Guest Lecture on
Input Design

Cristian R. Rojas
Dept. of Automatic Control and ACCESS Linnaeus Centre
KTH - Royal Institute of Technology
Stockholm, Sweden

Uppsala, May 23, 2016
About myself

• BSc and MSc in Electronic Engineering from UTFSM, Chile
• PhD in Automatic Control from U. Newcastle, Australia
• Postdoc at KTH, Stockholm
• Currently: Assoc. Prof. in Automatic Control at KTH, Stockholm
• Research interests: System identification, signal processing, ...

http://people.kth.se/~crro/index.html
Outline

- Review of identification
- Basic input signals for system identification
- Optimal input design (via convex optimization)
- State-of-the-art: Application-oriented input design
- Recent extensions: Input design for nonlinear systems
Review of System identification

Basic idea: estimate system from measurements of $u(t)$ and $y(t)$

Many issues

- proper experiment conditions: sampling freq., input signal, etc.
- class of models: parametric/nonparametric, linear/nonlinear, etc
- estimating model parameters from sampled, finite and noisy data
System identification procedure

User choices

- *Experimental condition* input, sampling freq, open/closed loop, ...

- *Model structure* linear/nonlinear, parametric/nonparametric, ...

- *Identification method* least squares, PEM, ...

Cristian R. Rojas, *Input design*
Basic input design
Experiments and data collection

Often good to use a two-stage approach

1. Preliminary experiments
   - step/impulse response tests to get basic understanding of system dynamics
   - linearity, stationary gains, time delays, time constants, sampling interval

2. Data collection for model estimation
   - carefully designed experiment to enable good model fit
   - operating point, input signal type, number of data points to collect, etc.
Preliminary experiments

- The input signal in an identification experiment can have a significant influence on the resulting model

- Typical choices:
  - (Approximate) impulse functions
  - Step functions
  - Pseudorandom binary sequences (PRBS)
  - Sinusoids
Preliminary experiments: step response

Easy to apply and useful for obtaining basic information about system

- dead-times, static gain, time constants, resonances and aids sampling time selection (rule-of-thumb: 4-10 samples / rise time)
- More power at low freq’s than at high freq’s
- Very sensitive to noise (needs large amplitude to get good results)
- User choices: Amplitude $A$, duration $T$
Preliminary experiments: sinusoids

\[ u(t) = \alpha \sin(\omega t + \phi) \]

- Easy to apply
- Useful for directly obtaining freq. domain information
- Focuses on freq’s of interest
- Amplitude can be traded off for the duration of the experiment, i.e. for \( \alpha \) larger the experiment can be shortened
- Only provides information at one freq, hence many experiments needed to obtain an adequate model
Preliminary experiments: multi-sines

\[ u(t) = \sum_{k=1}^{m} \alpha_k \sin(\omega_k t + \phi_k), \quad \omega_k = l_k \omega_0, \quad l_k \in \mathbb{N} \]

- If \( \phi_k \)'s are chosen equal to 0, the amplitude can be large
- Schroeder method: Choose \( \phi_k = \frac{k^2 \pi}{m} \) (if \( \alpha_k \)'s are equal)
- \( \omega_k \)'s should be in the freq. region of interest, and multiples of \( h \)
Tests for verifying linearity

For linear systems, response is independent of operating point

- test linearity by a sequence of step response tests for different operating points
Tests for detecting friction

Friction can be detected by using small step increases in input

Input moves every two or three steps

Cristian R. Rojas, Input design
Designing experiment for model estimation

Input signal should excite all relevant frequencies
- estimated model accurate in freq’s where input has much energy
- good choice: binary signal with random ”hold times” (e.g., PRBS)

Trade-off in selection of signal amplitude
- constraints on the input (economic, safety, actuator limits, ...)
- large amplitude gives high signal-to-noise ratio, low variance
- most systems are nonlinear for large input amplitudes

Sampling freq? Typically $5 - 10 \times$ time constant of the process
Many pitfalls if estimating a model of a system under closed-loop!
Optimal input design
(for linear systems)
Brief history of (optimal) input design

- **Experimental design**: R. A. Fisher, *The Design of Experiments*, 1935 (randomized, factorial experiments, ...)


- **Application-oriented input design**: H. Hjalmarsson, *European Control Conference*, 2009
Review of convex optimization

• In the second stage, we can design an *optimal* input signal based on the knowledge obtained from the first experiment

• **Idea:** Pose the optimal input design problem as a *structured convex (semidefinite) optimization problem*

\[
\begin{align*}
\min_{x \in \mathbb{R}^n} & \quad c^T x \\
\text{s.t.} & \quad A_0 + A_1 x_1 + \cdots + A_n x_n \succeq 0 \quad (A_i = A_i^T \forall i)
\end{align*}
\]

(*positive semi-definiteness: \( A \succeq 0 \) iff \( x^T A x \geq 0, \forall x \)*)

• Semidefinite programs can be efficiently described and solved using standard software (*e.g.*, Matlab + CVX or YALMIP)
Input design as convex optimization

Optimal input design problems:

\[
\begin{align*}
\min_{\text{Input}} & \quad \text{Estimation error} \\
\text{s.t.} & \quad \text{Input power} \leq \alpha \\
\end{align*}
\]

or

\[
\begin{align*}
\min_{\text{Input}} & \quad \text{Input power} \\
\text{s.t.} & \quad \text{Estimation error} \leq \beta \\
\end{align*}
\]

(traditional design) (least costly design)

For large sample size, these problems are equivalent\(^a\) (their solutions are proportional to each other)

To “convexity” these problems for linear models, we will work in the frequency domain, so the input is characterized by its power spectrum

**Input spectrum**

\[ \Phi_u(\omega) = \sum_{k=-\infty}^{\infty} r_{|k|} e^{-jk\omega} \]

\[ r_k = \frac{1}{2\pi} \int_{-\pi}^{\pi} \Phi_u(\omega) e^{\pm jk\omega} d\omega \]

\[ = \cdots + r_1 e^{j\omega} + r_0 + r_1 e^{-j\omega} + \cdots \]

**Note:** The input power is given by \( \int_{-\pi}^{\pi} \Phi_u(\omega) d\omega = r_0! \)

The input spectrum \( \Phi_u \) is a function, i.e., an infinite-dimensional object: we need to discretize it

**Method 1: Finite-dimensional parametrization approach**

Restrict \( \Phi_u \) to

\[ \Phi_u(\omega) = \sum_{k=-n}^{n} r_{|k|} e^{-jk\omega} \]

This is the spectrum of a moving average (MA) process, to be parameterized by its autocovariance coefficients: \( r_0, \ldots, r_n \)
**Input spectrum (cont.)**

*Missing detail:* how to restrict \( r_0, \ldots, r_n \) so that \( \Phi_u(\omega) \geq 0 \) for all \( \omega \)?

**Key result:** Particular case of *Kalman-Yakubovich-Popov lemma*\(^a\)

\[
\Phi_u(\omega) \geq 0 \quad \forall \omega \quad \text{iff} \quad \text{there is a } Q = Q^T \text{ such that}
\]

\[
\text{KYP}(r_0, \ldots, r_n, Q) := \begin{bmatrix}
Q - A^TQA & C^T - A^TQB \\
C - B^TQA & r_0 - B^TQB
\end{bmatrix} \succeq 0,
\]

where \( A = \begin{bmatrix} 0_{1\times n-1} & 0 \\ I_{n-1} & 0_{n-1\times 1} \end{bmatrix} \), \( B = \begin{bmatrix} 1 & 0_{1\times n-1} \end{bmatrix}^T \), \( C = [r_1 \ldots r_n] \)

Thus, we should add the constraint “\( \text{KYP}(r_0, \ldots, r_n, Q) \succeq 0 \)”

Other frequency-dependent constraints can be imposed using a *generalized KYP lemma*\(^b\)


Cristian R. Rojas, *Input design*
Input spectrum (cont.)

Method 2: Partial correlation approach

\[
\Phi_u(\omega) = \sum_{k=\infty}^{\infty} r_k e^{-jk\omega} = \cdots + r_1 e^{j\omega} + r_0 + r_1 e^{-j\omega} + \cdots
\]

Instead of forcing \(r_{n+1} = \cdots = 0\), let’s parameterize \(\Phi_u\) in terms of \(r_0, \ldots, r_n\), ensuring that there exist \(r_{n+1}, \ldots\), such that \(\Phi_u(\omega) \geq 0 \ \forall \omega\)

**Key result:** Covariance extension\(^a\)

Given \(r_0, \ldots, r_n\), there exist \(r_{n+1}, r_{n+2}, \ldots\) such that \(\Phi_u(\omega) \geq 0 \ \forall \omega\) iff

\[
\mathbf{T}(r_0, \ldots, r_n) := \begin{bmatrix}
  r_0 & r_1 & \cdots & r_n \\
  r_1 & r_0 & \cdots & r_{n-1} \\
  \vdots & \vdots & \ddots & \vdots \\
  r_{n} & r_{n-1} & \cdots & r_0
\end{bmatrix} \succeq 0
\]


Cristian R. Rojas, *Input design* 21
Estimation error

In open loop \( \{u[k]\} \) indep. of \( \{w[k]\} \), if the sample size is large, and there is no undermodelling, the estimation error is directly related to the parameter covariance matrix:

\[
\text{cov}(\hat{\theta}) \approx \mathbb{E}\{\psi\psi^T\} = \left[ \frac{N}{2\pi\sigma^2} \int_{-\pi}^{\pi} \Gamma(e^{j\omega})\Gamma^H(e^{j\omega})\Phi_u(\omega)d\omega + \text{terms indep. of } u \right]^{-1}
\]

\[
\psi := \frac{\partial \hat{y}[k]}{\partial \theta} = \Gamma(q)u[k] + \Xi w[k]
\]

We can see that \( \text{cov}(\hat{\theta})^{-1} \) is affine in \( \Phi_u \), i.e., on \( r_0, \ldots, r_n \)!
Estimation error (cont.)

\[
\text{cov}(\hat{\theta})^{-1} \approx \sum_{k=-n}^{n} \left[ \frac{N}{2\pi\sigma^2} \int_{-\pi}^{\pi} \Gamma(e^{j\omega}) \Gamma^H(e^{j\omega}) e^{-jk\omega} d\omega \right] r_{|k|} + \cdots
\]

Example: FIR model

\[
y[k] = \theta_1 u[k-1] + \theta_2 u[k-2] + w[k] \Rightarrow \psi[k] = \frac{\partial \hat{y}[k]}{\partial \theta} = \begin{bmatrix} q^{-1} \\ q^{-2} \end{bmatrix} u[k] \\
\hat{y}[k]
\]

Therefore,

\[
\text{cov}(\hat{\theta})^{-1} \approx \sum_{k=-n}^{n} \left[ \frac{N}{2\pi\sigma^2} \int_{-\pi}^{\pi} \begin{bmatrix} 1 & e^{j\omega} \\ e^{-j\omega} & 1 \end{bmatrix} e^{-jk\omega} d\omega \right] r_{|k|} = \frac{N}{\sigma^2} \begin{bmatrix} r_0 & r_1 \\ r_1 & r_0 \end{bmatrix}
\]

(Recall: \( \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{jk\omega} d\omega = \begin{cases} 1, & k = 0 \\ 0, & \text{otherwise} \end{cases} \))
Estimation error (cont.)

Example: OE model

\[
y[k] = \frac{bq^{-1}}{1 + aq^{-1}} u[k] + w[k]; \quad \theta = [a \ b]^T
\]

\[
\Rightarrow \hat{y}[k] = \frac{bq^{-1}}{1 + aq^{-1}} u[k] \Rightarrow \psi[k] = -\frac{\partial \hat{y}[k]}{\partial \theta} = \begin{bmatrix} -\frac{bq^{-2}}{(1+aq^{-1})^2} \\ -\frac{q^{-1}}{1+aq^{-1}} \\ \Gamma(q) \end{bmatrix} u[k]
\]

Thus,

\[
\text{cov}(\hat{\theta})^{-1} \approx \sum_{k=-n}^{n} \left[ \frac{N}{2\pi\sigma^2} \int_{-\pi}^{\pi} \begin{bmatrix} b\\ -be^{j\omega}(1 + ae^{-j\omega})\\ -be^{-j\omega}(1 + ae^{j\omega}) \end{bmatrix} \begin{bmatrix} e^{-jk\omega} \\ |1 + ae^{-j\omega}|^{2} \\ |1 + ae^{j\omega}|^{4} \end{bmatrix} d\omega \right] r[k]
\]

To continue, the rational functions (in \( z = e^{j\omega} \)) in the integrand must be expanded in a Laurent series around \( |z| = 1 \)
Estimation error (cont.)

Example: OE model (cont.)

Another way to parameterize $\text{cov}(\hat{\theta})^{-1}$, more akin to the partial correlation approach, is to define an alternative input spectrum

$$\tilde{\Phi}_u(\omega) := \frac{\Phi_u(\omega)}{|1 + ae^{-j\omega}|^4} = \cdots + \tilde{r}_1 e^{j\omega} + \tilde{r}_0 + \tilde{r}_1 e^{-j\omega} + \cdots$$

Then,

$$\text{cov}(\hat{\theta})^{-1} \approx \sum_{k=-n}^{n} \left[ \frac{N}{2\pi \sigma^2} \int_{-\pi}^{\pi} \begin{bmatrix} b^2 & -be^{-j\omega}(1 + ae^{j\omega}) \\ -be^{j\omega}(1 + ae^{-j\omega}) & |1 + ae^{-j\omega}|^2 \end{bmatrix} e^{-jk\omega} d\omega \right] \tilde{r}|_k |$$

$$= N \begin{bmatrix} b^2 \tilde{r}_0 & -ab\tilde{r}_0 - b\tilde{r}_1 \\ -ab\tilde{r}_0 - b\tilde{r}_1 & (1 + a^2)\tilde{r}_0 + 2a\tilde{r}_1 \end{bmatrix}$$
Estimation error (cont.)

Example: OE model (cont.)

Note that now the input power is not given by $r_0$, but by

$$
\frac{1}{2\pi} \int_{-\pi}^{\pi} \Phi_u(\omega) d\omega
$$

$$
= \frac{1}{2\pi} \int_{-\pi}^{\pi} |1 + az^{-1}|^4 \tilde{\Phi}_u(\omega) d\omega \quad (z = e^{j\omega})
$$

$$
= \frac{1}{2\pi} \int_{-\pi}^{\pi} [a^2 z^{-2} + 2a(1 + a^2)z^{-1} + (1 + a^4 + 4a^2) + 2a(1 + a^2)z + a^2 z^2] \tilde{\Phi}_u(\omega) d\omega
$$

$$
= 2a^2 \tilde{r}_2 + 2a(1 + a^2)\tilde{r}_1 + (1 + a^4 + 4a^2)\tilde{r}_0
$$
Estimation error (cont.)

Since $\mathbf{P} = \text{cov}(\hat{\theta})$ is in general a matrix, we need to consider a scalar measure of its size. Examples:

- A-optimality: $\text{tr}[\mathbf{P}]$
- D-optimality: $\log \det[\mathbf{P}]$
- E-optimality: $\lambda_{\max}[\mathbf{P}]$
- L-optimality: $\text{tr}[\mathbf{W} \mathbf{P}]$, $\mathbf{W} = \mathbf{W}^T \succeq 0$

All these criteria are convex in $\mathbf{P}^{-1}$ and can be implemented in an SDP program.
Example: Putting it all together

Recall the previous OE example, with $y[k] = \frac{bq^{-1}}{1+aq^{-1}} u[k] + w[k]$

To find an input of power at most 1 such that the resulting estimates have a confidence ellipsoid of smallest volume, we can solve the following optimization problem:

$$\min_{\tilde{r}_0, \tilde{r}_1, \tilde{r}_2} -\log \det \begin{bmatrix} b^2 \tilde{r}_0 & -ab\tilde{r}_0 - b\tilde{r}_1 \\ -ab\tilde{r}_0 - b\tilde{r}_1 & (1 + a^2)\tilde{r}_0 + 2a\tilde{r}_1 \end{bmatrix}$$

s.t. 

$$2a^2\tilde{r}_2 + 2a(1 + a^2)\tilde{r}_1 + (1 + a^4 + 4a^2)\tilde{r}_0 \leq 1$$

This can be solved via Matlab + CVX, or other optimization packages

**Note:** This problem needs some knowledge of $a$ and $b$, the *true parameters*! Here we can plug in the estimates obtained with the initial experiment performed on the system
Example: Putting it all together (cont.)

White noise

Optimal input

Cristian R. Rojas, Input design
Signal generation

The result of optimal input design is an input spectrum $\Phi_u$. How do we translate this design into a time-domain signal?

**Method 1 (partial correlation): Multi-sine signal**
Given $r_0, \ldots, r_n$ satisfying the covariance extension constraint, it is always possible to find a multi-sine signal with $r_0, \ldots, r_n$ as first autocovariance coefficients. Its frequencies and amplitudes can be found, e.g., with MUSIC (see end of slides); for the phases, use Schroeder’s method

**Method 2 (finite parametrization): Filtered white noise**
Factorize $\Phi_u(\omega) = |F(e^{j\omega})|^2$, where $F(z) = f_0 + f_1 z^{-1} + \cdots + f_n z^{-n}$ Then, generate $u[k]$ as

$$u[k] = F(q) e[k], \quad \{e[k]\} : \text{white noise of variance 1}$$
Signal generation (cont.)

**Fact:** For a given constraint on the input amplitude (e.g., $|u[k]| \leq \alpha$), *binary signals have the largest power* ($\Rightarrow$ lowest estimation error)

**Method 3 (finite parametrization):** Receding horizon approach

*Problem:* generate a $\{-1, 1\}$-binary signal with spectrum $\Phi_u$

At time $k$, pick the values of $u[k], \ldots, u[k + N - 1] \in \{-1, 1\}$ for which the sample autocovariance

$$\hat{r}(\tau) = \frac{1}{N + k - 1} \sum_{t=\tau+1}^{N+k-1} u[t]u[t - \tau], \quad \tau = 0, \ldots, T$$

is “as close as possible” to the autocovariance sequence $r_0, \ldots, r_T$ given by $\Phi_u$

---

Signal generation (cont.)

Example: *Generation of ‘1/f’ noise*

\[
\Phi_u(\omega) = \begin{cases} 
\frac{1}{\ln \omega - \ln \omega}, & \omega \in [\omega, \overline{\omega}] \\
0, & \text{otherwise}
\end{cases}
\]
State-of-the-Art:
Application-Oriented Input Design
Taking the application into account

Designing an industrial controller involves multiple stages:

So far, we have discussed input design with standard cost functions, but... how can we really take into account the purpose of the application of our model (e.g., control)?

**Key:** Ideally, the best controller is designed when we know the process exactly \((\theta_o)\), and performance degrades if we use a model \(\theta \neq \theta_o\).
Application cost function

Let \( V_{\text{app}} : \mathbb{R}^p \rightarrow \mathbb{R}_0^+ \) be a function that measures the performance degradation when using a model \( \theta \) instead of \( \theta_0 \). We require that

\[
V_{\text{app}}(\theta_0) = 0
\]

so if it twice differentiable, it holds also that

\[
V'_{\text{app}}(\theta_0) = 0, \quad V''_{\text{app}}(\theta_0) \geq 0
\]

The purpose of input design is then to generate experiments that deliver (with high probability) models \( \hat{\theta}_N \) such that

\[
V_{\text{app}}(\hat{\theta}_N) \leq \frac{1}{\gamma}
\]

where \( \gamma > 0 \) indicates the desired accuracy of the obtained models.
System identification set

From the asymptotic theory of PEM, we know that for large sample size $N$,

$$(\hat{\theta}_N - \theta_o)^T P^{-1} (\hat{\theta}_N - \theta_o) \leq \chi^2_\alpha (p), \text{ with probability } \alpha$$

where $p =$ number of parameters, $\chi^2_\alpha (p) = \alpha$-percentile of $\chi^2$ distribution with $p$ degrees of freedom, and $P := \text{cov}(\hat{\theta}_N)$
Application-oriented input design

Within a “least-costly” framework, we then want to find an input of minimum power such that \( V_{\text{app}}(\hat{\theta}_N) \leq 1/\gamma \) with probability at least \( \alpha \). We can guarantee this by forcing that

\[
(\hat{\theta}_N - \theta_o)^T \mathbf{P}^{-1} (\hat{\theta}_N - \theta_o) \leq \chi^2_\alpha (p) \implies V_{\text{app}}(\hat{\theta}_N) \leq \frac{1}{\gamma}
\]

or, equivalently,

\[
E_{\text{id}}(\alpha) \subseteq E_{\text{app}}(\gamma)
\]

where

\[
E_{\text{id}}(\alpha) := \{ \theta : (\hat{\theta}_N - \theta_o)^T \mathbf{P}^{-1} (\hat{\theta}_N - \theta_o) \leq \chi^2_\alpha (p) \}
\]

\[
E_{\text{app}}(\gamma) := \left\{ \theta : V_{\text{app}}(\theta) \leq \frac{1}{\gamma} \right\}
\]
**Application-oriented input design (cont.)**

*Least-costly application-orient input design problem:*

\[
\begin{align*}
\min_{\text{input}} & \quad \text{input power} \\
\text{s.t.} & \quad \mathcal{E}_{\text{id}}(\alpha) \subseteq \mathcal{E}_{\text{app}}(\gamma)
\end{align*}
\]

If \( V_{\text{app}} \) is approximated by a quadratic function around \( \theta_o \):

\[
V_{\text{app}}(\theta) \approx 0.5(\theta - \theta_o)^T V''_{\text{app}}(\theta - \theta_o),
\]

then \( \mathcal{E}_{\text{id}}(\alpha) \subseteq \mathcal{E}_{\text{app}}(\gamma) \) can be approximated by

\[
\frac{1}{\chi_2^2(p)} P^{-1} \succeq \frac{\gamma}{2} V''_{\text{app}}
\]

which can be implemented in an SDP program

**Note:** A remaining problem is how to estimate the Hessian \( V''_{\text{app}} \). This can be computed in closed form for some applications (e.g., minimum variance control), but for others it can only be done via simulations.  

Example: A four-tank model

\[
\begin{align*}
\dot{x}_1 &= -\frac{a_1}{A_1} \sqrt{2gx_1} + \frac{a_3}{A_1} \sqrt{2gx_3} + \frac{\gamma_1 k_1}{A_1} u_1 \\
\dot{x}_2 &= -\frac{a_2}{A_2} \sqrt{2gx_2} + \frac{a_4}{A_2} \sqrt{2gx_4} + \frac{\gamma_2 k_2}{A_2} u_2 \\
\dot{x}_3 &= -\frac{a_3}{A_3} \sqrt{2gx_3} + \frac{(1 - \gamma_1) k_2}{A_3} u_1 \\
\dot{x}_4 &= -\frac{a_4}{A_4} \sqrt{2gx_4} + \frac{(1 - \gamma_2) k_1}{A_4} u_2 \\
y_1 &= x_1, \quad y_2 = x_2 \\
\theta &= [a_1 \cdots a_4 \gamma_1 \gamma_2 k_1 k_2]^T
\end{align*}
\]

MPC cost: 
\[
100 \sum_{i=1}^{10} \| \hat{y}[k+i|k] - r[k+i|k] \|^2 + 10 \sum_{i=0}^{9} \| \Delta \hat{u}[k+i|k] \|^2
\]

Application cost: 
\[
V_{app}(\theta) = \frac{1}{150} \sum_{t=1}^{150} \| y(t, \theta) - y(t, \theta_0) \|^2
\]

\[\text{Taken from M. Annergren's PhD thesis, KTH, 2016.}\]
Example: A four-tank model (cont.)

Responses:

Red: Gaussian white noise
Dotted: Desired response
Blue: Optimal input
MOOSE2
MOdel based Optimal input Signal dEsign Toolbox (version 2)

- MATLAB-based
- Solves optimization problems via YALMIP
- Function-based interface, including dedicated functions for
  - Application constraints
  - Quality constraints
  - Spectrum constraints
Recent Advances:
Input Design for Nonlinear Systems
Difficulties with nonlinear systems

- Non-convexity
- Constraints on model structure
- Restrictions on the input signal

How can we overcome these limitations?
Problem formulation

\[ G_o : \text{SISO system} \]
\[ G : \text{model struct.} \quad (G(\cdot; \theta_0) = G_o(\cdot)) \]
\[ e[k] : \text{white noise (variance } \lambda_e) \]
\[ u[k] : \text{input} \]
\[ y[k] : \text{system output} \]

**Goal:** Design

\[ \mathcal{U}_{n_{\text{seq}}} = (u[n_{\text{seq}}], \ldots, u[1]) \]

as the realization of a stochastic process that **minimizes** \( P = \text{cov}(\hat{\theta}_N) \)
Information matrix

\[
P^{-1} = \frac{1}{\lambda e n_{\text{seq}}} \mathbb{E} \left\{ \sum_{k=1}^{n_{\text{seq}}} \psi(U_k) \psi^T(U_k) \right\}
\]

\[
= \frac{1}{\lambda e n_{\text{seq}}} \int_{\mathbb{R}^{n_{\text{seq}}}} \sum_{k=1}^{n_{\text{seq}}} \psi(U_k) \psi^T(U_k) dP(U_{n_{\text{seq}}})
\]

\[
\psi(U_k) = \frac{d\hat{y}(U_k)}{d\theta} \bigg|_{\theta=\theta_0}, \quad U_k = (u[k], \ldots, u[1])
\]

\[
\hat{y}(U_k) = G(U_k; \theta)
\]
Information matrix (cont.)

\[ P^{-1} = \frac{1}{\lambda_e n_{\text{seq}}} \int_{\mathbb{R}^{n_{\text{seq}}}} \sum_{k=1}^{n_{\text{seq}}} \psi(U_k)\psi^T(U_k) \, dP(U_{n_{\text{seq}}}) \]

Key idea:

\( P^{-1} \) is linear in \( P(U_{n_{\text{seq}}}) \)

Difficulties:

- How to describe the set of possible \( P(U_{n_{\text{seq}}}) \)'s?
- If this set is too large, how to approximate it?
Information matrix (cont.)

The set of possible probability measures is

\[ \mathcal{P}_{\mathbb{R}^{n_{\text{seq}}}} = \left\{ P : \mathcal{B}(\mathbb{R}^{n_{\text{seq}}}) \to \mathbb{R}_0^+ \mid P \text{ is } \sigma \text{- additive}, \right. \]

\[ \left. \int_{\mathbb{R}^{n_{\text{seq}}}} dP(x) = 1 \right\} \]
Input design problem

Problem
Design $\mathcal{U}_{n_{seq}}^{opt} \in C^{n_{seq}}$ as a realization from

$$P^{opt} = \arg \min_{P \in \mathbb{P}_{\mathbb{R}^{n_{seq}}}} h(P^{-1}(P))$$

where $h : \mathbb{R}^{m \times m} \rightarrow \mathbb{R}$ is convex, and

$$P^{-1}(P) = \frac{1}{\lambda e n_{seq}} \int_{\mathbb{R}^{n_{seq}}}^{n_{seq}} \sum_{k=1}^{n_{seq}} \psi(U_k)\psi^T(U_k) dP(U_{n_{seq}})$$

Issue:

- $P^{-1}(P)$ is based on an $n_{seq}$-dimensional integral ($n_{seq} \gg 1$)
Input design problem (cont.)

To address the high dimensionality issue in $P^{-1}(P)$, we consider the following assumptions:

1. $u[1], \ldots, u[n_{\text{seq}}] \in \mathcal{C}$ (finite)

2. $\mathcal{U}_{n_{\text{seq}}}$ is the realization of a stationary Markov chain of memory $n_m$

   $(n_m \ll n_{\text{seq}})$

$\Rightarrow P^{-1}(P)$ now finitely parametrized in terms of $p \in \mathcal{P}_{C_{n_m}}^S$

Under these assumptions, the space of probability measures becomes a polyhedron, whose vertices can be efficiently characterized using graph theory.$^a$

---

$G(U_k; \theta) = G_1(q; \theta)u[k] + G_2(q; \theta)u[k]^2$

$G_1(q; \theta) = \theta_1 + \theta_2q^{-1}$

$G_2(q; \theta) = \theta_3 + \theta_4q^{-1}$

$e[k]$: Gaussian white noise, zero mean, variance $\lambda_e = 1$

$h(\cdot) = \text{det}(\cdot)^{-1}$, $c_{\text{seq}} = 3$, $n_m = 1$, $C = \{-1, 0, 1\}$ and $N = 5 \cdot 10^3$
Example (cont.)

Stationary probabilities:
Summary

- Basic input signals for system identification
- Optimal input design (via convex optimization)
- State-of-the-art: Application-oriented input design
- Recent extensions: Input design for nonlinear systems
Thank you for your attention
Bonus slide: MUSIC-like method

Let \( r_0, \ldots, r_n \) be such that \( \mathbf{R} = \mathbf{T}(r_0, \ldots, r_n) \) is singular. Then,

\[
\mathbf{Rx} = 0 \iff \mathbf{x}^H \mathbf{Rx} = 0
\]

\[
\iff \sum_{i,k=0}^{n} r_{i-k} x_i^* x_k = 0
\]

\[
\iff \sum_{i,k=0}^{n} \mathbb{E}\{u[i]u[k]\} x_i^* x_k = 0
\]

\[
\iff \sum_{i,k=0}^{n} \mathbb{E}\{(x_i u[i])^* (x_k u[k])\} = 0
\]

\[
\iff \mathbb{E}\{|G_{\mathbf{x}}(q)u[k]|^2\} = 0 \quad \text{where } G_{\mathbf{x}}(q) := \sum_{i=0}^{n} x_i q^i
\]

\[
\iff \frac{1}{2\pi} \int_{-\pi}^{\pi} |G_{\mathbf{x}}(e^{j\omega})|^2 \Phi_u(\omega) d\omega = 0
\]

\[
\iff G_{\mathbf{x}}(e^{j\omega}) = 0 \text{ for all the frequencies } \omega \text{ of the multi-sine } u[k]
\]
Therefore, to obtain the frequencies of the multi-sine $u[k]$,

1. Find a basis $\{x^1, \ldots, x^m\}$ for the nullspace of $R$ (Matlab’s `null`)
2. For each $x^i$, compute the zeros of $G_{x^i}(q) = \sum_{k=0}^{n} x^i_k q^k$
3. Collect the zeros of $G_{x^i}$ of the form $e^{j\omega}$ in the set $Z_i$
4. The frequencies of $u[k]$ are the arguments of the zeros in $\bigcap_{i=1}^{m} Z_i$

Since the autocovariance of a sinewave $\cos(\omega k + \phi)$ is $r^\omega_i := \cos(\omega i)/2$, we can determine the amplitudes of $u[k]$ by solving

$$
\begin{align*}
r_1 &= \alpha_1 r_1^{\omega_1} + \cdots + \alpha_p r_1^{\omega_p} \\
& \vdots \\
r_n &= \alpha_1 r_n^{\omega_1} + \cdots + \alpha_p r_n^{\omega_p}
\end{align*}
$$

where $p = \text{number of frequencies in } u[k]$

---

What if \( R \) is non-singular?

In this case, the representation of \( u[k] \) as a multi-sine is not unique. One way to fix a representation is to pick one of the frequencies, e.g., \( \omega = 0 \) (a constant term). This is done by finding the smallest \( c > 0 \) such that \( R - c \mathbf{T}(1, \ldots, 1) = R - c \mathbf{1} \) is singular; this is a generalized eigenvalue problem (Matlab’s \texttt{eig}).

Then, apply the same procedure as before to \( R - c \mathbf{1} \), and add the constant term \( c \) to the generated \( u[k] \).