Laplaceova transformacija u programskom sustavu MATLAB

Car, Antonio

Undergraduate thesis / Završni rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Rijeka, Faculty of Engineering / Sveučilište u Rijeci, Tehnički fakultet**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:190:870330

Rights / Prava: Attribution 4.0 International/Imenovanje 4.0 međunarodna

Download date / Datum preuzimanja: **2024-05-19**

Repository / Repozitorij:

Repository of the University of Rijeka, Faculty of Engineering

SVEUČILIŠTE U RIJECI

TEHNIČKI FAKULTET

Preddiplomski sveučilišni studij elektrotehnike

Završni rad

LAPLACEOVA TRANSFORMACIJA U PROGRAMSKOM SUSTAVU MATLAB

Rijeka, srpanj 2022.

Antonio Car
0069084247

SVEUČILIŠTE U RIJECI

TEHNIČKI FAKULTET

Preddiplomski sveučilišni studij elektrotehnike

Završni rad

LAPLACEOVA TRANSFORMACIJA U PROGRAMSKOM SUSTAVU MATLAB

Mentor: doc. dr. sc. Ivan Dražić

Rijeka, srpanj 2022.

Antonio Car
0069084247

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
POVJERENSTVO ZA ZAVRŠNE ISPITE

Rijeka, 18. ožujka 2022.

Zavod: **Zavod za matematiku, fiziku i strane jezike**
Predmet: **Inženjerska matematika ET**
Grana: **1.01.07 primijenjena matematika i matematičko modeliranje**

ZADATAK ZA ZAVRŠNI RAD

Pristupnik: **Antonio Car (0069084247)**
Studij: Preddiplomski sveučilišni studij elektrotehnike

Zadatak: **Laplaceova transformacija u programskom sustavu MATLAB // Laplace transform in the MATLAB software system**

Opis zadatka:

U radu je potrebno definirati Laplaceovu transformaciju te navesti i objasniti njena temeljna svojstva, kao i najčešće primjene s naglaskom na primjene u elektrotehnici.
U drugom dijelu rada potrebno je dati kratki uvod u programski jezik MATLAB te detaljno opisati biblioteku naredbi koja se koristi kod primjene Laplaceovih transformacija.
Poseban naglasak treba staviti na klasične primjene Laplaceovih transformacija u elektrotehnici i njihovu implementaciju u MATLAB-u, pri čemu je potrebno riješiti nekoliko povezanih primjera.

Rad mora biti napisan prema Uputama za pisanje diplomskih / završnih radova koje su objavljene na mrežnim stranicama studija.

Antonio Car
Zadatak uručen pristupniku: 21. ožujka 2022.

Mentor:

Doc. dr. sc. Ivan Dražić

Predsjednik povjerenstva za završni ispit:

Prof. dr. sc. Viktor Sučić

Scanned with CamScanner

IZJAVA

Sukladno članku 8. Pravilnika o završnom radu, završnom ispitu i završetku prediplomskih sveučilišnih studija/stručnih studija Tehničkog fakulteta Sveučilišta u Rijeci od 1. veljače 2020., izjavljujem da sam samostalno izradio završni rad prema zadatku preuzetom dana 21. ožujka 2022.

Rijeka, 01. srpnja 2022.

Antonio Car

Antonio Car

Scanned with CamScanner

Želim zahvaliti svojoj obitelji koja je od prvog do posljednjeg dana studiranja bila moja podrška. Također, velika zahvala doc. dr. sc. Ivanu Dražiću koji me tijekom pisanja rada usmjeravao te mi maksimalno olakšao da prijeđem ovu životnu stepenicu.

Sadržaj

1. Uvod **3**

2. Povijesni razvoj MATLAB-a **4**

2.1. Prva verzija MATLAB-a 4

2.2. Komercijalni MATLAB 5

2.3. Moderni MATLAB 7

2.3.1. Alati u modernom MATLABu 10

2.3.2. Biblioteke u MATLAB-u 10

2.3.3. Simbolički izračun u MATLABu 11

2.3.4. Primjena MATLAB-a u struci 15

3. Osnove korištenja MATLAB-a **16**

3.1. Varijable 16

3.1.1. Interne varijable 16

3.1.2. Eksterne varijable 17

3.1.3. Definiranje varijabli 17

3.2. Funkcije 18

3.2.1. Funkcije sa jednim izlazom 19

3.2.2. Funkcije s više izlaza 19

3.2.3. Zadavanje simboličkih matematičkih funkcija 20

3.2.4. Crtanje funkcija 22

3.2.5. Simboličko rješavanje jednadžbi i uvrštavanje rješenja u neku drugu funkciju 25

3.2.6. Simboličko rješavanje diferencijalnih jednadžbi i grafički prikaz rješenja . . 26

4. Laplaceove transformacije **28**

4.1. Povijest Laplaceovih transformacija 28

4.2. Egzistencija Laplaceove transformacije 29

4.3. Svojstva Laplaceove transformacije 30

4.4. Primjena Laplaceove transformacije 34

5. Laplaceove transformacije u programskom sustavu MATLAB **35**

5.1. Određivanje Laplaceove slike i originala u MATLAB-u 35

5.2. Primjeri određivanja Laplaceove transformacije i inverzne Laplaceove transformacije u MATLAB-u 36
5.3. Analiza RC kruga tehnikom Laplaceovih transformacija 39
5.4. Analiza RL kruga tehnikom Laplaceovih transformacija 46
5.5. Analiza RLC kruga tehnikom Laplaceovih transformacija 52

6. Zaključak **58**

Literatura **59**

Sažetak i ključne riječi **60**

Summary and key words **61**

1. Uvod

Laplaceova transformacija integralna je transformacija koja se danas široko primjenjuje u elektrotehnici, posebice prilikom rješavanja diferencijalnih i integralnih jednadžbi koje opisuju elektrotehničke fenomene. Isto tako, primjenu Laplaceove transformacije zapazit ćemo u području signala i sustava, kao i u mnogim drugim znanstvenim područjima.

S matematičkog stanovišta, iako idejno jednostavna Laplaceva transformacija zahtjevan je alat koji traži veću količinu vremena i često mukotrpan račun koji nameće primjenu informatičkih tehnologija. Stoga, cilj ovog rada je pokazati na koji su način Laplaceove transformacije integrirane u programski sustav MATLAB i kako se koriste na realnim problemima iz struke.

Prvi dio rada upoznat će nas s povijesnim razvojem MATLAB – a, primjenom nekad i danas kao i sa mogućnostima koje nam se danas pružaju. Detaljnije će biti objašnjeno kako je iz interaktivnog matričnog kalkulatora nastao softver koji danas koristi gotovo svaki inženjer. Također, upoznat ćemo se s načinima definiranja varijabli, njihovim uvođenjem u softver kao i sa mogućnostima rješavanja različitih tipova jednadžbi. Upoznat ćemo se i s implementacijom različitih operacija vezanih uz funkcije, kao što je primjerice njihovo deriviranje.

Drugi dio rada obrađuje Laplaceove transformacije, objašnjava njihov povijesni razvoj te današnju primjenu. Objasnit će se svojstva Laplaceove transformacije, a na dva primjera bit će prikazana primjena Laplaceove transformacije i njenih svojstava.

U zadnjem dijelu rada prikazat će se implementacija Laplaceovih transformacija u programskom sustavu MATLAB, pri čemu će se naglasak staviti na naredbe za Laplaceovu transformaciju i inverznu Laplaceovu transformaciju. U ovom će poglavlju biti riješeno nekoliko primjera povezanih s elektrotehnikom kao što su izračuni struje u RC, RL i RLC strujnom krugu.

2. Povijesni razvoj MATLAB-a

Prvi MATLAB nije bio programski jezik. Bio je u pitanju jednostavni interaktivni matrični kalkulator koji nije imao programe, alate i grafičke prikaze. Praćenjem povijesnog razvoja MATLAB-a, primjećuje se da dodaci koji se danas nude za transformiranje funkcija (Laplaceova transformacija, Fourierova transformacija), analize podataka u statistici, 'Destkop i Live Editor' značajke tada nisu postojali.

U ovom radu, povijesni razvoj MATLAB-a bit će opisan kroz tri poglavlja, a to su:

1. Prva verzija MATLAB-a
2. Komercijalni MATLAB
3. Moderni MATLAB

2.1. Prva verzija MATLAB-a

Između 1965. i 1970. godine, objavljen je niz istraživačkih radova koje su napisali J. H. Wilkinson[1]i njegovih 18 kolega. Ti radovi su prezentirali algoritme koji su služili za rješavanje matričnih linearnih jednadžbi. Navedeni istraživački radovi su zapravo poslužili kako bi se dobila matematička osnova za prvu, povijesnu verziju MATLAB-a.

U 70-im i ranim 80-im godinama prošlog stoljeća, Cleve Moler[2] je na Sveučilištu u Novom Meksiku predavao linearnu algebru i numeričku analizu te je htio svojim studentima omogućiti pristup LINPACK-u[3] i EISPACK-u[4] bez potrebe za pisanjem Fortran[5] programa. Cleve Moler je kroz neku vrstu hobija napravio prvi MATLAB u Fortranu s matricom kao jedinim tipom podataka. Zapravo je želja Clevea Molera bila napraviti nešto što će njegovi studenti moći koristiti.

[1]James Hardy Wilkinson (Engleska, 27. rujna 1919 – Engleska, 5. listopad 1986.) bio je važna osoba u polju numeričke analize. Radeći u tom području, otkrio je mnoge važne algoritme.

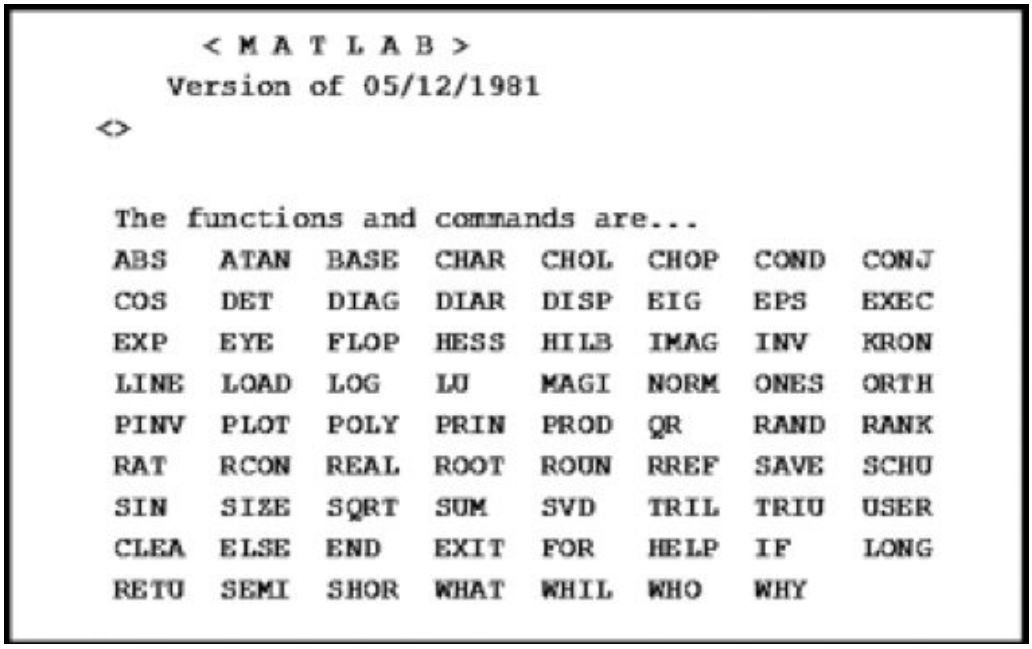
[2]Cleve Moler (Salt Lake City, 17. kolovoza 1939.) je matematičar i jedan od osnivača kompanije MathWorks. Moler je na Sveučilištu u Michiganu, Stanfordu i Novom Meksiku bio profesor matematike i računalnih znanosti skoro 20 godina. MathWorksu se 1989. pridružio za stalno.

[3]LINPACK je softverska biblioteka za izvođenje numeričke linearne algebre. Napisana je u Fortranu od strane Jacka Dongarra, Jima Buncha, Clevea Molera i Gilberta Stewarta.

[4]EISPACK je softverska biblioteka koja se koristi za numeričko računanje svojstvenih vrijednosti i svojstvenih vektora matrice, napisana u Fortranu.

[5]Fortran je programski jezik koji se razvio 50-ih godina prošlog stoljeća. Njegova upotreba je uglavnom u znanstvene svrhe.

Za pomoć u razvoju prvog MATLAB-a, Cleve Moler koristio je knjigu Niklausa Wirtha[6] pod nazivom 'Algoritmi + Struktura podataka = Program' te je pomoću nje naučio kako analizirati programske jezike.



Slika 2.1. Prikaz prvih funkcija i naredbi u MATLAB-u, Izvor: [1]

Slika 2.1 prikazuje funkcije i naredbe koje su bile dostupne u prvoj verziji MATLAB-a koja je svjetlo dana ugledala u prosincu 1981. godine. Tada bih, ukoliko se željelo dodati neku novu funkciju, bilo potrebno napisati Fortran potprogram, dodati ime funkcije u priloženu tablicu te ponovo pokrenuti MATLAB. Tek bi se nakon ponovnog pokretanja prikazala dodana funkcija. Slika 2.1 pokazuje da su u prvoj verziji MATLAB-a bile dostupne razne trigonometrijske funkcije, računanje korijena broja, sumiranje, apsolutne vrijednosti itd.

2.2. Komercijalni MATLAB

Cleve Moler proveo je akademsku godinu 1979./1980. na Stanfordu gdje je predavao numeričku analizu i u sklopu tog kolegija studentima je predstavio prvu, povijesnu verziju matričnog kalkulatora. Neki studenti pohađali su kolegije poput signala i sustava o kojima Cleve Moler nije znao ništa. Matrice su bile matematički oslonac u tim kolegijima te je ta verzija MATLAB-a koju je Cleve Moler napravio iz hobija, postala iznimno korisna tim studentima.

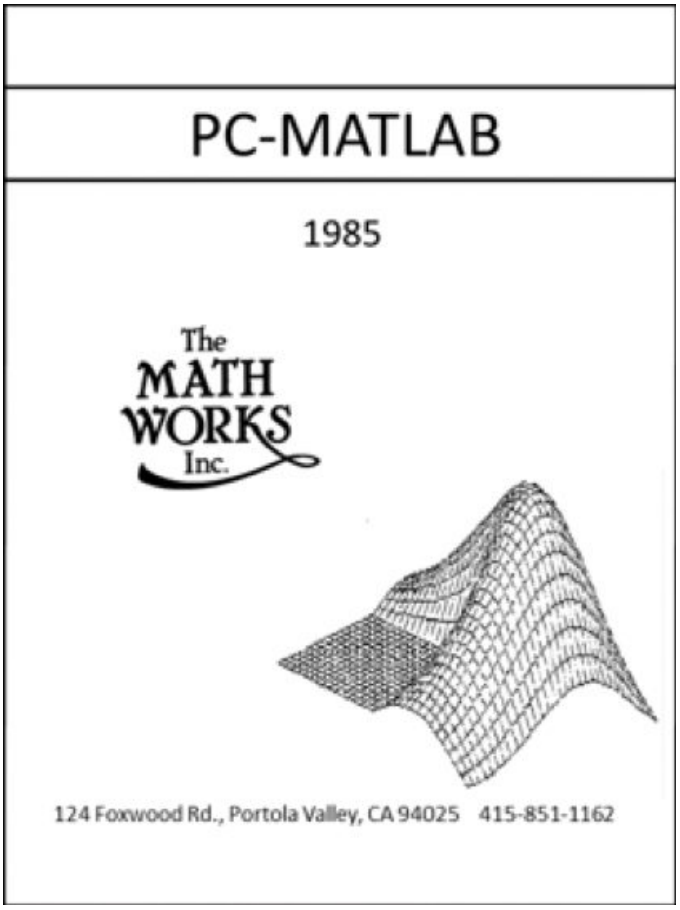
1983. godine, Jack Little predložio je komercijalni proizvod koji bi se bazirao na MATLABu.

[6]Niklaus Emil Wirth (Švicarska, 15. veljače 1934.) je švicarski znanstvenik i dobitnike raznih nagrada u području računalstva. Zaslužan je za razvoj programskog jezika PASCAL.

[6]Jack Little (Virginia, 1. veljače 1928.) je predsjednik i suosnivač MathWorksa. Bio je koautor i arhitekt ranijih verzija MATLAB-a. 1978. godine diplomirao je elektrotehniku i računalstvo na MIT-u.

Dvije godine prije bio je predstavljen 'The IBM® PC'[7]. To računalo bilo je jedva dovoljno jako za pokretanje programa kao što je MATLAB, ali Jack Little je predvidio razvoj računala te odlučio napustiti svoj posao, kupiti računalo Compaq® PC, odseliti se u brda iza Stanforda te uz poticaj Clevea Molera, napisati novu, proširenu (komercijalnu) verziju MATLAB-a u programskom jeziku C. Prijatelj Clevea Molera i jedan od osnivača kompanije MathWorks, Steve Bangert[8] također je, tijekom svojeg slobodnog vremena, sudjelovao u stvaranju novog MATLAB-a.

U prosincu 1984. godine, pojavila se verzija MATLAB-a za osobna računala. Također, godinu dana nakon pojave 'PC - MATLAB' pojavila se i verzija 'Pro - MATLAB'. Razlika između te dvije verzije koje su se pojavile unutar godinu dana bila je u nekoliko dodanih funkcija i mogućnosti boljih grafičkih prikaza. Jack Little i Steve Bangert su u odnosu na povijesnu verziju MATLAB-a napravili jako puno bitnih poboljšanja i modifikacija. Najvažnije su bile funkcije iz područja analize signala, detaljnijeg izračuna matrica (inverz matrice, kvadratni korijen matrice, karakteristični polinom), različiti alati i grafički prikazi.



Slika 2.2. Komercijalni MATLAB, Izvor: [1]

[7]IBM Personal Computer, kratica IBM PC, bio je naziv modela prvog osobnog računala američke tvrtke IBM iz 1981. godine.

[8]Steve Bangert jedan je od osnivača kompanije MathWorks kojoj se službeno za stalno pridružio tek 1985. godine kao treći zaposlenik.

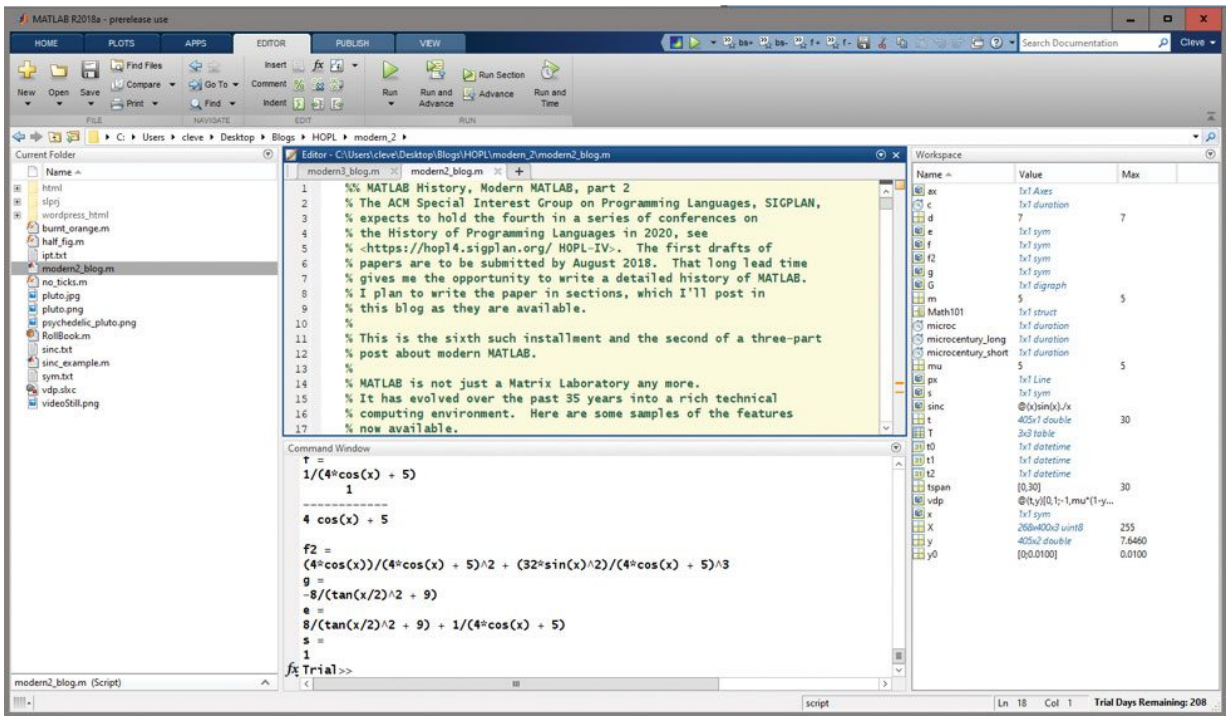
2.3. Moderni MATLAB

Unatoč tome što su korijeni MATLAB-a u matričnom računu, s obzirom na razvoj znanosti te potrebe inženjera, MATLAB se nastavio širiti i u druga područja primijenjene matematike.

Jedna od važnih opcija koju moderni MATLAB nudi je rješavanje običnih diferencijalnih jednadžbi, a i danas iznimno važan dodatak Simulink®[9]se temelji na rješavanju diferencijalnih jednadžbi. Simulink je integriran s MATLAB-om te ima mogućnost ubaciti MATLAB-ove algoritme u modele te rezultate simulacije prenijeti u MATLAB za daljnje analize.

Dugi niz godina MATLAB je imao samo jednu vrstu numeričkih podataka, što znači da nije mogao prikazivati rješenja u drukčijem obliku u odnosu na tu jednu vrstu. Sa povećanjem broja korisnika, programeri MATLAB-a su poslušali potrebu inženjera te su ubačeni razni načini za prezentaciju podataka što znači da se više nije sve svodilo na prikaz rješenja kroz jednu vrstu numeričkih podataka. Danas se koriste: realni brojevi, cijeli brojevi, rijetke matrice[10], guste matrice, ćelije, strukture, objekti, ...

Kako bi MATLAB postao pristupačniji korisnicima, u modernom MATLAB-u su dodani odvojeni prozori za grafičke prikaze, uređivanje i ostale alate. Ti dodaci su omogućili da se MATLAB puno lakše koristi. To je posebno značilo korisnicima bez iskustva i znanja u području programiranja. Dvije najvažnije značajke koje su unaprijedile MATLAB bile su 'Destkop' i 'Live Editor.'



Slika 2.3. MATLAB Destkop danas, Izvor: [1]

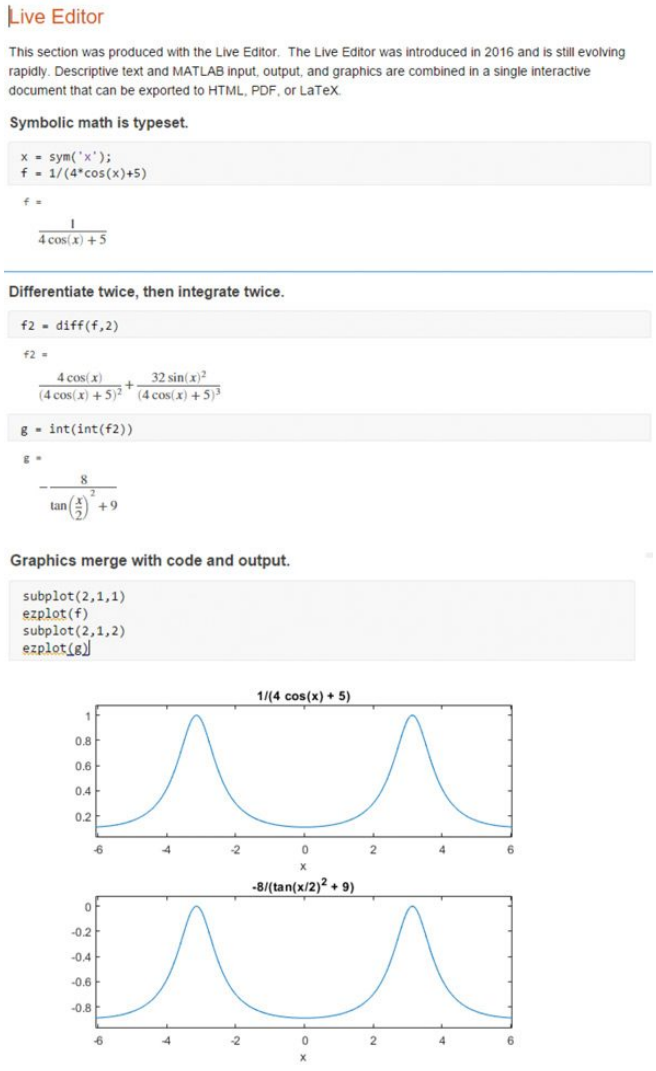
[9]Simulink je softver u kojem se uz pomoć blok dijagrama slažu modeli koji se simuliraju prije prebacivanja u hardverski model bez pisanja koda.

[10]Rijetke matrice su matrice u kojima je većina elemenata 0

MATLAB-ova značajka 'Destkop' dodana je u MATLAB 2000. godine. Slika 2.3 prikazuje nam kako to izgleda danas. Primijetimo četiri prozora, svaki od njih može biti zatvoren ili pomaknut u samostalni prozor. Na lijevoj strani nam je prikazana mapa u kojoj je spremljena datoteka u kojoj radimo. Desna strana nam prikazuje radni prostor, a na vrhu vidimo prostor za upisivanje željenog koda. Na dnu vidimo prozor gdje će se pojaviti rezultat izvršenja napisanog koda. Također, ukoliko želimo, u 'Destkop' značajku možemo još dodati povijest naredbi.

Jedna od značajki koja je također iznimno olakšala i unaprijedila korištenje MATLAB-a je 'Live Editor' koje se u MATLABu pojavio 2016. godine i svakodnevno se nastavlja razvijati za lakše korištenje. MATLAB ulaz/izlaz, grafički prikaz i opisni tekst sadržani su u posebnom dokumentu koji može biti izvezen u HTML, PDF ili LaTex.

Jedna od opcija koju 'Live editor' pruža je stvaranje skripte koja trenutačno pokazuje rezultate koda. Slika 2.5 prikazuje situaciju kada se stvara skripta koja se može dodatno uređivati. Pod uređivanje se smatra dodavanje jednadžbi, fotografija, poveznica. Bitna stvar je da se ta skripta putem MATLAB-a može podijeliti kao interaktivni dokument.

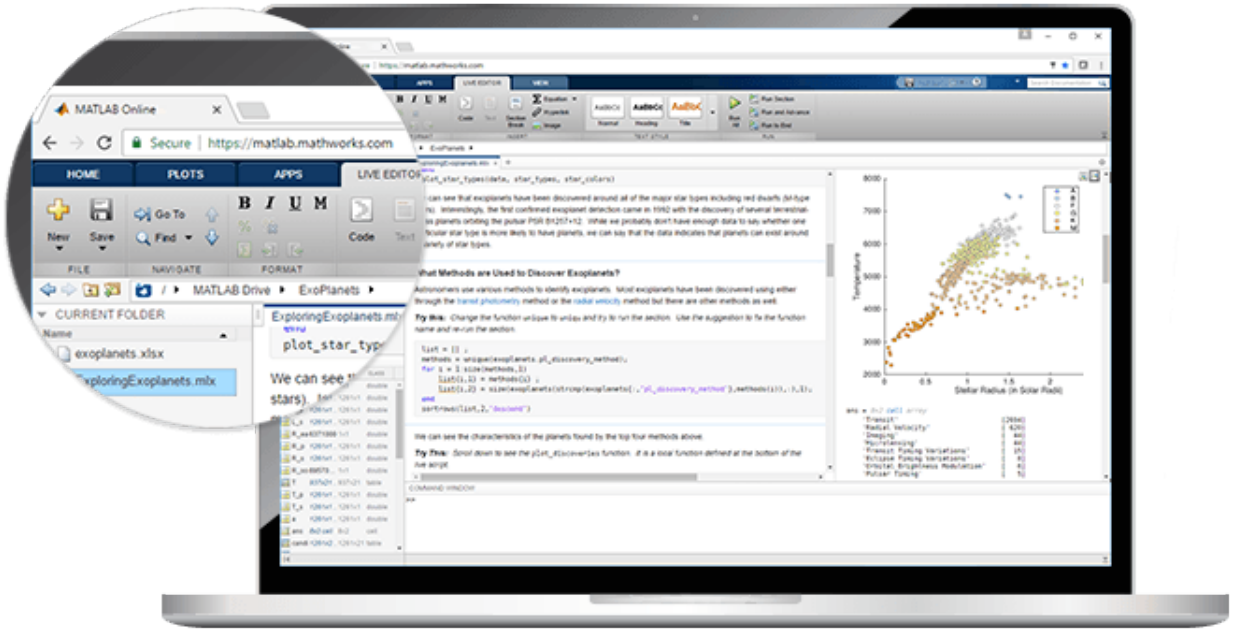


Slika 2.4. Live editor, Izvor: [1]

Pod moderni MATLAB nam također spada i online inačica MATLAB-a. Ona nam omogućava pristup MATLAB-u i Simulinku preko bilo kojeg internet pretraživača ukoliko imamo pristup internetu. Potrebno je samo prijaviti se na web stranicu, a ne posebno skidati instalacijsku datoteku i potom instalirati MATLAB na vlastito računalo. Ova online inačica iznimno je pogodna za poučavanje, učenje te ima praktičan i lagan pristup.

MATLAB Online daje mogućnost dijeljenja skripte i ostalih MATLAB datoteka sa ostalim korisnicima. Također, online inačica se automatski ažurira na posljednju verziju što svim korisnicima omogućava korištenje najmodernijih i najnovijih alata. Objava skripti na internetu moguća je u PDF ili HTML obliku te se URL poveznica može podijeliti sa svima.

Treba dodati kako potpuna integracija s MATLAB diskom svakom korisniku dodjeljuje 5GB prostora za pohranjivanje datoteka. Također, potpuna integracija omogućit će korisniku pristup datotekama s bilo koje lokacije. Za sinkronizaciju datoteka između računala i online inačice MATLAB-a koristi se MATLAB Drive Connector koji nam eliminira potrebu za podizanjem datoteke na sustav ili skidanjem iste.



Slika 2.5. Online MATLAB, Izvor: [13]

Alati koje nam online MATLAB pruža su raznoliki, a neki od njih su: audio alati, komunikacijski alati, alati za analizu i pohranjivanje podataka (SQLite baze podataka su podržane), alati za uređivanje fotografija, MATLAB koder, alati za paralelno računanje u MATLAB-u, razne Simulink opcije (treba napomenuti da nije moguće koristiti sve opcije Simunlika kao u verziji MATLAB-a za osobna računala).

2.3.1. Alati u modernom MATLABu

Većina mogućnosti koje moderni MATLAB pruža potječe iz alata koji su dostupni u posebnim softverima. Izdavanjem verzije MATLAB-a 2018a, postoje 63 alata. Kod svakog od alata, u zagradi će biti naveden broj različitih mogućnosti koje taj alat može pružiti.

Alati su:

1. Implementacija aplikacije (3)
2. Generiranje koda (7)
3. Računalna biologija (2)
4. Računalne financije (8)
5. Upravljački sustavi (8)
6. Pristup bazi podataka i slanje izvješća (2)
7. Obrada slike (6)
8. Matematika, statistika i optimizacija (9)
9. Paralelno računanje (2)
10. Obrada signala i bežična komunikacija (11)
11. Testiranje i mjerenje (5)

2.3.2. Biblioteke u MATLAB-u

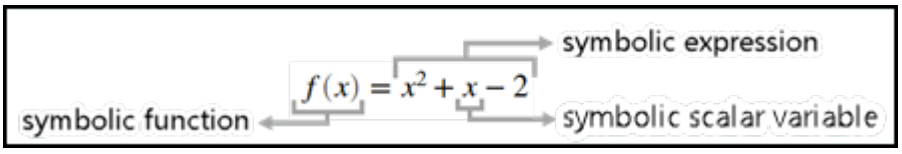
Biblioteke u programskim jezicima su skupina prije napisanih kodova koje korisnici koriste prilikom programiranja. Jedan od primjera gdje se u MATLAB-u koriste biblioteke je slaganje Simulink modela. Koriste se razni blokovi pomoću kojih kreiramo podsistem, funkcije koje nam omogućavaju imenovanje bloka, upravljanje blokom i daju informacije o nekom bloku. Također, koriste se alati koji nam pomažu u pronalasku i dodavanju bloka. Treba još dodati prikazivanje, skrivanje, mijenjanje veličine i pozicije Simulinkova prozora. Biblioteke MATLAB-a koristimo u slučaju kada želimo pozvati funkcije programskih jezika u MATLAB-u.

MATLAB nam pruža mogućnost kreiranja vlastite biblioteke koju ćemo koristiti u željenu svrhu. Kreiranjem vlastite biblioteke bit će nam lakše ponovo koristiti komponente u modelima.

Za usporedbu, alati su tipična grupa MATLAB-ovih funkcija dok su biblioteke skupina Simulink blokova. Naravno, mnogi MATLAB-ovi alati također nude Simulink biblioteke. Primjeri su alati za kontrolu sustava, alati za identifikaciju sustava, alati za kontrolu instrumenata.

2.3.3. Simbolički izračun u MATLABu

Simbolički matematički alati omogućavaju nam izvođenje simboličkih izračuna iz MATLAB-ova naredbena retka definiranjem posebnog tipa podataka – simboličkih objekata. Funkcije koje se koriste u simboličkom računanju su slične MATLAB-ovoj sintaksi i omogućavaju nam integriranje, diferenciranje, pojednostavljenje, rješavanje sustava jednadžbi te ostale matematičke operacije. Simboličkim izračunima, povećat ćemo numeričke mogućnosti koje nam MATLAB pruža. Kod ovakve vrste računanja, možemo prebaciti simbolički rezultat u numerički te taj isti rezultat koristiti van simboličko matematičkog alata.



Slika 2.6. Objašnjenje simboličke funkcije, Izvor: [12]

Slika 2.6 prikazuje nam definiranje simboličke funkcije $f(x)$. Funkciju $f(x)$ u MATLABu definiramo naredbom syms $f(x)$. Primjerice $x^2 + x - 2$ je simbolički izraz, a varijabla x je simbolički skalar. U nastavku prikazujemo neke od opcija koje nam simbolički izračun nudi.

Simbolički izračun u MATLAB-u bit će prikazan kroz umnožak dvije simboličke matrice. Na naredne dvije slike bit će vidljivo kako se definiraju simboličke matrice. Primijetit ćemo da ne postoji razlika u definiranju u odnosu na numeričku (slova su zamijenila brojeve). Umnožak simboličkih matrica radi se isto kao i kod numeričkih. U kodu na slici 2.7 su detaljno opisani svi potrebni elementi za njegovo uspješno izvršavanje.

```
diary('rezultat.txt');

diary on ;

                                %definiranje varijabli
                                      syms a b c;

                               %definiranje prve matrice
                                A = [a b c; b a c; c a b]

                               %definiranje druge matrice
                                B = [a b a; c a b; b a c]

                                %množenje dvije matrice
                                      C = A * B
diary off ;
```

Slika 2.7. Kod za simbolički izračun umnoška dvije matrice

U kodu primjećujemo funkciju naziva 'Diary'. Ona se koristi kada rezultat izvršenja koda

želimo izvesti iz MATLAB-a. Ona nam kopira sve (osim grafičkih prikaza) što se nalazi u prozoru 'Command Window' u kojem se prikazuju rezultati izvršenja koda.

Na idućoj slici prikazat ćemo rezultat izvršavanja koda sa slike 2.7.

```
A =                B =

[ a, b, c]         [ a, b, a]
[ b, a, c]         [ c, a, b]
[ c, a, b]         [ b, a, c]


C = A * B

[      a^2 + 2*b*c,      2*a*b + a*c, a^2 + b^2 + c^2]
[ a*b + a*c + b*c, a^2 + c*a + b^2,      c^2 + 2*a*b]
[      b^2 + 2*a*c, a^2 + b*a + b*c, a*b + a*c + b*c]
```

Slika 2.8. Prikaz rezultata za simbolički izračun umnoška dvije matrice

Primijetimo da je MATLAB prvo prikazao kako su zadane simboličke matrice, a tek je potom ispisao rješenje umnoška dvije simboličke matrice.

Slika 2.8 prikazuje nam korištenje funkcije 'Diary' u tri dijela. Prvi je u definiciji naziva datoteke u koju će se rezultati izvršenja koda spremiti. Drugo korištenje je naredba 'Diary on' koja MATLAB-u daje znak da se sve što se pojavi u 'Command Window', a vezano je za kod koji se nalazi između 'Diary on' i 'Diary off' spremi u datoteku koja je definirana u redu prije. Posljednje korištenje je 'Diary off' koji MATLAB-u daje znak da više ne sprema sadržaj iz 'Command Window' u datoteku. Također, koristi se i naredba syms a b c; koja nam služi za definiranje simboličkih varijabli koje koristimo u matricama.

Jedna od opcija koju nam kod rada sa matricama MATLAB pruža je i izračunavanje inverza matrice. U sljedećem primjeru, usporedit ćemo inverz simboličke i numeričke matrice, a na kraju ćemo zaključiti za koju od njih je MATLAB brže i jednostavnije izvršio napisani kod. Kod traženja inverza simboličke matrice (B) iskoristit ćemo matricu iz prethodnog primjera, a za numeričku matericu (A) uzet ćemo jednu proizvoljnu matricu istog formata. Matrice glase:

$$A = \begin{bmatrix} 1 & 0 & 2 \\ -1 & 5 & 0 \\ 0 & 3 & -9 \end{bmatrix}, \qquad B = \begin{bmatrix} a & b & c \\ b & a & c \\ c & a & b \end{bmatrix}.$$

Na sljedeće četiri slike bit će prikazan:

1. Kod koji će biti potreban za inverz simboličke i numeričke matrice.
2. Rezultat izvršenja koda.

3. Trajanje izračuna.

Na sljedećoj slici možemo vidjeti kod za simboličko izračunavanje inverza matrice.

```
diary('rezultat3.txt');

diary on ;
                              %definiranje varijabli
                                   syms a b c;

                              %definiranje matrice
                            B = [a b c; b a c; c a b]

                         %naredba za inverznu transformaciju matrice
                                    B = inv(B)

                       %naredba koja mjeri vrijeme trajanja tranformacije
                                   timeit(@()inv(B))


diary off ;
```

Slika 2.9. Kod za inverz simboličke matrice

Rezultat izvršavanja koda s prethodne slike prikazan je na idućoj slici.

```
B =

[ a, b, c]
[ b, a, c]
[ c, a, b]


B =

[ a/(a^2 + c*a - b^2 - c*b), 1.1
(- b^2 + a*c)/(a^2*b - a^2*c + a*b*c - a*c^2 - b^3 + b*c^2), 1.2
-c/(b^2 + a*b - c^2 - a*c)] 1.3

[ -(b + c)/(a^2 + c*a - b^2 - c*b), 2.1
(- c^2 + a*b)/(a^2*b - a^2*c + a*b*c - a*c^2 - b^3 + b*c^2), 2.2
 -c/(b^2 + a*b - c^2 - a*c)] 2.3

[a/(a^2 + c*a - b^2 - c*b), 3.1
 (- a^2 + b*c)/(a^2*b - a^2*c + a*b*c - a*c^2 - b^3 + b*c^2), 3.2
 (a + b)/(b^2 + a*b - c^2 - a*c)] 3.3


ans =

    0.0330
```

Slika 2.10. Rješenja dobivena izvršenjem koda za inverz simboličke matrice.

Možemo primijetiti kako je simbolički izračun dan prilično nečitko, međutim za situacije u kojem nam treba simbolički izraz ovakav pristup izračuna je daleko brži od klasičnog računa.

Treba naglasiti da je bilo bitno definirati simboličke varijable, inverz smo računali putem naredbe *inv*, a naredba *diary* ima istu ulogu kao i prije.

Na sljedeće dvije slike prikazati ćemo računanje inverza numeričke matrice.

```
diary('rezultat4.txt');

diary on ;

                        %definiranje matrice
                    A = [1 0 2; -1 5 0; 0 3 -9]

                %naredba za inverznu transformaciju matrice
                            A = inv(A)

            %naredba koja mjeri vrijeme trajanja tranformacije
                           timeit(@()inv(A))


diary off ;
```

Slika 2.11. Kod za inverz numeričke matrice

```
A =

     1     0     2
    -1     5     0
     0     3    -9


A =

    0.8824   -0.1176    0.1961
    0.1765    0.1765    0.0392
    0.0588    0.0588   -0.0980


ans =

   4.0949e-06
```

Slika 2.12. Rješenja dobivena izvršenjem koda za inverz numeričke matrice.

Korištenjem naredbe 'timeit' došli smo do vremena koje je potrebno za inverznu transformaciju numeričke i simboličke matrice. Obzirom kako je MATLAB za svako novo pokretanje koda izbacio drugačije rješenje, uzeli smo prosječnu vrijednost vremena.

Inverzna transformacija numeričke matrice trajala je $4.0949 \cdot 10^{-6}$, a inverzna transformacija simboličke matrice trajala je 0.0330. Ukoliko stavimo vremena trajanja inverzna transformacije u

omjer, tada vidimo da nam je vrijeme inverzne transformacije simboličke matrice 8059 puta veće u odnosu na vrijeme inverzne transformacije numeričke matrice.

2.3.4. Primjena MATLAB-a u struci

MATLAB® i Simulink® inženjerima omogućuju rješavanje problema koji se pojavljuju u modernoj industriji. Za područje automatizacije, inženjeri koriste 'Model – Based' dizajn u MATLAB-u i Simulinku kako bi mogli:

1. Dizajnirati i testirati automatiku strojeva
2. Pokretati testiranja na opremi

Primjena MATLAB-a može se pronaći i u analizi podataka. Većinom se analiza radi pomoću ostalih softvera i alata, ali MATLAB se koristi za stvaranje sustava gdje će se pohraniti velika količina podataka. MATLAB se u tome koristi jer je u MATLAB-u dosta jednostavno stvoriti sustav za pohranu velike količine podataka. Također, jedan od razloga je i prikladnost budući da se radi sa već poznatim sustavima kao što su SQL i NoSQL.

Također, u MATLAB-u se često izrađuju algoritmi za kontrolu brzine motora. Neki od algoritama mogu poslužiti za zaštitu i kontrolu sustava.

Pametna kuća jedan je od mnoštva primjera primjene MATLAB-a u elektrotehnici. Pomoću softvera, izrađuju se sustavi koji upravljaju kućom. MATLAB nije jedini softver koji se koristi u te svrhe, ali je jedan od njih.

Slika 2.13. Pametna kuća, Izvor: [5]

3. Osnove korištenja MATLAB-a

3.1. Varijable

Kako bismo u MATLAB-u mogli definirati neku funkciju $f(t)$ ili izvršiti neku matematičku operaciju koju smo zamisliti, potrebno je prije svega definirati varijable. Kao i svaki programski jezik, MATLAB također ima više vrsti varijabli. Kod MATLAB-a su u pitanju:

1. Interne varijable
2. Eksterne varijable

Sve varijable koje se definiraju se mogu obrisati. Uklanjanje varijabli iz radnog prostora MATLAB-a vrši se pomoću naredbe 'clear'.

3.1.1. Interne varijable

Interne varijable su varijable koje se već nalaze u MATLAB-u te imaju pridruženu neku vrijednost. Tim varijablama mi možemo promijeniti vrijednost ukoliko imenu varijable pridružimo neku drugu vrijednost. Interne varijable se koriste za provođenje raznih matematičkih operacija te ih nije moguće izbrisati. Ukoliko i promijenimo vrijednost varijable, naredbom 'clear' obrisat ćemo tu novo dodanu vrijednost te joj pridružiti staru, u MATLAB-u već upisanu vrijednost.

Neki od primjera internih varijabli su:

1. π = 3.14159265358979,
2. i i j kao imaginarne jedinice kompleksnog broja,
3. $NaN = 0/0$ neodređena vrijednost.

Naravno, to nisu sve interne varijable. Još postoje varijable u kojima je definirana najmanja i najveća vrijednost pozitivnog realnog broja. Također, postoji i varijabla u kojoj je definirana vrijednost dijeljenja sa 0 i varijabla za broj floating point operacija.

Varijabla flops koja je definirana za broj floating point operacija služi za mjerenje efikasnosti nekog algoritma. Na početku algoritma ju pomoću naredbe *flops(0)* može postaviti na 0, a na kraju ćemo dobiti neku vrijednost koja je jednaka broju floating point operacija.

Sadržaj internih varijabli MATLAB-a može se koristiti u funkcijama i matematičkim operacijama.

3.1.2. Eksterne varijable

Sam pojam eksternih varijabli govori nam kako su u pitanju neke vanjske varijable koje moramo definirati i dodati im vrijednost. Eksterna varijable se mogu sastojati od jednog pa sve do devetnaest alfanumeričkih znakova. Naravno, velika i mala slova nisu jednaka pa se dvije varijable smatraju različitima ukoliko nije sve identično. Eksterne varijable su one koje korisnik MATLAB-a definira ili će njihova definicija proizaći iz rezultata neke funkcije ili matematičke operacije.

3.1.3. Definiranje varijabli

Varijablu možemo istovremeno definirati i dodijeliti joj vrijednost. Primjer definiranja varijable i dodjeljivanja vrijednosti je:

```
C = 3;
```

Ovime ćemo definirati eksternu varijablu C i dodijelit ćemo joj vrijednost 3. Točka zarez (;) koji se nalaze nakon broja nam kazuju da se vrijednost varijable C neće prikazati u 'Command Windowu'.

Definiciju matrica prikazali smo u poglavlju simboličkog izračuna gdje smo definirali simboličke, ali i numeričke matrice. Varijable koje smo koristili za definiciju simboličke matrice morali smo definirati pomoću naredbe `syms a b c;`. Za definiciju numeričke matrice, nije bilo potrebno definirati neku dodatnu varijablu nego jedino upisati brojeve. To je ilustrirano na iduće dvije slike.

```
diary('DefinicijaMatricaUmnozak.txt');
diary on;
                          %definiranje varijabli
                               syms a b c;
                          %definiranje matrice
                      B = [a b c; a b c; a b c];
                      C = [1 2 3; 1 2 3; 1 2 3];
                      E = B * C
diary off ;
```

Slika 3.1. Kod za definiciju numeričke i simboličke matrice.

```
B =                          C =

[ a, b, c]                    [1  2   3]
[ a, b, c]                    [1  2   3]
[ a, b, c]                    [1  2   3]
```

Slika 3.2. Matrice dobivene izvršenjem koda sa slike 3.1.

Bilo kakva matematička operacija koja bi uključivala ove dvije matrice stvorila bi novu varijablu koja bi zbog tipova ove dvije matrice automatski bila definirana te bi joj bila dodijeljena

vrijednost, primjerice ako bi željeli pomnožiti matrice B i C te rezultat spremiti u matricu (varijablu) E unijeli bi kod:

```
E=B*C.
```

```
E =

[ a + b + c, 2*a + 2*b + 2*c, 3*a + 3*b + 3*c]
[ a + b + c, 2*a + 2*b + 2*c, 3*a + 3*b + 3*c]
[ a + b + c, 2*a + 2*b + 2*c, 3*a + 3*b + 3*c]
```

Slika 3.3. Umnožak dvije matrice dobivene izvršenjem koda sa slike 3.1.

Matrica E će umnoškom matrica B i C automatski biti definirana kao matrica što je navedeno u prijašnjem odlomku.

3.2. Funkcije

Funkcije su izrazi koji povezuju jednu nezavisnu varijablu sa drugom, također nezavisnom varijablom. Što se tiče samih funkcija, u MATLAB-u se definiraju ovisno o njihovom broju izlaza. Prihvatljivo ime funkcije započinje sa nekim slovom abecede, a može sadržavati slova, brojeve ili znakove poput _.

Sintaksa pisanja funkcije glasi:

```
function [y1,...,yN] = myfun(x1,...,xM)

        tijelo funkcije

              end.
```

Osim navedenog u prethodnom odlomku, za funkcije je važno znati da se mogu pohranjivati na dva načina. Prvi od načina je da se funkcija nalazi u istoj skripti kao i sve ostale naredbe što će biti vidljivo u primjerima u nastavku. Tada je važno znati da naziv skripte ne može biti jednak kao i naziv funkcije unutar datoteke. Drugi od načina je taj da naziv skripte bude jednak nazivu funkcije, a ukoliko bi unutar skripte bilo više funkcija, tada bi naziv skripte bio jedan nazivu prve funkcije.

Iz sintakse pisanja funkcije vidimo kako ona započinje. Treba reći da je to jedini način kako zapisivanje funkcije u MATLAB – u može započeti. Funkcija završava sa naredbom 'end' nakon koje se više ništa ne upisuje jer kao što je navedeno u prethodnom odlomku, u slučaju da su funkcije i pozvane naredbe u istoj datoteci, tada funkcija ide posljednja.

3.2.1. Funkcije sa jednim izlazom

Kod funkcije sa jednom izlazom, napravit ćemo primjer gdje će na ulazu biti niz brojeva te se računa srednja vrijednost tog niza, a kao izlaz ćemo dobiti cijeli broj. MATLAB datoteka definirana je kao 'DefinicijaFunkcijeProsjek.m'. Funkciju koja računa prosječnu vrijednost niza se ne definira u posebnoj datoteci nego unutar iste gdje je definiran niz. Funkcija se definira nakon što se zatvori *.txt datoteku gdje se spremaju rješenja datoteke.

```
diary('DefinicijaFunkcijeProsjek.txt');
diary on;
z = 1:5;
prosjek = average(z) %prosjek je varijabla
diary off;

            function pr = average(x) %pr je naziv funkcije
            pr = sum(x(:))/numel(x);
            end
```

Slika 3.4. Kod za izračun prosjeka niza brojeva.

```
DefinicijaFunkcijeProsjek

prosjek =

     3
```

Slika 3.5. Prikaz dobivenog rezultata izvršenja koda.

Za niz su uzeti brojevi od 1 do 5. Pomoću MATLAB-a i funkcije izračunata je prosječna vrijednost niza. Kako je navedeno prije kod objašnjavanja zapisivanja funkcija u MATLAB – u, primjećuje se da je funkcija bila posljednja te da nakon naredbe 'end' više ništa nije upisivano jer je funkcija definirana u istoj datoteci kao i naredbe koje su pozivane.

3.2.2. Funkcije s više izlaza

Kod funkcije s više izlaza napravit ćemo primjer gdje će nam MATLAB izračunati vrijednost opsega i površine kvadrata. Ulazna varijabla bit će dužina stranice kvadrata, a dužina stranice mijenjat će se po nizu od 0 do 3. MATLAB datoteka nazvana je 'DefinicijaFunkcijeViseIzlaza.m'. Malo više pažnje potrebno je obratiti na pisanje formule za opseg i površinu kvadrata. Kada se u MATLAB-u provodi matematička operacija umnoška, tada se ispred znaka * treba staviti točku, pa umnožak zapravo izgleda: `x.*x`.

```
diary('DefinicijaFunkcijeViseIzlaza.txt');
diary on;
a = 0:3;
[O,P] = OP(a)
diary off;
```

```
function [O,P] = OP(x)
O = 2.*x;
P = x.*x;
end
```

Slika 3.6. Kod za izračun opsega i površine kvadrata.

```
O =
     0     2     4     6

P =
     0     1     4     9
```

Slika 3.7. Prikaz dobivenog rezultata izvršenja koda.

Ovdje je osim množenja dvije varijable, također važno pripaziti da se iza naredbe 'end' kod funkcije više ništa ne upisuje zato što će MATLAB izbacivati pogrešku i neće izvršiti kod.

3.2.3. Zadavanje simboličkih matematičkih funkcija

Da bi smo mogli definirati neku funkciju $f(x)$, potrebno je definirati varijable koje će biti korištene unutar te funkcije. U prethodnim primjerima to je rađeno pomoću naredbe 'syms.' Ta simbolička funkcija može poslužiti za razne primjere. Na narednim slikama bit će prikazano zadavanje neke simboličke funkcije $f(x)$, a potom izračun njezine vrijednosti u nekoj točki i prva derivacija te zadane funkcije.

Funkcija koju smo zadali za provođenje navedenih primjera je:

$$f(x) = \frac{x}{x^2 + 5}$$

Računanje vrijednosti funkcije kada je zadana vrijednost varijable

U primjeru izračunavanja vrijednosti simboličke funkcije $f(x)$ za neku vrijednost varijable, odabran je $x = 5$. Nastavak nam prikazuje kod koji je potreban za definiranje funkcije koja je odabrana, a potom i rješenje dobiveno uvrštavanjem vrijednosti varijable u definiranu funkciju.

```
diary('VrijednostFunkcijeUTC.txt');

diary on;

syms x; %definiranje varijable

f(x) = x/(x^2+5); %definiranje simbolicke funkcije

x = 5; %definiranje vrijednosti varijable

f(5) %vrijednost simb. funkcije kada je x zadan

diary off;
```

Slika 3.8. Kod potreban za definiranje simboličke funkcije $f(x)$ te izračunavanje vrijednosti simboličke funkcije kada je x zadan.

```
VrijednostFunkcijeUTC

ans =

1/6
```

Slika 3.9. Rezultat dobiven izvršenjem koda.

Prva derivacija zadane simboličke funkcije

Primijetimo da u prethodnom primjeru nije bila potrebna nikakva posebna naredba. U primjeru diferenciranja simboličke funkcije $f(x)$ bit će korištena naredba $diff$ pomoću koje dobijemo prvu derivaciju simboličke funkcije.

Kod koji se koristi za diferenciranje zadane simboličke funkcije će u odnosu na prethodni primjer imati razliku samo u toj naredbi za diferenciranje. Dio koda koji je zadužen za zadavanje simboličke funkcije bit će identičan kao u prethodnom primjeru.

Osim pronalaska prve derivacije, u ovom primjeru bit će izračunata i vrijednost prve derivacije kada nam je $x = 5$.

```
diary('DiferenciranjeSimbFunkcije.txt');

diary on;

syms x; %definiranje varijable

f(x) = x/(x^2+5); %definiranje simbolicke funkcije

df(x) = diff(f(x)) %diferenciranje simboličke funkcije

x = 5;

df(5) %izračunavanje vrijednosti prve derivacije u x = 5

diary off;
```

Slika 3.10. Kod potreban za definiranje simboličke funkcije $f(x)$ te izračunavanje prve derivacije i vrijednosti prve derivacije ukoliko je x = 5.

```
df(x) =

1/(x^2 + 5) - (2*x^2)/(x^2 + 5)^2


ans =

-1/45
```

Slika 3.11. Rezultat dobiven izvršenjem koda.

3.2.4. Crtanje funkcija

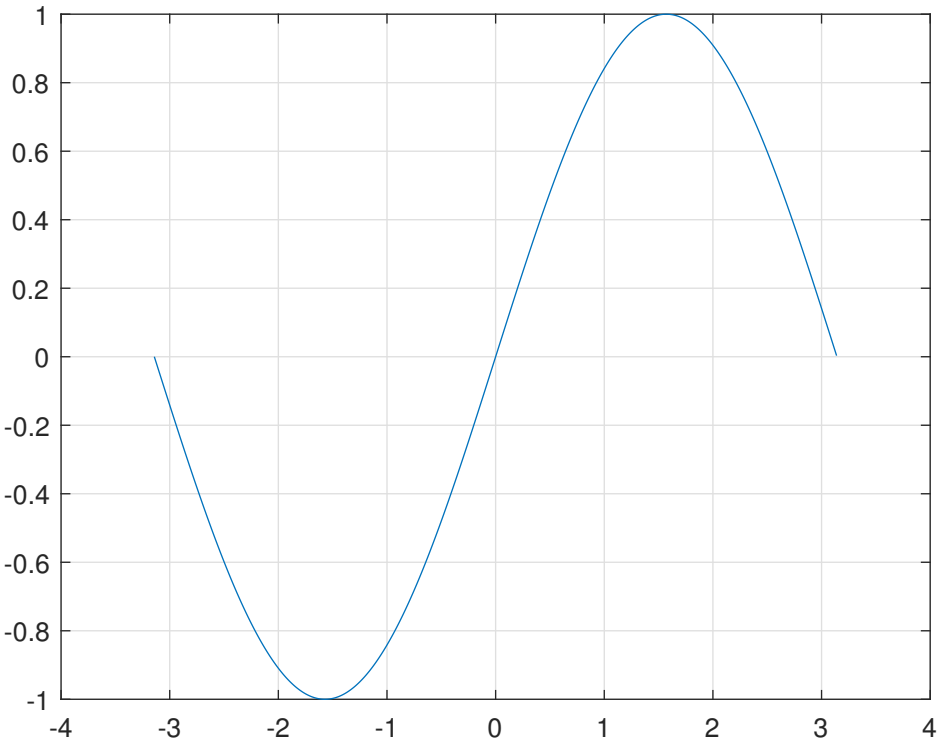
U poglavlju crtanja funkcija, odabrana su tri primjera. Prvi primjer bit će sinusna trigonometrijska funkcija, drugi primjer je u pitanju eksponencijalna funkcija, a za treći, posljednji primjer odabrana je funkcija $f(t) = t^2 \cos(t^3 + 5)$.

Po samom izrazu primjećuje se da će posljednja funkcija u odnosu na prethodne dvije biti kompleksnija za crtanje, ali MATLAB je softver koji s crtanjem funkcija nema problema.

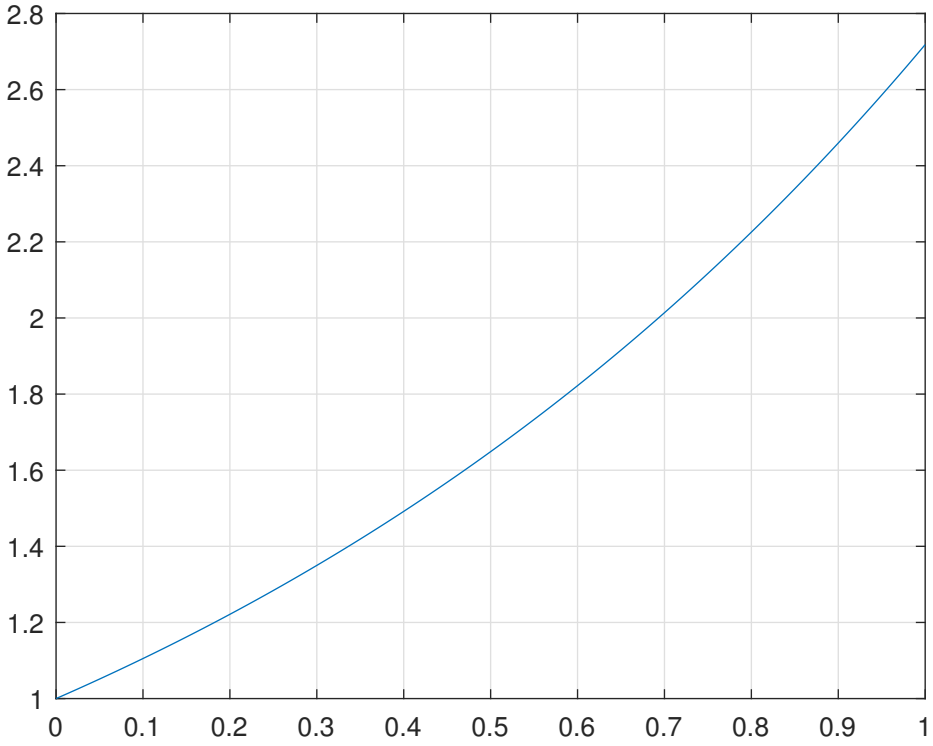
Za grafički prikaz neke željene funkcije u MATLAB – u, obično se koristi naredba $plot()$; koja prikazuje odabranu funkciju. Uz navedeni $plot()$;, u MATLAB – u se često koristi i naredba $fplot()$; koja služi za grafički prikaz nekih kompleksnijih rješenja. Što se tiče zapisa, naredba $plot(x, y)$; se zapisuje tako da se prvo definiraju vrijednosti na x – osi, a potom na y – osi. Kod $fplot(y)$; je potrebno samo navesti funkciju koja se želi grafički prikazati.

```
figure(); %otvaranje prozora
xlabel('t'); %definiranje x - osi
ylabel('f(t)'); %definiranje y - osi
t = -pi:0.01:pi; %definiranje vremenske osi
plot(t,sin(t))
grid on; %postavljanje resetke

figure(); %otvaranje prozora
xlabel('t'); %definiranje x - osi
ylabel('f(t)'); %definiranje y - osi
t = 0:0.01:1; %definiranje vremenske osi
plot(t,exp(t)) %crtanje grafa
grid on; %postavljanje resetke

syms t; %definiranje varijable
figure(); %otvaranje prozora
xlabel('t'); %definiranje x - osi
ylabel('f(t)'); %definiranje y - osi
t = 0:0.01:1; %definiranje vremenske osi
y = t.^2.*cos(t.^3+5); %definiranje funkcije
plot (t,y); %crtanje grafa
grid on; %postavljanje resetke
```

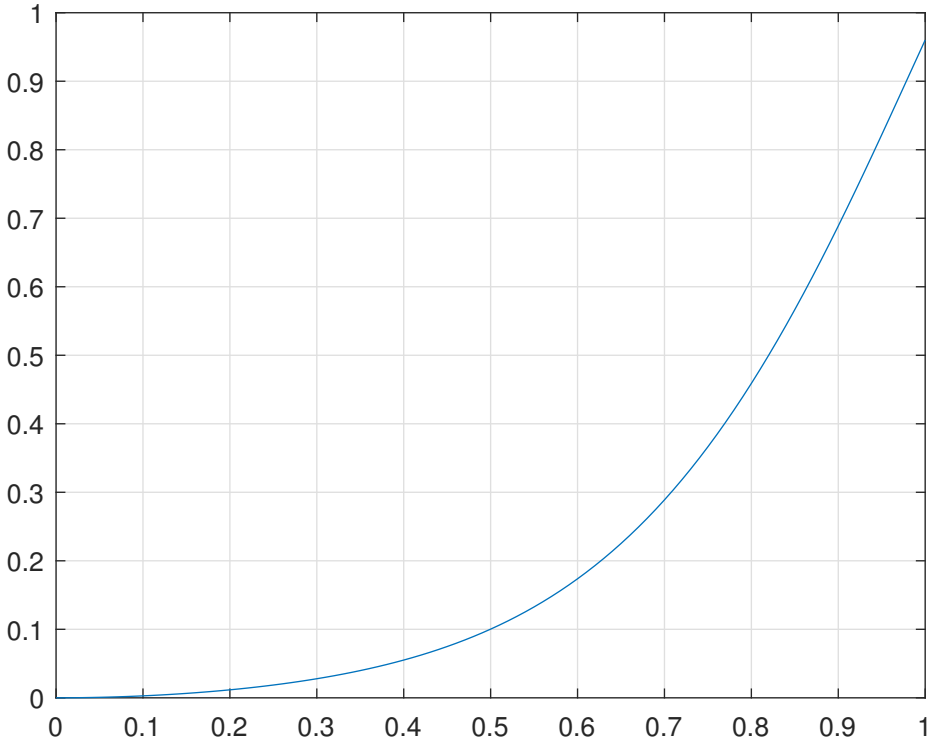
Slika 3.12. Kod za crtanje navedenih funkcija.



Slika 3.13. Sinusna trigonometrijska funkcija, $t = -\pi$ do π.



Slika 3.14. Eksponencijalna funkcija, t = 0 do 1.



Slika 3.15. Funkcija $f(t) = t^2 \cos(t^3 + 5)$, *t = 0 do 1.*

3.2.5. Simboličko rješavanje jednadžbi i uvrštavanje rješenja u neku drugu funkciju

Simboličko rješavanje jednadžbi, a potom uvrštavanje rješenja u neku drugu funkciju bit će pokazano na primjeru izračuna opsega i površine nekog pravokutnika. Odabrane su dvije jednadžbe s dvije nepoznanice koje su ubačene u MATLAB iz čega slijedi izračun stranica pravokutnika, a potom se dobivena rješenja ubacuju u funkciju koja izračunava opseg i površinu pravokutnika.

Jednadžbe koje su odabrane za dobivanje stranice pravokutnika su:

1. $2a + 4b = 6$
2. $4a - 8b = 4$

Gdje su a i b stranice pravokutnika.

```
diary('Sustav2Jed.txt');
diary on;
syms a b;
jed = [2.*a+4.*b-6==0,4.*a-8.*b-4==0];
rj = solve(jed,[a b])
a = rj.a
b = rj.b
[Opseg, Povrsina] = OP(a,b)
diary off;
function [Opseg,Povrsina] = OP(x,y)

     Opseg = 2.*x + 2.*y;
     Povrsina = x.*y;
end
```

Slika 3.16. Kod za rješavanje primjera

```
a = 2, b = 1/2, Opseg = 5, Povrsina = 1
```

Slika 3.17. Dobivena rješenja.

Kao što smo već naveli u uvodu, u ovom poglavlju smo pokazali pomoću kojih naredbi MATLAB može riješiti sustav dvije jednadžbe s dvije nepoznanice te ih potom ubaciti u neku drugu funkciju. Sustav dvije jednadžbe s dvije nepoznanice smo riješili pomoću naredbe:

`solve(jednadžbe, trazene varijable).`

Odabran je primjer opsega i površine pravokutnika čisto radi jednostavnosti budući da ipak to nije glavna tema rada. U sljedećem poglavlju, će uz izračune diferencijalnih jednadžbi biti grafički prikazano rješenje diferencijalne jednadžbe.

3.2.6. Simboličko rješavanje diferencijalnih jednadžbi i grafički prikaz rješenja

U odnosu na prethodno poglavlje i rješavanje sustava dvije jednadžbe s dvije nepoznanice, kod rješavanja diferencijalnih jednadžbi, imamo malo kompleksniju sintaksu pisanju koda.

Sintaksa koda za rješavanje diferencijalnih jednadžbi:

```
uvjeti = [uvjet1 uvjet2];
rjesenje = dsolve(jednadzba,uvjeti);
```

Dodatno se može ubaciti naredba $simplify()$ koja bi maksimalno pojednostavila dobiveno rješenje.

U ovom poglavlju nije smišljan neki posebni primjer nego se pomoću MATLAB-a rješava početni problem čije je rješenje potom grafički prikazano u MATLAB-u. Rješavamo sljedeći početni problem:

$$\begin{aligned} &\frac{d^2y}{dx^2} = \cos(2x) - y, \\ &y(0) = 1, y'(0) = 0. \end{aligned} \tag{3.1}$$

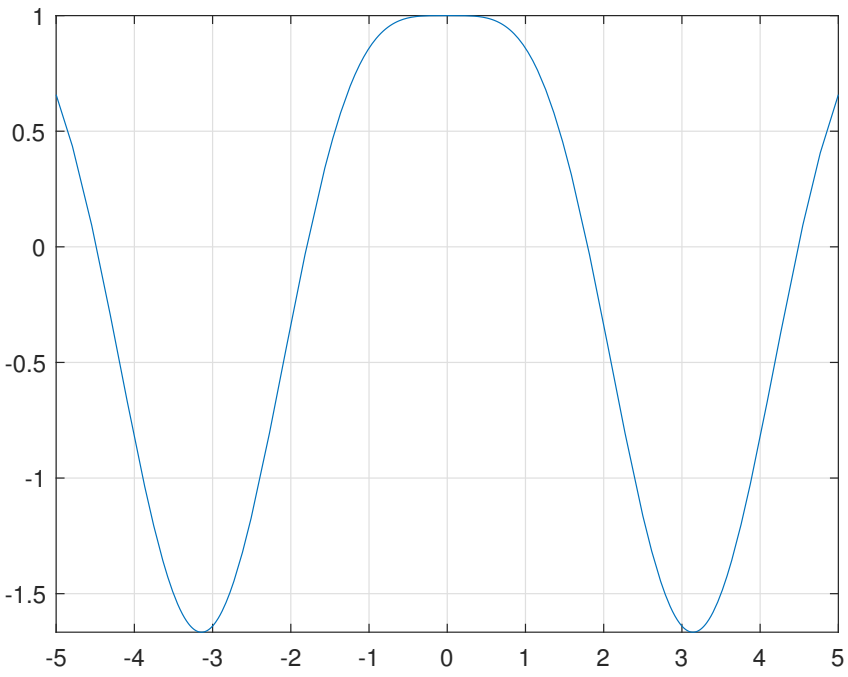
U nastavku slijede slike na kojima će biti prikazan kod potreban za rješavanje odabranog početnog problema, a potom i grafički prikaz dobivenog rješenja.

```
diary('DifJed.txt');
diary on;
syms x y(x); %definiranje varijabli
dy = diff(y);
uvjet1 = y(0) == 1; %definiranje uvjeta
uvjet2 = dy(0) == 0; %definiranje uvjeta
uvjeti = [uvjet1 uvjet2];
jedn = diff(y,x,2) == cos(2.*x) - y; %definiranje jednadzbe
rjesenje = dsolve(jedn,uvjeti); %rješavanje dif.jed.
rjesenje = simplify(rjesenje) %pojednostavljenje rj.
diary off;
figure();
x = 0:0.01:1;
fplot(rjesenje) %printanje grafa
grid on
```

Slika 3.18. Kod potreban za rješavanje diferencijalne jednadžbe i grafički prikaz dobivena rješenja.

```
rjesenje = 1 - (8*sin(x/2)^4)/3
```

Slika 3.19. Dobiveno rješenje provođenjem koda sa prethodne slike (bez grafičkog prikaza).



Slika 3.20. Grafički prikaz rješenja diferencijalne jednadžbe.

4. Laplaceove transformacije

U ovom poglavlju proći ćemo kroz povijesni razvoj Laplaceove transformacije te kroz život njezina utemeljitelja, Pierrea - Simona Laplacea. Također, analizirat ćemo neka svojstva Laplaceove transformacije te prikazat pozivanje Laplaceove transformacije i inverzne Laplaceove transformacije. Na kraju poglavlja bit će riješeno pet primjera u kojima će biti prikazano korištenje svega do tada navedenog kroz rad.

4.1. Povijest Laplaceovih transformacija

Pierre - Simon Laplace, rođen 23. ožujka 1749. godine, poznati je francuski matematičar i fizičar koji je najpoznatiji po tome što je istraživao stabilnost solarnog sustava. Pierre - Simon Laplace je prema [2] uspješno izračunao sva odstupanja (devijacije) planeta od teorijskih orbita planeta. Prema sadržaju koji je objavljen na [2], doznaje se kako je Pierre - Simon Laplace bio sin farmera, no o njegovom životu se ne zna mnogo. Zna se da je brzo počeo pokazivati svoje sposobnosti za matematiku još na vojnoj akademiji u Beaumontu. 1776. godine, Pierre - Simon Laplace je započeo pohađati Sveučilišta u Caenu, ali već se godinu dana kasnije, bez diplome, odlučio zaputiti u Pariz. U glavni francuski grad stigao je sa pismom preporuke za matematičara. Jean d'Alemberta[1] koji mu je osigurao poziciju profesora na Ecole Militaire[2] gdje je podučavao od 1769. do 1776. godine.

Na [3] pronalazimo informaciju da sam naziv integralne transformacije dolazi upravo u čast Pierre - Simona Laplacea zato što je koristio transformaciju u svojoj teoriji vjerojatnosti. Treba napomenuti da Pierre - Simon Laplace nije bio prvi u tome području nego je samo nastavio istraživanje koje je provodio Leonhard Paul Euler[3]. Prema [4], integrali koje je koristio Euler tijekom svojeg promatranja i izračuna glasili su:

$$z = \int X(x)e^{ax}dx, \tag{4.1}$$

gdje je $X(x)$ original funkcije, e^{ax} jezgra integralne transformacije, a z slika funkcije, te

$$z = \int X(x)x^{A}dx, \tag{4.2}$$

gdje je $X(x)$ original funkcije, x^A jezgra integralne transformacije, z slika funkcije, a A konstanta.

[1] Jean Le Rond d' Alembert, (Pariz, 17. studenog 1717. - Paris, 29. listopada 1783.) bio je francuski matematičar, znanstvenik, filozof i pisac.

[2] Ecole Militaire je vojna škola u Francuskoj prijestolnici

[3] Leonhard Paul Euler, (Basel, 15. travnja 1707. - Petersburg, 18. rujna 1783.) bio je švicarski matematičar i fizičar te jedan od utemeljitelja čiste matematike.

Joseph - Louis Lagrange[4] iznimno je poštovao Eulerov rad i u svojoj je teoriji vjerojatnosti istraživao izraz koji je glasio:

$$z = \int X(x)e^{-ax}e^{x}dx, \tag{4.3}$$

gdje je $X(x)$ original funkcije, $e^{-ax} \cdot e^{x}$ jezgra integralne transformacije, a z - slika funkcije.

Navedeni integrali privukli su Pierrea - Simona Laplacea 1782. godine gdje je on, ponukan Leonhardom Paulom Eulerom, koristio integrale kao rješenje jednadžbe.

1785. godine, Pierre - Simon Laplace napravio je potreban korak prema naprijed i odlučio u integrale ne gledati kao rješenja nego koristiti transformaciju koja je naknadno postala poznata.

Spomenimo da je promatrao također i integral oblika

$$\int x^{s}\phi(x)dx, \tag{4.4}$$

gdje je $\phi(x)$ original funkcije, x^{s} jezgra integralne transformacije, a s kompleksni broj. Ovaj integral vrlo je sličan definicijskom integralu transformacije koju danas zovemo Mellinovom[5].

Danas se Laplaceova transformacija definira na sljedeći način:

Definicija 4.1. *Neka je $f(t)$ zadana funkcija definirana na vremenskoj domeni $[0, \infty\rangle$. Funkciju*

$$\mathcal{L}\{f(t)\} = F(s) = \int_{0}^{\infty} f(t)e^{-st}dt \tag{4.5}$$

zovemo Laplaceovom transformacijom funkcije $f(t)$, pri čemu se funkcija $f(t)$ zove original, funkcija $F(s)$ slika, dok je e^{-st} jezgra integralne transformacije.

4.2. Egzistencija Laplaceove transformacije

Egzistencija Laplaceove transformacije jednaka je konvergenciji odnosno postojanju nepravog integrala:

$$\int_{0}^{\infty} f(t)e^{-st}dt, \tag{4.6}$$

tako da funkcija $f(t)$ može biti transformirana iz vremenske u Laplaceovu domenu tj. Laplaceova transformacija funkcije postoji ukoliko su ispunjeni sljedeći uvjeti:

1. Funkcija f je neprekidna po dijelovima s konačnim brojem prekida prve vrste. Funkcija ima prekid prve vrste ukoliko su limesi slijeva ili zdesna u točki konačni i različiti.

[4]Joseph – Louis Lagrange, originalno, Giusepee Luigi Gagrangia (Turin, 25. siječnja 1736. – Paris, 10. travnja 1813.) bio je talijansko – francuski matematičar koji je imao veliki doprinos u teoriji brojeva i analitičkoj mehanici.

[5]Robert Hjalmar Mellin (Finska, 19. lipnja 1854. - Finska, 4. travnja, 1933.) je poznati finski matematičar i teoretičar.

2. Funkcija f je eksponencijalnog rasta, odnosno ukoliko je funkciju f moguće omeđiti sa dvije eksponencijalne funkcije

Prema [10], upravo zbog drugog uvjeta egzistencije Laplaceove transformacije, neke funkcije poput $f(t) = e^{t^2}$ ne mogu se transformirati iz vremenske u Laplaceovu domenu.

Pokažimo sada na jednom primjeru kako se izračunava Laplaceova transformacija funkcije.

Primjer 4.1. *Odredimo Laplaceovu transformaciju funkcije $f(t) = e^t$. Prema definiciji Laplaceove transformacije vrijedi:*

$$\begin{aligned} F(s) = \mathcal{L}\{e^t\} &= \int_0^\infty e^t \cdot e^{-st} dt = \int_0^\infty e^{-t(s+1)} dt = \frac{-1}{s+1} \cdot e^{-t(s+1)} \Big|_0^\infty \\ &= \frac{-1}{s+1} \cdot e^{-\infty(s+1)} + \frac{1}{s+1} \cdot e^0 = 0 + \frac{1}{s+1} = \frac{1}{s+1}, \end{aligned} \tag{4.7}$$

pri čemu smo koristili da je $e^{-\infty} = 0$ i $e^0 = 1$.

U sljedećoj tablici prikazane su Laplaceove transformacije najčešće korištenih funkcija.

Tablica 4.1. Tablica Laplaceovih transformacija

$f(t)$	$F(s)$	$f(t)$	$F(s)$	$f(t)$	$F(s)$
$u(t) = 1$	$\frac{1}{s}$	$e^{\pm at} \sin \omega t$	$\frac{\omega}{(s \mp a)^2 + \omega^2}$	$t \sin \omega t$	$\frac{2\omega s}{(s^2 + \omega^2)^2}$
$e^{\pm at}$	$\frac{1}{s \mp a}$	$e^{\pm at} \cos \omega t$	$\frac{s \mp a}{(s \mp a)^2 + \omega^2}$	$t \cos \omega t$	$\frac{s^2 - \omega^2}{(s^2 + \omega^2)^2}$
$\sin \omega t$	$\frac{\omega}{s^2 + \omega^2}$	$e^{\pm at} \sinh \omega t$	$\frac{\omega}{(s \mp a)^2 - \omega^2}$	$t \sinh \omega t$	$\frac{2\omega s}{(s^2 - \omega^2)^2}$
$\cos \omega t$	$\frac{s}{s^2 + \omega^2}$	$e^{\pm at} \cosh \omega t$	$\frac{s \mp a}{(s \mp a)^2 - \omega^2}$	$t \cosh \omega t$	$\frac{s^2 + \omega^2}{(s^2 - \omega^2)^2}$
$\sinh \omega t$	$\frac{\omega}{s^2 - \omega^2}$	t^n	$\frac{n!}{s^{n+1}}$	$\delta(t)$	1
$\cosh \omega t$	$\frac{s}{s^2 - \omega^2}$	te^{-at}	$\frac{1}{(s+a)^2}$	$\delta(t-a)$	e^{-as}

4.3. Svojstva Laplaceove transformacije

U ovom ćemo poglavlju navesti najvažnija svojstva Laplaceove transformacije.

Teorem 4.1 (Linearnost Laplaceove transformacije)**.** *Neka su $f_1(t)$ i $f_2(t)$ zadane funkcije te c_1 i c_2 konstante. Tada vrijedi:*

$$\mathcal{L}\{c_1 f_1(t) + c_2 f_2(t)\} = c_1\mathcal{L}\{f_1(t)\} + c_2\mathcal{L}\{f_2(t)\}. \tag{4.8}$$

Teorem 4.2 (Teorem sličnosti)**.** *Neka je $f(t)$ zadana funkcija, a $\mathcal{L}\{f(t)\} = F(s)$ njena Laplaceova transformacija. Tada vrijedi:*

$$\mathcal{L}\{f(ct)\} = \frac{1}{c}F\left(\frac{s}{c}\right), \tag{4.9}$$

pri čemu je c konstanta.

Teorem 4.3 (Teorem o derivaciji slike)**.** *Neka je $f(t)$ zadana funkcija, a $\mathcal{L}\{f(t)\} = F(s)$ njena Laplaceova transformacija. Tada vrijedi:*

$$\mathcal{L}\{t^n f(t)\} = (-1)^n F^{(n)}(s), n \in \mathbb{N}. \tag{4.10}$$

Teorem 4.4 (Teorem o prigušenju (pomaku u slici))**.** *Neka je $f(t)$ zadana funkcija, a $\mathcal{L}\{f(t)\} = F(s)$ njena Laplaceova transformacija. Tada vrijedi:*

$$\mathcal{L}\{e^{\pm at} f(t)\} = F(s \mp a), \tag{4.11}$$

pri čemu je a realna konstanta.

Teorem 4.5 (Teorem o pomaku originala)**.** *Neka je $f(t)$ zadana funkcija, a $\mathcal{L}\{f(t)\} = F(s)$ njena Laplaceova transformacija. Tada vrijedi:*

$$\mathcal{L}\{f(t-a)u(t-a)\} = e^{-as}F(s), \tag{4.12}$$

pri čemu je a realna konstanta.

Teorem 4.6 (Teorem o slici derivacije)**.** *Neka je $f(t)$ zadana funkcija, a $\mathcal{L}\{f(t)\} = F(s)$ njena Laplaceova transformacija. Tada vrijedi:*

$$\mathcal{L}\{f^{(n)}(t)\} = s^n F(s) + \sum_{k=0}^{n-1} s^{n-k-1} f^{(k)}(0). \tag{4.13}$$

Teorem 4.7 (Teorem o slici integrala)**.** *Neka je $f(t)$ zadana funkcija, a $\mathcal{L}\{f(t)\} = F(s)$ njena Laplaceova transformacija. Tada vrijedi:*

$$\mathcal{L}\left\{\int_0^t f(\tau)d\tau\right\} = \frac{F(s)}{s} \tag{4.14}$$

Pokažimo sada na nekoliko primjera kako se primjenjuju navedena svojstva Laplaceove transformacije.

Primjer 4.2. *Riješimo diferencijalno integralnu jednadžbu zadanu s*

$$y'(t) - y = 2\int_0^t y(\tau)d\tau, \tag{4.15}$$

uz uvjet da je $y(0) = 3$*.*

Primijenimo li Laplaceovu transformaciju na zadanu jednadžbu dobivamo:

$$-y(0) + sY(s) - y(0) + sY(s) - Y(s) = \frac{2Y(s)}{s} \tag{4.16}$$

pri čemu smo označili da je $\mathcal{L}\{y(t)\} = Y(s)$*. Prilikom transformacije koristili smo se Teoremima 4.6 i 4.7.*

Rješavanjem dobivene algebarske jednadžbe i uvrštavanjem vrijednost $y(0)$ *slijedi:*

$$\begin{aligned} Y(s)\left(s - 1 - \frac{2}{s}\right) &= y(0), \\ Y(s)\frac{s^2 - s - 2}{s} &= 3, \\ Y(s) &= \frac{3s}{s^2 - s - 2}. \end{aligned} \tag{4.17}$$

Kako bi našli original funkcije $Y(s)$*, potrebno ju je rastaviti na parcijalne razlomke:*

$$\frac{3s}{s^2 - s - 2} = \frac{A}{s+1} + \frac{B}{s-2}. \tag{4.18}$$

Množenjem ovog izraza sa zajedničkim nazivnikom slijedi:

$$3s = A(s-2) + B(s+1). \tag{4.19}$$

Uvrstimo li sada da je $s = 2$ *te* $s = -1$ *odmah dobivamo da je* $A = 1$ *i* $B = 2$*, tj. vrijedi:*

$$Y(s) = \frac{1}{s+1} + \frac{2}{s-2}. \tag{4.20}$$

Kako bi odredili original funkcije $Y(s)$*, odnosno funkciju* $y(t)$ *koja je rješenje polazne jednadžbe, izraze s desne strane jednakosti* (4.20) *tražimo u tablici 4.1. Dobivamo:*

$$y(t) = e^{-t} + 2e^{2t}, \tag{4.21}$$

što je rješenje zadanog problema.

Primjer 4.3. *Riješimo diferencijalnu jednadžbu zadanu s*

$$y''(t) - y'(t) - 2y(t) = 4t^2, \tag{4.22}$$

ako vrijede početni uvjeti $y(0) = 1$ *i* $y'(0) = 4$*.*

Primjenom Teorema 4.6 na navedenu diferencijalnu jednadžbu te tablice 4.1 uz oznaku da je $\mathcal{L}\{y(t)\} = Y(s)$ *dobivamo:*

$$-y'(0) - sy(0) + s^2Y(s) + y(0) - sY(s) - 2Y(s) = 4 \cdot \frac{2!}{s^3}. \tag{4.23}$$

Uvrštavanjem početnih uvjeta slijedi:

$$-4 - s + s^2Y(s) + 1 - sY(s) - 2Y(s) = \frac{8}{s^3}. \tag{4.24}$$

Rješavanjem dobivene algebarske jednadžbe dobivamo:

$$Y(s)(s^2 - s - 2) = \frac{8}{s^3} + s + 3, \tag{4.25}$$

pa korištenjem jednakosti $s^2 - s - 2 = (s + 1)(s - 2)$ *slijedi:*

$$Y(s) = \frac{8 + s^4 + 3s^3}{s^3(s + 1)(s - 2)}. \tag{4.26}$$

Kao i u prethodnom primjeru dobivenu funkciju rastavljamo na parcijalne razlomke:

$$Y(s) = \frac{8 + s^4 + 3s^3}{s^3(s + 1)(s - 2)} = \frac{A}{s} + \frac{B}{s^2} + \frac{C}{s^3} + \frac{D}{s + 1} + \frac{E}{s - 2}. \tag{4.27}$$

Množenjem ovog izraza sa zajedničkim nazivnikom dobivamo jednakost:

$$s^4+3s^3+8 = As^2(s+1)(s-2)+Bs(s+1)(s-2)+C(s+1)(s-2)+Ds^3(s-2)+Es^3(s+1), \tag{4.28}$$

koja sređivanjem prelazi u:

$$s^4+3s^3+8 = A(s^4-s^3-2s^2)+B(s^3-s^2-2s)+C(s^2-s-2)+D(s^4-2s^3)+E(s^4+s^3). \tag{4.29}$$

Sada izjednačavamo koeficijente uz iste potencije pa dobivamo sustav jednadžbi:

$$\begin{aligned} s^4 &: 1 = A + D + E \\ s^3 &: 3 = -A + B - 2D + E \\ s^2 &: 0 = -2A - B + C \\ s^1 &: 0 = -2B + C \\ s^0 &: 8 = -2C, \end{aligned} \tag{4.30}$$

Rješavanjem ovog sustava dobivamo $A = -3$, $B = -2$, $C = -4$, $D = 2$ *i* $E = 2$. *Uvrstimo li dobivene koeficijente u jednadžbu* (4.27) *dobivamo:*

$$Y(s) = \frac{-3}{s} + \frac{2}{s^2} + \frac{-4}{s^3} + \frac{2}{s + 1} + \frac{2}{s - 2} \tag{4.31}$$

Sada, inverznom transformacijom izraza $Y(s)$ *uz korištenje tablice 4.1 dobijemo:*

$$y(t) = -3 + 2t - 2t^2 + 2e^{-t} + e^{2t}, \tag{4.32}$$

što je rješenje zadanog problema.

4.4. Primjena Laplaceove transformacije

U svakodnevnom radu, inženjeri se često susreću s Laplaceovom transformacijom. Primjena je jako puno, a ona najvažnija koju u ovom radu opisujemo je analiza električnih krugova, pri čemu se uvelike koristi rješavanje diferencijalnih jednadžbi opisano u prethodnim primjerima.

Jedna od važnijih primjena Laplaceove transformacije je u području analize signala i sustava. Veliki broj problema koji se pojavljuje u području analize signala i sustava nije rješiv bez korištenja Laplaceove transformacije.

Laplaceova transformacija koristi se u svim područjima u kojim se problemi rješavanju pomoću diferencijalnih jednadžbi, posebice ako te jednadžbe uključuju impulsne signale. Tako se one primjerice koriste u nuklearnoj fizici i analizi mehaničkih opterećenja.

5. Laplaceove transformacije u programskom sustavu MATLAB

U ovom poglavlju ćemo fokus staviti na primjenu Laplaceove transformacije te kroz nekoliko primjera pokazati implementaciju tehnika Laplaceove transformacije u programskom sustavu MATLAB

Prva dva primjera bit će transformacija odabranih funkcija iz vremenske u Laplaceovu domenu i obrnuto. U sljedeća tri primjera, radimo izračun struje unutar RC, RL i RLC kruga, pri čemu će se vrijednost ulaznog napona mijenjati od jednostavnijeg prema kompleksnijem.

5.1. Određivanje Laplaceove slike i originala u MATLAB-u

Kako bi mogli provesti Laplaceovu transformaciju funkcije $f(t)$ najprije je potrebno definirati nezavisne varijable originala i slike što radimo naredbom:

```
syms t s;
```

Nakon toga definira se funkcija u t-domeni, što radimo primjerice na sljedeći način:

```
f = exp(t);
```

gdje smo uzeli da je $f(t) = e^t$.

Laplaceovu transformaciju funkcije $f(t)$, odnosno funkciju $F(s)$ izračunavamo pomoću naredbe:

```
F (s) = laplace f(t);
```

Istaknimo da u odnosu na neke druge softvere, kod MATLAB-a nije potrebno definirati iz koje u koju domenu se transformira funkcija prilikom provođenja transformacije.

Inverznu Laplaceovu transformaciju izračunavamo naredbom

```
f (t) = ilaplace F(s);
```

Kod inverzne Laplaceove transformacije također nije potrebno definirati iz koje u koju varijablu se transformira odabrana funkcija.

Oba smjera transformacije bit će detaljnije prikazani u sljedećim primjerima.

5.2. Primjeri određivanja Laplaceove transformacije i inverzne Laplaceove transformacije u MATLAB-u

U sljedećem primjeru ćemo pomoću MATLAB-a odrediti Laplaceove transformacije nekoliko funkcija. Pritom ćemo uzeti primjere jednostavnih elementarnih funkcija i primjer funkcije koja je definirana na samo jednom dijelu domene.

Primjer 5.1. *Odredimo MATLAB kod za Laplaceovu transformaciju sljedećih funkcija:*

1. $f(t) = e^{3t} + \cos(6t) - e^{3t}\cos(6t)$,
2. $f(t) = \begin{cases} 2t, & 0 < t < 1 \\ 0, & t > 1 \end{cases}$,
3. $f(t) = t^{\frac{3}{2}}$.

Na sljedećoj slici možemo vidjeti kod potreban za određivanje traženih integralnih transformacija. U kod smo dodali i naredbe kojima ćemo mjeriti vrijeme potrebno za izračunavanje pojedine transformacije. Možemo primijetiti da je drugu funkciju najprije trebalo zapisati pomoću Heavisideove funkcije.

```
syms t s; %definiranje varijable
f1 = exp(3*t) + cos(3*t) - cos(6*t)*exp(3*t) %definiranje prve funkc.
f2 = 2*t*(heaviside(t-1) - heaviside(t)) %definiranje druge funkc.
f3 = t^(3/2) %definiranje trece funkc.

F1 = simplify(laplace(f1))
F2 = simplify(laplace(f2))
F3 = simplify(laplace(f3))

%naredba simplify sluzi za pojednostavljenje rjesenja.

t1 = timeit(@()laplace(F1))
t2 = timeit(@()laplace(F2))
t3 = timeit(@()laplace(F3)
```

Slika 5.1. Kod potreban za transformaciju odabranih funkcije iz vremenske u Laplaceovu domenu.

Na idućoj slici možemo vidjeti rezultat izvršavanja navedenog koda, odnosno Laplaceove transformacije zadanih funkcija.

```
F1 = 1/(s - 3) - (s - 3)/((s - 3)^2 + 36) + s/(s^2 + 9)

F2 = (2*exp(-s) + 2*s*exp(-s) - 2)/s^2

F3 = (3*pi^(1/2))/(4*s^(5/2))
```

Slika 5.2. Rješenja dobivena provođenjem koda sa prethodne slike.

Prve dvije transformacije su očekivane i lako ih možemo provjeriti korištenjem tablice 4.1. Funkcija iz trećeg primjera nije tablična i njenu bi transformaciju napravili, korištenjem definicije Laplaceove transformacije, na sljedeći način:

$$\mathcal{L}\{t^{\frac{3}{2}}\} = \int_0^\infty t^{\frac{3}{2}} e^{-st} dt. \tag{5.1}$$

Supstitucijom $u = st$ *dobivamo:*

$$\mathcal{L}\{t^{\frac{3}{2}}\} = \int_0^\infty \left(\frac{u}{s}\right)^{\frac{3}{2}} e^{-u} \frac{du}{s} = \frac{1}{s^{\frac{5}{2}}} \int_0^\infty u^{\frac{3}{2}} e^{-u} du = \frac{1}{s^{\frac{5}{2}}} \Gamma\left(\frac{5}{2}\right), \tag{5.2}$$

pri čemu je

$$\Gamma(x) = \int_0^\infty t^{x-1} e^{-x} dx \tag{5.3}$$

gama funkcija. Za gama funkciju vrijedi

$$\Gamma(x+1) = x\Gamma(x), \qquad \Gamma\left(\frac{1}{2}\right) = \sqrt{\pi}. \tag{5.4}$$

Korištenjem ovih svojstva dobivamo:

$$\mathcal{L}\{t^{\frac{3}{2}}\} = \frac{1}{s^{\frac{5}{2}}} \Gamma\left(\frac{5}{2}\right) = \frac{1}{s^{\frac{5}{2}}} \cdot \frac{3}{2} \cdot \frac{1}{2} \cdot \Gamma\left(\frac{1}{2}\right) = \frac{3\sqrt{\pi}}{4s^{\frac{5}{2}}}, \tag{5.5}$$

što je isti rezultat dobiven i pomoću MATLAB-a.

Na idućoj slici prikazano je i vrijeme izvršavanja koda. Usporedbom možemo primijetiti da je softveru najviše vremena trebalo za transformaciju funkcije $f(t) = e^{3t} + \cos(6t) - e^{3t}\cos(6t)$.

```
t1 = 0.4811

t2 = 0.0605

t3 = 0.0158
```

Slika 5.3. Vrijeme potrebno za Laplaceovu transformaciju odabrane funkcije.

U idućem se primjeru bavimo inverznom Laplaceovom transformacijom.

Primjer 5.2. *Odredimo MATLAB kod za inverznu Laplaceovu transformaciju sljedećih funkcija:*

1. $F(s) = \dfrac{3}{s^2+s-6}$,
2. $F(s) = \dfrac{e^{-s}}{s+1}$,
3. $F(s) = \dfrac{1}{s+1} + \dfrac{e^{-s}}{(s+1)^2+1} + e^{-2s}\dfrac{1}{(s+1)^2+1}$.

Na sljedećoj slici možemo vidjeti kod potreban za određivanje traženih inverznih transformacija. U kod smo dodali i naredbe kojima ćemo mjeriti vrijeme potrebno za izračunavanje pojedine transformacije.

```
syms s t; %definiranje varijable

F1 = 3/(s^2+s-6);
F2 = exp(-s)/(s+1);
F3 = 1/(s+1) + exp(-s)/((s+1)^2+1) + exp(-2*s)/((s+1)^2+1);

f1 = ilaplace(F1)
f2 = ilaplace(F2)
f3 = ilaplace(F3)

t1 = timeit(@()ilaplace(F1))
t2 = timeit(@()ilaplace(F2))
t3 = timeit(@()ilaplace(F3))
```

Slika 5.4. Kod potreban za transformaciju odabranih funkcije iz Laplaceove u vremensku domenu.

Na idućoj slici možemo vidjeti rezultat izvršavanja navedenog koda, odnosno inverzne Laplaceove transformacije zadanih funkcija.

```
f1 = (3*exp(2*t))/5 - (3*exp(-3*t))/5


f2 = heaviside(t - 1)*exp(1 - t)


f3 = exp(-t) + heaviside(t - 1)sin(t - 1)exp(1 - t) ...
     + heaviside(t - 2)sin(t - 2)exp(2 - t)
```

Slika 5.5. Rješenja dobivena provođenjem koda sa prethodne slike.

Svi dobiveni rezultati mogu se jednostavno provjeriti pomoću tablice Laplaceovih transformacija te Teorema o pomaku u originalu u slučaju druge dvije funkcije. Rezultat mjerenja vremena izvršavanja prikazan je na sljedećoj slici.

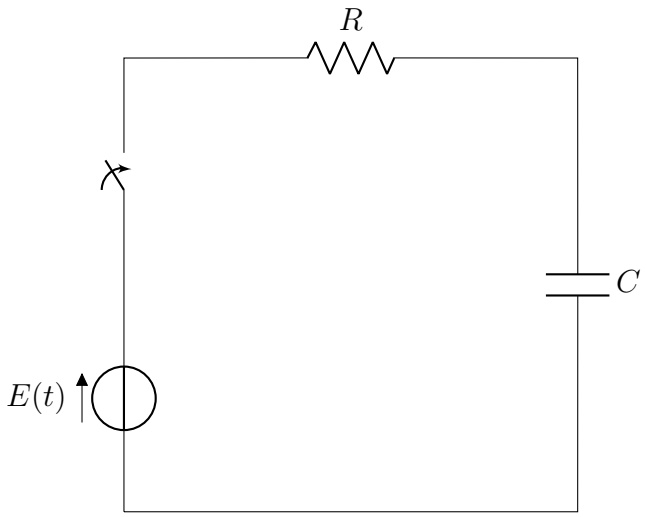
```
t1 = 0.0137

t2 = 0.0070

t3 = 0.0073
```

Slika 5.6. Vrijeme potrebno za inverznu transformaciju odabrane funkcije.

Usporedbom možemo primijetiti da je najviše vremena za inverznu transformaciju trebalo zapravo za najjednostavniju funkciju od sve tri odabrane, drugim riječima transformiranje skokovitih funkcija nije poseban problem za programski sustav MATLAB.

5.3. Analiza RC kruga tehnikom Laplaceovih transformacija

RC strujni krug sa slike 5.7 je serijski spoj otpornika (R) i kondenzatora (C). Pritom pretpostavljamo da su otpor otpornika i kapacitet kondenzatora vremenski neovisni. Na komponentama kroz koje prolazi struja, ostvaruje se pad napona čija suma, prema 2. Kirchhoffovom zakonu mora biti jednaka ulaznom naponu.



Slika 5.7. RC krug

U ovom ćemo poglavlju za RC strujni krug sa slike napisati MALTAB kod koji će izračunavati struju strujnog kruga ukoliko nam je ulazni napon jednak $E(t) = 10\,\mathrm{V}$, dok će vrijednosti komponenata strujnog kruga biti postavljene na $R = 1\,\Omega$ i $C = 1\,\mathrm{F}$.

Analiza strujnog kruga započinje već spomenutom primjenom 2. Kirchhoffova zakona prema kojemu ulazni napon strujnog kruga mora biti jednak sumi padova napona na svim komponentama unutar strujnog kruga. Primjenom Ohmova zakona

$$U_R = I \cdot R, \tag{5.6}$$

gdje je U_R napon na otporniku, I jakost struje, a R iznos otpora te same definicije kapaciteta kondenzatora

$$U_C = \frac{Q}{C}, \tag{5.7}$$

pri čemu je U_C napon na kondenzatoru, Q količina naboja, a C kapacitet kondenzatora, kao i definicije struje

$$i(t) = Q'(t), \tag{5.8}$$

odnosno

$$Q = \int_0^t i(t)dt \tag{5.9}$$

dolazimo do diferencijalno-integralne jednadžbe:

$$E(t) = i(t) \cdot R + \frac{1}{C} \int_0^t i(\tau) d\tau. \tag{5.10}$$

koja opisuje struju promatranog RC kruga.

Laplaceovom transformacijom dobivene jednadžbe, pri čemu primjenjujemo Teorem o slici integrala i uvrštavanjem vrijednosti otpora i kapaciteta dolazimo do sljedeće jednadžbe:

$$U(s) = I(s) + \frac{I(s)}{s}, \tag{5.11}$$

čijim rješavanjem dobivamo izraz za struju u s-domeni:

$$I(s) = E(s)\frac{s}{s+1}. \tag{5.12}$$

Uvrstimo li vrijednost ulaznog napona transformiranog iz vremenske u Laplaceovu domenu, tj. $E(s) = \frac{10}{s}$ dobit ćemo:

$$I(s) = \frac{10}{s+1}. \tag{5.13}$$

Korištenjem tablice 4.1 iz jednakosti (5.13) dolazimo do traženog rješenja u t-domeni:

$$i(t) = 10e^{-t}. \tag{5.14}$$

Sada ćemo navedeno rješenje dobiti pomoću softvera MATLAB. Nadolazeće slike prikazat će programski kod potreban za rješavanje.

```
syms s t tau e I(t) I(tau) I_L;
R = 1;
C = 1;
e(t) = @(t) 10;
jednadzba = e(t) - I(t)*R - 1/C*int(I(tau),tau,0,t) == 0
jednadzbaLT = laplace(e(t) - I(t)*R - 1/C*int(I(tau),tau,0,t)== 0)
jednadzbaLT = subs(jednadzbaLT,[laplace(I(t),t,s) laplace(I(t),t,s)/s] ...
,[I_L I_L])
I_L = solve(jednadzbaLT,I_L)
I = ilaplace(I_L)

fplot(e(t))
xlabel('t [s]');
ylabel('E [V]');
xlim([0 10])
ylim([0 15])
grid on;

fplot(I)
xlabel('t [s]');
ylabel('I [A]');
xlim([0 10])
ylim([0 15])
grid on;
```

Slika 5.8. Kod potreban za izračun struje RC strujnog kruga

Slijedi objašnjenje koda na slici 5.8. Nakon definiranja ulaznih varijabli, četvrti red sadrži jednadžbu (5.10). U sljedećem retku radi se Laplaceova transformacija jednadžbe (5.10), a potom i supstitucija kako bih jednadžba bila što jednostavnija. Pogledom na sliku 5.9 primijetimo što je supstitucija uz pomoć naredbe $subs$ donijela. Nadalje, pomoću naredbe $solve()$ rješavamo izvedenu algebarsku jednadžbu. U zadnjem retku gornjeg dijela koda, korištena je i naredba za inverznu Laplaceovu transformaciju koja je poslužila za vraćanje izraza struje iz Laplaceove s u vremensku t domenu.

Drugi dio koda služi za grafički prikaz dobivenog rješenja. Pomoću naredbe $fplot()$ grafički se prikazuje dobiveno rješenje. S obzirom da je izraz za struju u vremenskoj domeni jednak $i(t) = 10 \cdot e^{-t}$, to znači da će struja u trenutku $t = 0$ biti jednaka 10, a to je i grafički vidljivo. Napon nam je tijekom cijelog perioda konstantan te se krivulja napona neće promijeniti u niti jednom trenutku.

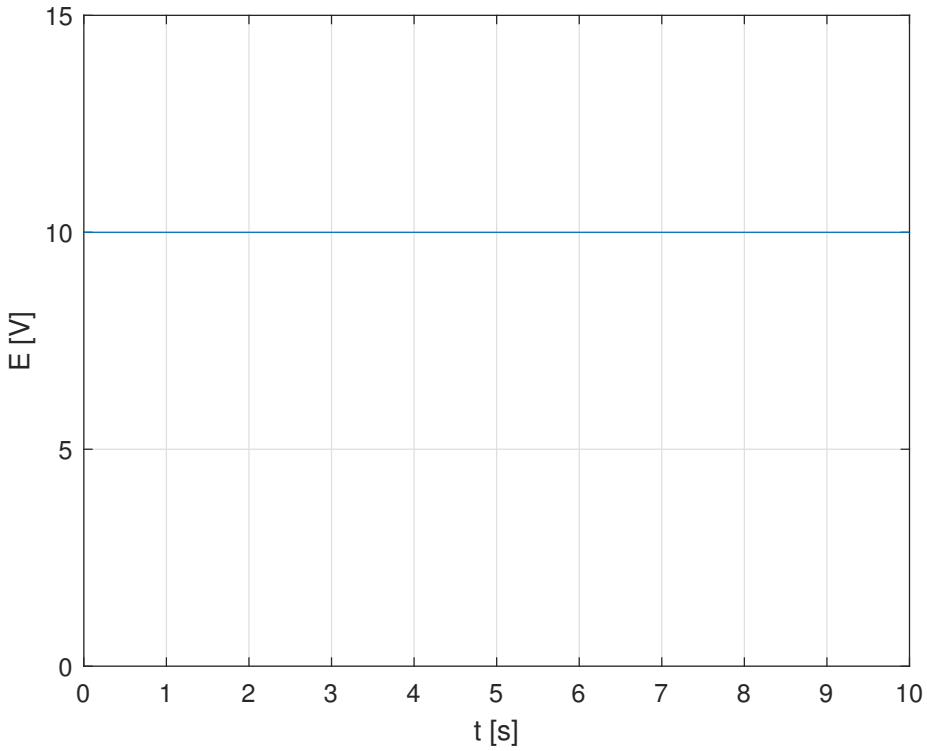
Slika 5.9 prikazuje dobiveno rješenje pokretanjem koda sa slike 5.8.

```
jednadzba =

10 - int(I(tau), tau, 0, t) - I(t) == 0


jednadzbaLT =

10/s - laplace(I(t), t, s)/s - laplace(I(t), t, s) == 0


jednadzbaLT =

10/s - I_L/s - I_L == 0


I_L =

10/(s + 1)


I =

10*exp(-t)
```

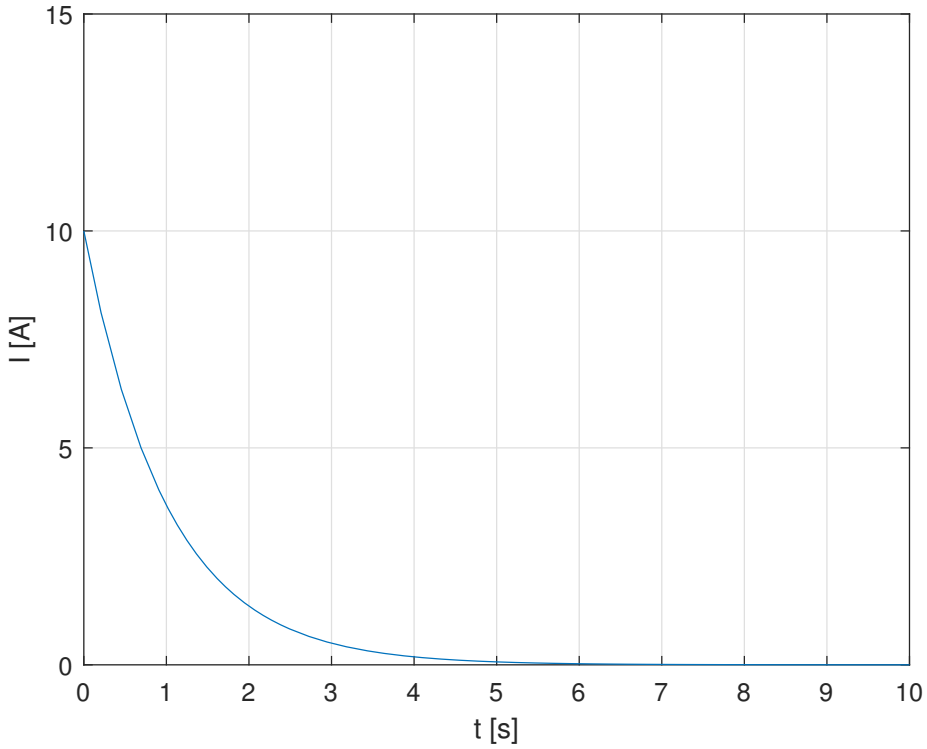
Slika 5.9. Rješenje dobiveno izvršenjem koda sa slike 5.8

Prvi red rješenja prikazuje jednadžbu u vremenskoj domeni, potom transformiranu u Laplaceovu domenu. Treći redak dobije se nakon što se pomoću naredbe $subs$ supstituiraju kompleksniji izrazi. Četvrti redak rješenja prikazuje konačnu struju u Laplaceovoj domeni, a peti redak prikazuje provođenje inverzne transformacije i rješenje u vremenskoj domeni.

Ulazni napon i dobiveno rješenje prikazati ćemo i grafički.



Slika 5.10. Grafički prikaz ulaznog napona RC kruga.



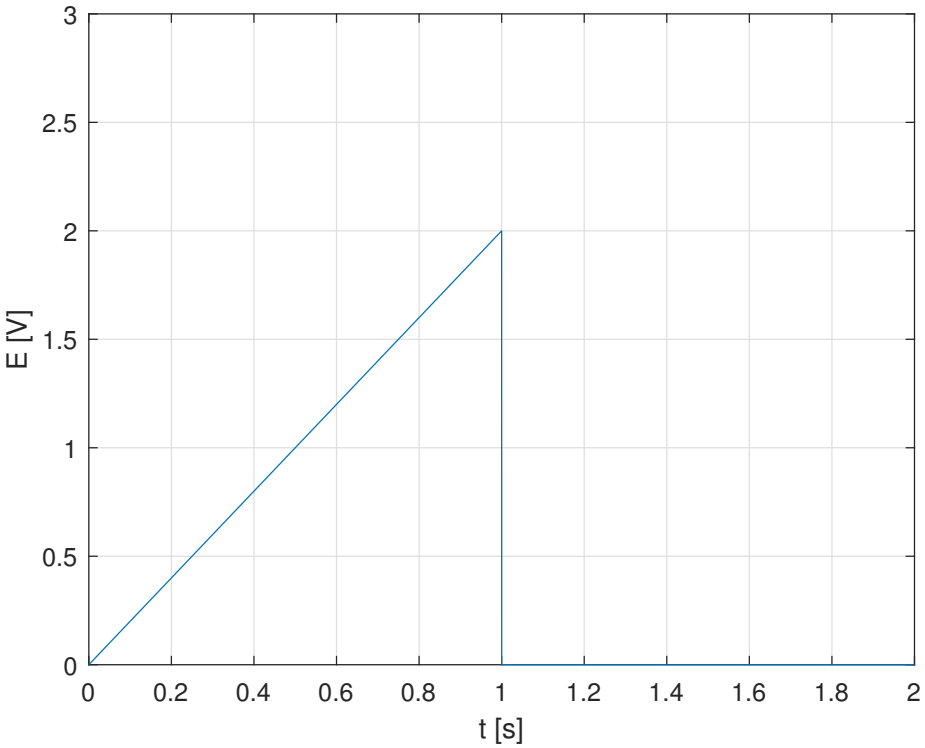
Slika 5.11. Grafički prikaz struje RC kruga.

Kod sa slike 5.8 ne mora nužno biti korišten samo za konstantnu vrijednost ulaznog napona. Također, vrijednosti komponenata R i C se mogu izmijeniti. Njihova promjena značila bi naravno promjenu u konačnom izrazu prije, ali i nakon inverzne transformacije. Što se tiče ulaznog napona, mogla je vrijednost ulaznog napona biti postavljena kao jedna od trigonometrijskih funkcija ili grafički prikazana sa raznim promjenama.

Kako bi se dokazala univerzalnost koda sa slike 5.8, pomoću softvera MALTAB napravit ćemo još jedan primjer gdje će biti promijenjeni parametri ulaznog napona i komponenata strujnog kruga. Vrijednosti komponenata strujnog kruga bit će $R = 5\,\Omega$ i $C = 10\,\mu\text{F}$. Ulazni napon bit će

$$E(t) = \begin{cases} 2t, & 0 < t < 1 \\ 0, & t > 1 \end{cases}. \tag{5.15}$$

Ovaj ulazni napon grafički je prikazan na sljedećoj slici.

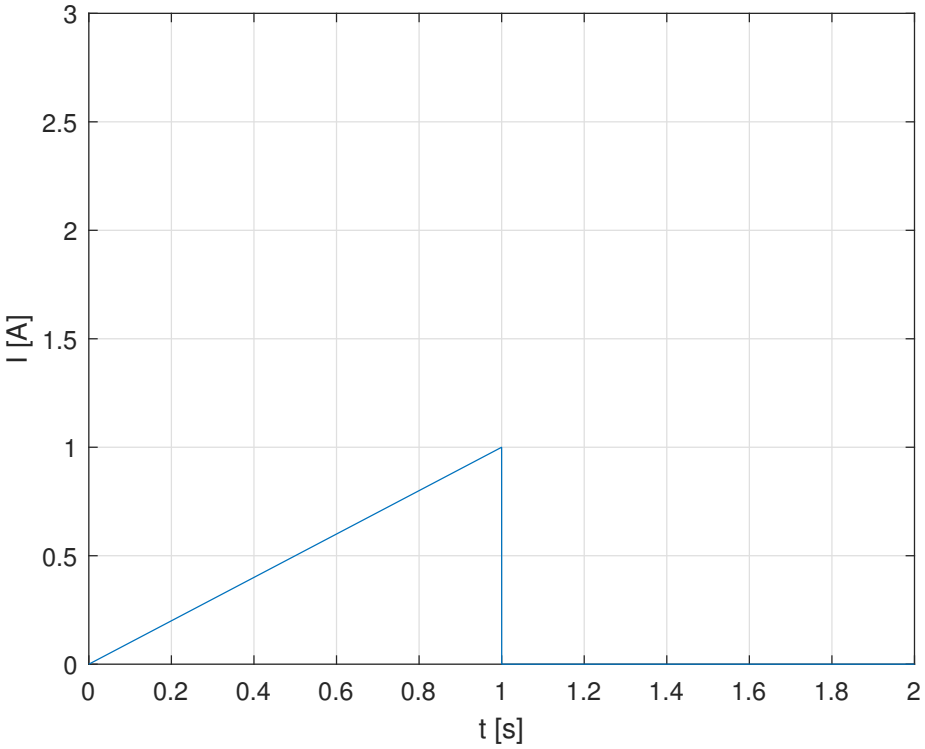


Slika 5.12. Grafički prikaz ulaznog napona RC kruga nakon unesenih promjena.

Za ovako postavljene uvjete MATLAB-ov kod dao je struju kruga oblika:

$$i(t) = t - u(t-1) - (t-1)u(t-1), \tag{5.16}$$

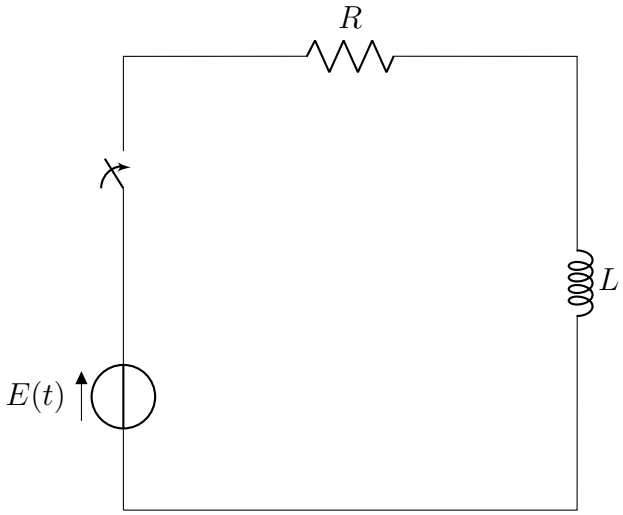
koja je prikazana na sljedećoj slici.



Slika 5.13. Grafički prikaz struje RC kruga nakon unesenih promjena.

5.4. Analiza RL kruga tehnikom Laplaceovih transformacija

RL strujni krug sa slike 5.14 je serijski spoj otpornika (R) i zavojnice (L). Kroz komponente prolazi struja, a njezinim prolaskom se ostvaruje pad napona na komponentama čija suma prema 2. Kirchhoffovom zakonu mora biti jednaka ulaznom naponu. Otpor otpornika i induktivitet zavojnice su vremenski neovisni.



Slika 5.14. RL krug

U ovom poglavlju ćemo za RL strujni krug sa slike 5.14 napisati MATLAB kod koji će izračunati struju strujnog kruga. Ulazni napon bit će nam zadan sinusnom funkcijom, odnosno bit će oblika

$$E(t) = \sin(\omega t), \tag{5.17}$$

pri čemu je ω kružna frekvencija. U proračunu ćemo uzeti da je $\omega = 1\,\frac{\text{rad}}{\text{sec}}$, a vrijednosti komponenata strujnog kruga bit će postavljene na $R = 1\,\Omega$ i $L = 1\,\text{H}$. Također pretpostavljamo da je vrijednost struje u trenutku $t = 0$ jednaka $0\,\text{A}$.

Analiza strujnog kruga započinje već spomenutom primjenom 2. Kirchhoffova zakona prema kojemu ulazni napon strujnog kruga mora biti jednak sumi padova napona na svim komponentama unutar strujnog kruga. Za napon na otporniku, kao i u prethodnom primjeru koristimo Ohmov zakon, tj.

$$U_R = I \cdot R. \tag{5.18}$$

Za napon na zavojnici koristimo Faradayev zakon i definiciju induktiviteta zavojnice prema kojima je

$$U_L = L \cdot \frac{di}{dt}, \tag{5.19}$$

pri čemu je U_L napon zavojnice, L induktivitet zavojnice, a i trenutačna vrijednost struje te time dolazimo do diferencijalne jednadžbe:

$$E(t) = i(t) \cdot R + L \cdot \frac{di(t)}{dt}, \tag{5.20}$$

koja opisuje struju promatranog RL kruga.

Primjenom Laplaceove transformacije na diferencijalnu jednadžbu (5.20) dolazi se do sljedećeg:

$$E(s) = I(s)R + L(s \cdot I(s) - i(0)). \tag{5.21}$$

Uvrste li se zadane vrijednosti i riješi dobivena algebarska jednadžba, slijedi

$$I(s) = \frac{1}{(s^2+1)(s+1)}. \tag{5.22}$$

Jednadžbu 5.22 potrebno je rastaviti na parcijalne razlomke, a to se radi na sljedeći način:

$$\frac{1}{(s^2+1)(s+1)} = \frac{As+B}{s^2+1} + \frac{C}{s+1} \tag{5.23}$$

Množenjem i sređivanjem izraza 5.23 dobije se sljedeća jednakost:

$$1 = A(s^2+s) + B(s+1) + C(s^2+1). \tag{5.24}$$

Izjednačenjem koeficijenta uz istu potenciju, dobivamo:

$$\begin{aligned} s^2 &: 0 = A + C, \\ s^1 &: 0 = A + B, \\ s^0 &: 1 = B + C. \end{aligned} \tag{5.25}$$

Rješavanjem ovog sustava jednadžbi, dolazi se do rješenja: $A = -\frac{1}{2}$, $B = \frac{1}{2}$ i $C = \frac{1}{2}$. Uvrštavanjem koeficijenata A, B i C u jednadžbu (5.23) dolazi se do sljedećeg izraza:

$$I(s) = -\frac{1}{2} \cdot \frac{s}{s^2+1} + \frac{1}{2} \cdot \frac{1}{s^2+1} + \frac{1}{2} \cdot \frac{1}{s+1}. \tag{5.26}$$

Tražeći u Tablici 4.1 desnu stranu jednakosti izraza (5.26), dolazi se do:

$$i(t) = \frac{-1}{2} \cdot \cos(t) + \frac{1}{2}\sin(t) + e^{-t}. \tag{5.27}$$

Kao i u poglavlju 5.3., do ovog rješenja ćemo doći i pomoću softvera MATLAB. Pritom koristimo sljedeći kod.

```
syms s t e I(t) I_L w;
R = 1;
L = 1;
w = 1;
e(t) = sin(w*t);
jednadzba = e(t) - I(t)*R - L*diff(I(t),t) == 0
jednadzbaLT = laplace(jednadzba)
jednadzbaLT= subs(jednadzbaLT,[laplace(I(t),t,s) s*laplace(I(t),t,s) I(0)]
,[I_L I_L 0])
I_L = solve(jednadzbaLT, I_L)
I = ilaplace(I_L)

fplot(e(t))
xlabel('t [s]');
ylabel('E [V]');
xlim([-2 2])
ylim([-1.5 1.5])
grid on;
```

Slika 5.15. Kod za izračunavanje struje RL strujnog kruga

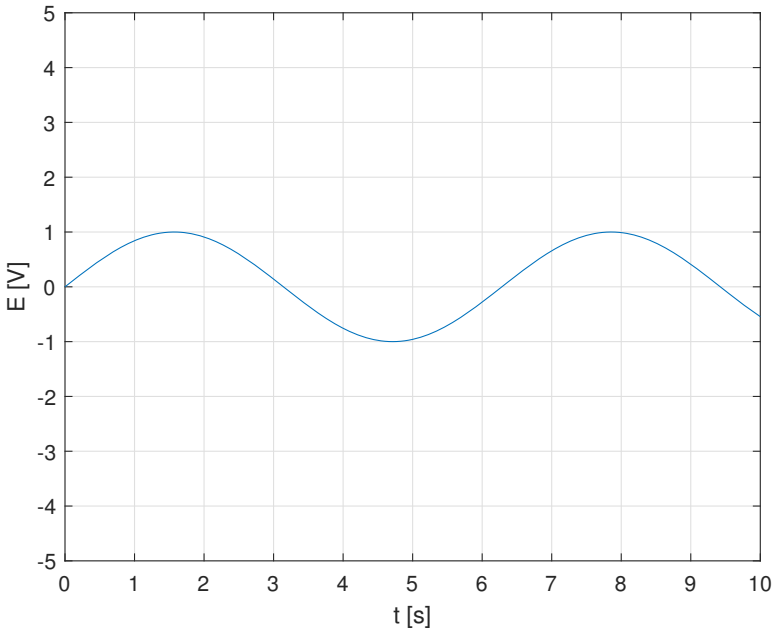
Kao i u prethodnom primjeru, objasnit ćemo kod sa slike 5.15. Kod započinje definiranjem ulaznih varijabli (R i L) i ulaznog napona. Potom se definira jednadžba (5.20) koja se u sljedećem koraku transformira pomoću Laplaceove transformacije. Predzadnji korak je supstitucija kompliciranijih izraza te izračunavanje struje strujnog kruga u Laplaceovoj domeni. Zadnji korak u gornjem dijelu koda je inverzna Laplaceova transformacija struje strujnog kruga. Drugi dio koda će omogućiti grafički prikaz ulaznog napona i struje strujnog kruga.

Rezultat izvršavanja ovog programskog koda možemo vidjeti na sljedećoj slici.

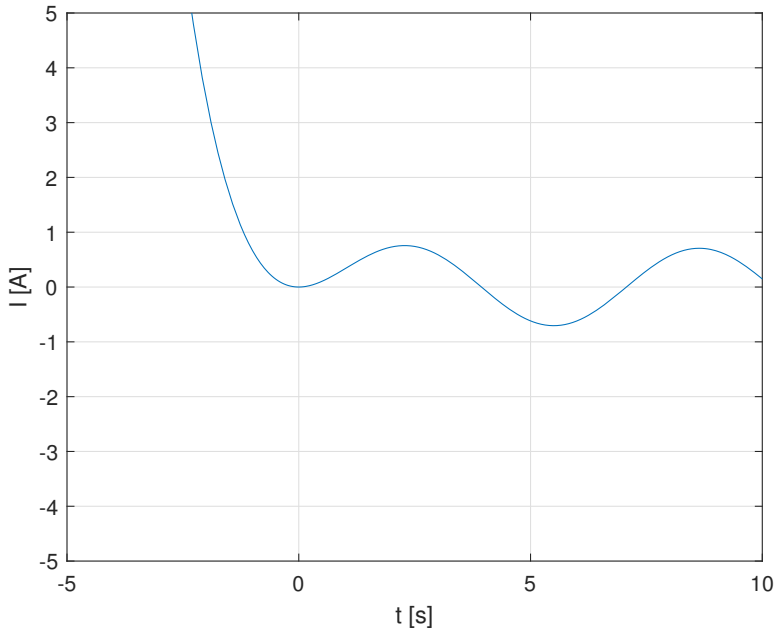
```
jednadzba =

sin(t) - I(t) - diff(I(t), t) == 0


jednadzbaLT =

I(0) + 1/(s^2 + 1) - s*laplace(I(t), t, s) - laplace(I(t), t, s) == 0


jednadzbaLT =

1/(s^2 + 1) - I_L*s - I_L == 0


I_L =

1/(s^3 + s^2 + s + 1)


I =

exp(-t)/2 - cos(t)/2 + sin(t)/2
```

Slika 5.16. Kod za rješavanje primjera 5.4.

Na sljedeće dvije slike možemo vidjeti ulazni napon i struju u strujnom krugu.



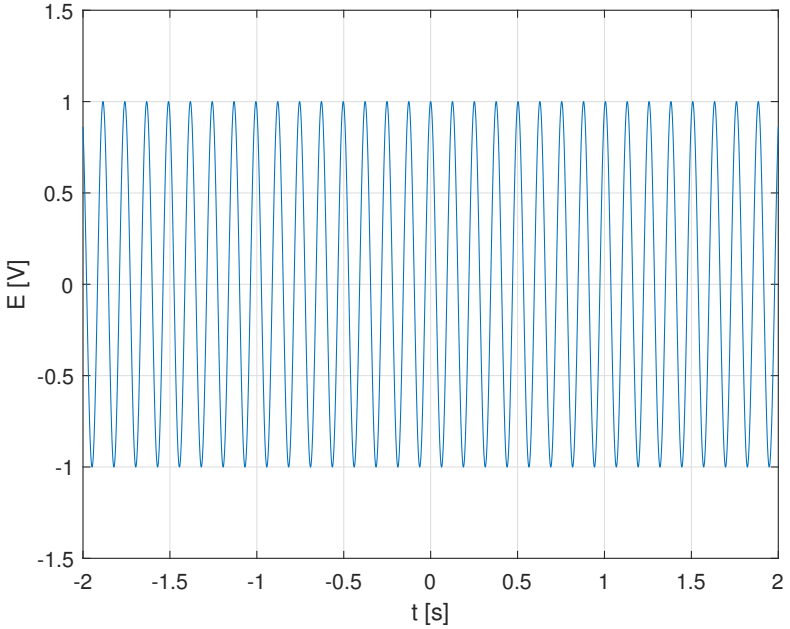
Slika 5.17. Grafički prikaz ulaznog napona RL kruga.



Slika 5.18. Grafički prikaz struje RL kruga.

Kao i kod RC strujnog kruga, ovdje će se također pokazati univerzalnost koda. Promijenit će se vrijednost varijable ω kao i vrijednost komponenata strujnog kruga. Ulazni napon će biti promijenjen iz sinusnog u kosinusni. Vrijednosti komponenata strujnog kruga bit će postavljene na $R = 15\,\Omega$ i $L = 2\,\mathrm{mH}$, a $\omega = 50\,\frac{\mathrm{rad}}{\mathrm{sec}}$.

Grafički prikaz novog ulaznog napona prikazan je na sljedećoj slici.

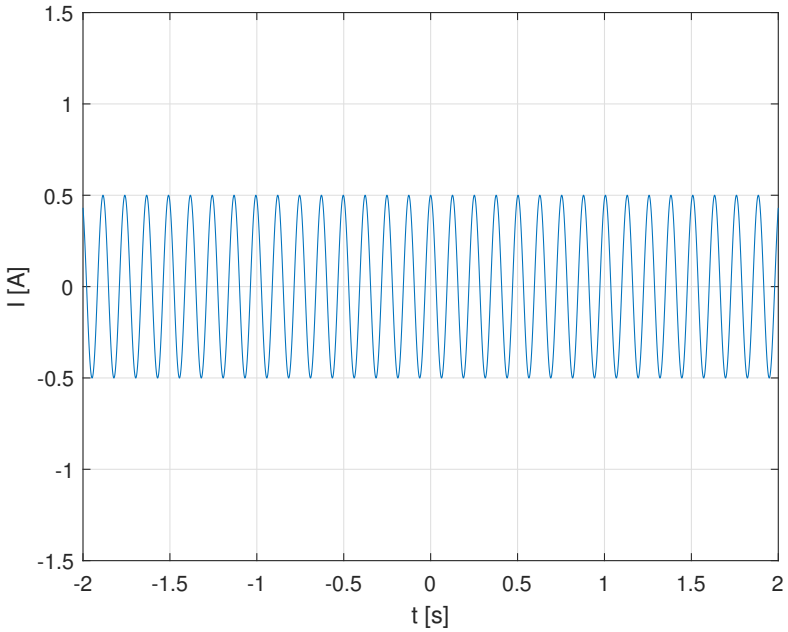


Slika 5.19. Grafički prikaz napona RL kruga nakon unesenih promjena.

Za ovako postavljene uvjete MATLAB-ov kod dao je sljedeću struju kruga:

$$i(t) = \frac{\cos(50 \cdot t)}{2}, \tag{5.28}$$

čiji grafički prikaz možemo vidjeti na sljedećoj slici.



Slika 5.20. Grafički prikaz struje RL kruga nakon unesenih promjena.

U ovom slučaju, nakon promjene nekih parametara, struja je zadržala oblik kao i ulazni napon, što nije bio slučaj kada je ulazni napon bio sinusna funkcija. Matematički gledano, to je uzrokovano eksponencijalnom funkcijom koja je na slici 5.18 početni dio rješenja. Naknadno se izgled funkcije vraća ka izvornoj, ali kao i u slučaju promjene ulaznog napona i vrijednosti komponenata, zbog utjecaja komponenata RL kruga, dolazi do minimalne razlike u fazi između napona i struje, a to se zove fazni pomak. Na oko nije vidljiv fazni pomak između ulaznog napona i struje, ali to je zato što je fazni pomak jednak 2.4°. Dokažimo.

$$\varphi = \arctan(\frac{X_L}{R}), \tag{5.29}$$

gdje je:

$$X_L = \omega \cdot L. \tag{5.30}$$

Uvrštavanjem vrijednosti, dobije se:

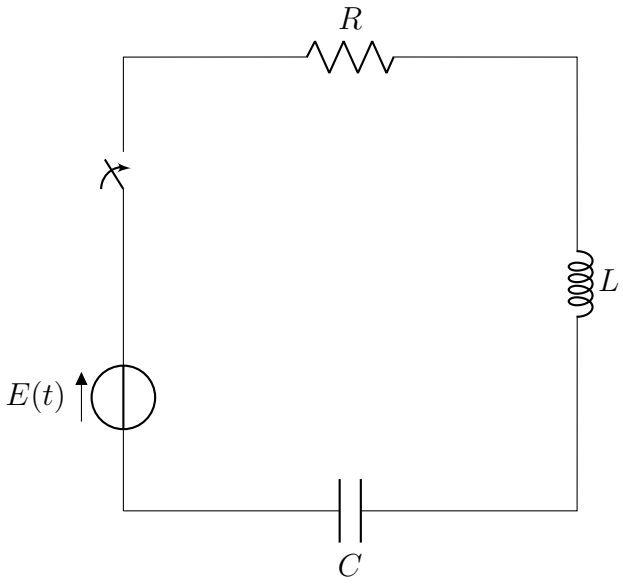
$$X_L = 2 \cdot 10^{-3} \cdot 50 = 0.1\,\Omega. \tag{5.31}$$

Povratkom na jednadžbu (5.29) i uvrštavanjem vrijednosti za X_L, dobije se:

$$\varphi = \arctan\left(\frac{0.1}{15}\right) = 0.38°. \tag{5.32}$$

5.5. Analiza RLC kruga tehnikom Laplaceovih transformacija

RLC strujni krug sa slike 5.21 je serijski spoj otpornika (R), zavojnice (L) i kondenzatora (C). Kroz komponente prolazi struja, a njezinim prolaskom se ostvaruje pad napona na komponentama čija suma prema 2. Kirchhoffovom zakonu mora biti jednaka ulaznom naponu. Otpor otpornika, induktivitet zavojnice kao i kapacitet kondenzatora su vremenski neovisni.



Slika 5.21. RLC krug

U ovom poglavlju ćemo za RLC strujni krug sa slike 5.21 napisati MATLAB kod koji će izračunati struju strujnog kruga. Ulazni napon bit će zadan grafičkim prikazom.

U proračunu ćemo uzeti da su vrijednosti komponenata strujnog kruga postavljene na $R = 1\,\Omega$, $L = 1\,\mathrm{H}$ i $C = 1\,\mathrm{F}$. Također, pretpostavlja se da je vrijednost struje u trenutku $t = 0$ jednaka $0\,\mathrm{A}$.

Analiza strujnog kruga započinje već spomenutom primjenom 2. Kirchhoffova zakona prema kojemu ulazni napon strujnog kruga mora biti jednak sumi padova napona na svim komponentama unutar strujnog kruga. Za napon na otporniku, primjenjuje se Ohmov zakon

$$U_R = I \cdot R, \tag{5.33}$$

gdje je U_R napon na otporniku, I jakost struje, a R iznos otpora.

Za napon na zavojnici koristimo Faradayev zakon i definiciju induktiviteta zavojnice prema kojima je

$$U_L = L \cdot \frac{di}{dt}, \tag{5.34}$$

pri čemu je U_L napon zavojnice, L induktivitet zavojnice, a i trenutačna vrijednost struje, a iz definicije kapaciteta kondenzatora je

$$U_C = \frac{Q}{C}, \tag{5.35}$$

pri čemu je U_C napon na kondenzatoru, Q količina naboja, a C kapacitet kondenzatora. Također koristimo i definiciju struje

$$i(t) = Q'(t), \tag{5.36}$$

odnosno

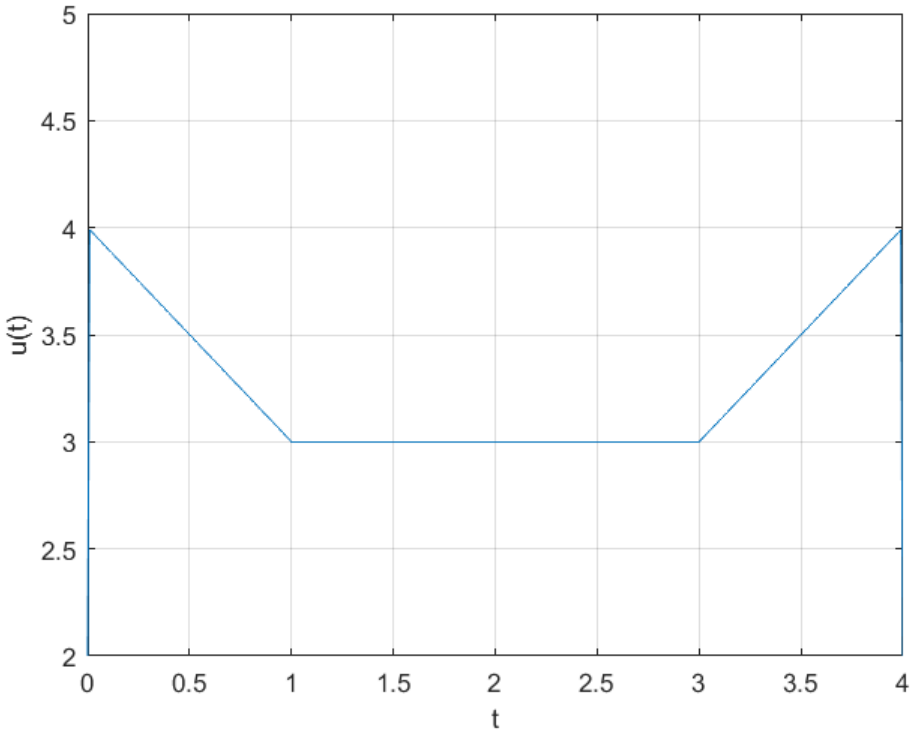
$$Q = \int_0^t i(t)dt. \tag{5.37}$$

Tako dolazimo do diferencijalno - integralne jednadžbe:

$$u(t) = i(t) \cdot R + L \cdot \frac{di(t)}{dt} + \frac{1}{C} \int_0^t i(\tau)d\tau, \tag{5.38}$$

koja opisuje struju promatranog RLC kruga.

Ulazni napon definiran je grafičkim prikazom danim na sljedećoj slici.



Slika 5.22. Ulazni napon.

U svrhu rješavanja zadatka, ulazni napon zapisat ćemo u algebarskom obliku:

$$
\begin{aligned}
u(t) &= (4-t)\cdot(u(t)-u(t-1)) + 3\cdot(u(t-1)-u(t-3)) \\
&\quad + t\cdot(u(t-3)-u(t-4)) \\
u(t) &= 4u(t) - tu(t) - (4-t)u(t-1) + 3u(t-1) \\
&\quad - 3u(t-3) + tu(u-3) - tu(t-4) \\
u(t) &= 4u(t) - tu(t) + (t-1)u(t-1) - 3u(t-1) + 3u(t-1) - 3u(t-3) + ... \\
&\quad ((t-3)+3)u(t-3) - ((t-4)+4)u(t-4) \\
u(t) &= 4u(t) - tu(t) + (t-1)u(t-1) - 3u(t-3) + (t-3)u(t-3) + 3u(t-3)... \\
&\quad - (t-4)u(t-4) - 4u(t-4).
\end{aligned} \tag{5.39}
$$

Na jednadžbu (5.39) djelujemo Laplaceovom transformacijom te dobivamo:

$$U(s) = 4\cdot\frac{1}{s} - \frac{1}{s^2} + \frac{1}{s^2}\cdot e^{-s} - 3\cdot\frac{1}{s}\cdot e^{-3s} + \frac{1}{s^2}\cdot e^{-3s} + \frac{3}{s}\cdot e^{-3s} - \frac{1}{s^2}\cdot e^{-4s} - \frac{4}{s}\cdot e^{-4s}, \tag{5.40}$$

što sređivanje prelazi u:

$$U(s) = \frac{4s-1}{s^2} - \frac{3s-1}{s^2}\cdot e^{-s} + \frac{3}{s}\cdot e^{-s} + \frac{1}{s^2}\cdot e^{-3s} - \frac{4}{s}\cdot e^{-4s} - \frac{1}{s^2}\cdot e^{-4s}. \tag{5.41}$$

Nakon ulaznog napona, sada se primjenjuje Laplaceova transformacija na jednadžbu (5.38). Dobivamo:

$$U(s) = I(s)R + L(s\cdot I(s) - i(0)) + \frac{I(s)}{s}. \tag{5.42}$$

Budući da je u tekstu zadatka zadano da je vrijednost struje u trenutku $t = 0$ jednaka nuli, tada taj dio možemo izbaciti pa nam jednadžba za $U(s)$ glasi:

$$U(s) = I(s)R + Ls\cdot I(s) + \frac{I(s)}{s}. \tag{5.43}$$

Uvrstimo li vrijednosti za R, L i C dobivamo:

$$U(s) = I(s) + s\cdot I(s) + \frac{I(s)}{s}. \tag{5.44}$$

Sada izlučimo $I(s)$, sredimo jednadžbu te nam izraz za izračun struje $I(s)$ glasi:

$$I(s) = \frac{U(s)\cdot s}{s^2+s+1} \tag{5.45}$$

Uvrštavanjem ulaznog napona u jednadžbu 5.45, sređivanjem izraza (parcijalni razlomci) te primjenom inverzne Laplaceove transformacije dobivamo sljedeći izraz:

$$
\begin{aligned}
i(t) = &-u(t) + e^{-\frac{1}{2}t}\cos\left(\frac{\sqrt{3}}{2}t\right) + 3\sqrt{3}\cdot e^{\frac{-1}{2}t}\sin\left(\frac{\sqrt{3}}{2}t\right) + u(t-1) \\
&- e^{-\frac{1}{2}(t-1)}\cos\left(\frac{\sqrt{3}}{2}(t-1)\right)u(t-1) - \frac{7}{\sqrt{3}}e^{-\frac{1}{2}(t-1)}\sin\left(\frac{\sqrt{3}}{2}(t-1)\right)u(t-1) \\
&+ 2\sqrt{3}e^{-\frac{1}{2}(t-1)}\sin\left(\frac{\sqrt{3}}{2}(t-1)\right)u(t-1) + u(t-3) - \\
&e^{-\frac{1}{2}(t-3)}\cos\left(\frac{\sqrt{3}}{2}(t-3)\right)u(t-3) - \frac{1}{\sqrt{3}}e^{-\frac{1}{2}(t-3)}\sin\left(\frac{\sqrt{3}}{2}(t-3)\right)u(t-3) \\
&\frac{8}{\sqrt{3}}e^{-\frac{1}{2}(t-4)}\sin\left(\frac{\sqrt{3}}{2}(t-4)\right)u(t-4) - u(t-4) + \\
&e^{-\frac{1}{2}(t-4)}\cos\left(\frac{\sqrt{3}}{2}(t-4)\right)u(t-4) - u(t-4) \\
&+ e^{-\frac{1}{2}(t-4)}\cos\left(\frac{\sqrt{3}}{2}(t-4)\right)u(t-4) + \frac{1}{\sqrt{3}}e^{-\frac{1}{2}(t-4)}\sin\left(\frac{\sqrt{3}}{2}(t-4)\right)
\end{aligned}
\tag{5.46}
$$

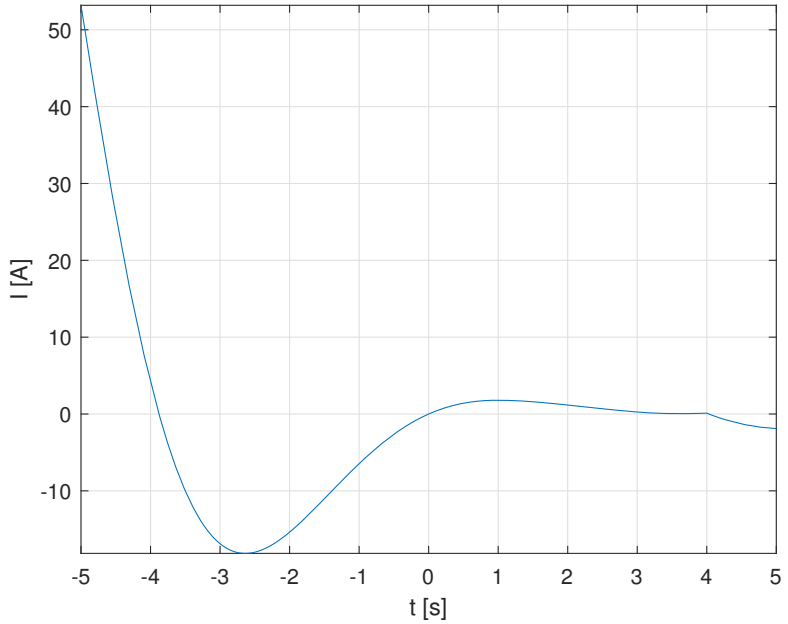
Isto rješenje bilo je potrebno dobiti pomoću softvera MATLAB. Na sljedećim slikama bit će prikazan prvo kod koji je bio potreban za izračun struje u strujnom krugu, a potom i prikaz rješenja.

```
diary('RLCKrug.txt')
diary on;
syms s t tau e I(t) I(tau) I_L;
R = 1;
C = 1;
L = 1;
e(t) = (4-t)*heaviside(t) - (4-t)*heaviside(t-1) + 3*(heaviside(t-1)- ...
    heaviside(t-3)) + (t)*(heaviside(t-3)-heaviside(t-4));
jednadzba = e(t) - I(t)*R - 1/C*int(I(tau),tau,0,t) - L*diff(I(t),t) == 0
jednadzbaLT=laplace(e(t)-I(t)*R-1/C*int(I(tau),tau,0,t)-L*diff(I(t),t)== 0)
jednadzbaLT = subs(jednadzbaLT,[laplace(I(t),t,s) laplace(I(t),t,s)/s I(0)
s*laplace(I(t),t,s)],[I_L I_L 0 I_L])

I_L = solve(jednadzbaLT,I_L);
I = ilaplace(I_L)
I = simplify(I)

fplot(I)
xlabel('t [s]');
ylabel('I [A]');
grid on;
```

Slika 5.23. Kod za rješavanje problema u RLC krugu.

```
I =

ilaplace(exp(s)/(s*exp(4*s) + s^2*exp(4*s) + s^3*exp(4*s)), s, t)
+ ilaplace(exp(3*s)/(s*exp(4*s)+ s^2*exp(4*s) + s^3*exp(4*s)), s, t)
- ilaplace(exp(4*s)/(s*exp(4*s) + s^2*exp(4*s) + s^3*exp(4*s)), s, t)
+ (8*3^(1/2)*exp(-t/2)*sin((3^(1/2)*t)/2))/3 - 4*ilaplace(s/(s*exp(4*s)
+ s^2*exp(4*s) + s^3*exp(4*s)), s, t) - ilaplace(1/(s*exp(4*s)
+ s^2*exp(4*s) + s^3*exp(4*s)), s, t)


I =

heaviside(t - 1) + heaviside(t - 3) - heaviside(t - 4)
+ exp(-t/2)*cos((3^(1/2)*t)/2)
- heaviside(t - 1)*cos(3^(1/2)*(t/2 - 1/2))*exp(1/2 - t/2)
+ heaviside(t - 4)*cos(3^(1/2)*(t/2 - 2))*exp(2 - t/2)
- heaviside(t - 3)*cos(3^(1/2)*(t/2 - 3/2))*exp(3/2 - t/2)
+ 3*3^(1/2)*exp(-t/2)*sin((3^(1/2)*t)/2)
- (3^(1/2)*heaviside(t - 1)*sin(3^(1/2)*(t/2 - 1/2))*exp(1/2 - t/2))/3
- (7*3^(1/2)*heaviside(t - 4)*sin(3^(1/2)*(t/2 - 2))*exp(2 - t/2))/3
- (3^(1/2)*heaviside(t - 3)*sin(3^(1/2)*(t/2 - 3/2))*exp(3/2 - t/2))/3 - 1
```

Slika 5.24. Rješenje izračuna struje RLC kruga.



Slika 5.25. Grafički prikaz struje RLC kruga.

Rješenje je u MATLAB - u zapisano u malo drugačijem obliku. Softver ne dopušta zapisati $\sqrt{3}$ u klasičnom obliku nego ga je potrebno zapisivati sa potencijama. Rezultati dobiveni izračunom 'na ruke' i pomoću softvera MATLAB se apsolutno poklapaju. Još se može uočiti korištenje naredbe 'Simplify()' koja je višestruko pojednostavila rješenje dobiveno nakon primjene inverzne Laplaceove transformacije.

U odnosu na prethodna dva primjera, razlika je u ulaznom naponu koji je zadan pomoću step funkcije. Također, razlika je uočljiva i u struji kruga. Kod prethodna dva primjera se sve svodilo na jednostavniji oblik jednadžbe, a kod RLC kruga imamo kompleksniji izraz za struju koji bez naredbe 'Simplify()' ne bi bilo moguće nacrtati.

6. Zaključak

U ovom radu cilj je bio opisati kako se tehnika Laplaceovih transformacija integrira unutar programskog sustava MATLAB, pri čemu se objasnio povijesni razvoj MATLAB – a, njegova današnja primjenu i mogućnosti koje nam pruža.

Laplaceova transformacija objašnjena je i s teorijskog stanovišta, objašnjen je njen povijesni razvoj, svojstva i primjena.

U praktičnom dijelu rada prikazano je nekoliko primjera transformacije i inverzne transformacije uz analizu trajanja izvršavanja programskog koda. Također su riješeni i primjeri koji su se temeljili na rješavanju problema izračuna struje RC, RL i RLC strujnog kruga.

Može se zaključiti kako je MATLAB moćan softver koji danas ima zaista široku primjenu u svim područjima znanosti, a posebice u elektrotehnici i bez problema su riješili svi postavljeni praktični problemi. Ipak kao problematiku treba istaknuti cijenu softvera koja je prilično visoka.

Što se tiče rada sa MATLAB – om općenito, nije bilo nekih neočekivanih i ne rješivih problema. Istaknimo samo problem kompleksnog prikaza rješenja kod RLC, ali se pomoću naredbe *simplify()* i taj problem mogao riješiti. Vezano uz vremensko trajanje izvršavanja naredbi ne može se reći da je i u tom aspektu bilo nekih problema, odnosno sve su se naredbe izvršile u očekivanom vremenskom okviru. Također treba istaknuti da je i grafički prikaz dobivenih rješenja bio iznimno jednostavan za implementaciju.

Na kraju možemo reći kako se MATLAB kao programski sustav za rješavanje problema vezanih uz Laplaceove transformacije pokazao kao dobar izbor s obzirom na jednostavnost i brzinu izvođenja i kao njegovu jedinu manu možemo istaknuti samo visoku cijenu licence.

Literatura

[1] Moler, C.: "A Brief History of MATLAB", s Interneta, https://www.mathworks.com/company/newsletters/articles/a-brief-history-of-matlab.html, 22. ožujka 2022.

[2] Whitrow, G. J.: "Pierre-Simon, marquis de Laplace", s Interneta, https://www.britannica.com/biography/Pierre-Simon-marquis-de-Laplace, 22. ožujka 2022.

[3] Schiff, J.L.: "The Laplace Transform: Theory and Applications", Cambridge University Press, 1988.

[4] s Interneta, https://en.wikipedia.org/wiki/Laplace_transform, 22 ožujka 2022.

[5] Stojković, N.; Naglić, V.; Mijat. N.: "Teorija mreža i linija", Sveučilište u Rijeci, Tehnički fakultet, 2005.

[6] Prevolšek, R.; Marovt, U.: "What is a smart home?", s Interneta, https://www.1home.io/blog/what-is-a-smart-home/, 23. ožujka.

[7] Boyer, C.B.: "Leonhard Euler Swiss mathematician", s Interneta, https://www.britannica.com/biography/Leonhard-Euler, 24. ožujka 2022.

[8] Hoschs, W.L.: "Niklaus Emil Wirth Swiss computer scientist", s Interneta, https://www.britannica.com/biography/Niklaus-Emil-Wirth, 10. travnja 2022.

[9] "Symbolic Objects to Represent Mathematical Objects", s Interneta, https://www.mathworks.com/help/symbolic/symbolic-objects-represent-mathematical-objects.html, 15. travnja 2022.

[10] "Matlab Online", s Interneta, https://www.mathworks.com/products/matlab-online.html, 20. travnja 2022.

[11] Dražić, I.: "Laplaceova transformacija - skripta", interna skripta, Sveučilište u Rijeci, Tehnički fakultet, 2017.

[12] Galović, S.; Srdelić, M.: "Fizika 2", s Interneta, https://edutorij.e-skole.hr/share/proxy/alfresco-noauth/edutorij/api/proxy-guest/452e1469-e362-4711-abcb-6f535c3b5254/html/7589_Izmjenicna_struja.html, 31. svibnja 2022.

[13] "Solve Differential Equations Using Laplace Transform", s Interneta, https://www.mathworks.com/help/symbolic/solve-differential-equations-using-laplace-transform.html., 10. lipnja 2022.

Sažetak i ključne riječi

U ovom radu prikazana je implementacija Laplaceovih transformacija u programskom sustavu MATLAB, s posebnim naglaskom na analizu električnih krugova, odnosno rješavanja pripadnih diferencijalno-integralnih jednadžbi. Ukratko je opisan povijesni razvoj MATLAB - a, njegovi alati, biblioteke te sama primjena. Također, opisan je povijesni razvoj Laplaceove transformacije, njezina svojstva i primjena u praksi.

Ključne riječi: MATLAB, Laplaceova transformacija, inverzna Laplaceova transformacija, analiza električnih krugova.

Summary and key words

This paper presents the implementation of Laplace transforms in the MATLAB programming system, with particular emphasis on the analysis of electrical circuits, i.e., the solution of the associated differential integral equations. The historical development of MATLAB is briefly described, its tools, libraries, and the application itself. The historical development of the Laplace transform, its properties and its application in practice are also described.

Keywords: MATLAB, Laplace transform, inverse Laplace transform, analysis of electrical circuits.