Mr Nenad Dodić,
dipl. inž.
Vojnotehnički institut VJ,
Beograd

SAVREMENI PRISTUP RAČUNARSKI PODRŽANOJ SIMULACIJI SERVOSISTEMA

UDC: 681.513.3.001.573:519.872

Rezime:

Problem računarske simulacije servosistema pristupa se sa stanovišta praktične i lako ostvarljive realizacije rada servosistema u virtuelnoj (računarskoj) stvarnosti. Postupak simulacije ostvaruje se korišćenjem koncepta stanja, numeričkih postupaka rešavanja diferencijalnih jedinačina (metoda Runge-Kutta, primena konačnih razlika) i blok-dijagrama. Daje se jednostavan i efikasan postupak podešavanja koraka integraljenja, kojim se postiže približno jednako rastojanje tačaka hodografa stanja i izbegava porast grešaka simulacije u slučaju brzih promena izvoda veličina stanja.

Ključne reči: simulacija, servosistem, metoda Runge-Kutta, konačne razlike, računar.

CONTEMPORARY APPROACH TO COMPUTER AIDED SIMULATION OF SERVOSYSTEMS

Summary:

The approach to the problem of computer aided simulation is based on practical and easy-to-use realization of the servosystem operation in virtual (computer) reality. The simulation procedure is performed using the concept of state, numerical methods to solve differential equations (the Runge-Kutta method, the application of finite differences) and block-diagrams. A simple and efficient procedure to tune the integration step is given, so that the approximate equidistance of the state hodograph points is achieved and the growth of simulation errors, in case of fast changes of derivatives of state variables, is avoided.

Key words: simulation, servosystem, Runge-Kutta method, finite differences, computer.

Uvod

Ukoliko je poznat matematički model servosistema u celini, odnosno ukoliko su poznati matematički modeli svih njegovih podsistema, kao i brojčane vrednosti parametara koji se pojavljuju u modelima, može se pristupiti računarski podržanoj simulaciji rada servosistema. Ona predstavlja oponašanje procesa koji se dešavaju u servosistemu, pri dejstvu zadatih ulaza. Praktično, rad sistema se

ostvaruje u virtuelnoj (veštačkoj) stvarnosti, unutar računara – bez korišćenja stvarnih resursa servosistema.

Matematički gledano, računarski podržana simulacija sistema predstavlja rešavanje sistema (diferencijalnih) jednačina koje ga opisuju, za zadate ulaze i početne uslove. Ulazi sistema se zadaju kao funkcije vremena ili vremenske sekvence (serije). Ako su ulazi jednostavne funkcije (odskočna, impulsna, sinusna, linearna) jednačine sistema se u određe-

nim slučajevima mogu rešiti eksplicitno – u zatvorenom obliku. Rešenja su tada veličine stanja (i izlazi) sistema, kao funkcije vremena. U tom slučaju simulacija se svodi na jednostavno izračunavanje tih funkcija, za zadati skup diskretnih trenutaka vremena. Ovakvo rešavanje sistema diferencijalnih jednačina moguće je kada je on linearan, i u nekim (retkim) specijalnim slučajevima nelinearnih jednačina (npr. Rikatijske jednačine).

Ako sistem jednačina nije moguće rešiti u zatvorenom obliku, pomenuti način simulacije nije moguć. Tada se jednačine sistema moraju rešavati približno, nekim od mnogobrojnih numeričkih postupaka.

Matematički opis servosistema

Radi primene koncepta simulacije servosistema, on se, odnosno njegovi podsystemi, opisuje kao skup diferencijalnih jednačina prvog reda:

$$\frac{dx_i}{dt} = f_i(t, x_1, x_2, \dots, x_n, u_1, u_2, \dots),$$

$$i = 1, 2, \dots, n \quad (1)$$

gde su:

x_1, x_2 – veličina stanja [1],
 u_1, u_2 – ulazi servosistema.

Jednačine (1) nazivaju se jednačine stanja. Izlazi servosistema mogu biti neke od veličina stanja. U opštem slučaju, oni su funkcije veličina stanja:

$$y_i = g_i(t, x_1, x_2, \dots, x_n, u_1, u_2, \dots),$$

$$i = 1, 2, \dots, n \quad (2)$$

Poseban slučaj je linearni stacionarni analogni jednostruko prenosni servosistem [1]. On se najčešće sreće u praksi ili, bolje rečeno, njemu se najčešće pribegava zbog jednostavnosti. U slučaju ova-

kvog sistema, skupovi jednačina (1), (2) dobijaju sledeće oblike:

$$\frac{dx_i}{dt} = f_i = a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n + b_iu, \quad i = 1, 2, \dots, n \quad (3)$$

$$y = c_{11}x_1 + c_{12}x_2 + \dots + c_{1n}x_n + du \quad (4)$$

Servosistem, odnosno neki njegov podsystem, često se opisuje linearnom ulazno-izlaznom diferencijalnom jednačinom:

$$\frac{d^ny}{dt^n} + a_{n-1} \frac{d^{n-1}y}{dt^{n-1}} + \dots + a_0y =$$

$$b_0u + b_1 \frac{du}{dt} + \dots + b_m \frac{d^mu}{dt^m} \quad (5)$$

ili odgovarajućom prenosnom funkcijom:

$$\frac{Y(s)}{U(s)} = \frac{b_ms^m + \dots + b_0}{s^n + a_{n-1}s^{n-1} + \dots + a_0}$$

Da bi se ostvario koncept simulacije, potrebno je jednačinu (5) prevesti u sistem jednačina stanja (3), što se može ostvariti primenom sledećih relacija [1]:

$$x_1 = y - b_nu$$

$$x_2 = \frac{dx_1}{dt} + a_{n-1}y - b_{n-1}u$$

...

$$x_{n-1} = \frac{dx_{n-2}}{dt} + a_2y - b_2u$$

$$x_n = \frac{dx_{n-1}}{dt} + a_1y - b_1u$$

$$\frac{dx_n}{dt} = -a_0y + b_0x_1$$

Model (5) tada postaje:

$$\frac{dx_j}{dt} = -a_{n-j}x_1 + x_{j+1} + (b_{n-1} - a_{n-j}b_n)u,$$

$$j = 1, 2, \dots, n, x_{n+1} \equiv 0$$

$$y = x_1 + b_n u$$

gde je $b_j = 0$ za $j = m + 1 + 2, \dots, n$.

Numeričko rešavanje diferencijalnih jednačina stanja

Neka je servosistem (ili njegov pod-sistem) opisan sistemom jednačina (1), (2), a poznati početni uslovi $x_1(0)$, $x_2(0), \dots$, $dx_1(0)/dt$, $dx_2(0)/dt$, ..., kao i ulazi $u_1(t)$, $u_2(t), \dots$ u toku datog trajanja simulacije. Treba rešiti sistem diferencijalnih jednačina (1). U nastavku je objašnjeno rešavanje primenom metode Runge-Kutta 4. reda, koja je verovatno najefikasnija, i primenom konačnih razlika, što je svakako najjednostavniji i, čini se, najčešće korišćeni postupak.

Metoda Runge-Kutta

Jednačine (1) rešavaju se rekurentno, korišćenjem sledećih relacija [2]:

$$x_i(t) = x_i(t-h) + R_j, j = 1, 2, \dots, n \quad (6)$$

gde je h korak integraljenja i

$$R_j = \frac{1}{6} (A_{1j} + 2A_{2j} + 2A_{3j} + A_{4j}),$$

$$j = 1, 2, \dots, n$$

$$A_{1j} = hf_j(t-h, x_1(t-h), \dots, x_n(t-h), u_1(t-h), u_2(t-h), \dots), j = 1, 2, \dots, n$$

$$A_{2j} = hf_j\left(t - \frac{h}{2}, x_1(t-h) + \frac{A_{11}}{2}, \dots, x_n(t-h) + \frac{A_{1n}}{2}, u_1\left(t - \frac{h}{2}\right), \dots\right), j = 1, 2, \dots, n$$

$$A_{3j} = hf_j\left(t - \frac{h}{2}, x_1(t-h) + \frac{A_{21}}{2}, \dots, x_n(t-h) + \frac{A_{2n}}{2}, u_1\left(t - \frac{h}{2}\right), u_2\left(t - \frac{h}{2}\right), \dots\right), j = 1, 2, \dots, n$$

$$A_{4j} = hf_j\left(t, x_1(t-h) + \frac{A_{31}}{2}, \dots, x_n(t-h) + \frac{A_{3n}}{2}, u_1(t), u_2(t), \dots\right), j = 1, 2, \dots, n$$

$$A_{4j} = hf_j(t, x_1(t-h) + \frac{A_{31}}{2}, \dots, x_n(t-h) + \frac{A_{3n}}{2}, u_1(t), u_2(t), \dots), j = 1, 2, \dots, n$$

$$A_{4j} = hf_j(t, x_1(t-h) + \frac{A_{31}}{2}, \dots, x_n(t-h) + \frac{A_{3n}}{2}, u_1(t), u_2(t), \dots), j = 1, 2, \dots, n$$

Ako ulazi $u_1, u_2, \dots$ u trenutku t_x , gde t_x može biti t , $t - h/2$ ili $t - h$, nisu poznati, koristi se interpolacija. Na primer, korišćenjem linearne interpolacije dobija se:

$$u_i(t - t_x) = u_i(t_2) - \frac{t_2 - t_x}{t_2 - t_1} [u_i(t_2) - u_i(t_1)], i = 1, 2, \dots \quad (7)$$

gde je $t_1 < t_x < t_2$, a ulazi $u_i(t_1)$ i $u_i(t_2)$ su poznati.

Korak integraljenja h može biti konstantan. Ako se želi jednako rastojanje tačaka, može se koristiti sledeći jednostavan, ali efikasan postupak podešavanja koraka:

$$h \leftarrow h \{ [x_1(t) - x_1(t-h)]^2 + \dots + [x_n(t) - x_n(t-h)]^2 \}^{1/2} / d_z \quad (8)$$

d_z je zadato rastojanje između tačaka hodografa stanja u prostoru stanja. Simbol „ $\leftarrow$ “ znači „postaje“.

Algoritam za rešavanje sistema (1), kada su date sledeće sekvence ulaznih veličina: $\{u_1(0), u_1(\Delta t), u_2(2\Delta t), \dots\}$, $\{u_2(0), u_2(\Delta t), u_2(2\Delta t), \dots\}, \dots$, početni uslovi $t = 0, x_1(0), x_2(0), \dots, dx_1(0)/dt, dx_2(0)/dt, \dots, h = \Delta t$, krajnji trenutak t_{kraj} i kada se vrši adaptacija koraka integraljenja, je:

1. Računanje počinje od trenutka $t = h$.
2. Ulazi $u_1(t - h/2), \dots, u_2(t - h/2), \dots$ izračunavaju se interpolacijom, npr. korišćenjem (7).
3. Veličine stanja $x_1(t), x_2(t), \dots$ izračunavaju se korišćenjem (6), a izlazi $y_1(t), y_2(t), \dots$ korišćenjem (2).
4. Korak integraljenja se menja korišćenjem (8).
5. Povećava se vreme $t \leftarrow t + h$. Ako je $t \geq t_{kraj}$ ide se na tačku 8.
6. Ulazi $u_1(t), u_1(t - h), \dots, u_2(t), u_2(t - h), \dots$ se izračunavaju interpolacijom, npr. korišćenjem (7).
7. Ide se na tačku 2.
8. Kraj rada.

Ako se ne vrši adaptacija koraka integraljenja, 4. i 6. tačka se izbacuju.

Primena konačnih razlika

Izvodi u (1) zamenjuju se potonjim konačnim razlikama [3]:

$$dx_i(t)/dt = [x_i(t + h) - x_i(t)]/h, \\ i = 1, 2, \dots, n$$

Tada (1) dobija sledeći oblik:

$$x_i(t + h) = x_i(t) + hf_i(x_1(t), x_2(t), \dots, u_1(t), u_2(t), \dots), i = 1, 2, \dots, n \quad (9)$$

Rekurentnim izračunavanjem (9), polazeći od datih početnih uslova, rešava se sistem (1). Za rešavanje sistema diferencijalnih jednačina (1) može se koristiti dati algoritam, s tim što se izbacuje 2. tačka, a umesto relacija (6) koriste se relacije (9).

Primena blok-dijagrama

Kada je servosistem visokog reda, odnosno veoma složene strukture, da bi se postupak simulacije pojednostavio i da bi se smanjile greške simulacije, preporučljivo je servosistem „podeliti“ na nekoliko podsistema, odnosno predstaviti sistem u obliku blok-dijagrama. U njemu svaki blok predstavlja jedan podsistem.

Simulacija servosistema primenom blok-dijagrama objasniće se na primeru jednostruko prenosnog sistema automatske regulacije – slika 1.

Prikazani sistem je podeljen na četiri bloka: blok C – regulator, blok P – objekat upravljanja, blok F – povratna sprema i blok A – sabirač (komparator). Značenja veličina sa slike 1 su:

y_w – ulaz servosistema – zadata vrednost izlaza,

u_c, u_p, u_f – ulazi regulatora, objekta upravljanja i bloka povratne petlje, respektivno,

y_c, y_p, y_f – izlazi regulatora, objekta upravljanja i bloka povratne petlje, respektivno,

$x_{c1}, x_{c2}, \dots$ – veličine stanja regulatora,

$x_{p1}, x_{p2}, \dots$ – veličine stanja objekta upravljanja,

$x_{f1}, x_{f2}, \dots$ – veličine stanja bloka povratne petlje,

n_p – poremećaj,

n_f – šum merenja izlaza.

Blokovi C, P, i F definisani su jednačinama stanja, koje se mogu rešavati korak po korak, primenom jednog od opisanih postupaka. Ako se usvoji isti korak integraljenja h za sve blokove, onda on mora biti bar deset puta manji od vremenske konstante najbržeg podsistema (bloka). Korišćenje različitih koraka integraljenja je složenije, ali može uštedeti vreme računanja. Sabirač je definisan jednostavnom jednačinom: $u_c = y_w - y_f$, koju ne treba numerički rešavati.

Neka je ulazna sekvenca $\{y_w(0), y_w(\Delta t), y_w(2\Delta t), \dots\}$, i neka su početni uslovi poznati. Algoritam za simulaciju je sledeći:

1. Kada se koristi metoda Runge-Kutta, pri izračunavanju $u_c(t)$ veličina $y_f(t)$ nije poznata (jer još nije izračunata). Zato je treba predvideti na osnovu prethodno izračunatih vrednosti y_f , što se može postići nekim od postupaka ekstrapolacije. Očigledno, ovaj problem se ne javlja kada se za rešavanje diferencijalnih jednačina koriste kasnije razlike.

2. Kada se odredi $y_f(t)$, izračunava se $u_c(t) = y_w(t) - y_f(t)$.

3. Rešava se jednačina stanja regulatora za trenutak t i izračunava njegov izlaz $y_c(t)$.

4. Za $u_p(t) = y_c(t) + n_p(t)$ i trenutak t rešava se jednačina stanja objekta. Tako se dobija $y_p(t)$.

5. Za $u_f(t) = y_p(t) + n_f(t)$ i trenutak t rešava se jednačina stanja povratne petlje. Tako se dobija $y_f(k\Delta t)$.

6. Povećava se vreme $t \leftarrow t + h$ i ponavljaju koraci 1. do 6. sve do isteka zadatog trajanja simulacije.

Primer simulacije elektrohidrauličnog servosistema

Postupak simulacije biće ilustrovan na primeru elektrohidrauličnog servosistema, prikazanog na slici 2. Njegov blok-dijagram prikazan je na slici 3. Dinamika linearnih blokova opisana je prenosnim funkcijama, što je uobičajena praksa pri projektovanju i analizi servosistema. Prisutne nelinearnosti opisane su statičkim karakteristikama. To su: nelinearnost servorazvodnika, tipa neosetljivosti (mrtva zona) sa zasićenjem i nelinearnost opterećenog motora – suvo (Kulonovo)

trenje. Relevantne veličine su: zadata ($\dot{\theta}_w$) i ostvarena ($\dot{\theta}_m$) ugaona brzina motora, struja servorazvodnika (Δi), pomeraње klipnjače razvodnika (x_v), napon tahogeneratora (U_{tg}), signal greške (e) i napon proporcionalan zadatoj ugaonoj brzini (U_w). U_1 i U_2 su naponi u upravljačkoj elektronici (pomoćne veličine).

Da bi se primenio opisani postupak simulacije, dinamičke linearne blokove treba opisati odgovarajućim jednačinama stanja i izlaza:

– integralni uskladnik:

$$dx_{cl} / dt = -0,999x_{cl} + 0,0436e$$

$$U_2 = y_c = x_{cl} + 0,00191e$$

– nelinearnost servorazvodnika [4]:

$$Q_v = \begin{cases} 0,001 \operatorname{sgn}(x_v), & |x_v| \geq 0,0017 \\ g(x_v - 0,0001) \operatorname{sgn}(x_v), & 0,0001 \leq |x_v| \leq 0,0017 \\ 0, & |x_v| \leq 0,0001 \end{cases}$$

$$g(x) = 0,65x - 7812,9x^3$$

$$\operatorname{sgn}(x) = \begin{cases} 1, & x > 0 \\ 0, & x = 0 \\ -1, & x < 0 \end{cases}$$

– nelinearnost motora:

$$\dot{\theta}_m = (\dot{\theta}_{m1} + 0,5) \operatorname{sgn}(\dot{\theta}_{m1})$$

– ostale statičke veze:

$$U_w = 0,095 \dot{\theta}_w, U_1 = 0,00498U_w$$

$$e = U_w - U_{tg}, \Delta i = U_1 + U_2$$

$$U_{tg} = 0,095 \dot{\theta}_m$$

Simulacija dinamičkih blokova predstavlja rešavanje datih diferencijalnih jed-

– 1. stepen servorazvodnika:

$$dx_{sl} / dt = -166,67x_{sl} + 1,00\Delta i$$

$$x_v = y_s = x_{sl}$$

– 2. stepen servorazvodnika sa motorom:

$$dx_{m1} / dt = x_{m2}$$

$$dx_{m2} / dt = -208,8x_{m2} - 13456x_{m1} + 8,0280 \cdot 10^8 Q_v$$

$$\dot{\theta}_{m1} = y_m = x_{m1}$$

Statički blokovi opisani su običnim algebarskim jednačinama:

načina, dok simulacija statičkih blokova predstavlja prosto izračunavanje datih algebarskih jednačina. Simulacija celog servosistema vrši se korišćenjem blok-dijagrama i jednog od datih postupaka rešavanja sistema diferencijalnih jednačina. Ulaz sistema simuliran je kao signal koji svakih 0,8 sekundi slučajno menja svoju vrednost (slika 4). Usvojen je korak integraljenja $h = 0,001$ s. Za rešavanje diferencijalnih jednačina korišćena je metoda Runge-Kutta. Simulirani izlaz servosistema prikazan je na slici 5.

Sl. 4 – Simulirani ulaz servosistema

Sl. 5 – Simulirani izlaz servosistema

Umesto zaključka

Priprema matematičkog modela je prvi bitan problem koji treba rešiti. Ukoliko je sistem visokog reda, treba ga podeliti na blokove što nižih redova (najbolje prvog ili drugog reda), pri čemu oni

mogu, ali ne moraju da imaju fizički smisao. Ukoliko je sistem nelinearan, potrebno je, kad god je to moguće, nelinearne delove (nelinearnosti) izdvojiti u posebne blokove, tako da, po mogućnosti, ni jedan blok ne bude opisan nelinearnim diferencijalnim jednačinama.

Kada se koristi metoda Runge-Kutta, ova pravila mogu se ublažiti. Tako se ova metoda može primeniti na jednačinu stanja celog sistema, osim kada je njegov red vrlo visok (recimo deseti ili viši). U tom slučaju sistem treba razbiti na blokove (tj. formirati blok-dijagram), s tim što ti blokovi mogu biti složeniji, odnosno viših redova nego u slučaju primene konačnih razlika.

U opštem slučaju, uprošćavanje matematičkog modela sistema i rešavanje diferencijalnih jednačina nižih redova povećava numeričku stabilnost i preciznost postupka simulacije. S druge strane, formiranje vrlo složenog blok-dijagrama, s namerom da blokovi budu jednostavni, postupak simulacije može učiniti komplikovanim i nepreglednim. Ako se zbog toga radi s manjim brojem složenijih blokova, treba primeniti metodu Runge-Kutta.

Sledeći problem je izbor koraka integraljenja. Ukoliko se piše sopstveni program za simulaciju, a sistem je izdelfjen na blokove, može se opredeliti da svaki blok ima drugačiji korak integraljenja, čime se, verovatno, može postići optimalna simulacija, naročito ako je brzina izvršavanja presudna. Komercijalni računarski programi obično zahtevaju da se korak integraljenja izabere za sistem u celini. U tom slučaju korak integraljenja treba prilagoditi najbržem bloku, to jest bloku sa najvećom dinamikom. Postoji čitav niz pravila za izbor koraka integraljenja, počevši od uslova konvergencije

numeričkog postupka koji se primenjuje, pa do Šenonove teoreme koja vodi računa o tome da između dve tačke računanja nema značajnih promena relevantnih veličina. Sva ta pravila se teško primenjuju u praksi.

Postupak za izbor koraka integraljenja je jednostavan, praktičan i pouzdan. On se sprovodi kroz pilot-probe računarske simulacije, pre nego što se počne s njenim praktičnim korišćenjem. Izabere se prizvoljan korak integraljenja i izvrši simulacija. Ukoliko se dobije nestabilno rešenje (obično s oscilacijama čiji je period istog reda veličine kao izabrani korak integraljenja), a pretpostavlja se da je simulirani sistem stabilan, drastično se smanjuje korak integraljenja. Ako je simulacija stabilna, ona se ponavlja s dvostruko manjim korakom integraljenja. Ukoliko se rezultat ove simulacije znatno razlikuje od prethodnog, poslednji korišćeni korak integraljenja predstavlja dobar izbor. Ukoliko je razlika rezultata dve uzastopne simulacije velika, nastavlja se s polovljenjem koraka i ponavljanjem simulacije, sve dok se ne dobije dobro

slaganje rezultata uzastopnih simulacija. To je znak da je poslednji korišćeni korak integraljenja onaj pravi. S druge strane, ukoliko se rezultati dve poslednje uzastopne simulacije beznačajno razlikuju, usvaja se pretposlednji korak integraljenja. Ukoliko se već pri drugom ponavljanju simulacije dobije ova situacija, početni korak integraljenja je, verovatno, suviše mali. Ako se, u tom slučaju, želi uštedeti vreme rada, ponavljaju se simulacije, udvostručujući korak integraljenja svaki put, sve dok se ne dobije razlika rezultata dve uzastopne simulacije, koja je u granicama prihvatljivih grešaka. Tada se usvaja pretposlednji korak integraljenja.

Literatura:

- [1] Milojković, B., Grujić, Lj.: Automatsko upravljanje, Mašinski fakultet, Beograd, 1981.
- [2] Radojević, M., Radović, P., Dodić, N., Viličić, A.: Obuka u oblasti projektovanja, analize i merenja u servosistemima – sistemi upravljanja vatrom, simulacija i merenje – skripta, VTI VJ, Beograd, 1997.
- [3] Simonović, V.: Numeričke metode, Mašinski fakultet, Beograd, 1979.
- [4] Dodić, N.: Analiza domena asimptotske stabilnosti na Lurijeovom sektoru – diplomski rad, Mašinski fakultet, Beograd, 1984.