UNIVERSITATEA “POLITEHNICA” DIN BUCURESTI

Facultatea de Electronica, Telecomunicatii si Tehnologia Informatiei

Proiect

Semnale si programare

Analiza semnalelor in Matlab

Haiducescu Andrei

Gr. 421F

2021

Cuprins

Introducere

Semnalul este, în electronică, un curent electric sau câmp electromagnetic ce transmite informații de la un dispozitiv sau o rețea la altele. Acesta reprezintă elementul cheie din spatele domeniilor precum rețelistica și telecomunicațiile. Procesul prin care un semnal preia datele ce trebuie transmise se numește modulație și poate fi de două feluri: digitală sau analogică.

Semnalul analogic reprezintă un semnal continuu, ce variază în timp, reprezentând o mărime fizică ce variază tot în timp.

Semnalul digital este o mulțime de forme de undă ce pot fi considerate secvențe de cod.

Matlab este un mediu de programare specializat pentru ingineri și cercetători. Limbajul de programare oferă un mod eficient și simplu de a efectua operații matematice complexe asupra unui număr foarte mare de date.

De asemenea, platforma include modalități de vizualizare a datelor, caracteristică pe care o vom folosi în proiect pentru afișarea semnalelor reprezentate cu ajutorul funcțiilor matematice ce le definesc.

Se consideră semnalele:

$x(t) = a_2 t^2 + a_1 t + a_0$; $t \in [0,1]$

$x_i(t) = x(t) * \delta_{T_i}(t)$; $T_1 = 0,5$; $T_2 = 1$; $T_3 = 2$

$y_i(t) = x(t) * \sum_{k=-\infty}^{\infty} (-1)^k \delta(t - kT_i)$; $i = \overline{1,3}$; T_i

$z_i(t) = x(t) * \sum_{k=-\infty}^{\infty} (-1)^{k+1} \delta(t - kT_i)$; $i = \overline{1,3}$; T_i

Se cere:

a) Să se reprezinte grafic în matlab $x(t)$ pe suportul $[0;1]$ și pe suportul $[-1;3]$.

b) Să se construiască și să reprezinte grafic în matlab semnalele $x_i(t)$, $y_i(t)$, $z_i(t)$ pentru 3 perioade și pentru 37 perioade.

c) Să se construiască semnalele redresate monoalternanță și dublă alternanță și să reprezinte grafic pentru 20 perioade.

d) Să se construiască și să reprezinte grafic $x_{par}(t)$, $x_{impar}(t)$, $x(2t)$, $x(2t-1)$, $x(2t-2)$, $x\left(\frac{t}{2}\right)$, $x\left(\frac{t}{2}-1\right)$, $x^2(t)$, $x^2(2t)$, $\sqrt{|x(t)|}$ pe suportul $t \in [-4;6]$

e) Să se calculeze analitic componenta continuă C_0 pt $x_i(t)$, $y_i(t)$, $z_i(t)$.

f) Să se reprezinte grafic semnalele $x_i(t)$, $y_i(t)$, $z_i(t)$ fără c.c. pe o perioadă și pe 10 perioade.

g) Să se scrie în matlab un program care să calculeze componenta continuă cu o precizie de 10^{-5}.

h) Să se scrie un program în matlab care să calculeze puterea totală pe o perioadă cu o precizie de 10^{-4} și apoi să se calculeze analitic.

i) utilizând funcția heaviside, rectangularPulse, triangularPulse și fplot, să se reprezinte grafic:

$w_1 = n\,u(t-n) - (n-17)\,u(t-n-4) + (2n+15)\,u(t-n-7) - (30+2n)\,u(t-n-9) - 2u(t-n-10)$; $t \in [n-4; n+12]$

$w_2 = t\,u(t) + (nt-20)\,u(t-2) - (n-10)\,u(t-5) - (nt-10)\,u(t-7)$; $t \in [-2; 11]$

$w_3 = 5u(t-n) - 9\frac{t-3n}{n}[u(t-2n) - u(t-3n)] + 9\cdot\left(-\frac{t-2n}{n}+1\right)[u(t-3n) - u(t-4n)] - 5u(t-5n)$; $t \in [-n; 6n]$

Analiza Fourier a semnalelor periodice

Fie $x_p(t) = x_p(t+kT) = \sum_{k=-\infty}^{\infty} x(t+kT)$ periodic, $(\forall) t \in \mathbb{R}$

$T[s]$ - perioada principală

$f_0[Hz] = \frac{1}{T}$ - frecvența fundamentală

$\omega_0[rad/s] = 2\pi f_0$ - pulsația fundamentală

Seria Fourier exponențială

$$x_p(t) = \sum_{k-\infty}^{\infty} a_{k,c} \cdot e^{jk\omega_0 t}$$

$$a_{k,c} = \frac{1}{T}\int_T x_p(t) e^{-j\omega_0 k t} dt$$

$$a_{0,c} = \frac{1}{T}\int_T x_p(t) dt$$

Seria Fourier trigonometrică

$$x_p(t) = C_0 + \sum_{k=1}^{\infty} (C_k \cdot \cos(k\omega_0 t) + S_k \sin(k\omega_0 t))$$

$$C_k = \frac{2}{T}\int_T x(t) \cos(k\omega_0 t) dt$$ - comp pară

$$S_k = \frac{2}{T}\int_T x(t) \sin(k\omega_0 t) dt$$ - comp impară

$$C_0 = \frac{1}{T}\int_T x(t) dt$$

Seria Fourier armonică

$$x_p(t) = A_0 + \sum_{k=1}^{\infty} A_k \cdot \cos(k\omega_0 t + \varphi_k)$$

$$\varphi_k = -\mathrm{arctg}\left(\frac{S_k}{C_k}\right)$$

$$A_0 = \frac{1}{T}\int_T x(t)\,dt$$

Relații de legătură între SFA, SFE și SFT

$$a_{0C} = C_0 = A_0$$

$$a_{kC} = \frac{C_k}{2} - j\frac{S_k}{2} = \frac{C_k}{2} + j\frac{S_k}{2}$$

$$|a_{kC}| = \sqrt{\frac{C_k^2 + S_k^2}{2}} = \frac{A_k}{2} \Rightarrow A_k^2 = C_k^2 + S_k^2$$

$$S_k = -A_k \sin(\varphi_k)$$

$$C_k = A_k \cos(\varphi_k)$$

Proprietățile seriilor Fourier

1. Limetele de integrare se alege convenabil pentru calcule pe o perioadă.

2. $C_0 = A_0 = a_{0C} = \frac{1}{T}\int_T x_p(t)\,dt = \frac{1}{T}\cdot$ Arie

3. Proprietatea de paritate a semnalelor:

$x(t) = x(-t)$, $(\forall)\, t \in \mathbb{R} \Rightarrow$ simetric față de Oy

SFE: $a_{kc} \in \mathbb{R}$

SFT: $S_k = 0$

$C_k = \frac{4}{T}\int_0^{\frac{T}{2}} x_p(t)\cos(k\omega_0 t)\,dt$ $\begin{cases} >0 \Rightarrow SFT \equiv SFA \\ (A_k = C_k) \\ <0 \Rightarrow SFA \equiv SFT \end{cases}$

$k \in \mathbb{N}^*$

cu excepția $\varphi_k = \pi$

$A_k = |C_k|$

4. Proprietatea de imparitate a semnalelor

$x(t) = -x(-t) \Rightarrow$ O punct de simetrie

a_{kc} - pur imaginar

$C_0 = 0$; $C_k > 0$ pt $(\forall)\, k \in \mathbb{N}^*$

$S_k = \frac{4}{T}\int_0^{\frac{T}{2}} x_p(t)\sin(k\omega_0 t)\,dt$

5. Proprietatea de simetrie prin rotație

$x(t) = -x(t \pm \frac{T}{2})$, $(\forall)\, t \in \mathbb{R}$

Un semnal are simetrie de rotație dacă prin deplasarea la stânga sau la dreapta cu jumătate de perioadă $(T/2)$ și apoi după rotire obținem semnalul original.

Semnalele cu simetrie de rotație au componenta continuă și armonicele pare nule.

Pot avea și semnalele pare și cele impare.

6. Proprietatea de simetrie ascunsă

Fie $x(t)$ un semnal fără proprietatea de simetrie prin rotație și imparitate.

Fie $x_1(t) = x(t) - C_0$, $x_1(t)$ poate fi impar $\to$ $x_1(t)$ transmite proprietățile lui $x(t)$ fără proprietățile componentei continue $\to$ C_0 a ascuns proprietatea de paritate.

7. Proprietatea de deplasare în timp (întârziere)

SFE: $x(t) = \sum_{k=-\infty}^{\infty} a_k e^{jk\omega_0 t}$

$y(t) = x(t-t_0) = \sum_{k=-\infty}^{\infty} b_k \cdot e^{jk\omega_0 t}$, $b_k = a_k \cdot e^{jk\omega_0 t}$

SFT: $x(t) = C_0 + \sum_{k=1}^{\infty} (C_k \cos(k\omega_0 t) + S_k \sin(k\omega_0 t))$

$y(t) = x(t-t_0) = C_0' + \sum_{k=1}^{\infty} (C_k' \cos(k\omega_0 t) + S_k' \sin(k\omega_0 t))$

$C_0' = C_0$

$C_k' = C_k \cos(k\omega_0 t) - S_k \sin(k\omega_0 t)$

$S_k' = C_k \sin(k\omega_0 t) + S_k \cos(k\omega_0 t)$

SFA: $x(t) = A_0 + \sum_{k=1}^{\infty} (A_k \cos(k\omega_0 t) + \varphi_k)$

$x(t-t_0) = A_0 + \sum_{k=1}^{\infty} A_k \cos(k\omega_0 t + \varphi_k - k\omega_0 t_0)$

$y(t) = B_0 + \sum_{k=1}^{\infty} B_k \cos(k\omega_0 t + \varphi_k')$

$A_0 = B_0$ $A_k = B_k$ $\varphi_k' = \varphi_k - k\omega_0 t_0$

8. Proprietatea de derivare

SFE: $x(t)=\sum_{\substack{k=-\infty \\ k\neq 0}}^{\infty} a_{kc} e^{j\omega_0 kt} + a_{0c}$

$y(t)=\sum_{\substack{k=-\infty \\ k\neq 0}}^{\infty} b_{kc} e^{j\omega_0 kt} = \frac{dx(t)}{dt}$, $b_{kc}=jk\omega_0 \cdot a_{kc} (\forall) k\in Z^*$, $b_{0c}=0$

SFT: $x(t)=C_0+\sum_{k=1}^{\infty}(C_k \cos(k\omega_0 t)+S_k \sin(k\omega_0 t))$

$y(t)=\frac{dx(t)}{dt}=\sum_{k=1}^{\infty}(-k\omega_0 C_k \sin(k\omega_0 t)+k\omega_0 S_k \cos(k\omega_0 t))$

SFA: $x(t)=A_0+\sum_{k=1}^{\infty} A_k \cos(k\omega_0 t+\varphi_k)$

$y(t)=\frac{dx(t)}{dt}=\sum_{k=1}^{\infty}(-k\omega_0) A_k \sin(k\omega_0 t+\varphi_k)$

9. Proprietatea de integrare

$x(t)=\sum_{k=-\infty}^{\infty} a_{kc} e^{jk\omega_0 t}$, $a_{0c}=0$

$y(t)=\int^{t} x(\tau)\,d\tau=\sum_{k=-\infty}^{\infty}\frac{a_{kc}}{jk\omega_0} e^{jk\omega_0 t}+C$

Analiza Fourier a semnalelor neperiodice

$X(\omega)=\mathcal{F}\{x(t)\}=\int_{-\infty}^{\infty} x(t)e^{-j\omega t}dt$

$x(t)=\mathcal{F}\{X(\omega)\}=\frac{1}{2\pi}\int_{-\infty}^{\infty} X(\omega)e^{j\omega t}d\omega$

$E=\int_{-\infty}^{\infty} x^2(t)\,dt=\frac{1}{2\pi}\int_{-\infty}^{\infty}|X(\omega)|^2 d\omega$

Proprietățile transformatei Fourier

1. Liniaritate

$$\left.\begin{aligned} x(t) &= \alpha_1 x_1 + \alpha_2 x_2(t) \\ X(\omega) &= \alpha_1 X_1(\omega) + X_2(\omega) \end{aligned}\right\} \Rightarrow \mathcal{F}\{\alpha_1 x_1(t) + \alpha_2 x_2(t)\} = \alpha_1 \mathcal{F}\{x_1(t)\} + \mathcal{F}\{x_2(t)\}$$

2. Conjugare

$$x(t) \in \mathbb{R} \Rightarrow \begin{cases} X(\omega) = |X(-\omega)| \\ \varphi(\omega) = -\varphi(-\omega) \end{cases}$$

$$x_{par}(t) = \frac{x(t) + x(-t)}{2} \longleftrightarrow \frac{X(\omega) + X(-\omega)}{2} = Re\{X(\omega)\}$$

$$x_{impar}(t) = \frac{x(t) - x(-t)}{2} \longleftrightarrow \frac{X(\omega) - X(-\omega)}{2} = j \cdot Im\{X(\omega)\}$$

3. Întârzierea în timp

$$x(t) \longleftrightarrow X(\omega)$$

$$x(t - t_0) \longleftrightarrow X(\omega) \cdot e^{-j\omega t_0}$$

$$x(t + t_0) \longleftrightarrow X(\omega) \cdot e^{j\omega t_0}$$

4. Deplasarea în frecvență

$$x(t) \longleftrightarrow X(\omega)$$

$$x(t) e^{j\omega_0 t} \longleftrightarrow X(\omega - \omega_0)$$

$$\begin{cases} x(t) \cdot \cos(\omega_0 t) \longleftrightarrow \frac{1}{2}[X(\omega - \omega_0) - X(\omega + \omega_0)] \\ x(t) \cdot \sin(\omega_0 t) \longleftrightarrow \frac{1}{2j}[X(\omega - \omega_0) - X(\omega + \omega_0)] \end{cases}$$

10

5. Simetria TF

$x(t) \longleftrightarrow X(\omega)$

$X(t) \longleftrightarrow 2\pi x(-\omega)$

6. Schimbarea de scală

$x(t) \in L_1(\mathbb{R}),\ a \in \mathbb{R}^*$

$x(at) \longleftrightarrow \frac{1}{|a|} X\left(\frac{\omega}{a}\right)$

7. Derivarea în timp

$x(t) \longleftrightarrow X(\omega)$

$x'(t) \longleftrightarrow j\omega X(\omega)$

$x^{(n)}(t) \longleftrightarrow (j\omega)^n \cdot X(\omega)$

8. Integrarea în timp

$x(t) \in L_1(\mathbb{R}),\ X_0 = 0 \Rightarrow$

$\Rightarrow x(t) \longleftrightarrow X(\omega)$

$$\int_{-\infty}^{t} x(\tau)\, d\tau \longleftrightarrow \frac{X(\omega)}{j\omega}$$

Convoluția semnalelor

$$x(t) = \int_{-\infty}^{\infty} x(\tau) \cdot x_2(t-\tau)\, d\tau = x_1(t) * x_2(t)$$

Teorema integrării de convoluție

$x_1(t) \longleftrightarrow X_1(\omega)$

$x_2(t) \longleftrightarrow X_2(\omega)$

$x(t) = x_1(t) * x_2(t) \longleftrightarrow X(\omega) = X_1(\omega) \cdot X_2(\omega)$

Consecințe:

1. Comutativitate

$$x_1(t) * x_2(t) = x_2(t) * x_1(t)$$

2. Asociativitate

$$x_1 * (x_2 * x_3) = (x_1 * x_2) * x_3$$

3. Distributivă

$$x_1 * (x_2 + x_3) = x_1 * x_2 + x_1 * x_3$$

4. Produsul de convoluție nu se schimbă la convoluție / integrare simetrică

$$x'(t) = x_1(t) * x_2(t) = x_1'(t) * \int_{-\infty}^{t} x_2(\tau)\,d\tau = \int_{-\infty}^{t} x_1(\tau)\,d\tau * x_2'(t)$$

5. Derivarea produsului de convoluție

$$x(t) = x_1(t) * x_2(t)$$

$$x'(t) = x_1'(t) * x_2(t) = x_1(t) * x_2'(t)$$

$$x''(t) = x_1'(t) * x_2'(t) = x_1''(t) * x_2(t) = x_1(t) * x_2''(t)$$

6. Definim $supp\{x(t)\} = \{t \in \mathbb{R} \mid x(t) \neq 0\}$

$$\left.\begin{array}{l} supp\{x(t)\} = t_2 - t_1 \\ x(t) = x_1(t) * x_2(t) \end{array}\right\} \Rightarrow supp\{x(t)\} = supp\{x_1(t)\} + supp\{x_2(t)\}$$

7. $x(t) = x_1(t) * x_2(t) \to x_1(t-t_1) * x_2(t-t_2) =$
$= x(t-t_1-t_2)$

Distribuții

Distribuția este un proces de atribuire printr-o funcțională $f(t)$ a unor valori $V(t)$ pentru o funcție $x(t)$.

Distribuții regulate (de tip funcție)

Proprietăți

1. Anularea unei distribuții

Dacă $\langle f, \varphi \rangle = 0; (\forall) \varphi \in D \Rightarrow f = 0$

2. Egalitatea a două distribuții

Dacă $\langle f_1, \varphi \rangle = \langle f_2, \varphi \rangle; (\forall) \varphi \in D \Rightarrow f_1 = f_2$

3. Limita unui șir de distribuții

Dacă $\lim_{n \to \infty} \langle f_n, f \rangle = \langle f, f \rangle; (\forall) f \in D \Rightarrow \lim f_n = f$

4. Schimbarea de variabile

$\langle f(at+b), f(t) \rangle = \frac{1}{|a|} \langle f(t), \varphi\left(\frac{t-b}{a}\right) \rangle$
$(\forall) \varphi \in D, (\forall) a \in \mathbb{R}^*, b \in \mathbb{R}$

5. Înmulțirea unei distribuții cu o funcție $g \in C^\infty$

$\langle gf, \varphi \rangle = \langle f, g\varphi \rangle; \varphi \in D$

6. Derivarea unei distribuții

$$< \varphi^{(n)}, \varphi > = (-1)^n < f, g^{n} > ; (\forall)\, \varphi \in D$$

7. Liniaritatea

$$< \alpha_1 f_1 + \alpha_2 f_2, \varphi > = \alpha_1 < f_1, \varphi > + \alpha_2 < f_2, \varphi > ; (\forall)\, \varphi \in D, \alpha_1, \alpha_2 \in \mathbb{C}$$

Distribuția Dirac

Proprietăți

1. $\delta(t)$ - distribuția punctuală

dacă $supp\{\varphi\} = (-\infty; -\varepsilon); \varepsilon > 0 \Rightarrow < \delta, \varphi > = \varphi(0) = 0$

dacă $supp\{\varphi\} = (\varepsilon; \infty); \varepsilon > 0 \Rightarrow < \delta, \varphi > = \varphi(0) = 0$

2. $\delta(at) = \frac{1}{|a|} \delta(t)$

$\delta(-t) = \delta(t)$

3. $g(t) \in C^{\infty}(\mathbb{R})$

$g(t) \cdot \delta(t) = g(0) \cdot \delta(t)$

4. $< \delta(t - t_0), f(t) > = f(t_0)$

5. $g(t) \cdot \delta'(t) = g(0) \cdot \delta''(t) - g''(0) \cdot \delta(t)$

6. $\delta(t) = \frac{du(t)}{dt}$; $u(t)$ - Heaviside

7. Fie $f(t)$ - cont pe $\mathbb{R}$, cu o discontinuitate de prima speță

$s_{t_0} = f(t_0 + 0) - f(t_0 - 0)$

$\Rightarrow f'(t) = s_{t_0}\,\delta(t-t_0) + f'_-(t)$

generalitare: $f'(t) = \sum_k s_k\,\delta(t-t_k) + f'_-(t)$, t_k - disc sp I

$\mathcal{F}\{\delta(t)\} = 1(\omega)$

$\delta(t) \leftrightarrow 1(\omega)$

$1(t) \leftrightarrow 2\pi\delta(-\omega) = 2\pi\delta(\omega)$ ($\delta(t)$ pară)

$x(t)e^{j\omega_0 t} \leftrightarrow X(\omega-\omega_0)$

$1(t)e^{j\omega_0 t} \leftrightarrow 2\pi\delta(\omega-\omega_0) \Rightarrow e^{j\omega_0 t} \leftrightarrow 2\pi\delta$

$\cos(\omega_0 t) \leftrightarrow \pi[\delta(\omega-\omega_0)+\delta(\omega+\omega_0)]$

$\sin(\omega_0 t) \leftrightarrow \frac{\pi}{j}[\delta(\omega-\omega_0)-\delta(\omega+\omega_0)]$

$1(t) \leftrightarrow 2\pi\delta(\omega)$

$u(t) \leftrightarrow \frac{1}{j\omega} + \pi\delta(\omega)$

Distribuția Dirac periodic

$\delta_T(t) = \sum_{n=-\infty}^{\infty} \delta(t-nT) = \frac{1}{T}\sum_{k=-\infty}^{\infty} e^{jk\omega_0 t}$

$e^{j\omega_0 t} \leftrightarrow 2\pi\delta(\omega-\omega_0) \Rightarrow e^{jk\omega_0 t} \leftrightarrow 2\pi\delta(\omega-\omega_0)$

$\delta_T(t) \leftrightarrow \omega_0\,\delta_{\omega_0}(\omega)$

$\delta_{\omega_0}(\omega) = \sum_{n=-\infty}^{\infty} \delta(\omega - n\omega_0)$

Structura programului

Am citit variabilele a0, a1 si a2 din fisierul excel.

n	a2	a1	a0
4	11,333	- 0.0469	0.4118

```
read_variables.m

function [a2, a1, a0] =
readVariables(file,  sheet,  number)
table = readtable(file, 'Sheet', sheet);
variable_table = table(number,2:4);
a2 = str2double(variable_table.a2);
a1 = str2double(variable_table.a1);
a0 = variable_table.a0;
end
```

```
main.m

file = 'Teme proiect SP 421F 422F.xlsx';
number = 4;
sheet = 'Sheet1';
[a2, a1, a0] = readVariables(file, sheet,
number);
t1 = linspace(0, 1);
t2 = linspace(-1, 3);
T = [0.5, 1, 2];
period = 1;
number_of_periods = [3, 37, 20, 1, 10];
s = linspace(-4, 6);
CC = [1.827839, 4.166016, 2.083008];
precision = 100000;
precision_t = linspace(0,1,precision);
```

Am definit fiecare variabila de care este nevoie pe parcursul programului.

Am creat o functie care sa schimbe designului graficelor afisate.

```
design_graph.m

function [] = design_graph(x_label,
y_label, title_)
grid minor;
xlabel(x_label);
ylabel(y_label);
title(title_);
end
```

```
display_signal.m

function [] = display_signal(signal, T,
j, i, name, number_of_periods, period,
signal_i)

[xi, si] = signal_i(signal, T(j),
number_of_periods(i), period);
subplot(3, 1, j);

plot(si, xi);
y_label = name + string(j) + '(t)';
title_ = 'Reprezentarea grafica ' +
y_label + ' pe ' +
string(number_of_periods(i)) + '
perioade';
design_graph('t', y_label, title_);
end
```

Am creat o functie care sa afiseze un semnal dat ca argument.

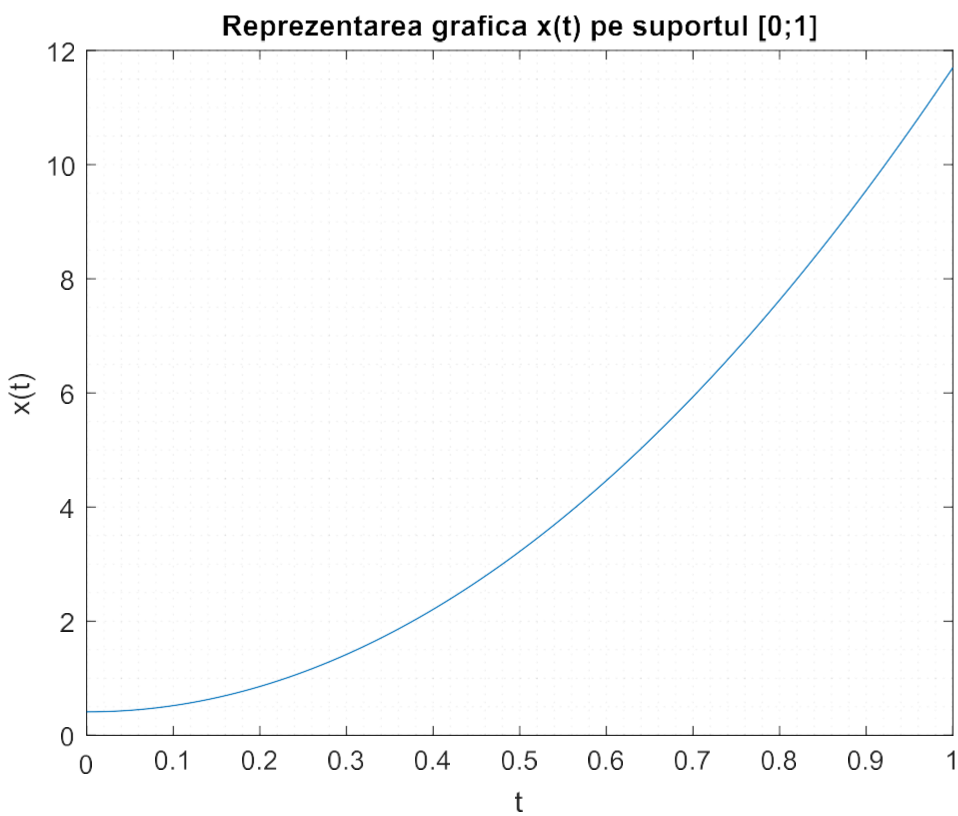
Apasarea tastei Enter continua afisajul semnalelor de la o cerinta la alta.

```
main.m

input('next');
```

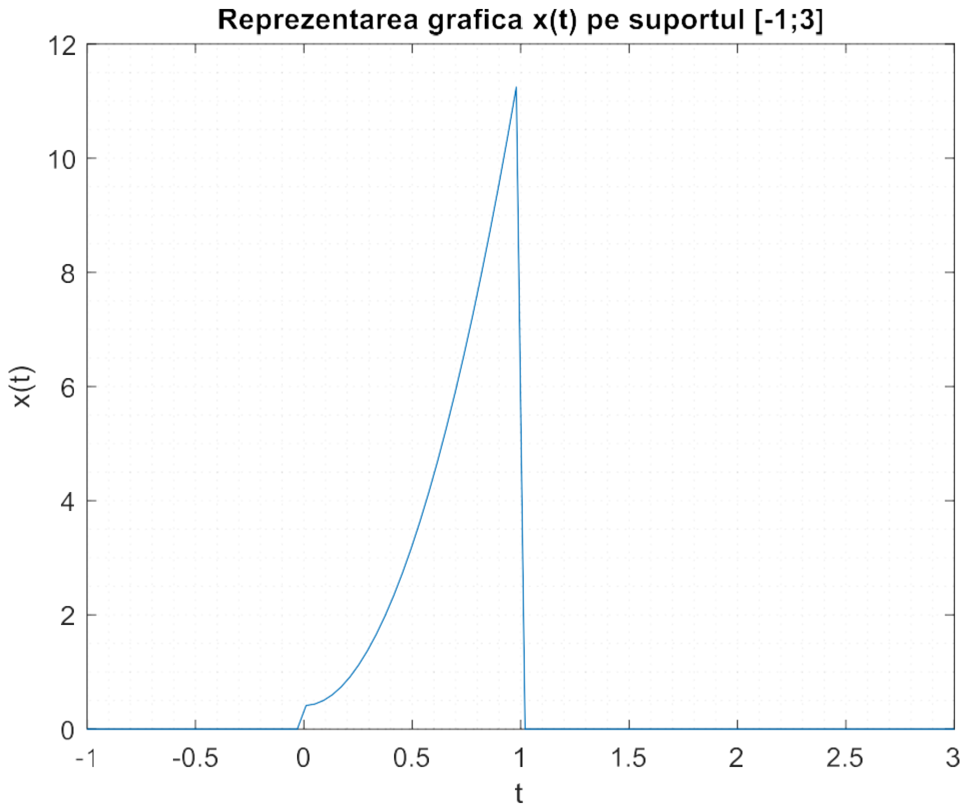
Rezolvarea cerintelor

a) x(t)

```
% MAIN SIGNAL
plot(t1, signal_main(a0, a1, a2, t1));
design_graph('t', 'x(t)', 'Reprezentarea grafica x(t)');

input('next');
```



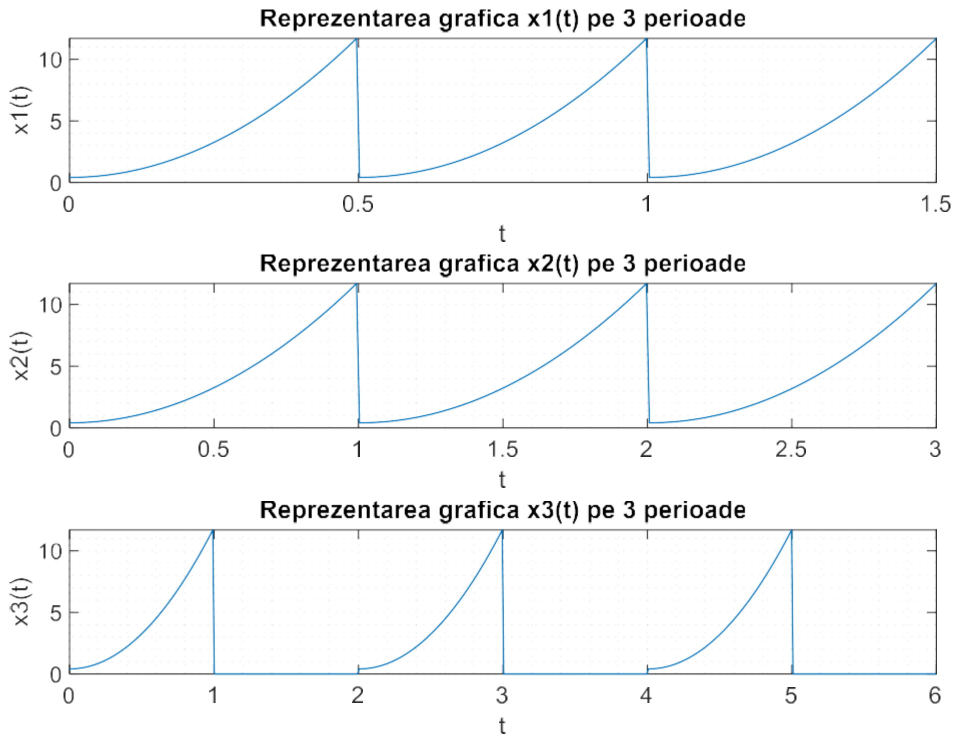

```
plot(t2, signal_main(a0, a1, a2, t2));
design_graph('t', 'x(t)', 'Reprezentarea grafica x(t) pe suportul [-1;3]');

input('next');
```

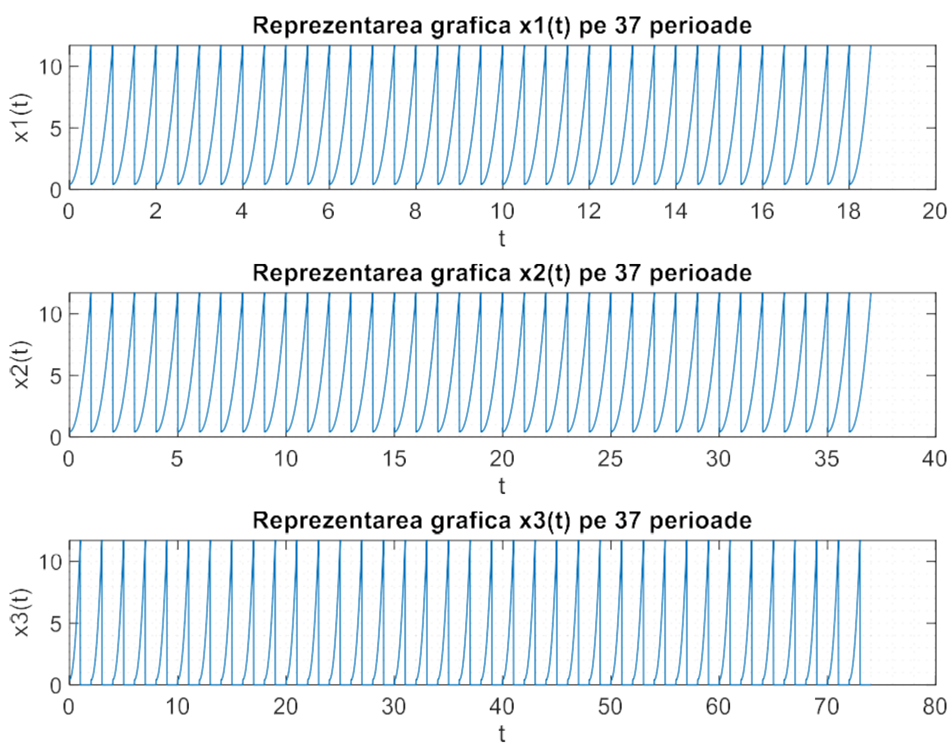


b) xi(t), yi(t), zi(t)

```
% xi signal
for i = 1:2
    for j = 1:3
        name = 'x';
        signal = signal_main(a0,
a1, a2, t1);
        display_signal(signal, T,
j, i, name, number_of_periods,
period, @signal_xi);
    end
    input('next');
end
```

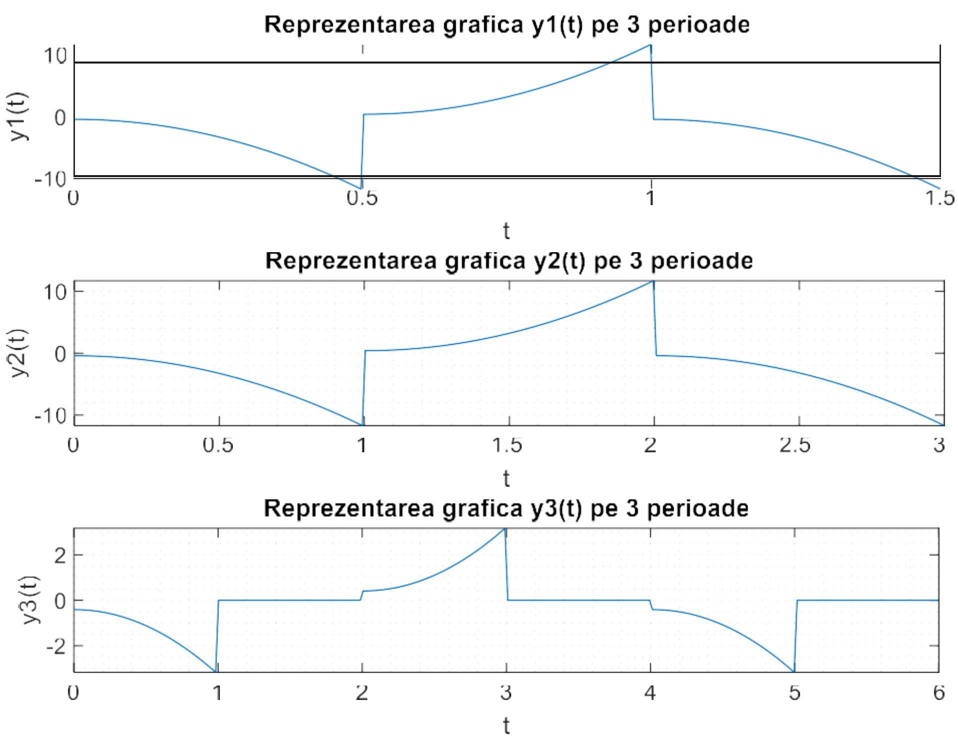
```
signal_xi.m

function [xi, si] = signal_xi(x, T,
number_of_periods, period)
% return signal xi
if T > period
    for i = 1:100 * T
        if i > 100 * (T - period)
            x(i) = 0;
        end
    end
end
xi = x'*ones(period, period *
number_of_periods);
xi = xi(:);
if T > period
    si = linspace(0, T *
number_of_periods, 100 *
number_of_periods * T);
else
    si = linspace(0, T *
number_of_periods, 100 *
number_of_periods);
end
    si = si(:);
end
```

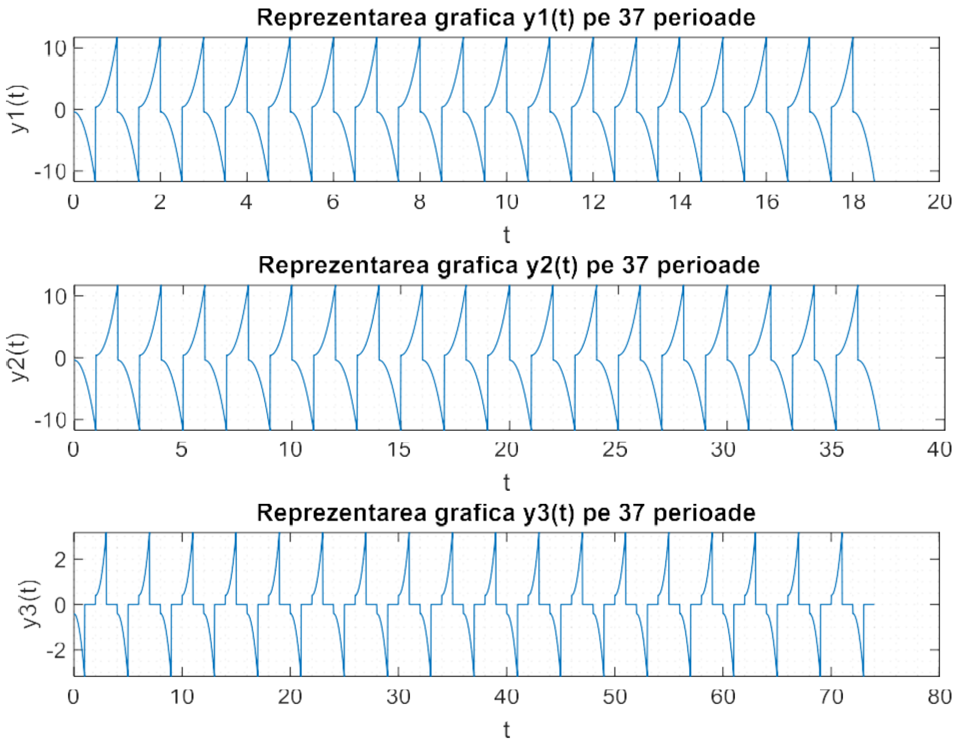

Reprezentarea grafica x1(t) pe 3 perioade
x1(t)
t
Reprezentarea grafica x2(t) pe 3 perioade
x2(t)
t
Reprezentarea grafica x3(t) pe 3 perioade
x3(t)
t


Reprezentarea grafica x1(t) pe 37 perioade
x1(t)
t
Reprezentarea grafica x2(t) pe 37 perioade
x2(t)
t
Reprezentarea grafica x3(t) pe 37 perioade
x3(t)
t

```
signal_yi.m

function [yi, si] = signal_yi(x, T,
number_of_periods, period)
% return signal yi
yi = x'*ones(period, period *
number_of_periods);
for i=1:number_of_periods
    for j=1:100
        yi(j,i)=((-1)^i)*yi(j,i);
        if T > period
            if j > 100 * period/T
                yi(j,i) = 0;
            end
        end
    end
end
yi = yi(:);
si = linspace(0, T * number_of_periods,
100 * number_of_periods);
si = si(:);
end
```

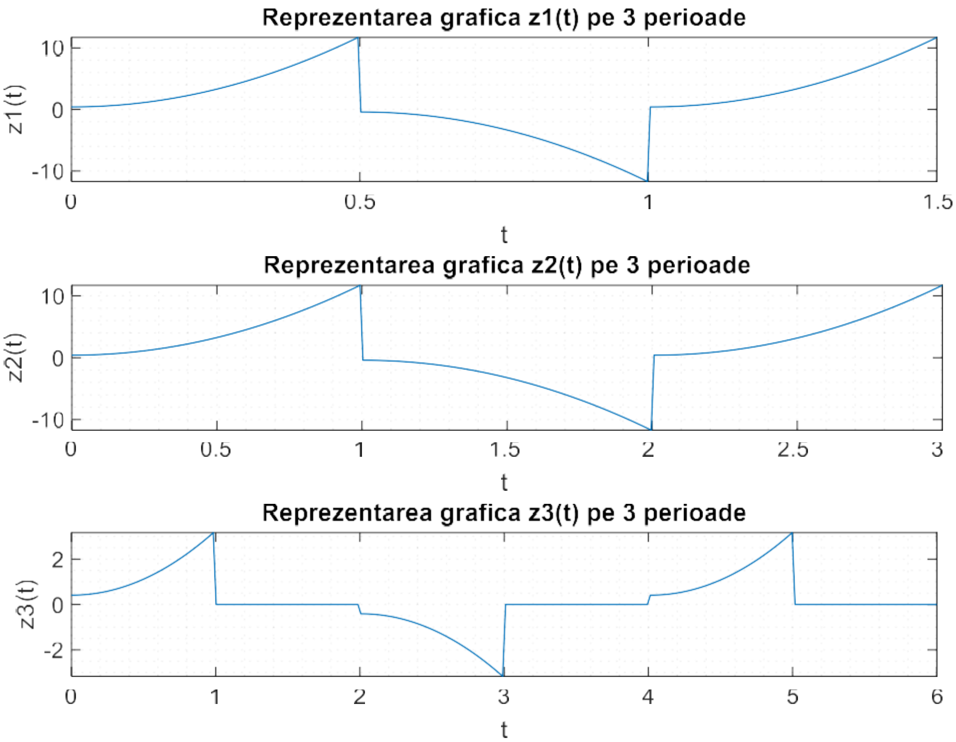
```
% yi signal
for i = 1:2
    for j = 1:3
        name = 'y';
        signal = signal_main(a0,
a1, a2, t1);
        display_signal(signal, T,
j, i, name, number_of_periods,
period, @signal_yi);
    end
    input('next');
end
```

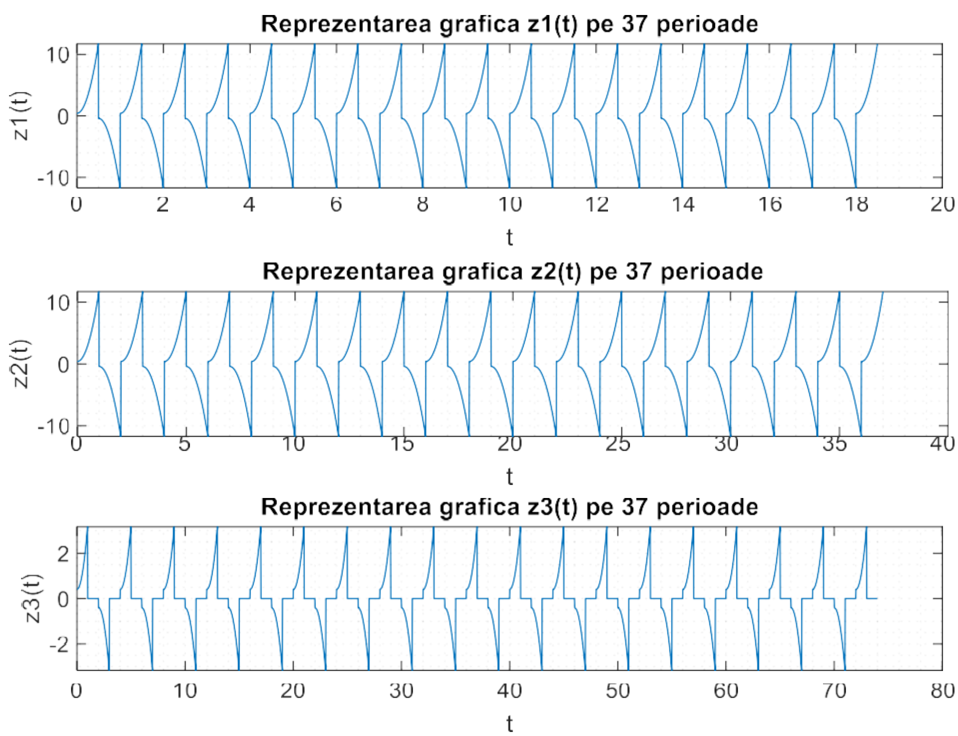





```
% zi signal
for i = 1:2
    for j = 1:3
        name = 'z';
        signal = signal_main(a0,
a1, a2, t1);
        display_signal(signal, T,
j, i, name, number_of_periods,
period, @signal_zi);
    end
    input('next');
end
```

```
signal_zi.m

function [zi, si] = signal_zi(x, T,
number_of_periods, period)
% return signal zi
zi = x'*ones(period, period *
number_of_periods);
for i=1:number_of_periods
    for j=1:100
        zi(j,i)=((-1)^(i+1))*zi(j,i);
        if T > period
            if j > 100 * period/T
                zi(j,i) = 0;
            end
        end
    end
end
zi = zi(:);
si = linspace(0, T * number_of_periods,
100 * number_of_periods);
si = si(:);
end
```

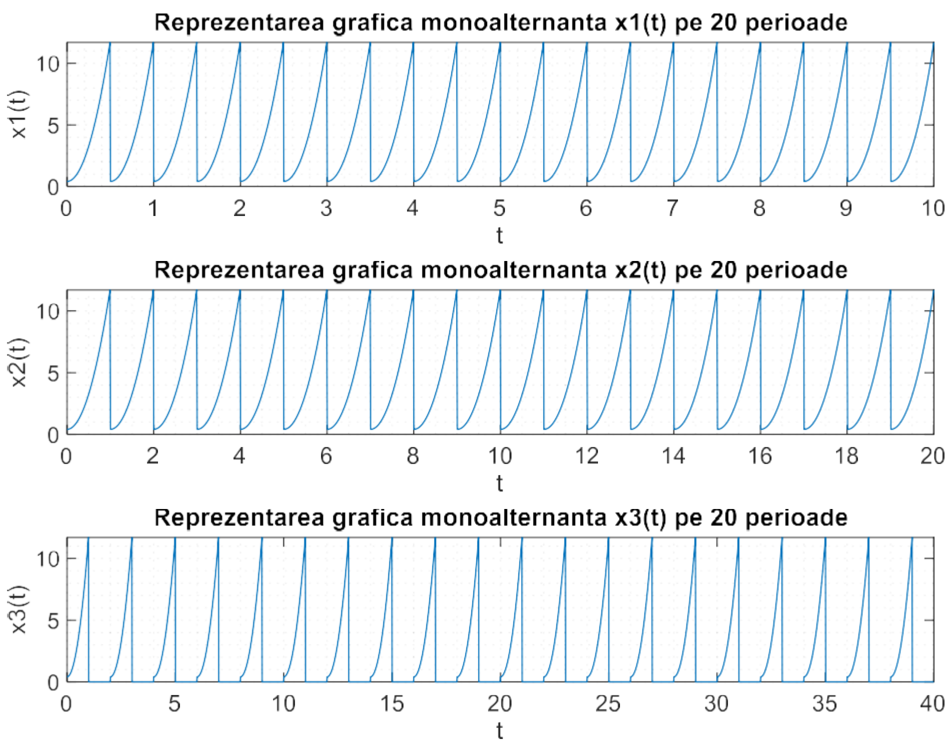

Reprezentarea grafica z1(t) pe 3 perioade
z1(t)
t
Reprezentarea grafica z2(t) pe 3 perioade
z2(t)
t
Reprezentarea grafica z3(t) pe 3 perioade
z3(t)
t


Reprezentarea grafica z1(t) pe 37 perioade
z1(t)
t
Reprezentarea grafica z2(t) pe 37 perioade
z2(t)
t
Reprezentarea grafica z3(t) pe 37 perioade
z3(t)
t

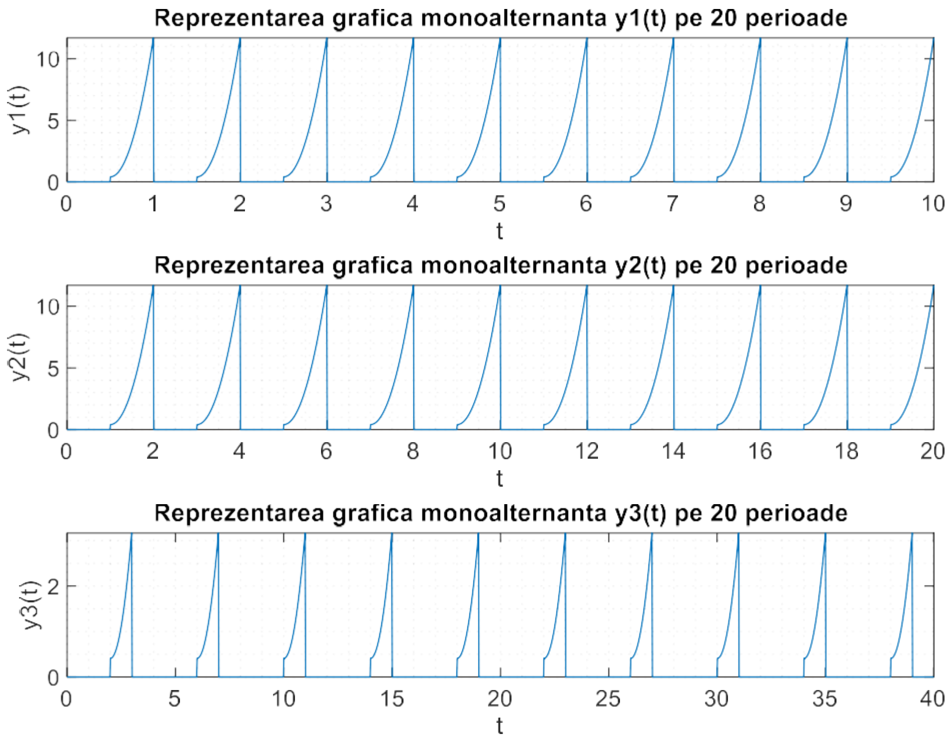
c) monoalternanta si dubla alternanta

```
% xi monoalternate
number_of_periods[3] = 20 => i =
3
for j = 1:3
    name = 'x';
    signal = signal_main(a0, a1,
a2, t1);
    display_mono_signal(signal,
T, j, 3, name, number_of_periods,
period, @signal_xi, @mono);
end
```

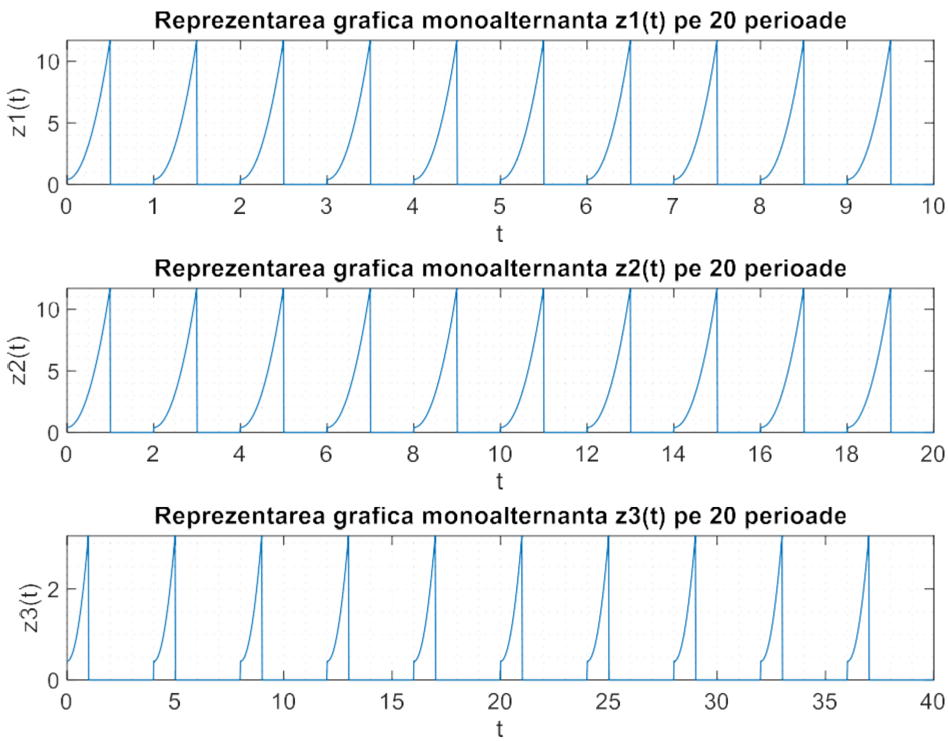
```
mono.m

function [mono_signal] = mono(x, s)
for i=1:1:length(s)
    if (x(i)<0)
        x(i)=0;
    end
end
mono_signal = x;
end
```



```
% yi monoalternate number_of_periods[3] = 20 => i = 3
for j = 1:3
    name = 'y';
    signal = signal_main(a0, a1, a2, t1);
    display_mono_signal(signal, T, j, 3, name, number_of_periods, period, @signal_yi,
@mono);
end
```

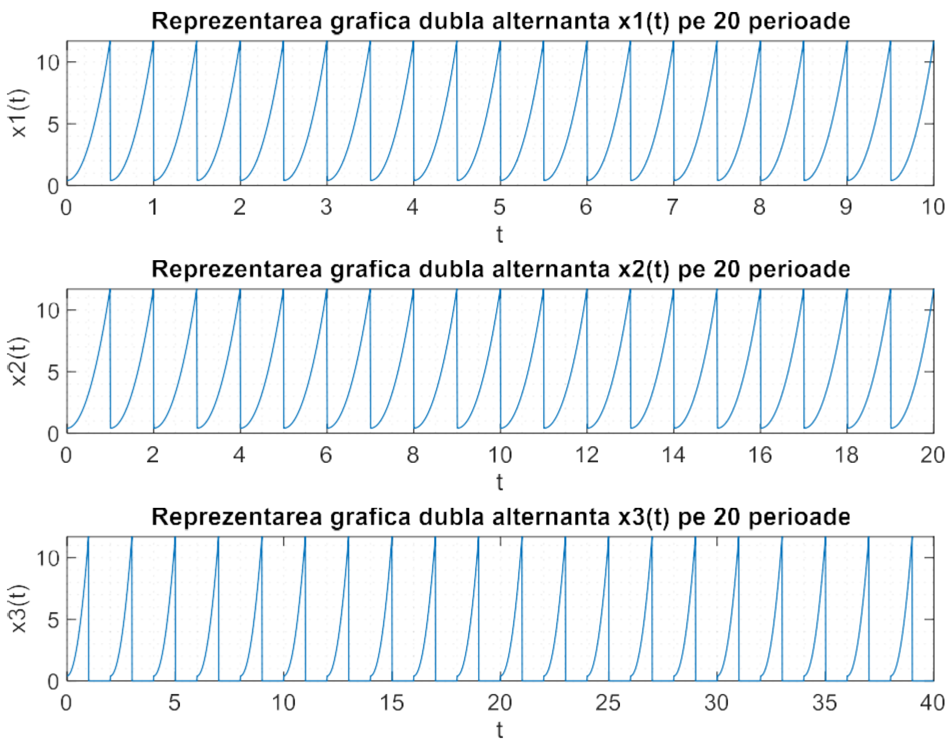


```
%zi monoalternate
for j = 1:3
    name = 'z';
    signal = signal_main(a0, a1, a2, t1);
    display_mono_signal(signal, T, j, 3, name, number_of_periods, period, @signal_zi,
@mono);
end
```

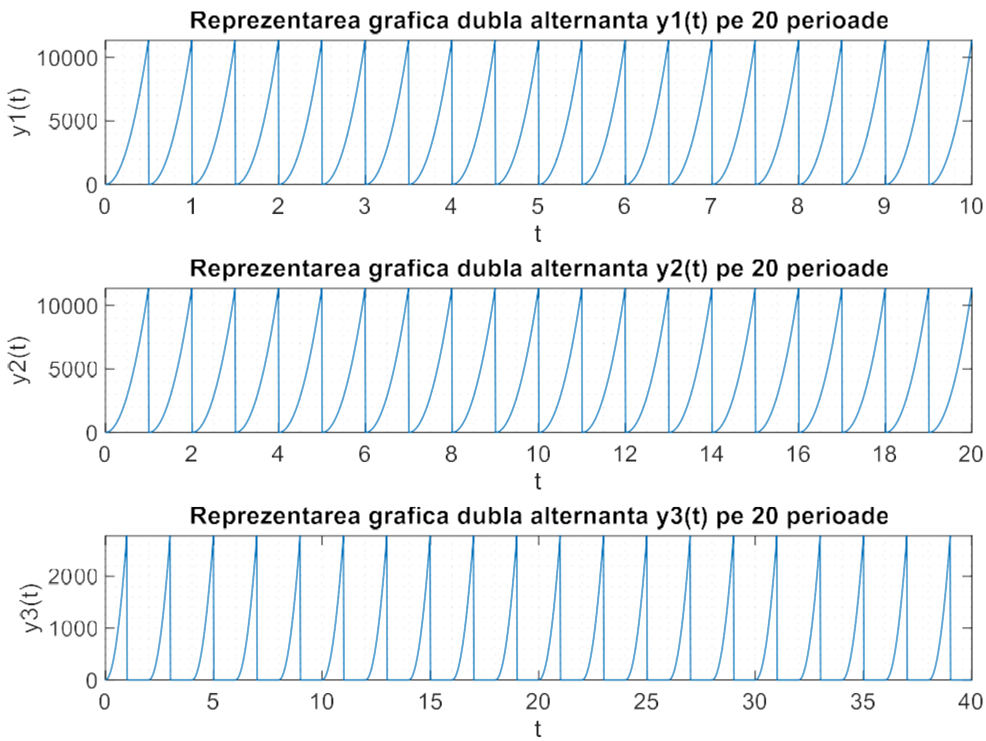


```
double.m

function [double_signal] = double_(x, s)
for i=1:1:length(s)
    if (x(i)<0)
        x(i)=-x(i);
    end
end
double_signal = x;
end
```

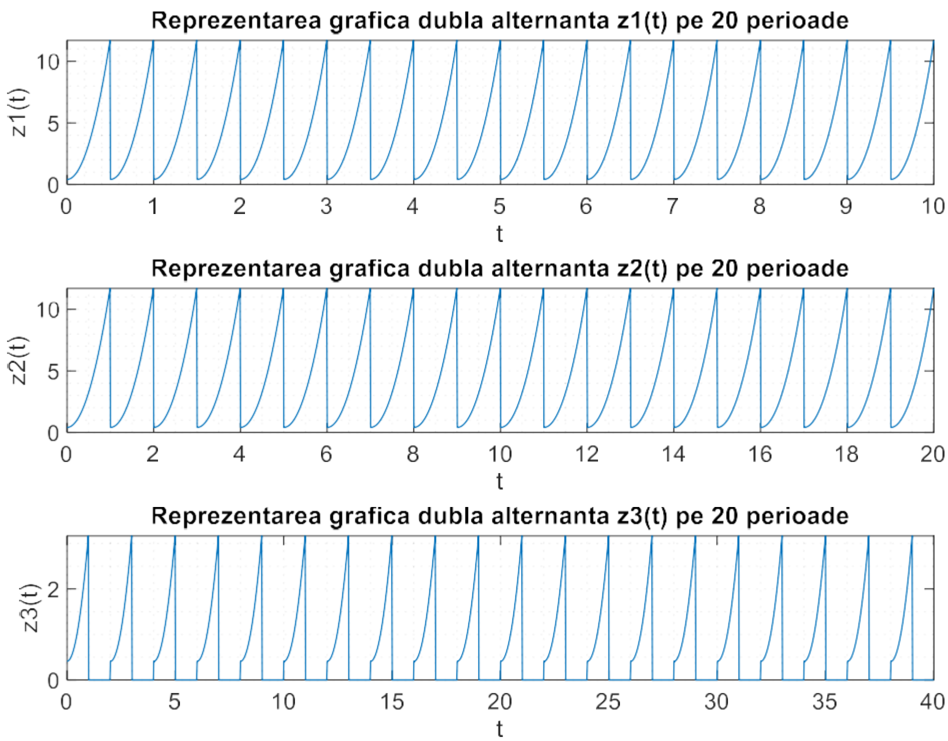
```
%xi double alternate
for j = 1:3
    name = 'x';
    signal = signal_main(a0, a1, a2, t1);
    display_double_signal(signal, T, j, 3, name, number_of_periods, period, @signal_xi, @double_);
end
```



```
% yi double alternate
for j = 1:3
    name = 'y';
    signal = signal_main(a0, a1, a2, t1);
    display_double_signal(signal, T, j, 3, name, number_of_periods, period,
@signal_yi, @double_);
end
```



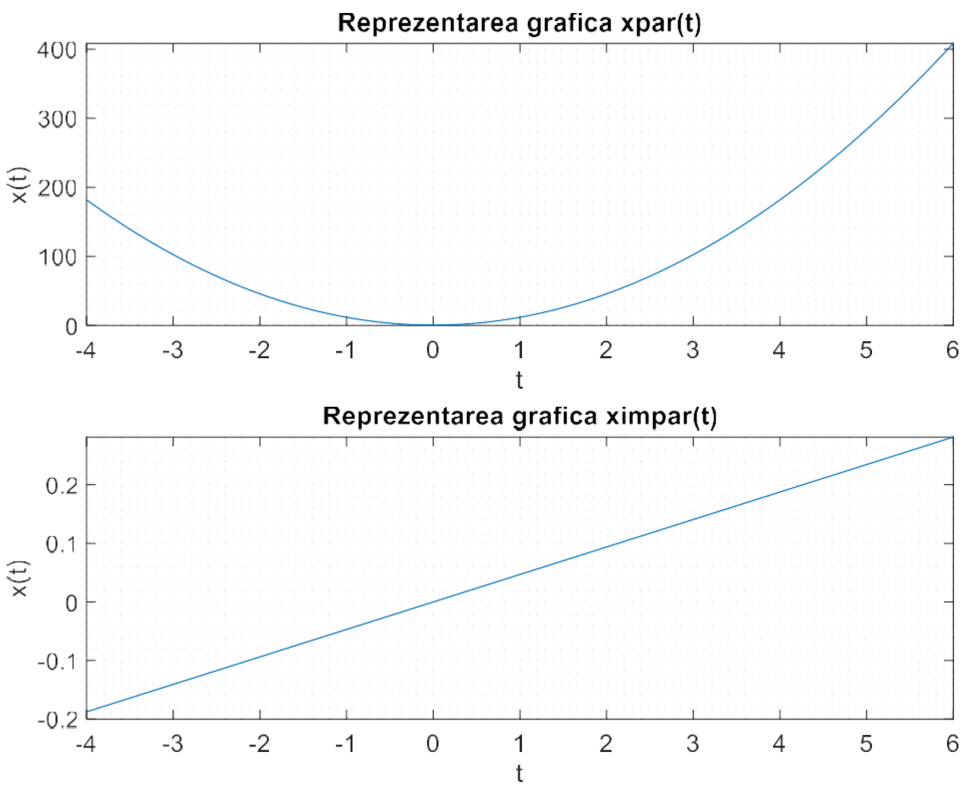
```
%zi double alternate
for j = 1:3
    name = 'z';
    signal = signal_main(a0, a1, a2, t1);
    display_double_signal(signal, T, j, 3, name, number_of_periods, period,
@signal_zi, @double_);
end
```



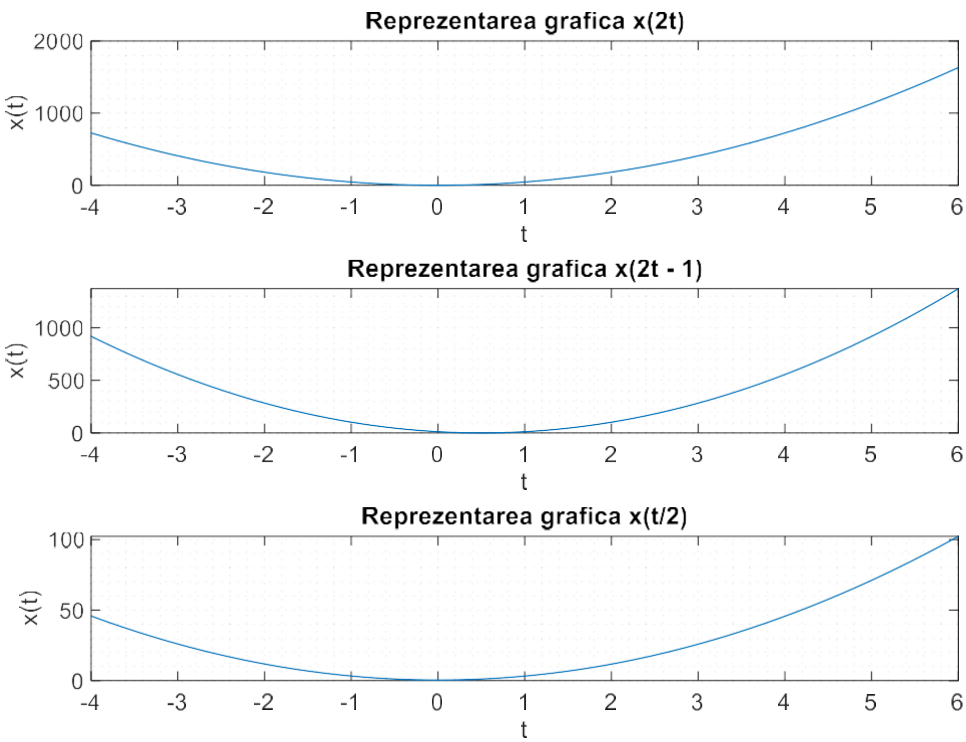
d) xpar(t), ximpar(t), x(t/2), x(2t)…

```
% x par => a1 = 0
subplot(2, 1, 1);
plot(s, signal_second(a0, 0, a2,
s));
design_graph('t', 'x(t)',
'Reprezentarea grafica xpar(t)');

% x impar => a2 = 0  a1 = -a1  a0
= 0
subplot(2, 1, 2);
plot(s, signal_second(0, -a1, 0,
s));
design_graph('t', 'x(t)', 'Reprezentarea grafica ximpar(t)');
```

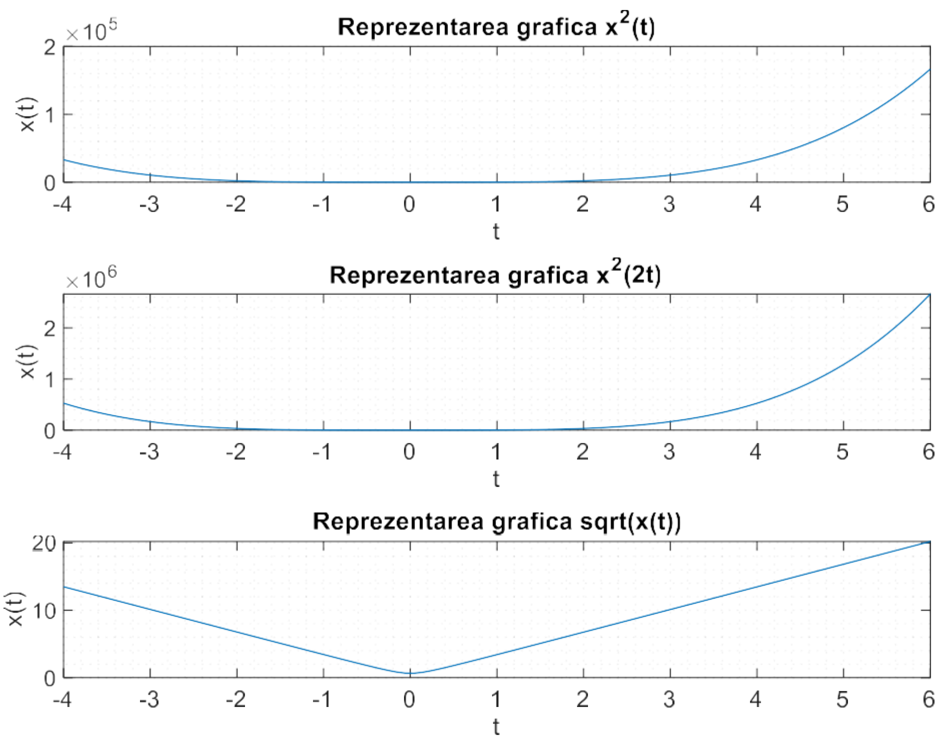
```
Signal_second.m

function [x] = signal_second(a0, a1, a2,
t)
% return the main signal any t
x = a2*(t.^2) + a1 * t + a0;
end
```



```
% x(2t)
subplot(3, 1, 1);
plot(s, signal_second(a0, a1, a2, 2*s));
design_graph('t', 'x(t)', 'Reprezentarea grafica x(2t)');

% x(2t-1)
subplot(3, 1, 2);
plot(s, signal_second(a0, a1, a2, (2*s -1)));
design_graph('t', 'x(t)', 'Reprezentarea grafica x(2t - 1)');

% x(t/2)
subplot(3, 1, 3);
plot(s, signal_second(a0, a1, a2, s/2));
design_graph('t', 'x(t)', 'Reprezentarea grafica x(t/2)');
```



```
% x^2(t)
subplot(3, 1, 1);
plot(s, signal_second(a0, a1, a2, s) .^ 2);
design_graph('t', 'x(t)', 'Reprezentarea grafica x^2(t)');

% x^2(2t)
subplot(3, 1, 2);
plot(s, signal_second(a0, a1, a2, 2 * s) .^ 2);
design_graph('t', 'x(t)', 'Reprezentarea grafica x^2(2t)');

% x(sqrt(abs(2t)))
subplot(3, 1, 3);
plot(s, sqrt(abs(signal_second(a0, a1, a2, s))));
design_graph('t', 'x(t)', 'Reprezentarea grafica sqrt(x(t))');
```



e) Calcul analitic componenta continua

e) $x(t) = a_2 t^2 + a_1 t + a_0, \quad t \in [0,1]$

$$C_0 = \frac{1}{T}\int_T x(t)\,dt = \frac{1}{T}\int_T (a_2 t^2 + a_1 t + a_0)\,dt =$$

$$= a_2 \frac{t^3}{3}\Big|_0^1 + a_1 \frac{t^2}{2}\Big|_0^1 + a_0 t$$

$$= 11{,}333 \cdot \frac{1}{3} - 0{,}0469 \cdot \frac{1}{2} + 0{,}4118$$

$$= 3{,}77766 - 0{,}02345 + 0{,}4118$$

$$= 4{,}166016$$

$x_1(t): \; T_1 = 0{,}5$

$$C_{01} = \left(11{,}333 \cdot \frac{(0{,}5)^3}{3} - 0{,}0469 \cdot \frac{(0{,}5)^2}{2} + 0{,}4118\right) \cdot \frac{1}{0{,}5} =$$

$$= (0{,}4722 - 0{,}0039 + 0{,}4118)\, 2 = 0{,}8801$$

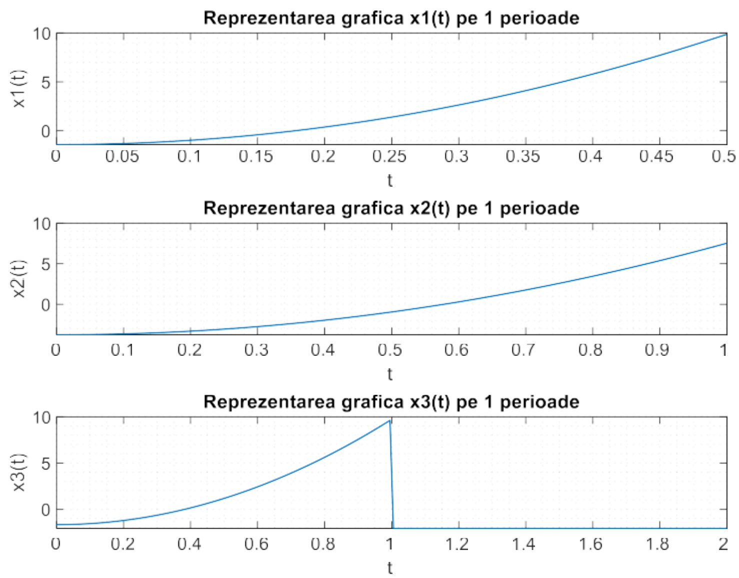
$x_2(t): \; T_2 = 1$

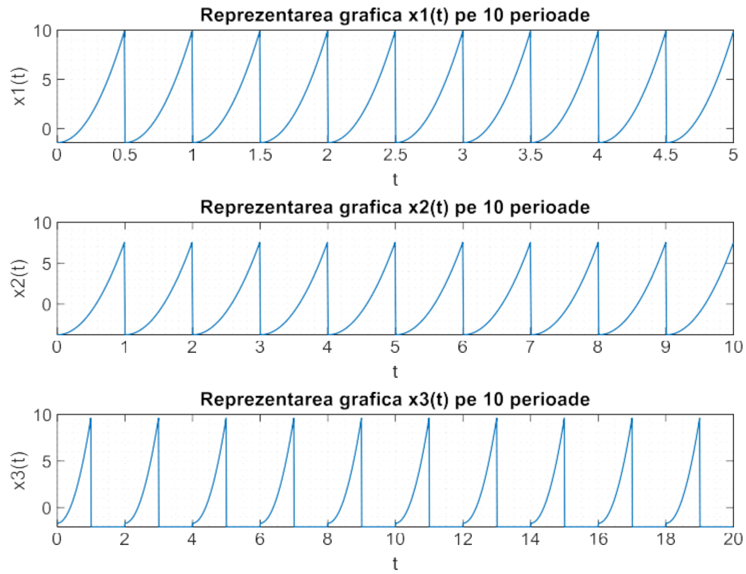
$$C_{02} = C_0 = 4{,}166016$$

$x_3(t): \; T_3 = 2 > 1 \Rightarrow C_{03} = \frac{1}{T_3} C_{01} = 2{,}083008$

f) xi(t), yi(t), zi(t) fara CC (yi, zi – impare => C0 = 0)

```
% xi - CC
for i = 4:5
    for j = 1:3
        name = 'x';
        signal = signal_main(a0, a1, a2, t1);
        display_signal_nocc(signal, T, j, i, name, number_of_periods, period,
@signal_xi);
    end
    input('next');
end
```





g) Calculul componentei continue in matlab

```
% cc: precision = 10^5
for i = 1:3
    disp(cc(signal_main(a0, a1, a2,
    precision_t), precision, T(i)));
end

input('next');
```

```
cc.m

function [cc] = cc(x, precision, T)
area = 0;
for i = 1:precision
    area = area + x(i);
end
cc = (1 / T) * area * (1/precision);
end
```

h) Pt

$$x(t) = a_2 t^2 + a_1 t + a_0 \; ; \; t \in [0,1]$$

$$P_T = \frac{1}{T_0} \int_{T_0} x^2(t)\,dt = \frac{1}{T_0} \int_{T_0} \left(a_2^2 t^4 + a_1^2 t^2 + a_0^2 + 2a_2 a_1 t^3 + 2a_0 a_2 t^2 + 2a_1 a_0 t \right) dt = \frac{1}{T_0} \left(a_2^2 \frac{t^5}{5} + a_1^2 \frac{t^3}{3} + a_0^2 t + 2a_2 a_1 \frac{t^4}{4} + 2a_0 a_2 \frac{t^3}{3} + 2a_1 a_0 \frac{t^2}{2} \right) \Big|_0^{T_0}$$

$$\Rightarrow P_T = 25,687 + 0,00073 + 0,16957 - 0,2657 + 3,112 - 0,0193$$

$$\Rightarrow P_T = 28,6843$$

$$P_{T_2} = P_T$$

$$P_{T_3} = \frac{P_{T_2}}{2} = \frac{P_T}{2} = 28,6843$$

$$P_{T_1} = 1,6054 + 0,00009 + 0,16957 - 0,0166 + 0,3889 - 0,531$$

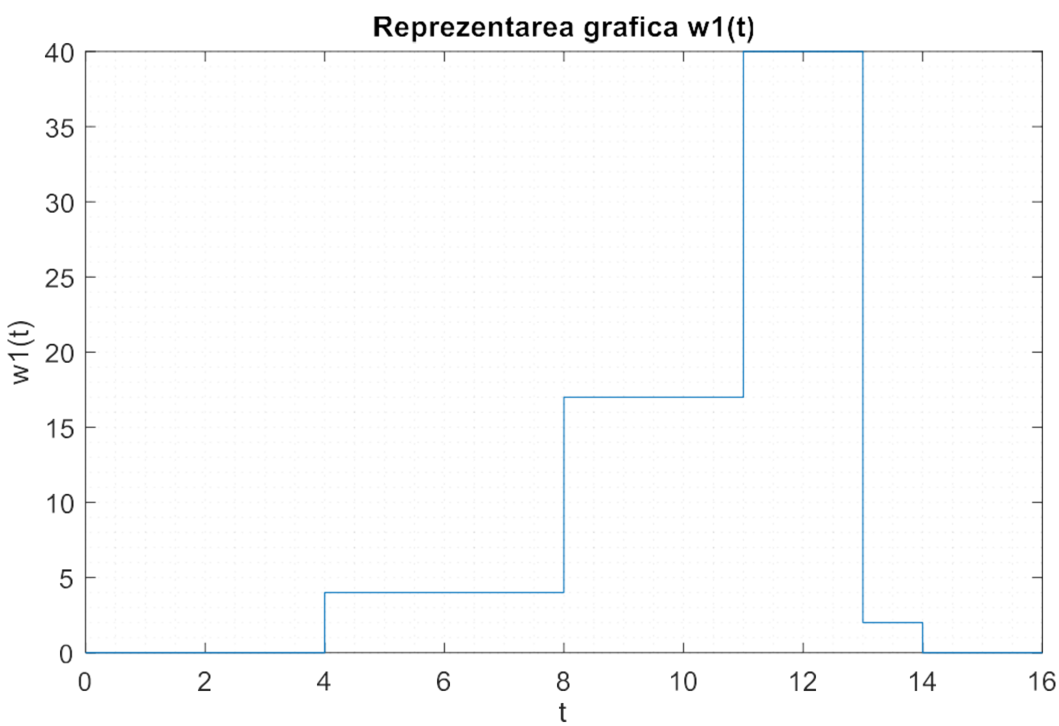
$$= 1,616$$

```
calculate_power.m

function [pt] = calculate_power(x, precision, T)
area = 0;
for i = 1:precision
    area = area + (x(i).^2);
end
pt = (1 / T) * area * (1/precision);
end
```

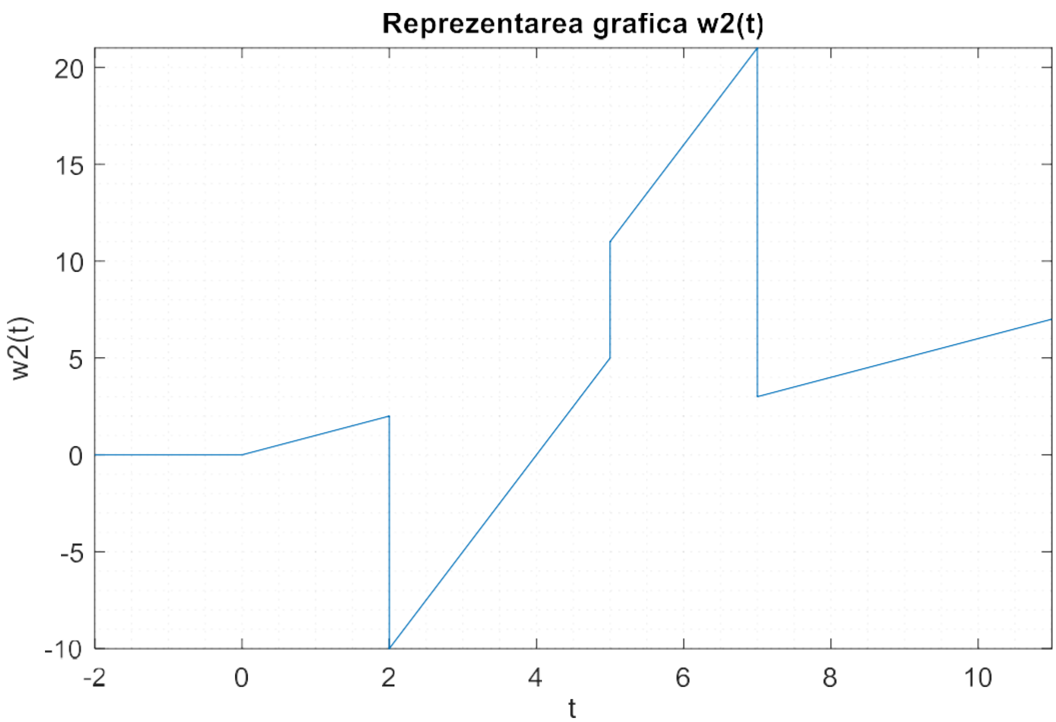
```
% pt: precision = 10^4
for i = 1:3

disp(calculate_power(signal_main(a0, a1, a2, pprecision_t), pprecision, T(i)));
end
```

i) w1(t), w2(t), w3(t)

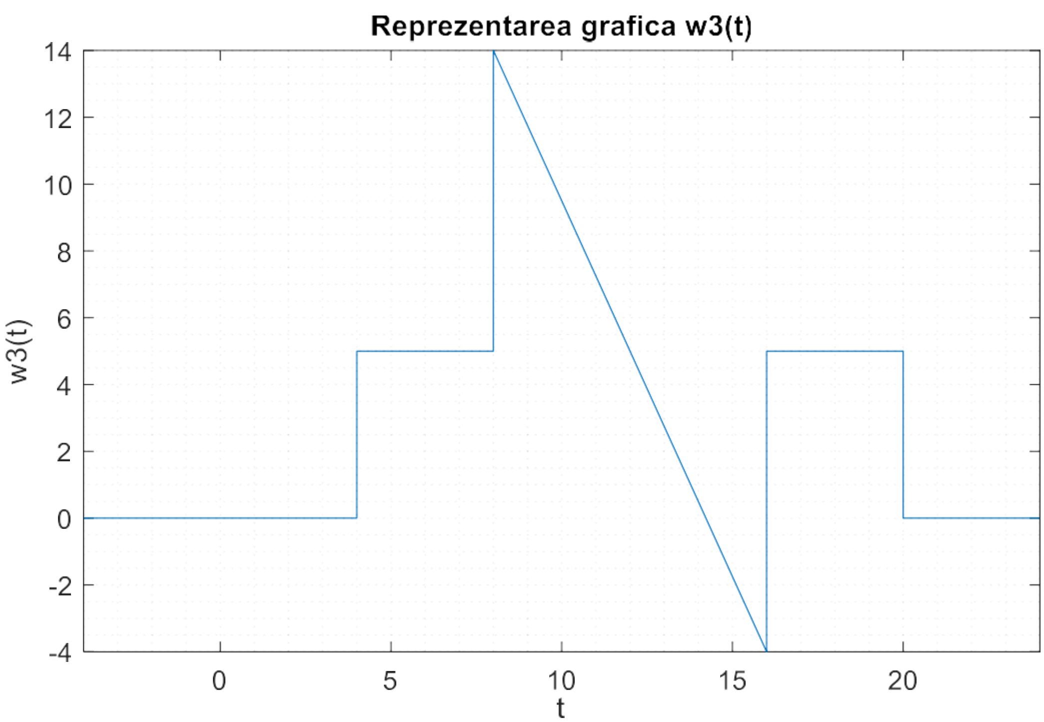
```
% w1(t)
syms t;
tw1 = [n-4, n+12];
w1 = @(t) n*heaviside(t-n) - (n-17)*heaviside(t-n-4) + (2*n+15)*heaviside(t-n-7) - (30+2*n)*heaviside(t-n-9) - 2*heaviside(t-n-10);
fplot(w1(t), tw1);
y_label = 'w1(t)';
title_ = 'Reprezentarea grafica w1(t)';
design_graph('t', y_label, title_);
```



```
% w2(t)
syms t;
tw2 = [-2, 11];
w2 = @(t) t*heaviside(t) + (n*t-20)*heaviside(t-2) - (n-10)*heaviside(t-5) - (n*t-
10)*heaviside(t-7);
fplot(w2(t), tw2);
y_label = 'w2(t)';
title_ = 'Reprezentarea grafica w2(t)';
design_graph('t', y_label, title_);
```



```
% w3(t)
syms t;
tw3 = [-n, 6*n];
w3 = @(t) 5*heaviside(t-n) - 9*((t-3*n)/n)*(heaviside(t-2*n)-heaviside(t-3*n)) +
9*(1+(2*n-t)/n)*(heaviside(t-3*n)-heaviside(t-4*n)) - 5*heaviside(t-5*n);
fplot(w3(t), tw3);
y_label = 'w3(t)';
title_ = 'Reprezentarea grafica w3(t)';
design_graph('t', y_label, title_);

input('next');
```



Concluzii

In acest proiect am reprezentat grafic, cu ajutorul platformei Matlab, semnale in diferite conditii. Cele 9 semnale initiale au fost ulterior modificate conform cerintelor, schimband nu numai definitia acestora, dar si suportul pe care au fost reprezentate.

Limbajul de programare Matlab are functionalitati datorita carora gestionarea semnalelor poate fi simplificata. Un exemplu este variabila simbolica, care este foarte folositoare in definirea unui semnal dependent de timp.

De asemenea, Matlabul ofera diferite metode de citire si prelucrarea a datelor, lucru care confera libertatea de a manevra cantitati mari de informatie.

Reprezentarea vizuala a unei notiuni initial abstracte este un pas foarte important spre intelegerea cu adevarat a unui concept. Imbinarea semnalelor cu programarea antreneaza doua dintre calitatile principale necesare unui electronist.

In concluzie, in urma realizarii acestui proiect, am aprofundat informatiile deja cunoscute despre semnale si sisteme si am dezvoltat abilitatile de programare, in special programarea functionala.

Bibliografie

https://www.mathworks.com/help/matlab/
https://www.wikipedia.org/
https://www.techopedia.com/definition/14154/signal-electronics
https://www.techtarget.com/searchnetworking/definition/signal
Cursul de „Semnale si Sisteme”