

Sisteme cu F. P. G. A. și D. S. P. – Prezentarea introductivă a mediului Altair – Embed –

I. INTRODUCERE:

Mediul de simulare, testare și programare Altair Embed, regăsit și sub denumirile SolidThinking Embed sau la origine VisSim, reprezintă o soluție tehnologică în lanțul de concepere, simulare și testare apoi producție finală a unui prototip funcțional. Spre exemplu, în domeniul Ingineriei Electrice, mediul Altair Embed poate fi utilizat pentru a rezolva probleme cu specific ingineresc, legate de:

- proiectarea circuitelor electrice clasice (ex. filtre, atenuatoare etc...);
- proiectarea circuitelor electronice (ex. amplificatoare, convertoare de putere etc...);
- proiectarea codului program pentru aplicații micro – programabile;
- proiectarea buclor și strategiilor complexe de control;
- proiectarea algoritmilor de analiză și procesare audio;
- proiectarea algoritmilor de analiză și procesare video;

Domeniul de aplicabilitate al mediului Altair Embed NU se restrânge doar la cel al Ingineriei Electrice, ci are un caracter pronunțat inter – disciplinar. Capacitatea de a rezolva probleme complexe de inginerie cu ajutorul ecuațiilor liniare, diferențiale și transcendente, conferă mediului Altair Embed deschiderea înspre celelalte domenii tehnice sau ingineresti de aplicabilitate decât cel al Ingineriei Electrice.

De asemenea, faptul că, în cadrul acestui mediu, au fost integrate o sumedenie de soluții în ceea ce privește achiziția, procesarea și condiționarea semnalelor, adică, au fost introduse modalități și metode de programare ale unor platforme de dezvoltare cu DSP, micro-controller, microprocesor, plăci de achiziție și alte dispozitive programabile, oferă de asemenea un plus de valoare față de programele clasice de calcul și simulare. Două dintre avantajele tehnologice majore implementate în cadrul mediului Altair Embed, la ora actuală, le reprezintă generarea automată de cod și comunicarea în timp real cu platformele DSP de la Texas Instruments prin interfața JTAG, cu ajutorul unui protocol de comunicație propriu denumit HotLink, un derivat al protocolului Real – Time Data eXchange.

Din punct de vedere istoric, VisSim, a apărut pentru prima dată în anul 1989 ca și un produs al companiei Visual Solutions din Westford, Massachusetts SUA. În luna August din anul 2014, compania Visual Solutions a fost cumpărată de către compania Altair, care a redenumit produsul sub numele de Altair Embed. Numele de SolidThinking Embed a fost atribuit în perioada de tranziție 2014 – 2016, după care, programul a funcționat și funcționează până în prezent sub numele de Altair Embed.

Din punct de vedere al aplicabilității în mediul didactic sau universitar, atât Visual Solutions cât și Altair, au susținut programul academic de licențiere gratuită a mediului Altair Embed, cu scopul de a încuraja studenții și cadrele didactice să utilizeze acest mediu. Pe lângă faptul că a fost promovată o politică de licențiere benefică mediului academic, totodată au fost introduse, pe lângă platformele de dezvoltare cu DSP de la Texas Instruments, și alte platforme de dezvoltare accesibile ca preț dar și populare în spațiul academic, precum Arduino, RaspberryPI, plăci de achiziție și echipamente de la National Instruments, camere web etc...

II. IMPLEMENTAREA MODELELOR ÎN MEDIUL ALTAIR EMBED:

În cadrul mediului Altair Embed există ne-numărate metode de implementare pe baza aparatului matematic al ecuațiilor liniare, matriceale, diferențiale sau transcendente. Pe lângă aparatul matematic standard, în cadrul mediului Altair Embed, există și algoritmi specifici programării, precum bucle iterative, sisteme condițional-decizionale, operații logice și operații la nivel de bit (concatenare, deplasare la stânga sau la dreapta etc...). O altă gamă largă de instrumente o constituie paleta cu operații și algoritmi specifici teoriei sistemelor de reglare automată sau a teoriei analizei de semnal (grafice, analizoare de spectru, operații și transformări de coordonate și domenii, surse de semnal, operatori statistici etc...).

De asemenea mediul Altair Embed, permite implementarea modelelor, atât pentru execuția lor pe calculatorul gazdă, cât și pentru execuția lor, la nivelul procesoarelor dedicate ale platformei de dezvoltare (ex. DSP). În acest sens, există două direcții de abordare ale aparatului matematic:

- abordarea modelului pe bază de numere cu virgulă mobilă (eng. floating point);
- abordarea modelului pe bază de numere cu virgulă fixă (eng. fixed point);

Din punct de vedere al optimizării algoritmului pentru a rula pe un procesor cu resurse de calcul relativ limitate se recomandă abordarea metodei de reprezentare cu virgulă fixă. Astfel, algoritmul de control care urmează a fi implementat la nivelul unui procesor digital de semnal, trebuie să fie proiectat cu ajutorul operațiilor specifice reprezentării în virgulă fixă!

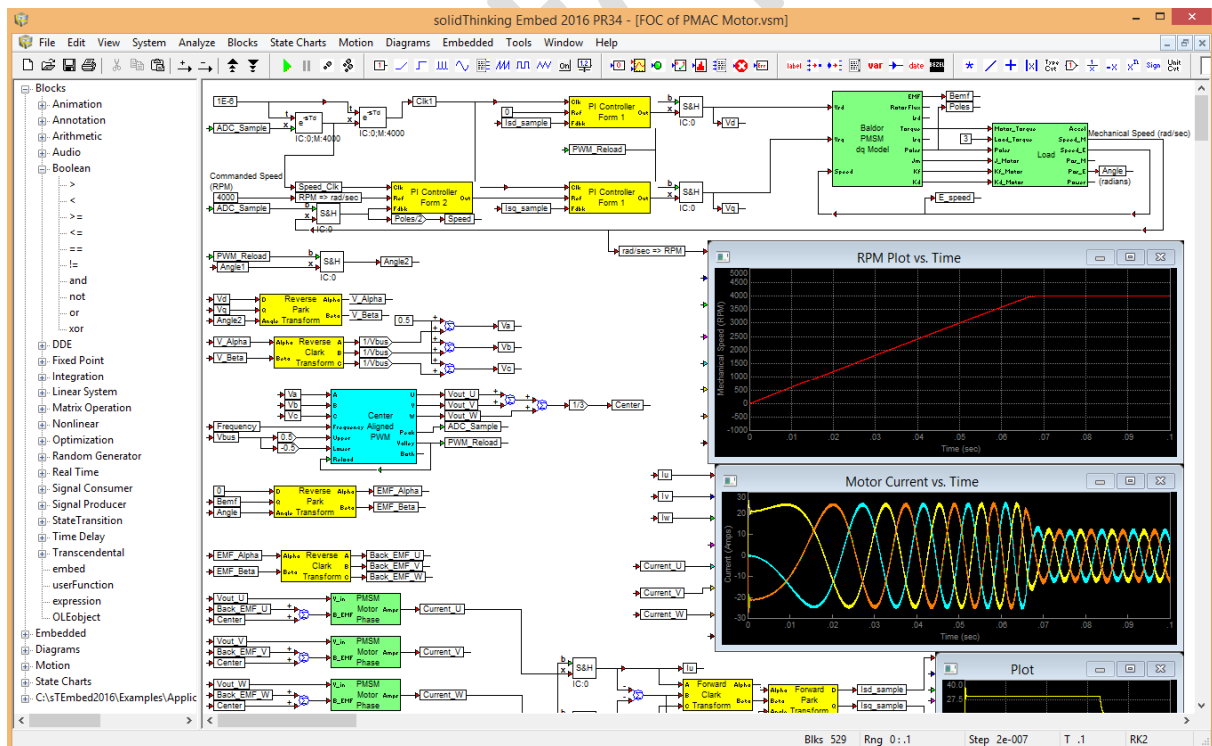


Fig. 1 – Modul de reprezentare al unui model în cadrul mediului Altair Embed

În cazul de față, vor fi analizate doar situațiile în care, se dorește realizarea unui model matematic de simulare la nivelul calculatorului gazdă cu o reprezentare în virgulă mobilă.

Pentru o mai bună cunoaștere a modului de funcționare al mediului Altair Embed, se propune analiza unor modele matematice simple de implementat:

- modelul unui generator sinusoidal monofazat cu parametri variabili;
- modelul unui generator sinusoidal trifazat cu parametri variabili;

A. IMPLEMENTAREA GENERATORULUI SINUSOIDAL MONOFAZAT CU PARAMETRII VARIABILI:

Pentru a implementa în mod digital un generator sinusoidal cu parametri variabili, trebuie mai întâi să se cunoască ecuația armonică de variație în timp a unei mărimi instantanee, precum o tensiune. Ecuația de variație în timp, general valabilă, a unei tensiuni alternativ - sinusoidală ar putea fi exprimată astfel:

$$u(t) = (U \cdot \sqrt{2} \cdot \sin(\omega \cdot t \pm \varphi)) \pm \Delta U$$

- „ $u(t)$ ” – reprezintă valoarea instantanee a tensiunii la momentul „ t ”;
- „ $U\sqrt{2}$ ” – reprezintă valoarea vârf la vârf a tensiunii;
- „ U ” – reprezintă amplitudinea tensiunii;
- „ \sin ” – reprezintă funcția armonic – trigonometrică a variației în timp;
- „ ω ” – reprezintă pulsația mărimii;
- „ t ” – reprezintă timpul parcurs pentru a descrie variația;
- „ φ ” – reprezintă defazajul dintre două mărimi;
- „ ΔU ” – reprezintă diferența de tensiune față de zero (componenta de curent continuu);

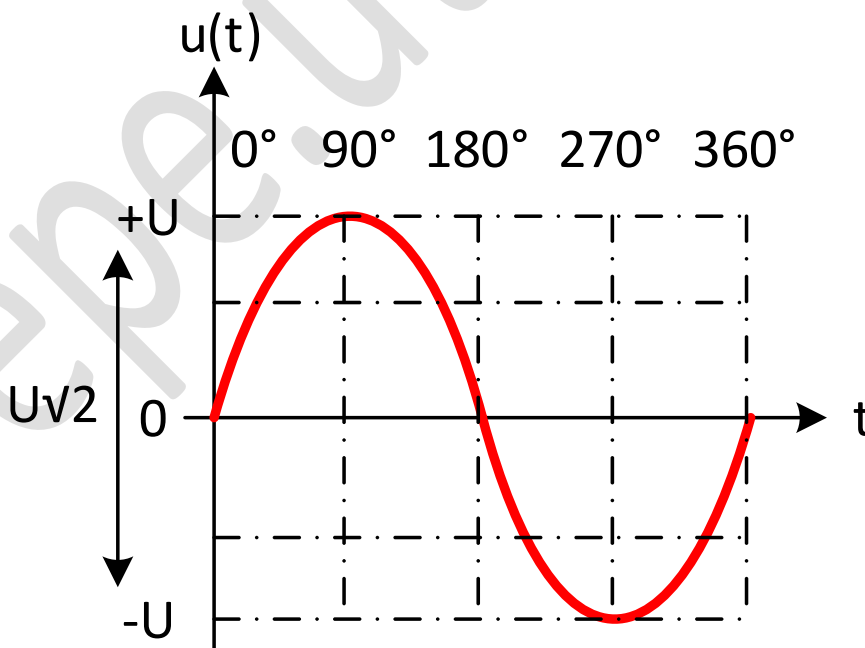


Fig. 2 – Reprezentarea grafică a tensiunii instantanee „ $u(t)$ ”

Pentru a descrie forma de undă a unei tensiuni alternative, este necesară etapa de particularizare a ecuației. Spre exemplu, pentru a obține valoarea instantanee a tensiunii pe durata unui interval de timp, este necesară etapa de discretizare a ecuației prin atribuirea de valori consecutive crescătoare parametrului „t” (timp). Acest lucru poate fi obținut prin intermediul unui semnal de tip rampă „x(t)”, sau dinte de fierăstrău (rampă resetabilă).

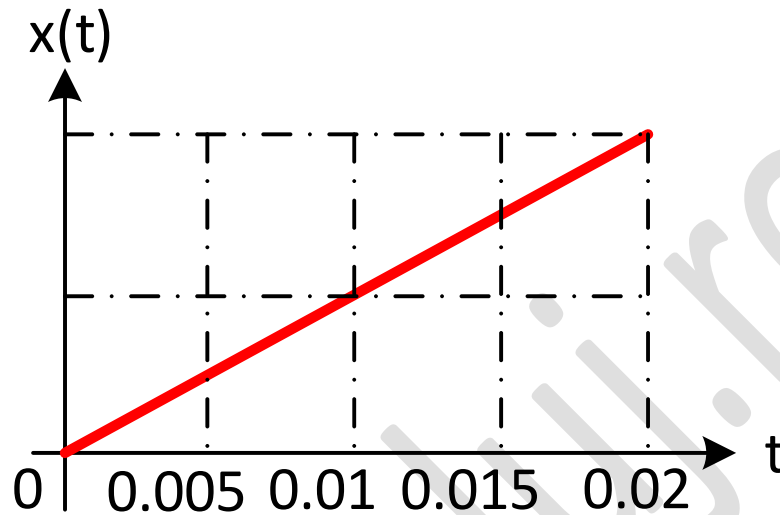


Fig. 3 – Reprezentarea grafică a unui semnal de tip rampă „x(t)” care descrie variația parametrului „t” (timp)

Pasul de creștere al semnalului rampă, este în strânsă legătură atât cu timpul de eșantionare al modelului cât și cu frecvența semnalului descris prin ecuația analitică „u(t)”. Astfel, pentru a descrie în mod complet variația sinusoidală a tensiunii și dependența evoluției acesteia în funcție de frecvență și timp, se apelează la următoarea relație:

$$u(t) = (U \cdot \sqrt{2} \cdot \sin(2 \cdot \pi \cdot f \cdot t \pm \varphi)) \pm \Delta U$$

- termenul „ 2π ” reprezintă echivalentul a 360° , practic, descrierea unei rotații complete;
- „f” – reprezintă frecvența semnalului sinusoidal;

Particularizarea ecuației:

În vederea simulării evoluției în timp a tensiunii de alimentare de la rețeaua publică națională, se vor impune următorii parametrii ecuației „u(t)”:

- amplitudinea tensiunii: $U = 220$ [V];
- frecvența tensiunii: $f = 50$ [Hz];
- defazaj: faza „A”: $\varphi = 0$ [rad]; faza „B”: $\varphi = 2\pi / 3$ [rad]; faza „C”: $\varphi = 4\pi / 3$ [rad];
- componenta de curent continuu: „ ΔU ” = 0;
- timp de simulare: $T = 0.02$ [s];
- timp de eșantionare: $t_s = 10^{-4}$ [s];

În mediul Altair Embed se va inițializa un model nou, după care, se vor realiza următoarele parametrizări inițiale:

Pasul 1 – Crearea unui model de simulare:

- În cadrul mediului Altair Embed, se vor închide oricare alte modele deschise, apoi, din meniul „File” se va alege opțiunea „New”. În urma efectuării operațiilor indicate, se va salva modelul nou creat sub numele spre exemplu „three_phase”.

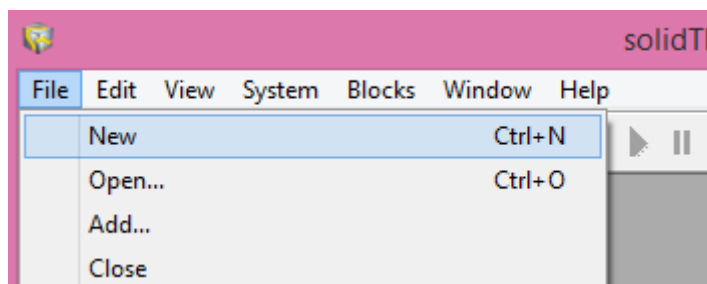


Fig. 4 – Crearea unui model nou

Pasul 2 – Parametrizarea modelului de simulare:

- Din meniul „System” se va alege opțiunea „System Properties...”:

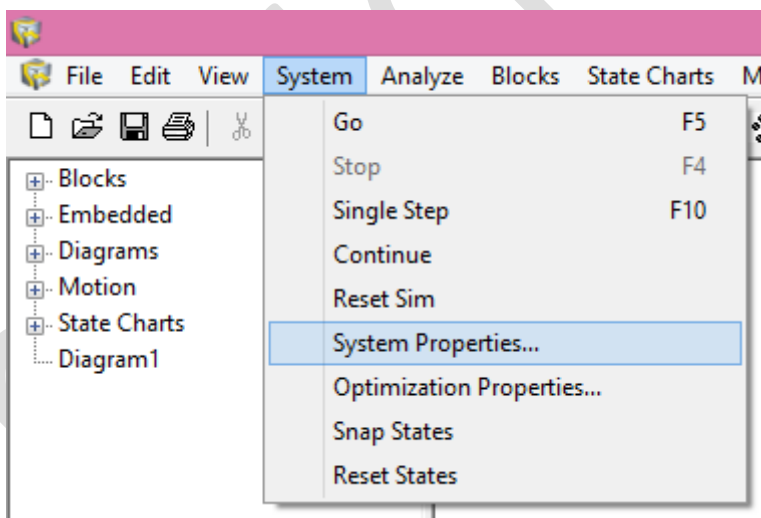


Fig. 5 – Parametrizarea modelului prin intermediul opțiunii „System Properties...”

În fereastra nou deschisă, se vor introduce următorii parametrii:

- Start (sec.): 0;
- Time Step: 1e-4; → Seconds;
- End (sec.): 0.02;
- Se va bifa opțiunea „Run in Real Time”;

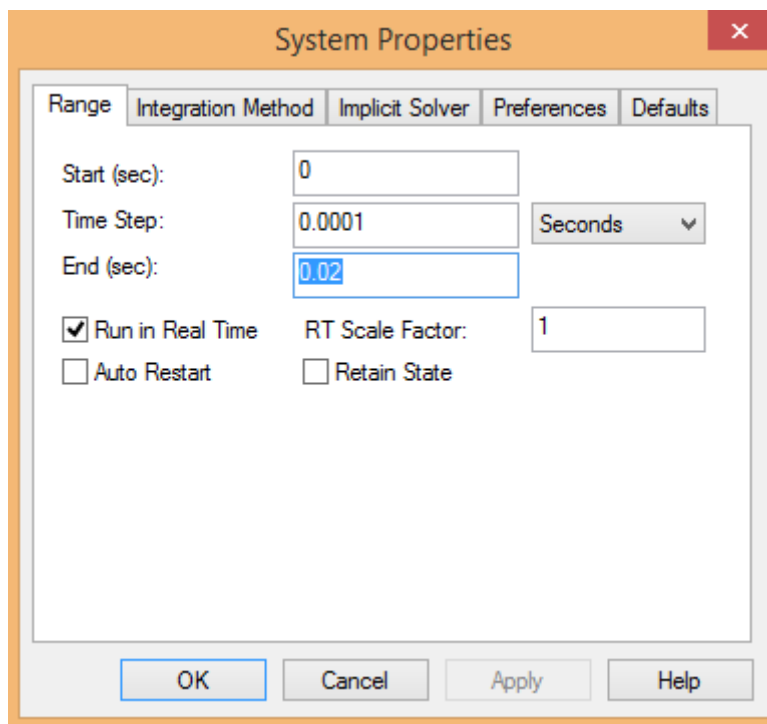


Fig. 6 – Fereastra „System Properties” pentru parametrizarea modelului

Procedura se va finaliza prin apăsarea butonului „Apply” apoi „Ok”.

Pasul 3 – Implementarea modelului unei singure faze pe baza ecuației de variație în timp:

- Cu ajutorul blocurilor din meniul „Blocks” și sub-categoriile „Arithmetic”, „Transcendental”, „Signal Consumer” și „Signal Producer”, se va implementa următorul model:

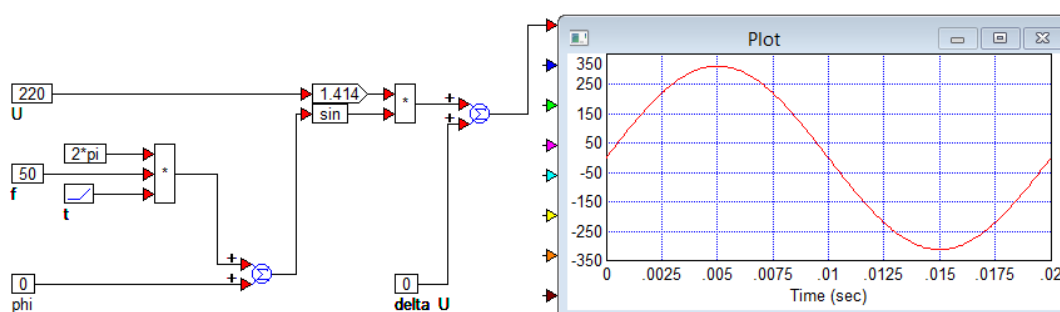


Fig. 7 – Modelul generatorului sinusoidal monofazat

În vederea realizării modelului reprezentat în figura nr. 7, se vor obține următoarele blocuri funcționale: cinci blocuri de tip „constantă”, două blocuri de tip „sumator”, două blocuri de tip „înmulțire”, un bloc de tip „factor de amplificare” (eng. Gain), un bloc de tip

„grafic cu afișare planară 2D”, un bloc de tip „rampă”, un bloc de tip „trigonometric sinus”. Respectivul blocuri se regăsesc în următoarele meniuri și sub-categorii:

Nume sau simbol bloc	Număr de blocuri necesare (eng.)	Nume meniu categorie (eng.)	Nume meniu sub-categorie (eng.)
const	5	Blocks	Signal Producer
summingJunction	2	Blocks	Arithmetic
*	2	Blocks	Arithmetic
gain	1	Blocks	Arithmetic
plot	1	Blocks	Signal Consumer
ramp	1	Blocks	Signal Producer
sin	1	Blocks	Transcendental

Pentru a particulariza un bloc de tip „constantă” spre exemplu, prin intermediul comenzii „click dreapta” asupra blocului, se va deschide o fereastră de particularizare, prin intermediul căreia se vor putea specifica parametrii precum valoarea constantei și simbolul ei.

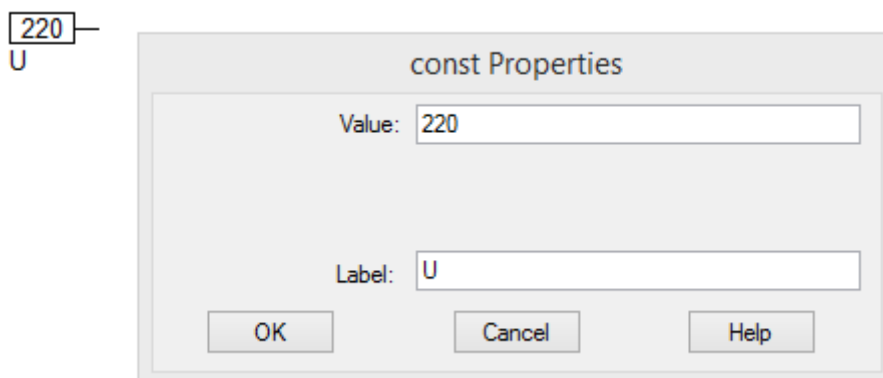


Fig. 8 – Particularizarea blocului de tip „constantă”

OBSERVAȚIE: Simbolul blocului nu este vizibil decât numai după re-poziționat blocului!
Pentru a particulariza un bloc de tip „rampă” se procedează în mod similar.

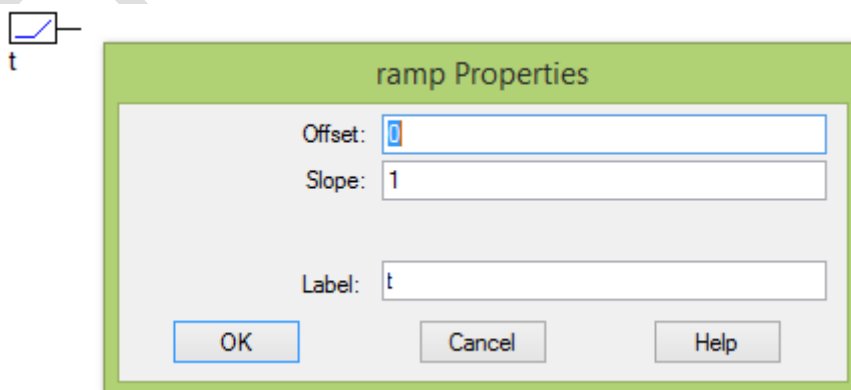


Fig. 9 – Particularizarea blocului de tip „rampă”

Pentru a adăuga un nou conector unui bloc, se utilizează funcția „Add Connector...” din cadrul meniului „Edit”. Cursorul își va schimba forma într-o săgeată având în partea de sus simbolul „+”! Multiplicarea terminalelor se va realiza prin comanda „click stânga” asupra conectorului care se dorește a fi multiplicat. Pentru operația inversă se va utiliza comanda „Remove Connector...” din cadrul aceleiași meniu, procedând în mod similar.



Fig. 10 – Adăugarea unui nou conector

Pentru afișarea corectă a formei de undă a semnalului, în cadrul blocului de afișare grafică „plot” se vor impune următorii parametri în fereastra „Plot Properties” (click dreapta): În categoria „Options”:

- Fixed Bounds;
- Grid Lines;

Iar în categoria „Axis” se vor impune valorile „Y Upper Bound”: 350 și „Y Lower Bound”: -350.

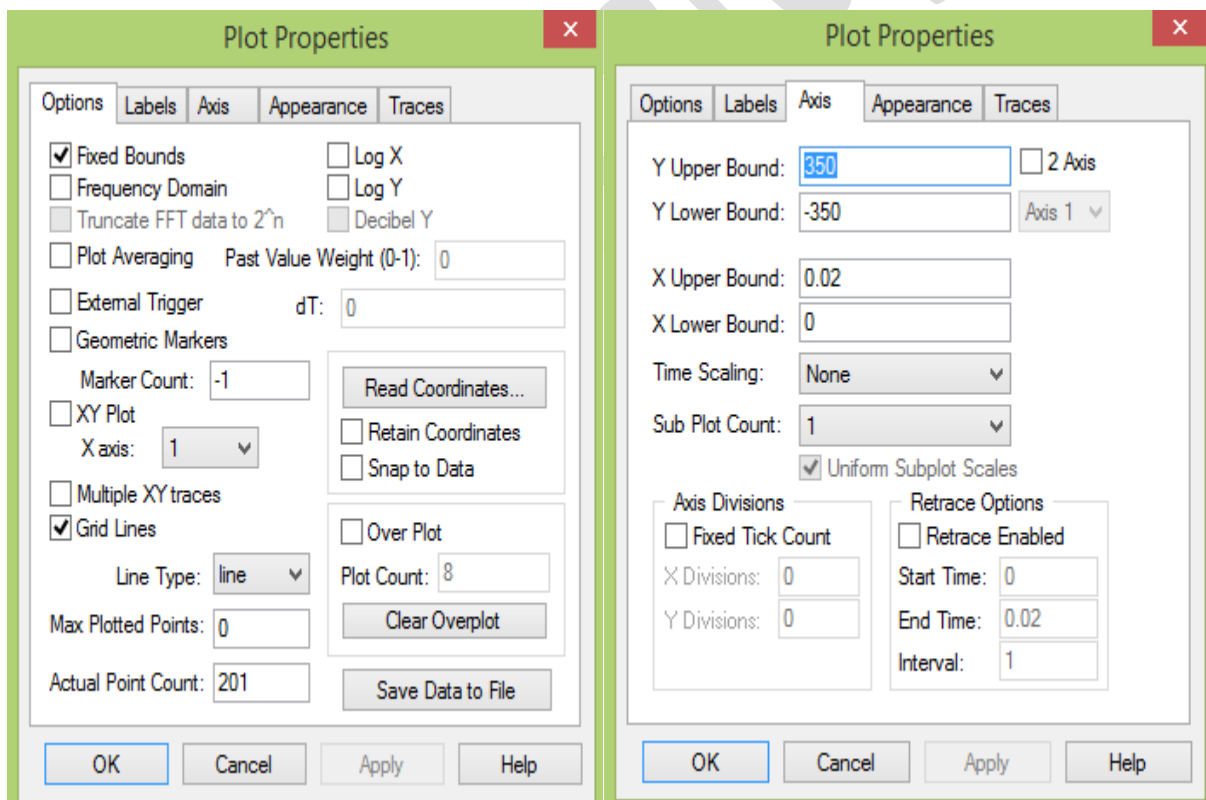


Fig. 11 – Parametrizarea afișajului grafic al formelor de undă

Procedura de parametrizare va fi încheiată prin apăsarea butoanelor „Apply” apoi „Ok”.

Se propun următoarele valori implicite pentru modelul dat:

Parametru / simbol bloc	Valoare
U	220
2 * pi	2 * pi
f	50
„t” panta de creștere (eng. slope)	1
phi	0
gain	1.414 (adică $\sqrt{2}$)
delta_U	0

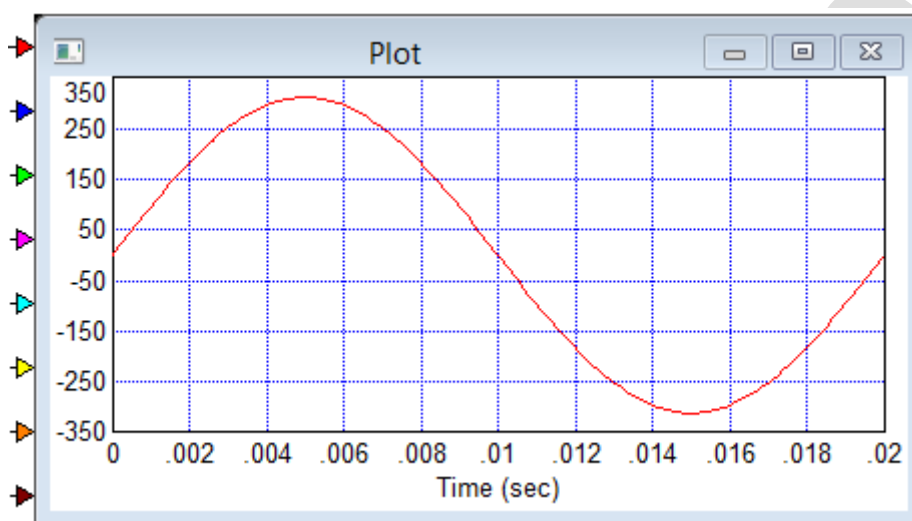


Fig. 12 – Forma de undă a valorii instantanee „u(t)” conform parametrizării constantelor

B. IMPLEMENTAREA GENERATORULUI SINUSOIDAL TRIFAZAT CU PARAMETRII VARIABILI:

Pentru a implementa un model de generator sinusoidal trifazat, simetric, se va utiliza următorul sistem de ecuații, care descriu tensiunea instantanee pentru fiecare fază în parte:

$$\begin{aligned}
 u_A(t) &= (U \cdot \sqrt{2} \cdot \sin(2 \cdot \pi \cdot f \cdot t + 0)) \pm \Delta U \\
 u_B(t) &= \left(U \cdot \sqrt{2} \cdot \sin\left(2 \cdot \pi \cdot f \cdot t + \frac{4 \cdot \pi}{3}\right) \right) \pm \Delta U \\
 u_C(t) &= \left(U \cdot \sqrt{2} \cdot \sin\left(2 \cdot \pi \cdot f \cdot t + \frac{2 \cdot \pi}{3}\right) \right) \pm \Delta U
 \end{aligned}$$

Prin intermediul sistemului de trei ecuații descris mai sus, se vor obține trei forme de undă sinusoidale, defazate la 120° adică „ $2\pi / 3$ ” radiani. Cele trei forme de undă, descriu astfel un sistem trifazat, simetric de tensiuni, cu posibilitatea ajustării parametrilor.

În vederea implementării unui model de generator trifazat, simetric, este necesară utilizarea variabilelor în modelul Altair Embed. Blocul pentru implementarea variabilei se găsește în meniul „Blocks” în sub-categoria „Annotation”, opțiunea „variable”. Cu ajutorul

comenzii „click dreapta”, poate fi definit numele variabilei care se va apela în model. Un aspect important de menționat, este faptul că, blocul variabilei nu trebuie să fie legat fizic în model de alte elemente în continuare, ci el poate fi apelat ori de câte ori este nevoie.

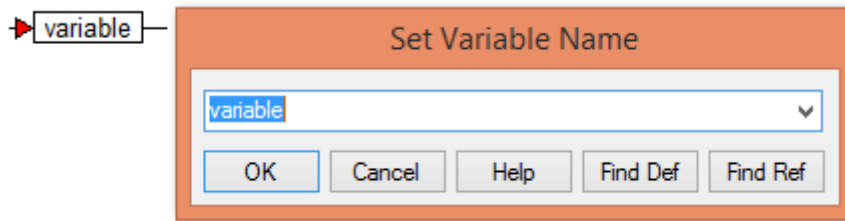
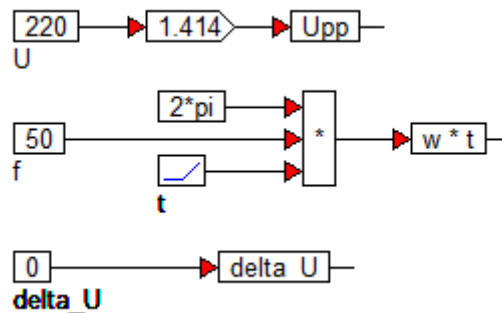
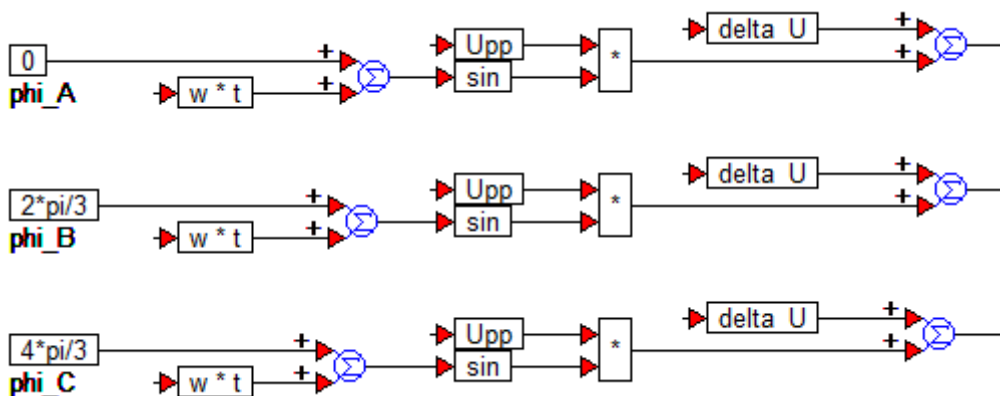


Fig. 13 – Definirea variabilei introduse în model

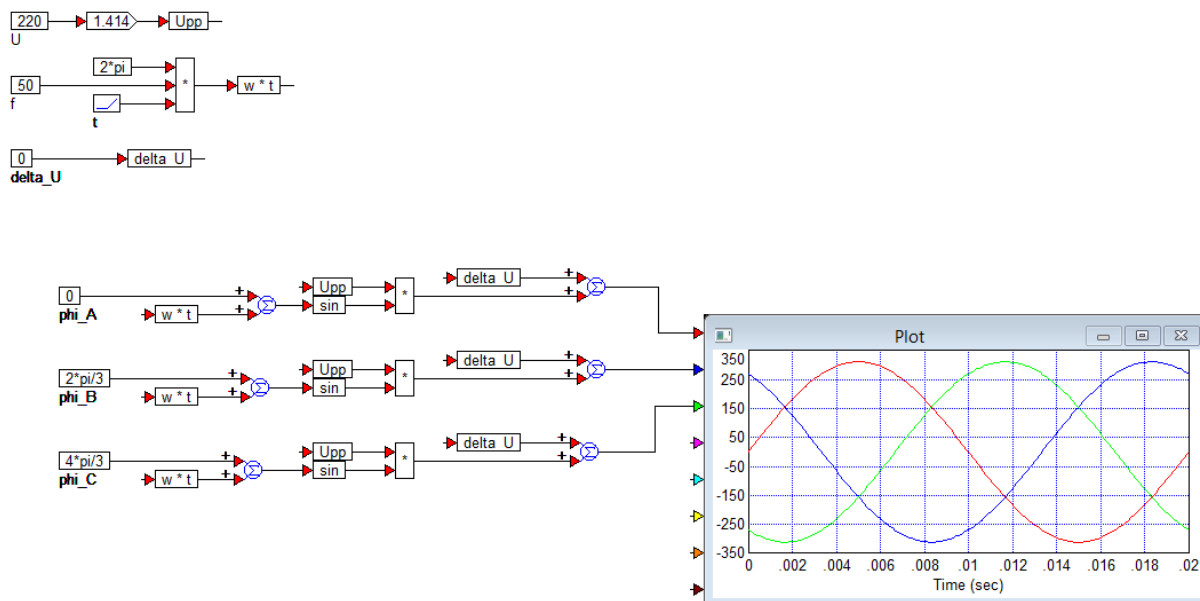
Pe baza acestor blocuri, se va implementa următorul model (format din două părți) în mediul Altair Embed (a se vedea figura „C” pentru întreg modelul). Pentru implementare se vor defini următoarele variabile „Upp” – tensiunea vârf la vârf, „w * t” – pulsația în timp, „delta_U” – componenta de curent continuu, sau decalajul față de zero:



A. – Prima parte a modelului



B. – A doua parte a modelului

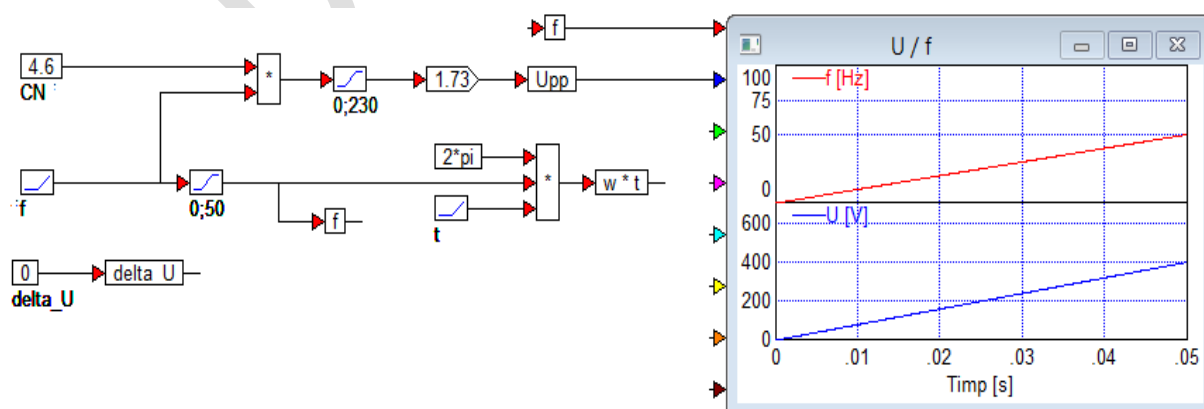


C. – Modelul complet

Fig. 14 – Modelarea sistemului trifazat simetric

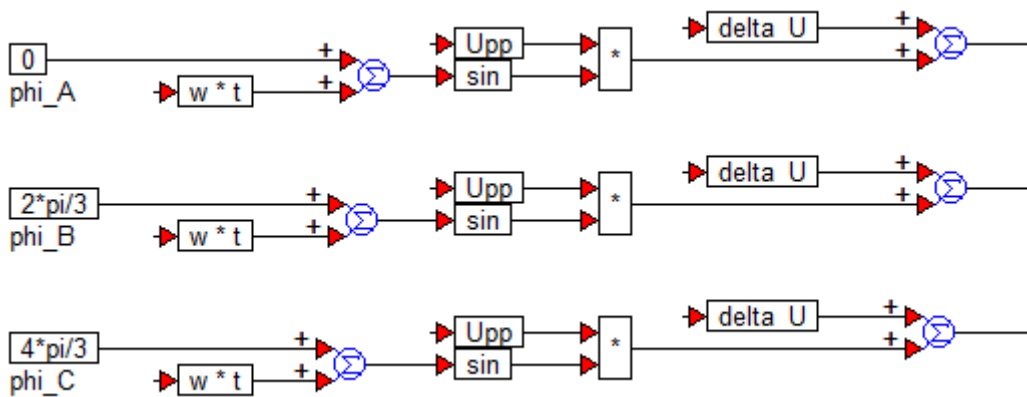
Implementând un astfel de model matematic, la nivelul unei platforme de dezvoltare cu DSP, se vor putea obține în mod „sintetizat”, cele trei tensiuni variabile în timp, dar construite numeric în mod artificial de către convertorul digital – analog al platformei. Semnalul generat va putea fi utilizat ca și un semnal de comandă sau referință, pentru un circuit electronic de putere. În mod similar, se procedează și în cazul invertoarelor trifazate, sau al variatoarelor de frecvență.

Una dintre cele mai bine cunoscute aplicații ale acestui aparat matematic este variatorul de frecvență demarator pentru motoarele de inducție asincrone trifazate, controlate scalar. Mai precis metoda „ $U / f = \text{constant}$ ”. În vederea implementării acestei metode în memoria DSP a unui variator de frecvență se implementează următorul model:

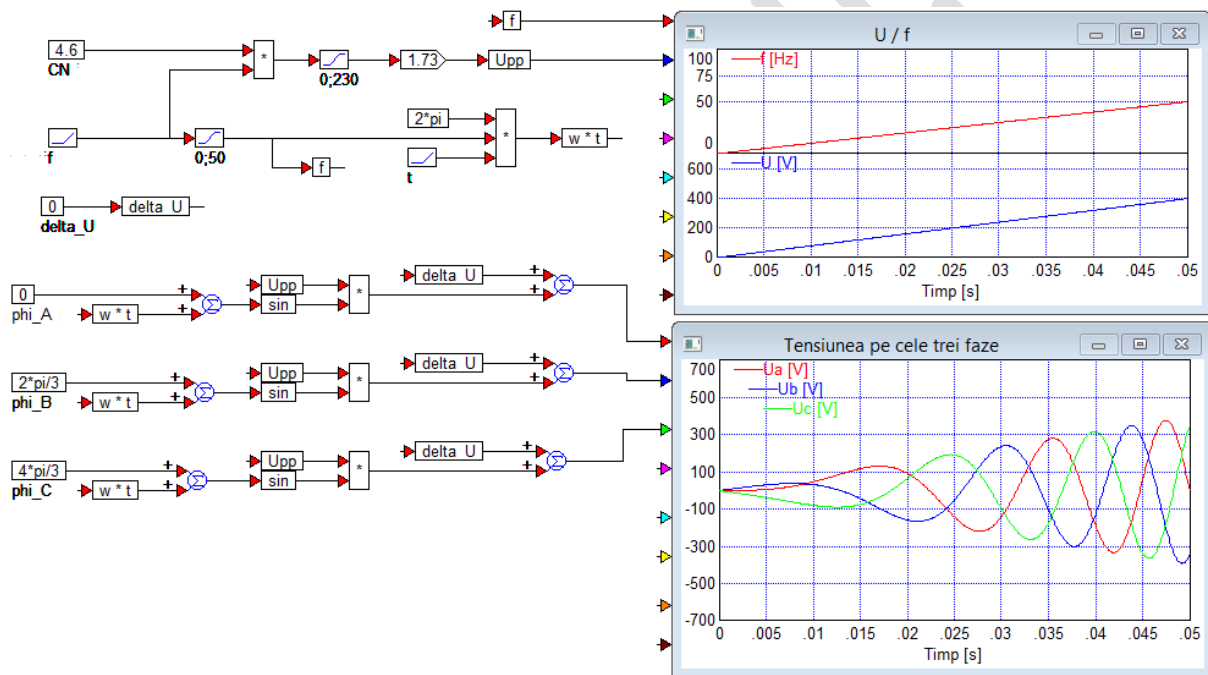


A. – Prima parte a modelului

Pentru a obține rezultatele corecte, timpul total de simulare va trebui modificat, din meniul „System” opțiunea „System Properties...” opțiunea „End (sec)” se va impune valoarea 0.05.



B. – A doua parte a modelului



C. – Modelul final complet

Fig. 15 – Implementarea unui variator de frecvență controlat scalar (metoda $U / f = \text{const.}$)

Modelul matematic implementat pornește de la relațiile matematice al controlului scalar „ $U / f' = \text{constant}$ ”. Pentru a stabili o relație între tensiune și frecvență, se determină constanta raportului „ U / f' ” pentru valorile nominale. Fie $U_N = 380$ [V], iar $f_N = 50$ [Hz]:

$$\frac{U_N}{f_N} = C_N \rightarrow \frac{380}{50} = 4,6 \rightarrow U_N = 4,6 \cdot f_N$$

Astfel, pentru ca între tensiune și frecvență să se păstreze o relație de proporționalitate, se va înmulți frecvența impusă cu constanta de proporționalitate determinată, pentru a obține amplitudinea tensiunii necesară pentru acea frecvență. Frecvența, va fi variabilă, iar amplitudinea tensiunii va rezulta pe baza calculelor. Pentru un sistem demarator (eng. soft starter), se propune variația frecvenței în rampă, până la valoarea nominală a frecvenței rețelei (50 [Hz]). Pentru a varia frecvența în rampă se utilizează un bloc generator de semnal rampă la care se impune panta de creștere la valoarea 1000, și se va denumi „f”. Totodată pentru frecvență se va introduce și o variabilă cu numele „f”.

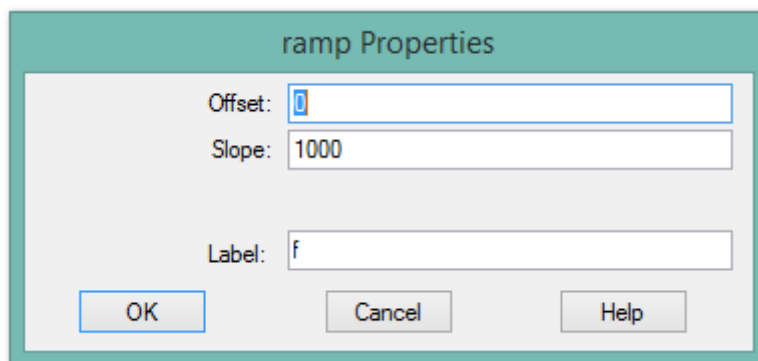


Fig. 16 – Stabilirea pantei de creșterea frecvenței

Creșterea frecvenței conform relației date mai sus, va determina și o creștere în amplitudine a tensiunii, doar că, tensiunea, va fi limitată la valoarea nominală (380 [V]) prin intermediul blocului „limit” care se regăsește în meniul „Blocs” categoria „Nonlinear”.

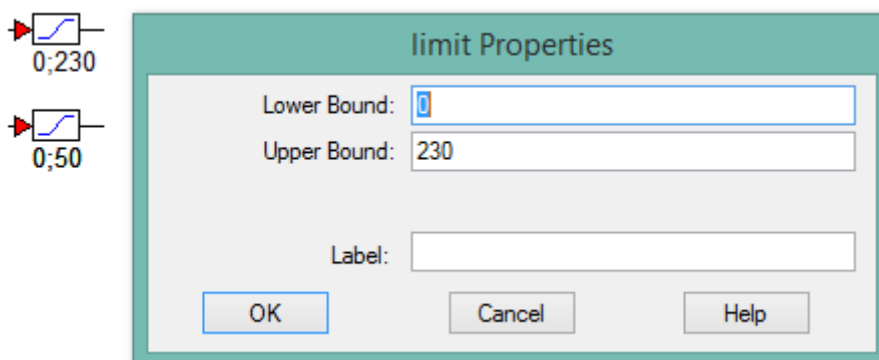


Fig. 17 – Limitarea amplitudinii tensiunii și a valorii frecvenței

Pentru a afișa două forme de undă în paralel se procedează astfel:

- în meniul „Plot Properties” se va alege secțiunea „Axis”;
- în secțiunea „Axis” se va selecta „Sub Plot Count” valoarea „2”;
- se va debifa opțiunea „Uniform Subplot Scales”;
- iar în partea dreaptă se va selecta axa „Y” care se dorește a fi parametrizată;
- pentru „Axis 1” se vor stabili limitele „0 – 100”, iar pentru „Axis 2” limitele „0 – 700”;

Realizat de: ing. drd. Pintilie Lucian - Nicolae

Pentru disciplina: „Sisteme cu FPGA și DSP”

Adresă de e-mail: Lucian.Pintilie@emd.utcluj.ro



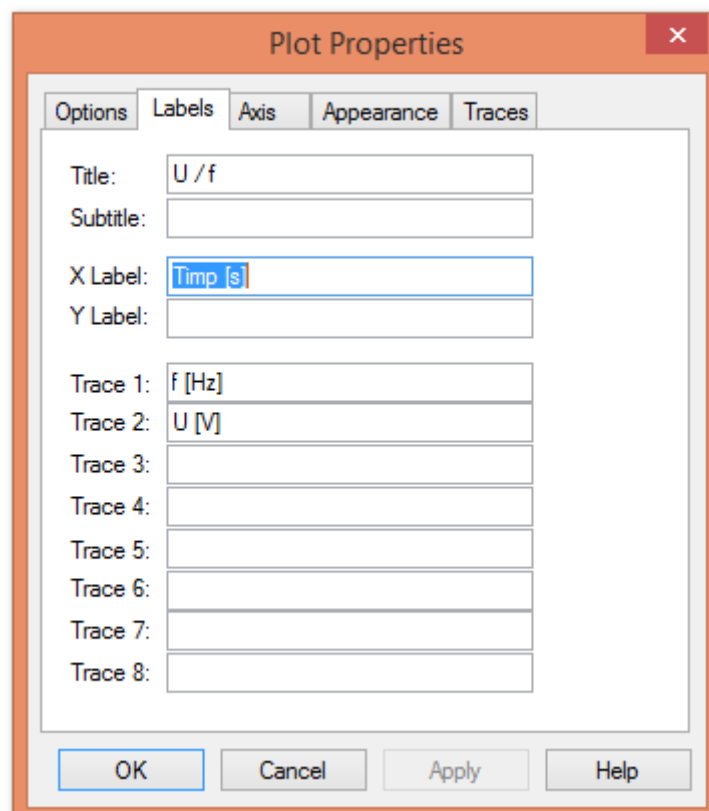
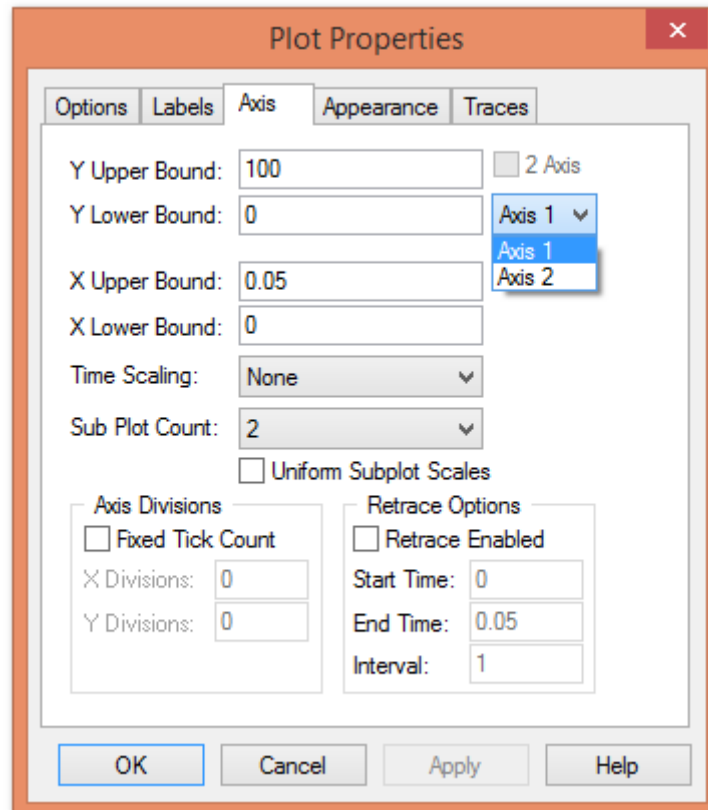


Fig. 18 – Parametrizarea graficelor cu mai multe axe „Y”

III. BIBLIOGRAFIE:

1. Visual Solutions, Inc. – „VisSim User’s Guide – Version 4.5” – Copyright © 2000 Visual Solutions, Inc., 487 Groton Road Westford, MA 01886;
2. Andre Veltman Duco W.J. Pulle Rik W. De Doncker – „Fundamentals of Electrical Drives” – VisSim and Plexim applications in electrical machines and drives using embedded systems;
3. Prof. dr. ing. Emil Simion – „Electrotehnică – Manual pentru subingineri” – Editura Didactică și Pedagogică – București – 1977;
4. Teodor Crișan Pană – „Sisteme de calcul cu microprocesoare, FPGA și DSP” – Editura UTPRESS, Cluj – Napoca, 2016 – ISBN 978-606-737-206-9;
5. Ioana – Cornelia GROS, Lucian – Nicolae PINTILIE, Teodor Crișan PANĂ – „SISTEME EMBEDDED ÎN INGINERIE ELECTRICĂ - GHID DE APLICAȚII” – Editura UTPress Cluj – Napoca, 2020 ISBN 978-606-737-431-5:
(<https://biblioteca.utcluj.ro/files/carti-online-cu-coperta/431-5.pdf>);