

JuliaCAP

Studenti:

Glorija Došlo 2019/0065

Nikola Radojević 2019/0176

Tamara Petković 2019/0177

Ivana Stanojević 2019/0257

Kristina Rajković 2019/0447

Mentori:

prof. dr Dejan Tošić

prof. dr Milka Potrebić

Elektrotehnički fakultet, Teorija električnih kola, februar 2022. g.

Opis

- JuliaCAP je biblioteka za rešavanje linearnih, vremenski nepromenljivih električnih kola.
- Koristi modifikovanu nodalnu analizu pri formulaciji i rešavanju jednačina.
- Programski kod pisan je u programskom jeziku Julia ([The Julia Programming Language \(julialang.org\)](https://julialang.org)).
- Preporučena okruženja za korišćenje su Visual Studio Code i Jupyter Notebook.

Podržani elementi

- R – otpornik
- V_g – idealni naponski generator
- I_g – idealni strujni generator
- OpAmp – idealni operacioni pojačavač
- VCVS – naponski izvor upravljan naponom
- VCCS – strujni izvor upravljan naponom
- CCVS – naponski izvor upravljan strujom
- CCCS – strujni izvor upravljan strujom
- L – kalem
- C – kondenzator

Podržani elementi (1)

- IdealT – idealni transformator
- InductiveT – induktivan transformator
- ABCD – četvoropol, pogonski parametri
- Y – admitansa
- Z – impedansa
- T – vod

Metode

- `noviGraf()` – pravi novi graf i vraća ga kao rezultat
- `dodajGranu(graph, Grana(tip, ime, cvor1, cvor2, param))` – u postojeći graf dodaje novi element
 - ❖ `tip` – `TipGrane`, uzima vrednosti ranije prikazanih oznaka za elemente
 - ❖ `ime` – `String`, ime elementa
 - ❖ `cvor1` – `Vector of Integer`, pozitivan terminal
 - ❖ `cvor2` – `Vector of Integer`, negativan terminal
 - ❖ `param` – `Vector`, parametri elementa
- `resiKolo(graph, omega = 'w')` – rešava električno kolo

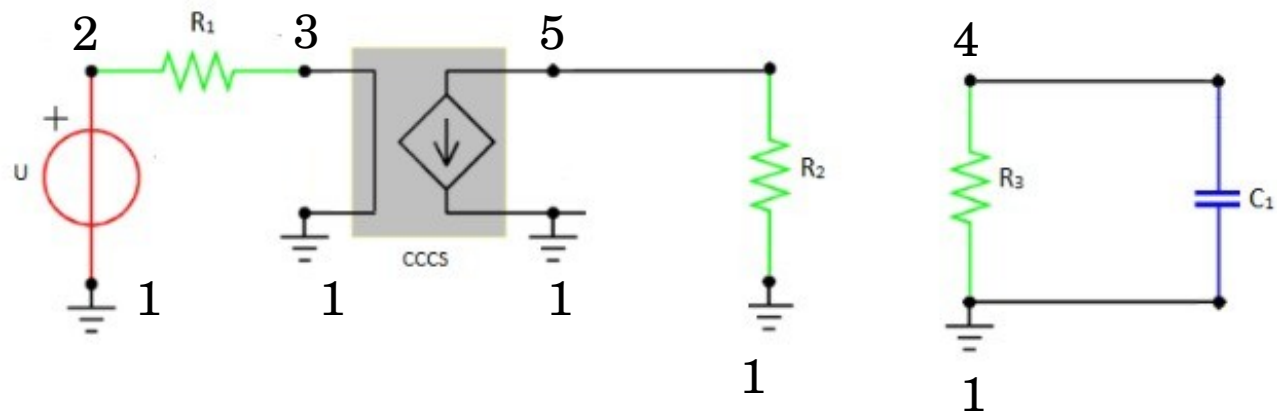
Metode (1)

- `ispisi_rezultate (rezultat)` – ispisuje napone čvorova i struje
- `ispisi_rezultate_latex (rezultat)` – ispisuje izraze u latex - u
- `ispisi_jednacine ()` – ispisuje postavljene jednačine
- `ispisi_jednacine_latex ()` – ispisuje postavljene jednačine u latex - u
- `ispisi_specifican_rezultat (rezultat, “x”)` – ispisuje samo traženu struju ili napon zadat drugim parametrom
- `ispisi_specifican_rezultat_latex (rezultat)` – ispisuje samo traženu struju ili napon zadat drugim parametrom u latex - u
- `ispisi_specifikacije_kola ()` – ispisuje broj čvorova, elemente, varijable...

Korišćenje

- `include ("JuliaCAP.jl")` – učitava se biblioteka u korisnički program
- `using .JuliaCAP`
- `graf = noviGraf ()` – kreiranje novog grafa
- `...` – dodavanje elemenata na prethodno prikazan način
- `rezultat = resiKolo (graf; omega = "w")` – rešavanje kola
- `ispisi_rezultate (rezultat)` – ispis rezultata

Primer 1 – kolo



Primer 1 – kod

```
include("JuliaCap.jl")
using .JuliaCAP

graf = noviGraf()

dodajGranu(graf, Grana(Vg, "U", [2], [1], ["U"]))
dodajGranu(graf, Grana(R, "R1", [2], [3], ["R1"]))
dodajGranu(graf, Grana(R, "R2", [1], [5], ["R2"]))
dodajGranu(graf, Grana(R, "R3", [1], [4], ["R3"]))
dodajGranu(graf, Grana(C, "C1", [1], [4], ["C1"]))
dodajGranu(graf, Grana(CCCS, "CCCS", [3, 1], [5, 1], ["a"]))

rezultat = resiKolo(graf; omega = "w")

ispisi_jednacine()

print("\n Resenje \n")

ispisi_rezultate_latex(rezultat)
```

Primer 1 – izlaz

$$U2 \sim U$$

$$U3 \sim 0$$

$$IU + (1 / R1) * (U2 - U3) \sim 0$$

$$ICCCS - (1 / R1) * (U2 - U3) \sim 0$$

$$U4 * (1 / R3) + C1 * U4 * j_w \sim 0$$

$$ICCCS * a + U5 * (1 / R2) \sim 0$$

Resenje

```
\begin{equation}
```

$$ICCCS = U \frac{1}{R1} - R1 \frac{1}{R1} \left(U \frac{1}{R1} - \left(\frac{1}{R1} \right)^2 R1 U \right)$$

```
\end{equation}
```

```
\begin{equation}
```

$$U5 = \frac{U a \frac{1}{R1} - R1 a \frac{1}{R1} \left(U \frac{1}{R1} - \left(\frac{1}{R1} \right)^2 R1 U \right)}{- \frac{1}{R2}}$$

```
\end{equation}
```

```
\begin{equation}
```

$$U3 = R1 \left(U \frac{1}{R1} - \left(\frac{1}{R1} \right)^2 R1 U \right)$$

```
\end{equation}
```

```
\begin{equation}
```

$$U2 = U$$

```
\end{equation}
```

```
\begin{equation}
```

$$U4 = 0$$

```
\end{equation}
```

```
\begin{equation}
```

$$IU = \frac{- R1 U \frac{1}{R1}}{R1}$$

```
\end{equation}
```

Primer 1 – rešenje

$$ICCS = U \frac{1}{R1} - R1 \frac{1}{R1} \left(U \frac{1}{R1} - \left(\frac{1}{R1} \right)^2 R1 U \right) \quad (1)$$

$$U5 = \frac{U a \frac{1}{R1} - R1 a \frac{1}{R1} \left(U \frac{1}{R1} - \left(\frac{1}{R1} \right)^2 R1 U \right)}{-\frac{1}{R2}} \quad (2)$$

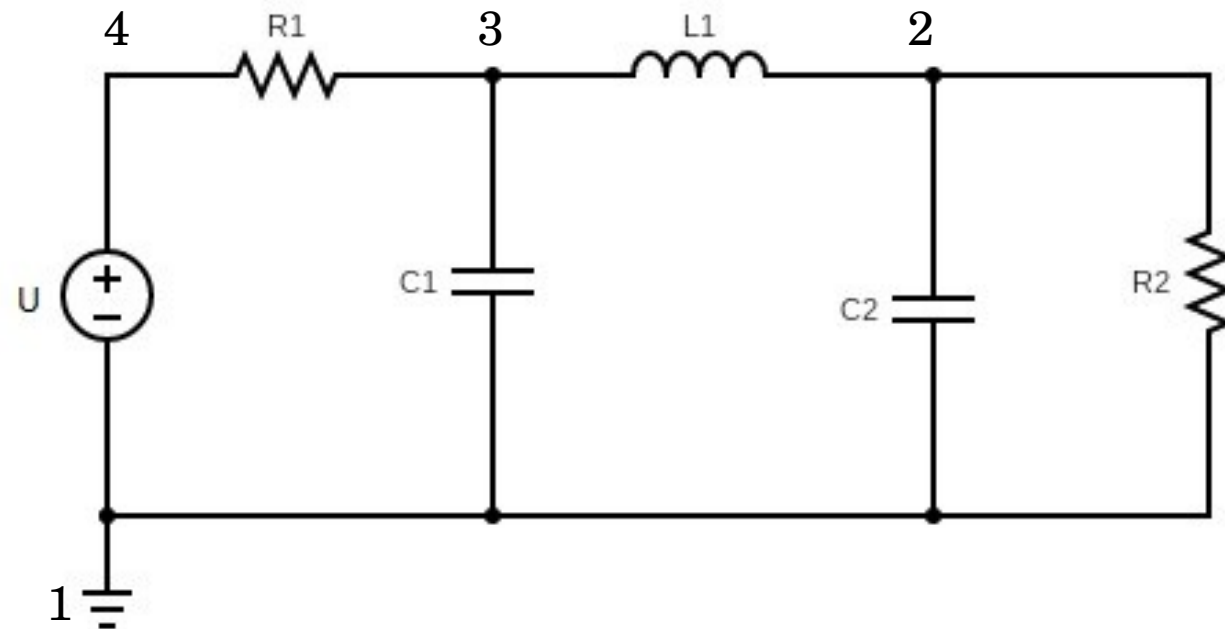
$$U3 = R1 \left(U \frac{1}{R1} - \left(\frac{1}{R1} \right)^2 R1 U \right) \quad (3)$$

$$U2 = U \quad (4)$$

$$U4 = 0 \quad (5)$$

$$IU = \frac{-R1 U \frac{1}{R1}}{R1} \quad (6)$$

Primer 2 – kolo



Primer 2 – kod

```
include("JuliaCap.jl")
using .JuliaCAP

graf = noviGraf()

dodajGranu(graf, Grana(Vg, "U", [4], [1], ["U"]))
dodajGranu(graf, Grana(R, "R1", [4], [3], ["R"]))
dodajGranu(graf, Grana(C, "C1", [3], [1], ["C"]))
dodajGranu(graf, Grana(C, "C2", [1], [2], ["C"]))
dodajGranu(graf, Grana(L, "L1", [2], [3], ["L"]))
dodajGranu(graf, Grana(R, "R2", [2], [1], ["R"]))

rezultat = resiKolo(graf; omega = "w")

ispisi_jednacine()

print("\n Resenje \n")

ispisi_rezultate_latex(rezultat)
```

Primer 2 – izlaz

```
U4 ~ U
U2*(1 / R) + (U2 - U3)*(1 / (L*jw)) + C*U2*jw ~ 0
(U3 - U2)*(1 / (L*jw)) + C*U3*jw - (1 / R)*(U4 - U3) ~ 0
IU + (1 / R)*(U4 - U3) ~ 0
```

Resenje

```
\begin{equation}
U3 = - L jw \frac{ - U \left( \frac{-1}{R} + \frac{-1}{L jw} - C jw \right) \frac{1}{R} }{\frac{1}{L jw} - \left( \frac{-1}{R} + \frac{-1}{L jw} - C jw \right)^2 L jw}
\end{equation}
```

```
\begin{equation}
U2 = \frac{ - U \frac{1}{R} }{\frac{1}{L jw} - \left( \frac{-1}{R} + \frac{-1}{L jw} - C jw \right)^2 L jw}
\end{equation}
```

```
\begin{equation}
U4 = U
\end{equation}
```

```
\begin{equation}
IU = - U \left( \frac{1}{R} + \frac{ - \left( \frac{1}{R} \right) \right)^2 L jw \left( \frac{-1}{R} + \frac{-1}{L jw} - C jw \right) \frac{1}{L jw} - \left( \frac{-1}{R} + \frac{-1}{L jw} - C jw \right)^2 L jw \right)
\end{equation}
```

Primer 2 – rešenje

$$U3 = -Ljw \frac{-U \left(\frac{-1}{R} + \frac{-1}{Ljw} - Cjw \right) \frac{1}{R}}{\frac{1}{Ljw} - \left(\frac{-1}{R} + \frac{-1}{Ljw} - Cjw \right)^2 Ljw} \quad (1)$$

$$U2 = \frac{-U \frac{1}{R}}{\frac{1}{Ljw} - \left(\frac{-1}{R} + \frac{-1}{Ljw} - Cjw \right)^2 Ljw} \quad (2)$$

$$U4 = U \quad (3)$$

$$IU = -U \left(\frac{1}{R} + \frac{-\left(\frac{1}{R}\right)^2 Ljw \left(\frac{-1}{R} + \frac{-1}{Ljw} - Cjw \right)}{\frac{1}{Ljw} - \left(\frac{-1}{R} + \frac{-1}{Ljw} - Cjw \right)^2 Ljw} \right) \quad (4)$$

Problemi

- Julia je generalno nov programski jezik i još uvek se razvija.
- Shodno tome, nije bilo moguće rešavati kola uz pomoć Laplasove transformacije.
- Takođe, biblioteke koje su korišćene za rad sa matematičkim izrazima (Symbolics, SymbolicUtils) imaju ograničenja kada je reč o uprošćavanju. Do neke mere se mogu koristiti funkcije sa ovom namenom (simplify), međutim nije utvrđeno kada izraz postaje suviše „komplikovan“ za uprošćavanje.
- Problem je nastao i kod funkcije solve_for koja nije vraćala Vector of Any tako da je moralo da se izvrši prebacivanje rezultata u novi niz i dalje radi sa njim.

Hvala na pažnji!