

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
VERSLO INFORMATIKOS KATEDRA

Laurynas Simaitis

**Hibridinių sistemų modeliavimas naudojant
agregatinį metodą**

Magistro darbas

Darbo vadovas

prof. H. Pranevičius

Kaunas, 2007

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
VERSLO INFORMATIKOS KATEDRA

Laurynas Simaitis

**Hibridinių sistemų modeliavimas naudojant
agregatinį metodą**

Magistro darbas

Recenzentas

prof. habil. dr. V. Štuikys
2007-05-24

Darbo vadovas

prof. H. Pranevičius
2007-05-

Atliko

IFM-1/1 gr. stud.
Laurynas Simaitis

2007-05-24

Kaunas, 2007

Santrauka

Šiame darbe pristatomas PLA formalizmo metodas, skirtas aprašyti ir imituoti hibridines sistemas, kurių tolydiniai komponentai aprašomi pirmos eilės diferencialinėmis lygtimis. Pasinaudojus PLA formalizmu suformuluosime Kvantuotų būsenų sistemos modelį (QSS). QSS modelis yra integravimu pagrįstas metodas, skirtas spręsti pirmos eilės diferencialines lygtis. Šis modelis buvo suformuluotas DEVS formalizmo aplinkoje, kuris yra giminingas PLA formalizmui. Hibridinės sistemos pavyzdžiu pasirinktas farmakokinetinis modelis. Farmakokinetika yra farmakologijos mokslo šaka, tyrinėjant vartojamų vaistų pasisavinimą ir absorbciją žmogaus kūne. Farmakokinetiniame modelyje vaistų pasiskirstymas aprašomas pirmos eilės diferencialines lygtimis. Tam, kad būtų galima aprašyti šį modelį QSS pagalba, buvo reikalinga patobulinti QSS modelį. Rezultate gautas pakeistas QSS modelis. Naudojant pakeistą QSS modelį buvo sudarytas farmakokinetinis modelis ir atliktas vaistų koncentracijos kitimo imitavimas paciento kūne, kai pacientas reguliuoja vaistų suleidimą.

Summary

This paper presents the method of using Piece Linear Aggregates formalism (PLA) for specification and simulation of hybrid systems, when continuous components are described in ordinary differential equations (ODEs). PLA is used to create Quantized State System (QSS) model. QSS is integration based method, created for ODE solving. Since QSS method first was introduced for use in Discrete Event Systems (DEVS) formalism, we adapted it for PLA formalism, which is fundamentally consanguineous to DEVS formalism. Pharmacokinetics model is used as example of hybrid system. Pharmacokinetics is a branch of pharmacology which describes administered drug absorption and distribution in human body. In pharmacokinetics drug distribution is described using ODE. In order to use drug injection in pharmacokinetics model, we have to modify standard QSS model and add some specific capabilities. In the result: modified QSS method was obtained and used for pharmacokinetic model specification. Simulation of morphine concentration in plasma was performed using real pharmacokinetics data to test Patient-Controlled Analgesia method effectiveness.

Turinys

1	Įvadas	6
2	Diskrečiųjų būsenų sistemos	9
2.1	Įvadas į PLA.....	9
2.2	Įvadas į DEVS.....	10
3	Hibridinės sistemos modelis	12
3.1	Kvantuotų būsenų sistemos modelis	12
3.1.1	<i>Matematinis modelis</i>	12
3.1.2	<i>QSS modelio savybės</i>	14
3.1.3	<i>PLA modelis</i>	14
3.2	Funkcijos diferencialo apskaičiavimas.....	16
3.3	Valdymo signalų generavimas	19
3.4	Bendras hibridinės sistemos modelis	20
4	Kvantuotų būsenų sistemos panaudojimas hibridinėms sistemoms modeliuoti	23
4.1	Elektroninės schemos aprašymo pavyzdys	23
4.2	Hibridinės sistemos agregatai.....	23
4.3	Modeliavimo rezultatai	24
5	Kvantuotos būsenų sistemos modelio modifikacija	27
5.1	Reikalingi pakeitimai	27
5.2	Modifikuotas QSS modelis, leidžiantis keisti funkcijos reikšmę, aprašytas naudojant PLA formalizmą	27
5.3	Kitos galimos QSS modelio modifikacijos	29
6	Pakeisto Kvantuotų būsenų sistemos modelio panaudojimas	30
6.1	PCA modelis	30
6.2	Farmakokinetinis modelis	31
6.3	Paciento modelis	32
6.4	Vaistų pompos modelis	33
6.5	Farmakokinetinis modelis	34
6.6	Modelio imitavimo rezultatai	35
7	Išvados	37
8	Naudota literatūra	38
9	Priedai	39
9.1	Standartinis QSS modelis, aprašytas naudojant DEVS formalizmą.....	39
9.2	Elektroninės schemos modelis, sudarytas naudojant imitavimo įrankį PowerDEVS.....	41
9.3	Farmakokinetinis modelis su PCA, sudarytas naudojant imitavimo įrankį PowerDEVS	43

1 Įvadas

Visiems tenka susidurti su įvairaus tipo hibridinėmis sistemomis. Jos slypi gamyboje, daugumoje šiuolaikinių buities prietaisų, net žaisluose. Kuriant ir projektuojant šias sistemas, reikia laikytis vis didesnių laiko sąnaudos ir kokybės reikalavimų. Šių sistemų modelių sudarymas ir imitavimas jau seniai tapo neatskiriama sistemų kūrimo proceso dalis. Tai net tik padeda sumažinti klaidų skaičių, bei gali žymiai sumažinti darbo kaštus. Modelio sudarymas ankstyvoje kūrimo fazėje gali padėti atrasti tokias nepageidautinas sistemos ypatybes, kurios būtų pastebimos tik pabaigus sistemos kūrimą. Jei sistema turi klaidų, tai gali sukurti kritines situacijas ir ne tik atnešti didelius nuostolius, bet ir kainuoti žmonių gyvybes.

Hibridinė sistema yra sistema, kurią sudaro tolydieji ir diskretieji komponentai. Pastaruoju metu vis dažniau tenka susidurti su sistemomis, kurios susideda iš elektromechaninių ir skaitmeninių komponentų. Tokios sistemos yra dažniausiai pasitaikantys hibridinių sistemų pavyzdžiai. Galimi ir kitokie pavyzdžiai – biologinės, mechaninės, cheminių procesų hibridinės sistemos. Tačiau visose sistemose diskrečioji dalis yra tam tikro tipo diskretus įrenginys, skirtas valdyti tolydžiąją sistemos komponentę. Taigi, hibridinės sistemos yra glaudžiai susijusios su sistemų valdymo uždaviniu. Hibridinės sistemos, tiek iš dalies automatinės, tiek pilnai savarankiškos, yra fizinės sistemos, veikiančios realiame laike. Realus laiko sistemų veikimo imitavimas yra populiarus uždavinys, kuriam spręsti yra sukurta daug įvairių metodų. Priklausomai nuo sistemos pobūdžio, gali skirtis imitavimo uždavinys.

Norint ištirti sistemos savybes, reikia formaliai apibrėžti sistemos modelį. Galimi įvairūs sistemos modelių abstrakcijos lygmenys, kurie priklauso nuo sistemos sudėtingumo ir klausimų, kuriuos norima išspręsti naudojant imitavimą. Priklausomai nuo modelio sudėtingumo, galimi įvairūs modelių užrašymo būdai. Esant aukštam abstrakcijos lygmeniui, gali būti naudojama SPIN [2] modelių verifikavimo sistema. Tačiau nelabai tinka tirti sistemoms, kurios veikia realiame laike, nes tada modelį reikia supaprastinti. Tiriant realaus laiko sistemas labiau tiktų panaudoti UPPALL [3] įrankį. Tačiau ir šis sistemų modeliavimo metodas yra skirtas tik abstraktiems modeliams tirti.

Norint sudaryti hibridinės sistemos modelį, kuris smulkmeniškai atvaizduotų tolydžiosios komponentės būsenų trajektorijas. Reikia tokio metodo, kuris gebėtų spręsti diferencialines lygtis ir būtų lengvai suderinamas su automatais. Įprastiniai sprendimo metodai yra netinkami, nes jie paremti diskrečiu laiku. Ši metodų savybė yra labai nepatogi, kai yra būtina tiksliai nustatyti, kada sistema pasiekia konkrečias diskrečiąsias būsenas. Laiko

tinklėlio smulkinimas reikalauja didelių skaičiavimo pajėgumų, tačiau negarantuoja, kad bus pastebėti visi įvykiai. Vieni iš labiausiai tinkamų metodų hibridinėms sistemoms modeliuoti būtų diskrečių būsenų sistemos. Populiariausi formalizmai, skirti diskrečių būsenų sistemoms aprašyti yra Agregatinis metodas (PLA [1]) ir DEVS [4]. Šios formaliosios kalbos yra iš principo giminingos, turinčios tik keletą mažareikšmių skirtumų.

Naudojant diskrečių būsenų sistemas, būtų galima sudaryti vientisą modelį, kuris gerai atitiktų tiek tolydžiąją komponentę, kurios veikimas aprašomas diferencialinėmis lygtimis, tiek diskrečiąją. Reikalinga metodika, kuri spręstų diferencialines lygtis diskrečių būsenų sistemų aplinkoje. Kvantuotų būsenų sistema (QSS) [5] yra DEVS modelis, skirtas spręsti pirmos eilės diferencialines lygtis (ODEs). Yra sukurta keletas šio metodo modifikacijų – QSS2 [5] ir QSS3 [6], kurios pasižymi geresnėmis veikimo charakteristikomis, tačiau yra paremtos ta pačia idėja.

Nors buvo parodyta, kaip sudaryti tam tikrų hibridinių sistemų modelius naudojant kvantuotų būsenų sistemą [4], tačiau norint ją panaudoti įvairesnėms, reikia šiam modeliui pridėti daugiau funkcinių savybių. Šiame darbe bus paimtas QSS modelis ir aprašytas kaip atkarpomis tiesinis jungtinis agregatas. Toliau bus tiriamos įvairaus tipo hibridinės sistemos, naudojant Kvantuotų būsenų sistemą ir jos modifikacijas. Elektros grandinė su jungikliais būtų paprasčiausias hibridinės sistemos pavyzdys. Norint aprašyti tokią sistemą, reikia gebėti imituoti sistemą, kurios diferencialas gali kisti šuoliškai.

Pagrindinis hibridinės sistemos pavyzdys – farmakokinetinio modelio sudarymas. Farmakokinetinis modelis aprašo vaistų cirkuliaciją žmogaus kūne, kurių koncentracijos kitimas aprašomas pirmos eilės diferencialinėmis lygtimis. Diskrečioji sistemos komponentė yra vaistų suleidimas, kuris šuoliškai keičia vaistų koncentraciją kraujyje. Šiam modeliui aprašyti bus reikalinga pakeisti standartinį kvantuotų būsenų modelį, kad jis leistų šuoliškai keisti ne tik funkcijos diferencialą, bet ir pačią funkcijos reikšmę. Norint sudaryti farmakokinetinį modelį, kai pacientas reguliuoja sau suleidžiamus vaistus (PCA¹), reikalingas ne tik vaistų koncentracijos kitimo modelis, bet ir paciento terapeutinio lygio² kitimo modelis bet paciento elgesio modelis, kuris priklausomai nuo terapeutinio lygio ir esamos vaistų

¹ *Patient controlled analgesia* – vaistų dozavimo būdas, kai pacientas pats nusprendžia, kiek jam reikia vaistų, ir naudodamasis specialia vaistų suleidimo įranga, vykdo vaistų suleidimą.

² Terapeutinis lygis – vaistų koncentracija, kuri pacientui užtikrina geriausią poveikį. Terapeutinis lygis gali kisti per dienos laikotarpį, sumažėdamas ar padidėdamas net keletą kartų.

koncentracijos, imituotų, kaip dažnai pacientas pageidauja vaistų. Vaistų suleidimui imituoti bus sukurtas atskirtas vaistų pompos modelis, paskirstantis vaistų dozę į keletą injekcijų per nustatymą laiko tarpą.

Šio darbo tikslas yra panaudoti PLA formalizmą hibridinių sistemų modelių sudarymui ir imitavimui atlikti. Parodysime, kad Agregatinis metodas turi tas pačias galimybes kaip ir DEVS formalizmas ir yra tinkamas hibridinėms sistemos modeliuoti. Imitavimams atlikti bus naudojamas DEVS aplinkai sukurtas įrankis PowerDEVS [4], todėl norint jį panaudoti, reikia parodyti, kaip atkarpomis tiesinius agregatus galima vienareikšmiškai perversi į DEVS modelį ir jį aprašyti. Imitavimai bus panaudoti įvertinti hibridinių sistemų savybes. Apjungus PLA formalizmą, QSS modelį ir PowerDEVS modeliavimo aplinką parodysime, kaip modeliuoti įvairaus tipo hibridines sistemas.

Pirmoje darbo dalyje bus pateikta naudojamų PLA ir DEVS formalizmų apžvalga. Apžvelgsime modelių struktūrą ir pagrindinius veikimo principus, panašumus ir skirtumus. Toliau apžvelgsime Kvantuotų būsenų sistemą (QSS), jos formalų aprašą ir savybes. Pagal pateiktą matematinį modelį aprašyme QSS sistemą naudojant PLA formalizmą. Pateiksime hibridinės sistemos pavyzdį, kaip ją aprašyti ir simuliuoti naudojant standartinį QSS modelį.

Tolesnėje darbo dalyje įvesime keletą pakeitimų į QSS, suteikdami modeliui daugiau pritaikymo galimybių aprašant hibridines sistemas. Naudojant pakeistą QSS aprašysime farmakokinetinį modelį su paciento atliekamu nuskausminimu (PCA [9]). Naudojant surinktus statistinius duomenis apie pacientų farmakokinetinius koeficientus, bus sudarytas paciento vaistų poreikio modelis, kuris valdys vaistų dozavimą. Atliekant imitavimą bus bandoma nustatyti kaip kinta vaistų koncentracija kraujo plazmoje ir kaip imitavimo rezultatai atitinka statistinius pacientų duomenis.

2 Diskrečiųjų būsenų sistemos

2.1 Įvadas į PLA

Agregatinio metodo formalizmas yra kalba, skirta aprašyti atkarpomis tiesinius agregatus (Piece Linear Aggregates). Agregatas yra aprašomas nurodant:

$$A = (Z, Y, H, G, E)$$

čia:

- Z - būsenų aibė,
- Y – įėjimo signalų aibė,
- H – perėjimo signalų atvaizdavimas,
- G – išėjimo operatorius,
- E – įvykių aibė. $E = \{E', E''\}$, čia E' – išorinių įvykių aibė, E'' – vidinių įvykių aibė.

Agregatas veikia laiko momentų $t \in T$ aibėje. Atkarpomis tiesinio agregato būsenos struktūra yra tokia:

$$z(t) = (v(t), z_v(t)) \in Z$$

čia: $v(t)$ – diskrečioji būsenos dedamoji,

$z_v(t)$ – tolydi būsenos dedamoji.

Bendruoju atveju

$$v(t) = \{v_1(t), v_2(t), \dots, v_m(t)\},$$

$$z_v(t) = \{z_{v1}(t), z_{v2}(t), \dots, z_{vk}(t)\},$$

čia: $v_i(t)$ – i -toji diskrečiosios dedamosios komponentė,

$z_{vi}(t)$ – i -toji tolydžiosios dedamosios komponentė.

Agregato būseną keičiasi atėjus išoriniam signalui – įvykus išoriniam įvykiui – $e'_i \in E' \subseteq E$, arba kai viena iš tolydžiosios dedamosios koordinačių įgyja tam tikrą reikšmę – įvykus vidiniam įvykiui – $e''_i \in E'' \subseteq E$. Išėjimo signalas gali būti generuojamas pakitus agregato būsenai (įvykus tiek vidiniam, tiek ir išoriniam įvykiui).

2.2 Įvadas į DEVS

Diskrečiųjų įvykių sistemų specifikuojimo formalizmas (*Discreet event system*) yra kalba, skirta aprašyti sistemas, kurių įėjimo/išėjimo signalus galima aprašyti įvykių seka, laikantis sąlygos, kad baigtinio ilgio laiko tarpe pasikeičia baigtinis būsenų skaičius. Sistemos būsenų trajektorija priklauso nuo jos pradinės būsenos parametrų ir įėjimo signalų sekos. Formaliai DEVS modelis aprašomas tokia struktūra:

$$M = (X, Y, S, \delta_{\text{int}}, \delta_{\text{ext}}, \lambda, ta)$$

čia:

- X įėjimo įvykių (parametrų) aibė (visų įvykių aibė, kuri gali būti kaip įėjimo parametras),
- Y išėjimo įvykių aibė,
- S būsenų aibė,
- $\delta_{\text{int}}, \delta_{\text{ext}}, \lambda$ ir ta - funkcijos, kurios aprašo sistemos dinamiką.

Kiekviena galima būsena s ($s \in S$) susieta su laiku apskaičiuotu pagal laiko funkciją $ta(s)$ ($ta(s) : S \rightarrow R_0^+$). ta (*time advance*) tai neneigiamas realus skaičius nusakantis kaip ilgai sistema pasiliks duotoje būsenoje, kai nėra naujo įėjimo parametro. Taigi jei būsena įgyja reikšmę s_1 laiko momentu t_1 , tai po $ta(s_1)$ (t.y. po $ta(s_1) + t_1$ laiko) laiko sistema įvykdys vidinį perėjimą (*internal transition*) į būseną s_2 . Naujoji būsena apskaičiuojama $s_2 = \delta_{\text{int}}(s_1)$. Funkcija δ_{int} ($\delta_{\text{int}} : S \rightarrow S$) vadinama vidinio perėjimo funkcija (*Internal Transition Function*).

Kai būsena pakinta iš s_1 į s_2 išėjimo signalas sudaromas su reikšme $y_1 = \lambda(s_1)$. Funkcija λ ($\lambda : S \rightarrow Y$) vadinama išėjimo funkcija (*Output Function*). Funkcijos ta, δ_{int} ir λ apibūdina autonominių DEVS modelio elgesį.

Kai paduodamas įėjimo parametras, būsena pakeičiama akimirksniu. Nauja būsenos reikšmė priklauso ne tik nuo naujo įėjimo parametro, bet ir nuo anksčiau buvusios būsenos reikšmės, bei praėjusio laiko nuo paskutinio būsenos perėjimo. Jei sistema pereina į būseną s_3 laiko momentu t_3 ir įėjimo įvykis pasirodo laiku $t_3 + e$ su reikšme x_1 , nauja būsena apskaičiuojama taip $s_4 = \delta_{\text{ext}}(s_3, e, x_1)$ (čia $ta(s_3) > e$). Tokiu atveju sakoma, kad sistema įvykdė išorinį perėjimą (*External Transition*). Funkcija δ_{ext} ($\delta_{\text{ext}} : S \times R_0^+ \times X \rightarrow S$) vadinama

išorinio perėjimo funkcija (*External Transition Function*). Nėra jokio išėjimo signalo vykdant išorinį perėjimą.

PLA formalizmas yra analogiškas DEVS formalizmui. Tiek PLA, tiek DEVS aprašo diskrečių būsenų sistemas, kurių vidiniai įvykių laikas aprašomas tolydžiosiomis laiko komponentėmis. PLA metodas turi papildomą savybę, kurios neturi DEVS, nes jis gali išorinio įvykio metu generuoti išėjimo signalą. Dėl šios priežasties, bet koks DEVS modelis gali būti vienareikšmiškai aprašytas PLA kalba, tačiau atvirkštinis perversimas ne visada yra tiesioginis. Jei PLA modelyje egzistuotų tokie būsenų perėjimai, kurių metu esant išoriniam įvykiui išduodamas išėjimo signalas, reikalinga sukurti papildomų tarpinių būsenų, kad šis išėjimo signalas būtų generuojamas vidinio įvykio metu. Po tokio modelio koregavimo, jį galima aprašyti DEVS kalba.

3 Hibridinės sistemos modelis

3.1 Kvantuotų būsenų sistemos modelis

3.1.1 Matematinis modelis

Kvantuotų būsenų sistema (*Quantized State System* – QSS) yra skirta aprašyti ir spręsti pirmos eilės diferencialines lygtis. Šis modelis tinka ne tik paprastom diferencialinėms lygtims, bet tokios, kurios turi diskrečių komponentų, t.y. komponentų, kurių reikšmė gali pasikeisti šuoliškai per be galo trumpą laiko tarpą. Šis metodas buvo pasiūlytas naudoti DEVS aplinkoje [4]. Šiame dokumente QSS bus aprašytas agregatiniame metode. Taip pat bus pateiktas paprastos elektroninės schemos pavyzdys, ir kaip ją aprašyti diferencialine lygtimi, bei kaip jai sudaryti QSS modelį.

QSS modelis sudarytas iš dviejų pagrindinių elementų: funkcijos reikšmę skaičiuojančio integratoriaus, ir funkcijos reikšmės kvantavimo funkcijos. Kvantavimui panaudota speciali funkcija, pasižyminti histerezės savybėmis. Histerezė reikalinga tam, kad būtų išvengta situacijų, kai per baigtinį laiko tarpą įvyksta begalinis būsenų skaičius:

Diskretizavimo funkciją su histereze galima apibrėžti taip:

Tarkim, kad $Q = \{Q_0, Q_1, \dots, Q_r\}$ - realiųjų skaičių aibė tokia, kad $Q_{k-1} < Q_k$, su visais $1 \leq k \leq r$. Tegul Ω - aibė atkarpomis tolydžių trajektorijų, ir $x \in \Omega$ - atkarpomis tolydi trajektorija. Tegul $b: \Omega \rightarrow \Omega$ žymėsime funkciją, kuri pagal x reikšmę parenka kvantuotą reikšmę ir $q = b(x)$, čia trajektorija q tenkina tokia sąlyga:

$$q(t) = \begin{cases} Q_m, & \text{jei } t = t_0 \\ Q_{j+1} & \text{jei } x(t) = Q_{j+1} \wedge q(t^-) = Q_j \wedge j < r \\ Q_{j-1} & \text{jei } x(t) = Q_j - \varepsilon \wedge q(t^-) = Q_j \wedge j > 0 \\ q(t^-) & \text{kitu atveju} \end{cases} \quad (1a)$$

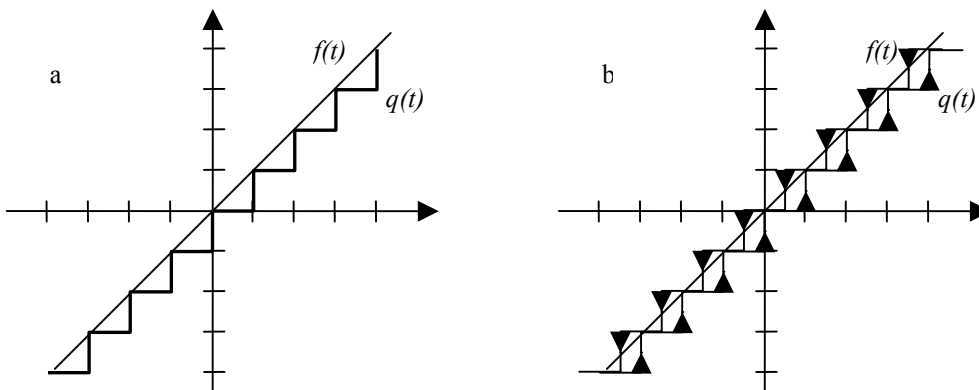
ir

$$m = \begin{cases} 0 & \text{jei } x(t_0) < Q_0 \\ r & \text{jei } x(t_0) \geq Q_r \\ k & \text{jei } Q_k \leq x(t_0) < Q_{k+1} \end{cases} \quad (1b)$$

$$\Delta Q = Q_{i+1} - Q_i \quad (1c)$$

Diskretizavimo funkcija su histereze pasižymi „atmintimi“, t. y. funkcijai $f(x)$ pasiekus

sekančią aukštesnę slenkstinę reikšmę, diskretizavimo funkcija $q(t)$ gali grįžti į ankstesnę žemesnę būseną tik tada, jei funkcija $f(x)$ sumažėja per tam tikrą dydį. Ši savybė leidžia išvengti nepageidaujamo būsenų švytavimo dideliu dažniu.



0 pav. a) Paprasta diskretizavimo funkcija, b) diskretizavimo funkcija su histereze

Norėdami aprašyti kvantuotą būsenų sistemą, pradėkime nuo paprastos diferencialinės lygties. Duota lygtis (2) aprašo procesą, kurio diferencialas priklauso nuo sistemos būsenos vektoriaus $x(t)$ ir nuo įėjimo signalo vektoriaus $u(t)$. Šį signalą generuoja specialiai tam skirtas signalų generatoriaus modelis, pateiktas 3.3 skyriuje. Įėjimo signalas ir yra diskrečioji sistemos dalis – keičiantis jo reikšmei diferencialo reikšmė kinta šuoliškai:

$$\frac{dx(t)}{dt} = f(x(t), u(t)) \quad (2)$$

kur:

$x \in R^n$, $u \in R^m$ ir $f: R^n \rightarrow R^n$, QSS modelis gaunamas apksimavus šią sistemą:

$$\frac{dx(t)}{dt} = f(q(t), u(t)) \quad (3)$$

kur:

$q(t)$ yra susieta su $x(t)$ per diskretizavimo funkciją su histereze (1).

QSS modelio integratorių galima apibūdinti kaip objektą, kuris priima skaičiuojamos funkcijos diferencialą ir skaičiuoja funkcijos reikšmę apibrėžtame reikšmių tinklelyje, o jo veikimas apibūdinamas (3) lygtimi. Objekto būseną priklauso nuo pradinės jo būsenos ir įeinančios diferencialo reikšmės. Jei išorinis signalas nesikeičia, sistema veikia kaip normali diferencialinių lygčių sistema. Pasikeitus įėjimo signalui, diferencialo reikšmę aprašanti išraiška irgi kinta (3). Nepakeistą QSS modelio specifikaciją, aprašyta DEVS kalba galima rasti 1 priede.

3.1.2 QSS modelio savybės

Diskrečios sistemos kartais gali patekti į būsenas, kai per labai trumpą laiko tarpą įvyksta be galo daug būsenų pasikeitimų [5:21]. Tokios situacijos yra dažnos diskrečių įvykių formalizmuose. Šiaip problemai išspęsti buvo panaudoja histerezės funkcija (1) kvantuojant funkcijos reikšmę. Parodyta, kad QSS modelis niekada nepatenka į tokias būsenas, kai bet be galo trumpą laiką įvyksta be galo daug būsenų pasikeitimų [5:31-32].

Vertinant QSS modelį iš trikdžių teorijos perspektyvos, yra parodyta [5:42-44], kad modelis gali patekti į būseną, kai funkcijos reikšmė pradeda svyruoti apie pusiausvyros tašką, tačiau apibrėžtose ribose.

QSS modelio tikslumas priklauso nuo parinkto funkcijos diskretizavimo tinklelio smulkumo. Yra parodyta [5:44], kad galima parinkti pakankamai mažą kvantavimo žingsnį ΔQ , norint gauti norimą funkcijos rezultato tikslumą, arba kad sistema konverguotų į norimą reikšmių sritį [5:47].

3.1.3 PLA modelis

Kaip paminėta diskrečiųjų sistemų skyriuje 2.2, bet kokį DEVS modelį galima lengvai perrašyti į atkarpomis tiesinį agregatą. Toliau pateiksime standartinio QSS modelio aprašą agregatiniame metode. Kvantuotų būsenų sistemos modelį DEVS kalboje galima rasti 1 priede. Palyginus DEVS ir PLA aprašus nesunku įsitikinti, kad jie aprašo vienodo veikimo modelius:

QSS agregatinė specifikacija (A1)

1. Įėjimo signalų aibė $X = x_v(t_m) \in R$ - skaičiuojamos funkcijos diferencialas,
2. Išėjimo signalų aibė $Y = Q_j(t_m), j = 1K r$ - funkcijos kvantuota reikšmė,
3. Išorinių įvykių aibė $E' = \{e'_1\}$,
čia: e'_1 – atėjo nauja diferencialo reikšmė,
4. Vidinių įvykių aibė $E'' = \{e''_1\}$,
čia: e''_1 – sistema pasiekė sekantį funkcijos kvantinį slenkstį,
5. Diskrečioji agregato būsenos dedamoji $v(t_m) = \{x(t_m), d_x(t_m), j(t_m)\}$,
čia $x(t_m) \in R$ – apskaičiuota funkcijos reikšmė,
 $d_x(t_m) \in R$ – diferencialo reikšmė,

$j(t_m) \in Z$ – funkcijos kvantuotos būsenos numeris,

6. Tolydžioji būsenos dedamoji $z_v(t_m) = \{w(e_1'', t_m)\}$ – laiko momentas, kada funkcija pasieks naują kvantinį slenkstį,

$$w(e_1'', t_m) = \begin{cases} < \infty, d_x(t_m) \neq 0 \\ \infty \text{ priešingu atveju} \end{cases}$$

7. Valdymo sekos $e_1' \alpha \{\sigma_1\}$, $e_1'' \alpha \{\sigma_2\}$,

čia:

σ_1 – yra laiko tarpas, per kurį funkcija pasieks sekančią kvantuotą reikšmę po išorinio įvykio, σ_2 – yra laiko tarpas, per kurį funkcija pasieks sekančią kvantuotą reikšmę po vidinio įvykio,

$$\sigma_1 = \begin{cases} \frac{Q_{j(t_m)+1} - (x(t_m) + (t_{m+1} - t_m) \cdot d_x(t_m))}{x_v} & \text{kai } x_v > 0 \\ \frac{(x(t_m) + (t_{m+1} - t_m) \cdot d_x(t_m)) - (Q_{j(t_m)-1} - \varepsilon)}{|x_v|} & \text{kai } x_v < 0 \\ \infty & \text{kai } x_v = 0 \end{cases}$$

ir

$$\sigma_2 = \begin{cases} \frac{Q_{j(t_m)+2} - (x(t_m) + (t_{m+1} - t_m) \cdot d_x(t_m))}{d_x(t_m)} & \text{kai } d_x(t_m) > 0 \\ \frac{(x(t_m) + (t_{m+1} - t_m) \cdot d_x(t_m)) - (Q_{j(t_m)-1} - \varepsilon)}{|d_x(t_m)|} & \text{kai } d_x(t_m) < 0 \\ \infty & \text{kai } d_x(t_m) = 0 \end{cases}$$

kai:

ε – histerezės langas³,

Q_1, Q_2, \dots, Q_r – funkcijos diskretizavimo tinklelis,

$Q_k - Q_{k-1} = \Delta Q$ – funkcijos kvantas,

8. Pradinė būseną $v(t_0) = \{x(t_0), d_x(t_0), j(f(x(t_0)))\}$,

$$z_v(t_0) = \{t_0 + \sigma_2\},$$

³ **Histerezės langas** yra fiksuotas dydis, kiek turi sumažėti funkcijos reikšmė nuo slenkstinės reikšmės, kad jos kvantuota reikšmė grįžtų į žemesnę būseną. Jis yra reikalingas tam, kad būtų išvengta funkcijos didelio dažnio virpesių esant labai mažiems funkcijos pokyčiams.

9. Perėjimo ir išėjimo operatoriai:

$H(e_1'(x_v))$: / pasikeitė diferencialo reikšmė /

$$x(t_{m+1}) = x(t_m) + (t_{m+1} - t_m) \cdot d_x(t_m)$$

$$d_x(t_{m+1}) = x_v$$

$$j(t_{m+1}) = j(t_m)$$

$$w(e_1', t_{m+1}) = t_m + \sigma_1$$

$H(e_1'')$: / pasiekta sekanti funkcijos reikšmė /

$$x(t_{m+1}) = x(t_m) + (t_{m+1} - t_m) \cdot d_x(t_m)$$

$$d_x(t_{m+1}) = d_x(t_m)$$

$$j(t_{m+1}) = j(t_m) + \text{sgn}(d_x(t_m))$$

čia:

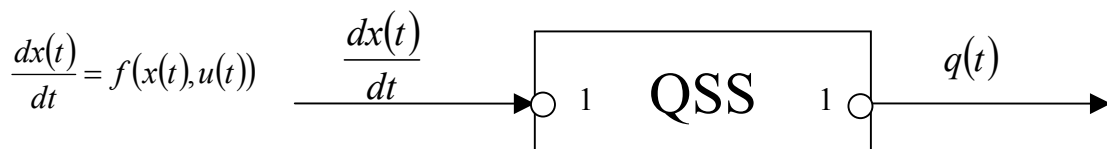
$$\text{sgn}(v) = \begin{cases} 1 & \text{kai } v > 0 \\ -1 & \text{kai } v < 0 \end{cases}, \text{ jei įvyko vidinis įvykis, vadinasi } v = d_x(t_m) \neq 0$$

$$w(e_1'', t_{m+1}) = t_m + \sigma_2$$

$G(e_1'')$:

$$y = Q_{j(t_m) + \text{sgn}(d_x(t_m))}$$

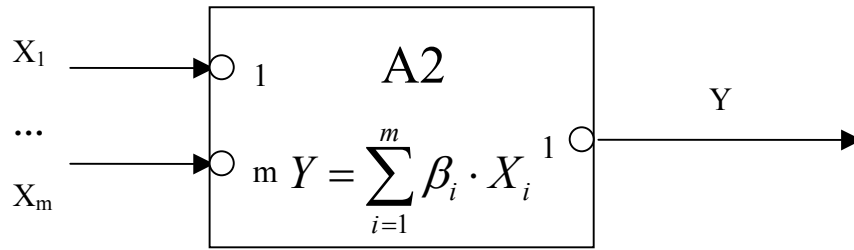
Pateikta agregato specifikacija pilnai aprašo QSS modelio veikimą, esant išoriniams signalams, ir kai sistema veikia autonomiškai, kaip tolyti diferencialinių lygčių sistema. Atėjus išoriniam signalui, sistema pradeda skaičiuoti naują trajektoriją, kuri priklauso nuo atėjusio signalo reikšmės. Jei diferencialas yra mažas, sistemos vidiniai įvykiai generuojamai retai, ir tai gali sumažinti skaičiavimo žingsnių skaičių atliekant imitavimą.



0 pav. QSS modelio – „integratoriaus“ – diagrama

3.2 Funkcijos diferencialo apskaičiavimas

Prieš tai pateiktas QSS agregatas naudoja jau apskaičiuotą diferencialo reikšmę. Todėl reikalingas atskiras statinis objektas, kuris iš valdymo signalų ir funkcijų reikšmių suskaičiuotų diferencialo reikšmę. Tam tikslui yra sudaromas „sudėtuvo“ agregatas (3 pav.):



0 pav. sudėtumo agregato diagrama

„Sudėtuvo“ agregatas apskaičiuoja algebrinę išraišką, kurią sudaro suma įėjimo signalų, padaugintų už iš anksto apibrėžtų koeficientų. Įėjimo signalai – tai valdymo signalai ir funkcijos reikšmės, iš kurių yra suskaičiuojamas funkcijos diferencialas.

„Sudėtuvo“ agregatinė specifikacija (A2), bendrasis atvejis

1. Įėjimo signalų aibė $X = \{\{X^1\}, \{X^2\}, K, \{X^m\}, K\}$ – skaičiuojamos sumos dėmenys, čia $X^i \subseteq R, i = 1K m$ – sumos dėmenys,
 $K = \{1K m\}$ – atėjusio operando numeris,
2. Išėjimo signalų aibė $Y = S(t_m)$ – suskaičiuota suma,
3. Išorinių įvykių aibė $E' = \{e'_1\}$,
čia: e_1 – pasikeitęs sumos operandas,
4. Vidinių įvykių aibė $E'' = \emptyset$ – agregatas yra statiškas, ir generuoja išėjimus tik išorinių įvykių atveju,
5. Diskrečioji agregato būsenos dedamoji
 $v(t_m) = \{\alpha_1(t_m), \alpha_2(t_m), K, \alpha_m(t_m), \beta_1, \beta_2, K, \beta_m, S(t_m)\}$,
čia $a_i(t_m) \in R, i = 1K m$ – sumos operandai,
 $\beta_i(t_m) \in R, i = 1K m$ – iš anksto nustatyti sumos operandų koeficientai,
 $S(t_m)$ – suskaičiuota suma,
6. Tolydžioji būsenos dedamoji $z_v(t_m) = \infty$ – agregatas vidinių įvykių neturi,
7. Valdymo sekos – \emptyset ,
8. Pradinė būsena:
 $v(t_m) = \{\alpha_1(t_0), \alpha_2(t_0), K, \alpha_n(t_0), \beta_1, \beta_2, K, \beta_n, S(t_0)\} = \{0, 0, K, 0, \lambda_1, \lambda_2, K, \lambda_m, 0\}$,
 $z_v(t_0) = \infty$,

9. Perėjimo ir išėjimo operatoriai:

$$\begin{aligned}
 & H(e'_1(x_k, k)): \\
 & a_k(t_{m+1}) = x_k \\
 & a_i(t_{m+1}) = a_i(t_m), \quad 1 \leq i \leq n, \quad i \neq k \\
 & S(t_{m+1}) = \sum_{i=1}^m a'_i \cdot \beta_i \\
 & \text{čia } a'_i = \begin{cases} a_i(t_m), & i \neq k \\ x_k, & i = k \end{cases} \quad 1 \leq i \leq n \\
 & z_v(t_{m+1}) = \infty \\
 & G(e''_1): \\
 & Y = \sum_{i=1}^m a'_i \cdot \beta_i
 \end{aligned}$$

Verta pastebėti, kad šį agregatą perrašant į DEVS modelį, reikia sukurti papildomą tarpinę būseną, išėjimo signalui generuoti. Tam atvejui pateikiame pakoreguotus specifikacijos punktus, aprašančius modelį, kurį galima būtų paprastai perrašyti į DEVS formalizmą:

4. Vidinių įvykių aibė $E'' = \{e''_1\}$ – fiktyvus vidinis įvykis, išėjimo signalui generuoti,
7. Valdymo sekos – $e'_1 \alpha \{\sigma\}$,
9. Perėjimo ir išėjimo operatoriai:

$$\begin{aligned}
 & H(e'_1(x_k, k)): && / \text{ atėjo sumos operandas } / \\
 & a_k(t_{m+1}) = x_k \\
 & a_i(t_{m+1}) = a_i(t_m), \quad 1 \leq i \leq n, \quad i \neq k \\
 & S(t_{m+1}) = \sum_{i=1}^m a'_i \cdot \beta_i \\
 & \text{čia } a'_i = \begin{cases} a_i(t_m), & i \neq k \\ x_k, & i = k \end{cases} \quad 1 \leq i \leq n \\
 & z_v(t_{m+1}) = 0 && / \text{ po signalo atėjimo iš karto turi sekti vidinis } / \\
 & && / \text{ įvykis, išėjimo signalui išduoti } / \\
 & H(e''_1): && / \text{ fiktyvus vidinis įvykis signalui generuoti } /
 \end{aligned}$$

$$\begin{aligned}
a_i(t_{m+1}) &= a_i(t_m), \quad 1 \leq i \leq n \\
S(t_{m+1}) &= S(t_m) \\
z_v(t_{m+1}) &= \infty \\
G(e_1'') &: \\
Y &= S(t_m)
\end{aligned}$$

4 ir 9 specifikacijos punktai atspindi, kokie pakeitimai reikalingi, norint duotąjį PLA modelį A2 perrašyti į DEVS modelį. Kaip matome, kiekvienam išėjimui, kuris turėtų būti išduodamas išorinio įvykio metu, reikalinga sukurti po vieną fiktyvų vidinį įvykį. Įvykdžius šiuos pakeitimus, galima bet kokią PLA modelį paprastai perrašyti DEVS formaliąja kalba.

3.3 Valdymo signalų generavimas

Realiose sistemose valdymo signalai gaunami iš žmogaus operatoriaus arba iš specialaus valdymo kontrolierio. Sudarant hibridinės sistemos modelį, valdymo signalus gauti reikalingas atskiras objektas, kuris laiko momentais $t_1, t_2, \dots, t_i, t_{i+1}, \dots, t_n$ generuotų atitinkamai reikšmes $v_1, v_2, \dots, v_i, v_{i+1}, \dots, v_n$ [5:36]. Tai gali būti iš anksto nustatyta signalų seka, ar pagal tam tikrą taisyklę generuojami reikšmės. Toliau pateikta tokio agregato prototipo specifikacija.

Generatoriaus agregatinė specifikacija (A3)

1. Įėjimo signalų aibė – $X = \emptyset$,
2. Išėjimo signalų aibė – $Y = \{v_1, v_2, \dots, v_n\}$,
3. Išorinių įvykių aibė $E' = \emptyset$,
4. Vidinių įvykių aibė $E'' = \{e_1\}$ - sekanti generatoriaus būseną,
5. Diskrečioji agregato būsenos dedamoji $v(t_m) = \{j\}, j = 1K n$,
6. Tolydžioji agregato būseną $z_v(t_m) = w(e_1, t_m)$ - laikas iki būsenos pasikeitimo,
7. Valdymo sekos $e_1 \alpha \{\sigma\}$,
 σ - laiko tarpas iki sekančios būsenos,
8. Pradinė būseną $v(t_0) = \{v_1\}$:
 $z_v(t_0) = \{t_0 + \sigma\}$
čia:
 $\sigma = t_1 - t_0$

9. Perėjimo ir išėjimo operatoriai:

$$H(e_1):$$

$$j(t_{m+1}) = j(t_m) + 1$$

$$w(e_1, t_{m+1}) = t_m + \sigma$$

čia:

$$\sigma = t_{m+1} - t_m$$

$$G(e_1):$$

$$Y = v_j$$

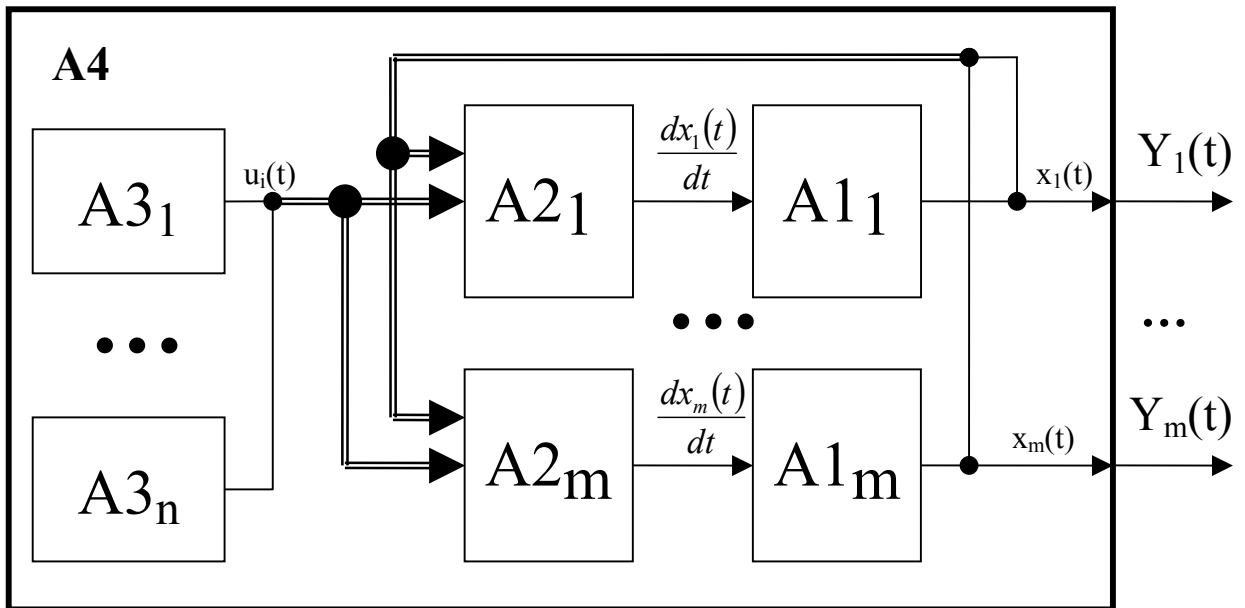
3.4 Bendras hibridinės sistemos modelis

Naudojantis ankščiau pateiktais QSS, „sudėtuvo“ ir „generatoriaus“ agregatais, galima sudaryti jungtinį hibridinės sistemos agregatą. Jis susideda iš valdymo signalo generatorių, diferencialo apskaičiavimo agregatų ir funkcijos apskaičiavimo agregatų (pav. 4).

Vienai hibridinės sistemos diferencialinei lygčiai aprašyti reikia:

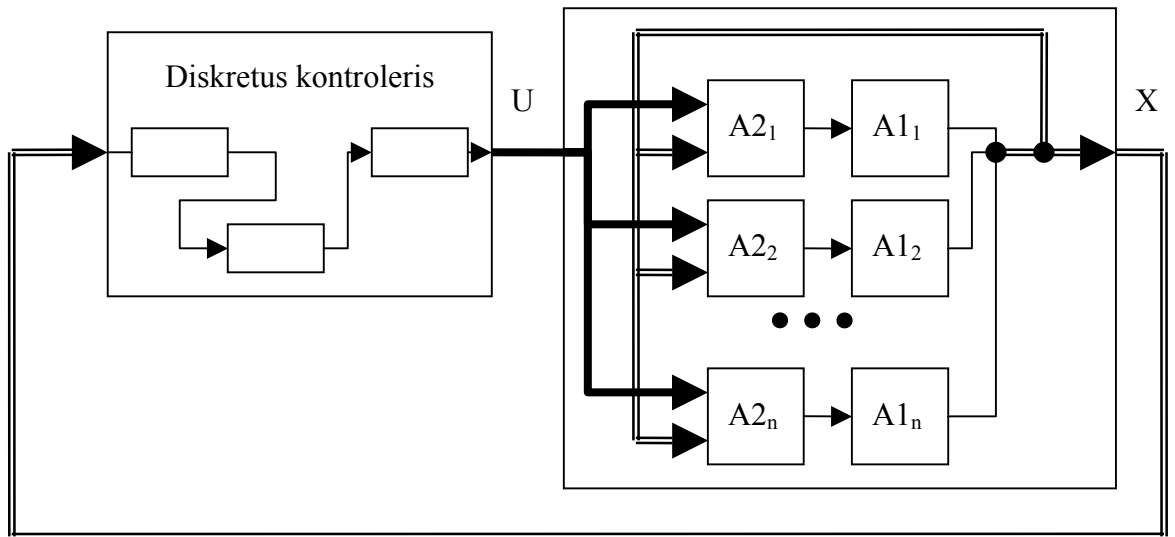
- Generatoriaus agregato, kuris generuoja valdymo signalus, kurie šuoliškai keičia funkcijos diferencialą,
- „Sudėtuvo“ agregato, kuris iš funkcijos kvantuotos reikšmės ir generatoriaus agregato gautų signalų apskaičiuoja funkcijos diferencialą,
- „Integratoriaus“ agregatas, kuris, sujungtas su sudėtuvo agregato išėjimu, skaičiuoja funkcijos reikšmę.

Jei modeliuojamos sistemos būseną susideda iš keleto komponentų, kurių reikšmių trajektorijos randamos diferencialinių lygčių pagalba, bendrą jungtinį modelį sudaro sumatorių ir integratorių masyvai, kurių kiekvienas aprašo po vieną diferencialinę lygtį.



0 pav. Bendros hibridinės sistemos modelio diagrama

4 pav. parodyta bendra hibridinės sistemos modelio diagrama. Modelis sudarytas iš n generatorių $A3$, kurie generuoja valdymo signalus u . „Sudėtuvai“ $A2$ iš valdymo signalų ir funkcijų reikšmių apskaičiuoja funkcijų diferencialus. Integratoriai $A1$ apskaičiuoja kvantuotas funkcijų reikšmes. Pastorinta linija reiškia kad perduodamas duomenų vektorius, t. y. reikia traktuoti kaip keletą „laidų“ abstrakciją. Jei agregatą yra sujungtas su pastorinta linija, tai reiškia, kad jis gali būti sujungtas su keletu įėjimų. Valdymo signalų generatoriais gali būti tiek atominiai agregatai, generuojantis tam tikras reikšmes, tiek junginiai agregatai, sudarantys kitos sistemos modelį. Dažnai valdymo signalai negali būti paprastai generuojami iš anksto determinuota tvarka, o yra sudėtingo proceso, susijusio su visos sistemos veikimu, rezultatas. 5 pav. parodyta kaip atrodytų realus hibridinės sistemos modelis, esant sudėtinei sistemai, kuri sudaryta iš sistemos valdančio diskretaus kontrolerio, ir tolydaus proceso komponentės.



0 pav. Uždaro ciklo hibridinės sistemos modelio diagrama

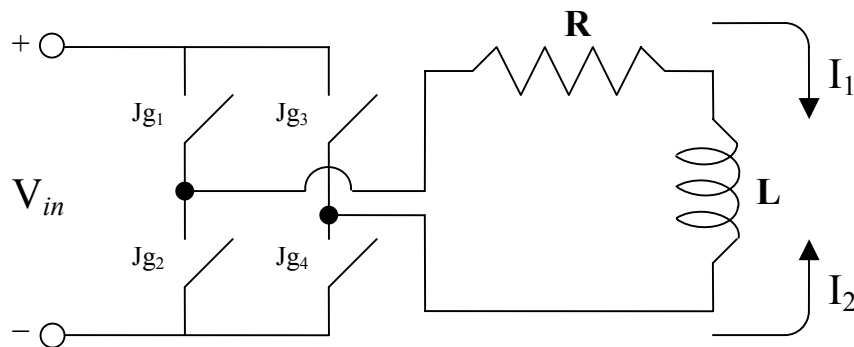
5 pav. diagrama yra pavyzdys galima sudaryti sudėtingos sistemos modelius, sudarytus iš tarpusavyje sujungtų diskrečiųjų ir tolydžiųjų sistemos komponentių. Hibridinę sistemą (5pav.) sudaro diskretus kontroleris, sujuntas su tolydžiąja komponente. Priklausomai nuo gaunamo signalo X , kontroleris išduoda valdymo signalus U . Duotas modelio pavyzdys galėtų aprašyti gamyklos modelį, kur tam tikras gamybos procesas yra valdomas diskretaus valdiklio.

4 Kvantuotų būsenų sistemos panaudojimas hibridinėms sistemoms modeliuoti

4.1 Elektroninės schemos aprašymo pavyzdys

Šiame pavyzdyje aprašysime DC/AC⁴ srovės keitėją, prijungtą prie elektros grandinės, turinčios R varžą ir L induktyvumą (4 pav.). Jungikliai yra diskrečioji sistemos dalis, o prijungta grandinė – tolydžioji. Jungiklių padėtis lemia, kuria kryptimi gali tekėti srovė. Jei yra įjungti 1 ir 4 jungikliai, elektros srovė teką grandine I₁ kryptimi, jei 2 ir 3 – I₂ kryptimi. Duotojoje (4) lygtyje matome, kad hibridinę sistemą sudaro tolydi dalis, aprašyta $-(R/L) \cdot i_L$ komponente, ir diskreti dalis, jungikliai, kurių padėtis išreikšta kintamuoju j_g . Kai $j_g = 1$, reiškia 1 ir 4 įjungtus jungiklius, kai $j_g = -1$, reiškia 2 ir 3 įjungtus jungiklius:

$$\frac{di}{dt} i_L = -\frac{R}{L} i_L + j_g \cdot V_{in} \quad (4)$$



0 pav. DC/AC srovės keitėjas su prijungta grandine

4.2 Hibridinės sistemos agregatai

Toliau parodysime, kaip gauti diferencialo reikšmę naudojant „sudėtuvo“ agregatą. Remiantis duota diferencialine lygtimi (4), sudarome „sudėtuvo“ agregatą su dvejais įėjimais (A2, m=2) funkcijos diferencialui apskaičiuoti:

$$\frac{di}{dt} i_L = \beta_1 \cdot X_1 + \beta_2 \cdot X_2 \quad (5)$$

kur:

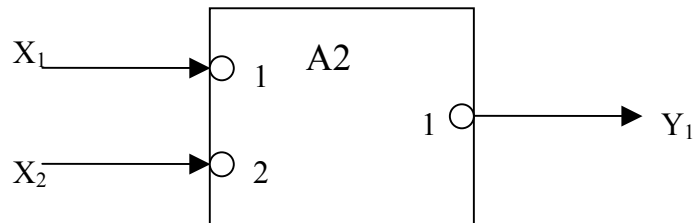
⁴ AC / DC (an. alternating current/direct current) – kintanti srovė / nuolatinė srovė.

$$\beta_1 = -\frac{R}{L}$$

$$\beta_2 = V_{in}$$

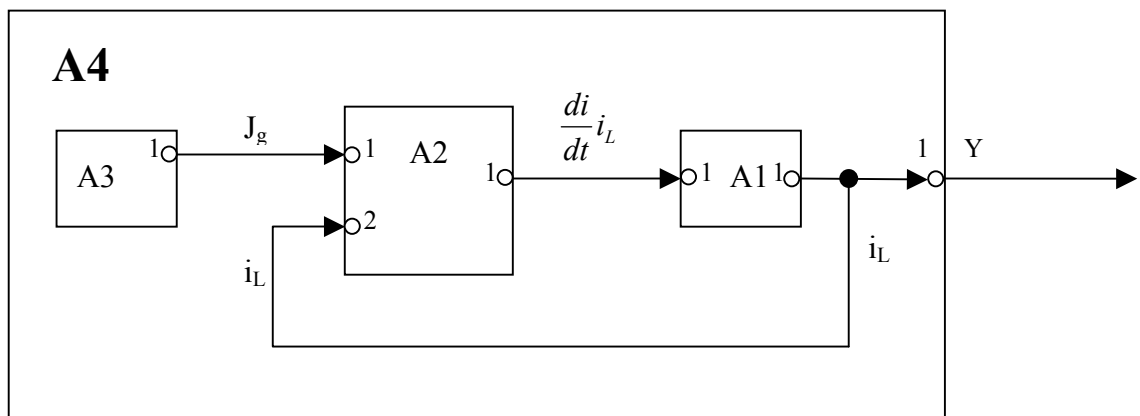
$$X_1 = i_L$$

$$X_2 = j_g$$



0 pav. „Sudėtuvo“ agregatas (5) išraiškai

Agregatas yra šatiškas, ir išduoda signalus tik tada, kai pasikeičia vienas iš įėjimo signalų. Turėdami paruošta „sudėtuvo“ agregatą, galime sudaryti modelį (4), „sudėtuvą“ sujungdami su „integratoriaus“ agregatu. Valdymo signalas generuoti pasirinktas paprastas signalų sekos generatorius, kuris generuoja reikšmes 1, -1, 1 vienos sekundės periodu:



0 pav. Duoto pavyzdžio elektroninės grandinės agregatinio modelio diagrama

Jungtinį agregatą sudaro signalų „generatorius“ A3, funkcijos diferencialo reikšmę apskaičiuojantis „sudėtuvas“ A2 ir „integratorius“ A1, skaičiuojantis funkcijos reikšmę.

4.3 Modeliavimo rezultatai

Remiantis pateikta grandinės schema (6 pav.) buvo sudarytas modelis (8 pav.). Sprendžiant uždavinį naudotos reikšmės:

$$R = 0.6 \Omega,$$

$$L = 100 \text{ mH},$$

Iš čia:

$$\beta^1 = -\frac{R}{L} = -6 \Omega/\text{H},$$

$$\beta^2 = V_{in} = 10 \text{ V}.$$

Diferencialinė lygtis įgauna tokią išraišką:

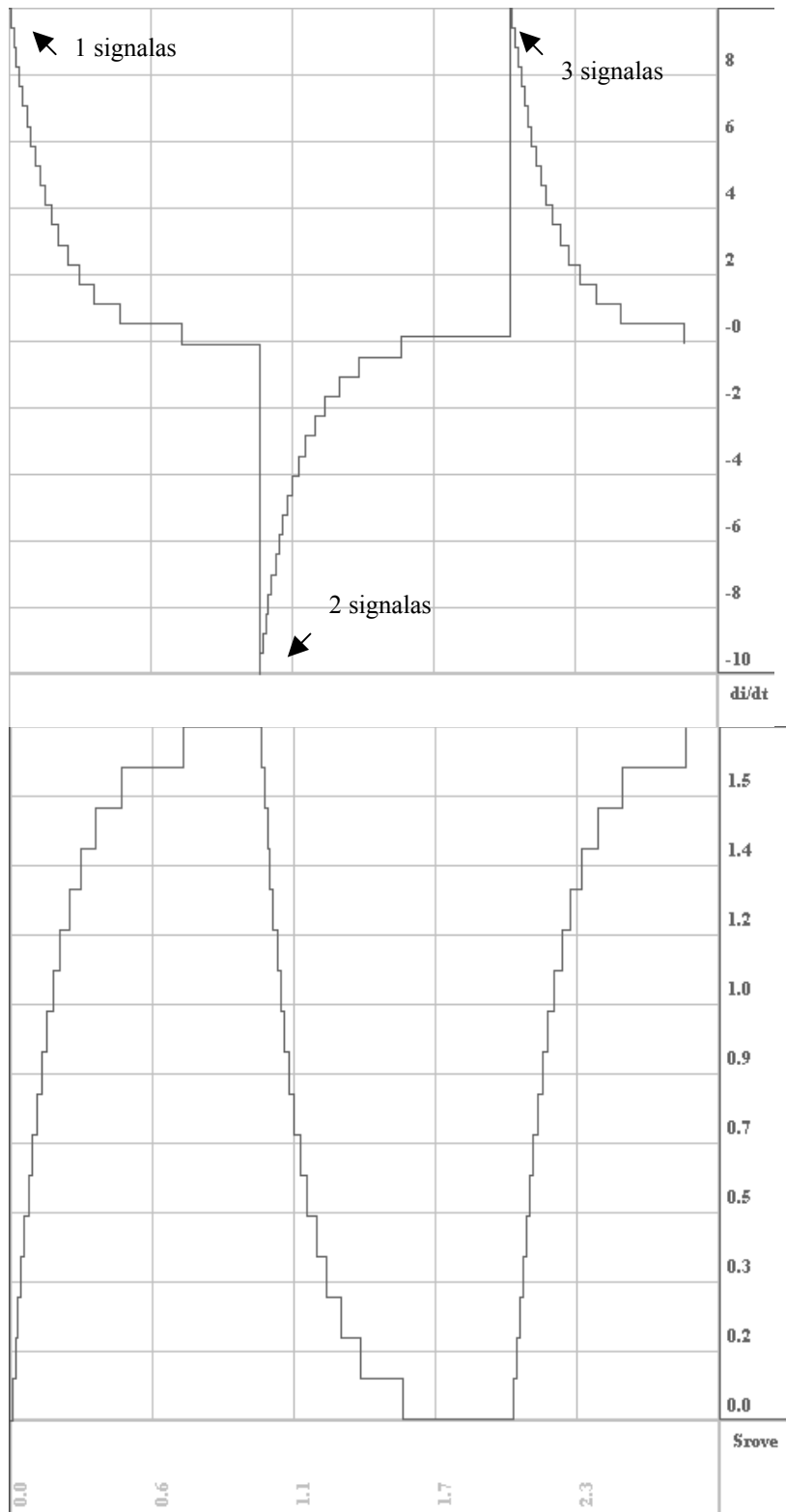
$$\frac{di}{dt} i_L = -6 \cdot i_L + 10 \cdot j_g \quad (6)$$

Papildomai parenkame „integratoriaus“ parametrus – funkcijos diskretizavimo tinklelį ir signalų generatoriaus dažnį:

$$\Delta Q = \varepsilon = \Delta i_L = 0.1 \text{ A (A1 modelio funkcijos reikšmės kvantas)}$$

$$T = 2 \text{ s (A3 modelio signalo pasikeitimo periodas)}$$

Skaičiavimo laiko tarpas – 3 s. Signalų generatorius sugeneravo 3 signalus (t, v) – (0, 1), (1, -1) ir (2, 1). Srovės kitimui apskaičiuoti prirėikė 104 žingsnių. 2 priede yra pateiktas modelio aprašymas *PowerDEVS* aplinkoje [4]. 9 pav. (a) dalyje matome, kaip šuoliškai pakinta funkcijos diferencialo reikšmė, pasikeitus jungiklių padėčiai, o (b) dalyje matome, kaip priklausomai nuo diferencialo keičiasi srovė.



0 pav. a) Funkcijas diferencālo reikšmēs trajektorija, b) funkcijas reikšmēs (srovēs stiprumas) trajektorija.

5 Kvantuotos būsenų sistemos modelio modifikacija

5.1 Reikalingi pakeitimai

Duotasis Kvantuotų būsenų sistemos modelis aprėpia tik dalį hibridinių sistemų, kurias galima modeliuoti. QSS modelis leidžia aprašyti sistemas, aprašytas pirmos eilės diferencialinėmis lygtimis, kurių diferencialai gali kisti šuoliškai. Tačiau kartais reikalinga, kad būtų galima šuoliškai keisti ne tik diferencialo reikšmę, tačiau ir pačios funkcijos reikšmę. Norint, kad QSS turėtų tokią savybę, reikalinga jį papildyti nauju išoriniu signalu. Šis papildomas signalas būtų naudojamas šuoliškam funkcijos reikšmės pakeitimui. Galimi įvairūs funkcijos įtakojimo variantai, funkcijos reikšmės padidinimas per perduotą reikšmę, funkcijos nustatymas į perduotą reikšmę arba funkcijos perkrovimas į numatytą pradinę reikšmę. Toliau pateiktoje specifikacijoje pasirinktas atvejis, kai funkcijos reikšmė padidinama per tam tikrą dydį.

5.2 Modifikuotas QSS modelis, leidžiantis keisti funkcijos reikšmę, aprašytas naudojant PLA formalizmą

Pakeisto QSS modelio specifikacija PLA formalizme (A1-M1)

- Įėjimo signalų aibė $X = \{x_v(t_m), y_r\}$,
čia: $x_v(t_m) \in R$ - skaičiuojamos funkcijos diferencialas,
 $y_r \in R$ - momentinio funkcijos pakeitimo reikšmė,
- Išėjimo signalų aibė $Y = Q_j(t_m)$, $j = 1K r$ - funkcijos kvantuota reikšmė,
- Išorinių įvykių aibė $E' = \{e'_1, e'_2\}$,
čia: e'_1 - atėjo pasikeitusi diferencialo reikšmė,
 e'_2 - atėjo signalas, keičiantis funkcijos reikšmę,
- Vidinių įvykių aibė $E'' = \{e''_1\}$,
čia: e''_1 - funkcija pasiekė kitą kvantinį slenkstį,
- Diskrečioji agregato būsenos dedamoji $v(t_m) = \{x(t_m), d_x(t_m), j(t_m)\}$
čia $x(t_m) \in R$ - apskaičiuota funkcijos reikšmė,
 $d_x(t_m) \in R$ - esama diferencialo reikšmė,
 $j(t_m) \in Z$ - funkcijos kvantuotos būsenos numeris,

6. Tolydzioji būsenos dedamoji $z_v(t_m) = \{w(e_1'', t_m)\}$ - laiko momentas, kada funkcija pasieks naują kvantinį slenkstį,

$$w(e_2'', t_m) = \begin{cases} < \infty, d_x(t_m) \neq 0 \\ \infty \text{ priešingu atveju} \end{cases}$$

7. Valdymo sekos $e_1' \alpha \{\sigma_1\}$, $e_1'' \alpha \{\sigma_2\}$, $e_2' \alpha \{\sigma_3\}$

čia:

σ_1 yra laiko tarpas, per kurį funkcija pasieks sekančią kvantuotą reikšmę po išorinio įvykio. σ_2 – laiko tarpas, per kurį funkcija pasieks sekančią kvantuotą reikšmę po vidinio įvykio, σ_3 – laiko tarpas, po funkcijos padidinimo:

$$\sigma_1 = \begin{cases} \frac{Q_{j(t_m)+1} - (x(t_m) + (t_{m+1} - t_m) \cdot d_x(t_m))}{x_v} & \text{kai } x_v > 0 \\ \frac{(x(t_m) + (t_{m+1} - t_m) \cdot d_x(t_m)) - (Q_{j(t_m)+1} - \varepsilon)}{|x_v|} & \text{kai } x_v < 0 \\ \infty & \text{kai } x_v = 0 \end{cases}$$

$$\sigma_2 = \begin{cases} \frac{Q_{j(t_m)+2} - (x(t_m) + (t_{m+1} - t_m) \cdot d_x(t_m))}{d_x(t_m)} & \text{kai } d_x(t_m) > 0 \\ \frac{(x(t_m) + (t_{m+1} - t_m) \cdot d_x(t_m)) - (Q_{j(t_m)+1} - \varepsilon)}{|d_x(t_m)|} & \text{kai } d_x(t_m) < 0 \\ \infty & \text{kai } d_x(t_m) = 0 \end{cases}$$

ir

$$\sigma_3 = \begin{cases} \frac{Q_{j(t_m) + \lfloor (t_{m+1} - t_m) d_x(t_m) + x_y / \Delta Q \rfloor + 1} - (x(t_m) + (t_{m+1} - t_m) \cdot d_x(t_m) + x_y)}{d_x(t_m)} & \text{kai } d_x(t_m) > 0 \\ \frac{(x(t_m) + (t_{m+1} - t_m) \cdot d_x(t_m) + x_y) - (Q_{j(t_m) + \lfloor (t_{m+1} - t_m) d_x(t_m) + x_y / \Delta Q \rfloor} - \varepsilon)}{|d_x(t_m)|} & \text{kai } d_x(t_m) < 0 \\ \infty & \text{kai } d_x(t_m) = 0 \end{cases}$$

8. Pradinė būsena:

$$v(t_0) = \{x(t_0), d_x(t_0), j(f(x(t_0)))\}$$

$$z_v(t_0) = \{t_0 + \sigma_2\}$$

9. Perėjimo ir išėjimo operatoriai:

$H(e'_1(x_v))$: / atėjo nauja diferencialo reikšmė /

$x(t_{m+1}) = x(t_m) + (t_{m+1} - t_m) \cdot d_x(t_m)$ / šis įvykis nepakeistas iš QSS/

$d_x(t_{m+1}) = x_v$

$j(t_{m+1}) = j(t_m)$

$w(e'_1, t_{m+1}) = t_m + \sigma_1$

$H(e'_2(x_y))$: / atėjo funkcijos pakeitimo reikšmė /

$x(t_{m+1}) = x(t_m) + (t_{m+1} - t_m) \cdot d_x(t_m) + x_y$ / funkcijos pakeitimas /

$d_x(t_{m+1}) = d_x(t_m)$

$j(t_{m+1}) = j(t_m) + \lfloor (t_{m+1} - t_m) \cdot d_x(t_m) + x_y / \Delta Q \rfloor$

$w(e'_1, t_{m+1}) = t_m + \sigma_2$

$G(e'_2): y = Q_{j(t_m) + \lfloor (t_{m+1} - t_m) \cdot d_x(t_m) + x_y / \Delta Q \rfloor}$

$H(e''_1)$:

$x(t_{m+1}) = x(t_m) + \sigma \cdot d_x(t_m)$ / vidinis įvykis, nepakeistas iš QSS /

$d_x(t_{m+1}) = d_x(t_m)$

$j(t_{m+1}) = j(t_m) + \text{sgn}(d_x(t_m))$

$w(e''_1, t_{m+1}) = t_m + \sigma_2$

$G(e''_1): y = Q_{j(t_m) + \text{sgn}(d_x(t_m))}$

5.3 Kitos galimos QSS modelio modifikacijos

Aukščiau pateikta A1-M1 modelio specifikacija yra vienas iš galimų funkcijos reikšmės keitimo variantų. Esant tam tikroms sistemų savybėms, gali būti reikalinga, kad funkcijos reikšmė būtų keičiama ir kitaip:

- Nustatoma į signalo pasiūstą reikšmę,
- Nustatoma į standartinę (pradinę) reikšmę,
- Kiti pakeitimo variantai, priklausomai nuo poreikio.

Norint aprašyti tokius modelius, reikėtų pakeiti σ_3 apskaičiavimo išraišką, taip pat e'_2 įvykio perėjimo operatorių. Kitos specifikacijos dalys lieka nepakitusios.

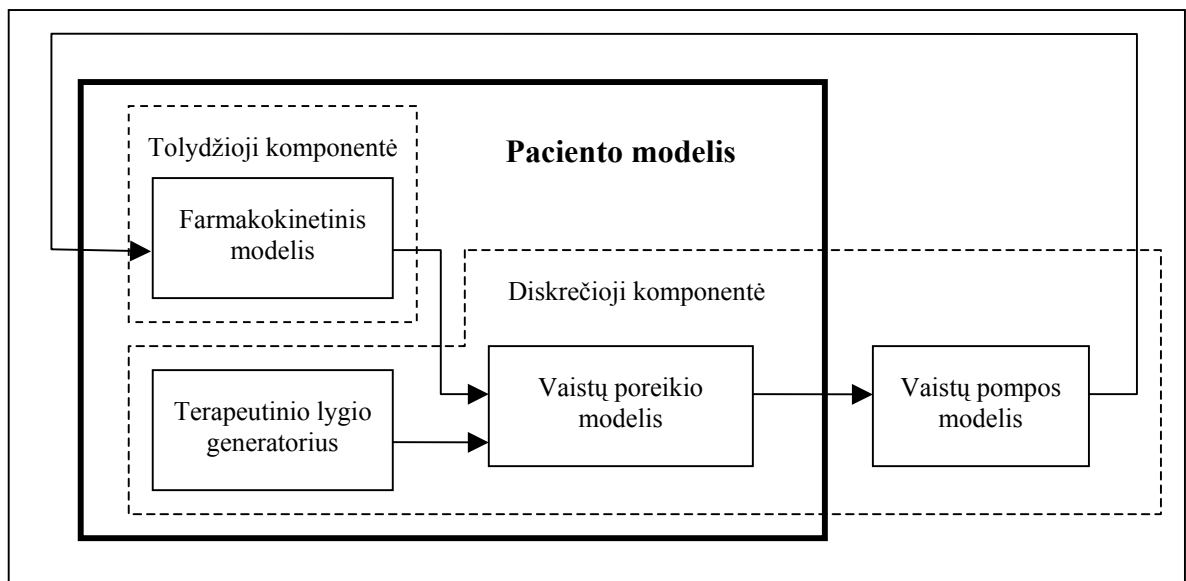
6 Pakeisto Kvantuotų būsenų sistemos modelio panaudojimas

6.1 PCA modelis

Norint aprašyti modelį sistemos, kuri imituota vaistų koncentraciją kraujyje, kai pacientas reguliuoja vaistų suleidimą, reikia tokių sistemos komponentų:

- Farmakokinetinio modelio, kuris skaičiuoja vaistų koncentraciją kraujyje,
- Paciento terapeutinio lygio generatoriaus, kuris nustato pageidaujamos vaistų koncentracijos viršutinę ir apatinę ribas,
- Paciento vaistų poreikio modelį, kuris, priklausimai nuo vaistų koncentracijos ir terapeutinio lygio, nusprendžia kaip dažnai pacientas pageidauja vaistų dozės,
- Vaistų pompos modelio, kuris priklausomai nuo vaistų pageidavimo, vykdo vaistų injekcijas.

10 pav. parodyta, kaip turėtų būti sujungti sistemos komponentai. Vaistų poreikio modelis valdo vaistų pompą, kuri vykdo vaistų injekciją. Vaistų injekcijos signalai perduodami farmakokinetiniam modeliui. Farmakokinetinis modelis skaičiuoja vaistų koncentraciją kraujyje. Terapeutinio lygio generatorius imituoja, kaip keičiasi organizmo reakcija į vaistus.



10 pav. Farmakokinetinis modelio, kai pacientas reguliuoja vaistų suleidimą, diagrama

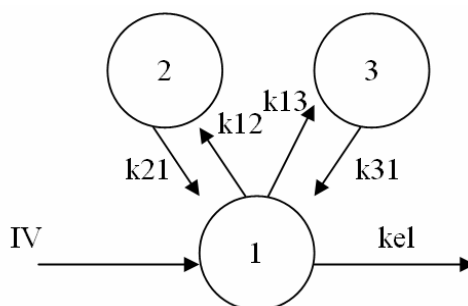
Farmakokinetinis modelis yra tolydžioji sistemos komponentė, nes jos veikimą aprašo

diferencialinių lygčių sistema. Likusieji modeliai sudaro diskrečiąją komponentę, kuri gali būti laiko sistemos kontrolieriu, valdančiu tolydžiąją komponentę.

6.2 Farmakokinetinis modelis

Farmakokinetika nagrinėja vaisto absorbcija, pasiskirstymą, metabolizmą ir eliminaciją iš organizmo. Naudojami vienos sekcijos arba keleto sekcijų modeliai. Kiekviena sekcija atvaizduoja tam tikrą žmogaus kūno dalį – audinių grupę, organą arba keletą organų. 10 pav. parodyta 2 sekcijų modelis. Linijos, jungiančios sekcijas, rodo medžiagų judėjimo kryptis tarp sekcijų. Koeficientai prie rodyklių rodo medžiagų srauto stiprumą – kokia medžiagos dalis pereina į kitą sekciją per laiko vienetą.

Šiame pavyzdyje naudosime modelį, kurį sudaro trys skyriai (10 pav.). Vaistai suleidžiami į 1 sekciją, laikoma, kad jų koncentracija sekcijoje padidėja šuoliškai.



0 pav. Trijų sekcijų farmakokinetinis modelis

Sudarome diferencialines lygtis šiam modeliui:

$$\begin{cases} \frac{dX_1}{dt} = k_{21} \cdot X_2 - k_{12} \cdot X_1 + k_{31} \cdot X_3 - k_{13} \cdot X_1 - kel \cdot X_1 \\ \frac{dX_2}{dt} = k_{12} \cdot X_1 - k_{21} \cdot X_2 \\ \frac{dX_3}{dt} = k_{13} \cdot X_1 - k_{31} \cdot X_3 \end{cases} \quad (7)$$

X_1 , X_2 ir X_3 yra atitinkamų sekcijų vaistų koncentracija kraujo plazmoje. Vaistų koncentracijos kitimas priklauso nuo vaistų mainų tarp sekcijų ir vaistų šalinimo, kuris priklauso nuo kel konstantos. Vaistų išvirkštymas IV diferencialinių lygčių sistemoje (7) nepažymėtas, nes jo metu funkcijos reikšmė apskaičiuojama nenaudojant dif. lygčių, o tiesiogiai padidinant funkcijos reikšmę per nustatytą dydį:

$$X_1' = X_1 + IV \quad (8)$$

Nagrinėjame pavyzdyje pagrindinis tikslas yra sukurti modelį, kuris reguliuotų vaistų koncentraciją 2-joje sekcijoje. Todėl X_2 funkcijos reikšmė bus naudojama vaistų suleidimams reguliuoti. Vaistai turi būti suleidžiami taip, kad jų koncentracija 2-joje sekcijoje būtų tam tikrose teraputinėse ribose.

6.3 Paciento modelis

Paciento modelį sudaro 3 komponentai: teraputinio lygio negeranorius, vaistų poreikio modelis ir farmakokinetinis modelis. Apjungti komponentai sudaro „virtualaus“ paciento modelį.

Teraputinio lygio generatoriui aprašyti pasirinkta sinusinė funkcija, pagal kurią nustatomas viršutinis ir apatinis teraputiniai lygiai:

$$\begin{aligned} f_{ter}(t) &= Amp \cdot \sin(t \cdot v) \\ T_V(t) &= f_{ter}(t) + V \\ T_A(t) &= f_{ter}(t) + A \end{aligned} \quad (9)$$

čia:

- Amp – svyravimų amplitudė,
- v – lygio kitimo dažnis,
- T_V – teraputinio lygio viršutinė riba,
- T_A – teraputinio lygio apatinė riba,
- V, A – baziniai viršutiniai ir apatiniai teraputiniai lygiai.

Paciento vaistų poreikio modelio veikimas aprašomas eksponentine funkcija, kuri nusprendžia, kaip ilgai pacientas lauks iki sekančio vaistų pareikalavimo:

$$L(C, C_A, C_V) = \begin{cases} \frac{e^{\frac{C-C_A}{C_V-C_A}+2}}{ex} + \xi & \text{kai } C \geq C_A \\ \frac{e^{\frac{2 \cdot C}{C_A}}}{ex} + \xi & \text{kai } C < C_A \end{cases} \quad (10)$$

čia:

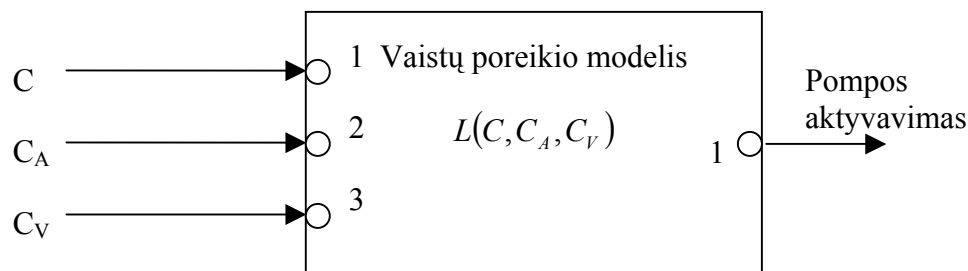
- C – vaistų koncentracija stebimoje sekcijoje,
- C_A – apatinė teraputinio lygio riba,

C_V – viršutinė teraputinio lygio riba,

ex – koeficientas funkcijos derinimui,

ξ - atsitiktinis dydis.

Laukimo laikas L priklauso nuo vaistų koncentracijos stebimoje sekcijoje ir teraputinio lygio. Farmakokinetinį modelį aprašysime 6.5 skyriuje.

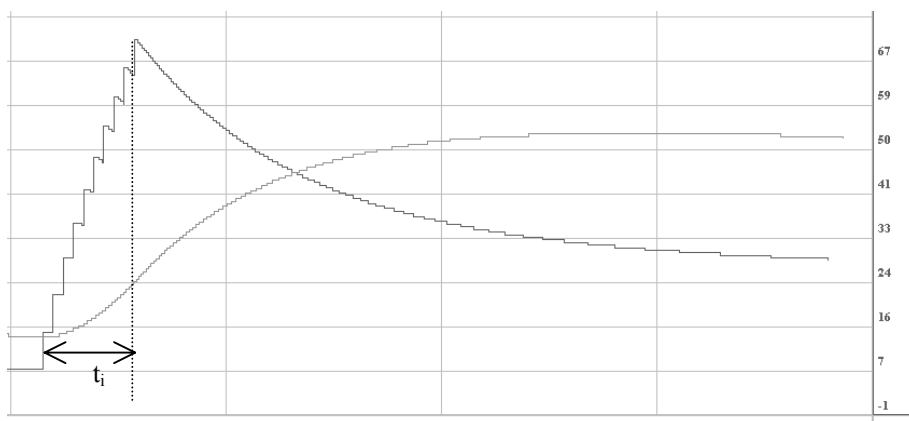


0 pav. Vaistų poreikio agregato diagrama

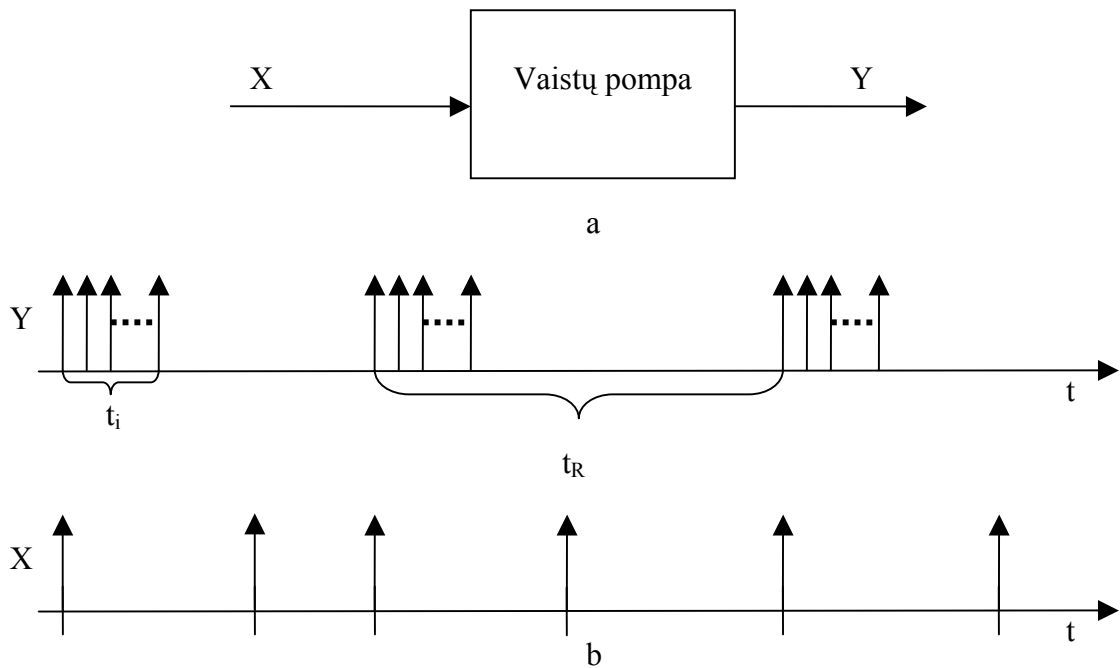
6.4 Vaistų pompos modelis

Vaistų pompa yra skirta vaistų suleidimui paciento kraujotakos sistemą. Kad pacientas neperdozuotų vaistų, yra nustatomas apribojimas, kaip dažnai pompa gali suleisti vaistus, kuris paprastai tai būna 10 min. laiko tarpas. Pacientui nuspaudus vaistų suleido mygtuką, pompa pradeda leisti vaistus, ir 10 min. nekreipia dėmesio į tolesnius paciento „signalus“ suleisti daugiau vaistų. Norint imituoti tolydų vaistų suleidimą, vaistų dozės suleidimas yra paskirstomas į keletą mažesnių dozių, paskirstytų laiko tarpe, pavyzdžiui 1 min. laikotarpyje.

14 pav. parodytas konceptualus vaistų pompos veikimas. t_i yra laiko tarpas, į kurį yra paskirstomas vaistų suleidimas, t_R – laiko tarpas, kurio metu pompa yra „užrakinta“ ir nereaguoja į paciento tolesnius prašymus suleisti vaistų.



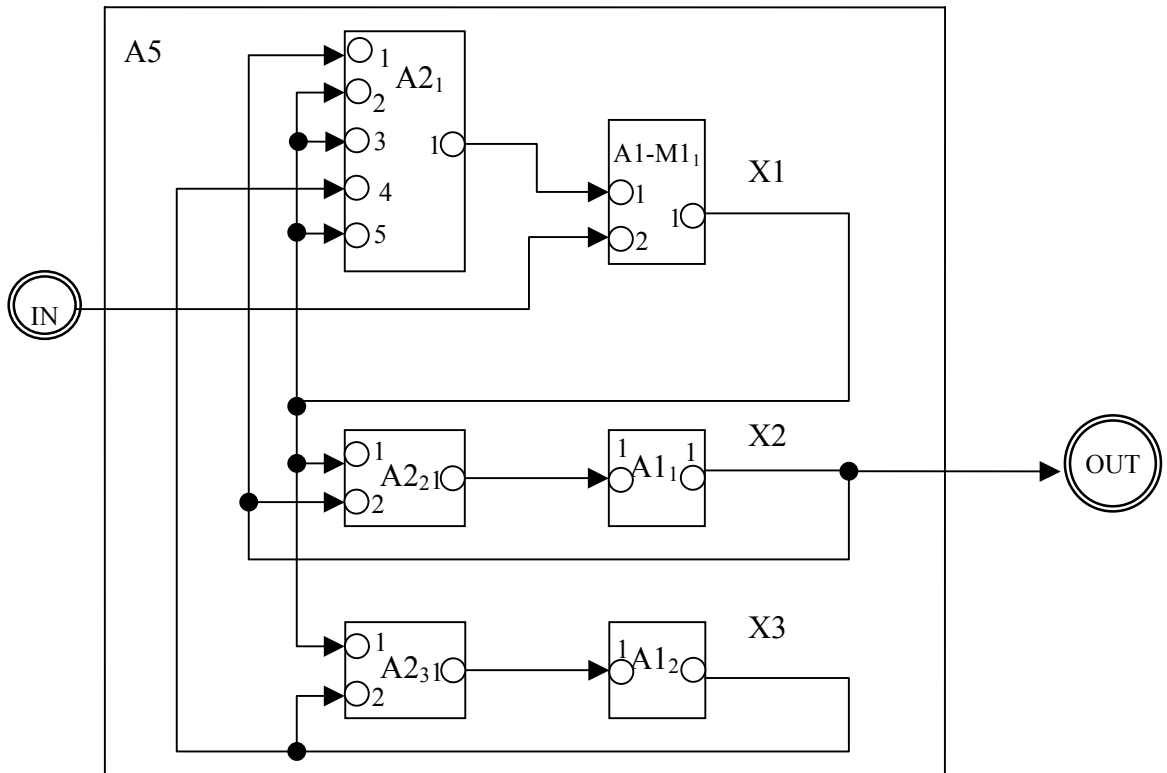
14 pav. Vaistų suleidimo paskirstymas laiko tarpe t_i



1 pav. a) vaistų pompos modelio diagram, b) vaistų pompos veikimo iliustracija

6.5 Farmakokinetinis modelis

Šiam modeliui aprašyti buvo panaudoti standartinis modelis A1 ir pakeistas QSS modelis A1-M1, panaudotas aprašyti 1-sios sekcijoje atliekamam vaistų suleidimui (12 pav.). Vaistų suleidimas buvo atliekamas naudojant atskirą sistemą, prijungtą per IN pažymėtą jungtį, kuri perduoda išorės signalus, reiškiančius vaistų suleidimą. Vaistų suleidimas buvo reguliuojamas priklausomai nuo vaistų koncentracijos 2-joje sekcijoje. Todėl X_2 reikšmė yra perduodama į išėjimo jungtį OUT , kuri yra sujunta su paciento poreikio modeliu (12 pav.).



1 pav. Farmakokinetinio modelio su trejomis sekcijomis diagrama

6.6 Modelio imitavimo rezultatai

Imituojant vaistų koncentracijos kitimą, buvo pasirinkti tokie farmakokinetiniai parametrai [9]:

$$k_{12}=0.385, k_{13}=0.233, k_{21}=0.228, k_{31}=0.021, k_{el}=0.158,$$

$V=13l$ – pirmosios sekcijos tūris, vaistų dozė - 1 mg. Vaistų suleidimas paskirstytas į 10 injekcijų per 1 min. Kiekviena injekcija vaistų koncentraciją 1 sekcijoje padidina:

$$\Delta C = \frac{1 \text{ mg}}{13 \text{ l} \cdot 10} = 7.69 \text{ } \mu\text{g/ml} \quad (11)$$

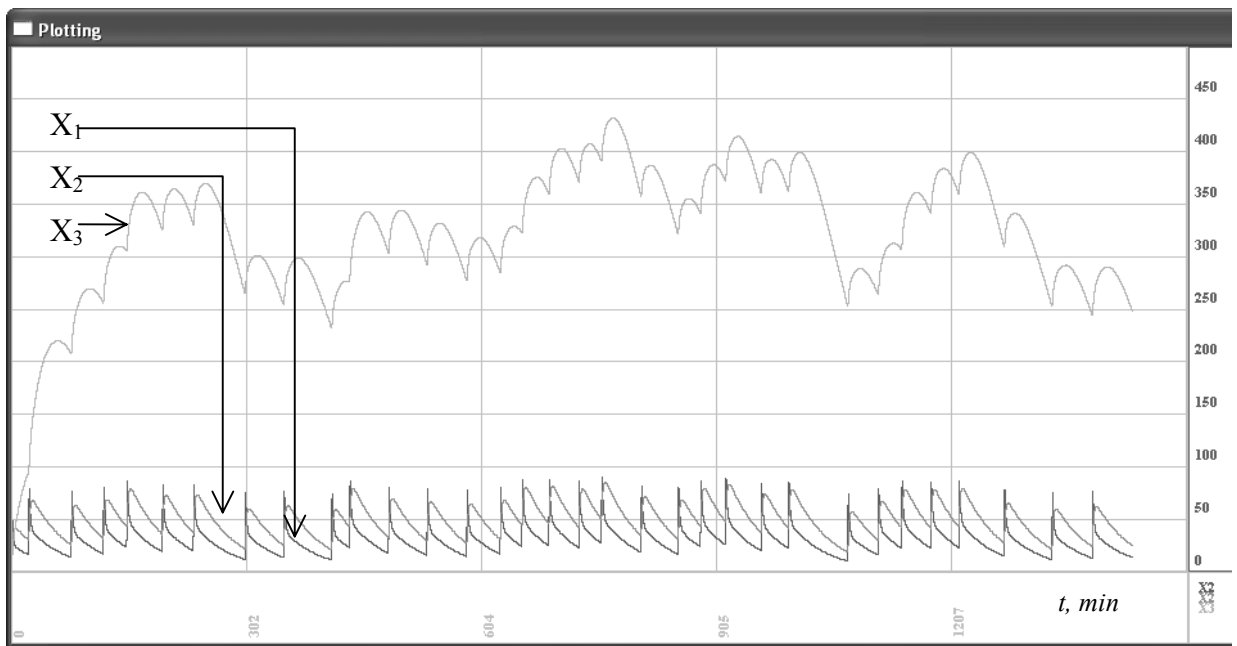
Pompos užrakinimo laikas – 10 min.

Terapeutinio lygio generatoriaus parametrai:

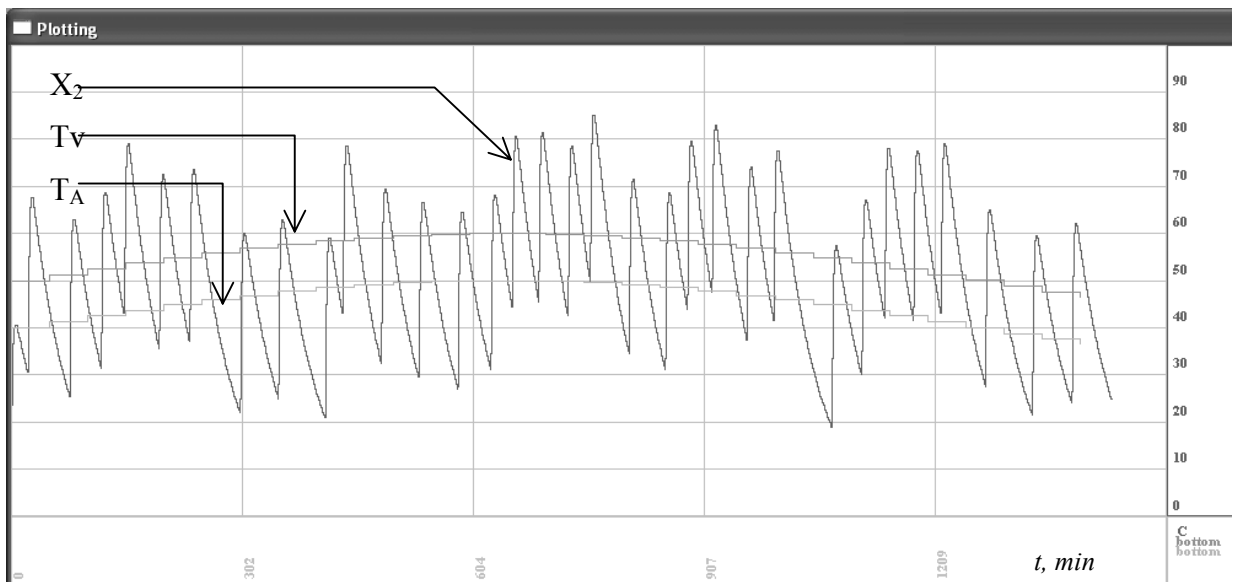
$$\tilde{A} \text{mp} - 10 \text{ (ng/ml)}, \tilde{v} - 0.0004, \tilde{V} - 50 \text{ (ng)}, \tilde{A} - 40 \text{ (ng)}.$$

Paciento vaistų poreikio parametrai:

$$ex = 5.$$



2 pav. Vaistų koncentracijos kitimas 24 val. laikotarpyje, ng/ml



3 pav. Vaistų koncentracijos kitimas 2 sekcijoje ir terapeutiniai lygiai per 24 val. ng/ml

Iš 17 pav. matome, kad vaistų koncentracija smarkiai svyruoja tarp terapeutinių lygių. Galima daryti išvadą, kad laikas, kada vaistų koncentracija yra terapeutinio lygio ribose, yra palyginti mažas, ir pacientas dažniausiai jaučia vaistų stygių arba jų perteklių.

7 Išvados

Apibendrinant darbo rezultatus, galima teigti, kad agregatinis metodas yra tinkamas modeliuoti hibridinėms sistemoms. Pritaikant DEVS modelį panaudojimui agregatiniame metode, aiškiai matome, kad galima nesunkiai DEVS modelį perrašyti naudojant PLA formalizmą. Tuo galima pasinaudoti pritaikant DEVS formalizmui sukurtus modelius ir imitavimo įrankius PLA agregatams aprašyti ir imituoti.

Farmakokinetinio modelio sudarymui agregatiniame metode buvo pritaikytas DEVS kvantuotų būsenų sistemos modelis. Vykdamas farmakokinetinio modelio imitavimą, kai pacientas pats reguliuoja vaistų injekcijas, nustatyta kaip kinta vaistų koncentracija paciento kūne. Rezultatai rodo, kad vaistų koncentracija smarkiai svyruoja virš ir žemiau terapeutinio lygio. Galima daryti išvada, kad vaistus leidžiant dozėmis, yra sunku išlaikyti pastovią vaistų koncentraciją pageidaujamoje kūno vietoje.

Tolesniam darbui rekomenduojama ištirti kvantuotų būsenų sistemos metodo efektyvumą ir skaičiavimo paklaidas. Šiame darbe buvo nagrinėtas QSS modelis. Šiuo metu jau yra suformuluoti QSS2 ir QSS3 modeliai. Jie pasižymi efektyvesniu veikimu, ir gali sumažinti skaičiavimų skaičių net keletą kartų, lyginant su QSS modeliu. Rekomenduotina aprašyti šiuos modelius naudojant PLA formalizmą.

8 Naudota literatūra

- [1] Henrikas Pranevičius, *Kompiuterinių tinklų protokolų formalus specifkavimas ir analizė: agregatinis metodas*. Kauno technologijos universitetas, 2005.
- [2] Gerard J. Holzmann, *The Model Checker SPIN. Transactions on software engineering* vol. 23, No. 5, May 1997.
- [3] J. Bengtsson, K. Larsen, F. Larsson, P. Pettersson, Wang Yi. *UPPAAL - a Tool Suite for Automatic Verification of Real-Time Systems*. Hybrid Control System Symposium, 1996, LNCS, Springer-Verlag.
- [4] Kofman, E., Lapadula, M. and Pagliero, E. *PowerDEVS: A DEVS-Based Environment for Hybrid System Modeling and Simulation*. Technical Report LSD0306, LSD, Universidad Nacional de Rosario, Argentina, 2003.
- [5] E. Kofman. *Discrete Event Based Simulation and Control of Continuous Systems: daktaro disertacija*, Universidad Nacional de Rosario, Argentina, 2003.
- [6] E. Kofman, *A third order discrete event method for continuous system simulation*. FCEIA – UNR – CONICET. Riobamba 245 bis, Rosario, Argentina 2000.
- [7] Tasi'c, Bratislav, *Numerical Methods for Solving ODE Flow by Bratislav Tasi'c*. Eindhoven: Technische Universiteit Eindhoven, 2004. Proefschrift. - ISBN 90.386.0952.3.
- [8] R.R.H. Schiffelers and D.A. van Beek and K.L. Man and M.A. Reniers and J.E. Rooda, *A Hybrid Language for Modeling, Simulation and Verification*. IFAC Conference on Analysis and Design of Hybrid Systems, Saint-Malo Brittany 2003.
- [9] H.F. Hill, A.M. Mackie, B.A. Coda, K. Iverson, C.R. Chapman. *Patient-controlled analgesic administration. A comparison of steady-state morphine infusions with bolus doses*. Cancer, John Wiley & Sons, Inc., 1991, 67, 873–882.
- [10] Ginsberg B.; Gil K.M.; Muir M.; Sullivan F.; Williams D.A.; Glass P.S.A, *The influence of lockout intervals and drug selection on patient-controlled analgesia following gynecological surgery*. Pain, Volume 62, Number 1, July 1995, pp. 95-100(6).
- [11] Mies, R J : van der Aa, J J : Dixon, C L : Kaltenbach, M : Derendorf, H : Gravenstein, N, *Computer-modulated patient-controlled analgesia. Preliminary evaluation of a prototype*. Reg-Anesth. 1994 Jul-Aug; 19(4): 270-6.

9 Priedai

9.1 Standartinis QSS modelis, aprašytas naudojant DEVS formalizmą

$$M_{QSS} = (X, Y, S, \delta_{\text{int}}, \delta_{\text{ext}}, \lambda, ta)$$

$$X = Y = R \times N$$

$$S = R^2 \times Z \times R^+$$

$$\delta_{\text{int}}(s) = \delta_{\text{int}}(x, d_x, j, \sigma) = (x + \sigma \cdot d_x, d_x, j + \text{sgn}(d_x), \sigma_1)$$

$$\delta_{\text{ext}}(s, e, x_u) = \delta_{\text{ext}}(x, d_x, j, \sigma, e, x_v, p) = (x + e \cdot d_x, x_v, j, \sigma_2)$$

$$\lambda(s) = \lambda(x, d_x, j, \sigma) = (Q_{j+\text{sgn}(d_x)}, 1)$$

$$ta(s) = ta(x, d_x, j, \sigma) = \sigma$$

$$\sigma_1 = \begin{cases} \frac{Q_{j+2} - (x + \sigma \cdot d_x)}{d_v} & \text{jei } d_x > 0 \\ \frac{(x + \sigma \cdot d_x) - (Q_{j-1} - \varepsilon)}{|d_x|} & \text{jei } d_x < 0 \\ \infty & \text{jei } d_x = 0 \end{cases}$$

$$\sigma_2 = \begin{cases} \frac{Q_{j+1} - (x + e \cdot x_v)}{x_v} & \text{jei } x_v > 0 \\ \frac{(x + e \cdot x_v) - (Q_j - \varepsilon)}{|x_v|} & \text{jei } x_v < 0 \\ \infty & \text{jei } x_v = 0 \end{cases}$$

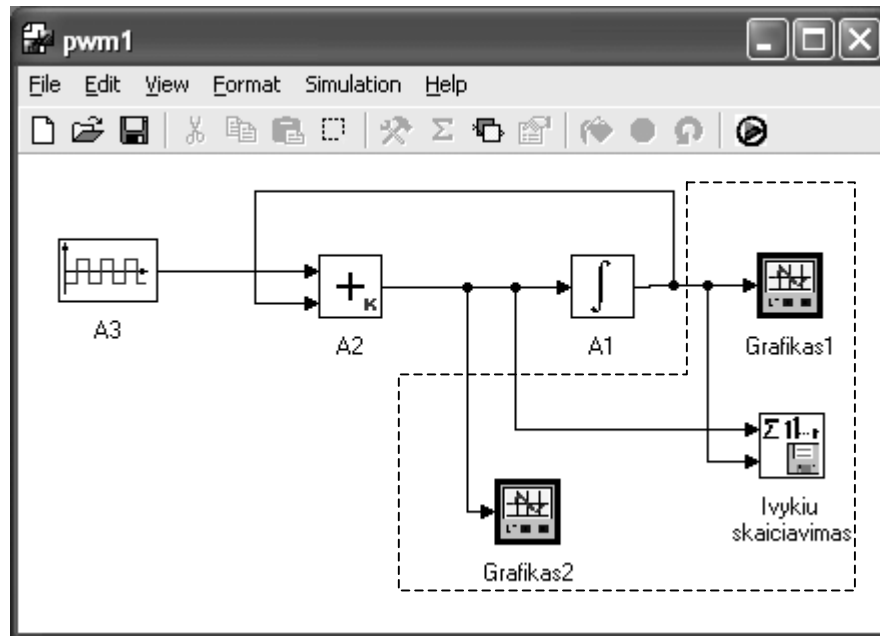
čia:

- X – įėjimo signalas – funkcijos diferencialas, realus skaičius,
- Y – išėjimo signalas – kvantuota funkcijos reikšmė, realus skaičius,
- S – Sistemos būseną, kur $R^2 \times Z \times R^+$:
 - Funkcijos reikšmės, realus skaičius,
 - Funkcijos diferencialas, realus skaičius,

- Funkcijos kvantuotos reikšmės numeris, natūrinis skaičius,
- Laiko tarpas iki sekančio vidinio įvykio realus neneigiamas skaičius.
- δ_{int} – vidinio perėjimo funkcija. Skaičiuoja sekančią kvantuotą funkcijos reikšmę, ir kada jis bus pasiekta, jei nepasirodys išorinis signalas. $j + \text{sgn}(d_x)$ yra sekančios kvantuotos reikšmės numeris. Jei diferencialas buvo teigiamas, numeris didėja, jei neigiamas – numeris mažėja.
- δ_{ext} – išorinio perėjimo funkcija. Atėjus išoriniam signalui apskaičiuoja pasiektą funkcijos reikšmę įvykusio įvykio momentui ir apskaičiuoja sekančio vidinio įvykio laiką su naujai gauta diferencialo reikšme. Išorinio įvykio metu kvantuotos būsenos numeris j nesikeičia.
- $\lambda(s)$ – išėjimo signalo funkcija. Funkcijos reikšmei pasiekus sekančią kvantuotą reikšmę, ši reikšmė perduodama per išėjimo signalą.
- $t_a(s)$ – laiko funkcija, rodanti, kada bus sekantis vidinis įvykis.

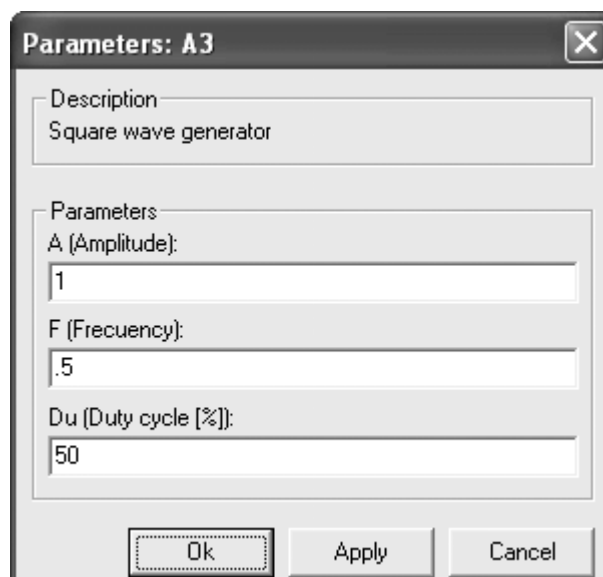
9.2 Elektroninės schemos modelis, sudarytas naudojant imitavimo įrankį PowerDEVS

Modelis sudarytas naudojant standartinius programos elementus.



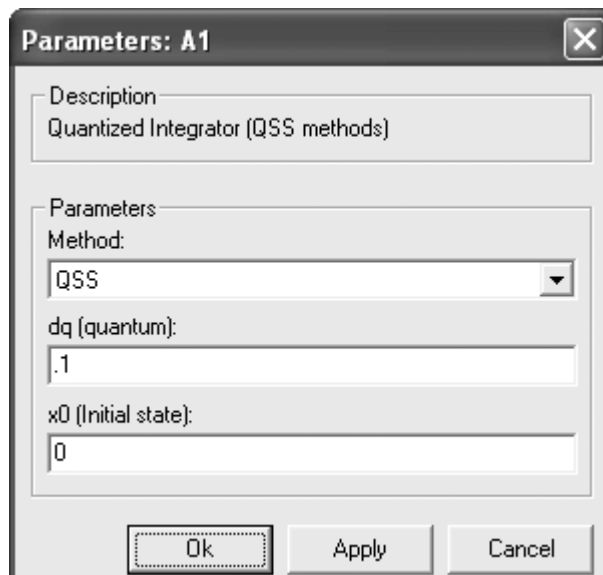
4 pav. DC/AC srovės keitėjo modelis PowerDEVS aplinkoje

Punktyrine linija pažymėti agregatai yra skirti sistemos būsenai stebėti, ir sistemos veikimo neįtakoja. Agregatas A3 – *square wave generator* – signalų generatorius, kurio parametrai pateikti sekančioje iliustracijoje.



5 pav. Signalų generatoriaus parametrai

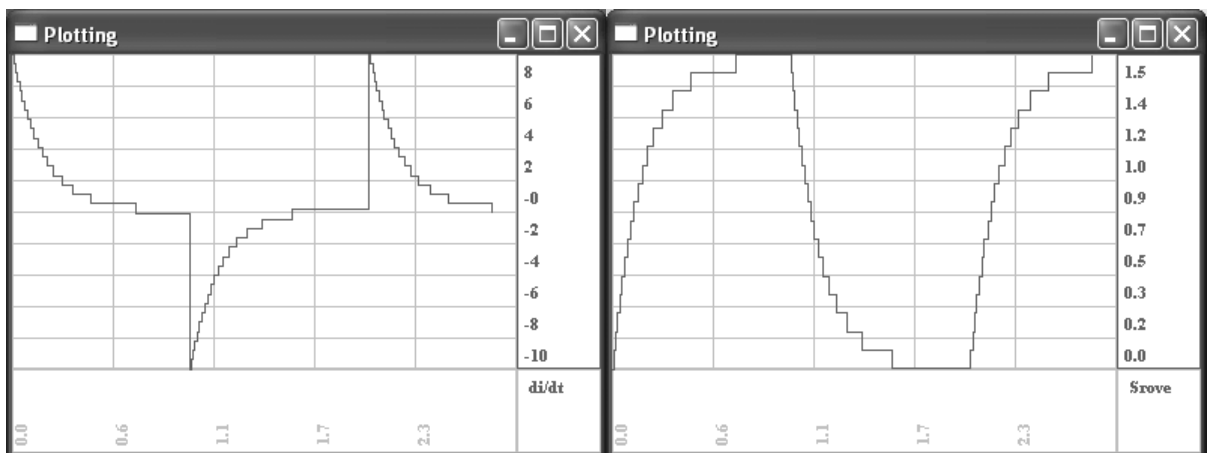
Funkcijos reikšmės skaičiavimui panaudotas standartinis QSS modelis:



6 pav. Integratoriaus parametrai

Funkcijos reikšmių diskretizavimo tinklelio smulkumas – 0,1. Skaičiavimo žingsnių skaičius, atlikus imitavimą 2.99 laiko vienetams:

- 54 „sudėtuvo“ skaičiavimai,
- 52 „integratoriaus“ išėjimo signalai.



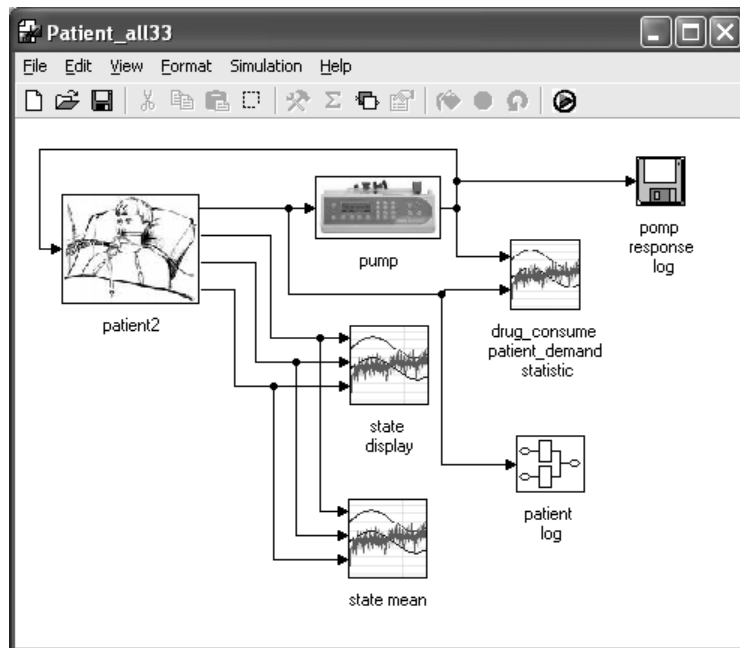
a

b

6 pav. Sistemos imitavimo rezultatų grafikai. a) funkcijos diferencialas, b) funkcijos reišmė

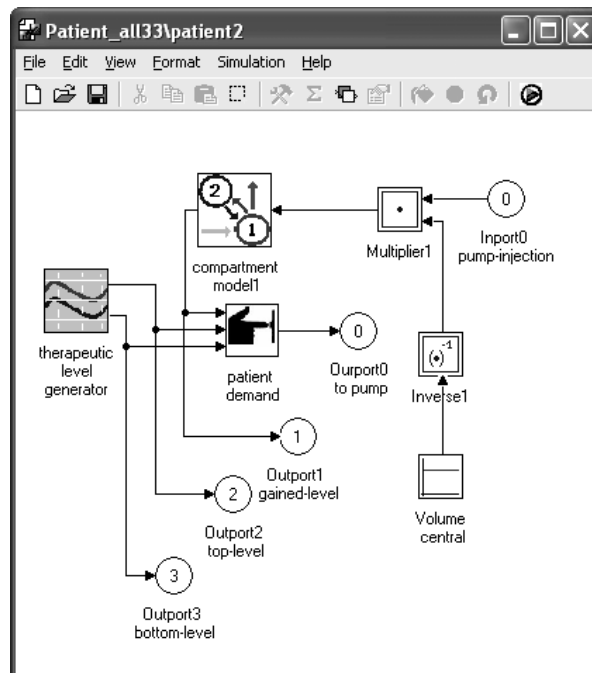
Rezultatų grafikuose galima matyti, kaip vyksta pereinamasis procesas, pasikeitus jungiklių padėčiai grandinėje (6 pav.).

9.3 Farmakokinetinis modelis su PCA, sudarytas naudojant imitavimo įrankį PowerDEVS

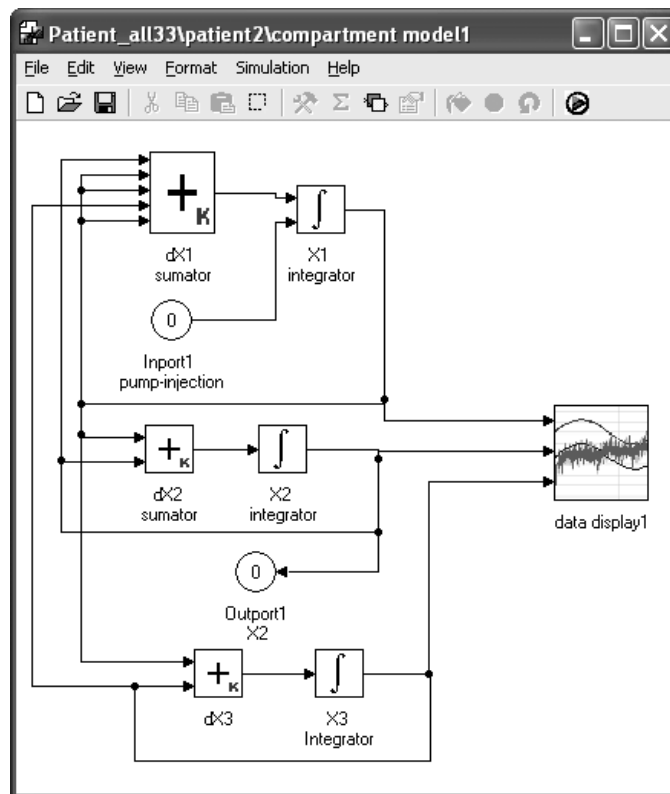


7 pav. Farmakokinetinis modelis su PCA PowerDEVS aplinkoje

22 pav. matome paciento virtualų modelį *patient2*, kurio išėjimas yra sujungtas su vaistų pompa *pump*. Vaistų pompa prijunta prie paciento. Toliau pateiksime paciento modelio detalesnį vaizdą (23 pav.).



8 pav. Paciento modelis PowerDEVS aplinkoje



9 pav. Farmakokinetinis modelis *PowerDEVS* aplinkoje

Diagramose matomi elementai *Data display* yra agregatai yra skirti sistemos būsenai išvesti grafikuose.

Patvirtinu, kad straipsnis „*The use of piece-linear aggregates for specification and simulation of hybrid systems: pharmacokinetics model and patient-controlled analgesia*“ buvo priimtas į doktorantų vasaros mokyklos, kurios tema „Formalūs sistemų analizės metodai informatikoje“, leidinį.

Mokyklos vadovas prof. H. Pranevičius

THE USE OF PIECE–LINEAR AGGREGATES FOR SPECIFICATION AND SIMULATION OF HYBRID SYSTEMS: PHARMACOKINETICS MODEL AND PATIENT-CONTROLLED ANALGESIA

L. Simaitis, supervisor Prof. Habil. Dr. H. Pranevičius

Kaunas University of Technology, Department of Business informatics, Studentų g. 56, LT-51368 Kaunas

Abstract

This paper presents the method of using Piece Linear Aggregates formalism (PLA) for specification and simulation of hybrid systems, when continuous components are described in ordinary differential equations (ODEs). PLA is used to create Quantized State System (QSS) model. QSS is integration based method, created for ODE solving. Since QSS method first was introduced for use in Discrete Event Systems (DEVS) formalism, we adapted it for PLA formalism, which is fundamentally consanguineous to DEVS formalism. Pharmacokinetics model is used as example of hybrid system. Pharmacokinetics is a branch of pharmacology which describes administered drug absorption and distribution in human body. In pharmacokinetics drug distribution is described using ODE. In order to use drug injection in pharmacokinetics model, we have to modify standard QSS model and add some specific capabilities. **Result:** modified QSS method was obtained and used for pharmacokinetic model specification. Simulation of morphine concentration in plasma was performed using real pharmacokinetics data to test Patient-Controlled Analgesia (PCA) method effectiveness.

Key words: piece linear aggregates, hybrid system simulation, discrete event systems, quantized state system, pharmacokinetics, patient-controlled analgesia.

1. Introduction

Computer aided simulations becomes more and more important in many study fields, from theoretical system characteristics research to practical engineering problem solving. There are two genuinely different ways to characterize and simulate continuous or hybrid systems. The most common is based on time discretization, the other one – by event discretization. The first method is well described in literature and has many use examples. But there are many disadvantages of using time discretization, especially in hybrid system simulations. Besides of high computation cost, which is needed to obtain high accuracy, the use of time discretization is inconvenient, when it is important to observe and react to system's state changes. There are two major formalisms dealing with discrete event systems – Piece Linear Aggregates formalism, and Discrete Event System specification formalism.

We need a method, which solves ODE and allows change function value or function differential instantaneously. To obtain such method we shall adapt Quantized State System, described in DEVS, for use in PLA formalism. Originally QSS method was created for solving ODE for continuous systems simulation. There is already technique of using QSS to solve ODE with instant differential changes, so we only need to add function value change functionality.

Previously described method is needed to create pharmacokinetic model. We need means to solve the ordinary differential equations system, which would allow us to simulate morphine concentration in plasma. Virtual patient behavior and drug injection pump models will be used to imitate the regulation of morphine concentration. This will allow us to calculate time-span, when morphine concentration was in desirable therapeutic level, and when wasn't. Obtained results will be used to evaluate and improve PCA.

2. Discrete event systems

2.1 Piece Linear Aggregates formalism

Piece Linear Aggregates is formalism for a discrete event system specification language. The systems are represented as a set of interacting piece-linear aggregates. The aggregate object is defined by:

- Z – set of state values;
- X – set of input signals;
- Y – set of output signals;

- H – transition function;
- G – output generator function;
- E – set of discrete events, where $E = \{E', E''\}$
 - E' - set of external events;
 - E'' - set of internal events;

The aggregate operates in time set $t \in T$. The structure of a state is:

$$z(t) = (v(t), z_v(t)) \in Z \text{ where:}$$

$v(t)$ – discrete part of a state;

$z_v(t)$ – continuous part of a state. It denotes the next time of internal event in the current state.

Events occur when system arrives to a determined time point when the next internal event is scheduled or when input signal arrives. Aggregate may commit output signal when internal or external event occur.

2.2 Discrete event systems specification formalism

Discrete Event System formalism is consanguineous to the PLA formalism. DEVS defines a systems whose input/output behavior is characterized by a sequence of events. DEVS model process input signal and depending on initial states generates output signals. An atomic DEVS model can be defined by:

$M = (X, Y, S, \delta_{int}, \delta_{ext}, \lambda, ta)$, where:

- X – set of input signals;
- Y – set of output signals;
- S – set of state values;
- δ_{int} – set of internal events;
- δ_{ext} – set of external events;
- λ - output function;
- ta – time advance function;

Each possible state s ($s \in S$) has an associated Time Advance calculated by the Time Advance Function $ta(s)$ ($ta(s) : S \rightarrow \mathbb{R}_0^+$). The Time Advance is a non-negative real number saying how long the system remains in a given state in absence of input events. The new state is calculated as $s_2 = \delta_{int}(s_1)$. When the state goes from s_1 to s_2 an output event is produced with value $y_1 = \lambda(s_1)$. When an input event arrives the state changes instantaneously. The new state value depends not only on the input event value but also on the previous state value and the elapsed time since the last transition. No output signal is produced in external event [1].

3 Hybrid systems specification and simulation

To solve ODEs system we'll adopt and slightly modify Quantized State System method, which was defined by Ernesto Kofman [2]. QSS method computes function's value with a given differential at the fixed quantum grid. The next quantization level time-point is calculated based on given differential value with integrator object, marked as \int (Fig. 1a). Integrator and quantization function are defined in one aggregate, further Quantized State Integrator (QSI) aggregate. Objects, named f_i , are aggregates, calculating functions differentials from given functions values and input values vectors. The all compound of summation and QSI aggregates forms Quantized State System (Fig. 1). Below is given QSI model specification in DEVS formalism:

$M_1 = (X, Y, S, \delta_{\text{int}}, \delta_{\text{ext}}, \lambda, ta)$, where

$X = \mathfrak{R} \times \{\text{import}\}$

$Y = \mathfrak{R} \times \{\text{outport}\}$

$S = \mathfrak{R}^2 \times Z \times \mathfrak{R}_0^+ \infty$

$\delta_{\text{int}}(s) = \delta_{\text{int}}(x, d_x, k, \sigma) = (x + \sigma \cdot d_x, d_x, k + \text{sgn}(d_x), \sigma_1)$

$\delta_{\text{ext}}(s, e, x_u) = \delta_{\text{ext}}(x, d_x, k, \sigma, e, x_v, \text{port}) = (x + e \cdot d_x, x_v, k, \sigma_2)$

$\lambda(s) = \lambda(x, d_x, k, \sigma) = (Q_{k+\text{sgn}(d_x)}, \text{outport})$

$ta(s) = ta(x, d_x, k, \sigma) = \sigma$

where

$$\sigma_1 = \begin{cases} \frac{Q_{k+2} - (x + \sigma \cdot d_x)}{d_x} & \text{if } d_x > 0 \\ \frac{(x + \sigma \cdot d_x) - (Q_{k+1} - \varepsilon)}{|d_x|} & \text{if } d_x < 0 \\ \infty & \text{if } d_x = 0 \end{cases} \quad \sigma_2 = \begin{cases} \frac{Q_{k+1} - (x + e \cdot x_v)}{d_x} & \text{if } x_v > 0 \\ \frac{(x + e \cdot x_v) - (Q_k - \varepsilon)}{|x_v|} & \text{if } x_v < 0 \\ \infty & \text{if } x_v = 0 \end{cases}$$

Given QSS model is enough to simulate a system of ODEs. It can react to function differential values changes, depending on other function values and input signals - $u(t)$ - and accordingly calculate function value trajectory.

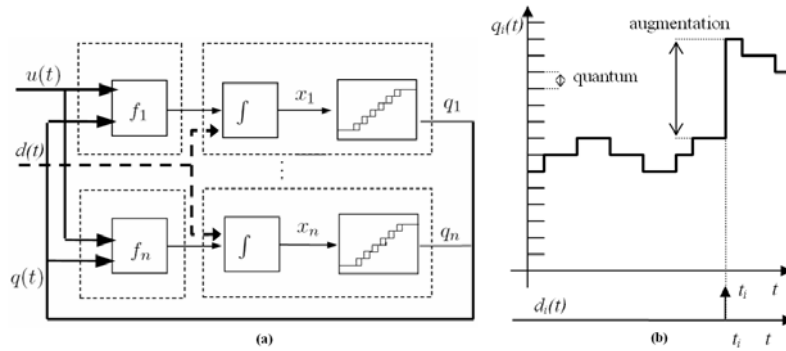


Fig. 1: a) Block diagram of QSS. Segmented line shows added augmentation input; b) Function augmentation;

In order to utilize QSS method for drug injection simulation, we need add function augmentation characteristic. To do that, we add new input vector $d(t)$ (Fig. 1a), which carries function augmentation value. When augmentation of function occurs, function value changes accordingly:

$$x_i(t) = x_i(t^-) + d_i(t) \quad (1)$$

The rest of the QSS model is left unchanged. Below is given modified QSS model specification in PLA formalization language:

1. Set of input signals $X = \{x_v(t_m), y_r\}$ where: $x_v(t_m) \in R$ - function differential;
 $y_r \in R$ - function augmentation value;
2. Set of output signals $Y = Q_j(t_m), j = 1K r$ - quantized function value;
3. Set of external events $E' = \{e'_1, e'_2\}$; where: e'_1 - new differential value;
 e'_2 - function augmentation value;

4. Set of internal events $E'' = \{e_1''\}$ where: e_1'' - function reached next quantum step;
5. Discreet component of state $v(t_m) = \{x(t_m), d_x(t_m), j(t_m), \varepsilon, \Delta Q\}$ where:
 $x(t_m) \in R$ - function value, $d_x(t_m) \in R$ - current differential, $j(t_m) \in Z$ - number of quantized function value, ε - hysteresis window, Q_1, Q_2, \dots, Q_r ; $Q_k - Q_{k-1} = \Delta Q$ - quantum;
6. Continuous part of state $z_v(t_m) = \{w(e_1'', t_m)\}$ - next internal event time point;
where: $w(e_2'', t_m) = \begin{cases} < \infty, d_x(t_m) \neq 0 \\ \infty \text{ otherwise} \end{cases}$
7. Controlling sequences $e_1' \propto \{\sigma_1\}$, $e_1'' \propto \{\sigma_2\}$
8. Initial state $v(t_0) = \{x(t_0), d_x(t_0), j(f(x(t_0)))\}$
 $z_v(t_0) = \{t_0 + \sigma_2\}$

9. Transfer operators:

$$\begin{aligned}
H(e_1'): & \quad \text{(event - new differential value arrived)} \\
x(t_{m+1}) &= x(t_m) + e \cdot x_v \quad \text{(new function value calculated at the point of event using old differential)} \\
d_x(t_{m+1}) &= x_v \quad \text{(new differential assigned)} \\
j(t_{m+1}) &= j(t_m) \quad \text{(new quantized value number)} \\
w(e_1', t_{m+1}) &= t_m + \sigma_1, \quad \text{when} \quad \text{(next internal event point)} \\
\sigma_1(t_{m+1}) &= \begin{cases} \frac{Q_{j(t_m)+1} - (x(t_m) + e(t_m) \cdot x_v)}{x_v} & \text{if } x_v > 0 \\ \frac{(x(t_m) + e(t_m) \cdot x_v(t_m)) - (Q_{j(t_m)-1} - \varepsilon)}{|x_v|} & \text{if } x_v < 0 \\ \infty & \text{if } x_v = 0 \end{cases}
\end{aligned}$$

where: $e(t_m)$ - elapsed time after previous event;

$$\begin{aligned}
H(e_2'): & \quad \text{(event - function augmentation - added modification)} \\
x(t_{m+1}) &= x(t_m) + e \cdot d_x + x_y \quad \text{(function augmentation)} \\
d_x(t_{m+1}) &= d_x(t_m) \\
j(t_{m+1}) &= j(t_m) + \lfloor (e \cdot d_x + x_y) / \Delta Q \rfloor \quad \text{(new quantized value number)} \\
w(e_2', t_{m+1}) &= t_m + \sigma_2 \\
G(e_2'): & \quad y = Q_{j(t_m) + \lfloor x_y / \Delta Q \rfloor} \quad \text{(calculated output value)} \\
H(e_1''): & \quad \text{(new function step reached)} \\
x(t_{m+1}) &= x(t_m) + \sigma \cdot d_x(t_m) \quad \text{(new function value calculated)} \\
d_x(t_{m+1}) &= d_x(t_m) \\
j(t_{m+1}) &= j(t_m) + \text{sgn}(d_x(t_m))
\end{aligned}$$

$$w(e_1^n, t_{m+1}) = t_m + \sigma_2, \quad \text{when}$$

$$\sigma_2(t_{m+1}) = \begin{cases} \frac{Q_{j(t_m)+2} - (x(t_m) + \sigma(t_m) \cdot d_x(t_m))}{d_x(t_m)} & \text{if } d_x(t_m) > 0 \\ \frac{(x(t_m) + \sigma(t_m) \cdot d_x(t_m)) - (Q_{j(t_m)-1} - \varepsilon)}{|d_x(t_m)|} & \text{if } d_x(t_m) < 0 \\ \infty & \text{if } d_x(t_m) = 0 \end{cases}$$

$$G(e_1^n): y = Q_{j(t_m) + \text{sgn}(d_x(t_m))}$$

4. Pharmacokinetics model specification and simulation

When drug enters the body, numerous processes begin to work on the drug. Pharmacokinetics describes how absorption, distribution, metabolism and elimination influences drug in human body. Different body parts and tissues are represented as sections. The drug distribution between sections is marked as arrows with rate constants [3].

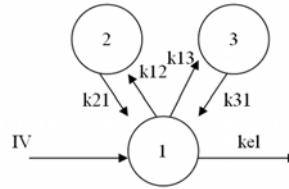


Fig. 2: A three compartment pharmacokinetics model; IV – drug intravenous injection, kel – drug elimination constant; 1-compartment is called central compartment;

The rate constants shows how quickly drug move form one compartment to another. The drug concentration change is described in first order process:

$$\begin{cases} \dot{X}_1 = k_{21} \cdot X_2 - k_{12} \cdot X_1 + X_3 \cdot k_{31} - X_1 \cdot k_{13} - kel \cdot X_1 \\ \dot{X}_2 = k_{12} \cdot X_1 - k_{21} \cdot X_2 \\ \dot{X}_3 = k_{13} \cdot X_1 - k_{31} \cdot X_3 \end{cases} \quad (2)$$

The values X_1 , X_2 and X_3 are drug concentration in blood plasma. The given ODEs system can be easily modeled using QSS model (Fig 2a). Additionally, using function augmentation feature (1), drug injection can be implemented. The actual increment value is calculated depending on bolus dose and compartment volume. Drug injection is executed by drug pump model. Drug demand is executed using virtual patient model with exponential impatience function (3):

$$wait = \begin{cases} \frac{e^{2C/C_B}}{K} + \xi & \text{if } C < C_B \\ \frac{e^{((C-C_B)/(C_T-C_B)+2)}}{K} + \xi & \text{otherwise} \end{cases} \quad (3)$$

Where: C – drug concentration in observed compartment; C_B – bottom of therapeutic level of drug concentration; C_T – top of therapeutic level of drug concentration; K – adjustment constant. Virtual patient periodically sends signals to the pump, which are based on current concentration level. It has to be noted, that therapeutics level is not constant and during the day changes. ξ - normal distribution random value from 0 to 80, mean=40.

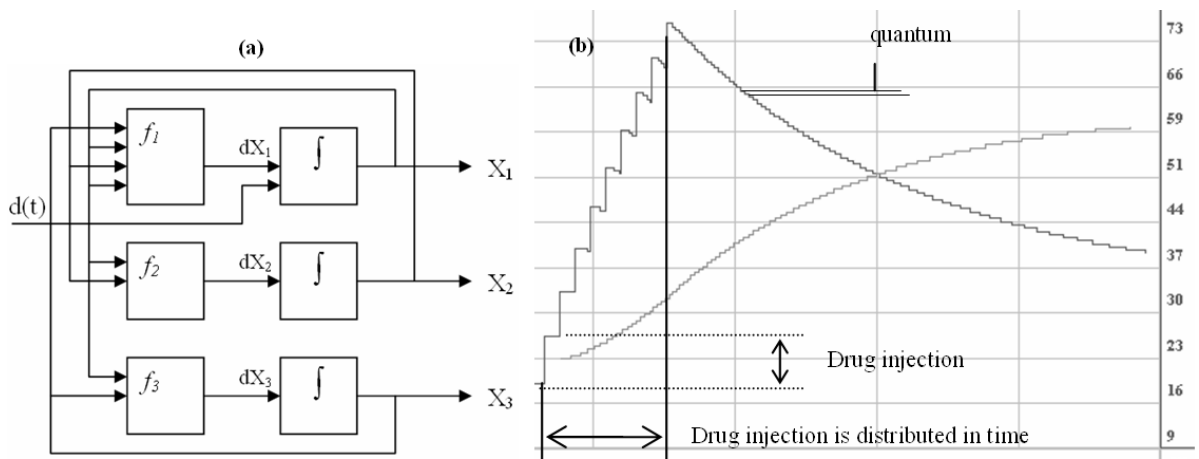


Fig. 3: a) QSS block diagram for the given pharmacokinetics model; b) Distributed drug injection;

For model simulation used pharmacokinetics parameters: $kel=0.158$, $k12=0.385$, $k13=0.233$, $k21=0.228$, $k31=0.021$. Central section volume $V=13l$, bolus dose – 1 mg. Concentration increment on injection, when injection is distributed to 10 parts $\Delta C = 1mg \div 13l \div 10 = 7.69 \mu g/ml$. Pump lockout – 10 min. simulation precision – $\Delta Q=0.1$. The simulation is performed using PowerDEVS 1.0 environment. Time scale showed in minutes.

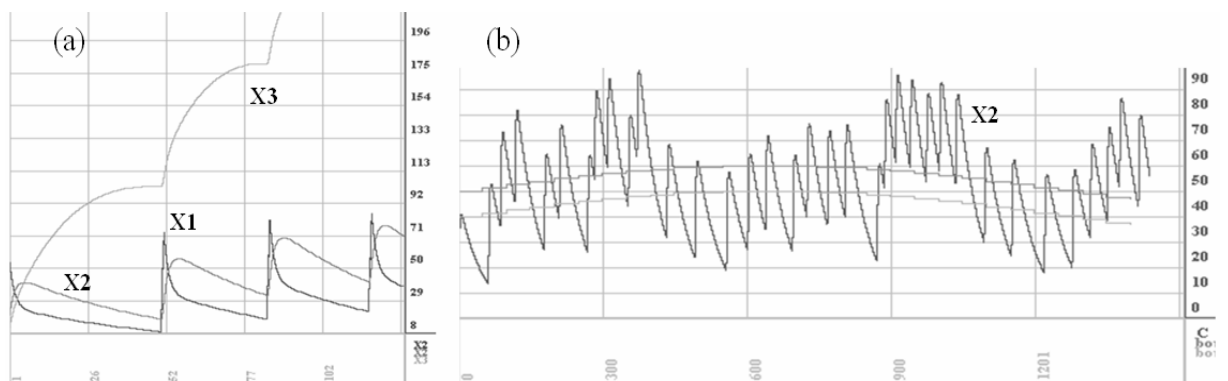


Fig. 4: a) drug concentrations in all apartments; b) Drug concentration in 2-nd compartment corresponding to therapeutic level;

5. Conclusions

QSS method was proved to be convenient for pharmacokinetics model simulation. Slightly modified Quantized State System was easily implemented in PowerDEVS environment. Simulation results shows, that it is very hard to maintain steady drug concentration in therapeutic level using intravenous bolus injections (Fig 4b). During 24 hours period, high concentration's overdrafts detected – up to 100% from therapeutic level. It is possible, that patient demand is inaccurate, so further virtual patient model development needed. In order to get better simulation efficiency, implementation of QSS2 and QSS3 methods is recommended.

References

- [1] E. Kofman. Discreet Event Simulation of Hybrid Systems. SIAM Journal on Scientific Computing, 2002.
- [2] E. Kofman, M. Lapadula, E. Pagliero. PowerDEVS: A DEVS-Based Environment for Hybrid System Modeling and Simulation. Technical Report LSD0306, LSD, Universidad Nacional de Rosario, 2003.
- [3] H.F. Hill, A.M. Mackie, B.A. Coda, K. Iverson, C.R. Chapman. Patient-controlled analgesic administration. A comparison of steady-state morphine infusions with bolus doses. Cancer, John Wiley & Sons, Inc., 1991, 67, 873–882.