Control of Discrete Systems

ENSICA

Yves Briere
ISAE

General Introduction
Overview

Prerequisites:
AUTOMATIQUE ENSICA 1A ET 2A
TRAITEMENT NUMÉRIQUE DU SIGNAL ENSICA 1A
(Control and signal processing, basics)

Tools:
Matlab / Simulink

References:

Planning
22 slots of 1h15

Overview

Discrete signals and systems
Sampling continuous systems
Identification of discrete systems
Closed loop systems
Control methods
Control by computer
I. Introduction

II. Discrete signals and systems

Signal processing / Control

→ Signal processing gives tools to describe and filter signals
→ Control theory use these tools to deal with closed loop systems
→ More generally, control theory deal with :
   → discrete state system analysis and control (Petri nets, etc...)
   → Complex systems, UML, etc...
II. Discrete signals and systems

What we deal with in this course

The same problem as seen by the control system engineer:

Or, simply:

II. Discrete signals and systems
II. Discrete signals and systems

Reminder: the $z$-transform

Discrete signal: list of real numbers (samples)

$$s(k) = \{s_0, s_1, s_2, \ldots\}$$

$Z$-transform: function of the complex $z$ variable

$$s(z) = Z(s(k)) = \sum_{k=0}^{\infty} s(k) \cdot z^{-k}$$

Existence of $s(z)$: generally no problem (convergence radius: $s(z)$ exist for a given radius $|z| > R$)

Reminder: properties of the $z$-transform

1. $s(z) = Z(s(k)) = \sum_{k=0}^{\infty} s(k) \cdot z^{-k}$
2. $Z(\alpha \cdot s_1(k) + \beta \cdot s_2(k)) = \alpha \cdot Z(s_1(k)) + \beta \cdot Z(s_2(k))$
3. $Z(k \cdot s(k)) = -z \frac{ds(z)}{dz}$
4. $Z(z^k \cdot s(k)) = \left(\frac{z}{c}\right)^k$
5. $Z(s(k-1)) = s(k=1) + z^{-1} \cdot s(z)$
6. $Z(s(k+1)) = z \cdot s(z) - z \cdot s(k=0)$
7. $s(k=0) = \lim_{z \to 0} (z^{-1} \cdot s(z))$
8. $s(k \to \infty) = \lim_{z \to \infty} \left(\frac{z^{-1}}{z} \cdot s(z)\right)$

Linearity

Delay theorem

Final value theorem
II. Discrete signals and systems

Reminder: basic signals

<table>
<thead>
<tr>
<th>$s(k)$</th>
<th>$S(z)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\delta(k)$: unit impulse</td>
<td>1</td>
</tr>
<tr>
<td>$u(k)$: unit step</td>
<td>$\frac{z}{z-1}$</td>
</tr>
<tr>
<td>$k \cdot u(k)$</td>
<td>$\frac{z}{(z-1)^2}$</td>
</tr>
<tr>
<td>$c^k \cdot u(k)$</td>
<td>$\frac{z}{z-c}$</td>
</tr>
<tr>
<td>$\sin(\omega \cdot k) \cdot u(k)$</td>
<td>$\frac{z \cdot \sin(\omega)}{z^2 - 2 \cdot z \cdot \sin(\omega) + 1}$</td>
</tr>
</tbody>
</table>

II. Discrete signals and systems

Reminder: from « z » to « k »

First approach: use the z-transform equations (we don’t give here the inverse z-transform equation, it is too ugly...)

Second approach: use tricks
   → recurrence inversion
   → Polynoms division
   → Singular value decomposition

Third approach: computer (Matlab)
II. Discrete signals and systems

Reminder: discrete transfer function

We deal with Linear Time Invariant (LTI) systems

\[ e(k) \rightarrow \text{Discrete LTI system} \rightarrow s(k) \]

The sequence of output and input samples are consequently simply related by:

\[
s(k) + a_1 \cdot s(k-1) + a_2 \cdot s(k-2) + \ldots + a_n \cdot s(k-n) = \\
b_0 \cdot e(k) + b_1 \cdot e(k-1) + b_2 \cdot e(k-2) + \ldots + b_m \cdot e(k-m)
\]

The delay theorem gives:

\[
\frac{s(z)}{c(z)} = F(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \ldots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \ldots + a_n \cdot z^{-n}}
\]

Normalized: \( a_0 = 1 \)

Remarks

→ We prefer to use \( z^{-1} \) rather than \( z \). \( z^{-1} \) is a « shift » operator
→ We often use the \( q^{-1} \) notation instead of \( z^{-1} \): this way we don’t bother with radius convergence and other fundamental mathematic stuff.
→ Impulse response: \( e(z) = 1 \)
  The transfer function is also the impulse response
  (function = signal)
→ Causality: the output depends on past, not future
  → the impulse response is null for \( k < 0 \)
  → Confusion between « causal system » and « causal signal »
II. Discrete signals and systems

Reminder: discrete transfer function

Properties

→ Impulse response: the inverse z-transform of the transfer function

→ Step response

\[
s(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \ldots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \ldots + a_n \cdot z^{-n}} \times \frac{1}{1 - z^{-1}}
\]

→ Static gain (Final value theorem applied to the last equation)

\[
F_0 = \frac{\sum_{i=0}^{m} b_i}{\sum_{i=1}^{n} a_i}
\]

II. Discrete signals and systems

Reminder: discrete transfer function

Properties: stability

→ Any transfer function can be expressed as:

\[
F(z) = \frac{c_0 \cdot (1 + c_1 \cdot z^{-1}) \cdot (1 + c_2 \cdot z^{-1}) \ldots (1 + c_m \cdot z^{-1})}{(1 + c_{i1} \cdot z^{-1}) \cdot (1 + c_{i2} \cdot z^{-1}) \ldots (1 + c_{i \cdot n} \cdot z^{-1})}
\]

Coefficients \( c_i \) are either real or complex conjugates

For a stable system each \( c_i \) coefficient must verify \( |c_i| < 1 \), in other words each pole must belong to the unit circle.

Properties: singular value decomposition

\( F(z) \) can be decomposed in a sum of first order and second order systems

→ It is good to know how first and second order behaves
II. Discrete signals and systems

Reminder: first order systems

Properties: first order system

Reminder: second order systems

Properties: second order systems
II. Discrete signals and systems

Reminder: second order systems

Properties: second order systems

Reminder: frequency response

Reminder: continuous systems

\[
\begin{cases}
  c(t) = \cos(2 \cdot \pi \cdot f \cdot t) + j \cdot \sin(2 \cdot \pi \cdot f \cdot t) = e^{j2\pi ft} \\
  f \in [0, \infty]
\end{cases}
\]

Analogy: discrete system

\[
\begin{cases}
  c(k) = \cos(2 \cdot \pi \cdot f \cdot k) + j \cdot \sin(2 \cdot \pi \cdot f \cdot k) = e^{j2\pi fk} \\
  f \in [0, 1]
\end{cases}
\]

The output signal looks like

\[
\begin{cases}
  s(k) = A(f) \cdot e^{j(2\pi f k + \phi(f))} \\
  f \in [0, 1]
\end{cases}
\]

With \( F \) of phase \( \phi \) (degrees) and module \( A \) (decibels):

\[
\begin{cases}
  F(f) = F(z = e^{j2\pi f}) \\
  f \in [0, 1]
\end{cases}
\]
II. Discrete signals and systems

Reminder: frequency response

Definition: frequency response

\[
F(f) = \mathcal{F}\{e^{j2\pi f}\}
\]

- \(f\): normalized frequency (between 0 and 1)
- \(F(f)\): frequency response

Property: Bode diagram

Bode diagram of a second order system. Note the linear scale for frequency.

---

II. Discrete signals and systems

Reminder: Nyquist-Shannon sampling theorem
III. Sampling continuous systems
A real system is generally continuous (and described by differential equations)

An interface provides sampled measurements at a given frequency:
\[\to\] Digital to Analog Converter (DAC)
\[\to\] smart sensor

A computer provides at a given frequency the input of the system

An interface transforms the computer output into a continuous input for the system
\[\to\] Analog to Digital Converter (ADC)
\[\to\] smart actuator

\[\to\] Sampling is regular
\[\to\] Sampling is synchronous
\[\to\] The computer computes the control according to the current measurement and a finite set of past measurements.
III. Sampled continuous systems

Sampling a continuous transfer function

Hypothesis :
- The continuous transfer function is known
- The ADC is a ZOH

We must now deal with F+ZOH in a whole

![Diagram](ZOH F(p))

Zero Holder Hold: the sampled input is blocked during one sampling delay.

The impulse response of a zoh is consequently:

\[ ZOH(s) = \frac{1}{s} e^{-\frac{t}{s}} \]

The Laplace transform of the above signal is:
III. Sampled continuous systems

Sampling a continuous transfer function

The impulse response of ZOH+F is:

\[
s(s) = \left( \frac{1}{s} - e^{-Ts} \right) \cdot F(s) = \frac{1}{s} \cdot F(s) - e^{-Ts} \cdot \frac{1}{s} \cdot F(s)
\]

Two terms for the impulse response:

\[
\begin{align*}
s(k) &= Z \left( \frac{1}{s} \cdot F(s) \right) - z^{-1} \cdot Z \left( \frac{1}{s} \cdot F(s) \right) = \left( 1 - z^{-1} \right) \cdot Z \left( \frac{1}{s} \cdot F(s) \right)
\end{align*}
\]

Usually the z-transfer function of ZOH+F is given by tables:

<table>
<thead>
<tr>
<th>F(s)</th>
<th>(F_{ZOH}(z))</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\frac{1}{s})</td>
<td>1</td>
</tr>
<tr>
<td>(\frac{1}{1 + T \cdot s})</td>
<td>(\frac{\text{b}_1}{1 + \text{a}_1 \cdot z^{-1}})</td>
</tr>
<tr>
<td>(\frac{e^{-T_s}}{1 + T \cdot s})</td>
<td>(\frac{\text{b}_1 \cdot z^{-1} + \text{b}_2 \cdot z^{-2}}{1 + \text{a}_1 \cdot z^{-1}})</td>
</tr>
<tr>
<td>(L &lt; T_s)</td>
<td>(</td>
</tr>
</tbody>
</table>

Matlab:

- % Definition of a continuous system
  \(\text{sysc}=\text{tf}(1,[1 1]);\)
- % z-transfer function after sampling
  \(\text{Te}=0.1;\)
  \(\text{sysd}=\text{c2d(sysc,Te,'zoh'});\)
- % Notation with \(z^{\cdot-1}\)
  \(\text{sysd.variable}=z;\)
III. Sampled continuous systems

Sampling time delay equivalence

A sampling time of $T_s$ can be approximated as a delay of $T_s/2$:

- instability in closed loop.*

![Nyquist Diagram](image)

```matlab
>> F=2*s/(1+s+s^2)/s/(1+s)
>> G=F;
>> G.ioDelay=0.5
>> nyquist(F,G)
```

III. Sampled continuous systems

Choice of sampling time

The sampling time is chosen according to the closed loop expected performance.

Sampling frequency: 5 to 25 times the expected closed loop bandwidth.

Example: third order system

```matlab
>> F=1.5/(1+s+s^2)/(1+s)
```

Closed loop bandwidth at -3dB of $F/(1+F)$: 0.3 Hz

Sampling time: 5 Hz
III. Sampled continuous systems

Choice of sampling frequency

Bode Diagram

System: F
Frequency (Hz): 0.196
Magnitude (dB): -3

System: untitled1
Frequency (Hz): 0.269
Magnitude (dB): -7.36

Choice of sampling frequency

Real Axis

System: F
Gain: 1.64
Pole: -0.0488 + 1.35i
Damping: 0.0362
Overshoot (%): 89.2
Frequency (rad/sec): 1.35

System: F
Gain: 1.69
Pole: -1.92
Damping: 1
Overshoot (%): 0
Frequency (rad/sec): 1.92
III. Sampled continuous systems

Choice of sampling frequency

Sampling effect: discretization

Sampling is always associated to a discretization of the signal!
Example:
- Signal sampled at 0.2s and discretized at a resolution of 0.1.
- The numerical derivative of the signal
III. Sampled continuous systems

First approach: discretise a continuous controller

The sampling effect (ZOH) is neglected.

Example: continuous PD:

\[ \text{PID}(s) = \frac{e(s)}{\epsilon(s)} = k_p + k_D \cdot s \]

\[ \epsilon(t) = k_p \cdot \epsilon(t) + k_D \frac{d\epsilon(t)}{dt} \]

Continuous derivative is replaced by a discrete derivative:

\[ \epsilon(k) = k_p \cdot \epsilon(k) + k_D \cdot \frac{\epsilon(k) - \epsilon(k-1)}{T_e} \]

Transfer function:

\[ c(z) = k_p \cdot \epsilon(z) + k_D \cdot \frac{\epsilon(z) - z^{-1}\epsilon(z)}{T_e} = \left( k_p + k_D \cdot \frac{1 - z^{-1}}{T_e} \right) \cdot \epsilon(z) \]

S to z equivalence:

\[ s = \frac{1 - z^{-1}}{T_e} \]

III. Sampled continuous systems

More generally the problem is to approximate a differential equation: cf numerical methods of integration

<table>
<thead>
<tr>
<th>Method</th>
<th>s to z equivalence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Euler's forward method</td>
<td>( s = \frac{1 - z^{-1}}{z^{-1} \cdot T_e} )</td>
</tr>
<tr>
<td>Euler's backward method</td>
<td>( s = \frac{1 - z^{-1}}{T_e} )</td>
</tr>
<tr>
<td>Tustin method</td>
<td>( s = \frac{2}{T_e} \cdot \frac{1 - z^{-1}}{1 + z^{-1}} )</td>
</tr>
</tbody>
</table>
III. Sampled continuous systems

First approach: discretise a continuous controller

Warning: methods are not equivalent!

(No problem if sampling time is small)

Equivalence of continuous poles after « s to z » conversion

(a) : continuous
(b) : Euler forward
(c) : Euler backward
(d) : Tustin
III. Sampled continuous systems

End C3 C4

IV. Identification of discrete systems
IV. Identification of discrete systems

Introduction

Before the design process of a controller one must have a discrete model of the plant

→ First approach : continuous model and/or identification then discretization (cf last chapter)

→ Second approach : tests on the real system, measurements (sampled), identification algorithm

For a full course on this topic see 3d year course “airplane identification”

Naive approach

(But easy, fast and easy to explain)

→ Standard test (step, impulse)

→ Try to fit as well as possible a model (first order, second order with delay, etc...)

... « fit as well as possible » ...

→ criteria ?...

→ optimisation ?...
IV. Identification of discrete systems

Naive approach

Example: real step test (o) and simulated first order step (x)

\[
J = \frac{1}{N} \sum_{t=1}^{N} (y_{\text{réel}}(t) - y_{\text{mesuré}}(t))^2
\]

After some iterations one obtain J minimal with \( \tau = 1.81 \) s

Better model with a delay, two parameters to tune
IV. Identification of discrete systems

Naive approach

Drawbacks:
→ Test signals with high amplitude (is the system still linear?)
→ Reduced precision
→ Perturbation noise not taken into account
→ Perturbation noise reduces the performance of the identification
→ Long, not very rigorous...

Advantage:
→ Easy to understand...

Second method (the good one) : « estimation »

→ Perform a « rich test » (random input signal, rich in frequency)
→ Fit the model that best predict the output.
IV. Identification of discrete systems

Basic principle of parametric identification

- Physical system
- Numerical model
- Parameters of the model

IV. Identification of discrete systems

Example of an Identification Process

**Linear model** (ARX):

\[ A \cdot y = B \cdot u + e \]

White noise

\[ y(t+1) + a_1 \cdot y(t) + a_2 \cdot y(t-1) + \ldots + a_{n_y} \cdot y(t-n_y+1) = b_1 \cdot u(t) + b_2 \cdot u(t-1) + \ldots + b_{n_u} \cdot u(t-n_u+1) + e(t) \]

Model used as a **predictor**:

\[ y_{\text{predict}}(t+1) = -\left( a_1 \cdot y(t) + a_2 \cdot y(t-1) + \ldots + a_{n_y} \cdot y(t-n_y+1) \right) + \]

\[ \ldots \ldots b_1 \cdot u(t) + b_2 \cdot u(t-1) + \ldots + b_{n_u} \cdot u(t-n_u+1) \]

→ Try to minimize the difference between prediction and actual measurement.
IV. Identification of discrete systems

Example of an Identification Process

Example : orders \( n_a = 3 \) et \( n_b = 2 \)

<table>
<thead>
<tr>
<th>measurements</th>
<th>prediction</th>
<th>error</th>
</tr>
</thead>
<tbody>
<tr>
<td>u1 ( y_1 )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>u2 ( y_2 )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>u3 ( y_3 )</td>
<td>( \text{yp3} )</td>
<td>( \text{eps3} )</td>
</tr>
<tr>
<td>u4 ( y_4 )</td>
<td>( \text{yp4} )</td>
<td>( \text{eps4} )</td>
</tr>
<tr>
<td>u5 ( y_5 )</td>
<td>( \text{yp5} )</td>
<td>( \text{eps5} )</td>
</tr>
<tr>
<td>u6 ( y_6 )</td>
<td>( \text{yp6} )</td>
<td>( \text{eps6} )</td>
</tr>
<tr>
<td>u7 ( y_7 )</td>
<td>( \text{yp7} )</td>
<td>( \text{eps7} )</td>
</tr>
<tr>
<td>u8 ( y_8 )</td>
<td>( \text{yp8} )</td>
<td>( \text{eps8} )</td>
</tr>
<tr>
<td>Total</td>
<td>( J )</td>
<td></td>
</tr>
</tbody>
</table>

\[
y_{5\text{predict}} = -(a_1 \cdot y_1 + a_2 \cdot 2 + a_3 \cdot y_3) + \ldots + b_1 \cdot u_1 + b_2 \cdot u_2
\]
\[
\text{eps5} = y_{5\text{predict}} - y_5
\]
\[
J = \frac{1}{N} \sum_{i=1}^{N} (\text{eps}_i)^2
\]

Unknown : coefficients \( \{ a_1, a_2, a_3, b_1, b_2 \} \)

Recursive Least Square Algorithm

→ Possible since model is linear
→ Possible since quadratic criteria
→ Interesting because of the recursive version

\[
y_{\text{predict}}(t+1) = -(a_1 \cdot y(t) + a_2 \cdot y(t-1) + \ldots + a_{n_a} \cdot y(t-n_a+1)) + \ldots + b_1 \cdot u(t) + b_2 \cdot u(t-1) + \ldots + b_{n_b} \cdot u(t-n_b+1)
\]

Prediction equation can be re-written:

\[
y_{\text{predict}} = \theta^T(t) \cdot \phi(t)
\]

Where:

\[
\theta(t)^T = [a_1, a_2, \ldots, b_1, b_2, \ldots]
\]

\[
\phi(t)^T = [-y(t) - y(t-1) - u(t) - u(t-1), \ldots]
\]
IV. Identification of discrete systems

Least Square Algorithm

Estimate $\theta$ taking into account the full set of measurements

$\hat{\theta} = F(t) \sum_{i=1}^{t} (y(i) \cdot \varphi(i-1))$

With:

$F(t)^{-1} = \sum_{i=1}^{t} \varphi(i-1) \cdot \varphi(i-1)^T$

Recursive Least Square (rls) Algorithm

Estimate $\theta$ recursively for $t = 0...N$

$\hat{\theta}(t) = \hat{\theta}(t-1) + F(t) \cdot \varphi(t) \cdot \left( y(t) - \hat{\theta}(t-1) \cdot \varphi(t) \right)$

with:

$F(t)^{-1} = F(t-1)^{-1} + \varphi(t) \cdot \varphi(t)^T$

Algorithm can be applied in real time

$\rightarrow$ parameters supervision

$\rightarrow$ diagnostics

Algorithm can be improved with a forget factor: $\lambda = 0.95...0.99$

$\hat{\theta}(t) = \hat{\theta}(t-1) + F(t) \cdot \varphi(t) \cdot \left( y(t) - \hat{\theta}(t-1) \cdot \varphi(t) \right)$

with:

$F(t)^{-1} = \lambda \cdot F(t-1)^{-1} + \varphi(t) \cdot \varphi(t)^T$
IV. Identification of discrete systems

Identification algorithms (general)

Les algorithmes d’identification sont en général des variantes des moindres carrés récursifs
En tous cas ils ont tous, pour les algorithmes récursifs, la forme suivante :

\[ \theta(t) = \theta(t-1) + F(t) \cdot \varphi(t) \cdot \left( y(t) - \theta^T(t-1) \cdot \varphi(t) \right) \]
avec
\[ F(t)^{-1} = \lambda \cdot F(t-1)^{-1} + \varphi(t) \cdot \varphi(t)^T \]

IV. Identification of discrete systems

Identification method

1. Choice of a frequency sampling
   - Not too big
   - Not too small
   - Adapted to the fastest expected dynamic (usually closed loop dynamic)
IV. Identification of discrete systems

Identification method

2. Excitation signal
   → Le signal doit être riche en fréquence
   → Signal can (and must) be of small amplitude (a few %)

Example 1 : Pseudo Random Binary Signal

Matlab / System Identification Toolbox
>> u=idinput(500,'prbs')

Example 2: excitation 3-2-1-1 (airplane identification)

3. Test and measurements
   → Choice of a setpoint
   → Test
   → Remove mean of measurements
   → Remove absurd values

4. Model choice
   → Example : ARX, degree of A and B

5. Apply algorithm
   → Example : rls

6. Validate
   → Validation criteria?

Iterate
IV. Identification of discrete systems

Validation criteria

1. The model must predict the output

2. The model behavior must be the same than the original for perturbations

The model:

\[ y(t+1) + a_1 \cdot y(t) + a_2 \cdot y(t-1) + \ldots + a_n \cdot y(t-n_a + 1) = \\
 b_1 \cdot u(t) + b_2 \cdot u(t-1) + \ldots + b_n \cdot u(t-n_b + 1) + e(t) \]

With the estimated model \((a_i, b_i)\) one can estimate the error!

\[ e_{est}(t) = y_{est}(t+1) - y(t+1) \]
\[ c_{est}(t) = a_1 \cdot y(t) + a_2 \cdot y(t-1) + \ldots + a_n \cdot y(t-n_a + 1) \]
\[ \ldots \ldots \ldots - b_1 \cdot u(t) + b_2 \cdot u(t-1) + \ldots + b_n \cdot u(t-n_b + 1) \]
\[ \ldots \ldots \ldots - y(t+1) \]

→ Check that \( e_{est}(t) \) is a white noise!

Use dedicated software

Example: Matlab / System Identification Toolbox...
IV. Identification of discrete systems

Use dedicated software

- Accelerate the identification process
  Matlab licence = 3000 € + toolboxes licence...
- "homemade" toolboxes are used for dedicated applications
IV. Identification of discrete systems

Non parametric identification

From input/output datas, estimate the transfer function :

→ Input / output cross correlation
  → FFT, Hamming window and tutti quanti
  → Gain and phase, Bode diagram

From input / output datas, estimate the impulse response

→ Apply RLS algorithm with $n_b$ big and $n_a = 0$
→ Allow for instance to quickly estimate the pure delay

V. Closed loop systems
V. Closed loop systems

Reminder

Open loop transfer function:
\[ F = S \cdot F_1 \cdot F_2 \cdot F_3 \cdot T \]

Closed loop transfer function:
\[ G = \frac{F}{1+F} \]
\[ G = \frac{B}{A+B} \]
V. Closed loop systems

Static gain

As seen before:

\[ G_0 = \frac{\sum_{i=0}^{\infty} b_i}{1 + \sum_{i=0}^{\infty} b_i + \sum_{i=1}^{\infty} a_i} \]

Condition to have a unitary closed loop static gain (no static error):

\[ G_0 = \frac{\sum_{i=0}^{\infty} b_i}{1 + \sum_{i=0}^{\infty} b_i + \sum_{i=1}^{\infty} a_i} = 1 \]

\[ F(z) = \frac{s(z)}{c(z)} = \frac{1}{1 - z^{-1}} \cdot \frac{B(z)}{A'(z)} \]

Integrator

Perturbation rejection

Perturbation-output transfer function

\[ S_{yp}(z) = \frac{s(z)}{p(z)} = \frac{A(z^{-1})}{A(z^{-1}) + B(z^{-1})} \]

⇒ Same condition on A(1) to perfectly reject the perturbations (integrator in the loop)

Generally the rejection perturbation condition is weaker:

⇒ |S_{yp}(e^{j2\pi f})| < S_{max} for a subinterval of [0 0.5]
V. Closed loop systems

Stability: poles

System is stable if the poles belong to the unit circle
→ Compute closed loop transfer function
 → Jury criteria (equivalent to Routh)
 → Compute poles
 → Draw the root locus

Example: root locus

System: sys
Gain: 0.0169
Pole: 0.564 + 0.281i
Damping: 0.706
Overshoot (%): 4.35
Frequency (rad/sec): 0.654

System: sys
Gain: 0.0162
Pole: 0.196
Damping: 1
Overshoot (%): 0
Frequency (rad/sec): 1.63
V. Closed loop systems

Stability: frequency criteria

Nyquist criteria: as in the continuous case

Margins:
- Module margin $\Delta M$
- Gain module $\Delta G$
- Phase module $\Delta P$

V. Closed loop systems

Nyquist criteria

Margins:
V. Closed loop systems

end C5 C6

VI. Control of sampled systems
VI. Control of closed loop systems

Hypothesis

Requisites:

→ Synchronous sampling
→ Regular sampling
→ Sampling time adapted to the desired closed loop performance

VI. Control of closed loop systems

Introduction

Controller computes $e(t)$ given $r(t)$ and $s(t)$:

If $K$ is linear, the RST form is canonical:

R, S et T are polynomials
VI. Control of closed loop systems

Discretization of a continuous controller

1. Process is modelled as a linear transfer function (Laplace)
   → No need to know a sampled model of the system
2. Design of a continuous controller
3. Discretization of the continuous controller
   → Many discretization methods
   → Problem if $T_e$ too big (stability)
   → Problem if $T_e$ too small (numerical round up)

VI. Control of closed loop systems

Design of a discrete controller

1. System is known as a discrete transfer function (z-transform)
   → Discretization of a continuous model + ZOH
   → Direct identification
2. Design of a discrete controller
   1. PID 1
   2. PID 2
   3. Pole placement
   4. Independant tracking and regulation goals
VI. Control of closed loop systems

PID 1

Continuous PID:

\[ K_{PID}(s) = K \left( 1 + \frac{1}{T_1} \cdot s + \frac{T_d \cdot s}{1 + \frac{T_d}{N} \cdot s} \right) \]

Discretization (Euler):

\[ K_{PID}(q) = K \left( 1 + \frac{1}{T_1} \cdot q^{-1} + \frac{T_d \cdot q^{-1}}{1 + \frac{T_d}{N} \cdot q^{-1}} \right) \]

After re-arranging:

\[ r_0 = K \left( 1 + \frac{T_c}{T_1} + \frac{N \cdot T_d}{T_d + N \cdot T_c} \right) \]

\[ r_1 = -K \left( 1 + \frac{T_d}{T_1} + \frac{N \cdot T_d}{T_d + N \cdot T_c} \right) \]

\[ r_2 = K \left( \frac{T_d}{T_d + N \cdot T_c} \right) \]

\[ s_i = \frac{T_d}{T_d + N \cdot T_c} \]
VI. Control of closed loop systems

PID 1

Closed loop transfer function is:

\[ H_{CL} = \frac{B \cdot R}{A \cdot S + B \cdot R} = \frac{B \cdot R}{P} \]

S as an integral term (the I of PID):

\[ P = A \cdot S + B \cdot R = A \cdot \left(1 - z^{-1}\right) \cdot S + B \cdot R \]

Desired dynamics is given by the roots of P:

\[ A, B, P \to R \cdot S' \]

Small degrees: by hand
Large degrees: Bezout algorithm

Main drawback: new zeros given by R at the denominator...
Solution: PID2
VI. Control of closed loop systems

PID 2

Starting from the already known PID1:

\[
\begin{align*}
\text{r}(t) & \rightarrow R \rightarrow \frac{1}{S} \rightarrow B/A \rightarrow s(t) \\
\end{align*}
\]

R is replaced by \( R(1) \):

\[
H_{CL} = R(1) \cdot \frac{B}{P}
\]

Static gain is one, desired dynamic remains the same

→ Same performances in rejection perturbation
→ Better performance (smaller overshoot) in tracking

Pole placement

Can be seen as a more general PID 2 where degrees of R and S are not constrained.

\[
H_{CL} = \frac{q^{-d} \cdot B(q^{-1})}{A(q^{-1})}
\]

\[
\begin{align*}
A(q^{-1}) &= 1 + a_1 \cdot q^{-1} + \ldots + a_n \cdot q^{-n} \\
B(q^{-1}) &= b_1 \cdot q^{-1} + \ldots + b_m \cdot q^{-m} = q^{-1} \cdot B'(q^{-1})
\end{align*}
\]

Tracking performance

\[
H_{CL} = \frac{q^{-d} \cdot B(q^{-1}) \cdot T(q^{-1})}{P(q^{-1})}
\]

\[
P(q^{-1}) = A(q^{-1}) \cdot S(q^{-1}) + q^{-d} \cdot B(q^{-1}) \cdot R(q^{-1}) = 1 + p_1 \cdot q^{-1} + \ldots + p_n \cdot q^{-n}
\]

Regulation performance:

\[
S_{yy} = \frac{A(q^{-1}) \cdot S(q^{-1})}{P(q^{-1})}
\]
VI. Control of closed loop systems

Pole placement

P poles gives the closed loop dynamic:

→ $P_D$ Dominant poles (second order, natural frequency, damping)

→ $P_F$ auxiliary poles, faster

$A, B, P \rightarrow P :$ compute $R$ and $S$

Static gain:

$S = (1-q^{-1}).S'$

Rejection of harmonic perturbation

$S = H_S.S'$ where $|H_S|$ is small at a given frequency

Remove sensibility to an given frequency:

$R = H_R.R'$ where $|H_R|$ is small at a given frequency
VI. Control of closed loop systems

Pole placement

Tracking : choice of $T'$

1. $T'$ is chosen such as the transfer $y^* \rightarrow y$ is as « transparent » as possible
2. $B'/A'$ chosen such as $r \rightarrow y^*$ as the desired dynamic

\[
T' = \frac{B'/A'}{R}
\]

\[
1.\ T' \ is \ chosen \ such \ as \ the \ transfer \ y^* \rightarrow y \ is \ as \ « \ transparent \ » \ as \ possible
2.\ B'/A' \ chosen \ such \ as \ r \rightarrow y^* \ as \ the \ desired \ dynamic
\]
VI. Control of closed loop systems

Placement de pôle

Tracking : choice of $A^*$ and $B^*$ :

→ chosen according to tracking specifications

---

VI. Control of closed loop systems

Indepant tracking and perturbation rejection specifications

$P$ chosen to cancel poles of $B^*$

→ $B$ must be stable

Tracking : choice of $T$

$$S_{yy}(q^{-1}) = T(q^{-1}) \frac{q^{-d} \cdot q^{-1} \cdot B^* (q^{-1})}{P(q^{-1}) \cdot B^* (q^{-1})}$$

With :

$$T(q^{-1}) = P(q^{-1})$$

One obtain :

$$S_{yy}(q^{-1}) = q^{-(d+1)}$$