

Tehničko rešenje: Softver za simulaciju merenja u frekvencijskom domenu

Rukovodilac projekta: Vladimir Vujičić

Odgovorno lice: Vladimir Vujičić

Autori: Platon Sovilj, Vladimir Vujičić, Mirjana Trobok, Bojan Vujičić, Dragan Pejić, Nebojša Pjevalica

Razvijeno: u okviru projekta tehnološkog razvoja TR-32019 i pokrajinskog projekta 114-451-2723

Godina: 2011.

Primena: 10.12.2011.

Kratak opis

Oblast na koju se ovo tehničko rešenje odnosi je merenje u frekvencijskom domenu, primenom metode stohastičkog digitalnog merenja na intervalu. Problem koji se ovim tehničkim rešenjem rešava je određivanje parametara bitnih za projektovanje sistema za merenje u frekvencijskom domenu.

Prikazani su blok dijagrami, algoritmi i matematičko-statističke relacije implementirane u softveru. Softver je realizovan u programskom jeziku C, primenjuje se za potrebe analize i projektovanja sistema merenje u frekvencijskom domenu signala u niskonaponskoj mreži i elektroencefalografskih signala, ali i sa mogućnošću primene u merenjima u frekvencijskom domenu različitih stacionarnih i nestacionarnih signala. Softver je verifikovan u Laboratoriji za merne sisteme i Laboratoriji za metrologiju Fakulteta tehničkih nauka u Novom Sadu, poređenjima sa teorijskim analizama i eksperimentalnim rezultatima, i verifikacija je pokazala pouzdanost i tačnost rezultata koje daje softver.

Realizatori:

Fakultet tehničkih nauka u Novom Sadu

Korisnici:

ION Solutions d.o.o. kao participant; Fakultet tehničkih nauka u Novom Sadu; Medicinski fakultet u Novom Sadu; Laboratorija za eksperimentalnu psihologiju u Novom Sadu;

Podtip rešenja:

Softver (M 85)

Stanje rešenosti u svetu

Savremena merna instrumentacija je zasnovana na hardverskim komponentama za digitalizaciju signala. Mereni signali se često kondicioniraju, tako da se vremenski neprekidan kondicionirani signal uzorkuje i pretvara u diskretne digitalne vrednosti. U procesu A/D pretvaranja, tačnost i brzina su suprotstavljeni zahtevi, i merenje niskih, zašumljenih i izobličjenih signala je izazovan problem za teoriju i praksu nauke i tehnologije merenja.

Mogućnosti za pouzdan rad instrumenata sa sopstvenom slučajnom greškom su istraživane počevši od 1956. godine. Osobina ovakvog pristupa je vrlo jednostavan hardver, koji može da radi veoma brzo. Pokazano je da dodavanje slučajnog uniformnog ditera na ulaz A/D konvertora raspreže grešku merenja od ulaznog signala. Ovaj diter takođe potiskuje grešku merenja koja ima uzrok u maloj rezoluciji A/D konverzije i u spoljnjem šumu koji se superponira sa ulaznim signalu.

Sledeći ovaj opšti princip, nekoliko posebnih metoda je razvijeno za merenje srednje vrednosti jednosmernih signala, naizmeničnih signala i izobličjenih naizmeničnih signala. Implementirano je nekoliko prototipskih i komercijalnih instrumenata, za koje se pokazalo da je

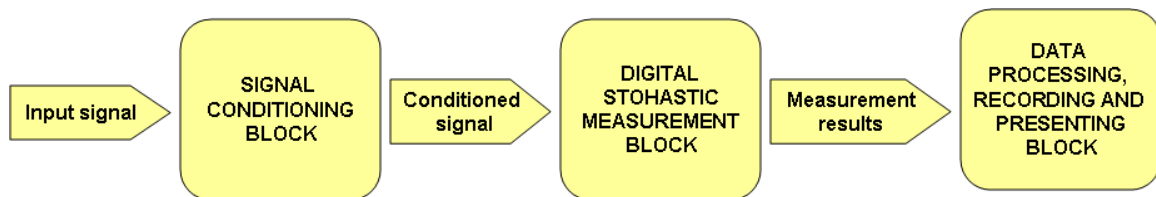
njihova merna nesigurnost ekstremno niska. Ovi metodi su nazvani metodima stohastičkog digitalnog merenja, a ovi instrumenti stohastičkim digitalnim instrumentima.

Jedan od razvijenih prototipskih instrumenata je stohastički digitalni instrument za merenje harmonika (merenje u frekvencijskom domenu) mrežnog napona i struje. Ovaj instrument vrši harmonijsku analizu jednosmerne komponente i do 49 harmonika (i za sinusne i za kosinusne komponente) na svakom od sedam različitih ulaznih kanala. Njegov rad je baziran na stohastičkoj A/D konverziji i akumulaciji, sa hardverskom strukturom projektovanom za merenje harmonika. Ovaj metod i predviđena merna nesigurnost su provereni u putem simulacije i eksperimenta, koristeći frekvenciju uzorkovanja od 250kHz po kanalu.

U svrhu jednostavnijeg i bržeg projektovanja ove vrste instrumenata, koji se mogu koristiti i za merenje šire klase stacionarnih signala, kao i nestacionarnih signala, napravljen je softver za simulaciju merenja u frekvencijskom domenu, prikazan u ovom tehničkom rešenju.

Opis tehničkog rešenja

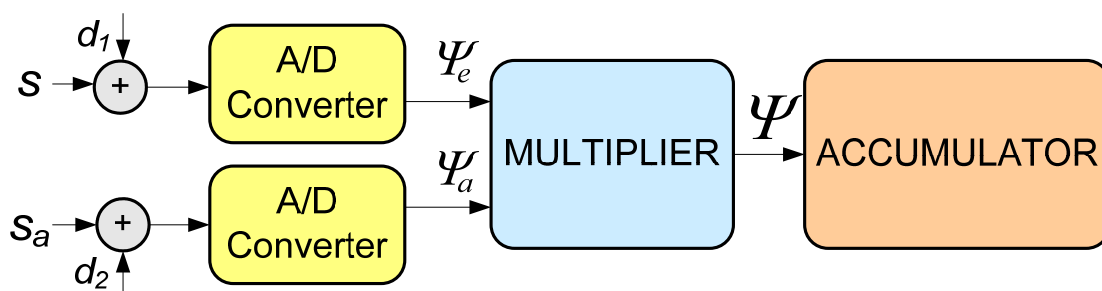
Opis obuhvata blok dijagrame, algoritme i matematičko-statističke relacije implementirane u softveru. Merni sistem zasnovan na stohastičkom digitalnom merenju može se podeliti na tri bloka: blok za kondicioniranje, blok za stohastičko digitalno merenje i blok za obradu podataka, zapisivanje i prikazivanje (slika 1).



Slika 1. Merni sistem zasnovan na stohastičkom digitalnom merenju.

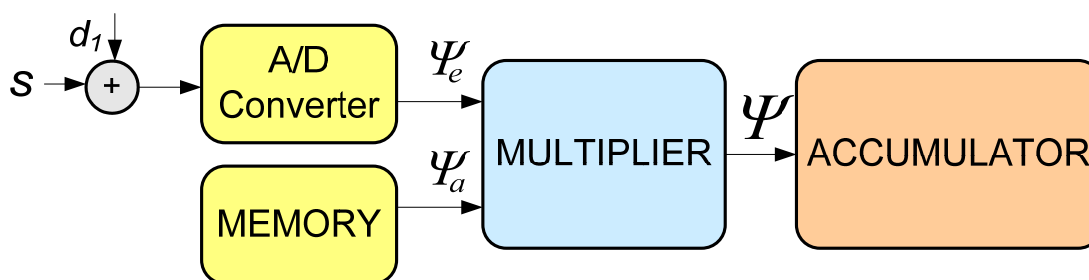
Signal na ulazu u stohastički digitalni merni blok je kondicionirani signal. Uloga ovog kondicioniranja može biti pojačanje, linearizacija, pomeranje nivoa, filtriranje, galvanska izolacija, različite tehnike za potiskivanje šuma itd. U ovom delu rešenja, fokus analize je na implementaciji stohastičkog digitalnog merenja, zanemarujući eventualne nelinearne pojave komponenti koje čine blok za kondicioniranje. Stoga, pretpostavljeno je da je kondicionirani signal s suma linearno pojačanog signala s_e (koji će se zvati samo „pojačani ulazni signal“ u nastavku teksta) i belog šuma n sa uniformnom ili Gausovom raspodelom amplituda.

Šum n predstavlja sumu svih šumova koji nisu potisnuti pre bloka za stohastičko digitalno merenje. Izvori šuma mogu biti posledica pojava unutar bloka za kondicioniranje, ali i pojava u interfejsu na ulazu bloka za kondicioniranje i pojava u interfejsu između bloka za kondicioniranje i bloka za stohastičko digitalno merenje. Konceptualni blok dijagram za stohastičko digitalno merenje jednog Furijeovog koeficijenta pojačanog ulaznog signala se može predstaviti na slici 2, što je za sledeće razmatranje jednostavniji oblik dijagrama sa slike 2.



Slika 2. Konceptualni blok dijagram za stohastičko digitalno merenje jednog Furijeovog koeficijenta pojačanog ulaznog signala, primenjen u softveru. Izlaz akumulatora se koristi za izračunavanje koeficijenta.

Pomoćni signal s_a je diterovana bazna (kosinusna ili sinusna) funkcija. Tj. važi, $s_a = R \cos k\omega_0 t$ za merenje k -tog kosinusnog Furijeovog koeficijenta, ili $s_a = R \sin k\omega_0 t$ za merenje k -tog sinusnog Furijeovog koeficijenta. Konceptualni blok dijagram se može implementirati kao na slici 3, tako da s_a nije mereni signal, već diterovani sinusni ili kosinusni signal, unapred generisan i uskladišten u memoriju.



Slika 3. Unapređeni konceptualni blok dijagram stohastičkog digitalnog merenje jednog Furijeovog koeficijenta pojačanog signala, primenjen u softveru. Umesto korišćenja dva A/D konvertora, digitalni odmerci diterovane bazisne funkcije su uskladišteni u memoriji.

d_1 i d_2 su generisani diterovani signali i oni zadovoljavaju sledeće uslove koji ograničavaju njihovu amplitudu i definišu njihovu funkciju raspodele verovatnoće:

$$0 \leq |d_i| \leq \frac{\Delta_i}{2} \tag{1}$$

$$p(d_i) = \frac{1}{\Delta_i}, \text{ za } i = 1, 2$$

Neka su uzorkovane vrednosti kondicioniranog signala s i pomoćnog signala s_a u diskretnim vremenskim trenucima unutar vremenskog intervala (T) date kao ψ_e i ψ_a , respektivno. Izmerena vrednost razlikuje se od proizvoda ulaznih signala za mernu grešku e , koja uključuje efekat kvantizacije A/D konvertora i uneseni dither:

$$\Psi = \Psi_e \cdot \Psi_a = s \cdot s_a + e \tag{2}$$

Kako se mereni kondicionirani signal sastoji od pojačanog ulaznog signala i šuma, onda je:

$$\Psi = s_e \cdot s_a + n \cdot s_a + e \quad (3)$$

Prvi član izlaza množača je signal koji je potrebno meriti a drugi član je posledica šuma. Tri člana u (3) su statistički nezavisni, a srednja vrednost $\bar{\Psi}$ je suma srednjih vrednosti pomenutih članova. Srednja vrednost trećeg člana u (3) je nula, i ne utiče na srednju vrednost očekivanog izlaza $\bar{\Psi}$ u mernom intervalu. Konačni ulazni opseg $\pm R$ bloka za stohastičko digitalno merenje definiše granice integracije za usrednjavanje efekta šuma. Stoga su dva preostala člana u srednjoj vrednosti data kao:

$$\bar{\Psi} = \frac{1}{T} \int_0^T s_e \cdot s_a dt + \left(\int_{-R}^R n \cdot p(n) dn \right) \frac{1}{T} \int_0^T s_a dt \quad (4)$$

Ako pretpostavimo da šum ima nepolarisanu Gausovu raspodelu, njegova srednja vrednost je nula tako da drugi član u (4) postaje takođe nula, i onda je:

$$\bar{\Psi} = \frac{1}{T} \int_0^T s_e \cdot s_a dt \quad (5)$$

U digitalnom merenju, za N odmeraka kondicioniranog signala na intervalu $[0, T]$, srednja vrednost je:

$$\bar{\Psi} = \frac{1}{N} \sum_{k=1}^N \Psi_k \quad (6)$$

Sumiranje uzoraka tokom mernog intervala je urađeno akumulatorom i ova suma je izlaz akumulatora. Ovaj izlaz se može obraditi mikroprocesorom koji deli izraz akumulatora sa brojem odmeraka N , i koji takođe izračunava svaku sinusnu (i kosinusnu) komponent k -tog harmonika (indeksi $\sin k$ i $\cos k$ ukazuju da je u pitanju izmereni k -ti sinusni i k -ti kosinusni Furijeov koeficijent).

$$a_k = \frac{2\bar{\Psi}_{\cos k}}{R}, \quad b_k = \frac{2\bar{\Psi}_{\sin k}}{R} \quad (7)$$

U bloku za stohastičko digitalno merenje, varijansa izlaza množača se sastoji od determinističke varijanse, slučajne varijanse i stohastičke varijanse greške. Ovi delovi su nekorelisani, tako da ukupna varijansa iznosi:

$$\sigma_{\Psi}^2 = \sigma_d^2 + \sigma_r^2 + \sigma_e^2 \quad (8)$$

Deterministička varijansa σ_d^2 , je određena kao:

$$\sigma_d^2 = \frac{1}{T} \int_0^T (s_e s_a)^2 dt - \left(\frac{1}{T} \int_0^T s_e s_a dt \right)^2 \quad (9)$$

Deterministička varijansa σ_d^2 je svojstvo signala i ne treba da bude uračunata u mernu nesigurnost. Slučajna varijansa i varijansa greške, σ_r^2 i σ_e^2 , zadovoljavaju centralnu graničnu

teoremu i varijansa njihove srednje vrednosti zavisi od broja odmeraka N unutar mernog intervala T :

$$\sigma_{\bar{r}}^2 = \frac{\sigma_r^2}{N}, \quad \sigma_{\bar{e}}^2 = \frac{\sigma_e^2}{N} \quad (10)$$

Standardna merna nesigurnost srednje vrednosti $\bar{\Psi}$ je definisana preko standardne devijacije:

$$u(\bar{\Psi}) = \sqrt{\sigma_{\bar{r}}^2 + \sigma_{\bar{e}}^2} \quad (11)$$

Relativna merna nesigurnost u se izračunava kao:

$$u = \frac{\sqrt{\sigma_{\bar{r}}^2 + \sigma_{\bar{e}}^2}}{\bar{\Psi}} \quad (12)$$

Standardna merna nesigurnost i relativna merna nesigurnost su ograničene sa:

$$u(\bar{\Psi}) \leq \frac{S_a \cdot (\sigma_n + \frac{\Delta_1}{2})}{\sqrt{N}} \quad (13)$$

$$u \leq \frac{S_a \cdot (\sigma_n + \frac{\Delta_1}{2})}{\bar{\Psi} \cdot \sqrt{N}}$$

Gornja granica standardne merene nesigurnosti je određena efektivnom vrednošću pomoćnog signala (S_a), šumom (σ_n), i rezolucijom A/D konvertora (Δ_1), kao i brojem odmeraka (N) u mernom intervalu. Ako je R amplituda pomoćnog signala, tada je:

$$S_a = R / \sqrt{2} \quad (14)$$

Prema prethodnim relacijama, standardna merna nesigurnost za bilo koji Furijeov koeficijent, izmeren ovom metodom, je ograničena sa:

$$u(a_k) = u(b_k) \leq \frac{\sqrt{2} \cdot (\sigma_n + \frac{\Delta_1}{2})}{\sqrt{N}} \quad (15)$$

Iz (15) se može izvesti da je standardna merna nesigurnost za amplitudu harmonika:

$$u(\sqrt{a_k^2 + b_k^2}) \leq \frac{2 \cdot (\sigma_n + \frac{\Delta_1}{2})}{\sqrt{N}} \quad (16)$$

U slučaju idealnog A/D konvertora, kvant A/D konvertora iznosi $\Delta_1=0$ i desna strana jednačine (15) se transformiše u $\sqrt{2} \cdot \sigma_n / \sqrt{N}$, što je kvadratni koren Kramer-Raove donje granice (Cramér–Rao lower bound - CRLB).

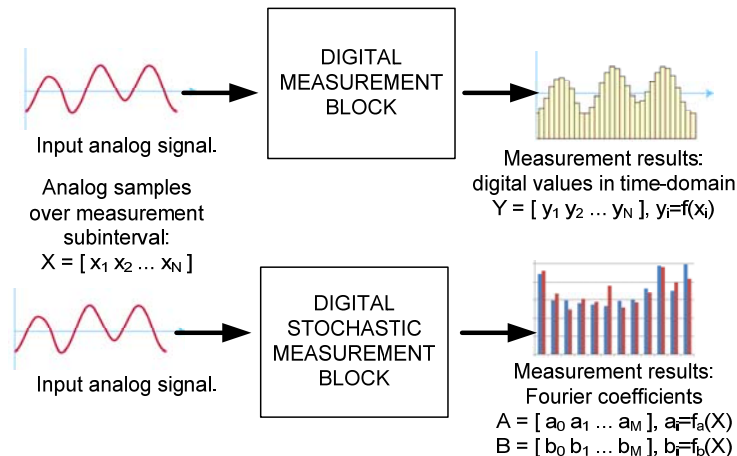
Ako se primeni kratkotrajna Furijeova transformacija na signal s_e , za koji ćemo sad pretpostaviti da je nestacionarni signal, sa prozorskom funkcijom širine jednake mernom intervalu T . Ako rezultati ove transformacije budu koeficijenti a_i i b_i , onda se s_e može predstaviti u formi trigonometrijskog polinoma oblika:

$$s_e(t) = \frac{a_0}{2} + \sum_{n=1}^M a_n \cos n\omega_0 t + \sum_{n=1}^M b_n \sin n\omega_0 t, \quad (17)$$

$0 < t < T$

$\omega_0 = 2\pi / T$ a M je red trigonometrijskog polinoma.

Poređenje koncepta stohastičkog digitalnog merenja sa tipičnim digitalnim merenjem nestacionarnog signala je dato na slici 4. Izlazi digitalnog merenja su digitalni odmerci u vremenskom domenu. Svaka digitalna vrednost je zapravo digitalizovana vrednost odgovarajućeg analognog odmerka sa ulaza i to je dobro poznati klasični pristup digitalnog merenja – odmerak po odmerak. Za razliku od ovakvog pristupa, izlazi digitalnog stohastičkog merenja su Furijeovi koeficijenti a_i i b_i . Svaki Furijeov koeficijent je funkcija svih analognih odmeraka sa ulaza tokom intervala merenja, tako da ovaj metod nije baziran na pristupu “odmerak po odmerak” već na intervalskom pristupu.



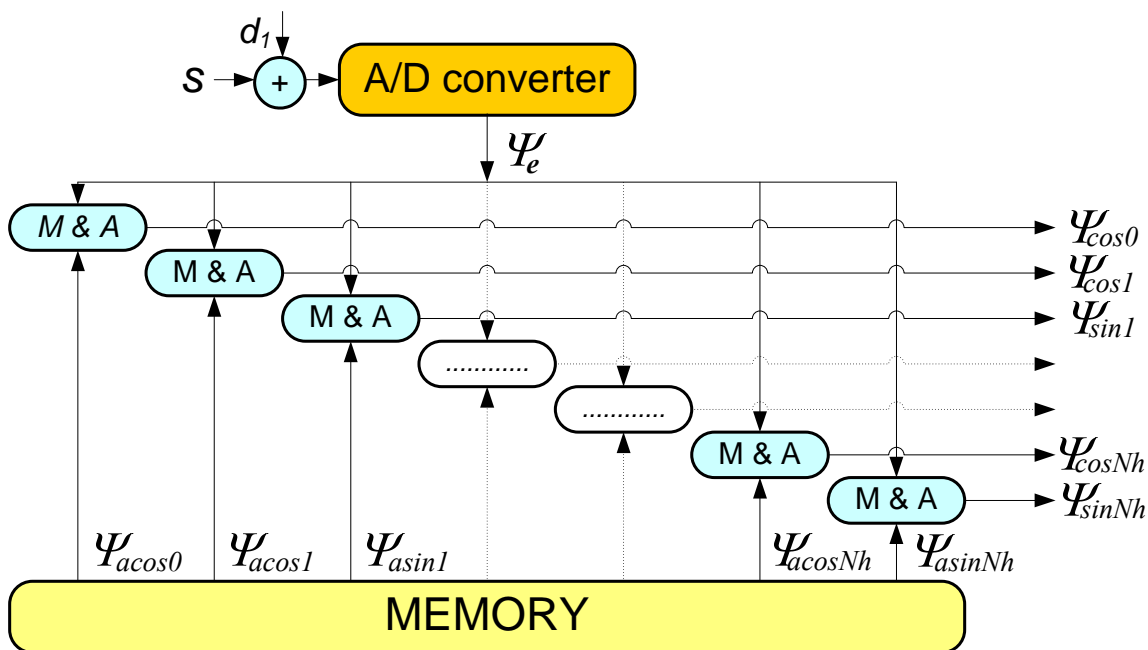
Slika 4. Poređenje koncepta stohastičkog digitalnog merenja signala u frekventnom domenu, primenjenog u softveru, sa tipičnim digitalnim merenjem u vremenskom domenu.

Na prvi pogled se može zaključiti da je stohastičko digitalno merenje pogodno samo za merenje harmonika stacionarnih signala, međutim ono se može koristiti i za merenje nestacionarnih signala. Rezultat merenja je skup harmonika koji su ulaz za kalkulaciju digitalnih vrednosti signala u vremenskom domenu. Ova kalkulacija može biti jednostavno kalkulacija trigonometrijskog polinoma u svakom diskretnom vremenskom trenutku u toku mernog intervala T , ili može biti inverzna brza Furijeova transformacija (IFFT) što je brži metod. U svakom slučaju, završni rezultati su a) skup harmonika signala na intervalu T i b) vremenske serije signala na mernom intervalu T .

Ovaj metod ne mora biti ograničen samo na merni interval T . Ako želimo da merimo signal

na širem vremenskom intervalu $[0, T_m]$, gde je $T_m = m \cdot T$, onda je moguće podeliti originalni merni interval na merne podintervale $[0, T]$, $[T, 2T]$, ... , $[(m-1)T, mT]$. Merenje i kalkulacija se mogu izvršiti na prvom intervalu, i dalje redom, interval po interval, vrednosti signala u vremenskom domenu se mogu rekonstruisati na celom velikom mernom intervalu $[0, T_m]$.

Koncept merenja Furijeovog koeficijenta pojačanog ulaznog signala može biti proširen kao na slici koja predstavlja složeniji konceptualni blok dijagram hardversko-softverske za merenje predefinisane skupa harmonika kondicioniranog signala na ulazu u stohastički digitalni merni blok. Osim jednosmerene komponente, predefinisani skup uključuje sve harmonike koji su od interesa za analizu signala. Memorija daje diterovane bazisne funkcije za svaku sinusnu i kosinusnu komponentu, i svaka sinusna i kosinusna komponenta zahtevaju jedan digitalni množač i jedan digitalni akumulator. Stoga, ako bi sistem trebao meriti jednosmernu komponentu i N_h harmonika ova struktura zahteva $2N_h+1$ množača i $2N_h+1$ akumulatora.



Slika 5. Konceptualni blok dijagram algoritma softvera stohastičkog digitalnog merenja predefinisane skupa harmonika na jednom mernom intervalu signala. Svaki element označen sa M&A se sastoji od jednog množača i jednog akumulatora. Svaki izlaz bi trebao biti podeljen sa N za kalkulaciju odgovarajućeg usrednjelog izlaza $\bar{\Psi}$, koji je neophodna za dalje kalkulacije Furijeovih koeficijenata.

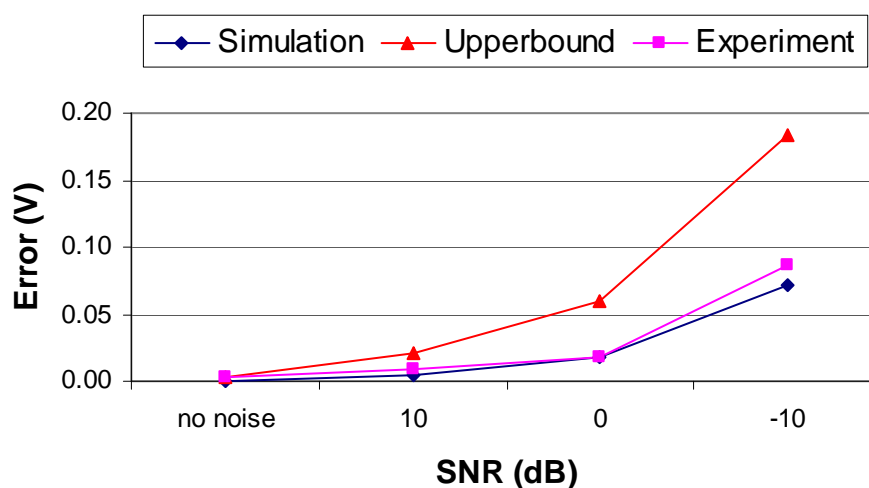
Ako se pogledaju prethodne formule, može se videti da je kvant Δ_1 definisan rezolucijom A/D konvertora, a broj odmeraka N se može izabrati kao kompromis između potrebne brzine merenja i zahtevane tačnosti. Zbog toga ovaj sistem može imati veoma dobru tačnost, čak i kada je šum značajan, zahvaljujući povećanom broju odmeraka N .

Softver je realizovan u programskom jeziku C, primenjuje se za potrebe analize i projektovanja sistema merenje u frekvencijskom domenu signala u niskonaponskoj mreži i elektroencefalografskih signala, ali i sa mogućnošću primene u merenjima u frekvencijskom domenu različitih stacionarnih i nestacionarnih signala (poput npr. seizmičkih signala).

Softver je verifikovan u Laboratoriji za merne sisteme i Laboratoriji za metrologiju Fakulteta tehničkih nauka u Novom Sadu, poređenjima sa teorijskim analizama i eksperimentalnim rezultatima, i verifikacija je pokazala pouzdanost i tačnost rezultata koje daje softver.

Tabela 1. Rezultati verifikacije: srednja greška harmonika eksperimenta i simulacije, poređena sa teorijskom gornjom granicom za mernu nesigurnost.

SNR (dB)	bez šuma	10	0	-10
Eksperimentalna greška (V)	2.50E-03	9.24E-03	1.75E-02	8.71E-02
Simulaciona greška (V)	1.34E-04	3.93E-03	1.73E-02	7.21E-02
Teorijska gornja granica (V)	1.41E-03	1.05E-02	3.00E-02	9.16E-02



Slika 6. Rezultati verifikacije: grafičko poređenje srednje greške eksperimenta i simulacije po harmoniku sa teorijskom gornjom granicom merne nesigurnosti.

Prilog: izvorni kod ključnih blokova softvera za simulaciju merenja u frekvencijskom domenu

```
// ADC za SAADK merenje
int RB = 8; //rezolucija u bitima
int R=round(pow(2.0,(double)RB)-1); //broj nivoa ADC
double Umax=2.5; //max vrednost ADC [V]
```



```

double kadc=2.0*Umax/((double)R-1.0); // Koeficijent AD konvertora. Pretvara
digitalni broj u napon. Inverzan pretvara napon u izlazni digitalni broj AD konvertora.
// amplituda ditera;
double D = Umax/((double)R-1.0);

// POJACALO
double G=7680.0;
double Skal=G/1000000.0;

// Parametri za bazisnu funkciju
int RBb = 10; //rezolucija u bitima
int Rb=round(pow(2.0,(double)RBb)-1); //broj nivoa ADC
double kadcb=2.0*Umax/((double)Rb-1.0); // Koeficijent AD konvertora. Pretvara
digitalni broj u napon. Inverzan pretvara napon u izlazni digitalni broj AD konvertora.
double B=Umax; //amplituda bazisnih funkcija
double Db = Umax/((double)Rb-1.0);// amplituda bazisnog ditera;

// Parametri AD konvertora za klasicko merenje
double fk = 256.0;
int RBk = 12;
int Rk = round(pow(2.0,(double)RBk)-1); //broj nivoa ADC
double Umaxk=2.5; //max vrednost ADC [V]
double kadck=2*Umaxk/((double)Rk-1.0); // Koeficijent AD konvertora. Pretvara
digitalni broj u napon. Inverzan pretvara napon u izlazni digitalni broj AD konvertora.
double fur_fakt = 2.0/(B*(double)N); // faktor za dobijanje furijevih koeficijenata

// OCITAVANJE originalnih vrednosti iz fajla
double eeg[L]; // ORIGINALNI signal
char filename[255];
strcpy(filename,"c:\\temp\\eeg_values.txt");
//strcpy(filename,"c:\\temp\\tdomen_ulaz.txt");
fstream file_stream(filename);
char temp[100];
char* pCnvEnd;
double As=0.0; // RMS amplituda signala
for(int i= 0; i < L; i++) {
    file_stream.getline(temp, 100);
    eeg[i]=Skal*strtod(temp,&pCnvEnd);
    As=As+eeg[i]*eeg[i]; // akumulacija za RMS amplitudu signala
}

// Parametri SUMA
As=pow(As/(double)L, 0.5); // RMS
amplituda signala
//double Anoise=(As/pow(10.0,(SNRdb/20.0)))/0.577; // Amplituda uniformnog suma.
(Imamo deljenje sa 0.577 jer amplituda i RMS za uniformni sum stoje u odnosu 1/0.577)
double Anoise=As/pow(10.0,(SNRdb/20.0)); // Amplituda gausovog suma.

srand ( time(NULL) );//Inicijalizacija semena za random number generator

// Klasicko merenje
double yk[L];// = new (nothrow) double[T*fk];

// Primena SAADK METODE - akumulacija i klasickne metode
double Ukumsin[Nh];// Inicijalizacija akumulatora sin i cos odbiraka za SA metodu
double Ukumcos[Nh];
for (int j= 0; j < Nh; j++) {
    Ukumsin[j] = 0.0;
    Ukumcos[j] = 0.0;
}
for (int i=0; i<N; i++) {

    int index = i*L/N;

    double wt=2.0*pi*f*((double)index)/fk; // omega*t
    // eeg signal + sum + diter;

    double eeg_zasumljen=eeg[index]+1.0*Anoise*random_normal();// ZASUMLJIVANJE
signala

    int test_odbirak_u_originalnom_vrem_momentu = (i*L)%N;
    if (test_odbirak_u_originalnom_vrem_momentu==0)
        yk[index] = ((double)round(eeg_zasumljen/kadck))*kadck;//klasicko
merenje

```

```

double y1=eeg_zasumljen+2.0*D*(random_uniform()-0.5);// dodaj diter za SAADK
merenje

// (eeg signal + sum + diter)*(pomocni signal + diter);
for (int j=0; j<Nh; j++) {
    double ysin=B*sin(((double)j)*wt)+2.0*Db*(random_uniform()-0.5);
    double ycos=B*cos(((double)j)*wt)+2.0*Db*(random_uniform()-0.5);
    // izlaz ADC;
    double Uout1 = round(y1/kadc)*kadc;
    double Uoutsin = round(ysin/kadcb)*kadcb;
    double Uoutcos = round(ycos/kadcb)*kadcb;
    Ukumsin[j] = Ukumsin[j] + Uout1*Uoutsin;
    Ukumcos[j] = Ukumcos[j] + Uout1*Uoutcos;
}

// FFT originalnog signala
double ulazniAK[Nh], ulazniBK[Nh];
FFT fftCalc;
double xim[L], yre[L], yim[L];
for (int t=0; t<L; t++) {
    xim[t]=0.0;yre[t]=0.0;yim[t]=0.0;
}
fftCalc.fft(L,eeg,xim,yre,yim);
for (int t=0; t<L; t++) {
    yre[t]=yre[t]*2.0/(double)L;
    yim[t]=yim[t]*2.0/(double)L;
}
// Skaliramo da bi dobili uV originalnog, nepojacanog EEG signala
for (int j=0; j<Nh; j++) {
    ulazniAK[j] = yre[j]/Skal;
    ulazniBK[j] = -1.0*yim[j]/Skal;//patch (FFT i SAADK daju suprotan znak;
uzrok za sada nepoznat)
}

// SAADK spektar

double sin_koeficijent[Nh];
double cos_koeficijent[Nh];
double AK_apsSAADKGreskaAVG = 0.0;
double AK_apsSAADKGreskaMAX = 0.0;
double BK_apsSAADKGreskaAVG = 0.0;
double BK_apsSAADKGreskaMAX = 0.0;
for (int j=0; j<Nh; j++) {

    // Izmereni Furijeovi koeficijenti
    sin_koeficijent[j]=fur_fakt*(double)Ukumsin[j]/Skal; // sinusni koeficijent
    cos_koeficijent[j]=fur_fakt*(double)Ukumcos[j]/Skal; // kosinusni
koeficijent

    AK_apsSAADKGreskaAVG = AK_apsSAADKGreskaAVG + abs(ulazniAK[j]-
cos_koeficijent[j]);
    BK_apsSAADKGreskaAVG = BK_apsSAADKGreskaAVG + abs(ulazniBK[j]-
sin_koeficijent[j]);
    if (abs(ulazniAK[j]-cos_koeficijent[j])>AK_apsSAADKGreskaMAX)
        AK_apsSAADKGreskaMAX = abs(ulazniAK[j]-cos_koeficijent[j]);
    if (abs(ulazniBK[j]-sin_koeficijent[j])>BK_apsSAADKGreskaMAX)
        BK_apsSAADKGreskaMAX = abs(ulazniBK[j]-sin_koeficijent[j]);
}
AK_apsSAADKGreskaAVG = AK_apsSAADKGreskaAVG/Nh;
BK_apsSAADKGreskaAVG = BK_apsSAADKGreskaAVG/Nh;
fprintf (fileCOSSaadkAVG, "%10.10f\t", AK_apsSAADKGreskaAVG);
fprintf (fileSINSaadkAVG, "%10.10f\t", BK_apsSAADKGreskaAVG);
fprintf (fileCOSSaadkMAX, "%10.10f\t", AK_apsSAADKGreskaMAX);
fprintf (fileSINSaadkMAX, "%10.10f\t", BK_apsSAADKGreskaMAX);

// FFT klasicno merenih odbiraka
double AK_apsKLASGreskaAVG = 0.0;
double AK_apsKLASGreskaMAX = 0.0;
double BK_apsKLASGreskaAVG = 0.0;
double BK_apsKLASGreskaMAX = 0.0;

```

```

for (int t=0; t<L; t++) {
    xim[t]=0.0;yre[t]=0.0;yim[t]=0.0;
}
fftCalc.fft(L,yk,xim,yre,yim);
for (int t=0; t<L; t++) {
    yre[t]=yre[t]*2.0/(double)L;
    yim[t]=yim[t]*2.0/(double)L;
    yre[t]=yre[t]/Skal;
    yim[t]=-1.0*yim[t]/Skal;
}
for (int j=0; j<Nh; j++) {

    AK_apsKLASGreskaAVG = AK_apsKLASGreskaAVG + abs(ulazniAK[j]-yre[j]);
    BK_apsKLASGreskaAVG = BK_apsKLASGreskaAVG + abs(ulazniBK[j]-yim[j]);
    if (abs(ulazniAK[j]-yre[j])>AK_apsKLASGreskaMAX)
        AK_apsKLASGreskaMAX = abs(ulazniAK[j]-yre[j]);
    if (abs(ulazniBK[j]-yim[j])>BK_apsKLASGreskaMAX)
        BK_apsKLASGreskaMAX = abs(ulazniBK[j]-yim[j]);
}
AK_apsKLASGreskaAVG = AK_apsKLASGreskaAVG/Nh;
BK_apsKLASGreskaAVG = BK_apsKLASGreskaAVG/Nh;

double AK_AVG_RATIO = AK_apsKLASGreskaAVG/AK_apsSAADKGreskaAVG;
double BK_AVG_RATIO = BK_apsKLASGreskaAVG/BK_apsSAADKGreskaAVG;

double AK_MAX_RATIO = AK_apsKLASGreskaMAX/AK_apsSAADKGreskaMAX;
double BK_MAX_RATIO = BK_apsKLASGreskaMAX/BK_apsSAADKGreskaMAX;

fprintf (fileCOSKlasikAVG, "%10.10f\t", AK_apsKLASGreskaAVG);
fprintf (fileSINKlasikAVG, "%10.10f\t", BK_apsKLASGreskaAVG);
fprintf (fileCOSKlasikMAX, "%10.10f\t", AK_apsKLASGreskaMAX);
fprintf (fileSINKlasikMAX, "%10.10f\t", BK_apsKLASGreskaMAX);

fprintf (fileCOSRatioAVG, "%10.10f\t", AK_apsKLASGreskaAVG/AK_apsSAADKGreskaAVG);
fprintf (fileCOSRatioMAX, "%10.10f\t", AK_apsKLASGreskaMAX/AK_apsSAADKGreskaMAX);
fprintf (fileSINRatioAVG, "%10.10f\t", BK_apsKLASGreskaAVG/BK_apsSAADKGreskaAVG);
fprintf (fileSINRatioMAX, "%10.10f\t", BK_apsKLASGreskaMAX/BK_apsSAADKGreskaMAX);

```

Softver za simulaciju merenja u frekvencijskom domenu razvijen je na Fakultetu tehničkih nauka u Novom Sadu, u okviru tekućeg projekta br. TR-32019 kod Ministarstva prosvete, nauke i tehnološkog razvoja Republike Srbije i tekućeg projekta br. 114-451-2723 kod Pokrajinskog sekretarijata za nauku i tehnološki razvoj AP Vojvodine.

Štampano – Decembar 2011.