

# Symulacja i pomiar, czyli LTspice i Analog Discovery 2 w rękach konstruktora (23)

## O czym nie mówiliśmy, a powinniśmy

*Niniejszy cykl trwa już bardzo długo. To wystarczający powód do tego, by przeanalizować poruszane zagadnienia. Okazało się, że jest kilka tematów, o których nie było mowy, a z racji tego, że należałoby je zaliczyć do podstawowych, powinny być choćby wspomniane. Nadrabiamy zaległości.*

Rozpoczynamy od ważnego zagadnienia, które nie było omawiane wprost, a jest jedną z ważniejszych cech symulatora LTspice.

### Hierarchiczna struktura schematów

Hierarchiczna struktura schematów symulatora LTspice jest jedną z najważniejszych cech tego programu. Zwiększa ich czytelność, szczególnie wtedy, gdy są znacznie rozbudowane, pozwala też

definiować własne elementy. Zalety schematów hierarchicznych pokazano na przykładzie. Naszym zadaniem jest symulacja układu elektronicznego selektywnie filtrującego kilka częstotliwości. Użytkownik musi mieć oczywiście możliwość łatwego strojenia filtrów. Problem rozwiązujemy, stosując mostki „podwójne T”.

Pierwszym, naturalnym podejściem jest narysowanie kilku takich mostków na jednym schemacie. Ponieważ mamy zapewnić łatwą możliwość strojenia, warto zapewnić parametryczne zadawanie wartości każdego elementu. Jeden mostek zawiera 6 elementów. Gdybyśmy chcieli zastosować np. 4 takie mostki, przestrojenie ich wszystkich wymagałoby zmiany wartości aż 24 elementów. Parametryzacja zredukuję tę liczbę do 8. Przykład kompletnego schematu pokazano na **rysunku 1a**, a wykresy częstotliwościowe na **rysunku 1b**. Parametry elementów każdego filtra umieszczono tak, aby było wyraźnie widoczne, do którego z nich się odnoszą.

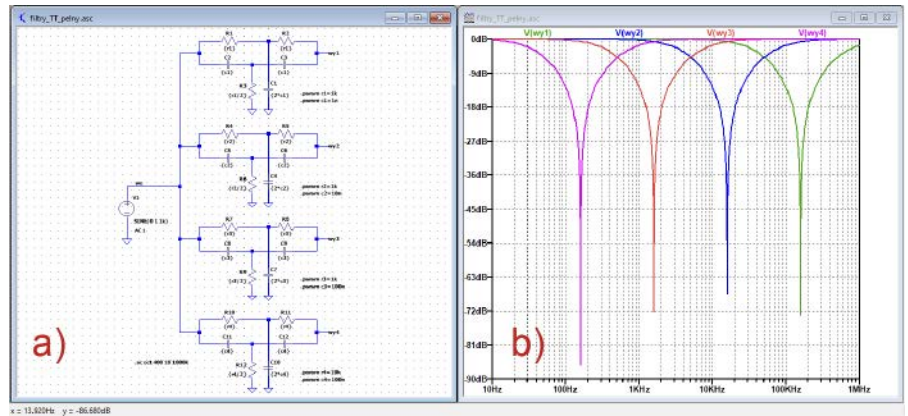
Teraz tę samą symulację przeprowadzimy z zastosowaniem schematu hierarchicznego. Najpierw należy zbudować podstawową strukturę filtra. Rysujemy więc schemat tak, jak to robimy w przypadku każdego symulowanego układu. Nie wprowadzamy jednak konkretnych wartości elementów, a definiujemy je parametrycznie (**rysunek 2**). Ważne jest nadanie etykiet wszystkim węzłom, do których będzie konieczny dostęp ze schematu o wyższym poziomie hierarchicznym. W tym przypadku jest to węzeł wejściowy („we”) i wyjściowy („wy”). Schemat zapisujemy w formacie ASC, a więc takim, jak każdy symulowany układ. Warto ponadto nadać nazwę jednoznacznie kojarzącą się z funkcją układu. Niech to będzie *filtrTT.asc*.

Aby korzystać z tak zaprojektowanego filtra, konieczne jest zdefiniowanie jego oznaczenia schematowego. Otwieramy więc odpowiednie okno edytora: File → New Symbol (**rysunek 3**). Na ekranie ukazuje się pusta plansza z symbolem kotwicy tworzonego elementu. Rysujemy wokół niej prostokąt (Draw → Rect lub przycisk R na klawiaturze) o rozmiarach 4×2 oczka siatki. Kolejną czynnością jest naniesienie wyprowadzeń filtra (Edit → Add Pin Port lub przycisk P), do których doprowadzamy połączenia z prostokątem symbolizującym filtr. Każdy taki element musi mieć swoje oznaczenie (etykieta) i numer porządkowy, które wprowadzamy bezpośrednio po wybraniu powyższej komendy. Dodatkowo możemy decydować o tym, czy etykieta ma być wyświetlona na schemacie, czy nie. Jeśli decydujemy się na wyświetlanie, możemy dodatkowo określić jej położenie względem pinu – opcje TOP, LEFT, RIGHT, DOWN, Vertical Text i Offset (**rysunek 4**). Aby uczynić symbol filtra bardziej czytelny, w obrębie prostokąta umieściliśmy ponadto grafikę przypominającą charakterystykę filtra TT. Jest ona wykonana przy użyciu polecenia Draw-Line.

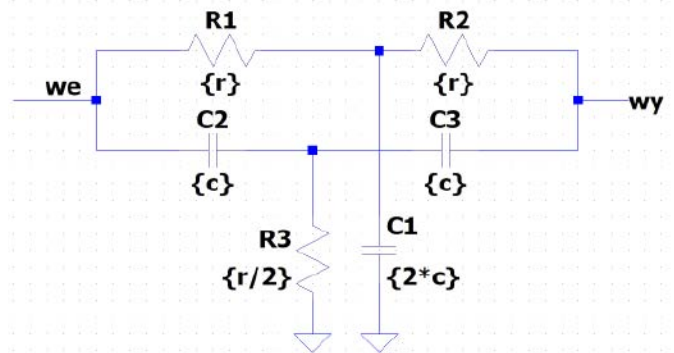
Umieszczanie etykiet z jednej strony ułatwia później, w trakcie tworzenia schematu symulowanego układu, rozpoznawanie funkcji poszczególnych wyprowadzeń, jednak nie mamy możliwości ustawiania wielkości liter i dla niewielkich elementów będą one zaciemniały schemat. W takich przypadkach korzystniejsze jest wyłączenie wyświetlania etykiet – opcja NONE (Not Visible) i zamiast nich wstawienie tekstu komentarza (Draw → Text). Teraz już można wybrać wielkość liter, np.: Font size=0,625. Tak zdefiniowany symbol będzie bardziej czytelny na schemacie.

Pozostaje jeszcze uzyskanie dostępu do parametrów filtra, które powinny być jakoś przekazywane do najwyższego poziomu hierarchii przez symbol schematowy. W tym celu definiujemy domyślne parametry R i C. Przyjmujemy np., że R będzie równe 1 kΩ, a C 10 nF. Przy każdym wprowadzeniu filtra na schemat jego parametry będą określone tak przyjętymi wartościami. Możemy je oczywiście zmienić już na schemacie układu symulowanego w dowolnym momencie. Będzie to potrzebne do strojenia filtra w trakcie symulacji. Parametry nanosimy poprzez atrybut SpaceLine. Wykonujemy polecenie Edit-Attributes, wybieramy opcję Edit Attributes i wypełniamy odpowiednie pole (**rysunek 5**). Warto ponadto uczynić ten atrybut widocznym na schemacie, co następuje po wskazaniu go w oknie atrybutów (*Ctrl W*) i umieszczeniu w stosownym miejscu na schemacie. Wynik powyższych prac zapisujemy w pliku o takiej samej nazwie jak schemat, tylko z automatycznie zmienionym rozszerzeniem na ASY (*filtrTT.asy*). Element warto zapisać w domyślnym folderze zawierającym biblioteki elementów. Będzie wówczas zawsze widoczny w programie. Jeśli o to nie zadbamy, może być później problem z dostępem do niego. Aby stał się widoczny, tworzony schemat układu symulowanego powinien być zapisany w folderze, w którym znajduje się plik *filtrTT.asy* lub w którymś z nadrzędnych.

Na tym kończą się prace nad definiowaniem własnego elementu należącego do najniższego poziomu struktury schematu. Możemy przystąpić do symulacji z jego zastosowaniem. Teraz nie musimy rysować każdego filtra, korzystamy z przygotowanego elementu schematowego.

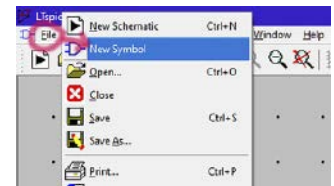


**Rysunek 1. Symulacja układu z czterema filtrami selektywnymi: a) schemat, b) charakterystyki częstotliwościowe**

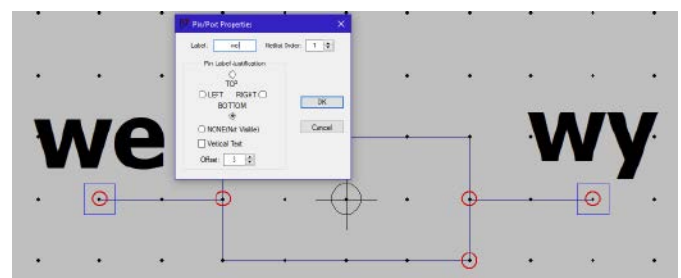


**Rysunek 2. Schemat filtra „TT” z parametrycznie określonymi wartościami elementów**

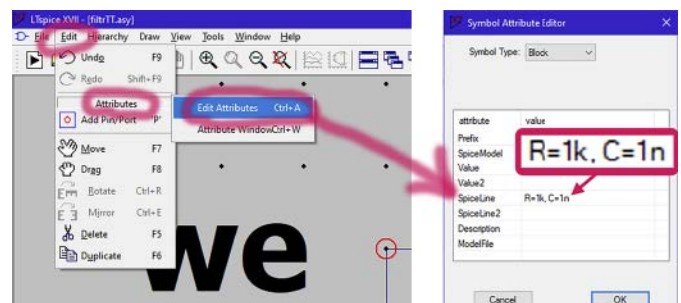
Otwieramy czysty arkusz i umieszczamy na nim symbol filtra (Edit → Component lub przycisk F2). Jeśli elementu nie ma w zestawieniu, wybieramy na widocznym u góry okna liście Top Directory katalog, w którym został zapisany nasz filtr. Powinien się po tym pokazać element filtrTT (**rysunek 6**).



**Rysunek 3. Menu z poleceniami otwierającymi arkusz edycji nowego symbolu**

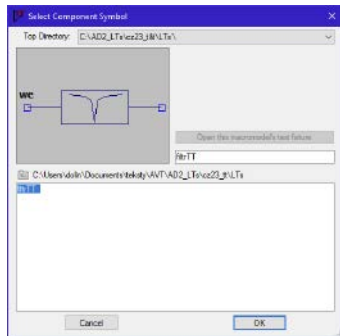


**Rysunek 4. Okno właściwości wprowadzanego pinu**



**Rysunek 5. Okno wprowadzania domyślnych wartości elementów filtra**

Wybieramy go i od razu powielamy w czterech egzemplarzach. Następnie stroimy filtry. Można to wykonać na kilka sposobów. Po pierwsze, klikamy na wartości elementów wyświetlane pod prostokątem symbolizującym filtr (rysunek 7). Otwiera się po tym okno edycji parametrów, w którym wpisujemy żądane wartości. Druga metoda polega na kliknięciu prawym przyciskiem myszki w obrębie elementu symbolizującego filtr.



Rysunek 6. Okno wyboru elementu bibliotecznego użytkownika

Oprócz możliwości zmiany parametrów filtru można również przeprowadzić edycję jego symbolu schematowego (przycisk ekranowy Open Symbol) oraz schematu filtru (Open Schematic) – rysunek 8.

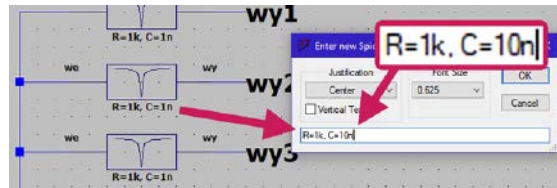
Filtry stroimy dokładnie tak samo jak w pierwszej symulacji, tzn.: R1=1 kΩ, C1=1 nF (1. filtr), R2=1 kΩ, C2=10 nF (2. filtr), R3=1 kΩ, C2=100 nF (3. filtr) i R4=10 kΩ, C2=100 nF (4. filtr). Warto zauważyć, że nadal pozostaje możliwość parametrycznego określania wartości elementów tak, jak to zaprezentowano w ostatnim 5. filtrze. Pełny schemat układu filtrującego i wyniki symulacji zostały pokazane na rysunku 9.

### Funkcje

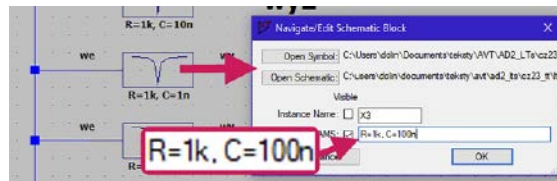
Funkcje były już wprawdzie opisywane w 13. odcinku, w którym omawialiśmy symulację metodą Monte Carlo. Wracamy do nich, ale w nieco innym kontekście. Tym razem, oprócz typowej funkcji definiowanej przez użytkownika, czyli przez nas, zastosujemy jedną z wielu funkcji bibliotecznych symulatora LTSpice. Ale po kolei...

Ostatnio zadano mi pytanie: „No dobrze, ale właściwie po co nam symulator?” Poniższy przykład może być odpowiedzią. Wyobraźmy sobie, że budujemy system nawigacji bezwładnościowej oparty na akcelerometrze. Nie jesteśmy pewni ani rozwiązań układowych, ani parametrów technicznych, które należy przyjąć do realizacji projektu. Zamiast budować w ciemno urządzenie i pracować na jego fizycznym modelu, co oczywiście wiąże się z kosztami, może lepiej sprawdzać jego działanie na modelu wirtualnym. Zastosujemy to podejście do projektowanej nawigacji. Nie będziemy tu budować kompletnego urządzenia, sprawdzimy jedynie koncepcję na uproszczonym jednowymiarowym rozwiązaniu. Mamy zatem jednoosiowy akcelerometr ustawiony w kierunku ruchu. Jeżeli nie dysponujemy żadnym fizycznym akcelerometrem, możemy wygenerować jakieś teoretyczne dane wejściowe. Ja akurat miałem takie urządzenie, więc użyłem go do zebrania rzeczywistych danych wejściowych. Problem wózka rozwiązałem, korzystając z samochodzika zabawki. Akcelerometr zainstalowałem w jego kabinie. Wózek rozpędzamy i puszczaamy, pozwalając na wyhamowanie na skutek sił tarcia. Przyspieszenie jest w trakcie eksperymentu rejestrowane, a następnie zapisywane w formacie tekstowym w postaci par liczb: (czas, przyspieszenie). Dane takie można bez trudu zaimportować w symulatorze do źródła napięciowego z funkcją PWL. Zapewniamy w ten sposób połączenie świata rzeczywistego z wirtualnym.

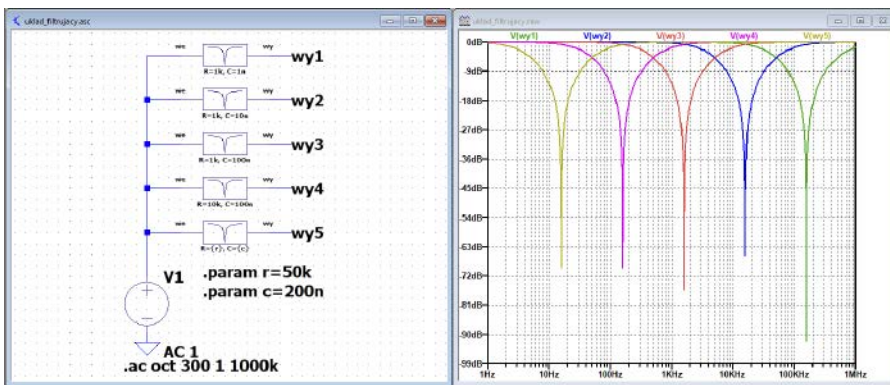
Na schemacie z rysunku 10 jest to źródło V1, a pochodzący z tego wirtualnego akcelerometru sygnał jest dostępny w węzle opisanym etykietą AX. Po wykonaniu wstępnej symulacji czasowej okazuje się, że sygnał ten jest dość mocno zaszumiony (rysunek 11). Widać w nim każdą nierówność podłoża, po którym porusza się wózek. Istnieje więc podejrzenie, że konieczna będzie jakaś filtracja tego sygnału. Ale jak dobrać rząd filtru i jego częstotliwość graniczną? Czy w ogóle ten zabieg faktycznie będzie potrzebny?



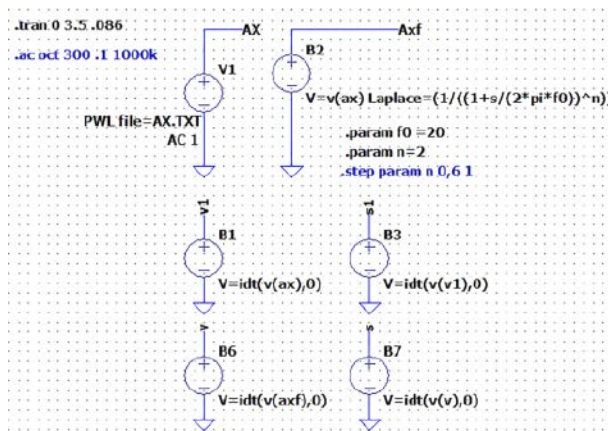
Rysunek 7. Okno wprowadzania parametrów elementu z niższego poziomu hierarchicznego – metoda 1



Rysunek 8. Okno wprowadzania parametrów elementu z niższego poziomu hierarchicznego – metoda 2

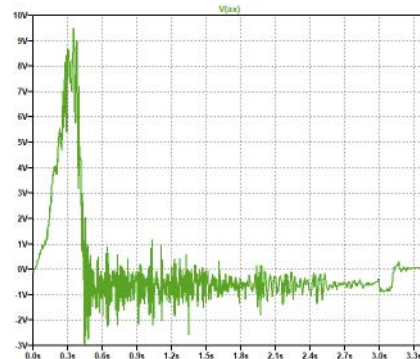


Rysunek 9. Wyniki symulacji układu filtrującego z zastosowaniem schematu hierarchicznego



Rysunek 10. Schemat do sprawdzenia koncepcji układu nawigacji bezwładnościowej

Do rozstrzygnięcia tej wątpliwości zastosujemy model filtru dolnoprzepustowego. Nie będziemy go jednak budować na piechotę z elementów elektronicznych – wzmacniaczy operacyjnych, kondensatorów, rezystorów itp. Skorzystamy z cyfrowego modelu opisanego tylko transmitancją. W symulacji posłużymy



Rysunek 11. Wykres rzeczywistego przyspieszenia zarejestrowanego podczas poruszania się wózka

się behawioralnym źródłem napięciowym B2, którego napięcie jest opisane przekształceniem Laplace'a:  $V=v(ax)$   
 $Laplace=(1/((1+s/(2\cdot\pi\cdot f_0))^n))$ . Zbudowaliśmy w ten sposób filtr dolno-przepustowy filtrujący sygnał z węzła AX. Ma on parametrycznie zadawaną częstotliwość graniczną  $f_0$ , regulowany jest też rząd filtru  $n$ .

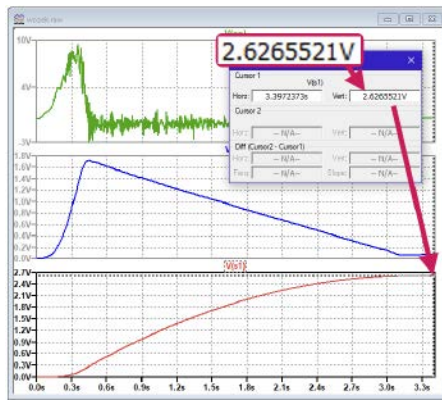
Przed nami najważniejsze zadanie: musimy obliczyć trajektorię wózka. Z fizyki wiemy, że całka z przyspieszenia to prędkość, a całka z prędkości to droga. Powinniśmy więc dwukrotnie scałkować sygnał z akcelerometru. Znowu korzystamy ze źródła behawioralnego. Tym razem napięcie źródła jest obliczane za pomocą funkcji bibliotecznej `idt` liczącej całkę. Ma ona co najmniej jeden parametr, którym jest funkcja podcałkowa. W naszym przypadku do obliczenia prędkości poprzez całkowanie przyspieszenia AX użyjemy źródła B1, a do obliczenia drogi skorzystamy z analogicznego źródła B3 całkującego tym razem prędkość (węzeł „v1”). Sygnał ze źródła B3 opisany etykietą „s1” jest odległością, którą pokonał wózek. Wynik symulacji został pokazany na **rysunku 12**.

Dystans, jaki pokonał wózek, można wyznaczyć, mierząc końcową wartość na wykresie `v(s1)`. Ustawiamy więc kursor na końcu tego wykresu i odczytujemy wynik: 2,63. Jednostka „wolty” wynika oczywiście z zastosowania w symulacji źródła napięciowego. Wyniki z akcelerometru są podawane w  $m/s^2$ , mamy więc proste przełożenie metrów na wolty. Ostatecznie możemy więc stwierdzić, że wózek przejechał 2,63 metra. Pozostaje jeszcze weryfikacja wyniku z rzeczywistym pomiarem odległości i... okazuje się, że zmierzony taśmą dystans to 2,65 m. To wręcz niewiarygodna zgodność... Ciesząc się z takiego wyniku, nie mogę jednak odnieść wrażenia, że to chyba jednak zbieg jakichś wyjątkowo korzystnych okoliczności.

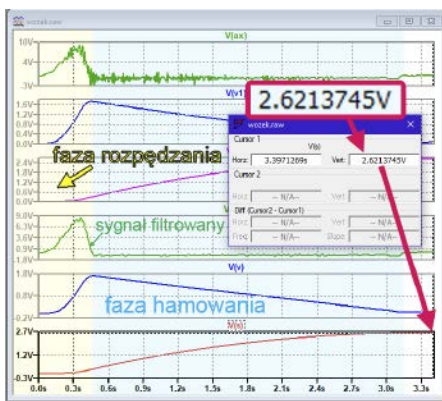
W tej sytuacji wracamy do pytania zadanego na wstępie: czy konieczna jest filtracja sygnału akcelerometru? Trudno przypuszczać, abyśmy uzyskali tą metodą lepszy wynik. Dla formalności jednak zastosujemy filtr. Ustalamy częstotliwość graniczną 20 Hz i rząd filtru równy 2. Wyniki przedstawiono na **rysunku 13**. Wniosek jest dość zaskakujący. Okazuje się, że można zrezygnować z filtrowania sygnału.

## Wykresy

Obsługę wykresów powinniśmy już w zasadzie dobrze znać. Jest jednak kilka zagadnień, które warto zebrać w jednym miejscu. Wiemy już, że w jednym oknie wykresu można umieszczać kilka paneli, a każdy z nich może zawierać pogrupowane tematycznie wykresy. Dopuszczalne



**Rysunek 12. Wyniki symulacji koncepcyjnego układu nawigacji bezwładnościowej bez filtrowania sygnału**



**Rysunek 13. Wyniki symulacji koncepcyjnego układu nawigacji bezwładnościowej z zastosowaniem filtru 2. rzędu o częstotliwości granicznej 20 Hz**

jest kopiowanie wykresów z poszczególnych węzłów obwodu do kilku, a nawet wszystkich paneli. Panel dodaje się, np. klikając prawym przyciskiem myszki w oknie wykresu, a następnie wybierając polecenie Add Plot Pane. Autorzy programu przyjęli, że nowy panel pojawia się u góry okna. Mnie osobiście bardzo to przeszkadza. Zwykle operacja ta jest wykonywana po to, by wprowadzić jakiś dodatkowy wykres, który nie jest najważniejszy w całej kolekcji, a jako ten mniej ważny powinien znaleźć się – moim zdaniem – na dalszej pozycji. To oczywiście bardzo subiektywna ocena. Szkoda, że nie ma możliwości ustalania jakiejś opcji w parametrach symulatora, która dawałaby pewien manewr w tym zakresie. Tymczasem chyba nie ma możliwości zmiany kolejności paneli, pozostaje więc ręczne przerzucanie wykresów. Czynność jest prosta, ale na skutek różnych reakcji symulatora na naciskanie poszczególnych przycisków, czy to klawiatury komputera lub myszki, należy zachować odpowiednią kolejność ich naciskania.

Aby przenieść dany wykres z jednego panelu do innego, należy najechać kursorem myszki na nazwę wykresu widoczną nad nim, nacisnąć lewy przycisk myszki i trzymając go, przenieść kursor dożądanego panela. W chwili zwolnienia przycisku wykres zostanie przeniesiony. Czasami chcemy pozostawić oryginał w pierwotnym panelu, a dany wykres tylko skopiować do innego. W tym przypadku, po naciśnięciu lewego przycisku myszki, dodatkowo naciskamy przycisk `Ctrl` na klawiaturze. Powoduje to dodanie znaku plus ma kursorze myszki, co wyraźnie sygnalizuje zamiar wykonania kopii wykresu.

Co zrobić, jeśli chcemy usunąć jakiś wykres, a nawet cały panel? Do kasowania wszelkich elementów przewidziano użycie przycisku `F5`. Jego naciśnięcie powoduje zmianę wyglądu kursora myszki, który wybiera teraz postać nożyczek. Naprowadzenie kursora na nazwę wykresu (w górnej części panelu) i kliknięcie lewym przyciskiem myszki powoduje usunięcie wykresu. Cały panel wraz z jego zawartością usuwamy, klikając prawym przyciskiem myszki w obszarze wykresu, a następnie wybierając komendę Delete this Pane. Na szczęście operacja jest odwracalna dzięki komendzie Undo (ikonka zawiniętej strzałki w lewo na pasku narzędziowym).

Czasami konieczne jest powiększenie jakiegoś fragmentu wykresu. Intuicyjnie korzystamy z ikonki lupy ze znakiem `+`. Możemy wówczas otoczyć interesujący nas fragment wykresu, korzystając z lewego przycisku myszki. Bardzo często, szczególnie wtedy, gdy wycinamy jakiś wąski przedział czasowy, oś pionowa ulega przeskalowaniu. W rezultacie wykres wygląda tak, jakby uległ spłaszczeniu, a więc skala pionowa nie jest wykorzystywana w pełni. Do optymalnego rozciągnięcia wykresu w pionie służy ikonka z wizerunkiem wykresu z dwiema strzałkami pionowymi (**rysunek 14**). Przydatną, chociaż nieco dziwnie działającą czynnością jest przesuwanie wykresu w dowolnym kierunku wewnątrz panelu. W tym celu należy naprowadzić kursora myszki na wykres, nacisnąć przycisk `Ctrl` i poruszając myszką, ustawić wykres w żądanym położeniu. Po zwolnieniu przycisku wykres jest jednak odświeżany z minimalną zmianą położenia. Jest to efekt niepożądany i nieco irytujący.

Symulując układy z dużą liczbą elementów i połączeń, często chcemy sprawdzić wzajemne zależności między poszczególnymi sygnałami. Tworzymy więc wykresy z kilkoma panelami, a w każdym umieszczamy po kilka przebiegów. Coraz trudniej przychodzi nam orientowanie się, który wykres należy do którego węzła obwodu. Można sobie radzić, nadając etykiety poszczególnym sygnałom. Bywa, że schemat jest tak gęsty, że nie ma już miejsca na etykietę, albo traci on na czytelności z powodu nadmiaru symboli. W takich przypadkach dobrym trikiem jest naprowadzenie kursora myszy na nazwę interesującego nas wykresu (widoczną nad panelem), naciśnięcie lewego `Alt` i kliknięcie lewym przyciskiem myszki. Wskazany węzeł lub element zostaje podświetlony (standardowo na żółto) – **rysunek 15**.

I jeszcze jeden trik związany z wykresami i opcjonalnymi pomiarami, które można za ich pomocą wykonać. I tym razem korzystamy z lewego przycisku `Alt`, po naciśnięciu którego wskazujemy jakiś element na schemacie i klikamy lewym przyciskiem myszki. Pojawia się wówczas dodatkowy wykres mocy wydzielanej na tym elemencie. Jest to wykres

czasowy, a więc widzimy wartości chwilowe tej mocy. Po naciśnięciu *Ctrl* i kliknięciu na nazwę wykresu pojawia się okno, w którym jest wyświetlana moc średnia oraz energia wydzielona w elemencie w przedziale czasowym obejmowanym przez wykres.

Na zakończenie rozważań na temat wykresów warto wspomnieć o możliwości niezależnego skalowania osi czasu w poszczególnych panelach. Zwykle chcemy, aby wszystkie wykresy były wyświetlane wspólnie. Pozwala to obserwować wzajemne powiązania między poszczególnymi sygnałami. Są jednak wyjątkowe sytuacje, których niezależne skalowanie może być przydatne. Przykładem niech będzie multiwibrator przedstawiony na schemacie z **rysunku 16**. Generuje on przebieg prostokątny dostępny na kolektorach obu tranzystorów. Można go obejrzeć w dłuższym czasie, co pozwoli np. przeprowadzić analizę widmową. Jednak w układzie można dostrzec stan nieustalony w chwili jego włączania, któremu warto przyjrzeć się z bliska, nie tracąc przy tym wglądu w całość. Znałe już nam zoomowanie wykresu za pomocą lupy obejmuje wszystkie wykresy, we wszystkich panelach. Korzystając z tego narzędzia, utracilibyśmy zatem możliwość podglądu całego przebiegu. Aby na jednym wykresie wyświetlić wybrany przebieg w powiększeniu, musimy kliknąć prawym przyciskiem myszki na wykresie i wyłączyć opcję *Sync.Horiz.Axes*. Następnie otwieramy nowy panel i umieszczamy w nim kopię przebiegu wybranego do powiększenia. Teraz klikając na nim prawym przyciskiem myszki, wybieramy komendę: *View → Manual Limits*, i w polach *Horizontal Axis* wprowadzamy odpowiednie parametry czasowe, które zdefiniują nam okno powiększenia. Niestety, od tego momentu nie jest możliwe stosowanie narzędzia „lupa”.

### Nieudokumentowane polecenia LTspice

Takie nieudokumentowane polecenia to prawdziwy kłosek dla użytkowników. Znając je i umiejętnie wykorzystując, stawiamy się – to nic, że we własnym mniemaniu – na pozycji wybitnych fachowców od tematu. Potrafimy przecież robić to, o czym inni użytkownicy nie mają pojęcia.

Kilka takich sztuczek było już użytych w niniejszym cyklu, choć nie było to specjalnie nagłaśniane. Okazuje się, że takich nieudokumentowanych poleceń jest dość dużo. Ciekawe, dlaczego autorzy programu nie zawarli ich w opisie programu? Może nie dają gwarancji na poprawne ich działanie? Kto wie, może pod jakąś kombinacją klawiszy ukryty jest jakiś symulator lotu, jak to było w jednej z pierwszych wersji Excela. Takie polecenia są mniej lub bardziej potrzebne, ale warto je znać. Wymienimy więc tylko niektóre.

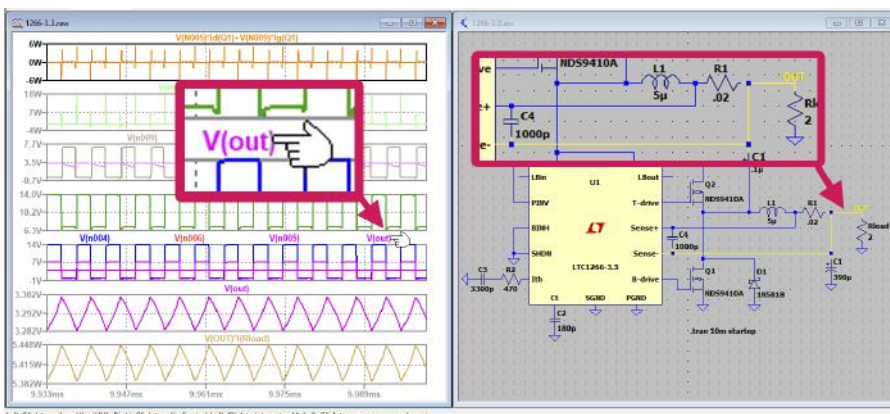
1. Etykieta z negacją. To polecenie jest szczególnie przydatne podczas symulacji układów cyfrowych. Bardzo często występują w nich sygnały proste i zanegowane. Sygnały zanegowane są oznaczane zwykle nadkreślonym napisem tak, jak to pokazano na **rysunku 17**. Efekt taki jest uzyskiwany przez dodanie znaku podkreślenia przed tekstem etykiety.
2. Wyświetlanie ukrytych parametrów elementów. Elementy takie, jak kondensatory czy cewki mają szereg dodatkowych parametrów, które nie są standardowo wyświetlane na schemacie, aby nie zmniejszać jego czytelności. Na przykład dla kondensatora są to: pojemność (to oczywiste), napięcie znamionowe, prąd znamionowy, ekwiwalentna rezystancja szeregową (ESR), ekwiwalentna indukcyjność szeregową, ekwiwalentna rezystancja równoległa i ekwiwalentna

pojemność równoległa. Parametry te można sprawdzić, wyświetlając właściwości elementu (kliknięcie prawym przyciskiem myszki na elemencie). Wyświetlana jest wówczas tabelka z parametrami. Podobne informacje można uzyskać, naprowadzając kursor myszki na dany kondensator i naciskając jednocześnie przyciski *Shift-Ctrl-Alt-H*. Wszystkie zdefiniowane dla tego elementu ukryte parametry są wyświetlone na schemacie standardowo żółtą czcionką w pobliżu oznaczenia schematowego (**rysunek 18**). Domyślny, żółty kolor nie jest najlepszym rozwiązaniem, tym bardziej że zastosowano najmniejszą czcionkę.

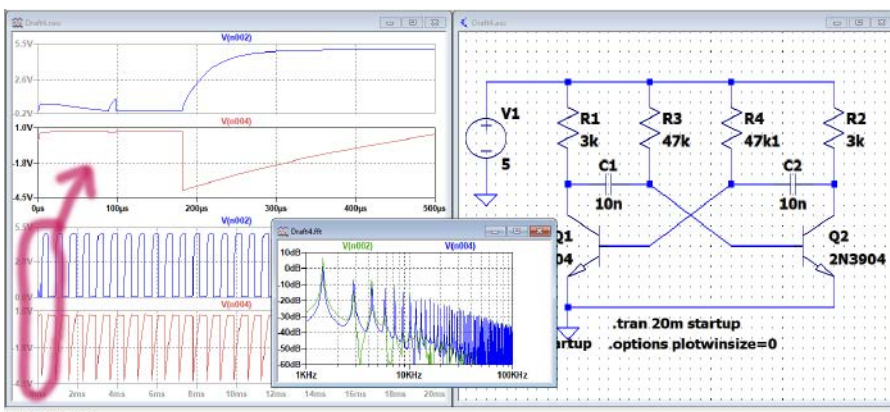
3. Generowanie przebiegów definiowanych numerycznie w pętli skończonej lub nieskończonej. Z tej cechy już korzystaliśmy, ale warto



Rysunek 14. Ikona rozciągania wykresu w pionie

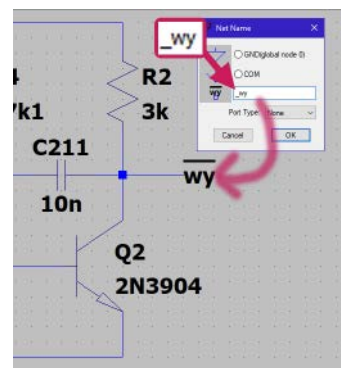


Rysunek 15. Ilustracja metody podświetlenia węzła na schemacie, do którego odnosi się wskazany przebieg na wykresach



Rysunek 16. Przykład stosowania niezależnego skalowania osi czasu w różnych panelach na wykresie

ją przytoczyć jeszcze raz tutaj. Jak wiemy, użytkownik może wygenerować niemal dowolny własny przebieg, korzystając np. ze źródła napięciowego z opcją *PWL*. Dane numeryczne przebiegu są zapisywane w wierszach zawierających parametry **czas, napięcie**. Dane te, jeśli jest ich niewiele i można je wprowadzić ręcznie, najwygodniej jest zapisać w wewnętrznej tablicy symulatora. Gdy robi



Rysunek 17. Ilustracja metody wprowadzania etykiet ze znakiem negacji

się ich zbyt dużo, można je zachowywać w plikach dyskowych w formacie tekstowym.

W poniższym przykładzie zapisano w ten sposób krótkie sygnały sinusoidalne, które wykorzystamy do generowania tekstu zapisanego alfabetem Morse'a. Kropka trwa 150 ms, kreska 450 ms, odstęp między elementami znaku trwa 150 ms, a odstęp między znakami trwa 450 ms. Do generowania telegramu zastosujemy źródło napięciowe V1 z opcją PWL. Argumentami będą pliki z zapisanymi kropkami i kreskami połączone za pomocą nieudokumentowanej opcji REPEAT ENDREPEAT. Naszym zadaniem jest nadanie komunikatu „AVT” za pomocą dźwiękowych znaków Morse'a. Dla sprawdzenia efektu końcowego zastosujemy komendę `.wave avt.wav 16 44.1k v(wy)` zapisującą efekt pracy naszego automatycznego klucza telegraficznego w pliku dźwiękowym `avt.wav`. Definicja sygnału generowanego przez źródło V1 jest następująca:

```
PWL repeat for 1(file=kropka.txt)endrepeat repeat
for 1(file=kreska.txt)
endrepeat +450m 0 repeat for
3(file=kropka.txt)endrepeat
repeat for 1(file=kreska.txt)
endrepeat +450m 0 repeat for
1(file=kreska.txt)endrepeat
```

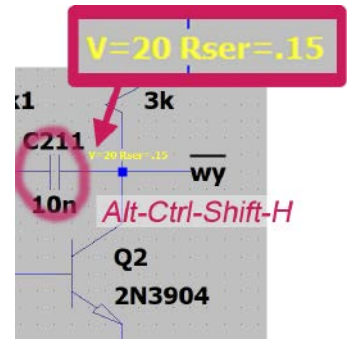
Jak widać, w powyższym zapisie łączone są różne techniki określania parametrów. Opcje REPEAT mogą być ponadto zagnieżdżane. Na **rysunku 19** pokazano wygenerowany przebieg w całości i jeden powiększony jego fragment. Komunikat można odsłuchać z pliku `avt.wav`.

4. Renumeracja elementów. Niezwykle przydatna funkcja dla użytkowników lubiących porządek na schemacie. W trakcie projektowania często usuwamy ze schematu różne elementy, a zamiast nich wstawiamy inne. LTspice automatycznie numeruje każdy wstawiany do schematu element, nadając mu kolejny numer. Przy wielokrotnie powtarzanej operacji zamiany elementów tworzą się więc dziury w numeracji. Naciskając kombinację przycisków `Shift-Ctrl-Alt-R`, wymuszamy renumerację elementów od zera. Prosty zabieg, a jakże przydatny.
5. Symulacja wymiany typów elementów. Załóżmy, że projektujemy wzmacniacz tranzystorowy. Punkty pracy wyznaczone, napięcia się zgadzają, ale parametry częstotliwościowe zwykle nie są łatwe do obliczenia. Wymagają wielu obliczeń, a i tak trudno uwzględnić wszystkie elementy, które mogą wpływać na charakterystykę częstotliwościową. Jedna mała pojemność złącza kolektor-baza zastosowanego tranzystora może spowodować, że cały projekt nadaje się do kosza. Ratunkiem jest symulacja metodą prób i błędów. Po prostu bierzemy kolejne typy tranzystorów i patrzymy, który z nich nadaje się do projektu, a który nie. Wspaniale byłoby móc porównywać np. charakterystyki układu z poszczególnymi typami tranzystorów, ale trzeba by tworzyć kilka niemal identycznych schematów różniących się jedynie typami tranzystorów. Jest prostsza metoda polegająca na zastosowaniu polecenia `.step param`, którego argumentem byłby typ tranzystora. Problem jednak polega na tym, że polecenie `param` nie dopuszcza użycia nienumerycznych argumentów. Na szczęście w symulacjach dozwolone jest używanie aliasów. Będziemy więc w nich ukrywać alfanumeryczne oznaczenia elementów. Alias poprzedzamy tekstem „ako:”, po którym następuje biblioteczny typ tranzystora. W przykładzie użyjemy tranzystorów:

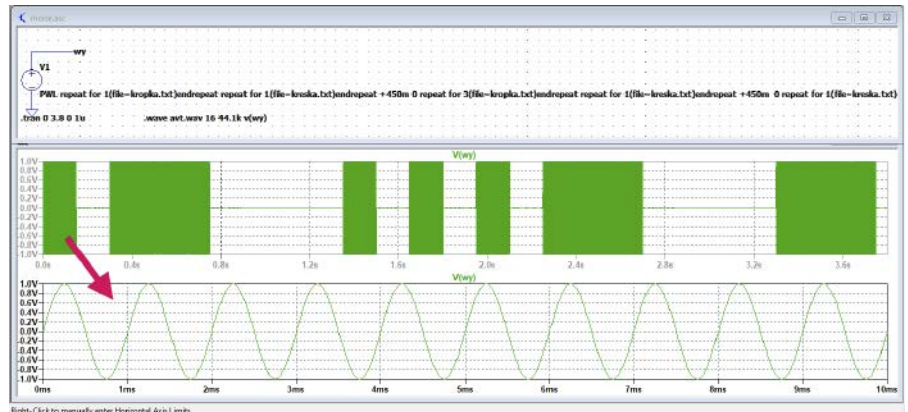
2N3904, MAT02 (oznaczenie w bibliotece MAT-2 – jest to para tranzystorów w jednej obudowie) oraz tranzystorów BC547B i BC547C. Zaczynamy od przygotowania aliasów:

```
.model 3904 ako:2N3904
.model 2 ako:MAT-02
.model 5472 ako:BC547B
.model 5473 ako:BC547C
```

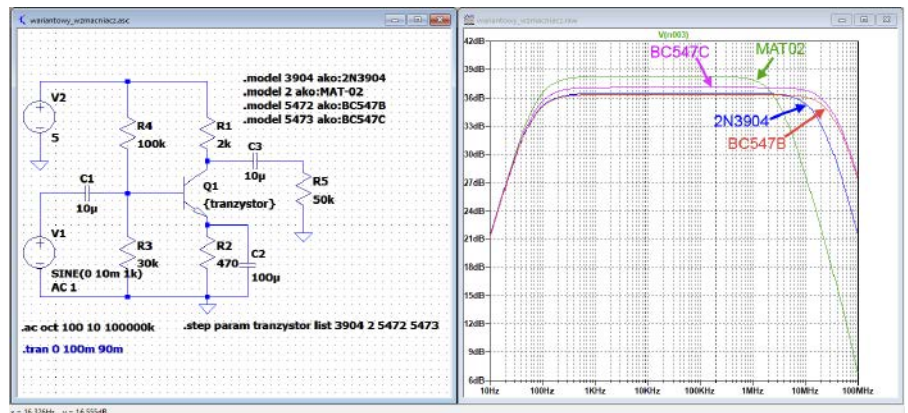
Nadajemy parametryczny typ tranzystora na schemacie, np.:



**Rysunek 18.** Wyświetlanie ukrytych parametrów elementów



**Rysunek 19.** Przebieg generowany przez automatyczny klucz telegrafu Morse'a



**Rysunek 20.** Wariantowe charakterystyki wzmacniacza zależne od zastosowanych typów tranzystorów

{tranzystor}. Musimy jeszcze wstawić odpowiednio zredagowane polecenie wymiany tranzystorów:

```
.step param tranzystor list 3904 2 5472 5473
```

Wszystko jest gotowe, rozpoczynamy symulację. Zbadamy charakterystykę częstotliwościową naszego wariantowego wzmacniacza. Jest ona pokazana wraz ze schematem na **rysunku 20**. Wszystko stało się jasne. Musimy tylko zdecydować, czy bardziej nam zależy na wzmocnieniu, czy na paśmie.

### Czas leci...

Ten odcinek ukazuje się dokładnie po dwóch latach od zapowiedzi cyklu. To czas, po którym może już występować „zmęczenie materiału”. Na temat symulatora LTspice można by napisać jeszcze wiele, ale podjąłem decyzję o zakończeniu kursu. Ostatni odcinek ukaże się za miesiąc. Nie wykluczam powrotu do pewnych tematów, jeśli uznam, że są warte opublikowania. Będą to jednak już tylko pojedyncze artykuły.

Jarostaw Doliński, EP