

## VEŽBA 9 – Adaptivni digitalni sistemi

### Potrebno predznanje

- Poznavanje programskog jezika C
- Urađena Vežba 1 – Uvod u Digitalnu Obradu Signala
- Urađena Vežba 7 – FIR filteri
- Odslušana predavanja iz predmeta OAIS DSP na temu Diskretni sistemi
- Odslušana predavanja iz predmeta OAIS DSP na temu FIR filteri
- Odslušana predavanja iz predmeta OAIS DSP na temu Adaptivni digitalni sistemi

### Šta će biti naučeno tokom izrade vežbe

U okviru ove vežbe naučićete:

- Kako funkcionišu adaptivni digitalni sistemi
- Na koji način adaptivni digitalni sistemi vrše adaptaciju prenosne karakteristike
- Neke od primena adaptivnih digitalnih sistema u realnim sistemima

### Motivacija

Diskretni FIR i IIR filtri omogućuju efikasnu kontrolu spektra u linearnim vremenski nezavisnim sistemima. U određenim slučajevima parametri sistema se menjaju u vremenu te je potrebno da se i diskretni sistem prilagođava tim promenama. Dodatno postoji potreba da se koristi neko prethodno ili naučeno znanje o obrađivanim signalima da bi se poboljšao rezultat obrade. Rešavanju problema bi mogli pristupiti korišćenjem većeg broja diskretnih FIR/IIR filtera i izborom odgovarajuće grupe filtera na osnovu analize ulaznog signala. U praksi se koriste diskretni sistemi čiji parametri menjaju u vremenu adaptirajući se na osnovu ulaznog signala. Primer sistema je diskretni FIR filter čiji koeficijenti se menjaju u vremenu.

## 1 TEORIJSKE OSNOVE

U dosadašnjoj analizi je pretpostavljeno da su parametri sistema nepromenljivi. Tako su analizirani linearni (filteri) i nelinearni diskretni sistemi i metode obrade pod uslovom da su vremenski nepromenljivi. Međutim u praksi digitalne obrade signala javljaju se i vremenski zavisni sistemi – sistemi čiji se parametri menjaju u vremenu. Najčešće su u pitanju obrade signala gde se koristi i neko prethodno ili naučeno znanje o obrađivanim signalima da bi se poboljšali rezultati obrade. Takvi sistemi se nazivaju adaptivni digitalni sistemi.

### 1.1 Adaptivni filter

Primer vremenski zavisnih sistema jeste adaptivni filter. Adaptivni filter predstavlja filter kod koga vrednosti koeficijenata zavise od vremena, i menjaju se sa indeksom  $n$ . U slučaju adaptivnog filtera sa konačnim impulsnim odzivom jednačina filtera data je sa:

$$y(n) = \sum_{k=0}^{L-1} h_n(k) \cdot x(n-k)$$

Postoje različiti tipovi adaptivnih filtera u zavisnosti od načina na koji se računaju koeficijenti filtra u svakoj iteraciji. Jedan od najčešće korišćenih algoritama jeste metoda minimizacije srednje kvadratne greške ili LMS (eng. *Least Mean Square Error*). Ovakav pristup podrazumeva računanje koeficijenata tako da se pokušava minimalizovati signal greške koji predstavlja razliku izlaza iz sistema i referentnog (željenog) izlaznog signala. Vrednost koeficijenata se koriguje proporcionalno gradijentu (prvom izvodu) kvadrata greške. Izračunavanje koeficijenata kod LMS vrši se sledećom jednačinom:

$$\bar{h}(n+1) = \bar{h}(n) + \lambda \cdot e(n) \cdot \bar{t}(n)$$

gde:

- $h(k)$  – predstavlja koeficijente filtera (niz dužine  $L$ ) u trenutku  $k$
- $t(k)$  – predstavlja niz od poslednjih  $L$  odbiraka ulaznog signala, oni koji utiču na trenutni rezultat (niz dužine  $L$ ) u trenutku  $k$
- $e(k)$  – predstavlja grešku predikcije u trenutku  $k$  (skalarna promenljiva čija je vrednost razlika referentnog i izlaznog signala)
- $\lambda$  - predstavlja koeficijent adaptacije

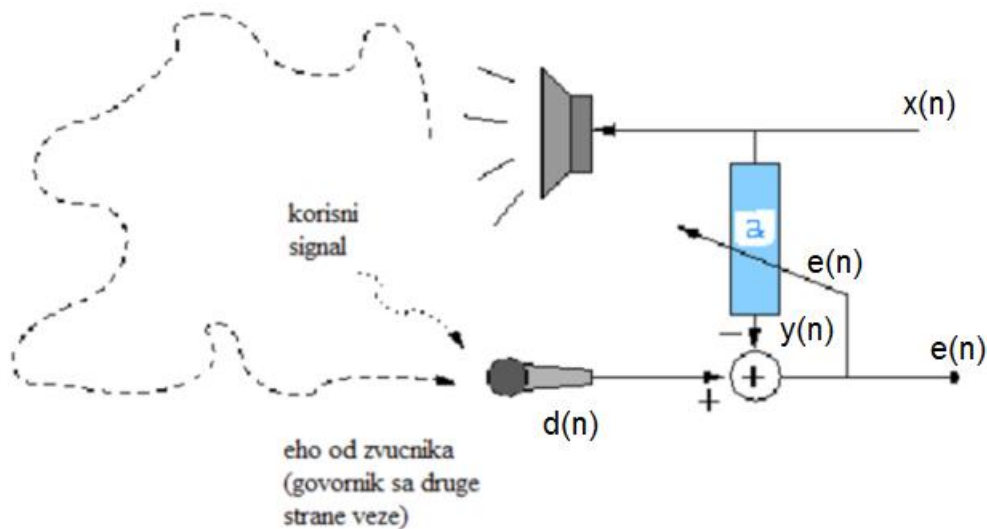
Jedna modifikacija prethodnog algoritma jeste normalizovana LMS ili NLMS. Kod ovog algoritma adaptivna konstanta se prilagođava snazi signala, pa je:

$$\bar{h}(n+1) = \bar{h}(n) + \frac{\lambda \cdot e(n) \cdot \bar{t}(n)}{\gamma + \bar{t}^T(n) \cdot \bar{t}(n)}$$

gde  $\gamma$  predstavlja zaštitnu konstantu, koja onemogućava da rezultat adaptivne formule (koeficijent filtera) postane prevelik kada je proizvod  $\bar{t}^T(k) \cdot \bar{t}(k)$  privremeno mali.

## 1.2 Primena adaptivnih filtera

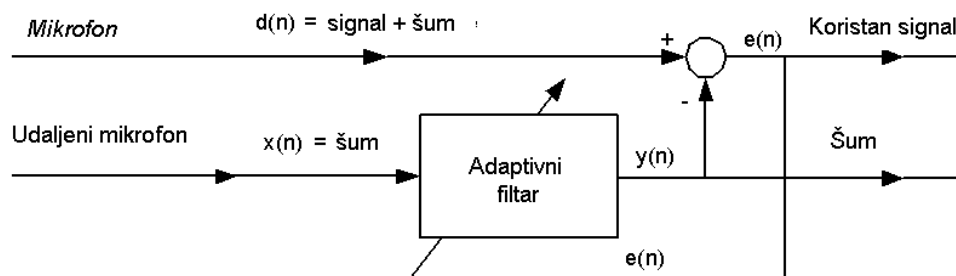
Adaptivni filteri se često koriste u sistemima za uklanjanje šuma ili akustičnog eha. Osnovna ideja kod ovakve primene je da se nepoznati sistem koji proizvodi šum u signalu pokuša modelovati adaptivnim filtrom, koji će tokom vremena menjati svoje karakteristike. Jedan od primera pojave akustičnog eha jeste u sistemima za komunikaciju, kada se koriste u tzv. *hands-free* režimu. Neželjeni signal ili šum je zapravo signal koji dolazi sa druge strane, šalje se na zvučnik, i potom se vraća na mikrofonski manifestujući se kao neželjeni eho.



Slika 1 - Sistem za uklanjanje akustičnog eha

U opisanom slučaju zadatak adaptivnog filtera jeste da modeluje sistem koji se sastoji iz zvučnika, sobe i mikrofona. Cilj je da se propuštanjem signala koji dolazi na zvučnik kroz adaptivni filter dobije signal približno jednak neželjenom delu signala učitano sa mikrofona. Rezultat filtriranja se oduzima od signala primljenog na mikrofonsu i na taj način se dobija samo koristan signal. Ova razlika ujedno predstavlja i signal greške koji se koristi za adaptaciju koeficijentata. Opisani sistem prikazan je na slici 1.

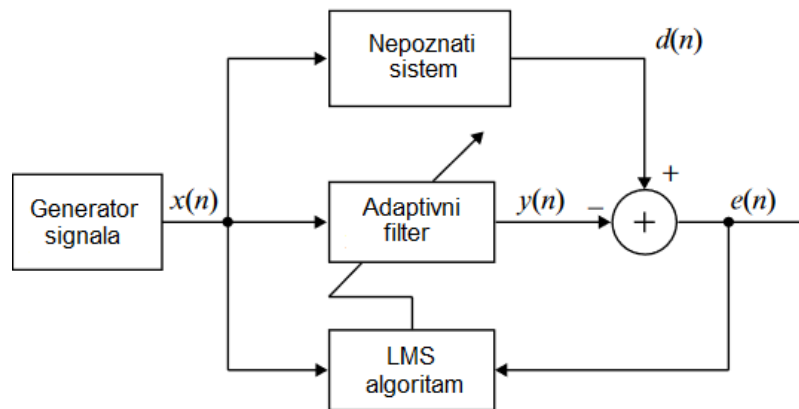
Još jedan slučaj korišćenja adaptivnih filtera za otklanjanje šuma jeste pristup sa dva izvora zvuka (dva mikrofona). Ovaj pristup često se koristi u pilotskim kabinama. Jedan mikrofonski koristi pilot, za komunikaciju, dok se drugi mikrofonski nalazi na određenoj udaljenosti od njega.



Slika 2 - Sistem za adaptivno uklanjanje šuma

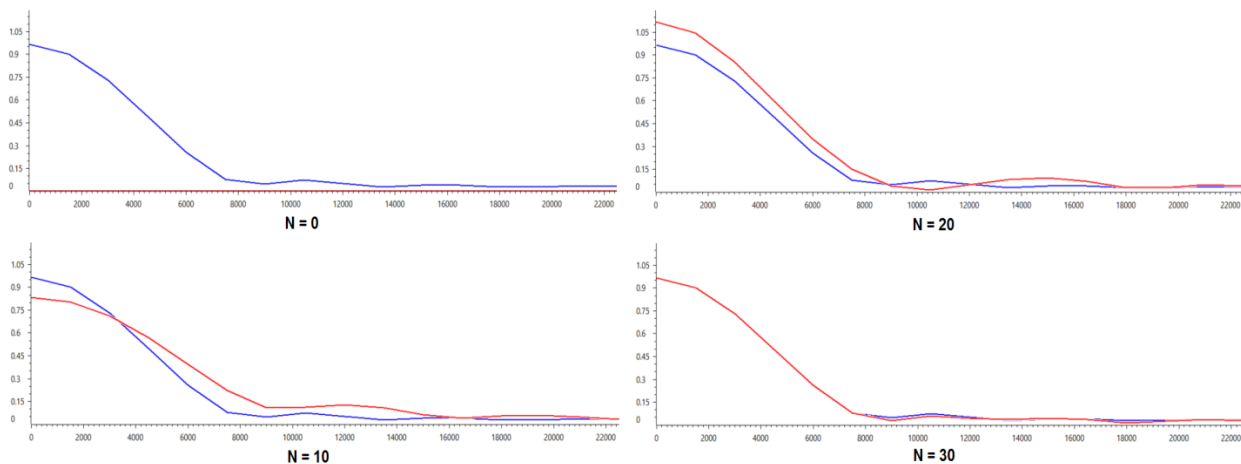
Prvi mikrofoni služi za snimanje zašumljenog signala koji je potrebno preneti dalje, a drugi za snimanje samog šuma. Zadatak adaptivnog filtera jeste da na osnovu snimljenog šuma izvrši predikciju komponente šuma u zašumljenom signalu. Signal sa mikrofona kojim se snima šum ima istu funkciju kao i signal koji pristiže na zvučnik u prethodnom primeru.

Još jedna česta primena adaptivnih sistema jeste identifikacija nepoznatih sistema za obradu signala. Identifikacija sistema predstavlja eksperimentalni pristup modelovanju sistema za obradu signala. Osnovna ideja jeste da se na osnovu odziva nepoznatog sistema konstruiše model sa jednakim ili približno jednakim odzivom. Kao model se uzima adaptivni FIR ili IIR filter. Identifikacija se vrši tako što se nepoznati sistem i model izvršavaju istovremeno za istu pobudu, pri tom se izlaz iz nepoznatog sistema uzima kao referentni signal za računanje greške prilikom adaptacije koeficijenata. Blok dijagram sistema za identifikaciju koristeći adaptivni filter prikazan je na slici 3.



Slika 3 - Identifikacija sistema koristeći adaptivni filter

Na slici 4 prikazani su impulsni odzivi sistema koji se identifikuje (plavo) i adaptivnog filtera (crveno), nakon obrađenih  $N$  odbiraka. Sistem koji se identifikuje u ovom slučaju jeste niskopropusni FIR filter 8-og reda sa graničnom učestanosti 5 kHz. Kao model se koristi adaptivni FIR filter 10-og reda.



Slika 4 - Prenosna karakteristika sistema koji se identifikuje (plavo) i adaptivnog filtra (crveno) nakon  $N$  obrađenih odbiraka

### 1.3 Implementacija adaptivnog filtra upotrebom programskog jezika C

U ovom poglavlju dat je primer celobrojne implementacije adaptivnog FIR filtra koji koristi LMS algoritam za adaptaciju koeficijenata. Primer je realizovan tako da odgovara sistemu za adaptivno uklanjanje šuma prikazanom na slici 2. Filter je predstavljen funkcijom *lms\_filter*. Parametri funkcije su odbirak ulaznog signala *x*, odbirak referentnog signala *d*, potom niz sa koeficijentima filtra *coeffs*, niz koji služi za pamćenje prethodnih ulaznih odbiraka *history*, broj koeficijenata, pokazivač na trenutni indeks čitanja iz *history* niza i koeficijent adaptacije  $\lambda$  (*lambda*) predstavljen kao broj u opsegu [-1, 1) skaliran na opseg tipa Int16.

```
Int16 lms_filter(Int16 x, Int16 d, Int16 *coeffs, Int16 *history, Uint16 n_coeff,
Uint16 *p_state, Int16 lambda)
{
    Int16 y, error;

    y = fir_circular(x, coeffs, history, n_coeff, p_state);
    error = d - y;
    lms(error, coeffs, history, n_coeff, p_state, lambda);
    return error;
}
```

Unutar prikazane funkcije vrši se poziv funkcije za filtriranje koja koristi kružni bafer (*fir\_circular*). Ova funkcija je opisana u okviru vežbe 7. Kao ulazni odbirak filteru se prosleđuje odbirak ulaznog signala. Dobijena vrednost filtriranog odbirka se koristi za računanje vrednosti greške tako što se oduzme od referentnog signala. Potom se vrši adaptacija koeficijenata koristeći LMS algoritam odnosno pozivom funkcije *lms*. Nakon izvršene adaptacije funkcija vraća vrednost greške kao izlazni odbirak sistema (kao što je prikazano na slici 2).

Funkcija *lms* kao parametre prima vrednost greške, niz sa koeficijentima, niz sa prethodnim ulaznim odbircima potreban za računanje novih koeficijenata, broj koeficijenata, pokazivač na trenutni indeks čitanja iz *history* niza i koeficijent adaptacije  $\lambda$ . Podrazumeva se da prosleđeni indeks čitanja ima vrednost indeksa najstarijeg elementa u okviru niza *history*. Prvi korak unutar tela funkcije jeste množenje vrednosti greške sa koeficijentom adaptacije. Potom se računa nova vrednost svakog koeficijenta primenom jednačine date u okviru LMS algoritma. Svaki koeficijent jednak je zbiru trenutne vrednosti koeficijenta i proizvoda koeficijenta adaptacije, vrednosti greške i odgovarajućeg ulaznog odbirka. Za sabiranje koristi se 32-bitna promenljiva *accum* kako ne bi došlo do prekoračenja opsega koje bi dovelo do neispravnog rezultata. Nakon izvršenog sabiranja potrebno je vrednost promenljive *accum* ograničiti na opseg 16-bitnog označenog tipa.

```
void lms(Int16 error, Int16 *coeffs, Int16 *history, Uint16 n_coeff, Uint16 *p_state,
Int16 lambda)
{
    Int32 accum;
    Int16 lambda_error, x, i;
    Int16 state = *p_state;

    lambda_error = _smpy(lambda, error);

    for (i = n_coeff - 1; i >= 0; i--)
```

```
{
    x = history[state];
    if (++state >= n_coeff)
    {
        state = 0;
    }

    accum = coeffs[i];
    accum += _smpy(lambda_error, x);

    if(accum > 32767)
        coeffs[i] = 32767;
    else if(accum < -32768)
        coeffs[i] = -32768;
    else
        coeffs[i] = (Int16)accum;
}
}
```

## 2 ZADACI

### 2.1 Zadatak 1

U okviru datog projektnog zadatka data je realizacija adaptivnog sistema za uklanjanje šuma prikazana na slici 2. Sistem je dizajniran tako da očekuje signal  $d(n)$  (snimljeni signal + šum) na levom ulaznom kanalu, a snimljeni šum  $x(n)$  na desnom kanalu. Za svaki primljeni odbirak poziva se funkcija *lms\_filter* koja vrši filtriranje i adaptaciju koeficijenata. Povratna vrednost funkcije jeste vrednost greške koja se prosleđuje na desni izlazni kanal. Na levom izlaznom kanalu je prosleđen neizmenjen ulazni signal snimljen sa levog ulaznog signala.

1. Uvući projektni zadatak 1 u radni prostor
2. Povezati *line out* izlaz vašeg računara na *line in* ulaz na razvojnoj ploči.
3. Koristeći program *Audacity* otvoriti datoteku *SpeechPlusNoise.aup*. Prvi signal koji se nalazi u okviru datoteke govorni signal, koji predstavlja signal od značaja i podešen je tako da se reprodukuje na levom kanalu. Naredna dva signala su sinusni signal frekvencije 3 kHz, amplitude 0.5, i kosinusni signal iste frekvencije i amplitude 0.4 na desnom kanalu. Prvim signalom zašumljen je govor, dok drugi signal predstavlja snimljeni šum udaljenim mikrofonom. Pored ovih signala projekat sadrži i sinusni i kosinusni signal frekvencije 200 Hz kao i signale belog šuma, takođe podešene za reprodukciju na levom i desnom kanalu. Poslednja 2 signala su utišani koristeći opciju *Mute*.
4. Reprodukovati datu datoteku koristeći prečicu *Shift+Space*.
5. Pokrenuti izvršavanje projektnog zadatka na DSP uređaju
6. Zaustaviti izvršenje programa upotrebom tačke prekida
7. Iscrtati vrednosti izlaznih kanala u vremenskom i frekventnom domenu kao i prenosnu karakteristiku filtera.
8. Ukloniti tačku prekida, i poslušati zvuk na levom i desnom izlaznom kanalu. Komentarisati razliku.

9. U toku izvršenja menjati šum pritiskom na taster *Mute* u okviru Audacity programa. Isprobati sve tri vrste šuma posebno, kao i kombinacije po 2 ili sva 3 šuma. Komentarisati rezultate.
10. U izvornom kodu promeniti vrednost koeficijenta adaptacije *lambda* sa 0.2 na 0.05. Ponovo pokrenuti projekat. Komentarisati razliku u odnosu na prethodno pokretanje.

## 2.2 Zadatak 2

U ovom zadatku potrebno je izvršiti modelovanje nepoznatog sistema korišćenjem adaptivnog filtera. Kao nepoznati sistem se koristi filter sa konačnim impulsnim odzivom 32-og reda. Aproksimacija se vrši koristeći adaptivni FIR filter 50-og reda.

1. Uvući projektni zadatak 1 u radni prostor
2. Unutar funkcije main izvršiti poziv funkcije *fir\_circular* nad levim ulaznim kanalom. Izlaz iz filtera smestiti u izlazni niz *OutputBufferL*. Koeficijenti filtera dati su u zaglavlju *lowpass\_coeff.h*
3. Pozvati funkciju *lms\_filter*. Ulazni signal je isti kao i u prethodnom koraku, a izlaz iz FIR filtra koristi se kao referentni signal. Koeficijent adaptacije *lambda* postaviti na 0,2 (6553).
4. Povezati *line out* izlaz vašeg računara na *line in* ulaz na razvojnoj ploči.
5. Koristeći program *Audacity* reprodukovati beli šum na oba kanala.
6. Pokrenuti program.
7. Iscrtati impulsni odziv i prenosnu karakteristiku oba filtera (vrednost koeficijenata i FFT nad koeficijentima).
8. Postaviti tačku prekida i posmatrati iscrtane vrednosti nakon svake iteracije.
9. Promeniti koeficijent *lambda* na 0,1. Ponoviti korake 5-8. Komentarisati uticaj koeficijenta adaptacije na brzinu adaptacije i stabilnost sistema.
10. Promeniti red adaptivnog filtra na 10. Ponoviti korake 5-8. Komentarisati uticaj reda filtra na brzinu adaptacije i stabilnost sistema.

## 2.3 Zadatak 3

Kao i u prethodnom zadatku potrebno je izvršiti modelovanje nepoznatog sistema korišćenjem adaptivnog filtera. Kao nepoznati sistem se koristi u filter sa beskonačnim impulsnim odzivom 4-og reda. Aproksimacija se vrši koristeći adaptivni FIR filter 32-og reda.

1. Uvući projektni zadatak 1 u radni prostor
2. Unutar funkcije main izvršiti poziv funkcije **fourth\_order\_IIR** nad levim ulaznim kanalom. Izlaz iz filtera smestiti u izlazni niz *OutputBufferL*. Koeficijenti filtera dati su u zaglavlju *IIR\_lowpass\_coeff.h*
3. Pozvati funkciju *lms\_filter*. Ulazni signal je isti kao i u prethodnom koraku, a izlaz iz IIR filtra koristi se kao referentni signal. Koeficijent adaptacije *lambda* postaviti na 0,2.
4. Povezati *line out* izlaz vašeg računara na *line in* ulaz na razvojnoj ploči.
5. Koristeći program *Audacity* reprodukovati beli šum na oba kanala.
6. Pokrenuti program.
7. Iscrtati impulsni odziv i prenosnu karakteristiku oba filtera.
8. Postaviti tačku prekida i posmatrati iscrtane vrednosti nakon svake iteracije.

### 3 Zaključak

U okviru ove vežbe upoznali ste se sa strukturom adaptivnih digitalnih sistema i načinom funkcionisanja adaptivnih digitalnih sistema. Upoznali ste se načinom adaptacije prenosne karakteristike adaptivnih digitalnih sistema. Takođe, upoznali ste se sa nekim od primena adaptivnih digitalnih sistema. Naučili ste kako se primenjuju adaptivni digitalni sistemi za uklanjanje šuma. Evaluirali ste uticaj koeficijenta adaptacije na brzinu adaptacije adaptivnog digitalnog sistema. Takođe ste evaluirali uticaj dužine adaptivnog filtra na kvalitet adaptacije praćenjem uticaja dužine adaptivnog filtra na kvalitet otklanjanja šuma iz signala.