**1. Моделирање и модели**

Моделирање представља један од основних процеса људскога ума. Оно је уско везано за начин људског размишљања и решавања проблема.

Као резултат процеса који називамо интелигентно људско понашање, моделирање представља свакодневну активност и велики део онога што нас чини људским (интелигентним) бићима.

Моделирање изражава нашу способност да: мислимо и замишљамо, користимо симболе и језике, комуницирамо, вршимо генерализације на основу искуства, се суочавамо са неочекиваним.

Oмогућава нам да: уочавамо обрасце, процењујемо и предвиђамо, управљамо процесима и објектима, излажемо значење и сврху.

У најширем смислу, моделирање представља исплативо (у смислу трошкова) коришћење нечега (модел) уместо нечега другог (реални систем) са циљем да се дође до одређеног сазнања. Резултат моделирања је модел.

МОДЕЛ је апстракција реалности у смислу да он не може да обухвати све њене аспекте.

МОДЕЛ је упрошћена и идеализована слика реалности.

Модел нам омогућава да се суочимо са реалним светом: на поједностављен начин, избегавајући његову комплексност и иреверзибилност, као и све опасности које могу проистећи из експеримента над самим реалним системом.

Mодел не садржи само објекте и атрибуте реалног система, већ и одређене претпоставке о условима његове валидности.

Циљ модела НИЈЕ: да прецизно репродукује стварност у свој њеној сложености, ЈЕСТЕ: да уобличи на видљив, често формалан начин, оно што је суштинско за разумевање неког аспекта његове структуре или понашања.

Апстракција

Модели су увек апстракције реалног система, због тога задржавају само оне карактеристике оригинала које су битне за сврху његовог изучавања.

Ниво апстракције у процесу моделирања утиче на валидност модела, односно на успешност представљања реалног система моделом

Сувише сложени или савршени модели који имају способност да за исти скуп улазних величина производе исте излазне вредности као и реални системи, чак иако су оствариви, по правилу су прескупи и неадекватни за експериментисање.

С друге стране, сувише поједностављени модели не одсликавају на прави начин посматрани систем, а резултати који се добијају њиховом применом могу да буду неадекватни и погрешни.

Опредељујући се за ниво апстракције неопходно је у одређеном тренутку повући границу у реалном систему и то тако да модел што верније одсликава посматрани систем, али и да, с друге стране, његова сложеност и цена не буду ограничавајући фактори.

Реални систем и модел

Са слике уочавамо да граница која је "повучена" приликом креирања модела не садржи све мерљиве улазе референтног система.

Модел даје излазе који се разликују од излаза реалног система.

Врсте модела

|  |  |
| --- | --- |
| * Мисаони, * Вербални, * Математички, * Структурни, * Физички, * Аналогни, * Симулациони, * Рачунарски, * итд. | * Формални * Неформални   Често их делимо и на:   * + материјалне     - модел хемијске структуре молекула     - модел авиона   + симболичке моделе     - математички,     - концептуални,     - рачунарски,     - симулациони и др. |

Ментални модели

Ментални моделису структуре које људски мозак непрекидно конструише како би био у стању да повеже низ чињеница са којима се човек сусреће, а потом на основу тога делује. Такви модели омогућују, на пример, разумевање физичког света, комуникацију међу људима и планирање акција.

Вербални модели

Директна последица менталних модела и представљају њихов израз у говорном језику, а уобичајено се представљају у писаном облику. Вербални модели спадају у класу неформалнихмодела.

Физички модели

Представљају умањене моделе реалног система, који се понашају на исти начин као и њихови оригинали. Углавном се праве на основу теорије сличности или, у бољем случају, на основу физичких закона сличности.

Математички модели

Уколико су везе између објеката модела описане математичким (нумеричким) релацијама, тада се ради о математичким моделима. Код формулисања математичког модела полази се од вербалног модела који се трансформисањем доводи у стање које се може описати математичким језиком. Одавно је уочена чињеница да се различити физички објекти могу описивати истим математичким моделом. Између два физичка модела који имају исте математичке моделе, каже се да постоји математичка аналогија. Истоветност математичких модела двају објеката пружа могућност да један од физичких објеката буде коришћен за анализу математичког модела другог објекта.

Физички модели

Физички објекат који се користи за анализу математичког модела другог објекта, а са којим има исти или сличан математички модел, назива се аналогни модел. Према томе, између физичког објекта који се испитује и аналогног модела постоји математичка аналогија (аналогија у понашању).

Концептуални модели

Концептуални модели се стварају на основу представе о структури и логици рада система или проблема који се моделира и приказују се у облику чије је значење прецизно дефинисано на пример, дијаграми са тачно дефинисаним симболима. Њихова посебна важност проистиче из чињенице да они представљају основу за израду рачунарских модела. Ови се модели често називају и структурни, пошто у графичком облику указују на структуру посматраног система. Поред тога, графички приказ модела обезбеђује релативно једноставан приказ сложених система и несумњиво значи важан корак ка бољем разумевању ситема који се моделира.

Рачунарски (симулациони) модели

Рачунарски (симулациони) модели су приказ концептуалних модела у облику програма за рачунар. Као средство за изражавање, рачунарски модели користе програмске језике и стога су блиско везани за развој рачунарских наука.

Неформални модели

Неформални опис модела даје основне појмове о моделу и тежи се његовој потпуности и прецизности, што он најчешће није. Приликом изградње неформалног описа врши се подела на: објекте, описне променљиве и правила интеракције објеката

***Неформални опис модела***

ОБЈЕКТИ су делови из којих је модел изграђен.

ОПИСНЕ ПРОМЕНЉИВЕ (преко вредности које узимају) описују стања у којима се објекти налазе у одређеним временским тренуцима.

ПРАВИЛА ИНТЕРАКЦИЈЕ ОБЈЕКАТА дефинишу како објекти модела утичу један на други у циљу промене њиховог стања.

Не постоји правило за избор објеката, описних променљивих и правила интеракције. Најчешће није конзистентан и јасан, нарочито када су у питању сложени модели.

*Аномалије*

НЕКОМПЛЕТАН ОПИС МОДЕЛА - уколико модел не садржи све ситуације које могу да наступе.

НЕКОНЗИСТЕНТАН ОПИС МОДЕЛА - уколико су у опису модела за исту ситуацију предвиђена два или више правила чијом се применом добијају контрадикторне акције

НЕЈАСАН ОПИС МОДЕЛА - ако у једној ситуацији треба обавити две или више акција, а при томе није дефинисан њихов редослед

***Формални опис модела***

Формализам специфицира класу посматраних објеката на недвосмислен и генералан начин, коришћењем конвенција и правила.

ФОРМАЛНИ ОПИС МОДЕЛА треба да обезбеди већу прецизност и потпуност у описивању модела, а понекад омогућава и да се формализује поступак испитивања некомплетности, неконзистентности и нејасности.

УВОЂЕЊЕ ФОРМАЛИЗАМА у методологију моделирања омогућава да сву своју пажњу усмеримо на оне карактеристике објеката које су од највећег значаја за наше истраживање, дакле да користимо апстракције.

Фазе изградње формалног модела - Интеракција између моделара, рачунара и реалног система

Препоруке при изградњи формалног модела

1. Граница система са околином мора бити одабрана тако да систем, односно његов модел, обухвата само феномене од интереса.
2. Модели не смеју бити сувише сложени нити детаљни, већ треба да садрже само релевантне елементе система.
3. Модел не сме сувише да поједностави проблем, нпр. избацивањем важних променљивих потребних за адекватан опис система или сувише великим степеном агрегације компоненти система.
4. Модел је разумно раставити на више добро дефинисаних и једноставних модула с тачно одређеном функцијом, које је лакше и изградити и проверити.
5. У развоју модела препоручује се коришћење неке од проверених метода за развој алгоритама и програма.
6. Потребна је провера логичке и квантитативне исправности модела, и то како појединачних модула тако и целог модела.

**РАЧУНАРСКА СИМУЛАЦИЈА**

Понашање модела може се одредити на практичан начин: АНАЛИТИЧКИ, НУМЕРИЧКИ ИЛИ ПУТЕМ ЕКСПЕРИМЕНТА, где се за извесне, углавном случајне улазе, посматрају одговарајући излази. Овај последњи процес се зове СИМУЛАЦИЈА

У свакодневној употреби може да означи већи број различитих активности, као на пример: сложене видео игре, испитивање утицаја бројних фактора на лет нових модела авиона, део експеримента у социо-психолошким истраживањима итд.

Под симулацијом подразумева се процес изградње апстрактних модела за неке системе или подсистеме реалног света и обављање већег броја експеримената над њима.

Посебно нас интересује случај када се ти експерименти одвијају на рачунару. Тада говоримо о РАЧУНАРСКОМ МОДЕЛИРАЊУ и СИМУЛАЦИЈИ.

**Моделирање и симулација-** У моделирању рачунари се користе у две сврхе: РАЗВОЈ МОДЕЛА и ИЗВОЂЕЊE ПРОРАЧУНА НА ОСНОВУ СТВОРЕНОГ МОДЕЛА.

МОДЕЛИРАЊЕ је процес којим се успоставља веза између реалног система и модела, док је СИМУЛАЦИЈА процес који успоставља релацију између модела и рачунара.

**Релације моделирања и симулација**

Израз моделирање и симулација изражава сложену активност која укључује три елемента:

**Реални систем**

Уређен, међузависан скуп елемената који формирају јединствену целину и делују заједнички како би остварили задати циљ или функцију, без обзира да ли се ради о природном или вештачком систему, и такође, без обзира да ли тај систем у посматраном тренутку постоји или се његово постојање планира у будућности.

**Модел**

Скуп инструкција (програм) који служи да се генерише понашање симулираног система (временска серија вредности променљивих симулираног система).

**Рачунар**

Уређај способан за извршење инструкција модела, које на бази улазних података генеришу развој модела у времену.

**Моделирање**

Моделирање је процес којим се успоставља веза између реалног система и модела, док је симулација процес који успоставља релацију између модела и рачунара.

Релација моделирања односи се на валидност модела.

Валидност или ваљаност модела описује колико верно један модел представља симулирани систем.

**Симулација**

Релација симулације односи се на проверу да ли симулациони програм верно преноси модел на рачунар као и на тачност којом рачунар извршава инструкције модела.

**Процес моделирања и симулација:**

Процесом моделирања се управља на основу циљева који се генеришу ван граница система.

Сваки нови циљ иницира активност синтезе модела.

При синтези модела се користи расположиво знање из базе модела и базе података које чувају и организују прикупљене податке о реалном систему.

Фазе симулације и валидације следе фазу изградње модела.

**Предмет моделирања и симулације**

* Рачунарска симулација има у основи модел система.
* Систем је уређај или процес који постоји или се планира.
* На пример: Производни погон, Пошта, банка или нека слична услужна организација, Дистрибуциона мрежа, Служба за хитне интервенције, Сервисна служба са потенцијалним клијентима

**Историјски преглед развоја симулације:** 1600. Физичко моделирање; 1940. Појава електронских рачунара; 1955. Симулација у авио – индустрији; 1960. Симулација производних процеса; 1970. Симулација великих система укључујући економске, друштвене и еколошке; 1975. Системски приступ у симулацији; 1980.Симулација дискретних стохастичких система и виши ниво учешћа у системима за подршку одлучивању; 1990. Интеграција рачунарске симулације, вештачке интелигенције, мултимедијалних технологија и рачунарских мрежа.

**Потребе за симулацијом**

* Експеримент над реалним системом може да буде скуп или чак немогућ (на пример, у економским системима или хемијским постројењима).
* Аналитички модел нема аналитичко решење (на пример сложенији модели масовног опслуживања)
* Систем може да буде сувише сложен да би се описао аналитички (на пример, системом диференцијалних једначина).
* Симулирање услова под којима наступа разарање система

**Када је могуће експериментисати на систему?**

ПРИМЕР 1:Неки градови имају на улазним саобраћајницама инсталирану светлосну сигнализацију и могуће је експериментисати са сигналним плановима како би се систем подесио, тако да проток саобраћаја буде што већи и безбеднији у време јутарњих или поподневних шпицева.

Када је могуће експериментисати на систему?

ПРИМЕР 2: Менаџер самопослуге може да испроба различите начине управљања набавком и расподелом задатака запосленима како би дошао до комбинације која пружа најбољу услугу и доноси највећи профит.

Када је могуће експериментисати на систему?

ПРИМЕР 3: У рачунарској мрежи могуће је експериментисати са различитим мрежним параметрима и приоритетима за *job* –ове да би се сагледало како они утичу на искоришћеност уређаја и брзину рада.

**Када није могуће експериментисати на систему ?**

ПРИМЕР 1: Немогуће је експериментисати са алтернативним производним програмом фабрике која не постоји. Чак и у случају да фабрика постоји, било би веома скупо прећи на неки експериментални производни програм који можда неће дати добре резултате.

Када није могуће експериментисати на систему ?

ПРИМЕР 2: Тешко је “угурати” у банку или пошту два пута више клијената него што је уобичајено како би се испитало шта се дешава у систему када се број клијената приближи граници функционисања система.

ПРИМЕР 3: Увођење нове неиспитане процедуре за предају пртљага на аеродрому могло би изазвати такве гужве, да велики број путника пропусти своје летове.

**Могућности примене симулације**

Информационе и организационе промене или промене у окружењу могу се симулирати, а уједно се могу посматрати ефекти тих промена на понашање модела.

Знање стечено у процесу изградње модела и симулације може бити од великог значаја код побољшања система који се испитује.

Мењањем симулационих улаза и посматрањем резултујућих излаза, долазимо до важног сазнања о томе које су променљиве система најважније и како променљиве утичу једна на другу.

Симулација се може користити и као педагошко средство са циљем да побољшава методологије аналитичких решења. Симулација се може користити за експериментисање са новим концепцијама или политикама пре него што се изврши њихова имплементација.

**Предности коришћења симулације**

* Вишеструко коришћење истог модела
* Помоћ код анализе и код непотпуних улазних података
* Економска прихватљивост
* Лакша примена
* Мање поједностављујућих претпоставки
* Понекад једна солуција
* Моделирање сложених динамичких система са случајним варијаблама

**Недостаци коришћења симулација**

* Цена
* Време
* Неопходан већи број симулационих експеримената
* Не добијају се зависности излазних варијабли од улазних
* Нема оптималних решења
* Неопходно је познавање више метода и алата
* Сложено вредновање модела

**СИМУЛАЦИОНИ ПРОЦЕСИ**

Структура решавања стварних проблема помоћу симулационог моделирања.

Приказује се у облику низа корака који описују поједине фазе решавања проблема овом методом (животни циклус симулације).

Структура симулационог процеса није строго секвенцијална, већ је могућ и повратак на претходне кораке процеса.

**Симулациони процес**

**Поделе симулационих модела**

Разликујемо два основна начина поделе симулационих модела:

* први, према врсти променљивих у моделу
* Детерминистички и
* Стохастички
* други, према начину на који се стање у моделу мења у времену.
* Дискретни и
* Континуални

**Детерминистики модели**

Модели чије се понашање може предвидети, односно у којима је ново стање система у потпуности одређено претходним стањем.

**Стохастички модели**

Модели чије се понашање не може унапред предвидети, али се могу одредити вероватноће промена стања система.

Карактеристично случајно понашање, односно постојање случајних променљивих у систему.

**Дискретни модели**

|  |  |
| --- | --- |
| Стање система се мења само у појединим тачкама у времену (нема континуалне промене стања).  Такве промене се називају догађаји. |  |

**Континуални модели**

|  |  |
| --- | --- |
| Променљиве стања се мењају континуално у времену.  Пример континуалне промене је лет авиона чији се положај и брзина мењају континуално у времену. |  |

**Мешовити континуално-дискретни модел**

**Врсте симулационих модела**

* Монте Карло (*Monte Carlo*) симулација
* Континуална симулација
* Симулација дискретних догађаја
* Мешовита, континуално-дискретна симулација.

**Монте Карло симулација**

Монте Карло симулацијама називају било коју врсту програма који се користе случајним бројевима. Употребљава се за статичке типове симулације код којих се у решавању проблема користи стварање узорака из расподела случајних променљивих. При томе, проблеми могу бити било детерминистичког, било стохастичког карактера.

**Континуална симулација**

Користи се за динамичке проблеме код којих се променљиве стања мењају континуално у времену. Постоје две основне класе проблема који се решавају овом методом:

1. Релативно једноставни проблеми који су описани детаљно и код којих су промене "глатке" и природно се описују диференцијалним једначинама. Проблеми из физике, биологије и инжењерства.
2. Проблеми који настају описом веома сложених система у агрегираном облику, у којем се низ елемената система редукује на мањи број компоненти и у којима се промене у систему апроксимирају константним брзинама промене. Проблеми из подручја економије и друштвених наука.

**Типови континуалних симулационих модела**

* Модели који се описују обичним диференцијалним једначинама (системи обичних диференцијалних једначина)
* Модели који се описују системима парцијалних диференцијалних једначина
* Модели динамике система (System Dinamics)

**Симулација дискретних догађаја**

Бави се моделирањем система који се могу представити скупом догађаја.

Догађај подразумева дискретну промену стања ентитета система.

Догађај наступа у одређеном тренутку времена, односно промене стања ентитета се дешавају дисконтинуално у времену, тј. само у неким временским тренуцима (када наступи догађај). Између два узастопна догађаја, стање система се не мења.

**Мешовита симулација**

Код појединих врста система, континуална симулација као и симулација дискретних догађаја, не могу у потпуности да опишу начин рада система.

Садрже процесе који теку континуално и догађаје који доводе до дисконтинуитета у понашању система.

Развијена је мешовита симулација која омогућава интегрисање континуалних и дискретних елемената система

Веза између дискретног и континуалног приступа постиже се увођењем два типа догађаја.

Временски догађаји су догађаји које генерише механизам управљања догађајима, какав постоји у симулацији дискретних догађаја. Они могу да изазову тренутну промену стања континуалне променљиве.

С друге стране, догађаји стања су они догађаји које активира механизам помака времена са константним прирастом, чији је временски интервал мали, а који је карактеристичан за континуалну симулацију. Ови догађаји могу да активирају догађаје дискретног дела модела.

**Избор типа симулационог модела**

Тип симулационог модела најчешће се одабира тако да буде једнак типу оригиналног система. То ипак не значи да увек морамо дискретни ситем приказати дискретним моделом, односно континуални систем континуалним моделом.

Избор типа симулационог модела зависи највише од циља симулационе студије.

Најважније при избору симулационог модела је да модел буде што једноставнији и разумљивији, како због његовог лакшег развоја и модификације, тако и због потребе да га корисник што лакше разуме.

Осим тога, значајан фактор при избору модела увек мора да буде и његова цена, као и ефикасност у погледу утрошка ресурса рачунара, како би се могао ефикасно користити за решавање проблема.

**Питања:** Шта је моделирање? Врсте модела? Неформални модели.Аномалије неформалних модела? Формални модели. Фазе изградње формалног модела. Шта је рачунарска симулација? Преглед моделирање и симулације? Предности симулације? Недостаци симулације? Симулациони процес? Поделе симулационих модела. Врсте симулационих модела.

**2. Класификација модела**

**Класификација модела**

Класификација у односу на променљиве

Класификација у односу на природу опсега вредности променљивих модела

Класификација у односу на природу опсега вредности променљиве "време"

Класификација у односу на временску зависност модела

Класификација у односу на детерминизам

Класификација у односу на предвиђање будућности

Класификација у односу на линеарност

**Класификација у односу на променљиве**

Код сваког модела могуће је идентификовати описне променљиве значајне за његово разумевање.

Свака променљива има свој опсег или домен.

Скуп описних променљивих се може поделити на подскуп оних које је могуће и подскуп оних које није могуће посматрати (оне које је могуће мерити неким расположивим средствима и друге које није могуће мерити).

Све описне променљиве једног модела могу се поделити на: улазне, излазне и променљиве стања.

ПРОМЕНЉИВА СТАЊА

Модели који немају ни једну променљиву стања називају се модели без меморије или тренутне функције.

Модели са бар једном променљивом стања називају се модели са меморијом.

УЛАЗНЕ ПРОМЕНЉИВЕ

Зависно од улазних променљивих модели могу бити:

неаутономни (са улазним променљивама)

аутономни (без улазних променљивих)

Класификација у односу на променљиве

ИЗЛАЗНЕ ПРОМЕНЉИВЕ

Аутономни модел који не садржи излазну променљиву назива се затворени.

Отворени модели су аутономни модели који имају излазне променљиве и сви неаутономни модели.

Неаутономни модели такође могу бити са и без излазних променљивих.

**Класификација у односу на природу опсега вредности променљивих модела**

Под опсегом променљивих се подразумева скуп свих вредности које променљива може да узме.

Описне променљиве могу узимати вредности из: пребројивог (дискретног) или непребројивог (континуалног) скупа.

У складу са тим разликујемо три врсте модела:

* **Модели са дискретним стањима -** Све три врсте описних променљивих (улазне, излазне и променљиве стања) узимају вредности из скупова чији су елементи дискретне вредности.
* **Модели са континуалним стањима -** Код ових модела све три врсте описних променљивих узимају вредности из подскупова реалних бројева.
* **Модели са мешовитим стањима -** Поједине променљиве узимају вредности из разних подскупова чији су елементи дискретне вредности, а остале променљиве из различитих подскупова реалних бројева.

**Класификација у односу на природу опсега вредности променљиве "време"**

Скуп вредности које се додељују променљивој "време“ може бити пребројив или непребројив.

Стога разликујемо моделе са континуалним временом и моделе са дискретним временом.

МОДЕЛИ СА КОНТИНУАЛНИМ ВРЕМЕНОМ

Разликујемо две подкласе ових модела:

* моделе са континуалним временом и континуалним променама стања.
* моделе са континуалним временом и дискретним променама стања (иако време тече континуално, промене стања се могу дешавати само у дискретним скоковима).

МОДЕЛИ СА ДИСКРЕТНИМ ВРЕМЕНОМ

У дискретним временским моделима, време се повећава у инкрементима који не морају бити еквидистантни.

И овде разликујемо две подкласе модела:

* моделе са дискретним временом и континуалним променама стања,
* моделе са дискретним временом и дискретним променама стања.

Променљиве симулационог модела се могу мењати на било који од четири наведена начина.

Какве ће се променљиве користити зависи од:

* ситуације која се моделира,
* од сврхе модела,
* рачунара (аналогни, дигитални или хибридни),
* симулатора и
* самог симулационог језика.

Код сложених модела, у већој мери се користе променљиве које узимају вредности из дискретног скупа и у дискретној временској основи.

**Класификација у односу на природу опсега вредности променљиве “време”**

**Класификација у односу на временску зависност модела**

Ова класификација третира питање експлицитне зависности правила интеракције модела од времена.

Уколико је структура модела (правила интеракције између објеката модела) зависна од времена, тада се ради о временски променљивом - ВАРИЈАНТНОМ МОДЕЛУ.

Kада је структура модела независна од времена, модел се назива временски непроменљив – ИНВАРИЈАНТАН или СТАЦИОНАРНИ МОДЕЛ.

**Класификација у односу на детерминизам**

ДЕТЕРМИНИСТИЧКИ МОДЕЛИ

Класификација у односу на детерминизам разматра укључивање случајних променљивих у модел.

У детерминистичким моделима, вредности променљивих стања и улазних променљивих у једном тренутку једнозначно одређују вредности променљивих стања у следећем тренутку.

Такви модели не садрже случајне променљиве.

СТОХАСТИЧКИ МОДЕЛИ

У недетерминистичким (СТОХАСТИЧКИМ) моделима постоји бар једна случајна променљива.

Већина проблема у реалном свету поседује особине стохастичности.

**Класификација у односу на предвиђање будућности**

Aнтиципаторски модели су модели који за израчунавање вредности променљивих стања узимају у обзир будућe будуће вредности улазних променљивих.

Неантиципаторски модели су модели код којих то није случај.

Код модела који служе за планирање будућности (производња, инвестиције, становништво) неопходно је узети у обзир будуће вредности улазних променљивих.

Код антиципаторских модела постоји посебан модул помоћу кога се генеришу будуће вредности улазних променљивих, а затим се симулација понавља уз коришћење претходно генерисаних вредности.

**Класификација у односу на линеарност**

Линеарни модели мењају стања и дају излазе поштујући законитости линеарних трансформација.

Једна линеарна трансформација ***L: U* → *Y*** задовољава принцип суперпозиције, ако за ***u1*, *u2* ∈ *U*** и ***c1*, *c2*** важи:

*L[c1u1(t) + c2u2(t)] = c1L[u1(t)]+c2L[u2(t)]*

Другим речима, ако су улази у систем ***u1(t)*** и ***u2(t)*,** излази из система ***y1(t)*** и ***y2(t)***, а ***c1*** и ***c2*** скалари, тада важи:

***u(t) = c1u1(t) + c2u2(t)***

***y(t) = c1y1(t) + c2y2(t)***

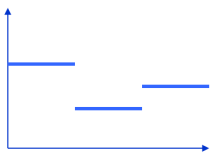
**Класификација у односу на формални опис модела**

Под формалним описом модела се подразумева прецизан, математички опис модела.

Модели са континуалним променама времена се описују диференцијалним једначинама и спадају у континуалне временске моделе.

На пример, модели са дискретним догађајима су континуални у односу на промену времена и дискретни у односу на промену стања.

Континуални временски модели не морају да мењају своје стање континуално.



На пример, модели са дискретним догађајима су континуални у односу на промену времена и дискретни у односу на промену стања.

Модели са континуалним деловима трајекторија између којих постоји дисконтинуитет такође се описују диференцијалним једначинама и могу имати две врсте дисконтинуитета и то:

дисконтинуитет извода променљиве и

дисконтинуитет променљиве.

Дисконтинуални модели су они модели у којима су најмање једна променљива стања и/или њен извод дисконтинуални.

Дискретни модели мењају своје стање у дискретним временским тренуцима.

Између два узастопна временска тренутка, стање модела остаје непромењено.

Дискретни модели могу бити: дискретни временски и континуални временски модели.

**Формална спецификација модела**

Теорија скупова омогуђава конструисање формализама који се користе за описивање објеката модела.

Свака класа објеката, може се представити одговарајуђим формализмом који дефинише њене параметре и ограничења.

Да би се дефинисао посебан објекат неке класе у оквиру неког формализма, параметрима формализма додељују се вредности које задовољавају ограничења

Класе објеката су најчешће повезане тако да се те везе могу формализовати као пресликавања из једне класе у другу.

Посебно су интересантне три врсте таквих пресликавања: апстракција, асоцијација и спецификација.

***Апстракција***

Пресликавање које подразумева контролисано укључивање детаља приликом описивања модела.

Раздвајање битних особина од небитних, како би се указало на суштину неког објекта.

Пресликавање објекта једне класе у другу, мање сложену класу.

Изворну класу називамо "конкретном", а ону другу, циљну, "апстрактном".

***Асоцијација***

Асоцијација је врста пресликавања од вишег ка нижем нивоу у хијерархији спецификације система.

Инверзно пресликавање називамо реализацијом или имплементацијом.

***Спецификација***

За дату класу објеката могуће је дефинисати подкласе, увођењем новог формализма за сваку од подкласа.

Објекте подкласа могуће је, затим, користити унутар нових формализама.

Међутим, да би успоставили релације између објеката различитих сродних подкласа, потребно је дефинисати пресликавање које преводи један посебан формализам у други, генералнији.

**Формални модел улазно-излазног система**

Посматрајмо улазно-излазни систем као скуповну структуру:



где су :

T - Временска база

U - Скуп улаза

Ω - Скуп улазних сегмената

S - Скуп интерних стања

Y - Скуп излаза

**σ** - Функција прелаза стања

λ - Функција излаза

***Временска база***

Временска база је скуп вредности времена у моделу.

На основу вредности из *T*, планира се редослед догађаја.

Избор домена за *T* је област целих бројева, или област реалних бројева *R*.

***Скуп улаза***

Скуп улаза *(U)* је део интерфејса преко кога окружење утиче на систем.

Избор домена за *U* најчешће је *Rn* за неко *n є I+*, које представља број улазних променљивих.

Друга могућност избора домена за U је: , где представља скуп екстерних догађаја.



***Скуп улазних сегмената***

Скуп улазних сегмената описује облик улаза у систем за неки временски период.

Улазни сегмент је одређен окружењем система и дефинисан пресликавањем облика



где је: - интервал временске базе између почетног и крајњег тренутка.

Скуп свих таквих улазних сегмената назива се *(U,T)*. Скуп улазних сегмената *Ω* је подскуп од *(U,T)* .

***Скуп интерних стања***

Скуп интерних стања представља меморију система, тј. утицај предхотних догађаја на његов садашњи и будући одзив.

Од избора интерних стања и њиховог карактера зависи структура модела.

Функција прелаза стања је пресликавање



Интерпретација функције прелаза стања огледа се у следећем: када се на систем са стањем *S* у времену доведе улазни сегмент



тада σ(S,ω) пресликава стање система са временом *t1*. Према томе, интерно стање у почетном тренутку и улазни сегмент од тог тренутка, надаље јединствено одређују стање система на крају сегмента.

За свако *s є S*  и *ω є Ώ* и *t* у области *ω* важи:

где су:



део *ω* између *t0* и *t1*

део *ω* између *t* и *t0*



Ово захтева да стање које припада било којем тренутку *t*, сумира потребне претходне догађаје, тако да настављање експеримента од тог стања резултује истим крајњим стањем.

***Скуп излаза***

Скуп излаза представља део интерфејса којим систем утиче на окружење.

Дефиниција је иста као и за скуп улаза с тим што је правац деловања супротан.

У оквиру једног вишекомпонентног система, излаз једне компоненте система представља улаз у другу компоненту система

У најпростијем облику функција излаза *(λ)* представља пресликавање *λ:S🡪Y* које повезује претпостављено стање система са утицајем система на његову околину.

Општија функција излаза је пресликавање *λ:SxUxT🡪Y* .

Често *λ* није једнозначно пресликавање, тако да се из окружења, стање система не може директно посматрати.

На основу дефинисаног формализма могуће је одредити појам понашања система.

Понашање система је спољашња манифестација његове интерне структуре, па је релација помоћу које се може одредити понашање система задата са *(U,T)x(Y,T)* .



Сваком стању *s є S* и улазном сегменту у ***ω*** придружена је јединствена трајекторија стања



где је:



Таква трајекторија стања резултат је аналитичког решења, или је израчуната у току извршења симулације на рачунару.

Осмотрива пројекција ове трајекторије је трајекторија излаза која се дефинише заједно са *y є Y*  и *ω є Ώ* на следећи начин:



У случају просте функције излаза *λ(s)* , добија облик

Понашање система се дефинише на основу улазно-излазне релације *Rs* :



Сваки елемент *(ω,ρ) є Rs* назива се пар улазно-излазног сегмента и представља резултат посматрања или експеримента на систему у коме је *ω* улаз у систем, док је *ρ* посматрани излазни сегмент.

С обзиром да се систем може наћи у било ком почетном стању, за један улазни сегмент *ω* може постојати више излазних сегмената *ρ*.

**Питања**: Класификација у односу на променљиве? Класификација у односу на природу опсега вредности променљивих модела? Класификација у односу на природу опсега вредности променљиве "време"? Класификација у односу на временску зависност модела? Класификација у односу на детерминизам? Класификација у односу на предвиђање будућности? Класификација у односу на линеарност? Формална спецификација модела ? Формални модел улазног-излазног система?

**3. Оцена параметара модела**

**Оцена параметара модела**

То je поступак експерименталног одређивања вредности параметара који се појављују у математичком опису модела. Полази се од претпоставке да је структура модела, односно везе између променљивих модела и параметара, експлицитно дата.

Промена вредности параметра из не-нулте у нулту вредност, може довести до поједностављене структуре модела. Претпоставка да сви параметри имају не-нулте вредности, тако да се добија јасно дефинисана структура модела и његових параметара, чије вредности треба одредити.

У пракси је тешко "повући црту” између структуре модела и одговарајућег скупа параметара.

Оцена параметара модела

Посматрајмо систем дефинисан математичким моделом у простору стања:



где су:

s - вектор стања димензије ***n***,

u - вектор улаза димензије ***m*,**

y - вектор излаза димензије ***k***,

p - вектор састављен од ***np*** непознатих параметара,

f и g - су одговарајуће векторске функције,

so- почетни услови.

Потребно је за дати модел и скуп улазних и излазних података, одредити вектор непознатих параметара ***p***.

Питање: да ли се параметри могу идентификовати, односно да ли их је могуће математички одредити?

Проблем одређивања параметара могуће је посматрати и као оптимизацију погодно дефинисане функције циља, што захтева мање почетних претпоставки.

Док јединствени приступ оцени параметара модела није могуће реализовати.

За оцену параметара за поједине класе модела користе се алгоритми чија структура зависи од :

1. Формализма модела:

* континуално-временски,
* дискретно-временски,
* линеарни или нелинеарни,
* детерминистички, стохастички.

1. Контекста моделирања:

* тип променљивих система,
* априорно знање,
* сврха модела.

1. Филозофија процене:

* критеријуми,
* нумеричка процедура,
* прилаз израчунавању

**Ангажовање ресурса по фазама изградње модела**

Поступак процене параметара није лак и једноставан посао. Напротив, то је фаза у процесу изградње модела где је потребно уложити највише истраживачког напора, времена и новца.

**Оцена параметара детерминистичког модела**

За **модел са познатом структуром** могу се увести следеће дефиниције:



За скаларни параметар каже се да је идентификабилан на интервалу ,ако постоји коначни број решења за која произилазе из релација датог модела. Скаларни параметар је неидентификабилан ако постоји бесконачан број решења за , која произилазе из релација модела.



Опис модела је системски идентификабилан, ако су сви параметри идентификабилни. У супротном, ако је један параметар неидентификабилан, опис модела је системски неидентификабилан.



Скаларни параметар је јединствено идентификабилан на интервалу ,ако постоји јединствено решење за које произилази из релација модела са датим улазом.



Спецификација модела је параметарски идентификабилна на интервалу ако су сви параметри јединствено идентификабилни.

Идентификабилност система и параметара зависи од два фактора:

1. ***Специфичности примењеног улаза и почетних услова.***

Почетни услови су битни јер идентификабилност система и параметара може зависити од њихових вредности.

Кад је у питању улаз, може се догодити да се за дати систем улазни сегмент може поделити у више посебних категорија.

Неке од њих могу обухватити тип идентификабилности, док друге могу изазвати различите проблеме неидентификабилности.

1. ***Структуре једначина и ограничења***

Неке структуре се могу идентификовати ако се примене одговарајући улази, док су друге неидентификабилне, без обзира на то какав се улаз примени.

**Оцена параметара стохастичког модела**

Код стохастичких система, излазне секвенце су стохастички процеси, тако да се проблем усложњава захтевом за укључивање параметара појединих реализација процеса.

Проблем идентификабилности се може поделити у три дела, односно идентификабилност зависи од следећа три фактора:

**Структуре** - која параметре ставља у релацију једне с другима као и са улазима и излазима.

**Улаза** - од којих неки могу бити сувише "прости" да би се постигла идентификабилност.

**Процедуре процене** - које треба да буду конзистентне у смислу да конвергирају ка стварним вредностима параметара.

**Статистички приступ процени параметара статичких модела**

Статистичке оцене параметара статичких модела се могу реализовати у:

* За идентификацију параметара система за које су подаци о њиховом понашању прикупљени и меморисани на неком медијуму, најједноставније је применити секвенцијалне статистичке методе.
* За прикупљање података у реалном времену, када је број података такав да се не може меморисати, треба користити одговарајуће статистичке формуле у рекурзивној форми.

Предност рекурзивног приступа се огледа у томе што је за рачунање најчешће потребно само неколико променљивих које се ажурирају у току експеримента (на тај начин се штеди у меморијским ресурсима) и што су приликом сваког ажурирања статистичких показатеља познате њихове текуће вредности.

**Оцена средње вредности случајне промењиве**

Као илустрација за процену средње вредности случајне променљиве за *N* узорака, користићемо секвенцијални метод и рекурзивни метод.

СЕКВЕНЦИЈАЛНИ МЕТОД

Уз претпоставку да су сви резултати мерења на реалном систему доступни у време израчунавања, процедура за процену средње вредности може се дефинисати на следећи начин:



Дакле, сва мерења се сабирају и резултат дели са бројем узорака.

РЕКУРЗИВНИ МЕТОД

За рекурзивно израчунавање је карактеристично следеће:

* база података се проширује током рачунања, међурезултати за одређени број узорака су такође доступни и они теже секвенцијалном решењу како израчунавање одмиче.

Рачунање се одвија по следећој процедури:



где је:

 - текући узорак (*к =1,2,..,N*)

**Оцена непознатог параметара по методи најмањих квадрата**

Посматрајмо статички модел код кога је веза између параметра модела и резултата експеримента дата релацијом:





Претпоставке су:



- познате вредности

*ei* - грешке мерења

Потребно је проценити непознати параметар *a* користећи доступне резултате мерења.

Оцена непознатог параметара по методи најмањих квадрата

СЕКВЕНЦИЈАЛНО РЕШЕЊЕ

Циљ је да се минимизира одступање између података које генерише реални систем и података који се добијају на основу модела, односно да се минимизира квадрат грешке:



одакле следи:

из чега се директно може одредити непознати параметар *а*.

Оцена непознатог параметара по методи најмањих квадрата

РЕКУРЗИВНО РЕШЕЊЕ

Уведимо следеће ознаке: 

где је:



У рекурзивној форми ове формуле можемо написати:

**



Ако дефинишемо kk у облику:



тада се оцена параметра  може изразити као:

****

**Процена *к* непознатих параметара**

Посматрајмо случај где је веза између *k* параметара модела и резултата неког експеримента дата следећом релацијом:



где су параметри непознати.

Претпоставимо да смо *n* пута обавили експеримент на систему, чији се резултати могу приказати следећом табелом:

*Општи облик резултата за n експеримената*

С обзиром да у већини случајева није могуће извршти тачна мерења, експериментални резултати неће задовољавати горњу релацију, већ ће постојати одређена одступања (грешке) која ћемо обележити са *е* .

За *n* експеримената и *k* променљивих добија се систем једначина облика:



Оцене непознатих параметара одређују се минимизацијом критеријумске функције:



Минимум функционала *J* добија се изједначавањем његових парцијалних извода по параметримa са нулом, односно решавањем система од *k* једначина.



У овом случају је знатно једноставније применити матрични рачун.



Процена *к* непознатих параметара



Критеријумска функција се тада може представити у облику:



Применом правила диференцирања за матрице и векторе добија се



одакле се добија



односно

је релација која даје непознате параметре модела.

**4. Валидација и верификација**

Валидација и верификација

Модел се мора тестирати како би се установило да ли је он поуздан, без грешака и да ли је довољно уверљив за све оне који ће га користити.

Поступци којим испитујемо колико верно и прецизно један модел представља реални систем су:

**Верификација**

Односи се на проверу да ли је симулациони програм, којим се имплементира модел, без грешака и конзистентан са моделом.

То је у ствари поређење концептуалног модела са рачунарским кодом којим се таква концепција имплементира.

**Валидација**

Односи се на поступак одређивања да ли јe модел прецизна репрезентација реалног система.

Најчешће је то једна итеративна процедура у којој се понашање модела пореди са понашањем реалног система и уочена неслагања и разлике користе за доградњу и исправку модела.

Поступак побољшања модела се наставља, све док се не одлучи да добијена тачност модела задовољава.

**Изградња модела, верификација и валидација**

**Кораци у изградњи модела**

***Први корак*** у изградњи модела обухвата:

* посматрање реалног система и интеракција између његових различитих компоненти
* прикупљање података о понашању система.

Међутим, у том поступку моделар не сме да се ослони само на своја запажања, пошто то често води погрешном схватању самог система и његовог понашања, већ је неопходно ангажовање стручњака који добро познају систем или неке његове делове.

Специфична знања о појединим аспектима система, која ови стручњаци различитих профила поседују, од великог су значаја за моделара и будући ток изградње модела.

***Други корак*** у изградњи модела је формирање:

* концептуалног модела,
* скупа претпоставки за компоненте модела и структуру система и
* хипотеза за вредности параметара модела.

Концептуална валидација је поређење реалног система и концептуалног модела.

***Трећи корак*** у изградњи модела је писање рачунарских програма за симулационе моделе.

**Валидација симулационих модела**

Проблем валидације модела настаје пошто се у фази изградње модела уносе многе АПРОКСИМАЦИЈЕ реалног система.

На таj начин, ограничавамо модел (успостављамо границе у односу на окружење), игноришући све оно изван њега што није експлицитни улаз и одбацујемо факторе за које сматрамо да су неважни.

Врсте апроксимација:

* Функционална апроксимација
* Апроксимација расподеле
* Апроксимација независности
* Апроксимација агрегације
* Апроксимација стационарности

***Функционална апроксимација***

Изразито нелинеарне функције често се апроксимирају неким једноставнијим функцијама, на пример линеарним.

Важна претпоставка је да једноставнија фукција треба да буде приближна "оригиналној" функцији у области где ће систем вероватно функционисати.

Ако програм мора да функционише и у области где је поклапање функција слабо, потребно је обезбедити да он штампа поруку упозорења.

***Апроксимација расподеле***

Реалне вероватноће расподела које су и саме познате једино апроксимативно, често се апроксимирају једноставнијим расподелама, као што су нормална или експоненцијална.

Најекстремнији пример апроксимације је онај када се случајна променљива замени константом.

***Апроксимација незавaсности***

Модел се често поједностављује тако што се претпоставља да су различите компоненте (описане случајним променљивим) статистички независне.

***Апроксимација агрегације***

Под агрегацијом подразумевамо ситуацију када више елемената посматрамо као једну целину.

* Временска агрегација- Интервал времена као што је дан, третира се као један појединачни период. За све догађаје који се десе током дана, претпоставља се да су се десили истовремено.
* Међу-секторска агрегација- Више оделења, фирми, производних линија, итд. посматра се као једна целина.
* Агрегација помоћних средстава - Овде се више помоћних средстава посматра као једно. На пример, рачунарски систем са два централна процесора у паралелној вези, након агрегације посматраћемо као један, два пута бржи централни процесор.

***Апроксимација стационарности***

Код неких система параметри и друге карактеристике система не мењају се у времену.

За извесне физичке процесе ово може бити прихватљива почетна апроксимација а за неке не.

За извесне физичке процесе, као што су неке астрономске појаве, ово може бити прихватљива почетна апроксимација. Међутим, у политичким, економским, организационим и др. системима, свакодневно искуство указује да је већина појава нестационарне природе те је претпоставка постојане структуре неодржива.

**Циљ процеса валидације**

Да произведе модел који представља понашање реалног система и који је довољно близак реалном систему, тако да се може користити за обављање експеримента.

Да поузданост модела повећа на прихватљив ниво, тако да модел могу користити различити доносиоци одлука.

**Итеративни процес валидације модела**

**Валидација симулационих модела**

Процес валидације модела треба посматрати као интегрални и незамењиви део развоја модела.

Концептуални модел се пореди са реалним системом и уколико постоје неслагања и разлике за које се сматра да нису оправдане, врши се исправка (или чак значајнија промена) на моделу, затим се тако добијени модел поново пореди са реалним системом и врше нове исправке уколико је то потребно, итд.

Итеративни процес се наставља све док се не постигне задовољавајућа тачност модела.

Поређење модела са реалним системом врши се помоћу више различитих тестови:

Субјективни тестови најчешће подразумевају учешће људи који добро познају систем или неке његове аспекте и који, примењујући посебна знања о систему, доносе закључке о моделу и његовом излазу.

Објективни тестови увек захтевају неке податке о понашању система, као и податке које производи модел.

**Практични приступ процесу валидације**

1. Изградити модел који верно представља реални систем
2. Потврдити претпоставке модела
3. Упоредити улазно - излазне трансформације модела са одговарајућим У-И трансформацијама реалног система.

**Процена валидности модела**

Изградити модел који ће изгледати разумно корисницима или другима који поседују знања о реалном систему који се симулира.

Уkључивање потенцијалних корисника модела у процес изградње модела од почетка, како би се осигурало да се у модел "угради" висок степен реализма кроз разумне претпоставке које се односе на структуру система и поуздане податке.

За проверу валидности модела, може се користити и АНАЛИЗА ОСЕТЉИВОСТИ.

То је поступак тестирања осетљивости модела на различите претпоставке и промене улазних величина.

Промене излазног понашања модела, као последица промена улазних величина, могу пружити корисне информације о моделу и кориснику и моделару.

Код већине симулационих модела, постоји велики број улазних величина, па самим тим и велики број могућих тестова осетљивости.

Зато је потребно да моделар одабере оне улазне величине за које сматра да су "најкритичније" и њих тестира, пошто је врло вероватно да време и финансијска средства неће дозволити да се испита осетљивост модела на све улазне променљиве.

Уколико подаци о реалном систему омогућавају барем две измене улазних параметара, тестови осетљивости се могу допунити коришћењем одговарајућих статистичких техника.

**Валидација претпоставки модела**

Претпоставке модела је могуће сврстати у две генералне категорије:

* Претпоставке о структури се тичу питања како функционише систем и често обухватају поједностављења и апстракције реалног система.
* Претпоставке о подацима треба да буду засноване на скупу поузданих података и исправној статистичкој анализи тих података.

Статистичка анализа углавном се састоји из три корака:

1. Идентификација одговарајућих расподела добијених података.
2. Процена параметара изабране расподеле.
3. Валидација претпостављеног статистичког модела неким тестом (нпр. **χ** - квадрат или Колмогоров-Смирнов тест) .

**Валидација улазно-излазних трансформација**

Једини објективни тест модела као целине је провера способности модела да предвиди будуће понашање реалног система, када улазни подаци модела одговарају улазним подацима реалног система.

Aко се ниво неке улазне величине повећава или смањује, модел треба прецизно да предвиди шта ће се десити у реалном систему под истим околностима.

Структура модела треба да буде довољно прецизна да модел даје добра предвиђања и то не само за један скуп улазних података већ за цео опсег скупова улазних података који су од интереса

Модел се посматра као улазно - излазна трансформација.

Да би се извршила валидација улазно - излазне трансформације, неопходан услов је постојање система који се проучава, како би било могуће прикупити податке о систему за барем један скуп улазних података.

**Формални критеријум за утврђивање валидности модела**

* Хомоморфизам (грчка реч: homo - сличан, morph - структура)

Формални критеријум за утврђивање валидности упрошћеног модела за дате експерименталне услове, у односу на основни модел чије су карактеристике објашњене раније.

Суштина формалног поступка за проверу валидности је у томе да се нађе пресликавање *H* помоћу кога је могуће из сваког стања основног модела (нпр. ***s***) прећи у одговарајуће стање упрошћеног модела (нпр. ***s*′**).

Ако овакво пресликавање постоји, тада се закључује да су улазно - излазна понашања основног и упрошћеног модела иста за дате експерименталне услове.

Да би се утврдио хомоморфизам, морају бити испуњени следећи услови:

* Очување функције наступања времена
* Очување функције прелаза стања
* Очување излазне функције

**Очување функције наступања времена**

Стање у основном моделу и њему одговарајуће стање у упрошћеном моделу морају имати исту функцију наступања времена.

То значи да ће се стања основног модела и стања упрошћеног модела мењати у истим тренуцима времена.

**Очување функције прелаза стања**

Ако стању *s* основног модела одговара стање *s′* упрошћеног модела (тј. применом пресликавања *H* на стање *s* добија се стање *s*′), тада и следећа стања у која ће модел прећи у наредном тренутку посматрања морају одговарати једно другом.



Следеће стање основног модела ће бити , а упрошћеног модела , што значи да применом пресликавања *H* на стање треба да се добије стање .



**Очување излазне функције *λ***

Нека је *s* стање основног модела и нека је могуће применом пресликавања *H* прећи у стање *s′* упрошћеног модела.

Примена излазне функције основног модела (за оне експерименталне услове за које је извршено упрошћавање (*λ*) да би се добио упрошћени модел за стање *s*), даје исти излаз *Y* као и примена излазне функције упрошћеног модела (*λ′*) на стање *s′*.

Овде се јасно уочава да се валидност упрошћеног модела утврђује и у односу на задате експерименталне услове.

Уколико је пресликавање *H* типа *1:1*, у питању је ИЗОМОРФИЗАМ.

**Верификација симулационих модела**

Процес *верификације* треба да покаже да ли је и у којој мери, концептуални модел на одговарајући начин представљен рачунарским кодом,

У којој се мери слажу концептуални (претпоставке за компоненте система и структуру; вредности параметара; апстракције и поједностављења у моделу) и рачунарски код.

У поступку верификације нема стандардног рецепта.

Зато је потребно извршити више различитих провера:

* **Ручна верификација логичке исправности**: модел се извесно време пропушта на рачунару и ручно, а потом пореде добијени резултати.
* **Модуларно тестирање**: појединачно тестирање сваког модула како би се установило да ли даје разумне излазе за све могуће улазе.

Зато је потребно извршити више различитих провера:

* **Провера у односу на позната решења**: подесимо модел тако да представља систем чија су решења позната и упоређујемо их са резултатима модела.
* **Тестирање осетљивости**: варирамо један параметар, док остали остају непромењени и проверавамо да ли је понашање модела осетљиво на промену тог параметра.
* **Тестирање на поремећаје**: постављамо параметре модела на неприродне вредности и проверавамо да ли се модел понаша на несхватљив начин. На тај начин се могу открити грешке у програму које је врло тешко уочити на други начин.

**5. Средства за симулацију**

Као средства за симулацију користе се три врсте рачунара: аналогни, дигитални и хибридни рачунар.

**АНАЛОГНИ РАЧУНАР**

Физички објект А, који се користи за анализу математичког модела објекта Б, са којим има сличан математички модел, назива се АНАЛОГНИ МОДЕЛ.

Између физичког објекта који се испитује и аналогног модела постоји математичка аналогија (аналогија у понашању).

Идеја аналогних рачунара (односно свих аналогних рачунских средстава) састоји се у изналажењу таквих физичких објеката који ће моћи да се користе за анализу математичких модела помоћу којих су ти објекти математички описани.

**Пример - Аналогни рачунар**

Претпоставимо да је потребно да се изврши множење две величине x и y. Према томе, задати математички модел је

z =xy (5.1)

За електрично коло важи Омов закон, који даје везу између напона (U) , струје (I) и отпора (R), тако да је

U =RI (5.2)

Како електрично коло има математички модел који је идентичан са математичким моделом (z=xy), само што ознаке у моделу (5.1) представљају математичке величине, а у моделу (5.2) физичке величине, то електрично коло на слици 5.1. може да представља АНАЛОГНИ УРЕЂАЈ ЗА МНОЖЕЊЕ.

Принципи рада аналогних рачунара засновани су на физичким законима који важе за онај физички објект који се користи у рачунске сврхе.

Аналогни рачунари (АP) деле се најчешће у две групе и то: специјални и универзални аналогни рачунари.

*Специјални аналогни рачунари* граде се у специјалне сврхе и служе за анализу једног или евентуално мањег броја специфичних математичких модела.

*Универзални аналогни рачунари* служе за анализу разних математичких модела. Они се састоје из више рачунских компоненти, при чему свака од њих може да обави једну или више математичких операција.

**Подела аналогних рачунарских средстава**

**Репетитивни АP**

Раде великом брзином и на њима се решење обично понавља 20 до 1000 пута у секунди.

Велика учестаност решења омогућава његово визуелно праћење на екрану осцилоскопа (уређаја који на екрану показује промену напона у времену) у виду криве линије.

**Спори АP**

Спори АP раде у тзв. реалном времену, тј. тако што се решење добија само у једном циклусу рада рачунара.

Да би решење било трајно записано, обично се црта на специјалном *XY* писачу.

Савременији АP могу да раде комбиновано - било као спори, било као репетитивни.

**Електронски аналогни** рачунари за нас су интересантни са становишта симулације континуалних система.

Развој електронике и појава електронских цеви, четрдесетих година овог века, а нарочито појавам транзистора и технологије интегрисаних кола, омогућени су услови за развој аналогних рачунара.

Електронски АР свој рад заснивају на аналогији између понашања електронских кола која свој рад заснивају на физичким законима и њиховим описом помоћу диференцијалних једначина.

**Електронско аналогни рачунари**

* Симулацију неке појаве постављамо у облику диференцијалне једначине.
* Диф. једначину представљамо помоћу електронских склопова.
* Решења диф. једначине добијају се на основу мерења напона и струје у електронским склоповима.

Повезивању рачунарских компоненти АР у циљу постављања конкретног математичког модела на рачунару, назива се ПРОГРАМИРАЊЕ.

Математички модели који се најчешће решавају на аналогним рачунарима су системи диференцијалних једначина, па се зато ови рачунари називају често ДИФЕРЕНЦИЈАЛНИ АНАЛИЗАТОРИ.

**Рачунарски елементи електронског АР**

**Рачунски елементи АР** деле се у шест група и то према **функцији** коју обављају:

1. Множење променљиве величине константом
2. Алгебарско сабирање више променљивих величина
3. Интеграција једне или више функција
4. Множење и дељење променљивих величина
5. Генерисање функција
6. Логичке операције.

**Главни елементи електронског АР**

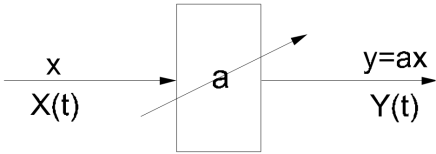
* Напонски извор (константа)
* Потенциометар (улаз множи са константом мањом од 1)
* Појачивач (улаз множи са константом већом од 1)
* Интегратор (излазни напон је интеграл улазног)
* Разни диодни ограничавачи (нелинеарни елементи)
* Диодни генератори нелинеарних функција.

**Потенциметар** се реализује као омски отпорник са клизачем, на чије се крајеве доводи променљиви напон *X(t)*, а са клизача, који може да се постави у ма који положај између крајева потенциометра, одводи се напон *Y(t)* .

По природи овог уређаја мора да буде *X(t) ≤ Y(t)*

Потенциометар омогућује множење променљивог улазног напона са константом мањом од јединице.

Вредност константе одређена је положајем клизача на скали потенциометра.



Савремени ЕАР имају потенциометре који су снабдевени скалама од 0 до 1.

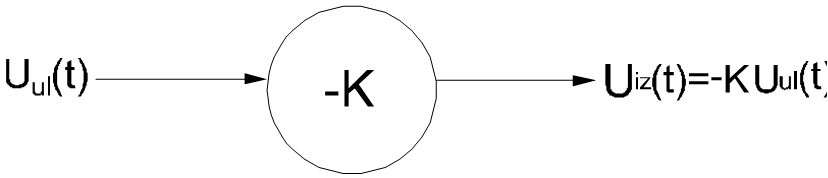
Лако се бира жељени положај клизача (вредност константе), али се потенциометар не може никад подесити идеално тачно, јер свако постављање клизача на неку дужину уноси извесну грешку .

**Појачавач**

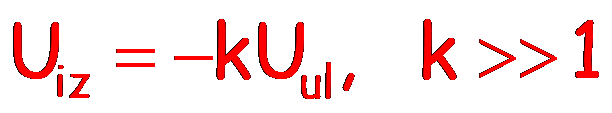
Код ЕАР најважнију улогу има појачавач, који се назива још и операциони или рачунски појачавач.

Појава овог елемента је омогућила настанак ЕАР и зато се с правом он третира као главни рачунски елемент аналогних рачунара.

Појачавач се реализују искључиво у форми интегрисаних кола.



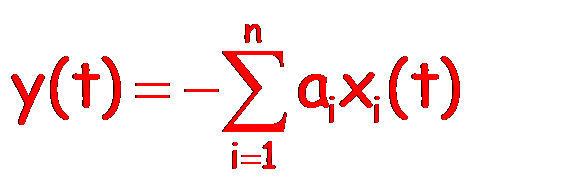
То је електронски уређај који има врло велико појачање, односно појачава улазни сигнал (улазни напон), тако да се на излазу добија напон знатно појачан у односу на онај на улазу.



Поред тога, операциони појачавач има велику улазну, а малу излазну отпорност.

**Сабирач**

Математичка операција алгебарског сабирања *n* променљивих величина коју желимо да извршимо, дата је изразом:

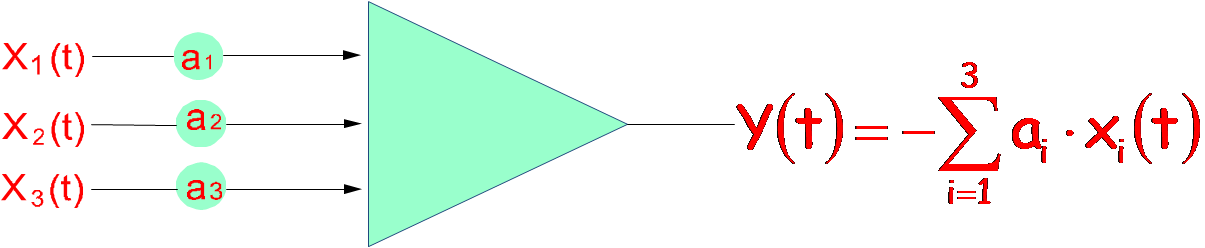


где су  константе којима се множи одговарајућа променљива величина  .

Рачунски елемент који омогућава реализацију ове математичке операције назива се сабирач.

Физичке величине,које одговарају променљивим ,су улазни напони сабирача. Напон као излазна величина из сабирача одговара математичкој величини .

Сабирач се реализује коришћењем одређеног броја отпорника и увођењем негативне повратне спреге.

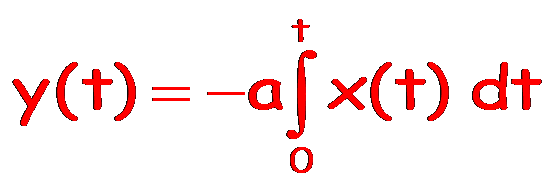


Сабирач може да обави две математичке операције:множење улазног напона са константом и алгебарско сабирање више улазних напона.

Сабирач је могуће користити и као ИНВЕРТОР, тако што ће се довести само један улаз са константом.

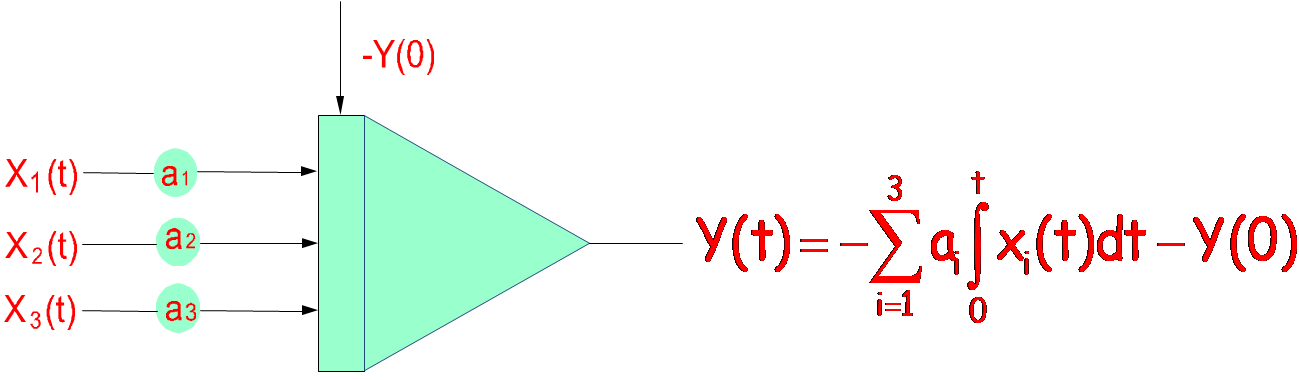
**Интегратор**

Математичка операција која се жели реализовати помоћу интегратора има облик:



где је ***а***константа. Иста ознака *t* употребљена је за горњу границу интеграла и независну променљиву у подинтегралној функцији, јер је време *t* једина независна променљива на рачунару.

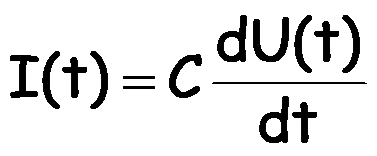
**Шематски приказ интегратора**

****

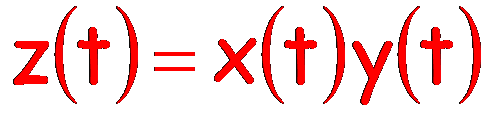
Физички елемент који се код аналогних рачунара користи за обављање операције интеграције је КОНДЕНЗАТОР.

Закон физике о односу између струје, напона и капацитета кондензатора.

*ако је на крајевима кондензатора капацитета C доведен напон U, онда је струја I кроз кондензатор одређена односом:*

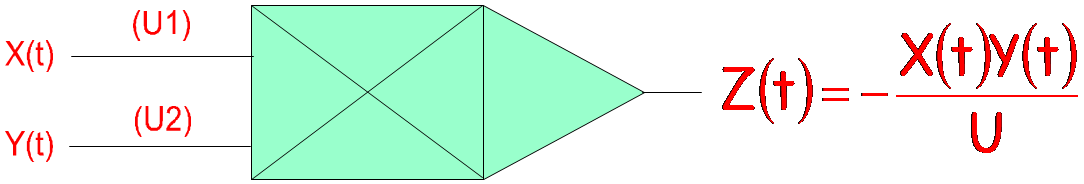
**

**Множач** је рачунски елемент на електронском аналогном рачунару који може да обавља следећу математичку операцију:

**** где су *x(t)* и *y(t)* променљиве величине са временом.

Сви типови множача могу поделити у две главне групе и то:

* множачи који користе електромеханичке методе и који се називају сервомножачи.
* множачи који користе електронске методе и који се називају електронски множачи.



**Сервомножач**

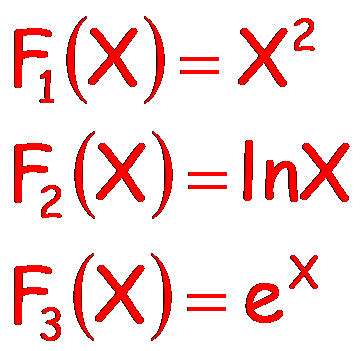
Идеја сервомножача потекла је од особине потенциометра да улазни напон множи са неком константом мањом од јединице.

Код ове врсте множача за покретање клизача потенциометра се користи тзв. позициони сервомеханизам, па отуда и име сервомножач.

**Диодни множачи**

Диодни множачи припадају групи електронских множача, а састављени су од диодних генератора неких фиксних функција и рачунских појачивача.

Најчешће се користе диодни генератори функција који дају следеће фиксне функције:



Први тип диодног електронског множача базира се на основној једначини:



Други тип електронског диодног множача базира се на основној једначини:

, где је .

**Генератори функција**

У аналогној рачунарској техници, физички уређај који може да оствари математичку операцију назива се генератор функција.



Математички модел који треба да има генератор функција биће: 

Оператор *f* може се изразити било у математичком облику:

* као нека аналитичка функционална зависност,
* као нека графичка зависност (крива линија) или
* као табела парова дискретних вредности за *x* и *y*.

У аналогној рачунској техници данас се користе две групе генератора функција.

* Генератори специјалних, фиксних функција, код којих се жељена функција поставља једном за свагда.
* Универзални генератори функција који се могу по жељи подешавати на ону функцију за коју се укаже потреба приликом решавања одређеног проблема.

**Поступци при раду са аналогним рачунаром**

1. Први корак у аналогном моделовању и симулацији је да се коректно формулише математички модел физичког објекта.
2. На основу математичког модела, програмер саставља блок дијаграм.
3. На основу блок шеме, врши се повезивање рачунарских елемената помоћу електричних водова.

Добра страна овакве представе је да се решења нарочито за алгебарске једначине добијају готово тренутно.

Лоша страна је да је максимална тачност решења ограничена:

* тачношћу реализације електронских елемената,
* тачношћу мерних инструмената и мерним методама на основу којих добијамо решења.

Због наведених разлога сматра се да је максимална тачност аналогног рачунара ограничена грешком од приближно 1% .

**Дигитални рачунар**

Структура класичног дигиталног рачунара је следећа: процесор, меморија и улазно-излазни уређаји.

Дигитални рачунари имају велику тачност рачунања (која у пракси иде до 20 значајних цифара) али и незгодну особину да се за решавање алгебарских једначина користе итерационе методе које у већини случајева траже доста рачунарског времена.

Структура симулационих модела који се симулирају на дигиталном рачунару може бити врло сложена и састављена од низа паралелних процеса који међусобно координирају и синхронизују своје акције.

Симулација таквих модела на класичном једнопроцесорском рачунару захтева врло сложене софтверске концепте за синхронизацију и координацију, па је заступљеност симулационих метода тиме била ограничена.

Да би се наведена ограничења премостила, развојем технологије хардвера и софтвера уведени су нови концепти који су омогућили смањење трошкова изградње и експлоатације симулационих модела и омогућили брз развој симулације.

Хардверски новитети који су значајни за развој симулације и проширење делокруга примене су:

* Векторски процесори
* Рачунари са више процесора
* Јефтине и брзе динамичке меморије
* Јефтине и брзе масовне меморије
* Појава брзих графичких процесора.

Развој софтверских компоненти које имају значајан утицај на развој симулације су у следећим областима:

* Развој вештачке интелигенције
* Развој алгоритама паралелног процесирања
* Рачунарске мреже
* Развој мултимедијалних технологија

**Хибридни рачунар** је спој аналогног и дигиталног рачунара изведен на такав начин да се предности сваког од њих користе у потпуности док се недостаци међусобно анулирају.

Код хибридног рачунара, аналогни рачунар изводи имплицитне операције и извршава интеграљење, док дигитални рачунар извршава алгебарске функције.

Због високе цене хибридних рачунара они се користе углавном за симулацију у авионској индустрији и управљању летелицама, где је од изузетне важности одређивање решења диференцијалне једначине у кратком временском интервалу.

Хибридна обрада података представља комбинацију паралелне и секвенцијалне обраде података.

Таквом начину обраде одговара тзв. хибридни процесор.

Хибридни процесор је хардверско-софтверски систем који може да врши и паралелну и секвенцијалну обраду. Састоји се од паралелног (аналогног) процесора и једног или више секвенцијалних процесора.

**Предности и фибридног рачунара**

Циљ оваквог повезивања је да се добре стране оба начина обраде података повежу и да се, колико је то могуће, елиминишу ограничења која извесно сваки од наведена два начина обраде поседује.

Другим речима, хибридна обрада је покушај да се комбинују, с једне стране, тачност и могућност меморисања података дигиталног рачунара и с друге стране, велика брзина извођења операција аналогног рачунара.

За време паралелне обраде, обрада свих података почиње и завршава се у истом тренутку.

Ова се обрада обавља на аналогном процесору, који сачињава више оперативно независних подпроцесора.

Секвенцијална обрада може се дефинисати као обрада коначног броја података у низу (један за другим) и обавља се најчешће на само једном дигиталном процесору.

Типичан хибридни рачунарски систем састоји се од три повезана дела:

* Самосталног дигиталног рачунара са свом стандардном периферном опремом коју једна таква инсталација подразумева,
* Једног аналогног рачунара одговарајуће величине и
* Интерфејса који повезује та два рачунара.

Интерфејс је неопходан због различитог начина функционисања и различитог начина представљања података у оба рачунара.

**6. Симулација континуалних система**

Симулација континуалних система (СКС) – извођење експеримената над моделима система чија се стања мењају континуално у времену.

Ови системи су већином динамички и могу бити детерминистички или стохастички.

Најчешће се представљају одређеним бројем диференцијалних једначина, које се решавају нумеричким путем, како би се одредило понашање модела. Време је најчешћа независна променљива у овим моделима.

*Пример* - рад електричног грејача са термостатом. Када се укључи, он загрева просторију све док се не достигне одређена температура, подешена на термостату. Када се та температура постигне, термостат искључује грејач.

|  |  |
| --- | --- |
| Picture12.png | Повећање температуре просторије и њено постепено смањивање које настаје након искључивања грејача, представља континуални процес који се може представити одговарајућим диференцијалним једначинама и потом симулирати. |

**Формални модел СКС**

Математички модел за симулацију континуалних система може се представити шесторком:



где су:

**U** - скуп улаза

**Y** - скуп излаза

**S** - скуп променљивих стања

**δ** - функција преноса 

**λ** - функција излаза 

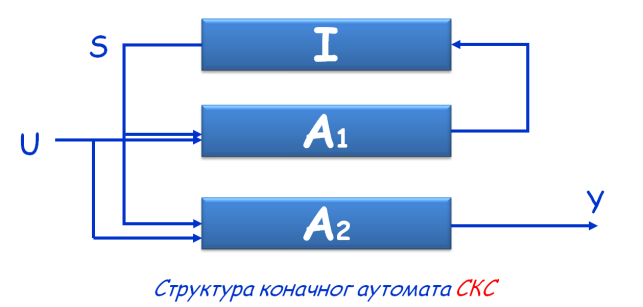
**S0** - скуп почетних стања (почетних услова)

Процес СКС може се описати једначинама континуалног аутомата:

 и 

Преносна функција, у горњим једначинама, раздељена је у две врсте оператора: **A1** и **A2** су алгебарске функције; ***I*** представља интеграцију.

**Формални модел СКС**

****

Основна разлика између аналогне и дигиталне симулације континуалних система није у структури већ у процедури.

**Функција једне или више промењивих- БЛОК**

АР- решавање рекурзивних релација.

ДР - мора применити итеративну процедуру.

АР - све операције представљене операторима и извршавају истовремено;

ДР - најпре мора да изврши све алгебарске операције, а потом све интеграције, једну за другом.

**Формални модел СКС Блок дијаграм детаљне структуре СКС**

Ако разложимо оператор на његове елементарне или примитивне функције, које могу да буду представљене алгебарским блоковима, добијамо детаљнију структуру процеса СКС.

**Функција једне или више промењивих - БЛОК**

Функција једне променљиве **ϕ** је пресликавање непразног скупа *X,* променљивих ***x***, званог домен, у непразан скуп *Y,* променљивих ***y***, званог опсег. Симболички се представља са: 

Функција више променљивих ϕ је пресликавање облика 

Блок се може представити уређеном тројком .

**Апстрактни континуални симулациони системи**

Нека је *V* скуп променљивих, *t* време симулације и *B* скуп блокова.

Тада се тројка: назива АПСТРАКТНИ КОНТИНУАЛНИ СИМУЛАЦИОНИ СИСТЕМ КСС*.*

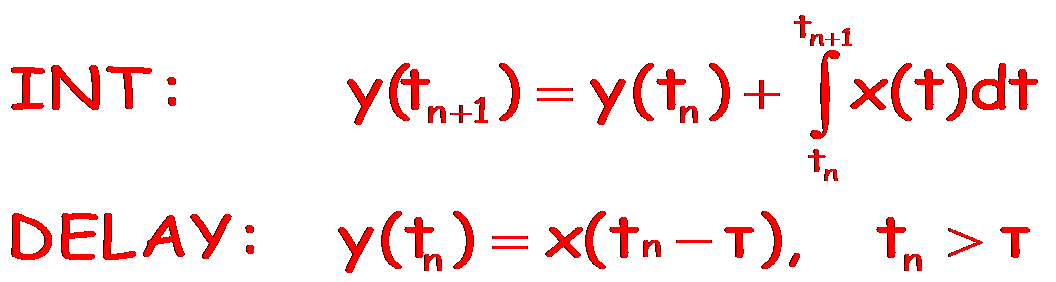
Нека су: *X* скуп свих улаза једног блока и *Y* скуп свих излаза тога блока.

Тада се скуп свих променљивих *V* може поделити у три дисјунктна подскупа и то:

* скуп улаза 
* скуп излаза 
* скуп веза 

**Функције стања, алгебарске функције, променљиве стања**

Функције стања:



Променљиве стања: које представљају излазе блокова за претходно наведене функције.

Mогу бити из скупа: веза ( *C* ) или излаза ( *R* ). 

**Уредљивост**

Најосновније својство било ког нетривијалног математичког модела КСС је повратна спрега.

Модел има повратну спрегу ако је немогуће означити све блокове  b тако да, ако је излаз блока bi везан на улаз блока bj будe .

Појава затворене повратне спреге намеће неопходан услов за израчунљивост математичког модела, који се назива услов уредљивости

Скуп *A* свих пребројивих алгебарских блокова модела *M* назива се уредљивим ако сви чланови  могу бити уређени (сортирани) у линеарну листу такву да су улази сваког члана, елементи неког од следећих скупова:

* скупа улаза *U*
* скупа стања *S*
* подскупакоји је дефинисан као 

**Симулација континуалних система помоћу аналогног рачунара**

Аналогни рачунари заснивају свој рад на чињеници да се различити физички системи описују истим математичким моделима.Такав математички опис физичких појава се назива физички закон. Из тога произилази да велики број система можемо описати истим једначинама, користећи принцип налогије у поношању.

Аналогни рачунари могу симулирати било који систем за који се физички систем, чији је математички опис аналоган систему који симулирамо може поставити.

По својој природи, физички системи су континуални системи, па се речи аналогни и континуалничесто користе као синоними.

***Предности симулације КС помоћу аналогних рачунара***

* могућност директног приступа било ком функционалном делу аналогног програма;
* могућност обављања великог броја понављајућих операција комбинованих са веома великом брзином израчунавања.

***Недостаци симулације КС помоћу аналогних рачунара***

* не постоји еквивалент репрезентацији података у покретном зарезу;
* мора се водити рачуна да се избегну могућа прекорачења и грешке сведу у подношљиве оквире, па се подаци скалирају.
* не могу се избећи различита техничка ограничења;
* ко рукује аналогним рачунаром мора да поседује одређена знања о компонентама рачунара и раду са АР.

**Симулација континуалних система помоћу дигиталног рачунара**

Основна разлика између аналогног и дигиталног рачунара није у начину представљања података, већ у принципу функционисања.

Основна карактеристика дигиталног рачунара је секвенцијално обављање операција.

***Предности дигиталног рачунара у СКС:***

* велики ниво апстракције код писања програма
* велика тачност решења.

***Недостаци дигиталног рачунара у СКС:***

* брзина извођења операција знатно мања него код АР
* могућa нестабилност нумеричке интеграције.

Да би се извршила дигитална СКС све функције независно променљиве ***t*** треба да буду представљене секвенцом дискретних бројева у тачкама: . Величина корака може, али не мора бити константна.

Дигитална СКС је дискретан алгоритамски процес, који се извршава корак по корак.

Резултат процедуре симулације је ЛИСТА ДИСКРЕТНИХ БРОЈЕВА за свако стање променљиве и сваки излаз.

На почетку симулације, потребно је означити нумеричке вредности сваког улаза и сваке променљиве. Те вредности називамо почетним условима.

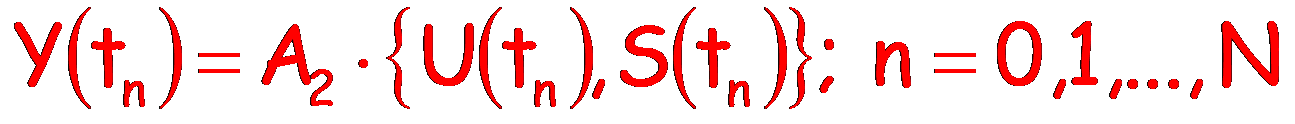
Извршавање симулације се састоји из фазе иницијализације и фазе рачунања.

У фази иницијализације постављају се:

* почетне вредности и почетни услови,
* Параметри и
* врше нека израчунавања везана за припрему нумеричке интеграције.

У симулационом алгоритму n-ти корак се састоји од израчунавања вредности функције свих алгебарских блокова, на основу скупа улаза и променљивих стања датих у tn  и израчунавања свих променљивих стања за наредни корак tn+1 у складу са основним једначинама:

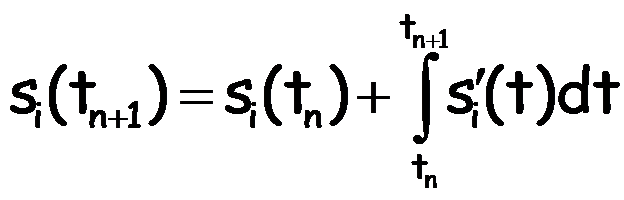


****

Прва једначина означава да се крећемо кроз уређену листу алгебарских функција, и израчунавамо по датом редоследу, за сваки блок, вредност његовог излаза.

Потом променљиве стања нумерички интегралимо зависно од тренутне и неке претходне вредности променљиве стања  и њеног првог извода.

Величина треба да буду израчунате неком од нумеричких метода тако да функција



буде апроксимирана онолико тачно колико је потребно

**Интеграција у симулацији континуалних система**

У аналогној симулацији основни проблеми настају због ниског нивоа апстракције у програмирању, техничке преосетљивости рачунара, а нарочито због потребе за скалирањем.

Основни проблем дигиталне симулације везан је за нумеричку интеграцију.

Оно што нас посебно интересује су интеграционе методе које се код симулације континуалних система могу "убацити" као интеграциони блокови.

Интеграционе методе такве врсте могу се класификовати у 2 групе:

* једнокорачне интеграционе методе и
* вишекорачне интеграционе методе.

Свака таква метода је једна апроксимација стварне интеграције дискретном алгебарском шемом, те је стога најважније питање, питање тачности и стабилности изабране методе.

Проблем нумеричке интеграције састоји се у томе да се дефинише неки оператор *F* тако да једначина

 апроксимира на датом интервалу.

Интеграл  где се користи скраћена нотација



називамо к – корачном интеграционом формулом.

Једнокорачне интеграционе методе:



* Ојлерова (Еuler) метода



* Метода трапез



* Метода Рунге-Кута (Runge-Kutta) другог реда

Вишекорачне нумеричке методе интеграције заснивају се на замени функције неким интерполационим полиномом, у изабраним чворовима интерполације, који се потом интеграли.

Тако се формира диференцна једначина вишег реда помоћу које се вредност sn+1 траженог решења израчунава не само на основу претходне sn,већ и на основу вредности sn-1,sn-2,...итд.

Општа формула за вишекорачну методу интеграције има облик:



**Одређивање интервала интеграције**

Процес симулације на дигиталном рачунару је дискретан процес. Из овога произилази да функције независне променљиве *t*, треба да се представе низом дискретних бројева у тачкама . Коначна, ненулта вредност назива се корак интеграције.

*Представљање континуалне f-је у дигиталном рачунару*

* дискретизована функција (по времену и по нивоима *ε)*
* ε-најмања вредност са којом може да се престави промена
* корак интеграције.



Функција интеграције (INT), мора се приказати у складу са правилима модела са дискретним временом.

Познато је да важи:



Ако је (*T* или *h*) довољно мало, тада приближно важи



Што је мањи интервал , то је тачност апроксимације већа, али је и време рада рачунара дуже.

Циљ је да се одабере такав интервал интеграције, да се с једне стране што је могуће више смањи грешка рачунања, а да се с друге стране време израчунавања интеграла не повећа превише.

Могући начин за одређивање интервала интеграције:

* + - дискретна лапласова трансформација
    - експеримент

При великим интервалима интеграције губи се увид о понашању система за време интервала интеграције.

При сувише малим интервалима интеграције може доћи до нагомилавања грешки услед великог броја рачунских операција и заокруживања.

**Извођење имплицитних операција**

Ако математички модел симулације континуалног система није уредљив, онда он садржи једну или више функција у којој је вредност функције аргумент исте функције:



или мање уочљив случај код скупа једначина:



Алгебарском петљомназивамо случај када су један или више алгебарских блокова повезани у затворену петљу која не садржи меморијске елементе типа интегратора или јединичног кашњења.

Врло често алгебарска петљасе може отклонити алгебарским манипулацијама на оригиналном систему једначина. Тиме се добија тзв. канонички облик система.

Међутим, има случајева када се систем не може довести у канонички облик и не може се извести симулација.

У неким системима за симулацију, међу којима је и CSMP, наведени проблем се решава итеративним решавањем једначине.

У ту сврху убацујемо додатни блок за имплицитне елементе (IMP) у петљу и добијамо случај који се може описати као:



Овај метод се употребљава у случају када нема других начина за решавање имплицитно задатих операција.

Недостатак овог метода је знатно продужавање времена извршења сваког интервала интеграције.

**Рачунарска реализација симулатора континуалних система**

Сваки систем за дигиталну симулацију континуалних процеса састоји се од:

* симулационог језика,
* процесора и
* скупа функција тј. блокова.

Симулациони језикслужи за опис: математичких модела, параметара и контролних команди и то у облику који је разумљив како за корисника тако и за рачунар.

Процесор је програм који извршава следеће задатке:

1. учитава, преводи и меморише математички модел
2. сортира математички модел
3. учитава, преводи и меморише параметре и управљачке команде
4. извршава симулацију математичког модела.

****

**Симулациони језик**

При спецификацији симулационих језика треба поштовати следећа правила:

* Формат улазних података треба да буде што флексибилнији
* Избегавање непотребних ограничења.
* За идентификацију блокова и веза између њих треба користити симболичка имена, пре него бројеве.
* Језик треба да буде такав да се лако учи, чак кад је у питању и неискусан програмер.

**Процесор**

Процесорје рачунарски програм. Састоји се из више модула. Први модул преводи опис модела у интерни језик и назива се преводилац. Други модул процесора служи за извршавање симулације и назива се симулатор. Контролу тока симулације врши модул који се назива монитор.Интерпретер имонитор су модули који се не разликују много од сличних модула у другим програмским системима, што значи да нису специфични за један симулациони систем. Много специфичнији је симулатор***.***

**7. Симулација дискретних догађаја**

Симулација дискретних догађаја је метода симулационог моделирања система код којих се дискретне промене стања у систему или његовом окружењу догађају дисконтинуално у времену.

Типичан пример система са дискретним догађајима је банка, променљиве стања, као што је БРОЈ КЛИЈЕНАТА у банци као и стања на њиховим рачунима, мењају искључиво у одређеним тренуцима времена, када клијент уђе у банку, изврши неку трансакцију и изађе.

Самопослуга, телефонска централа, болница, сервис за оправку аутомобила су примери система са дискретним догађајима.

Сервис за оправку аутомобила.

Дискретни догађаји су:

* долазак аутомобила у сервис,
* радник прихвата аутомобил и почиње поправку,
* оправљени аутомобил напушта сервис, итд.

Овакви догађаји представљају промене променљивих система које се дешавају у дискретним тренуцима времена.

Долазак возача у сервис је дискретан догађај који увећава за један број аутомобила у сервису, док одлазак возача након поправке представља дискретан догађај који тај број умањује за један.

Симулација дискретних догађаја углавном се користи за анализу динамичких система са стохастичким карактеристикама.

С обзиром да се под догађајем подразумева дискретна промена стања ентитета система (тј. променљиве која описује то стање), догађајнаступа у одређеном тренутку времена.

Типични примери система са дискретним догађајима су банка, болница, телефонска централа, сервис за оправку аутомобила, али и други системи масовног опслуживања.

Како су величине у оваквим системима најчешће случајне природе, говоримо о дискретним стохастичкимсистемима. За моделара система варијације променљивих стања не могу се предвидети, оне су случајне величине. За њихово описивање могу се користити различити статистички модели, тј. расподеле вероватноћа које одговарају датој појави.

**Формални модел СДД**



Формални опис система дискретних догађаја може се представити уређеном шесторком:

где су:

- скуп екстерних догађаја



- скуп секвенцијалних стања дискретних догађаја



- скуп излаза



- квази-преносна функција

- излазна функција



- функција наступања времена



**Догађај, активност и процес**

Код модела система са дискретним догађајима, поред концепата који описују структуру, уведени су и концепти за опис динамике. То су: догађај, активност и процес.

***Догађај*** представља дискретну промену стања ентитета у систему или његовом окружењу. Између два узастопна догађаја стање система се не мења.

Он може наступити због:

* уласка/изласка привременог ентитета у односно из система (долазак новог клијента, одлазак клијента),
* промене атрибута појединих објеката система (опслужилац у систему постаје слободан или заузет).

Догађаји могу бити:

УСЛОВНИ ДОГАЂАЈИ могу да наступе тек када је испуњен услов њиховог наступања. Обично су повезани са заузимањем неког ресурса, односно почетком активности.

БЕЗУСЛОВНИ ДОГАЂАЈИ су они чији је једини услов да текуће време симулације буде једнако времену његовог наступања. То је, по правилу, планирано време завршетка неке активности. Обично су повезани са ослобађањем ресурса који су били активирани у току трајања активности.

***Активности*** је скуп догађаја који мењају стање једног или више ентитета. Трајање активности се може унапред дефинисати, али може да зависи и од испуњења неких услова у моделу, тако да је време завршетка такве активности непознато.

***Процес*** је низ узастопних, логички повезаних догађаја кроз које пролази неки привремени објекат. Другим речима, процес је хронолошки уређена секвенца догађаја која описује једну појаву од настајања до терминирања. Процес може да обухвата део или целокупан "живот" привременог ентитета.

**Однос догађаја, активности и процеса може се графички представити као на слици:**

Пример- Сервис за оправку аутомобила

Догађаји:

* долазак возача (аутомобила) у сервис,
* заузимање канала и отпочињање поправке,
* завршетак оправке аутомобила,
* почетак прања аутомобила након поправке,
* завршетак прања,
* одлазак возача (аутомобила) из сервиса након обављене поправке и прања.

Активности:

* сервисирање (оправка) аутомобила и
* прање аутомобила (ове се активности не преклапају).

Процес би обухватио све догађаје од доласка возача у сервис до његовог одласка након завршене поправке и прања.

Користећи се претходно дефинисаним појмовима, можемо описати СИМУЛАЦИЈУ ДИСКРЕТНИХ ДОГАЂАЈА на следећи начин: **Ентитети** који се описују **атрибутима** и узајамно делују учествујући у **активностима** под одређеним **условима** стварају **догађаје** који мењају **стања система.**

**Развој симулације дискретних догађаја**

Кључни елементи развоја симулационих програма у симулацији система са дискретним догађајима су:

* МЕХАНИЗАМ ПОМАКА ВРЕМЕНА
* ПРИСТУПИ ГЕНЕРИСАЊУ ДОГАЂАЈА

**Механизам помака времена**

У симулацији система са дискретним догађајима користе се два основна механизма помака времена:

***1. Помак времена за константни прираштај***

Код ове стратегије, време у симулационом моделу се мења тако да се увек додаје константни прираштај. Након сваког помака времена испитује се да ли је у претходном интервалу времена требало да дође до наступања неких догађаја. Уколико јесте, тада се ти догађаји планирају за крај интервала.

НЕДОСТАЦИ: Померањем догађаја на крај временског интервала у којем би они требало да наступе, уводи се грешка у извођење симулације.

Догађаји се претстављају као истовремени што у стварности није случај. Тако да је потребно дефинисати редослед извршења догађаја.

Смањењем временског прираста те се грешке смањују, али се зато повећава време које се троши на извођење симулације. Овај механизам помака времена може се успешно применити у ситуацијама када се догађаји заиста нижу један по један у константним временским интервалима. На пример у економским системима где се стања мењају у константним временским периодима.

Приступ је приказан на слици:

***2. Помак времена на наредни догађај***

Код овог приступа, симулациони сат се помера на време у којем ће наступити први наредни догађај.

У том тренутку се догађај изведе и направи се одговарајућа промена стања система; затим се поново испитује који ће догађај следећи наступити, итд.

**Генерисање догађаја**

У рачунарској имплементацији симулационог процеса, догађај се описује са више атрибута, који формирају СЛОГДОГАЂАЈА.

С обзиром на променљив број догађаја у времену, слогови догађаја се меморишу у ЛИСТАМА ДОГАЂАЈА.

**Приступи генерисању догађаја**

Дефинисање догађаја унапред. Код овог приступа сви догађаји су унапред познати и дефинисани, а листа догађаја садржи слогове свих догађаја.

Приступ заснован на наредном догађају. Код овог приступа, познат је једино први наредни догађај, а листа догађаја садржи само један слог, слог познатог догађаја.

Догађаје можемо сврстати у две основне категорије у односу на место настанка (генерисања):

*Екстерни* догађајису они догађаји који не зависе од модела и представљају утицај околине на систем.

* долазак возача у ауто-сервис,
* појаву телефонског позива,
* долазак купца у самопослугу, итд.

*Интерни* догађајизависе од модела и у њему се генеришу.

* долазак купца у ред за касу,
* прекидање везе по завршетку разговора,
* завршетак операције болесника, итд.

Листа догађаја у тренутку *t = t1* , пре генерисања интерних догађаја *i2* , *i4* и *i6* :

Листа догађаја у тренутку *t = t2* , након генерисања интерних догађаја:

**Стратегије извођења симулације**

Концепти као што су догађај, активност и процес чине полазну основу за дефинисање стратегија извођења симулације система са дискретним догађајима.

Те стратегије су:

* РАСПОРЕЂИВАЊЕ ДОГАЂАЈА
* СКАНИРАЊЕ АКТИВНОСТИ
* ИНТЕРАКЦИЈА ПРОЦЕСА

**Распоређивање догађаја**

Механизам распоређивања догађаја подразумева да се догађаји планирају унапред и држе у листи будућих догађаја.

Процедура планирања догађаја је следећа:

1. Генерише се слог догађаја.
2. Затим се доделе вредности његових атрибута.
3. Затим се догађај ставља у листу будућих догађаја.

Функционисање симулатора се одвија на следећи начин:

* са листе будућих догађаја узима се први догађај.
* Ажурира се симулациони сат на време његовог наступања.
* Када се изврше сви догађаји који имају исто време наступања, симулациони сат се ажурира на време следећег догађаја, из листе будућих догађаја.

***Дијаграм тока распоређивања догађаја***

**Сканирање активности**

Активност смо дефинисали као скуп операција које трансформишу стања ентитета.

Да би се извршила симулација, потребно је реализовати рачунарски програм који сканира и распоређује активности модела. Сканирање активности подразумева да се догађаји имплицитно распоређују тако да се промена стања извршава преко функција које се називају активности.

Свака активност има услови акцију*.*

За сваки временски корак, активности се сканирају и тражи се прва активност која има задовољен услов.

Тада се извршава одговарајући програмски сегмент који специфицира акцију за задату активност.

Процес сканирања наставља се све дотле док све активности не буду блокиране.

Онда и само онда се симулациони сат ажурира за следећи временски корак.

За извршење активности у реалном систему потребно је извесно време, док при симулацији није потребно пратити кретање активности од почетка до завршетка.

Активности које у реалном свету трају извесно време, при симулацији не троше рачунарско време јер се не дефинишу експлицитно, док прелази са активности на активност који се у реалном свету одвијају тренутно, троше извесно рачунарско време јер се описују експлицитно одређеним секцијама програма.

**Интеракција процеса**

Интеракција процеса представља технику симулације која је настала комбинацијом распоређивања догађаја и сканирања активности.

Процес можемо посматрати као скуп узајамно искључивих активности, повезаних тако да терминирање једне активности дозвољава иницијализацију неке друге активности из скупа активности процеса.

У овом прилазу, могући конфликт који проистиче из преклапања процеса, решава се увођењем две наредбе које се употребљавају при спецификацији догађаја.

То су:

* WAIT
* DELАY

и то у оба контекста, условном и безусловном.

WAIТ наредба у условном контексту има облик: WАIТ UNTIL <услов>. Уколико је услов задовољен, процес који чека на тај услов, наставиће свој ток. У супротном, биће блокиран. У том случају испитује се који од осталих процеса може да се активира и он се извршава.

Наредбом DELAY која се у безусловном контексту у литератури означава и са:

ADVANCE <број временских јединица> наставак процеса одлаже се за специфициран број временских јединица. У том случају, испитује се који од осталих процеса може да оствари свој ток. На тај начин се у ову методологију преносе добре особине методологије распоређивања догађаја.

***Алгоритам рада процесора при интеракцији процеса*:**

У оквиру процесора, одржавају се две листе догађаја:

1. Листа текућих догађаја:

* садржи условне догађаје;
* сортирана је по приоритету.

2. Листа будућих догађаја:

* садржи распоређене догађаје који тек треба да наступе;
* сортирана је по времену наступања.

Планирање догађаја процесор врши по стратегији наредног догађаја. При сваком премештању догађаја, из листе будућих догађаја у листу текућих догађаја, планира се његов наследник који се ставља у листу будућих догађаја, да тамо чека своје време наступања. Пре почетка симулације, за сваки процес генерише се први догађај и ставља се у листу будућих догађаја. Затим се време симулације ажурира на време догађаја са врха листе будућих догађаја.

Рад процесора одвија се у две фазе:

1. Фаза ажурирања времена симулације:

* симулациони сат се поставља на вредност времена наступања првог догађаја из листе будућих догађаја,
* затим се пребаце сви догађаји са тим временом наступања у листу текућих догађаја.
* за сваки премештен догађај се планира наследник, и ставља се у листу будућих догађаја.

2. Фаза сканирања листе текућих догађаја:

* сканира се листа текућих догађаја и уколико има неблокираних догађаја, они се извршавају.
* Када нема догађаја у листи текућих догађаја, или су сви догађаји блокирани, тада се преноси први догађај из листе будућих догађаја и ажурира се сат симулације на његово време.