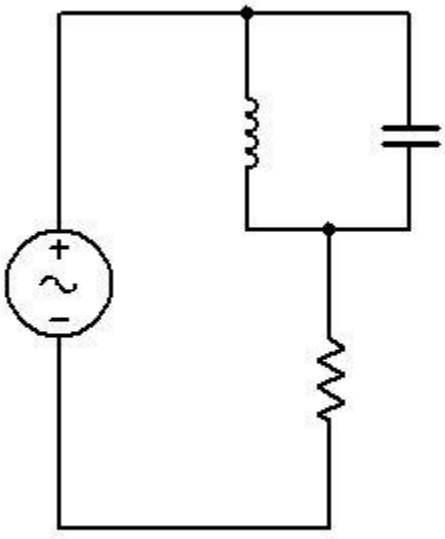
Techniki obliczeniowe
Projekt zima 2019/2020
Sajdak Radosław i Mateusz Kozyra

Tematem naszego projektu, była analiza układów RLC. Ponieważ był to pierwszy, realizowany przez nas projekt, do zadania podeszliśmy z pewną niepewnością, ale również ekscytacją. Postanowiliśmy napisać aplikację. Już na samym początku założyliśmy, że zależy nam na tym, aby aplikacja oferowała jak najwięcej interaktywnych elementów interfejsu. Miało to służyć nie tylko temu, aby sam projekt wypadł dobrze, ale również temu, by użytkownik niemający pojęcia o omawianych tematach czy zjawiskach, mógł ich w pewnym sensie dotknąć, zobaczyć coś, co trudno wytłumaczyć osobie, która nie wie czym jest kondensator, lub cewka. Dla której każde urządzenie elektroniczne po prostu działa, a to, co dzieje się, gdy naciska jego włącznik nie ma najmniejszego znaczenia.

W ten sposób, narodziła się idea. Zróbmy cztery, charakterystyczne układy RLC. Niech użytkownik sam wybierze, który układ go interesuje, niech zobaczy, że różnią się one czymś poza schematem. Wybraliśmy więc układy oparte na elementach RLC:

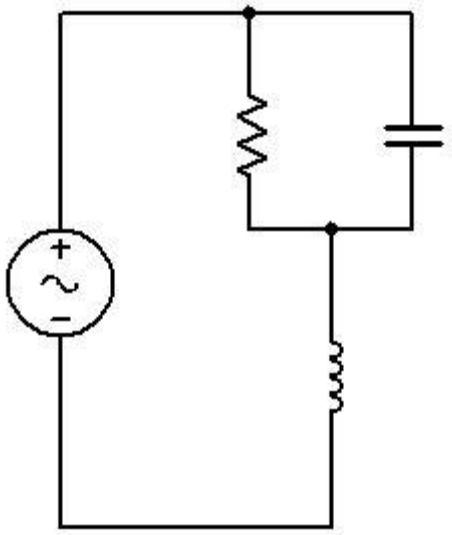
- Równoległe połączenie cewki i kondensatora
- Równoległe połączenie rezystora i kondensatora
- Równoległe połączenie rezystora i cewki
- Szeregowe połączenie wszystkich elementów

Podstawą naszego projektu, miała być analiza częstotliwościowa układów. Stwierdziliśmy jednak, że przy okazji zaimplementujemy też stany nieustalone, by pokazać użytkownikowi, że gdy naciska przycisk on/off dowolnego urządzenia nie od razu wszystko jest tak, jak chce żeby było. Chcieliśmy też, aby użytkownik sam mógł wybrać, który element właśnie obserwuje. Zacznijmy jednak od analizy częstotliwościowej. Naszym pierwszym zadaniem było wyznaczenie wszystkich 12 równań transmitancji. Ponieważ jednak w trzech układach (zawierających połączenia równoległe) ilość równań redukowała się, ostatecznie musieliśmy obliczyć jedynie 9 równań dla 4 układów.



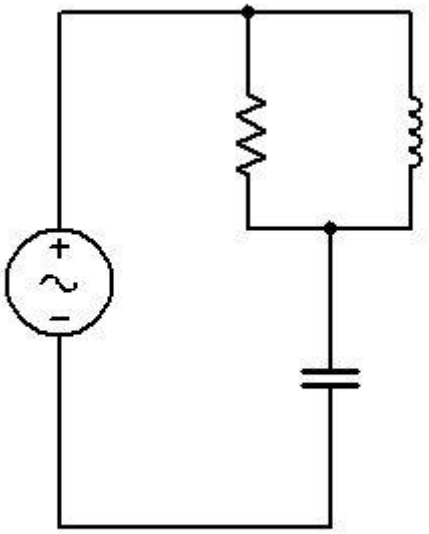
$$H(s) = \frac{Ls}{RLCs^2+Ls+R}$$ –elementy równoległe

$$H(s) = \frac{RLCs^2+R}{RLCs^2+Ls+R}$$ – rezystancja



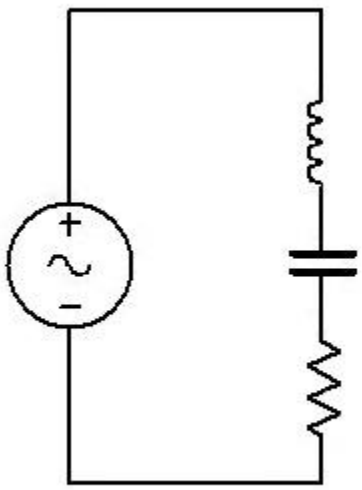
$H(s) = \frac{R}{RLCs^2+Ls+R}$ – elementy równoległe

$H(s) = \frac{RLCs^2+Ls}{RLCs^2+Ls+R}$ – cewka



$H(s) = \frac{Ls+R}{RLC^2+Ls+R}$ – elementy równoległe

$H(s) = \frac{RLCs^2}{RLCs^2+Ls+R}$ – kondensator



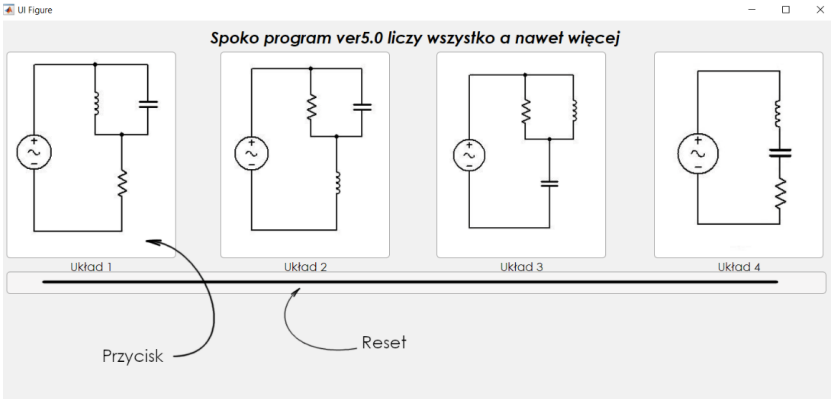
$$H(s) = \frac{LCs^2}{LCs^2+RCs+1}$$ – cewka

$$H(s) = \frac{RC}{LCs^2+RCs+1}$$ – rezystancja

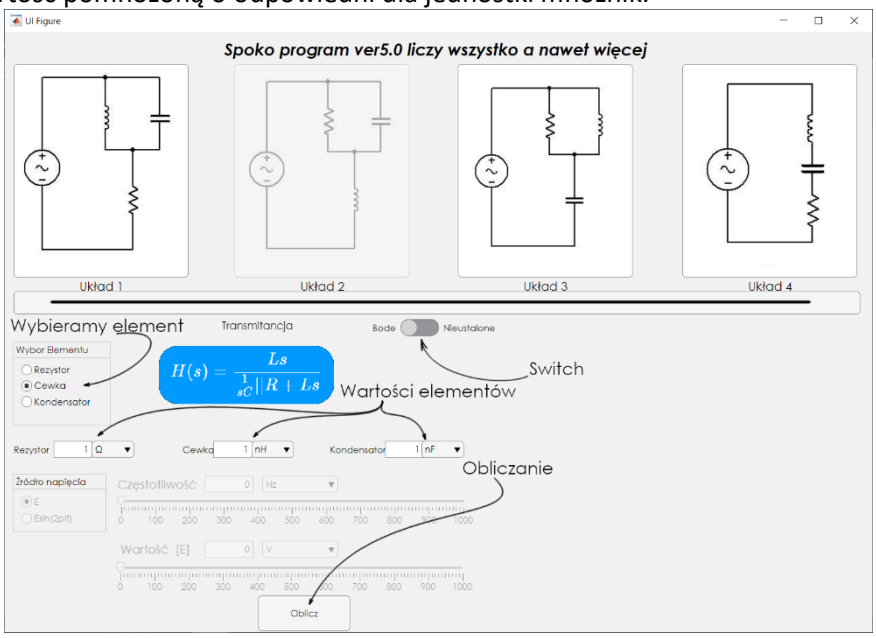
$$H(s) = \frac{1}{LCs^2+RCs+1}$$ – kondensator

Po wykonaniu stosownych obliczeń, podjęliśmy decyzję o wykorzystaniu narzędzia appdesigner, które miało znacznie ułatwić stworzenie nam interaktywnej aplikacji. Ponieważ cała nasza aplikacja oparta jest właśnie o to narzędzie, bezsensownym byłoby opowiadać o jej działaniu bez jej prezentacji. Dlatego dalsza część raportu (wraz z późniejszymi obliczeniami stanów nieustalonych) oparta będzie na analizie i prezentacji aplikacji. Należy tutaj zaznaczyć, że aplikacja ma ustalone na twardo wymiary 1147x796 oraz wyłączone skalowanie – ma to na celu utrzymanie stworzonej przez nas estetyki. Oczywiście da się stworzyć aplikację, która skaluje się do wyświetlacza, jednak nie jest to elementem przedmiotu i kosztowałoby nas wiele cennego czasu. **Do poprawnego działania aplikacji, należy zainstalować Control System Toolbox** (get more apps w matlabie).

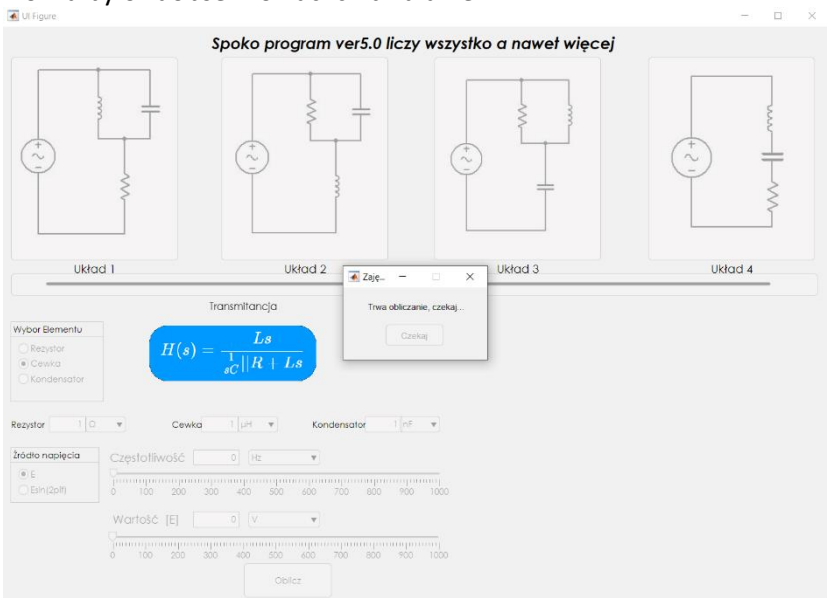
Po uruchomieniu aplikacji, widzimy wyskakujące okno. Każdy układ, jest tutaj przyciskiem. Naciskając go, wybieramy który układ zamierzamy analizować. Poniżej znajduje się przycisk reset – jest on rzadko używany, jednak przydatny podczas prezentacji działania (ułatwił też pisanie kodu). Przywraca on program do stanu „po uruchomieniu”. Po wybraniu układu, wybrany „wyszarza się”, a następnie pokazywane są kolejne opcje.



Gdy już wybierzemy układ, należy za pomocą przełącznika (Switch) ustalić, co będziemy analizować. Wybierać możemy pomiędzy Bode (analiza amplitudowo-częstotliwościowa oraz fazowo częstotliwościowa) lub stanami nieustalonymi. Ponieważ na razie omawiamy analizę częstotliwościową, musimy wybrać element, dla którego transmitancję obliczymy. Ponieważ transmitancje są określone w kodzie „na twardo", po wybraniu elementu transmitancja wyświetla nam się w postaci grafiki. Na tym etapie, transmitancja nie jest jeszcze obliczana. Dopiero, gdy wybierzemy wartości elementów (oczywiście możemy pozostawić je domyślne) i naciśniemy przycisk „Oblicz", wykonywana jest funkcja *oblicz_bode* z argumentami zadeklarowanych w aplikacji zmiennych tj. wartości elementów oraz wybranego układu. Funkcja *oblicz_bode* odnosi się do *oblicz_transmitancje*, aby nie powielać kodu. Dodatkowo, wybierając jednostkę elementu lub zmieniając jego wartość, każdorazowo uruchamiamy funkcję, która do zadeklarowanych zmiennych przypisuje wartość pomnożoną o odpowiedni dla jednostki mnożnik.



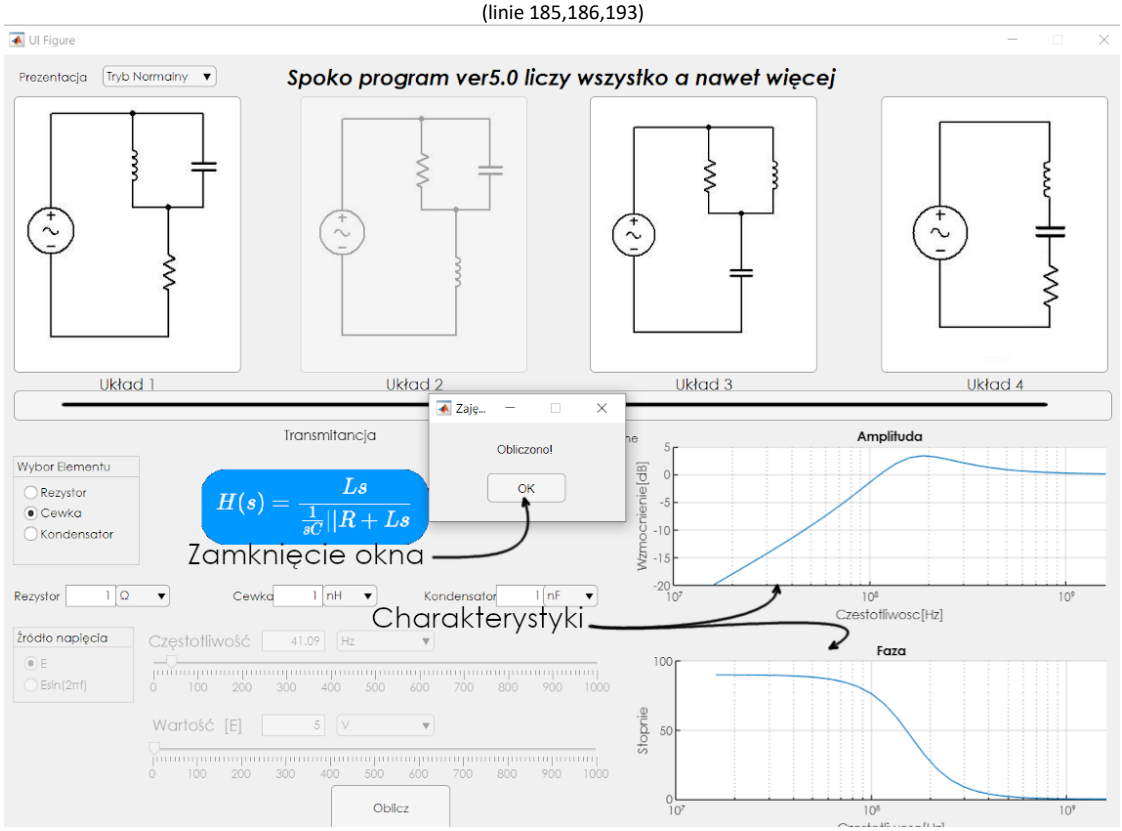
Gdy program wykonuje obliczenia, na ekranie widzimy wyskakujące okienko. Informuje ono o tym, że program jest w trakcie obliczanie oraz uniemożliwia jakiekolwiek interakcje z programem (w tym jego zamknięcie). Ma to uczynić program bardziej „studentoodpornym" a więc uniemożliwić podmianę danych podczas obliczeń i wysypanie programu. Kolejne okienka są sztucznie opóźnione o 1s, aby w ogóle można było zaobserwować ich działanie.



Po wykonaniu wszystkich obliczeń, będziemy mogli zamknąć okienko przyciskiem „OK”. Po prawej stronie, zobaczymy również charakterystyki: amplitudowo częstotliwościową oraz fazowo częstotliwościową. Na poziomie kodu, ich stworzenie wykonuje się w funkcji *oblicz_bode*. Do wektora [mag,ph,wout] przypisujemy kolejno amplitudę(tablica 3D), fazę(tablica 3D) oraz częstotliwość(wektor) zwracane przez funkcję Bode, do której podajemy obliczoną wcześniej w postaci symbolu transmitancję. Następnie, tworzymy wykresy w skali półlogarytmicznej i przypisujemy go do odpowiednich rodziców.

$$[mag, ph, wout] = bode(app.HS);$$
$$semilogx\left(app.BodeShowAmpl, wout/(2*pi), 20*log10\left(squeeze(mag)\right)\right)$$
$$semilogx\left(app.BodeShowPhas, wout/(2*pi), squeeze(ph)\right)$$

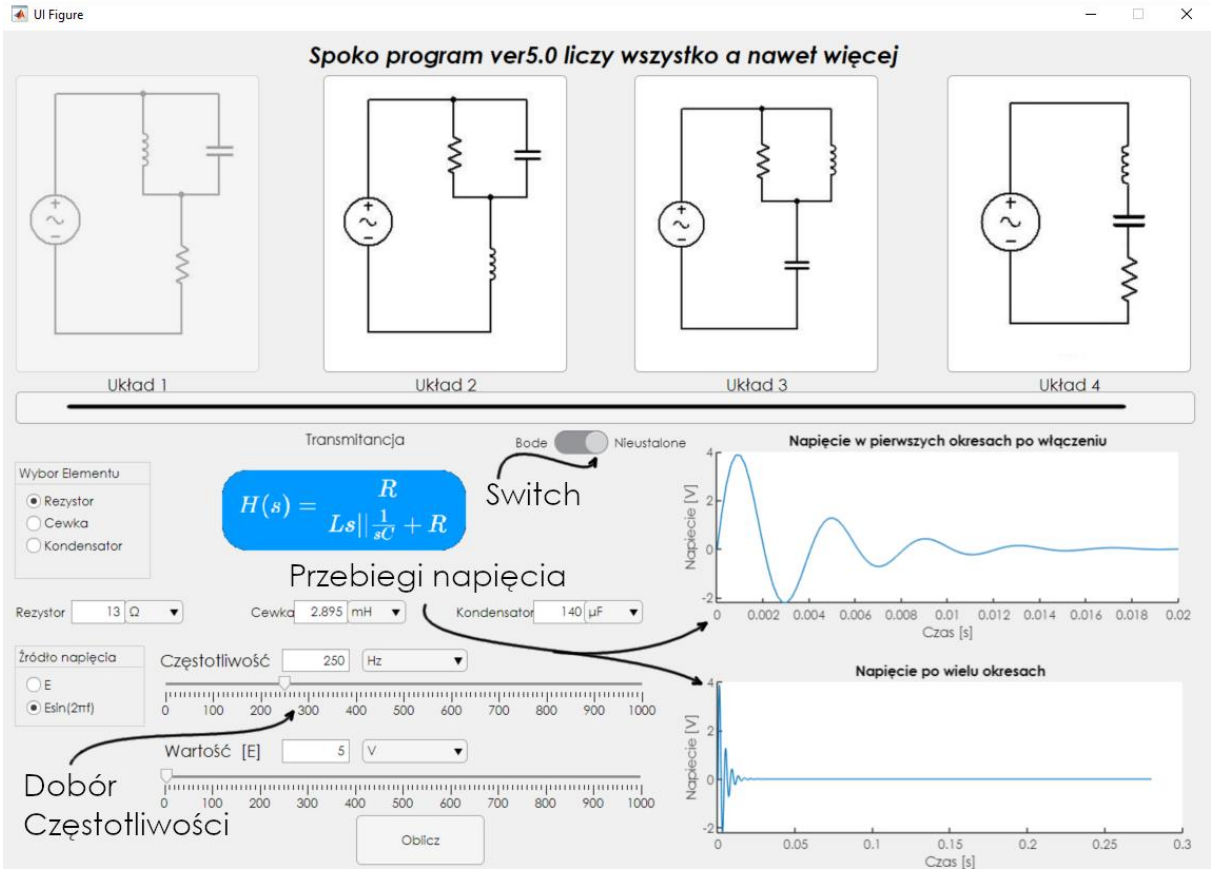
(linie 185,186,193)



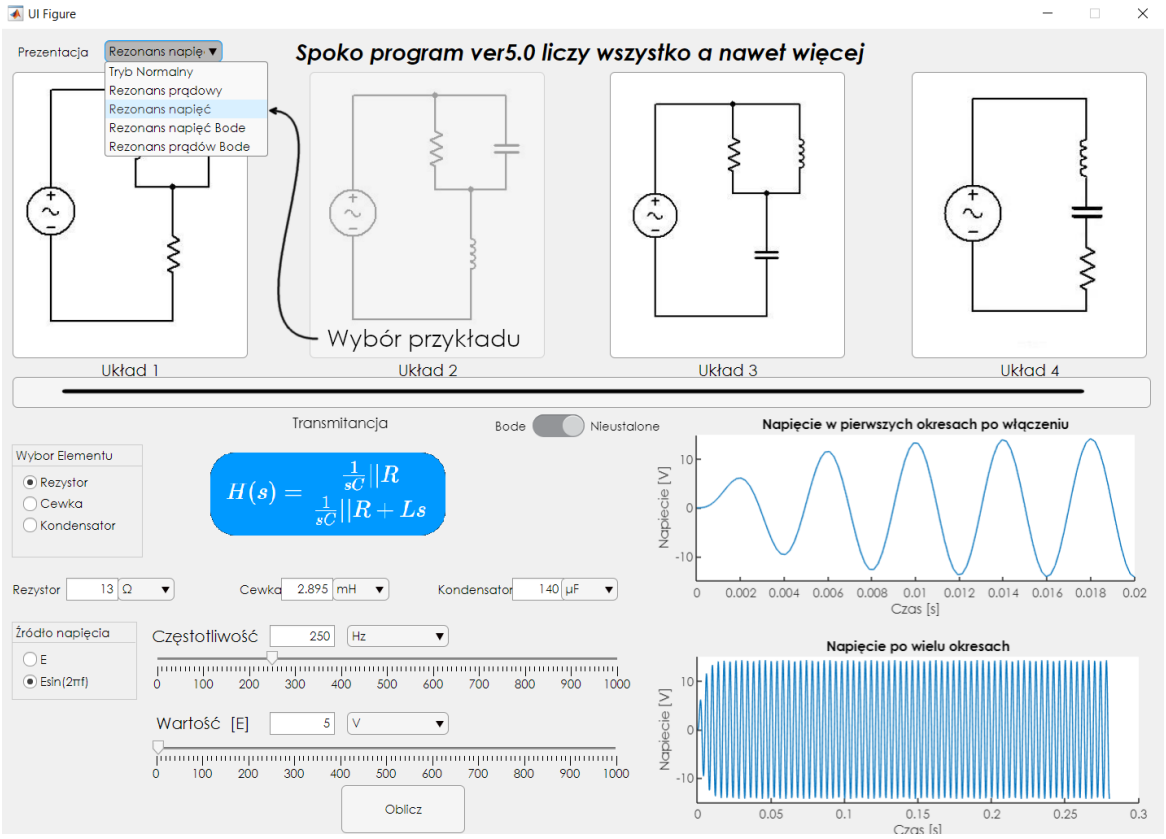
W tym momencie, możemy zmienić wartości oraz element, a cały cykl powtórzy się od początku.

Kolejną funkcjonalnością naszej aplikacji jest wyznaczanie stanów nieustalonych. Aby zająć się nimi, ustawiamy wspomniany wcześniej przełącznik(Switch) w tryb „Nieustalone”. Jeśli wcześniej analizowaliśmy charakterystyki amplitudowo-częstotliwościowe, znikną one z ekranu, a w aplikacji uaktywniona zostanie opcja wyboru źródła napięcia oraz jego wartości przy użyciu Slidera. Z oczywistych przyczyn, częstotliwość pozostaje nieaktywna, gdy wybieramy stałe źródło napięcia.

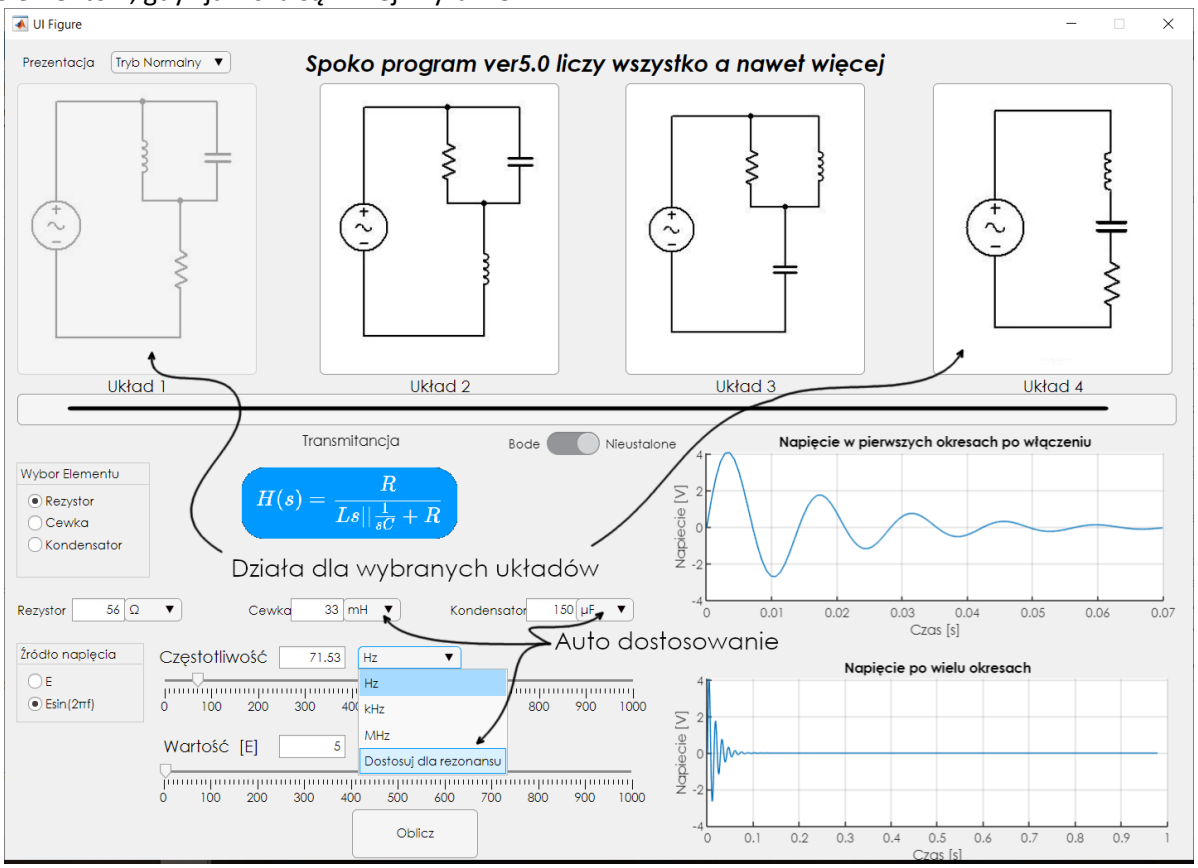
Aby ograniczyć ilość wzorów oraz obliczeń, zastosowaliśmy jednak przybliżenie. Gdy użytkownik wybierze stałe źródło napięcia, częstotliwości przypisywana jest wartość 100MHz (linia 211). Daje to dostatecznie dobre przybliżenie przebiegów. Możemy również jako źródło napięcia wybrać Esin(2πf), co odblokuję możliwość dobrania częstotliwości. Również tutaj, dla jednostek zastosowaliśmy mnożniki. Tym samym zmiana jednostki (lub wartości) wywołuje kolejne funkcje, które przypisują odpowiednią wartość do zadeklarowanej w programie zmiennej. Warto w tym miejscu zaznaczyć, że zwiększanie wartości napięcia źródła oraz jego częstotliwości, znacznie wydłuża obliczenia. Ponownie, wciśnięcie przycisku *„Oblicz”* przywołuje na ekran okienko, blokujące aplikację. Następnie wywoływana jest funkcja *oblicz_nieustalone* do której podawane są wartości rezystancji, indukcyjności, pojemności, częstotliwości, napięcia oraz wybrany obecnie układ. W liniach 226-240 przekształcane są wykresy z półlogarytmicznych na liniowe oraz zmieniane są ich etykiety. Linie 241 i 242 można wykorzystać do porównania tego co zwraca aplikacja z wynikami zwykłego, liniowego *plota.* Wartości elementów na poniższym obrazku, dobrane zostały nieprzypadkowo. Występuje dla nich rezonans prądów pomiędzy elementami LC. W związku z tym, w stanie ustalonym, według teorii obwodów moglibyśmy zastąpić połączenie równoległe przerwą. Jak widać, nasza aplikacja pozwala na obserwowanie rezonansów – mimo napięcia na źródle o wartości 5V, po około 18ms spadek napięcia na rezystorze jest równy 0V, a co za tym idzie, przez rezystor nie płynie prąd.



Dodatkową opcją, która w naturalny sposób nasunęła się podczas tworzenia powyższych funkcji, był wybór kilku przykładów. W końcu nie każdy musi potrafić sam wyznaczyć elementy rezonansowe, prawda? Przykładowe konfiguracje, zostały zrealizowane jako Dropdown w lewym górnym rogu. Wybrać możemy 4 opcje rezonansu, oraz *Tryb Normalny* będący trybem domyślnym. Tryb normalny pozwala na wprowadzenie wszystkich danych „od zera”, jednak wybranie przykładu nie blokuje zmiany danych po wykonaniu obliczeń. Funkcja ta „symuluje” użytkownika. Kolejno wpisuje wartości i wywołuje kolejne callbacki. Zrealizowana jest w liniach 904-1023.



Ostatnim naszym pomysłem, w kwestii rozwoju aplikacji, było umożliwienie użytkownikowi badanie własnych obwodów rezonansowych. Zrealizowaliśmy to, poprzez dodanie dodatkowych funkcji *jestrezonans* oraz *niemarezonansu*. Fukcje te, gdy wybierzemy *Switch -> Nieustalone*, rozszerzają (lub pomniejszają) dropdowny jednostek pojemności, indukcyjności i częstotliwości o opcję *Dostosuj dla rezonansu*. Opcja ta, działa dla *Układu1* oraz *Układu4* , ponieważ są to najbardziej obrazowe przykłady połączenia równoległego i szeregowego. Użytkownik sam może wybrać wartości elementów (lub częstotliwość), a następnie wybierając opcję *Dostosuj dla rezonansu*, dopasować brakujący element układanki w taki sposób, aby otrzymać rezonans. Pozwala to na obserwację rezonansów nie tylko dla wybranych przez nas przykładów, ale również dla dużych wartości elementów, gdy zjawiska są mniej wyraźne.



Podczas tworzenia aplikacji, natrafiliśmy na wiele problemów związanych z obsługą danych czy narzędzia appdesigner. Ostatecznie jednak, aplikacja zdecydowanie spełniła nasze wymagania. Nie tylko prezentuje dane w sposób przejrzysty, ale wprowadza możliwość interakcji użytkownika ze środowiskiem. Niestety zdarza się, że aplikacja zawiesi się na danych, które wcześniej obliczała bezproblemowo. Być może jest to problem systemu lub komputera na którym działa aplikacja. Pomaga wówczas prosty restart środowiska matlab.