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 Institute of Technology, Stockholm

DDCLS’18
May 26, 2018
The problem

There is abundant evidence in industrial practice that when modeling for control is not based on criteria related to the actual end use, the results can sometimes be quite disappointing.

Main objective

Predictable performance
Ogunnaike 1996:

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

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
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
An application example: MPC of a DC-motor

- **Input:** Voltage $V$
- **Output:** Angle $\phi_L$
- **Model parameters $\theta$:** Resistance $R$, Moment of inertia $J_L$, Elasticity $K$, ...
- **True parameters:** $\theta_o$
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 \( \theta \) 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 $\theta$ 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 $\theta$ 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 $\theta$ used in MPC

\[
E_{\text{app}} = \left\{ \theta : V_{\text{app}}(\theta) \leq \frac{1}{\gamma} \right\} \quad (\gamma = \text{accuracy})
\]
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 $\theta$ used in MPC

![Graph showing ideal and actual responses]

Performance degradation / Set of acceptable models

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

$$\mathcal{E}_{\text{app}} = \left\{ \theta : V_{\text{app}}(\theta) \leq \frac{1}{\gamma} \right\} \quad (\gamma = \text{accuracy})$$
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 $\mathcal{E}_{\text{app}}$: 
Performance degradation for application: $V_{\text{app}}(\theta)$
Performance degradation for application: $V_{\text{app}}(\theta)$

Set of acceptable models: $\mathcal{E}_{\text{app}} = \left\{ \theta : V_{\text{app}}(\theta) \leq \frac{1}{\gamma} \right\}$
Summary of concepts

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

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

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

**Application oriented experiment design**

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

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

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

An optimal experiment design problem
System identification

Random innovations with variance $\lambda_e$
System identification

Random innovations with variance $\lambda_e$

Stationary signals
System identification

- Random innovations with variance $\lambda_e$
- Stationary signals
- True system in the model set: $S_o \leftrightarrow \theta_o$ (to be relaxed later)
Random innovations with variance $\lambda_e$

Stationary signals

True system in the model set: $S_o \Leftrightarrow \theta_o$ (to be relaxed later)

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

Stationary signals

True system in the model set: $S_o \Leftrightarrow \theta_o$ (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

\[ V_{id}(\theta) = E[\epsilon_t^2(\theta)] - \lambda_e \]
Application oriented experiment design

- $V_{id}(\theta) = E[\varepsilon_t^2(\theta)] - \lambda_e$
- High accuracy $\gamma$ (implies large sample size $N$)
Application oriented experiment design

- $V_{id}(\theta) = E[\varepsilon_t^2(\theta)] - \lambda_e$
- High accuracy $\gamma$ (implies large sample size $N$)
- $\sqrt{N} \left( \hat{\theta}_N - \theta_o \right) \sim \text{AsN} \left( 0, 2\lambda_e V_{id}'(\theta_o)^{-1} \right)$
Application oriented experiment design

- \( V_{id}(\theta) = E[\varepsilon^2_t(\theta)] - \lambda_e \)
- High accuracy \( \gamma \) (implies large sample size \( N \))
- \( \sqrt{N} \left( \hat{\theta}_N - \theta_o \right) \sim \text{AsN} (0, 2\lambda_e V''_{id}(\theta_o)^{-1}) \)
- Cannot guarantee \( \hat{\theta}_N \in E_{app} \)
Application oriented experiment design

- \( V_{id}(\theta) = E[\varepsilon_t^2(\theta)] - \lambda_e \)
- High accuracy \( \gamma \) (implies large sample size \( N \))
- \( \sqrt{N} \left( \hat{\theta}_N - \theta_o \right) \sim \text{AsN} \left( 0, 2\lambda_e V_{id}''(\theta_o)^{-1} \right) \)
- Cannot guarantee \( \hat{\theta}_N \in E_{app} \)
- Relaxation: Use standard (asymptotic) confidence ellipsoid \( E_{id} \)
Application oriented experiment design

- $V_{id}(\theta) = \mathbb{E}[\varepsilon^2_t(\theta)] - \lambda_e$
- High accuracy $\gamma$ (implies large sample size $N$)
- $\sqrt{N} \left( \hat{\theta}_N - \theta_o \right) \sim \text{AsN} \left( 0, 2\lambda_e V''_{id}(\theta_o)^{-1} \right)$
- Cannot guarantee $\hat{\theta}_N \in \mathcal{E}_{\text{app}}$
- Relaxation: Use standard (asymptotic) confidence ellipsoid $\mathcal{E}_{id}$
Application oriented experiment design

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

Probability($\hat{\theta}_N \in \mathcal{E}_{id}$) $\approx \alpha$ (e.g. 99%)
Application oriented experiment design

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

Probability($\hat{\theta}_N \in \mathcal{E}_{id}$) $\approx \alpha$ (e.g. 99%)
Application oriented experiment design

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

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

Application oriented experiment design

\[
\min \; N E[u_t^2] \\
s.t. \; \mathcal{E}_{id} \subseteq \mathcal{E}_{app} \subseteq \mathbb{R}^n
\]
An alternative formulation

Application oriented experiment design

\[
\begin{align*}
\min & \quad NE[u_t^2] \\
\text{s.t.} & \quad E_{id} \subseteq E_{app} \subseteq \mathbb{R}^n
\end{align*}
\]
An alternative formulation

Application oriented experiment design

\[
\begin{align*}
& \min \ N E[u_t^2] \\
& \text{s.t. } \mathcal{E}_{id} \subseteq \mathcal{E}_{app} \subseteq \mathbb{R}^n
\end{align*}
\]

can approximately be formulated as

\[
\begin{align*}
& \min \ N E[u_t^2] \\
& \text{s.t. } NV_{id}(\theta) \geq \lambda e \gamma n V_{app}(\theta), \ \forall \theta \in \mathcal{E}_{app}
\end{align*}
\]
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
True system: \( y_t = G_0(q)u_t + e_t, \) open loop
Output error models

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

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

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

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

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

\( V_{id}(\theta) = E[\varepsilon_t^2(\theta)] - \lambda_e \)
Output error models

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

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

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

\[ V_{id}(\theta) = E[\varepsilon_t^2(\theta)] - \lambda_e \]
\[ = E[((G_o(q) - G(q, \theta))u_t)^2] + E[e_t^2] - \lambda_e \]
Output error models

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

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

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

\[
V_{id}(\theta) = E[\varepsilon_t^2(\theta)] - \lambda_e \\
= E[(G_o(q) - G(q, \theta))u_t]^2 + E[e_t^2] - \lambda_e \\
= \frac{1}{2\pi} \int_{-\pi}^{\pi} \Phi_{u}^{id}(e^{j\omega}) \left| G_o(e^{j\omega}) - G(e^{j\omega}, \theta) \right|^2 \, d\omega
\]
Output error models

Application oriented experiment design

\[
\min \text{NE}[u^2_t]
\]

s.t.

\[
\text{NV}_{id}(\theta) \geq \lambda e \gamma n V_{\text{app}}(\theta), \forall \theta \in \mathcal{E}_{\text{app}}
\]

\[
\frac{1}{2\pi} \int_{-\pi}^{\pi} N \Phi_{id}^{ij}(e^{j\omega}) |G_o(e^{j\omega}) - G(e^{j\omega}, \theta)|^2 \, d\omega
\]
Output error models

Application oriented experiment design

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

s.t. $$\frac{1}{2\pi} \int_{-\pi}^{\pi} N \Phi_{id}^{u}(e^{j\omega}) \left| G_o(e^{j\omega}) - G(e^{j\omega}, \theta) \right|^2 \, d\omega \geq \lambda e^{\gamma n} V_{app}(\theta), \ \forall \theta \in \mathcal{E}_{app}$$
Minimization with respect to energy density spectrum $N \Phi_u$
Output error models

Application oriented experiment design

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

s.t.

\[
\begin{align*}
NV_{id}(\theta) & \geq \lambda \gamma nV_{app}(\theta), \quad \forall \theta \in \mathcal{E}_{app} \\
\frac{1}{2\pi} \int_{-\pi}^{\pi} N\Phi_i^d(e^{j\omega}) |G_o(e^{j\omega}) - G(e^{j\omega}, \theta)|^2 \, d\omega & \geq \lambda \gamma nV_{app}(\theta), \quad \forall \theta \in \mathcal{E}_{app}
\end{align*}
\]

- Minimization with respect to energy density spectrum \( N\Phi_i^d \)
- Optimization tries to achieve

\[
NV_{id}(\theta) = \lambda \gamma nV_{app}(\theta), \quad \forall \theta \in \mathcal{E}_{app}
\]
Output error models

**Application oriented experiment design**

\[
\min N E[u_t^2] = \frac{1}{2\pi} \int_{-\pi}^{\pi} N \Phi_{ud}^t(e^{j\omega}) \, d\omega
\]

s.t.

\[
\begin{aligned}
NV_{id}(\theta) & \geq \lambda_e \gamma n V_{app}(\theta), \ \forall \theta \in \mathcal{E}_{app} \\
\frac{1}{2\pi} \int_{-\pi}^{\pi} N \Phi_{ud}^t(e^{j\omega}) |G_o(e^{j\omega}) - G(e^{j\omega},\theta)|^2 \, d\omega
\end{aligned}
\]

- Minimization with respect to energy density spectrum \(N \Phi_{ud}^t\)
- Optimization tries to achieve

\[
NV_{id}(\theta) = \lambda_e \gamma n V_{app}(\theta), \ \forall \theta \in \mathcal{E}_{app}
\]

*Identification cost matched to performance degradation*
Controller $C = C(G)$, $G$ output error model
Model Reference Control

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

Desired sensitivity function: $S_\xi$

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

Desired sensitivity function: $S_\xi$

Achieved sensitivity function: $S(G) = \frac{1}{1 + C(G)G_o}$

Performance degradation: $V_{app}(G) := \left\| \frac{S(G) - S_\xi}{S_\xi} \right\|^2_2$
\[
\begin{align*}
\min NE[u^2(t)] \\
\text{s.t. } NV_{id}(\theta) \geq \gamma \lambda_e n \ V_{app}(\theta)
\end{align*}
\]
Model Reference Control

\[ \min NE[u^2(t)] \]
\[ \text{s.t. } NV_{id}(\theta) \geq \gamma \lambda e n V_{app}(\theta) \]

- Matching condition: \( NV_{id}(\theta) = \gamma \lambda e n V_{app}(\theta) \)

Experimental conditions during identification should be a scaled version of the desired operating conditions!
Model Reference Control

\[ \min N E[u^2(t)] \]
\[ \text{s.t. } NV_{id}(\theta) \geq \gamma \lambda e_n V_{app}(\theta) \]

- Matching condition: \( NV_{id}(\theta) = \gamma \lambda e_n V_{app}(\theta) \)

\[ \Rightarrow N \Phi_u^{id} = \gamma \lambda e_n \Phi_u^{desired} \]
$$\min NE[u^2(t)]$$

s.t. $NV_{id}(\theta) \geq \gamma \lambda e_n V_{app}(\theta)$

- Matching condition: $NV_{id}(\theta) = \gamma \lambda e_n V_{app}(\theta)$

$\Rightarrow NV_{id} = \gamma \lambda e_n V_{desired}$

- **Experimental conditions during identification should be a scaled version of the desired operating conditions!**
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
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: \( V_{\text{app}}(\theta) = (\sum \theta_k - \sum \theta^o_k)^2 \)
Static gain estimation

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

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

\( V_{\text{app}}(\theta) = 0 \)
Static gain estimation

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

Performance degradation: \[ V_{\text{app}}(\theta) = (\sum \theta_k - \sum \theta^o_k)^2 \]

\[ V_{\text{app}}(\theta) = 0 \]

Optimal input: \( u_t = u \) (constant)
Static gain estimation

<table>
<thead>
<tr>
<th>Model order:</th>
<th>low</th>
<th>true</th>
<th>high</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy:</td>
<td></td>
<td><strong>good</strong></td>
<td></td>
</tr>
</tbody>
</table>

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

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

\[ V_{\text{app}}(\theta) = 0 \]

- Optimal input: \( u_t = u \) (constant)
Static gain estimation

<table>
<thead>
<tr>
<th>Model order:</th>
<th>low</th>
<th>true</th>
<th>high</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy:</td>
<td></td>
<td><strong>good</strong></td>
<td></td>
</tr>
</tbody>
</table>

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

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

\[ V_{\text{app}}(\theta) = 0 \]

- Optimal input: \( u_t = u \) (constant) \( \Rightarrow \) \( y_t = (\sum_k \theta_k^o) \ u + e_t \)
Static gain estimation

<table>
<thead>
<tr>
<th>Model order:</th>
<th>low</th>
<th>true</th>
<th>high</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy:</td>
<td></td>
<td>good</td>
<td></td>
</tr>
</tbody>
</table>

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

Performance degradation: \( V_{\text{app}}(\theta) = (\sum \theta_k - \sum \theta_k^0)^2 \)

\[ V_{\text{app}}(\theta) = 0 \]

- Optimal input: \( u_t = u \) (constant) \( \Rightarrow y_t = (\sum_k \theta_k^0) u + e_t \)
- Property of interest visible
Static gain estimation

<table>
<thead>
<tr>
<th>Model order:</th>
<th>low</th>
<th>true</th>
<th>high</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy:</td>
<td></td>
<td><strong>good</strong></td>
<td></td>
</tr>
</tbody>
</table>

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

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

\[ V_{\text{app}}(\theta) = 0 \]

- Optimal input: \( u_t = u \) (constant) \( \Rightarrow \) \( y_t = (\sum_k \theta^o_k) u + e_t \)
- Property of interest visible
- No other system property visible (due to min energy crit.)
Static gain estimation

<table>
<thead>
<tr>
<th>Model order:</th>
<th>low</th>
<th>true</th>
<th>high</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy:</td>
<td></td>
<td><em>good</em></td>
<td></td>
</tr>
</tbody>
</table>

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

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

\[ V_{\text{app}}(\theta) = 0 \]

- Optimal input: \( u_t = u \) (constant) \( \Rightarrow y_t = (\sum_k \theta^o_k) u + e_t \)
- 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

<table>
<thead>
<tr>
<th>Model order:</th>
<th>low</th>
<th>true</th>
<th>high</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy:</td>
<td></td>
<td><strong>good</strong></td>
<td></td>
</tr>
</tbody>
</table>

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

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

\[ V_{app}(\theta) = 0 \]

- Optimal input: \( u_t = u \) (constant) \( \Rightarrow \) \( y_t = (\sum_k \theta^o_k) u + e_t \)
- 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

<table>
<thead>
<tr>
<th>Model order:</th>
<th>low</th>
<th>true</th>
<th>high</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy:</td>
<td><strong>good</strong></td>
<td><strong>good</strong></td>
<td></td>
</tr>
</tbody>
</table>

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

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

\[ V_{\text{app}}(\theta) = 0 \]

- Optimal input: \( u_t = u \) (constant) \( \Rightarrow \) \( y_t = (\sum_k \theta^o_k) u + e_t \)
- 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

<table>
<thead>
<tr>
<th>Model order:</th>
<th>low</th>
<th>true</th>
<th>high</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy:</td>
<td>good</td>
<td>good</td>
<td>good</td>
</tr>
</tbody>
</table>

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

Performance degradation: \[ V_{\text{app}}(\theta) = (\sum \theta_k - \sum \theta^o_k)^2 \]

\[ V_{\text{app}}(\theta) = 0 \]

- Optimal input: \( u_t = u \) (constant) \( \Rightarrow \) \( y_t = (\sum_k \theta^o_k) u + e_t \)
- 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

Aims at achieving

\[ N \, V_{id}(\theta) = \lambda_e \, \gamma \, n \, V_{app}(\theta) \]

using minimum energy
Application oriented experiment design

Aims at achieving

\[ N \, V_{id}(\theta) = \lambda_e \, \gamma \, n \, V_{app}(\theta) \]

using minimum energy

Typically leads to similar operating conditions during identification as the ones desired during the application
Application oriented experiment design

Aims at achieving

\[ N \, V_{id}(\theta) = \lambda e^{-\gamma} n \, V_{app}(\theta) \]

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

Aims at achieving

\[ N V_{id}(\theta) = \lambda_e \gamma n V_{app}(\theta) \]

using minimum energy

- Typically leads to similar operating conditions during identification as the ones desired during the application
- To achieve this requires *parsimonious excitation*:
  1. System properties important to the application should be visible in the data
Application oriented experiment design: Summary

Application oriented experiment design

Aims at achieving

\[ N \, V_{id}(\theta) = \lambda_e \, \gamma \, n \, V_{app}(\theta) \]

using minimum energy

- Typically leads to similar operating conditions during identification as the ones desired during the application
- To achieve this requires *parsimonious excitation*:
  1. System properties important to the application should be visible in the data
  2. System properties not important to the application should not be visible in the data, unless necessary for 1).
     (The let sleeping dogs lie paradigm)
Application oriented experiment design

Aims at achieving

\[ N \ V_{id}(\theta) = \lambda e \gamma n \ V_{app}(\theta) \]

using minimum energy

- Typically leads to similar operating conditions during identification as the ones desired during the application.
- To achieve this requires **parsimonious excitation**:
  1. System properties important to the application should be visible in the data.
  2. 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

Aims at achieving

\[ N V_{id}(\theta) = \lambda_e \gamma n V_{app}(\theta) \]

using minimum energy

- Typically leads to similar operating conditions during identification as the ones desired during the application
- To achieve this requires \textit{parsimonious excitation}:
  1. System properties important to the application should be visible in the data
  2. System properties not important to the application should not be visible in the data, unless necessary for 1). (The let sleeping dogs lie paradigm)
- As a result, the entire system may not have to be identified!
  - Choice of model structure less critical
The design constraint can be written as

$$\mathcal{E}_{id} \subseteq \mathcal{E}_{app},$$
The design constraint can be written as

\[ \mathcal{E}_{id} \subseteq \mathcal{E}_{app} \text{, or} \]
\[ N \ V_{id}(\theta) \geq \lambda e \gamma n \ V_{app}(\theta), \ \forall \theta \in \mathcal{E}_{app} \]
The design constraint can be written as

\[ E_{id} \subseteq E_{app}, \quad \text{or} \quad N V_{id}(\theta) \geq \lambda e \gamma n V_{app}(\theta), \quad \forall \theta \in E_{app} \]

but also as

\[ \mathcal{I}_1^N(\theta_o) \leq \frac{\gamma}{2n} V_{app}''(\theta_o) \]
The design constraint can be written as

\[ \mathcal{E}_{id} \subseteq \mathcal{E}_{app}, \text{ or} \]
\[ N V_{id}(\theta) \geq \lambda e \gamma n V_{app}(\theta), \ \forall \theta \in \mathcal{E}_{app} \]

but also as

\[ I_{1}^{N}(\theta_{o}) \geq \frac{\gamma}{2n} V''_{app}(\theta_{o}) \]

where \( I_{1}^{N}(\theta_{o}) \) is the Fisher information
The design constraint can be written as

\[ \mathcal{E}_{id} \subseteq \mathcal{E}_{app}, \text{ or} \]
\[ N \, V_{id}(\theta) \geq \lambda e \gamma n \, V_{app}(\theta), \quad \forall \theta \in \mathcal{E}_{app} \]

but also as

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

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

\[ = \frac{NV''_{id}(\theta_o)}{2\lambda_e} \]
The design constraint can be written as

\[ \mathcal{E}_{id} \subseteq \mathcal{E}_{app}, \text{ or} \]

\[ NV_{id}(\theta) \geq \lambda e \gamma n V_{app}(\theta), \quad \forall \theta \in \mathcal{E}_{app} \]

but also as

\[ \mathcal{I}_1^N(\theta_o) \geq \frac{\gamma}{2n} V''_{app}(\theta_o) \]

where \( \mathcal{I}_1^N(\theta_o) \) is the Fisher information \( = \frac{NV_{id}'''(\theta_o)}{2\lambda e} \)

_The Information Application Inequality_
The design constraint can be written as

\[ \mathcal{E}_{id} \subseteq \mathcal{E}_{app}, \text{ or} \]
\[ NV_{id}(\theta) \geq \lambda e \gamma n V_{app}(\theta), \forall \theta \in \mathcal{E}_{app} \]

but also as

\[ \mathcal{I}_{1}^{N}(\theta_{o}) \succeq \frac{\gamma}{2n} V''_{app}(\theta_{o}) \]

where \( \mathcal{I}_{1}^{N}(\theta_{o}) \) is the Fisher information

\[ = \frac{NV''_{id}(\theta_{o})}{2\lambda e} \]

The Information Application Inequality

Recall: \( V_{id} \) linear in the input spectrum
The design constraint can be written as

$$\mathcal{E}_{id} \subseteq \mathcal{E}_{app}, \text{ or } \quad NV_{id}(\theta) \geq \lambda_e \gamma n \ V_{app}(\theta), \ \forall \theta \in \mathcal{E}_{app}$$

but also as

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

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

$$= \frac{NV_{id}(\theta_o)}{2\lambda_e}$$

**The Information Application Inequality**

Recall: $$V_{id}$$ linear in the input spectrum

*Information Application Inequality is an LMI in the input spectrum*
min \quad NE[u_t^2]

\text{s.t.} \quad I_1^N \geq \frac{\gamma n}{2} V_{app}
Computations

\[
\begin{align*}
\min & \quad NE[u_t^2] \\
\text{Input spectrum} & \\
\text{s.t.} & \quad \mathcal{I}_1^N \geq \frac{\gamma n}{2} V''_{\text{app}}
\end{align*}
\]
\[
\min_{\text{Input spectrum}} \quad NE[u_t^2] \\
\text{s.t.} \quad \mathcal{I}_1^N \succeq \frac{\gamma n}{2} V_{\text{app}}
\]

- Semi-Definite Program (SDP) in input spectrum \(\Phi_{u,u}^{id}\)
Computations

\[
\begin{align*}
\min_{\text{Input spectrum}} & \quad NE[u_i^2] \\
\text{s.t.} & \quad \mathcal{I}_1^N \geq \frac{\gamma n}{2} V_{app}
\end{align*}
\]

- Semi-Definite Program (SDP) in input spectrum \( \Phi_{id}^u \)
- Optimal experiment design \( \Rightarrow \) Input spectrum \( \Phi_{id}^u \)
min \limits_{\text{Input spectrum}} \quad NE[u_i^2] \\
\text{s.t.} \quad \mathcal{I}_1^N \geq \frac{\gamma n}{2} V_{\text{app}} \\

- Semi-Definite Program (SDP) in input spectrum $\Phi_u^{id}$
- Optimal experiment design $\Rightarrow$ Input spectrum $\Phi_u^{id}$
- Spectral factorization: $\Phi_u^{id}(e^{j\omega}) = |R_u(e^{j\omega})|^2$
\begin{align*}
\min_{\text{Input spectrum}} \quad & NE[u_t^2] \\
\text{s.t.} \quad & I_1^N \succeq \frac{\gamma n}{2} V_{\text{app}}
\end{align*}

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

Hypothesis $H_o$

\begin{align*}
G_o & \quad u_t \\
H_o & \quad v_t \\
y_t & \quad y_t
\end{align*}
Computations

\[
\begin{aligned}
\min_{\text{Input spectrum}} & \quad N E[u_t^2] \\
\text{s.t.} & \quad \mathcal{I}_1^N \geq \frac{\gamma n}{2} V_{\text{app}}
\end{aligned}
\]

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

![Diagram](image_url)
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
Experimental results: Water tank process

Diagram of water tank process:
- Pumps 1 and 2
- Valves 1 and 2
- Tanks 1, 2, 3, 4
MPC: Black: based on AOID-model. Red: based on white noise excitation
Computations

Application oriented experiment design

\[
\min \mathbb{E}[u^2(t)] \\
\text{s.t. } I_N^N(\theta_0) \geq \frac{\gamma n}{2} V''_{\text{app}}(\theta_0)
\]
Application oriented experiment design

\[
\begin{align*}
\min & \quad \mathbb{E}[u^2(t)] \\
\text{s.t.} & \quad \mathcal{I}_1^N(\theta_o) \geq \gamma_n \frac{n}{2} V_{\text{app}}''(\theta_o)
\end{align*}
\]
Application oriented experiment design

\[
\min N \mathbb{E}[u^2(t)] \\
\text{s.t. } \mathcal{I}_1^N(\theta_0) \geq \frac{\gamma n}{2} V''_{\text{app}}(\theta_0)
\]

- Optimization problem depends on the unknown system!
Application oriented experiment design

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

- Optimization problem depends on the unknown system!
- Solutions:
Application oriented experiment design

\[
\text{min } \mathcal{N}[u^2(t)] \\
\text{s.t. } \mathcal{I}_1^N(\theta_o) \geq \frac{\gamma n}{2} V_{\text{app}}''(\theta_o)
\]

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

Solutions:

- Robust experiment design
  (e.g. Rojas, Welsh, Goodwin, Feuer 2007)

- Adaptive (or sequential) experiment design
Optimization problem depends on the unknown system!

Solutions:

- Robust experiment design
  (e.g. Rojas, Welsh, Goodwin, Feuer 2007)

- Adaptive (or sequential) experiment design
  ⇒ Active application oriented learning
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

Recursive Id
Exp. design
\[ \hat{\theta}(t) \]
\[ R_u(t) \]
An adaptive feedback system
But measured signal not fed back directly
Exp. design limits input power
⇒ Stability when
\[ G_o \]
\[ H_o \]

Key questions:
Convergence?
Accuracy?
Active application oriented learning

Recursive Id

Exp. design \( \dot{\theta}(t) \)

But measured signal not fed back directly

\( e_t \)

\( H_o \)

\( u_t \)

\( G_o \)

\( v_t \)

\( y_t \)

Key questions:

Convergence?

Accuracy?
Active application oriented learning

An adaptive feedback system

But measured signal not fed back directly

⇒ Stability when $G_o$ stable

Key questions:

Convergence?

Accuracy?
Active application oriented learning

Recursive Id

Exp. design

But measured signal not fed back directly

⇒ Stability when $G_o$ stable

Key questions:

Convergence?

Accuracy?
Active application oriented learning

An adaptive feedback system
Active application oriented learning

An adaptive feedback system

But measured signal not fed back directly
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

\[ \hat{\theta}(t) \text{ has the same asymptotic accuracy as the off-line estimate that uses data collected under the optimal experimental conditions (using knowledge of } \theta_o) \]
What happens when true system is not in the model set?
Example: Non-minimum phase zero estimation

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

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

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

Model: \[ y_t = \frac{\theta_1 q + \theta_2}{q^2} u_t + e_t \]
Example: Non-minimum phase zero estimation

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

Model: \( y_t = \frac{\theta_1 q + \theta_2}{q^2} u_t + e_t \)
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
Application oriented dual control

\[
\min_{\text{Input spectrum}} \quad \mathbb{E}[u_t^2], \quad \text{s.t.} \quad \mathcal{I}_1^N(\theta_o) \geq \frac{\gamma n}{2} V''_{\text{app}}(\theta_o)
\]
Application oriented dual control

\[
\min_{\text{Input spectrum}} \quad \mathbb{E}[u_t^2], \quad \text{s.t.} \quad \mathcal{I}^N_1(\theta_o) \geq \frac{\gamma n}{2} V_{\text{app}}(\theta_o)
\]

Idéa: Replace cost function with control objective.
Application oriented dual control

\[
\min_{\text{Input spectrum}} \quad \mathbb{E}[u_t^2], \quad \text{s.t.} \quad I_1^N(\theta_o) \geq \frac{\gamma n}{2} V_{\text{app}}(\theta_o)
\]

Idéa: Replace cost function with control objective.

Policy (control rule): \( \pi = (\pi_1, \pi_2, \ldots) \)
Application oriented dual control

\[
\min_{\text{Input spectrum}} \quad NE[u_t^2], \quad \text{s.t.} \quad \mathcal{I}_1^N(\theta_o) \geq \frac{\gamma n}{2} V_{\text{app}}(\theta_o)
\]

Idée: Replace cost function with control objective.

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

Initial state distribution: \( \beta(x) \)
Application oriented dual control

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

Idéa: Replace cost function with control objective.

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

Initial state distribution: \( \beta(x) \)

Instantaneous cost: \( c(x_t, u_t) \)
Application oriented dual control

\[
\min_{\text{Input spectrum}} \quad \mathbb{E}[u_t^2], \quad \text{s.t.} \quad \mathcal{I}_1^N(\theta_o) \geq \frac{\gamma n}{2} V_{\text{app}}(\theta_o)
\]

Idéa: Replace cost function with control objective.

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

Initial state distribution: \( \beta(x) \)

Instantaneous cost: \( c(x_t, u_t) \)  \( \text{e.g.} \; y_t^T Q y_t + u_t S u_t \)
Application oriented dual control

\[
\min_{\text{Input spectrum}} \quad NE[u_t^2], \quad \text{s.t.} \quad \mathcal{I}_1^N(\theta_o) \geq \frac{\gamma n_2}{2} V_{\text{app}}(\theta_o)
\]

Idéa: Replace cost function with control objective.

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

Initial state distribution: \( \beta(x) \)

Instantaneous cost: \( c(x_t, u_t) \) \quad e.g. \( y_t^TQy_t + u_tSu_t \)

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

\[
\min_{\text{Input spectrum}} \quad N \mathbb{E}[u_t^2], \quad \text{s.t.} \quad \mathcal{I}_1^N(\theta_o) \geq \frac{\gamma n}{2} V_{\text{app}}(\theta_o)
\]

Idéa: Replace cost function with control objective.

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

Initial state distribution: \( \beta(x) \)

Instantaneous cost: \( c(x_t, u_t) \) \quad \text{e.g.} \quad y_t^T Q y_t + u_t S u_t

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

Constraints: \( x_t \in \mathcal{X} \subseteq \mathbb{R}^n, \quad y_t \in \mathcal{Y} \subseteq \mathbb{R}^p, \quad u_t \in \mathcal{U} \subseteq \mathbb{R}^m \)
Application oriented dual control

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

Idée: Replace cost function with control objective.

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

Initial state distribution: \( \beta(x) \)

Instantaneous cost: \( c(x_t, u_t) \) \quad \text{e.g.} \quad y_t^T Q y_t + u_t S u_t 

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

Constraints: \( x_t \in \mathcal{X} \subseteq \mathbb{R}^n, \ y_t \in \mathcal{Y} \subseteq \mathbb{R}^p, \ u_t \in \mathcal{U} \subseteq \mathbb{R}^m \)

\[
\min_{\pi} \quad C_\beta(\pi, N) \\
\text{s.t.} \quad \mathcal{I}_1^N(\theta_o) \geq \frac{\gamma n}{2} V''_{\text{app}}(\theta_o) \quad \text{Reward} \\
x_t \in \mathcal{X} \subseteq \mathbb{R}^n, \ y_t \in \mathcal{Y} \subseteq \mathbb{R}^p, \ u_t \in \mathcal{U} \subseteq \mathbb{R}^m
\]
Application oriented dual control

\[
\min_{\pi} \quad C_\beta(\pi, N) \\
\text{s.t.} \quad \mathcal{L}_1^N(\theta_o) \geq \frac{\gamma n}{2} V_{\text{app}}(\theta_o) \\
x_t \in \mathcal{X} \subseteq \mathbb{R}^n, \quad y_t \in \mathcal{Y} \subseteq \mathbb{R}^p, \quad u_t \in \mathcal{U} \subseteq \mathbb{R}^m
\]
Application oriented dual control

$$\min_{\pi} \quad C_{\beta}(\pi, N)$$

s.t. $$\mathcal{I}^N_1(\theta_o) \geq \frac{\gamma n}{2} V''_{\text{app}}(\theta_o)$$

$$x_t \in \mathcal{X} \subseteq \mathbb{R}^n, \quad y_t \in \mathcal{Y} \subseteq \mathbb{R}^p, \quad u_t \in \mathcal{U} \subseteq \mathbb{R}^m$$

We will look at two approaches to solve this problem:
Application oriented dual control

\[
\begin{align*}
\min_{\pi} & \quad C_{\beta}(\pi, N) \\
\text{s.t.} & \quad \mathcal{L}_1^N(\theta_o) \geq \frac{\gamma n}{2} V_{\text{app}}(\theta_o) \\
& \quad x_t \in \mathcal{X} \subseteq \mathbb{R}^n, \quad y_t \in \mathcal{Y} \subseteq \mathbb{R}^p, \quad u_t \in \mathcal{U} \subseteq \mathbb{R}^m
\end{align*}
\]

We will look at two approaches to solve this problem:

- Markov Decision Process (MDP) formulation
Application oriented dual control

\[
\begin{align*}
\min_{\pi} & \quad C_\beta(\pi, N) \\
\text{s.t.} & \quad \mathcal{L}_1^N(\theta_o) \geq \frac{\gamma n}{2} V''_{\text{app}}(\theta_o) \\
& \quad x_t \in \mathcal{X} \subseteq \mathbb{R}^n, \ y_t \in \mathcal{Y} \subseteq \mathbb{R}^p, \ u_t \in \mathcal{U} \subseteq \mathbb{R}^m
\end{align*}
\]

We will look at two approaches to solve this problem:

- Markov Decision Process (MDP) formulation
- Receding horizon formulation: MPC-X
Infinite horizon cost: \( C_\beta(\pi) = \limsup_{N \to \infty} C_\beta(\pi, N) \)
Markov Decision Process formulation

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

Reward: \( \mathcal{I}_{\beta}(\pi, \theta) \triangleq \lim_{N \to \infty} \frac{1}{N} \mathcal{I}_1^N(\theta) \)
Infinite horizon cost: \( C_\beta(\pi) = \lim \sup_{N \to \infty} C_\beta(\pi, N) \)

Reward: \( I_\beta(\pi, \theta) \triangleq \lim_{N \to \infty} \frac{1}{N} I_1^N(\theta) \) (per sample information)
Markov Decision Process formulation

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

Reward: \( I_\beta(\pi, \theta) \triangleq \lim_{N \to \infty} \frac{1}{N} I_1^N(\theta) \) (per sample information)

Theory based on discretized state and action spaces
Markov Decision Process formulation

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

Reward: \( I_\beta(\pi, \theta) \triangleq \lim_{N \to \infty} \frac{1}{N} I_1^N(\theta) \) (per sample information)

Theory based on discretized state and action spaces

\( \Rightarrow \) State \( x \) & input \( u \) take only finite number of values
Infinite horizon cost: $C_\beta(\pi) = \limsup_{N \to \infty} C_\beta(\pi, N)$

Reward: $I_\beta(\pi, \theta) \triangleq \lim_{N \to \infty} \frac{1}{N} I_1^N(\theta)$ (per sample information)

Theory based on discretized state and action spaces

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

⇒ Need to approximate state-space description
Infinite horizon cost: $C_\beta(\pi) = \limsup_{N \to \infty} C_\beta(\pi, N)$

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

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

⇒ Need to approximate state-space description

⇒ Transition probabilities: $p_{x\bar{x}}(u) = \mathbb{P}\{x_{t+1} = \bar{x} \mid x_t = x, u_t = u\}$
Markov Decision Process formulation

Infinite horizon cost: \( C_\beta(\pi) = \lim \sup_{N \to \infty} C_\beta(\pi, N) \)

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: \( p_{x\bar{x}}(u) = \mathbb{P}\{x_{t+1} = \bar{x} \mid x_t = x, u_t = u\} \)

Can be computed based on geometry of discretization and knowledge of distributions of disturbances
Markov Decision Process formulation

Infinite horizon cost:  \( C_\beta(\pi) = \lim \sup_{N \to \infty} C_\beta(\pi, N) \)

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

\[
\begin{align*}
\min_{\pi} & \quad C_\beta(\pi) \\
\text{s.t.} & \quad N I_\beta(\pi, \theta_o) \geq \frac{\gamma n}{2} V_{\text{app}}(\theta_o) \\
& \quad x_t \in \mathcal{X} \subseteq \mathbb{R}^n, \quad y_t \in \mathcal{Y} \subseteq \mathbb{R}^p, \quad u_t \in \mathcal{U} \subseteq \mathbb{R}^m
\end{align*}
\]
\[
\begin{align*}
\min_{\pi} & \quad C_\beta(\pi) \\
\text{s.t.} & \quad N I_\beta(\pi, \theta_o) \geq \frac{\gamma_n}{2} V''_{\text{app}}(\theta_o) \\
& \quad x_t \in \mathcal{X} \subseteq \mathbb{R}^n, \; y_t \in \mathcal{Y} \subseteq \mathbb{R}^p, \; u_t \in \mathcal{U} \subseteq \mathbb{R}^m
\end{align*}
\]
Solution?
\[
\begin{align*}
\min_{\pi} & \quad C_\beta(\pi) \\
\text{s.t.} & \quad N \mathcal{I}_\beta(\pi, \theta_o) \geq \frac{\gamma_n}{2} V''_{\text{app}}(\theta_o) \\
& \quad x_t \in \mathcal{X} \subseteq \mathbb{R}^n, \; y_t \in \mathcal{Y} \subseteq \mathbb{R}^p, \; u_t \in \mathcal{U} \subseteq \mathbb{R}^m
\end{align*}
\]

Solution?

Define $z_{xu}$ as the probability of being in state $x$ and taking action $u$.
\[
\min_{\pi} \quad C_\beta(\pi) \\
\text{s.t.} \quad N I_\beta(\pi, \theta_o) \geq \frac{\gamma n}{2} V_{\text{app}}(\theta_o) \\
\quad x_t \in \mathcal{X} \subseteq \mathbb{R}^n, \quad y_t \in \mathcal{Y} \subseteq \mathbb{R}^p, \quad u_t \in \mathcal{U} \subseteq \mathbb{R}^m
\]

Solution?

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

*Occupancy measure*
Markov Decision Process formulation: Implementation

\[
\begin{align*}
\min_{\pi} & \quad C_\beta(\pi) \\
\text{s.t.} & \quad N \mathcal{I}_\beta(\pi, \theta_o) \geq \frac{\gamma n}{2} V_{\text{app}}(\theta_o) \\
& \quad x_t \in \mathcal{X} \subseteq \mathbb{R}^n, \ y_t \in \mathcal{Y} \subseteq \mathbb{R}^p, \ u_t \in \mathcal{U} \subseteq \mathbb{R}^m
\end{align*}
\]

Solution?

Define \( \tau_{xu} \) as the probability of being in state \( x \) and taking action \( u \)

**Occupancy measure**

**MDP problem is a semi-definite program in \( \{\tau_{xu}\} \).**
\[
\begin{align*}
x_{t+1} &= -\theta_1 x_t + \theta_2 u_t - \theta_1 v_t, \\
y_t &= x_t + v_t,
\end{align*}
\]
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 \(1 \times 10^{-3}\)
Markov Decision Process formulation: Simulation study

\[
\begin{align*}
    x_{t+1} &= -\theta_1 x_t + \theta_2 u_t - \theta_1 v_t, \\
    y_t &= x_t + v_t,
\end{align*}
\]

\(v_t:\) Gaussian white noise with variance \(1 \times 10^{-3}\)

\(\theta_o = [0.5, 0.5]^T\)
\[
\begin{cases}
  x_{t+1} = -\theta_1 x_t + \theta_2 u_t - \theta_1 v_t, \\
  y_t = x_t + v_t,
\end{cases}
\]

\(v_t\): Gaussian white noise with variance \(1 \times 10^{-3}\)

\(\theta_o = [0.5, 0.5]^T\)

\(x\) split in 51 regions.

\(u\) split in 21 regions.
Markov Decision Process formulation: Simulation study

\[
\begin{align*}
x_{t+1} &= -\theta_1 x_t + \theta_2 u_t - \theta_1 v_t, \\
y_t &= x_t + v_t,
\end{align*}
\]

\(v_t\): Gaussian white noise with variance \(1 \times 10^{-3}\)

\(\theta_o = [0.5, 0.5]^T\)

\(x\) 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

\[
\begin{align*}
    x_{t+1} &= -\theta_1 x_t + \theta_2 u_t - \theta_1 v_t, \\
    y_t &= x_t + v_t,
\end{align*}
\]

\(v_t\): Gaussian white noise with variance \(1 \times 10^{-3}\)

\(\theta_o = [0.5, 0.5]^T\)

\(x\) 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

\[
\begin{align*}
x_{t+1} &= -\theta_1 x_t + \theta_2 u_t - \theta_1 v_t, \\
y_t &= x_t + v_t,
\end{align*}
\]

\(v_t\): Gaussian white noise with variance \(1 \times 10^{-3}\)

\(\theta_o = [0.5, 0.5]^T\)

\(x\) 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

\[
\begin{align*}
    x_{t+1} &= -\theta_1 x_t + \theta_2 u_t - \theta_1 v_t, \\
    y_t &= x_t + v_t,
\end{align*}
\]

\(v_t\): Gaussian white noise with variance \(1 \times 10^{-3}\)

\(\theta_o = [0.5, 0.5]^T\)

\(x\) 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

\[
\begin{align*}
x_{t+1} &= -\theta_1 x_t + \theta_2 u_t - \theta_1 v_t, \\
y_t &= x_t + v_t,
\end{align*}
\]

\(v_t\): Gaussian white noise with variance \(1 \times 10^{-3}\)

\(\theta_o = [0.5, 0.5]^T\)

\(x\) 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
Receding horizon formulation

Cost at time $t$:

$$C_t = F \sum_{k=1}^{\infty} c_t(x_k, u_k) = F \sum_{k=1}^{\infty} \|y_{k+1} - r_{t+k+1}\|_2 Q + F \sum_{k=1}^{\infty} \|u_k\|_2 S$$

minimize \{u_k\} for $k = 1, \ldots, N$

subject to

$$x_1 = \hat{x}_t,$$

$$x_{k+1} = A(\theta_0) x_k + B(\theta_0) u_k,$$

$$y_k = C(\theta_0) x_k$$

$u_k \in U, x_k \in X, y_k \in Y$

$I_{t+N} \preceq \kappa t \gamma n \|v''_{\text{app}}(\theta_0)\|

Scaling $\kappa_t$ monotonically increasing from 0 to 1 at $t = N - N_1(\theta_0)$

Major problems:

$\theta_0$ unknown

Data not stationary & feedback $\Rightarrow$ Cannot use spectrum as design variable
Receding horizon formulation

Cost at time $t$:

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

Cost at time $t$:

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

minimize $C_t$ subject to

$\{u_k\}_{k=1}^{F}$
Receding horizon formulation

Cost at time $t$:

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

minimize \[ C_t \]

subject to \[ x_1 = \hat{x}_t, \]
Receeding horizon formulation

Cost at time $t$: 

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

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

subject to $x_1 = \hat{x}_t,$

$$ x_{k+1} = A(\theta_o)x_k + B(\theta_o)u_k, $$

Scaling $\kappa_t$ monotonically increasing from 0 to 1 at $t = N - N_I$.

Major problems: $\theta_o$ unknown, data not stationary & feedback $\Rightarrow$ Cannot use spectrum as design variable.
Receding horizon formulation

Cost at time $t$:

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

minimize $C_t$

subject to $x_1 = \hat{x}_t,$

$$x_{k+1} = A(\theta_o)x_k + B(\theta_o)u_k, \quad y_k = C(\theta_o)x_k$$
Receding horizon formulation

Cost at time $t$:

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

minimize $C_t$

subject to

$x_1 = \hat{x}_t$,

$x_{k+1} = A(\theta_o)x_k + B(\theta_o)u_k$, $y_k = C(\theta_o)x_k$

$u_k \in \mathcal{U}$, $x_k \in \mathcal{X}$, $y_k \in \mathcal{Y}$
Receding horizon formulation

Cost at time $t$:

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

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

subject to $x_1 = \hat{x}_t,$

$x_{k+1} = A(\theta_o)x_k + B(\theta_o)u_k,$ $y_k = C(\theta_o)x_k$

$u_k \in \mathcal{U}, x_k \in \mathcal{X}, y_k \in \mathcal{Y}$

$I_{t+N_I}^t(\theta_o) \geq \kappa_t \frac{\gamma n}{2} V_{\text{app}}(\theta_0)$
Receding horizon formulation

Cost at time $t$:

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

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

subject to

$x_1 = \hat{x}_t$,

$x_{k+1} = A(\theta_o)x_k + B(\theta_o)u_k$, $y_k = C(\theta_o)x_k$

$u_k \in \mathcal{U}$, $x_k \in \mathcal{X}$, $y_k \in \mathcal{Y}$

$I_{1+N_I}^{t+N_I}(\theta_o) \geq \kappa_t \frac{\gamma n}{2} V_{\text{app}}''(\theta_0)$

Scaling $\kappa_t$ monotonically increasing from 0 to 1 at $t = N - N_I$
Receding horizon formulation

Cost at time $t$:

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

minimize \( C_t \)

subject to \( x_1 = \hat{x}_t \),

\( x_{k+1} = A(\theta_o)x_k + B(\theta_o)u_k \), \( y_k = C(\theta_o)x_k \)

\( u_k \in \mathcal{U} \), \( x_k \in \mathcal{X} \), \( y_k \in \mathcal{Y} \)

\( \mathcal{I}_{t+N_I}^t(\theta_o) \geq \kappa_t \frac{\gamma n}{2} V''_{\text{app}}(\theta_0) \)

Scaling \( \kappa_t \) monotonically increasing from 0 to 1 at \( t = N - N_I \)

Major problems:

- \( \theta_o \) unknown
Receding horizon formulation

Cost at time $t$: 

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

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

subject to $x_1 = \hat{x}_t,$

$$x_{k+1} = A(\theta_o)x_k + B(\theta_o)u_k,$$

$$y_k = C(\theta_o)x_k,$$

$$u_k \in \mathcal{U}, x_k \in \mathcal{X}, y_k \in \mathcal{Y}$$

$$\mathcal{I}_{1+NI}(\theta_o) \geq \kappa_t \frac{\gamma n}{2} V_{app}(\theta_0)$$

Scaling $\kappa_t$ monotonically increasing from 0 to 1 at $t = N - NI$

Major problems:

- $\theta_o$ unknown
- Data not stationary & feedback
Receding horizon formulation

Cost at time $t$:

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

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

subject to
$$x_1 = \hat{x}_t,$$
$$x_{k+1} = A(\theta_o)x_k + B(\theta_o)u_k, \ y_k = C(\theta_o)x_k$$
$$u_k \in \mathcal{U}, \ x_k \in \mathcal{X}, \ y_k \in \mathcal{Y}$$
$$\mathcal{I}_{1+N_I}(\theta_o) \geq \kappa_t \frac{\gamma n}{2} V''_{\text{app}}(\theta_0)$$

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

Major problems:
- $\theta_o$ unknown
- Data not stationary & feedback
  $$\Rightarrow \text{Cannot use spectrum as design variable}$$
Receding horizon formulation: Implementation

Approximations:
Approximations:

- Initial estimate $\hat{\theta}$ replaces $\theta_o$
Receding horizon formulation: Implementation

Approximations:

- Initial estimate $\hat{\theta}$ replaces $\theta_o$
- $I_1^N(\hat{\theta})$ sample approximation of $I_1^N(\theta_o)$
Approximations:

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

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

- Initial estimate $\hat{\theta}$ replaces $\theta_o$
- $I_{1}^{N}(\hat{\theta})$ sample approximation of $I_{1}^{N}(\theta_o)$

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

Lifting:
Receding horizon formulation: Implementation

Approximations:

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

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

Lifting: Introduce $U = \bar{u}\bar{u}^T$
Receding horizon formulation: Implementation

Approximations:
- Initial estimate $\hat{\theta}$ replaces $\theta_o$
- $I_1^N(\hat{\theta})$ sample approximation of $I_1^N(\theta_o)$

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

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

$$\begin{bmatrix} U & \bar{u} \\ \bar{u}^T & 1 \end{bmatrix} \succeq 0, \quad \text{rank} \begin{bmatrix} U & \bar{u} \\ \bar{u}^T & 1 \end{bmatrix} = 1$$
Approximations:
- Initial estimate $\hat{\theta}$ replaces $\theta_o$
- $I_1^N(\hat{\theta})$ sample approximation of $I_1^N(\theta_o)$

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

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

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

Convex relaxation: Drop the rank constraint
Approximations:
- Initial estimate $\hat{\theta}$ replaces $\theta_o$
- $I_1^N(\hat{\theta})$ sample approximation of $I_1^N(\theta_o)$

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

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

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

Convex relaxation: Drop the rank constraint

Alternative formulation:
Receding horizon formulation: Implementation

Approximations:
- Initial estimate $\hat{\theta}$ replaces $\theta_o$
- $I_1^N(\hat{\theta})$ sample approximation of $I_1^N(\theta_o)$

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

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

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

Convex relaxation: Drop the rank constraint

Alternative formulation: Minimum time
Approximations:

- Initial estimate $\hat{\theta}$ replaces $\theta_o$
- $I_1^N(\hat{\theta})$ 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 $U = \bar{u}\bar{u}^T$ $\iff$

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

Convex relaxation: Drop the rank constraint

Alternative formulation: Minimum time (maximize $\kappa_t$)
Receding horizon formulation: Implementation

Approximations:
- Initial estimate $\hat{\theta}$ replaces $\theta_o$
- $I_1^N(\hat{\theta})$ sample approximation of $I_1^N(\theta_o)$

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

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

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

Convex relaxation: Drop the rank constraint

Alternative formulation: Minimum time (maximize $\kappa_t$)

**MPC-X: Model Predictive Control with eXperimental constraints**
Receding horizon formulation: Alternative approaches

\[ y_t = \sum_{k=1}^{n_b} \theta_k u_{t-k} + e_t = \theta^T \phi_t + e_t, \]
Receding horizon formulation: Alternative approaches

\[ y_t = \sum_{k=1}^{n_b} \theta_k u_{t-k} + e_t = \theta^T \phi_t + e_t, \quad \phi_t = [u_{t-1} \cdots u_{t-n_b}]^T \]
Receding horizon formulation: Alternative approaches

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

Persistence of excitation condition: \( \sum_{k=t-P}^{t+1+F} \phi_k \phi_k^T \succeq \rho I \)
Receding horizon formulation: Alternative approaches

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

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

- 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)): \( F = 0 \)


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

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

- 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
Receding horizon formulation: Simulation study
Receding horizon formulation: Simulation study

\[
\begin{align*}
\begin{cases}
x_{t+1} &= \begin{bmatrix} \theta_3 & \theta_4 \\ 1 & 0 \end{bmatrix} x_t + \begin{bmatrix} 4.5 \\ 0 \end{bmatrix} u_t,
\end{cases}
\end{align*}
\]
Receding horizon formulation: Simulation study

\[
\begin{align*}
\begin{cases}
    x_{t+1} &= \begin{bmatrix} \theta_3 & \theta_4 \\ 1 & 0 \end{bmatrix} x_t + \begin{bmatrix} 4.5 \\ 0 \end{bmatrix} u_t, \\
y_t &= \begin{bmatrix} \theta_1 & \theta_2 \end{bmatrix} x_t + e_t
\end{cases}
\end{align*}
\]
Receding horizon formulation: Simulation study

\[
\begin{align*}
    x_{t+1} &= \begin{bmatrix} \theta_3 & \theta_4 \\ 1 & 0 \end{bmatrix} x_t + \begin{bmatrix} 4.5 \\ 0 \end{bmatrix} u_t, \\
    y_t &= \begin{bmatrix} \theta_1 & \theta_2 \end{bmatrix} x_t + e_t \quad \text{lower tank level}
\end{align*}
\]
Receding horizon formulation: Simulation study

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

\[
\theta_o = \begin{bmatrix} 0.12 & 0.059 & 0.74 & -0.14 \end{bmatrix}^T \quad \text{Noise var.: 0.01.}
\]

\[
N = 200, \quad F = 5
\]
Receding horizon formulation: Simulation study

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

\[\theta_o = \begin{bmatrix} 0.12 & 0.059 & 0.74 & -0.14 \end{bmatrix}^T\] Noise var.: 0.01.

\[N = 200, \quad F = 5\]

Performance degradation cost: 
\[V_{\text{app}}(\theta) = \sum_{t=1}^{T} \|y_t(\theta_o) - y_t(\theta)\|_2^2\]
Receding horizon formulation: Simulation study

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

\[\theta_o = \begin{bmatrix} 0.12 & 0.059 & 0.74 & -0.14 \end{bmatrix}^T\quad \text{Noise var.: 0.01.}\]

\[N = 200, \quad F = 5\]

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

PE-MPC: \(\rho = 0.5, \quad P = 5, \quad F = 0\)
Receding horizon formulation: Simulation study

\[
\begin{align*}
\dot{x}_{t+1} &= \left[ \begin{array}{cc} \theta_3 & \theta_4 \\ 1 & 0 \end{array} \right] x_t + \left[ \begin{array}{c} 4.5 \\ 0 \end{array} \right] u_t, \\
y_t &= \left[ \begin{array}{cc} \theta_1 & \theta_2 \end{array} \right] x_t + e_t \quad \text{lower tank level}
\end{align*}
\]

\[\theta_o = \begin{bmatrix} 0.12 & 0.059 & 0.74 & -0.14 \end{bmatrix}^T\]

Noise var.: 0.01.

\[N = 200, \quad F = 5\]

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

PE-MPC: \[\rho = 0.5, \quad P = 5, \quad F = 0\]

MPC-X: Minimum time formulation
Receding horizon formulation: Simulation study

![Graphs showing output $y_t$ over time](image-url)
Receding horizon formulation: Simulation study

Time (sample)

Input $u_t$

-1 -0.5 0 0.5 1

Time (sample)

Input $u_t$

-1 -0.5 0 0.5 1

Time (sample)

Input $u_t$

-1 -0.5 0 0.5 1
Receding horizon formulation: Simulation study

\[ \lambda_{\min} \left\{ I_1 - \gamma n V_{app} \right\} \]

- Regular MPC (yellow)
- PE-MPC with \( \rho = 0.5 \) (red)
- Minimum time MPC-X (green)
Receding horizon formulation: Simulation study

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Var $u$</th>
<th>Var $y$</th>
<th>$N$</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPC-X, minimum time</td>
<td>0.203</td>
<td>0.146</td>
<td>82</td>
</tr>
<tr>
<td>PE-MPC, $\rho = 0.5$</td>
<td>0.175</td>
<td>0.120</td>
<td>211</td>
</tr>
</tbody>
</table>
MPC-X experimental study: Let’s travel
Secunda, South Africa
SASOL Synthetic Fuels Refinery
Synfuels Catalytic Cracker (SCC)
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
Figure 6.9: The excitation signals generated by MPC-X in the experiment closed-loop reidentification experiment of the plant. The two signals important for the control, MV 2 and 3, are excited while MV 1, which is typically not used by the controller, is not excited. The scale for MV 1 covers the full variable range in the MPC and the changes in the signal value are due to operator manipulations and not to MPC-X.
Depropanizer: Model fit

Open loop data

Closed loop data

- The plant output
- Model identified in open-loop
- Model identified in closed-loop MPC-X experiment
## Depropanizer: Closed loop performance

<table>
<thead>
<tr>
<th>Model</th>
<th>Variance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CV 1</td>
</tr>
<tr>
<td>Before MPC-X</td>
<td>$95 \times 10^3$</td>
</tr>
<tr>
<td>After MPC-X model update</td>
<td>$36 \times 10^3$</td>
</tr>
</tbody>
</table>

$\text{MV 5} = C_4$ content in the feed
Sample version of Information Application Inequality added as a matrix inequality constraint in MPC
Sample version of Information Application Inequality added as a matrix inequality constraint in MPC

Convex relaxation
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

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
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
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
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!!!