

Informatyczne Systemy Sterowania

Ćwiczenie 3: Regulacja dwu- i trójpółosiowa

Krzysztof Przybylski 239266

12 stycznia 2019

1 Wstęp

1.1 Cel ćwiczenia

Celem tego ćwiczenia jest symulacja działania systemu regulacji z przekąźnikami dwu- i trójpółosiowymi. Ćwiczenie ma umożliwić zapoznanie się z nieliniowymi algorytmami sterowania (przełącznikami dwu- i trójpółosiowymi) oraz zapoznanie się ze środowiskiem Simulink oraz Matlab w zakresie nieliniowych algorytmów sterowania.

1.2 Plan badań

1. Symulacja systemu regulacji. Dobór parametrów regulatora.
W trakcie realizacji zadania należy zasymulować działanie systemu regulacji pracującego z regulatorem oraz należy zbadać wpływ wartości parametrów przekąźników na przebieg błędu regulacji
 - (a) Regulator dwupółosiowy bez histerezy.
 - (b) Regulator dwupółosiowy z histerezą.
 - (c) Regulator trójpółosiowy bez histerezy.
 - (d) Regulator trójpółosiowy z histerezą.
2. Zastosowanie członów korekcyjnych.
 - (a) Modyfikacja systemów z zadania 1 o korekcję w postaci członu liniowego o transmitancji:

$$K_k(s) = \frac{k_k}{T_k s + 1} \quad (1)$$

- (b) Doświadczalny dobor parametrów członu korekcyjnego.

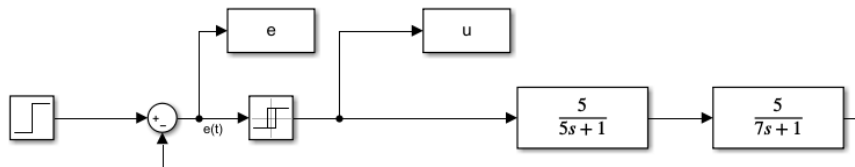
2 Realizacja planu i wyniki

2.1 Symulacja systemu regulacji. Dobór parametrów regulatora.

System regulacji będziemy symulować przy użyciu programu Simulink będącego częścią pakietu MATLAB.

2.1.1 Regulator dwupołożeniowy bez histerezy

Schemat systemu do symulacji regulatora dwupołożeniowego przedstawiony został na poniższym rysunku.



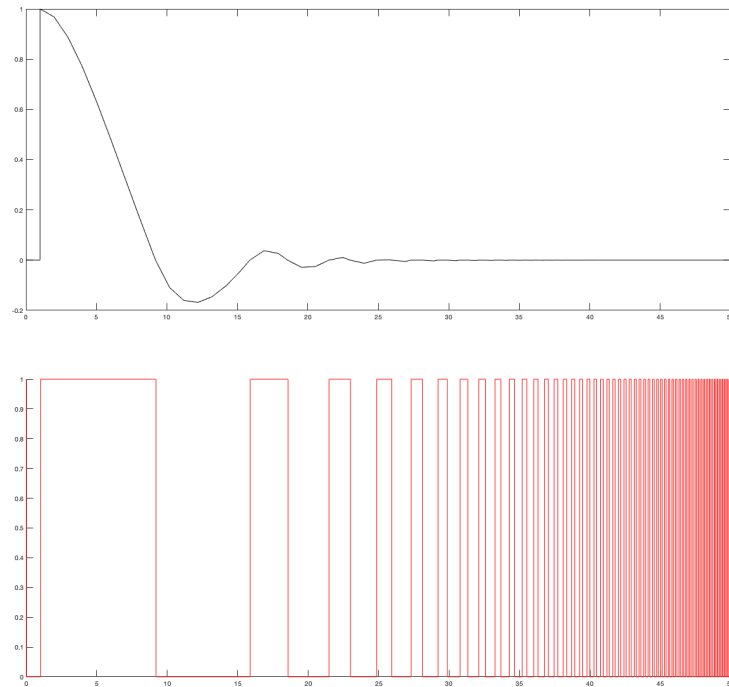
Rys. 1: Schemat symulacji regulatora dwupołożeniowego.

Aby zasymulować regulator dwupołożeniowy bez histerezy trzeba ustawić parametry 'On' i 'Off' obiektu 'Relay' na wartość 0. Za przeprowadzenie testu odpowiedzialna jest poniższa funkcja.

```
1 function testDwupolozeniowyBezHisterezy()
2 load_system('dwupolozeniowy.slx');
3 hold on;
4
5 set_param('dwupolozeniowy/Relay', 'OnSwitchValue', num2str(0));
6 set_param('dwupolozeniowy/Relay', 'OffSwitchValue', num2str(0));
7
8 sim('dwupolozeniowy.slx');
9 figure(1);
10 plot(u.time, u.signals.values, '-r');
11 figure(2);
12 plot(e.time, e.signals.values, '-k');
13 end
```

Fun. 1: Funkcja testująca regulator dwupołożeniowy bez histerezy.

Po wywołaniu tej funkcji otrzymałem poniższe wykresy.

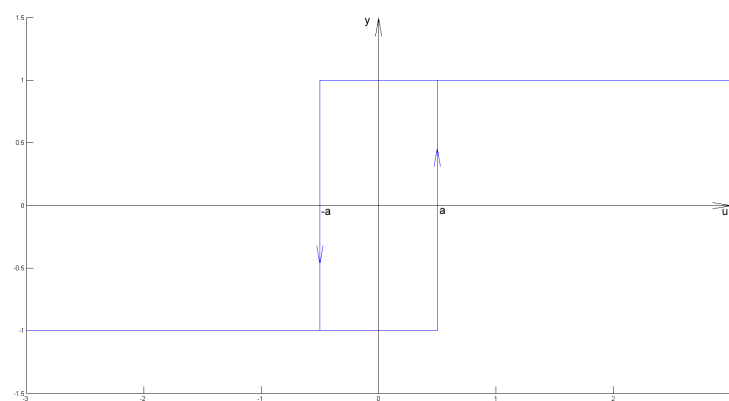


Rys. 2: Wykresy $\varepsilon(t)$, oraz $u(t)$ dla regulatora dwupołożeniowego bez histerezy.

Widać z nich, że wykres błędu oscyluje z czasem coraz bliżej zera, więc wartość na wyjściu jest blisko pożądaną, lecz z czasem rośnie również częstotliwość przełączania regulatora, przez co znacznie rośnie jego zużycie i maleje żywotność.

2.1.2 Regulator dwupołożeniowy z histerezą

Histereza dwupoziomowa sprawia, że przełączanie stanu z wysokiego na niski nie zachodzi przy tej samej wartości co przełączanie stanu z niskiego na wysoki. Jej działanie przedstawione zostało na poniższym obrazku.



Rys. 3: Jak działa histereza dwupoziomowa.

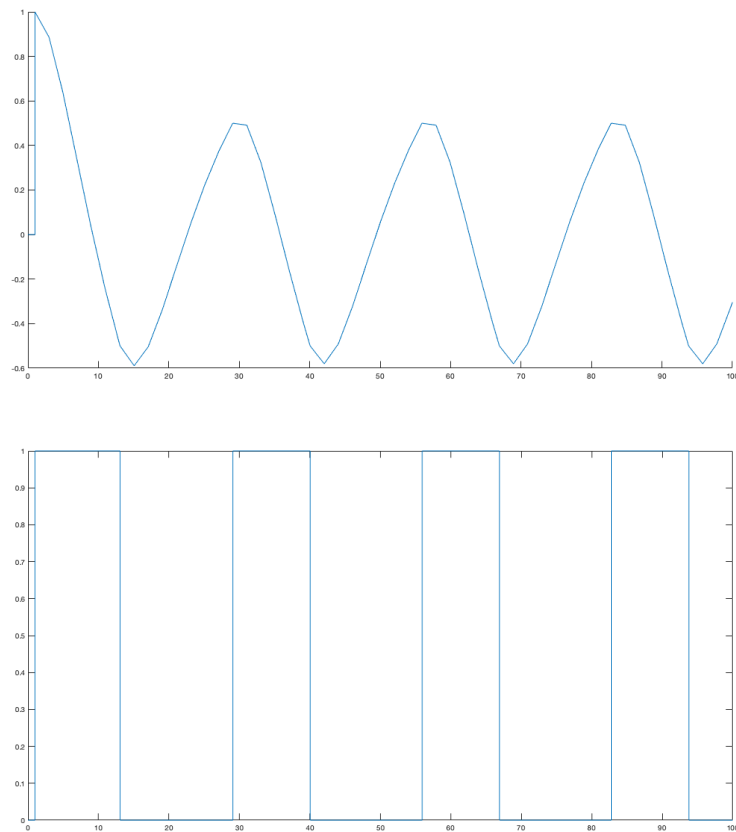
Symulacje regulatora dwupoziomowego z histerezą przeprowadzę na tym samym modelu, który posłużył mi w zadaniu 2.1.1, jednak w obiekcie 'Relay' parametry 'On' i 'Off' będą różne, dzięki czemu ustawimy żadaną histerezę.

Za przeprowadzenie testu odpowiedzialna jest poniższa funkcja.

```
1 function testDwupolozeniowyZHistereza(step, stop, drawU)
2 load_system('dwupolozeniowy.slx');
3 hold on;
4 i = step;
5
6 while (i <= stop)
7 set_param('dwupolozeniowy/Relay', 'OnSwitchValue', num2str(i));
8 set_param('dwupolozeniowy/Relay', 'OffSwitchValue', num2str(-i));
9
10 sim('dwupolozeniowy.slx');
11 figure(1);
12
13 plot(e.time, e.signals.values, 'DisplayName', strcat('a=', num2str(i)));
14
15 if drawU
16 figure(2);
17 plot(u.time, u.signals.values);
18 end
19
20 i=i+step;
21 hold all;
22 end
23 end
```

Fun. 2: Funkcja testująca regulator dwupółłożeniowy z histerezą.

Po wywołaniu powyższej funkcji dla wartości (0.5, 0.6, true) otrzymałem poniższe wykresy.

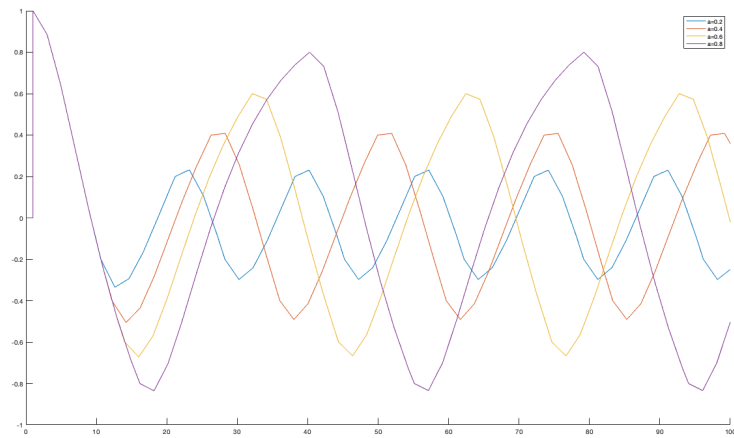


Rys. 4: Wykresy $\varepsilon(t)$, oraz $u(t)$ dla regulatora dwupółłożeniowego z histerezą.

Widać na nich, że amplituda oscylacji błęd wokół zera z czasem stabilizuje się, a liczba przełączeń znacznie maleje w stosunku do regulatora dwupółłożeniowego bez histerezy, a więc

i wydłuża się jego żywotność.

Aby zbadać wpływ zakresu histerezy na wykres $\varepsilon(t)$ wywołałem funkcję dla wartości (0.2, 0.8, false), w wyniku czego otrzymałem poniższe wykresy.

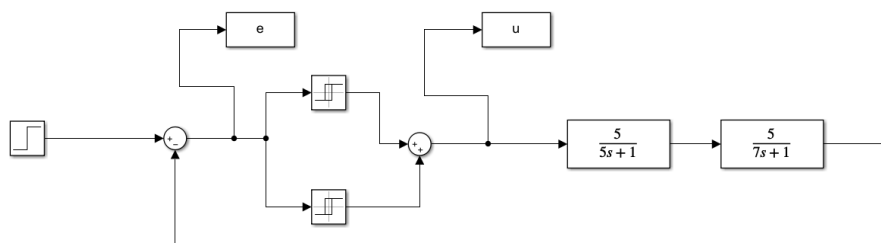


Rys. 5: Wykres $\varepsilon(t)$ dla regulatora dwupołożeniowego z histerezą przy rosnącym a .

Widać na nich, że w miarę zwiększania zakresu histerezy rośnie amplituda oscylacji wykresu błędu wokół zera, oraz maleje częstotliwość przełączeń regulatora.

2.1.3 Regulator trójpółożeniowy bez histerezy

Schemat systemu do symulacji regulatora trójpółożeniowego przedstawiony został na poniższym rysunku.



Rys. 6: Schemat symulacji regulatora trójpółożeniowego.

Aby zasymulować regulator trójpółożeniowy bez histerezy należy parametry 'On' i 'Off' obiektów 'Relay1' i 'Relay2' na wartość 0.

Za przeprowadzenie testu odpowiedzialna jest poniższa funkcja.

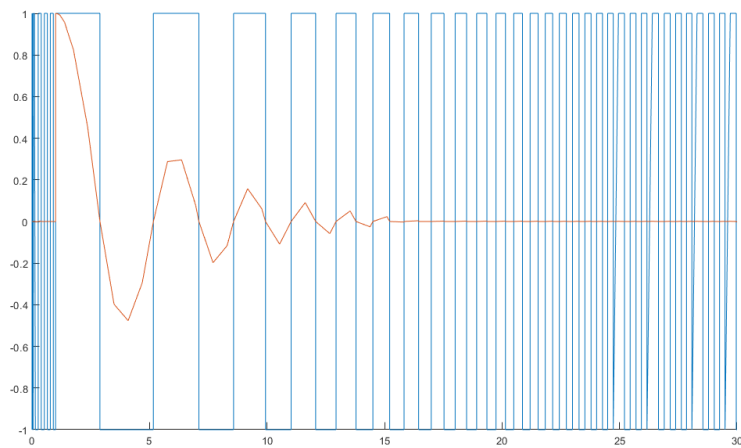
```

1 function testTrojpolozeniowyBezHisterezy()
2 load_system('trojpolozeniowy.slx');
3 hold on;
4
5
6 set_param('trojpolozeniowy/Relay1', 'OnSwitchValue', num2str(0));
7 set_param('trojpolozeniowy/Relay1', 'OffSwitchValue', num2str(0));
8 set_param('trojpolozeniowy/Relay2', 'OnSwitchValue', num2str(0));
9 set_param('trojpolozeniowy/Relay2', 'OffSwitchValue', num2str(0));
10
11
12 sim('trojpolozeniowy.slx');
13 figure(1);
14 plot(u.time, u.signals.values);
15 plot(e.time, e.signals.values);
16 %plot(fun.time, fun.signals.values);
17 end

```

Fun. 3: Funkcja testująca regulator trójpółożeniowy bez histerezy.

Po wywołaniu tej funkcji otrzymałem poniższy wykres.

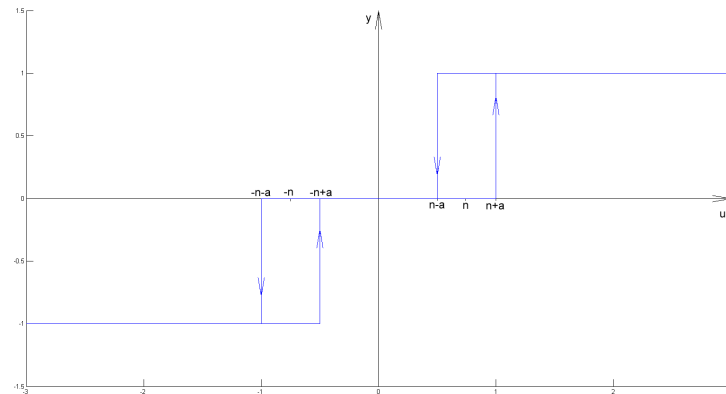


Rys. 7: Wykresy $\varepsilon(t)$, oraz $u(t)$ dla regulatora trójpółożeniowego bez histerezy.

Widać z nich, że wykres błędu oscyluje z czasem coraz bliżej zera, więc wartość na wyjściu jest blisko pożądaney, lecz z czasem rośnie również częstotliwość przełączania regulatora, przez co znacznie rośnie jego zużycie i maleje żywotność. Widać również, że regulator trójpółożeniowy nie wszedł w stan zerowy ani razu. Dzieje się tak, ponieważ przełączenia regulatora są zbyt szybkie.

2.1.4 Regulator trójpółożeniowy z histerezą

Histereza trójpółożeniowa to nic innego niż połączenie dwóch histerez dwupółożeniowych. Jej działanie przedstawione zostało na poniższym obrazku.



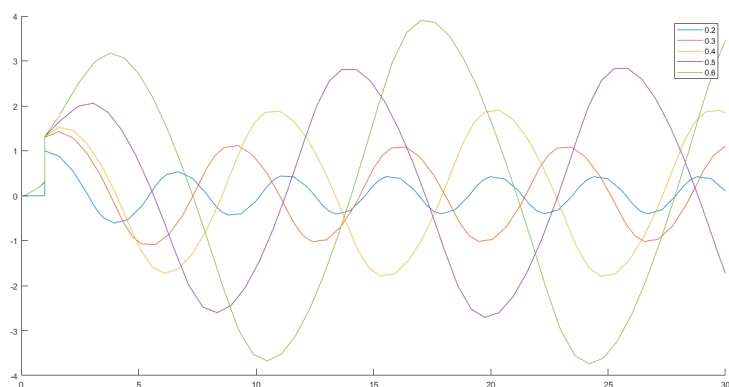
Rys. 8: Jak działa histereza trójpółożeniowa.

Symulacje regulatora trójpółożeniowego z histerezą przeprowadzę na tym samym modelu, który posłużył mi w zadaniu 2.1.3, jednak w obiektach 'Relay1' i 'Relay2' parametry 'On' i 'Off' będą różne, dzięki czemu uzyskam żadaną histerezę. Za przeprowadzenie testu odpowiedzialna jest poniższa funkcja.

```
1 function testTrojpolozeniowyZHistereza(start, step, stop)
2 load_system('trojpolozeniowy.slx');
3 hold on;
4 i = 1;
5
6 s1 = start;
7 s2 = start;
8 while(s1 <= stop)
9
10 s1 = s1 + step;
11 s2 = s2 - step;
12
13 set_param('trojpolozeniowy/Relay1', 'OnSwitchValue', num2str(s1*i));
14 set_param('trojpolozeniowy/Relay1', 'OffSwitchValue', num2str(s2*i));
15 set_param('trojpolozeniowy/Relay2', 'OnSwitchValue', num2str(-(s2*i)));
16 set_param('trojpolozeniowy/Relay2', 'OffSwitchValue', num2str(-(s1*i)));
17
18
19 sim('trojpolozeniowy.slx');
20 figure(1);
21 %plot(relay.time, relay.signals.values);
22 plot(e.time, e.signals.values, 'DisplayName', num2str(s1));
23 %plot(fun.time, fun.signals.values);
24 i = i + 1;
25 end
26 hold all;
27 end
```

Fun. 4: Funkcja testująca regulator trójpółożeniowy z histerezą.

Po wywołaniu powyższej funkcji dla wartości (0.2, 0.1, 0.8) otrzymałem poniższy wykres.



Rys. 9: Wykresy $\varepsilon(t)$, oraz $u(t)$ dla regulatora trójpoleźniowego z histerezą.

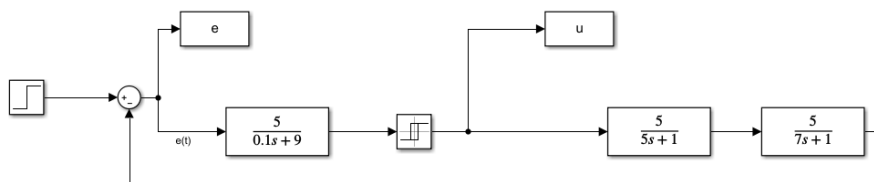
Widać na nim, że w miarę zwiększania zakresu histerezy rośnie amplituda oscylacji wokół zera wykresu błędu.

2.2 Zastosowanie członów korekcyjnych.

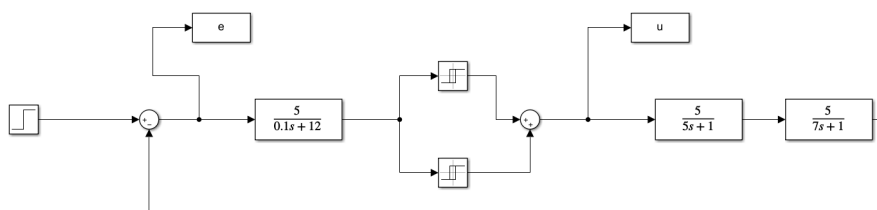
Podstawową zaletą regulacji dwupoleźniowej jest prostota realizacji. Niestety, cecha ta jest okupiona pogorszeniem jakości parametrów regulacji w porównaniu regulacją ciągłą. Jedną z możliwości poprawienia jakości regulacji jest zastosowanie układu z korekcją. Człon korekcyjny posiada następującą transmitancję:

$$G_w = \frac{k}{T_k s + 1} \quad (2)$$

2.2.1 Modyfikacja systemów z zadania 1



Rys. 10: Schemat regulatora dwupoleźniowego z korekcją.



Rys. 11: Schemat regulatora trójpoleźniowego z korekcją.

Człon korekcyjny powoduje modyfikację sygnału błędny dzięki czemu przełączanie następuje szybciej.

2.2.2 Doświadczalny dobor parametrów członu korekcyjnego.

1. Regulator dwupołożeniowy.

Do sprawdzania regulatorów wykorzystałem następujące funkcje testujące.

```
1 function testDwupolozeniowyKor(start, step, stop)
2 load_system('dwupolozeniowy_kor.slx');
3 hold on;
4 s=start;
5
6
7 while(s <= stop)
8 set_param('dwupolozeniowy_kor/Transfer Fcn3', 'Denominator', strcat('[' ,←
    num2str(s), ' 1]'));
9 %set_param('dwupolozeniowy_kor/Transfer Fcn3', 'Denominator', strcat(" [0.1 ←
    ", num2str(s), ' ]'));
10 %set_param('dwupolozeniowy_kor/Transfer Fcn3', 'Denominator', strcat(" [0.1 ←
    0.3]"));
11 sim('dwupolozeniowy_kor.slx');
12
13 figure(1);
14 plot(e.time, e.signals.values, 'DisplayName', strcat('k=', num2str(s)));
15 s= s+step;
16
17 end
18
19
20 hold all;
21
22
23 end
```

Fun. 5: Funkcja testująca regulator dwupoziomowy z korekcją zmianą T.

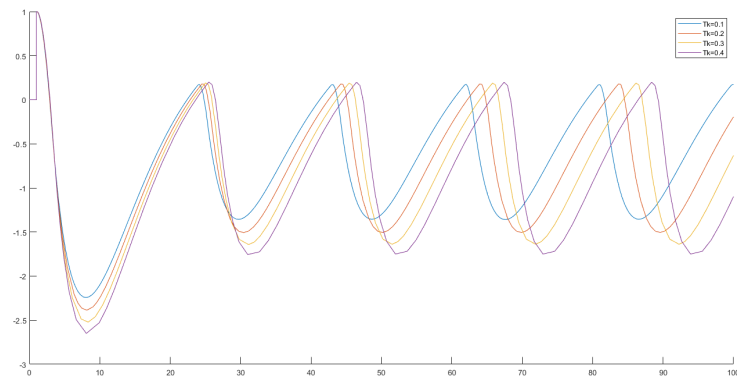
Analogicznie do testowania k

```
1 set_param('dwupolozeniowy_kor/Transfer Fcn3', 'Denominator', strcat(" [0.1 ",←
    num2str(s), ' ]'));

```

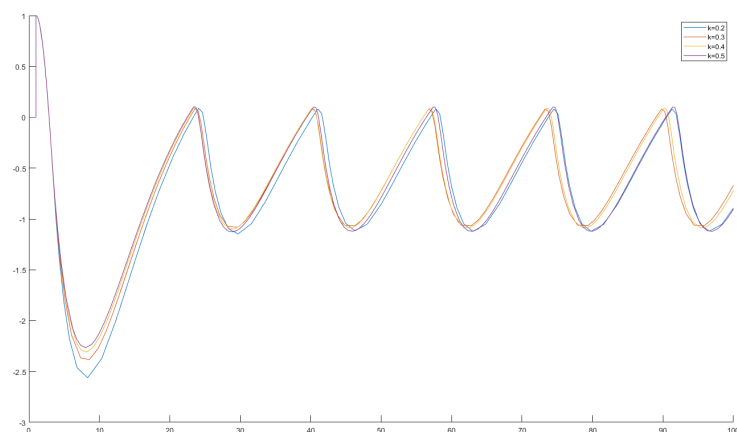
Fun. 6: set param dla k.

Na początek testowany był układ z histerezą. Zadanie polegało on doświadczalnym dobraniu parametrów układu korekcyjnego. Parametrami nas interesującymi są T_k , k .



Rys. 12: Zmiana T_k przy stałym $k=1$

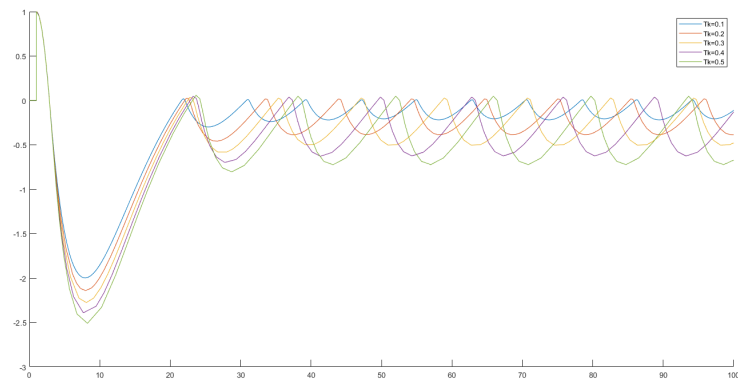
Z wyresu można zauważyć, że dla wartości $k=1$, wartość optymalna $T_k=0.1$ ze względu na najmniejsze wachania wartości.



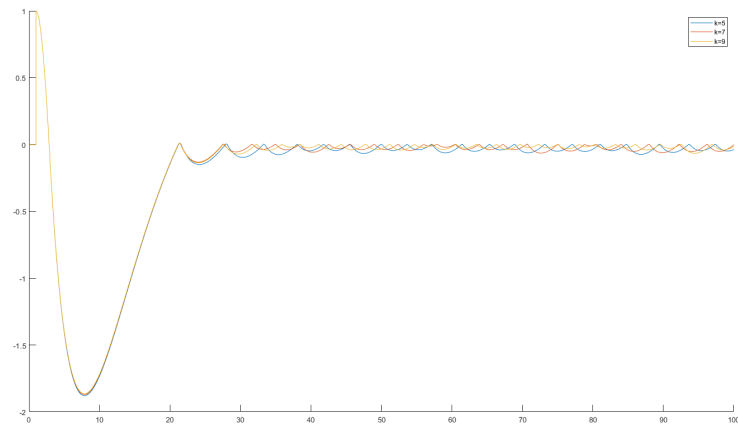
Rys. 13: Zmiana k przy stałym $T_k=0.1$

Gdy $T_k=0.1$ najbardziej optymalne $k=0.3$ ze względu, że oscyluje najmniej.

Dla układu bez histerezy otrzymałem zbliżone wyniki:



Rys. 14: Zmiana T_k przy stałym $k=1$



Rys. 15: Zmiana K

2. Regulator trójpółożeniowy Do testowania regulatora trójpółożeniowego wykorzystałem poniższe funkcje.

```

1 function testTrojpolozeniowyBezHisterezyKor(start, step, stop)
2 load_system('trojpolozeniowy_kor.slx');
3 hold on;
4 i = 1;
5
6 s1 = start;
7 while(s1 <= stop)
8 set_param('trojpolozeniowy_kor/Relay1', 'OnSwitchValue', num2str(0));
9 set_param('trojpolozeniowy_kor/Relay1', 'OffSwitchValue', num2str(0));
10 set_param('trojpolozeniowy_kor/Relay2', 'OnSwitchValue', num2str(0));
11 set_param('trojpolozeniowy_kor/Relay2', 'OffSwitchValue', num2str(0));
12
13 %set_param('trojpolozeniowy_kor/Transfer Fcn3', 'Denominator', strcat('1', ←
    num2str(s1), ' 1'));
14 set_param('trojpolozeniowy_kor/Transfer Fcn3', 'Denominator', strcat("0.1 ←
    ", num2str(s1), ' '));
15
16
17 sim('trojpolozeniowy_kor.slx');
18 figure(1);
19 %plot(relay.time, relay.signals.values);
20 plot(e.time, e.signals.values, 'DisplayName', strcat('Tk=', num2str(s1)));
21 %plot(fun.time, fun.signals.values);
22 s1 = s1 + step;
23 i = i + 1;
24 end
25 hold all;
26 end

```

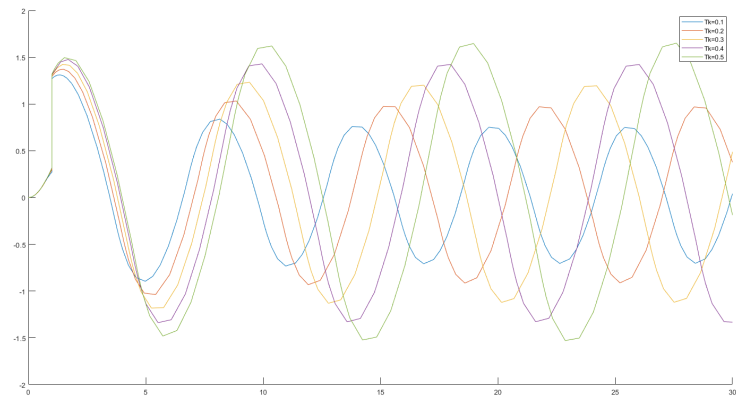
Fun. 7: Funkcja testująca regulator trójpółożeniowy bez histerezy.

```

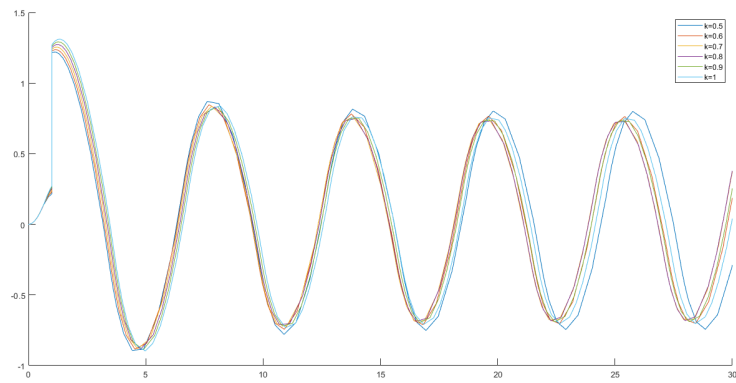
1 function testTrojpolozeniowyBezHisterezyKor(start, step, stop)
2 load_system('trojpolozeniowy_kor.slx');
3 hold on;
4 i = 1;
5
6 s1 = start;
7 while(s1 <= stop)
8 set_param('trojpolozeniowy_kor/Relay1', 'OnSwitchValue', num2str(0.8));
9 set_param('trojpolozeniowy_kor/Relay1', 'OffSwitchValue', num2str(-0.6));
10 set_param('trojpolozeniowy_kor/Relay2', 'OnSwitchValue', num2str(0.6));
11 set_param('trojpolozeniowy_kor/Relay2', 'OffSwitchValue', num2str(-0.8));
12
13 %set_param('trojpolozeniowy_kor/Transfer Fcn3', 'Denominator', strcat('1', ←
    num2str(s1), ' 1'));
14 set_param('trojpolozeniowy_kor/Transfer Fcn3', 'Denominator', strcat("0.1 ←
    ", num2str(s1), ' '));
15
16
17 sim('trojpolozeniowy_kor.slx');
18 figure(1);
19 %plot(relay.time, relay.signals.values);
20 plot(e.time, e.signals.values, 'DisplayName', strcat('k=', num2str(s1)));
21 %plot(fun.time, fun.signals.values);
22 s1 = s1 + step;
23 i = i + 1;
24 end
25 hold all;
26 end

```

Fun. 8: Funkcja testująca regulator trójpółożeniowy z histerezą.



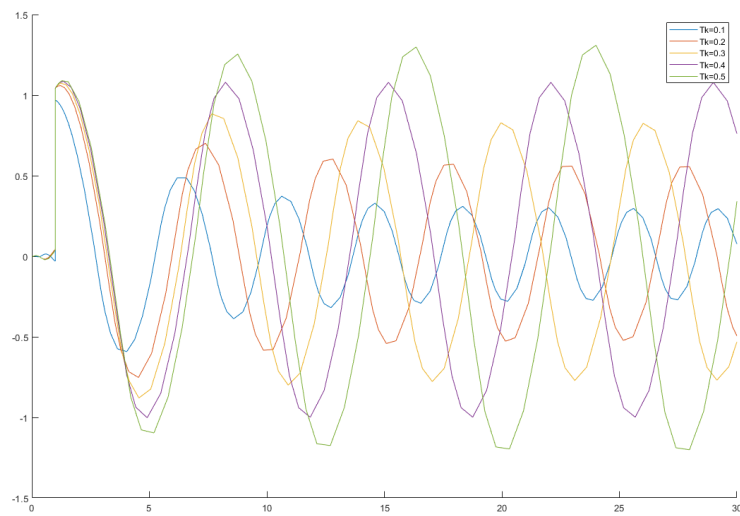
Rys. 16: Regulator trójpółeniowy z histerezą przy zmianie T_k



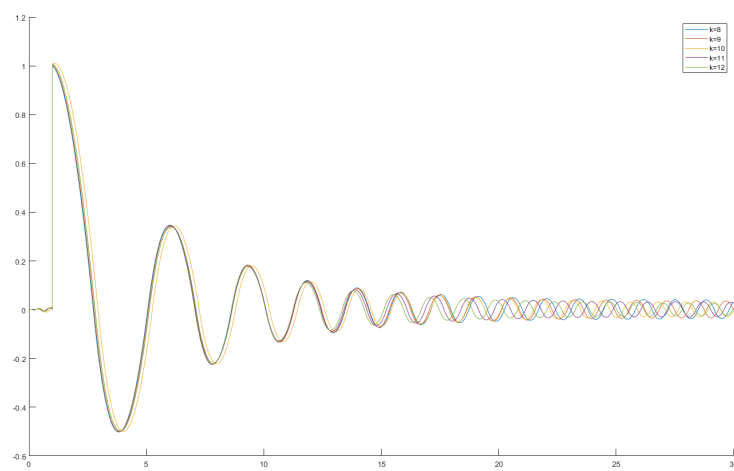
Rys. 17: Regulator trójpółeniowy z histerezą przy zmianie k

Wybrane parametry to: $T_k=0.1$ $k=1$

Dla układu bez histerezy wykonałem takie same pomiary.



Rys. 18: Regulator trójpółeniowy bez histerezy przy zmianie T_k



Rys. 19: Regulator trójpółeniowy bez histerezy przy zmianie k

3 Wnioski.

Histereza umożliwia zmniejszenie zużycia sprzętu przez zmniejszenie częstotliwości przełączeń między stanami, kosztem odchyłków od pożądaných wartości. Wybór zakresu histerezy zależy od systemu w którym działa regulator oraz od tego jakiej dokładności wyjściowej wymaga.

Regulator trójpołożeniowy pozwala na określenie 3 stanów, dzięki którym uzyskujemy większą kontrolę nad działaniem systemu.

Człon korekcyjny pozwala dostosować system do naszych potrzeb, wpływa on na zmianę częstotliwości przełączania.