Układ regulacji 2 położeniowej – część 1

Cel ćwiczenia

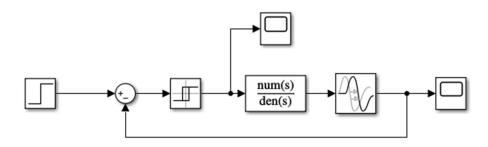
Celem ćwiczenia jest zapoznanie się z działaniem i właściwościami układu regulacji II położeniowej (przekaźnikowej), który jest najprostszym układem regulacji stosowanym w praktyce. Dodatkowo podczas wykonania ćwiczenia należy się zapoznać z środowiskiem SIMULINK, będącym częścią MATLABA pozwalającą na modelowanie systemów dynamicznych z użyciem schematów blokowych. Przed przystąpieniem do realizacji ćwiczenia należy uważnie wysłuchać wstępu teoretycznego, żeby wykonanie ćwiczenia było zrozumiałe.

Rozważany układ regulacji

Jako układ regulacji rozważmy <u>zamknięty układ regulacji</u> składający się z obiektu regulacji opisanego transmitancjami z opóźnieniem o postaci:

Obiekt statyczny:
$$G_1(s) = \frac{e^{-\tau_1 s}}{T_1 s + 1}$$
 (1)

Obiekt astatyczny:
$$G_2(s) = \frac{e^{-\tau_2 s}}{T_2 s}$$
 (2)

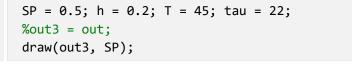


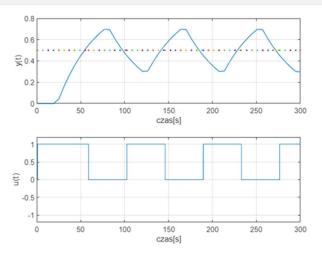
Rys. 1. Układ zbudowany w Simulinku

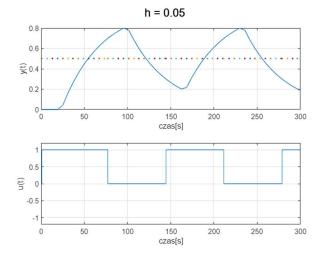
Eksperyment 1: Wpływ szerokości histerezy regulatora h

h = 0.05 h = 0.2

```
SP = 0.5; h = 0.05; T = 45; tau = 22;
%out1 = out;
draw(out1, SP);
```

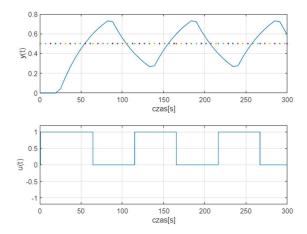






h = 0.1

```
SP = 0.5; h = 0.1; T = 45; tau = 22;
%out2 = out;
draw(out2, SP);
```



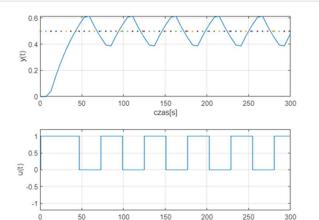
Wraz ze wzrostem wartości h obserwujemy "rozciąganie się" obu funkcji - wzrost ich okresu (spadek częstotliwości). Im większa histereza, tym rzadsze przełączenia regulatora i tym większy maksymalny uchyb (amplituda).

Eksperyment 2: Wpływ wartości opóźnienia obiektu tau

tau = 10

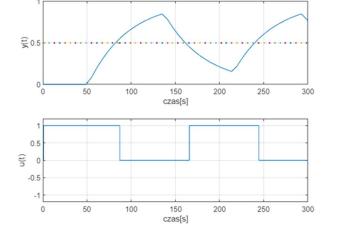
draw(out21, SP);

```
SP = 0.5; h = 0.05; T = 45; tau = 10;
%out21 = out;
```



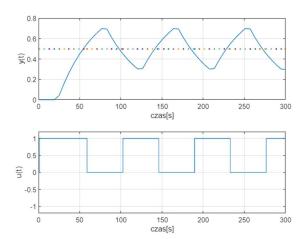
tau = 35

```
SP = 0.5; h = 0.05; T = 45; tau = 35;
%out23 = out;
draw(out23, SP);
```



tau = 22

```
SP = 0.5; h = 0.05; T = 45; tau = 22;
%out22 = out;
draw(out22, SP);
```



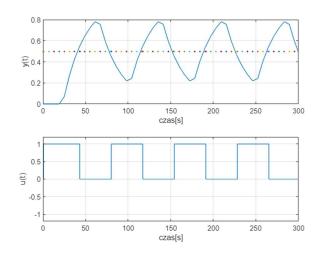
Zwiększanie opóźnienia obiektu, również powoduje zwiększenie uchybu maksymalnego (amplitudy) i spadek częstotliwości

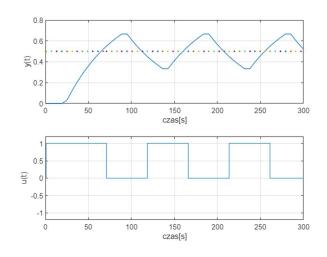
Eksperyment 3: Wpływ wartości stałej czasowej obiektu T

T = 25 T = 60

```
SP = 0.5; h = 0.05; T = 25; tau = 22;
%out31 = out;
draw(out31, SP);
```

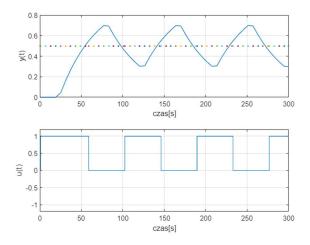
```
SP = 0.5; h = 0.05; T = 60; tau = 22;
%out33 = out;
draw(out33, SP);
```





T = 45

```
SP = 0.5; h = 0.05; T = 45; tau = 22;
%out32 = out;
draw(out32, SP);
```



Zwiększanie stałej czasowej obiektu powoduje zmniejszenie amplitudy y(t) i spadek częstotliwości sygnałów

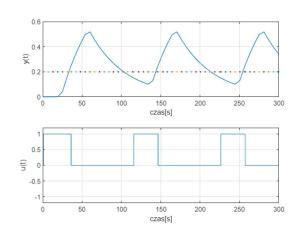
Eksperyment 4: Wpływ poziomu wartości zadanej SP

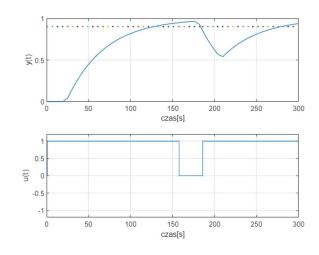
SP = 0.2

```
SP = 0.9
```

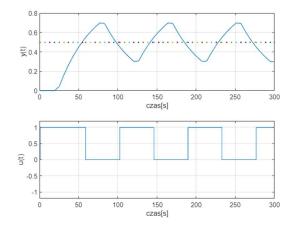
```
SP = 0.2; h = 0.05; T = 45; tau = 22;
%out41 = out;
draw(out41, SP);
```

```
SP = 0.9; h = 0.05; T = 45; tau = 22;
%out43 = out;
draw(out43, SP);
```





SP = 0.5



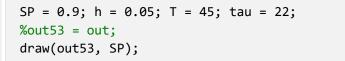
Kiedy wartość zadana jest mała, jest szybko osiągana przez obiekt, i przez większość czasu regulator jest przełączony, tak że zmniejsza wartość wyjścia - przez większość czasu ma ono wartość, która przekracza wartość zadaną. Układ zachowuje się analogicznie przy dużej wartości SP - wartość rzeczywista przez większość czasu jest poniżej wartości zadanej. Zmiana SP zmniejsza częstotliwość sygnałów.

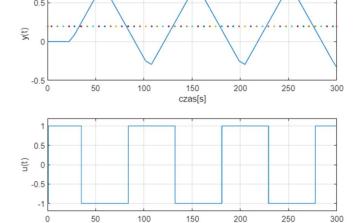
Obiekt astatyczny potrzebował w bloku "Relay" wartości "Output when off" na poziomie -1, ponieważ w przypadku podania wartości 0, sygnał wyjściowy nie dążyłby do zera, a pozostawał na stałym poziomie

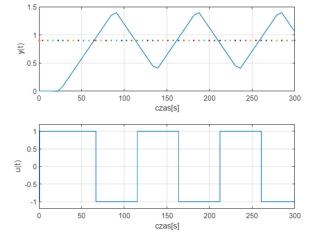
Eksperyment 4: Wpływ poziomu wartości zadanej SP dla obiektu astatycznego

SP = 0.2 SP = 0.9

```
SP = 0.2; h = 0.05; T = 45; tau = 22;
%out51 = out;
draw(out51, SP);
```

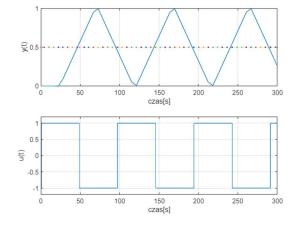






SP = 0.5

czas[s]



Amplituda w przypadku obiektu astatycznego jest większa niż w przypadku statycznego, natomiast różni się kształt sygnału y(t) - w astatycznym rośnie i spada zakresami liniowo. W przypadku małej wartości SP, sygnał wyjściowy osiąga wartości ujemne.

```
function [] = draw(out, SP)
    figure;
    subplot(2,1,1)
    plot(out.y.time,out.y.signals.values,out.y.time,SP,'.')
    grid
    xlabel('czas[s]')
    ylabel('y(t)')
    subplot(2,1,2)
    plot(out.u.time,out.u.signals.values)
    grid
    xlabel('czas[s]')
    ylabel('u(t)')
    axis([0,300,-1.2,1.2])
end
```