

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Milivoj Mandić

**ELEKTROENERGETSKI PRORAČUN U
ELEKTROVUČNOM SUSTAVU 25 kV, 50 Hz**

MAGISTARSKI RAD

Zagreb, 2006.

Magistarski rad je izrađen u **Zavodu za visoki napon i energetiku Fakulteta elektrotehnike i računarstva u Zagrebu.**

Mentor: **prof. dr. sc. Ivo Uglešić, dipl. ing.**

Magistarski rad ima: **159** stranica

Magistarski rad br.:

Povjerenstvo za ocjenu magistarskog rada:

1. Prof.dr.sc. Zdravko Hebel – predsjednik
2. Prof.dr.sc. Ivo Uglešić – mentor
3. Prof.dr.sc. Milenko Stegić, Fakultet strojarstva i brodogradnje Sveučilišta u Zagrebu

Povjerenstvo za obranu magistarskog rada:

1. Prof.dr.sc. Zdravko Hebel – predsjednik
2. Prof.dr.sc. Ivo Uglešić – mentor
3. Prof.dr.sc. Milenko Stegić, Fakultet strojarstva i brodogradnje Sveučilišta u Zagrebu

Magistarski rad je obranjen 25. svibnja 2006. godine.

Zahvaljujem svima koji su pomogli u pisanju ovog rada svojim stručnim savjetima i prijedlozima, a posebno mentoru.

Zahvaljujem svojoj obitelji na podršci i pomoći kroz cijeli studij.

Zahvaljujem djevojci Marijeti na bezgraničnom strpljenju koje je iskazala u periodu pisanja ovog rada.

Sadržaj:

1. UVOD	1
2. SIMULATOR KRETANJA VLAKOVA U IZMJENIČNOM ELEKTROVUČNOM SUSTAVU	3
2.1. Fizikalna slika kretanja vlaka.....	3
2.2. Potrebna snaga za vuču vozila	11
2.3. Matematički model za simulator kretanja vlakova	15
2.4. Režimi vožnje vlaka	18
2.5. Programska podrška simulacije električne vuče.....	20
3. ODREĐIVANJE ELEKTRIČNIH PARAMETARA KONTAKTNE MREŽE	30
3.1. Proračun impedancija kontaktne mreže 25 kV, 50 Hz.....	30
3.1.1. Općenito o impedanciji	30
3.1.2. Impedancija voznog voda	34
3.1.3. Impedancija i napon povratnog voda	36
3.1.4. Međusobna impedancija vozni vod – povratni vod.....	39
3.1.5. Impedancija kontaktne mreže.....	41
3.1.6. Primjer proračuna impedancija kontaktne mreže	42
3.1.7. Prikaz rezultata dobivenih proračunima	44
4. FORMIRANJE ELEKTRIČNE MREŽE	49
4.1. Algoritam formiranja električne mreže	50
4.2. Implementacija algoritma u potprogramu PEPES i prikaz izlaznih rezultata	52
5. PRORAČUN ELEKTRIČNIH PRILIKA U ELEKTROVUČNOM SUSTAVU	57
5.1. Proračun tokova snaga primjenom matrične metode	59
5.2. Implementacija algoritma za proračun električnih prilika u potprogramu GAUS.....	62
5.3. Prikaz rezultata dobivenih elektrovnim proračunom kontaktne mreže	65
5.4. Rezime i analiza rezultata	79
6. ZAKLJUČAK	80
7. SAŽETAK I KLJUČNE RIJEČI	81
8. LITERATURA	83

9. POPIS OZNAKA.....	85
10. PRILOZI.....	85
11. ŽIVOTOPIS.....	86

1. Uvod

Elektrifikacija pruga, pored prednosti koje uvodi u željeznički promet, nosi sa sobom i čitav niz problema koje je potrebno riješiti da bi se ista prije svega mogla izvesti a naravno i što ekonomičnije i pouzdanije koristiti. O problemima pri izvođenju elektrifikacije željeznica (izbor sustava napajanja, kontaktne mreže i elektrovučnog motora) raspravljalo se još početkom prošlog stoljeća kada su i nastali glavni sustavi napajanja elektrificiranih željeznica. Znatnije preinake na tom polju danas uglavnom nisu moguće prije svega što su željeznički sustavi već dovoljno razvijeni pa bi bilo kakva promjena vodila velikim financijskim izdacima. Ipak, na novim prugama, se pokušavaju uvoditi razne novine od sustava napajanja, kontaktne mreže, električne lokomotive pa do sasvim novih vrsta prometnih sustava.

Problemi ekonomičnog i pouzdanog korištenja električne vuče su naravno uvijek aktualni. Kako pri elektrifikaciji pruge, uz zadane prometne parametre, ekonomično rasporediti EVP-e da bi se zadovoljili isti, koliko je takvo napajanje pouzdano? Kako, na već elektrificiranoj pruzi, napraviti vozni red da bi potrošnja električne energije i iznos angažirane snage bili minimalni, a sve uz uvjet ispunjenja zadanih prometnih parametara? Kakva je mogućnost paralelnog rada, kod sustava izmjenične električne vuče, više EVP-a? Koliko i kako se paralelnim radom više EVP-a povećava propusna moć pruge? Što se zbiva pri ispadu EVP-a? Koliko su susjedni EVP-i u mogućnosti kvalitetno napajati vlakove po tekucem redu vožnje ili nekom planiranom? Kako paralelni pogon više EVP-a utječe na primarnu elektroenergetsku mrežu s obzirom da su isti monofazni potrošači priključeni na istu fazu?

Javlja se i *problem održavanja amplitude napona* kontaktne mreže koji je posebno izražen kod dugih voznih vodova opterećenih vlakovima koji pri pokretanju vuku veliku jalovu snagu iz mreže. Mogućnost propada napona, zbog potrebe za velikom jalovom snagom kao i zbog induktivnog karaktera vodova kontaktne mreže, dovodi do potrebe za održavanjem amplitude napona u svakoj točki KM-e. Za poboljšanje stabilnosti napona se koristi kompenzator jalove snage (TCR), ali je on također izvor viših harmonika u mreži. Zato se ugrađuju dodatni filtri da bi prigušili te harmonike. TCR zajedno s filtrima formira sustav za regulaciju napona [2].

Ovo su samo neka od pitanja koja se susreću pri elektrifikaciji pojedinih pruga i u svakodnevnom pogonu sustava elektrificiranih željeznica. Na svako pojedino pitanje, naravno, nije moguće odgovoriti bez detaljnog proračuna. U taj se proračun, kao prvo, moraju uzeti parametri pruge, lokomotive i vlaka da bi se dobila snaga koju zahtjeva pojedini vlak. Kako se vlak kreće to je naravno, zbog točnosti, potrebno vršiti simulaciju kretanja u što kraćim vremenskim intervalima. Jedan EVP napaja općenito više vlakova pa je jasno da se električna mreža koju on napaja svakog trenutka mijenja. Za razliku od EVP-a, koji je u takvoj mreži fiksno čvorište, vlakovi predstavljaju promjenjiva čvorišta i udaljenost između tih čvorišta se mijenja. Vlak se može zaustaviti u stanici, pa takvo čvorište može u potpunosti nestati. Potrebno je opet i takvu mrežu proračunavati u što kraćem vremenskom intervalu. Jasno se vidi, i na prvi pogled, da je prije donošenja bilo kakvog zaključka potrebno vršiti izuzetno opsežan i naporan proračun. Kada se dodaju uz to i razne nelinearnosti i potreba njihovog iterativnog rješavanja, jasno je da rješavanje ovog zadatka zahtjeva složeni računarski program kao i računalo koje može vršiti ovakve proračune da neuporedivo kraće vrijeme nego prije 10-ak godina.

Naravno, razvijanje programa koji bi mogao da vrši sve moguće proračune i modeliranja u kontaktnoj mreži elektrificiranih željeznica zahtjeva više potprograma za dobivanje prikladnih rezultata o električnim prilikama na nekoj pruzi.

Svi oni zajedno čine program koji je prvenstveno namijenjen za proračun električnih prilika u izmjeničnom sustavu 1 x 25 kV kakav je u upotrebi na HŽ (osim Riječke pruge). Potreba za ovim programom je došla zbog izmjene sustava električne vuče na pruzi Moravice – Rijeka i na priključnim vodovima. Sadašnji sustav napajanja na toj dionici je 3 kV istosmjerno i trebao bi se promijeniti na izmjenični elektrovučni sustav 25 kV.

U ovom magistarskom radu se pokušalo napraviti što bolju spregu među tim potprogramima tako da bi izvršavanje bilo kompatibilnije sa Windows sučeljem. Dana je usporedba rezultata prvog potprograma napisanog u FORTRANU i BORLAND C++. Uobičajeni rad takvog sustava je da pojedini EVP napaja određeni dio pruge (radijalni pogon). Između takvih dijelova nalazi se neutralna sekcija. EVP-i u radijalnom pogonu često su spojeni na različite faze primarne elektroenergetske mreže da bi se postigla kakva takva simetrija opterećenja. Neutralne sekcije imaju funkciju da odvoje dijelove kontaktne mreže koji su pod naponom različitih faza. Nasuprot tome susjedni EVP-i se mogu spojiti na istu fazu primarne elektroenergetske mreže, ako je ista dovoljno jaka, i pomoću neutralne sekcije spojiti u paralelni rad. Ovakav način rada sigurno doprinosi poboljšanju naponskih prilika u kontaktnoj mreži ali može dovesti i do toga da poteku struje izjednačenja između pojedinih EVP-a. Struje izjednačenja se javljaju kao posljedica nejednakosti napona napajanja kontaktne mreže, po iznosu, kutu ili obadvoje. Ta nejednakost je najčešće posljedica nejednakosti napona primarne elektroenergetske mreže u čvorištima u kojima su priključeni EVP-i, a može biti i zbog drugih uzroka.

Za proračun električnih prilika u kontaktnoj mreži potrebno je također poznavati i maksimalni vozni red vlakova za pojedinu dionicu. Vozni red se izrađuje (dimenzionira) na osnovi istraživanja putničkih i robnih tokova u zavisnosti o raspoloživih kapaciteta željeznice i on ima veliku ulogu za korisnike i željezničke radnike jer se prema njemu odvija tehnološki proces proizvodnje željeznice. Pomoću njega treba biti određeno sigurno i uredno odvijanje prometa vlakova i racionalno korištenje kapaciteta. Imamo **opći** i **posebni** vozni red. Temeljni element voznog reda jest plan prijevoza putnika i tereta, a ostali su: vrsta vuče i potreban broj lokomotiva, masa i broj vlakova, maksimalna brzina, vozna vremena, stanični intervali, intervali slijeđenja, službena mjesta zaustavljanja vlakova, podjela manevarskog rada i plan sastava vlakova.

Tema ovog rada je i određivanje električnih parametara KM 25 kV, 50 Hz. Posebna pažnja se posvetila proračunu impedancija kontaktne mreže direktnim (postepenim) i matričnim (blok-transformacija) postupkom i uspoređivanju dobivenih rezultata.

2. Simulator kretanja vlakova u izmjeničnom elektrovučnom sustavu

Za promatranje električnih prilika u kontaktnoj mreži, tj. iznosa napona u pojedinim točkama i opterećenja pojedinih elektrovučnih podstanica (EVP), potrebno je poznavati položaje vlakova i snage koje su potrebne za ostvarenje kretanja.

U simulaciji kretanja najprije se izračunavaju položaji vlakova, kao i njihove mehaničke i električne snage potrebne za vuču. Nakon toga je moguće odrediti električne prilike u kontaktnoj mreži, poput padova napona, struja kao i opterećenja EVP-a. Da bi se vlak mogao kretati određenom dionicom pruge, potrebno je da vučno vozilo ima potrebnu snagu za savladavanje ukupnih otpora predviđenom brzinom, te da uz kočenje vlak može pouzdano zaustaviti na bilo kojem dijelu pruge. Za određivanje snage vučnog vozila moraju se izračunati otpori kretanja konkretnog vlaka na konkretnoj dionici. Dakle, ulazni podaci neophodni za takav proračun su parametri profila pruge, planirane brzine kretanja na pojedinim dionicama, te karakteristike vlaka i lokomotive [5].

2.1. Fizikalna slika kretanja vlaka

U ovom poglavlju je korištena pretežno literatura [7], [8], i [14].

Pri kretanju vozila prugom pojavljuju se različiti otpori vožnje koji se protive tom kretanju. Da bi otpori bili savladani, vučno vozilo mora na obodu pogonskih kotača ostvariti vučnu silu jednaku zbroju svih otpora.

Otpori vožnje su promjenjivi. Postoji više različitih otpora vožnje, a to su:

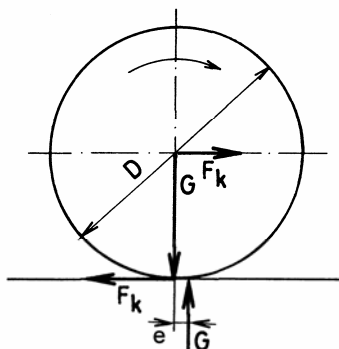
F_p - stalni otpori, F_v - povremeni otpori, F_a - otpor ubrzanja

Otpori ovise o mnoštvu različitih faktora, a utjecaj nekih je vrlo teško analitički utvrditi. To je razlog da se za izračunavanje nekih otpora vožnje koriste iskustvene formule koje se mogu razlikovati u različitim zemljama.

Stalni otpori se pojavljuju se uvijek pri kretanju vlaka. Ove otpore čine: F_k - otpori kotrljanja, F_{par} - otpori parazitskih kretanja, F_z - otpori zraka

Otpori kotrljanja su posljedica deformacije kotača i tračnica, te trenja koje se između njih pojavljuje. Pri kretanju kotača po tračnici nastaju elastične deformacije (i kotača i tračnice) koje se opiru kretanju kotača. Kod idealno elastičnog kotača i tračnice, sila za stvaranje deformacije s nailazeće strane bila bi jednaka potisnoj sili na strani odlaženja kotača.

U stvarnosti je sila za deformaciju veća od potisne sile. Rezultantna sila R ima normalnu komponentu G na udaljenosti e od osi kotača (slika 2.1).



Slika 2.1. Pojednostavljena shema sila koje djeluju na vučeni kotač po tračnici

Sila potrebna za savladavanje otpora kotrljanja je:

$$F_k = G \cdot \frac{2 \cdot e}{D} \quad (2.1.1)$$

Pokusima je utvrđeno da vrijedi odnos: $\frac{G}{F_k} = 1000$ (2.1.2)

G je opterećenje kojim kotač djeluje na tračnicu.

Za promjer kotača $D = 1000$ mm dobiva se krak otpor kotrljanja:

$$e = \frac{F_k \cdot D}{2 \cdot G} = \frac{1}{1000} \cdot \frac{1000}{2} = 0.5 \quad \text{mm} \quad (2.1.3)$$

Otpor kotrljanja W_k po apsolutnom je iznosu jednak sili F_k a suprotan po smjeru, pa je:

$$W_k = F_k = G \cdot \frac{2 \cdot e}{D} \quad \text{N} \quad (2.1.4)$$

Specifični otpor kotrljanja dobije se svođenjem otpora na jedinicu mase koja uzrokuje opterećenje kotača G :

$$w_k = \frac{W_k}{m} = g \cdot \frac{W_k}{G} = g \cdot \frac{2 \cdot e}{D} \quad \frac{\text{N}}{\text{kg}} \quad (2.1.5)$$

Uz promjer kotača $D=1000$ mm i krak $e=0.5$ mm dobije se:

$$w_k = 9.81 \cdot \frac{2 \cdot 0.5}{1000} \approx 0.01 \quad \frac{\text{N}}{\text{kg}} \quad (2.1.6)$$

Očito je da specifični otpor kotrljanja ovisi o promjeru kotača. Nema izravne ovisnosti tog otpora o brzini vožnje. Međutim, ta ovisnost postoji posredno, jer se povećanjem brzine vožnje povećava sila ispred kotača u odnosu na silu iza kotača, čime se zapravo povećava krak otpora kotrljanja e . Ovom otporu treba pridodati otpor kotrljanja u ležajevima kolnih slogova (pod kolnim slogom podrazumijeva se osovina s dva čvrsto navučena kotača), koji ovisi o osovinskom opterećenju te faktoru trenja μ u ležajevima. Faktor trenja je veličina koja ovisi o mnogim faktorima: o konstrukciji ležaja, opterećenju, brzini vožnje, temperaturi, te kvaliteti podmazivanja... U proračunima stalnih otpora, otpor trenja u ležajevima često se uzima zajedno s otporom kotrljanja, kao član konstantnog iznosa.

Otpori parazitskih kretanja su posljedica vijugavog kretanja i ljuljanja vozila po pruži, pri čemu vozilo naizmjenično naliježe na obje tračnice, čime se dodatno povećava trenje. Vrlo ih je teško posebno odrediti, pa se obično dodaju stalnim otporima kao konstantna veličina. Ovi otpori su relativno manji od ostalih otpora, pa nepreciznost u njihovom određivanju nema osjetnog utjecaja na izračunavanje ukupnih otpora vožnje.

Otpori zraka se mogu podijeliti na otpore tlaka i otpore trenja. Otpor tlaka se pojavljuje na mjestima relativnog mirovanja zraka u odnosu na vozilo (vlak). To su čelna površina vlaka gdje nastaje nadtlak i stražnja površina gdje se pojavljuje podtlak.

Otpor trenja nastaje na površinama po kojima zrak struji, to su bočne stranice te krov i pod. Nije moguće određivanje pojedinih komponenti otpora zraka, već se računa ukupan otpor zraka.

Najčešće se u tu svrhu koristi formula:

$$F_z = \frac{\rho \cdot v_r^2}{2} \cdot C_x \cdot S \quad \text{N} \quad (2.1.7)$$

Gdje je:

ρ - gustoća zraka, kg/m³

v_r - relativna brzina vlaka u odnosu na zrak, m/s

C_x - faktor otpora zraka

S - najveća površina poprečnog presjeka vlaka, m²

Relativna brzina vlaka u odnosu na zrak jednaka je stvarnoj brzini vlaka u slučaju da nema vjetra. Pojavom vjetra (ovisno o njegovoj brzini i smjeru) relativna će brzina vlaka biti različita od stvarne brzine. U proračunima se to najčešće uzima u obzir tako da se stvarna brzina vlaka poveća za dodatak Δv prilikom proračuna otpora zraka. Obično se uzima $\Delta v = 3.33$ m/s (12 km/h).

Faktor C_x ovisi o obliku vučnog vozila, odnosno cijelog vlaka. Određuje se eksperimentalno u zračnim tunelima. Njegove vrijednosti se kreću od 1, za stare konstrukcije oštih rubova i skokovitih promjena presjeka, do 0.2 kod veoma aerodinamičnih oblika. Obzirom da otpor zraka raste sa kvadratom brzine, a potrebna snaga za savladavanje tih otpora sa trećom potencijom brzine, nizak faktor C_x je neophodan za vlakove koji voze velikim brzinama. Pri malim brzinama otpor zraka može se zanemariti.

Zbrajanjem svih navedenih otpora dobije se ukupni stalni otpor vožnje vlaka:

$$F_p = F_k + F_{par} + F_z \quad (2.1.8)$$

Računanje pojedinih komponenti je vrlo složeno, pa se u većini slučajeva proračun vrši na temelju empirijskih izraza koji su, obično, polinomi brzine. To su formule oblika:

$$F_p = A \cdot v^2 + B \cdot v + C \quad (2.1.9)$$

Koeficijenti A, B i C ovise o vrsti vozila i sastavu vlaka. Željeznice u različitim zemljama koriste vlastite formule, koje se međusobno razlikuju u vrijednostima koeficijenata. U HŽ-u se koriste formule po Strahl-u, gdje se specifični otpori posebno računaju za vučno vozilo, a posebno za vlak. Za vučno vozilo vrijedi izraz:

$$f_l = \left(20 + \frac{v^2}{240} \right) \cdot 10^{-3}, \quad \frac{\text{N}}{\text{kg}} \quad (2.1.10)$$

Ova jednakost vrijedi za električne lokomotive 1142, ali ćemo je koristiti i za druge tipove lokomotiva i elektrovučno vozilo.

Specifični otpor vučenih vozila računa se prema formuli:

$$f_{vl} = \left(20 + k \cdot \frac{v^2}{10} \right) \cdot 10^{-3}, \quad \frac{\text{N}}{\text{kg}} \quad (2.1.11)$$

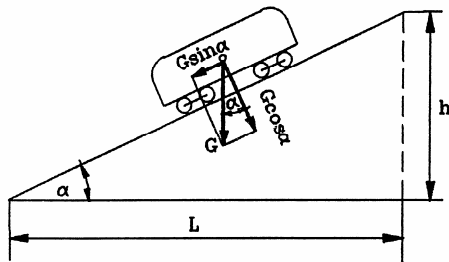
U prethodnim formulama brzina je izražena u km/h. Apsolutne vrijednosti otpora dobit će se umnoškom specifičnog otpora i mase. Faktor k ovisi o vrsti vlaka:

$k=0.108$ - prazna teretna kola
 $k=0.057$ - teretni vlak srednje tovareni
 $k=0.047$ - brzi teretni vlak
 $k=0.040$ - putnički vlak
 $k=0.032$ - brzi putnički vlak

Povremeni otpori su otpori koji se pojavljuju ovisno o profilu pruge. Tu spadaju otpori na usponu i otpori u zavoju.

Otpor na usponu na vozilo koje se nalazi na usponu, čiji je nagib pod kutom α , se određuje pomoću sile F_i koja je paralelna tračnicama (slika 2.2): $F_i = \pm G \cdot \sin \alpha$, N

Taj otpor mora biti savladan da bi se vozilo moglo kretati uz uspon. U slučaju kretanja niz uspon ovaj otpor ima negativan predznak i djeluje kao vučna sila.



Slika 2.2. Prikaz sila na usponu (otpor na usponu)

Za vrlo male nagibe α (kakvi se sreću na prugama) vrijede slijedeći izrazi (slika 2.2):

$$\sin \alpha \approx \tan \alpha \quad \tan \alpha = \frac{h}{L} \quad i = \frac{h}{L} \cdot 1000 \quad (2.1.12)$$

Iz toga slijedi pojednostavljeni izraz za F_i : $F_i = \pm G \cdot \frac{i}{1000}$, N

Svede li se taj otpor na jedinicu mase vozila odnosno vlaka, dobit će se specifični otpor na usponu:

$$f_i = \frac{F_i}{m_{vl}} = \pm \frac{\frac{G \cdot i}{1000}}{\frac{G}{9.81}} = \pm \frac{9.81 \cdot i}{1000} \approx \pm \frac{i}{100}, \quad \frac{\text{N}}{\text{kg}} \quad (2.1.13)$$

Otpori u zavoju su posljedica konstrukcije željezničkih vozila i to čvrsto navučenih kotača na osovinu, te međusobno paralelnih osovina na vozilu ili okretnom postolju.

Otpore u zavoju stvaraju:

- trenje između oboda kotača i tračnica zbog klizanja koje nastaje jer kotači istog kolnog sloga prelaze različite putove,
- trenje zbog poprečnog klizanja kotača pri zakretanju vozila,
- trenje između grebena bandaža kotača i tračnice zbog prisilnog skretanja vozila kao i zbog centrifugalne sile koja djeluje na vozilo u zavoju.

Utjecaj centrifugalne sile u zavoju na povećanje trenja smanjuje se nadvisivanjem vanjske tračnice. Paralelnost kolnih slogova onemogućuje njihovo radijalno postavljanje u zavoju.

Ugradnjom okretnih postolja u kojima su kolni slogovi međusobno blizu, taj utjecaj na otpore u zavoju se smanjuje. U proračunu otpora u zavoju koriste se eksperimentalno dobivene formule. U tim izrazima uglavnom egzistira polumjer zavoja kao najutjecajnija veličina na otpor u zavoju. Jedna od formula za izračunavanje otpora u zavoju je:

$$f_z = \frac{8000}{R} \cdot 10^{-3}, \quad \frac{N}{kg} \quad (2.1.14)$$

Otpor ubrzavanja se pojavljuje pri svakoj promjeni brzine. Promatra li se vlak kao kruto tijelo, može se zamijeniti materijalnom točkom (u težištu) koja se giba translatorno. Da bi se vlaku povećala brzina neophodno je djelovanje vučne sile na nekom putu. Kinetičku energiju

pri kretanju vlaka čini: energija translatornog gibanja: $E_{kt} = \frac{m \cdot v^2}{2}$ i energija dijelova koji

rotiraju: $E_{kr} = \sum \frac{J_0 \cdot \omega^2}{2} + \sum \frac{J_i \cdot \omega_i^2}{2}$

Gdje je:

- J_0 polarni moment tromosti kolnog sloga
- J_i polarni moment tromosti i-tog rotirajućeg dijela na vozilu (npr. rotor vučnog vozila)
- ω_i kutna brzina i-tog rotirajućeg dijela na vozilu

Budući da vrijede slijedeći izrazi: $\omega = \frac{v}{r}$ i $\omega_i = \frac{v \cdot p_i}{r}$ ukupna kinetička energija vlaka je:

$$E_k = \left(m + \sum \frac{J_0}{r^2} + \sum \frac{J_i}{r^2} \cdot p_i^2 \right) \cdot \frac{v^2}{2} \quad \text{ili} \quad E_k = m \cdot (1 + \varepsilon) \cdot \frac{v^2}{2} \quad (2.1.15)$$

Gdje je ε koeficijent rotirajućih masa, a karakteristične vrijednosti koeficijenta ε su od 0.06 do 0.08. Diferencijal kinetičke energije materijalne točke jednak je elementarnom radu sile koja djeluje na točku tj.:

$$dE_k = m \cdot (1 + \varepsilon) \cdot v \cdot dv = dA = (F_t - F_w) \cdot ds \quad (2.1.16)$$

$$\text{Također vrijedi: } F_a = F_t - F_w \quad v = \frac{ds}{dt} \quad a = \frac{dv}{dt} \quad (2.1.17)$$

gdje je: F_t - vučna sila
 F_w - zbroj otpora vožnje

$$\text{Otpor ubrzavanja je: } F_a = m \cdot (1 + \varepsilon) \cdot a \quad (2.1.18)$$

$$\text{a ubrzanje je dano izrazom: } a = \frac{F_t - F_w}{m \cdot (1 + \varepsilon)} = \frac{dv}{dt} \quad (2.1.19)$$

Nakon što smo odredili sile otpora koje se pojavljuju prilikom kretanja pojedinog vlaka potrebno je i odrediti **vučni pasoš** (slika 2.3.) koji po svojoj konturi predstavlja granične

mogućnosti vučnog vozila [7]. Grafički se prikazuje kao ovisnost vučne sile o brzini. Vučni pasoš daje osnovno obilježje vučnih mogućnosti i popratni je tehnički dokument svakog vučnog vozila.

Ako vučno vozilo ima instaliranu najveću snagu P_{maks} i koju može davati u cijelom rasponu brzina, idealna promjena vučne sile u odnosu na brzinu vožnje bit će po zakonu hiperbole jer je [7]:

$$P_{maks} = F_t \cdot v = konst. \quad (2.1.20)$$

Pri malim brzinama vučna sila može dostići veoma velike vrijednosti i premašiti silu adhezije i nastupi proklizavanje. Da ne bi došlo do proklizavanja pogonskih kotača po tračnicama mora biti ispunjen slijedeći uvjet [7]:

$$F_t \leq \psi \cdot G_{at} = F_{at} \quad (2.1.21)$$

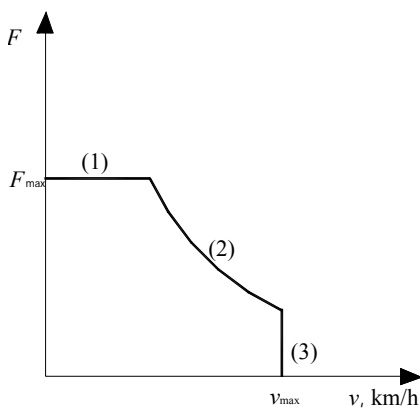
tj. vučna sila mora biti manja ili jednaka sili adhezije F_{at} , gdje je:

G_{at} – ateziona težina,

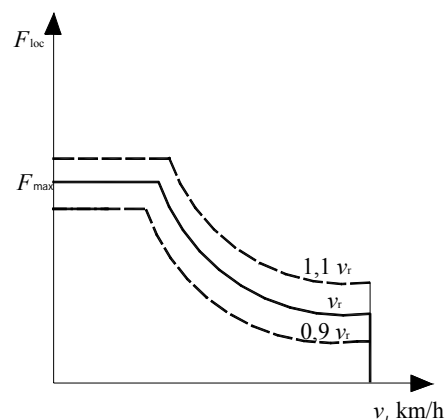
ψ - faktor adhezije.

Tok crte ovog ograničenja ima isti oblik kao faktor adhezije ψ . Još jedna od bitnih stavki vučnog pasoša je i maksimalna brzina vučnog vozila v_{maks} što predstavlja jednu od graničnih crta u vučnom pasošu. Vučna sila ovisi i o naponu kontaktne mreže (slika 2.4).

Iz tih razmatranja dobiva se kompletna granična crta vučnog pasoša u općem obliku, koju čini crta sile adhezije (1), crta najveće snage (2) i crta najveće brzine (3) (slika 2.3.).



Slika 2.3. Prikaz mogućeg vučnog pasoša



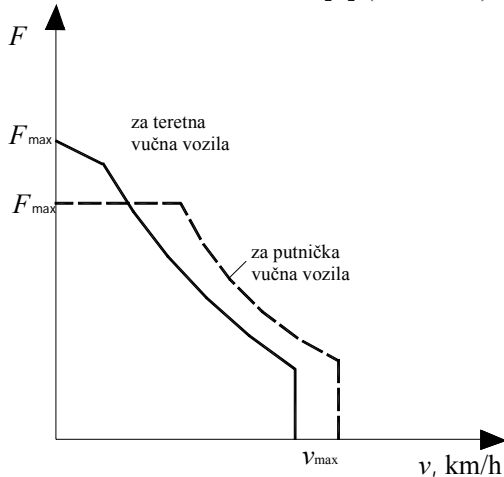
Slika 2.4. Ovisnost F_{loc} o naponu

Osim spomenutih graničnih crta u vučnom pasošu pojedina vučna vozila s elektrovučnim motorima, zbog velike sile adhezije, često imaju ograničenu vučnu silu po termičkom kriteriju (granica dopuštenog zagrijavanja elektromotora). Tako se definira trajna vučna sila (F_{tr}) koju vučno vozilo može trajno ostvarivati, satna vučna sila (F_h) kojom se vučno vozilo smije opteretiti najduže jedan sat, a postoje i kraća vremenska ograničenja (30 minuta, 15 minuta, itd.). Potreba korištenja povećanih vučnih sila pojavljuje se pri pokretanju vlaka, većem ubrzanju i na većim usponima.

Unutar površine vučnog pasoša, čija je kontura dana navedenim graničnim crtama, može se ostvariti veliki broj radnih točaka (kombinacija vučna sila-brzina vožnje). Broj mogućih radnih točaka u vučnom pasošu ovisi o broju stupnjeva regulacije vučne sile i brzine što ih vučno vozilo ima ugrađeno.

Vučni pasoš električnih vozila razlikuje se od ostalih vučnih vozila po velikom broju stupnjeva regulacije, što omogućuje veliki izbor radnih točaka. Električna vozila također često omogućuju znatno preopterećenje vučnih motora u kratkotrajnim razdobljima, čime se još više proširuje područje izbora radnih točaka u vučnom pasošu.

Vučna vozila namijenjena vuči teških teretnih vlakova imaju veće vučne sile i manje maksimalne brzine, dok vučna vozila za brze putničke vlakove imaju nešto manje vučne sile, ali veće maksimalne brzine [7] (slika 2.5.).



Slika 2.5. Oblik vučnog pasoša za teretna i putnička vučna vozila

Sada slijedi kratak pregled vrsta lokomotiva u HŽ-u i ujedno je dana njihova karakteristika vučnog pasoša koja je modelirana u algoritmu za simulaciju kretanja vlaka (Borland C++).

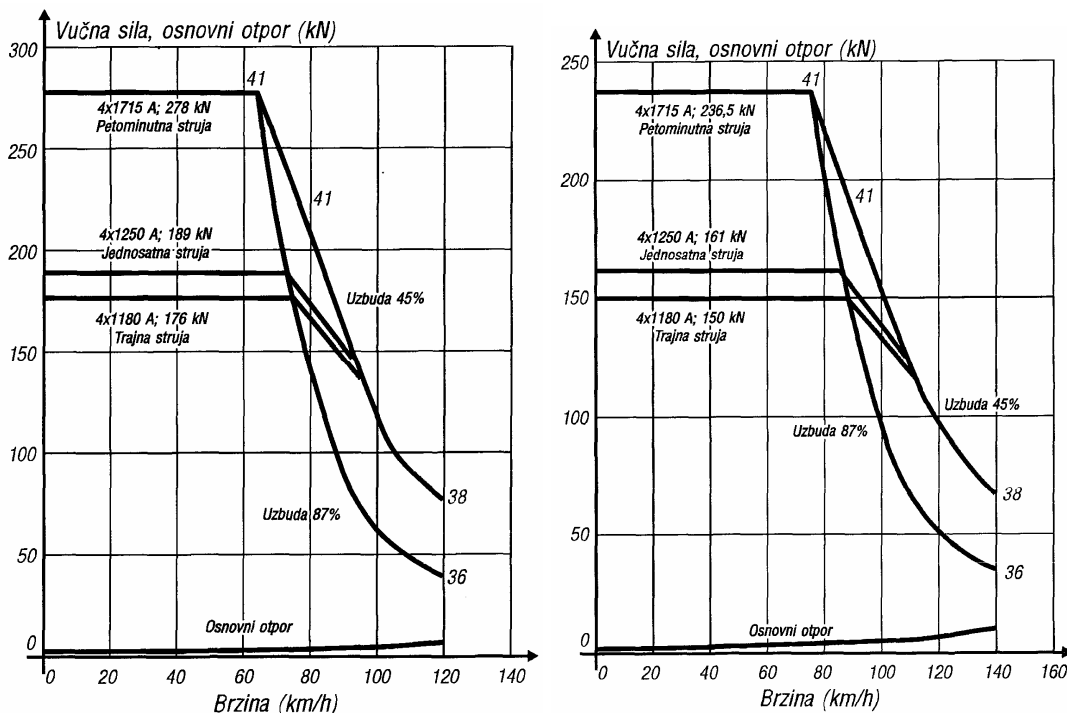
Hrvatske željeznice raspolažu dvjema vrstama električnih lokomotiva, i to:

- lokomotivama izmjeničnoga sustava 25 kV, 50 Hz
- lokomotivama istosmjernoga sustava 3 kV.

Najveći dio pruga u Hrvatskoj elektrificiran je izmjeničnim sustavom 25 kV, 50 Hz. U usporedbi s istosmjernim sustavom taj sustav je suvremeniji jer je razmak između podstanica za napajanje znatno dulji, a kontaktna mreža puno lakša. Hrvatske željeznice raspolažu trima serijama takvih lokomotiva (1141, 1142, 1161) i jednom serijom elektromotornog vlaka (6111).

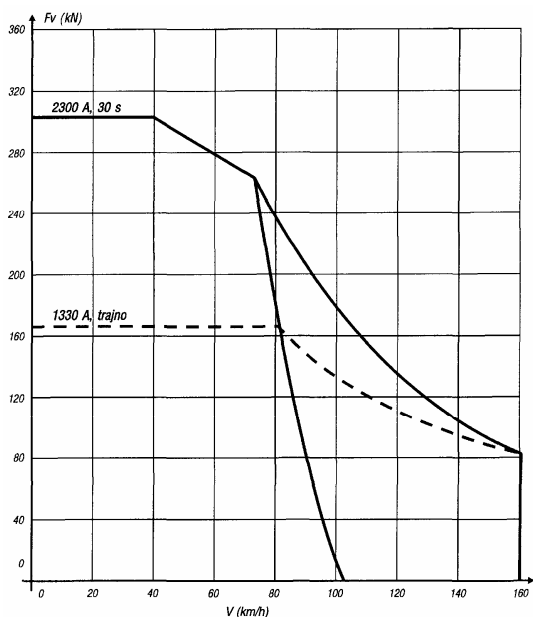
Električni istosmjerni sustav 3 kV postoji samo na malome dijelu pruge oko Rijeke. Taj sustav prilično je stariji od izmjeničnoga, a vučna vozila po izvedbi su jednostavnija. U istosmjernom sustavu 3 kV Hrvatske željeznice raspolažu jednom serijom lokomotiva i jednom serijom elektromotornog vlaka.

Lokomotive serije HŽ 1141 su diodne, četveroosovinske, s pojedinačnim osovinskim pogonom (osovinski raspored B'o-B'o) i namijenjene su za vuču putničkih i teretnih vlakova na nizinskim i brdskim prugama, a građene su za brzine od 120 i 140 km/h. Električna oprema konstruirana je za rad s jednofaznom izmjeničnom strujom nazivnoga napona 25 kV, 50 Hz. Lokomotive su sagrađene s električnom kočnicom ili bez nje, s uređajem za podmazivanje vijenaca kotača ili bez njega i s mogućnošću daljinskog upravljanja kod vožnje dviju lokomotiva. U svezi s tim osnovnim razlikama, kao i s razlikama zbog preinaka, lokomotive su razvrstane u podserijske skupine 000, 100, 200 i 300 (s tiristorskom regulacijom snage nastala pregradnjom podserije 000).



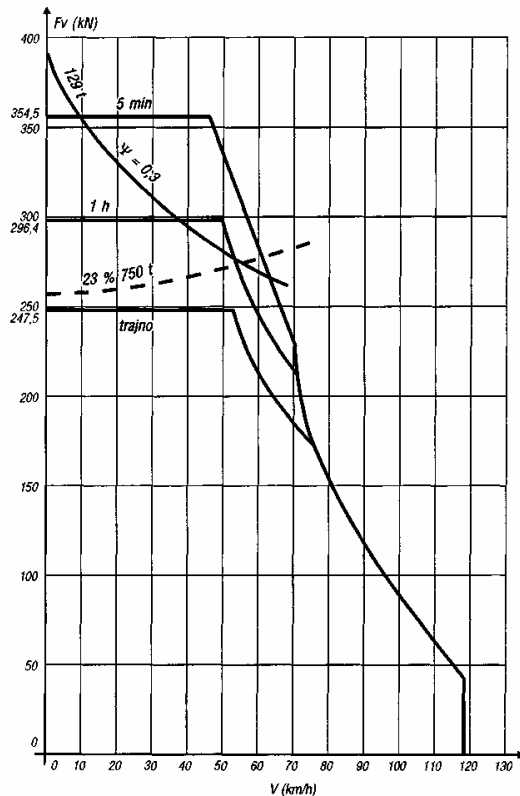
Slika 2.6. Vučne karakteristike lokomotiva serije HŽ 1141 za brzinu od 120 km/h i 140 km/h

Električne tiristorske lokomotive serije HŽ 1142 su četveroosovinske lokomotive s pojedinačnim osovinskim pogonom Bo' Bo', snage 4400 kW za napon napajanja 25 kV, 50 Hz. Lokomotive su namijenjene za vuču vlakova s prijevozom putnika brzinom do 160 km/h te za vuču teretnih vlakova pretežito na nizinskim prugama, ali i na brdskim prugama uz odgovarajući postupak. Glavni elektromotorni lokomotivski pogon je pogon reguliran četirima komutatorskim istosmjernim motorima s neovisnom uzbudom, koji su napajani preko tiristorskih ispravljača. Na slijedećoj slici je prikazan vučni pasoš za jednu tiristorsku lokomotivu.

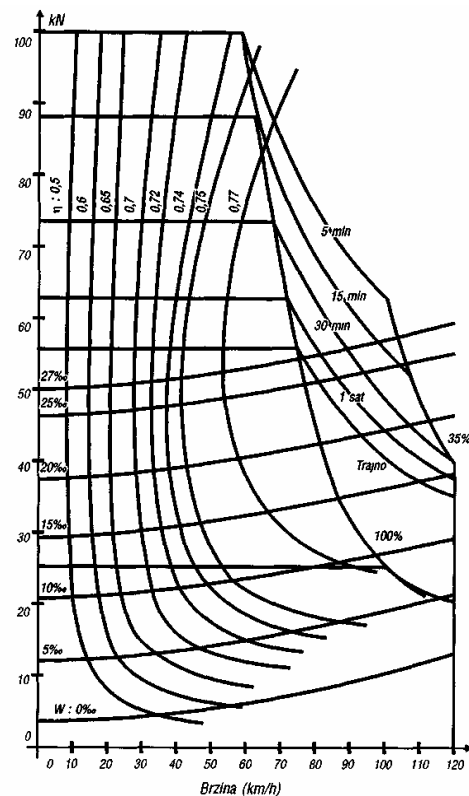


Slika 2.7. Vučne značajke lokomotive serije HŽ 1142-000 za najveću brzinu od 160 km/h pri naponu kontaktnog voda 22.5 kV

Električne lokomotive serije HŽ 1161 su šestosovinske lokomotive s pojedinačnim osovinskim pogonom Bo' Bo' Bo' za napon napajanja 25 kV, 50 Hz. Nastale su pretvorbom lokomotive serije HŽ 1061 za istosmjerni napon napajanja 3 kV. Rekonstrukcija je izvedena tako da su postojeća okretna postolja zadržana, a vučni motori dorađeni za rad s izmjeničnom strujom. U strojarnici je ugrađena identična ili slična oprema kakva se rabi na lokomotivama serije HŽ 1141. Vučni pasoš za taj tip lokomotive se nalazi na slijedećoj slici.



Slika 2.8. Vučne značajke lokomotive serije HŽ 1161 za napon kontaktnoga voda 22.5 kV i kada su kotači poluistrošeni



Slika 2.9. Vučne značajke elektromotornog vlaka serije HŽ 6111

Ostale značajke pojedinih tipova lokomotiva se nalaze u lit.[7].

2.2. Potrebna snaga za vuču vozila

Poznavajući potrebnu vučnu silu za ostvarenje kretanja moguće je izračunati mehaničku snagu na obodu kotača kao:

$$P_m = F_t \cdot v \quad W \quad (2.2.1)$$

gdje je sila izražena u N, a brzina u m/s.

Za izračunavanje djelatne snage koju vlak uzima iz mreže potrebno je poznavati faktor korisnosti (η) vučnog vozila koji ovisi o brzini kretanja vlaka i naponu mreže. U slučaju da nije poznata krivulja promjene faktora η može se pretpostaviti neka konstantna vrijednost

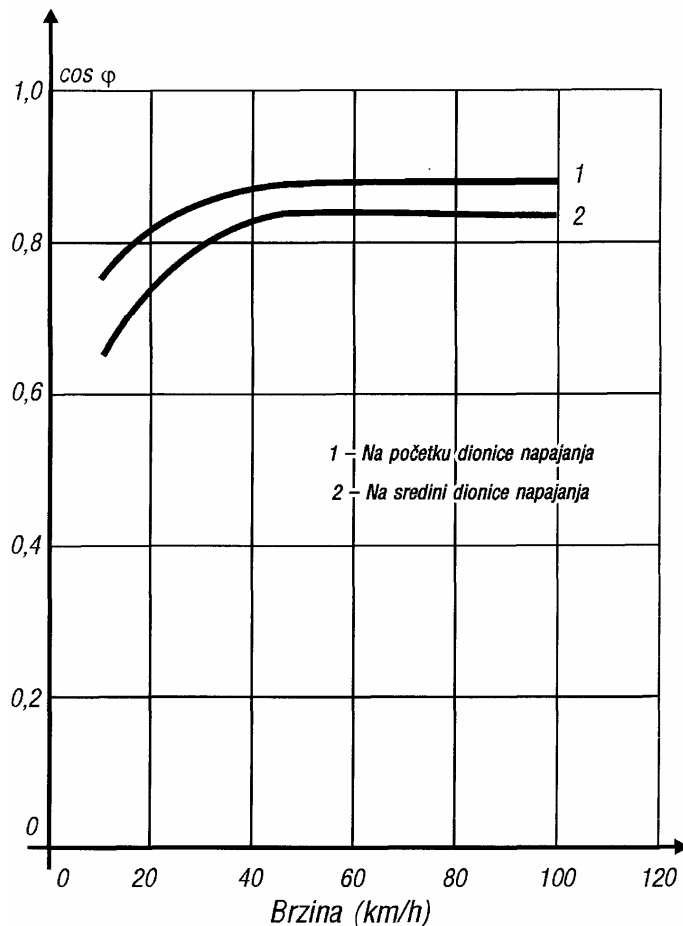
npr. 0.8 ili slična. Potrebno je još poznavati snagu pomoćnih pogona vlaka (hlađenje motora, grijanje vagona i sl.). Kad je to poznato moguće je odrediti djelatnu snagu koju vlak uzima iz mreže [7]:

$$P_{el} = \frac{P_m}{\eta} + P_{pom} \quad (2.2.2)$$

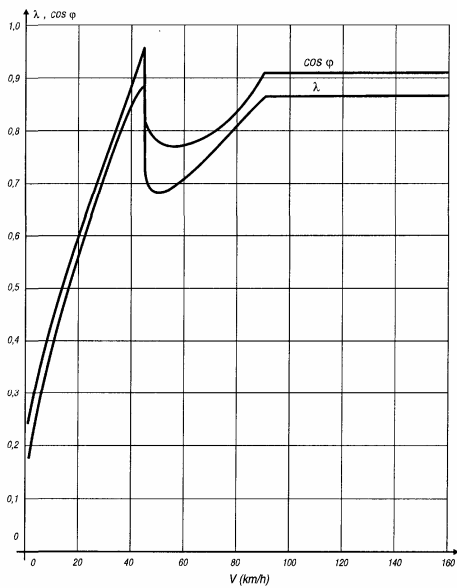
U slučaju izmjeničnog napajanja pored djelatne vlak uzima i jalovu snagu iz mreže. Jalova snaga se računa prema izrazu:

$$Q_{el} = \frac{P_m}{\eta} \cdot \tan(\arccos(\cos \varphi)) + \frac{P_{pom}}{\eta} \cdot \tan(\arccos(\cos \varphi_{pom})) \quad (2.2.3)$$

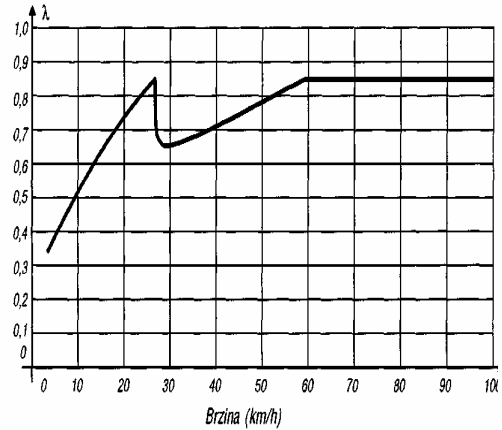
Faktor snage $\cos(\varphi)$ je ovisan o brzini kretanja vozila. Poznavanje faktora $\cos(\varphi)$ je naročito važno kod prometa sa čestim pokretanjem i zaustavljanjem jer je kod malih brzina faktor snage manji (slika 2.10.). Krivulja faktora snage je različita za pojedine tipove vlakova što se vidi iz slijedećih slika. Zbog udjela viših harmonika u mreži umjesto $\cos \varphi$ se koristi λ faktor.



Slika 2.10. Krivulja faktora snage $\cos \varphi$ lokomotiva serije HŽ 1141-000, 100 i 200

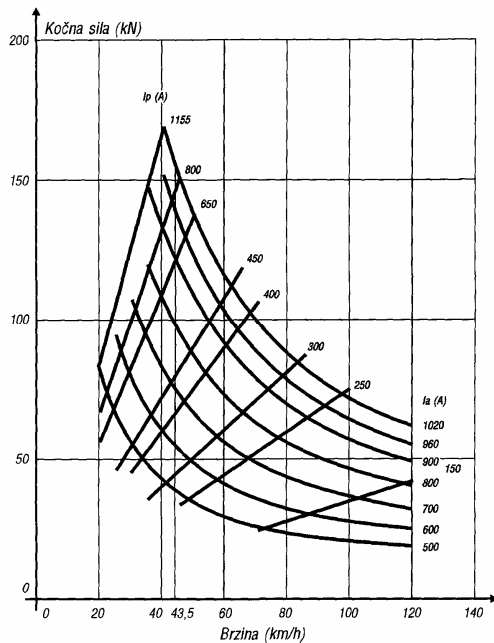


Slika 2.11 a). Krivulja faktora snage $\cos \varphi$ i λ lokomotive serije 1142

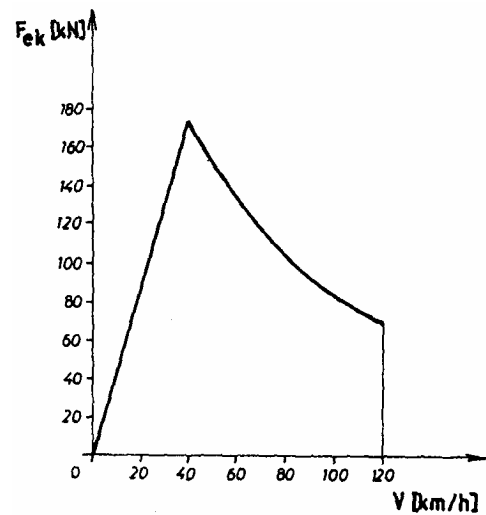


Slika 2.11 b). Krivulja ukupnog faktora λ elektromotornog vlaka serije HŽ 6111

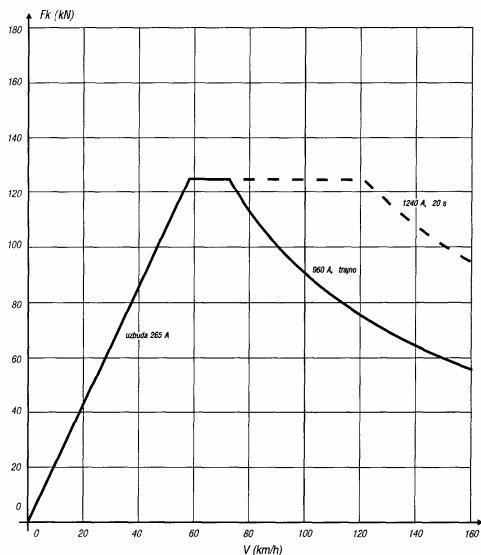
Za kočenje vlaka koristi se mehanička i električna kočnica. Ovisnost sile električne kočnice o brzini za pojedini tip lokomotive se zadaje na sličan način kao i vučna sila i ta ovisnost je prikazana na slijedećim slikama.



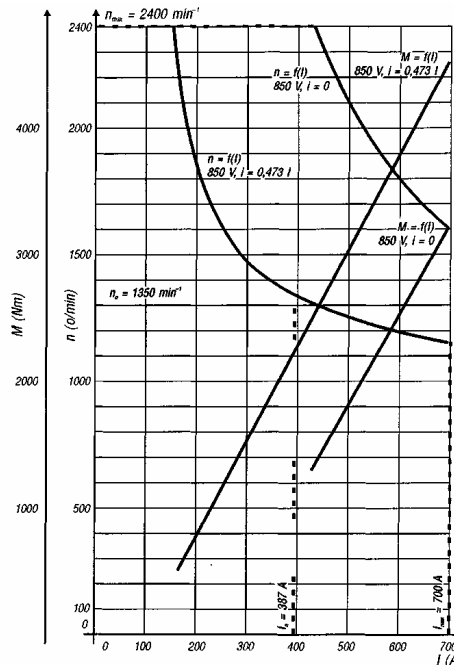
Slika 2.12. Krivulja elektrodinamičkoga kočenja lokomotive serije HŽ 1141-100



Slika 2.13. Krivulja elektrodinamičkoga kočenja diodne lokomotive korištena u algoritmu



Slika 2.14a) Krivulja elektrodinamičkog kočenja lokomotive serije HŽ 1142-000 za najveću brzinu od 160 km/h



Slika 2.14b). Krivulja elektrodinamičke kočnice elektromotornog vlaka serije HŽ 6111

Snaga kočnice se računa prema slijedećem izrazu:

$$P_{ek} = F_k \cdot v, \quad W \quad (2.2.4)$$

Ako je vučno vozilo predviđeno za rekuperaciju tada je moguće vratiti dio snage električne kočnice u mrežu. Faktor korisnosti električne kočnice kreće se od 0,66 – 0,85 ovisno o konstrukciji kočnice. U algoritmu je pretpostavljena vrijednost $\eta_k = 0,85$. Snagu je moguće vratiti u mrežu u slučaju da vlak pri kretanju koči el. kočnicom (na većim strminama) ili prilikom zaustavljanja jer u tim slučajevima je $P_m < 0$. Budući da prilikom kočenja mehanička snaga na obodu kotača ima negativni predznak u mrežu je moguće vratiti snagu ovisno da li je apsolutna vrijednost mehaničke snage veća ili manja od snage kočnice pa imamo dva slučaja:

1. Ako je P_m veće od P_{ek} u mrežu je moguće vratiti snagu:

$$P_e = P_{ek} \cdot \eta_k - P_{pom}, \quad W \quad (2.2.5)$$

2. Ako je P_m manje od P_{ek} u mrežu je moguće vratiti snagu:

$$P_e = |P_m| \cdot \eta_k - P_{pom}, \quad W \quad (2.2.6)$$

2.3. Matematički model za simulator kretanja vlakova

Jednadžba kretanja vlaka predstavlja matematičku međuovisnost ubrzanja vlaka i sila koje djeluju na vlak [3]. Pri izvođenju jednadžbe polazi se od pretpostavke da je vlak kruto tijelo čija je masa koncentrirana u njegovom težištu. U tom slučaju vrijedi:

$$\begin{aligned} dE_k &= dA \\ dE_k &= m_{vl} \cdot (1 + \varepsilon) \cdot v \cdot dv \\ dA &= \left(F_t - \sum F_w \right) \cdot ds \end{aligned} \quad (2.3.2)$$

gdje je: ε - koeficijent korekcije mase zbog rotacije

F_t - vučna sila vlaka

$\sum F_w$ - suma otpora vožnje

Uvrste li se izrazi za dE_k i dA dobit će se: $m_{vl} \cdot (1 + \varepsilon) \cdot \frac{ds}{dt} \cdot \frac{dv}{ds} = F_t - \sum F_w$

Iz ovog izraza slijede jednadžbe kretanja vlaka:

$$a = \frac{dv}{dt} = \frac{F_t - \sum F_w}{m_{vl} \cdot (1 + \varepsilon)}, \quad \text{m/s}^2 \quad (2.3.3)$$

$$v = \int a \cdot dt = \int \frac{F_t - \sum F_w}{m_{vl} \cdot (1 + \varepsilon)} \cdot dt, \quad \text{m/s} \quad (2.3.4)$$

$$s = \int v \cdot dt = \int dt \cdot \int \frac{F_t - \sum F_w}{m_{vl} \cdot (1 + \varepsilon)} \cdot dt, \quad \text{m} \quad (2.3.5)$$

Za analitičko rješavanje dobivenih jednadžbi kretanja potrebno je poznavati funkcije $F_t(t)$ i $\sum F_w(t)$. To je velika poteškoća, jer te ovisnosti nisu unaprijed poznate, pa se rješenje pronalazi računanjem sa konačnim priraštajima Δt unutar kojih je moguće pretpostaviti konstantne sile što djeluju na vlak. Dakle, vrši se približno integriranje tj. zbrajanje konačnih priraštaja (diferencijala) brzine udaljenosti. Pa tako umjesto izraza:

$$dv = a \cdot dt = \frac{F_t - \sum F_w}{m_{vl} \cdot (1 + \varepsilon)} \cdot dt \quad (2.3.6)$$

račun se provodi sa:

$$\Delta v = a \cdot \Delta t = \frac{F_t - \sum F_w}{m_{vl} \cdot (1 + \varepsilon)} \cdot \Delta t \quad (2.3.7)$$

U ovom slučaju je potrebno izabrati dovoljno mali interval Δt u kojem će prirast brzine ($v_2 - v_1$) biti toliko malen da su sile koje djeluju na vlak mogu smatrati približno konstantnim (unutar razdoblja od maksimalno par sekundi). To znači da je akceleracija tijekom intervala konstantna pa se prirast puta može računati prema:

$$\Delta s = v_0 \cdot \Delta t + \frac{a \cdot (\Delta t)^2}{2} \quad (2.3.8)$$

Gdje je v_0 brzina na početku intervala.

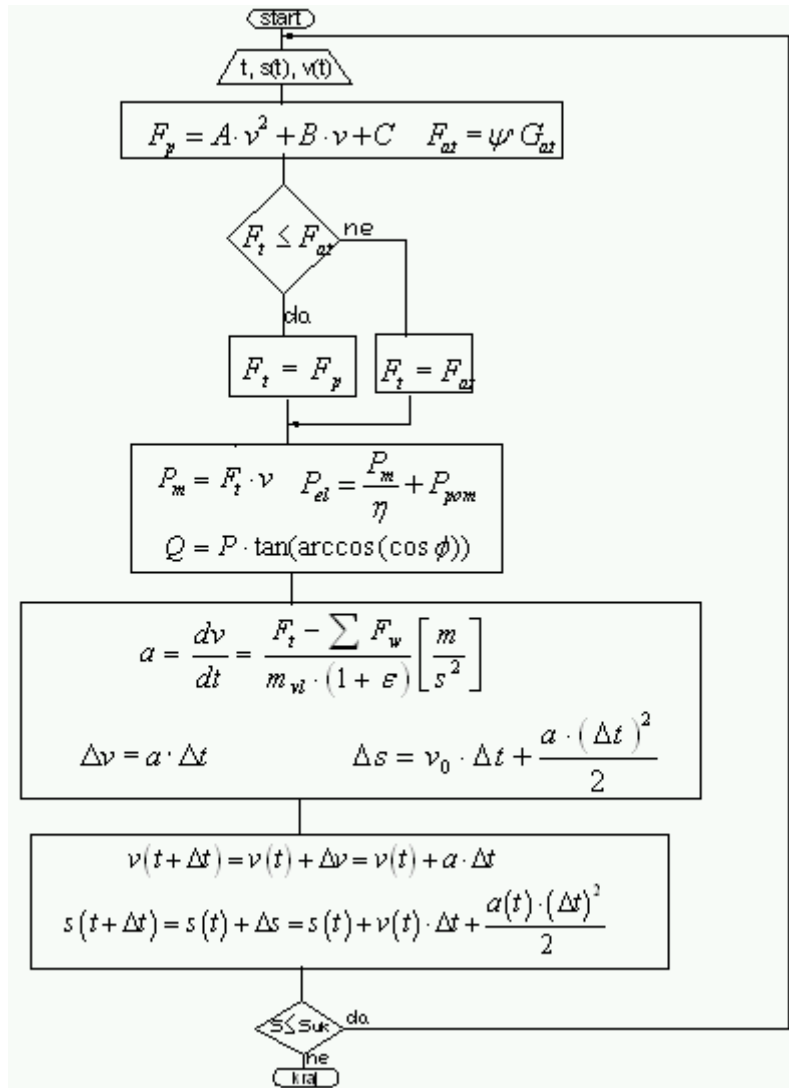
Neka su a , v i s vrijednosti koje vlak ima u trenutku t . Vrijednosti brzine i prijednog puta nakon intervala Δt su:

$$v(t + \Delta t) = v(t) + \Delta v = v(t) + a \cdot \Delta t \quad (2.3.9)$$

$$s(t + \Delta t) = s(t) + \Delta s = s(t) + v(t) \cdot \Delta t + \frac{a(t) \cdot (\Delta t)^2}{2} \quad (2.3.10)$$

Što je interval Δt manji, manja je i greška koja se neminovno čini jer se umjesto integriranja provodi zbrajanje sa konačnim vrijednostima. U slučaju električne vuče (zbog velike inercije) promjena brzine, a s njom i F_t te $\sum F_w$ su polagane pa je interval proračuna reda par sekundi sasvim prihvatljiv.

Na kraju se ukratko može opisati dijagram toka za simulator kretanja vlakova (slika 2.15.)



Slika 2.15. Dijagram toka za simulator kretanja vlakova

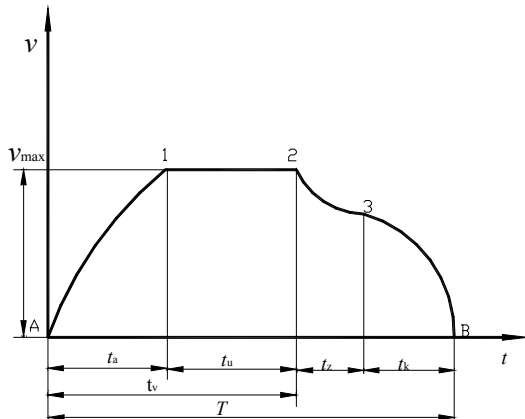
Ako je u trenutku t poznata brzina vlaka i njegov položaj na pruzi moguće je izračunati otpore vožnje, adhezionu silu te moguću vučnu silu lokomotive. Kad je sve to poznato izračuna se vučna sila koja ovisi o režimu vožnje vlaka. Na osnovi poznate vučne sile i brzine određuje se mehanička snaga te djelatna i reaktivna električna snaga za trenutak t . Nakon toga računa se akceleracija (ako ubrzava) vlaka, a zatim prirasti brzine i puta. Kad su određeni brzina i položaj vlaka u trenutku $t+\Delta t$ moguće je odrediti nove vrijednosti sila i snaga potrebnih za kretanje. Postupak se dalje ponavlja za naredni vremenski interval - korak proračuna dok se ne dođe do kraja ukupnog puta.

2.4. Režimi vožnje vlaka

Tijekom vožnje vlak se može kretati na različite načine ovisno o vučnoj sili i trenutnoj akceleraciji.

Karakteristična su četiri načina vožnje i to:

- Ubrzavanje,
- vožnja konstantnom brzinom,
- vožnja sa zaletom,
- kočenje .



Slika 2.16. Dijagram ovisnosti brzine o vremenu na kojem se vide sva četiri režima vožnje vlaka

- t_a vrijeme ubrzanja,
 t_u vrijeme ustaljene vožnje,
 t_e vrijeme vožnje sa zaletom,
 t_k vrijeme kočenja,
 t_v vrijeme aktivne vuče.

Ovisno o režimu vožnje vlaka, mijenjat će se potrebna vučna sila, a s njom i snaga koju vlak vuče iz mreže. Prva dva režima su režimi aktivne vuče jer su F i $P > 0$.

Za **vožnju s konstantnom brzinom** vrijede slijedeći izrazi [8]:

$$v = \text{konst.} \quad a = 0 \quad \Delta s = v \cdot \Delta t . \quad (2.4.1)$$

Vučna sila tijekom vožnje konstantne brzine je: $F_t = \sum F_w . \quad (2.4.2)$

U ovom slučaju vučna sila ovisi samo o otporima vožnje tj. profilu pruge i brzini vlaka. U algoritmu potprograma za simulaciju kretanja vlaka je predviđeno da vlak vozi konstantnom brzinom od trenutka kad dosegne planiranu brzinu za tu dionicu.

Za **ubrzanje** vrijede slijedeći izrazi:

$$v \neq \text{konst.} \quad a > 0 \quad F_t = m_{vl} \cdot (1 + \varepsilon) \cdot a + \sum F_w \quad (2.4.3)$$

Prirasti brzine i puta su: $\Delta v = a \cdot \Delta t$ i $\Delta s = v \cdot \Delta t + \frac{a \cdot (\Delta t)^2}{2} \quad (2.4.4)$

U odnosu na vožnju konstantnom brzinom potrebna je mnogo veća vučna sila (za isti profil pruge). U programu je početak ubrzavanja definiran na način da ako je $v_{\text{plan}}(i) > v$, i ako vlak u tom trenutku ne koči niti vozi sa zaletom, slijedi da vlak treba ubrzavati. Ubrzavanje će trajati sve dok se ne postigne planirana brzina na toj dionici ili do nailaska na mjesto od kojeg je potrebno započeti kočiti. Vlak u cijelom dijelu ubrzava s akceleracijom od $a = 0,5 \text{ m/s}^2$ ako može (tj. ako u tom trenutku vučno vozilo raspolaže s dovoljno vučne sile i sila ubrzavanja ne premašuje adhezionu silu). Inače vlak ubrzava iznosom akceleracije koju u tom trenutku vučno vozilo može ostvariti.

Za vožnju sa zaletom vrijedi:

$$F_t = 0 \quad a = \frac{-\sum F_w}{m_{\text{vl}} \cdot (1 + \varepsilon)} \quad (2.4.5)$$

U ovom slučaju vučna sila je jednaka 0, a vlak se kreće pod djelovanjem sila otpora vožnje. Ovisno o $\sum F_w$ vlak može ubrzavati ili usporavati. Mehanička snaga potrebna za ostvarenje ovakvog kretanja jednaka je 0, a vlak iz mreže uzima samo snagu koju zahtijevaju pomoćni pogoni.

U algoritmu potprograma za simulaciju kretanja vlaka se vožnja sa zaletom mora definirati pomoću varijable u ulaznoj datoteci (0 - vožnja sa zaletom nije dozvoljena ili 1 - vožnja sa zaletom je dozvoljena). Da bi vlak počeo voziti sa zaletom potrebno je prvo da vlak dosegne brzinu koja je planirana za tu dionicu. Tek nakon što postigne tu brzinu, bit će moguće voziti i sa zaletom. U algoritmu je predviđena vožnja sa zaletom sve do trenutka kad brzina padne na 80% planirane brzine ili (u slučaju da se vlak pod djelovanjem otpora vožnje ubrzava) do trenutka kad brzina dosegne planiranu. Naravno, moguće je da vlak i prije tog trenutka prekine takvu vožnju ovisno o tome je li vlak treba ubrzavati ili kočiti.

Za kočenje vrijede slijedeći izrazi:

$$a < 0, \quad F_b = a \cdot m_{\text{vl}}(1 + \varepsilon) + \sum F_w . \quad (2.4.6)$$

U ovom režimu vožnje vlak se usporava konstantnom akceleracijom a_{koc} na putu kočenja s_{koc} :

$$a_{\text{koc}} = -\frac{v^2}{2 \cdot s_{\text{koc}}} \quad (2.4.7)$$

U algoritmu potprograma za simulaciju kretanja vlaka su predviđena dva slučaja kad treba kočiti. Prvi slučaj je onaj kad se vlak treba zaustaviti na kraju neke dionice, a drugi nastupa onda kad je potrebno smanjiti brzinu, jer je planirana brzina na slijedećoj dionici manja od brzine kojom se vlak kreće. U oba slučaja koristi se akceleracija kočenja od: $a_{\text{koc}} = -0,5 \text{ m/s}^2$. Dakle, kad vlak dođe do mjesta na kojem je potrebno započeti sa kočenjem (program na svakoj dionici izračuna trenutak kada je potrebno izračunati početno mjesto kočenja), vlak počne kočiti s navedenom akceleracijom. Put kočenja se računa kao:

$$s_{\text{koc}} = \frac{v_{\text{plan}}^2(i+1) - v_{\text{plan}}^2(i)}{2 \cdot a_{\text{koc}}}, \text{ m} \quad (2.4.8)$$

U drugom slučaju, kad je potrebno smanjiti brzinu, na početku programa vrše se uspoređivanja planiranih brzina po dionicama te ako je ispunjen uvjet:

$v_{\text{plan}}(i) > v_{\text{plan}}(i+1)$ to je znak da je pri kraju i -te dionice potrebno kočiti da bi vlak na početku sljedeće dionice imao planiranu brzinu. Kočenje se odvija jednakim principom kao i u prvom slučaju, jedina je razlika da na kraju te dionice vlak ne staje već nastavlja vožnju planiranom brzinom na toj dionici. Pretpostavka u algoritmu je da cijeli vlak (vučno vozilo i vučena vozila-vagoni) raspolažu sa dovoljnom silom za kočenje cijelog vlaka.

2.5. Programska podrška simulacije električne vuče

Algoritam za *simulaciju električne vuče* je napisan u programskim jezicima FORTRAN [6] i Borland C++ [5]. To je algoritam za simulator kretanja vlaka po pruzi. Za zadanu dionicu pruge i zadani vlak računa položaje i snagu koju vlak uzima iz mreže. Vremenski korak proračuna zadaje korisnik prije početka proračuna. U lit. [17] je ukratko dan opis simulatora električne vuče.

Fortranski algoritam se nalazi u sklopu potprograma TMS (ispis se nalazi u prilogu 1.) Potprogram je napisan da može simulirati sva četiri moguća režima vožnje vlaka (ubrzavanje, vožnja konstantnom brzinom, vožnja sa zaletom i kočenje). Kod vožnje sa zaletom vučna sila je jednaka 0, a vlak se kreće pod djelovanjem sila otpora vožnje tj. mehanička snaga je jednaka 0, a vlak iz mreže uzima samo snagu koju zahtijevaju pomoćni pogoni u vlaku (rasvjeta, grijanje, ...). Ako je predviđeno da vlak na i -toj dionici treba voziti sa zaletom onda prilikom upisa varijabli $zal(i)$ treba dodijeliti vrijednost 1. Ako je $zal(i) = 0$ to znači da na i -toj dionici vlak ne treba voziti sa zaletom. Napravljene su određene preinake u vezi ispisa rezultata u formi izlazne datoteke. Za izvršavanje programa potrebno je imati određene datoteke u kojima se nalaze podaci o profilu pruge, podaci o vlaku, podaci o načinu vožnje, podaci o vučnom vozilu, podaci o koeficijentima za proračun stalnih otpora vožnje itd. Primjer jedne takve ulazne datoteke je prikazan na slici 2.17.

```
* Podaci o vrsti vlaka
* Sifra vlaka, vrsta lokom,tezina vlaka( t), Eps, vvl, P_pom(kw),cos_fip,nlok
* pocetna udaljenost od EVP u (m),smjer kretanja, pocetno i krajnje vrijeme
05102 442 750 0.07 1 250 0.9 2
-12000 1 23 21 23 29
*
endt
*
* Podaci o profilu pruge i voznom redu vlaka
* duljina dionice(km), uspon u (1/1000), radijus krivine (m), planirana
* (.u istom redu) brzina(km/h), stati na kraju dionice(0-1), zalet (0-1)
* Skrad - Zalesina
5.440 17.23 0 58 0 0
0.551 -1.67 0 58 0 0
0.909 1.93 0 58 1 0
* Zalesina
*
enddd
```

Slika 2.17. Primjer ulazne datoteke za potprogram simulacije kretanja vlakova (Fortran)

U ovom algoritmu je pretpostavljeno da je vlak vuče lokomotiva 1142. Vučni pasoš lokomotive je zadan na sljedeći način:

$$\text{- za brzine } v < v_{\text{konst}} \Rightarrow F_{\text{lok}} = F_{v_{\text{max}}} \quad (F_{v_{\text{max}}} = 226 \text{ kN})$$

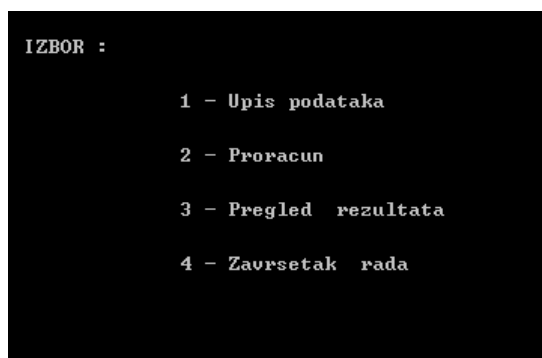
$$\text{- za brzine } v_{\text{kons}} < v < v_{\text{max}} \Rightarrow F_{\text{lok}} = k_0 + k_1 \cdot v + k_2 \cdot v^2 + k_3 \cdot v^3 + k_4 \cdot v^4$$

Može se zaključiti da je krivulja $F_{lok}(v)$ za brzine veće od v_{kons} (pretpostavljeno je $v_{konst} = 60$ km/h) aproksimirana polinomom 4. stupnja pri čemu su iznosi faktora slijedeći: ($k_0 = 821,14$, $k_1 = -19,4161$, $k_2 = 0,22488$, $k_3 = -0,001278$, $k_4 = 2,8524 \cdot 10^{-6}$).

Faktor k (za proračun specifičnog otpora vučenih vozila u izrazu 2.1.11.) iznosi 0.032.

Krivulje faktora snage $\cos\varphi(v)$, odnosno faktora $\lambda(v)$ koja je korištena u ovom algoritmu se nalaze na slici 2.11. Ta krivulja je modelirana tako da je po odsječcima aproksimirana pravicima.

Prilikom izvršavanja ovog potprograma pojavljuje se glavni prozor prikazan na slici 2.18. Izbor se vrši pritiskom na broj željene opcije.



Slika 2.18. Prikaz glavnog meni-a prilikom izvršavanja TMS potprograma

Rezultat izvršenja ovog potprograma je izlazna datoteka koja sadrži parametre potrebne sa slijedeće proračune (formiranje električne mreže i proračun električnih prilika). Primjer jedne takve izlazne datoteke se nalazi na slici 2.19.

```
5102 23 21 23 29
      0 -12000 0.250 0.121
      2 -12000 0.405 1.040
      4 -11999 0.560 1.712
      6 -11998 0.716 2.229
      8 -11996 0.871 2.642
     10 -11993 1.026 2.980
     12 -11990 1.181 3.264
     14 -11987 1.336 3.507
     16 -11983 1.491 3.717
     18 -11979 1.646 3.900
     20 -11974 1.801 4.062
     22 -11968 1.955 4.206
     24 -11962 2.110 4.334
     26 -11955 2.264 4.449
     28 -11948 2.419 4.551
     30 -11941 2.573 4.643
     32 -11932 2.727 4.724
     34 -11924 2.881 4.797
     36 -11914 3.034 4.861
     38 -11905 3.188 4.916
     40 -11894 3.341 4.963
     42 -11884 3.494 5.003
     44 -11872 3.647 5.036
     46 -11860 3.799 5.061
     48 -11848 3.952 5.079
     50 -11835 4.104 5.090
     52 -11822 4.256 5.094
     54 -11808 4.407 5.091
     56 -11793 4.559 5.081
     58 -11779 4.710 5.065
     60 -11763 4.860 5.043
```

Slika 2.19. Prvi dio izlazne datoteke tr05102.isl dobivene u Fortranu

Iz slike 2.19. se vidi da: za vlak koji u redu vožnje ima internu šifru 5102 kao rezultat njegove simulacije dobije se datoteka tr05102.isl. U toj datoteci su podaci o vremenu polaska i vremenu dolaska tog vlaka, te sukcesivno između ta dva vremena podaci o

rastojanju vlaka od neke referentne točke (referentnog EVP-a), te djelatna i jalova snaga vlaka. Ulazni podaci potrebni za takav proračun su parametri profila pruga, planirane brzine kretanja na pojedinim dionicama, te karakteristike vlaka i lokomotive.

Na osnovi rezultata dobivenih u izlaznoj datoteci imamo grafički i numerički prikaz rezultata sljedećih veličina:

- v , km/h – brzina vlaka u određenom trenutku,
- s , m – put koji vlak prevali do određenog trenutka,
- P_m , MW – mehanička snaga potrebna za kretanje vlaka,
- P_e , MW – električna djelatna snaga koju vlak uzima iz mreže,
- Q_e , MVar – električna jalova snaga koju vlak uzima iz mreže,
- F_i – faktor snage vučnog vozila $\cos\varphi$.

Sve ove veličine su grafički prikazane u ovisnosti o vremenu t , s. Numerički je još prikazana i akceleracija vlaka – a , m/s^2 , koja može biti pozitivna, da bi vlak ubrzao do maksimalne brzine na pojedinoj dionici, ali i negativna pri kočenju.

Mogućnost grafičkog prikazivanja prometa vlakova praktički pojednostavljuje mnoga složena pitanja organizacije željezničkog prometa. Budući da je promet vlakova određen dvjema dimenzijama – vremenom i prostorom, to svladavanje tih odrednica daje rezultantu, u stvari projekciju vremena i prostora. Ta projekcija praktički je trasa vlaka. Prema tome, vrijeme i prostor mogu se predstaviti u koordinatnom sustavu, u koji se ucrtavaju trase vlakova. Apscisa je podijeljena na vrijeme od 0 do 24 sata i to debljim crtama satna podjela, a tanjim crtama minutna podjela sata na vremena po 10 minuta. Pri tome je uobičajen razmjer za 1 sat i 30 min. Dužina pruge predstavljena je s ordinatom. Na ordinati se prema udaljenosti, nanose službena mjesta. Brzina, odnosno vrijeme putovanja, dana je veličinom kuta što ga zatvara apscisa s trasom vlaka. Znači manji kut, manja je i brzina i veće vrijeme vožnje, tj. vlakovi s većim brzinama imaju okomitije trase od onih vlakova s manjim brzinama (slika 4.1). Kada vlak nema bavljenje u međustanicama, tj. kada te stanice prolazi bez zadržavanja, onda se radi o tzv. čistom vremenu vožnje (t). To je vrijeme u direktnoj ovisnosti o tehničkoj brzini vlaka. Međutim, ako se vlakovi zadržavaju u međustanicama tada su potrebna dodatna vremena: za pokretanje i za zaustavljanje koja su sadržana u vremenima vožnje. Prema tome, trase vlakova ne lome se u međustanicama već se radi o njihovom toku po crti apscise.

Iako ovaj potprogram daje očekivane rezultate, nedostatak mu je što se izvršava u DOS sučelju i što omogućuje simulaciju samo određenog tipa lokomotive (1142). Budući da ova pruga Rijeka – Moravice nema baš uvjete za rekuperaciju (povrat snage u mrežu), u ovom algoritmu nije predviđen proračun povrata snage u mrežu. Također su krivulje faktora snage ($\cos\varphi$) i λ (zbog viših harmonika) kao i *vučnog pasoša* jednoznačno određene za taj tip lokomotive.

Zato se težilo preglednijem algoritmu koji će podržati windows sučelje i u tu svrhu je napisan algoritam u Borland C++ programu koji omogućuje raznovrsniji izbor ulaznih podataka. Ispis koda programa je izložen u prilogu 1. Algoritam je napravljen za nazivne karakteristike električnih lokomotiva tipa 1141, 1142, 1161 te elektromotorni vlak tipa 6111. Posebna se pažnja obratila na modeliranje vučnog pasoša (ovisnost vučne sile o brzini vučnog vozila), ovisnosti kočne sile o brzini vučnog vozila, te faktora snage o brzini vučnog vozila. Primjer ulazne datoteke za ovaj algoritam je prikazan na slici 2.20.

5102	23	21	23	29	-12000	1
Redni broj dionice	Duljina dionice (m)	Radius krivine (m)	Uspon (‰)	Planirana brzina	Dozvola vožnje zaletom	
1	5440	0	17.23	58	0	
2	551	0	-1.67	58	0	
3	909	0	1.93	58	1	

Slika 2.20. Primjer ulazne datoteke za potprogram simulacije kretanja vlakova (Borland)

Redni broj dionice ide od 1 pa do zadnjeg rednog broja dionice. Program prepoznaje tu vrijednost kao integer u glavnom programu i pomoću te varijable uspoređuje brzine i određuje kako vlak treba voziti. Duljina dionice se unosi u metrima, a program prepoznaje tu vrijednost kao float. Polumjer krivine se također unosi u metrima, a ta vrijednost je potrebna za računanje otpora vožnja. Program prepoznaje tu vrijednost kao float. Uspon se unosi u datoteku brojem, a predstavlja uspon u promilima na određenom dijelu pruge. Program prepoznaje tu vrijednost kao float i koristi je za računanje otpora vožnje. Uspon može imati negativan ili pozitivan predznak ovisno je li vlak vozi nizbrdo ili uzbrdo. Planirana brzina se unosi u km/h i ona predstavlja programu varijablu tipa float. Pomoću nje program računa do koje se brzine treba ubrzati ili zakočiti.

Dozvola vožnje zaletom označava je li vožnja sa zaletom moguća ili ne. U datoteku unosimo 1 ako je vožnja sa zaletom moguća i 0 ako nije.

Da bi algoritam ispravno radio, potrebno je bilo modelirati krivulje prikazane slikama u potpoglavlju 2.2. i 2.3. Modeliranje je bilo napravljeno u programu Mathematica 4.0, na način da su prije korištenja programa očitane pojedine vrijednosti sa karakteristike i date kao koordinate kroz koje treba prolaziti polinom kojim se aproksimira prava krivulja. Polinom se jednostavno dobije korištenjem naredbi: `Expand[Simplify[InterpolatingPolynomial[{...}]]]`. Sve su krivulje modelirane za trajnu vuču tj. trajni režim rada.

Tako je vučni pasoš za lokomotivu serije HŽ 1141 za brzinu od 140 km/h (slika 2.6. desno) modeliran na slijedeći način:

Brzina v, km/h:	Sila, kN:
00.00 – 88.57	150
88.57 – 110.00	- 1.41657· v + 275.465
110.00 – 140.00	- 0.000595· v^3 + 0.24991· v^2 - 36.1792· v + 1867.39

Krivulja faktora snage za diodnu lokomotivu (slika 2.10., krivulja 1) je modelirana kao:

Brzina v, km/h:	$\cos\phi$:
00.00 – 60.00	- 2·10 ⁻⁹ · v^5 + 3.70833·10 ⁻⁷ · v^4 - 0.0000245833· v^3 + 0.000602917· v^2 + 0.00167833· v + 0.646
60.00 – 140.00	0.858

Krivulja elektrodinamičkog kočenja na slici 2.13. je modelirana pomoću slijedećih polinoma:

Brzina v, km/h:	Sila, kN:
00.00 – 40.00	4.4· v +
40.00 – 140.00	+ 0.0114· v^2 - 3.1486· v + 283.6

Za tiristorske lokootive tipa 1142 vučni pasoš (slika 2.7.) je modeliran na slijedeći način:

Brzina v, km/h:	Sila, kN:
00.00 – 81.25	166.25
81.25 – 160.00	4.4735·10 ⁻¹² · v^8 - 4.37454·10 ⁻⁹ · v^7 + 1.85933·10 ⁻⁶ · v^6 - 0.000448607· v^5 + 0.0671942· v^4 - 6.39767· v^3 + 378.129· v^2 - 12687.2· v + 185274

Faktor λ za tiristorske lokomotive (slika 2.11.) je modeliran pomoću slijedećih polinoma:

Brzina v , km/h:	λ :
00.00 – 45.00	$-3.0582 \cdot 10^{-9} \cdot v^5 + 1.89153 \cdot 10^{-7} \cdot v^4 + 1.46296 \cdot 10^{-6} \cdot v^3 - 0.000410423 \cdot v^2 + 0.0272994 \cdot v + 0.141$
45.00 – 45.67	$-0.247761 \cdot v + 12.0323$
45.67 – 90.00	$-1.07016 \cdot 10^{-8} \cdot v^5 + 3.8664 \cdot 10^{-6} \cdot v^4 - 0.000554279 \cdot v^3 + 0.0394308 \cdot v^2 - 1.38768 \cdot v + 19.9539$
90.00 – 160.00	0.866

Karakteristika kočne sile za tiristorsku lokomotivu za trajni režim rada (slika 2.12.) je modelirana pomoću slijedećih dvaju polinoma:

Brzina v , km/h:	Sila, kN:
00.00 – 70.00	$1.8722 \cdot v +$
70.00 – 160.00	$+0.0059 \cdot v^2 - 2.2241 \cdot v + 257$

Modeliranje vučnih pasoša i krivulja faktora snage kao i elektrodinamičke kočnice za ostale tipove lokomotiva se nalazi u lit.[5].

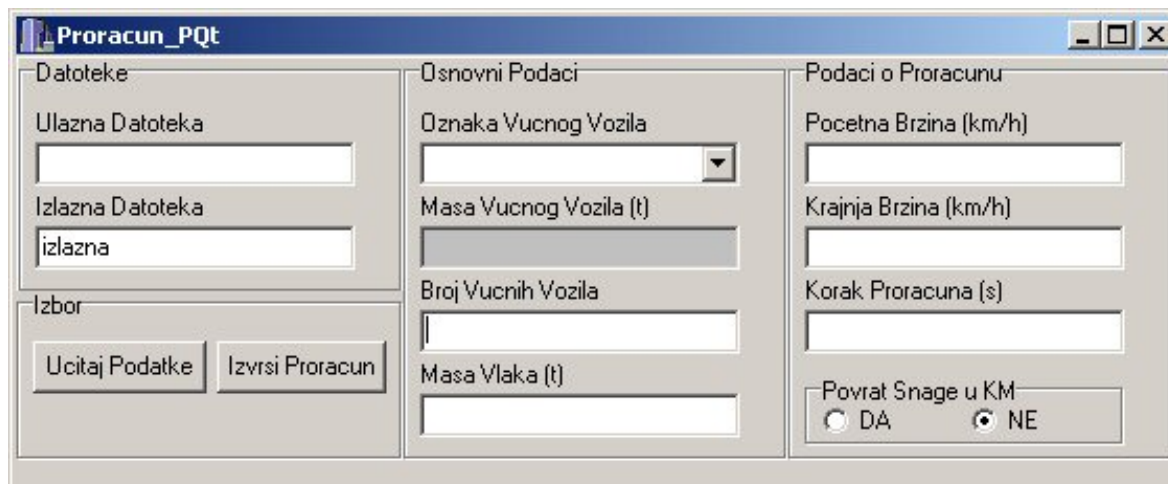
Rezultat ovog potprograma je izlazna datoteka s ekstenzijom «.izl» koja se koristi kao ulazni parametar za drugi potprogram, a prvi dio jedne takve izlazne datoteke se nalazi na slici 2.21. Može se zaključiti da su izlazne datoteke za oba algoritma istog formata.

5102	23	21	23	29	
	0	-12000		0.250	0.121
	2	-12000		0.435	1.098
	4	-11998		0.621	1.693
	6	-11996		0.806	2.101
	8	-11992		0.991	2.402
	10	-11988		1.176	2.635
	12	-11983		1.361	2.821
	14	-11977		1.546	2.972
	16	-11970		1.731	3.096
	18	-11962		1.915	3.197
	20	-11953		2.100	3.278
	22	-11943		2.284	3.341
	24	-11932		2.468	3.388
	26	-11920		2.651	3.419
	28	-11907		2.835	3.434
	30	-11894		3.018	3.434
	32	-11879		3.200	3.420
	34	-11863		3.383	3.392
	36	-11847		3.565	3.349
	38	-11830		3.746	3.294
	40	-11811		3.927	3.228
	42	-11792		4.108	3.151
	44	-11772		4.288	3.068
	46	-11751		4.468	2.980
	48	-11729		4.648	2.893
	50	-11706		4.826	2.813
	52	-11682		5.005	2.750
	54	-11657		5.182	4.002
	56	-11631		5.359	5.321
	58	-11605		5.536	5.689
	60	-11577		5.712	5.977

Slika 2.21. Prvi dio izlazne datoteke tr05102.isl dobivene u Borland C++

Ako se sadržaj ove datoteke usporedi sa sadržajem izlazne datoteke (slika 2.9.) može se zaključiti da se podaci približno podudaraju za oba algoritma, a rješenje pomoću Borland-a je mnogo praktičnije jer omogućuje windows sučelje.

Program se poziva dvoklikom na ikonu programa. Na ekranu će se pojaviti glavni prozor programa koji izgleda kao na slici 2.22.



Slika 2.22. Prikaz glavnog prozora potprograma za simulaciju električne vuče u Borland C++

Glavni prozor se sastoji od slijedećih potprozora:

- datoteke (ulazna i izlazna),
- izbor,
- osnovni podaci,
- podaci o proračunu.

U potprozoru "Izbor" klikom na "Ucitaj Podatke" otvara se standardni Windows prozor za "traženje" datoteke. U polje za izlaznu datoteku možemo upisati naziv izlazne datoteke koja se ispisuje u isti direktorij gdje je locirana i ulazna datoteka. Kao što se vidi pretpostavljeno ime izlazne datoteke je "izlazna.isl". Ulazna datoteka je već prije opisana i lako se može ispuniti u Excelu. Klikom na "Izvrši Proracun" program vrši proračun.

U potprozoru "Osnovni Podaci" nalazi se leteći izbornik "Oznaka Vucnog Vozila" gdje biramo vrstu vučnog vozila i samim tim činom pridjeljujemo programu globalnu varijablu pomoću koje program određuje vučnu silu, faktor snage, kočnu silu u svakom koraku proračuna te masu vučnog vozila. U polje "Broj Vucnih Vozila" upisujemo broj vozila koje vrše vuču (npr. 1 ili 2), dok u polje "Masa Vlaka" upisujemo masu cjelokupnog vlaka u tonama. Pod pojmom mase cjelokupnog vlaka smatramo masu vučnog vozila i svih vučenih vozila.

Ako u letećem izborniku "Oznaka Vucnog Vozila" odaberemo oznaku 1142 (što predstavlja oznaku elektromotornog vlaka serije HŽ 1142) ujedno se unaprijed određuje i broj vučnih vozila te masa cijelog vlaka.

U potprozoru "Podaci o Proracunu" upisujemo početnu brzinu vlaka te krajnju. Ako ne upišemo ništa u polja za početnu i krajnju brzinu, program pretpostavlja početnu i krajnju brzinu od 0 km/h. U polje "Korak Proracuna" upisujemo vrijeme koraka proračuna programa u sekundama, a ako ne upišemo ništa program pretpostavlja vrijednost od 1 sekunde. Na dnu se nalazi upit "Povrat Snage u KM", a odabirom te opcije je predviđeno vraćanje snage u kontaktnu mrežu pri kočenju, što nije bio slučaj za algoritam u Fortranu. Ako ne odaberemo ništa program pretpostavlja da vraćanje snage u kontaktnu mrežu nije moguće. Za proračun snage koju je moguće vratiti u mrežu korišteni su izrazi (2.2.5) i (2.2.6).

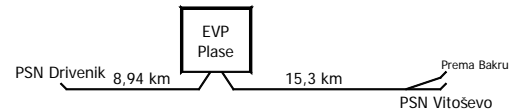
Osnovni kod programa se nalazi u lit.[5], ali su dodani određeni dijelovi i napravljene promjene koda da bi sadržaj izlazne datoteke bio adekvatan za ostale proračune (kao u prethodnom algoritmu).

Da bi se usporedili rezultati dobiveni s oba algoritma, nastojalo se usporediti rezultate dobivene za konkretne vlakove koji prometuju na dionici Rijeka – Moravice. U razmatranje su uzeta EVP Delnice i EVP Plase.

Najprije su uspoređeni rezultati za 4 vlaka koja su napajana EVP-om Delnice, a zatim slijede rezultati za 4 vlaka koja su napajana sa EVP-om Plase. Krakovi napajanja za pojedini EVP se nalaze na slici 2.23. i slici 2.24.

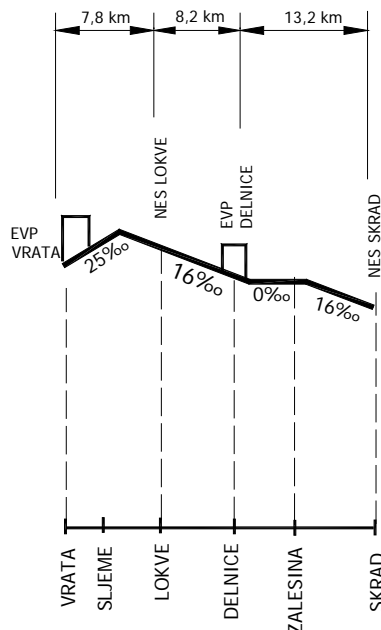


Slika 2.23. Krakovi napajanja EVP Delnice pri ispadu EVP Vrata

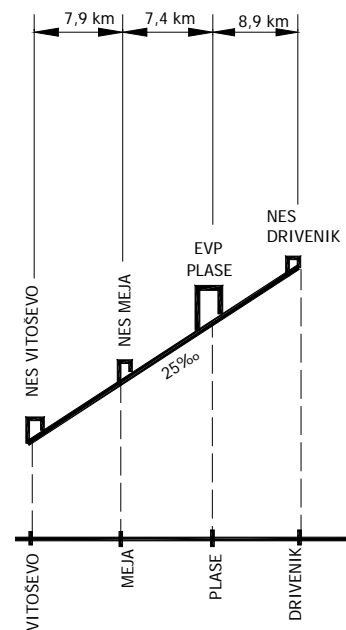


Slika 2.24. Krakovi napajanja EVP Plase

Pojednostavljeni profil pruge za dionice koje napaja pojedini EVP se nalazi na slici 2.25. i slici 2.26.



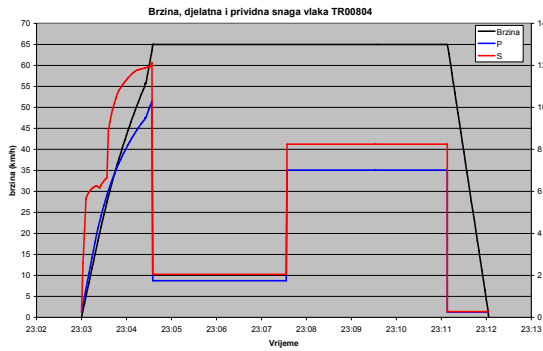
Slika 2.25. Pojednostavljeni profil pruge za dionicu koju napaja EVP Delnice



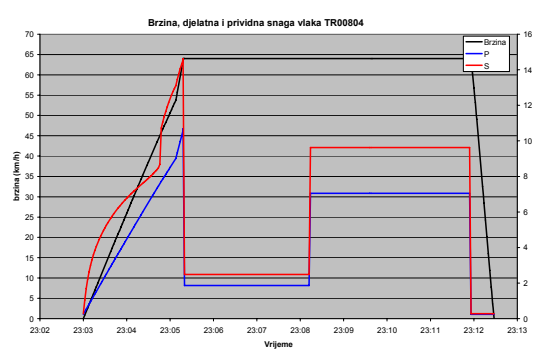
Slika 2.26. Pojednostavljeni profil pruge za dionicu koju napaja EVP Plase

U oba slučaja su u proračun uzete lokomotive tipa 1142, a u Borland C++ je pretpostavljeno da su to srednje natovareni teretni vlakovi. Vlakovi su težine 1100 t (3 lokomotive) i 750 t (2 lokomotive), što je naznačeno u ulaznoj datoteci.

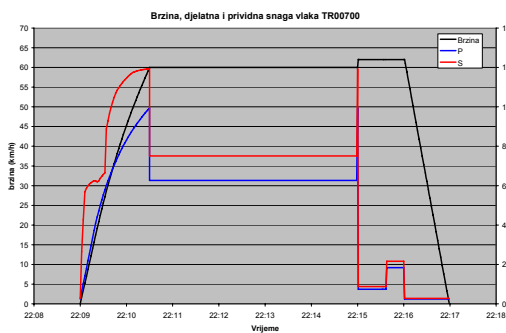
Na slijedećim slikama je grafički prikazan dio rezultata simulacije konkretnog vlaka iz voznog reda za dionicu koju napaja EVP Delnice, a to su: ovisnosti brzine, prividne i djelatne snage teretnog vlaka o vremenu. Također su uspoređeni rezultati dobiveni sa Fortranom (lijevo) i Borlandom C++ (desno).



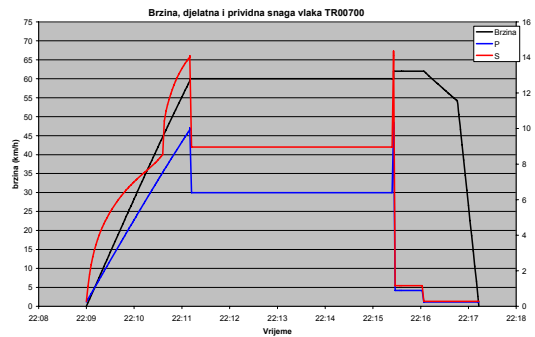
Slika 2.27. Grafički rezultati vlaka na dionici Delnice-Lokve pomoću Fortrana



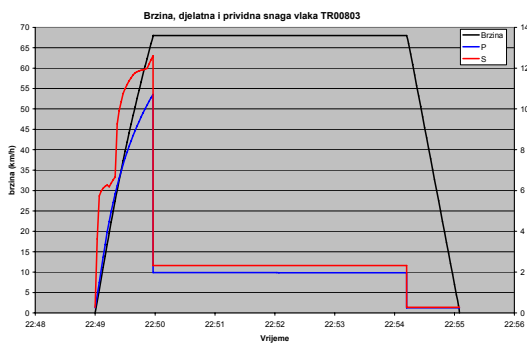
Slika 2.28. Grafički rezultati vlaka na dionici Delnice-Lokve pomoću Borland C++



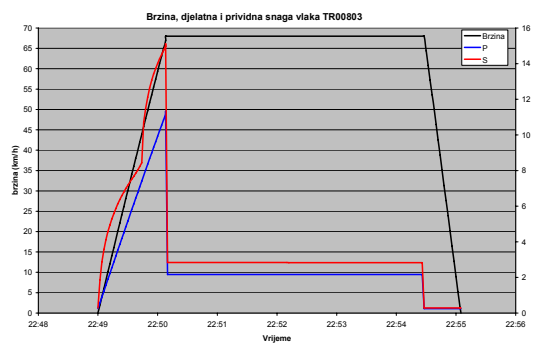
Slika 2.29. Grafički rezultati simulacije vlaka na dionici Skrad-Zalesina Fortranom



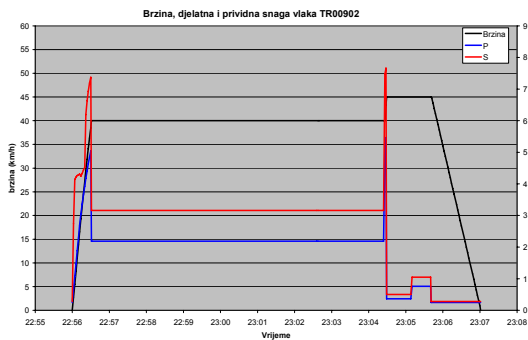
Slika 2.30. Grafički rezultati simulacije vlaka na dionici Skrad-Zalesina Borland C++-om



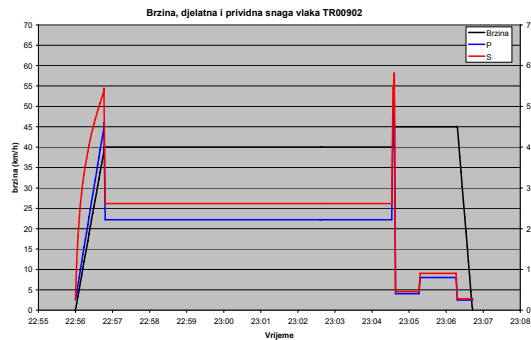
Slika 2.31. Grafički rezultati simulacije vlaka na dionici Zalesina-Delnice Fortranom



Slika 2.32. Grafički rezultati simulacije vlaka na dionici Zalesina-Delnice Borland C++-om

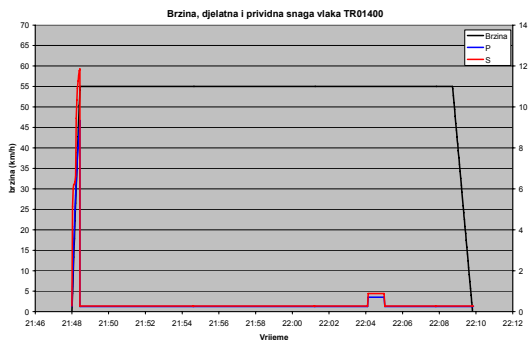


Slika 2.33. Grafički rezultati simulacije vlaka na dionici Skrad-Zalesina Fortranom

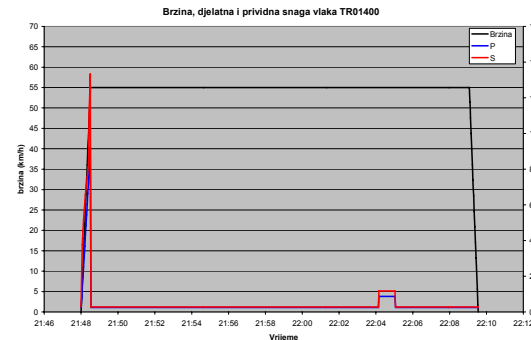


Slika 2.34. Grafički rezultati simulacije vlaka na dionici Skrad-Zalesina Borland C++-om

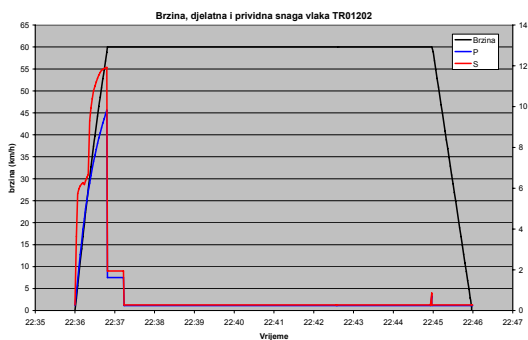
Sada slijede grafički prikazi rezultata dobivenih za pojedine vlakove koje napaja EVP Plase i usporedba rezultata dobivenih pomoću oba programska jezika.



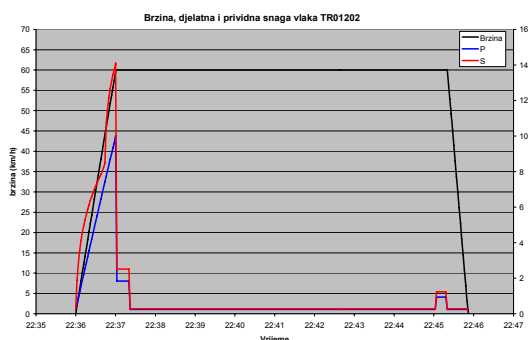
Slika 2.35. Grafički rezultati simulacije vlaka na dionici Plase-D.Stijena Fortranom



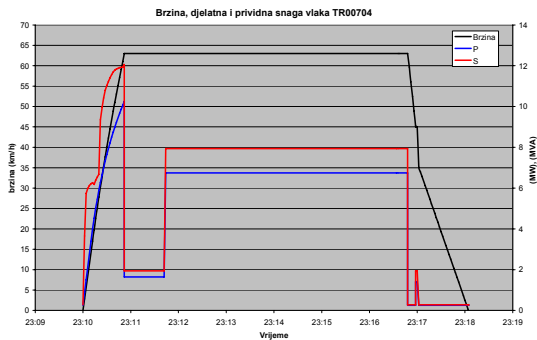
Slika 2.36. Grafički rezultati simulacije vlaka na dionici Plase-D.Stijena Borland C++-om



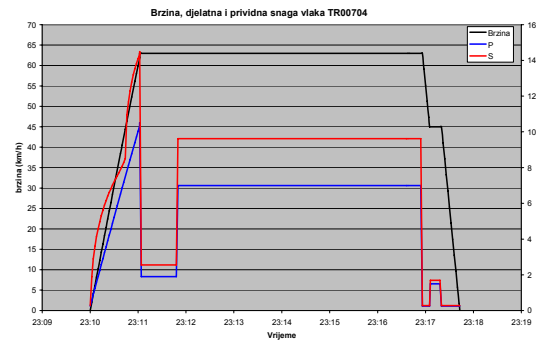
Slika 2.37. Grafički rezultati simulacije vlaka na dionici Drivenik-Plase Fortranom



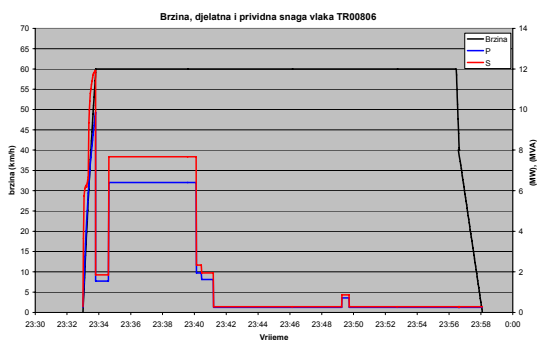
Slika 2.38. Grafički rezultati simulacije vlaka na dionici Drivenik-Plase Borland C++-om



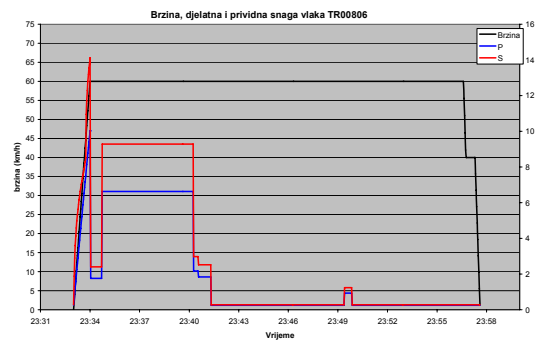
Slika 2.39. Grafički rezultati simulacije vlaka na dionici Fužine-Drivenik Fortranom



Slika 2.40. Grafički rezultati simulacije vlaka na dionici Fužine-Drivenik Borland C++-om



Slika 2.41. Grafički rezultati simulacije vlaka na dionici Fužine-Meja Fortranom



Slika 2.42. Grafički rezultati simulacije vlaka na dionici Fužine-Meja Borland C++-om

Iz prethodnih slika se vidi da se rezultati dobiveni pomoću ova dva algoritma dosta podudaraju. Može se uočiti da iznos snage koju pojedini vlak uzima iz mreže, u oba slučaja, poprima maksimalnu vrijednost prilikom ubrzavanja do određene brzine, koja je definirana za svaku dionicu. Nastojalo se odrediti brzine na pojedinoj dionici tako da snaga koju vuče pojedini vlak iz mreže bude što bolje raspoređena za vrijeme njegovog voznog reda.

Pojavljuje se malo odstupanje prvenstveno u iznosu prividne snage, a to je prije svega zbog boljih performansi algoritma u Borland C++, tj. opsežnijeg modeliranja krivulja vučnog pasoša i faktora snage.

Na kraju se da zaključiti da je algoritam u Borlandu C++-u omogućuje sofisticiraniji proračun električnih prilika u kontaktnoj mreži jer omogućuje proračune za sve tipove lokomotiva trenutno prisutnih u HŽ-u.

Budući da svaki tip lokomotive, kao što je već navedeno, ima jedinstven vučni pasoš, karakteristiku faktora snage i električne kočnice, koji bitno utječu na ukupni proračun električnih prilika u kontaktnoj mreži, algoritam u Borland C++ je praktičniji.

3. Određivanje električnih parametara kontaktne mreže

Projektiranje potpuno novog elektrovučnog sustava ili stavljanje u pogon novih vozila, uređaja za signalizaciju zahtjeva procjenu elektromagnetske kompatibilnosti (EMC) cijelog sustava i proračune međudjelovanja između komponenata i pojedinih dijelova sustava.

Model prijenosnih vodova sa više vodiča (MTL) je primijenjen za analiziranje električnih prilika u kontaktnoj mreži za obje domene (frekvencijska i vremenska). Ovaj model je uspješno primijenjen na željeznički prijenosni sustav s aspekta elektromagnetske kompatibilnosti i za međusobni utjecaj pojedinih komponenata kontaktne mreže. Bilo koja metoda, za određivanje električnih prilika u kontaktnoj mreži, ne može biti provedena bez poznavanja električnih parametara kontaktne mreže. Pojedini parametri mogu biti određeni s velikom točnošću, koristeći poznate formule. Ostali parametri (npr. električna karakteristika zemlje, vodljivost između tračnica i zemlje ili kapacitet među tračnicama) mogu biti određeni samo eksperimentalno, štoviše mogu mijenjati vrijednost s promjenom temperature i ostalih vremenskih uvjeta. Bilo bi logično te parametre odrediti na indirektan način mjereći dostupne podatke.

3.1. Proračun impedancija kontaktne mreže 25 kV, 50 Hz

3.1.1. Općenito o impedanciji

Impedancija elemenata protjecanih izmjeničnom strujom računa se kao kompleksni zbroj djelatne (R) i induktivne komponente (X) otpora pa za jednu petlju sa zemljom kao povratnim vodičem vrijedi sljedeći izraz [10]:

$$Z_i = R + R_z + jX, \quad \Omega/\text{km} \quad (3.1.1)$$

gdje je: R – vlastiti radni otpor vodiča posmatrane petlje,
 R_z – radni otpor zemlje kao povratnog voda,
 X – vlastiti induktivni otpor promatrane petlje.

Za određivanje direktnih i nultih komponenti uzdužnih impedancija nadzemnih vodova polazi se od Carsonovih formula za vlastite i međusobne impedancije faznih vodiča i zaštitnih užeta. Općeniti Carsonovi izrazi glase [11]:

$$\begin{aligned} Z_{ii-z} &= (R_i + \Delta R_{ii}) + j(2 \cdot \omega \cdot 10^{-4} \cdot \ln \frac{2h_i}{GMR_i} + \Delta X_{ii}) \\ & \quad , \quad \Omega/\text{km} \quad (3.1.2) \\ Z_{ik-z} &= \Delta R_{ik} + j(2 \cdot \omega \cdot 10^{-4} \cdot \ln \frac{D_{ik}}{d_{ik}} + \Delta X_{ik}) \end{aligned}$$

gdje je: R_i – jedinični otpor vodiča, Ω/km ,
 h_i – prosječna visina vodiča i iznad zemlje,
 GMR_i – reducirani polumjer vodiča,
 D_{ik} – razmak između vodiča i i zrcalne slike vodiča k ,

d_{ik} – razmak između vodiča i i vodiča k ,
 ω - kružna frekvencija,
 $\Delta R, \Delta X$ – Carsonovi korekcijski faktori.

Carsonovi korekcijski faktori ΔR i ΔX prikazuju utjecaj povratne petlje kroz zemlju i funkcija su kuta φ ($\varphi = 0$ za vlastitu impedanciju i $\varphi = \varphi_{ik}$ za međusobnu impedanciju) te parametra a :

$$a = 4\pi\sqrt{5} \cdot 10^{-4} \cdot D \cdot \sqrt{\frac{f}{\rho}} \quad (3.1.3)$$

gdje je: $D = 2h_i$ - u metrima za vlastitu impedanciju,
 ρ - specifični otpor zemlje u Ωm .

Ti korekcijski faktori su prikazani kao beskonačni integrali i kao redovi. Kod pogonske frekvencije potrebno je uzeti u obzir samo nekoliko članova reda. Tako da izraz za vlastitu impedanciju petlje vodič – zemlja za pogonsku frekvenciju ima sljedeći oblik:

$$Z_i = R_1 + 0.0493 + j \left[0,0445 \times \log \left(\frac{658}{r} \times \sqrt{\frac{\rho}{f}} \right) + 0,0157 \right], \quad \Omega/\text{km} \quad (3.1.4)$$

Izraz za međusobnu impedanciju dvaju vodiča pri frekvenciji od 50 Hz poprima oblik:

$$Z_m = 0.0493 + j \left[0,0445 \times \log \left(\frac{658}{D_{ik}} \times \sqrt{\frac{\rho}{f}} \right) \right], \quad \Omega/\text{km} \quad (3.1.5)$$

gdje je: R_1 – djelatni otpor vodiča uz utjecaj skin efekta,
 D_{ik} – međusobna udaljenost među vodičima,
 ρ – specifični otpor tla, 300 Ωm ,
 f – frekvencija 50 Hz,
 r - polumjer vodiča ako uzmemo u obzir skin efekt.

Izraz za radni otpor zemlje za pogonsku frekvenciju od 50 Hz glasi [10]:

$$R_z = \frac{\pi \cdot \omega}{2} \cdot 10^{-4} = 0.049348, \quad \Omega/\text{km} \quad (3.1.6)$$

Induktivna komponenta ili reaktancija sastoji se od unutrašnje i vanjske komponente [11]:

$$X = X_u + X_v = \omega \cdot (L_v + L_u), \quad \Omega/\text{km} \quad (3.1.7)$$

gdje je: L_u - unutarjni induktivitet vodiča ($0.5 \cdot 10^{-4}$), H/m
 L_v - vanjski induktivitet vodiča ($2 \cdot 10^{-4} \cdot \ln(658/r) \cdot (\rho/f)^{1/2}$), H/m

Unutrašnja komponenta reaktancije vezana je za raspodjelu magnetskog polja u samom vodiču i računa se prema izrazu [10]:

$$X_u = \frac{1}{2} \cdot \omega \cdot \mu \cdot 10^{-4}, \quad \Omega/\text{km} \quad (3.1.8)$$

gdje je: ω - kružna frekvencija izmjenične struje $\omega = 2 \cdot \pi \cdot f = 314$, rad/s^{-1} ,
 μ - magnetska permeabilnost.

Kod neferomagnetskih materijala, u koje spadaju bakar i aluminij od kojih se izrađuju vodiči, $\mu \approx 1$ pa je X_u konstantno za frekvenciju od 50 Hz i iznosi [10]:

$$X_u = \frac{1}{2} \cdot \omega \cdot 10^{-4} = 0.5 \cdot 314 \cdot 10^{-4} = 0.01571, \quad \Omega/\text{km} \quad (3.1.9)$$

vanjska reaktancija računa se prema izrazu [10]:

$$X_v = \omega \cdot L_v, \quad \Omega/\text{km} \quad (3.1.10)$$

gdje je: L – induktivitet vodiča, H/km.

Pollaczekov obrazac, koji se nalazi u « Propisima » Međunarodnog savjetodavnog komiteta za telefoniju (CCIF) glasi [10]:

$$X_v = \left[\frac{1}{2} + 2 \ln \frac{2}{\alpha \cdot \gamma \cdot r} \right] \cdot 10^{-4}, \quad \Omega/\text{km} \quad (3.1.11)$$

gdje je: $\alpha = \sqrt{4 \cdot \pi \cdot \kappa \cdot \omega}$,
 $\gamma = 1.7811$ – Besselova konstanta,
 r – polumjer vodiča, cm,
 κ - specifična vodljivost zemlje, S/cm.

To znači da vanjski induktivitet za poznatu frekvenciju i dimenzije vodiča ovisi samo o svojstvima zemljišta. Međutim ako u izraz za α uvrstimo nekoliko najčešćih vrijednosti κ , ustanovit ćemo ta κ nema praktičnog utjecaja na veličinu impedancije kontaktne mreže (< 5 %) i tako male promjene se mogu zanemariti.

Ako u izraz (3.1.11) uvrstimo poznate i pretpostavljene vrijednosti ($\omega = 314 \text{ rad/s}$, $\kappa = 10^{-4} \text{ S/cm}$) možemo ga transformirati u oblik pogodniji za upotrebu [10]:

$$X_v = 314 \cdot (21.4 - 2 \ln r) \cdot 10^{-4}, \quad \Omega/\text{km} \quad (3.1.12)$$

Slično se računa i međuimpedancija koja se javlja između 2 vodiča protjecana izmjeničnom strujom zbog pojave međuindukcije, a računa se prema izrazu [12]:

$$Z_m = jX_m = j\omega \cdot M = R_z + j\omega L_v, \quad \Omega/\text{km} \quad (3.1.13)$$

gdje je: M – međuinduktivitet između 2 vodiča, H/km

Induktivitet (L) i međuinduktivitet (M) računaju se prema Pollaczeku, [10] pa slijedi:

$$L = \left(1 + 2 \cdot \ln \frac{10^4}{\gamma \cdot r \cdot \sqrt{10 \cdot \pi \cdot \omega \cdot \kappa}} - j \frac{\pi}{2} \right) \cdot 10^{-4} \text{ , H/km} \quad (3.1.14)$$

$$M = \left(1 + 2 \cdot \ln \frac{10^4}{\gamma \cdot D \cdot \sqrt{10 \cdot \pi \cdot \omega \cdot \kappa}} - j \frac{\pi}{2} \right) \cdot 10^{-4} \text{ , H/km} \quad (3.1.15)$$

gdje je: γ - Besselova konstanta ($\gamma = 1.7811$),
 r - polumjer vodiča, m
 κ - specifična vodljivost zemlje, S/m
 D - međusobna udaljenost dva vodiča, m.

Kod neferomagnetskih materijala poznata je vrijednost unutarnje impedancije pa se za vodiče od takvih materijala odmah može odrediti i ukupna reaktancija pojednostavljenjem izraza (3.1.4) [12] pri čemu član 0.0157 dolazi samo za nemagnetske materijale.

:

$$Z_i = R + 0.0493 + j0.1446 \cdot (2.08 - \log r + 0.5 \cdot \log \rho) \text{ , } \Omega/\text{km} \quad (3.1.16)$$

Pojednostavljenjem izraza (3.1.5) izraz za međusobnu impedanciju dva voda [12]:

$$Z_m = jX_m = 0.0493 + j0.1446 \cdot (1.97 - \log D + 0.5 \cdot \log \rho) \text{ , } \Omega/\text{km} \quad (3.1.17)$$

gdje je: r - polumjer vodiča, m
 D - međusobna udaljenost dva vodiča, m
 ρ - specifični otpor tla ($\rho = 1/\kappa$), Ω
 f - pogonska frekvencija (50 Hz).

U literaturi [12] za izraze vlastite i međusobne impedancije je korištena specifična vodljivost tla (σ) umjesto specifičnog otpora tla i imamo sljedeće izraze :

Izraz za vlastitu impedanciju Z_i prema lit [12] glasi:

$$Z_i = R + 0.0493 - j0.1446 \cdot (1.42 + \log r + 0.5 \cdot \log \sigma) \text{ , } \Omega/\text{km} \quad (3.1.18)$$

dok izraz za međusobnu impedanciju Z_m ima sljedeći oblik:

$$Z_m = jX_m = 0.0493 - j0.1446 \cdot (1.53 + \log D + 0.5 \cdot \log \sigma) \text{ , } \Omega/\text{km} \quad (3.1.19)$$

gdje je: r - polumjer vodiča, cm
 D - međusobna udaljenost dva vodiča, cm
 σ - specifična vodljivost tla, S/cm
 f - pogonska frekvencija (50 Hz).

Na osnovi prethodno navedenog može se doći do zaključka da impedancija kontaktne mreže se sastoji od:

- impedancije petlje kontakti vodič – zemlja Z_k
- impedancije petlje nosivo uže – zemlja Z_u

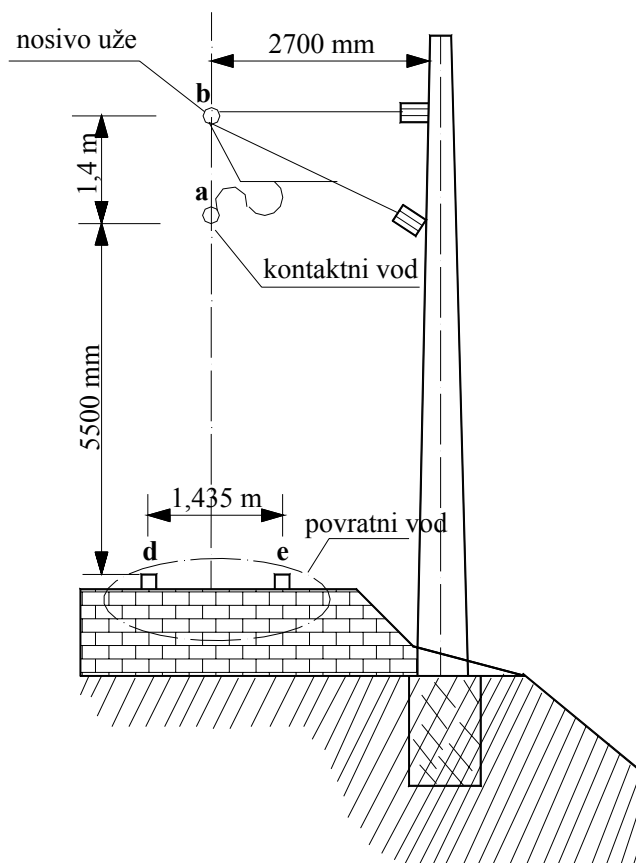
- međuimpedancije kontaktni vodič – nosivo uže Z_{ku}
- impedancije voznog voda Z_v
- impedancije povratnog voda Z_p
- međuimpedancije vozni vod – povratni vod Z_{vp} .

Utjecaj odvoda zemlje (vodljivosti) na vrijednosti impedancije (R i L) za visoke vrijednosti frekvencija je zanemariv i on je zamjetan samo za niže frekvencije [9]. Odvod zemlje je važan za određivanje matrice vodljivosti Y , a iznos mu ovisi o vrsti šljunka među tračnicama, temperaturi, vlažnosti, prljavštini i vegetaciji. Vrijednost odvoda zemlje varira u velikom rasponu (0,01 – 100 S/km) [9].

Sada će se odrediti izrazi za impedancije pojedinih petlji koji će se koristiti u proračunu impedancije kontaktne mreže (prilog 3.).

3.1.2. Impedancija voznog voda

Kao što je već navedeno vozni vod čine nosivo uže i kontaktni vodič koji su međusobno paralelno povezani ekvipotencijalnim vezama (slika 3.5.). Kada kroz njega protječe struja možemo ga promatrati kao paralelu dvije petlje koje čine dva navedena dijela sa zemljom.



Slika 3.5. Položaj kontaktnog vodiča i nosivog užeta na stupu

Impedancija petlje kontakti vodič – zemlja

Impedancija petlje kontakti vodič – zemlja računa se prema (3.1.16) [13]:

$$Z_k = R_k + 0.0493 + j0.1446 \cdot (2.08 - \log r_k + 0.5 \log \rho), \quad \Omega/\text{km} \quad (3.1.20)$$

gdje je: R_k – djelatni otpor kontaktnog vodiča, Ω/km
 r_k – polumjer kontaktnog vodiča, m
 ρ – specifični otpor tla, Ωm .

Ako djelatni otpor nije zadan može se izračunati prema izrazu:

$$R_k = \frac{\rho_k}{S_k} \cdot 10^3, \quad \Omega/\text{km} \quad (3.1.21)$$

gdje je: ρ_k – specifični djelatni otpor kontaktnog vodiča, $\Omega\text{mm}^2/\text{m}$
 S_k – presjek kontaktnog vodiča, mm^2 .

Na prugama Hrvatskih željeznica (25 kV, 50 Hz) uglavnom se upotrebljavaju kontakti vodiči Ri100 od tvrdo vučenog bakra nazivnog presjeka 100 mm^2 , polumjera 6 mm i djelatnog otpora $0.1759 \Omega/\text{km}$.

Impedancija petlje nosivo uže – zemlja se računa na isti način kao za kontakti vodič [13]:

$$Z_u = R_u + 0.0493 + j0.1446 \cdot (2.08 - \log r_u + 0.5 \log \rho) \quad (3.1.22)$$

Međusobna impedancija kontakti vodič – nosivo uže računa se po izrazu (3.1.17) :

$$Z_{KU} = 0.0493 + j0.1446 \cdot (1.97 - \log D_{KU} + 0.5 \log \rho) \quad (3.1.23)$$

gdje je D_{KU} – udaljenost između ekvivalentnog nosivog užeta i kontaktnog vodiča, m
 ρ – specifični otpor tla, Ωm .

Za proračun se koristi ekvivalentno nosivo uže jer razmak između kontaktnog vodiča i nosivog užeta nije jednak cijelom dužinom zbog oblika nosivog užeta. Na ovaj način se lančanica nosivog užeta aproksimira pravcem koji je cijelom svojom dužinom jednako udaljen od kontaktnog vodiča pa iz slike 3.6. slijedi [13]:

$$D_{KU} = S - \frac{2}{3} \cdot F, \quad \text{m} \quad (3.1.24)$$

gdje je: S – sistemska visina, m

F – najveći provjes (na sredini između 2 stupa), m.

Visina provjesa određuje se prema rasponu između stupova prema izrazu:

$$F = 1.95 \cdot 10^{-4} \cdot A^2, \text{ m} \quad (3.1.25)$$

gdje je: A – raspon između 2 susjedna stupa, m.

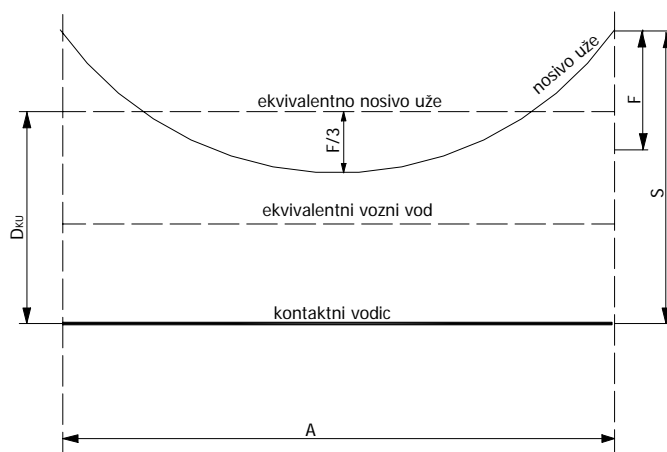
Sada možemo odrediti impedanciju voznog voda prema izrazu [13]:

$$Z_v = \frac{Z_k \cdot Z_u - Z_{ku}^2}{Z_k + Z_u - 2 \cdot Z_{ku}}, \text{ } \Omega/\text{km} \quad (3.1.26)$$

gdje je: Z_k – impedancija kontaktnog vodiča, Ω/km

Z_u – impedancija nosivog užeta, Ω/km

Z_{ku} – impedancija kontaktni vodič – nosivo uže, Ω/km .



Slika 3.6. Položaj ekvivalentnih vodiča u kontaktnoj mreži

3.1.3. Impedancija i napon povratnog voda

Koristeći Carsonove formule za tračnice treba naglasiti da su one predstavljene kao cilindrični vodiči. Te pretpostavke su zanemarive za određivanje induktiviteta, ali računanje otpora pri niskim frekvencijama nije tako precizno zbog neujednačene raspodjele struje po površini tračnica.

Kao povratni vod koristi se pruga i to jedna ili obje tračnice. Tračnica prolasku struje kroz nju pruža ohmsko (djelatno) – induktivni otpor. Djelatni otpor tračnice ovisi o magnetskom permeabilitetu tračnice koji utječe na pojavu skin - efekta tj. pojave potiskivanja struje prema površini tračnice povećavajući tako prividni otpor i računa se prema sljedećem izrazu [13]:

$$R_t = \frac{0.28}{O} \cdot \sqrt{\rho_t \cdot f \cdot \mu_t} \quad , \quad \Omega/\text{km} \quad (3.1.27)$$

gdje je: O – opseg poprečnog presjeka tračnice, cm
 ρ_t – specifični otpor tračničkog čelika pri 20 °C, $\Omega\text{mm}^2/\text{m}$
 f – frekvencija izmjenične struje kroz tračnicu, Hz
 μ_t – magnetska permeabilnost tračničkog čelika.

Raspodjelu polja u tračnicama i određivanje njihovih električnih parametara izračunavali su mnogi autori (Benda, Brileev), zato i postoji velik broj proračunskih metoda koje često daju rezultate koji se uvelike razilaze [10].

U proračunu se tračnica aproksimira krugom odgovarajućeg presjeka radi pojednostavljenja [13]. Za određivanje ekvivalentnog polumjera tračnica moglo bi se usvojiti sljedeće pravilo [6]:

- za visoke frekvencije, pri kojima se javlja «skin efekt», ekvivalentni polumjer tračnice se računa iz opsega tračnice (tj. iz izraza $2\pi r_{ekv} = O_t$).

- za niske frekvencije ekvivalentni polumjer se računa iz poprečnog presjeka tračnice (tj. iz izraza ($\pi r_{ekv}^2 = A_t$)).

Ta dva izraza daju različite vrijednosti za ekvivalentni polumjer, čak i za red veličine, ovisno o obliku vodiča.

Unutarnja reaktancija tračnice zbog promjenjivog permeabiliteta teško se izračunava pa se uzima da iznosi 75% djelatnog otpora tračnice ($X_u \approx 0.75 R_t$), a do tog odnosa je došao Zickler [12].

Vrijednost impedancije povratnog voda ovisi o broju kolosijeka i broju povratnih vodova. Ako se uvažavaju prethodne pretpostavke dobiju se sljedeći izrazi za ovu impedanciju:

- impedancija povratnog voda s jednom povratnom tračnicom (jednokolosiječna pruga) [12]:

$$Z_{p(1,1)} = R_t + 0.0493 - j0.1446 \cdot (1.97 + 5.2 \cdot R_t - \log r_t + 0.5 \cdot \log \rho) \quad (3.1.28)$$

- impedancija povratnog voda s dvije povratne tračnice (jednokolosiječna pruga):

$$Z_{p(1,2)} = 0.5 \cdot R_t + 0.0493 + j0.1446 \cdot (1.97 + 2.6 \cdot R_t - 0.5 \cdot \log r_t - 0.5 \cdot \log D_t + 0.5 \cdot \log \rho) \quad (3.1.29)$$

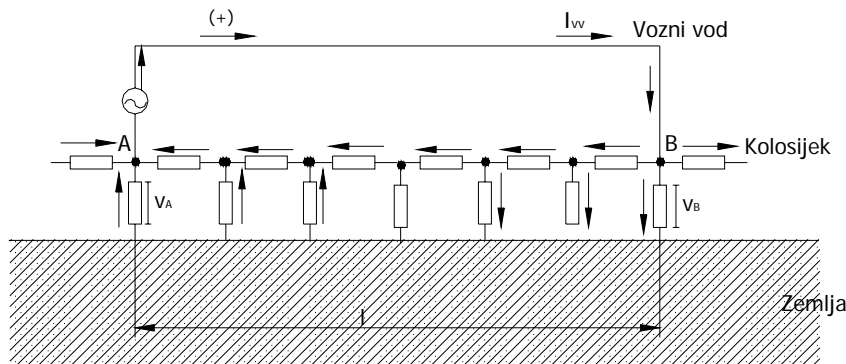
gdje je: R_t – djelatni otpor tračnice, Ω/km
 r_t – polumjer opsega tračnice, m
 D_t – razmak između težišta tračnica, m
 D_k – razmak između simetrala kolosijeka, m
 ρ – specifični otpor tla, Ωm .

Za ostale kombinacije vrijede analogne formule.

Klasične prijenosne diferencijalne jednačbe za kolosijek kao električni vod glase [10]:

$$\begin{aligned} -\frac{\delta U}{\delta x} &= Z_s \cdot I \\ -\frac{\delta I}{\delta x} &= Y_s \cdot U \end{aligned} \quad (3.1.30)$$

Ako na slici 3.7. sa A označimo točku priključka voznog voda na sabirnice elektrovučne podstanice, a sa B točku u kojoj je vozni vod vezan za kolosijek (odnosno točku u kojoj se u određenom trenutku nalazi vučno vozilo), onda je očigledno da je rješenje gornjeg sustava jednadžbi olakšano njegovom simetričnošću.



Slika 3.7. Pojednostavljena shema jedne dionice kontaktne mreže

U točki B struja ulazi u kolosijek, a u točki A ona izlazi iz njega. Potencijali tračnica u odnosu na zemlju su u tim točkama suprotnog predznaka. S navedenim oznakama sa slike 3.7. dolazimo do slijedećih izraza [10]:

$$\begin{aligned} U &= U_A \operatorname{ch}\gamma x - Z_{c\bar{s}} I_A \operatorname{sh}\gamma x \\ I &= I_A \operatorname{ch}\gamma x - \frac{U_A}{Z_{c\bar{s}}} \operatorname{sh}\gamma x \end{aligned} \quad (3.1.31)$$

gdje je: U_A – nazivni napon EVP-a (priključak voznog voda na sabirnice KM-e)

$Z_{c\bar{s}}$ – karakteristična (valna) impedancija tračnica ($Z_{c\bar{s}} = \sqrt{\frac{Z_s}{Y_s}}$)

γ – konstanta prostiranja ($\gamma = \sqrt{Z_s \cdot Y_s}$)

Z_s – impedancija tračnice

Y_s – odvod tračnica

x – udaljenost vučnog vozila od EVP-a .

Rješenje sustava jednadžbi (3.1.31) koji određuje raspodjelu struja i napona u tračnicama dobija se iz graničnih uvjeta koji rezultiraju iz njegove simetričnosti. Granični uvjeti su dakle:

$$\text{za } x = l/2, \quad U = 0 \quad \text{i} \quad |I_A| = |I_B|$$

a ulazne impedancije u točkama A i B su također jednake i određene izrazom [10]:

$$Z = Z_{c\bar{s}} \cdot \operatorname{th}\gamma l / 2 \quad (3.1.32)$$

S druge strane kako je kolosijek pretpostavljen kao beskonačno dugi vod s homogeno raspodijeljenim parametrima, možemo pisati da je na svojim krajevima zatvoren karakterističnom impedancijom $Z_{c\bar{s}}$. Dakle vrijedi da je:

$$|I_A| = |I_B| = \left| \frac{I_{vv} \cdot Z_{c\dot{s}}}{Z_{c\dot{s}} (1 + \text{th}\gamma l / 2)} \right| \quad (3.1.33)$$

$$I_A = -\frac{I_{vv}}{2} (1 + e^{-\gamma l})$$

gdje je: I_{vv} – struja voznog voda.

Sa ovim graničnim uvjetima jednadžba koja određuje «uzdužni profil» vučne struje u kolosijeku u intervalu AB dobiva konačni oblik:

$$I_{pv} = \frac{I_{vv}}{2} [e^{-\gamma x} + e^{-\gamma(1-x)}] \quad (3.1.34)$$

Slijedi da napon između tračnica i zemlje u bilo kojoj točki intervala AB iznosi:

$$U_{\dot{s}z} = Z_{c\dot{s}} \frac{I_{vv}}{2} [e^{-\gamma x} + e^{-\gamma(1-x)}] \quad (3.1.35)$$

3.1.4. Međusobna impedancija vozni vod – povratni vod

Ova impedancija se računa između ekvivalentnog voznog voda i odgovarajućih tračnica koje se koriste kao povratni vodovi. Položaj ekvivalentnog voznog voda je u sredini između kontaktnog vodiča i ekvivalentnog nosivog užeta (slika 3.6.) odnosno na visini:

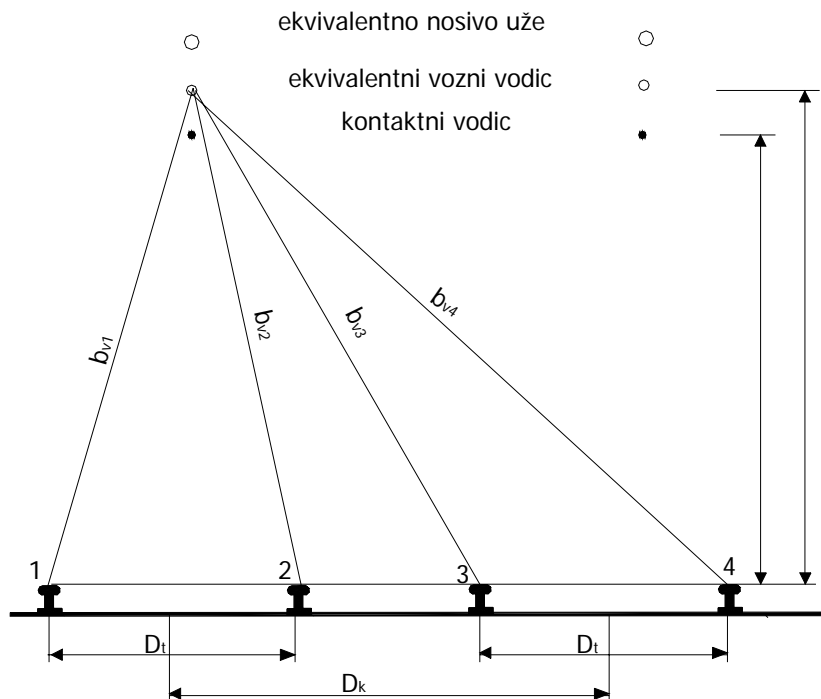
$$h_{ekv} = h + 0.5 \cdot S - \frac{1}{3} \cdot F, \quad \text{m} \quad (3.1.36)$$

gdje je: h – visina kontaktnog vodiča, m
 S – sistemska visina, m
 F – provjes nosivog užeta, m.

Kod izračuna ove impedancije potrebno je odrediti međusobnu udaljenost pojedinih tračnica od ekvivalentnog voznog voda (slika 4.5.) [12]:

$$b_1 = b_2 = \sqrt{\left(\frac{D_t}{2}\right)^2 + h_{ekv}^2}, \quad \text{m} \quad (3.1.37)$$

Ostale vrijednosti se iz geometrije prema slici 3.8.



Slika 3.8. Pojedine udaljenosti u kontaktnoj mreži

Kada smo odredili ove međusobne udaljenosti vodiča kontaktne mreže međusobna impedancija vodni vod - povratni vod se računa prema izrazu (3.1.11) što daje:

$$Z_{vp(m,n)} = 0.0493 + j0.1446 \cdot (1.97 - \log b + 0.5 \cdot \log \rho) \quad , \quad \Omega/\text{km} \quad (3.1.38)$$

gdje je: b – međusobna udaljenost osi voznog i povratnog voda, m
 m – broj kolosijeka (1 ili 2)
 n – broj povratnih tračnica (1,2 ili 4)
 ρ - specifični otpor tla, Ωm .

Kada bi tračnice bile potpuno izolirane od zastora i zemlje, struja opterećenja i struja kratkog spoja tekle bi samo kroz tračnice. Međutim tračnice su preko nasipa i pragova u mnogim točkama povezane sa zemljom pa struja teče dijelom kroz tračnice, a dijelom kroz zemlju koja djeluje kao vodič paralelno povezan s tračnicama.

Kad struja električne vuče teče kroz petlju «vozni vod – povratni vod» koja se sastoji od dvije paralelno povezane petlje – «vozni vod – zemlja», i «povratni vod – zemlja» vrijede sljedeće diferencijalne jednadžbe [13]:

$$\begin{aligned} \bar{Z}_{\bar{s}} \bar{I}_{\bar{s}} - \bar{Z}_{v\bar{s}} \bar{I} &= -\frac{d\bar{U}_{\bar{s}}}{dx} \\ g\bar{U}_{\bar{s}} &= -\frac{d\bar{I}_{\bar{s}}}{dx} \end{aligned} \quad (3.1.39)$$

gdje je: $I_{\bar{s}}$ – struja u povratnom vodu
 I – struja u voznom vodu

U_s – potencijal povratnog voda prema zemlji
 g - vodljivost «tračnica – zemlja»
 x - udaljenost od EVP-a
 Z_s – impedancija tračnica
 Z_{us} – međusobna impedancija dva kruga.

3.1.5. Impedancija kontaktne mreže

Impedancija kontaktne mreže dana je sljedećim izrazom [10]:

$$Z_{KM} = Z_v - p \cdot Z_{vp} + q \cdot (1 - p) \cdot (Z_p - Z_{vp}) \quad [\Omega/\text{km}] \quad (3.1.40)$$

pri čemu je:

$$p = \frac{Z_{vp}}{Z_p} \quad q = \frac{1 - e^{-k_p \cdot l}}{k_p \cdot l} \quad k_p = \sqrt{Z_s \cdot Y_s} \quad (3.1.41)$$

gdje je: Z_v – impedancija voznog voda, Ω/km
 Z_p – impedancija povratnog voda, Ω/km
 Z_{vp} – međusobna impedancija vozni vod – povratni vod, Ω/km
 k_p – konstanta rasprostiranja povratnog voda, km^{-1}
 l – duljina napojnog kraka od EVP-a, km
 Z_s – impedancija tračnice, Ω/km
 Y_s – vodljivost tračnice, S/km .

Umnožak $k_p \cdot l$ se često naziva « karakterističnom konstantom pruge » koja karakterizira jednu prugu prema njenoj duljini i lokalnim parametrima. Tom umnošku se ne može nikad odrediti točna vrijednost jer varira stalno pa se uzimaju dva granična slučaja izračunavanja impedancije kontaktne mreže s obzirom na udaljenost od elektrovučne podstanice i veličinu prijelaznog otpora tračnica – zemlja (g). [10]. Za slučaj opterećenja u blizini elektrovučne podstanice i dovoljno velikog prijelaznog otpora tračnica – zemlja tj. ako je ($k_p \cdot l \rightarrow 0$) slijedi:

$$Z_{KM} = Z_{KM0} = Z_v + Z_p - 2 \cdot Z_{vp}, \quad \Omega/\text{km} \quad (3.1.42)$$

U ovom slučaju cjelokupna struja I_s protječe kroz kolosijek i djeluje na elektromagnetsko polje voznog strujnog kruga pa je induktivni pad napona u petlji vozni vod – tračnice – zemlja veći, a u proračunu impedancije se mora uzeti u obzir i pad napona što ga u kolosijeku stvara promjenjivi dio struje u tračnicama – zemljovodna struja. U proračunu je uzeta srednja vrijednost ove impedancije [12].

Za suprotni slučaj, odnosno za slučaj značajno udaljenog opterećenja od EVP-a i zanemarivo malog prijelaznog otpora tračnica – zemlja može se reći da umnožak k_p i l teži ∞ i tada izraz (3.1.5.1) prelazi u:

$$Z_{KM} = Z_{KM\infty} = Z_v - p \cdot Z_{vp}, \quad \Omega/\text{km} \quad (3.1.43)$$

Stvarna i točna vrijednost impedancije se nalazi između ovih granica, ali se sa dovoljnom točnošću može uzeti da za pretežan broj elektrificiranih pruga odgovara minimalna vrijednost impedancije Z , dok ćemo mi u proračunu koristiti srednju vrijednost impedancije.

3.1.6. Primjer proračuna impedancija kontaktne mreže

Proračun je izvršen na dva načina, postepenim postupkom i matrično (blok transformacijom) i rezultati oba proračuna su dani u prilogu 3. Rezultati bi trebali biti približno isti jer su korištene identične formule i oba proračuna su rađena za isti primjer.

Primjer proračuna impedancija postepenim postupkom

Za ovaj proračun uzet je konkretan primjer za pruge HŽ-a: Podaci uzeti za proračun su prikazani u tablici 3.1. Izgled stupa KM-e za ovaj proračun je prikazan na slici 3.5.

Tablica 3.1. Podaci za proračun

kontakti vodič R_i 100	$R_k = 0.1759 \Omega/\text{km}$ $r_k = 6 \text{ mm}$ $S_k = 100 \text{ mm}^2$
nosivo uže B_z 65	$\rho_u = 0.02778 \Omega\text{mm}^2/\text{m}$ $r_u = 5.25 \text{ mm}; S_u = 65.8 \text{ mm}^2$
tračnica UIC 60	$O = 68.5 \text{ cm}$ $\rho_t = 0.22 \Omega\text{mm}^2/\text{m}$ $\mu_t = 250$
međusobne udaljenosti elemenata	$S = 1.4 \text{ m}; A = 60 \text{ m}$ $D_k = 4 \text{ m}; D_t = 1.435 \text{ m}$ $h = 5.5 \text{ m}$
okolno zemljište	$\rho = 100 \Omega\text{m}$ $R_z = 0.0493 \Omega/\text{km}$

Proračun je izveden u programu MATLAB 6.5, a ispis tog programa kao i rezultat su dani u prilogu 3. U programu je omogućen izbor povratnog voda s jednom ili dvije povratne tračnice, a u rezultatu proračuna je također dan ispis impedancija kontaktne mreže za oba slučaja (opterećenje blizu EVP-a i opterećenje značajno udaljeno od EVP-a) ovisno o udaljenosti opterećenja od EVP-a kao i srednja vrijednost impedancije KM koja se koristi u daljnjem proračunu. Korištene su formule iz prethodnog poglavlja. U programu je dovoljno promijeniti samo ulazne podatke (parametri kontaktne mreže) i dobije se odgovarajuća impedancija kontaktne mreže. Može se zaključiti da je impedancija KM ovisna o udaljenosti opterećenja od EVP-a i da je srednja vrijednost impedancije KM-e, za slučaj povratnog voda s jednom tračnicom, nešto veća nego u slučaju kada imamo povratni vod s dvije povratne tračnice.

Primjer proračuna impedancija matrično (blok - transformacijom)

Matrično to izgleda ovako:

$$[V] = [Z] * [I] \quad (3.1.44)$$

gdje je: $[V]$ - vektor stupac napona pojedinih vodiča kontaktne mreže,
 $[Z]$ - matrica impedancija vodiča kontaktne mreže,
 $[I]$ - vektor stupac struja pojedinih vodiča kontaktne mreže

Matrica impedancija $[Z]$ je dimenzija 4x4 i sastoji se od vlastitih i međusobnih impedancija pojedinih vodiča izračunatih u prethodnom poglavlju. Pojedini elementi matrice $[Z]$ su :

$$\begin{bmatrix} Z_{aa} & Z_{ab} & Z_{ad} & Z_{ae} \\ Z_{ba} & Z_{bb} & Z_{bd} & Z_{be} \\ Z_{da} & Z_{db} & Z_{dd} & Z_{de} \\ Z_{ea} & Z_{eb} & Z_{ed} & Z_{ee} \end{bmatrix}$$

Da bi mogli koristiti blok transformaciju krenut ćemo od sljedećih pretpostavki:

$$\begin{aligned} I_1 &= I_a + I_b & V_1 &= V_a = V_b \\ I_2 &= I_d + I_e & V_2 &= V_d = V_e \end{aligned} \quad (3.1.45)$$

Cilj nama je na lijevoj strani kod vektora napona $\begin{bmatrix} V_a \\ V_b \\ V_d \\ V_e \end{bmatrix}$ da pojedini članovi budu jednaki nuli

jer je tada omogućena blok transformacija. To ćemo postići računskim operacijama među recima i stupcima matrice $[Z]$.

Nakon obavljenih par takvih operacija dobijemo sljedeći vektor napona :

$$\begin{bmatrix} V_1 \\ 0 \\ V_2 \\ 0 \end{bmatrix}$$

Dok matrica impedancija nakon ovih transformacija ima sljedeći oblik:

$$\begin{bmatrix} Z_{aa} & Z_{ab} - Z_{aa} & Z_{ad} & Z_{ae} - Z_{ad} \\ Z_{ba} - Z_{aa} & Z_{bb} - Z_{ba} - Z_{ab} + Z_{aa} & Z_{bd} - Z_{ad} & Z_{be} - Z_{bd} - Z_{ae} + Z_{ad} \\ Z_{da} & Z_{db} - Z_{da} & Z_{dd} & Z_{de} - Z_{dd} \\ Z_{ea} - Z_{da} & Z_{eb} - Z_{ea} - Z_{db} + Z_{da} & Z_{ed} - Z_{dd} & Z_{ee} - Z_{ed} - Z_{de} + Z_{dd} \end{bmatrix}$$

To su elementi matrice $[A]$ kao što se vidi u prilogima, i tek sada se može raditi blok-transformacija dok se kod izvorne matrice $[Z]$ to nije moglo raditi. Najprije se ova matrica $[A]$ dobivena transformacijom matrice $[Z]$ oblika 4x4 svodi na matricu dimenzija 3x3 blok-transformacijom i dobije se matrica $[X]$, a zatim promjenom najprije drugog i trećeg stupca pa zatim retka se može izvršiti blok transformacija matrice dimenzija 3x3 i svesti na matricu $[N]$ dimenzija 2x2 što se vidi u prilogima.

To je matrica impedancija $[Z^{12}]$ svedena na impedanciju voznog voda i impedanciju povratnog voda. Ukupnu impedanciju kontaktne mreže odredit će se iz elemenata matrice $[N]$ ovisno o udaljenosti od EVP-a i koliko ima povratnih tračnica.

Izrazi za slučaj 1 ili 2 povratne tračnice su isti, samo se razlikuju vrijednosti za impedanciju povratnog voda Z_{pv} i međusobnu impedanciju povratnog i voznog voda Z_m . Tako ćemo navesti samo izraze za impedanciju KM u ovisnosti o udaljenosti opterećenja od EVP-a i njihovu srednju vrijednost impedancije KM.

Za slučaj ako je opterećenje blizu EVP-a imamo sljedeći izraz za impedanciju KM:

$$Z_{KM0} = Z_{vv} + Z_{pv} - 2 \cdot Z_m, \quad \Omega/\text{km} \quad (3.1.46)$$

pri čemu su: Z_{vv} – ekvivalentna impedancija voznog voda,
 Z_{pv} – ekvivalentna impedancija povratnog voda,
 Z_m – međusobna impedancija ova 2 voda.

i to su elementi matrice $[N]$ koja je dimenzija 2x2.

Za slučaj ako je opterećenje jako udaljeno od EVP-a imamo sljedeći izraz za impedanciju KM:

$$Z_{KM\infty} = Z_{vv} + Z_{pv} \cdot Z_{pv} / Z_m, \quad \Omega/\text{km} \quad (3.1.47)$$

Srednja vrijednost impedancije koja se obično uzima za daljnje proračune je jednaka:

$$Z_{KMS} = (Z_{KM0} + Z_{KM\infty}) / 2, \quad \Omega/\text{km} \quad (3.1.48)$$

Ispis programa za ovaj proračun dan je u prilogu 3. kao i rezultati tog programa. U ovom slučaju pojedine impedancije se mogu pročitati iz matrice impedancija.

U prilogu 3. je također ispisan proračun dobiven blok transformacijom i postepenim postupkom za slučaj a) i slučaj b) lit [13]. Za slučaj a) imamo jedan kontaktni vodič nominalnog presjeka 100 mm² i noseće uže nominalnog presjeka 120 mm² dok za slučaj b) imamo 2 kontaktna vodiča od po 100 mm² i noseće uže od 120 mm². Za slučaj a) uz specifični otpor tla od 3000 Ωm i prosječne udaljenosti opterećenja 10 km od EVP-a srednja vrijednost impedancije KM iznosila je: $Z_{KM} = 0,156 + j 0,475, \quad \Omega/\text{km}$, dok je za slučaj b) impedancija iznosila: $Z_{KM} = 0,145 + j 0,472 [\Omega/\text{km}]$. Rezultati proračuna za srednju vrijednost impedancije KM, zatim za impedanciju za slučaj opterećenja blizu EVP-a kao i kada i jako udaljenog opterećenja za slučaj a) dani su u tablici 3.3, a za slučaj b) su dani u tablici 3.4.

3.1.7. Prikaz rezultata dobivenih proračunima

Rezultati dobiveni ovim proračunima za slučaj iz tablice 3.1. su dani u tablici 3.2., a za slučaj iz lit. [13] u tablici 3.3. i tablici 3.4.

Tablica 3.2. Prikaz rezultata dobivenih proračunima za slučaj u tablici 3.1.

Vrsta pruge i proračuna		$Z_{KM} = R + j X, \quad \Omega/\text{km}$		
		kada $k_p \cdot l \rightarrow 0$	kada $k_p \cdot l \rightarrow \infty$	srednja vrijednost
Jednokolosiječna	s jednom povratnom tračnicom postepenim postupkom	0,4654 + j0,7876	0,2015 + j0,4882	0,3335 + j0,6379
	s dvije povratne tračnice postepenim postupkom	0,3019 + j0,5801	0,19180 + j0,4426	0,2469 + j0,5113
	Proračun blok-transformacijom	0,2994 + j 0,5824	0,1904 + j0,4435	0,2449 + j0,5130

Iz tablice 3.2. se može zaključiti da su rezultati približno isti kod proračuna blok-transformacijom i postepenim postupkom za dvije povratne tračnice što je bilo i za očekivati jer u proračunu blok-transformacijom smo zanemarili vodljivost tračnica. Potencijal tračnica se može zanemariti kod proračuna električnih prilika u normalnim pogonskim uvjetima jer je on neznatan u odnosu na pogonski napon (ekoliko volti). Također se može zaključiti da je induktivna komponenta u slučaju 2 povratne tračnice manja kao i djelatna što je bilo za očekivati. Prema vrijednostima impedancija podešava se i zaštita kontaktne mreže.

Tablica 3.3. Prikaz rezultata dobivenih proračunima za slučaj a) lit [13]

Vrsta pruge i proračuna		$Z_{KM} = R + j X, \Omega/\text{km}$		
		kada $k_p * l \rightarrow 0$	kada $k_p * l \rightarrow \infty$	srednja vrijednost
Jednokolosiječna	s jednom povratnom tračnicom postepenim postupkom	0,4090 + j0,7715	0,1592 + j0,5123	0,2841 + j0,6419
	s dvije povratne tračnice postepenim postupkom	0,2655 + j0,5625	0,1450 + j0,4491	0,1953 + j0,5058
	Proračun blok-transformacijom	0,2459 + j 0,5645	0,1450 + j0,4504	0,1955 + j0,5075

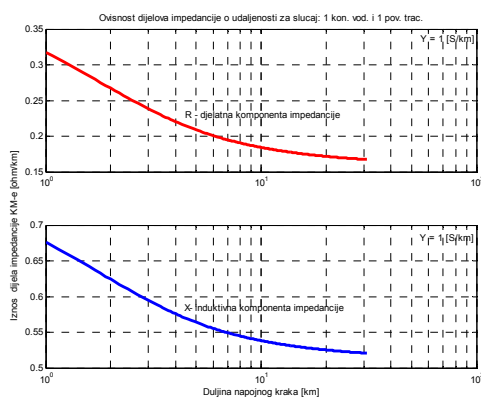
Tablica 3.4. Prikaz rezultata dobivenih proračunima za slučaj b) lit [13]

Vrsta pruge i proračuna		$Z_{KM} = R + j X, \Omega/\text{km}$		
		kada $k_p * l \rightarrow 0$	kada $k_p * l \rightarrow \infty$	srednja vrijednost
Jednokolosiječna	s jednom povratnom tračnicom postepenim postupkom	0,3872 + j0,7728	0,1374 + j0,5136	0,2623 + j0,6432
	s dvije povratne tračnice postepenim postupkom	0,2237 + j0,5639	0,1232 + j0,4505	0,1734 + j0,5072
	Proračun blok-transformacijom	0,2225 + j0,5653	0,1224 + j0,4513	0,1724 + j0,5083

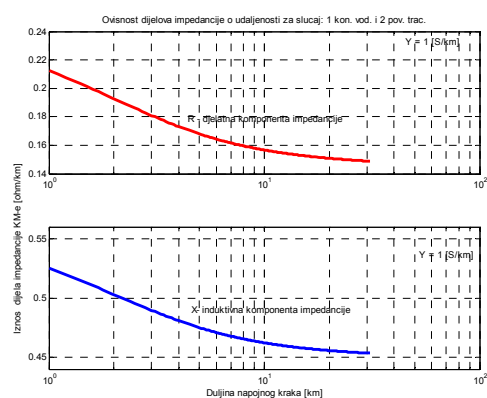
Iz tablice 3.3. i 3.4. je također vidljiva sličnost rezultata kod proračuna blok-transformacijom i postepenim postupkom za dvije povratne tračnice. Ako se usporede srednje vrijednosti impedancije kontaktne mreže iz tablice 3.3. s vrijednošću $Z_{KM} = 0.156 + j 0.475, \Omega/\text{km}$ koja je uzeta u lit.[13] vidimo malo odstupanje posebno djelatne komponente impedancije jer je u lit. [13] proračun izveden za udaljenost opterećenja od EVP-a od 10 km, ali rezultati se slažu s prikazom impedancije u ovisnosti od udaljenosti od EVP-a na slici 2.1 lit [13]. Može se također uočiti malo odstupanje srednje vrijednosti impedancije KM dobivene blok-transformacijom koji se nalazi u tablici 3.4. od vrijednosti $Z_{KM} = 0.145 + j 0.472, \Omega/\text{km}$ koji je dobiven u literaturi [13] za slučaj b) lit [13]. Budući da u daljnjim proračunima koristimo

matrične jednadžbe, proračun blok-transformacijom je prihvatljiviji i iz njega se jasno vide pojedine matrice impedancija nakon svake blok transformacije. Na slikama 3.9. – 3.14. se može uočiti blagi pad i djelatnog i induktivnog dijela impedancije kako se udaljavamo od EVP-a. U daljnjim proračunima se koristi srednja vrijednost impedancije. U programu je uzeto da je najdulji krak napajanja oko 30 km, što je i slučaj za prugu Rijeka – Moravice u slučaju ispada jednog od EVP-a s naznakom da se ta udaljenost može mijenjati po potrebi. Prikaz na sljedećim slikama je dobiven programskim paketom Matlab 6.5. Na crtežima je X – os u logaritamskom mjerilu, a uzet je odvod tračnica od 1 S/km.

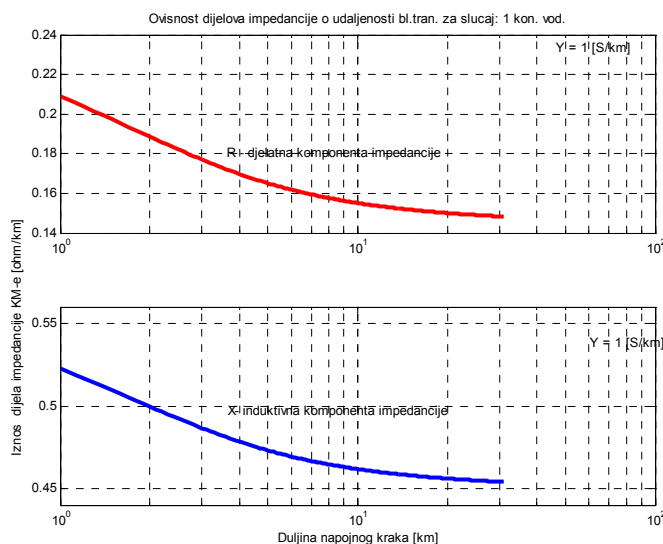
Promjena impedancije KM ovisno o kraku napajanja postepenim postupkom za slučaj kada imamo jedan kontaktni vodič i jednu povratnu tračnicu je prikazana na slici 3.9. Posebno je dana ovisnost djelatnog i induktivnog dijela impedancije. Na slici 3.10. imamo ovisnost impedancije o duljini kraka napajanja za slučaj 1 kontaktnog vodiča i 2 povratne tračnice, dok je na slici 3.11. je dana promjena impedancije za slučaj a) lit. [13] dobivena blok-transformacijom.



Slika 3.9. Ovisnost djelatnog i induktivnog dijela impedancije o duljini napojnog kraka dobiven postepenim postupkom za slučaj: 1 k.v. i 1 p.t.



Slika 3.10. Ovisnost djelat. i ind. dijela impedancije za slučaj: 1 k.v. i 2p.t.



Slika 3.11. Ovisnost djelatnog i induktivnog dijela impedancije o duljini napojnog kraka dobiven blok transformacijom za slučaj a) lit. [13]

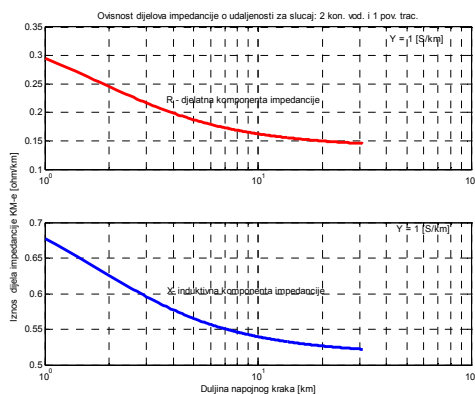
Ako se usporede rezultati na slici 3.9. i slici 3.10. (dobiveni postepenim postupkom) s rezultatima na slici 3.11. (dobiveni blok-transformacijom) vidi se da se oni dosta slični.

Također se može uočiti, da za slučaj kada imamo 2 povratne tračnice, kako se smanje iznosi i djelatnog i induktivnog dijela impedancije.

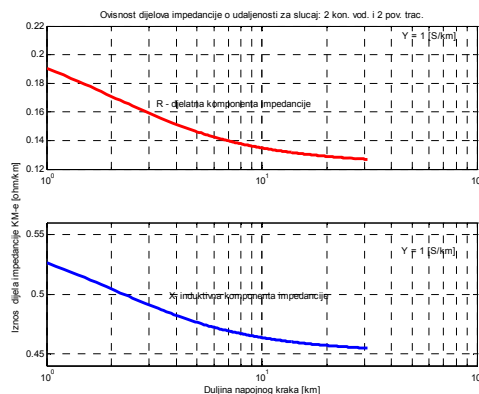
Sada će se napraviti usporedba rezultata dobivenih postepenim postupkom i blok transformacijom za slučaj b) lit [13]. Imamo dvije kombinacije vodiča:

- 2 kontaktna vodiča i 1 povratna tračnica
- 2 kontaktna vodiča i 2 povratne tračnice

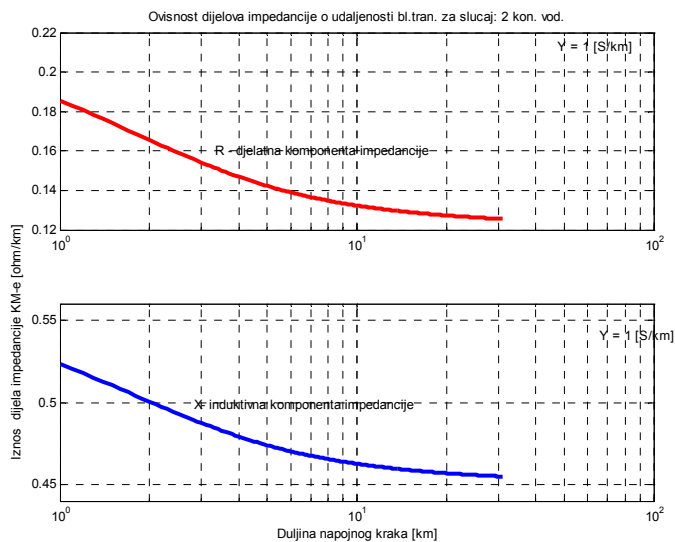
Na slici 3.12. i slici 3.13. su dani rezultati dobiveni postepenim postupkom za ta dva slučaja, a na slici 3.14. je prikaz dobiven blok-transformacijom.



Slika 3.12. Ovisnost djelatnog i induktivnog dijela impedancije o duljini napojnog kraka dobiven postepenim postupkom za slučaj: 2 k.v. i 1 p.t.



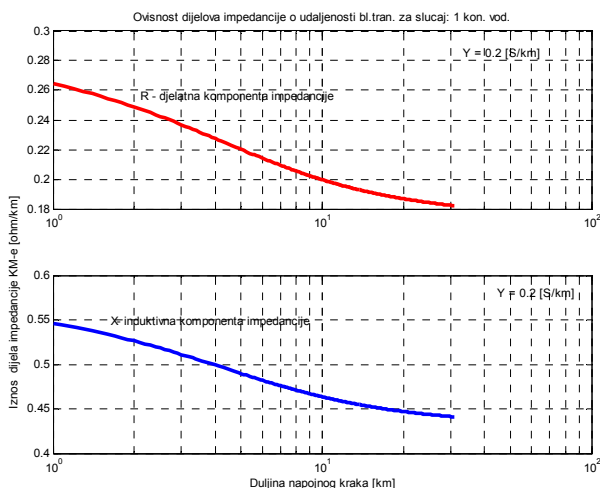
Slika 3.13. Ovisnost djel. i ind. dijela impedancije za slučaj: 2 k.v. i 2 p.t.



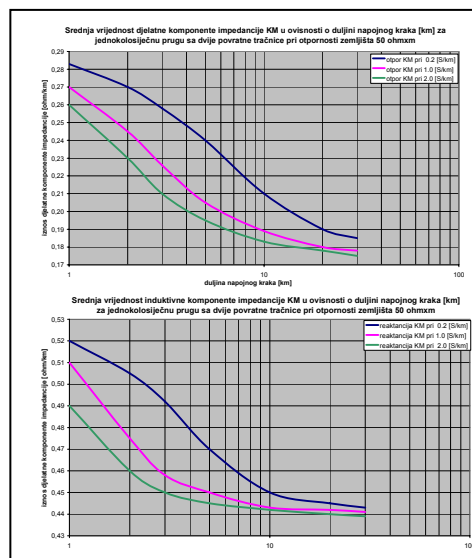
Slika 3.14. Ovisnost djelatnog i induktivnog dijela impedancije o duljini napojnog kraka dobiven blok transformacijom za slučaj b) lit. [13]

Ako se usporede rezultati vidi se da se oni podudaraju tokom cijele duljine napojnog kraka kao i u prethodnom slučaju, a isto vrijedi i za opadanje iznosa pojedinih dijelova impedancije.

Na sljedećim crtežima je prikazan utjecaj odvoda tračnica na iznos srednje vrijednosti impedancije KM-e za slučaj kada je zaštitno uže od bronce presjeka 65 mm^2 (Bz 65). U razmatranje su uzete tri vrijednosti odvoda (0,2 S/km, 1 S/km i 2 S/km) kao u lit.[12] crt.21a da bi se rezultati mogli usporediti. Najprije ćemo prikazati rezultate dobivene za vrijednost odvoda ($Y = 0.2 \text{ S/km}$) blok-transformacijom za slučaj 1 kontaktnog vodiča i 2 povratne tračnice na slici 3.15. Dobiveni rezultati za djelatnu komponentu (crvena krivulja) će se usporedit sa gornjom plavom krivuljom na slici 3.16., a zatim će se napraviti usporedba induktivnog dijela (plava krivulja) sa donjom plavom krivuljom na slici 3.16. lit [12]. Za proračun je uzeta otpornost tla od $50 \Omega\text{m}$.



Slika 3.15. Ovisnost djelatnog i induktivnog dijela impedancije o duljini napojnog kraka dobiven blok transformacijom za slučaj a) lit [12] (1 k.v. i 2 p.t.)



Slika 3.16. Rezultati iz lit.[12]

Ako se usporede rezultati mogu se uočiti sličnosti za obje komponente impedancije. Podudaranje rezultata se dobije i za ostale dvije vrijednosti odvoda (1 S/km i 2 S/km).

Na kraju se dolazi do zaključka da s povećanjem odvoda tračnica smanjuje se iznos impedancije i rezultati se slažu s rezultatima u lit.[12], a da s povećanjem otpornosti tla se povećava srednja vrijednost impedancije KM. Budući da se rezultati dobiveni s postepenim postupkom i blok-transformacijom dosta podudaraju, blok-transformacijski postupak je prihvatljiviji jer se u daljnjim proračunima koriste matricni izrazi. Za daljnje proračune bi se mogla uzeti srednja vrijednost impedancije s ovih crteža. Program omogućuje također i proračun za razne duljine napojnih krakova (ovdje je uzeta duljina od 31 km što je najveća moguća za ovu trasu). Programom je omogućeno proračun impedancija za dva karakteristična slučaja, kada je opterećenje jako blizu EVP-a i kada je ta udaljenost jako velika. Duljina napojnog kraka je prikazana u logaritamskom mjerilu da bi prikaz bio uočljiviji. Posebno su prikazane djelatna i induktivna komponenta jer se njihove vrijednosti dosta razlikuju pa je prikaz pregledniji.

4. Formiranje električne mreže

Kao rezultat simulacije kretanja vlaka dobije se za svaki pojedini vlak jedna izlazna datoteka pod imenom TR"broj".IZL. Npr. za vlak, koji u redu vožnje ima broj 202, kao rezultat simulacije njegovog kretanja dobije se datoteka TR00202.IZL. Analogno, za vlak koji u redu vožnje ima broj 32007 dobije se datoteka TR32007.IZL. Za formiranje električne mreže, a kasnije i proračun električnih prilika u elektrovučnom sustavu nisu značajni svi rezultati simulacije kretanja vlaka. Npr. trenutna brzina vlaka nije značajna za električni proračun, ali je značajna udaljenost vlaka od EVP-a koji ga napaja. Brzina kretanja vlaka, zajedno s drugim parametrima (uspon, težina vlaka, režim vožnje,...) je kvantificirana djelatnom i jalovom snagom koja je potrebna vlaku da bi vozio tom brzinom. Dakle u izlaznoj datoteci TR"broj".IZL su podaci o vremenu polaska i vremenu dolaska vlaka prema redu vožnje, te između ta dva vremena, s korakom 2 sekunde od početka vožnje, udaljenost vlaka od neke referentne točke (referentnog EVP-a) te djelatna i jalova snaga koju trenutno vlak uzima iz kontaktne mreže. Na slici 2.19. je prikazan prvi dio izlazne datoteke koja se dobije simulacijom kretanja vlaka.

Simulacijom kretanja vlaka se dobiju ovakve izlazne datoteke za sve vlakove koji se kreću na nekoj dionici pruge.

Za formiranje električne mreže a kasnije i proračun električnih prilika u elektrovučnom sustavu su potrebni i slijedeći podaci:

- vrijeme početka i vrijeme kraja električnog proračuna;
- vremenski korak električnog proračuna;
- snaga, napon kratkog spoja i broj transformatora u EVP-u odnosno u EVP-ima pri paralelnom radu;
- snaga kompenzacijskih baterija u EVP-u odnosno EVP-ima;
- duljine krakova napajanja pojedinog EVP-a odnosno svih EVP-a;
- podaci o kontaktnoj mreži odnosno impedanciji kontaktne mreže;

Ovi podaci se ili unose u kompjuterski program ili su u datotekama iz kojih će biti učitani. Sada se može formirati električna mreža unutar zadanog vremena i sa zadanim vremenskim korakom električnog proračuna. U svakom koraku se nakon formiranja električne mreže provodi proračun električnih prilika. Sada slijedi objašnjenje kako se formira električna mreža.

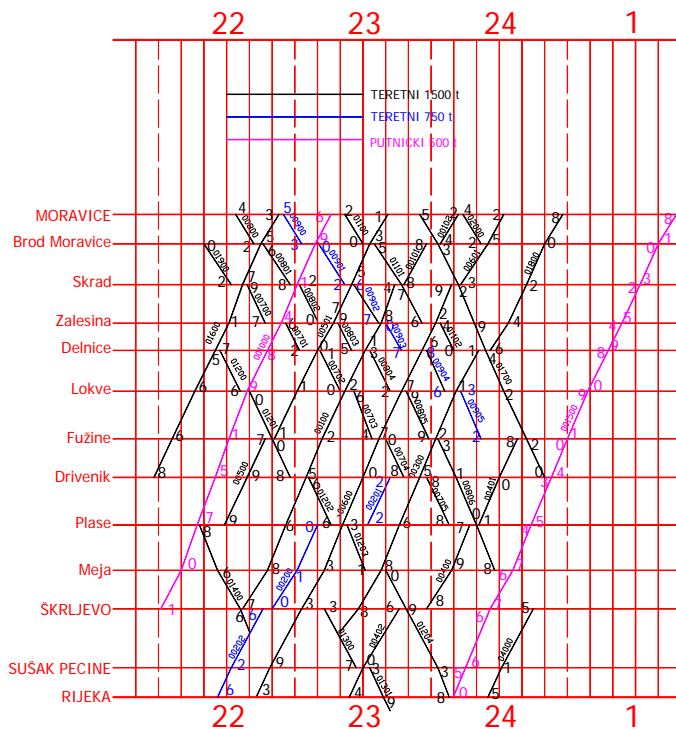
Prvo se provjerava, u određeno vrijeme t , da li pojedini vlak vozi. Ako vlak vozi u to vrijeme otvori se njegova datoteka i nađe podatak o udaljenosti od EVP-a i djelatnoj i jalovoj snazi koju uzima iz kontaktne mreže u danom trenutku. Da bi ovakav sustav pretraživanja funkcionirao vremenski korak mora biti višekratnik koraka simulacije kretanja vlaka, koji je 2 sekunde. Dakle vremenski korak formiranja električne mreže mora biti 2, 4, 6, 8, 10, 12... sekundi. Ako pojedini vlak u određeno vrijeme ne vozi, dakle nije krenuo ili se već zaustavio, taj vlak neće ući u formiranje električne mreže. Provodi se i provjera da li je pojedini vlak još uvijek na kraku napajanja predmetnog EVP-a. Pri paralelnom radu više EVP-a krakovi pojedinih EVP-a se zbroje te se ova provjera provodi na sličan način.

Nakon što se provede gornja procedura za sve vlakove po redu vožnje, pronađeni su svi vlakovi koji se u danom trenutku nalaze na krakovima napajanje EVP-a. Svaki vlak je u danom trenutku jedno čvorište električne mreže. EVP ili EVP-i pri paralelnom radu su također čvorišta električne mreže s tim da je vlak potrošačko čvorište, a EVP čvorište izvor. Sada se sva čvorišta sortiraju s obzirom na udaljenost od referentne točke (referentnog EVP-a). Na ovaj način je formirana električna mreža i može se prijeći na proračun električnih prilika.

4.1. Algoritam formiranja električne mreže

Proračun električnih prilika u elektrovučnom sustavu zahtjeva određene podatke koji mu služe kao osnova za proračun. Poznato je da se za svaku električnu mrežu, za koju se vrši proračun, moraju znati podaci o izvorima u toj mreži, potrošačima koje ti izvori napajaju, konfiguraciji mreže, te raznim drugim više i manje važnim činiteljima točnosti proračuna. Kod proračuna električnih prilika u elektrovučnom sustavu situacija je na određeni način specifična jer se shema napajanja mijenja općenito u svakom trenutku. Pojedini vlakovi dolaze u i izlaze iz "zanimljivog" područja napajanja, pojedini vlakovi se zaustavljaju pa na taj način nestaju kao potrošačko čvorište. Na drugoj strani ispad pojedinog EVP-a mijenja strukturu napajanja i broj čvorišta izvora.

Dakle jasno je da algoritam za formiranje električne mreže mora imati podatke o vlakovima koji se kreću prugom u onome intervalu vremena za koji se vrši proračun. Nadalje na raspolaganju moraju biti podaci o EVP-ima koji napajaju vlakove na dotičnoj pruzi, te podaci o krakovima napajanja pojedinih EVP-a. S ovim podacima, uz zadan maksimalni vozni red vlakova (slika 4.1.), se može formirati električna mreža i uz poznate konstante vodova kontaktne mreže vršiti proračun.

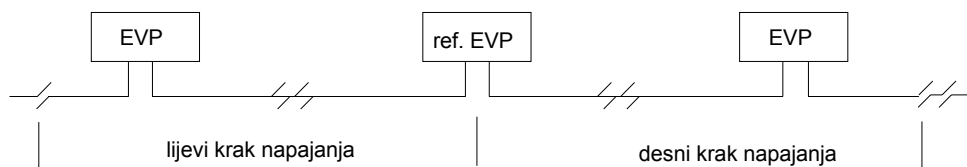


Slika 4.1. Grafikon voznog reda za prugu Rijeka - Moravice

Kada se imaju podaci o svim vlakovima koji se kreću "zanimljivom" prugom može se krenuti u formiranje električne mreže, a to nam omogućuje drugi potprogram (PEPES) o kojem će biti govora kasnije. Naravno ako nas zanima elektrovučni proračun a time i formiranje el. mreže u kratkom intervalu, npr. između 19 i 20 sati, dovoljno je imati navedene datoteke samo za one vlakove koji se kreću u to vrijeme dotičnom prugom. Zatim se unosi vrijeme promatranja el. prilika i korak u kojemu se hoće formirati el. mrežu odnosno vršiti el. proračun. Kao zadnji značajan podatak za formiranje el. mreže su podaci o EVP-ima, njihovom broju, krakovima napajanja svakog od njih, te koliko ih je u paralelnom radu. Sada se može početi s formiranjem el. mreže, unutar unesenog vremena i po zadanom vremenskom koraku pomoću drugog potprograma.

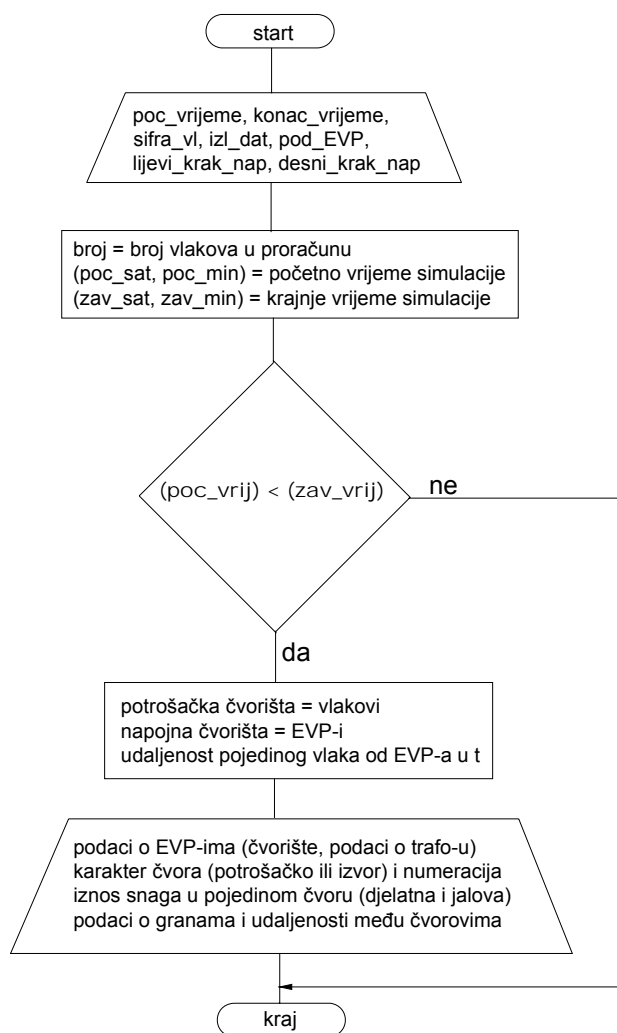
Dakle, u određenom trenutku t se provjerava da li se pojedini vlak nalazi na pruži. Ako se nalazi na pruži pretražuje se njegova datoteka dok se ne dođe do podatka koji odgovara trenutku t . Taj podatak se zapamti i prelazi na slijedeći vlak. Ako neki vlak nije u tom trenutku na pruži pretraživanje njegove datoteke se ne vrši. Na taj način se dobiju svi vlakovi koji će "učestvovati" u formiranju el. mreže.

Iz podataka o EVP-ima i njihovom načinu rada dolazi se do iznosa lijevog i desnog kraka napajanja. Ako se promatra jedan EVP onda su njegov lijevi i desni krak napajanja ujedno i krakovi napajanja. Ako ima dva ili više EVP-a u paralelnom radu onda se kao prvo jedan EVP usvoji za referentni. Ako se lijevo od njega nalazi jedan ili više EVP-a u paralelnom radu onda će lijevi krak napajanja biti lijevi krak referentnog EVP-a uvećan za duljine krakova svih EVP-a lijevo od njega. Analogno vrijedi i za desni krak napajanja. Paralelni rad EVP-a se izbjegava ako nije neophodno jer se pojavljuju struje izjednačenja zbog različitog faznog pomaka napona napajanja. Paralelni rad se koristi jedino ako u mreži postoje kritični padovi napona (ispod 19 kV). Time se postiglo povećanje kraka napajanja za slučaj jednog ili više EVP-a u paralelnom radu, što je značajno za daljnje korake formiranja el. mreže. Na slici 4.2. se nalazi shematski prikaz napajanja nekog referentnog EVP-a koji je korišten u algoritmu za formiranje kontaktne mreže.



Slika 4.2. Shematski prikaz krakova napajanja jednog referentnog EVP-a

Sada se određuje koliko se EVP-a nalazi na lijevom a koliko na desnom kraku napajanja. Treba napomenuti da se može desit slučaj da vlak vozi u promatranom trenutku, ali da nije niti na jednom kraku napajanja (jer se udaljio) pa kao takav nije zanimljiv. Kada se odredi koliko se stvarno vlakova nalazi na lijevom i desnom kraku napajanja vrši se sortiranje po njihovoj udaljenosti od referentne EVP-e. Na taj način se dobiju potrošačka čvorišta el. mreže u trenutku t . Sada se lako iz tih podataka i iz podataka o EVP-a, koji su uneseni, formiraju datoteke potrebne za el. proračun. U ovom potprogramu se koriste samo logički operatori usporedbe. Blok dijagram za ovaj algoritam, koji je implementiran u potprogramu PEPES, je prikazan na slici 4.3. U njemu se nalazi ukratko slijed operacija koje su korištene za formiranje kontaktne mreže.



Slika 4.3. Blok dijagram algoritma za formiranje električne mreže

4.2. Implementacija algoritma u potprogramu PEPES i prikaz izlaznih rezultata

Ovaj algoritam je izveden pomoću drugog potprograma zvanog PEPES. Njegov ispis se nalazi u prilogu 2., a opis algoritma slijedi u daljnjem tekstu.

Prvo se otvara datoteka SIFRE u koju su upisane šifre svih vlakova koji se kreću prugom odnosno za koje imamo datoteke tr"šifra".izl. Varijabli **broj** će se pridružiti broj vlakova u datoteci SIFRE. Varijabla **poziv** sadrži podatak koliko puta se formirala el. mreža odnosno vršio proračun. Kada na pruzi nema vlakova ne vrši se proračun i broj takvih koraka se smješta u varijablu **preskok**. Kada se odredi, koliko se vlakova nalazi u datoteci SIFRE tada se čitaju šifre vlakova u polje **sifravlaka** i pišu u polje **ime_dat** s odgovarajućim formatom. Ovim se kao elementi polja **ime_dat** dobiju imena datoteka tr"šifre".izl. Dalje slijedi provjera da li postoje izlazne datoteke za sve vlakove koji su nabrojani u datoteci SIFRE. Ako ne postoji bilo koja od njih dalje odvijanje programa je onemogućeno.

Zatim slijedi provjera da li postoji datoteka VRED.RES. Naime, u ovu datoteku se smještaju analitički rezultati proračuna. Brisanje slijedi ako se želi nova datoteku za rezultate, a u suprotnom se rezultati proračuna samo nadopisuju u staru datoteku. U varijable **poc_sat** i **poc_min** se upisuje vrijeme početka proračuna odnosno vrijeme od kojeg se hoće proračun el. prilika. Za te dvije varijable, naravno, postoje i odgovarajuća ograničenja. Slično vrijedi i za varijable **zav_sat** i **zav_min**. Vrijeme završetka promatranja mora biti veće od vremena početka promatranja. Varijabla **korak_t** sadrži vrijeme koraka proračuna. Varijable **krak_minus** i **krak_plus** poprimaju u ovom koraku vrijednost krakova napajanja referentnog EVP-a. Ako postoji samo jedan EVP ove vrijednosti će biti ujedno i krakovi napajanja. Ako imamo više EVP-a u paralelnom radu, tada će se ove vrijednosti korigirati. Varijabla **broj_lijavo** poprima vrijednost broja EVP-a koji su u paralelnom radu lijevo od referentnog. Polja **k_m_pom_lijavo** i **k_p_pom_lijavo** za svaki takav EVP sadrže njegove krakove napajanja. I za njih, naravno, vrijede ograničenja uvedena za krakove da se minus ili lijevi krak unosi kao negativan broj a plus ili desni kao pozitivan. Zbog EVP-a u paralelnom radu **krak_minus** se "produljuje" za krakove koje imaju ti EVP-i. U polje **udalj_minus** se smještaju i udaljenosti EVP-a od referentnog, a također i udaljenosti vlakova, ali u oba slučaja, na minus kraku. U ovoj fazi su smještene samo udaljenosti EVP-a. Polje **udalj_m_pom** također sadrži udaljenosti paralelnih EVP-a na minus kraku. Analogno vrijedi i za EVP-e u paralelnom radu desno od referentnog. Tu isto postoje analogne varijable i polja : **broj_desno**, **k_m_pom_desno**, **k_p_pom_desno**, **udalj_plus** i **udalj_p_pom**.

Dijelom programa označenim s "Brisanje starih datoteka" se brišu datoteke nastale prethodnim izvršenjem programa. Ovo se odnosi na datoteke za grafički prikaz rezultata proračuna, koje imaju ekstenziju SLK. Zatim slijedi otvaranje datoteka **U_L_KRAK.SLK** i **U_D_KRAK.SLK**. Ove datoteke će sadržavati podatke o naponu krajnjeg lijevog i krajnje desnog čvorišta. Ovi podaci su zanimljivi pri antenskom radu EVP-a a kako ovaj program može da proračunava oba načina rada oni su, zbog općenitosti, zadržani. Ispitivana su dva načina komunikacije između potprograma PEPES i GAUS (potprogram za električni proračun). O tome će biti više riječi kasnije ali se odmah može napomenuti da su naponi krajnjih točaka, kojima se aproksimiraju naponi krakova napajanja, odnosno njihove grafičke datoteke riješene na sasvim različit način od ostalih grafičkih datoteka.

Varijable **sat**, **min** i **sek** sadrže globalno vrijeme. Ono se inicijalno postavlja na početno vrijeme sadržano u varijablama **poc_sat** i **poc_min**. Od linije 50 počinje dio programa koji se izvršava onoliko puta koliko je potrebno da se od početnog vremena do završnog vremena stigne **korak_t** korakom.

Slijede linije za uvećanje varijable **poziv** i ispis aktualnog vremena na ekran.

Varijabla **no** je brojač vlakova koji se kreću prugom u danom trenutku. Slijedi DO petlja po svim vlakovima upisanim u datoteku SIFRE i ona završava nešto prije linije komentara :

*** određivanje koliko se vlakova nalazi na "-" kraku napajanja ***

U toj petlji se uzimaju u razmatranje, jedna za drugom, datoteke tr"šifra".izl. Dakle, otvori se prva datoteka po redoslijedu kako su imena tih datoteka smještene u polje **ime_dat**. Čita se prva linija datoteke koja sadrži : šifru vlaka, vrijeme polaska i vrijeme dolaska u krajnju stanicu. Vrijeme polaska vlaka se smješta u varijable **pol_sat** i **pol_min** a vrijeme dolaska u **dol_sat** i **dol_min**. Sada slijedi provjera, da li se vlak nalazi na pruzi u trenutku promatranja. Naravno, provjera se vrši usporedbom s globalnim vremenom (**sat**, **min**, **sek**). Logička varijabla **lo1** poprima vrijednost .true. ili .false. ovisno o tome da li se vlak nalazi na pruzi ili ne. Provjera se provodi odgovarajućom kombinacijom if, elseif i else uvjeta. Ako se vlak u danom trenutku nalazi na pruzi (**lo1=.true.**) potrebno je, čitajući njegovu datoteku, pronaći onaj podatak koji odgovara globalnom vremenu. U tu svrhu postavlja se lokalno vrijeme, okarakterizirano varijablama **sat_lok** i **min_lok**, na vrijeme polaska vlaka upisano u varijable **pol_sat** i **pol_min**. Varijabla **k** ima ulogu kontrole da li se traženi podatak našao. Dakle, otvara se DO WHILE petlja sa kontrolom pomoću varijable **k**. Sukcesivno se čita jedan po jedan red datoteke tr"šifra".izl. Svaki red sadrži vrijeme u sekundama od polaznog vremena, poziciju vlaka prema referentnom EVP-u te njegovu djelatnu i jalovu snagu. Za to čitanje služe varijable **real_time**,

pozicija, Pel, Qel. Dalje slijedi niz naredbi za ispravan rad lokalnog sata. Varijable **d_min** i **d_sat** imaju funkciju povećanja minuta i sati lokalnog vremena za 1. Poslije svakog čitanja slijedi kontrola da li je lokalno vrijeme jednako globalnom odnosno da li se došlo do odgovarajućeg podatka. Kada je lokalni sat jednak globalnom, po sekundama, minutama i satima, došlo se do odgovarajućeg podatka.

Sada se brojač vlakova no povećava za jedan i u polje slogova, **teret**, pišu relevantni podaci : šifra vlaka, njegova djelatna i jalova snagu, udaljenost od referentnog EVP-a, te oznaka 3 kao interna oznaka za vlak.

Poslije ovog pisanja slijedi kontrola udaljenosti vlaka od referentnog EVP-a. Naime, ako je udaljenost između dva čvorišta el. mreže premala to može izazvati problem pri njenom rješavanju, odnosno singularitet [Y] matrice. Zbog toga se vlak koji je u trenutku proračuna u neposrednoj blizini referentnog EVP-a pomjera nazad ili naprijed 10 m što neće izazvati grešku u proračunu a time se izbjegavaju problemi oko singulariteta. Nakon ovoga može se zatvoriti datoteka. Ako se prilikom čitanja dođe do kraja datoteke, bez da se nađe traženi podatak, skokom na liniju 113 datoteka se zatvara. Ovaj slučaj se ne bi smio dogoditi, ali se može eventualno dogoditi ako u datoteci nisu podaci za sve točke između polaznog i dolaznog vremena ili ako korak simulacije vlaka nije djelitelj **korak_t**. DO petlja omogućava provedene iste procedure za sve vlakove sadržane u datoteci SIFRE.

Slijedi dio programa za određivanje koliko se čvorišta nalazi na "-" kraku napajanja. Ponovo slijedi kontrola udaljenosti pojedinih vlakova ali sada od paralelnih EVP-a na "-" kraku. Jednostavno se s dvije petlje ide, jednom po EVP-ima, drugom po vlakovima, i kontrolira ta udaljenost. Ako je ona manja od 1 m vlak se pomjera 10 m unazad. Varijabla **br_minus** sadržava broj čvorišta na lijevom kraku. U čvorišta spadaju i vlakovi i paralelni EVP-i. Zbog toga je **br_minus** inicijalno postavljen na broj paralelnih EVP-a lijevo od referentnog, **broj_lijevo**. Sada se ide po svim vlakovima koji su na pruzi i provjerava da li su na minus kraku. Uvjeti koji su postavljeni su jasni i jednostavni. Vlak je na minus kraku ako je njegova udaljenost od referentnog EVP-a manja od nule i ako je ta ista udaljenost manja od minus kraka napajanja. Važno je još jednom napomenuti da su sve udaljenosti na lijevom kraku negativni brojevi pa su i postavljeni uvjeti sukladno napisani.

Slična procedura se provodi i za desni krak. Oznake varijabli su analogne oznakama za lijevi krak tako da nije potrebno nešto posebno reći. Jedina razlika je u tome što je ovdje iskorištena ista DO petlja, po i, i za dodjeljivanje udaljenosti paralelnih EVP-a iz pomoćnog polja i za kontrolu udaljenosti vlakova od istih. Vlakovi se eventualno pomjeraju 10 m naprijed.

Slijedi sortiranje čvorišta na lijevom i desnom kraku prema njihovoj udaljenosti od referentnog EVP-a. Za sortiranje je korištena najjednostavnija metoda sa dvije petlje i uspoređivanjem jednog člana sa svim preostalim članovima koji još nisu složeni. Za zamjenu je korištena varijabla **stavi**. Sortiranje na "-" kraku se vrši od najudaljenijih čvorišta pa prema onima koja su bliža referentnom EVP-u. Nasuprot tome, sortiranje na plus kraku se vrši od čvorišta bližih referentnom EVP-u pa prema daljima. Ovo je posljedica ideje da se mreža formira s lijeva na desno što ima određenih prednosti u nekim daljim aplikacijama.

Kada je sortiranje završeno prelazi se na ispis datoteka potrebnih za treći potprogram GAUS (elektrovučni proračun). Potrebno je ispisati slijedeće datoteke: VRED.TF, VRED.NOD i VRED.CN. Sadržaj tih datoteka je prikazan na slijedećim slikama:

3	5.50	15.00	0.00
---	------	-------	------

Slika 4.4. Sadržaj datoteke VRED.TF

U datoteku VRED.TF (slika 4.4.) upisuju se podaci o EVP-ima odnosno energetskim transformatorima kojima se predstavljaju. U nju trebaju biti upisana imena čvorišta u kojima je energetski transformator, te napon kratkog spoja i snaga transformatora. Pojam ime čvorišta će biti objašnjen pri opisu slijedeće datoteke.

```
0.0025 27.5

1 3 6.205 4.740 6400
2 3 0.250 0.121 7400
3 1 0.000 0.000 0
4 3 0.250 0.121 5901
5 3 0.376 0.252 7600
```

Slika 4.5. Sadržaj datoteke VRED.NOD

U datoteku VRED.NOD (slika 4.5.) su već upisani ε (točnost iterativnog postupka) i Unaz (nazivni napon sekundarne strane transformatora). Dalje treba upisati imena čvorišta, njihov karakter, te u slučaju vlaka, djelatnu i jalovu snagu te šifru vlaka, a u slučaju transformatora, na mjestu djelatne i jalove snage te šifre upisuju se nule. Ispis ove dvije datoteke se vrši paralelno, zbog sličnih uvjeta koji se postavljaju. U datoteci VRED.NOD su čvorišta obilježena brojevima 1, 2, 3 ... i ti brojevi predstavljaju imena čvorišta. Pod pojmom karakter čvorišta ovdje se misli na to da li je to potrošačko čvorište (vlak) ili čvorište izvor (transformator). Potrošačko čvorište se obilježava s 3, a čvorište izvor s 1.

```
0.145 0.472 0.0 0.0 0.0 0.0
0.0 0.0 0.24 0.72 0.0 0.0
0.0 0.0 0.0 0.0 0.006 0.05

0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 100.0 0.0

1 2 3.900
2 3 13.867
3 4 1.820
4 5 0.352
```

Slika 4.6. Sadržaj datoteke VRED.CN

U datoteku VRED.CN (slika 4.6.) ispisuju se podaci o granama, odnosno udaljenosti između pojedinih čvorišta među kojima postoji "veza". Dakle, otvori se datoteka VRED.CN. Prvo se vrši pozicioniranje u deveti red čitanjem prvih osam redova. U tim redovima su upisane matrica uzdužnih impedancija i poprečnih admitancija vodiča kontaktne mreže. Ispis se obavlja po sličnom principu s lijeva na desno. Dakle, DO petljom se ide po svim čvorištima na lijevom kraku i ispisuje udaljenost između dva susjedna. Kada se dođe do posljednjeg čvorišta lijevog kraka ($i = \mathbf{br_minus}$), koje je ujedno i najbliže referentnom čvorištu ispisuje se samo njegova udaljenost jer je udaljenost referentnog čvorišta 0. Ispis se vrši u obliku ime jednog, ime drugog čvorišta te udaljenost u km između njih. Po istom principu se vrši i ispis čvorišta na desnom kraku s razlikom da su čvorišta na desnom kraku sortirana od bližih ka daljima, pa postoji "veza" između referentnog i prvog čvorišta desnog kraka. Kontrola s obzirom na to je riješena if else petljom kao i u slučaju lijevog kraka samo drugim uvjetima.

Slijedi objašnjenje ispisa spomenutih datoteka: Kao što se vidi, iz datoteke VRED.NOD se prvo čitaju dva reda čime se praktično pozicionira u treći red. Već je rečeno u prvom redu su napisani ε i Unaz, zatim slijedi prazan red i od trećeg reda kreće ispis čvorišta. Varijabla **k** služi kao brojač transformatora a varijabla **yes** ima vrijednost 0 ako je to transformatorsko čvorište a 1 ako je potrošačko. Otvara se DO petlja po j od 1 do **br_minus**, dakle po svim čvorištima na lijevom kraku. Varijabla yes se postavlja na 0, a DO petlja se kreće po svim vlakovima.

Uzme se **udalj_minus(j)** pa se ide po svim vlakovima. Ako se nađe vlak s tom udaljenošću, on se ispisuje na ekran i u datoteku VRED.NOD, a yes se postavlja na 1 što znači da je to potrošačko čvorište. Ako dano čvorište nije potrošačko onda je transformatorsko (yes=0) i ispisuju se na ekran i u datoteku VRED.NOD podaci za transformatorsko čvorište a u datoteku VRED.TF podaci tog transformatora. U poljima **transformatori_Uk** i **transformatori_snage** smješteni su naponi kratkih spojeva i snage transformatora. Na taj način se u datoteku VRED.NOD ispisuju sva čvorišta na lijevom kraku a u datoteku VRED.TF podaci o transformatorima. Slijedi ispis podataka referentnog EVP-a na ekran i u spomenute datoteke.

Potpuno ista procedura se provodi za desni krak, naravno s varijablama i poljima koja njega karakteriziraju (**br_plus**, **udalj_plus** ...).

Formatima 891 i 991 su prilagođeni ispisi u datoteke VRED.NOD i VRED.TF.

Datoteka VRED.TIM nije potrebna za električni proračun odnosno nju ne zahtjeva program GAUS ali se u nju piše aktualno vrijeme. Varijabla **time** sadrži vrijeme u sekundama od početka promatranja. Slijedi kontrola da li ima vlakova na pruzi i ako ima poziva se program GAUS. Ako nema vlakova na pruzi varijabla **preskok** se povećava za jedan i vrši se skok na liniju 159, jer nema ni rezultata el. proračuna.

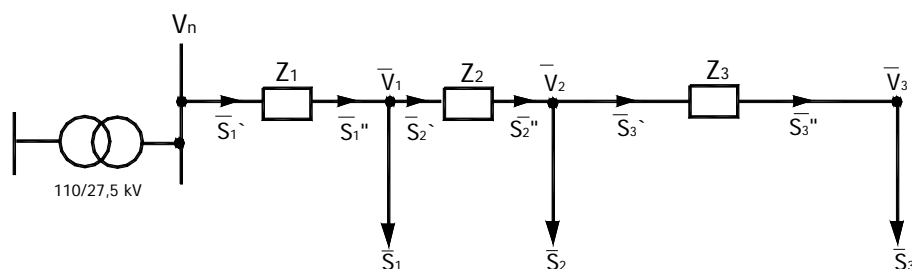
5. Proračun električnih prilika u elektrovučnom sustavu

Za određivanje strujno-naponskih prilika u kontaktnoj mreži korišten je program za proračun padova napona čiji je matematički model zasnovan na metodi čvorišta. Ispis algoritma, koji je napisan u Matlabu, kao i rezultati za navedeni slučaj se nalaze u prilogu 4. Podaci za transformatore u EVP-u i kontaktnu mrežu preuzeti su iz literature [10]. Opterećenja vlakova modelirana su konstantnim snagama utvrđenim na temelju njihove brzine i trenutnog položaja prema podacima iz voznog reda.

Za primjer proračuna padova napona uzet je dio kontaktne mreže koji se napaja iz EVP Moravice u trenutku 22h 23 min 40 sec. U tom trenutku su se na dionici koju napaja ovaj EVP našla tri vlaka s ulaznim podacima koji su prikazani u tablici 5.1. Vlakovi se nalaze na desnom kraku EVP-a, a shematski prikaz je predložen na slici 5.1. Impedancija kontaktne mreže preuzeta iz literature [10] iznosi: $Z_{KM} = 0,145 + j 0,472 \Omega/\text{km}$.

Tablica 5.1. Stvarna opterećenja na dionici pruge Rijeka – Moravice koji se napaja iz EVP Moravice

Šifra vlaka	Udaljenost od EVP-a , m	Djelatna snaga , MW	Jalova snaga , (Mvar)
1600 (Driv – Mor)	900,1	0,250	0,121
801 (B.Mor. – Skrad)	14575,7	5,918	4,521
1000 (Skrlj-Mor)	24568,1	0,397	0,262



Slika 5.1. Shema opterećenja voznog voda za proračun pada napona duž voda opterećenog u više točaka

Pad napona u pojedinom čvorištu KM-e će se izračunati iterativnim putem koristeći shemu na slici 5.1. pretpostavljajući u nultoj iteraciji da su naponi u pojedinim čvorištima jednaki.

$$\begin{aligned} V_1^{(0)} &= V_n, \\ V_2^{(0)} &= V_n, \\ V_3^{(0)} &= V_n. \end{aligned} \quad (5.1)$$

Za razliku snaga među čvorištima se koriste slijedeći izrazi:

$$\Delta S_3 = \frac{(P_3'')^2 + (Q_3'')^2}{V_3^2} R_3 + j \frac{(P_3'')^2 + (Q_3'')^2}{V_3^2} X_3$$

$$S_3' = S_3'' + \Delta S_3$$

$$S_2'' = S_3' + S_2$$

$$\Delta S_2 = \frac{(P_2'')^2 + (Q_2'')^2}{V_2^2} R_2 + j \frac{(P_2'')^2 + (Q_2'')^2}{V_2^2} X_2 \quad (5.2)$$

$$S_2' = S_2'' + \Delta S_2$$

$$S_1'' = S_2' + S_1$$

$$\Delta S_1 = \frac{(P_1'')^2 + (Q_1'')^2}{V_1^2} R_1 + j \frac{(P_1'')^2 + (Q_1'')^2}{V_1^2} X_1$$

$$S_1' = S_1'' + \Delta S_1$$

Sada će se pomoću snaga koje izlaze iz čvorišta (S_1' , S_2' i S_3' tj. njihovih komponenti P i Q) odrediti padovi napona između pojedinih čvorišta pomoću sljedećih izraza:

$$\begin{aligned} \Delta V_1 &= \frac{P_1' R_1 + Q_1' X_1}{V_1} + j \frac{P_1' X_1 - Q_1' R_1}{V_1} \\ \Delta V_2 &= \frac{P_2' R_2 + Q_2' X_2}{V_2} + j \frac{P_2' X_2 - Q_2' R_2}{V_2} \\ \Delta V_3 &= \frac{P_3' R_3 + Q_3' X_3}{V_3} + j \frac{P_3' X_3 - Q_3' R_3}{V_3} \end{aligned} \quad (5.3)$$

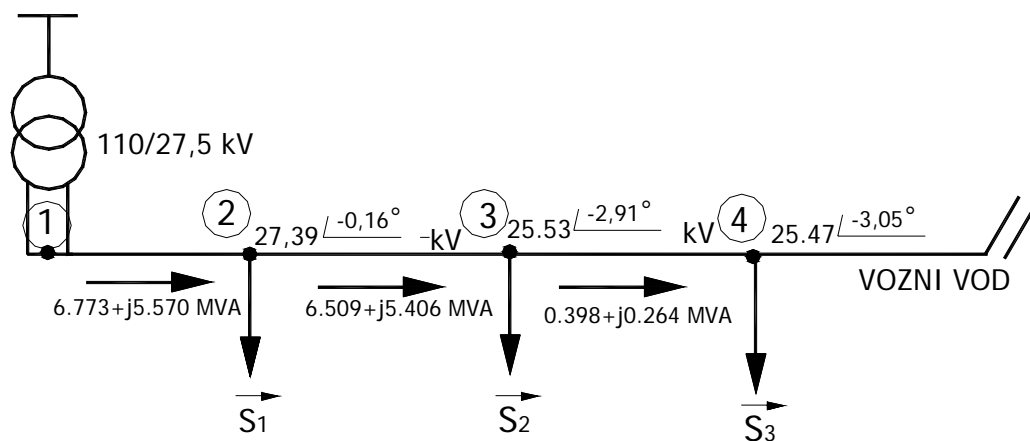
Na kraju se odrede naponi u n -toj iteraciji pomoću izraza:

$$\begin{aligned} V_1^{(n)} &= V_n - \Delta V_1^{(n-1)} \\ V_2^{(n)} &= V_1^{(n)} - \Delta V_2^{(n-1)} \\ V_3^{(n)} &= V_2^{(n)} - \Delta V_3^{(n-1)} \end{aligned} \quad (5.4)$$

Rezultati proračuna dobiveni postepenim postupkom se nalaze u tablici 5.2. i može se zaključiti da je pad napona na kraju voda oko 2 kV, što se slaže s padom napona na ovoj dionici i u danom trenutku (1,96 kV) koji je dobiven programom za proračun snaga i kratkog spoja u elektroenergetskoj mreži (GAUS 4). Rezultati dobiveni pomoću Matlaba se nalaze u prilogu 4, a ujedno su grafički prikazani na slici 5.2.

Tablica 5.2. Prikaz napona u čvorištima u pojedinoj iteraciji

Br. iter.	ΔV_1 , kV	ΔV_2 , kV	ΔV_3 , kV	V_1 , kV	V_2 , kV	V_3 , kV
0.	0,0	0,0	0,0	27,5	27,5	27,5
1.	0,11375	1,67328	0,06332	27,38625	25,71297	25,64965
2.	0,11565	1,81235	0,06793	27,38435	25,57200	25,50407
3.	0,11578	1,82436	0,06832	27,38422	25,55986	25,49154
4.	0,11579	1,82540	0,06836	27,38421	25,55881	25,49045
5.	0,11579	1,82549	0,06836	27,38421	25,55872	25,49036



Slika 5.2. Grafički prikaz dobivenih rezultata (napona i tokova snaga) u pojedinom čvoru

Proračunom padova napona mogu se odrediti naponi na priključcima (pantografima) svih vlakova koji se u određenom trenutku napajaju iz promatranog EVP-a. Na temelju izračunatih napona mogu se izračunati struje u kontaktnim vodovima, odnosno tokovi snaga u kontaktnoj mreži, te utvrditi eventualna ograničenja. Navedeni proračun padova napona, stoga, u potpunosti zadovoljava prilikom planiranja kontaktne mreže.

5.1. Proračun tokova snaga primjenom matrične metode

Nakon modeliranja elemenata mreže elektrificiranih željeznica može se preći na rješavanje problema tokova snaga, koji se kao i u elektroenergetskim mrežama zapravo svodi na određivanje napona čvorišta promatrane mreže. Osnovu za proračun čini metoda čvorišta, koja se zapravo svodi na rješavanje matrične jednadžbe:

$$[I] = [Y] \times [V] \quad (5.1.1)$$

pri čemu je:

$[I]$ – vektor injektiranih struja u čvorištima kontaktne mreže

$[V]$ – vektor napona čvorišta

$[Y]$ – matrica admitancije čvorišta

S obzirom na vrstu napajanja kontaktne mreže postoje i određene razlike, kako prilikom formiranja same $[Y]$ matrice tako i prilikom rješavanja problema tokova snaga.

Kontaktnu mrežu sačinjavaju čvorišta i grane, te elementi sustava spojeni u čvorišta. Čvorišta su mjesta gdje se elementi sustava priključuju na kontaktnu mrežu. Pri tome su neka čvorišta fiksna, kao npr. mjesto priključka EVP-a, a neka promjenjivog karaktera, kao npr. trenutni položaj vlakova na pruzi. Za ta čvorišta mijenja se ne samo prostorna koordinata već ona mogu u određenim slučajevima, (zaustavljanje vlaka u stanici), potpuno nestati, da bi se opet nakon određenog vremena pojavila u mreži (polazak vlaka iz stanice). Grane mreže povezuju pojedina čvorišta. Njihova se duljina i broj također mijenjaju u toku proračuna, zbog promjenjivog karaktera nekih čvorišta kontaktne mreže.

U sustavu 1x25 kV, za kontaktnu mrežu od n čvorišta može se za i -to čvorište napisati slijedeća jednadžba:

$$I_i^c = \sum_{\substack{j=1 \\ j \neq i}}^n (V_i^c - V_j^c) \cdot y_{i,j}^{cc} + \sum_{\substack{j=1 \\ j \neq i}}^n (V_i^r - V_j^r) \cdot y_{i,j}^{cr} + (V_i^c - V_i^r) \cdot y_{i,i}^{cr} \quad (5.1.2)$$

$$I_i^r = \sum_{\substack{j=1 \\ j \neq i}}^n (V_i^c - V_j^c) \cdot y_{i,j}^{rc} + \sum_{\substack{j=1 \\ j \neq i}}^n (V_i^r - V_j^r) \cdot y_{i,j}^{rr} + (V_i^r - V_i^c) \cdot y_{i,i}^{rc} \quad (5.1.3)$$

Admitancije $y_{i,j}^{cc}$, $y_{i,j}^{cr}$ obuhvaćaju u sebi uzdužne admitancije te poprečne admitancije voznog voda prema zemlji. Admitancija $y_{i,i}^{cr}$ predstavlja elemente koji se u kontaktnoj mreži nalaze između voznog voda i tračnice.

Sređivanjem prethodnih jednadžbi dobiva se za ovaj sustav :

$$I_i^c = (V_i^c) \times \left(\sum_{\substack{j=1 \\ j \neq i}}^n y_{i,j}^{cc} + y_{i,i}^{cr} \right) + (V_i^r) \times \left(\sum_{\substack{j=1 \\ j \neq i}}^n y_{i,j}^{cr} - y_{i,i}^{cr} \right) + \sum_{\substack{j=1 \\ j \neq i}}^n (V_j^c) \times (-y_{i,j}^{cc}) + \sum_{\substack{j=1 \\ j \neq i}}^n (V_j^r) \times (-y_{i,j}^{cr}) \quad (5.1.4)$$

$$I_i^r = (V_i^c) \times \left(\sum_{\substack{j=1 \\ j \neq i}}^n y_{i,j}^{rc} - y_{i,i}^{rc} \right) + (V_i^r) \times \left(\sum_{\substack{j=1 \\ j \neq i}}^n y_{i,j}^{rr} + y_{i,i}^{rc} \right) + \sum_{\substack{j=1 \\ j \neq i}}^n (V_j^c) \times (-y_{i,j}^{rc}) + \sum_{\substack{j=1 \\ j \neq i}}^n (V_j^r) \times (-y_{i,j}^{rr}) \quad (5.1.5)$$

Analognim postupcima sređuju se i preostale jednadžbe te se konačno može, za zadanu kontaktnu mrežu, napisati sljedeća matrična jednadžba:

$$\begin{bmatrix} I_1^{(cr)} \\ I_2^{(cr)} \\ \dots \\ I_n^{(cr)} \end{bmatrix} = \begin{bmatrix} \left[\sum_{j=1}^n y_{1,j}^{(cr)} \right] \left[-y_{1,2}^{(cr)} \right] \dots \left[-y_{1,n}^{(cr)} \right] \\ \left[-y_{2,1}^{(cr)} \right] \left[\sum_{j=1}^n y_{2,j}^{(cr)} \right] \dots \left[-y_{2,n}^{(cr)} \right] \\ \dots \\ \left[-y_{n,1}^{(cr)} \right] \left[-y_{n,2}^{(cr)} \right] \dots \left[\sum_{j=1}^n y_{n,j}^{(cr)} \right] \end{bmatrix} \times \begin{bmatrix} V_1^{cr} \\ V_2^{cr} \\ \dots \\ V_n^{cr} \end{bmatrix} \quad (5.1.6)$$

Jednostupčani vektori : $[I_1^{(cr)}]$, $[I_2^{(cr)}]$, ..., $[I_n^{(cr)}]$ predstavljaju injektirane struje u zadanim čvorištima mreže, a jednostupčani vektori : $[V_1^{(cr)}]$, $[V_2^{(cr)}]$, ..., $[V_n^{(cr)}]$ napone pojedinih čvorišta prema referentnom čvoru (zemlji) tj. vrijedi sljedeće:

$$\begin{bmatrix} I_i^{(cr)} \end{bmatrix} = \begin{bmatrix} I_i^c \\ I_i^r \end{bmatrix} \quad \begin{bmatrix} V_i^{(cr)} \end{bmatrix} = \begin{bmatrix} V_i^c \\ V_i^r \end{bmatrix} \quad (5.1.7)$$

Dijagonalne podmatrice u matrici admitancije čvorišta navedene u (5.1.6) imaju za i-to čvorište slijedeći oblik:

$$\begin{bmatrix} \sum_{j=1}^n y_{i,j}^{(cr)} \end{bmatrix} = \begin{bmatrix} \sum_{\substack{j=1 \\ j \neq i}}^n y_{i,j}^{cc} + y_{i,i}^{cr} \sum_{\substack{j=1 \\ j \neq i}}^n y_{i,j}^{cr} - y_{i,i}^{cr} \\ \sum_{\substack{j=1 \\ j \neq i}}^n y_{i,j}^{rc} - y_{i,i}^{rc} \sum_{\substack{j=1 \\ j \neq i}}^n y_{i,j}^{rr} + y_{i,i}^{rc} \end{bmatrix} \quad (5.1.8)$$

Vandijagonalne podmatrice imaju slijedeći oblik:

$$\begin{bmatrix} -y_{i,j}^{(cr)} \end{bmatrix} = \begin{bmatrix} -y_{i,j}^{cc} - y_{i,j}^{cr} \\ -y_{i,j}^{rc} - y_{i,j}^{rr} \end{bmatrix} \quad (5.1.9)$$

Energetski transformator, kojim je kontaktna mreža vezana za elektroenergetski sustav, je modeliran s dva jednaka naponska izvora i reaktancijom kratkog spoja. Za proračun se naponski izvori pretvaraju u strujne prema izrazu $I_{TR} = 2 \cdot U_{TR} / Z_{TR}$ pri čemu je Z_{TR} impedancija kratkog spoja energetskog transformatora.

Teret (vlak) modeliran je u proračunu impedancijom, koja se mijenja u ovisnosti o naponu na koji je priključen vlak. Naime, cijeli postupak rješavanja matrice jednadžbe određene izrazom (5.1.1) odvija se iterativno, a impedancija vlaka priključenog između kontaktnog vodiča i tračnice određena je izrazom:

$$Z_{TRAIN} = \frac{[\Delta V_i^{(k)}]^2}{S_{TRAIN}} \quad (5.1.10)$$

pri čemu je:

$\Delta V_i^{(k)} = (V_i^c - V_i^r)^{(k)}$ - razlika napona između kontaktnog vodiča i tračnice u čvorištu i
u k-toj iteraciji

S_{TRAIN} – snaga vlaka u i-tom čvorištu

Ova se impedancija vlaka pretvara u admitanciju, te se dodaje u svakom koraku u Y matricu kontaktne mreže.

Za rješavanje sustava jednadžbi (5.1.1) korištena je Gauss-ova metoda eliminacije, poznata i pod nazivom FORWARD – BACKWARD metoda jer se pokazalo da se ta metoda može primijeniti za ovaj sustav napajanja, a potrebni broj iteracija i brzina konvergencije postupka su sasvim zadovoljavajući. Kako se ovaj proračun mora izvoditi ciklički s programom za slaganje električne mreže, u proračunu bi trebalo koristiti rezultate proračuna (napone čvorišta) iz prethodnog koraka, čime bi se znatno ubrzala konvergentnost postupka.

Kratak opis iterativnog postupka dan je u slijedećih osam koraka:

1. KORAK – Unos ulaznih podataka za proračun (podaci o kontaktnoj mreži, energetskim transformatorima, vlakovima)
2. KORAK – Postavljanje početnih vrijednosti za proračun (naponi i impedancije vlakova u 0-toj iteraciji)
3. KORAK – Formiranje $[Y]$ matrice
4. KORAK – Modifikacija $[Y]$ matrice
5. KORAK – Određivanje admitancije vlakova iz vrijednosti napona u svakoj iteraciji i dodavanje u Y matricu
6. KORAK – Rješavanje sustava nelinearnih jednačbi (rješavaju se naponi u svakoj iteraciji)
7. KORAK – Ispitivanje konvergentnosti postupka (ako je $|V_i^{(k+1)} - V_i^{(k)}| < \varepsilon$ onda korak 8 inače povratak na korak 5)
8. KORAK – Ispis rezultata proračuna.

Navedeni algoritam opisuje proračun tokova snaga za slučaj napajanja iz samo jednog EVP-a (radijalni pogon), jer je to uobičajeni način rada za ovaj sustav napajanja. Prelazak na paralelni rad ne izaziva u navedenom algoritmu nikakve globalne promjene.

5.2. Implementacija algoritma za proračun električnih prilika u potprogramu GAUS

Treći potprogram GAUS namijenjen za elektrovučni proračun, za već zadanu mrežu, radi na slijedeći način. Odmah na početku program uključuje dvije datoteke: MCV.PAR i GAU.PAR. U te dvije datoteke su sadržani parametri proračuna te varijable i polja koja se koriste skoro u svim potprogramima. Da se te varijable, a posebno polja, koja su dvodimenzijaska, ne bi prenosila pomoću poziva potprograma ona su definirana kao zajedničke zone - COMMON.

Nakon deklaracija varijabli i polja slijedi dodjela karakternoj varijabli **udat** vrijednosti 'red' što se prvobitno unosilo sa tastature. Dalje slijedi otvaranje datoteke VRED.RES i VRED.POM te određena pozicioniranja u tim datotekama.

Nakon toga se formira $[Y]$ matrica pozivom potprograma YFORM. Početne vrijednosti napona se postavljaju u slijedećem dijelu programa. Kako je program rađen za proračun u sustavu 2 x 25 kV to je u njemu bilo tri vrste vodiča : kontaktni - C, napojni - F i tračnica - R. To nije bilo potrebno korigirati jer na napojni vodič neće biti ništa spojeno. Naravno, red $[Y]$ matrice bi bio manji za trećinu da je ta korekcija izvršena. Vidi se da postoje dva puta po dva polja za napon. To je zbog toga što napon ima i realni i imaginarni dio, a potrebno je, zbog provjere konvergenije, imati napon u dvije susjedne iteracije.

Dalje slijedi određivanje vrijednosti naponskih izvora, koji modeliraju transformatore, i pretvaranje tih izvora u strujne izvore. Iza toga dolazi dio programa za spremanje Y matrice na mjestu vlaka u pomoćne varijable i korekcija $[Y]$ matrice zbog ovisnosti impedancije vlaka o naponu. Kada se to obavilo ide se na rješavanje matrice jednačbe (5.1.1) Gauss-ovom metodom NAPRIJED – NATRAG

Kako je već rečeno matrica $[Y]$ je singularna pa je potrebno, prije njene inverzije, izbaciti jedan čvor. To se vrši pozivom potprograma IZBACI. Inverzija se vrši pozivom potprograma INV a ponovno vraćanje izbačenog čvora pozivom potprograma VRATI. Ako je matrica Y i pored toga singularna varijabla **ks** će dobiti vrijednost 1 što mijenja daljnji tok odvijanja programa. Ako je gornja procedura protekla regularno slijedi izračunavanje vrijednosti napona u 0-toj iteraciji. Nakon toga slijedi ispitivanje konvergentnosti napona. Ispis vrijednosti napona u pojedinim iteracijama program GAUS ne vrši, ali je i taj dio ostao.

Ako je broj iteracija prešao 25, smatra se da metoda ne konvergira. U koliko naponi odstupaju više od zadane točnosti, skokom na liniju 90 ponavlja se opisana procedura, od korekcije $[Y]$ matrice zbog ovisnosti impedancije vlaka o naponu. Kada je postupak određivanja napona završen prelazi se na ispis. Na kraju se zatvaraju sve ranije otvorene datoteke.

Sada slijedi opis pojedinih funkcija koje su korištene u ovom potprogramu.

Funkcija YFORM služi za učitavanje ulaznih podataka iz ulaznih datoteka i formiranje $[Y]$ matrice za proračun tokova snaga.

$[Y]$ matrica se formira uz pomoću četiri funkcije. To su:

Funkcija CVORS - čita datoteku VRED.NOD u koju su upisani ϵ , Unaz, te podaci o čvorištima.

Funkcija TRAFO - u potprogramu GAUS ova funkcija je služila za učitavanje podataka o energetskom transformatoru dok u programu GAUS ona ima i drugih funkcija. Kako može da postoji više energetskih transformatora to je potrebno, prije svega, utvrditi njihov broj i ime čvorišta u kojima su transformatori spojeni. Za to služi varijabla **broj_tr** i polje **pnod**. Za svaki transformator unose se odgovarajuće vrijednosti u $[Y]$ matricu pozivom potprograma DODAJP. Kako su svi transformatori direktno uzemljeni to je potrebno modelirati veze između uzemljenja transformatora. To se postiže dodavanjem u $[Y]$ matricu velike vodljivosti između čvorišta u kojima su spojena uzemljenja transformatora. Odgovarajućim pozivima funkcije DODAJU obavlja se spomenuta rutina. I kao zadnje, u funkciji TRAFO se otvaraju pomoćne datoteke za prijenos rezultata proračuna u program PEPES. Otvara se onoliko pomoćnih datoteka koliko je transformatora plus jedna za napone krakova napajanja.

Funkcija ATF - služi za učitavanje podataka o autotransformatorima ali za proračun u sustavu 1 x 25 KV on nije zanimljiv jer isti ne postoje.

Funkcija VOD - služi da se u $[Y]$ matricu unose vodljivosti vodova kontaktne mreže. Podaci o konstantama vodova kontaktne mreže dani su matricama u datoteci VRED.CN. Matrica serijskih impedancija se prvo invertira a zatim, pomoću funkcije DODAJU, dodaje u $[Y]$ matricu. Poprečne admitancije se jednostavno, pomoću funkcije DODAJP, dodaju u $[Y]$ matricu.

U potprogramu GAUS se mogu naći još slijedeće funkcije neophodne za njegovo funkcioniranje, a spomenute su ranije.

Funkcija IZBACI – služi da bi se jedan čvor, koji je usvojen kao referentni, izbaci iz $[Y]$ matrice prije inverzije. Izbacuje se čvorište na koje je spojeno uzemljenje transformatora, jer je napon tog čvorišta nula. Kako u mreži može biti više transformatora to se ide redom po čvorištima dok se ne dođe do prvog transformatora i izbaci se njegov odgovarajući čvor. U $[Y]$ matrici se to odražava izbacivanjem odgovarajućeg retka i stupca pa je potrebno vršiti i "pomjerenja" odgovarajućih članova matrice.

Funkcija VRATI služi za vraćanje izbačenog čvorišta. U $[Z]$ matrici se to odražava dodavanjem odgovarajućeg retka i stupca. To naravno zahtjeva "pomjerenje" odgovarajućih elemenata matrice.

Funkcije INVERZ i INV obavljaju inverziju matrice, ali INV je prilagođen inverziji $[Y]$ matrice a INVERZ vrši inverziju matrice impedancija kontaktne mreže.

Funkcija DODAJU vrši dodavanje u [Y] matricu podmatrica, koje su sadržane u pozivu funkcije. Naravno, moraju se znati i imena čvorišta između kojih se vrši dodavanje, da bi se mogle odrediti pozicije za dodavanje u [Y] matrici. Kao što i samo ime funkcije kaže ovdje je riječ o dodavanju vodljivosti uzdužnih elemenata. Dakle u dijagonalnu podmatricu matrice Y treba dodati pozitivne vrijednosti a u vandijagonalnu negativne.

Funkcija DODAJP služi da se u [Y] matricu dodaju vodljivosti poprečnih elemenata. One se dodaju s predznakom plus.

Funkcija ISPIS služi pored ispisa datoteke VRED.RES i za izračunavanje struje fidera i njen djelomični ispis u pomoćnu datoteku, te ispis napona krakova napajanja također u pomoćnu datoteku. Izračunavanje struje između dva čvorišta se vrši funkcijom ISVOD. Nizom uvjeta je riješeno izračunavanje struje svakog pojedinog transformatora na obadje strane. Za izračunavanje struje vlaka poziva se funkcija ISVL a za izračunavanje opterećenja transformatora funkcija ISTR.

Funkcija ISVL vrši izračunavanje djelatne i jalove komponente struje, koju uzima vlak, iz poznatog napona čvorišta između kojih je priključen vlak te snage vlaka. Te dvije komponente struje te napon kontaktnog vodiča na mjestu vlaka se ispisuju u datoteku VRED.RES.

Funkcija ISVOD je u potprogramu GAUS prilagođena izračunavanju i ispisu struje fidera. Zbog toga se u njenom pozivu, pored dva čvorišta između kojih se struja računa, navodi i oznaka od kojeg čvorišta prema kojemu se struja računa te za koji je to transformator. Izračunatu struju fidera ovaj potprogram piše u pomoćnu datoteku odgovarajućeg transformatora.

Funkcija ISTR vrši izračunavanje djelatne i jalove komponente struje kao i djelatnog, jalovog i prividnog opterećenja te $\cos\phi$ -a opterećenja transformatora. Podaci o snagama i $\cos\phi$ -u opterećenja transformatora se pišu u odgovarajuću pomoćnu datoteku.

Nakon izvršenog el. proračuna program GAUS je stvorio datoteke EVP1.pom , EVP2.pom ,..., onoliko datoteka koliko je bilo transformatora, te datoteku KRAKOVI.pom u koju su upisani naponi krakova napajanja. Kako uvijek postoji bar jedan transformator (i u slučaju da nema paralelnog rada) datoteka EVP1.pom je sigurno stvorena izvršenjem programa GAUS. Otvara se dakle ta datoteka i čitaju slijedeći podaci :

I_fid_l, I_fid_d - struja fidera lijevo i desno

ptr, qtr, str - djelatno , jalovo i prividno opterećenje transformatora

cos_fi - $\cos\phi$ opterećenja transformatora.

Nakon zatvaranja datoteke u polje slogova **EVP1** pišu se ti podaci zajedno s vremenom na koje se odnose.

Ista procedura se provodi i za sve eventualne transformatore u paralelnom radu. Razlika je jedino u tome što oni ne moraju postojati pa se vrši kontrola da li postoji pomoćna datoteka. Logičke varijable **postoji2** , **postoji3** ,... imaju vrijednost .true. ako postoji odgovarajuća datoteka.

Na sličan način čitaju se iz datoteke KRAKOVI.pom napon lijevog i desnog kraka napajanja. Kao što se vidi, rezultati se u ovom slučaju ispisuju direktno u grafičku datoteku. Sada će se nešto reći o ova dva načina komunikacije koji su spomenuti ranije. Prvobitna zamisao je bila da program GAUS stvara pomoćne datoteke s rezultatima el. proračuna a da se one nakon završenog proračuna čitaju i podaci iz njih odmah pišu u datoteke za grafički prikaz rezultata (ekstenzija SLK). Nakon svakog proračuna bi se u grafičke datoteke dopisivao po jedan red. Kako je predviđeno 5 transformatora u paralelnom radu a za svaki transformator se stvara po 6 datoteka za grafički prikaz, to je ukupno 30 datoteka. DOS ne podržava toliki broj otvorenih datoteka pa je takav koncept zadržan samo za datoteke U_L_KRAK.SLK i U_D_KRAK.SLK

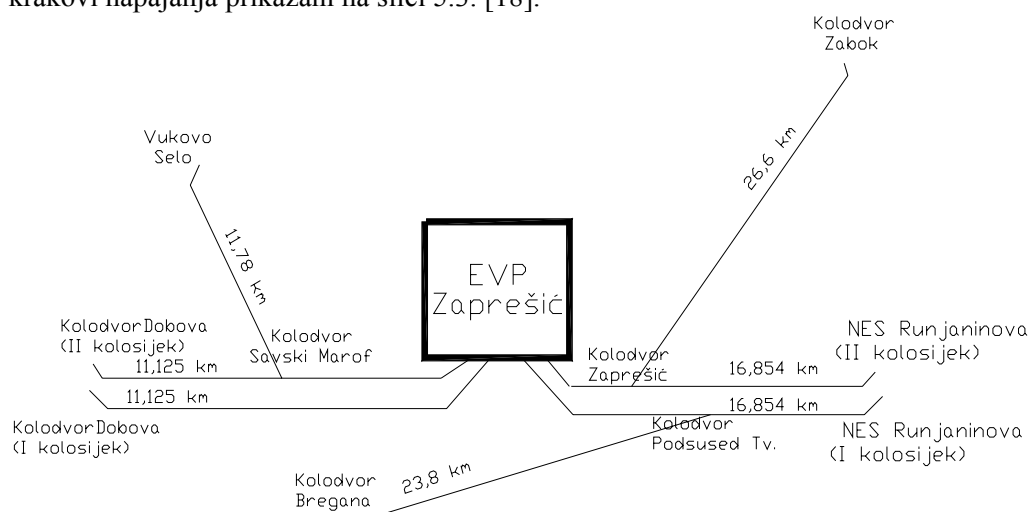
koje sadrže podatke o naponu krakova napajanja. Za sve druge grafičke datoteke rezultati se prvo smještaju u polje slogova a nakon završenog kompletnog proračuna ispisuju u te datoteke. Nakon linije 159 vrši se uvećanje vremena za **korak_t**. Tu naravno postoje naredbe za vođenje globalnog sata. Nakon toga slijedi provjera da li se dostiglo vrijeme završetka proračuna. Ako se nije dostiglo to vrijeme vrši se skok na liniju 50 i cijela opisana procedura se ponavlja.

Kada se dostiglo vrijeme završetka proračuna, prelazi se na ispis grafičkih datoteka. Kako je već ranije navedeno, jedan transformator mora postojati pa se otvara 6 datoteka za prikaz 6 njegovih karakterističnih veličina: P , Q , S , $\cos\varphi$, te struje fidera na jednu i drugu stranu. Datoteke za grafički prikaz rezultata proračuna imaju u prve tri linije određene podatke koje zahtjeva program za crtanje a iza njih slijedi ispis u dvije kolone vremena i vrijednosti veličine u tom trenutku. Dakle, u prvu liniju se upisuje 1 ako je to datoteka prividne snage 2 za $\cos\varphi$, 3 za djelatnu, 4 za jalovu snagu, 5 za struju, a 6 za napon. U drugu liniju se upisuje broj krivulja, što je u ovom slučaju 1, a u treću liniju broj točaka ispisanih u datoteci. Onoliko puta koliko se formirala mreža i koliko se izveo proračun toliko ima točaka u datoteci, pa se u treću liniju SLK datoteke upisuje vrijednost varijable **poziv** umanjene za vrijednost varijable **preskok**, odnosno onoliko puta koliko se nije vršio proračun. Nakon ispisa po gornjim pravilima prva tri reda u SLK datoteke se pišu odgovarajuće vrijednosti ranije smještene u slogove polja **EVPI**. Sada se mogu zatvoriti SLK datoteke.

Ako je **postoji2**=true. znači da postoji transformator u paralelnom radu pa se ponavlja opisana procedura otvaranja upisa i zatvaranja šest njegovih SLK datoteka. **Postoji3**, **postoji4** i **postoji5** se odnose na treći, četvrti i peti transformator. Već je rečeno da je za SLK datoteke krakova napajanja korišten drugi princip i u njih su upisani svi podaci osim broja točaka, pa se nakon povratka na njihov početak, vrši pozicioniranje u treći red i upisuje i taj podatak. Nakon toga i te datoteke se mogu zatvoriti. Zadnjim linijama programa se brišu datoteke sa ekstenzijom POM koje su služile za komunikaciju između programa GAUS i PEPES

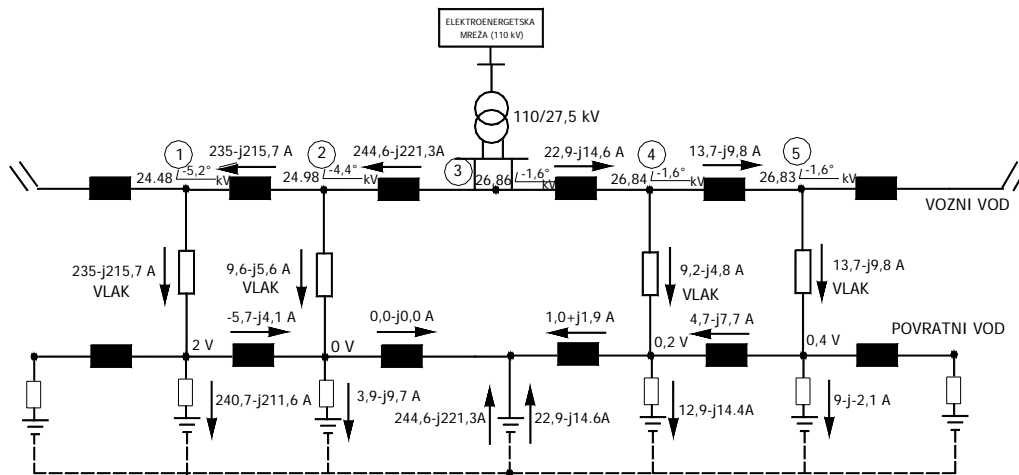
5.3. Prikaz rezultata dobivenih elektrovučnim proračunom kontaktne mreže

Primjena sva tri algoritma će biti pokazana na jednom konkretnom primjeru. U proračun je uzet vozni red vlakova napajanih EVP-om Zaprešić u normalnom pogonu čiji su krakovi napajanja prikazani na slici 5.3. [18].



Slika 5.3. Krakovi napajanja EVP Zaprešić

Proračun je rađen za vremenski interval 6h – 8h. Iz maksimalnih graf0va voznog reda za magistralnu prugu Zagreb G.K. – Dobova i tri priključne pruge (prilozi 5 – 8.) se vidi da u tom intervalu pretežno prometuju elektromotorni vlakovi EMV 6111. Najprije se napravi simulacija kretanja navedenih vlakova, a kao rezultat dobivamo izlazne datoteke formata kao na slici 2.19. Simulacija je rađena pomoću Borland C++-a. Te izlazne datoteke služe kao ulazni podaci za drugi i treći potprogram (PEPES i GAUS). Slika 5.4. prikazuje rezultate jednog proračuna, koji se ponavljaju svake dvije sekunde, tako da se u vremenu od 2 sata provede 3600 takvih proračuna

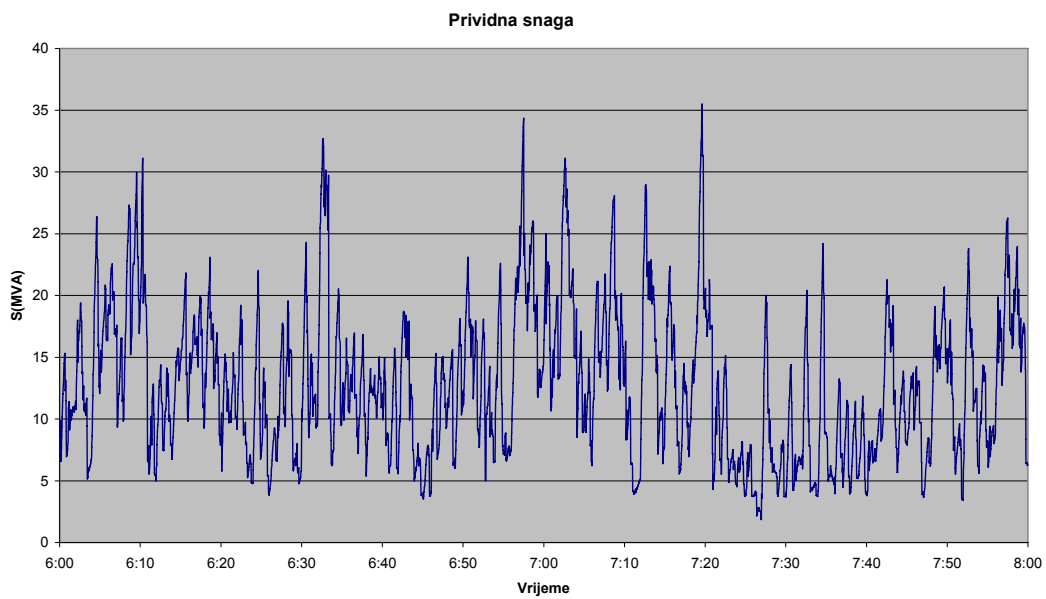


Slika 5.4. Raspodjela struja po granama kontaktne mreže i napona u čvorištima

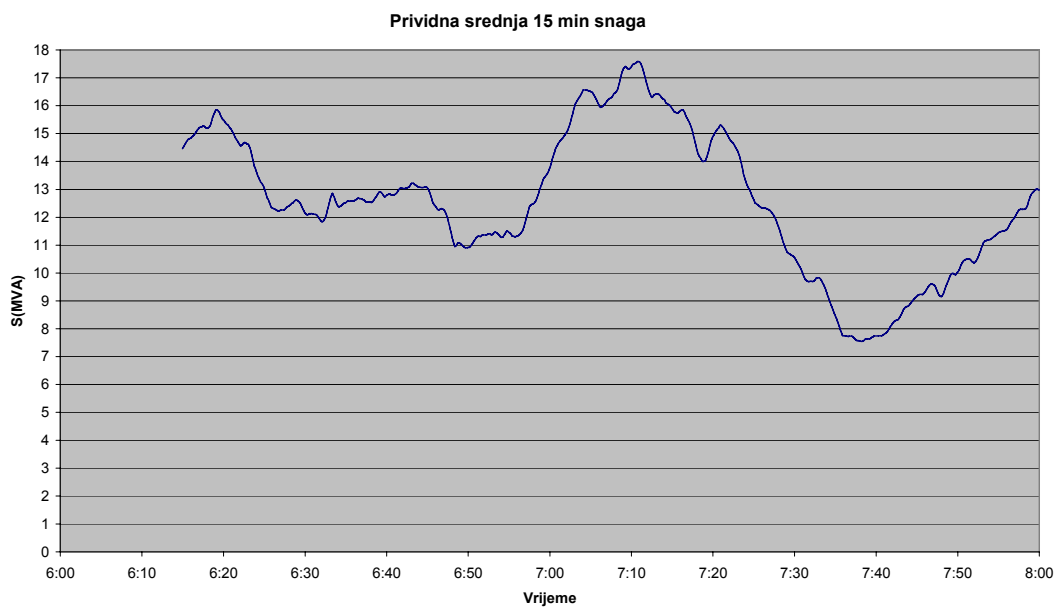
Ulazni podaci za proračun su:

- Impedancija kontaktne mreže: $Z_{KM} = 0,181 + j0,447 \Omega/\text{km}$;
- Energetski transformatori: $2 \times 10 \text{ MVA}$, $u_k = 10\%$
- Elektroprivredna mreža je zamjenjena idealnim naponskim izvorom nazivnog napona 110 kV koji se transformira u omjeru 110/27,5 kV;
- Proračun je proveden sa kompenzacijom jalove snage u kontaktnoj mreži iznosa $Q_c = 1,853 \text{ Mvar}$.

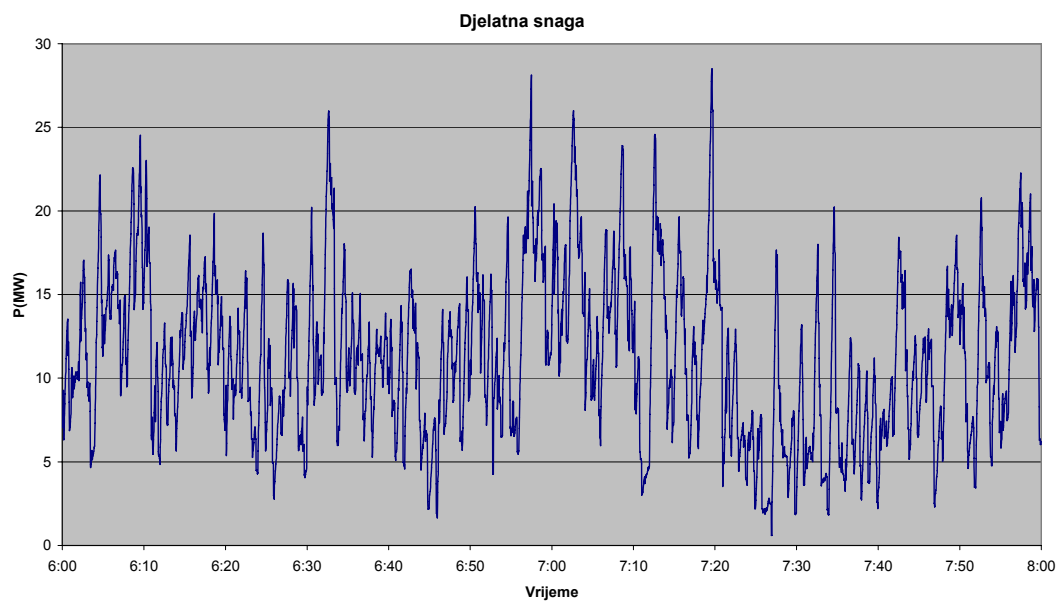
Grafički prikaz pojedinih rezultata dobivenih elektrovučnim proračunom za normalni pogon EVP-a Zaprešić se nalaze na slijedećim slikama.



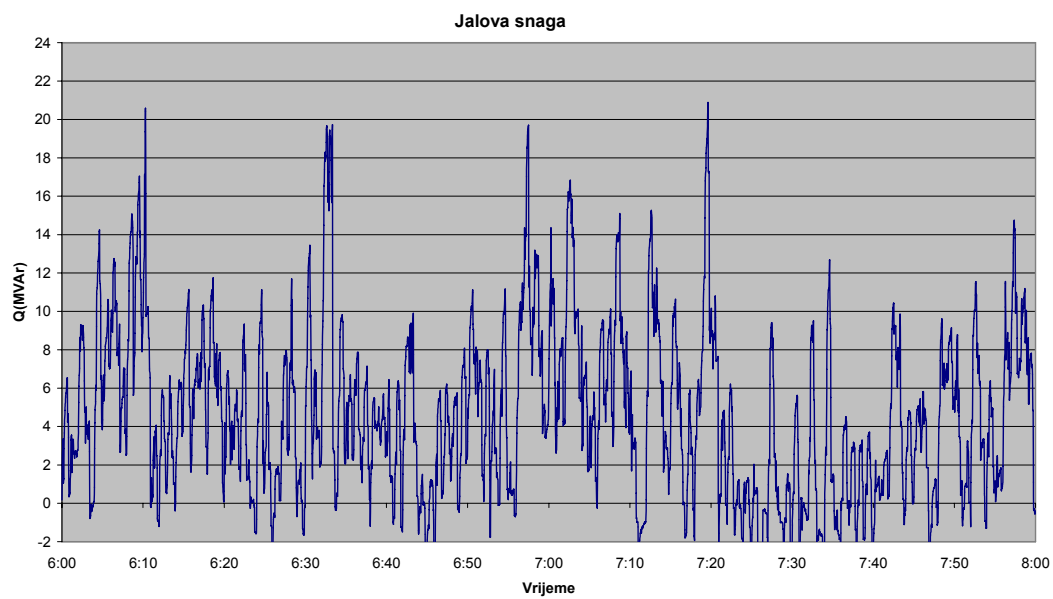
Slika 5.5. Opterećenje EVP Zaprešić prividnom snagom



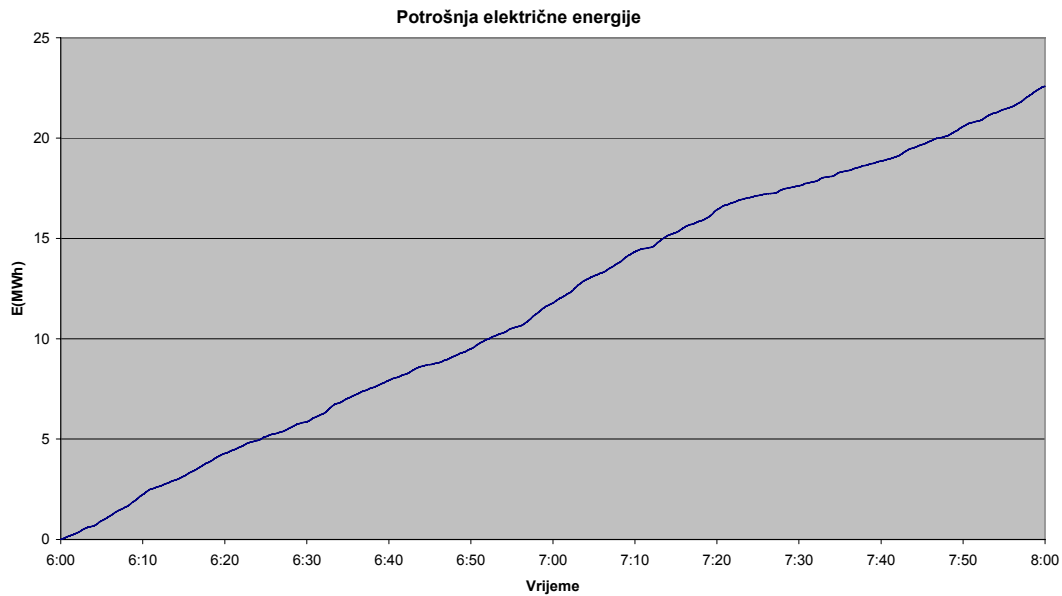
Slika 5.6. Opterećenje EVP Zaprešić prividnom srednjom 15 min. snagom



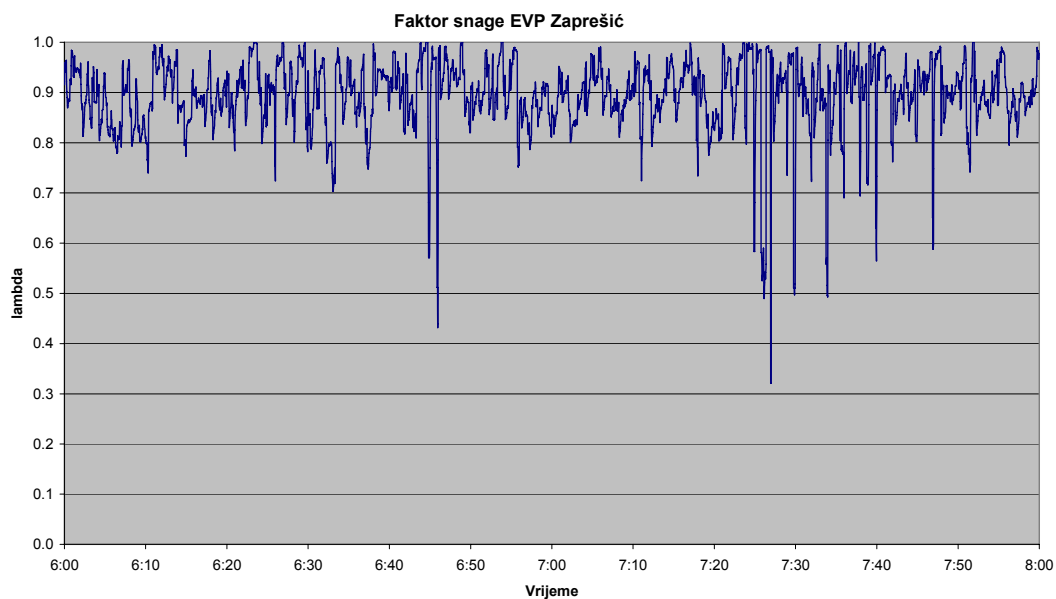
Slika 5.7. Opterećenje EVP Zaprešić djelatnom snagom



Slika 5.8. Opterećenje EVP Zaprešić jalovom snagom (kompenzacija $Q_c=1,853$ Mvar)



Slika 5.9. Potrošnja električne energije EVP Zaprešić od 6 do 8 sati

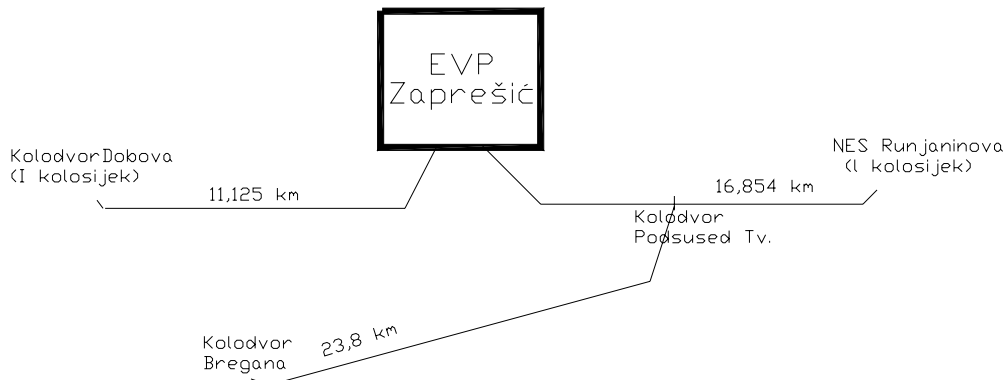


Slika 5.10. Faktor snage EVP Zaprešić (kompenzacija $Q_c = 1,853$ Mvar)

Grafikon na slici 5.6. pokazuje da maksimalna srednja 15-minutna snaga EVP-a Zaprešić za ovaj proračun iznosi 10 MVA, a maksimalna prividna snaga 26 MVA (slika 5.5).

Budući da I kolosijek i II kolosijek magistralne pruge nisu električno povezani, ostali proračuni (pad napona, struje federa i nadtemperatura pojedine dionice) grafički prikazani zasebno za svaki kolosijek. U daljnjim proračunima je uzeto da EVP Zaprešić napaja 1 transformator ($S_n = 15$ MVA, $u_k = 5$ %). Na slici 5.11. su shematski prikazani krakovi napajanja za I kolosijek magistralne pruge Zagreb G.K. – kol. Dobova i kol. Podsused Tv. – kol. Bregana.

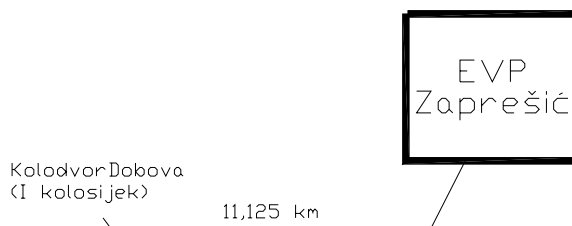
I kolosijek



Slika 5.11. Krakovi napajanja EVP Zaprešić (prema kol. Dobova_I kolosijek i prema kol. Bregana)

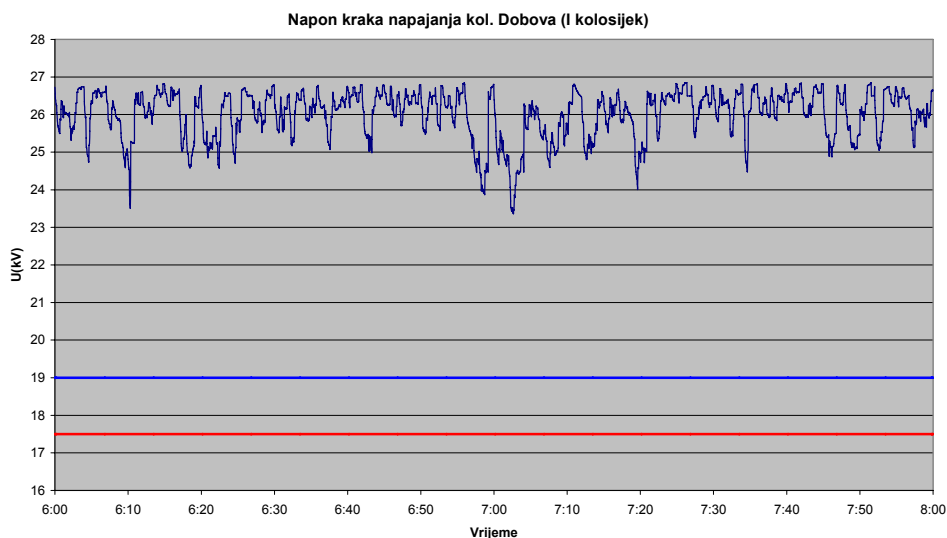
Duljina lijevog kraka napajanja prema kol. Dobova (I kolosijek) je 11,125 km, a duljina desnog kraka napajanja prema kol. Bregana iznosi 23,8 km.

a) Krak napajanja prema kol. Dobova

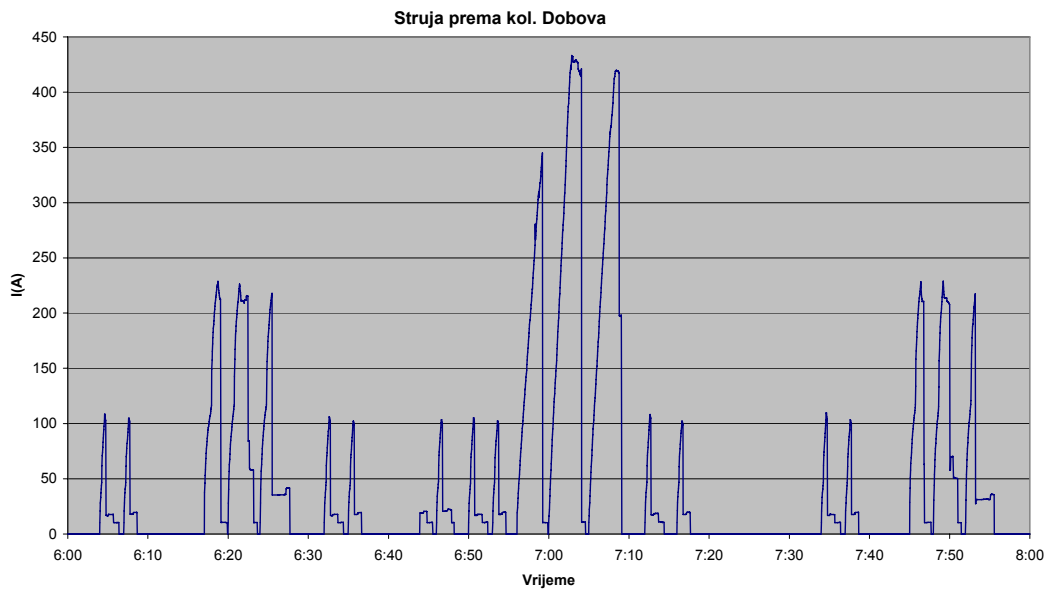


Slika 5.12. Krak napajanja EVP Zaprešić prema kol. Dobova_I kolosijek

Grafički prikaz pada napona, struje fidera i nadtemperature kontaktnog voda za ovu dionicu se nalazi na slijedećim slikama.



Slika 5.13. Napon kraka napajanja prema kol. Dobova_I kolosijek



Slika 5.14. Struja fidera prema kol. Dobova_I kolosijek

Proračun zagrijavanja vodova kontaktne mreže najčešće se provodi za najnepovoljniji slučaj (1 kontakti vodič) uz istrošenost kontaktnog voda od 30%.

Povišenje temperature vodiča kroz koji teče struja (u našem slučaju kontaktnog voda) nastaje zbog djelatnog otpora vodiča. Ako se u prvom trenutku zanemari odvođenje topline (hlađenje), zagrijavanje je proporcionalno toplini koja se stvara u vodiču [4]:

$$G \cdot c \cdot \Delta \vartheta = I_t^2 \cdot R \cdot t \quad (5..3.1)$$

gdje je: G - masa vodiča;
 c - specifična toplina vodiča;
 $\Delta \vartheta$ - povišenje temperature vodiča;
 I_t - efektivna vrijednost struje;
 R - djelatni otpor vodiča;
 t - vrijeme.

Za masu vodiča vrijedi:

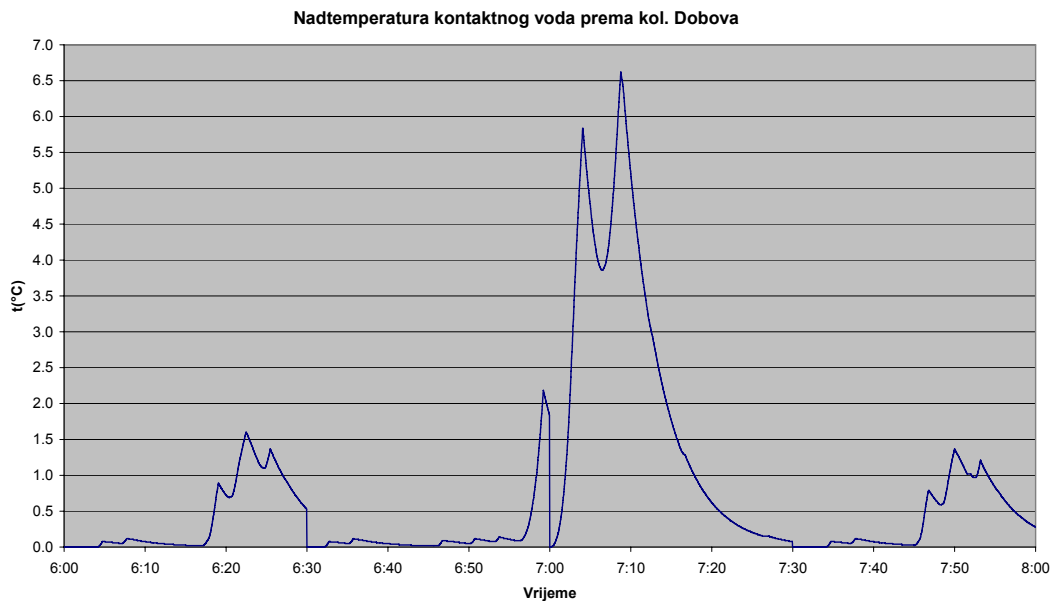
$$G = \gamma \cdot S \cdot l \quad (5..3.2)$$

gdje je: γ - gustoća vodiča;
 S - presjek vodiča;
 l - duljina vodiča.

Ako se sada izraz (5.1.12) uvrsti u (5.1.11), uz poznati izraz za djelatni otpor vodiča, dobije se izraz za određivanje nadtemperature vodiča:

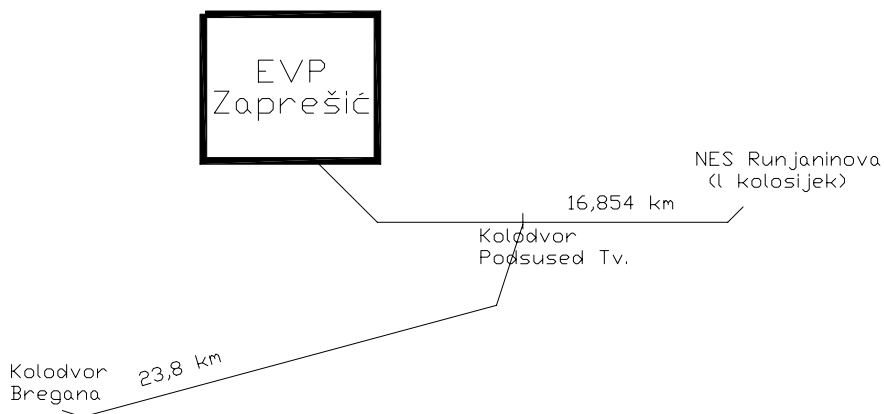
$$\Delta \vartheta = \frac{I_t^2 \cdot \rho \cdot t}{S^2 \cdot \gamma \cdot c} \quad (5..3.3)$$

Grafički prikaz rezultata za nadtemperaturu kontaktnog vodiča na dionici EVP Zaprešić – kol. Dobova_I kolosijek dobivenih električnim proračunom se nalazi na slici 5.15.



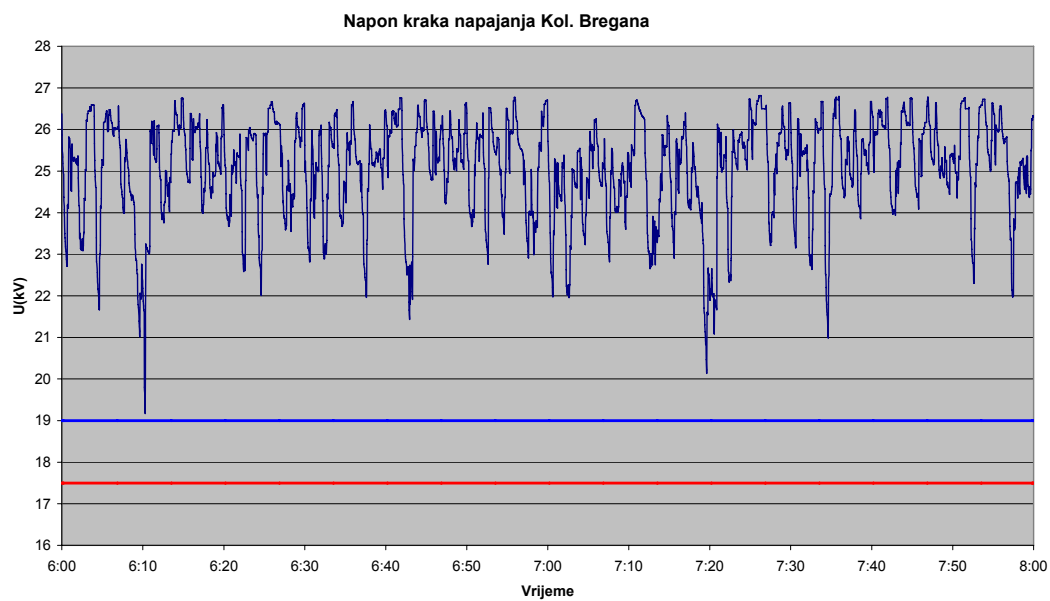
Slika 5.15. Nadtemperatura kontaktnog voda prema kol. Dobova_I kolosijek

b) Krak napajanja prema kol. Bregana

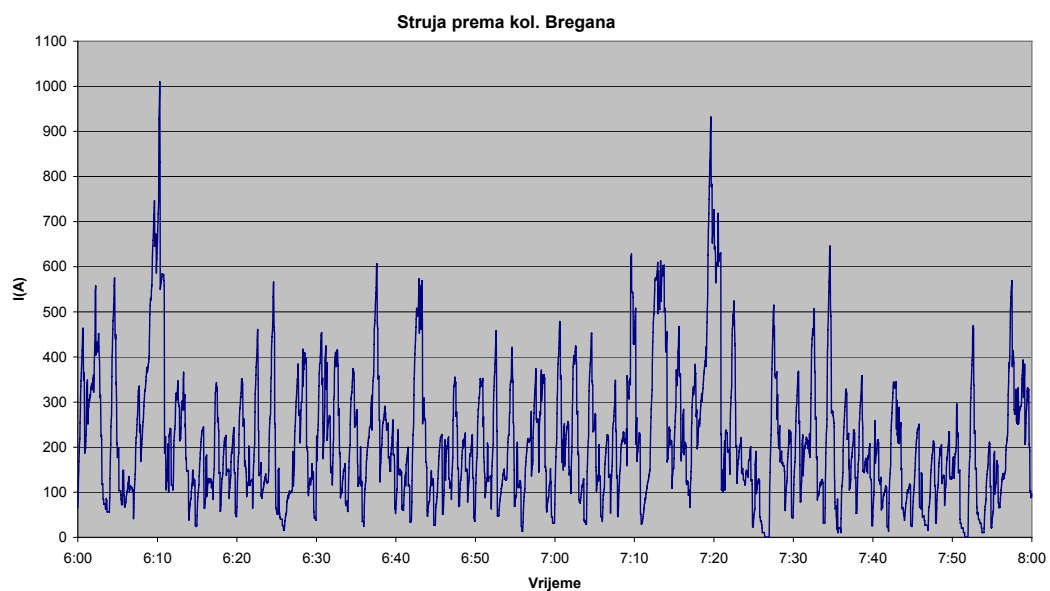


Slika 5.16. Krak napajanja EVP Zaprešić prema kol. Bregana

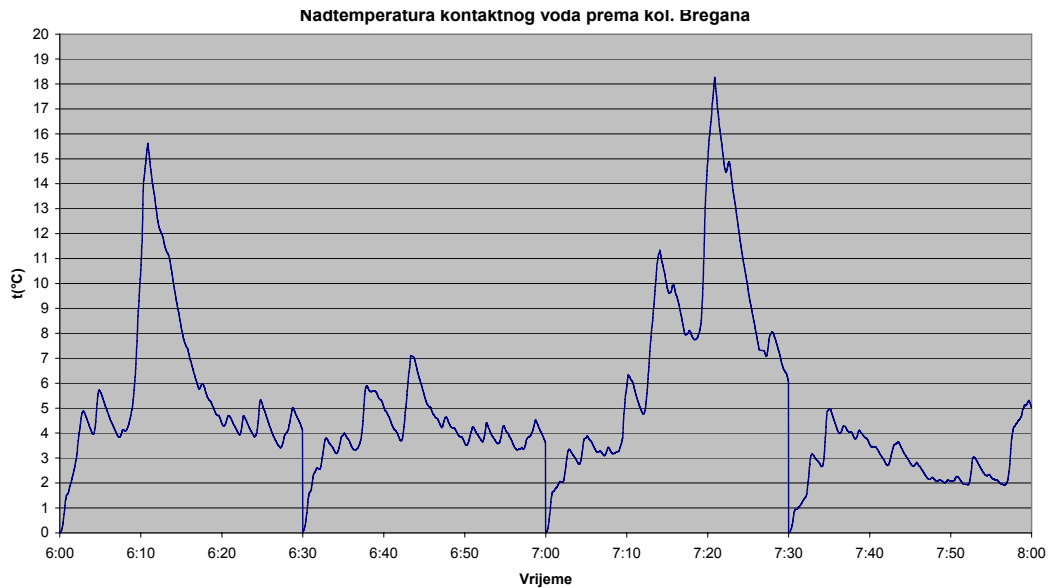
Grafički prikaz pada napona, struje fidera i nadtemperature kontaktnog voda za ovaj krak napajanja je se nalazi na sljedećim slikama.



Slika 5.17. Napon kraka napajanja prema kol. Bregana



Slika 5.18. Struja fidera prema kol. Bregana

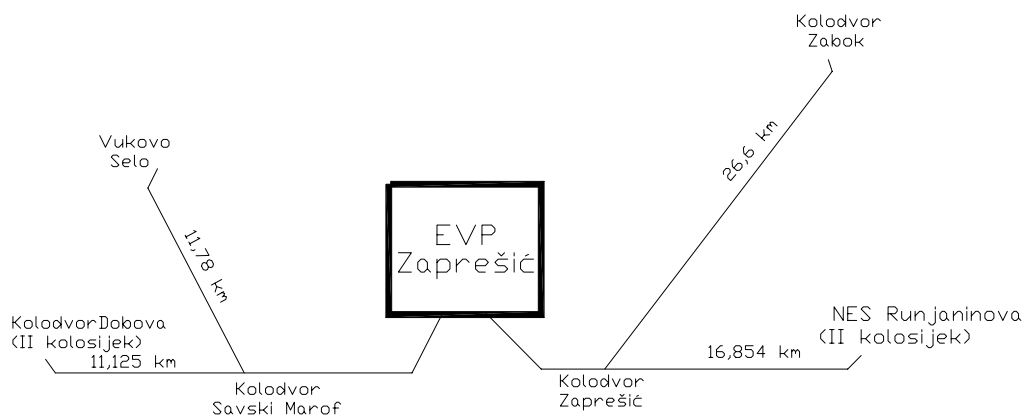


Slika 5.19. Nadtemperatura kontaktnog voda prema kol. Bregana

Iz struja kontaktnog voda prema pojedinom kraku i nadtemperature vodiča se vidi da je maksimum opterećenja dosta veći na kraku prema kol. Bregana (1000 A).

II kolosijek

Na slici 5.20. su shematski prikazani krakovi napajanja za II kolosijek magistralne pruge i pripadne priključne pruge.

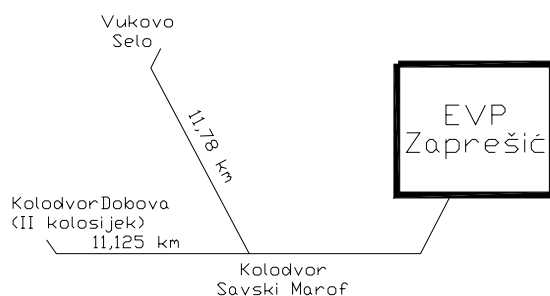


Slika 5.20. Krakovi napajanja EVP Zaprešić prema Vukovom Selu i kol. Zaprešić

Slika 5.20. pokazuje da se desni krak napajanja sastoji od pruge EVP Zaprešić – NES Runjaninova _II kolosijek i pruge kol. Zaprešić – kol. Zabok, a lijevi krak napajanja se sastoji od pruge EVP Zaprešić – kol. Dobova_II kolosijek i pruge kol. Savski Marof – Vukovo Selo.

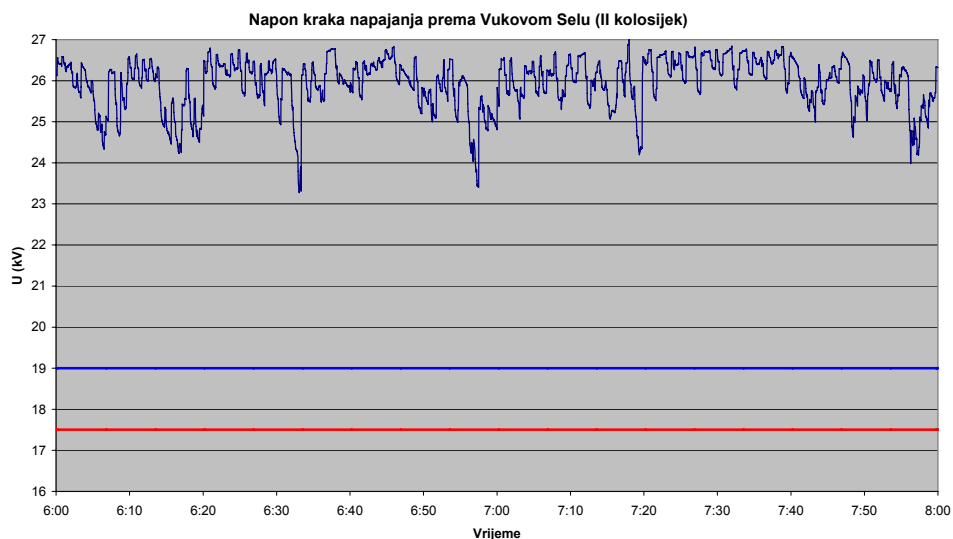
c) Krak napajanja prema Vukovom Selu

U proračunu je uzeto da je duljina kraka napajanja 11,78 km.

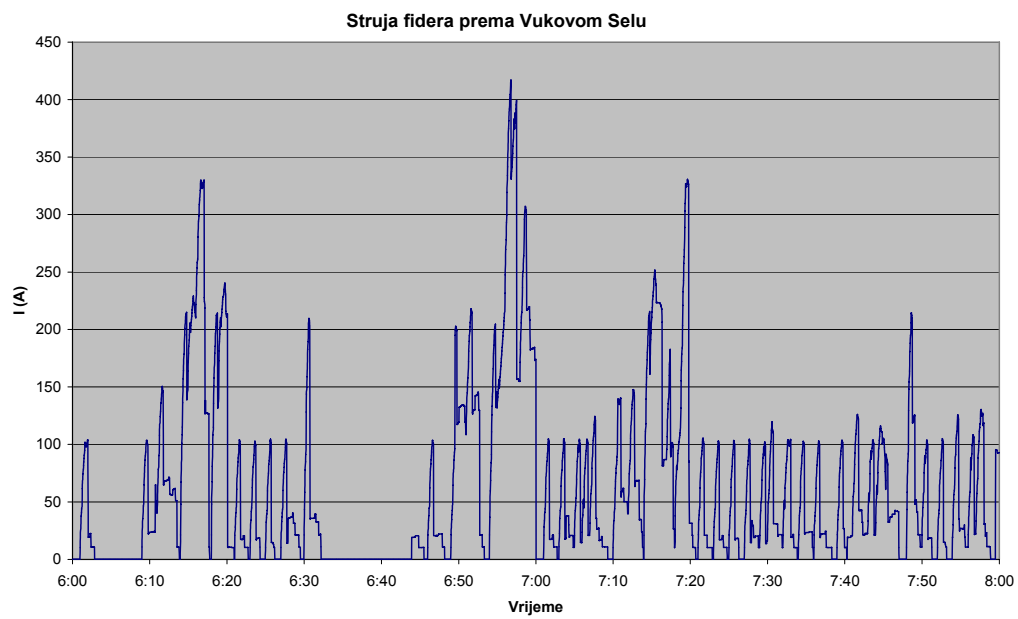


Slika 5.21. Krak napajanja EVP Zaprešić prema Vukovom Selu

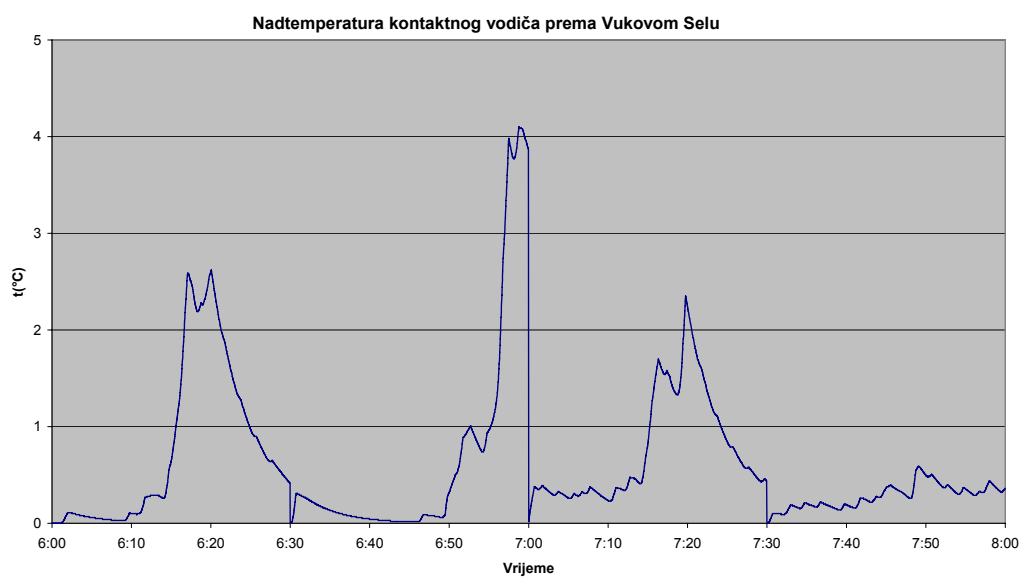
Grafički prikaz pada napona, struje fidera i nadtemperature kontaktnog voda za ovaj krak napajanja je se nalazi na sljedećim slikama.



Slika 5.22. Napon kraka napajanja prema Vukovom Selu



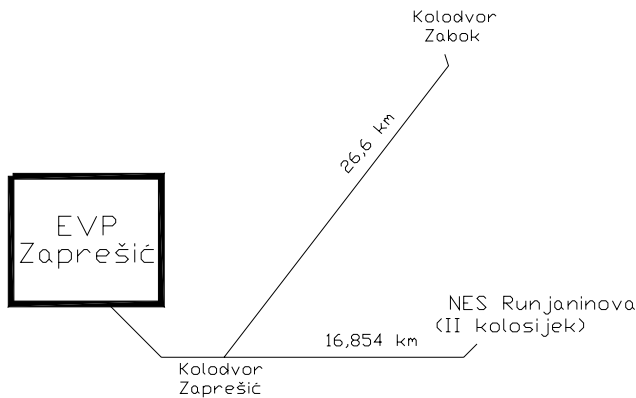
Slika 5.23. Struja fidera prema Vukovom Selu



Slika 5.24. Nadtemperatura kontaktnog voda prema Vukovom Selu

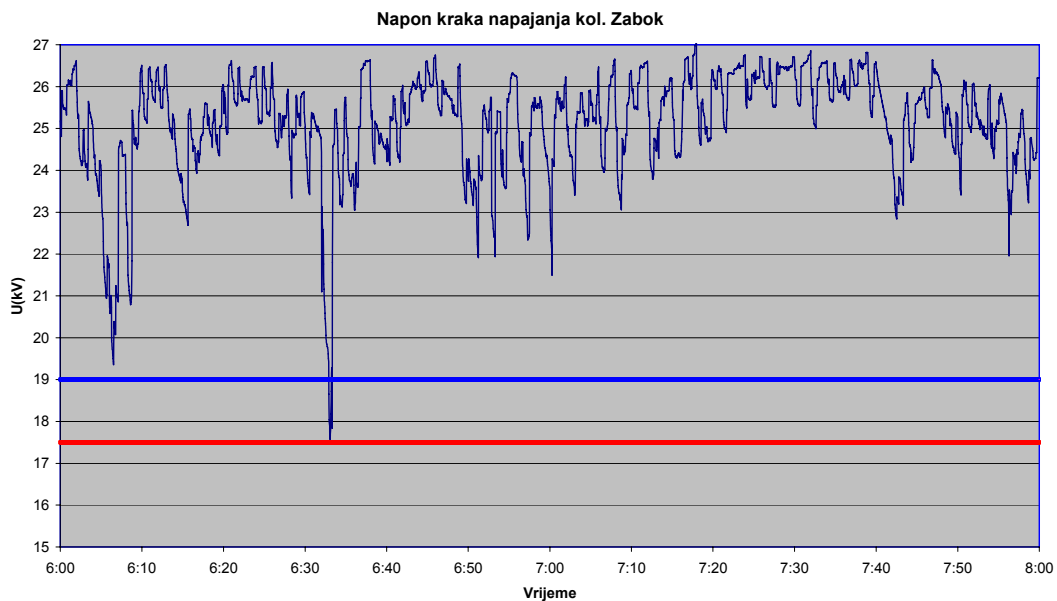
d) Krak napajanja prema kol. Zabok

U proračunu je uzeto da duljina kraka napajanja prema kol. Zabok iznosi 26,6 km. Shematski prikaz kraka napajanja predočen je na slici 5.25.

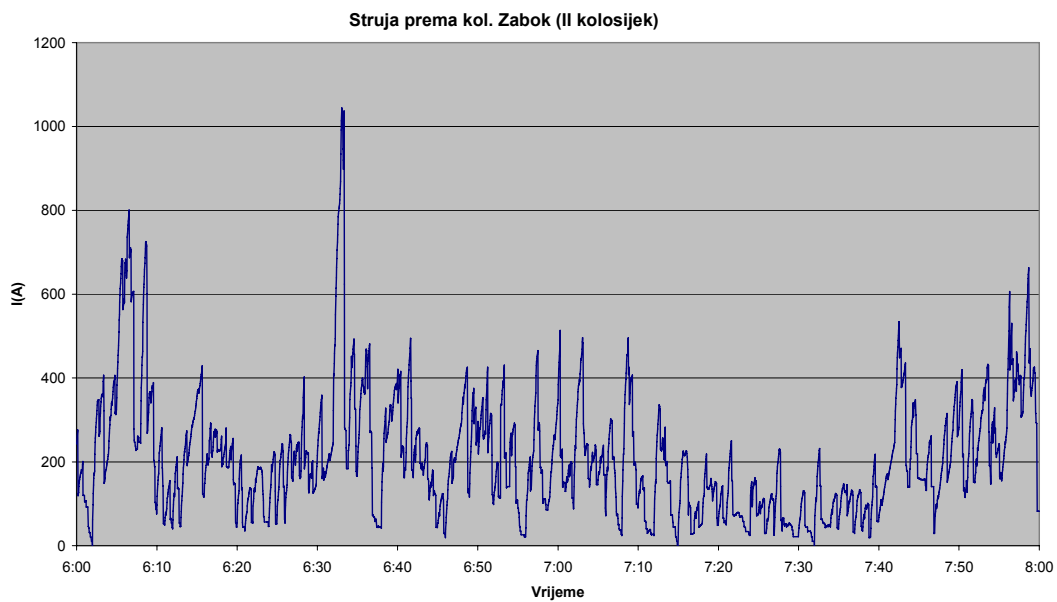


Slika 5.25. Krak napajanja EVP Zaprešić prema kol. Zabok

Grafički prikaz pada napona, struje fidera i nadtemperature kontaktnog voda za ovaj krak napajanja je na sljedećim slikama.



Slika 5.26. Napon kraka napajanja prema kol. Zabok



Slika 5.27. Struja fidera prema kol. Zabok



Slika 5.28. Nadtemperatura kontaktnog voda prema kol. Zabok

Iz slike 5.27. se vidi da maksimalno strujno opterećenje kontaktnog vodiča napojnog kraka prema kol. Zabok, iznosi oko 1000A, dok je strujni maksimum za napojni krak prema Vukovom Selu 400 A (slika 5.23).

Na osnovu rezultata dobivenih elektrovnim proračunom samo se u nekim slučajevima dešavaju kratkotrajni propadi napona ispod 19 kV, ali nikada ispod 17,5 kV, pa se može zaključiti kako su električne prilike zadovoljavajuće.

5.4. Rezime i analiza rezultata

Nakon provedenog proračuna daje se slijedno pregled dobivenih rezultata.

EVP ZAPREŠIĆ - za redovno napajanje električne vuče na zadanim prugama:

- Maksimalna prividna snaga : $S = 36 \text{ MVA}$
- Maksimalna prividna srednja 15 minutna snaga: $S = 17,5 \text{ MVA}$
- Maksimalna djelatna snaga : $P = 27 \text{ MW}$
- Maksimalna jalova snaga : $Q = 21 \text{ Mvar}$
- Minimalni faktor snage : $\lambda = 0,32$
- Potrošnja električne energije od 6,00 do 8,00 h : $E = 23 \text{ MWh}$

Na osnovu rezultata dobivenih elektrovučnim proračunom, što se tiče naponskih prilika, samo se u nekim slučajevima dešavaju kratkotrajni propadi napona ispod 19 kV, ali nikada ispod 17,5 kV, pa se može zaključiti kako su naponske prilike zadovoljavajuće.

Svi proračuni, pri redovnom napajanju EVP-a Zaprešić su provedeni za izvedbu kontaktne mreže s jednim kontaktnim vodičom presjeka 100 mm^2 i nosivog užeta BZ II 70 mm^2 , pa se može zaključiti da navedeni presjeci zadovoljavaju na svim dionicama magistralne pruge Zagreb G.K. – Dobova i sporednim prugama s obzirom na strujna opterećenja i dozvoljene padove napona.

6. ZAKLJUČAK

Ovim magistarskim radom se, u prvom redu, nastojalo poboljšati izvedbu proračuna el. prilika u izmjeničnom elektrovučnom sustavu 1 x 25 kV. Korišteni su podaci iz studije "Elektroenergetski proračun napajanja za prvu fazu nastavka izmjene sustava električne vuče na pruzi Zagreb – Rijeka i na priključnim prugama" i elaborata "Prometno-tehnološki projekt, Željeznička pruga II reda Podsused Tv. - Samobor – Bregana".

Elektroenergetskim proračunom se mogu dobiti slijedeći rezultati u izmjeničnom elektrovučnom sustavu 1 x 25 kV:

- maksimalne trenutne i 15-minutne snage EVP-a;
- struje u izlaznim linijama EVP-a;
- faktor snage;
- naponi na krajevima napajanog kraka;
- zagrijavanje vodiča kontaktne mreže.

Elektrovučna vozila predstavljaju potrošače, pa o njihovoj snazi i položaju zavise električne prilike u kontaktnoj mreži od kojih je za proračun najinteresantnija snaga kojom je opterećena EVP, struje u fiderima, te padovi napona u kontaktnoj mreži.

Iz opterećenja transformatora u EVP-ima prividnom srednjom 15-minutnom snagom, kao i maksimalnom prividnom snagom može se odrediti dovoljna instalirana snaga pojedinog EVP-a za odvijanje predviđenog prometa.

Proračuni za redovno napajanje EVP Zaprešić su provedeni uz pretpostavku da postoji kompenzacija jalove snage kondenzatorskim baterijama u kontaktnoj mreži ($Q = 1,853$ Mvar), što popravljiva faktor snage EVP Zaprešić. Djelovanje kompenzacije, međutim ne utječe na prevelike padove napona.

Iz izračunate potrošnje električne energije može se korigirati planirani vozni red kako bi se ostvarila i ušteda u maksimalnoj angažiranoj djelatnoj snazi. Također se može odrediti i potrebna snaga kompenzacije kako bi se smanjila potrošnja jalove snage.

U radu je također dan i postupak proračuna impedancija kontaktne mreže i usporedba rezultata dobivenih pomoću programa Matlab postepenim postupkom kao i matrično (blok-transformacijom).

Doprinos ovog magistarskog rada se prvenstveno temelji na poboljšanoj izvedbi potprograma za simulaciju električne vuče (Borland C++) i objedinjavanju s potprogramom za formiranje električne mreže (PEPES) i potprogramom za proračun električnih prilika u kontaktnoj mreži (GAUS).

7. SAŽETAK I KLJUČNE RIJEČI

Ključne riječi:

Simulator kretanja vlaka, vozni vod, povratni vod, proračun impedancije kontaktne mreže, formiranje električne mreže, elektrovučni proračun u mreži.

SAŽETAK:

Cilj rada je objedinjavanje i poboljšanje dosadašnjih potprograma za simulaciju električne vuče.

Najprije je opisan simulator kretanja vlaka (fizikalna slika kretanja), za kojeg su razvijeni potprogrami na Fortranu i Borlandu C++. Dana je usporedba rezultata navedena dva potprograma, pomoću grafičkog prikaza.

Zatim se prešlo na proračun impedancija kontaktne mreže koristeći Matlab 6.5 i dana je usporedba rezultata dobivenih postepenim postupkom i blok transformacijom matrice impedancija.

U četvrtom poglavlju je opisan algoritam za formiranje električne mreže pomoću potprograma PEPES, a u slijedećem poglavlju algoritam za proračun elektrovučnih prilika u kontaktnoj mreži (pomoću potprograma GAUS). Dan je i grafički prikaz rezultata dobivenih proračunima.

U radu su tri potprograma (simulator kretanja vlaka, formiranje električne mreže i elektrovučni proračun u mreži) povezana, a njihovo izvršavanje je pojednostavljeno. Pomoću njih je omogućena simulacija za razne tipove lokomotiva koje su prisutne u HŽ-u i za razne uvjete na dionicama.

Calculation of power supply for 25 kV, 50 Hz Electric Railway System

Keywords:

Simulator of train movement, catenary, traction, line, calculation impedance of railway line, forming of electric power networks, electric railway system analyzer.

SUMMARY:

The purpose of this work it is joined and improvement of previous subprograms for network analyzer.

It is first described simulator of train movement (physical illustration of movement), for which are developed subprograms in *Fortran* and *Borland C++*. It is also reperesented comparison of result, above mentioned two subprograms, by using graphical description.

After that it is described calculation impedance of railway line by using software tools *Matlab 6.5* and it is reperesented comparison of result assigned *gradual method* and block transformation matrix of impedance.

In the fourth chapter are described algorithm by forming of electric power networks using subprogram called PEPEs, and in the next chapter it is described algorithm for network analyzer (using subprogram called GAUS). It is also represented graphical description of results obtained with calculations.

In this work, it three subprograms (simulator of train movement, forming of electric power networks and electric railway system analyzer) are connected, and theirs implementation are improved.

Using this subprograms it is enabled simulation for various types locomotives which is present in HŽ, and for various condition on stock trail.

8. Literatura

- [1] V. Milardić: *"Diplomski zadatak br. 1702 - Proračun električnih prilika u elektrovučnom sustavu"*, Zagreb: uvez, 1995.
- [2] Z. A. Styezynski, S. Bacha, A. Bachry, I. Etxeberria: Improvement of EMC in Railway Power Networks: Harmonics and Quality of Power, 2002. 10th International Conference on Volume 2, 6-9 Oct. 2002 Page(s):754 - 759 vol.2
- [3] E. Pilo, L. Rouco, A. F. Cardador, A. H. Velilla: A simulation tool for the design of the electrical supply system of high-speed railway lines: Power Engineering Society Summer Meeting, 2000. IEEE: Volume 2, 16-20 July 2000 Page(s):1053 - 1058 vol. 2
- [4] I. Uglešić, I. Pavić, V. Milardić, M. Mandić : Elektroenergetski proračun napajanja za prvu fazu nastavka izmjene sustava električne vuče na pruzi Zagreb – Rijeka i na priključnim prugama, Studija, Zagreb, lipanj 2004.
- [5] M. Veža: *"Diplomski zadatak br. 2061 - Simulacija kretanja vlakova u elektrovučnom sustavu"*, Zagreb: uvez, 2004.
- [6] N. Omerhodžić: *"Diplomski zadatak br. 1657 - Simulacija električne vuče"*, Zagreb: uvez, 1993.
- [7] I. Švaljek, T. Kožulj; M. Bošnjak: *"Tehničko-eksploatacijski pokazatelji i značajke vučnih vozila Hrvatskih željeznica"*, Zagreb: Željeznička tiskara d.o.o., 2003.
- [8] D. Dinić: *"Električna vuča vozova I i II"*, Beograd: Građevinska knjiga, 1973.
- [9] A. Mariscotti, P. Pozzobon: Determination of the Electrical Parameters of Railway Traction Lines: IEEE TRANSACTION ON POWER DELIVERY, VOL.19.NO.4, OCTOMBER 2004
- [10] S. Božinović: Proračun impedanse vučnog strujnog kola na prugama elektrificiranim jednofaznom strujom industrijske učestanosti, Seith «Elektrotehnika» broj 4/1963
- [11] I. Uglešić, I. Pavić, V. Milardić, S. Hutter: Električni parametri i proračun raspodjele struja kratkog spoja duž DV 400 kV Tumbri – Heviz, Zagreb, srpanj 1999.
- [12] Jugoslovenske Željeznice: Opšti projekat kontaktne mreže monofaznog sistema 25 kV, 50 Hz, Beograd, jun 1967.
- [13] I. Uglešić, Č. Čavlina, A. Marušić, Z. Presečki, Z. Hebel, I. Kopecki, T. Tomiša, M. Bobetko, P. Pavlović: Izmjena sustava električne vuče na relaciji Moravice – Rijeka – Šapjane, Zagreb, 1987.
- [14] C.S. Chang, C.S. Chua, H.B. Quek, X.Y. Xu, S.L. Ho : Development of train movement simulator for analysis and optimisation of railway signalling systems: Developments in Mass Transit Systems, 1998. International Conference on (Conf.Publ. No. 453) Publication Date: 20-23 April 1998 On page(s): 243 – 248

[15] T. Kneschke, P. Mbika: Determination of traction power distribution system impedances and susceptances for AC railroad electrification systems : **Rail Conference, 2004. Proceedings of the 2004 ASME/IEEE Joint** Publication Date: 6-8 April 2004, On page(s): 27 - 34

[16] A. Mariscotti, P. Pozzobon: Measurement of the internal impedance of traction rails at 50 Hz : **IEEE Transactions on Instrumentation and Measurement** Volume: 49, No. 2, April 2000

[17] M.P. Toffolo, G.E. Dawson, A.R. Eastham: An improved train network simulator: Dept. of Electr. & Comput. Eng., Queen's Univ., Kingston, Ont., Canada; Electrical and Computer Engineering, 1994. Conference Proceedings. 1994 Canadian Conference on Publication Date: 25-28 Sept. 1994, On page(s): 121 - 124 vol.1

[18] I. Uglešić, I. Pavić, V. Milardić, M. Mandić, Z. Presečki: Elektroenergetski proračun napajanja električne vuče na pruzi Podsused Tvornica – Samobor – Bregana, Zagreb, ožujak 2006.

9. POPIS OZNAKA

Poglavlje II

EVP – elektrovučna potstanica
 F_p - stalni otpori
 F_v - povremeni otpori
 F_a - otpor ubrzanja
 F_k - otpori kotrljanja
 F_{par} - otpori parazitskih kretanja
 F_z - otpori zraka
 P_{maks} – najveća instalirana snaga

Poglavlje III

MTL - model prijenosnih vodova sa više vodiča
KM – kontaktna mreža
 R – djelatna komponenta ukupne impedancije
 X – induktivna komponenta ukupne impedancije
 Z_v – impedancija voznog voda, Ω/km
 Z_p – impedancija povratnog voda, Ω/km
 Z_{vp} – međusobna impedancija vozni vod – povratni vod, Ω/km
 k_p – konstanta rasprostiranja povratnog voda, km^{-1}
 l – duljina napojnog kraka od EVP-a, km
 Z_s – impedancija tračnice, Ω/km
 Y_s – vodljivost tračnice, S/km

10. Prilozi

Prilog 1: Ispis potprograma za simulaciju električne vuče u Fortranu i Borland C++-u;
Prilog 2: Ispis potprograma za formiranje električne mreže(PEPES) i proračun električnih prilika (GAUS)
Prilog 3: Ispis programa za proračun impedancije kontaktne mreže
Prilog 4: Ispis programa za proračun padova napona u kontaktnoj mreži
Prilog 5: Maksimalni grafikoni voznog reda dionice: Zagreb G.K. – Dobova od 6h do 8h.
Prilog 6: Maksimalni grafikoni voznog reda dionice Podsused Tv. - Bregana od 6h - do 8h;
Prilog 7: Maksimalni grafikoni voznog reda dionice: Zaprešić – Zabok od 6h do 8h;
Prilog 8: Maksimalni grafikoni voznog reda dionice: Savski Marof – Vukovo Selo od 6h do 8h;

11. ŽIVOTOPIS

Rođen sam 04. prosinca 1978. godine u Splitu. Osnovnu školu sam završio u Krivodolu. Nakon toga, u razdoblju 1993. – 1997. godine pohađao sam gimnaziju «dr. Mate Ujević» u Imotskom, prirodoslovno-matematički smjer gdje sam maturirao iste godine.

Fakultet elektrotehnike i računarstva upisao sam 1997. godine. Upisujem završetak studija na Zavodu za visoki napon i energetiku s naglaskom na znanstveno istraživačkom radu, tema « Elektromagnetska kompatibilnost u visokonaponskim postrojenjima», pod vodstvom prof. dr. sc. Ive Uglešića. Fakultet elektrotehnike i računarstva sam završio 2002. godine. s izvrsnim uspjehom. Dobitnik sam godišnje nagrade «Hrvoje Požar» , za 2002. godinu, za izvrstan uspjeh na studiju energetskog usmjerenja.

Od listopada 2002. godine zaposlen sam kao znanstveni novak na Fakultetu elektrotehnike i računarstva (Zavod za visoki napon i energetiku). Sudjelujem u nastavi na predmetima, Tehnika visokog napona, Osnove elektrotehnike II i Primjena računala.

Koautor sam nekoliko studija vezanih uz elektrifikaciju željezničkih pruga, ispitivanje otpora uzemljenja, prenaponsku zaštitu postrojenja i transformatorskih stanica, te nekoliko članaka iz područja objavljenih na domaćim i stranim konferencijama.

Govorim engleski i francuski jezik.

Prilog 1**Ispis potprograma za simulaciju električne vuče u FORTRANU:**

```
integer izbor1,I1
      integer*2 SYSTEM

      I1= system(' cls 'C)

*     call naslov
1     print 5
5     format(///,5X,' IZBOR :',///,20X,'1 - Upis podataka',///,
* 20X,'2 - Proracun',///,20X,'3 - Pregled rezultata',///,20X,
*4 - Zavrsetak rada')
      read*,izbor1
      if(izbor1.ne.1.and.izbor1.ne.2.and.izbor1.ne.3.and.izbor1.ne.4)
* then
      I1= system(' cls 'C)
      print*,' POGRESAN IZBOR - POKUSAJTE PONOVO '
      goto 1
      endif
      select case(izbor1)
      case(1)
        call upis
      case(2)
        call proracun
      case(3)
        call pregled
      case(4)
        I1= system(' cls 'C)
      end select

      if(izbor1.eq.1.or.izbor1.eq.2.or.izbor1.eq.3) then
      I1= system(' cls 'C)
      goto 1
      endif

      END

*-----
      subroutine upis
*-----

      integer brdion,izbor1,izbor2,rednibr,izbor3,statl

      real mlok,ldion,u,r,A,B,C,eps,vplan,vkon,Fvmax,k0,k1,k2,k3,k4,
      $mvl

7     print 10
10    format(///,'IZABERITE :',///,10X,'1 - upis podataka o profilu pr
      $uge',///,10X,'2 - upis podataka o vlakovima',///,10X
      $,'3 - upis podataka o voznom redu za svaki vlak',///,10X,'4 - upis
      $ podataka o lokomotivama',///,9X,' 5 - upis koeficijenata za prora
      *cun stalnih otpora vuče',///,10X,'6 - kraj upisa',///)
```

```
read*, izbor2
if(izbor2.ne.1.and.izbor2.ne.2.and.izbor2.ne.3.and.izbor2.ne.4.
&and.izbor2.ne.5.and.izbor2.ne.6) then
  print*, ' Pogresan izbor - pokusajte ponovo '
  go to 7
endif
*-----
*   Upis podataka o profile pruge ( ldion , uspon ,polumjer krivine)
*-----
if(izbor2.eq.1) then
  open(1,file='pruga.dat',access='direct',recl=40,form='formatted')
12  print*, ' Upisite redni broj dionice cije parametre upisujete '
    read*,rednibr
    if(rednibr.le.0.or.rednibr.gt.500) then
      print*, ' Pogresan redni broj - ponovite upis '
      go to 12
    endif
  print*
  print*, 'Upisite duljinu dionice (km);uspon i polumjer krivine (m) '
  read*,ldion,u,r
  write(1,15,rec=rednibr) ldion,u,r
15  format(5X,F8.4,5X,F5.1,5X,F7.1)
  print 41
  read*, izbor3
  if(izbor3.ne.0) goto 12
endif

*-----
*   Upis podataka o vlakovima ( sifra , masa,lokomotiva,eps,A,B,C )
*-----
if(izbor2.eq.2) then
  open(2,file='vlak.dat',access='direct',recl=50,form='formatted')
17  print*, ' Upisite redni broj vlaka ( 0 - 10 )'
    read*,rednibr
    if(rednibr.le.0.or.rednibr.gt.10) then
      print*, ' Pogresan redni broj - ponovite upis '
      go to 17
    endif

  print*
18  print*, 'Upisite sifru vlaka, masu vlaka (t),eps,A,B,C '
    read*,sifravl,mvl,eps,A,B,C
    write(2,25,rec=rednibr)sifravl,mvl,eps,A,B,C
25  format(A6,2X,F7.1,2X,F6.3,2X,F7.3,2X,F8.3,2X,F6.1)
  print 41
  read*, izbor3
  if(izbor3.ne.0) goto 18
endif

*-----
*   Upis podataka o voznom redu tj, Vplan (km/h) i zaustavljanje
*-----
if(izbor2.eq.3) then
  print*
```

```
open(3,file='voznired.dat',access='direct',recl=20,
$form='formatted')
35 print*,'Upisite redni broj dionice '
read*,rednibr
if(rednibr.le.0.or.rednibr.gt.500) then
print*,' Pogresan redni broj dionice - Ponovite upis '
goto 35
endif
print*
print*,'Upisite planiranu brzinu ( km/h ) te 1 za zaustavljanje,
$ ili 0 za prolaz '
read*,vplan,stati
write(3,40,rec=rednibr)vplan,stati
40 format(5X,F6.1,5X,I2)
print 41
41 format(///,' Pritisni :',//,10X,' 1 - za nastavak upisa',//,10X,
'$ 0 - za kraj upisa'//)
read*,izbor3
if(izbor3.ne.0) goto 35
endif
*-----
* Upis podataka o lokomotivama Fv(v),cosFi(v),Eta(v)
*-----
if(izbor2.eq.4)then
open(4,file='lokom.dat',access='direct',recl=90,form='formatted')
42 print*,' Upisi redni broj lokomotive '
read*,rednibr
print*
print*,'Upisite masu lokomotive (t), Fvmax (N), vkon (km/h)'
read*,mlok,Fvmax,vkon
print*
print*,'Upisite koeficijente izraza F(v) : K0, K1, K2, K3, K4 '
read*,k0,k1,k2,k3,k4
write(4,45,rec=rednibr)mlok,Fvmax,vkon,k0,k1,k2,k3,k4
45 format(F5.1,2X,F9.1,2X,F6.1,2X,F8.2,2X,F9.4,2X,2(F9.6,2X),
$E12.5)
print 41
read*,izbor3
if(izbor3.ne.0) goto 42
endif
*-----
* Upis koeficijenata za proracun stalnih otpora vuce
*-----
if(izbor2.eq.5) then
open(10,file='abc.dat',access='direct',recl=80,form='formatted')
70 print*,'Naznacenim redom upisite koeficijente A,B,C za lokomotivu'
read*,a1,b1,c1
print*
print*,'Naznacenim redom upisite koeficijente A,B,C za vlak'
read*,a2,b2,c2
write(10,500,rec=1)a1,b1,c1,a2,b2,c2
500 format(6e12.5)
print 41
```

```
read*, izbor3
  if(izbor3.ne.0) goto 70
if(izbor2.ne.6) then
  goto 7
else
  call cls
endif
  endif
  RETURN
  END

*-----
      SUBROUTINE PREGLED
      INTEGER*2 SYSTEM
      INTEGER I9,I18,izb

      I1= system( ' cls 'C)
11  print*
      print*, ' IZBOR : '
      print 107
107  format(////, ' 1 - Pregled izlazne datoteke Rezul pomocu BRIEF-a'
      *,//, ' 2 - Graficki prikaz rezultata',//)
      read*, izb
      if(izb.ne.1.and.izb.ne.2) then
        I1= system( ' cls 'C)
        print*, ' Pogresan izbor - pokusajte ponovo '
        go to 11
      elseif(izb.eq.1) then
        I18 = SYSTEM( ' edit rezul'C )
      else
        I9 = SYSTEM( 'pr_evp'C )
      endif
      RETURN
      END

*-----
      SUBROUTINE PRORACUN

      INTEGER*2 SYSTEM

*-----
*   Proracun kretanja vlaka duz zadanog profila pruge
*-----

      integer stati[allocatable](:),koci[allocatable](:),
+ zal[allocatable](:)
      integer i,k,brvlak,sifralok,kraj,tocke,uspor,vars,varv,varp,varq,
+ varpm,brform,I7,coast,brzal,brd,greska,nlok,vvl,smjer_k,sifratr,
+ invert,poc_sat,poc_min,dol_sat,dol_min

      real sz[allocatable](:),ldion[allocatable](:),
+ r[allocatable](:),u[allocatable](:),vplan[allocatable](:),
+ akoc[allocatable](:),szaus[allocatable](:),
```



```

+ tkoc[allocatable](:),dsz1[allocatable](:),dsz2[allocatable](:)

real s,dl1,Fv,F1,Fath,v,v0,cos_fi,aub,eps,eta,A,B,C,k0,k1,k2,k3,
+ k4,mlok,mvl,Wpr,Wst,dl2,t1,t2,t,dt,delta_t,Pm,P_pom,Pe,Qe,tstop,
+cos_fip,Fvmax,vkon,poc_s,U1,UIN,UIN_1,RL,R1,XL,X1,Q_pom

character*1 krajc,zapis,imeul*40,imeizlaz*12,sifrav1*5

      I7 = SYSTEM( 'del *.slk'C )
      I7 = SYSTEM( 'del rezul'C )

      uspor=0
      brvlak=1
      kraj=0
      zalet=0
      brzal=0
      i=1
      k=0
      tocke=0
      t=0
      s=0

*****
call cls
write(*,'(A\)\') ' Upisite ime ulazne datoteke -----> '
      read(*,'(A)') imeul
      open(1,file=imeul)
      krajc='n'
      brd=0
      call cls
do while (krajc.eq.'n')
      read(1,'(A)',err=200,end=250) zapis
      if (zapis.ne.'*' .and. zapis.ne.'e' .and. zapis.ne.'E') then
            backspace(1)
            read(1,*) sifrav1,sifralok,mvl,Eps,vvl,P_pom,cos_fip,nlok
            read(1,*) poc_s,smjer_k,poc_sat,poc_min,dol_sat,dol_min
            P_pom=P_pom * 1e3
            if((poc_s.lt.0.and.smjer_k.eq.1).or.(poc_s.gt.0.and.smjer_k.eq.-1)
+) then
                  poc_s=-poc_s
                  invert=-1
            else
                  invert=1
            endif
      endif
      if(zapis.eq.'e'.or.zapis.eq.'E') then
            krajc='d'
      endif
enddo
      krajc='n'
print*
do while (krajc.eq.'n')
      read(1,'(A)',err=200,end=250) zapis

```

```
        if (zapis.ne.'*' .and.zapis.ne.'e' .and.zapis.ne.'E') then
            brd=brd+1
        endif
        if(zapis.eq.'e') then
            krajc='d'
        endif
    enddo
    krajc='n'
    print*
***** Rezerviranje memorije za polja *****
    ALLOCATE(stati(brd+1),koci(brd),zal(brd),sz(brd),ldion(brd),
+ r(brd),u(brd),vplan(brd+1),akoc(brd),szaus(brd),tkoc(brd),
+ dsz1 (brd),dsz2(brd),STAT=greska)
*****
    rewind(1)
    zapis='*'
    do while(krajc.eq.'n')
        read(1,'(A)',err=200,end=250) zapis
        if(zapis.eq.'e'.or.zapis.eq.'E') then
            krajc='d'
        endif
    enddo
    ii=0
    krajc='n'
    do while (krajc.eq.'n')
        read(1,'(A)',err=200,end=250) zapis
        if (zapis.ne.'*' .and.zapis.ne.'e' .and.zapis.ne.'E') then
            ii=ii+1
            backspace(1)
            read(1,*) ldion(ii),u(ii),r(ii),vplan(ii),stati(ii),zal(ii)
        endif
        if(zapis.eq.'e'.or.zapis.eq.'E') then
            krajc='d'
        endif
    enddo
*   krajc='n'
c   print*
c   ii=0
c   do while (krajc.eq.'n')
c   read(1,'(A)',err=200,end=250) zapis
c   if (zapis.ne.'*' .and.zapis.ne.'e' .and.zapis.ne.'E') then
c   ii=ii+1
c   backspace(1)
c   read(1,*) vplan(ii),stati(ii),zal(ii)
c   endif
c   if(zapis.eq.'e'.or.zapis.eq.'E') then
c   krajc='d'
c   endif
*   enddo
    close(1)
    goto 300
```

200 print*, 'Greska kod citanja iz ulazne datoteke'

```
250  print*,' Dosegnut kraj ulazne datoteke'
      stop
300  continue
*****
      s=0
      eta=0.85
      call cls
*      print*,' Upisi vrijeme proracuna '
*      read*,poc_sat,poc_min,dol_sat,dol_min
6  print*,' Upisi pocetnu brzinu (km/h) koju vlak ima u trenutku t=0'
      read*,v0
      if(v0.lt.0) then
          call cls
          print*
          print*,'   POGRESAN PODATAK BRZINA MORA BITI > 0 '
          go to 6
      endif
      v=v0
      call cls
8  print*,'   Upisite vrijednost koraka proracuna u (s)'
      read*,delta_t
      if(delta_t.le.0) then
          call cls
          print*
          print*,'   POGRESAN PODATAK KORAK PRORACUNA MORA BITI > 0 '
          go to 8
      endif
      call cls
      print*
17  print*,' IZBOR FORMULA ZA PRORACUN STALNOG OTPORA VUCE '
      PRINT*
      PRINT*,' PRITISNITE : '
      PRINT*
      print*,' 1 - za HRVATSKE ZELJEZNICE '
      PRINT*
      PRINT*,' 2 - za AMTRAC '
      print*
      read*,brform
      call cls
      print*,' PRICEKAJTE VRSIM PRORACUN '
      if(brform.ne.1.and.brform.ne.2) then
          call cls
          print*,' POGRESAN IZBOR - POKUSAJTE PONOVO'
          go to 17
      endif
      write(imeizlaz,1111)sifrav1
1111 format('TR',A5,'.izl')
      open(4,file=imeizlaz)
*      open(4,file=sifrav1/' .IZL')
      open(7,file='s-t.slk',access='direct',recl=40,form='formatted')
      open(8,file='Pe.slk',access='direct',recl=40,form='formatted')
      open(9,file='Qe.slk',access='direct',recl=40,form='formatted')
```

```

*   write(14,15)sifravl,poc_sat,poc_min,dol_sat,dol_min
15  format(A5,5X,I2.2,3X,I2.2,5X,I2.2,3X,I2.2)
*-----
*   Citanje podataka o dionicama i racunanje puta sz od kojeg treba
*   poceti sa kocenjem
*-----

do j=1,brd
if(j+1.gt.brd) then
  if(stati(j).eq.1) then
    vplan(j+1)=0
    stati(j+1)=0
  else
    vplan(j+1)=vplan(j)
    stati(j+1)=0
  endif
endif

if(vplan(j).gt.vplan(j+1).or.stati(j).eq.1) then
  if(stati(j).eq.1) then
    szaus(j)=500.
    akoc(j)=-(vplan(j)/3.6)**2/(szaus(j)*2.)
    tkoc(j)=-vplan(j)/(3.6*akoc(j))
  else
    if(zal(j).ne.1) then
      akoc(j)=-0.5
      tkoc(j)=(vplan(j)-vplan(j+1))/(-3.6*akoc(j))
      szaus(j)=vplan(j)*tkoc(j)/3.6+akoc(j)*tkoc(j)**2/2.
    else
      szaus(j)=250.
    endif
  endif
  koci(j)=1
else
  koci(j)=0
  szaus(j)=0
endif
*   print* ,j,'koci=',koci(j),'skoc=',szaus(j)
*   pause
enddo
*****
**

do i=1,brd
  ldion(i)=ldion(i)*1000
  if(stati(i).eq.1.or.koci(i).eq.1) then
    k=k+1
    sz(k)=0
    do j=1,i
      sz(k)=sz(k)+ldion(j)
    enddo
    sz(k)=sz(k)-szaus(i)
  endif

```

```
        enddo
        if(t.eq.0) then
            dl1=ldion(i-1)
*       dsz1(k)=sz(k)
        endif
        i=1
        k=1
*-----

        write(4,15)sifrav1,poc_sat,poc_min,dol_sat,dol_min

        if(sifralok.eq.442) then
            mlok=80.
            Fvmax=226000.
            vkon=60.
            k0=821.14
            k1=-19.416
            k2=0.22488
            k3=-0.001278
            k4=0.28524e-5
        else
            stop ' Nemam podataka o vucnoj sili zadane lokomotive '
        endif

5    continue

        if(t.eq.0) then
            dl1=ldion(i)
            dsz1(k)=sz(k)
        endif
*-----
*   Ovaj dio programa racuna sile stalnog i promjenjivog otpora vuce
*-----
        call otpori(r(i),mvl,u(i),v,Wpr,Wst,brform,mlok,nlok,vvl,sifralok)
c-----
c   Racunanje raspolozive sile lokomotive i athezione sile
c-----
        call sila(Fvmax,vkon,v,k0,k1,k2,k3,k4,F1,nlok)
        Fath=nlok*(mlok*3.3/(1+0.01*v))*1000.
c-----
c   Dio programa u kojem se ispituje da li vlak vozi planiranom
c   brzinom te da li treba ubrzati ili usporiti (zaustaviti se)
c-----
        if(v.ne.vplan(i)) then
            if(uspor.ne.1) then
                if(v.lt.vplan(i)) then
                    if(zalet.ne.1) then
                        if(Fath.ge.F1) then
                            Fv=F1
                        else
                            Fv=Fath
                        endif
                    aub=(Fv-Wpr-Wst)/(mvl*1e3*(1+eps))
```

```

        tub=(vplan(i)-v)/(aub*3.6)
    else
*----- ako je predviđena voznja sa zaletom na i - toj dionici ---
        Fv=0
        aub=(Fv-Wpr-Wst)/(mvl*1e3*(1+eps))
        if(aub.lt.0) then
* ----- Tijekom zaleta vlak se usporava -----
            tub=(0.8*vplan(i)-v)/(aub*3.6)
        else
* ----- Tijekom zaleta vlak ubrzava -----
            if(v.lt.vplan(i)) then
                tub =(vplan(i)-v)/(aub*3.6)
                coast=1
            else
                aub=0
                tub=0
            endif
        endif
    endif
    else
*---- Brzina je veća od planirane pa treba voziti sa zaletom
        Fv=0
        zalet=1
        aub=(Fv-Wpr-Wst)/(mvl*1e3*(1+eps))
        tub=(vplan(i)-v)/(aub*3.6)
        Fv=Wst+Wpr+mvl*1e3*aub*(1+eps)
    endif
*----- ubrzavanje vlaka -----
        if(tub.gt.delta_t) then
            dt=delta_t
        else
            dt=tub
        endif

        if(t.eq.0) then
            CALL COSFI(cos_fi,v,sifralok)
            call snage(Fv,v,Pm,Pe,Qe,eta,cos_fi,P_pom,cos_fip)
*        call NAPONI(s,Pe,Qe,U1)
        call ispis(t,s,Pe,Qe,tocke,poc_s,invert)
        ds=v*dt/3.6+(aub*dt**2)/2.
        CALL jedan(s,ds,dl1,dsz1,aub,dt,t,v,brd,k)
        call otpori(r(i),mvl,u(i),v,Wpr,Wst,brform,mlok,nlok,vvl,sifralok)
        call sila(Fvmax,vkon,v,k0,k1,k2,k3,k4,F1,nlok)
        Fath=nlok*(mlok*3.3/(1+0.01*v))*1000.
        if(zalet.ne.1) then
            if(Fath.ge.F1) then
                Fv=F1
            else
                Fv=Fath
            endif
        endif
        aub=(Fv-Wpr-Wst)/(mvl*1e3*(1+eps))
        tub=(vplan(i)-v)/(aub*3.6)
    endif

```

```
CALL COSFI(cos_fi,v,sifralok)
call snage(Fv,v,Pm,Pe,Qe,eta,cos_fi,P_pom,cos_fip)
* call NAPONI(s,Pe,Qe,U1)
call ispis(t,s,Pe,Qe,tocke,poc_s,invert)
else
  ds=v*dt/3.6+(aub*dt**2)/2.
  dl2=dl1-ds
  dsz2(k)=dsz1(k)-ds
  if(dsz2(k).gt.0) then
*---- nije potrebno zapoceti sa kocenjem u ovom intervalu
    if(dl2.gt.0) then
*--- vlak ne prelazi na novu dionicu tijekom intervala ubrzanja
      if(tub.lt.delta_t) then
        if(zalet.eq.1) then
          brzal=brzal+1
          zalet=0
        endif
      CALL jedan(s,ds,dl1,dsz1,aub,dt,t,v,brd,k)
      aub=0
    call otpori(r(i),mvl,u(i),v,Wpr,Wst,brform,mlok,nlok,vvl,sifralok)
    call sila(Fvmax,vkon,v,k0,k1,k2,k3,k4,F1,nlok)
    Fv=Wst+Wpr+aub*mvl*(1+eps)*1e3
    CALL COSFI(cos_fi,v,sifralok)
    call snage(Fv,v,Pm,Pe,Qe,eta,cos_fi,P_pom,cos_fip)
* call NAPONI(s,Pe,Qe,U1)
* call ispis(t,s,Pe,Qe,tocke,poc_s,invert)
* write(4,321)
321 format(10X,' Zavrsetak ubrzanja ')
    t2=delta_t-tub
    ds=v*t2/3.6
    if(ds.lt.dl1) then
      dt=t2
    call otpori(r(i),mvl,u(i),v,Wpr,Wst,brform,mlok,nlok,vvl,sifralok)
    CALL jedan(s,ds,dl1,dsz1,aub,dt,t,v,brd,k)
    if(coast.ne.1) then
      if(zal(i).eq.1) then
        zalet=1
        Fv=0
        aub=(Fv-Wpr-Wst)/(mvl*1e3*(1+eps))
        if(brzal.ge.0) then
          aub=0
          Fv=Wpr+Wst
        endif
      else
        Fv=Wst+Wpr+mvl*aub*(1+eps)*1e3
      endif
    endif
    if(brzal.gt.0) then
      zalet=0
    endif
  endif
  CALL COSFI(cos_fi,v,sifralok)
  call snage(Fv,v,Pm,Pe,Qe,eta,cos_fi,P_pom,cos_fip)
* call NAPONI(s,Pe,Qe,U1)
```

```
call ispis(t,s,Pe,Qe,tocke,poc_s,invert)
  else
    dt=dl1*3.6/v
    ds=v*dt/3.6+(aub*dt**2)/2.
    CALL jedan(s,ds,dl1,dsz1,aub,dt,t,v,brd,k)
call otpori(r(i),mvl,u(i),v,Wpr,Wst,brform,mlok,nlok,vvl,sifralok)
  Fv=Wst+Wpr+mvl*aub*(1+eps)*1e3
  CALL COSFI(cos_fi,v,sifralok)
  call snage(Fv,v,Pm,Pe,Qe,eta,cos_fi,P_pom,cos_fip)
*
  call NAPONI(s,Pe,Qe,U1)
call ispis(t,s,Pe,Qe,tocke,poc_s,invert)
  i=i+1
  write(4,333)i
  if(i.gt.brd) then
!
  call crtaj(varv,vars,varp,varq,varpm,tocke)
    kraj=1
  else
    dl1=ldion(i)
    dt=t2-dt
    ds=v*dt/3.6+(aub*dt**2)/2.
    CALL jedan(s,ds,dl1,dsz1,aub,dt,t,v,brd,k)
call otpori(r(i),mvl,u(i),v,Wpr,Wst,brform,mlok,nlok,vvl,sifralok)
  if(coast.ne.1) then
  if(zal(i).eq.1) then
    zalet=1
    Fv=0
    aub=(Fv-Wpr-Wst)/(mvl*1e3*(1+eps))
  else
    Fv=Wst+Wpr+mvl*aub*(1+eps)*1e3
  endif
  endif
  CALL COSFI(cos_fi,v,sifralok)
  call snage(Fv,v,Pm,Pe,Qe,eta,cos_fi,P_pom,cos_fip)
*
  call NAPONI(s,Pe,Qe,U1)
call ispis(t,s,Pe,Qe,tocke,poc_s,invert)
  endif
  endif
  else
    dt=delta_t
    ds=v*dt/3.6+(aub*dt**2)/2.
    CALL jedan(s,ds,dl1,dsz1,aub,dt,t,v,brd,k)
call otpori(r(i),mvl,u(i),v,Wpr,Wst,brform,mlok,nlok,vvl,sifralok)
  call sila(Fvmax,vkon,v,k0,k1,k2,k3,k4,F1,nlok)
  Fath=nlok*(mlok*3.3/(1+0.01*v))*1000.
  if(zalet.ne.1) then
  if(Fath.ge.F1) then
    Fv=F1
  else
    Fv=Fath
  endif
  aub=(Fv-Wpr-Wst)/(mvl*1e3*(1+eps))
  tub=(vplan(i)-v)/(aub*3.6)
  endif
```



```
        CALL COSFI(cos_fi,v,sifralok)
        call snage(Fv,v,Pm,Pe,Qe,eta,cos_fi,P_pom,cos_fip)
*       call NAPONI(s,Pe,Qe,U1)
    call ispisi(t,s,Pe,Qe,tocke,poc_s,invert)
    endif
    else
*--- prijelaz na novu dionicu tijekom intervala-----
        ds=d11
        CALL UPITI(aub,ds,v,t1,t2,dt)
        dt = t1
        CALL jedan(s,ds,d11,dsz1,aub,dt,t,v,brd,k)
        i=i+1
*   print*,' ds= ',ds,' dt=',dt,'i=',i,' aub=',aub
*   pause
        write(5,333)i
        if(i.gt.brd) then
!       call crtaj(varv,vars,varp,varq,varpm,tocke)
        kraj=1
        else
            if(zal(i).eq.0) then
                zalet=0
            endif
        call otpori(r(i),mvl,u(i),v,Wpr,Wst,brform,mlok,nlok,vvl,sifralok)
        call sila(Fvmax,vkon,v,k0,k1,k2,k3,k4,F1,nlok)
        Fath=nlok*(mlok*3.3/(1+0.01*v))*1000.
        if(zalet.ne.1) then
            if(Fath.ge.F1) then
                Fv=F1
            else
                Fv=Fath
            endif
        else
            Fv=0
        endif
        aub=(Fv-Wpr-Wst)/(mvl*1e3*(1+eps))
        tub=(vplan(i)-v)/(aub*3.6)
*   print*,' Wpr = ',Wpr,' Wst= ',Wst,' Fv= ',Fv
*   print*,' s=',s,' v=',v,' aub=',aub,' tub=',tub
*   pause
        CALL COSFI(cos_fi,v,sifralok)
        call snage(Fv,v,Pm,Pe,Qe,eta,cos_fi,P_pom,cos_fip)
*       call NAPONI(s,Pe,Qe,U1)
    call ispisi(t,s,Pe,Qe,tocke,poc_s,invert)
        d11=ldion(i)
        dt = t2
        ds=v*dt/3.6+aub*dt**2/2.
        CALL jedan(s,ds,d11,dsz1,aub,dt,t,v,brd,k)
        CALL COSFI(cos_fi,v,sifralok)
    CALL otpori(r(i),mvl,u(i),v,Wpr,Wst,brform,mlok,nlok,vvl,sifralok)
        call sila(Fvmax,vkon,v,k0,k1,k2,k3,k4,F1,nlok)
        if(zalet.ne.1) then
            if(Fath.ge.F1) then
                Fv=F1
```

```
        else
            Fv=Fath
        endif
        else
            Fv=0
        endif
        aub=(Fv-Wpr-Wst)/(mvl*1e3*(1+eps))
        tub=(vplan(i)-v)/(aub*3.6)
*   print*, ' Wpr = ',Wpr, ' Wst= ',Wst, 'Fv= ',Fv
*   print*, ' s=',s, ' v=',v, ' aub=',aub, ' tub=',tub
*   pause
        endif
        call snage(Fv,v,Pm,Pe,Qe,eta,cos_fi,P_pom,cos_fip)
*       call NAPONI(s,Pe,Qe,U1)
        call ispis(t,s,Pe,Qe,tocke,poc_s,invert)
    endif
    else
*-----treba zapoceti s kocnjem unutar intervala dt-----
        uspor=1
        if(coast.eq.1) then
            coast=0
        endif
        if(dl2.gt.0) then
*----- ne prelazi na novu dionicu u intervalu dt -----
            ds=dsz1(k)
*       print*, 't=',t, ' v=',v, 's=',s, ' ds=',ds, ' koci=',koci(i)
*       print*, 'aub=',aub, ' akoc=',akoc(i), ' tkoc=',tkoc(i)
*       pause
            CALL UPITI(aub,ds,v,t1,t2,dt)
            dt = t1
            CALL jedan(s,ds,dl1,dsz1,aub,dt,t,v,brd,k)
            call uspori(akoc(i),tkoc(i),stati(i),v,vplan(i),szaus(i),
*           koci(i),brd,i)
            aub=akoc(i)
*       print*, ' aub=',aub, ' tkoc=',tkoc(i)
*       pause
            if(stati(i).eq.1) then
                aub=-((v/3.6)**2/(2*dl1))
                tkoc(i)=-v/(3.6*aub)
            endif
            Fv=Wst+Wpr+aub*mvl*1e3*(1+eps)
            CALL COSFI(cos_fi,v,sifralok)
            call snage(Fv,v,Pm,Pe,Qe,eta,cos_fi,P_pom,cos_fip)
*       call NAPONI(s,Pe,Qe,U1)
                call ispis(t,s,Pe,Qe,tocke,poc_s,invert)
            dt=t2
            ds=v*dt/3.6+aub*dt**2/2.
            CALL jedan(s,ds,dl1,dsz1,aub,dt,t,v,brd,k)
            CALL otpori(r(i),mvl,u(i),v,Wpr,Wst,brform,mlok,nlok,vvl,sifralok)
            Fv=Wst+Wpr+mvl*aub*(1+eps)*1e3
            CALL COSFI(cos_fi,v,sifralok)
            call snage(Fv,v,Pm,Pe,Qe,eta,cos_fi,P_pom,cos_fip)
*       call NAPONI(s,Pe,Qe,U1)
```

```
        call ispis(t,s,Pe,Qe,tocke,poc_s,invert)
        tkoc(i)=tkoc(i)-dt
c*----- prelazi na novu dionicu i zapocinje kocenje unutar dt-----
        else
            if(dl1.gt.dsz1(k)) then
*----- pocetak kocenja je na staroj dionici -----
                ds=dsz1(k)
                CALL UPITI(aub,ds,v,t1,t2,dt)
                dt = t1
                CALL jedan(s,ds,dl1,dsz1,aub,dt,t,v,brd,k)
                CALL COSFI(cos_fi,v,sifralok)
                call snage(Fv,v,Pm,Pe,Qe,eta,cos_fi,P_pom,cos_fip)
*
                call NAPONI(s,Pe,Qe,U1)
                call ispis(t,s,Pe,Qe,tocke,poc_s,invert)
                dt=t2
                ds=dl1-dsz1(k)
                call uspori(akoc(i),tkoc(i),stati(i),v,vplan(i),szaus(i),
*koci(i),brd,i)
                    t1=-v/(3.6*aub)-sqrt((v/(3.6*aub)**2+2.*ds/aub))
                    t2=dt-t1
                    dt = t1
                    CALL jedan(s,ds,dl1,dsz1,aub,dt,t,v,brd,k)
                    tkoc(i)=tkoc(i)-dt
                CALL otpori(r(i),mvl,u(i),v,Wpr,Wst,brform,mlok,nlok,vvl,sifralok)
                    Fv=Wst+Wpr+mvl*aub*(1+eps)*1e3
                    CALL COSFI(cos_fi,v,sifralok)
                    call snage(Fv,v,Pm,Pe,Qe,eta,cos_fi,P_pom,cos_fip)
*
                    call NAPONI(s,Pe,Qe,U1)
                    call ispis(t,s,Pe,Qe,tocke,poc_s,invert)
                    i = i + 1
                    if(i.gt.brd) then
!
                        call crtaj(varv,vars,varp,varq,varpm,tocke)
                        kraj=1
                    else
*
                        write(4,333)i
                        dl1=ldion(i)
                CALL otpori(r(i),mvl,u(i),v,Wpr,Wst,brform,mlok,nlok,vvl,sifralok)
                    dt = t2
                    ds = v*dt/3.6 + aub*dt**2/2.
                    CALL jedan(s,ds,dl1,dsz1,aub,dt,t,v,brd,k)
                    tkoc(i)=tkoc(i)-dt
                    CALL COSFI(cos_fi,v,sifralok)
                    call snage(Fv,v,Pm,Pe,Qe,eta,cos_fi,P_pom,cos_fip)
*
                    call NAPONI(s,Pe,Qe,U1)
                    call ispis(t,s,Pe,Qe,tocke,poc_s,invert)
                    endif
*----- pocetak kocenja je na novoj dionici -----
                    else
                        ds=dl1
                        CALL UPITI(aub,ds,v,t1,t2,dt)
                        dt = t1
                        CALL jedan(s,ds,dl1,dsz1,aub,dt,t,v,brd,k)
                        CALL COSFI(cos_fi,v,sifralok)
```

```

      call snage(Fv,v,Pm,Pe,Qe,eta,cos_fi,P_pom,cos_fip)
*      call NAPONI(s,Pe,Qe,U1)
          call ispis(t,s,Pe,Qe,tocke,poc_s,invert)
      dt=t2
      ds=dsz1(k)-dl1
      i=i+1
      if(i.gt.brd) then
!      call crtaj(varv,vars,varp,varq,varpm,tocke)
      kraj=1
      else
          dl1=ldion(i)
      CALL otpori(r(i),mvl,u(i),v,Wpr,Wst,brform,mlok,nlok,vvl,sifralok)
          call sila(Fvmax,vkon,v,k0,k1,k2,k3,k4,F1,nlok)
          Fath=nlok*(mlok*3.3/(1+0.01*v))*1000.
          if(Fath.ge.F1) then
              Fv=F1
          else
              Fv=Fath
          endif
          aub=(Fv-Wpr-Wst)/(mvl*1e3*(1+eps))
          CALL UPITI(aub,ds,v,t1,t2,dt)
          dt=t1
          ds=v*dt/3.6+aub*dt**2/2.
          CALL jedan(s,ds,dl1,dsz1,aub,dt,t,v,brd,k)
          CALL COSFI(cos_fi,v,sifralok)
      CALL otpori(r(i),mvl,u(i),v,Wpr,Wst,brform,mlok,nlok,vvl,sifralok)
          Fv=Wst+Wpr+mvl*aub*(1+eps)*1e3
          call snage(Fv,v,Pm,Pe,Qe,eta,cos_fi,P_pom,cos_fip)
*          call NAPONI(s,Pe,Qe,U1)
              call ispis(t,s,Pe,Qe,tocke,poc_s,invert)
          call uspori(akoc(i),tkoc(i),stati(i),v,vplan(i),szaus(i),
*koci(i),brd,i)
          t1=-v/(3.6*aub)-sqrt((v/(3.6*aub)**2+2.*ds/aub))
          dt=dt-t1
          ds= v*dt/3.6+aub*dt**2/2.
          CALL jedan(s,ds,dl1,dsz1,aub,dt,t,v,brd,k)
          tkoc(i)=tkoc(i)-dt
          CALL COSFI(cos_fi,v,sifralok)
          call snage(Fv,v,Pm,Pe,Qe,eta,cos_fi,P_pom,cos_fip)
*          call NAPONI(s,Pe,Qe,U1)
              call ispis(t,s,Pe,Qe,tocke,poc_s,invert)
          endif
          endif
          endif
          endif
          endif
      else
*----- call kocenje vlaka   uspor = 1 -----
          coast=0
          if(tkoc(i).ge.delta_t) then
              dt = delta_t
              ds= v*dt/3.6+aub*dt**2/2.
              CALL jedan(s,ds,dl1,dsz1,aub,dt,t,v,brd,k)

```

```

CALL otpori(r(i),mvl,u(i),v,Wpr,Wst,brform,mlok,nlok,vvl,sifralok)
Fv=Wst+Wpr+aub*mvl*1e3*(1+eps)
CALL COSFI(cos_fi,v,sifralok)
call snage(Fv,v,Pm,Pe,Qe,eta,cos_fi,P_pom,cos_fip)
* call NAPONI(s,Pe,Qe,U1)
    callispis(t,s,Pe,Qe,tocke,poc_s,invert)
tkoc(i)=tkoc(i)-delta_t
else
c---- Vrijeme kocenja manje je od koraka proracuna -----
ds=dl1
if(stati(i).eq.1) then
dt=-v/(3.6*aub)
Fv=0
else
dt=(vplan(i+1)-v)/(3.6*aub)
endif
t2=delta_t-dt
i=i+1
if(i.gt.brd) then
CALL jedan(s,ds,dl1,dsz1,aub,dt,t,v,brd,k)
CALL COSFI(cos_fi,v,sifralok)
call snage(Fv,v,Pm,Pe,Qe,eta,cos_fi,P_pom,cos_fip)
* call NAPONI(s,Pe,Qe,U1)
aub=0
    call ispis(t,s,Pe,Qe,tocke,poc_s,invert)
! call crtaj(varv,vars,varp,varq,varpm,tocke)
kraj=1
else
CALL jedan(s,ds,dl1,dsz1,aub,dt,t,v,brd,k)
CALL otpori(r(i),mvl,u(i),v,Wpr,Wst,brform,mlok,nlok,vvl,sifralok)
Fv=Wst+Wpr
CALL COSFI(cos_fi,v,sifralok)
call snage(Fv,v,Pm,Pe,Qe,eta,cos_fi,P_pom,cos_fip)
* call NAPONI(s,Pe,Qe,U1)
* write(4,333)i
aub=0
    call ispis(t,s,Pe,Qe,tocke,poc_s,invert)
c----- prelazak na novu dionicu-----
dl1=ldion(i)
dt = t2
if(stati(i-1).ne.1) then
ds=v*dt/3.6+aub*dt**2/2.
CALL otpori(r(i),mvl,u(i),v,Wpr,Wst,brform,mlok,nlok,vvl,sifralok)
Fv=Wpr+Wst+mvl*aub*(1+eps)*1e3
CALL jedan(s,ds,dl1,dsz1,aub,dt,t,v,brd,k)
CALL COSFI(cos_fi,v,sifralok)
call snage(Fv,v,Pm,Pe,Qe,eta,cos_fi,P_pom,cos_fip)
* call NAPONI(s,Pe,Qe,U1)
aub=0
    callispis(t,s,Pe,Qe,tocke,poc_s,invert)
v=vplan(i)
else
ds=0

```

```

Fv=0
CALL jedan(s,ds,dl1,dsz1,aub,dt,t,v,brd,k)
CALL COSFI(cos_fi,v,sifralok)
call snage(Fv,v,Pm,Pe,Qe,eta,cos_fi,P_pom,cos_fip)
*
call NAPONI(s,Pe,Qe,U1)
    call ispis(t,s,Pe,Qe,tocke,poc_s,invert)
print*, ' t=',t,'s    Vrijeme stajanja u stanici (s) ?'
read*,tstop
*****
t1stop=t    !!! trenutak zaustavljanja vlaka
bkoraka=aint(tstop/delta_t)
print*, ' bkoraka = ',bkoraka
do i=1,bkoraka
    t=t+delta_t
    call snage(Fv,v,Pm,Pe,Qe,eta,cos_fi,P_pom,cos_fip)
*
    call NAPONI(s,Pe,Qe,U1)
    call ispis(t,s,Pe,Qe,tocke,poc_s,invert)
enddo
t=t1stop+tstop
*****
    call snage(Fv,v,Pm,Pe,Qe,eta,cos_fi,P_pom,cos_fip)
*
    call NAPONI(s,Pe,Qe,U1)
    call ispis(t,s,Pe,Qe,tocke,poc_s,invert)
endif
uspor=0
endif
endif
endif
*----- call voznja konstantnom brzinom -----
else
    aub=0
    Fv=Wst+Wpr
    dt = delta_t
    ds=v*dt/3.6
    dl2=dl1-ds
    dsz2(k)=dsz1(k)-ds
    if(dsz2(k).gt.0)then
*----- ne treba zapoceti s kocenjem unutar intervala dt
        if(dl2.gt.0) then
*----- ne prelazi na novu dionicu -----
            if(zal(i).eq.1.and.coast.ne.1) then
                Fv=0
                aub=(Fv-Wpr-Wst)/(mvl*1e3*(1+eps))
                tub=(vplan(i)-v)/(aub*3.6)
                ds=v*delta_t/3.6+aub*delta_t**2/2.
                CALL jedan(s,ds,dl1,dsz1,aub,dt,t,v,brd,k)
                CALL COSFI(cos_fi,v,sifralok)
                call snage(Fv,v,Pm,Pe,Qe,eta,cos_fi,P_pom,cos_fip)
*
                call NAPONI(s,Pe,Qe,U1)
            callispis(t,s,Pe,Qe,tocke,poc_s,invert)
                zalet=1
            else
                CALL jedan(s,ds,dl1,dsz1,aub,dt,t,v,brd,k)

```

```
        CALL COSFI(cos_fi,v,sifralok)
        call snage(Fv,v,Pm,Pe,Qe,eta,cos_fi,P_pom,cos_fip)
*          call NAPONI(s,Pe,Qe,U1)
    call ispis(t,s,Pe,Qe,tocke,poc_s,invert)
    endif
    else
*----- prelazak na novu dionicu -----
        ds=dl1
        t1=ds*3.6/v
        t2=dt-t1
        dt = t1
        CALL jedan(s,ds,dl1,dsz1,aub,dt,t,v,brd,k)
    CALL otpori(r(i),mvl,u(i),v,Wpr,Wst,brform,mlok,nlok,vvl,sifralok)
        Fv=Wst+Wpr+aub*mvl*1e3*(1+eps)
        CALL COSFI(cos_fi,v,sifralok)
        call snage(Fv,v,Pm,Pe,Qe,eta,cos_fi,P_pom,cos_fip)
        i=i+1
*          write(4,333)i
*          call NAPONI(s,Pe,Qe,U1)
    call ispis(t,s,Pe,Qe,tocke,poc_s,invert)
333    format(5X,' Prelazak na novu dionicu br. ',I2)
    if(i.gt.brd) then
!          call crtaj(varv,vars,varp,varq,varpm,tocke)
        kraj=1
    else
        dl1=ldion(i)
        dt = t2
        ds=v*dt/3.6+aub*dt**2/2.
        CALL jedan(s,ds,dl1,dsz1,aub,dt,t,v,brd,k)
    CALL otpori(r(i),mvl,u(i),v,Wpr,Wst,brform,mlok,nlok,vvl,sifralok)
    if(v.lt.vplan(i)) then
        call sila(Fvmax,vkon,v,k0,k1,k2,k3,k4,F1,nlok)
        Fath=nlok*(mlok*3.3/(1+0.01*v))*1000.
        if(zalet.ne.1) then
            if(Fath.ge.F1) then
                Fv=F1
            else
                Fv=Fath
            endif
            aub=(Fv-Wpr-Wst)/(mvl*1e3*(1+eps))
            tub=(vplan(i)-v)/(aub*3.6)
        endif
    else
        Fv=Wst+Wpr+aub*mvl*(1+eps)*1e3
    endif
        CALL COSFI(cos_fi,v,sifralok)
        call snage(Fv,v,Pm,Pe,Qe,eta,cos_fi,P_pom,cos_fip)
*          call NAPONI(s,Pe,Qe,U1)
    callispis(t,s,Pe,Qe,tocke,poc_s,invert)
    endif
    endif
    else
*----- treba zapoceti sa kocenjem unutar dt -----
```

```

    uspor=1
    if(dl2.gt.0) then
*----- ne prelazi na novu dionicu u intervalu dt -----
        ds=dsz1(k)
        t1=ds*3.6/v
        t2=dt-t1
        dt=t1
        CALL jedan(s,ds,dl1,dsz1,aub,dt,t,v,brd,k)
        if(v.ne.vplan(i)) then
            if(stati(i).eq.1) then
                akoc(i)=-v/(3.6**2)/(2.*szaus(i))
                tkoc(i)=-v/(3.6*akoc(i))
            elseif(koci(i).eq.1) then
                akoc(i)=(v-vplan(i))*(vplan(i)+v)/(2.*szaus(i))
                tkoc(i)=(vplan(i)-v)/(3.6*akoc(i))
            endif
        endif
        aub=akoc(i)
        Fv=Wst+Wpr+aub*mv1*(1+eps)*1e3
        CALL COSFI(cos_fi,v,sifralok)
        call snage(Fv,v,Pm,Pe,Qe,eta,cos_fi,P_pom,cos_fip)
*
        call NAPONI(s,Pe,Qe,U1)
        call ispis(t,s,Pe,Qe,tocke,poc_s,invert)
        k=k+1
        dsz1(k)=sz(k)-s
        dt=t2
        ds=v*dt/3.6+aub*dt**2/2.
        Fv=Wst+Wpr+aub*mv1*1e3*(1+eps)
        CALL jedan(s,ds,dl1,dsz1,aub,dt,t,v,brd,k)
        CALL COSFI(cos_fi,v,sifralok)
        call snage(Fv,v,Pm,Pe,Qe,eta,cos_fi,P_pom,cos_fip)
*
        call NAPONI(s,Pe,Qe,U1)
        call ispis(t,s,Pe,Qe,tocke,poc_s,invert)
        tkoc(i)=tkoc(i)-dt
c*----- prelazi na novu dionicu i zapocinje kocenje unutar dt-----
        else
            if(dl1.gt.dsz1(k)) then
*----- pocetak kocenja je na staroj dionici -----
                ds=dsz1(k)
                t1 = ds*3.6/v
                t2=dt - t1
                dt = t1
                CALL jedan(s,ds,dl1,dsz1,aub,dt,t,v,brd,k)
                call uspori(akoc(i),tkoc(i),stati(i),v,vplan(i),szaus(i),
*
                *koci(i),brd,i)
                aub=akoc(i)
                Fv=Wst+Wpr+aub*mv1*1e3*(1+eps)
                CALL COSFI(cos_fi,v,sifralok)
                call snage(Fv,v,Pm,Pe,Qe,eta,cos_fi,P_pom,cos_fip)
*
                call NAPONI(s,Pe,Qe,U1)
                call ispis(t,s,Pe,Qe,tocke,poc_s,invert)
                dt=t2
                ds=dl1-dsz1(k)

```



```

t1=-v/(3.6*aub)-sqrt((v/(3.6*aub))**2+ds*2./(aub))
t2=dt-t1
dt = t1
CALL jedan(s,ds,dl1,dsz1,aub,dt,t,v,brd,k)
tkoc(i)=tkoc(i)-dt
Fv=Wst+Wpr+aub*mvl*1e3*(1+eps)
CALL COSFI(cos_fi,v,sifralok)
call snage(Fv,v,Pm,Pe,Qe,eta,cos_fi,P_pom,cos_fip)
*      call NAPONI(s,Pe,Qe,U1)
      call ispis(t,s,Pe,Qe,tocke,poc_s,invert)
if((tkoc(i).le.1e-2).or.(abs(vplan(i+1)-v).lt.0.11)) then
k=k+1
tkoc(i)=0
v=vplan(i+1)
aub=0
uspor=0
endif
dsz1(k)=sz(k)-s
i = i + 1
if(i.gt.brd) then
! call crtaj(varv,vars,varp,varq,varpm,tocke)
kraj=1
else
*      write(4,333)i
      dl1=ldion(i)
CALL otpori(r(i),mvl,u(i),v,Wpr,Wst,brform,mlok,nlok,vvl,sifralok)
dt = t2
ds = v*dt/3.6+aub*dt**2/2.
CALL jedan(s,ds,dl1,dsz1,aub,dt,t,v,brd,k)
Fv=Wst+Wpr+aub*mvl*1e3*(1+eps)
CALL COSFI(cos_fi,v,sifralok)
call snage(Fv,v,Pm,Pe,Qe,eta,cos_fi,P_pom,cos_fip)
if(aub.lt.0) then
endif
*      call NAPONI(s,Pe,Qe,U1)
      call ispis(t,s,Pe,Qe,tocke,poc_s,invert)
endif
*----- pocetak kocenja je na novoj dionici -----
else
ds=dl1
t1 = ds*3.6/v
t2=dt-t1
dt = t1
CALL jedan(s,ds,dl1,dsz1,aub,dt,t,v,brd,k)
CALL COSFI(cos_fi,v,sifralok)
call snage(Fv,v,Pm,Pe,Qe,eta,cos_fi,P_pom,cos_fip)
*      call NAPONI(s,Pe,Qe,U1)
      call ispis(t,s,Pe,Qe,tocke,poc_s,invert)
dt=t2
ds=dsz1(k)-dl1
i=i+1
*      write(5,333)i
if(i.gt.brd) then

```

```
!      call crtaj(varv,vars,varp,varq,varpm,tocke)
      kraj=1
      else
        dl1=ldion(i)
      CALL otpori(r(i),mvl,u(i),v,Wpr,Wst,brform,mlok,nlok,vvl,sifralok)
      CALL jedan(s,ds,dl1,dsz1,aub,dt,t,v,brd,k)
      CALL COSFI(cos_fi,v,sifralok)
      call snage(Fv,v,Pm,Pe,Qe,eta,cos_fi,P_pom,cos_fip)
*      call NAPONI(s,Pe,Qe,U1)
        call ispis(t,s,Pe,Qe,tocke,poc_s,invert)
        call uspori(akoc(i),tkoc(i),stati(i),v,vplan(i),szaus(i),
*      koci(i),brd,i)
        t1=-v/(3.6*aub)-sqrt((v/(3.6*aub)**2+2.*ds/aub))
        dt=dt-t1
        ds= v*dt/3.6+aub*dt**2/2.
        CALL jedan(s,ds,dl1,dsz1,aub,dt,t,v,brd,k)
        tkoc(i)=tkoc(i)-dt
        Fv=Wst+Wpr+aub*mvl*1e3*(1+eps)
        CALL COSFI(cos_fi,v,sifralok)
        call snage(Fv,v,Pm,Pe,Qe,eta,cos_fi,P_pom,cos_fip)
*      call NAPONI(s,Pe,Qe,U1)
        call ispis(t,s,Pe,Qe,tocke,poc_s,invert)
      endif
      endif
      endif
      endif
      if(kraj.ne.1) then
        goto 5
      endif
      DEALLOCATE(stati,koci,zal,sz,ldion,r,u,vplan,akoc,szaus,tkoc,
+ dsz1,dsz2,STAT=greska)

      close(4)
*      close(6)
      close(7)
      close(8)

      close(10)
*      close(11)
*      close(12)
*      close(13)
      RETURN
      END
*-----
*      Potprogram za citanje podataka o novoj dionici
*-----
      SUBROUTINE citaj(tkoc,szaus,koci,vplan,stati,akoc,brd)

      integer brd,stati(brd+1),koci(brd)
      real vplan(brd+1),tkoc(brd),szaus(brd),akoc(brd)

      do j=1,brd
```

```
if(j+1.gt.brd) then
  if(stati(j).eq.1) then
    vplan(j+1)=0
    stati(j+1)=0
  else
    vplan(j+1)=vplan(j)
    stati(j+1)=0
  endif
endif

if(vplan(j).gt.vplan(j+1).or.stati(j).eq.1) then
  if(stati(j).eq.1) then
    szaus(j)=500.
    akoc(j)=-vplan(j)/3.6**2/(szaus(j)*2.)
    tkoc(j)=-vplan(j)/(3.6*akoc(j))
  else
    akoc(j)=-0.5
    tkoc(j)=(vplan(j)-vplan(j+1))/(-3.6*akoc(j))
    szaus(j)=vplan(j)*tkoc(j)/3.6+akoc(j)*tkoc(j)**2/2.
  endif
  koci(j)=1
else
  koci(j)=0
  szaus(j)=0
endif
enddo
RETURN
END

*-----
*   Potprogram za izracunavanje vucne sile ovisno o brzini
*-----
SUBROUTINE sila(Fvmax,vkon,v,a0,a1,a2,a3,a4,F1,nlok)
real a0,a1,a2,a3,a4
if(v.le.vkon) then
  F1=fvmax
elseif(v.le.120) then
  F1=(a0+a1*v+a2*(v**2)+a3*(v**3)+a4*(v**4))*1000.
else
  F1=0
endif
F1=nlok*F1
RETURN
END

*-----
*   Proracun otpora vuce
*-----
SUBROUTINE otpori(r,mvl,u,v,Wpr,Wst,brform,mlok,nlok,vvl,sifralok)

integer sifralok,brform,vvl
real u,r,v,Wpr,Wst,mvl,a1,c1,a2,c2,mlok

if(r.eq.0) then
  Wpr=(mvl*u)*10.
```

```

else
    Wpr=mvl*(u+800./r)*10.
endif
if(brform.eq.1) then
    if(sifralok.eq.442) then
        a1=240.
        c1=20.
    else
        a1=280.
        c1=20.
    endif
    select case (vvl)
    case (1)
        a2=310
        c2=20
    case(2)
        a2=280
        c2=25
    end select
    Wst=((v**2)/a1+c1)*mlok*nlok+((v**2)/a2+c2)*(mvl-mlok*nlok)
else
*   Wst=A*(v/3.6)**2+B*v/3.6+C
endif
RETURN
END
*-----
*   BRISANJE EKRANA
*-----
SUBROUTINE CLS
integer ii
integer*2 SYSTEM
ii=SYSTEM(' cls 'C)
RETURN
END
*-----
*   ISPIS REZULTATA U DATOTEKE
*-----
*   poc_s = a invert = b

SUBROUTINE ISPIS(t,s,Pe,Qe,tocke,a,b)

integer toc,tocke,b,toc2
real s,a,Q_pom

*   Q_pom=P_pom*tan(acos(cos_fip))
toc=tocke+4
toc2=tocke+2
if((a.ge.0.and.b.eq.-1).or.(a.le.0.and.b.eq.-1)) then
write(4,111)t,-a+s,Pe,Qe
else
write(4,111)t,a-s,Pe,Qe
endif
write(7,812,rec=toc)t,poc_s-s

```

```

        write(8,812,rec=toc)t,Pe*1e6
        write(9,812,rec=toc)t,Qe*1e6
        tocke=tocke+1
111  format(F8.2,5X,F8.1,5X,F7.3,5X,F7.3)
112  format(F8.2,3X,F8.1,3X,F7.3,3X,F7.3)
812  format(2e15.5)
      RETURN
      END
*-----
*   RACUNANJE DJELATNE I JALOVE SNAGE
*-----
      SUBROUTINE SNAGE(Fv,v,Pm,Pe,Qe,eta,cos_fi,P_pom,cos_fip)

      real Pekn,eta_k,Fek

      eta_k=0.7

      Pm = Fv*v/3.6e6

      if(Pm.lt.0) then
        if(v.le.64) then
          Fek=2048*v
          Pekn=Fek*v/3.6e6
        elseif(v.le.120) then
          Fek=(133.2377+1.39255*v-0.0297893*v**2+1.1727e-4*v**3)*1e3
          Pekn=Fek*v/3.6e6
        endif
        if(abs(Pm).gt.Pekn) then
          Pe=-Pekn*eta_k+p_pom/1e6
          Qe=Pe*tan(acos(cos_fip))
        else
          Pe=Pm*eta_k+p_pom/1e6
          Qe=Pe*tan(acos(cos_fip))
        endif
      else
        Pe=Pm/eta+p_pom/1e6
        Qe=Pe*tan(acos(cos_fi))
      endif
      if(Pm.le.0.) then
        Pe=P_pom/1e6
        Qe=Pe*tan(acos(cos_fip))
      endif
      RETURN
      END
*-----
*   RACUNANJE NAPONA U MREZI
*-----
      SUBROUTINE NAPONI(s,Pe,Qe,U1)

      UIN_1=27.5
      RL=0.17
      XL=0.45

```

```

        R1=RL*s/1000
          X1=XL*s/1000
          U1=25.0
10      UIN= U1+(Pe*R1+Qe*X1)/U1
        U_d=UIN_1-UIN
        if(U_d.gt.10e-3) then
          U1=U1+0.001
        goto 10
        endif
        RETURN
        END

*-----
* dio proračuna prilikom kretanja vlaka sa ubrzanjem  aub
*-----
      SUBROUTINE UPITI(aub,ds,v,t1,t2,dt)
        if(aub.gt.0) then
          t1=-v/(3.6*aub)+sqrt((v/(3.6*aub))**2+2.*ds/aub)
        else
          t1=-v/(3.6*aub)-sqrt((v/(3.6*aub))**2+2.*ds/aub)
        endif
        t2=dt-t1
      RETURN
      END

*-----
* dio proračuna prilikom kretanja vlaka sa ubrzanjem  aub
*-----
      SUBROUTINE jedan(s,ds,dl1,dsz1,aub,dt,t,v,brd,k)
        integer k,brd
        real dv,dsz1(brd)
        s=s+ds
        dl1=dl1-ds
        dsz1(k)=dsz1(k)-ds
        dv=aub*dt*3.6
        v=v+dv
        t=t+dt
      RETURN
      END

*-----
* opis podataka neophodnih za program PR_EVP
*-----

      SUBROUTINE crtaj(varv,vars,varp,varq,varpm,tocke)

        integer p,varp,varv,varpm,varq,vars,tocke

        do p=1,3
          if(p.eq.1) then
            VARS=1
            VARV=2
            VARP=3
            VARPM=3
            VARQ=4
          elseif(p.eq.2) then

```

```
        VARV=1
        VARS=1
        VARP=1
        VARPM=1
        VARQ=1
    else
        VARV=tocke
        VARP=tocke
        VARPM=tocke
        VARS=tocke
        VARQ=tocke
    endif
800    format(15)
!      write(6,800,rec=p)varv
      write(7,800,rec=p)vars
      write(8,800,rec=p)varp
      write(9,800,rec=p)varq
      write(11,800,rec=p)varpm
      write(12,800,rec=p)varv
    enddo
    return
end

*-----
*  Izracunavanje usporenja vlaka
*-----
subroutine uspori(akoc,tkoc,stati,v,vplan,szaus,koci,brd,i)

integer brd,stati(brd),koci(brd),i

real szaus(brd),akoc(brd),tkoc(brd),vplan(brd),v

*  print*,' Uspori',' koci=',koci(i),' stati=',stati(i)
*  print*,' v= ',v,' vplan=',vplan(i)
  if(v.ne.vplan(i)) then
    if(stati(i).eq.1) then
      akoc(i)=(-(v/3.6)**2)/(2.*szaus(i))
      tkoc(i)=-v/(3.6*akoc(i))
    elseif(koci(i).eq.1) then
      akoc(i)=(v-vplan(i))*(vplan(i)+v)/(2.*szaus(i))
      tkoc(i)=(vplan(i)-v)/(3.6*akoc(i))
    endif
  endif
  return
end

*-----
*  Izracunavanje faktora snage u ovisnosti o brzini
*-----
*  brzina je izrazena u km/h

SUBROUTINE COSFI(cos_fi,v,sifralok)

integer sifralok
if(sifralok.eq.442) then
```

```
if(v.le.5) then
  cos_fi=0.24
elseif(v.le.20) then
  cos_fi=(0.1*v+0.22)/3.
elseif(v.le.28) then
  cos_fi=0.018125*v+0.3775
elseif(v.le.32) then
  cos_fi=-0.00375*v+0.785
elseif(v.le.37) then
  cos_fi=0.002*v+0.601
elseif(v.le.50) then
  cos_fi=6.154e-3*v+0.447
elseif(v.le.62) then
  cos_fi=7.916667e-3*v+0.3592
else
  cos_fi=0.85
endif

elseif(sifralok.eq.441) then
  if(v.le.4) then
    cos_fi=0.40
  elseif(v.le.10) then
    cos_fi=0.03333*v+0.26667
  elseif(v.le.40) then
    cos_fi=0.12449*v-6.64193e-3*v**2+1.57073e-4*v**3-1.366294e-6*v**4-
    *0.1231
  elseif(v.le.80) then
    cos_fi=0.000825*v+0.754
  elseif(v.gt.80) then
    cos_fi=0.82
  endif
endif
return
end
```


Ispis u BORLAND C++

```
//-----
/*u datoteci vcl.h nalaze se funkcije Borland Buildera C++ potrebne
za razvoj GUI-a*/
#include <vcl.h>
#include <math.h>
#pragma hdrstop

/*U datoteci Dodatak.h su deklarirane sve varijable i funkcije*/
/*Uključujući i one koje su potrebne za GUI*/
#include "Dodatak.h"

//-----
#pragma package(smart_init)
/*U datoteci sa indeksom.dfm nalaze se postavke glavnog prozora i
svih ostalih komponenti GUI-a koje se koriste u programu(oznake, polja
unosa, izbornici)*/

#pragma resource "*.dfm"
/*definicija globalnih konstanti*/
/*lako se mijenjaju ako se pokazu pogresnima, a nisu*/

const float epsilon=0.06;
const float eta=0.85;
const float a=0.5;
const float CosFiPom=0.9;
const float Ppom=0.25;
const float Qpom=Ppom*sqrt(1-pow(CosFiPom,2))/(CosFiPom);
Boolean Ucitan=false;

TForm1 *Form1;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
: TForm(Owner)
{
}
//-----

/* ako se mijenja oznaka vucnog vozila*/
void __fastcall TForm1::ComboBoxSifraLokChange(TObject *Sender)
{
    EditMasaLok->Text=ComboBoxSifraLok->Text;
    if((ComboBoxSifraLok->ItemIndex!=3) && EditMasaVlaka->ReadOnly==true)
    {
        EditMasaVlaka->ReadOnly=false;
        EditMasaVlaka->Color=clWindow;
        EditMasaVlaka->Text="";
        EditBrojLok->ReadOnly=false;
        EditBrojLok->Color=clWindow;
        EditBrojLok->Text="";
    }
    /*ovisno o oznaci vucnog vozila, upisuje njegovu masu*/
```

```
switch(ComboBoxSifraLok->ItemIndex)
{
    case 0:
        EditMasaLok->Text="82";
        break;
    case 1:
        EditMasaLok->Text="82";
        break;
    case 2:
        EditMasaLok->Text="123";
        break;
    /*upisuje i masu cijele kompozicije*/
    case 3:
        EditMasaLok->Text="67";
        EditMasaVlaka->Text="180";
        EditMasaVlaka->ReadOnly=true;
        EditMasaVlaka->Color=clSilver;
        EditBrojLok->Text="1";
        EditBrojLok->ReadOnly=true;
        EditBrojLok->Color=clSilver;
        break;
}
SifraLok=ComboBoxSifraLok->Text.ToInt();

}
//-----
/*racuna raspolozivu vucnu silu*/
float TForm1::Sila()
{
    float SilaLok;

    switch(SifraLok){
        case 1141:
            if(VTren<=88.57) SilaLok=150;
            else if(VTren<=110) SilaLok=-1.41657*VTren+275.465;
            else if(VTren<=140) SilaLok=-0.000595*pow(VTren,3)+
                0.24991*pow(VTren,2)-36.1792*VTren+1867.39;
            else SilaLok=0;
            break;
        case 1142:
            if(VTren<=81.25) SilaLok=166.25;
            else if(VTren<=160) SilaLok=4.4735*pow(10,-12)*pow(VTren,8)-
                4.37454*pow(10,-9)*pow(VTren,7)+1.85933*pow(10,-6)*pow(VTren,6)-
                0.000448607*pow(VTren,5)+0.0671942*pow(VTren,4)-
                6.39767*pow(VTren,3)+
                378.129*pow(VTren,2)-12687.2*VTren+185274;
            else SilaLok=0;
            break;
        case 1161:
            if(VTren<=58.89) SilaLok=236.5;
            else if(VTren<=84.44) SilaLok=0.0000979835*pow(VTren,4)-
                0.0289928*pow(VTren,3)+3.25908*pow(VTren,2)-167.299*VTren+3528.95;
            else if(VTren<=118.06) SilaLok=-0.000459768*pow(VTren,4)+
```

```
        0.175031*pow(VTren,3)-24.8272*pow(VTren,2)+1551.64*VTren-35840.7;
        else SilaLok=0;
        break;
    case 6111:
        if(VTren<=73.68) SilaLok=55.172;
        else if(VTren<=120) SilaLok=3.42759*pow(10,-7)*pow(VTren,5)-
        0.000165625*pow(VTren,4)+0.0318603*pow(VTren,3)-
        3.04356*pow(VTren,2)+
        143.602*VTren-2609.62;
        else SilaLok=0;
        break;
    }
    SilaLok=BrojLok*SilaLok*1000;
    return(SilaLok);
}
/*racuna CosFi*/
float TForm1::CosFi()
{
    double CosF;

    switch(SifraLok){
        case 1141:
        case 1161:
            if(VTren<=60)          CosF=-2*pow(10,-9)*pow(VTren,5)+3.70833*pow(10,-
            7)*pow(VTren,4)-
            0.0000245833*pow(VTren,3)+0.000602917*pow(VTren,2)+0.00167833*VTren+0.646;
            else CosF=0.858;
            break;
        case 1142:
            if(VTren<=45)      CosF=-3.0582*pow(10,-9)*pow(VTren,5)+1.89153*pow(10,-
            7)*pow(VTren,4)+
            1.46296*pow(10,-6)*pow(VTren,3)-
            0.000410423*pow(VTren,2)+0.0272994*VTren+0.141;
            else if(VTren<=45.67) CosF=-0.247761*VTren+12.0323;
            else if(VTren<=90) CosF=-1.07016*pow(10,-8)*pow(VTren,5)+
            3.8664*pow(10,-6)*pow(VTren,4)-0.000554279*pow(VTren,3)+
            0.0394308*pow(VTren,2)-1.38768*VTren+19.9539;
            else CosF=0.866;
            break;
        case 6111:
            if(VTren<=27)          CosF=-2.61905*pow(10,-6)*pow(VTren,3)-
            0.000136429*pow(VTren,2)
            +0.0279262*VTren+0.25;
            else if(VTren<=27.17) CosF=-0.741176*VTren+20.8448;
            else if(VTren<=40) CosF=-0.00030469*pow(VTren,3)
            +0.0321124*pow(VTren,2)-1.11452*VTren+13.414;
            else if(VTren<=59)CosF=0.00721053*VTren+0.424579;
            else CosF=0.85;
            break;
    }

    return(CosF);
}
```

```
}
/*racuna otpore voznje*/
float TForm1::Otpori()
{
    float OtporPR, OtporST, SumaOtpora;

    if(R==0) OtporPR=MVlak*Uspon*10;
    else OtporPR=MVlak*(Uspon+(800/R))*10;

    /*ovisno o koeficijentu k kojeg ucitava iz glavnog prozora racuna
    stalne otpore vuce*/
    OtporST=(20+(VTren*VTren/240))*MLok*BrojLok+(20+(k*VTren*VTren/10))*
        (MVlak-MLok*BrojLok);

    SumaOtpora=OtporPR+OtporST;
    return(SumaOtpora);
}
/*potprogram vraca raspolozivu silu za kocenje*/
float TForm1::SilaKoc()
{
    float FKocTemp;

    switch(SifraLok){
        case 1161:
            if(VTren<=40) FKocTemp=6.16*VTren;
            else FKocTemp=0.016*pow(VTren,2)-4.408*VTren+397.04;
            break;
        case 1141:
            if(VTren<=40) FKocTemp=4.4*VTren;
            else FKocTemp=0.0114*pow(VTren,2)-3.1486*VTren+283.6;
            break;
        case 1142:
            if(VTren<=70) FKocTemp=1.8722*VTren;
            else FKocTemp=+0.0059*pow(VTren,2)-2.2241*VTren+257;
            break;
        case 6111:
            if(VTren<=40) FKocTemp=0.9808*VTren;
            else FKocTemp=-0.0076*pow(VTren,2)+1.2588*VTren+1.2318;

            break;
    }
    FKocTemp=BrojLok*FKocTemp*1000;
    return(FKocTemp);
}

void TForm1::ConstBrzina()
{
    float Brzina=VTren/3.6;
    /*oznaka modela voznje*/
    ModelVoznje=0;
    /*fizikalni model*/
    FLok=Otpori();
    if(FLok<0)
```

```
{
FKoc=-SilaKoc();
if(FAth<abs(FLok)) FLok=-FAth;
/*pretpostavka da uvijek ima dovoljno kocne sile*/
if(abs(FLok)<abs(FKoc)) FKoc=FLok;
/*ako je odabrana opcija da se snaga ne vraća u KM*/
if(RadioGroupPovrat->ItemIndex==1)
{
    PEI=Ppom;
    QEI=Qpom;
}
/*snaga se vraća u KM*/
else
{
    PMeh=FKoc*Brzina/1000000;
    PEI=PMeh*eta+Ppom;
    QEI=PMeh*eta*sqrt(1-pow(CosFi(),2))/(CosFi()+Qpom);
}
}
else
{if(FAth<FLok) FLok=FAth;
if(FLok>Sila()) FLok=Sila();
    PMeh=FLok*Brzina/1000000;
    PEI=PMeh/eta+Ppom;
    QEI=PMeh/eta*sqrt(1-pow(CosFi(),2))/(CosFi()+Qpom);
}
}
/*upisuje podatke u datoteku*/
if((s_poc>=0 && smjer_kret==1) || (s_poc<=0 && smjer_kret==-1))
    fprintf(fi,"%6.0f %8.0f %10.3f %10.3fn",Vrijeme,s_poc-SPrijedjeni,PEI,QEI);
else
    fprintf(fi,"%6.0f %8.0f %10.3f %10.3fn",Vrijeme,s_poc+SPrijedjeni,PEI,QEI);

/*racunanje novih vrijednosti*/
    SPrijedjeni=SPrijedjeni+Brzina*KorakProracuna;
}
void TForm1::Ubrzavanje()
{
    float Brzina;
    /*varijabla koja ce se koristiti ako nemma dovoljno
    raspolozive sile da ubrzava sa a*/
    float atemp=a;

    Brzina=VTren/3.6;
    /*oznaka modela voznje*/
    ModelVoznje=1;
    /*fizikalni model*/
    FLok=MVlak*(1+epsilon)*a*1000+Otpori();
    if(FAth<FLok) FLok=FAth;
    if(FLok>Sila()) {
        FLok=Sila();
        atemp=(FLok-Otpori())/(MVlak*(1+epsilon)*1000);}
    /*racunanje snaga*/
    PMeh=FLok*Brzina/1000000;
```

```
PEI=PMeh/eta+Ppom;
QEI=PMeh/eta*sqrt(1-pow(CosFi(),2))/(CosFi()+Qpom);
/*upisuje podatke u datoteku*/
if((s_poc>=0 && smjer_kret==1) || (s_poc<=0 && smjer_kret==-1))
    fprintf(fi,"%6.0f %8.0f %10.3f %10.3fn",Vrijeme,s_poc-SPrijedjeni,PEI,QEI);
else
    fprintf(fi,"%6.0f %8.0f %10.3f %10.3fn",Vrijeme,s_poc+SPrijedjeni,PEI,QEI);
/*medjukorak - trazi vrijeme kada ce vlak dosegnuti planiranu brzinu*/
if(Brzina+atemp*KorakProracuna>VPlan/3.6)
    {float t=(VPlan/3.6-Brzina)/atemp;
    SPrijedjeni=SPrijedjeni+Brzina*t+0.5*atemp*pow(t,2);
    Brzina=VPlan/3.6;
    VTren=VPlan;
    /*fizikalni model*/
    FLok=MVlak*(1+epsilon)*a*1000+Otpori();
    if(FAth<FLok) FLok=FAth;
    if(FLok>Sila()) {
        FLok=Sila();
        atemp=(FLok-Otpori())/(MVlak*(1+epsilon)*1000);}
    /*racunanje snaga*/
    PMeh=FLok*Brzina/1000000;
    PEI=PMeh/eta+Ppom;
    QEI=PMeh/eta*sqrt(1-pow(CosFi(),2))/(CosFi()+Qpom);
    /*upisuje podatke u datoteku*/
    if((s_poc>=0 && smjer_kret==1) || (s_poc<=0 && smjer_kret==-1))
        fprintf(fi,"%6.0f %8.0f %10.3f %10.3fn",Vrijeme,s_poc-SPrijedjeni,PEI,QEI);
    else
        fprintf(fi,"%6.0f %8.0f %10.3f %10.3fn",Vrijeme,s_poc+SPrijedjeni,PEI,QEI);
    SPrijedjeni=SPrijedjeni+Brzina*(KorakProracuna-t)+0.5*atemp*pow((KorakProracuna-
t),2);
    }
    /*ako u narednom koraku postignuta brzina nije veca od planirane*/
    else{
    SPrijedjeni=SPrijedjeni+Brzina*KorakProracuna+0.5*atemp*pow(KorakProracuna,2);
    Brzina=Brzina+atemp*KorakProracuna;
    VTren=Brzina*3.6;}
}

void TForm1::Kocenje()
{
    float Brzina;
    float atemp=a;
    /*Oznaka modela voznje*/
    ModelVoznje=-1;
    /*korekcija kocenja na zadnjih 200 metara, potrebna zbog
gresaka do kojih dolazi pri procesorskom zaokruzivanju*/
    if((UkupniPutDoIduceD-SPrijedjeni<200))
    {
        if(UkupniPutDoIduceD!=SPrijedjeni)
            atemp=(VTren*VTren-VIduca*VIduca)/(2*(UkupniPutDoIduceD-
SPrijedjeni)*3.6*3.6);
        }
    Brzina=VTren/3.6;
```

```
/* fizikalni model*/
FLok=-MVlak*(1+epsilon)*atemp*1000+Otpori();
FKoc=-SilaKoc();
if(FAth<abs(FLok)) FLok=-FAth;
/* pretpostavka da uvijek ima dovoljno kocne sile*/
if(abs(FLok)<abs(FKoc)) FKoc=FLok;
/* ako je odabrana opcija da se snaga ne vraca u KM*/
if(RadioGroupPovrat->ItemIndex==1)
{
    PEI=Ppom;
    QEI=Qpom;
}
/* snaga se vraca u KM*/
else
{
    PMeh=FKoc*Brzina/1000000;
    PEI=PMeh*eta+Ppom;
    QEI=PMeh*eta*sqrt(1-pow(CosFi(),2))/(CosFi())+Qpom;
}
/* upisuje podatke u datoteku*/
if(s_poc>=0 && smjer_kret==1) || (s_poc<=0 && smjer_kret==-1)
    fprintf(fi,"%6.0f %8.0f %10.3f %10.3f\n",Vrijeme,s_poc-SPrijedjeni,PEI,QEI);
else
    fprintf(fi,"%6.0f %8.0f %10.3f %10.3f\n",Vrijeme,s_poc+SPrijedjeni,PEI,QEI);
/* medjukorak - trazi vrijeme kada ce vlak dosegnuti planiranu brzinu*/
if(Brzina-atemp*KorakProracuna<VIDuca/3.6 && (SPrijedjeni<UkupniPut))
{float t=- (VIDuca/3.6-Brzina)/atemp;
SPrijedjeni=SPrijedjeni+Brzina*t-0.5*atemp*pow(t,2);
Brzina=VIDuca/3.6;
VTren=VIDuca;
/* fizikalni model*/
FLok=-MVlak*(1+epsilon)*atemp*1000+Otpori();
FKoc=-SilaKoc();
if(FAth<abs(FLok)) FLok=-FAth;
/* pretpostavka da uvijek ima dovoljno kocne sile*/
if(abs(FLok)<abs(FKoc)) FKoc=FLok;
/* ako je odabrana opcija da se snaga ne vraca u KM*/
if(RadioGroupPovrat->ItemIndex==1)
{
    PEI=Ppom;
    QEI=Qpom;
}
}
/* snaga se vraca u KM*/
else
{
    PMeh=FKoc*Brzina/1000000;
    PEI=PMeh*eta+Ppom;
    QEI=PMeh*eta*sqrt(1-pow(CosFi(),2))/(CosFi())+Qpom;
}
/* upisuje podatke u datoteku*/
if(SPrijedjeni) {
if((s_poc>=0 && smjer_kret==1) || (s_poc<=0 && smjer_kret==-1))
    fprintf(fi,"%6.1f %8.0f %10.3f %10.3f\n",Vrijeme+t,s_poc-SPrijedjeni,PEI,QEI);
```

```
        else
            fprintf(fi,"%6.0f %8.0f %10.3f %10.3f\n",Vrijeme+t,s_poc+SPrijedjeni,PEI,QEI);
        }
    }
    if(SPrijedjeni<UkupniPut)        SPrijedjeni=SPrijedjeni+Brzina*(KorakProracuna-t)-
0.5*atemp*pow((KorakProracuna-t),2);
    }
    /*ako u narednom koraku postignuta brzina nije manja od planirane*/
    else{SPrijedjeni=SPrijedjeni+Brzina*KorakProracuna-
0.5*atemp*pow(KorakProracuna,2);
    Brzina=Brzina-atemp*KorakProracuna;
    VTren=Brzina*3.6;}
}

void TForm1::SaZaletom()
{
    float asz, Brzina;

    Brzina=VTren/3.6;
    /*Oznaka modela voznje*/
    ModelVoznje=2;
    /*fizikalni model*/
    FLok=0;
    asz=-Otpori()/(MVlak*1000*(1+epsilon));
    /*racunanje snaga*/
    PMeh=0;
    PEI=PMeh/eta+Ppom;
    QEI=PMeh*eta*sqrt(1-pow(CosFi(),2))/(CosFi()+Qpom);
    /*upis podataka u datoteku*/
    if((s_poc>=0 && smjer_kret==1) || (s_poc<=0 && smjer_kret==-1))
        fprintf(fi,"%6.1f %8.0f %10.3f %10.3f\n",Vrijeme,s_poc-SPrijedjeni,PEI,QEI);
    else
        fprintf(fi,"%6.0f %8.0f %10.3f %10.3f\n",Vrijeme,s_poc+SPrijedjeni,PEI,QEI);
    /*izracunanje novih vrijednosti*/
    SPrijedjeni=SPrijedjeni+Brzina*KorakProracuna+0.5*asz*pow(KorakProracuna,2);
    Brzina=Brzina+asz*KorakProracuna;
    VTren=Brzina*3.6;
}
/*ako je odabrana opcija ucitavanja*/
void __fastcall TForm1::ButtonUcitajClick(TObject *Sender)
{
    /*pokrece dijalog za ucitavanje*/
    if(OpenDialog->Execute()&&OpenDialog->FileName!="")
    {
        String Ulazna=OpenDialog->FileName;
        char temp;
        int i;

        Ucitana=true;
        i=0;
        while(Ulazna.SubString(Ulazna.Length()-i,1)!="\\") i++;
            i--;
        Izlazna=Ulazna.SubString(1,Ulazna.Length()-i-1);
        EditUlaznaDatoteka->Text=Ulazna.SubString(Ulazna.Length()-i,i-3);
    }
}
```



```
char* cp = new char[ Ulazna.Length() + 1 ];
strcpy( cp, Ulazna.c_str() );
i=0;
/*otvara odabranu ulaznu datoteku, sa svrhom odredjivanja
broja dionica. Ovo je potrebno da se zna koliko memorije
se treba zauzeti*/
if(fp=fopen(cp,"r"))
{
    while(fscanf(fp,"%c", &temp)!=EOF)
    {
        while(temp!='\n')
            fscanf(fp,"%c", &temp);
        i++;
    }
}

BrojDionica=i-2;
}

}

float s_poc;
int smjer_kret;

void TForm1::GlavniProgram()
{
    /*deklaracija varijabli*/
    int *BrojDionice =new int[BrojDionica];
    float *DuljinaDionice =new float[BrojDionica];
    float *RKrivine =new float[BrojDionica];
    float *PromilUspona =new float[BrojDionica];
    float *BrzinaPlanirana=new float[BrojDionica+1];
    int *Zalet =new int[BrojDionica];
    float *SumaDionica =new float[BrojDionica];
    int i;
    int BrD=0;
    char temp;
    int sifra_vlaka, poc_sat, poc_min, kon_sat, kon_min;

    /*citanje podataka iz ulazne datoteke*/
    rewind(fp);
    fscanf(fp,"%d;%d;%d;%d;%d;%f;%d",&sifra_vlaka,
        &poc_sat,&poc_min,&kon_sat,&kon_min,&s_poc,&smjer_kret);
    fscanf(fp,"%c", &temp);
    fscanf(fp,"%c", &temp);
    while(temp!='\n')
        fscanf(fp,"%c", &temp);

    for(i=0;i<BrojDionica;i++)
    {
        fscanf(fp,"%d;%f;%f;%f;%f;%d",&BrojDionice[i],&DuljinaDionice[i],
            &RKrivine[i],&PromilUspona[i],&BrzinaPlanirana[i],&Zalet[i]);
    }
}
```

```
        fscanf(fp,"%c",&temp);
    }

    /*cita podatke iz polja unosa - glavni prozor*/
    VTren=(float)(EditPocetnaBrzina->Text.ToDouble());
    BrzinaPlanirana[i]=(float)(EditKrajnjaBrzina->Text.ToDouble());
    BrojLok=EditBrojLok->Text.ToInt();
    MLok=(float)(EditMasaLok->Text.ToDouble());
    MVlak=(float)(EditMasaVlaka->Text.ToDouble());
    SifraLok=ComboBoxSifraLok->Text.ToInt();
    KorakProracuna=(float)(EditKorakProracuna->Text.ToDouble());
    /*pridjeljuje pocetne vrijednosti*/
    SPrijedjeni=0;
    SKocioni=1;
    SumaDionica[0]=DuljinaDionice[0];
    UkupniPut=0;
    for(i=0; i<BrojDionica;i++)
    {
        SumaDionica[i]=SumaDionica[i-1]+DuljinaDionice[i];
        UkupniPut+=DuljinaDionice[i];
    }

    Vrijeme=0;
    ModelVoznje=0;
    /*racunanje athezione sile*/
    FAth=BrojLok*(MLok*3.3/(1+0.01*VTren))*1000;
    /*sprema vrijednosti prve dionice u globalne varijable*/
    VPlan=BrzinaPlanirana[BrD];
    VIduca=BrzinaPlanirana[BrD+1];
    R=RKrivine[BrD];
    Uspon=PromilUspona[BrD];
    /*otvara datoteku za ispis*/
    String chtemp = Izlazna+EditIzlaznaDatoteka->Text+".izl";
    char* cp = new char[ chtemp.Length() + 1 ];
        strcpy( cp, chtemp.c_str() );
    fi=fopen(cp,"w");
    /*upisuje prvi red sa opisima pojedinih stupaca*/
    fprintf(fi,"%d %d %d %d %d\n",sifra_vlaka,poc_sat,
        poc_min,kon_sat,kon_min);
    /*fprintf(fi,"Vrijeme [s];Predjeni Put [m];P [MW];Q [MVAr];Trenutna Brzina
    [km/h];Model Voznje\n")*/

    /*glavna petlja, vrti se dok se ne predje citavi put*/
    while(SPrijedjeni<UkupniPut)
    {
        /*provjeri da li je pocela nova dionica puta,
        ako je - pridjeli globalnim varijablama njene vrijednosti*/
        if(SPrijedjeni>=SumaDionica[BrD])
        {
            BrD++;
            VPlan=BrzinaPlanirana[BrD];
            VIduca=BrzinaPlanirana[BrD+1];
            R=RKrivine[BrD];
        }
    }
}
```

```
        Uspon=PromilUspona[BrD];
    }
    /*vidi odnos brzina trenutne i planirane na idućoj dionici*/
    if(VTren>VIDuca)
    {
        /*racunaj put potreban za kocenje, varijabla SKocioni
        ce biti <=0 ako treba poceti kociti*/
        SKocT=(pow(VIDuca,2)-
        VTren*VTren)/(3.6*3.6*2*a);
        UkupniPutDoIduceD=SumaDionica[BrD];
        SKocioni=SumaDionica[BrD]+SKocT-SPrijedjeni;
    }
    else SKocioni=1;
    /*treba poceti kociti*/
    if(SKocioni<=0)
        {ModelVoznje=-1;Kocenje();}
    else
        /*ako je u voznji sa zaletom i smije tako voziti i dalje*/
        if(ModelVoznje==2 && Zalet[BrD])
            /*ako se brzina spustila ispod dozvoljene*/
            if(VTren<=0.8*VPlan) Ubrzavanje();
            else SaZaletom();
        else
            /*ako se trenutna brzina jednaka planiranoj*/
            if(VTren==VPlan)
                /*ako smije voziti sa zaletom, vozi tako*/
                if(Zalet[BrD])SaZaletom();
                /*inace vozi konstantnom brzinom*/
                else ConstBrzina();
            /*ako trenutna brzina nije jednaka planiranoj*/
            else
                /*ako je veca koci*/
                if(VTren>VPlan) Kocenje();
                /*u protivnom ubrzaj*/
                else Ubrzavanje();
            /*vrijeme se povecava za zadani korak proracuna*/
            Vrijeme+=KorakProracuna;
        }
    }

    /*zatvara izlaznu datoteku*/
    fclose(fi);

    /* oslobadja memoriju*/
    delete [] BrojDionice;
    delete [] DuljinaDionice;
    delete [] RKrivine;
    delete [] PromilUspona;
    delete [] BrzinaPlanirana;
    delete [] Zalet;
}
//-----

void __fastcall TForm1::ButtonIzvrsiClick(TObject *Sender)
{
```

```
if(Ucitan)
{
    /*Ako se trazi izvrsavanje, a nisu upisani svi
    podaci, sam upise pocetnu i krajnju brzinu 0, te
    korak proracuna 1*/
    if(EditBrojLok->Text=="")
        EditBrojLok->Text="1";
    if(EditMasaVlaka->Text=="")
        EditMasaVlaka->Text="100";
    if(EditKrajnjaBrzina->Text=="")
        EditKrajnjaBrzina->Text="0";
    if(EditPocetnaBrzina->Text=="")
        EditPocetnaBrzina->Text="0";
    if(EditKorakProracuna->Text=="")
        EditKorakProracuna->Text="1";
    if(ComboBoxSifraLok->Text=="")
    {
        ComboBoxSifraLok->ItemIndex=0;
        ComboBoxSifraLok->Text="1141";
        EditMasaLok->Text="82";
    }
    GlavniProgram();
}
}
//-----

void __fastcall TForm1::ComboBoxVrstaVlakaChange(TObject *Sender)
{
    /* ako se mijenja vrsta vlaka mijenja se i koeficijent k o kojem ovisi proracun
    stalnih otpora vuce*/
    switch(ComboBoxVrstaVlaka->ItemIndex)
    {
        case 0:
            k=0.108;
            break;
        case 1:
            k=0.057;
            break;
        case 2:
            k=0.047;
            break;

        case 3:
            k=0.04;
            break;
        case 4:
            k=0.032;
            break;
    }
}
//-----
```

Prilog 2

Ispis potprograma za formiranje električne mreže(PEPES) i proračun električnih prilika (GAUS)

```
PROGRAM PEPES2

INCLUDE 'EXEC.FI'

integer*2 SYSTEM
integer I1
character*1 izbor

I1=SYSTEM(' cls ' c)

      open(223,file='evpx1.pom')
      write(223,*) Vrijeme I_fd_1 I_fd_d Qkv_1 Qkv_d P    Q
+S   lambda U_1_k U_d_k'

***** podaci o voznom redu **** (datoteka vred.vred) *****
      open(unit=81,file='vred.vred')
      write(81,*) Vrijeme Vlakovi i udaljenosti od EVP-a'

1   write(*,5)
5   format(///,10x,'IZBOR :',///,20x,'1 - Kontrola reda voznje',///,
+20x,'2 - Proracun ',///,20x,'3 - Zavrsetak rada',///)

      read(*,'(A)')izbor
      if(izbor.ne.'1'.and.izbor.ne.'2'.and.izbor.ne.'3')then
          I1=SYSTEM(' cls ' c)
          goto 1
      endif

      select case(izbor)
          case('1')
              call KONTROLA
          case('2')
              call PRORACUN
c          case('3')
c              call PREGLED
          case('3')
              I1=SYSTEM(' cls ' c)
      end select

      if(izbor.eq.'1'.or.izbor.eq.'2')then
          I1=SYSTEM(' cls ' c)
          goto 1
      endif
      close(223)
```

```

        close(81)

c   RETURN
    END

*-----*
    SUBROUTINE KONTROLA
*-----*

    integer*2 SYSTEM
    integer I1

    I1=SYSTEM(' cls ' c)

    write(*,6)
6   format(///,5x,'Svi nabrojani vlakovi moraju imati tr"sifra".izl da
+toteku')
    read(*,*)
    I1=SYSTEM(' edit sifre ' c)

    return
    end

*-----*
c   SUBROUTINE PREGLED
*-----*

c   integer*2 SYSTEM
c   integer I1
c   character*1 izbor
c
c   I1=SYSTEM(' cls ' c)
c 7  write(*,8)
c 8  format(///,10x,'IZBOR :',////,20x,'1 - Pregled datoteke rezultata
c   +',///,20x,'2 - Graficki prikaz rezultata',////)
c
c   read(*,(A))izbor
c   if(izbor.ne.'1'.and.izbor.ne.'2')then
c     I1=SYSTEM(' cls ' c)
c     goto 7
c   elseif(izbor.eq.'1')then
c     I1=SYSTEM(' cls ' c)
c     I1=SYSTEM(' edit vred.res ' c)
c   else
c     I1=SYSTEM(' PR_EVP ' c)
c   endif
c
c   return
c   end

*-----*
    SUBROUTINE PRORACUN
*-----*
*   Proracun elektricnih prilika u elektrovucnom sustavu   *
*-----*

```

```

integer*2 SYSTEM
  parameter (bvl=70,btr=5,btp=365)
  integer sifravlaka(bvl),no,broj_desno,broj_lijeco,yes,
+k_m_pom_desno(btr),k_p_pom_desno(btr),k_m_pom_lijeco(btr),
+k_p_pom_lijeco(btr)
  integer broj,k.pol_sat,pol_min,d_min,d_sat,poziv,preskok,time,
+dol_sat,dol_min,sat,min,sek,korak_t,poc_sat,poc_min,
+sek_lok,sat_lok,min_lok,sifra,zav_sat,zav_min,br_minus,br_plus

  real udalj_minus(bvl+btr),udalj_plus(bvl+btr),stavi,ptr,qtr,str
  real Pel,Qel,rel_time,pozic,I_fid_l,I_fid_d,U_r,
+ krak_minus,krak_plus,udalj_m_pom(btr),udalj_p_pom(btr),
+transformatori_Uk(btr),transformatori_snage(btr),U_l(btr),
+U_d(btr),U_tr(btr),kut_tr(btr),kut_d(btr),kut_l(btr),kut_r,
+C_baterije(btr)

  character*12 ime_dat(bvl),red*60,da_ne*1

  logical postoji,postoji2,postoji3,postoji4,postoji5,
+ uvjet_1,uvjet_2,uvjet_3,uvjet_4,lo1

  structure / prisutni_vlakovi /
    integer sif
    integer vrsta
    real P
    real Q
    real udaljenost
    character Plus_Minus
  end structure
  structure /rezultati /
    integer vrijeme
    real P
    real Q
    real S
    real cos_fi
    real I_fid_l
    real I_fid_d
  end structure
  record / prisutni_vlakovi / teret(bvl)
  record / rezultati / EVP1(btp)
  record / rezultati / EVP2(btp)
  record / rezultati / EVP3(btp)
  record / rezultati / EVP4(btp)
  record / rezultati / EVP5(btp)
  DATA transformatori_Uk / 5.5 , 5.5 , 10. , 10. , 10. /
  DATA transformatori_snage / 15. , 15. , 15. , 15. , 15. /

  ***** Kapacitet kompenzacijske baterije u mikro Faradima
  DATA C_baterije / 0.0 , 0.0 , 4.209 , 4.209 , 4.209 /

  open(100,file='sifre',ERR=18)

```

```
        broj=0
        poziv=0
        preskok=0
5      continue
        read(100,*,end=10)sifra
        broj=broj + 1
        goto 5
10     continue
        rewind(100)
        I1 = SYSTEM('cls' C)
        print*, ' Ukupan broj vlakova je : ',broj
        print*

        do j=1,broj
            read(100,*) sifravlaka(j)
            write(ime_dat(j),15) sifravlaka(j)
        enddo
15     format('tr',I5.5,'.izl')
        close(100)
        goto 20
18     print*, ' Greska pri otvaranju datoteke. '
        stop
20     continue
        do j=1,broj
            inquire(file=ime_dat(j),EXIST=postoji)
            if (postoji) then
                write(*,*) ime_dat(j)
            else
                print*, ' Ne postoji izlazna datoteka za vlak : ',sifravlaka(j)
                stop
            endif
        enddo
c      inquire(file='vred.res',exist=postoji)
c      if(postoji) then
c          write(*,147)
c147  format(///,' Postoji datoteka vred.res. Izbrisati ? (d/n)')
c          read(*,'(A)') da_ne
c          if(da_ne.eq.'d'.or.da_ne.eq.'D') then
c              I1=SYSTEM('del vred.res'C)
c          endif
c      endif
c      continue
25     continue

        read*
        I1 = SYSTEM('cls'C)
        print*
        print*, ' Unesite vrijeme pocetka promatranja el. prilika'
        print*, ' npr. za 13 h 15 min unesite dva broja 13 i 15 : '
        read*,poc_sat,poc_min
        if((poc_sat.lt.0.or.poc_sat.gt.24).OR.(poc_min.lt.0.or.poc_min
+ .gt.60))goto 25
28     continue
        print*
```



```
    print*, 'Unesite vrijeme zavrsetka promatranja el. prilika'  
    print*, 'npr. za 13 h 25 min unesite dva broja 13 i 25 :'  
    read*, zav_sat, zav_min  
    if((zav_sat.lt.0.or.zav_sat.gt.24).OR.(zav_min.lt.0.or.zav_min  
+gt.60)) goto 28  
    if(zav_sat.lt.poc_sat.or.(zav_sat.eq.poc_sat.and.zav_min.lt.  
+poc_min)) then  
        print*, 'Vrijeme zavrsetka mora biti vece od pocetnog vremena.'  
        GOTO 28  
    ENDIF  
40  continue  
    print*, 'Unesite korak proracuna el. prilika (sec.). '  
    read*, korak_t  
    if(korak_t.le.0) goto 40  
c    I1 = SYSTEM('cls'C)  
30  print*, 'Unesite duljinu " - " i " + " kraka napajanja referentn  
+og EVP-a (m).'  
    read(*,*)krak_minus,krak_plus  
    if((krak_minus.ge.0).OR.(krak_plus.le.0)) goto 30  
c31  WRITE(*,*)' Unesite napon referentnog EVP-a po iznosu i kutu (KV ,  
c  + stupnjevi).'  
c    READ(*,*)U_r,kut_r  
c    IF((U_r.gt.30.or.U_r.lt.25).OR.(kut_r.gt.90.or.kut_r.lt.-90))  
c  +goto 31  
  
222  FORMAT(/, 'Unesite "-" i "+" krak napajanja', I2, '-og paralelnog  
+ desnog EVP-a (m).')  
223  FORMAT(/, 'Unesite "-" i "+" krak napajanja', I2, '-og paralelnog  
+ lijevog EVP-a (m).')  
224  FORMAT(' Unesite napon', I2, '-og paralelnog desnog EVP-a po iznosu  
+ i kutu (KV , stupnjevi).')  
225  FORMAT(' Unesite napon', I2, '-og paralelnog lijevog EVP-a po izn  
+osu i kutu (KV , stupnjevi).')  
  
    WRITE(*,*)  
c    WRITE(*,*)'Koliko EVP-a je u paralelnom radu lijevo od referentnog  
c  + EVP-a ?'  
c    READ(*,*)broj_lijevo  
    DO i=1,broj_lijevo  
37  WRITE(*,223)i  
    READ(*,*)k_m_pom_lijevo(i),k_p_pom_lijevo(i)  
    IF((k_m_pom_lijevo(i).ge.0).OR.(k_p_pom_lijevo(i).le.0))goto 37  
c*****  "produljenje" krakova napajanja referentnog EVP-a  
    krak_minus=krak_minus-k_p_pom_lijevo(i)+k_m_pom_lijevo(i)  
  
        udalj_minus(i)=krak_minus-k_m_pom_lijevo(i)  
        udalj_m_pom(i)=udalj_minus(i)  
c    print*, 'udalj_minus=', udalj_minus(i), ' i=', i  
c    pause  
c32  WRITE(*,225)i  
c    READ(*,*)U_l(i),kut_l(i)  
c    IF((U_l(i).gt.30.or.U_l(i).lt.25).OR.(kut_l(i).gt.90.or.kut_l(i)  
c  +.lt.-90))goto 32
```

```

        ENDDO

c      WRITE(*,*)
c      WRITE(*,*)'Koliko EVP-a je u paralelnom radu desno od referentnog
c + EVP-a ?'
c      READ(*,*)broj_desno
c      DO i=1,broj_desno
36     WRITE(*,222)i
c      READ(*,*)k_m_pom_desno(i),k_p_pom_desno(i)
c      IF((k_m_pom_desno(i).ge.0).OR.(k_p_pom_desno(i).le.0))goto 36
c***** "produljenje"
c      krak_plus=krak_plus+k_p_pom_desno(i)-k_m_pom_desno(i)

c      udalj_plus(i)=krak_plus-k_p_pom_desno(i)
c      udalj_p_pom(i)=udalj_plus(i)
c33    WRITE(*,224)i
c      READ(*,*)U_d(i),kut_d(i)
c      IF((U_d(i).gt.30.or.U_d(i).lt.25).OR.(kut_d(i).gt.90.or.kut_d(i)
c +.lt.-90))goto 33

        ENDDO
c      print*,'krak plus =',krak_plus,' krak minus =',krak_minus
c      pause
c      *-----*
c      *      Slaganje napona transformatora sa lijeva na desno      *
c      *-----*

        j=0

        DO i=broj_lijevo,1,-1
        j=j+1
        U_tr(j)=U_l(i)
        kut_tr(j)=kut_l(i)
        ENDDO

        j=j+1
        U_tr(j)=U_r
        kut_tr(j)=kut_r

        DO i=1,broj_desno
        j=j+1
        U_tr(j)=U_d(i)
        kut_tr(j)=kut_d(i)
        ENDDO

        I1=SYSTEM ('cls 'c)
c      *-----*
c      *      Brisanje starih datoteka      *
c      *-----*
c      I1=SYSTEM('del Ptr1.slk'c)

c      I1=SYSTEM('del Qtr1.slk'c)

```

```

c  I1=SYSTEM (' del Str?.slk 'c)
c  I1=SYSTEM (' del cos_ftr?.slk 'c)
c  I1=SYSTEM (' del I_fid_??.slk' c)
c  I1=SYSTEM (' del U_?_krak.slk' c)
c  I1=SYSTEM (' cls 'c)

*-----*
*  Datoteke sa podaci o naponima krakova napajanja      *
*-----*
c  open(88,file='U_l_krak.slk',status='replace')
c  open(89,file='U_d_krak.slk',status='replace')
c    write(88,83)6,1
c    write(89,83)6,1

*-----*
*                      Globalno vrijeme                    *
*-----*

    sat=poc_sat
    min=poc_min
    sek=0

50  continue
    poziv=poziv+1
    print*,sat,' h ',min,' min ',sek,' sec '
    no=0
    do j=1,broj
        open(unit=j,file=ime_dat(j),Err=18)
        read(j,*)sifra,pol_sat,pol_min,dol_sat,dol_min

**** Da li se vlak nalazi na pruzi u trenutku promatranja ****

    if(pol_sat.lt.sat) then
        if(dol_sat.gt.sat) then
            lo1=.true.
        elseif(dol_sat.eq.sat.and.dol_min.ge.min) then
            lo1=.true.
        else
            lo1=.false.
        endif
    elseif(pol_sat.eq.sat) then
        if(dol_sat.gt.sat) then
            lo1=.true.
        elseif(dol_sat.eq.sat.and.pol_min.le.min.and.
+ dol_min.ge.min) then
            lo1=.true.
        else
            lo1=.false.

```

```
        endif
    else
        lo1=.false.
    endif

    if(lo1) then
        k=1
        sat_lok=pol_sat
        min_lok=pol_min
        d_sat=10
        do while (k.eq.1)
            read(j,*,end=113) rel_time,pozic,Pel,Qel
            sek_lok=mod(rel_time,60.0)
            d_min=int((rel_time-sek_lok)/60)
            min_lok=pol_min+d_min
            if(min_lok.ge.60) then
                d_sat=1
                sat_lok=sat_lok+d_sat
                min_lok=min_lok-60
            endif
        enddo

c    print*,'d_sat= ',d_sat,' sat_lok= ',sat_lok,' min_lok =',min_lok
c    pause

        if(sek_lok.eq.sek.and.min_lok.eq.min.and.sat_lok.eq.sat) then
            k=0
            elseif(d_sat.ne.10) then
                sat_lok=sat_lok-d_sat
            endif
        enddo
    ***** no je brojac vlakova koji se krecu u nekom trenutku
        no=no+1
        teret(no).sif=sifra
        teret(no).P=Pel
        teret(no).Q=Qel
        teret(no).udaljenost=pozic
        teret(no).vrsta=3
        if(ABS(teret(no).udaljenost).lt.1) then
            if(teret(no).udaljenost.lt.0) then
                teret(no).udaljenost=teret(no).udaljenost - 10
            else
                teret(no).udaljenost=teret(no).udaljenost + 10
            endif
        endif
    endif

    else
    endif
113    continue
        close(j,Err=19)
    enddo
    go to 21
19    stop 'Greska pri zatvaranju datoteke '
21    continue
```

```

*   print*, ' Broj vlakova na pruzi : ', no

*****
**  odredjivanje koliko se cvorista nalazi na " - " kraku napajanja **
*****

*-----Zbog sortiranj u prethodnom koraku-----*
  do i=1, broj_lijevano
    udalj_minus(i)=udalj_m_pom(i)
  enddo
*-----*
*----- Kontrola da li su pojedina cvorista preblizu -----*
  do i=1, broj_lijevano
    do j=1, no
      if (abs(udalj_minus(i)-teret(j).udaljenost).lt.1) then
        teret(j).udaljenost=teret(j).udaljenost-10
      endif
    enddo
  enddo
*-----*

  br_minus=broj_lijevano

  do j = 1 , no

    if(teret(j).udaljenost.lt.0) then
      if(abs(teret(j).udaljenost).lt.abs(krak_minus)) then
        br_minus=br_minus + 1
        udalj_minus(br_minus)=teret(j).udaljenost
        teret(j).Plus_Minus='M'
      else
        teret(j).Plus_Minus='O'
*   pause ' Vlak je izašao sa kraka napajanja " - " '
      endif
    endif
  enddo

*****

*-----*
  do i=1, broj_desno
    udalj_plus(i)=udalj_p_pom(i)
*-----*
    do j=1, no
      if (abs(udalj_plus(i)-teret(j).udaljenost).lt.1) then
        teret(j).udaljenost=teret(j).udaljenost+10
      endif
    enddo
  enddo
*-----*

  br_plus=broj_desno

```

```
do j = 1 , no
if(teret(j).udaljenost.gt.0) then
  if(teret(j).udaljenost.lt.krak_plus) then
    br_plus=br_plus + 1
    udalj_plus(br_plus)=teret(j).udaljenost
    teret(j).Plus_Minus='P'
  else
    teret(j).Plus_Minus='O'
*   pause ' Vlak je izasao sa kraka napajanja " + " . '
  endif
endif
enddo
*****

do i = 1,br_minus
do j = i+1,br_minus
  if(udalj_minus(j).lt.udalj_minus(i)) then
    stavi = udalj_minus(j)
    udalj_minus(j)=udalj_minus(i)
    udalj_minus(i)=stavi
  endif
enddo
enddo
do i = 1,br_plus
do j = i+1,br_plus
  if(udalj_plus(j).lt.udalj_plus(i)) then
    stavi = udalj_plus(j)
    udalj_plus(j)=udalj_plus(i)
    udalj_plus(i)=stavi
  endif
enddo
enddo

*   do j = 1, br_minus
*   print*,udalj_minus(j)
*   enddo

*   do j = 1, br_plus
*   print*,udalj_plus(j)
*   enddo
*   pause
***** podaci o voznom redu **** (datoteka vred.vred) *****
  write(81,*)
  write(81,992) sat,min,sek

***** podaci o transformatorima *** ( datoteka vred.tf)*****
  open(unit=80,file='vred.tf',form='formatted',status='unknown')

***** prikaz cvorova *** ( datoteka vred.nod ) *****
c   I1=system('cls')
  open(unit=77,file='vred.nod',form='formatted',status='unknown')
  do j=1,2
    read(77,'(A)') red
```

```
        enddo
            k=1
*       print*, ' br_minus=',br_minus
        do j=1,br_minus
            yes=0
            do i=1,no
                if(teret(i).udaljenost.eq.udalj_minus(j)) then
                    write(*,991)j,teret(i).vrsta,teret(i).P,teret(i).Q,
+teret(i).sif
                    write(77,991)j,teret(i).vrsta,teret(i).P,teret(i).Q,
+teret(i).sif
                    write(81,993)teret(i).sif,teret(i).udaljenost
                yes=1
            endif
        enddo
        IF(yes.eq.1) GOTO 59
        write(*,991)j,1,0.,0.,0
        write(77,991)j,1,0.,0.,0
        write(80,891)j,transformatori_Uk(k),transformatori_snage(k),
+C_baterije(k)
*       +U_tr(k),kut_tr(k)
            k=k+1
59      continue

        enddo
        write(*,991)br_minus+1,1,0.,0.,0
        write(77,991)br_minus+1,1,0.,0.,0
        write(80,891)br_minus+1,transformatori_Uk(k),transformatori_snag
+e(k),C_baterije(k)
*       +U_tr(k),kut_tr(k)
            k=k+1
333   format(2X,I4,5X,F6.1,3X,F6.1)
992   format(1x,i2,':',i2,':',i2,$)
993   format(1x,I5,1x,f9.1,$)
*       print*, ' br_plus=',br_plus
*       pause
        do j = 1 , br_plus
            yes=0
            do i = 1 , no
                if(teret(i).udaljenost.eq.udalj_plus(j)) then
                    write(*,991)j+1+br_minus,teret(i).vrsta,
+teret(i).P,teret(i).Q,teret(i).sif
                    write(77,991)j+1+br_minus,teret(i).vrsta,
+teret(i).P,teret(i).Q,teret(i).sif
                    write(81,993)teret(i).sif,teret(i).udaljenost
                yes=1
            endif
        enddo
        IF(yes.eq.1) GOTO 60
        write(*,991)br_minus+j+1,1,0.,0.,0
        write(77,991)br_minus+j+1,1,0.,0.,0
        write(80,891)br_minus+j+1,transformatori_Uk(k),
+ transformatori_snage(k),C_baterije(k)
```

```
*   U_tr(k),kut_tr(k)
      k=k+1
60   continue

      enddo
991  format(2X,I2,3X,I2,5X,F7.3,4X,F7.3,5X,I5)
891  format(2X,I2,3X,F6.2,3X,F6.2,3X,F6.2)
      close(77)
      close(80)
      print*
***** prikaz grana *** ( datoteka vred.cn ) *****
      open(unit=79,file='vred.cn',form='formatted',status='unknown')
      do j=1,8
        read(79,'(A)') red
      enddo
      print*
      do i=1,br_minus
        if(i.eq.br_minus) then
          write(*,999)i,i+1,abs(udalj_minus(i))/1e3
          write(79,999)i,i+1,abs(udalj_minus(i))/1e3
        else
          write(*,999)i,i+1,abs(udalj_minus(i)-udalj_minus(i+1))/1e3
          write(79,999)i,i+1,abs(udalj_minus(i)-udalj_minus(i+1))/1e3
        endif
      enddo
      do i = 1,br_plus
        if(i.eq.1) then
          write(*,999)br_minus+1,i+1+br_minus,abs(udalj_plus(i))/1e3
          write(79,999)br_minus+1,i+1+br_minus,abs(udalj_plus(i))/1e3
        else
          write(*,999)i+br_minus,i+1+br_minus,
+ abs(udalj_plus(i)-udalj_plus(i-1))/1e3
          write(79,999)i+br_minus,i+1+br_minus,
+ abs(udalj_plus(i)-udalj_plus(i-1))/1e3
        endif
      enddo
999  format(2X,I2,3X,I2,5X,F7.3)
      close(79)
*****
      open(60,file='vred.tim',status='unknown')
      write(60,1001) sat,min,sek
c     write(*,1001) sat,min,sek
1001 format(I2,2X,I2,2X,I2)
      close(60)
      time=((sat-poc_sat)*3600)+((min-poc_min)*60)+sek
      I1=SYSTEM('cls 'C)
      if((no.gt.0).and.((br_minus.ne.0).or.(br_plus.ne.0))) then
        print*,' Proracun elektricnih prilika !!! '
c     I1=SYSTEM('gaus5 'C)
        call gaus(sat,min,sek)
      else
        write(*,*) ' Nema vlakova , preskacem poziv programa Gauss'
        preskok=preskok+1
      endif

```



```
        write(223,166) sat,min,sek,0,0,0,0,0,0,1.0,27.5,27.5
        goto 159
    endif
166  format(1x,i2,':',i2,':',i2,1x,f6.1,1x,f6.1,1x,f6.2,1x,f6.2,1x,f5.2
      +,2x,f5.2,2x,f5.2,2x,f6.2,1x,f6.3,1x,f6.3)

c   open(191,file='EVP1.pom')
c   read(191,*)I_fid_l, I_fid_d
c   read(191,*)I_fid_d
c   read(191,*)ptr,qtr,str,cos_fi
c   close(191)

c   EVP1(poziv-preskok).vrijeme=time
c   EVP1(poziv-preskok).P=ptr
c   EVP1(poziv-preskok).Q=qtr
c   EVP1(poziv-preskok).S=str
c   EVP1(poziv-preskok).cos_fi=cos_fi
c   EVP1(poziv-preskok).I_fid_l=I_fid_l
c   EVP1(poziv-preskok).I_fid_d=I_fid_d

c   inquire(file='EVP2.pom',EXIST=postoji2)
c   if(broj_lijevo+broj_desno.ge.1)then

c   open(192,file='EVP2.pom',status='unknown')
c   read(192,*)I_fid_l
c   read(192,*)I_fid_d
c   read(192,*)ptr,qtr,str,cos_fi
c   close(192)

c   EVP2(poziv-preskok).vrijeme=time
c   EVP2(poziv-preskok).P=ptr
c   EVP2(poziv-preskok).Q=qtr
c   EVP2(poziv-preskok).S=str
c   EVP2(poziv-preskok).cos_fi=cos_fi
c   EVP2(poziv-preskok).I_fid_l=I_fid_l
c   EVP2(poziv-preskok).I_fid_d=I_fid_d

c   endif

c   inquire(file='EVP3.pom',EXIST=postoji3)
c   if(broj_lijevo+broj_desno.ge.2)then

c   open(193,file='EVP3.pom',status='unknown')
c   read(193,*)I_fid_l
c   read(193,*)I_fid_d
c   read(193,*)ptr,qtr,str,cos_fi
c   close(193)

c   EVP3(poziv-preskok).vrijeme=time
c   EVP3(poziv-preskok).P=ptr
c   EVP3(poziv-preskok).Q=qtr
```

```
c  EVP3(poziv-preskok).S=str
c  EVP3(poziv-preskok).cos_fi=cos_fi
c  EVP3(poziv-preskok).I_fid_l=I_fid_l
c  EVP3(poziv-preskok).I_fid_d=I_fid_d

c  endif

c  inquire(file='EVP4.pom',EXIST=postoji4)
c  if(broj_lijevo+broj_desno.ge.3)then

c  open(194,file='EVP4.pom',status='unknown')
c  read(194,*)I_fid_l
c  read(194,*)I_fid_d
c  read(194,*)ptr,qtr,str,cos_fi
c  close(194)

c  EVP4(poziv-preskok).vrijeme=time
c  EVP4(poziv-preskok).P=ptr
c  EVP4(poziv-preskok).Q=qtr
c  EVP4(poziv-preskok).S=str
c  EVP4(poziv-preskok).cos_fi=cos_fi
c  EVP4(poziv-preskok).I_fid_l=I_fid_l
c  EVP4(poziv-preskok).I_fid_d=I_fid_d

c  endif

c  inquire(file='EVP5.pom',EXIST=postoji5)
c  if(broj_lijevo+broj_desno.ge.4)then

c  open(195,file='EVP5.pom',status='unknown')
c  read(195,*)I_fid_l
c  read(195,*)I_fid_d
c  read(195,*)ptr,qtr,str,cos_fi
c  close(195)

c  EVP5(poziv-preskok).vrijeme=time
c  EVP5(poziv-preskok).P=ptr
c  EVP5(poziv-preskok).Q=qtr
c  EVP5(poziv-preskok).S=str
c  EVP5(poziv-preskok).cos_fi=cos_fi
c  EVP5(poziv-preskok).I_fid_l=I_fid_l
c  EVP5(poziv-preskok).I_fid_d=I_fid_d

c  endif

c  open(196,file='KRAKOVI.pom',status='unknown')
c  read(196,*)U_l_kraka
c  read(196,*)U_d_kraka
c  close(196)
c  write(88,157)time,U_l_kraka
c  write(89,157)time,U_d_kraka
```

```

157 format(5x,i5,2x,f12.3)
158 format(5x,i5,2x,e13.7)
159 continue

      sek=sek+korak_t
      if(sek.ge.60) then
        sek=sek-60
        min=min+1
        if(min.eq.60) then
          min=0
          sat=sat+1
        endif
      endif
      endif

      I1=SYSTEM('cls'C)
      uvjet_1=(sat.eq.zav_sat.and.min.lt.zav_min)
      uvjet_2=(sat.eq.zav_sat.and.min.eq.zav_min.and.sek.eq.0)
      uvjet_3=(uvjet_1.or.uvjet_2)
      uvjet_4=(sat.lt.zav_sat)
      if(uvjet_4.or.uvjet_3) goto 50
183 format(I5)
371 continue
82  format(I5/,I5/,I5)
83  format(I5/,I5/)

c   open(81,file='Ptr1.slk',status='replace')
c   open(91,file='Qtr1.slk',status='replace')
c   open(101,file='Str1.slk',status='replace')
c   open(111,file='cos_ftr1.slk',status='replace')
c   open(131,file='I_fid_11.slk',status='replace')
c   open(141,file='I_fid_d1.slk',status='replace')

c       write(81,82)3,1,poziv-preskok
c       write(91,82)4,1,poziv-preskok
c       write(101,82)1,1,poziv-preskok
c       write(111,82)2,1,poziv-preskok
c       write(131,82)5,1,poziv-preskok
c       write(141,82)5,1,poziv-preskok

      do j=1,poziv-preskok
c       write(81,*)EVP1(j).vrijeme,EVP1(j).P
c       write(91,*)EVP1(j).vrijeme,EVP1(j).Q
c       write(101,*)EVP1(j).vrijeme,EVP1(j).S
c       write(111,*)EVP1(j).vrijeme,EVP1(j).cos_fi
c       write(131,*)EVP1(j).vrijeme,EVP1(j).I_fid_1
c       write(141,*)EVP1(j).vrijeme,EVP1(j).I_fid_d
      enddo

c   close(81)
c   close(91)

```

```
c close(101)
c close(111)
c close(131)
c close(141)
```

```
if(postoji2)then
```

```
open(82,file='Ptr2.slk')
open(92,file='Qtr2.slk')
open(102,file='Str2.slk')
open(112,file='cos_ftr2.slk')
open(132,file='I_fid_12.slk')
open(142,file='I_fid_d2.slk')
```

```
write(82,82)3,1,poziv-preskok
write(92,82)4,1,poziv-preskok
write(102,82)1,1,poziv-preskok
write(112,82)2,1,poziv-preskok
write(132,82)5,1,poziv-preskok
write(142,82)5,1,poziv-preskok
```

```
do j=1,poziv-preskok
write(82,*)EVP2(j).vrijeme,EVP2(j).P
write(92,*)EVP2(j).vrijeme,EVP2(j).Q
write(102,*)EVP2(j).vrijeme,EVP2(j).S
write(112,*)EVP2(j).vrijeme,EVP2(j).cos_fi
write(132,*)EVP2(j).vrijeme,EVP2(j).I_fid_1
write(142,*)EVP2(j).vrijeme,EVP2(j).I_fid_d
enddo
```

```
close(82)
close(92)
close(102)
close(112)
close(132)
close(142)
```

```
endif
```

```
if(postoji3)then
```

```
open(83,file='Ptr3.slk')
open(93,file='Qtr3.slk')
open(103,file='Str3.slk')
open(113,file='cos_ftr3.slk')
open(133,file='I_fid_13.slk')
open(143,file='I_fid_d3.slk')
```

```
write(83,82)3,1,poziv-preskok
write(93,82)4,1,poziv-preskok
write(103,82)1,1,poziv-preskok
write(113,82)2,1,poziv-preskok
write(133,82)5,1,poziv-preskok
write(143,82)5,1,poziv-preskok

do j=1,poziv-preskok
write(83,*)EVP3(j).vrijeme,EVP3(j).P
write(93,*)EVP3(j).vrijeme,EVP3(j).Q
write(103,*)EVP3(j).vrijeme,EVP3(j).S
write(113,*)EVP3(j).vrijeme,EVP3(j).cos_fi
write(133,*)EVP3(j).vrijeme,EVP3(j).I_fid_l
write(143,*)EVP3(j).vrijeme,EVP3(j).I_fid_d
enddo

close(83)
close(93)
close(103)
close(113)
close(133)
close(143)

endif

if(postoji4)then

open(84,file='Ptr4.slk')
open(94,file='Qtr4.slk')
open(104,file='Str4.slk')
open(114,file='cos_ftr4.slk')
open(134,file='I_fid_l4.slk')
open(144,file='I_fid_d4.slk')

write(84,82)3,1,poziv-preskok
write(94,82)4,1,poziv-preskok
write(104,82)1,1,poziv-preskok
write(114,82)2,1,poziv-preskok
write(134,82)5,1,poziv-preskok
write(144,82)5,1,poziv-preskok

do j=1,poziv-preskok
write(84,*)EVP4(j).vrijeme,EVP4(j).P
write(94,*)EVP4(j).vrijeme,EVP4(j).Q
write(104,*)EVP4(j).vrijeme,EVP4(j).S
write(114,*)EVP4(j).vrijeme,EVP4(j).cos_fi
write(134,*)EVP4(j).vrijeme,EVP4(j).I_fid_l
write(144,*)EVP4(j).vrijeme,EVP4(j).I_fid_d
enddo

close(84)
close(94)
```

```
close(104)
close(114)
close(134)
close(144)

endif

if(postoji5)then

open(85,file='Ptr5.slk')
open(95,file='Qtr5.slk')
open(105,file='Str5.slk')
open(115,file='cos_ftr5.slk')
open(135,file='I_fid_15.slk')
open(145,file='I_fid_d5.slk')

write(85,82)3,1,poziv-preskok
write(95,82)4,1,poziv-preskok
write(105,82)1,1,poziv-preskok
write(115,82)2,1,poziv-preskok
write(135,82)5,1,poziv-preskok
write(145,82)5,1,poziv-preskok

do j=1,poziv-preskok
write(85,*)EVP5(j).vrijeme,EVP5(j).P
write(95,*)EVP5(j).vrijeme,EVP5(j).Q
write(105,*)EVP5(j).vrijeme,EVP5(j).S
write(115,*)EVP5(j).vrijeme,EVP5(j).cos_fi
write(135,*)EVP5(j).vrijeme,EVP5(j).I_fid_1
write(145,*)EVP5(j).vrijeme,EVP5(j).I_fid_d
enddo

close(85)
close(95)
close(105)
close(115)
close(135)
close(145)

endif

c  rewind(88)
c    read(88,'(A)')red
c    read(88,'(A)')red
c    write(88,183)poziv-preskok
c  close(88)

c  rewind(89)
c    read(89,'(A)')red
c    read(89,'(A)')red
```

```

c      write(89,183)poziv-preskok
c      close(89)

c      I1=SYSTEM (' del EVP?.pom ' c)
c      I1=SYSTEM (' del KRAKOVI.pom ' c)
      return
      end

C***** Model povecanja temperature prema empirijskim formulama
      SUBROUTINE TEMPERATURA(sfid,thetaNU_old,thetaNU_new,thetaKV_old,
+thetaKV_new)

      real sfid(2),thetaNU_old(2),thetaNU_new(2),thetaKV_old(2),
+thetaKV_new(2),I_nu(2),I_kv(2),II_nu(2),II_kv(2)
      PARAMETER (A=0.475,B=1-A)

      do i=1,2
        I_nu(i)=A*sfid(i)
        I_kv(i)=B*sfid(i)
        II_nu(i)=I_nu(i)*I_nu(i)
        II_kv(i)=I_kv(i)*I_kv(i)

        thetaNU_new(i)=(((1.6e-4*II_nu(i))/(1.62-(6e-7*II_nu(i)))-(2e-9*II
+_nu(i)*thetaNU_old(i)))-thetaNU_old(i))*(1.62-(6e-7*II_nu(i))-(2e-
+9*II_nu(i)*thetaNU_old(i)))/3.47)+thetaNU_old(i)

        thetaKV_new(i)=(((1.9e-4*II_kv(i))/(1.47-(7e-7*II_kv(i)))-(3e-9*II
+_kv(i)*thetaKV_old(i)))-thetaKV_old(i))*(1.47-(7e-7*II_kv(i))-(3e-
+9*II_kv(i)*thetaKV_old(i)))/2.89)+thetaKV_old(i)
      enddo

      return
      end

C*****Model povecanja temperature prema formulama grijanja i hladenja
      SUBROUTINE TEMPERATURA2(sfid,thetaNU_old,thetaNU_new,thetaKV_old,
+thetaKV_new)

      real sfid(2),thetaNU_old(2),thetaNU_new(2),thetaKV_old(2),
+thetaKV_new(2),I_nu(2),I_kv(2),II_nu(2),II_kv(2),D_thetaKV(2),
+ro(2)
      PARAMETER (A=0.475,B=1-A,korak_t=2,ce=385,T=270)
      REAL, PARAMETER :: ro20=1.72E-08,alfa=0.0039,ku=7e-05,gama=8960

      do i=1,2
        I_nu(i)=A*sfid(i)
        I_kv(i)=B*sfid(i)
        II_nu(i)=I_nu(i)*I_nu(i)
        II_kv(i)=I_kv(i)*I_kv(i)

C***** korekcija ro s porastom temperature
        ro(i)=ro20*(1+alfa*(20+thetaKV_old(i)))

```

```
D_thetaKV(i)=(II_kv(i)*korak_t*ro(i))/(ku*ku*gama*ce)
thetaKV_old(i)=thetaKV_old(i)+D_thetaKV(i)
thetaKV_new(i)=thetaKV_old(i)*exp(-korak_t/T)
enddo
return
end
```


Prilog 3

Ispis programa za proračun impedancije kontaktne mreže

```
% PROGRAM ZA ODREĐIVANJE IMPEDANCIJE KONTAKTNE MREŽE I
% BLOK-TRANSFORMACIJA MATRICE IMPEDANCIJA
% a - kontakti vod (Cu, 100 mm2)
% b - nosece uže (Cu, 120 mm2)
% d,e - tračnice (povratni vod)
% ro - specifični otpor tla (ohmxm)
ro = 3000;
% vodljivost tračnica (S/km)
y=100;
% l - udaljenost vlaka od EVP-a
% nazivna frekvencija (Hz)
f = 50;
% Ri- djelatni otpor vodiča uz utjecaj skin efekta (ohm/km)
Rk = 0.1759;
Ru = 0.153;
Rs = 0.327;
% Dij - udaljenosti među pojedinim vodovima (m)
Dab = 1.4;
Dad = 5.40;
Dae = 5.41;
Dbd = 6.79;
Dbe = 6.79;
Dde = 1.5;
% rk – polumjer vodica s obzirom na skin efekat (m)
rke = 0.006;
ru = 0.007;
rs = 0.1;
% proračun impedancija pomoću Carsonovih formula
% vlastita impedancija i međuiimpedancije kontaktnog vodica
Zaar = Rk+0.0493;
Zaai = 0.1446*(2.08-log10(rke)+0.5*log10(ro));
Zaa = complex(Zaar,Zaai);
Zabr = 0.04935;
Zabi = 0.1446*(1.97-log10(Dab)+0.5*log10(ro));
Zab = complex(Zabr,Zabi);
Zadr= 0.04935;
Zadi = 0.1446*(1.97-log10(Dad)+0.5*log10(ro));
Zad = complex(Zadr,Zadi);
Zaer= 0.04935;
Zaei = 0.1446*(1.97-log10(Dae)+0.5*log10(ro));
Zae = complex(Zaer,Zaei);
% vlastita impedancija i međuiimpedancije nosećeg užeta
Zbbr =Ru+0.0493;
Zbbi = 0.1446*(2.08-log10(ru)+0.5*log10(ro));
Zbb = complex(Zbbr,Zbbi);
Zba = Zab;
Zbdr = 0.04935;
Zbdi = 0.1446*(1.97-log10(Dbd)+0.5*log10(ro));
Zbd = complex(Zbdr,Zbdi);
```

```

Zber = 0.04935;
Zbei = 0.1446*(1.97-log10(Dbe)+0.5*log10(ro));
Zbe = complex(Zber,Zbei);
% vlastita impedancija i medjuimpedancije tračnica
Zddr = Rs+0.0493;
Zddi = 0.1446*(1.97+5.2*Rs-log10(rs)+0.5*log10(ro));
Zdd = complex(Zddr,Zddi);
Zda = Zad;
Zdb = Zbd;
Zder = 0.04935;
Zdei = 0.1446*(1.97-log10(Dde)+0.5*log10(ro));
Zde = complex(Zder,Zdei);
Zee = Zdd;
Zea = Zae;
Zeb = Zbe;
Zed = Zde;
% stvaranje matrice pogodne za blok-transformaciju
A=[];
A(1,1) = Zaa;
A(1,2) = Zab-Zaa;
A(1,3) = Zad;
A(1,4) = Zae-Zad;
A(2,1) = Zba-Zaa;
A(2,2) = Zbb-Zba-Zab+Zaa;
A(2,3) = Zbd-Zad;
A(2,4) = Zbe-Zbd-Zae+Zad;
A(3,1) = Zda;
A(3,2) = Zdb-Zda;
A(3,3) = Zdd;
A(3,4) = Zde-Zdd;
A(4,1) = Zea-Zda;
A(4,2) = Zeb-Zea-Zdb+Zda;
A(4,3) = Zed-Zdd;
A(4,4) = Zee-Zed-Zde+Zdd;
% blok-transformacija matrice A 4*4 na 3*3
Y=A;
pivot=Y(4,4);
% brisanje 4 stupca matrice A
Y(:,4) = [];
% brisanje 4 retka matrice A
Y(4,:) = [];
% Z matrica 3x3
k=4;
X=[];
for i=1:k-1
    for j=1:k-1
        X(i,j)=Y(i,j)- A(i,k)*A(k,j)/A(k,k);
    end
end
% zamjena treceg i drugog retka matrice Z
K=[];
K(1,:)=X(1,:);
K(3,:)=X(2,:);

```

```
K(2,:)=X(3,:);
% zamjena treceg i drugog stupca matrice Z
L=[];
L(:,1)=K(:,1);
L(:,2)=K(:,3);
L(:,3)=K(:,2);
% blok-transformacija matrice Z 3*3 na 2*2
% brisanje 3 stupca matrice Z
M=[];
M=L;
M(:,3) = [];
% brisanje 3 retka matrice Z
M(3,:) = [];
k=3;
N=[];
% N matrica impedancija voznog i povratnog voda 2*2
for i=1:k-1
    for j=1:k-1
        N(i,j)=M(i,j)- L(i,k)*L(k,j)/L(k,k);
    end
end
ZKM0=N(1,1)+N(2,2)-2*N(1,2);
ZKM1=N(1,1)-(N(1,2)*N(1,2)/N(2,2));
ZKMs=(ZKM0+ZKM1)/2;
disp( 'Elementi matrice transformirane za blok-transformaciju:')
A
disp( 'Elementi matrice oblika 3x3 nakon prve blok-transformacije:')
X
disp( 'Elementi matrice impedancije voznog i povratnog voda:')
N
disp( 'Impedancija kontaktne mreže kada je opterećenje blizu EVP-a:')
ZKM0
disp( 'Impedancija kontaktne mreže kada je opterećenje značajno udaljeno od EVP-a:')
ZKM1
disp( 'Srednja vrijednost impedancije kontaktne mreže ')
ZKMs
S=[];
i=1;
e=2.7183;
for l=1:5:31
    p=N(1,2)/N(2,2);
    kp=sqrt(abs(Zee)*y);
    q=(1-realpow(e,-kp*1))/(kp*1);
    Zkm1=N(1,1)-p*N(1,2)+q*(1-p)*(N(2,2)-N(1,2));
    sprintf('Zkm1 za %d-i km udaljenosti od EVP-a za slučaj 1 kon vod. blok-transf,l)
    Zkm1
    S(i)=Zkm1;
    i=i+1;
end
x=1:5:31
y1=real(S);
figure(1);
plot(x,y1);
```

```
xlabel(' Udaljenost vlaka od EVP-a [km]');
ylabel(' Odogovarajuca impedancija KM-e [ohm/km]');
hold on;
y2=imag(S);
plot(x,y2);
grid;
title('Ovisnost impendancije KM o udaleljenosti za slucaj: 1 kon. vod. blok-transf');
gtext(' X - induktivna komponenta impedancije');
gtext(' R - djelatna komponenta impedancije ');
hold off;
```

Rezultati su sljedeći:

>> Elementi matrice transformirane za blok-transformaciju:

A =

```
0.2252 + 0.8734i -0.1759 - 0.3583i 0.0493 + 0.4304i 0 - 0.0001i
-0.1759 - 0.3583i 0.3288 + 0.7070i 0 - 0.0144i 0 + 0.0001i
0.0493 + 0.4304i 0 - 0.0144i 0.3763 + 0.9267i -0.3270 - 0.4159i
0 - 0.0001i 0 + 0.0001i -0.3270 - 0.4159i 0.6539 + 0.8319i
```

Elementi matrice oblika 3x3 nakon prve blok-transformacije:

X =

```
0.2252 + 0.8734i -0.1759 - 0.3583i 0.0493 + 0.4303i
-0.1759 - 0.3583i 0.3288 + 0.7070i 0 - 0.0143i
0.0493 + 0.4303i 0 - 0.0143i 0.2128 + 0.7188i
```

Elementi matrice impedancije voznog i povratnog voda:

N =

```
0.1314 + 0.6919i 0.0492 + 0.4230i
0.0492 + 0.4230i 0.2129 + 0.7185i
```

Impedancija kontaktne mreže kada je opterećenje blizu EVP-a:

ZKM0 =

```
0.2459 + 0.5645i
```

Impedancija kontaktne mreže kada je opterećenje značajno udaljeno od EVP-a:

ZKM1 =

```
0.1450 + 0.4504i
```

Srednja vrijednost impedancije kontaktne mreže

ZKM_s =

```
0.1955 + 0.5075i
```

ans =

Zkm1 za 1-i km udaljenosti od EVP-a za slucaj 1 kon vod. blok-transf

Zkm1 =

$$0.1551 + 0.4618i$$

ans =

Zkm1 za 6-i km udaljenosti od EVP-a za slucaj 1 kon vod. blok-transf

Zkm1 =

$$0.1467 + 0.4523i$$

ans =

Zkm1 za 11-i km udaljenosti od EVP-a za slucaj 1 kon vod. blok-transf

Zkm1 =

$$0.1459 + 0.4514i$$

ans =

Zkm1 za 16-i km udaljenosti od EVP-a za slucaj 1 kon vod. blok-transf

Zkm1 =

$$0.1457 + 0.4511i$$

ans =

Zkm1 za 21-i km udaljenosti od EVP-a za slucaj 1 kon vod. blok-transf

Zkm1 =

$$0.1455 + 0.4509i$$

ans =

Zkm1 za 26-i km udaljenosti od EVP-a za slucaj 1 kon vod. blok-transf

Zkm1 =

$$0.1454 + 0.4508i$$

ans =

Zkm1 za 31-i km udaljenosti od EVP-a za slucaj 1 kon vod. blok-transf

Zkm1 =

$$0.1454 + 0.4507i$$

x = 1 6 11 16 21 26 31

Prilog 4

Ispis programa za proračun padova napona u kontaktnoj mreži

```
% Proracun padova napona duz voda
P32=0.397;
Q32=0.262;
S32=complex(P32,Q32);
P2=5.918;
Q2=4.521;
S2=complex(P2,Q2);
P1=0.250;
Q1=0.121;
S1=complex(P1,Q1);
R=0.147;
X=0.472;
R1=R*0.9001;
X1=X*0.9001;
R2=R*13.6756;
X2=X*13.6756;
R3=R*9.9924;
X3=X*9.9924;
V1=27.5;
V2=27.5;
V3=27.5;
Vn=27.5;
for i=1:5
    dP3=(P32*P32+Q32*Q32)/abs(V3*V3)*R3;
    dQ3=(P32*P32+Q32*Q32)/abs(V3*V3)*X3;
    dS3=complex(dP3,dQ3);
    P31=P32+dP3;
    Q31=Q32+dQ3;
    S31=S32+dS3;
    P22=P31+P2;
    Q22=Q31+Q2;
    S22=S31+S2;
    dP2=(P22*P22+Q22*Q22)/abs(V2*V2)*R2;
    dQ2=(P22*P22+Q22*Q22)/abs(V2*V2)*X2;
    dS2=complex(dP2,dQ2);
    P21=P22+dP2;
    Q21=Q22+dQ2;
    S21=S22+dS2;
    P12=P21+P1;
    Q12=Q21+Q1;
    S12=S21+S1;
    dP1=(P12*P12+Q12*Q12)/abs(V1*V1)*R1;
    dQ1=(P12*P12+Q12*Q12)/abs(V1*V1)*X1;
    dS1=complex(dP1,dQ1);
    P11=P12+dP1;
    Q11=Q12+dQ1;
    S11=S12+dS1;
    dV1r=(P11*R1+Q11*X1)/abs(V1);
    dV1i=(P11*X1-Q11*R1)/abs(V1);
```

```
dV1=complex(dV1r,dV1i);  
dV2r=(P21*R2+Q21*X2)/abs(V2);  
dV2i=(P21*X2-Q21*R2)/abs(V2);  
dV2=complex(dV2r,dV2i);  
dV3r=(P31*R3+Q31*X3)/abs(V3);  
dV3i=(P31*X3-Q31*R3)/abs(V3);  
dV3=complex(dV3r,dV3i);  
sprintf('V1,V2,V3 u %d. iteraciji',i)  
V1=Vn-dV1  
V2=V1-dV2  
V3=V2-dV3  
end
```

Rezultat toga programa je sljedeći:

ans =

V1,V2,V3 u 1. iteraciji

V1 =

27.3828 - 0.0778i

V2 =

25.6602 - 1.2105i

V3 =

25.5938 - 1.2645i

ans =

V1,V2,V3 u 2. iteraciji

V1 =

27.3810 - 0.0782i

V2 =

25.5152 - 1.2906i

V3 =

25.4439 - 1.3487i

ans =

V1,V2,V3 u 3. iteraciji

V1 =

27.3809 - 0.0782i

V2 =

25.5029 - 1.2973i

V3 =

25.4312 - 1.3557i

ans =

V1,V2,V3 u 4. iteraciji

V1 =

27.3808 - 0.0782i

V2 =

25.5019 - 1.2979i

V3 =

25.4301 - 1.3563i

ans =

V1,V2,V3 u 5. iteraciji

V1 =

27.3808 - 0.0782i

V2 =

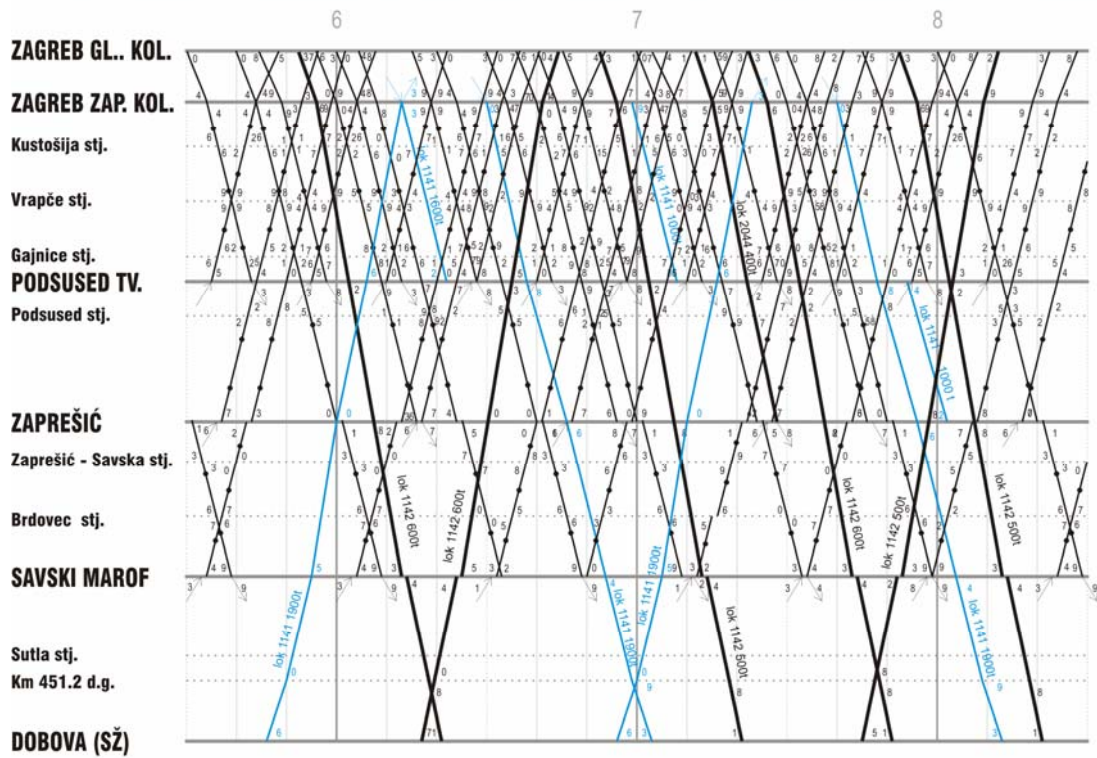
25.5018 - 1.2980i

V3 =

25.4300 - 1.3564i

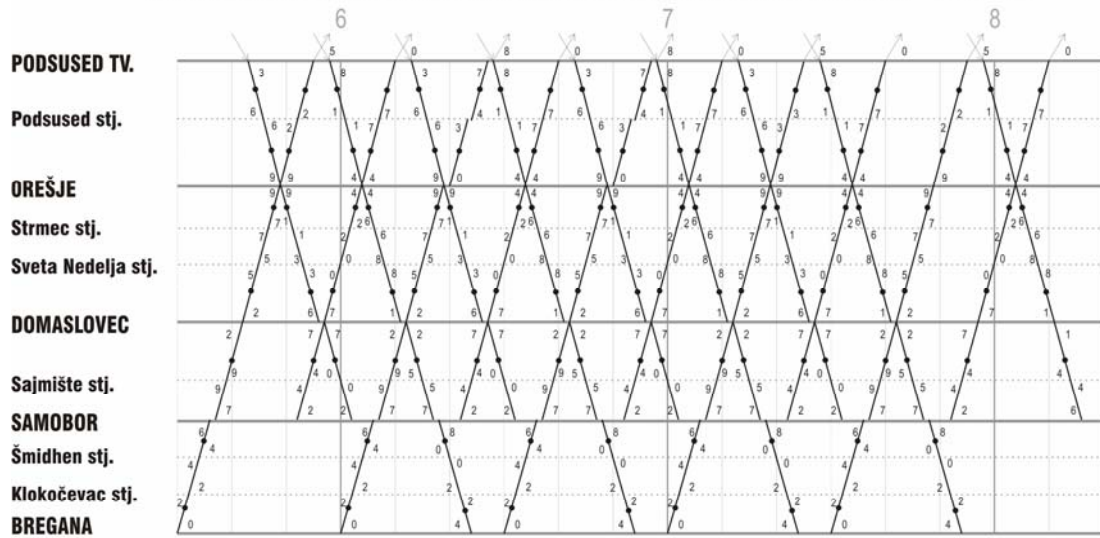
>>

VRŠNI DVOŠATNI MAKSYMALNI GRAFIKON VOZNOG REDA PRUGE ZAGREB G.K. - DOBOVA



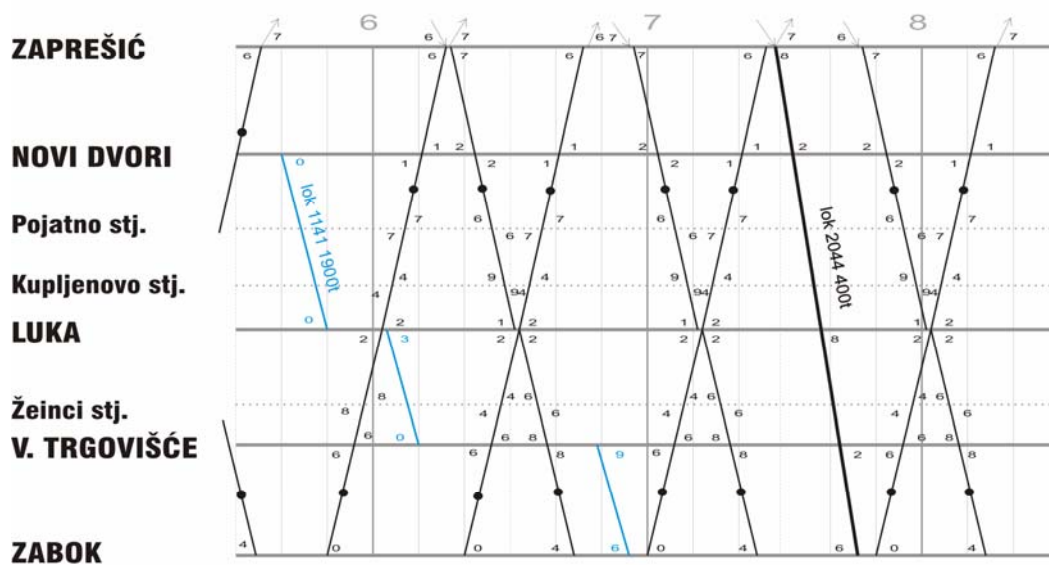
Prilog 5

VRŠNI DVOŠATNI MAKSIMALNI GRAFIKON VOZNOG REDA PRUGE PODSUSED TV. - BREGANA



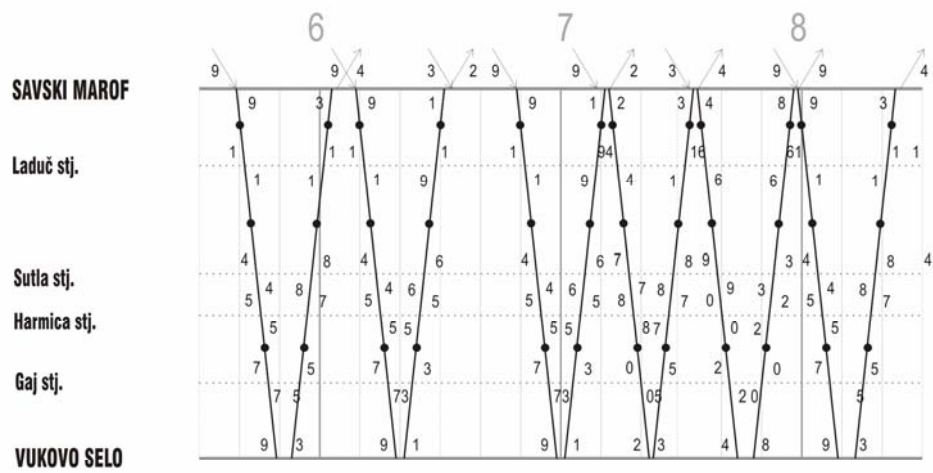
Prilog 6

VRŠNI Dvosatni MAKSIMALNI GRAFIKON VOZNOG REDA PRUGE ZAPREŠIĆ - ZABOK



Prilog 7

VRŠNI DVOSATNI MAKSIMALNI GRAFIKON VOZNOG REDA PRUGE SAVSKI MAROF - VUKOVO SELO



Prilog 8