

## **Obiectivele lucrării de laborator:**

- Descrierea programelor utilizate: Matlab, Orcad/PSpice
- Editarea unei scheme electrice
- Utilizarea funcțiilor imread/imshow

### **1.1 Introducere în Matlab.**

**MATLAB (Matrix Laboratory) - The Language of Technical Computing**” ceea ce se traduce prin „**MATLAB – Limbajul calculelor tehnice**” este un mediu de dezvoltare pentru calcul numeric și analiză statistică care conține limbajul de programare cu același nume, creat de **MathWorks** și dezvoltat pornind de la două pachete de programe, LINPACK și EISPACK, reprezentând bazele soft-ului pentru calcul matriceal.

Astfel, Matlab este un mediu interactiv, de înaltă performanță, utilizat pentru calcule științifice și ingineresti, ce înglobează diverse posibilități de lucru, cum ar fi analiza numerică, calculul matriceal, procesare de semnale și trasarea grafică, într-un mediu ușor de utilizat, enunțurile și soluțiile fiind exprimate exact cum sunt scrise matematic, nefiind necesară o programare tradițională.

Elementele de bază ale pachetului MATLAB sunt matricile, nefiind necesară o dimensionare a acestora. Acest mediu de programare este util în rezolvarea problemelor numerice iar resursele sale de calcul și reprezentare grafică sunt bogate, permițând operații matematice fundamentale, analiza datelor, programare, reprezentări grafice 2D și 3D, realizarea de interfețe grafice etc.

Pachetul de programe MATLAB a cunoscut o puternică evoluție în decursul ultimilor ani, reprezentând astăzi în mediile universitare o unealtă standard de calcul, fiind asociată diverselor cursuri introductive sau avansate în matematică, știință și inginerie. În industrie, MATLAB este recunoscut ca un mijloc de investigație numerică performant, utilizat în sprijinul unei activități de cercetare, dezvoltare și analiză de înalt nivel.

## 1.2 Lansarea aplicației

Programul Matlab se lansează la fel ca orice alt program instalat sub Windows, GNU/Linux, UNIX și Mac OS, fie prin dublu click pe icon-ul caracteristic programului, fie prin lansarea acestuia dintr-o linie de comandă. În cele ce urmează se vor prezenta ferestrele de lucru la rularea programului, explicând rolul fiecăreia.

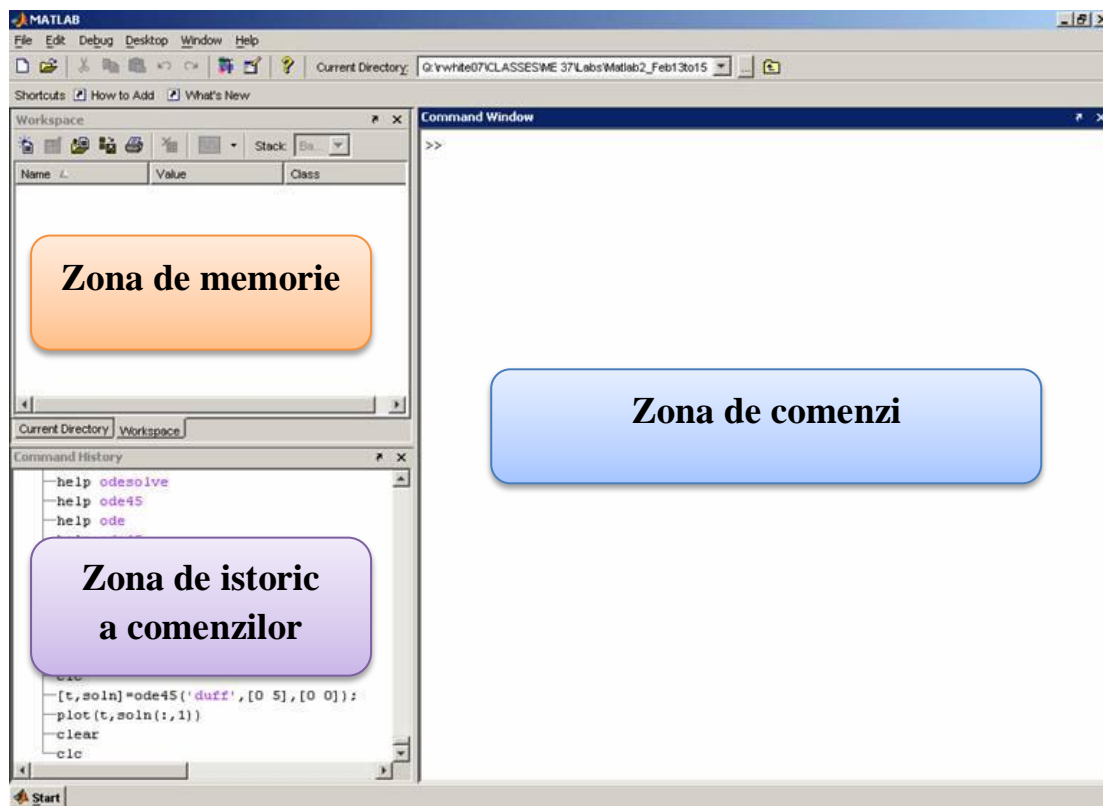


Figura 1.1 Interfața programului Matlab

Așa cum se poate observa și în figura 1.1 la rularea programului acesta returnează o interfață ce cuprinde 3 ferestre de interes.

- Fereastra denumită sugestiv **zona de comenzi** se caracterizează prin faptul că programul acceptă introducerea după prompterul ”>>” a unor linii de cod (comenzi) pe care programul urmează să le execute.
- Fereastra denumită **zona de istoric a comenzilor** conține toate comenzile date pe parcursul unei sesiuni de lucru, adică între deschiderea și închiderea programului. Mai mult conține toate comenzile din sesiunile trecute, fiecare sesiune fiind despărțită de precedenta printr-un rând ce conține data și ora deschiderii sesiunii respective. Această zonă permite utilizarea unor comenzi care au fost date în sesiuni anterioare reducând astfel timpul de scriere a unei secvențe.
- Fereastra denumită **zona de memorie** conține informații despre toate datele, variabilele, structurile încărcate în memorie în acea sesiune, ușurând astfel depanarea sau verificarea acestora în cadrul unor proiecte ample.
- Există o a patra fereastră denumită **fereastră grafică** ce se deschide doar dacă este dată o comandă specifică reprezentărilor grafice.

### **1.3 Variabilele, constantele, funcții generale și caracterele speciale în Matlab**

În cadrul unei sesiuni MATLAB, putem defini variabile de tip constante numerice, vectori sau matrici, dându-le nume diferite și ținând cont de faptul că programul MATLAB este "case-sensitive" (face diferențierea între variabila ex: a și A).

Introducerea unei operații în linia de comandă determină afișarea rezultatului sau acțiunii liniei introduse. Astfel, dacă rezultatului liniei introduse nu i s-a desemnat un nume prin atribuiri explicite va fi afișat numele "ans" (answer) ce desemnează ultimul rezultat nenominalizat.

- Exemplu: >>     a=10  
                  a =  
                  10  
>> 7  
                  ans =  
                  7

Așadar o serie de variabile pe care programul Matlab le poate returna în linia de comandă sunt prezentate în tabelul 1.1.

<b>Constanta</b>	<b>Explicația constantei</b>
<b>ans</b>	Variabilă creată de Matlab cu ajutorul căreia returnează valoarea unei operații: <pre>&gt;&gt; 8+5 ans =     13</pre>
<b>pi</b>	Returnează valoarea lui $\pi$ : <pre>&gt;&gt; pi ans =     3.1416</pre>
<b>i sau j</b>	Variabilă folosită la scrierea numerelor complexe. Unitate imaginară.
<b>inf</b>	Variabilă folosită pentru reprezentarea lui $+\infty$ . <pre>&gt;&gt; 1/0 ans =     inf</pre>
<b>NaN</b>	Not-A-Number. Variabilă returnată în cazul în care un număr nu poate fi returnat: <pre>&gt;&gt; 0/0 Ans =     NaN</pre>
<b>eps</b>	Variabilă în care este memorată eroarea relativă pentru calcule efectuate în virgulă mobilă.
<b>flops</b>	Numărul de operații în virgulă mobilă.
<b>realmax / realmin</b>	Valoarea maximă/minimă în virgulă mobilă.
<b>nargin / nargout</b>	Numărul de argumente de intrare/ieșire a unei funcții.

Tabelul 1.1. Constante returnate de Matlab

În timpul unei sesiuni de lucru se pot verifica variabilele și starea acestora precum și formatul lor sau documentația unei funcții utilizând una din funcțiile predefinite în zona de comandă a interfeței programului.

<b>Funcția</b>	<b>Explicația funcției</b>		
<b>help</b>	Oferă explicații despre elementele limbajului Matlab.		
<b>help nume_funcție</b>	Returnează informații despre funcția nume_funcție. <pre>&gt;&gt; help sin SIN   Sine of argument in radians. SIN(X) is the sine of the elements of X.</pre>		
<b>who</b>	Afișază numele variabilelor utilizate în sesiunea curentă. <pre>&gt;&gt; a=7; &gt;&gt; b=10; &gt;&gt; c=a+b; &gt;&gt; who Your variables are: a b c</pre>		
<b>whos</b>	Afișază informații suplimentare asupra variabilelor din sesiunea curentă. <pre>Name      Size      Bytes Class  Attributes a         1x1         8 double b         1x1         8 double c         1x1         8 double</pre>		
<b>format</b>	Stabilește formatul de afișare:		
	<b>short</b>	<b>long</b>	<b>Hex</b>
	<pre>&gt;&gt; format short &gt;&gt; pi ans =     3.1416</pre>	<pre>&gt;&gt; format long &gt;&gt; pi ans = 3.141592653589793</pre>	<pre>&gt;&gt; format hex &gt;&gt; 96586 ans = 40f794a000000000</pre>

Tabel 1.2. Funcții predefinite asupra variabilelor

Caracterele speciale pe care programul le recunoaște sunt prezentate în tabelul 1.3.

<b>Caracterul</b>	<b>Atributul</b>
:	Se folosește la generarea diviziunilor
()	Folosite pentru corpul unei funcții
[]	Pentru desemnarea unei matrici / vector
@	Simbol pentru anunțarea unei funcții de tip FUNCTION_HANDLE
.	Indicator de operație aritmetică element cu element
...	Continuarea unei comenzi Matlab pe linia de comandă următoare
,	Separator între instrucțiuni pe aceeași linie de comandă
;	Separator între instrucțiuni pe aceeași linie de comandă. În cazul utilizării acestui caracter la finalul unei linii în zona de comandă nu se va returna rezultatul acelei linii.
'	Transpusa unei matrici
[,]	Separator între elementele aceleiași linii într-o matrice/vector
[;]	Separator între liniile unei matrici

Tabel 1.3. Caractere special recunoscute de Matlab

### 1.4 Operatori aritmetici în Matlab

<b>Operatorul</b>	<b>Operația</b>
+	Adunare
-	Scădere
*	Înmulțire
/	Împărțire
\	Împărțire la stânga
^	Ridicarea la o putere
'	Transpusa complex conjugată

Tabel 1.4. Operatori aritmetici

## 1.5 Introducere în Orcad Pspice

**Spice** (Simulation Program with Integrated Circuit Emphasis) este un program de simulare a circuitelor electronice orientat pe simularea structurilor de circuite integrate dezvoltat la începutul anilor '70 la University of California, Berkeley fiind în prezent unul din cele mai utilizate simulatoare în domeniul ingineriei electronice și nu numai, atât în cadrul universitar, de cercetare ori industrie.

Lansarea programului se va face fie din meniul Windows-**Start**-Program Files-Orcad 9.2-Capture CIS, fie prin dublu click pe icon-ul caracteristic aplicației. Aplicația Capture CIS este un integrator al modulelor pachetului de programare OrCAD:

- Express și Layout pentru editarea circuitelor electronice și proiectarea cablajelor;
- Pspice cu componentele:
  - Pspice AD - simulatorul Pspice;
  - Pspice Model Editor – editorul de modele Pspice;
  - Pspice Optimizer – programul de optimizare a circuitelor electronice;
  - Pspice Stimulus Editor – editorul de stimuli ai programului Pspice.

Aceste fișiere au ca punct comun fișierul schematic al circuitului electronic, care se editează cu componenta Express și generează un fișier de legături specific fiecărui modul al pachetului OrCAD.

## 1.6 Realizarea unui proiect nou în Pspice

Pentru a crea un proiect de simulare Pspice, din meniul principal al aplicației OrCAD Capture se va selecta File-New-Project. În fereastra de dialog, care va apare în urma selectării opțiunii de creare a unui nou proiect, se va specifica numele acestuia, tipul și calea directorului unde se dorește salvarea acestuia. Figura 1.2 prezintă etapele inițiale de realizarea a unui proiect.

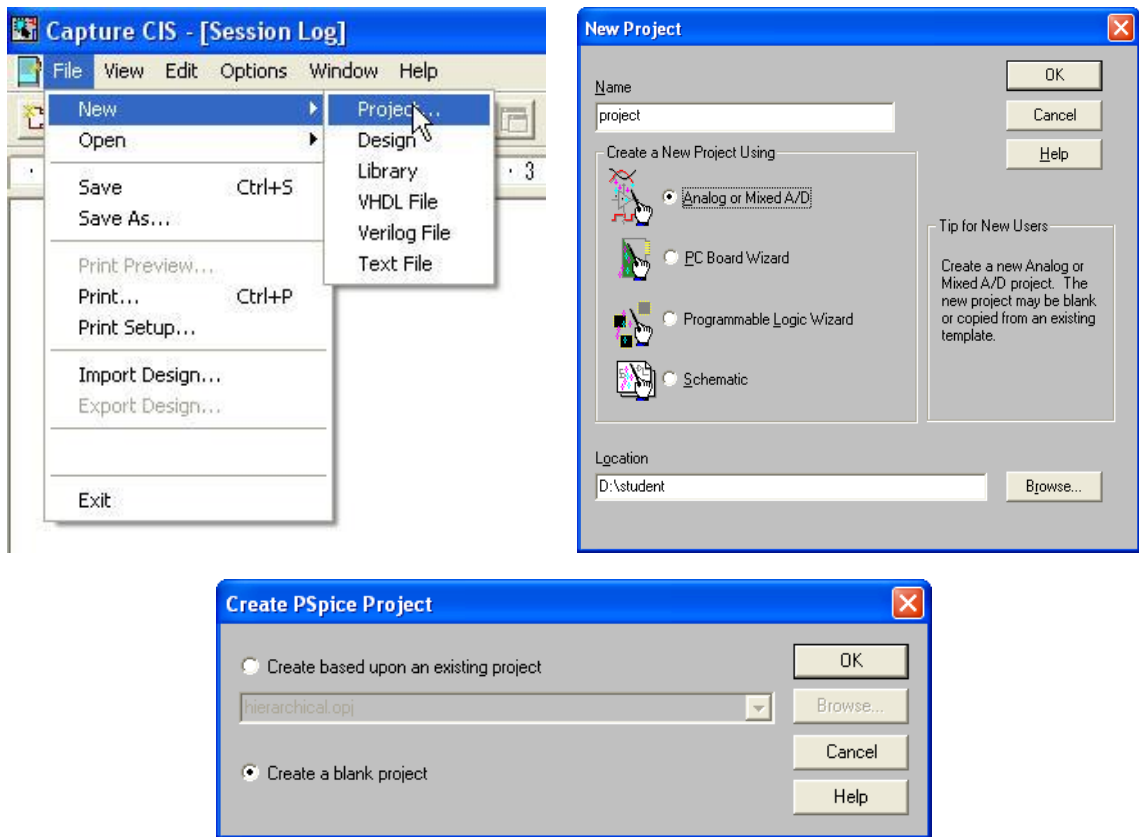


Figura 1.2. Crearea unui nou proiect în Pspice

Pentru tipul proiectului se poate observa că există mai multe categorii. Prima opțiune ne permite editarea schemelor electronice și simularea acestora, a doua este exclusiv pentru proiectarea cablajelor imprimate, a treia pentru simulare circuitelor digitale, iar a patra pentru realizarea unui proiect fără o anumită specificație. Astfel, pentru a obține un proiect de simulare este obligatorie specificația Analog or Mixed-Signal Circuit Wizard. Specificarea acestor date determină lansarea în execuție a unui modul Express modificat pentru editarea și simularea circuitelor, iar meniul principal este prezentat în figura 1.3.



Figura 1.3. Meniul principal al unui proiect PSpice



## 1.7 Editarea circuitelor electronice

Pentru editarea schemei electrice este nevoie de cunoașterea câtorva etape:

- Plasarea pe suprafața de lucru a simbolurilor specifice componentelor;
- Căutarea unor componente în bibliotecile programului;
- Adăugarea sau eliminarea unor biblioteci;
- Editarea proprietăților componentelor;
- Trasarea firelor de legătură între componente;
- Operații speciale.

Pentru amplasarea unui simbol se va alege din meniul principal opțiunea Place-Part sau, de pe bara de instrumente din dreapta, simbolul corespunzător operației Place-Part, ca în figura 1.4. Plasarea unei componente pe spațiul de lucru implică specificarea numelui componentei dorite și a bibliotecii din care aceasta face parte.

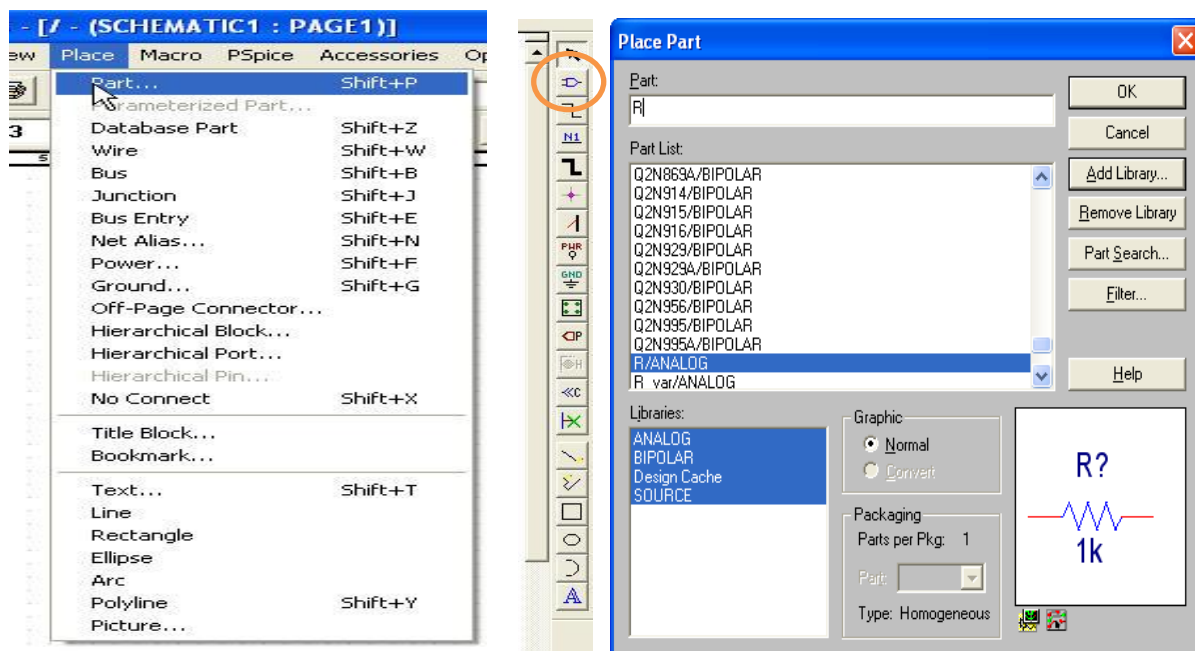


Figura 1.4. Meniul de plasare a componentelor și fereastra de dialog Place Part.

Fiecare componentă are un anumit prefix de identificare cu ajutorul căruia le vom căuta și astfel, plasa pe spațiul de lucru. Pentru cele mai uzuale componente prefixele de căutare sunt:

- R – pentru rezistoare;
- C – pentru condensatoare;
- L – pentru bobine;
- V – pentru surse de tensiune;
- I – pentru surse de curent;
- GND – pentru masa analogică (0/SOURCE);
- Q – pentru tranzistoare bipolare;
- D – pentru diode.

Orice altă componentă ce se dorește a fi adăugată se plasează fie prin prefixul acesteia, fie prin codurile specificate de producător. Pentru adăugarea sau eliminarea unei biblioteci din lista de căutare se vor folosi butoanele Add/Remove Library, iar căutarea unei componente se realizează prin selectarea butonul Part Search din fereastra de dialog Place Part.



Figura 1.5. Adăugarea/Ștergerea unei biblioteci de componente

Editarea proprietăților unei componente amplasate pe suprafața de lucru se realizează fie prin selectarea Edit Properties după ce aceasta fost selectată și s-a executat un click dreapta pe ea, fie prin dublu click pe valoarea sau numele componentei.

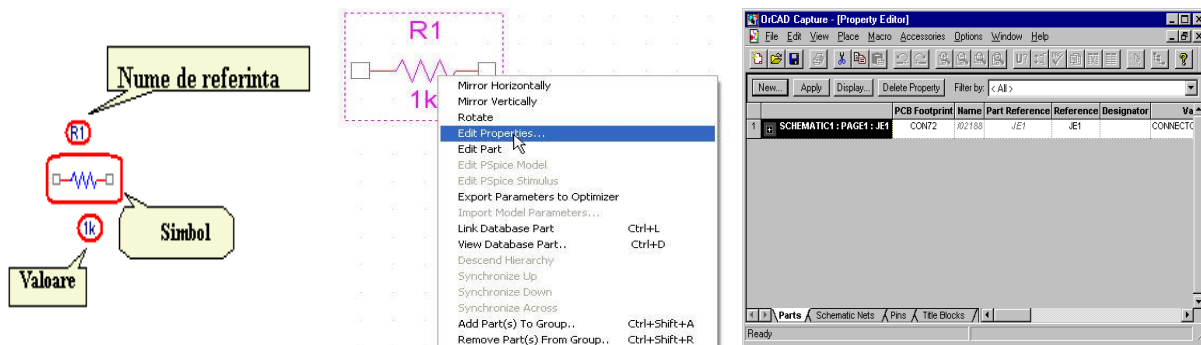


Figura 1.6. Editarea unei componente

Valorile numerice se introduc precizînd valoarea numerică și apoi ordinul de mărime printr-o literă, conform tabelului alăturat.

Valoare numerică	Semnificație
1n	1 nano = $10^{-9}$
1u	1 micro = $10^{-6}$
1m	1 mili = $10^{-3}$
1,10,100	1,10,100
1k	1kilo = $10^3$
1meg	1mega = $10^6$

Tabel 1.5. Unitățile de măsură

Odată ce componentele necesare realizării schemei propuse au fost amplasate și aranjate, realizarea legăturilor între acestea se face prin selectarea butonului Place Wire din fereastra din dreapta spațiului de lucru, sau folosind scurtătura "W" de la tastatură. Funcționarea oricărui circuit electronic este condiționat de alimentarea acestuia. Totodată este necesar să existe o masă analogică a circuitului, masă ce se amplasează prin selectarea bunonului GND din meniul dreapta și se află în biblioteca SOURCE sub denumirea de 0/SOURCE.

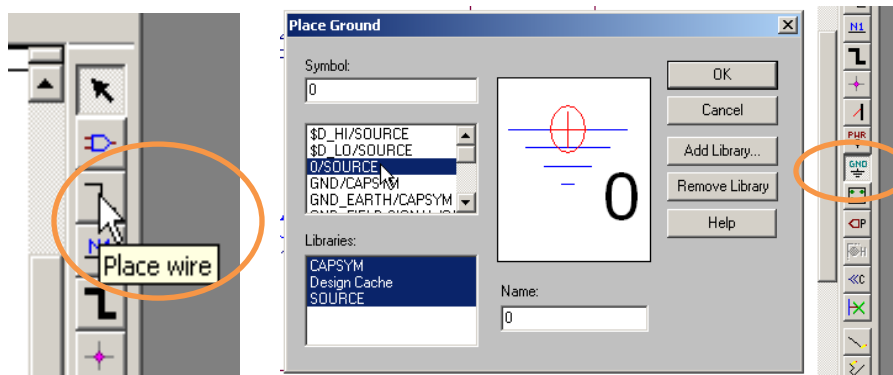


Figura 1.7. Place wire și place GND pe schema electronică

## 1.8 Desfășurarea lucrării de laborator

1. Se editează cu ajutorul OrCad PSpice schema din figura 1.8.

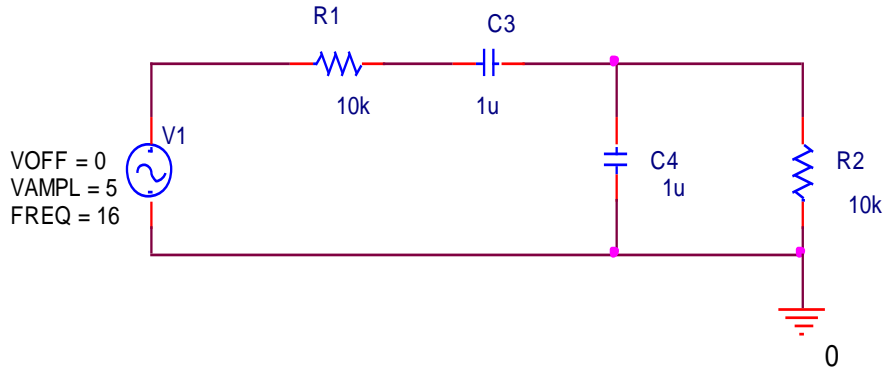


Figura 1.8. Filtru trece bandă

2. Se salvează schema rezultată sub forma unei imagini cu una din extensiile cunoscute: .bmp, .png, .jpeg, .jpg, etc.

Punctul 1 se realizează urmând pașii de editare a unei scheme electronice explicată la punctul 1.7, iar punctul 2 cu ajutorul cunoștințelor de windows dobândite până acum.

3. Se lansează programul Matlab și se crează un fișier de tip M-File în corpul căruia vom crea o nouă figură.

Crearea unui fișier de tip M-File în Matlab se realizează din meniul programului File -> New -> M-File, așa cum se prezintă în figura 1.9.

Aceste fișiere de tip M-File, sunt fișiere text ce conțin instrucțiuni MATLAB. Folosind **MATLAB Editor** pentru a crea un astfel de fișier, instrucțiunile pe care le-ați tasta de la linia de comandă MATLAB în **fereastra de comenzi** sunt acum parcurse secvențial și returnează rezultatul secvenței de cod scrise. Un fișier de tip M-File este de forma: nume și extensia **.m**. (ex: **laborator1.m**)

În Matlab vom defini două astfel de fișiere: fișierele de tip **Script**, cele în interiorul cărora vom rezolva diverse probleme sau vom atribui variabilelor valori; și fișiere de tip **Funcție**, fișiere ce pot fi apelate din interiorul fișierelor de tip Script.

Distincția dintre cele două fișiere constă în prezența pe prima linie a editorului în cazul fișierelor de tip Funcție, a sintaxei **function(parametru)**. Aceste tipuri de fișiere vor fi prezentate pe larg în lucrările anterioare.

**ATENȚIE!** Fișierele de tip M-File se salvează cu un nume format din litere, litere și cifre, dar niciodată doar din cifre, sau cu denumiri a unor funcții predefinite în Matlab. Apelarea acestor fișiere se face din fereastra de comenzi scriind numele fișierului ce se dorește a fi apelat.



Figura 1.9 Crearea unui fișier M-File

Pentru a crea o nouă figură (spațiu de lucru / fereastră) este necesar a se folosi funcția predefinită **figure**, iar pe liniile editorului vom scrie următoarele instrucțiuni:

```
clear all;
close all;
```

```
Fig=figure('Name','Figura Noua Matlab',... %---Numele ferestrei---%
           'Units','normalized',...%---Unitatile de masura - 0->1-%
           'Position',[.1 .1 .5 .5],...%---Pozitia ferestrei pe ecran
           'NumberTitle','off'); %---Numarul ferestrei - nu se atribuie---%
```

Rezultatul liniilor de cod de mai sus returnează o nouă fereastră ca cea prezentată în figura 1.10. Funcția **clear** șterge toate variabilele încărcate anterior, iar funcția **close** închide toate ferestrele ce au fost deschise înainte de rularea aplicației curente.

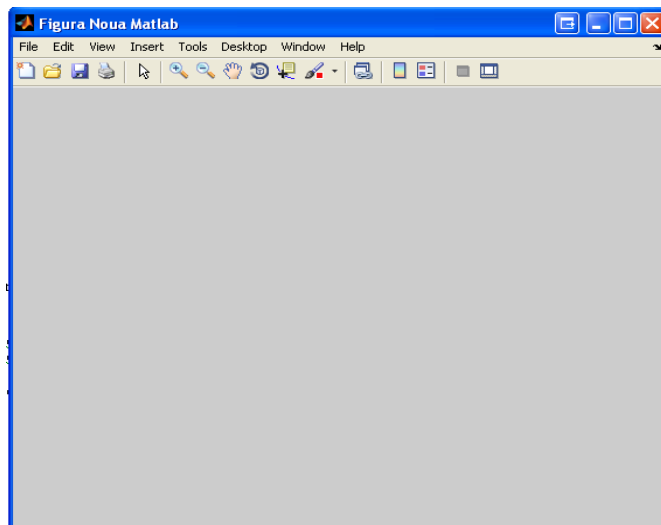


Figura 1.10. O fereastră creată în Matlab

4. În figura nou creată la punctul anterior se dorește afișarea imaginii salvate la punctul 2.

Afișarea imaginii realizate la punctul 2 în fereastra de lucru creată se realizează cu ajutorul funcțiilor *image* sau *imshow*.

Astfel liniilor de cod anterioare se adaugă:

```
x = imread('schema.jpg');    %---citirea imaginii---%
image(x);                    %---afisarea imaginii---%
```

Rezultatul scrierii acestor linii de cod in continuarea liniilor de la punctul 3 returnează o fereastră de forma celei prezentate în figura 1.11.

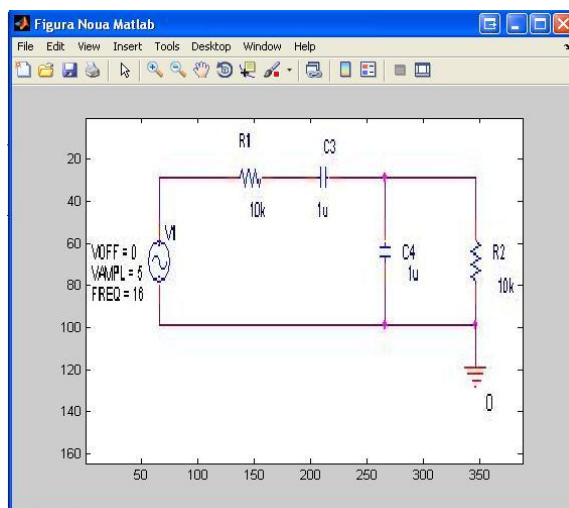


Figura 1.11. Rezultatul utilizării funcției *image*

**ATENȚIE!** Exemplul prezentat implică salvarea fișierului de tip M-File și a imaginii salvate cu numele **schema.jpg** în același director de lucru. În cazul în care imaginea ce se dorește a fi atribuită unei astfel de ferestre nu se regăsește în directorul de lucru curent, sintaxa funcției **imread** se modifică astfel încât să atribuim calea unde se regăsește imaginea dorită.

```
x = imread('C:\Desktop\schema.jpg');    %---citirea imaginii---%
    image(x);                            %---afisarea imaginii---%
```

O alternativă la funcția `image` este funcția `imshow`, funcție prezentată prin următoarele linii de cod:

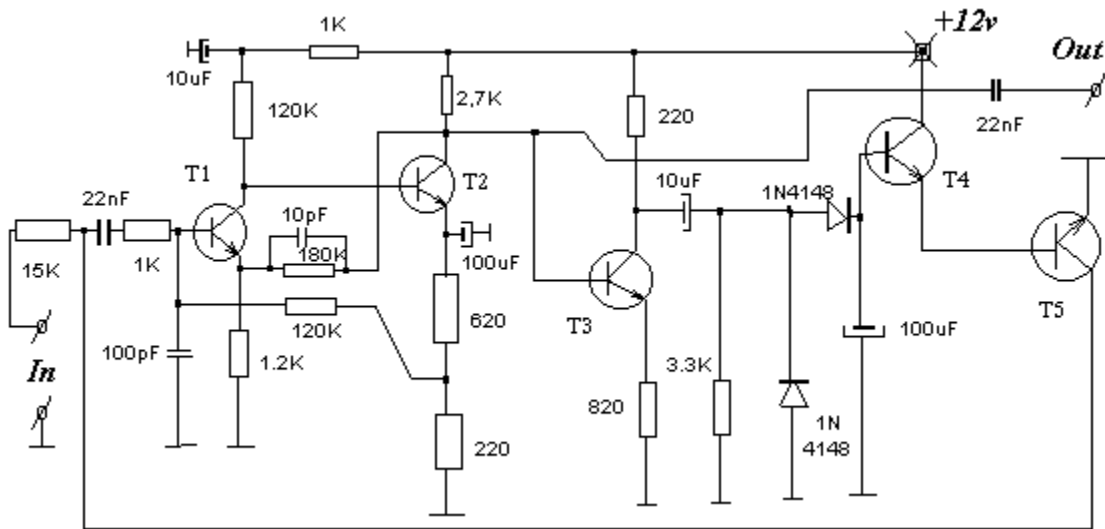
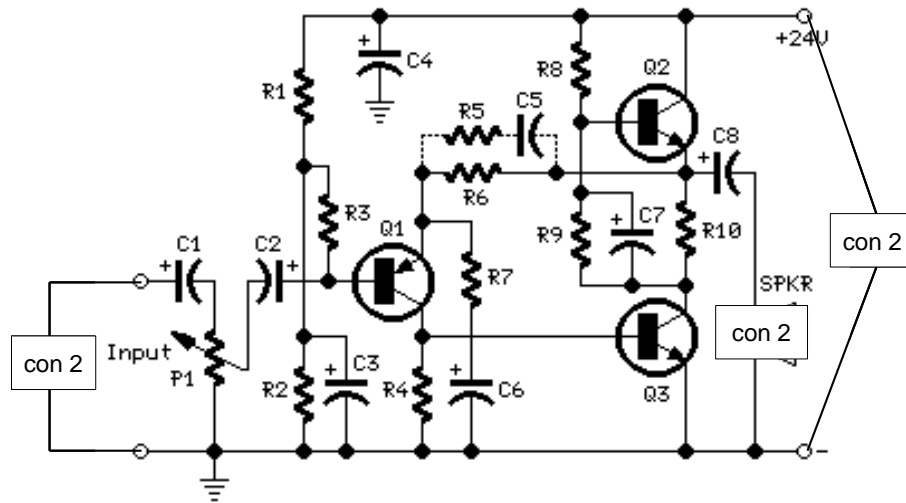
```
w = imread('schema.jpg');                %---citirea imaginii---%
    imshow(w, 'InitialMagnification',150) %---afisarea imaginii---%
```

5. Modificați parametri figurii create la punctul 3 astfel: culoare, dimensiune, nume, etc.
6. Identificați diferențele dintre funcțiile `image` și `imshow`.
7. Personalizați axele ferestrei în care afișați imaginea de la punctul 2 și dați un nume acesteia.

Punctele 5, 6, 7 se rezolvă utilizând proprietățile funcțiilor *figure*, *image*, *imshow* și *axis*. Proprietățile acestor funcții se pot obține utilizând fie comanda **help** în linia de comandă, fie prin apăsarea butonului `help` din meniul programului.

### 1.10 Temă

- Se vor repeta procedurile din laborator pentru schemele propuse din figurile următoare:



- Identificați parametrii funcțiilor utilizate în acest laborator. (**figure**, **imread**, **imshow**, **image**)