

Hrvatska zaklada za znanost
Fakultet elektrotehnike i računarstva, Zagreb
Zavod za visoki napon i energetiku

FLEXIBASE

Popis uočenih nepravilnosti u koordiniranom upravljačkom sustavu
koje je potrebno ispraviti

prof. Hrvoje Pandžić
izv. prof. Danijel Topić
dr. sc. Mateo Beus
dr. sc. Ivan Pavić
Domagoj Badanjak, mag. ing.
Ivan Grcić, mag. ing.
Marija Miletić, mag. ing.



Europska unija
Zajedno do fondova EU



01-2022

Sadržaj

1	Uvod	1
2	Pogon mikromreže s obzirom na postavke zaštite	2
2.1	Optimalni tokovi snaga	3
2.2	AC OPF formulacija	4
3	Implementacija algoritma	7
3.1	Implementacija optimizacijskog problema	7
3.2	Rješavanje optimizacijskog problema	9
4	Nepravilnosti	11
4.1	Popis uočenih nepravilnosti u upravljačkom sustavu	11
5	Zaključna razmatranja	13
	Literatura	14

1. Uvod

Jedna od vrlina mikromreža jest mogućnost rada u mrežnom i otočnom načinu rada. Problemi često nastaju pri (iznenadnim) tranzicijama između ta dva načina rada. Naime, moguća je pojava narušavanja frekvencijske stabilnosti, ali i prekoračenje naponskih te strujnih ograničenja. Upravo je neplaniran prelazak u otočni način rada, kao nepovoljniji oblik, jedna od bitnijih sastavnica istraživanja u sklopu ovog projekta. Takav prelazak najčešće uzrokuje djelovanje frekvencijske zaštite, odnosno prekid opskrbe određenog dijela tereta ili zaustavljanje proizvodnje generatora. Što može imati negativne ekonomske i tehničke posljedice. Promjena radnog stanja također utječe na promjenu toka snage unutar mikromreže dovodeći do preopterećenja vodova ili promjene profila napona. Djelovanje nadstrujnih i naponskih releja pri navedenim pojavama je neizbježno, a za posljedicu opet ima prekid napajanja. Pritom valja naglasiti kako fizikalna svojstva komponenti mikromreže omogućavaju kratkotrajna prekoračenja nazivnih ograničenja bez štetnih posljedica po sustav, no algoritmi nisu bili razvijani tako da iskorištavaju tu činjenicu. Zbog navedenih razloga nužan je razvoj algoritma koji u obzir uzima postavke zaštite mikromreže, ali i mrežna ograničenja poput maksimalne snage voda, odabira generatorskih jedinica koje će biti u pogonu te uzimanja u obzir promjenjivost inercije.

2. Pogon mikromreže s obzirom na postavke zaštite

Tijekom dosadašnjeg istraživanja mikromreže su proučavane kroz dva odvojena upravljačka sloja. Gornja i donja razina upravljačkog modela zajednički su za cilj imale brinuti o upravljanju i zaštiti u takvim mikromrežama te o njihovoj interakciji s distribucijskim sustavom i tržišnim sudionicima u svrhu monetizacije fleksibilnosti i upravljivosti koje mikromreže s visokim udjelom pretvarača posjeduju. Kako je svaka razina rješavala specifičan problem, nužno je bilo osigurati i efikasne sustave koordinacije dviju razina.

Gornja razina upravljačkog modela se bavila problemom optimalnog nastupa mikromreža na tržištima energije i pomoćnih usluga, pritom osiguravajući uvjete za prelazak na otočni način rada u bilo kojem trenutku. Drugim riječima, optimizacijski problem gornje razine je za cilj imao minimizaciju operativnih troškova, a tako dobivene optimalne vrijednosti radnih točaka upravljivih jedinica su bili ulazni parametri optimizacijskog problema donje razine.

Na temelju navedenih ulaznih parametara, u donjoj razini se provodio proračun tokova snaga koristeći iterativnu Newton-Raphsonovu (NR) metodu. Primijenjena NR metoda koristila je 0.1 MVA kao baznu vrijednost snage, preciznost je bila postavljena na 0.0001 te je maksimalan broj iteracija bio ograničen na 10000. Sukladno rezultatima proračuna tokova snage, postavke nadstrujne te nad/podnaponske zaštite su bile postavljene na manje konzervativne vrijednosti tijekom određenog vremenskog perioda, ako je to situacija tako zahtijevala. Naime, vodovi su projektirani tako da moraju moći podnijeti uvjete preopterećenja u određenom vremenskom trajanju, a upravo tu činjenicu ovdje razvijeni algoritmi iskorištavaju. U sklopu optimizacijskog problema donje razine je bilo određeno kako uvjeti preopterećenja mogu biti najviše u trajanju od 15-minutnog perioda, nakon kojeg se postavke nadstrujne i naponske zaštite postavljaju natrag na prijašnje postavke. To je sve bilo napravljeno vodeći se razmišljanjem kako je moguće proširiti radno područje mikromreže, osiguravajući da zaštita ne uzrokuje prekid rada prilikom redispeča naređenog od strane gornjeg upravljačkog sloja.

Gore opisane dvije upravljačke razine su pritom na razini cjelokupnog upravljačkog sustava trebale raditi koordinirano kako bi rad mikromreže bio stabilan i siguran, kako u slučaju mrežnog i otočnog načina rada, ali isto tako pri iznenadnom prijelazu jednog u drugi. Pritom su se javljale zapreke i izazovi opisani u prethodnim isporučevinama. Objedinjavanje gornje i donje razine u jednu se pokazalo kao potez koji će ukloniti neke od potencijalnih manjkavosti prethodnog modela. Uzevši u obzir kako se u gornjoj upravljačkoj razini vrti optimizacijski proračun koji za cilj ima odrediti vrijednosti varijabli upravljačkih jedinica, na temelju kojih se onda u donjoj razini računaju tokovi snaga, logičnim se nametnulo rješenje u kojemu se odmah računaju optimalni tokovi snaga.

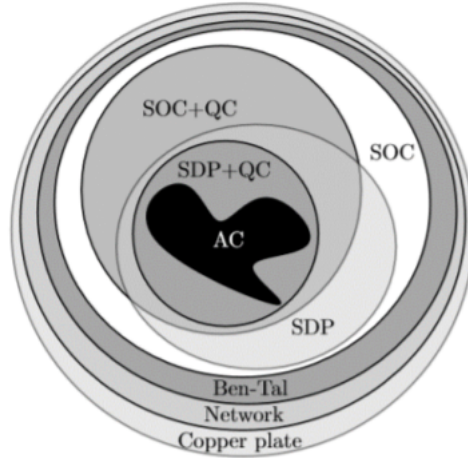
2.1 Optimalni tokovi snaga

Proračun optimalnih tokova snaga (eng. *Optimal Power Flow (OPF)*) je već veoma istraživana tema u akademskim krugovima, te daleko od nepoznanice u industriji. U načelu, to je optimizacijski problem određivanja radnih točaka upravljivih jedinica na takav način da što bolje doprinose funkciji cilja, pritom poštivajući zadana ograničenja. Ovdje su navedena neka od učestalijih ograničenja:

- Ograničenja generatora
- Ograničenja vodova
- Ograničenja naponskih razina na sabirnicama
- Kirchoffovi i Ohmovi zakoni

Odnosno ekonomski dispeč i proračun tokova snaga u jednom. Problem može biti minimizacijski ili maksimizacijski, ovisno kako je postavljena i što je za cilj funkcije cilja. Minimizirati negativne posljedice, ili maksimizirati pozitivne. Primjerice, ukoliko je cilj smanjiti operativne troškove mikromreže, minimizacija troškova se nameće kao logična funkcija cilja. Ograničenja optimizacijskog problema pritom opisuju fizikalne zakonitosti promatrane mreže i njenih komponenti, ali isto tako mogu biti zadana i dodatna ograničenja koja modeliraju neke specifičnosti modeliranog problema. Konkretno, u slučaju ovog projekta, uz standardna ograničenja, model će biti proširen ograničenjima koja u slučaju nenadane situacije omogućavaju privremeno preopterećenje sustava.

Sam po sebi, OPF je veoma složen, načelno nelinearan i nekonveksan optimizacijski problem koji uključuje binarne i kontinuirane varijable. Stoga postoje različite relaksacije punokrvne inačice, koje, ovisno o vrsti samog problema, mogu biti znatno jednostavnije i brže za riješiti nauštrb preciznosti u opisivanju fizikalnih i inih zakonitosti. OPF možemo podijeliti na dvije glavne inačice. AC i DC OPF. AC OPF jest punokrvna verzija s kompletnim AC jednadžbama tokova snage, uključujući aktivnu i reaktivnu snagu. Iako takva formulacija zorno predočava promatrani model (mikro)mreže, problem se javlja prilikom rješavanja optimizacijskog problema, naime nelinearnost i nekonveksnost uzrokuju poteškoće, ali i nemogućnost dobivanja krajnjih rezultata. Kako bi se proračun prospješio/ubrzao tada su potrebne određene relaksacije problema, s naglaskom na transformaciju problema u konveksan i linearizaciju nelinearnih izraza. DC OPF je linearna aproksimacija AC OPF-a koja je nastavak na model ekonomskog dispeča te uključuje i tokove snaga. No ne razmatra aktivnu snagu niti gubitake, oslanjajući se na naponske kuteve te aktivnu snagu generatora. Kako određene situacije trebaju brze izračune uz nešto manju preciznost, tada DC OPF predstavlja izvrsno rješenje, unatoč manjkavosti aspekata koje razmatra. No, u slučaju razvoja algoritma za potrebe mikromreža, DC OPF nije opcija. Veoma bitan faktor u mikromrežama su naponska odstupanja, odnosno razine jalove snage. Stoga se za potrebe ovog projekta koristi AC OPF uz potrebne relaksacije.



Slika 2.1: Relaksacije AC OPF modela [1]

Slika 2.2 ilustrira kako različite relaksacije problema proširuju područje mogućih rješenja, te tako prevladavaju probleme konveksnosti i nelinearnosti nauštrb preciznosti dobivenih rezultata. "AC" jest najprecizniji, ali i matematički najzahtjevniji model, dok dno ljestvice preciznosti (i vrh jednostavnosti proračuna) drži slučaj u kojemu se mreža razmatra kao "bakrena ploča". Drugim riječima, tada fizikalna ograničenja mreže uopće nisu uzeta u obzir.

2.2 AC OPF formulacija

U nastavku je dana AC OPF formulacija kojom se problem koji je do sada razmatran u dvije razine svodi na jednu razinu. Osnovni model koji sadrži glavne karakteristike kako bivše gornje, tako i bivše donje razine će još biti nadograđen skupom ograničenja koja će za cilj imati nesmetane nenadane prelaske u otočni način rada bez bespotrebnih aktivacija instalirane zaštite.

Funkcija cilja (2.1) u osnovnom obliku za cilj ima minimizaciju troškova pogona, čemu dolje priložena matematička formulacija i svjedoči.

$$\min_{p_t^{\text{gen}}} \sum_t^T \sum_i^N p_{i,t}^{\text{gen}} \cdot \lambda^{\text{gen}} \quad (2.1)$$

Ograničenje (2.2) predstavlja klasičnu jednadžbu toka aktivne snage. Naime, tok aktivne snage između dva čvorišta ij ovisi o otporu voda koji ih povezuje, struji koja protječe tim vodom, neto iznosu proizvodnje/potrošnje i tokovima snaga prema daljnjim povezanim čvorištima.

$$p_{ij,t} = r_{ij} \cdot i_{ij,t} - (p_{j,t}^{\text{gen}} - p_{j,t}^{\text{d}}) + \sum_{m:j \rightarrow m} p_{jm,t}, \quad \forall m, i, j, t \quad (2.2)$$

Analog slučaju za tok aktivne snage jest ograničenje (2.3). Ono modelira tok reaktivne snage između dva čvorišta, s time da se umjesto otpora u obzir uzima admitancija x_{ij} , iznosi jalove snage te susceptancija b_{ij} .

$$q_{ij,t} = x_{ij} \cdot i_{ij,t} - (q_{j,t}^{\text{gen}} - q_{j,t}^{\text{d}}) + \sum_{m:j \rightarrow m} q_{jm,t} + v_j \cdot \frac{b_{ij}}{2}, \quad \forall m, i, j, t \quad (2.3)$$

Pad napona između dva susjedna čvorišta opisan je izrazom (2.4), a ovisi o karakteristikama voda (otpor, admitancija) te toku snage i struje tim vodom.

$$v_{j,t} = v_{i,t} - 2 \cdot (r_{ij} \cdot p_{ij,t} + x_{ij} \cdot q_{ij,t}) + i_{ij,t} \cdot (r_{ij}^2 + x_{ij}^2), \quad \forall i, j, t \quad (2.4)$$

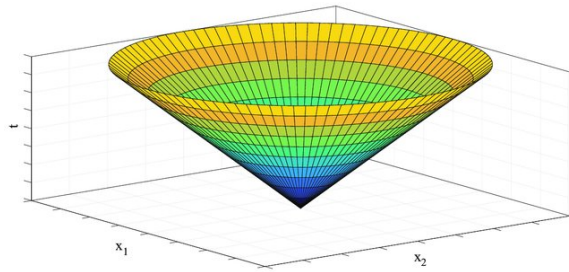
Ograničenje (2.5) u najpreciznijoj formi bi trebalo biti jednakost, a ne nejednakost. Nažalost, takva formulacija za posljedicu ima nekonveksnost cijelog problema, čime isti postaje nerješiv za većinu današnjih komercijalnih solvera, a u najboljem slučaju usporava vrijeme izračuna. Iz tog razloga je korištena metoda stošca drugog reda (eng. *Second Order Cone Programming (SOCP)*), te je izraz relaksiran u nejednakost koja je rezultirala stožastim prikazom u prostoru. Umjesto promatranja samo plašta stošca sa slike [2] promatra se prostor cijelog geometrijskog tijela, a time je očuvana konveksnost problema. Ograničenje kao takvo povezuje četiri varijable: i) kvadrat struje, ii) tok radne snage, iii) tok jalove snage i iv) kvadrat napona. Drugim riječima, ograničenje predstavlja maksimalno termalno opterećenje svakog voda u normalnom pogonu.

$$i_{ij,t} \geq \frac{p_{ij,t}^2 + q_{ij,t}^2}{v_{i,t}}, \quad \forall i, j, t \quad (2.5)$$

Ograničenja (2.6) - (2.8) definiraju minimalne i maksimalne dozvoljene vrijednosti za radnu snagu, jalovu snagu i kvadrate napona. Binarne varijable $x_{i,t}^{\text{gen}}$ određuju za svaki generator je li u pogonu (ako je binarna varijabla jednaka jedinici) ili je ugašen u jedinici vremena t .

$$P_i^{\text{gen,min}} \cdot x_{i,t}^{\text{gen}} \leq p_{i,t}^{\text{gen}} \leq P_i^{\text{gen,max}} \cdot x_{i,t}^{\text{gen}}, \quad \forall i, t \quad (2.6)$$

$$Q_i^{\text{gen,min}} \cdot x_{i,t}^{\text{gen}} \leq q_{i,t}^{\text{gen}} \leq Q_i^{\text{gen,max}} \cdot x_{i,t}^{\text{gen}}, \quad \forall i, t \quad (2.7)$$



Slika 2.2: Prikaz stošca u trodimenzionalnom prostoru [1]

$$V_i^{\min} \leq v_{i,t} \leq V_i^{\max}, \quad \forall i, j, t \quad (2.8)$$

Na gore prikazanu uobičajenu AC OPF formulaciju, potrebno je bilo napraviti još neke promjene kako bi ovaj jednoslojni upravljački model radio isto ono za što su bile zadužene dvije razine zajedničkim snagama iz prijašnje formulacije. Tako je naglasak stavljen na mogućnosti privremenog prekoračenja naponskih i strujnih ograničenja, ali samo u strogo ograničenom vremenskom intervalu te također za unaprijed određene maksimalne vrijednosti. Drugim riječima, u određenim vremenskim intervalima je dopušteno prelaziti granice opterećenja karakteristične za normalni pogon bez aktivacije zaštite, kako bi nenadani prijelazi iz mrežnog u otočni način rada mogli proći. Tu se iskorištava činjenica da komponente mikromreže moraju moći podnijeti preopterećanje u kraćim vremenskim intervalima bez ikakvih negativnih posljedica po njih. Pritom odmah valja naglasiti jedan od problema koji se javlja, a to su vremenske skale modeliranih procesa. Naime, dok se procesi u normalnom pogonu promatraju na (recimo) satnoj razini, privremena prekoračenja imaju puno kraći dozvoljeni period trajanja. Tu dolazi do dileme kako razmatrati pogon mikromreže, uzimajući u obzir i događanja koja se događaju u puno kraćim vremenskim periodima, a da ne pate niti brzina izračuna, niti točnost.

3. Implementacija algoritma

Algoritam za pronalazak optimalnih tokova snaga s ograničenjima za zaštitu opisan u prethodnom poglavlju je potrebno implementirati koristeći neki od dostupnih alata sposobnih za rješavanje optimizacijskih problema ili s mogućnošću povezivanja s vanjskim rješavačem (engl. *solver*). Zahtjevi koji se nameću su vezani za zahtjevnost proračuna (engl. *computational overhead*) odnosno ukupno zauzeće memorije prilikom rješavanja problema te vremensko trajanje istog. Naravno, ovaj se parametar želi minimizirati kako bi se potrebne performanse računala na kojem se problem rješava smanjile, a s time i njegova cijena. Valja napomenuti da će manje zauzeće memorije prilikom izvođenja programa zahtijevati manje radne memorije računala dok će smanjenje kompleksnosti proračuna olakšati procesorskoj jedinici izvršavanje i samim time skratiti trajanje. Naglasak također treba staviti na kompatibilnost alata. Alat koji je lako poveziv, odnosno nudi aplikacijsko programsko sučelje prema ostalim alatima, olakšava te ubrzava testiranje validnosti algoritma.

Imajući u vidu navedene prednosti te ograničenja, za ovaj zadatak je odabran programski alat Matlab. Njegova kompatibilnost s platformom dSpace koja omogućava efikasno ispitivanje algoritama u stvarnom vremenu je bila presudan faktor pri odabiru. Podsjetimo se, platforma dSpace se koristi za ispitivanje sustava u stvarnom vremenu koristeći simulaciju čitavog sustava ili njegova dijela. Konkretno, u ovom projektu se koristi za simulaciju mikromreže na kojoj se tada testiraju algoritmi čime se brzo dobiva povratna informacija o ispravnosti algoritma, kako bi se dalje prilagođavao. dSpace je prilagođen simulaciji modela napravljenih u Matlab Simulink alatu što je, kako je prije navedeno, olakšalo izbor programskog alata.

3.1 Implementacija optimizacijskog problema

Programski alat Matlab nudi dva pristupa rješavanju optimizacijskih problema. Prvi je problemski pristup (engl. *problem-based*), a drugi solverski pristup (*solver-based*). Oba pristupa nude određene prednosti no isto tako sadržavaju nedostatke. Solverski pristup predstavlja funkciju cilja te ograničenja kao funkciju ili matricu. Neke od prednosti su da ne zahtijeva prevođenje problema u matričnu formu te omogućava uštedu memorije pri rješavanju zahtjevnih problema. Međutim, ovakav pristup zapisivanja te pronalaska grešaka, pogotovo ako se radi o velikom problemu, je kompliciran što ga čini manje pogodnim za upotrebu. Problemski pristup dopušta simboličko predstavljanje optimizacijskog problema što implementaciju čini značajno jednostavnijom. Pronalazak grešaka je također olakšan, međutim potreban je prijevod problema u matričnu formu što generalno rezultira duljim vremenom izvedbe. Zbog navedenih razloga je odabran problemski pristup rješavanju problema, te će u nastavku biti opisan.

Simbolički zapis optimizacijskog problema uključuje predstavljanje svih varijabli simboličkim varijablama. Varijable u ovom optimizacijskom problemu označavaju npr. proizvodnju gener-

atora, napone čvorišta, snage između čvorišta te sve ostale navedene u formulaciji problema. Svaka od varijabli može biti ograničena ili neograničena te se zajedno mogu kombinirati u jedno ograničenje. Ograničenja mogu biti u obliku jednakosti ili nejednakosti, što se također može vidjeti iz formulacije problema.

Tradicionalan pristup zapisu ograničenja ili funkcije cilja optimizacijskog problema bi bio korištenjem *for* petlji koje nude vrlo intuitivan zapis problema olakšavajući tako unos korisniku. Pseudokod takvog zapisa jednog od ograničenja problema prikazan je u algoritmu 1. Prilikom ovakvoga unosa jasno je vidljiva međusobna zavisnost varijabli unutar jednadžbe, te ovisnost varijabli o vremenu. Postupak za ostatak formulacije je isti. Međutim, izvršavanje ovog dijela koda je prilično zahtjevno pošto je ukupan broj prolazaka kroz petlju jednak umnošku broja čvorova na kvadrat i duljini vremenskog horizonta. Za fiksni vremenski horizont kompleksnost izvedbe raste s kvadratom broja čvorova, a broj čvorova u mreži može biti znatan. Broj instanci vremenskog horizonta također može rasti ovisno o njegovoj diskretizaciji. Navedeni zaključak je potvrđen i u praksi pošto je ovakva implementacija algoritma rezultirala dugotrajnim vremenom izvedbe. Naime, pokazuje se da formiranje većeg broja ograničenja na ovakav način nije učinkovito u programskom alatu Matlab kao što je to slučaj u većini ostalih alata. Iz tog razloga je odabran, Matlabu povoljniji, matrični pristup kodiranju algoritma. Ipak, ne smije se pomiješati matrični zapis iz solverskog pristupa s ovdje spomenutim matričnim pristupom. Ovdje su varijable i dalje simboličke ali su *for* petlje zamijenjene matričnim množenjem.

Algorithm 1 Kodiranje ograničenja optimizacijskog problema pomoću *for* petlje

```

1: procedure POWER FLOW ▷ Tok snage između čvorova
2:   for i=1:number of nodes do
3:     for j=1:number of nodes do
4:       for t=1:length of horizon do
5:         optim_constr =  $\mathbf{p}(i, j, t) == \mathbf{r}(i, j) * \mathbf{I}(i, j, t) - \mathbf{P}(j, t) + \sum_{k, k \neq i} \mathbf{p}(j, k, t)$ 
6:         ▷ Jednadžba toka snage
7:       end for
8:     end for
9:   end for
10: end procedure

```

Nastavno na prethodni primjer, isto ograničenje je zapisano matrično u algoritmu 2. Sad je tok snage između čvora $i - j$ u trenutku t , prije modeliran kao posebno ograničenje, uklopljen u trodimenzionalnu matricu \mathbf{p} koja na poziciji (i, j, t) sadrži zbroj svih ostalih elemenata na istoj lokaciji u ostalim matricama s druge strane jednakosti. Primijetimo, jedno ograničenje optimizacijskog problema sad, u matričnoj formi, objedinjuje sva ograničenja tokova snaga između čvorišta. Također se može primijetiti da je korištena pomoćna matrica \mathbf{H} koja ima ulogu sume s uvjetom $k \neq i$ kao što je prikazano u algoritmu 1. Ovakav pristup je znatno ubrzao formiranje optimizacijskog problema, spustivši vrijeme izvedbe na nekoliko sekundi.

Algorithm 2 Matrično kodiranje ograničenja optimizacijskog problema

```
1: procedure POWER FLOW ▷ Tok snage između čvorova  
2:    $\text{optim\_constr} = \mathbf{p} == \mathbf{r} \circ \mathbf{I} - \mathbf{P} + \mathbf{H} \cdot \mathbf{p}$  ▷ Jednadžba toka snage  
3: end procedure
```

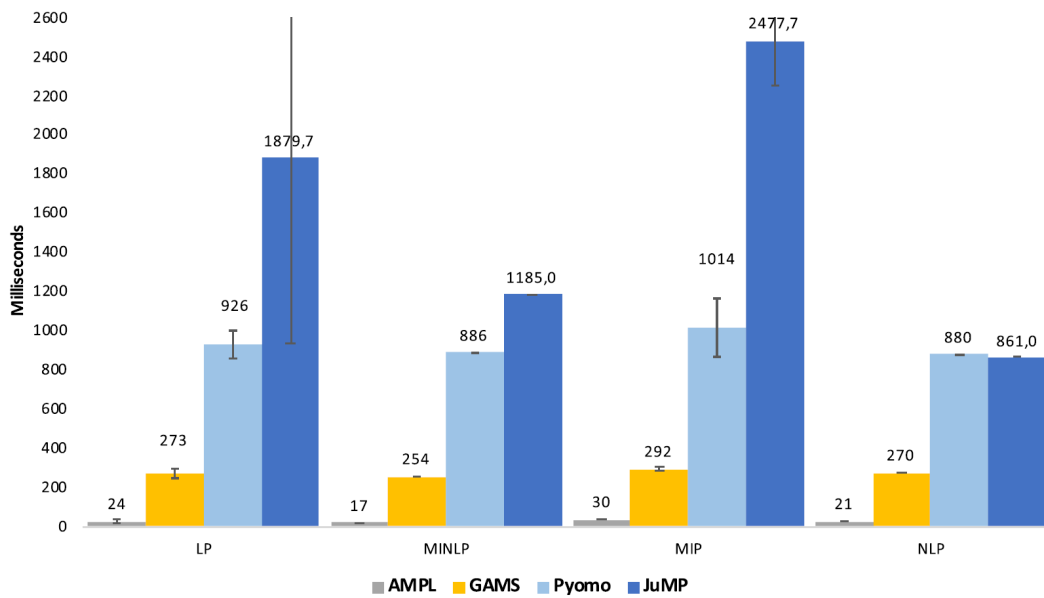
3.2 Rješavanje optimizacijskog problema

Rješavanje optimizacijskih problema se u programskom alatu Matlab provodi pomoću Matlabovih vlastitih rješavača podijeljenih prema vrsti ograničenja (neograničen problem/problem s ograničenjima) i prema vrsti problema (kvadratni, linearni, nelinearni, mješoviti cjelobrojni, itd.). Problem optimalnog toka snaga s ograničenjima za zaštitu s kojim se susrećemo je po svojoj prirodi nekonveksan problem što može komplicirati pronalazak rješenja. Međutim, relaksacijom ograničenja, odnosno ranije opisanim pretvaranjem jednakosti ograničenja u nejednakost ovaj problem postaje konveksan problem koji se mnogo lakše riješi. Ovako relaksiran problem optimalnog toka snaga je po svojoj prirodi stožast (engl. *conic*) [3]. Matlab Optimization Toolbox uključuje rješavače takvih problema (*coneprog* rješavač), no ovaj problem se može riješiti i korištenjem rješavača za općenite nelinearne funkcije više varijabli (*fmincon* rješavač).

Govoreći o vremenu pronalaska rješenja problema, koje se sastoji od vremena potrebnog alatu da formulira problem te vremena potrebnog rješavaču da ga riješi, pokazuje se da je vrijeme formulacije odnosno kreiranja ograničenja te funkcije cilja problema znatno kraće od vremena njegovog rješavanja. Pokazuje se da rješavanje ovog problema koristeći Matlab na stolnom računalu traje prilično dugo (više desetaka minuta) što je previše za stvarnu primjenu pošto će se ovaj algoritam u praksi pozivati svakih nekoliko minuta. Stoga je potrebno unaprjeđenje algoritma s ciljem redukcije vremena izvođenja. Postoji više načina na koji se ovaj cilj može postići, a neki od njih su npr. korištenje metoda za ubrzano rješavanje optimizacijskog problema ili korištenje naprednijeg rješavača. Potonje rješenje znatno olakšava implementaciju no dolazi uz cijenu nabavke softvera. Također, kompatibilnost sa sustavom dSpace je potrebna kako se brzina ispitivanja algoritma ne bi povećala, sprječavajući tako njegov razvoj.

S navedenim problemima na umu, daljnji razvoj će teći u nekom od smjerova koji će raditi na skraćivanju vremena traženja rješenja optimizacijskog problema. Jedan od smjerova koji je moguć je primjena akceleratorne metode za stožaste optimizacijske probleme predložene u [4]. U navedenom radu autori postižu ubrzanje pronalaska rješenja do 87.24%. U radu [5] u kojem se obrađuje problem blizak problemu neplaniranog prelaska mikromreže u otočni način rada, srodne tematike članku [6] objavljenom u sklopu ovog projekta, trajanje rješavanja optimizacijskog problema je svedeno s preko 19 minuta na neznatno iznad jedne minute koristeći Benderovu dekompoziciju. Navedena dekompozicija dopušta razlaganje velikih linearnih programa na blokove reducirajući tako vrijeme izvedbe. Nadalje, postoji mogućnost primjene optimizacijskog alata YALMIP zasnovanog na programskom jeziku Matlab. Njegova je prednost

što može pozivati više vrsta rješavača problema. Tako bi se za rješavanje ovog problema mogao koristiti rješavač Gurobi, dokazan alat za rješavanje konveksnih i nekonveksnih optimizacijskih problema. Također je potrebno ispitati kompatibilnost ostalih optimizacijskih softvera poput GAMS, Julia i AMPL ili neki od šire dostupnih programskih jezika, npr. Python. Usporedba brzine navedenih softvera je provedena u [7], a rezultat se može vidjeti na slici 3.1, sugerirajući na koje alate je potrebno obratiti pozornost. Uspješno povezivanje ovih alata s platformom dSpace bi moglo pridonijeti brzini izvedbe te uspješnosti testiranja algoritma te bi, u krajnjem slučaju, dopustilo diskretizaciju horizonta na kraće periode.



Slika 3.1: Usporedba različitih optimizacijskih softvera [7]

4. Nepravilnosti

U ovom poglavlju je prikazan pregled uočenih nepravilnosti u objedinjenom upravljačkom sustavu, nevezan za implementaciju opisanog algoritma već s naglaskom na ostale parametre koji će morati biti odabrani tako da uzimaju u obzir implementacijska ograničenja algoritma opisana u prethodnom poglavlju. Analiza uočenih nepravilnosti stvorila je pretpostavke za uklanjanje istih, tj. unapređenje rada upravljačkog sustava mikromreže.

4.1 Popis uočenih nepravilnosti u upravljačkom sustavu

Tablica 4.1: Nepravilnosti u upravljačkom sustavu

1.	Duljina predikcijskog horizonta upravljačkog algoritma.
2.	Problem povezivanja rješenja optimizacijskog algoritma s I-t karakteristikama zaštitnih uređaja.
3.	Učestalost pozivanja optimizacijskog algoritma .
4.	Brzina izračuna optimizacijskog problema.
5.	Određivanje vremena diskretizacije dinamičkog modela mikromreže iz koje se izvode ograničenja vezana uz stabilnost frekvencije nakon iznenadnog prelaska iz mrežnog u otočni pogon.
6.	Promjena inercije u mikromreži tokom vremena.

Popis uočenih nepravilnosti u upravljačkom sustavu prikazan je u tablici 4.1. Općenito govoreći najveći problem u koordiniranom upravljačkom sustavu primijećen je kod povezivanja rješenja optimizacijskog problema gornjeg upravljačkog sloja s I-t karakteristikama zaštitnih uređaja u mikromreži. U tom smislu potrebne su prilagodbe koje se odnose na praktičnu implementaciju rješenja optimizacijskog problema. Kao rješenje tog problema predloženo je spajanje gornjeg i donjeg upravljačkog problema u jedinstven optimizacijski problem koristeći algoritam AC OPF-a (engl. Optimal Power Flow - OPF). Naime, u inicijalnom predloženom rješenju drugi upravljački sloj je zadužen za izračun/provjeru tokova snaga u mikromreži koristeći iterativnu Newton-Raphson metodu. U modificiranoj verziji upravljačkog rješenja izračun tokova snaga izveo bi se koristeći algoritam AC OPF-a čija formulacija je dana u Poglavlju 2. Glavna prednost ovakve formulacije je što bi se I-t karakteristike zaštitnih uređaja mogle uključiti kako dodatna ograničenja optimizacijskog problema te bi se na taj način eliminirao problem povezivanja rješenja optimizacijskog problema s I-t karakteristikama zaštitnih uređaja. Problem kod početne formulacije optimizacijskog problema je što je u suštini to nelinearni i nekonveksni optimizacijski problem te je stoga potrebna relaksacija određenih ograničenja kako bi problem bio rješiv standardnim, danas lako dostupnim, rješavačima. Nadalje, kao dodatni nedostatak inicijalno postavljenih upravljačkih slojeva je da oni nisu uzimali u obzir promjenu inercije u

mikromreži do koje dolazi u ovisnosti kako pojedine proizvodne jedinice ulaze u pogon odnosno izlaze iz pogona. U slučaju DED algoritma koji je u osnovi ekonomsko dispečiranje pretpostavlja se da su sve proizvodne jedinice cijelo vrijeme uključene što ne mora nužno biti točno. Prelaskom na AC OPF uvođenjem binarnih varijabli moguće je modelirati i stanje uključenosti pojedinih proizvodnih jedinica. U ovom slučaju iz algoritma zasnovanog na ekonomskom dispečiranju, uvođenjem binarnih varijabli prelazi se na algoritam koji se koristi za određivanje optimalnog pogona proizvodnih jedinica (eng. Unit Commitment). Glavni nedostatak tranzicije s algoritma ekonomskog dispečiranja na AC OPF algoritam je što nakon relaksacije AC OPF algoritam ima oblik mješovito cjelobrojnog optimizacijskog problema čije vrijeme rješavanja može biti dulje nego prethodno korištenog problema ekonomskog dispečiranja koji je bio linearni optimizacijski problem (engl. Linear Programming-LP). Dakle, prelaskom na AC OPF algoritam eliminira se većina nepravilnosti/poteškoća identificiranih kod inicijalno formuliranih problema gornjeg i donjeg upravljačkog sloja. Međutim, potencijalne poteškoće koje se odnose na brzinu izračuna optimalnog rješenja ostaju i u ovom slučaju budući da se koristi složeniji optimizacijski algoritam. Brzina izračuna optimizacijsko algoritma provjerit će se provodeći simulacije koristeći simulator koji ima mogućnost izvođenja simulacija u stvarnom vremenu (engl. real-time simulator). U tu svrhu koristit će se simulator dSPACE SCALEXIO koji je dostupan u Laboratoriju za napredne elektroenergetske mreže (Smart Grid Lab - SGLab) na Fakultetu elektrotehnike i računarstva u Zagrebu.

Govoreći o učestalosti pozivanja optimizacijskog problema nailazi se na ograničenje spomenuto u prethodnom poglavlju. Naime, vrijeme pronalaska rješenja može postaviti donju vremensku granicu pozivanja algoritma što može biti nepovoljno ako se želi postići kraći vremenski period između slijednih poziva algoritma. Nadalje, trajanje perioda je odozgo ograničeno tehničkim ograničenjima vodova pa je pronalazak idealnog trajanja perioda otvoren problem. Inercija mikromreže određuje ponašanje frekvencije prilikom promjene načina pogona npr. prelaskom mikromreže u otočni način rada. Kako je inercija određena svim upogonjenim proizvodnim jedinicama unutar mikromreže, tijekom vremena joj se iznos mijenja te tako utječe na frekven-
cijski odziv u pojedinoj instanci vremenskog horizonta.

5. Zaključna razmatranja

U ovom izvještaju je predstavljen algoritam koji objedinjuje gornji i donji upravljački sloj u jedan zajednički s istim ciljem. Kako bi se to postiglo, AC OPF algoritam je proširen tako da dopušta veću fleksibilnost ograničenja za kratkotrajan pogon mikromreže osiguravajući tako njezin neprekinut pogon. Problemi i nedostaci vezani za samu implementaciju algoritma te odabir hiperparametara upravljačkog sustava su navedeni. Problemi vezani za implementaciju se odnose na brzinu izvedbe algoritma koja se želi svesti na što nižu razinu. Kompatibilnost alata za optimizaciju s alatom za testiranje u stvarnom vremenu pri tome ima važnu ulogu pošto olakšava validiranje. Pošto se optimizacijski alati u osnovi razlikuju po vremenu formiranja i rješavanja problema, njihov odabir također ima značajnu ulogu. Nadalje, problemi vezani za odabir generalnih parametara poput učestalosti pozivanja optimizacijskog algoritma i ostalih navedenih u tablici 4.1 zahtijevaju prilagodbu novonastalom objedinjenom upravljačkom algoritmu.

Mišljenja, nalazi i zaključci ili preporuke navedene u ovom materijalu isključiva su odgovornost autora i ne odražavaju nužno stajališta Hrvatske zaklade za znanost, Ministarstva znanosti i obrazovanja i Europske komisije.

Literatura

- [1] Joshua Adam Taylor. *Convex Optimization of Power Systems*. Cambridge University Press, 2015.
- [2] Slavisa Tomic, Marko Beko, Luis Camarinha-Matos, and Luís Oliveira. Distributed localization with complemented rss and aoa measurements: Theory and methods. *Applied Sciences*, 10:272, 12 2019.
- [3] Carleton Coffrin, Hassan Hijazi, and Pascal Van Hentenryck. Network flow and copper plate relaxations for ac transmission systems. In *2016 Power Systems Computation Conference (PSCC)*, pages 1–8, 2016.
- [4] Sleiman Mhanna, Gregor Verbič, and Archie C. Chapman. Accelerated methods for the socp-relaxed component-based distributed optimal power flow. In *2018 Power Systems Computation Conference (PSCC)*, pages 1–7, 2018.
- [5] Masoud Javadi, Yuzhong Gong, and C. Y. Chung. Frequency stability constrained microgrid scheduling considering seamless islanding. *IEEE Transactions on Power Systems*, 37(1):306–316, 2022.
- [6] Mateo Beus, Ivan Grcić, and Hrvoje Pandžić. Microgrid dispatch with protection constraints. In *2021 International Conference on Smart Energy Systems and Technologies (SEST)*, pages 1–6, 2021.
- [7] Vaidas Jusevičius, Richard Oberdieck, and Remigijus Paulavičius. Experimental analysis of algebraic modelling languages for mathematical optimization. *Informatika*, 32(2):283–304, 2021.