

Wykorzystanie karty dźwiękowej do badania układów elektrycznych RL i RC w obwodach prądu przemiennego

Cel ćwiczenia

Celem ćwiczenia jest stworzenie aplikacji, której zadaniem będzie wykorzystanie komputera PC do badania układów elektrycznych RL i RC w obwodach prądu przemiennego. Źródłem, jak i miernikiem napięcia przemiennego, będą dwie karty muzyczne zainstalowane w komputerze pomiarowym. Sterowanie oraz pomiar, przeprowadzane będzie wykorzystując środowisko programistyczne *LabVIEW 7.1*.

1. Wprowadzenie teoretyczne

Prądem przemiennym nazywamy prąd zmieniający w czasie napięcie i natężenie w taki sposób, że ich wartość średnia w czasie jest równa zero. Napięcie i natężenie tego prądu zmieniają się sinusoidalnie w czasie:

$$u = U_0 \cos(\omega t) \quad (1)$$

$$i = I_0 \cos(\omega t + \varphi) \quad (2)$$

gdzie: i – natężenie chwilowe, u – napięcie chwilowe, I_0 – natężenie szczytowe, U_0 – napięcie szczytowe, φ - przesunięcie fazowe, $\omega = 2\pi f = 2\pi/T$ – częstością kołową lub pulsacją, f – częstością, T – okresem.

W obwodach prądu przemiennego rolę oporu spełniają nie tylko zwykłe oporniki, lecz również kondensatory i cewki. Kondensator (o pojemności $C=Q/U$) stanowi przerwę w obwodzie prądu stałego, natomiast przewodzi prąd przemienny. Przewodzenie polega na ładowaniu kondensatora w pierwszym półokresie w jednym kierunku, a w drugim półokresie w kierunku przeciwnym. Cewka, która jest zwojnicą z drutu miedzianego ma znikomo mały opór dla prądu stałego. W cewce włączonej w obwód prądu przemiennego, zgodnie z prawem indukcji Faraday'a indukuje się siła przeciwelektromotoryczna indukcji własnej

$$E_L = -L \frac{di}{dt} \quad (3)$$

gdzie L jest współczynnikiem indukcji własnej. Siła przeciwelektromotoryczna indukcji własnej spowalnia narastanie i zmniejszanie natężenia prądu chwilowego.

Opór jaki stawia prądowi przemiennemu odbiornik zawierający pojemność elektryczną i indukcję własną nazywa się **zawadą** lub **impedancją** i wyraża się wzorem

$$Z = \sqrt{R^2 + (X_L - X_C)^2} \quad (4)$$

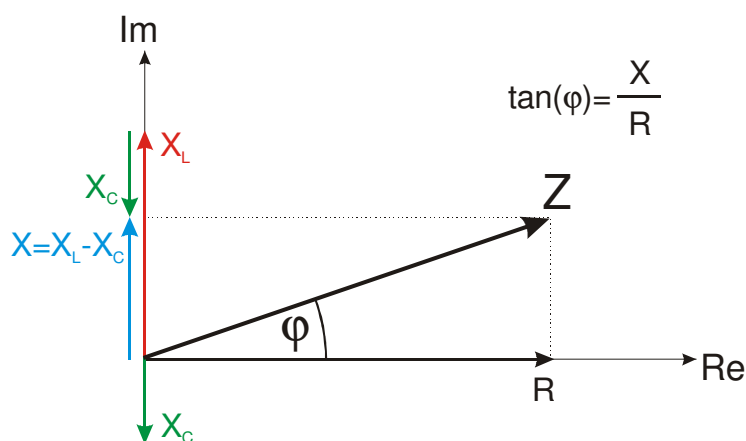
gdzie

$$X_L = \omega L \quad (5)$$

$$X_C = \frac{1}{\omega C} \quad (6)$$

nazywane są odpowiednio **reaktancją indukcyjną i pojemnościową**. Wielkości te określają opór jaki stawiają elementy indukcyjne i pojemnościowe w czasie przepływu prądu przemiennego o określonej częstotliwości kołowej ω .

Obecność w obwodzie elementów indukcyjnych lub pojemnościowych powoduje przesunięcie prądu w fazie względem napięcia. Przesunięcie fazowe wyznaczyć można wprowadzając zespoloną reprezentację zawady Z , w której opór „ohmowy” R jest oporem rzeczywistym, natomiast reaktancje indukcyjne i pojemnościowe X są oporami urojonymi (Rys. 1)



Rys. 1.1 Zespolona reprezentacja zawady Z .

Tangens przesunięcia fazowego wyraża się zatem wzorem

$$\tan(\varphi) = -\frac{(X_L - X_C)}{R} \quad (7)$$

Prawo Ohma, wiążące szczytowe wartości (amplitudy) napięcia i prądu, przybiera w tym przypadku postać:

$$U_0 = ZI_0 \quad \text{lub} \quad U_0 = I_0 \sqrt{R^2 + \left(\omega L - \frac{1}{\omega C}\right)^2} \quad (8)$$

W doświadczeniu interesować nas będą obwody, które oprócz oporu R zawierać będą wyłącznie indukcyjność własną (RL) lub wyłącznie pojemność elektryczną (RC).

1.1 Układ RL

W przypadku indukcji prąd wyprzedza w fazie napięcie, a wzory na impedancję układu oraz na przesunięcie fazowe przyjmują postać:

$$Z = \sqrt{R^2 + (\omega L)^2} \quad (9)$$

$$\tan(\varphi) = -\frac{\omega L}{R} \quad (10)$$

Prawo Ohma dla powyższego układu wyraża się następująco:

$$I = \frac{U}{Z}, \quad I = \frac{U}{\sqrt{R^2 + \omega L^2}}, \quad R = R_p + R_L, \quad (11)$$

gdzie R_p jest wartością rezystancji opornika pomiarowego, natomiast R_L jest rezystancją cewki. Spadek napięcia mierzony na oporniku pomiarowym wynosi:

$$U_R = IR_p, \quad (12)$$

a więc

$$U_R = \frac{U}{\sqrt{(R_p + R_L)^2 + \omega L^2}} R_p. \quad (13)$$

Wykonując proste przekształcenie powyższego równania otrzymać można zależność na kwadrat stosunku napięcia zasilającego do spadku napięcia na oporniku pomiarowym w postaci:

$$\left(\frac{U}{U_R}\right)^2 = \left(\frac{L}{R_p}\right)^2 \omega^2 + \frac{(R_p + R_L)^2}{R_p^2} \quad (14)$$

Równania (10) oraz (14) pozwalają na wyznaczenie wartości indukcyjności cewki L , jeżeli znane są zależności częstotliwościowe przesunięcia fazowego $\tan(\varphi)$ oraz stosunku napięć U/U_R .

1.2 Układ RC

Dla obwodów zawierających jedynie pojemność elektryczną, zawadę i przesunięcie fazowe wyznaczyć można z następujących relacji:

$$Z = \sqrt{R^2 + \left(\frac{1}{\omega C}\right)^2} \quad (15)$$

$$\tan(\varphi) = \frac{1}{\omega RC} \quad (16)$$

Prawo Ohma dla układu RC wyraża się następująco:

$$I = \frac{U}{Z}, \quad I = \frac{U}{\sqrt{R^2 + \left(\frac{1}{\omega C}\right)^2}}, \quad R \equiv R_p \quad (17)$$

W powyższym przypadku całkowita rezystancja układu określona jest przez wartość opornika pomiarowego R_p , a spadek napięcia mierzony na oporniku pomiarowym wynosi w tym przypadku:

$$U_R = IR_p, \quad U_R = \frac{U}{\sqrt{R_p^2 + \left(\frac{1}{\omega C}\right)^2}} R_p \quad (18)$$

Proste przekształcenie powyższego równania prowadzi do wyrażenia na kwadrat stosunku napięcia zasilającego do spadku napięcia na oporniku pomiarowym:

$$\left(\frac{U}{U_R}\right)^2 - 1 = \frac{1}{(R_p C)^2} \frac{1}{\omega^2} \quad (19)$$

Podobnie jak we wcześniej rozpatrywanym układzie RL, tak i w przypadku układu RC wykorzystanie równań (16) i (19) pozwala na wyznaczenie wartości pojemności kondensatora C, jeżeli znane są zależności częstotliwościowe przesunięcia fazowego $\tan(\varphi)$ oraz stosunku napięć U/U_R .

2. Opis układu pomiarowego

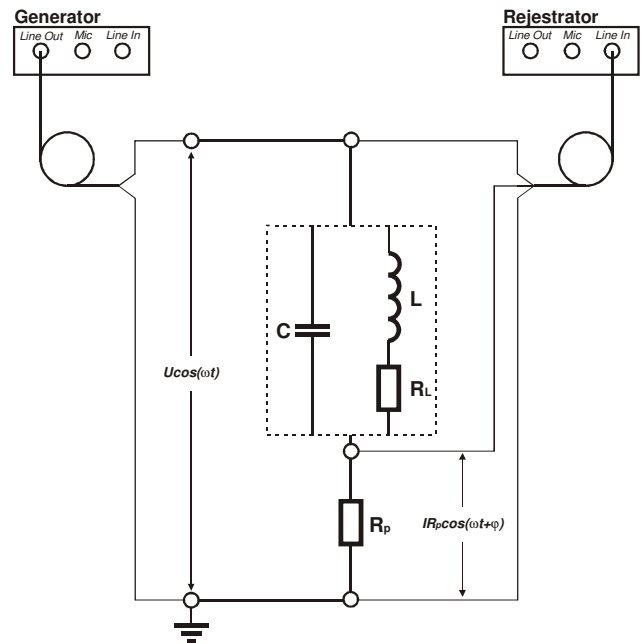
Jak wynika z powyższego wstępu teoretycznego, wykonanie doświadczenia wymaga wyznaczenia charakterystyk częstotliwościowych tangensa kąta przesunięcia fazowego $\tan(\varphi)$ oraz stosunku napięć U/U_R , w celu ich porównania z zależnościami teoretycznymi danymi równaniami (10), (14) oraz (16), (19). Do wyznaczenia wspomnianych charakterystyk konieczne są w ogólności dwa przyrządy: 1) źródło napięcia przemiennego posiadające możliwość zmiany częstotliwości oraz 2) miernik amplitudy oraz fazy napięcia przemiennego.

Przeprowadzenie ćwiczenia polegać będzie na wykorzystaniu środowiska programistycznego *LabVIEW 7.1* w celu oprogramowania dwóch standardowych kart muzycznych znajdujących się w komputerze PC, w taki sposób, aby jedna z nich pełniła rolę źródła napięcia, natomiast druga była miernikiem. Stworzona aplikacja powinna mieć możliwość sterowania pomiarem, akwizycji danych oraz ich wstępnej analizy.

Układ pomiarowy składa się z połączonych w szereg: opornika pomiarowego R_p , oraz w zależności od badanego układu, cewki L lub kondensatora C . Opornik pomiarowy służy do pomiaru spadku napięcia, które zgodnie z równaniem (12) proporcjonalne jest do płynącego w układzie prądu (rys.2.1).

Źródłem napięcia przemiennego doprowadzanego do badanego układu jest jeden z dwóch kanałów *Line Out* pierwszej karty muzycznej. Oba kanały *Line In* drugiej karty wykorzystane są w celu rejestracji napięcia na zaciskach źródła oraz spadku napięcia na oporniku pomiarowym.

Pamiętać należy, aby wejście liniowe rejestrującej karty muzycznej (*Line-In*) było jedynym źródłem rejestrowanego sygnału. Odpowiednich ustawień można dokonać w *Panelu Sterownia* systemu operacyjnego w ustawieniach *dźwięków i urządzeń audio* (*Mixer*).



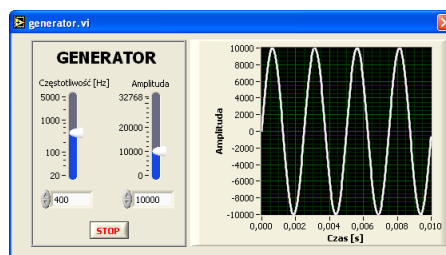
Rys. 2.1 Schemat połączeń

3. Generacja i rejestracja sygnałów sterowana z poziomu LabVIEW

Przed rozpoczęciem tworzenia aplikacji służącej do sterowania parametrami sygnału, należy ustalić która z dwóch kart muzycznych zainstalowanych w komputerze pomiarowym posłuży jako generator, a która jako rejestrator sygnału. W poniższym opisie przyjęto, że generować sygnał będzie karta 0 (występująca jako pierwsza w systemie operacyjnym), natomiast rejestratorem sygnału będzie karta 1 (druga w systemie). W celu poprawnego skonfigurowania każdego urządzenia pomiarowego niezbędna jest znajomość jego parametrów. W przypadku opisywanego ćwiczenia urządzeniami pomiarowymi wykorzystywanymi zarówno do generacji jak i do rejestracji sygnałów są karty muzyczne. Każda standardowa karta muzyczna posiada dwa przetworniki cyfrowo-analogowe (po jednym na każdy kanał), które mogą być wykorzystane do generacji sygnału, oraz dwa przetworniki analogowo-cyfrowe służące do rejestracji sygnałów. Parametry generowanego jak i rejestrowanego sygnału zdefiniowane są parametrami wspomnianych dwóch par przetworników. W przypadku standardowej karty muzycznej sygnał próbkowany jest z częstotliwością 44,1kHz i dokładnością 16 bitów. Są to parametry pozwalające z zadowalającą dokładnością wykonać pomiary charakterystyk częstotliwościowych w zakresie od 10 – 10000Hz.

3.1 Generator elektrycznego sygnału sinusoidalnego

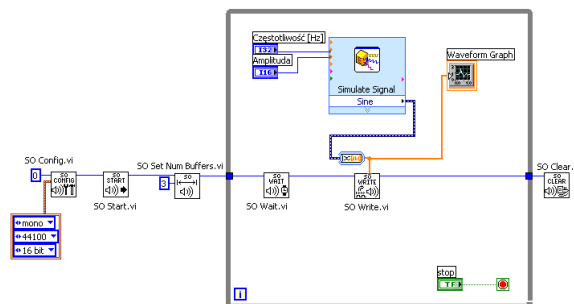
Tworzenie generatora sygnału rozpoczniemy od *Panelu Frontowego* nowostworzonego VI'a o nazwie *Generator.vi*, na którym umieszczone zostaną kontrolki (np. suwaki) służące do zmiany wartości parametrów sygnału, opisane odpowiednio jako *Częstotliwość* i *Amplituda*. Dodatkowo umieszczony zostanie przycisk STOP służący do zatrzymania programu oraz wskaźnik graficzny *Waveform Graph* przedstawiający generowany sygnał w postaci wykresu (rys. 3.1).



Rys. 3.1 Panel frontowy generatora sygnału sinusoidalnego *Generator.vi*.

Wszystkie funkcje wymagane do zapisu generowanego sygnału na wyjście karty muzycznej, dostępne są z poziomu *Diagramu Blokowego* środowiska *LabVIEW 7.1* w *Fuctions/Graphic and Sounds/Sound/Sound Output*.

Programowanie generatora sygnału wykorzystującego kartę muzyczną odbywać się będzie w trzech etapach: 1) Konfiguracja sprzętu, 2) generacja funkcji i jej zapis na wyjście urządzenia generującego, 3) zakończenie obsługi sprzętu.



Rys. 3.2 Diagram blokowy generatora *Generator.vi*.

3.1.1 Konfiguracja sprzętu

W pierwszej kolejności należy skonfigurować wyjście karty muzycznej mającej generować sygnał. W tym celu skorzystamy z funkcji *SO config.vi*. Parametrami wejściowymi tej funkcji jest indeks urządzenia oraz format dźwięku będący klastrem z parametrami wyjściowego sygnału: prędkość próbkowania, ilość bitów na próbkę, jakość

dźwięku (mono/stereo). Zgodnie z wcześniej przytoczonymi ustaleniami indeksem urządzenia generującego jest 0 natomiast interesujące nas parametry sygnału to 44,1kHz/16bit. W przypadku opisywanego ćwiczenia nie będzie wymagana generacja dwóch niezależnych sygnałów (*stereo*), dlatego w ustawieniach jakości dźwięku wystarczy wybrać *mono*. Po wprowadzeniu powyższych wartości funkcja *SO config.vi* (rys.3.2) zwraca tzw. *Task ID* czyli numer identyfikacyjny związany z konfiguracją konkretnego urządzenia. Wszystkie następane funkcje sterujące kartą muzyczną wymagać będą podania („przeciągnięcia”) owego *Task ID*. Programowanie generatora wymaga uruchomienie zapisu dźwięku do karty muzycznej. Operacja ta obsługiwana jest przez funkcję *SO Start.vi* (rys.3.2). W celu zapewnienia ciągłości generowanego sygnału, niezależnie od prędkości posiadanego komputera oraz od zadań systemowych wykonywanych w tle, korzystnie jest uaktywnienie i ustawienie rozmiaru bufora wysyłanych danych, wykorzystując funkcję *SO Set Num Buffers.vi* (rys.3.2). Wstępnie rozmiar bufora warto ustalić na 3. Wielkość tą w razie konieczności można zwiększyć.

3.1.2 Generacja funkcji i jej zapis na wyjście urządzenia generującego

Nim rozpoczniemy operację zapisu, wcześniej ustalić należy jaką funkcję chcemy wysłać na wyjście liniowe karty muzycznej. W tym celu najwygodniej skorzystać z *Express VI*a o nazwie *Simulate signal*, będącym generatorem podstawowych funkcji, dostępnym z poziomu diagramu blokowego w *Functions/Express/Signal Analysis*. Pamiętać należy, iż konstrukcja karty muzycznej nie pozwala na generowanie sygnałów stałych. Jak wspomniano wcześniej, w aktualnym ćwiczeniu, interesuje nas sygnał sinusoidalny. Podczas konfiguracji *Express VI*a, w polu *Signal* wybieramy zatem typ sygnału *Sine*. Dodatkowo w polu *Timing* ustalamy prędkość próbkowania (*Samples per second*) zgodną z parametrami stosowanego urządzenia, a więc 44100Hz. Ilość generowanych próbek (*Number of samples*) można ustawić właściwie dowolną, jednak aby rozmiar bufora nie był zbyt duży, sugerowanym rozmiarem jest 1000 próbek. Do złącz *Frequency* oraz *Amplitude* podłączamy ikony odpowiadające wcześniej stworzonym kontrolkom *Częstotliwość* i *Amplituda*. Wpływ wartości parametrów sygnału na jego kształt obejrzeć można na wskaźniku *Waveform Graph*.

Wszystkie wcześniej wprowadzone funkcje, a więc *SO config*, *SO Start* oraz *SO Set Num Buffers*, wymagały jedynie jednokrotnego uruchomienia i należały do procedury konfiguracyjnej sprzętu. Obecnie wprowadzane funkcje służyć będą do zapisu wcześniej przygotowanego sygnału na wyjście liniowe karty muzycznej.

Jako, że generowanie sygnału odbywać się powinno bez przerwy, dlatego *Express VI Simulate signal*, jak i kolejne funkcję obsługujące kartę muzyczną, znajdować powinny się wewnątrz pętli *While* zatrzymywanej przy pomocy kontrolki *STOP* (rys. 3.2). Do krawędzi pętli dostarczyć należy *Task ID* utworzony podczas konfiguracji sprzętu. Zapis sygnału na wyjście karty muzycznej wymaga wprowadzenia dwóch funkcji: *SO Wait.vi* oraz *SO Write.vi* (do każdej „przeciągnąć” należy *Task ID*). Pierwsza funkcja zapewnia ciągłość generowanego sygnału czekając dopóki wyjście urządzenia zakończy emitowanie sygnału. Jeżeli funkcja ta została użyta razem z *SO Set Num Buffers*, jak to ma miejsce w obecnym przykładzie, *SO Wait* czeka dopóki określona ilość buforów zostanie zapełniona. Sam zapis sygnału odbywa się przy pomocy funkcji *SO Write.vi*. Sygnał wychodzący z *Simulate signal* doprowadzić należy do odpowiedniego złącza, zależnego od początkowej konfiguracji sprzętu – w obecnym przykładzie jest to złącze *mono 16-bit*.

3.1.3 Zakończenie obsługi sprzętu

Naduszenie przycisku *STOP* powinno spowodować zakończenie generowania sygnału. Kontrolka ta kończy działanie pętli, wewnątrz której znajdują się funkcję odpowiedzialne za zapis generowanej funkcji do wyjścia liniowego karty muzycznej. Po zakończeniu pętli należy dokonać zamknięcia wyjścia karty wykorzystując funkcję *SO Clear.vi*. Po

dostarczeniu *Task ID* urządzenia do wejścia tejże funkcji, zwolnione zostają wszystkie zasoby systemowe wykorzystywane przez to urządzenie. W celu ponownego wykorzystania funkcji zapisujących, należy ponownie skonfigurować kartę muzyczną.

Jeżeli wszystkie powyższe etapy zostały wykonane poprawnie, wyjście liniowe pierwszej karty muzycznej powinno stać się generatorem sinusoidalnego sygnału elektrycznego o zadanej częstotliwości. Po podłączeniu głośników do wyjścia karty, sygnał elektryczny zostanie zamieniony na sygnał akustyczny. Jeżeli program uruchamia się nie zgłaszając błędów oraz słyszany jest sygnał akustyczny, którego parametry można zmieniać przy pomocy stworzonych kontrolki oznacza to, że generator sygnału działa poprawnie. Kolejnym etapem będzie napisanie analogicznego programu, służącego do obsługi drugiej karty muzycznej jako rejestratora sygnału.

3.2 Rejestrator sygnału elektrycznego

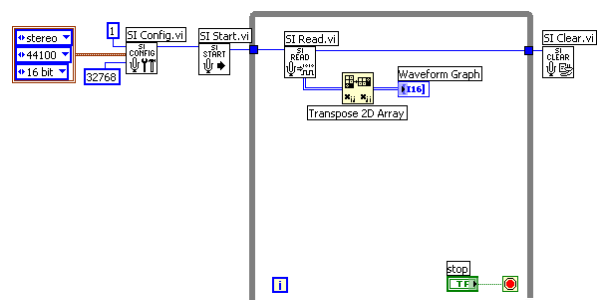
Programowanie rejestratora sygnału odbywać się będzie w sposób analogiczny jak podczas programowania generatora. Rozpoczniemy od stworzenia czystego *VI* a o nazwie *Rejestrator.vi*, na którego panelu frontowym należy umieścić wskaźnik graficzny *Waveform Graph*. Na wskaźniku tym rejestrowane sygnały będą prezentowane w postaci wykresów czasowych. Konieczne będzie umieszczenie przycisku *STOP* służącego do zatrzymywania programu.

Wszystkie funkcje wymagane do sterowania odczytem danych z wejścia liniowego karty muzycznej dostępne są z poziomu *Diagramu Blokowego* środowiska *LabVIEW* w *Fuctions/Graphic and Sounds/Sound/Sound Input*.

Programowanie rejestratora sygnału wykorzystującego kartę muzyczną odbywać się będzie w analogiczny sposób do tego użytego podczas programowania generatora i będzie podzielone na trzy etapy: 1) konfiguracja sprzętu, 2) odczyt przebiegów czasowych napięcia na wejściu urządzenia rejestrującego, 3) zakończenie obsługi sprzętu.

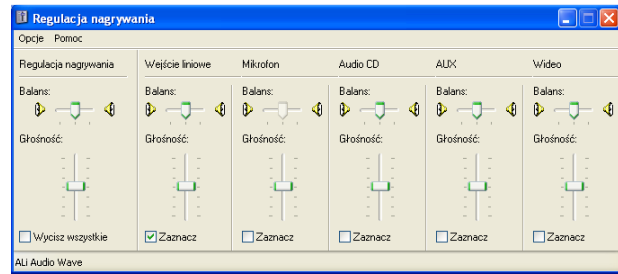
3.2.1 Konfiguracja sprzętu

Konfiguracja wejścia liniowego karty muzycznej odbywa się w sposób zbliżony do tego jaki wprowadzono podczas programowania generatora. Funkcja *SI Config.vi* wymaga podania indeksu oraz parametrów wejściowych urządzenia (rys. 3.4). Jak wspomniano wcześniej, w obecnym przykładzie, rolę urządzenia rejestrującego pełni druga karta muzyczna (indeks *I*). Parametry urządzenia ustalamy jak poprzednio na 44,1kHz/16-bit. Przeprowadzenie eksperymentu wymaga jednoczesnego rejestrowania dwóch sygnałów, zatem w ustawieniach jakości dźwięków wybieramy opcję *stereo*. Dodatkowo funkcja *SI Config.vi* wymaga podania rozmiaru bufora w bajt'ach, używanego przez *LabVIEW* do gromadzenia danych pochodzących z urządzenia pomiarowego. W obecnym ćwiczeniu rozmiar bufora ustalono na 32768. Po podaniu wszystkich wymaganych parametrów, funkcja *SI Config.vi* (podobnie jak poprzednio *SO Config.vi*) zwraca *Task ID*. Wielkość ta powinna zostać doprowadzona do kolejnej funkcji – *SI Start.vi*, która uaktywnia wejście karty dźwiękowej w celu rozpoczęcia gromadzenia nadchodzących danych.



Rys. 3.4 Diagram blokowy rejestratora *Rejestrator.vi*.

Aby rejestrowany sygnał pochodził jedynie z wejścia liniowego (*Line In*) karty muzycznej w ustawieniach *Regulacji nagrywania* drugiej karty należy zaznaczyć „*Wejście liniowe*” (rys.3.5)



Rys. 3.5 Konfiguracja źródła sygnału wejściowego z poziomu systemu operacyjnego

3.2.2 Rejestracja sygnału

Za odczyt wartości napięcia wejściowego odpowiedzialna jest funkcją *SI Read.vi*. Rejestrowanie sygnału, podobnie jak uprzednio jego generowanie, odbywać się powinno w sposób ciągły, dlatego funkcja czytająca dane z wejścia liniowego urządzenia powinna znajdować się wewnątrz pętli *While* zatrzymywanej kontrolką *STOP*. Po „dociągnięciu” *Task ID* do wejścia funkcji *SI Read.vi*, co krok pętli, na odpowiednim złączu wyjściowym (w naszym przypadku *stereo16-bit*) pojawi się tablica liczb, będąca wynikiem rejestracji sygnału. Rozmiar oraz wymiar tablicy zależy od wstępnej konfiguracji sprzętu. Jeżeli podczas konfiguracji wybrano *stereo/16-bit*, jak ma to miejsce w naszym przypadku, wówczas zarejestrowane dane gromadzone są wewnątrz dwurzędowej tablicy (po jednym rzędzie na kanał) o rozmiarze 8192 punktów (32768/ 2 kanały/ 2 bajty). Aby zarejestrowane dane przedstawić na wskaźniku *Waveform Graph* w postaci dwóch wykresów, tablicę dostarczaną przez *SI Read* należy transponować do postaci macierzy dwukolumnowej, wykorzystując funkcję *Transpose 2D Array* (rys. 3.4).

3.2.3 Zakończenie obsługi sprzętu

Kończenie działania rejestratora sygnału wymaga wykonania analogicznych czynności omówionych podczas tworzenia programu *generator.vi*. Naduszenie przycisku *STOP* powoduje zakończenie działania pętli *While*, za którą umieszczona jest funkcja *SI Clear.vi* zamykająca wejścia karty muzycznej oraz zwalniająca wszystkie zasoby systemowe wykorzystywane przez to urządzenie.

3.3 Testowanie poprawności działania Generatorsa i Rejestratora

Na obecnym etapie przeprowadzić można pierwsze próby działania dwóch stworzonych programów – *Generator.vi* oraz *Rejestrator.vi*.

- Zmontować układ według schematu przedstawionego na rysunku 2.1. Przyjmijmy umownie, że napięcie zasilające mierzone będzie na kanale 0, natomiast kanał 1 mierzyć będzie spadek napięcia na oporniku R_p (proporcjonalny do prądu – równanie 12)
- Uruchomić program *Generator.vi*
- Uruchomić program *Rejestrator.vi*. Na wskaźniku graficznym widoczne powinny być dwa wykresy prezentujące przebiegi czasowe napięcia szczytywanego z dwóch kanałów wejścia liniowego karty muzycznej.
- Zmieniając wartość częstotliwości generowanego sygnału prześledzić zmiany przesunięcia fazowego pomiędzy napięciem a prądem oraz zmiany spadku napięcia na oporniku pomiarowym R_p .

4. Analiza rejestrowanego sygnału w celu wyznaczenia tangensa kąta przesunięcia fazowego $\tan(\varphi)$ oraz stosunku napięć U/U_R

Jak już wspomniano wcześniej, wykonanie opisywanego ćwiczenia wymaga wyznaczenia charakterystyk częstotliwościowych tangensa kąta przesunięcia fazowego $\tan(\varphi)$ i stosunku napięć U/U_R , oraz ich porównania z zależnościami teoretycznymi danymi równaniami (10), (14) oraz (16), (19). W celu wyznaczenia poszukiwanych wielkości konieczna jest znajomość amplitudy oraz fazy każdego z dwóch kanałów rejestrowanego sygnału.

W tej części opisu stworzymy *subvi*'a, który po dostarczeniu zarejestrowanego sygnału w postaci tablicy dwukolumnowej przez funkcję *SI Read*, znajdującą się w programie *Generator.vi*, wykona odpowiednią analizę, zwracając interesujące nas wartości $\tan(\varphi)$ oraz U/U_R .

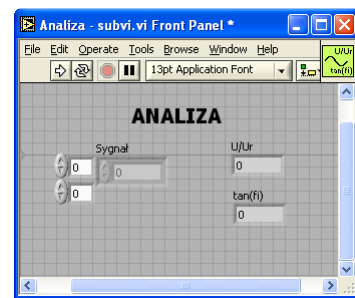
Rozpoczniemy od stworzenia pustego *VI*'a o nazwie *Analiza-subvi.vi*. Na panelu frontowym umieścić należy kontrolkę tablicy dwuwymiarowej o nazwie *Sygnal* reprezentującą wejściowe dane pomiarowe. Umieścimy też dwa wskaźniki liczbowe U/U_R oraz $\tan(\varphi)$, na których pojawiać się będą obliczone wartości odpowiednich wielkości (rys. 4.1).

Programowanie diagramu blokowego rozpoczniemy od wydzielenia z dostarczanej przez rejestrator tablicy dwuwymiarowej dwóch kolumn, w których zbierane są dane pomiarowe z dwóch kanałów wejścia liniowego karty muzycznej. W tym celu użyć należy funkcji *Index Array* znajdującej się w katalogu funkcji tablicowych diagramu blokowego (*Functions/Array*). Wydzielone w ten sposób tablice jednowymiarowe, analizowane będą wykorzystując funkcję *Extract Single Tone Information* będącą jedną ze specjalistycznych funkcji środowiska *LabVIEW* dostępną z poziomu diagramu blokowego w *Functions/Analyze/Waveform Measurements*. Funkcja ta szuka w tablicy, zawierającej poddawany analizie sygnał, ton o maksymalnej amplitudzie i zwraca jego częstotliwość, amplitudę i fazę (w stopniach). W naszym przypadku funkcja ta zostanie użyta w celu wyznaczenia amplitudy i fazy sygnałów pochodzących z kanałów 0 i 1. Aby wyznaczyć pierwszą z poszukiwanych wielkości – stosunek napięć U/U_R – wystarczy podzielić amplitudę sygnału z kanału 0 przez analogiczną amplitudę odczytaną z kanału 1. Poszukiwaną wartość $\tan(\varphi)$ obliczamy z poniższej zależności:

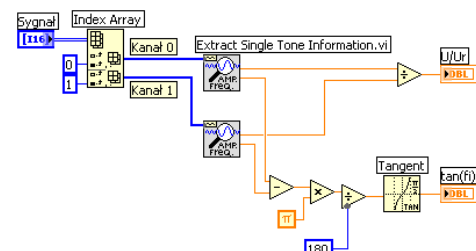
$$\tan(\varphi) = \tan\left((\varphi_I - \varphi_U) \frac{\pi}{180}\right) \quad (20)$$

gdzie φ_U oraz φ_I są zwróconymi przez funkcję *Extract Single Tone Information* wartościami faz, odpowiednio, napięcia (kanał 0) oraz prądu (kanał 1) wyrażonymi w stopniach (rys. 4.2).

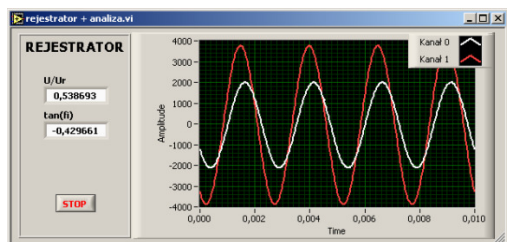
Po skonfigurowaniu złącz (terminali) oraz opcjonalnej edycji ikony, nowostworzony *subvi* jest gotowy do użycia w programie *Generator.vi* (rys.4.3, 4.4).



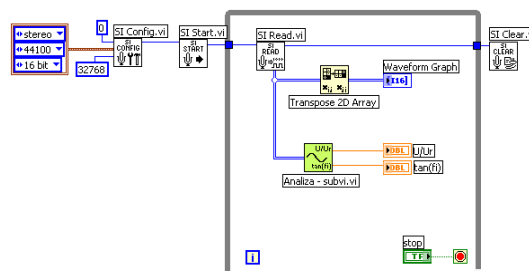
Rys. 4.1 Panel frontowy *subvi*'a *Analiza-subvi.vi*



Rys. 4.2 Diagram blokowy *subvi*'a *Analiza-subvi.vi*



Rys. 4.3 Panel frontowy rejestratora wzbogaconego o funkcję analizy sygnału.

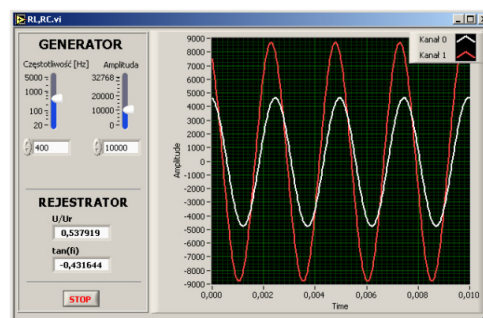


Rys. 4.4 Diagram blokowy rejestratora wzbogaconego o funkcję analizy sygnału.

5. Komunikacja z Generatorem oraz Rejestratorem sygnału z poziomu odrębnego VI'a – zmienne globalne

Do tej pory skupiliśmy się na stworzeniu dwóch wyspecjalizowanych programów. Pierwszy program, nazwany *Generator.vi*, obsługuje pierwszą kartę muzyczną i służy jedynie jako generator sygnału o zadanych parametrach (częstotliwość i amplituda). Drugi program o nazwie *Rejestrator.vi*, obsługuje drugą kartę muzyczną i służy jedynie jako rejestrator i analizator sygnału.

W tej części ćwiczenia skupimy się na stworzeniu aplikacji, z poziomu której możliwa będzie komunikacja oraz sterowanie dwoma wcześniej powstałymi programami. Programowanie rozpoczniemy od stworzenia nowego VI'a o nazwie *RL,RC.vi*. Na panelu frontowym umieszczone zostaną kontrolki umożliwiające zmianę parametrów generowanego sygnału, wskaźnik graficzny *Waveform Graph*, na którym możliwa będzie obserwacja sygnałów rejestrowanych, dwa wskaźniki numeryczne prezentujące wartości $\tan(\varphi)$ i U/U_R , oraz przycisk *STOP* służący do zatrzymania wszystkich programów (rys.5.1).

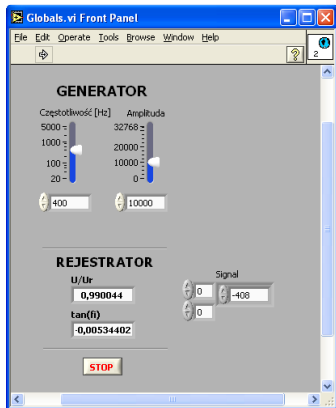


Rys. 5.1 Panel frontowy programu RL,RC.vi

Wartości stworzonych w *RL,RC.vi* kontrolki odpowiadające częstotliwości i amplitudzie sygnału jaki zamierzamy wygenerować, należy „przesłać” do programu *Generator.vi* w miejsce analogicznych kontrolki. Jednocześnie sygnał zarejestrowany oraz przeanalizowany przez program *Rejestrator.vi* należy przesłać do *RL,RC.vi* wykreślając go na wskaźnikach graficznym i numerycznych. W ten sposób, w oknie jednej aplikacji, możliwe będzie wykonywanie zmian parametrów sygnału zasilającego szeregowy układ RL bądź RC oraz obserwacja odpowiedzi tegoż układu, analizując zmiany kształtu rejestrowanego sygnału oraz związane z tym zmiany wartości obliczanych wielkości $\tan(\varphi)$ oraz U/U_R .

5.1 Wprowadzenie zmiennych globalnych

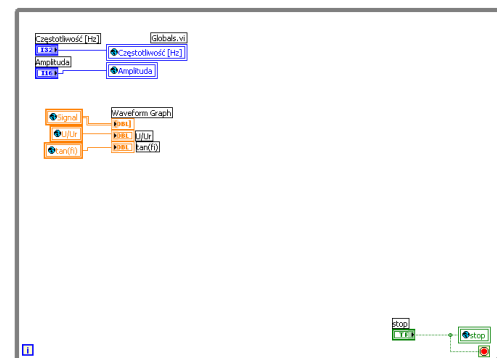
Przesyłanie wartości kontrolki i wskaźników pomiędzy VI'ami realizowane jest w środowisku *LabVIEW* przy pomocy tzw. *zmiennych globalnych (Global Variables)*. Stworzenie zmiennej globalnej polega na wybraniu odpowiedniej ikony znajdującej się w katalogu struktur diagramu blokowego (*Functions/Structures/Global Variable*). W momencie stworzenia zmiennej globalnej, *LabVIEW* automatycznie tworzy specjalnego, globalnego VI'a, który posiada panel frontowy, ale nie posiada diagramu blokowego. Dostęp do panelu frontowego zmiennej globalnej odbywa się po podwójnym „kliknięciu” ikony zmiennej. Jeżeli jest to zmienna nowostworzona, panel frontowy jest pusty. Należy na nim umieścić kontrolki, bądź wskaźniki, do których będziemy chcieli mieć dostęp z poziomu większej



Rys. 5.2 Panel frontowy utworzony dla zmiennych globalnych

ilości VI 'i. Kopiujemy zatem kontrolki generatora (amplituda i częstotliwość), przycisk *STOP*, wskaźniki generatora ($\tan(\varphi)$, U/UR). Stworzymy również kontrolkę tablicy dwuwymiarowej *Sygnal*, którą wykorzystamy do zapisu i odczytu przebiegu czasowego rejestrowanego sygnału (rys.5.2). Po zapisaniu panelu frontowego zmiennej globalnej, nadając jej nazwę np. *Globals.vi*, wszystkie znajdujące się na panelu kontrolki bądź wskaźniki dostępne są z poziomu dowolnego VI 'a. Powróćmy zatem do programu *RL,RC.vi*. Stworzymy pętlę *While* zatrzymywaną przyciskiem *STOP*. Wewnątrz pętli, umieścimy wszystkie ikony znajdujących się na panelu frontowym obiektów.

Wartości kontrolek odpowiadające częstotliwości oraz amplitudzie generowanego sygnału, przekazywane będą do odpowiedniego elementu zmiennej globalnej. Wczytujemy zatem globalnego VI 'a *Globals.vi* (*Functions/Search a VI...*). Na diagramie blokowym pojawić się powinna ikona zmiennej globalnej, w której widnieje nazwa jednego z elementów umieszczonego wcześniej na panelu frontowym *Globals.vi*. Klikając prawym klawiszem myszy na ikonie zmiennej globalnej, wybrać można odpowiedni element z listy *Select Item*. Wybierzmy zatem element *Częstotliwość* i „przeciągnijmy” do niego wartość zwracaną przez kontrolkę o tej samej nazwie (rys. 5.3). Następnie skopiujemy zmienną globalną, wybierzmy kolejny element – *Amplituda* – i analogicznie prześlijmy wartość kontrolki do zmiennej globalnej. W ten sam sposób prześlijmy wartość logiczną przycisku *STOP* do odpowiedniego elementu zmiennej globalnej. Przywołajmy panele frontowe *RL,RC.vi* oraz *Globals.vi* i uruchommy program. Zauważmy, że zmiana wartości kontrolki na panelu frontowym *RL,RC.vi*, powoduje zmianę wartości odpowiednich kontrolki wewnątrz *Globals.vi*. Stąd, zmodyfikowany program *Generator.vi*, będzie pobierał nowe wartości i generował sygnał o nowych parametrach (modyfikacją programów *Generator* i *Rejestrator* zajmiemy się w następnym podpunkcie).



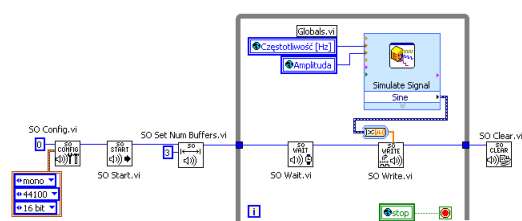
Rys. 5.3 Diagram blokowy programu *RL,RC.vi*

Zmodyfikowany program *Rejestrator* wyśle zarejestrowany sygnał (w postaci przebiegu czasowego napięcia), jak i obliczone wielkości $\tan(\varphi)$ oraz U/U_R , do odpowiednich elementów zmiennej globalnej. Główny program *RL,RC* pobierze znajdujące się wewnątrz zmiennej globalnej dane, a znajdujące się na jego panelu frontowym wskaźniki, wykorzystane staną w celu ich wizualizacji. Umieścimy zatem dodatkowe trzy kopie zmiennej globalnej wybierając element każdej z kopii odpowiednio do istniejących wskaźników: *Sygnal*, $\tan(\varphi)$, U/UR . Przed „przeciągnięciem” wartości zmiennej globalnej do wskaźnika, należy zmienić ustawienie zmiennej, dotyczące kierunku przesyłania danych. Domyślnie ustawienia umożliwiają zapis wartości do zmiennej, podczas gdy w obecnym przypadku, pragniemy odczytać wartość zmiennej w celu jej zapisu do wskaźnika. Zmian dokonujemy klikając prawym klawiszem na ikonie zmiennej i wybierając *Change to Read*.

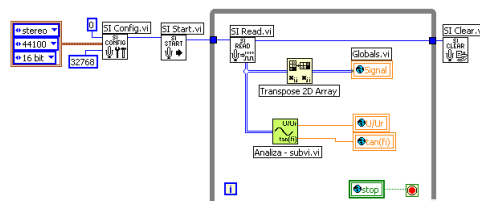
Ostatnim elementem, wartym umieszczenia w głównej pętli programu, jest niewielkie opóźnienie (*Functions/Time&Dialog/Wait(ms)*), którego celem jest odciążenie procesora komputera pomiarowego (rys. 5.3).

5.2 Modyfikacja programów *Generator* oraz *Rejestrator*

Wprowadzana modyfikacja ograniczać się będzie do zmiany miejsca pobierania i wysyłania danych. Zamiast pobierać dane z kontrolki znajdujących się na panelu frontowym programu *Generator*, będziemy pobierać je ze zmiennej globalnej. Analogicznie zamiast wysyłać zarejestrowany sygnał na wskaźniki umieszczone na panelu frontowym programu *Rejestrator*, wyślemy je do odpowiednich elementów zmiennej globalnej. Modyfikacja obu programów polegać będzie zatem na usunięciu z panelu frontowego wszystkich istniejących kontrolki i wskaźników oraz zastąpieniu ich ikon na diagramie blokowym, odpowiednimi zmiennymi globalnymi stworzonymi w poprzednim podpunkcie (rys. 5.4, 5.5)



Rys. 5.4 Diagram blokowy zmodyfikowanego Generatora sygnału. *Generator-globals.vi*



Rys. 5.5 Diagram blokowy zmodyfikowanego Rejestratora sygnału. *Rejestrator+analiza-globals.vi*

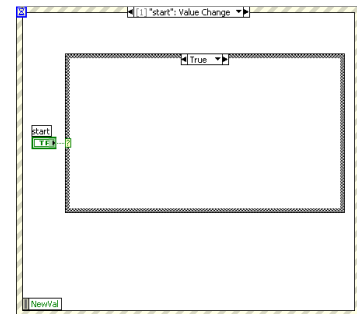
Zmodyfikowane programy zapisujemy pod nazwami: *Generator-globals.vi* oraz *Rejestrator+analiza-globals.vi*

5.3 Testowanie poprawności komunikacji

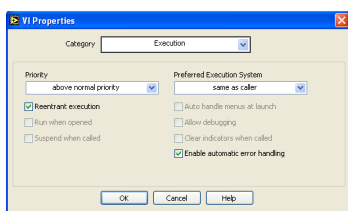
W celu sprawdzenia poprawności komunikacji między *RL,RC.vi* a zmodyfikowanymi generatorem i rejestratorem, uruchamiamy równocześnie wszystkie 3 programy. Jeżeli komunikacja przebiega poprawnie, to zmianom wartości parametrów sygnału zasilającego szeregowy układ RL bądź RC towarzyszyć powinny zmiany kształtu rejestrowanego sygnału oraz związane z tym zmiany wartości obliczanych wielkości $\tan(\varphi)$ oraz U/U_R .

6. Dynamiczne wywoływanie VI'i

W tej części wprowadzimy kilka zmian w aplikacji *RL,RC.vi*, których celem będzie dynamiczne uruchomienie wcześniej stworzonych *subVI*i: *Generator-globals.vi* oraz *Rejestrator+analiza-globals.vi*. W pierwszej kolejności na panelu frontowym utworzyć należy dodatkowy przycisk *START*, którego celem będzie uruchamianie i zatrzymywanie procedur generacji i rejestracji sygnału. Obsługa zdarzenia z jednoczesnym sprawdzeniem wartości kontrolki *START* wymaga umieszczenia na diagramie blokowym aplikacji *RL,RC* struktury *Event Structure* obsługującej zdarzenie zmiany wartości (*Value Change*), wewnątrz której znajdować powinna się struktura *Case* sprawdzająca wartość kontrolki *START* (rys. 6.1). Jeżeli wartością kontrolki jest *True* (przycisk został wdużony), to w odpowiedniej zakładce struktury *Case* powinny zostać wykorzystane funkcje do dynamicznego uruchomienia *subVI*i generator i rejestrator. Do chwili ponownego naciśnięcia przycisku *START*, generator będzie wysyłać sygnał o parametrach zadanych na panelu frontowym aplikacji *RL,RC*, natomiast rejestrator będzie przysyłać na panel frontowy szczytywany sygnał wraz z obliczonymi wielkościami $\tan(\varphi)$ oraz U/U_R . Jeżeli wartością kontrolki *START* jest *False* (przycisk został wyduszyony), to w odpowiedniej zakładce struktury wyboru *Case* użyte zostaną funkcje zatrzymujące pracujące do tej pory *subVI*e.



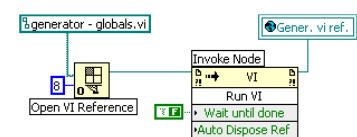
Rys. 6.1 Struktury *Event Structure* i *Case*. Obsługa zdarzenia *Value Change* z jednoczesnym sprawdzeniem wartości kontrolki *START*.



Rys. 6.2 Okno ustawień właściwości uruchomienia VI'a

Chcemy aby programy generator i rejestrator uruchamiany były „w tle” oraz żeby pracowały równolegle. Ten typ pracy nie jest domyślnym, zatem należy odpowiednio przygotować stworzone programy. Tryb pracy *Generator-globals.vi* oraz *Rejestrator+analiza-globals.vi* zmieniamy w ustawieniach VI'a na tryb współużywalny (*File/VI Properties/Execution/Reentrant Execution*). W celu zapewnienia szybszego dostępu do zasobów systemowych, priorytet zadania ustawić można na wyższy od normalnego (*above normal priority*) (rys. 6.2).

Dynamiczne uruchomienie VI'a wymaga stworzenia tzw. odnośnika do uruchamianego VI'a (*VI Reference; refnum*) będącym jego unikalnym identyfikatorem. Funkcją zwracającą pożądany *refnum* jest *Open VI Reference* znajdująca się w funkcjach kontroli aplikacji diagramu blokowego (*Functions/Application Control*). W przypadku obecnego ćwiczenia, jako parametry wejściowe tejże funkcji wystarczy podać: ścieżkę dostępu do żądanego VI'a (*Generator-globals.vi* lub *Rejestrator+analiza-globals.vi*) oraz numer opcji precyzującej sposób uruchamiania i zachowania się VI'a (przygotowanie VI'a do pracy równoległej wymaga podania jako opcji liczby 8). Uruchomienie VI'a odbywa się poprzez wybranie odpowiedniej opcji z tzw. węzła wywołań (*Functions/Application Control/Invoke Node*), dzięki któremu przywołać można dowolną metodę (*Method*) obiektu, w tym także VI'a, opisanego odpowiednim odnośnikiem (*refnum*). Po „przeciągnięciu” *refnum* do węzła wywołań, z listy *Methods* wybrać należy *Run VI* oraz ustawieniu parametru *Wait until done* na *False*. Parametr ten określa czy zakończenie



Rys. 6.3 Dynamiczne uruchomienie VI'a *Generator – globals.vi*

działania funkcji *Run VI* ma nastąpić po zakończeniu pracy uruchomionego *VI*'a. Pozostawienie domyślnej wartości parametru (*True*) spowoduje zatrzymanie się programu *RL,RC* (z którego poziomu uruchamiamy metodę *Run VI*) w miejscu wywołania *subVI*'i generatora lub rejestratora, do czasu zakończenia ich pracy, i tym samym spowoduje zablokowanie panelu frontowego programu *RL,RC*. Drugi parametr funkcji *Run VI* (*Auto Dispose Ref*) nie będzie omawiany i na etapie obecnego ćwiczenia wystarczy, aby pozostawić jego domyślną wartość. Identyfikatory *VI*'i (*refnum*) konieczne będą w następnym etapie w celu zatrzymania działania *subVI*'i generatora i rejestratora, i będą zapamiętane w postaci zmiennych globalnych o nazwach *Gener. vi ref.* oraz *Rejes. vi ref.*, które tworzymy w sposób analogiczny do opisanego w części 5.1 (rys. 6.3).

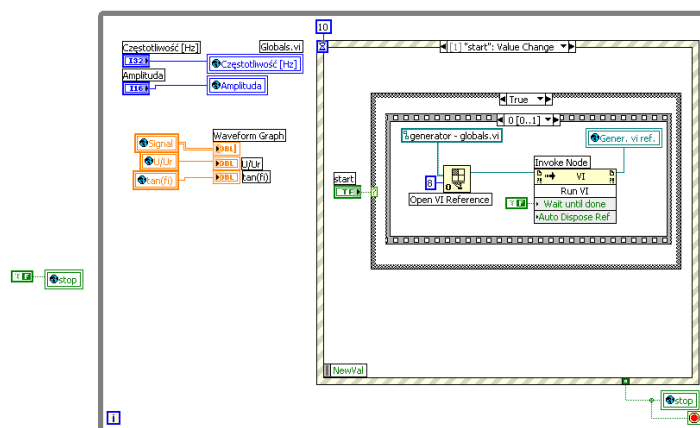
W zakładce struktury *Case* odpowiadającej wartości *False* kontrolki *START* powinny zostać umieszczone funkcje, których zadaniem będzie zatrzymanie wcześniej uruchomionych *subVI*'i. Zatrzymanie *VI*'a jest, podobnie jak jego uruchomienie, metodą dostępną z opcji węzła wywołań (*Invoke Node*). Wcześniej stworzony i zapamiętany w postaci zmiennej globalnej identyfikator *VI*'a (*refnum*) przesyłamy do węzła wywołań, a z opcji wybieramy *Abort VI* (rys. 6.4).



Rys. 6.4
Dynamiczne zatrzymanie *VI*'a *Generator* – *globals.vi*

7. Kończenie aplikacji

W celu zakończenia i sprawdzenia poprawności działania aplikacji *RL,RC* należy umieścić procedury umożliwiające dynamiczne uruchamianie i kończenie *subVI*'i *Generator-globals.vi* oraz *Rejestrator+analiza-globals.vi*, w odpowiednich zakładkach struktury *Case* sprawdzającej stan przycisku *START*. Dodatkowo, przed rozpoczęciem pracy głównego programu (przed wejściem do pętli *While*) warto upewnić się czy „globalny” *STOP* jest „wyłączony” przypisując mu wartość *False* (rys. 7.1). Wartość *True* przycisku *STOP* spowodowałaby natychmiastowe zakończenie pracy generatora i rejestratora uniemożliwiając ich uruchomienie.



Rys. 7.1 Diagram blokowy aplikacji *RL,RC.vi*

W powyższym opisie pokazano w jaki sposób, wykorzystując środowisko *LabVIEW*, stworzyć główny rdzeń aplikacji wykorzystującej dwie karty muzyczne jako generator i rejestrator sinusoidalnego sygnału elektrycznego. Opracowana aplikacja umożliwia obserwację zmian wartości parametrów $\tan(\varphi)$ oraz U/U_R , wywołanych zmianą częstotliwości sygnału zasilającego szeregowy układ RL bądź RC. Możliwości środowiska *LabVIEW* są znacznie większe i w celu dalszego usprawnienia wykonywania pomiarów oraz zwiększenia funkcjonalności aplikacji, sugerowane jest stworzenie dodatkowych procedur, których celem będzie:

- Zapis danych pomiarowych do pliku
- Automatyzacja pomiaru zależności częstotliwościowych
- Analiza wyników
- Inne formy prezentacji sygnału (np. w postaci krzywych Lissajous)