Informatyczne Systemy Sterowania

Ćwiczenie 4: Sterowanie szybkością transmisji w sieci komputerowej

Krzysztof Przybylski 239266

19 stycznia 2019

1 Wstęp

1.1 Cel ćwiczenia

Wykonanie symulacji systemu sterowania ruchem w sieci komputerowej celem:

1. Poznania problemu sterowania ruchem w sieci komputerowej jako przykładu sterowania systemem komputerowym.
2. Nabycia umiejętności wykorzystania pakietu Matlab oraz Simulink do symulacji ww. systemu.

1.2 Plan działań

1. Symulacja systemu sterowania szybkością transmisji.
   W trakcie realizacji zadania należy zamodelować obiekt sterowania (mechanizmy sieci komputerowej uwzględniane w problemie sterowania ruchem w sieci komputerowej).
2. Dobór optymalnego regulatora.
   Realizacja zadania polega na
   (a) Doborze odpowiedniej struktury urządzenia sterującego z grupy regulatorów PID (P, PI, PD, PID).
   (b) Doborze optymalnych wartości parametrów regulatora.
   (c) Przeprowadzeniu badań pozwalających na ocenę jakości działania rozpatrywanego systemu.
3. Zastosowanie członów korekcyjnych.
   Realizacja zadania polega na:
   (a) Zaproponowaniu zastosowania odpowiednich członów korekcyjnych, pozwalających na polepszenie jakości działania systemu.
   (b) Przeprowadzeniu symulacji działania systemu z takimi członami.
   (c) Dobraniu optymalnych wartości parametrów członu korekcyjnego.

2 Realizacja planu i wyniki

2.1 Symulacja systemu sterowania szybkością transmisji

Systemem symulowanym w tym ćwiczeniu będzie system opisany w artykule "*Complete Stability Region Characterization for PI-AQM*" podanym w literaturze. Jest to sieć, w której:
C - przepustowość sieci,
N - ilość otwartych sieci TCP,
d - opóźnienie pakietów, uzyskaliśmy następującą transmitancję.

$$P(s) = \frac{B}{(s+\alpha)(s+\beta)} e^{-sd} \tag{1}$$

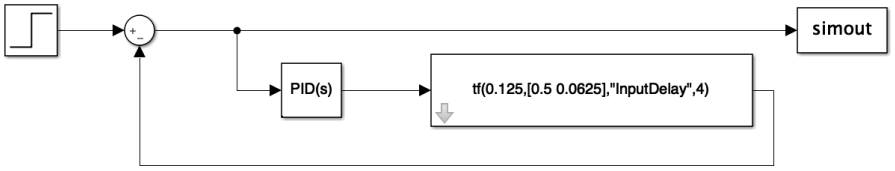
Gdzie $\alpha = \frac{2N}{d^2 C}$, $\beta = \frac{1}{d}$ i $B = \frac{C^2}{2N}$.

Do regulowania systemu będę używał regulatorów z rodziny PID.

Żeby zasymulować system w Simulinku pomnożyłem mianownik transmitancji obiektu tak, aby można było wykorzystać go w bloku LTI system.

$$P(s) = \frac{B}{s^2 + (\alpha+\beta)s + \alpha\beta} e^{-sd} \tag{2}$$

Gotowy schemat służący nam do symulacji przedstawiony jest na poniższym obrazku.



Rys. 1: Schemat systemu służący do symulacji.

2.2 Dobór optymalnego regulatora

Przy doborze optymalnego regulatora i jego parametrów używałem poniższej funkcji napisanej w Matlabie.

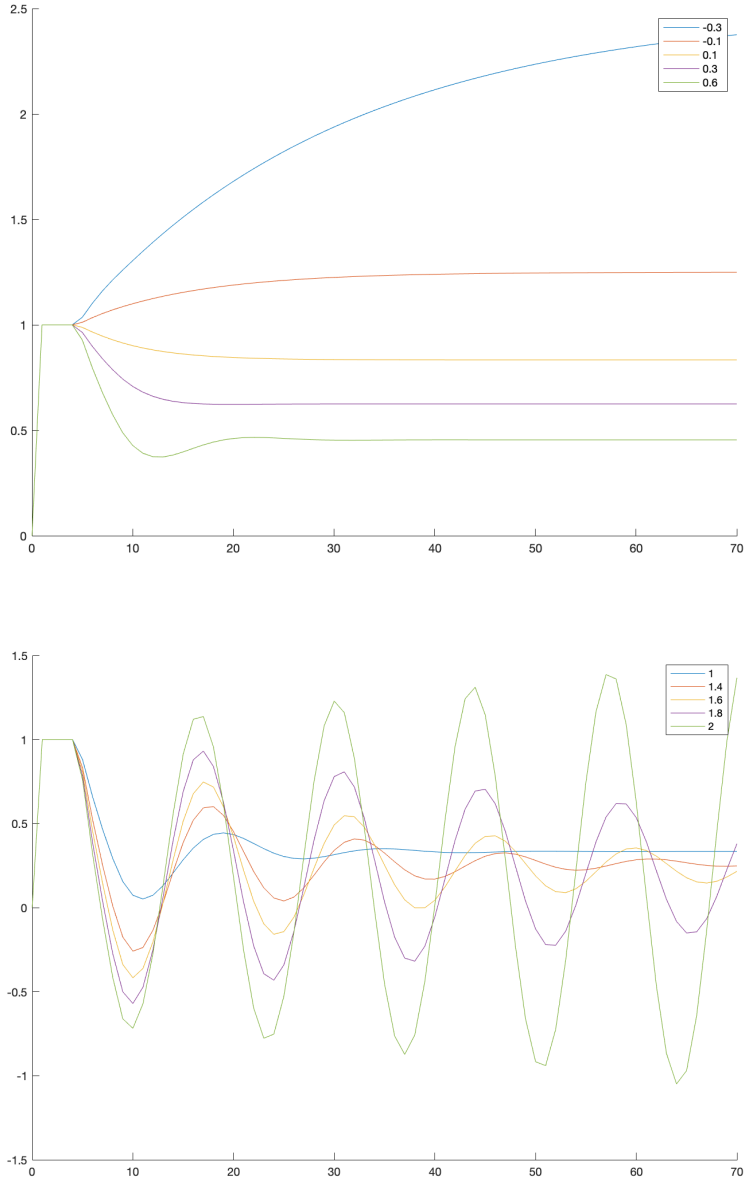
```
function testLTI(P, I, D, N, C, d)
hold on;
load_system('LTI.slx');
set_param('LTI/PID Controller1', 'P', num2str(P));
set_param('LTI/PID Controller1', 'I', num2str(I));
set_param('LTI/PID Controller1', 'D', num2str(D));

a=2*N/((d^2)*C);
b=1/d;
B=(C*C)/(2*N*N);

num =  B;

set_param('LTI/LTI System','SYS', ['tf(',num2str(num),',[',num2str(a+b),' ',num2str(a*b),'],"InputDelay",',num2str(d),')']);

sim('LTI.slx');
figure(1);
plot(simout.time, simout.signals.values, 'DisplayName', num2str(D));

end
```

Fun. 1: Funkcja testująca system.

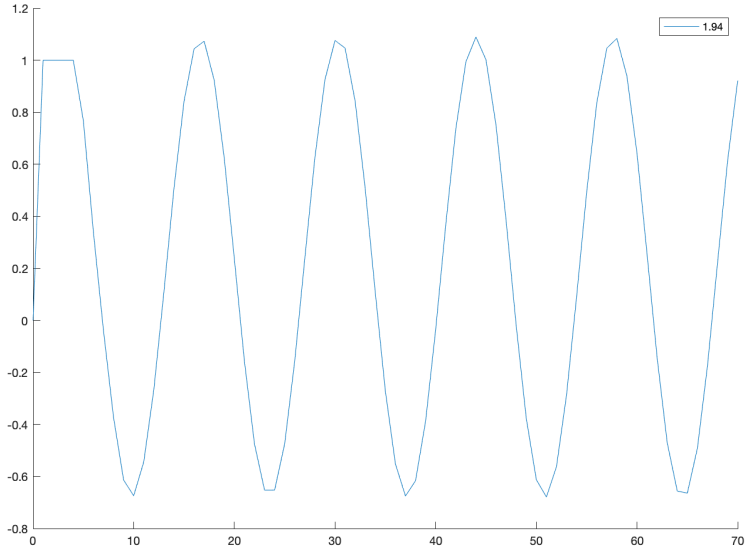
Do testów wybrałem następujące parametry $N = 6$, $C = 3$, $d = 4$

2.2.1 Regulator P

Symulując system z regulatorem P wywoływałem powyższą funkcję, podając wartość 0 dla parametrów I, oraz D. W ten sposób uzyskałem poniższe wykresy.



Rys. 2: Wykresy przebiegu błędu regulacji przy zmianie parametru P



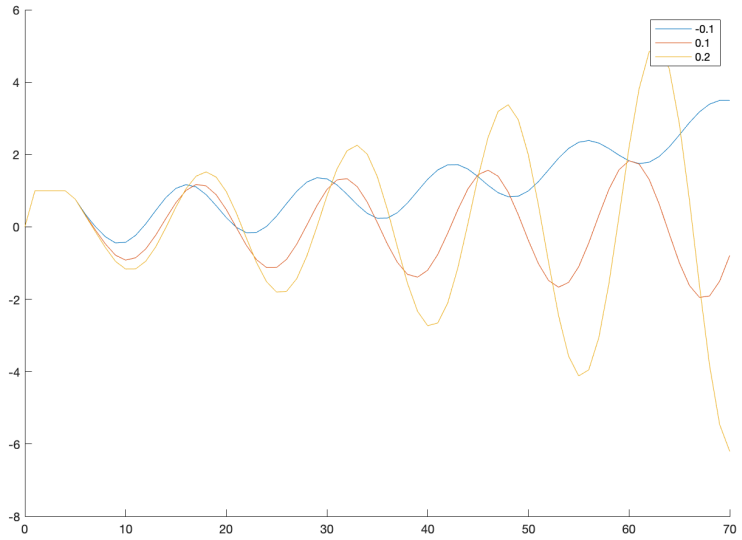
Rys. 3: Wykresy przebiegu błędu regulacji dla parametru $P = 1.94$

Na wykresach możemy zaobserwować, że:

- Dla ujemnych wartości parametru P wartość błędu szybko stabilizuje się na pewnym poziomie. Poziom ten jest tym większy, im mniejsza jest wartość parametru P.
- Dla małych (< 1.4) dodatnich wartości parametru P wartość błędu również szybko stabilizuje się na pewnym poziomie, a poziom ten jest tym mniejszy, im większa jest wartość parametru P.
- Dla większych wartości dodatnich (> 1.4) wykres wartości błędu regulacji wolniej stabilizuje się, a dla $P \approx 1.94$ przybiera kształt drgań o stałej amplitudzie. Powyżej tej wartości amplituda drgań z czasem rośnie.

2.2.2 Regulator PI

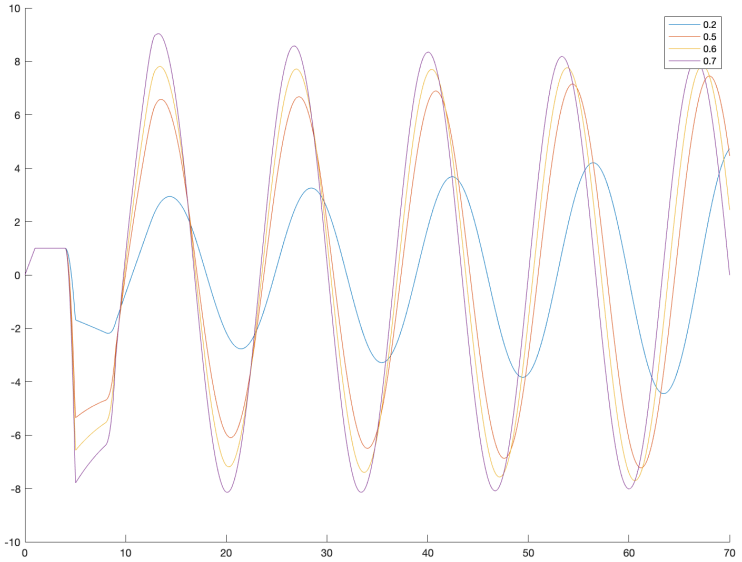
Dla regulatora PI wywołałem funkcję z parametrami $P = 1.94$ oraz $D = 0$. Podobnie jak w przypadku regulatora P, tutaj za wartość optymalną uznałem 0.1, dla niej wykres ma prawie stałą amlitudę błędu.



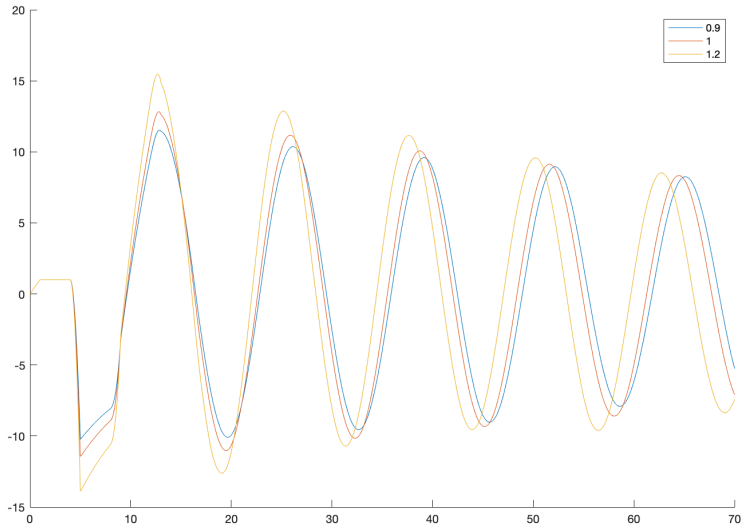
Rys. 4: Wykresy przebiegu błędu regulacji przy zmianie parametru I

2.2.3 Regulator PID

Dla regulatora PID wywołałem funkcję z parametrami $P = 1.94$ oraz $I = 0.1$.



Rys. 5: Wykresy przebiegu błędu regulacji przy zmianie parametru D



Rys. 6: Wykresy przebiegu błędu regulacji przy zmianie parametru D

Dla wartości poniżej 0.6 amplituda jest tym większa im mniejsza wartość, dla wartości równej 0.6 błąd ma stałą amplitudę, natomiast powyżej 0.6 amplituda maleje.

2.2.4 Metoda Zieglera-Nicholsa

Zadanie polega na doborze parametrów regulatora PID według zasad Zieglera– Nicholsa

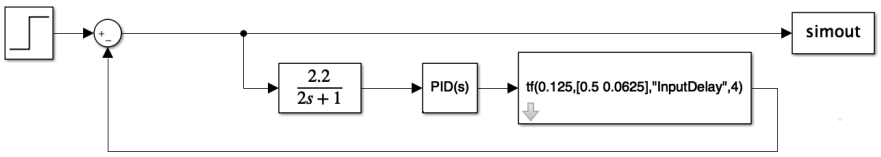
Doświadczalnie znaleźć współczynnik wzmocnienia, dla którego układ traci stabilność.
$p = 1.94$

B. Ustalić okres oscylacji i wzmocnienie krytyczne.
$K_u = 13$
$K_p = 1.07$

C. Według odpowiednich rekomendacji określić wartości parametrów regulatora PID.
$P = K_p * 0.6 = 0.642$
$I = K_u * 0.5 = 6.5$
$D = K_u * 0.125 = 1.625$

2.3 Zastosowanie członu korekcyjnego

2.3.1 Symulacja systemu sterowania szybkością transmisji z członem korekcyjnym



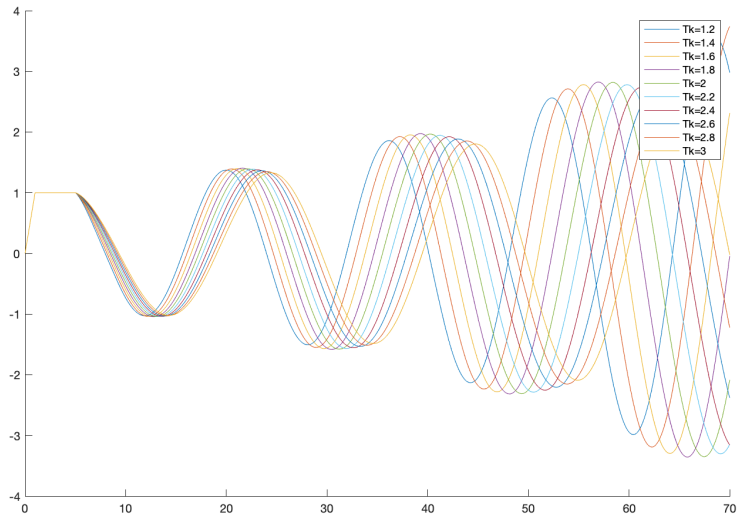
Rys. 7: Schemat systemu z członem korygującym

2.3.2 Dobór optymalnych parametrów

```
function testLTIKor(P, I, D, N, C, d, start, step, stop)
load_system('LTI_kor.slx');
hold on;
set_param('LTI_kor/PID Controller1', 'P', num2str(P));
set_param('LTI_kor/PID Controller1', 'I', num2str(I));
set_param('LTI_kor/PID Controller1', 'D', num2str(D));

a=2*N/((d^2)*C);
b=1/d;
B=(C*C)/(2*N*N);
num =  B;
set_param('LTI_kor/LTI System','SYS', ['tf(',num2str(num),',[',num2str(a+b),' ',num2str(a*b),'],"InputDelay",',num2str(d),')']);



s=start;
while(s <= stop)
%set_param('LTI_kor/Transfer Fcn', 'Denominator', strcat('[',num2str(s), ' 1]'));
set_param('LTI_kor/Transfer Fcn', 'Denominator', strcat('[3 1]'));
set_param('LTI_kor/Transfer Fcn', 'Numerator', strcat('[',num2str(s),']'));

sim('LTI_kor.slx');
s= s+step;
figure(1);
plot(simout.time, simout.signals.values, 'DisplayName', strcat('Tk=',num2str(s)));



end
hold all;
end
```

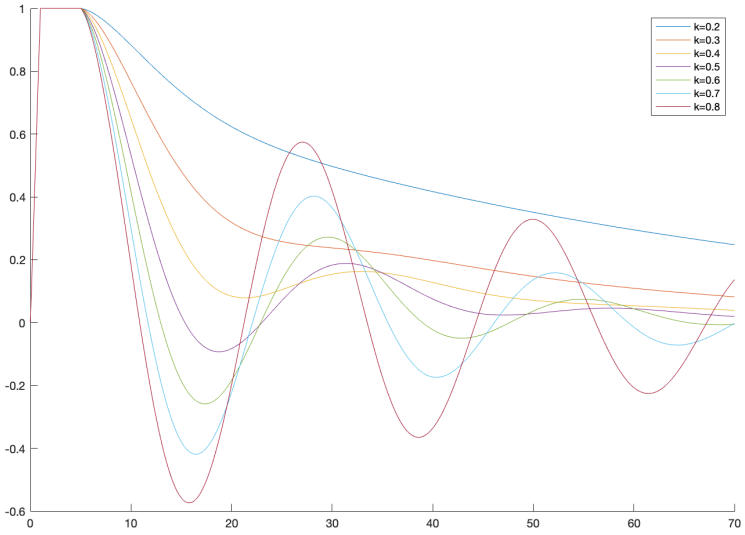
Fun. 2: Funckja do symulacji systemu z blokiem korekcyjnym

Ustawienia parametrów: $P = 1.94$, $I = 0.1$, $D = 0.6$, $N = 6$, $C = 3$, $d = 4$



Rys. 8: Wykres błędu dla różnych wartości Tk

Za optymalną wartość uznałem $Tk = 3$, gdyż wtedy amplituda jest najmniejsza



Rys. 9: Wykres błędu dla różnych wartości k, przy $Tk = 3$

Za optymalną wartość uznałem $k = 0.5$, gdyż wtedy amplituda jest najmniejsza

3 Wnioski

Z przeprowadzonego ćwiczenia wynika, że największe możliwości sterowania daje regulator PID, najbardziej można wówczas dostosować regulację do naszej funkcji. Człon korekcyjny redukuje wartość błędu. Parametry regulatora można dobrać doświaczalnie - poobnie jak w przypadku regulatorów dwu- i trójpołożeniowych.