrix} U & \bar{u} \\ \bar{u}^T & 1 \end{bmatrix} = 1$$

Convex relaxation: Drop the rank constraint

Receeding horizon formulation: Implementation

Approximations:

- Initial estimate $\hat{\theta}$ replaces θ_o
- $I_1^N(\hat{\theta})$ sample approximation of $\mathcal{I}_1^N(\theta_o)$

Quadratic in design variables $\bar{u} = \left[u_1, \ldots, u_F\right]^T$

Lifting: Introduce $U = \bar{u}\bar{u}^T \Leftrightarrow$

$$\begin{bmatrix} U & \bar{u} \\ \bar{u}^T & 1 \end{bmatrix} \succeq 0, \quad \operatorname{rank} \begin{bmatrix} U & \bar{u} \\ \bar{u}^T & 1 \end{bmatrix} = 1$$

Convex relaxation: Drop the rank constraint

Alternative formulation:

Receeding horizon formulation: Implementation

Approximations:

- Initial estimate $\hat{\theta}$ replaces θ_o
- $I_1^N(\hat{\theta})$ sample approximation of $\mathcal{I}_1^N(\theta_o)$

Quadratic in design variables $\bar{u} = \left[u_1, \ldots, u_F\right]^T$

Lifting: Introduce $U = \bar{u}\bar{u}^T \Leftrightarrow$

$$\begin{bmatrix} U & \bar{u} \\ \bar{u}^T & 1 \end{bmatrix} \succeq 0, \quad \operatorname{rank} \begin{bmatrix} U & \bar{u} \\ \bar{u}^T & 1 \end{bmatrix} = 1$$

Convex relaxation: Drop the rank constraint

Alternative formulation: Minimum time

Receeding horizon formulation: Implementation

Approximations:

- Initial estimate $\hat{\theta}$ replaces θ_o
- $I_1^N(\hat{\theta})$ sample approximation of $\mathcal{I}_1^N(\theta_o)$

Quadratic in design variables $\bar{u} = \left[u_1, \ldots, u_F\right]^T$

Lifting: Introduce $U = \bar{u}\bar{u}^T \Leftrightarrow$

$$\begin{bmatrix} U & \bar{u} \\ \bar{u}^T & 1 \end{bmatrix} \succeq 0, \quad \mathrm{rank} \begin{bmatrix} U & \bar{u} \\ \bar{u}^T & 1 \end{bmatrix} = 1$$

Convex relaxation: Drop the rank constraint

Alternative formulation: Minimum time (maximize κ_t)

Receeding horizon formulation: Implementation

Approximations:

- Initial estimate $\hat{\theta}$ replaces θ_o
- $I_1^N(\hat{\theta})$ sample approximation of $\mathcal{I}_1^N(\theta_o)$

Quadratic in design variables $\bar{u} = \left[u_1, \ldots, u_F\right]^T$

Lifting: Introduce $U = \bar{u}\bar{u}^T \Leftrightarrow$

$$\begin{bmatrix} U & \bar{u} \\ \bar{u}^T & 1 \end{bmatrix} \succeq 0, \quad \text{rank} \begin{bmatrix} U & \bar{u} \\ \bar{u}^T & 1 \end{bmatrix} = 1$$

Convex relaxation: Drop the rank constraint

Alternative formulation: Minimum time (maximize κ_t)

MPC-X: Model Predictive Control with eXperimental constraints

Receeding horizon formulation: Alternative approaches

$$y_t = \sum_{k=1}^{n_b} \theta_k u_{t-k} + e_t = \theta^T \phi_t + e_t,$$

Receeding horizon formulation: Alternative approaches

$$y_t = \sum_{k=1}^{n_b} \theta_k u_{t-k} + e_t = \theta^T \phi_t + e_t,\ \phi_t = \begin{bmatrix} u_{t-1} & \cdots & u_{t-n_b} \end{bmatrix}^T$$

Receeding horizon formulation: Alternative approaches

$$y_t = \sum_{k=1}^{n_b} \theta_k u_{t-k} + e_t = \theta^T \phi_t + e_t,\ \phi_t = \begin{bmatrix} u_{t-1} & \cdots & u_{t-n_b} \end{bmatrix}^T$$

Persistence of excitation condition: $\sum_{k=t-P}^{t+1+F} \phi_k \phi_k^T \succeq \rho I$

Receeding horizon formulation: Alternative approaches

$$y_t = \sum_{k=1}^{n_b} \theta_k u_{t-k} + e_t = \theta^T \phi_t + e_t,\ \phi_t = \begin{bmatrix} u_{t-1} & \cdots & u_{t-n_b} \end{bmatrix}^T$$

Persistence of excitation condition: $\sum_{k=t-P}^{t+1+F} \phi_k \phi_k^T \succeq \rho I$

- MPCI (Genceli and Nikolaou (1996)): $P = 0$
- Multiobjective MPC with identification (Aggelogiannaki and Sarimveis (2006)): $P = 0$
- Dual control by information maximization (Rathhouský and Havlena (2011)): $P = 0$
- PE-MPC (Marafioti (2010)): $F = 0$

Receeding horizon formulation: Alternative approaches

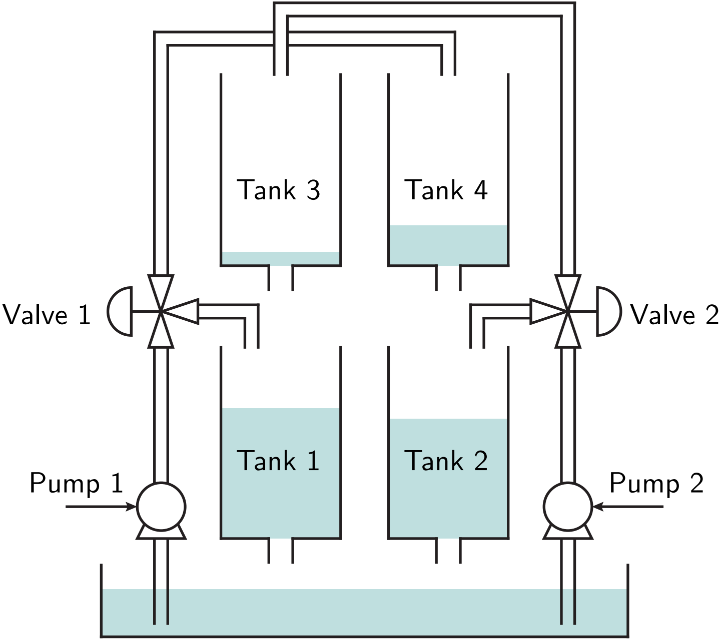
$$y_t = \sum_{k=1}^{n_b} \theta_k u_{t-k} + e_t = \theta^T \phi_t + e_t,\ \phi_t = \begin{bmatrix} u_{t-1} & \cdots & u_{t-n_b} \end{bmatrix}^T$$

Persistence of excitation condition: $\sum_{k=t-P}^{t+1+F} \phi_k \phi_k^T \succeq \rho I$

- MPCI (Genceli and Nikolaou (1996)): $P = 0$
- Multiobjective MPC with identification (Aggelogiannaki and Sarimveis (2006)): $P = 0$
- Dual control by information maximization (Rathhouský and Havlena (2011)): $P = 0$
- PE-MPC (Marafioti (2010)): $F = 0$

Do not take application into account explicitly

Receeding horizon formulation: Simulation study



Receeding horizon formulation: Simulation study

$$\begin{cases} x_{t+1} = \begin{bmatrix} \theta_3 & \theta_4 \\ 1 & 0 \end{bmatrix} x_t + \begin{bmatrix} 4.5 \\ 0 \end{bmatrix} u_t, \\ \end{cases}$$

Receeding horizon formulation: Simulation study

$$\begin{cases} x_{t+1} = \begin{bmatrix} \theta_3 & \theta_4 \\ 1 & 0 \end{bmatrix} x_t + \begin{bmatrix} 4.5 \\ 0 \end{bmatrix} u_t, \\ y_t = \begin{bmatrix} \theta_1 & \theta_2 \end{bmatrix} x_t + e_t \end{cases}$$

Receeding horizon formulation: Simulation study

$$\begin{cases} x_{t+1} = \begin{bmatrix} \theta_3 & \theta_4 \\ 1 & 0 \end{bmatrix} x_t + \begin{bmatrix} 4.5 \\ 0 \end{bmatrix} u_t, \\ y_t = \begin{bmatrix} \theta_1 & \theta_2 \end{bmatrix} x_t + e_t \quad \text{lower tank level} \end{cases}$$

Receeding horizon formulation: Simulation study

$$\begin{cases} x_{t+1} = \begin{bmatrix} \theta_3 & \theta_4 \\ 1 & 0 \end{bmatrix} x_t + \begin{bmatrix} 4.5 \\ 0 \end{bmatrix} u_t, \\ y_t = \begin{bmatrix} \theta_1 & \theta_2 \end{bmatrix} x_t + e_t \quad \text{lower tank level} \end{cases}$$

$\theta_o = \begin{bmatrix} 0.12 & 0.059 & 0.74 & -0.14 \end{bmatrix}^T$ Noise var.: 0.01.
$N = 200,\ F = 5$

Receeding horizon formulation: Simulation study

$$\begin{cases} x_{t+1} = \begin{bmatrix} \theta_3 & \theta_4 \\ 1 & 0 \end{bmatrix} x_t + \begin{bmatrix} 4.5 \\ 0 \end{bmatrix} u_t, \\ \quad y_t = \begin{bmatrix} \theta_1 & \theta_2 \end{bmatrix} x_t + e_t \quad \text{lower tank level} \end{cases}$$

$\theta_o = \begin{bmatrix} 0.12 & 0.059 & 0.74 & -0.14 \end{bmatrix}^T$ Noise var.: 0.01.
$N = 200$, $F = 5$

Performance degradation cost: $V_{\mathsf{app}}(\theta) = \sum_{t=1}^{T} \|y_t(\theta_o) - y_t(\theta)\|_2^2$

Receeding horizon formulation: Simulation study

$$\begin{cases} x_{t+1} = \begin{bmatrix} \theta_3 & \theta_4 \\ 1 & 0 \end{bmatrix} x_t + \begin{bmatrix} 4.5 \\ 0 \end{bmatrix} u_t, \\ y_t = \begin{bmatrix} \theta_1 & \theta_2 \end{bmatrix} x_t + e_t \quad \text{lower tank level} \end{cases}$$

$\theta_o = \begin{bmatrix} 0.12 & 0.059 & 0.74 & -0.14 \end{bmatrix}^T$ Noise var.: 0.01.
$N = 200, \;\; F = 5$

Performance degradation cost: $V_{\mathsf{app}}(\theta) = \sum_{t=1}^{T} \|y_t(\theta_o) - y_t(\theta)\|_2^2$

PE-MPC: $\rho = 0.5$, $P = 5$, $F = 0$

Receeding horizon formulation: Simulation study

$$\begin{cases} x_{t+1} = \begin{bmatrix} \theta_3 & \theta_4 \\ 1 & 0 \end{bmatrix} x_t + \begin{bmatrix} 4.5 \\ 0 \end{bmatrix} u_t, \\ \quad y_t = \begin{bmatrix} \theta_1 & \theta_2 \end{bmatrix} x_t + e_t \quad \text{lower tank level} \end{cases}$$

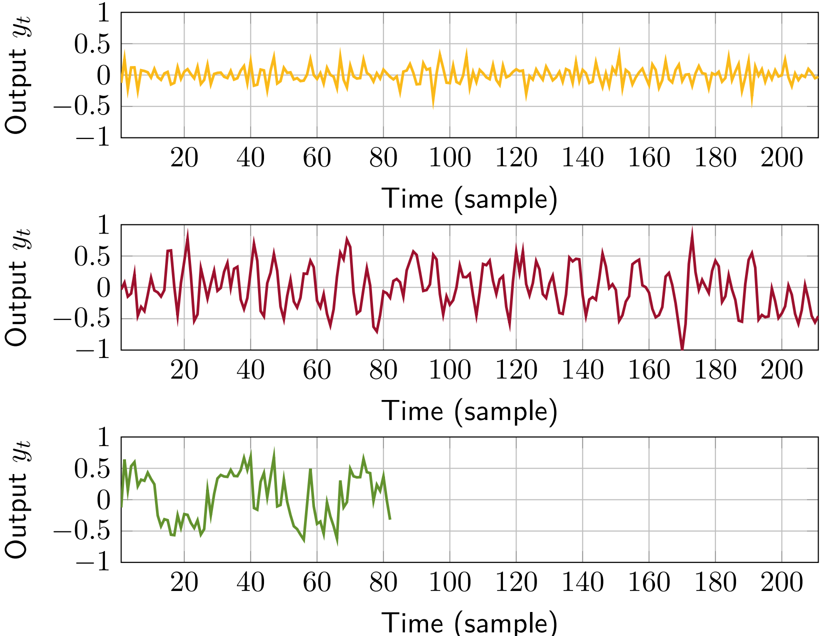
$\theta_o = \begin{bmatrix} 0.12 & 0.059 & 0.74 & -0.14 \end{bmatrix}^T$ Noise var.: 0.01.
$N = 200, \ F = 5$

Performance degradation cost: $V_{\mathsf{app}}(\theta) = \sum_{t=1}^{T} \|y_t(\theta_o) - y_t(\theta)\|_2^2$

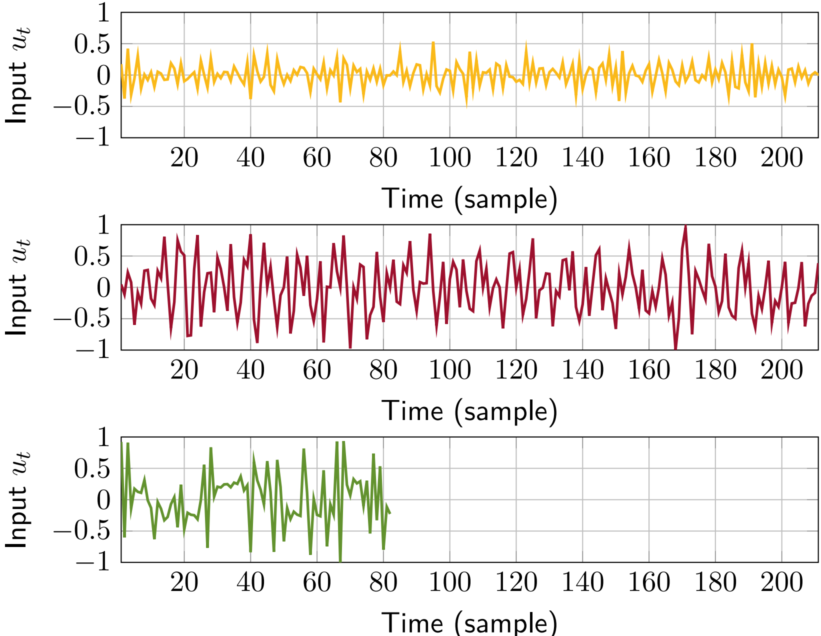
PE-MPC: $\rho = 0.5$, $P = 5$, $F = 0$

MPC-X: Minimum time formulation

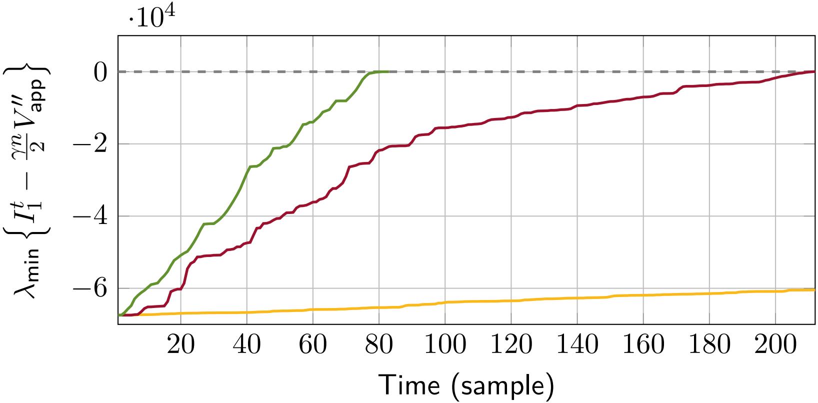
Receeding horizon formulation: Simulation study



Receeding horizon formulation: Simulation study



Receeding horizon formulation: Simulation study

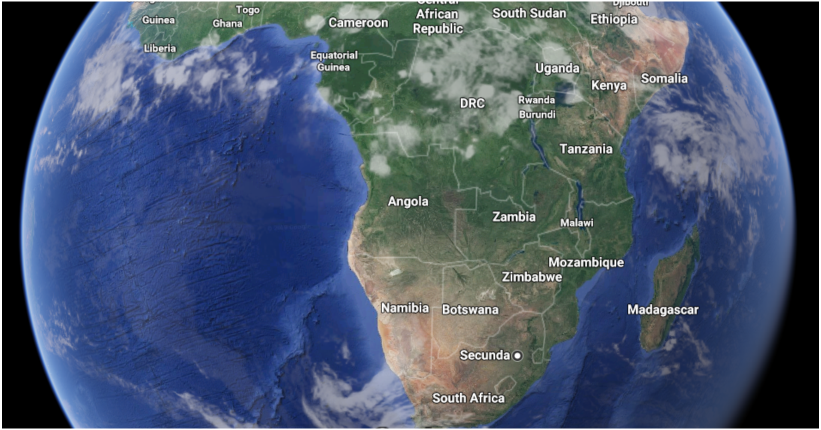


- Regular MPC (—),
- PE-MPC with $\rho = 0.5$ (—)
- Minimum time MPC-X (—)

Receeding horizon formulation: Simulation study

Algorithm	$\mathrm{Var}\, u$	$\mathrm{Var}\, y$	N
MPC-X, minimum time	0.203	0.146	82
PE-MPC, $\rho = 0.5$	0.175	0.120	211

MPC-X experimental study: Let's travel

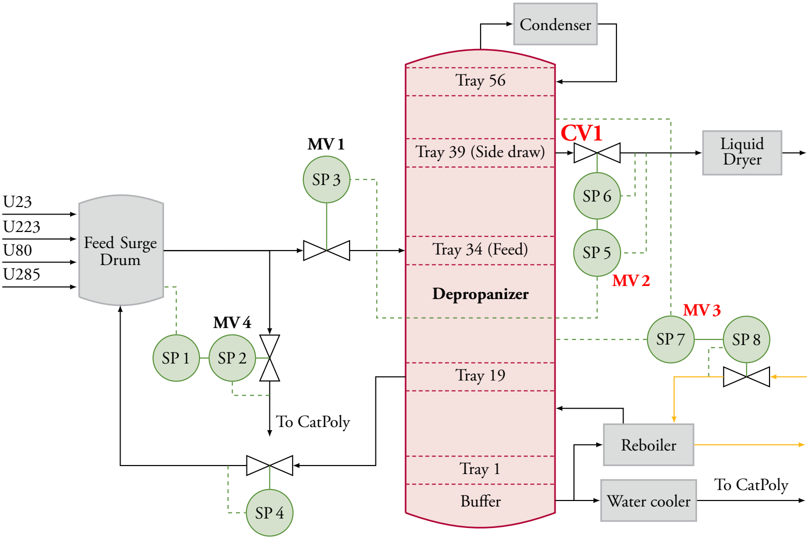


Secunda, South Africa

SASOL Synthetic Fuels Refinery

Synfuels Catalytic Cracker (SCC)

Depropanizer



Depropanizer

Separates three-carbon hydrocarbons (C_3) from four carbon hydrocarbons (C_4)

Objective: Set point for CV1=C_4 concentration in side draw

MV2: Side draw to feed ratio

MV3: Column differential pressure

Depropanizer

Separates three-carbon hydrocarbons (C_3) from four carbon hydrocarbons (C_4)

Objective: Set point for CV1$=C_4$ concentration in side draw

MV2: Side draw to feed ratio

MV3: Column differential pressure

Performance drop obtained by changing poles of model

Depropanizer

Separates three-carbon hydrocarbons (C_3) from four carbon hydrocarbons (C_4)

Objective: Set point for CV1$=C_4$ concentration in side draw

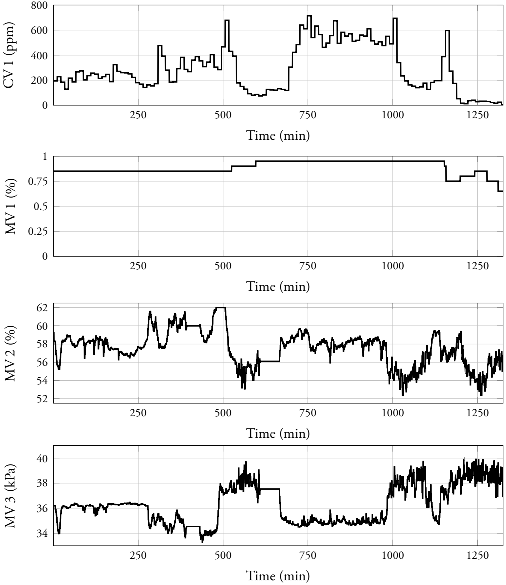
MV2: Side draw to feed ratio

MV3: Column differential pressure

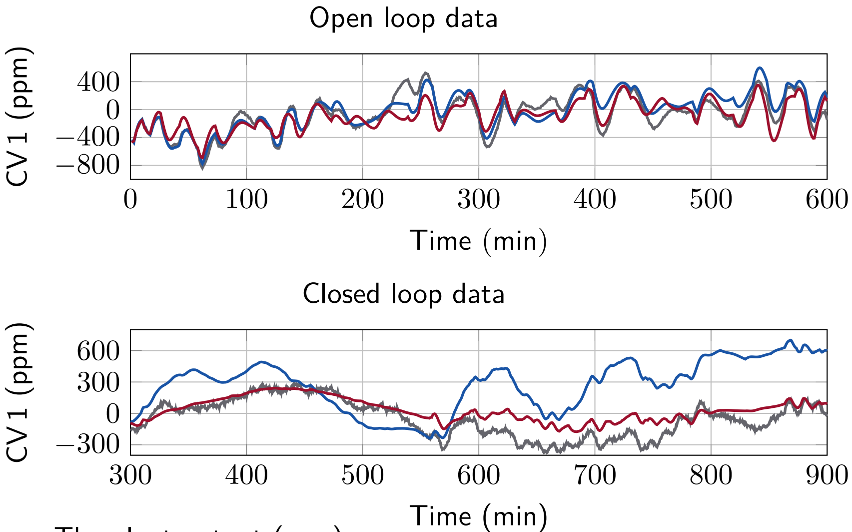
Performance drop obtained by changing poles of model

Excitation level manually controlled

Depropanizer: MPC-X experiment



Depropanizer: Model fit



- The plant output (——)
- Model identified in open-loop (——)
- Model identified in closed-loop MPC-X experiment (——)

Depropanizer: Closed loop performance

	Variance	
Model	CV 1	MV 5
Before MPC-X	95×10^3	34×10^7
After MPC-X model update	36×10^3	37×10^7

MV 5 $= C_4$ content in the feed

MPC-X: Summary

- Sample version of Information Application Inequality added as a matrix inequality constraint in MPC

MPC-X: Summary

- Sample version of Information Application Inequality added as a matrix inequality constraint in MPC
- Convex relaxation

MPC-X: Summary

- Sample version of Information Application Inequality added as a matrix inequality constraint in MPC
- Convex relaxation
- Current limitation: Output error models (disturbances not modeled)

Outline

Outline

What have we learnt?

- A framework for experiment design where the application is taken into account

What have we learnt?

- A framework for experiment design where the application is taken into account
- The optimal experiment matches the identification criterion to the performance degradation using parsimonious excitation
  (The let sleeping dogs lie paradigm)

What have we learnt?

- A framework for experiment design where the application is taken into account
- The optimal experiment matches the identification criterion to the performance degradation using parsimonious excitation

  (The let sleeping dogs lie paradigm)
- Simplifies the identification problem

What have we learnt?

- A framework for experiment design where the application is taken into account
- The optimal experiment matches the identification criterion to the performance degradation using parsimonious excitation
  (The let sleeping dogs lie paradigm)
- Simplifies the identification problem
- Active application oriented learning practical implementation

What have we learnt?

- A framework for experiment design where the application is taken into account
- The optimal experiment matches the identification criterion to the performance degradation using parsimonious excitation
  (The let sleeping dogs lie paradigm)
- Simplifies the identification problem
- Active application oriented learning practical implementation
- Adding the Information Application Inequality to an optimal control problem leads to dual control

Acknowledgements

- Former PhD-students: Kristian Lindqvist, Henrik Jansson, Jonas Mårtensson, Märta Barenthin, Christian Larsson, Afrooz Ebadat, Mariette Annergren

- Xavier Bombois, László Gerencsér, Ali Mesbah, Per-Erik Modén, Cristian Rojas, Paul Van den Hof, Bo Wahlberg

THANK YOU!!!