

Sallen-Key Low-pass Filter

Informatyka

Dokumentacja projektu

Arkadiusz Ziółkowski

22.01.2017

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
6.1	Wprowadzanie danych przez użytkownika	6
6.2	Wprowadzanie danych z pliku	7
7	Format danych	8
8	Możliwy rozwój projektu	9

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

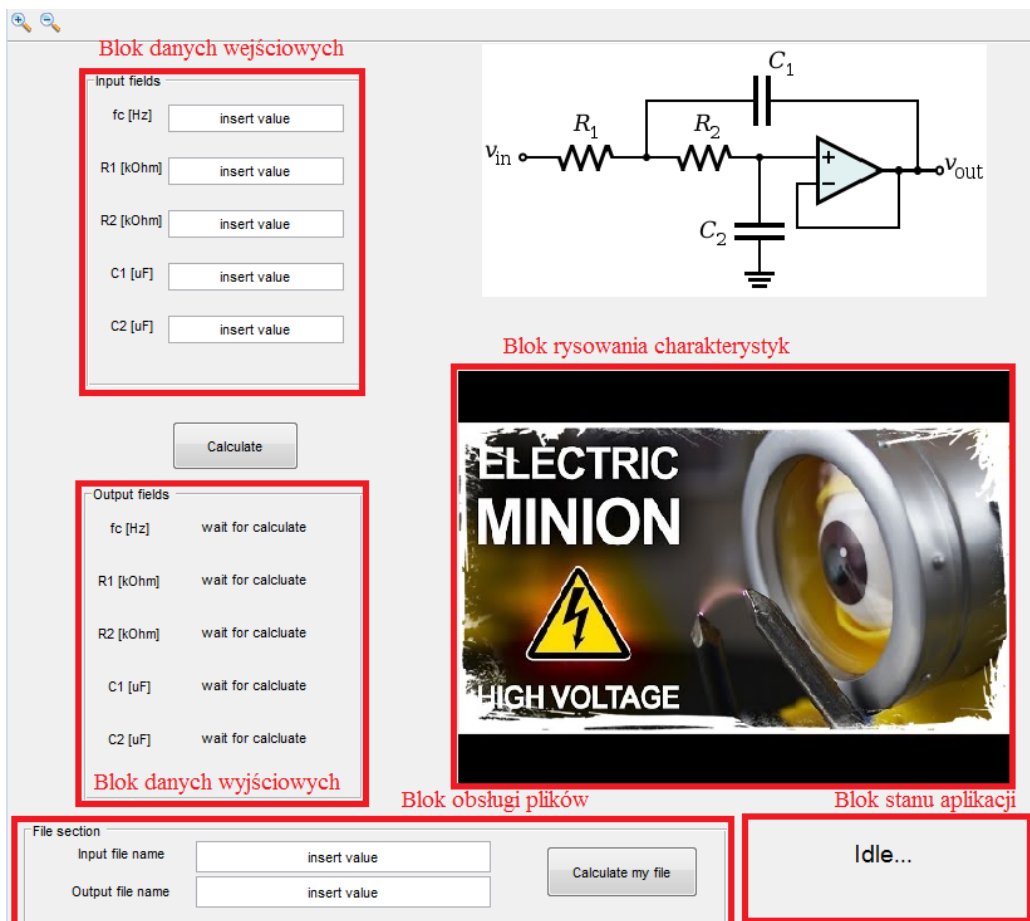
Najtrudniejszym elementem w realizacji programu okazała się walidacja danych wprowadzanych przez użytkownika, ponieważ należało rozpatrzyć wszystkie możliwości niekompatybilności tych danych z api programu i ograniczeniami fizycznymi (np. niemożność wykonania kondensatora o ujemnej pojemności).

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. Ponadto bardzo przydatnym a zarazem użytecznym elementem aplikacji jest blok informujący użytkownika o aktualnym stanie programu (np. bezczynności, przetwarzaniu) widoczny w dolnym, prawym rogu okna aplikacji.

6 Obsługa programu

Na rysunku nr. 1 zostało przedstawione okno aplikacji, na którym opisano najważniejsze bloki (zaznaczone czerwonymi ramkami).



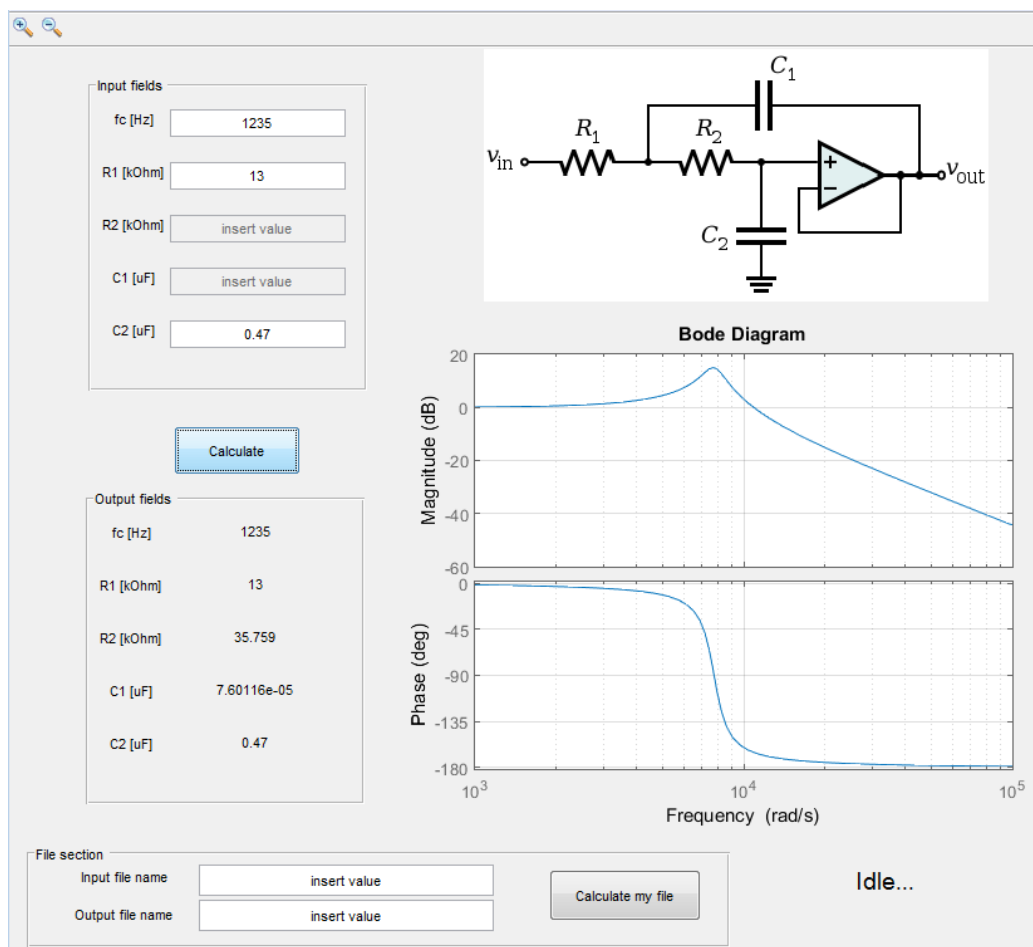
Rysunek 1: Podział okna programu na bloki główne.

6.1 Wprowadzanie danych przez użytkownika

Aby policzyć wartości elementów pasywnych (opór rezystora i pojemność kondensatora) dla pojedynczego układu należy:

1. Wprowadzić w pole f_c bloku danych wejściowych wartość oczekiwanej częstotliwości granicznej filtra wyrażoną w Hz.
2. Wprowadzić w pole R1 albo R2 bloku danych wejściowych wartość rezystancji wybranego rezystora wyrażoną w $k\Omega$. W przypadku wprowadzenia wartości dla jednego z rezystorów pole odpowiadające drugiemu rezystorowi staje się nieaktywne, aby odblokować to pole należy usunąć wartość z uzupełnionego pola.
3. Wprowadzić w pole C1 albo C2 bloku danych wejściowych wartość pojemności wybranego kondensatora wyrażoną w μF . W przypadku wprowadzenia wartości dla jednego z kondensatorów pole odpowiadające drugiemu kondensatorowi staje się nieaktywne, aby odblokować to pole należy usunąć wartość z uzupełnionego pola.
4. Kliknąć przycisk "Calculate" rozpoczynający proces obliczeniowy, po zakończeniu którego wartości wszystkich elementów pasywnych układu zostają wyświetlone w bloku danych wyjściowych, a w bloku rysowania charakterystyk wyświetlone zostają charakterystyki amplitudowo-fazowe dla danego układu.

Na rysunku nr. 2 przedstawiony został widok aplikacji po wykonaniu obliczeń dla danych wprowadzonych przez użytkownika.



Rysunek 2: Widok okna po zakończeniu obliczeń na podstawie danych wprowadzonych przez użytkownika.

6.2 Wprowadzanie danych z pliku

Aby policzyć wartości elementów pasywnych dla większej ilości układów do programu można wprowadzić plik w formacie .xls gdzie pojedynczy układ jest reprezentowany przez pojedynczy wiersz zgodnie z opisem w rozdziale nr. 7. Aby tego dokonać należy:

1. Przygotować plik z danymi wejściowymi zgodnie z opisanym formatem danych i umieścić go w katalogu uruchomieniowym aplikacji.
2. Wprowadzić nazwę pliku z danymi wejściowymi w pole "Input file name" w oknie aplikacji.

3. Wprowadzić nazwę pliku do którego zapisane zostaną dane wyjściowe w pole "Otput file name" w oknie aplikacji.
4. Kliknąć przycisk "Calculate my file" rozpoczynający proces wczytania danych, dokonania obliczeń i zapisania wyników do pliku.

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. Każdy kolejny wiersz traktowany jest jako kolejny układ do obliczenia. Plik z danymi wyjściowymi reprezentowany jest podobnie jak plik wejściowy w pięciu kolumnach z tą różnicą, że występuje wiersz nagłówkowy opisujący kolumny. Wiersze niespełniające powyższych kryteriów będą przez program ignorowane. Jeżeli w pliku wystąpi więcej niż 5 kolumn cały plik zostanie zignorowany. Poprawny format danych wejściowych i wyjściowych w plikach został przedstawiony na rysunku nr. 3.

Dane wejściowe						Dane wyjściowe					
	A	B	C	D	E		A	B	C	D	E
1	1000	10	0	0,47	0	1	f[Hz]	R1[kOhm]	R2[kOhm]	C1[uF]	C2[uF]
2	10000	10	0	0,47	0	2	1000	10	17,519	0,47	0,000308
3						3	10000	10	8,947	0,47	6,02E-06

Rysunek 3: Format danych w pliku wejściowym i wyjściowym.

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.