

REZUMAT EXTINS - RAPORTUL ȘTIINȚIFIC ȘI TEHNIC

Cuprins

1	PROIECTAREA ȘI MODELAREA MICROREȚELEI REALIZATĂ PRIN INTERCONECTAREA SUBANSAMBLELOR.....	2
1.1	MODELAREA MICROREȚELEI REZULTATE.....	2
1.2	CONTROLUL MICROREȚELEI.....	3
1.2.1	<i>Nivelul de control primar.....</i>	3
1.2.2	<i>Nivelul de control secundar.....</i>	4
1.3	REZULTATELE MODELULUI PROPUȘ.....	5
2	ELABORAREA ALGORITMILOR DE MANAGEMENT ȘI DE COMUNICAȚIE AI SISTEMULUI BAZAȚI PE CONTROLUL CENTRALIZAT, CONTROLUL PREDICTIV ȘI CONTROLUL TENSIUNEE FRECVENȚA (DROOP CONTROL).....	7
2.1	ALGORITMI DE COMUNICAȚIE.....	7
2.1.1	<i>Interfața de comunicație și control invertoare SMA.....</i>	7
2.1.2	<i>Introducere.....</i>	7
2.1.3	<i>Schema de conectare a invertoarelor.....</i>	8
2.1.4	<i>Funcțiile YASDI.....</i>	9
2.1.5	<i>Funcții utilizate pentru transferul pachetelor de date SMA.....</i>	10
2.1.6	<i>Interfața de comunicație.....</i>	11
2.1.7	<i>Comunicarea între interfața C# și programul de comunicație.....</i>	14
2.1.8	<i>Interfața C#.....</i>	15
2.2	ALGORITMI DE MANAGEMENT AL ENERGIEI – STADIU ACTUAL.....	17
2.2.1	<i>Algoritmul Genetic.....</i>	17
2.2.2	<i>Algoritm de management folosind optimizare de tip roi de particule.....</i>	18
2.2.3	<i>Algoritmi de management folosind programare liniară.....</i>	19
3	PROIECTAREA ȘI SIMULAREA SISTEMULUI EMBEDDED BAZAT PE DSP SAU MICROCONTROLLER PENTRU MANAGEMENTUL ENERGIEI.....	21
4	PROIECTAREA ȘI SIMULAREA PROGRAMULUI PENTRU DIMENSIONAREA MICROREȚELEI.....	22
4.1	PREDICȚIA RADIAȚIEI SOLARE.....	23
4.2	UNGHIUL ORAR PENTRU RĂSĂRITUL SOARELUI.....	24
4.3	STRUCTURAREA INTERFEȚEI DE SIMULARE ȘI OPTIMIZAREA MICROREȚELEI.....	24
5	CONSTRUIREA SUBANSAMBLULUI RESPONSABIL CU OBȚINEREA ENERGIEI PRIN CONVERSIE FOTOVOLTAICĂ. TESTAREA ÎN VEDEREA OPTIMIZĂRII.....	25
1	CONSTRUIREA SUBANSAMBLULUI RESPONSABIL CU OBȚINEREA ENERGIEI PRIN CONVERSIE FOTOVOLTAICĂ.....	25
2	TESTAREA ÎN VEDEREA OPTIMIZĂRII.....	25
5.1	CONVERSIA ENERGIEI MECANICE ÎN ENERGIE ELECTRICĂ. GENERATORUL CU INDUCȚIE.....	26
5.1.1	<i>Echilibrarea generatorului.....</i>	26
5.1.2	<i>Rezultate simulate.....</i>	27
5.1.3	<i>Rezultate experimentale obținute.....</i>	28
5.2	MODELUL DE SEMNAL MICAL AL GENERATORULUI ASINCRON.....	30
5.2.1	<i>Modelul matematic.....</i>	30
5.2.2	<i>Modelul de simulare.....</i>	33
5.2.3	<i>Rezultate și discuții.....</i>	33

1 Proiectarea și modelarea microrețelei realizată prin interconectarea subansamblelor

Structura microrețelei rezultată este reprezentată în figura următoare:

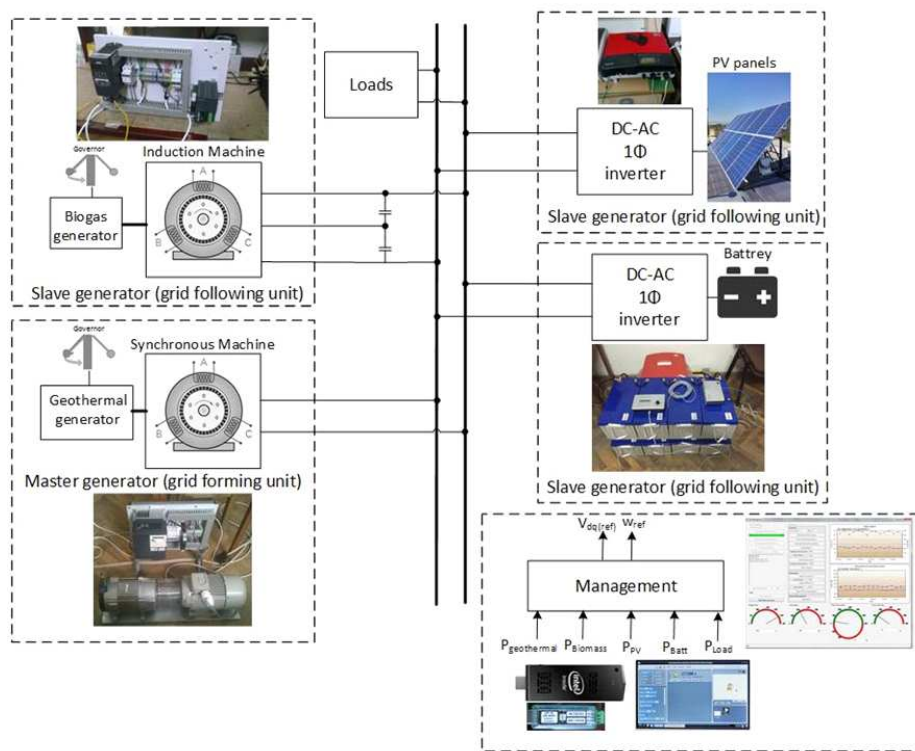


Fig. 1-1 Micro rețeaua rezultată

1.1 Modelarea microrețelei rezultate.

O microrețea este o parte integrantă a unui Smart Grid. De obicei acestea sunt rețele de joasă tensiune compuse din generatoare regenerabile distribuite interconectate, stocare, și sarcini. Microrețelele pot funcționa conectate la rețeaua națională sau izolat (în mod insulă). Pentru a conecta generatoarele se folosesc de obicei echipamente de putere. În acest sens microrețeaua este o colecție de invertor conectate în paralel cu generatoare funcționând împreună pentru a alimenta sarcinile. Se pot defini trei mari categorii de generatoare electrice: generatoare formatoare de rețea, generatoare urmăritoare de rețea și generatoare necontrolabile conectate în paralel pe rețea. Generatoarele formatoare de rețea stabilesc tensiunea și frecvența rețelei. Generatoarele urmăritoare de rețea vor furniza putere activă și reactivă în rețea în funcție de măsurători de tensiune și frecvență și propriile caracteristici de „droop”. Generatoarele necontrolabile cuprind: generatoare eoliene, fotovoltaice, etc. Aceste generatoare vor furniza în rețea puterea maximă disponibilă de la resursa regenerabilă.

Funcționarea corespunzătoare a acestor generatoare depinde de sistemul de control al microrețelei. Metodele de control implementate în literatură se împart în: control de tip master/slave și control de tip droop. Metodele de control Master/Slave au performanțe ridicate dar au un dezavantaj major: funcționarea lor depinde de master și de un sistem de comunicație rapid între master și generatoarele slave. Metoda de control „droop” realizează controlul de putere folosind deviații de tensiune și frecvență măsurate local. Folosind metoda de control de tip „droop” toate generatoarele participă la generarea de energie fără o linie de comunicație, în acest fel defectarea unui generator nu are un impact mare asupra funcționării întregii rețele. Un dezavantaj al acestei metode este variația inerentă a tensiunii și a frecvenței rețelei. Pentru a se evita problemele de mai sus se folosește de obicei o schemă de control pe trei nivele: control primar, control secundar, control terțiar. Controlul

primar este responsabil cu controlul frecvenței și tensiunii astfel încât să se realizeze o împărțire echilibrată a puterii între generatoare. Controlul secundar se ocupă de obicei de restaurarea tensiunii și frecvenței datorate deviațiilor produse de inerția generatoarelor și impedanțelor virtuale. De asemenea la acest nivel se realizează sincronizarea generatoarelor. Controlul terțiar se ocupă de importul și exportul de energie în rețeaua națională. Tot la acest nivel se stabilesc referințele de putere pentru generatoare.

1.2 Controlul microrețelei

Structura de control propusă pentru microrețea este reprezentată în Fig. 1-2.

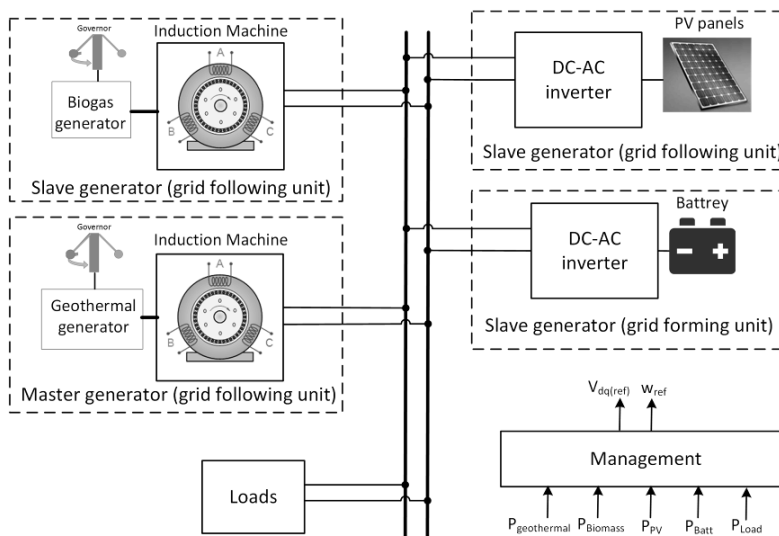


Fig. 1-2 Structura de control a microrețelei

Structura propusă este o combinație a controlului de tip Master/Slave și a controlului de tip „droop”. Se optează pentru o structură de control cu două nivele: nivelul primar în care controlul se bazează pe caracteristicile tensiune-putere reactivă și frecvență-putere activă ale fiecărui generator; și un nivel secundar implementat de sistemul de management a energiei care stabilește referințele de tensiune și frecvență pentru inverterul master în funcție de puterile măsurate din microrețea.

1.2.1 Nivelul de control primar

Toate generatoarele microrețelei sunt conectate în paralel pe rețeaua monofazată. Generatoarele regenerabile furnizează energie în această rețea în funcție de disponibilitatea resurselor regenerabile. Pentru a se putea menține rețeaua stabilă doar un singur generator va putea funcționa ca sursă de tensiune iar restul ca surse de curent. Generatorul geotermal va fi generatorul master și va funcționa ca sursă de tensiune stabilind frecvența și tensiunea rețelei. Referințele de tensiune și frecvență sunt primite de la sistemul de management a energiei (PMS). Frecvența rețelei va putea să varieze în domeniul [49 – 51] Hz. Invertoarele Slave își vor controla nivelul de putere activă de ieșire în funcție de valoarea frecvenței măsurate și caracteristica proprie „P-f droop”.

Controlerile PI se folosesc în mod obișnuit pentru controlul invertoarelor dar au două dezavantaje principale: nu pot urmări semnale sinusoidale fără eroare staționară și au o capacitate redusă de rejecție a perturbațiilor. Pentru a evita aceste dezavantaje controlul invertoarelor va folosi controlerile PI implementate în sistem de referință d-q.

Invertoarele master sunt compuse din două bucle: o buclă de curent și o buclă de tensiune. Referința pentru buclă de curent, Fig. 1-3, este furnizată de buclă externă de tensiune. Referința pentru buclă de tensiune va fi furnizată de sistemul de management a energiei. Tot invertoarele master stabilesc și frecvența rețelei, referința de frecvență fiind furnizată de PMS. În acest fel invertoarele master pot controla puterea de ieșire a invertoarelor slave.

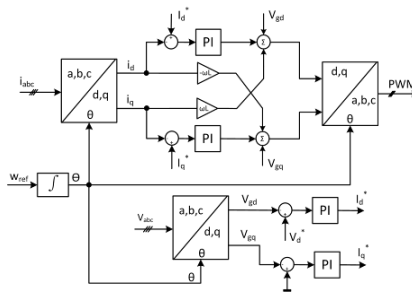


Fig. 1-3 Controlul invertorului Master

Invertorul fotovoltaic este controlat in mod curent constant. Acest tip de control permite funcționarea corectă a acestui inverter în paralel cu celelalte generatoare din microrețea. Referința de curent pentru controlul acestui inverter este derivată din caracteristica de droop a acestui inverter, sistemul de management a energiei și de puterea maximă de la panourile solare. Structura de control propusă pentru acest inverter este prezentată în Fig. 1-4.

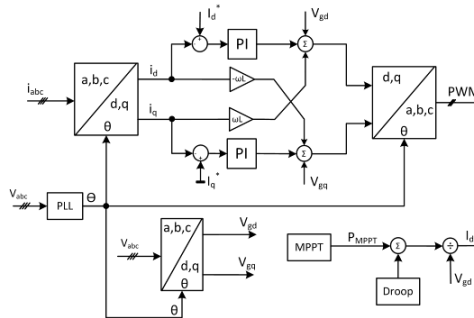


Fig. 1-4 Controlul invertorului fotovoltaic

Caracteristica de droop a acestui inverter este descrisă de Eq.1 și reprezentată în Fig. 1-5. Această caracteristică face ca invertorul fotovoltaic să imite funcționarea unui generator asincron.

$$P(f) = P_o + k \cdot (f_0 - f) \tag{1}$$

unde $P_o = 5\text{ kW}$, $k = 1571\text{ W/Hz}$, $f_0 = 50\text{ Hz}$

Generatorul cu biomasă folosește un generator asincron pentru injectarea energiei în rețea. Acesta funcționează cu viteză constantă controlată de un guvernator. Acest tip de control va face generatorul să funcționeze cu o caracteristică de droop dată de (2).

$$P_m(f) = k_m \cdot (f_0 - f) \tag{2}$$

unde $k_m = 3.84\text{ kW/Hz}$, și $f_0 = 51\text{ Hz}$.

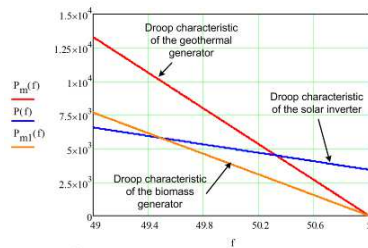


Fig. 1-5 Caracteristicile de droop

1.2.2 Nivelul de control secundar

La nivelul de control secundar sistemul de management a energiei măsoară puterea generată și consumată de sarcini și considerând caracteristicile de droop ale fiecărui generator va decide referința de frecvență pe care o va trimite invertorului master. Nivelul

secundare de control este implementat într-un calculator de uz general și comunică datele pe o magistrală de comunicație RS485 Modbus RTU.

Principala sarcină a PMS este de a furniza puterea necesară sarcinilor critice, de a menține sistemul de stocare încărcat, de implementare a funcțiilor de protecție și de a prezice cazurile cele mai defavorabile și pentru a putea opri sistemul de alimentare. Ca sarcini secundare ale sistemului de management se pot menționa controlul predictiv folosind date meteorologice, prezervarea resurselor limitate (în acest caz biogazul).

Algoritmul de management are două stagii. Primul stadiu implementează funcția de pompare a microrețelei bazat pe resursele regenerabile și starea curentă a generatoarelor. Secvența de start primește date de la generatoare și bazându-se pe disponibilitatea resurselor regenerabile și sarcini va decide ce generatoare vor fi folosite pentru alimentarea sarcinilor. O primă decizie se poate lua pe baza stării de încărcare a bateriilor ținând cont de faptul că invertorul master trebuie să poată alimenta sarcinile în cazul în care celelalte generatoare sunt oprite. În continuare dacă și restul resurselor regenerabile sunt disponibile PMS poate lua decizia să le folosească. Dacă nu toate resursele sunt disponibile atunci PMS va calcula dacă puterea maximă disponibilă este îndeajuns pentru alimentarea sarcinilor. Pot să apară trei cazuri: 1) puterea necesară depășește puterea produsă de generatoare dar sarcinile critice pot fi susținute, caz în care sarcinile necritice vor fi deconectate de la microrețea; 2) consumul datorat sarcinilor nu depășește puterea produsă maximă, caz în care necesarul de energie este împărțit între generatoare; 3) dacă nici sarcinile critice nu pot fi susținute atunci se va decida la trecerea în regim de oprire controlată a generatoarelor. După ce secvența de start se încheie algoritmul de management urmează logica din diagrama din Fig. 16.

Se vor măsura date de la fiecare generator iar puterea maximă disponibilă este calculată și considerată pentru alimentarea sarcinilor. Cunoscând iradiația solară și starea de încărcare a bateriilor și considerând caracteristicile de droop ale generatoarelor se va stabili referința de frecvență astfel încât să nu se consume energie din baterii. Dacă bateriile trebuie încărcate atunci frecvența de referință va fi modificată astfel încât starea de încărcare a bateriilor să revină la cea prestabilită.

1.3 Rezultatele modelului propus

Un model al microrețelei a fost dezvoltat și simulat în PSIM. Fig. 1-6 prezintă răspunsul tranzitoriu al microrețelei la variația sarcinii (P_{load} crește de la 4kW la 5kW, Fig. 1-6a). Se poate observa că invertorul master ține constantă tensiunea de ieșire, Fig. 1-6b, furnizând diferența de putere, Fig. 17e. În Fig. 1-6d sunt de asemenea prezenți curenții I_d și I_q . Trei scenarii vor fi prezentate pentru demonstrarea principiului de funcționare al microrețelei.

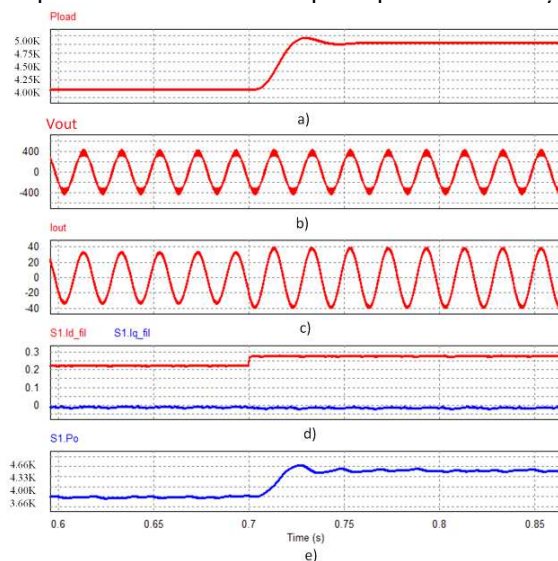


Fig. 1-6 Răspunsul tranzitoriu al sistemului

Caz I: Răspunsul la salt de sarcină

Fig. 1-7 prezintă răspunsul la salt de sarcină al micrețelei. Scenariul considerat se caracterizează prin următorii parametri: puterea generatorului fotovoltaic este 1.2kW, puterea generatorului geotermal este 0.8kW, puterea generatorului cu biomasă este 1.6kW iar consumul este 4kW. Scenariul prezintă un salt de sarcină de 0.75kW la 1.1 sec. Se poate observa în Fig. 1-7 că surplusul de energie este furnizat de inverterul de baterii în primă fază apoi sistemul de management ajustează referința de frecvență astfel încât generatoarele geotermal și cu biomasă să-si mărească puterea de la ieșire și consumul de energie din baterii sa revină la zero.

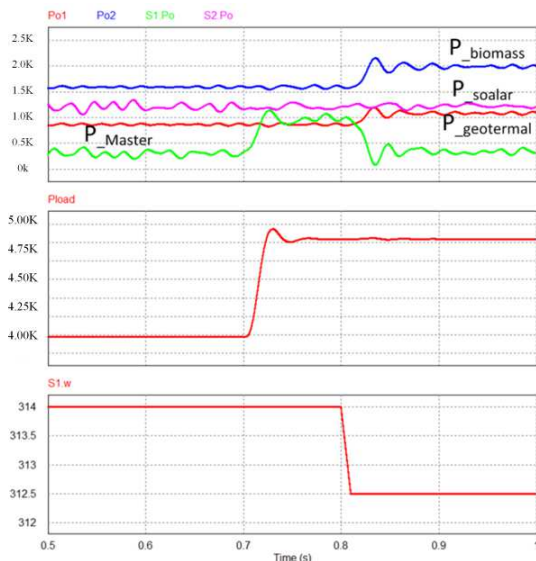


Fig. 1-7 Răspunsul sistemului la salt de sarcină

Caz II: Încărcarea bateriilor

Fig. 1-8 prezintă răspunsul la necesitatea sistemului de a încărca bateriile și se caracterizează prin: puterea generatorului solar 1kW, Sarcină constantă 3.5kW, puterea generatorului geotermal 0.75kW, puterea generatorului cu biomasă 1.5kW. La 0.7s sistemul de management a energiei decide că bateriile au nevoie de încărcare și coboară referința de frecvență la 49.8 Hz. Ca rezultat generatoarele geotermal și cu biomasă își cresc puterea la 0.8kW și respectiv 1.75kW corespunzător cu caracteristicile proprii de droop astfel încât surplusul de energie să fie absorbit de baterii. Această stare va fi menținută până bateriile sunt încărcate.

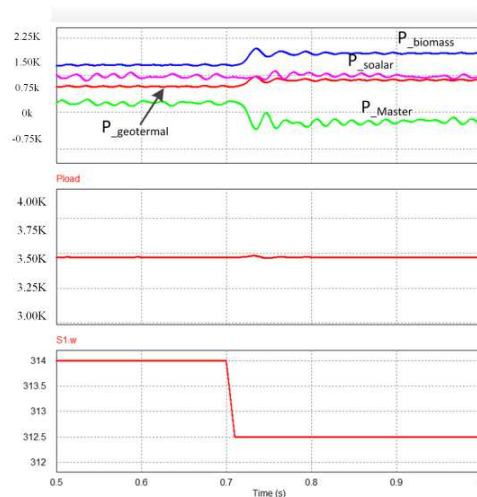


Fig. 1-8 Răspunsul sistemului la încărcarea bateriilor

Caz III: Lipsa energiei solare

Fig. 1-9 prezintă scenariul în care panourile sunt umbrite și generatorul fotovoltaic nu furnizează energie. La 0.5s puterea generatorului solar scade de la 1kW la 0. Ca rezultat inverterul master va furniza diferența de putere necesară până când sistemul de management va coborî referința de frecvență astfel încât puterea de la generatoarele geotermale și cu biomasă să crească.

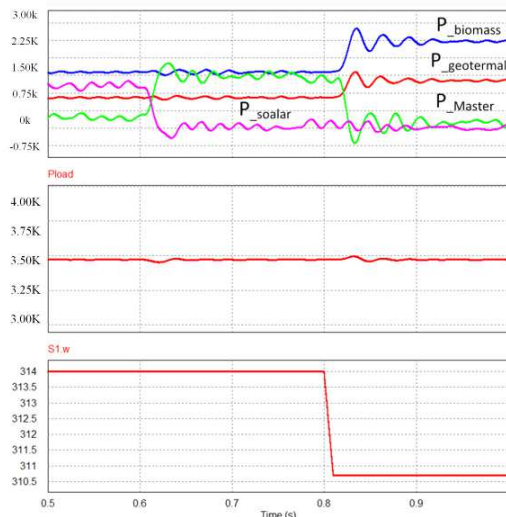


Fig. 1-9

2 Elaborarea algoritmilor de management și de comunicație ai sistemului bazați pe controlul centralizat, controlul predictiv și control tensiune frecvență (droop control)

2.1 Algoritmi de comunicație

2.1.1 Interfața de comunicație și control inverteare SMA

Scopul principal al acestui proiect este integrarea de energii regenerabile în cogenerarea energiei electrice într-o micrețea independentă de rețeaua electrică națională. Pentru obținerea energiei electrice la un preț rentabil și pentru a minimiza consumul de resurse care necesită un proces mai amplu și mai costisitor de producere, este nevoie de un management inteligent al rețelei. Acest management trebuie să țină cont de următoarele aspecte: să permită diferite topologii ale micrețelei, să asigure puterea necesară proceselor critice chiar și în cele mai defavorabile cazuri, să poată asigura energie rețelei la un cost redus, să minimizeze emisiile de CO₂, să asigure elemente de siguranță și să permită un algoritm predictiv bazat pe fenomenele meteorologice și date statistice ale micrețelei.

Pentru implementarea și testarea algoritmului de management este necesară prelevarea datelor referitoare la parametrii rețelei de la echipamentele utilizate. Pentru acest scop ar putea fi utilizat un logger de date dar în cadrul proiectului se dorește obținerea unui algoritmul inteligent, independent care să ofere pe lângă centralizarea datelor și prelucrarea statistică a datelor și un acces la parametrii de control ai echipamentelor. Prin implementarea unui program de management pe o platformă Windows sau Linux se poate obține o interfață utilizator pentru afișarea datelor și accesul la anumite valori prescrise, managementul bazându-se pe un algoritm proprietar.

2.1.2 Introducere

Prezentul capitol explică modul de implementare și funcționare al programului de comunicație și control al invertoarelor SMA conectate pe rețeaua RS485. Programul utilizează biblioteca YASDI implementată de firma SMA și un convertor RS485-USB. Cele două invertoare

sunt echipate cu module de comunicație RS-485 și folosesc un protocol de transfer al datelor derivat din Modbus. Biblioteca YASDI este realizată în Visual C 2005 și conține funcții pentru realizarea unui master capabil să recepționeze date și să transmită comenzi către dispozitivele din rețea pe baza unei adrese.

2.1.3 Schema de conectare a invertoarelor

În Fig. 2-1 este prezentată schema de conectare a invertoarelor.

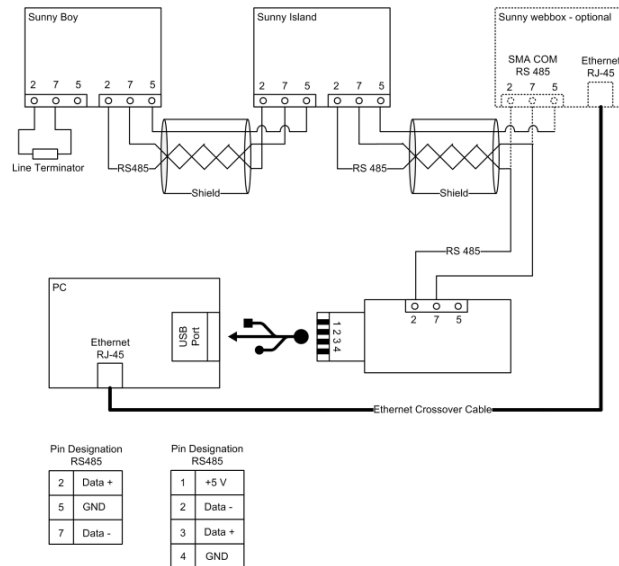


Fig. 2-1 Schema de conectare a invertoarelor

După cum se poate observa în Fig. 2-1 există două moduri de a accesa prin intermediul unui PC datele transmise de invertoarele SMA. O modalitate implică utilizarea unui dispozitiv dedicat de achiziție de date care are implementate ambele protocoale de date marca SMA, Sunny Net și SMA Net. Acest dispozitiv poate fi conectat cu invertoarele pe o linie 485 și oferă accesul datelor fie pe Ethernet către un PC fie direct pe web prin intermediul unui router. O altă posibilitate este de a salva datele recepționate pe un card de memorie care poate fi atașat dispozitivului. Sunny Webbox oferă avantajul de a avea deja implementat protocolul de date, putând fi pus în stare de funcționare într-un timp scurt. Modul standard de funcționare oferă o pagină web prin intermediul căreia se poate accesa rețeaua formată. Dezavantajul major al acestui dispozitiv este că nu permite accesul la parametrii invertoarelor decât prin intermediul interfeței web. Ținând cont că în cadrul proiectului se dorește implementarea unui algoritm de management, aceasta abordare nu este corectă. Managementul rețelei implică accesul la parametrii de control ai invertoarelor automat pentru a aplica deciziile luate de algoritm. Prin urmare s-a apelat la o metodă de comunicație bidirecțională a datelor pe magistrala de date RS485 între un computer (PC) pe post de master și restul componentelor aflate în rețeaua de comunicație (Invertor Sunny Island (SI), invertor Sunny Boy (SB), powermeter, convertor digital-analogic pe Modbus).

În Fig. 2-1 este reprezentată rețeaua de comunicație care conține doar echipamentele de la firma SMA. Aceste echipamente folosesc unul din cele două protocoale proprietate SMA mai sus menționate.

Prin intermediul software-ului Yasdi se poate realiza o aplicație Windows care permite citirea datelor de la echipamente cât și setarea unor parametri. Aplicația este realizată în așa fel încât poate fi portată către diferite platforme (sisteme de operare). În cazul în care este utilizată pentru o platformă Windows fișierele sunt executate ca niște librării (DLL). Aplicația platformă este realizată în principal pentru a implementa funcționarea la nivel de master a protocolului de date SMA. Aplicația este implementată respectând straturile OSI pentru protocoale de rețea (librăriile sunt grupate pe fiecare strat în parte).

Stratul 1: Stratul fizic responsabil cu transmiterea semnalelor individuale pe canalul fizic. În aplicația Windows pot fi controlate driverele de interfață serială RS232 sau RS485 corespunzătoare porturilor (Com1 -Com8).

Stratul 2: Dat fiind faptul că echipamentele SMA comunică fie prin protocolul SMA Net, fie prin Sunny Net este nevoie de acest strat pentru a determina șablonul corect și pentru a calcula CRC-ul.

Stratul 3: Se ocupă cu implementarea protocolului SMA Data. Prin intermediul acestui strat pachetul de date este rutat și transmis către interfața corectă pentru un anumit echipament (abstractizat din punct de vedere al unor comenzi specifice sistemului de operare – ușor de adaptat pentru alte platforme țintă).

Straturile 5/6: În aceste straturi este implementat un master de date SMA capabil să detecteze dispozitivele SMA și să interogheze valorile canalelor. Pe lângă citirea valorilor măsurate oferă opțiunea de a citi sau scrie parametri. Lista canalelor dispozitivului cu care se comunică este stocată în memorie. Prin intermediul acestor straturi o structură a dispozitivului este generată în corespondență cu dispozitivele SMA detectate. Identificatoare (handle-uri) sunt generate pentru dispozitive și canale.

Stratul 7: Descrie interfața externă a masterului SMA. Prin intermediul acestei interfețe masterul poate fi controlat complet.

2.1.4 Funcțiile YASDI

Funcția de inițializare „YasdiMasterInitialize” trebuie executată pentru a putea utiliza oricare altă funcție din cadrul librăriei. Parametrii de apelare a funcției sunt:

„CinifileName”- este un fișier de inițializare gen „Yasdi.ini: care trebuie să se găsească în directorul curent.

„pDriverCount” - este un pointer care indică o variabilă ce returnează numărul de drivere active ale interfeței.

Fișierul „Yasdi.ini” pentru Windows poate avea următoarea structură:

```
[DriverModules]
Driver0=yasdi_drv_serial
Driver1=yasdi_drv_ip

# Configs for serial port 1
[COM1]
Device=COM1
Media=RS485
Baudrate=1200
Protocol=SMANet

# Configs for communication over Ethernet/UDP
# Replace 127.0.0.1 with the real IP address of your device
[IP1]
Protocol=SMANet
Device0=127.0.0.1

[Misc]
#DebugOutput=stderr
```

[Driver Modules]- trebuie activat driverul pentru **serial** necesar stratului (1) fizic. Configurarea portului COM se face în funcție de care Com Port va fi utilizat (COM1-COM8), tipul de transmisie serială (RS485, RS232), rata de transfer și tipul de protocol SMA utilizat (SMANet sau SunnyNet).

Funcția „YasdiMasterShutdown”- eliberează spațiul alocat pentru librăria Yasdi Master.

„YasdiReset”- resetează complet software-ul. Orice dispozitiv detectat va fi eliminat.

Funcții pentru citirea canalelor de măsură:

„GetDeviceHandles” se alocă câte un handle pentru fiecare dispozitiv SMA detectat.

„GetDevicename”- returnează numele dispozitivului și numărul de serie conținut de un handle.

„GetDeviceSN”- returnează numărul serial al dispozitivului indicat prin handle.

„GetDeviceType”- returnează tipul dispozitivului referit prin handle.

„[GetChannelHandle](#)”- se determină toate canalele dispozitivului. Fiecărui canal o sa îi fie alocat un handle existând posibilitatea de a transmite o mască pentru a selecta doar canalele de un anumit tip.

„[FindChannelName](#)”- se poate căuta un anumit canal pe baza numelui pe care îl are alocat. Funcția returnează handle-ul primului canal care corespunde cu numele indicat.

„[GetChannelName](#)”- returnează numele unui canal indicat prin handle-ul său.

„[GetChannelValue](#)”- returnează valoarea unui canal indicat prin handle-ul său. Modul de lucru al funcției este sincron, aceasta așteptând până când o valoare este returnată sau un mesaj de eroare generat. Valorile care se regăsesc în software pentru un anumit canal pot să fie actualizate automat dacă se specifică un timp maxim acceptat. În cazul în care este setată valoarea zero pentru timpul maxim acceptat atunci va fi transmisă valoarea curentă a canalului pe care dorim să îl citim.

„[GetChannelValueTimeStamp](#)”- returnează indexul în timp al valorii canalului referit prin handle.

„[GetChannelUnit](#)”- returnează unitatea de măsură a unui canal în formă de șir de caractere.

„[GetMasterStateIndex](#)”- returnează starea curentă a masterului YASDI:

„1” - starea inițială a sistemului;

„2” - Stare de detectare a dispozitivelor;

„3” - Configurarea adreselor de rețea;

„4” - Cerere a liste de canale;

„5” - Procesare a comenzilor de către modulul master;

„6” - Citirea valorilor canalelor;

„7” - Setarea (scrierea) unor valori pentru canale.

„[SetChannelValue](#)”- setează valoarea numerică a unui canal referit prin handle.

„[GetChannelStateTextCnt](#)”- returnează numărul de șiruri de caractere existente pentru un canal.

„[GetChannelStatText](#)”- returnează un text care indică starea canalului. Indexul textului la care se face referire trebuie transmis. Înainte de apelarea acestei funcții este indicat să fie folosită funcția „[GetChannelStateTextCnt](#)”.

„[GetChannelMask](#)”- returnează masca unui canal și indexul acestuia. Canalul este referit prin handle-ul său.

„[DoMasterCmdEx](#)”- permite transmiterea de comenzi către masterul Yasdi. Singura comandă existentă este „detection” care permite detectarea unui număr de dispozitive specificat.

Pe lângă funcțiile mai sus prezentate care se referă la modul de implementare a unei aplicații Yasdi master există funcțiile care permit accesul la nivel jos al protocolului de date SMA implementat. Aceste funcții pot fi folosite pentru a controla driverele interfețelor de comunicație cât și pentru a utiliza comenzile SMA standard.

Funcții utilizate pentru controlul driverele interfeței de transmisie a datelor:

„[yasdiGetDriver](#)”- returnează interfețele disponibile. Acestea pot fi accesate ulterior pe baza unui identificator. Fișierul „Yasdi.in” indică tipul interfețelor care sunt disponibile.

„[yasdiGetDriverName](#)”- returnează numele unei interfețe referită prin identificatorul ei.

„[YasdiSetDriverOnline](#)”- activează interfața referită prin indicatorul ei. În urma activării interfața poate fi utilizată în cadrul aplicației.

„[yasdiSetDriverOffline](#)”- interfața referită este dezactivată nemaiputând fi utilizată în cadrul aplicației.

2.1.5 Funcții utilizate pentru transferul pachetelor de date SMA

În Fig. 2-2 este prezentată organigrama ce trebuie urmată pentru a iniția comunicația între un dispozitiv master ce utilizează aplicația YASDI și un dispozitiv slave ce are implementat unul din protocoalele proprietate SMA.

conține o structură repetitivă care așteaptă apăsarea unei taste corespunzătoare unei comenzi și o structură de tip case care va apela funcțiile corespunzătoare comenzii introduse.

Lista comenzilor existente la momentul întocmirii acestui document este prezentată mai jos.

```
printf("Comenzi posibile:\n");
printf(" ? : Lista comenzi\n");
printf(" e : Start detectare sincrona echipamente SMA\n");
printf(" b : Start detectare asincrona echipamente SMA\n");
printf(" z : Opreire detectare asincrona echipamente SMA\n");
printf(" d : Afisare lista dispozitive SMA detectate\n");
printf(" a : Afisare lista canale 'spot' SI sau SB\n");
printf(" p : Afisare lista canale parametri SI sau SB\n");
printf(" t : Afisare lista canale de test SI sau SB\n");
printf(" v : Citire valoare canal SI sau SB\n");
printf(" s : Setare valoare parametru SI sau SB \n");
printf(" c : Citire continua valori tensiune, putere generator electric,
putere SI si putere SB pana la apasarea unei taste\n");
printf(" f : Citire tensiune si putere generator electric\n");
printf(" k : Setare frecventa inverter motor (8500->50.8Hz)\n");
printf(" o : secventa setare frecventa inverter motor (8500->50.8Hz) si
citire valori (putere si tensiune generator electric, putere SI si putere
SB)\n");
printf(" r : resetare date SMA\n");
printf(" m : Eliminare echipamente SMA\n");
printf(" x : Opreire comunicatie SMA\n");
printf(" q : Iesire program\n");
```

Fiecare comandă va fi explicată în parte cât și funcțiile utilizate pentru a putea fi realizată. Pentru o funcționare corectă a aplicației trebuie urmărită orga nigră ma următoare :

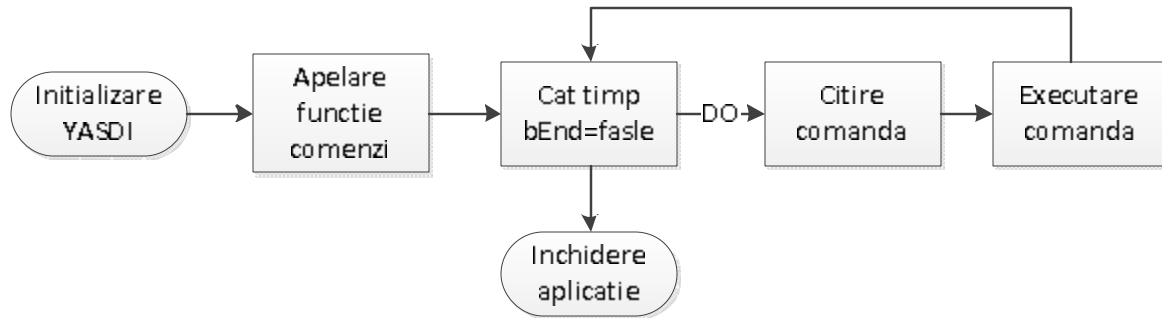


Fig. 2-3. Funcționarea aplicației C

Comanda de detectare a echipamentelor SMA indiferent de metoda utilizată, sincronă sau asincronă, trebuie executată de fiecare dată când este pornit sistemul de comunicație. Restul funcțiilor care comunică cu echipamente SMA vor returna rezultate corecte doar dacă dispozitivele au fost ulterior descoperite și informațiile despre ele salvate.

Secvența de cod pentru *comanda „e”* referitoare la detecția sincronă a dispozitivelor va conține în primul rând activarea driverului pentru portul de comunicație configurat conform fișierului yasdi.ini. Ultima funcție care va fi apelată în cadrul secvenței de cod care deservește *comanda „e”* va fi cea de eliberare a resurselor portului de comunicație, pentru ca această să poate fi utilizat și de către echipamentele Modbus standard în cazul în care se dorește acest lucru.

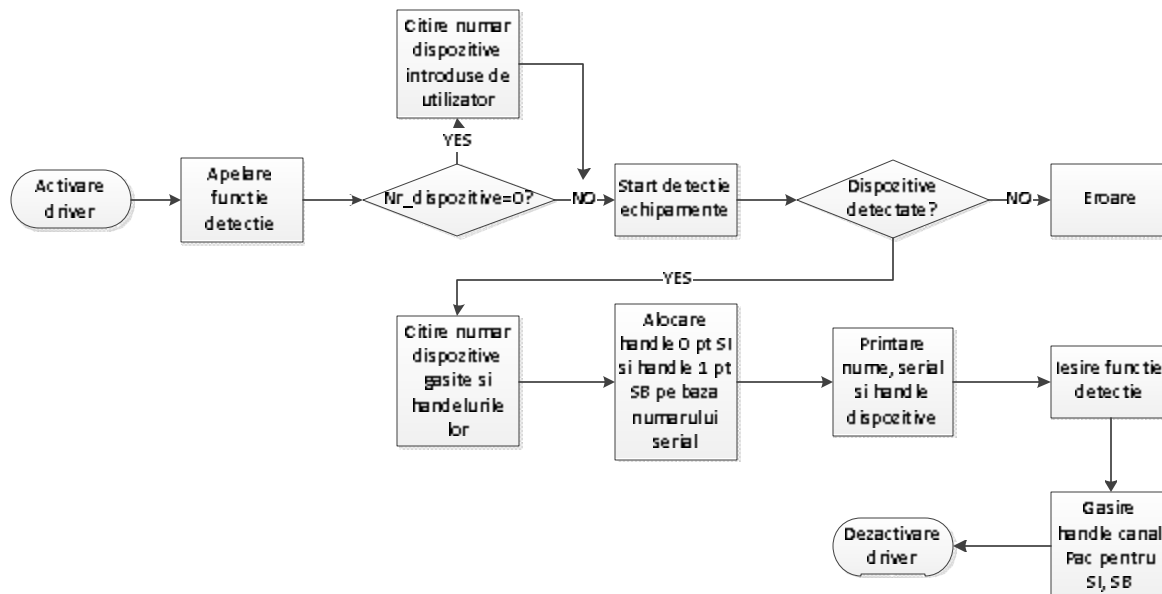


Fig. 2-4 Funcționarea aplicației

Comanda „b” cea care deservește pornirea unei detecții asincrone respectă același tipar ca și detecția sincronă doar că funcția va rămâne blocată până la identificare dispozitivelor sau transmiterea unei comenzi de oprire a detecției asincrone. În cadrul programului de comunicație este indicat să fie utilizată funcția de detecție sincronă.

Comanda „d” permite afișarea dispozitivelor SMA detectate. Va fi afișată denumirea dispozitivului, numărul serial cât și handlel dispozitivului.

În cazul în care au fost identificate dispozitive SMA se pot interoga canalele acestora cu una din comenzile „a”, „p” sau „t”. Aceste comenzi au aceeași structură și apelează aceeași funcție „PrintChannelValues” având ca parametri „Testchannels”, „Spotchannels” sau „Paramchannels” în funcție de informația dorită și numărul handleului pentru care se interoghează canalele. În cazul aplicației curente la momentul redactării documentului există doar două dispozitive SMA iar handleurile pentru acestea au fost setate după cum urmează. **0 - inverter SI** și **1 - inverter SB**. Utilizatorul trebuie să introducă dispozitivul pentru care se dorește interogarea canalelor.

În cazul în care se dorește citirea unui canal individual din lista de date primite de la un dispozitiv SMA se va utiliza *comanda „v”*. Aceasta apelează funcția „ReadChannelValue” având ca parametru transmis handleul dispozitivului. Funcția „ReadChannelValue” poate fi utilizată pentru a citi valoarea oricărui canal al dispozitivului prin apelarea funcției „FindChannelName” pentru a determina handleul unui canal a nume al cărui denumire a fost introdusă de utilizator. În implementarea curentă funcția va returna direct valoarea canalului care se referă la valoarea puterii active „Pac” fie a invertorului SI sau SB.

Comanda „s” permite setarea unui parametru al dispozitivelor SMA. Prin selectarea acestei comenzi se apelează funcția „SetParamValue”. Utilizatorul este interogat asupra parametrului ce urmează să fie modificat. Acesta este căutat în lista de parametri iar valoarea care urmează să fie transmisă către dispozitiv este introdusă de utilizator.

Comanda „c” permite citirea continuă a următoarelor valori până la apăsarea unei taste: tensiune rețea, putere generator electric (valoarea recepționată de la power meter), putere activă SI și putere activă SB.

În ciclul repetitiv se apelează funcția „nemo” care permite citirea datelor pe Modbus standard. Adresa unde se găsește valoarea tensiunii monitorizate de power meter este 4096 iar cea a puterii active este 4140. Funcția „nemo” va returna valoarea canalului interogat. În corpul funcției trebuie setat portul de comunicație serială pentru a transmite datele cu viteza de transfer stabilită pentru dispozitivele modbus standard. Pentru setarea portului serial s-a utilizat funcția *CreatFile* având ca parametri portul fizic și modul în care pot fi accesate datele (scriere și citire). Viteza de transfer selectată pentru dispozitivele modbus standard este de 38400, numărul de biți de date 8, fără bit de paritate și un singur bit de stop. În urma setării canalului de comunicație se poate popula cu valori structura de date

care se referă la comunicația în modul modbus standard. Pachetul de date utilizator se populează prin funcția „CreateNemoData” având ca parametri transmiși un pointer, adresa canalului care urmează să fie interogată și numărul de bytes. Structura de date pachet va conține adresa dispozitivului, comanda de citire, adresa ce urmează să fie citită și numărul de bytes. Toți octeții ce urmează să fie transmiși sunt salvați într-un vector de date cu rol de buffer de date intermediar respectând ordinea acestora conform standardului modbus standard și calcularea cuvântului de control pe baza datelor din pachetul de date. Prin apelarea funcției „WriteModbus” se va transmite pe RS485 șirul de octeție corespunzător comenzii curente utilizând funcția „WriteFile”. În urma unei comenzi de citire a datelor dispozitivelor modbus standard se apelează funcția „ReadModbus” care va recepționa utilizând „ReadFile” datele de pe magistrala de date RS485. Datele vor fi salvate într-un buffer de date putând fi utilizate ulterior. În urma unei comunicații cu dispozitivele Modbus standard se va închide portul de comunicație, acesta fiind disponibil pentru a fi setat fie în modul standard fie în modul de comunicație SMA.

În cadrul comenzii „c” după eliberarea portului de comunicație acesta va fi setat pentru a putea fi transmise comenzile de citire a puterilor active ale invertoarelor SI și SB.

Comanda „f” este foarte asemănătoare cu cea anterior descrisă doar că se va interoga doar powermeterul fiind afișate valoare tensiunii și a puterii active.

Comanda „k” setează frecvența invertoarelor ce controlează ansamblul motor-generator electric. Prin intermediul comenzii se transmite mărimea prescrisă modului convertor valoare modbus – mărime analogică. Adresa dispozitivului este 4 iar valoarea introdusă de utilizator respectă relația :

$$marime_prescrisa = \frac{8500 \cdot val_frecventa}{50.8} \quad (1)$$

Marime_prescrisa se referă la valoarea care va fi transmisă de către modulul de conversie către invertoarele motorului iar val_frecvență este mărimea frecvenței dorite.

Funcția „writenemo” este asemănătoare cu funcția utilizată pentru comanda „c”. Prin intermediul acestei funcții se deschide portul de comunicație pentru transmiterea datelor utilizând protocolul modbus standard și se populează structura de date și bufferul de transmitere cu valorile corespunzătoare (adresă, număr de bytes, datele ce urmează să fie transmise etc).

Comanda „o” include cele două comenzi „k” și „f”. În prima fază este setată frecvența invertoarelor după care sunt citite tensiunea și puterea activă a ansamblului motor-generator.

Alte comenzi implementate sunt „r” care apelează funcția de resetare a datelor referitoare la rețeaua SMA, „m” care permite eliminarea din program a echipamentelor SMA, „x” închiderea modului de comunicație SMA și „q” ieșirea din program.

Pentru o funcționare corectă a comenzilor mai sus enumerate nivelul de securitate active trebuie să fie cel de instalator și nu cel de utilizator standard. Nivelul de securitate în cazul implementării curente este setat ca instalator. În funcția ysecurity.c lui „currLevel” îi este atribuită valoarea LEV_2.

Programul prezentat până în acest punct se poate utiliza ca aplicație Windows fiind un program executabil. Pentru utilizarea cu interfața de management implementată în C# au fost aduse modificări acestui program. În cele ce urmează sunt descrise diferențele între implementarea pentru a obține un program executabil și programul pentru interfață.

2.1.7 Comunicarea între interfața C# și programul de comunicație

Comunicarea între interfața și programul de comunicație se realizează pe baza de fișiere text. Astfel interfața va scrie într-un fișier comandă ce urmează să fie executată. Pe lângă comanda și valorile care urmează să fie setate este transmisă și amprenta de timp. Această amprentă este utilizată pentru a nu executa o comandă de mai multe ori decât dacă acest lucru este dorit de către utilizator. În figura următoare este prezentată organigrama buclei repetitive care permite selectarea unei comenzi.

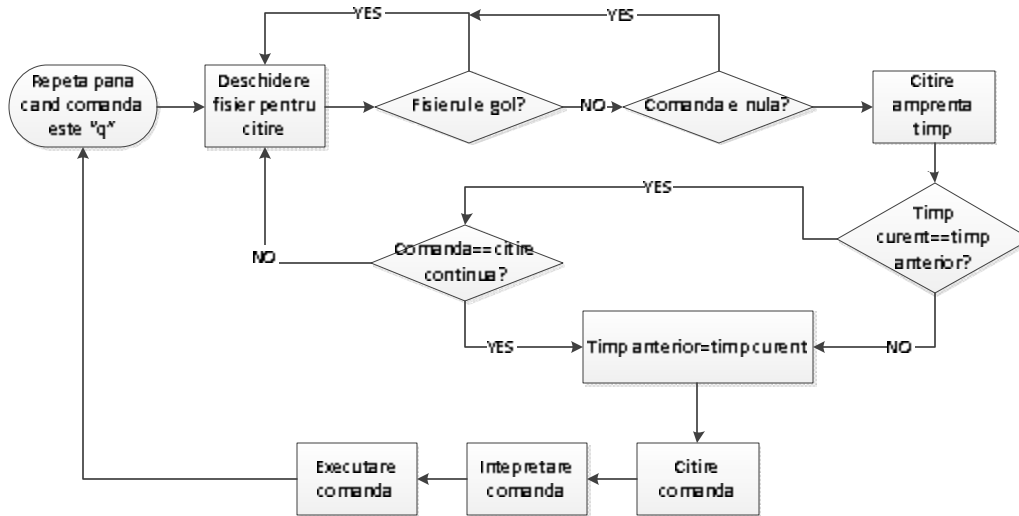


Fig. 2-5. Selectarea unei comenzi

Comenzile care sunt implementate sunt aceleași ca și în cazul utilizării programului fără interfață utilizator. Datele care vor fi vehiculate în fișierele text sunt prezentate în paragraful următor.

Comanda transmisă prin intermediul fișierului va fi compusă din 11 caractere plus amprenta de timp. Formatul unei astfel de comenzi este următorul:

Byte											
1	2	3	4	5	6	7	8	9	10	11	12
Comandă	Număr dispozitive	Selectare inverter (SI sau SB)	Nume parametru (canal)				Valoare				Amprentă timp

Tabelul 26. Formatul unei comenzi – fișier text

Comanda este parametrul care trebuie transmis de fiecare dată și respectă meniul explicat în cazul implementării fără interfață utilizator.

Byte-ul 2 cel care se referă la numărul de dispozitive SMA este utilizat doar pentru comenzile „e” și „b” putând fi în rest neutilizat adică poziția sa să ia valoarea 0. Cel de-al treilea byte se referă la comenzile care transmit date în rețeaua SMA și trebuie identificat cărui inverter îi vor fi transmise datele (SB sau SI). Comenzile care necesită interpretarea acestui caracter sunt: „t”, „a”, „p”, „s”, „v” și „c”. Caracterele 4-7 sunt folosite pentru a transmite numele parametrului care urmează a fi citit sau setat când ne referim la dispozitivele SMA. Comenzile care au nevoie de o valoare diferită de zero pe aceste poziții sunt: „s”, „v”, „o”. Caracterele de la 8-11 sunt utilizate pentru a transmite valoarea care trebuie scrisă la adresa unui anumit canal referit prin caracterele 4-7. Ultimele caractere din fișier o să se refere la amprenta de timp necesară pentru a identifica o nouă comandă.

2.1.8 Interfața C#

Cum a fost menționat anterior, comunicarea între interfața și programul de comunicație se realizează pe baza de fișiere text:

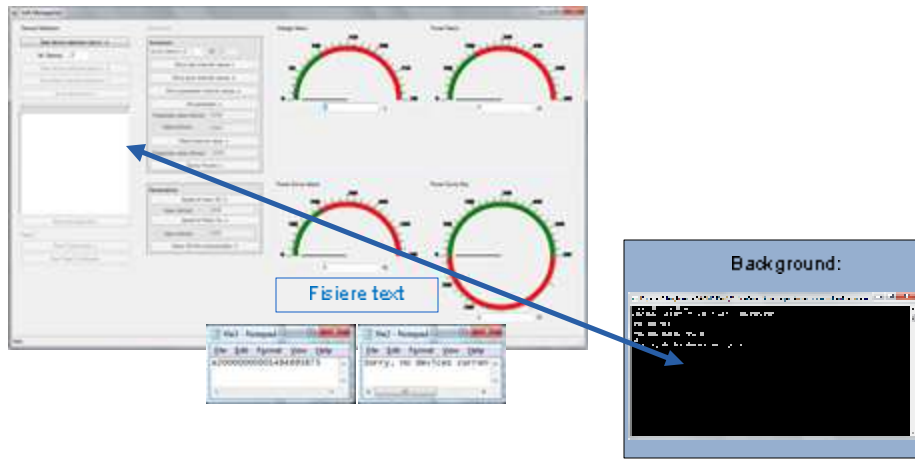


Fig. 2-6 Comunicarea între interfața și programul de comunicație

Interfața C# este prezentată în Fig. 2-6.

Apăsând butonul "Start Device detection (sync): e", textul următor va fi scris în fișierul text: "e2000000001464686448". Comanda "e" este activată, aplicația încearcă să identifice 2 dispozitive. Dacă identificarea a fost realizată cu succes și au fost identificate ambele dispozitive, toate butoanele vor fi activate. Comenzile butoanelor sunt aceleași ca și în cazul utilizării programului fără interfață utilizator. Răspunsurile primite de la SI, SB și powermeterele sunt afișate în căsuța text.



Fig. 2-7. Interfața C#

În cazul butonului "Sunny Powers: c" tensiune rețelei, puterea generatorului electric (valoare recepționată de la power meter), putere activă SI și putere activă SB sunt prezentate cu ajutorul a 4 display-uri. Butonul "Read Continuously: c" afișează valorile și în textboxul de pe partea stângă. Acest buton dătește și scrie valorile la un interval de 2sec.



Fig. 2-8. Interfața C# - Citire în mod continuu

Pe lângă cele două implementări utilizând o parte din funcțiile YASDI s-a mai realizat un program care pornește la comenzile SMA și Modbus standard. Această implementare este independentă de funcțiile YASDI și permite transferul funcțiilor pe un alt sistem care nu rulează sisteme de operare Microsoft. Singura problemă în cazul acestei implementări este aceea că trebuie cunoscută poziția canalelor care urmează a fi interogate sau scrise în fluxul de date recepționat de la echipamentele SMA.

Implementarea cu funcții YASDI este prezentată în anexa 1, cea care permite integrarea cu aplicația grafică în anexa 2 iar cea care nu utilizează funcțiile YASDI în anexa 3.

2.2 Algoritmi de management al energiei – Stadiu actual

Scopul proiectului este de a oferi o soluție inovatoare și completă de management a energiei care monitorizează consumul, generează cantitatea necesară de energie și încarcă/descarcă bateriile într-o micronețea. Există mai multe tipuri de metode de management energetic. Au fost studiate: Algoritmul Genetic (Eng: Genetic Algorithm - GA), Optimizare Roi de Particule (Eng: Particle Swarm Optimization - PSO) și Programare Liniară (Eng: Mixed Integer Linear Programming - MILP).

2.2.1 Algoritmul Genetic

Unul dintre cei mai cunoscuți algoritmi de calcul evoluționist este Algoritmul Genetic. Acest algoritm este inspirat de teoria lui Charles Darwin asupra evoluției – "Supraviețuiește cel care este cel mai bine adaptat.". Ideea calculului evoluționist prima dată a fost aplicată de I. Rechenberg în anul 1960 în lucrarea "Evolution Strategies", iar metoda însăși a devenit cunoscută începând cu anii '75 prin lucrările lui John Holland.

În spațiul soluțiilor potențiale trebuie să fie efectuată o căutare care să dea ca și rezultat soluția optimă. De aceea procesul de rezolvare poate fi privit ca și un proces de optimizare. Metodele clasice exhaustive sunt folosite în probleme mai simple cu spațiile soluțiilor mici, iar pentru rezolvarea problemelor complexe cu spațiul soluțiilor potențiale mai mari trebuie să fie folosite tehnicile speciale ale inteligenței artificiale. Algoritmii genetici sunt aplicați la problemele complexe care au nevoie de o optimizare globală.

Acești algoritmi sunt proceduri adaptive care au la bază mecanismele de selecție naturală și evoluție genetică. Soluțiile posibile la o problemă sunt codificate în structuri de date, numite **cromozomi**. Peste aceste structuri sunt aplicați diferiți operatori genetici ca să mențină informații critice.

Noțiunile de bază ale algoritmului sunt prezentate în Tabelul 3-1.

Natura		Algoritmul genetic
Mediu	□	Problemă
Populație	↔	Mulțimi de soluții potențiale
Individ	↔	Soluție potențială
Cromozom	↔	Codarea unei soluții potențiale
Fitness	↔	Funcție de calitate
Selecție, încrucișare, mutație	↔	Operatori genetici

Tabelul 4-1. Noțiunile de bază

Algoritmul Genetic construiește mulțimea inițială a soluțiilor (în general aleator), numită în literatură: populația inițială. Această mulțime a soluțiilor conține cromozomi și fiecare individ (cromozom) reprezintă o soluție potențială a problemei. Pe durata iterațiilor succesive, numite generații, cromozomii evoluează în mod continuu. Folosind unele măsuri de potrivire, numite funcții **fitness**, cromozomii sunt evaluați în fiecare generație. Din populația curentă (populație părinte) sunt **selecți** cromozomii cei mai buni și este creată populația nouă (populație copii) utilizând operatori genetici: **încrucișare (crossover) și mutație**.

Selecția constă în operatorul genetic, care asigură că cromozomii cu grad de măsuri de potrivire mai ridicate au o probabilitate mai mare la formarea generației noi. Încrucișarea este un alt operator genetic care asigură procesul prin care pe baza a doi cromozomi din populația curentă sunt formați doi cromozomi pentru populația următoare. Prin mutația cromozomul din generația curentă este modificat, și apoi salvat în populație nouă.

Organigrama de funcționare a algoritmului genetic este prezentată în Fig. 2-9.

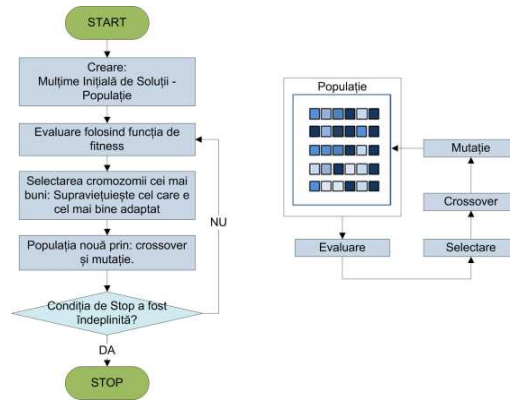


Fig. 2-9 Algoritmul Genetic

Algoritmul Genetic a fost folosit cu succes în problemele de management energetic. Pedro Vergara caută costul de funcționare optimă a unei microrețele folosind acest algoritm. Ținta lui este de a genera momentele de start/stop ale generatorului diesel, ale panourilor fotovoltaice și turbinei de vânt, pentru a evalua valoarea de încărcare/descărcare a bateriilor și conectarea la rețeaua națională, având în vedere costul de funcționare să fie minim.

2.2.2 Algoritm de management folosind optimizare de tip roi de particule

Algoritmii evolutivi simulează evoluția naturii, iar algoritmii inspirați de comportamentul de grup simulează adaptarea colectivă și procesele sociale dintr-un colectiv. Se numește inteligența de grup o populație de indivizi care interacționează pentru atingerea unor obiective prin adaptarea colectivă la un mediu local sau global. Algoritmul a fost inspirat de zborul păsărilor, bancurile de pești și de furnicile care caută mâncare, deoarece controlul este distribuit, comunicarea dintre indivizi se realizează local, sistemul este robust și se poate adapta schimbărilor de mediu.

PSO este un algoritm multi-obiectiv de optimizare inspirat de comportamentul social al stolurilor de păsări și a bancurilor de pești. Se bazează pe modul în care păsările și peștii caută hrana. Algoritmul a fost propus de Kennedy și Eberhart în anul 1995.

La acest algoritm, populația și indivizii sunt numiți roi (swarm) și particule. Fiecare particulă are o vecinătate de particule și se deplasează în spațiul de căutare cu o viteză, urmărind cele mai performante particule. Particula păstrează poziția unde a obținut rezultatul cel mai bun ("personal best") în istoria ei. Liderul este particula care stă la poziția cea mai bună (cel mai aproape la mâncare). Particulele cooperează, schimbă informațiile între ele despre descoperirile făcute. Fiecare particulă încearcă să se apropie de lider. Pentru a realiza acest lucru parametri săi trebuie să fie modificați. Schimbarea se întâmplă cunoscând cea mai bună poziție (în istoria ei) și poziția liderului (în momentul de față). După schimbare, particula primește o poziție nouă. După ce fiecare particulă a fost evaluată și a primit poziția nouă, va fi ales un nou lider.

Algoritmul are la bază două comportamente: comportamentul cognitiv și social. Comportamentul social înseamnă că particulele au memorie, adică ele își amintesc de cunoștințele acumulate în trecut. Comportamentul social reprezintă faptul că fiecare individ ia în considerare și cunoștințele celorlalți membri ai roiului.

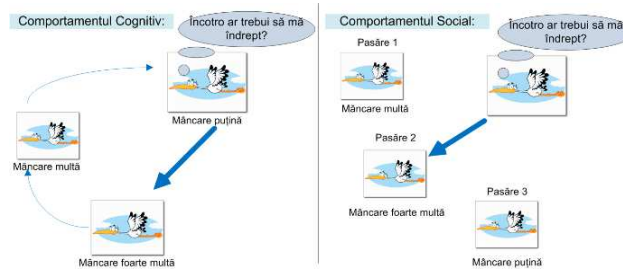


Fig. 2-10 Comportamentul cognitiv și social

Funcționarea algoritmului prezentat devine mai clar studiind organigrama în Fig. 2-11.

Algoritmul se poate aplica cu succes în problemele de management al energiei. Se caută costul optim de funcționare a unei microrețele folosind PSO. Microrețeaua propusă de G. R. Anghajani include panourile fotovoltaice, turbinele de vânt, micro-turbina, pila de combustie, bateriile și sarcinile.

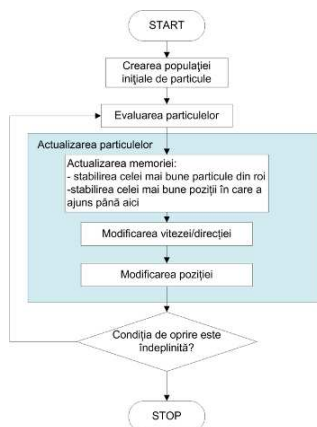


Fig. 2-11 Organigrama PSO

2.2.3 Algoritmi de management folosind programare liniară

Programarea liniară se ocupă de o clasă specială de probleme de optimizare care apar deseori în aplicațiile economice. Aceste probleme constau în maximizarea sau minimizarea unei funcții liniare, numite **funcții obiectiv**, ale cărei variabile trebuie să satisfacă:

- un sistem de relații date sub forma unor ecuații și/sau inecuații liniare nestricte, denumite generic restricții;
- cerința de a lua numai valori numerice nenegative (≥ 0).

După construirea modelului matematic a problemei urmează rezolvarea ei, determinarea soluției optime. Schema generală de construire a unui model matematic pentru o problemă de optimizare din domeniul economic este:

$$\begin{cases} \max(\min)z = \sum_{j=1}^n c_j \times x_j \\ \sum_{j=1}^n a_{ij} \times x_j \leq b_i, 1 \leq i \leq m \\ x_j \geq 0, 1 \leq j \leq n \end{cases} \quad (2)$$

Sistemul se poate scrie în **forma standard**. În acest caz toate restricțiile sunt ecuații și tuturor variabilelor li se impun condiții de nenegativitate:

$$\begin{cases} \max(\min)z = \sum_{j=1}^n c_j \times x_j \\ \sum_{j=1}^n a_{ij} \times x_j = b_i, 1 \leq i \leq m \\ x_j \geq 0, 1 \leq j \leq n \end{cases} \quad (3)$$

unde:

- A este matricea sistemului de restricții
- x este vectorul coloană al necunoscuteilor
- b este vectorul coloană al termenilor liberi

Forma canonică reprezintă faptul că toate restricțiile sunt inegalități în același sens și tuturor variabilelor li se impun condiții de nenegativitate.

$$\begin{cases} \max c^T \times x \\ A \times x \leq b, A \in M_{m,n}(R) \\ x \geq 0, x = (x_1, x_2, \dots, x_n) \end{cases} \quad (4)$$

$$\begin{cases} \min c^T \times x \\ A \times x \geq b, A \in M_{m,n}(R) \\ x \geq 0, x = (x_1, x_2, \dots, x_n) \end{cases} \quad (5)$$

- c este vectorul linie a coeficienților funcției obiectiv

Mulțimea soluțiilor admisibile este:

$$S = \{x_j \in R^n / A \cdot x_j = b, x_j \geq 0\} \quad (6)$$

Se numește soluție optimă acea soluție admisibilă careia îi corespunde o valoare maximă (pentru problemele de maximizare) sau minimă (pentru problemele de minimizare) a funcției obiectiv pe mulțimea soluțiilor admisibile.

De obicei problemele de programare liniară nu se găsesc în formele prezentate. Pentru a le putea rezolva, acestea trebuie rescrise la forma standard.

Există mai multe metode de căutare ale valorii optime dintre care cele mai importante sunt:

- Metoda grafică: se poate aplica doar în cazul când sistemul are două variabile
- Algoritmul simplex: numărul variabilelor necunoscute nu este limitat

Metoda grafică

Funcționarea metodei grafice este prezentată folosind două exemple. Organigrama din Fig. 3.4 prezintă pașii algoritmului.

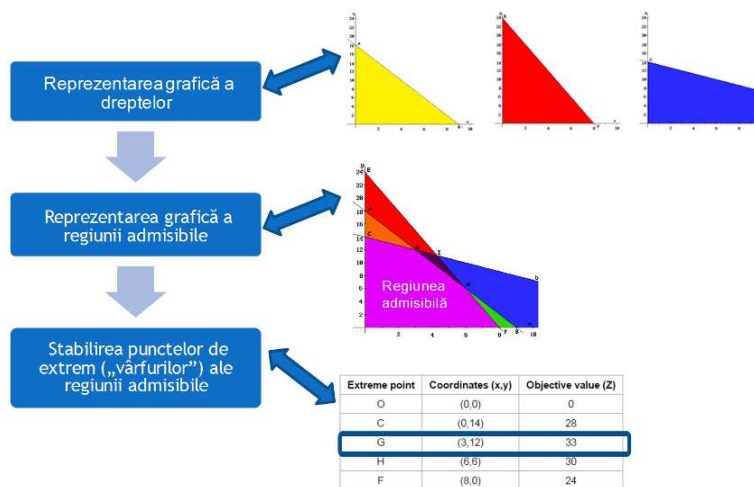


Fig. 2-12 Metoda grafică – funcționare

Algoritm simplex

Algoritm simplex a fost dezvoltat de George Dantzig în 1947. Este o metodă eficientă, aplicată în rezolvarea problemelor de programare liniară. Majoritatea problemelor conțin multe variabile, de aceea metoda grafică nu mai este valabilă. Momentan, metoda lui Dantzig este cea mai importantă metodă în găsirea soluțiilor LP.

Rezolvarea problemei folosind metoda simplex are pașii următori:

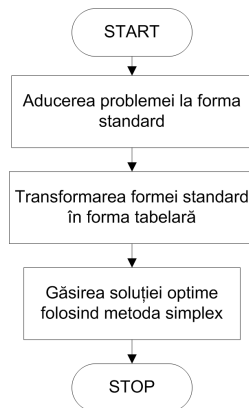


Fig. 2-13 Metoda Simplex

Algoritmul simplex este o metodă iterativă care este aplicată peste forma tabelară a problemei LP. Procedura generează forma nouă a tabelului în fiecare pas și îmbunătățește valoarea funcției obiectiv. Atunci când nu se poate realiza nici o modificare, a fost găsită soluția optimă.

3 Proiectarea și simularea sistemului embedded bazat pe DSP sau microcontroler pentru managementul energiei

Din considerente de cost pentru sistemul de management al energiei s-a optat pentru un mini-sistem de calcul bazat pe procesor Intel și afișarea datelor pe un mini-display LCD cu touchscreen. Prețul unui astfel de sistem este sub 200EUR.

Atât procesoarele de uz general cât și procesoarele de semnal se pot folosi în aplicații embedded. În funcție de puterea de calcul cerută se poate opta pentru GPP sau pentru DSP. Procesoarele de semnal sunt destinate în special pentru aplicații care necesită analize de semnal având unități de calcul care facilitează aceste analize. De asemenea ambele tipuri de procesoare se pot folosi pentru aplicații unde costul este important (low-end) dar și pentru aplicații unde se cer performanțe ridicate (frecvențe de ceas ridicate, paralelism), Fig. 3-1.

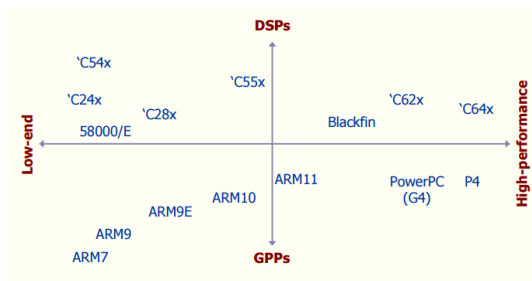


Fig. 3-1 Domeniile de utilizare a GPP și DSP

Tabelul de mai jos prezintă o comparație între un sistem embedded bazat pe DSP și sistemul bazat pe mini-procesorul de uz general. Acest tabel justifică utilizarea unui GPP pentru controlul microrețelei deoarece sistemul nu realizează procesare de semnale în timp real. Algoritmul de management al microrețelei se execută la intervale de timp de ordinul secundelor cerință îndeplinită de ambele sisteme. Utilizarea unui sistem de operare facilitează dezvoltarea algoritmilor de management și asigură atât portabilitatea cât și actualizarea ușoară ulterioară a sistemului. Printre alte avantaje se numără controlul la distanță și monitorizarea de la distanță a microrețelei.

Tabel 3-1 Comparație între un sistem bazat pe DSP și un sistem bazat pe mini-procesor

Criteriu	Sistem embedded bazat pe DSP	Sistem bazat mini-procesor
Pret	--	+++
OS	--	Windows, Linux...

Performanțe	<ul style="list-style-type: none"> - Suportă până la 8 unități aritmetice - unități aritmetice speciale: MAC, Viterbi - suporta tipuri de date multiple - posibilități multiple de manipulare a datelor la nivel de bit - în general 32 bit - capabilitate SIMD (single instruction multiple data) - arhitectura Harvard - folosesc uneori memorie cache (DMA preferată de obicei) - foarte eficiente în utilizarea memoriei - neprietenoase cu compilatoarele (not compiler friendly) - necompatibile cu versiuni anterioare - arhitecturi proprietare - sisteme de dezvoltare dificile - simulație puternică 	<ul style="list-style-type: none"> - 1-3 unități aritmetice - unități aritmetice speciale: Virgulă fixă, virgulă flotantă - suporta tipuri de date multiple - posibilități limitate de manipulare a datelor la nivel de bit - în general 32 bit - capabilitate SIMD (single instruction multiple data) - arhitectura Harvard - folosesc memorie cache - mai puțin eficiente în utilizarea memoriei - prietenoase cu compilatoarele (compiler friendly) - compatibilitatea se asigură pentru majoritatea procesoarelor - arhitecturi partajate - sisteme de dezvoltare user-friendly - simulație în dezvoltare
Utilizare:	DSP	GPP
Procesare de semnal	+	-
Aplicații de control	-	+ (portabilitatea și densitatea codului)
Necesitatea de compatibilitate cu versiuni anterioare	Nu este de obicei nevoie	obligatorie
Vanzători multipli	Nedot	Dorit
Integrare în aplicații	++	++

Sistemul de management al energiei este format din: un minicalculator bazat pe procesor Intel Atom pe care rulează Windows 10, un LCD display de 5 inch cu touchscreen și un convertor USB - RS485, Fig. 3-2.

Caracteristicile necesare pentru calculatorul folosit sunt: procesor Intel Atom X5-Z3800, 2GB memorie RAM, 32GB memorie flash, port HDMI pentru conexiunea afișajului, 2 conexiuni USB (1xUSB3.0 și 1xUSB2.0), comunicație Wireless și Bluetooth. Alimentarea sistemului de management se realizează printr-un conector MicroUSB din microrețea.

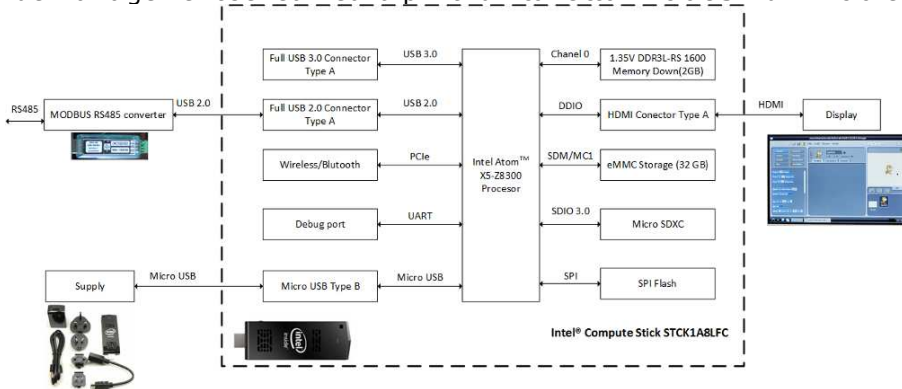


Fig. 3-2 Sistem de management al energiei

4 Proiectarea și simularea programului pentru dimensionarea microrețelei.

Scopul programului de dimensionare a microrețelei este de a determina atât numărul de panouri și de baterii cât și dimensiunile generatoarelor geotermale și cu biomasă ce urmează să fie folosite. Acest capitol prezintă proiectarea programului compatibil Windows PC care are acest rol. Microrețeaua propusă permite integrarea mai multor tipuri de surse regenerabile: energia geotermală, energia solară și biogazul, dar și asigurarea unei rezerve de energie electrică care este stocată într-un sistem hibrid format din baterii de condensatoare și acumulatori de mare capacitate. Sistemul cercetat este de tip off-grid, neconectat la rețeaua electrică națională.

Programul propus are la bază doi algoritmi. Primul algoritm calculează capacitatea generatoarelor folosind valoarea medie anuală a consumului de energie electrică cât și pe baza valorii medii anuale a cantității de energie produse de generatoare. Cel de-al doilea, (LPSP - Probabilitatea pierderii sursei de alimentare), pornind de la rezultatele oferite de primul optimizează capacitatea generatoarelor ținând cont de evoluția datelor meteorologice și a sarcinilor la un interval de o oră, estimate pe o durată de 1 an. Algoritmii LPSP folosesc o subrutină de predicție a irradiației solare din datele meteorologice mediate pe durata unei luni. Necesitatea acestei subrutine derivă din faptul că majoritatea surselor care oferă date meteorologice furnizează doar media lunară a irradiației solare.

Cei doi algoritmi prezentați mai sus au la bază calculele de dimensionare prezentate în capitolele de proiectare a generatoarelor microrețelei.

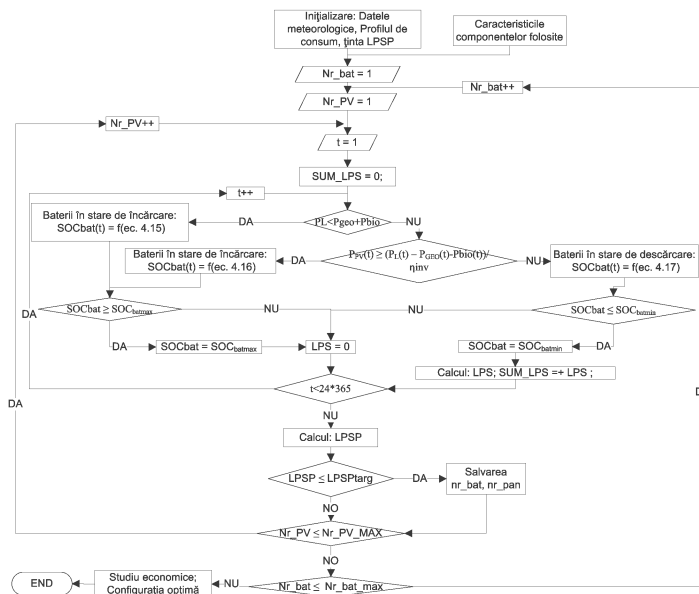


Fig. 4-1 Algoritm LPSP

Funcționarea algoritmului LPSP este reprezentată în organigrama din Fig. 4-1. După setarea profilului de consum și inițializarea procentului LPSP dorit, algoritmul de optimizare va porni.

Valoarea LPSP este calculată pentru o durată de timp de un an. Numărul bateriilor și panourilor este salvat numai dacă valoarea calculată pentru LPSP este mai mare decât o valoare setată de utilizator. La finalul procesului are loc un studiu economic și este afișată configurația optimă a sistemului.

4.1 Predicția irradiației solare

Datele aferente irradiației solare se găsesc sub formă de medie lunară. Se poate determina irradiația solară orară folosind ecuația (157).

$$r_i = \frac{G_t}{H} \quad (3)$$

Unde: G_t - irradiația solară orară; H - irradiația solară lunară; r_t - proporție dintre radiația orară și radiația de zi. Valoarea lui r_t este calculată folosind studiile lui Collares-Pereira și Rabl [5,6]:

$$r_i = \frac{\pi}{24} \cdot (a + b \cdot \cos \omega) \cdot \left(\frac{\cos \omega - \cos \omega_s}{\sin \omega_s - \frac{\pi \cdot \omega_s}{24} \cdot \cos \omega_s} \right) \quad (4)$$

Coefficienții a și b sunt descriși de următoarele ecuații:

$$a = 0.409 + 0.5016 \cdot \sin(\omega_s - 60) \quad (5)$$

$$b = 0.6609 + 0.4767 \cdot \sin(\omega_s - 60) \quad (6)$$

În (159) și (170) ω și ω_s reprezintă unghiul orar ("hour angle"), respectiv unghiul orar pentru răsăritul soarelui ("sunset hour angle"). Unghiul orar, ω , descrie poziția instantanee a soarelui și se calculează cu subrutina ω , Fig. 4-2. Este pozitiv înainte de amiază și negativ după amiază. Unghiul orar depinde de variabilele următoare: LT - timpul local; $LSTM$ - timpul solar local Meridian; N - ziua aleasă din an; B - o variabilă temporară; ΔT_{gmt} - diferența dintre timpul local și timpul Greenwich; EoT - Ecuația de timp; $Long$ - longitudine unde se află observatorul; TC - factorul de corecție de timp; LST - timpul local solar

Organigrama ω (prezentată în Fig. 4-2.) prezintă funcționarea subrutinei ω .

4.2 Unghiul orar pentru răsăritul soarelui

Se poate utiliza relația următoare pentru calculul unghiului orar pentru răsăritul soarelui:

$$\omega_s = \cos^{-1}(-\tan \phi \tan \delta) \quad \omega_s = \cos^{-1}(-\tan \Phi \tan \delta) \quad (7)$$

Unde: Φ - latitudinea; δ - declinația.

Declinația este unghiul dintre direcția de la observator spre un punct de pe bolta cerească și planul paralel la planul ecuatorului prin punctul în care se află observatorul. Calcularea acestui unghi se face utilizând relația (162):

$$\delta = 23.45 \sin\left(360 \times \frac{284 + n}{365}\right) \quad (8)$$

După ce se cunosc toate variabilele se poate calcula valoarea r_t , apoi G_t pentru 8760 de valori (365 zile, 24 ore/zi). În Fig. 4-2. este reprezentată funcționarea subrutinei de predicție a irradiației solare din oră în oră.

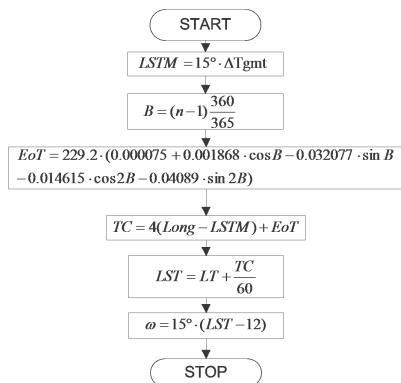


Fig. 4-2. Subrutina ω

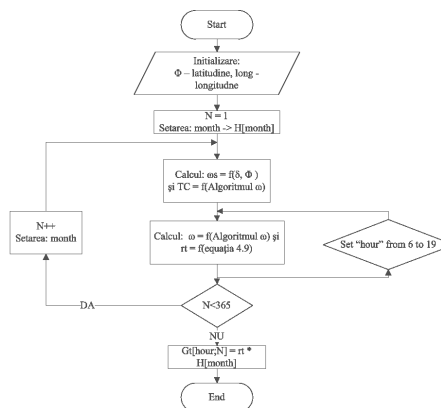


Fig. 4-3. Predicție iradiației solare din oră în oră

4.3 Structura interfeței de simulare și optimizare a microrețelei

Structura interfeței de simulare și optimizare a microrețelei este prezentată în figura următoare și are următoarele componente.

- Fereastra de consum - Primary Load
- Fereastra Set de date meteorologice - Solar Resource Inputs
- Fereastra Caracteristicile panourilor fotovoltaice - PV Characteristics
- Fereastra Caracteristicile bateriilor - Battery Characteristics
- Fereastra Energia Geotermală - Geothermal Energy Generator
- Fereastra Energie din biogaz - BioEnergy Generator.

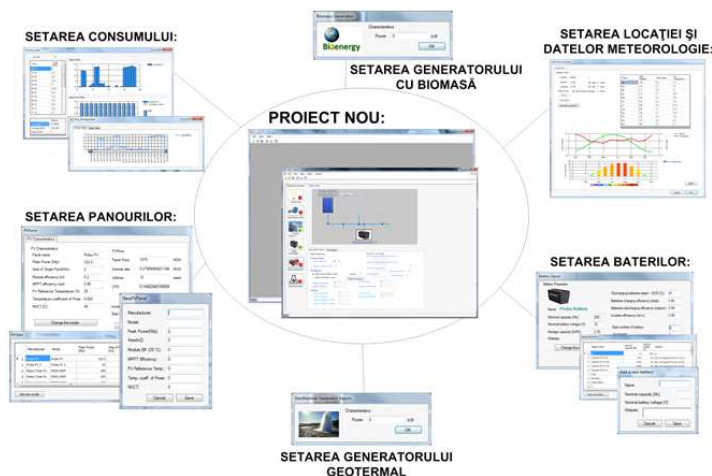


Fig. 4-4. Structura Interfeței PC pentru dimensionarea microrețelei

5 Construirea subansamblului responsabil cu obținerea energiei prin conversie fotovoltaică. Testarea în vederea optimizării

1 Construirea subansamblului responsabil cu obținerea energiei prin conversie fotovoltaică

Instalația panourilor solare este reprezentată în figura următoare :

Instalația piranometrului este reprezentată în figura următoare.

2 Testarea în vederea optimizării

Pentru optimizarea sistemului fotovoltaic a fost realizat un model 3D în PV*Sol, Fig. 5-1



Fig. 5-1 Schema bloc sistem fotovoltaic – PV*Sol

Sistemul fotovoltaic a fost instalat pe una din clădirile Universității Tehnice din Cluj-Napoca. Sistemul fotovoltaic este format din 12 panouri fotovoltaice de la ET-Solar, ET-P660250WW a câte 250W fiecare și un invertor solar de la SMA, Sunny Boy 3600TL-21. Panourile sunt conectate în serie și legate la una din cele două intrări MPPT ale invertorului. Invertorul furnizează maximum de energie solară obținută de la panourile mici.



Fig. 5-2 Instalare sistem fotovoltaic – 3D

Panourile sunt așezate pe două rânduri pe un cadru metalic înclinat la 45°. Orientarea panourilor depinde de orientarea clădirii pe care sunt panourile instalate. Panourile vor fi orientate la 138°, azimut -42°. Suprafața acoperită de panouri este de 19.5m². Conexiunea panourilor se realizează cu conductor cu secțiunea de 6mm², lungime 100m la invertorul solar. Invertorul se conectează la rețea cu conductor cu secțiunea de 2.5mm² și lungime 4m. Pierderile totale în cabluri sunt de 0.97%.

Panourile solare sunt umbrite de diferite obiecte aflate pe acoperiș. O analiză a umbririi panourilor solare este reprezentată în Fig. 5-3. Reducerea de energie datorată umbririi este 3.8 %/an. Temperatura panourilor și temperatura ambiantă sunt reprezentate în Fig. 5-4.

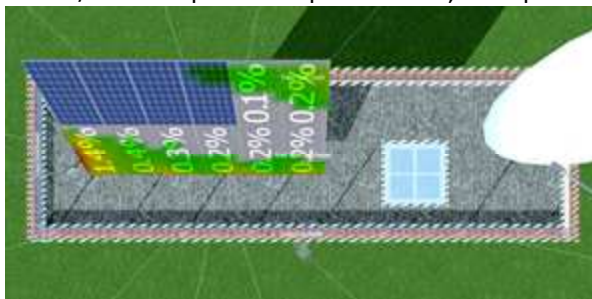


Fig. 5-3 Umbrirea panourilor

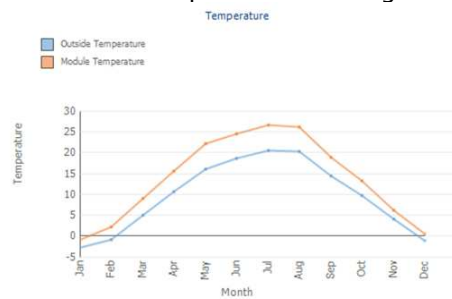


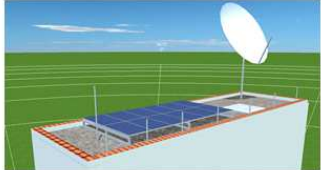
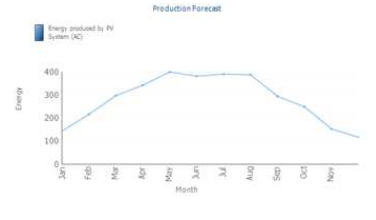
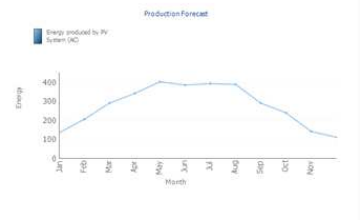
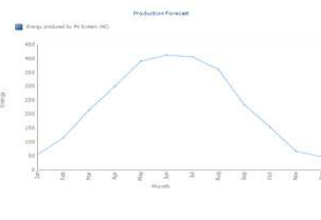


Fig. 5-4 Temperatura panourilor

Tabelul următor prezintă energia anuală produsă de sistem în funcție de orientarea panourilor. Emisiile de CO₂ evitate anual folosind sistemul fotovoltaic sunt 1,967 kg/an

Tabel 5-1

		
Orientare panouri 153°	Orientare panouri 138°	Orientare panouri orizontal
		
Energie/an = 3378kWh/an	Energie/an = 3,292.1kWh/an	Energie/an = 2754 kWh/an

5.1 Conversia energiei mecanice în energie electrică. Generatorul cu inducție

Generatoarele asincrone sunt generatoarele la care viteza rotorului este mai mare decât viteza câmpului rotitor. Se împart în generatoare cu rotor bobinat și generatoare cu rotorul în scurtcircuit. Generatoarele asincrone cu rotorul în scurt sunt cel mai robust și cel mai ieftin tip de generator electric. Acest lucru se datorează faptului că nu folosesc materiale scumpe (ex magneți permanenți cu pământuri rare) sau perii colectoare. **Din acest motiv pentru conversia energiei din biogaz în energie electrică se va folosi un generator cu inducție cu rotorul în scurtcircuit.**

În zone izolate de obicei rețeaua electrică este monofazată deoarece pentru aceeași investiție poate fi electrificată o zonă mai mare. Există deci interes pentru dezvoltarea generatoarelor cu inducție monofazate. Aceste generatoare au o înfășurare principală și o înfășurare auxiliară montată la un unghi de 90° față de înfășurarea principală. Această înfășurare împreună cu un condensator înseriat se folosește pentru a crea artificial câmpul rotitor care induce curenți în circuitul rotorului și a produce cuplul mecanic. S-a arătat însă că pentru puteri cuprinse între 2kW-15kW un generator cu inducție trifazat și un circuit de echilibrare corespunzător este mai ieftin. Un circuit de echilibrare robust constă doar în utilizarea componentelor pasive: condensatori bobine rezistențe.

Dimensionarea circuitului de echilibrare al generatorului de inducție se bazează pe circuitul echivalent al acestuia. În Fig. 5-1 este reprezentat circuitul echivalent generatorului format din rezistența statorului R_s , reactanța statorului X_s , reactanța de magnetizare X_m , reactanța rotorului X_r și rezistența rotorului R_r .

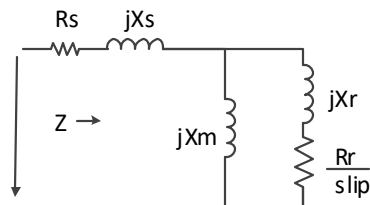


Fig. 5-1 Circuitul echivalent al generatorului cu inducție

5.1.1 Echilibrarea generatorului

Pentru a reduce dezechilibrul generatorului trifazat alimentat de la rețeaua monofazată o metodă utilizată este injectarea unui curent în circuit artificial în terminalul „liber”. Fig. 5-2 următoare ilustrează acest principiu.

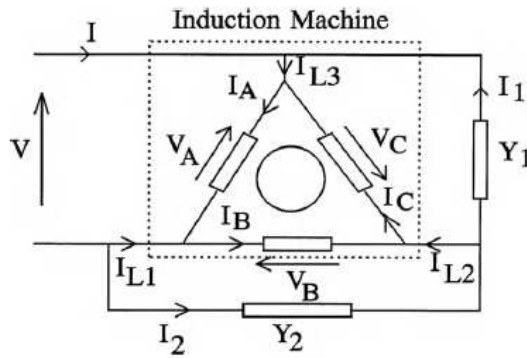


Fig. 5-2. Conectarea generatorului de inducție trifazat la rețea monofazată

În urma calculelor de echilibrare prezentate în primul raport de cercetare rezulta susceptanțele B_1 și B_2 descrise de ecuațiile de mai jos:

$$B_1 = \sqrt{3} \cdot G_p + B_p = 2|Y_p| \sin\left(2\pi/3 - \Phi_p\right) \quad (9)$$

$$B_2 = -\sqrt{3} \cdot G_p + B_p = 2|Y_p| \sin\left(\Phi_p - \pi/3\right) \quad (10)$$

5.1.2 Rezultate simulate

Pentru modelare și testare în laborator sa folosit un generator de 750W, 230V, 50Hz cu următoarele caracteristici:

- Stator resistance R_s = 9.141Ω;
- Stator leakage inductance L_s = 0.013H;
- Positive-sequence rotor resistance R_{rp} = 2.53Ω;
- Negative-sequence rotor resistance R_{rn} = 4.0 Ω;
- Rotor leakage inductance L_r = 0.014H;
- Magnetizing inductance at nominal voltage = 0.178H;

Cuplu produs de această mașină electrică este:

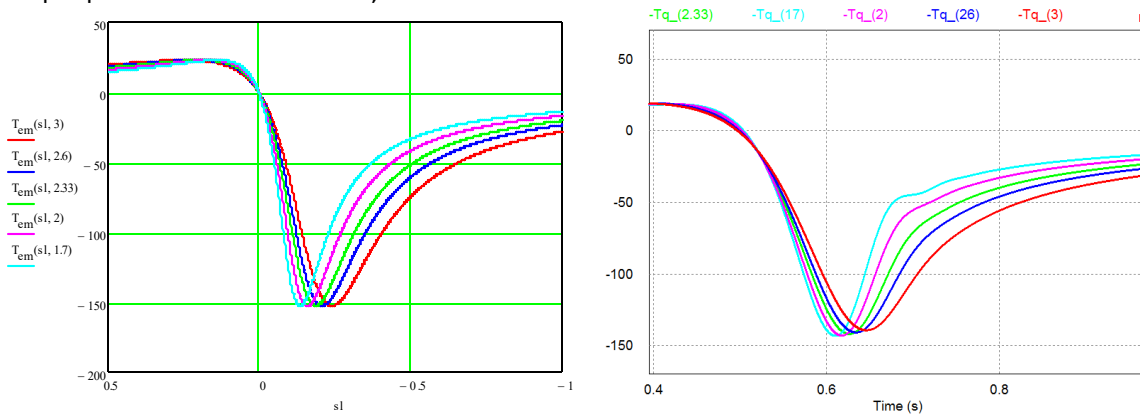


Fig. 5-3 a) Cuplul dezvoltat de mașină: a) calculat, b) simulat

Puterea generată este proporțională cu alunecarea. Cu metoda componentelor simetrice a fost determinate susceptanțele B_1 și B_2 în funcție viteza rotorului, Fig. 5-4.

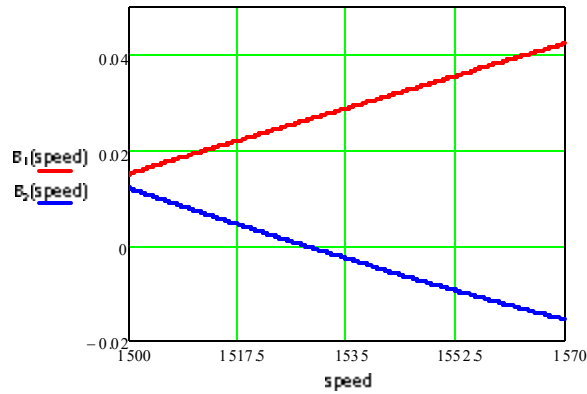


Fig. 5-4 B_1 și B_2 în funcție de viteza rotorului

Se observă că până la viteza de 1528 rpm generatorul poate fi echilibrat folosind doar condensatori iar pentru viteze și impedanțe puteri mai mari este nevoie pentru echilibrare de condensator și bobină. Condensatorii C1 și C2 se calculează pentru o viteză a rotorului fixă = 1510 rpm. Rezultă $C1 = 60\mu F$ iar $C2 = 24\mu F$.

Rezultatele simulării cu modelul prezentat sunt reprezentate în Fig. 5-5. Se observă că variind viteza rotorului generatorul se va echilibra la 1514 rpm. La această viteză generatorul produce aproximativ 170W. Cuplul generat la această viteză este de aproximativ 3Nm. În jurul turației de 1514rpm se poate observa că curenții prin cele trei înfășurări ale motorului sunt egali și defazați cu 120 de grade. Cuplul mecanic în această zonă are cea mai mică amplitudine. Deoarece generatorul poate fi echilibrat perfect doar la o anumită sarcină în afara acestui domeniu riplul cuplului mecanic crește. Acest lucru este observabil de asemenea și în amplitudinea și defazajul curenților pe fiecare fază.

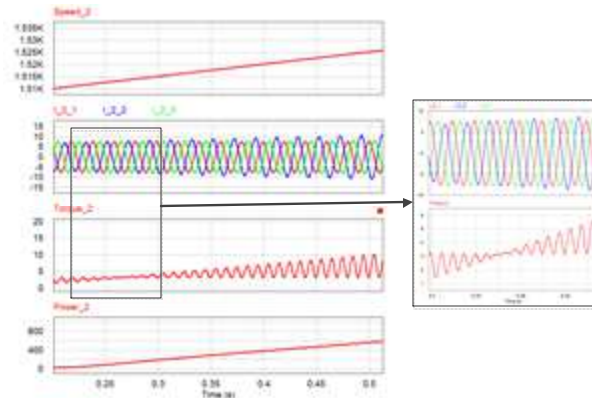


Fig. 5-5 Echilibrarea generatorului

5.1.3 Rezultate experimentale obținute

Standul experimental folosit pentru testarea circuitului de echilibrare al generatorului de inducție este prezentat în Fig. 5-6. Generatorul cu inducție este antrenat de un motor de inducție comandat de un inverter care poate simula comportamentul generatorului cu biomasă și pe cel geotermal.

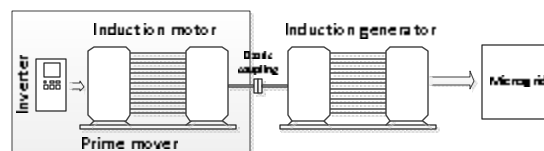


Fig. 5-6 Stand experimental

Măsurătorile care validează calculele de dimensionare sunt prezentate in figurile următoare:

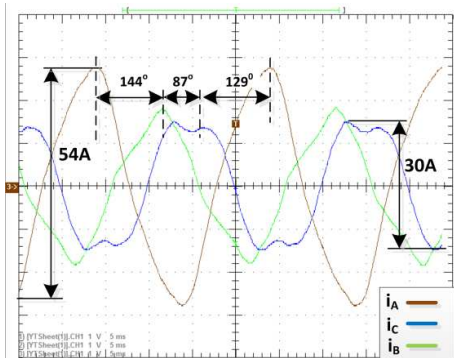


Fig. 5-7 Putere de iesire 100W

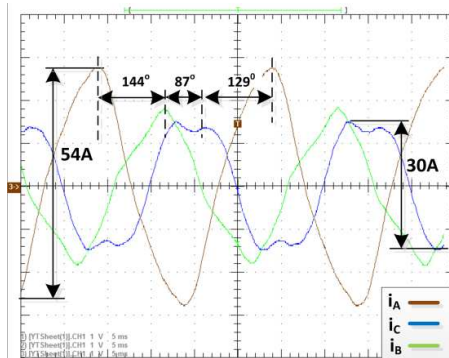


Fig. 5-8 Putere de iesire 100W

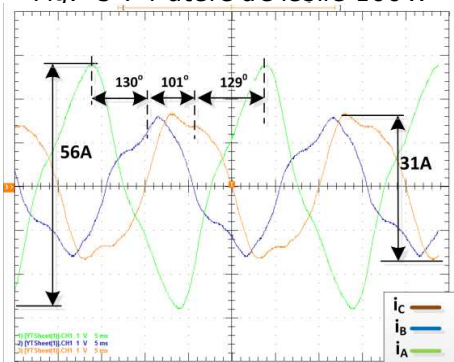


Fig. 5-9 Putere de ieșire 150W

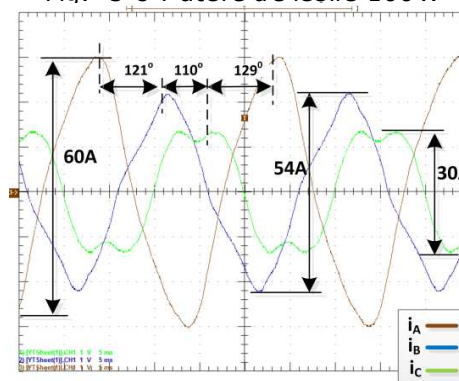


Fig. 5-10 Putere de ieșire 200W

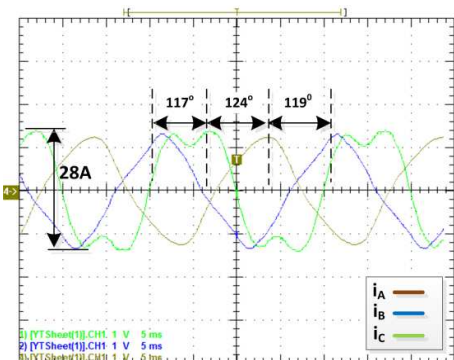


Fig. 5-11 Putere de ieșire 250W

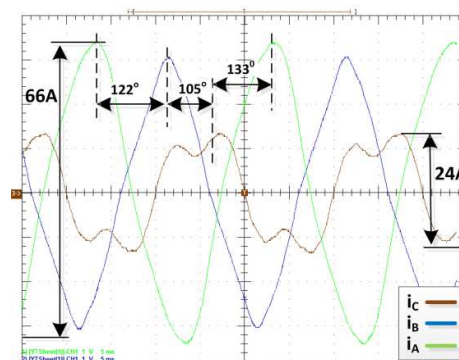


Fig. 5-12 Putere de ieșire 300W

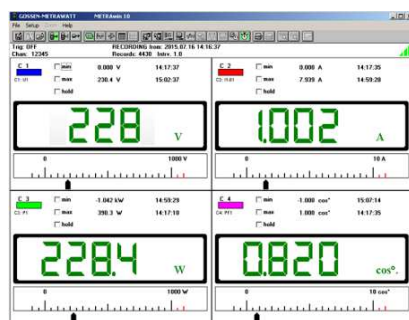


Fig. 5-13 Putere injectată in magistrala AC

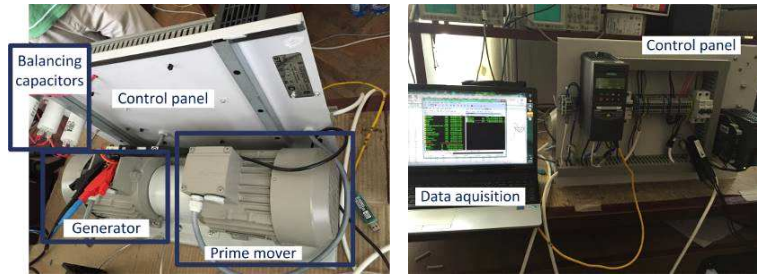


Fig. 5-14 Setup de laborator

Rezultatele obținute experimental confirmă calculele și modelul de simulare.

5.2 Modelul de semnal mic al generatorului asincron

Generatoarele cu inducție sunt de obicei folosite în microrețele de putere mică însă atenția pe care au primit-o a crescut în ultima perioadă datorită avantajelor pe care le au: dimensiuni reduse, cost scăzut, fiabilitate, nu au perii și nu au nevoie de mentenanță.

În microrețelele care folosesc atât generatoare electrice cât și invertoare pot să apară oscilații de putere care conduc la pierderi de energie și degradarea calității energiei produse suprasolicitând generatoarele și invertoarele folosite. Acest capitol dezvoltă un model de semnal mic pentru generatorul cu inducție folosit în microrețeaua propusă care va fi folosit pentru analiza stabilității microrețelei. De asemenea un model de simulare va fi propus pentru a valida rezultatele calculelor.

5.2.1 Modelul matematic

Se va folosi un generator trifazat cu inducție simetric cu următoarele caracteristici: trei înfășurări simetrice identice, întrefier uniform, circuit magnetic liniar (nu se consideră efectele saturației), flux magnetic rotor sinusoidal. În ecuațiile următoare variabilele au următorul înțeles: w_b este frecvența de bază, r_s este rezistența statorului, r_r este rezistența rotorului reflectată în stator, L_s este inductanța statorului, L_r este inductanța rotorului reflectată în stator, L_m este inductanța mutuală între rotor și stator, w_{e0} este pulsația electrică, w_{r0} este pulsația mecanică, P este numărul de poli, J este momentul de inerție, și V_{ds} , V_{qs} , V_{dr} , V_{qr} sunt tensiunile statorice și rotorice reprezentate în sistem de coordonate DQ. X_m , X_r și X_s sunt impedanțele mutuală a statorului și a rotorului calculate astfel:

$$X_m = w_b \cdot L_m; X_s = w_b \cdot L_s + X_m; X_r = w_b \cdot L_r + X_m \quad (1)$$

Ecuția (2) descrie funcționarea unei mașini electrice în sistem de coordonate DQ.

$$\begin{bmatrix} v_{qs} \\ v_{ds} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} r_s + \frac{1}{w_b} \frac{d}{dt}(X_s) & \frac{w}{w_b} X_s & \frac{1}{w_b} \frac{d}{dt}(X_m) & \frac{w}{w_b} X_m \\ -\frac{w}{w_b} X_s & r_s + \frac{1}{w_b} \frac{d}{dt}(X_s) & -\frac{w}{w_b} X_m & \frac{1}{w_b} \frac{d}{dt}(X_m) \\ \frac{1}{w_b} \frac{d}{dt}(X_m) & \frac{w-w_r}{w_b} X_m & r_r + \frac{1}{w_b} \frac{d}{dt}(X_r) & \frac{w-w_r}{w_b} X_r \\ -\frac{w-w_r}{w_b} X_m & \frac{1}{w_b} \frac{d}{dt}(X_m) & -\frac{w-w_r}{w_b} X_r & r_r + \frac{1}{w_b} \frac{d}{dt}(X_r) \end{bmatrix} \cdot \begin{bmatrix} i_{qs} \\ i_{ds} \\ i_{qr} \\ i_{dr} \end{bmatrix} \quad (2)$$

Din ecuația (2) rezultă după rearanjare ecuația (3):

$$\bar{v} = \bar{R} \cdot \bar{i} + \frac{w_e}{w_b} \cdot \bar{F} \cdot \bar{i} + \frac{w_r}{w_b} \cdot \bar{G} \cdot \bar{i} + \frac{1}{w_b} \frac{d}{dt} [\bar{X}_1 \cdot \bar{i}] \quad (3)$$

Unde:

$$\bar{R} = \begin{bmatrix} r_s & 0 & 0 & 0 \\ 0 & r_s & 0 & 0 \\ 0 & 0 & r_r & 0 \\ 0 & 0 & 0 & r_r \end{bmatrix}; \bar{X}_1 = \begin{bmatrix} X_s & 0 & X_m & 0 \\ 0 & X_s & 0 & X_m \\ X_m & 0 & X_r & 0 \\ 0 & X_m & 0 & X_r \end{bmatrix};$$

$$\bar{F} = \begin{bmatrix} 0 & X_s & 0 & X_m \\ -X_s & 0 & -X_m & 0 \\ 0 & X_m & 0 & X_r \\ -X_m & 0 & -X_r & 0 \end{bmatrix}; \bar{G} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -X_m & 0 & -X_r \\ X_m & 0 & X_r & 0 \end{bmatrix};$$

$$\bar{v} = \begin{bmatrix} \bar{v}_s \\ \bar{v}_r \end{bmatrix}; \bar{i} = \begin{bmatrix} \bar{i}_s \\ \bar{i}_r \end{bmatrix}$$

Și unde:

$$\bar{v}_s = \begin{bmatrix} v_{qs} & v_{ds} \end{bmatrix}^T; \bar{v}_r = \begin{bmatrix} v_{qr} & v_{dr} \end{bmatrix}^T;$$

$$\bar{i}_s = \begin{bmatrix} i_{qs} & i_{ds} \end{bmatrix}^T; \bar{i}_r = \begin{bmatrix} i_{qr} & i_{dr} \end{bmatrix}^T$$

Cuplul, T_l , dezvoltat de generator poate să fie exprimat astfel:

$$T_l = \frac{3}{2} \frac{P}{2 \cdot w_b} \cdot (\bar{i})^T \bar{G} \cdot \bar{i} - \frac{2 \cdot w_b}{P} \cdot D \cdot \frac{w_r}{w_b} - \frac{2 \cdot w_b^2}{P} J \cdot \frac{1}{w_b} \frac{d}{dt} \left(\frac{w_r}{w_b} \right) \quad (4)$$

Combinând (3) și (4) rezultă ecuația (5):

$$\begin{bmatrix} \bar{v} \\ T_l \end{bmatrix} = \begin{bmatrix} \bar{R} + \frac{w_e}{w_b} \cdot \bar{F} & \bar{G} \cdot \bar{i} \\ \frac{3}{2} \cdot \frac{P}{2 \cdot w_b} \cdot (\bar{i})^T \cdot \bar{G} & -\frac{2 \cdot w_b}{P} \cdot D \end{bmatrix} \begin{bmatrix} \bar{i} \\ \frac{w_r}{w_b} \end{bmatrix} + \frac{1}{w_b} \cdot \frac{d}{dt} \cdot \begin{bmatrix} \bar{X}_1 & \bar{0} \\ \bar{0}^T & \frac{2 \cdot w_b^2}{P} \cdot J \end{bmatrix} \cdot \begin{bmatrix} \bar{i} \\ \frac{w_r}{w_b} \end{bmatrix} \quad (5)$$

Liniazând ecuația (5) în vecinătatea punctului static de funcționare și considerând ca mărimi de intrare de semnal mic variabilele: \hat{v} , \hat{w}_e/w_b , \hat{w}_r/w_b și \hat{T}_l rezultă ecuația (6):

$$\frac{d}{dt} \hat{i}_e = \begin{pmatrix} -w_b \cdot X_1^{-1} \left(\bar{R} + \frac{w_{eo}}{w_b} \cdot \bar{F} + \frac{w_{ro}}{w_b} \cdot \bar{G} \right) \\ \frac{3}{2} \cdot \left(\frac{P/2}{w_b} \right)^2 \cdot \frac{1}{J} \cdot I_0^T (\bar{G} + \bar{G}^T) \end{pmatrix} \cdot \hat{i}_e + \begin{pmatrix} -w_b \cdot \bar{X}_1^{-1} \\ -w_b \cdot \bar{X}_1^{-1} \cdot \bar{F} \cdot I_o \\ -w_b \cdot \bar{X}_1^{-1} \cdot \bar{G} \cdot I_o \\ \bar{0} \end{pmatrix} \cdot b \cdot \begin{pmatrix} \hat{v} \\ \frac{\hat{w}_e}{w_b} \\ \frac{\hat{w}_r}{w_b} \\ \hat{T}_l \end{pmatrix} \quad (6)$$

Forma ecuației care descrie funcționarea motorului rescrisă în spațiul stărilor este:

$$\begin{cases} \frac{d}{dt} \hat{x} = A_{rez} \cdot \hat{x} + B_{rez} \cdot \hat{u} \\ \hat{y} = C_{rez} \cdot \hat{x} + D_{rez} \cdot \hat{u} \end{cases} \quad (7)$$

Unde: \hat{x} este vectorul de stare, \hat{y} este vectorul de ieșire, \hat{u} este vectorul de control, A_{rez} este matricea de stare, B_{rez} este matricea de intrare, C_{rez} este matricea de ieșire, iar D_{rez} este matricea feedforward. A_{rez} și B_{rez} pot fi calculate astfel:

$$A_{rez} = \begin{pmatrix} -w_b \cdot X_1^{-1} \left(\bar{R} + \frac{w_{eo}}{w_b} \cdot \bar{F} + \frac{w_{ro}}{w_b} \cdot \bar{G} \right) \\ \frac{3}{2} \cdot \left(\frac{P/2}{w_b} \right)^2 \cdot \frac{1}{J} \cdot I_0^T (\bar{G} + \bar{G}^T) \end{pmatrix} \quad (8)$$

$$B_{rez} = \begin{pmatrix} -w_b \cdot \bar{X}_1^{-1} \\ -w_b \cdot \bar{X}_1^{-1} \cdot \bar{F} \cdot I_o \\ -w_b \cdot \bar{X}_1^{-1} \cdot \bar{G} \cdot I_o \\ \bar{0} \end{pmatrix} \cdot b \quad (9)$$

Unde b se alege din tabelul 1 în funcție de intrarea dorită:

Table 1

Stator inputs	b vector
Stator voltage amplitude ΔV_s	$[1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^T$
Stator voltage phase	$[0 \ V_{r0} \ 0 \ 0 \ 0 \ 0 \ 0]^T$
Stator frequency Δf_e	$[0 \ 0 \ 0 \ 0 \ 0 \ 1/f_s \ 0]^T$

C_{rez} și D_{rez} se aleg din tabelul 2

Table 2

Output ΔY	C_{rez} vector	D_{rez} vector
Stator voltage amplitude ΔV_s	$\bar{0}$	$\begin{bmatrix} \frac{V_{qs0}}{\sqrt{(V_{qs0})^2 + (V_{ds0})^2}} & \frac{V_{ds0}}{\sqrt{(V_{qs0})^2 + (V_{ds0})^2}} & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \cdot b$
Rotor voltage amplitude ΔV_r	$\bar{0}$	$\begin{bmatrix} \frac{V_{qr0}}{\sqrt{(V_{qr0})^2 + (V_{dr0})^2}} & \frac{V_{dr0}}{\sqrt{(V_{qr0})^2 + (V_{dr0})^2}} & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \cdot b$
Stator current amplitude Δi_s	$\begin{bmatrix} \frac{I_{qs0}}{\sqrt{(I_{qs0})^2 + (I_{ds0})^2}} & \frac{I_{ds0}}{\sqrt{(I_{qs0})^2 + (I_{ds0})^2}} & 0 & 0 & 0 \end{bmatrix}^T$	$\bar{0}$
Rotor current amplitude Δi_r	$\begin{bmatrix} 0 & 0 & \frac{I_{qr0}}{\sqrt{(I_{qr0})^2 + (I_{dr0})^2}} & \frac{I_{dr0}}{\sqrt{(I_{qr0})^2 + (I_{dr0})^2}} & 0 \end{bmatrix}^T$	$\bar{0}$
Stator power ΔP_s	$\begin{bmatrix} \frac{3}{2} \cdot V_{qs0} & \frac{3}{2} \cdot V_{ds0} & 0 & 0 & 0 \end{bmatrix}^T$	$\begin{bmatrix} \frac{3}{2} \cdot I_{qs0} & \frac{3}{2} \cdot I_{ds0} & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \cdot b$
Rotor power ΔP_r	$\begin{bmatrix} \frac{3}{2} \cdot V_{qr0} & \frac{3}{2} \cdot V_{dr0} & 0 & 0 & 0 \end{bmatrix}^T$	$\begin{bmatrix} \frac{3}{2} \cdot I_{qr0} & \frac{3}{2} \cdot I_{dr0} & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \cdot b$
Electromagnetic torque ΔT_e	$\frac{3}{2} \cdot \frac{P}{2} \cdot \frac{X_m}{w_b} \cdot [I_{dr0} \ -I_{qr0} \ -I_{ds0} \ I_{qs0} \ 0]^T$	$\bar{0}$

Punctul static de funcționare al generatorului se calculează astfel:

$$I_0 = \begin{pmatrix} I_{qs} \\ I_{ds} \\ I_{qr} \\ I_{dr} \end{pmatrix} = \left(\bar{R} + \frac{w_{e0}}{w_b} \bar{F} + \frac{w_{r0}}{w_b} \bar{G} \right) \cdot \begin{pmatrix} V_{qs} \\ V_{ds} \\ V_{qr} \\ v_{dr} \end{pmatrix} \quad (10)$$

Puterea activă și reactivă sunt:

$$P = \frac{3}{2} (V_{ds0} \cdot I_{ds0} + V_{qs0} \cdot I_{qs0}) \quad (11)$$

$$Q = \frac{3}{2} (V_{qs0} \cdot I_{ds0} - V_{ds0} \cdot I_{qs0}) \quad (12)$$

Alegând corespunzător matricile din tabelele 1 și 2 modelul de semnal mic al generatorului rezultă din ecuația (13):

$$H(s) = C_{rez} (s \cdot I - A_{rez})^{-1} B_{rez} + D_{rez} \quad (13)$$

Unde s este operatorul Laplace și I este matricea unitate. Dacă în tabelul 2 se alege puterea statorului iar în tabelul 1 frecvența statorului rezultă

$$H(s) = \frac{\hat{p}_s(s)}{\hat{f}_e(s)} \quad (14)$$

5.2.2 Modelul de simulare

Un model de simulare în PSIM a fost propus pentru validarea calculelor. Modelul este prezentat în Fig. 5-15.

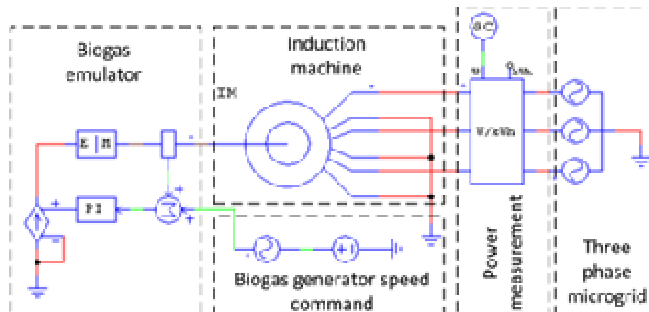


Fig. 5-15 Model de simulare

Modelul de simulare este compus din un generator de inducție, un modul de măsurare a puterii active, un model simplificat al microrețelei, și un emulator de biogaz. Controlul puterii active injectate în rețea se realizează modificând viteza generatorului cu biogaz. Comanda de modificare a vitezei generatorului de biogaz vine de la sistemul de management a energiei și se modelează cu o sursă de tensiune continuă.

5.2.3 Rezultate și discuții

Mașina de inducție folosită în microrețea are următorii parametri:

- Stator resistance R_s = 9.141Ω;
- Stator leakage inductance L_s = 0.013H;
- Positive-sequence rotor resistance R_{rp} = 2.53Ω;
- Negative-sequence rotor resistance R_{rn} = 4.0 Ω;
- Rotor leakage inductance L_r = 0.014H;
- Magnetizing inductance at nominal voltage = 0.178H;
- Number of poles P = 4
- Base frequency ω_b = 50Hz
- Mechanical angular velocity ω_0 = 51Hz
- Generator inertia J = 0.01kgm²

O comparație între valorile punctului static de funcționare simulat și calculat este prezentată în tabelul 3:

Table 3

	Calculated	Simulated
I_{ds0}	6.185A	6.07A
I_{qs0}	1.35A	1.77A
V_{ds0}	326.6V	326.6V
V_{qs0}	0	0
P	661.3W	661.22W
Q	3.03kW	2.74kW

Funcția de transfer dată de ecuația (14) calculată și simulată este reprezentată în Fig. 5-16.

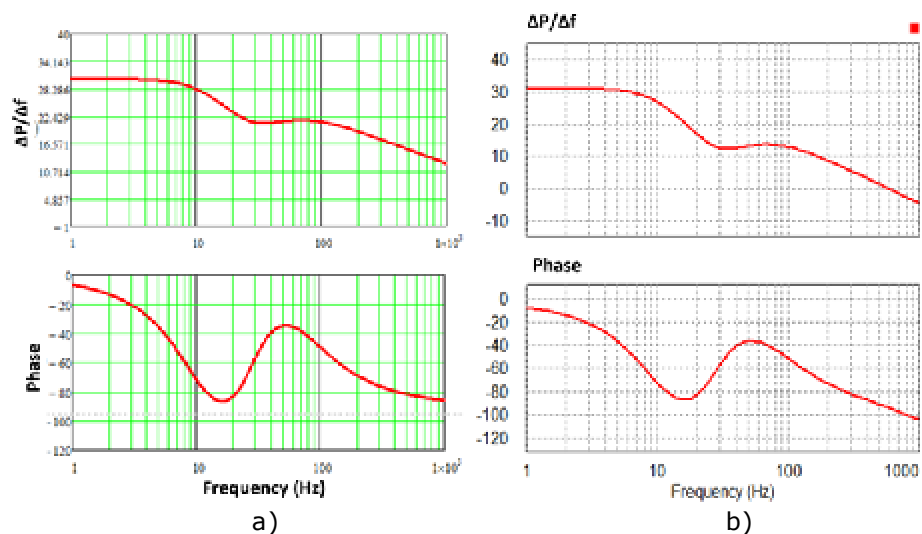


Fig. 5-16 Funcția de transfer (14) - a) calculat; b) simulat

Răspunsul la salt calculat și simulat pentru o variație de viteză de 5 rot/min este reprezentat în Fig. 5-17.

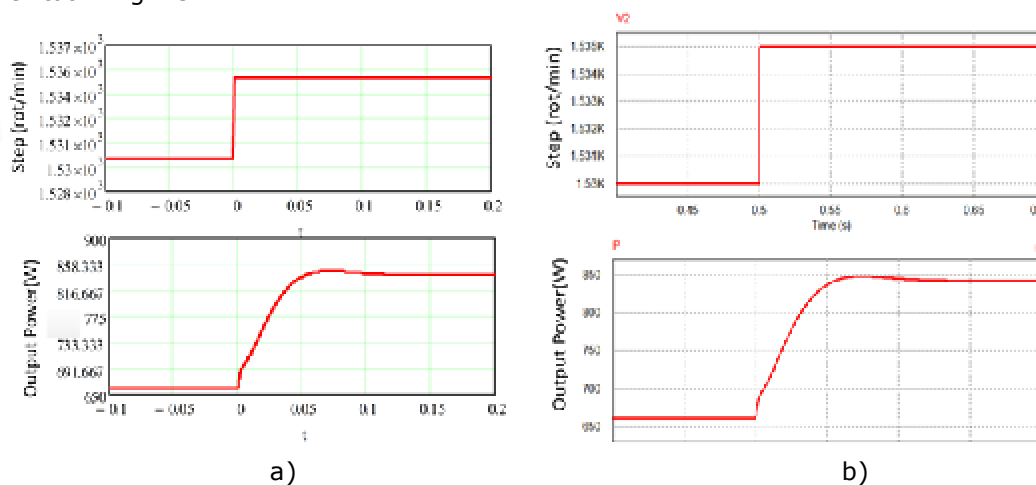


Fig. 5-17 Răspunsul la salt - a) calculat; b) simulat

Se poate observa că modelul matematic propus este validat de rezultatele simulate.