Petru Chioncel, Elisabeta Spunei, Cristian Paul Chioncel

Calculation of Control Circuits in Frequency Domain using Scilab Environment

The paper presents the computing of control circuits in the frequency domain, starting from the mathematical model of the frequency response H(jŠ) obtained from the transfer function H(s) where the operational variable keeps just the image part. For PT1 and PT2 elements, using Scilab, the geometrical place is illustrated and the frequency diagrams are determinate for different duping constants (PT2). The logarithmic frequency diagrams (Bode), determine the amplitude and phase reserve for a control circuits with three PT1 elements.

Keywords: mathematical model, transfer function, frequency response function, Bode diagrams, Scilab / Xcos environment

1. Introduction

The geometric locus – hodograph – is a graphical representation of the frequencies / pulsation response function $H(j\omega)$. The values of the $H(j\omega)$ function represents the peaks of the vector in the domain $0 < \omega <$, in a complex plane. With help of the Niquist criteria, the stability of the closed loop system can be verified starting from the open loop circuit system.

Using the Scilab program for determining the geometrical place and the Bode diagrams, the polynomial form of the frequency response function is used. The Bode diagram represents the amplitude, $lgH(j\omega)$ and phase, in separate diagrams, in function of $lg(\omega)$. The control circuit has to be tuned in such way, so that the stability limit should not be achieved. It is necessary that referring to the stability limit, a big enough reserve has to exist, so that in case of parameter modification of the control circuit, instability is not allowed to appear. This is stated through the amplitude margin A and phase reserve $\{_r$.

2. Mathematical model, geometric locus and Bode diagrams computed with Scilab

The mathematical model of the response frequency function is [5]:

$$H(j\omega) = \frac{y(j\omega)}{u(j\omega)} = \left|\frac{y}{u}\right| \cdot e^{j\varphi} \,, \tag{1}$$

and the geometrical locus results representing the values of H(jω) for 0 < ω <   . Using Scilab environment, the transfer function will be expressed in polynomial form:

$$H(s) = \frac{b_m \cdot s^m + b_{m-1} \cdot s^{m-1} + ... + b_1 \cdot s + b_0}{a_n \cdot s^n + a_{n-1} \cdot s^{n-1} + ... + a_1 \cdot s + a_0} \tag{2}$$

The real and imaginary part of the frequency response function is computed and plotted in Scilab using the function nyquist (sl, [fmin, fmax][,step]), with the arguments sl – continuous or discrete time SIMO linear dynamical system; fmin, fmax - real scalars (frequency lower and upper bounds in Hz); step – if not given, an adaptive discretization is used. [2], [7]

The Scilab files for representing the geometrical place for a PT1 and PT2 element is based on the transfer functions [1]:

$$H_{PT1}(s) = \frac{K_{P1}}{1 + sT_1} \tag{3}$$

$$H_{PT2}(s) = \frac{K_{P2} \cdot \omega_0^2}{s^2 + 2 \cdot D \cdot \omega_0 s + \omega_0^2} \tag{4}$$

The relations (3) and (4) have following parameters: amplification factors K_{p1}, K_{p2}, own pulsation ω_o and D – amortization factor [4].

Table 1 contains the Scilab script under which the Nyquist diagram, figure 1, is computed for PT1 – and PT2 elements.

Table 1. Scilab Script – file computing Nyquist diagram

```
Kp1=1; T1=4;                    // gain and time constant
Kp2=1; w0=0.5; D=0.25          // Parameter
s=%s;
num1=Kp1;                       // numerator
den1=s*T1+1;                    // denominator
tf1=num1/den1;                  // transfer polynominal

num2=Kp2*(w0^2);                // numerator
den2=(s^2)+2*D*w0*s+w0^2;       // denominator
tf2=num2/den2;                  // transfer polynominal
```

```scilab
// create a Scilab-usable continous system
ansys1=syslin('c',Kp1/(T1*s+1));
ansys2=syslin('c',tf2);
t=linspace(-2,2,50);
nyquist(ansys1);
nyquist(ansys2);
```


Figure 1. Nyquist diagram for PT1 and PT2 element

The next analyze, table 2, represents a PT2 element (4) for different amortizations, D, in the domain 0.1 ≤ D ≤ 2, figure 2.

Table 2. Scilab Script –Bode answer for PT2 element at different D

```scilab
//Bode - diagramm for PT2 element at different amortizations
s=%s;
D=[0.1, 0.2, 0.3, 0.5, 0.707, 1, 2];          // Amortization values
for k=1:7
  num=1;                                       // numerator
  den=s^2+2*D(k)*s+1;                          // denominator
```

```
    tf(k)=syslin('c',num/den);
end

clf();
bode([tf(1);tf(2);tf(3);tf(4); tf(5); tf(6); tf(7)], 0.1, 10, ...
['D=0.1'; 'D=0.2'; 'D=0.3'; 'D=0.5'; '0.707'; 'D=1'; 'D=2'])
```
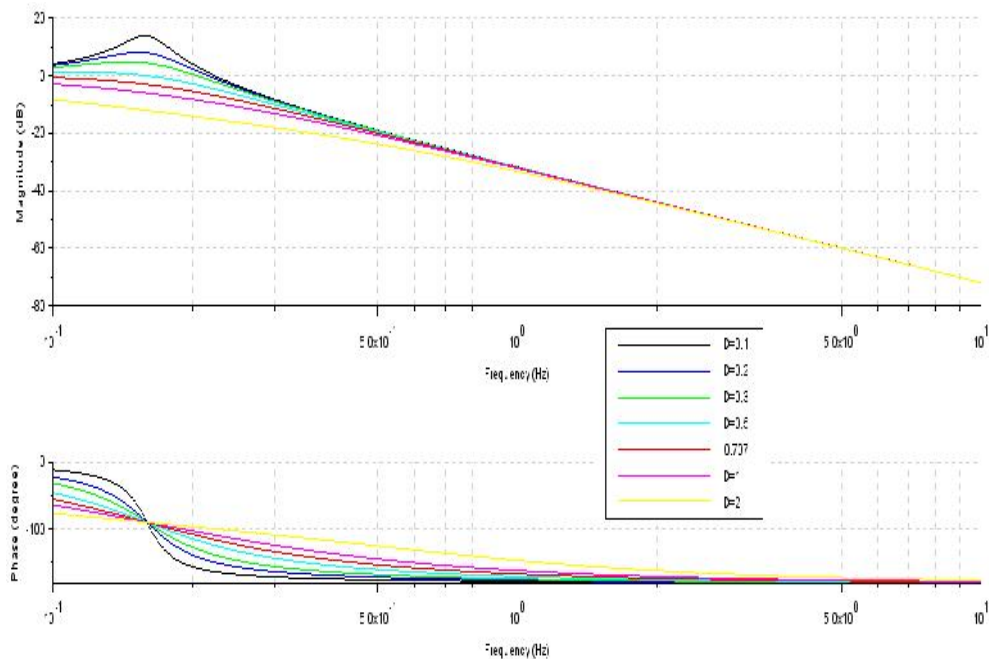

Figure 2. Bode diagrams for a PT2 element, at different amortizations

The Bode diagrams are exemplified with an open loop control circuit with 3 PT$_1$ elements, where H$_C$ – represents the controller transfer function and HP – the process transfer function, having the transfer function [5]:

$$H_{CP}(s) = H_C(s) \cdot H_P(s) = \frac{K_C \cdot K_P}{(1 + s \cdot T_1) \cdot (1 + s \cdot T_2)(1 + s \cdot T_3)} \quad (5)$$

The simulation of the bode diagrams, figure 3, computing and displaying the gain and phase margin, together with the associated crossover frequencies of linear system is computed in Scilab, table 3, through the 'show_margins' function, represented by the calling sequence: show_margins (h,'bode'), were h – is a SISO linear system in continuous time [3], [7].

306

```
Ks=2; T1=1; T2=0.1; T3=T2; KR=5;        // Parameter
s=%s;
num=Ks*KR;                              // numerator
den=(1+s*T1)*(1+s*T2)+(1+s*T3);        // denominator
tf=num/den;                             // transfer polynominal

// create a Scilab-usable continous system
ansys=syslin('c',tf);
show_margins (ansys);
```
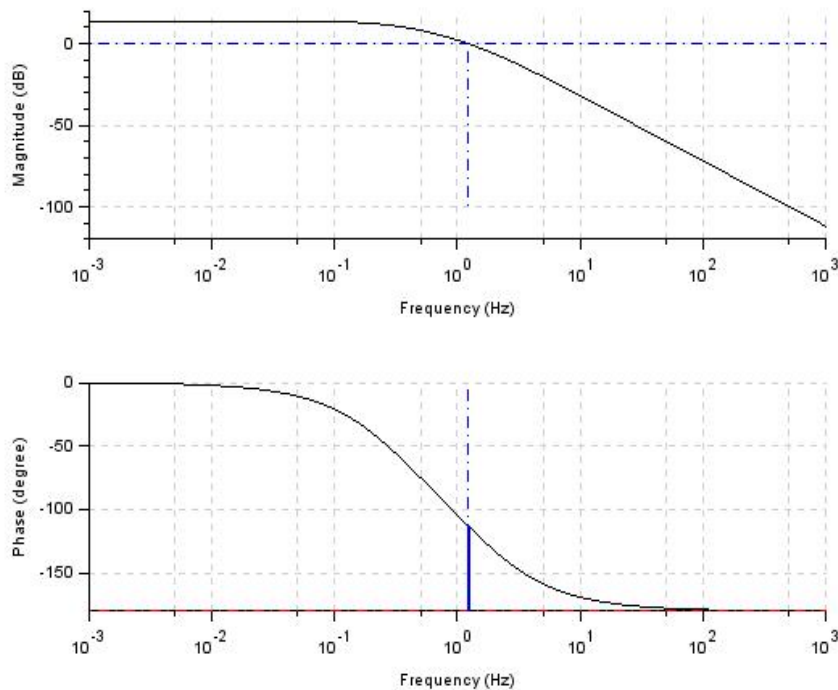

Figure 3. Bode diagrams of an open loop with three PT1 elements, with the marked gain and phase reserve

4. Conclusion

With those presented, the control circuits in frequency / pulsation domain can be computed, using the availabilities offered by Scilab. Specific Scilab functions are defined to determine the geometrical locus and Bode frequency diagrams.

With their help the amplitude and phase reserve can be determinate and, through this, the stability of the controlled systems [6].

The Scilab / Xcos modeling and simulation programming language offers an effective practice instrument to compute the control circuits in frequency domain.

References

[1]   Anghel C.V., Visan A.D., Practice used of DOMS PSS 500 Controller, X[th] International Symposium "Tinerii i cercetarea multidisciplinar " Timi oara, 2008.

[2]   Chioncel C.P., Limbaje de simulare. Aplicatii de modelare si simulare. Editura Eftimie Murgu, 2004

[3]   Gilberto E.U., Time series and spatial data analysis with Scilab, infoc-learinghouse.com, 2001.

[4]   Giulio A., SPICE equivalent circuits of frequency-domain response, IEEE Transactions on electromagnetic compatibility, vol.45, no.3, 2003, pp.502–512

[5]   Lutz W., Wendt W., Taschenbuch der Regelungstechnik. Verlag Harri Deutsch, erweiterte Auflage 6., 2005.

[6]   Ungureanu-Anghel D., Belci D.I., Raduca E., Raduca M., Minimum vicinity interpolation algorithms for calculation of trajectory processing for a model of CNC machine tool, Computational Cybernetics and Technical Informatics ICCC, 2010, pp. 367–372.

[7]   www.scilab.org

Addresses:

- Prof. Dr. Eng. Petru Chioncel, "Eftimie Murgu" University of Re i a, Pia a Traian Vuia, nr. 1-4, 320085, Re i a, p.chioncel@uem.ro
- Lect. Dr. Eng. Elisabeta Spunei, "Eftimie Murgu" University of Re i a, Pia a Traian Vuia, nr. 1-4, 320085, Re i a, e.spunei@uem.ro
- Assoc.Prof. Dr. Eng. Cristian P. Chioncel, "Eftimie Murgu" University of Re i a, Pia a Traian Vuia, nr. 1-4, 320085, Re i a, c.chioncel@uem.ro