

Sallen-Key Low-pass Filter

Informatyka

Dokumentacja projektu

Arkadiusz Ziółkowski

21.01.2017

Spis treści

1	Założenia projektowe	3
2	Ogólny mechanizm działania	3
3	Opis funkcji dodatkowych	3
4	Najtrudniejsze elementy w realizacji programu	4
5	Najciekawsze elementy programu	5
6	Obsługa programu	5
7	Format danych	5
8	Możliwy rozwój projektu	5

1 Założenia projektowe

1. Obliczanie wartości dwóch elementów pasywnych (rezystancji rezystora i pojemności kondensatora) filtru dolnoprzepustowego o strukturze Sallen-Key'a na podstawie podanych przez użytkownika dwóch wybranych wartości elementów pasywnych (jednego rezystora i jednego kondensatora) oraz częstotliwości granicznej.
2. Rysowanie charakterystyki amplitudowo-fazowej dla filtru o wyliczonych parametrach.
3. Możliwość wczytania wielu konfiguracji filtru z pliku .xls, wyliczenia brakujących wartości (pojemności kondensatora i oporu rezystora) i zapisania obliczonych konfiguracji w nowym pliku .xls.

2 Ogólny mechanizm działania

Mechanizm działania programu opiera się o przepływ danych pomiędzy interfejsem użytkownika stanowiącym widok aplikacji a jednostką obliczeniową. Wartości wprowadzane przez użytkownika są wstępnie walidowane względem typu wprowadzonych danych przez moduł walidacji danych a następnie zostają przekazane do jednostki obliczeniowej. Jednostka obliczeniowa implementuje algorytm pozwalający wyznaczyć wartości elementów pasywnych filtru po wykonaniu, którego przekazuje obliczone wartości do widoku w celu prezentacji ich użytkownikowi w postaci liczbowej oraz graficznej (charakterystyki Bodego).

Możliwe jest również skorzystanie z opcji pracowania na plikach .xls gdzie zastępowany jest fragment pobierania danych od użytkownika i prezentacji ich użytkownikowi poprzez moduł wczytujący dane z pliku i zapisujący wyniki do pliku.

3 Opis funkcji dodatkowych

1. Klasa "Data Validator" posiadająca funkcje:
 - `retval = IsANumber(src)` - funkcja sprawdza, czy w polu tekstowym reprezentowanym przez uchwyt "src" występuje dodatnia wartość liczbową. W przypadku spełnienia tego warunku zwraca wartość logiczną `true`, w przeciwnym wypadku wartość `false`;

- `retval = IsAText(src)` - funkcja sprawdza, czy w polu tekstowym reprezentowanym przez uchwyt "src" występuje wprowadzony przez użytkownika niepusty ciąg znaków i różny od "insert value". W przypadku spełnienia tych warunków zwraca wartość logiczną true, w przeciwnym wypadku wartość false;
2. Klasa "MSKLPF" reprezentuje model danych przechowujący wartości charakteryzujące filtr:
 - `fc` - częstotliwość graniczna filtru
 - `r1` - wartość rezystancji opornika R1 [kOhm]
 - `r2` - wartość rezystancji opornika R2 [kOhm]
 - `c1` - wartość pojemności kondensatora C1 [uF]
 - `c2` - wartość pojemności kondensatora C2 [uF]
 - `H` - współczynniki transmitancji filtru
 3. Klasa "SKLPF" posiadająca funkcje:
 - `calculatedModel = Calculate(model)` - oblicza wartość elementów filtru na podstawie częściowo uzupełnionego modelu (argument `model`) reprezentowanego przez klasę MSKLPF. Zwraca model uzupełniony o obliczone wartości.
 4. `list = ReadFile(fileName)` - funkcja wczytuje z pliku `fileName` dane filtrów i zwraca wektor modeli (MSKLPF) filtrów.
 5. `function WriteFile(fileName, data)` - funkcja zapisuje do pliku o nazwie `fileName` dane filtrów wyłuskane z modeli znajdujących się w wektorze `data`.

4 Najtrudniejsze elementy w realizacji programu

Najtrudniejszym elementem w realizacji programu okazała się walidacja danych wprowadzanych przez użytkownika, ponieważ należało rozpatrzyć wszystkie możliwości niekompatybilności tych danych z api programu i ograniczeniami fizycznymi (np. niemożność wykonania kondensatora o ujemnej pojemności).

5 Najciekawsze elementy programu

Najciekawszymi elementami programu są klasy MSKLPF, SKLPF, VSKLPF gdyż są one klasami w których model danych został oddzielony od logiki oraz widoku programu.

6 Obsługa programu

7 Format danych

Dane wejściowe programu są liczbami dodatnimi opisującymi filtr:

- f_c - częstotliwość graniczna filtru
- r_1 - wartość rezystancji opornika R_1 [kOhm]
- r_2 - wartość rezystancji opornika R_2 [kOhm]
- c_1 - wartość pojemności kondensatora C_1 [uF]
- c_2 - wartość pojemności kondensatora C_2 [uF]

podczas wprowadzania danych należy pamiętać, że należy podać wartości f_c , oraz po jednej ze zbiorów r_1 , r_2 , c_1 , c_2 . Program posiada zabezpieczenia przez niepoprawnym wprowadzeniu danych.

W przypadku wprowadzania danych z pliku .xls należy utworzyć 5 kolumn, które reprezentują wartości opisujące filtr w kolejności wyżej wymienionej. Wartości elementów do obliczenia muszą w pliku wynosić 0. Plik z danymi wyjściowymi reprezentowany jest w podobnie w pięciu kolumnach jak plik wejściowy z tą różnicą, że występuje wiersz nagłówkowy opisujący kolumny.

8 Możliwy rozwój projektu

1. Dodanie opcji generowania charakterystyk Bodego dla wprowadzonych przez użytkownika kompletnych danych opisujących filtr.
2. Wprowadzenie algorytmu doboru elementów pasywnych filtru spośród wartości należących do istniejących szeregów np. E24.