

# WOJSKOWA AKADEMIA TECHNICZNA

## WYDZIAŁ MECHATRONIKI I LOTNICTWA



### Projektowanie układów regulacji

*Temat: Projekt regulatora dyskretnego dla zadanych warunków końcowych*

Prowadzący: dr inż. Marek Jaworowicz

Wykonali:

inż. Violetta Munar Ernandes

inż. Piotr Prusaczyk

inż. Maciej Dobrzyński

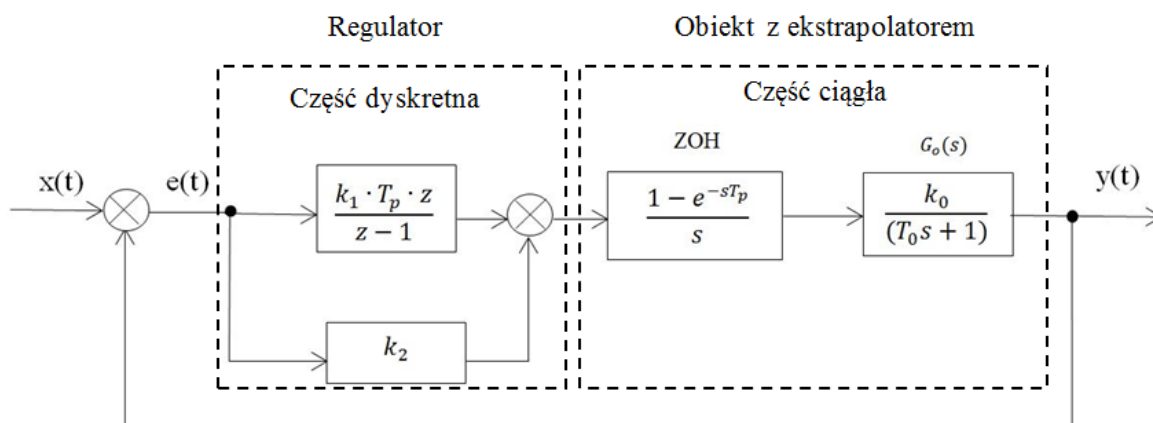
Grupa: A4A1S4

## Spis treści

1.	Sformułowanie problemu .....	3
1.1.	Założenia .....	3
1.2.	Rozwiązanie zadania .....	4
2.	Modele matematyczne układu regulacji .....	5
2.1.	Dyskretyzacja części ciągłej układu sterowania .....	5
2.2.	Transmitancja zastępcza regulatora .....	6
2.3.	Transmitancja układu otwartego .....	6
2.4.	Warunek aperiodyczności procesu przejściowego .....	7
2.5.	Wyznaczenie wartości stałej A .....	8
2.6.	Wyznaczenie parametrów regulatora .....	9
3.	Modele numeryczne układów regulacji .....	10
3.1.	Przypadek 1 - parametry regulatora R1 dla $tr1 = 60 \cdot Tp$ .....	10
3.2.	Przypadek 2 - Parametry regulatora R2 dla $tr2 = 30 \cdot Tp$ .....	10
3.3.	Transmitancje układu z regulatorem R1 .....	11
3.4.	Transmitancje układu z regulatorem R2 .....	11
3.5.	Modele stanowe układów regulacji .....	12
3.6.	Stabilność układów sterowania .....	12
4.	Wyniki symulacji .....	13
4.1.	Charakterystyki czasowe układów regulacji .....	13
4.2.	Charakterystyki częstotliwościowe układów regulacji .....	14
4.3.	Charakterystyki czasowe UR – język strukturalny C++ .....	15
5.	Wnioski .....	18
	Przykładowy wydruk programu <i>PID_NUMERYCZNY.m</i> .....	19
	Przykładowy wydruk programu <i>pid_numeryczny_c++.cbp</i> .....	23

## 1. Sformułowanie problemu

Celem realizowanego projektu jest wyznaczenie parametrów regulatora  $k_1$  oraz  $k_2$  w wersji dyskretniej, który zapewni kompensację stałej czasowej obiektu regulacji  $T_0$ , w układzie sterowania silnikiem DC przedstawionym na rys. 1.



Rys. 1. Uproszczony schemat strukturalny napędu DC

Do wyznaczenia w sposób analityczny parametrów regulatora przyjęto następujące dane:

$$k_0 = 4.5$$

$$T_0 = 0.05 \text{ [s]}$$

### 1.1. Założenia

Projektowany układ regulacji powinien spełniać następujące warunki jakościowe:

- ustalenie się wartości wyjściowej dla czasu regulacji:  $tr_1 = 60 \cdot Tp$  i  $tr_2 = 30 \cdot Tp$ ,
- spełnienia warunku aperiodyczności procesu przejściowego w układzie (przeregulowanie  $p = 0\%$ ),
- spełnienia warunku stabilności w układzie,
- mechaniczna stała czasowa silnika ma być dużo większa od elektrycznej stałej czasowej ( $T_m \gg T_e$ )

## 1.2. Rozwiązanie zadania

Poprawne rozwiązanie zadania zakłada wyznaczenie:

- dyskretnej transmitancji regulatora  $G(z)$
- modelu matematycznego układu regulacji
- modelu numerycznego układu regulacji
- charakterystyk czasowych i częstotliwościowych układu regulacji

## 2. Modele matematyczne układu regulacji

Z treści zadania można wywnioskować, że w układzie regulacji zastosowano regulator PI. Jest on złożeniem członu proporcjonalnego (P) i całkującego (I). Regulator znajduje się w części dyskretnego układu. Transmitancja operatorowa w dziedzinie ciągłej idealnego regulatora PI ma postać:

$$G_{PI} = K_p \left( 1 + \frac{1}{T_i s} \right) \quad (2.1)$$

gdzie:

$k$  – wzmacnienie

$T_I$  – stała zdwojenia

W zadaniu rozpatrywany jest rzeczywisty regulator PI którego ogólna transmitancja operatorowa ma następującą postać:

$$G_R(s) = k_1 + \frac{k_2}{s} \quad (2.2)$$

### 2.1. Dyskretyzacja części ciągłej układu sterowania

Dyskretyzacja części ciągłej układu polega na wyznaczeniu dyskretnego transmitancji obiektu regulacji. Do dyskretyzacji modelu obiektu wykorzystania zostanie metoda ZOH oraz transmitancja operatorowa obiektu (2.3).

$$G_o(s) = \frac{k_0}{T_0 s + 1} \quad (2.3)$$

Okres próbkowania  $T_p$  dobrano na podstawie kryterium o dominującej stałej czasowej.

$$T_p \leq 0,1 T_0 \quad (2.4)$$

$$T_p \leq 0,1 \cdot 0,05 \text{ [s]} \quad (2.5)$$

Przyjmuje się:

$$T_p = 0,004 \text{ [s]} \quad (2.6)$$

Analityczne przebieg dyskretyzacji części ciągłej układu sterowania przedstawia równanie (2.7)

$$\begin{aligned}
 G_O(z) &= G_E(z) * G_O(z) = (1 - z^{-1}) * Z\{h(s)\} = (1 - z^{-1}) * Z\left\{\frac{G_0(s)}{s}\right\} = \\
 &= (1 - z^{-1}) * Z\left\{k_0 * \frac{1}{s(T_0 s + 1)}\right\} = (1 - z^{-1}) * Z\left\{k_0 * \frac{T_0}{T_0} * \frac{\frac{1}{T_0}}{s\left(s + \frac{1}{T_0}\right)}\right\} = \\
 &= \frac{(z-1)}{z} * k_0 * \frac{z * \left(1 - e^{-\frac{T_p}{T_0}}\right)}{(z-1) \left(z - e^{-\frac{T_p}{T_0}}\right)} = k_0 * \frac{\left(1 - e^{-\frac{T_p}{T_0}}\right)}{z - e^{-\frac{T_p}{T_0}}}
 \end{aligned} \tag{2.7}$$

## 2.2. Transmitancja zastępcza regulatora

Wyznaczanie transmitancji zastępczej regulatora przedstawia równanie (2.9).

$$G_R(z) = \frac{k_1 * T_p * z}{z-1} + k_2 = \frac{z(k_1 * T_p + k_2) - k_2}{z-1} = \frac{a * z + b}{(z-1)} \tag{2.9}$$

gdzie:

$$a = (k_1 * T_p + k_2) \tag{2.10}$$

$$b = -k_2 \tag{2.11}$$

## 2.3. Transmitancja układu otwartego

Na podstawie transmitancji dyskretnej części ciągłej (2.8) oraz transmitancji zastępczej regulatora (2.9) transmitancja układu otwartego ma postać:

$$G_{OT}(z) = G_R(z) * G_O(z) = k_0 * \frac{\left(1 - e^{-\frac{T_p}{T_0}}\right)}{z - e^{-\frac{T_p}{T_0}}} * \frac{z(k_1 * T_p + k_2) - k_2}{z-1} = \tag{2.12}$$

$$= \frac{k_0 * \left(1 - e^{-\frac{T_p}{T_0}}\right) * (k_1 * T_p + k_2) \left(z - \frac{k_2}{k_1 * T_p + k_2}\right)}{(z - e^{-\frac{T_p}{T_0}})(z - 1)}$$

#### 2.4. Warunek aperiodyczności procesu przejściowego

Warunek kompensacji dynamiki zostaje spełniony gdy współczynniki mianownika obiektu regulacji są równe współczynnikom licznika regulatora, tzn. sprowadza się to do zniesienia dynamiki w torze sygnałowym obiektu z licznikiem regulatora. Dla spełnienia warunku kompensacji przyjmujemy:

$$\frac{k_2}{k_1 * T_p + k_2} = e^{-\frac{T_p}{T_0}} \quad (2.13)$$

$$\left(z - \frac{k_2}{k_1 * T_p + k_2}\right) = \left(z - e^{-\frac{T_p}{T_0}}\right) \quad (2.14)$$

Uproszczoną postać modelu otwartego układu regulacji przedstawia równanie (2.15).

$$G_{OT}(z) = \frac{k_0 * \left(1 - e^{-\frac{T_p}{T_0}}\right) * (k_1 * T_p + k_2)}{(z - 1)} \quad (2.15)$$

Warunek aperiodyczności jest warunkiem jakościowym regulacji. Zostaje narzucony przez model równania otwartego w postaci:

$$G_{OT}(z) = \frac{1-A}{z-1} \quad (2.16)$$

Transmitancja modelu zamkniętego układu ma zatem postać:

$$G_{MZ}(z) = \frac{G_{OT}(z)}{1+G_{OT}(z)} = \frac{1-A}{z-A} = \frac{Y(z)}{X(z)} \quad (2.17)$$

gdzie:

$$1 - A = k_0 * (k_1 * T_p + k_2) * \left(1 - e^{-\frac{T_p}{T_0}}\right) \quad (2.18)$$

stąd:

$$A = 1 - k_0 * (k_1 * T_p + k_2) * \left(1 - e^{-\frac{T_p}{T_0}}\right) \quad (2.19)$$

Zgodnie z warunkami stabilności na płaszczyźnie  $Z$  układ będzie stabilny jeżeli pierwiastki równania charakterystycznego spełniają warunek:

$$|z| = |A| < 1 \Big|_{M_{MZ}(z)=z-A} \quad (2.20)$$

Wyznaczenie stałej  $A$  zapewnia nam spełnienie następującego warunku:

$$0 < A < 1 \quad (2.21)$$

Wyznaczenie wartości stałej  $A$  pozwoli na znalezienie szukanych parametrów regulatora.

## 2.5. Wyznaczenie wartości stałej $A$

Wyznaczenie stałej  $A$  wymaga przekształcenia transmitancji modelu zamkniętego na równanie różnicowe jednorodne.

Po przekształceniu wyrażenia (2.17), uzyskuje się równanie operatorowe o postaci:

$$Y(z) * (z - A) = 0 \quad (2.22)$$

$$Y(z) * z - Y(z) * A = 0 \quad (2.23)$$

W dziedzinie czasu dyskretnego równanie różnicowe jednorodne ma następującą postać:

$$Y(n - 1) - A * Y(n) = 0 \quad (2.24)$$

Dla powyższego równania różnicowego jednorodnego zakłada się rozwiązanie w postaci:

$$Y(n) = Y_0 * A^n \quad (2.25)$$

Zakładamy że:

$$\text{a) } Y(60) = 0.05Y_0 \quad t_{r1} = 60T_p$$

$$\text{b) } Y(30) = 0.05Y_0 \quad t_{r2} = 30T_p$$



Po przekształceniu równania (2.25) otrzymuje się:

$$A = \sqrt[n]{\frac{Y(n)}{Y_0}} \quad (2.26)$$

Dla założonego czasu regulacji  $t_{r1} = 60T_p$  wartość stałej A wynosi:

$$A_1 = \sqrt[60]{\frac{0,05*Y_0}{Y_0}} = 0,951297 \quad (2.27)$$

Dla założonego czasu regulacji  $t_{r2} = 30T_p$  wartość stałej A wynosi:

$$A_2 = \sqrt[30]{\frac{0,05*Y_0}{Y_0}} = 0,904966 \quad (2.28)$$

## 2.6. Wyznaczenie parametrów regulatora

Wyznaczone wartości stałej A w podpunkcie 2.5 posłużyło do rozwiązania układu równań procesu przejściowego. Korzystając z zależności (2.13) i po przekształceniu równania (2.18) otrzymano następujący układ równań:

$$\begin{cases} \frac{k_2}{k_1*T_p + k_2} = e^{\frac{-T_p}{T_0}} \\ k_0 * (k_1 * T_p + k_2) = \frac{1-A}{1-e^{\frac{-T_p}{T_0}}} \end{cases} \quad (2.29)$$

Przekształcając układ równań otrzymujemy równania do wyznaczenia parametrów regulatora  $k_1$  i  $k_2$ .

$$\begin{cases} k_2 = \frac{e^{\frac{-T_p}{T_0}} * k_1 * T_p}{1 - e^{\frac{-T_p}{T_0}}} \\ k_1 = \frac{\frac{1-A}{1-e^{\frac{-T_p}{T_0}}} - k_0 * k_2}{k_0 * T_p} \end{cases} \quad (2.30)$$

### 3. Modele numeryczne układów regulacji

Do uproszczenia obliczeń posłużono się środowiskiem do wykonywania obliczeń naukowych i inżynierskich - *MATLAB*.

#### 3.1. Przypadek 1 - parametry regulatora R1 dla $tr_1 = 60 \cdot Tp$

Podstawiając poniższe wartości pod zmienne w układzie równań (2.30):

$$k_0 = 4.5$$

$$A_1 = 0,951297$$

$$T_0 = 0.05[s]$$

$$T_p = 0.004[s]$$

otrzymuje się:

$$k_1 = 2.7057 \approx 2.71 \quad (2.31)$$

$$k_2 = 0.1299 \approx 0.13 \quad (2.32)$$

Na podstawie wyznaczonych wzmocnień regulatora  $k_1$  i  $k_2$  przekształconą postać regulatora przedstawia równanie (2.32):

$$G_{R1}(z) = \frac{a \cdot z + b}{(z-1)} = \frac{(k_1 \cdot T_p + k_2)z - k_2}{z-1} = \frac{0.1408z - 0.1299}{z-1} \quad (2.32)$$

#### 3.2. Przypadek 2 - Parametry regulatora R2 dla $tr_2 = 30 \cdot Tp$

Podstawiając poniższe wartości pod zmienne w układzie równań (2.30):

$$k_0 = 4.5$$

$$A_2 = 0.904966$$

$$T_0 = 0.05[s]$$

$$T_p = 0.004[s]$$

otrzymuje się wartości wzmocnień dla regulatora R2:

$$k_1 = 5.2797 \quad (2.33)$$

$$k_2 = 0.2536 \quad (2.34)$$

Na podstawie wyznaczonych wzmocnień regulatora  $k_1$  i  $k_2$  przekształconą postać regulatora przedstawia równanie (2.35):

$$G_{R2}(z) = \frac{a \cdot z + b}{(z-1)} = \frac{(k_1 \cdot T_p + k_2)z - k_2}{z-1} = \frac{0.2747z - 0.2536}{z-1} \quad (2.35)$$

### 3.3. Transmitancje układu z regulatorem R1

Transmitancja modelu układu otwartego dla przypadku pierwszego ma postać:

$$G_{OT1}(z) = \frac{0.0487z - 0.04496}{(z^2 - 1.923z + 0.9231)} \quad (2.36)$$

Transmitancję zamkniętego układu sterowania przedstawiono w równaniu (2.37):

$$G_{MZ1}(z) = \frac{Y(z)}{X(z)} = \frac{G_{OT1}(z)}{1 + G_{OT1}(z)} = \frac{0.0487z - 0.04496}{z^2 - 1.874z + 0.8782} \quad (2.37)$$

Równanie różnicowe dla układu regulacji (2.37) ma następującą postać:

$$y(n) = 1.874y(n-1) - 0.8782y(n-2) + 0.0487x(n-1) - 0.04496x(n-2) \quad (2.38)$$

### 3.4. Transmitancje układu z regulatorem R2

Transmitancja modelu układu otwartego dla przypadku 2 - go ma postać:

$$G_{OT2}(z) = \frac{0.09503z - 0.08773}{(z^2 - 1.923z + 0.9231)} \quad (2.39)$$

Transmitancję zamkniętego układu sterowania przedstawiono w równaniu (2.37):

$$G_{MZ2}(z) = \frac{Y(z)}{X(z)} = \frac{G_{OT2}(z)}{1 + G_{OT2}(z)} = \frac{0.09503z - 0.08773}{z^2 - 1.828z + 0.8354} \quad (2.40)$$

Równanie różnicowe dla układu regulacji (2.3) ma następującą postać:

$$y(n) = 1.828y(n-1) - 0.8354y(n-2) + 0.09503x(n-1) - 0.08773x(n-2) \quad (2.41)$$

### 3.5. Modele stanowe układów regulacji

Opracowane modele w języku C++ i w środowisku *MATLAB* umożliwiają otrzymanie jawnych modeli stanowych układu regulacji dla rozważanych przypadków regulatorów R1 i R2. Poniżej przedstawiono równania stanu dyskretnego układu regulacji z regulatorem R1.

$$x(n+1) = \begin{bmatrix} 1.874 & -0.8782 \\ 1 & 0 \end{bmatrix} x(n) + \begin{bmatrix} 0.25 \\ 0 \end{bmatrix} u(n) \quad (3.1)$$

$$y(n) = [0.1948 \quad -0.1798] x(n) \quad (3.2)$$

Równia stanu (3.3) i (3.4) dyskretnego układu regulacji z regulatorem R2:

$$x(n+1) = \begin{bmatrix} 1.828 & -0.8354 \\ 1 & 0 \end{bmatrix} x(n) + \begin{bmatrix} 0.5 \\ 0 \end{bmatrix} u(n) \quad (3.3)$$

$$y(n) = [0.1901 \quad -0.1755] x(n) \quad (3.4)$$

### 3.6. Stabilność układów sterowania

Pierwiastki transmitancji układu zamkniętego z regulatorem R1 wyznaczono za pomocą funkcji  $P=eig(Gur\_z1)$ :

$$\varphi_1 = 0.9513 \quad (3.5)$$

$$\varphi_2 = 0.9231 \quad (3.6)$$

Pierwiastki transmitancji układu zamkniętego z regulatorem R2 wyznaczono za pomocą funkcji  $P=eig(Gur\_z2)$ :

$$\varphi_1 = 0.9231 \quad (3.7)$$

$$\varphi_2 = 0.9050 \quad (3.8)$$

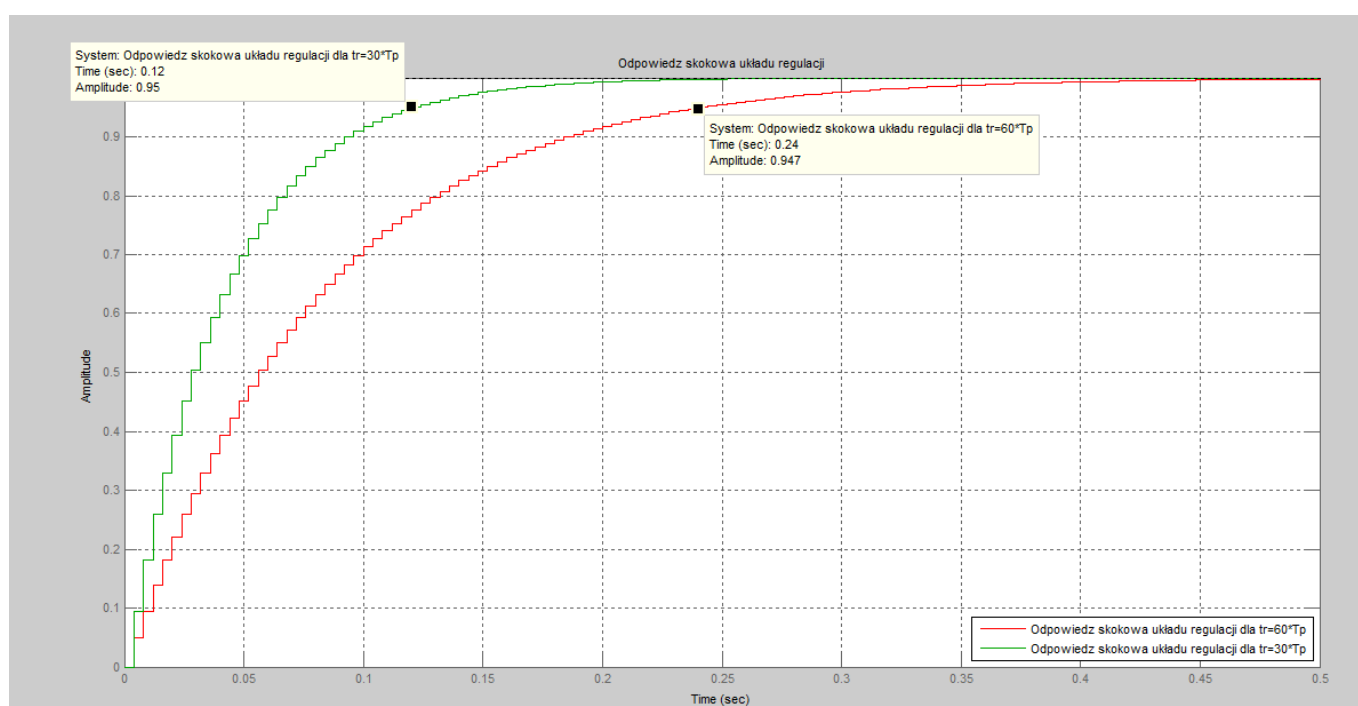
Na podstawie wartości biegunów obu układów regulacji można stwierdzić, że układy te spełniają warunek stabilności ponieważ pierwiastki znajdują się w okręgu jednostkowym.

## 4. Wyniki symulacji

Działanie wyznaczonego układu regulacji zasymulowano w środowisku *MATLAB*. Wyniki symulacji w postaci charakterystyk czasowych i częstotliwościowych dla układów regulacji przedstawiono w podpunktach 4.1 i 4.2.

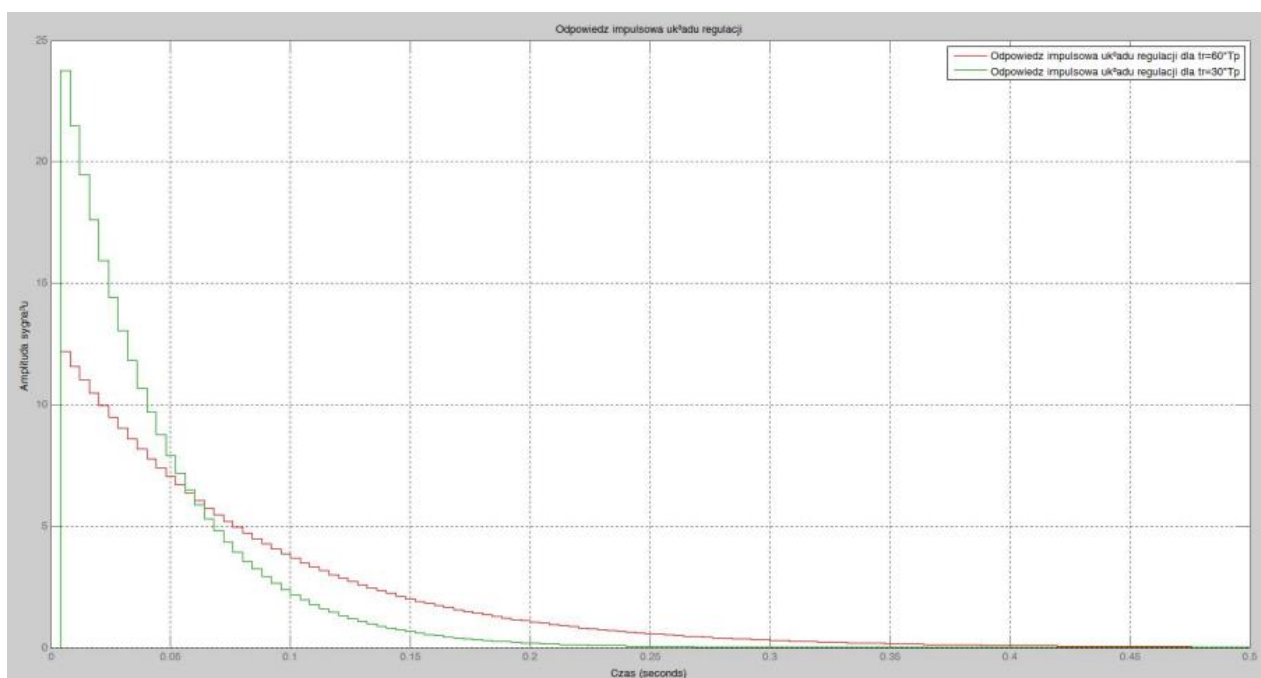
### 4.1. Charakterystyki czasowe układów regulacji

Na rysunku 2 przedstawiono odpowiedzi skokowe układu regulacji, wygenerowane na podstawie m-pliku *PID\_NUMERYCZNY*. Wyniki przedstawiają przebiegi dla czasu regulacji  $tr = 30 \cdot Tp$  oraz  $tr = 60 \cdot Tp$ .



Rys. 2 Odpowiedź skokowa dyskretnych układów regulacji

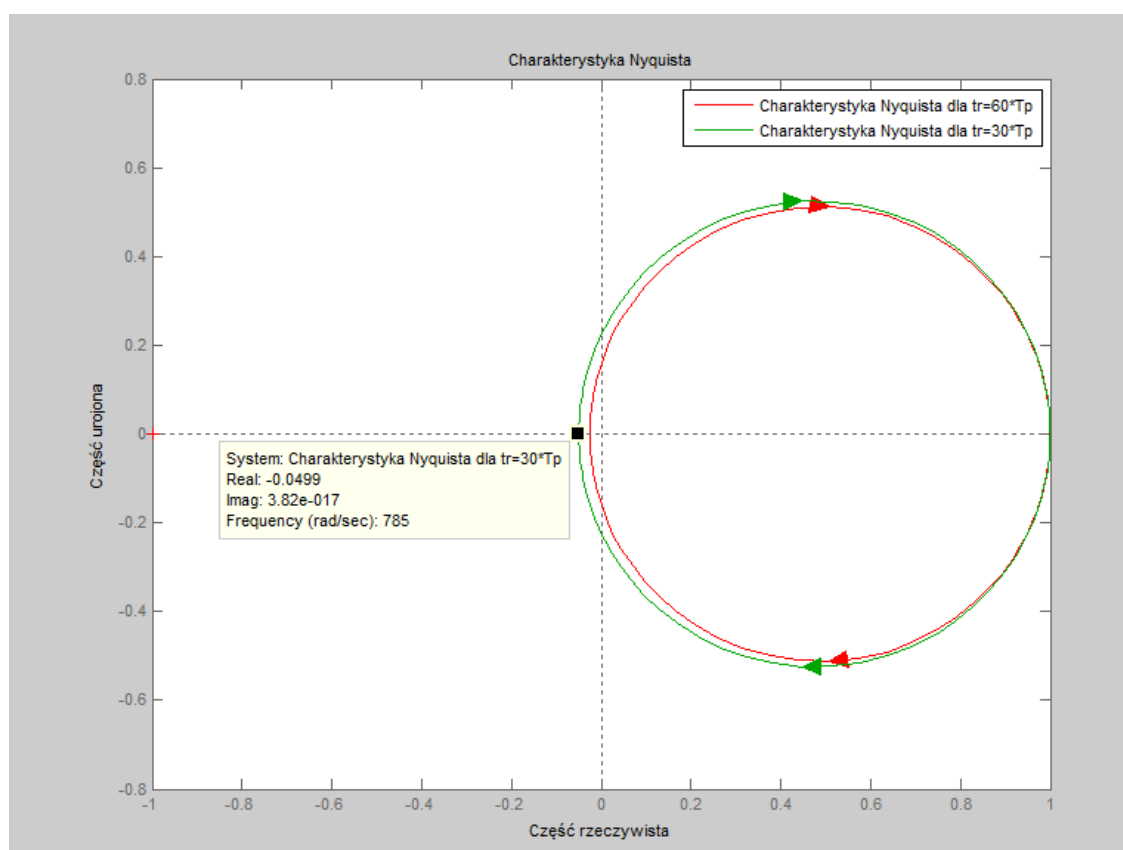
Na rysunku 3 przedstawiono odpowiedzi impulsowe wspomnianych powyżej układów sterowania.



Rys. 3 Odpowiedź impulsowa dyskretnych układów regulacji

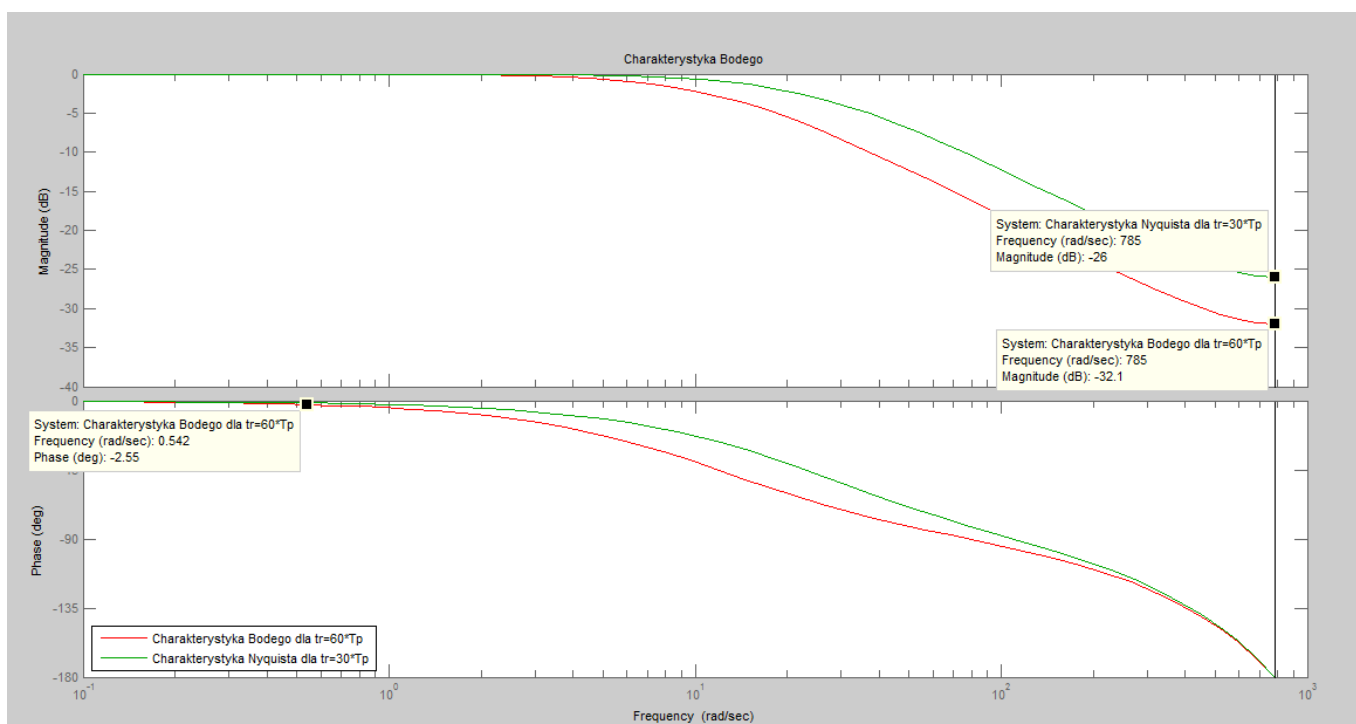
## 4.2. Charakterystyki częstotliwościowe układów regulacji

Działanie układu zbadano na podstawie charakterystyk częstotliwościowych Nyquista i Bodego. Na rysunku 4 przedstawiono charakterystykę Nyquista, z widocznym położeniem biegunów układu regulacji.



Rys. 4. Wykres Nyquista układów sterowania

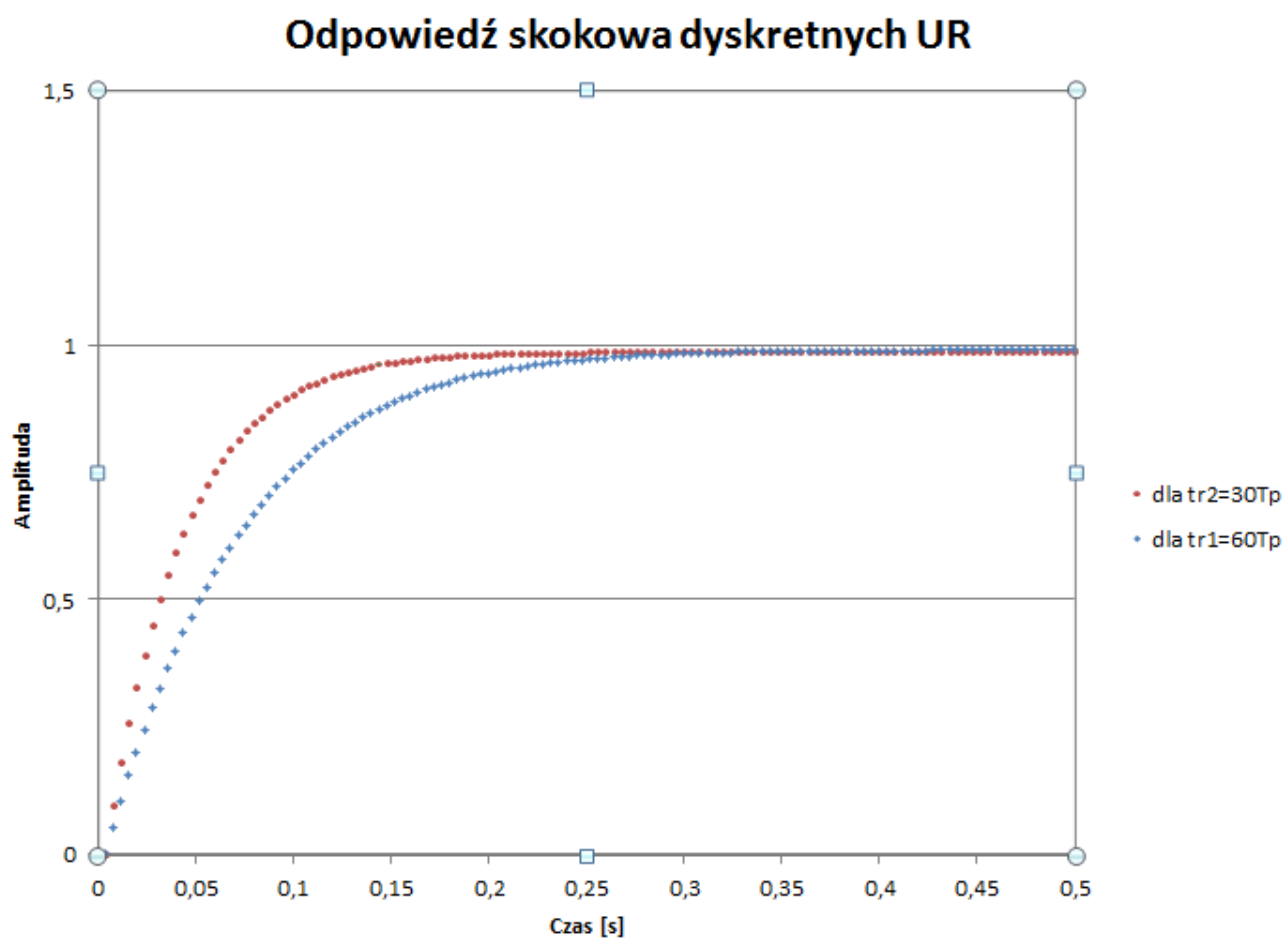
Na rys.6 przedstawiono charakterystykę Bodego dyskretnych układów regulacji.



Rys. 5 Charakterystyka Bodego układów zamkniętych

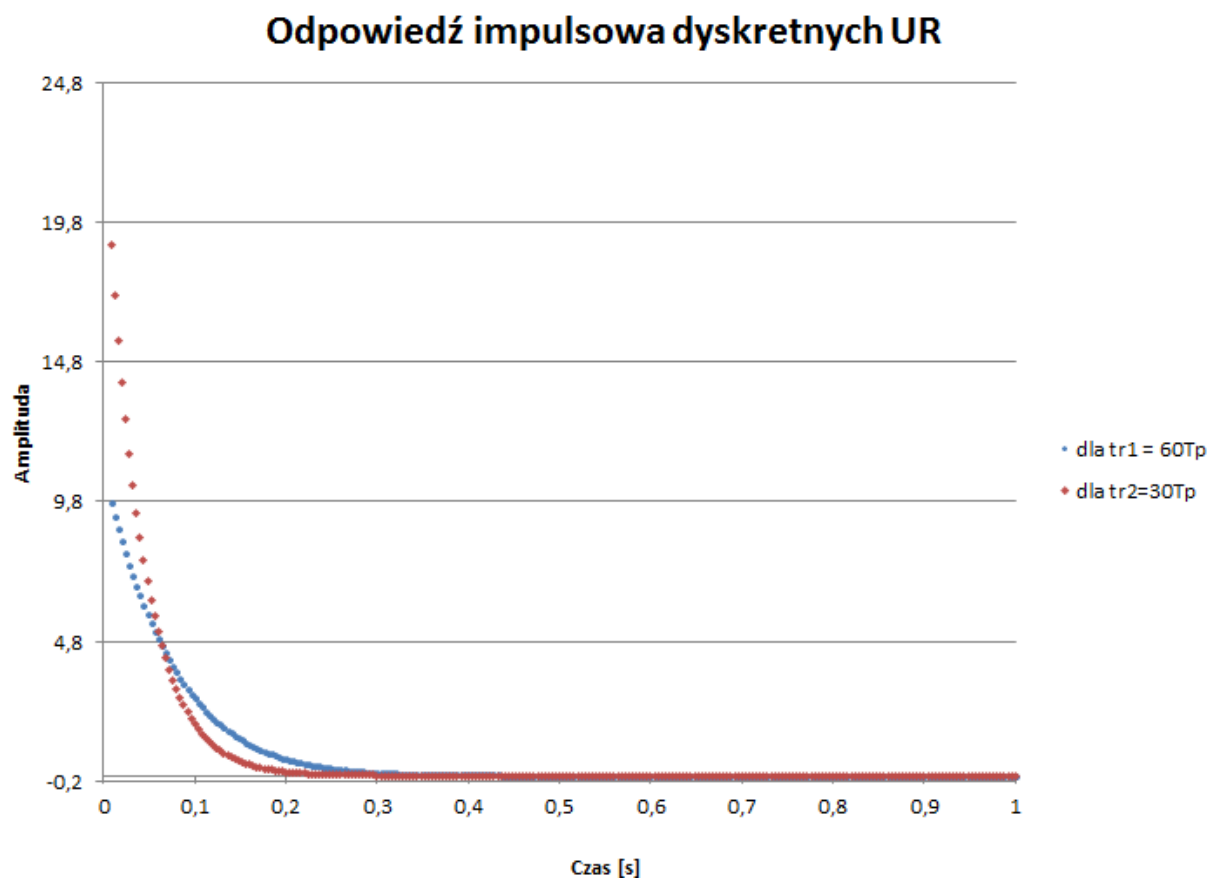
### 4.3. Charakterystyki czasowe UR – język strukturalny C++

Symulację działania układu regulacji przeprowadzono również z wykorzystaniem strukturalnego języka programowania C++ w środowisku *Code::Blocks*. Na podstawie otrzymanych próbek utworzono przebiegi charakterystyk skokowej (rys. 6) i impulsowej (rys.7)



Rys. 6 Odpowiedzi skokowe układów regulacji





*Rys. 7 Odpowiedzi impulsowe układów regulacji*

Powyższe przebiegi wyznaczone zostały na podstawie równań różnicowych układów regulacji.

## 5. Wnioski

Celem zadania było wyznaczenie transmitancji oraz modelu w przestrzeni stanów dyskretnej postaci zamkniętego układu regulacji z regulatorem PI dla warunku kompensacji stałej czasowej obiektu oraz zapewnienie aperiodycznego procesu przejściowego. Zadanie zrealizowano dla dwóch przypadków:

- $t_r = 60T_p$
- $t_r = 30T_p$

Na podstawie otrzymanych wyników można stwierdzić, że:

- Układ spełnia warunek stabilności dla obu przypadków. Świadczą o tym wyznaczone bieguny układów regulacji, które znajdują się wewnątrz okręgu jednostkowego,
- Analizując charakterystyki czasowe: skokową (rys.2) i impulsową (rys. 3) odczytujemy, iż czas regulacji układu z regulatorem R1 wynosi  $t_{r1}=0.24$  [s] przy amplitudzie  $A=0.95$ , zaś wartość na wyjściu układu regulacji z regulator R2 stabilizuje się po  $t_{r2} = 0,12$ [s] osiągając wartość amplitudową  $A=0.95$  - co mieści się w założonym zakresie. Znajomość odpowiedzi impulsowej pozwala nam przewidzieć odpowiedź układu na każde inne wymuszenie. Odpowiedź układu na dowolne pobudzenie jest bowiem splotem sygnału pobudzającego oraz odpowiedzi impulsowej układu. Analiza funkcji odpowiedzi impulsowej pozwala na określenie cech układu m.in czy mamy do czynienia z układem o skończonej, czy nieskończonej odpowiedzi impulsowej. Mówi nam jak szybko jest wstanie powrócić układ do położenia równowagi,
- Analizując charakterystyki częstotliwościowe Bodego (rys. 4) i Nyquist'a (rys. 5) wyznaczono zapas amplitudy  $G_m = 32,1$  dB dla czasu regulacji  $t_{r1}=0.24$  [s] oraz  $G_m = 26$ dB dla czasu regulacji  $t_{r2}=0.12$  [s]. Zapas fazy wynosi  $P_m = 180^\circ$ , dla obu przypadków, który mówi o tym, o ile można zwiększyć fazę układu otwartego bez zmiany jego wzmocnienia, aby układ pozostawał jeszcze stabilnym,
- Na podstawie charakterystyki odpowiedzi skokowej można stwierdzić, że spełniony został warunek aperiodyczności dla wyznaczonych parametrów  $k_1$ ,  $k_2$  i czasu regulacji w obu przypadkach – nie występują przeregulowania ( $p = 0$  [%] ),
- Wyznaczone przebiegi za pomocą transmitancji bądź równań różnicowych w środowiskach *MATLAB* i *Code::Blocks* są ze sobą zgodne,
- Zaprojektowane układy regulacji charakteryzują się bardzo dobrymi parametrami, a mianowicie bardzo małym czasem regulacji oraz zerowym przeregulowaniem.

### Przykładowy wydruk programu *PID\_NUMERYCZNY.m*

Przedstawiono program realizujący projektowanie regulatora metodą analityczną oraz numeryczną w celu porównania otrzymanych wyników. Symulacja zachowania się układu regulacji realizowana jest poprzez wymuszenia: step, impulse, bode, nyquist - podane na wejście układu.

```
%   PROJEKTOWANIE UKŁADÓW REGULACJI

%   Program do wyznaczania dyskretnego regulatora PID na
%   podstawie zadanych
%   parametrów

%   Wykonanie:
%   inż. Violetta Munar Ernandes
%   inż. Piotr Prusaczyk
%   inż. Maciej Dobrzynski

%   Grupa: A4A1S4

clc
clear all

%%%%DANE%%%%%%%%
Tp=0.004;
T0=0.05;
k0=4.5;
A1=0.951297;
A2 = 0.904966;
e=2.71828182846;
s=tf('s');
z=tf('z', Tp);
Go_c=(k0/(T0*s+1)) %transmitancja ciągła obiektu regulacji
pot = (-0.08);

%%% Wyznaczenie wzmocnień K11 i K12 dla czasu regulacji
tr=60*Tp%%%%%%%%

k11L =(1-A1)/(1-e^pot);
k11M = k0*((e^pot*Tp)/(1-e^pot));

k11=k11L/(k0*Tp+k11M)
disp(['Współczynnik regulatora R1 k1 wynosi = ' num2str(k11)])

k12=(( (e^pot)*Tp)/(1-e^pot))*k11
disp(['Współczynnik regulatora R1 k2 wynosi = ' num2str(k12)])

%%% 1. Wyznaczenie wzmocnień K21 i K22 dla czasu regulacji
tr=30*Tp%%%%%%%%

k21L =(1-A2)/(1-e^pot);
```

```

k21M = k0*((e^pot*Tp)/(1-e^pot));

k21=k21L/(k0*Tp+k21M)
disp(['Współczynnik k1 regulatora R2 wynosi = ' num2str(k21)])

k22=((e^pot*Tp)/(1-e^pot))*k21
disp(['Współczynnik k2 regulatora R2 wynosi = ' num2str(k22)])

% współczynniki przekształconej postaci regulatora R1
a1 = k11*Tp+k12;
b1 = - k12;
% współczynniki przekształconej postaci regulatora R2
a2 = k21*Tp+k22;
b2 = - k22;

%%%---Licznik i mianownik obiektu po dyskretyzacji ZOH
wyznaczone analitycznie----%%%%%%%%

GobL=(k0*(1-e^pot)); % licznik obiektu sterowania
GobM = z-(e^pot); % mianownik obiektu sterowania

disp('transmitancja zastępcza otwartego układu sterowania GOT1
wyznaczona analitycznie')

Got1 = (k0*(1-e^pot)*(k11*Tp+k12))/(z-1)

disp('transmitancja dyskretna obiektu regulacji Go(z)
wyznaczona numerycznie')
Go_d=c2d(Go_c,Tp,'zoh')

disp('transmitancja obiektu sterowania Go(z)')
TrOb = GobL/GobM

GrL1=(a1*z+b1); % licznik regulatora R1
GrM1=(z-1); % mianownik regulatora R1

GrL2=(a2*z+b2); % licznik regulatora R2
GrM2=(z-1); % mianownik regulatora R2

%%%%%%%% 2. Wyznaczenie regulatora dla tr=60*Tp %%%%%%%%%
disp('transmitancja regulatora GR1')
GR1 = GrL1/GrM1

%%%%%%%% Wyznaczenie regulatora dla tr=30*Tp %%%%%%%%%
disp('transmitancja regulatora GR2')
GR2 = GrL2/GrM2

%%%%%%%% Transmitancja układu otwartego %%%%%%%%%

disp('Transmitancja dyskretnego układu otwartego dla
tr=60*Tp')
Got_z1 = GR1*TrOb

```

```

disp('Transmitancja dyskretnego układu otwartego dla
tr=30*Tp')
Got_z2 = GR2*TrOb

%%%%%%%%% Transmitancja układu zamkniętego %%%%%%%%%%

disp('Transmitancja układu zamkniętego dla tr=60*Tp')
Gur_z1 = feedback(Got_z1,1,-1)

disp('Transmitancja układu zamkniętego dla tr=30*Tp')
Gur_z2 = feedback(Got_z2,1,-1)

%%%%%%%%%MODEL STANOWY%%%%%%%%%
disp('model stanowy dyskretnego układu regulacji-Gur_z1')
MOD_z1 = ss(Gur_z1) %model stanowy układu regulacji
disp('model stanowy dyskretnego układu regulacji-Gur_z2')
MOD_z2 = ss(Gur_z2)

%%%%%%%%%BIEGUNY UKŁADOW REGULACJI%%%%%%%%%
disp('Bieguny układu zamkniętego dla tr=30*Tp');
P2=eig(Gur_z2)

disp('Bieguny układu zamkniętego dla tr=60*Tp');
P1=eig(Gur_z1)

%%%%%%%%%%%%% WYKRESY %%%%%%%%%%%%%%

%%%%%%%%% ODPOWIEDZ SKOKOWA UR %%%%%%%%%%%%%%

figure(1), step(Gur_z1,'r',Gur_z2,'g'),
%axis([0 120 0 1.1])
%axis([0 12/10 0 1.1])
title('Odpowiedz skokowa układu regulacji');
oznacz = legend('Odpowiedz skokowa układu regulacji dla
tr=60*Tp','Odpowiedz skokowa układu regulacji dla tr=30*Tp');
set(oznacz,'Location','SouthEast')
xlabel('Czas');
ylabel('Amplituda sygnału');
grid;

%%%%%%%%% ODPOWIEDZ IMPULSOWA UR %%%%%%%%%%%%%%

figure(2), impulse(Gur_z1,'r',Gur_z2,'g'),
title('Odpowiedz impulsowa układu regulacji');
oznacz2=legend('Odpowiedz impulsowa układu regulacji dla
tr=60*Tp','Odpowiedz impulsowa układu regulacji dla
tr=30*Tp');
set(oznacz2,'Location','NorthEast')
xlabel('Czas');
ylabel('Amplituda sygnału');
grid;

```

```
%%%%%%%% Ch-ka Nyquista UR %%%%%%%%%%
```

```
figure(3),nyquist(Gur_z1,'r',Gur_z2,'g'),
title('Charakterystyka Nyquista');
oznacz3 = legend('Charakterystyka Nyquista dla
tr=60*Tp','Charakterystyka Nyquista dla tr=30*Tp');
set(oznacz3,'Location','NorthEast')
xlabel('Część rzeczywista');
ylabel('Część urojona');
grid off;
```

```
%%%%%%%% Ch-ka Bodego UR %%%%%%%%%%
```

```
figure(4)
bode(Gur_z1,'r',Gur_z2,'g');
axis([0 120 -180 0])
title('Charakterystyka Bodego');
oznacz3 = legend('Charakterystyka Bodego dla
tr=60*Tp','Charakterystyka Nyquista dla tr=30*Tp');
set(oznacz3,'Location','SouthWest')
grid off;
```

### Przykładowy wydruk programu *pid\_numeryczny\_c++.cbp*

```

1. #include <stdio.h>
2. #include <string.h>
3. #include <iostream>
4. #include <ostream>
5. #include <fstream>
6. #define Tp 0.004 // okres probkowania
7. #define n 1000 // liczba probek
8. using namespace std;
9.
10. unsigned skok (unsigned arg) { // funkcja do implementacji wymuszen
11.
12.     unsigned wartosc;
13.     if ( arg == 0 )
14.     { // implementacja skoku jednostkowego
15.         wartosc = 0;
16.     }
17.     else
18.     {
19.         wartosc = 1;
20.     }
21.     return wartosc;
22. }
23.
24. int impuls(int i)
25. {
26.     if(i == 1)
27.     {
28.         return 200;
29.     }
30.     else
31.     {
32.         return 0;
33.     }
34. }
35.
36.

```

```

37. int main()
38. {
39.     double y[n];
40.     double x[n];
41.                                     // zadeklarowanie dwóch tablic na probki wejsciowe i wyjsciowe
42.
43. //fstream plik;
44. ofstream plik;
45. plik.open("D:\\MEGA\\Semestr IX\\Pur\\Zadania
    egzaminacyjne\\Zad\\Zad1\\dane_skok_R1.txt",ios::out);
46.
47. //plik.open( "dane.txt", ios::out | ios::trunc );
48. for(int i = 0; i < n; i++)
49. {
50.     x[i] = skok(i);
51. }
52.
53. //regulator R1
54. for(int i = 0; i < n; i++)
55. {
56.
57.     if(i == 0) {
58.         y[i] = 0;
59.     }
60.     else if(i <= 1)
61.     {
62.         y[i]=1.874*y[i-1]+0.0487*x[i-1];
63.     }
64.     else {
65.         y[i]=1.874*y[i-1]-0.8782*y[i-2]+0.0487*x[i-1]-0.04496*x[i-2];
66.     }
67.     // plik << 0.004*i << "," << y[i] << "\n";
68. }
69.
70. //zapis do pliku .txt
71.
72. double t = 0;
73. plik << "T" << "\t" << "X[nTp]" << "\t" << "Y[nTp]" << endl;
74. for ( int i = 0; i < n; i++)
75. {

```



```

76.     plik << t << "\t" << x[i] << "\t" << y[i] << endl;
77.     t += Tp;
78. }
79. plik.close();
80.
81. ofstream plik2;
82. plik2.open("D:\\MEGA\\Semestr IX\\Pur\\Zadania
    egzaminacyjne\\Zad\\Zad1\\dane_skok_R2.txt",ios::out);
83.
84.     //regulator R2
85.
86.     for(int j = 0; j < n; j++)
87.     {
88.         x[j] = skok(j);
89.     }
90.
91.     for(int j = 0; j < n; j++)
92.     {
93.
94.         if(j == 0) {
95.             y[j] = 0;
96.         }
97.         else if(j <= 1)
98.         {
99.             y[j]=1.828*y[j-1]+0.09503*x[j-1];
100.        }
101.        else {
102.            y[j]=1.828*y[j-1]-0.8354*y[j-2]+0.09503*x[j-1]-0.08773*x[j-2];
103.        }
104.        // plik2<< 0.004*i << "," << y[i] << "\n";
105.
106.    }
107.
108.    double t1 = 0;
109.    plik2 << "T" << "\t" << "X[nTp]" << "\t" << "Y[nTp]" << endl;
110.    for ( int i = 0; i < n; i++)
111.    {
112.        plik2 << t1 << "\t" << x[i] << "\t" << y[i] << endl;
113.        t1 += Tp;

```

```

114.         }
115.
116.     plik2.close();
117.
118.
119.
120.     //odpowiedz impulsowa
121.     ofstream plik1_impuls;
122.     plik1_impuls.open("D:\\MEGA\\Semestr IX\\Pur\\Zadania
egzaminacyjne\\Zad\\Zad1\\dane_impuls_R1.txt",ios::out);
123.         for(int i = 0; i < n; i++)
124.         {
125.             x[i] = impuls(i);
126.         }
127.         for(int i = 0; i < n; i++)
128.         {
129.             //regulator R1
130.             if(i == 0) {
131.                 y[i] = 0;
132.             }
133.             else if(i <= 1) {
134.                 y[i]=1.874*y[i-1]+0.0487*x[i-1];
135.             }
136.             else {
137.                 y[i]=1.874*y[i-1]-0.8782*y[i-2]+0.0487*x[i-1]-0.04496*x[i-2];
138.             }
139.             //plik1_impuls << 0.004*i << "," << y[i] << "\\n";
140.         }
141.
142.         double t2 = 0;
143.         plik1_impuls << "T" << "\\t" << "X[nTp]" << "\\t" << "Y[nTp]" << endl;
144.         for ( int i = 0; i < n; i++)
145.         {
146.             plik1_impuls<< t2 << "\\t" << x[i] << "\\t" << y[i] << endl;
147.             t2 += Tp;
148.         }
149.
150.         plik1_impuls.close();
151.
152.     ofstream plik2_impuls;

```

```

153.     plik2_impuls.open("D:\\MEGA\\Semestr IX\\Pur\\Zadania
        egzaminacyjne\\Zad\\Zad1\\dane_impuls_R2.txt",ios::out);
154.
155.         //regulator R2
156.     for(int i = 0; i < n; i++)
157.     {
158.         if(i == 0) {
159.             y[i] = 0;
160.         }
161.         else if(i <= 1) {
162.             y[i]=1.828*y[i-1]+0.09503*x[i-1];
163.         }
164.         else {
165.             y[i]=1.828*y[i-1]-0.8354*y[i-2]+0.09503*x[i-1]-0.08773*x[i-2];
166.         }
167.         // plik2_impuls << 0.004*i << "," << y[i] << "\\n";
168.     }
169.     double t3 = 0;
170.     plik2_impuls << "T" << "\\t" << "X[nTp]" << "\\t" << "Y[nTp]" << endl;
171.     for ( int i = 0; i < n; i++)
172.     {
173.         plik2_impuls << t3 << "\\t" << x[i] << "\\t" << y[i] << endl;
174.         t3 += Tp;
175.     }
176.
177.     plik2_impuls.close();
178.
179.
180.     cout << "\\n Dane zostaly zapisane - spawdz"<<endl;
181.     return 0;
182. }

```