

Dinaminių sistemų valdymo kombinuotas modeliavimas

Robertas ALZBUTAS (KTU, LEI), Vytautas JANILIONIS (KTU)

el. paštas: robertas@isag.lei.lt, vyjan@mf.ktu.lt

1. Įvadas

Šiuo metu matematinis modeliavimas tapo neatskiriama sudėtingų dinaminių sistemų analizės dalimi. Dažnai jis yra vienintelis metodas, leidžiantis tirti jų valdymą. Yra sukurta nemažai modeliavimo sistemų, skirtų imitaciniams ir analiziniais modeliams kurti bei tirti. Kiekvienos modeliavimo sistemos bazę sudaro matematinė schema, kurios formalizavimo galimybės apibrėžia modeliuojamų sistemų klasę bei jų detalizacijos lygį.

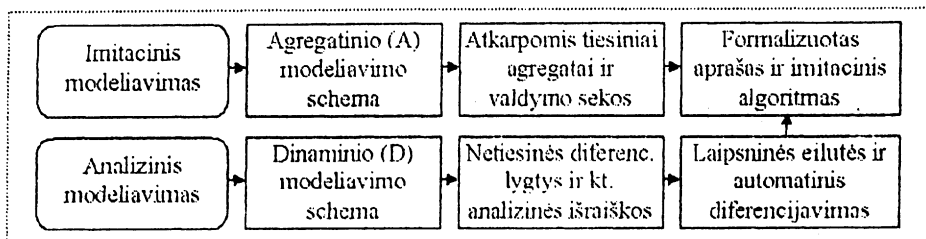
Dinaminių sistemų ir jų valdymo procesui modeliuoti naudojamos imitacinio modeliavimo arba analizinio modeliavimo sistemos. Kauno technologijos universitete jau daug metų tokių sistemų imitaciniam modeliavimui naudojama agregatinė schema (A schema). Yra sukurtos atitinkamos programinės modeliavimo sistemos PRANAS ir SIMAS [1,2], kurios automatizuoja agregatinių imitacinių modelių sudarymą ir tyrimą. Jų taikymo patirtis parodė, kad norint sukurti sudėtingesnių dinaminių sistemų valdymo proceso modelius ir adekvačiau atvaizduot jų funkcionavimą, nepakanka vienos matematinės schemos. Vienoms sistemos dalims modeliuoti geriau tinka imitacinio modeliavimo priemonės, o kitoms – analizinio. Pavyzdžiui, tolydinių sistemos komponentų formalizavimui geriau tinka matematinė dinaminio modeliavimo schema (*D* schema), kuri pagrįsta diferencialinių lygčių sudarymu ir jų sprendimu.

Įvertinus atskirų modeliavimo metodų privalumus bei trūkumus, autorių pasiūlyta kombinuoto modeliavimo metodika ir sukurta programinės priemonės leidžia viename modelyje tos pačios sistemos atskiras dalis formalizuoti ir modeliuoti naudojant skirtingas matematines *A* ir *D* schemas. Šiame darbe aprašoma sukurto kombinuoto modeliavimo priemonių realizacija ir taikymas.

2. Kombinuotas modeliavimas

Net ir tais atvejais, kai konkretūs modeliavimo metodai skiriasi viens nuo kito, galima išskirti bendrus realios sistemos *S* modelio *M* kūrimo principus ir juos panaudoti sudarant modelių kūrimo ir modeliavimo metodiką. Viena iš tokių metodikų pateiktą [3,4].

Sudarant sistemos *S* modelį *M* ir jį tiriant naudojamos imitacinio ir analizinio modeliavimo priemonės (1 pav.). Imitacinės modelio dalies formalizavimui naudojama matematinė *A* schema, kurios pagrindiniai elementai yra atkarpomis tiesiniai agregatai ir valdymo sekos [5]. Analizinė



1 pav. Formalizavimo priemonių tarpusavio ryšių schema.

modelio dalis formalizuojama naudojant diferencialinėmis lygtimis pagrįstą D schemą. Kadangi realių sistemų formalizavimui gali būti naudojamos ne tik diferencialinės lygtys, bet ir kitos analizinės išraiškos, šiame darbe siūloma išplėsta D schema. Ji realizuota taikant automatinio diferencijavimo [3,6] ir laipsninių eilučių teoriją. Programinis sistemos modelis sudaromas ir tiriamas naudojant integruotas modeliavimo priemones, kurias sudaro autorių sukurti agregatinio ir dinaminio modeliavimo paketai SIMAS ir ADPRO [3,4].

3. Dinaminio modeliavimo schemos realizacija

Tolydinės determinuotos sistemos dažniausiai aprašomos netiesinėmis diferencialinėmis lygtimis ar kitomis analizinėmis išraiškomis. Siekiant atlikti sistemos atskirų dalių funkcinę ir struktūrinę sintezę reikalinga bendra lygčių sprendinių ir analizinių išraiškų užrašymo forma. Nagrinėjamu atveju siūloma naudoti laipsnines eilutes.

Įvairių netiesinių n -tos eilės diferencialinių lygčių sprendinys duoto taško aplinkoje gali būti išreiškiamas Teiloro eilute. Šiuo atveju pagrindinė problema – aukštesnių eilių išvestinių $y^{(k)}(x)$, kai $k > n$, apskaičiavimas. Šiai problemai spręsti siūloma nagrinėti Koši forma išreikštas netiesines n -tos eilės diferencialines lygtis arba pirmos eilės m diferencialinių lygčių sistemą bei taikyti automatinio diferencijavimo metodą [3,6].

Simbolinis diferencijavimas naudoja taisykles operavimui su simboliais, o automatinis diferencijavimas naudoja taisykles skirtas operavimui su skaitinėmis reikšmėmis. Skaičiavimai dažniausiai vykdomi vektorių sistemoje. Pavyzdžiui nagrinėjant funkciją $h(x) = \sin(u(x))$, jei taške x_0 funkcijos $u(x)$ ir jos išvestinių reikšmių vektorius $(u(x_0), u'(x_0), u''(x_0), \dots, u^{(n)}(x_0))$ žinomas, o ieškomos vektoriaus $(h(x_0), h'(x_0), h''(x_0), \dots, h^{(n)}(x_0))$ elementų reikšmės, tuomet visai nesudėtinga gauti formulę $h''(x_0)$ skaičiavimui išreikštą per $u(x_0)$, $u'(x_0)$ ir $u''(x_0)$. Tačiau bendras algoritmas apskaičiuoti funkcijos $y(u(x))$ išvestinei $y^{(n)}(x)$, kai n iš anksto apibrėžtas, reikalauja rekurentinių formulių bei iteracinio skaičiavimo. Lyginant su simboliu diferencijavimo metodu, kur operuojama su didelėmis simbolių sekomis, naudojant automatinio diferencijavimo metodą atliekama mažiau operacijų, todėl naudojama žymiai mažiau atminties ir skaičiavimo laikas yra daug trumpesnis.

Ieškant vektorių $(y, y'_x, y''_x, \dots, y^{(k)}_x)$, pagrindiniai skaičiavimai atliekami vektorinėje formoje, todėl metodo realizacijai buvo pasirinkta matematinė programavimo kalba APL2 [6,7]. Naudojant APL2 kalbą sukurta diferencialinių lygčių tyrimo programa ir kalba DIFLAN. Sufor-

Lentelė 1
Funkcijų išvestinių vektoriai

Funkcija	Skaičių vektorius
$h(x)$	$H = (h(x_0), h'(x_0), h''(x_0), \dots, h^{(n)}(x_0))$
$u(x)$	$U = (u(x_0), u'(x_0), u''(x_0), \dots, u^{(n)}(x_0))$
$v(x)$	$V = (v(x_0), v'(x_0), v''(x_0), \dots, v^{(n)}(x_0))$
$e(x)$	$X = (x_0, 1, 0, 0, \dots, 0)$
$c(x)$	$C_s = (s, 1, 0, 0, \dots, 0)$

muluosime Koši uždavinį šioje kalboje. Rasti m -tos eilės diferencialinės lygties

$$(DY\ m) = F(X, Y, (DY\ 1), (DY\ 2), \dots, (DY\ m-1))$$

sprendinį $Y(X)$, kai duotos $Y, (DY\ 1), (DY\ 2), \dots, (DY\ m-1)$ reikšmės taške X_0 , čia $(DY\ k)$ k -os eilės Y išvestinė X atžvilgiu, m – diferencialinės lygties eilė. Funkcijos F užrašymui naudojami kintamieji: $X, Y, (DY\ 1), \dots, (DY\ m-1)$, specialūs operatoriai bei kalbos taisyklės. Analogiškai formuluojamas ir uždavinys pirmos eilės diferencialinių lygčių sistemai

$$\begin{aligned} dY[1;]/dX &= F1(X, Y[1;], \dots, Y[m;]) \\ dY[m;]/dX &= Fm(X, Y[1;], \dots, Y[m;]), \end{aligned}$$

čia $Y[k;]$ yra k -oji sistemos funkcija $Y_k(X)$, o m – pirmos eilės diferencialinių lygčių kiekis sistemoje. Funkcijai Fm užrašyti naudojami tie patys operatoriai ir taisyklės, kaip ir aprašant diferencialinę lygtį, tik šiuo atveju naudojami kintamieji: $X, Y[1;], \dots, Y[m;]$.

Aprašant diferencijavimo operatorius ir jų taikymo principus naudosime 1 lentelėje pateiktus žymėjimus.

Toliau tekste praleidus x_0 rašysime trumpiau, pavyzdžiui $U = (u, u', u'', \dots, u^{(n)})$.

Tarkim kad, žinant funkcijas $u(x)$ ir $v(x)$ atitinkančius vektorius U ir V , reikia rasti vektorių H , kuris atitinka sudėtinę funkciją $h(x)$ išreiškiamą per funkcijas $u(x)$ ir $v(x)$. Vektoriaus H radimui reikalinga vektorinė operacija, kurios argumentai yra U ir V vektoriai.

Jei $h(x) = u(x) \cdot v(x)$, tada reikia sandaugos operatoriaus TIMES tokio, kad $H = U \text{ TIMES } V$. Aukštesnės eilės išvestinės yra formuojamos taip pat, kaip binominėje formulėje

$$h^{(k)} = (0!k)u^{(k)}v + (1!k)u^{(k-1)}v' + (2!k)u^{(k-2)}v'' + \dots + (k!k)uv^{(k)}. \quad (3.1)$$

Čia naudojamas APL kalbos pažymėjimas $(n!m) = C_m^n$. Realizuojant Leibnico taisyklę ir kitus operatorius, apibrėžiama vektorinė operacija BDOT (angl. „binomial dot product“).

Kai žinomi vektoriai $P = (p_0, p_1, \dots, p_k)$ ir $Q = (q_0, q_1, \dots, q_k)$, tai

$$P \text{ BDOT } Q = (0!k)p_kq_0 + (1!k)p_{k-1}q_1 + \dots + (k!k)p_0q_k. \quad (3.2)$$

Naudojant šią vektorinę operaciją vienu veiksmu galima rasti ieškomo vektoriaus H kiekvieną k -ąjį elementą $h^{(k)}$. Pavyzdžiui, jei $h(x) = u(x) \cdot v(x)$, tuomet

$$h^{(k)} = (u, u', \dots, u^{(k)}) \text{BDOT}(v, v', \dots, v^{(k)}). \quad (3.3)$$

Tarkime, kad $\forall k \geq 0$ žinomas binominių koeficientų vektorius $B_k = ((0!k)(1!k) \dots (k!k))$, tuomet operatoriaus BDOT realizacija APL2 kalboje galime užrašyti

```

VSCALAR ← PBDOT Q; M; BINOMCOEFS
[1] K ← p, P
[2] BINOMCOEFS ← K ↑ ((K - 1) × K ÷ 2) ↓ BCVECTOR
[2] SCALAR ← +/BINOMCOEFS × (ϕP) × Q
[3] ∇.

```

čia $BCVECTOR = (B_0 \ B_1 \ B_2 \dots B_K) = ((1) \ (1 \ 1) \ (1 \ 2 \ 1) \ (1 \ 3 \ 3 \ 1) \ (1 \ 4 \ 6 \ 4 \ 1) \dots ((0!K) \ (1!K) \dots (K!K)))$.

Turint BDOT operatorių formalizuotos rekurentinės formulės skirtos įvairių funkcijų k -tos eilės diferencijavimui. Atliekant vektorinius pertvarkymus bei naudojantis jau turimais operatoriais sudaryti DIFLAN kalbos operatoriai: DIV, ARCSIN, TAN, ARCTAN, COT, SEC, CSC, SQRT ir t.t. Pavyzdžiui, jei $h(x) = \csc(u(x))$, tuomet $H = \text{CSC } U = C1 \text{DIV}(\text{SIN } U)$ arba, jei $h(x) = u(x)^r$, tuomet galime užrašyti $H = U \text{POWER } r = \text{EXP}(r \times (\text{LN } U))$. Kadangi diferencijuojamos funkcijos aukštos eilės išvestinių apskaičiavimui naudojamas tikslus metodas, tai šiuo atveju galimos tik kompiuterio skaičių vaizdavimo (apvalinimo) paklaidos, kurios gali būti mažinamos tobulinant vektoriaus B_k gavimo būdus.

Aprašytas metodas gali būti taikomas sprendžiant netiesines bet kokios eilės diferencialines lygtis. Esant poreikiui jį galima taikyti ir diferencialinės lygties bendro sprendinio paieškai. Jei visos nagrinėjamos funkcijos išvestinės nuo tam tikros eilės yra lygios nuliui, galima rasti tikslią diferencialinių lygčių sprendinio analizinę išraišką. Automatinio diferencijavimo metodas gali būti panaudotas ne tik diferencialinių lygčių sprendimui, bet ir įvairių netiesinių analizinių išraiškų aproksimavimui, ribų ir integralų apskaičiavimui. Šiuo metodu įmanoma apskaičiuoti ir tokius integralus, kurie baigtinėje formoje neišreiškiami elementariomis funkcijomis.

Visais automatinio diferencijavimo taikymo atvejais rezultatą galima išreikšti vienuodu pavidalu – Teiloro eilutės koeficientais arba nagrinėjamos funkcijos išvestinių vektoriumi $(y, y'_x, y''_x, \dots, y^{(k)}_x)$. Toks rezultatų formos vienodumas labai naudingas sudarant sudėtingų sistemų funkcionavimo ir jų valdymo modelius.

Pastebėsime, kad nuoseklaus diferencijavimo naudojimas bendru atveju nesuteikia galimybės tyrinėti gautos eilutės konvergavimą, nes daugeliu atveju neįmanoma gauti tiriamos eilutės bendro nario analizinės išraiškos. Šis metodas labiausiai tinka tuomet, kai iš anksto yra žinoma, kad egzistuoja laipsninė eilutė išreiškiamas sprendinys.

Nagrinėti metodai realizuoti panaudojus APL2 kalbą. Sudaryta universali dialogu ar duomenų failais valdoma dinaminio modeliavimo sistema ADPRO, kurioje iškirtas diferencialinių

lygčių sprendimo posistemis DERECUR bei pirmos eilės diferencialinių lygčių sistemos sprendimo posistemis DESYS. Sukurtos dinaminio modeliavimo priemonės integruotos į agregatinio modeliavimo sistemą SIMAS [1,2]. Sistemos ADPRO struktūra ir jos panaudojimo galimybės pateiktos publikacijose [3,4].

4. Valdymo sistemų modeliavimas

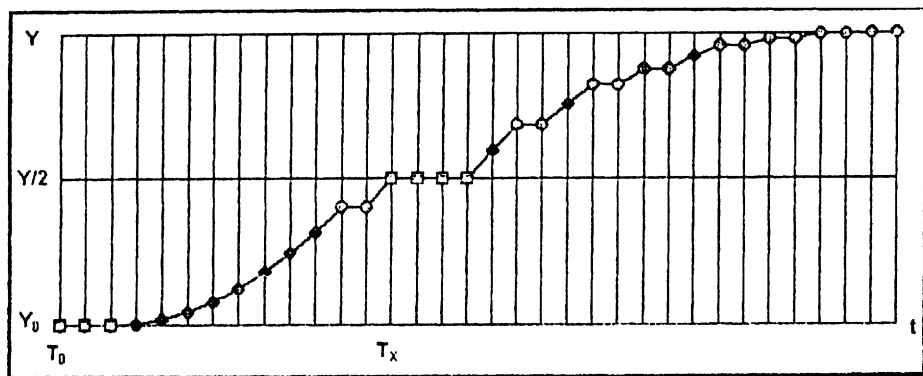
Modeliuojamos valdymo sistemos (VS) išskaidomos į posistemius. Modeliavimui naudojamos struktūros, kuriose suderintas globalus (centralizuotas) sistemos valdymas ir lokalus atskirų objektų (ar jų grupių) valdymas. Globalaus valdymo lygio VS imitacinių modelių realizavimas yra pakankamai išnagrinėtas ir ypatingų problemų nesudaro. Šio valdymo lygio sistemas galima modeliuoti panaudojant vien matematinę A schemą ir jos pagrindu sudarytas imitacinio modeliavimo priemones. Tačiau nagrinėjant netiesinių dinaminių sistemų funkcionavimą aktuali ir lokalaus valdymo adekvataus modeliavimo galimybė. Modeliuojant netiesinių sistemų valdymą tikslinga išskirti sistemoje vykstančius netiesinius procesus, o jų funkcionavimą bei valdymą modeliuoti atskirai. Siekiant unifikuoti netiesinių procesų ir jų valdymo modeliavimo įvairovę darbe išskirti ir spręsti du pagrindiniai uždaviniai: 1) proceso ir jo valdymo dekompozicijos; 2) proceso ir jo valdymo imitacijos. Netiesinių procesų dekompozicijai atlikti, nustatyti dekompozicijos principai ir apibrėžti skirtingi dekompozicijos lygiai. Pagrindinius dekompozicijos principus atitinka du dekompozicijos lygiai, kurie priklauso nuo:

- 1) proceso dalių išskyrimo remiantis jų atskaitos sistemų nepriklausomumu;
- 2) pačių proceso dalių analizinio formalizavimo ir imitavimo dekompozicijos. Pirmame dekompozicijos lygyje gautų valdymo elementų tarpusavio sąveika gali būti realizuojama panaudojant klasikinės valdymo struktūras: S_D – decentralizuotą, S_C – centralizuotą, S_P – paskirstytą, S_H – hierarchinę bei agregatinio modeliavimo principus. Kiekvieno proceso P_i , kuris antrame dekompozicijos lygyje gali būti suskaidytas į atskiras dalis, ir jį valdančio lokalus valdymo objekto C_i , $i \in \{1, 2, 3, \dots\}$ funkcionavimo bei valdymo imitavimas realizuojamas panaudojus specialiai sudaryta lokalaus valdymo sistemą [3,4]. Ji susideda iš valdymo, stebėjimo ir proceso funkcionavimo agregatų. Sudaryta lokalaus valdymo sistema užtikrina proceso P_i atskirų dalių $P_i^1, P_i^2, \dots, P_i^m$ suderinto funkcionavimo imitavimą. Naudojant sudarytą lokalaus valdymo sistemą galimi šie imitavimo atvejai:

- 1) tolydus proceso būsenų kitimas pereinant nuo vienos prie kitos dalies imitavimo;
- 2) pažingsninis pačio proceso ar jo atskirų dalių funkcionavimo dėsnio keitimas.

Iliustruojant aprašytas kombinuoto modeliavimo galimybes pateiksime paprastą valdomo proceso P_1 atvejį (2 pav.).

Procesas P_1 susideda iš dviejų analizinių dalių. Pirmoji dalis P_1^1 (2 pav. nuo būsenos Y_0 iki būsenos $Y/2$) DIFLAN kalba aprašoma dėsniu $(DY \ 1) = Z \times X$. Kita proceso dalis P_1^2 (nuo būsenos $y(T_X)$ iki būsenos Y) aprašoma dėsniu $(DY \ 1) = Z \times (T_X - X)$. Čia DIFLAN kalbos kintamasis X atitinka 2 pav. pažymėtą laiko kintamąjį t , o $Z = Z(Y)$ atitinka proceso intensyvumo parametą, priklausančią nuo ribinės proceso būsenos Y . Pradėjus nuo laiko momentų T_0 ir T_X , valdymo agregate nustatant ar keičiant proceso dėsni, proceso būsena nekinta atsitiktinai



2 pav. Proceso būsenos įvertinant valdymo ir stebėjimo laiko sąnaudas.

laiko intervalą. Nuo pradinio laiko momento T_0 procesas P_1 stebimas apibrėžtu dažniu. Šio proceso stebėjimui su tikimybe $y(t)/Y$ sugaištamas laiko intervalas nelygus 0. Tokiu būdu funkcionuojantis procesas realizuotas naudojant sudarytus valdymo ir stebėjimo agregatus bei agregatą, imituojančią analiziškai aprašomą proceso dalį P_1 . Aišku, daugkartinis kiekvieno lokalaus valdymo posistemo su imitacinio eksperimento metu kintančiais parametrais funkcionavimas taip pat priklauso ir nuo modeliujamos sistemos išorinės aplinkos ir globalaus valdymo algoritmo.

5. Išvados

1. Pasiūlyta kombinuoto modeliavimo metodika leidžia modeliuoti tokias sistemas, kurių atskirų dalių funkcionavimas formalizuojamas panaudojant matematinės A ir D schemas.
2. Pritaikius automatinio diferencijavimo metodą, sukurtas dinaminio modeliavimo paketas ADPRO, kuris gali spręsti netiesines, Koši forma išreikštas, aukštos eilės diferencialines lygtis, pirmos eilės diferencialinių lygčių sistemas bei aproksimuoti netiesines funkcijas. Bendroju atveju sprendiniai nagrinėjamo taško aplinkoje išreiškiami Teiloro eilute, kuri gali būti panaudota modeliujamos sistemos posistemių funkciniai ir struktūriniai sintezai.
3. Pasiūlytas valdymo sistemų modelių sudarymo metodas, kuris pagrįstas sistemos valdymo dekompozicija į globalaus ir lokalaus valdymo posistemius bei sukurtu lokalaus netiesinio proceso ir jo valdymo imitavimo modeliu. Šis metodas leidžia modeliuoti ir tirti tokias sistemas, kurių valdymą galima suskaidyti į keletą hierarchinių lygių, o žemiausiame valdymo lygyje esantys netiesiniai tolydūs procesai yra aprašomi keliomis analizinėmis išraiškomis, kurių determinuoti ar stochastiniai parametrai gali būti keičiami diskrečiais laiko momentais.
4. Panaudojus pasiūlytą metodiką ir sukurtas kombinuoto modeliavimo priemonės galima paprasčiau ir adekvačiau modeliuoti valdymo sistemas. Sukurtos priemonės leidžia modeliuoti ir tirti sistemų, kurios aprašomos netiesiniais tolydiniais ir diskretiniais procesais su pastoviais ir kintamais, determinuotais ir stochastiniais parametrais, funkcionavimą ir valdymą. Jos gali būti panaudotos sprendžiant taikomuosius ir teorinius modeliavimo uždavinius, o taip pat mokymo procese.

Literatūra

- [1] Г. Й. Праневичюс, В. В. Яниленис, Построение проблемно-ориентированных систем имитационного моделирования на базе системы СИМАС, *Моделирование систем информатики*, Новосибирск, 90–92 (1988).
- [2] Г. Й. Праневичюс, В. В. Яниленис, Системы имитационного моделирования агрегативных систем (СИМАС), *Перспективы развития вычислительных систем* (Применение идей адаптации и эволюции), РПИ, Рига, 147–149 (1985).
- [3] R. Alzbutas, V. Janilionis, Dinaminių sistemų modeliavimas, *Matematika ir matematinis modeliavimas, Technologija*, Kaunas, 2-a knyga, 82–88 (1998).
- [4] R. Alzbutas, V. Janilionis, The extension of aggregate simulation system SIMAS with means of dynamical simulation, *Matematika ir matematinis modeliavimas, Technologija*, Kaunas, 71–77 (1999).
- [5] H. Pranevichus, Formal specification and snalysis of distributed systems, in: *Lecture Notess in Applications of AI to Production Engineering*, Nordic Baltic Summer SchoolT97, 269–322 (1997).
- [6] R.D. Neidinger, Automatic differentiation and APL, *College Mathematics Journal*, 20(3) (May), 238–251 (1989).
- [7] J.A. Brown, *Pakins Polivka APL2 at a Glance*, Prentice-Hall (1988).

The combined modelling of dynamic systems control

R. Alzbutas, V. Janilionis

The advantages and disatvantages of modelling methods were evaluated, afterwards the combined modelling was proposed and the corresponding software was developed. This software allows formalisation and modelling of different parts of the same system using different aggregate and dynamic mathematical modelling schemes (*A*-scheme and *D*-scheme). The usage and application of the developed combined modelling means are presented.