

Implementacija Laplaceovih transformacija u programskom jeziku Python

Stojković, Luka

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://um.nsk.hr/um:nbn:hr:190:568821>

Rights / Prava: [Attribution 4.0 International](#)/[Imenovanje 4.0 međunarodna](#)

Download date / Datum preuzimanja: **2025-04-01**



Repository / Repozitorij:

[Repository of the University of Rijeka, Faculty of Engineering](#)



SVEUČILIŠTE U RIJECI

TEHNIČKI FAKULTET

Preddiplomski sveučilišni studij elektrotehnike

Završni rad

**Implementacija Laplaceovih transformacija u programskom jeziku
Python**

Rijeka, rujan 2022.

Luka Stojković
0069080921

SVEUČILIŠTE U RIJECI

TEHNIČKI FAKULTET

Preddiplomski sveučilišni studij elektrotehnike

Završni rad

**Implementacija Laplaceovih transformacija u programskom jeziku
Python**

Mentor: doc. dr. sc. Ivan Dražić

Komentor: prof. dr. sc. Viktor Sučić

Rijeka, rujan 2022.

Luka Stojković
0069080921

Rijeka, 4. ožujka 2021.

Zavod: **Zavod za matematiku fiziku, strane jezike i kineziologiju**
Predmet: **Inženjerska matematika ET**
Grana: **2.03.06 automatizacija i robotika**

ZADATAK ZA ZAVRŠNI RAD

Pristupnik: **Luka Stojković (0069080921)**
Studij: **Preddiplomski sveučilišni studij elektrotehnike**

Zadatak: **Implementacija Laplaceovih transformacija u programskom jeziku Python //
Implementation of Laplace transformations in the Python programming
language**

Opis zadatka:


U radu je potrebno dati kratki uvod u programski jezik Python 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 programskom jeziku Python, pri čemu je potrebno riješiti nekoliko povezanih primjera. Kod svih primjera potrebno je napraviti analizu pripadnih algoritama s obzirom na robusnost, preciznost i vrijeme izvođenja.

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



Zadatak uručen pristupniku: 15. ožujka 2021.

Mentor:



Doc. dr. sc. Ivan Dražić

Predsjednik povjerenstva za
završni ispit:



Prof. dr. sc. Viktor Sučić

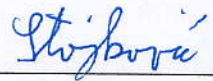


Prof. dr. sc. Viktor Sučić (komentor)

IZJAVA

Sukladno članku 8. Pravilnika o završnom radu, završnom ispitu i završetku preddiplomskih sveučilišnih studija/stručnih studija Tehničkog fakulteta Sveučilišta u Rijeci od 1. veljače 2020., izjavljujem da sam samostalno izradio završni rad prema zadatku preuzetom dana 15. ožujka 2021.

Rijeka, 19. rujna 2022.



Luka Stojković

Želio bih se zahvaliti svojoj obitelji na velikoj podršci kroz cijeli studij. Također, želio bih se zahvaliti doc. dr. sc. Ivanu Dražiću koji me je usmjeravao u pisanju ovoga rada te mi uvelike pomogao da završim ovo poglavlje svoga života.

Sadržaj

1. Uvod	3
2. Python	4
2.1. Guido van Rossum	4
2.2. Uvod u Python	4
2.3. Izvođenje programa u Pythonu	5
2.4. Varijable	6
2.5. Naredbe unosa i ispisa	6
2.6. Naredbe za grananje	7
2.7. Petlje	8
2.7.1. <i>While</i> petlja	8
2.7.2. <i>For</i> petlja	9
2.8. Moduli	10
2.8.1. Uvod u module	10
2.8.2. <i>Math</i> modul	10
2.8.3. <i>SymPy</i> modul	11
3. Laplaceova transformacija	12
3.1. Pierre-Simon Laplace	12
3.2. Laplaceova transformacija	13
3.3. Egzistencija Laplaceove transformacije	13
3.4. Svojstva Laplaceove transformacije	14
3.5. Laplaceova transformacija periodičke funkcije	19
3.6. Laplaceova transformacija konvolucije	20
3.7. Tablica Laplaceovih transformacija	21
4. Rješavanje električkih krugova pomoću Laplaceove transformacije	22
4.1. Serijska <i>RL</i> mreža	22
4.2. Serijska <i>RC</i> mreža	25
4.3. Serijska <i>RLC</i> mreža	27
5. Laplaceova transformacija u Pythonu	32
5.1. <i>SymPy</i>	32

5.2. Izračunavanje Laplaceovih transformacija unutar Pythona	32
5.3. Rješavanje električnih krugova unutar Pythona	35
5.3.1. Serijska <i>RL</i> mreža	35
5.3.2. Serijska <i>RC</i> mreža	38
5.3.3. Serijska <i>RLC</i> mreža	39
6. Zaključak	41
Literatura	42
Sažetak i ključne riječi	43
Summary and key words	44

1. Uvod

Laplaceova transformacija integralna je transformacija koja u današnje vrijeme ima široku primjenu u elektrotehnici, ponajviše se koristi prilikom rješavanja diferencijalnih i integralnih jednadžbi. Kada pogledamo u matematičkom smislu, iako po ideji relativno jednostavna transformacija, za rješavanje kompliciranijih inženjerskih zadataka potrebno je uložiti mnogo vremena i truda. Upravo iz tog razloga, cilj ovoga rada je pokazati kako možemo rješavati diferencijalne jednadžbe preko Laplaceovih transformacija unutar programskog jezika Pythona.

Prvi dio rada upoznat će nas s programskim jezikom Python te različitim naredbama koje možemo koristiti unutar njega. Uz to, upoznat ćemo se s modulima unutar Pythonove besplatne biblioteke koji će kasnije biti potrebni za rješavanje matematičkih zadataka.

U drugom dijelu rada detaljnije ćemo se upoznati s Laplaceovom transformacijom, njenom egzistencijom i svojstvima i u završnom dijelu poglavlja bit će riješeni primjeri iz struke koristeći upravo Laplaceovu transformaciju.

Završni dio pokazat će kako pozvati Laplaceovu transformaciju, inverznu Laplaceovu transformaciju unutar Pythona te kako pomoću Pythona dobiti graf rješenja. Uz to prikazat će se kôd za rješavanje primjera zadataka koje smo riješili u drugom dijelu rada i rješenja koje će Python generirati.

2. Python

2.1. Guido van Rossum

Python je jedan od najpopularnijih programskih jezika današnjice. Autor jezika je Guido van Rossum, a prva verzija jezika objavljena je 1990. godine. Guido van Rossum rođen je 1956. godine u Nizozemskoj. Završio je matematički i informatički fakultet u Amsterdamu. Dok je radio u Centrum Wiskunde and Informatica (CWI), Nizozemskom istraživačkom centru za matematiku i informatiku pomogao je i u stvaranju ABC programskog jezika¹. Van Rossum se zalagao da Python bude jednostavan i intuitivan jezik, ali i dalje jednako dominantan u programskom svijetu kao i drugi jezici. Nadalje, Pythonov izvorni kôd napravio je besplatnim tako da ga svi mogu koristiti i mijenjati za razvoj nekog svog programa. Python pripada skupini skriptnih i objektno orijentiranih jezika. Zbog iznimne jednostavnosti i praktičnosti brzo se razvijao te je već do 1998. godine imao bazu od preko 300.000 korisnika, a početkom 21. stoljeća počele su ga koristiti i kompanije poput Google-a, MIT-a, NASA-e, Yahoo-a, itd. [8]



Slika 2.1. Guido van Rossum, izvor: [1]

2.2. Uvod u Python

Python nije jezik koji je promijenio programiranje ili donio neke nove značajne ideje, već je ujediniio najbolje ideje i principe rada drugih programskih jezika. Nakon početnih inačica, 2000. godine izdan je Python 2.0 čija je završna podverzija 2.7 objavljena 2010. godine. Paralelno s inačicom 2, krajem 2008. godine objavljen je Python 3.0. koji je unio značajne promjene u sintaksi jezika, te se zbog toga u njemu često ne mogu pokretati programi pisani za starije inačice. Za svrhu lakšeg prelaska iz jedne u drugu inačicu razvijen je alat 2to3 koji direktno prevodi kod pisan u inačici 2.x u kod za inačicu 3.x. Python se koristi za razne svrhe u računarstvu poput jezika

¹ABC programski jezik je dizajniran za uvod u svijet programiranja te učenje osnova programiranja

za pisanje skripti ili za razvoj najzahtijevnijih aplikacija, ali ima i svoje primjene u elektrotehnici za pojednostavljenje računanja nekih matematičkih transformacija poput Laplaceove ili Fourierove transformacije. Python ima jednostavnu i jasnu sintaksu i dosljednost, koja ga čini izuzetno lakim za učenje. Također, ta specifična sintaksa potiče na urednost pri pisanju koda, pa je kod veoma čitljiv i pregledan.

2.3. Izvođenje programa u Pythonu

Python nudi nekoliko različitih načina za pokretanje naredbi ili programa i ima vrlo praktično sučelje koje vam omogućuje da odmah upisujete i izvršavate naredbe. Interaktivni način rada izvrstan je za učenje jezika i eksperimentiranje s različitim jezičnim konstruktima i bibliotekama, kao i testiranje modula za izradu složenijih programa. On omogućava da napišemo naredbu koja se odmah izvršava i ispisuje rezultat.

Prikaz koda u interaktivnom načinu rada:

```
>>> print("Ovo je početak završnog rada!")
      Ovo je početak završnog rada!
```

Međutim za pisanje nekih duljih i kompliciranijih kodova bit će potrebno pohraniti taj kod u datoteku. Programske datoteke najčešće imenujemo s nastavkom .py, iako to nije nužno za pokretanje datoteke. Taj nastavak je neophodan jedino ako se želi moći učitati i pokrenuti tu datoteku iz drugog programa. Na primjer, može se napisati sljedeća skripta i spremiti pod imenom prvi.py.

Primjer pisanja skripte:

Ulaz:

```
print(2*9)
tekst="Ulančavanje teksta"
print(tekst+"u Pythonu")
```

Izlaz:

18

Ulančavanje teksta u Pythonu

2.4. Varijable

Varijable su memorijske lokacije koje služe za pohranu vrijednosti. To bi značilo da se prilikom kreiranja nove varijable u memoriji rezervira određeni dio prostora za pohranu. Ovisno o tipu podataka, raspodjeljuje se manji ili veći dio memorije i to omogućava spremanje različitih tipova podataka. Neki osnovni podatci su: brojevi, znakovi i polja znakova, ali naravno Python može raditi i s kompleksnijim tipovima podataka. Važno je napomenuti da je sve u Pythonu objekt, odnosno tip podataka je predstavljen kao objekt određenog tipa podataka. Vrste varijabli u Pythonu ne moraju biti eksplicitno zapisane jer se deklaracije događaju automatski kada upišete vrijednost u varijablu. Znak jednakosti (=) koristi se za povezivanje vrijednosti s varijablom. Lijeva strana znaka jednakosti je varijabla, a desna je njezina vrijednost.

Pridruživanje vrijednosti varijablama:

```
Prirodni_broj=6
```

```
Udaljenost_u_metrima=560
```

```
Ime_korisnika="Matija"
```

```
Ime_studenta, kolegij="Luka", "Elektronika"
```

2.5. Naredbe unosa i ispisa

Da bi se moglo manipulirati podacima i izvoditi različite operacije na njima, potrebno je osigurati podatke na kojima se mogu izvoditi različite operacije. Kako bi se omogućilo pisanje teksta ili numeričkih vrijednosti u programskom jeziku Python, potrebno je koristiti naredbe za unos. Sam unos naredbe dovoljan, već ga se mora povezati s varijablom kao jednostavan broj ili tekstualnu vrijednost. Obično dodjeljivanjem vrijednosti varijabli (npr.: $a = 17$), vrijednost varijable a se određuje unutar skripte programskog jezika. Kako bi se korisniku omogućio unos podataka u sučelje Python ljuske putem tipkovnice za izvođenje programa, potrebna je naredba za unos. Kao i prije, korištenjem asocijativnog simbola "=" za pridruživanje vrijednosti varijabli, naredba za unos i njezina sintaksa sada će biti pridružene varijabli a . Budući da u Pythonu nema unaprijed definiranih tipova podataka, on svaki unos tumači kao niz, a ne broj. Stoga je prije unosa naredbe potrebno odrediti tip podataka (int, float) broja s kojim se rukuje.

Međutim, naredba koja ispisuje vrijednost ne predstavlja značajno odstupanje od sintakse naredbe unosa. Naredba za ispis aktivira se ključnom riječi *print*. Nakon te riječi otvaraju se i zatvaraju zagrade. Unutar zagrada može se ispisati tekst, kao i vrijednost varijable. Glavna razlika je u tome što je tekst uvijek u navodnicima. On signalizira Pythonu da je to poruka koju korisnik želi ispisati. Vrijednost varijable ispisuje se na način da se varijabla u naredbi za ispis

pozove tako da su joj slova ispisana na mjestu. Sve tekstualne vrijednosti unutar naredbe za ispis, kao i vrijednosti varijable, mogu se odvojiti zarezima.

Primjer korištenja naredbi za unos podataka:

```
>>> a=int(input("unesi cijeli broj"))
      unesi cijeli broj 17
>>> a
17

>>> b=float(input("unesi decimalni broj"))
      unesi decimalni broj 40.25
>>> b
40.25

>>> c=input("unesi ime")
      unesi ime Filip
>>> c
Filip
```

Primjer korištenja naredbi za ispis podataka:

```
>>> print("Print je naredba za ispisivanje podataka,
      a uz to možemo dodavati i brojeve kao na primjer", 27)
Print je naredba za ispisivanje podataka,
      a uz to možemo dodavati i brojeve kao na primjer 27
```

2.6. Naredbe za grananje

Naredba *if* pripada kontroli toka za odluke. U najjednostavnijem obliku ova struktura sastoji se od zaglavlja odluke i tijela odluke. Zaglavlje odluke sastoji se od ključne riječi *if* i uvjeta, dok se tijelo odluke sastoji od jedne ili više naredbi koje rade neku korisnu akciju. Kada je uvjet zadovoljen, izvršava se tijelo odluke, a ukoliko uvjet nije zadovoljen, tijelo odluke neće se izvršiti. Sintaksa *if* odluke zahtijeva da se napiše ključna riječ *if* te poslije nje logički izraz koji će uvijek imati ili *True* ili *False* vrijednost. Ključne riječi su već unaprijed rezervirane riječi u Pythonu i iza njih stoje neke naredbe. Nakon logičkog izraza obavezno je staviti dvotočku kako bi Python prepoznao da je to kraj logičkog izraza i da slijedi tijelo odluke. Uz naredbu *if* često susrećemo i naredbu *else* koja upotpunjuje naredbu *if*. Te dvije naredbe zajedno čine *if-else* odluku u kojoj će uvijek jedan dio funkcije biti izvršen, tj. ako je uvjet zadovoljen izvršiti će se *if* odluka, a ukoliko uvjet nije zadovoljen izvršiti će se *else* odluka. Također, može se dogoditi slučaj da nam treba više

else funkcija i tada ćemo koristiti funkciju *elif*. Ona služi za neograničeni broj uvjeta. Bitno je naglasiti da kod *elif* funkcije tijelo funkcije će se izvršiti samo kod onog uvjeta koji će biti prvi zadovoljen, a nakon toga ostali uvjeti se ne provjeravaju. U slučaju kada niti jedan od uvjeta nije zadovoljen izvršiti će se tijelo funkcije koje pripada *else* dijelu *if-elif-else* sintakse.

Primjer korištenja if-else funkcija:

Ulaz:

```
x=7
if x<10:
    print("uvjet je zadovoljen!")
else:
    print print("uvjet nije zadovoljen!")
```

izlaz:

Uvjet je zadovoljen!

Primjer korištenja if-elif-else funkcija:

Ulaz:

```
x=16
if x==5:
    print("x=5")
elif x==16:
    print("x=16")
else:
    print("niti jedan uvjet nije ispravan!")
```

izlaz:

x=16

2.7. Petlje

2.7.1. While petlja

Ova petlja prije izvršavanja programskog kôda koji joj pripada, ispituje je li uvjet istinit ili ne. Petlja funkcionira na način da ukoliko je uvjet istinit tada će se izvršiti tijelo petlje, a ako uvjet nije istinit tijelo petlje se preskače te se izvršava programski kod napisan ispod petlje. Sintaksa *while*

petlje sastoji se od ključne riječi *while* te poslije nje logičkog izraza i na kraju dvotočke koja odvaja logički izraz i tijelo petlje. Kod *while* petlje mora se paziti da se ne napravi beskonačna petlja. Ona se dogodi kada je uvjet provjere uvijek istinit i onda se *while* petlja izvršava beskonačno puta.

Primjer korištenja while petlje:

Ulaz:

```
x=0
while x<5:
    print("Dobrodošli na fakultet!")
    x=x+1
```

izlaz:

```
Dobrodošli na fakultet!
Dobrodošli na fakultet!
Dobrodošli na fakultet!
Dobrodošli na fakultet!
Dobrodošli na fakultet!
```

2.7.2. *For* petlja

U Pythonu *for* petlja koristi se na drugačiji način u odnosu na neke druge programske jezike. Kod Pythona *for* petlja nema niti početnu niti završnu vrijednost, a nema ni aritmetički faktor uvećanja. Ona se iterira kroz elemente zadane sekvence koja može biti niz znakova ili lista. *For* petlja se obično koristi uz funkciju *range* koja određuje koliko će se puta *for* petlja izvršiti. *Range* funkcija može imati više vrijednosti, ovisno o tome kako ju se definira.

Primjer korištenja for petlje:

Ulaz:

```
for i in range(1,6):
    print(i)
```

izlaz:

```
1
2
3
4
5
```

2.8. Moduli

2.8.1. Uvod u module

Osnovna ideja programske arhitekture u Pythonu je da se veći programi obično grade kao skup modula koji učitavaju alate definirane u drugim modulima. Operacije učitavanja modula pokreću naredba `import`. Između ostalog, time se pokreće izvršavanje naredbi modula, pa je to i jedan od načina pokretanja programa. Istodobno, treba se znati da se naredbe modula izvršavaju samo tijekom prvog učitavanja. Naknadna ponovna pokretanja naredbe uvoza neće imati učinka. Čak i ako se šifra modula promijeni tijekom tog vremena, modul se neće ponovno učitati. Razlog ovakvog ponašanja u Pythonu je taj što je učitavanje modula skupa operacija. Potrebno je pronaći datoteku modula, pretvoriti je u bajt kod i izvršiti naredbu. Zanimljivo, iako je Python interpretirani jezik, on zapravo prevodi module u bajt kod kada se učitaju, koji se zatim izvršava na virtualnom računalu. Ovo se radi kako bi se kasnije brže izvršili neki dijelovi učitanoog modula. Prevedeni bajt-kôd pohranjuje se u datoteku s nastavkom `.pyc`. Ako se još uvijek želi ponovno učitati i izvršiti (moguće modificirani) modul, može se koristiti funkcija ponovnog učitavanja. U Python 2, `reload` je ugrađena funkcija, dok je u Python 3 u `imp` modulu. Samo moduli koji su uspješno učitani mogu se ponovno učitati. Također mora se imati na umu da je ponovno učitavanje funkcija dok je uvoz naredba. Svrha modula je organiziranje biblioteka i alata, a moduli zapravo predstavljaju prostor imena. Učitavanjem, modul postaje punopravni programski objekt u Pythonu čiji atributi odgovaraju imenima definiranim učitanim modulom. Program koji učitava modul ima pristup svim imenima dodijeljenim na najvišoj razini modula. Ova imena se obično dodjeljuju uslugama koje pružaju moduli: funkcije, klase, varijable, za korištenje u drugim programima. Ovim nazivima može se pristupiti pomoću naredbi `import` i `from` ili pozivanjem funkcije ponovnog učitavanja. Module i pakete koji dolaze sa standardnom instalacijom Pythona nazivamo standardnom bibliotekom.

2.8.2. *Math* modul

Kada je potreban neki paket koji ne dolazi uz standardnu instalaciju potrebno je otići na internet i tamo pronaći željeni paket te ih instalirati zasebno na računalo. Modul `math` je jedan od standardnih modula koji se poziva uz pomoć ključne riječi `import`. Pomoću njega mogu se koristiti osnovne matematičke funkcije, na primjer izračunavanje korijena nekog realnog pozitivnog broja, potencije realnih brojeva, vrijednosti kuteva pomoću trigonometrijskih funkcija, itd.

Primjer korištenja math modula:

Ulaz:

```
from math import *
print(sqrt(16))
print(pow(2,3))
print(sin(72))
```



```
print(cos(29))
```

izlaz:

```
4.0
```

```
8.0
```

```
0.25382336276203626
```

```
-0.7480575296890003
```

2.8.3. *SymPy* modul

Dok se *math* modul koristi za neke jednostavnije matematičke funkcije, ako se želi raditi sa kompliciranijim matematičkim funkcijama ili transformacijama trebati će *SymPy* modul. *SymPy* je besplatni modul napravljen kompletno u Pythonu i služi za lakše rješavanje matematičkih problema u njemu. Ovaj modul će se koristiti u nastavku ovog rada jer će se pomoću njega izračunavati Laplaceova transformacija u Pythonu.

3. Laplaceova transformacija

U ovom poglavlju bavimo se definicijom i svojstvima Laplaceove transformacije, a poglavlje je uglavnom obrađeno prema izvoru [3].

3.1. Pierre-Simon Laplace

Pierre-Simon Laplace rođen je 23. ožujka 1749. godine u mjestu Beaumont-en-Auge u Francuskoj. Bavio se matematikom i astronomijom te su njegovi radovi znatno pridonijeli razvoju matematike, astronomije i statistike. Počeo je studirati teologiju u Caenu gdje mu 2 učitelja matematike bude zanimanje za matematiku i Laplace prestaje studirati teologiju te kreće u Pariz kod tada već poznatog matematičara Jeana le Ronda d'Alemberta. Na njegovu preporuku Laplace kreće predavati matematiku u Visokoj vojnoj školi u Parizu 1771. godine. Obzirom da predavanja nisu bila zahtjevna za njega, Laplace se mogao skoro u potpunosti posvetiti svojim istraživanjima. U tom razdoblju od otprilike 17 godina rada u vojnoj školi napisao je većinu svojih radova iz astronomije i matematike. Kao njegov najvažniji rad smatra se Laplaceova transformacija o kojoj će se više pričati u nastavku. Član je Francuske Akademije znanosti od 1773. te se njegovo ime nalazi na listi 72 znanstvenika ugraviranih na Eifellovom tornju.



Slika 3.1. Pierre-Simon Laplace, izvor: [2]

3.2. Laplaceova transformacija

Navedimo najprije strogu matematičku definiciju Laplaceove transformacije.

Definicija 3.1. *Neka je f funkcija realnog argumenta t , definirana za $t > 0$ i s vrijednostima u skupu realnih ili kompleksnih brojeva. Neka je s realni ili kompleksni parametar. Laplaceov transformat funkcije f je funkcija F definirana s:*

$$F(s) = \int_0^{\infty} f(t)e^{-st} dt \quad (3.1)$$

za svaki s za koji ovaj nepravilni integral konvergira. Funkcija f naziva se original, a $F(s) = \mathcal{L}(f(t))$ Laplaceova slika.

Preslikavanje $f \rightarrow F$ nazivamo Laplaceova transformacija i označavamo ga s \mathcal{L} .

Za nastavak je bitno napomenuti da je kod Laplaceove transformacije t -domena realna, a s -domena kompleksna.

3.3. Egzistencija Laplaceove transformacije

Laplaceova transformacija neće se moći primijeniti na svaku funkciju. Postoje neki određeni uvjeti koje funkcija mora zadovoljavati kako bi pripadna Laplaceova transformacija postojala. Na početku mora se napomenuti da se Laplaceovom transformacijom većinom razmatra t -domena koja označava vrijeme, a pošto nam je donja granica integrala 0 prvi uvjet nam je:

$$f(t) = 0, \quad \forall t < 0. \quad (3.2)$$

Kako se već prije spomenulo definicijski integral Laplaceove transformacije je nepravilni i egzistencija Laplaceove transformacije vezana je uz egzistenciju tog nepravog integrala. U tom je kontekstu najprije potrebno osigurati integrabilnost podintegralne funkcije odakle slijedi drugi uvjet za postojanje Laplaceove transformacije:

$$f \in \tilde{C}[0, \infty]. \quad (3.3)$$

$\tilde{C}[0, \infty]$ nam predstavlja klasu po dijelovima neprekidnih funkcija na intervalu $[0, \infty]$ s najviše prebrojivo prekida što je dovoljan uvjet da bi funkcija bila integrabilna.

Još ostaje osigurati konvergenciju nepravog integrala, a za to ćemo se poslužiti sljedećim kriterijem koji navodimo bez dokaza.

Teorem 3.1 (Konvergencija nepravog integrala). *Neka su zadane funkcije f i g definirane na intervalu $[a, \infty)$ takve da vrijedi*

$$|f(t)| \leq g(t), \quad (3.4)$$

za sve $t \in [a, \infty)$. Tada konvergencija integrala

$$\int_a^{\infty} g(t) dt \quad (3.5)$$

povlači konvergenciju integrala

$$\int_a^{\infty} f(t) dt. \quad (3.6)$$

Ovaj kriterij motivira uvođenje posebne klase funkcija koje ćemo zvati funkcije eksponencijalnog rasta koje definiramo na sljedeći način:

Definicija 3.2. Za funkciju f kažemo da je funkcija eksponencijalnog rasta ako postoje konstante $M > 0$ i $a > 0$ takve da vrijedi

$$|f(t)| < Me^{at}, \quad (3.7)$$

za sve t iz domene funkcije f .

Sada možemo uvesti uvjete za postojanje Laplaceove transformacije.

Teorem 3.2 (Egzistencija Laplaceove transformacije). Laplaceova transformacija funkcije $f : [0, \infty) \rightarrow \mathbb{C}$ postoji ako su ispunjeni sljedeći uvjeti:

1. Funkcija f je po dijelovima neprekidna s najviše prebrojivo prekida prve vrste.
2. Funkcija f je funkcija eksponencijalnog rasta.

3.4. Svojstva Laplaceove transformacije

Sada ćemo navesti neka najvažnija svojstva Laplaceove transformacije.

Teorem 3.3 (Linearnost Laplaceove transformacije). Neka su zadane funkcije $f, g : \mathbb{R} \rightarrow \mathbb{C}$ te neka je \mathcal{L} Laplaceova transformacija. Vrijedi:

$$\mathcal{L}\{\alpha f(t) + \beta g(t)\} = \alpha \mathcal{L}\{f(t)\} + \beta \mathcal{L}\{g(t)\}, \quad (3.8)$$

pri čemu su $\alpha, \beta \in \mathbb{C}$ proizvoljne konstante.

Teorem 3.4 (Teorem o slici derivacije). Ako je $\mathcal{L}\{f(t)\} = F(s)$, tada je

$$\mathcal{L}\{f'(t)\} = -f(0) + sF(s). \quad (3.9)$$

Dokaz. Iz definicije Laplaceove transformacije slijedi

$$\mathcal{L}\{f'(t)\} = \int_0^{\infty} f'(t)e^{-st} dt. \quad (3.10)$$

Sada ćemo primijeniti formulu parcijalne integracije uzimavši

$$u = e^{-st}, \quad dv = f'(t)dt \quad (3.11)$$

odnosno

$$du = -se^{-st}dt, \quad v = f(t). \quad (3.12)$$

Dobivamo:

$$\mathcal{L}\{f'(t)\} = \underbrace{e^{-st}f(t)}_{=-f(0)} \Big|_0^\infty + s \int_0^\infty e^{-st}f(t)dt = -f(0) + sF(s), \quad (3.13)$$

čime je tvrdnja dokazana. □

Teorem 3.5 (Teorem o slici derivacije n -tog reda). *Ako je $\mathcal{L}\{f(t)\} = F(s)$, tada je*

$$\mathcal{L}\{f^{(n)}(t)\} = s^n F(s) - \sum_{k=0}^{n-1} s^{n-k-1} f^{(k)}(0). \quad (3.14)$$

Dokaz. Dokaz provodimo metodom matematičke indukcije.

Baza indukcije ($n = 1$) Dokazujemo da tvrdnja vrijedi za $n = 1$. U tom slučaju tvrdnja glasi

$$\mathcal{L}\{f'(t)\} = sF(s) - f(0) \quad (3.15)$$

što je tvrdnja teorema o slici derivacije pa je ova faza dokaza gotova.

Pretpostavka indukcije ($n = m$) Pretpostavljamo da tvrdnja vrijedi za $n = m$, pa imamo:

$$\mathcal{L}\{f^{(m)}(t)\} = s^m F(s) - \sum_{k=0}^{m-1} s^{m-k-1} f^{(k)}(0). \quad (3.16)$$

Korak indukcije ($n = m + 1$) Uvažavajući pretpostavku indukcije potrebno je dokazati da tvrdnja vrijedi i za $n = m + 1$. Koristeći se teoremom o slici derivacije dobivamo:

$$\mathcal{L}\{f^{(m+1)}(t)\} = \mathcal{L}\{(f^{(m)})'(t)\} = s\mathcal{L}\{f^{(m)}(t)\} - f^{(m)}(0). \quad (3.17)$$

Uvrštavanjem izraza iz pretpostavke indukcije slijedi:

$$\begin{aligned} \mathcal{L}\{f^{(m+1)}(t)\} &= \mathcal{L}\{(f^{(m)})'(t)\} = s \left(s^m F(s) - \sum_{k=0}^{m-1} s^{m-k-1} f^{(k)}(0) \right) \\ &\quad - f^{(m)}(0) = s^{m+1} F(s) - \sum_{k=0}^{m-1} s^{m-k} f^{(k)}(0) - f^{(m)}(0) \\ &= s^{m+1} F(s) - \sum_{k=0}^m s^{m-k} f^{(k)}(0), \end{aligned} \quad (3.18)$$

čime je tvrdnja dokazana.

□

Teorem 3.6 (Teorem o derivaciji slike). *Ako je $\mathcal{L}\{f(t)\} = F(s)$, tada je*

$$\mathcal{L}\{t^n f(t)\} = (-1)^n F^{(n)}(s), \quad n \in \mathbb{N}. \quad (3.19)$$

Dokaz. Deriviramo li funkciju

$$F(s) = \int_0^{\infty} f(t)e^{-st} dt \quad (3.20)$$

dobivamo

$$F'(s) = - \int_0^{\infty} t f(t) e^{-st} dt = -\mathcal{L}\{t f(t)\}. \quad (3.21)$$

Daljnim deriviranjem ovog izraza odmah dobivamo traženu tvrdnju. □

Ovo svojstvo se koristi kod određivanja Laplaceove transformacije potencija, tj. funkcija oblika $t \mapsto t^n$.

Teorem 3.7 (Teorem o prigušenju (pomaku u slici)). *Neka je $\mathcal{L}\{f(t)\} = F(s)$ i $a \in \mathbb{R}$. Tada je*

$$\mathcal{L}\{e^{at} f(t)\} = F(s - a). \quad (3.22)$$

Dokaz. Iz definicije Laplaceove transformacije dobivamo

$$\mathcal{L}\{e^{at} f(t)\} = \int_0^{\infty} e^{at} f(t) e^{-st} dt = \int_0^{\infty} f(t) e^{(a-s)t} dt = \int_0^{\infty} f(t) e^{-(s-a)t} dt = F(s - a), \quad (3.23)$$

čime je dokaz završen. □

Teorem 3.8 (Teorem o slici integrala). *Ako je $\mathcal{L}\{f(t)\} = F(s)$, tada je*

$$\mathcal{L}\left\{\int_0^t f(\tau) d\tau\right\} = \frac{F(s)}{s}. \quad (3.24)$$

Dokaz. Iz definicije Laplaceove transformacije slijedi

$$\mathcal{L}\left\{\int_0^t f(\tau) d\tau\right\} = \int_0^{\infty} \left(\int_0^t f(\tau) d\tau\right) e^{-st} dt. \quad (3.25)$$

Sada ćemo primijeniti formulu parcijalne integracije uzimajući

$$u = \int_0^t f(\tau) d\tau, \quad dv = e^{-st} dt \quad (3.26)$$

odnosno

$$du = f(t)dt, \quad v = -\frac{1}{s}e^{-st}. \quad (3.27)$$

Dobivamo:

$$\mathcal{L}\left\{\int_0^t f(\tau)d\tau\right\} = \underbrace{-\frac{1}{s}e^{-st}\int_0^t f(\tau)d\tau}\bigg|_0^\infty + \frac{1}{s}\int_0^\infty e^{-st}f(t)dt = \frac{F(s)}{s}, \quad (3.28)$$

čime je tvrdnja dokazana. \square

Teorem 3.9 (Teorem o integralu slike). *Ako je $\mathcal{L}\{f(t)\} = F(s)$, tada je*

$$\mathcal{L}\left\{\frac{f(t)}{t}\right\} = \int_s^\infty F(s)ds \quad (3.29)$$

uz uvjet da integral s desne strane jednakosti konvergira.

Dokaz. Iz definicije Laplaceove transformacije dobivamo:

$$\int_s^\infty F(s)ds = \int_s^\infty \int_0^\infty f(t)e^{-st}dtds. \quad (3.30)$$

Nakon promjene poretka integracije slijedi:

$$\begin{aligned} \int_s^\infty F(s)ds &= \int_0^\infty \left(\int_s^\infty f(t)e^{-st}ds\right) dt = \\ \int_0^\infty f(t) \left(\underbrace{\int_s^\infty e^{-st}ds}_{=\frac{1}{t}e^{-st}}\right) dt &= \int_0^\infty e^{-st}\frac{f(t)}{t}dt = \mathcal{L}\left\{\frac{f(t)}{t}\right\}, \end{aligned} \quad (3.31)$$

čime je tvrdnja dokazana. \square

Teorem 3.10 (Teorem sličnosti). *Neka je f funkcija koja zadovoljava uvjete za postojanje Laplaceove transformacije. Tada za realnu konstantu $\omega > 0$ vrijedi*

$$\mathcal{L}\{f(\omega t)\} = \frac{1}{\omega}F\left(\frac{s}{\omega}\right). \quad (3.32)$$

Dokaz. Po definiciji Laplaceove transformacije imamo

$$\mathcal{L}\{f(\omega t)\} = \int_0^\infty f(\omega t)e^{-st}dt. \quad (3.33)$$

Uvođenjem supstitucije $\omega t = u$, tj. $\omega dt = du$ dobivamo

$$\mathcal{L}\{f(\omega t)\} = \int_0^\infty f(u)e^{-\frac{su}{\omega}}\frac{du}{\omega} = \frac{1}{\omega}\int_0^\infty f(u)e^{-u\frac{s}{\omega}}du = \frac{1}{\omega}F\left(\frac{s}{\omega}\right). \quad (3.34)$$

\square

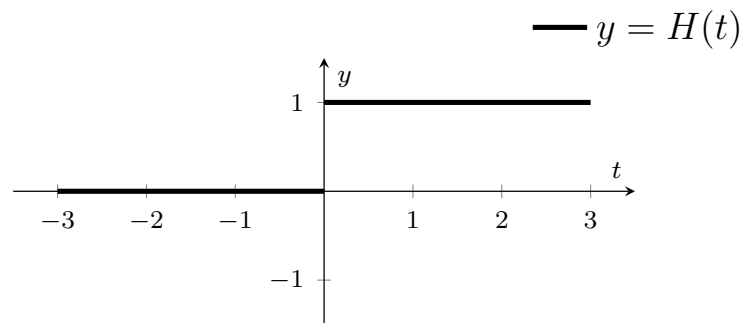
U svojstvu koje slijedi koristi se Heavisideova funkcija koju ćemo najprije definirati.

Definicija 3.3. Funkciju oblika

$$u(t) = H(t) = \begin{cases} 1, & t \geq 0 \\ 0, & t < 0 \end{cases} \quad (3.35)$$

nazivamo **Heavisideova (step) funkcija**.

Heavisideova funkcija prikazana je na slici 3.2.



Slika 3.2. Grafički prikaz Heavisideove funkcije

Teorem 3.11 (Teorem o pomaku originala). *Ako je $\mathcal{L}\{f(t)\} = F(s)$, a $u(t)$ Heavisideova funkcija, tada je*

$$\mathcal{L}\{f(t-a)u(t-a)\} = e^{-as}F(s). \quad (3.36)$$

Dokaz. Prema definiciji Laplaceove transformacije i definiciji Heavisideove funkcije imamo

$$\mathcal{L}\{f(t-a)u(t-a)\} = \int_0^{\infty} f(t-a)u(t-a)e^{-st}dt = \int_a^{\infty} f(t-a)e^{-st}dt. \quad (3.37)$$

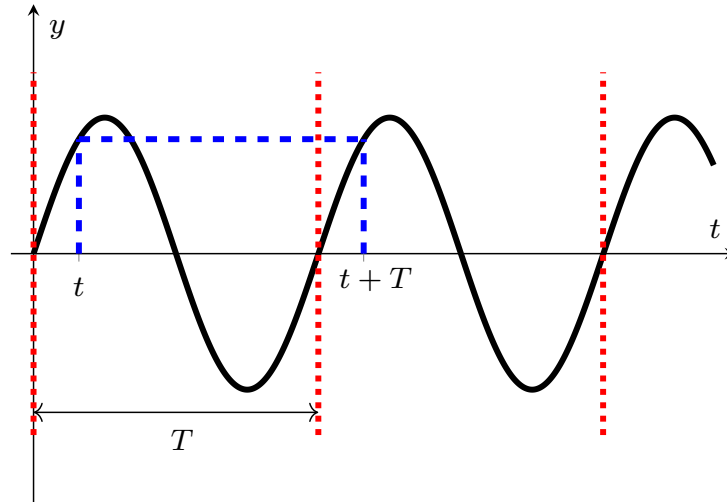
Supstitucijom $t-a = u$, tj. $dt = du$ dobivamo:

$$\begin{aligned} \mathcal{L}\{f(t-a)u(t-a)\} &= \int_0^{\infty} f(u)e^{-s(u+a)}du \\ &= \int_0^{\infty} f(u)e^{-su}e^{-as}du = e^{-as} \int_0^{\infty} f(u)e^{-su}du = e^{-as}F(s). \end{aligned} \quad (3.38)$$

□

3.5. Laplaceova transformacija periodičke funkcije

Periodička funkcija je funkcija koja ima svojstvo $f(t + T) = f(t)$ za sve vrijednosti t iz svoje domene. Jedan primjer takve funkcije prikazan je na slici 3.3.



Slika 3.3. Grafički prikaz periodičke funkcije

Postupak nalaženja Laplaceove transformacije ovakvih funkcija dan je u sljedećem teoremu.

Teorem 3.12 (Laplaceova transformacija periodičke funkcije). *Neka je f periodička funkcija s periodom T , tj. funkcija za koju vrijedi $f(t + T) = f(t)$. Tada je*

$$\mathcal{L}\{f(t)\} = F(s) = \frac{1}{1 - e^{-Ts}} \int_0^T f(t)e^{-st} dt. \quad (3.39)$$

Dokaz. Iz definicije Laplaceove transformacije dobivamo:

$$F(s) = \int_0^{\infty} f(t)e^{-st} dt = \int_0^T f(t)e^{-st} dt + \int_T^{\infty} f(t)e^{-st} dt. \quad (3.40)$$

Uzmemo li u drugom integralu supstituciju $t = u + T$, odnosno $dt = du$ dobivamo

$$F(s) = \int_0^T f(t)e^{-st} dt + \int_0^{\infty} f(u + T)e^{-s(u+T)} du. \quad (3.41)$$

Iskoristimo li svojstvo periodičnosti funkcije f iz ovog izraza dobivamo

$$F(s) = \int_0^T f(t)e^{-st} dt + e^{-sT} \underbrace{\int_0^{\infty} f(u)e^{-su} du}_{F(s)}, \quad (3.42)$$

odakle slijedi

$$F(s) - e^{-sT}F(s) = \int_0^T f(t)e^{-st}dt, \quad (3.43)$$

što direktno povlači tvrdnju teorema. \square

3.6. Laplaceova transformacija konvolucije

Definicija 3.4. Neka su f i g dvije integrabilne funkcije. Njihova **konvolucija**, u oznaci $f * g$, definirana je s:

$$(f * g)(x) = \int_{-\infty}^{\infty} f(y)g(x - y)dy. \quad (3.44)$$

Budući da kod Laplaceovih transformacija promatramo funkcije koje su aktivne samo na području $t > 0$ možemo pojednostaviti definiciju konvolucije sa:

$$(f * g)(t) = \int_0^{\infty} f(\tau)g(t - \tau)d\tau. \quad (3.45)$$

Također, kako bi konvolucija bila definirana mora biti $t - \tau > 0$, tj. $\tau < t$ te zbog toga gornji integral prelazi u:

$$(f * g)(t) = \int_0^t f(\tau)g(t - \tau)d\tau. \quad (3.46)$$

Teorem 3.13. Neka je $\mathcal{L}\{f(t)\} = F(s)$ i $\mathcal{L}\{g(t)\} = G(s)$. Tada je

$$\mathcal{L}\{f(t) * g(t)\} = F(s) \cdot G(s). \quad (3.47)$$

Dokaz. Iz definicije Laplaceove transformacije dobivamo:

$$\begin{aligned} \mathcal{L}\{f(t) * g(t)\} &= \int_0^{\infty} (f(t) * g(t))e^{-st}dt = \int_0^{\infty} \int_0^t f(\tau)g(t - \tau)d\tau e^{-st}dt = \\ &= \int_0^{\infty} \int_0^{\infty} f(\tau)g(t - \tau)u(t - \tau)d\tau e^{-st}dt. \end{aligned} \quad (3.48)$$

Promjenom poretka integracije slijedi

$$\begin{aligned} \int_0^{\infty} \int_0^{\infty} g(t - \tau)u(t - \tau)e^{-st}dt f(\tau)d\tau &= \int_0^{\infty} \mathcal{L}\{g(t - \tau)u(t - \tau)\}f(\tau)d\tau = \\ &= \int_0^{\infty} G(s)e^{-\tau s}f(\tau)d\tau = G(s) \int_0^{\infty} f(\tau)e^{-\tau s}d\tau = G(s) \cdot F(s). \end{aligned} \quad (3.49)$$

\square

3.7. Tablica Laplaceovih transformacija

U nastavku je prikazana tablica Laplaceovih transformacija.

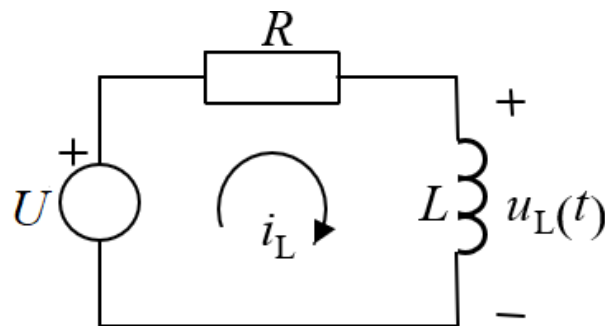
	$f(t)$	$F(s)$
1.	1	$\frac{1}{s}$
2.	c	$\frac{c}{s}$
3.	t	$\frac{1}{s^2}$
4.	t^n	$\frac{n!}{s^{n+1}}$
5.	$\frac{1}{\sqrt{\pi t}}$	$\frac{1}{\sqrt{s}}$
6.	e^{-at}	$\frac{1}{s+a}$
7.	te^{-at}	$\frac{1}{(s+a)^2}$
8.	sinat	$\frac{a}{s^2+a^2}$
9.	cosat	$\frac{s}{s^2+a^2}$
10.	$\frac{t}{2a} \cdot \sin at$	$\frac{s}{(s^2+a^2)^2}$
11.	t·cosat	$\frac{s^2-a^2}{(s^2+a^2)^2}$
12.	$e^{-at} \cdot \sin bt$	$\frac{b}{(s+a)^2+b^2}$
13.	$e^{-at} \cdot \cos bt$	$\frac{s+a}{(s+a)^2+b^2}$
14.	$e^{-at} \cdot f(t)$	$F(s+a)$
15.	$f'(t)$	$sF(s)-f(0)$
16.	$f''(t)$	$s^2F(s)-sf(0)-f'(0)$

Slika 3.4. Tablica Laplaceovih transformacija

4. Rješavanje električnih krugova pomoću Laplaceove transformacije

4.1. Serijska RL mreža

Prvi primjer koji će se proučavati biti će serijska RL mreža sa nezavisnim naponskim izvorom. U označava nezavisni naponski izvor, R je iznos otpora u mreži, L je iznos na induktivitetu mreže i i_L je iznos struje u mreži.



Slika 4.1. Serijska RL mreža, izvor: [4]

Naponski izvor U može imati istosmjernu ili izmjeničnu pobudu te će se primjeri riješavati u obje varijante.

Za prvi zadatak uzet će se da je:

$$u(t) = E, \quad (4.1)$$

odnosno kada prebacimo u s domenu:

$$U(s) = \frac{E}{s}, \quad (4.2)$$

a tražit će se struja $i(t)$.

Izraz za struju $i(t)$ dobit će se preko Kirchhoffovog zakona za napone koji glasi da je zbroj padova napona u zatvorenom strujnom krugu ili konturi jednak ukupnom naponu izvora:

$$Ri(t) + L \frac{di(t)}{dt} = u(t). \quad (4.3)$$

Prebacivanjem ovog izraza u s domenu dobit će se:

$$I(R + sL) = \frac{E}{s}. \quad (4.4)$$

Sređivanjem izraza dolazi se do rastava na parcijalne razlomke gdje nam A i B predstavljaju nepoznanice:

$$I = \frac{E}{s(sL + R)} = \frac{\frac{E}{L}}{s(s + \frac{R}{L})} = \frac{A}{s} + \frac{B}{sL + R}. \quad (4.5)$$

Iz ovog izraza dolazi se do jednadžbi koje će dati vrijednosti nepoznanica A i B :

$$As + A\frac{R}{L} + Bs = \frac{E}{L}, \quad (4.6)$$

$$A + B = 0. \quad (4.7)$$

Izjednačavanjem koeficijenata uz s s lijeve i desne strane dobit će se vrijednost nepoznanice A :

$$A\frac{R}{L} = \frac{E}{L}. \quad (4.8)$$

Množenjem jednadžbe s L dobit će se:

$$AR = E. \quad (4.9)$$

Dijeljenjem s R dobit će se vrijednost nepoznanice A :

$$A = \frac{E}{R}. \quad (4.10)$$

Iz prijašnje jednadžbe dobit će se vrijednost nepoznanice B :

$$B = -A, \quad (4.11)$$

$$B = -\frac{E}{R}. \quad (4.12)$$

Vraćanjem nepoznanica A i B u izraz za struju dobijemo vrijednost struje u s domeni:

$$I = \frac{\frac{E}{R}}{s} - \frac{\frac{E}{R}}{s + \frac{R}{L}}. \quad (4.13)$$

Prebacivanjem u vremensku domenu dobije se tražena struja $i(t)$:

$$i(t) = \frac{E}{R} - \frac{E}{R}e^{-\frac{R}{L}t} = \frac{E}{R}(1 - e^{-\frac{R}{L}t})S(t). \quad (4.14)$$

U slijedećem zadatku promatrat će se ista mreža, ali ovoga puta pobuda će biti:

$$u(t) = U_0 \sin(\omega_0 t), \quad (4.15)$$

odnosno u s domeni:

$$U(s) = \frac{U_0 \omega_0}{s^2 + \omega_0^2}. \quad (4.16)$$

Struja će se dobiti istim postupkom kao i u prethodnom zadatku:

$$Ri(t) + L\frac{di(t)}{dt} = u(t). \quad (4.17)$$

Prebacivanjem ove jednadžbe u s domenu dobit će se izraz za struju $I(s)$:

$$I(R + sL) = \frac{U_0 \omega_0}{s^2 + \omega_0^2}. \quad (4.18)$$

Sređivanjem izraza dolazimo do rastava na parcijalne razlomke gdje nam A , B i C predstavljaju nepoznanice:

$$I = \frac{U_0\omega_0}{(s^2 + \omega_0^2)(R + sL)} = \frac{\frac{U_0\omega_0}{L}}{(s^2 + \omega_0^2)(s + \frac{R}{L})} = \frac{As + B}{s^2 + \omega_0^2} + \frac{C}{s + \frac{R}{L}}. \quad (4.19)$$

Iz ovih parcijalnih razlomaka dobiju se jednadžbe koje će dati vrijednosti nepoznanica A , B i C :

$$As^2 + As\frac{R}{L} + Bs + B\frac{R}{L} + Cs^2 + C\omega_0^2 = \frac{U_0\omega_0}{L}. \quad (4.20)$$

Izjednačavanjem koeficijenata uz s^2 s lijeve i desne strane dobit će se:

$$A + C = 0, \quad (4.21)$$

$$C = -A. \quad (4.22)$$

Izjednačavanjem koeficijenata uz s s lijeve i desne strane dobit će se:

$$A\frac{R}{L} + B = 0, \quad (4.23)$$

$$B = -A\frac{R}{L}. \quad (4.24)$$

Izjednačavanjem koeficijenata bez s s lijeve i desne strane dobiti će se:

$$B\frac{R}{L} + C\omega_0^2 = \frac{U_0\omega_0}{L}. \quad (4.25)$$

Ubacivanjem nepoznanice A umjesto nepoznanica B i C dobit će se:

$$-A\frac{R}{L}\frac{R}{L} - A\omega_0^2 = \frac{U_0\omega_0}{L}. \quad (4.26)$$

Izlučivanjem nepoznanice A te prebacivanjem ostatka izraza uz A na desnu stranu dobit će se vrijednost nepoznanice A :

$$A(R^2 + \omega_0^2L^2) = -U_0\omega_0L, \quad (4.27)$$

$$A = -\frac{U_0\omega_0L}{R^2 + \omega_0^2L^2}. \quad (4.28)$$

Ubacivanjem vrijednosti A u jednadžbe za C i B dobit će se vrijednosti svih nepoznanica:

$$C = \frac{U_0\omega_0L}{R^2 + \omega_0^2L^2}, \quad (4.29)$$

$$B = \frac{U_0\omega_0L}{R^2 + \omega_0^2L^2}\frac{R}{L} = \frac{U_0\omega_0R}{R^2 + \omega_0^2L^2}. \quad (4.30)$$

Vraćanjem nepoznanica A , B i C u jednadžbu za struju I dobit će se njena vrijednost u s domeni:

$$I = -\frac{U_0\omega_0L}{R^2 + \omega_0^2L^2}\frac{s}{s^2 + \omega_0^2} + \frac{U_0R}{R^2 + \omega_0^2L^2}\frac{\omega_0}{s^2 + \omega_0^2} + \frac{U_0\omega_0L}{R^2 + \omega_0^2L^2}\frac{1}{s + \frac{R}{L}}. \quad (4.31)$$

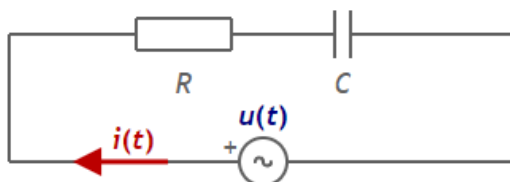
Prebacivanjem nazad u vremensku domenu dobije se tražena vrijednost struje $i(t)$:

$$i(t) = \frac{U_0R}{R^2 + \omega_0^2L^2}\sin(\omega_0t) - \frac{U_0\omega_0L}{R^2 + \omega_0^2L^2}\cos(\omega_0t) + \frac{U_0\omega_0L}{R^2 + \omega_0^2L^2}e^{-\frac{R}{L}t}S(t), \quad (4.32)$$

$$i(t) = \frac{U_0}{R^2 + \omega_0^2L^2}(R\sin(\omega_0t) - \omega_0L\cos(\omega_0t) + \omega_0Le^{-\frac{R}{L}t})S(t). \quad (4.33)$$

4.2. Serijska RC mreža

Neka je sada zadana mreža kako prikazuje Slika 4.2. Na slici je prikazana serijska RC mreža sa nezavisnim naponskim izvorom. $u(t)$ označava nezavisni naponski izvor, R je iznos otpora u mreži, C je iznos kapaciteta mreže i $i(t)$ je struja koja teče u mreži.



Slika 4.2. Serijska RC mreža, izvor: [5]

Opet će se gledati dva slučaja, prvi je za istosmjernu pobudu, a drugi za izmjeničnu pobudu. Uzeti će se iste vrijednosti napona U kao u prošlom primjeru u oba slučaja. Za prvi slučaj imamo:

$$u(t) = E, \quad (4.34)$$

odnosno kada prebacimo u s domenu:

$$U(s) = \frac{E}{s}. \quad (4.35)$$

Izraz za struju $i(t)$ u RC mreži za istosmjernu pobudu dobiti će se preko Kirchhoffovog zakona koji je objašnjen u prvom primjeru:

$$Ri(t) + \frac{1}{C} \int_0^t i(\tau) d\tau = u(t). \quad (4.36)$$

Prebacivanjem u s domenu dobit će se izraz za struju $I(s)$:

$$I\left(R + \frac{1}{sC}\right) = \frac{E}{s}. \quad (4.37)$$

Sređivanjem jednadžbe dobit će se rješenje u s domeni:

$$I = \frac{E}{s\left(R + \frac{1}{sC}\right)} = \frac{E}{sR + \frac{1}{C}} = \frac{\frac{E}{R}}{s + \frac{1}{RC}}. \quad (4.38)$$

Prebacivanjem nazad u vremensku domenu dobit će se tražena struja $i(t)$:

$$i(t) = \frac{E}{R} e^{-\frac{1}{RC}t} S(t). \quad (4.39)$$

Napon za izmjeničnu pobudu iznosi:

$$U(t) = U_0 \sin(\omega_0 t), \quad (4.40)$$

Odnosno u s domeni:

$$U(s) = \frac{U_0 \omega_0}{s^2 + \omega_0^2}. \quad (4.41)$$

Izraz za struju $i(t)$ u RC mreži za izmjeničnu pobudu isti je kao i za istosmjernu pobudu :

$$Ri(t) + \frac{1}{C} \int_{-\infty}^t i(\tau) d\tau = u(t). \quad (4.42)$$

Prebacivanjem u s domenu dobit će se izraz za struju $I(s)$:

$$I\left(R + \frac{1}{sC}\right) = \frac{U_0\omega_0}{s^2 + \omega_0^2}. \quad (4.43)$$

Sređivanjem jednadžbe dobije se:

$$I = \frac{U_0\omega_0}{(s^2 + \omega_0^2)\left(\frac{1}{sC} + R\right)} = \frac{U_0\omega_0 s}{(s^2 + \omega_0^2)\left(\frac{1}{C} + sR\right)}. \quad (4.44)$$

Jednadžba će se riješiti pomoću parcijalnih razlomaka gdje su A , B i D nepoznanice:

$$I = \frac{\frac{U_0\omega_0 s}{R}}{(s^2 + \omega_0^2)\left(s + \frac{1}{RC}\right)} = \frac{As + B}{s^2 + \omega_0^2} + \frac{D}{s + \frac{1}{RC}}. \quad (4.45)$$

Množenjem s brojnikom dobit će se:

$$As^2 + As\frac{1}{RC} + Bs + \frac{B}{RC} + Ds^2 + D\omega_0^2 = \frac{U_0\omega_0 s}{R}. \quad (4.46)$$

Izjednačavanjem koeficijanata uz s^2 s lijeve i desne strane znak jednakosti dolazi se do:

$$A + D = 0. \quad (4.47)$$

Iz čega se može zaključiti da je:

$$A = -D. \quad (4.48)$$

Nadalje, izjednačavanjem koeficijanata bez s s lijeve i desne strane znak jednakosti dobit će se:

$$\frac{B}{RC} + D\omega_0^2 = 0. \quad (4.49)$$

Izlučit će se B na jednu stranu, a sve ostalo na drugu stranu znaka jednakosti:

$$B = -D\omega_0^2 RC. \quad (4.50)$$

Zadnja jednadžba dobit će se izjednačavanjem koeficijanata uz s s lijeve i desne strane znak jednakosti:

$$\frac{A}{RC} + B = \frac{U_0\omega_0}{R}. \quad (4.51)$$

Ubacivanjem nepoznanice D umjesto nepoznanica A i B dolazi se do:

$$-\frac{D}{RC} - D\omega_0^2 RC = \frac{U_0\omega_0}{R}. \quad (4.52)$$

Izlučit će se nepoznanica D :

$$D\left(\frac{1}{RC} + \omega_0^2 RC\right) = -\frac{U_0\omega_0}{R}. \quad (4.53)$$

Dijeljenjem izraza uz nepoznanicu D dolazimo do njezine vrijednosti:

$$D = -\frac{U_0\omega_0 RC}{R(\omega_0^2 R^2 C^2 + 1)} = -\frac{U_0\omega_0 C}{\omega_0^2 R^2 C^2 + 1}. \quad (4.54)$$

Vraćanjem vrijednosti nepoznanice D dolazi se do vrijednosti nepoznanica A i B :

$$A = \frac{U_0\omega_0 C}{\omega_0^2 R^2 C^2 + 1}, \quad (4.55)$$

$$B = \frac{U_0\omega_0 C}{\omega_0^2 R^2 C^2 + 1} \omega_0^2 RC = \frac{U_0\omega_0^3 C^2 R}{\omega_0^2 R^2 C^2 + 1}. \quad (4.56)$$

Ubacivanjem nepoznanica u jednadžbu za $I(s)$ dobit će se rješenje u s domeni:

$$I = \frac{U_0\omega_0 C}{\omega_0^2 R^2 C^2 + 1} \frac{s}{s^2 + \omega_0^2} + \frac{U_0\omega_0^3 C^2 R}{\omega_0^2 R^2 C^2 + 1} \frac{\omega_0}{s^2 + \omega_0^2} - \frac{U_0\omega_0 C}{\omega_0^2 R^2 C^2 + 1} \frac{1}{s + \frac{1}{RC}}. \quad (4.57)$$

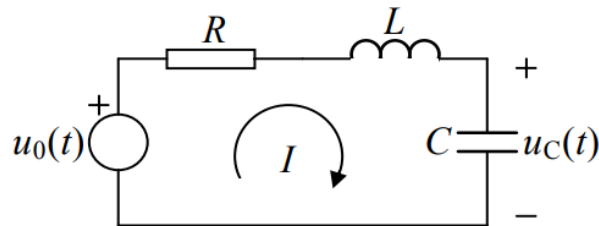
Prebacivanjem nazad u vremensku domenu dobit će se tražena struja $i(t)$:

$$i(t) = \frac{U_0\omega_0 C}{\omega_0^2 R^2 C^2 + 1} \cos(\omega_0 t) + \frac{U_0\omega_0^3 C^2 R}{\omega_0^2 R^2 C^2 + 1} \sin(\omega_0 t) - \frac{U_0\omega_0 C}{\omega_0^2 R^2 C^2 + 1} e^{-\frac{1}{RC}t} S(t), \quad (4.58)$$

$$i(t) = \frac{U_0\omega_0 C}{\omega_0^2 R^2 C^2 + 1} (\omega_0 RC \sin(\omega_0 t) + \cos(\omega_0 t) - e^{-\frac{1}{RC}t}) S(t). \quad (4.59)$$

4.3. Serijska RLC mreža

Za zadnji primjer pogledat će se serijska RLC mreža sa nezavisnim naponskim izvorom prikazana na Slici 4.3. Nezavisni naponski izvor označen je s $u_0(t)$, R je iznos otpora u mreži, L je iznos induktiviteta mreže i C je iznos kapaciteta mreže.



Slika 4.3. Serijska RLC mreža, izvor: [4]

U slučaju istosmjerne pobude:

$$u(t) = E, \quad (4.60)$$

odnosno u s domeni:

$$U(s) = \frac{E}{s}. \quad (4.61)$$

Izraz za struju u vremenskoj domeni glasi:

$$Ri(t) + L \frac{di(t)}{dt} + \frac{1}{C} \int_0^t i(\tau) d\tau = u(t). \quad (4.62)$$

Prebacivanjem u s domenu dobivamo izraz:

$$I \left(R + sL + \frac{1}{sC} \right) = \frac{E}{s}. \quad (4.63)$$

Sređivanjem jednadžbe dobije se vrijednost struje u s domeni:

$$\begin{aligned} I &= \frac{E}{s(R + sL + \frac{1}{sC})} = \frac{E}{sR + s^2L + \frac{1}{C}} = \frac{\frac{E}{L}}{s^2 + s\frac{R}{L} + \frac{1}{LC}} \\ &= \frac{\frac{E}{L}}{(s + \frac{R}{2L})^2 + (\frac{1}{LC} - \frac{R^2}{4L})} = \frac{\frac{E}{L}}{(s + \frac{R}{2L})^2 + \left(\sqrt{\frac{1}{LC} - \frac{R^2}{4L}} \right)^2} \\ &= \frac{\frac{E}{L}}{\sqrt{\frac{1}{LC} - \frac{R^2}{4L}}} \cdot \frac{\sqrt{\frac{1}{LC} - \frac{R^2}{4L}}}{(s + \frac{R}{2L})^2 + \left(\sqrt{\frac{1}{LC} - \frac{R^2}{4L}} \right)^2}. \end{aligned} \quad (4.64)$$

Prebacivanjem nazad u vremensku domenu dobijemo traženu vrijednost struje $i(t)$:

$$i(t) = \frac{\frac{E}{L}}{\sqrt{\frac{1}{LC} - \frac{R^2}{4L}}} \cdot e^{-\frac{R}{2L}t} \cdot \sin \left(\left(\sqrt{\frac{1}{LC} - \frac{R^2}{4L}} \right) t \right) S(t). \quad (4.65)$$

U slučaju izmjenične pobude vrijednost naponskog izvora je:

$$u(t) = U_0 \sin(\omega_0 t), \quad (4.66)$$

odnosno u s domeni:

$$U(s) = \frac{U_0 \omega_0}{s^2 + \omega_0^2}. \quad (4.67)$$

Izraz za struju u vremenskoj domeni je isti kao i u prethodnom zadatku:

$$Ri(t) + L \frac{di(t)}{dt} + \frac{1}{C} \int_{-\infty}^t i(\tau) d\tau = u(t), \quad (4.68)$$

odnosno prebacivanjem u s domenu uz novu vrijednost napona dobije se izraz za struju:

$$I \left(R + sL + \frac{1}{sC} \right) = \frac{U_0 \omega_0}{s^2 + \omega_0^2}. \quad (4.69)$$

Sređivanjem jednadžbe dolazimo do izraza za struju $I(s)$:

$$\begin{aligned} I &= \frac{U_0 \omega_0}{(s^2 + \omega_0^2) \left(R + sL + \frac{1}{sC} \right)} \\ &= \frac{U_0 \omega_0 s}{(s^2 + \omega_0^2) \left(sR + s^2L + \frac{1}{C} \right)} \\ &= \frac{s \frac{U_0 \omega_0}{L}}{(s^2 + \omega_0^2) \left(s^2 + s\frac{R}{L} + \frac{1}{LC} \right)}. \end{aligned} \quad (4.70)$$

Ova jednadžba riješit će se pomoću parcijalnih razlomaka:

$$I = \frac{As + B}{s^2 + \omega_0^2} + \frac{Ds + E}{s^2 + s\frac{R}{L} + \frac{1}{LC}}. \quad (4.71)$$

Vrijednosti nepoznanica A , B , D i E dobit će se pomoću jednadžbe:

$$As^3 + As^2\frac{R}{L} + As\frac{1}{LC} + Bs^2 + Bs\frac{R}{L} + B\frac{1}{LC} + Ds^3 + Ds\omega_0^2 + Es^2 + E\omega_0^2 = s\frac{U_0\omega_0}{L}. \quad (4.72)$$

Izjednačavanjem koeficijenata uz s^3 s lijeve i desne strane znaka jednakosti dolazi se do:

$$A + D = 0. \quad (4.73)$$

Iz čega se može zaključiti da je:

$$D = -A. \quad (4.74)$$

Izjednačavanjem koeficijenata bez s s lijeve i desne strane znaka jednakosti dolazi se do:

$$B\frac{1}{LC} + E\omega_0^2 = 0. \quad (4.75)$$

izlučivanjem nepoznanice E na jednu stranu, a sve ostalo na drugu stranu znaka jednakosti dobit će se vrijednost nepoznanice E preko nepoznanice B :

$$E = -B\frac{1}{\omega_0^2 LC}. \quad (4.76)$$

Izjednačavanjem koeficijenata uz s^2 s lijeve i desne strane znaka jednakosti dolazi se do:

$$A\frac{R}{L} + B + E = 0. \quad (4.77)$$

Zamjenom nepoznanice E sa ekvivalentnom vrijednosti preko nepoznanice B dobit će se:

$$A\frac{R}{L} + B - B\frac{1}{\omega_0^2 LC} = 0. \quad (4.78)$$

Nadalje, prebacit će se nepoznanica A i izraz uz nju te izlučiti nepoznanica B :

$$B\left(1 - \frac{1}{\omega_0^2 LC}\right) = -A\frac{R}{L}. \quad (4.79)$$

Dijeljenjem izraza uz nepoznanicu B dobiti će se njena vrijednosti preko nepoznanice A :

$$B = -A\frac{\omega_0^2 RC}{\omega_0^2 LC - 1}. \quad (4.80)$$

Izjednačavanjem koeficijenata uz s s lijeve i desne strane znaka jednakosti dolazi se do:

$$A\frac{1}{LC} + B\frac{R}{L} + D\omega_0^2 = \frac{U_0\omega_0}{L}. \quad (4.81)$$

Zamjenit će se nepoznanice B i D s ekvivalentnom vrijednošću izraza preko nepoznanice A :

$$A\frac{1}{LC} - A\frac{\omega_0^2 RC}{\omega_0^2 LC - 1}\frac{R}{L} - A\omega_0^2 = \frac{U_0\omega_0}{L}. \quad (4.82)$$

Izlučivanjem nepoznanice A dobit će se:

$$A \left(\frac{1}{C} - \frac{\omega_0^2 R^2 C}{\omega_0^2 LC - 1} - \omega_0^2 L \right) = U_0 \omega_0. \quad (4.83)$$

Dijeljenjem izraza uz A dobit će se vrijednost nepoznanice A :

$$A = \frac{U_0 \omega_0}{\frac{1}{C} - \frac{\omega_0^2 R^2 C}{\omega_0^2 LC - 1} - \omega_0^2 L} \cdot \frac{C(\omega_0^2 LC - 1)}{C(\omega_0^2 LC - 1)}. \quad (4.84)$$

Sada će se srediti jednadžba kako bi se kasnije lakše došlo do vrijednosti ostalih nepoznanica:

$$\begin{aligned} A &= \frac{U_0 \omega_0 C(\omega_0^2 LC - 1)}{\omega_0^2 LC - 1 - \omega_0^2 R^2 C^2 - \omega_0^2 LC(\omega_0^2 LC - 1)} \\ &= \frac{U_0 \omega_0 C(\omega_0^2 LC - 1)}{\omega_0^2 LC - 1 - \omega_0^2 R^2 C^2 - \omega_0^4 L^2 C^2 + \omega_0^2 LC} \\ &= -\frac{U_0 \omega_0 C(\omega_0^2 LC - 1)}{\omega_0^4 L^2 C^2 + \omega_0^2 R^2 C^2 + 1}. \end{aligned} \quad (4.85)$$

Preko nepoznanice A dobit će se vrijednosti nepoznanica B i D iz prijašnjih jednadžbi:

$$D = \frac{U_0 \omega_0 C(\omega_0^2 LC - 1)}{\omega_0^4 L^2 C^2 + \omega_0^2 R^2 C^2 + 1}, \quad (4.86)$$

$$B = \frac{U_0 \omega_0 C(\omega_0^2 LC - 1)}{\omega_0^4 L^2 C^2 + \omega_0^2 R^2 C^2 + 1} \cdot R \cdot \frac{\omega_0^2 C}{\omega_0^2 LC - 1}, \quad (4.87)$$

$$B = \frac{U_0 \omega_0^3 RC^2}{\omega_0^4 L^2 C^2 + \omega_0^2 R^2 C^2 + 1}. \quad (4.88)$$

Za kraj, preko nepoznanice B dolazi se do vrijednosti nepoznanice E :

$$E = -\frac{U_0 \omega_0^3 RC^2}{\omega_0^4 L^2 C^2 + \omega_0^2 R^2 C^2 + 1} \cdot \frac{1}{\omega_0^2 LC}, \quad (4.89)$$

$$E = -\frac{\frac{U_0 \omega_0 RC}{L}}{\omega_0^4 L^2 C^2 + \omega_0^2 R^2 C^2 + 1}. \quad (4.90)$$

Vraćanjem nepoznanica nazad u jednadžbu za struju $I(s)$ dobit će se vrijednost struje u s domeni:

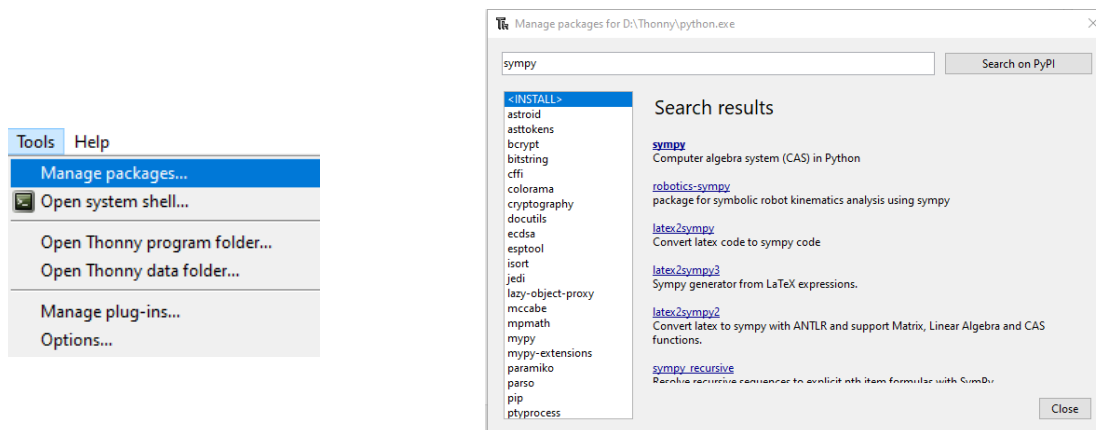
$$\begin{aligned} I &= -\frac{U_0 \omega_0 C(\omega_0^2 LC - 1)}{\omega_0^4 L^2 C^2 + \omega_0^2 R^2 C^2 + 1} \cdot \frac{s}{s^2 + \omega_0^2} \\ &+ \frac{U_0 \omega_0^2 RC^2}{\omega_0^4 L^2 C^2 + \omega_0^2 R^2 C^2 + 1} \cdot \frac{\omega_0}{s^2 + \omega_0^2} \\ &+ \frac{U_0 \omega_0 C(\omega_0^2 LC - 1)}{\omega_0^4 L^2 C^2 + \omega_0^2 R^2 C^2 + 1} \cdot \frac{s + \frac{R}{2L}}{\left(s + \frac{R}{2L}\right)^2 + \left(\sqrt{\frac{1}{LC} - \frac{R^2}{4L}}\right)^2} \\ &- \frac{\frac{U_0 \omega_0 RC(\omega_0^2 C + \frac{1}{L})}{2}}{(\omega_0^4 L^2 C^2 + \omega_0^2 R^2 C^2 + 1) \cdot \sqrt{\frac{1}{LC} - \frac{R^2}{4L}}} \cdot \frac{\sqrt{\frac{1}{LC} - \frac{R^2}{4L}}}{\left(s + \frac{R}{2L}\right)^2 + \left(\sqrt{\frac{1}{LC} - \frac{R^2}{4L}}\right)^2}. \end{aligned} \quad (4.91)$$

Prebacivanjem nazad u vremensku domenu dobit će se traženi $i(t)$:

$$\begin{aligned}
 i(t) = & -\frac{U_0\omega_0 C(\omega_0^2 LC - 1)}{\omega_0^4 L^2 C^2 + \omega_0^2 R^2 C^2 + 1} \cdot \cos \omega_0 t \\
 & + \frac{U_0\omega_0^2 RC^2}{\omega_0^4 L^2 C^2 + \omega_0^2 R^2 C^2 + 1} \cdot \sin \omega_0 t \\
 & + \frac{U_0\omega_0 C(\omega_0^2 LC - 1)}{\omega_0^4 L^2 C^2 + \omega_0^2 R^2 C^2 + 1} \cdot e^{-\frac{R}{2L}t} \cdot \cos \left(\left(\sqrt{\frac{1}{LC} - \frac{R^2}{4L}} \right) t \right) \\
 & - \frac{U_0\omega_0 RC(\omega_0^2 C + \frac{1}{L})}{2(\omega_0^4 L^2 C^2 + \omega_0^2 R^2 C^2 + 1) \cdot \sqrt{\frac{1}{LC} - \frac{R^2}{4L}}} \cdot e^{-\frac{R}{2L}t} \cdot \sin \left(\left(\sqrt{\frac{1}{LC} - \frac{R^2}{4L}} \right) t \right) S(t).
 \end{aligned} \tag{4.92}$$

5. Laplaceova transformacija u Pythonu

Kako bi se koristila Laplaceova transformacija u Pythonu, potrebno je u njemu instalirati i odgovarajuću biblioteku *SymPy*. U ovoj verziji Pythona to se napravilo odabравši naredbu *Tools* → *Manage packages* te u novom prozoru koji se otvorio mora se upisati biblioteka *SymPy* i kliknuti *INSTALL*.



Slika 5.1. Primjer instalacije biblioteke *SymPy*

Uz biblioteku *SymPy* biti će potrebna i biblioteka *matplotlib* koja će služiti za crtanje grafova u Pythonu. Nju također instaliramo kao i *SymPy* samo u tražilicu umjesto *SymPy* upisujemo *matplotlib*.

5.1. *SymPy*

Navedimo najprije neke osnovne informacije o biblioteci *SymPy*. *SymPy* se koristi za rješavanje problema povezanim sa simboličkim računom, kao što je primjerice pojednostavljivanje različitih algebarskih izraza. U ovom radu fokusirati ćemo se na izračun Laplaceovih transformacija i traženje Laplaceovih originala. Naredba koja će služiti za tu svrhu je `sympy.integrals.transforms`. Ona poziva već unaprijed napravljene integralne transformacije unutar *SymPy*-a. Ovdje koristimo Laplaceove transformacije pa se koristiti naredba `from sympy.integrals.transforms import Laplace transform`.

5.2. Izračunavanje Laplaceovih transformacija unutar Pythona

U ovom poglavlju će se pokazati rješavanje nekih jednostavnijih primjera pomoću Pythona. Kako se prije navelo prvo se treba *importati* *SymPy* i *matplotlib*. Nakon toga moraju se definirati

simbol s kojima će se raditi, u ovom slučaju to će biti t , s i a . U Pythonu bi to trebalo izgledati ovako:

```
1 import sympy
2 sympy.init_printing()
3 import matplotlib.pyplot as plt
4 t, s = sympy.symbols('t, s')
5 a = sympy.symbols('a', real=True, positive=True)
6 f = sympy.exp(-a*t)
7 f=sympy.integrate(f*sympy.exp(-s*t), (t, 0, sympy.oo))
```

Slika 5.2. Definiranje simbola

Sada se mora definirati funkcija koja će se koristiti u Laplaceovoj transformaciji e^{-at} . Direktnom integracijom ove funkcije dobiva se komplicirano rješenje jer dolazi nepravilni integral koji ide u beskonačnost.

```
1 import sympy
2 sympy.init_printing()
3 import matplotlib.pyplot as plt
4 t, s = sympy.symbols('t, s')
5 a = sympy.symbols('a', real=True, positive=True)
6 f = sympy.exp(-a*t)
7 f=sympy.integrate(f*sympy.exp(-s*t), (t, 0, sympy.oo))
```

Slika 5.3. Traženje Laplaceove transformacije direktnom integracijom

Izlaz:

$$\begin{cases} \frac{1}{s \left(\frac{a}{s} + 1 \right)}, \text{ for } |arg(s)| \leq \frac{\pi}{2} \\ \int_0^{\infty} e^{-at} e^{-st} dt, \text{ otherwise} \end{cases}$$

Kako ovaj pristup očito nije zadovoljavajući, Laplaceove transformacije izračunavati ćemo korištenjem gotovih naredbi kako je prikazano u nastavku.

Izlaz:

$$\left(\frac{1}{a + s}, 0, True \right)$$

Može se vidjeti da se nije dobilo samo rješenje koje se očekivalo $\frac{1}{a+s}$, nego skup rješenja jer nisu bili postavljeni nikakvi početni uvjeti. Ukoliko se želi dobiti samo rješenje Laplaceove transformacije mora se iskoristiti funkcija *noconds* i postaviti njena vrijednost na *True*.

```

1 import sympy
2 sympy.init_printing()
3 import matplotlib.pyplot as plt
4 t, s = sympy.symbols('t, s')
5 a = sympy.symbols('a', real=True, positive=True)
6 f = sympy.exp(-a*t)
7 f=sympy.laplace_transform(f, t, s)

```

Slika 5.4. Izračun Laplaceove transformacije pomoću ugrađenih naredbi

```

1 import sympy
2 sympy.init_printing()
3 import matplotlib.pyplot as plt
4 t, s = sympy.symbols('t, s')
5 a = sympy.symbols('a', real=True, positive=True)
6 f = sympy.exp(-a*t)
7 F=sympy.laplace_transform(f, t, s)
8 F = sympy.laplace_transform(f, t, s, noconds=True)
9

```

Slika 5.5. Korištenje `noconds` funkcije

Izlaz:

$$\frac{1}{a + s}$$

Sada se može kodirati i inverzna Laplaceova transformacija koja nam vraća početnu funkciju. Ona se kao i Laplaceova transformacija nalazi u biblioteci *SymPy* te se poziva naredbom `sympy.inverse Laplace transform`. Pripadni kod prikazan je na slici 5.6.

Ovaj kod kao izlaz daje izraz:

$$e^{-at}\theta(t)$$

θ je simbol u *SymPy*-u za Heavisideovu step funkciju koja ograničava funkciju da počinje nakon $t = 0$. Ukoliko stavimo da nam je $a = 2$, pomoću funkcije `plot` možemo nacrtati graf naše funkcije, a kod za taj proces prikazan je na slici 5.7. Sam grafički prikaz dan je na slici 5.8.

Kako se može vidjeti na slici crvenom bojom prikazan je graf funkcije $e^{-2t}\theta(t)$, tj. rješenje inverzne Laplaceove transformacije. Plavom bojom prikazan je graf funkcije e^{-2t} . Iz ovog primjera vidi se važnost Heavisideove *step* funkcije kod inverzne Laplaceove transformacije.


```

1 import sympy
2 sympy.init_printing()
3 import matplotlib.pyplot as plt
4 t, s = sympy.symbols('t, s')
5 a = sympy.symbols('a', real=True, positive=True)
6 f = sympy.exp(-a*t)
7 F = sympy.laplace_transform(f, t, s, noconds=True)
8 def invL(F):
9     return sympy.inverse_laplace_transform(F, s, t)

```

Slika 5.6. Inverzna Laplaceova transformacija

```

1 import sympy
2 sympy.init_printing()
3 import matplotlib.pyplot as plt
4 t, s = sympy.symbols('t, s')
5 a = sympy.symbols('a', real=True, positive=True)
6 f = sympy.exp(-a*t)
7 F = sympy.laplace_transform(f, t, s, noconds=True)
8 def invL(F):
9     return sympy.inverse_laplace_transform(F, s, t)
10 invL(F).subs({a: 2})
11 p = sympy.plot(f.subs({a: 2}), invL(F).subs({a: 2}),
12              xlim=(-1, 4), ylim=(0, 3), show=False)
13 p[1].line_color = 'red'
14 p.show()

```

Slika 5.7. Crtanje grafa funkcije

5.3. Rješavanje električnih krugova unutar Pythona

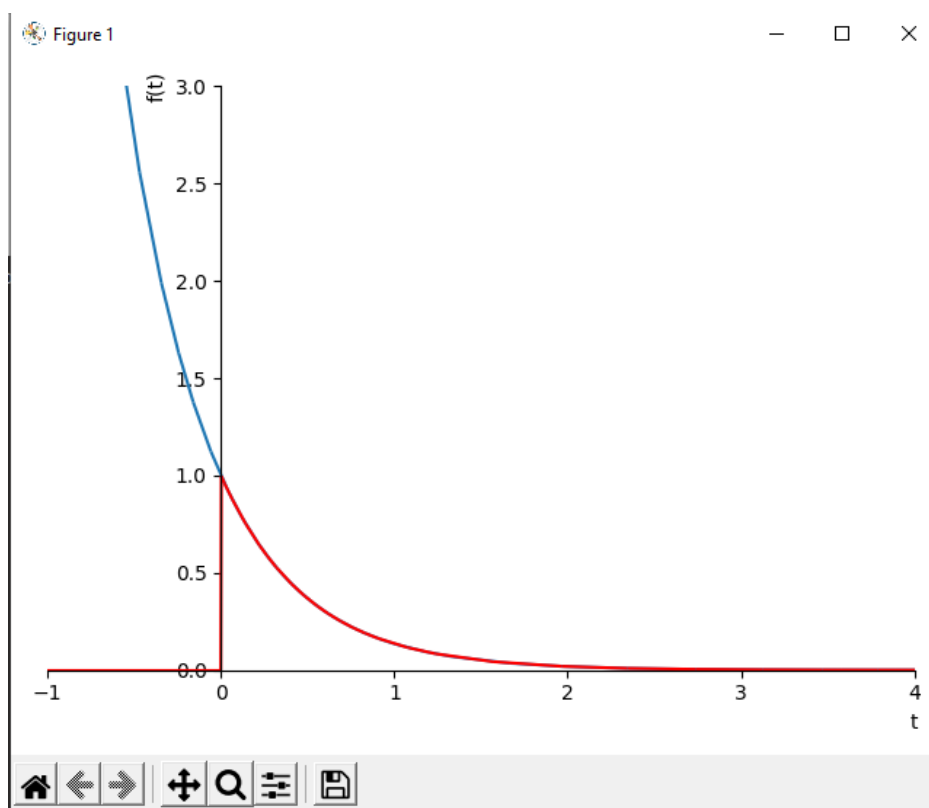
Sada će se pokazati kako unutar Pythona riješiti primjere koji su bili analitički riješeni u prethodnom poglavlju.

5.3.1. Serijska RL mreža

Za prvi primjer uzela se serijska RL mreža sa istosmjernom pobudom, s obzirom da Python ne može računati s općim brojevima za rješavanje zadatka u Pythonu uzet će se slijedeće vrijednosti:

$$R = 1, \quad (5.1)$$

$$L = 1, \quad (5.2)$$



Slika 5.8. Graf inverzne Laplaceove transformacije

$$E = 10.$$

(5.3)

Kod za rješavanje diferencijalne jednačbe:

```

1 from sympy.integrals.transforms import Laplace_transform
2 from sympy.integrals.transforms import inverse_Laplace_transform
3 from sympy import *
4 import sympy as sympy
5 from sympy.abc import t, i, s, I
6 init_printing(use_unicode=True)
7 i, I = symbols('i I', cls = Function)
8 R = 1
9 L = 1
10 E = 10
11 eom = Eq(I(s)*(R+s*L), E/s)
12 print(eom)
13 I = solve(eom,I(s))
14 print(I[0])
15 soln = inverse_Laplace_transform(I[0],s,t)
16 print(soln)

```

Slika 5.9. RL mreža, istosmjerna pobuda

Izlaz:

$$10 * (\exp(t) - 1) * \exp(-t) * \text{Heaviside}(t)$$

Rješenje dobiveno Pythonom je vrijednošću isto kao i kada se rješavalo analitički, ali je malo kompliciranije. Određenim matematičkim funkcijama rješenje se može pojednostavniti kako bi se dobilo identično rješenje.

Slijedeći primjer koji se rješavao bio je serijska RL mreža s izmjeničnom pobudom. Opet će se umjesto općih brojeva uzimati zadane vrijednosti:

$$R = 1, \quad (5.4)$$

$$L = 1, \quad (5.5)$$

$$U_0 = 10, \quad (5.6)$$

$$\omega_0 = 1. \quad (5.7)$$

```

1 from sympy.integrals.transforms import Laplace_transform
2 from sympy.integrals.transforms import inverse_Laplace_transform
3 from sympy import *
4 import sympy as sympy
5 from sympy.abc import t, i, s, I
6 init_printing(use_unicode=True)
7 i, I = symbols('i I', cls = Function)
8 R = 1
9 L = 1
10 U0 = 10
11 w0 = 1
12 eom = Eq(I(s)*(R+s*L), U0*w0/(s**2+w0**2))
13 print(eom)
14 I = solve(eom,I(s))
15 print(I[0])
16 soln = inverse_Laplace_transform(I[0],s,t)
17 print(soln)

```

Slika 5.10. RL mreža, izmjenična pobuda

Izlaz:

$$5 * (\exp(t) * \sin(t) - \exp(t) * \cos(t) + 1) * \exp(-t) * \text{Heaviside}(t)$$

Ponovno je rješenje u Pythonu malo kompliciranije, ali kada bi se sredilo dobila bi se ista vrijednost kao i kada se zadatak riješio analitički.

5.3.2. Serijska RC mreža

Druga mreža koja se gledala bila je serijska RC mreža sa istosmjernom pobudom. Ponovno se moraju opći brojevi zamijeniti zadanim vrijednosti koje su:

$$R = 1, \quad (5.8)$$

$$C = 1, \quad (5.9)$$

$$E = 10. \quad (5.10)$$

```

1 from sympy.integrals.transforms import Laplace_transform
2 from sympy.integrals.transforms import inverse_Laplace_transform
3 from sympy import *
4 import sympy as sympy
5 from sympy.abc import t, i, s, I
6 init_printing(use_unicode=True)
7 i, I = symbols('i I', cls = Function)
8 R = 1
9 C = 1
10 E = 10
11 eom = Eq(I(s)*(R+1/s*C), E/s)
12 print(eom)
13 I = solve(eom,I(s))
14 print(I[0])
15 soln = inverse_Laplace_transform(I[0],s,t)
16 print(soln)

```

Slika 5.11. RC mreža, istosmjerna pobuda

Izlaz:

```
10*exp(-t)*Heaviside(t)
```

Može se vidjeti da se ovaj put dobilo isto rješenje u Pythonu kao i kada se zadatak rješavao analitički.

Za primjer serijske RC mreže sa izmjeničnom pobudom kao i do sada moraju se staviti zadane vrijednosti umjesto općih brojeva:

$$R = 1, \quad (5.11)$$

$$L = 1, \quad (5.12)$$

$$U_0 = 10, \quad (5.13)$$

$$\omega_0 = 1. \quad (5.14)$$

```

1 from sympy.integrals.transforms import Laplace_transform
2 from sympy.integrals.transforms import inverse_Laplace_transform
3 from sympy import *
4 import sympy as sympy
5 from sympy.abc import t, i, s, I
6 init_printing(use_unicode=True)
7 i, I = symbols('i I', cls = Function)
8 R = 1
9 C = 1
10 U0 = 10
11 w0 = 1
12 eom = Eq(I(s)*(R+1/s*C), U0*w0/(s**2+w0**2))
13 print(eom)
14 I = solve(eom,I(s))
15 print(I[0])
16 soln = inverse_Laplace_transform(I[0],s,t)
17 print(soln)

```

Slika 5.12. RC mreža, izmjenična pobuda

Izlaz:

```
5*(exp(t)*sin(t) + exp(t)*cos(t) - 1)*exp(-t)*Heaviside(t)
```

Ponovno je rješenje u Pythonu malo kompliciranije, ali kada bi se sredilo dobila bi se ista vrijednost kao i kada se zadatak riješio analitički.

5.3.3. Serijska *RLC* mreža

Zadnja mreža koja se razmatrala bila je serijska *RLC* mreža. Za početak opet ćemo riješiti primjer sa istosmjernom pobudom i zadanim vrijednostima umjesto općih brojeva:

$$R = 1, \quad (5.15)$$

$$L = 2, \quad (5.16)$$

$$C = 1, \quad (5.17)$$

$$E = 10. \quad (5.18)$$

Izlaz:

```
20*sqrt(7)*exp(-t/4)*sin(sqrt(7)*t/4)*Heaviside(t)/7
```

```

1 from sympy.integrals.transforms import Laplace_transform
2 from sympy.integrals.transforms import inverse_Laplace_transform
3 from sympy import *
4 import sympy as sympy
5 from sympy.abc import t, i, s, I
6 init_printing(use_unicode=True)
7 i, I = symbols('i I', cls = Function)
8 R = 1
9 L = 2
10 C = 1
11 E = 10
12 eom = Eq(I(s)*(R+s*L+1/s*C), E/s)
13 print(eom)
14 I = solve(eom,I(s))
15 print(I[0])
16 soln = inverse_Laplace_transform(I[0],s,t)
17 print(soln)

```

Slika 5.13. RLC mreža, istosmjerna pobuda

Za kraj riješit će se serijska RLC mreža sa izmjeničnom pobudom, naravno opet se opći brojevi moraju prebaciti u zadane vrijednosti:

$$R = 1, \quad L = 2, \quad L = 1, \quad U_0 = 10, \quad \omega_0 = 1. \quad (5.19)$$

```

1 from sympy.integrals.transforms import Laplace_transform
2 from sympy.integrals.transforms import inverse_Laplace_transform
3 from sympy import *
4 import sympy as sympy
5 from sympy.abc import t, i, s, I
6 init_printing(use_unicode=True)
7 i, I = symbols('i I', cls = Function)
8 R = 1
9 L = 2
10 C = 1
11 U0 = 10
12 w0 = 1
13 eom = Eq(I(s)*(R+s*L+1/s*C), U0*w0/(s**2+w0**2))
14 print(eom)
15 I = solve(eom,I(s))
16 print(I[0])
17 soln = inverse_Laplace_transform(I[0],s,t)
18 print(soln)

```

Slika 5.14. RLC mreža, izmjenična pobuda

Za ovaj primjer Python nije uspio doći do rješenja.

6. Zaključak

Unutar ovog rada cilj je bio opisati kako se unutar Pythona primjenjuje Laplaceova transformacija za rješavanje matematičkih zadataka, s posebnim naglaskom na primjere iz elektrotehnike.

Prvo smo se upoznali s programskim jezikom Pythonom, njegovim povijesnim razvojem i počecima, te današnjom primjenom Pythona.

Laplaceovu transformaciju prvo smo objasnili iz teorijskog stajališta, gdje smo se bavili njenom egzistencijom i svojstvima, a kasnije smo pokazali i njenu primjenu u struci.

U praktičnom dijelu rada riješeno je nekoliko primjera zadataka riješenih preko Laplaceove transformacije koji su vezani uz struku, gdje smo radili izračune struje u RL , RC i RLC strujnom krugu.

Python, iako primarno programski jezik za programiranje, može se koristiti i za rješavanje određenih simboličkih matematičkih problema na brz i jednostavan način. Uz to moramo napomenuti kako je velika prednost Pythona što je besplatan te lagana dostupnost kako njemu tako i svim njegovim dodatnim bibliotekama.

Kada smo s Pythonom riješili primjere, možemo vidjeti da smo na lakši i brži način dobili točna rješenja za sve primjere osim zadnjega, koji je bio kompliciraniji za rješavanje. Zato možemo zaključiti da ukoliko želimo riješiti neke jednostavnije primjere možemo koristiti Python, ali za teže matematičke izračune morat ćemo koristiti neke naprednije programske sustave, npr. MATLAB ili Mathematicu.

Za kraj možemo reći kako se Python pokazao kao dobar, brz i jednostavan izbor za rješavanje lakših matematičkih problema, ali ukoliko imamo kompliciranije matematičke izračune ne možemo se pouzdati da ćemo dobiti krajnje rješenje.

Literatura

- [1] <https://en.wikipedia.org/wiki/File:Guido-portrait-2014-drc.jpg> 31.srpnja.2021
- [2] <https://sh.wikipedia.org/wiki/Pierre-Simon-Laplace> 31.srpnja.2021
- [3] Dražić, I. Laplaceova transformacija - skripta,interna skripta, Sveučilište u Rijeci, Tehnički fakultet, 2019. 25.kolovoza.2021
- [4] Stojković, N. Teorija mreža i linija, Sveučilište u Rijeci, Tehnički fakultet, Rijeka, 2002. 26.srpnja.2022
- [5] <https://osnove.tel.fer.hr/VJEZBEOE/AC-5.htm?x=174> 26.srpnja.2022
- [6] Elezović, N. Fourierov red i integral, Laplaceova transformacija, Zagreb, Element, 2007.
- [7] D.V. Widder, The Laplace Transform, Princeton University Press, Princeton , 1946.
- [8] M. Essert, I. Vazler, Python-osnove, Sveučilište u Osijeku, Odjel za matematiku Sveučilišta u Osijeku, Osijek, 2007
- [9] N. Stojković., Elektronika 1 - zavodska skripta, Sveučilište u Rijeci, Tehnički fakultet, Rijeka, 2015.
- [10] "Pierre Simon de Laplace", s Interneta, <http://www.enciklopedija.hr/natuknica.aspx?id=35431>, 05.09.2018.
- [11] "Poisson's and Laplace's equations", s Interneta, <https://ece.illinois.edu/>, 30.kolovoza.2022.
- [12] Laplace transform, Wikipedia, s Interneta, <https://en.wikipedia.org/wiki/Laplace-transform> 3.rujna.2022
- [13] G. Doetsch, Introduction to the Theory and Application of the Laplace Transformation, Springer-Verlag, Berlin, 1974.
- [14] D.V. Widder, The Laplace Transform, Princeton University Press, Princeton , 1946.
- [15] s Interneta, <http://josephcslater.github.io/solve-ode.html> 3.rujna.2022

Sažetak i ključne riječi

Unutar ovog rada prikazana je implementacija Laplaceovih transformacija u programskom jeziku Python, s naglaskom na rješavanje diferencijalnih jednadžbi vezanih uz struku. Ukratko je opisana povijest Pythona, naredbe unutar njega te biblioteke koje su nam bile potrebne za rad. Također, opisana je i Laplaceova transformacija, njena egzistencija i teoremi te primjena u praksi. Za primjer su odabrani RL , RC i RLC serijski spoj u kojima se računala struja kroz elemente spoja. Programskim rješavanjem dobivena su ista rješenja kao i prilikom ručnog rješavanja.

Ključne riječi: Python, Laplaceova transformacija, inverzna Laplaceova transformacija, analiza električne mreže

Summary and key words

Within this paper, also the implementation of Laplace transforms in the Python programming language is presented, with an emphasis on solving differential equations related to the profession. The history of Python, the commands inside it and the libraries we needed to work are briefly described. Also, the Laplace transformation, its existence and theorems and application in practice are described. For the example, RL , RC and RLC series circuits were selected, in which the current through the elements of the circuit was calculated. The same solutions were obtained by program solving as during manual solving.

Keywords: Python, Laplace transform, inverse Laplace transform, electrical network analysis