



Dragan S. Marković

UVOD U TEORIJU SISTEMA

BEOGRAD, 2012.

UNIVERZITET SINGIDUNUM

Doc. dr Dragan S. Marković

UVOD U TEORIJU SISTEMA

Prvo izdanje

Beograd, 2012.

UVOD U TEORIJU SISTEMA

Autor:

Doc. dr Dagan S. Marković

Recenzenti:

Prof. dr Radica Prokić-Cvetković

Prof. dr Dragan Cvetković

Izdavač:

UNIVERZITET SINGIDUNUM

Beograd, Danijelova 32

www.singidunum.ac.rs

Za izdavača:

Prof. dr Milovan Stanišić

Priprema za štampu:

Novak Njeguš

Dizajn korica:

Aleksandar Mihajlović

Godina izdanja:

2012.

Tiraž:

300 primeraka

Štampa:

Mladost Grup

Loznica

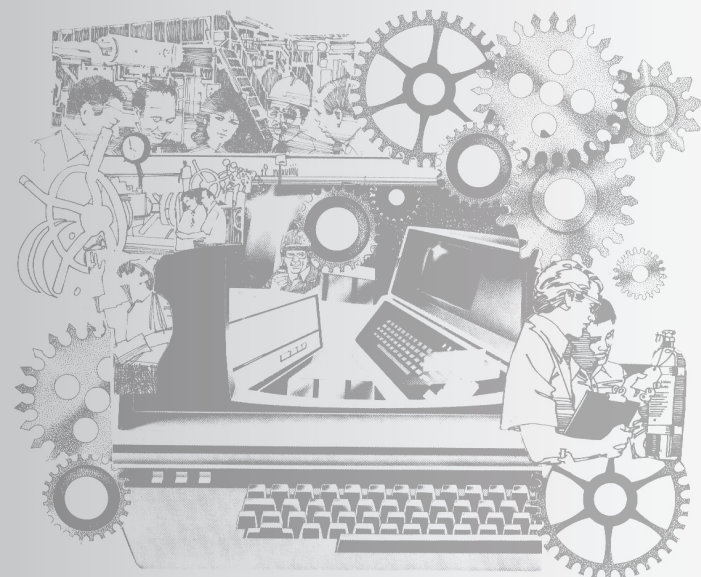
ISBN 978-87-7912-399-2

Copyright:

© 2012 Univerzitet Singidunum

Izdavač zadržava sva prava.

Reprodukcija pojedinih delova ili celine ove publikacije nije dozvoljeno.



PREDGOVOR

Posmatrajući savremenu teoriju i praksu primećujemo da se pojam „sistem“ veoma često koristi, tako da teoretičari i praktičari iz brojnih oblasti (psihologije, ekonomije, informatike, fizike, biologije, kibernetike itd.) izučavaju raznorodne sisteme i metode opisivanja njihove strukture i upravljanja.

Zbog toga su se, već od početka, teorija sistema koja tretira strukturu, i kibernetika koja tretira teoriju upravljanja, razvile u celovitu naučnu disciplinu prilagođenu i predodređenu da se uključuju i utiču na sve kurseve iz ekonomije, marketinga, menadžmenta, organizacije, politike, sociologije i brojnih drugih oblasti.

Nauka o sistemima kroz opštu zamisao strukture sistema i upravljanja sistemima pruža delotvornu metodologiju za proračun različitih sistema, bez obzira na njihovu posebnu fizičku prirodu i način ponašanja.

Proučavanjem najopštijih zamisli teorije sistema detaljno se razmatra problem izbora stanja i funkcionisanja sistema kroz prevođenje različitih modela u oblike pogodne za proučavanje.

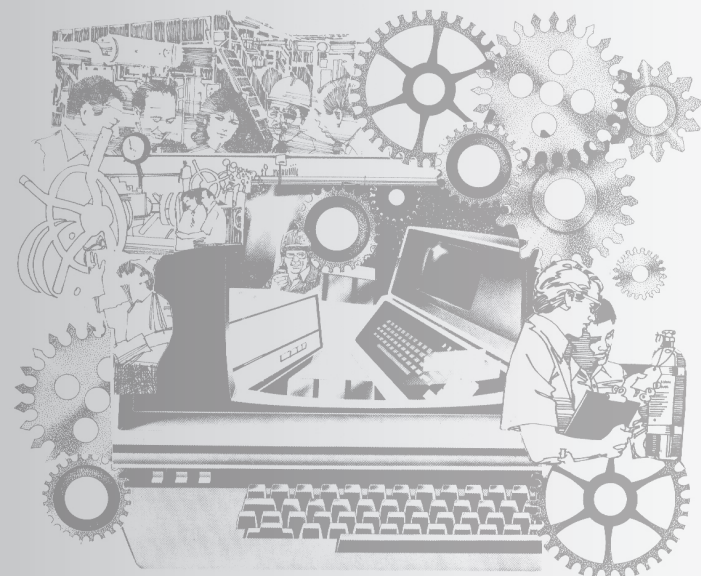
Sadržaj knjige je kroz postupno uvođenje osnovnih pojmova i principa prilagođen studentima inženjerskog menadžmenta koji se osposobljavaju za primenu inženjerskih principa u planiranju, operativnom menadžmentu u industriji, trgovini, saobraćaju, kao i u proizvodnim operacijama.

Svestan sam da zbog opsežnosti oblasti neke teme nisu obrađene ili nisu obrađene u dovoljnoj meri. Takođe, moguće je da su se potkrale neke greške ili nepreciznosti, stoga biću zahvalan svakome ko ukaže na moguće propuste ili da konstruktivne primedbe i sugestije kako bi udžbenik u sledećem izdanju bio kvalitetniji kao nastavno sredstvo.

Beograd, oktobar 2011.

Autor





SADRŽAJ

Predgovor	III
1. Uvod	1
2. Sistemski pristup	9
3. Model	21
4. Dinamički sistem	29
5. Uvod u teoriju informacija	39
6. Upravljanje	49
7. Regulatori	57
8. Optimalno upravljanje	65
9. Automati	75
10. Upravljanje tehnološkim procesima	81
11. Adaptacija	105
12. Uvod u teoriju igara	111
13. Obučavanje	131
14. Veliki sistemi	139
15. Upravljanje operacijama	153

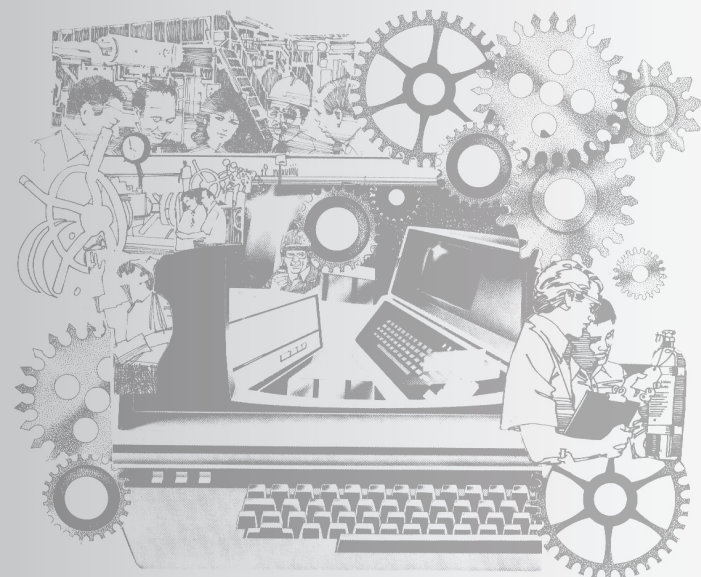
Praktikum

1. Diferencijalne jednačine	157
2. Diferencne jednačine	163
3. Signali i sistemi	169
4. Laplasova transformacija	177
5. Kratki potsetnik iz Matlab-a	184
6. Matematičko modelovanje objekta i procesa	194
7. Uvod u simulink	204
8. Uvod u simulacioni program VENSIMPLE	253

Literatura

279





1. UVOD

Teorija sistema je nauka tj. naučna disciplina, koja se bavi proučavanjem složenih pojava koje nazivamo sistemima. Nastala iz potrebe pronalaženja takvih naučnih i praktičnih metoda pomoću kojih bi se na naučni način mogli rešavati i oni problemi kod kojih tradicionalne i uobičajene naučne metode razvijene u drugim naučnim područjima ne daju zadovoljavajuće rezultate. Teorija sistema i njen pristup u velikom broju slučajeva su vrlo uspešno sredstvo naučnog istraživanja i stručnog rešavanja problema, ali nisu univerzalno sredstvo za rešavanje svih mogućih problema.

Već na početku razvoja nauke pojedini naučnici su uočili da postoje tzv. zakonitosti celina, tj. zakonitosti koje se ne mogu neposredno identifikovati kao zakonitosti delova koji čine celinu.

Npr. delovi nekog složenog proizvoda, teško da mogu dati predstavu o tome:

- ♦ kakav je to proizvod;
- ♦ čemu on služi;
- ♦ kako funkcioniše;
- ♦ kako se upotrebljava;

odnosno, količina informacija koju nam može dati posmatranje delova nije dovoljna da bismo shvatili i spoznali svojstva celine.

Primer slova i reči: - pomoću slova A, E, K, P i T može se napisati PETAK ili PAKET;

Primer reči i teksta: - popis reči i njihove učestalosti u nekom tekstu ne omogućava identifikaciju teksta kojem te reči pripadaju.

Među prve naučnike za koje se smatra da su uočili činjenicu da se celina bitno razlikuje od sume delova ubraja se čuveni grčki filozof Aristotel: „Celina je više od sume delova”.

Slična mišljenja su tokom vekova imali ne samo grčki nego i mnogi drugi naučnici, međutim službena nauka se na tu tvrdnju nije mnogo osvrtdala.

Američki biolog *Ludvig Bertalanfi* (*Ludwig von Bertalanffy*) - je 1937. na naučnom seminaru iz filozofije na univerzitetu u Čikagu, prvi puta javno, na jednom naučnom skupu, izneo svoje ideje o potrebi i mogućnostima stvaranja jedne opšte teorije koja bi upućivala na proučavanje sistemskih fenomena, tj. svih tipičnih sistemskih svojstava, i važila za sve sisteme bilo koje prirode oni bili.

Bertalanffy je tada pod pojmom "sistem" podrazumevao ono što se podrazumeva pod pojmom funkcionalna celina, a sama ideja nije imala posebnog odjeka.

Bertalanfi 1956. godine u SAD osniva naučno društvo *Society for General Systems Research* (SGSR) (Društvo za opštu teoriju sistema) sa suosnivačima:

- ♦ *Keneth Boulding* (ekonomista)
- ♦ *Anatole Rappaport* (biomatematičar)
- ♦ *Richard Gerard* (psiholog)

Smatra se da je osnivanjem tog Društva zapravo osnovana Teorija sistema kao naučna disciplina.

Po Bertalanfiju zamišljena i osnovana Teorija sistema najčešće se naziva Opšta teorija sistema (*General System Theory*) ili Teorija otvorenih sistema (za razliku od Kibernetike koju nazivaju Teorijom zatvorenih sistema) i njen je značaj sledeći:

- ♦ ukazala je i dokazala objektivno postojanje sistemskih zakonitosti;
- ♦ ukazala je na nužnost proučavanja tih zakonitosti;
- ♦ dala je osnovni pojmovni koncept i pristup rešavanju sistemskih problema i razumevanju sistemskih pojava.

Uvod u opštu teoriju sistema

U naučnoj teoriji XX veka sve češće se primenjuje pojam sistemskog (sistematskog) istraživanja kao posebnog naučnog, metodološkog pristupa. Sistemski pristup u osnovi akcenat daje analizi sveobuhvatnih integrativnih svojstava objekata. Krajem 40-tih i 50-tih godina XX veka ideja sistemskih pristupa i osnove za formiranje opšte teorije sistema vezuje se za ime *Ludviga Bertalanfija*. Prve njegove publikacije iz opšte teorije sistema javljaju se sa pojavom radova *Noberta Vinera* iz oblasti kibernetike. Tumačenje povratne sprege kao principa veze i regulacije koji je zajednički za mašine, čoveka i žive organizme, tj. za tehničke, biološke i društvene sisteme, poslužio je kao osnova za naučno proučavanje u oblasti upravljanja. Saznanjem da je sistem regulacije u principu istovetan kod svih sistema dovodi do toga da se formira nova naučna oblast koju je Viner nazvao *kibernetika* ili regulacija i komuniciranje u živom organizmu i mašini.

Sistemski pristup predstavlja opšti metodološki pravac u cilju analize i razrade specifičnih metodoloških pristupa teorijskog realizovanja i saznanja o celinama objekta kao sistema. Opšta teorija sistema proistekla je iz potrebe razvoja posebnih naučnih disciplina kao što su: kibernetika, teorija informacija, teorija upravljanja, izgrađujući nov metodološki pristup u biologiji, ekonomiji i tehnici. U tim naukama prvi put se javljaju i definicije sistema, zato su neujednačene definicije pojma sistema. Postoji veliki broj definicija sistema.

Reč sistem je najčešće usko povezana sa pojmom reda ili poretka, tj. nečega u čemu vlada nekakav red i neka sistematičnost.

Prva i sa filozofskog gledišta najtačnija definicija: „Sistem je nešto suprotno od haosa tj. nereda, dakle nešto u čemu vlada nekakav red i poredak”.

Bertalanffy: „Sistemi su skupine elemenata u međusobnom i uzajamnom delovanju na koje se sistemski zakoni mogu primeniti”.

Praktične definicije sistema:

- ♦ „Sistem je relativno odvojeni skup međusobno povezanih pojava koji se ponaša prema nekim svojim zakonima”;
- ♦ „Sistem je skup pojava sa određenim ustrojstvom ili organizacijom koji ima neku svrhu ili razlog postojanja”;
- ♦ „Sistemom možemo smatrati sve ono što sa nekog stanovišta ima tri osnovne k-ke: elemente, strukturu, funkciju”.

Definicije možemo podeliti u tri grupe:

I grupa definicija – tu spadaju definicije koje određuju sistem kao klasu matematičkih modela pomoću kojih se grade ideje dinamičkih pojava.

II grupa definicija – je najbrojnija i nju karakterišu pojmovi kojima se definiše sistem elemenata, veza ili celina.

III grupu definicija – formiraju definicije koje se obrazuju pomoću stanja ulaz-izlaz, obrade informacija itd.

Sistem je skup elemenata ili procesa povezanih odnosima (vezama), sa zajedničkom svrhom postojanja (zajedničkim ciljem).

Elementi mogu biti materijalni, ideje, funkcije, živa bića, kombinacije.

Da bi nešto bilo sistem moraju biti ispunjeni uslovi:

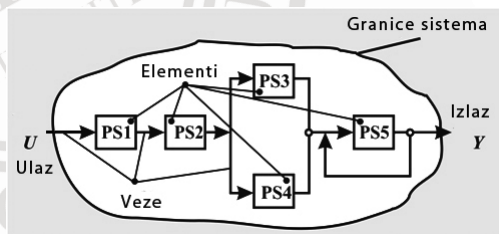
1. postojanje elemenata,
2. postojanje veza – relacija,
3. postojanje zajedničke svrhe (cilja) postojanja,
4. funkcioniše po određenim pravilima,
5. relativno izolovana celina.

Svaki sistem na osnovu ovih 5 elemenata se strukturiira da bi:

- ♦ imao određenu strukturu,
- ♦ vršio određenu funkciju,
- ♦ dao ili prerađivao informacije.

Formalno, sistem se može posmatrati kao skup funkcionalnih komponenti koje zajedničkim delovanjem postižu cilj ili izvršavaju zadatak u okviru zadatih granica. Svaki sistem sučeljava se sa okruženjem koje ga okružuje. S te tačke gledišta, gotovo sve može se kvalifikovati za naziv „sistem” – i stoga je upotreba ovog termina tako česta.

Posmatrani sistem može biti raščlanjen (dekomponovan) na konačan broj delova, koje nazivamo podsistemima složenog sistema. Svaki od ovih podsistema sa svoje strane



Sl. 1.1 Sistem

može se raščlanjivati na više manjih podsistema, dok se nakon konačnog broja koraka ne dođe do takvih delova koje zovemo elementima složenog sistema. Da li ćemo neku pojavu posmatrati kao sistem ili kao elemenat nekog sistema zavisi od: svrhe posmatranja, načina posmatranja i pristupa problem. Da bi skup elemenata činio sistem između njih mora postojati neka veza ili odnos.

Dekompozicija zavisi od zahtevane detaljizacije i od našeg stepena razumevanja posmatranog sistema.

Elementi su delovi ili pojave koje ne raščlanjujemo (elementi mogu biti i podsistemi).

Elementi sistema ne funkcionišu izolovano jedan od drugog, već su u uzajamnoj povezanosti, pri kojoj se svojstva elemenata određuju i ponašanjem i funkcijom ostalih elemenata sistema.

Svojstva složenog sistema se ne određuju samo pojedinačnim svojstvima sastavnih delova, nego i karakterom njihovih međusobnih veza i uticaja. Svojstva sistema se mogu menjati u zavisnosti od rezultata rada sistema, kao i uslova okoline u kojoj sistem egzistira.

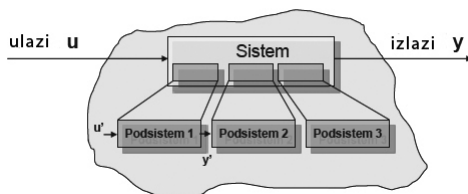
Veze su sredstva koja drže sistem zajedno. Veze su sve ono što povezuje elemente i svojstva, tako da sve to funkcioniše kao jedno celo (kao jedan proces – sistem). Veze postoje kako između elemenata sistema, tako i između podsistema posmatranog sistema, a isto tako između elemenata i delova drugih sistema iz okoline.

Struktura je skup svih veza, odnosno stabilni poredak i relativno stabilan raspored uloga. Strukturu čine elementi, ali elementi između kojih postoji određena zakonitost svojstvena datom sistemu kao integralnoj celini, nasuprot svojstvima samih elemenata.

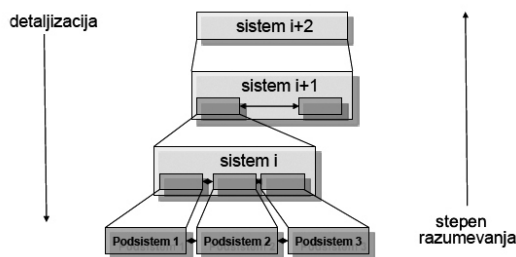
Kod realnih sistema pojam **funkcija** obuhvata: tok operacija, ponašanje i način delovanja.

Na primer, električno zvonce ima određenu strukturu, koja se sastoji od određenih delova: provodnika, prekidača, magneta, čekića, zvona. Njegova funkcija obuhvata niz mehaničkih i električnih operacija, koje čine tok delovanja.

Važna zajednička osobina svih sistema je da se svojstva sistema ne mogu identifikovati kao običan zbir svojstava sastavnih delova sistema. Da bi se mogla otkriti svojstva sistema treba pažnju posvetiti strukturi sistema, tj. odnosima i vezama unutar sistema i odnosima i vezama sistema sa njegovom okolinom.



Sl. 1.2 Dekompozicija sistema



Sl. 1.3 Dekompozicija sistema zavisi od detaljizacije i stepena razumevanja sistema

Podela sistema

Priroda elemenata sistema i karakter njihovih veza može biti veoma različita tj. i podela i klasifikacija sistema je različita. Tako, npr. prema prirodi elemenata sisteme delimo na:

- ♦ realne
- ♦ apstrakne.

Realni sistemi:

- ♦ *tehnički sistemi* (zasnivaju se na funkcionisanju prirodnih zakonitosti i njih je stvorio čovek, pri čemu je ugradio takav redosled elemenata čija je realizacija ostvarivanje funkcije koja se zahteva od tog sistema) ;
- ♦ *organizacioni sistemi* (i ove sisteme je stvorio čovek na taj način što je organizovao i prirodne i tehničke sisteme) ;
- ♦ *privredna organizacija*;
- ♦ *tržište*, itd.

Apstraktni sistemi predstavljaju modele realnih sistema i njihov karakter je određen prirodom realnih sistema.

Prema poreklu elemente sistema možemo podeliti na:

- ♦ prirodne (koje je stvorila priroda) i
- ♦ veštačke (koje je stvorio čovek).

Tehnički sistemi se zasnivaju na funkcionisanju prirodnih zakonitosti i njih je stvorio čovek radi zadovoljenja nekog oblika potreba pri čemu je ugradio takav raspored i povezanost elemenata, čiji je glavni cilj ostvarivanje funkcije koja se zahteva od tog sistema (npr. sistem za automatsko zavarivanje).

Organizacione sisteme je stvorio čovek na taj način što je organizovao prirodne i tehničke sisteme (sistem preduzeća). Prema stepenu složenosti u odnosu na broj elemenata i njihovih međusobnih veza sistemi se mogu podeliti na složene i proste. Međutim, ova podela je relativna jer ako podelu sistema vršimo na proste i složene, onda kod prirodnih i bioloških sistema jednoćelijski organizmi predstavljaju proste sisteme, a čovek predstavlja složeni sistem.

Kod tehničkih sistema čekić bi po broju elemenata predstavljao prost sistem, jer se sastoji iz dve celine, a mašina bi predstavljala složen sistem jer se sastoji od većeg broja delova.

Sistem čovek – mašina bi po broju elemenata predstavljao prost organizacioni sistem, jer se sastoji iz dva elementa. Ako se posmatra struktura sistema kao kriterijum podele sistema na proste i složene, onda i prosti sistemi mogu da imaju veoma složenu strukturu veza. Upravo iz ovih razloga je podela sistema na proste i složene relativna.

KARAKTERISTIKE SISTEMA

Na osnovu analize oblika veza elemenata u sistemu mogu se razmotriti oblici ponašanja sistema. Sistem se razmatra kao celina, koja poseduje određene osobine koje se razlikuju od osobina elemenata i ima sopstvene zakone ponašanja. Ponašanje sistema predstavlja promenu stanja sistema u toku vremena. Stanje sistema predstavlja skup podataka koji

daju informaciju o prošlosti i sadašnjosti sistema na čijoj osnovi možemo odrediti ili predvideti ponašanje sistema u budućnosti. U sistem **ulazi** materija, energija i informacija u nekom stanju koje se može smatrati kao **polazno stanje sistema**. U sistemu se kroz konačan niz promena stanja formiraju **novi oblici materije, energije i informacije** sa nekim novim stanjem koji se kao **izlazi** iz sistema posmatraju kao **konačno stanje**. Niz promena u sistemu od nekog početnog do konačnog stanja predstavlja **proces**. Pod procesom se podrazumeva proizvoljna kvantitativna i kvalitativna promena tokom vremena, tj. vremenska promena u osobinama i količinama.



Sl. 1.4 Karakteristike sistema

Promene možemo da posmatramo kao promene pritiska i temperature u nekom reaktoru, promene temperature i pritiska u radnom prostoru, itd. Procesi mogu biti: mehanički, hemijski, toplotni itd. Procese možemo podeliti na različite načine, u zavisnosti od toga šta uzimamo za osnovu podele.

Podela procesa (šematski prikaz)

Podela procesa		
Prema:	Na:	
Međusobnoj povezanosti promenljivih	Linearne	Nelinearne
Vremenskoj zavisnosti parametara	Promenjive	Nepromenjive
Načinu odvijanja procesa	Neprekidne	Diskontinualne
Stepenu određenosti veza između promenljivih	Determinističke	Stohastičke
Prema promeni promenljivih tokom vremena	Statičke	Dinamičke
Zavisnosti prostornih koordinata	Koncentrisane	Raspodeljene

Poseban značaj u praćenju stanja i ponašanja sistema imaju informacioni sistemi i to procesni informacioni sistemi. Procesni informacioni sistemi služe za vođenje procesa i zamenjuju čoveka ili mu pomažu gde je to neophodno zbog njegovih određenih mogućnosti sa obzirom na brzinu odziva, količinu i tačnost prijema i obrade informacija. S obzirom na složenu strukturu informacionog sistema razlikujemo spoljne i unutrašnje funkcije informacionog sistema.

Spoljne funkcije određuju veze informacionog sistema i procesa tj. čoveka kome taj sistem služi, a unutrašnje funkcije su vezane za prijem i obradu informacija o procesu sistema, nadzoru, upravljanju i regulaciji procesa itd.

Prema ponašanju sistemi su razvrstani na determinističke i stohastičke.

Kod determinističkih sistema stanje sistema je u bilo kom trenutku vremena jednoznačno određeno stanjem tog sistema u predhodnom trenutku vremena tj. možemo pouzdano predvideti kako će se ponašati pod uslovom da su poznata ulazna stanja.

Kod stohastičkih sistema izlazna stanja sistema ili ponašanja sistema nije jednoznačno određeno ulaznim stanjem ni u datom momentu niti u predhodnom momentu vremena, već se pojavljuje slučajno, znači da je transformacija ulaznih stanja u izlazna višeznačna u ovom slučaju možemo izlazna stanja utvrditi samo stohastičkom zavisnošću od ulaznih stanja tj. da možemo utvrditi samo neku verovatnoću s kojom pri datom ulaznom dejstvu mogu nastati pojedina izlazna stanja.

Analiza sistema

Da bi smo vršili upoređenje ponašanja jednog sa ponašanjem drugog sistema ili stanja datog sistema u različitim vremenskim periodima potrebno nam je **stanje sistema**. Stanje jednog sistema je odraz pojedinačnih elemenata u sistemu u određenim vremenskim periodima. Za analizu u sistemu, prikupljanje i korišćenje potrebnih informacija može se ostvariti na dva načina:

- ♦ kada je nepoznata unutrašnja struktura elemenata (iz određenih razloga nije moguće rastaviti sistem na elemente),
- ♦ kada je broj elemenata sistema i njegovih veza veliki da je nemoguće uzeti sve u obzir. Za rešavanje pitanja primenjuje se eksperimentalna metoda i modeliranje.

Eksperimentalni pristup u istraživanju složenih sistema je otežan jer složenost po pravilu onemogućuje sprovođenje eksperimenta a eksperimentat sa elementima ne omogućuje dobijanje predstave o opštem stanju sistema te se najčešće primenjuje metod modeliranja. Metod predstavlja osnovu za određivanje metodologije izračunavanja ponašanja sistema. Konstruisanje modela podrazumeva formiranje uslovnog odraza slike realnog sistema i izučavanje njegovog svojstva u cilju dobijanja informacija o realnom sistemu. Taj odraz predstavlja model.

Modeliranje ponašanja sistema zasniva se na činjenici da se pod određenim uslovima može opaziti jednako ponašanje kod sistema bitno različitih po obliku, broju elemenata i fizičkoj prirodi procesa u njima.

Podela modeliranja:

- ♦ materijalno-predmetno:
 - fizičko
 - analogno
- ♦ misaono-idejno:
 - simboličko: matematičko, grafičko i logičko
 - intuitivno: metod scenarija, operacione igre i eksperimentalno.

Materijalni modeli predstavljaju ponašanja sistema pomoću materijalno – fizičko – tehničkih sredstava.

Simbolički modeli prikazuju ponašanje sistema pomoću skupa matematičkih i logičkih relacija. Specifičan slučaj simboličkih modela su matematički modeli.

Metod crne kutije primenjuje se za objekte istraživanja kod kojih su dostupne samo ulazne i izlazne veličine, dok je unutrašnje vraćanje nepoznato. Ponašanje se određuje pomoću promena izlaznih veličina koje su nastale usled promena ulaznih veličina.

Na svoj način se izučavaju sistemi čije je unutrašnje uređenje nepoznato ili su više složena, da bi je izneli zahvaljujući ponašanju sistema na osnovu ponašanja elemenata i strukture veza među njima.

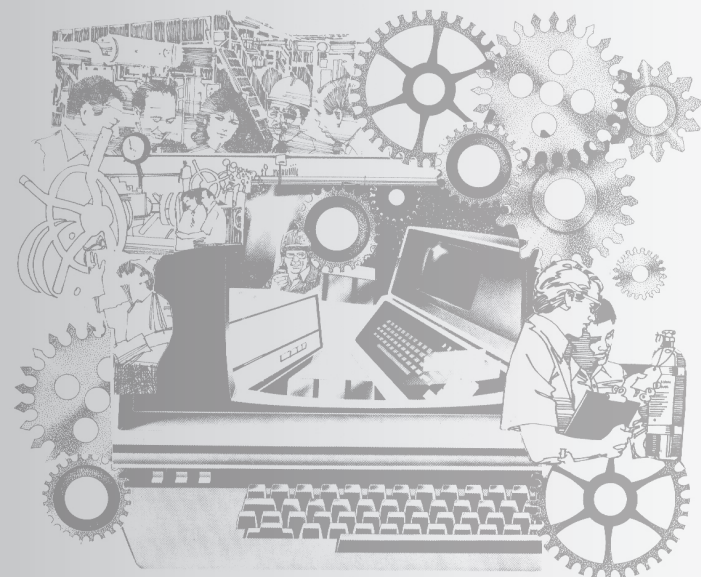
Upravljanje sistemima

Upravljanje složenim sistemima se sprovodi radi ostvarivanja zadatih ciljeva na osnovu prijema predavanja i prerade informacija koje služe kao osnova za preduzimanje odgovarajućih mera u postupku upravljanja. Prirodni sistemi su nastali pod uticajem prirodnih zahteva bez učešća čoveka. Ako uzmemo za primer sisteme onda je jedan od osnovnih ciljeva ovih sistema obezbeđenje optimalnih uslova za opstanak i za umnožavanje. Proces ostvarivanja ovih ciljeva je određen prirodnim zahtevima i ogleda se u adaptaciji prema spoljnim uslovima, što se objašnjava stalnom evolucijom sistema. Kada su u pitanju sistemi tj. sistemi koje je stvorio čovek, onda se problem zasniva na drugi način.

Ciljeve veštačkih sistema određuje čovek. Ako posmatramo preduzeće kao veštački sistem onda cilj predstavlja ostvarivanje plana i programa.

Upravljanje je postupak preduzimanja određenih akcija tako da se sistem dovede u stanje ostvartivanja postavljenog cilja. To je u osnovi novo stanje koje se razlikuje od stanja u kojem bi se inače sistem našao u slučaju odsustva usmerene akcije od strane subjekta upravljanja. Izvoz stanja sistema kojem dajemo određenu prednost u odnosu na ostala stanja predstavlja donošenje odluke.

U postupku donošenja odluke primenjuju se različiti modeli na osnovu kojih se dolazi do izvora prihvatljive alternative.



2. SISTEMSKI PRISTUP

Jedna od osnovnih karakteristika opšte teorije sistema je njena težnja ka stvaranju **jedinstvene naučne metodologije** međudisciplinarnog karaktera, pri čemu osnovu jedinstva čini analogija između procesa koji se zbivaju u sistemima različite prirode.

Sistemske pristupe se zasniva na sledećim polazištima:

- Svi predmeti stvarnosti, bilo materijalni (realno postojeći), bilo zamišljeni, kao **problemi poimaju se složenim i predstavljaju se kao sistem**;
- Složeni predmet kao sistem nalazi se u **okolini** (koja se, takođe, sa nekog stanovišta može posmatrati kao sistem), sa kojom je povezan eksternim vezama;
- **Okolina utiče na promene oblika postojanja, svojstava i stanja sistema, ali i sistem utiče na okolinu**;
- **Delovi koji sačinjavaju složeni predmet kao sistem su «integralni članovi jedne sintetičke, nedeljive povezanosti kojoj pripadaju» - strukture**;
- **Strukturna analiza** složenih predmeta kao sistema je bitan korak u primeni sistemskog prilaza;
- **svi predmeti** u stvarnosti u prostoru i vremenu **menjaju svoje oblike postojanja, svoja svojstva i svoja stanja**;
- Jedan od bitnih zadataka sistemskog prilaza i, na osnovu njega, operacionalizovanih sistemskih postupaka, jeste **pronalaženje mehanizama** tih transformacija u složenim predmetima kao sistemima.

Polazišta sistemskog prilaza zasnivaju se na tri fundamentalna principa:

- **princip složenosti** svih predmeta (stvari i procesa) stvarnosti,
- **princip povezanosti i međusobnog delovanja** objekata stvarnosti po raznim osnovama i
- **princip neprekidnih promena** svih predmeta stvarnosti.

Princip složenosti

Savremena nauka je opovrgla teze o postojanju **apsolutno** prostih elemenata i prostih stvari, time što je otkrivena složenost organizama, složenost ćelija, složenost elemenata (izotopi), složenost atoma itd. Ipak, postoje više ili manje složeni predmeti, postoje relativno izdvojene celine iz stvarnosti, koje i pored svoje složenosti ulaze u sastav složenijih predmeta i smatraju se, a objektivno i jesu, elementi, tj. relativno prosti sastavni delovi složenijih predmeta. Prosti predmeti predstavljaju samo **granične** slučajeve složenih predmeta.

Princip povezanosti i međusobnog delovanja

U svakodnevnom govoru pominju se porodične i rodbinske veze, saobraćajne veze, telefonske veze, informatičke veze, energetske veze, žičani i bežični prenos signala, itd.

Gotovo je nemoguće naći dva predmeta ili pojave koji nisu baš ni u kakvoj vezi.

Veze sačinjavaju mreže, koje se dalje hijerarhijski stepenuju po složenosti.

Predmet sistemskog prilaza je da identifikuje i modelira interne i eksterne veze složenih predmeta kao sistema.

Princip neprekidnih promena

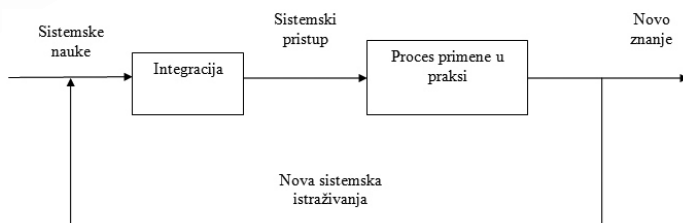
U svetu su stalne jedino promene: i kosmosa, i prirode, i čoveka, i društva. Metodička skepsa je sastavni i nezaobilazni deo u naučnim istraživanjima. Uz pitanje promena vezana su još neka pitanja:

- ♦ pitanje **uzroka**, zakona i posledica razvoja (princip kauzaliteta, princip finaliteta).
- ♦ pitanje **određenosti** promena, ili određenosti ponašanja sistema (princip determinizma).

U razvoju filozofije postojali su i drugi pravci, na primer: **dijalektički prilaz** i **dijalektička metoda**, **geštaltizam** (Gestalt), **funkcionalizam**, **strukturalizam** itd., koji su takođe istraživali složenost predmeta stvarnosti.

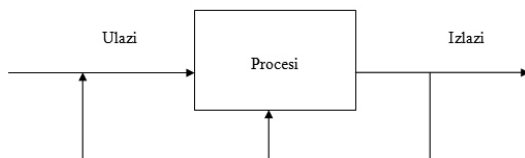
Metodološka osnova sistemskog pristupa

Sistemski pristup proučavanja inženjerskih procesa i uopšte sistema čine metodološke osnove sistemskih nauka. Sistemski pristup se ostvaruje uz pomoć primene intelekta (mišljenja), tehnike i sredstava zasnovanih na sistemskom mišljenju i opštoj teoriji sistema.



Sl. 2.1 Sistemski pristup proučavanja nekog procesa

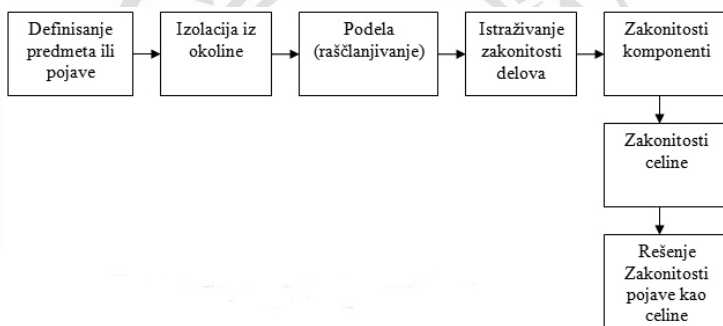
Koncept provođenja tog sistemskog pristupa prikazan je na sledećoj slici. Pri sistemskom pristupu se analiziraju ulazne veličine, izlazne veličine, sami procesi, i cilj je da li su izlazne veličine u granicama definisanih (projektovanih) vrednosti. Ako nisu vrši se korekcija ulaznih veličina i ponekad korekcija samog procesa. Na osnovu sistema pravi se **model**, a na osnovu njega vrši se **istraživanje** i stiču se **nova znanja**.



Sl. 2.2 Koncept provođenja sistemskog pristupa

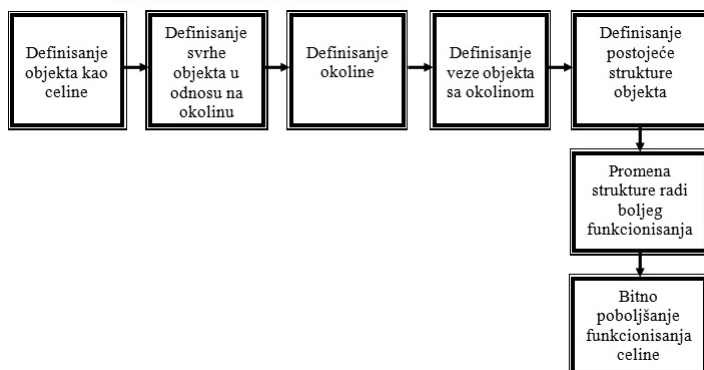
Kod sistemskog pristupa proučavanja traži se **optimalno rešenje**. Na modelu sistema vrše se **eksperimenti** koji trebaju da sadrže ključne karakteristike originala a **kriterijum ocenjivanja** treba da odluči **šta treba korigovati** na prvobitnom objektu.

Kod klasičnog pristupa proučavanja inženjerskih procesa vrši se otkrivanje zakonitosti pojava koje se proučavaju i odatle dolazi do određenih zaključaka. Karakterističnu pojavu u sistemu koji proučavamo, izolujemo iz okoline, te je proučavamo, analiziramo i raščlanjujemo.



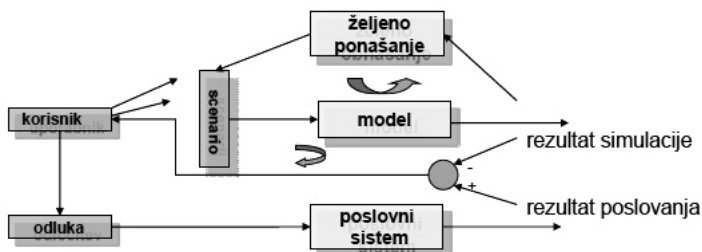
Sl. 2.3 Klasičan pristup proučavanja inženjerskih procesa i sistema

Kod sistemskog pristupa proučavanja određenih inženjerskih procesa i sistema zadatak je poboljšavanje funkcionisanja samog sistema.



Sl. 2.4 Sistemski pristup proučavanja inženjerskih procesa i sistema

Izabranu metodologiju (sredstva) u inženjerskom i poslovnom svetu proveravamo, tj. analiziramo se i poredi ponašanje originala i simulacionog modela. Ukoliko je ostvarena saglasnost, simulacioni model može da se koristi za dalje proučavanje sistema-originala.



Sl. 2.5 Metodologija sistemskog pristupa

Kretanje

Termin »kretanje« se u mehanici primenjuje u uskom smislu te reči i označava promenu položaja nekog objekta u prostoru i vremenu. U kibernetici kretanje ima mnogo opštiji smisao, a to je — *svaka promena objekta u toku vremena**.

* Ovo je u suštini, tradicionalno gledište dijalektike. Opšte poznata formula **Hegela** glasi: »Kretanje je promena uopšte«.

Kretanjem se nazivaju, na primer, promena temperature tela, promena punjenja kondenzatora, promena zapremine ili pritiska gasa, promena sume na tekućem računu u banci, promena zaliha sirovina na skladištu, na kraju, čak i takvi procesi, kao život i mišljenje, takođe mogu i moraju da se posmatraju kao određeni, mada i veoma složeni, oblici kretanja.

Pošto u zakonitostima kretanja najraznovrsnijih objekata ima mnogo zajedničkog, osobito sa stanovišta upravljanja procesima koji se u njima odvijaju, korisno je da se ne razmatraju zakoni kretanja konkretnih sistema (kojih je veoma mnogo), već apstraktnih kibernetičkih sistema. Metode opisivanja kretanja, izložene ovde, biće neophodne za izlaganje materije u nastavku kursa.

Stanje sistema i prostor stanja

Stanje bilo kog sistema može se, s određenom tačnošću, okarakterisati kao skup vrednosti veličina, koje određuju njegovo ponašanje. Ove veličine dozvoljavaju da se međusobno porede stanja odvojenih sistema i ocene njihove razlike, a i da se sravnjuju stanja jednog istog sistema u različitim momentima radi rasvetljavanja njegovog kretanja.

Postoje različiti načini opisivanja stanja sistema. Moguće je, na primer, nabrojati vrednosti svih veličina $X_1, X_2, \dots, X_n$, koje određuju stanje sistema u određenim trenucima i dati spisak njihovih vrednosti u fiksiranim momentima. U tom slučaju niz stanja sistema može se predstaviti u vidu tablice. Na primer, stanje bolesnika obolelog od akutnog nefritisa može se okarakterisati sa tabelom 2.1.

Tabela 2.1. Niz stanja bolesnika obolelog od akutnog nefritisa

Datum, vreme	Temperatura	Arter. krvni pritisak [mm Hg]		Preostali azot mg %
		Maksimalan	Minimalan	
7. II, 9 čas.	37,8	190	120	103
8. II, 9 čas.	37,6	180	120	95
9. II, 9 čas.	37,4	170	95	89
10. II, 9 čas.	37	165	90	90
...	...			
16. II, 9 čas	36,7	145	85	49

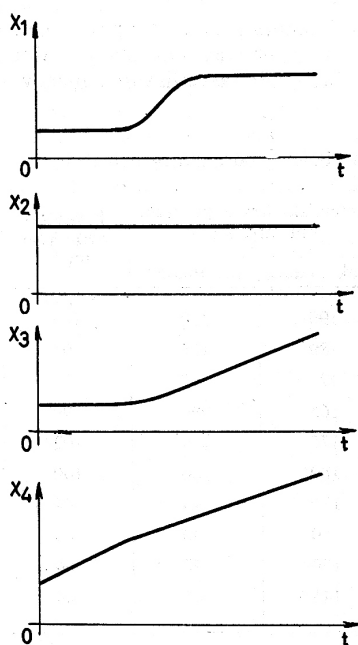
Stanje sistema se može okarakterisati grafički, prikazujući vrednost svake veličine iz skupa veličina X_1, X_2, X_n tačkom na brojnoj osi, čiji položaj u određenoj razmeri odgovara vrednosti veličine X_i ($i=1, 2, \dots, n$). Ako se stanje sistema menja vremenom, tada će se, pri ovakvom načinu opisivanja stanja sistema, njegovo kretanje predstavljati skupom grafika: $X_1(t), X_2(t), \dots, X_n(t)$, gde je t —vreme, računato od nekog momenta, uslovno izabranog za početak računanja vremena. Na sl. 2.6 je pokazan skup grafika koji opisuju kretanje broda. Ovde je X_1 —kurs broda, X_2 —njegova brzina, X_3 i X_4 —dužina i širina položaja broda.

Međutim, mi ćemo koristiti, po pravilu, drugi, za naše ciljeve pogodniji način predstavljanja stanja i kretanja sistema — način koji se bazira na pojmu **prostora stanja sistema**.

U prethodnom izlaganju smo već koristili način prikazivanja vrednosti neke veličine tačkom na brojnoj osi, koja predstavlja prostor s jednom dimenzijom. Ako se zahteva da se u vidu tačke prikaže skup dveju veličina (X_1 i X_2), već tada jednodimenzionalni prostor postaje nedovoljan i u tom slučaju mora da se koristi dvodimenzionalni prostor. Uzmimo ravan sa zadanim ortogonalnim koordinatnim sistemom u njoj, kao što je pokazano na sl. 2.7. Ovde tačka a prikazuje stanje sistema, koje karakterišu vrednosti veličina: $X_1=X_{1a}, X_2=X_{2a}$, a tačka b — u skladu stim: $X_1=X_{1b}, X_2=X_{2b}$.

U slučaju kada stanje sistema određuju tri veličine, ono se, očigledno, može opisati tačkom u trodimenzionalnom prostoru, kao što je pokazano na sl. 2.8.

Označićemo sa n broj veličina koje određuju stanje razmatranog sistema. Za $n=1, n=2$ i $n=3$ stanje sistema se može očigledno prikazati u prostorima s brojem dimenzija jednakim n . Ako je $n>3$ gubi se mogućnost jasnog predstavljanja stanja sistema.



Sl. 2.6. Grafički prikaz kretanja broda

Uprkos tome, rasuđivanjem je moguće izvesti veoma važne zaključke o osobinama sistema, ako se koristi predstavljanje stanja sistema u vidu tačke u odgovarajućem prostoru i u slučajevima kada je $n > 3$. Pri tome se mora, istina, koristiti pojam višedimenzionalnog prostora (ili, kako ga još nazivaju, *hiperprostor*), ali nas ta okolnost ne sme obeshrabriti. Mada je pojam n — dimenzionalnog prostora apstraktan, njegova svojstva su u mnogo čemu slična svojstvima, za nas uobičajenih, jednodimenzionalnih, dvodimenzionalnih i trodimenzionalnih prostora.

Naime, jedan od osnovnih geometrijskih pojmova — rastojanje između dve tačke — može da se uvede za n — dimenzionalni prostor na isti način kao i za trodimenzionalni. Rastojanje d između tačaka a i b u trodimenzionalnom prostoru nije ništa drugo već dužina dijagonale paralelopipeda, pokazanog na sl. 2.9, u čijim se temenima nalaze te tačke, a ivice su paralelne s koordinatnim osama.

Kao što je poznato iz stereometrije, dužina d dijagonale pravougaonog paralelopipeda s ivicama:

$$x_1 = X_{1b} - X_{1a}, x_2 = X_{2b} - X_{2a}, x_3 = X_{3b} - X_{3a}$$

nalazi se iz izraza

$$d = \sqrt{x_1^2 + x_2^2 + x_3^2} \quad (2.1)$$

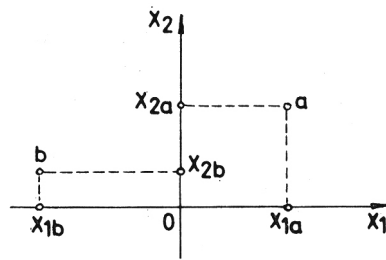
Analogno ovome, možemo za n — dimenzionalni prostor rastojanje između tačke a s koordinatama $(X_{1a}, X_{2a}, \dots, X_{na})$ i tačke b s koordinatama $(X_{1b}, X_{2b}, \dots, X_{nb})$ da odredimo kao veličinu

$$d = \sqrt{x_1^2 + x_2^2 + \dots + x_n^2} \quad (2.2)$$

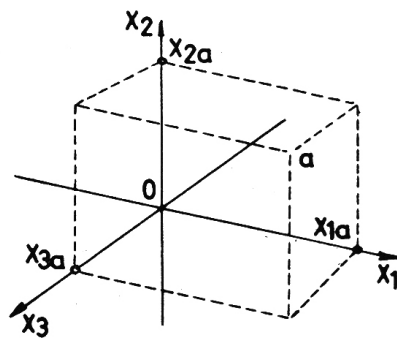
gde su:

$$x_1 = X_{1b} - X_{1a}, x_2 = X_{2b} - X_{2a}, \dots, x_n = X_{nb} - X_{na}.$$

Veličine $x_1, \dots, x_n$ jednake su dužinama ivica n — dimenzionalnog paralelopipeda, prikazanog na sl. 2.9.

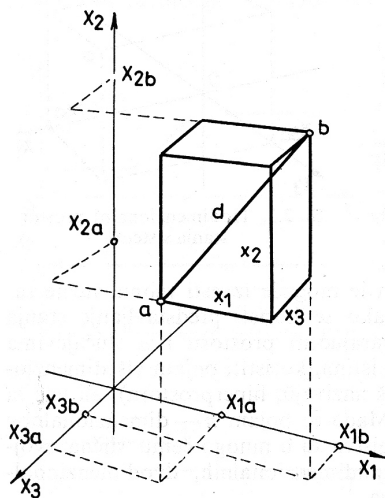


Sl. 2.7. Dvodimenzionalni prostor stanja sistema



Sl. 2.8. Trodimenzionalni prostor stanja sistema

Prostor u kome je rastojanje određeno formulom (2.2) naziva se **Euklidov prostor**. U njemu važe sve teoreme uobičajene geometrije tj. Euklidove geometrije, posebno Pitagorina teorema.



Sl. 2.9. Rastojanje između tačaka u Euklidovom prostoru

Prostor u kome se svako stanje sistema prikazuje određenom tačkom nazvaćemo *prostor stanja sistema*. Broj dimenzija prostora stanja je jednak broju nezavisnih veličina koje određuju stanje sistema. Ove nezavisne promenljive se često nazivaju *stepeni slobode* sistema. Svako stanje sistema se karakteriše skupom određenih vrednosti promenljivih $X_1, X_2, \dots, X_n$. U prostoru stanja njemu odgovara tačka s istim vrednostima koordinata $X_1, X_2, \dots, X_n$. Ova tačka se naziva *reprezentativna tačka* (ona »reprezentuje« dato stanje sistema), a promenljive $X_1, X_2, \dots, X_n$ se nazivaju **koordinate sistema**.

U stvarnim sistemima ne mogu sve njihove koordinate da se menjaju u neograničenim opsezima (na primer, $-\infty < X < \infty$). Veliki broj koordinata može da dobije samo vrednosti koje leže u ograničenom intervalu, tj. koje zadovoljavaju uslov

$$X'_i \leq X_i \leq X''_i$$

gde su X'_i i X''_i — granice intervala mogućih vrednosti koordinata X_i . Oblast prostora stanja, u kojoj se može nalaziti reprezentativna tačka, naziva se *oblast dopuštenih stanja*. U daljem tekstu, govoreći o prostoru stanja, imaćemo u vidu samo njegovu dopuštenu oblast.

Međutim, i u granicama oblasti dopuštenih stanja, svaka tačka ne prikazuje moguće stanje sistema. Takvu osobinu poseduje samo *neprekidan prostor stanja*, koji odgovara sistemu čije koordinate mogu da imaju *bilo koju vrednost* (u dozvoljenim granicama).

Ali postoje sistemi koji se nazivaju diskretni, u kojima koordinate mogu da imaju samo konačan broj određenih vrednosti. I prostor stanja ovakvih sistema je *diskretan*. U ovom slučaju reprezentativna tačka može da zauzme samo konačan broj S položaja

$$S = s_1 \bullet s_2 \bullet \dots s_n,$$

gde je s_i — broj diskretnih stanja i -te koordinate. Vrednosti koordinata sistema, koji se nalazi u kretanju, menjaju se vremenom. Pri tome, reprezentativna tačka menja svoj položaj u prostoru stanja opisujući neku trajektoriju.

Ulazne i izlazne veličine

Kretanje sistema — promena njegovog stanja — može da nastane kako pod uticajem spoljnjih dejstava, tako i kao rezultat procesa koji se odvijaju unutar samog sistema.

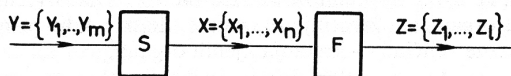
Na svaki sistem, strogo govoreći, utiče bezbrojno mnoštvo različitih spoljnjih dejstava, ali nisu svi oni bitni. Tako, očigledno, sila privlačenja Meseca nema bitnog uticaja na kretanje automobila u odnosu na Zemlju, mada u principu takav uticaj postoji. Iz mnoštva svih dejstava biraju se samo ona koja bitno utiču na stanje sistema u uslovima rešavanog zadatka. Ova spoljnja dejstva se nazivaju *ulazne veličine* (ili ulazna dejstva, ulazne promenljive sistema), a elementi sistema, kojima su pridružena ulazna dejstva — *ulazi* sistema.

Na kretanje aviona, na primer, bitno utiču takvi faktori, kao što su jačina i pravac vetra, gustina atmosfere, položaj upravljača i vučna sila motora. Svi ti faktori se smatraju kao ulazna dejstva na avion.

Često se pokazuje korisnim da se smatraju izlaznim veličinama sistema ne koordinate X , koje određuju njegovo stanje, već neke druge veličine Z , koje su jednoznačno određene koordinatama tog sistema. Pri tome je svaka od l izlaznih veličina Z_i povezana s koordinatama sistema svojom funkcionalnom zavisnošću.

$$Z_i = \Phi_i(X) \quad (i=1,2,\dots,l) \quad (2.3)$$

Upravljeni sistem se u ovom slučaju može predstaviti u vidu dela S , koji pretvara ulazna dejstva Y u koordinate X , i skupa funkcionalnih pretvarača Φ , koji pretvaraju koordinate sistema u izlazne veličine (sl. 2.10).



Sl. 2.10. Šema pretvaranja ulaznih veličina u izlazne: Y — ulazne veličine, X — koordinate upravljano sistema, Z —izlazne veličine, S —pretvarač ulaznih veličina u koordinate, F —pretvarač koordinata u izlazne veličine

Neophodno je uzeti u razmatranje izlazne veličine, koje ne ulaze neposredno u skup koordinata a određuju stanje sistema, u onim slučajevima kada se zadatak upravljanja ne sastoji u tome da se sistem dovede u zadato stanje, već u postizanju ciljeva funkcionalno povezanih sa stanjem upravljanog sistema. Zadatak upravljanja procesom izrade sintetičkog vlakna, na primer, sastoji se u dobijanju vlakna zahtevane jačine Z_1 i elastičnosti Z_2 .

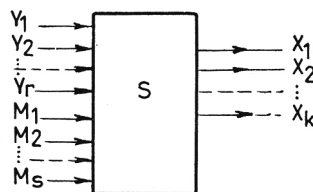
Ove veličine su funkcionalnom zavisnošću povezane s koordinatama procesa: temperaturom mase (X_1), i sastavom primesa ($X_2, X_3, \dots$) u osnovnoj sirovini i sl. Jasno je da u sličnim slučajevima treba razlikovati izlazne veličine od koordinata koje karakterišu stanje sistema.

Kod rešavanja zadataka upravljanja važno je razlikovati dva tipa ulaznih veličina: **upravljačka** dejstva i **poremećajna** dejstva. Upravljačkim dejstvima pripadaju veličine čije se vrednosti mogu upotrebljavati pri upravljanju sistemom, i koje možemo menjati s ciljem da se ostvari ono kretanje, kome dajemo prednost u odnosu na druga moguća kretanja upravljanog sistema. U navedenom primeru aviona upravljačka dejstva su dejstva stvorena krilima za upravljanje i vučna sila motora, koje pilot upotrebljava po svom nahođenju.

U poremećajna dejstva spadaju ostala bitna dejstva na sistem, kao što su, na primer, uticaj vetra i gustina atmosfere na kretanje aviona. Tamo gde treba razlikovati vrste ulaznih dejstava označavaćemo upravljačka dejstva simbolima $Y_1, Y_2, \dots, Y_r$, a poremećajna dejstva simbolima $M_1, M_2, \dots, M_s$.

Dejstvo sistema na okolnu sredinu karakterišu vrednosti njegovih *izlaznih veličina*. Skup izlaznih veličina i njihovih promena određuje ponašanje sistema; zapravo, one omogućuju spoljnjem posmatraču da ocenjuje saglasnost kretanja sistema sa ciljevima upravljanja. U primeru upravljanja kretanjem aviona izlazne veličine su njegov kurs i brzina kretanja, pošto vrednosti tih veličina određuju pravac i brzinu prenošenja tovara. Cilj upravljanja se u datom slučaju sastoji u tome da se tovar dopremi na zadato mesto za dato vreme. Ulazna dejstva na organizam životinje su, naime, dejstva koja primaju njena čula, a izlazne veličine — kretanja njenih organa.

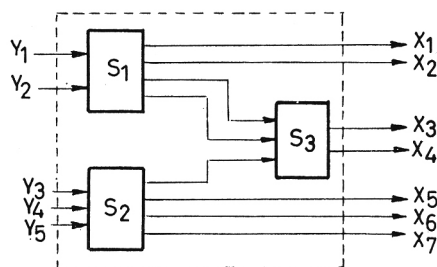
Promena ulaznih veličina, po pravilu, izaziva promenu izlaznih veličina. Međutim, promene izlaznih veličina se ne javljaju uvek odmah; mogu ponekad da zakasne, ali ne mogu nikada da preteču promene ulaznih veličina, jer su prve posledica, a druge — uzrok kretanja sistema. Na sl. 2.11 šematski je prikazan sistem S sa ulaznim i izlaznim veličinama koje se odnose na njega.



Sl. 2.11. Ulazne i izlazne veličine

Treba приметiti, da poremećajna dejstva, koja utiču na kretanje sistema, ne moraju biti samo spoljnjeg porekla, već mogu nastati i unutar sistema, na primer, usled promene svojstava njegovih elemenata posle dužeg rada i, uopšte, kao rezultat narušavanja normalnog funkcionisanja elemenata sistema.

Ponekad je podesno sisteme razmatrati raščlanjene na delove, koji međusobno deluju jedni na druge. U ovom slučaju neke izlazne veličine mogu istovremeno biti ulazne veličine drugog dela sistema, kao što je prikazano na sl. 2.12.



Sl. 2.12. Primer uzajamne povezanosti delova sistema

Pretvaranje

Kretanje sistema mozemo posmatrati kao niz pretvaranja/transformacija njegovih stanja. Može se smatrati da je prelaz sistema iz stanja a_1 u momentu t_1 u stanje a_2 u trenutku t_2 rezultat pretvaranja (a_1, t_1) u (a_2, t_2) . Promene izlaznih veličina nekog sistema ili elementa, nastale pod uticajem promena ulaznih dejstava mogu se posmatrati i kao njihova pretvaranja.

Govori se da se *pretvaranje* jednog objekta u drugi ostvaruje delovanjem *operatora* na objekt. Objekt, podvrgnut pretvaranju, naziva se *operand*, a rezultat pretvaranja — *lik* (operator »preslikava« jedan objekt na drugi).

Koristeći se sada uvedenim terminima može se ovako opisati svako pretvaranje: *rezultat delovanja operatora na operand je da se operand pretvara u lik*. Razume se, pri sledećem delovanju operatora na lik, dobijen u prethodnom pretvaranju, mora taj lik da se posmatra kao operand. Uzastopni prelaz sistema u stanja $a_0, a_1, a_2, \dots$ može da nastane kao rezultat delovanja operatora P prema šemi:

Korak	Operand	Pretvaranje	Lik
1	a_0	Pa_0	a_1
2	a_1	Pa_1	a_2
3	a_2	Pa_2	a_3
...	...	...	...

Označavajući n -tostruko delovanje operatora P sa P^n , dobijamo

$$a_n = P^n a_0$$

Ako smatramo izlaznu koordinatu sistema X rezultatom pretvaranja ulazne veličine Y , tada vezu između X i Y možemo napisati u obliku:

$$X=KY,$$

gde je K —operator koji karakteriše osobine posmatranog sistema. Ako sistem ima n izlaznih i m ulaznih koordinata, tada je, u skladu s ovim,

$$\{X_1, X_2, \dots, X_n\} = \{K\} \{Y_1, Y_2, \dots, Y_m\}.$$

Ovde je sa $\{K\}$ simbolički označen čitav skup pretvaranja, koji uzima u obzir uticaj svakog ulaza na svaki izlaz.

Ako je posmatrani sistem neinercioni linearni pretvarač (takav, na primer, kao što su: elektronski pojačavač ili mehanički reduktor ili fotoelement), tada operator K dobija smisao koeficijenta pretvaranja (koeficijenta prenosa) i predstavlja broj k s kojim treba pomnožiti ulaznu veličinu da bi se dobila vrednost izlazne veličine pretvarača

$$X=kY.$$

Za nelinearni neinercioni pretvarač izlazna veličina je neka funkcija ulazne veličine, a operator dobija smisao simbola F , koji označava određeno nelinearno pretvaranje

$$X=F(Y).$$

Ako se izlazne koordinate ne uspostavljaju trenutno na vrednostima koje odgovaraju vrednostima ulaznih dejstava, tada je operator složeniji i ne može se izraziti samo algebarskim operacijama nad operandima.

Primeri:

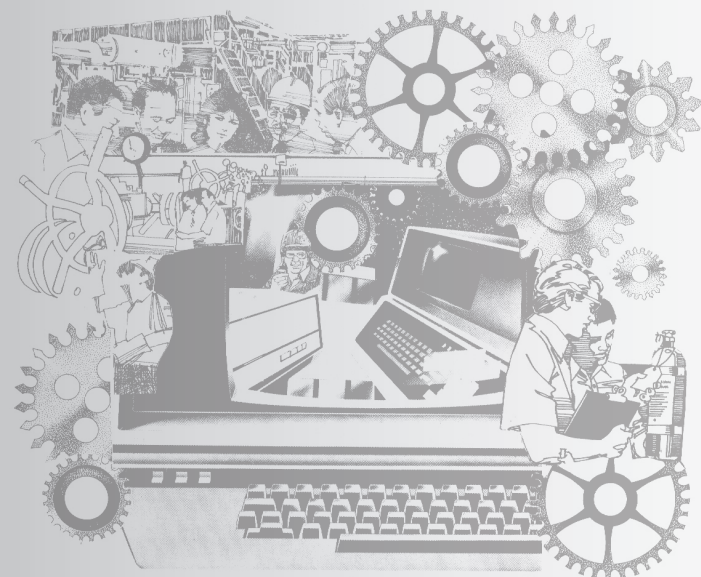
1. Na telefonu se bira šestocifreni broj. Odrediti dimenzionalnost prostora stanja sistema. Šta su koordinate sistema? Kako se ponaša reprezentativna tačka pri uzastopnoj promeni poslednje cifre u broju telefona od 1 do 0 (1, 2, 9, 0)?

Odgovor. Dimenzionalnost prostora stanja je $n=6$. Koordinate sistema odgovaraju ciframa u broju (uzima se u obzir da se čitav broj sastoji iz cifara). Reprezentativna tačka će se kretati paralelno poslednjoj koordinati.

2. Fabrika se specijalizovala za izradu automobila od gotovih delova. Šta su ulazne i izlazne veličine za dati sistem? Šta su u ovom slučaju poremećajna ulazna dejstva?

Odgovor. Osnovna ulazna dejstva su gotovi delovi, isporučeni fabrici, i plan proizvodnje automobila, koji je izradila organizacija za planiranje; izlazna veličina — broj proizvedenih automobila. Poremećajna dejstva nastaju u slučaju prekida isporuke gotovih delova, a i pri narušavanju ritma rada unutar preduzeća.





3. MODEL

Postojanje sličnih crta među različitim objektima davno je stavljeno u osnovu naučnog prilaza izučavanju prirode najraznovrsnijih pojava. U suštini, u svim naukama, u očiglednoj ili neočiglednoj formi, uvodi se pojam modela, koji odražava slične osobine izučavanih pojava i objekata. Ali nigde se koncepcija modeliranja ne ostvaruje tako jasno i dosledno kao u kibernetici (teoriji sistema), gde ono figuriše u najopštijem vidu i predstavlja fundamentalan pojam, određujući metodologiju izučavanja ponašanja sistema.

Problemi konstruisanja i korišćenja modela postavljaju se i rešavaju sa različitih aspekata. Nas može interesovati model, koji se razlikuje od originala razmerom geometrijskih dimenzija ili brzinom odvijanja procesa, a koji predstavlja određenu olakšicu za eksperimentalna istraživanja. Često je podesno izučavati osobine objekta, koristeći model druge fizičke prirode, oslanjajući se na formalnu sličnost jednačina, koje opisuju kretanje originala i modela. Veliki značaj za nauku ima pojam uprošćenog modela, koji omogućuje da se prouče veoma složeni objekti i sistemi, zadržavajući u modelu samo one karakteristike originala koje su bitne za oblast izučavanih pojava. Poslednja okolnost je osobito važna, jer dovodi do shvatanja da teorija uopšte, uvek i svuda razmatra ne realne objekte (čija je složenost bezgranična), već njihove idealizovane modele.

Original i model

Pojam modela se zasniva na postojanju neke sličnosti između dva objekta. Pri ovome se reči »**sličnost**« i »**objekt**« shvataju u veoma širokom smislu. Sličnost može biti čisto spoljna, ona može da se odnosi na unutrašnju strukturu spolja sasvim različitih objekata ili na određene crte ponašanja objekata, koji nemaju ničeg zajedničkog ni po obliku ni po strukturi. Pojam sličnosti se primenjuje na vrlo široku klasu materijalnih objekata, uključujući objekte žive i mrtve prirode, veštačke objekte koje je stvorio čovek, likove, simbole itd.

Ako se između dva objekta može ustanoviti sličnost u bilo kakvom određenom smislu, tada između tih objekata postoji odnos originala i modela.

Ovo znači da se jedan od tih objekata može smatrati *originalom*, a drugi, njegovim *modelom*.

Sličnost originala i modela označavaćemo znakom $\sim$ tako, da ako je objekt A model objekta B, to će se obeležavati u obliku $A \sim B$. Pri tome će uvek važiti i $B \sim A$, pošto je sličnost objekata uvek uzajamna. Odnosi original-model mogu važiti ne samo između dva, već i između bilo kog broja objekata. Tako, na primer, za skup objekata $A \sim B \sim C \sim D$ bilo koji od njih, na primer B može se posmatrati kao model objekata A, C i D ili kao original za modele A, C i D.

Spoljnu sličnost — sličnost oblika — poseduju takvi objekti, kao što su brod i njegov lik (u vidu slike, trodimenzionalnog modela ili kompleta crteza broda), metalni odlivak i njegov drveni model. Sličnu strukturu mogu da imaju: sistem upravljanja državom i njegova strukturna šema, gradska vodovodna mreža i gradska električna mreža. Za kibernetiske sisteme najvažnija sličnost među sistemima, koja dovodi do odnosa original-model, je sličnost njihovih ponašanja, što dozvoljava da se modelira kretanje. U osnovi modeliranja ponašanja leži činjenica da se jednako ponašanje može opažati, pri određenim uslovima, kod sistema bitno različitih po obliku, strukturi i po fizičkoj prirodi procesa koji se odvijaju u njima.

Crna kutija

Za postavljanje i rešavanje zadataka modeliranja upravljanih sistema pokazao se korisnim pojam »**crne kutije**«. Pod crnom kutijom se podrazumeva sistem, za koji su spoljnjem posmatraču dostupne samo ulazne i izlazne veličine, a njegovo unutrašnje uređenje mu je nepoznato. Pri tome se pokazuje da se niz važnih zaključaka o ponašanju sistema može izvesti posmatrajući samo promene izlaznih veličina nastale usled promena ulaznih. Takav prilaz, naime, stvara mogućnosti da se objektivno prouče sistemi, čije je ustrojstvo ili nepoznato ili suviše složeno da bi bilo moguće izvesti zaključke na osnovu ponašanja sastavnih delova tih sistema i struktura veza među njima.

Neka je ponašanje sistema određeno njegovim ulaznim dejstvima $Y_1, Y_2, \dots, Y_m$ i izlaznim dejstvima $X_1, X_2, \dots, X_n$ (sl. 3.1).

Posmatrajući ponašanje sistema dovoljno dugo i, ako je potrebno, vršeći neke aktivne eksperimente sa njim, može se postići takav stepen poznavanja svojstava sistema, da postoji mogućnost predviđanja kretanja njegovih izlaznih koordinata pri bilo kakvim zadanim promenama na ulazima. Međutim, ma kako detaljno da proučimo ponašanje crne kutije, ne možemo izvesti obrazložene zaključke o njegovom unutrašnjem ustrojstvu, jer jedno isto ponašanje mogu da poseduju različiti sistemi. **Sistemi koje karakterišu jednaki skupovi ulaznih i izlaznih veličina i jednako reagovanje na spoljnja dejstva, nazivaju se izomorfni.**



Sl. 3.1 Crna kutija

Izomorfni sistemi se, očigledno, ne razlikuju jedni od drugih za posmatrača kome su dostupne samo njihove ulazne i izlazne koordinate. Uslovi izomorfnosti sistema A i B mogu se izraziti sledećim sistemom jednačina ako je:

$$Y_{1A}(t) = Y_{1B}(t), Y_{2A}(t) = Y_{2B}(t), \dots, Y_{mA}(t) = Y_{mB}(t),$$

tada je:

(3.1)

$$X_{1A}(t) = X_{1B}(t), X_{2A}(t) = X_{2B}(t), \dots, X_{nA}(t) = X_{nB}(t),$$

za bilo koji moment t .

Napomena: Pod aktivnim eksperimentom podrazumevamo dejstvo istraživača na ulazima sistema s ciljem da se prouči njegovo ponašanje, za razliku od pasivnog eksperimenta, koji se sastoji samo u posmatranju ponašanja sistema,

Na taj način, proučavanje sistema metodom crne kutije principijelno ne može dovesti do jednoznačnog zaključka o njegovoj unutrašnjoj strukturi, pošto se ponašanje datog sistema, posmatranog kao crna kutija, ni po čemu ne razlikuje od ponašanja svih sistema koji su s njim izomorfni.

Pri ovome treba paziti na to da se za bilo koji konkretan sistem može izabrati neograničeno mnoštvo konkretnih, s njim izomorfni, sistema.

Među bilo kojim izomornim sistemima, očigledno, postoji odnos original-model u smislu da se svaki iz skupa izomornih sistema može smatrati modelom ili originalom ostalih. Ali, uslovi izomorfnosti nisu jedini uslovi saglasnosti modela originalu. Sistem A može da služi kao model ponašanja sistema B i tada kada njihova sličnost nije tako potpuna kako to zahtevaju uslovi (3.1).

Pojam »crne kutije« se mnogo koristi u nauci i tehnici, mada, istina, ne uvek u očiglednom vidu. U suštini, **crna kutija je bilo koji objekt o kome donosimo zaključke na osnovu proučavanja njegovih spoljnjih svojstava, ne pribegavajući istraživanju njegove strukture i osobina najsitnijih elemenata od kojih je sastavljen dati objekt.** Tako izrađujemo i koristimo električne provodnike, ne ulazeći u prefinjenosti mehanizma provođenja električne struje kroz metale; obrađujemo agrotehniku biljaka na osnovu proučavanja njihovog ponašanja a ne molekularne strukture.

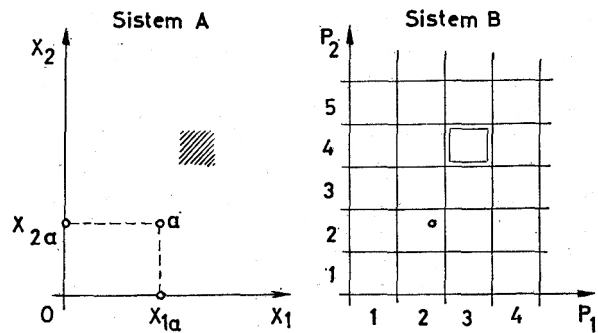
Metod crne kutije je posebno važan za izučavanje ponašanja složenih sistema. A pošto su kibernetički sistemi složeni i pošto pri rešavanju zadataka upravljanja nas interesuje najviše ponašanje sistema, metod crne kutije je osnovni metod za proučavanje kibernetičkih sistema.

Uprošćeni model

Među koordinatama sistema, koje određuju njegovo stanje, mogu postojati glavne i sporedne koordinate u odnosu na zadatak koji rešava istraživač tog sistema. Ako isključimo iz razmatranja sporedne koordinate, tada umesto polaznog sistema A s dimenzionalnošću

prostora stanja n dobijamo prostiji sistem B s dimenzionalnošću prostora stanja $n' < n$. Pri tome, svakom zadatom stanju sistema A odgovaraće po jedno stanje sistema B (jer je značaj vrednosti sporednih koordinata takav, da ne ometa određivanje stanja B prema glavnim koordinatama sistema A). Ali, svakom određenom stanju sistema B neće odgovarati jedna vrednost stanja sistema A , pošto, zadajući stanje sistema B , mi ne utvrđujemo vrednosti sporednih koordinata. Zato će sva stanja sistema A pri bilo kakvim kombinacijama vrednosti sporednih koordinata odgovarati jednom stanju sistema B , određenom zadatim vrednostima glavnih koordinata.

Uprošćavanje polaznog sistema postiže se i objedinjavanjem nekog mnoštva njegovih stanja u jedno; na primer, prelazom sa razmatranja svih mogućih položaja reprezentativne tačke u prostoru stanja na razmatranje samo toga, da li ona pripada jednoj ili drugoj oblasti, na koje se razbija prostor stanja. Takva zamena, takođe, dovodi do jednoznačnog slaganja stanja sistema A sa stanjima sistema B , i do nejednoznačne obrnute saglasnosti. Zapravo, kao što se vidi sa sl. 3.4, zadajući koordinate tačke a u sistemu A , mi jednoznačno zadajemo kvadrat koji određuje stanje sistema B [kvadrat (2,2)]. Ali, zadajući kvadrat sistema B ne određujemo jednoznačno položaj tačke u sistemu A , jer zadatom kvadratu (3,4) odgovaraju sva stanja sistema A , koja leže u ošenoj oblasti njegovog prostora stanja.



Sl. 3.2. Homomorfizam prostora stanja sistema

Sistem B , dobijen od polaznog sistema A njegovim uprošćavanjem (na račun smanjenja broja posmatranih koordinata ili grublje ocene njihovih vrednosti), naziva se *homomorfni* ili *uprošćeni model sistema A*. Odnos između originala A i njegovog homomornog modela B je neravnotežan, pošto se A ne može smatrati homomornim modelom sistema B .

Analogni sistemi

Da bi B služio kao model za A , dovoljno je da, u krajnjoj meri, jedna od izlaznih veličina sistema B u nekoj razmeri bude saglasna sa nekom od izlaznih veličina sistema A , ako se opaža određena saglasnost između uslova na ulazima tih sistema. Ovi zahtevi u pogledu modela mogu se napisati u vidu uslova

$B \sim A$, ako se pri

$$Y_{1A}(t) = k_1 Y_{1B}(k_0 t),$$

$$Y_{2A}(t) = k_2 Y_{2B}(k_0 t),$$

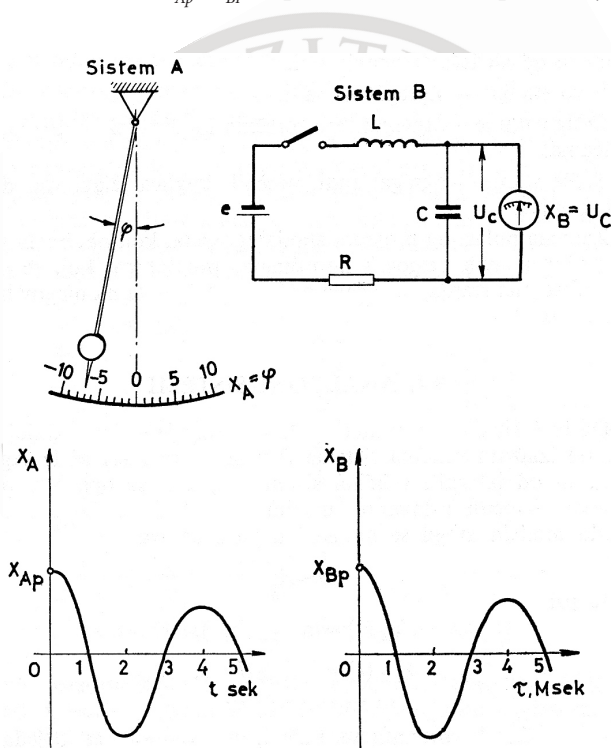
...

među izlaznim koordinatama nalazi, u krajnjoj meri, jedan par X_{iA} i X_{jB} takvih da, ako se u nekom momentu $t = t_0$ ustanovi saglasnost stanja tih sistema, za bilo koji moment važi

$$X_{iA}\{t\} = kX_{jB}\{k_0 t\} \quad (3.2)$$

gde su $k, k_0, k_1, k_2, \dots$ — koeficijenti razmera.

Sistemi, koji zadovoljavaju uslov (3.2), nazivaju se **analogni sistemi**. Analogni su, na primer, takvi sistemi, kao klatno i električno oscilatorno kolo, čije izlazne veličine $\{X_A$ — Ugao odstupanja klatna od vertikalne ose i X_B — napon kondenzatora), pošto su izvedeni iz ravnotežnog stanja, a zatim prepušteni sami sebi, ostvaruju prigušene sinusne oscilacije, kao što je prikazano na sl. 3.3 (X_{Ap}, X_{Bp} — početne vrednosti promenljivih X_A i X_B).



Sl. 3.3. Analogni sistemi: A — mehanički (klatno), B — električni (oscilatorno kolo)

Postojanje analognih sistema je rezultat prisustva formalne sličnosti među nekim karakteristikama ponašanja homomorfnih modela sistema, različitih po svojoj prirodi i uređenju. Ta sličnost se javlja samo posle uprošćenja, koja idu dovoljno daleko, u procesu postavljanja homomorfnih modela polaznih sistema. Ako se pokuša da se odustane od nekih uprošćavanja, može se izgubiti analogija.

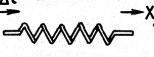
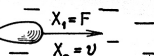
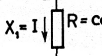
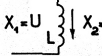
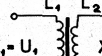
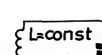
Tako, na primer, ako se uzme u obzir suvo trenje u tački vešanja klatna ili emitovanje elektromagnetnih talasa u oscilatornom kolu, forme kretanja koordinata X_A i X_B neće biti analogne.

Analogne koordinate originala A i njegovog modela B (ili tačnije, koordinate homomorfnih modela A i B) nazivaju se **reprezentativne veličine**. Veza između odgovarajućih reprezentativnih veličina daje se koeficijentima razmera $k_p, k_z, \dots$. Svaki od tih koeficijenata pokazuje koliko jedinica reprezentativne veličine modela dolazi na jedinicu mere odgovarajuće veličine originala. U razmatranom primeru koeficijent razmere, koji povezuje napon kondenzatora s uglom otklona klatna, pokazuje koliko volti napona U_c dolazi na jedan stepen otklona φ klatna. Ovaj koeficijent je izražen jedinicama (volt/stepen).

Razlika u razmerama vremena za procese koji se odvijaju u originalu i modelu, određuje se bezdimenzionalnim koeficijentom k_0 , koji pokazuje koliko se puta procesi u modelu odvijaju brže nego u originalu.

U tabeli 3.1 pokazani su neki analogni sistemi koji se koriste za modeliranje procesa u upravljanim sistemima.

Tabela 3.1 Analogni sistemi

MEHANIČKI ANALOGNI	PNEUMATSKO HIDRAULIČNI ANALOG.	ELEKTRIČNI ANALOGNI
$X_1 = \Delta l$  $X_2 = F$	$X_1 = \Delta l$ $X_2 = P$ 	$X_1 = I$ $X_2 = \phi$ 
$X_1 = F$ $X_2 = v$ 	$X_1 = F$ $X_2 = v$ 	$X_1 = I$ $X_2 = U$ $R = \text{const}$ 
$X_1 = F$ $X_2 = v$ 	—	$X_1 = U$ $X_2 = I$ L 
$X_1 = F_1$ $X_2 = F_2$ 	$X_2 = F_2$ $X_1 = F_1$ 	$X_1 = U_1$ $X_2 = U_2$ L_1, L_2 
$X_1 = F$ $X_2 = A$ $l = \text{const}$ $m = \text{const}$ 	—	$X_1 = U$ $X_2 = I_0$ $L = \text{const}$ C 

Matematički model

Opis sistema nekim formalnim jezikom naziva se njegovim matematičkim modelom, a omogućava da se izvedu zaključci o nekim karakteristikama ponašanja tog sistema, primenjujući formalnu proceduru nad njegovim opisom.

Pošto matematički opis ne može biti sveobuhvatan i idealno tačan, matematički modeli ne opisuju realne sisteme već njihove uprošćene (homomorfne) modele.

Vidovi matematičkih modela su veoma raznovrsni: oni mogu predstavljati karakteristike sistema, zadate funkcionalnim zavisnostima ili graficima; jednačine, koje opisuju kretanja sistema; sl. tablice ili grafici prelaza sistema iz jednih stanja u druga i sl.

Na slici 3.4 prikazan je primer matematičkog modela indukcionog električnog motora, zadatog u obliku skupa karakteristika koje povezuju obrtni moment motora M i njegovu ugaonu brzinu ω za razne vrednosti napona napajanja $U_1, U_2, \dots$. Koristeći taj model može se predvideti, na primer, kako će se menjati ugaona brzina motora pri raznim opterećenjima i raznim vrednostima napona u napojnoj mreži.

Izraz Njutnovog zakona

$$F = \frac{m_1 m_2}{r^2},$$

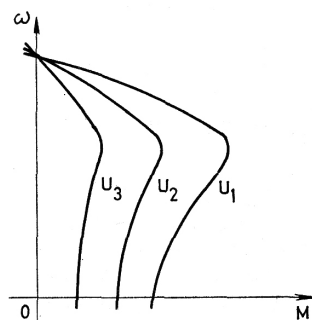
gde su m_1 i m_2 — mase materijalnih tačaka, koje se nalaze na rastojanju r jedna od druge, a F — sila međusobnog dejstva među njima, može se smatrati matematičkim modelom sistema sastavljenog od dve materijalne tačke, pošto ta formula dozvoljava da se izvedu zaključci o njihovom uzajamnom dejstvu u različitim uslovima.

Ponašanje sistema, koji naizmenično prelaze u razna stanja, može biti zadato dijagramom ili tablicom prelaza. Tako se matematički model životnog ciklusa biljaka (u vrlo uprošćenom vidu) može predstaviti grafom, prikazanim na sl. 3.5.

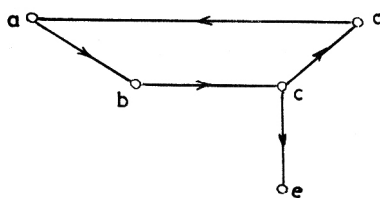
Ovde temena prikazuju stanje sistema, a strelice — prelaze iz jednih stanja u druga. Sa a je označeno stanje »klica«, b — »biljka«, c — »rascvetana biljka«. Iz stanja c sistem može preći ili u stanje e — »neoprašena biljka«, ili u stanje d — »oprašena biljka«, što će kao rezultat dati ponovo početno stanje a — »klica«.

Niz pretvaranja, prikazan grafom, datim na sl. 3.5, može biti zadat u obliku tabele 3.2, koja takođe može da služi kao matematički model razmatranog sistema.

Primitimo da iz stanja c sistem može da pređe ne samo u jedno, već u jedno od dva stanja: d i e , što pokazuje zavisnost prelaza od nekog faktora, u datom slučaju od toga da li je cvet oprašen. Po svoj prilici, teško je uzeti u obzir sve uslove koji dovode do oprašivanja, i predvideti oprašivanje za svaku pojedinu biljku. Ali, za veliki broj biljaka može se na osnovu statističkih podataka oceniti srednja frekvencija oprašivanja. Tada odnos broja



3.4. Skup karakteristika indukcionog električnog motora



Sl. 3.5. Prikaz životnog ciklusa jednogodišnje biljke u vidu grafa

oprašenih biljaka prema njihovom ukupnom, dovoljno velikom, broju daje veličinu p , koja karakteriše verovatnoću da svaka biljka bude oprášena. Veličina $1-p$ je, očigledno, verovatnoća da svaka biljka ne bude oprášena.

Za sisteme čje ponašanje zavisi od slučajnih faktora, nije dovoljno samo pokazati u koje stanje sistem prelazi, već treba dati i verovatnoću, tog ili drugog, prelaza.

Za razmatrani primer statistički model sistema će imati vid prikazan u tabeli 3.3.

Tabela 3.2

	a	b	c	d	e
a				←	
b	←				
c		←			
d			←		
e			←		←

Tabela 3.3

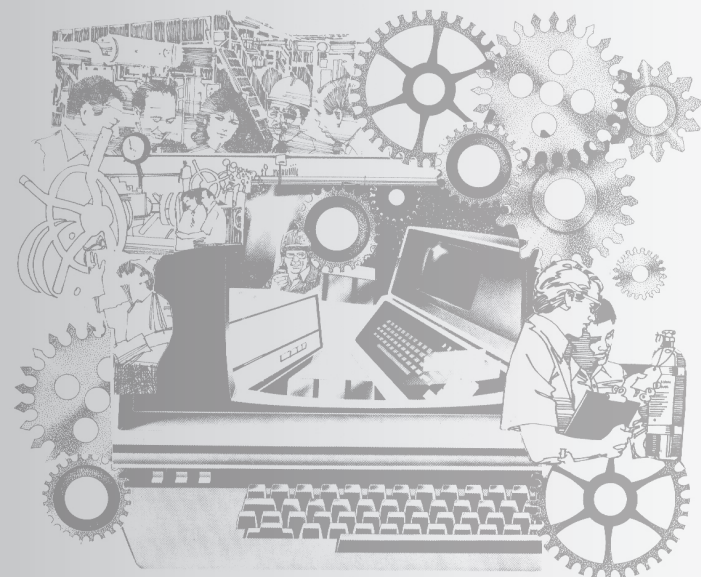
	a	b	c	d	e
a				1	
b	1				
c		1			
d			p		
e			1-p		1

Primer:

Koji se skupovi niže nabrojanih objekata mogu smatrati skupovima originala i modela, i u kom smislu: *a*) knjiga, *b*) sunčev sistem, *c*) oscilatorno kolo, *d*) atom, *e*) klatno sata, *f*) gramofonska ploča, *g*) memorija računara.

Odgovor:

- 1) Knjiga, gramofonska ploča, memorija računara.
- 2) Sunčev sistem, oscilatorno kolo, atom, klatno sata.



4. DINAMIČKI SISTEMI

Izučavajući ponašanje upravljanih sistema susrećemo se s potrebom proučavanja njihovog kretanja-promene njihovog stanja. Ali, promena stanja nekog sistema nije moguća bez procesa pretvaranja i prenosa energije i mase u njegovim sastavnim elementima. Tako je, na primer, promena temperature tela povezana s promenom njegove unutrašnje energije, a za promenu nivoa u rezervoaru potrebno je menjati količinu tečnosti koja se nalazi u njemu. Životinja, radi promene svog položaja u prostoru, u toku konačnog vremena, mora da ostvari brzinu različitu od nule, što sa svoje strane iziskuje akumulaciju rezervi kinetičke energije.

Ako bi promena stanja sistema mogla da nastane trenutno, to bi značilo da je zaliha energije ili materije u njemu za beskonačno malo vreme dobila konačan priraštaj. Ali za to bi bilo potrebno da jačina energetske struje ili toka materije ima beskonačno veliku vrednost, što je nemoguće. U skladu s ovim, stanje realnog sistema ne može da se menja trenutno, već se njegova promena dešava u toku vremena — kao rezultat određenog procesa, nazvanog *prelazni proces*.

Sistemi, čiji prelazi iz jednog stanja u drugo ne mogu da se izvrše trenutno, već nastaju kao rezultat prelaznog procesa, nazivaju se dinamički sistemi. Iz izloženog je jasno da su, strogo govoreći, svi realni sistemi dinamički sistemi. Ali, može u slučajevima kada je trajanje prelaznog procesa zanemarivo malo u odnosu na trajanje ispitivane pojave, a karakter odvijanja prelaznog procesa ne utiče bitno na ponašanje sistema, da se ne obraća pažnja na dinamičke osobine proučavanog sistema i da se približno smatra da uzroci izazivaju trenutne promene njegovih stanja.

Režimi dinamičkog sistema

Treba razlikovati tri karakteristična tipa ponašanja sistema—tri režima, u kojima može da se nalazi dinamički sistem: *ravnotežni*, *prelazni* i *periodični*.

Govorićemo da se sistem nalazi u ravnotežnom režimu, ako se njegovo stanje ne menja vremenom. Stanje u kome se nalazi sistem kada se nijedna od njegovih koordinata ne menja, nazvaćemo njegovim *ravnotežnim stanjem*. U prostoru stanja sistema njegova ravnotežna stanja biće prikazana nepokretnim tačkama. Jasno je da se sistem ne može u svim tačkama prostora stanja nalaziti u ravnotežnom režimu, već samo u nekim — *posebnim* tačkama, ili u posebnim skupovima tačaka. Razmotrimo kao primer kretanje broda oko svoje podužne horizontalne ose (sl. 4.1).

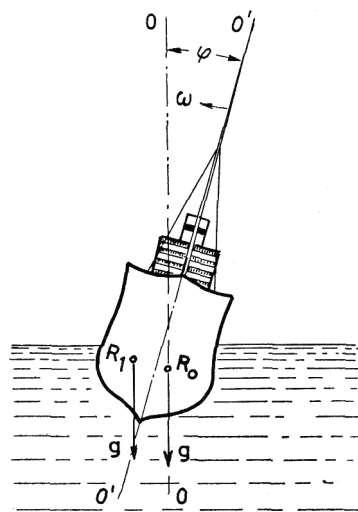
Dve koordinate određuju stanje posmatranog sistema: ugao φ , koji zaklapa vertikalna osa $O'-O'$ broda s pravcem prema centru zemlje $O-O$ (sl. 4.1) i ugaona brzina ω vertikalne ose broda.

Stanje ravnoteže u posmatranom sistemu može nastupiti samo pri takvom položaju broda, kada se njegovo težište R nalazi na osi $O-O$ (položaj R_0), a da je pri tome ugaona brzina $\omega=0$. Tačka a_0 (sl. 4.2), koja prikazuje ravnotežno stanje sistema, u ovom slučaju ima koordinate $\omega=0$; $\varphi=\varphi_0$. Ako se premesti deo tereta sa levog boka broda na desni, promeniće se ravnotežni položaj broda i biće prikazan, na primer, tačkom a_1 s koordinatama $\omega=0$; $\varphi=\varphi_0$. Ali za svaki raspored tereta sistem ima samo jedno ravnotežno stanje.

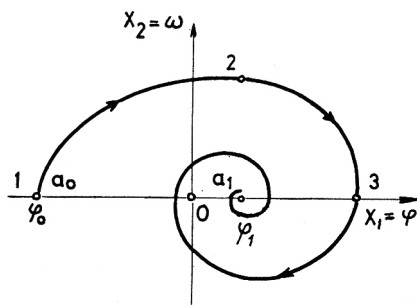
Pod prelaznim režimom podrazumevaćemo režim kretanja dinamičkog sistema iz nekog početnog stanja u bilo koji njegov *ustaljeni režim* — ravnotežni ili periodični.

Prelazni režim nastaje u sistemu pod uticajem promene spoljnog dejstva ili promene unutrašnjih svojstava sistema. Na primer, u posmatranom sistemu prelazni proces se može javiti pod uticajem vetra, koji menja nagib broda ili kao rezultat promene položaja težišta R (usled premeštanja tereta).

Ako se premeštanje tereta, koje izaziva promenu ravnotežnog stanja broda, izvrši toliko brzo, da njegov nagib ne uspeva za to vreme bitno da se promeni, može se smatrati da se kretanje odvija iz početnog položaja a_0 pri nepromenjenom položaju težišta broda $R-R_1$.

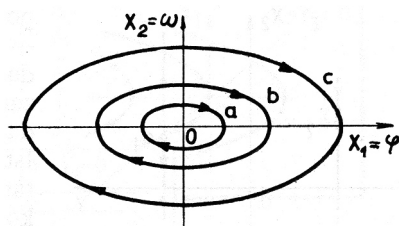


Sl. 4.1. Kretanje broda oko podužne ose



Sl. 4.2. Fazna trajektorija prigušenih oscilacija broda oko njegove podužne ose

Težina G , upravljena paralelno osi $O—O$ i priložena u težištu R , stvaraće obrtni moment prinuđujući brod da se rastućom brzinom zaokreće u pravcu novog položaja ravnoteže. Pri tome, reprezentativna tačka će se kretati po trajektoriji 1—2, dok se težište R ne nađe na osi $O—O$. Mada stanje sistema prikazano tačkom 2 odgovara ravnotežnoj vrednosti nagiba ($\varphi=\varphi_0$) i, mada se obrtni moment, stvoren delovanjem sile G , u toj tački anulira, brod se u tom položaju neće zaustaviti, pošto ugaona brzina ω nije jednaka nuli, već će produžiti da skreće od ravnotežnog položaja. Pri ovome se stvara obrtni moment koji deluje u suprotnom smeru od smera okretanja broda tako da njegova brzina opada, a reprezentativna tačka se kreće po trajektoriji 2—3. Na taj način, brod počinje da osciluje oko svog položaja ravnoteže φ_0 . Usled kočenja, koje izazivaju sile viskoznog trenja trupa broda o vodu, te oscilacije će se prigušivati i trajektorija reprezentativne tačke će imati vid spirale koja se zavija, kao što je pokazano na sl. 4.2.



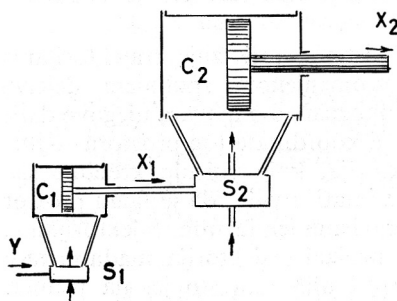
Sl. 4.3. Fazne trajektorije neprigušenih oscilacija broda oko njegove podužne ose

Režim, kada sistem u jednakim vremenskim razmacima dolazi u jedno isto stanje naziva se *periodični režim*. Samo u dva slučaja u posmatranom sistemu može da postoji periodični režim: pod uticajem talasanja vode (prinudni periodični režim) i u slučaju odsustva trenja o vodu (režim slobodnih neprigušenih oscilacija). Neprigušene slobodne oscilacije u ovakvom sistemu, razume se, treba shvatiti samo teorijski mogućim pri približnom ispitivanju procesa. Fazne trajektorije mogućih kretanja posmatranog sistema u periodičnom režimu pokazane su na sl. 4.3.

Fazni prostor

Efikasno proučavanje ponašanja dinamičkog sistema nije moguće u bilo kom prostoru njegovih stanja. Pri proizvoljnom izboru koordinata uključenih u prostor stanja dešava se da nije moguće predvideti kretanje sistema.

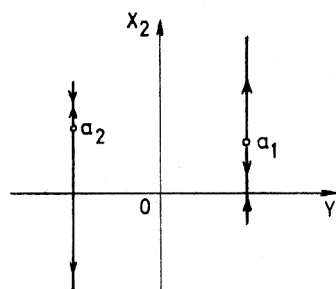
Neka, na primer, istraživač proučava ponašanje sistema hidrauličnog pogona s pojačavačem, pokazanog na sl. 4.4. On se sastoji iz razvodnika S_1 i S_2 i cilindara C_1 i C_2 , sjedinjenih, kao što je pokazano na šemi, tako da položaj Y razvodnika S_1 određuje brzinu pomeranja klipa u cilindru C_1 , a položaj razvodnika S_2 , koga pomera klip cilindra C_1 , određuje brzinu kretanja klipa u cilindru C_2 . Posmatraćemo kao izlaznu veličinu položaj X_2 klipa cilindra C_2 , a kao ulaznu veličinu — položaj Y razvodnika S_1 .



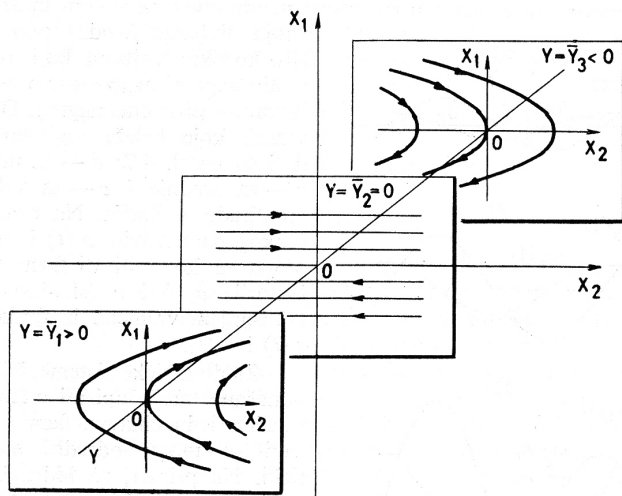
Sl. 4.4 Hidraulični pogon sa pojačavačem

Pretpostavimo da istraživač pokušava da objasni zakonitosti kretanja tog sistema proučavajući trajektorije kretanja reprezentativne tačke u ravni Y, X_2 (sl. 4.5). Tada se otkriva da u nizu slučajeva jednom istom položaju na ulazu Y i na izlazu X_2 sistema, prikazanog tačkom a_p , odgovaraju razne trajektorije kretanja reprezentativne tačke. Ponekad se iz tog položaja klip kreće udesno, ponekad — ulevo, a ponekad klip počinje da se kreće u jednom smeru, a zatim se sam zaustavlja i počinje da se kreće u suprotnom smeru. Ista takva slika se opaža pri kretanju iz početnog stanja a_2 i bilo kog drugog. Takva nepredvidljivost ponašanja sistema lišava istraživača mogućnosti da prouči njegova svojstva i svedoči o postojanju principijelne greške u prilazu proučavanoj pojavi. U datom slučaju spor se objašnjava time što istraživač nije uveo u razmatranje koordinatu X_1 koja karakteriše položaj razvodnika S_2 . U prostoru Y, X_1, X_2 svakoj određenoj vrednosti ulaza $Y = Y_j$ odgovara skup trajektorija koje se ne seku i koje jednoznačno određuju kretanje sistema iz bilo kog njegovog početnog stanja, kao što je pokazano na sl. 4.6.

Prostor u kome se kretanje sistema prikazuje trajektorijama koje se ne seku, tako da pri nepromenjenom spoljnjem dejstvu svakom početnom stanju sistema jednoznačno odgovara njegovo dalje ponašanje, naziva se *fazni prostor*, a koordinate tog prostora — *fazne koordinate*. Skup faznih trajektorija, koje opisuju kretanje sistema, naziva se *fazni portret*. Treba imati u vidu, da je fazni prostor bilo kog dinamičkog sistema potpuno ispunjen faznim trajektorijama, tj. kroz svaku tačku tog prostora prolazi trajektorija, mada se radi preglednosti pokazuju samo neke od njih kao što je, na primer, učinjeno na skupu faznih portreta datih na sl. 4.6. Ova slika pokazuje takođe, da, menjajući spoljna dejstva na sistem, možemo bitno promeniti njegov fazni portret. Nije teško uočiti da je opis bočnog ljuľanja broda, dat ranije, takođe izveden u faznom prostoru; trajektorija, pokazana na sl. 4.2 je fazna trajektorija, a skup trajektorija na sl. 4.3 — fazni portret sistema.



Sl. 4.5 Prostor ulaznih i izlaznih promenljivih hidrauličnih pogona



Sl. 4.6. Fazni prostor hidrauličnog pogona s pojačavačem

Broj dimenzija faznog prostora naziva se *red sistema*. Kretanje sistema pokazanih na sl. 4.1 i 4.4 jednoznačno se prikazuje u dvo-dimenzionalnom faznom prostoru, što dozvoljava da ih svrstamo u sisteme drugog reda.

Složeniji sistemi će pripadati sistemima višeg reda, a njihovi fazni prostori će biti višedimenzionalni. Tako na primer, ako se u hidraulični pogon pokazan na sl. 4.4 uključe dopunska kaskadna pojačanja, red sistema će se povisiti do vrednosti n , jednake broju cilindara u postrojenju.

Fazne trajektorije dinamičkih sistema mogu se konstruisati na osnovu eksperimentalnih podataka. Radi toga mogu se meriti vrednosti faznih koordinata u toku procesa kretanja sistema pri stalnim vrednostima ulaznih dejstava. Položaj reprezentativne tačke se određuje, pri tome, vrednostima koordinata u određenim momentima vremena. Ispitujući ove procese pri različitim stanjima sistema, može se naći skup faznih trajektorija i konstruisati, na taj način, fazni portret sistema.

Za dinamičke sisteme, čije se ponašanje može opisati odgovarajućim jednačinama, fazni portreti se mogu odrediti analitički. Na primer, za hidraulični pogon, koji je prethodno opisan, moguće je konstruisati skup faznih likova, datih na sl. 4.6, na sledeći način.

Označimo sa k koeficijent proporcionalnosti kojim su povezane brzina kretanja klipa i položaj upravljačkog razvodnika. Tada će se, pri stalnom položaju $\bar{Y}$ razvodnika S_p , položaj poluge njegovog klipa (koordinata X_1) menjati po zakonu

$$X_1(t) = X_{1p} + k \bar{Y} t \quad (4.1)$$

gde je $X_{1p} = X_1(0)$ — početna vrednost koordinata X_1 . Analogno ovome, za koordinatu X_2 imamo

$$X_2(t) = X_{2p} + k X_1 t \quad (4.2)$$

Eliminišući vreme t iz (4.1) i (4.2) dobijamo

$$X_2 = X_{2p} - \frac{X_{1p}}{\bar{Y}} X_1 + \frac{1}{\bar{Y}} X_1^2 \quad (4.3)$$

Izraz (4.3) određuje funkcionalnu vezu između koordinata X_1 i X_2 . U datom slučaju ta funkcija ima oblik parabole. Kao što se vidi iz (4.3), oblik fazne trajektorije zavisi od koordinata tačke, koja prikazuje početno stanje sistema (X_{1p} i X_{2p}), i od vrednosti ulaznog dejstva $\bar{Y}$. Koristeći izraz (4.3), mogu se konstruisati fazni likovi za slučajeve, date na sl. 4.6, i proveriti saglasnost te slike sa izrazom (4.3).

Stabilnost

Stabilnost je jedna od osnovnih osobina kretanja dinamičkih sistema. Kvalitetna po svojoj prirodi, ova osobina kretanja je intuitivno lako shvatljiva. Mi ćemo često reći „stabilna ekonomija“, „stabilna osoba“ itd., bez bojazni da nećemo biti shvaćeni. Izvestan sistem,

pod izvesnim datim okolnostima ponaša se (kreće se) na određen način. Drugim rečima, prelazi iz jednog stanja u drugo saglasno nekom određenom zakonu. Svesni da okolnosti, u kojima se system nalazi, nisu nepromenljive i da se na njih redovno ne može uticati, mi se pitamo: Da li male promene ovih okolnosti imaju mali efekat na određeno kretanje sistema, ili se pod ovim promenama kretanje sistema drastično razlikuje od predviđenog? U prvom slučaju imamo stabilno, a u drugom nestabilno kretanje.

Mada ideja stabilnosti intuitivno, na prvi pogled, izgleda prosta, postoji veliki broj preciznih matematičkih definicija stabilnosti, koje se jedva mogu razlikovati jedna od druge. Naravno, samo izvestan broj ovih definicija ima široku primenu u opisu ponašanja realnih, fizičkih sistema. Budući da smo principijelno zainteresovani za primenu dinamičkih sistema upravljanja, usredsredićemo se isključivo na one pojmove stabilnosti koji se koriste u ovim primenama.

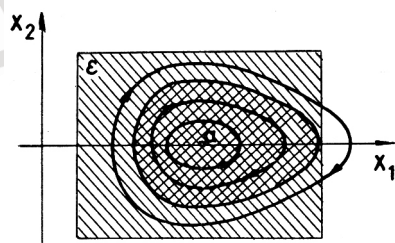
Stabilnost karakteriše jednu od najvažnijih osobina ponašanja sistema i predstavlja fundamentalan pojam, koji se koristi u fizici, biologiji, tehnici, ekonomiji, a i u kibernetici. Pojam stabilnosti se koristi za opisivanje *stalnosti* neke karakteristike ponašanja sistema, shvatajući je u veoma širokom smislu. To može biti stalnost stanja sistema (njegova nepromenjenost u toku vremena) ili stalnost nekog niza stanja, kroz koja prolazi sistem u procesu svog kretanja, ili stalnost broja ličnosti određenog tipa koje žive na zemlji i sl.

Tačnu i strogu definiciju pojma stabilnosti ravnotežnog stanja dinamičkog sistema dao je istaknuti ruski naučnik A. Ljapunov. Uzmimo da nepokretna tačka a predstavlja u faznom prostoru sistema njegovo ravnotežno stanje (sl. 4.7). *To ravnotežno stanje će biti stabilno po Ljapunovu, ako se za bilo koju zadatu oblast dopuštenih odstupanja od stanja ravnoteže (oblast ε) može naći takva oblast δ (koja obuhvata stanje ravnoteže), da trajektorija bilo kog kretanja započetog u oblasti δ nikada ne dostiže granice oblasti ε .*

Oblik faznih trajektorija, pokazan na sl. 4.7, obezbeđuje zadovoljenje tog uslova, što svedoči o stabilnosti stanja ravnoteže prikazanog tačkom a .

Nije teško proveriti da je i stanje sistema prikazano koordinatnim početkom na sl. 4.3 stabilno u navedenom smislu, mada se fazne trajektorije ne koncentrišu u toj tački.

Međutim, položaj ravnoteže sistema nije uvek stabilan. Razmotrimo, na primer, režim rada hidrauličnog pogona pri nultom pomeranju ulaznog razvodnika ($Y = \bar{Y} = 0$), pokazanog na sl. 4.8. Bilo koja tačka na osi X_1 prikazivaće ravnotežno stanje sistema. Pri srednjem položaju razvodnika S_1 i S_2 brzine klipova su jednake nuli i stanje sistema ne može da se menja. I ono stvarno neće da se menja ako pretpostavimo da su uslovi $Y = 0$ i $X_1 = 0$ idealno ispunjeni. Ali, u stvarnosti se svaki sistem nalazi pod uticajem unutrašnjih i spoljašnjih poremećajnih dejstava. Ma kako mala bila ta dejstva, ona ipak izazivaju promenu stanja sistema, a kao rezultat toga je lutanje reprezentativne tačke oko svog srednjeg položaja u nekoj oblasti β .



Sl. 4.7. Uz definiciju stabilnosti po Ljapunovu

Pošto se, pri tome, reprezentativna tačka neizbežno nalazi iznad ili ispod ose X_1 , trajektorija kretanja reprezentativne tačke ranije ili kasnije seče granicu oblasti ε , kao što je pokazano na sl. 4.8, ma kako da izaberemo granice oblasti δ . Iz ovoga sledi da nijedno od ravnotežnih stanja tog sistema nije stabilno.

Razmotrimo sada primer uspostavljanja tržišne cene, u kome postoji i stabilno i nestabilno stanje ravnoteže. Neka zavisnosti potražnje D i ponude S neke robe od cene P na tržištu imaju oblik pokazan na sl. 4.9, a brzina d promene cena neka bude direktno proporcionalna razlici između potražnje i ponude:

$$d=k(D-S) \quad (4.4)$$

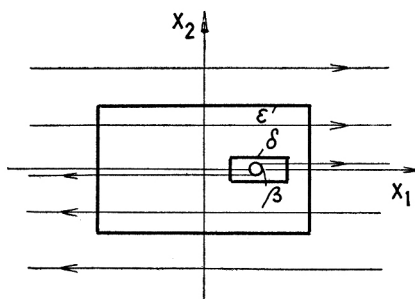
Ovde je k — koeficijent koji pokazuje koliko raste cena robe u jedinici vremena, ako je razlika između potražnje i ponude jednaka jedinici. Saglasno smislu modela koeficijent $k>0$.

Uzroci smanjenja potražnje i povećanja ponude pri povišenju cena su poznati. Povećanje ponude pri sniženju cena niže od P_k može se desiti u nekim posebnim slučajevima, na primer, prilikom prelaza na metode masovne proizvodnje robe pri sniženju cena i porasta potražnje. Iz sl. 4.9 vidi se da posmatrani sistem ima dva ravnotežna stanja a_1 i a_2 , pošto je u ovim tačkama potražnja jednaka ponudi i cena robe se, u skladu sa (4.4), ne menja ($d=0$).

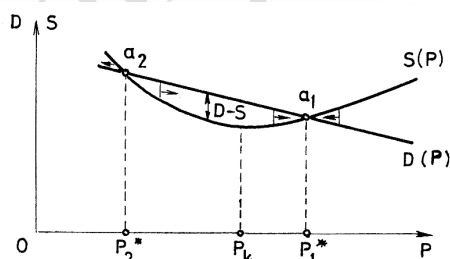
Radi objašnjenja stabilnosti stanja ravnoteže, pokažimo kako će se menjati cena posle svog slučajnog, malog odstupanja od ravnotežnih vrednosti P_1^* i P_2^* . Na sl. 4.9 vidi se da u tački a_1 odstupanju cene P od vrednosti P_1^* odgovara takva razlika $D-S$, da izaziva promenu cene uspostavljajući narušenu ravnotežu, i da tačka a_1 prikazuje stanje stabilne ravnoteže sistema. U tački a_2 , suprotno tome, bilo koje odstupanje cene od P_2^* izaziva njenu dalju promenu u istom smeru, te je stanje sistema u ovoj tački nestabilno.

Ciklusi

Pojam »stabilnosti« je primenljiv ne samo za ocenu vida *ravnotežnog stanja* sistema, već je veoma važan i za ocenu karaktera *kretanja* sistema. Tako na primer, veliki značaj ima rasvetljavanje pitanja stabilnosti cikličnih kretanja (kretanja po nekoj zatvorenoj trajektoriji u prostoru stanja) i, posebno, periodičnih kretanja.



Sl. 4.8. Nestabilna ravnoteža dinamičkog sistema



Sl. 4.9. Zavisnosti potražnje D i ponude S od cene robe P

Razmotrimo, kao primer, frikciono klatno, pokazano na sl. 4.10, koje se razlikuje od običnog klatna po tome, što se osovinu, o koju je ono obešeno, obrće konstantnom brzinom ω_0 . Pri oscilovanju klatna ugaona brzina ω njegovog naglavka menjaće se periodično i zato će s vremena na vreme da se poklapa po smeru sa ugaonom brzinom ω_0 osovine. U toku intervala τ_1 , zaostajanja naglavka za pokretnom osovinom ($\omega < \omega_0$), sile trenja između njih će ubrzavati klatno, a u ostalom delu perioda τ_2 — usporavati njegovo kretanje.

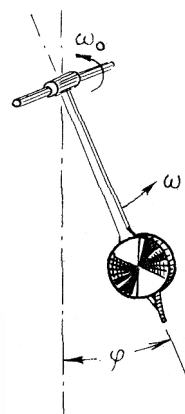
Pri tome, rad sile trenja u toku intervala τ_1 , će povećati rezervu energije klatna za veličinu ΔE_1 u toku jedne oscilacije, a u toku intervala τ_2 — smanjiti rezervu energije za veličinu ΔE_2 . Kao što pokazuju proračuni, zavisnost priraštaja energije ΔE_1 i ΔE_2 od totalne amplitude A oscilacija ima karakter pokazan na sl. 4.11. Nagomilavanje energije u klatnu približno je proporcionalno prvom stepenu totalne amplitude, a rasipanje — proporcionalno kvadratu totalne amplitude.

Jasno je da akumulirana energija u klatnu, a u skladu s tim, i amplituda oscilacija mogu ostati nepromenjene samo u slučaju kada postoji ravnoteža između dobijene i utrošene energije, tj. $\Delta E_1 = \Delta E_2$. Takav uravnoteženi režim može se ostvariti pri totalnoj amplitudi A^* . Oscilovanje klatna s totalnom amplitudom A^* prikazano je na sl. 4.12 zatvorenim krivom 1 u ravni (φ, ω) , gde je φ — skretanje klatna od vertikalne ose.

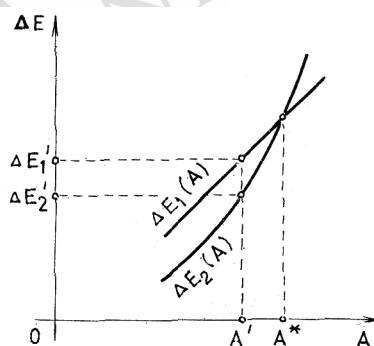
Razjasnimo sada da li će se uspostaviti kretanje po trajektoriji 1 posle slučajnih odstupanja kretanja od te trajektorije.

Pretpostavimo da se usled nekog uzroka rezerva energije u klatnu smanjila i da se reprezentativna tačka našla unutar oblasti ograničene trajektorijom 1. Tada iz dijagrama na sl. 4.11 nalazimo da je $\Delta E_1' > \Delta E_2'$ i da će se rezerva energije u toku svake oscilacije povećavati, a u skladu s tim će i totalna amplituda rasti. Pri tome će se reprezentativna tačka kretati po trajektoriji 2, približavajući se vremenom polaznoj trajektoriji 1. Analogna razmišljanja o slučaju kada početna rezerva energije prelazi svoju uravnoteženu vrednost, pokazuju da će se kretanje odvijati po trajektoriji 3 približavajući se takođe trajektoriji 1.

Ova razmatranja omogućuju da se zaključi da trajektorija 1 prikazuje *stabilne oscilacije sistema*, koje se uspostavljaju nezavisno od početnog stanja u kome se on nalazio. Amplituda i učestanost oscilacija ne zavise od početnih uslova. Takve oscilacije se nazivaju *sop-*



Sl. 4.10. Frikciono klatno



Sl. 4.11. Zavisnost promene energije klatna od totalne amplitude oscilacija A

stvene oscilacije. One se razlikuju od prinudnih oscilacija po tome što je spoljnje dejstvo neperiodičnog karaktera. Tako se u našem primeru osovina obrće konstantnom brzinom s kojom je posredno povezana učestanost oscilacija.

Trajektorija koja u faznom prostoru prikazuje stabilne oscilacije naziva se *stabilan graničan krug*.

Zatvorena trajektorija, međutim, nije uvek stabilni granični krug. Tako je, na primer, granični krug, prikazan na sl. 4.13, nestabilan; Razmotrimo drugi primer, cikličnog kretanja — najprostiji sistem upravljanja uličnim saobraćajem na raskrsnici. Označimo sa X_1 kolonu, koja očekuje slobodan prolaz, pri čemu je $X_1=1$, ako postoji povorka koja čeka dozvolu za prolaženje, $X_1=0$ u suprotnom slučaju. Sa X_2 označimo odobravanje prolaza — (zeleni signal na semaforu). Ovde je $X_2=1$ ako je slobodan prolaz za saobraćaj, $X_2=0$ u suprotnom slučaju.

Stanje takvog sistema se određuje položajem reprezentativne tačke u ravni (X_1, X_2) , koja može da zauzme jedno od četiri položaja prikazana na sl. 4.14. Prikažimo sistem tako da se prelaz iz nekog njegovog i -tog stanja (koje karakterišu vrednosti koordinata X_{1i}, X_{2i}) u $(i+1)$ -vo stanje $(X_{1(i+1)}, X_{2(i+1)})$ ostvaruje dejstvom operatora P na operand X_{1i}, X_{2i}

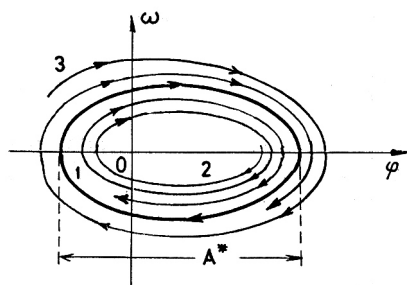
$$\{X_{1(i+1)}, X_{2(i+1)}\} = \{P\} \{X_{1i}, X_{2i}\},$$

gde operator P ostvaruje tablicu prelaza (tabl. 4.1).

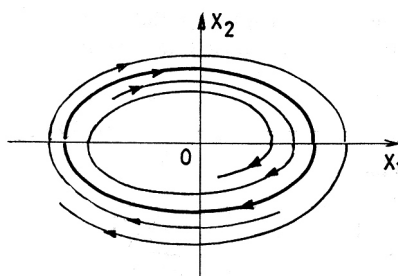
Tabela 4.1

X_{1i}	0	1	1	0
X_{2i}	0	0	1	1
$X_{1(i+1)}$	1	1	0	0
$X_{2(i+1)}$	0	1	1	0

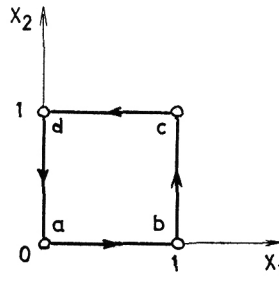
Lako je uočiti da će tada sistem uzastopno prelaziti u svako od svoja četiri moguća stanja, ostvarujući ciklus $a-b-c-d-a$, i da će trajektorija kretanja njegove reprezentativne tačke imati oblik kvadrata, kao što je pokazano na sl. 4.14.



Sl. 4.12. Fazni portret frikcionog klatna



Sl. 4.13. Nestabilni granični krug



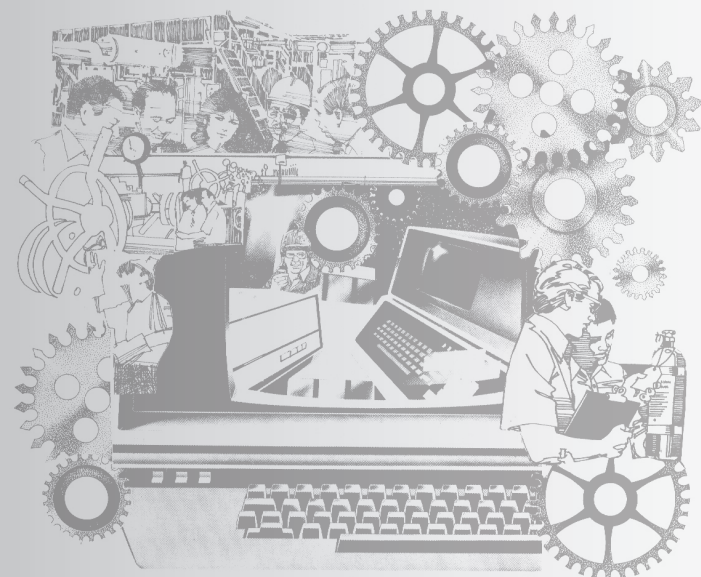
Sl. 4.14. Dijagram prelaza za sistem regulisanja uličnog saobraćaja

Primer:

Obično se raspored kretanja autobusa u gradskom prevozu sastavlja tako da oni treba da stižu na stanice u jednakim vremenskim intervalima. Međutim, usled velikog broja putnika na nekoj stanici autobus se zadržava duže od predviđenog vremena, a sledeći, koji ide iza njega na kraćem rastojanju nego obično, prima manje putnika i zadržava se manje nego obično. Da li je nastala neravnomernost kretanja stabilna ili nestabilna?

Odgovor:

Na žalost, takva neravnomernost je stabilna i to je stalan uzrok žalbi putnika i neprijatnosti za organizatore saobraćaja.



5. UVOD U TEORIJU INFORMACIJA

Među pojedinim elementima nekog sistema i među različitim sistemima postoje veze; a preko njih oni međusobno utiču jedni na druge. Te veze mogu da se sastoje u razmeni energije ili materije među objektima koji se nalaze u uzajamnom delovanju. Međutim, veze mogu biti i takve da u prvi plan dolazi *informacioni* sadržaj veza, tj. podaci koje dobija dati objekt o stanjima drugih objekata. Pri tome, materijalna forma, kojom su izražene te informacije, ima drugorazredni značaj.

Takve informacione veze se ostvaruju *signalima* koji kruže sistemima. Signale možemo prenositi na rastojanje i time ostvariti vezu među objektima odvojenim u prostoru.

Pamćenje/memorisiranje signala omogućuje da se tokom vremena ostvari njihov prenos, što znači, međusobno povezivanje vremenski odeljenih objekata.

Kodiranje

Do sada smo posmatrali sisteme čiji prostori stanja mogu biti kako diskretni tako i neprekidni. U diskretnom prostoru stanja dopuštali smo mogućnost postojanja stanja koja su prikazana samo *odvojenim* tačkama, kao što smo istovremeno u neprekidnom prostoru dozvoljavali mogućnost postojanja stanja, prikazanog *bilo kojom* tačkom u nekoj dopuštenoj oblasti prostora stanja.

Ako želimo da organizujemo kontrolu stanja sistema, potrebno je da se na neki način ocenjuju vrednosti njegovih koordinata. Pri tome se pokazuje da nijedan način posmatranja ne može pružiti posmatraču opsolutno tačne podatke o vrednostima koordinata sistema. Kako vizuelnom opažanju, tako i bilo kom merenju svojstvena je određena ograničena moć utvrđivanja. Tako na primer, ako bismo merili lenjirom dužine dveju osovina, koje se razlikuju za manje od širine crte mernog podeoka, dobili bismo jednake vrednosti dužina za obe osovine. Greške merenja, smetnje i promene merenih veličina ograničavaju tačnost svakog merenja. To znači da uvek postoje takve, dovoljno bliske, ali nejednake vrednosti svake koordinate, čije se razlike ne mogu uočiti datim načinom posmatranja.

Stoga, čak i sistemi koji se matematički opisuju kao neprekidni predstavljaju, pri nekom konkretnom načinu kontrole, diskretne sisteme s konačnim brojem mogućih stanja.

Posmatraćemo *mnoštva stanja* sistema koja uključuju sva njihova različita stanja. Neka nas, na primer, interesuje stanje čoveka, koje karakteriše vrednost temperature njegovog tela. Ako isključimo slučaj veštačke hipotermije, može se smatrati da temperatura živog čoveka ne izlazi van granica intervala od 34°C do 42°C. Jasno je da temperatura može imati bilo koju vrednost u tom intervalu. Međutim, u praksi merimo temperaturu običnim medicinskim termometrom, čija moć utvrđivanja iznosi 0,1 °C.

Mnoštvo stanja u našem slučaju sastoji se samo od 81 elementa (elementi su određene vrednosti temperature), zapravo: 1) 34,0°C; 2) 34,1°C; 3) 34,2°C; ... ; 81) 42°C. Ako se stanje sistema predstavlja vektorom, čije komponente mogu nezavisno jedne od drugih da primaju: $X_1 = r_1$ vrednosti, $X_2 = r_2$ vrednosti, ... $X_n = r_n$ vrednosti, tada je broj elemenata (tj. svih mogućih kombinacija vrednosti $X_1, \dots, X_n$) koji ulaze u mnoštvo stanja sistema, jednak

$$N = r_1 \cdot r_2 \cdot \dots \cdot r_n.$$

Nazivaćemo *događajem* stanje sistema u određenom momentu. Ako se u bilo kom momentu t sistem može nalaziti u bilo kom svom stanju iz skupa X , tada će (X, t) predstavljati mnoštvo mogućih događaja za svaki trenutak.

Može se uslovno svakom stanju sistema staviti u saglasnost određena vrednost, ili niz vrednosti neke fizičke veličine. Ovim veličinama može se ostvariti prenos saopštenja (podataka o događajima) od jednog objekta drugom objektu.

Fizički proces koji predstavlja materijalno ovaploćenje saopštenja, naziva se signal. Sistem ili sredina u kojoj se ostvaruje prenos signala, naziva se kanal veze.

Pošto svakom stanju sistema X odgovara određeno saopštenje X_c , mnoštvu mogućih događaja odgovara *mnoštvo saopštenja*, koja se prenose pomoću signala. Tako se na primer podaci o stanju atmosfere u određenoj tački prenose u meteorološki centar komunikacionim kanalom pomoću određenog niza električnih signala.

Formiranje saopštenja može se smatrati pretvaranjem stanja sistema $X = \{X_1, X_2, \dots, X_n\}$ u X_c — jedno iz mnoštva mogućih saopštenja $X_c = \{X_c^1, X_c^2, \dots, X_c^n\}$, primenom nekog operatora P :

$$X_c^i = \{P\}\{X_i\}.$$

Operator P pretvaranja nekog operanda u njegov lik — saopštenje naziva se *kôd*, a operacija pretvaranja pomoću kôda — *kodiranje*. Kao operand takvog pretvaranja može se tretirati ne samo stanje X sistema ili događaj (X, t) , već i saopštenje X_j . Operacija prekodiranja saopštenja je neophodna u slučajevima, kada je, radi pogodnosti prenosa saopštenja ili s ciljem očuvanja tajnosti prenosa, potrebno saopštenje X_c^i , kodirano na jedan način, pretvoriti u saopštenje X_c^j , kodirano na drugi način. Takva pretvaranja saopštenja mogu se predstaviti kao uzastopno delovanje operatora $P_1, P_2, \dots, P_i$ na X prema šemi.

$$X_c^1 = P_1 X, X_c^2 = P_2 X_c^1, X_c^l = P_l X_c^{(l-1)}.$$

Za reprodukovanje predatog saopštenja, čak i u slučajevima kada je ono bilo podvrgnuto višestrukom prekodiranju, nije potrebno naizmenično obnavljati sve usputne kodove. Dovoljno je izvršiti nad događajem X_c^l jednu operaciju pretvaranja

$$X = P^{-1} X_c^l$$

gde je

$$P^{-1} = P_1^{-1} P_2^{-1} \dots P_l^{-1}.$$

Operatori označeni sa »—1« ostvaruju pretvaranje obratno onom koje realizuju operatori bez ovog znaka. Operatori P_i i P_c su međusobno povezani zavisnošću

$$P_i^{-1} P_i X = X.$$

Kao primer složenog sistema formiranja saopštenja, njegovog kodiranja i dekodiranja, može se navesti televizijski sistem prenosa slika. U njemu se ostvaruje sledeći lanac pretvaranja: stanje sistema (tj. raspodela svetlosnih tonova predate slike) — optički prikaz na ekranu — električni signal u vidu promenljive vrednosti struje u mreži, stvoren elektronskim zrakom koji naizmenično prelazi ekran, — radiosignal u obliku elektromagnetskih talasa promenljive učestanosti, emitovanih predajnikom — prikaz na ekranu prijemnikai koji se dobija kao rezultat obratnog pretvaranja (dekodiranja) saopštenja prispelog u televizor.

Informacija

Iz prethodnog izlaganja vidi se da signal može sadržavati neke podatke o stanju sistema, o događaju ili procesu. Ipak još nije jasno da li se može, i ako može, kako da se kvantitativno ocene podaci sadržani u signalu, da se oceni količina informacija koju nosi signal. Takva ocena je neophodna za proračune propusne moći kanala namenjenih prenosu signala, za određivanje karakteristika uređaja koji pretvaraju signale, kao i svrsishodnih načina kodiranja i sl.

Zadatak kvantitativne ocene informacionih procesa ne može se smatrati potpuno rešenim. Do sada se uspelo da se za izvesnu dovoljno široku klasu zadataka povezanih sa prenosom informacija od jednog objekta ka drugom, zapisivanjem i čuvanjem informacija, stvori iscrpna teorija koja daje kvantitativne karakteristike saopštenja. Ovo je postignuto uglavnom zahvaljujući osnovnim radovima američkog naučnika Kloda Elvuda Šenona.

Ako se apstrahuje smisaoni sadržaj informacije, njena vrednost za primaoca i forma u kojoj je izražena, tada se svako saopštenje može posmatrati kao podatak o određenom događaju (X, t_r) , koji sadrži obaveštenja o tome u kom se stanju, iz mnoštva mogućih stanja, nalazio sistem S u momentu t_r .

Razmotrimo sada detaljnije takozvana diskretna saopštenja. *Diskretnim saopštenjem* naziva se niz simbola uzetih iz nekog skupa simbola — *azbuke*. Svaki pojedini simbol naziva se *slovo* azbuke. Kao primer diskretnog saopštenja može poslužiti običan tekst na

nekom jeziku. Uočite da je azbuka takvog saopštenja u stvari šira od obične azbuke tog jezika: osim slova u nju treba uključiti prazna mesta između slova (vrlo važan simbol!) i interpunkciju. Konačan niz simbola uzetih iz neke azbuke naziva se *reč u datoj azbuci*. U primeru s tekstom na nekom jeziku reči su, osim običnih reči, isto tako i bilo koje kombinacije slova, razmaka i interpunkcije.

Korišćenje diskretnih saopštenja dozvoljava prenošenje podataka o stanju izabranom iz ma kako velikog broja mogućih stanja, korišćenjem malog broja različitih simbola koji pripadaju azbuci (broj tih simbola naziva se *osnova koda*). Tako se pokazuje da se bilo koje i bilo kako složeno saopštenje može preneti nizom, sastavljenim samo od dva različita simbola, na primer od simbola 0 i 1, kojima mogu odgovarati 0 — odsustvo signala, 1 — postojanje signala. U stvari, ako se sistem može nalaziti u jednom od N različitih stanja X_p čije je mnoštvo $X_1, X_2, \dots, X_N$ poznato primaocu saopštenja, tada je za prenos podataka o stanju sistema dovoljno dati broj i ($i = 1, 2, \dots, N$) stanja u kome se on nalazi. Taj broj predstavlja reč u azbuci čija su slova cifre.

Kao što je poznato, količina različitih cifara od kojih je sastavljen broj zavisi od sistema brojanja. U dvočlanom sistemu brojanja svaki broj se izražava kombinacijom nula i jedinica, koje čine razrede tog broja. Tada se svaki broj i može napisati u obliku

$$a_m a_{m-1} \dots a_1, \quad (5.1)$$

gde svako a može primiti samo dve vrednosti: 0 ili 1, a oznaka (5.1) označava

$$i = a_m 2^{m-1} + a_{m-1} 2^{m-2} + \dots + a_1.$$

Ako je, na primer, broj i u decimalnom obeležavanju $i = 27$,
tada će u dvočlanom označavanju imati oblik $i = 11011$

$$(1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2 + 1) \quad (5.2)$$

Reč (5.2) može se preneti nizom signala koji predstavljaju impulse struje i pauze, u vidu serije pokazane na sl. 5.1.

Na taj način, saopštenje o nekom događaju može se napisati u obliku reči azbuke od dva slova. Različitih dvočlanih nizova dužine m ima 2^m , što je lako proveriti, (pošto svaki simbol može primiti dve vrednosti nezavisno od drugih). Na taj način, pomoću dvočlanog niza dužine m može se preneti saopštenje o događaju izabranom iz N mogućih događaja, gde je $N = 2^m$, ili, drukčije, $m = \log_2 N$. Ako bismo prenosili isto saopštenje, ne dvočlanim, već, na primer, decimalnim kodom (tj. koristili bismo deset različitih simbola, recimo, arapske cifre), bio bi nam potreban niz dužine $m' = \log_{10} N$. Tada je $m' = m \log_{10} 2$, tj. m' se razlikuje od m konstantnim množiteljem, te ne zavisi od N . Uopšte, ma kakva bila osnova koda, dužina niza potrebna za prenos nekog saopštenja je proporcionalna logaritmu broja mogućih saopštenja. Opravdano je smatrati da je maksimalna količina informacija H_{max} , koja se sadrži u saopštenju, proporcionalna njegovoj dužini, tj.

$$H_{\max} \approx m \approx \log N.$$

Izbor koeficijenta proporcionalnosti svodi se na izbor osnove logaritma i označava izbor jedinice količine informacija. Najčešće se bira logaritam s osnovom dva. Tada se za jedinicu usvaja količina informacija koja se sadrži u jednom dvočlanom razredu, tj. u izboru jednog od dva moguća saopštenja. Takva jedinica informacije naziva se *bit* (engl. binary digit, tj. »dvočlani razred«).

Pri tome je

$$H_{\max} = \log_2 N. \quad (5.3)$$

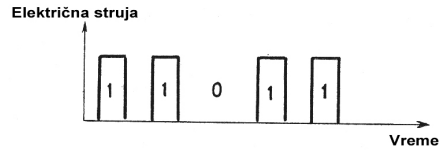
Napomena: U našoj stručnoj literaturi upotrebljava se termin »binaran« (od latinskog *bini* — po dva, *binarius* — koji sadrži dva) — dvojni, od dva dela od dva člana.

Ovu meru maksimalne količine informacija, koja može da se sadrži u saopštenju, predložio je, još pre stvaranja teorije informacije (1928. godine), američki naučnik L. Hartli. Ona poseduje dve važne osobine: **aditivna je i monotono raste s porastom N** . Svojstvo aditivnosti označava sledeće: neka se saopštenje a bira među N_1 mogućih saopštenja, a nezavisno od njega saopštenje b — iz N_2 mogućih saopštenja. Koliko se informacija sadrži u složenom saopštenju koje se sastoji iz saopštenja a i saopštenja b ? Očigledno je broj svih mogućih takvih saopštenja jednak $N_1 N_2$. U skladu s tim,

$$H_{\max}(N_1 \cdot N_2) = \log_2 N_1 N_2 = \log_2 N_1 + \log_2 N_2 = H_{\max}(N_1) + H_{\max}(N_2),$$

tj. količina informacija u skupu od dvaju nezavisnih saopštenja jednaka je sumi količnika informacija u svakom od tih saopštenja. To se slaže s našim intuitivnim predstavama.

Veličina $H_{\max}$ daje gornju granicu količine informacija, koje se mogu sadržavati u saopštenju. Stvarna količina informacija zavisi ne samo od broja mogućih saopštenja, već i od njegovih verovatnoća. Na primer, količina informacija u saopštenju o tome, da se Vašim poznanicima rodilo dete — muško ili žensko — znatno je veća nego u saopštenju o tome, da li su se rodile trojke ili ne. Kako je slučaj rađanja trojki veoma redak (ukoliko se ne radi o veštačkoj oplodnji), skoro sigurno možete predvideti da one neće biti. Ovde je umesno podvući, da se ne govori o količini informacija u određenom saopštenju (na primer, u saopštenju »rođene su trojke«), već se govori o *srednjoj* količini informacija koja se sadrži u saopštenju odabranom sa zadatom verovatnoćom iz mnoštva mogućih saopštenja. Pošto se neočekivano saopštenje (»rođene su trojke«) sreće vrlo retko, skoro uvek dobijate »banalno« saopštenje (»trojke nisu rođene«), te će srednja količina informacija biti mala.



Sl. 5.1. Signal (niz strujnih impulsa), koji odgovara binarnom saopštenju 11011

U graničnom slučaju kada verovatnoće svih događaja, osim jednog, postaju jednake nuli, i količina informacija jednaka je nuli —jer je u tom slučaju unapred poznato kakvo će se saopštenje dobiti. To znači, da ono ne nosi ništa novo. Nasuprot tome, kada su apriori, tj. pre dobijanja saopštenja, sva moguća saopštenja jednako verovatna, količina informacija mora biti maksimalna. Ove intuitivne predstave pomažu da se shvati smisao kvantitativne mere informacija koju je uveo K. Šenon 1947. godine.

Zamislamo da se izvodi opit čiji rezultat nije unapred poznat. Određeno je samo mnoštvo mogućih rezultata $X_1, X_2, \dots, X_n$ i verovatnoća rezultata $p(x_1), p(x_2), \dots, p(x_n)^*$. (Koristeći terminologiju teorije verovatnoće može se reći da se govori o *slučajnoj veličini* X , koja prima vrednosti $x_1, x_2, \dots, x_n$ s verovatnoćama $p(x_1), \dots, p(x_n)$.)

Količina informacija (u bitima) u saopštenju o rezultatu takvog eksperimenta, saglasno Šenonu, jednaka je

$$H(X) = - \sum_{i=1}^{i=N} p(x_i) \log_2 p(x_i) \quad (5.4)$$

Ako su svi rezultati opita jednako verovatni tada je $p(x_i) = 1/N$ za svako x_i i

$$H = \log_2 N = H_{\max}.$$

Ako je $p(x_k)=1$ za neki rezultat x_k , a za ostale ishode $p(x_i)=0$, ($i \neq k$), tada je $H=0$. (Smatra se da je veličina $0 \log 0$ jednaka nuli).

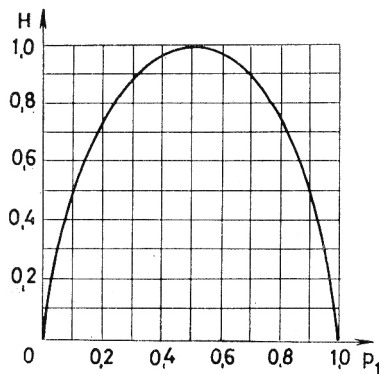
U ostalim slučajevima važi nejednačina

$$0 < H < \log_2 N.$$

Veličina H se naziva *entropija* slučajnog opita (slučajne veličine) X . Ona je mera prvobitne neodređenosti ishoda tog opita, mera statističke raznovrsnosti njegovih rezultata. Saopštenje o ishodu opita potpuno odstranjuje, »ukida« tu neodređenost i time daje količinu informacija jednaku H .

U teorijskim istraživanjima podesno je koristiti, ne logaritme s bazom 2, već prirodne logaritme. Odgovarajuća jedinica informacije naziva se prirodna jedinica (skraćeno »nit« ili »nat«). Entropija izražena u prirodnim jedinicama jednaka je

$$H(X) = - \sum_{i=1}^{i=N} p(x_i) \ln p(x_i). \quad (5.5)$$



Sl. 5.2. Entropija opita sa dva moguća ishoda: p_1 — verovatnoća jednog od ishoda

Izraz (5.5) poklapa se s izrazom za entropiju u statističkoj fizici (ako se pod ishodima $x_1, \dots, x_n$ podrazumevaju različita stanja fizičkog sistema). Ovo podudaranje nije samo formalno. Još početkom 20-tog veka veliki nemački fizičar A. Bolcman je napisao: »*Entropija je mera nedostajanja informacije o stanju fizičkog sistema*«. Danas je utvrđena duboka veza između teorije informacija i statističke fizike, koja dozvoljava izgradnju teorije informacija kao fizikalne teorije.

Na sl. 5.2 pokazano je kako se menja entropija opita s dvama mogućim ishodima x_1 i x_2 u zavisnosti od verovatnoće p_1 rezultata x_1 (verovatnoća ishoda x_2 pri tome je, očigledno, jednaka $p_2 = 1 - p_1$).

Prenos signala

Prenos signala se uvek ostvaruje kroz neki kanal veze: telefonsku liniju, radiokanal, kroz vazduh, magnetski način zapisivanja, itd. U kanale veza obično se učvršćuju i uređaji za kodiranje, prekodiranje i dekodiranje. Kanal veze se može posmatrati kao neki sistem na čiji se ulaz dovodi emitovani signal Y , a na izlazu se dobija prijemni signal, što je pokazano na sl. 5.3. Kao i svaki drugi sistem, kanal veze je podvrgnut unutrašnjim i spoljnim poremećajima M . Izvori tih poremećaja su slučajne spoljne i unutrašnje smetnje, a ponekad i neispravnosti elemenata koji čine kanal.

Pod uticajem poremećaja, signali, koji se prenose kroz kanal, ponekad bivaju izobličeni, ili uopšte ne dolaze do primaoca. Pretpostavimo, na primer, da primete saopštenje koje se Morzeovom azbukom radiotelegrafski prenosi u uslovima postojanja jakih atmosferskih smetnji. Tada će neki impulsi signala (tačka i crtica) »tonuti« u šumovima i, obratno, impulse šuma možete pogrešno shvatiti kao signale. Zato pri postojanju smetnji izlazni signal X ne predstavlja uzajamno jednoznačnu funkciju ulaznog sistema Y , već je povezan s njim samo verovatnoćama, statističkim zavisnostima. Postavlja se pitanje, kako kvantitativno oceniti informaciju, sadržanu u poznavanju vrednosti jedne slučajne veličine (izlaznog signala X), o vrednosti koju prima druga slučajna veličina (ulazni signal Y). Takvu, najopštiju meru količine informacija dao je i Šenon. Da bismo došli do tog pojma upoznajmo se prethodno s pojmom uslovne verovatnoće.

Razmotrimo kao primer proizvodnju delova iz sirovina I i II klase. Neka se od materijala I klase izrađuje 70% delova, a od II klase 30%. Na svakih 100 delova, izrađenih iz sirovine I klase, 83 su upotrebljiva, a iz sirovine II klase, na svakih 100 delova 63 su upotrebljiva.

Verovatnoću p događaja A , koji se sastoji u tome da slučajno izabrani deo bude upotrebljiv (označimo je sa $p(A)$), lako je izračunati kao srednju količinu upotrebljivih delova u čitavoj proizvodnji

$$p(A) = \frac{83}{100} \cdot \frac{70}{100} + \frac{63}{100} \cdot \frac{30}{100} = 0,77.$$

Pretpostavimo sada da nam je poznato da je deo izrađen iz sirovine I klase. Tada je verovatnoća toga da deo bude upotrebljiv, jednaka 0,83. Očigledno je da dodavanje uslova (u datom slučaju podatak o klasi sirovine) menja verovatnoću događaja. Označavajući sa B uslov

od koga zavisi događaj A , a verovatnoću događaja A pri uslovu B — *uslovnu verovatnoću* tog događaja — sa $p(A/B)$, možemo za razmatrani primer napisati

$$p(A/B) = 0,83.$$

Uopšte, ako slučajna veličina Y uzima vrednosti $y_1, y_2, \dots, y_N$ a slučajna veličina X — vrednosti $x_1, x_2, \dots, x_M$, tada se *uslovnom verovatnoćom* $p(y_i/x_j)$ naziva verovatnoća toga da Y dobija vrednost y_i , ako je poznato da je X dobilo vrednost x_j . Bezuslovna verovatnoća $p(y_i)$ jednaka je uslovnoj verovatnoći dobijenoj kao srednja vrednost po svim mogućim vrednostima X :

$$p(y_i) = \sum_{j=1}^M p(x_j)p(y_i/x_j),$$

gde je $p(x_j)$ — verovatnoća j -te vrednosti veličine X . Veličina $p(x_j)p(y_i/x_j)$ je verovatnoća da X dobije vrednost x_j a Y — vrednost y_i . Ona se naziva *zajednička verovatnoća (ili složena verovatnoća)* događaja (x_j, y_i) i obeležava se sa $p(x_j, y_i)$.

Pokušajmo sada da objasnimo kakvu informaciju o poslatom signalu dobijamo na prijemnom kraju kanala veze. Prvobitna (apriorna) neodređenost signala Y jednaka je njegovoj entropiji $H(Y)$:

$$H(Y) = - \sum_{i=1}^N p(y_i) \log_2 p(y_i).$$

Ako bi primljeni signal X bio jednoznačno povezan sa poslatim, posle njegovog dobijanja neodređenost bi iščezla i mi bismo dobili količinu informacija jednaku $H(Y)$. Ali u stvarnosti posle prijema, recimo, signala x_j neodređenost poslatog signala Y postaje jednaka veličini

$$H_{x_j}(Y) = - \sum_{i=1}^N p(y_i/x_j) \log_2 p(y_i/x_j),$$

pošto su nam poznate samo uslovne verovatnoće $p(y_i/x_j)$ različitih vrednosti Y .

Pošto primljeni signal X može da ima bilo koju od vrednosti $x_1, \dots, x_M$ sa verovatnoćama $p(x_1), \dots, p(x_M)$, tada je *srednja neodređenost poslatog signala, pri poznatom primljenom, jednaka*:

$$H(Y/X) = - \sum_{j=1}^M \sum_{i=1}^N p(x_j)p(y_i/x_j) \log_2 p(y_i/x_j). \quad (5.6)$$

Ovo je — *uslovna entropija* slučajne veličine Y pri zadatoj slučajnoj veličini X . Uslovna entropija uvek je manja (tačnije, nije veća) od bezuslovne:

$$H(Y/X) < H(Y),$$

pri čemu jednakost važi samo u slučaju kada poznavanje veličine X ne menja verovatnoću vrednosti veličine Y , tj. $p(y_i/x_j) = p(y_i)$, ma kakvo bilo x_j . (Tada se slučajne veličine X i Y nazivaju *nezavisne* ili *statistički nezavisne* veličine.)

Prirodno je smatrati kao meru *količine informacija u slučajnoj veličini X o slučajnoj veličini Y* veličinu za koju se smanjuje (u proseku) neodređenost veličine Y, ako nam je poznata vrednost veličine X, tj. razliku između bezuslovne i uslovne entropije

$$I(X, Y) = H(Y) - H(Y/X) = \sum_{j=1}^M \sum_{i=1}^N p(x_j, y_i) \log_2 \frac{p(x_j, y_i)}{p(x_j)p(y_i)} \quad (5.7)$$

U teoriji informacija dokazuje se da je, na taj način uvedena, mera količine informacija veoma svrsishodna i, osobito, omogućuje da se potpuno reši pitanje o tome, kakav mora biti kanal veze da bi se kroz njega moglo preneti saopštenje stvoreno u nekom izvoru saopštenja, nezavisno od konkretne prirode i smisla tih saopštenja (to će biti ljudski govor, muzika, slika, meteorološka obaveštenja, nervni impulsi u živom organizmu itd.).

Važne osobine količine informacija su njena pozitivnost i simetričnost: $I \geq 0$ i $I(X, Y) = I(Y, X)$. Drugo svojstvo znači da je količina informacija u primljenom signalu o poslatom jednaka količini informacija u poslatom signalu o primljenom. Količina informacija jednaka je nuli, ako su ulazni i izlazni signali nezavisni, tj. nikako, čak ni statistički, nisu povezani jedan s drugim. Količina informacija dostiže maksimum — veličina $H(Y)$ — kada primljeni signal jednoznačno određuje poslati, tj. $H(Y/X) = 0$ — neodređenost poslatog signala pri poznatom primljenom jednaka je nuli. (Važno je da svakoj vrednosti X jednoznačno odgovara određena vrednost Y. Međutim, jednoj istoj vrednosti Y mogu pri tome odgovarati nekoliko različitih vrednosti X.)

U opštem slučaju količina informacija zadovoljava nejednačine

$$I(X, Y) \leq H(Y) \text{ i } I(X, Y) \leq H(X).$$

Neka, na primer, na ulaz kanala dolazi signal Y, koji može primiti samo dve vrednosti $y_0 = 0$ i $y_1 = 1$ s verovatnoćama $p(y_0) = p(y_1) = 0,5$. Tada je, kao što se vidi sa sl. 5.2, $H(Y) = 1$ bit za svako slovo. Pretpostavimo da je posle mnogobrojnih ispitivanja utvrđeno da se, pri dovođenju na ulaz signala $y_1 = 1$, izlazni signal $x_1 = 1$ pojavljuje u 90% slučajeva, tj. uslovna verovatnoća $p(y_1/x_1) = 0,9$, i uslovna verovatnoća toga, da se pri ulaznom signalu $y_0 = 0$ na izlazu dobija signal $x_0 = 0$, takođe je jednaka $p(y_0/x_0) = 0,9$. Tada iz sl. 5.2 nalazimo $H(Y/X) = 0,5$ i količina informacija koja se sadrži u svakom slovu izlaznog signala ne iznosi više 1 bit, već se, u skladu sa (5.7) smanjuje do vrednosti

$$I = 1 - 0,5 = 0,5 \text{ bitova.}$$

Prethodno uvedeni pojmovi i odnosi za kvantitativnu ocenu informacija, koje se sadrže u saopštenjima, koriste se naveliko u kibernetici. Oni se primenjuju ne samo za proračune kanala veza, već i za određivanje kapaciteta memorijskih uređaja, za kvantitativne karakteristike procesa prerade informacija u sistemima, za izbor strukture informacionih mreža u složenim sistemima i slično.

Primer:

Pretpostavimo da se nekom vašem rođaku rodilo dete i da ga pitate: „Ko se rodio – dečak ili devojčica? Kolika količina informacija se sadrži u odgovoru?

Odgovor:

$$N = \log_2 2 = 1 \text{ bit.}$$

Smatra se da su rođenje devojčice i rođenje dečaka jednako verovatni događaji.

Primer:

Saopštenje je napisano u vidu decimalnog broja od 5 cifara, pri čemu se pretpostavlja da su sve cifre jednako verovatne i nezavisne. Koliku količinu informacija nosi to saopštenje? Koliko puta je manja količina informacija sadržana u saopštenju koje se sastoji iz pet binarnih cifara?

Odgovor:

Količina informacija u decimalnom broju sa pet cifara je:

$$I = 5 \log_2 10 = 5 \cdot 3,32 = 16,6 \text{ bitova,}$$

a u binarnom broju sa pet cifara:

$I = 5 \cdot \log_2 2 = 5$ bitova, tj. ovo saopštenje nosi 3,32 puta manju količinu informacija, nego saopštenje koje je izraženo sa pet decimalnih cifara.

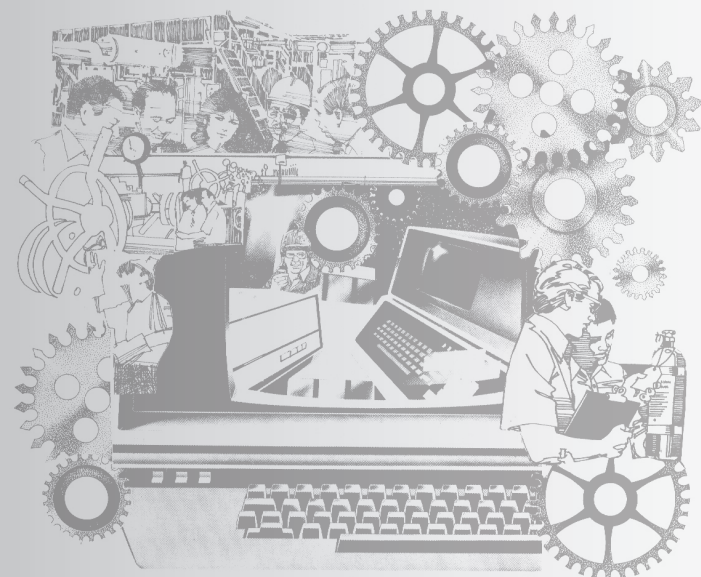
Primer:

Motor trolejbusa može da radi u jednom od 5 režima rada. Verovatnoća da se on nalazi u prvom režimu iznosi $p_1=0,08$, u drugom $p_2=0,12$, u trećem $p_3=0,15$, u četvrtom $p_4=0,28$ i u petom $p_5=0,37$. Nađite entropiju skupa (mnoštva) mogućih režima rada motora?

Odgovor:

$$H = -(0,08 \log 0,08 + 0,12 \log 0,12 + 0,15 \log 0,15 + 0,28 \log 0,28 + 0,37 \log 0,37) = 1,565$$

bitova.



6. UPRAVLJANJE

Zahtevano ponašanje upravljanog sistema, kao što je spomenuto ranije, postiže se upravljačkim dejstvima, tako da pod njihovim uticajem sistem dobija bolje (u određenom smislu) stanje, nego što bi imao pri nepostojanju upravljačkih dejstava.

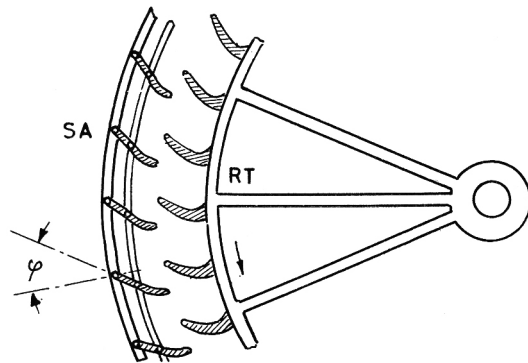
Objasnilo u kom se smislu upotrebljava reč »bolje«.

Ako se govori o veštačkom upravljanoj sistemu, koji je stvorio čovek i koristi ga za svoje ciljeve, ponašanje sistema ocenjuje njegov konstruktor, i reč *bolje* znači bolje u odnosu na cilj subjekta-konstruktor sistema. Biološki upravljani sistemi stvarali su se u procesu evolucionog razvoja žive prirode i za njih nije moguće naći subjekt koji ima određene ciljeve, radi kojih se ostvaruje upravljanje. Međutim, i za biološke sisteme pojam *boljeg ponašanja* ima smisla. On se sastoji u tome, da karakter ponašanja organizma u njegovoj prirodnoj sredini ima bitan uticaj na njegovo bitisanje i razmnožavanje. Pri tome se ocena ponašanja organizma kao upravljanog sistema određuje njegovim uzajamnim odnosom sa sredinom, tako da je *bolje* ono ponašanje koje povećava izgled datog organizma da ostane u životu i da stvara potomstvo.

U prethodnim izlaganjima je spomenuto da su neka spoljna dejstva na sistem — upravljačka dejstva, i to upravo ona koja se mogu upotrebljavati za upravljanje tim sistemom. Uticaj na ponašanje sistema može se postići kako delovanjem na njegove koordinate, tako i promenom parametara upravljanog sistema — *objekta upravljanja*. Tako na primer, upravljanje ugaonom brzinom turbine može se ostvariti promenom pritiska h vode, što predstavlja dejstvo na koordinatu koja opisuje obrtni moment turbine. Ali, isti efekat se postiže ako se pri nepromenjenom pritisku menja ugao zaokreta lopatica skretnog aparata (SA) i time utiče na pravac struje u odnosu na lopatice rotora turbine (RT), kao što je pokazano na sl. 6.1. Promena ugla φ izaziva promenu parametara koji određuju karakteristike turbine, tj. njena unutrašnja svojstva.

Mogućnosti upravljanja su utoliko šire, a upravljanje je utoliko efikasnije, ukoliko je opseg vrednosti, koje mogu imati upravljačka dejstva u procesu upravljanja, širi. Međutim, neophodno je potrebno uzeti u obzir okolnost, da je u stvarnim sistemima opseg promena svakog upravljačkog dejstva ograničen. U navedenom primeru upravljanja turbinom i pritisak h i ugao φ mogu se menjati samo u određenim granicama

$$h' \leq h \leq h'', \varphi' \leq \varphi \leq \varphi''$$



Sl. 6.1. Upravljanje ugaonom brzinom turbine pomoću promene ugla zaokreta lopatica

Pošto se upravljanje nekim objektom može ostvariti pomoću nekoliko upravljačkih dejstava, od kojih je svako ograničeno nekim graničnim vrednostima, može se u prostoru upravljačkih dejstava $Y_1, Y_2, \dots, Y_m$ izdvojiti oblast Ω , koja zadovoljava uslove

$$Y_i' \leq Y_i \leq Y_i'' \quad (i = 1, 2, \dots, m),$$

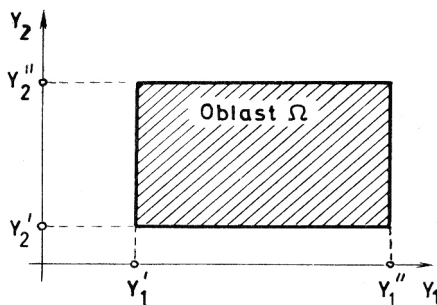
a da se unutar nje nalaze tačke koje prikazuju sve moguće skupove upravljačkih dejstava (sl. 6.2). Tu oblast ćemo nazvati *oblast mogućih dejstava*.

Često upravljačka dejstva mogu imati samo konačan broj utvrđenih vrednosti ili se mogu posmatrati kao takve veličine. Tada oblast mogućih upravljačkih dejstava sadrži konačan broj mogućih skupova upravljačkih dejstava, koja ćemo nazvati *mnoštvo mogućih dejstava*.

Temperatura u hladnjaku, na primer, može da se održava blizu svoje zadate vrednosti uključivanjem i isključivanjem rashladnog uređaja. Mnoštvo mogućih dejstava takvog sistema sastoji se iz dva upravljačka dejstva: »uključeno« i »isključeno«.

Objasnimo sada na koji način se može u svakom konkretnom slučaju izabrati, za date uslove i za dati objekt, potreban skup upravljačkih dejstava.

Da bi se upravljalo nekim objektom, potrebno je na određeni način menjati upravljačka dejstva na taj objekt. Takva promena upravljačkih dejstava može se ostvariti pomoću signala upravljanja, koji nose podatke o zahtevanim vrednostima upravljačkih dejstava.



Sl. 6.2. Oblast mogućih upravljačkih dejstava

Skup elemenata sistema, u kojima se proizvode signali upravljanja, naziva se *upravljački uređaj*. Ako su traženo ponašanje, uslovi rada objekta, a i njegova svojstva, unapred poznati, tada se u upravljački uređaj može ranije uvesti informacija o nizu upravljačkih dejstava u vidu programa upravljanja. U drugim slučajevima, kada su unapred nepoznati svi neophodni podaci za sastavljanje programa upravljanja, stvaranje upravljačkih dejstava može se organizovati u upravljačkom uređaju na osnovu informacije o situaciji nastaloj u toku rada sistema. Takve informacije mogu biti podaci o stanju upravljanog sistema, o njegovom zahtevanom stanju, o poremećajnim dejstvima i, o karakteristikama upravljanog sistema.

Prerada te informacije u upravljačkom uređaju na osnovu određenih pravila može služiti za formiranje upravljačkih dejstava. Skup pravila, po kojima se informacija, koja ulazi u upravljački uređaj prerađuje u signale upravljanja, naziva se *algoritam upravljanja*.

Potrebno je primetiti, da je upravljanje neophodno, ne samo za normalno funkcionisanje sistema, već i za obezbeđivanje njegovog razvoja u zahtevanom procesu: za razvoj organizma iz zametka, za razvoj preduzeća, za razvoj transportnog sistema, i sl. Upravljanje razvojem sastoji se u stvaranju plana razvoja objekta i u ostvarenju tog plana. Plan razvoja živih organizama je utemeljen u naslednoj informaciji, ispoljenoj u vidu strukture makromolekula koji ulaze u sastav jezgra ćelije. Plan razvoja bilo kog ekonomskog sistema je dokument, u kome se nalazi informacija o dejstvima (u vidu ulaganja kapitala, reorganizacije objekta, i sl.), kojima se ostvaruju zahtevane vremenske promene njegovih funkcija i strukture.

S obzirom na izložene postavke može se na sledeći način odrediti pojam »upravljanja«.

Upravljanje je dejstvo na objekt koje poboljšava funkcionisanje ili razvoj datog objekta, a koje je izabrano iz mnoštva mogućih dejstava, na osnovu za to raspoložive informacije.

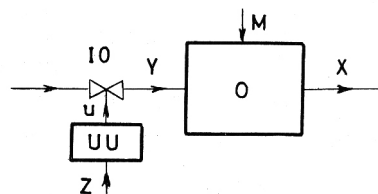
Ako se upravljanje sastoji u ustaljivanju stanja upravljanog objekta, tada se upravljanje može interpretirati kao *aktivna zaštita* od poremećaja, principijelno različita od pasivnog načina zaštite.

Pasivna zaštita se sastoji u pojačavanju takvih osobina objekta, da njegove izlazne veličine (tj. funkcije njegovih stanja koje nas interesuju) dovoljno malo zavise od poremećajnih dejstava. Evo nekoliko primera pasivne zaštite: brod je usidren i time je njegov položaj učinjen nezavisnim od vetra i struje; Džaulov sud, u kome je njegova sadržina izolovana te nema razmene toplote s okolinom; visoke carine kojima se obezbeđuje konkurentna moć industrije date zemlje na unutrašnjem tržištu; lipoidna opna Kohovog štapića tuberkuloze koja ga štiti od lekovitih supstanci; oblici polarnih biljaka poleglih po zemlji i mala površina isparavanja kod rastinja u sušnim oblastima; anti-korozivne prevlake delova mašine, itd.

Za razliku od pasivnih načina zaštite od poremećaja, u upravljanim sistemima se organizuju upravljačka dejstva koja se aktivno suprotstavljaju poremećajima. Tako se, u navedenim primerima, ustaljivanje položaja broda može postići manevrisanjem; održavanje stalne temperature nekog tela regulisanjem dovoda toplote iz spoljnog izvora; konkurentna moć industrije može se obezbediti povećanjem produktivnosti rada i smanjenjem troškova proizvodnje; bakterije mogu razlagati lekovite supstance ili menjati razmenu materije tako da lekovi postaju neškodljivi po njih; biljke mogu upravljati veličinom isparavanja otvaranjem i zatvaranjem pora, raspodelom i opadanjem lišća, itd.

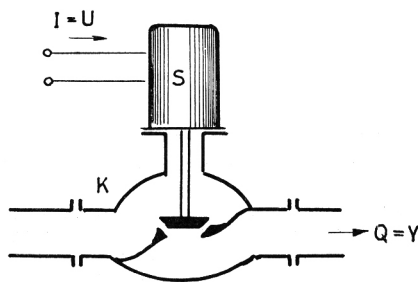
Sistem upravljanja

Upravljeni sistem i njemu priključeni upravljački uređaj obrazuju *sistem upravljanja*. Da bi signali upravljanja u , proizvedeni u upravljačkom uređaju na osnovu obrade informacije Z , mogli da menjaju upravljačka dejstva Y , potrebni su organi koji menjaju upravljačka dejstva u skladu sa signalima upravljanja — *izvršni organi*. Šema povezanosti upravljačkog uređaja (UU) i objekta O preko izvršnog organa (IO) je pokazana na sl. 6.3 za najprostiji slučaj objekta s jednim izvršnim organom.



Sl. 6.3. Šema povezanosti upravljačkog uređaja s objektom upravlja

Funkciju izvršnih organa obavljaju takvi tehnički uređaji, kao što je, na primer, solenoidni ventil (sl. 6.4) K za proticanje tečnosti, koji se otvara ili zatvara pod dejstvom električnog signala I — struje kroz namotaje solenoida S . Ulogu izvršnog organa može da obavlja i čovek, na primer krmanoš, koji okreće kormilarski točak kormila broda u skladu sa dobijenim komandama.



Sl. 6.4. Solenoidni ventil

U sistemima upravljanja rešavaju se četiri osnovna tipa zadataka: stabilizacija (ustaljivanje), izvršenje programa, praćenje i optimizacija (traženje najboljeg).

Zadaci stabilizacije sistema su zadaci održavanja nekih njegovih izlaznih veličina — *upravljanih veličina* X u blizini nekih nepromenljivih zadatah vrednosti X , uprkos dejstvima poremećaja M , koji utiču na vrednosti X . Tako na primer, radi obavljanja normalnih životnih aktivnosti organizma toplokrvne životinje, moraju biti stabilizovane takve veličine kao što su temperatura tela, sastav i pritisak krvi, i pored promena u spolnjoj sredini. U napojnim energetskim sistemima napon i struja u mreži moraju biti stabilizovani tako, da ne zavise od promena potrošnje energije.

Zadatak izvršenja programa nastaje u slučajevima, kada se zadate vrednosti upravljanih veličina X_0 menjaju u toku vremena, na ranije poznati način. Na primer, pri upravljanju balističkom raketom njeno izvođenje na zadatu trajektoriju mora da se odvija po ranije poznatom programu $X_0(t)$ promene njenog položaja u prostoru i njene brzine. Pri upravljanju položajem teleskopske cevi, u cilju kompenziranja obrtanja zemlje, takođe je potrebno pomerati je po određenom programu. Analogni zadatak se pojavljuje u proizvodnji pri izvršavanju rada prema ranije predviđenom grafiku.

Upadljivi primeri delatnosti tipa »izvršenja programa« u biologiji su razvoj organizma iz jajne ćelije, sezonski letovi ptica i metamorfoza insekata.

U slučajevima kada je promena zadanih vrednosti upravljanih veličina unapred nepoznata, javlja se zadatak *praćenja*, tj. što je moguće tačnijeg održavanja saglasnosti tekućih stanja sistema $X(t)$ sa vrednostima $X_0(t)$. Potreba za praćenjem postoji, na primer, pri upravljanju proizvodnjom robe u uslovima nepredvidivih promena potražnje; ritam i dubina disanja moraju pratiti promene fizičkih napora organizma; antena radara mora pratiti nepredviđeno kretanje aviona. (Da bi se izbegao nesporazum, podvucimo da u ovom slučaju upravljana veličina, bez sumnje, nije položaj aviona, već položaj antene radara. $X_0(t)$ — to je takav niz položaja antene, pri kome je ona sve vreme upravljena prema avionu. Za razliku od zadataka »izvršenja programa«, taj niz ne može biti zadat ranije, već ga određuje spoljni faktor — kretanje aviona.)

U nizu slučajeva zadatak upravljanja ne može se formulisati kao zadatak obezbeđivanja saglasnosti stanja sistema sa njegovim zadatim stanjem (stalnim ili promenljivim), pošto podaci o zadatom stanju ne mogu ranije da budu uvedeni u sistem upravljanja, niti se mogu dobiti tokom procesa njegovog rada. Takva situacija se pojavljuje, na primer, pri upravljanju energetskim agregatom koji radi u složenim promenljivim uslovima, kada se cilj upravljanja sastoji u obezbeđivanju optimalne (maksimalno moguće) vrednosti koeficijenta korisnog dejstva agregata, u bilo kom režimu njegovog rada. Zadaci optimizacije — utvrđivanja optimalnog, u određenom smislu, režima rada upravljanog objekta, sreću se dosta često. Primeri za njih su: upravljanje ekonomskim sistemom s ciljem maksimiziranja dobiti, upravljanje tehnološkim procesima s ciljem minimiziranja gubitka sirovina i polufabrikata, i mnogi drugi.

Direktna i povratna veza

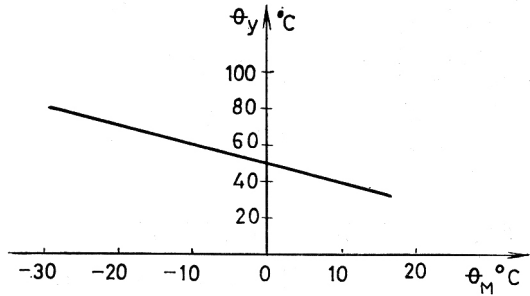
Svojstva sistema upravljanja bitno zavise od toga kakvi se izvori informacija koriste u upravljačkom uređaju za stvaranje signala upravljanja. Razmotrimo u početku sisteme u kojima informacija Z , prispela u upravljački uređaj, ne sadrži podatke o stanju X upravljanog objekta. Pri tome u Z može da ulazi program niza promena upravljačkih dejstava $Y_0(t)$, ili podaci o poremećajnim dejstvima $M(t)$. U poslednjem slučaju, radi dobijanja signala upravljanja u , u upravljačkom uređaju moraju postojati podaci o tome kakva treba da bude vrednost Y za svaku vrednost M , da bi se postigao cilj upravljanja. Algoritam upravljanja u takvim sistemima sastoji se u pretvaranju

$$U=PM \quad (6.1)$$

Operator P tog pretvaranja je postavljen u upravljački uređaj ranije, na osnovu podataka o ciljevima upravljanja i o osobinama upravljanog objekta.

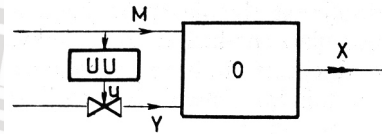
Neka je, na primer, za sistem grejanja osnovno poremećajno dejstvo temperatura spoljašnjeg vazduha, Θ_M , i neka se zadatak upravljanja sastoji u tome da se temperatura u grejanim prostorijama Θ_x održava na vrednosti bliskoj zadatoj temperaturi Θ_0 . To se može postići sa zahtevanom tačnošću utvrđivanjem temperature Θ_y vode u sistemu grejanja u skladu s grafikom, prikazanim na sl. 6.5. Pri tome osoblje, koje opslužuje sistem

grejanja, nadzirući pokazne termometre za merenje Θ_M i Θ_y deluje na organe upravljanja obezbeđujući zahtevanu funkcionalnu zavisnost $\Theta_y(\Theta_M)$. Opisani sistem upravljanja karakteriše to, da se za stvaranje upravljačkih dejstava ovde ne koriste podaci o upravljanoj veličini—temperaturi Θ_x u grejanim prostorijama.



Sl. 6.5. Grafik za određivanje potrebne temperature vode u sistemu grejanja

Sistemi, u kojima se za stvaranje upravljačkih dejstava ne koriste informacije o vrednostima upravljanih veličina, ostvarenih u procesu upravljanja, nazivaju se otvoreni sistemi upravljanja. Struktura takvog sistema pokazana je na sl. 6.6.



Sl. 6.6. Šema otvorenog sistema upravljanja

Algoritam upravljanja (6.1) koji ostvaruje upravljački uređaj UU zasniva se na ideji kompenzacije poremećaja: za svaki poremećaj M pomoću pretvaranja (6.1) bira se takva vrednost Y koja kompenzuje uticaj M na upravljanu veličinu X . Pri tome, upravljačko

dejstvo Y mora da se bira tako da suma odstupanja: $\Delta X(M)$, nastalog pod uticajem poremećaja, i $\Delta X(Y)$, nastalog pod uticajem upravljanja, bude jednaka nuli, te je

$$\Delta X(Y) = -\Delta X(M). \quad (6.2)$$

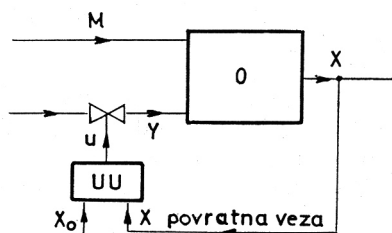
Iz (6.2) se vidi, da je za izbor upravljačkog dejstva važno raspolagati podacima o uticaju poremećaja na upravljane veličine, a ne podacima o samim poremećajima. Prema tome, upravljanje (izbor upravljačkih dejstava) se može organizovati ne izvođeci neposredno merenje poremećaja, već nadzirući samo odstupanja upravljane veličine, izazvana tim poremećajima. Tada se skupljanje podataka o odstupanjima upravljane veličine X od svoje zadate vrednosti X_0 može posmatrati kao neki posredni metod dobijanja podataka o poremećajnim dejstvima.

Iz izloženih razmišljanja sledi da se signali upravljanja mogu stvoriti i na osnovu informacija o odstupanjima upravljane veličine od svoje zadate vrednosti. Na primer, može se organizovati upravljanje sistemom grejanja, zadužujući osoblje da povećava temperaturu Θ vode u sistemu grejanja, ako je temperatura Θ_x u grejanim prostorijama niža od zadate vrednosti Θ_0 , ili da je smanjuje u suprotnoj situaciji. U algoritam upravljanja tada već ulazi informacija o vrednosti upravljane veličine, a upravljački uređaj ostvaruje pretvaranje

$$u = P(X, X_0), \quad (6.3)$$

gde je P operator koji dodeljuje svakoj kombinaciji vrednosti X i X_0 određenu vrednost u .

Sistemi, u kojima se za formiranje upravljačkih dejstava koriste informacije o vrednostima upravljanih veličina, nazivaju se zatvoreni sistemi upravljanja. Struktura zatvorenog sistema upravljanja je pokazana na sl. 6.7. Takva struktura je dobila naziv »zatvoreni« sistem, usled postojanja zatvorene konture u lancu prenosa dejstava. Stvarno, polazeći od bilo koje tačke konture i krećući se po njoj u smeru prenosa dejstava, dolazimo u polaznu tačku, na primer, po lancu $u \rightarrow Y \rightarrow X \rightarrow u$.



Sl. 6.7. Šema zatvorenog sistema upravljanja

Veza između izlaznog dejstva X_i i-tog elementa sistema i ulaza Y_j bilo kog drugog, j-tog elementa naziva se direktna veza (sprega). Veza između izlaza X_i i ulaza Y_i istog elementa naziva se povratna veza.

Povratna veza se može ostvariti bilo neposredno od izlaza elementa sistema na njegov ulaz, bilo preko drugih elemenata datog sistema.

Kao što se vidi iz sl. 6.6 i sl. 6.7 u otvorenim sistemima upravljanja koriste se samo direktne veze, a u zatvorenim sistemima i povratna veza. Tako, u šemi na sl. 6.7 ulazno—upravljačko dejstvo Y na objekt O , zahvaljujući povratnoj vezi preko upravljačkog uređaja UU , zavisi od njegove izlazne veličine X . Veza između izlaza i ulaza elementa sistema naziva se povratna, zato što je prenos dejstava, u tom slučaju, suprotnog smera od smera prenosa dejstava kroz taj element.

Povratna veza je jedan od najvažnijih pojmova kibernetike i pomaže da se shvate mnoge pojave koje se dešavaju u upravljanim sistemima različite prirode. Povratna veza može da se otkrije pri izučavanju procesa koji se odvijaju u živim organizmima, ekonomskim strukturama i sistemima automatskog regulisanja.

Povratna veza, koja povećava uticaj ulaznog dejstva na izlaznu veličinu elementa sistema, naziva se pozitivna, a ona koja smanjuje taj uticaj — negativna.

Negativna povratna veza obezbeđuje, na primer, smanjenje dijapazona promena sjajnosti slike na mrežnjači oka u poređenju sa dijapazonom sjajnosti u vidokrugu, pomoću promena prečnika zenice, u zavisnosti od osvetljenosti mrežnjače. Zahvaljujući negativnoj povratnoj vezi, uspešno je ostvareno praktično odstranjivanje uticaja parametara elektronskih pojačala na njihove radne osobine. Pozitivna povratna veza koristi se u mnogim tehničkim uređajima za povećanje koeficijenta prenosa; ona pak predstavlja uzrok pojava svih lančanih reakcija.

Negativna povratna veza, uopšte, pomaže uspostavljanje ravnoteže u sistemu kada je narušava spoljnje dejstvo, a pozitivna povratna veza izaziva još veće odstupanje od onog koje bi izazvalo spoljnje dejstvo pri nepostojanju povratne veze.

Potrebno je primetiti, da bilo koji sistem, koji sadrži povratnu vezu, predstavlja sistem sa zatvorenim konturom prenosa dejstava.

Preimućstvo zatvorenih sistema upravljanja sastoji se u tome, da se u njima može obezbediti postizanje cilja upravljanja u uslovima kada je mnogo poremećajnih dejstava, i kada se ona sva ne mogu uvek meriti, a i u slučajevima kada je unapred nepoznat uticaj poremećaja na upravljane veličine.

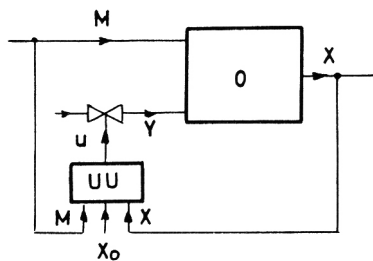
Prednost otvorenih sistema upravljanja sadrži se u tome, da se upravljačka dejstva menjaju u skladu s promenama poremećajnih dejstava istovremeno, još pre nego što poremećaji bitno promene vrednost upravljane veličine.

Sjedinjenje preimućstava otvorenih i zatvorenih sistema može se postići u *kombinovanom sistemu upravljanja*, čija je struktura prikazana na sl. 6.8.

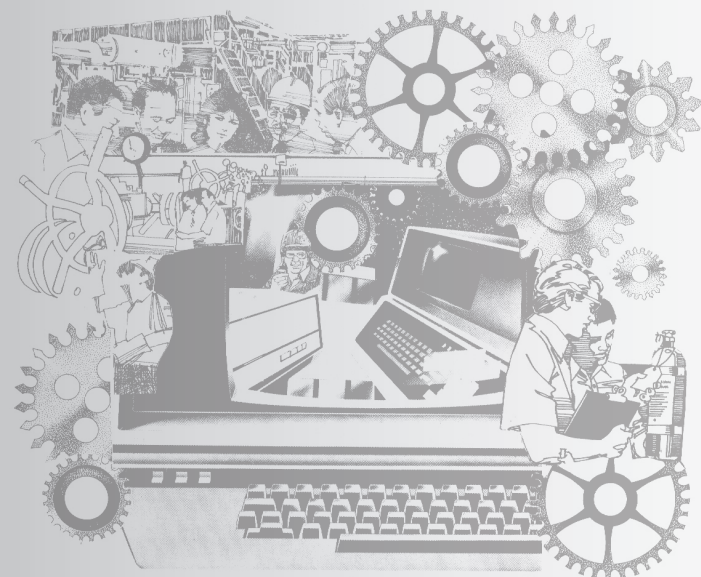
Ovde u stvaranju signala upravljanja učestvuju, kako informacija o osnovnim poremećajnim dejstvima, tako i informacija o vrednostima upravljanih veličina, a algoritam rada upravljačkog uređaja sastoji se u realizaciji pretvaranja

$$u = P(M, X, X_o). \quad (6.4)$$

Takav način upravljanja ostvaruje se, u razmatranom primeru, pomoću sledećeg uputstva osoblju: određivati temperaturu Θ_y u skladu sa grafikom 6.5, a zatim, pazeći na odstupanje temperature Θ_x od zadate Θ_o , menjati Θ_y tako, da se to odstupanje svede na minimum. Pri tome se brža, ali približna kompenzacija poremećaja postiže uticajem prve komponente upravljačkog dejstva, koja zavisi od kontrolisanog poremećaja. Tačnije, ali postupno upravljanje izvršava druga komponenta upravljačkog dejstva, koja zavisi od odstupanja upravljane veličine, i koja svodi to odstupanje na zadatu vrednost, ma kakav bio uzrok njegove pojavi.



Sl. 6.8 Šema kombinovanog sistema upravljanja



7. REGULATORI

U prethodnom odeljku je bilo reči o objektu upravljanja kao delu sistema (postrojenja, procesa, uređaja), koji primenom tehnika upravljanja treba da bude stavljen pod kontrolu. Pri tome regulator, kao deo regulacione konture, igra ključnu ulogu.

Regulator predstavlja komponentu regulacione konture koja na osnovu poređenja zadate vrednosti i merene vrednosti regulisane veličine određuje regulaciono delovanje i njime deluje na sistem kojim se upravlja. U prethodnim odeljcima je jasno pokazano da objekti upravljanja (sistemi) mogu imati vrlo različita ponašanja. Tako postoje brzi objekti (sistemi), objekti (sistemi) sa velikim transportnim kašnjenjem, kao i objekti (sistemi) sa dinamičkim kašnjenjem kod kojih nije moguća trenutna promena regulisane veličine.

Na svaki od ovih objekata se mora delovati na različit način da bi se postigli ciljevi upravljanja. Radi toga i postoje različiti tipovi regulatora, koji deluju na različit način. Jedan od osnovnih zadataka projektanta sistema upravljanja se sastoji u određivanju optimalnog zakona upravljanja za dati objekat upravljanja.

Tipovi regulatora

Regulatori se mogu ponašati na različit način, odnosno na različit način mogu davati regulaciono dejstvo kao izlazni signal na osnovu regulacione greške kao ulaznog signala. Postoje dve osnovne klase regulatora:

- ♦ regulatori sa kontinualnom vrednošću izlaza - na svom izlazu daju kontinualno promenljiv signal, koji zavisi od regulacione greške.

- ♦ regulatori sa diskretnim vrednostima izlaza - kao izlazni signal mogu se pojaviti samo vrednosti iz konačnog skupa unapred određenih vrednosti. Najpoznatiji regulatori iz ove klase su dvopoložajni regulatori, koji na svom izlazu daju samo regulaciona dejstva "uključeno" i "isključeno". Primer ovakvog regulatora je termostat kakav se susreće u pegli, bojleru, grejalici i sl. Kada je regulaciona greška pozitivna, odnosno kada je zadata vrednost temperature veća od trenutne vrednosti temperature kao regulisane veličine, termostat daje regulaciono dejstvo "uključeno", čime se dovodi struja na grejač, što prouzrokuje povećanje temperature. Kada je regulaciona greška negativna, odnosno kada je trenutna vrednost temperature premašila zadatu vrednost temperature, termostat daje regulaciono dejstvo "isključeno", odnosno prekida se strujno kolo grejača. Radi toga, zbog gubitaka toplote, temperature postepeno opada.

I dok se dvopoložajni regulatori jako često susreću u aparatima koji se koriste u širokoj potrošnji, regulatori koji se koriste u složenijim sistemima, mašinama, te procesima različite prirode, su u najvećem broju slučajeva kontinualni. Upravo ovi regulatori se nalaze u fokusu teorije automatskog upravljanja i u nastavku će biti razmatrani osnovni tipovi kontinualnih regulatora.

Svaki objekat upravljanja poseduje svoj karakterističan vremenski odziv. Ovaj vremenski odziv je praktično određen konstrukcijom mašine, postrojenja ili tehnologijom procesa, i najčešće ga nije moguće modifikovati. Vremenski odziv nekog sistema se može odrediti eksperimentalno ili teorijskom analizom. Regulator takođe predstavlja dinamički sistem, koji ima svoj karakteristični vremenski odziv.

Međutim, za razliku od vremenskog odziva sistema kojim se upravlja, vremenski odziv regulatora određuje projektant sistema automatskog upravljanja i to na taj način da sistem u celini zadovolji tražene performanse. Drugim rečima, vremenski odziv celokupnog sistema regulacije treba da ispuni postavljene zahteve.

Regulaciono dejstvo klasičnih kontinualnih regulatora se sastoji od tri osnovne komponente:

- ♦ proporcionalnog delovanja (P-delovanja),
- ♦ integralnog delovanja (I-delovanja),
- ♦ diferencijalnog delovanja (D-delovanja).

Ove tri osnovne komponente delovanja regulatora proizlaze iz načina na koji se određuje regulaciono dejstvo, na osnovu ulaznog signala u regulator.

Regulator sa proporcionalnim dejstvom

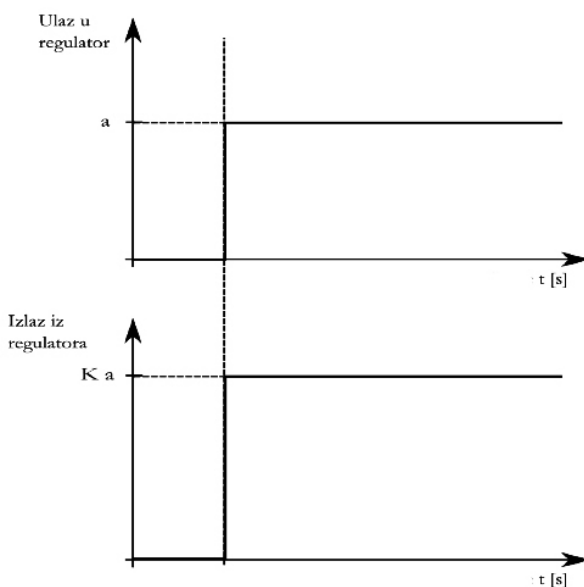
Kod proporcionalnog regulatora je vrednost regulacionog dejstva proporcionalna vrednosti ulaznog signala u regulator (slika 7.1.). Kada je regulaciona greška velika, i regulaciono dejstvo će biti veliko. U protivnom, kada je regulaciona greška mala, i regulaciono dejstvo će biti malo.

Na slici se može uočiti da vremenski odziv proporcionalnog regulatora idealno prati promene ulaznog signala u regulator (regulacione greške):

$$r(t) = \begin{cases} K \cdot a, & t \geq t_0 \\ 0, & t < t_0 \end{cases}$$

Osnovni parametar proporcionalnog regulatora je pojačanje regulatora K , koje se može izračunati kao količnik vrednosti izlaznog i ulaznog signala regulatora.

Dobra strana proporcionalnog regulatora je njegova jednostavna realizacija i jednostavno podešavanje, obzirom da poseduje samo jedan parametar. Međutim, loša strana proporcionalnog regulatora je to što on daje regulaciono dejstvo samo u slučaju da postoji nenulta vrednost regulacione greške. Radi toga nije moguće postići da sistem ima regulacionu grešku jednaku nuli ukoliko se koristi samo proporcionalni regulator, jer u tom slučaju regulator ne daje nikakvo regulaciono dejstvo, te će vrednost regulisane veličine neizbežno pod uticajem poremećaja otklizati sa zadate vrednosti. Dakle, ukoliko se koristi samo proporcionalni regulator, uvek će postojati neka preostala vrednost regulacione greške koju ovaj regulator nije u stanju ukloniti.



Sl. 7.1 Vremenski odziv proporcionalnog regulatora na step ulazni signal

Regulator sa integralnim dejstvom

Ukoliko regulator ima integralno dejstvo, njegovo regulaciono delovanje će predstavljati signal regulacione greške sumiran tokom vremena (odnosno integrisana regulaciona greška) (slika 7.2.). To znači da za kratkotrajna odstupanja od zadate vrednosti integralni regulator neće dati znatno regulaciono dejstvo. Međutim, ukoliko regulaciona greška traje duže vremena, regulaciono dejstvo će postepeno porasti i neće iščeznuti sve dok regulaciona greška ne opadne na nultu vrednost.

Sa slike se vidi da kod integralnog regulatora vrednost regulacionog dejstva nije proporcionalna regulacionoj greški, nego je brzina promene regulacionog dejstva proporcionalna regulacionoj greški. Tako, ako je do skokovite promene regulacione greške došlo u trenutku t_0 , vremenska promena regulacionog dejstva se može odrediti kao:

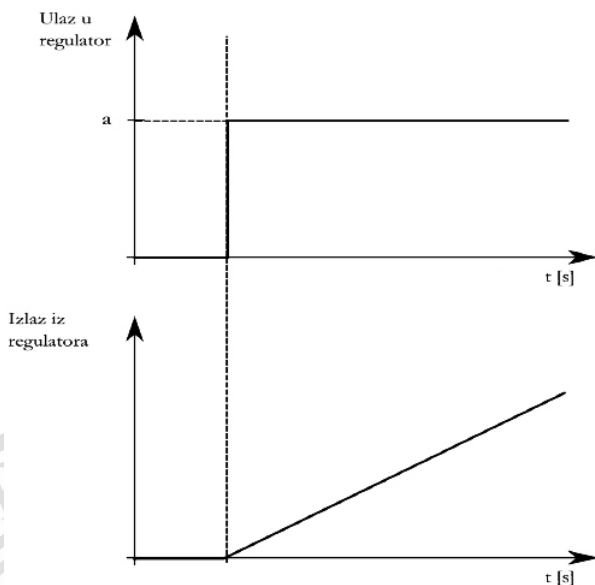
$$r(t) = \frac{1}{T_i} \cdot \int_{t_0}^{\infty} a \cdot dt,$$

odnosno prenosna funkcija
I- regulatora je data sa:

$$G_r(s) = \frac{1}{T_i \cdot s}$$

Regulator sa integralnim dejstvom je pogodan za uklanjanje preostale regulacione greške. Ako se regulaciona greška jako promeni, regulaciono dejstvo počinje da raste velikom brzinom. Kao posledica, regulaciona greška počinje postepeno da opada, što dovodi i do postepenog smanjivanja regulacionog dejstva, sve do postizanja nulte vrednosti regulacione greške.

Međutim, treba imati na umu da regulator gotovo nikada ne poseduje samo integralno dejstvo. Ovo dejstvo je nepogodno za veliki broj objekata upravljanja koji poseduju velika kašnjenja. Usled toga regulator sa integralnim dejstvom ovakve objekte može dovesti u stanje oscilovanja.



Sl. 7.2. Vremenski odziv integralnog regulatora na step ulazni signal

Regulator sa diferencijalnim dejstvom

Regulator sa diferencijalnim dejstvom se koristi za uklanjanje brzo promenljivih vrednosti regulacione greške. Kod ovog regulatora se prati brzina promene regulacione greške, i na osnovu nje se određuje vrednost regulacionog dejstva. Drugim rečima, signal regulacione greške se diferencira. Ukoliko se signal regulacione greške brzo menja, regulator sa diferencijalnim dejstvom će trenutno dati veliku vrednost regulacionog dejstva. Vremenski odziv idealnog regulatora sa diferencijalnim dejstvom je dat na slici 7.3. Na slici je predstavljen odziv idealnog diferencijatora na step ulazni signal. Već iz prirode prikazanog odziva se vidi da idealni diferencijator ne postoji, nego se diferencijalno dejstvo uvodi preko realnog diferencijatora.

Na slici 7.4. je predstavljen odziv realnog diferencijalnog regulatora, sa koje se vidi da je amplituda skoka vrednosti izlaza regulatora konačna i zavisna od parametara regulatora, te postepeno eksponencijalno opada prema nuli. Vremenski odziv realnog diferencijatora na step ulazni signal i njegova prenosna funkcija su:

$$r(t) = \frac{T_d}{T_s} \cdot a \cdot e^{-\frac{t}{T_d}},$$

$$G_r(s) = \frac{T_d \cdot s}{T_s \cdot s + 1}$$

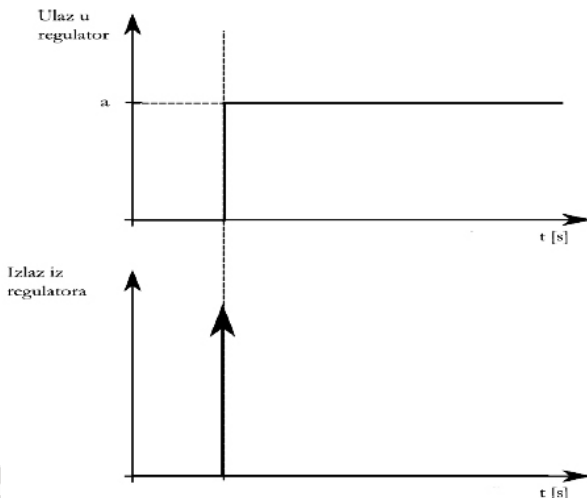
Međutim, primena regulatora koji poseduje samo diferencijalno dejstvo nema smisla,

- ♦ jer regulaciono delovanje postoji samo u slučaju da postoji promena vrednosti
- ♦ regulacione greške. Znači da ovaj regulator ne bi davao nikakvo regulaciono dejstvo
- ♦ ukoliko bi regulaciona vrednost imala nenultu, ali konstantnu vrednost. Zato se ovakvo dejstvo pojavljuje isključivo u kombinaciji sa nekim od gore navedenih (proporcionalnim, integralnim ili u kombinaciji sa oba).

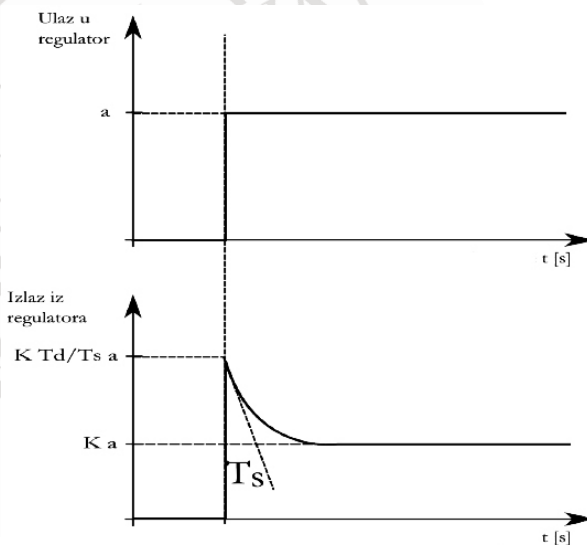
PI regulator

PI regulator predstavlja regulator sa kombinovanim proporcionalnim i integralnim delovanjem. Na taj način se koriste dobre osobine oba ova regulatora: brzo reagovanje na pojavu regulacione greške i uklanjanje preostale vrednosti regulacione greške. PI regulator se koristi u velikom broju slučajeva, a osim pojačanja proporcionalnog delovanja, poseduje još jedan parametar, koji određuje intenzitet delovanja integralne komponente. Vremenski odziv PI regulatora je predstavljen na slici 7.5., a izraz na osnovu kojega formira regulaciono dejstvo i njegova prenosna funkcija su najčešće dati u formi:

$$r(t) = K \cdot \left(e(t) + \frac{1}{T_i} \cdot \int_{t_0}^{\infty} e(t) dt \right),$$



Sl. 7.3. Vremenski odziv idealnog diferencijalnog regulatora na step ulazni signal



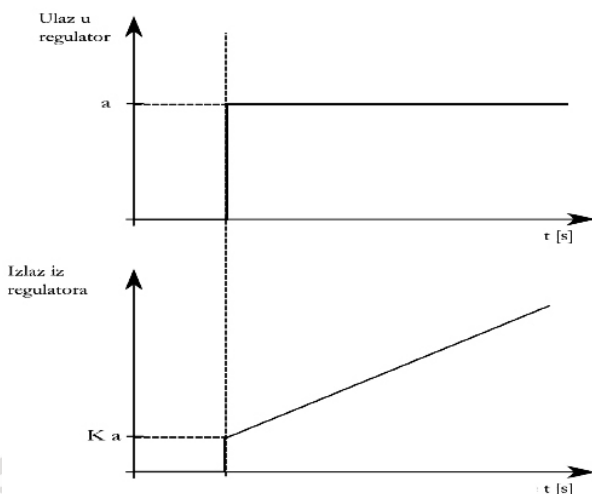
Sl. 7.4. Vremenski odziv realnog diferencijalnog regulatora na step ulazni signal

$$G_r(s) = K \cdot \left(1 + \frac{1}{T_i \cdot s} \right)$$

Može se reći da parametar T_i određuje koliko je PI regulator brži od čistog integralnog regulatora. Što je T_i manje, integralna komponenta u regulacionom dejstvu raste brže. Suprotno, što je T_i veće, integralna komponenta je manja i za $T_i \rightarrow \infty$ ponašanje PI regulatora se približava ponašanju čistog P regulatora.

Delovanje PI regulatora je veće što je vrednost proporcionalnog pojačanja K veća, a vrednost parametra T_i manja. Međutim, ukoliko se P i I dejstvo regulatora previše povećaju, može doći do pojave oscilacija regulisane veličine, odnosno do pojave nestabilnosti. Za svaku regulacionu konturu su uslovi pojave oscilatornog ponašanja (nestabilnosti) različiti i potrebno ih je odrediti, da bi se parametric regulatora mogli podesiti na vrednosti koje garantuju kvalitetno ponašanje konture.

Sl. 7.5. Vremenski odziv PI regulatora na step ulazni signal



PD regulator

PD regulator kombinuje proporcionalno i diferencijalno delovanje. Pri tome proporcionalno delovanje omogućava da regulator daje regulaciono dejstvo u skladu sa veličinom regulacione greške. S druge strane, diferencijalno delovanje omogućava da regulator reaguje na promenu regulacione greške, te da regulaciono dejstvo koje će uticati da regulaciona greška ne poprimi velike vrednosti. U praktičnoj primeni se ne koristi čisto diferencijalno dejstvo (tzv. idealni diferencijator), nego njegova modifikacija (tzv. realni diferencijator). Za ovo postoje razlozi tehničke (konstruktivne) i praktične prirode. Regulaciono dejstvo PD regulatora za step signal na ulazu i njegova prenosna funkcija su najčešće date u formi:

$$r(t) = K \cdot \left(a + \frac{T_d}{T_s} \cdot a \cdot e^{-\frac{t}{T_i}} \right),$$

$$G_r(s) = K \cdot \left(1 + \frac{T_d \cdot s}{T_s \cdot s + 1} \right)$$

Vremenski odziv PD regulatora na step ulazni signal je dat na slici 7.6. Treba napomenuti da se PD regulator ne susreće često u upotrebi iz dva osnovna razloga. Prvi razlog

je što nije u stanju da ukloni preostalu vrednost regulacione greške, a drugi razlog je da postoji opasnost da diferencijalna komponenta u regulacionom dejstvu, ukoliko nije pažljivo određena, sistem dovede do nestabilnosti.

PID regulator

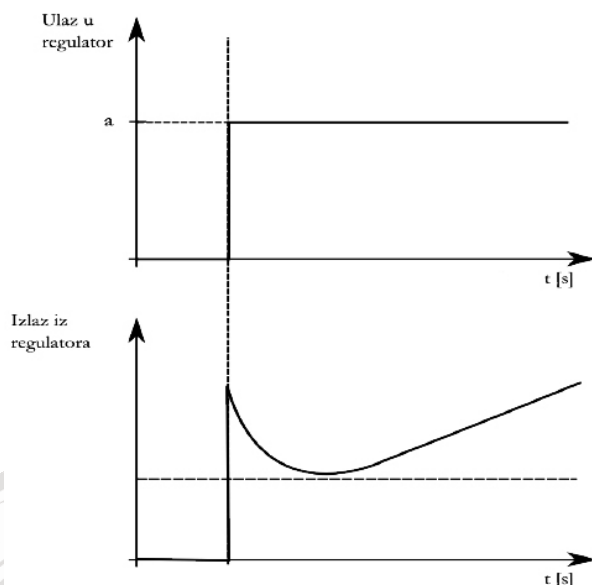
PID regulator kombinuje sva do sada pomenuta dejstva u okviru jedinstvenog regulatora. Vremenski odziv PID regulatora je predstavljen na slici 7.7., a regulaciono dejstvo za step signal na ulazu se formira najčešće na osnovu izraza:

$$r(t) = K \cdot \left(a + \frac{1}{T_i} \cdot \int_{t_0}^{\infty} a \cdot dt + \frac{T_d}{T_s} \cdot a \cdot e^{-\frac{t}{T_i}} \right),$$

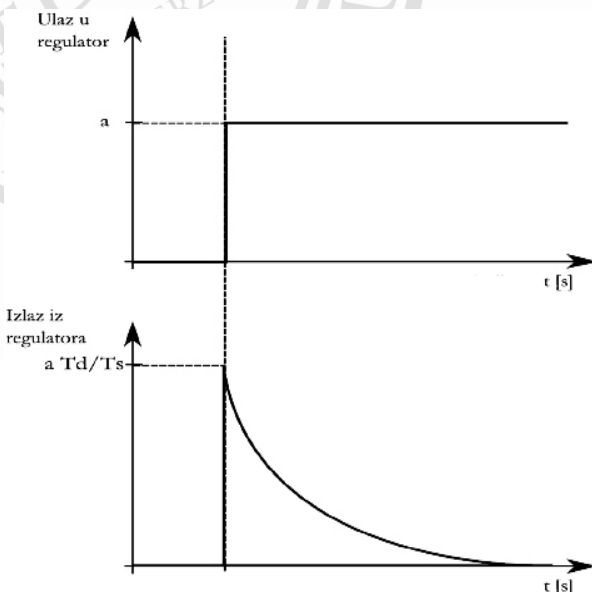
a njegova prenosna funkcija je:

$$G_r(s) = K \cdot \left(1 + \frac{1}{T_i \cdot s} + \frac{T_d \cdot s}{T_s \cdot s + 1} \right)$$

PID regulator predstavlja standardni industrijski regulator, čiji parametri su pojačanje K , te parametri integralnog i diferencijalnog člana T_i , T_d i T_s . Podešavanje PID regulatora se sastoji u određivanju vrednosti ovih parametara za koje se postiže optimalno ponašanje regulacione konture kao celine.



Sl. 7.6. Vremenski odziv PD regulatora na step ulazni signal



Sl. 7.7. Vremenski odziv PID regulatora na step ulazni signal

Tehnička realizacija regulatora

U prethodnim odeljcima su navedene osobine osnovnih tipova regulatora i data njihova formalna predstava. Tehnička realizacija datih izraza može biti različita: mehanička, pneumatska, električna, elektronska i sl. Iako se i danas mogu susresti mehaničke i pneumatske izvedbe regulatora, može se slobodno reći da u savremenom automatskom upravljanju jedino električne i elektronske izvedbe regulatora igraju značajnu ulogu.

Električni/elektronski regulatori rade sa električnim ulaznim i izlaznim signalima. Radi toga je neophodno da se u regulacionoj konturi koristi odgovarajući merni pretvarač fizikalne veličine u električnu i izvršni organ za pretvaranje električnog regulacionog signala u manipulativnu veličinu.

Da bi se pojednostavilo povezivanje mernih pretvarača i izvršnih organa različitih proizvođača sa industrijskim regulatorima, definisane su standardne vrednosti strujnih i naponskih signala, što je dato u tabeli 3.1.

Tabela 3.1. Standardni električni signali

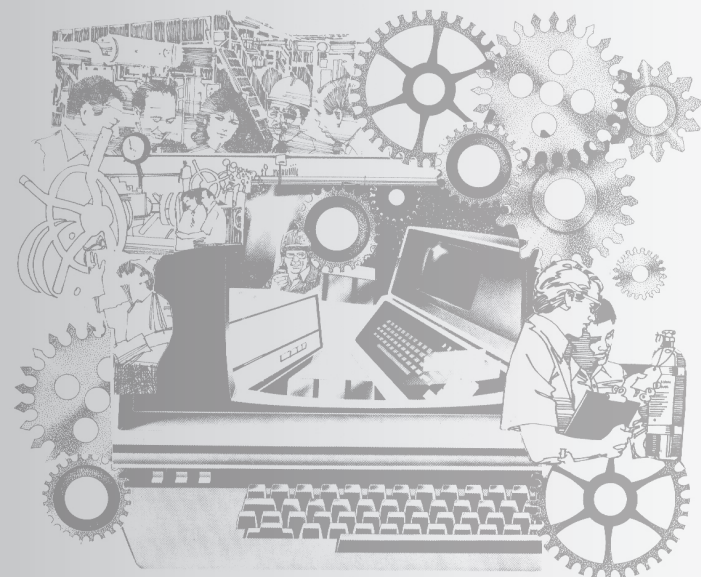
Vrsta signala	Opseg vrednosti
Strujni	0 .. 20 mA 4 .. 20 mA
Naponski	0 .. 5 V 0 .. 10 V -5 .. 5 V -10 .. 10 V

Unutar regulatora, signali se mogu procesirati na različite načine. Standardni kontinualni industrijski regulatori su najčešće elektronske izvedbe, uz upotrebu operacionih pojačavača. Moderni industrijski regulatori koji se susreću danas su gotovo isključivo realizovani kao mikroračunarski sistemi na bazi mikroprocesora.

Osnovna razlika između analogne i digitalne realizacije regulatora je sledeća:

- ♦ Kod regulatora na bazi operacionih pojačavača se naponski ili strujni signal direktno prevodi u odgovarajuću formu regulacionog signala.
- ♦ Kod digitalnih regulatora se analogni ulazni signali pretvaraju u digitalne signale. Digitalni signali se procesiraju primenom odgovarajućeg algoritma i formira se digitalna forma regulacionog dejstva. Ovaj digitalni signal se zatim na odgovarajući način pretvara u analogni signal koji predstavlja regulaciono dejstvo.

Iako su analogna i digitalna tehnologija realizacije regulatora sa stanovišta teorije bitno različite, ne postoje razlike u izvedbama regulatora sa praktične strane.



8. OPTIMALNO UPRAVLJANJE

Zadatak svakog upravljanja, kao što je već spomenuto ranije, sastoji se u aktivnom delovanju na objekt, s ciljem da se poboljša njegovo ponašanje. Ali, da bi se mogli porediti različiti vidovi ponašanja upravljanog sistema, i izdvojiti među njima najbolji, neophodno je raspolagati nekom merom, pogodnom za taj cilj, veličinom, koja će karakterisati efikasnost upravljanja — *kriterijumom efikasnosti* (kriterijum optimalnosti, indeks optimalnosti ili indeks performansi).

Kao kriterijum efikasnosti mogu se usvojiti razne veličine, zavisno od namene sistema i uslova u kojima radi. Tako na primer, za sistem upravljanja kretanjem voza vreme T njegovog kretanja od polazne stanice do stanice opredeljenja može biti kriterijum efikasnosti; za sistem upravljanja navodnjavanjem — dobit G od realizacije letine, skupljene sa navodnjenog zemljišta.

Svakoj varijanti upravljanja odgovara određena vrednost kriterijuma efikasnosti J , i zadatak *optimalnog upravljanja* sastoji se u tome, da se nađe i ostvari ona varijanta upravljanja, pri kojoj će odgovarajući kriterijum efikasnosti imati *najpovoljniju* vrednost. U navedenim primerima zadatak se sastoji u određivanju programa promene vuče lokomotive, pri kome će trajanje vožnje biti minimalno: $J = T = \min$, ili programa navodnjavanja, kojim bi se ostvarila maksimalna dobit od realizacije letine: $J = G = \max$. Pri tome je neophodno uzeti u obzir činjenicu da upravljačka dejstva mogu da se menjaju samo u određenim, ograničenim opsezima, tj. ne smeju da izađu van granica oblasti mogućih (dopuštenih) dejstava. Osim toga, sistemu mogu biti postavljena dopunska ograničenja: ograničenja vrednosti faznih koordinata, ograničenja složenosti algoritma upravljanja, opsega korišćene informacije i dr.

Pod optimalnim upravljanjem podrazumevaćemo mnoštvo upravljačkih dejstava, u skladu sa ograničenjima postavljenim sistemu, koje obezbeđuje najpovoljniju vrednost kriterijuma efikasnosti.

Optimalni proces

Pretpostavimo da se zadatak sastoji u prevođenju sistema iz nekog početnog stanja X_p u zadato stanje X_k pomoću upravljačkog dejstva Y . U faznom prostoru stanja X_p i X_k odgovaraju tačke a_p i a_k , a prelazu sistema iz a_p u a_k — neka trajektorija koja spaja te tačke (sl. 8.1).

Raspolažući upravljačkim dejstvom Y , može se izabrati mnogo varijanti upravljanja, koje zadovoljavaju zahtev, da se sistem prevede iz stanja a_p u a_k . Svako od tih varijanti upravljanja odgovara po jedna trajektorija, koja povezuje a_p sa a_k . Međutim, u odnosu na kriterijum efikasnosti J , te trajektorije nisu jednako vredne; svakoj od njih odgovara određena vrednost J , jednaka $J_1, J_2, \dots$.

Zadatak nalaženja optimalnog upravljanja može se u tom slučaju tumačiti kao zadatak, da se iz mnoštva mogućih trajektorija, koje povezuju a_p i a_k , izabere ona, za koju kriterijum efikasnosti ima najpovoljniju vrednost.

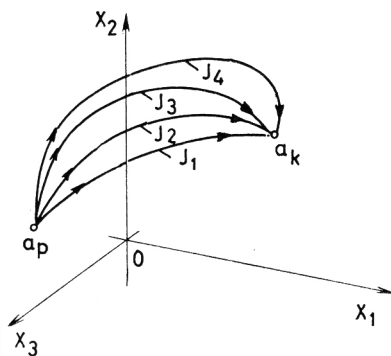
Razmotrićemo, kao primer, zadatak nalaženja optimalnog upravljanja lokomotivom, koja treba za najkraće vreme da preveze kompoziciju iz jedne stanice S_p u drugu S_k . Od faktora, koji ograničavaju moguća upravljanja, uzećemo u obzir, da ubrzanje voza u za vreme zaleta i kočenja ne sme da izađe van dozvoljenih granica

$$-u < u < +u,$$

a brzina kretanja $\mathcal{G}$ ne sme da pređe graničnu dopuštenu brzinu $\bar{\mathcal{G}}$. Da bi voz za najkraće vreme prešao rastojanje od S_p do S_k , neophodno je da srednja brzina kretanja za vreme prelaza bude što je moguće veća. Radi toga je nužno u početku procesa povećavati brzinu što je moguće intenzivnije, tj. uvećavati brzinu kompozicije s najvećim dozvoljenim ubrzanjem $+u$. Tada će brzina $\mathcal{G}$ rasti po linearnom zakonu

$$\mathcal{G} = \bar{u} \cdot t.$$

Ako bi se takav režim održavao do dolaska kompozicije u zadatu stanicu S_k , njena brzina bi bila različita od nule, a usled postavljenog ograničenja intenziteta kočenja trenutno smanjenje brzine je nemoguće, i voz bi prošao mesto opredeljenja. U skladu s tim, da bi se zaustavio u određenoj stanici, neophodno je početi kočenje pre nego što sistem prispe u



Sl. 8.1. Skup mogućih trajektorija prelaza sistema iz stanja a_p u a_k

položaj S_k . Pri tome, ukoliko kočenje bude intenzivnije, utoliko se ono može kasnije započeti; znači, srednja brzina kretanja će biti veća. Ovo pokazuje, da usporavanje kompozicije treba obaviti pri njegovom najvećem dopuštenom intenzitetu $-u$.

U procesu usporavanja pod uticajem konstantnog negativnog ubrzanja $-u$, brzina će opadati po linearnom zakonu $\mathcal{G} = \mathcal{G}_M - \bar{u} \cdot t$.

Na taj način se zaključuje, da proces najbržeg prebacivanja kompozicije iz S_p u S_k treba da se sastoji iz dva intervala: zaleta sa graničnim ubrzanjem $+u$ i kočenja sa graničnim negativnim ubrzanjem $-u$. Izborom trajanja intervala zaleta t_z i kočenja t_k može se obezbediti ispunjenje uslova, da se u trenutku završetka kočenja, (kada je brzina kretanja jednaka nuli) voz nalazi tačno u određenoj stanici S_k .

Takva — optimalna u smislu vremena prelaza — forma procesa kretanja voza pokazana je na sl. 8.2.

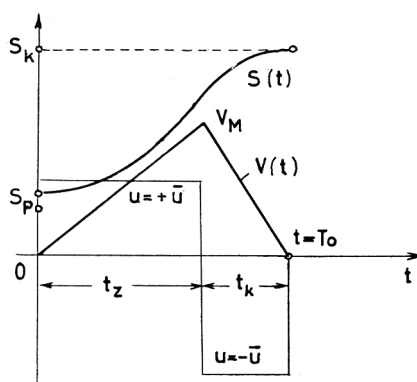
Ako je rastojanje između stanica S_p i S_k toliko veliko, da pri takvom karakteru procesa brzina kretanja $\mathcal{G}$ dostiže svoju graničnu vrednost $\mathcal{G}$, tada će proces optimalnog kretanja, očigledno, imati još i treći interval t_b , u toku koga će se kompozicija kretati konstantnom brzinom, jednakom graničnoj $\mathcal{G}$, što je pokazano na sl. 8.3.

Kao što je poznato, u periodima jednako ubrzanog i jednako usporenog kretanja voza, njegov put će se vremenski menjati po zakonu kvadratne parabole, a za vreme kretanja konstantnom brzinom — po linearnom zakonu.

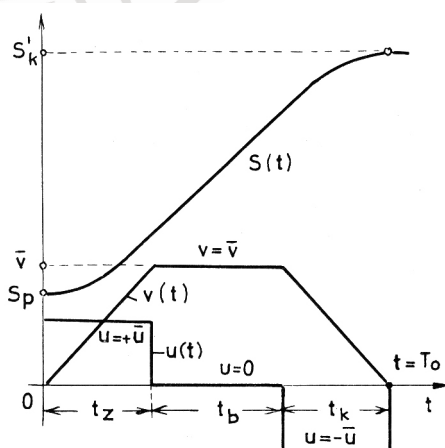
Ako se posmatra uticaj mašinovode na ubrzanje ili usporenje voza kao upravljačko dejstvo Y , a put S — kao izlazna veličina, tada je lako uočiti analogiju između zakona kretanja razmatranog sistema i sistema hidrauličnog pogona, analiziranog ranije. Stoga su fazni portreti tih sistema analogni i predstavljaju skupove kvadratnih parabola, čiji se parametri menjaju u funkciji od vrednosti Y na način koji je prikazan na sl. 4.6.

Koristeći fazne portrete sa sl. 4.6, na osnovu prethodnih razmišljanja, može se konstruisati u faznoj ravni

$$X_1 = S - S_k, X_2 = \mathcal{G}$$

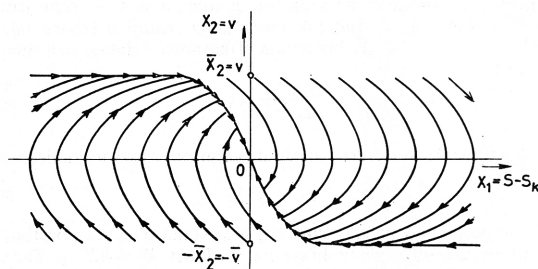


Sl. 8.2 Optimalni grafik kretanja voza pri ograničenju ubrzanja



Sl. 8.3 Isti grafik pri dopunskom ograničenju brzine

skup optimalnih trajektorija prelaza iz bilo kog početnog stanja u bilo koje zadato stanje analiziranog sistema (sl. 8.4). Isti dijagram optimalnih trajektorija važiće za bilo koji sistem, u kome su ograničeni brzina i ubrzanje izlazne veličine X_{I_i} , a kriterijum efikasnosti upravljanja je vreme prelaza.



Sl. 8.4. Skup optimalnih trajektorija prelaza

Optimalna strategija

Često se zadatak optimalnog upravljanja sastoji u izboru optimalnog niza rešenja za neki *višestepeni proces*. Sa zadacima takve vrste srećemo se, na primer, prilikom planiranja ulaganja kapitala u preduzeće na nekoliko godina ako težimo izboru takvog kontinuiteta ulaganja kapitala svake godine, da se za određeni period ostvari maksimalan učinak za privredu. Analogan zadatak se javlja prilikom izbora niza rešenja za modernizaciju proizvodnje ako stremimo ostvarenju maksimalne proizvodnje u određenom periodu, uzimajući u obzir pad produktivnosti preduzeća za vreme reorganizacije proizvodnje. Rešavanje zadataka tog tipa — zadataka obrade optimalne strategije usvajanja rešenja, može se zasnivati na matematičkim metodama, razrađenim za tu svrhu: **principu maksimuma** L. S. Pontrjagina ili **dinamičkom programiranju** R. Belmana.

Razmotrimo zadatak određivanja optimalne strategije višestepenog procesa usvajanja rešenja na primeru upravljanja stočarskom farmom.

Neka je kriterijum efikasnosti J dobit ostvarena prodajom stoke u toku N godina. Označavajući sa Y_i — broj grla (od ukupnog broja grla X_i) prodatih na kraju i -te godine, a sa C — cenu jednog grla, uočavamo da se zadatak upravljanja sastoji u izboru takvog niza $Y_1, Y_2, \dots, Y_N$ da kriterijum efikasnosti J dobije maksimalnu vrednost

$$J = \sum_{i=1}^N C \cdot Y_i = \max. \quad (8.1)$$

Pri tome Y_i očigledno, može da se menja samo u granicama

$$0 < Y_i < X_i. \quad (8.2)$$

Pretpostavimo da ukupan broj grla stada X_{i-1} na početku i -te godine naraste do njenog kraja na vrednost $X_i = kX_{i-1}$. Tada se broj grla, koji ostaje za odgoj na kraju i -te godine, nalazi iz izraza

$$X_i^* = k(X_{i-1} - Y_{i-1}) \quad (8.3)$$

Postojanje optimalne strategije u datom slučaju proizilazi iz činjenice, da sa povećanjem prodaje stoke u i -toj godini raste i dobit za tu godinu, ali će zato u narednim godinama biti manja, pošto će se sporije uvećavati broj grla stada. Ako se pak proda suviše malo stoke, tada će svaki godišnji doprinos iznosu (8.1) biti veoma mali i optimalna efikasnost se, takođe, neće postići. Postoji, očigledno, neka najpovoljnija, *optimalna strategija, koja maksimizira ukupnu dobit ostvarenu za N godina*. Tu strategiju i treba pronaći.

U skladu s metodom dinamičkog programiranja počnimo razmatranje od poslednjeg koraka, tj. sa izborom vrednosti Y_N . Očigledno je da će se najveći doprinos iznosu (8.1) u N -toj godini postići ako izaberemo najveću moguću vrednost za Y_N , što saglasno ograničenju (8.2), daje vrednost $Y_N = X_N$. Za prethodnu ($N-1$)-u godinu nalazimo da će, ma kakva bila vrednost Y_{N-1} , doprinos ostvaren u njoj iznositi CY_{N-1} novčanih jedinica, a smanjenje uloga će u N -toj godini (uzimajući u obzir da se broj grla stoke povećava k — puta godišnje), izazvano prodajom Y_{N-1} grla stoke u prethodnoj godini, iznositi kCY_{N-1} novčanih jedinica, što bi se u rezultatu negativno odrazilo na ukupnu vrednost za J .

Izvedeći analogna rasuđivanja za sve prethodne godine, doći ćemo do zaključka, da u datom slučaju ne treba prodavati stoku za sve vreme, osim u toku poslednje godine.

Na taj način se saznaje, da optimalna strategija za analizirani primer ima sledeći karakter:

$$\left. \begin{array}{l} Y_1 = 0 \\ Y_2 = 0 \\ \dots \\ Y_{N-1} = 0 \\ Y_N = X_N \end{array} \right\} \quad (8.4)$$

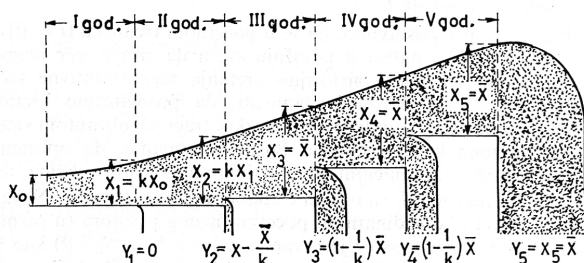
Pokažimo sada da se optimalna strategija suštinski menja, postavljajući dopunska ograničenja sistemu. Uzmimo u obzir, na primer, da je broj grla stoke na farmi ograničen na $\bar{X}$ grla:

$$X_i \leq \bar{X}, \quad (8.5)$$

što je određeno veličinom prostora za smeštaj stoke, izvorima stočne hrane, i sl. Tada, izvedeći rasuđivanja, analogna napred iznetim, krećući se od kraja procesa ka njegovom početku, doći ćemo do zaključka, da se optimalna strategija sastoji u dovođenju broja grla do granične vrednosti $\bar{X}$ u najkraćem roku, i njegovom daljem održavanju na toj vrednosti do poslednje godine, kada stado treba da bude potpuno likvidirano. Optimalna strategija za slučaj $N = 5$ slikovito je prikazana dijagramom na sl. 8.5. Ona predstavlja sledeći niz vrednosti za Y :

$$\left. \begin{aligned} Y_1 &= 0, \\ Y_2 &= X_2 - \frac{\bar{X}}{k}, \\ Y_3 &= \left(1 - \frac{1}{k}\right) \cdot \bar{X}, \\ Y_4 &= \left(1 - \frac{1}{k}\right) \cdot \bar{X}, \\ Y_5 &= X_5 = \bar{X}. \end{aligned} \right\} \quad (8.6)$$

Iz navedenih primera, a i na osnovu razmatranja optimalnih procesa, izvršenog ranije, vidi se, da je jedna od karakterističnih osobina optimalnog upravljanja, da u bilo kom trenutku optimalnog procesa upravljani sistem treba da se nalazi na granici ograničenja.



Sl. 8.5. Optimalna strategija za stočarsku farmu

Izopovršine

Pronalaženje optimalnog upravljanja u većini slučajeva predstavlja veoma težak zadatak. I, mada su metode za njegovo rešavanje razrađene za široku klasu zadataka, praktična primena tih metoda je, po pravilu, povezana s potrebom za toliko velikim brojem izračunavanja.

Međutim, proračun se može bitno uprostiti ako se preinači zadatak u tom smislu, da se umesto nalaženja optimalnih trajektorija prelaza sistema iz početnih stanja u zadato, postavi zadatak određivanja granica oblasti u prostoru stanja, koje se odlikuju određenom vrednošću kriterijuma efikasnosti. Pokazuje se, da se tada uspevaju dobiti neophodni podaci i o optimalnim prelazima.

Ograničimo se, na primer, na klasu zadataka za koje je kriterijum efikasnosti vreme prelaza, tj. na klasu sistema optimalnih po brzini dejstva. Smatraćemo glavnim ograničenjima, ograničenja brzine i ubrzanja upravljane veličine, kao što je važno za analizirani zadatak optimalnog upravljanja kretanjem voza. Ista takva ograničenja sreću se u zadacima optimalnog upravljanja liftovima, pratećim sistemima, rudarskim dizalicama i u mnogim drugim.

Napomena: Izopovršina je površina na kojoj leže tačke prostora (stanja) za koje neka veličina (npr. J) ima istu vrednost.

Pokušajmo da izdvojimo u faznom prostoru oblast, iz čije se svake tačke može preći u određenu tačku a_0 pri određenoj vrednosti kriterijuma efikasnosti $J = \bar{J}$.

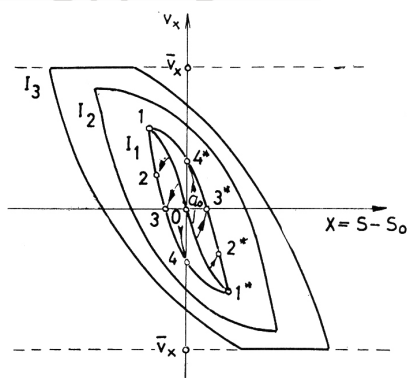
Radi toga pretpostavimo, da se u početnom trenutku ($t = 0$) reprezentativna tačka nalazi u položaju a_0 , a da vreme teče unazad, i ispitajmo na taj način »istoriju« kretanja reprezentativne tačke ka zadatom stanju. Možemo zamisliti, da posmatramo kretanje reprezentativne tačke snimljeno na filmskoj traci pri obrnutom smeru obrtanja pogona kinoprojektora. Pošto već znamo, da optimalno kretanje nastaje pri graničnim vrednostima ograničenih veličina, težićemo ispunjenju tog uslova u svim etapama kretanja sistema. Neka se tačka a_0 nalazi u koordinatnom početku faznog prostora (u razmatranom slučaju fazni prostor je ravan sa osama $S - S_0$ i $\mathcal{G}$) kao što je pokazano na sl. 8.6. Pošto se optimalni prelaz u tu tačku može završiti samo pri kretanju s graničnom vrednošću usporenja $-u$ (v. sl. 8.4), smatraćemo da je u prvoj etapi »obratnog« kretanja ubrzanje sistema jednako u u toku vremena T_1 . Tada će reprezentativna tačka doći u tačku 1 ili 1^* po trajektoriji $0-1$, ili $0-1^*$ (sl. 8.6). Ali kretanje sistema je moglo da se ne odvija sve vreme T_1 sa usporenjem $-u$, već samo u nekom njegovom delu t_3 . U ostalom vremenu $T_1 - t_3$ sistem je mogao da se kreće s graničnim ubrzanjem $+u$. Tada, posmatrajući kretanje u oba intervala, čija je suma jednaka T_1 , sa ubrzanjima $+u$ i $-u$, nalazimo tačke 2, 2^* , 3, 3^* , ... , u koje dospeva reprezentativna tačka. Postavljajući dovoljan broj takvih tačaka, može se s potrebnom tačnošću odrediti linija I_1 , na kojoj leže te tačke tako, da se iz bilo koje tačke sa te linije može po optimalnoj trajektoriji preći u tačku a_0 za vreme T_1 . Dokazano je, da se iz bilo koje tačke unutar granice I_1 može preći u tačku a_0 za vreme jednako T_1 , ne narušavajući ograničenja postavljena sistemu; ali, ni iz jedne tačke van granica oblasti, koja leži unutar linije I_1 , takav prelaz nije moguć. Oblast, koja leži unutar granice I_1 konstruisane na taj način, naziva se *oblast izohrona* za vreme T_1 u odnosu na pol a_0 .

Zadajući vreme prelaza $T_2 > T_1$; može se na isti način postaviti granica I_2 oblasti izohrona sa vremenom prelaza T_2 . Pri dovoljno velikim vrednostima T počinje da se ispoljava uticaj ograničenja brzine $\mathcal{G}$ i granice oblasti izohrona se deformišu, kao što se vidi iz oblika izohrone I_3 , na sl. 8.6.

Napomena: Izohron — znači istovremen, koji traje podjednako dugo. Izohrona je linija koja spaja tačke u koje se iz jedne tačke može za isto vreme stići.

U toku konstruisanja granica oblasti izohrona svaki put je bio dat određeni program upravljačkih dejstava, koja dovode reprezentativnu tačku na granicu. Jasno je, da ti isti programi pri obratnom smeru proticanja vremena predstavljaju program optimalnog prelaza sistema iz te tačke granice u pol a_0 . Tako da, ako su konstruisane granice oblasti izohrona, samim tim su nadene i forme procesa optimalnog upravljanja sistemom.

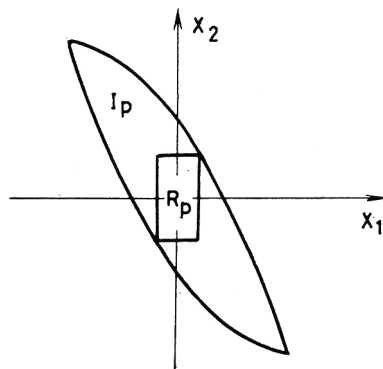
Osim toga, postavljanje granica oblasti izohrona omogućuje da se reše i drugi zadaci optimalnog upravljanja. Na primer, za datu oblast mogućih početnih stanja sistema može se odrediti ostvarljiva brzina dejstva T_p na osnovu oblasti izohrona I_p , u kojoj je upisana



Sl. 8.6. Izohrone

oblast R_p mogućih početnih stanja (sl. 8.7). Očigledno je da se prelaz u pol, koji se nalazi u koordinatnom početku, može izvršiti za vreme T_p iz bilo koje tačke oblasti R_p , ako R_p leži potpuno unutar I_p .

Mogu se zamisliti, razume se, i druge izopovršine koje izdvajaju u prostoru stanja sistema oblasti, koje ne karakterišu određene vrednosti vremena T , već nekog drugog kriterijuma efikasnosti J , a koje takođe mogu biti korisne za nalaženje optimalnog upravljanja, ili za određivanje graničnih dostižnih vrednosti J za različite prelaze sistema iz početnog stanja u zadato.



Sl. 8.7. Određivanje brzine dejstva sistema prema oblasti početnih stanja R_p i izohroni I_p

Sistemi optimalnog upravljanja

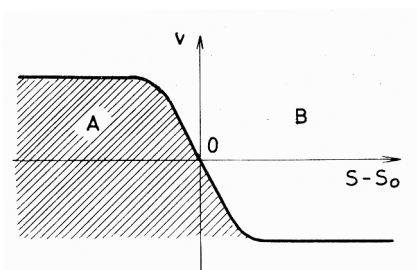
Izučavanje zakona optimalnog upravljanja stvorilo je uslove za automatsko ostvarenje optimalnih procesa u tehničkim sistemima, s ciljem poboljšanja njihovih tehničkih i ekonomskih pokazatelja. Sistemi optimalnog upravljanja našli su primenu u pratećim sistemima s brzim dejstvovanjem, u sistemima upravljanja električnim pogonima, u energetici, u saobraćaju i u raketnoj tehnici.

Prilikom realizacije sistema optimalnog upravljanja postavlja se zadatak: izabrati strukturu i parametre upravljačkog uređaja tako, da se u stvarnim uslovima rada sistema procesi njegovog kretanja odvijaju na optimalan način ili, u krajnjoj liniji, da su bliski optimalnim. Razmotrimo neke načine rešavanja takvih zadataka.

Ako sistem radi u režimu fiksiranih prelaza, kada tačka, koja prikazuje njegovo stanje, treba da pređe iz određenih početnih u određena zadata stanja, programi optimalnog upravljanja mogu biti unapred izabrani i fiksirani u memoriji upravljačkog uređaja. Pri pojavi potrebe za određenim prelazom, odgovarajući program se izvlači iz memorije i ostvaruje u vidu optimalnog niza upravljačkih dejstava. Takvi sistemi upravljanja naveliko se primenjuju za upravljanje teretnim i putničkim dizalicama, bagerima jaruzarima, električnim pogonima reverzibilnih mašina za valjanje metala i za upravljanje zaletom i kočenjem električnog vozila.

Ako su zahtevani prelazi sistema unapred nepoznati, tada se zadatak ostvarenja optimalnih sistema ne može rešiti korišćenjem krutih programa upravljanja, i upravljačka dejstva treba da se stvaraju u toku kretanja sistema. Pretpostavimo da je potrebno ostvariti optimalan po brzini dejstva sistem upravljanja kretanjem voza u uslovima opisanim ranije, ili hidrauličnog pogona, ili bilo kog drugog sistema, u kome su glavna ograničenja u pogledu brzine promene i ubrzanja upravljane veličine. Skup optimalnih procesa u takvim sistemima je pokazan na sl. 8.4, iz koje se vidi, da je za ostvarenje optimalnog prelaza sistema iz nekog početnog njegovog stanja u zadato ravnotežno stanje potrebno i dovoljno da

upravljačka dejstva odgovaraju dijagramu, prikazanom na sl. 8.8. Tu je fazni prostor podeljen u dve oblasti: A i B. U bilo kojoj tački oblasti A upravljačko dejstvo treba da se izabere tako, da ubrzanje bude $+\bar{u}$, a u oblasti B ono treba da bude $-\bar{u}$. Granica koja deli oblasti A i B naziva se *linija preključivanja*.



Sl. 8.8 Linija preključivanja upravljačkih dejstava

Sada se zadatak ostvarivanja optimalnog upravljanja sveo na određivanje oblasti (A ili B) u kojoj se nalazi tačka, koja prikazuje stanje upravljanog sistema, pošto pripadanje reprezentativne tačke jednoj od oblasti, razdvojenih linijom preključivanja, jednoznačno određuje vrednost upravljačkog dejstva.

Iz dijagrama skupa optimalnih procesa mogu se odrediti položaj i oblik, a u skladu s tim, i jednačina linije preključivanja. Za razmatranu klasu sistema linija preključivanja, kao što se vidi sa sl. 8.4, sastoji se iz delova kvadratnih parabola, koje prolaze kroz koordinatni početak. Njena jednačina može se napisati u obliku

$$X_1 + c \cdot X_2^2 \operatorname{sgn} X_2 = 0, \quad (8.7)$$

gde je c — koeficijent, koji zavisi od granične vrednosti ubrzanja $\bar{u}$ upravljane veličine X_1 , a znakom $\operatorname{sgn} X_2$ označena je funkcija promenljive X_2 , koja prima vrednost

$$\operatorname{sgn} = \begin{cases} +1, & X_2 > 0, \\ -1, & X_2 < 0. \end{cases}$$

Ako su vrednosti X_1 i X_2 takve da tačka, koja prikazuje stanje sistema, ne leži na liniji preključivanja, tada je za oblast A

$$X_1 + c \cdot X_2^2 \operatorname{sgn} X_2 < 0,$$

a za oblast B

$$X_1 + c \cdot X_2^2 \operatorname{sgn} X_2 > 0.$$

Iz prethodnog izlaganja sledi da signal upravljanja u , proizveden u upravljačkom uređaju, treba da uzme vrednosti

$$\left. \begin{aligned} u &= +\bar{u}, & X_1 + c \cdot X_2^2 \operatorname{sgn} X_2 &< 0, \\ u &= -\bar{u}, & X_1 + c \cdot X_2^2 \operatorname{sgn} X_2 &> 0. \end{aligned} \right\} \quad (8.8)$$

Uslovi (8.8) izražavaju algoritam upravljanja, koji realizuje optimalne procese prelaza u sistemima analizirane klase.

Taj algoritam se može realizovati linearnim pretvaračem *LP*, nelinearnim pretvaračem *NP* i relejnim izvršnim elementom *RIE*, koji su povezani prema šemi na sl. 8.9.

Elementi tog sistema ostvaruju sledeća pretvaranja:

$$\left. \begin{aligned} X_{c1} &= k \cdot X_1, \\ X_{c2} &= k \cdot c \cdot X_2^2 \operatorname{sgn} X_2, \\ u &= -\bar{u} \operatorname{sgn}(X_{c1} + X_{c2}). \end{aligned} \right\} \quad (8.9)$$

Nije teško proveriti da upravljački uređaj sistema upravljanja, dat na sl. 8.9, sa karakteristikama pretvarača koje odgovaraju relacijama (8.9), ostvaruje algoritam upravljanja (8.8).

Radi ostvarenja sistema optimalnog upravljanja složenijim objektima (objektima višeg reda) i za složenije kriterijume efikasnosti potrebni su, u skladu s tim, složeniji upravljački uređaji.

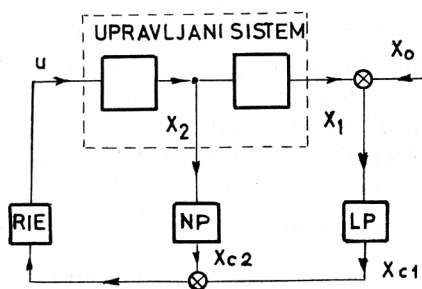
Ideje i metode optimalnog upravljanja pokazale su se veoma plodonosne za razvoj teorije i tehnike upravljanja. One su omogućile da se shvati uloga ograničenja, koja u stvarnim sistemima uvek postoje i u suštini određuju ostvarljive pokazatelje rada upravljanih sistema, da se preciznije formulišu ciljevi i zadaci sinteze upravljačkih uređaja, i da se bolje izraze u teoriji upravljanja stvarni uslovi rada upravljanih sistema.

Teorija optimalnog upravljanja dostigla je danas visok stepen razvoja. Sada umemo, ne samo

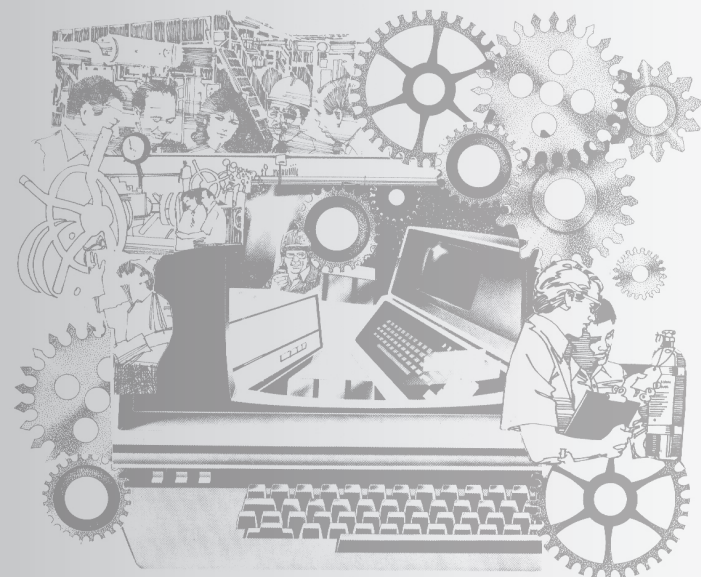
da konstruišemo optimalne sisteme upravljanja mašinama, već i da rešavamo složenije zadatke. Tako na primer, razrađeni su principi optimalnog upravljanja prostornom raspodelom stanja u telima, što je otkrilo mogućnosti, na primer, da se pronađe najpovoljniji režim zagrevanja metalnih polufabrikata i drugih masivnih tela, kada se radi o prelazu sa početne raspodele temperature u telu na zadatu raspodelu. Te metode su korisne za rešavanje zadataka upravljanja vremenom, kada je potrebno stvoriti nužnu raspodelu temperature, vlažnosti i pritiska u atmosferi.

Dobijeni su važni naučni rezultati u teoriji optimalnog upravljanja sistemima, koji se nalaze pod uticajem slučajnih poremećaja, što omogućava rešavanje zadataka projektovanja optimalnih filtara, kojima će se obezbediti najbolje izdvajanje korisnog signala od šuma, kao i ostvarenje efikasnog upravljanja objektima pri nepotpunoj informaciji o njihovim osobinama, i sl.

Pošto su se, kao rezultat evolucionog razvoja, upravljani sistemi živih organizama neprekidno usavršavali, i pošto izvršavaju svoje funkcije na optimalan način, teorija optimalnog upravljanja olakšava nam da shvatimo strukturu i osobine tih prirodnih mehanizama.



Sl. 8.9. Šema po kojoj se realizuje algoritam optimalnog upravljanja



9. AUTOMATI

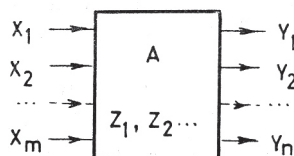
U tehnici termin »automat« se koristi za označavanje takvog sistema mehanizama i uređaja, da se procesi dobijanja, pretvaranja, prenosa i korišćenja energije, materije i informacije, neophodni za izvršenje njegovih funkcija, ostvaruju bez neposrednog učesća čoveka. Sistemi tog tipa su: mašine alatke — automati, automati za snimanje i izradu fotografije, trgovački automati i mnogi drugi.

U kibernetiku je, međutim, ušao i u njoj je potpuno prihvaćen termin »diskretni automat« ili jednostavno kraće »automat« za označavanje daleko apstraktnijeg pojma, zapravo - modela sa sledećim osobinama:

- a) na ulaze modela u svakom diskretnom trenutku $t_1, t_2, \dots$ dolaze m veličina $x_1, x_2, \dots, x_m$, od kojih svaka može uzeti samo konačan broj određenih vrednosti iz ulazne azbuke X ;
- b) na izlazima modela mogu se uočiti n izlaznih veličina $y_1, y_2, \dots, y_n$, od kojih svaka može primiti samo konačni broj određenih vrednosti iz izlazne azbuke Y ;
- c) u svakom trenutku model može da se nalazi u jednom od stanja $z_1, z_2, \dots, z_N$;
- d) stanje modela u svakom momentu određuju ulazna veličina x u tom momentu i stanje z u prethodnom momentu;
- e) model ostvaruje preslikavanje situacije na ulazu $x = \{x_1, x_2, \dots, x_m\}$ u situaciju na izlazu $y = \{y_1, y_2, \dots, y_n\}$ u zavisnosti od svog stanja u prethodnom trenutku.

Takav model (sl. 9.1) je prikladan za opisivanje mnogih kibernetičkih sistema.

Automati, kod kojih situaciju y na izlazima jednoznačno određuje situacija x na ulazima, pripadaju klasi *automata bez memorije*. Automati, kod kojih y ne zavisi samo od vrednosti x u datom momentu, već i od stanja modela z , određenog vrednošću x u prethodnim momentima, pripadaju klasi *automata s konačnom memorijom*.



Sl. 9.1. Diskretni automat

Napomena: Osim automata s konačnom memorijom postoje i *automati s beskonačnom memorijom*.

Ograničimo se ovde na razmatranje samo najprostijih diskretnih automata, čije se ulazne i izlazne azbuke sastoje svega iz dva slova: 0 i 1. Ovo se opravdava time da su, što je dokazano u teoriji automata, automati s tako »bednim« azbukama sposobni da rešavaju iste one zadatke koje rešavaju automati s bilo kakvim drugim azbukama.

Teorija diskretnih automata, ili, kako se još naziva, teorija relejnih uređaja, stekla je veliki značaj za rešavanje nekih fundamentalnih problema kibernetike, povezanih sa principijelnim mogućnostima prerade informacija u kibernetiskim sistemima, a i za analizu i sintezu složenih relejnih šema, odnosno sistema automatskog upravljanja.

Logički automat

Preslikavanja ulaznih veličina u izlazne, ostvarena u diskretnim automatima bez memorije, koji rade sa binarnom azbukom, ekvivalentna su preslikavanjima u formalnoj logici. Stoga ćemo ih zvati *logičkim automatima*, a funkcije, koje opisuju preslikavanja izvršena u logičkim automatima — *logičkim funkcijama*. Matematički aparat, korišćen za rešavanje zadataka analize i sinteze logičkih automata, je *algebra logike*. Istorijski posmatrano, prvu varijantu algebre logike razradio je engleski naučnik Džordž Bul 1843. godine, te se ona često naziva *Bulova algebra*.

Svaki izlaz y_i logičkog automata može primiti vrednost 0 ili 1 u zavisnosti od vrednosti ulaznih promenljivih x . Odredimo broj svih mogućih logičkih funkcija preslikavanja x u y , ako je broj ulaznih veličina jednak m , a svaka od njih može uzeti vrednost 0 ili 1. Radi toga ćemo rasporediti sve ulazne veličine u niz $x_1, x_2, \dots, x_m$ i posmatraćemo ih kao razrede binarnog broja. Jasno je, da je broj r različitih kombinacija vrednosti ulaznih veličina jednak broju različitih binarnih brojeva, sastavljenih iz r razreda. Odatle sledi, da je $n = 2^m$. Ali, svakoj od r situacija na ulazu odgovara jedna od dve vrednosti izlaza 0 ili 1. Stoga je ukupan broj N svih *različitih* logičkih funkcija logičkog automata sa m binarnih ulaza jednak

$$N=2^r=2^{(2^m)}. \quad (9.1)$$

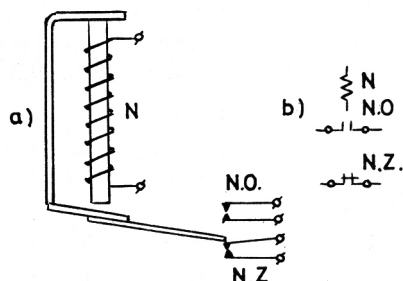
Logičke funkcije se zasnivaju na nekim elementarnim logičkim funkcijama. Koristićemo tri elementarne logičke funkcije:

1. $\bar{x}$ — *negacija promenljive x* (čita se »ne x «). Funkcija negacije označava da je $\bar{x} = 0$ ako je $x = 1$, i $\bar{x} = 1$ ako je $x = 0$.
2. $x_1 \cdot x_2$ — *logičko množenje* ili konjunkcija (čita se » x_1 i x_2 «). Funkcija logičkog množenja označava da je njegov rezultat jednak jedinici samo tada, kada su i $x_1 = 1$ i $x_2 = 1$, a jednak nuli u svim ostalim slučajevima.
3. $x_1 \vee x_2$ — *logičko sabiranje* ili disjunkcija (čita se » x_1 ili x_2 «). Funkcija logičkog sabiranja označava da je rezultat jednak nuli samo onda, kada su $x_1 = 0$ i $x_2 = 0$, i da je jednak jedinici u svim ostalim slučajevima.

Logičke funkcije se mogu zadavati tablicama u kojima su prikazane vrednosti funkcije y (indeks i ćemo izostavljati) za sve kombinacije argumenata x . U tabeli 9.1 date su vrednosti za dve elementarne logičke funkcije dvaju argumenata: x_1 i x_2 . Tu tablicu treba čitati po redovima: »ako je $x_1 = \dots$, a $x_2 = \dots$, tada je $x_1 \cdot x_2 = \dots$, a $x_1 \vee x_2 = \dots$ «.

Tabela 9.1

x_1	x_2	$x_1 \cdot x_2$	$x_1 \vee x_2$
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1



Sl. 9.2. Element relejnih šema

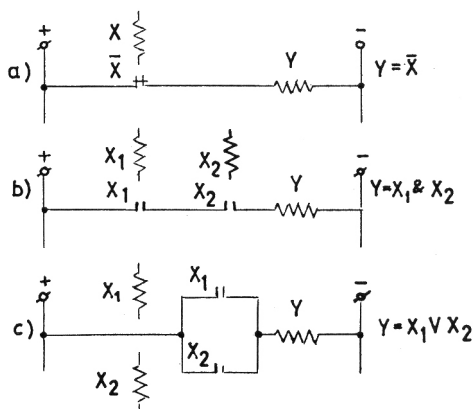
Pomoću elementarnih logičkih funkcija mogu se stvarati logičke funkcije, koje opisuju osobine raznih logičkih automata, uključujući relejne šeme.

Elementi relejnih šema su električni releji. Ulazni organ relea je njegov namotaj N (sl. 9.2), a izlazni organ — kontakti dvaju tipova: *normalno-otvoreni n.o* (otvoreni pri nepostojanju struje u namotaju, i zatvoreni pri proticanju struje) i *normalno-zatvoreni n.z* (zatvoreni u odsustvu struje u namotaju, i otvoreni pri postojanju struje). Uslovne oznake organa relea u relejnim šemama pokazane su na sl. 9.2, b.

Smatraćemo da su struja u namotaju relea i stanje kontakta logičke promenljive, koje mogu primiti vrednosti 0 i 1. Logičke promenljive, koje karakterišu stanje jednog relea, označavaćemo istim simbolima sa istim indeksima. Tako na primer, ako se stanje namotaja i -tog relea označi sa x_i , tada će se stanje njegovih n.o. kontakta takođe označavati sa x_i , a stanje n.z. kontakta — sa $\bar{x}_i$.

Na sl. 9.3. pokazane su relejne šeme za ostvarivanje pretvaranja, koja odgovaraju elementarnim logičkim funkcijama: negacija — sl. 9.3, a, logičko množenje — sl. 9.3, b i logičko sabiranje — sl. 9.3, v.

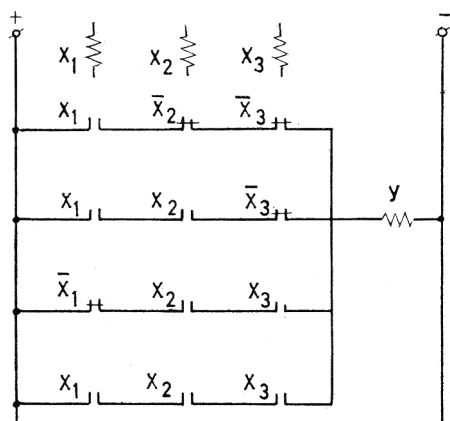
Koristeći metode teorije relejnih uređaja može se ostvariti sinteza relejne šeme, koja realizuje traženu logičku funkciju. Neka je, na primer, potrebno postaviti relejnu šemu za funkciju pretvaranja $y = F(x_1, x_2, x_3)$ zadatu tabelom 9.2.



Sl. 9.3. Relejne šeme koje realizuju elementarne logičke funkcije

Tabela 9.2.

x_1	x_2	x_3	y
0	0	0	0
1	0	0	1
0	1	0	0
0	0	1	0
1	1	0	1
1	0	1	0
0	1	1	1
1	1	1	1



Sl. 9.4. Šema koja realizuje logičku funkciju (9.2)

Prepisujući uslove (redove) za koje je $y=1$ i objedinjavajući te uslove logičkim sabiranjem (disjunkcijom), dobijamo izraz za funkciju F u takozvanoj normalnoj disjunktivnoj formi

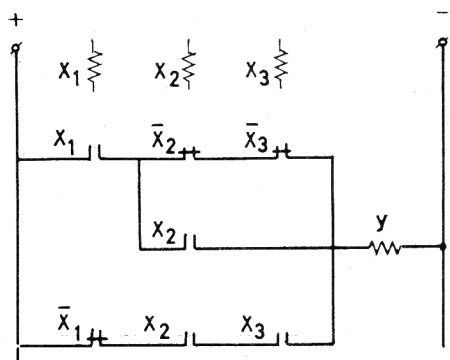
$$y = (x_1 \cdot x_2 \cdot x_3) \vee (x_1 \cdot x_2 \cdot \bar{x}_3) \vee (\bar{x}_1 \cdot x_2 \cdot x_3) \vee (x_1 \cdot x_2 \cdot x_3) \quad (9.2)$$

Izraz (9.2) pokazuje da y mora biti jednako 1, ako su: $x_1=1$ i $x_2=0$ i $x_3=0$ ili $x_1=1$ i $x_2=1$ i $x_3=0$ ili . . . , kako sledi iz tabele 9.2. Na osnovu izraza (9.2) može se postaviti takva šema (sl. 9.4) da svakom redu odgovara grana sastavljena od redno spojenih kontakata (n.o. za simbole bez negacije i n.z. za simbole s negacijom), a celom izrazu odgovara paralelna veza sve četiri grane.

Lako je proveriti da se takvom šemom ostvaruje preslikavanje dato tabelom 9.2. Međutim, osim te šeme postoji neograničeno mnoštvo drugih šema, koje takođe realizuju zadanu funkciju preslikavanja. Jasno je, da šema dobijena neposredno iz normalne disjunktivne forme ne mora biti najbolja po broju elemenata ili po pouzdanosti da će sistem nesmetano raditi i sl. U datom slučaju šema se može uprostiti bez promena u funkciji koju realizuje, kao što je pokazano na sl. 9.5.

Pri tome je, kao što se vidi sa šeme, broj upotrebljenih kontakata smanjen od 12 na 7.

Logički automat se može izvesti i korišćenjem beskontaktnih elemenata (elektronski sklopovi).



Sl. 9.5. Jednostavnija šema ekvivalentna šemi na sl. 9.4

Automat sa konačnom memorijom

Prilikom proučavanja automata s konačnom memorijom, obično se analiziraju samo ustaljena stanja, u kojima se oni nalaze posle dovoljno dugog vremena nakon promene ulaznih dejstava. U ovom izlaganju se smatra da procesi prelaza sistema iz jednog ustaljenog stanja u drugo protiču dovoljno brzo u odnosu na dužinu vremenskih intervala između promena ulaznih dejstava. Stoga je podesno posmatrati ponašanje automata s konačnom memorijom u diskretnim trenucima $t_1, t_2, \dots$, međusobno odvojenih intervalima Δt . Uz to ćemo pretpostaviti da se i izlazna dejstva menjaju samo u momentima $t_1, t_2, \dots$, koji se nazivaju taktovi (trenuci odabiranja, semplovanja).

U skladu s definicijom izlaz automata s konačnom memorijom u j -tom taktu zavisi od stanja automata u $(j - 1)$ -om taktu i stanja ulaza u j -tom taktu. Stoga se prelazi takvog automata iz jednog stanja u drugo, u opštem obliku, opisuju izrazima

$$\left. \begin{aligned} y^j &= F(z^{j-1}, x^j), \\ z^j &= G(z^{j-1}, x^j), \end{aligned} \right\} \quad (9.3)$$

gde su y^j — izlaz automata u j -tom taktu, z^{j-1} — stanje automata u $(j - 1)$ -om taktu,

$$x^j = (x_1^j, x_2^j, \dots, x_m^j)$$

— ulaz automata u j -tom taktu, F i G — neke logičke funkcije (funkcija izlaza i funkcija prelaza) stanja i ulaza.

Da bi automat ostvario pretvaranje 9.3 neophodno je da, osim elemenata koji realizuju logičke funkcije, sadrži i *element kašnjenja*, čiji je izlaz određen vrednošću njegovog stanja u prethodnom taktu, tj. element, čiji je izlaz y povezan sa ulazom x izrazom

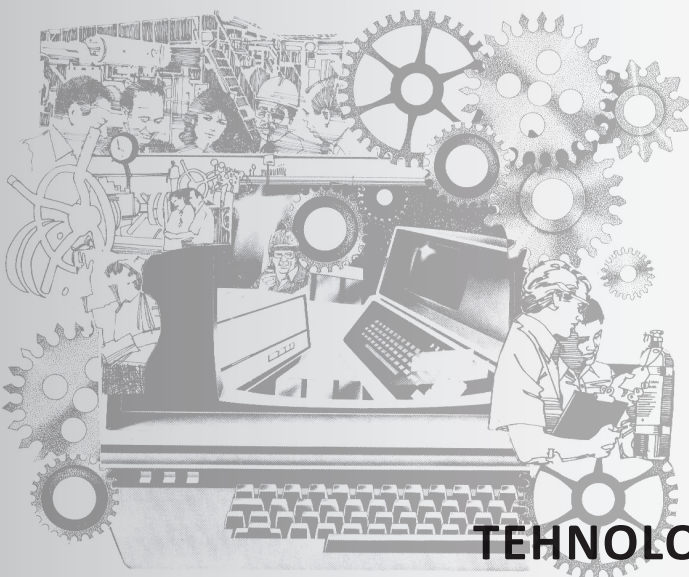
$$y^j = f(z^{j-1}),$$

ili u posebnom slučaju

$$y^j = z^{j-1}.$$

Element kašnjenja treba da ima memoriju; u njoj mora da se čuva trag prethodnog stanja, inače njegovo stanje ne bi moglo da zavisi od prethodnog stanja. Jedan od rasprostranjenih diskretnih elemenata sa memorijom je *okidno kolo*, koje predstavlja uređaj sa dva stabilna stanja. Taj uređaj može da prelazi iz jednog od tih stanja u drugo pod uticajem signala upravljanja. Okidno kolo se može izvesti sa raznim konstruktivnim elementima, a posebno sa elektronskim elementima.



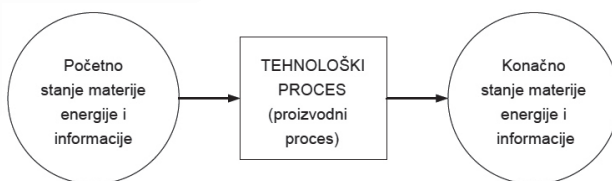


10. UPRAVLJANJE TEHNOLOŠKIM PROCESIMA

Tehnološki proces je kategorija realnih, organizovanih, dinamičkih sistema u kojima se odvijaju oblikovne, sadržajne, fizičko-hemijske i energetske modifikacije ulaznih tokova materije energije i informacije u jednoj, dve ili sve tri prostorne dimenzije.

Tehnološki procesi se odvijaju u postrojenjima za proizvodnju, distribuciju i preradu gasa, nafte, električne i toplotne energije, u postrojenjima za proizvodnju i preradu metala i nemetala, u pogonima za proizvodnju hemijskih preparata, lekova, hrane, keramike, obuće, odevnih predmeta itd. U svakom od navedenih tehnoloških procesa ostvaruje se ulazni tok materije, energije i informacija, izvrši se odgovarajuća promena u cilju stvaranja izlaznih tokova materije, energije i informacija u vidu proizvoda, poluproizvoda i nusproizvoda.

Promene u tehnološkim procesima odvijaju se u skladu sa opštim zakonitostima odvijanja prirodnih pojava, uz ograničenja koja određuju ulazni tokovi materije i/ili energije i konstruktivne odlike materijalne osnove proizvodnog procesa i njegovih veza sa okruženjem. Kod velikog broja realnih tehnoloških procesa u prirodnoj interakciji sa okruženjem ne postiže se željeni nivo organizovanosti i funkcionalnosti, niti se ostvaruje racionalno korišćenje materijalnih, energetskih, finansijskih i ekoloških resursa. Razloge za odstupanja između željenog i postignutog treba tražiti u zakonitostima delovanja samog procesa, konstruktivnim odlikama primenjene aparature, prisustvu raznih poremećaja, sporosti i spontanosti reakcija itd.



Sl. 10.1 Tehnološki proces kao događanje koje izaziva promenu stanja

Da bi se sprečile spontane promene uslova odvijanja proizvodnje, i umanjile eventualne težnje ka dezorganizovanosti, na tokove materije, energije i informacije između proizvodnog procesa i okruženja treba svesno i promišljeno delovati tj. proizvodnim procesima se mora upravljati.

U inženjerskoj praksi je odomaćena podela upravljačkih zadataka na sledeće kategorije:

- ♦ upravljanje,
- ♦ regulacija,
- ♦ vođenje i
- ♦ nadzor.

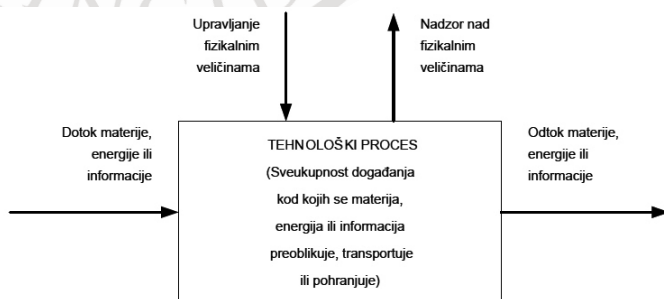
Upravljanje predstavlja skup aktivnosti kojima se interventne akcije izvode na osnovu odgovarajućih zakonitosti sa ciljem da se postigne odgovarajuće uzročno-posledično odvijanje transformacije materije, energije i informacija. Upravljanje se ostvaruje u otvorenoj sprezi. Razlikujemo logičko, logičko-sekvencijalno, programsko, koračno itd. upravljanje. Kod logičkog upravljanja, logičke vrednosti ulaza izražene logičkim vrednostima (0 i 1) primenom elementarnih logičkih zakonitosti se pretvaraju u izlaze. Kod logičko-sekvencijalnog upravljanja, primenom memorijskih i vremenskih članova, ostvaruje se logičko upravljanje na osnovu logičkih ulaza koji nastaju i u drugim trenucima vremena. Kod programskog upravljanja faze i zakonitosti formiranja izlaza smenjuju se saglasno unapred postavljenom programu. Koračno upravljanje predstavlja vid upravljanja u kom prethodna faza (korak) odvijanja procesa određuje uslove za odvijanje naredne faze.

Regulacija ima zadatak da izjednači odabrane procesne signale (zadate, upravljane veličine). U osnovnoj podeli razlikujemo stabilizirajuću i prateću regulaciju. Zadatak stabilizirajuće regulacije je održavanje odabranih upravljanih veličina na unapred zadatoj vrednosti. Prateća regulacija treba da obezbedi uslove praćenja upravljane veličine u skladu sa nekom zadatom zakonitošću. Regulacija se uvek odvija u zatvorenoj sprezi.

Vođenje je aktivnost koja inicira, održava ili menja neki proces.

Nadzor je aktivnost koja na osnovu informacija iz upravljanog sistema izveštava o stanjima upravljanog procesa. Poseban zadatak nadzora je alarmiranje tj. upozoravanje na nepoželjna ili nepredviđena stanja.

Savremeni upravljački sistemi u principu istovremeno obavljaju funkcije upravljanja, regulacije, vođenja i nadzora.



Slika 10.2 Tehnološki proces kao sveukupnost događanja sa aspekta upravljanja i nadzora

Upravljački uređaj logičkog i logičko-sekvencijalnog upravljanja je automat. Automat ima konačan broj "m" ulaza i konačan broj "n" izlaza. Automat funkcioniše u diskretnim

vremenskim trenucima, koji se najčešće nazivaju taktovi. Razlikuju se sledeće klase automata:

- ♦ automat bez memorije;
- ♦ automat sa konačnom memorijom i
- ♦ automat sa beskonačnom memorijom.

Tehničko sredstvo za ostvarivanje automata je digitalni računar, koji je i sam automat. Namenski uređaj za realizaciju automata je slobodno programirajući uređaj, tj. programabilni logički automat.

Podela upravljačkih zadataka prema vremenu obuhvaćenom upravljanjem

Sa stanovišta vremena koje je obuhvaćeno upravljanjem razlikuju se:

- ♦ operativno,
- ♦ taktičko i
- ♦ strategijsko upravljanje.

Operativno upravljanje je upravljanje u realnom vremenu, sa vrlo brzim upravljačkim odlučivanjima i reagovanjem, u skladu sa promenama u okruženju i događajima koji se odvijaju u upravljačkom sistemu. To je najkonkretniji vid upravljanja i zasniva se na obradi informacija koje se prikupljaju u realnom vremenu. Operativno upravljanje predstavlja izvršni nivo upravljanja.

Taktičko upravljanje se uglavnom bavi optimizacijom i upravljanjem stacionarnim stanjima više zajednički delujućih procesa.

Strategijsko upravljanje se bavi dugoročnim problemima ostvarivanja upravljanja složenim tehnološkim procesima.



Sl. 10.3 Vremenski nivoi upravljanja

Postupak određivanja zadataka upravljanja

Upravljački sistemi ostvaruju samo one funkcije koje su predviđene i ugrađene u upravljačke sisteme.

Određivanje zadatka upravljanja je uvodna radnja u formiranju upravljačkog sistema i predstavlja zajednički rad tehnologa, konstruktora opreme i stručnjaka za upravljačku tehniku. Pri tom se obavezno razmatraju sledeća pitanja:

1. Koje veličine su značajne sa stanovišta upravljanja i nadzora. Pri tom se pod veličinom podrazumeva: pritisak u cevovodu, temperatura destilacione kolone, nivo u rezervoaru, protok, pH vrednost, vlažnost, itd.
2. Koja vrsta obrade se može primeniti nad odabranim veličinama. Pri tom se pod obradom podrazumeva: prikaz, memorisanje, registracija, alarmiranje, indikacija, ručno ili automatsko upravljanje, upravljanje u otvorenoj i zatvorenoj sprezi itd.

3. Nad kojim veličinama i sa kojom opremom se mogu realizovati izvršna dejstva.
4. U koje tačke tehnološke opreme se mogu montirati davači i izvršni elementi da se osnovni tehnološki proces što manje remeti. Na primer, mora se odrediti gde se u cevovodu može postaviti merač pritiska i ventil, u kojoj platformi destilacione kolone treba meriti temperaturu itd.
5. Gde se moraju smestiti uređaji za opsluživanje, ili obradu upravljačkih informacija i uređaji za akviziciju.

Zadatak se definiše u gruboj formi u *Tehnološkoj šemi*, a zatim se nakon analiza i sinteza upravljačkih algoritama, razrađuje u šemi instrumentacije i cevovoda, funkcionalnom dijagramu toka, projektu instalacija i montaže itd.

Razrada se u daljem toku ostvaruje u timskom radu tehnologa, konstruktora i stručnjaka za upravljačku tehniku. Razrada prvenstveno obuhvata preispitivanje postavljenih zadataka sa stanovišta ostvarljivosti. Odgovori se traže rešavanjem pitanja sledećeg tipa: da li je upravljanje pritiskom u jednoj destilacionoj koloni bolje ostvariti putem menjanja protoka pare ili protoka rashladnog vazduha itd.

U sledećem koraku se vrši izbor odgovarajuće opreme. Pri tom se određuju merna područja, opsezi izvršnih dejstava, ocenjuju efekti ambijentalnih uslova kao što su: prisustvo eksplozivnih smesa, smetnji, vlage, prašine, elektromagnetnih smetnji, način ugradnje, vrsta potrebne pomoćne energije, itd. Aktivnosti u ovom postupku se ostvaruju uz maksimalno oslanjanje na savremene računarike podržane metode kao što su: CAD-računarsko projektovanje, CAQ-računarska kontrola kvaliteta, CAL-računarom podržana logistika, CAO-računarom podržano kancelarijsko poslovanje, CAP-računarom podržano planiranje itd.

Tehnološke šeme

Tehnološke šeme definišu tokovi informacija u konturama upravljanja i razmene energije i materijala u tehnološkom procesu. Tehnološke šeme se formiraju od slovnih i slikovnih simbola. U daljem će se navesti neki od slovnih i slikovnih simbola koji se uobičajeno koriste pri crtanju tehnoloških šema.

Materijalna oprema za ostvarivanje zadataka upravljanja procesima obuhvata:

- ♦ opremu za formiranje informacija o stanju procesa i okruženja,
- ♦ opremu za obradu informacija,
- ♦ izvršne uređaje koji neposredno menjaju vrednost upravljano ulaza, pomoćne komponente kao što su elementi za menjanje i podešavanje vrednosti referentnih signala, preklopničke za izbor načina rada, signalne elemente za indikaciju rada, izvore pomoćne energije, osigurače itd. i
- ♦ komunikacijsku opremu za prenos informacija i funkcionalno objedinjavanje upravljačkog sistema.

Komponente materijalne opreme procesnog upravljanja se uglavnom izrađuje serijski. Proizvođači ih predstavljaju kroz kataloge i prospekte u kojima se specificiraju osnovne osobine komponenti kao što su: podaci o napajanju, podaci o ulazu, podaci o izlazu,

konstrukcione i montažne karakteristike, podaci o kabliranju, podaci o uslovima koje mora zadovoljiti ambijent u kom se oprema montira itd.

Komponente materijalne opreme procesnog upravljanja se montiraju u proces, u kontrolne ormane uz proces ili na nekom drugom, često i vrlo udaljenom mestu.

Instalacije sistema za upravljanje procesima obezbeđuju funkcionalno objedinjavanje komponenti materijalne opreme procesnog upravljanja, prenos informacija između komponenti, i snabdevanje komponenti pomoćnom energijom.

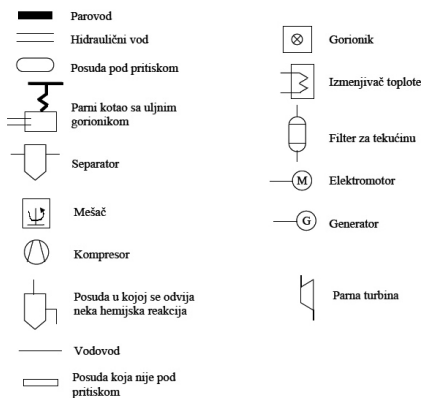
Materijalnu opremu instalacije čine provodnici, cevovodi, razvodnici, razdelnici itd. U objektima u kojima se odvijaju tehnološki procesi može se naći sledeće instalacije:

1. Instalacije vodovoda i pare,
2. Instalacije gasa,
3. Instalacije vazduha pod pritiskom,
4. Instalacije centralnog grejanja,
5. Instalacije opšteg elektroenergetskog napajanja,
6. Energetske instalacije procesne opreme,
7. Telefonske instalacije,
8. Instalacije računarskih mreža,
9. Instalacije protivpožarne zaštite,
10. Instalacije zaštite od provale,
11. Instalacije upravljačkog sistema,
12. Instalacije unutrašnje i spoljašnje gromobranske zaštite, itd.

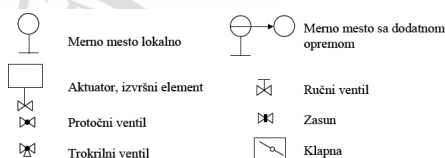
Standardima i pravilnicima su regulisani uslovi postavljanja pojedinih instalacija, kao i uslovi koji se moraju zadovoljiti da ove instalacije ne ometaju jedna drugu u toj meri da se izgubi funkcionalnost i pouzdanost sistema.

Tehnička sredstva za prikupljanje informacija

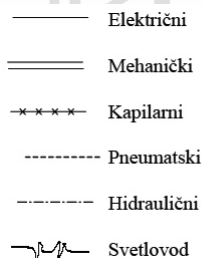
Elementarne podprocese spajaju energetske, materijalne i informacione tokove tj. procesni signali. Procesne signale karakterišu pokazatelji stanja, kao što su: masa, protok, napon, struja, pritisak, temperatura itd. Pokazatelji stanja procesnih signala su merljive veličine.



Sl. 10.4 Oznake za tehnološku opremu



Sl. 10.5 Oznake mernih i izvršnih elemenata



Sl. 10.6 Oznake prenosnih vodova

Signali su vremenske, prostorne funkcije koje pored svog fizičkog iznosa i konkretnog efekta na okruženje uvek nose sa sobom i informacije. Analizom informacionog sadržaja signala utvrđuje se stanje i ispunjenost tehnoloških zahteva u tehnološkom procesu.

U tehničkim sistemima signal je uvek fizička veličina koja je usmerena tj. ima svoj izvor i odredište. Sa stanovišta upravljanja procesima značaj uvek ima informacioni, a ne materijalno-energetski karakter signala.

Informacija je deo saznanja kojim se smanjuje ili u potpunosti odstranjuje neka neizvesnost. Npr. neizvesnost u oceni stanja nekog tehnološkog procesa može se otkloniti samo ako se poseduje određena količina informacija o procesu. Informacije koje nosi signal kvantitativno se mogu iskazati preko parametara signala. Informacije iz signala dobijaju se odabirom i obradom podataka o parametrima signala.

Podatak je svaka kvalitativno i kvantitativno izražena veličina koja je na neki način izmerena, zabeležena ili računskim operacijama određena. Neka veličina može biti podatak ako postoji određena, makar i minimalna verovatnoća, da će biti iskorišćena tj. da će biti pretvorena u informaciju. Podaci o signalima tj. stanjima tehnološkog procesa prikupljaju se posmatranjem, logičkim zaključivanjem i merenjem parametara signala.

Prenošenje, obrada i čuvanje podataka ili informacija moguća je samo pomoću signala kao nosioca informacija. Nosioc informacija, tj. signal mora bar preko jednog svog kvalitativnog ili kvantitativnog pokazatelja tj. parametra da prenosi informaciju.

Analizom informacionog sadržaja signala tj. obradom parametara signala, može se steći uvid u događaje koji se odvijaju u izvoru signala, ali se mogu predvideti i događaji koji će se odvijati u delu sistema koji će signal prihvatiti.

Uporednom analizom više signala iz sistema ili amplituda istog signala u različitim trenucima vremena, može se izvesti zaključak o stanju i parametrima sistema koji predstavlja izvor signala.

Svaka akcija upravljanja se zasniva na informacijama koje se pribavljaju merenjem. Mere se fizičke veličine koje karakterišu stanje procesa kao što su: temperatura, pritisak, protok, nivo, električna struja, položaj, brzina i sl. U upravljanju tehnološkim procesima na merenje temperature i pritiska otpada više od polovine svih merenja koja se izvode za potrebe upravljanja.

Merenja se baziraju na pretvaranju fizičkih veličina u električni signal. Pretvaranje mehaničkih, hemijskih i drugih procesih signala u električni signal vrši se pomoću različitih davača, pretvarača, transdusera, transmitera itd. U novije vreme za ove uređaje ustalio se termin senzor obzirom na visok nivo razvoja materijala i tehnologije, primene mikromašinstva i visokog stepena integracije komponenata, te velikim mogućnostima u obradi informacija primenom mikroračunara i mikrokontrolera.

Reč «senzor» je izvedena od latinske reči sentire koja znači »opisati«. Senzor daje informacije o fizičkim i hemijskim signalima koji ne bi mogli biti direktno opaženi ljudskim čulima. U savremenoj praksi senzor predstavlja »crnu kutiju» koja informacije o stanju i kvalitetu procesa pretvara u takve informacije koje se bez teškoća mogu obrađivati i prenositi na daljinu.

U zavisnosti od nivoa obrade signala u senzoru skraćemo primarne senzorske elemente ili davače, senzore sa transdjuserom (mernim pretvaračem) i inteligentne senzore.

Primarni senzorski element ili davač pretvara fizičku veličinu u oblik koji je pogodniji za merenje. Primarni senzorski elementi su otpornički davači, termoparovi itd.

Senzor sa transmitterom sadrži senzorski element i transdjuser. Transdjuser pretvara merenu fizičku veličinu u neki standardni signal. Standardni signal je najčešće električni strujni ili naponski signal, a može biti i pneumatski signal.

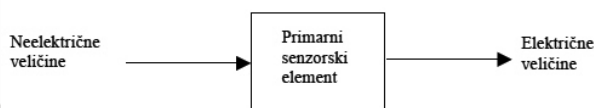
Standardni strujni signal iz transdjusera je 0 – 20mA ili 4 – 20mA. Standardni pneumatski signal je 0.02 – 0.01 MPa, a standardni naponski signal može biti 0 – 1mV, 0 – 10mV, 0 – 100mV itd.

Inteligentni senzori sadrže u sebi primarni senzorski element, mikroračunar za digitalnu obradu signala i komunikacioni podsistem. Signal iz primarnog senzorskog elementa se pojačava, konvertuje u digitalni signal, obrađuje (linearizuje, kalibriše itd.) i priprema za prenos u vidu standardnih strujnih signala ili preko računarskih komunikacionih kanala (RS 232, RS 485 itd).

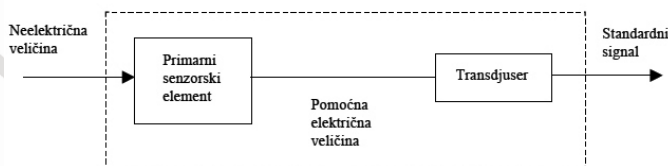
Više inteligentnih senzora se sprežu u senzorski sistem koji u principu sadrži različite primarne senzorske elemente tako da se sa jednim senzorom meri pritisak, temperatura, vlažnost, itd.

Tehnički pokazatelji statičkih svojstava senzora su neophodni za ocenu merenih kvaliteta relevantnog senzora, ali i za poređenje sa drugim senzorima, tako da se za datu aplikaciju može izabrati najpovoljnija varijanta. Međutim, pored navedenih pokazatelja, za tehničku praksu veliki značaj imaju cena nekog uređaja i cena njegovog održavanja.

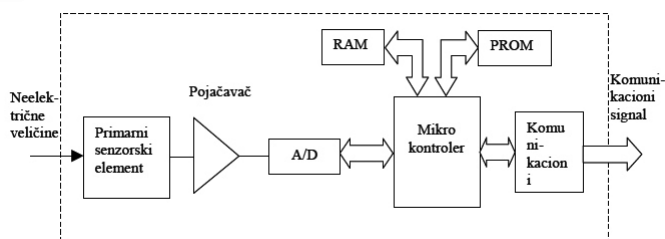
Cena senzora je u direktnoj vezi sa njegovim kvalitetom. Veća tačnost, osetljivost, rezolucija i slično mogu se dobiti samo većim novčanim ulaganjima. Zato se o investicionim sredstvima za nabavku nekog senzora može raspravljati tek nakon što su tačno specificirani pokazatelji statičke



Sl. 10.7 Primarni senzorski element



Sl. 10.8 Senzor sa transdjuserom



Sl.10.9. Inteligentni senzor

karakteristike potrebni za konkretnu aplikaciju. U tom poslu važnu ulogu imaju metrološko iskustvo, poznavanje i dostupnost tržišta, te raspoloživost proizvođačkih kataloga.

Investiciona sredstva za nabavku senzora ne određuju se samo na osnovu mernih kvaliteta. Isplati se uložiti veća sredstva za senzor sa istim mernim kvalitetima, ali sa boljim parametrima pouzdanosti i manjim troškovima održavanja. Prema tome, izbor adekvatnog senzora je izvestan kompromis između cene mernih kvaliteta, troškova pouzdanosti i održavanja, te tržišne cene.

Uvod u programabilne logičke kontrolere

Programabilni logički kontroleri (engl. *programmable logic controller*, PLC), kao onaj na slici 10.10, danas se veoma mnogo koriste u tehnologijama upravljanja industrijskim procesima. Programabilni logički kontroler je računar namenjen za upotrebu u industrijskim okruženjima, koji se može programirati tako da obavlja upravljačke funkcije. Njegovom primenom eliminisan je najveći deo žičanih veza koje su neophodne u upravljačkim kolima realizovanim pomoću releja. Druge prednosti PLC-a su: jednostavno programiranje i instaliranje, velika brzina odziva, mrežna kompatibilnost, lako otklanjanje grešaka i testiranje, te visoka pouzdanost.

Programabilni logički kontroler je projektovan tako da prihvata više različitih rasporeda ulaza i izlaza, podnosi širok opseg radnih temperatura, neosetljiv je na električni šum i otporan je na vibracije i udarce. Programi koji upravljaju proizvodnjom mašina i opreme najčešće se smeštaju u memoriju koja je trajna ili ima vlastito napajanje. PLC je primer sistema koji radi u *realnom vremenu* pošto stanje na izlazima sistema kojim upravlja PLC zavisi od stanja na ulazima tog sistema.

Prema tome, programabilni logički kontroler je u suštini digitalni računar koji je projektovan za upravljanje određenom mašinom ili postrojenjem. Za razliku od personalnog računara, PLC je projektovan da radi u industrijskom okruženju, opremljen je specijalnim ulaznim i izlaznim interfejsima i programira se pomoću posebnog jezika za

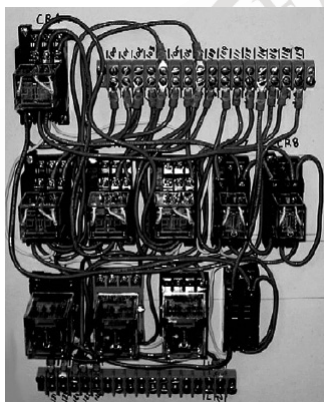


Sl. 10.10 Izgled programabilnih logičkih kontrolera

upravljanje procesima. Uobičajena skraćenica PC koja se koristi za tu vrstu uređaja u industriji, može prouzrokovati zabunu zato što je to takođe skraćenica za „personalni računar“ (engl. *personal computer*, PC). Iz tog razloga, većina proizvođača svoje programabilne logičke kontrolere naziva skraćeno PLC“.

U početku se PLC koristio kao zamena za relejnu logiku, ali ga zbog stalno rastućeg broja njegovih funkcija srećemo u sve više primena od kojih su mnoge veoma složene. Budući da se struktura PLC-a zasniva na istim principima kao što su oni koji se primenjuju u računarskoj arhitekturi, PLC može obavljati ne samo iste poslove kao relejna tehnika, nego je pogodan i za druge primene – na primer, merenje vremena, prebrojavanje, izračunavanje, poređenje i obrada analognih signala.

Programabilni kontroleri pružaju više prednosti nad tradicionalnim upravljanjem pomoću releja. Releji se moraju povezati žicama kako bi mogli da obavljaju određenu funkciju. Kada se promene zahtevi sistema, žičane veze između releja treba izmeniti ili dopuniti. U ekstremnim slučajevima, kao u automobilskoj industriji, trebalo je zameniti kompletne ploče s relejima jer nije bilo isplativo menjati ožičavanje postojećih ploča pri svakoj promeni modela vozila. PLC je učinio nepotrebnim većinu žičanih veza koje su bile neophodne u tradicionalnim relejnim kolima (slika 10.11). On je mali i jeftin u poređenju s ekvivalentnim sistemom za upravljanje procesima u relejnoj izvedbi.



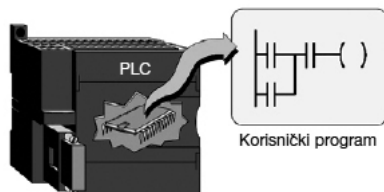
Sl. 10.11 Relejna upravljačka ploča



Sl. 10.12 PLC upravljačka ploča

Osim nižih troškova, PLC pruža i mnoge druge prednosti, među kojima su i sledeće:

- ♦ *Veća pouzdanost.* Pošto napišete i testirate program, lako ga možete instalirati na druge PLC sisteme. Budući da se celokupna logika smešta u memoriju PLC-a, ne postoji mogućnost uvođenja greške u logici zbog pogrešnog ožičavanja (slika 10.13). Program zamenjuje veći deo spoljašnjih kablovskih veza koje bi inače bile potrebne za upravljanje procesom. Direktnne kablovske veze i dalje su potrebne za povezivanje sa spoljnim uređajima, ali je njihov broj znatno manji. Pouzdanosti PLC značajno doprinose i komponente izvedene u tehnologiji integrisanih kola.



Sl. 10.13 Celokupna logika smeštena je u memoriju PLC-a

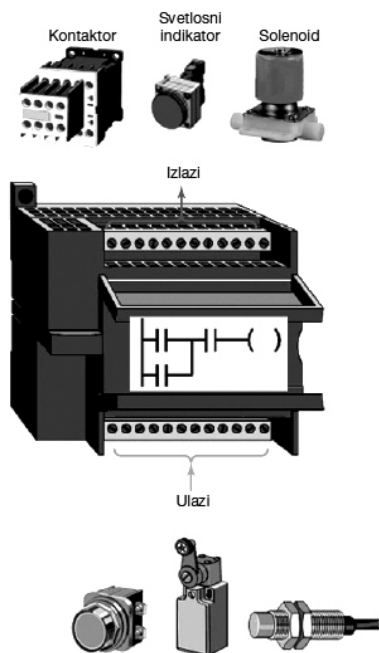
- ♦ *Veća fleksibilnost.* Lakše je napisati i izmeniti program instaliran u PLC nego menjati ožičenje električnog kola. Odnose između ulaza i izlaza programabilnog kontrolera određuje program koji piše korisnik, a ne način na koji su ti ulazi i izlazi međusobno povezani (slika 10.14). Proizvođači izvorne opreme mogu da prosleđuju korisnicima izmene tako što će im poslati nove verzije programa. Krajnji korisnici mogu da sami menjaju programe na licu mesta, a ako je poželjno, program se može zaštititi određenim hardverskim merama, kao što je zaključavanje ormara sa opremom, ili pomoću softverske lozinke.

- ♦ *Niži troškovi.* Prvobitna namena PLC-a bila je da zameni upravljačku logiku izvedenu pomoću releja, ali su uštede postale toliko značajne da se upravljanje pomoću relejne tehnike sve češće smatra zastarelim, osim kada se radi o jakim strujama i visokim naponima. Ako je za određenu aplikaciju potrebno više od pola tuceta upravljačkih releja, najverovatnije bi bilo jeftinije da se upotrebi PLC.

- ♦ *Mogućnost komuniciranja.* PLC može da komunicira s drugim kontrolerima ili drugom računarskom opremom radi obavljanja funkcija kao što su nadziranje sistema, akvizicija podataka, praćenje parametara uređaja i procesa, i instaliranje programa (slika 10.15).

- ♦ *kraće vreme odziva.* PLC je projektovan za brze aplikacije i rad u realnom vremenu. Programabilni kontroler radi u realnom vremenu, što znači da će svaki događaj na spoljnom ulaznom uređaju pokrenuti određenu operaciju na izlaznom uređaju kojim upravlja PLC. Mašine koje obrađuju hiljade elemenata u sekundi ili predmeti koji se zadržavaju samo delić sekunde ispred senzora, zahtevaju da se PLC brzo odaziva.

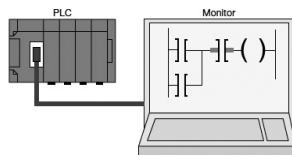
- ♦ *Lakše otklanjanje grešaka.* PLC ima ugrađene funkcije za dijagnostiku koje korisniku omogućavaju da lako prati rad sistema i ispravlja softverske i hardverske probleme. Da bi otkrili i otklonili probleme, korisnici mogu da prikažu rad upravljačkog programa na monitoru i posmatraju u realnom vremenu kako se on izvršava.



Sl. 10.14 Odnose između ulaza i izlaza PLC-a određuje program koji korisnik instalira u PLC



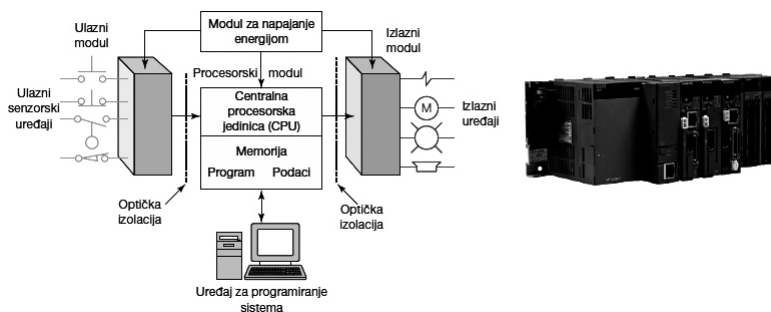
Sl. 10.15 Komunikacioni modul



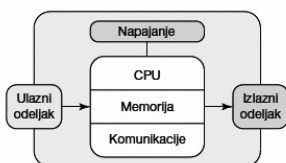
Sl. 10.16 Rad upravljačkog programa može se pratiti na monitoru u realnom vremenu

Sastavni delovi PLC sistema

Tipičan PLC sistem sastoji se od više delova, kao što je prikazano na slici 10.17. To su: *centralna procesorska jedinica* (engl. *central processing unit*, CPU), *ulazne/izlazne komponente* (engl. *input/output*, U/I), napajanje i uređaj za programiranje. Izraz *arhitektura* može se odnositi na hardver PLC-a, na softver PLC-a ili na neku njihovu kombinaciju. Sistem koji ima *otvorenu* arhitekturu lako se povezuje sa uređajima i programima drugih proizvođača. U otvorenim arhitekturama koriste se gotove komponente koje su usklađene s proverenim standardima. Sistem *zatvorene* arhitekture je onaj čiji dizajn određuje proizvođač sistema, što otežava povezivanje sa uređajima drugih proizvođača. Pošto su mnogi PLC sistemi zapravo zatvorene arhitekture, morate obavezno proveriti da li je određen generički hardver ili softver koji možda koristite kompatibilan sa određenim PLC uređajem. Osim toga, iako su glavni koncepti isti u svim metodama programiranja, u pojedinim modelima mogu postojati male razlike u adresiranju, načinu dodeljivanja memorije ili u načinu učitavanja i obrade podataka. Iz tih razloga, programi za PLC nisu međusobno zamenjivi između uređaja različitih proizvođača PLC sistema.



Sl. 10.17 Modularni tip PLC-a

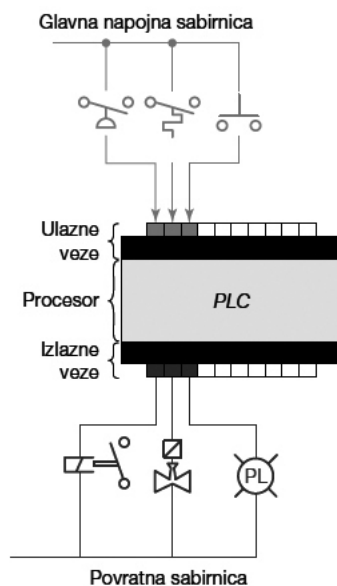


Sl. 10.18 Fiksni tip PLC-a

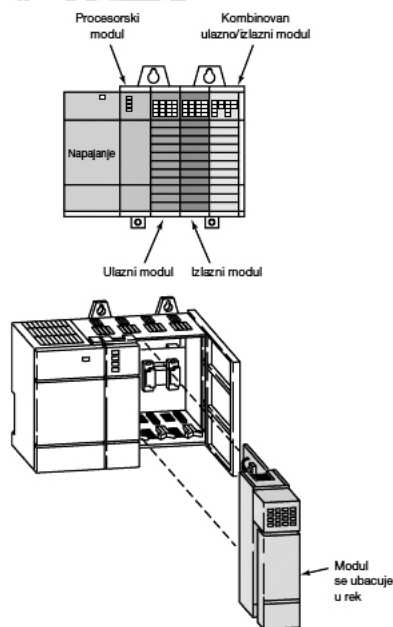
Postoje dva načina na koje se ulazi/izlazi mogu pridružiti PLC: fiksni i modularni. *Fiksni U/I* (slika 10.19) tipični su za male kontrolere koji se proizvode u kućistima na koje se ne mogu priključiti dodatne jedinice. Procesor i U/I komponente smešteni su u zajedničko kućište na kojem se nalazi fiksni broj klemata za priključivanje spoljnih ulaznih i izlaznih uređaja. Glavna prednost te vrste kućišta jeste niska cena. Broj U/I klemata na raspolaganju može se povećati kupovanjem dodatnih jedinica s fiksnim brojem U/I klemata. Jedan od nedostataka fiksnih U/I komponenata jeste nedovoljna fleksibilnost jer ste po pitanju količine i vrste priključivih uređaja ograničeni na ono što dobijete u kućištu PLC-a. Uz to, u nekim modelima, ako se pokvari bilo koji deo jedinice, morate zameniti celu jedinicu.

Modularni U/I (slika 10.20) podeljen je na odeljke u koje se mogu utaknuti zasebni moduli. Ta osobina znatno povećava vaše mogućnosti i fleksibilnost jedinice. Možete birati između više modula jednog proizvođača i kombinovati ih kako god vam odgovara. Osnovni modularni kontroler sastoji se od reka, napajanja, procesorskog modula (CPU), ulaznih/izlaznih komponenata (U/I modula) i operatorskog interfejsa koji omogućava programiranje i praćenje rada programa. Moduli su napravljeni tako da se utaknu u zajednički rek. Kada se modul utakne u rek, automatski se zatvara niz električnih kontakata, nazvanih pozadinska sabirnica (engl. *backplane*), koji se nalaze na poledini reka. Procesor ugrađen u PLC takođe je povezan na pozadinsku sabirnicu i može da komunicira sa svim modulima u reku.

Napajanje obezbeđuje jednosmernu struju za sve module koji su utaknuti u rek (slika 10.21). U sistemima s velikim brojem PLC jedinica, to napajanje najčešće ne snabdeva energijom spoljne uređaje. U opsežnijim sistemima, spoljni uređaji imaju vlastito napajanje naizmeničnom ili jednosmernom strujom. U nekim malim PLC sistemima, napajanje PLC jedinica može se upotrebiti i za pogon spoljnih uređaja u sistemu.



Sl. 10.19 Fiksna konfiguracija ulaza i izlaza

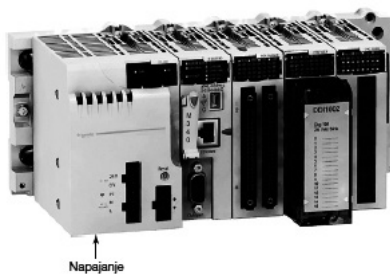


Sl. 10.20 Modularna konfiguracija ulaza i izlaza

Procesor (CPU) je „mozak“ PLC-a. Tipičan procesor (slika 10.22) sastoji se od mikroprocesora koji izvršava program i upravlja komunikacijom između modula. Procesoru je neophodna memorija gde se smeštaju rezultati logičkih operacija koje izvršava mikroprocesor. Potrebna je memorija tipa EPROM ili EEPROM u kojoj se čuva program, i memorija tipa RAM.

CPU upravlja svim aktivnostima PLC-a i projektovan je tako da korisnik može da instalira potrebne programe napisane u relejnoj lestvičastoj logici. PLC izvršava program kao postupak koji se može ponavljati a jedno izvođenje programa zove se *ciklus* (engl. *scan*) programa (slika 10.23). Tipičan PLC ciklus počinje time što CPU očitava stanje na ulazima PLC sistema. Zatim se izvršava aplikativni program. Nakon izvršavanja programa, CPU obavlja poslove interne dijagnostike i komunikacija. Zatim ažurira stanje na izlazima PLC sistema. Ceo opisani postupak neprekidno se ponavlja dok je PLC u radnom režimu.

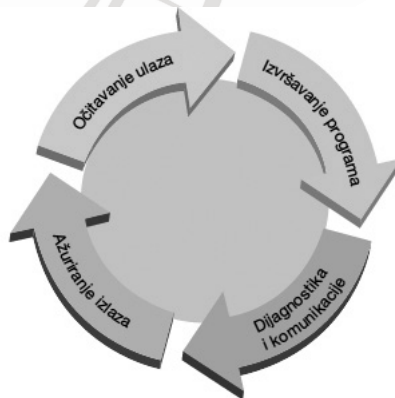
U/I sistem predstavlja interfejs kroz koji se spoljni uređaji povezuju s kontrolerom (slika 10.24). Svrha tog interfejsa je da prilagođava razne signale koje PLC prima od spoljnih uređaja ili ih šalje tim uređajima. Ulazni uređaji, kao što su tasteri, krajnji prekidači i senzori, direktno se kablovima povezuju na ulazne klemne PLC modula. Izlazni uređaji, kao što su mali motori, starteri motora, solenoidni ventili i svetlosni indikatori povezuju se kablovima na izlazne klemne PLC modula. U PLC sistemima je uobičajeno da se ulazna kola razdvajaju od izlaznih pomoću optičke izolacije. Spoljni uređaji se takođe zovu i „stvarni ulazi/izlazi“, ili „ulazi/izlazi u spoljnom svetu“. Reči „stvarni“ i „spoljni svet“ koriste se radi razlikovanja fizičkih spoljnih uređaja – koji postoje i koje treba povezati kablovima – od naredaba internog korisničkog programa koje simuliraju funkcije releja, tajmera i brojača.



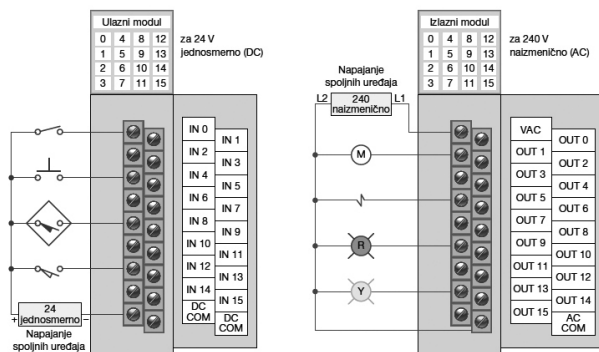
Sl. 10.21 Napajanje obezbeđuje jednostranu struju za module koji su utaknuti u reku



Sl. 10.22 Tipičan procesorski PLC modul

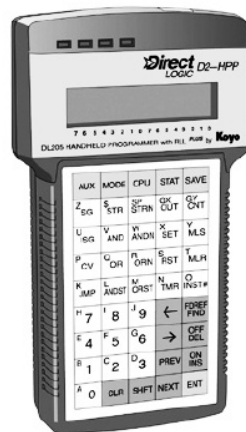


Sl. 10.23 Tipičan ciklus izvršavanja PLC programa



Sl. 10.24 Tipičan ciklus izvršavanja PLC programa

Uređaj za programiranje PLC služi za unošenje potrebnih programa u memoriju procesora. Program se može uneti u obliku relejne lestvičaste logike (engl. *relay ladder logic*, **RLL**), što je jedan od najpopularnijih jezika za programiranje. Umesto reči, u ovom programskom jeziku koriste se grafički simboli koji predstavljaju željeni konačan rezultat. Program napisan na tom jeziku sličan je šemi upravljačkog kola koje se sastoji od releja. To je specijalan jezik koji olakšava programiranje PLC-a ljudima koji poznaju relejnu logiku. Za programiranje malih PLC sistema ponekad se koriste prenosivi programatori (slika 10.25) jer su jeftini i jednostavno se koriste. Nakon priključenja na PLC, mogu se koristiti za unošenje i praćenje rada programa. U proizvodnim pogonima, i kompaktni programatori i obični prenosni računari često se koriste za otkrivanje grešaka u opremi, menjanje programa i instaliranje programa na više mašina.

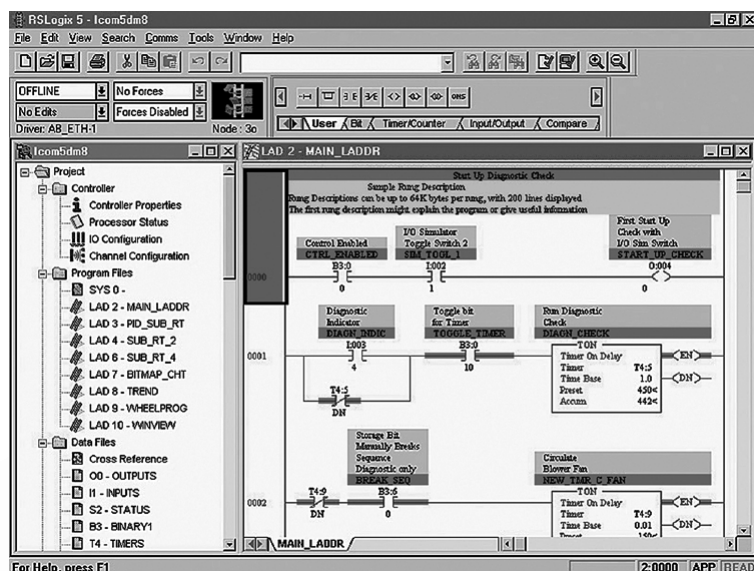


Sl. 10.25 Tipičan prenosni programator

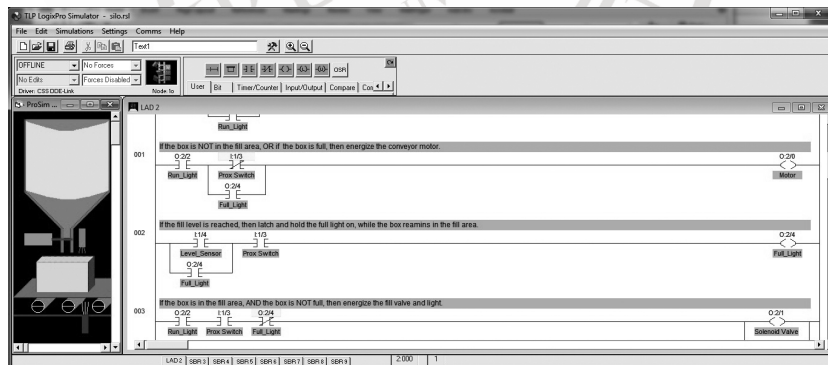
Ipak, za programiranje PLC, najčešće se koristi personalni računar (PC). Većina proizvođača personalnih računara nudi softver koji omogućava da se PC koristi kao uređaj za programiranje PLC-a. Taj softver omogućava korisniku da piše, menja, dokumentuje, čuva i ispravlja programe napisane u relejnoj lestvičastoj logici (slika 10.26). Monitor prenosnog računara omogućava prikazivanje većeg dela logike programa nego ekran prenosnog programatora, što olakšava tumačenje programa. Personalni računar komunicira s procesorom PLC-a preko serijskog ili paralelnog kabla za razmenu podataka, ili preko Ethernet mreže. Ako uređaj za programiranje nije u upotrebi, može se isključiti i razdvojiti od PLC-a. Razdvajanje uređaja za programiranje ne utiče na rad programa u PLC.

Program je niz naredaba koje autor programa zadaje da bi PLC izvršavao određene akcije. Jezik za programiranje određuje pravila za kombinovanje naredaba da bi se pokrenule željene akcije. Relejna lestvičasta logika je standardni programski jezik koji se koristi za

programiranje PLC-a. Potiče od upravljačke logike izvedene na osnovu elektromehaničkih releja. RLL program sadrži grafičke predstave nizova kontaktnih klem, namotaja motora i specijalnih blokova naredaba. RLL je prvobitno osmišljen tako da se lako koristi i razume, a zatim je izmenjen u skladu s rastućim potrebama industrije.



Sl. 10.26 Tipičan softver za PC koji omogućava izradu programa u relejnoj lestvičastoj logici



Sl. 10.27 TLP LogixPro Simulator

Principi rada

Da biste stekli utisak kako PLC radi, razmotrićemo jednostavan problem upravljanja procesom, ilustrovan na slici 10.28. Motor mešalice treba da automatski promeša tečnost u posudi kad god temperatura i pritisak dosegnu zadate vrednosti. Osim toga, postoji i mogućnost direktnog ručnog uključivanja mešalice pomoću posebnog tastera. Proces

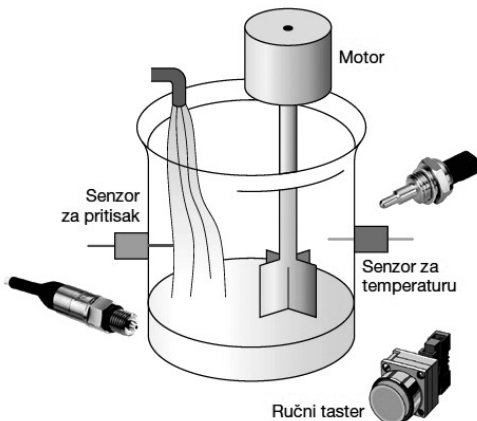
se nadzire pomoću prekidača povezanih sa senzorima za temperaturu i pritisak; prekidači (termostat i presostat) zatvaraju svoje kontakte kada se ispune unapred zadati uslovi za vrednosti merenih parametara.

Ovaj problem upravljanja može se rešiti pomoću relejne metode za upravljanje motorom mešalice, za koju je na slici 10.29 prikazan relejni lestvičasti dijagram. Namotaj startera motora (M) dobija napon kada se zatvore prekidači oba senzora (za pritisak i temperaturu), ili kad se pritisne ručni taster.

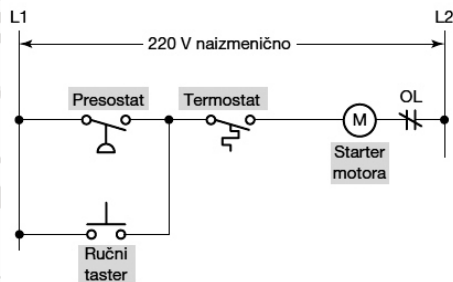
Sada ćemo razmotriti kako se za navedenu aplikaciju može upotrebiti programabilni logički kontroler. Koristićemo iste spoljne ulazne uređaje (senzor za pritisak, senzor za temperaturu i ručni taster). Ti uređaji bi bili kablovima povezani sa ulaznim modulom prema adresnoj šemi koju određuje proizvođač modula. Na slici 10.30 prikazano je tipično ožičenje modula koji se napaja na 220 V naizmjenično.

U PLC verziji možemo upotrebiti isti spoljni izlazni uređaj (namotaj startera motora). Taj uređaj bi bio kablovima povezan na odgovarajući izlazni modul prema adresnoj šemi koju određuje proizvođač modula. Slika 10.31 prikazuje tipično ožičenje za izlazni modul koji radi na 220 V naizmjenično.

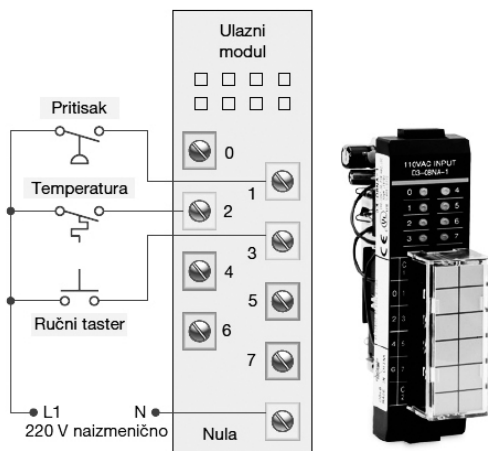
Sledeći korak bi bio pisanje programa za PLC u relejnoj lestvičastoj logici i njegovo unošenje u memoriju CPU. Na slici 10.32 prikazan je tipičan program relejne lestvičaste logike za upravljanje navedenim procesom. Format programa je sličan električnoj šemi kola s relejima. Simboli predstavljaju naredbe, dok brojevi predstavljaju adrese mesta u memoriji na ko-



Sl. 10.28 Problem upravljanja procesom mešanja tečnosti



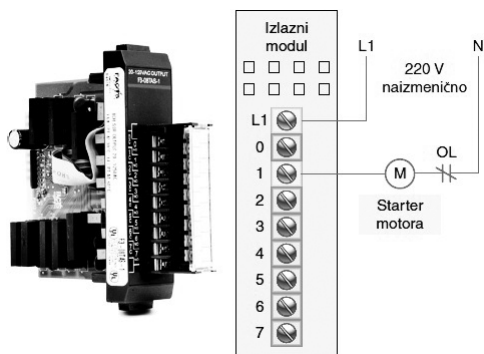
Sl. 10.29 Relejni lestvičasti dijagram upravljanja procesom



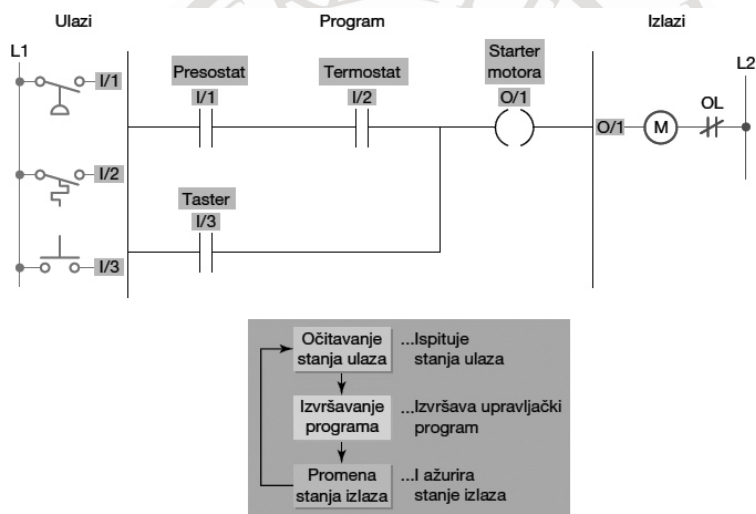
Sl. 10.30 Tipično ožičavanje ulaznog modula u modularnoj konfiguraciji s napajanjem na 220 V naizmjenično

jima se nalaze naredbe. Da biste programirali kontroler, te naredbe treba da unesete jednu po jednu u memoriju procesora pomoću uređaja za programiranje. Svakom ulaznom i izlaznom uređaju dodeljuje se adresa, što PLC procesoru omogućava da „zna“ gde je uređaj fizički povezan.

Zapamtite da se format U/I adrese razlikuje u zavisnosti od modela i proizvođača PLC sistema. Naredbe se smeštaju u korisnički deo memorije procesora. Tokom ciklusa programa, kontroler očitava stanje na ulazima, izvršava naredbe programa i u skladu s rezultatima programa menja stanje na izlazima sistema.



Sl. 10.31 Tipično ožičavanje izlaznog modula u modularnoj konfiguraciji s napajanjem na 220 V naizmenično



Sl. 10.32 Program za upravljanje procesom napisan na jeziku RLL, s tipičnim načinom adresiranja

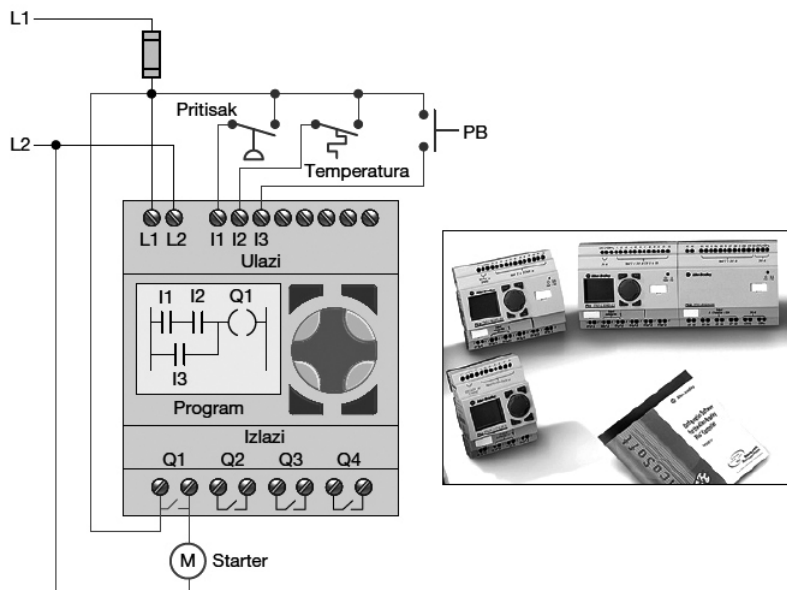
Kako bi program mogao da radi, kontroler treba prebaciti u izvršni režim (RUN), tj. u operativni (radni) ciklus. Tokom svakog radnog ciklusa, kontroler ispituje stanje ulaznih uređaja, izvršava naredbe programa i u skladu s njihovim rezultatima menja stanje izlaza. Svaki simbol u programu možemo zamisliti kao par radnih kontakata (standardno otvoreni). Simbol $\text{---}(\text{---})\text{---}$ predstavlja namotaj releja koji, kada dobije napon, zatvara odgovarajuće kontakte. U RLL programu na slici 10.32, namotaj O/1 dobija napon kada se zatvore kontakti I/1 i I/2, ili kada se zatvori kontakt I/3. Oba uslova formiraju neprekidnu logičku putanju sleva nadesno duž prečke lestvice koja obuhvata i namotaj startera motora.

Programabilni logički kontroler radi u realnom vremenu jer će svaki događaj koji se odigra na spoljnom ulaznom uređaju rezultovati određenom akcijom ili promenom stanja na izlazu kontrolera. Režim rada RUN procesora može se opisati sledećim nizom događaja:

- ♦ Prvo se ispituje stanje prekidača senzora za pritisak (presostat), prekidača senzora za temperaturu (termostat) i ručnog tastera. Ta stanja se beleže u memoriji kontrolera.
- ♦ Zatvoren kontakt se evidentira u memoriji kao logička jedinica, a otvoren kontakt kao logička nula.
- ♦ Zatim se analizira lestvičasti dijagram, pri čemu svaki interni kontakt u programu dobija status „otvoren (OPEN) ili „zatvoren“ (CLOSED) u zavisnosti od toga da li je program evidentirao logičko stanje 1 ili 0.
- ♦ Kada stanje ulaznih kontakata stvara logički kontinuitet sleva nadesno duž cele prečke, na memorijsku adresu koja predstavlja izlazni namotaj motora postavlja se logičko 1, što zatvara kontakte u interfejsu izlaznog modula.
- ♦ Kada ne postoji logički kontinuitet na programskoj prečki, na memorijsku adresu izlaznog namotaja motora postavlja se logička nula, što otvara kontakte u interfejsu izlaznog modula.
- ♦ Kada kontroler završi ceo navedeni niz akcija, kaže se da je kontroler završio jedan *ciklus* (engl. *scan*). Vreme ciklusa, tj. vreme potrebno za izvršavanje jednog celog ciklusa, pruža utisak o vremenu odziva PLC-a.
- ♦ Vrednost na memorijskoj adresi koja predstavlja dati izlazni uređaj najčešće se ažurira tokom ciklusa programa, ali se stanje odgovarajućeg stvarnog izlaznog uređaja menja tek na kraju ciklusa programa, u delu ciklusa koji je određen za ažuriranje stanja U/I uređaja.

Slika 10.33 prikazuje tipične kablovske veze koje su potrebne za upravljanje procesom pomoću fiksnog PLC-a. U ovom primeru, procesom upravlja i nadzire ga kontroler proizvođača Allen-Bradley Pico opremljen sa osam ulaza i četiri izlaza. Postupak instaliranja bio bi sledeći:

- ♦ Linije za napajanje energijom, zadate vrste i jačine • napona, povezuju se kroz osigurače na kleme L1 i L2 kontrolera.
- ♦ Udaljeni ulazni uređaji (presostat, termostat i taster) • povezuju se kablovima između klem L1 i klem I1, I2 odnosno I3 kontrolera.
- ♦ Starter motora se vezuje direktno na L2 i redno sa • izlaznim kontaktima releja Q1 na L1.
- ♦ Program lestvičaste logike unosi se pomoću tastature na prednjoj ploči kontrolera i LCD ekrana.
- ♦ Na raspolaganju je i Pico softver za programiranje koji omogućava izradu i testiranje programa pomoću personalnog računara.

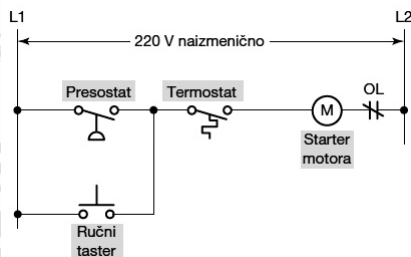


Sl. 10.33 Tipično ožičenje potrebno za upravljanje procesom pomoću fiksnog PLC

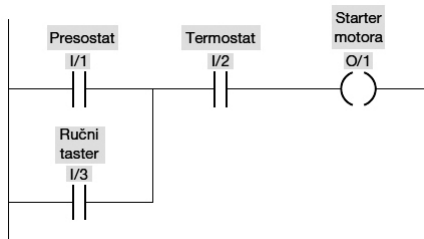
Menjanje načina rada programa

Jedna od važnih osobina PLC-a jeste lakoća s kojom se program može menjati. Pretpostavimo da moramo promeniti prvobitni proces upravljanja mešaćem na način prikazan na relejnom lestvičastom dijagramu sa slike 10.34. Izmena zahteva da uključivanje motora pomoću ručnog tastera bude dozvoljeno na bilo kom pritisku, ali ne dok nije dosegnuta zadata temperatura.

Da je bio upotrebljen sistem s relejima, bila bi potrebna izmena ožičenja strujnih kola kao na slici 10.34 da bi se postigla zahtevana izmena. Međutim, u slučaju PLC sistema, ne bi bila potrebna nikakva izmena ožičenja. Izlazi i ulazi i dalje su isti. Treba izmeniti samo program lestvičaste logike koji je instaliran u PLC (slika 10.35).



Sl. 10.34 Relejni lestvičasti dijagram za izmenjeni proces



Sl. 10.35 PLC relejni lestvičasti program za izmenjeni proces

PLC u poređenju sa standardnim računarima

Arhitektura PLC u suštini je ista kao arhitektura personalnog računara. Personalni računar (PC) može se programirati tako da radi kao programabilni logički kontroler ako na određeni način omogućite računaru da prima informacije od uređaja kao što su tasteri ili prekidači. Treba vam i program koji obrađuje ulazne podatke i odlučuje kada će uključivati i isključivati izlazne uređaje.

Međutim, PLC se razlikuje od personalnog računara po nekoliko važnih osobina. Prvo, za razliku od personalnih računara, PLC je projektovan za rad u industrijskim okruženjima, u širokom opsegu okolne temperature i vlažnosti. Na dobro projektovanu industrijsku PLC instalaciju, najčešće ne utiče električni šum koji je uobičajen u većini industrijskih pogona.

Za razliku od personalnog računara, PLC se programira u relejnoj lestvičastoj logici ili na drugim jezicima koji se lako uče. PLC se isporučuje s programskim jezikom već unetim u memoriju i nije potrebno trajno povezivanje s tastaturom, CD uređajem, niti monitorom. Umesto toga, PLC ima klemne za priključivanje spoljnih ulaznih i izlaznih uređaja, kao i komunikacione priključke.

Računari su složene računarske mašine koje su u stanju da izvršavaju više programa ili poslova u isto vreme i proizvoljnim redosledom. S druge strane, većina PLC izvršava samo jedan program, redom od prve do poslednje naredbe.

PLC upravljački sistemi projektovani su tako da se jednostavno instaliraju i održavaju. Otkrivanje grešaka je pojednostavljeno pomoću svetlosnih indikatora grešaka i poruka koje se prikazuju na ekranu programera. Sistemu se vrlo lako dodaju novi i uklanjanju postojeći ulazni i izlazni moduli za povezivanje sa udaljenim uređajima.

Softver namenjen za PLC ali napisan na personalnom računaru i izvršava se na njemu, deli se u dve opsežne kategorije:

- ♦ PLC softver koji korisniku omogućava da programira i dokumentuje programe stavlja korisniku na raspolaganje alatke za pisanje PLC programa – pomoću lestvičaste logike ili drugog programskog jezika – i dokumentuje ili objašnjava program sa onoliko detalja koliko smatra potrebnim.
- ♦ PLC softver koji korisniku omogućava da nadzire procese i upravlja njima zove se *interfejs čovek/mašina* (engl. *human machine interface*, **HMI**) i omogućava korisniku da prikaže proces – ili grafičku prezentaciju procesa – na monitoru, pregleda kako sistem radi, stekne uvid u trend promena vrednosti i prima obaveštenja o alarmima (slika 10.36). Mnogi operatorski interfejsi ne sadrže PLC softver. PLC se može integrisati u HMI, ali isti softver ne omogućava programiranje obe vrste uređaja.

U novije vreme, većina proizvođača opreme za automatizovanje procesa reagovala je na pojačane zahteve korisnika industrijskih upravljačkih sistema tako što je kombinovala prednosti koje pruža upravljanje u PLC stilu sa onim koje omogućavaju sistemi zasnovani na personalnim računarima. Ta vrsta uređaja nazvana je *programabilni automatizovan kontroler* (engl. *programmable automation controller*, **PAC**). Programabilni automatizovan kontroler kombinuje robusnost PLC-a i funkcionalnost personalnog računara. PAC

omogućava građenje složenih sistema sa softverskim odlikama kao što su napredno upravljanje, komunikacije, akvizicija podataka i obrada signala na robusnom hardveru koji omogućava izvršavanje programirane logike, pokreta, upravljanje procesima i daljinsko nadziranje.

Veličina PLC-a i aplikacije

PLC uređaji se razvrstavaju po odlikama kao što su funkcionalnost, broj ulaza i izlaza, cena i fizička veličina (slika 10.38). Od svih navedenih, najvažnija osobina je broj U/I tačaka (engl. *I/O count*). Tip uređaja nazvan nano najčešće se odnosi na najmanje PLC uređaje, s manje od 15 U/I tačaka. Tome slede tipovi mikro (od 15 do 128 U/I tačaka), srednji (od 128 do 512 U/I tačaka) i veliki (više od 512 U/I tačaka).

Uparivanje PLC sistema sa aplikacijom ključni je deo postupka izbora uređaja. Najčešće se ne preporučuje kupovina PLC sistema koji je veći nego što je neophodno za tekuće potrebe. Međutim, treba predvideti i buduće uslove i veličinu sistema dimenzionirati tako da ispunjava tekuće, ali eventualno i buduće zahteve aplikacije.

Postoje tri glavne kategorije PLC aplikacija: samostalne, višeprogramske i nadzorne. U *samostalnoj* (engl. *single-ended*) PLC aplikaciji, jedan PLC upravlja jednim procesom (slika 10.39). To je samostalna jedinica koja ne komunicira s drugim računarima, niti PLC uređajima. Veličina i složenost procesa kojim PLC upravlja očigledni su činioci koji utiču na odluku o izboru PLC uređaja. Aplikacije mogu nametnuti jak procesor, ali ta kategorija PLC aplikacija najčešće zahteva mali PLC.

U *višeprogramskoj* (engl. *multitask*) PLC aplikaciji, jedan PLC upravlja s više



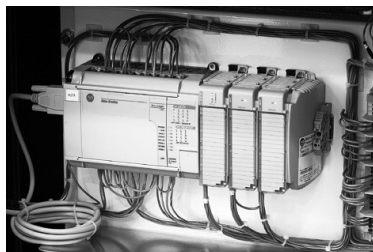
Sl. 10.36 Operaterski interfejs za PLC i monitor



Sl. 10.37 PAC (programabilni automatizovani kontroler)



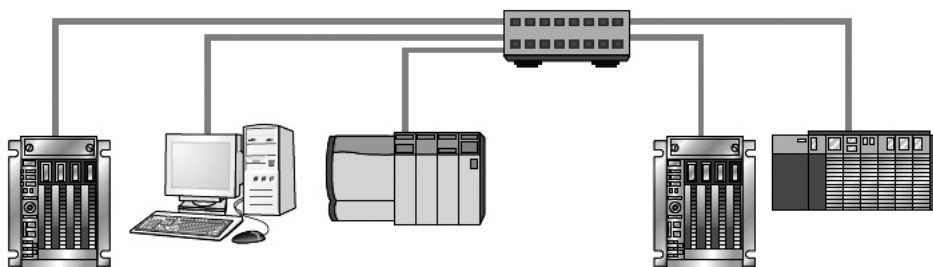
Sl. 10.38 Tipičan opseg veličina programabilnih kontrolera



Sl. 10.39 Samostalna PLC aplikacija

procesa. U toj vrsti instalacije značajan činilac je odgovarajući U/I kapacitet. Osim toga, ako je PLC podsistem nekog opsežnijeg procesa i treba da komunicira c centralnim PLC-om ili računarom, potrebna je i mreža za prenošenje podataka.

U *nadzornoj* (engl. *control management*) PLC aplikaciji, jedan PLC upravlja s više drugih PLC (slika 10.40). Ta vrsta aplikacije zahteva jak PLC procesor koji je projektovan za komuniciranje s drugim PLC sistemima i računarima ako zatreba. Nadzorni PLC nadgleda više drugih PLC sistema tako što im šalje programe koji nalažu podređenim PLC sistemima šta treba da se uradi. Vodeći PLC treba da uspostavlja veze s drugim PLC sistemima kako bi pomoću odgovarajućeg načina adresiranja mogao da komunicira s proizvoljnim podređenim PLC sistemom.



Sl. 10.40 Nadzorna PLC aplikacija

Memorija je komponenta PLC sistema gde se smeštaju podaci, naredbe i upravljački program. Veličina memorije najčešće se izražava u K vrednostima: 1 K, 6 K, 12 K itd. Prefiks „kilo“ (skraćeno k) najčešće znači 1000 osnovnih jedinica. Međutim, kada su u pitanju računari ili memorija PLC-a, 1 K znači 1024, zato što je u pitanju mera izražena u binarnom numeričkom sistemu ($2^{10} = 1024$). U zavisnosti od vrste memorije, 1 K znači 1024 bitova, 1024 bajtova ili 1024 reči.

Mada je uobičajeno da se kapacitet memorije PLC-a izražava kao broj reči, moramo znati broj bitova koji čine svaku reč kako bismo mogli da poredimo veličine memorija. U savremenim računarima uobičajene su veličine reči od 16, 32 ili 64 bita. Na primer, PLC koji radi sa 8-bitnim rečima ima smeštajni kapacitet od 49.152 bita ako je veličina memorije 6 K ($6 \times 1024 = 49.152$), dok PLC koji radi sa 32-bitnim rečima ima smeštajni kapacitet od 196.608 bitova sa istom memorijom veličine 6 K ($6 \times 1024 = 196.608$). Potrebna količina memorije zavisi od aplikacije. Veličina memorije potrebna za određenu PLC instalaciju zavisi od sledećih činilaca:

- ◆ Broj U/I tačaka u upotrebi
- ◆ Veličina upravljačkog programa
- ◆ Zahtevi po pitanju akvizicija podataka
- ◆ Da li su potrebne nadzorne funkcije
- ◆ Buduća proširenja

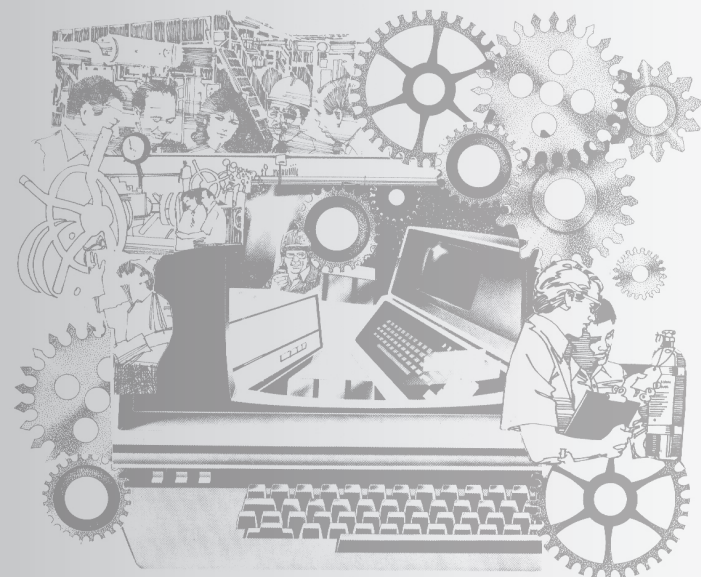
Skup programskih naredaba (engl. *instruction set*) za određeni PLC čine naredbe koje

on podržava. Njihov broj se kreće u opsegu od 15 naredaba na manjim jedinicama, do 100 naredaba na većim i moćnijim jedinicama.

Tabela 10.1 Tipične PLC naredbe

Naredba	Operacija
XIC (Examine ON).	Ispituje da li je dati bit u stanju ON
XIO (Examine OFF).	Ispituje da li je dati bit u stanju OFF
OTE (Output Energize).	Postavlja bit u stanje ON (bez zadržavanja stanja)
OTL (Output Latch).	Blokiranje stanja bita (sa zadržavanjem stanja)
OTU (Output Unlatch).	Deblokiranje stanja bita (sa zadržavanjem stanja)
TOF (Timer Off-Delay).	Postavlja izlaz u stanje ON ili OFF pošto je prečka bila u stanju OFF tokom zadatog vremenskog interval
TON (Timer On-Delay).	Postavlja izlaz u stanje ON ili OFF pošto je prečka bila u stanju ON tokom zadatogvremenskog interval
CTD (Count Down).	Softverski brojač za odbrojanje od zadate vrednosti do nule
CTU (Count Up).	Softverski brojač za odbrojanje od nule do zadate vrednosti





11. ADAPTACIJA

Posmatrajući postojeće u prirodi raznovrsne visoko organizovane sisteme, njihovu sposobnost prilagođavanja promenljivoj sredini, njihov razvoj i samoobnavljanje, pomišljamo, da u osnovi svih tih neobičnih pojava leži neki spoznajni mehanizam, koji daruje tim sistemima osobine, ne samo da ne gube svoju uređenost, organizovanost (koja obezbeđuje izvršavanje funkcija sistema), nego da je vremenom čak i povećavaju. Postanak i razvoj takvih sistema su, nesumnjivo, jedna od najgrandioznijih manifestacija stvaralačke moći *Prirode*.

Rađanje visoko organizovanih sistema može se objasniti time, da se u ogromnoj raznovrsnosti slučajnih sjedinjavanja elemenata, ranije ili kasnije, morala pojaviti kombinacija koja se pokazala u tolikoj meri prilagodljiva okolnoj sredini, da je ispoljila sposobnosti da živi, tj. sposobnost stabilnog održavanja svoje organizovane strukture. Ali, ne treba zamišljati pojavu sistema tako, kao da je on od samog početka posedovao toliko savršene osobine, koje bi pri bilo kakvim promenama uslova obezbeđivale najbolje, ili bar prihvatljivo njegovo ponašanje. Radi održavanja sposobnosti sistema da živi, pri nastajanju uslova u kojima je njegov mehanizam nesposoban da stvori potrebnu reakciju, neophodno je menjati strukturu sistema i formu njegovog ponašanja, menjati karakter reakcije sistema na poremećaje.

Proces menjanja osobina sistema, koji mu omogućava postizanje najboljeg ili u krajnjoj meri prihvatljivog funkcionisanja u promenljivim uslovima, naziva se *adaptacija* (ili *prilagođavanje*).

Uprkos tome što je suština prirodnih procesa adaptacije još uvek nedovoljno proučena, uspešno su konstruisani neki modeli, koji, mada u vrlo grubom i uprošćenom vidu, realizuju proces adaptacije. Takvi modeli omogućuju stvaranje predstave o mogućim mehanizmima spontanog formiranja struktura sposobnih da održavaju svoju uređenost i da formiraju ponašanja koja im daju mogućnost prilagođavanja promenljivim uslovima.

Svojstvo adaptacije jasno se ispoljava u mehanizmu *homeostaze*, koji se sastoji u tome, da živi organizmi poseduju osobinu održavanja svojih glavnih koordinata u dopuštenim (fiziološkim) granicama pri znatnim promenama uslova u kojima živi organizam. Adaptivnost mehanizma homeostaze može se ilustrovati primerom promena reagovanja organizama toplokrvnih životinja na promene temperature sredine. Pri relativno niskim temperaturama sredine termoregulacija u organizmu se ostvaruje promenom navale krvi u površinska tkiva, što obezbeđuje optimalne uslove razmene toplote između organizma i sredine. Pri dovoljno visokim temperaturama okoline u toku procesa termoregulacije uključuju se mehanizmi lučenja znoja i disanja, koji obezbeđuju intenzivno izlučivanje suvišne toplote. Dakle, organizam se prilagođava promenljivim uslovima — adaptira se i menja svoje ponašanje težeći obezbeđivanju homeostaze, u datom primeru težeći održavanju temperature tela u dozvoljenim granicama.

Veštački upravljani sistemi takođe, po pravilu, rade u izrazito promenljivim uslovima i za obezbeđivanje njihovog efikasnog funkcionisanja pokazuje se primamljivim korišćenje mehanizama sličnih prirodnom mehanizmu adaptacije. Sposobnost tehničkih ili ekonomskih upravljanih sistema da se prilagođavaju promenljivim uslovima može se postići korišćenjem različitih načina izbora režima rada ili algoritma formiranja upravljačkih dejstava u skladu sa promenama uslova u kojima sistem radi. Te promene uslova mogu se sastojati u promenama spoljnje sredine, ili u promenama osobina i karakteristika pojedinih delova upravljanih sistema. Izbor najpovoljnijeg ili prihvatljivog režima rada sistema može se ostvariti, i često se ostvaruje, putem *traganja*.

Izbor najpovoljnijeg režima

Da bi sistem normalno funkcionisao, potrebno je da veličine, koje određuju režim njegovog rada, ne izlaze van granica dozvoljenih vrednosti. Tako na primer, radi održavanja vitalnosti životinje, veličine kao što su temperatura tela, sadržaj kiseonika i glikoze u krvi i dr., ne smeju izaći van dopuštenih granica; radi normalnog rada parnog kotla neophodno je održavati u opsegu dopuštenih vrednosti pritisak i temperaturu pare, nivo vode u dobošu i temperaturu napojne vode.

Označimo veličine, koje karakterišu režim rada sistema, a čijim vrednostima možemo raspomagati, sa $R_1, R_2, \dots, R_n$.

U najprostijem slučaju, kada granične vrednosti režimskih veličina ne zavise od režima rada sistema, ta ograničenja imaju karakter:

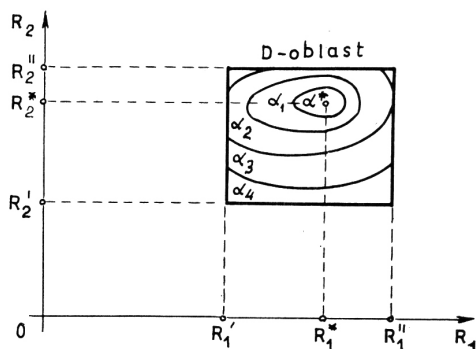
$$\left. \begin{array}{l} R_1' \leq R_1 \leq R_1'' \\ R_2' \leq R_2 \leq R_2'' \\ \dots\dots\dots \\ R_n' \leq R_n \leq R_n'' \end{array} \right\} \quad (11.1)$$

gde su sa jednom crticom obeležene najmanje dopuštene vrednosti, a dvema crticama — najveće dopuštene vrednosti režimskih veličina.

U prostoru režima sistema — prostoru, duž čijih osa se nanose vrednosti režimskih veličina, nejednačine (11.1) izdvajaju D — oblast (n -dimenzionalni pravougli paralelepiped), unutar koje mora uvek da se nalazi tačka koja prikazuje režim rada sistema. Sistem upravljanja, dakako, mora uvek obezbediti održavanje režimskih veličina u granicama vrednosti koje obezbeđuju da se reprezentativna tačka nalazi u granicama D -oblasti. Ali, time se ne iscrpljuju zadaci upravljanja, pošto tačke, koje leže unutar D -oblasti nisu u svakom pogledu jednako vredne. Svakom režimu sistema odgovara određena vrednost kriterijuma efikasnosti a , koji odražava stepen usklađenosti svakog režima sa ciljem upravljanja. Za žive organizme kriterijum a može određivati verovatnoću da će ostati živ ili intenzitet razvoja, za parni kotao to može biti stepen korisnosti ili proizvodnja pare, za transportni sistem — dobit od prevoženja ili tone — kilometara prevezenog tereta. Obeležimo vrednost a u svakoj tački D -oblasti i spojimo tačke koje odgovaraju jednakim vrednostima a , kao što je urađeno, na primer, na sl. 11.1 za slučaj dvodimenzionalnog prostora režima.

Ako je najpovoljnija vrednost za a njegova najveća vrednost, tada će najpovoljniji režim rada sistema biti režim prikazan tačkom a^* i sistemu upravljanja se može postaviti zadatak da odredi vrednosti veličina $R_1^*, R_2^*, \dots, R_n^*$, koje karakterišu režim, tako, da obezbede najpovoljniju vrednost kriterijuma efikasnosti $a = a^*$. U slučajevima kada promena uslova, u kojima se nalazi sistem, ne dovodi do bitnih promena vrednosti koordinata tačke a^* , zadatak se rešava odgovarajućim sistemom stabilizacije.

Međutim, kada se položaj tačke a^* menja u procesu rada sistema i to na nepredviđeni način, zadatak upravljanja postaje složeniji. Sistem upravljanja mora uvek da određuje vrednosti veličina $R_1^*, R_2^*, \dots, R_n^*$, koje karakterišu režim, tako da a prima najpovoljniju vrednost a^* , u skladu s ograničenjima tih veličina, ma kako se menjao položaj tačke a^* .

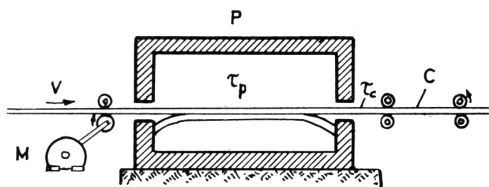


Sl. 11.1. Krive jednakih vrednosti a u prostoru režima sistema

Neka je, na primer, upravljani objekt prolazna peć P za termičku obradu čeličnih cevi C , koja je pokazana na sl. 11.2. Režim rada tog agregata je određen temperaturom τ_p u radnom prostoru peći i brzinom v pomeranja cevi.

Kao kriterijum efikasnosti agregata usvaja se potrošnja goriva q , sagorelog u peći, po toni obrađenog materijala

$$a = \frac{q}{r \cdot g} = \min_{g, \tau_p} \quad (11.2)$$



Sl. 11.2. Prolazna peć za termičku obradu čeličnih cevi: M — motor pogona za pomeranje cevi, P — peć, C — cevi

gde je r — težina jednog metra cevi. Vrednost a u datom slučaju mora biti minimizirana po $\mathcal{G}$ i τ_p , tj. treba da se izaberu takve dozvoljene vrednosti za $\mathcal{G}$ i τ_p da bi a bilo što je moguće manje vrednosti. Granice D -oblasti u datom slučaju određuju se s obzirom na:

ograničenja brzine

$$\mathcal{G}' \leq \mathcal{G} \leq \mathcal{G}'', \quad (11.3)$$

gde je $\mathcal{G}'$ — najmanja dozvoljena brzina pomeranja cevi koja obezbeđuje potrebnu produktivnost agregata, $\mathcal{G}''$ — najveća brzina koju razvija pogon;

ograničenja temperature

$$k' \cdot \mathcal{G} \leq \tau_p \leq k'' \cdot \mathcal{G}, \quad (11.4)$$

gde su k' i k'' — koeficijenti koji određuju dozvoljene odnose između temperature u peći i brzine pomeranja cevi, obezbeđujući odgovarajuću najmanju dopuštenu i najveću dopuštenu vrednost temperature τ_c cevi na izlazu iz peći. Određena na taj način D -oblast u prostoru režima $(\mathcal{G}, \tau_p)$ pokazana je na sl. 11.3 sa unetim u nju nivoskim linijama $a = \text{const}$. Najpovoljnijoj vrednosti $a = a^*$ odgovara ovde tačka s koordinatama $\mathcal{G}^*$ i τ_p^* . Dijagrami pokazuju granice D -oblasti i položaj nivoskih linija a za cevi sa debelim zidom (sl. 11.3, a) i za cevi sa tankim zidom (sl. 11.3, b).

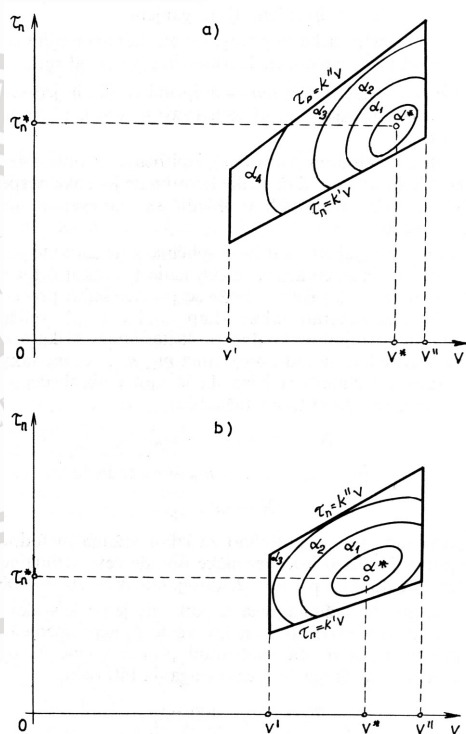
U sistem upravljanja se ne uvodi informacija o obliku nivoskih linija ili o najpovoljnijim vrednostima veličina koje karakterišu režim, već su, po pravilu, unapred nepoznate. Stoga je potrebno, da najpovoljniji režim bude izabran empirijski analiziranjem raznih režima i njihovim upoređivanjem, tj. *traganjem*.

Radi organizacije efikasnog traganja za najpovoljnijim režimom razrađeni su određeni postupci. Razmotrimo neke od njih.

Metoda detaljnog ispitivanja

Najprimitivniji ali jednostavan i pouzdan metod traganja. Sastoji se u uzastopnom analiziranju svih različitih režima.

Pri traganju metodom detaljnog ispitivanja D -oblast se razbija na N ćelija, čije su veličine duž svake koordinate jednake, respektivno $\Delta R_1, \Delta R_2, \dots, \Delta R_n$, izabranih u skladu sa zahtevanom tačnošću određivanja režima.



Sl. 11.3. Krive jednakih vrednosti kriterijuma a : a) za cevi sa debelim zidom; b) za cevi sa tankim zidom

Uzastopno usvajajući vrednosti veličina koje karakterišu režim rada tako, da se reprezentativna tačka nađe u svakoj ćeliji i, pamteći vrednost a za svaku ćeliju, može se po završetku procesa detaljnog ispitivanja izabrati takav skup vrednosti tih veličina, da odgovaraju najpovoljnijoj vrednosti kriterijuma efikasnosti a^* . Broj ćelija N određuje se tada brojevima $m_1, m_2, \dots, m_n$ takvih diskretnih vrednosti režimskih veličina da je njima obezbeđeno uspostavljanje režima sa zahtevanom tačnošću:

$$N = m_1 \cdot m_2 \cdot \dots \cdot m_n.$$

Ako su $m_1 = m_2 = \dots = m_n = m$ tada je

$$N = m^n. \quad (11.5)$$

Ocenimo sada vreme potrebno za izbor režima metodom detaljnog ispitivanja. Za primer termičke obrade cevi, uzimajući da je $m = 20$ (greška do 5%) pri $n = 2$, dobijamo $N = 20^2 = 400$.

Da bi se utvrdio režim agregata, potrebno je posle svake promene režimskih veličina održavati ih neko vreme T_s nepromenjenim i tek tada oceniti vrednost a . Za analizirani proces vreme T_s ne može biti manje od 1 min. Stoga će vreme traganja biti reda

$$T_t = NT_s = 400 \text{ min},$$

što je svakako neprihvatljivo, ako se klasa obrađivanih cevi menja svaka 2—3 časa, kao što obično biva u praksi.

Ako bismo hteli da pronađemo najpovoljniji režim metodom detaljnog ispitivanja za proces koji se karakteriše sa 10 režimskih veličina, s greškom do 1%, tada bi čak, ako pretpostavimo da je potrebno utrošiti 1 sek. za jedan korak, bilo potrebno izvesti

$$N = 100^{10}$$

koraka, a vreme traganja bi iznosilo

$$T = 100^{10} \text{ sek. (približno } 3 \cdot 10^{12} \text{ godina),}$$

što predstavlja period koji je mnogo veći od starosti Zemlje.

Očigledno je da metoda detaljnog ispitivanja postaje neprihvatljiva za traganje, čim je sistem makar i malo složeniji. Znatno efikasnije traganje obezbeđuje sledeći metod.

Metoda Gaus-Zajdela

Neka je potrebno naći vrednosti v i τ koje minimiziraju a u D -oblasti, pokazanoj na sl. 11.4, gde tačka a_p prikazuje početni režim. U skladu sa metodom Gausa-Zajdela, dajmo veličini v_p prirastaj jednak Δv , i proverimo u kom smislu se izmenila vrednost a . Birajući smer u kome se a smanjuje dodavaćemo vrednosti v po Δv na svakom koraku, proveravajući svaki put da li se a smanjuje. U tacki a_1 , gde promena v ne dovodi do smanjenja a , počinje se kretanje po koordinati τ na isti način do tačke a_2 , gde a prestaje da opada. Zatim se postupak ponavlja duž ose v , dok u sistemu ne nastupi takav režim a^* , da promena v za $\pm \Delta v$ ili promena τ za $\pm \Delta \tau$ ne vode daljem smanjenju a .

Broj koraka koje pri tome treba izvršiti da bi se došlo u blizinu tačke a^* , određen je karakterom zavisnosti a od režimskih veličina i početnim režimom sistema, ali red veličine broja koraka može se grubo oceniti veličinom

$$N = m \cdot n. \quad (11.6)$$

Za peć pri $m = 20$, $n = 2$, $T = 1$ min. nalazimo vreme traganja

$$T_t = m \cdot n \cdot T_s = 20 \cdot 2 \cdot 1 = 40 \text{ min.},$$

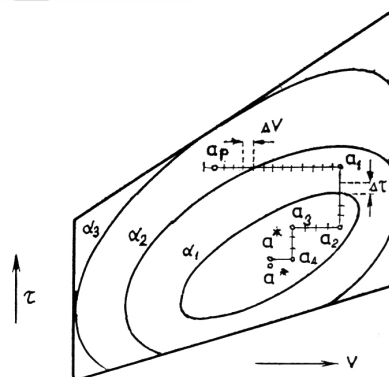
što je za jedan red veličine manje nego pri traganju metodom detaljnog ispitivanja, a za dati primer $m = 100$, $n = 10$, $T_s = 1$ sek.

$$T_t = 100 \cdot 10 \cdot 1 = 1000 \text{ sek.}, \text{ tj. približno 3 časa}$$

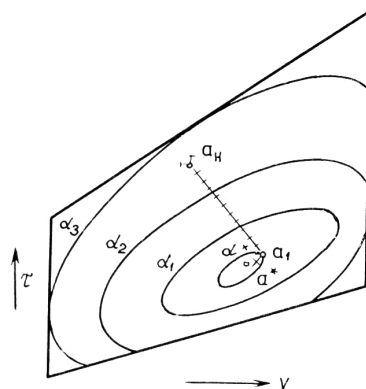
umesto $T_t = 3 \cdot 10^{12}$ godina pri traganju detaljnim ispitivanjem.

Još veće skraćivanje vremena može se postići korišćenjem metoda, zasnovanih na istovremenom menjanju nekoliko režimskih veličina u potrebnom smeru (metoda gradijenta, metoda najkraćeg pada (sl. 11.5), metoda ponora).

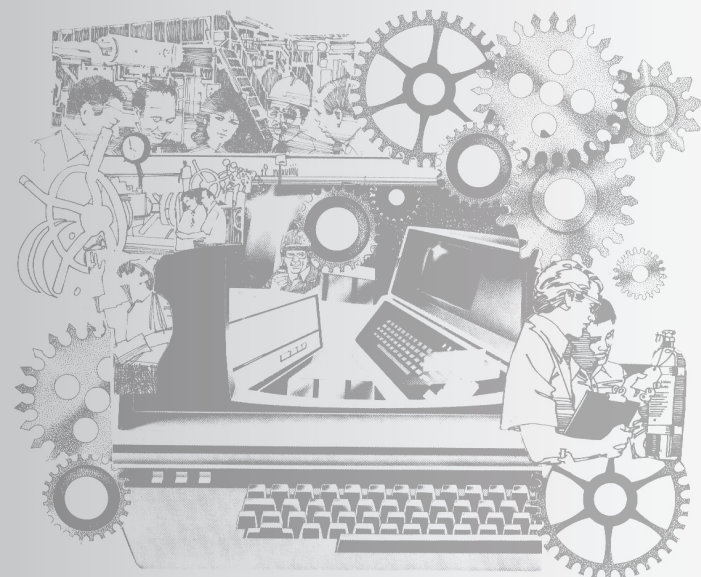
Na zadatak traženja ekstremuma funkcije više promenljivih svode se, ne samo zadaci izbora najpovoljnijeg režima, već i mnogi drugi zadaci, na primer određivanje najpovoljnijih dimenzija transformatora, izbor optimalne konfiguracije vodovodne ili gasne mreže, formiranje optimalnog proizvodnog plana preduzeća i sl., za čije se rešavanje koriste spomenuti postupci traganja.



Sl. 11.4. Traganje po metodi Gaus-Zajdela



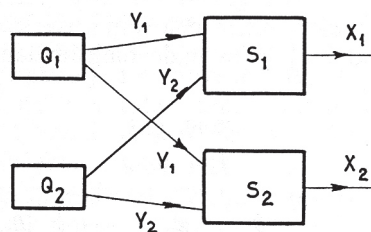
Sl. 11.5. Traganje po metodi najkraćeg pada



12. UVOD U TEORIJU IGARA

Do sada smo analizirali ponašanje sistema čiji upravljački uređaji rade tako da se funkcionisanje upravljanog sistema odvija na najpovoljniji način pri promenama poremećaja slučajnog karaktera. Postoje, međutim, situacije kada neki faktori koji bitno utiču na funkcionisanje sistema S_1 , opremljenog upravljačkim uređajem Q_1 , zavise i od rada upravljačkog uređaja Q_2 sistema S_2 (sl. 12.1). Tada se može dogoditi da su »interesi« sistema S_1 i S_2 suprotni u tom smislu, da je svako poboljšanje funkcionisanja sistema S_1 povezano s pogoršanjem funkcionisanja sistema S_2 i obratno. Tada će upravljački uređaji, težeći da poboljšaju rad »svog« sistema, proizvoditi upravljačka dejstva što je moguće štetnija za »tuđi« sistem. Takve situacije se nazivaju *konfliktne situacije*. Konfliktne situacije se javljaju u procesu borbe živih organizama za postojanje, u klasnoj borbi, pri ekonomskom rivalstvu, u vojnim sukobima i u mnogim drugim slučajevima, kada zadatke upravljanja treba rešavati uzimajući u obzir protivdejstva protivnika, usmerena na poboljšanje rada njegovog upravljanog sistema. Postojanje »razumnog protivnika« iz osnova menja karakter zadatka formiranja svrshodnog upravljanja, jer će se ma kakvo upravljanje da izaberemo, protivnik potruditi da nas dovede u najgore uslove, uzimajući u obzir naše delovanje.

Zadatak upravljanja u konfliktnoj situaciji sastoji se u izboru takve reakcije na situaciju stvorenu u procesu borbe, — takve *strategije*, pri kojoj bi se upravljani sistem nalazio u što je moguće povoljnijem položaju, čak i pri najnepovoljnijim dejstvima protivnika. Razradi metoda za rešavanje zadataka te vrste posvećena je *teorija igara*.



Sl. 12.1. Unakrsni uticaj dvaju upravljačkih sistema na objekte upravljanja

Parna igra i minimaksna strategija

Uprošćeni formalizovani model konfliktne situacije naziva se *igra*. Igra se formuliše skupom pravila koja definišu dozvoljena dejstva strana-učesnika igre i načine određivanja njihovog dobitka ili gubitka. Igra u kojoj učestvuju dve strane (dva igrača) naziva se *parna igra*. *Igra s nultom sumom* je igra, u kojoj je algebarska suma dobitaka svih strana jednaka nuli.

Razmotrimo najprostiju klasu igara—parne igre s nultom sumom. U takvim igrama interesi strana su suprotni, jer je dobitak jednog igrača jednak gubitku drugog. Stoga se možemo ograničiti na razmatranje samo dobitka jednog igrača, koji on teži da maksimizira, a njegov protivnik — da minimizira. Igra se sastoji u tome, da svaki od protivnika izvodi jedno proizvoljno dejstvo iz mnoštva njemu dostupnih dejstava. Pri tome se pretpostavlja da je izbor svakog igrača nepoznat njegovom protivniku. Ali, oba protivnika znaju sve moguće varijante svojih i tuđih dejstava kao i veličinu dobiti za bilo koju njihovu kombinaciju. Takve igre (nazvane *konačnim igrama*) pogodno je predstavljati matricom M , čiji svaki element a_{ij} pokazuje veličinu dobitka igrača A , ako on bira dejstvo A_i a njegov protivnik B — dejstvo B_j . Ako igrač A ima m različitih dejstava, a igrač B — n različitih dejstava, tada će matrica igre sadržati m vrsta i n kolona (tabela 12.1). Matrica igre se naziva i matrica plaćanja (njeni elementi su iznosi plaćanja igrača B igraču A).

Rešiti igru $m \times n$ znači naći za svakog igrača takvu strategiju tj. takav način delovanja, da srednji iznos njegovog dobitka za veliki broj igara bude maksimalan. *Teorija igara preporučuje svakom igraču da izabere takvo dejstvo, pri kome će primiti maksimalan dobitak u slučaju najnepovoljnijeg delovanja protivnika*. Takva strategija se naziva **minimaksna strategija**.

Tabela 12.1

	B_1	B_2	...	B_n
A_1	a_{11}	a_{21}	...	a_{1n}
A_2	a_{21}	a_{22}	...	a_{2n}
...	...	...	...	...
A_m	a_{m1}	a_{m2}	...	a_{mn}

Najpovoljnija (optimalna) strategija A_k igrača A može tada da se odredi iz matrice igre pronalaženjem onog njenog elementa koji zadovoljava uslov

$$a_{kp} = \min_j \max_i a_{ij},$$

tj. takvog elementa, koji je izabran kao maksimalni (po vrstama) među minimalnim u svakoj vrsti (po kolonama).

Najpovoljnija strategija B_i za igrača B je određena elementom

$$A_{q_i} = \min_i \max_j a_{ij},$$

tj. takvim elementom koji je minimalan (po kolonama) među maksimalnim (po vrstama) u svakoj koloni.

Razmotrimo sledeći primer konfliktne situacije. Preduzeće (igrač A) treba da obezbedi rezervu crne boje za tkanine. Potreba za bojom zavisi od konjunktura tržišta (igrač B) u predstojećoj sezoni. Ta potreba može iznositi 10 t pri smanjenoj potražnji, 15 t pri normalnoj potražnji i 20 t — pri povećanoj potražnji crne tkanine. Cena boje u trenutku donošenja rešenja iznosi 10 hiljada novčanih jedinica po toni, što odgovara sniženoj potražnji. Pri normalnoj potražnji cena boje raste do 15 hiljada novčanih jedinica po toni, a pri povišenoj — do 20 hiljada novčanih jedinica po toni.

Stoga igrač A treba da odabere jednu od tri strategije — kupovinu 10, 15 ili 20 t boje (količina koju kupuje odmah). Igrač B može obezbediti jednu od tri potražnje: smanjenu, normalnu ili povećanu. Matrica igre u datom slučaju je tipa 3×3 . Njene elemente — troškove za nabavku boje — lako je izračunati za devet mogućih kombinacija strategija oba igrača, na osnovu prikazanih podataka. Na primer element a_{12} , koji odgovara kupovini 10 t boje (strategija A_1) i normalnoj konjunkturi na tržištu u narednoj sezoni (strategija B_2), izračunava se na sledeći način: za kupovinu 10 t boje po 10 hiljada novčanih jedinica po toni potrebno je 100 hiljada novčanih jedinica. Osim toga, pri normalnoj konjunkturi dokupljuje se u toku sezone dopunskih 5 t po ceni 15 hiljada novčanih jedinica po toni. Ukupni troškovi iznose 175 hiljada novčanih jedinica, tj. $a_{12} = 175$. Cela matrica plaćanja je data u tabeli 12.2.

Iz prikazane tabeli se vidi da su minimumi vrsta, respektivno, -300, -250 i -200 hiljada novčanih jedinica, a maksimum kolona: -100, -150 i -200 hiljada novčanih jedinica i maksimalni minimum vrsta (po kolonama) poklapa se s minimalnim maksimumom kolona po vrstama (oba su jednaka a_{33}). Stoga, razumno rešenje predstavlja kupovinu rezerve 20 t boje po ceni od 10 hiljada novčanih jedinica po toni.

Tabela 12.2

Rezerva boje u tonama	Potražnja		
	B_1 smanjena	B_2 normalna	B_3 povećana
$A_1 = 10$	$a_{11} = -100$	$a_{12} = -175$	$a_{13} = -300$
$A_2 = 15$	$a_{21} = -150$	$a_{22} = -150$	$a_{23} = -250$
$A_3 = 20$	$a_{31} = -200$	$a_{32} = -200$	$a_{33} = -200$

U navedenom primeru je $a_{kp} = a_{ql} = a_{kl} = a_{33}$, što svedoči o postojanju tačke u matrici igre, nazvane *sedlasta tačka*, koja je istovremeno *maksimin* za igrača A i *minimaks* za igrača B . Takvo podudaranje ne postoji uvek i matrica igre u opštem slučaju može i da nema sedlastu tačku.

Sedlasta tačka a_{kl} se odlikuje sledećim važnim osobinama: ako jedan od igrača, na primer A , odabere strategiju A_k koja odgovara sedlastoj tački, tada za igrača B nikakvo odstupanje od strategije B_l ne može biti povoljno, jer ono može samo da održi ili da poveća dobitak za A , ali ga ne može smanjiti. U navedenom primeru igrač A će se pridržavati strategije A_3 i njegov dobitak neće biti manji od dobitka u sedlastoj tački, ma kakvu strategiju izabrao igrač B . Optimalne strategije A_k i B_p koje odgovaraju sedlastoj tački a_{kp} nazivaju se čiste strategije. Napominjemo, da postojanje sedlaste tačke čini neobavezanim poštovanje tajnosti. Oba igrača u tom slučaju mogu otvoreno objaviti svoje izabrane strategije.

Mešovita strategija, vrednost igre i dominiranje

Ako matrica igre nema sedlastu tačku, u tom slučaju ne postoje optimalne čiste strategije za svakog od igrača. Takve igre imaju složenije rešenje.

Razmotrimo primer igre 2×2 , koja nema sedlastu tačku. Neka je igrač A — komandir vojne jedinice koja treba da napadne jedno od dva odbrambena utvrđenja protivnika — igrača B . Protivnik može uspešno da brani samo jedno od tih utvrđenja, ali ne oba istovremeno. Poznato je, da je prvo utvrđenje trostruko važnije od drugog. Šta treba da rade oba protivnika?

Označimo strategije napada i odbrane prvog utvrđenja indeksom 1, drugog utvrđenja — indeksom 2. Tada matrica plaćanja ima vid:

	B_1	B_2
A_1	0	3
A_2	1	0

(Ako B odbrani ono utvrđenje koje napada A , tada je dobitak za A jednak nuli.) Lako je uočiti da ta matrica nema sedlastu tačku:

Primena čistih strategija u tom slučaju nije povoljna za igrača. Zaista, ako bi se A uvek pridržavao jednog istog dejstva (na primer, napada na važnije utvrđenje), B bi znajući to sveo dobit na nulu. Isto tako, ako bi B , na primer, uvek štitio važnije utvrđenje (što, reklo bi se, diktira »zdrav razum«), tada bi A sigurno dobio 1 — osvojio drugo utvrđenje. Zato se onaj koji primenjuje čistu strategiju nalazi u gorem položaju nego njegov »elastičniji« protivnik. Očigledno je pogodnije u tajnosti odabirati čas jednu, čas drugu čistu strategiju, ali ne po nekom unapred poznatom zakonu, već *slučajno*, recimo, koristeći kocku ili tablicu slučajnih brojeva. Na primer, ako bi igrač A s jednakim verovatnoćama napadao čas jedno, čas drugo utvrđenje tada, ma kako dejstvovao B , srednja dobit za A ne bi bila manja od $1/2$, tj. u svakom slučaju veća od nule. Postavlja se pitanje, kolike su pak optimalne verovatnoće primene čistih strategija? Za slučaj igre 2×2 teorija igara daje relativno prost odgovor: verovatnoće čistih strategija treba da se izračunaju po formulama:

$$\left. \begin{aligned} p(A_1) &= \frac{a_{22} - a_{21}}{(a_{11} + a_{22}) - (a_{12} + a_{21})}, \\ p(B_1) &= \frac{a_{22} - a_{12}}{(a_{11} + a_{22}) - (a_{12} + a_{21})}, \\ p(A_2) &= \frac{a_{11} - a_{12}}{(a_{11} + a_{22}) - (a_{12} + a_{21})}, \\ p(B_2) &= \frac{a_{11} - a_{21}}{(a_{11} + a_{22}) - (a_{12} + a_{21})}. \end{aligned} \right\} \quad (12.1)$$

Za naš zadatak dobijamo sledeće verovatnoće:

$$\begin{aligned} p(A_1) &= 1/4 & p(A_2) &= 3/4 \\ p(B_1) &= 3/4 & p(B_2) &= 1/4 \end{aligned}$$

Srednja dobit za A (tj. gubitak za B) jednaka je tada 3/4.

Strategija, koja se sastoji u slučajnoj primeni s određenim verovatnoćama nekih čistih strategija, naziva se mešovita strategija. Mešovita strategija se određuje davanjem verovatnoća primene čistih strategija koje u nju ulaze. U našem primeru optimalna mešovita strategija igrača S_A i S_B može se napisati u vidu

$$S_A = \begin{bmatrix} 1/4 \\ 3/4 \end{bmatrix}, \quad S_B = \begin{bmatrix} 3/4 \\ 1/4 \end{bmatrix}$$

pri čemu su čiste strategije numerisane odozgo nadole.

Za razliku od slučaja igre sa sedlastom tačkom, pri primeni mešovite strategije izbor određenog dejstva treba da ostane nepoznat protivniku. (Mada, razume se, optimalnu mešovitu strategiju igrača protivnik uvek može da izračuna iz matrice igre.)

Važna osobina optimalne mešovite strategije je da ona obezbeđuje igraču, pri bilo kakvoj strategiji protivnika (čistoj ili mešovitoj), srednji dobitak, ne manji od dobitka kada protivnik primenjuje svoju optimalnu mešovitu strategiju. Taj srednji dobitak, koji može dobiti »dobar« igrač od »dobrog« igrača, naziva se *vrednost igre*. U našem primeru vrednost igre je jednaka 3/4 (u korist A). U igri sa sedlastom tačkom vrednost igre je jednaka veličini plaćanja u sedlastoj tacki. Ako je vrednost igre jednaka nuli, tada igru možemo smatrati »pravednom«. U suprotnom slučaju za jednog od igrača (za koga je vrednost igre negativna) bolje je da se uzdrži od igre, ako je to moguće (to je, uostalom, takođe neka čista strategija koja svodi igru na igru sa sedlastom tačkom i vrednošću jednakom nuli).

Napominjemo, da se optimalna strategija igrača ne menja, ako sva plaćanja pomnožimo konstantnim brojem (izrazimo ih, recimo, u drugoj valuti), ili dodamo konstantan broj. Jasno je da se vrednost igre saglasno tome menja.

Naći rešenja igara sa većim brojem čistih strategija, nego 2×2 , je komplikovanije. Međutim, kao što je dokazao Džon fon Nojman, svaka igra $m \times n$ ima, u krajnjoj liniji, jedno rešenje u vidu optimalnih (čistih ili mešovitih) strategija za oba igrača.

Obratimo pažnju na osobinu matrice plaćanja, koja ponekad omogućuje, da se svede igra na igru s manjim brojem strategija. Razmotrimo, na primer, igru 3×3 s matricom zadatom tabelom 12.3.

Tabela 12.3

	B_1	B_2	B_3
A_1	2	-1	2
A_2	3	4	0
A_3	1	2	-2

Uporedimo međusobno plaćanja koja odgovaraju strategijama A_2 i A_3 . Lako je uočiti, da ma kakvu strategiju izabrao igrač B , strategija A_2 daje igraču A veći dobitak od A_3 . U takvom slučaju se govori da strategija A_2 *dominira* nad strategijom A_3 . Očigledno je za igrača A nepovoljno da ikada koristi strategiju A_3 , te se ona može isključiti iz razmatranja. Upravo tako, poredeći strategije B_1 i B_3 , nalazimo, da je B_1 gore za B (tačnije, *nije bolje*, pošto su pri strategiji A_1 dobiti za B jednaki). Zato je dovoljno razmatrati igru s matricom (tabela 12.4).

Tabela 12.4

	B_1	B_2
A_1	-1	2
A_2	4	0

Dominiranje nam je omogućilo da svedemo igru 3×3 na igru 2×2 . Koristeći formule (12.1) i uzimajući u obzir dominiranje, nalazimo optimalnu mešovitu strategiju za prvo-bitnu igru:

$$S_A = \begin{vmatrix} 4/7 \\ 3/7 \\ 0 \end{vmatrix}; \quad S_B = \begin{vmatrix} 0 \\ 2/7 \\ 5/7 \end{vmatrix}.$$

Vrednost igre je jednaka $8/7$. Moguć je i složeniji slučaj dominiranja, kada je bilo koja čista strategija gora od neke mešovite strategije, sastavljene od ostalih čistih strategija.

Upoznali smo se s najprostijim tipovima parnih konačnih igara s nultom sumom. Mnogo su složenije igre s velikim brojem strategija, posebno takozvane *beskonačne igre* (s beskonačnim mnoštvom čistih strategija), a i igre više lica, osobito *koalicione igre*, u kojima igrači mogu obrazovati koaliciju s ciljem povećanja ukupnog dobitka. Dostignuća teorije igara nalaze primenu u najraznovrsnijim oblastima, kao što su, na primer, ekonomija,

analiza ponašanja ljudi i socijalnih grupa u društvu, izbor optimalnog načina delovanja pri nepotpunoj informaciji o objektu (takozvana »igra s prirodom«) itd.

Vrste igara

Igre se mogu deliti po raznim osnovama: prema broju igrača, prema broju raspoloživih strategija, prema funkciji vremena, prema izboru strategija, prema tome da li je suma konstantna ili promenljiva.

- ♦ **Podela prema broju igrača**

Prema broju igrača igre se dele na igre sa dve, tri, ... , n osoba. Igre sa dva igrača najviše se koriste u teoriji igara. Ovo nije slučajno budući da se i u svakodnevnom životu najviše susrećemo sa ovakvim tipom igara (npr. šah). Ove igre detaljno su analizirane u teoriji igara. Cilj je da se stanje opiše formom koja je dobro poznata, tj. igrom čije karakteristike su unapred poznate i zapravo predvidive. Igre sa n igrača su generalizacije igara u kojima učestvuju dva igrača. U daljem izlaganju akcenat je na igrama u kojima učestvuju dva igrača.

- ♦ **Podela prema broju raspoloživih strategija**

Prema broju raspoloživih strategija igre se dele na igre sa konačnim i beskonačnim brojem strategija. Ako bar jedan igrač ima beskonačan skup strategija, reč je o igri sa beskonačnim brojem strategija.

- ♦ **Podela prema funkciji vremena**

U zavisnosti od toga da li su strategije igrača u funkciji vremena, govori se o dinamičkim ili statičkim igrama. Ako igrači svoje odluke donose istovremeno tada se radi o statičkim igrama, u suprotnom govori se o dinamičkim igrama.

- ♦ **Podela prema izboru strategija**

Izbor strategije u igri može biti deterministički ili stohastički. Igre gde je izbor strategija slučajan (npr. bacanje kocke) nazivaju se stohastičke igre.

- ♦ **Igre sa konstantnom i promenljivom sumom**

Isplata za svakog igrača u igri rezultat je izabrane strategije tog igrača, ali i akcija svih ostalih učesnika u igri. Prema krajnjem ishodu (ukupnoj isplati) igre mogu biti sa konstantnom ili promenljivom sumom.

Igre dva igrača u opštem smislu su predstavljene matricama (tabelama/tablicama) čiji su elementi uređeni parovi. Igrač 1 ostvaruje svoju strategiju birajući jednu od vrsta matrice, dok igrač 2 ostvaruje strategiju birajući kolonu. Čelija koja se nalazi u preseku izabranih strategija naziva se rezultat igre. Prvi član uređenog para rezultata igre predstavlja dobitak igrača 1, a drugi član dobitak igrača 2. U daljem izlaganju podrazumevano je da su oba igrača racionalna, tj. žele da maksimiziraju dobitak. Negativan dobitak je gubitak.

Specijalan slučaj igara sa konstantnom sumom su *igre nulte sume*.

Predstavljanje igara

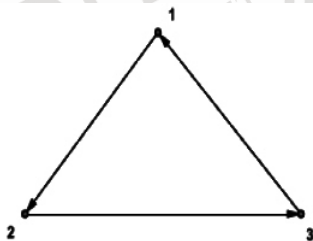
Sve igre se mogu predstaviti na tri različita načina, kao igre u ekstenzivnoj formi, kao igre u normalnoj formi i kao igre u obliku karakteristične funkcije. Na koji način će se predstaviti igra zavisi od prirode problema. U nastavku su prikazane igre nulte sume u ekstenzivnoj i normalnoj formi.

♦ Igra u ekstenzivnoj formi

Svaku situaciju karakteriše odgovarajuća raspoloživost informacija, kao i precizna procedura odvijanja igranja. Neophodno je da se razviju igre koje će odražavati tok igre kao i njenu informacionu strukturu. U ovakvim situacijama koriste se igre u ekstenzivnoj formi koje se predstavljaju u obliku drveta. Drvo predstavlja skup čvorova i grana povezanih na odgovarajući način. Svaka grana predstavlja jednu raspoloživu akciju za igrača koji donosi odluku u tom čvoru odlučivanja, dok čvor, ukoliko nije krajnji čvor igre, predstavlja poziciju u igri gde jedan od igrača mora da donese odluku (tj. rezultat toka igre do date etape). Početna tačka igre se naziva koren ili inicijalni čvor. Čvorovi kojima se igra završava zovu se krajnji ili terminalni čvorovi i često se njihove vrednosti označavaju kao rezultat igre.

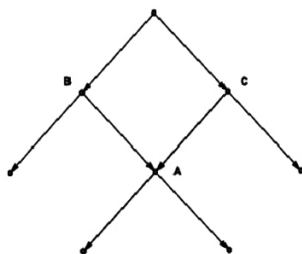
Da bi se uspešno prikazale igre u ekstenzivnoj formi moraju biti ispunjeni sledeći uslovi:

1. Nije dozvoljena cirkularnost, što znači da se ne sme dogoditi situacija kao na sledećoj slici.

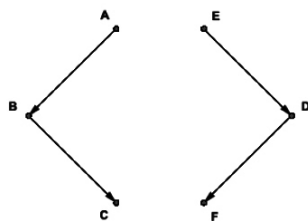


U ovom primeru je nemoguće utvrditi početni čvor, kao ni terminalne čvorove igre.

2. Nije dozvoljena višestrukost, što znači da do svakog čvora odlučivanja, koji nije inicijalni može da vodi najviše jedna grana, a ne kao na sledećoj slici gde do čvora A vode dve grane tj. do čvora A se može stići iz čvorova B i C.



3. Nije dozvoljena odvojenost strategija kao što vidimo na sledećoj slici. Ovaj grafik ne predstavlja igru u obliku drveta, tj. ove dve igre nisu deo jedne igre.



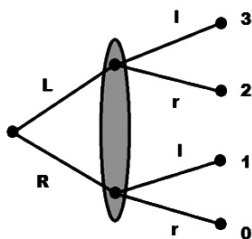
♦ Igra u normalnoj formi

Kod ovih igara naglasak nije na proceduralnosti i toku igre, niti na njenim informacionim aspektima, kao u ranijem slučaju. Bitna stavka je da igrači u isto vreme donose odluke. Oni ne moraju da donesu odluke baš u isto vreme, ali budući da nemaju mogućnost da saznaju kako je odigrao njihov protivnik, izgleda kao da se odluke donose istovremeno. Ovaj oblik se najčešće primenjuje u slučajevima kad u igri učestvuje dva igrača. U ovakvom načinu prikazivanja igara data je matrica u kojoj vrste predstavljaju raspoložive akcije prvog igrača, a kolone akcije drugog igrača. Svaki od igrača ima na raspolaganju određen broj strategija. Sve raspoložive akcije jednog igrača čine njegov strategijski skup. Ovde je naveden primer igre nulte sume u normalnoj formi:

	l	r
L	3	2
R	1	0

Objašnjenje: ako igrač 1 bira strategiju L, a igrač 2 bira strategiju l, tada će igrač 1 ostvariti dobitak 3, a igrač 2 dobitak -3, tj. gubitak 3.

Ova igra se može predstaviti i u ekstenzivnoj formi na sledeći način:



Igrač 1 kreće iz inicijalnog čvora i ima na raspolaganju dve strategije L i R, izborom jedne od njih dolazi se do sledećeg čvora, u kome je na potezu igrač 2. On ne zna izbor igrača 1 (igra je istovremena, pa su čvorovi odlučivanja igrača 2 obeleženi sivo) tako da mora doneti odluku za svaki čvor.

Neki tipovi igara omogućavaju lakše pronalaženje rešenja, a samim tim i viši stepen predvidivosti. U teoriji igara se razlikuju mnoge vrste igara i postoji mnoštvo različitih načina rešavanja, koji se uglavnom izvode za jednostavnije igre.

Nešov ekvilibrijum

Nešov ekvilibrijum je takav izbor strategija, da nijedan igrač nema interes da od njega odstupi. Jednostrano odstupanje ma kog igrača od Nešovog ekvilibrijuma dovodi do toga da taj igrač ostvaruje lošiji rezultat. U tom smislu Nešov ekvilibrijum definiše optimalne strategije za sve igrače. Dakle, kod igara nulte sume u prostoru čistih strategija Nešov ekvilibrijum je u stvari sedlasta tačka, a u prostoru mešovitih strategija on se poklapa sa minmaks ekvilibrijumom.

Radi lakšeg razumevanja, Nešov ekvilibrijum je ilustrovan sledećim primerom bima-trične igre dva igrača, tj. igre dva igrača sa promenljivom sumom. Data je matrica

	M	N
L	(6,4)	(5,3)
R	(4,2)	(3,1)

Ako igrač 2 odigra strategiju M, očigledno je da je za igrača 1 isplativije da odigra L ($6 > 4$). Ako pak igrač 2 odigra strategiju N, za igrača 1 je takođe isplativija strategija L ($5 > 3$). Racionalne strategije igrača 1 su podvučene u sledećoj tabeli

	M	N
L	(<u>6</u> ,4)	(<u>5</u> ,3)
R	(4,2)	(3,1)

Ako igrač 1 odigra strategiju L, igrač 2 će birati strategiju M, a ako igrač 1 odigra strategiju R, igrač 2 će birati strategiju M. Racionalne strategije igrača 2 su podvučene u narednoj tabeli

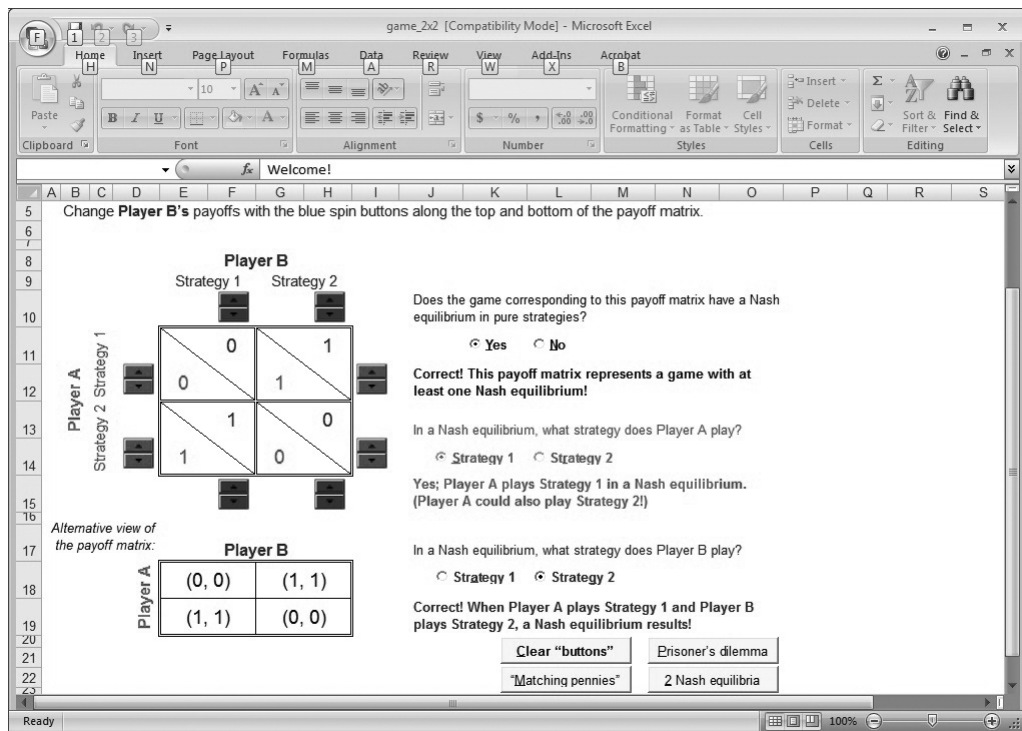
	M	N
L	(6, <u>4</u>)	(5,3)
R	(4, <u>2</u>)	(3,1)

Spajanjem ovih tabela dolazi se do tačke (tačaka) ekvilibrijuma ako ona postoji

	M	N
L	(<u>6,4</u>)	(<u>5</u> ,3)
R	(4, <u>2</u>)	(3,1)

Tačka (**6,4**) predstavlja uzajamno najbolja rešenja za oba igrača. Izborom strategije L igrača 1 i strategije M od strane igrača 2 dolazi se do Nešovog ekvilibrijuma.

Na sledećoj slici prikazana je implementacija Nešovog ekvilibrijuma u igri sa čistim strategijama i matricom plaćanja 2x2 u Excelu.



Primena teorije igara u ekonomskoj nauci

Ilustrovaćemo na nekoliko relativno jednostavnih primera kako može da izgleda primena teorije igara u analizama konkretnih ekonomskih problema. Postoje brojni radovi koji se bave konkretnom primenom teorije igara u ekonomskoj analizi, a brojne su i ekonomske oblasti u kojima se ostvaruje ova primena.

Ovde ćemo ilustrovati ovu primenu na nekoliko karakterističnih primera. *Nešova ravnoteža* se koristi u analizi brojnih konkretnih ekonomskih situacija: opis ponašanja monopola i različitih rešenja (ravnoteža) koje mogu da se ostvare, aukcije, problem javnih dobara, osiguranje, principal-agent problem i slično. Ovo nije konačan spisak tema nego samo neka probрана pitanja koja su u ekonomskoj literaturi najčešće prisutna. Izabrane su one teme koje od čitalaca ne zahtevaju nikakvo ekonomsko specijalističko predznanje i čija je analiza relativno jednostavna.

Duopoli - polazna situacija

Najpre ćemo pokazati na koji način se upotrebom teorije igara analizira konkurencija u slučaju tržišne strukture duopola. Pođimo od igre u kojoj učestvuju dve firme koje međusobno konkurišu na istom tržištu. Neka je data matrica isplate igre (brojevi označavaju profit jedne i druge firme). Nećemo sada ulaziti u ekonomsku logiku kojom se dolazi do

pojedinih rezultata. To je područje ekonomske teorije i zainteresovani čitalac može o tome da se informiše u knjigama koje se bave mikroekonomijom. Strategije predstavljaju izabrani nivo proizvodnje jedne i druge firme. Tako su prvom preduzeću na raspolaganju dve strategije, A_1 i A_2 , a drugom su na raspolaganju strategije, B_1 i B_2 .

Tabela 117. Matrica dobitaka

	B_1	B_2
A_1	(75, 75)	(70, 78)
A_2	(78, 70)	(72, 72)

Šta predstavlja rešenje, odnosno Nešovu ravnotežu ovakve igre? Odgovor nije jednoznačan. Važno je najpre primetiti da rešenje ove igre zavisi od toga da li se ona posmatra kao nekooperativna ili kao kooperativna. To drugim rečima znači, zavisi od pitanja da li je moguće (i da li je dozvoljeno) da se igrači unapred dogovaraju oko strategija koje će izabrati. Predstavićemo rešenje igre u oba slučaja.

Nekooperativno rešenje

Ukoliko ovu igru posmatramo kao nekooperativnu, možemo da primenimo jedan od nama poznatih načina njenog rešavanja. Na primer, možemo da primenimo kriterijum striktno dominacije, odnosno postupak eliminisanja strategije koja je dominirana nekom drugom strategijom. Pođimo od igrača 1. Vidimo na osnovu isplate igre da strategija A_2 dominira nad strategijom A_1 . Ona daje bolji rezultat za igrača 1 bez obzira na to kako će se ponašati igrač 2. S druge strane, kod igrača 2 strategija B_2 dominira nad strategijom B_1 . Nekooperativna ravnoteža ove igre duopola (Nešova ravnoteža) je kombinacija strategija (A_2 , B_2) sa ravnotežnom isplatom (72, 72). Oba igrača će ostvariti po 72 jedinice profita. To je ravnotežni rezultat igre. Čitalac sigurno primećuje (uporedite isplate) da ovo nije najbolje rešenje ni za jednog učesnika u igri. To je, međutim, nekooperativno rešenje do kojeg dovodi racionalno ponašanje (svako sledi svoj vlastiti interes) oba učesnika u igri. Ova ravnoteža, kako se to obično kaže, „stavlja samu sebe na snagu”.

Kooperativno rešenje

Posmatrajmo sada istu matricu igre, ali uz pretpostavku da je igra kooperativna. To znači da je procedura igre izmenjena. Kooperativno rešenje počiva na tome da igrači unapred mogu da se dogovore o načinu igre (o tome koje će strategije izabrati) i da se tog dogovora pridržavaju. Vidimo da ravnoteža do koje se dolazi kombinacijom strategija (A_1 , B_1) i koja donosi isplatu, odnosno ravnotežni rezultat (75, 75) predstavlja poboljšanje i za jednog i za drugog igrača u odnosu na prethodno nekooperativno rešenje igre kada je svaki igrač ostvarivao dobitak od 72 jedinice. Ekonomisti bi rekli da je ovo kooperativno rešenje igre superiorno (i to u tzv. *Paretovom* smislu) u odnosu na prethodno nekooperativno rešenje. Mi za sada samo konstatujemo da načelno postoji takvo kooperativno rešenje, ali

ne objašnjavamo na koji način se do njega stiže. Zanimarili smo i probleme koji se mogu pojaviti pri uspostavljanju kooperativnog rešenja u ovakvim situacijama.

Nejedinstveno rešenje

Podimo sada od nešto drugačije matrice igre. Ponovo imamo igru dva igrača (dve firme koje konkurišu na istom tržištu) sa po dve raspoložive strategije, ali je isplata igre nešto izmenjena u odnosu na prethodno posmatrani slučaj.

Tabela 118. Matrica dobitaka

	B_1	B_2
A_1	(65, 75)	(55, 88)
A_2	(72, 36)	(48, 32)

Tabela 119. Matrica dobitaka

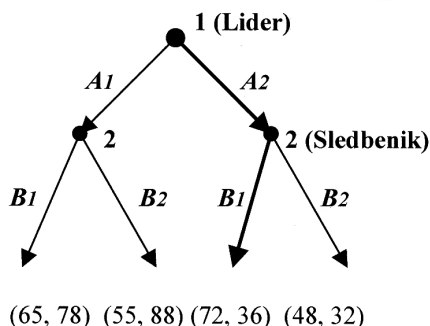
	B_1	B_2
A_1	(65, 75)	(55, 88)
A_2	(72, 36)	(48, 32)

Šta predstavlja Nešovu ravnotežu ove igre? Podimo ponovo od primene kriterijuma striktno dominacije. Lako uočavamo da u ovakvoj igri nema strategije koja je dominantna nad drugom strategijom ni za jednog ni za drugog igrača. Prinudjeni smo da primenimo Nešov kriterijum i potražimo šta predstavlja najbolji odgovor svakog igrača na fiksiranu (datu) akciju protivnika. Ukoliko igrač 1 izabere strategiju A_1 tada je najbolji odgovor za igrača 2 strategija B_2 . Označićemo posebno ovaj rezultat u matrici isplate igre za igrača 2, to jest isplatu (88). Ukoliko igrač 1 izabere strategiju A_2 , tada je najbolji odgovor za igrača 2 izbor strategije B_1 . Označićemo i ovaj rezultat za igrača 2 u matrici isplate igre. Sada ćemo ponoviti proceduru iz ugla igrača 1. Ukoliko igrač 2 izabere strategiju B_1 najbolji odgovor za igrača 1 jeste strategija A_2 . Označimo ovaj rezultat za igrača 1 (72) u matrici isplate igre. Ako igrač 2 izabere strategiju B_2 , tada je najbolji odgovor za igrača 1 strategija A_1 . Označimo ovaj rezultat za igrača 1 (55) u matrici isplate igre. Posmatrajmo sada matricu isplate igre gde su posebno označeni rezultati najboljih odgovora jednog i drugog igrača u odnosu na datu strategiju protivnika.

Kombinacije strategija koje dovode do ovih posebno naglašenih isplata predstavljaju Nešove ravnoteže igre. Vidimo da su ovi odgovori (strategije) jednog i drugog igrača uzajamno najbolji odgovori. Imamo, dakle, dve Nešove ravnoteže igre. Prva Nešova ravnoteža je kombinacija strategija (A_1 , B_2) sa ravnotežnom isplatom igre (55, 88), a druga Nešova ravnoteža je kombinacija strategija (A_2 , B_1) sa ravnotežnom isplatom (72, 36). Treba imati u vidu da nijedna od ovih ravnoteža nema neku prednost u odnosu na drugu. Drugim rečima, ne postoji kriterijum koji nam omogućuje da predvidimo jedinstveni rezultat ove igre.

Igra lidera i sledbenika

Predstavimo sada prethodno analiziranu igru u ekstenzivnoj formi, ali uz dodatnu pretpostavku da je igrač 1 u poziciji „lidera”, te da ima mogućnost da prvi bira svoju strategiju. Reč je o situaciji koju ekonomisti nazivaju Štakelbergov duopol (prema nemačkom ekonomisti Hajnrihu Štakelbergu koji je ovu situaciju prvi uveo u ekonomsku analizu). Da ne bude zabune, Štakelberg nije koristio teoriju igara, jer ona u to doba nije bila još razvijena. Njegova analiza, međutim, odražava logiku koju teorija igara koristi.



Slika 47. Štakelbergov duopol

Kako se dolazi do rešenja ove igre u ekstenzivnoj formi? Budući da je reč o igri sa savršenim i potpunim informacijama, moguće je da se primeni tzv. postupak povratne indukcije. To znači da treba da pođemo od racionalnog ponašanja igrača 2.

Ukoliko igrač 1 izabere strategiju A_1 tada je najbolji odgovor za igrača 2 izbor strategije B_2 . U ovom slučaju igrač 2 će ostvariti dobitak od 88 jedinica.

Ako igrač 1 izabere strategiju A_2 , tada je najbolji odgovor za igrača 2 izbor akcije B , koja mu donosi dobitak od 36 jedinica. Ovo nam je već poznato iz predstavljanja ove igre u matricnoj formi. Racionalan igrač 1 („lider”) anticipira ovakvo ponašanje igrača 2 i, shodno tome, prilagođava vlastiti izbor. Izborom strategije A_1 a uz očekivano racionalno ponašanje igrača 2, on može da očekuje dobitak od 55 jedinica. Ako izabere strategiju A_2 , tada može, takođe uz anticipirano ponašanje igrača 2, da očekuje dobitak od 72 jedinice. Vidimo da mu kriterijum racionalnosti nalaže da između ove dve opcije izabere onu koja mu donosi veći dobitak - to je strategija A_2 . Tako je Nešova ravnoteža igre kombinacija strategija (A_2, B_1) i isplata igre (72, 36).

Ova igra predstavlja dobru ilustraciju situacije u kojoj mogućnost da se igra prvi može da bude značajna prednost u igri. To ne znači da je u svim igrama takav slučaj.

Borba za tržište

U jednom gradu nalaze se dve radio-stanice koje treba da odluče koji tip muzike će emitovati. Cilj im je, sasvim razumljivo, da ostvare što veće tržišno učešće. Pođimo od toga da

obe radio- stanice mogu da emituju rok, džez ili bluz. Igra je sa konstantnom sumom, tako da porast tržišnog učešća jednog igrača ide na štetu drugog. Imamo sledeću matricu igre:

Tabela 120. Borba za tržište

	Rok	Džez	Bluz
Rok	(55, 45)	(30, 70)	(50, 50)
Džez	(75, 25)	(50, 50)	(70, 30)
Bluz	(50, 50)	(25, 75)	(45, 55)

Igra može da se reši već poznatim postupcima (Nešov kriterijum) i njeno rešenje je kombinacija strategija „džez-džez”, koja će rezultirati podelom tržišta. Važno je da se primeti da se ovakva igra koja je nenulte ali konstantne sume (zbir isplata jednak je 100), može transformisati u igru s nultom sumom. To se postiže tako što isplate igre mogu da predstavljaju tržišno učešće koje je iznad polovine. Tako se matrica igre može transformisati u narednu matricu (isplate predstavljaju dobitak odnosno gubitak igrača 1).

Tabela 121. Borba za tržište - kao igra nulte sume

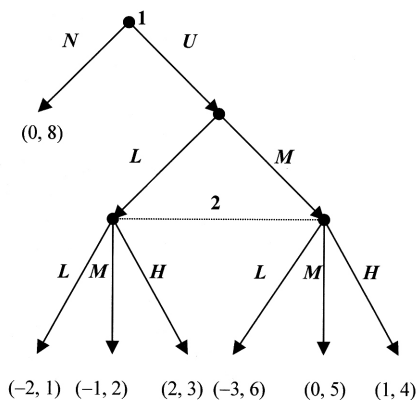
	Rok	Džez	Bluz
Rok	5	-20	0
Džez	25	0	20
Bluz	0	-25	-5

U rešavanju ovakve igre primenjuje se Fon Nojmanovo rešenje (*maximin* kriterijum). Rešenje je upravo ono koje smo prethodno utvrdili (džez-džez). Mogućnost transformacije jedne igre u igru sa nultom sumom ponekad može bitno da olakša njeno rešavanje. Razvijeni su programi za rešavanje ovakvog tipa igara i oni su posebno korisni kada je reč o igrama gde je broj raspoloživih strategija igrača velik.

Igra ulaska na tržište (engl. entry game)

Firme se često susreću s problemom da li ući na neko već postojeće tržište ili ne. U analizama ovog problema obično se koriste dinamičke igre. Pođimo od situacije gde imamo jednog starosedeoca na tržištu (*incumbent*) — igrač 2 i jednog potencijalnog pridošlicu (*entrant*) — igrač 1. Pretpostavimo da se igra odvija na sledeći način: pridošlica može da odluči da ne uđe na tržište — strategija N ili da uđe - strategija U. Ukoliko ne uđe na tržište, ne izlaže se troškovima, ali ne ostvaruje ni dobitak, te je vrednost isplate u tom slučaju za njega jednaka nuli, a dobitak starosedeoca je recimo 8. Ako pridošlica uđe na tržište mora da se odluči koju će strategiju cena voditi: može da nastupi s niskom cenom — strategija L ili s nekom srednjom cenom — strategija M. Pretpostavimo da starosedelac može

da bira jednu od tri strategije cena: niska L, srednja M i visoka H. Igrači moraju da ovu odluku donesu istovremeno, tako da je reč o igri sa nesavršenim informacijama. Različite kombinacije strategija cena jednog i drugog igrača rezultiraju odgovarajućim isplatama igre. Imamo, dakle, igru sledećeg oblika.



Slika 48. Igra ulaska na tržište

Da bismo rešili ovu igru neohodno je da najpre rešimo njen deo koji sledi nakon ulaska pridošlice na tržište. Transformisaćemo taj deo igre u normalnu formu.

Tabela 122. Normalna forma „igre cena”

	H	M	L
M	(1, 4)	(0, 5)	(-3, 6)
L	(2, 3)	(-1, 2)	(-2, 1)

Kada primenimo Nešov kriterijum dolazimo do toga da je ravnoteža ovog dela igre kombinacija strategije L igrača 1 i strategije H igrača 2 s ravnotežnim rezultatom (2, 3). Pridošlica poredi isplatu koju ostvaruje u ovom slučaju (dobitak od 2 novčane jedinice) s rezultatom kada ne ulazi na tržište (dobitak 0). Na osnovu toga donosi odluku da je za njega bolje rešenje strategija ulaska na tržište. Iz toga sledi da je ravnotežna putanja igre predstavljena strategijama U-L-H.

Primena karakterističnih igara u međunarodnoj ekonomiji

U nastavku ovog poglavlja ilustrovaćemo primenu teorije igara u analizama međunarodnih ekonomskih odnosa.

Reprogramiranje duga

Velike dužničke krize koje su potresale svetsku privredu dovele su do porasta intere-

sovanja analitičara za problem reprogramiranja duga. U tim analizama je pored ostalog primenjivana i teorija igara. Kada se modelira ovakva igra neophodno je brižljivo utvrditi ko su igrači, koje strategije su im raspoložive, kontekst njihovih strateških akcija, kao i kako akteri vrednuju moguće rezultate koji nastaju kao posledica izbora svih strana koje učestvuju u igri.

Posmatrajmo sledeću pojednostavljenu igru reprogramiranja duga u kojoj učestvuju zemlja-dužnik (igrač 1) i zemlja-poverilac (igrač 2). Suočeni sa ozbiljnim pritiskom na-raslog duga poverilac i dužnik mogu da se odluče da pregovaraju o reprogramiranju. Pri tome, igrač 1 može da izabere jednu od dve strategije: može da se odluči da brzo i radikalno prilagodi ponašanje (ekonomsku politiku) — strategija C_1 ili da to prilagođavanje bude zanemarljivo (neke minorne promene) - strategija D_1 . Igrač 2 ima na raspolaganju takođe dve strategije: da se odluči na velike ustupke (da nastavi s kreditiranjem, da otpiše jedan deo duga i slično) — strategija C_2 , ili da ne pristaje ni na kakve veće ustupke (nema promena uslova niti novih kredita) — strategija D_2 .

Kombinacija ovih strategija rezultira sledećim isplatama igre: ukoliko se dužnik odluči na brzo i radikalno prilagođavanje privrede u cilju stvaranja neophodnih pretpostavki za vraćanje duga, a poverilac mu učini određene olakšice ili odobri nova sredstva, doći će do uspešnog reprogramiranja dugova. Ako se dužnik ponaša tako što ignoriše potrebu za radikalnom reformom privrede (vrši neke male kozmetičke promene), a poverilac nastavi sa prilivom sredstava, reč je zapravo o odbijanju plaćanja duga (*debt repudiation*). Ako poverilac nije spreman na ustupke, a dužnik se odlučuje na radikalne izmene u privredi, dolazi do jednostranog prilagođavanja. Konačno, ako poverilac nije spreman na ustupke, a ni dužnik nije spreman na promene, dolazi do sloma pregovora.

Tabela 129. Igra reprogramiranja duga

	Veliki ustupci	Mali ustupci
Radikalno prilagođavanje	Uspešno reprogramiranje	Unilateralno prilagođavanje
Blago prilagođavanje	Odbijanje plaćanja duga	Slom pregovora

Kako će izgledati konkretni brojevi u ovoj igri zavisi od toga na koji način jedna i druga strana procenjuju ove moguće ishode.

Blefiranje

Podimo od sledeće situacije. Dužnik, na primer, može da proceni da će izdržati i bez dodatnih sredstava poverioca, to jest da je za njega manje bolno da ostane bez tih dodatnih sredstava nego da sprovodi radikalne ekonomske mere. Pored toga, dužnik je spremniji od poverioca da prihvati situaciju sloma pregovora. S druge strane, ako se zemlja dužnik prilagođava, poverilac smatra da je bolje ne daje dodatne kredite već da čeka da se vrati već postojeći dug. Matrica isplate igre ima u ovom slučaju sledeći oblik:

Tabela 130. Igra reprogramiranja duga - blefiranje

	C_2	D_2
C_1	(3, 3)	(1, 4)
D_1	(4, 2)	(2, 1)

Ovde nije reč o simetričnoj igri. Igra ovakve strukture naziva se „blefiranjem”. Zapravo, sa stanovišta poverioca igra ima oblik igre „pilića”, a sa stanovišta dužnika reč je o igri „zatvorenikove dileme”. Kada se pogleda struktura isplate igre, lako se primećuje da je strategija D_1 dominantna za dužnika. Za poverioca u ovakvoj situaciji bolje rešenje predstavlja izbor strategije C_2 (da igra popustljivo) nego strategije D_2 . Vidimo da je rezultat (D_1, D_2) najnepovoljniji za poverioca, ali ne i za dužnika. Poverilac može da preti da će odigrati strategiju D_2 , ali takva njegova pretnja je u ovakvoj strukturi igre prazna — blefiranje. Tako će se dužnik odlučiti da ne ispunjava obaveze (na ovaj način se ponašala Argentina 1984. godine), a poverilac će igrati pomirljivo, te će konačan rezultat igre biti (4, 2).

„Pilići”

Podimo sada od toga da igrači vrednuju moguće ishode igre nešto drugačije, to jest da igra ima sledeći oblik:

Tabela 131. Igra reprogramiranja duga - „pilići”

	C_2	D_2
C_1	(3, 3)	(2, 4)
D_1	(4, 2)	(1, 1)

Ovde je reč o tzv. klasičnoj igri „pilića”. Ovakva igra se zasniva na tome da jedna strana nastoji da prisili drugu da „kapitulira”. To jasno uočavamo iz simetričnih isplata (4, 2) i (2, 4). Takođe, kompromisno rešenje igre (3, 3) vrednuje se kao bolje od nekooperativnog (1, 1). Mi već znamo da ovakva igra ima dve Nešove ravnoteže u čistim strategijama, to jest znamo da nije moguće predvideti njeno konačno rešenje. U ovakvom slučaju bitna je reputacija koju uživaju igrači, to jest da li su poznati kao „tvrđi” ili kao „meki”. Ukoliko se zna da se igra s nepopustljivim igračem, bolje je odigrati popustljivo. I obrnuto. O temi reputacije igrača čitalac može potražiti više informacija u stručnoj literaturi.

Multinacionalna kompanija protiv lokalne firme

Ova igra ima oblik koji je u stručnoj literaturi poznat kao igra „racionalnih svinja”. Posmatrajmo situaciju gde imamo dve firme: igrač 1 je multinacionalna kompanija (snažniji igrač), a igrač 2 je lokalna kompanija koja proizvodi samo za domaće tržište (slabiji igrač). Obe strane razmišljaju o ulasku na neko treće tržište. Podimo od pretpostavke da su troškovi ulaska na treće tržište jednaki za oba igrača uz napomenu da su ovi troškovi veći ukoliko firma ulazi prva na tržište i manji su ukoliko firma ulazi nakon što je njen protivnik već na tom tržištu. Ukupan iznos profita koji može da se zaradi na ovom trećem tržištu fiksiran je. Kombinacija strategija igrača i njihove isplate je data sledećom tabelom:

Tabela 132. Kompanija protiv lokalne firme

	Ulazi prvi	Ulazi drugi
Ulazi prvi	(5, -1)	(4, 4)
Ulazi drugi	(9, -1)	(0, 0)

Šta predstavlja Nešovu ravnotežu ovako strukturirane igre? Vidi se lako da je ravnotežno rešenje da multinacionalna kompanija prva uđe na novo tržište, a da je lokalna kompanija nakon toga prati (4, 4). Primenite kriterijum eliminacije strategije koja je dominirana nekom drugom strategijom. Vidimo da je za igrača 2 (lokalnu kompaniju) strategija „ulazi drugi” striktno dominantna u odnosu na strategiju „ulazi prvi”. Ekonomska logika koja stoji iza ovakvog ponašanja lako je razumljiva i nema potrebe da je dodatno objašnjavamo.

Igra jastreba i goluba

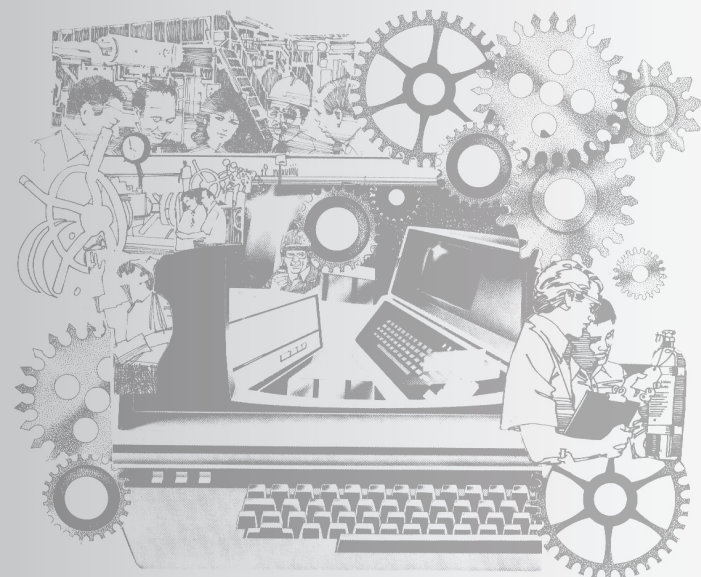
Posmatrajmo na kraju situaciju u kojoj dve kompanije mogu da se odluče da „žive mirno” ili „da napadnu” jedna drugu na tržištu koje kontroliše onaj drugi. Strategije igrača su da se ponašaju agresivno („napadaj”) ili kooperativno („pusti druge da žive”). Kombinacije strategija rezultiraju i odgovarajućim isplatama za jednog i drugog igrača (prvi član u ovim parovima je dobitak igrača čije su strategije horizontalne, a drugi član je dobitak igrača čije su strategije date vertikalno).

Tabela 133. „Jastreb protiv goluba”

	Napad	Saradnja
Napad	(-1, -1)	(2, 0)
Saradnja	(0, 2)	(1, 1)

Vidimo iz matrice isplate igre da je najnepovoljnija situacija ukoliko se obe strane ponašaju beskompromisno (-1, -1). Tada će doći do iscrpljujućeg međusobnog rata koji će završiti gubitkom za oba igrača. Primećujemo takođe da se strategija agresivnosti isplati samo kada je protivnik spreman da bude kooperativan. Šta je rešenje ove igre? Primena kriterijuma eliminacije striktno dominiranih strategija u ovom slučaju ne daje rezultat. Stoga je neophodno da se primeni Nešov kriterijum — pronalaženje najboljeg odgovora na datu strategiju protivnika. Primajući taj postupak utvrđujemo da postoje dve Nešove ravnoteže igre i, shodno tome, dva ravnotežna rezultata: (2, 0) i (0, 2). Nema, kao što znamo odranije, jedinstvenog rešenja ove igre u čistim strategijama. Jedinstveno rešenje ove igre može da se pronađe samo u mešovitim strategijama. Tako se objašnjava ponašanje koje se često primećuje u privrednom životu, gde se ista firma u različitim situacijama različito ponaša: ponekad igra agresivno, a ponekad se odlučuje da sarađuje. Čitalac može sam da potraži rešenje ove igre u mešovitim strategijama.





13. OBUČAVANJE

Proces obučavanja sastoji se uvek u prenošenju znanja ili metoda rešavanja zadataka od onoga koji obučava (učitelja) onome koji se obučava (učeniku). Obučavanje u rešavanju zadataka može se ostvariti dvema suštinski različitim metodama. Prvi metod se sastoji u saopštavanju učeniku algoritma rešavanja zadataka. Drugi metod se sastoji u obučavanju pomoću primera. Tako se obučavanje ljudi ili mašina da obavljaju aritmetičke ili logičke operacije postiže tumačenjem algoritma njihovog izvršavanja, ili sastavljanjem programa koji realizuje taj algoritam. Ali metode rešavanja zadataka kao što su, na primer, čitanje (raspoznavanje) slova ili cifara, odštampanih raznim pismima ili napisanih rukom, učitelj prenosi učeniku ne tumačenjem strukture tih simbola, ili opisivanjem mehanizma raspoznavanja koji je osim toga i učitelju nepoznat, *već obučavanjem na primerima*. Pri tome je važno da se, usvojivši neki ograničen broj primera objekata koje treba raspoznavati, učenik osposobljava da pravilno raspozna nove objekte, koji mu nisu bili pokazani za vreme obučavanja.

Obučavanje na primerima ima ogroman značaj za sposobnost mnogih vrsta živih organizama da žive.

Obučavajući potomstvo da pronalazi hranu i da izbegava opasnost, životinje mu ne saopštavaju algoritam rešavanja tih zadataka, već koriste metod obučavanja na primerima. Veliki deo iskustva i metoda koji omogućuje ljudima da rešavaju raznovrsne životne zadatke, isto tako se stiče posmatranjem, po analogiji, bez nalaženja algoritma.

Do sada rešavanje raznovrsnih zadataka pomoću automata, uključujući tako savršene automate kao što su računari (cifarske mašine), ostvarivano je na osnovu programa koji su sadržavali u očiglednom vidu algoritam rešavanja svakog zadatka. Može se shvatiti da čovek, određujući strukturu automata ili razrađujući program za računar, priprema *obučavanje* za rešavanje zadataka određenog tipa, koristeći prvi metod obučavanja. Taj metod je, međutim, beskorisan za zadatke, čiji nam je algoritam rešavanja nepoznat, makar i da ih umemo rešiti intuitivno. Čovek, na primer, lako nauči da razlikuje mačku od psa, ili da prepozna svog poznanika, ili da hvata loptu u letu, ali još uvek ima dosta problema da sastavi program po kome bi mašina mogla da učini to isto.

Težnja ka proširenju mogućnosti mašina pobudila je naučnike i inženjere da pokušaju da konstruišu mašine, sposobne da se obuče na primerima za rešavanje različitih zadataka.

Prepoznavanje likova

Prvi uspešan pokušaj u obučavanju mašine na primerima bio je konstruisanje automata, sposobnog da se obuči za prepoznavanje takvih vizuelnih likova, kao što su geometrijske figure, slova, cifre i drugi simboli. Zadatak prepoznavanja/raspoznavanja likova sastoji se u sledećem. Postoji mnoštvo X , koje sadrži veoma veliki broj n različitih objekata

$$X = (x_1, x_2, \dots, x_n),$$

koji pripadaju relativno malom broju m zadatih klasa $X_1, X_2, \dots, X_m$. Smatraćemo da automat A rešava zadatak raspoznavanja, ako on uvek (ili, barem, s dovoljno visokom pouzdanošću) ubraja svaki objekt x u zadatu klasu — lik X_i .

Ako, na primer, mnoštvo X predstavlja skup svih mogućih oblika cifara, tada se zadatak raspoznavanja sastoji u tome, da se svaki oblik ubroji u jednu od deset klasa: 0, 1, 2, ..., 9, od kojih svaka predstavlja *lik* odgovarajuće slike. Mnoštvo X može da sadrži različite geometrijske figure, među kojima treba razlikovati figure koje pripadaju klasi trouglova, klasi kvadrata, klasi krugova i slično.

Automat, predviđen za raspoznavanje likova, treba, očigledno, da ima ulazni uređaj za prijem informacije o objektu za raspoznavanje. Ulazni uređaj takvog automata naziva se *receptorno polje*. On može da predstavlja mozaik fotoelemenata, na koji se projektuje slika za raspoznavanje. Ako se receptorno polje sastoji od r elemenata, od kojih se svaki može nalaziti u jednom od dva stanja (pobuđenom ili nepobuđenom), tada je broj svih mogućih konfiguracija na ulazu jednak $n = 2^r$. Broj svih mogućih konfiguracija n , čak i pri relativno malom r (reda deset), postaje toliko veliki (reda milijardi), da je sa aspekta praktične primene problematično smestiti u memoriju mašine podatke o svakoj pojedinoj konfiguraciji.

Izlazni uređaj automata, koji raspoznaje, treba da sadrži M izlaza. Na osnovu toga koji je od izlaza pobuđen, može se zaključiti u koju klasu automat ubraja predstavljeni mu objekt.

Da bi automat mogao da se obuči, on treba da poseduje dovoljno veliki broj N mogućih unutrašnjih stanja $z = (z_1, z_2, \dots, z_N)$, među kojima treba da postoje takva stanja, u kojima automat A klasifikuje objekte na zahtevani način.

Zadatak obučavanja takvog automata sastoji se u tome da se, predočavanjem automatu relativno malog broja $l \leq n$ primera ubrajanja objekata x u određene klase X_i , prevede automat A u jedno od takvih stanja z^* , u kojima će on ostvariti zahtevanu klasifikaciju objekata, uključujući i one objekte koji mu nisu bili predočeni u procesu obučavanja. Pokazuje se da, ako se izabere takvo stanje automata da on pravilno klasifikuje dovoljno veliki broj s slučajno izabranih objekata iz mnoštva X , tada će on s određenom verovatnoćom dovoljno dobro klasifikovati i sve objekte datog mnoštva X .

Odatle se zaključuje, da se proces obučavanja automata A za raspoznavanje likova u opštim crtama sastoji u sledećem.

Automatu A (sl. 13.1), dejstvom na njegovo receptorno polje, predočava se l objekata, slučajno izabranih za obučavanje iz mnoštva X . Pri tome učitelj, etalon-automat A_e (ulogu etalon-automata može da igra i čovek) pokazuje automatu A kojoj klasi pripada svaki od l objekata niza obučavanja $x_{j1}, x_{j2}, \dots, x_{jl}$.

Uređaj koji obučava UO u procesu obučavanja poredi reakcije y_e etalon-automata s reakcijama y obučavanog automata A i tako menja stanje z automata A , da bi se njegova reakcija što je moguće češće podudarala s reakcijama A_e . Zatim se automatima predočava i slučajno odabrani ispitni

niz objekata $x_{k1}, x_{k2}, \dots, x_{ks}$. Ako na ispitnom nizu automat ne napravi više grešaka nego što je dozvoljeno, tada se proces obučavanja smatra završenim; u suprotnom slučaju, izvodi se doučavanje automata A sa sledećim ispitom (sesijom). Takav postupak se nastavlja dok se ne postigne potrebna pouzdanost raspoznavanja, ili dok se ne razjasni, da dati automat nije sposoban da se obuča za raspoznavanje datih objekata sa zahtevanom tačnošću.

Efikasnost procesa obučavanja suštinski zavisi od tri faktora: a) od načina prikazivanja objekta konfiguracijom na receptornom polju automata, tj. od toga, koje se zapravo karakteristike realnih objekata dostavljaju na ulaz automata A i kako se one kodiraju, b) od raznovrsnosti pretvaranja ulazne konfiguracije u izlaznu reakciju automata, tj. od broja njegovih mogućih različitih stanja i c) od algoritma rada uređaja koji obučava UO .

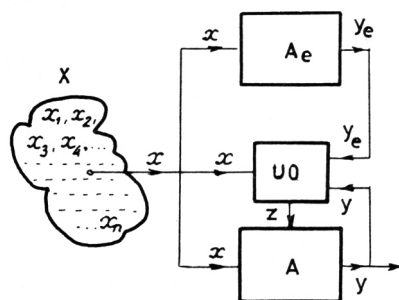
Prikaz treba da se bira tako, da se na ulazu klasifikujućeg uređaja nalazi informacija, dovoljna za klasifikaciju. Raznovrsnost pretvaranja treba da bude dovoljno bogata, da bi automat bio sposoban da se obuča za rešavanje dovoljno široke klase zadataka, ali ne suviše velike, da ne bi nastale prevelike teškoće prilikom nalaženja potrebnog pretvaranja. Algoritam rada uređaja koji obučava treba da obezbedi, što je moguće veću, pouzdanost raspoznavanja ili, što je moguće manju, dužinu niza obučavanja.

Zadatak obučavanja mašina za raspoznavanje likova može se interpretirati geometrijski. Svaku konfiguraciju na receptornom polju, koja se sastoji iz r receptora, može prikazivati (predstavljati) tačka u r -dimenzionalnom prostoru receptora, duž čijih se osa $a_1, a_2, \dots, a_r$ nanosi stepen pobuđenosti svakog receptora.

Ako mnoštvo objekata X podelimo na zadate klase, tada je u prostoru receptora moguće konstruisati površine koje dele taj prostor na oblasti tako, da se u svakoj od njih nalazi tačka koja prikazuje objekte samo jedne klase. Za slučaj podele mnoštva X na dve klase X_1 i X_2 zadatak obučavanja za raspoznavanje može se posmatrati kao zadatak određivanja jednačine

$$F(a_1, a_2, \dots, a_r) = 0$$

površine, koja deli prostor receptora na dve takve oblasti, da se svi objekti koji pripadaju klasi X_1 nađu u oblasti



Sl. 13.1. Šema obučavanja automata radi raspoznavanja likova

$$F(a_1, a_2, \dots, a_r) > 0,$$

a objekti, koji pripadaju klasi X_2 , — u oblasti

$$F(a_1, a_2, \dots, a_r) < 0,$$

kao što je šematski pokazano na sl. 13.2. Svako stanje z automata A može da se posmatra kao jedan od mogućih oblika podeone površine, i obučavanje se sastoji u izboru jedne od površina koje obezbeđuju podelu objekata na klase X_1 i X_2 , dovoljno blisku podeli koju ostvaruje automat A_e .

Danas je razrađen veliki broj algoritama obučavanja mašina za raspoznavanje likova. Svi su oni zasnovani na poređenju izlaza obučavanog automata i etalon-automata i na prevođenju automata koji obučava (promenom njegovih parametara) u takva stanja, u kojima se broj razilaženja izlaza automata A i A_e smanjuje.

Kao primer najprostijeg algoritma podele objekata u dve klase X_1 i X_2 može da posluži sledeća procedura. Predstavnici dveju klasa kodiraju se binarnim kodom. Zatim se dobijene kodne kombinacije zapisuju u vrste jedne ispod drugih u obliku tabele. Pretpostavlja se, da postoji izvestan broj predstavnika klasa X_1 i X_2 , koji se biraju za obučavanje i čine niz obučavanja. Za svaku kolonu tabele s nizovima obučavanja izračunava se empirijska (iskustvena) učestanost pojave 1 ili 0. Na primer, ako su u prvoj koloni jedna jedinica i devet nula, tada je učestanost pojave jedinica $p_1 = 1/10$, a nule $p_0 = 9/10$. Zatim se uzimaju logaritmi tih veličina i pravi horizontalna tabela s brojem ćelija jednakim dužini kodne kombinacije. U gornji red se zapisuje razlika

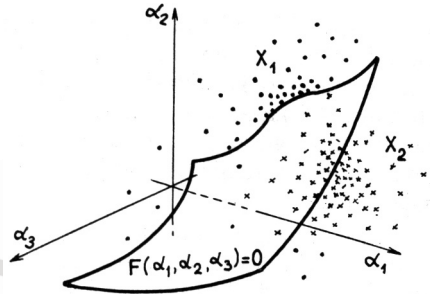
$$R_0 = \log p_{01} - \log p_{02},$$

a u donji — razlika

$$R_1 = \log p_{11} - \log p_{12},$$

izračunata za svaku kolonu tabele sa nizovima obučavanja klasa X_1 i X_2 . Procedura stvaranja takve tabele oponaša proces obučavanja u mašini. Posle toga počinje ispit. Gotova tabela se pridružuje uz tabelu s kodnim kombinacijama, tako da se poklope brojevi njihovih ćelija po horizontali i izvodi se sledeća operacija: iz reda R_0 tabele u početku se prepisuju brojevi kojima u kodu odgovara nula, a zatim iz reda R_1 — brojevi, kojima odgovara jedinica. Svi ti brojevi se sabiraju. Ako je dobijena suma pozitivna smatra se da dati objekt pripada klasi X_1 , a ako je negativna — klasi X_2 .

Teorija i tehnika konstruisanja sistema, obučavanih za raspoznavanje likova, je u stalnom razvoju. Ono što se može konstatovati, danas postoje uspešno konstruisu uređaji i programi za ovu svrhu. Zahvaljujući tome, uspešno se rešavaju tako teški zadaci obučavanja.



Sl. 13.2. Površina, koja u prostoru receptora odeljuje likove različitih klasa

vanja mašina, kao što su raspoznavanje rukopisnih slova i cifara, interpretacija geoloških podataka, raspoznavanje govornih glasova i dijagnostika nekih bolesti. Pri tome se mašina u nizu slučajeva pokazala sposobnom da nauči da rešava zadatke raspoznavanja mnogo bolje, nego što je to dostupno čoveku.

Obučavanje u ponašanju

Kao model sistema obučavanog u ponašanju, pogodno je posmatrati slučajni automat A. Ponašanje automata A se karakteriše verovatnoćom određene reakcije, koja se opaža na izlazu automata na pobudu, dovedenu na njegov ulaz. Ograničavajući se na proste situacije, kada su poznati konačan broj mogućih pobuda $x_1, x_2, \dots, x_n$ i sve moguće reakcije $y_1, y_2, \dots, y_m$, može se opisati ponašanje posmatranog automata matricom verovatnoća M (tabela 13.1).

Tabela 13.1

	y_1	y_2	...	y_m
x_1	p_{11}	p_{12}	...	p_{1m}
x_2	p_{21}	p_{22}	...	p_{2m}
...	...	...	...	...
x_n	p_{n1}	p_{n2}	...	p_{nm}

Elementi p_{ij} te matrice karakterišu verovatnoću datog automata A da u određenom stanju z odgovori reakcijom y_j na pobudu x_i . Ako je skup $y_1, y_2, \dots, y_m$ potpuni skup svih mogućih reakcija i ako na ulaznu pobudu automat uvek odgovara nekom izlaznom reakcijom, tada je suma verovatnoća svih reakcija na svaku pobudu jednaka jedinici

$$\sum_{j=1}^{j=m} p_{ij} = 1 \quad (i = 1, 2, \dots, n).$$

Svakom od N mogućih stanja $z_1, z_2, \dots, z_N$ automata A odgovara određena matrica $M(z)$, koja određuje njegovo ponašanje.

Svaki određeni akt ponašanja automata A određuje se parom (x_i, y_j) , tj. j -tom reakcijom na i -tu pobudu, i svakom takvom paru može se dodeliti odgovarajuća ocena a_{ij} , koja karakteriše efikasnost ponašanja automata. Tada se obučavanjem automata može smatrati njegov prelaz u takva stanja, u kojima se verovatnoće dobijanja visokih ocena povećavaju.

Opisani model obučavanog automata pokazao se korisnim za proučavanje procesa obučavanja određenim formama ponašanja čoveka i životinja, a i za konstruisanje veštačkih upravljačkih uređaja za obučavanje.

Pokazano je, da se slučajni automati mogu obučiti svrsishodnom ponašanju, ako ih »bodri« ili »kažnjava« učitelj ili sredina, i ako se pri tome svaki put, kada je posle određenog prelaza automat »kažnjen«, verovatnoća prelaza u to stanje smanjuje, a u sva ostala

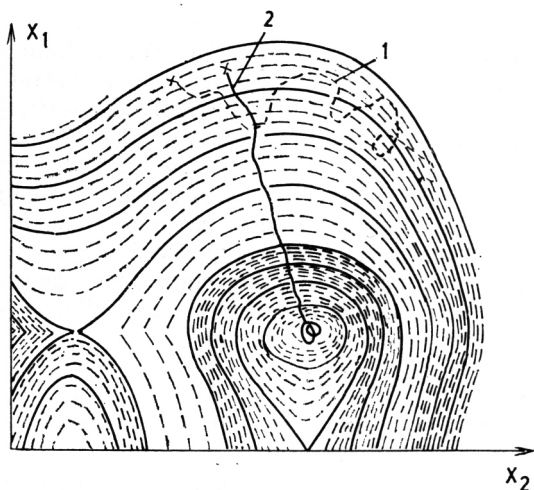
— povećava. Na taj način se uspelo, naročito, da se obuču slučajni automat brzom nalaženju minimuma funkcije dveju promenljivih. (Na takav zadatak se svodi traganje za najpovoljnijim režimom sistema, čija efikasnost dejstva zavisi od dva upravljačka dejstva.) Automat odabira smer i veličinu brzine pomeranja reprezentativne tačke u ravni (x_1, x_2) (sl. 13.3) u zavisnosti od promene funkcije

$$y = R(x_1, x_2).$$

Nivoske linije y pokazane su na sl. 13.3. Kažnjavanje automata je utoliko manje, ukoliko se reprezentativna tačka približava minimumu za y . Izvedeni eksperimenti su pokazali efikasnost takvog sistema obučavanja, o čemu se može suditi poredeći trajektoriju 1 kretanja reprezentativne tačke pre obučavanja automata, sa trajektorijom 2 — posle njegovog obučavanja. Iz sl. 13.3 se vidi da je automat mogao da se obuču da radi veoma usredsređeno na postizanju određenog cilja — pronalaženja potrebnih vrednosti promenljivih x_1 i x_2 .

Analogne metode se mogu koristiti pri obučavanju automata za rešavanje mnogih složenih zadataka i svrsishodno ponašanje u različitim složenim situacijama.

Efikasnost obučavanja automata ponašanju može se znatno povisiti, ako se između etape obučavanja na primerima i etape ispita predvidi etapa *treniranja*. Treniranje automata može da se sastoji, na primer, u višestrukom proveravanju dejstva automata sa jednim istim nizom obučavanja, a reakcija sredine ili podsticanje i kažnjavanje od strane učitelja koriste se za doučavanje automata potrebnom ponašanju.



Sl. 13.3. Traganje automata za minimumom funkcije dveju promenljivih $y = R(x_1, x_2)$

Obučavanje složenim formama ponašanja, ne samo automata već i ljudi, bilo bi nemoguće, ako bi onaj koji se obučava trebalo, radi dobijanja konačnog rezultata, da usvoji postupke složenog ponašanja samo na osnovu podsticanja ili kažnjavanja. Zaista, da bi automat bio sposoban da se obuču složenom ponašanju — rešavanju složenih zadataka, treba i sam da bude veoma složen, da broj njegovih mogućih stanja bude vrlo veliki. Ali, u tom slučaju bi pronalaženje, u ogromnom mnoštvu stanja, jednog od onih malobrojnih, u kojima taj složeni automat ima potrebne osobine, predstavljalo zadatak skopčan sa izvanrednim teškoćama, usled posebne težine analize stanja automata.

Taj se zadatak, međutim, može bitno olakšati, ako se u procesu obučavanja koriste usputni ciljevi, koji dopuštaju razbijanje obučavanja u etape. Tada se u prvim etapama obučavanja automat uču da rešava relativno jednostavne zadatke, a zatim se na osnovu već

formiranih u automatu struktura, prilagođenih za rešavanje prostih zadataka, izvodi obučavanje za rešavanje sve složenijih zadataka na svakoj novoj etapi dotle, dok se ne postigne konačan cilj. Takva taktika obučavanja automata podseća na uobičajeni način obučavanja ljudi, kojima se na svakoj etapi obučavanja postavljaju sve teži zadaci, koji odgovaraju učeniku, zahvaljujući veštini stečenoj u prethodnim etapama obučavanja.

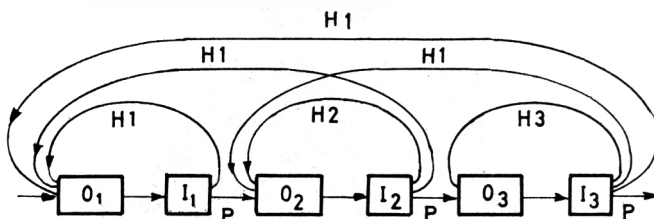
Mašina za obučavanje

Obučavanje ljudi sastoji se u organizaciji takvog procesa, da u njegovom rezultatu čovek usvaja neke činjenice i ovladava spretnošću, što mu omogućuje da koristi svoje znanje za rešavanje različitih zadataka. To mogu biti jednostavni zadaci, kao što su prepisivanje/prekucavanje teksta, ili složeniji zadaci — prevod teksta s jednog jezika na drugi, ili određivanje neispravnosti nekog tehničkog uređaja ili, na kraju, rešavanje matematičkih zadataka. Ma kakav bio karakter delatnosti za koju se čovek priprema obučavanjem, njegov cilj se sastoji u skupljanju nekih znanja i »veština«.

Do pre deceniju-dve ljude su obučavali *samo ljudi*, koji su retko upotrebljavali za obučavanje tehnička sredstva u vidu filmova, uređaja za snimanje i reprodukciju zvukova i očiglednih učila tipa računara i specijalizovanih simulatora. U svetu savremenih dostignuća proces obučavanja se može posmatrati kao neki upravljani proces, koji se odvija prema zakonima svojstvenim za upravljane procese; proces, koji se može u određenom stepenu optimizovati i automatizovati pomoću odgovarajućih tehničkih sredstava.

Tehnička sredstva, kojima se putem uzajamnog delovanja s učenicima bez neposrednog učešća učitelja ostvaruju određene etape obučavanja, nazivaju se mašine za obučavanje.

Proces obučavanja obično se izvodi prema šemi datoj na sl. 13.4. Šema se sastoji iz elemenata tipa *O*, koji prikazuju etape obučavanja, elemenata tipa *I*, koji prikazuju ispite, i veza, koje pokazuju povezanost etapa obučavanja. Na šemi se nalaze dva tipa veza: *P* — pravilan odgovor na ispitu i *N* — nepravilan odgovor. Kao što se vidi sa sl. 13.4, niz obučavanja zavisi od rezultata ispita i dozvoljava prelaz na sledeću etapu obučavanja, samo pri pozitivnim rezultatima ispita. U suprotnom slučaju, ponavljaju se prethodne etape obučavanja dotle, dok se izučavani materijal dovoljno dobro ne asimilira.



Sl. 13.4. Šema procesa obučavanja

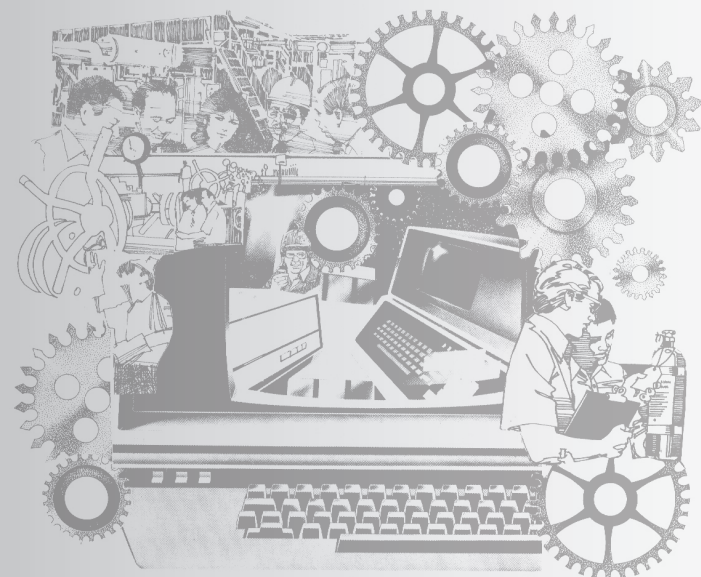
Takav način obučavanja može realizovati ne samo učitelj, već i mašina za obučavanje, programirana u skladu sa šemom na sl. 13.4. Predočavanje niza obučavanja učeniku tada se

ostvaruje projektovanjem kadra, koji sadrži potrebnu informaciju (tekst, crtež, fotografija, video snimak itd.) na ekranu projekcionog aparata ili televizijskog prijemnika, odnosno simulatora. Ispit se ostvaruje jednom od dve metode: predočavanjem alternativnih odgovora, ili samoproveravanjem. Prilikom ispitivanja po prvoj metodi, svako pitanje je proučeno skupom mogućih odgovora, među kojima je jedan pravilan odgovor; učenik, pritiskanjem odgovarajućeg dugmeta, pokazuje koji odgovor smatra pravilnim. Pri drugoj metodi provere znanja učenika, on svoj odgovor upisuje, a posle toga, pritiskanjem dugmeta izaziva kadar s pravilnim odgovorom i poredi te odgovore. Zavisno od rezultata ispita, bira se nastavak obučavanja, saglasno šemi na sl. 13.4.

Za obučavanje *spretnosti*, koriste se *aparati za treniranje* različite vrste, koji oponašaju uslove zadataka da bi učenik postao spreman za rešavanje tih zadataka. Pomoću aparata za treniranje izvodi se obučavanje u spretnosti upravljanja avionom ili brodom, spretnosti podešavanja elektronskih šema itd.

Najuspešnije obučavanje u spretnosti postiže se onda, kada se karakter i brzina obučavanja odabiraju u skladu sa individualnim osobinama učenika: sa njegovim sposobnostima i temperamentom, stečenim iskustvom u rešavanju sličnih zadataka. Individualni prilaz učeniku može se realizovati pomoću *samopodešavajućih* mašina za obučavanje. Jedan od principa, na kojima se zasniva konstruisanje samopodešavajućih mašina za obučavanje, sastoji se u automatskom merenju uvođenja zadataka u niz obučavanja, i brzine njihovog predočavanja, u zavisnosti od karaktera i učestanosti grešaka koje napravi učenik.

Tako na primer, pri obučavanju za tzv. slepo kucanje, reči ili znaci, koji se zadaju učeniku u procesu treniranja, mogu se menjati tako, da se češće sreću znaci, u kojima taj učenik greši. Osim toga, brzina „diktiranja“ može se automatski smanjiti, ako broj napravljenih grešaka za određeno vreme prelazi zadatu normu. Na taj način se u sistemu »učenik — mašina za obučavanje« (u ovom slučaju softver) uspostavlja dinamička ravnoteža, koju karakteriše određeni odnos između tačnosti i brzine rešavanja zadataka, a za koje se proces obučavanja odvija optimalno.



14. VELIKI SISTEMI

Pojam »veliki sistem« je nastao kao izraz *sistemskog prilaza* postavljanju i rešavanju zadataka upravljanja, karakterističnog za kibernetiku. Taj pojam se uvodi ne s ciljem klasifikacije sistema (podele na »velike« i »male«), već s ciljem izdvajanja *metode proučavanja* ponašanja upravljanih sistema, proučavanja koje uzima u obzir svu njima svojstvenu složenost. Izučavanje velikih sistema predodređuje korišćenje, radi toga, samo takvih metoda koje ne ignorišu postojanje tesne uzajamne veze među velikim brojem faktora, koji određuju ponašanje razmatranog sistema. Karakteristična osobina sistemskog prilaza je i uzimanje u obzir veće ili manje neodređenosti ponašanja sistema u celini, kao i njegovih sastavnih delova. Ta neodređenost predstavlja rezultat dejstva slučajnih poremećaja i učešća ljudi u sistemu, pošto se slučajni faktori takve vrste ne mogu kompenzovati idealno upravljačkim uređajem.

Pojam »veliki sistem«, za razliku od uobičajenih lokalnih shvatanja, zasniva se na tome, da se bilo koji sistem razmatra u svojoj uzajamnoj vezi s drugim sistemima, kao deo nekog većeg sistema.

Na taj način, *predmet izučavanja teorije velikih sistema je upravljani sistem posmatran kao skup uzajamno povezanih podsistema, objedinjenih opštim ciljem funkcionisanja*. Karakteristične osobine takvog sistema su postojanje odvojenih delova, učešće ljudi, mašina i prirodne sredine u sistemu, postojanje materijalnih, energetskih i informacionih veza među delovima sistema, a i spoljnjih veza razmatranog sistema s drugim sistemima.

Sistemske prilaz je neophodan za naučno proučavanje zadataka ekonomije, biologije i tehnike. Nemoguće je izvesti argumentovan zaključak o ponašanju takvog sistema kao što je mravinjak, ne posmatrajući ga kao veliki sistem. Da bi se izgradila svrsishodna šema upravljanja energetskim sistemom, neophodno je posmatrati kao veliki sistem skup: prirodni izvori energije (reke, nalazišta hemijskog i nuklearnog »goriva«, sunčeva energija, energija vetra), elektrane, podstanice, linije prenosa i potrošači energije.

Zadaci upravljanja velikim sistemom

Zadaci upravljanja, koji se pojavljuju u velikim sistemima, veoma su raznovrsni; možemo ih uslovno podeliti na dve klase: *operativne i funkcionalne*. U operativne spadaju zadaci povezani s izborom strukture veza među delovima sistema, planiranje taktike i strategije ponašanja sistema u celini i njegovih delova — podсистема, zadaci analize ponašanja sistema i ocene rezultata njegovog ponašanja. U funkcionalne uglavnom spadaju zadaci realizacije planova i strategija, izrađenih pri rešavanju operativnih zadataka, u okolnosti neizbežnog pojavljivanja situacija, koje nisu bile predviđene prilikom planiranja.

Ako se, na primer, govori o upravljanju proizvodnim preduzećem, tada se *operativni zadaci* sastoje u sledećem:

1. *Upravljanje zalihama* — skupljanje i održavanje rezervi (ljudskih, materijalnih i finansijskih).
2. *Ostvarivanje složenih kompleksa operacija*. Izgradnja ili rekonstrukcija preduzeća, stvaranje i osvajanje serije novih delova.
3. *Održavanje radne sposobnosti sistema* — staranje oko opreme i zgrade, popravke, obezbeđivanje odmora i zdravstvena služba za osoblje.
4. *Iskorišćavanje izvora* — manevrisanje ljudskim, materijalnim i finansijskim izvorima, koji učestvuju u radu sistema.
5. *Izbor maršruta* — upravljanje kretanjem tokova materijala, polufabrikata, delova; raspodela tokova energije; upravljanje kretanjem informacije.
6. *Razvoj preduzeća* — prelaz na sve efikasnije metode, oblasti i merila rada.
7. *Izbor rešenja pri takmičenjima* — takmičenje (nadmetanje, konkurencija) organizacija, borba s prirodom.
8. *Plasiranje proizvodnje* — proučavanje potražnje, formiranje proizvodne nomenklature, organizacija reklama, regulisanje cena.
9. *Organizacija rada* — pripremanje kadrova, usavršavanje sistema isplate rada, poboljšanje uslova rada.

Funkcionalni zadaci u sistemu upravljanja proizvodnjom sastoje se u obezbeđivanju zahtevanog toka tehnoloških operacija, ispunjenju zadatog kontinuiteta tehnoloških operacija, usklađivanju režima rada pojedinih proizvodnih sektora.

Prirodni upravljački sistemi obezbeđuju savršenstvo upravljanja prirodnim velikim sistemima, u koje se, nesumnjivo, mogu ubrajati živi organizmi i njihove zajednice (jato riba, stado životinja, jato ptica, roj pčela, mravinjaci).

Živi organizmi rešavaju takve operativne zadatke, kao što su izgradnja skloništa (gnezda, jazbina), vaspitanje potomstva, stvaranje sezonskih zaliha hrane i borba s neprijateljima. Funkcionalni zadaci se sastoje u obezbeđivanju životne aktivnosti organizma ishranom, disanjem, odmaranjem i staranjem o potomstvu.

Za rešavanje zadataka upravljanja velikim sistemima može da služi i služi čitav arsenal metoda upravljanja, uključujući regulisanje, adaptaciju, formiranje i ostvarenje planova i programa, obučavanje, igre i sl. Ali u velikim sistemima svi ti zadaci se pojavljuju u mnogo

složenijem vidu, nego u sistemima, razmatranim u prethodnim poglavljima, usled izvanredne složenosti samog objekta. Zaista, pošto veliki sistem, po definiciji, treba posmatrati kao skup velikog broja uzajamno povezanih podsistema, broj veličina koje karakterišu njegovo stanje, je veoma veliki. Ako se, osim toga, uzmu u obzir neizbežni elementi neodređenosti ponašanja samog sistema i slučajna dejstva na razmatrani sistem stvorena od strane drugih, s njim povezanih sistema, tada neminovno dolazimo do zaključka, da model velikog sistema predstavlja dinamički sistem veoma visoke dimenzionalnosti sa slučajno promenljivom strukturom i parametrima.

Kriterijum efikasnosti

Kao što je bilo pokazano ranije, radi efikasnog upravljanja ma kojim sistemom, neophodno je raspolagati nekim kriterijumom, koji karakteriše stepen usklađenosti ponašanja sistema sa zadatkom upravljanja.

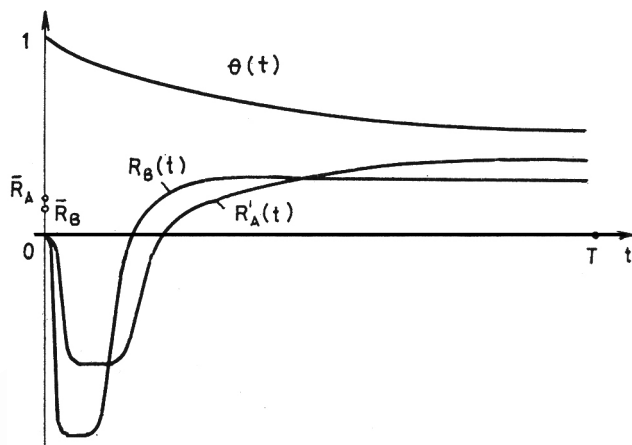
Za razliku od tradicionalnog postavljanja zadataka upravljanja, u teoriji velikih sistema kriterijumi, koji određuju efikasnost rešenja zadataka upravljanja, ne mogu se smatrati zadatim. Za svaki zadatak upravljanja u skladu sa svakim konkretnim sistemom treba da se odredi kriterijum I , čija brojna vrednost jednoznačno karakteriše efikasnost upravljanja. Zadatak formulisanja kriterijuma I u skladu s velikim sistemom postaje složeniji, usled okolnosti da je svaki sistem podsistem nekog drugog sistema višeg ranga, u koji on ulazi kao sastavni deo. Ali, zadatak upravljanja delom sistema određuje se zadacima upravljanja sistemom kome pripada. Zato se, teorijski, ni za jedan konkretan sistem ne može tačno i strogo formulisati zadatak upravljanja, ne podižući se pri tome na najviši nivo, koji omogućuje, da se uvedu u razmatranje socijalno-ekonomski faktori globalnih razmera.

Razume se, takav prilaz u savršenom vidu doveo bi do neophodnosti odustajanja uopšte od praktičnog rešavanja takvih zadataka, zbog njihove izuzetne složenosti. Uzimajući u obzir, međutim, da samo prvi koraci podizanja nivoa razmatranja bitno utiču na karakter zadatka upravljanja i kriterijum efikasnosti, moguće je pronaći razumne nivoe razmatranja za konkretne zadatke, zamenjujući pri tome uticaj svih viših nivoa popravkama, zasnovanim na grubim ocenama.

Upravljanje sistemima bilo bi veoma primitivno i nedalekovidno, ako bi kriterijum efikasnosti uzimao u obzir samo ocenu trenutnog stanja sistema. Očigledno je nesvrhsishodno usvajati takvu varijantu rešenja bilo kog zadatka, koja obećava dobit kroz nekoliko stoleća, a dotle zahteva samo rashode. Ali, istovremeno i kratkovida strategija, koja ne dopušta gubitak u sadašnjosti radi bitnih dobiti u bližoj budućnosti, nerazumna je. Jasno je, da izbor kriterijuma I efikasnosti upravljanja treba da se zasniva na svrshishodnom kompromisu između veličina dobitaka (ili gubitaka) i rokova njihovog sticanja. Pri tome će, razume se, usvojena rešenja zavisiti ne samo od toga, koji faktori i na kakav način ulaze u izraz za kriterijum I , već i od toga, kolika će im se težina davati u zavisnosti od roka sticanja odgovarajućih dobitaka i gubitaka.

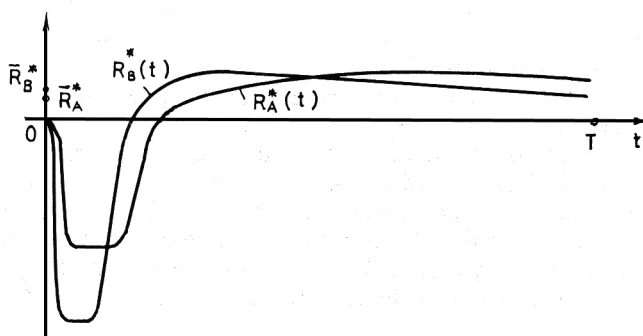
Jedna od mogućih formi proračuna raspodele dobitaka i gubitaka tokom vremena je uvođenje, u tom cilju, funkcije $\Theta(t)$, izabrane u skladu s karakterom zadatka. To treba

da bude monotonno opadajuća funkcija (da bi davala prednost kraćim rokovima sticanja dobitka), koja treba da ulazi kao množitelj u izraz za kriterijum I . Neka je, na primer, potrebno doneti rešenje o izboru jedne od dveju varijanti A i B izgradnje novog preduzeća s različitim očekivanim dijagramima $R_A(t)$ i $R_B(t)$ promena dobiti vremenom (sl. 14.1), zadatim na intervalu $(0-T)$.



Sl. 14.1. Dijagrami promena dobiti i funkcije gubitka

Ovde negativne vrednosti R označavaju troškove. Kriterijum za izbor je srednja dobit tokom vremena T . Smatraćemo, da nam je poznat karakter funkcije $\Theta(t)$, pokazane na sl. 14.1. Množeći svaku ordinatu $R(t)$ sa odgovarajućom ordinatom $\Theta(t)$, dobijamo dijagrame *svedenih vrednosti dobiti* $R_A^*(t)$ i $R_B^*(t)$, pokazane na sl. 14.2. Iz tih dijagrama je lako naći srednje vrednosti $\bar{R}_A$ i $\bar{R}_B$ dobiti za vreme T , koje mogu da služe kao osnova za usvajanje rešenja. U datom primeru prednost ima varijanta B , pošto je $\bar{R}_A^* < \bar{R}_B^*$, bez obzira što je srednja dobit $\bar{R}_A > \bar{R}_B$.



Sl. 14.2. Dijagrami svedenih vrednosti dobiti

Dobijeni rezultat je posledica uzimanja u obzir okolnosti da se varijantom A veći deo dobiti ostvaruje tokom dužeg vremena.

Na tom primeru se vidi koliko je važno uzimati u obzir rokove dobijanja dobiti prilikom donošenja rešenja.

Struktura sistema upravljanja

Jedan od važnih i složenih zadataka upravljanja velikim sistemima je zadatak određivanja racionalne strukture sistema upravljanja. Pošto se govori o upravljanju složenim skupom uzajamno povezanih objekata, može izgledati na prvi pogled, da je svrsishodno izgraditi sistem upravljanja na principu *centralizovanog upravljanja*. U sistemu centralizovanog upravljanja celokupna informacija o stanju svakog objekta upravljanja i o spoljnim dejstvima na sistem i njegove pojedine delove dolazi u centralno mesto upravljanja. U tom centralnom mestu, na osnovu posedovane informacije o stanju sistema i zadacima upravljanja, proizvode se upravljačka dejstva za svaki sastavni objekt sistema. Neupućenom čoveku takva struktura sistema upravljanja može čak izgledati idealna, pošto je u tom slučaju na jednom mestu usredsređena celokupna informacija o sistemu, zahvaljujući čemu postoji principijelna mogućnost najtačnijeg izračunavanja vrednosti kriterijuma efikasnosti i donošenja rešenja, koje obezbeđuje optimalna upravljačka dejstva. Ali, u stvarnosti takvo gledište se pokazuje veoma pogrešnim. Pre svega, za iole složeni skup objekata centralizovani sistem je praktično neostvarljiv. I zaista, radi efikasnog upravljanja čak i jednim objektom, neophodno je dobiti i preraditi dovoljno veliku količinu informacija, a za sisteme koji sadrže veliki broj objekata, količina informacija u skladu s tim raste. Zato bi u centralnom mestu upravljanja bilo potrebno skupljati ogromnu količinu raznovrsnih informacija i obezbediti njihovu preradu. Pri tome je jedan od glavnih zadataka prerade dobijenih informacija pronalaženje optimalnog režima rada sistema. Kao što je poznato, pronalaženje najpovoljnijeg režima rada sistema svodi se na zadatak određivanja ekstremuma funkcije, koja karakteriše vrednost kriterijuma I efikasnosti upravljanja. Teškoća nalaženja ekstremuma veoma brzo raste pri povećanju broja argumenata funkcije, tj. dimenzionalnosti n prostora stanja optimiziranih sistema. U centralizovanom sistemu upravljanja zadatak pronalaženja najpovoljnijeg režima mora da se rešava u prostoru veoma velike dimenzionalnosti (n je reda stotina ili hiljada), što dovodi do praktično veoma zahtevnih izračunavanja, tj. velikog angažovanja računarskih resursa. Osim toga, čak i ako bi bilo moguće naći najpovoljniji režim za neke složene sisteme, tada bi za to bilo potrebno suviše veliko vreme, te bi upravljačka dejstva stizala sa prevelikim kašnjenjem. Na taj način, kapacitete sredstava prerade informacija ograničava mogućnost efikasnog centralizovanog upravljanja složenim skupom objekata.

Nedostaci centralizovanog sistema upravljanja postaju posebno opasni, ako su njegovi elementi ljudi. Razume se, čovek je, kao i svaki kibernetički sistem, sposoban efikasno da preradi samo konačnu, dosta malu količinu informacija. Međutim, psihologija čoveka je takva, da mu rešenje koje je usvojio *izgleda* potpuno argumentovano, doneto posle analize svih faktora, i najbolje moguće, mada je ono stvarno daleko od optimalnosti, pošto u principu on ne može adekvatno uzeti u obzir sve neophodne informacije. Stoga, slična

rešenja, potpuno dopuštena u običnoj svakodnevnoj praksi, mogu imati teške posledice, ako se prošire na važne i široke oblasti delatnosti mnogih ljudi.

Takođe je potrebno primetiti, da sistem s centralizovanim upravljanjem odlikuje velika *krutost* strukture, odsustvo elastičnosti usled toga, što se njegovo prilagođavanje promenama, kako slučajnim (kolebanja) tako i onim koje izražavaju evoluciju samog sistema i okolne sredine, odvija ne u pojedinim delovima sistema, već u centralnom mestu upravljanja. Centralizovani sistem omogućuje da se za duže vreme ostvari stabilizacija sistema, suzbijajući kako kolebanja, tako i evolucione promene u pojedinim delovima sistema, ne preuređujući ga. Ali, u krajnjem rezultatu to može biti kobno po sistem, pošto protivrečnosti između nepromenljive strukture sistema i promena, povezanih s evolucijom, dobijaju globalne razmere i zahtevaju takvu radikalnu i oštru reorganizaciju, kakva je već nemoguća u okviru date strukture i dovodi do njenog razaranja (tj. prelaza na kvalitetno novu strukturu).

Centralizovano upravljanje smanjuje *pouzdanost* funkcionisanja sistema. Greške u radu centralnog mesta upravljanja ne mogu se ničim korigovati i oštro utiču na stanje čitavog sistema.

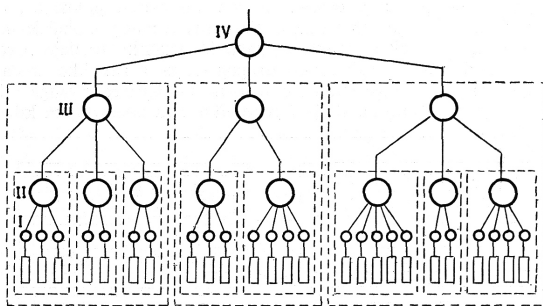
Na taj način se zaključuje, da je sistem s centralizovanim upravljanjem u nepovoljnijem položaju u odnosu na druge sisteme. Zaista, na primer, u biološkim sistemima slična centralizacija ne postoji. Zamislite šta bi bilo, kada bi mozak morao da upravlja svakim činom razmene materije u svakoj ćeliji organizma? Nemogućnost toga je očigledna. Samo zahvaljujući tome, što životnu aktivnost svake ćelije gotovo potpuno reguliše autonomni upravljački sistem — nukleinska kiselina jezgra — organizam je sposoban da tako uspešno funkcioniše, da se prefinjeno i plastično prilagođava promenama, kako spoljašnjim tako i onim koje su povezane s razvojem samog organizma.

U istoriji čovečanstva centralizovano upravljanje je vodilo društvenom i ekonomskom zastoju, i ranije ili kasnije, dovodilo je sličnu društvenu strukturu do propasti. Klasičan primer takve centralizacije bila je država kaluđera u Južnoj Americi. U toj državi je čitav društveni, privredni i privatni život bio najstrožije propisan i nadziran. Svaka delatnost, na primer, rokovi i metode poljoprivrednih radova, gde, šta i u kojoj količini proizvoditi itd. preduzimala se samo na zapovest vrhovnog upravljača — *sapa-inka*, podležući bezuslovnom izvršenju. Kao rezultat toga ogromnu imperiju, koja je imala armiju od dve stotine hiljada ratnika, bez teškoća je pokorila četa od 168 ljudi španskog konkvistadora Pizara.

Pokazani nedostaci centralizovane strukture mogu se u znatnoj meri savladati pomoću *hijerarhijske strukture* sistema upravljanja.

Karakteristična osobina hijerarhijske strukture je uzastopno raščlanjivanje (dekompozicija) sistema na delove (podsisteme), između kojih se uspostavljaju odnosi koordinacije. Tada upravljački uređaj višeg ranga upravlja velikim jedinicama sistema, od kojih svaka ima svoj upravljački uređaj. Svaka takva jedinica se, sa svoje strane, raščlanjava na manje, koje se takođe opremaju odgovarajućim upravljačkim uređajima itd., do elementarnih jedinica sistema, čije je dalje raščlanjivanje nesvrishodno. Primer hijerarhijske strukture upravljanja je pokazan na sl. 14.3. Raščlanjivanje sistema na potčinjene delove izvodi se tada na taj način, da svaki deo sadrži objekte najtešnje povezane jedne s drugim. Ili, drugim rečima, raščlanjivanje se izvodi po »slabim« vezama.

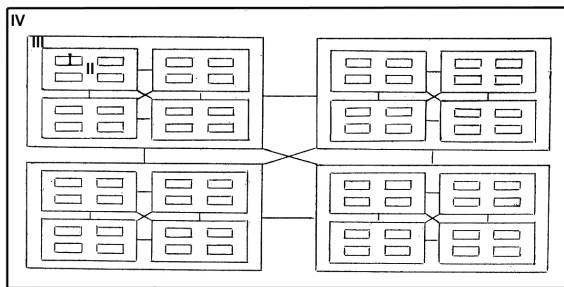
U sistemima sa hijerarhijskom strukturom, upravljački uređaj nižeg ranga — I treba da rešava relativno proste lokalne zadatke upravljanja, koji stoga odgovaraju upravljačkim uređajima s ograničenim kapacitetom prerade informacija. Pri tome, u dužnost upravljačkih uređaja sledećeg ranga — II ostaju samo oni zadaci, koje je neophodno rešavati u cilju međusobnog usklađivanja rada elementarnih objekata i koji se mogu rešavati na osnovu manje detaljne informacije o stanju objekata. To se odnosi i na upravljačke uređaje višeg ranga u sve opštijem i sistematizovanijem obliku, tako da upravljački uređaj prvog ranga dobija najdetaljniju i najkonkretniju informaciju o stanju objekata, a prelaskom na nivoe sve višeg ranga, ta informacija se uopštava u skladu s karakterom zadataka, koje rešavaju ti uređaji. Naredbe upravljanja u sistemima sa hijerarhijskom strukturom izdaje upravljački uređaj najvišeg ranga u najopštijoj formi i sve se više konkretizuju i detaljizuju pri svom prenošenju na upravljačke uređaje nižih rangova. Sistemi s hijerarhijskom strukturom formirali su se u toku prirodnog razvoja bioloških, tehničkih i ekonomskih sistema. Možemo ih danas primetiti u svim složenim sistemima, kao što je upravljanje granama privrede i pojedinim preduzećima, u transportnim i energetskim sistemima, vojnim jedinicama i dr.



Sl. 14.3. Hijerarhijska struktura upravljanja

Ne treba smatrati da hijerarhijska struktura, slično centralizovanoj, obavezno predstavlja izdvajanje upravljačkih uređaja višeg ranga u vidu posebnih delova sistema, na »višem položaju«, od upravljačkih uređaja nižeg ranga. Naprotiv, u mnogim slučajevima bi bila najbolja takva struktura, kada bi nekoliko uređaja prvog ranga zajedno obrazovali uređaj drugog ranga, a nekoliko uređaja drugog ranga, sa svoje strane, obrazovali uređaj trećeg ranga itd., kao što je pokazano na sl. 14.4 (uporedite i sa hijerarhijom kibernetičkih sistema). Pri tome svaki uređaj samostalno rešava svoje lokalne zadatke, a zajedno s drugim uređajima rešava samo opštije zadatke, svojstvene upravljačkom uređaju višeg ranga. Još šire »ujedinjavanje« takvih uređaja rešava još uži krug još opštijih zadataka itd.

Pri tome postaje moguće donositi rešenja putem glasanja, što, kako je Džon fon Nojman prvi pokazao, bitno povećava pouzdanost funkcionisanja sistema.



Sl. 14.4. Princip izgradnje hijerarhijskog sistema ujedinjavanjem sistema nižeg ranga u sisteme višeg ranga

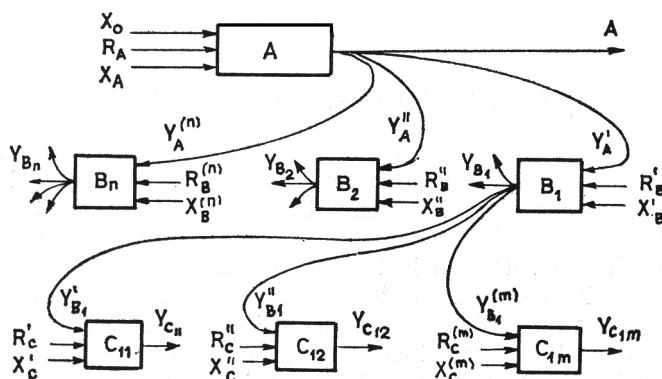
Radovi sovjetskih naučnika Kolmogorova i Ofmana pokazali su da sistem, u kome su svi elementi potpuno »ravnopravni« a pri tome je svaki element povezan samo s malim brojem drugih, može pri dovoljnom broju elemenata da rešava ma kako složene zadatke koje rešavaju drugi uređaji (tj., drugim rečima, takav sistem predstavlja takozvani univerzalni automat).

Stvaranje hijerarhijske strukture

Često je rad velikih sistema, kao što je, na primer, fabrika, teško ili sasvim nemoguće matematički opisati. Pa ipak, neophodno je upravljati sistemom na takav način, da rezultati njegovog rada zadovoljavaju neki kriterijum. Upravljanje, zasnovano na primeni strogih matematičkih metoda, ponekad je toliko složeno, da ga je nemoguće iskoristiti, čak i ako se raspolaze savremenim moćnim računarima. U takvim okolnostima se javlja ideja o podražavanju dejstva čoveka-rukovodioca, koji donosi rešenja često intuitivno. Pokazuje se, da većina ljudi podsvesno koristi jedne iste metode, koje im omogućuju da rešavaju nastale složene zadatke. Metode te vrste nazivaju se *heurističkim*.

Neke heurističke metode su pogodne samo za rešavanje jednog konkretnog zadatka, ali postoje metode pogodne za rešavanje velike klase zadataka.

Jedna od opštih heurističkih metoda, koja se može koristiti za upravljanje velikim sistemima, sastoji se u sledećem. Zamislimo upravljački uređaj velikog sistema u vidu nekog bloka A (sl. 14.5), na čiji ulaz dolaze podaci o cilju upravljanja X_0 , o tekućem stanju sistema X_A , a i o postavljenim ograničenjima na upravljanje R_A .



Sl. 14.5. Šema upravljanja velikim sistemom

Rezultat rada bloka A je izlazno dejstvo, ili izlazni operator Y_A , koji može imati karakter kompleksa naredbi, instrukcija, odluka i sl. Ako je operator takve vrste nedovoljan za efikasno upravljanje sistemom, tada ga upravljački uređaj A može razbiti određenim načinom na skup prostijih operatora $Y_A^{(1)}, Y_A^{(2)}, \dots, Y_A^{(n)}$ i predati ih blokovima $B_1, B_2, \dots, B_n$, koji se nalaze na nižem nivou, radi dešifrovanja i detaljnije razrade. Sada skup operatora, predatih

sledećem redu, postaje skup ciljeva za blokove $B_1, B_2, \dots, B_n$. Istovremeno je tim blokovima, radi ostvarenja ciljeva, potrebna detaljnija informacija $X_B, X_B'', \dots, X_B^{(n)}$ o stanju sistema i njima mogu biti postavljena ograničenja $R_B, R_B'', R_B^{(n)}$. Rezultat rada blokova drugog reda su operatori $Y_{B1}, Y_{B2}, \dots, Y_{Bn}$, koji se mogu, ako je neophodno, rastaviti na komponente i poslati u vidu lokalnog cilja u blokove sledećeg reda. Takva izgradnja hijerarhije može da se produži dotle, dok upravljačka dejstva, koja dolaze u sistem upravljanja, ne budu dovoljna za traženi rad sistema, za približavanje njegovog tekućeg stanja X_A cilju X_0 . Takođe je neophodno primetiti, da za tako organizovani upravljački sistem nije potrebno slati svu ulaznu informaciju koja dolazi u blokove najnižeg reda, a informaciju koja dolazi od svakog bloka bilo kog reda, u sve blokove koji leže u višim redovima. Na primer, podaci o tekućem stanju sistema X_B treba da sadrže informaciju, dobijenu od blokova neposredno povezanih s blokom B_1 , tj. od $C_{11}, C_{12}, \dots$ koji leže niže, i onu informaciju, koja nedostaje za obrazovanje sastavnog operatora Y_{B1} . Na taj način, u svaki viši red informacija o stanju sistema dospeva kao da je filtrirana kroz redove koji leže niže. To bitno smanjuje opseg i vreme obrade informacije prispele u blok, i omogućuje da se nađe rešenje ili upravljačka dejstva u prihvatljivom roku i sa dopuštenim utroškom sredstava. Istina, rešenja dobijena takvim načinom upravljanja, nisu uvek optimalna ili čak ni blizu optimalnih; ali u mnogim situacijama, ograničenja vremena i sredstava su toliko oštra, da se mora odustati od optimalnosti u korist nekog prihvatljivog rešenja.

Statističko modeliranje – metoda Monte-Karlo

Proučavanje osobina velikih sistema, njihovih reakcija na promene njihovih karakteristika sredine ili na promene elemenata, suštinski otežavaju njihova izvanredna složenost i neodređenost ponašanja. Radi ispitivanja konkretnog sistema te vrste, moguće je, razume se, pribeci eksperimentalnim metodama istraživanja posmatranjem njegovog ponašanja (pasivni eksperiment) ili ispitivanjem reakcije sistema na raznovrsne poremećaje, specijalno stvorene radi njegovog proučavanja (aktivni eksperiment). Međutim, izvođenje eksperimentalnih istraživanja na realnim sistemima je, po pravilu, veoma skupo i ograničeno dozvoljenim režimima, koji isključuju mogućnost ispoljavanja ponašanja u neobičnim, na primer štetnim, uslovima. Osim toga, takve metode su, očigledno, neprimenljive za prvi put stvorene sisteme, čije osobine treba ispitati, da bi se ocenilo kako su realizovani.

U vezi s tim, za proučavanje velikih sistema, naveliko se primenjuje *metoda statističkog modeliranja*.

Napomena: Metoda se naziva i metoda statističkih ispitivanja, ili metoda Monte-Karlo (po imenu grada u kneževini Monako, gde se nalazi najveća svetska kockarnica sa čuvenim ruletom).

Metoda statističkog modeliranja zasniva se na podražavanju slučajnih događaja u modelu ispitivanog sistema. Po pravilu, statističko modeliranje se ostvaruje pomoću računara. Tada se u računar uvode podaci o poznatim osobinama modeliranog sistema i spoljnje sredine. Slučajni faktori se modeliraju pomoću izvora slučajnih veličina sa zadatim statističkim svojstvima. Takav izvor može da bude tabela slučajnih brojeva ili pseudoslučajnih brojeva, koje proizvodi računarski program na osnovu specijalnog algoritma. Važna osobina metode

statističkog modeliranja je, da svako *pojedino* rešenje zadatka samo po sebi ne karakteriše sistem, jer je slučajno. Tražene zakonitosti se utvrđuju samo na osnovu dobijanja *velikog broja* pojedinih rešenja putem određivanja njihove *sredine*.

Metodom statističkog modeliranja mogu se uspešno rešiti takvi zadaci, kao što su, na primer, određivanje optimalnih intervala između popravki opreme, obezbeđujući dopušteno rasipanje kvaliteta proizvodnje, izrađene tom opremom; predviđanje veličina redova, koji se obrazuju u sistemima masovnog opsluživanja (sistemima veza, saobraćajnim sistemima, trgovačkim organizacijama); optimalna raspodela tolerancija parametara međuproizvoda u lančanoj proizvodnji.

Radi ilustracije, razmotrimo primenu metode statističkog modeliranja prilikom rešavanja relativno jednostavnog zadatka.

Pretpostavimo, da od velikog broja delova, kojima raspolaže radnik u nekoj proizvodnji, polovina delova ima dimenzije s pozitivnim odstupanjima od nominalnih, a polovina — s negativnim odstupanjima dimenzija od nominalnih. Prilikom sklapanja, u svaki proizvod se ugrađuju tri dela. Pravilno funkcionisanje proizvoda biće narušeno samo u tom slučaju, ako se u proizvodu nađu istovremeno tri dela s pozitivnim odstupanjima. Potrebno je odrediti, kolika je verovatnoća dobijanja proizvoda, koji pravilno funkcioniše, ako radnik uzima delove iz ambalaže nasumice. Zadatak se može rešiti analitičkom metodom. Ako označimo sa p verovatnoću, da nasumice uzeti deo ima pozitivno odstupanje od nominalne vrednosti, tada se verovatnoća sklapanja upotrebljivog proizvoda p_{pp} može lako izračunati:

$$p_{pp} = 1 - p^3 = 1 - \left(\frac{1}{2}\right)^3 = 0,875.$$

Isti zadatak se može rešiti metodom statističkog modeliranja bacanjem tri kovana novca: ako prilikom bacanja novca ispadne grb, smatraćemo da je »plus«, a ako ispadne pismo — »minus«. Ako se pri istovremenom bacanju triju moneta pojave tri grba, smatraćemo, da sklopljeni mehanizam neće pravilno funkcionisati; pri svakoj drugoj kombinaciji (dva grba i jedno pismo, ili jedan grb i dva pisma, ili tri pisma), sklapanje će biti pravilno. Ako se ponove opiti s bacanjem novca više puta i oceni verovatnoća p_{pp} , tada se možemo uveriti da će ta verovatnoća biti blizu 0,875.

Ostvarujući metodu statističkog modeliranja pomoću računara, moguće je, u prihvatljivom vremenu, dobiti rešenja veoma složenih zadataka, uključujući i zadatke istog tipa kao dati primer, ali pri većem broju delova koji se sklapaju u složeni sklop. Metod Monte-Karlo je veoma efikasan i za dobijanje numeričkih rešenja mnogih zadataka van oblasti verovatnoće, na primer za izračunavanje višestrukih integrala, rešavanje složenih sistema diferencijalnih jednačina, traženje optimalne varijante pri planiranju i za mnoge druge.

Masovno opsluživanje

Prilikom rešavanja zadataka upravljanja velikim sistemima, srećemo slučajeve, kada je neophodno uzeti u obzir propusnu moć onih tehničkih uređaja, za koje treba obezbediti optimalno upravljanje. Za rešavanje tih zadataka obično se primenjuje jedna od oblasti

teorije verovatnoće — *teorija masovnog opsluživanja*. Tako, na primer, u složenim proizvodnim kompleksima, svaka vrsta opreme (reaktori, mašine alatke, mašine za valjanje metala) treba u određenom periodu da ostvari zadatu veličinu rada, tj. da posluži, obradi i preradi određenu količinu polaznog materijala. Analogno, svaka od jedinica preduzeća (odeljenja, sektori, pogoni) može se posmatrati kao odgovarajuća uslužna jedinica, koja treba u toku određenog perioda da opsluži (proizvede, obradi, izradi) niz radnih predmeta (niz trebovanja, delova, sirovina, sklopova itd.). Očigledno je, da se i privredne jedinice na višem nivou — preduzeće, trgovačka kuća, grana industrije — mogu posmatrati kao sistemi masovnog opsluživanja. Pri planiranju rada takvih sistema, uzima se u obzir njihova propusna moć; broj trebovanja koji se mogu uslužiti u jedinici vremena; vreme čekanja trebovanja u redu i dr. Razume se, da od tih karakteristika u znatnoj meri zavise tehničko-ekonomski pokazatelji rada pojedinih agregata i jedinica. Zaista, od tačnosti proračuna broja zahteva, koji se mogu poslužiti, zavisi pravilnost pripreme i izvršenje plana; vreme čekanja određuje neophodan broj bunkera i drugih uređaja za čuvanje polufabrikata za vreme njegovog čekanja u redu na posluživanje, a od toga zavisi neophodna veličina obrtnih sredstava i brzina njihovog obrta.

Razmotrimo najprostiji primer, donekle arhaičan, zadatka masovnog opsluživanja. Telegrafskom sektoru pošte pristizu »hitni« telegrami. Učestanost pritanja takvih telegrama λ jednaka je 2 telegrama na čas ($\lambda = 2$). Svaki telegram se odmah po prijemu predaje poštaru, koji ga dostavlja primaocu. Pretpostavimo, da je srednje vreme dostavljanja telegrama primaocu i vraćanja poštara u poštu $\mu = 2,5$ časa. Važne karakteristike takvog sistema masovnog opsluživanja su srednji broj poštara, koji se istovremeno nalazi na putu (a) i verovatnoća (p_k) da istovremeno bude zaposleno k poštara. Srednji broj istovremeno zauzetih poštara raste s povećanjem učestanosti pritanja telegrama i udaljenosti primaoca od pošte.

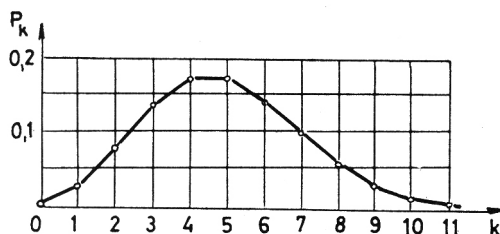
Pošto su broj telegrama, prispelih na čas, i vreme dostavljanja — nezavisne slučajne veličine, srednji broj zaposlenih poštara a biće određen učestanošću pritanja telegrama i srednjim vremenom dostavljanja kao njihov proizvod:

$$a = \lambda \cdot \mu = 5.$$

Jasno je da je slučaj istovremene zauzetosti veoma velikog broja poštara, isto kao i slučaj da su svi poštari slobodni od raznošenja telegrama, u navedenom primeru malo verovatan. Tačna vrednost verovatnoće da su istovremeno zauzeti k poštara može se odrediti na osnovu sledeće formule:

$$p_k = \frac{a^k \cdot e^{-a}}{k!},$$

To je takozvana **Poasonova raspodela**. Karakter te funkcije za $a = 5$ pokazan je na sl. 14.6. Najverovatnije je (za ceo broj a), da će istovremeno biti zauzeti a ili ($a - 1$), tj. 5 ili 4 poštara. Ta verovatnoća iznosi $p_5 = p_4 = 0,175$. Verovatnoća da svi poštari budu slobodni je $p_0 = 0,0067$.



Sl. 14.6. Raspodela verovatnoća istovremene zauzetosti k poštara

Odatle sledi, da će, u proseku, u toku 1000 časova radnog vremena istovremeno svi poštari biti slobodni samo 6,7 časova.

Većina praktičnih zadataka teorije masovnog opsluživanja je toliko složena da se njihovo rešenje ili ne može dobiti analitički, ili su dobijene formule osobito glomazne i nepogodne za proračun. U takvim slučajevima posebno je efikasna metoda Monte-Karlo, koja se realizuje na računaru.

Vežbanje

1. Nabrojte operativne i funkcionalne zadatke nastale pri upravljanju energetskeg istemom velikog preduzeća. (U energetskeg sistem ulaze: termoelektrana, podstanica, parni kotlovi, kompresori, linije prenosa vazduha, električne i toplotne energije, crpne stanice i dispečerski pult.)

Odgovor:

Operativni zadaci su: manevrisanje ljudskim, materijalnim i finansijskim izvorima; rekonstrukcija opreme sistema; održavanje opreme, remont, montažni radovi i popravke; dispečersko upravljanje (raspodela različitih oblika energije po jedinicama preduzeća, dovoz goriva i odvoz iskorišćenog materijala, razmena informacija unutar sistema i sa spoljnom sredinom); obezbeđivanje odmora i medicinska služba za osoblje; pripremanje kadrova; poboljšanje uslova rada; učešće u internom takmičenju i štetni radovi.

Funkcionalni zadaci su: održavanje zahtevanih režima rada agregata; ispunjenje zadatog redosleda uključivanja i isključivanja agregata, linija, podstanica, pumpi itd.; kao i ispunjenje dijagrama opterećenja.

2. Projektovani sistem snabdevanja električnom energijom nekoliko regiona uključuje veliki broj termoelektrana a i komplekse prenosa i raspodele energije. Pretpostavlja se da upravljački uređaji nalaze široku primenu u tom sistemu. Kakvu je strukturu upravljanja s upotrebom upravljačkih uređaja svrsishodno izabrati za dati sistem i zašto?

Odgovor:

Svrishodno je izabrati strukturu koja osim globalne upravljačkog uređaja uključuje i lokalne. Takav sistem upravljanja će imati nekoliko redova hijerarhije i omogućiti elastično upravljanje sistemom u celini, ne zalazeći u detalje upravljanja svakom pojedinom elektranom ili podstanicom.

3. U jedno od odeljenja izdavačkog preduzeća pristižu rukopisi knjiga, pri čemu srednji broj prispelih rukopisa u toku jedne godine iznosi 30. Svaki rukopis se posle prijema daje na recenziju. Srednje vreme potrebno za recenziju iznosi 2 meseca. Odredite srednji broj istovremeno zaposlenih recenzenata i verovatnoću da broj istovremeno zaposlenih recenzenata ne prelazi 9 ljudi.

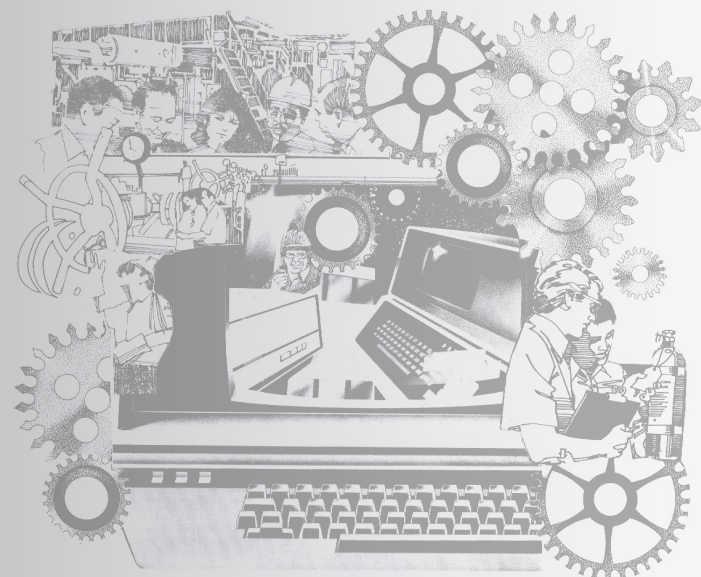
Rešenje:

Srednji broj recenzenata je $a = \lambda \cdot \mu = 5$. Vreme, u toku koga broj istovremeno zauzetih recenzenata ne prelazi 9 ljudi, sastoji se iz vremena, kada su svi recenzenti istovremeno slobodni, kada je zauzet 1 recenzent, kada su zauzeta 2 recenzenta, itd. do 9, zaključno. Stoga je tražena verovatnoća (saglasno sl. 14.6).

$$p \leq p_0 + p_1 + \dots + p_9 \approx 0 + 0,03 + 0,08 + 0,14 + \\ + 0,175 + 0,175 + 0,15 + 0,10 + 0,07 + 0,03 = 0,95$$

Odatle sledi, da s verovatnoćom 0,95 mogu 9 recenzenata da urade sve recenzije.





15. UPRAVLJANJE OPERACIJAMA

Operativni zadaci, nastali u velikim sistemima, odlikuju se specifičnim osobinama, što dovodi do neophodnosti korišćenja posebnog prilaza njihovom rešavanju. Jedna od najvažnijih osobina operativnih zadataka sastoji se u tome, što je objekt upravljanja ovde, po pravilu, kolektiv izvođača, snabdeven odgovarajućim tehničkim sredstvima i određen za postizanje utvrđenog cilja. To može biti građevinska organizacija, koja gradi industrijsko preduzeće; konstruktorski biro, koji razrađuje projekt novog automobila; marketing agencija, koja organizuje kulturnu manifestaciju ili naučni kolektiv, koji rešava određeni problem.

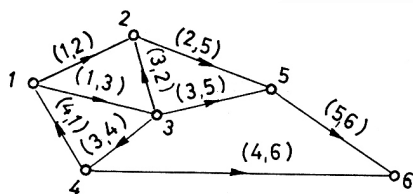
Delatnost kolektiva izvođača, usmerenu ka postizanju postavljenog cilja, nazivaćemo operacija. Operacija se uvek može raščlaniti na odvojene etape — radove, čije je izvršenje neophodno potrebno i dovoljno za njeno ostvarenje u celini. Pojedine etape — radovi, koji čine operaciju, — povezane su jedne s drugim određenim uslovima, koji ograničavaju izbor redosleda njihovog izvršenja, a i zajedničkim ljudskim, materijalnim i finansijskim izvorima. Radi efikasnog upravljanja operacijama neophodno je rešiti zadatke dva tipa: 1) izraditi optimalan plan izvođenja operacije i 2) obezbediti realizaciju operacije, dovoljno blisku optimalnom planu, u uslovima neizbežne pojave situacija, koje nisu bile predviđene prilikom sastavljanja plana.

Naučni prilaz rešavanju tih zadataka moguć je samo uz uslov, da postoji matematički model, koji dovoljno dobro odražava osobine i karakteristike objekta i koji je pogodan za proučavanje formalizovanim metodama. Jedan od najpogodnijih i najprostranjenijih modela, primenjenih u rešavanju zadataka upravljanja operacijama, je *mrežni model*.

Mrežni model

Mrežni model operacije (ili »mreža«, kako ćemo ga radi kratkoće zvati) predstavlja poseban vid grafa, zapravo — *konačni orijentisani graf bez kontura*. Radi objašnjenja termina, iskorišćenih u ovoj definiciji, neophodno je osvrnuti se još jednom na teoriju grafova.

Pod grafom se podrazumeva skup tačaka povezanih neprekidnim linijama. Ako te spojne linije, nazvane rebra grafa, imaju strelice, koje ukazuju na razlike prelaza u smeru strelice i u suprotnom smeru, tada je takav graf *orijentisan*. Tačke, povezane rebrima, nazivaju se *temena grafa*. Graf je *konačan*, ako je konačan broj njegovih temena (a stoga, i rebara). Primer konačnog orijentisanog grafa je dat na sl. 15.1.



Sl. 15.1. Konačni orijentisani graf

Svakom temenu ćemo pripisati broj — njegov broj. Rebro ćemo označavati brojevima temena (i, j), koja ono povezuje. Niz rebara, u kome se kraj svakog prethodnog rebara poklapa s početkom sledećeg, naziva se *put*. *Kontura* je put, u kome se početno teme poklapa s krajnjim. Graf, prikazan na sl. 15.1 sadrži konturu 1—3—4—1 i zato, saglasno definiciji, nije mreža.

U mrežnim modelima operacija, obično rebra prikazuju pojedine etape operacije — *radove*; temena grafa prikazuju *događaje*, koji označavaju završetak određene etape operacije. U razmatranim mrežama ističu se dva posebna događaja: *početni* (početak svih operacija) i *konačni* (završetak svih operacija). Pojam *rada* u mrežnim modelima upotrebljava se u nekoliko značenja: on može da označava sopstveni rad, koji zahteva utrošak napora, materijalnih izvora i vremena, ili *očekivanje* koje ne zahteva ulaganje napora ni izvora, ali ispunjava neko vreme, i *fiktivni rad*, koji ne zahteva utrošak ni napora, ni izvora, ni vremena, već ukazuje na nemogućnost počinjanja nekih radova pre nastupanja određenog događaja.

Razmotrimo kao primer mrežni model jednostavne operacije, koja se sastoji u instaliranju elektromotora na fundamentnu ploču. U operaciju ulaze sledeći radovi:

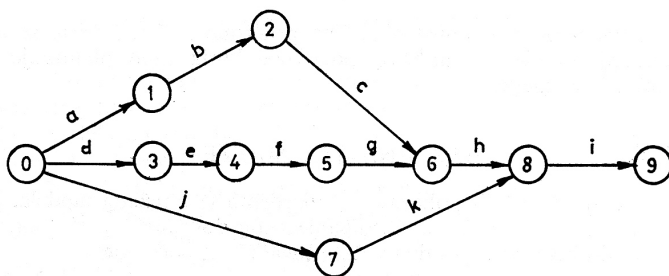
1. Uobličenje narudžbine za fundamentnu ploču.
2. Izrada fundamentne ploče.
3. Prevoz ploče na mesto postavljanja motora.
4. Priprema temelja za fundament.
5. Izgradnja oplata za fundament.
6. Betoniranje fundamenta.
7. Stvrđnjavanje betona.

8. Montaža fundamentne ploče.
9. Poručivanje i dobijanje motora iz prodavnice/stovarišta.
10. Prevoz motora na mesto ugradnje.
11. Montaža motora.

Polazeći od redosleda radova, određenih tehnologijom njihovog izvršenja, može se sastaviti mreža pokazana na sl. 15.2. Konfiguracija te mreže odražava uslove dozvoljenosti otpočinjanja izvršenja svakog rada u skladu s pravilima: rad, koji izlazi iz i -tog temena mreže može se započeti samo posle završetka svih radova, koji ulaze u i -to teme. U razmatranom primeru u teme (događaj) 6 »ulaze« dva rada: »prevoz fundamentne ploče« i »stvrđnjavanje betona« a izlazi — »montaza fundamentne ploče«. To znači da se sa montažom ploče može početi samo posle isporuke ploče i posle završetka stvrđnjavanja betona.

0. Događaji, prikazani temenima mreže, imaju sledeći smisao:

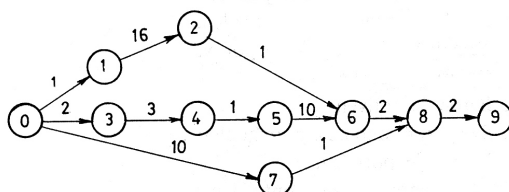
1. Donošenje rešenja o instaliranju. *Početak operacije.*
2. Fundamentna ploča je poručena.
3. Fundamentna ploča je izrađena.
4. Temelj za fundament je gotov.
5. Oplata je gotova.
6. Beton je zaliven.
7. Sve je spremno za montažu ploče.
8. Motor je poručen.
9. Sve je spremno za montažu motora.
10. Motor je instaliran. *Kraj operacije.*



Sl. 15.2. Mrežni grafikon instaliranja elektromotora. a — Poručivanje fundamentne ploče. b — Izrada fundamentne ploče. c — Prevoz fundamentne ploče. d — Priprema za fundament, e — Izgradnja oplata, f — Betoniranje. g — Stvrđnjavanje betona. h — Montaža fundamentne ploče. i — Montaža motora. j — Poručivanje i dobijanje motora. k — Prevoz motora

Radove, koji čine operaciju, karakterišu određene vrednosti vremena i izvora, neophodnih za njihovo izvršenje. Ograničićemo se samo na razmatranje vremenskih ocena mreža i zato ćemo uzeti u obzir samo utrošeno vreme $t(i,j)$ za svaki rad (i,j) , raspoređen

između događaja i i događaja j . Vreme izvršenja rada skoro nikada ne može biti predviđeno s visokom tačnošću; moguće je samo proceniti verovatnoću njegovog izvršenja u određenom roku. Ali za grube proračune mogu se koristiti izvesne srednje vrednosti očekivanog trajanja rada, kojima ćemo karakterisati radove u razmatranim mrežnim modelima. Na sl. 15.3 je dat mrežni model s vremenskim karakteristikama radova za razmatrani primer. Očekivano trajanje svakog rada, izraženo u danima, označeno je iznad svakog rebra mreže.



Sl. 15.3. Mrežni grafik sa vremenskim karakteristikama radova na instaliranju elektromotora

Mrežni modeli, konstruisani na prikazani način, koriste se za planiranje operacija i njihovo ponovno, korigovano planiranje u procesu realizacije.

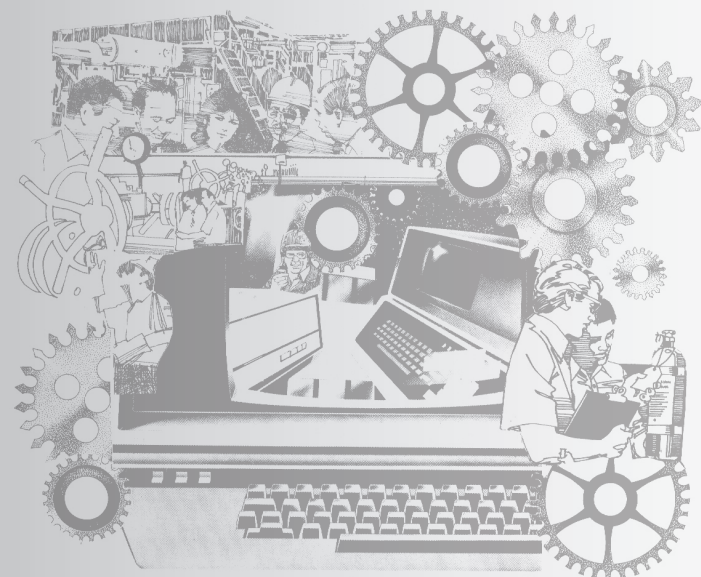
U procesu ostvarivanja operacija, po pravilu se javljaju odstupanja od plana, izazvana mnoštvom okolnosti, nerazmatranih prilikom planiranja. Trajanje izvršavanja pojedinih radova odstupa od proračunatih vrednosti; pojavljuju se kašnjenja usled bolesti izvođača, kašnjenja u isporuci materijala i opreme, lomovi mehanizama itd. Često nastaje potreba za izvršavanjem nepredviđenih radova, a neki isplanirani radovi mogu se pokazati nepotrebnim. Sve te okolnosti dovode do toga, da za efikasno izvođenje operacija nije moguće ograničiti se na izradu optimalnog plana već je neophodno kontrolisati njegovo ispunjenje i ostvariti **operativno upravljanje** njegovom realizacijom.

Operativno upravljanje se ostvaruje promenama zadataka izvođačima i preraspodelom izvora. U tom cilju, mora u toku izvršavanja operacija periodično da se *prerađuje plan*, s obzirom na izvršene radove, stvarno postojanje izvora i moguće promene propisanih rokova.

Periodično prerađivanje plana operacije izvodi se određivanjem tzv. kritičnog puta, rezervi vremena i optimalne raspodele izvora istim metodama kao i pri prvobitnom planiranju, s tom razlikom, što se očekivani datumi njihovog završavanja zamenjuju stvarnim, a očekivano postojanje izvora — stvarnim njihovim postojanjem, stavljenim rukovodiocu operacije na raspolaganje.

Tada mrežni model operacije već postaje dinamički a ne statički model, koji se menja u skladu sa dobijenim podacima o stanju upravljanog sistema. Ako prvobitni plan izvođenja operacije predstavlja određeni *program*, po kome ona treba da se ostvari, tada operativno upravljanje, zasnovano na povratnoj vezi, koja prenosi informaciju o stvarnom stanju sistema, pretvara njega u *zatvoreni sistem upravljanja*.

Sistemi mrežnog planiranja i upravljanja našli su široku primenu u raznim oblastima ljudskih delatnosti. Oni omogućuju bitno poboljšanje organizacionog upravljanja u industriji, građevinarstvu, saobraćaju itd.



PRAKTIKUM

1. DIFERENCIJALNE JEDNAČINE

U suštini, diferencijalne jednačine obuhvataju izvode (diferencijale) koji specificiraju kako se neka količina (kvanitet) menja; rešavanjem diferencijalne jednačine dobija se formula za samu količinu koja ne obuhvata izvode.

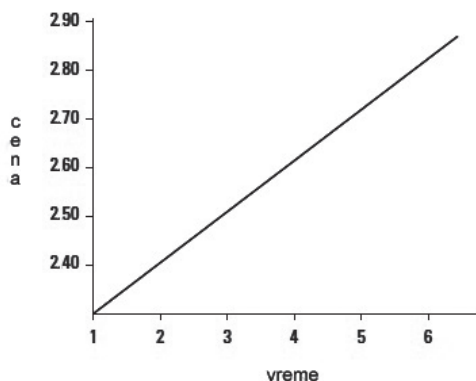
Kako su izvodi osnova za diferencijalne jednačine, ukratko ćemo se podsetiti nekih osnovnih činjenica kroz jedan primer.

Recimo da već dugo vremena kupujete jedan te isti artikal u obližnjoj prodavnici. U sledećoj tabeli dato je kretanje cene tog artikla u par prethodnih meseci.

mesec	1	2	3	4	5	6
cena	240	250	260	270	280	290

Posmatranjem podataka zaključuje se da cena artikla postojano raste, što se uočava i sa grafikona cene. Sa takvim kretanjem rasta cene kolika će cena artikla biti kroz godinu dana?

Poznato je iz srednjoškolske matematike da je nagib krive određen sa $\Delta y / \Delta x$, tj. promenom po y podeljenom sa promenom po x . Za konkretan primer koristićemo simbol Δc za promenu u ceni a Δt za promenu po vremenu. Prema tome, nagib krive sa prethodne slike je $\Delta c / \Delta t$.



Sl. 1. Mesečni rast cene artikla

Kako cena artikla raste 10 dinara svakog meseca, znamo da je nagib krive jednak:

$$\frac{\Delta y}{\Delta x} = 10 \text{ din/mesec.}$$

Nagib krive je konstantan i pokazuje njenu brzinu promene. Izvod količine takođe daje njenu brzinu promene u bilo kojoj tački, tako da izvod možemo posmatrati kao nagib u konkretnoj tački. Kako je brzina promene krive konstantna, možemo pisati:

$$\frac{dc}{dt} = \frac{\Delta c}{\Delta t} = 10 \text{ din/mesec.}$$

U ovom slučaju, dc/dt je izvod cene artikla po vremenu. (Kada vidite simbol d , znajte da se radi o izvodu.)

Shodno prethodnom, dobija se *diferencijalna jednačina*:

$$\frac{dc}{dt} = 10 \text{ din/mesec.}$$

Prethodna jednačina je diferencijalna jednačina zato što je to jednačina koja sadrži izvod, u ovom slučaju dc/dt . To je prilično jednostavna diferencijalna jednačina čije rešenje (cena artikla) je funkcija vremena i ima sledeći oblik:

$$c = 10 \cdot t + k.$$

U ovoj jednačini, c je cena (izražena u dinarima), t je vreme (izraženo u mesecima), a k je proizvoljna konstanta koja se koristi da zadovolji *početne uslove* problema. (Potrebna nam je konstanta k zato što kada uzmemo izvod od $10 \cdot t + k$, dobijamo rezultat 10, tako da ne možemo reći da li postoji konstanta koju treba dodati članu $10 \cdot t$ — to će nam saopštiti zadovoljeni početni uslovi.)

Nedostajuća karika je vrednost k , zato ćemo uključiti brojeve koje imamo za cenu i vreme da bi odredili tu vrednost. Na primer, cena artikla prvog meseca je 240 dinara, zamenimo tu vrednost u jednačini za cenu artikla:

$$240 = 10 + k$$

Rešavanjem ove jednačine dobijamo da je $k = 230$, tako da je rešenje naše diferencijalne jednačine:

$$c = 10 \cdot t + 230.$$

Primena početnih uslova

Kada je data diferencijalna jednačina u obliku $dy/dt = f(t, y)$, naš cilj je da nađemo funkciju, $y(t)$, koja je rešava (zadovoljava). Možemo početi sa integraljenjem jednačine koje daje rešenje koje uključuje konstantu a zatim da primenimo početne uslove da prilagodimo rešenje. Primena početnog uslova omogućava da izaberemo jedno rešenje od beskonačno raspoloživih koja su rezultat integraljenja (integracije).

Razmotrimo jednu jednostavnu linearnu diferencijalnu jednačinu prvog reda:

$$\frac{dy}{dt} = \alpha,$$

gde je neki regularni broj. Kako možemo da rešimo ovu jednačinu? Pre svega, treba uočiti da se ova jednačina može zapisati i na sledeći način:

$$dy = \alpha \cdot dt.$$

Sada možemo da integralimo jednačinu:

$$\int_{y_0}^y dy = \int_{t_0}^t \alpha \cdot dt.$$

Posle integracije dobijamo sledeću jednačinu:

$$y - y_0 = \alpha \cdot t - \alpha \cdot t_0.$$

Ako član y_0 prebacimo na desnu stranu, možemo izraz $y_0 - \alpha \cdot t_0$ da pretvorimo u novu konstantu c i konačno dobijamo:

$$y = \alpha \cdot t + c.$$

Možemo da zaključimo da je rešenje naše diferencijalne jednačine $y = \alpha \cdot t + c$.

Na primer, ako je $\alpha = 3$ u diferencijalnoj jednačini, naša diferencijalna jednačina je:

$$\frac{dy}{dt} = 3.$$

Rešenje te jednačine je $y = 3 \cdot t + c$.

Treba imati u vidu da član c (rezultat integracije), može biti bilo koja vrednost a posledica toga je beskonačan skup rešenja: $y = 3t + 5$, $y = 3t + 6$, ..., $y = 3t + 567382$, Kako ćemo znati koja vrednost c je ona prava? To zavisi od tzv. početnih uslova; na primer možemo da specificiramo da je za $t = 0$ vrednost od y jednaka 15. Postavljanje početnog uslova omogućava da se formuliše ceo problem — diferencijalna jednačina i početni uslov, kao što sledi:

$$\frac{dy}{dt} = 3$$

$$y_0 = 15$$

Zamenom početnog uslova, $y(0) = 15$, u rešenje $y = 3t + c$ dobija se sledeća jednačina:

$$y(t) = 3t + 15.$$

Sledeći korak je da rešimo diferencijalnu jednačinu koja sadrži funkcije od t a ne samo prost broj.

Ovaj tip diferencijalnih jednačina još uvek sadrži član dy/dt i članove koji su funkcija od t . Sledi osnovni oblik diferencijalne jednačine ovog tipa:

$$\frac{dy}{dt} = g(t),$$

gde je $g(t)$ neka funkcija od t .

Jedan primer diferencijalne jednačine ovog tipa je:

$$\frac{dy}{dt} = t^3 - 3t^2 + t.$$

Napišimo tu jednačinu u sledećem obliku:

$$dy = t^3 dt - 3t^2 dt + t dt$$

Integracijom dobija se sledeća jednačina:

$$y = \frac{t^4}{4} - t^3 + \frac{t^2}{2} + c.$$

Sada ćemo situaciju malo zakomplikovati dodajući u diferencijalnu jednačinu par konstanti, a i b , tako da dobije oblik:

$$\frac{dy}{dt} = ay - b.$$

Uz malo algebarskog preuređivanja ova jednačina se može napisati u sledećem obliku:

$$\frac{\frac{dy}{dt}}{y - (b/a)} = a.$$

Integracijom obe strane dobija se sledeća jednačina:

$$\ln|y - (b/a)| = at + c,$$

gde je c proizvoljna konstanta. Sada izdvojimo y iz prirodnog algoritma i dobija se:

$$y = (b/a) + de^{at},$$

gde je $d = e^c$.

Ponekad integracija linearnih diferencijalnih jednačina prvog reda nije jednostavna. U takvim slučajevima treba jednačine „nekako” konvertovati (transformisati) u nešto što se lako integriše. Termin „nekako” znači da treba pronaći integracioni factor (množilac) koji je neka funkcija $\mu(t)$. Ideja se sastoji u tome da se diferencijalna jednačina pomnoži sa takvim integracionim faktorom tako da se novodobijena jednačina može lako da integriše i rešava.

Podimo od toga da treba da rešimo diferencijalnu jednačinu sledećeg oblika:

$$\frac{dy}{dt} + 2y = 4.$$

Prvo, pomnožimo tu jednačinu sa članom $\mu(t)$, koji je zamena za nepoznati integracioni faktor. Rezultat množenja je:

$$\mu(t) \frac{dy}{dt} + 2\mu(t)y = 4\mu(t).$$

Sada treba izabrati $\mu(t)$ tako da je moguće prepoznati levu stranu jednačine kao izvod nekog izraza. Na taj način se integracija lako obavlja.

Tajna je u sledećem: leva strana prethodne jednačine mnogo liči na diferenciranje proizvoda $\mu(t)*y$. Iz tog razloga izaberimo $\mu(t)$ tako da leva strana jednačine zaista bude izvod proizvoda $\mu(t)*y$.

Izvod od $\mu(t)*y$ po t je:

$$\frac{d[\mu(t)y]}{dt} = \mu(t) \frac{dy}{dt} + \frac{d\mu(t)}{dt} y.$$

Poredeći prethodne dve jednačine član po član dobijamo da je:

$$\frac{d\mu(t)}{dt} = 2\mu(t).$$

Dobili smo diferencijalnu jednačinu koju je moguće rešiti. Preuređivanjem jednačine dobija se:

$$\frac{d\mu(t)/dt}{\mu(t)} = 2.$$

Ova jednačina se sada može preurediti tako da dobije sledeći oblik:

$$\frac{d\mu(t)}{\mu(t)} = 2dt.$$

Integracijom dobija se:

$$\ln|\mu(t)| = 2t + b,$$

gde je b proizvoljna konstanta integracije.

Eksponcenciranjem (stepenovanjem broja e) obe strane jednačine dobija se:

$$\mu(t) = c * e^{2t},$$

gde je c proizvoljna konstanta.

Da zaključimo, dobijen je integracioni faktor — $\mu(t) = c * e^{2t}$.

Pošto je određen integracioni faktor, sada može da se pomnoži početna diferencijalna jednačina sa $\mu(t)$ i dobija se:

$$c * e^{2t} * \frac{dy}{dt} + 2c * e^{2t} * y = 4 * c * e^{2t}.$$

Iz ove jednačine moguće je izbaciti c ako se uradi deljenje sa c (takođe, može se uzeti da je $c=1$ zato što mi tražimo proizvoljni integracioni faktor). Jednačina sada ima oblik:

$$e^{2t} * \frac{dy}{dt} + 2e^{2t} * y = 4 * e^{2t}.$$

Da se podsetimo, kada koristimo integracioni faktor, pokušavamo da pronađemo funkciju $\mu(t)$ koja, kada pomnožimo obe strane diferencijalne jednačine sa njom, kao rezultat daje levu stranu diferencijalne jednačine koja je izvod proizvoda. To nam omogućava da rešimo diferencijalnu jednačinu.

U datom primeru uočavamo da je leva strana izvod od $e^{2t} * y$.

Drugim rečima, diferencijalna jednačina je „savladana”, tako da sada ima oblik:

$$\frac{d(e^{2t}y)}{dt} = 4 * e^{2t}.$$

Integracijom obe strane jednačine dobija se:

$$e^{2t} * y = 2 * e^{2t} + c.$$

Uz pomoć pogodnih algebarskih zahvata (deljenje sa e^{2t}) dobija se rešenje oblika:

$$y = 2 + c * e^{-2t}.$$

U nekim slučajevima metoda integracionog faktora neće biti od pomoći i tada treba pokušati sa nekim drugim metodama rešavanja.

2. DIFERENCNE JEDNAČINE

Diferencne (diferentne) jednačine su, jednostavno rečeno, jednačine u kojima se kao nepoznate javljaju funkcije definisane na nekom diskretnom skupu, odnosno funkcije čije se vrednosti izračunavaju *rekurzivno* iz datog skupa vrednosti. U slučaju kad je domen takvih funkcija skup prirodnih brojeva, onda su one ustvari nizovi, pa će se u diferencnim jednačinama kao nepoznate uglavnom pojavljivati nizovi. Veliki broj procesa koji se odvijaju oko nas (fizičkih, hemijskih, bioloških, ekonomskih, društvenih,...) mogu biti predstavljeni diskretnim modelima, koji su ili diferencna jednačina ili sistem diferencnih jednačina. Uglavnom se ti procesi prate u diskretnom vremenu, za razliku od procesa za koje je bitno vreme u kontinualnom (kontinuiranom) smislu i koji se opisuju diferencijalnim jednačinama.

Pimeri koji slede imaju za cilj da ilustruju kako u stvarnosti nastaju diferencne jednačine i to u različitim sferama života.

Obračun kamate

Jedna od najčešćih situacija u životu savremenog čoveka jeste raspolaganje novcem. Skoro svaki čovek ima otvoren račun u banci, a poznato je da se na uložena sredstva obračunava kamata. Budući da se kamata obračunava u diskretnom vremenu, jasno je da će se ovde pojaviti diskretni model, za koji ćemo videti da se opisuje diferencnom jednačinom.

Pretpostavimo da smo uložili 500 evra u banku, gde se kamata zaračunava mesečno po godišnjoj stopi od 6%. Potražimo odgovore na sledeća pitanja:

- i) Koliko ćemo novca imati u banci nakon 10 godina?
- ii) Koliko vremena treba da protekne od momenta ulaganja pa da se uloženi novac udvostruči?

Neka je I_n iznos novčanog uloga nakon n -tog meseca. Tada je $I_0 = 500$. Nakon $(n + 1)$ -og meseca ulog iz prethodnog meseca će biti uvećan za iznos kamate obračunate na ulog I_n , to jest

$$\begin{aligned} I_{n+1} &= I_n + 0,005I_n \\ &= 1,005I_n \end{aligned}$$

za $n = 0, 1, 2, \dots$ (što je diferencna jednačina, jer je u njoj nepoznata niz I_n).

Očito je sada:

$$I_n = 1,005I_{n-1}$$

$$= (1,005)^2 I_{n-2}$$

.

.

.

$$= (1,005)^n I_0$$

$$= (1,005)^n \cdot 500$$

i) Nakon 10 godina, tj. 120 meseci, iznos uloženog novca s kamatama je:

$$I_{120} = (1,005)^{120} \cdot 500$$

$$= 909,70 \text{ evra.}$$

ii) Iz jednačine:

$$I_n = 2I_0 = (1,005)^n I_0$$

sledi:

$$n = \frac{\log 2}{\log 1,005} = 139,$$

dakle, za 139 meseci, ili za 1, 6 godina, uloženi novac će se udvostručiti.

Populacioni model

Posmatrajmo jednu populaciju čija veličina je poznata na diskretnom skupu vremen-
skih intervala $t_n = (\Delta t)n$, gde je t fiksiran interval, a n je nenegativan ceo broj. Označimo
veličinu populacije u vremenu t_n sa P_n .

Pretpostavimo sada, dok je populacija mala, da je promena populacije od vremena t_n
do t_{n+1} data sa

$$P_{n+1} = \alpha P_n,$$

gde je α pozitivna konstanta. Dalje, pretpostavimo da u velikim veličinama popu-
lacije postoji kompeticija među članovima populacije (za hranu i druge resurse). Ova
kompeticija može se modelovati (modelirati) dodavanjem prethodnoj jednakosti člana
oblika $-\beta P_n (P_n - 1)$, gde je $\beta > 0$. Grubo, ovo znači da je veličina kompeticije proporci-
onalna ukupnom broju mogućih interakcija među članovima populacije i da kompeticija
ima negativan efekat na rast populacije. Kombinovanjem, ova dva rezultata daju sledeću
populacionu dinamičku jednačinu:

$$P_{n+1} = \alpha P_n - \beta P_n (P_n - 1),$$

ili

$$P_{n+1} = (\alpha + \beta) P_n - \beta P_n^2.$$

Koeficijent $\alpha + \beta$ je efektiv ili neto „stopa rađanja”, dok je β mera kompeticije. Uočimo da je α stopa rađanja u odsustvu kompeticije. Iz ovog jednostavnog modela možemo zaključiti da, kad se javlja kompeticija, onda raste neto stopa rađanja.

Radioaktivnost

Pokazaćemo sada kako se proces raspada radioaktivne supstance može predstaviti diskretnim modelom, budući da se u ovakvim situacijama proces može pratiti u diskretnom vremenu (u ovom slučaju svakog sata).

Poznato je da je količina radioaktivnog izotopa olova *Pb-209* na kraju svakog sata proporcionalna zatečenoj količini na početku sata. Ako je vreme poluraspada olova *Pb-209* 3,3 sata, može se, recimo, postaviti pitanje: za koliko vremena će količina olova *Pb-209* opasti za 20%?

Označimo sa m_n količinu olova *Pb-209* na kraju n -tog sata. Prema uslovima zadatka imamo:

$$m_{n+1} = \alpha m_n,$$

gde je α koeficijent proporcionalnosti (koji je pozitivan). Kao i u slučaju obračuna kamate, i ovde zaključujemo da je:

$$m_n = \alpha^n m_0.$$

Budući da je period poluraspada olova *Pb-209* jednak 3,3 sata, iz poslednje jednačine dobijamo:

$$\frac{1}{2} m_0 = \alpha^{3,3} m_0,$$

odakle sledi:

$$\alpha = e^{-\frac{\ln 2}{3,3}}.$$

Zbog toga je:

$$\frac{4}{5} m_0 = \left(e^{-\frac{\ln 2}{3,3}} \right)^n m_0,$$

odakle, konačno, imamo:

$$n = 1,06.$$

Dakle, nakon 1,06 sati količina olova će pasti za 20%.

Diferencne jednačine pri rešavanju diferencijalnih jednačina

Ovde ćemo demonstrirati primer pojave diferencnih jednačina pri rešavanju diferencijalnih jednačina pomoću redova.

Posmatrajmo diferencijalnu jednačinu:

$$y'' + 3xy' + 3y = 0.$$

Potražimo jedno rešenje date jednačine u obliku stepenog reda:

$$y(x) = \sum_{n=0}^{\infty} a_n x^n,$$

pri čemu su a_n ($n = 0, 1, \dots$) koeficijenti koje treba odrediti. Tako imamo:

$$xy' = x \sum_{n=0}^{\infty} n a_n x^{n-1} = \sum_{n=2}^{\infty} (n-2) a_{n-2} x^{n-2},$$

$$y'' = \sum_{n=0}^{\infty} n(n-1) a_n x^{n-2} = \sum_{n=2}^{\infty} n(n-1) a_n x^{n-2},$$

$$y = \sum_{n=0}^{\infty} a_n x^n = \sum_{n=2}^{\infty} a_{n-2} x^{n-2}.$$

Zamenom tih izraza u datu diferencijalnu jednačinu, dobijamo:

$$\sum_{n=2}^{\infty} [n(n-1)a_n + 3(n-1)a_{n-2}] x^{n-2} = 0.$$

Odavde sledi da svaki koeficijent dobijenog reda mora biti jednak nuli, tj.

$$n(n-1)a_n + 3(n-1)a_{n-2} = 0, \quad n = 2, 3, 4, \dots$$

odnosno:

$$na_n + 3a_{n-2} = 0,$$

što je diferencna jednačina. Rešavanjem ove jednačine dobija se konkretan izgled stepenog reda $y(x) = \sum a_n x^n$, te nakon utvrđivanja područja konvergencije, moguće je zaključiti o kojoj je funkciji reč, koja ustvari predstavlja sumu dobijenog reda.

Diferencni račun

Kao što u rešavanju i analizi diferencijalnih jednačina značajnu ulogu igra diferencijalni i integralni račun, tako isto u rešavanju i analizi (posebno linearnih) diferencnih jednačina vrlo značajno mesto zauzima diferencni račun, koji predstavlja diskretni analogon diferencijalnog i integralnog računa.

Diferencni operator

U diferencnom računu bitna su dva operatora, koja ćemo definisati i za koje ćemo navesti neke osnovne osobine. Reč je o diferencnom (diferencijskom) operatoru i translacijskom operatoru.

Definicija: Neka je $x(t)$ funkcija realne ili kompleksne promjenljive t . *Diferencni operator* (ili razliku prvog reda) definišemo jednakošću

$$\Delta x(t) = x(t+1) - x(t).$$

Uglavnom ćemo u budućim primenama smatrati da je domen funkcije x skup uzastopnih celih brojeva, kao npr. skup $N = \{1, 2, 3, \dots\}$. U tom slučaju promenljivu t zamenićemo oznakom n , a izraz $x(n)$ sa x_n , pa ćemo prethodnu jednakost pisati u obliku:

$$\Delta x_n = x_{n+1} - x_n.$$

Međutim, ovde ćemo uglavnom, zbog opštosti, za domen funkcije x uzimati neki neprekidni skup vrednosti promenljive t , kao što je interval $[0, +\infty)$ ili kao što je kompleksna ravan.

Uočimo da veličina koraka od jedne jedinice, korišćen u definiciji, realno gledajući nije nikakva restrikcija. Naime, posmatramo li diferencni operator $y(s+h) - y(s)$, s veličinom koraka $h > 0$, i uvedemo li smenu $x(t) = y(th)$, imaćemo:

$$y(s+h) - y(s) = y(th+h) - y(th) = x(t+1) - x(t) = \Delta x(t).$$

Takođe, uočimo da se diferencni operator može primenjivati i na funkcije dve ili više promenljivih, ali je potrebno u indeksu operatora naznačiti koja će promenljiva biti translirana za jednu jedinicu. Na primer:

$$\Delta_t(t^2 \cdot 2^n) = (t+1)^2 \cdot 2^n - t^2 \cdot 2^n = (2t+1) \cdot 2^n,$$

dok je

$$\Delta_n(t^2 \cdot 2^n) = t^2 \cdot 2^{n+1} - t^2 \cdot 2^n = t^2 \cdot 2^n.$$

Razlike višeg reda uobičajeno se definišu pomoću kompozicije diferencnog operatora sa samim sobom. Tako je razlika drugog reda:

$$\begin{aligned}\Delta^2 x(t) &\equiv \Delta(\Delta x(t)) = \Delta(x(t+1) - x(t)) \\ &\equiv (x(t+2) - x(t+1)) - (x(t+1) - x(t)) \\ &\equiv x(t+2) - 2x(t+1) + x(t).\end{aligned}$$

Uopšteno, razlika reda n definiše se *indukcijom*:

$$\Delta^n x(t) = \Delta(\Delta^{n-1} x(t)), \quad (n = 2, 3, \dots),$$

stoga zaključujemo da važi:

$$\begin{aligned}\Delta^n x(t) &= x(t+n) - nx(t+n-1) + \frac{n(n-1)}{2}x(t+n-2) + \dots + (-1)^n x(t) \\ &= \sum_{k=0}^n (-1)^k \binom{n}{k} x(t+n-k).\end{aligned}$$

3. SIGNALI I SISTEMI

Signal je vremenski promenljiv **fizički fenomen** koji nosi neku **informaciju**. Signali imaju veoma različite pojavne oblike: govorni signal kod koga je informacija sadržana u izgovorenim glasovima, zvučni signal koji informaciju predstavlja pomoću boje i visine tona, toplotni signal od koga je informacija iskazana različitim temperaturama, svetlosni signal koji informaciju predstavlja vizuelno ili pomoću različitog intenziteta svetlosti, ili pomoću različitog trajanja svetlih i tamnih segmenata, električni signal kod koga je informacija sadržana u nekoj od karakteristika signala (amplituda ili učestanost) itd.

Pri osmatranju (opažanju; observiranju) nekog fizičkog fenomena o kome se želi dobiti informacija veoma često se koriste merni instrumenti koji informaciju o nekoj fizičkoj veličini daju u formi električnog signala. Ova forma signala je veoma pogodna i ukoliko se želi prenos signala na daljinu.

Šum (slučajan signal) je vremenski promenljiv fizički fenomen koji ne nosi informaciju. Šta više, on najčešće zamađuje informaciju koju nosi signal i kao takav je nepoželjan.

Sistem je skup međusobno povezanih elemenata koji deluju kao jedna celina, koja ima sposobnost da kada je *pobuđena nekim signalom* na svom ulazu proizvede odgovarajuću *reakciju (odziv)* na svom izlazu. Sistem može biti električni, mehanički, biološki, ekonomski, politički itd. Neki sistemi su *prirodni*, nastali evolucijom tokom niza godina, a neki su *veštački*, projektovani sa idejom da se omogući obavljanje neke funkcije (npr. telekomunikacioni sistemi za prenos signala na daljinu).

Signali se obrađuju u sistemu. Sama svrha obrade može biti raznovrsna. Signal se može obrađivati da bi se iz njega izvukla informacija i predstavila na neki razumljiviji ili pogodniji način, ili da bi se na osnovu nje predvideli neki fenomeni koji će se dogoditi u budućnosti. Pored toga, obradom u sistemu signal se može transformisati u oblik pogodniji za prenos na daljinu. Isto tako, obrada signala može imati za cilj da se na izlazu sistema postigne neki željeni efekat.

Uprkos izuzetnoj raznovrsnosti signala i sistema, pokazuje se da oni imaju neke zajedničke karakteristike i da se mogu, bar u nekoj meri, analizirati na isti način. U osnovi analize signala i sistema leži njihovo predstavljanje pomoću odgovarajućih jednačina, odnosno *formiranje matematičkih modela*.

Matematički model signala i sistema je skup jednačina kojima se oni mogu opisati. Pri tome, budući da i signali i sistemi mogu biti veoma složeni, matematički model je veoma često samo aproksimacija realnog fizičkog signala ili sistema. To nadalje znači da se i rezultati analize moraju kritički posmatrati i da se stalno mora voditi računa o odnosu teorijskih rezultata i njihove fizičke ostvarljivosti.

U zavisnosti od vrste signala, matematička funkcija kojom se oni modeliraju može imati jednu ili više nezavisnih promenljivih. Kod jedne široke klase signala nezavisna promenljiva je *vreme*. Pokazaće se međutim, da je sa gledišta analize, veoma često pogodno da se određenim transformacijama vremenski promenljive funkcije preslikaju u domen neke druge nezavisno promenljive veličine. U tom smislu signali i sistemi se analiziraju u *vremenskom, frekvencijskom i kompleksnom* domenu.

Procesni signali i informacije o procesu

Elementarne podprocese spajaju energetski, materijalni i informacioni tokovi tj. procesni signali. Procesne signale karakterišu pokazatelji stanja, kao što su: masa, protok, napon, struja, pritisak, temperatura itd. Pokazatelji stanja procesnih signala su merljive veličine.

Signali su vremenske, prostorne funkcije koje pored svog fizičkog iznosa i konkretnog efekta na okruženje uvek nose sa sobom i informacije. Analizom informacionog sadržaja signala utvrđuje se stanje i ispunjenost tehnoloških zahteva u tehnološkom procesu.

U tehničkim sistemima signal je uvek fizička veličina koja je usmerena tj. ima svoj izvor i odredište. Sa stanovišta upravljanja procesima značaj uvek ima informacioni, a ne materijalno-energetski karakter signala.

Informacija je deo saznanja kojim se smanjuje ili u potpunosti odstranjuje neka neizvesnost. Npr. neizvesnost u oceni stanja nekog tehnološkog procesa može se otkloniti samo ako se poseduje određena količina informacija o procesu. Informacije koje nosi signal kvantitativno se mogu iskazati preko parametara signala. Informacije iz signala dobijaju se odabirom i obradom podataka o parametrima signala.

Podatak je svaka kvalitativno i kvantitativno izražena veličina koja je na neki način izmerena, zabeležena ili računskim operacijama određena. Neka veličina može biti podatak ako postoji određena, makar i minimalna verovatnoća, da će biti iskorišćena tj. da će biti pretvorena u informaciju.

Podaci o signalima tj. stanjima tehnološkog procesa prikupljaju se posmatranjem, logičkim zaključivanjem i merenjem parametara signala. Prenosjenje, obrada i čuvanje podataka ili informacija moguća je samo pomoću signala kao nosioca informacija. Nosioc informacija, tj. signal mora bar preko jednog svog kvalitativnog ili kvantitativnog pokazatelja tj. parametra da prenosi informaciju. Analizom informacionog sadržaja signala tj. obradom parametara signala, može se steći uvid u događaje koji se odvijaju u izvoru signala, ali se mogu predvideti i događaji koji će se odvijati u delu sistema koji će signal prihvatiti.

Uporednom analizom više signala iz sistema ili amplituda istog signala u različitim trenucima vremena, može se izvesti zaključak o stanju i parametrima sistema koji predstavlja izvor signala.

Uzmimo, na primer, da je posmatrani sistem neka izdvojena sredina u koji smo smestili električno grejno telo. Menjanjem napona napajanja grejača menjaćemo i količinu oslobođene toplotne energije iz grejača. Oslobođena toplotna energija se ili akumulira u izdvojenoj sredini ili se odaje okruženju. Pokazatelj stanja akumulacije toplotne energije je temperatura sredine. Informacija o vrednosti temperature može se dobiti merenjem. Ako se merenje temperature vrši primenom termoelementa tada naponski signal termoelementa postaje nosioc informacija o temperaturi. Obradom promene napona termoelementa može se utvrditi karakter promene temperature. U ovom slučaju informacija o stanju temperature se pojavljuje u vidu naponskog signala čija amplituda u svakom trenutku mora biti srazmerna vrednosti temperature. Promena temperature se znači preslikava u promenu amplitude naponskog signala tj. u signal koji se može predstaviti funkcionalnom zavisnošću čiji je argument vreme. Između promena temperature i promena amplitude

naponskog signala mora postojati određena i unapred poznata međuzavisnost. Signali koji zadovoljavaju ovaj uslov spadaju u kategoriju analognih signala.

Informacije o vremenski promenljivim signalima raspoređuju se na ceo vremenski tok postojanja signala tj. za svaki trenutak vremena može se odrediti odgovarajuća vrednost merenog signala. Koncentracija informacije u trenutnoj vrednosti signala je mala, s obzirom da sadrži samo informaciju o amplitudi i vremenu. Zadovoljavajući nivo saznanja o promenama u tehnološkom procesu u ovom slučaju može se dobiti samo obradom amplituda signala dobijenih u različitim trenucima vremena.

Povećavanje informatičkog sadržaja signala može se postići matematičkim preslikavanjem funkcionalnih zavisnosti signala iz vremenskog domena u neki drugi često imaginarni domen kao što je npr. kompleksna ravan.

Kvalitet i efikasnost ovih preslikavanja zavisi od:

- ♦ potrebnog vremena za prikupljanje polaznog skupa,
- ♦ broja elemenata polaznog skupa,
- ♦ analitičkog oblika procedure preslikavanja,
- ♦ vremena potrebnog za ostvarivanje preslikavanja,
- ♦ jednoznačnosti preslikavanja,
- ♦ mogućnosti interpretacije rezultata preslikavanja itd.

Za realizaciju zadataka upravljanja procesima od posebnog je značaja izbor merne metode za dobijanje informacija, pretvaranje mernog signala u formu pogodnu za prenos preko raspoloživih komunikacionih kanala i određivanje postupka obrade.

Vrste signala

U zavisnosti od aspekta koji se posmatra postoje različite podele signala.

Definisanost signala u vremenu (nezavisna promenljiva)

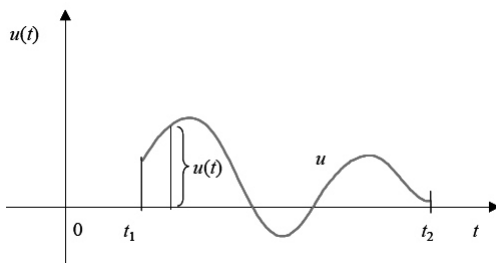
Kontinualni signali – definisani u svakom trenutku vremena na nekom vremenskom intervalu. Ovi signali se označavaju i kao *analogni* signali jer je njihova promena analogna promeni neke fizičke veličine koja se posmatra i o kojoj oni nose informaciju.

Treba istaći da je najveći broj fizičkih fenomena koji se posmatra u po svojoj prirodi kontinualan.

Diskretni signali – definisani samo u određenim trenucima vremena na nekom vremenskom intervalu.

Diskretni signali mogu nastati tako što se neki kontinualni fizički fenomen prati (meri) u određenim diskretnim trenucima vremena. Tako se, na primer, meteorološki podaci u jednom gradu (temperatura, pritisak, brzina vetra, vlažnost vazduha itd.) beleže svakog sata. Pored toga, diskretni signali mogu nastati i u nekom procesu koji je po svojoj prirodi diskretan (na primer obračun dnevnog interesa na računu štednje). Konačno, čak i pod uslovima da se vrši kontinualno merenje na nekom fizičkom procesu kontinualni signali se, zbog potreba obrade, mogu *diskretizovati* tako što se u nekim određenim trenucima

vremena uzimaju vrednosti kontinualnog signala – *odbirci (odmerci)* i na osnovu njih se formira diskretni signal. Ovaj postupak se naziva *odabiranje* (engl. *sampling*), a vremenski interval uzimanja odbiraka se zove *perioda odabiranja*. Samo se po sebi razume da se u ovom slučaju postavlja pitanje veze periode odabiranja i tačnosti sa kojom dobijeni diskretni signal opisuje kontinualni fenomen.

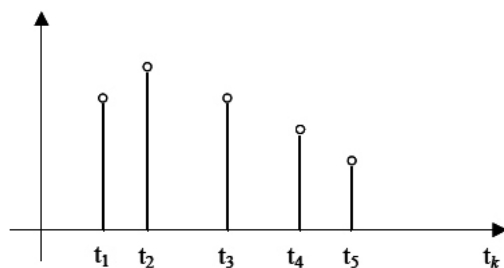


Sl. 3.1 Kontinualni signal

Vrednosti amplitude

Signali kontinualne amplitude – imaju amplitudu koja uzima vrednost iz kontinualnog skupa vrednosti definisanog na nekom segmentu.

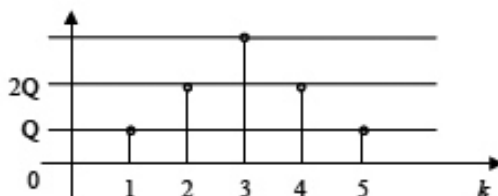
Signali diskretne amplitude – imaju amplitudu koja uzima vrednosti iz nekog konačnog skupa vrednosti definisanog na nekom segmentu. Ovi signali se najčešće dobijaju *kvantizacijom* signala kontinualne amplitude. U postupku kvantizacije vrednost signala kontinualne amplitude u nekom trenutku vremena se zaokruživanjem ili odsecanjem dovodi na jednu od diskretnih vrednosti amplitude. U principu, signali diskretne amplitude mogu biti kontinualni ili diskretni po vremenu.



Sl. 3.2 Vremenski diskretni signal

Treba istaći da *obrada signala pomoću računara nužno dovodi do diskretizacije i kvantizacije signala*. Diskretizacija je neophodna zato što se vrednosti signala u računar mogu uneti samo u određenim trenucima vremena. Kvantizacija se zahteva zbog konačne dužine reči računara koja samim tim određuje i broj različitih kvantizacionih nivoa.

Pri tome, kasnije će biti pokazano, da se pod određenim uslovima iz diskretizovanog, makar teorijski, može izvršiti rekonstrukcija originalnog kontinualnog signala. Međutim, proces kvantizacije je u potpunosti ireverzibilan i nema načina da se izvrši rekonstrukcija kvantizovanog signala.



Sl. 3.3 Diskretizacija signala po amplitudi (kvantizacija) i vremenu

Karakteristične kontinualne funkcije

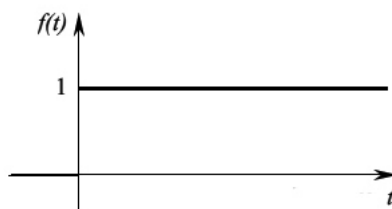
Signal jediničnog skoka

Ovo je signal koji je najlakše generisati i grafički predstaviti. Mnogi ulazni signali objekta upravljanja se mogu približno aproksimirati ovom funkcijom (uključivanje napona u električnom kolu, skokovita promena sile u mehaničkom sistemu, zakret kormila broda u cilju promene pravca). Pretpostavka je da ovaj signal ima vrednost jedan od trenutka posmatranja ($t=0$) što je dato na sledećoj slici. Ovaj signal se obično opisuje sa:

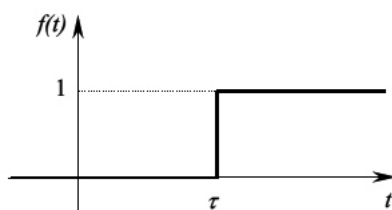
$$f(t) = \begin{cases} 1: t > 0 \\ 0: t < 0 \end{cases}$$

Treba primetiti da funkcija nije definisana za $t=0$, jer je pretpostavka da se u tom trenutku vrednost skokovito promeni sa nula na jedan. To predstavlja matematičku idealizaciju jer je za takvu promenu neophodno neko, ma koliko kratko, vreme. Funkcija se često označava i sa $1(t)$.

Napomena: Od interesa je navesti i zakašnjenu jediničnu funkciju za vrednost τ , gde je τ konstanta. Ova funkcija se zapisuje u obliku $1(t-\tau)$, a predstavljena je grafički na slici.



Sl. 3.4 Funkcija jediničnog skoka



Sl. 3.5 Funkcija jediničnog skoka zakašnje-
nja τ

Impulsna funkcija - impulsni signal

Ova funkcija se jednostavno dobije diferenciranjem funkcije jediničnog skoka. Za lakše razumevanje oblika ove funkcije pogodno je poći od njene aproksimacije date na sledećoj slici.

Ova aproksimacija odgovara razlici dve odskočne funkcije sa skokom $1/\varepsilon$ pri čemu je druga zakašnjena za ε u odnosu na prvu. U graničnom slučaju kada ε teži vrednosti nula dobija se impulsna funkcija. Ona se obično označava sa δ i piše u obliku:

$$\delta = \lim_{\varepsilon \rightarrow 0} \delta^*(t) = \begin{cases} 0; t \neq 0 \\ \infty; t = 0 \end{cases}$$

$$\int_{-\infty}^{\infty} \delta(t) dt = 1.$$

Iz zadnjeg se vidi da funkcija ima beskonačnu vrednost za $t=0$ ali je istovremeno površina ispod funkcije konačna i jednaka 1. Ona se grafički obično predstavlja kao na slici.

Bez obzira na neuobičajenu matematičku definiciju ova funkcija je pogodna za aproksimaciju nekih signala. Signali intenzivnih smetnji koji su vrlo kratkog trajanja (snažni ali kratki udari vetra na letelicu, kratkotrajan kratak spoj u električnoj mreži i slično) mogu se dosta uspešno aproksimirati ovom funkcijom.

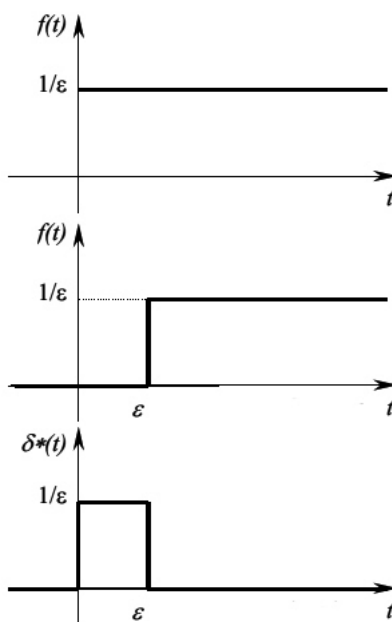
Linearna (rampa) funkcija

Ako se uzme integral jedinične odskočne funkcije dobije se funkcija čija vrednost raste linearno u vremenu od vrednosti nula, koju ima u trenutku $t=0$. Ova funkcije se predstavlja izrazom:

$$f(t) = \begin{cases} t; & t > 0 \\ 0; & t < 0 \end{cases}$$

Očigledno se takođe može predstaviti sa: $f(t)=t*1(t)$. Grafički je ova funkcija predstavljena na sledećoj slici.

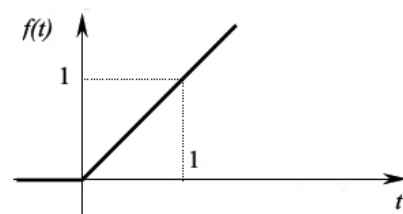
Sa dijagrama je očigledno zašto se ova funkcija naziva i brzinskom funkcijom. Ona je pogodna za predstavljanje signala koji se karakterišu konstantnom promenom (ugao zakreta osovine motora koja se obrće konstantnom brzinom). Specifičnost ove funkcije je to što sa vremenom njena vrednost teži prema beskonačnosti.



Sl. 3.6 Aproksimacija impulsne funkcije



Sl. 3.7 Grafički prikaz impulsne funkcije



Sl. 3.8 Grafički prikaz linearne funkcije

Eksponencijalna funkcija

Polazi se od eksponencijalne funkcije

$$f(t) = e^{-at}; a > 0; e \approx 2.718$$

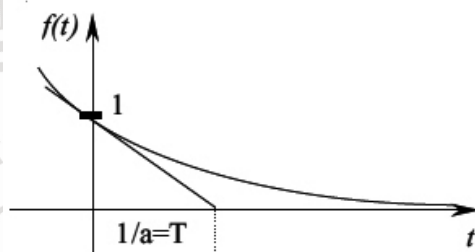
Tangenta na krivu u $t=0$ ima nagib određen prema:

$$\left. \frac{df}{dt} \right|_{t=0} = \left. \frac{d(e^{-at})}{dt} \right|_{t=0} = -a.$$

Kako je eksponent neimenovan broj zgodno je usvojiti $T=1/a$, jer se dobije konstanta koja ima dimenziju vremena. Tada je:

$$\left. \frac{df}{dt} \right|_{t=0} = -\frac{1}{T}.$$

Očigledno za veće vrednosti T funkcija $f(t)$ ima sporiju promenu (opadanje prema stacionarnoj vrednosti). Važi i obrnuto za poznat oblik $f(t)$ može se odrediti vrednost T . Funkcija je grafički predstavljena na slici.



Sl. 3.9 Eksponencijalna funkcija

Funkcija je očigledno različita od nule za $t < 0$, pa je od većeg praktičnog značaja njena modifikacija data sa:

$$f(t) = (1 - e^{-t/T})1(t); T > 0.$$

Grafik ove funkciju je dat na sledećoj slici.

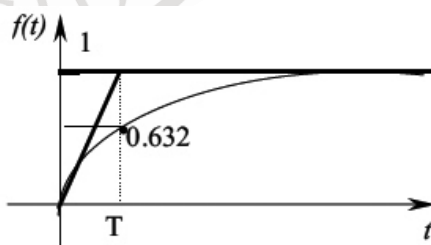
Tangenta na krivu u $t=0$ ima nagib određen prema:

$$\left. \frac{df}{dt} \right|_{t=0} = \left. \frac{d(1 - e^{-t/T})}{dt} \right|_{t=0} = \frac{1}{T}.$$

Očigledno za veće vrednosti T funkcija $f(t)$ ima sporiji porast prema stacionarnoj vrednosti. Važi i obrnuto; za poznat oblik $f(t)$ može se odrediti vrednost T . Takođe iz izraza:

$$f(T) = 1 - e^{-1} = 1 - 0.368 = 0.632$$

se vidi da funkcija za $t=T$ ima vrednost 63.24% od vrednosti $f(\infty) = \lim_{t \rightarrow \infty} f(t)$ (stacionarnog stanja).



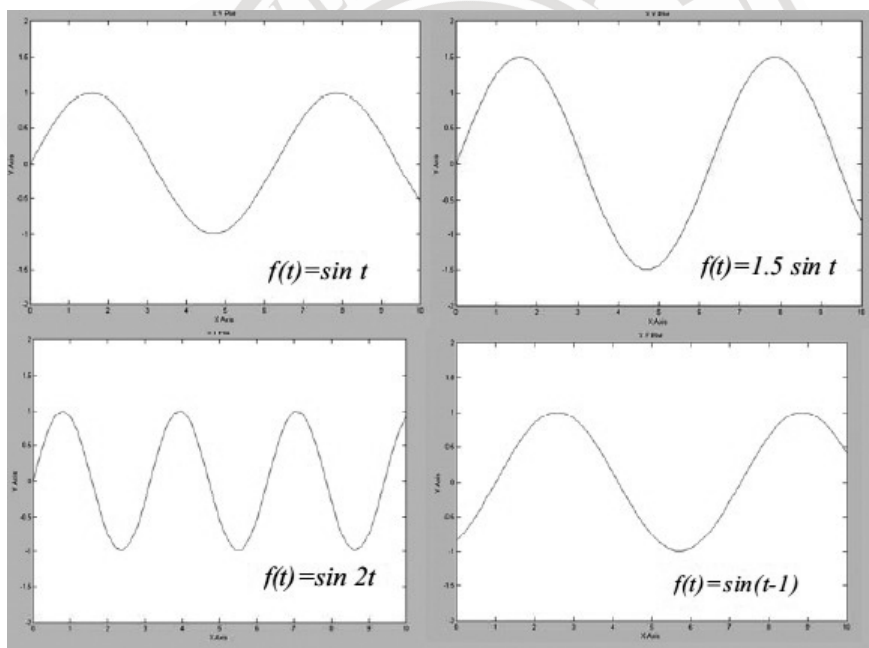
Sl. 3.10 Eksponencijalna funkcija

Sinusna funkcija

Ova funkcija u slučaju jedinične amplitude se može napisati u obliku:

$$f(t) = \begin{cases} \sin \omega t; t \geq 0 \\ 0; t < 0 \end{cases}$$

Funkcija je osnova za kompletnu frekventijsku analizu. Fenomen oscilovanja promenljive bilo da se radi o ulaznom delovanju, smetnji ili signalu koji generiše sam sistem od posebnog su interesa. Veličine bitne za kompletno predstavljanje ovih funkcija su: amplituda, frekvencija i fazni pomak. U odnosu na osnovnu funkciju (sa jediničnom amplitudom, jediničnom frekvencijom i nultim faznim pomakom) na sledećim slikama su dati prikazi iste za: amplitudu 1.5, dvostruko veću frekvenciju i fazno kašnjenje za jedan radijan, respektivno.



Napomena: Režim oscilacija sa konstantnom amplitudom je samo specijalan slučaj režima oscilovanja sa promenljivom amplitudom (rastućom ili opadajućom).

4. LAPLASOVA TRANSFORMACIJA

Problemi analize i sinteze svakog dinamičkog elementa/sistema redovno su vezani za rešavanje diferencijalnih jednačina. Jedan od najjednostavnijih postupaka rešavanja tih jednačina je vezan za primenu Laplasove transformacije. Iz tog razloga biće ukratko date teoretske osnove ove transformacije i neke jednostavnije primene.

Definicija

Za posmatrani kontinualan signal (funkciju) $f(t)$, $0 \leq t < \infty$, Laplasova transformacija je definisana sa

$$L\{f(t)\} = F(s) = \int_0^{\infty} e^{-st} f(t) dt.$$

Nakon operacije integracije nestaje nezavisna promenljiva t , pa ostaje samo zavisnost od promenljive s . Kompleksna promenljiva $s = \sigma + j\omega$ je takva da izraz e^{-st} u zadnjoj jednačini predstavlja prigušenje. Navedeni integral će konvergirati ako realna vrednost promenljive s zadovoljava uslov $\sigma > \sigma_a$, gde je σ_a realna pozitivna konstanta za koju važi

$$\int_0^{\infty} e^{-s_a t} |f(t)| dt < \infty.$$

Za većinu signala u sistemima upravljanja ne mora se posebno voditi računa o ovom problemu. Uvedena transformacija prevodi funkciju $f(t)$ (original) definisanu u vremenu u kompleksno područje $F(s)$ (slika). Napomenimo još da je transformacija ograničena na funkcije koje zadovoljavaju uslov $f(t)=0$, $t<0$. Takve se funkcije nazivaju **kauzalnim**.

Za poznatu funkciju $F(s)$ korišćenjem inverzne Laplasove transformacije moguće je odrediti original prema

$$L^{-1}\{F(s)\} = f(t) = \frac{1}{2\pi j} \int_{\sigma-j\infty}^{\sigma+j\infty} e^{st} F(s) ds$$

Koristeći definicioni izraz može se odrediti Laplasova transformacija funkcija $f(t)$ koje se pojavljuju u sistemima. Za složenije originale je računanje definicionog integral složeno. Iz tog razloga se određivanje Laplasove transformacije složenijih funkcija svodi na izračunavanje preko transformacije elementarnih funkcija uz korišćenje pravila koja važe za Laplasovu transformaciju.

Laplasova transformacija elementarnih funkcija

- ♦ *Funkcija jediničnog skoka.*

U skladu sa definicijom Laplasove transformacije važi:

$$F(s) = L\{1(t)\} = \int_0^{\infty} e^{-st} 1(t) dt = \int_0^{\infty} e^{-st} dt = -\frac{1}{s} e^{-st} \Big|_0^{\infty} = -\frac{1}{s} (e^{-\infty} - e^{-0}) = \frac{1}{s}$$

Napomena: Na isti način se može pokazati da za Laplasovu transformaciju odskočne funkcije nejediničnog skoka a važi

$$L\{a1(t)\} = F(s) = \frac{a}{s}.$$

- ♦ *Impulsna funkcija*

Prema definiciji Laplasove transformacije i impulsne funkcije važi:

$$F(s) = L\{\delta(t)\} = \int_0^{\infty} e^{-st} \delta(t) dt = \int_0^{\infty} \delta(t) dt = 1$$

Na sličan način se mogu odrediti Laplasove transformacije složenijih funkcija. U tim slučajevima moguće su računske greške kod izračunavanja odgovarajućih integrala. Iz tog razloga se često koriste tablice Laplasovih transformacija elementarnih funkcija.

Tablica Laplasove transformacije elementarnih funkcija

Uobičajeni naziv funkcije	$f(t)$	$F(s)$
Impulsna funkcija	$\delta(t)$	1
Jedinična odskočna funkcija	$1(t)$	$\frac{1}{s}$
Ekspencijalna funkcija	e^{-at}	$\frac{1}{s+a}$
Sinusna funkcija	$\sin at$	$\frac{a}{s^2 + a^2}$
Kosinusna funkcija	$\cos at$	$\frac{s}{s^2 + a^2}$

Osobine Laplasove transformacije

1. Linearnost

$$L\left\{\sum_{i=1}^l a_i f_i(t)\right\} = \sum_{i=1}^l a_i F_i(s)$$

2. Vremensko (transportno) kašnjenje

$$L\{f(t-\tau)\} = e^{-s\tau} F(s)$$

3. Pomeranje kompleksnog lika (prigušenje originala)

$$L\{e^{-at} f(t)\} = F(s+a)$$

4. Laplasova transformacija izvoda

$$L\left\{\frac{d^k f(t)}{dt^k}\right\} = s^k F(s) - \sum_{i=1}^k s^{k-i} \frac{d^{i-1} f(t)}{dt^{i-1}}$$

Specijalno u slučaju prvog izvoda je očigledno

$$L\left\{\frac{df(t)}{dt}\right\} = sF(s) - f(t)|_{t=0}$$

5. Laplasova transformacija integrala

$$L\left\{\int_0^t f(\tau) d\tau\right\} = \frac{1}{s} F(s)$$

6. Izvod kompleksnog lika

$$\frac{d^k F(s)}{ds^k} = (-1)^k L\{t^k f(t)\}.$$

Isto svojstvo se češće koristi u obliku tako da odgovara množenju originala linearnom funkcijom. Tada je prethodnu jednačinu zgodno transformisati na

$$L\{t^k f(t)\} = (-1)^k \frac{d^k F(s)}{ds^k}.$$

Specijalno za $k=1$ važi $L\{tf(t)\} = (-1) \frac{dF(s)}{ds}$

7. Teorema početne vrednosti $\lim_{t \rightarrow 0} f(t) = \lim_{s \rightarrow \infty} sF(s)$

8. Teorema konačne vrednosti $\lim_{t \rightarrow \infty} f(t) = \lim_{s \rightarrow 0} sF(s)$

Odrediti Laplasovu transformaciju funkcija datih u narednim primerima

Primer 1:

$$f(t) = a + bt + ct^2, a, b, c = \text{const}$$

Rešenje:

Prema osobini linearnosti i izvodu kompleksnog lika imamo

$$L\{f(t)\} = L\{a1(t)\} + L\{bt\} + L\{ct^2\}$$

$$L\{bt\} = bL\{t1(t)\} = b \frac{-d(\frac{1}{s})}{ds} = b \frac{1}{s^2}$$

$$L\{ct^2\} = cL\{t * t\} = c \frac{-d(\frac{1}{s^2})}{ds} = c \frac{2}{s^3}$$

$$F(s) = \frac{a}{s} + \frac{b}{s^2} + \frac{c}{s^3}$$

Primer 2:

$$f(t) = \cos(at)$$

Rešenje:

$$\frac{d(\sin at)}{dt} = a \cos at, \text{ odatle sledi } \cos at = \frac{1}{a} \frac{d(\sin at)}{dt}$$

Sada je očigledno da važi:

$$L\{\cos at\} = \frac{1}{a} L\left\{\frac{d(\sin at)}{dt}\right\} = \frac{1}{a} s \frac{a}{s^2 + a^2} = \frac{s}{s^2 + a^2}$$

Inverzna Laplasova transformacija

Već je navedeno da je za poznatu funkciju $F(s)$ korišćenjem inverzne Laplasove transformacije moguće odrediti original prema

$$L^{-1}\{F(s)\} = f(t) = \frac{1}{2\pi j} \int_{\sigma-j\infty}^{\sigma+j\infty} e^{st} F(s) ds.$$

Računanje inverzije po ovom izrazu je veoma komplikovano pa se ista određuje na drugi način. Kao osnova se koristi poznavanje inverzne Laplasove transformacije elementarnih funkcija, već datih tabelom Laplasove transformacije. Dalje ćemo posmatrati likove koji su oblika količnika polinoma:

$$F(s) = \frac{P(s)}{Q(s)} = \frac{b_m s^m + \dots + b_1 s + b_0}{s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0}$$

gde je $n > m$ jer se u teoriji sistema najčešće susreću ovake funkcije. Podsetimo da su nule polinoma $P(s)$ i $Q(s)$ nule i polovi funkcije $F(s)$, respektivno. Za nalaženje inverzne

Laplasove transformacije su od posebnog interesa polovi funkcije koji predstavljaju rešenja jednačine:

$$Q(s) = s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0 = 0$$

Zavisno od tih polova se razlikuje nekoliko slučajeva kod određivanja inverzne Laplasove transformacije koji su dalje navedeni.

Svi polovi su realni i prosti (jednostruki)

U ovom slučaju se $F(s)$ može napisati u obliku:

$$F(s) = \frac{P(s)}{(s-s_1)(s-s_2)\dots(s-s_n)}$$

ili u obliku :

$$F(s) = \frac{K_1}{(s-s_1)} + \frac{K_2}{(s-s_2)} + \dots + \frac{K_n}{(s-s_n)}.$$

Koeficijenti K_i se lako određuju metodom neodređenih koeficijenata. Nakon toga se inverzna Laplasova transformacija direktno dobija iz tablice Laplasovih transformacija.

Postoje konjugovano kompleksni polovi

Pretpostavimo da pored realnih postoje i kompleksni polovi funkcije $F(s)$. U tom slučaju $F(s)$ se može predstaviti u obliku:

$$F(s) = \frac{K_1 s + K_2}{s^2 + as + b} + \frac{K_3}{s - s_1} + \dots = F_1(s) + \frac{K_3}{s - s_1} + \dots$$

Određivanje inverzne Laplasove transformacije od $F(s)$ je sada specifično samo za komponentu $F_1(s)$ koja ima par konjugovano kompleksnih polova. Istu je potrebno svesti na oblik

$$F_1(s) = \frac{K_1 s + K_2}{(s + \alpha)^2 + \beta^2}.$$

Vrednosti α i β je lako odrediti iz $2\alpha = a; \alpha^2 + \beta^2 = b$.

Za poznate α i β se $F_1(s)$ lako napiše u obliku:

$$F_1(s) = \frac{K_1(s + \alpha)}{(s + \alpha)^2 + \beta^2} + \frac{K_2 + \alpha K_1}{(s + \alpha)^2 + \beta^2}.$$

Obe komponente funkcije su sada u obliku da se direktno mogu odrediti njihovi originali.

Postoje višestruki polovi

Uzmimo da $F(s)$ ima trostruki realan pol u $s = s_1$ a da su ostali polovi jednostruki. Tada $F(s)$ treba napisati u obliku:

$$F(s) = \frac{K_{11}}{(s - s_1)} + \frac{K_{12}}{(s - s_1)^2} + \frac{K_{13}}{(s - s_1)^3} + R(s).$$

gde $R(s)$ odgovara komponentama koje su posledica svih preostalih polova. Očigledno se računanje u ovom slučaju svodi na korektnu primenu osobine o diferenciranju kompleksnog lika jer je svaki član koji odgovara trostrukom polu, oblika diferencijala prethodnog.

Odrediti inverznu Laplasovu transformaciju funkcija datih u narednim primerima.

Primer:

$$F(s) = \frac{s+1}{s^2+2s}$$

Rešenje:

Funkcija ima dva realna i prosta pola $s_1 = 0, s_2 = -2$ pa se može direktno napisati u obliku:

$$F(s) = \frac{K_1}{s} + \frac{K_2}{s+2}.$$

Postupkom neodređenih koeficijenata lako se dobije:

$$K_1 = \frac{1}{2}; K_2 = \frac{1}{2}.$$

$$f(t) = L^{-1} \left\{ \frac{1}{2s} + \frac{1}{2(s+2)} \right\} = \frac{1}{2} 1(t) + \frac{1}{2} e^{-2t}.$$

Primer:

$$F(s) = \frac{s+5}{s^2+2s+10}$$

Rešenje:

Polovi funkcije su: $s_1 = -1 + j3, s_2 = -1 - j3$.

Funkciju treba transformisati u oblik $F(s) = \frac{s+5}{(s+\alpha)^2 + \beta^2}.$

Tada je:

$$2\alpha = 2 \Rightarrow \alpha = 1;$$

$$\alpha^2 + \beta^2 = 10 \Rightarrow \beta = 3.$$

$$F(s) = \frac{s+5}{(s+1)^2 + 3^2} = \frac{s+1}{(s+1)^2 + 3^2} + \frac{4}{(s+1)^2 + 3^2}$$

$$f(t) = e^{-t} \cos 3t + \frac{4}{3} e^{-t} \sin 3t.$$

5. KRATKI PODSETNIK IZ MATLABA

Unos promenljivih i funkcija

Svaka promenljiva se definiše komandom **syms**, npr.

```
syms a b c x
```

Definišimo sada kvadratnu funkciju

```
f=a+b*x+c*x^2
```

Rešavanje sistema jednačina

Primer: Nađimo x i y koji zadovoljavaju sistem jednačina

$$x^2y^2 = 0$$

$$x-y/2 = a.$$

Prvo zapišimo obe jednačine tako da je desna strana jednaka nuli, tj.

$$x^2y^2 = 0$$

$$x-y/2-a = 0.$$

Rešavanje jednačina se dobija komandom **solve**,

```
syms x y a
```

```
f1 = x^2*y^2
```

```
f2 = x-y/2-a
```

```
[x,y]=solve(f1,f2)
```

Nastavljamo prethodni primer i sada stavimo da je

```
f1 = x^2*y^2-1
```

Matlab javlja grešku. Zašto? Razlog je u tome što su x i y sada vektori (dobijeni kao rešenja prethodnog zadatka), dok je 1 skalar (broj). Možemo obrisati sve dosadašnje vrednosti sa naredbom

```
clear all
```

ili redefinisati vrednosti za x , y pomoću **syms** komande.

```
syms x y
```

```
f1 = x^2*y^2
```

```
f2 = x-y/2-1
```

```
[x,y]=solve(f1,f2)
```

Ili

```
[x,y]=solve('x^2*y^2,x-y/2-1')
```

Izrazi dati pod znacima navoda su simbolički izrazi u Matlabu.

Matrice

Matrice se unose u uglastim zagradama, npr.

$$A = [1 \ 2 \ 3; 4 \ 5 \ 6]$$

$$B = [2 \ 1; 3 \ 3; -1 \ 2]$$

Transponovana matrica:

$$A'$$

Matricno množenje matrica:

$$A * B$$

Množenje matrica po komponentama bi bilo:

$$A . * B$$

(naravno u ovom primeru se dimenzije ne slažu).

Primer: Sistemi linearnih jednačina dati u matricnom obliku $Ax=b$, gde je A matrica, b vektor, mogu se rešiti npr. nalaženjem inverzne matrice (ako je A kvadratna matrica), pa je

$$x = \text{inv}(A) * b$$

nešto bolja rešenja se dobiju pomoću matricnog levog deljenja (bazira se na Gausovom postupku eliminacije)

$$x = A \backslash b$$

a još mnogo bolje je koristiti naredbu **linsolve**, kojoj se mogu navesti opcije da li da koristi LU ili QR dekompoziciju matrice, ili neki drugi numerički postupak.

$$x = \text{linsolve}(A, b)$$

Primer: Rešiti dati sistem jednačina:

$$2x + 3y = 5$$

$$-x + 4y = 7.$$

Rešenje:

$$A = [2 \ 3; -1 \ 4];$$

$$b = [5; 7];$$

$$\text{linsolve}(A, b)$$

M-fajlovi

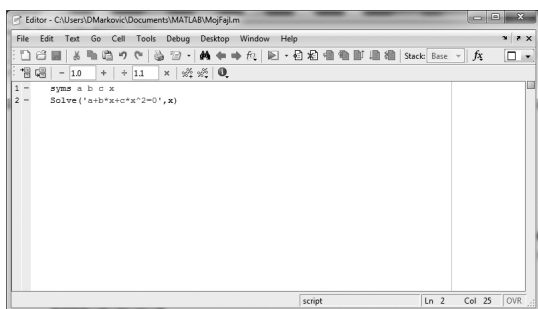
M-fajlovi se pišu u Matlab Editoru, tako što se u Matlabu u meniju **File** izabere opcija **New** a potom **Blank M-File**. Na ekranu se pojavljuje prozor editora.

Sada možemo napisati neki program, npr.

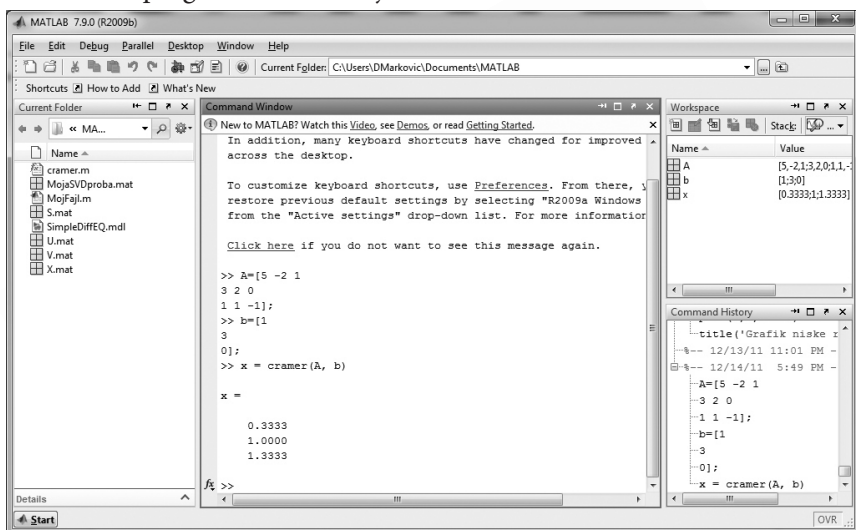
$$\text{syms } a \ b \ c \ x$$

$$\text{Solve}('a+b*x+c*x^2=0', x)$$

Snimimo to pod imenom **MojFajl.m** u podrazumevani (default) folder.



Sada se vratimo u Matlabov komandni prozor i otkucajmo **MojFajl** (bez ekstenzije .m). Ovo će izvršiti program i dati rešenje.



U M-fajlove možemo ubacivati komentare da bi imali podsetnik šta programski kôd konkretno radi. Svaki red koji u m-fajlu počinje znakom % Matlab će ignorisati u smislu izvršavanja.

Korisno je snimiti svoje funkcije u M-fajlove, naročito ako se često koriste. Time i korisničke funkcije imaju isti status kao i Matlabove ugrađene funkcije (npr. sin, cos, exp, ...)

Primer: Napravimo funkciju f koja će svakom elementu vektora x dodati broj 5.

Function izlaz = pluspet(x)

% Ova funkcija svakom elementu vektora x dodaje broj 5

izlaz=x+5;

Zatim sve to snimimo (obavezno!) pod imenom **pluspet.m**. Ovaj program je sada definisao funkciju koja se zove *pluspet*. Izlazna promenljiva koja nosi ime *izlaz* izjednačava se sa vrednošću funkcije u argumentu x. Sada otkucajmo u Matlab komandnom prozoru naziv prethodno definisane funkcije

pluspet(10)

Za slučaj da je ime funkcije manje zvučno pa zaboravite šta vaše funkcije rade, možete pozvati pomoć:

help pluspet

i Matlab će ispisati komentar iz drugog reda M-fajla.

Grafički prikaz podataka

Jedna od možda najboljih osobina MATLAB-a je ta što ima veoma velike grafičke mogućnosti. U MATLAB-u postoji velik broj naredbi pomoću kojih se grafičko prikazivanje podataka može prikazati u dve ili tri dimenzije. Urađeni grafici mogu biti sačuvani u nekom od standardnih formata i kao takvi biti korišćeni u nekim drugim programskim paketima.

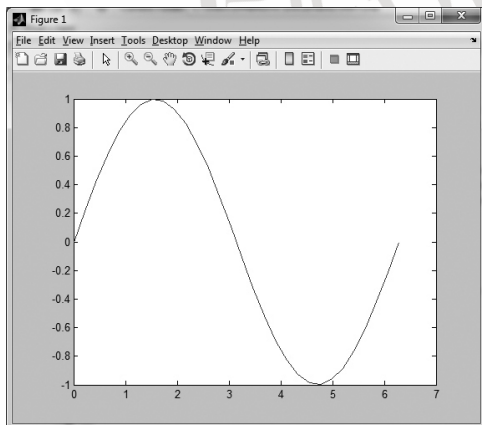
Najjednostavnija naredba za prikazivanje podataka je naredba plot. Prikažimo ovde funkciju:

$y = \sin(x)$ za x od 0 do 2π . Prvo je potrebno definisati interval sa konačnim brojem tačaka između 0 i 2π .

```
» x = linspace ( 0, 2*pi, 30 );
```

```
» y = sin ( x );
```

```
» plot ( x, y )
```



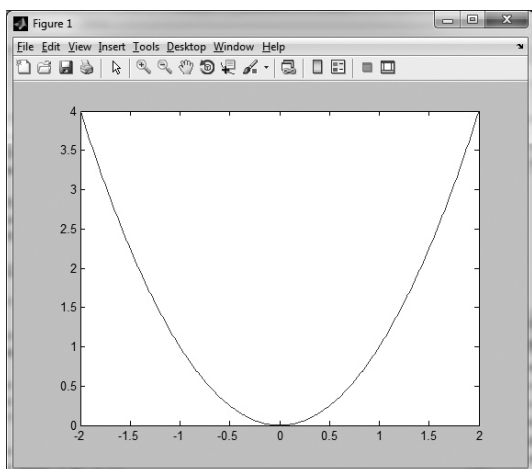
2D grafik crta se zadavanjem dva vektora istih dužina, jednog sa vrednostima zavisne promenljive i drugog sa odgovarajućim vrednostima nezavisno promenljive veličine

Primer: Nacrtati funkciju $y = x^2$ u intervalu od -2 do 2

```
» x = -2 : 0.01 : 2 ;
```

```
» y = x.^2 ;
```

```
» plot ( x , y )
```



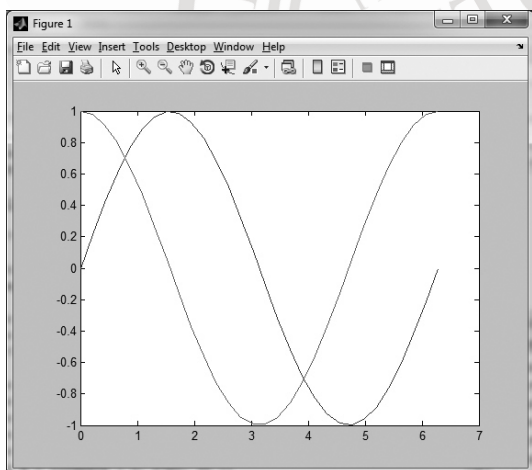
Koristeći naredbu plot moguće je i više grafika prikazati odjednom.

```
» x = linspace ( 0, 2*pi, 30 );
```

```
» y=sin ( x );
```

```
» z=cos ( x );
```

```
» plot ( x, y, x, z )
```



Označavanje grafika i osa u MATLAB-u se postiže naredbama title, xlabel i ylabel.

Evo jednog primera gde su ilustrovane ove naredbe:

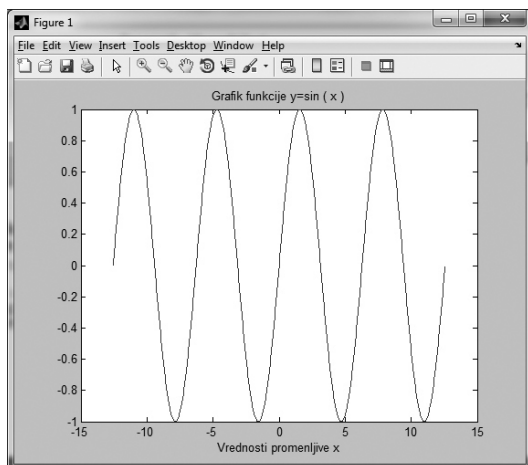
```
» x=-4*pi : pi/100 : 4*pi ;
```

```
» y=sin ( x );
```

```
» plot ( x, y ),
```

```
» title ( ' Grafik funkcije y=sin ( x ) ' ),
```

```
» xlabel ( ' Vrednosti promenljive x ' )
```

- ◆ Komanda/naredba figure svaki put otvara novi prozor za grafik.
- ◆ Raniji prozor se aktivira sa figure(n).
- ◆ Prethodni crtež u aktivnom prozoru se zadržava sa hold on pre iscrtavanja novog grafika (hold all da bi novi grafik bio iscrtan drugom bojom).
- ◆ Skaliranje osa: axis([xmin xmax ymin ymax zmin zmax]), mada može i u prozoru za grafik.

% parabola u 41 tacki

x = -10:0.5:10;

y = x.^ 2;

plot(x,y)

U MATLAB-u postoji mogućnost i da se nacrtaju više različitih grafika na jednom ekranu, za šta se koristi naredba *subplot* koja se navodi pre naredbe za crtanje. Crtanje više nezavisnih grafika u istom prozoru postiže se sa *subplot(m,n,p)* čime se prozor deli na *m* redova i *n* kolona, a elementu se pristupa preko *p* (broji se po redovima)

% parametarski zadata elipsa u 63 tacke

t = 0:0.1:2*pi;

x = cos(t);

y = sin(t);

% parametarski zadata elipsa u 11 tacaka (niza rezolucija)

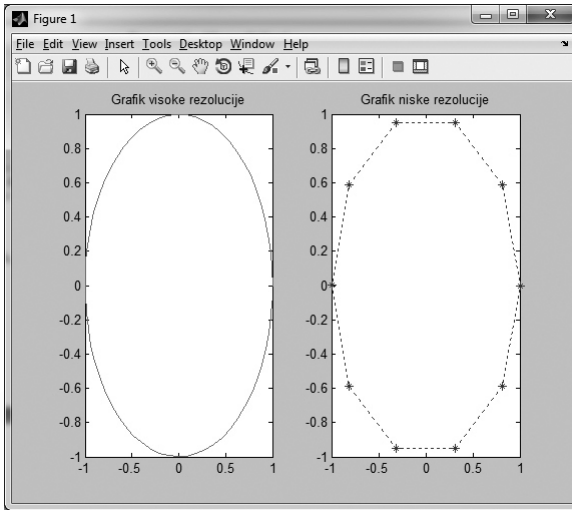
t = 0:pi/5:2*pi;

u = cos(t);

v = sin(t);

% r, g, b, c, m, y, k, - puna, -- isprekidana, : tackasta,
-. crtatacka

```
figure
subplot(1,2,1)
plot(x,y, 'r-')
title('Grafik visoke rezolucije')
subplot(1,2,2)
plot(u,v,'b*:')
title('Grafik niske rezolucije')
```



Sledi nekoliko primera funkcija sa različitim opcijama iscrtavanja.

```
% fazno pomerene sinusoide
x = 0:pi/100:2*pi;
y1 = sin(x);
y2 = sin(x-pi/3);
plot(x,y1,x,y2,'.')
title('Fazno pomerene sinusoide ')
grid on
% ukljucene pomocne linije
% polulogaritamski grafik
x = 0:.1:10;
semilogy(x,10.^x)
% dobija se prava linija
% 3D spirala
t = 0:pi/50:10*pi;
plot3(sin(t),cos(t),t)
```

```

grid on
axis square
% sve tri ose iste duzine
% SEDLO
% 3D grafik f-je dve promenljive
[x, y] = meshgrid(-3:0.1:3,-3:0.1:3);
z = x.^2 - y.^2;
mesh(x,y,z)
figure
surf(x,y,z)
% SOMBRERO
[X,Y] = meshgrid(-8:.5:8);
% Isto sto i meshgrid(-8:.5:8,-8:.5:8)
R = sqrt(X.^2 + Y.^2) + eps;
Z = sin(R)./R;
mesh(X,Y,Z)
hidden off
% Providan mesh grafik
% Grafik funkcije sin(x)/x
x = -10:.15:10;
z = sin(x)./x;
plot(x,z)

```

Kramerovo pravilo

Razmotrimo linearni sistem $Ax = b$, gde je $A = [a_{ij}]$ $n \times n$ invertibilna matrica,

$$x = \begin{bmatrix} x_1 \\ \cdot \\ \cdot \\ x_n \end{bmatrix}, \text{ a } b = \begin{bmatrix} b_1 \\ \cdot \\ \cdot \\ b_n \end{bmatrix}.$$

Sistem ima jedinstveno rešenje dato sa

$$x_k = \frac{\det(B_k)}{\det(A)}, \text{ za } k = 1, \dots, n \quad (1^*)$$

Gde je B_k matrica dobijena iz A kada se k -ta kolona vektor iz A zameni sa kolona vektorom b .

Primer: Reši sledeći linearni sistem jednačina

$$5x_1 - 2x_2 + x_3 = 1$$

$$3x_1 + 2x_2 = 3$$

$$x_1 + x_2 - x_3 = 0,$$

koristeći Kramerovo pravilo.

Rešenje:

Koristeći notaciju jednačine (1*) imamo da je

$$\det(A) = \begin{vmatrix} 5 & -2 & 1 \\ 3 & 2 & 0 \\ 1 & 1 & -1 \end{vmatrix} = -15, \quad \det(B_1) = \begin{vmatrix} 1 & -2 & 1 \\ 3 & 3 & 2 \\ 0 & 1 & -1 \end{vmatrix} = -5,$$

$$\det(B_2) = \begin{vmatrix} 5 & 1 & 1 \\ 3 & 3 & 0 \\ 1 & 0 & -1 \end{vmatrix} = -15, \quad \det(B_3) = \begin{vmatrix} 5 & -2 & 1 \\ 3 & 2 & 3 \\ 1 & 1 & 0 \end{vmatrix} = -20.$$

Iz čega sledi da je

$$x_1 = \frac{-5}{-15} = \frac{1}{3}, \quad x_2 = \frac{-15}{-15} = 1, \quad x_3 = \frac{-20}{-15} = \frac{4}{3}.$$

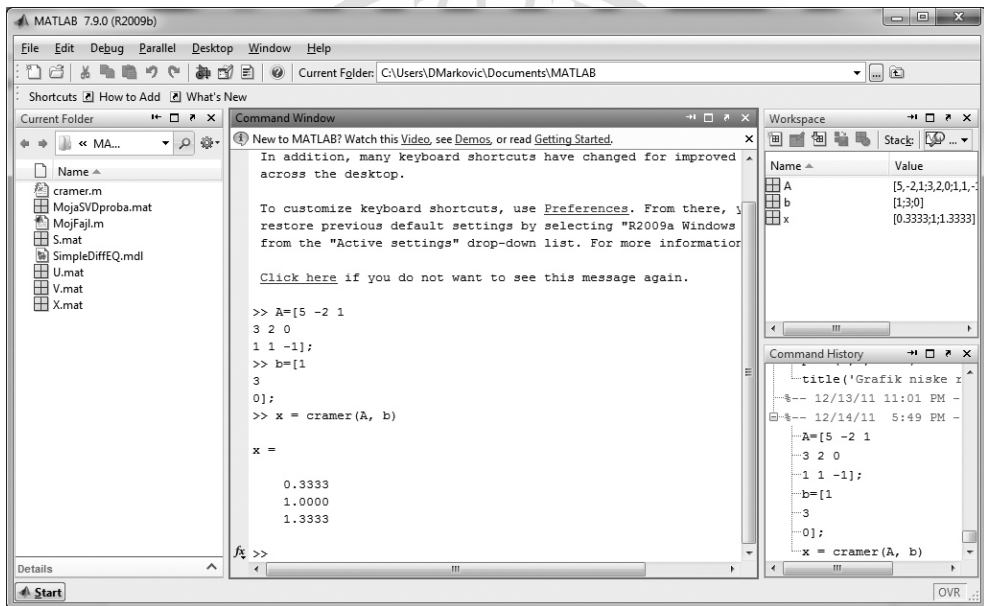
Namenska funkcija (fajl `cramer.m`) napravljena za rešavanje sistema linearnih jednačina ovim pravilom ima sledeći sadržaj:

```
function x = cramer(A, b)
% cramer Solve the system Ax=b.
% The matrix A is square and invertible.
%
% x = cramer(A, b) solves the square system Ax = b.
[m, n] = size(A);
```

```

if m ~= n
    error('Matrix is not square.')
end
if det(A) == 0
    error('Matrix is singular.')
end
for j = 1:n
    B = A;
    B(:, j) = b;
    x(j) = det(B) / det(A);
end
x = x';

```



6. MATEMATIČKO MODELOVANJE OBJEKATA I PROCESA

Kao suštinski reprezent unutrašnjih važnih osobina objekta, koristi se **jednačina ponašanja objekta**. Postupak iznalaženja tih osobina u vidu matematičkih relacija, operacija i simbola, označava se kao **matematičko modelovanje** (modeliranje). Pri tome, same te matematičke relacije, operacije i simboli koji omogućavaju postupnu spoznaju dinamičkih osobina objekata i njegovo dinamičko ponašanje za proizvoljne promene ulaznih veličina i proizvoljne početne i granične uslove predstavlja matematički model objekta ili procesa.

Predmet modelovanja (i simulacije) mogu biti:

- ◆ Proizvodni pogoni;
- ◆ Banke, pošte, samoposluge, ...;
- ◆ Distributivne mreže – transport materijala;
- ◆ Distribucija vode, struje, gasa;
- ◆ Službe za hitne intervencije;
- ◆ Računarski sistemi;
- ◆ Saobraćajni sistemi (raskrsnice, luke, ...);
- ◆ Fabrike;
- ◆ Restorani „brze hrane”;
- ◆

Imajući u vidu kompleksnost i raznolikost objekata modelovanja, najčešće bi njihovo potpuno opisivanje zahtevalo veoma složen matematički model objekta izražen nelinearnim parcijalnim diferencijalnim jednačinama. Iz tog razloga se gotovo uvek pristupa delimičnoj ili potpunoj idealizaciji sistema u cilju modelovanja.

Stepen idealizacije procesa zavisi od broja i karaktera usvojenih pretpostavki. Sa ciljem da se dobijeni matematički model što bolje iskoristi u praksi, neophodno je uvek analizirati neophodnost učinjenih pretpostavki.

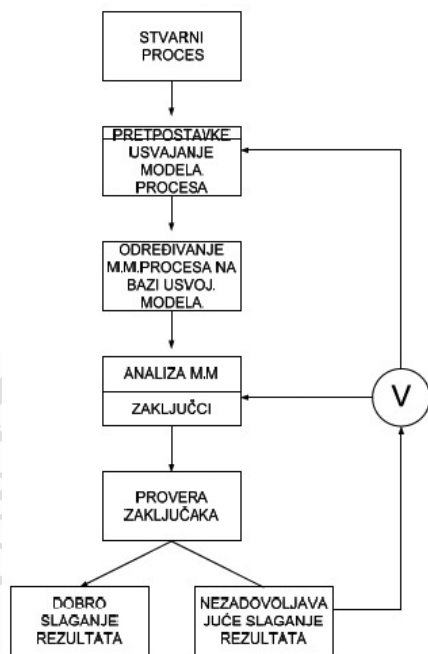
Na taj način, svesni činjenice da matematički model treba što bolje da odražava realni proces a i saznanje do kojih granica seže savremeni matematički aparat, postavlja se pitanje u kojoj meri treba idealizovati realni proces tako da ne budu zapostavljene njegove primarne osobine a da sa druge strane matematički model ne bude suviše kompleksan u smislu dalje matematičke analize.

Konačna ocena pouzdanosti matematičkog modela može se samo približno izreći i to na osnovu detaljne analize svih usvojenih pretpostavki i rezultata kojima ishoduje razmatrani model. U tom smislu, prava provera vrednosti matematičkog modela može se u praktičnim uslovima sprovesti na dva načina:

Prvi način (M1) se zasniva na pretpostavci da je matematički model objekta predstavljen u vidu sistema diferencijalnih jednačina. Rešenja tog sistema jednačina se mogu dobiti njihovom simulacijom na računaru, uvođenjem standardnih ulaznih signala. Potom se sada realni sistem (objekat upravljanja) pobuđuje istim ulaznim veličinama i meri se izlaz sistema (odziv). Poređenjem odziva sistema diferencijalnih jednačina koje predstavljaju

model objekta i stvarnog odziva objekta, donose se zaključci o validnosti formiranog modela.

Na slici 6.1 je prikazana principijalna šema postupka određivanja i verifikacije matematičkog modela objekta ili procesa.



Sl.6.1 Algoritam postupka verifikacije matematičkog modela

U drugom prilazu (M2), polazeći od eksperimentalno snimljenih funkcionalnih zavisnosti ponašanja realnog objekta u nestacionarnom režimu, moguće je korišćenjem različitih vidova identifikacije odrediti matematički model objekta. Poređenjem ovoga i analitički izvedenog matematičkog modela mogu se izvući kvalitativni i kvantitativni zaključci. O ovoj tehnici, koja se prvenstvo zasniva na linearnoj i nelinearnoj statistici, čitalac može više saznati iz odgovarajuće stručne literature. Prvi način formiranja modela preko sistema diferencijalnih jednačina biće opisan kasnije.

Stepeni vrednosti (valjanosti) modela, dobijeni procenom modela su:

- ♦ Replikativna valjanost (najniži stepen)
 - Uporedivi su izlazi modela i sistema
- ♦ Prediktivna valjanost
 - Model proizvodi dobre vrednosti na izlazima per nego što su stvarno izmerene u realnom sistemu (predviđanja modela)
 - Omogućava istraživanje situacija koje nisu posmatrane u sistemu

- ♦ Strukturna valjanost
 - Model u potpunosti odslikava **način** na koji realan sistem funkcioniše
 - Omogućava istraživanje operacija sistema koje se ne mogu meriti.

Kod predstavljanja modela sistemom diferencijalnih jednačina (M1) struktura modela proizilazi direktno na bazi poznatih teorijskih saznanja i naučnih zakonitosti koje važe za dati sistem, dok se kod postupka zasnovanom na statističkoj analizi (M2) struktura modela mora pretpostaviti unapred.

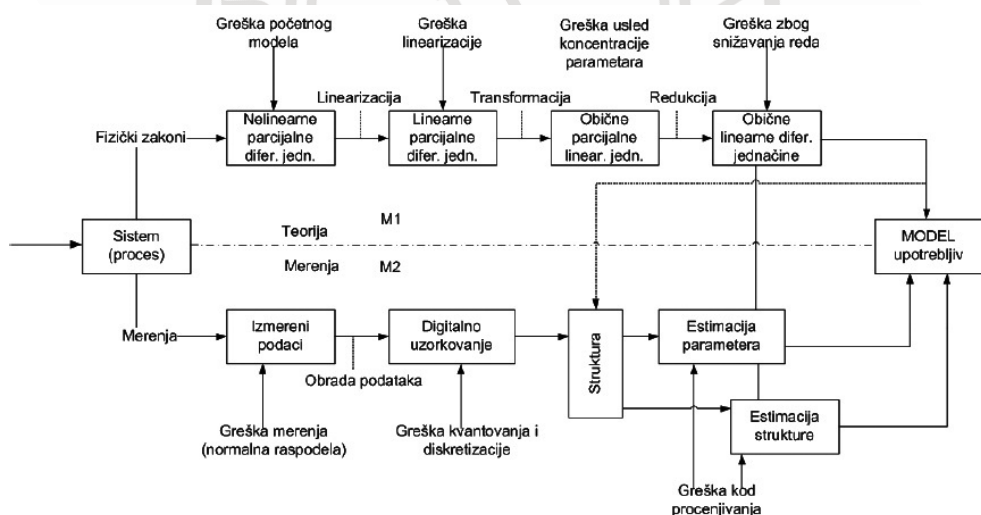
Kod M1 se reprodukuju relacije između ulaza, internih promenljivih procesa i izlaza, dok se kod M2 uglavnom dobija model tipa crne kutije, tj. pomoću M2 se samo reprodukuju relacije između ulaza i izlaza.

Kod M1 su parametri modela povezani sa stvarnim parametrima originala, dok su kod M2 to najčešće samo neke brojne vrednosti i ne moraju biti, određeni fizički parametri realnog procesa.

Model dobijen pomoću M1 vredi i za srodne procese i razne režime rada, dok se sa M2 model odnosi jedino na određeni proces na kojem se vrše merenja. Iz tog razloga se putem M2 ne mogu analizirati sva stanja, kao što su npr. havarijska stanja procesa.

Ipak, programska rešenja formirana za postupak M2 se mogu primenjivati za dobijanja modela različitih procesa, dok se za M1 za svaki proces modelovanje mora vršiti iz početka.

Na slici 6.2 je prikazan odnos M1 i M2 procedura kod formiranja modela procesa.



Sl. 6.2 Odnos pristupa modelovanju na osnovu teorijskih saznanja (M1) i rezultata eksperimentalnog merenja (M2) u sistemu.

Diferencijalne jednačine dinamičkih elemenata

U dinamičkim sistemima se analizira preraspodela energije, materije i informacija u vremenu, zbog toga je vreme nezavisna promenljiva. Promena, materije, energije i informacija se ispoljava kroz promene odgovarajućih veličina u sistemu. Iz tog razloga, za analizu sistema, su od interesa veze između izvoda promenljivih u funkciji vremena (izvoda po vremenu). Ove zavisnosti se mogu izraziti diferencijalnim jednačinama. Kako je već rečeno, ovo je osnov modelovanja sistema prema teorijskim zakonitostima koje ga karakterišu (M1).

Modelovanje bilo kog dinamičkog elementa, prema metodologiji M1, se svodi na postavljanje diferencijalne jednačine koja opisuje njegovo ponašanje. Ukoliko se radi o elementu sa jednim ulazom i jednim izlazom tada odgovarajuća diferencijalna jednačina treba da daje vezu između izvoda ovih dveju veličina. Veza njihovih izvoda se može zapisati diferencijalnom jednačinom višeg reda. Ovo su modeli tipa ulaz-izlaz i jedan od oblika je:

$$F\left(\frac{d^n x_i(t)}{dt^n}, \dots, x_i(t); \frac{d^m x_u(t)}{dt^m}, \dots, x_u(t)\right) = 0 \quad (6.1)$$

Gde je F – nelinearna funkcija više promenljivih. Promenljiva $x_u(t)$ je ulaz elementa, a $x_i(t)$ njegov izlaz.

Kada na element deluje neka ulazna veličina kažemo da se radi o neautonomnom elementu koji se opisuje nehomogenom diferencijalnom jednačinom. Ukoliko diferencijalna jednačina odgovara interakciji u stvarnom elementu tada će njeno rešenje odgovarati ponašanju modelovanog elementa. Prema tome, rešavanje diferencijalne jednačine odgovara analizi ponašanja modelovanog dinamičkog elementa.

Kako ne postoji opšte rešenje za rešavanje raznih tipova diferencijalnih jednačina od posebnog je značaja da se elementi (kad god je to moguće) predstavljaju linearnim diferencijalnim jednačinama sa konstantnim koeficijentima. Ovo naravno uvodi izvesnu grešku aproksimacije u analizi sistema, jer je poznato da se realni sistemi ne ponašaju tako da ih linearna aproksimacija može adekvatno opisati.

Linearizacija modela datog nelinearnom jednačinom (6.1) se svodi na to da levu stranu jednačine razvijemo u Taylorov red u okolini nekog ravnotežnog stanja (x_{u0}, x_{i0}). Pretpostavimo da se određeno ravnotežno stanje sistema uspostavlja pri datoj konstantnoj vrednosti ulaza $x_u(t) = x_{u0}$ i da je izlaz sistema u tom stanju jednak određenoj konstantnoj vrednosti $x_i(t) = x_{i0}$. Prema tome u ravnotežnom stanju, izraz (6.1), postaje:

$$F(0, \dots, x_{i0}; 0, \dots, x_{u0}) = 0 \quad (6.2)$$

Usvojimo da su i dalje od interesa promene ulaza $\Delta(x_u(t)) = x_u(t) - x_{u0}$ i izlaza $\Delta(x_i(t)) = x_i(t) - x_{i0}$ u okolini ravnotežnog stanja (x_{u0}, x_{i0}). Zbog jednostavnijeg prikaza ovakvih

jednačina najčešće se Δ uz sve promenljive izostavlja, ali se uvek ima u vidu da se radi o promenama (odstupanjima) ovih promenljivih u odnosu na njihove vrednosti u razmatranom ravnotežnom stanju. Razvoj jednačine (6.1) u Tajlorov red se može predstaviti kao:

$$\begin{aligned} a_n \frac{d^n x_i(t)}{dt^n} + a_{n-1} \frac{d^{n-1} x_i(t)}{dt^{n-1}} + \dots + a_0 x_i(t) = \\ = b_m \frac{d^m x_u(t)}{dt^m} + b_{m-1} \frac{d^{m-1} x_u(t)}{dt^{m-1}} + \dots + b_0 x_u(t) \end{aligned} \quad (6.3)$$

Gde su koeficijenti a_k ($k = 0, \dots, n-1$); b_i ($i = 0, \dots, m$) određeni vrednostima parcijalnih izvoda funkcije F u ravnotežnoj tački prema:

$$a_0 = \left(\frac{\partial F}{\partial x_i} \right)_0, a_1 = \left(\frac{\partial F}{\partial \dot{x}_i} \right)_0, \dots, b_0 = \left(\frac{\partial F}{\partial x_u} \right)_0, b_1 = \left(\frac{\partial F}{\partial \dot{x}_u} \right)_0$$

Prema tome, jednačina (6.3), se može prikazati jednostavnije kao:

$$\begin{aligned} a_n x_i^{(n)}(t) + a_{n-1} x_i^{(n-1)}(t) + \dots + a_2 \ddot{x}_i(t) + a_1 \dot{x}_i(t) + a_0 x_i(t) = \\ = b_m x_u(t) + b_{m-1} \dot{x}_u(t) + \dots + b_2 \ddot{x}_u(t) + b_1 \dot{x}_u(t) + b_0 x_u(t) \end{aligned} \quad (6.4)$$

Iz navedenog proizilazi da je **prvi korak u analizi dinamičkog elementa formiranje odgovarajuće diferencijalne jednačine koja opisuje njegovo ponašanje**. Ovo se postiže primenom odgovarajućih zakonitosti naučne discipline iz koje se posmatrani sistem razmatra.

Kako je već rečeno, sledeći korak se zasniva na **linearizaciji** ovako definisane diferencijalne jednačine. Na bazi aproksimacije prvog reda za funkciju F , koja se dobija izostavljanjem članova višeg reda u Tajlorovom razvoju (6.3), dobija se linearna diferencijalna jednačina u odnosu na navedene promene ulaza i izlaza, koja se koristi za dalju analizu sistema.

Prenosna funkcija sistema

Prenosna funkcija sistema je jedan od osnovnih pojmova vezanih za dinamičke sisteme, kakav je sistem prikazan jednačinom (6.4). Prenosna funkcija sistema omogućava dinamičku analizu sistema u kompleksnom domenu. To automatski znači da jednačina (6.4) sa vremenskog treba da se transformiše u **kompleksni domen**. To se radi **primenom Laplasove transformacije** na linearizovanu funkciju sistema. Obzirom da prenosna funkcija omogućuje dinamičku analizu sistema u kompleksnom domenu, na njoj se zasnivaju mnoga efikasna rešenja raznih problema pri proučavanju sistema.

Posle primene Laplasove transformacije na diferencijalnu jednačinu ulazno-izlaznog linearizovanog modela sistema n -tog reda (jednačina 6.4), dobija se:

$$a_n s^n X_i(s) + a_{n-1} s^{n-1} X_i(s) + \dots + a_2 s^2 X_i(s) + a_1 s X_i(s) + a_0 X_i(s) = \\ = b_m s^m X_u(s) + b_{m-1} s^{m-1} X_u(s) + \dots + b_2 s^2 X_u(s) + b_1 s X_u(s) + b_0 X_u(s) + f(s, x_{u0}) \quad (6.5)$$

Gde $f(s, x_{u0})$ zavisi od početnih uslova. Kada su početni uslovi jednaki nuli, tada je:

$$X_i(s) = W(s) \cdot X_u(s) \quad (6.6)$$

Gde je

$$W(s) = \frac{B(s)}{A(s)} \quad (6.7)$$

Veličine $B(s)$ i $A(s)$ u jednačini (6.7), predstavljaju:

$$A(s) = a_n s^n + a_{n-1} s^{n-1} + \dots + a_0$$

$$B(s) = b_m s^m + b_{m-1} s^{m-1} + \dots + b_0$$

Veličina $W(s)$ predstavlja **prenosnu funkciju**, i na osnovu jednačine (6.6), izražava se kao:

$$W(s) = \frac{X_i(s)}{X_u(s)}$$

Odakle je očigledno da ona predstavlja **količnik Laplasovih transformacija ulaza i izlaza elemenata pri nultim početnim uslovima**.

Prenosna funkcija je od posebnog značaja za analizu linearnih sistema, jer se za poznato $W(s)$ i $X_u(s)$ direktno može odrediti $X_i(s)$.

Napomena: U prethodnom razmatranju pretpostavljeno je da odziv na ulazu $x_i(t)$ nekog sistema (elementa) zavisi samo od jednog ulaza $x_u(t)$. Nakon razvoja diferencijalne jednačine modela bilo je moguće odrediti njegovu funkciju prenosa kako je već opisano. Međutim, mnogo je češći slučaj da na izlaz sistema utiče veći broj spoljašnjih ulaza (delovanja), *sistemi sa više ulaza*. Naprimera, ukoliko se radi o objektu upravljanja tada pored manipulativne promenljive postoje i različiti poremećaji. Takođe je tipična i situacija da neki poremećaji imaju značajan uticaj na ponašanje sistema, dok se delovanje ostalih poremećajnih veličina u ovom pogledu može zanemariti.

Impulsni i odskočni odziv linearnih elemenata sistema

Kao što je već rečeno, testiranje valjanosti (validacije) bilo kojeg matematičkog modela sistema (objekta ili procesa) vrši se dovođenjem standardnih signala na ulaz sistema i merenjem odziva sistema, koji se potom upoređuje sa odzivom modela sistema kojem je na ulazu doveden isti signal. Iz tog razloga su od posebnog značaja za analizu linearnih kontinualnih elemenata/sistema, njihovi odzivi na ulaze oblika Dirakove (delta) funkcije

vremena. Kako je već rečenom, ova funkcija se može dobiti u graničnom prelazu pravougaonog impulsa (površine jednake jedinici) kada mu trajanje teži nuli. Takođe je pokazano da je Laplasova transformacija Dirakovog impulsa jednaka jedinici, te je iz tog razloga odziv elementa/sistema sa funkcijom prenosa $W(s)$ i pri nultim početnim uslovima jednostavno:

$$X_i(s) = W(s) X_u(s) = W(s) \quad (6.8)$$

Prema tome, funkcija prenosa kontinualnog elementa/sistema je jednaka Laplasovoj transformaciji njegovog impulsnog odziva pri nultim početnim uslovima.

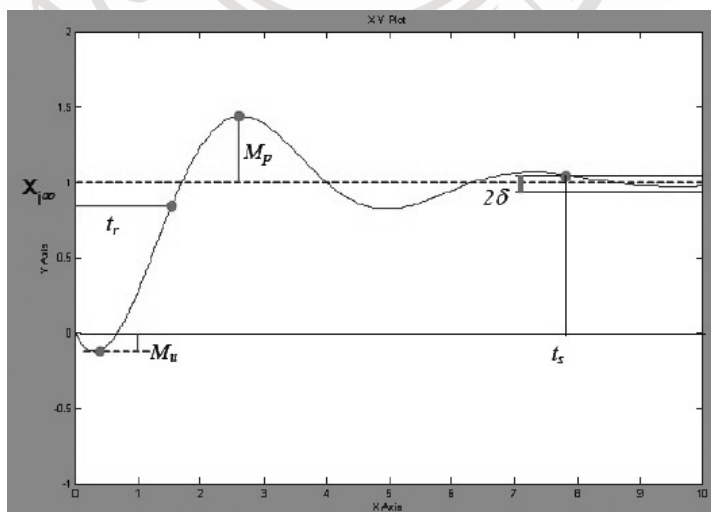
Pored Dirakove funkcije, takođe se posmatraju odzivi elementa/sistema na ulaze odskočne funkcije vremena. Kako je već rečeno, Laplasova transformacija jedinične odskočne funkcije vremena $X_u(s) = 1/s$. Odatle je kompleksni lik odskočnog odziva dat sa:

$$X_i(s) = W(s) X_u(s) = W(s) \frac{1}{s} \quad (6.9)$$

Koristeći teoremu o graničnoj vrednosti, za vrednost odskočnog odziva u stacionarnom stanju (ukoliko postoji) pri jediničnom odskočnom ulazu dobijamo:

$$\lim_{t \rightarrow \infty} x_i(t) = x_{i\infty} = \lim_{s \rightarrow 0} s W(s) \frac{1}{s} = W(0) \quad (6.10)$$

Prema jednačini (6.10), **ukoliko je sistem stabilan** tada će prelazne komponente odskočnog odziva eksponencijalno iščezavati ka nuli i postojaće njegova granična vrednost u stacionarnom stanju. Da bi se definisali relevantni parametri koji karakterišu dinamiku nekog elementa/sistema biće razmatran jedan odskočni odziv prikazan na slici 6.3.



Sl. 6.3 Karakteristike odskočnog odziva nekog sistema

Vreme porasta t_r je ono vreme koje protekne dok odskočni odziv prvi put dosegne vrednost $k_r x_{i\infty}$, pri čemu konstanta k_r varira između 0.9 i 1.

Preskok (engl. *overshoot*) M_p je maksimalna trenutna vrednost za koju odskočni odziv prelazi svoju konačnu vrednost.

Podbačaj (engl. *undershoot*) M_u je maksimalna (apsolutna) vrednost za koju odskočni odziv pada ispod nule (za koju se odskočni odziv menja u suprotnu stranu od $x_{i\infty}$).

Vreme smirenja t_s je vreme potrebno da odskočni odziv uđe i ostane u granicama $\pm\delta$ oko svoje konačne vrednosti. Ova devijacija (δ) se često izražava u procentima od konačne vrednosti (na primer od 1 do 5%).

Polovi, nule i vremenski odzivi

Polovi

Kako je već poznato, svaka racionalna funkcija se može razviti u parcijalne razlomke od kojih svaki član odgovara jednostrukim ili višestrukim, realnim ili parovima konjugovano kompleksnih polova. U tom smislu je korisno razmotriti efekte pojedinih polova na prelazne procese.

Efekat npr. jednostrukog realnog pola se može sagledati posmatrajući odziv elementa/sistema sa funkcijom prenosa tipa:

$$W_1(s) = \frac{K}{Ts+1} \quad (6.11)$$

Gde su K i T realne i pozitivne konstante.

Odziv elementa na ulaz oblika jedinične odskočne funkcije vremena je dat kao:

$$x_i(t) = L^{-1} \left\{ \frac{K}{Ts+1} \cdot \frac{1}{s} \right\} \quad (6.12)$$

Kako je već ranije opisano, ovakav oblik jednačine se rešava tako što se formira izraz:

$$\frac{K}{Ts+1} \cdot \frac{1}{s} = \frac{K_1}{s} + \frac{K_2}{Ts+1} = \frac{K_1(Ts+1) + K_2s}{(Ts+1)s}$$

Odatle se formira sistem jednačina:

$$K_1 = K$$

$$K_1T + K_2 = 0$$

Odatle je :

$$K_1 = K$$

$$K_2 = -KT$$

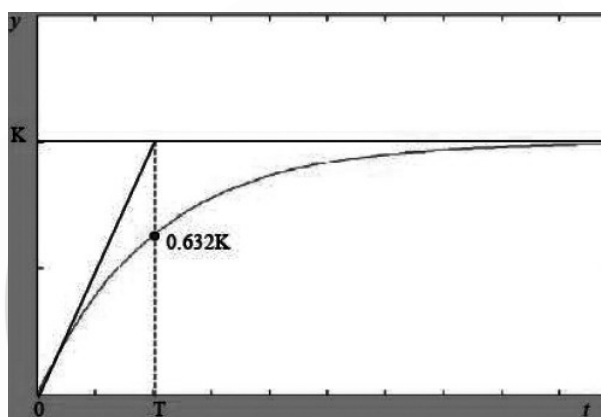
Te sada jednačina (6.12), postaje:

$$x_i(t) = L^{-1} \left\{ \frac{K}{s} - \frac{KT}{Ts+1} \right\} = K \left(1 - e^{-\frac{t}{T}} \right), t \geq 0 \quad (6.13)$$

Vrednost odskočnog odziva u stacionarnom stanju je:

$$x_{i\infty} = W(0) = \lim_{s \rightarrow 0} \frac{K}{Ts+1} = K$$

Tako da K predstavlja odnos ulaza i izlaza u stacionarnom stanju. T je vremenska konstanta elementa i određuje brzinu njegovog odziva. Elementi sa većom vremenskom konstantom T imaju sporiju reakciju. Na slici 6.4 je prikazana skica odziva razmatrane funkcije. Sa slike se vidi da se odziv može skicirati direktno na osnovu poznavanja parametara funkcije K i T .



Sl. 6.4. Prikaz odziva elementa prvog reda

Nule

Kao što se na osnovu prethodnog razmatranja može zaključiti, polovi određuju pojedine modove (oblik komponenti odziva) nezavisno jednih od drugih. Za razliku od toga, nule funkcije prenosa predstavljaju neku interakciju više polova (njihov kombinovani uticaj) na odziv sistema.

Za ilustraciju uticaja nula na odziv sistema biće uzet primer sistema sa jednom konačnom nulom ($s = -c$) i polovima (-1 i -2). Prenosna funkcija ovog sistema je:

$$W_2(s) = \frac{2(s+c)}{(s+1)(s+2)} \quad (6.14)$$

Uticaj pola (-1) na odziv ovog sistema će biti zanemarljiv ukoliko je nula sistema (c) bliska po vrednosti (-1), jer tada izraz (6.14) asimptotski teži nuli. Situacija je analogna i sa drugim polom (-2). U svakom slučaju, ukoliko je nula bliska nekom od polova elementa/sistema oni čine dipol i njihov uticaj na dinamiku elementa se može zanemariti.

Primer: Odrediti odskočni odziv elementa opisanog diferencijalnom jednačinom:

$$T \cdot \frac{dx_i}{dt} + x_i = kT \frac{dx_u}{dt}$$

Gde je $T = 30 \text{ sec}$ i $k = 5$.

Rešenje:

Gornju diferencijalnu jednačinu transformišemo u:

$$T \cdot \dot{x}_i(t) + x_i(t) = kT \dot{x}_u(t)$$

Odnosno

$$T \cdot s x_i(s) + x_i(s) = kT s x_u(s)$$

Funkcija prenosa sistema je:

$$W(s) = \frac{x_i(s)}{x_u(s)} = \frac{kTs}{1+Ts}$$

Odatle je

$$x_i(s) = \frac{kTs}{1+Ts} x_u(s)$$

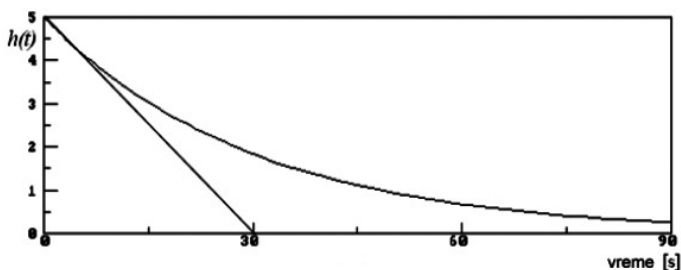
Za jedinični odskočni ulaz je $x_u(s) = 1/s$

Te je:

$$x_i(s) = \frac{kTs}{1+Ts} \cdot \frac{1}{s} = \frac{kT}{1+Ts}$$

Odatle je odziv funkcije:

$$X_i(t) = L^{-1} \left\{ \frac{KT}{Ts+1} \right\} = L^{-1} \left\{ \frac{5 \cdot 30}{30 \cdot s + 1} \right\} = L^{-1} \left\{ \frac{5}{s + 1/30} \right\} = 5 \cdot e^{-t/30}$$



Sl. 6.5 Odziv razmatranog sistema

7. UVOD U SIMULINK

Programski paket Matlab sadrži niz specijalizovanih softverskih modula (engl. Toolboxes). Jedan od njih je Simulink - softverski modul za modeliranje/modelovanje, simulaciju i analizu dinamičkih sistema. Podržava linearne i nelinearne sisteme, modele u kontinualnom vremenu, diskretnom vremenu odabiranja/odmeravanja ili bilo koji hibrid od ova dva. Takođe, sistemi mogu biti višebrzinski, tj. imati delove koji se odmeravaju ili ažuriraju sa različitim brzinama.

Za modeliranje, tj. gradnju modela kao blok dijagrama, Simulink obezbeđuje grafički korisnički interfejs (GUI) u okviru koga se operacijama miša tipa prevuci-i-pusti obavlja gradnja modela.

Pomoću ovog interfejsa mogu se crtati modeli kao što bi to radili sa olovkom i papirom. Simulink sadrži bogatu biblioteku potrošača (engl. *sink*), izvora (engl. *source*), linearnih i nelinearnih komponenti i konektora. Pored toga, korisnik može da prilagodi postojeće i kreira sopstvene blokove.

Modeli su hijerarhijski, tako da korisnik može da gradi modele koristeći *top-down* ili *bottom-up* pristup. Korisnik može da posmatra sistem na visokom nivou, zatim da dvaput klikne na blok da bi silazio unutra kroz nivoe i video rastući nivo detalja modela. Ovaj pristup omogućuje korisniku da sagleda strukturu modela i vidi kako je organizovan, i kako njegovi pojedini delovi interaguju.

Nakon što je model definisan, može se simulirati, koristeći niz integracionih metoda, bilo sa menija Simulinka, ili unošenjem komandi u komandni prozor MATLAB-a. Meniji su naročito pogodni za interaktivni rad, dok pristup preko komandne linije je pogodan za slučajeve kada se izvršava više simulacija, (napr. ako radimo Monte Carlo simulacije, ili želimo da predemo kroz ukupan opseg vrednosti parametara). Koristeći osciloskope (engl. *scopes*) i druge blokove za prikazivanje podataka, korisnik može videti rezultate simulacije dok se simulacija još izvršava. Nadalje, korisnik može promeniti parametar i trenutno videti šta se dešava, za istraživanja tipa "šta ako" (*what if*).

Rezultati simulacije se mogu staviti u radni prostor MATLAB-a za dalje postprocesiranje i vizuelizaciju.

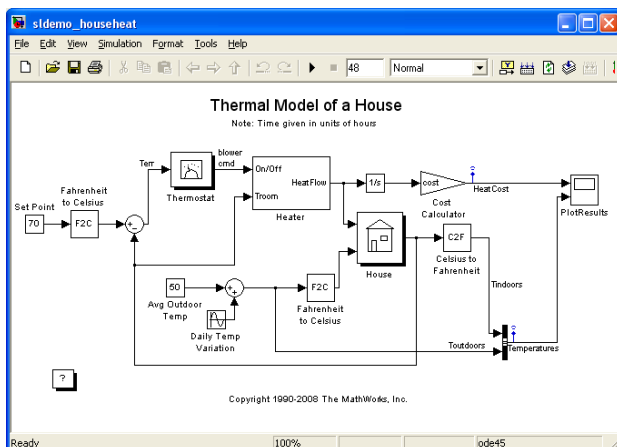
Alati za analizu modela uključuju linearizaciju i alate za pojednostavljenje modela, kojima se može pristupiti sa komandne linije MATLAB-a.

Demonstracioni model

U okviru modula nalazi se demonstracioni termodinamički model kuće. Model se startuje ukucavanjem niže navedenih naredbi u komandnom prozoru MATLAB-a .

```
mdl='sldemo_househeat';  
open_system mdl;
```

Ove komande će pokrenuti Simulink i otvoriti prozor kao na sledećoj slici:



Da bi se pokrenula simulacija treba otvoriti meni **Simulation** i kliknuti na **Start** ili kliknuti na mali trougao usmeren udesno na paleti alatki.

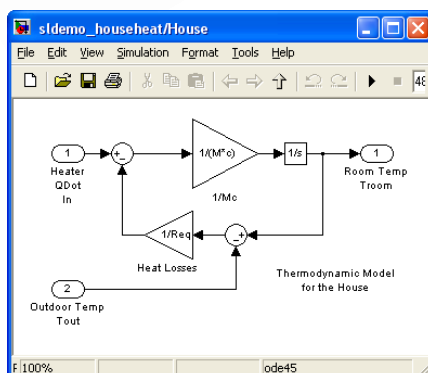
Za zaustavljanje simulacije dovoljno je kliknuti na **Simulation/Stop** ako se već nije ranije završila sama prema vremenu trajanja simulacije u konfiguracionom polju **Simulation/Configuration parameters/Stop Time**.

Opis modela

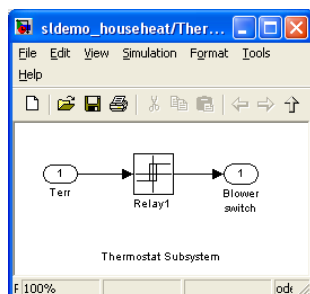
Model koristi podsisteme (engl. *subsystems*) da pojednostavi dijagram modela i kreira module koji se mogu više puta koristiti. Podsystem je grupa blokova koji su predstavljeni blokom podsistema. Ovaj model sadrži pet podsistema:

- ◆ *thermostat*
- ◆ *house*
- ◆ *temp convert F/C (2)*
- ◆ *temp convert C/F (1)*

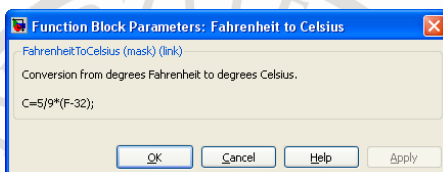
Da bi videli unutrašnju strukturu podsistema, treba kliknuti dva puta na njegov blok. Tako, na primer, ako kliknemo dvaput na blok "House", otvoriće se blok struktura tog podsistema kao na slici:



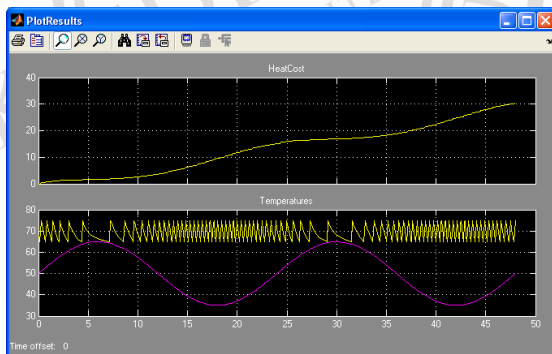
Podsistem termostata modelira rad termostata, određujući kada će se grejanje uključiti ili isključiti. Ako dvaput kliknemo na taj blok, otvoriće sa njegov podsistem kao na slici:



Spoljašnja i unutrašnja temperatura su pretvarane iz Farenhajtovih stepeni u Celzijusove i obratno od strane identičnih podsistema (blok funkcionalnih parametara) kao na slici:



Kada je grejanje uključeno, troškovi grejanja se računaju i iscrtavaju na dijagramu/plotu "Heat cost (\$)".



Ovaj demo ilustruje nekoliko taskova/poslova koji su zajednički svim modelima:

- ♦ Izvršavanje simulacije se startuje sa komandom Start, i zaustavlja sa komandom Stop, koje su dostupne iz menija Simulation ili direktno na paleti alatki.
- ♦ Pri gradnji modela možemo enkapsulirati grupu povezanih blokova u jedan blok, koji se naziva podsistem.
- ♦ Možemo kreirati prilagođenu ikonu i dizajnirati okvire za dijalog za svaki blok, koristeći osobinu maske (engl. *masking feature*),
- ♦ Scope blokovi prikazuju grafičke izlaze vrlo slično kako to čine i osciloskopi.

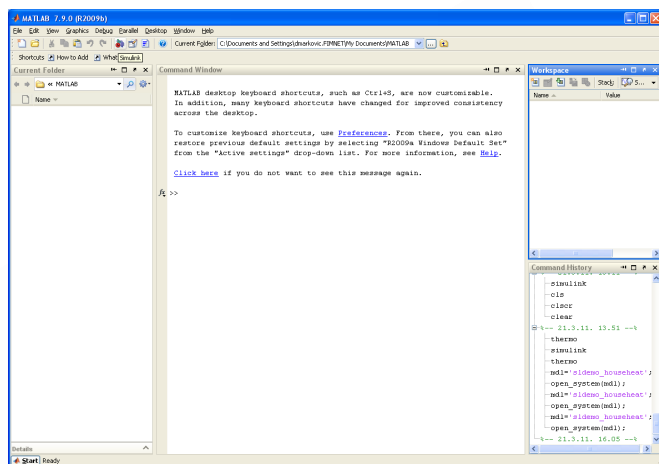
Gradnja jednostavnog modela

Na sledećem primeru ćemo pokazati kako se gradi jednostavni model koristeći uobičajene komande i akcije koje se koriste kod gradnje modela. Model se gradi/crta tako što se iz Simulink biblioteke elementi prevlače mišem u prozor modela i potom povezuju linijama.

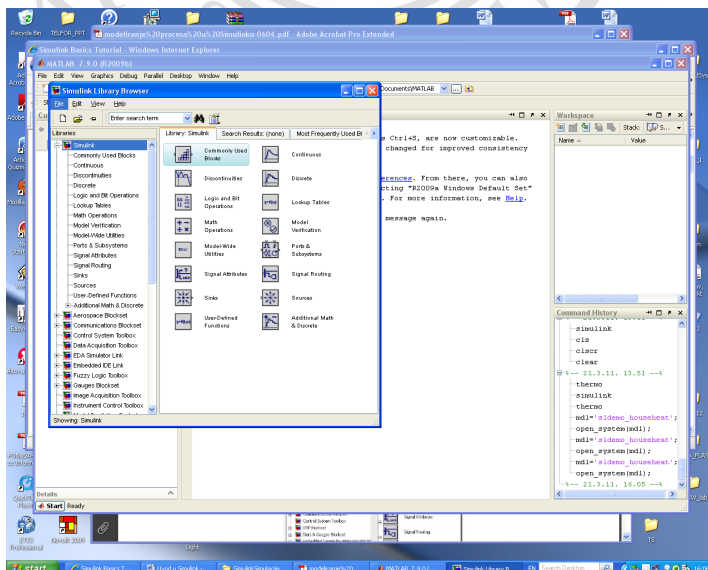
Simulink se startuje iz MATLAB komandnog prozora kucanjem sledeće komande:

```
simulink
```

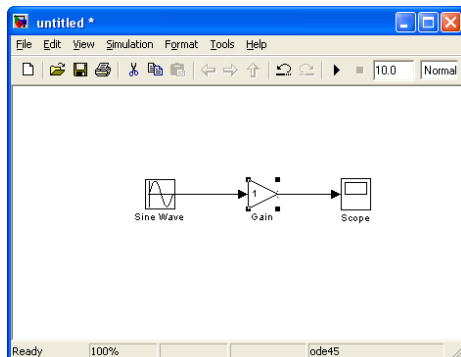
Alternativno, možete mišem kliknuti na dugme **Simulink Library Browser** na paleti MATLAB-a:



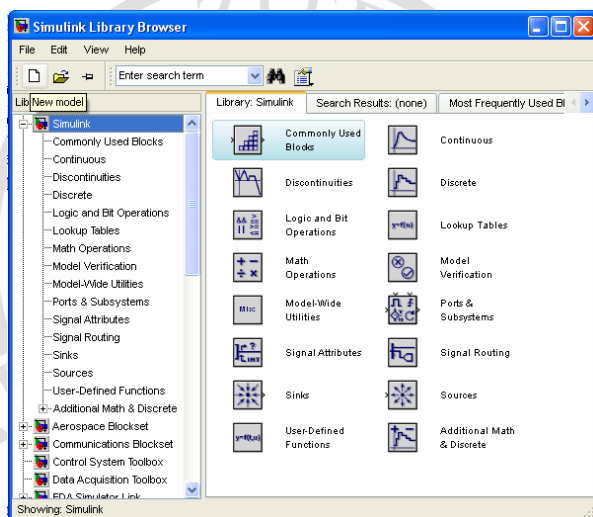
Otvara se prozor Simulink biblioteke.



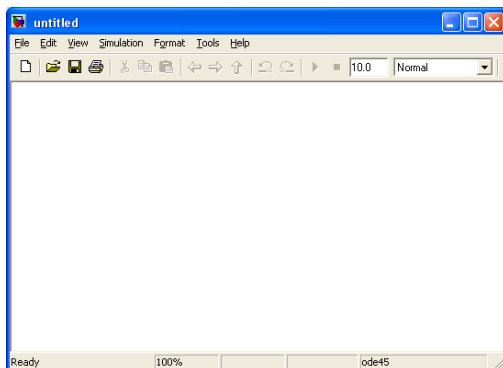
Model uključuje blok generisanja sinusnog talasnog oblika i prikazuje taj oblik. Blok dijagram modela izgleda kao na sledećoj slici:



Da se kreira novi model treba izabrati ili kliknuti na dugme **New Model** na paleti alatki.



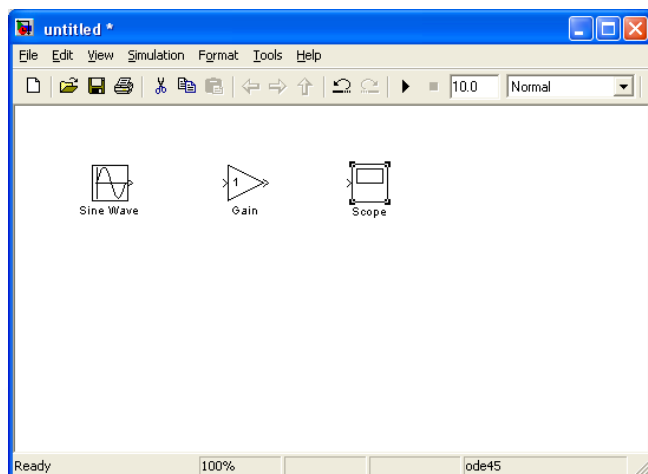
Simulink će otvoriti novi prozor:



Da bi kreirali zadati model, treba da kopiramo blokove iz biblioteka Simulinka:

- ♦ iz bibliote **Sources** (izvori) – blok **Sine Wave**
- ♦ iz bibliote **Math Operations** – blok **Gain**
- ♦ iz biblioteke **Sinks** (potrošači signala) – blok **Scope**

Kada su svi blokovi kopirani u prozor modela, izgledaće otprilike kao :

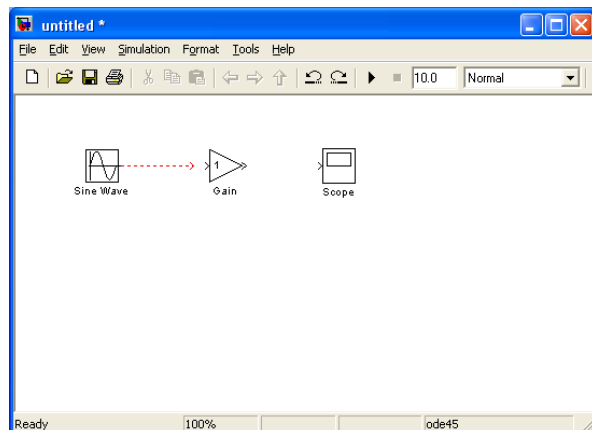


Ako pogledamo pažljivije ikone blokova, videćemo male strelice na izlazu iz sinusnog bloka i bloka pojačanja, kao strelice na ulazu bloka pojačanja i osciloskopa. Simbol > se naziva **izlaznim portom (output port)**, a ako je simbol usmeren prema bloku, onda je to **ulazni port (input port)**. Signal putuje iz izlaznog porta u ulazni port sledećeg bloka kroz liniju koja ih spaja.

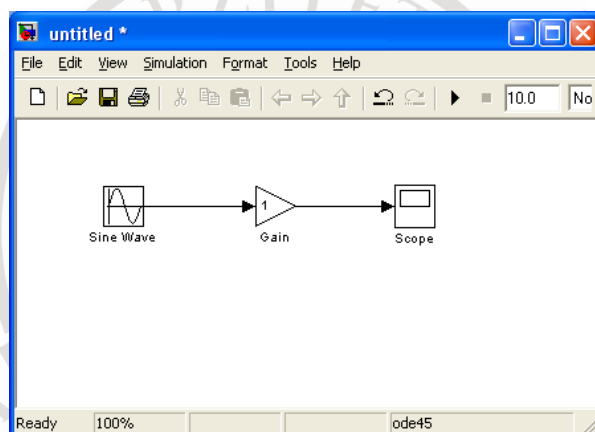
Povezivanje blokova

Da bi blok dijagram precizno odražavao (reflektovao) sistem koji modelujemo, blokovi Simulinka moraju biti na odgovarajući način povezani. U našem primeru, signal izlazi iz bloka Sine Wave, prenosi se u blok Gain koji pojačava taj signal i prosleđuje ga bloku Scope koji ga prikazuje kao funkciju vremena. Prema tome, potrebno je prema navedenoj šemi da linijama povežemo odgovarajuće blokove.

Linije se crtaju povlačenjem miša od mesta iz koga kreće/izvire signal (izlazni port bloka) do mesta gde uvire (ulazni port drugog bloka). Kada se crta linija jako je važno da signal dopre do ciljanog odredišta. Simulink će transformisati pokazivač miša u končanicu kada je dovoljno blizu tačke izlaznog porta da bi počeo sa crtanjem linije, a pokazivač miša će se transformisati u dvostruku končanicu kada je dovoljno blizu odredišne tačke (ulazni port odredišnog bloka). Signal je odgovarajuće povezan kada je strelica na kraju linije popunjena. Ako strelica nije popunjena, to znači da signal nije povezan sa oba bloka. Da biste otklonili problem, treba otvorenu strelicu posmatrati kao izlazni port i nastaviti sa crtanjem linije od te pozicije do odredišnog bloka.



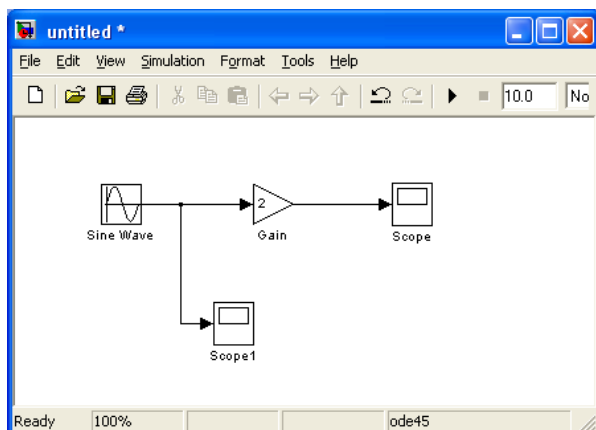
Nekorektno povezan signal.



Korektna veza.

Da ne bi morali da brinete o ovome primenite sledeću proceduru: selektujte mišem blok iz kog polazi signal, zatim pritisnite taster Ctrl na tastaturi i na kraju selektujte mišem odredišni blok i Simulink će automatski odraditi korektno povezivanje.

Kod nekih modela biće potrebno da se uradi grananje signala tako da može da se prenese ka dva ili više različitih ulaznih terminala. To se može izvesti tako što ćete postaviti pokazivač miša na mesto gde se signal račva. Zatim, koristeći ili taster Ctrl zajedno sa levim tasterom miša ili samo desni taster miša, prevući novu liniju na željeno odredište. Taj metod je korišćen u ovom primeru da se kreira grananje izlaznog signala bloka Sine Wave kao što je prikazano na sledećoj slici:



Usmeravanje linija i lokacija grananja mogu se promeniti njihovim prevlačenjem na novu željenu poziciju. Da biste obrisali nekorektno nacrtanu liniju, jednostavno kliknite mišem na nju da je selektujete i pritisnite taster **Delete**.

Promena parametara bloka

Simulink dozvoljava da korisnik modifikuje blokove u modelu kako bi što preciznije reflektovali karakteristike sistema koji se analizira. Na primer, možemo da izmenimo blok Sine Wave dvo-klikom miša. Usled ove aktivnosti na ekranu će se pojaviti sledeći prozor:

Source Block Parameters: Sine Wave

Sine Wave

Output a sine wave:

$$O(t) = \text{Amp} * \sin(\text{Freq} * t + \text{Phase}) + \text{Bias}$$

Sine type determines the computational technique used. The parameters in the two types are related through:

Samples per period = $2 * \pi / (\text{Frequency} * \text{Sample time})$

Number of offset samples = $\text{Phase} * \text{Samples per period} / (2 * \pi)$

Use the sample-based sine type if numerical problems due to running for large times (e.g. overflow in absolute time) occur.

Parameters

Sine type: Time based

Time (t): Use simulation time

Amplitude: 1

Bias: 0

Frequency (rad/sec): 1

Phase (rad): 0

Sample time: 0

☒ Interpret vector parameters as 1-D

OK Cancel Help

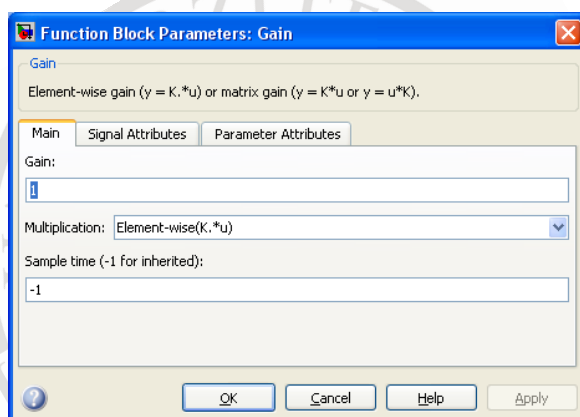
U ovom prozoru može da se podesi amplituda, frekvencija i fazni pomak sinusoide. Vrednost “Sample time” označava vremenski interval između sukcesivnih očitavanja signala. Postavljanjem te vrednosti na 0 postiže se kontinualno očitavanje signala.

Za naš sistem definišimo sledeće vrednosti:

- Amplitude = 2
- Frequency = pi
- Phase = pi/2

Upišimo te vrednosti u odgovarajuća polja (ostavimo vrednost za “Sample time” na 0) i kliknimo mišem na dugme “OK” da ta podešavanja budu prihvaćena i da se zatvori prozor sa parametrima. Vodite računa da su frekvencija i faza za sistem zadati sa vrednošću ‘pi’ (3.1415...). Te vrednosti treba da budu unete u Simulink baš tako kako je pokazano.

U sledećem koraku ćemo modifikovati blok Gain dvo-klikom miša na njega u modelu. Na ekranu se pojavljuje sledeći prozor:



Simulink daje kratko objašnjenje funkcije bloka u gornjem delu ovog prozora. U ovom slučaju (blok Gain), ulazni signal u blok (u) se množi sa konstantom (k) da bi oformio izlazni signal bloka (y). Izmenom parametra “Gain” u ovom prozoru, menja se vrednost konstante k.

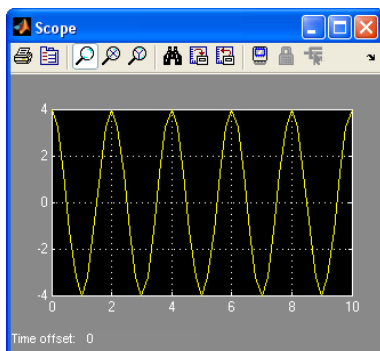
Za naš sistem, stavimo da je $k = 2$. Upišimo tu vrednost u polje “Gain” i kliknimo na dugme “OK” da zatvorimo prozor.

Blok Scope samo iscrtaava svoj ulazni signal kao funkciju vremena i stoga nema sistemskih parametara koje možemo promeniti.

Izvršavanje simulacije

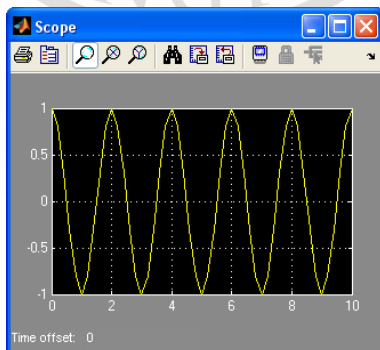
Model je konstruisan i sve je spremno za simulaciju sistema. Da bismo aktivirali izvršavanje simulacije potrebno je da na meniju **Simulation** izaberemo opciju **Start**, ili da kliknemo mišem na dugme “Start/Pause Simulation” na paleti alatki ispod menija (izgleda kao dugme “Play” na videorekorderu). Obzirom da je konstruisan relativno jednostavan model, simulacija se izvršava skoro trenutno. Međutim, kod komplikovanijih sistema moći

ćete da vidite napredovanje simulacije osmatranjem vremena izvršavanja u donjem ddelu prozora modela. Dvo-klikom miša na blok Scope vidi se izlaz bloka Gain kao funkcija vremena. Kada je aktiviran prozor Scope, kliknite na dugme “Autoscale” na njegovoj paleti alatki (izgleda kao dvogled) da skalirate grafik da što bolje zahvati prozor.



Posmatranjem izlaza našeg sistema uočava se da je to kosinusna kriva sa periodom 2 sekunde i amplitudom jednakom 4. Da li se rezultat slaže sa parametrima sistema koje smo postavili? Amplituda ima smisla obzirom da je amplituda ulaznog signala 2 a konstanta bloka pojačanja je 2 ($2 \times 2 = 4$). Perioda izlaza treba da je ista kao i kod ulaznog signala, a ta vrednost je funkcija frekvencije koju smo upisali kod bloka Sine Wave block (postavili da je jednaka π). Konačno, oblik izlaza je kosinusna kriva zato što smo kod ulaznog signala postavili vrednost faze od $\pi/2$ (sinusna i kosinusna kriva su fazno pomerene za $\pi/2$).

Šta će se desiti ako promenimo pojačanje sistema da ima vrednost 0.5? Kako će to uticati na izlaz bloka Gain što može da se posmatra na bloku Scope? Uradimo tu promenu dvo-klikom miša na blok Gain i upišimo za vrednost pojačanja iznos od 0.5. Zatim ponovo pokrenimo simulaciju i pregledajmo blok Scope (grafik bloka Scope neće se promeniti sve dok se ne izvrši simulacija sa novopostavljenim čak i kada je promenjena vrednost pojačanja). Grafik osciloskopa sada ima sledeći izgled:



Uočava se promena samo kod iznosa amplitude kosinusne krive.

Modeliranje jednačina

Jedan od najvećih izazova kod novih korisnika Simulinka je modeliranje jednačina. Poćemo sa jednim jednostavnim primerom a kasnije nastaviti sa složenijim.

Konvertovanje stepeni Celzijusa u Farenhajte

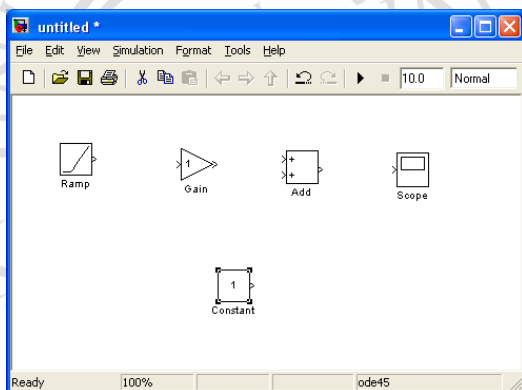
Zadatak je da modelujemo navedenu jednačinu koja pretvara temperaturu izraženu u Celzijusovim stepenima u Farenhajtove stepene:

$$TF = 9/5(TC) + 32$$

Prvo, razmotrimo blokove koji su nam potrebni za gradnju modela:

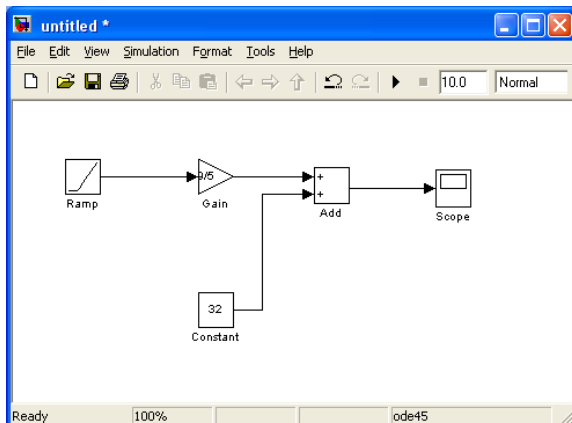
- ♦ Blok **Ramp** iz biblioteke Sources za ulazni signal temperature;
- ♦ Blok **Constant** iz biblioteke Commonly Used Blocks, da definišemo konstantu 32;
- ♦ Blok **Gain** iz biblioteke Math Operations, da pomnožimo ulazni signal sa 9/5;
- ♦ Blok **Add** iz biblioteke Math Operations, da saberemo dve veličine;
- ♦ Blok **Scope** iz biblioteke Sinks, da prikazemo izlaz.

Postavimo sada blokove u prozor našeg modela.



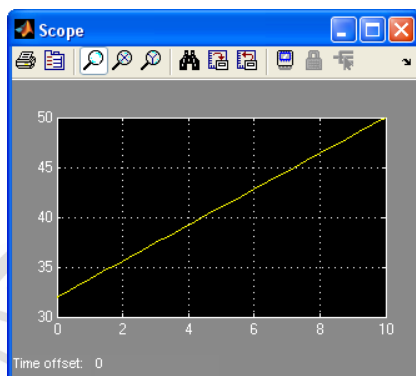
Dodelimo parametarske vrednosti blokovima Gain i Constant dvo-klikom miša na njih i upisivanjem odgovarajućih vrednosti. Zatim, kliknimo mišem na dugme OK da zatvorimo pripadajući okvir za dijalog.

Na kraju, povežimo blokove.



U blok Ramp unosi se temperatura u stepenima Celzijusa. Otvorimo taj blok i promenimo parametar **Initial output** na vrednost 0. Blok Gain množi tu temperaturu sa konstantom 9/5. Blok Add dodaje vrednost 32 rezultatu i kao izlaz daje temperaturu u Farenhajtima.

Otvorite blok Scope da vidite izlaz. Sada, izaberite opciju **Start** menija **Simulation** da biste pokrenuli simulaciju. Simulacija će se izvršavati prema vremenu koje definišete u polju koje se nalazi desno od dugmeta Play na paleti alatki prozora modela.



Zadatak 1.

Kreirajte Simulink model da simulirate signal koji ima sledeći oblik:

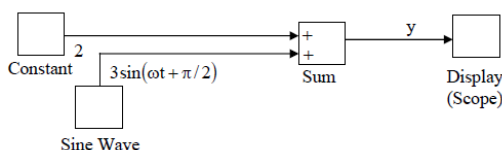
$$y = 2 + 3\sin(\omega t + \pi/2), \quad (t = 0 \text{ to } 10, \omega = 2 \text{ rad/s})$$

U simulacionom programu Simulink uradite sledeće:

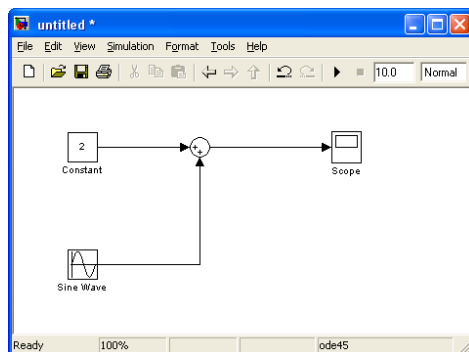
- ◆ Izračunajte vrednost y u zavisnosti od vremena t (od 0 do 10 sekundi)
- ◆ Prikažite vrednost y
- ◆ Snimite model

Rešenje

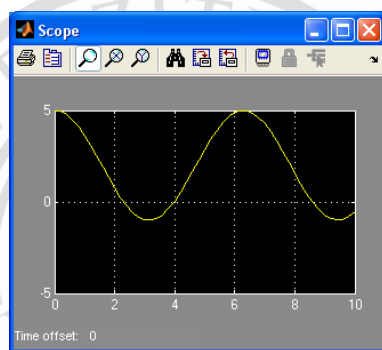
Blok dijagram algoritma za navedenu funkciju prikazan je na sledećoj slici.



Model je prikazan na sledećoj slici.

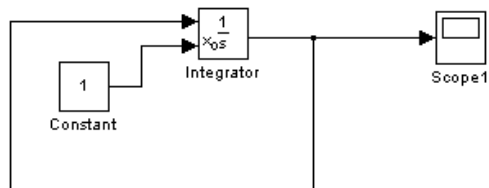


Grafički prikaz rezultatnog signala:



Zadatak 2.

Kreirajte model integratora sa početnom vrednošću prikazanog na sledećoj slici.



Rešenje:

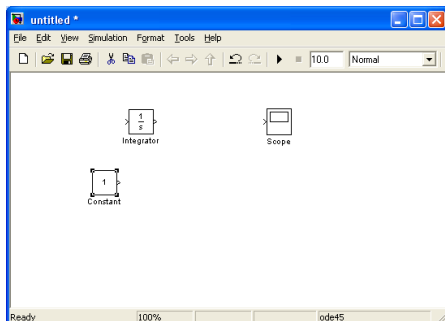
Postavljanje blokova:

Model koristi sledeće blokove:

/Continous/Integrator

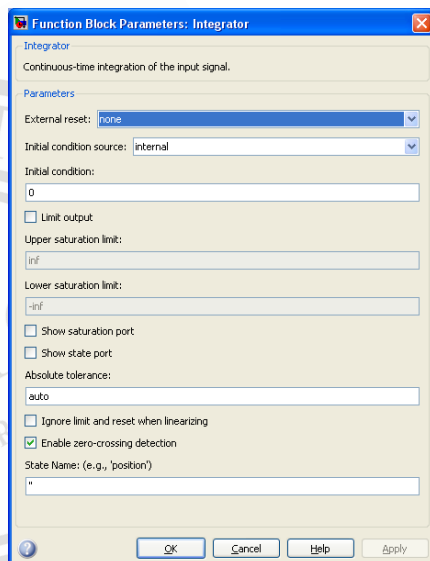
/Commonly Used Blocks/Constant

/Commonly Used Blocks/Scope

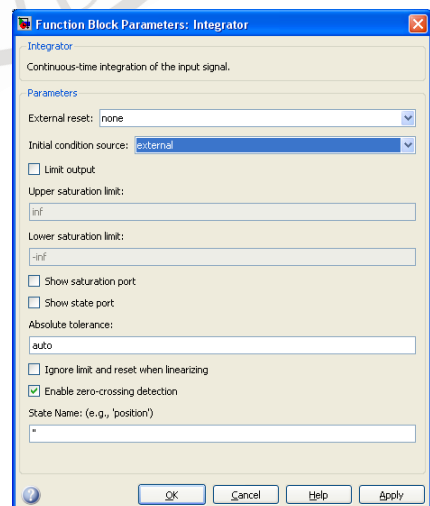


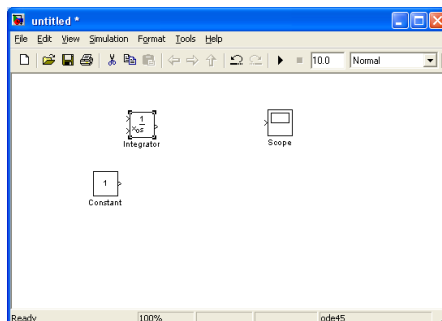
Konfigurisanje blokova:

Dvo-kliknite mišem na blok integratora da biste podesili njegove parametre



Selektujte kod opcije **Initial condition source** vrednost **external**. Blok integracije sada ima sledeći oblik:

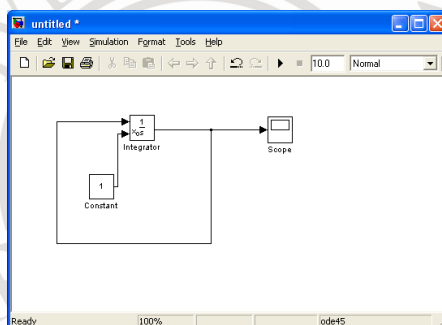




Kod bloka Constant neka ostanu podrazumevana podešavanja.

Povezivanje blokova:

Povežite izlaz bloka integratora sa blokom Scope, a zatim izlaz bloka Constant sa donjim ulazom bloka Integrator. Postavite miša na vezu blokova Integrator i Scope, pritisnite taster Ctrl i držeći levi taster miša pritisnut povucite izvedenu granu do gornjeg ulaza u blok Integrator.

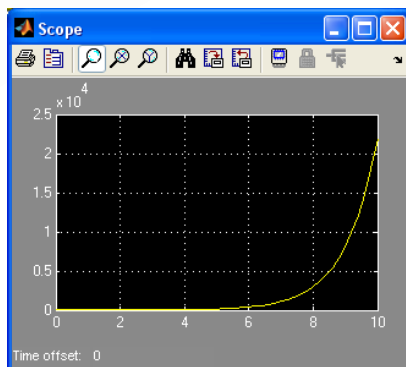


Pokretanje simulacije:

Kliknite mišem na dugme Start Simulation („Play”) na paleti alatki.

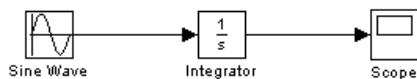
Rezultat:

Dvo-kliknite mišem na blok Scope da vidite rezultat simulacije. Možda će biti potrebno da uključite opciju Autoscale kod prikaza bloka Scope.



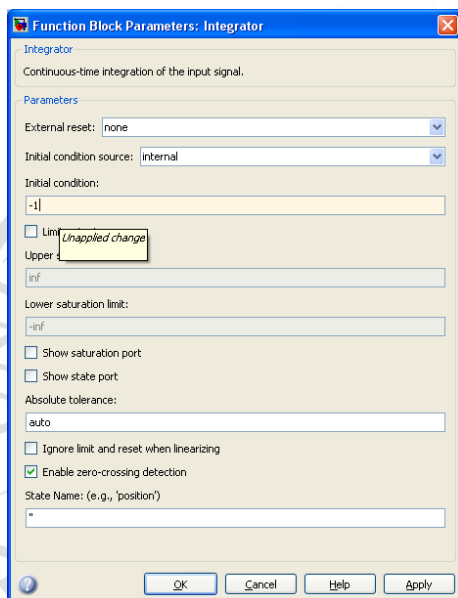
Zadatak 3.

Kreirajte blok dijagram prikazan na sledećoj slici.



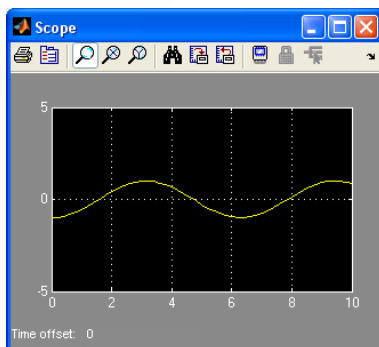
Rešenje:

Postavite i podesite blokove.



Ostale blokove ostavite na podrazumevanim podešavanjima.

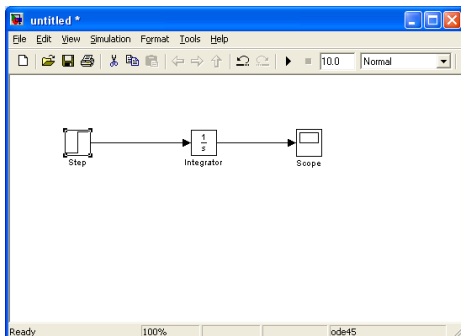
Pokrenite simulaciju i pogledajte rezultat.



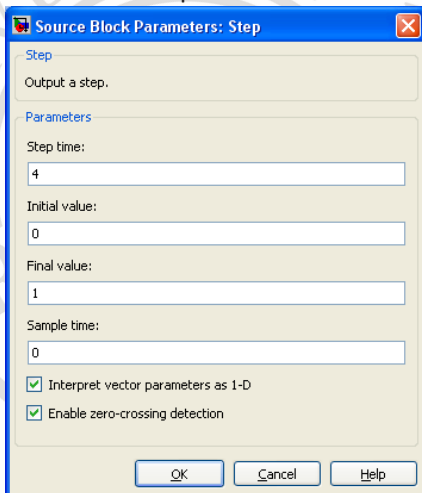
Zadatak 4.

Zameniti u prethodnom primeru blok Sine Wave sa blokom Step i uraditi simulaciju.

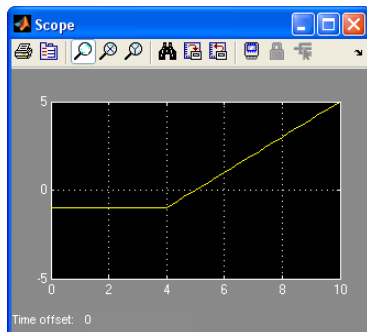
Rešenje:



Podesiti blok Step tako da ima sledeće parametre:



Rezultat simulacije ima sledeći oblik:



Zadatak 5.

Rešite sledeću diferencijalnu jednačinu:

$$\frac{dx}{dt} = -2x + 1, \quad t > 0.$$

$$x(0) = 0.$$

Napomena: Ulaz je 1 posle $t > 0$. Prema tome, može se koristiti step funkcija iz biblioteke blokova **Sources**. Međutim, potrebno je imati u vidu da u ovom slučaju **step time** nije $t=0$ već $t=1$.

Rešenje:

Ukoliko niste baš vični diferencijalnim jednačinama, koristite diferencijalni operator D .

Neka je $D = \frac{d}{dt}$. Diferencijalna jednačina sada ima oblik

$$Dx = -2x + 1 \tag{1}$$

ili

$$Dx + 2x = (D + 2)x = 1.$$

Konačno se dobija

$$x = \frac{1}{D + 2} \tag{2}$$

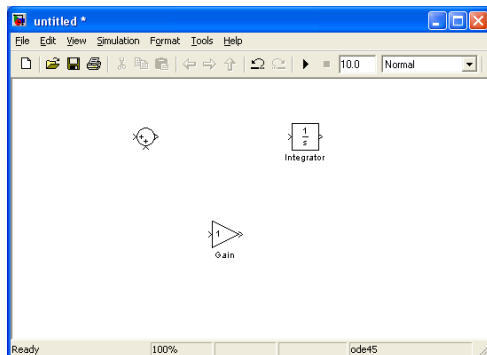
Za konfigurisanje modela u Simulinku može se koristiti ili (1) ili (2). Koristićemo sada:

$$Dx = -2x + 1.$$

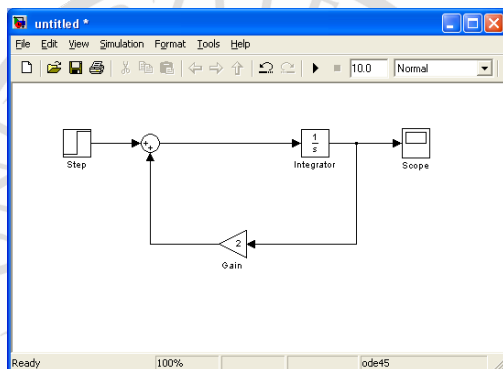
Laplasov operator transformacije s je skoro isti kao D , izuzev početnih uslova. U Simulinku $\frac{1}{s}$ znači integraciju.

Ulaz integratora je $Dx = \frac{dx}{dt}$ a izlaz x .

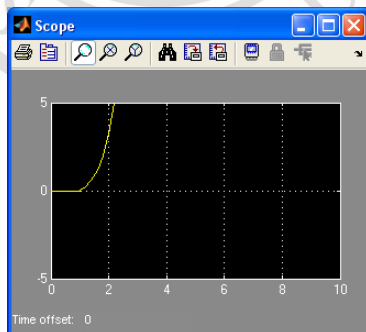
Prema tome, u konfiguraciji ćemo postaviti desnu stranu i povezati sve sa ulazom integratora.



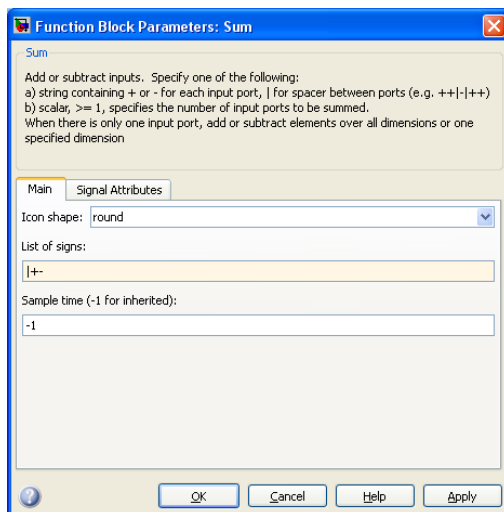
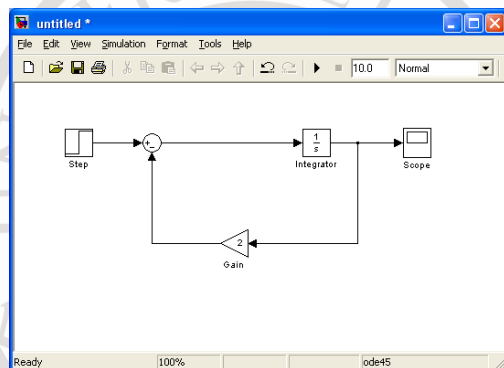
Da biste blok Gain okrenuli na suprotnu stranu potrebno je da ga selektujete miše, pritisnete desni taster miša i sa pomoćnog menija izaberete opciju Format/Flip Block.



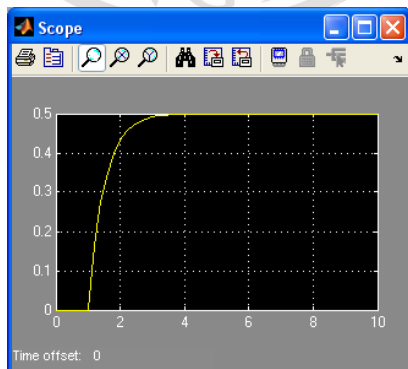
Pokrenite simulaciju i pregledajte rezultat.



Ako malo bolje pogledate jednačinu i dobijeni rezultat primetićete da je kod modela napravljena greška. Naime, kod sumiranja je napravljena greška u znaku. Korigujmo to u modelu i ponovo pokrenimo simulaciju.

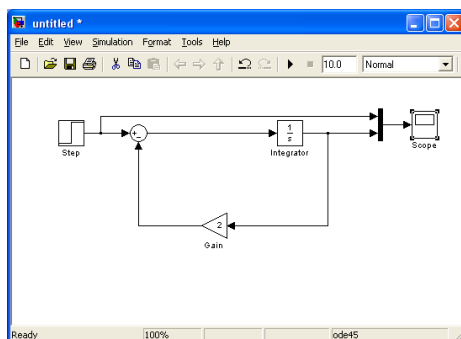


Rezultat sada ima sledeći oblik.

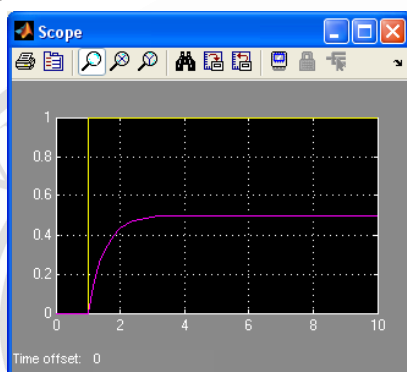


Ukoliko želite da istovremeno vidite i ulazni i izlazni signal potrebno je da u model ubacite blok **Mux** (multiplekser), koji ćete pronaći i **Signal Routing** grupi blokova.

Model sada ima sledeći oblik.



Blok Scope sada prikazuje i ulaz i izlaz.



Diferencijalne jednačine prvog i drugog reda se često proučavaju u kursovima o dinamičkim sistemima jer se takvi sistemi često sreću u praksi. Iz tog razloga ovde će biti razmatrano modelovanje tih jednačina u Simulinku.

Podsetnik iz fizike – vibracije/oscilacije

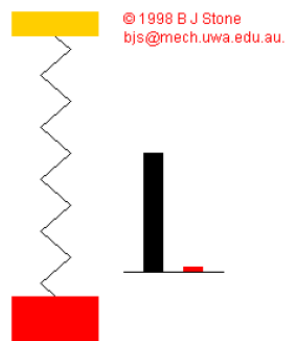
Sistem usled dejstva sila može da vibrira ukoliko postoje uslovi za konvertovanje energije iz jednog u drugi oblik.

Kod mehaničkog sistema masa-opruga reč je konvertovanju kinetičke energije kretanja u energiju koja se “skladišti” u opruzi, i obrnuto.

Slika prikazuje krutu masu vezanu preko opruge bez mase na čvrsti oslonac.

Ukupna energija sistema je ista – konvertuje se kinetička energija mase (crveno) u energiju opruge (crno) i obrnuto.

Posmatrajmo sledeći primer **sistema masa-opruga-prigušivač (sa prigušenjem)**:



Osnovne komponente mehaničkog sistema su masa, opruga i prigušivač.

Masa m krutog tela koji osciluje izražava se u kg.

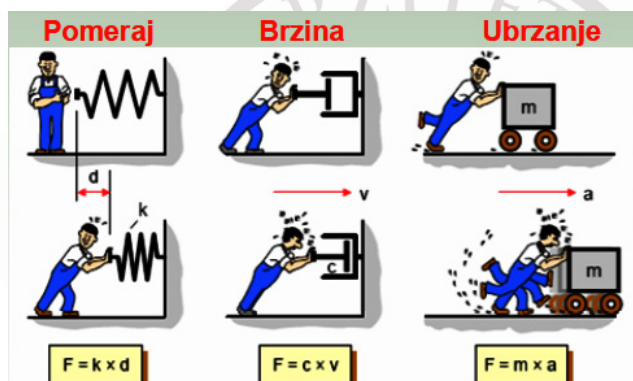
Opruga se karakteriše krutošću k [N/m]. Krutost se definiše kao statička sila koja je potreba da dovede do jediničnog pomeranja opruge.

Prigušivač se može zamisliti kao posuda sa viskoznom tečnošću u kojoj se kreće masa koja vibrira.

Prigušivač se karakteriše konstantom prigušenja c [Nm/s] koja se definiše kao sila prigušenja po jedinici brzine.

Delovanje silom na komponente sistema rezultira:

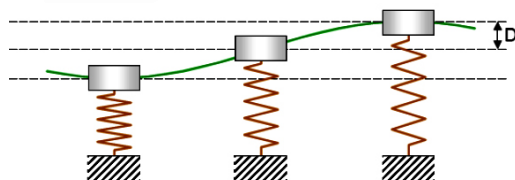
- ♦ ugibanjem opruge - pomeranje
- ♦ Promenom brzine u prigušivaču
- ♦ promenom ubrzanja mase



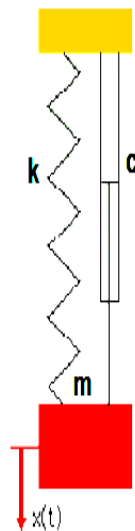
Sile koje se javljaju pri oscilovanju mase mehaničkog sistema:

- ♦ **elastična sila opruge** je proporcionalna pomeraju tela
- ♦ **sila prigušenja** je proporcionalna brzini kretanja tela
- ♦ **inercijalna sila** je proporcionalna ubrzanju

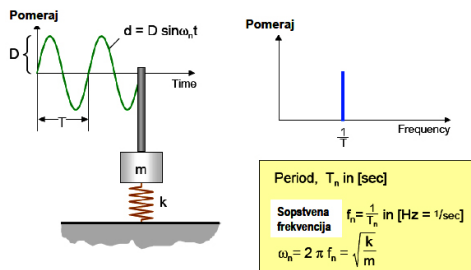
Nastajanje vibracija



Kada se sistem izvede iz ravnoteže (sabije se opruga) elastična sila teži da telo vrati u prvobitni položaj. Kada se to desi, pod dejstvom inercije tela (inercione sile) telo nastavlja da se kreće do trenutka kada je opruga maksimalno rastegnuta i kad elastična sila počinje ponovo da deluje ali u suprotnom pravcu.



Proces se ponavlja što dovodi do oscilatornog kretanja tela na sopstvenoj frekvenciji sistema koja zavisi od osobina sistema - mase krutog tela i krutosti opruge.



Period oscilovanja predstavlja vreme koje je potrebno da sistem izvrši jednu punu oscilaciju. **Frekvencija oscilovanja** predstavlja broj izvršenih oscilacija u jedinici vremena.

Elongacija (pomeraj) predstavlja rastojanje materijalne tačke ili tela od ravnotežnog položaja.

Amplituda predstavlja maksimalni pomeraj kod prostoperiodičnog kretanja.

Kada **sistem masa-opruga** osciluje energija sistema je ista ali u toku oscilovanja menja oblik iz kinetičke, kao posledicu kretanja tela, u potencijalnu, kao posledicu sabijanja i rastezanja opruge, i obrnuto.

Kinetička energija je maksimalna u ravnotežnom položaju - potencijalna je nula.

Na maksimalnom rastojanju od ravnotežnog položaja potencijalna energija je maksimalna a kinetička je jednaka nuli.

Uz zavisnosti od toga šta dovodi do vibracija postoje dve opšte klase vibracija:

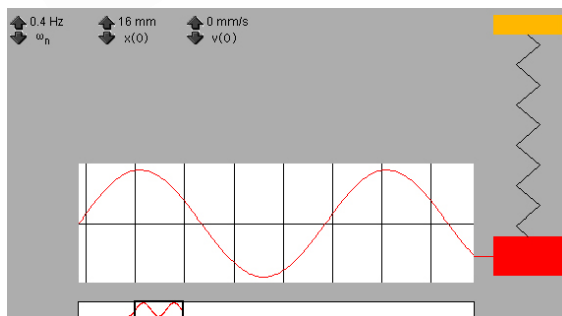
- ♦ Slobodne vibracije
- ♦ Prinudne vibracije

Slobodne i prinudne vibracije mogu biti **prigušene i neprigušene**.

Slobodne vibracije nastaju kada sistem osciluje oko ravnotežnog položaja pod dejstvom unutrašnjih sila – **spoljašnje sile ne postoje**.

Slobodne vibracije nastaju kada se sistem izvede iz ravnotežnog položaja i pusti da osciluje.

Sistem osciluje konstantnom amplitudom na jednoj frekvenciji - sopstvenoj frekvenciji.

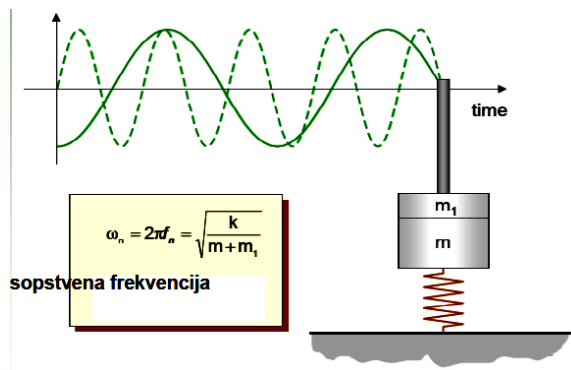


ω_n – sopstvena frekvencija

$x(0)$ – početni pomeraj

$v(0)$ – početna brzina

Povećanje mase dovodi do povećanja perioda oscilovanja odnosno do smanjenja frekvencije. Amplituda ostaje ista.



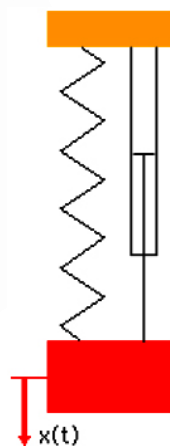
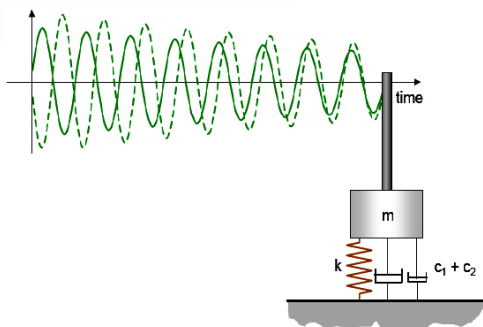
Ako u sistemu ne postoji prigušenje process pretvaranja kinetičke u potencijalnu energiju se kontinualno ponavlja, vibracije su kontinualne i amplituda vibracija ostaje nepromenjena.

Mehanizmi prigušenja (viskozno prigušenje, trenje ...) uzrokuju da se vibraciona energija nepovratno gubi npr. pretvaranjem u toplotnu energiju pri trenju.

Amplituda vibracija se smanjuje i nakon određenog vremena vibracije potpuno nestaju.

Sistem vibrira frekvencijsom koja se neznatno razlikuje od sopstvene frekvencije neprigušenog sistema.

Dodavanjem prigušivača smanjuje se amplituda oscilovanja sa vremenom i neznatno menja frekvencija oscilovanja.



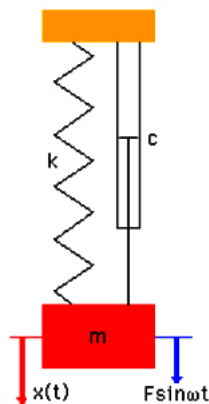
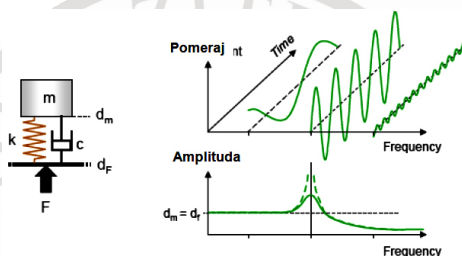
Ako se kod sistema sa prigušenjem želi održavanje vibracija neophodno je primeniti spoljašnju silu koja će da nadoknadi gubitak energije usled prigušenja.

Usled dejstva neke spoljašnje sile nastaju prinudne vibracije.

Frekvencija oscilovanja sistema zavisi od frekvencije pobudne sile.

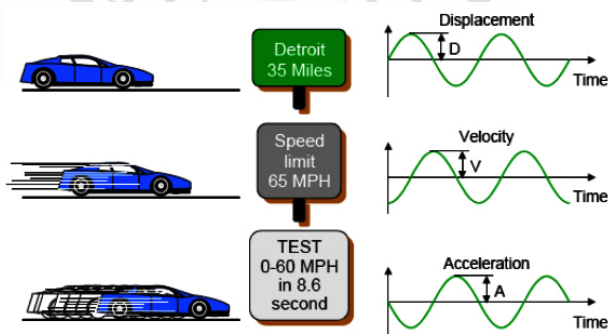
Ukoliko se na sistem deluje spoljašnjom sinusoidalnom silom, sistem će pratiti silu, što znači da će kretanje sistema imati istu frekvenciju kao i spoljna sila.

Za frekvencije koje su manje od sopstvene frekvencije, amplituda vibracionog sistema će se povećavati sa porastom frekvencije prinudne sile. Maksimum se postiže na sopstvenoj frekvenciji. Ukoliko u sistemu ne postoji prigušivač ($c=0$), amplituda dostiže beskonačnu vrednost.



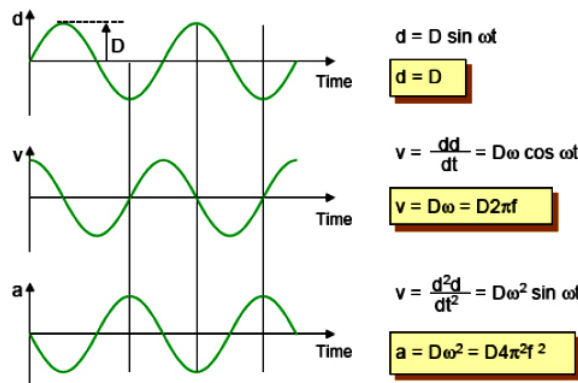
Osnovne veličine koje se koriste za merenje i opisivanje vibracija su:

- pomeraj $d[m]$
- brzina $v[m/s]$
- ubrzanje $a[m/s^2]$



Tri veličine su usko povezane. Ako se posmatra harmonijsko kretanje oblik i period signala ostaju isti za sve tri veličine.

Osnovna razlika je u postojanju fazne razlike između amplitudno-vremenskih karakteristika ova tri parametra.



Ukoliko je signal pomeraja poznat, ostali parametri mogu se odrediti jednostrukim ili dvostrukim diferenciranjem ovog signala.

Ako se fazna razlika ignoriše (kao što je uobičajeno), brojne vrednosti brzine i ubrzanja mogu se odrediti jednostavnim množenjem.

Ukoliko je parametar koji se poznaje ubrzanje, druga dva parametra mogu se jednostavno odrediti jednostrukim ili dvostrukim integraljenjem signala ubrzanja.

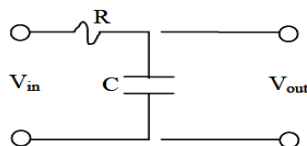
* * *

Zadatak 1.

U praksi se često jednostavno **RC-kolo** koristi kao tzv. **niskopropusni filter**. Napraviti u Simulinku model tog kola.

Rešenje:

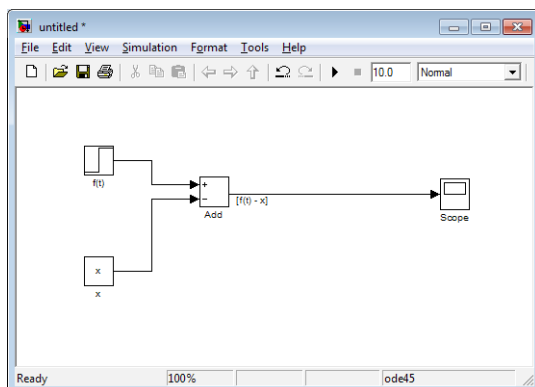
Na sledećoj slici prikazano je RC kolo.



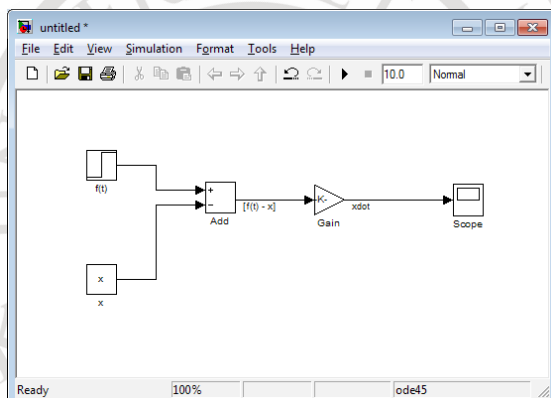
To kolo se modeluje diferencijalnom jednačinom prvog reda.

$$\dot{x} = \frac{1}{RC} [f(t) - x]$$

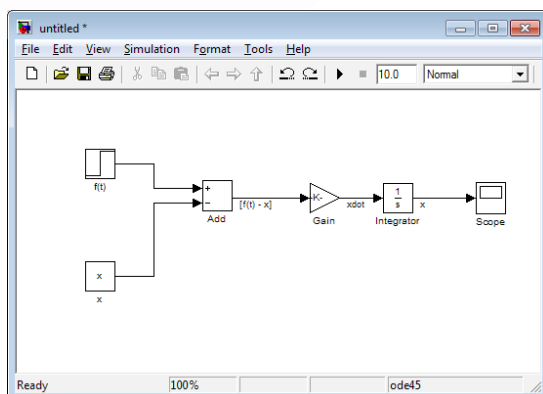
Ovde je $\dot{x}$ (xdot) brzina promene izlaznog napona, R i C su konstante, $f(t)$ je prinudna funkcija (ulazni napon) a x je izlazni napon. Model ćemo formirati član-po-član. Prvo, u zagradi vidimo da se član x oduzima od člana $f(t)$. Ako zamislimo da svaki od ovih članova daje signal, tu relaciju možemo modelovati kao što je prikazano na sledećoj slici.



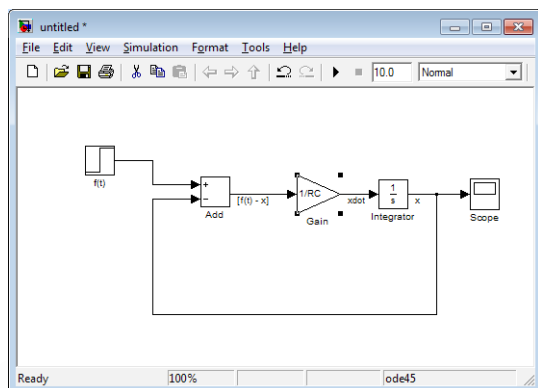
Daljim posmatranjem izraza, zapazamo da se sadržaj zagrade množi konstantom $1/RC$. U Simulinku ćemo u tu svrhu koristiti blok **Gain**. Pošto su svi članovi sa desne strane izraza tu, znamo da izlazni signal mora biti jednak levoj strani jednačine, koja je $\dot{x}$.



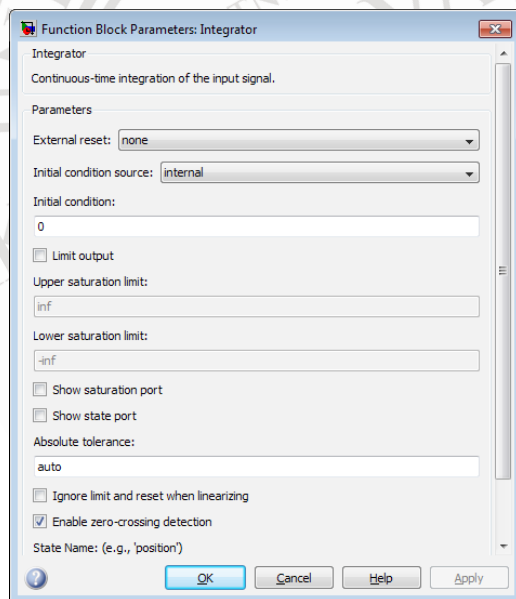
Međutim, mi smo zainteresovani za x a ne za $\dot{x}$. Kako možemo da uzmemo $\dot{x}$ izlazni signal i dobijemo x izlaz? Problem razrešava blok **Integrator**.



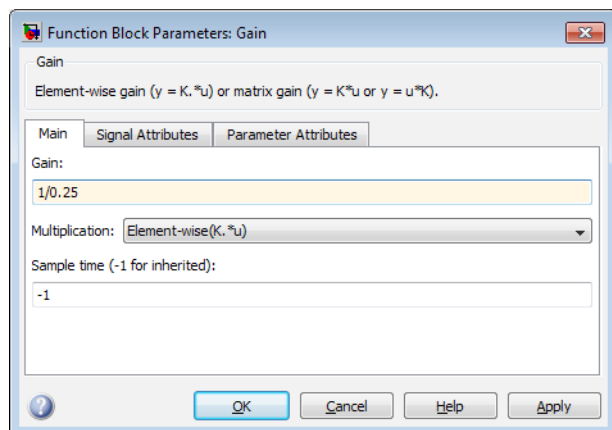
Sada imamo željeni x izlaz, ali primećujemo da je x takođe i ulaz sistema. U ovom našem modelu ulaz x je „mrtva” grana. Drugim rečima, ovde ne postoji realni signal. Kako možemo da napravimo da x bude i izlaz i ulaz sistema? Odgovor je u povratnoj petlji, tj. uzmemo izlazni signal x i vratimo ga na ulaz sistema. Posle kraćeg manipulisanja linijama, model će imati oblik prikazan na sledećoj slici.



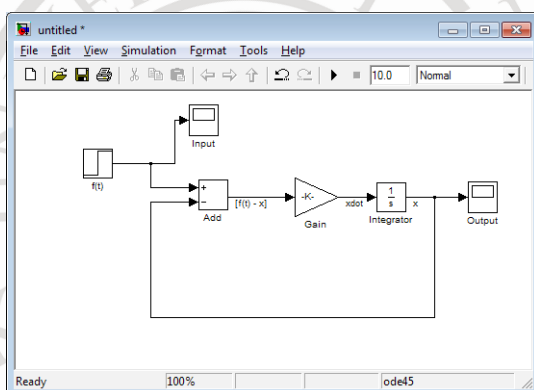
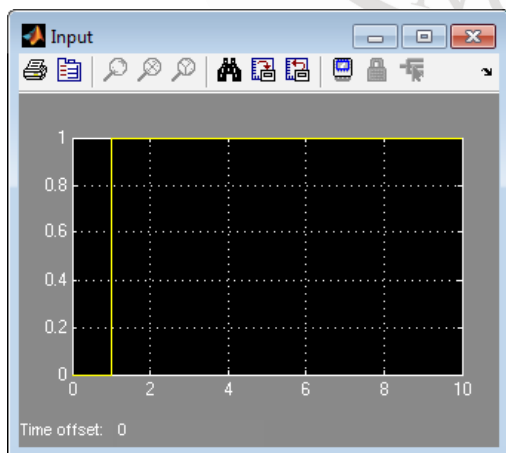
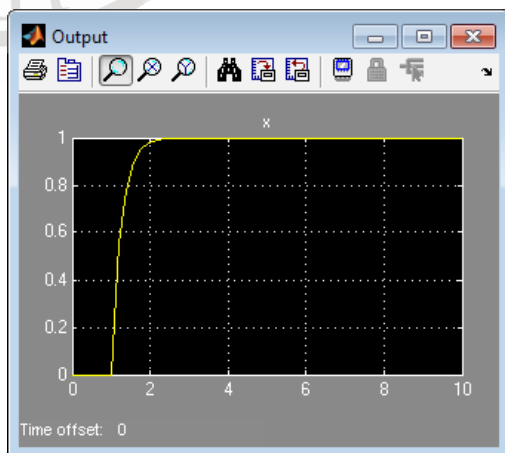
Dvo-kliknite mišem na blok **Integrator** i podesite parametre kao na datoj slici.



Sada dvo-kliknite na blok **Gain** i podesite parametre kao na sledećoj slici.



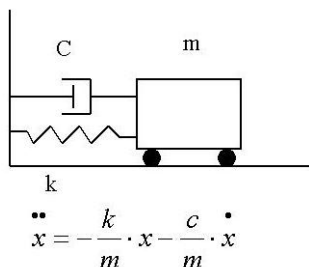
Na izlazu modela je već postavljen blok **Scope**, pokrenimo simulaciju i pogledajmo rezultat.



Uočite kako odziv sistema kasni u odnosu na step ulaz. Odziv je definisan **vremenskom konstantom sistema**, $\tau = RC$.

Zadatak 2.

Treba simulirati sistem prikazan na sledećoj slici. Polazi se od pretpostavke da nema dejstva nikakve spoljašnje sile.



Rešenje:

Sistem se opisuje/modeluje na osnovu sledećih formula:

Sila opruge $F_s = -kx$;

Sila prigušenja $F_d = -C\dot{x}$;

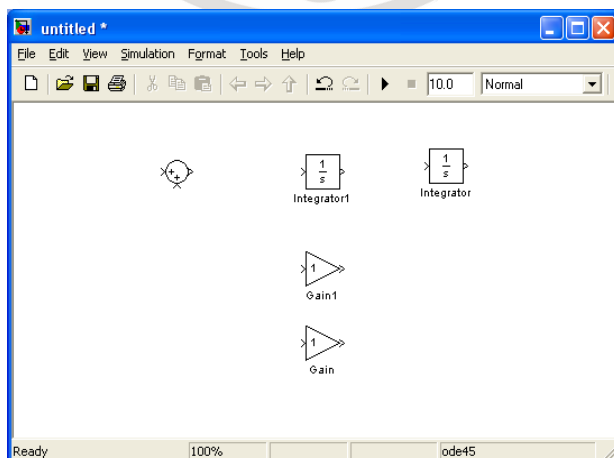
Sila ubrzanja $F_a = m\ddot{x}$.

Kada je sistem u ravnoteži stanje se opisuje sledećom diferencijalnom jednačinom:

$$\ddot{x} + \frac{C}{m} \dot{x} + \frac{k}{m} x = 0$$

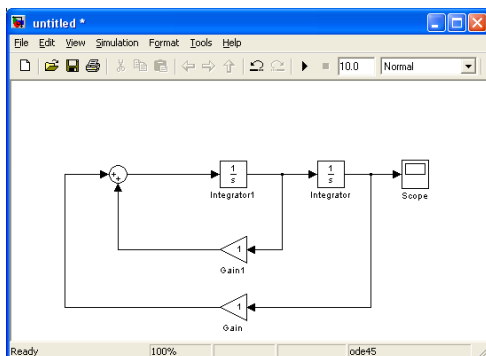
Posmatranjem navedene jednačine zaključujemo da sistem sadrži dva integratora (od ubrzanja do brzine i od brzine do položaja).

Pokrenimo Simulink i kreirajmo novi model. U prozor modela prevucimo iz biblioteke Continuous dva integratora, a zatim dva bloka Gain i jedan blok za sumiranje. Model sada ima oblik prikazan na sledećoj slici.

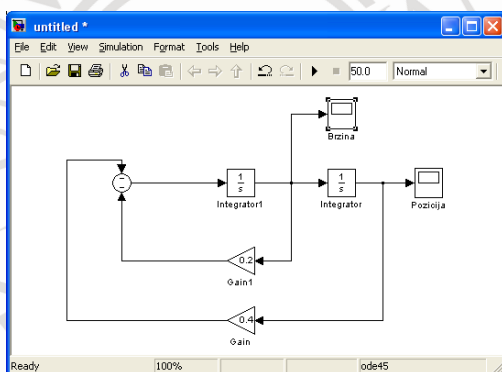


Desnim klikom miša na blokove pojačanja aktivirajmo kontekstni meni i promenimo orijentaciju ovih blokova na levu stranu.

Sada treba dodati blok Scope i povezati blokove. Konačni izgleda modela dat je na sledećoj slici.

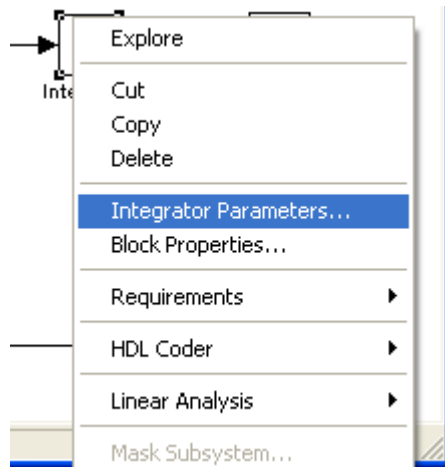
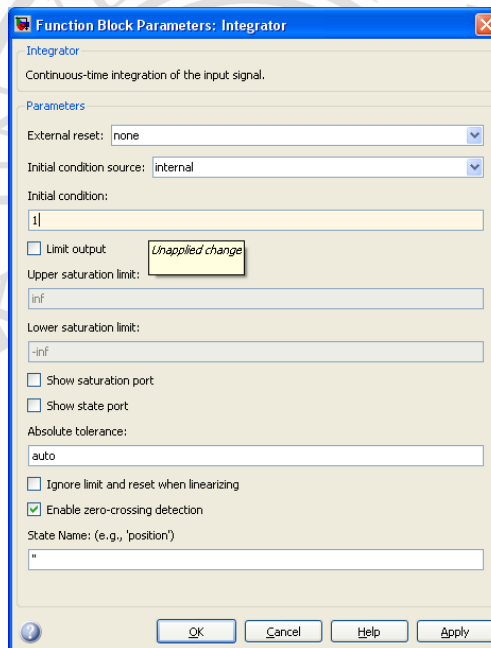


Podesimo vrednosti blokova pojačanja i kod sumatora postavimo odgovarajuće predznake.

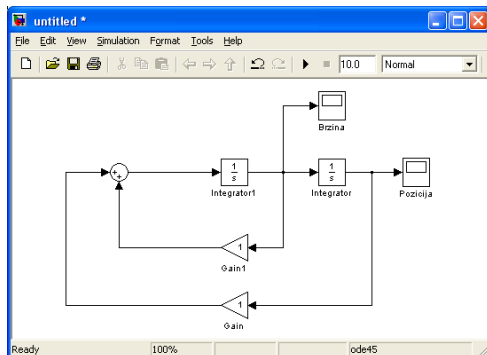


Posmatrajući model i diferencijalnu jednačinu uočava se da prvi blok pojačanja predstavlja član C/m a drugi blok pojačanja član k/m .

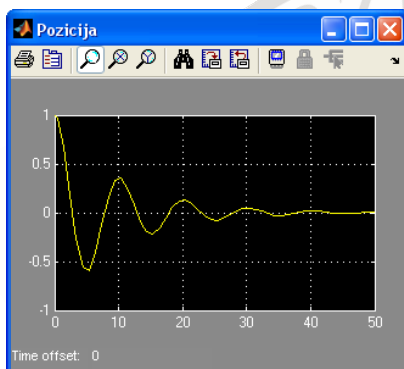
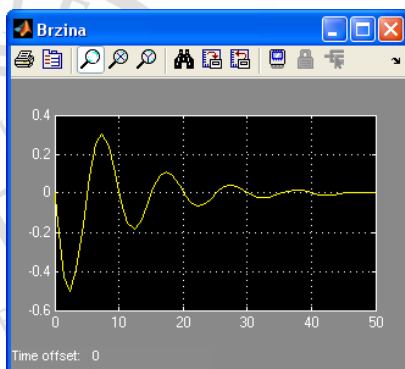
Iako nema dejstva spoljašnje sile, kretanje je moguće ako početni uslovi nisu nula. Kako se radi o diferencijalnoj jednačini drugog reda, postoje dva početna uslova. Kliknite desnim tasterom miša na drugi integrator i izaberite stavku **Integrator Parameters** i u istoimenom okviru za dijalog kod parametra **Initial Conditions** upišite vrednost **1**.



Ovime je postavljen početni uslov mase na 1. U suštini to je isto kao da je masa primorana da bude na toj poziciji od strane spoljašnje sile. Sila iščezava u trenutku nula. Dodajmo modelu još jedan blok Scope da bi vizuelizovali brzinu (izlaz prvog integratora).



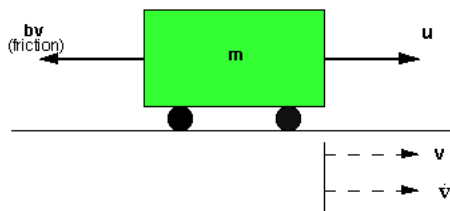
Spremni smo za vizuelizaciju simulacije. Pokrenimo simulaciju a zatim dvo-klikom na blokove Scope pregledajmo vremenske dijagrame pozicije i brzine.



Posmatranjem dijagrama uočava se da je početna vrednost pozicije 1 dok je početna brzina 0.

Model kolica

Model kolica je vrlo jednostavan. Ako se inercija točkova zanemari a uzme u obzir frikcija (trenje, koje je proporcionalno brzini kolica) koja se suprotstavlja kretanju kolica, tada se problem svodi na sistem mase i opruge koji je ranije razmatran.



Prema Njutnovom zakonu jednačina modela ovog sistema je:

$$m \frac{dv}{dt} = u - bv$$

Gde je u vučna sila (pogonskog motora). Neka za ovaj primer važe sledeće vrednosti:

$$m = 1000\text{kg}$$

$$b = 50\text{Nsec/m}$$

$$u = 500\text{N}$$

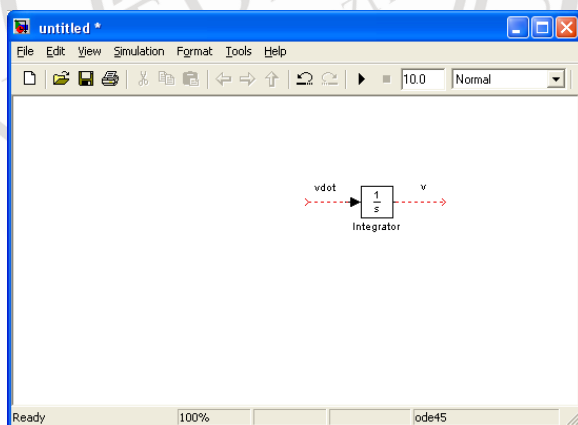
Kreiranje modela

Sistem će se modelirati sumiranjem sila koje deluju na masu i integracijom ubrzanja da bi se dobila brzina. Prvo ćemo modelirati u Simulinku integral ubrzanja

$$\int \frac{dv}{dt} = v$$

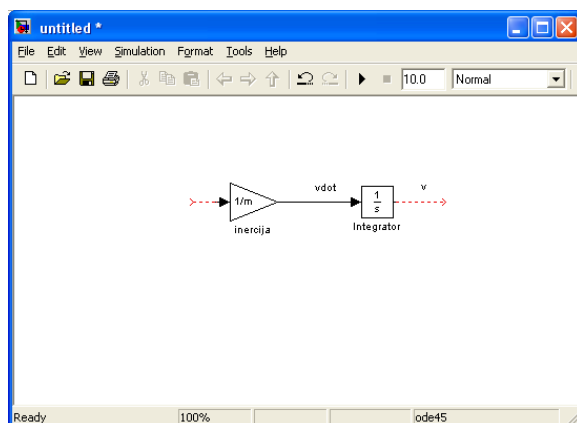
Pokrenimo Simulink, kreirajmo novi model i krenimo sa ubacivanjem blokova koji će sačinjavati model.

- ♦ Ubacimo blok Integrator i nacrtajmo ulaznu i izlaznu liniju.
- ♦ Označimo ulaznu liniju sa “ $\dot{v}$ ” a izlaznu sa “ v ”. Da bi dodali te oznake treba da dvo-kliknemo na prazan prostor neposredno iznad linije i upišemo te oznake.



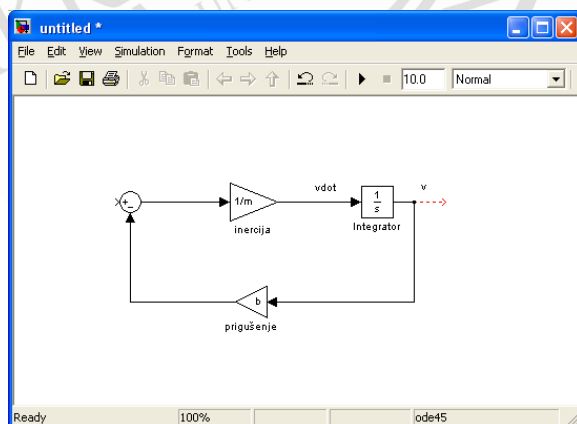
Kako je ubrzanje (dv/dt) jednako sumi sila podeljeno sa masom, podelićemo dolazeći signal sa masom.

- ♦ Ubacićemo blok Gain i povezati sa ulazom integratora a nacrtati ulaznu liniju blok pojačanja.
- ♦ Dvo-klikom miša na blok Gain promenićemo njegovu vrednost na vrednost “ $1/m$ ”.
- ♦ Takođe, promenićemo oznaku bloka Gain u “inercija” dvo-klikom miša na reč “Gain” neposredno ispod bloka.



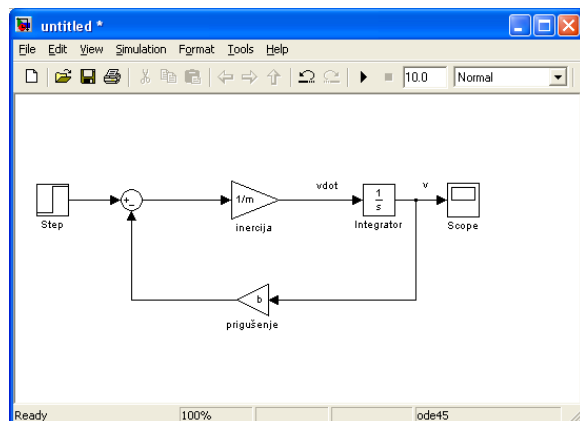
Sada ćemo dodati sile koje su zastupljene u polaznoj jednačini. Prvo, dodaćemo silu prigušenja (otpora).

- ♦ Ubacimo blok sumiranja u model i povežimo ga sa ulazom bloka pojačanja.
- ♦ Izmenimo znakove sumiranja ovog bloka u “+”.
- ♦ Postavimo još jedan blok pojačanja ispod bloka inercije i orijentišimo ga sa desna na levo (tasterska prečica **Ctrl+I**).
- ♦ Postavimo vrednost pojačanja na “b” i preimenujmo blok u “prigušenje”.
- ♦ Postavimo miša na izlaz integratora i pritisnimo taster Ctrl i dok je taster pritisnut iscrtajmo liniju koja povezuje integrator sa ulazom bloka prigušenja.
- ♦ Povežimo izlaz bloka prigušenja sa negativnim ulazom bloka sumiranja.

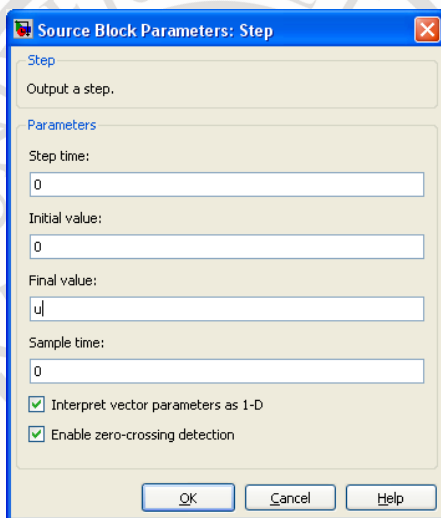


Drua sila koja deluje na masu je ulaz u . Kao ulaz primenićemo step blok (funkciju).

- ♦ Postavimo u model blok Step i povežimo ga sa pozitivnim ulazom sumatora.
- ♦ Da bi mogli da vidimo izlaznu brzinu postavimo na izlaz integratora blok Scope.



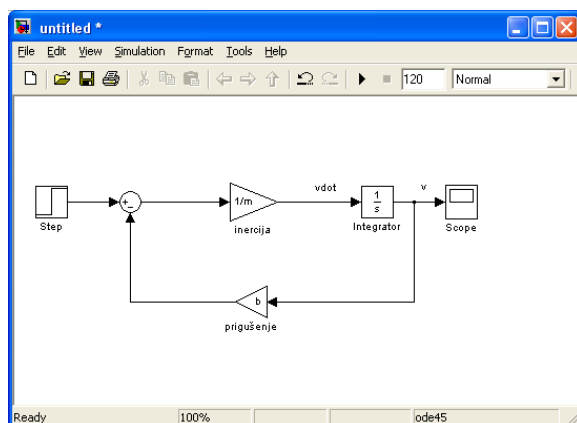
- ♦ Da bi obezbedili odgovarajući step ulaz od 500 u $t=0$, dvo-kliknimo mišem na blok Step i postavimo parametre **Step Time** na "0" a **Final Value** na "u".



Odziv otvorenog sistema

Da bi simulirali sistem mora se postaviti vreme simulacije.

- ♦ Postavimo parametar Stop Time na "120". To je sasvim dovoljno vremena da se vidi odziv otvorenog sistema.



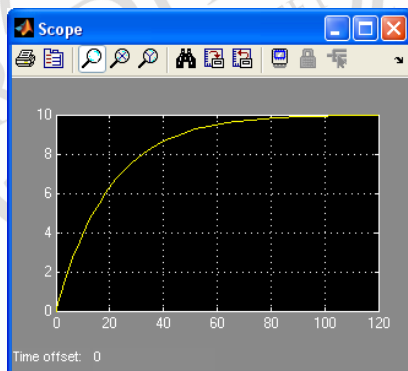
Sada se moraju postaviti fizički parametri. U komandnom modu **MATLAB**-a zadajmo sledeće komande:

$m=1000;$

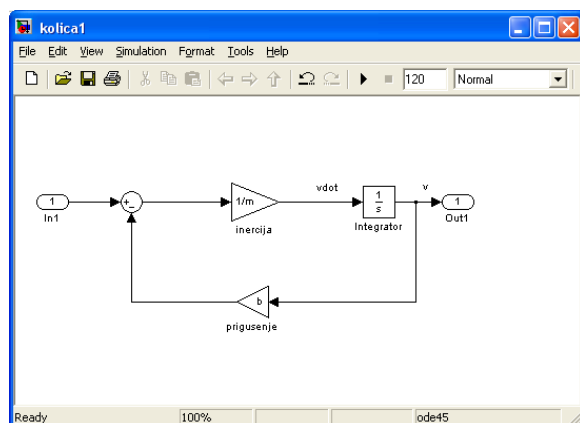
$b=50;$

$u=500;$

Pokrenimo simulaciju (**Ctrl+t** ili Start na meniju Simulation). Kada je simulacija završena, dvo-klikom miša na blok Scope pogledajmo izlaz. Aktivirajmo i opciju za autoskaliranje da bi preciznije videli izlaz.



Zamenimo sada blok **Step** sa blokom **In1** a blok **Scope** sa blokom **Out1**. Ovime definišemo ulaz i izlaz sistema za potrebe procesa ekstrakcije modela. Snimimo model pod nazivom “**kolica1**”.

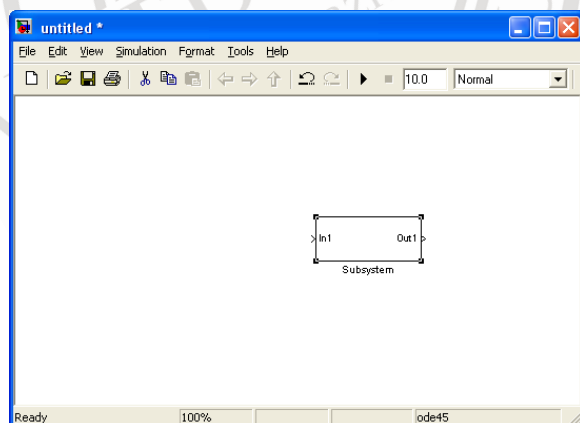


Zatvorimo prozor modela.

Implementiranje PI regulatora (kontrole)

Neka je projektovan PI regulator sa parametrima $K_p=800$ i $K_i=40$ da bi se dobio željeni odziv. Implementaciju ćemo nastaviti tako što ćemo prvo prethodno kreirani i sačuvani model sistema u otvorenoj vezi postaviti u blok **Subsystem** novog modela.

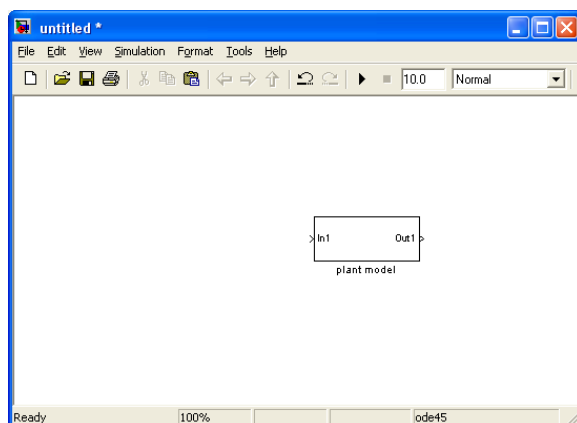
- ♦ Kreirajmo novi model.
- ♦ Postavimo blok Subsystem u naš novi model.



- ♦ Dvo-kliknimo mišem na ovaj blok i videćemo da su ulazni terminalni i izlazni terminalni blok povezani linijom. Možemo sve te elemente da obrišemo.
- ♦ Otvorimo prethodno sačuvani model kolica1.mdl.
- ♦ Izaberimo u meniju prozora Edit komandu Select All (Ctrl+A), a zatim Copy (Ctrl+C).
- ♦ Prebacimo se sada u prazan prozor bloka Subsystem našeg novog modela i izaberimo stavku Paste u meniju Edit (Ctrl+V). Sada ćemo videti naš prvobitni model u

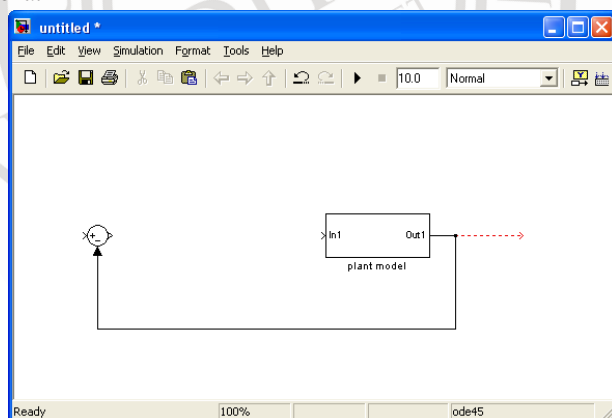
prozoru podsistema. Zatvorimo taj prozor. Zatvorimo i prozor originalnog modela.

- ♦ U prozoru novog modela videćemo samo ulazni i izlazni terminal bloka Subsystem. Preimenujmo taj blok "plant model".



Sada ćemo formirati PI regulator uz plant model. Prvo, uradimo povratnu spregu.

- ♦ Postavimo na levoj strani prozora blok sumiranja i promenimo redosled znakova u "+-".
- ♦ Prekinimo izlaznu liniju bloka podsistema i iscrtajmo povratnu liniju do negativnog ulaza sumatora.



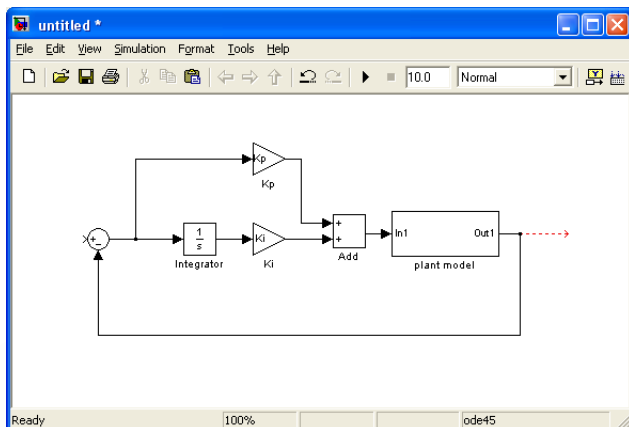
Izlaz iz sumatora je signal greške. Od te tačke generišemo proporcionalnu i integralnu komponentu.

- ♦ Posle sumatora postavimo integrator i povežemo ga sa sumatorom.
- ♦ Postavimo posle integratora blok pojačanja i povežimo ga sa integratorom kako bi obezbedio integralno pojačanje.
- ♦ Označimo integrator sa Ki i dodelimo bloku vrednost od Ki.
- ♦ Postavimo još jedan blok pojačanja iznad bloka pojačanja integratora. Prekinimo izlaz iz sumatora i povežemo ga sa ovim pojačanjem.

- ♦ Označimo ovaj blok pojačanja sa K_p i dodelimo mu vrednost od K_p .

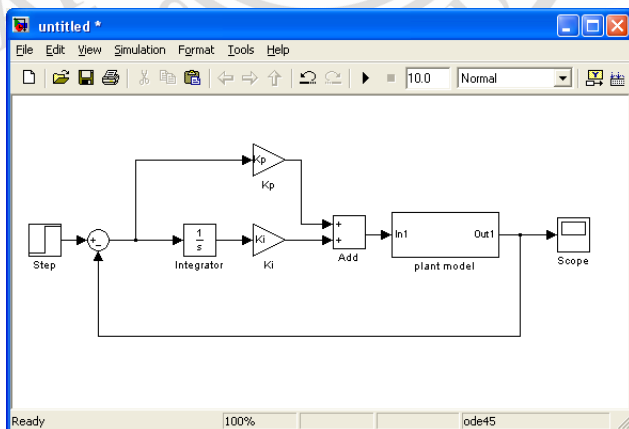
Upravo smo dodali proporcionalnu i integralnu komponentu i sada njihovo zbirno dejstvo treba da primenimo na naš sistem/model (plant).

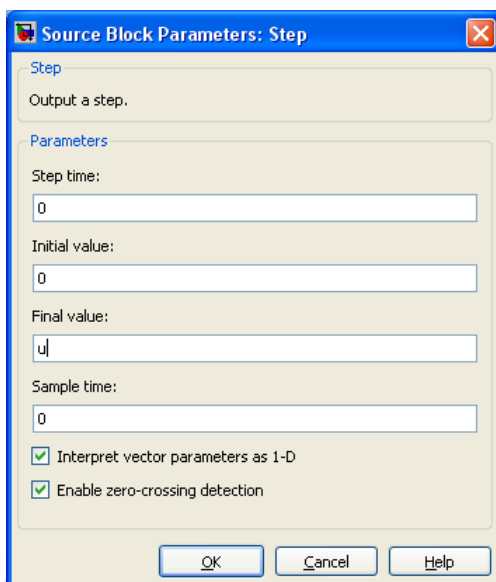
- ♦ Postavimo sabirač između bloka K_i i plant modela a zatim povežimo izlaze blokova pojačanja sa ulazima sabirača.
- ♦ Povežimo izlaz sabirača sa ulazom plant modela.



Na kraju, primenimo step ulaz i za pregled izlaza blok Scope.

- ♦ Priključimo blok Step na pozitivan ulaz sumatora povratne sprege.
- ♦ Priključimo blok Scope na izlaz plant modela.
- ♦ Dvo-kliknimo na blok Step i postavimo **Step Time** na "0" a **Final Value** na "u". Ovo omogućava da se veličina magnitude menja izvan Simulinka (iz Matlaba).

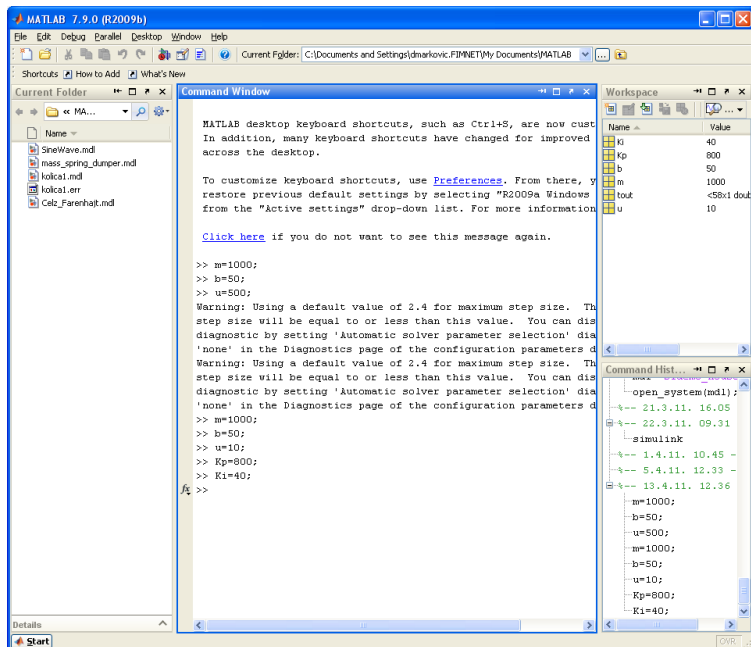




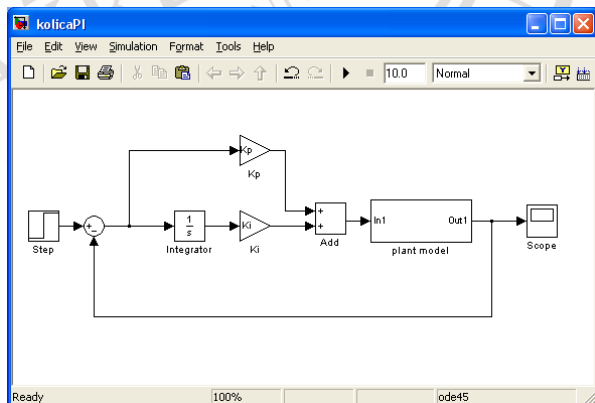
Odziv u zatvorenoj sprezi

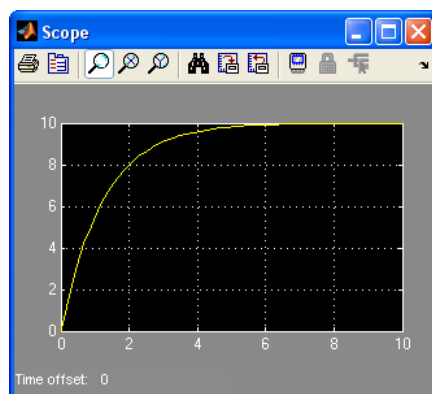
Za simulaciju sistema treba postaviti odgovarajuće vreme. Za ovaj slučaj postavimo vrednost “10” u polje **Stop Time**. Kod dizajniranja vremena uspostavljanja zahtve je bio da bude manje od 5 sekundi, stoga je trajanje simulacije od 10 sekundi sasvim dovoljno da se vidi odziv sistema. Fizički parametri se moraju sada postaviti. U komandnom modu MATLAB-a unesimo sledeće komande:

```
m=1000;  
b=50;  
u=10;  
Kp=800;  
Ki=40;
```



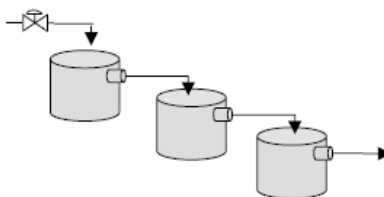
Pokrenimo simulaciju (Ctrl+t ili Start na meniju Simulation). Kada se simulacija za-
vrši, dvo-kliknimo mišem na blok Scope i aktivirajmo opciju autoskaliranja. Trebalo bi
da vidite sledeći prikaz.





Modeliranje složenijih sistema

Neka imamo sistem koji sačinjavaju kaskadno povezana tri reaktora:



i neka je prenosna funkcija tog sistema:

$$G_p(s) = \frac{6}{(2s+1)(4s+1)(6s+1)}$$

Zadatak:

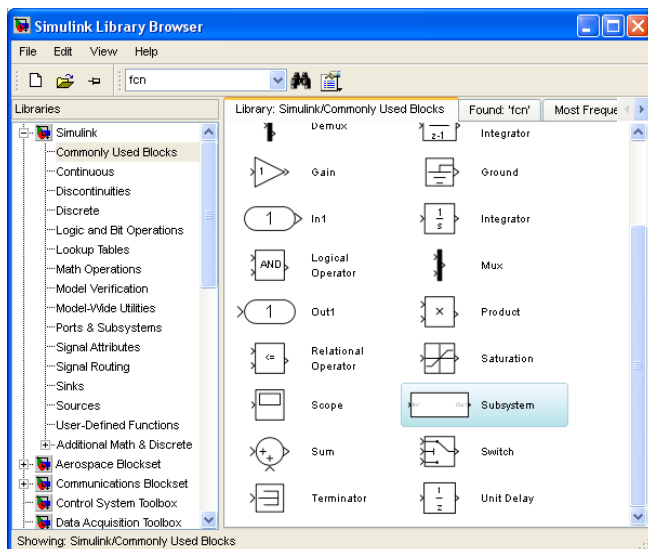
Sistem treba da regulišemo koristeći PID regulator. Da bi podesili parametre regulatora koristićemo simulaciju u Simulinku tako što ćemo podesiti PID regulator koristeći praktičnu Ziegler-Nicholovu metodu.

Potrebno je prvo da napravimo model sistema sa povratnom spregom u koji je uključen regulator.

Prenosnu funkciju PID regulatora koristićemo u sledećem obliku:

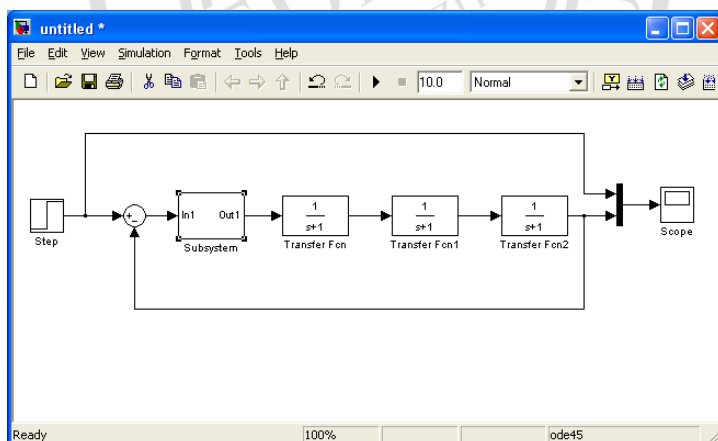
$$G(s) = K^* \left(1 + \frac{s}{\tau_{aul}} + \tau_{auD} s \right)$$

a u Simulinku ćemo je realizovati preko bloka **Subsystem** koji se u biblioteci nalazi u uobičajeno korišćenim blokovima.



Prenosne funkcije reaktora ćemo realizovati preko tri kaskadno povezana bloka **Transfer Fcn** iz grupe **Continuous**.

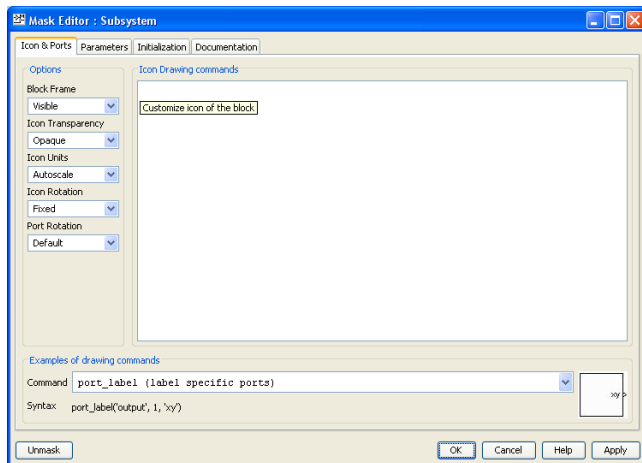
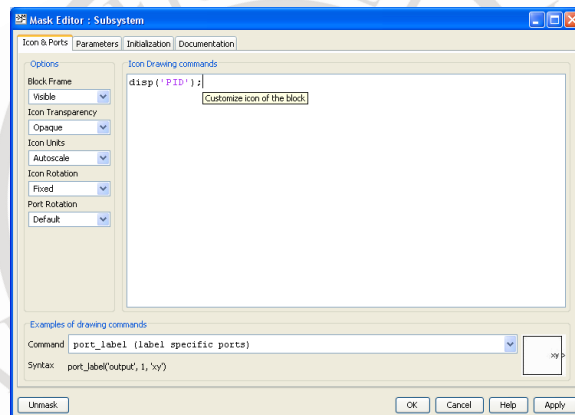
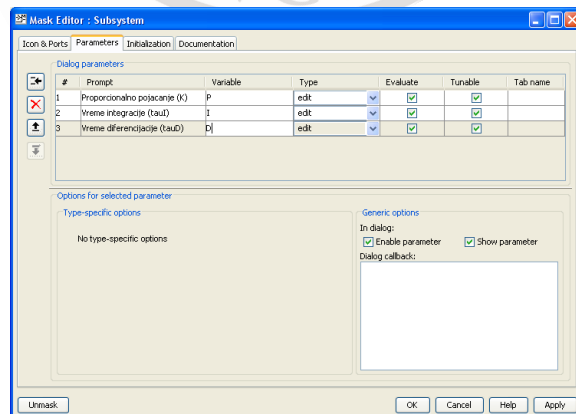
Model ima sledeći opšti oblik:

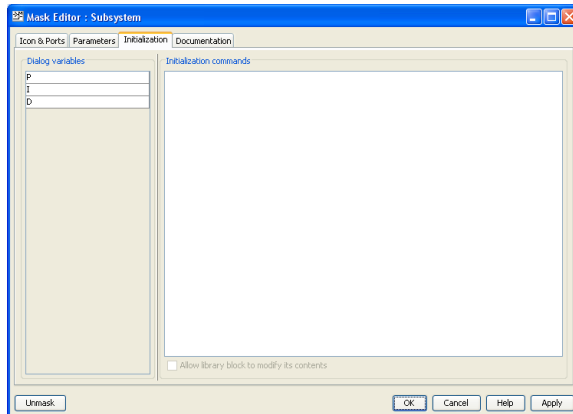
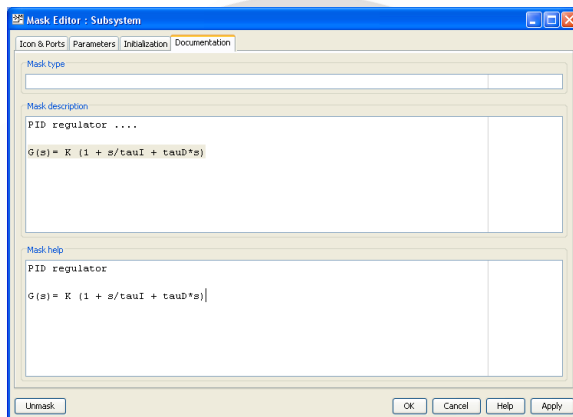


Sada ćemo prilagoditi model da dobije konačni izgled za simulaciju.

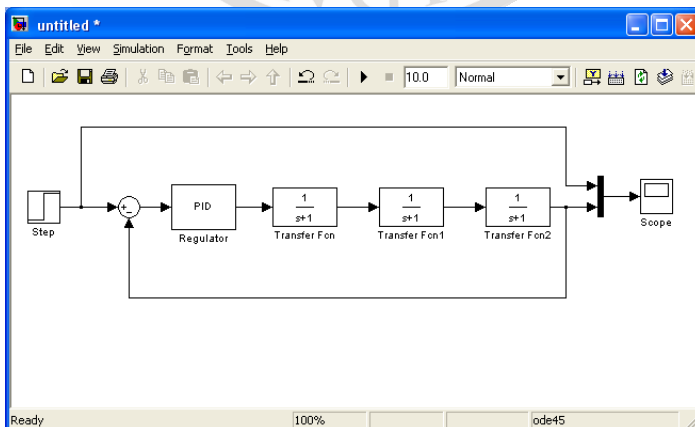
Prilagodimo (maskirajmo) prvo podsistemski blok a zatim blokove prenosnih funkcija reaktora.

Selektujemo mišem blok podsistema i pritisnimo desni taster miša a zatim iz kontekstnog menija selektujemo opciju **Mask Subsystem**.

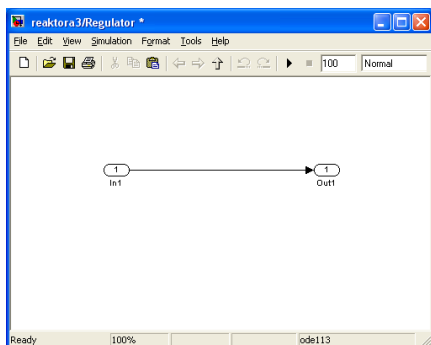




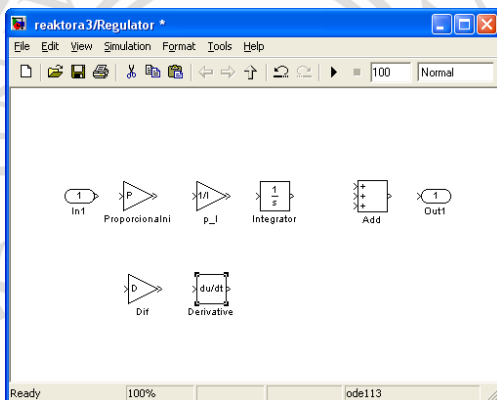
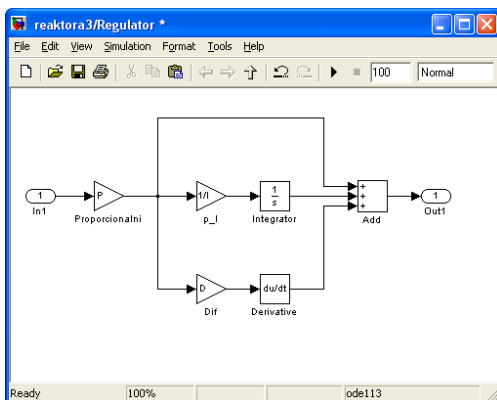
Blok regulatora je deklarativno prilagođen.



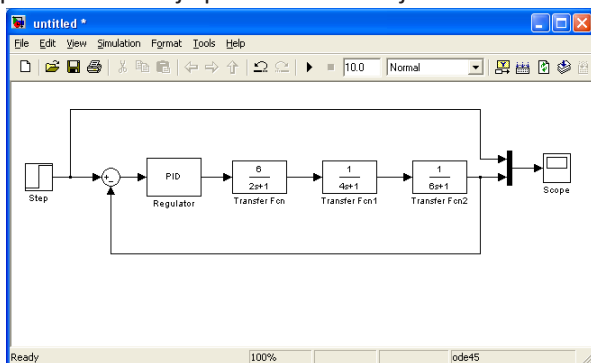
Sada uradimo desni klik mišem na blok PID regulatora i u kontekstnom meniju selektujemo opciju **Look Under Mask**. U novo-otvorenom prozoru definisaćemo naš PID regulator.



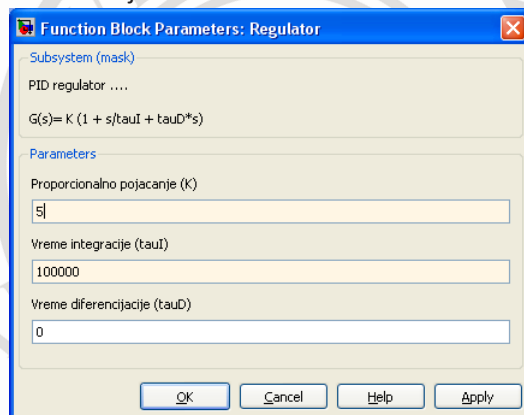
Prevlačenjem elemenata iz Simulink biblioteka formi ćemo PID regulator prema ranije navedenom obliku prenosne funkcije ($G(s) = K \cdot (1 + s/\tau_{aul} + \tau_{auD} \cdot s)$).



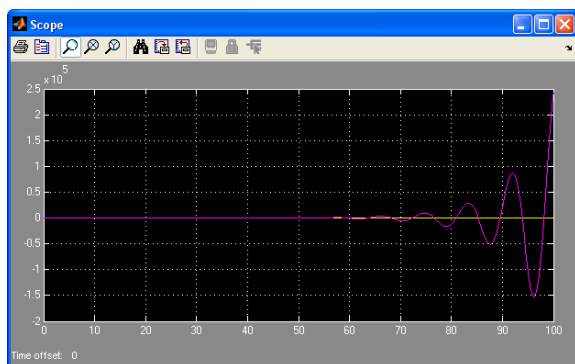
Izgled modela posle definisanja prenosnih funkcija reaktora.



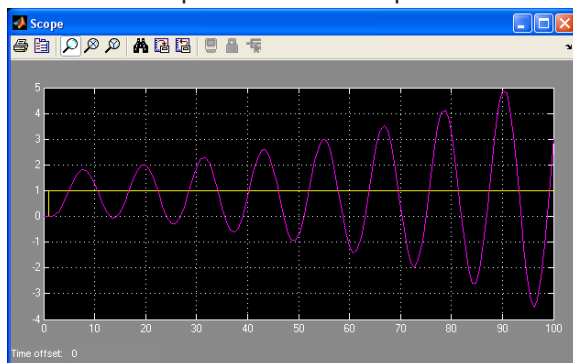
Postavimo vreme simulacije na 100 sekundi a zatim pređimo na eksperiment po Ziegler-Nicholsovoj metodi, tj. postavimo veliko vreme integracije, recimo **10 000**, a vreme diferencijacije na nulu, dok ćemo pojačanje menjati dok na izlazu sistema ne postignemo stabilne oscilacije.



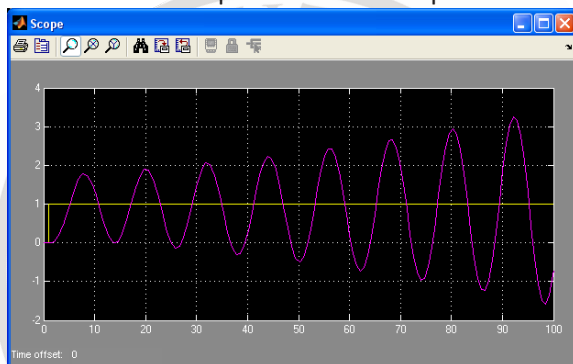
Za pojačanje $K=5$ na bloku Scope imamo sledeći prikaz:



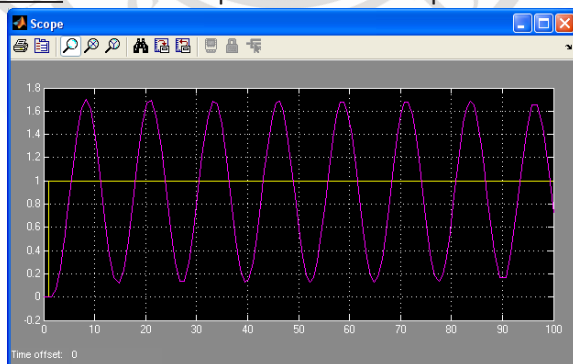
Za pojačanje $K = 2$ na bloku Scope imamo sledeći prikaz:



Za pojačanje $K = 1.88$ na bloku Scope imamo sledeći prikaz:



Za pojačanje $K = 1.66$ na bloku Scope imamo sledeći prikaz:



Sa ovim pojačanjem postignute su neprigušene oscilacije. Perioda oscilovanja se određuje sa dijagrama i približno je oko 13 sekundi. Prema ZN jednačinama izračunavamo sledeće vrednosti parametara regulatora:

$$K_p = 0.53 * K_0 \text{ à}$$

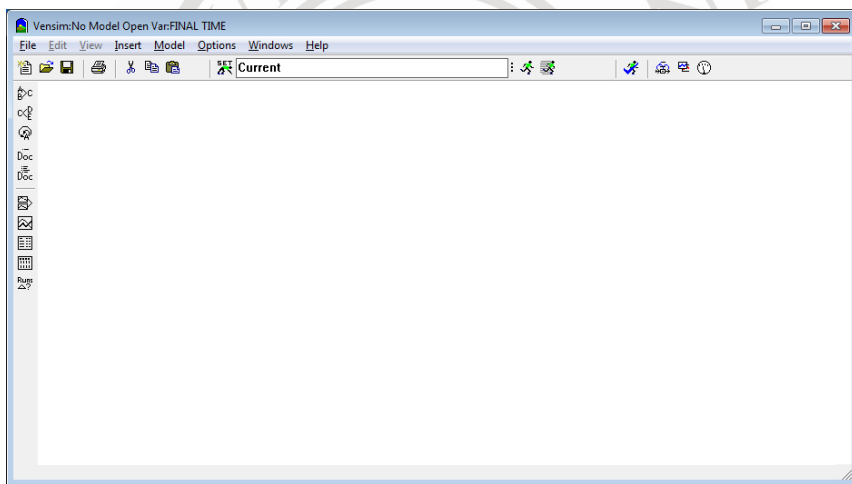
8. UVOD U SIMULACIONI PROGRAM VENSIM PLE

Softver *Vensim PLE* firme **Ventana Systems, Inc.** koristi se za modelovanje dinamičkih sistema. Dinamički sistemi su obično vrlo kompleksni zato što sadrže mnoštvo komponenti koje su povezane raznorodnim relacijama.

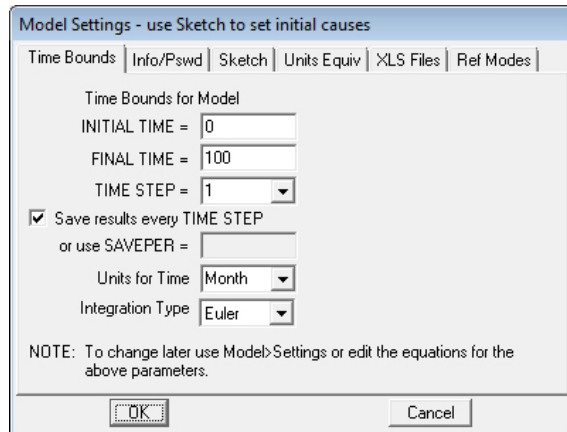
Kao primer razmatraćemo neograničeni rast populacije. U primeru brzina promene populacije jednaka je 10% broja individua u populaciji a inicijalna populacija broji 100 individua. Prema tome, imamo sledeću **diferencijalnu jednačinu** ili jednačinu koja sadrži izvod:

$$\frac{dP}{dt} = 0.1 \cdot P, \quad P_0 = 100$$

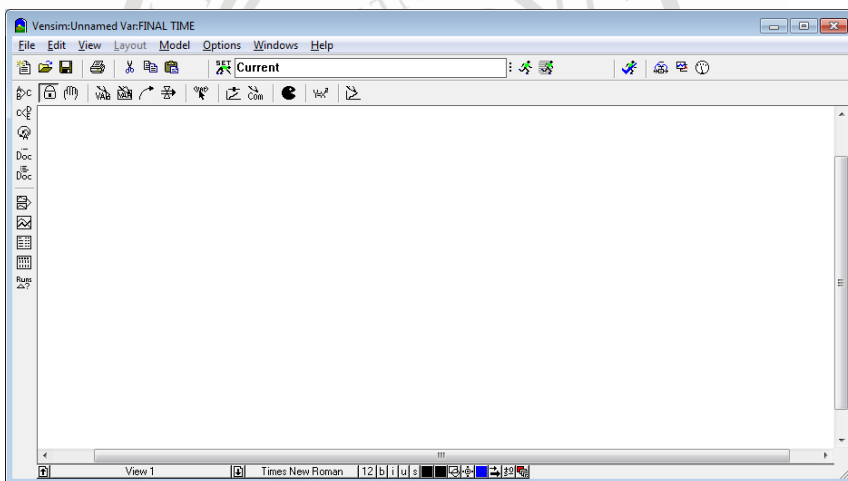
Dvo-klikom miša na radnoj površini računara na ikonu *Vensim PLE* pokrenimo softver za simulaciju.



U meniju **File** izaberimo opciju **New** da bi kreirali novi model. Na ekranu se pojavljuje okvir za dijalog **Model Settings**. Uobičajeno je kod kreiranja novog modela da se promene ponuđene vrednosti u ovom prozoru. Međutim, mi se sada samo upoznajemo sa osnovnim svojstvima ovog programa tako da ćemo prihvatiti ponuđene vrednosti klikom miša na dugme **OK**.



Posle ove aktivnosti interfejs programa sadrži radnu oblast *Build Window* a između ostalog dopunjen je i paletom alatki (ispod naslovne trake i glavne palete programa) koja sadrži vrlo bitne alatke/blokovske za gradnju simulacionog modela. Sada možemo da pristupimo kreiranju dijagrama modela sa jednačinama.

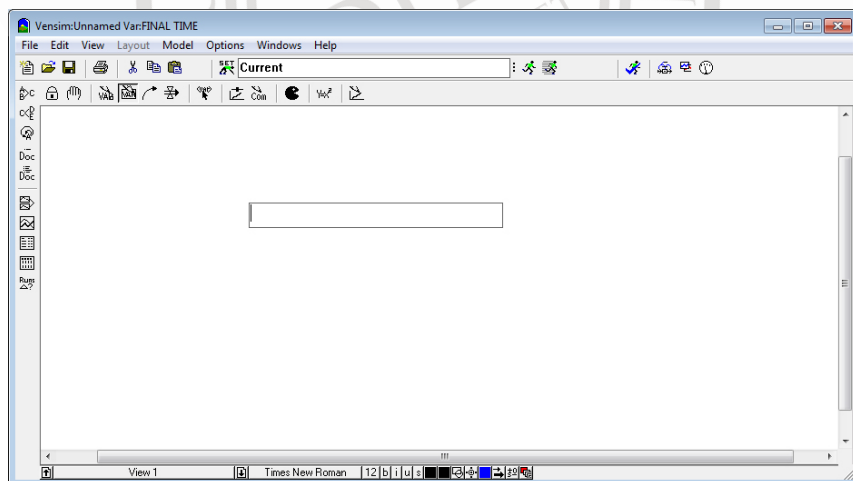


U sledećoj tabeli navedeni su osnovni gradivni blokovi programa *Vensim PLE*.

Gradivni blok	Ikona	Značenje
Box Variable ili Stock		imenica, nešto što akumulira (skladišti); stanje/nivo
Rate ili Flow		glagol, aktivnost koja menja veličinu skladišta (stanja/nivoa)
Variable ili Converter		transformiše, smešta jednačinu ili konstantu, ne akumulira
Arrow ili Connector		prenosi ulaze i informaciju

U *Vensim PLE* terminologiji **Box Variable** ili **Stock** je imenica i predstavlja skladište. Neki primeri skladišta su populacija, radioaktivnost, koncentracija enzima, novac itd. U svakom od tih primera veličine skladišta daju nam snimak sistema.

Izaberimo mišem alatku **Box Variable** i kliknimo mišem u radnu oblast da postavimo objekat skladišta. Na ekranu se pojavljuje okvir za upis teksta, tj. naziv skladišta.



Upišimo naziv skladišta – *population* i pritisnimo taster Enter.

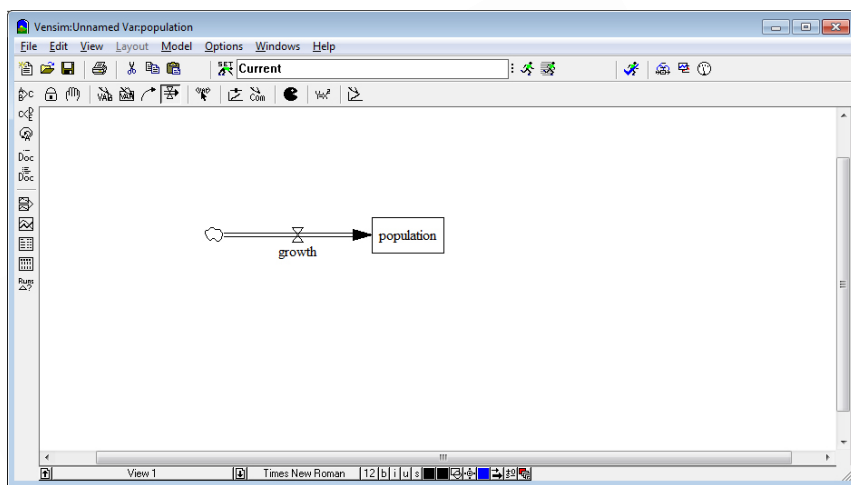
Ukoliko kasnije budemo želeli da promenimo naziv skladišta biće potrebno da izaberemo alatku **Box Variable** i kliknemo mišem na željeno skladište i upišemo novi naziv.

Savet: Selektovanjem alatke **Hand** i klikom miša na skladište možemo promeniti lokaciju skladišta na radnoj oblasti.

U meniju **File** selektujemo opciju **Save** (tasterska prečica Ctrl-S) ili kliknimo mišem na ikonu **Save** na glavnoj paleti alatki. Dodelimo naziv koji ćemo lako pamtit, recimo “*Vensim Tutorial 1*”.

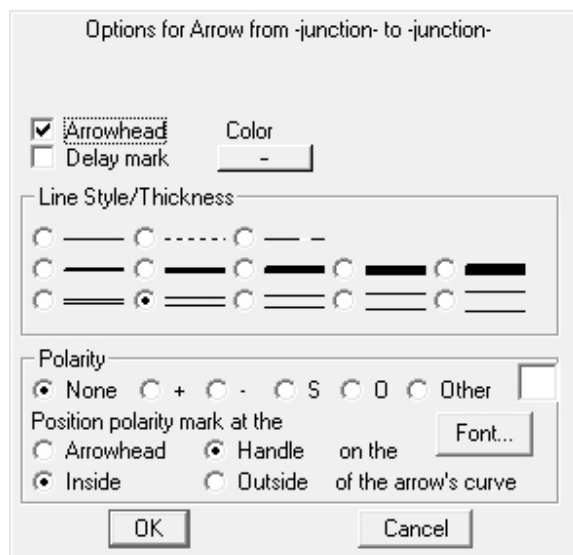
Dok je skladište imenica u jeziku *Vensim*, tok (**Rate** ili **Flow**) je glagol. Tok je aktivnost koja menja veličinu skladišta. Neki primeri takvih aktivnosti su rađanje populacije, opadanje radioaktivnosti itd. Ikona **Rate** predstavlja usmereni cevovod sa ventilom.

Na paleti alatki mišem izaberimo alatku **Rate**, kliknimo mišem u radnoj oblasti na izvesnoj udaljenosti levo od skladišta a zatim bez povlačenja miša kliknimo na skladište tako da se pojavi pravougaonik u koji upisujemo *growth* i konačno pritisnimo taster Enter. Uočavamo da tok počinje sa leve strane oblakom, koji je u ovom primeru izvor.

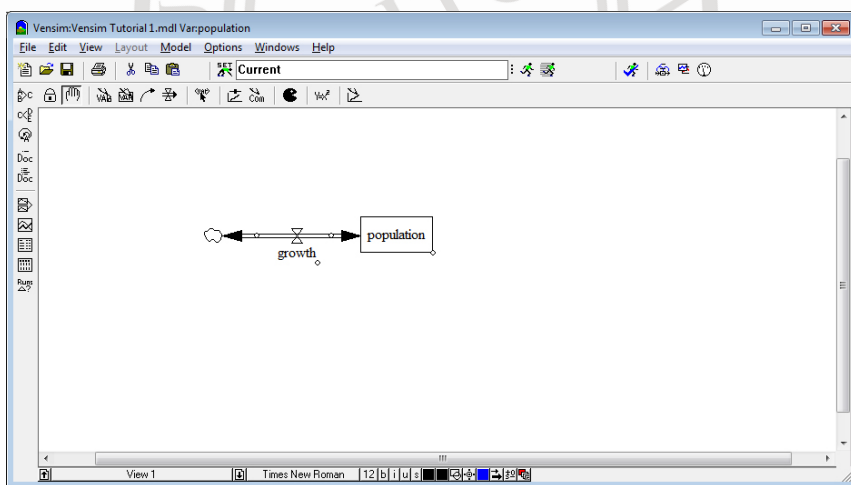


Napomena: Povucite mišem skladište *population* udesno. Šta se dešava sa dijagramom? Ukoliko se desi da se prilikom pomeranja skladišta ne pomera strelica toka, već se pojavljuje novi oblak, potrebno je obrisati tok i kreirati drugi koji dodiruje skladište.

Možemo da promenimo smer toka da ne bude samo u skladište već i u i iz skladišta tako da populacija može da se povećava ili smanjuje. Prema tome, možemo da promenimo da tok ne bude jednosmeran, već dvosmeran. Da bi to postigli selektujemo prvo alatku **Hand**, zatim postavimo pokazivač miša neposredno kod levog kružića na toku (levo od reči *growth*), kada pokazivač miša promeni oblik pritisnimo desni taster na mišu. Na ekranu se pojavljuje kontekstni meni za podešavanje opcija strelica.



Potvrdimo polje kod opcije **Arrowhead** i kliknimo mišem na dugme **OK**. U našem slučaju tok postaje dvosmeran što predstavlja mogućnost dodavanja u populaciju i uklanjanja iz populacije.

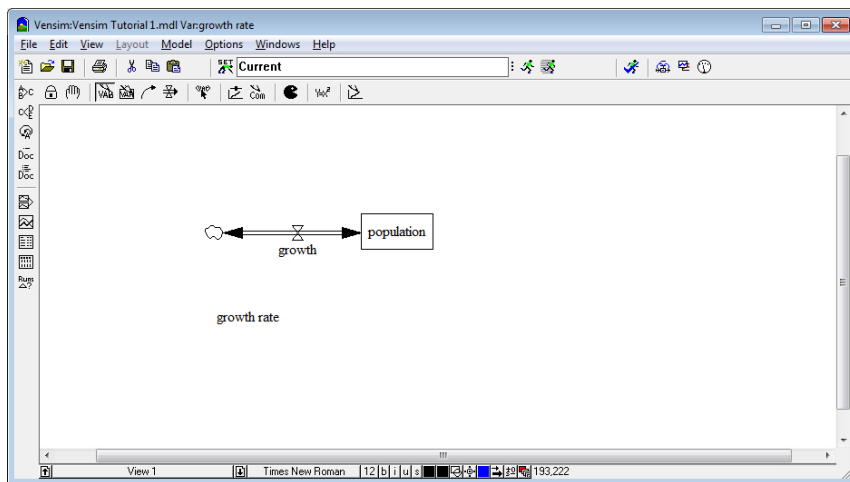


Napomena: Prethodno stanje toka može se vratiti tako što se dok je izabrana alatka **Hand** uradi desni klik mišem u ovom slučaju na levi kružić na toku i u kontekstnom meniju isključi opcija **Arrowhead**.

Aktivnost možemo izmeniti/modifikovati korišćenjem **Variable** ili **Convertera**. Varijabla (promenljiva) može da drži jednačinu ili konstantu. Na primer, za populacioni model varijabla može da drži konstantnu brzinu rasta (growth rate), recimo $10\% = 0.1$.

Kao primer za radioaktivno raspadanje, radioaktivna supstanca bizmut-210 raspada se u radioaktivnu supstancu polonijum-210. Sa A predstavimo količinu bizmuta-210 a sa B količinu polonijuma-210, odnos B/A je bitan u modelu raspadanja. Konvertor može da drži taj odnos.

Na paleti alati izaberimo alatku **Variable** (ikona na kojoj je “VAR” i olovka a nema okvira). Kliknimo mišem u radnoj oblasti ispod i levo od naziva toka – *growth*. Dodelimo konvertoru naziv *growth rate*. Razmaci (beline) su dozvoljeni u nazivu.

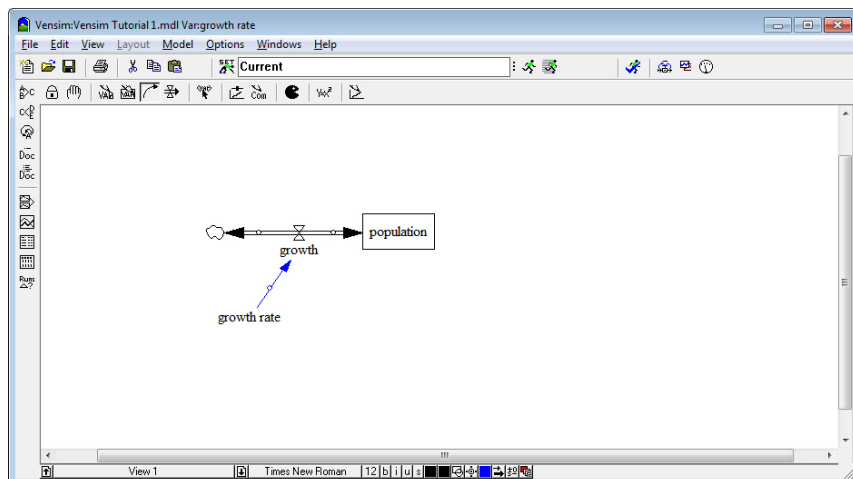


Dijagram posle postavljanja konvertora *growth rate*

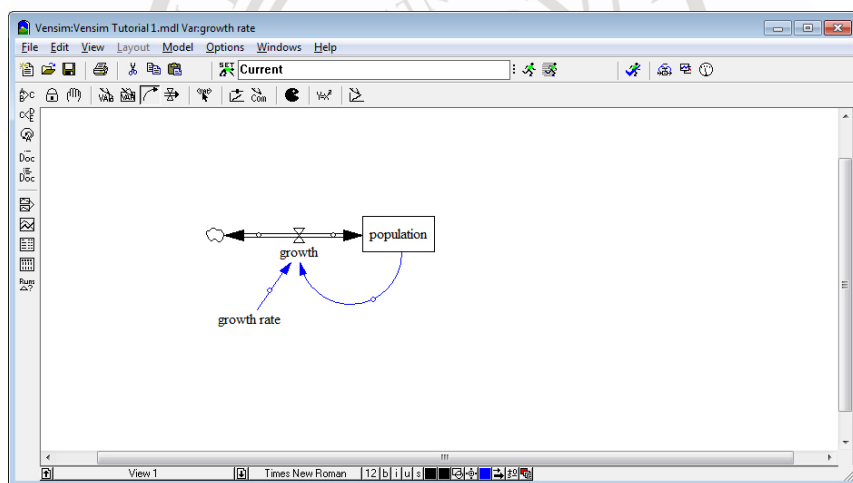
Strelica (**Arrow** ili **Connector**) prenosi ulaz ili izlaz. Na primer, u populacionom modelu strelica može da prenese vrednost brzine rasta od varijable (konvertora) *growth rate* do toka *growth*.

U modelu radioaktivnog raspadanja, strelice (konektori) od skladišta bizmut-210 (A) i od skladišta polonijum-210 (B) do konvertora za odnos B/A prenose respektivne iznose radioaktivnosti koje će konvertor koristiti.

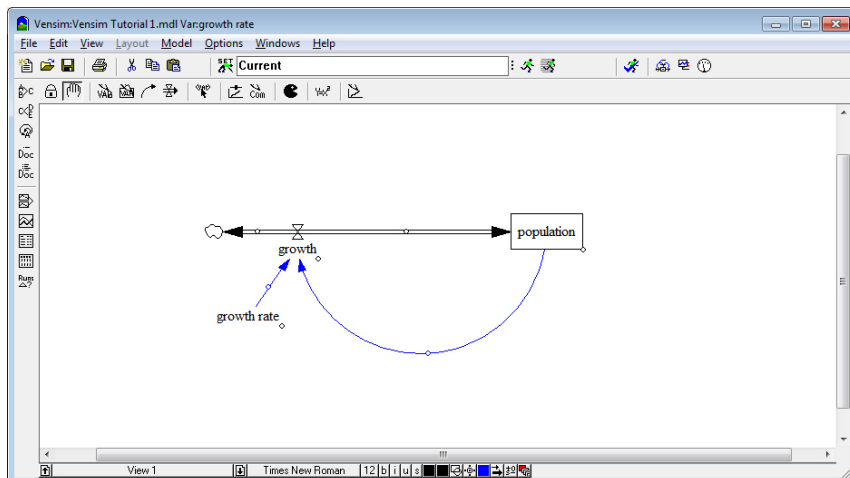
U populacionom modelu, i brzina rasta (*growth rate*) i tekuća populacija utiču na tekući rast. Na primer, ako je brzina rasta viša, isto je i sa rastom. Pored toga, veća populacija izložena je većoj promeni u populaciji. Mi pokazujemo te relacije povezivanjem konvertora (varijable) *growth rate* i skladišta *population* sa tokom *growth*. Po selektovanju ikone konektora kliknimo mišem na *growth rate* a zatim na *growth*. Rezultujući dijagram je prikazan na sledećoj slici.



Ukoliko nismo selektovali drugu alatku, možemo ponovo da koristimo istu alatku. Povežimo *population* sa *growth* tako što ćemo mišem kliknuti na *population*, kliknuti ispod i između *population* i *growth* da bi kreirali luk, a zatim *growth*. Mali kružići pokazuju sidrišta (ankere) koje možemo povlačiti da bi kreirali lukove. Snimimo model.



Napomena: Ukoliko je selektovana alatka **Hand**, klikom miša na mali kružić na strelici možemo ga pomerati i deformisati oblik strelice. Takođe, ako istom alatkom povučemo skladište (*population*) videćemo da se luk (konekcija) prilagođava oblikom ali se veza zadržava.

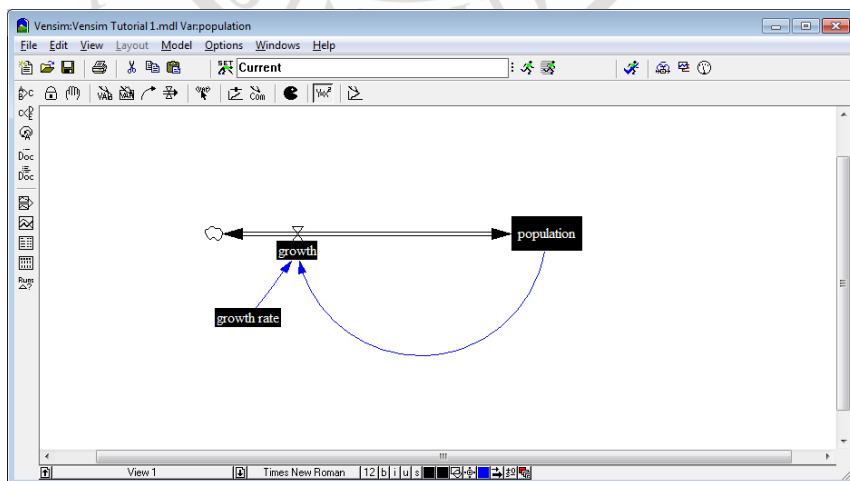


Za uklanjanje komponente sa dijagrama koristi se alatka **Delete** (treća sa desne strane na paleti alatki) ili alatka **Cut** sa glavne palete programa. Primenom tastera Delete komponenta se ne eliminiše u potpunosti iz modela.

Kada sa alatkom **Delete** uklonimo neku stavku, proces eliminiše stavku i sve povezane konektore i tokove. Model se može vratiti u prethodni oblik aktiviranjem komande **Undo** menija **Edit** ili zatvaranjem dokumenta bez snimanja i ponovnim otvaranjem dokumenta.

Jednačine i početne vrednosti

Spremni smo da upišemo jednačine i inicijalne vrednosti. Da bi definisali inicijalnu populaciju kliknimo mišem na alatku **Equations**, koja je pretposlednja ikona na paleti alatki. Tri stavke koje imaju vrednosti ili jednačine (*population*, *growth* i *growth rate*) postaju zاتمnjene.



Kliknimo mišem na varijablu *population* i na ekranu se pojavljuje okvir za dijalog **Editing equation for – population**.

Za inicijalnu populaciju od 100 bakterija upišimo 100 u tekst polje **Initial Value**. Kliknimo mišem na dugme **Check Syntax** u donjem levom delu okvira za dijalog. *Vensim* na proveru sintakse odgovara porukom *Equation OK* koja se prikazuje u polju **Errors**.

Padajući meni **Units** nudi više alternativa, uključujući **Dimensionless** (bezdimenzionalnost), koja se koristi kada vrednost nema jedinice. Kako odgovarajuća jedinica za *population* nije na listi, upisujemo *bacteria* u odgovarajuće polje.

Polja u okviru za dijalog pokazuju da je *population* integral (**INTEG**) od *growth* i da je inicijalna vrednost za *population* 100 bakterija. Prema tome, *Vensim* nam saopštava sledeće:

$$P = \int_a^b growth \cdot dt$$

gde je vreme (*t*) u rasponu od *a* do *b*.

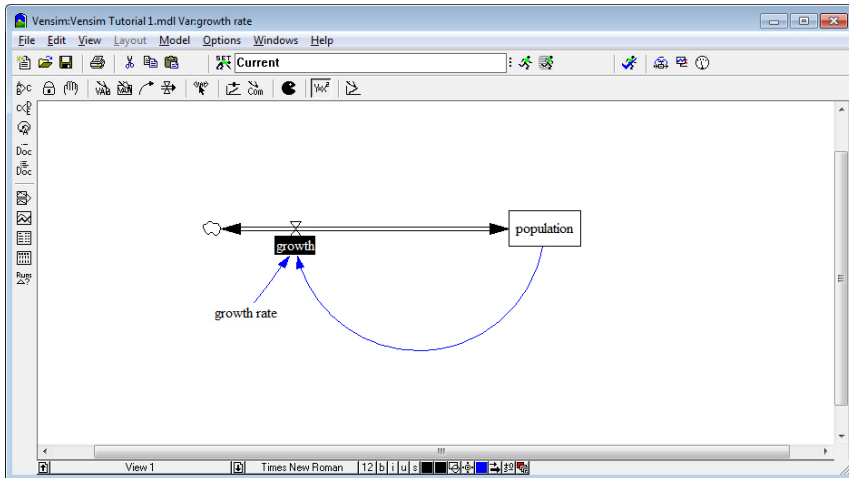
Da bi koristili *Vensim* nije neophodno da razumemo integraljenje. Navedeni iskaz (formula) je ekvivalentna sledećem:

$$\begin{aligned} (\text{new population}) &= (\text{old population}) + (\text{change in population}) \\ &= (\text{old population}) + growth * dt \\ &= (\text{old population}) + (\text{growth over 1 unit}) * (\text{length of time step}) \end{aligned}$$

Da bi kompletirali ulaz za *population*, kliknimo mišem na dugme **OK** u donjem levom delu okvira za dijalog.

Napomena: Da bi postavili *growth rate* kao 10% = 0.1, prvo kliknimo mišem na varijablu (konvertor) sa alatom **Equations**.

Upišimo *0.1* u pripadajuće tekst polje. U polje **Units** upišimo *1/Hour* a zatim kliknimo mišem na dugme **OK**. Uočavamo da po unosu vrednosti za brzinu rasta i inicijalnu populaciju elementi dijagrama nisu više zatamnjeni.

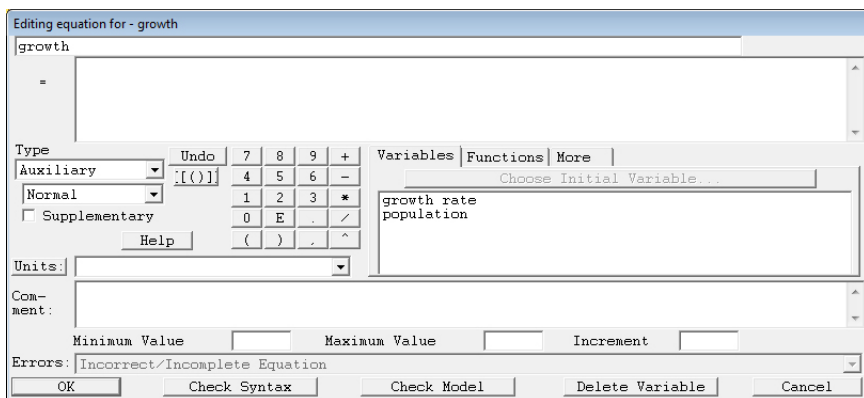


U praksi često se koristi zamenjivanje razmaka sa podvlakom u tekstu da bi se izbegla zabuna kod naziva komponenti. Prema tome, *growth rate* je ekvivalentno sa *growth_rate* u ovom tekstu.

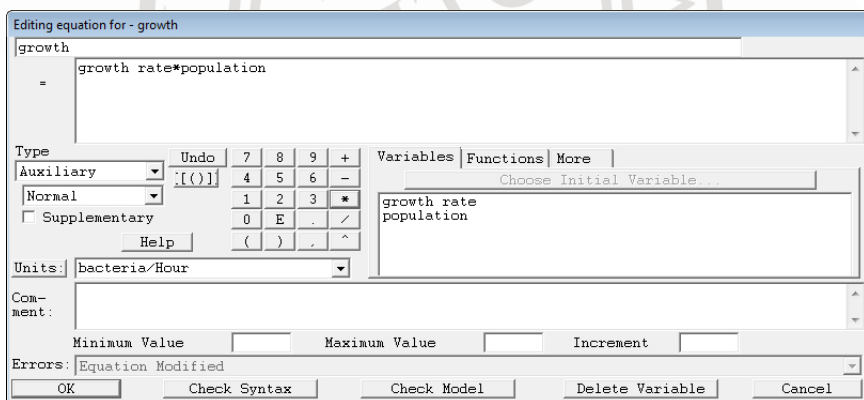
Za razliku od *growth_rate*, tok *growth* nije konstanta; međutim rast populacije se menja sa vremenom kako se populacija menja. U našem primeru, u svakom trenutku, brzina promene populacije, ili *growth*, je 10% (*growth_rate*) tekuće populacije (*population*). U matematičkoj terminologiji, trenutna brzina promene populacije je vremenski izvod populacije, stoga imamo sledeću formulu:

$$\begin{aligned}\frac{d(\text{population})}{dt} &= \text{growth_rate} \cdot \text{population} \\ &= 0.1 \cdot \text{population}\end{aligned}$$

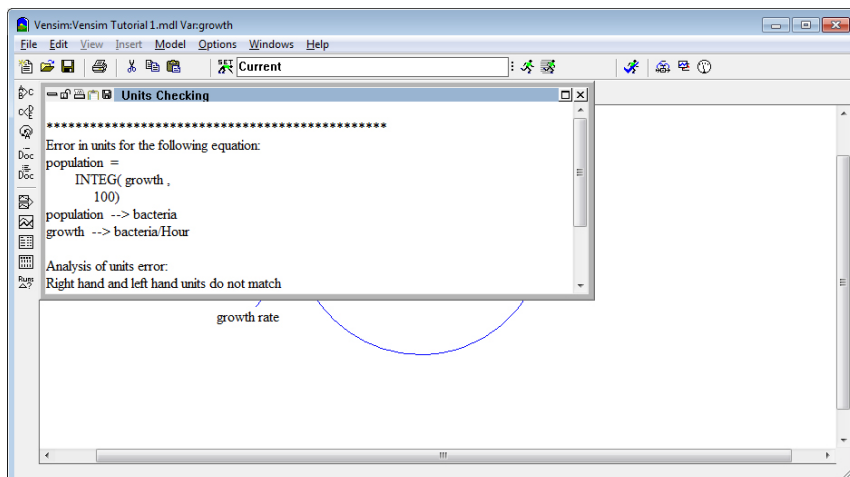
Klikom miša na *growth* pojavljuje se okvir za dijalog prikazan na sledećoj slici.



Napomena: Podmeni **Choose Initial Variable ...** navodi stavke koje imaju konektore sa *growth*, odnosno *population* i *growth_rate*. Te varijable su uključene u formulu za *growth*. U našem primeru, ova trenutna brzina promene populacije je $(0.1)(\text{population})$ bakterija po satu. Koristeći * za množenje i klikom miša na odgovarajuće varijable u podmeniju upisujemo formulu za *growth*. Za jedinice upisujemo *bacteria/Hour*. Na kraju, kliknimo mišem na dugme **OK**.



Sada možemo da verifikujemo da li su naše jedinice konzistentne selektovanjem opcije **Units Check** u meniju **Model**. Nažalost, *Vensim* pokazuje da imamo grešku. Program daje analizu prikazanu na sledećoj slici.



Error in units for the following equation:

population =

```
INTEG( growth,
      100)
```

population --> bacteria

growth --> bacteria/Hour

Analysis of units error:

Right hand and left hand units do not match

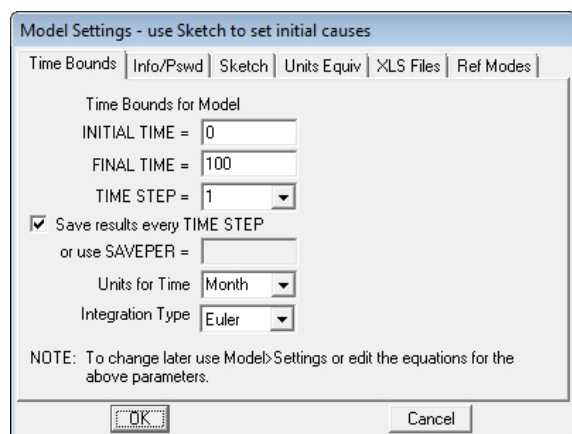
population

Has units: **bacteria**

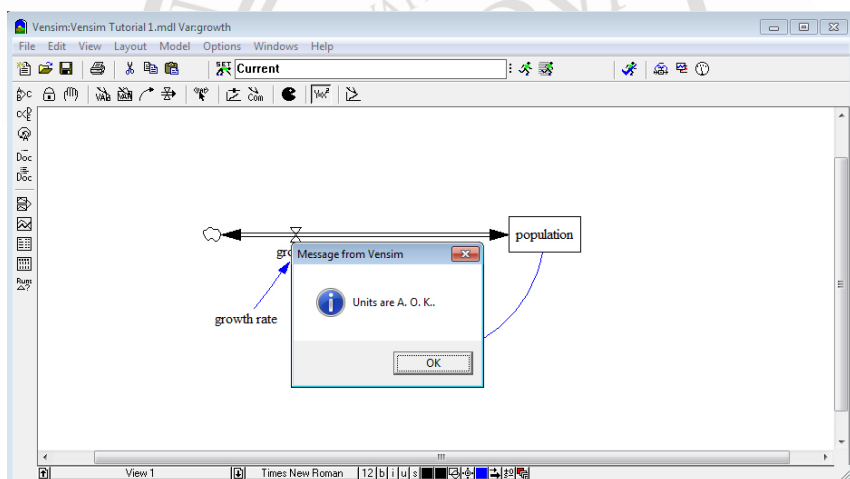
```
INTEG( growth,
      100)
```

Has units: **Month*bacteria/Hour**

Postoji nekonzistentnost (podebljani termini) vremenskih jedinica između *Hour* i *Month*. Problem je nastao zato što *Vensim* kao podrazumevanu jedinicu za vreme koristi *Month*. Da bi to prevazišli u meniju **Model** izaberimo opciju **Settings ...**. U okviru za dijalog **Model Settings** promenimo vrednost stavke **Units for Time** u vrednost *Hours*. Ova promena koriguje nekonzistentnost, stoga da bi zatvorili okvir za dijalog kliknimo mišem na dugme **OK**.

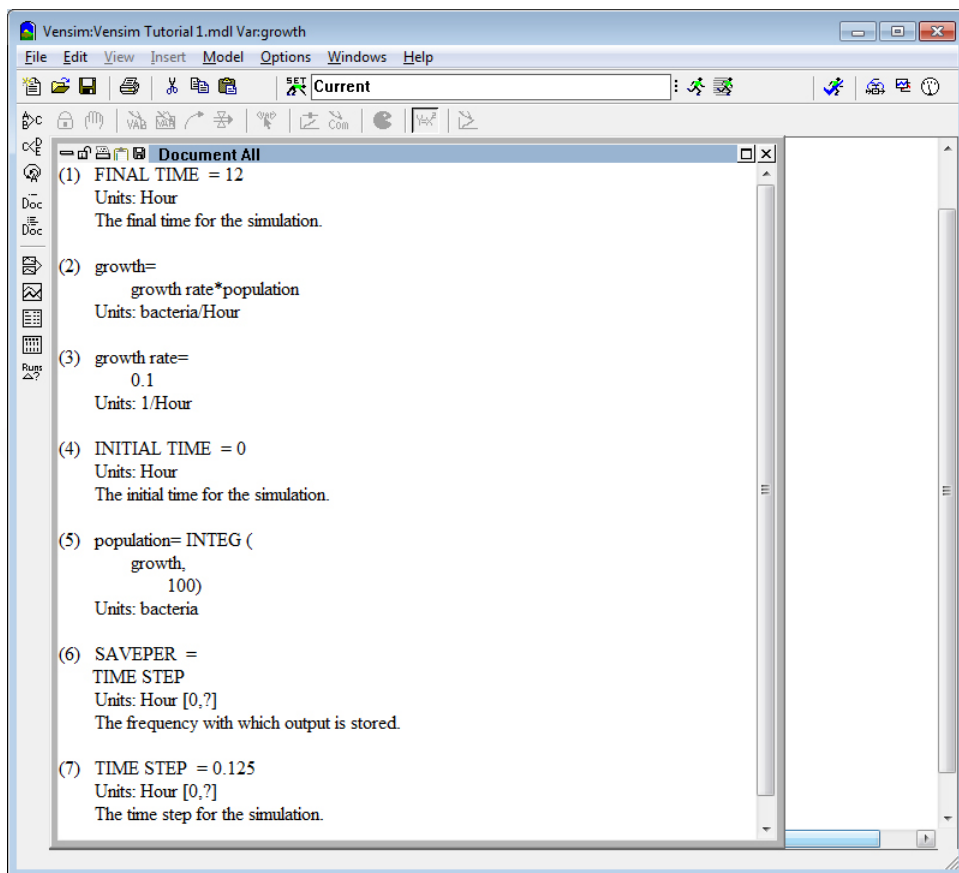


Posle promene jedinica uradimo ponovo proveru jedinica. Program sada javlja da je sve korektno.



Za potrebe ovog primera promenimo dužinu trajanja simulacije i vremenski korak simulacije. Ponovo selektujemo opciju **Settings...** menija **Model**. U pripadajućem okviru za dijalog ostavimo **INITIAL TIME** na *0 Hours* ali promenimo **FINAL TIME** da bude *12 Hours* kako se simulacija ne bi izvršavala predugo. Takođe, u padajućem meniju **TIME STEP** selektujemo vrednost *0.125*. Prema tome, u simulaciji će se izvršavati izračunavanja za svakih 0.125 sata umesto svakog sata. Obično manji vremenski korak (**TIME STEP**) generiše preciznije rezultate ali utiče da simulacija duže traje. Po završenom podešavanju mišem kliknimo na dugme **OK**.

Klikom miša na ikonu **Document All** na paleti koja se nalazi na levoj bočnoj strani prozora programa na ekranu se prikazuje prozor sa formulama. Postavili smo vrednosti za *FINAL TIME* (12), *INITIAL TIME* (0), *TIME STEP* (0.125) i jedinice za vreme (Hour) u okviru za dijalog Model Settings. Koristeći alatku **Equations** postavili smo vrednost za *growth_rate* (0.1), pripadajuću jedinicu (*1/Hour*), inicijalnu vrednost za *population* (100), pripadajuću jedinicu (*bacteria*) i jednačinu za *growth* ($growth_rate * population$).

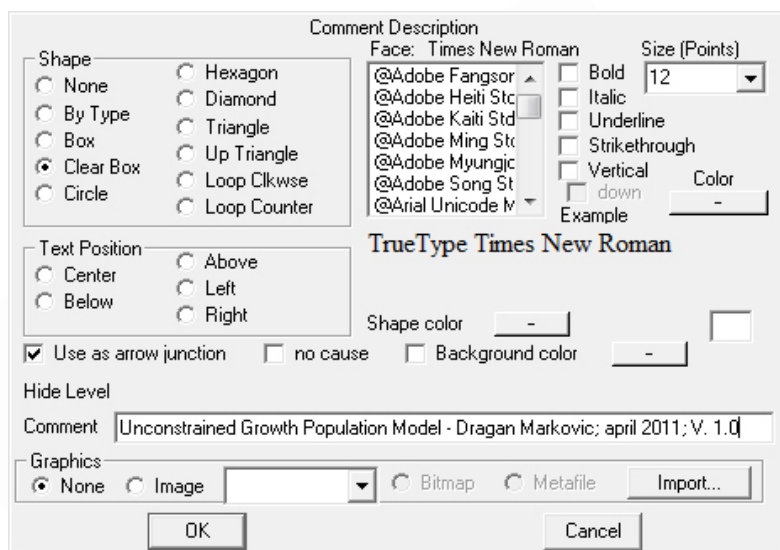
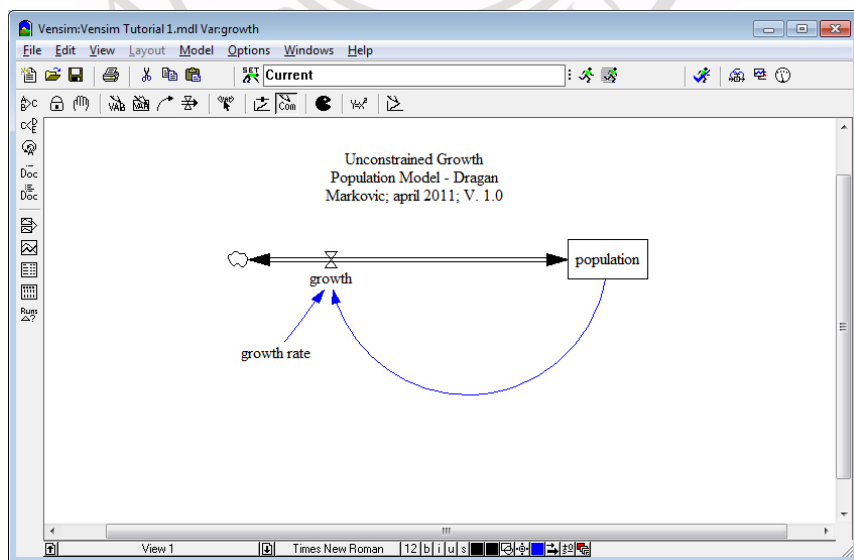


Dokumentovanje

Dokumentovanje svog rada je izuzetno važno. Mi želimo da drugi ljudi budu u mogućnosti da shvate model što je moguće lakše. Pored toga, možda ćemo vrlo brzo zaboraviti šta smo nameravali sa tim modelom. Takođe, može se desiti da imamo više sličnih verzija istog modela tako da nam je potrebno dodatno dokumentovanje kako bi mogli da ih razlikujemo. Ono što svakako želimo je da ne gubimo vreme ni mi ni neko drugi na razumevanje jednačina raznog nivoa koje se koriste u modelu.

Napomena: Da bi upisali komentar, kliknimo mišem na ikonu **Sketch Comment** (četrta sa desne strane na paleti alatki) da bi se pojavio okvir za dijalog **Comment Description** koji sadrži tekst polje **Comment** za tekst komentara.

Selektujemo alatku **Sketch Comment** i kliknimo mišem na sredinu gornjeg dela radne oblasti da postavimo polje za komentar. U okviru za dijalog **Comment Description**, u polje **Comment** upišimo recimo *Unconstrained Growth Population Model*, *svoje ime*, *datum*, *broj verzije* i *sažet opis modela*. Po završenom upisu kliknimo mišem na dugme **OK** da zatvorimo okvir za dijalog. Povucimo mišem mali kružić koji se nalazi na donjem desnom temenu polja komentara da bi promenili veličinu tekst polja komentara.

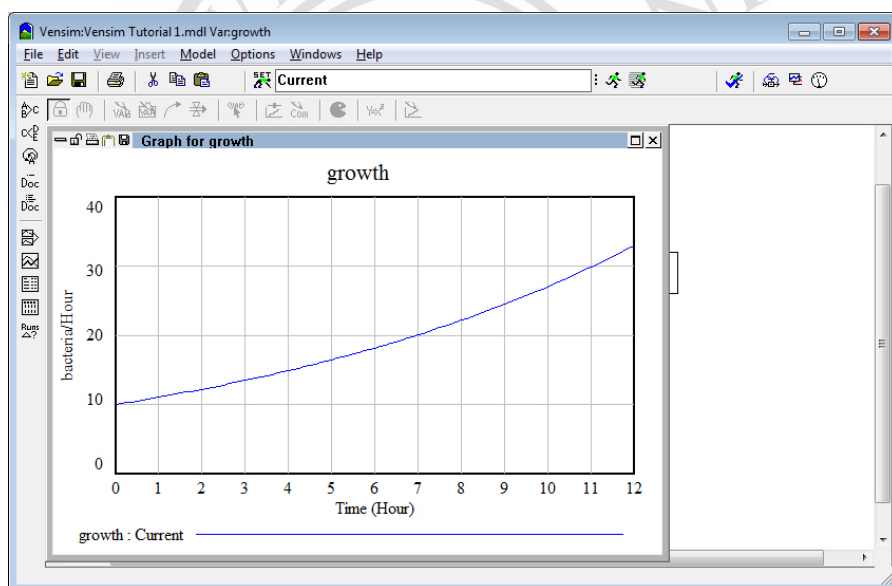


Pokretanje simulacije

Da bi generisali podatke simulacije koje možemo prikazivati bilo tabelarno bilo u obliku grafikona potrebno je da mišem kliknemo na ikonu **Run a Simulation**, koja se nalazi neposredno desno od tekst polja koje sadrži termin **Current**. Kada se simulacija završi naziv generisanog skupa podataka je **Current**, tj. naziv koji se nalazi u tekstualnom polju na glavnoj paleti programa. Po potrebi možemo promeniti parametre simulacije, kao što je na primer **TIME STEP** ili dužina simulacije i generisati drugi skup podataka koristeći neko drugo ime.

Grafikoni

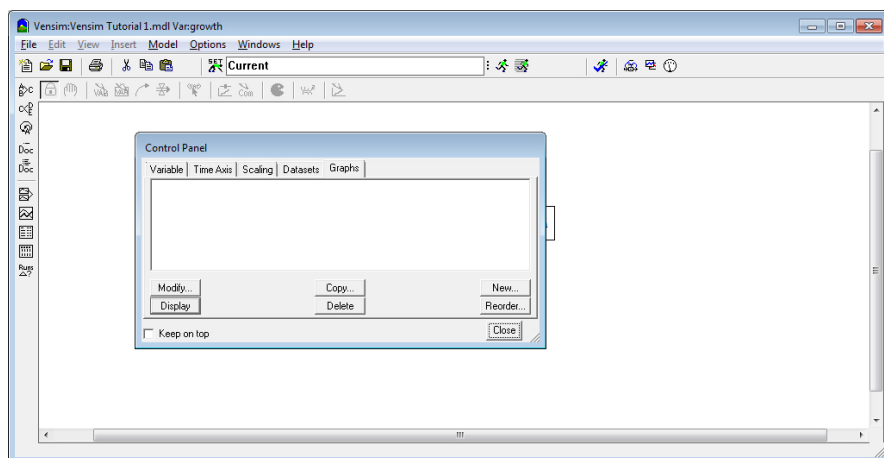
Ikona **Graph** pojavljuje se kao sličica sa dva mala grafikona u svojoj sredini a locirana je na bočnoj paleti alati. Posle klika mišem na tu ikonu odmah se pojavljuje prozor grafikona koji u ovom slučaju prikazuje vremensku zavisnost populacije. Položaj grafikona možemo podesiti prevlačenjem naslovne trake prozora grafikona a veličinu prozora povlačenjem mišem obodnih ivica prozora.



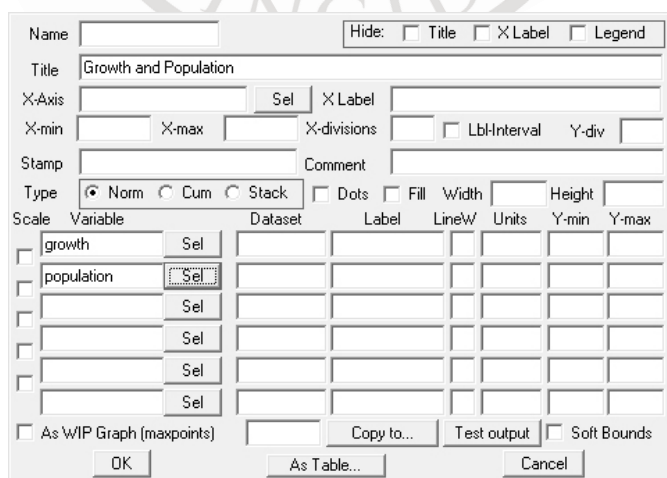
Grafikon zavisnosti populacije od vremena

Da bi ponovo prikazali izlaz, kao što su grafikon ili tabela koji nisu više vidljivi zato što smo kliknuli mišem izvan pripadajućeg prozora, kliknimo mišem na ikonu **Output Windows - show/circulate**, koja se nalazi kao druga sa desne strane na glavnoj paleti programa. Izlazni prozor možemo zatvoriti da ga opcija *show/circulate* ne može reaktivirati tako što ćemo kliknuti mišem na ikonu sa znakom X u gornjem desnom temenu prozora ili na ikonu sa horizontalnom crtom u gornjem levom temenu prozora.

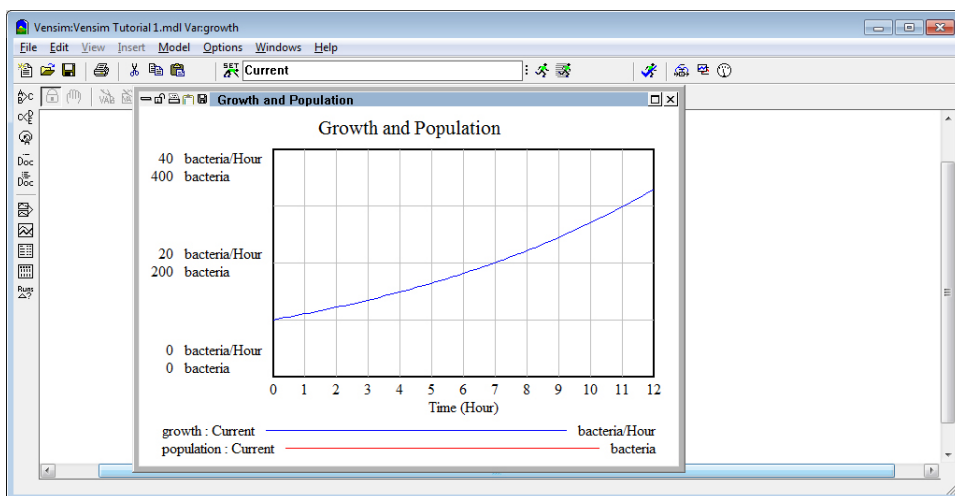
U praksi, često želimo da imamo više kontrole nad grafičkim prikazom, tj. želimo da dodelimo naziv grafikonu ili da imamo više krivih na istom dijagramu itd. Da bi na raspolaganju imali dodatne opcije uređivanja grafika kliknimo mišem na ikonu **Control Panel** koja je poslednja, na desnoj strani, na glavnoj paleti programa. Po aktiviranju ikone na ekranu se pojavljuje panel i možemo aktivirati njegovu karticu **Graphs** da definišemo prilagođenja.



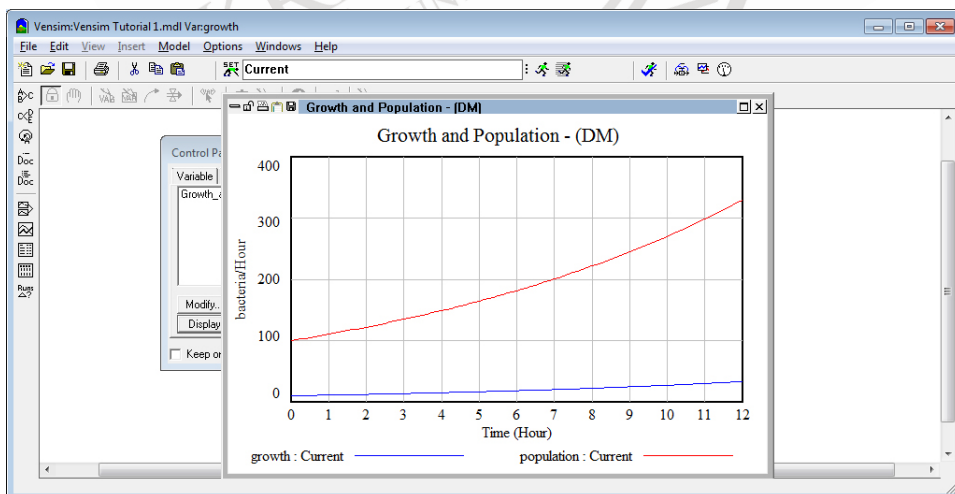
Da bi kreirali novi grafik kliknimo mišem na dugme **New** na desnoj strani panela. Na ekranu se prikazuje panel **Graphics** koji nam omogućava da definišemo niz karakteristika grafikona. Za naslov upišimo "Growth and Population" u tekst polje **Title**. Kod oznake **Variable** u donjem levom delu panela kliknimo na dugme **Sel** u prvom redu i iz padajućeg menija izaberimo *growth*. Ponovimo tu proceduru klikom miša na dugme **Sel** u drugom redu i izborom opcije *population*. Kliknimo mišem na dugme **OK** da se vratimo u kontrolni panel.



Da bi videli novi prikaz grafikona kliknimo mišem na dugme **Display** u donjem levom delu kontrolnog panela.

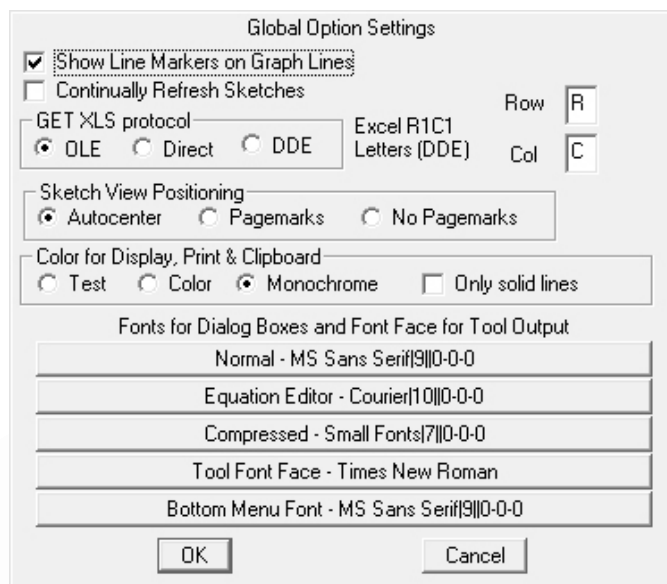
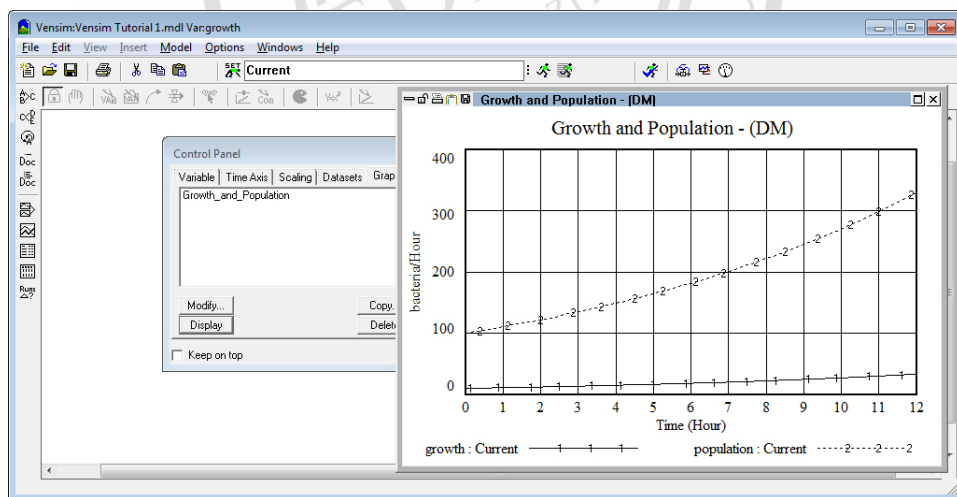


Kako grafikoni za *growth* i *population* imaju isti oblik a razmere (skale) su različite oni se prikazuju kao preklapljeni (jedan preko drugog). Da bi prikazali grafikone u istoj razmeri, vratimo se u kontrolni panel i kliknimo mišem na dugme **Modify**. U panelu **Graphics** potvrdimo mišem opcije **Scale** pored varijabli *growth* i *population*. Naslov grafikona možemo dopuniti svojim inicijalima tako što ćemo ih u zagradama upisati na kraju naslova. Kliknimo mišem na dugme **OK** a zatim na dugme **Display** u kontrolnom panelu.

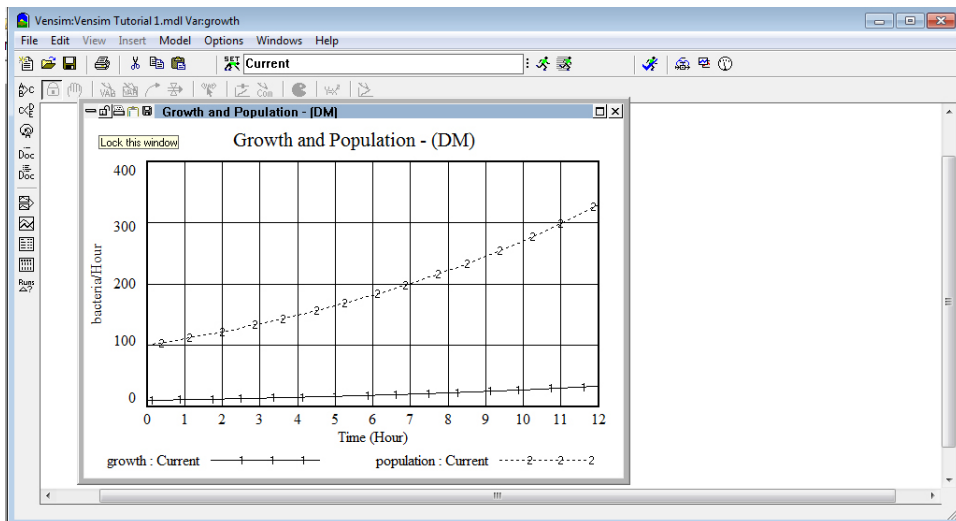


Inicijalno, *Vensim* koristi različite boje za razlikovanje grafikona. Za štampanje na crno-belom štampaču na raspolaganju je više alternativa kroz meni **Options** i njegovu stavku **Options**. Na ekranu se prikazuje panel **Global Option Settings**. Da bi *Vensim* numerisao

svaku krivu u ovom panelu potvrdimo opciju **Show Line Markers on Graph Lines**. U sredini panela u odeljku **Color for Display, Print & Clipboard** nalazi se druga korisna opcija - **Monochrome** za crno-bele grafikone i opcija **Only solid lines** za pune linije umesto tačkastih i crtičastih.

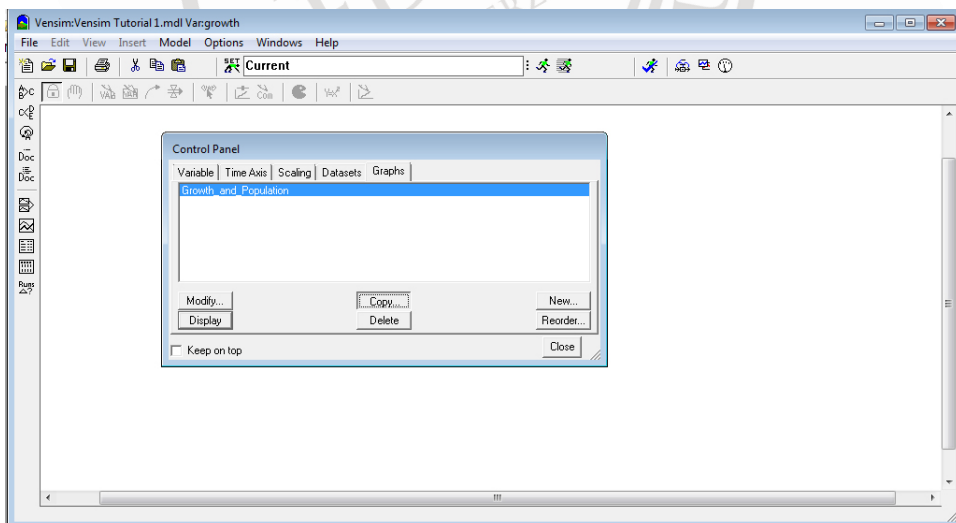


Da bi zadržali primenjena podešavanja na grafikonu kliknimo na ikonu **Lock this window** u gornjem levom uglu prozora grafikona. Do te ikone nalazi se ikona koja aktivira štampanje grafikona. Ikona **Export window contents** kopira tabelu u klipbord za postavljanje u drugu aplikaciju, kao što je program za obradu teksta.

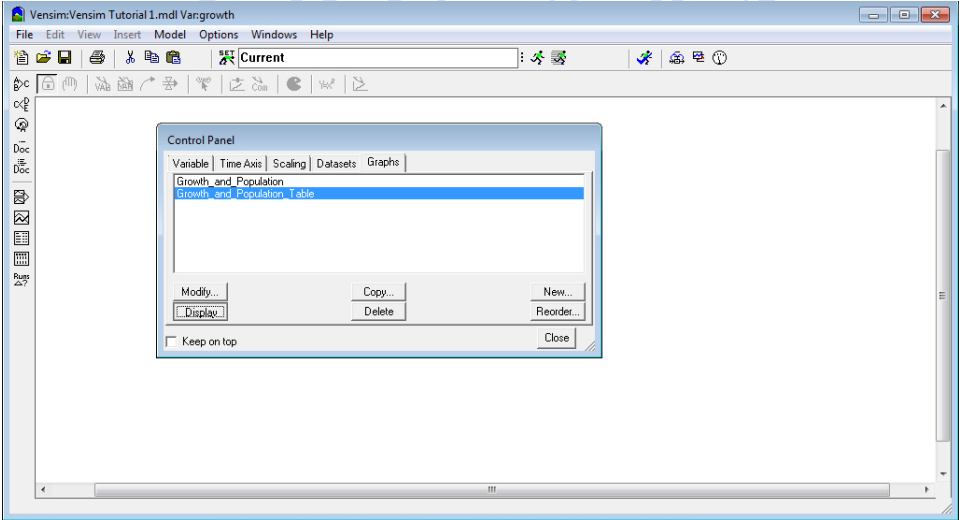


Tabele

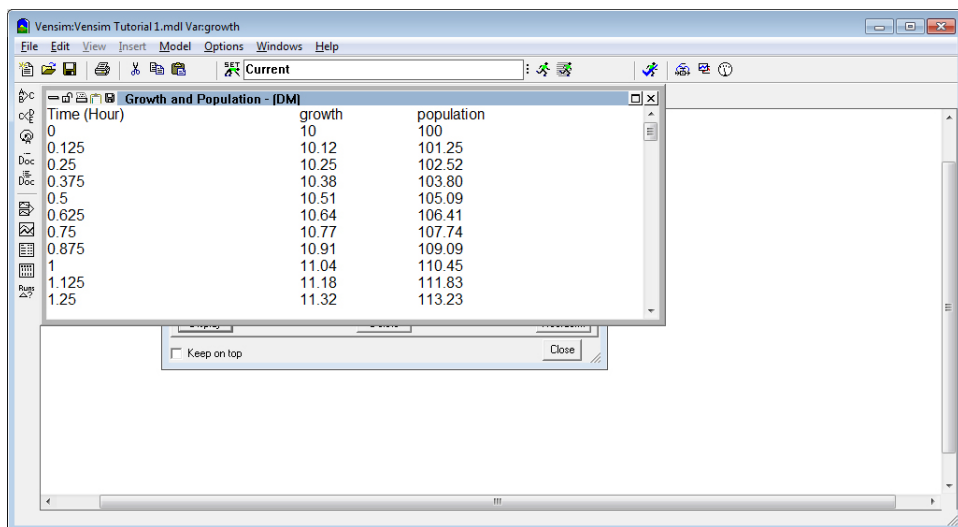
Tabelu ćemo generisati slično kao što smo to radili kod generisanja grafikona. Klikom miša na ikonu **Control Panel** aktiviramo prikaz tog panela. U panelu možemo kreirati novu tabelu ili izmeniti postojeću. U našem slučaju selektovaćemo naziv grafikona *Growth_and_Population* i kliknuti mišem na dugme *Copy*.



Rezultujući panel sada poseduje kopiju informacija za kreiranje grafikona *growth* i *population*. Kliknimo mišem na dugme **As Table...** u sredini donjeg dela panela da bi se na ekranu pojavio panel tabele. Pri vrhu panela u polje **Table Name** promenite naziv *Growth_and_Population_0* u *Growth_and_Population_Table*. Potvrdite mišem opciju **Running down** u odeljku **Time** da bi se vrednosti vremena pojavile u koloni tabele a zatim kliknimo mišem na dugme **OK**.



Na sledećoj slici dat je prikaz generisane tabele.



Ulazno/izlazni objekti

Inicijalne vrednosti i konstante u modelu možemo promeniti izborom alatke **Equations** i klikom miša na željenu varijablu. Za vizuelni prikaz kliknimo mišem na ikonu **Input Output Object**. Kliknimo mišem u oblast modela da postavimo objekat. Na ekranu se pojavljuje panel **Input Output Object settings**. Kliknimo mišem na dugme **Constant** i izaberimo **growth rate**. U odeljku **Slider Settings** kod parametra **Ranging from** upišimo vrednosti 0 i 0.2. Kliknimo mišem na dugme **OK** da zatvorimo panel.

Input Output Object settings

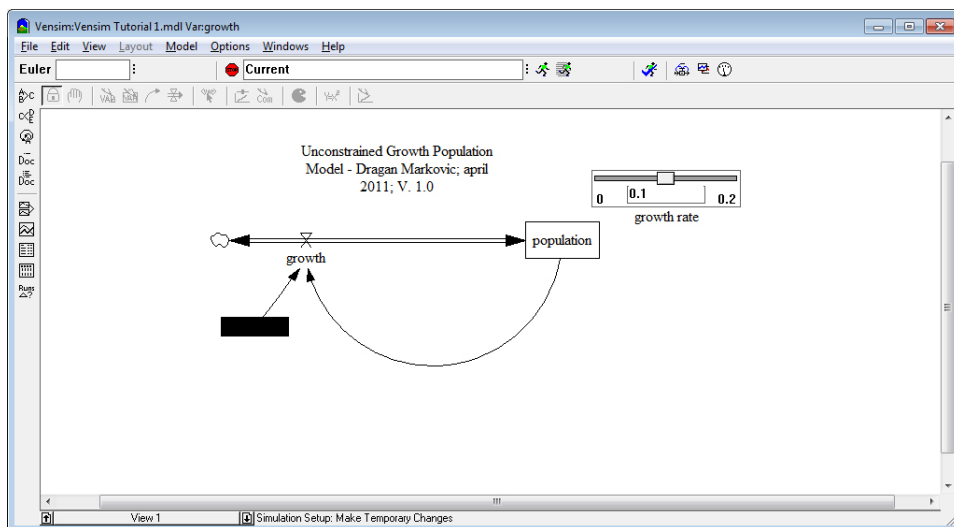
Object Type: ☒ Input Slider ☐ Output Workbench Tool ☐ Output Custom Graph

Variable name: Choose:

Slider Settings: Ranging from to with increment ☒ Label with varname

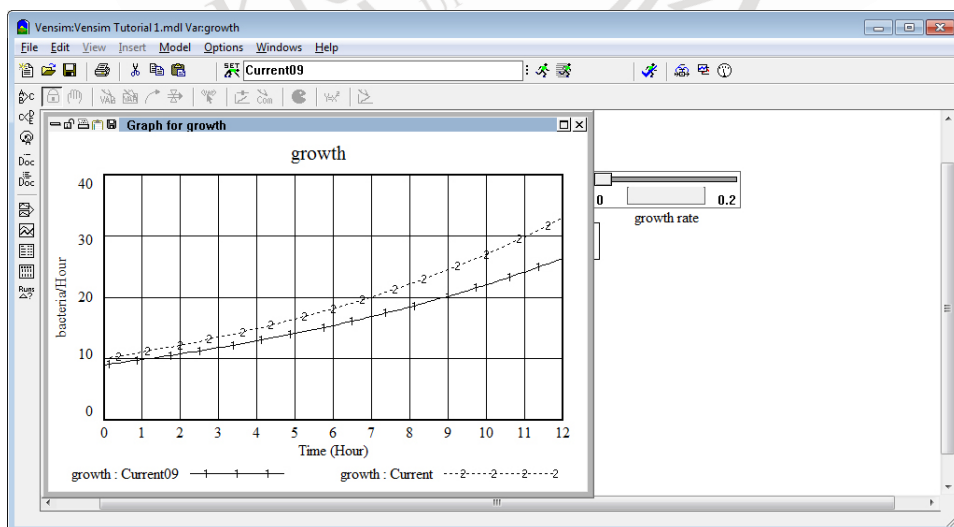
Custom Graph or Analysis Tool for Output:

Klikom miša na ikonu **SET** (nalazi se levo od skupa podataka **Current**) aktivira se grafička kontrola sa klizačem (engl. *slider*) u radnoj oblasti.



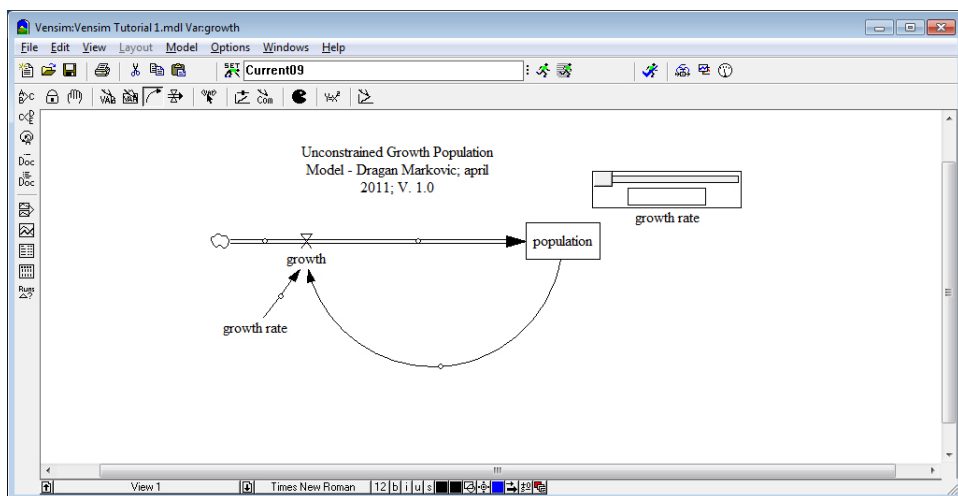
Upišite vrednost 0.09 kao alternativnu vrednost za *growth_rate* ili povucite mišem klizač na kontroli na tu vrednost. Promenite naziv skupa podataka u **Current09**. Kliknite mišem na ikonu **Run a Simulation** da bi pokrenuli simulaciju sa novom vrednošću za *growth_rate*. Da bi okončali postavku bez izvršavanja simulacije i vratili *growth_rate* na podrazumevanu vrednost kliknite mišem na ikonu **STOP** koja se nalazi levo od naziva skupa podataka.

Klikom miša na ikonu **Graph** dobijamo uporedni prikaz za obe vrednosti *growth_rate*.



Snimimo model i napustimo *Vensim PLE*.

Napomena: Da bi vizuelni prikaz u ovom modelu odgovarao razmatranoj logici modela treba tok vratiti na jednosmeran koristeći alatku **Hand** i opciju **Arrowhead** kontekstnog menija. Korektan vizuelni prikaz konačnog modela je prikazan na sledećoj slici.



Ugrađene funkcije i konstante

Vensim raspolaže nizom ugrađenih funkcija i konstanti koje se mogu koristiti u modelima. Kao što je ranije pomenuto, posle selektovanja mišem alatke **Equations** i klikom miša na neki od gradivnih blokova *Vensima* (Box Variable, Rate i Variable), možemo upisati tu komponentu u *Vensim* model. Klikom miša na jezičak kartice **Functions** u sredini prikazanog panela ostvarujemo pristup ugrađenim funkcijama. Jezičak **More** otkriva dodatne relacije i logičke operatore. Ovde ćemo razmotriti par tih funkcija i operatora koji omogućuju da efektivnije modelujemo većinu pojava iz stvarnosti. U sledećoj tabeli naveden je spisak većine *Vensim* funkcija, operatora i tipova njihovih formata i značenja. Neke od tih stavki se ne pojavljuju kao izbor u menijima **Functions** i **More**, ali ih korisnik može upisati u jednačinu. Dokumentacija koja se isporučuje uz sam programski paket daje opis svih tih funkcija i njihova svojstva.

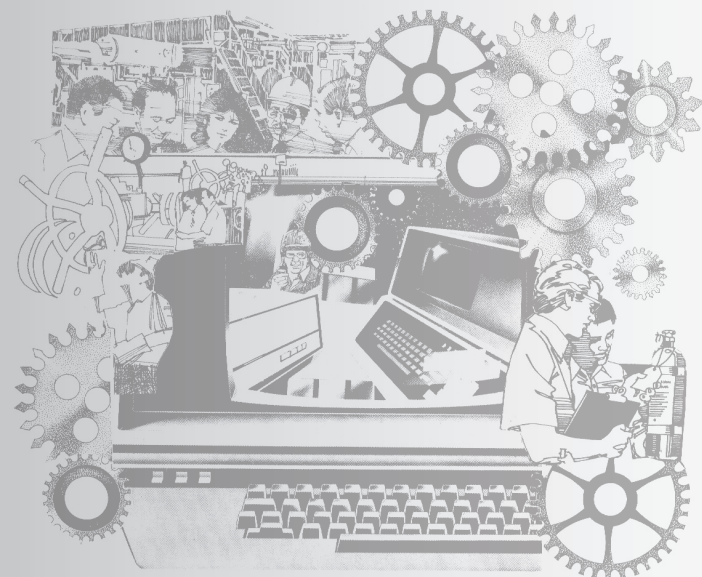
TABELA NEKE *VENSIM* FUNKCIJE, OPERATORI I TIPOVI

<i>ABS</i> (<i>n</i>)	$ n $, apsolutna vrednost od <i>n</i>
<i>I1 :AND: I2</i>	Logičko AND od <i>I1</i> i <i>I2</i> , gde <i>I1</i> i <i>I2</i> su logički izrazi
<i>COS</i> (<i>r</i>)	$\cos(r)$, gde je <i>r</i> ugao u radijanima
<i>EXP</i> (<i>x</i>)	e^x
<i>IF THEN ELSE</i> (<i>I</i> , <i>s1</i> , <i>s2</i>)	Ako je <i>I</i> tačno, <i>s1</i> se izvršava; ako je <i>I</i> netačno, <i>s2</i> se vraća
<i>Initial</i> (<i>x</i>)	Inicijalna vrednost od <i>x</i>
<i>INTEGER</i> (<i>x</i>)	Celobrojni deo od <i>x</i>
<i>LOG</i> (<i>x</i> , <i>b</i>)	$\log_b(x)$, logaritam od <i>x</i> za osnovu <i>b</i>
<i>LN</i> (<i>x</i>) $\ln(x)$	prirodni logaritam od <i>x</i>
<i>MAX</i> (<i>x1</i> , <i>x2</i>)	Maksimum od <i>x1</i> i <i>x2</i>
<i>MEAN</i> (<i>x1</i> , <i>x2</i>)	Aritmetička sredina od <i>x1</i> i <i>x2</i>
<i>MIN</i> (<i>x1</i> , <i>x2</i>)	Minimum od <i>x1</i> i <i>x2</i>
<i>MODULO</i> (<i>m</i> , <i>n</i>)	Celobrojni ostatak kada se <i>m</i> podeli sa <i>n</i>
<i>:NOT: I</i>	Logička negacija od <i>I</i> , gde je <i>I</i> logički izraz
<i>I1 :OR: I2</i>	Logičko OR (disjunkcija) ** od <i>I1</i> i <i>I2</i> , gde su <i>I1</i> i <i>I2</i> logički izrazi
<i>PULSE TRAIN</i> (<i>s</i> , <i>d</i> , <i>i</i> , <i>e</i>)	Povorka impulsa veličine 1.0 generisana u trenutku <i>s</i> , kod koje je razmak između impulsa <i>i</i> a koja se proteže sve do trenutka <i>e</i> ; svaki impuls je dužine trajanja <i>d</i>
<i>RANDOM 0 1</i> ()	Slučajni broj između 0 i 1
<i>SIN</i> (<i>r</i>)	$\sin(r)$, gde je <i>r</i> ugao u radijanima
<i>SQRT</i> (<i>x</i>)	Kvadratni koren od <i>x</i>
<i>STEP</i> (<i>h</i> , <i>t</i>)	Trenutna promena visine <i>h</i> u trenutku <i>t</i>
<i>TAN</i> (<i>a</i>)	$\tan(a)$, gde je <i>a</i> ugao u radijanima

TABELA NEKE *VENSIM* KONSTANTE

<i>FINAL TIME</i>	Završno vreme simulacije
<i>INITIAL TIME</i>	Startno vreme simulacije
<i>SAVEPER</i>	Vremenski interval između snimanja rezultata simulacije; treba da bude umnožak parametra <i>Time Step</i>
<i>Time</i>	Tekuće vreme simulacije modela
<i>TIME STEP</i>	Vremenski inkrement





LITERATURA

- L. Skyttner, "General Systems Theory: Problems, Perspectives, Practice", World Scientific Publishing Company, 2006.
- D. Radošević, "Osnove teorije sustava", Nakladni zavod Matice hrvatske, Zagreb, 2001.
- I. Mihajlović, Đ. Nikolić, A. Jovanović, "Osnove teorije sistema – inženjerski menadžment pristup", skripta, TF - Bor, 2009.
- Z. Jegeš, "Poglavlja iz upravljanja procesima", skripta, VTŠ Subotica, 2009.
- H. Babić, "Signali i sustavi, skripta", FER - Zagreb, 1996.
- B. Stojanović, "Teorija igara: elementi i primena", Službeni glasnik i Institut za evropske studije, 2005.
- Đ. Nadrljanski, M. Nadrljanski, "Kibernetika u obrazovanju", Učiteljski fakultet u Somboru, 2005.
- F. D. Petruzella, "Programmable Logic Controllers", 4th Edition, The McGraw-Hill Companies, Inc., 2011.
- S. Turajlić, "Programabilni logički kontroleri", skripta, ETF – Beograd, 2011.
- S. Holzner, "Differential Equations for Dummies", Wiley Publishing, Inc., 2008.
- B. Hahn, D. T. Valentine, "Essential MATLAB For Engineers and Scientists", Third Edition, Elsevier, 2007.
- R. J. Gran, "Numerical Computing with Simulink, Volume I *Creating Simulations*", SIAM, 2007.
- S. T. Karris, "Introduction to Simulink with Engineering Applications", Orchard Publications, 2006.
- A. B. Shiflet, G. W. Shiflet, "Introduction to Computational Science: Modeling and Simulation for the Sciences", Princeton University Press, 2006.

Na osnovu člana 23. stav 2. tačka 7. Zakona o porezu na dodatu vrednost („Službeni glasnik RS”, br. 84/04... i 61/07), Odlukom Senata Univerziteta Singidunum, Beograd, broj 260/07 od 8. juna 2007. godine, ova knjiga je odobrena kao osnovni udžbenik na Univerzitetu.

CIP - Каталогизација у публикацији
Народна библиотека Србије, Београд

519.876(075.8)

МАРКОВИЋ, Драган С., 1967-

Uvod u teoriju sistema / Dragan S.
Marković. - 1. izd. - Beograd : Univerzitet
Singidunum, 2011 (Loznica : Mladost grup). -
VI, 279 str. : graf. prikazi, tabele ; 25 cm

Tiraž 300. - Bibliografija: str. 279.

ISBN 978-86-7912-399-2

a) Теорија система
COBISS.SR-ID 188610316

© 2012.

Sva prava zadržana. Nijedan deo ove publikacije ne može biti reprodukovan u bilo kom vidu i putem bilo kog medija, u delovima ili celini bez prethodne pismene saglasnosti izdavača.