Robust Control with Classical Methods – QFT

Per-Olof Gutman

- Review of the classical Bode-Nichols control problem
- QFT in the basic Single Input Single Output (SISO) case
- Uncertainty and Fundamental Design Limitations
- QFT for non-minimum phase and computer controlled systems
- QFT for cascaded systems, and for a class on non-linear plants
- QFT for Multi-Input Multi-Output (MIMO) plants
- A comparison between QFT and other robust and adaptive control

QFT for non-minimum phase and computer controlled systems

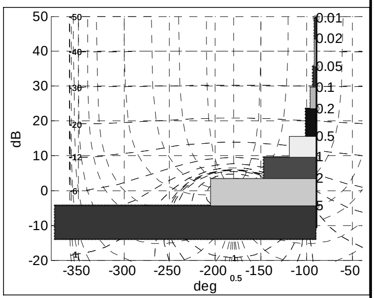
- **QFT for non-minimum phase and delay systems**
  - Design guidelines
  - An example
  - Exercises
- **Digital controller design**
  - Structure
  - Selection of sampling interval
  - Design alternatives
    ♥ **Analog design with controller translation**
    ♥ **Analog plant with digital controller**
    ♥ **Sampled plant model with digital controller**

QFT for non-minimum phase and delay systems - Design guidelines

- **Plant uncertainty** – as for MP plants.
  An MP nominal may be chosen, even if it is outside plant value set.
- **”Standard” closed loop gain specifications** – as for MP plants. Example: $a(\omega) \le |F(j\omega)S(j\omega)| \le b(\omega)$
  But: **ignore delay or undershoot** in transient.
- **Bounds: as usual**. Note that in case of an MP nominal, the bounds reflect the ”$\omega_c < \omega_{c\_crit}$” limitation.

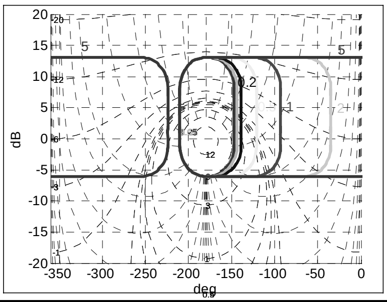
An example

- $P(s) = \frac{k e^{-\tau s}}{s}$
  $k \in [1,3],\ \tau \in [0,1]$
  $P_{nom}(s) = \frac{3}{s}$
- Spec: |S|<6dB
  max. ω_c



An examle, cont’d

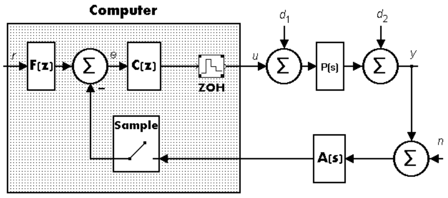
- Bounds creep ”rightwards”, reflecting the ω_c-limitation when loop shaping the MP nominal.
- roll-off vs. phase.



Exercises

- Complete the design of the example.
  What cross-over frequency, ω_c, did you achieve? What closed loop bandwidth?
- Redo the example with $P_{nom}(s) = \frac{3e^{-s}}{s}$.
  Compare the bounds, and explain the difference.
- Replace $e^{-\tau s}$ with $(1 - \tau s/2)/(1 + \tau s/2)$ and redo the design. Compare!

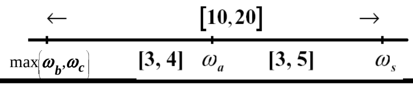
Digital controller design - structure



- Do not forget the analog anti-aliasing filter A(s)!

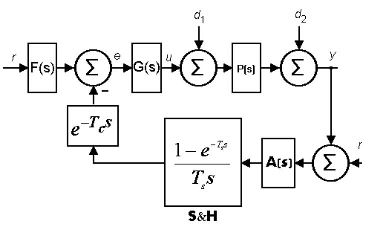
Selection of sampling interval

- **Make a preliminary analog design of $G(s)$ and $F(s)$ without anti-aliasing filter.**
- **Record the achieved closed loop bandwidths, ω_b, and cross-over frequencies, ω_c, for all plant cases.**
- **Select the sampling frequency** $\omega_s \in [10,20]\max(\omega_b, \omega_c)$
- **Design the anti-aliasing filter $A(s)$ with a bandwidth** $\omega_a \in [0.2, 0.33]\omega_s$



Design alternatives:

1. Analog design with controller translation



- **For implementation G(s) → C(z) , F(s) → F(z)**

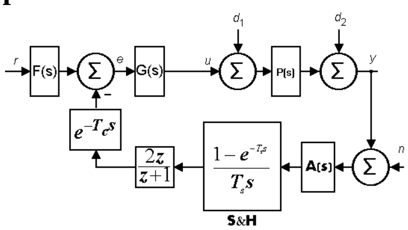
"Cancellation" of S&H effect

- Consider $\boldsymbol{R(z)=2z/(z+1)}$
- $\arg\left(R\left(e^{j\omega T_s}\right)\right) = +\omega T_s/2 = -\arg(S\&H)$
- $\left|R\left(e^{j\omega T_s}\right)\right| \approx 1/|S\&H|$ for low frequencies
- Note, by first order Padé and Tustin:

$$\frac{1-e^{-sT_s}}{T_s s} \approx \frac{1}{1+(T_s s)/2} = \left\{s = \frac{2}{T_s}\frac{z-1}{z+1}\right\} = \frac{z+1}{2z}$$

- Ref: Raviv and Djaja, Control Systems Magazine, June 1999

Analog design with controller translation, complemented with S&H "cancellation"



interpreted with $s=j\omega$, $z=e^{j\omega T_s}$ for $\omega<\omega_N$

Controller translation G(s) → C(z)

- **Recommended: use matched pole-zero translation of each factor in $G(s)$:**

$$(s-a) \quad \rightarrow \quad (z-e^{aT_s})$$

with adjustment of DC-gain, e.g.

$|G(0)| = |C(1)|$, **or** $|s\,G(s)|_{s\to 0} = |(z-1)C(z)|_{z\to 1}$, ...

- **Gives best frequency function matching up to almost ω_N.**
- **In Matlab** c2d(, , ,'matched');

TECHNION
Israel Institute of Technology

Matched pole-zero translation Example

$$(s-a) \quad \rightarrow \quad (z-e^{aT_s})$$

$$G(s)=\frac{(1+2\cdot 0.6s/4+s^2/16)}{s(1+s/3)} \quad \rightarrow$$

$$\rightarrow \quad C(z)=g\,\frac{\left(z-e^{(-2.4+3.2j)T_s}\right)\left(z-e^{(-2.4-3.2j)T_s}\right)}{(z-1)\left(z-e^{-3T_s}\right)}$$

$$=g\,\frac{\left(z^2-2e^{-2.4T_s}\cos(3.2T_s)z+e^{-4.8T_s}\right)}{(z-1)\left(z-e^{-3T_s}\right)}$$

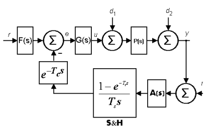
TECHNION
Israel Institute of Technology

Other controller translation methods

- Ref: Katz, Hanselmann, Franklin *et al*.
- **Prewarped Tustin for each factor**, with pre-warping frequency = ω_{knee}, **≡ matched**.
- **ZOH** assumes input to $G(s)$ to be constant over T_s: false. Freq. fcn matching bad. **Do not use!**
- **FOH** assumes input to $G(s)$ to be ramped over T_s. **Avoid!**
- **Tustin or Prewarped Tustin of all of $G(s)$ in one go**: very bad matching of frequency function. **Do not use!**

TECHNION
Israel Institute of Technology

Analog design with controller translation in Qsyn



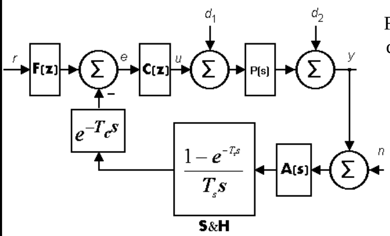
- Before translation:

```
function [G] = d3(s)
% d3.m Controller function description of anti-aliasing filter
% in series with the sample and hold transfer function, and
% analog compensator. No computational delay.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
A = [1]./(1 + 2*0.7*s/100 + s.*s/10000);% anti-aliasing
Ts = 0.020;                          % seconds, sampling period
H = (1 - exp(-Ts*s))./(Ts*s);        % Sample and hold
Ganal = (1 + 2*0.6*s + s.*s/16)./(s.*(1+s/3));
G = A.*H.*Ganal;
```

after matched translation ...

TECHNION
Israel Institute of Technology

Design alternatives:
2. Analog plant with digital controller



Frequency function calculation for $\omega<\omega_N$: $s=j\omega$, $z=e^{j\omega T_s}$

TECHNION
Israel Institute of Technology

Analog plant with digital controller in Qsyn

```
function [G] = d4(s)
% d4.m Controller function description of anti-aliasing filter, in series with the
% sample and hold transfer function, without computational delay,  and  the digital
% feedback compensator for the SISO plant ex2_1a.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
A = [1]./(1 + 2*0.7*s/100 + s.*s/10000);      % anti-aliasing
Ts = 0.020;                                    % seconds, sampling period
H = (1 - exp(-Ts*s))./(Ts*s);                  % Sample and Hold
z = exp(Ts*s);
C = 0.09* ((z - exp(-5*Ts)).*(z.*z - 2*exp(-2.4*Ts)*cos(3.2*Ts)*z + …
        exp(-4.8*Ts)))./((z - 1).*(z - exp(-3*Ts)).*(z - exp(-Ts)));
G = A.*H.*C;
```

... this form is also the result after matched translation

TECHNION
Israel Institute of Technology

Analog plant with digital controller, cont'd

Example:

$$P(s)=k\cdot\frac{s+a}{1+2\varsigma s/\omega_n+s^2/\omega_n^2},$$

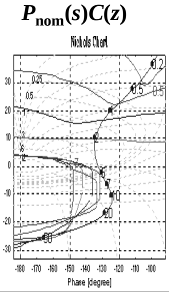
$k\in[2,5], a\in[1,3], \varsigma\in[0.1,0.6], \omega_n\in[4,8]$

Nom: $k=2,\ a=3,\ \varsigma=0.6,\ \omega_n=4$

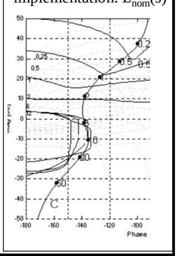
$$\begin{cases} M \le 10\% \\ t_s(5\%) \le 1.5 \text{ seconds} \end{cases}$$

$|S|<6$dB

$P_{\text{nom}}(s)C(z)$



Cf. first analog SISO design with analog implementation: $L_{\text{nom}}(s)$



TECHNION
Israel Institute of Technology

Analog plant with digital controller, cont’d

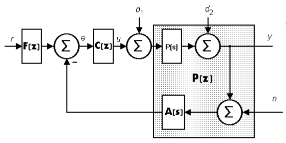
- **If more convenient for designer, parameterize the controller in the *s*-domain:**

```
function [G] = d5(s)
  A = [1]./(1 + 2*0.7*s/100 + s.*s/10000);     % anti-aliasing
  Ts = 0.020;                                   % seconds, sampling period
  H = (1 - exp(-Ts*s))./(Ts*s);                 % Sample and hold
  Gnum = [1/16  2*0.6/4  1]; Gden = [1/3 1 0];
  [Cnum,Cden] = c2d(Gnum,Gden,Ts,’matched’);
  z = exp(Ts*s);
  C = polyval(Cnum,z)./polyval(Cden,z);
G = A.*H.*C;
```


TECHNION
Israel Institute of Technology

Design alternatives:

3. Sampled plant model with digital controller



- **The digital plant model is obtained by Zero Order Hold translation**

$$A(s)P(s) \to P(z)$$

- **ZOH is OK, since plant input *u* is piecewise constant over T_s**
- **Controller design, $C(z)$, as above.**

TECHNION
Israel Institute of Technology

Sampled plant model in Qsyn

$$P(s) = k \cdot \frac{s+a}{1+2\varsigma s/\omega_n + s^2/\omega_n^2},\quad k \in [2,5],\ a \in [1,3],\ \varsigma \in [0.1, 0.6],\ \omega_n \in [4,8] \qquad A(s) = \frac{1}{1+2\cdot 0.7 s/100 + s^2/10000}$$

```
function   [Par,w_tpl,w_nom,method,P_num,P_den, n_dif,Uns_Par] = ex2_1a
%  Plant_name :  Example 2.1. Sampled plant model with anti-aliasing filter
% Definition of the parameters
Par = ['k=[2, 5, 2, 8]' , ...                       % uncertain gain
        'a=[1, 3, 3, 8]', ...                       % zero: s+a
        'zet=[0.3, 0.6, 0.6, 8]', ...               % complex pole
        'wn=[4, 8, 4, 8]' ]; % Definition of the frequency vectors [rad/sec]
         w_tpl = [0.2 0.5 1 2 5 10 20 50]; w_nom = logspace(-1,2);      %  frequencies [rad/s]
         method = ‘adgrid_[1,1]';                   % template computation method
% Plant definition: Functional Structure
        P_num='pfun(k,a,zet,wn,s,Ts)';              P_den='1';
```


TECHNION
Israel Institute of Technology

Sampled plant model in Qsyn, cont’d

```
Pz = pfun(k,a,z,wn,s)  %file pfun.m
Ts = 0.020;
z = exp(Ts*s);
num = k*[1 a];     den = conv([1/(wn*wn) 2*zet/wn 1],[1/10000  2*0.7/100   1]);
[dnum, dden] = c2dm(num,den,Ts,’zoh’);
Pz = polyval(dnum,z)./polyval(dden,z);
```

- Note that this use of plant.m and pfun.m enables arbitrary plant definitions.