Bartłomiej Bąk 193634

Mikołaj Goździelewski 193263

Data 14.06.2024

Raport z projektu MMM sem.4

1. Treść zadania

Projekt 10. Dany jest układ opisany za pomocą transmitancji:

$$G(s) = \frac{a_3 s^3 + a_2 s^2 + a_1 s + a_0}{b_4 s^4 + b_3 s^3 + b_2 s^2 + b_1 s + b_0}$$

gdzie a_i, b_i to parametry modelu. Należy zaimplementować symulator tego układu umożliwiając uzyskanie odpowiedzi czasowych układu na pobudzenie przynajmniej trzema rodzajami synagłów wejściowych (prostokątny o skończonym czasie trwania, trójkątny, harmoniczny). Symulator powinien umożliwiać zmianę wszystkich parametrów transmitancji oraz sygnałów wejściowych. Należy określać stabilność układu oraz wykreślić charakterystyki częstotliwościowe Bodego (amplitudową i fazową) oraz odpowiedź układu.

Zezwolono nam na założenie, że b4 nie przyjmuje wartości równej 0. Jedynym przypadkiem, gdy b4 może być równe zero jest przypadek, gdy całą transmitancję przedstawimy w postaci 1/1, aby zobaczyć sygnał wejściowy. Program nie pokazuje rzeczywistych wyjść układu, jeżeli transmitancja nie jest 4 stopnia lub nie jest równa 1/1.

2. Analiza kodu

1-24: Deklaracje bibliotek oraz stałych globalnych.
26-47: Struktury wykorzystane w projekcie.
49-93: Funkcja rysuj() odpowiadająca za rysowanie obiektów na ekranie.
95-108: Funkcja klik() wykrywająca czy kliknięcie odbyło się w polu interaktywnym(przycisk, pole tekstowe), jeśli jest to przycisk z sygnałem zmienia się zaznaczony sygnał na nowy, jeśli jest to pole tekstowe zmienia się zaznaczone pole, do którego można wpisywać liczby. Funkcja przeszukuje tablicę zawierającą położenia i wielkości wszystkich pól interaktywnych.
110-131: Funkcja wpisywanie() sprawdza jaki przycisk był kliknięty na klawiaturze, jeśli jest to kropka, cyfra lub minus wpisywany jest nowy znak do pola, jeśli coś innego jest to ignorowane. (minus może być tylko pierwszym znakiem pola, a kropka nie może być pierwszym znakiem pola)
133-172: Funkcja zatwierdz() zmienia wartość w polach tekstowych zapisanych w tablicach charów na floata. Sprawdza, którym znakiem jest ZNAK(kropka), a następnie odpowiednio wymnaża kolejne cyfry liczby w tablicy.
174-182: Funkcja usun() usuwa znak w zaznaczonym polu tekstowym.
184-221: Funkcja uzupelnijosie() uzupełnia wartości przy osiach odpowiedzi układu (maksymalną wartość x oraz maksymalną i minimalną wartość y)
223-329: Funkcja sinusoida() oblicza i rysuje odpowiedź na sygnał sinusoidalny.
331-438: Funkcja prostokat() oblicza i rysuje odpowiedź na sygnał prostokątny.
440-576: Funkcja trojkat() oblicza i rysuje odpowiedź na sygnał trójkątny.
578-588: Funkcja potegajeden() oblicza daną potęgę jednostki zespolonej.

590-638: Funkcja bodefun() oblicza i rysuje wykres amplitudowy i fazowy Bodego, korzystając z bibliotek zawierających liczby zespolone oraz operacje na nich. Funkcja rysuje wykres w skali logarytmicznej odpowiednio manipulując wartością x.
f – potęga liczby 10, która jest częstotliwością, na której zaczyna się wykres
x = 10^(i / 100 + f)
Powyższy wzór zmienia wartość x, która jest podstawiana w trakcie obliczeń tak, aby wykres był w skali logarytmicznej, wyświetlającej 4 przedziały potęgi liczby 10 (np. od 0,1 do 1000)
640-661: Funkcja stabilność() sprawdza czy układ ma stabilną transmitancję
664-1495: Główna część programu.
667-1376: Deklarowanie zmiennych i zapisywanie wartości początkowych programu takich jak transmitancja, częstotliwość, od której mają się zacząć wyświetlać wykresy Bodego, amplituda sygnału wejściowego, drugi parametr sygnału wejściowego…
737-738: Deklaracja zmiennej „obecny_sygnal" odpowiadającej za to, który rodzaj sygnału jest aktualnie wybrany.
740-741: Deklaracja zmiennej „obecne_pole" odpowiadającej za to, które pole tekstowe jest aktualnie wybrane.
782-867: Deklarowanie tablicy zmiennych „przyciski" odpowiadających za położenie i wielkość przycisków oraz pól tekstowych. Przypisanie położenia i wielkości poszczególnych pól. Struktura „wazneobszary" zawiera położenie, wielkość oraz wskaźnik do zmiennej z wartościami pola.
876-929: Deklaracje zmiennych zawierających informacje o wartościach poszczególnych zmiennych i miejscu wyświetlania ich na ekranie.
933-1326,1390-1407: Deklaracje elementów, które będą wyświetlane na ekranie takich jak prostokąty czy napisy oraz ustawianie ich wartości takich jak kolor, położenie czy czcionka.
1331-1387: Deklaracja zmiennej „ekran", która zawiera wszystkie stałe elementy ekranu i przypisanie do niej odpowiednich zmiennych.
1416-1492: Pętla wyświetlająca ekran.
1425-1441: Warunek reagujący na naciśnięcie przycisku lewego myszy. Wywołuje on funkcję klik(), zmienia kolor obramowania sygnału, jeżeli nastąpiła jego zmiana oraz rysuje na nowo ekran.
1442-1476: Warunek reagujący na naciśnięcie przycisku. Jeśli naciśnięty przycisk jest enterem, wywołuje on funkcję zatwierdz(), wpisującej w odpowiednie zmienne nowe wartości, wybraną wcześniej funkcję odpowiedzi sygnału, funkcję bodefun(), odpowiadającą za obliczenie i narysowanie wykresów Bodego, stabilnosc(), sprawdzającą stabilność układu, uzupełnia osie wykresów oraz rysuje na nowo ekran. Jeśli naciśnięty przycisk to backspace, wywołuje on funkcję usun(), usuwającą ostatnią wartość wybranego pola tekstowego oraz rysuję na nowo ekran.
1478-1481: Warunek wpisujący do zmiennej „nowy" znak, który został naciśnięty.
1483-1486: Warunek reagujący na inną wartość zmiennej „nowy" niż NULL. Wywołuje wtedy funkcję wpisywanie(), wpisującą nawy znak do wybranego pola tekstowego oraz rysuje na nowo ekran.

3. Obliczenia

a) Dla funkcji liczących wyjście układu o zadanej transmitancji i zadanym sygnale (funkcje sinusoida, prostokat i trojkat, linijki 223 – 536).

Najpierw zainicjalizowano tablicę próbek sygnału wejściowego u[ ] , sygnału wyjściowego y[ ] i macierz x[4] = {0,0,0,0} zawierającą każdą następną pochodną sygnału wejściowego.

Następnie utworzono funkcje, które próbkują sygnał zadany.

Dla sinusa:

u[i] = M * sin (f * 2 * PI * i * h);
h – krok próbkowania
f – częstotliwość sygnału
M – amplituda sygnału
PI - stała matematyczna w projekcie przyjęta jako równowartość 3.1415926535

Dla prostokąta:

u[i] = M do momentu, gdy i < t / h + 1, dla dalszych próbek u[i] = 0;
t - czas trwania sygnału prostokątnego
h – krok próbkowania

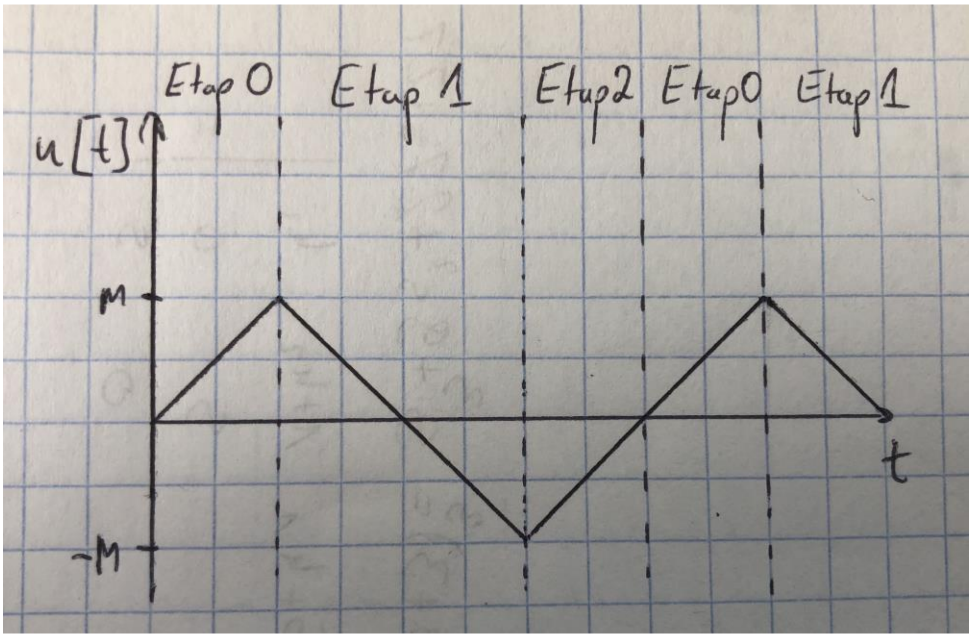
Dla trójkąta:

Sygnał trójkątny podzielono na trzy części powtarzające się okresowo.
Etap 0 – od wartości zerowych do wartości maksymalnej funkcji
Etap 1 – od wartości maksymalnej do minimalnej
Etap 2 – od wartości minimalnej do zera

Dla każdego etapu policzono funkcje nadające wartości sygnałowi wejściowemu

Etap 0: u[i] = j * h * 4 * M / w;
Etap 1: u[i] = (j * h * (-4) * M / w) + 2 * M;
Etap 2: u[i] = u[i] = (j * h * 4 * M / w) - 4 * M;

h – krok próbkowania
w – pulsacja sygnału
M – amplituda sygnału
j - współrzędna załączona w pętli w celu zapętlania etapów



W kolejnym kroku transmitancję w zadanej postaci w treści zadania zamieniono na model stanowy układu w postaci sterowalnej.

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{b_0}{b_4} & -\frac{b_1}{b_4} & -\frac{b_2}{b_4} & -\frac{b_3}{b_4} \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$C = \begin{bmatrix} \frac{a_0}{b_4}, & \frac{a_1}{b_4}, & \frac{a_2}{b_4}, & \frac{a_3}{b_4} \end{bmatrix} \qquad D = \begin{bmatrix} 0 \end{bmatrix}$$

Zainicjalizowano macierze i ich wartości początkowe, które będą wykorzystywane w obliczeniach wartości próbek wyjściowych:
Ax (1x4) = {0,0,0,0}
Bu (1x4) = {0,0,0,0}
Cx (1x1) = {0}

xi (1x4) = {0,0,0,0}

Utworzono funkcję, która liczy wartości próbek wyjściowych w takich krokach:

- Ax = A * x
- Bu = B * u[i]
- Cx = c*x
- xi = Ax + Bu
- xi = xi * h
- xi = xi + x
- x = xi
- y[i] = Cx

Funkcja ta trwa tak długo, dopóki nie skończą się próbki sygnału wejściowego.

Po nadaniu wartości próbkom wyjściowym następuje skalowanie próbek tak aby zmieściły się w okienku prezentującym funkcję wyjściową. Funkcja ta sprawdza dla której próbki wartość funkcji najbardziej odbiega od wartości 0 po czym skaluje wszystkie próbki z zachowaniem proporcji między nimi.

Na koniec funkcje rysują wykres wartości próbek sygnału wyjściowego.

b) Dla funkcji rysujących charakterystyki Bodego (linijki 538 – 598)

Najpierw zadeklarowano zmienne complex mianownika, licznika oraz poszczególnych elementów transmitancji.

Następnie każda wartość transmitancji przemnożono przez odpowiednią wartość zespoloną, jak na zdjęciu poniżej:

$$G(s)=\frac{a_3 s^3+a_2 s^2+a_1 s+a_0}{b_4 s^4+b_3 s^3+b_2 s^2+b_1 s+b_0}$$

$$G(j\omega)=\frac{a_3 j^3\omega^3+a_2 j^2\omega^2+a_1 j\omega+a_0}{b_4 j^4\omega^4+b_3 j^3\omega^3+b_2 j^2\omega^2+b_1 j\omega+b_0}=\frac{-a_3 j\omega^3-a_2\omega^2+a_1 j\omega+a_0}{b_4\omega^4-b_3 j\omega^3-b_2\omega^2+b_1 j\omega+b_0}$$

Potem zadeklarowano zmienne Image dla wykresu amplitudowego i fazowego, w nich rysowane będą kolejne punkty charakterystyk.

Wartość omegi każdej próbki obliczono w następujący sposób:

$$\omega = 10^{\frac{i}{100}+f}$$

Następnie liczone są mianownik i licznik po wymnożeniu i zsumowaniu wcześniej wyliczonych elementów z odpowiednimi potęgami omegi.

Obliczony licznik jest dzielony przez mianownik co daje wynik transmitancji w danej próbce.

Do uzyskania charakterystyki amplitudowej, wynik trafia do funkcji abs(), która zwraca moduł. Podobnie do uzyskania argumentu zmiennej wynik, trafia on do funkcji arg(). Wartość modułu mnożona jest następująco, aby przedstawić go w skali 3dB:

$$y = 20\log(abs(G(j\omega))$$

Następnie we wcześniej utworzonych zmiennych Image rysowane są kolejne punkty charakterystyk.

Funkcja ma w sobie pętlę liczącą charakterystyki Bodego próbka po próbce.

c) Dla funkcji sprawdzającej stabilność asymptotyczną układu (linijki 600-618)

Stabilność asymptotyczna występuje, gdy równanie charakterystyczne posiada tylko rozwiązania ujemne.
Równanie charakterystyczne rozpatrzono za pomocą kryterium Routha.
Po pierwsze wartości współczynników b_n występujących przy kolejnych potęgach parametru s muszą posiadać wspólny znak, to znaczy, że albo wartości b_n są dodatnie albo ujemne. Gdy to kryterium jest spełnione analizie zostaje poddana tablica Routha.

s^4	b_4	b_2	b_0
s^3	b_3	b_1	
s^2	x	y	
s^1	z		
s^0	q		

Aby układ był stabilny wartości stojące w pierwszej kolumnie tablicy Routha muszą posiadać wspólny znak.
Wyznaczono wartości x i y.

$$x = \frac{-\begin{vmatrix} b_4 & b_2 \\ b_3 & b_1 \end{vmatrix}}{b_3} \qquad y = \frac{-\begin{vmatrix} b_4 & b_0 \\ b_3 & 0 \end{vmatrix}}{b_3}$$

$$x = \frac{-b_4 b_1 + b_3 b_2}{b_3} \qquad y = b_0$$

Układ będzie mógł być stabilny wówczas, gdy x będzie posiadał wspólny znak z wartościami b_n. Mamy zatem dwie możliwości albo x jest dodatni, gdy wartości b_n są dodatnie albo x jest ujemny, gdy wartości b_n są ujemne.

$$b_n > 0 \qquad\qquad b_n < 0$$

$$x > 0 \qquad\qquad x < 0$$

$$\frac{-b_4 b_1 + b_3 b_2}{\underbrace{b_3}_{>0}} > 0 \qquad\qquad \frac{-b_4 b_1 + b_3 b_2}{\underbrace{b_3}_{<0}} < 0$$

$$-b_4 b_1 + b_3 b_2 > 0 \qquad\qquad -b_4 b_1 + b_3 b_2 > 0$$

Oba przypadki prowadzą do tego samego warunku jaki musi być spełniony, aby układ miał szansę być stabilny.

$$-b_4b_1+b_3b_2 > 0 \qquad (1)$$

Znając x i y można wyznaczyć wartość z.

$$z = \frac{-\begin{vmatrix} b_3 & b_1 \\ x & y \end{vmatrix}}{x}$$

$$z = \frac{-b_3 y + x b_1}{x}$$

Sytuacja wygląda tutaj tak samo jak w przypadku wcześniej rozważanego x-a, więc analogicznie działając zapisano kolejny warunek prowadzący do stabilności układu.

$$-b_3b_0+xb_1 > 0 \qquad (2)$$

Do wyznaczenia pozostała wartość q.

$$q = \frac{-\begin{vmatrix} x & y \\ z & 0 \end{vmatrix}}{z} = y$$

$$q = b_0$$

Zatem układ jest stabilny asymptotycznie, gdy jednocześnie są spełnione warunki na wartość x i z (równanie (1) i równanie (2)).
Należy jednak jeszcze rozpatrzyć dwa przypadki, jeden, gdy wartość x lub z będzie równa zero (wartość w pierwszej kolumnie równa zero), drugi, gdy wyzeruje się w tablicy cały jeden wiersz (x i y są równe naraz zero lub z jest równe zero lub q jest równe zero).
Rozpatrzono najpierw przypadek pierwszy.

$$x = 0 \iff x = \lim_{\varepsilon \to 0^+} \varepsilon$$

$$z = \lim_{\varepsilon \to 0^+} \varepsilon$$

$$z = \lim_{\varepsilon \to 0^+} \frac{-b_3 y + \varepsilon b_1}{\varepsilon}$$

$$z = \lim_{\varepsilon \to 0^+} \frac{\overbrace{-b_3 b_0}^{<0}}{\underbrace{\varepsilon}_{>0}} + b_1$$

$$z = -\infty$$

$$z = 0 \iff z = \lim_{\varepsilon \to 0^+} \varepsilon$$

$$q = \lim_{\varepsilon \to 0^+} \frac{-\begin{vmatrix} x & y \\ \varepsilon & 0 \end{vmatrix}}{\varepsilon}$$

$$q = \lim_{\varepsilon \to 0^+} \frac{-(-y\varepsilon)}{\varepsilon}$$

$$q = y$$

Aby układ był w dalszym ciągu stabilny, wartość “z” powinna mieć wspólny znak z wcześniejszymi wartościami w pierwszej kolumnie tablicy, tak jednak się nie dzieje, więc gdy x będzie równy zero układ nie będzie stabilny.
Inaczej dzieje się, gdy z = 0, wtedy q ma ten sam znak co pozostałe wartości w pierwszej kolumnie, więc układ będzie stabilny.
Podczas rozważania drugiego przypadku okazuje się, że nie trzeba tego przypadku rozważać, bo żaden z wierszy nie może być zerowy. Wiersz trzeci nie może być zerowy, bo y równa się b_0, które nie jest równe zero. Przypadek, gdy wiersz czwarty jest wyzerowany został już rozpatrzony powyżej (z = 0), a wiersz piąty nie może być równy zero, gdyż q = b_0, które nie jest równe zero.
Podsumowując wszystkie warunki, układ będzie stabilny, gdy:

- $-b_4b_1+b_3b_2 > 0$
- $-b_3b_0+xb_1 >= 0$

4. Trudności podczas projektu

   Podczas wykonywania projektu największymi trudnościami okazała się matematyka związana z próbkowaniem i przetwarzaniem sygnałów nad którą poświęciliśmy dużo czasu. Jedną z trudności okazało się także rysowanie funkcji w bibliotece graficznej, którą wybraliśmy do projektu. Wykresy częstotliwościowe Bodego nie muszą być linią ciągłą (jeżeli dla jakiejś wartości dążą do nieskończoności), co przeszkodziło nam w prezentacji tych charakterystyk za pomocą linii, przedstawiliśmy te funkcje za pomocą punktów.

5. Wnioski

   Program można wykorzystać do określenia parametrów układu widocznych na wykresie wyjścia układu jak i charakterystykach częstotliwościowych. Program można wykorzystać także, jeżeli nam zależy na sprawdzeniu stabilności układu liniowego o zadanej transmitancji 4 stopnia. Program może się także przydać do nauki podstaw automatyki, aby porównać wyniki otrzymane przez własne obliczenia do rzeczywistych wyników, które pokazuje program.

   Podczas tworzenia projektu nauczyliśmy się próbkować sygnały rzeczywiste, wykorzystując krok próbkowania, amplitudę i częstotliwość sygnałów. Nauczyliśmy się wyprowadzać numerycznie pochodne funkcji. Nauczyliśmy się wyświetlać wykresy funkcji z wykorzystaniem biblioteki graficznej niezawierającej możliwości rysowania funkcji. Nauczyliśmy się możliwości wprowadzania danych przez użytkownika, nie przez konsolę a rzeczywisty program. Powtórzyliśmy zasady sprawdzania stabilności układu i obliczania charakterystyk Bodego.

   Program można by jeszcze rozwinąć wprowadzając funkcje odpowiadające za wskazywanie na wykresie sygnału wyjściowego wartości charakterystycznych dla układów stabilnych takich jak maksymalne przeregulowanie czy czas ustalania. Można by też wprowadzić okienka, które umożliwiałyby także bezpośredni odczyt tych wartości z interfejsu. Dla wykresów częstotliwościowych można by wprowadzić rysowanie asymptot w miejscach w których funkcje dążą do nieskończoności.