Negative feedback control system analysis

1 Aims

Use of Matlab/Simulink to simulate the behaviour of dead time systems (systems with transport delay).

Stability analysis of negative feedback control systems.

2 Theoretical considerations

Gain margin and phase margin for stability analysis of NFCS (negative feedback control systems).

Replacement of the reference unity step signal to PWM signal (pulse width modulated signals).

2.1 Exemple of simulation NFCS in Matlab/Simulink

Considerring the simple NFCS in Figura 2.1 havin on the direct way a nonminumum phase term (dead time/ transport delay, denoted by $\tau_m = 0.2\ sec.$, and a proportional term standing for the „proportional effect”$k = 2$).

The blocks in Figura 2.1 can be found as follows:

- Simulink->Sources->*step* *(set the step time to 0 not to 1)*
- Simulink->Math operations->*gain/sum*
- Simulink->Contnuous->*zero-pole/transport delay*
- Simulink->Sinks->*scope*

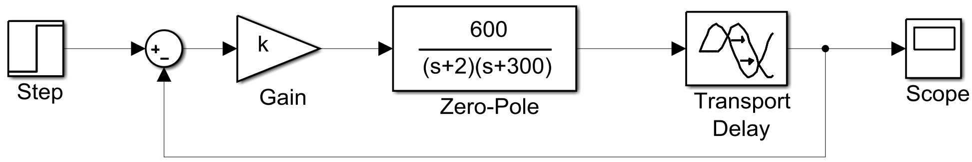


Figura 2.1 Introducerea elementului cu timp mort în Simulink

Make the necessary conections and save the simulink file as „lab7”.

For the case of constant gain (proportional gain/effect), the values of the dead time that assures stable closed loop can be found by applying „simplified Nyquist criterion”:

$$\gamma_k > 0 \Longrightarrow \ \tau_m < \frac{\pi - arctg\left(\frac{\omega_c}{2}\right) - artcg(\frac{\omega_c}{300})}{\omega_c} \Longrightarrow \tau_m < 0.6005$$

To simulate the step response depending on the dead time, three values are of interest:

- one smaller than 0.6005 resulting the underdamped behavior (asymptotically stable);
- one equal to 0.6005 resulting the undamped behavior (stability limit);
- one higher than 0.6005 resulting the unstable case (that must be avoided!)

If replace in the Simulink structure the parameter of the “transport delay” by the variable “tm”, the next script will generate the plots in figura 2.2 (the left side)

```
tmv=[0.4 0.6005 0.9]; k=2;num=600;den=conv([1 2],[1 300]);
disp={'stabil','limita de stabilitate','instabil'};
for i=1:3
    tm=tmv(i);
    sim('lab7');
    t=data.time;
    y=data.signals.values(:,1);
    subplot(3,1,i);plot(t,y);grid
    title(['Răspuns la semnal treaptă, [',disp{i},',\tau_m=
',num2str(tmv(i)),']']);
    clear y t;
end
```

In Figura 2.22 the three case can be evaluated: stable, stability ;limit and unstable case. It can be noticed that ω_c remains unchanged as the dead time varies while $\omega_{-\pi}$ decreases as the dead time increases; in conclusion, for this case, to obtain stable closed loop, it is sufficient to maintain the next relation: $\omega_c > \omega_{-\pi}$

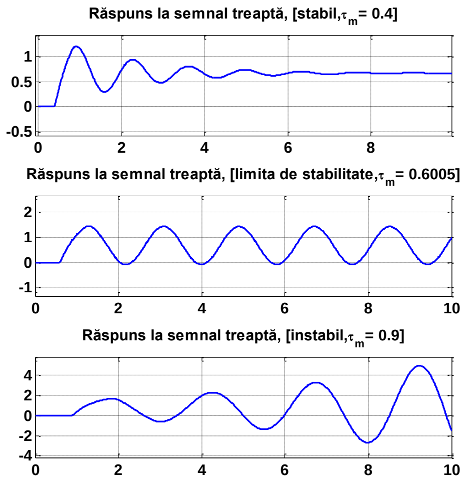


Figura 2.2 Simularea comportamentului la semnal treaptă

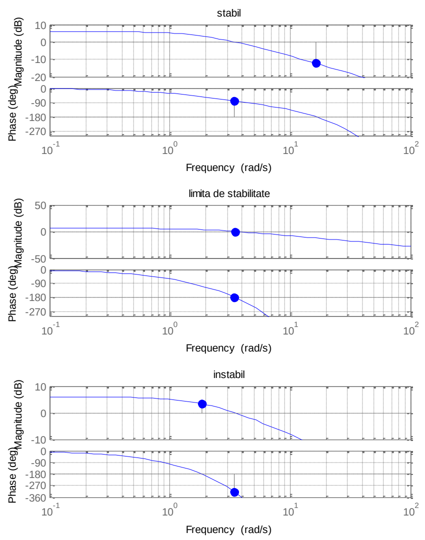


Figura 2.3 Răspunsul în frecvenţă

2.2 Requests

- Mentions in your report the negative feedback control structure using complex exponential for time delay term and block diagram representation similar to the one introduced at the course;
- Explain in more mathematical details the way of computing the domain for the dead time that assures stable closed loop;
- Repeat the example for a proportional gain k=3;
- For constant dead time ($\tau_m = 0.15$) design a proportional controller in order to obtain stable closed loop; use phase and gain margin in order to design/compute the proportional gain.

- In the suggested script, change the titles with the suitable terms in English;
- Modify the suggested script in order to obtain the frequency response analysis in Figura 2.3
- Modify the suggested script in order to verify if you computed correctly the proportional controller;

3 Use of PWM signals as reference signals

The migrations to microcontrollers instead of analog controllers (based on OpAmps) suppose among other very important aspects, the use of PWM signals. For this lab, the main concern is on how to choose the minimum frequency for the PWM signal.

The next simulink structure is proposed in order to compare the "analog case" with the unity step signal and the "discrete-time case" with the PWM signal as the reference for the negative structure.

To faster obtain the simulink file in figura 2.4, copy the entire structure and paste it below on the model.

The simulations are made for a stable case with $\tau_m = 0.2\,sec.$ and $k = 2$, for both structures.

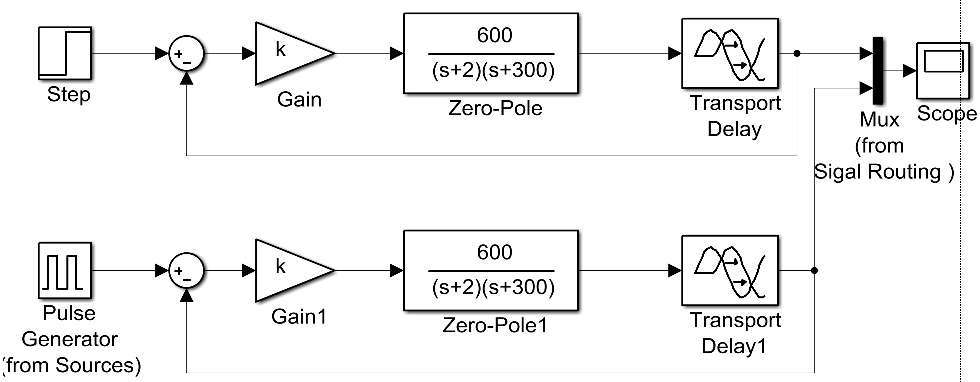


Figura 3.1 Semnal de intrare de tip PWM

The frequency (rad/s) of the PWM signal can be established from the open loop Bode diagram. If evaluating the magnitude for an attenuation of 40 dB, it will result the minimum frequency for the PWM signal. Applying this frequency will attenuate / cut all the higher frequencies in the input signal, remaining the effect of the continuous component.

3.1 Requests

- Draw in Matlab the bode diagram for the open loop with $\tau_m = 0.2\,sec.$ and $k = 2$;
- Read form Bode diagrams, the frequency at which the magnitude is -40dB and mention it on your report;
- Set in the 'Pulse generator' block in your structure, the frequency you have red it on Bode; notice that the parameter refers the period of the generator not the frequency (rad/s); set the pulse width to 99%;

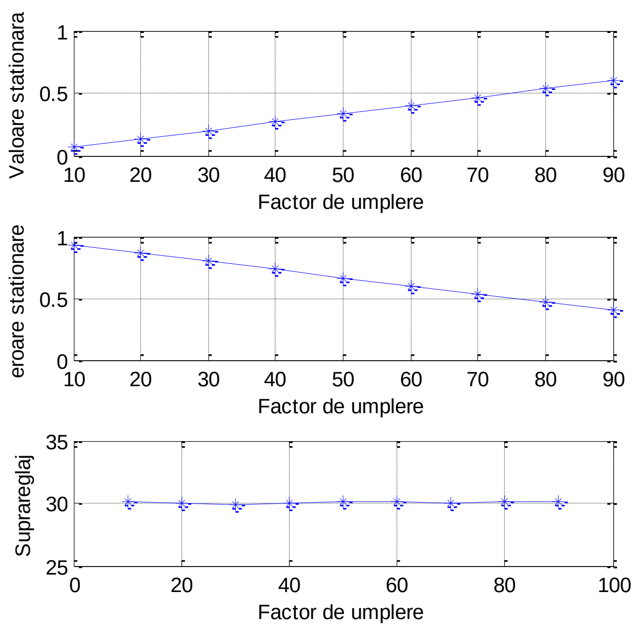
- Run the model in Simulink and compare the two outpus;
- Change the period of the pulse generator ten times higher than the previous case and notice the steady state responses.
- Establish the minimum frequency (Hz) your microcontroller should work to generate such a PWM signal

4 Problems

1. For the structure in Figura 3.1 with the PWM signal as reference, fill in the next table, according to the example:

Duty factor (Pulse Width)	y_{ss} (steady stae)	e_{ss}^{p} (position steady state error)	$\sigma(\%)$ (overshoot)	t_s (settling time)	t_r (rise time)
10%					
20%					
…					
50%	0.3335	0.6665	30.0646	0.4053	0.3833
…					
80%					
90%					

2. Draw in your report the variation of the steady state to the duty factor;
3. Draw in your report the variation of the steady state to the overshoot;
4. Make a Matlab script in order to automatically generate the table above and obtain the next plot:



5 Variantă de rezolvare

Se realizează următoarele setări (T ca perioadă a semnalului şi fu ca factor de umplere în procente, (Figura 5.1) în blocul „Pulse Generator”; pentru preluarea datelor în „workspace” se realizează aceleaşi modificări precizate în paragraful 2.1 (Figura 5.2).

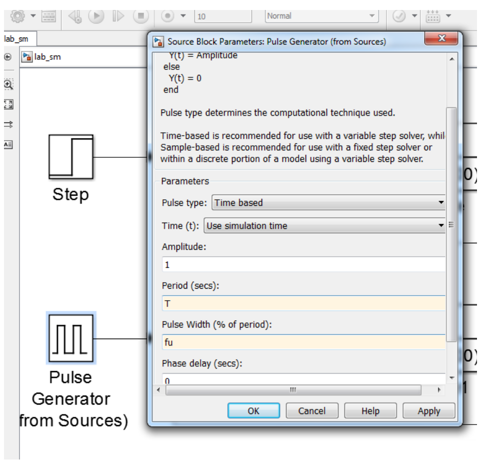


Figura 5.1 Parametrii blocului „Pulse Generator”

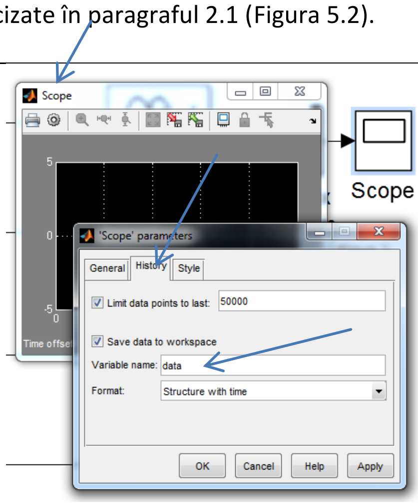


Figura 5.2 Parametrii blocului Scope

```
Clear
tm=0.2;k=2;T=2*pi/1e+3;
fuv=10:10:90;tabel=[];
for i=1:length(fuv)
     fu=fuv(i);
     sim('lab7');
     t=data.time;
     y=data.signals.values(:,2);
     yss=y(end);% valoarea stationara
     essp=1-yss; % eroarea stationara la pozitie
     sigma=(max(y)-yss)/yss*100; % suprareglaj
     ts=t(find((y>yss),1)); % timp de stabilizare
     tr=t(find((y>yss*0.9),1));%timp de ridicare
     tabel=[tabel; fu, yss, essp, sigma,ts, tr];
     clear y t;
end
```