*XXIII Simpozijum o novim tehnologijama u poštanskom i telekomunikacionom saobraćaju – **PosTel 2005**, Beograd, 13. i 14. decembar 2005.*

OBRADA SIGNALA I ANALIZA TELEKOMUNIKACIONIH SISTEMA KORIŠĆENJEM ALGEBARSKIH RAČUNARSKIH SISTEMA

Miroslav Lutovac
IRITEL, Beograd

Sadržaj: *Uloga i značaj simboličkog računanja objašnjeni su na primerima telekomunikacionih sistema. Prikazane su aplikacije iz prakse u kojima su sistemi simbolički rešeni i simulirani. Originalan pristup razvoju algoritama, projektovanju sistema i simboličkog procesiranja uspešno su rešili probleme koji su uočeni u klasičnom pristupu i numeričkim računarskim sistemima.*

Ključne reči: *simboličko procesiranje, predstavljanje signala, linearni sistemi, nelinearni sistemi, računarski alati*

1. Uvod

Uređaji koji se koriste u poštanskom i telekomunikacionom saobraćaju postaju sve složeniji pa je za njihovu analizu i projektovanje potrebno ekspertsko znanje iz više oblasti kao što su teorija sistema, obrade signala, telekomunikacije i softverskog inženjerstva. Klasični pristupi zasnovani na numeričkim izračunavanjima sve teže mogu da zadovolje potrebe uređaja i sistema koji se razvijaju i pojavljuju u poštanskom i telekomunikacionom saobraćaju. Zbog toga je interesovanje za korišćenje algebarskih računarskih sistema u obradi signala i analizi telekomunikacionih sistema značajno poraslo poslednjih godina. U ovom radu koristićemo skraćenicu CA za algebarske računarske sisteme (Computer Algebra System). Brojni softverski alati i programski paketi na osnovu algebarskih algoritama sve više se koriste u razvoju i analizi telekomunikacionih sistema od kojih su najznačajniji *Mathematica*, Maple i MATLAB Symbolic Toolbox [1, 2]. CA sistemi imaju značajan uticaj na edukaciju i obrazovanje i sve više će se koristiti u praksi zbog problema numeričkih alata sa tačnošću i značajnog povećanja složenosti analiziranih problema a pre svega zato što je analizirani prostor u suštini kontinualan i beskonačan [3, 4].

Ovaj rad ima za cilj da ukaže na ulogu primene CA sistema i da ukaže na trendove daljeg razvoja ali i na probleme koji su pratili razvoj ovih sistema. Kroz jednostavne ilustrativne primere biće pokazano kako se koriste algebarski sistemi i koje su to značajne prednosti u odnosu na sada već klasične numeričke računarske sisteme.

Prvi programi CA sistema pojavili su se još 1990. godine ali većina njih nije doživela ekspanziju kao numerički sistemi. Osnovni razlog je bio da su mogli da se reše samo jednostavniji praktični problemi, vremena izračunavanja su bila veoma duga a često je postojao i problem prikaza rezultata pre svega zbog ograničenog memorijskog prostora. Poslednjih nekoliko godina došlo je do značajnog povećanja procesorske snage računara, brzina rada i memorijski prostor su višestruko povećani a i sami algebarski programi su znatno efikasniji nego ranije tako da je već sada moguće rešiti složenije praktične probleme. Iako se sada već standardni numerički programski paketi, kao što je MATLAB, dominantno koriste u edukaciji i inženjerskoj praksi mogu se uočiti mnogi problemi koji se mogu rešiti jedino korišćenjem simboličkih programskih paketa.

CA sistemi rade sa simbolima a brojevi su samo jedna uža grupa simbola nad kojima se izvršavaju neke operacije kao što su operacije sabiranja i množenja. Stoga se pisanje programa u CA sistemima može posmatrati kao set instrukcija koji manipuliše sa simbolima i mogu se koristiti za izvršavanje mnogo šireg kruga poslova. Najjednostavniji primer razlike u radu numeričkog i algebarskog računarskog sistema je rešenje problema integraljenja signala sinusnog oblika. Naime, kod numeričkih sistema se najpre izračunaju numeričke vrednosti sa određenim korakom nezavisno promenljive, a zatim se prostim sabiranjem može naći signal koji odgovara integralu sinusnog signala. Sa malim brojem numeričkih vrednosti ne može se uočiti da se integraljenjem sinusnog signala dobija kosinusni signal, a kod velikog broja numeričkih vrednosti može doći do značajne greške usled akumulacije zbog predstavljanja brojeva sa konačnim brojem cifara. U slučaju korišćenja CA sistema, početni oblik signala je simbol sinusa, operacija koja se izvršava predstavlja se simbolom integrala a rezultat obrade je simbol kosinusa. Bez obzira na opseg nezavisno promenljive u kome se posmatra signal, rezultat obrade je uvek simbol koji se odmah prepoznaje i koji nije podložan grešci obrade. Što su analizirani sistemi i signali složeniji to je mogućnost greške sa numeričkim računarskim sistemima veća pa efikasnost CA sistema postaje značajnija.

U algebarskim računarskim sistemima se mogu definisati osobine signala koji se predstavljaju simbolima kao i osobine sistema koji se predstavljaju definisanjem simboličkih operacija nad signalima. Ako su signali predstavljeni brojevima, a operacije sistema se predstave skupom operacija sabiranja i množenja, tada će CA sistemi raditi isto što i numerički. Međutim, ako se analizirani sistem predstavi sistemom diferencnih jednačina, a pobudni signali su predstavljeni kao funkcije neke promenljive, tada rezultati obrade mogu takodje da se predstave kao funkcije. To znači da CA sistemi mogu da transformišu signal predstavljen u jednom obliku u neki drugi oblik na sličan način kao što se u klasičnoj matematici izvode dokazi teorema ili analiziraju osobine funkcija. I umesto da sami analiziramo osobine nekog sistema tako što veliki broj puta prepisujemo neke formule i postupno menjamo njihov oblik, kao kada izvodimo dokaz neke teoreme, računar je sposoban da automatski obavi potrebne simboličke operacije sve dok se ne dobije željeni oblik. To znači da su CA sistemi neka vrsta intelektualnih pojačavača, odnosno oni imaju mogućnost da povećaju naše mogućnosti razmišljanja i zaključivanja i da, na primer, u deliću sekunde nađu integral neke složene funkcije kao što digitron može da pomnoži dva broja predstavljena velikim brojem cifara. Pri ovome treba biti oprezan, jer kao što ne postoje algebarska rešenja za sve integrale, ni CA sistemi neće moći da reše sve probleme. Stoga se programiranje svodi na pisanje programa za komunikaciju između korisnika i računara kojim se računaru određuje koje akumulirano znanje treba da koristi, kao i kada i kako da primenjuje to znanje u rešavanju praktičnih problema.

Iako na prvi pogled izgleda da je korišćenje CA sistema veoma komplikovano, a što je takođe jedan od razloga da nisu tako popularni kao numerički sistemi, većina korisnika će najčešće koristiti specijalizovane pakete ili samo manji broj simboličkih funkcija. Specijalizovani programi namenjeni za korišćenje algebarskih računarskih sistema imaju grafički korisnički interfejs tako da se pojedine operacije izvršavaju jednostavnim aktiviranjem ikona na sličan način kao što se koristi standardni programi za crtanje ili obradu teksta. Svaka komanda takođe ima odgovarajući pomoćni tekst sa skraćenim uputstvom za korišćenje ali i obimno uputstvo za one koji žele da iskoriste i neke specifične osobine simboličkih funkcija.

Kroz jednostavne praktične primere biće pokazana efikasnost CA sistema. Na primer, korišćenjem palete sa grafičkim simbolima diskretnog sistema može da se nacrta blok dijagram analiziranog sistema, na primer sa više ulaznih i izlaznih portova, da se pojedinim blokovima definišu parametri ili simboličke vrednosti, a da se zatim automatski dobiju sve funkcije prenosa od svih ulaza ka svim izlazima. Iz funkcije prenosa se može odrediti odziv u vremenskom domenu za proizvoljne pobudne signale. U nekim složenijim slučajevima, kao što su višestepeni filtri za visoke brzine obrade, funkcije prenosa se teško mogu izračunati klasičnim metodama, dok se korišćenjem CA sistema za kratko vreme dobijaju tačna rešenja. Dobijeni rezultati se mogu koristiti za dalju analizu ovih filtara kao što je na primer izvođenje dokaza komplementarnosti po snazi. Na sličan način se mogu upoređivati različiti sistemi i pokazati u čemu se razlikuju ili dokazati da imaju istu funkcionalnost. Simboličko procesiranje je posebna karakteristika ovih sistema – a to je da se pobudni signali mogu predstaviti ne samo brojevima već i simbolima i da se analizira šta se dešava sa njima tokom obrade. Postavljanjem dodatnih uslova o željenom odzivu, mogu se naknadno odrediti parametri analiziranog sistema pa se time analiza koristi i za projektovanje sistema.

Sistemi se mogu analizirati ne samo kao linearizovani modeli već se mogu odrediti funkcije odziva i za nelinearne sisteme pa čak i za neke numeričke algoritme kao što je primer modifikovanog Njutn-Rapsonov metod aproksimacije recipročne vrednosti. Umesto određivanja granica greške i manje-više dobrih procena tačnosti rada nekog algorita, u nekim slučajevima se mogu izvesti funkcije greške u zatvorenom obliku. Ovako dobijeni rezultati se zatim mogu koristiti i za izbor pojedinih parametara na primer izbor početnog rešenje koje obezbeđuje najbržu konvergenciju numeričkog algoritma za željeni opseg nezavisno promenljivih.

Primena CA sistema omogućava da se posvetimo postavci problema, na primer kako postaviti sistem jednačina koji rešava neki problem, a da se nekreativan posao nalaženja rešenja prepusti računaru, kao što su na primer rešavanje sistema jednačina ili integrala složene funkcije. Ovi sistemi treba da služe kao mašine koje imaju mogućnost matematičkog razmišljanja i zaključivanja sa ciljem da nas oslobode nekreativnog posla a ne da se bavimo matematičkim aparatom. Vrtoglavi razvoj uređaji koji se koriste u poštanskom i telekomunikacionom saobraćaju zahteva da i sami budemo eksperti iz više različitih oblasti a to je moguće samo ako znamo da koristimo moćne računarske sisteme.

2. Predstavljanje brojeva u numeričkim i algebarskim računarskim sistemima

Većina računarskih sistema predstavlja brojeve prema IEEE standardu 754 u dvostrukoj tačnosti sa 64 bita. Mnogi brojevi se ne mogu predstaviti tačno a različiti sistemi će dati različite rezultate iako se koriste, na prvi pogled, iste komande.

Predpostavimo da smo zadali numeričke vrednosti `a=0.2` i `b=1-0.8`. Očekujemo da je razlika brojeva `a` i `b` jednaka nuli, ali u programima MATLAB i *Mathematica* dobijamo `a-b=5.5511`$\times 10^{-17}$. Razlog za ovaj iznenađujući rezultat se lako pronalazi ako znamo kako numerički računarski sistemi predstavljaju ova dva broja. Na primer, MATLAB komanda `sym(a,'d')` pokazuje da je broj `a` predstavljen kao broj `0.20000000000000001110223024625157` a ne kao `0.20000000000000`. U binarnom brojnom sistemu broj 0.2 ne može da se predstavit tačno sa konačnim brojem bita. Iako u komandnom prozoru MATLAB prikazuje da je `b=0.20000000000000`, komanda `sym(b,'d')` daje `0.19999999999999995559107901499374` i to je broj koji se koristi u izračunavanjima. Suprotno našim očekivanjima da su brojevi `a` i `b` isti oni su predstavljeni na različite načine pa je i razlika ova dva broja različita od 0.

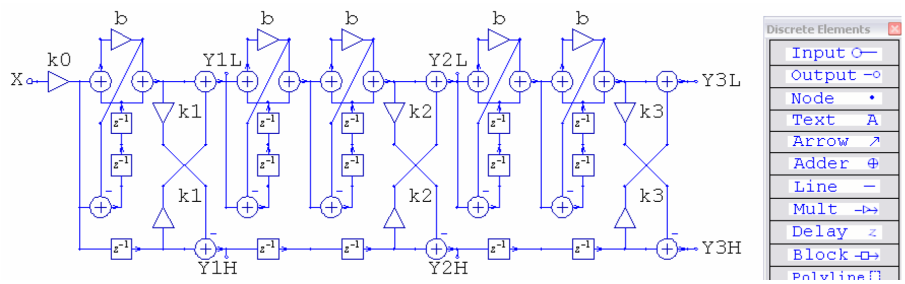
Iako razlika brojeva `a` i `b` nije jednaka nuli, komanda `==` u programu *Mathematica*, koja testira da li su leva i desna strana u izrazu `a==b` jednake, daje rezultat `True`, odnosno pokazuje da su brojevi `a` i `b` jednaki. Ista komanda u programu MATLAB daje rezultat 0, odnosno pokazuje da su brojevi `a` i `b` različiti. Naizgled ista komanda, koja se simbolički predstavlja dvostrukim znakom jednako, ==, daje različite rezultate u dva matematička programa.

Analizirani primer pokazuje da numerički sistemi izvršavaju opracije sa brojevima koji se razlikuju od onih koje prikazuju, i da generalno ne znamo tačno sa kojim brojevima programi rade. Tako na primer u programu MATLAB, komanda `sign(sin(2*pi))` daje `-1`, a komanda `sign(sin(3*pi))` daje `+1`, iako je u oba slučaja tačna vrednost jednaka 0.

Algebarski računarski sistemi rade sa tačnim vrednostima. Tako na primer komande `simplify(sym('sin(2*pi)'))` i `simplify(sym('sin(3*pi)'))` daju istu tačnu vrednost 0, pa je i `sign(0)=0`. Broj može da se predstavi kao racionalan broj; dodela broja može da se uradu simbolički `a=sym(1/5)` i `b=sym(1-4/5)` i tada oba broja imaju istu vrednost `a=1/5`, i `b=1/5`. U oba slučaja, brojevi `pi=`π i `1/5` nisu aproksimativne vrednosti već simboli koji sadrže tačne vrednosti. Na primer, komanda `a=sym(sin(pi/4))` daje tačan rezultat `a=sqrt(1/2)`, za razliku od numeričkog izračunavanja koji daje rezultat sa približno 15 tačnih decimalnih cifara `sin(pi/4)= 0.70710678118655`.

3. Predstavljanje sistema

Klasična analiza sistema opisuje sisteme i delove sistema matematičkim formulama. Ovakav pristup je postao nepraktičan u slučajevima analize i projektovanja složenijih sistema. Grafičko predstavljanje korišćenjem blok dijagrama omogućava bolji uvid u procese koji se odvijaju u nekom sistemu. MATLAB i Simulink pružaju jednostavan grafički interfejs ali procesiranje kontinualnih i vremenski diskretnih sistema može da se izvršava samo numerički a bez mogućnosti klasičnih teorijskih transformacija koje pojedine operacije mogu da predstave formulama. Stoga se uobičajeno sistem opisuje matematičkim formulama, slike sistema se crtaju korišćenjem programa za crtanje, procesiranje se izvršava numeričkim sistemom, a za bolji uvid u osobine procesiranih signala koriste se diskretne transformacije. Osim grešaka u predstavljanju brojeva sa konačnim brojem cifara, svaka od konverzija može dodatno da unese grešku, a često se dešava da je na slikama prikazan jedan sistem a da se analizira sasvim drugi sistem. Zbog toga su razvijeni posebni aplikativni programi koji omogućava jednostavan korisnički grafički interfejs (GUI – Graphical User Interface) za crtanje kontinualnih i diskretnih linearnih i nelinearnih sistema i automatsku analizu i obradu koriććenjem algebarskih računarskih sistema: DSSYM u programu MATLAB i *SchematicSolver* u programu *Mathematica* [5, 6]. Na slici 1 je primer sistema koji je nacrtan korišćenjem palete za crtanje koja je prikazana u desnom delu slike.



Slika 1. *Šema diskretnog sistema i paleta za GUI*

Postupak crtanja sistema započinje pokretanjem osnovnog algebarskog računarskog sistema, u ovom slučaju programa *Mathematica* i učitavanjem znanja iz aplikativnog programa *SchematicSolver* sa **Needs["SchematicSolver`"]**. Za crtanje se može koristiti jedna od većeg broja paleta, na primer paleta **Discrete Elements**, tako što se mišem klikne na polje **Start Drawing**. U dokumentu u kome se radi obrada, otvara se polje za crtanje. Korišćenjem polja u paleti sa imenima elemenata sistema može da se nacrta sistem prikazan na slici 1. Istovremeno, program generiše listu koja opisuje sistem a koja se sastoji od većeg broja listi od kojih svaka opisuje pojedine elemente sistema. Elementi sistema se opisuju listom koja sadrzi ime elementa, parove koordinata preko kojih se povezuje sa ostalim elementima, dodatne parametre neophodne za analizu i obradu kao i informacije neophodne za bolji vizuelni izgled šeme sistema – boja i debljina linija, tip i boja slova, veličina grafičkih elemenata i slova. Nakon što je nacrtana šema sa slike 1, deo liste koji opisuje sistem sastoji se od sledećih elemenata:

```
hsSystem = {
 {"Multiplier", {{6, 0}, {6, 3}}, k1},
 {"Delay", {{4, 5}, {4, 7}}, 1},
 {"Adder",{{7,8},{8,5},{9,8},{8,9}},{1,1,2,0}},
 {"Input", {0, 8}, "X"},
 {"Output", {9, 8}, "Y1L"},
...,
 {"Line", {{6, 8}, {7, 8}}} }
```

Prvi element u opisu sistema je množač koji je povezan sa ostalim elementima sistema u koordinatama `{{6,0},{6,3}}`, a vrednost koeficijenta množenja je `k1`. Ime sistema je hsSystem i ono sadrži opis potreban i za crtanje i za dalju obradu i analizu. Tako na primer, ako želimo da nacrtamo sistem, potrebno je da se izvrši komanda za crtanje sistema **ShowSchematic[hsSystem]** koja otvara novu ćeliju u dokumentu sa slikom 1.

4. Automatsko crtanje

U praksi se često koriste sistemi koji imaju sličnu strukturu ili više istih podsistema koji se ponavljaju. Na primer na slici 1 može da se uoči da je struktura od čvora Y1L do Y2L identična strukturi od čvora Y2L do Y3L. Sistem višeg reda imao bi više ovakvih kaskadno vezanih podsistema. U ovakvim slučajevima moguće je nacrtati jedan podsistem a zatim višestrukim kaskadnim povezivanjem dobiti složeniju šemu. Ceo postupak može da se automatizuje i time karakteristične šeme postaju deo znanja softverskog paketa. Za primer sistema sa slike 1 dovoljno je da se odredi broj kaskadnih sekcija: numSek=3. Zatim se automatski generišu simboličke vrednosti koeficijenata k123=UnitSymbolicSequence[numSek+1,k,0]. Simbolička vrednost koeficijenata rekurzivnih grana se dodaje listi sa koeficijentima p=Join[{b},k123]//Flatten. Šema celog sistema se generiše funkcijom za automatsko generisanje šema filtara za velike brzine na osnovu IIR i FIR filtara {hsSystem,inpCoords,outCoords} = HighSpeedIIR3FIRHalfbandFilterSchematic[parameterSymbols]. Nakon što se sistemu dodaju elementi koji označavaju ulazni i izlazne čvorove dobija se lista sa imenom hsSystem. Komanda ShowSchematic[hsSystem] će nacrtati sliku 1.

Funkcije za automatsko crtanje su omogućile da se dobije blok dijagram sistema i tekstualna lista sa opisom sistema na osnovu zadatog broja podsistema (3), simbola u direktnim granama (k) i simbola rekurzivne grane (b).

5. Analiza sistema

Numerički računarski sistemi ne mogu da generišu funkciju prenosa koja je važna za određivanje frekvencijske karakteristike linearnog sistema. Za poznati pobudni signal može se izračunati odziv sistema u zatvorenoj formi tako što se (1) nacrta blok šema sistema, (2) označe čvorovi celim brojevima, (3) pobudni signal izrazi u zatvorenoj formi, (4) označi izlazni čvor, (5) napišu izrazi u vremenskom domenu za svaki blok sistema, (6) primeni *z* transformacija na svaku jednačinu, (7) reši algebarski sistem jednačina u *z* domenu, (8) odredi odziv u vremenskom domenu korišćenjem inverzne *z* transformacije. Proizvod *z* transformacije pobudnog signala i funkcje prenosa daju *z* transformaciju odziva.

SchematicSolver omogućava da se automatski odredi funkcija prenosa linearnog sistema. Komanda tf=DiscreteSystemTransferFunction[hsSystem] izračunava funkcije prenosa svih izlaza u odnosu na ulaze i to direktno iz opisa sistema. Sistem hsSystem ima 6 izlaza i jedan ulaz tako da se dobija matrica kolona sa 6 funkcija prenosa. Na primer, jedna od funkcija prenosa h3L=tf[[5,1]] od ulaza X do izlaza Y3L nalazi se u petom redu i prvoj koloni. Predpostavimo da želimo da nađemo impulsni odziv u zatvorenom obliku za poznate numeričke vrednosti parametara sistema p={b→9/16, k0→0.24000685, k1→2.37428, k2→-0.54068, k3→0.1093268}. Komanda y=InverseZTransform[h3L /. p, z, n] izračunava odziv u zatvorenom obliku. Za n>6 odziv sistema ima sledeći oblik:

Out[70]=
$$\begin{aligned}
&0.75^{n}\Big(-0.0016568\,(-23.7812+n)\,(-10.6931+n)\,(-5.2192+n)\\
&\quad(1.17916+n)\,\mathrm{Cos}\Big[\frac{n\,\pi}{2}\Big]+4.33681\times10^{-18}\,(-14.7861+n)\\
&\quad(-7.62975+n)\,(-3.69273+n)\,(7.96133\times10^{15}+n)\,\mathrm{Sin}\Big[\frac{n\,\pi}{2}\Big]\Big)
\end{aligned}$$

Funkcija prenosa `h3L` u funkciji simboličkih vrednosti parametara je znatno složenija:

Out[22]=
$$\begin{aligned}
&\frac{1}{z^{5}\,(b+z^{2})^{5}}\\
&(k0\,(b^{5}\,k3-b^{4}\,k1\,k3\,z+b^{3}\,k2\,z^{2}+5\,b^{4}\,k3\,z^{2}-b^{3}\,k1\,k2\,k3\,z^{2}-\\
&\quad b^{2}\,k1\,k2\,z^{3}-4\,b^{3}\,k1\,k3\,z^{3}-b^{5}\,k1\,k3\,z^{3}-b^{2}\,k2\,k3\,z^{3}+b\,k1\,z^{4}+\\
&\quad 3\,b^{2}\,k2\,z^{4}+2\,b^{4}\,k2\,z^{4}+10\,b^{3}\,k3\,z^{4}-3\,b^{2}\,k1\,k2\,k3\,z^{4}-\\
&\quad 2\,b^{4}\,k1\,k2\,k3\,z^{4}+z^{5}-2\,b\,k1\,k2\,z^{5}-3\,b^{3}\,k1\,k2\,z^{5}-6\,b^{2}\,k1\,k3\,z^{5}-\\
&\quad 4\,b^{4}\,k1\,k3\,z^{5}-2\,b\,k2\,k3\,z^{5}-3\,b^{3}\,k2\,k3\,z^{5}+k1\,z^{6}+4\,b^{2}\,k1\,z^{6}+\\
&\quad 3\,b\,k2\,z^{6}+6\,b^{3}\,k2\,z^{6}+b^{5}\,k2\,z^{6}+10\,b^{2}\,k3\,z^{6}-3\,b\,k1\,k2\,k3\,z^{6}-\\
&\quad 6\,b^{3}\,k1\,k2\,k3\,z^{6}-b^{5}\,k1\,k2\,k3\,z^{6}+5\,b\,z^{7}-k1\,k2\,z^{7}-6\,b^{2}\,k1\,k2\,z^{7}-\\
&\quad 3\,b^{4}\,k1\,k2\,z^{7}-4\,b\,k1\,k3\,z^{7}-6\,b^{3}\,k1\,k3\,z^{7}-k2\,k3\,z^{7}-\\
&\quad 6\,b^{2}\,k2\,k3\,z^{7}-3\,b^{4}\,k2\,k3\,z^{7}+4\,b\,k1\,z^{8}+6\,b^{3}\,k1\,z^{8}+k2\,z^{8}+\\
&\quad 6\,b^{2}\,k2\,z^{8}+3\,b^{4}\,k2\,z^{8}+5\,b\,k3\,z^{8}-k1\,k2\,k3\,z^{8}-6\,b^{2}\,k1\,k2\,k3\,z^{8}-\\
&\quad 3\,b^{4}\,k1\,k2\,k3\,z^{8}+10\,b^{2}\,z^{9}-3\,b\,k1\,k2\,z^{9}-6\,b^{3}\,k1\,k2\,z^{9}-\\
&\quad b^{5}\,k1\,k2\,z^{9}-k1\,k3\,z^{9}-4\,b^{2}\,k1\,k3\,z^{9}-3\,b\,k2\,k3\,z^{9}-6\,b^{3}\,k2\,k3\,z^{9}-\\
&\quad b^{5}\,k2\,k3\,z^{9}+6\,b^{2}\,k1\,z^{10}+4\,b^{4}\,k1\,z^{10}+2\,b\,k2\,z^{10}+3\,b^{3}\,k2\,z^{10}+\\
&\quad k3\,z^{10}-2\,b\,k1\,k2\,k3\,z^{10}-3\,b^{3}\,k1\,k2\,k3\,z^{10}+10\,b^{3}\,z^{11}-\\
&\quad 3\,b^{2}\,k1\,k2\,z^{11}-2\,b^{4}\,k1\,k2\,z^{11}-b\,k1\,k3\,z^{11}-3\,b^{2}\,k2\,k3\,z^{11}-\\
&\quad 2\,b^{4}\,k2\,k3\,z^{11}+4\,b^{3}\,k1\,z^{12}+b^{5}\,k1\,z^{12}+b^{2}\,k2\,z^{12}-b^{2}\,k1\,k2\,k3\,z^{12}+\\
&\quad 5\,b^{4}\,z^{13}-b^{3}\,k1\,k2\,z^{13}-b^{3}\,k2\,k3\,z^{13}+b^{4}\,k1\,z^{14}+b^{5}\,z^{15}))
\end{aligned}$$

Ručno izvođenje funkcije prenosa ili impulsnog odziva iz analiziranog primera predstavlja ozbiljan problem i za dobro uvežbanog istraživača, dok numerički računarski sistemi ne mogu da pomognu u izvođenju. Stoga je veoma važno da se u edukaciji studenti obuče za postavljanje problema i korišćenje CA sistema u njihovom rešavanju kako bi u praksi mogli efikasno da rešavaju tehničke probleme.

6. Izvođenje novih i dokazivanje poznatih osobina sistema

Poznato je da sistemi sa slike 1 imaju osobinu komplementarnosti po snazi [7], odnosno da je zbir snaga na odgovarajućim izlazima konstantan i da ne zavisi od učestanosti. Da bi dokazali ovu osobinu potrebno je da se pokaže da važi sledeći izraz:

$$H_{3L}(z)H_{3L}(z^{-1})+H_{3H}(z)H_{3H}(z^{-1})=1 \tag{1}$$

Samo izvođenje funkcije prenosa je veoma komplikovano a potrebno je dodatno množenje i sabiranje četiri složena izraza. Numeričkim računarskim sistemima je moguće proveriti tačnost ali postoji odstupanje od tačne vrednosti za koju je teško odrediti da li je posledica konačne tačnosti predstavljanja brojeva ili postoji drugi razlog. Sa CA sistemom se dokazuje primenom samo jedne komande:

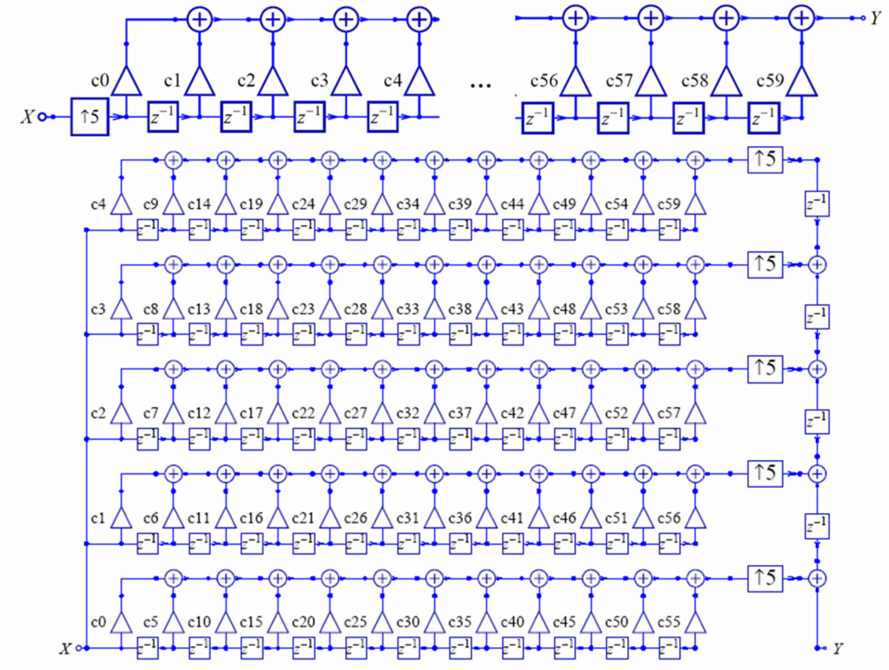
```
tfSum2 = H1*(H1/. z->1/z) + H2*(H2/. z->1/z) //FullSimplify
```

što daje rezultat `tfSum2 = 2 k0`^{2}`(1+k1`^{2}`)(1+k2`^{2}`)(1+k3`^{2}`)`. Rezultat pokazuje da izraz ne zavisi od `z`, odnosno od učestanosti, ali da ne zavisi ni od koeficijenta u rekurzivnoj grani. Da bi izraz bio jednak 1 potrebno je da postoji tačno određena relacija između koeficijenata `k0`, `k1`, `k2` i `k3`, i sada je lako utvrditi razlog odstupanja dobijen numeričkim postupcima. Iz uslova da je `tfSum2 = 1` lako se izvodi koju vrednost treba da ima `k0` u funkciji ostalih poznatih koeficijenata. Ovim postupkom smo dokazali poznatu osobinu analiziranog sistema.

Ostali koeficijenti se mogu odrediti primenom teorije aproksimacije. Međutim, poznato je da `h3L` ima nultu vrednost za `z=-1`, što se može iskazati na sledeći način: `num3 = Numerator[h3L//Together]/. z→-1// Factor.` Dobija se `num3 = -(1+b)`^{5} `k0(1-k1-k2-k3-k1 k2-k1 k3-k2 k3+k1 k2 k3)=0`. Samo treći činioc u zagradi može da bude jednak nuli i odatle se izvodi nova relacija koja pokazuje vrednost za `k2` u funkciji poznatih `k1` i `k3`. Ovim postupkom smo izveli novu osobinu koja kazuje da `k2` zavisi od vrednosti preostala dva koeficijenta čije vrednosti tek treba da odredimo a da pri tome nismo koristili ni jednu poznatu aproksimacionu funkciju.

7. Sistemi sa višestrukim odabiranjem

Svaki sistem može da se realizuje na više načina. Najjednosatvniji način da se utvrdi da dva linearna sistema imaju istu funkcionalnost jeste da se uporede njihove funkcije prenosa. Međutim, kod mnogih sistema koji nisu linearni, a kao što su sistemi sa višestrukim odabiranjem, to nije moguće. CA sistemi omogućavaju da se izvrši obrada simboličkog signala sa sistemima čiji su parametri takođe definisani simbolima, na primer prikazanim na slici 2, i da se zatim uporede njihovi odzivi u vremenskom domenu.

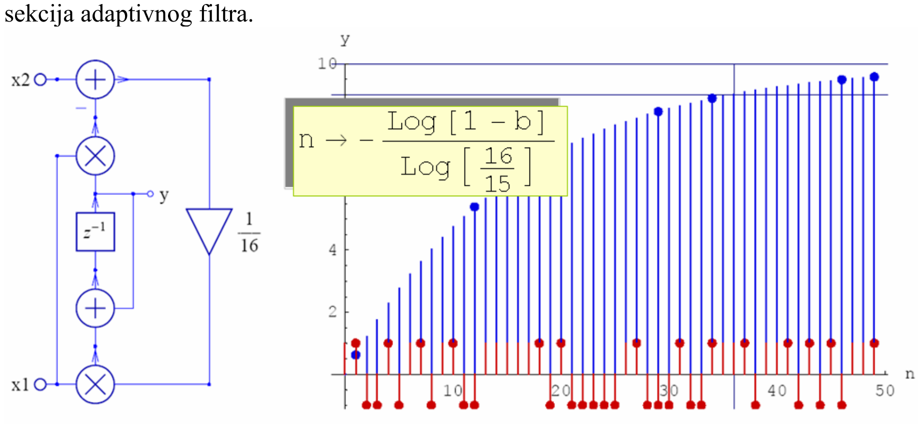


Slika 2. *Dva sistema sa višestrukim odabiranjem*

Ako sa `out1` označimo izlaz jednog sistema a sa `out2` izlaz drugog sistema, tada komanda `SameQ[out1,out2]` upoređuje odzive ta dva sistema i generiše `True` ako su sistemi identični.

8. Nelinearni sistemi

Odziv linearnog sistema može da se odredi u zatvornoj formi korišćenjem transformacija. Za nelinearne sisteme neophodno je da se sve jednačine postave u vremenskom domenu. Na slici 3, u levom delu, prikazan je nelinearni sistem – jedna sekcija adaptivnog filtra.



Slika 3. *Nelinearni sistem*

Nakon što se nacrta šema sistema, izvrši se komanda za generisanje implementacionog koda `DiscreteSystemImplementation[system, "implement"]`. Implementacioni kod se primeni za dve uzastopne simboličke ulazne vrednosti i primenom komande `Reduce` eliminišu prethodna stanja memorijskog bloka, tako da se dobija rekurzivna formula opšteg člana izlaznog signala. Komanda `RSolve` može da generiše opšti član izlaznog signala. Ovime je omogućeno da se uradi optimizacija; na primer da se odredi redni broj odbirka `n` kada izlazni signal dostigne vrednost `b`. Na kraju, verifikacija je urađena za slučajni pobudni signal. Sličan postupak se može primeniti i za nelinearne algoritme kao što je modifikovanog Njutn-Rapsonov metod aproksimacije recipročne vrednosti, stim da se najpre algoritam predstavi blok šemom.

9. Telekomunikacioni sistemi

CA sistemi omogućavaju da se analiziraju telekomunikacioni sistemi i veliki broj primera je analiziran u [6]. Diskretni analitički, realni i kompleksni signal analizirani su na primeru Hilbertovog transformatora. Pokazano je da je realizacija kvadraturne amplitudske modulacije efikasnija primenom Hilbertovim transformatorom u odnosu na klasičnu realizaciju. Vizuelizacija procesa obrade i simboličko procesiranje prikazani su na primerima adaptivnog LMS algoritma (Least Mean Squares), sistema za automatsku regulaciju pojačanja (AGC – Automatic Gain Control) , kvadraturni detektor anvelope, i nelinearni sistem automatske kontrole sa histerezisom.

10. Zaključci

Klasičan teorijski pristup analize telekomunikacionih sistema zasniva se na predstavljanju signala kao funkcije, a sistema kao operatora od kojih se sastoji blok dijagram sistema. Algebarski računarski sistemi omogućavaju da se na prirodan način uradi analiza i procesiranje i to tako da nekreativan deo posla urade računarski sistemi. Simbolička analiza i procesiranje su tačni i nisu podložni greškama.

Algebarski računarski sistemi obezbeđuju jednostavan grafički interfejs tako da se mnoge složene operacije mogu izvršiti aktiviranjem grafičkih ikona. Blok šema sistema nije samo nepokretna slika, već ona sadrži sve potrebne elemente i za generisanje visokokvalitetnih šema ali i za rešavanje, procesiranje i simulaciju. Težište problema nije analiza numeričkih podataka ili višestruko ponavljanje numeričkih eksperimenata, već postavka problema i priprema podataka tako da rezultat analize i obrade signala dobije željeni oblik.

Literatura

[1] S. Wolfram, "*The Mathematica Book*", Cambridge: Cambridge University Press, Wolfram Media, 2003.

[2] C. Moler, and P. J. Costa, "*Symbolic Math Toolbox*", The Mathworks Inc, Natick, MA 01760-2098, 1996.

[3] M. D. Lutovac, D. V. Tošić, B. L. Evans, "*Filter Design for Signal Processing Using MATLAB and Mathematica*", Upper Saddle River: Prentice Hall, 2001.

[4] M. Lutovac, D. Tošić, "*Symbolic signal processing and system analysis*", Facta Universitatis, Electronics and Energetics, vol. 16, no. 3, 2003, pp. 423-431.

[5] M. Lutovac, D. Tošić, "*SchematicSolver*", Ver. 2, www.schematicsolver.com, 2004.

[6] M.D. Lutovac, D.V. Tošić, "*SchematicSolver*", Sample notebooks and documentation, http://library.wolfram.com/infocenter/TechNotes/4814/, 2003.

[7] M. D. Lutovac, D. V. Tošić, "*High-Speed Filter Design using Mathematica*", EUROCON 2005, vol. II, Belgrade, 2005, pp. 1626-1629.

Abstract: *Role and importance of symbolic computation in communication systems is exemplified. Real-life applications are presented in which systems are simulated and solved symbolically. Original approach to algorithm development, system design, and symbolic processing successfully overcomes problems encountered in the traditional approach and the numeric-only computer systems.*

Keywords: *symbolic processing, signal representation, linear systems, nonlinear systems, computer tools*

SIGNAL PROCESSING AND ANALYSIS OF COMMUNICATION SYSTEMS USING COMPUTER ALGEBRA SYSTEMS

Miroslav Lutovac