

Informatyczne Systemy Sterowania

Ćwiczenie 3: Regulacja dwu- i trójpółożeniowa

Adam Jordanek 168139, Tomasz Klimek 168092

1 grudnia 2018

1 Wstęp

1.1 Cel ćwiczenia

Celem tego ćwiczenia jest symulacja działania systemu regulacji z przekaźnikami dwu- i trójpółożeniowymi. Ćwiczenie ma umożliwić zapoznanie się z nieliniowymi algorytmami sterowania (przełącznikami dwu- i trójpółożeniowymi) 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 przekaźników na przebieg błędu regulacji
 - (a) Regulator dwupółożeniowy bez histerezy.
 - (b) Regulator dwupółożeniowy z histerezą.
 - (c) Regulator trójpółożeniowy bez histerezy.
 - (d) Regulator trójpółożeniowy 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.

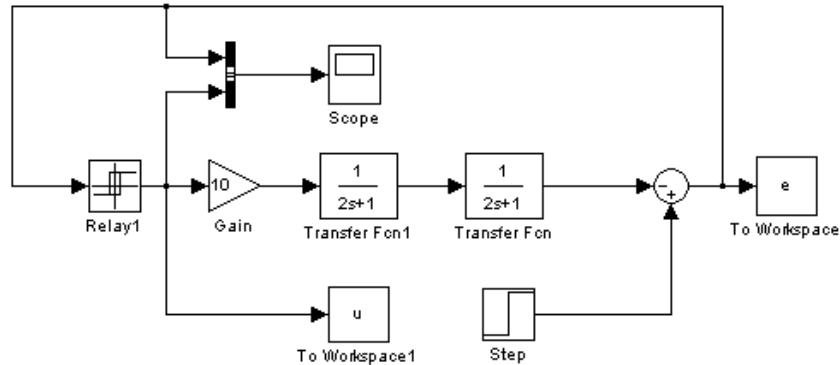
2 Realizacja planu i wyniki

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 służącego nam do symulacji regulatora dwupołożeniowego przedstawiony został na poniższym rysunku.



Rys. 1: Schemat symulacji regulatora dwupołożeniowego.

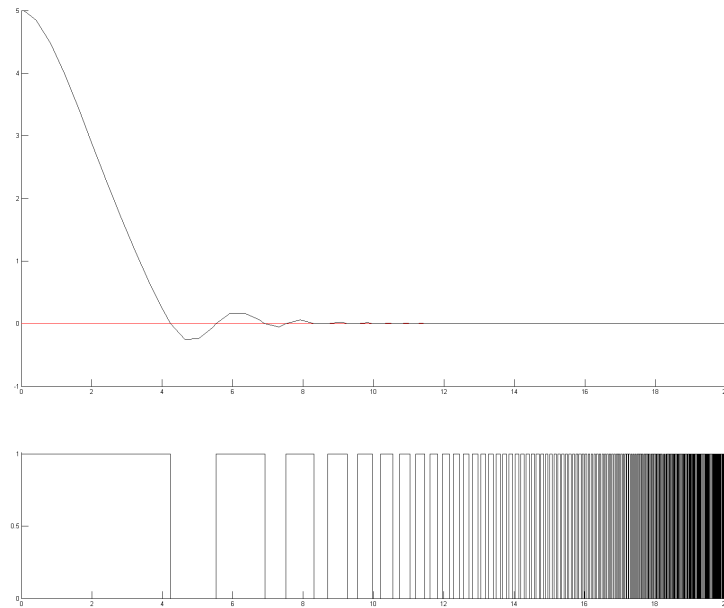
Aby zasymulować regulator dwupołożeniowy bez histerezy musimy ustawić parametry 'On' i 'Off' obiektu 'Relay1' na wartość 0.

Za przeprowadzenie testu odpowiedzialna jest poniższa funkcja.

```
1 function testDwupoziomowy1(step, flag)
2 load_system('dwupoziomowy.mdl');
3 hold on;
4
5 set_param('dwupoziomowy/Relay1', 'On_switch_value', num2str(0));
6 set_param('dwupoziomowy/Relay1', 'Off_switch_value', num2str(0));
7
8 sim('dwupoziomowy.mdl');
9 figure(1);
10 uwy= u.signals.values;
11 plot(tout, uwy, '-k');
12
13 figure(2);
14
15 if flag
16     hold on;
17     plot([0 20], [0 0], '-r');
18 end
19
20 ewy= e.signals.values;
21 plot(tout, ewy, '-k');
22 end
```

Fun. 1: Funkcja testująca regulator dwupołożeniowy bez histerezy.

Po wywołaniu tej funkcji otrzymaliśmy poniższe wykresy.

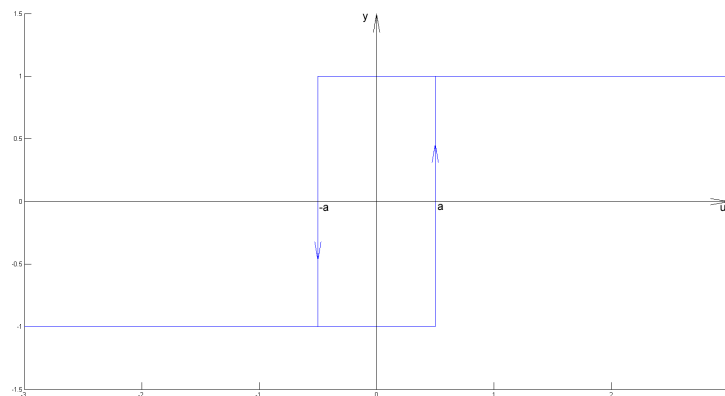


Rys. 2: Wykresy $\varepsilon(t)$, oraz $u(t)$ dla regulatora dwupołożeniowego bez histerezy.

Widzimy 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.

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 przeprowadzimy na tym samym modelu, który posłużył nam w zadaniu 2.1.1, jednak w obiekcie 'Relay1' parametry 'On' i 'Off' będą różne, dzięki czemu ustawimy żadaną histerezę.

Za przeprowadzenie testu odpowiedzialna jest poniższa funkcja.

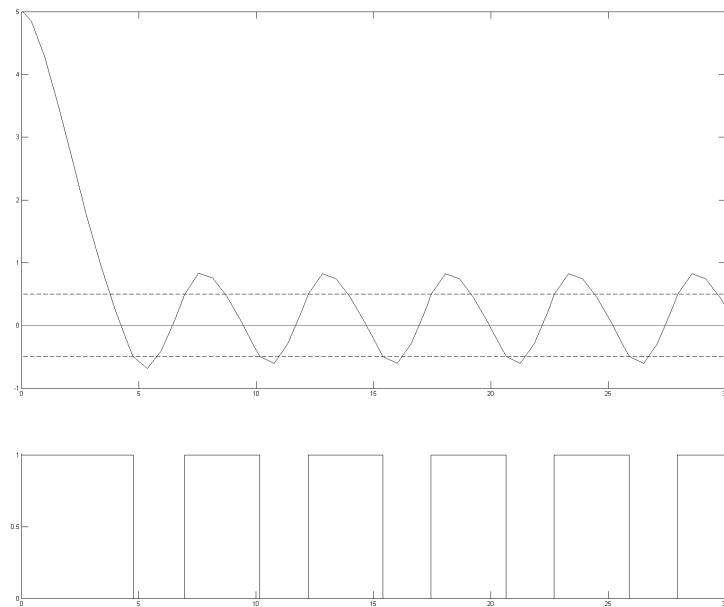
```

1 function testDwupoziomowy2(step, stop, flag)
2 load_system('dwupoziomowy.mdl');
3 hold on;
4 i=1;
5 color = char('y','k','b','g','r','m');
6
7 while (step*i <= stop)
8     set_param('dwupoziomowy/Relay1', 'On_switch_value', num2str(step*i));
9     set_param('dwupoziomowy/Relay1', 'Off_switch_value', num2str(-step*i));
10
11     sim('dwupoziomowy.mdl');
12     figure(1);
13     uwy= u.signals.values;
14     plot(tout, uwy, 'Color', color(mod(i,6)+1));
15
16     figure(2);
17     ewy= e.signals.values;
18     plot(tout, ewy, 'Color', color(mod(i,6)+1));
19
20     if flag
21         hold on;
22         plot([0 30], [0 0], '-r');
23         plot([0 30], [step*i step*i], strcat('--', color(mod(i,6)+1)));
24         plot([0 30], [-step*i -step*i], strcat('--', color(mod(i,6)+1)));
25     end
26
27     i=i+1;
28     hold all;
29 end
30 end

```

Fun. 2: Funkcja testująca regulator dwupołożeniowy z histerezą.

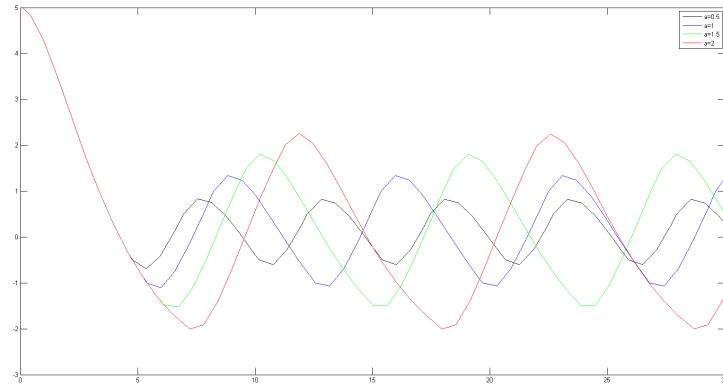
Po wywołaniu powyższej funkcji dla wartości (0.5, 0.6, true) otrzymaliśmy poniższe wykresy.



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

Widzimy na nich, że amplituda oscylacji błędu wokół zera z czasem stabilizuje się, a liczba przełączeń znacznie maleje w stosunku do regulatora dwupołożeniowego bez histerezy, a więc i wydłuża się jego żywotność.

Aby zbadać wpływ zakresu histerezy na wykres $\varepsilon(t)$ wywołamy funkcję dla wartości (0.5, 2, false), w wyniku czego otrzymamy poniższe wykresy.

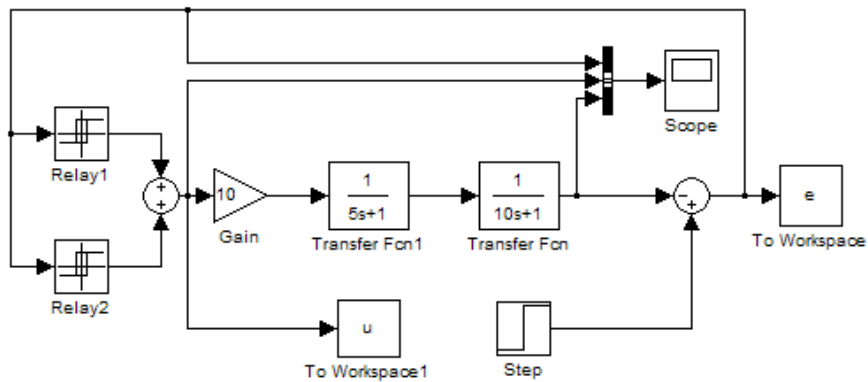


Rys. 5: Wykres $\varepsilon(t)$ dla regulatora dwupołożeniowego z histerezą przy rosnącym a .

Widzimy 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 służącego nam do symulacji regulatora trójpółożeniowego przedstawiony został na poniższym rysunku.



Rys. 6: Schemat symulacji regulatora trójpółożeniowego.

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

Za przeprowadzenie testu odpowiedzialna jest poniższa funkcja.

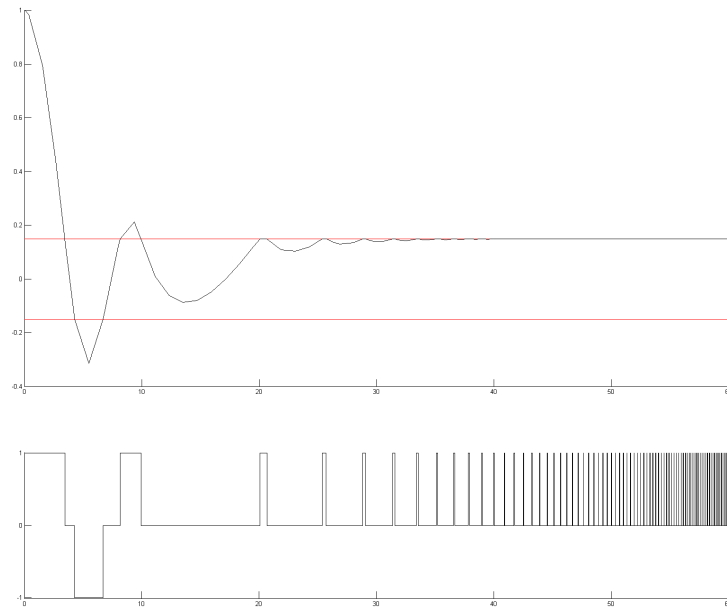
```

1 function testTrojpoziomowy(n, step, stop, flag)
2 load_system('trojpoziomowy.mdl');
3 hold on;
4 set_param('trojpoziomowy/Relay1', 'Off_switch_value', num2str(n));
5 set_param('trojpoziomowy/Relay1', 'On_switch_value', num2str(n));
6 set_param('trojpoziomowy/Relay2', 'Off_switch_value', num2str(-n));
7 set_param('trojpoziomowy/Relay2', 'On_switch_value', num2str(-n));
8
9 sim('trojpoziomowy.mdl');
10
11 figure(1);
12 uwy= u.signals.values;
13 plot(tout, uwy, '-k');
14
15 figure(2);
16
17 if flag
18     hold on;
19     plot([0 30], [n n], '-r');
20     plot([0 30], [-n -n], '-r');
21 end
22
23 ewy= e.signals.values;
24 plot(tout, ewy, '-k');
25 end

```

Fun. 3: Funkcja testująca regulator trójpółożeniowy bez histerezy.

Po wywołaniu tej funkcji otrzymaliśmy poniższe wykresy.



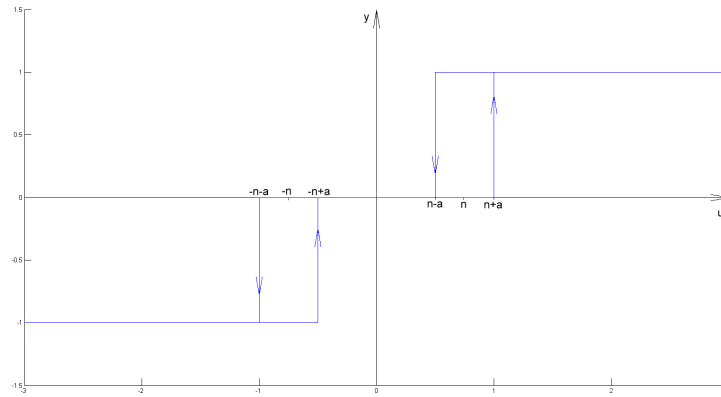
Rys. 7: Wykresy $\varepsilon(t)$, oraz $u(t)$ dla regulatora trójpółożeniowego bez histerezy.

Widzimy 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.

Widzimy również, że regulator trójpółożeniowy w stan niski wszedł tylko raz, a później przełączał się tylko na stan wysoki i zerowy. Dzieje się tak, ponieważ przełączenia regulatora z czasem są coraz krótsze, więc tempo malenia wartości błędu nie ma szans wzrosnąć na tyle, aby 'przelecieć' siłą bezwładności systemu do wartości $-n$, gdzie nastąpiłoby przełączenie na stan niski.

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.

Symulację regulatora trójpółożeniowego z histerezą przeprowadzimy na tym samym modelu, który posłużył nam w zadaniu 2.1.3, jednak w obiektach 'Relay1' i 'Relay2' parametry 'On' i 'Off' będą różne, dzięki czemu ustawimy żądaną histerezą. Za przeprowadzenie testu odpowiedzialna jest poniższa funkcja.

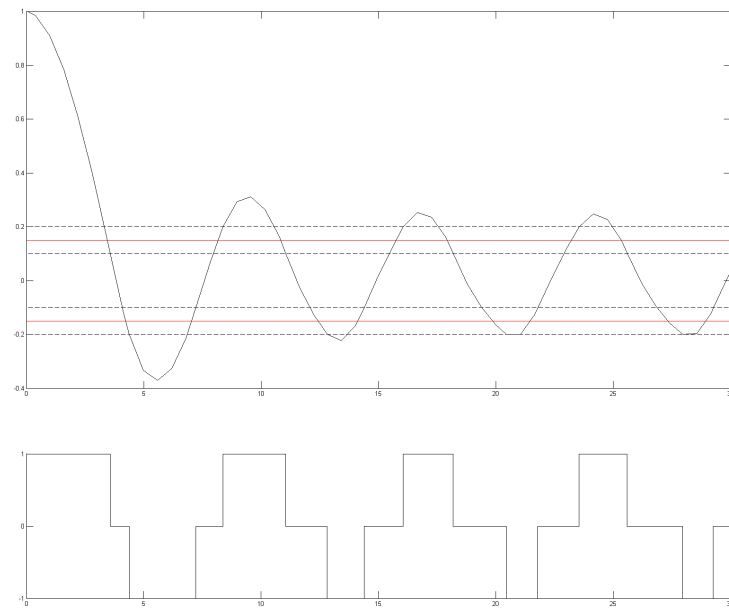
```

1 function testTrojpoziomowy(n, step, stop, flag)
2 load_system('trojpoziomowy.mdl');
3 hold on; i=1;
4 color = char('y', 'k', 'b', 'r', 'g', 'm');
5 while (step*i <= stop)
6     set_param('trojpoziomowy/Relay1', 'Off_switch_value', num2str(n-step*i));
7     set_param('trojpoziomowy/Relay1', 'On_switch_value', num2str(n+step*i));
8     set_param('trojpoziomowy/Relay2', 'Off_switch_value', num2str(-(n+step*i)));
9     set_param('trojpoziomowy/Relay2', 'On_switch_value', num2str(-(n-step*i)));
10    sim('trojpoziomowy.mdl');
11    figure(1);
12    uwy= u.signals.values;
13    plot(tout, uwy, 'Color', color(mod(i,6)+1));
14    figure(2);
15    ewy= e.signals.values;
16    plot(tout, ewy, 'Color', color(mod(i,6)+1));
17    if flag
18        plot([0 30], [n n], '-r');
19        plot([0 30], [-n -n], '-r');
20        plot([0 30], [n+step*i n+step*i], strcat('—', color(mod(i,6)+1)));
21        plot([0 30], [n-step*i n-step*i], strcat('—', color(mod(i,6)+1)));
22        plot([0 30], [-(n-step*i) -(n-step*i)], strcat('—', color(mod(i,6)+1)));
23        plot([0 30], [-(n+step*i) -(n+step*i)], strcat('—', color(mod(i,6)+1)));
24    end
25    i=i+1;
26    hold all;
27 end
28 end

```

Fun. 4: Funkcja testująca regulator trójpółożeniowy z histerezą.

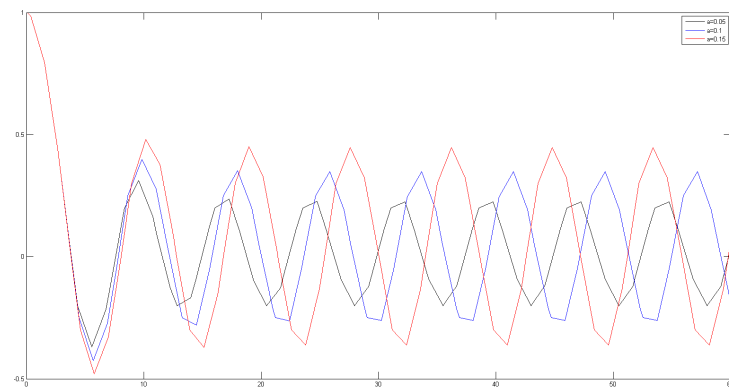
Po wywołaniu powyższej funkcji dla wartości (0.15, 0.05, 0.06, true) otrzymaliśmy poniższe wykresy.



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

Widzimy na nich, że amplituda oscylacji błędu wokół zera z czasem stabilizuje się, a liczba przełączeń znacznie maleje w stosunku do regulatora trójpółożeniowego bez histerezy, a więc i wydłuża się jego żywotność.

Aby zbadać wpływ zakresu histerezy na wykres $\varepsilon(t)$ wywołamy funkcję dla wartości (0.15, 0.05, 0.16, false), w wyniku czego otrzymamy poniższe wykresy.



Rys. 10: Wykres $\varepsilon(t)$ dla regulatora trójpółożeniowego z histerezą przy rosnącym a .

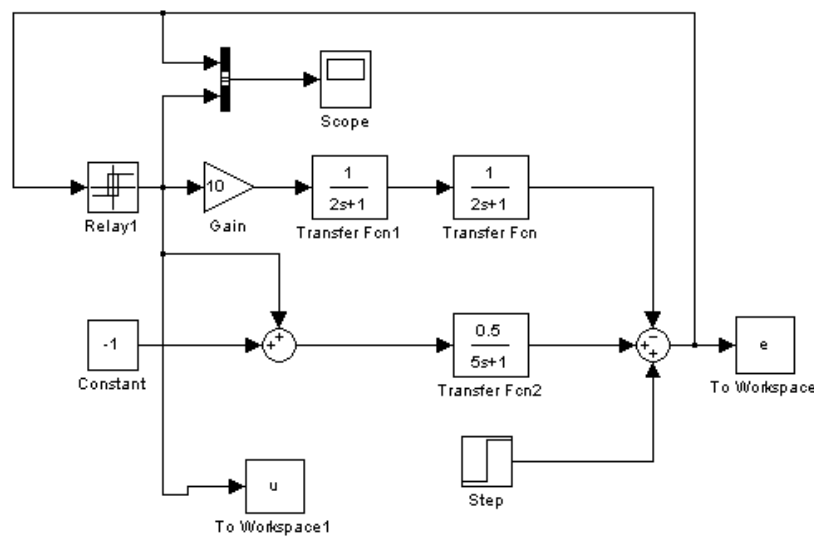
Widzimy na nich, że w miarę zwiększania zakresu histerezy rośnie amplituda oscylacji wokół zera wykresu błędu, oraz maleje częstotliwość przełączeń regulatora.

2.2 Zastosowanie członów korekcyjnych.

Podstawową zaletą regulacji dwupolożeniowej 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:

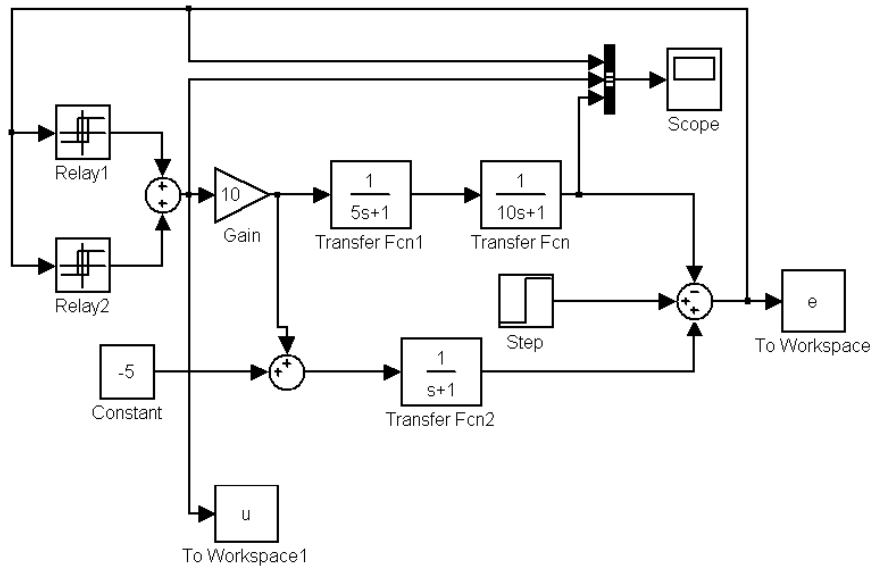
$$G_w = \frac{k}{T_k s + 1} \quad (2)$$

2.2.1 Modyfikacja systemów z zadania 1



Rys. 11: Schemat regulatora dwupolożeniowego z korekcją.

Człon korekcyjny włączony w obwód ujemnego sprzężenia zwrotnego przekaznika wprowadza modyfikację sygnału błędów $e(t)$, co powoduje częstsze niż bez korekcji przełączanie przekaznika. Korekcja jest skuteczna wówczas gdy, prędkość narastania sygnału z korektora w jest większa od prędkości zmian wejścia (dobór k_w i T_w). Zsumowanie sygnałów sprzężenia zwrotnego z obiektu i korektora oraz odjęcie ich sumy od wartości zadanej daje zastępczy sygnał błędów, którego prędkość i sposób narastania jest określona głównie przez sygnał.



Rys. 12: Schemat regulatora trojpołożeniowego z korekcją.

2.2.2 Doświadczalny dobór parametrów członu korekcyjnego.

1. Regulator dwupołożeniowy.

Do sprawdzania regulatorów wykorzystaliśmy następujące funkcje testujące.

```

1 function testDwupoziomowyKorekcjaT(start,step, stop)
2 load_system('dwupoziomowyKorekcja.mdl');
3 hold on;
4 i=1;
5 color = char('y','k','b','g','r','m');
6 legendtext{1}='';
7 legendtext1{1}='';
8 while (start+step*i <= stop)
9     set_param('dwupoziomowyKorekcja/Transfer Fcn2','Denominator', ←
10         strcat('1',num2str(start+step*i),' 1'));
11     sim('dwupoziomowyKorekcja.mdl');
12     figure(1);
13     uwy= u.signals.values;
14     plot(tout, uwy, 'Color', color(mod(i,6)+1));
15     legendtext{i}= strcat('Tk=', num2str(start+step*i));
16     legend(legendtext1);
17     figure(2);
18     ewy= e.signals.values;
19     plot(tout, ewy, 'Color', color(mod(i,6)+1));
20     legendtext1{i}= strcat('Tk=', num2str(start+step*i));
21     legend(legendtext1);
22     i=i+1;
23     hold all;
24 end
25 end

```

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

Analogicznie do testowania K i C

```

1 set_param('dwupoziomowyKorekcja/Transfer Fcn2','Numerator', num2str(start+ ←
2     step*i));

```

Fun. 6: set param dla K.

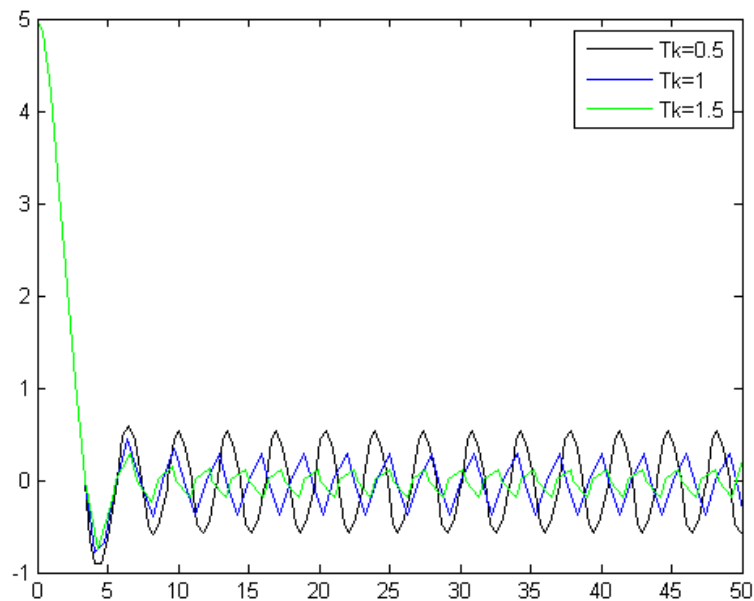
```

1 set_param('dwupoziomowyKorekcja/Constant','value', num2str(start+step*i));

```

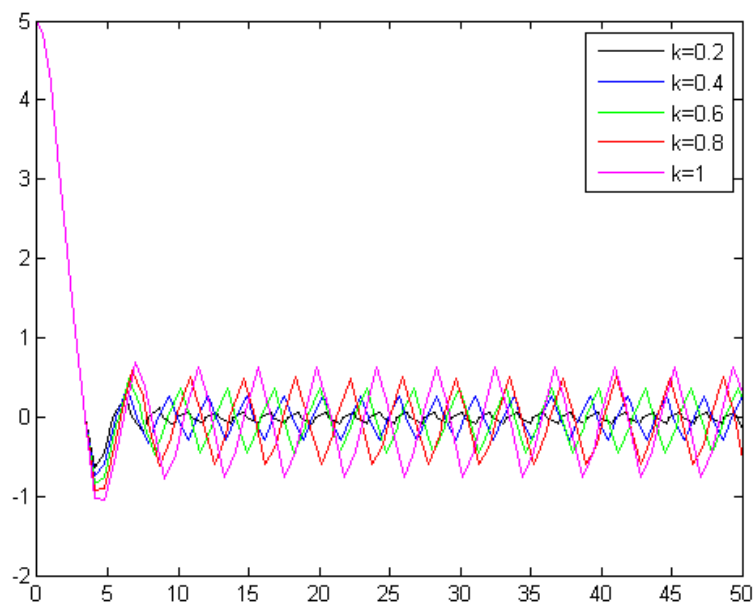
Fun. 7: set param dla C.

Na początek testowany był układ z histerezą. Zadanie polegało on doświadczalnym dobraniu parametrów układu korekcyjnego. Parametrami nas interesującymi jest T_k , K i C gdzie C to wartość stała (blok Constant).



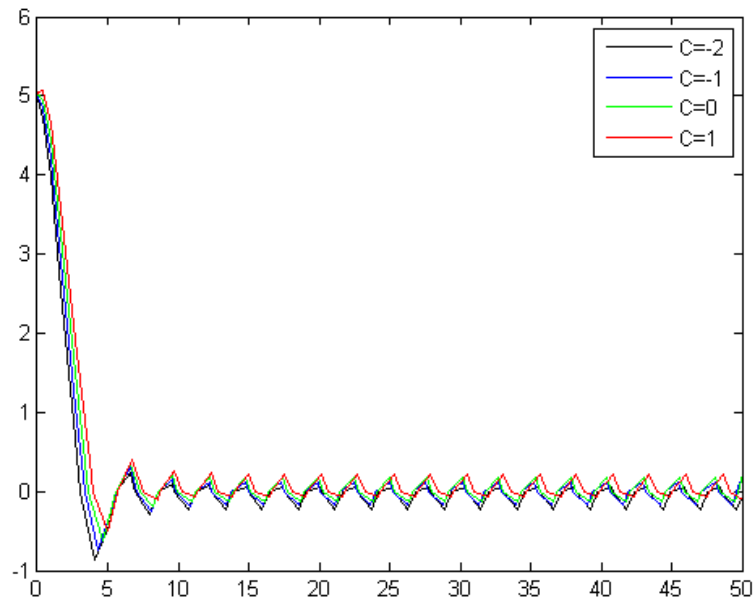
Rys. 13: Zmiana T_k przy stałym $k=1$ i $C=0$

Z Wyresu można zauważyć, że dla wartości $k=1$, wartość optymalna $T_k=1.5$ ze względu, że nie występują duże wachania wartości.



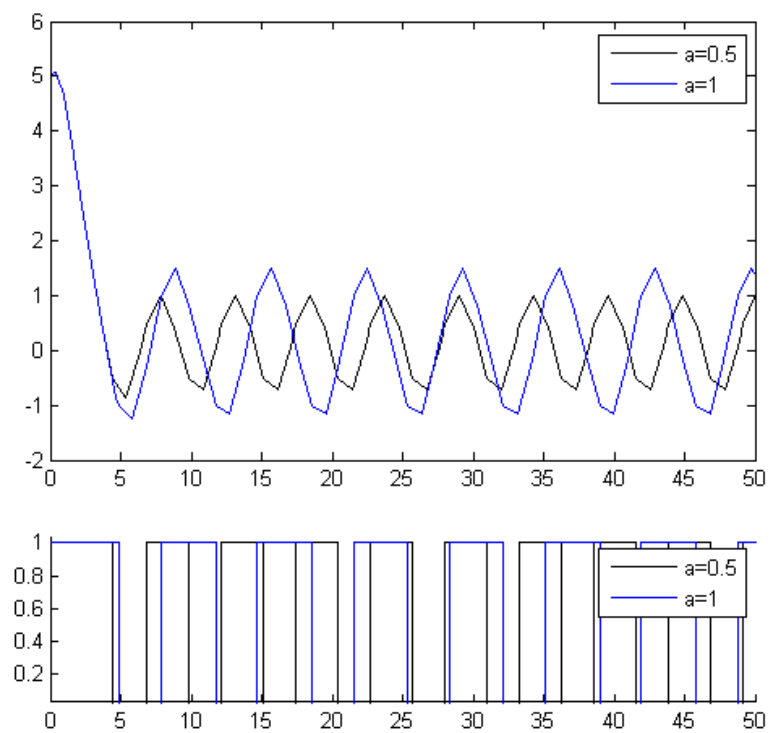
Rys. 14: Zmiana k przy stałym $T_k=1.5$ i $C=0$

Gdy $T_k=1$ najbardziej optymalne $k=0.5$ ze względu, że osiąga wartość 0. Z wykresu można zauważyć, że najlepiej układ się zachowuje kiedy C jest równe 0.



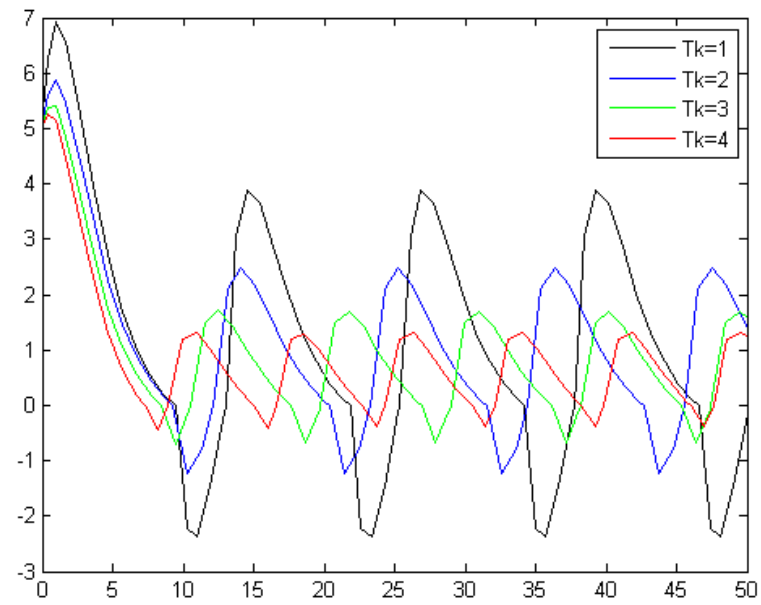
Rys. 15: Zmiana C przy stałym $k=0.5$ i $T_k=1.5$

Widać, że układ najlepiej zachoduje się gdy $K=0.5$ i $T_k=1$. Dla takich parametrów wykres dla $a=0.5$ wygląda:

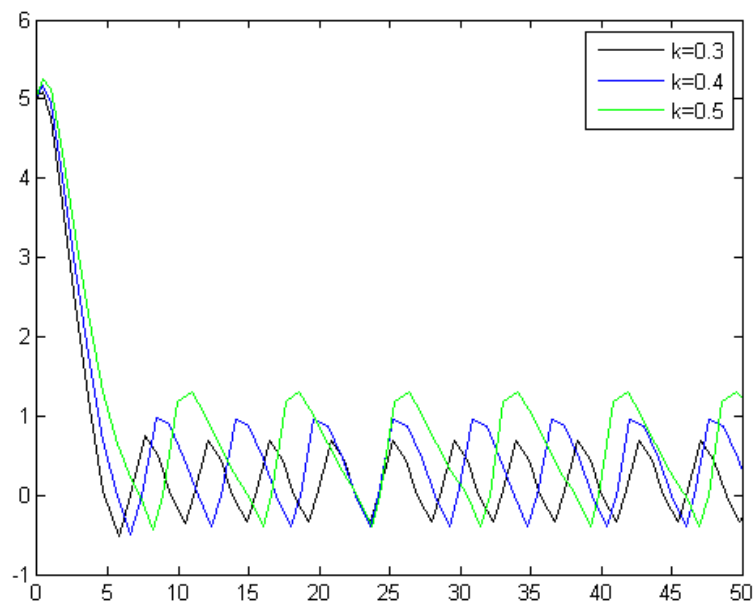


Rys. 16: Przy zmianie a

Dla układu bez histerezy otrzymaliśmy bardzo podobne wyniki:

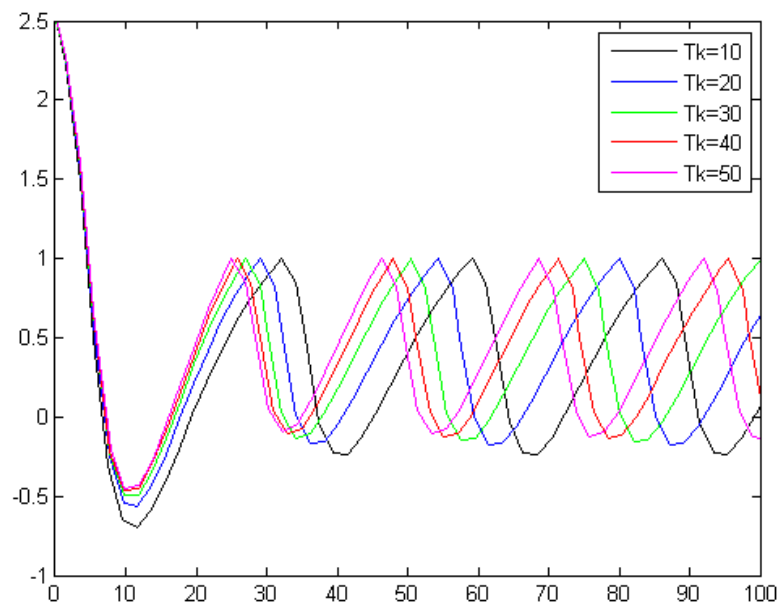


Rys. 17: Zmiana T_k przy stałym $k=0.5$ i $C=0$

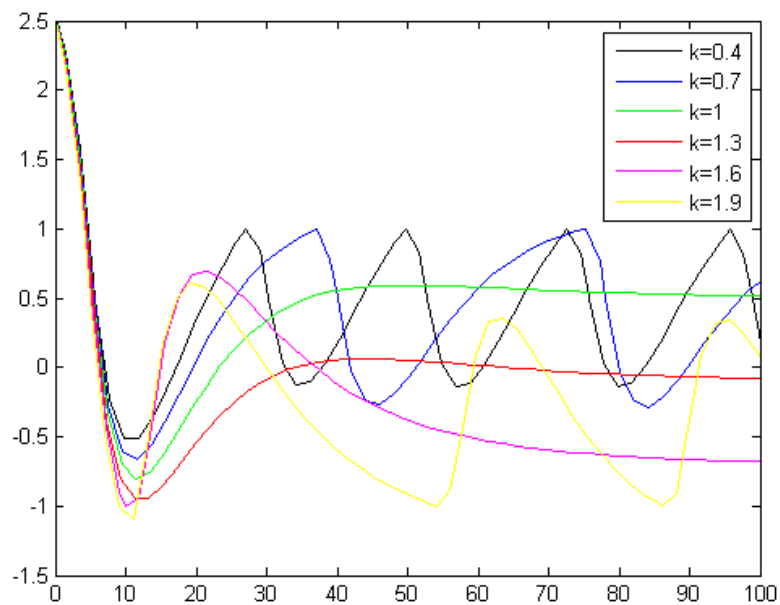


Rys. 18: Zmiana K

2. Regulator trójpółeniowy Do testowania regulatora wykorzystaliśmy funkcje z regulatora dwupółeniowego. Całe ćwiczzenie zostało wykonane w analogiczny sposób do poprzedniego.

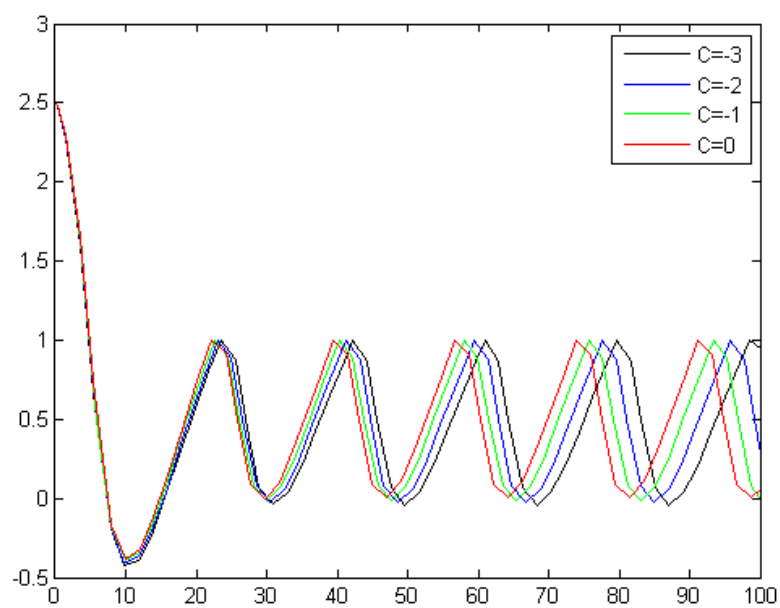


Rys. 19: Przy zmianie T_k



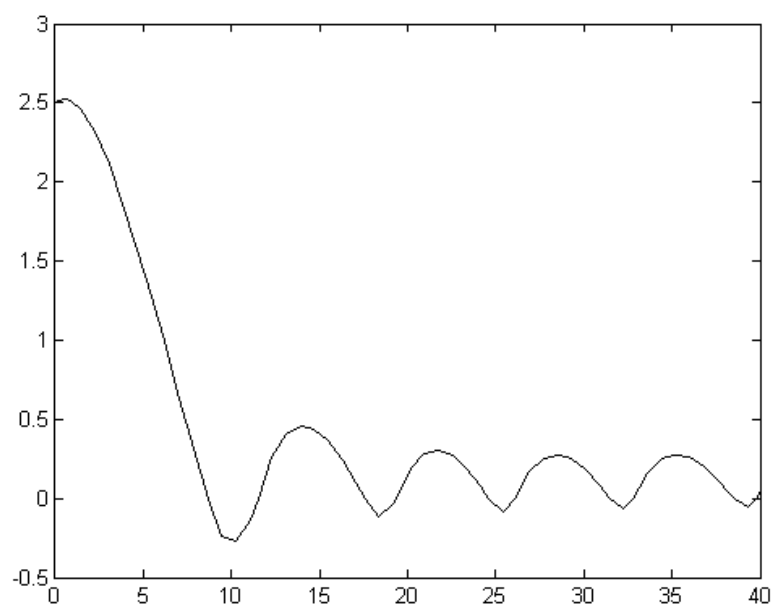
Rys. 20: Przy zmianie K

Z wykresów można wywnioskować, że stała nie ma wpływu na regulację. Wybrane parametry to: $T_k=20$ $K=0.1$ $C=-2$



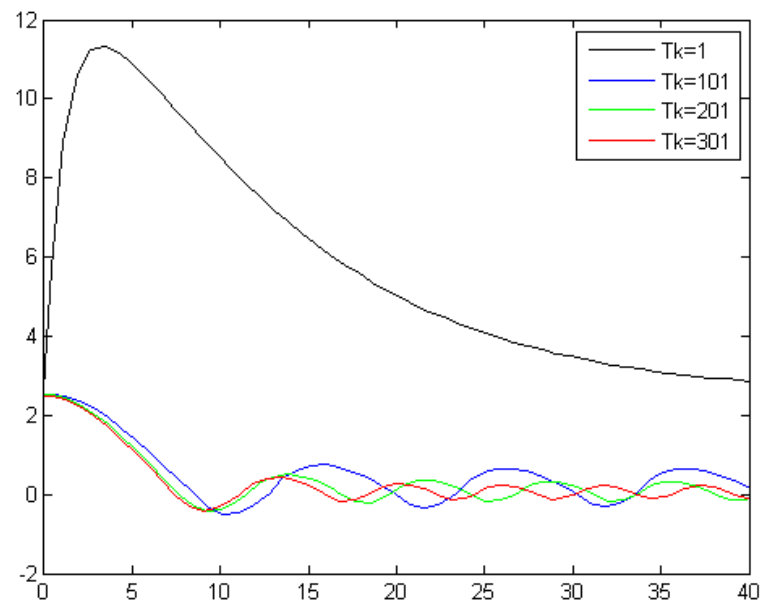
Rys. 21: Przy zmianie C

Przy zmianie 'a' wykres wygląda:

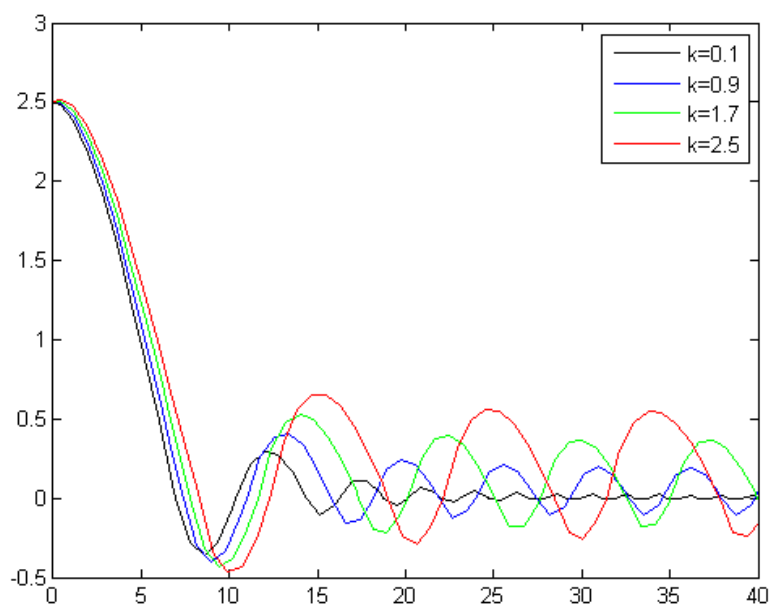


Rys. 22: Przy zmianie a

Dla układu bez histerezy wykonaliśmy takie same pomiary.



Rys. 23: Przy zmianie T_k bez histerezy



Rys. 24: Przy zmianie K bez histerezy

3 Wnioski.

Histeresa pozwala nam na zmniejszenie zużycia sprzętu przez zmniejszenie częstotliwości przełączeń między stanami, kosztem odchyłków od pożądaných wartości. Wybór zakresu histeresy 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, pozwala on na zmianę częstotliwości przełączania.

Koszt wydruku w ksero wzrósł dlatego dobrym rozwiązaniem będzie przerzucenie się na system elektroniczny. Pozwoli to na dodatkowe piwko oraz uratuje lasy deszczowe.