FAKULTET ORGANIZACIONIH NAUKA
UNIVERZITET U BEOGRADU

Simulacija u poslovnom odlučivanju

Skripta

Sanja Ilić

2018.

ПУШКИЦЕ
ФОН андерграунд

Ispitna pitanja

1. Modeliranje i modeli, vrste modela
2. Neformalni i formalni modeli
3. Modeliranje i simulacija
4. Karakteristike simulacionog modeliranja
5. Potreba za simulacijom
6. Mogućnosti primene simulacije, prednosti i nedostaci simulacije
7. Simulacioni proces
8. Podele i vrste simulacionih modela
9. Klasifikacije modela
10. Formalna specifikacija modela
11. Ocene parametara determinističkog i stohastičkog modela
12. Statistički pristup proceni parametara modela
13. Ocena nepoznatog parametra po metodi najmanjih kvadrata
14. Procena k nepoznatih parametara modela
15. Validacija simulacionih modela
16. Formalni kriterijum za utvrdivanje validnosti modela
17. Verifikacija simulacionih modela
18. Formalni opis sistema sa diskretnim dogadjajima
19. Događaj, aktivnost i proces
20. Razvoj simulacije diskretnih događaja, mehanizam pomaka vremena
21. Strategija rasporedivanja dogadaja u SDD
22. Strategija skaniranja aktivnosti u SDD
23. Strategija interakcije procesa u SDD
24. Pojam sistema: definicija, karakteristike i primeri sistema
25. Okruženje sistema ulazi i izlazi sistema
26. Ciljevi sistema: opstanak, rast i razvoj
27. Stanje sistema, prostor stanja sistema i oblast dopuštenih stanja sistema
28. Struktura sistema: pojam i karakteristike, adaptivni sistemi
29. Sistemsko razmišljanje i dinamika sistema
30. Povratno dejstvo, kolo povratnog dejstva, polaritet KPD
31. Karakteristike sistema sa i bez KPD
32. Pozitivno kolo povratnog dejstva: pojam, osobine, primeri
33. Negativno kolo povratnog dejstva: pojam, osobine, primeri
34. Dijagrami uzročno-posledičnih veza
35. Dijagrami skladišta i tokova
36. Računarski modeli u dinamici sistema
37. Modeliranje sistema u programskom jeziku SDS
38. Test funkcije: PULSE, RAMP, STEP
39. Kašnjenje: pojam, osobine, vrste
40. Modeliranje kašnjenja na materijalnim tokovima u SDS jeziku
41. Modeliranje kašnjenja na informacionim tokovima u SDS jeziku
42. Priraštaj: pojam, definisanje polariteta KPD preko priraštaja, primeri
43. Dinamika sistema, stabilni i nestabilni sistemi, stacionarno i prelazno ponašanje
44. Obrasci ponašanja sistema
45. Ponašanje sistema prvog reda sa +KPD i -KPD
46. Ponašanje sistema višeg reda sa +KPD i -KPD
47. Slučajni dogadaji, verovatnoća, slučajne promenljive

1. Modeliranje i modeli, vrste modela

Modeliranje izražava našu sposobnost da mislimo i zamišljamo, da koristimo simbole i jezike, da komuniciramo, da vršimo generalizacije na osnovu iskustva, da se suočavamo sa neočekivanim. Ono nam omogućava da uočavamo obrasce, da procenjujemo i predviđamo, da upravljamo procesima i objektima, da izlažemo značenje i svrhu.

U najširem smislu, modeliranje predstavlja isplativo (u smislu troškova) korišćenje nečega (model) umesto nečega drugog (realni sistem) sa ciljem da se dođe do određenog saznanja. Rezultat modeliranja je model. Model je apstrakcija realnosti u smislu da on ne može da obuhvati sve njene aspekte. Model je uprošćena i idealizovana slika realnosti. On nam omogućava da se suočimo sa realnim svetom (sistemom) na pojednostavljen način, izbegavajući njegovu kompleksnost, kao i sve opasnosti (u najširem smislu te reči) koje mogu proisteći iz eksperimenta nad samim realnim sistemom. Drugim rečima, model je opis realnog sistema sa svim onim karakteristikama koje su relevantne iz našeg ugla posmatranja. To znači da u procesu modeliranja moramo izvršiti izbor između onih elemenata i karakteristika sistema koje su od značaja za naše istraživanje i koje će biti obuhvaćene modelom i preostalih, za nas irelevantnih, koje naš model neće sadržati. Model ne sadrži samo objekte i atribute realnog sistema, već i određene pretpostavke o uslovima njegove validnosti. Cilj modela je da uobliči na vidljiv, često formalan način, ono što je suštinsko za razumevanje nekog aspekta njegove strukture ili ponašanja.

Nivo apstrakcije u procesu modeliranja utiče na validnost modela, odnosno na uspešnost predstavljanja realnog sistema modelom. Suviše složeni ili savršeni modeli koji imaju sposobnost da za isti skup ulaznih veličina proizvode iste izlazne vrednosti kao i realni sistemi, čak iako su ostvarivi, po pravilu su preskupi i neadekvatni za eksperimentisanje. S druge strane, suviše pojednostavljeni modeli ne odslikavaju na pravi način posmatrani sistem, a rezultati koji se dobijaju njihovom primenom mogu da budu neadekvatni i pogrešni. Stoga, u određenom trenutku treba povući granicu u realnom sistemu, i to tako da rezultujući model što vernije odslikava posmatrani sistem, ali da njegova cena i složenost ne budu ograničavajući faktori.

Vrste modela

Za predstavljanje sistema koriste se različiti modeli, kao što su:
1) mentalni (misaoni)
2) verbalni
3) strukturni ili konceptualni
4) fizički
5) analogni
6) matematički
7) simulacioni (računarski) i razni drugi modeli.

Često ih delimo i na:
• materijalne (model hemijske strukture molekula ili model aviona)
• simboličke modele (matematički,konceptualni, računarski, simulacioni idr).

1. Mentalni modeli su strukture koje ljudski mozak neprekidno konstruiše kako bi bio u stanju da poveže niz činjenica sa kojima se čovek susreće, a potom na osnovu toga deluje. Takvi modeli omogućuju, na primer, razumevanje fizičkog sveta, komunikaciju među ljudima i planiranje akcija. Iz ovog modela stvorili su se znatno komleksniji modeli kao posledica čovekove potrebe da objasni kompleksnije pojave u prirodi.

2. Verbalni modeli su direktna posledica mentalnih modela i predstavljaju njihov izraz u govornom jeziku, a uobičajeno se predstavljaju u pisanom obliku. Verbalni modeli spadaju u klasu neformalnih modela.

3. Fizički modeli predstavljaju umanjene modele realnog sistema, koji se ponašaju na isti način kao i njihovi originali. Uglavnom se prave na osnovu teorije sličnosti ili, u boljem slučaju, na osnovu fizičkih zakona sličnosti.

4. Matematički model je iz klase apstraktnih. Kod formulisanja matematičkog modela polazi se od verbalnog modela koji se transformisanjem dovodi u stanje koje se može opisati matematičkim jezikom. Takođe, ukoliko su veze između objekata modela opisane matematičkim (numeričkim) relacijama, tada se radi o matematičkim modelima. Ova klasa modela ima široku primenu, naročito u nauci i inženjerskim disciplinama.

5. Analogni model - Između dva fizička modela koji imaju iste matematičke modele, kaže se da postoji matematička analogija. Istovetnost matematičkih modela dvaju objekata pruža mogućnost da jedan od fizičkih objekata bude korišćen za analizu matematičkog modela drugog objekta. Fizički objekat koji se koristi za analizu matematičkog modela drugog objekta, a sa kojim ima isti ili sličan matematički model, naziva se analogni model. Prema tome, između fizičkog objekta koji se ispituje i analognog modela postoji matematička analogija.

6. Konceptualni modeli se stvaraju na osnovu predstave o strukturi i logici rada sistema ili problema koji se modelira i prikazuju se u obliku čije je značenje precizno definisano (na primer, dijagrami sa tačno definisanim simbolima). Oni predstavljaju ***osnovu za izradu računarskih modela***. Ovi se modeli često nazivaju i **strukturni**, pošto u grafičkom obliku ukazuju na strukturu posmatranog sistema. Takav prikaz omogućuje da se modeli vizualizuju i na taj način postaju zgodno sredstvo za komunikaciju među ljudima koji sa njima rade. Pored toga, grafički prikaz modela obezbeđuje relativno jednostavan prikaz složenih sistema i omogućuje bolje razumevanje sitema koji se modelira.

7. Računarski (simulacioni) modeli su prikaz konceptualnih modela u obliku programa za računar. U tom obliku modeli postaju sredstvo kojim se može efikasno analizirati rad modela u različitim spoljnim uslovima i sa različitim unutrašnjim parametrima i tako dobiti uvid u ponašanje sistema koji model opisuje. Kao sredstvo za izražavanje, računarski modeli koriste programske jezike.

2. Neformalni i formalni modeli

<u>**Neformalni opis modela**</u> daje osnovne pojmove o modelu i, mada se teži njegovoj potpunosti i preciznosti, on to najčešće nije. Prilikom izgradnje neformalnog opisa, radi eliminisanja pomenutih nedostataka, vrši se podela na **objekte**, **opisne promenljive** i **pravila interakcje objekata**.

Objekti su delovi iz kojih je model izgrađen, a **opisne promenljive** (preko vrednosti koje uzimaju) opisuju stanja u kojima se objekti nalaze u određenim vremenskim trenucima (parametri pomoću kojih se opisuju konstantne karakteristike modela su takođe uključeni u opisne promenljive). Postoje i **pravila interakcije objekata** koja definišu kako objekti modela utiču jedan na drugi u cilju promene njihovog stanja. Treba naglasiti da ne postoji pravilo za izbor objekata, opisnih promenljivih i pravila interakcije, već je taj izbor prepušten onome ko opisuje i izgrađuje model i treba ga vršiti tako da se dobije validan model. Neformalni opis modela priprema se dosta brzo i lako, ali on najčešće nije konzistentan i jasan, naročito kada su u pitanju složeni modeli.

Anomalije koje se javljaju prilikom neformalnog opisa modela najčešće se mogu opisati na sledeći način:

1. ***Nekompletan opis modela*** - Ukoliko model ne sadrži sve situacije koje mogu da nastupe.
2. ***Nekonzistentan opis modela*** - Ukoliko su u opisu modela za istu situaciju predviđena dva ili više pravila čijom se primenom dobijaju kontradiktorne akcije.
3. ***Nejasan opis modela*** - Ako u jednoj situaciji treba obaviti dve ili više akcija, a pri tome nije definisan njihov redosled.

Formalni opis modela treba da obezbedi veću preciznost i potpunost u opisivanju modela, a ponekad omogućava i da se formalizuje postupak ispitivanja nekompletnosti, nekonzistentnosti i nejasnosti. Ono što je najznačajnije jeste činjenica da uvođenje formalizama u metodologiju modeliranja omogućava da svu svoju pažnju usmerimo na one karakteristike objekata koje su od najvećeg značaja za naše istraživanje, dakle da koristimo apstrakcije.
Formalni model ili apstraktno predstavljanje često se spominje kao ključ uspeha interakcije čovek-realni svet koja se zasniva na „naučno-inžinjerskom“ pristupu. Ovako posmatrani tip interakcije čovek-realni sistem obuhvata **dva različita koraka**. **Prvo**, postoji uvek faza izgradnje modela ili formalizacija, gde se kao rezultat javlja model realnog sistema. **U drugoj fazi**, formalni model se analizira i koristi, a na osnovu dobijenih rezultata donose se odluke koje obezbeđuju efikasnije upravljanje samim realnim sistemom.

Neke opšte preporuke za izgradnju modela:

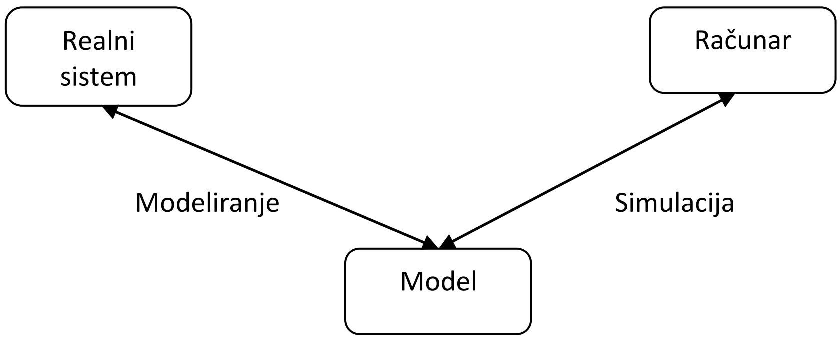
1. Granica sistema sa okolinom mora biti odabrana tako da sistem, odnosno njegov model, obuhvata samo fenomene od interesa. Okolina sistema modelira se tako da se ne opisuju detalji fenomena i uzročna veza među njima, već se daje samo njihov sažeti prikaz.
2. Modeli ne smeju biti suviše složeni niti detaljni, već treba da sadrže samo relevantne elemente sistema – suviše složene i detaljne modele gotovo nije moguće vrednovati ni razumeti.
3. Model ne sme suviše da pojednostavi problem.
4. Model je razumno rastaviti na više dobro definisanih i jednostavnih modula s tačno određenom funkcijom, koje je lakše i izgraditi i proveriti.
5. U razvoju modela preporučuje se korišćenje neke od proverenih metoda za razvoj algoritama i programa. Na primer, *top down* metoda koja se kreće odozgo na dole.
6. Potrebna je provera logičke i kvantitativne ispravnosti modela, i to kako pojedinačnih modula tako i celog modela.

3. Modeliranje i simulacija

Savremeno modeliranje nezamislivo je bez računara. U modeliranju računari se koriste u dve svrhe:
- u razvoju modela i
- u izvođenju proračuna na osnovu stvorenog modela.

Kada reč simulacija koriste računarski stručnjaci, organizatori, menadžeri ili statističari, obično pod simulacijom podrazumevaju proces izgradnje apstraktnih modela za neke sisteme ili podsisteme realnog sveta i obavljanje većeg broja eksperimenata nad njima. Izraz **modeliranje i simulacija** izražava složenu aktivnost koja uključuje tri elementa: **realni sistem**, **model** i **računar**.

Pod **realnim sistemom** podrazumevamo uređen, međuzavisan skup elemenata koji formiraju jedinstvenu celinu i deluju zajednički kako bi ostvarili zadati cilj ili funkciju, bez obzira da li se radi o prirodnom ili veštačkom sistemu, i bez obzira da li taj sistem u posmatranom trenutku postoji ili se njegovo postojanje planira u budućnosti. Realni sistem je izvor podataka o ponašanju, a ovi se podaci javljaju u obliku zavisnosti X (t) , gde je X bilo koja promenljiva koja interesuje istraživača, a *t* je vreme mereno u odgovarajućim jedinicama. Drugim rečima, realni sistem se može posmatrati kao izvor podataka za specifikaciju modela.



Model, kao i svaki realni sistem, ima svoje objekte koji se opisuju atributima ili promenljivim. On je apstraktni prikaz sistema i daje njegovu strukturu, njegove komponente i njihovo uzajamno delovanje. Pošto razmatramo samo slučaj računarske simulacije, to se pod modelom može podrazumevati skup instrukcija (program) koji služi da se generiše ponašanje simuliranog sistema. Ponašanje modela ne mora da bude u potpunosti jednako ponašanju simuliranog sistema, već samo u onom domenu koji je od interesa.

Računar predstavlja uređaj sposoban za izvršenje instrukcija modela, koje na bazi ulaznih podataka generišu razvoj modela u vremenu i uz različite metode i programske alate, omogućavaju pogodan ambijent za stvaranje složenih modela i efikasan rad nad njima.

Modeliranje je proces kojim se uspostavlja veza između realnog sistema i modela, dok je **simulacija** proces koji uspostavlja relaciju između modela i računara.

Relacija modeliranja odnosi se na validnost modela. Validnost ili valjanost modela opisuje koliko verno jedan model predstavlja simulirani sistem. **Validacija modela** je proces utvrđivanja stepena slaganja podataka o realnom sistemu sa podacima. Proces validacije je veoma značajan, jer se na osnovu njega donose odluke o upotrebljivosti rezultata simulacije, izmeni modela, izmeni podataka (ulaznih promenljivih, parametara), daljem nastavku simulacije, ponavljanju simulacije, itd.

Relacija simulacije odnosi se na proveru da li simulacioni program verno prenosi model na računar kao i na tačnost kojom računar izvršava instrukcije modela. Pre poređenja stvarnih podataka sa podacima koje generiše računar (simulator), mora se utvrditi tačnost, odnosno korektnost simulatora. **Verifikacija** je proces procene korektnosti simulatora.

Aktivnosti procesa modeliranja i simulacije sa bazom modela kao centralnim objektom

Procesom modeliranja se upravlja na osnovu ciljeva koji se generišu van granica sistema. Svaki novi cilj inicira aktivnost sinteze modela. Pri sintezi modela se koristi raspoloživo znanje iz baze modela i baze podataka. Ove baze čuvaju i organizuju prikupljene podatke o realnom sistemu. Faze simulacije (eksperimentisanje sa modelom) i validacije slede fazu izgradnje modela.
Validacija vodi novom eksperimentisanju nad realnim sistemom i može da zahteva dodatne modifikacije ili čak odbacivanje i reinicijalizaciju prvobitnog modela, a takođe može doći i do izmene ciljeva ili postavljanja novih. Na kraju se dobijaju modeli koji se koriste za dostizanje ciljeva. Te modele koriste donosioci odluka, a mogu se memorisati u bazi modela i koristiti kasnije.

4. Karakteristike simulacionog modeliranja

Računarska simulacija je proces rešavanja problema koji se tiče predviđanja i određivanja budućih stanja realnog sistema na osnovu proučavanja računarskog modela tog sistema.
Simulacioni eksperimenti najčešće se izvode sa ciljem da se prikupe određene informacije, čije bi dobijanje putem eksperimenta nad samim realnim sistemom bilo nepraktično ili suviše skupo. Te informacije se kasnije koriste u procesu odlučivanja. **Cilj simulacije** jeste da proučimo ponašanje sistema koji simuliramo, ali i da ustanovimo kako bi se isti sistem ponašao kada bi na njega delovao neki drugi skup promenljivih okolnosti.
Simulacioni modeli prikupljaju podatke o promenama stanja sistema i izlaza, fokusirajući se na ponašanje individualnih komponenti sistema. Značaj simulacionih modela proističe iz činjenice da se samo mali broj kompleksnih realnih sistema može adekvatno opisati preko analitičkih jednačina.
Primenom simulacionog modeliranja ne može se dobiti rešenje u analitičkom obliku, u kojem su zavisne promenljive funkcije nezavisnih promenljivih, već se rešenje problema dobija eksperimentisanjem nad modelom. Simulacioni eksperimenti daju kao rezultat skup tačaka, tj. vrednosti zavisnih promenljivih za pojedine vrednosti nezavisnih promenljivih. Zbog slučajnog karaktera promenljivih modela, dobija se čak i više različitih vrednosti zavisnih promenljivih za istu vrednost nezavisnih promenljivih. Pri tome, planiranje i analiza simulacionih eksperimenata zahtevaju statistički pristup.
Simulacioni modeli najčešće su modeli dinamičkih sistema, tj. sistema koji se menjaju u vremenu, s obzirom na to da su istraživači, u većini slučajeva, zainteresovani za simulaciju modela dinamičkih sistema. Ovi su modeli uglavnom dati u obliku konceptualnih (strukturnih) i računarskih modela.
Proces simulacije se oslanja i na metode operacionih istraživanja i numeričke analize. Zbog postojanja slučajnih promenljivih u simulacionim modelima, često se koriste i pristupi teorije verovatnoće i statistike.

5. Potreba za simulacijom

Može se postaviti pitanje zbog čega se uopšte jedan sistem (simulirani sistem) zamenjuje modelom, a zatim vrši simulacija.

Postoji više razloga:

- ✓ Eksperiment nad realnim sistemom može da bude skup ili čak nemoguć.
- ✓ Analitički model nema analitičko rešenje.
- ✓ Sistem može da bude suviše složen da bi se opisao analitički.

Pored ovih, navešćemo još nekoliko značajnih razloga:

• Eksperimentisanje sa realnim sistemom, čak i ako se zanemare drugi aspekti, uglavnom je neisplativo ili suviše složeno. Modeliranje, s druge strane, može da ukaže na to da li je dalje ulaganje u eksperiment ekonomski opravdano ili ne.

• Izgradnja modela i simulacija ponekad imaju za cilj da se shvati funkcionisanje postojećeg sistema čija je struktura nepoznata i ne može joj se prići.

• Prilikom pronalaženja optimalnog funkcionisanja nekog sistema, uobičajeno je da se menjaju razni parametri, a takav realan sistem ne postoji.

• Ponekad treba simulirati uslove pod kojima nastupa razaranje sistema, a razaranje realnog sistema najčešće nije dopustivo.

• Vreme može da bude vrlo jak razlog da se pribegne simulaciji. Pri simulaciji, vreme se može sažeti. To je značajno kod simulacije dugotrajnih procesa. U drugim slučajevima, vreme se može znatno produžiti.

• Kada se vrši realni eksperiment, uvek postoji izvesna greška pri merenju usled nesavršenosti mernih uređaja. Pri simulaciji ove greške nema. Postoji samo greška "zaokruživanja" usled konačne dužine reči u računaru, ali se ona može učiniti zanemarljivom.

6. Mogućnosti primene simulacije, prednosti i nedostaci simulacije

Nekoliko situacija kada se simulacija može uspešno primeniti:

• Simulacija omogućava proučavanje i eksperimentisanje koje uzima u obzir sveukupne interakcije složenog sistema ili podsistema unutar složenog sistema;
• Informacione i organizacione promene ili promene u okruženju mogu se simulirati, a ujedno se mogu posmatrati efekti tih promena na ponašanje modela;
• Znanje stečeno u procesu izgradnje modela i simulacije može biti od velikog značaja kod poboljšanja sistema koji se ispituje;
• Menjanjem simulacionih ulaza i posmatranjem rezultujućih izlaza, dolazimo do važnog saznanja o tome koje su promenljive sistema najvažnije i kako promenljive utiču jedna na drugu;
• Simulacija se može koristiti za eksperimentisanje sa novim koncepcijama ili politikama pre nego što se izvrši njihova implementacija;
• Simulacija se može koristiti i kao pedagoško sredstvo sa ciljem da poboljšava metodologije analitičkih rešenja;
• Simulacija se može koristiti za verifikaciju analitičkih rešenja.

Prednosti i nedostaci simulacije

Osnovne prednosti korišćenja simulacije :

1. Jednom izgrađeni model može se višestruko koristiti za analizu predloženih planova ili politika.
2. Simulacione metode mogu se koristiti kao pomoć kod analize, čak iako su ulazni podaci na neki način nepotpuni.
3. Čest je slučaj da se simulacioni podaci mogu mnogo jeftinije dobiti od sličnih podataka iz realnog sistema.
4. Simulacione metode lakše je primeniti nego analitičke metode.
5. Analitički modeli uglavnom zahtevaju više pojednostavljujućih pretpostavki koje ih čine matematički prilagodljivim. Simulacioni modeli takva ograničenja nemaju. Sa analitičkim modelima, najčešće se može izračunati jedino ograničeni broj merljivih karakteristika sistema, dok kod simulacionih modela generisani podaci mogu da se koriste za procenu bilo koje shvatljive i merljive karakteristike.
6. U nekim slučajevima, simulacija je jedino sredstvo za rešavanje odgovarajućeg problema.
7. Moguće je opisati i rešavati složene dinamičke probleme sa slučajnim promenljivim koji su nedostupni matematičkom modeliranju

Osnovni nedostaci korišćenja simulacije :

1. Simulacioni modeli za digitalne računare mogu biti skupi i mogu zahtevati značajno vreme za izgradnju i validaciju.
2. Zbog statističkog karaktera simulacije potrebno je izvođenje većeg broja simulacionih eksperimenata kako bi se dobio odgovarajući uzorak rezultata simulacije, a već i pojedinačno izvođenje eksperimenta može zahtevati dosta vremena i memorije računara.
3. Ne dobijaju se zavisnosti izlaznih promenljivih od ulaznih promenljivih modela niti optimalna rešenja.

4. Za ispravno korišćenje simulacionog modeliranja potrebno je poznavanje više različitih metoda i alata.
5. Vrednovanje modela je dosta složeno i zahteva dodatne eksperimente.

7. Simulacioni proces

Simulacioni proces je struktura rešavanja stvarnih problema pomoću simulacionog modeliranja. Struktura simulacionog procesa nije strogo sekvencijalna, već je moguć i povratak na prethodne korake procesa, zavisno od rezultata dobijenih u pojedinim fazama procesa.

Osnovni koraci simulacionog procesa:

1. Definicija cilja simulacione studije
Definicija željenog cilja i svrhe studije: problem koji treba rešiti (oblikovanje sistema, analiza sistema i sl.), granice sistem/okolina, nivo detaljnosti.

2. Identifikacija sistema
Opis komponenti sistema, interakcija komponenti, način rada, veze s okolinom, formalni prikaz sistema.

3. Prikupljanje podataka o sistemu i njihova analiza
Prikupljanje i merenje relevantnih podataka o sistemu, analiza tih podataka (izbor raspodela nezavisnih slučajnih promenljivih, ocena vrednosti parametara raspodela).

4. Izgradnja simulacionog modela
Stvaranje konceptualnog modela koji adekvatno opisuje sistem i omogućava rešavanje zadatog problema.

5. Izgradnja simulacionog programa
Izbor programskog jezika ili paketa i stvaranje simulacionog programa bilo pisanjem programa, bilo automatskim generisanjem programa na osnovu konceptualnog modela.

6. Verifikacija simulacionog programa
Testiranje simulacionog programa prema postavkama simulacionog modela (pojedinačne procedure, generisanje slučajnih promenljivih, povezivanje procedura). Ukoliko verifikacija programa nije dala zadovoljavajuće rezultate, potreban je povratak na korak 5.

7. Vrednovanje (validacija) simulacionog modela
Ispitivanje da li simulacioni model adekvatno predstavlja stvarni sistem (ispitivanjem podudarnosti izlaza modela i realnog sistema, analizom rezultata od strane eksperata, analizom osetljivosti). Ukoliko vrednovanje modela nije uspešno, potrebno je vratiti se na tačku 4 (izmene u simulacionom modelu).

8. Planiranje simulacionih eksperimenata i njihovo izvođenje
Planiranje simulacionih eksperimenata koji omogućuju ispunjenje cilja studije. Izvođenje simulacionih eksperimenata prema usvojenom planu.

9. Analiza rezultata eksperimenata
Statistička analiza rezultata simulacionih eksperimenata. Tokom analize rezultata može se ukazati potreba za dopunom koraka 8 (izvođenje dodatnih eksperimenata).

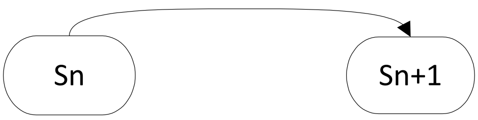
10. Zaključci i preporuke
Prezentacija relevantnih rezultata na osnovu kojih se mogu doneti odgovarajuće odluke (izbor konfiguracije sistema, izmene u sistemu, idr.).

8. Podele i vrste simulacionih modela

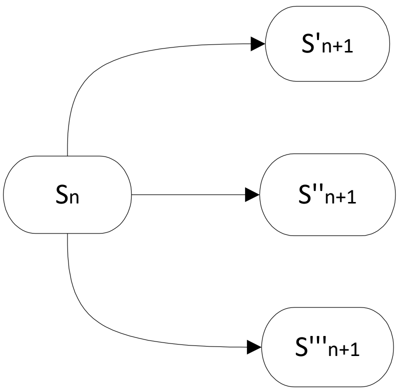
Razlikujemo **dva osnovna načina podele simulacionih modela**:
1. prema vrsti promenljivih u modelu i
2. prema načinu na koji se stanje u modelu menja u vremenu.

Deterministički i stohastički modeli

Deterministički modeli su oni čije se ponašanje može predvideti, odnosno u kojima je novo stanje sistema u potpunosti određeno prethodnim stanjem.
Stohastički modeli su oni čije se ponašanje ne može unapred predvideti, ali se mogu odrediti verovatnoće promena stanja sistema. Za stohastičke modele je karakteristično slučajno ponašanje, odnosno postojanje slučajnih promenljivih u sistemu.



Deterministički model u kojem se stanje sistema S_n menja pod uticajem aktivnosti A u stanje sistema S_{n+1}.



Stohastički model u kome se stanje sistema S_n može promeniti u jedno od stanja S'_{n+1}, S''_{n+1}, S'''_{n+1} pod uticajem aktivnosti A.

Diskretni i kontinualni modeli

U **diskretnim modelima** stanje sistema se menja samo u pojedinim tačkama u vremenu (nema kontinualne promene stanja). Takve promene se nazivaju događaji. *(npr. u modelu samoposluge mogu postojati promenljive koje opisuju broj kupaca u redovima pred kasama, a taj se broj može menjati samo u trenutku dolaska kupaca u red i u trenutku početka opsluživanja na kasi)*
U **kontinualnim modelima** promenljive stanja se menjaju kontinualno u vremenu. Treba imati u vidu da se na digitalnim računarima ne mogu izvoditi kontinualne promene veličina, već se one moraju aproksimirati skupom diskretnih vrednosti. *(npr. let aviona čiji se položaj i brzina menjaju kontinualno u vremenu)*
Mogući su i **mešoviti, kontinualno-diskretni modeli** koji sadrže i kontinualne i diskretne promenljive.

Vrste simulacionih modela

Četiri osnovne vrste simulacionih modela su:

- Monte Karlo (Monte Carlo) simulacija
- Kontinualna simulacija
- Simulacija diskretnih događaja
- Mešovita, kontinualno-diskretna simulacija.

Osim Monte Karlo simulacije, koja je statička, sve ostale nabrojane vrste su dinamičke.

- **Monte Karlo simulacija**

Monte Karlo simulacija (statistička simulacija) povezana je sa slučajnim fenomenima. Neki autori Monte Karlo simulacijama nazivaju bilo koju vrstu programa koji se koriste slučajnim brojevima. S druge strane, u većini literature ovaj termin se upotrebljava samo za statičke tipove simulacije kod kojih se u rešavanju problema koristi stvaranje uzoraka iz raspodela slučajnih promenljivih.
<u>Razlikujemo sledeće tipove primene Monte Karlo simulacije:</u>

<u>*Deterministički problemi koje je teško ili skupo rešavati*</u>
Tipičan primer ovog tipa je računanje vrednosti određenih integrala koji se ne mogu rešiti analitički, tj. čija je podintegralna funkcija takva da se ne može naći rešenje u obliku analitičkog izraza. Monte Karlo simulacija pristupa proračunu integrala tako što se generiše niz slučajnih tačaka (x_j,y_j) sa jednakim verovatnoćama unutar određenog pravougaonika i potom ispituje koliko je generisanih tačaka unutar površine koja odgovara integralu. Ovakav pristup se zasniva na generisanju slučajnih brojeva i analogan je onom koji se koristi kod simulacije sistema sa diskretnim događajima.

<u>*Složeni fenomeni koji nisu dovoljno poznati*</u>
Za njih je karakteristično da nije poznat način uzajamnog delovanja između elemenata već su poznate samo verovatnoće njegovog ishoda, koje se koriste za izvođenje niza eksperimenata koji daju uzorke mogućih stanja zavisnih promenljivih. Statističkom analizom takvih uzoraka dobija se raspodela verovatnoća zavisnih promenljivih koje su od interesa. Najčešće primenjuje kod analiziranja društvenih ili ekonomskih fenomena.

<u>*Statistički problemi koji nemaju analitička rešenja*</u>
Npr. procene kritičnih vrednosti ili testiranje novih hipoteza. Prilikom rešavanja takvih problema takođe se koristi generisanje slučajnih brojeva i promenljivih.

- **Kontinualna simulacija**

Kontinualna simulacija se koristi za dinamičke probleme kod kojih se promenljive stanja menjaju kontinualno u vremenu. Postoje dve osnovne klase problema koji se rešavaju ovom metodom.

U prvoj klasi su relativno jednostavni problemi koji su opisani detaljno i kod kojih su promene „glatke“ i prirodno se opisuju diferencijalnim jednačinama. To su problemi iz fizike, biologije iinženjerstva.
U drugoj klasi su problemi koji nastaju opisom veoma složenih sistema u agregiranom obliku, u kojem se niz elemenata sistema redukuje na manji broj komponenti i u kojima se promene usistemu aproksimiraju konstantnim brzinama promene. To su najčešće problemi iz područja ekonomije i društvenih nauka.

Razlikujemo tri osnovna tipa kontinualnih simulacionih modela:

1.Modeli koji se opisuju običnim diferencijalnim jednačinama (sistemi običnih diferencijalnih jednačina)
Problemi u vezi sa raznim procesima gde se radi o jednoj nepoznatoj funkciji (*y=y(t)*) jedne nezavisno promenljive (*t*), izražavaju se matematički jednačinama u kojima se, pored nezavisno promenljive i nepoznate funkcije, javljaju (obavezno) i izvodi te funkcije (*dy /dt*). Tim jednačinama se u svakom konkretnom slučaju uspostavlja veza između nepoznate funkcije (y), njene realne promenljive (t) i njenih izvoda (y') i takve se jednačine nazivaju obične diferencijalne jednačine. Modeli koji se opisuju običnim diferencijalnim jednačinama ili njihovim sistemima mogu se u određenim slučajevima rešiti analitički, ali je to dosta retko kod stvarnih problema. Zbog toga se za takve jednačine moraju koristiti numeričke metode. Numeričke metode koje služe za rešavanje diferencijalnih jednačina kod kojih se kao nezavisna promenljiva javlja vreme, uobičajeno se nazivaju metode kontinualne simulacije.

2.Modeli koji se opisuju sistemima parcijalnih diferencijalnih jednačina
Parcijalne diferencijalne jednačine sadrže više od jedne nezavisne promenljive (x_j) po kojima se traže izvodi zavisne promenljive. Tipični primeri problema koji se mogu opisati parcijalnim diferencijalnim jednačinama su problemi aerodinamike, hidrodinamike i meteorologije.

3.Modeli dinamike sistema (System Dinamics)
Dinamika sistema je metodologija istraživanja, modeliranja i simulacije složenih dinamičkih sistema. Sistemi sa povratnom vezom su osnovni tip sistema koji se modeliraju dinamikom sistema. Povratna veza može biti pozitivna ili negativna. Modeli sa povratnom vezom koriste se najčešće za modeliranje inženjerskih, bioloških, društvenih i ekonomskih sistema.
Dinamika sistema prikazuje sisteme kao povezane upravljačke petlje. Novo stanje sistema u narednom trenutku vremena računa se na osnovu stanja u prethodnom trenutku vremena i razlike ulaznih i izlaznih tokova za to stanje u prethodnom intervalu vremena. Da bi proračun mogao početi, neophodno je da budu zadate početne vrednosti veličina kao i vrednosti svih konstanti i parametara koje model sadrži.

- **Simulacija diskretnih događaja**

Simulacija diskretnih događaja je specifična metodologija simulacije koja se bavi modeliranjem sistema koji se mogu predstaviti skupom događaja. Pod **događajem** ovde se podrazumeva diskretna promena stanja entiteta sistema. Događaj nastupa u određenom trenutku vremena, odnosno promene stanja entiteta se dešavaju diskontinualno u vremenu, tj. samo u nekim vremenskim trenucima (kada nastupi dogadjaj). Simulacija opisuje svaki diskretni događaj, krećući se od jednog događaja do drugog pri čemu nastaje pomak (prirast) vremena simulacije. Između dva uzastopna događaja, stanje sistema se ne menja.

- **Mešovita simulacija**

Kod pojedinih vrsta sistema, kontinualna simulacija kao i simulacija diskretnih događaja, ne mogu u potpunosti da opišu način rada sistema. Da bi se takvi sistemi modelirali i simulirali, razvijena je mešovita simulacija koja omogućava integrisanje kontinualnih i diskretnih elemenata sistema.
Veza između diskretnog i kontinualnog pristupa postiže se uvođenjem **dva tipa događaja**. **Vremenski događaji** su događaji koje generiše *mehanizam upravljanja događajima*, kakav postoji u simulaciji diskretnih događaja. Oni mogu da izazovu trenutnu promenu stanja kontinualne promenljive. S druge strane, **događaji stanja** su oni događaji koje aktivira *mehanizam pomaka vremena sa konstantnim prirastom*, čiji je vremenski interval mali, a koji je karakterističan za kontinualnu

simulaciju. Ovi događaji mogu da aktiviraju događaje diskretnog dela modela. Poznatiji simulacioni jezici za mešovitu simulaciju su GASP i SLAM.

9. Klasifikacija modela

Modeli mogu da se klasifikuju prema raznim klasifikacionim kriterijumima. Neke od najvažnijih klasifikacija su izvršene u odnosu na:

1. promenljive
2. prirodu opsega vrednosti promenljivih modela
3. prirodu opsega vrednosti promenljive "vreme"
4. vremensku zavisnost modela
5. determinizam
6. predviđanje budućnosti
7. linearnost
8. vrstu računara
9. formalni opis modela

1. Klasifikacija u odnosu na promenljive

Kod svakog modela moguće je identifikovati opisne promenljive značajne za njegovo razumevanje, opis i upravljanje. Skup opisnih promenljivih se može podeliti na podskup onih koje je moguće i podskup onih koje nije moguće posmatrati. Sve opisne promenljive jednog modela mogu se podeliti na **ulazne**, **izlazne** i **promenljive stanja**. Svaka promenljiva ima svoj *opseg ili domen* - skup vrednosti koje joj se mogu dodeliti, kao i jednu funkciju kojom se opisuju promene tih vrednosti u vremenu.
Modeli koji nemaju ni jednu promenljivu stanja nazivaju se modeli *bez memorije ili trenutne funkcije*. Njihove izlazne promenljive zavise od vrednosti ulaznih promenljivih u trenutku posmatranja.
Zavisno od ulaznih promenljivih, modeli mogu biti autonomni (bez ulaznih promenljivih) i neautonomni (sa ulaznim promenljivama).
Autonomni model koji ne sadrži izlaznu promenljivu naziva se zatvoreni. S druge strane, autonomni modeli koji imaju izlazne promenjive i svi neautonomni modeli nazivaju se otvoreni modeli.

2. Klasifikacija u odnosu na prirodu opsega vrednosti promenljivih modela

Opseg promenljivih podrazumeva skup svih vrednosti koje promenljiva može da uzme. Opisne promenljive mogu uzimati vrednosti iz prebrojivog (diskretnog) ili neprebrojivog (kontinualnog) skupa. U skladu sa tim razlikujemo **tri klase modela**:

1. Modeli sa diskretnim stanjima

Kod ovih modela sve tri vrste opisnih promenljivih (ulazne, izlazne i promenljive stanja) uzimaju vrednosti iz skupova čiji su elementi diskretne vrednosti.

2. Modeli sa kontinualnim stanjima

Kod ovih modela sve tri vrste opisnih promenljivih uzimaju vrednosti iz podskupova realnih brojeva.

3. Modeli sa mešovitim stanjima

Pojedine promenljive uzimaju vrednosti iz raznih podskupova čiji su elementi diskretne vrednosti, a ostale promenljive iz različitih podskupova realnih brojeva.

3. Klasifikacija u odnosu na prirodu opsega vrednosti promenljive "vreme"

Skup vrednosti koje se dodeljuju promenljivoj "vreme" može biti prebrojiv ili neprebrojiv. Stoga razlikujemo modele sa kontinualnim vremenom i modele sa diskretnim vremenom.
Razlikujemo dve podklase ovih modela:

- modele sa kontinualnim vremenom i kontinualnim promenama stanja
- modele sa kontinualnim vremenom i diskretnim promenama stanja (iako vreme teče kontinualno, promene stanja se mogu dešavati samo u diskretnim skokovima)

U diskretnim vremenskim modelima, vreme se povećava u inkrementima koji ne moraju biti ekvidistantni.
Razlikujemo dve podklase ovih modela:

- modele sa diskretnim vremenom i kontinualnim promenama stanja
- modele sa diskretnim vremenom i diskretnim promenama stanja

Promenljive simulacionog modela se mogu menjati na bilo koji od četiri navedena načina:
kontinualno u kontinualnom vremenu, diskretno u kontinualnom vremenu, diskretno u diskretnom vremenu i kontinualno u diskretnom vremenu. Kod složenih modela, u većoj meri se koriste promenljive koje uzimaju vrednosti iz diskretnog skupa i u diskretnoj vremenskoj osnovi. Za modele kod kojih se promene vrednosti promenljivih stanja dešavaju u diskretnim vremenskim trenucima, trajanje simulacije zavisi od izbranog vremenskog koraka, tj. od izabrane vremenske jedinice. Vremenski korak može biti konstantan ili promenljiv.

4. Klasifikacija u odnosu na vremensku zavisnost modela

Ova klasifikacija tretira pitanje eksplicitne zavisnosti pravila interakcije modela od vremena. Ukoliko je struktura modela zavisna od vremena, tada se radi o *vremenski promenljivom* – varijantnom modelu. U suprotnom, kada je struktura modela nezavisna od vremena, model se naziva *vremenski nepromenljiv* - invarijantan. Takav model se još naziva i stacionarni model.

5. Klasifikacija u odnosu na determinizam

Ova klasifikacija razmatra uključivanje slučajnih promenljivih u opis modela. U determinističkim modelima, vrednosti promenljivih stanja i ulaznih promenljivih u jednom trenutku jednoznačno određuju vrednosti promenljivih stanja u sledećem trenutku. Takvi modeli ne sadrže slučajne promenljive. U nedeterminističkim (stohastičkim) modelima postoji bar jedna slučajna promenljiva.

6. Klasifikacija u odnosu na predviđanje budućnosti

Postoje modeli koji za izračunavanje vrednosti promenljivih stanja uzimaju u obzir i buduće vrednosti ulaznih promenljivih – **anticipatorski** modeli. Ukoliko to nije slučaj, model je **neanticipatorski**. Kod modela koji služe za planiranje budućnosti (proizvodnja, investicije, stanovništvo) neophodno je uzeti u obzir buduće vrednosti ulaznih promenljivih.
Kod anticipatorskih modela postoji poseban modul pomoću koga se generišu buduće vrednosti ulaznih promenljivih, a zatim se simulacija ponavlja uz korišćenje prethodno generisanih vrednosti.

7. Klasifikacija u odnosu na linearnost

Linearni modeli menjaju stanja poštujući zakonitosti linearnih transformacija. Jedna linearna transformacija $L: U \rightarrow Y$ zadovoljava princip superpozicije, ako za i važi (38.str)

8. Klasifikacija prema vrsti računara

Tri vrste računara se mogu koristiti za simulaciju:

- analogni
- digitalni
- hibridni

Skoro svi modeli se mogu simulirati na digitalnim i hibridnim računarima. Na analognim računarima se mogu simulirati samo kontinualni modeli sa kontinualnim vremenom.

9. Klasifikacija u odnosu na formalni opis modela

Pod formalnim opisom modela se podrazumeva precizan, matematički opis modela. Modeli sa kontinualnim promenama vremena se opisuju diferencijalnim jednačinama i spadaju u kontinualne vremenske modele. Prema tome, diskontinualni modeli su oni modeli u kojima su najmanje jedna promenljiva stanja i/ili njen izvod diskontinualni. Diskretni modeli menjaju svoje stanje u diskretnim vremenskim trenucima. Između dva uzastopna vremenska trenutka, stanje modela ostaje nepromenjeno. Diskretni modeli mogu biti diskretni vremenski i kontinualni vremenski modeli. (slika 2.2 40.str)

10. Formalna specifikacija modela

Svaka klasa objekata, može se predstaviti odgovarajućim formalizmom koji definiše njene parametre i ograničenja. Da bi se definisao poseban objekat neke klase u okviru nekog formalizma, parametrima formalizma dodeljuju se vrednosti koje zadovoljavaju ograničenja.
Struktura formalizma odnosi se kako na parametre, tako i na ograničenja.
Klase objekata su najčešće povezane tako da se te veze mogu formalizovati kao preslikavanja iz jedne klase u drugu. Posebno su interesantne **tri vrste takvih preslikavanja**: apstrakcija, asocijacija i specifikacija.

Apstrakcija
Apstrakcija je preslikavanje koje podrazumeva kontrolisano uključivanje detalja prilikom opisivanja modela. To je proces kojim se ostavljaju po strani pojedini detalji objekata, odnosno vrši se razdvajanje bitnih osobina od nebitnih, kako bi se ukazalo na suštinu nekog objekta. Izvornu klasu nazivamo „konkretnom“, a onu drugu, ciljnu, „apstraktnom“. Značaj apstrakcije je u tome što je moguće veći broj izvornih objekata predstaviti istim odredišnim apstrakcijama.

Asocijacija
Asocijacija je vrsta preslikavanja od višeg ka nižem nivou u hijerarhiji specifikacije sistema. Inverzno preslikavanje nazivamo realizacijom ili implementacijom.

Specifikacija
Da bi uspostavili relacije između objekata različitih srodnih podklasa, potrebno je definisati preslikavanje koje prevodi jedan poseban formalizam u drugi, generalniji. Ukoliko postoji takvo preslikavanje, svi koncepti i akcije koji se mogu primeniti nad objektima definisanim u generalnom formalizmu, primenljivi su i nad objektima definisanim u posebnom formalizmu.

11. Ocene parametara determinističkog i stohastičkog modela

Ocena parametara modela predstavlja postupak eksperimentapnog određivanja vrednosti parametara koji se pojavljuju u matematičkom opisu modela.

Za **deterministički model** sa poznatom strukturom mogu se uvesti sledeće definicije:

1. *Za skalarni parametar p_j kaže se da je identifikabilan na intervalu [t_0,T], ako postoji konačni broj rešenja za p_j koja proiziliaze iz relacija datog modela*. Skalarni parametar je neidentifikabilan ako postoji beskonačan broj rešenja za p_j koja prozilaze iz relacija modela.

2. *Opis modela je sistemski identifikabilan, ako su svi parametri identifikabilni.* U suprotnom, ako je jedan parametar neidentifikabilan, ops modela je sistemski neidentifikabilan.

3. Skalarni parametar p_j je jedinstveno identifikabilan na intervalu *[t_0,T],* ako postoji jedinstveno rešenje za p_j koje proizilazi iz relacija modela sa datim ulazom.

4. Specifikacija modela je parametarski identifikabilna na intervalu *[t_0,T]* ako su svi parametri jedinstveno identifikabilni.

Bitno je da **identifikabilnost sistema i parametara zavisi od dva faktora:**

- *Specifičnosti primenjenog ulaza i početnih uslova* – Početni uslovi su bitni jer identifikabilnost sistema i parametara može zavisiti od njihovih vrednosti. Može se dogoditi da se za dati sistem ulazni segment može podeliti u više posebnih kategorija.
- *Strukture jednačina i ograničenja* – Neke strukture se mogu identifikovati ako se primene odgovarajući ulazi, dok su druge neidentifikabilne, bez obzira na to kakav se ulaz primeni.

Kod stohastičkih sistema, identifikabilnost zavisi od sledeća tri faktora:

1. Strukture – koja parametre stavlja u relaciju jedne sa drugima kao i sa ulazima i izlazima. Kao i u determinističkom slučaju, neidentifikabilnost može biti uzrokovana specifičnim strukturalnim odnosima komponenti sistema.
2. Ulaza – od kojih neki mogu biti suviše "prosti" da bi se postigla identifikabilnost.
3. Procedure procene – koje treba da budu konzistentne u smislu da konvergiraju ka stvarnim vrednostima parametara.

12. Statistički pristup proceni parametara modela

Statističke ocene parametara statičkih modela se mogu realizovati u prostoj sekvencijalnoj formi ili rekurzivnoj formi.

Za identifikaciju parametara sistema za koje su podaci o njihovom ponašanju prikupljeni i memorisani na nekom medijumu, najjednostavnije je primeniti sekvencijalne statističke metode.

Međutim, za prikupljanje podataka u realnom vremenu, kada je broj podataka takav da se ne može memorisati, treba koristiti odgovarajuće statističke formule u rekurzivnoj formi. Za računanje u rekurzivnoj formi je najčešće potrebno samo nekoliko promenljivih koje se ažuriraju u toku eksperimenta. Na taj način se štedi u memorijskim resursima. Prednost rekurzivnog pristupa se ogleda i u tome što su prilikom svakog ažuriranja statističkih pokazatelja poznate njihove tekuće vrednosti.

Ocena srednje vrednosti slučajne promenljive

Sekvencijalni metod

Uz pretpostavku da su svi rezultati merenja na realnom sistemu dostupni u vreme izračunavanja, procedura za procenu srednje vrednosti može se definisati na sledeći način:

$$\bar{y}_N = \frac{1}{N}\sum_{i=1}^{N} y_i$$

Sva merenja se sabiraju i rezultat se deli sa brojem uzoraka.

Rekurzivni metod

Za rekurzivno izračunavanje je karakteristično sledeće:

• baza podataka (rezultati merenja) se proširuje tokom računanja,

• međurezulati za određeini broj uzoraka su takođe dostupni i oni teže sekvencijalnom rešenju kako izračunavanje odmiče.

Računanje se odvija po sledećoj proceduri:

$$\hat{a}_0 = 0$$
$$\hat{a}_k = \hat{a}_{k-1} - \frac{1}{k}(\hat{a}_{k-1} - y_k)$$

gde je:

y_k– tekući uzorak $(k = 1,2, \dots N)$

13. Ocena nepoznatog parametra po metodi najmanjih kvadrata

Veza između parametra modela i rezultata eksperimenta data je relacijom:

$$y_i = x_i a + e_i, \quad i = 1, 2, \dots, N$$

Pretpostavke su:
x_i – poznate vrednosti
e_i – greške merenja

Potrebno je proceniti nepoznati parametar a koristeći dostupne rezultate merenja.

Sekvencijalno rešenje

Cilj je da se minimizira odstupanje između podataka koje generiše realni sistem i podataka koji se dobijaju na osnovu modela, odnosno da se minimizira kvadrat greške:

$$J = \sum_{i=1}^{N} [y_i - x_i a]^2 = \sum_{i=1}^{N} e_i^2$$

Ova metoda je poznata kao metoda najmanjih kvadrata (MNK). Osnovna prednost ove metode leži u njenoj jednostavnosti.

Za kriterijumsku funkciju koja se minimizira uzima se kvadratna funkcija greške jer se njen minimum lako nalazi izjednačavanjem parcijalnih izvoda po nepoznatim parametrima sa nulom.

U gornjem slučaju imamo jedan nepoznati parametar, odnosno

$$\frac{\partial J}{\partial a} = 0 \text{ , tj.}$$

$$2 \sum_{i=1}^{N} [y_i - x_i a] * (-x_i) = 0$$

odakle sledi:

$$\left[\sum_{i=1}^{N} x_i^2 \right] * a = \sum_{i=1}^{N} x_i \, y_i$$

iz čega se direktno može odrediti nepoznati parametar a.

Rekurzivno rešenje

Uvedimo sledeće oznake:

$$\hat{a}_k = p_k b_k$$

gde je:

$$p_k = \left[\sum_{i=1}^{k} x_i^2\right]^{-1}$$

$$b_k = \sum_{i=1}^{k} x_i y_i$$

U rekurzivnoj formi ove formule možemo napisati:

$$p_k = p_{k-1} - p_k p_{k-1} x_k^2$$

$$b_k = b_{k-1} + x_k y_k$$

Ako definišemo k_k u obliku:

$$k_k = p_{k-1} x_k \left[1 + p_k p_{k-1} x_k^2\right]^{-1}$$

tada se ocena parametra *a* može izraziti kao:

$$\hat{a}_k = \hat{a}_{k-1} - k_k (\hat{a}_{k-1} - y_k)$$

Opisani izrazi čine rekurzivni algoritam metode najmanjih kvadrata za procenu jednog nepoznatog parametra.

14. Procena k nepoznatih parametara modela

Posmatrajmo slučaj gde je veza između k parametara modela i rezultata nekog eksperimenta data sledećom relacijom:

$$y = a_1 x_1 + a_2 x_2 + \ldots + a_k x_k \ ,$$

gde su parametri a_k nepoznati.
Pretpostavimo da smo *n* puta obavili eksperiment na sistemu, čiji se rezultati mogu predstaviti tabelom:

Redni broj eksperimenta	x_1	x_2	……..	x_k	y
1	x_{11}	x_{12}		x_{1k}	y_1
2	x_{21}	x_{21}		x_2k	y_2
. .	. .	. .	. .	. .	. .
j	x_{j1}	x_{j2}		x_{jk}	y_j
. .	. .	. .	. .	. .	. .
n	x_{n1}	x_{n2}	………	x_{nk}	y_n

Opšti oblik rezultata za n eksperimenata

Tabela prikazuje da, ako su u j-om eksperimentu realizovana merenja x_{j1}, x_{j2}, ..., x_{jk} , tada se kao rezultat dobija y_j. S obzirom da u većini slučajeva nije moguće izvršiti tačna merenja, a s druge strane, često usvojeni model nije adekvatan, eksperimentalni rezultati neće zadovoljavati gornju relaciju, već će postojati određena odstupanja (greške) koja ćemo obeležiti sa e_j.
Na osnovu izloženog, za *n* eksperimenata i *k* promenljivih dobija se sistem jednačina oblika:

$$\begin{gathered}
\boldsymbol{y_1} - (\boldsymbol{a_1x_{11}} + \boldsymbol{a_2x_{12}} + \ldots + \boldsymbol{a_kx_{1k}}) = \boldsymbol{e_1} \\
\boldsymbol{y_2} - (\boldsymbol{a_1x_{21}} + \boldsymbol{a_2x_{22}} + \ldots + \boldsymbol{a_kx_{2k}}) = \boldsymbol{e_2} \\
\ldots\ldots\ldots \\
\boldsymbol{y_j} - (\boldsymbol{a_1x_{j1}} + \boldsymbol{a_2x_{j2}} + \ldots + \boldsymbol{a_kx_{jk}}) = \boldsymbol{e_j} \\
\ldots\ldots\ldots \\
\boldsymbol{y_n} - (\boldsymbol{a_1x_{n1}} + \boldsymbol{a_2x_{n2}} + \ldots + \boldsymbol{a_kx_{nk}}) = \boldsymbol{e_n}
\end{gathered}$$

Ocene nepoznatih parametara a_1, a_2, ..., a_k odeđuju se minimizacijom kriterijumske funkcije

$$\boldsymbol{J} = \sum_{i=1}^{n} \boldsymbol{e_i^2} = \sum_{i=1}^{n} [\boldsymbol{y_i} - (\boldsymbol{a_1x_{i1}} + \boldsymbol{a_2x_{i2}} + \cdots + \boldsymbol{a_kx_{ik}})]^2$$

Ukoliko je *n* veće, odnosno ukoliko je veći broj eksperimenata (uzoraka) utoliko će se tačnije odrediti vrednost parametrta a_1, a_2, ..., a_k.
Minimum funkcionala *J* dobija se izjednačavanjem njegovih parcijalnih izvoda po parametrima a_1, a_2, ..., a_k sa nulom, odnosno rešavanjem sistema od k jednačina.

$$\sum_{i=1}^{n} [\boldsymbol{y_i} - (\boldsymbol{a_1x_{i1}} + \boldsymbol{a_2x_{i2}} + \ldots + \boldsymbol{a_kx_{ik}})] * (\boldsymbol{x_{ij}}) = \boldsymbol{0}\,, \qquad \boldsymbol{j} = \boldsymbol{1}, \boldsymbol{2}, \ldots, \boldsymbol{k}$$

Međutim, na postupak metode najmanjih kvadrata u ovom slučaju je znatno jednostavnije primeniti matrični račun.
Obeležimo sa ***y*** n-dimenzioni vektor rezultata eksperimenta, sa **a** k-dimenzioni vektor nepoznatih parametara, a sa **X** matricu merenja ulaza eksperimenta dimenzije (n x k).

$$y = \begin{bmatrix} \boldsymbol{y_1} \\ \boldsymbol{y_2} \\ \ldots \\ \boldsymbol{y_n} \end{bmatrix} \qquad \boldsymbol{a} = \begin{bmatrix} \boldsymbol{a_1} \\ \boldsymbol{a_2} \\ \ldots \\ \boldsymbol{a_n} \end{bmatrix} \qquad \boldsymbol{X} = \begin{bmatrix} \boldsymbol{x_{11}} & \boldsymbol{x_{12}} & \ldots & \boldsymbol{x_{1k}} \\ \boldsymbol{x_{21}} & \boldsymbol{x_{22}} & \ldots & \boldsymbol{x_{2k}} \\ \ldots & \ldots & \ldots & \ldots \\ \boldsymbol{x_{n1}} & \boldsymbol{x_{n2}} & \ldots & \boldsymbol{x_{nk}} \end{bmatrix}$$

Kriterijumska funkcija se tada može predstaviti u obliku

$$\boldsymbol{J} = (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{a})^{\boldsymbol{T}}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{a}).$$

Primenom pravila diferenciranja za matrice i vektore dobija se:

$$\frac{\partial \boldsymbol{J}}{\partial \boldsymbol{a}} = 2\boldsymbol{X}^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{a}) = \boldsymbol{0}$$

Odatle se dobija

$$\boldsymbol{X}^{\boldsymbol{T}}\,\boldsymbol{X}\boldsymbol{a} = \boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{y}$$

odnosno

$$\boldsymbol{a} = \left(\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{y}$$

je relacija koja daje nepoznate parametre modela.

15. Validacija simulacionih modela

Problem validacije modela nastaje pošto se u fazi izgradnje modela unose mnoge aproksimacije realnog sistema. Aproksimacije koje se najčešće primenjuju su:

1. **Funkcionalna aproksimacija**
   Izrazito nelinerane funkcije često se aproksimiraju nekim jednostavnijim funkcijama, na primer linearnim. Važna pretpostavka je da jednostavnija funkcija treba da bude približna „originalnoj“ funkciji u oblasti gde će sistem verovatno funkcionisati. Ako program mora da funkcioniše i u oblasti gde je poklapanje funkcija slabo, potrebno je obezbediti da on štampa poruku upozorenja.

2. **Aproksimacija raspodele**
   Realne verovatnoće raspodela koje su i same poznate jedino aproksimativno, često se aproksimiraju jednostavnijim raspodelama, kao što su normalna ili eksponencijalna. Najekstremniji primer aproksimacije je onaj kada se slučajna promenljiva zameni konstantom.

3. **Aproksimacija nezavisnosti**
   Model se često pojednostavljuje tako što se pretpostavlja da su različite komponente (opisane slučajnim promenljivim) statistički nezavisne.

4. **Aproksimacija agregacije**
   Veoma čest tip aproskimacije. Pod agregacijom se podrazumeva situacija kada se više elemenata posmatra kao jedna celina. Neki tipični primeri agregacija su:

   a) Vremenska agregacija
   Interval vremena kao što je dan, tretira se kao jedan pojedinačni period. Za sve događaje koji se dese tokom dana, pretpostavlja se da su se desili istovremeno.
   b) Među-sektorska agregacija
   Više odeljenja, firmi, proizvodnih linija itd. posmatra se kao celina.
   c) Agregacija pomoćnih sredstava
   Više pomoćnih sredstava se posmatra kao jedno. Npr, ako posedujemo računarski sistem sa dva centralna procesora u paralelnoj vezi, nakon agregacije ćemo smatrati da posedujemo jedan, dva puta brži centralni procesor.

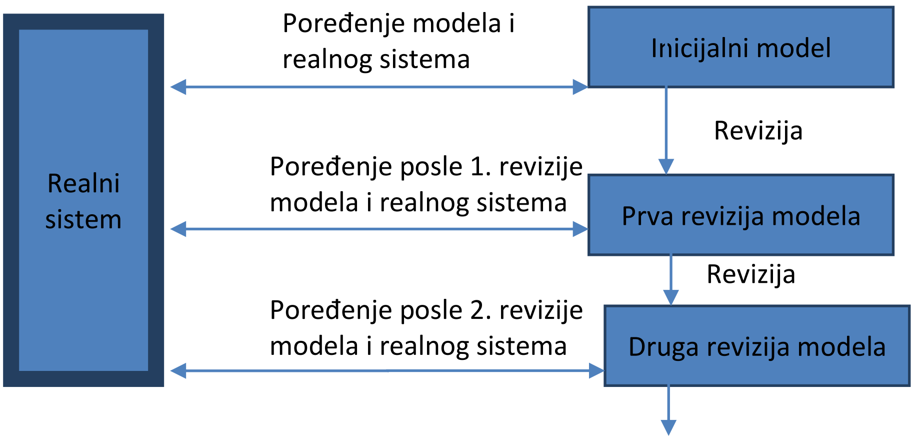
5. Aproksimacija stacionarnosti

Stvar pojednostavljuje činjenica da se parametri i druge karakteristike sistema ne menjaju u vremenu. Za izvesne fizičke procese, kao što su neke astronomske pojave, ovo može biti prihvatljiva početna aproksimacja. Međutim, u političkim ekonomskim, organizacionim i socijalnim sistemima, svakodnevno iskustvo ukazuje da je većina pojava nestacionarne prirode i da je kao posledica toga pretpostavka postojane strukture neodrživa.

Sve pretpostavke i aproksimacije koje unosimo u model mogu, na izvestan način, da dovedu u pitanje validnost tog modela, zato poredimo model i njegovo ponašanje sa realnim sistemom i njegovim ponašanjem.

Cilj procesa validacije:

- Da proizvede model koji predstavlja ponašanje realnog sistema i koji je dovoljno blizak realnom sistemu, tako da se može koristiti za obavljanje eksperimenta.
- Da pouzdanost modela poveća na prihvatljiv nivo, tako da model mogu koristiti različiti donosioci odluka.



- Iterativni proces validacije modela -

Proces validacije ne treba posmatrati kao izolovani skup mera i procedura koje slede nakon izgradnje modela, već kao integralni i nezamenjivi deo razvoja modela. Već je pomenuto da je proces validacije iterativni proces. Konceptualni model se poredi sa realnim sistemom i ukoliko postoje neslaganja i razlike za koje se smatra da nisu opravdane, vrši se ispravka na modelu, zatim se tako dobijeni model ponovo poredi sa realnim sistemom i vrše nove ispravke, ukoliko je to potrebno itd. Iterativni proces se nastavlja sve dok se ne postigne zadovoljavajuća tačnost modela.

Poređenje modela sa realnim sistemom se vrši pomoću više različitih testova:

- **Subjektivni testovi** najčešće podrazumevaju učešće ljudi koji dobro poznaju sistem koji primenjujući posebna znanja o sistemu donose zaključke o modelu i njegovom izlazu.
- **Objektivni testovi** uvek zahtevaju neke podatke o ponašanju sistema, kao i podatke koje proizvodi model. Prikupljeni podaci se obrađuju u različitim statističkim testovima koji porede određene aspekte skupa podataka i sistema i skupa podataka modela.

Praktičan pristup problemu validacije sastoji se iz tri faze:

1. Izgraditi model koji verno predstavlja realni sistem
2. Potvrditi pretpostavke modela
3. Uporediti ulazno-izlazne transformacije modela sa odgovarajućim ulazno-izlaznim transformacijama realnog sistema.

Procena validnosti modela

Prvi cilj modelara je da izgradi model koji će izgledati razumno korisnicima ili drugima koji poseduju znanja o realnom sistemu koji se simulira. Potrebno je da potencijalni korisnici modela budu uključeni u proces izgradnje modela od samog početka, kako bi se osiguralo da se u model "ugradi" visok stepen realizma. Posebno je važno da potencijalni korisnici, ali i drugi "ljudi od znanja", budu uključeni u proces iterativnog poboljšavanja modela, koji se sprovodi na osnovu uočenih neslaganja i odstupanja u postupku poređenja modela i realnog sistema. Na taj način se povećava poverenje u model.

Za proveru validnosti modela, može se koristiti i ***analiza osetljivosti.*** To je postupak testiranja osetljivosti modela na različite pretpostavke i promene ulaznih veličina.

Validacija pretpostavki modela

Pretpostavke modela je moguće svrstati u dve generalne kategorije: *pretpostavke o strukturi* i *pretpostavke o podacima*. *Pretpostavke o strukturi* se tiču pitanja kako funkcioniše sistem. *Pretpostavke o podacima* treba da budu zasnovane na skupu pouzdanih podataka i ispravnoj statističkoj analizi tih podataka. **Statistička analiza** se uglavnom sastoji iz 3 koraka:

1. Identifikacija odgovarajućih raspodela dobijenih podataka
2. Procena parametara izabrane raspodele
3. Validacija pretpostavljenog statističkog modela nekim testom

Validacija ulazno-izlaznih transformacija

Jedini objektivni test modela kao celine je provera sposobnosti modela da predvidi buduće ponašanje realnog sistema, kada ulazni podaci modela odgovaraju ulaznim podacima realnog sistema i kada je "politika" koja je implementirana u modelu, implementirana negde i u realnom sistemu. Struktura modela treba da bude dovoljno precizna da model daje dobra predviđanja.

U ovoj fazi valiadcije, model se posmatra kao ulazno-izlazna transformacija, tj. model prihvata vrednosti ulaza i transformiše ove ulaze u izlaze koji ukazuju na ponašanje modela.

Da bi se izvršila validacija ulazno-izlazne transformacije, neophodan uslov je postojanje sistema koji se proučava, kako bi bilo moguće prikupiti podatke o sistemu za barem jedan skup ulaznih podataka. Ukoliko je sistem u fazi planiranja i nije moguće prikupiti podatke, potpuna validacija ulazno-izlazne transformacije nije moguća.

16. Formalni kriterijum za utvrđivanje validnosti modela

Realan sistem treba posmatrati kao sistem opisan na višem nivou, dok model za koji treba utvrditi validnost treba da bude na nižem nivou opisa.

Homomorfizam predstavlja formalni kriterijum za utvrđivanje validnosti uprošćenog modela za date eksperimentalne uslove, u odnosu na osnovni model.

Suština formalnog postupka za proveru validnosti je u tome da se nađe preslikavanje *H* pomoću koga je moguće iz svakog stanja osnovnog modela (npr. *s*) preći u odgovarajuće stanje uprošćenog modela (npr. s’)

Da bi se utvrdio homomorfizam, moraju biti ispunjeni sledeći *uslovi*:

1. ***Očuvanje funkcije nastupanja vremena***

   Stanje u osnovnom modelu (*s*) i njemu odgovarajuće stanje u uprošćenom modelu (*s’*) moraju imati istu funkciju nastupanja vremena. To znači da će se stanja osnovnog modela i stanja uprošćenog modela menjati u istim trenucima vremena.

2. **Očuvanje funkcije prelaza stanja**

   Ako stanju *s* osnovnog modela odgovara stanje *s’* uprošćenog modela (tj. primenom preslikavanja *H* na stanje *s* dobija se stanje *s’),* tada i sledeća stanja u koja će model preći u narednom trenutku posmatranja moraju odgovarati jedno drugom. Sledeće stanje osnovnog modela će biti δ(s), a uprošćenog modela δ‘(s’), što znači da primenom preslikavanja *H* na stanje δ(s) treba da se dobije stanje δ‘(s’).

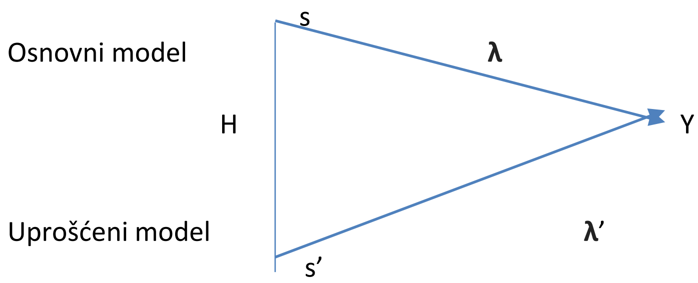
$$
\begin{tikzcd}
\text{Osnovni model} & s \arrow[r] \arrow[d, "H"] & \delta(s) \arrow[d, "H"] \\
\text{Uprošćen model} & s' \arrow[r] & \delta'(s').
\end{tikzcd}
$$

- Očuvanje funkcije prelaza stanja –

   Polazeći od stanja *s* osnovnog modela može se prvo vršiti preslikavanje *H* da bi se došlo u stanje *s’* uprošćenog modela, a zatim primeniti funkcija prelaza stanja uprošćenog modela da bi se došlo u stanje δ ‘(s’). Isto tako moguće je prvo primeniti funkciju prelaza stanja osnovnog modela, a zatim preslikavanje *H*. U oba slučaja dolazi se u isto stanje. Na osnovu izloženog može se zaključiti da preslikavanje *H* koje važi između stanja osnovnog i uprošćenog modela mora važiti i između njihovih budućih stanja koja se dobijaju na osnovu funkcije prelaza stanja.

3. ***Očuvanje izlazne funkcije***

   Neka je *s* stanje osnovnog modela i neka je moguće primenom preslikavanja *H* preći u stanje s’ uprošćenog modela. Primena izlazne funkcije osnovnog modela (za one eksperimentalne uslove za koje je izvršeno uprošćavanje (**λ)** da bi se dobio uprošćeni model za stanje s) , daje isti izlaz ***Y*** kao i primena izlazne funkcije uprošćenog modela (**λ’**) na stanje s’.



- Očuvanje izlazne funkcije –

Ako su zadovoljeni svi prethodno definisani uslovi, onda je uprošćeni model validan u odnosu na osnovni model, za date eksperimentalne uslove.

17. Verifikacija simulacionih modela

Proces ***verifikacije*** treba da pokaže da li je i u kojoj meri, konceptualni model na odgovarajući način predstavljen računarskim kodom, odnosno u kojoj se meri slažu konceptualni i računarski kod.
U postupku verifikacije nema standardnog recepta. Zato je potrebno izvršiti **više različitih provera**:

- **Ručna verifikacija logičke ispravnosti**:
  Model se izvesno vreme propušta na računaru i ručno, zatim se porede dobijeni rezultati.

- **Modularno testiranje**:
  Pojedinačno testiranje svakog modula kako bi se ustanovilo da li daje razumne izlaze za sve moguće ulaze.

- **Provera u odnosu na poznata rešenja**:
  Podesimo model tako da predstavlja sistem čija su rešenja poznata i upoređujemo ih sa reuzltatima modela.

- **Testiranje osetljivosti**:
  Variramo jedan parametar dok ostali ostaju nepromenjeni i proveravamo da li je ponašanje modela osetljivo na promenu tog parametra.

- **Testiranje na poremećaje**:
  Postavljamo parametre modela na neprirodne vrednosti i proveravamo da li se model ponaša na neshvatljiv način. Na taj način se mogu otkriti greške u programu koje je vrlo teško uočiti na drugi način.

18. Formalni opis sistema sa diskretnim događajima

Formalni opis sitema diskretnih događaja se može predstaviti uređenom šestorkom M_d:

$$\boldsymbol{M_d = (U, S, Y, \delta, \lambda, \tau)}$$

U – skup eksternih događaja
S – skup sekvencijalnih stanja diskretnih događaja
Y – skup izlaza
δ– kvazi-prenosna funkcija
λ– izlazna funkcija; preslikava S → Y

τ – funkcija nastupanja vremena; preslikava S → $R^{+}_{0,\infty}$

-pokazuje koliko će dugo sistem ostati u stanju s, pre nego što nastupi naredna promena stanja, pod pretpostavkom da neće nastupiti ni jedan eksterni događaj

19. Događaj, aktivnost i proces

Kod modela sistema sa disktretnim događajima, pored koncepta koji opisuju strukturu, kao što su objekti, relacije između njih i njihovi atributi, uvedeni su i koncepti za opis dinamike:

- događaj,
- aktivnost i
- proces.

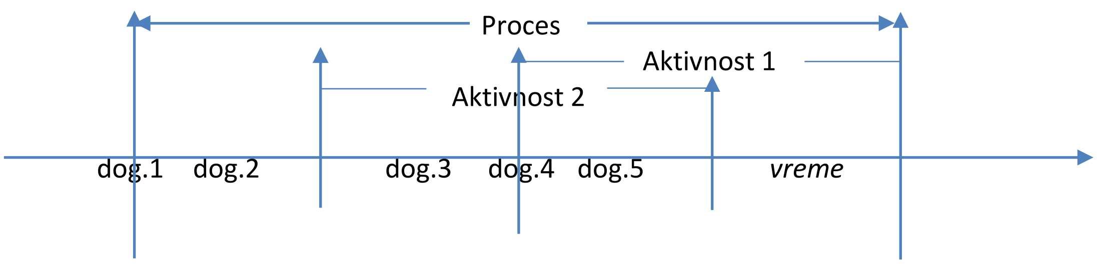
Događaj predstavlja diskretnu promenu stanja entiteta u sistemu ili njegovom okruženju. Između dva uzastopna događaja stanje sistema se ne menja. On može nastupiti zbog:

- ulaska/izlaska privremenog entiteta u odnosno iz sistema (dolazak novog klijenta, odlazak klijenta), ili
- promene atributa pojedinih objekata sistema (opslužilac u sistemu postaje slobodan ili zauzet).

Uslovni događaji mogu da nastupe tek kada je ispunjen uslov njegovog nastupanja. Obično su povezani sa zauzimanjem nekog resursa, odnosno početkom aktivnosti.

Aktivnost je skup događaja koji menjaju stanje jednog ili više entiteta. Trajanje aktivnosti se može unapred definisati, ali može da zavisi i od ispunjenja nekih uslova u modelu, tako da je vreme završetka takve aktivnosti nepoznato.

Proces je niz uzastopnih, logički povezanih događaja kroz koje prolazi neki privremeni objekat. Drugim rečima, proces je hronološki uređena sekvenca događaja koja opisuje jednu pojavu od nastajanja do terminiranja. U simulaciji, proces može da obuhvata deo ili celokupan „život“ privremenog entiteta.



- odnos događaja, aktivnosti i procesa -

20. Razvoj simulacije diskretnih događaja, mehanizam pomaka vremena

Ključne elemente razvoja simulacionih programa u simulaciji sistema sa diskretnim događajima čine:

1. **Mehanizam pomaka vremena**
2. **Pristupi generisanju događaja**
3. **Osnovne strategije simulacije**

1. **Mehanizam pomaka vremena**

U simulaciji sistema sa diskretnim događajima koriste se ***dva osnovna mehanizma pomaka vremena***:

- ***Pomak vremena za konstantni priraštaj***

  Vreme u simulacionom modelu se menja tako da se uvek dodaje konstantni priraštaj. Nakon svakog pomaka vremena, odnosno ažuriranja vrednosti simulacionog sata, ispituje se da li je u prethodnom intervalu vremena trebalo da dođe do nastupanja nekih događaja. Ukoliko jeste tada se ti događaji planiraju za kraj intervala.

  Ovaj pristup je najjednostavniji, ali ima i određene nedostatke. Pomeranjem događaja na kraj vremenskog intervala u kojem bi oni trebalo da nastupe uvodi se greška u izvođenju simulacije. Osim toga, događaji koji u stvarnosti nisu istovremeni u ovom se pristupu prikazuju kao istovremeni, a potom se određuje redosled njegovog izvođenja. Smanjenjem vremenskog prirasta te greške se smanjuju, ali se zato povećava vreme koje se troši na izvođenje simulacije. Pri tome je sve veći broj vremenskih intervala u kojima nema događaja, pa se oni uzalud prelaze u simulaciji.

  Ovaj mehanizam pomaka vremena može se uspešno primeniti i u situacijama kada se događaji zaista nižu jedan po jedan u konstantnim vremenskim intervalima.

- ***Pomak vremena na naredni događaj***

  Kod ovog pristupa, simulacioni sat se pomera na vreme u kojem će nastupiti prvi naredni događaj (ili više njih). U tom trenutku se događaj izvede i napravi se odgovarajuća promena stanja sistema; zatim se ponovo ispituje koji će događaj sledeći nastupiti, itd. Simulacija se završava kada nema više događaja ili kada je zadovoljen neki unapred definisani uslov završetka simulacije. Ovim pristupom se izbegava greška u vremenu izvođenja događaja, a ujedno se preskaču i intervali u kojima nema događaja.

  Iako dosta komplikovaniji za realizaciju, ovaj pristup je znatno efikasniji od prethodnog i ne unosi greške u vremenu izvođenja događaja. Zbog toga se i koristi u svim ključnim simulacionim jezicima i programskim paketima za simulaciju sistema sa diskretnim događajima.

21. Strategija raspoređivanja događaja u SDD

Mehanizam raspoređivanja događaja podrazumeva da se događaji planiraju unapred i drže u listi budućih događaja, najčešće sortirani po vremenu nastupanja i prioritetu.
Procedura planiranja događaja je sledeća: generiše se sled događaja, zatim se dodele vrednosti njegovih atributa. Atributi mogu biti: identifikatori tipa događaja, vreme njegovog nastupanja, prioritet kao i drugi. Lista budućih događaja je uređena po vremenu nastupanja događaja i njihovom prioritetu.
Funkcionisanje simulatora se odvija na sledeći način: sa liste budućih događaja uzima se prvi događaj. Tada se ažurira simulacioni sat na vreme njegovog nastupanja. Na osnovu tipa izabranog događaja poziva se odgovarajuća procedura koja izvršava sva ažuriranja u modelu i simulatoru, povezana sa nastupanjem izabranog događaja. Kad se izvrše svi događaji koji imaju isto vreme nastupanja, simulacioni sat se ažurira na vreme sledećeg događaja iz liste budućih događaja. Slika 7.7, strana 120.

22. Strategija skaniranja aktivnosti u SDD

Da bi se izvršila simulacija sistema sa diskretnim događajima, potrebno je realizovati računarski program koji skanira i raspoređuje aktivnosti modela. Skaniranje aktivnosti podrazumeva da se događaji implicitno raspoređuju tako da se promena stanja izvršava preko funkcija koje se nazivaju aktivnosti. Svaka aktivnost ima uslov i akciju.
Za svaki vremenski korak aktivnosti se skaniraju i traži se prva aktivnost koja ima zadovoljen uslov. Tada se izvršava odgovarajući programski segment koji specificira akciju za zadatu aktivnost. Proces skaniranja nastavlja se sve dotle dok sve aktivnosti ne budu blokirane. Onda i samo onda se simulacioni sat ažurira za sledeći vremenski korak.
Skaniranje aktivnosti ima prednosti nad raspoređivanjem događaja u slučajevima kad je broj aktivnosti u modelu mali, a broj događaja u okviru aktivnosti veliki. U tom slučaju se primenom skaniranja aktivnosti štedi na računarskim operacijama koje se odnose na stavljanje događaja na listu i njegovo skidanje sa liste događaja.

23. Strategija interakcije procesa u SDD

Interakcija procesa predstavlja tehniku simulacije koja je nastala kombinacijom raspoređivanja događaja i skaniranja aktivnosti.

S obzirom na to da model sistema predstavlja skup paralelnih procesa, od kojih neki mogu biti uzajamno isključivi, glavni problem je sinhronizacija procesa. Mogući konflikt koji proističe iz preklapanja procesa rešava se uvođenjem dve naredbe koje se upotrebljavaju pri specifikaciji događaja:

- **WAIT**
- **DELAY**

i to u oba konteksta, uslovnom i bezuslovnom.

WAIT naredba u uslovnom kontekstu ima oblik:

WAIT UNTIL <uslov>

Ukoliko je uslov zadovoljen, proces koji čeka taj uslov, nastaviće svoj tok. U suprotnom, biće blokiran.

Naredbom **DELAY** koja se u bezuslovnom konteksu u literaturi označava i sa

ADVANCE <broj vremenskih jedinica>

nastavak procesa odlaže se za specificiran broj vremenskih jedinica. Slika 7.9, strana 123.

-U okviru procesora održavaju se dve liste događaja:
1.Lista tekućih događaja – sadrži uslovne događaje i sortirana je po prioritetu;
2.Lista budućih događaja – sadrži raspoređene događaje koji tek treba da nastupe i sortirana je po vremenu nastupanja.

Pre početka simulacije za svaki proces se generiše prvi događaj i stavlja se u listu budućih događaja. Zatim se vreme simulacije ažurira na vreme događaja sa vrha liste budućih događaja.

-Rad procesora odvija se u dve faze:
1.Faza ažuriranja vremena simulacije
2.Faza skaniranja liste tekućih događaja

Kada se zadovolji kriterijum za završetak simulacije, vrši se završna obrada rezultata i procesor završava rad. Interakciju procesa koriste poznati simulacioni jezici **GPSS** i **SIMULA**.

24. Pojam sistema: definicija, karakteristike i primeri sistema

Sistem - skup delova koji funkcionišu zajedno radi ostvarenja zajedničkog cilja ili svrhe.
Sistem - *skup objekata ili elemenata povezanih relacijama na taj način da formiraju celinu. Objekti čine celinu radi zajedničkog cilja ili svrhe.*

Svaki sistem karakteriše se:

- svojstvima objekata koji čine taj sistem i
- vezama koje odražavaju uzajamnu zavisnost datog sistema i spoljašnje sredine.

Većina sistema deli **zajedničke karakteristike**, uključujući:

- Sistemi imaju strukturu, definisanu delovima sistema i njihovim vezama;
- Sistemi imaju ponašanje, koje uključuje ulaze, obradu i izlaze materije energije i informacija;
- Sistemi imaju unutrašnju povezanost, različiti delovi sistema imaju funkcionalne i strukturalne veze jedni sa drugima.

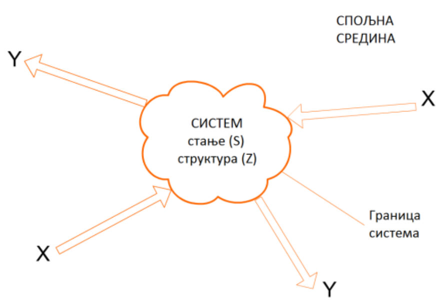
Otvoreni sistem vrši razmenu materije i energije sa okruženjem. Većina sistema spadaju u ovu klasu (automobil, aparat za kafu ili računar).
Zatvoreni sistem vrši razmenu energije, ali ne i materije sa okruženjem. (Zemlja)
Izolovani sistem ne razmenjuje ni materiju ni energiju sa okruženjem. (teoretski primer – univerzum)
Kontrolni sistem – skup fizičkih komponenti povezanih relacijama na taj način da komanduju, usmeravaju ili regulišu sebe ili neke druge sisteme.
Ostatak realnog sveta naziva se **spoljašna sredina** (okruženje) sistema.

25. Okruženje sistema, ulazi i izlazi sistema

Svaki sistem karakteriše se:

- **svojstvima objekata** koji čine taj sistem i
- **vezama** koje odražavaju uzajamnu zavisnost datog sistema i spoljašnje sredine.

Ostatak realnog sveta naziva se **spoljašna sredina** (okruženje) sistema.
Veze sistema sa sredinom – **ulazi i izlazi sistema**. U najopštijem smislu oni su nosioci materije energije i informacija koje se razmenjuju između posmatranog sistema i spoljašnje sredine.



Ulaz se nalazi na nekom toku i predstavlja pobudu spoljne sredine ili neophodnu razmenu sa spoljnom sredinom. Ulaz može ali ne mora biti iniciran ili pod kontrolom sistema.
Izlazi su stvarna reakcija sistema. Oni mogu ali ne moraju biti jednaki pobudi spoljne sredine preko ulaza. Izlaz zavisi od ulaza i stanja sistema. U opštem slučaju izlaz je različit od stanja sistema. On se ostvaruje na istom toku iza stanja sistema i njegova je funkcija.

26. Ciljevi sistema: opstanak, rast i razvoj

Kod najvećeg broja sistema poznati su ciljevi. Za veliki broj sistema ciljevi su vezani za opstanak, rast i razvoj.

- **Opstanak** – svaka jedinka se bori za svoj opstanak, samostalno ili uključena u neki drugi sistem. Opstanak kod preduzeća znači obezbediti i održavati unutrašnje odnose i odnose sa spoljnom sredinom tako da preduzeće može da postoji.
- **Rast** – je povezan sa integracijom više jedinki u nov sistem koji lakše obezbeđuje pre svega opstanak, ali i druge prednosti. Rast kod preduzeća opet znači povećanje njegovih potencijala i stepena efikasnosti u korišćenju tog potencijala.
- **Razvoj** je ono što se uočava kroz evoluciju i što obezbeđuje nove kvalitete.
- **Cilj** je uopšte uzevši, određen željenim stanjima i izlazima sistema u određenom trenutku ili intervalu vremena. Cilj je bliže određivanje ostvarenja svrhe preko potrebnih stanja i izlaza sistema.

27. Stanje sistema, prostor stanja sistema i oblast dopuštenih stanja sistema

- ***Stanje sistema*** – skup podataka koji daju potpunu informaciju o predistoriji i sadašnjem stanju atributa objekata sistema. Ako je moguće te informacije kvantifikovati dobija se skup svih veličina $S_1, S_2, ..., S_n$, koji određuje stanje sistema kao vektorsku veličinu Euklidovog n-dimenzionog prostora.
- Prostor u kome se svako stanje prikazuje određenom tačkom naziva se ***prostor stanja sistema***. Broj dimenzija prostora stanja je jednak broju atributa objekata sistema koje određuje njegovo stanje. U svakom vremenskom trenutku stanje sistema se može prikazati tačkom u prostoru stanja sistema.
- ***Oblast dopuštenih stanja*** - oblast prostora stanja u kojoj se može naći tačka stanja.

28. Struktura sistema: pojam i karakteristike, adaptivni sistemi

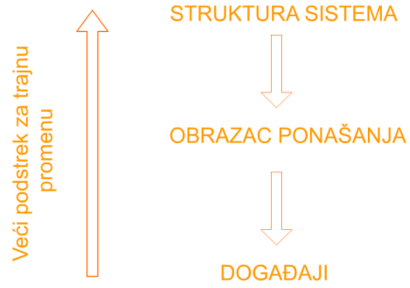
- ***Struktura sistema*** – opšti kvalitativno određen i relativno stabilan poredak unutrašnjih odnosa između elemenata sistema. Strukturu sistema određuje njegova svrha, odnosno cilj, uzimajući u obzir sve prirodne i društvene važeće relacije između objekata.
- ***Adaptivni sistemi*** - oni koji imaju sposobnost prilagođavanja strukture prema okolini na način koji je povoljan za nastavljanje postojanja.

29. Sistemsko razmišljanje i dinamika sistema

Pošto je naše zanimanje za poslovne procese, posmatraćemo sisteme ljudi i tehnologija namenjene da osmisle, trguju, proizvedu i distribuiraju proizvode i usluge. Skoro sve što se dešava u biznisu je deo jednog ili više sistema. Umesto da pretpostavimo da je neki problem upravljanja izazvao neki spoljašnji događaj, sa sistemskim pristupom zauzimamo drugačije gledište - *unutrašnja struktura sistema je često važnija za ponašanje nego spoljašnji događaji*.

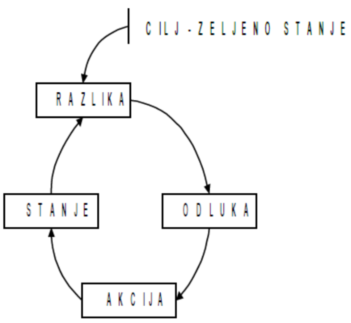
Ako se prebacimo sa orijentacije na događaj i usredsredimo na unutrašnju strukturu sistema, povećavamo mogućnost da usavršimo ponašanje poslovnog sistema. To je zato što je struktura sistema često suštintski izvor poteškoće. Ukoliko ne ispravimo nedostatke strukture sistema, verovatno je da će se problem ponovo pojaviti ili će biti zamenjen jos složenijim problemom.

Dinamika sistema je metodologija koja nam pomaže da shvatimo kako se sistem menja u vremenu. Dinamika sistema povezuje ponašanje sistema sa njegovom strukturom. Dinamika sistema se može koristiti za analizu uticaja strukturalnih promena u jednom delu sistema na ponašanje celokupnog sistema.



30. Povratno dejstvo, kolo povratnog dejstva, polaritet KPD

- Povratno dejstvo – relacije između elemenata mogu biti takve da jedan element posredno, preko drugih elemenata, utiče sam na sebe. Za sisteme kod kojih je to slučaj kažemo da poseduju povratno dejstvo.
- Sistem sa povratnim dejstvom:
  - ulaz zavisi ili je kontrolisan ranijim događajima u njemu (zavisi od izlaza ili nekog elementa koji se kontroliše);
  - sistem sa povratnim dejstvom ima strukturu sastavljenu od kola povratnog dejstva kroz koje prošle aktivnosti kontrolišu buduće.
- Povratno dejstvo koje povećava uticaj poremećaja na ulaz zove se pozitivno, a ono koje smanjuje zove se negativno.
- **Upravljačko kolo povratnog dejstva** je zatvoreni niz uzročno-povezanih elemenata kod kojih se bar jedan mora odnositi na *upravljački proces* (posmatranje i odlučivanje), jedan na *akcije* realizacije odluke, a jedan na *stanje sistema* kojim se upravlja (izlaz ili potencijal sistema). Ono ima sledeći raspored elemenata u sistemu:
  - odluka koja kontroliše akciju,
  - akcija,
  - stanje sistema koje se menja akcijom,
  - odluka.



Odluka i akcija predstavljaju element promene stanja sistema.

- ***Prirodno kolo povratnog dejstva*** – kolo koje je posledica prirode sistema. Ono ima sledeći raspored elemenata u sistemu:
  - element sistema I
  - element sistema II
  - element sistema III

- element sistema I

Dejstvo jednog elementa na drugi dešava se u vremenu. Promena na bilo kom elementu kola povratnog dejstva usloviće sukcesivne promene povezanih elemenata na zatvorenoj putanji kola. (Primer: kola povratnog dejstva za sitem zaliha prvog i drugog reda).

Polaritet KPD:

- Kolo povratnog dejstva kod kog impuls uslovljava promenu veličine posmatranog elementa na taj način što se predznak priraštaja na njegovoj veličini menja iz jednog u drugi ciklus prolaza impulsa naziva se negativnim.
- Kolo povratnog dejstva kod kog impuls uslovljava promenu veličine posmatranog elementa na taj način što se predznak priraštaja na njegovoj veličini neće menjati iz jednog u drugi ciklus prolaza impulsa naziva se pozitivnim.
- Pojava pozitivnih kola povratnog dejstva je posledica prirode međusobnih zavisnosti elemenata sitema. Akcije unutar pozitivnog kola povećavaju razliku između stvarnog i željenog stanja sistema (primer: razmnožavanje bakterija).
- U negativnom kolu povratnog dejstva kontrolnom odlukom se želi dovesti stanje sistema na vrednost datu ciljem koji se formira izvan samog kola. U ovim kolima nastaje smanjenje priraštaja ako se stanje približava željenoj veličini, odnosno, povećava se ako se veličina stanja udaljava od željene veličine.

31. Karakteristike sistema sa i bez kola povratnog dejstva

Osnovne klase sistema koji se modeliraju i simuliraju u dinamici sistema su **sistemi sa povratnim dejstvom**. Ukoliko u nekom sistemu postoje relacije između elemenata sistema takve da jedan element posredno, preko drugih elemenata utiče sam na sebe, kažemo da u takvom sistemu postoji povratno dejstvo (*feedback loop*). U suprotnom, reč je o sistemu bez povratnog dejstva. Drugim rečima, **povratno dejstvo** je *zatvoreni krug uzroka i posledica koji utiče na to da neki početni uzrok ima indirektan uticaj na samoga sebe*. Postojanje jednog ili više kola povratnog dejstva u sistemu izaziva njegovo složeno ponašanje u vremenu.

Sistem sa povratnim dejstvom je onaj čiji *ulaz zavisi ili je kontrolisan ranijim događajima u njemu*, odnosno zavisi, na neki način, od izlaza ili nekog elementa sistema koji se kontroliše.

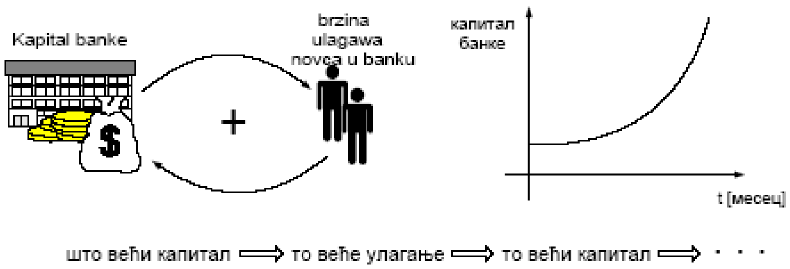
Sistem sa povratnim dejstvom ima strukturu u kojoj postoji barem jedno kolo povratnog dejstva kroz koje prošle aktivnosti kontrolišu buduće. Kola povratnog dejstva mogu biti pozitivna i negativna.

Sistem bez povratnog dejstva karakteriše se time što njegov izlaz zavisi od ulaza, ali ulaz ne zavisi od izlaza. U sistemima ove klase, prošli događaji ne utiču na buduće i sam sistem ne kontroliše svoje ponašanje, odnosno oni funkcionišu bez saznanja o rezultatima koje ostvaruju. Tačnost funkcionisanja ovih sistema određena je samo kvalitetom međusobne zavisnosti elemenata koji ih sačinjavaju. Kod ovih sistema ne javljaju se problemi vezani za nestabilnost koji se javljaju kod sistema sa povratnim dejstvom. Većina tehničkih sistema pripada ovoj klasi.

32. Pozitivno kolo povratnog dejstva: pojam, osobine, primeri

Povratno dejstvo koje povećava uticaj poremećaja na ulaz naziva se pozitivno. **Pozitivno povratno dejstvo** znači da neki uzrok, preko lanca posledica dovodi do promena uvek u istom smeru. Tako se stalnim povratnim delovanjem postiže stalni rast ili stalno smanjenje tog uzroka. Pojava kola pozitivnog povratnog dejstva u sistemu je posledica prirode međusobne zavisnosti elemenata sistema. Akcije unutar kola pozitivnog povratnog dejstva stalno povećavaju razliku između stvarnog i željenog stanja sistema. Ova kola u sistemu imaju ulogu u ostvarivanju ciljeva koji se odnose na rast i razvoj.

Jednostavan primer pozitivnog kola povratnog dejstva je ono u kome se prati uticaj brzine ulaganja novca na veličinu kapitala banke. Pretpostavimo da u početnom trenutku posmatranja banka raspolaže određenim kapitalom. Visina tog kapitala određuje položaj banke na tržištu novca. Veći kapital znači prednost za banku jer privlači ulagače, na primer preko veličine kamatne stope koju banka određuje ili preko garancija koju banka pruža u pogledu sigurnosti uloga. Uz određenu stopu ulaganja, moguće je za dati (početni) kapital banke odrediti količinu novca koja će biti uložena u banku u posmatranoj jedinici vremena (npr. mesec), koja će povećati početni kapital banke. Zbog povećanog kapitala banke, na početku narednog meseca, količina uloženog novca u tom mesecu biće još veća, što će dalje povećati kapital banke, itd.

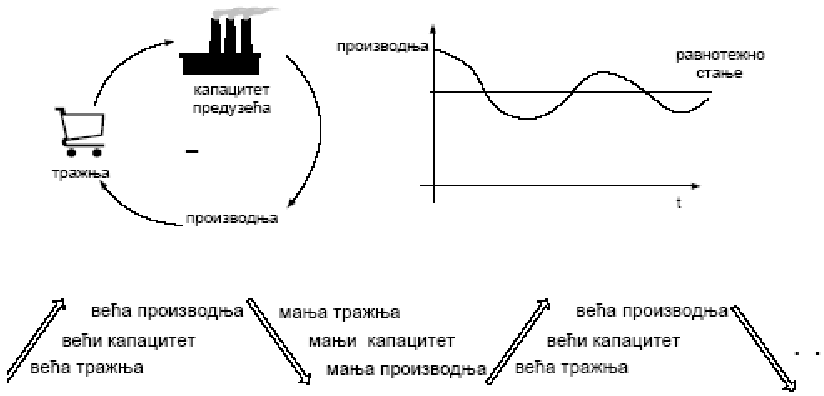


Слика 11.1 Позитивно коло повратног дејства

33. Negativno kolo povratnog dejstva: pojam, osobine, primeri

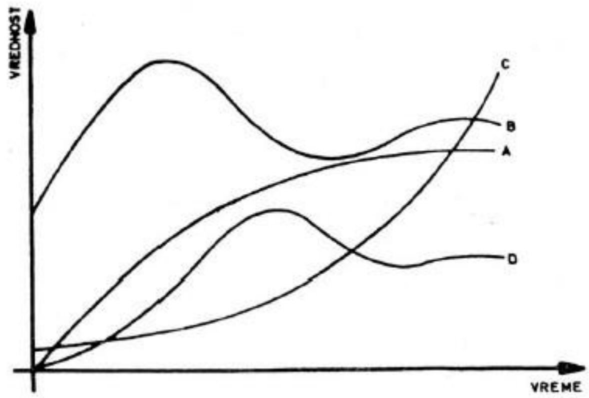
Negativno povratno dejstvo je ono koje smanjuje uticaj poremećaja na ulaz. Ono omogućava uspostavljanje ravnotežnog stanja u sistemu, kada je ono narušeno nekim poremećajem. Negativno povratno dejstvo zapravo znači da neki uzrok, preko lanca posledica dovodi do promene smera sopstvenog delovanja. Ukoliko posmatrani uzrok poraste iznad nekog ravnotežnog stanja, povratno dejstvo će smanjiti taj uzrok i obratno, kada uzrok postane manji od ravnotežnog stanja, tada će povratno dejstvo prouzrokovati povećanje tog uzroka. Na taj način pozitivno kolo povratnog dejstva "stabilizuje" uzrok oko nekog ravnotežnog stanja.

Jednostavan primer u kome se posmatra uzajamni uticaj tražnje za proizvodima nekog preduzeća, kapaciteta preduzeća i proizvodnje. Povećana tražnja na tržištu za proizvodima preduzeća utiče na povećanje ulaganja u proširenje kapaciteta preduzeća. Veći kapaciteti omogućavaju veću proizvodnju traženih proizvoda, a veća proizvodnja brže zadovoljava tražnju, koja se dalje smanjuje. Smanjenje tražnje u narednom ciklusu posmatranja, usloviće manje angažovanje raspoloživih kapaciteta, odnosno manju proizvodnju, što će ponovo dovesti do veće tražnje, itd.



Слика 11.2 Негативно коло повратног дејства

Opisana povratna dejstva po pravilu se ne ostvaruju u kratkim lancima uzroka i posledica, kakvi su prikazani u prethodnim primerima. U realnom sistemu, ona zapravo nastaju u znatno dužem i složenijem nizu uzroka i posledica. Takođe, realni sistemi najčešće sadrže mešavinu pozitivnih i negativnih kola povratnog dejstva koja određuju njihovo ponašanje u vremenu. Njihova analiza podrazumeva određivanje onih faktora koji dominantno utiču na ponašanje sistema.



Слика 11.3 Неки типични облици понашања система са повратним дејством

Ne traži se direktno ali nije teško pa možete da pročitate

Na slici 11.3 prikazani su neki ***karakteristični oblici ponašanja sistema sa povratnim dejstvom***.

- **Kriva A** je tipična kriva za jednostavniji sistem sa povratnim dejstvom kod koga promenljiva raste sa opadanjem veličine priraštaja (odnos veličine izlaza i ulaza). Posle izvesnog vremena priraštaj teži nuli i kriva se asimptotski približava nekoj vrednosti. Primer ovakvog ponašanja je rast u zapošljavanju. U sistemu dominira negativno kolo povratnog dejstva.
- **Kriva B** opisuje složenije dostizanje konačne vrednosti (oscilovanje oko konačne vrednosti). Može nastati zbog preteranog kašnjenja u kolu povratnog dejstva ili zbog nagle korekcije razlike između stvarnog i željenog stanja. Primeri ovakvog ponašanja su: rast ili opadanje industrijske proizvodnje unutar ekonomskih ciklusa i promena cene uslovljena zakonom ponude i potražnje. Sistemom dominira *negativno kolo povratnog dejstva*.
- **Kriva C** prikazuje eksponencijalni rast. Primeri ovakvog ponašanja su: razmnožavanje ćelija i rast prodaje koji zavisi samo od broja prodavaca. Dominantan uticaj ima *pozitivno kolo povratnog dejstva*.

- **Kriva D** prikazuje početni eksponencijalni rast, a potom fluktuacije u dostizanju neke vrednosti. Primeri ovakvog ponašanja su: ponašanje broja životinja i rast novog proizvoda. U početku u sistemu *dominira pozitivno, a potom negativno kolo povratnog dejstva*.

Osnovni pojmovi sistema sa povratnim dejstvom (ovo je zajedničko za pitanja 32 i 33 - može i ne mora)

Za opisivanje sistema sa povratnim dejstvom, u dinamici sistema upotrebljavaju se sledeći osnovni pojmovi:

- stanja sistema
- promene stanja sistema
- kašnjenja

♦ ***Stanja sistema (level - nivo)*** predstavljaju stanja resursa, odnosno različite akumulacije resursa. Stanja su merljive veličine, a njihove dimenzije izražavaju se u jedinicama resursa. Neki primeri stanja sistema su: broj stanovnika, količina vode u rezervoaru, količina robe na zalihama, novac na računu itd. Stanjima sistema se može upravljati i na taj način uticati na brzinu njihove promene i na njihovo ravnotežno stanje. Svoju vrednost, ove veličine imaju i kada sistem ne funkcioniše.

♦ ***Promene stanja sistema (rate – brzina)*** ili funkcije odlučivanja su akcije ili tokovi koji dovode do povećanja ili smanjenja stanja sistema, u jedinici vremena. To su promenljive koje pokazuju brzinu pretvaranja resursa iz jednog u drugo stanje. U dinamici sistema, ove se veličine posmatraju kao prosečne brzine promene u nekom periodu vremena, a izražavaju se u jedinicama resursa u jedinici vremena.

♦ ***Kašnjenja*** predstavljaju konsekvence promena koje se ne dešavaju istovremeno kada su pokrenute. Naime, u tehničkim, ekonomskim, društvenim ali i u većini drugih sistema, promene se ne dešavaju istovremeno kada su pokrenute, već je neophodno da protekne određeno vreme (kašnjenje) da bi se promena izazvana na jednom elementu u sistemu prenela na drugi element, koji je sa njim povezan.

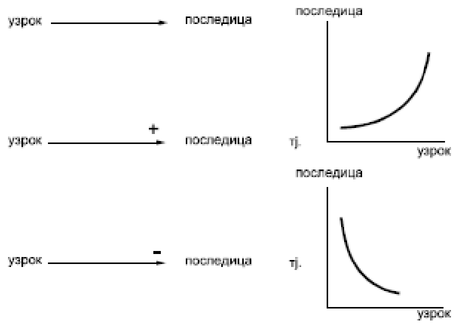
Jedan deo kašnjenja se odnosi na **materijalna kašnjenja**, a drugi na **kašnjenja informacija**. Kašnjenja u sistemu mogu imati značajne posledice na ponašanje sistema, i ona po pravilu izazivaju oscilacije u ponašanju sistema. U većini slučajeva kada je moguće smanjiti kašnjenja, smanjiće se i oscilacije i sistem će raditi ujednačenije.

34. Dijagrami uzročno-posledičnih veza

Konceptualni modeli omogućavaju eksplicitni prikaz elemenata sistema i relacije koje između njih postoje u grafičkom obliku. Na taj način oni olakšavaju razvoj simulacionih (računarskih) programa i povećavaju njihovu pouzdanost. U dinamici sistema koriste se dve klase konceptualnih modela:

- diajgrami uzročnih petlji
- dijagrami tokova.

U analizi sistema sa povratnim dejstvom uobičajeno je da se najpre odrede veze između elemenata sistema i da se identifikuju vrste tih veza, a potom identifikuju povratna dejstava u sistemu i odredi njihov polaritet. Najjednostavnije je da se to izrazi u grafičkom obliku. Za prikazivanje uzročno-posledičnih veza između elemanata sistema sa povratnim dejstvom u dinamici sistema koriste se ***dijagrami uzročnih petlji***. Grafički simboli ovih dijagrama prikazani su na slici 11.3.



Слика 11.3 Приказивање релација у дијаграмима узрочних веза

Linija sa strelicom označava smer veze između uzroka i posledice. Ako je na kraju strelice znak plus, to znači da se uzrok i posledica menjaju u istom smeru, a ako je na kraju strelice znak minus, tada se uzrok i posledica menjaju u suprotnom smeru.

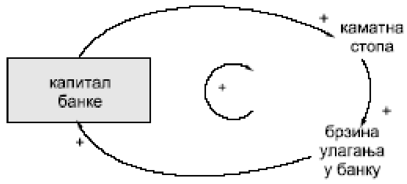
Polukružna strelica u sredini petlje sa znakom plus označava pozitivno povratno dejstvo, a sa znakom minus negativno povratno dejstvo. Kod složenijih kola, tip povratnog dejstva se najlakše može odrediti na sledeći način:

Samo pozitivne veze: pozitivno povratno dejstvo

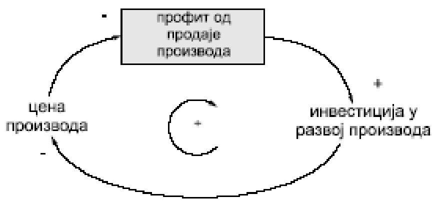
Paran broj negativnih veza: pozitivno povratno dejstvo

Neparan broj negativnih veza: negativno povratno dejstvo

Na slici 11.4 i 11.5 prikazano je nekoliko jednostavnijih primera dijagrama uzročnih petlji kod sistema sa pozitivnim i negativnim povratnim dejstvom.U praksi, retko se sreću sistemi koji se sastoje samo od jedne petlje povratnog dejstva, kako je to prikazano u prethodnim primerima, već je uobičajeno da se sistemi opisuju većim brojem petlji koje su povezane preko zajedničkih elemenata koji u njima učestvuju.

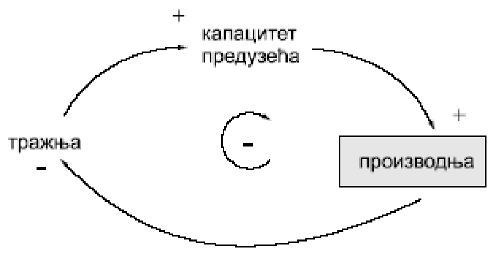


(а) Пораст капитала банке везан за пораст каматне стопе



(б) Предузеће које инвестира у развој производа

Слика 11.4 Неки примери позитивног повратног дејства

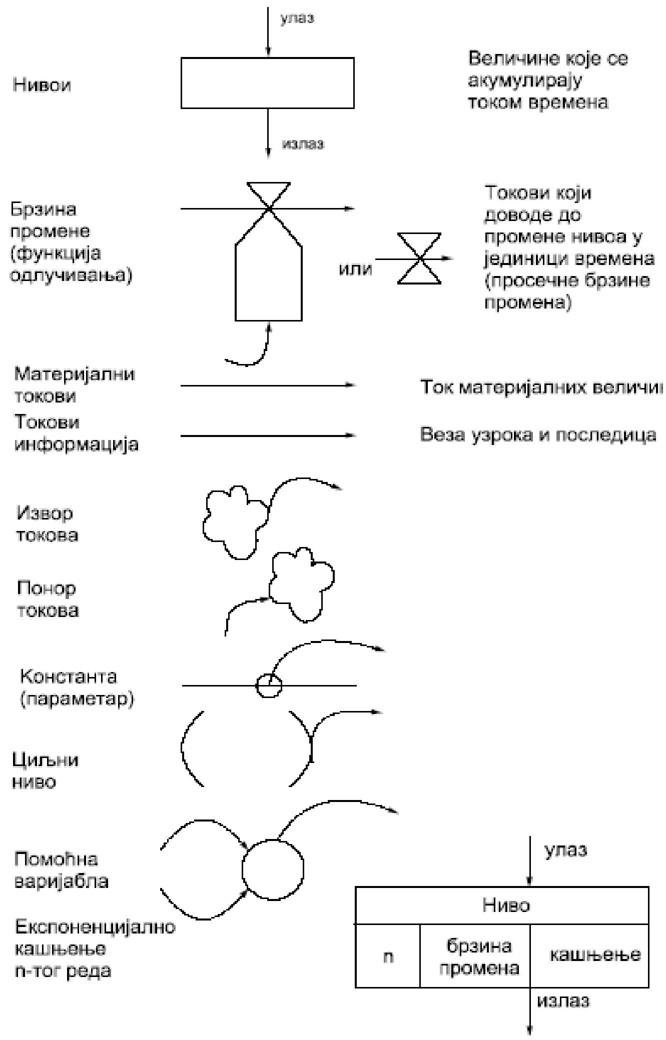


Тражња и капацитет производње

Слика 11.5 Пример негативне повратног дејства везе

35. Dijagrami skladišta i tokova

Konceptualni modeli kojima se identifikuju stanja sistema, promene stanja sistema i kašnjenja u sistemu nazivaju se ***dijagrami tokova*** i služe za preciznije grafičko predstavljanje posmatranog sistema. Ovi modeli čine osnovu za pisanje računarskih modela u dinamici sistema. Osnovni simboli koji se koriste za konstruisanje dijagrama toka dati su na slici 11.6.



Слика 11.6 Елементи дијаграма тока

Linijama se označavaju tokovi u sistemu. Njima se definiše način na koji resursi u sistemu prelaze iz jednog stanja u drugo. Materijalni tokovi označavaju se punom, a informacioni isprekidanom linijom. Svaki tok ima odgovarajući simbol za izvor (početak toka izvan sistema) i ponor (odredište toka izvan sistema).

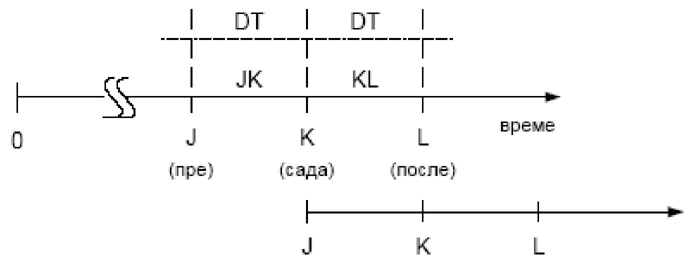
Stanja sistema su veličine koje se akumuliraju tokom vremena. Na dijagramu toka predstavljaju se pravougaonicima unutar kojih se upisuje njihov naziv. Promene stanja određuju protoke na tokovima. To su prosečne brzine (promena stanja u jedinici vremena) koje se često nazivaju i funkcije odlučivanja pošto se preko njih upravlja stanjima sistema. Promene stanja koje se nalaze na ulaznom toku stanja sistema, nazivaju se ulazne promene stanja, dok su izlazne promene stanja one koje se nalaze na izlaznom toku tog stanja. Konstante su veličine u modelu koje se ne menjaju tokom vremena. Pomoćne promenljive se nalaze na tokovima informacija i služe za opis elemenata promene stanja. Javljaju se kao izdvojeni elementi jer imaju svoje nezavisno značenje.

36. Računarski modeli u dinamici sistema

Na osnovu izrađenog dijagrama toka koji predstavlja detaljan konceptualni model sistema, stvara se računarski model sistema. ***Računarski modeli dinamike sistema*** prikazuju kvantitativne veze između elemenata sistema i omogućuju izvođenje planiranih simulacionih eksperimenata sa modelom sistema. Oni se prikazuju u obliku sistema diferencnih jednačina. Jednačine modela prikazuju promene vrednosti promenljivih modela između uzastopnih, vremenskih trenutaka, udaljenih za izabrani, konstantni, vremenski korak. Da bi se izvela simulacija, odnosno dobilo ponašanje sistema u vremenu, potrebno je rešiti sistem diferencnih jednačina računarskog modela, za šta se koriste numeričke u1084 metode integracije.

<u>Opisivanje promene vremena u jednačinama modela</u>

Stanja sistema u dinamici sistema izračunavaju se u diskretnim trenucima vremena, recimo t = 0,1,2, ... , T, međusobno udaljenim za konstantan vremenski interval *DT*. Veličinu *DT* bira modelar i od nje zavisi tačnost rezultata i vreme izvođenja simulacije. Matematički gledano, ovaj pristup podrazumeva korišćenje diferencnih jednačina. Vremenska osa u dinamici sistema sadrži tri uzastopna, ekvidistantna trenutka vremena (t-1, t , t+1), koji se označavaju sa **j**, **k** i **l** (slika 11.7).



Слика 11.7 Временска оса у динамици система

Vremenski trenutak j <u>predstavlja prethodno</u>, trenutak k <u>sadašnje</u>, a trenutak l <u>buduće vreme</u>. Osnovni vremenski interval je *<u>vreme između dva uzastopna vremenska trenutka</u>* i označava se sa dt. Lako je uočiti da je:

$$k = j + dt \quad \text{i} \quad l = k + dt$$

Interval vremena između trenutaka j i k obeležava se sa jk, a interval između k i l sa kl. Osnovni interval vremena određuje se na osnovu raspoloživih podataka i može predstavljati dan, nedelju mesec, kvartal, godinu itd.

Vrednosti stanja sistema obeležavaju se na sledeći način: **stanje.** $\boldsymbol{j}$; **stanje.** $\boldsymbol{k}$; **stanje.** $\boldsymbol{l}$

Vrednosti promena stanja predstavljaju prosečne brzine promena stanja i između dva uzastopna vremenska trenutka. Tretiraju se kao konstante, a označavaju se na sledeći način: promena **stanja.** $\boldsymbol{jk}$; promena **stanja. kl**.

Sada razmotrimo sledeću jednačinu:

$$\boldsymbol{y_t = y_{t-1} + \Delta y}$$

Ukoliko je brzina promene između $t-1$ i t konstantna, tada sledi:

$$\boldsymbol{\frac{\Delta y}{\Delta t} = c(t-1, t)}$$

Kombinovanjem gornja dva izraza dobija se:

$$\boldsymbol{y_t = y_{t-1} + \Delta y \cdot c(t-1, t)}$$

Prevođenjem gornjeg izraza u simboličku notaciju koja se koristi u dinamici sistema, dobija se: **stanje.**$\boldsymbol{k}$ **= stanje.**$\boldsymbol{j}$ **+** $\boldsymbol{dt}$ (*promena.*jk) što predstavlja opštu jednačinu za određivanje vrednosti elemenata stanja sistema. Ukoliko posmatrano stanje sistema ima i ulaz i izlaz, tada će brzina promene u vremenu biti jednaka razlici ulaznih i izlaznih tokova, odnosno **promena.**$\boldsymbol{jk}$ $=$ **ulaz.**$\boldsymbol{jk}$ $-$ **izlaz.**$\boldsymbol{jk}$

Rešavanje jednačina stanja i jednačina promene stanja u vremenu, vrši se u sledećim koracima:

1. Vreme se pomera na trenutak $\boldsymbol{k}$.
2. Izračunavaju se vrednosti stanja sistema u trenutku $\boldsymbol{k}$. Za proračun se koriste jednačine stanja sistema.
3. Izračunavaju se vrednosti svih promena stanja za naredni interval $\boldsymbol{l}$ (na osnovu dobijenih vrednosti stanja sistema u trenutku $\boldsymbol{k}$).
4. Preimenuju se stanja sistema i promene stanja sistema: $\boldsymbol{k} \rightarrow \boldsymbol{j}$; $\boldsymbol{l} \rightarrow \boldsymbol{k}$, da bi se omogućilo korišćenje istih jednačina.
5. Vreme se pomera za $\boldsymbol{dt}$ i novi sadašnji trenutak ponovo nazivamo $\boldsymbol{k}$.

Ceo opisani ciklus se ponavlja dok se simulacija ne završi (ne postigne zadato vreme trajanja simulacije). Da bi proračun mogao da otpočne, neophodne su početne vrednosti za sva stanja sistema, kao i vrednosti za sve konstante i parametre sistema. Na taj način dobija se uzastopni niz vrednosti promenljivih modela u vremenskim tačkama, međusobno udaljenim za DT.

37. Modeliranje sistema u programskom jeziku SDS

SDS je programski paket za simulaciju dinamike sistema. Prvenstveno je namenjen za simulacione modele koji se mogu opisati elementima stanja, elementima promene stanja i diskretnom promenom vremena, pri čemu se može koristiti Euler-ova metoda ili metoda Kutta-Merson-a, koja omogućava kontrolu tačnosti računanja za numeričko rešavanje sistema diferencnih jednačina. Takođe, sa uspehom se može koristiti za rešavanje sistema diferencijalnih jednačina tipa:

$$\boldsymbol{\frac{dx}{dy} = f_i(x, y_1, y_2, \dots, y_{n)}) \quad i = 1, 2, \dots, n}$$

Programski paket SDS omogućuje:

- simulaciju ponašanja sistema i
- analizu ponašanja sistema.

38. Test funkcije: PULSE, RAMP, STEP.

Impulsna funkcija

PULSE (PLH, PLT)
PLH – veličina impulsa
PLT – vreme starta impulsa
PULSE = 0 sem u trenutku PLT kada ima vrednost PLH.

Odskočna funkcija

STEP (STH, STT)
STH – veličina skoka
STT – vreme starta skoka
Do vremena STT vrednost funkcije je 0, a zatim je STH.

Linearno rastuća funkcija

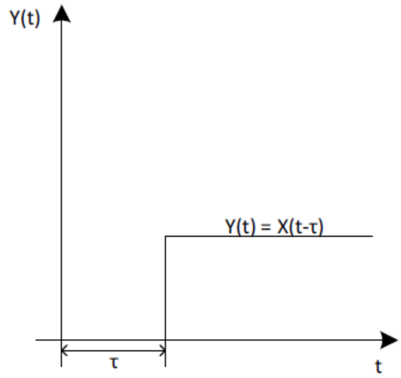
RAMP (RPSL, RPT)
RPSL - priraštaj
RPT – vreme starta
Vrednost funkcije je 0 do vremena RPT, a zatim linearno raste sa priraštajem RPSL.

39. Kašnjenje: pojam, osobine, vrste

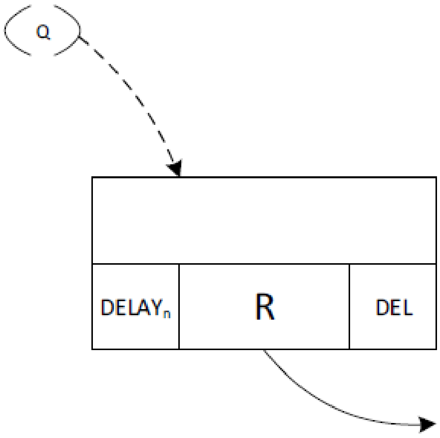
Kašnjenje – vreme potrebno da se promena izazvana na jednom elementu (u sistemu) kola prenese na drugi element.

- U tehničkim, ekonomskim i društvenim sistemima promene se ne dešavaju istovremeno kada su pokrenute (na primer nabavka robe).
- Deo kašnjenja se odnosi na kašnjenja materijala, a deo na kašnjenja informacija.
- Na neka kašnjenja se može uticati (brzina obrade porudžbina), a na neke ne (vreme potrebno pošti da dostavi porudžbinu).
- Imaju značajne posledice na ponašanje sistema i po pravilu uzrokuju oscilacije u ponašanju sistema.
- Ako se smanje kašnjenja, smanjuju se i oscilacije u sistemu (skladište sa drugim vremenom od trenutka slanja porudžbine do trenutka stizanja robe ima velike oscilacije u količini robe na skladištu).

I materijali i informacije šire se određenom (konačnom) brzinom što znači da postoji određeno vreme kašnjenja od trenutka slanja materijala i informacija do trenutka njihovog stizanja (na primer vreme potrebno za stizanje naručenih delova, stizanje informacija o povećanoj potražnji za proizvodima, trajanje gradnje stanova i slično).
Što je više različitih tipova kašnjenja u sistemu – više utiču na ponašanje sistema.
Neka kašnjenja u stvarnim sistemima mogu se prikazati jednostavnim pomakom vremena za konstantni vremenski interval, ali ima i kašnjenja koja su posledica agregiranja velikog broja događaja i koja imaju oblik postepenog reagovanja na inicijalnu promenu.



40. Modeliranje kašnjenja na materijalnim tokovima u SDS jeziku

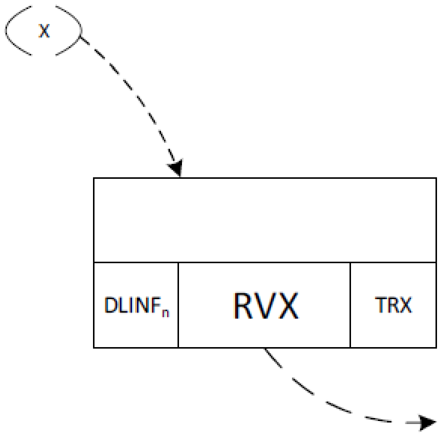


STANJE.K = STANJE.J + DT*(ULAZ.JK – IZLAZ.JK)
IZLAZ.KL = DELAY (RED KAŠNJENJA, ULAZ.JK, T)
R.KL = DELAY (N,Q.JK, DEL)

R – izlaz koji kasni (jed/vr.)
Q – ulazna promena stanja materijalnog toka
DEL – kašnjenje između Q i R

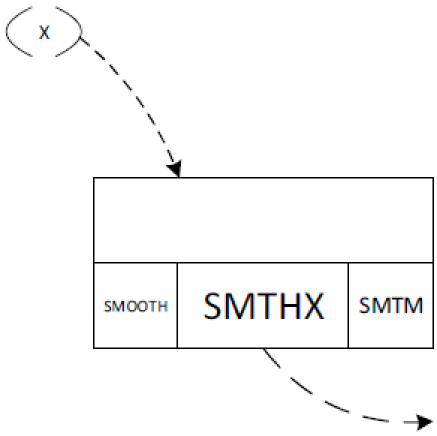
СТАЊЕ.К = СТАЊЕ.Ј + DT*(УЛАЗ.ЈК – ИЗЛАЗ.ЈК)
ИЗЛАЗ.KL = DELAY(РЕД КАШЊЕЊА, УЛАЗ.ЈК, Т)
R.KL = DELAY(N, Q.JK, DEL)

41. Modeliranje kašnjenja na informacionim tokovima u SDS jeziku



RVX.K = DLINF(n, X.K, TRX)

RVX – prepoznati iznos od X
X – stanje ili pomoćna jednačina čiji iznos kasni
TRX – vreme prepoznavanja iznosa X



SMTHX.K = SMOOTH(X.JK, SMTM)

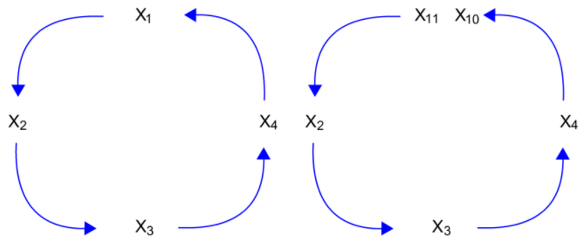
SMTHX – prosečan iznos od X
X – promena stanja koja se uprosečuje
SMTM – vreme uprosečavanja

42. Priraštaj: pojam, definisanje polariteta KPD preko priraštaja, primeri

Kada određujeumo polaritet KPD, izračunavamo ono što je u teoriji upravljanja poznato kao znak priraštaja otvorenog kola za KPD. Izraz ***priraštaj*** se odnosi na snagu signala koji se pronosi kroz KPD.

Na primer, priraštaj veličine dva znači da se promena varijable udvostručuje pri svakom prolasku impulsa kroz kolo, negativan priraštaj od 0.5 znači da se promena prenosi kroz kolo suprotstavljajući se sama sebi snagom koja je upola manja od nje same.

Termin *open loop* znači da se priraštaj proračunava za samo jedan ciklus, presecanjem (otvaranjem) kola u istoj tački. Razmotrimo proizvoljno KPD koje se sastoji od n promenljivih $X_1, X_2, \ldots, X_n$. Priraštaj otvorenog kola možemo izračunati u bilo kojoj tački. Neka X_1 predstavlja varijablu od koje ćemo krenuti. Određivanje polariteta veze je od izuzetnog značaja jer se na osnovu njega određuje da li je kolo kome pripadaju te veze pozitivno ili negativno.



Priraštaj otvorenog kola – (*open loop gain)*, se definiše kao parcijalni izvod X_{10} u odnosu na X_{11}, odnosno kao efekat povratnog dejstva male promene jedne varijable na samu sebe.

Polaritet kola je znak priraštaja otvorenog kola: $\boldsymbol{polaritet\ kola} = \boldsymbol{SGN}(\boldsymbol{\partial x_{10}/\partial x_{11}})$ $\boldsymbol{SGN}$**()**– ***signum funkcije***: ako je njegova vrednost +1 tada je reč o +KPD, a ako je -1 reč je –KPD. Za vrednost SGN() = 0, KPD ne postoji.

Priraštaj otvorenog kola se računa kroz lanac pravila koja se odnose na pojedinačne veze, $\boldsymbol{\partial x_j/\partial x_{j-1}}$

$$\boldsymbol{SGN\frac{\partial x_{10}}{\partial x_{11}} = SGN\left[\frac{\partial x_{10}}{\partial x_n}\cdot\frac{\partial x_n}{\partial x_{n-1}}\cdot\frac{\partial x_{n-1}}{\partial x_{n-2}}\cdot\ldots\cdot\frac{\partial x_2}{\partial x_{11}}\right]}$$

Polaritet kola se može takođe napisati kao:

$$\boldsymbol{SGN\frac{\partial x_{10}}{\partial x_{11}} = SGN\frac{\partial x_{10}}{\partial x_n}\cdot SGN\frac{\partial x_n}{\partial x_{n-1}}\cdot SGN\frac{\partial x_{n-1}}{\partial x_{n-2}}\cdot\ldots\cdot SGN\frac{\partial x_2}{\partial x_{11}}}$$

43. Dinamika sistema, stabilni i nestabilni sistemi, stacionarno i prelazno ponašanje

<u>Dinamika sistema</u> je metodologija istraživanja, modeliranja i simulacije složenih dinamičkih sistema. Nivo posmatranja sistema koji se modelira, a potom simulira, može se značajno razlikovati od slučaja do slučaja. Na primer, od modela nivoa nekog segmenta preduzeća, zatim modela celog preduzeća, preko modela nacionalne ekonomije, sve do modela svetske populacije ili ekologije. Ovakvi sistemi nisu kontinualnog karaktera, već menjaju svoja stanja kroz veći broj pojedinačnih diskretnih događaja.

<u>Grupisanje sistema prema obliku ponašanja:</u>

- ***<u>Stabilni sistemi</u>*** – oni koji teže da se posle nastanka poremećaja vrate u svoje prvobitno stanje.
- ***<u>Nestabilni sistemi</u>*** – oni koji posle nastalog poremećaja osciluju. Početni poremećaji se pretvaraju u stalni izvor unutrašnjih poremećaja.
- ***<u>Stacionarno ponašanje</u>*** – oblik ponašanja se ponavlja iz jednog u drugi vremenski interval (sezonska tražnja).
- ***<u>Prelazni oblik ponašanja</u>*** – onaj koji nastaje promenom karaktera sistema. On je neponovljiv (periodi rasta preduzeća, razvoj novog proizvoda itd.)

44. Obrasci ponašanja sistema

Obrasci ponašanja obično zahtevaju da se istraži kako se jedna ili više varijabli od interesa menjaju tokom vremena. *<u>Postoje obrasci ponašanja:</u>*

1. ***Eksponecijalni rast*** – kod eksponencijalnog rasta početna količina nečega raste i stopa rasta se povećava;
2. ***Goal seeking*** – kod asimptonskog ponašanja goal seeking veličina od interesa ima početnu vrednost iznad ili ispod cilja i vremenski se postepeno pomera prema cilju;
3. ***S-krive*** – kod rasta u obliku S-krive početni eksponencijalni rast je praćen asimptotskim ponašanjem što dovodi do toga da varijabla dostiže ciljno stanje;
4. ***Oscilacije*** – kod oscilacija varijabla od interesa se koleba (diže / spušta) oko nekog nivoa.

45. Ponašanje sistema prvog reda sa +KPD i -KPD

Pozitivno kolo povratnog dejstva nastaje kada se inicijalna promena u sistemu prenosi kroz sistem na taj način da to ima za posledicu nove promene, sve većeg intenziteta, u istom smeru *("efekat grudve snega")*. Karakteristično ponašanje za +KPD je eksponencijalni rast (pozitivan ili negativan). Eksponencijalan rast skladišta (nivoa, stanja) se karakteriše veličinom koja predstavlja vreme za koje se veličina skladišta udvostručuje – ***doubling_time***.

doubling_time= 0.7 ⁄ faktor_rasta

Negativno kolo povratnog dejstva nastaje kada inicijalna promena u sistemu proizvodi sve manje i manje promene u istom smeru sve dok se ne dostigne cilj ili kada inicijalna promena izaziva promene koje menjaju smer delovanja (*"izduvavanje balona", "dete koje uči da vozi bicikl"*). Zbog postojanja razlike između tekućeg stanja sistema i željenog stanja, negativno povratno dejstvo se karakteriše ponašanjem koje vodi ka postavljenom cilju – ***goal seeking***. Karakteristično ponašanje za –KPD za sistem I reda je asimptotsko ponašanje (rast ili pad). Asimptotsko ponašanje vrednosti skladišta se karakteriše veličinom koja predstavlja vreme za koje se veličina skladišta prepolovi – ***half_life*** ili ***halving_time***.

half_time=0.7*vremenska_const.= 0.7 ⁄ faktor_rasta

46. Ponašanje sistema višeg reda sa +KPD i -KPD

Sistem višeg reda sa +KPD - inicijalna promena se u sistemu prenosi kroz sistem na taj način da to ima za posledicu nove promene, sve većeg intenziteta, u istom smeru *("efekat grudve snega")*.

Sistem višeg reda sa –KPD - inicijalna promena u sistemu proizvodi sve manje i manje promene u istom smeru sve dok se ne dostigne cilj ili kada inicijalna promena izaziva promene koje menjaju smer delovanja (*"izduvavanje balona", "dete koje uči da vozi bicikl"*).

47. Slučajni događaji, verovatnoća, slučajne promenljive

Događaji koji se u datoj pojavi mogu ostvariti a isto tako i ne ostvariti, nazivamo ***slučajni događaji***, dok pojave koje izučavamo preko slučajnih događaja nazivamo ***eksperimenti***. Skup mogućih ishoda eksperimenta zovemo ***skup elementarnih događaja***, i bilo koji rezultat eksperimenta potpuno se opisuje jednim i samo jednim elementom iz skupa mogućih ishoda.

Skup elementarnih događaja može biti:

- Konačan;
- beskonačan i prebrojiv;
- beskonačan i neprebrojiv.

Karakteristike slučajnih događaja:

1. Slučajni događaj je podskup skupa elementarnih događaja.

2. Nemoguć događaj je podskup skupa elementarnih događaja koji nema nijedan element, i označava se sa ∅.
3. Neka je dat slučajan događaj A. Suprotan događaj je događaj $\bar{A}$ koji se ostvari onda i samo onda kada se A ne ostvari.
4. Suprotan događaj nemogućeg događaja je siguran događaj.
5. Data su dva slučajna događaja A i B. Ako se svaki put kada se ostvari događaj A ostvaruje i B , kažemo da A implicira B [A ⇒ B].
6. Za dva slučajna događaja A i B kažemo da su jednaki ako istovremeno A implicira B i B implicira A , tj. ako je [A ⇒ B] i [B ⇒ A].
7. Neka su dati slučajni brojevi A i B. Unija događaja A i B je događaj koji se ostvari onda i samo onda kada se ostvari bar jedan od događaja A i B, i označava se sa [A ∪ B].
8. Presek događaja A i B je slučajan događaj koji se ostvaruje onda i samo onda kad se ostvare oba događaja, tj. kada se istovremeno ostvari događaj A i događaj B, i označava se sa A·B ili AB ili [A ∩ B].
9. Za dva događaja A i B kažemo da se međusobno isključuju ako je njihov presek nemoguć događaj, tj. ako je [A ∩ B = ∅]
10. Ako se dva događaja A i B međusobno isključuju, tada je njihova unija zbir tih događaja i označavamo to sa [A ∪ B = A + B].
11. Za operacije unije i preseka koji su definisani na slučajnim događajima važe sledeća tri zakona:
    - Komutativnost
    - Asocijativnost
    - Distributivnost

Verovatnoća slučajnih događaja je svaka funkcija P definisana na slučajnim događajima koja preslikava slučajne događaje u realne brojeve i koja ima sledeće osobine:

1. ***Nenegativnost*** – Svakom slučajnom događaju A funkcija P pridružuje nenegativan broj $P(A)$.
2. ***Normiranost*** – Sigurnom događaju funkcija P pridružuje broj jedan.
3. ***Aditivnost*** – Ako se slučajni događaji : $A_1, A_2, \dots$ međusobno isključuju, onda funkcija P pridružuje njihovoj uniji broj koji je jednak zbiru brojeva koje funkcija P pridružuje događajima, tj. $\boldsymbol{P(A_1 + A_2 + A_3 + \cdots) = P(A_1) + P(A_2) + P(A_3) + \cdots}$

Potrebno je navesti još neke osobine verovatnoće:
1. Verovatnoća surotnog događaja jednaka je: $\boldsymbol{P(A) = 1 - P(A)}$
2. Verovatnoća nemogućeg događaja jednaka je nuli.
3. Verovatnoća bilo kog slučajnog događaja je broj koji se nalazi između $\boldsymbol{0 < P(A) < 1}$.

Praksa nameće problem nalaženja verovatnoće događaja A , ako se događaj B već desio, i takva verovatnoća $\boldsymbol{P(A/B)}$ naziva se ***uslovna verovatnoća***

$$\boldsymbol{P(A/B) = \frac{P(A \cdot B)}{P(B)}}$$

Slučajna promenljiva $\tilde{X}$ može se definisati kao promenljiva koja predstavlja ishod eksperimenta.

Ako sa $\boldsymbol{X = \{x_1, x_2, x_3, \dots\}}$ označimo rezultat eksperimenta, onda kažemo da promenljiva $\tilde{X}$ može uzeti jednu vrednost iz skupa mogućih rezultata eksperimenta i to na slučajan način. Potrebno je utvrditi koliko često, odnosno sa kojom verovatnoćom promenljiva $\tilde{X}$ uzima pojedine vrednosti iz datog skupa mogućih vrednosti.

U praksi se posmatraju dve kategorije slučajnih promenljivih i to:

- diskretne i
- kontinualne.

Promenljiva $\tilde{X}$ naziva se ***slučajna promenljiva diskretnog tipa*** ako vrednosti koje ona može uzeti obrazuju konačan ili prebrojiv niz realnih brojeva $\{\boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{x_3}, \ldots\}$ a uzimanje svake od ovih vrednosti je slučajan događaj sa određenom verovatnoćom.

Za promenljivu $\tilde{X}$ kažemo da je ***slučajna promenljiva kontinualnog tipa*** ako uzimanje bilo koje vrednosti iz intervala $(-\infty; +\infty)$ predstavlja slučajan događaj i ako postoji funkcija $f(x)$ takva da je verovatnoća da će se $\tilde{X}$ naći u proizvoljno maloj okolini tačke x u intervalu

$$\left[x - \frac{\Delta x}{2}; x + \frac{\Delta x}{2}\right]$$

48. Funkcija gustine raspodele i funkcija raspodele

Funkcija $f(x)$ naziva se ***funkcija gustine raspodele*** slučajne promenljive $\tilde{X}$. Funkcija gustine $f(x)$ mora da zadovolji sledeća dva uslova verovatnoće:

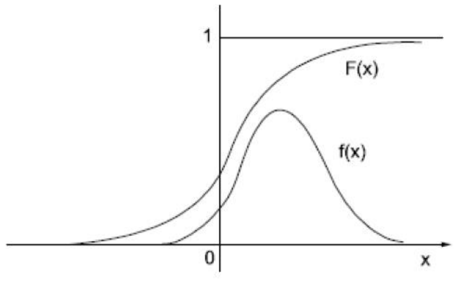
1. Verovatnoća da se slučajna promenljiva nađe u nekom intervalu $(x_1, x_2]$ je :

$$\boldsymbol{P(x_1 < X < x_2) = \int_{x_1}^{x_2} f(x)dx}$$

2. Da slučajna promenljiva $\tilde{X}$ uzme vrednost ne manju od x, pri čemu je x unapred određen broj iz intervala $(-\infty; +\infty)$. Npr. interesuje nas događaj da će u automatskoj telefonskoj centrali broj poziva biti manji od nekog broja koji predstavlja kapacitet centrale. Ta verovatnoća je:

$$\boldsymbol{P(X > x) = 1 - P(X > x).}$$

Funkcija raspodele slučajne promenljive $\tilde{X}$ je funkcija $F(x)$ кkoja za svako x iz $(-\infty; +\infty)$ predstavlja verovatnoću da slučajna promenljiva $\tilde{X}$ neće uzeti vrednost veću od z, tj. $\boldsymbol{F(x) = P(X < z)}$.



Слика П.1 Графички приказ функције густине расподеле и функције расподеле случајне променљиве

Funkcija raspodele okarakterisana je sledećim osobinama:

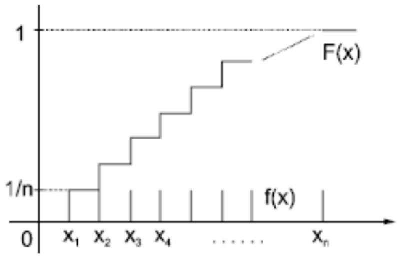
1. Funkcija raspodele ima vrednosti između nula i jedan, tj. $0 < F(z) < 1$.
2. $F(-\infty) = 0$.
3. $F(+\infty) = 1$.
4. Funkcija $F(x)$ je monotono neopadajuća funkcija.

5. Kod slučajne promenljive $\tilde{X}$ diskretnog (prekidnog) tipa funkcija raspodele $F(x)$ je stepenasta funkcija koja u tačkama $x_1, x_2, x_3, \ldots$ ima skokove jednake odgovarajućim verovatnoćama.
6. Ako je $\tilde{X}$ kontinualna (neprekidna) slučajna promenljiva funkcijom gustine raspodele $f(x)$, tada je za svako

$$\boldsymbol{x} \in (-\infty; +\infty) \Rightarrow \boldsymbol{F}(\boldsymbol{x}) = \int_{-\infty}^{\boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x})\boldsymbol{dx}$$

49. Diskretna i kontinualna uniformna raspodela

Diskretna uniformna raspodela slučajne promenljive $\tilde{X}$ može se grafički predstaviti na sledeći način:



Слика П.2 Функције расподеле и густине расподеле за униформну дискретну расподелу

Slučajna promenljiva $\tilde{X}$ uzima vrednosti iz skupa $\{x_1, x_2, \ldots, x_n\}$. ***Verovatnoća*** da slučajna promenljiva $\tilde{X}$ uzima vrednost x_i pri čemu je

$$\boldsymbol{P(X = x_i) = \frac{1}{n} \quad i = 1, 2, \ldots, n}$$

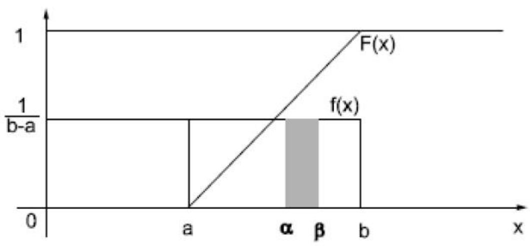
a ***matematičko očekivanje:***

$$\boldsymbol{\mu = \frac{1}{n}\sum_{i=1}^{n} x_i}$$

dok je ***varijansa*** jednaka:

$$\boldsymbol{\sigma^2 = \frac{1}{n}\sum_{i=1}^{n} (x_i - \mu)^2}$$

Kontinualna uniformna raspodela može se grafički prikazati na sledeći način:



Слика П.3 Функције расподеле и густине расподеле за униформну континуалну расподелу

gde je $x \in X = [a, b]$ interval u R, i

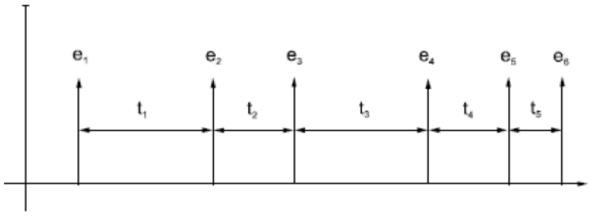
$$f(x) = \begin{cases} \dfrac{1}{a-b} & a \le x < b \\ 0 & \text{другачије} \end{cases}$$

Verovatnoća da će se slučajna promenljiva $\tilde{X}$ naći u intervalu $\{\alpha, \beta\}$ jednaka je

$$P(\alpha \le x < \beta) = \frac{\beta - \alpha}{b - a}$$

50. Eksponencijalna raspodela

Za predstavljanje realnih pojava kao što su dolazak kupaca u samoposlugu, pojava telefonskih poziva, pojava kvarova na mašini, i drugih sistema opsluživanja, iskustvom je utvrđeno da je najpogodniji oblik eksponencijalna raspodela. Događaje u okviru eksponencijalne raspodele možemo predstaviti sledećim grafikom:



t_i - времена између два суседна догађаја

Сл. П.4 Догађаји са експоненцијалном расподелом

Za ovu raspodelu važe polazne pretpostavke:

1. Vreme nastupanja događaja ne zavisi od prethodnog događaja
2. Verovatnoća da događaj nastupi u malom intervalu vremena proporcionalna je dužini tog intervala:

$$P\left(\tilde{t} < t + \frac{\Delta t}{\tilde{t}} \ge t\right) = \frac{P(t \le \tilde{t} < t + \Delta t)}{P(\tilde{t} \ge t)} = \frac{F(t+\Delta t) - F(t)}{1 - F(t)} = \lambda \cdot \Delta t$$

gde je $P(\tilde{t} < t) = F(t)$ tražena raspodela za $\tilde{t}$

Ako je $\Delta t \to 0 \Rightarrow F' = \lambda(1 - F(t)),\ t \ge 0$ pa je tražena raspodela:

$$F(t) = \begin{Bmatrix} 1 - e^{-\lambda t}, & t \ge 0 \\ 0 & t < 0 \end{Bmatrix}$$

dok je funkcija raspodele:

$$f(t) = \begin{Bmatrix} \lambda e^{-\lambda t}, & t \ge 0 \\ 0 & t < 0 \end{Bmatrix}$$

Matematičko očekivanje parametara ove raspodele je jednako:

$$E(\tilde{t}) = \int_0^\infty t\lambda e^{-\lambda t} dt = \frac{1}{\lambda}$$

Standardna devijacija je takođe jednaka : $\frac{1}{\lambda}$

51. Poisson-ova raspodela

Za slučajnu promenljivu $\tilde{X}$ kažemo da ima *Poisson*-ovu raspodelu ako može uzeti vrednost k iz niza nenegativnih celih brojeva $[0,1,2,3,\dots]$ sa verovatnoćom

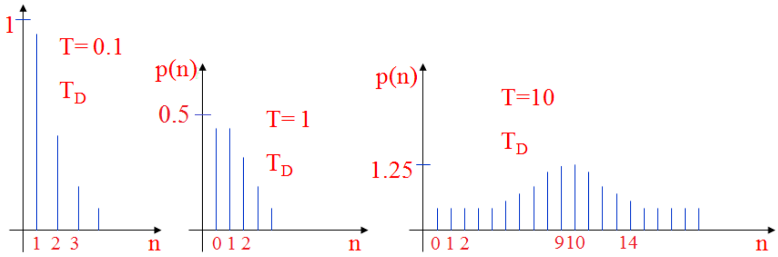
$$P(k) = P(X = k) = \frac{\lambda^k}{k!}e^{-\lambda}$$

Pri čemu je $\lambda > 0$, realan broj i predstavlja parametar raspodele.

Ako intervali između dva susedna događaja $(\tilde{t}\,)$ imaju eksponencijalnu raspodelu, tada je verovatnoća da se n događaja desi na intervalu T jednaka

$$P(\tilde{n} = n) = \frac{(\lambda T)^n e^{-\lambda T}}{n!}, \quad n = 0, 1, 2, \dots$$

gde je $\tilde{n}$ broj događaja u jedinici vremena.



Слика П.5 *Облици Poisson-ових расподела за различите односе*

52. Normalna raspodela

Normalna (Gauss-ova) raspodela ae koristi za predstavljanje stohastičkih pojava unošenjem šuma u determinističke promenljive. Definisana je sa 2 parametra:

- μ – srednja vrednost
- σ – standardna devijacija

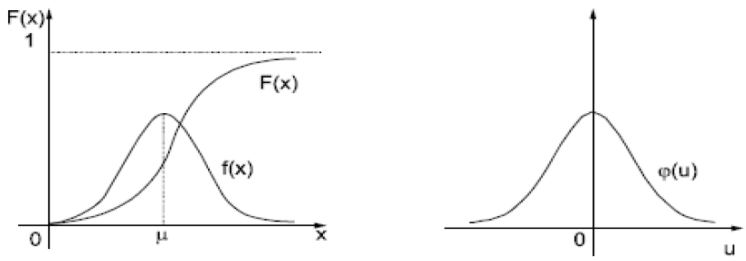
$$P(x < X) = \Phi\left(\frac{x-\mu}{\sigma}\right)$$

$$\Phi(z) = \int_{-\infty}^{\mu} (x)dx$$

Funkcija standardizovane normalne raspodele za koju važi da je $\mu = 0$ i $\sigma = 1$ ima oblik:

$$\Phi(z) = \frac{1}{\sqrt{2\pi}}e^{-\frac{\mu^2}{2}}, \quad z = \frac{x-\mu}{\sigma}$$

Grafički oblik funkcije raspodele i funkcije gustine raspodele za normalnu raspodelu:



Слика П.6 Нормална расподела

Na drugom grafikonu je standardizovana normalna raspodela sa vrednostima $\mu = 0$ и $\sigma = 1$.

53. Generisaje slučajnih brojeva

U toku specifikacije stohastičkih modela za promenljive koje opisuju promenu vremena, ulaza i stanja, definišu se statističke raspodele koje ih opisuju sa najmanjom greškom u odnosu na realni sistem. Pri simulaciji tako definisanih modela javlja se potreba za računarskim algoritmima koji će generisati slučajne promenljive za opisivanje promena vremena i stanja u modelu sa nekom zadatom raspodelom.

U statističkoj teoriji se dokazuje da je slučajnu promenljivu sa zadatom raspodelom moguće generisati na osnovu jedne ili više nezavisnih slučajnih promenljivih sa uniformnom raspodelom. Na taj način se problem svodi na generisanje slučajne promenljive sa uniformnom raspodelom.

Slučajnu promenljivu sa uniformnom raspodelom moguće je dobiti jednom od sledećih metoda:

- manuelne metode,
- tabele slučajnih brojeva,
- metode za rad sa analognim računarima i
- metode za rad sa digitalnim računarima.

Manuelne metode su malog opsega i zbog toga su neprimenljive u simulaciji. Eventualno se mogu koristiti za obučavanje kadrova. U ove metode spadaju: izvlačenje brojeva iz kutije, bacanje kockica itd.

Tabele slučajnih brojeva su serije slučajnih brojeva (SB) koji se proizvode na pogodan način, pomoću nekog fizičkog izvora ili nekom od metoda izvlačenja brojeva.

Najveća do sada publikovana tabela slučajnih brojeva, sadrži 1.000.000 cifara. Brojevi su dobijeni na sledeći način: bio je konstruisan specijalan rulet koji se pokretao i zaustavljao uz pomoć elektronike. Pokretni disk se naglo zaustavljao i birala se cifra koju je pokazivala strelica.

Ovako dobijene cifre se skupljaju u četvorocifrene ili petocifrene brojeve i unose u tabelu slučajnih brojeva. Iako ovakav postupak poseduje sva svojstva slučajnosti, ipak su posle dužeg vremena počele da se dobijaju cifre koje, prema proveri statističkim testom, nisu imale ravnomernu raspodelu.

Metode za generisanje SB na analognom računaru se baziraju na korišćenju slučajnih fizičkih procesa koji se odvijaju u analognom računaru i daju "prave slučajne brojeve". Nepodesne su za rad, jer u slučaju ponovljene simulacije sa istim redosledom slučajnih brojeva, fizički proces daje nove slučajne brojeve.

Metode za generisnje SB na digitalnom računaru baziraju se na sledećim postupcima:

1. Korišćenje fizičkih izvora + A/D konverzija,
2. Korišćenje tabela slučajnih brojeva na diskovima računara,
3. Korišćenje algoritama za dobijanje pseudoslučajnih brojeva.

Korišćenje fizičkih izvora (beli šum, radioktivni raspad) i konverzija fizičkih veličina u digitalne je skup i nepristupačan proces za svakodnevnu primenu.

Tabele slučajnih brojeva na masovnim memorijama računara su dobar ali spor način za dobijanje slučajnih brojeva pri simulacijama sasvim prosečnih sistema koji u toku simulacije traže nekoliko miliona slučajnih brojeva.

Algoritmi za generisanje pseudo-slučajnih brojeva su brz način za generisanje slučajnih brojeva. Međutim kvalitet takvih generatora zavisi od algoritma i računara na kome se implementira. Dužina sekvence brojeva bez ponavljanja je ograničena a uniformnost i rezolucija može da bude promenljiva. Zbog toga je potrebno pre upotrebe ovakvih generatora ispitati i utvrditi da li odgovaraju zahtevima modela.

Danas su pretežno u upotrebi ovakvi generatori.

54. Linearni kongruentni generator slučajnih brojeva

Kongruentni brojevi su takvi brojevi koji su međusobno deljivi bez ostatka. Koristeći osobine kongruentnih brojeva, možemo definisati sledeće vrste linearnih kongruentnih generatora pseudo-slučajnih brojeva sa uniformnom raspodelom:

1. Multiplikativni: $Z_i = (Z_{i-1} \cdot a) mod \cdot m$
2. Mešoviti: $Z_i = (Z_{i-1} \cdot a + c) mod \cdot m$
3. Aditivni: $Z_i = (Z_{i-1} \cdot a + Z_{i-k} \cdot b) mod \cdot m$

Gde su:

Z_i – pseudo-slučajni broj
Z_0 – početna vrednost ili seme generatora
a – multiplikator
b – konstanta
c – konstanta
m – modulus
mod – broj deljiv bez ostatka

Od navedenih generatora, najviše je u upotrebi linearni multiplikativni kongruentni generator pseudo-slučajnih brojeva.

Linearni multiplikativni kongruentni generator slučajnih brojeva generiše slučajne brojeve po sledećoj formuli:

$$\boldsymbol{Z_i = (a \cdot Z_{i-1} + b) mod \cdot m, \qquad m = 2^{b-1}}$$

Gde su:

Z_i – pseudo-slučajni broj
Z_0 – početna vrednost ili seme generatora
a – multiplikator
b – broj bitova celobrojne promenljive u računaru

Dokazuje se da je maksimalna moguća dužina niza generisanih brojeva, bez ponavljanja $N_{max} = m - 1$. Da bi se postiglo maksimalno moguće N_{max}, a i Z_0 treba da ispunjavaju sledeće uslove:

$$\boldsymbol{a = 8 \cdot j \pm 3, \quad j = 1, 2, 3, \ldots}$$

$$\boldsymbol{Z_0 = 1, 3, 5, 7, \ldots}$$

Gde je:

$$\boldsymbol{\frac{m}{4} \leq N_{max} \leq m - 1}$$

55. Testovi za proveru generatora slučajnih brojeva

Pseudo-slučajni brojevi dobijeni iz generatora slučajnih brojeva treba da zadovolje osobine koje se odnose na uniformnost raspodele unutar intervala (0,1) i nezavisnost pojavljivanja. Naime, iako je generisanje brojeva linearnim kongruentnim generatorima strogo deterministički algoritam, generisani brojevi moraju da zadovolje određene statističke testove i tada se smatraju slučajnim.

K-test:

Za ispitivanje da li neki skup generisanih slučajnih brojeva ima pretpostavljeni raspored. Neki skup od n elemenata (slučajnih brojeva) može se grupisati prema nekoj osobini u r grupa. Neka se u i-toj grupi nađe f_i elemenata tako da je $f_1 + f_2 + \cdots + f_r = n$.

Verovatnoća da će se element x naći u grupi f_i je p_i a broj elemenata koji prema teorijskoj raspodeli treba da se nađu u i-tom intervalu iznosi np_i.

Da bi se pretpostavka *(H_0)* da dati skup podleže određenoj raspodeli bila prihvaćena, treba da se nađe vrednost izraza $\mathbf{2(f_i - np_i)}$ i da se uporedi sa vrednošću κ koja se nalazi u tablicama za $r-1$ stepene slobode i za verovatnoću koja odgovara zadatom nivou značajnosti. $k < k_0$ skup podleže pretpostavljenoj raspodeli, tj. usvaja se hipoteza H_0, u suprotnom se hipoteza odbacuje.

Kolmogorov-Smirnov test

Omogućava poređenje da li ispitivani skup slučajnih brojeva podleže pretpostavljenoj raspodeli. Neka je generisano *n* slučajnih brojeva. Treba prvo obaviti njihovo sređivanje po veličini: $\boldsymbol{x_1} < \boldsymbol{x_2} < \cdots < \boldsymbol{x_n}$.

Kod uniformne raspodele, verovatnoća pojavljivanja svakog od njih iznosi $1/n$. Kumulativna funkcija raspodele može se konstruisati na taj način što će kod svakog x doći do porasta funkcije za $1/n$. Pošto je kumulativna funkcija uniformne raspodele $Fn(x)$ poznata, to se može izračunati izrazom

$$D_n = \max_{-\infty<z<\infty} |F(z) - F_e(z)|$$

Ukoliko je $D_n(x)$ manje od kritične vrednosti d_0 za prag značajnosti α, nulta hipoteza (hipoteza da se radi o uniformnoj raspodeli) biće prihvaćena.

Prednost nad *k-testom* – mogu se testirati mali skupovi brojeva. Međutim, potreba da se brojevi srede po redosledu zahteva dosta memorijskog prostora, jer je broj generisanih brojeva obično vrlo veliki. Drugi testovi (ređe se koriste): poker test, test serije istih cifara, intervalni test i testovi korelacije.

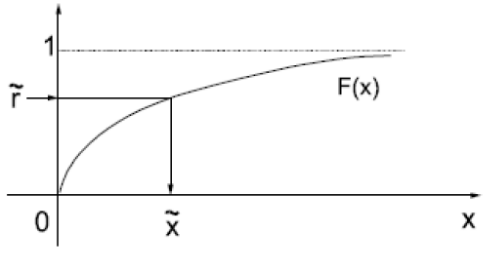
56. Metoda inverzne transformacije

Uvodi se smena $\tilde{X} = h(\tilde{r})$ takva da slučajnoj promenljivoj $\tilde{X}$ odgovara željena funkcija raspodele $F(X)$.

$$\boldsymbol{F(X) = P(\tilde{X} = X) = P(\tilde{r} = r) = P\left(\tilde{r} < h^{-1}(X)\right) = U\left(h^{-1}(X)\right) = h^{-1}(X)}$$

$$\boldsymbol{h(\tilde{r}) = F^{-1}(\tilde{r})}$$

Primena ove metode ograničena je na funkcije raspodele koje imaju inverznu funkciju i monotone su. Uglavnom se primenjuje za generisanje slučajnih promenljivih sa eksponencijalnom raspodelom.



Слика П.7 Графички приказ методе инверзне трансформације

57. Metoda odbacivanja

Uslov za primenu ove metode je ograničen opseg definisanosti funkcije gustine raspodele.

$$\boldsymbol{f(X) \approx 0 \;\; за \;\; X \notin [a, b].}$$

Ako se na površini $(b-a)*c$ generiše niz tačaka sa uniformnom raspodelom i ako se odbace sve tačke koje padaju iznad krive $f(X)$ preostale tačke imaće funkciju gustine raspodele $f(X)$.

Algoritam:

REPEAT

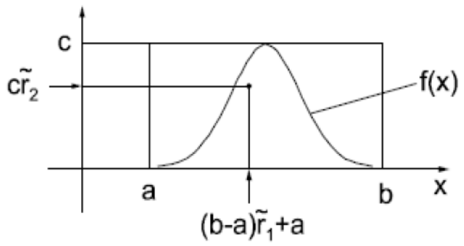
Generiši $r_1 i\, r_2$;

$X_1 := (b-a)*r_1 + a$

UNTIL $r_2 * c \leq f(X_1)$

$X \coloneqq X_1$

S obzirom na to da se prilikom svakog promašaja nanovo generišu slučajni brojevi sa uniformnom raspodelom, kao to se troši izvesno računarsko vreme, za primenu ove metode pogodnije su raspodele koje imaju veću površinu ograničenu apscisom i krivom funkcije raspodele.



Слика П.8 Графички приказ методе одбацивања

58. Metoda pravougaone aproksimacije

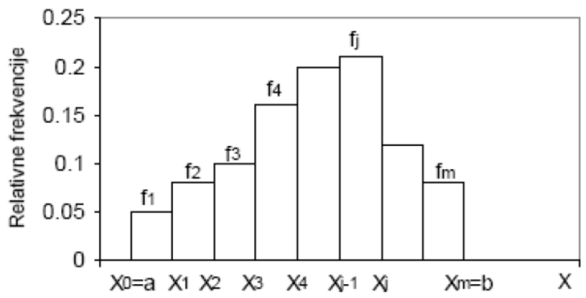
Koristi se za generisanje empirijskih raspodela, kada je funkcija raspodele monotona i zadata parovima tačaka.

Kod mnogih realnih problema, verovatnoća da će se desiti događaj izražava se u vidu empirijskih podataka. Ti podaci se mogu grupisati u proizvoljan broj frekvencijskih klasa (m) koje sačinjavaju histogram.

Svaka klasa je predstavljena pravougaonikom čija se donja i gornja granica mogu označiti sa $\boldsymbol{x_{j-1}, x_j,\;\; j = 1, 2, \ldots, m}$. Ukupan broj realizacija slučajne promenljive u okviru jedne klase se naziva ***apsolutna frekvencija*** i označava sa $n_j,\;\; j = 1,2,\ldots,m$.

Relativna frekvencija svake klase $\boldsymbol{f_j = 1/n_j}$, predstavlja verovatnoću da će realizacija slučajne promenljive $\tilde{X}$ uzeti vrednost iz klase j:

$$\boldsymbol{X_{j-1} \leq \tilde{X} \leq X_j}$$



Слика П.9 Хистограм расподеле релативних фреквенција

Zbir relativnih frekvencija je jednak jedinici:

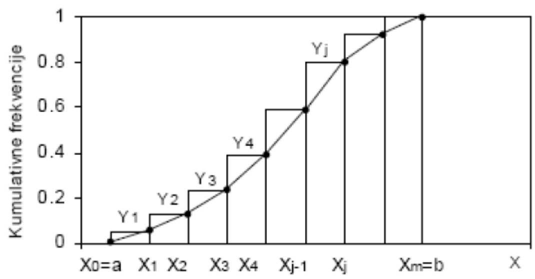
$$\sum_{j-1}^{m} f_j = 1$$

Na osnovu histograma se konstruiše funkcija raspodele, pri čemu je potrebno izračunati kumulativnu sumu prethodnih frekvencija Y_j:

$$Y_1 = f_1$$
$$Y_2 = f_1 + f_2$$
$$\vdots$$
$$Y_j = f_1 + f_2 + \cdots + f_j$$
$$\vdots$$
$$Y_m = f_1 + f_2 + \cdots + f_m = 1$$

Y_j predstavlja verovatnoću da slučajna vrednost $\tilde{X}$ ne prelazi vrednost X_j.

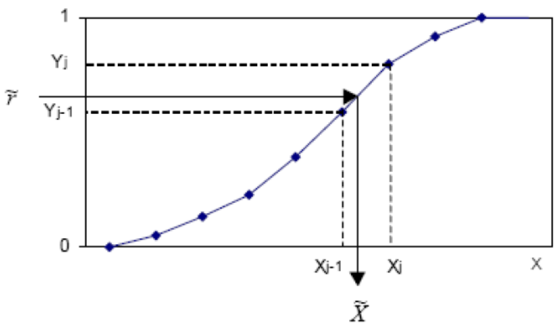
Da bi omogućili primenu metode inverzne transformacije, potrebno je aproksimativno odrediti funkciju raspodele, tako što se kroz kumulativni histogram provlači kontinualna kriva povlačenjem segmenata pravih linija:



Слика П.10 Кумулативна сума релативних фреквенција

Za dobijanje odgovarajuće vrednosti $\tilde{X}$ potrebno je generisati uniformno raspodeljen slučajni broj $\tilde{r}_i \in [0, 1)$ i koristiti **jednačinu za linearnu interpolaciju**:

$$\boldsymbol{\tilde{X}} = \boldsymbol{X_{j-1}} + \left(\frac{\tilde{r} - Y_{j-1}}{Y_j - Y_{j-1}}\right)(X_j - X_{j-1})$$



Слика П.11 Линеарна апроксимација функције расподеле

Jedini preostali problem jeste određivanje klase koja odgovara slučajnoj veličini $\tilde{r}$. To se može rešiti odgovarajućim pretraživanjem, startujući od krajnje leve klase (t.j. $j = 1$) i sukcesivnim poređenjem vrednosti $\tilde{r}$ sa vrednostima Y_j. Tražena klasa je prva klasa za koju je $\tilde{r} \leq Y_j$.

Pri implementaciji ove metode na računaru moraju biti poznate sledeće vrednosti: donja i gornja granica apcise funkcije raspodele ($a = X_0,\ b = X_m$), kao i vrednosti kumulativnih frekvencija Y_j . Kada su klase jednakih širina, njihove granice se računaju na sledeći način:

$$\boldsymbol{X_{j-1} = a + \left(\frac{b-a}{m}\right)(j-1)}$$

$$\boldsymbol{X_j = a + \left(\frac{b-a}{m}\right)j}$$

Ukoliko klase histograma nisu jednake širine, potrebno je pored pomenutih vrednosti učitati i granice svih klasa $X_j,\ \ j = 0,1,\ldots,m.$

<u>Gornje jednačine se mogu modifikovati u sledećim slučajevima:</u>

1. Ukoliko je funkcija raspodele zadata analitički, a potrebno je „pripremiti“ istu kao ulaz u neki od simulacionih programa, tada je moguće izvršiti podelu vrednosti funkcije raspodele na jednake intervale, za koje je potrebno izračunati vrednosti $X_i,\ \ i = 1,2,\ldots,n$..
2. Ako je poželjno izjednačiti vrednosti frekvencija klasa u histogramu, tada je potrebno formirati klase različitih širina. U tom slučaju su vrednosti funkcije raspodele, koje odgovaraju granicama klasa, na jednakim rastojanjima.
3. Kada se zahteva efikasniji algoritam za pretraživanje, pri čemu greška aproksimacije nije kritična, moguće je izvršiti podelu vrednosti funkcije raspodele (Y) na jednake intervale, za koje je potrebno izračunati vrednosti X.

U navedenim slučajevima, moguće je koristiti sledeći algoritam, koji je efikasniji u računskom smislu:

Generiši $\tilde{r}$;

$$i := INT\left(\frac{\tilde{r}}{\Delta Y}\right)$$

$$\tilde{X} := X_i + A_i * (\tilde{r} - i * \Delta Y)$$

gde je :

$$A_i := \frac{X_{i+1} - X_i}{\Delta Y}$$

$$\Delta Y := \frac{1}{m}$$

59. Metoda sumiranja

Koristi se za generisanje normalne (Gauss-ove) raspodele. Zasniva se na primeni centralne granične teoreme.

Centralna granična teorema

Neka je $X_1, X_2, \dots, X_n$ niz nezavisnih slučajnih promenljivih sa jednakim verovatnoćama, svaka sa srednjom vrednošću $\boldsymbol{\mu_x}$ i konačnom varijansom $\boldsymbol{\sigma_x^2}$. Njihova srednja vrednost data je sledećim izrazom:

$$\boldsymbol{\bar{X} = \frac{1}{n}\sum_{i=1}^{n} \tilde{X}_i}$$

Tada promenljiva

$$\boldsymbol{Y = \frac{(\overline{X - \mu_x})}{\sigma_x/\sqrt{n}}}$$

konvergira ka standardnoj normalnoj raspodeli sa srednjom vrednošću μ i standardnom devijacijom $\sigma/\sqrt{n}$, tj. za dovoljno veliko n razlika između promenljive Y i standardizovane normalne promenljive može se zanemariti iz praktičnih razloga.

Da bi generisali uzorak iz standardne normalne raspodele, možemo uzeti n nezavisnih uniformno raspodeljenih slučajnih brojeva $\tilde{r}_i \in [0, 1)$ sa srednjom vrednošću $E(\tilde{r}_i) = 1/2$ i varijansom $(\tilde{r}_i - \mu)^2 = 1/12$.

Slučajna promenljiva:

$$\boldsymbol{Z = \frac{\frac{1}{n}\sum_{i=1}^{n} \tilde{r}_i - \frac{1}{2}}{\sqrt{\frac{1}{12}}/\sqrt{n}} = \frac{\sum_{i=1}^{n} \tilde{r}_i - \frac{1}{2}}{\sqrt{\frac{n}{12}}}}$$

ima normalnu raspodelu sa srednjom vrednošću nula i varijansom jedan

$$\boldsymbol{\left(E(Z) = 0,\ \ E(Z-\mu)^2 = 1\right)}$$

kada $n \to \infty$. Zadovoljavajući rezultati aproksimacije dobijaju se za $n = 12$ u tom slučaju gornja jednačina postaje:

$$\boldsymbol{Z = \sum_{i=1}^{12} \tilde{r}_i - \sigma}$$

Ukoliko je potrebno generisati uzorke iz nestandardizovane normalne raspodele, potrebno je koristiti sledeći izraz: $\boldsymbol{X = \mu + \sigma + Z}$

60. Box-Muller-ov metod

Box-Muller-ov metod je egzaktan metod koji koristi dve nezavisne pseudo-slučajne promenljive r_1 и r_2 koje se koriste za generisanje standardizovanih normalno raspodeljenih slučajnih promenljivih korišćenjem oba ili jednog od sledećih izraza:

$$\boldsymbol{Z_1 = \cos(2\pi r_1)\sqrt{-\ln(r_2)}}$$

$$\boldsymbol{Z_{21} = \sin(2\pi r_1)\sqrt{-\ln(r_2)}}$$

Ukoliko je potrebno generisati uzorke iz nestandardizovane normalne raspodele, potrebno je koristiti sledeći izraz:

$$\boldsymbol{X = \mu + \sigma + Z}$$

Kod metode sumiranja potrebno je za svako Z imati na raspolaganju 12 različitih vrednosti $\tilde{r}_i$, dok su kod Box-Muller-ove metode potrebne samo dve takve vrednosti. Međutim, ovaj metod zahteva izračunavanje kvadratnog korena, logaritma, kosinusne ili sinusne funkcije, što zahteva više računarskog vremena nego što je potrebno za određivanje jednostavne sume. Takođe, tačnost rezultata zavisi i od ugrađenih funkcija (bibliotečkih podprograma) za izračunavanje kvadratnog korena, logaritma, kao i sinusne i kosinusne funkcije.