

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE

Fakulta chemickej a potravinárskej technológie

Katedra informatizácie a riadenia procesov

DIPLOMOVÁ PRÁCA

Bratislava 2004

Vypracovala: Bc. Tatiana Lipničanová
Vedúci projektu: Doc.Dr.Ing. Miroslav Fikar
Konzultant: Prof. Dr.-Ing. Christian Schmid

Ďakujem Doc. Dr. Ing. Miroslavovi Fikarovi a Prof. Dr.-Ing. Christianovi Schmidovi za odborné vedenie, cenné pripomienky a pomoc pri vypracovaní projektu ako i Dipl.-Ing. Jorgovi Uhligovi a Dipl.-Ing. Christianovi Falkenbergovi, ktorí mi pomohli riešiť mnohé problémy pri práci na animácii.

ABSTRACT

Animated displays are becoming widely used as a means of teaching about dynamic subject matter. With the advance of Internet technology in recent years, various computer-based learning courses employing animation techniques have been developed to assist learning. This reflects the reality that animations have been very easy and effective to produce. But we should use it with caution, because if poorly designed or applied in learning materials, it may create more learning problems than it solves.

University lecturers have taken up the opportunity to include animations in their teaching materials with the expectation that these presentations will make the content easier to understand for their students and enhance learning. The current generation computers give them facilities to produce their own animations that have never been available before to individual lecturers. Animation is a viable tool for learning, gaining insight, checking the behavior, or searching for failure models.

The age-old adage says “a picture speaks more than thousand words” and the adage is in particular symptomatic in the case of picturing dynamic subject matter.

Animations are a highly motivating form of presentation with the capacity to attract and sustain our attention. Dynamic visualization depicts change processes explicitly rather than static visualizations of the same content. When students are given static pictures that are meant to represent dynamic subject matter, they are faced with the cognitive burden of having to “mentally animate” the content. However, well-designed animations can eliminate this burden so that learner thinking processes are freed up and can be devoted to educationally worthwhile activity such as the development of understanding. In other words, the use of dynamic rather than static visualizations should free up cognitive capacity for comprehension of the process that is more central to the learning task.

The students of the Faculty of Chemical and Food Technology of the Slovak University of Technology in Bratislava in their sixth semester learn fundamentals of modeling, simulation, and process control and they work with MATLAB program during their exercises. The people from Department of Information Engineering and Process Control make every effort to make learning materials more interesting and make the main content easier to understand for students.

The creating of animated system of two tanks in MATLAB is one of these activities.

PREDSLOV

V posledných rokoch sa neoddeliteľnou súčasťou výučby stáva používanie moderných technológií, čo vedie k rapídnemu rozvoju elektronických vyučovacích pomôcky. Vo všeobecnosti ilustrácie zohrávajú významnú úlohu v priblížení a uľahčení pochopenia problematiky, ktorá je predmetom štúdia.

Animácia sa stala hodnotným nástrojom vyučovacích materiálov priamym zobrazovaním dynamickej podstaty obsahu. V porovnaní so statickými obrázkami sa už na prvý pohľad javí ako efektívnejší a mocnejší nástroj ponúkajúci množstvo možností, ktoré by mali viesť k skvalitneniu a k zatraktívneniu výučby.

Študovaná problematika na univerzitách je niekedy natoľko náročná na predstavivosť, že študenti vynakladajú množstvo úsilia, ktoré by inak mohli požiť na zaradenie študovaného poznatku do širších súvislostí. Pretože pochopenie sprostredkovanej informácie a jej začlenenie do širšieho kontextu vedomostí je podstatnejšie ako samotné interpretovanie opisovanej a statickou grafikou zobrazovanej informácie.

Animácia nemá nahradiť žiadny učebný materiál, alebo učiacich, ale mala by sa stať rovnocennou súčasťou napomáhajúcou študentom dosahovať lepšie študijné výsledky.

Autorka

OBSAH

1	Úvod.....	8
1.1	Model ľudského uvažovania a spracovania informácií.....	8
1.2	Pohyb a ľudská schopnosť vnemového /zrakového/ rozlišovania.....	9
1.3	Percepčná pozornosť.....	10
2	Vizualizácia.....	11
2.1	Animácia.....	12
2.1.1	Animácia a jej uplatnenie vo vyučovacom procese.....	13
2.2	Animované a statické obrázky.....	14
2.3	Od statickej k animovanej prezentácii.....	15
2.4	Tvorba animácie.....	16
3	Modelovanie, simulácia a animácia.....	19
3.1	Matematické modelovanie procesov.....	19
3.2	Elementy modelu.....	20
3.2.1	Premenné.....	21
3.2.2	Rovnice.....	22
3.2.3	Parametre.....	23
3.3	Linearizácia.....	24
3.4	S-funkcia.....	25
3.4.1	M-súbor S-funkcie.....	25
3.4.2	Definovanie bloku S-funkcie.....	26
3.4.3	Jednoduché príklady S-funkcie.....	27
3.5	Modelovanie a simulácia v procese vzdelávania.....	27
3.6	Vizualizácia výsledkov modelovania a simulácie.....	28
4	Grafika v MATLABE.....	29
4.1	XY graf.....	30
4.2	XYZ graf.....	31
4.3	Film.....	34
4.4	Animácia.....	34
4.5	Grafické objekty.....	37
4.5.1	Vlastnosti objektov.....	40
4.5.2	Object Handles.....	40
4.5.3	Hľadanie objektov funkciou <i>findobj</i>	41
5	Systém zásobníkov kvapaliny.....	42
5.1	Definovanie animovaného systému.....	42
5.2	Matematické modely.....	42
5.2.1	Dva zásobníky kvapaliny s interakciou.....	43
5.2.2	Dva zásobníky kvapaliny bez interakcie.....	44
5.3	Animácia systému zásobníkov kvapaliny.....	45
5.3.1	Animácia ako doplňujúci vizualizačný prostriedok simulačnej schémy.....	46
5.3.2	Animácia ako súčasť GUI okna.....	51
6	Záver.....	54
7	Literatúra.....	55

ZOZNAM ILUSTÁCIÍ A TABULIEK

Obrázok 2.1: Statické zobrazenia zásobníkov kvapaliny	15
Obrázok 4.1: Hierarchia grafických objektov	37
Obrázok 4.2: Grafické objekty	38
Obrázok 4.3: Príklad využitia funkcie <i>findobj</i> v programe <i>tanksnew.m</i>	41
Obrázok 5.1: Blok parametrov	46
Obrázok 5.2: Štruktúra subsystému <i>Zásobníky</i>	47
Obrázok 5.3: Simulačná schéma <i>hs2sanim.mdl</i>	48
Obrázok 5.4: Simulačná schéma <i>hs2riadan.mdl</i>	49
Obrázok 5.5: Vizualizácia dát získaných simuláciou pomocou grafov a animácie	49
Obrázok 5.6: GUI okno	52
Obrázok 5.7: Štruktúra subsystému <i>Zásobníky</i>	53
Tabuľka 3-1: Rutiny M-súboru S-funkcie	26
Tabuľka 3-2: Oblasti štruktúry rozmerov	27
Tabuľka 4-1: Znaky definujúce typ čiary	30
Tabuľka 5-1: Zoznam súborov	51

ZOZNAM SKRATIEK A SYMBOLOV

\dot{m}_0	- Vstupný hmotnostný prietok kvapaliny do ľavého zásobníka
\dot{m}_1	- Vstupný hmotnostný prietok kvapaliny do pravého zásobníka
\dot{m}_2	- Výstupný hmotnostný prietok kvapaliny z pravého zásobníka
\dot{q}_0	- Vstupný objemový prietok kvapaliny do ľavého zásobníka
\dot{q}_1	- Vstupný objemový prietok kvapaliny do pravého zásobníka
\dot{q}_2	- Výstupný objemový prietok kvapaliny z pravého zásobníka
φ_i	- Hmotnostná koncentrácia kvapaliny v i-tom zásobníku
k_{ii}	- Konštanta odporu ventilu i-tého zásobníka
h_i	- Výška hladiny kvapaliny v i-tom zásobníku
F_i	- Plocha povrchu hladiny kvapaliny v i-tom zásobníku - Plocha i-tého zásobníka
U_i	- Odchýlková veličina vstupnej veličiny i-tého zásobníka
X_i	- Odchýlková veličina stavovej veličiny i-tého zásobníka
K_i	- Konštanta vyjadrujúca závislosť odporu ventilu od výšky hladiny kvapaliny v i-tom zásobníku

1 Úvod

1.1 Model ľudského uvažovania a spracovania informácií

Aby sme mohli definovať efektívne spôsoby návrhu animácie, je potrebné najprv uviesť model ľudského uvažovania a spracovávaní informácií.

Odhaduje sa, že celkové množstvo informácií ktoré vstupujú do nášho tela je okolo 1 bilión bitov za sekundu, z ktorých len okolo 100 bitov za sekundu sme schopní spracovať. Mozog sa vždy snaží redukovať množstvo spracovaných informácií. Ak je príliš veľa informácií prezentovaných v rovnakom čase, potom sa naša pozornosť sústreďí len na ich určitú časť. Mozog je schopný filtrovať obrazy, ale aj zvuky aj hluk na pozadí [5].

Psychológovia sa dlho zaoberali týmto problémom spracovania informácií a navrhli niekoľko modelov. V kognitívnom procese budeme rozlišovať ukladanie vnemov krátkodobej pamäti, dlhodobej pamäti, plánovania a prevedenia odpovedajúceho zásahu.

Človek sa môže koncentrovať len na jednu vec v danom čase, ale môže veľmi ľahko upriamiť pozornosť z jednej veci na druhú. Z hľadiska animácie sú najdôležitejšie vizuálne a akustické vnemy.

Informácie zberané zmyslovými orgánmi sú prenášané do krátkodobej pamäti, kde môžeme vedome dávať na nich pozor. S krátkodobej pamäti sa informácie, v mnohých prípadoch len vlastným úsilím, prenášajú do dlhodobej pamäti. Krátkodobá pamäť je veľmi rýchla na pripomenutie, ale aj na zabúdanie. Je to naše vedomie. Udržiava potrebné informácie v čase premýšľania a poskytuje nám základňu pre prevedenie potrebných akcií. Pripamätanie a ukladanie do dlhodobej pamäte trvá dlhšie. Informácie krátkodobej pamäte sa môžu vybaviť za niekoľko sekúnd, ale v dlhodobej pamäti to môže trvať aj celý život [5].

V krátkodobej pamäti je miesto pre 7+2 informačných položiek. Nové prichádzajúce informácie vymažú už existujúce, ktoré ešte neboli spracované, alebo presunuté do dlhodobej pamäte. Podstatné je, že položky v krátkodobej pamäti sú na rovnakej

abstrakčnej úrovni (ak rozmýšľame o futbalovom skóre, nie je problém porovnať ho v rovnakom čase s inými futbalovými skóre, ale nemôžeme rozmýšľať o futbale a súčasne plánovať dovolenku) [5].

Ukladanie nových informácií do dlhodobej pamäte je ľahšie, ak sa tieto spájajú s informáciami, ktoré už existujú a sú uložené. Takisto si ľahšie zapamätáme určité fakty, ak nie sú prezentované samostatne, ale v určitých príčinných vzťahoch.

Ľudská pamäť nepracuje s priamo adresovateľnými bunkami ako počítač, ale pracuje na báze analógií a asociácií.

Výskumom bolo zistené že:

- zobrazenia sú nadradené nad slovami z hľadiska dlhodobej pamäte,
- zobrazenia v kombinácii s textom uľahčujú učenie, za predpokladu že zobrazenie súvisí s textom,
- deti mladšie ako 9 rokov čerpajú viac informácií zo zobrazení ako starší žiaci,
- obrazotvornosť sa vyvíja postupne a deti mladšie ako 9 rokov majú obmedzené schopnosti vytvárať predstavu pri čítaní textu.

1.2 Pohyb a ľudská schopnosť vnemového /zrakového/ rozlišovania

Vnímanie pohybu v prípade animácie je známe ako fenomén “zdanlivého pohybu”. Ľudská duševná schopnosť má tendenciu vnímať sériu “diskrétnych a separovaných” vstupov ako trvajúci pohyb.

Zdanlivý pohyb objektu nastáva pri zobrazovaní jeho malých prírastkových zmien polohy, vzájomne sa veľmi neodlišujúcich, v určitom časovom úseku. Minimálna frekvencia rýchlosti zobrazovania by mala byť 24 obrázkov za sekundu. Táto hodnota je odvodená od vlastností ľudského oka a pri tejto frekvencii už ľudské oko neregistruje prechody medzi jednotlivými statickými snímkami. Mieru a hladkosť vnímaného pohybu určujú faktory ako trvanie zobrazenia a synchronizácia, priestorová blízkosť, celková intenzita osvetlenia a predchádzajúce znalosti [7].

Táto optická ilúzia je rovnako vytváraná filmom, videom, televíziou a kreslenou animáciou.

1.3 Percepčná pozornosť

Percepčná pozornosť má vysokú selektivitu. Naše oči zohrávajú kľúčovú úlohu vo vytváraní vnútornej mapy vonkajšieho sveta. Pohybujúci objekt má okamžitú percepčnú nápadnosť, vyčnieva zo statického okolia, dokonca aj keď je zaregistrovaný periférnym videním. Schopnosť okamžitého spozorovania nepatrného pohybu objektu, hodnotenie smer a trajektórie pohybu je evolučnou prednosťou mnohých zvierat, vrátane človeka [7].

2 Vizualizácia

Vizualizácia je vizuálna reprezentácia dát a používa sa na lepšie pochopenie a nahliadnutie do údajov. Interaktívna vizualizácia poskytuje lepšie preskúmanie dát. Napríklad zobrazenie možno otáčať a pozorovať z rôznych uhl'ov. Interaktívna vizualizácia obsahujúca animáciu zobrazuje trendy dát [3].

Na abstraktnej úrovni tento pojem znamená zviditeľňovanie. Proces vizualizácie je zameraný na ľudský objekt s cieľom vyvolať v ňom vizuálny vnem.

Niektoré softvérové vizualizačné nástroje:

- Plotting tools GnuPlot
- MATLAB
- Mathematica
- Maple
- Image Processing Tools SDSC Image Library/Tools
- Khoros
- AVS
- 3D Tools OpenGL (low-level: polygons)
- Java3D (high-level: scene graph)
- SDSC Volume Toolkit
- VRML Virtual Reality Modeling Language
- MineSet SGI product for datamining and visualisation
- Data Explorer IBM data visualisation

Vizualizácia je použitie teoretických, technických, programových a komunikačných prostriedkov pre zviditeľňovanie definovaných (abstraktných alebo reálnych) objektov [5].

Presnejšia definícia vizualizácie je podmienená oblasťou použitia, tzn. definíciou objektov a použitými prostriedkami. Napríklad pre oblasť priemyselnej automatizácie je definícia vizualizácie nasledovná:

Vizualizácia je použitie teoretických, technických, programových a komunikačných prostriedkov pre zviditeľňovanie definovaných (abstraktných, alebo reálnych) objektov na ľubovoľnej úrovni informačného a riadiaceho systému s cieľom podporiť rozhodovania [5].

Pri vizualizácii sa nejedná len o grafické zobrazovanie objektov, ale o všetky činnosti týkajúce sa definovania, získania a spracovania objektov, ktorých grafická stránka je používateľským rozhraním medzi priemyselným procesom a človekom.

2.1 Animácia

O animácii hovoríme v prípade, keď sa jednotlivé obrazy, zobrazujúce malé prírastky polohy objektov, pohybujúce sa v rýchlom slede za sebou, čím vytvárajú ilúziu pohybu. Animácia bola v minulosti využívaná najmä v počítačových hrách. Uplatnenie v oblasti multimédií je však oveľa širšie. Možnosť zachytiť pohyb určitého objektu, ktorým môže byť text alebo obrázok, je možné využiť pri opise nejakého deja alebo činnosti, čo sa veľmi často využíva vo výukových programoch. Animácia predstavuje kombinovaný typ dát. Zahrňuje obraz, grafiku, zvuk a pohyb v čase. Má mnoho podôb, od jednoduchého presunu textu z jednej časti obrazovky do druhej, až po zložité modelovanie v trojrozmernom priestore.

Animácia je teda definovaná ako sekvencia obrázkov, ktoré môžu byť kombinované zo zvukom. Pomocou týchto pohyblivých obrázkov je možné vytvoriť obdobu trikového kresleného filmu. Animácia si veľmi rýchlo osvojila oblasť vizualizácie dejov, ktoré v reálnom svete nie sú pozorovateľné. Týmto spôsobom je možné vizualizovať scény, ktoré reálne neexistujú. Základný význam má animácia v oblasti simulácie.

Pri pohľade na bežnú webovú stránku pravdepodobne uvidíme niekoľko typov animovanej grafiky kmitajúcej na monitore. Animácia sa stala veľmi jednoducho produktovateľnou a súčasné technické vybavenie umožňuje intenzívne spracovanie operácií a prenos zobrazenia rafinovanej dynamickej animácie.

Cieľom návrhu dobrej animácie je pritiahnuť pozornosť na dôležité fakty a informácie, mala by byť prehľadná, málo náročná a mala by umožniť užívateľovi sústrediť sa na proces, bez toho aby jeho pozornosť bola rozptýlená interfejsom.

2.1.1 Animácia a jej uplatnenie vo vyučovacom procese

Animácia sa čoraz viac používa vo vyučovaní prostredníctvom počítačov, uľahčujúca učenie využitím predností vizuálneho učenia. Ak sú študentom predložené len statické obrázky reprezentujúce dynamickú podstatu, stoja zoči voči poznávacej záťaži pretože musia "mentálne animovať" tento obsah. Správne navrhnutá animácia odstraňuje túto záťaž, čím sú učenie a kognitívne procesy oslobodené od tejto záťaže. Študent sa tak môže viac sústrediť na zaradenie poznatku v širšom kontexte a ďalších súvislostí.

Mnohí prednášajúci transformujú tradičné statické obrázky a diagramy z učebníc do účinných animovaných prezentácií. Tvorba animácie je teraz pomerne jednoduchá aj pre univerzitných učiteľov bez pokročilých programátorských schopností. Univerzitní prednášajúci zahŕňajú animáciu do svojich učebných materiálov s očakávaním, že táto forma prezentácie urobí obsah ľahšie zrozumiteľný a zaujímavejší pre ich študentov, čo zefektívni učenie.

Animácia je forma vzdelávania, ktorá spája tri základné prvky procesu vyučovania. Je obsahom, metódou a prostriedkom ktorými dosahujeme ciele vzdelávania. Animácia zaujala významné miesto tam, kde nepostačujú klasické vzdelávacie metódy a prostriedky. Účinná animácia je informačne bohatá a zrozumiteľná.

Autori, ktorí sú obsahovými zmlcami, majú tendenciu zameriavať sa skôr na špecifický obsah ako na charakteristiky animácie, ktoré sú ale rozhodujúce v určovaní jej vzdelávacej účinnosti. Preto by návrh animácie mal byť dôsledne uvažovaný nielen po obsahovej ale aj po grafickej stránke. Tradičný prístup k tvorbe vzdelávacích pomôcok bol po dlhú dobu postavený na statických objektoch, preto návrh animácie je náročnejší a širokospektrálnejší.

2.2 Animované a statické obrázky

Existuje množstvo dôvodov prečo uprednostniť animáciu nad statickou grafikou v oblasti vzdelávania. Tieto argumenty majú súvislosť s citovými a poznávacími charakteristikami animácie. Animácia spája audiovizuálne senzorické vnímanie a pohyb, zároveň je emocionálne podfarbená.

Animácia sa vyznačuje atraktívnosťou a zábavnosťou, úspechy animovaného filmového priemyslu to potvrdzujú. Animácia je značne sugestívnejšia forma prezentácie so schopnosťou priťahnúť a udržať našu pozornosť. Bez počiatočnej a pretrvávajúcej pozornosti na danú prezentovanú informáciu, učiaci nedostatočne zachytí sprostredkovanú informáciu. Špecifický prínos animácie, na zlepšenie efektivity učenia sa prejavuje v lepšom pochopení, zapamätaní si a vybavení si informácie [8].

Nasledovné body popisuje niektoré potenciálne výhody animovanej grafiky v porovnaní s ich statickým náprotivkom vzhľadom na kognitívne procesy v rámci učenia.

V porovnaní so statickou grafikou, animovaná grafika:

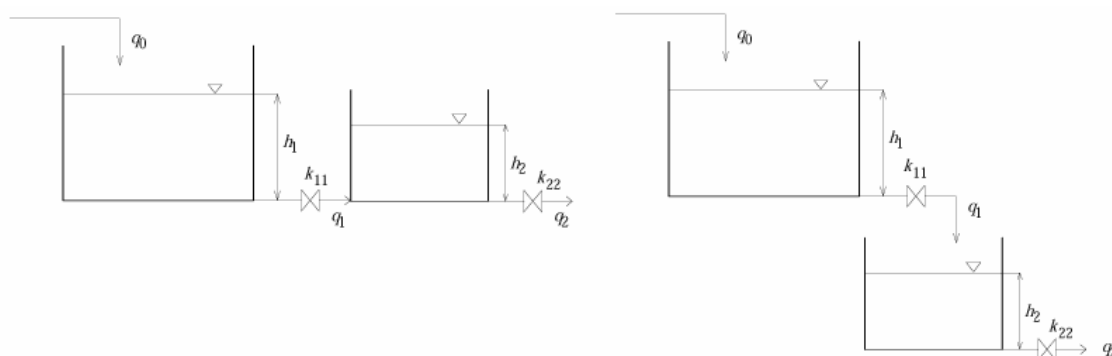
- ❖ Priamočiarejšia, pretože dáva lepší súvis medzi animovanými časťami a ich dynamickými vzťahmi, čo umožňuje preniknúť hlbšie do podstaty problému.
- ❖ Viac jasná, pretože každý dynamický aspekt je divákovi okamžite zobrazený skôr ako predstava o hybnosti, ktorá je vytvorená z nedynamického zobrazenia.
- ❖ Dochádza k menšiemu počtu omylov a nesprávnych výkladov.
- ❖ Viac informatívna, pretože ponúka viacmyslovú audiovizuálnu možnosť vnímania podstaty a tým aj väčšiu možnosť spätnej väzby.
- ❖ Zrozumiteľnejšia, pretože na rozdiel od statickej grafiky nevyžadujú rôzne pomocné symboly (šípky, bodkované čiary, atď.) ktoré majú nepriamo predstaviť dynamické hľadisko obsahu. Toto znamená, že zobrazenie je menej chaotické a učiaci nedešifruje procesy, ktoré tieto symboly vysvetľujú.

Je isté, že toto sú len možné výhody, ktoré automaticky nedosiahneme len preto, že konvertujeme statické zobrazenie do niektorého typu animácie. Inak povedané, animované

obrázky nie sú v každom prípade nadradené nad statickou grafikou. Bez dobrého návrhu, je možné produkovať animácie, ktoré sú horšie ako ich statický náprotivok [8].

2.3 Od statickej k animovanej prezentácii

Najčastejším štartovacím bodom na vytvorenie animácie je niektorý typ statickej grafiky (napr. obrázok z učebnice).



Obrázok 2.1: Statické zobrazenia zásobníkov kvapaliny

Za účelom zobraziť skutočnú dynamickú podstatu zobrazenia, musí statické zobrazenie použiť niekoľko metód, ktoré sú pridané k schéme za účelom usmerniť obsah rôznym spôsobom.

V oboch prípadoch zásobníkov kvapaliny (obrázok 2.1) sú použité šípky, ktoré majú naznačiť smer toku tekutiny. Výplň zásobníkov má naznačovať aktuálnu výšku kvapaliny.

Použitie zobrazenia s textom

Animácia bez vysvetlenia môže mať v podstate rovnaký efekt na pochopenie učeného obsahu ako žiadna výučba. Ale optimálnou kombináciou zobrazenia (animácia, statický obrázok) a textu môžeme dosiahnuť väčšieho efektu ako len ústnym opisom. Animácia, alebo textový materiál nie sú samostatne natoľko výkladové, ako keď sa navzájom dopĺňajú.

2.4 Tvorba animácie

Animačné okno by malo byť jednoduché a nemalo by obsahovať nepotrebné informácie. Tie môžu priťahovať pozornosť užívateľa, ktorú by mal venovať dôležitejším informáciám. Rovnako informačný obsah animácie má byť zameraný na používateľa, pretože expert v danej oblasti sa nezaujíma o informácie, ktoré však môžu byť rozhodujúce pre nováčika.

Dobre navrhnutá animácia by mala mať nasledovné charakteristiky:

- zodpovedať účelu, tzn. že neobsahuje viac informácií ako je potrebné,
- byť maximálne názorná a zreteľná,
- byť zhodná s ostatnými používanými materiálmi, tzn. že používa rovnaké symboly.

Kroky v navrhovaní animácie

Použijeme nasledovné kroky k prevodu od statického k animovanému zobrazeniu systému zásobníkov.

1) Analýza dynamických stavov a ich udalostí

Identifikujeme všetky podstatné entity, ich vlastnosti, ktoré sú súčasťou mechanizmu a kľúčové vzťahy existujúce medzi nimi. Vlastnosti rôznych entít a jednotlivé vzťahy sa menia v čase.

2) Výber entít, vzťahov a vlastností, ktoré budeme animovať

Výberom urobíme zjednodušenia, ktoré budú kompromisom medzi realistickým zobrazením a jednoduchosťou realizácie animácie.

Použijeme čo najmenšie množstvo grafického materiálu s maximálnou hodnotou výkladového efektu. Významnou mierou tak redukuje stupeň vizuálneho zmätku a rozptýlenia od zobrazovanej podstaty.

3) Určiť postupnosť hlavných udalostí

Môže sa zdať, že najlepší spôsob je usporiadať obsah prezentácie podľa prirodzenej následnosti udalostí, ktoré prebiehajú v zobrazovanej situácii. Uprednostniť túžbu po realizme pred všetkými ostatnými kritériami je ale neospravedlniteľné. Z hľadiska vzdelávania zoradíme postupnosť dejov tak, aby bolo pochopenie čo najjednoduchšie a zobrazenie čo najúčelnejšie.

V našom prípade zásobníkov kvapaliny animácia môže začínať z určitého nenulového ustáleného stavu, bez animácie predchádzajúceho priebehu, akým sa systém do neho dostal.

4) Obsahové a časové usporiadanie čiastkových dejov

V reálnom svete, čas plynie stále a spojitý. Toto môže byť jeden z dôvodov, prečo udalostiam, ktoré sa odohrajú v krátkom čase môže byť ťažko porozumieť "ako sa stanú". Stávajú sa viac zrozumiteľné, ak je daná možnosť popremýšľať o nich a analyzovať ich kauzalitu. Často potrebujeme niekoho, kto nám nakreslí daný problém a tak nás upozorní na hlavné aspekty podstaty vecí. Je tiež osožné, ak to čo vidíme, je spojené s niečím čo už poznáme. Interpretácia toho čo vidíme, je vo väčšine prípadov zjednodušená, ak si môžeme pomôcť rozložením súvislého toku deja do niekoľkých diskretných krokov. Ďalším faktorom, ktorý treba zväziť vzhľadom na časovú štruktúru, je množstvo informácií prezentovaných za jednotku času. Statické zobrazenia majú málo alternatív a prezentujú všetky ich informácie naraz. Ale animácia môže stupňovať prezentáciu tak, aby informácia bola vytváraná postupne v priebehu doby animácie, aby poznávací obsah bol vždy v rozumných hraniciach pochopenia.

5) Naznačiť rozhodujúcu informáciu

Statické zobrazenie použitím symbolov a rôznymi druhmi znakov naznačuje dej, alebo priebeh zmien, ktorý v skutočnosti trvá určitý časový úsek. Animácia zobrazuje tieto zmeny postupne až do konečnej podoby.

Nemožno tvrdiť že tieto odporúčané kroky, ktoré sú vyššie uvedené, sú všetky. Sú len určené na ponúknuť systematického prístupu k návrhu vzdelávacích animácií. Je veľa ďalších podstatných hľadísk spojených s návrhom vzdelávacích materiálov, ktoré sú známe skúseným tvorcom učebných prezentácií, ktoré tu ale nie sú spomínané [8].

Faktory ovplyvňujúce účinnosť animácie

- *vytýčenie cieľov animácie,*
- *správny výber dynamického aspektu a spôsobu zobrazenia,*

Animácia má zobrazovať len dynamické aspekty, ktoré sú predmetom učenia spôsobom, ktorý je zreteľný, názorný a výkladový.

- *úroveň predchádzajúcich vedomostí z danej oblasti,*

Pre nováčika môže byť zobrazenie detailov, ktoré sú pre odborníka z danej oblasti zbytočné, pre jeho pochopenie veľmi dôležité až rozhodujúce.

- *vek užívateľov a úroveň abstraktnej predstavivosti,*

Dospelí nie sú tak odkázaní na zobrazenia počas učenia tak, ako malé deti. Dospelí dokážu ľahšie vo svojej predstave vytvoriť abstraktný model opísaný čítaným textom, to však neznamená že zobrazenia a najmä animácia nie je prínosom. Univerzitný učebný obsah je v niektorých prípadoch natoľko náročný, že vyžaduje dobrú abstraktnú predstavivosť ktorou nie každý dospelý je vybavený.

- *dobrý animačný návrh berie do úvahy možnosti ľudského spracovania informácií.*

Proces vytvárania animácie, pozostáva z nasledujúcich krokov:

- vytvorenie obrázkov, respektívne grafiky (digitalizovaním alebo pomocou grafického editora) a pozadia,
- dodať pohyb jednotlivým objektom (zvoliť techniku),
- vyfarbením objektov za účelom získania realistického vzhľadu,
- synchronizácia pohybu so zvukom,
- uloženie animácie do súboru,
- prehrávanie animácie v aplikácii.

3 Modelovanie, simulácia a animácia

Animačné projekty začínajú predstavou, ako by mali konečné animácie vyzerat'. V ďalších krokoch nastupuje vedecká a inžinierska tvorivosť. Každá predstava zahŕňa nejaký reálny fyzikálny systém. Na to, aby sme dosiahli fyzikálne realistickú animáciu musíme systém pretransformovať do súboru zložiek, ktoré potom jednotlivo analyzujeme zákonmi fyziky. Tento proces sa nazýva "modelovanie".

Matematický model je nevyhnutne jednoduchší než reálny systém. Pri získavaní musíme mať na zreteli cieľ, pre ktorý matematický model použijeme a voliť rozumný kompromis medzi presnosťou modelu a jeho jednoduchosťou. To akým spôsobom zjednodušíme model reálneho systému, musí byť dôsledne uvážené a premyslené. Pretože všetko, čo pri odvádzaní modelu zanedbáme, má iste určitý vplyv na celok. Zanedbanie efektu časti systému prichádza v úvahu len v prípade rozhodnutia, že vplyv nie je rozhodujúci a významný vzhľadom na účel použitia modelu.

V prípade, kedy potrebujeme animáciou čo najreálnejšie zobrazit' vlastnosti systému, vytvoríme model, ktorý najlepšie vystihuje fyzikálnu a chemickú podstatu procesu. Takýto model je zvyčajne komplikovaný.

3.1 Matematické modelovanie procesov

Každý proces je určený svojou fyzikálno-chemickou podstatou, ktorá vyjadruje materiálovo-energetické väzby. Podľa charakteru týchto väzieb sa všetky procesy chemickej technológie rozdeľujú na hydraulické, tepelné, procesy prestupu tepla (difúzne), chemické (s reakciou) a mechanické [1].

Matematické modelovanie procesov je základnou metódou kybernetiky a jeho cieľom je získanie takého opisu procesu, ktorý nám umožní dobre spoznať jeho statické a dynamické vlastnosti ako celku i jeho jednotlivých častí [1].

Matematický model je matematická abstrakcia reálneho procesu. Opisuje vzťahy medzi vstupnými, stavovými a výstupnými veličinami procesu, a tým umožňuje určiť správanie sa procesu, ak sú známe jeho vstupy. Pri získavaní matematických modelov treba mať na zreteli cieľ, pre ktorý matematický model použijeme, a voliť rozumný kompromis medzi presnosťou modelu a jeho jednoduchosťou [1].

Úplný matematický model v sebe zahŕňa základné rovnice a veličiny, opis statiky, dynamiky (podmienok optimalizácie procesu a optimálneho riadenie, z hľadiska automatizácie) [1].

3.2 Elementy modelu

Modely môžu byť v princípe vytvárané ktorýmkoľvek programovacím jazykom. Veľké množstvo špeciálne určených jazykov bolo vyvinutých pre reprezentáciu správania procesov technológie.

Model by mal obsahovať nasledujúce časti:

- parametre,
- premenné,
- rovnice.

Definícia modelu nehovorí to, ako by mal byť model riešený, ale môže byť užitočné zahrnúť popis hlavnej podstaty riešenia modelu, napr. ustálený stav, neustálený alebo optimalizačný problém [21].

Model by mal byť jednoducho čitateľný a vysvetľujúci. Má obsahovať súhrnné a prídavné informácie o modelovanom systéme.

Systém

Systém je súbor prvkov, medzi ktorými existujú určité väzby a je definovaný jeho hranicami, ktoré ho oddeľujú od okolitého sveta. Tieto hranice ohraničujú objekt, ktorý je predmetom skúmania. Význam hraníc je vo vytvorení určitej zjednodušujúcej predstavy že hociktorá činnosť v hraniciach nemá žiadny efekt na okolie (všetok priestor mimo hraníc). Vytvorením tejto predstavy, že izolujeme systém od okolia, si zjednodušíme prácu analýzy

a výpočtu. Analyzovať systém aj s jeho okolím je príliš komplexný problém. Je to len aproximácia, ktorá v reálnom svete nie je možná, lebo žiadna časť fyzického sveta sa nedá úplne izolovať.

Z hľadiska druhu interakcií s okolím rozoznávame systém ktorý vymieňa (otvorený systém) alebo nevymieňa (uzavretý systém) materiálové prúdy. Systém ktorý nevymieňa ani energetické prúdy je izolovaný systém.

3.2.1 Premenné

Rozoznávame tri typy premenných:

- vstupné premenné,
- nezávislá premenná (čas),
- závislé premenné.

Vstupné premenné

Z definície systému predpokladáme, že aktivity vo vnútri hraníc systému nemajú žiadny vplyv na okolie. Tento vzťah však opačne neplatí. Všetko z okolitého sveta to, čo má vplyv na správanie systému, nazývame vstupnými veličinami. Sú to fyzikálne parametre vstupujúcich prúdov.

Vstupné veličiny rozoznávame:

- akčné – cieľavedomo meniteľné,
 - merateľné, ovládateľné,
- poruchové – náhodne sa meniace
 - merateľné, ovládateľné,
 - merateľné, neovládateľné.

V našom prípade systému zásobníkov kvapaliny je vstupnou akčnou veličinou prietok kvapaliny do prvého zásobníka. Poruchové veličiny zanedbávame.

Čas: Nezávislá premenná

Všetky závislé veličiny závisia od času rôznym spôsobom. Táto závislosť môže byť priama, alebo nepriama.

Závislé premenné

Každý fyzikálny dej, ktorý je animáciou zobrazovaný, je opísaný matematickými rovnicami, ktoré vyjadrujú závislosti medzi veličinami. Je možné vytvoriť animácie ktoré nemajú žiadne premenné ale animácie ktoré sa približujú k zobrazeniu reálneho správania systému obsahujú množstvo týchto závislých premenných [4].

Závislé veličiny rozoznávame:

- stavové – charakterizujú stav systému
 - merateľné,
 - nemerateľné,
- výstupné – fyzikálne parametre vystupujúcich prúdov zo systému
 - merateľné.

3.2.2 Rovnice

Teoretické (matematické) modely sa zostavujú na základe materiálových a energetických bilancií. Ďalej uvažujeme len materiálové bilancie.

Na získanie modelu procesu, ktorý je v ustálenom stave (jeho veličiny sa v čase nemenia), použijeme v prípade materiálovej bilancie rovnicu [24]:

$$\text{(súčet hmotností na vstupe)} = \text{(súčet hmotností na výstupe)}$$

V prípade, že sa veličiny procesu v čase menia, tzn. proces je v dynamickom stave, materiálové bilancie musia zohľadniť skutočnosť, že v systéme dochádza k akumulácii hmoty [24]:

$$\text{(súčet tokov hmotnosti na vstupe)} = \text{(súčet tokov hmotnosti na výstupe)} + \text{(rýchlosť akumulácie)}$$

Procesy so sústredenými parametrami môžeme opísať pomocou obyčajných diferenciálnych rovníc s deriváciami podľa času. Väčšina procesov má viacero vstupných a výstupných veličín a ich diferenciálne rovnice sú nelineárne.

Všeobecný model, ktorý opisuje chovanie veľkej skupiny procesov chemickej technológie v dynamickom stave je $\frac{d\mathbf{x}(t)}{dt} = \mathbf{f}[t, \mathbf{x}(t), \mathbf{u}(t), \mathbf{r}(t)]$ zo začiatočnou podmienkou $\mathbf{x}(t_0) = \mathbf{x}_0$ [2].

Model bude riešiteľný vtedy, ak počet rovníc, ktoré ho tvoria, sa bude rovnať počtu neznámych veličín, ktoré v ňom vystupujú.

Výpočet rovníc

Na získanie dynamických matematických modelov sa využívajú bilancie v neustálenom stave, kedy v systéme dochádza k akumulácii hmoty. Člen, ktorý opisuje rýchlosť akumulácie hmotnosti v systéme (tzv. akumulčný člen), je daný na základe celkovej hmotnosti v systéme vzťahom dm/dt .

V prípade zásobníka kvapaliny akumulčný člen sa rovná rozdielu vstupujúceho a vystupujúceho hmotnostného prietoku.

$$\frac{dm_i(t)}{dt} = \dot{m}_{i-1}(t) - \dot{m}_i(t) \quad (1.)$$

s počiatočnými podmienkou $m_i(0) = m_{0i}$, v čase $t=0$.

Aby sme získali správne numerické riešenie predpokladáme

$$\frac{dm_i(t)}{dt} = \lim_{\Delta t \rightarrow 0} \frac{\Delta m_i}{\Delta t} \quad (2.)$$

Kde grécke písmeno označuje malú zmenu a Δm_i je teda prírastok hmotnosti v zásobníku za malý prírastok času Δt ("step size"). Prírastok času by mal byť natoľko krátky časový úsek, aby hodnoty prírastku hmotnosti v časovom intervale $(t, t + \Delta t)$ sa veľmi nelíšili.

Ďalšie hodnoty akumulčného člena sa vypočítavajú podľa vzťahu

$$m_i(t + \Delta t) = m_i(t) + \Delta m_i = m_i(t) + \frac{dm_i(t)}{dt} \Delta t \quad (3.)$$

3.2.3 Parametre

Sú to nemenné veličiny popisujúce vlastnosti prvkov systému. Ich nemennosť môže vyplývať z rozmerov zariadenia, z konštantných fyzikálnych vlastností atď. Parametre nikdy nie sú funkciami závislých, alebo nezávislých premenných. V prípade, keď

parameter sa stáva závislým od času, musí byť odvodený kombináciou vstupných premenných a nových parametrov.

Pri určovaní parametrov treba byť opatrný, lebo môže dôjsť k nesprávnemu označeniu veličiny parametrom. V prípade zásobníkov kvapaliny zvyčajne povrch hladiny kvapaliny je označovaná ako parameter. Toto označenie je však mylné, ak zásobník kvapaliny nemá kolmé steny a povrch hladiny kvapaliny je funkciou výšky hladiny kvapaliny. Výrazom parametrom v tomto prípade možno označiť veličinu vyjadrujúcu sklon steny, ale nie povrch hladiny kvapaliny.

3.3 Linearizácia

Podstata linearizácie je založená na skutočnosti, že dynamické a statické vlastnosti nelineárnych systémov sa najlepšie sledujú v tesnom okolí ich rovnovážneho stavu, kde zmeny jednotlivých veličín sú malé. Matematická výhoda obmedzenia sa na sledovanie dynamiky nelineárnych systémov na okolie rovnovážneho stavu spočíva v tom, že v jeho okolí dynamiku nelineárnych systémov dostatočne presne vystihujú lineárne diferenciálne rovnice [1].

Modely, ktoré sa takto získavajú sa nazývajú **linearizované matematické modely**. Ich prednosťou je jednoduchosť. Ich nevýhodou je platnosť len v tesnom okolí rovnovážneho stavu, pre ktorý boli odvodené [1].

Linearizácia je založená na tom, že ľubovoľnú nelineárnu funkciu možno v okolí nejakého pracovného bodu aproximovať lineárnou závislosťou. Všeobecne môžeme povedať, že funkcia je lineárna vzhľadom na nejakú premennú, ak derivácia tejto funkcie podľa danej premennej je konštantná [24].

Postup pri linearizácii procesu opísaného diferenciálnou funkciou pozostáva z viacerých krokov [24]:

1. Napíšeme si nelineárny matematický model procesu v neustálenom tvare.
2. Odčítame od neho model procesu v ustálenom stave.
3. Definujeme odchýlkové veličiny. Odchýlková veličina je definovaná ako rozdiel veličiny v neustálenom a ustálenom stave a reprezentuje zmenu veličiny vzhľadom na ustálený stav.

4. Nelineárny člen aproximujeme Taylorovým rozvojom do prvého rádu, ktorý pre funkciu jednej premennej má tvar

$$f(x) = f(x^s) + \left. \frac{df(x)}{dx} \right|_{x=x^s} (x - x^s) \quad (4.)$$

a pre funkciu dvoch premenných $f(x, y)$:

$$f(x, y) = f(x^s, y^s) + \left. \frac{\partial f(x, y)}{\partial x} \right|_{x=x^s, y=y^s} (x - x^s) + \left. \frac{\partial f(x, y)}{\partial y} \right|_{x=x^s, y=y^s} (y - y^s) \quad (5.)$$

Tieto lineárne modely sú vhodnejšie pre analytické riešenie, návrh regulátorov a použitie v riadení než pôvodné nelineárne modely.

3.4 S-funkcia

S-funkcia (system-function) je silným mechanizmom rozširujúcim možnosti *Simulink*®. Umožňuje užívateľovi zadávať vlastné algoritmy do *Simulink* modelov.

S-funkcia je počítačový jazyk popisujúci dynamický systém. *S-funkciu* je možné napísať v MATLABe alebo v programovacom jazyku C.

Po napísaní *S-funkcie*, zadaní jej mena do *S-Function* bloku (dostupného v *Nonlinear Block* knižnici) je možné upraviť užívateľské rozhranie maskovaním.

Forma *S-funkcie* je všeobecnou funkciou a môže popisovať spojité, diskrétné a hybridné systémy. Z čoho vyplýva, že takmer všetky *Simulink* modely môžu byť opísané *S-funkciou*.

3.4.1 M-súbor S-funkcie

M-súbor, ktorý definuje *S-Function* blok musí obsahovať informácie o modeli, ktoré *Simulink* potrebuje počas simulácie. Počas priebehu simulácie, *Simulink*, ODE solver a M-súbor vzájomne pôsobia na vykonanie špecifických úloh. Tieto úlohy obsahujú definovanie počiatočných podmienok, vlastnosti bloku, diferenciálne rovnice, diskrétny stav a výstupy.

Simulink poskytuje vzor M-súboru S-funkcie, ktorý obsahuje nevyhnutné funkcie, a príkazy čím užívateľovi uľahčuje prácu pri písaní tohto kódu. Tento vzorový súbor **sfuntmpl.m** je v adresári *toolbox/simulink/blocks* pod MATLAB koreňovým adresárom.

M-súbor S-funkcie pracuje na základe postupnosti volaných programov v S-funkcii. Táto tabuľka uvádza jednotlivé rutiny S-funkcie, ktoré sa nachádzajú v M-súbore S-funkcie.

Rutiny S-funkcie	Popis
<code>mdlInitializesizes</code>	Definovanie základných charakteristík <i>S-Function</i> bloku, zahŕňajúc periódu vzorkovania, počiatočné podmienky, diskkrétne stavy a veľkosti polí.
<code>mdlDerivatives</code>	Výpočet derivácií spojitých stavových premenných.
<code>mdlUpdate</code>	Aktualizácia diskrétnych stavov, vzorkovacích časov a hlavných časovo založených príkazov.
<code>mdlOutputs</code>	Výpočet výstupov z S-funkcie.
<code>mdlGetTimeOfNextVarHit</code>	Výpočet času ďalšieho zásahu v absolútnom čase. Táto rutina sa používa len v prípade uvedenia premennej diskrétneho času.
<code>mdlTerminate</code>	Vykonáva nutné ukončenie simulácie.

Tabuľka 3-1: Rutiny M-súboru S-funkcie

3.4.2 Definovanie bloku S-funkcie

Aby *Simulink* rozoznal M-súbor S-funkcie, v súbore sa musia nachádzať špecifické informácie o S-funkcii. Týmito informáciami sú počet vstupov, výstupov, stavov

a statných vlastností bloku, ktoré získame volaním funkcie ***simsz*** na začiatku `mdlInitializeSizes` časti S-funkcie.

```
sizes = simsizes;
```

Táto funkcia vracia štruktúru neinicializovaných veľkostí.

Nižšie uvedená tabuľka uvádza oblasti štruktúry rozmerov a popisuje informáciu v nich uvedených.

Názov oblasti	Popis
<code>sizes.NumContStates</code>	Počet spojitých stavov
<code>sizes.NumDiscStates</code>	Počet diskretných stavov
<code>sizes.NumOutputs</code>	Počet výstupov
<code>sizes.NumInputs</code>	Počet vstupov
<code>sizes.DirFeedthrough</code>	Flag for direct feedthrough ???
<code>sizes.NumSampleTimes</code>	Počet vzorkovacích časov ??? (same times)

Tabuľka 3-2: Oblasti štruktúry rozmerov

Po inicializovaní štruktúry veľkostí funkcia ***simsizes*** je volaná opäť.

```
sys = simsizes(sizes);
```

Týmto príkazom informácie o štruktúre veľkostí prechádzajú do premennej s názvom **sys** do vektora, ktorá uchováva túto informáciu pre *Simulink*.

3.4.3 Jednoduché príklady S-funkcie

Najjednoduchší spôsob, ako pochopiť činnosť S-funkcie je si pozrieť príklady.

Simulink ponúka S-funkcie štyroch typov systémov:

- ✓ spojité,
- ✓ diskretné,
- ✓ hybridné,
- ✓ s premenných krokom

Všetky príklady sú na základe M-súboru S-funkcie s názvom **sfuntmpl.m**.

3.5 Modelovanie a simulácia v procese vzdelávania

Modelovanie a simulácia sú vzácnym a hodnotným nástrojom v oblasti výučby a cvičenia. Chyba, alebo omyl v procese modelovania a počas simulácie nevedie k vzniku rizika alebo spôsobeniu škôd. Veľký prínos simulácie pre študentov spočíva v poskytnutí možnosti pozorovať a riešiť kritické situácie, ktoré inak nie je možné sprostredkovať v

realite. Vykonávaním simulácie nevznikajú riskantné poškodenia technologických zariadení, poškodenie a znečistenie okolia, náklady spojené s obstaraním cvičných zariadení, ich prevádzkou, údržbou a opravami.

3.6 Vizualizácia výsledkov modelovania a simulácie

Ranné simulačné programy produkovali dlhý zoznam numerických výsledkov. V súčasnosti simulačné programy zobrazujú simulačné výsledky v grafických oknách. Užívateľ má možnosť skúmať grafické výsledky a jednoduchým spôsobom je mu umožnené získať numerické hodnoty v ľubovoľnom časovom okamihu. Ďalším krokom je vizualizácia simulácie.

V prípade zásobníkov kvapaliny závislými premennými sú výšky hladín kvapaliny v oboch zásobníkoch, vstupný a výstupný prietok do druhého zásobníka.

Aktuálne hodnoty výšok hladín určujú prírastky, ktoré sú v každom kroku simulácie vypočítavané pomocou diferenciálnych a aktuálne hodnoty prietokov z algebraických rovníc.

Tieto aktuálne hodnoty animácia znázorňuje obrazom a textom.

Veľkosť zmeny závislej premennej je vypočítavaná jej aktuálnou zmenou za malý prírastok času. Animácia je potom vytváraná za sebou idúcich aktuálnych obrazov systému v malých časových intervaloch.

Praktická realizácia

Animácia stojí čas. Simulácia by mala byť rýchla, ako je to len možné a práve animácia spôsobuje jej spomalenie. Preto by užívateľ mal mať možnosť animáciu vypnúť a zapnúť.

Na druhej strane simulácia je niekedy tak rýchla, že animácia pri zobrazovaní aktuálnych výsledkov animácie neposkytuje užívateľovi spozorovať zmeny, ktoré sú preňho potrebné.

4 Grafika v MATLABE

Spôsob ako hľadať je to najdôležitejšie čo treba pri práci vedieť. MATLAB poskytuje niekoľko možností, ktoré sa len trochu líšia v jednotlivých verziách, ale princíp ostáva rovnaký.

- MATLAB Demá
- Nápoveda zobrazená pri napísaní príkazu *help* do riadku v pracovnom okne
- Help Dokumenty v ktorých je možné sa pohybovať na základe kľúčových slov a hypertextových odkazov
- Internetová databáza, internetové diskusné fóra...

MATLAB poskytuje naozaj rozmanité a bohaté možnosti ako nájsť to čo potrebujeme, je to len vec praxe. Zo začiatku sa zdá, že v tom kvante informácií nie je možné sa orientovať. Postupne zistíte, že v MATLABe je tento problém nádherne vyriešený, čím sa získala maximálna dostupnosť a pohodlnosť získavania informácií.

Najjednoduchším a najrýchlejším spôsobom učenia je možnosť kontaktu s príkladmi a MATLAB obsahuje množstvo príkladov.

Zdroj dát

Vizualizované dáta pomocou MATLABovských funkcií môžu pochádzať priamo z bežiaceho MATLABovského programu, alebo z inej aplikácie a ako výstupné dáta sú importované a nahrané do MATLABu.

Príkazom ***load*** importujeme dáta, ktoré sú zoradené v stĺpcoch a uložené v súbore. Napríklad ak súbor *test.dat* obsahuje 20 riadkov v dvoch stĺpcoch, potom príkazom ***load test.dat*** importujeme dáta a ukladáme je do matice (20 riadkov, 2stĺpce) nazvanej ***test***. Ďalšími príkazmi na čítanie a nahrávanie z dátových súborov sú ***fread*** a ***fscf***.

4.1 XY graf

Typy XY grafov: Line, Bar, Stair, Error Bar, Polar, Stem.

PLOT(X,Y) vykreslí vektor Y proti X, kde Y je vektor závislých premenných a X nezávislých premenných. Ak Y, alebo X je matica, potom vektor je vykresľovaný po riadkoch, alebo stĺpcoch matice. Ak X je skalár a Y je vektor, potom čiara dĺžky vektora Y je vykreslená.

Príkazom **PLOT(X,Y,S)** možno definovať rôzne typy čiar, jednotlivých bodov a farby. Parameter S je znakový reťazec, ktorý môže zahŕňať po jednom znaku z nasledujúcich troch stĺpcov:

b	modrá	.	bod	-	plná čiara
g	zelená	o	kruh	:	bodkovaná
r	červená	x	krížik	-.	bodkočiarka
c	zelenomodrá	+	plus	--	
m	purpurová	*	hviezdica		
y	žltá	s	štvorec		
		v	trojuholník (dole)		
		^	trojuholník (hore)		
		<	trojuholník (doľava)		
		>	trojuholník (doprava)		
		p	pentagram		
		h	hexagram		

Tabuľka 4-1: Znaký definujúce typ čiary

Napríklad,

`plot(x,y,'ro-')` % XY graf , kde body sú znázornené krúžkami, ktoré sú % pospájané spojitou čiarou, všetko v červenej farbe

`plot(x,y,'gx')` % XY graf , kde body sú znázornené krížikmi

Viacero grafických závislostí v jednom grafe získame príkazom

PLOT(x1,y1,s1,x2,y2,s2,x3,y3,s3,...)

alebo,

plot (x1, y1, s1)

hold on;

plot (x2, y2, s2)

hold off;

PLOT vracia stĺpcový vektor ukazovateľov LINE objektov.

Každý grafický príkaz ktorý je zadaný medzi príkazmi **hold on** a **hold off** sa týka posledne generovaného grafu. Bez týchto príkazov je pre každý **plot** generované nové okno.

Pri práci s grafmi sú užitočné aj ďalšie príkazy ako napríklad:

```

  AXIS([Xmin Xmax Ymin Ymax])    % nastavenie mierky x, y osí
  TITLE ('název')                % názov umiestni do stredu nad graf
  XLABEL ('text')                % pomenovanie x osi
  YLABEL('text')                % pomenovanie y osi
  LEGEND(reťazec1, reťazec2, ...) % umiestni legendu ku každej
                                % závislosti
  TEXT (x, y, 'text')            % umiestni text na grafe podľa
                                % zadaných % súradníc % x, y

```

4.2 XYZ graf

Typy XYZ grafov: Mesh, Surf, Surfl, Contur, Quiver, Slice. Sú užitočné na prezentovanie niektorých typov premenných (teplota, tlak, atď.). Zobrazujú závislosti medzi tromi veličinami a farebne naznačujú trend jednej z nich.

PLOT3(X,Y,Z) je trojdimenzionálna analógia **PLOT()**, kde X, Y, Z sú vektory alebo matice rovnakej veľkosti.

PLOT3(x1,y1,z1,s1,x2,y2,z2,s2,x3,y3,z3,s3,...) kde s_i je znakový reťazec definujúci zobrazenie jednotlivých závislostí

PEAKS

MATLABovská funkcia **peaks** sa používa na ukážku grafických funkcií (MESH, SURF, PCOLOR, CONTOUR, atď.). Je to funkcia dvoch premenných,

$$Z = \text{peaks};$$

ktorá vracia maticu dát o veľkosti 49 x 49.

Príkazom

$$Z = \text{peaks}(n);$$

určujeme rozmer tejto matice dát ($n < 49$).

MESHGRID

Príkazy **Surface**, **mesh** a **contour** vyžadujú vstupy ako matice a príkaz **meshgrid** je vhodným príkazom na konvertovanie vektorov do matíc pre 3-D vykresľovanie.

$$[m,n] = \text{meshgrid}(x,y)$$

Výsledkom sú matice m a n . Riadky matice m obsahujú kópie vektora x a matica n obsahuje kópie vektora y v stĺpcoch.

Povrchové grafy (Surface plots)

MESH 3-D mriežkovaný povrch.

MESH(X,Y,Z,C) vykreslí trojrozmernú farebnú mriežku, kde X, Y, Z sú vektory určujúce súradnice bodov. Farebná škála je určená hranicou C.

Príkazom MESH(X,Y,Z) kde $C = Z$, farba je určovaná vzhľadom k najväčšej výške mriežky.

$$Z = \text{peaks}(24);$$
$$\text{mesh}(Z)$$

H = MESH(...) vracia ukazovateľa SURFACE objektu.

SURF 3D farebný povrch.

SURF(X,Y,Z,C) vykreslí trojrozmerný farebný povrch, kde X, Y, Z sú vektory určujúce súradnice bodov. Farebná škála je určená hranicou C.

SURF (X,Y,Z) kde $C = Z$, farba je určovaná vzhľadom k najväčšej výške mriežky.

Obrysové grafy (Contour plots)

Je možné vytvárať 2D aj 3D obrysové grafy.

- contour(Z)** 2D kontúrovité vykreslenie matice Z , kde Z -hodnoty sú interpretované ako výšky nad 2D rovinou
- contourf(Z)** 2D graf kontúr vykreslených farbou, ktorú určuje farba najbližšej nižšej kontúry
- contour3(Z)** 3D verzia kontúr, kde čiary obrysov sú vykreslené v trojrozmernom grafe v z -osi v zodpovedajúcej výške

Manipulácia s obrázkami grafov

Príkazom **rotate3d on** (stav *on* alebo *off*) bude nám umožnené pomocou kurzoru myši grafy otáčať v 3D rovine, čo nám umožní náhľad na závislosť z rôznych uhl'ov.

Pre 2D grafy, príkazom **zoom on** (stav *on* alebo *off*) bude možné klikaním ľavého tlačítka myši obraz priblížiť a pravým tlačítkom myši obraz oddialiť.

Zobrazovanie viacerých grafických závislostí

SUBPLOT(m, n, p) rozdelí okno na $m \times n$ maticu grafov, s p aktuálnym grafom. Grafy sú číslované zľava doprava a zhora dole.

Ak chceme vytvoriť okná pre každé vykreslenie, grafické príkazy pre každé okno musia byť oddelené príkazom **figure**.

$H = \text{SUBPLOT}(m,n,p)$ vracia *handle* osi.

4.3 Film

Movie funkcia vykresľuje postupnosť predznamenaných informácií.

Nasledujúci príklad vytvorí 10 snímkou funkcie $y = t \cdot e^{-x}$, $0 \leq x \leq 3$ v každom časovom intervale od 1 do 10.

```
M = moviein (10);           % vyčlenenie pamäte
x = 0 : 0.05 : 3.0;
axis ([0 3 0 10]);         % prestavenie mierky osí
title ('Sample Animation');
axis (axis)
for t =1 : 10
    plot (x, t.*exp(-x), 'g*')
    M(:,t) = getframe;      % ukladanie snímky do stĺpca matice M
end
```

Ak chcete zopakovať prehranie filmu, v príkazovom riadku napíšete príkaz ***movie(M,number)***, kde *number* určuje koľko krát postupnosť má byť prehraná. Ak *number* je záporné číslo, postupnosť je prehrávaná spätne.

Výhodou použitia tejto funkcie je že obraz môže byť prehrávaný mnohokrát, pričom dáta sú generované len raz. Nevýhodou je veľká náročnosť na pamäť.

4.4 Animácia

Animácia nie je to isté ako prehrávanie filmu. *Movie* je príkaz, ktorý prehráva predznamenanú informáciu a animácia je proces nepretržitého kreslenia objektov a menenia ich vlastností.

Nasledovný príklad vykresľuje tú istú závislosť.

```
x = 0 : 0.05 : 3.0;
p = plot(x, exp(-x), 'g*', 'EraseMode', 'xor', 'MarkerSize', 4)
axis ([0 3 0 10]);         % prestavenie mierky osí
title ('Sample Animation');
for t = 2 : 10
    drawnow                 % vykonanie posledného grafického
príkazu
    set (p, 'XData', x, 'YData', t.*exp(-x))
end
```

Výsledok je následovný. Funkcia $y = t \cdot e^{-x}$ ($0 \leq x \leq 3$) je vypočítavaná a kreslená v rozsahu x pre $t = 1$. V ďalšom kroku je funkcia vypočítaná v čase $t = 2$, vykreslená v rozsahu x a predchádzajúca séria bodov pre $t = 1$ je vymazaná. Tento spôsob vykresľovania dáva zdanie hladkého prechodu zmien funkcie.

Na rozdiel od predchádzajúceho príkladu ide o kompletný program, čo znamená že nie je možné animáciu spustiť Matlabovským príkazom a vykresľované dáta nie je potrebné nahrávať z pamäte.

V druhom riadku tohto príkladu je príkaz ***plot*** s definíciou vlastností čiary. Vlastnosť *MarkerSize* určuje veľkosť znakov bodov závislosti, prednastavená veľkosť je 6.

EraseMode je vlastnosť objektov, ktorá je podstatou triku animácie v MATLABe. Určuje ako a kedy sú objekty kreslené a mazané. Štandardne, keď je objekt pridaný, alebo zmenený, nový objekt, alebo zmeny nie sú zobrazené až vtedy, keď je os aktualizovaná. Obnovenie osí nastáva z príkazového riadku okamžite, ale pri použití M-súboru osi nie sú obnovované až do použitia príkazov *drawnow*, *getframe*, *pause* alebo *return*. Keď os je aktualizovaná, každý objekt je aktualizovaný. To znamená, že je prekreslený [10].

Nie každé objekty majú *EraseMode* vlastnosť, iba čiary, polygóny, povrchy a text.

EraseMode má štyri možné nastavenia:

normal

background

xor

none

Keď *EraseMode* vlastnosť je nastavená *normal* (štandardné nastavenie), osi určujú kedy sú objekty aktualizované. To znamená že osi blikajú, keď objekt je kreslený, menený alebo mazaný.

Keď *EraseMode* vlastnosť je nastavená *background*, tie objekty sa navzájom mažu tým že sú opätovne na seba kreslené. Hlavnou nevýhodou tohto nastavenia je že objekty, ktoré sú umiestnené za novovytvorenými objektmi sú poškodené.

Keď *EraseMode* vlastnosť je nastavená *xor*, objekty sú kreslené a mazané farbou, ktorá je za nimi. Keď nejaký objekt je vymazaný, objekt nachádzajúci sa pod ním nie je poškodený.

Keď *EraseMode* vlastnosť je nastavená *none*, objekty nie sú mazané, keď sú posunuté, alebo porušené.

Kedy sú tieto nastavenia *EraseMode* vlastnosti užitočné?

Nastavenie *normal* nepoužívať v prípade, kedy chceme eliminovať blikanie obrazu. Ak je dôležitá farba objektov a nie to čo sa stane s prekrytými objektmi, treba použiť nastavenie *background*. Poprekryvané objekty nebudú poškodené pri použití nastavenia *xor* [10].

Rýchlosť animácie

Vlastnosti, ktoré majú vplyv na rýchlosť animácie [10]:

➤ *EraseMode* – vlastnosť objektu

Použitím ktoréhokoľvek nastavenia, okrem *normal* sa výrazne zvýši rýchlosť animácie pretože os a objekty nie sú prekresľované.

➤ *BackingStore* – vlastnosť obrázku

Určuje obnovovanie okna po prekrytí iným oknom. Keď vlastnosť *BackingStore* je nastavená *On* (štandardné nastavenie) existujú dve kópie okna. Jedna je zobrazená na Okne Obrázka a druhá je používaná na obnovu miest okna, ktoré boli prekryté. Nastavením vlastnosti *Off*, existuje len jedna kópia okna, čím sa redukuje množstvo práce potrebnej na vytvorenie grafu.

Odstránenie blikania

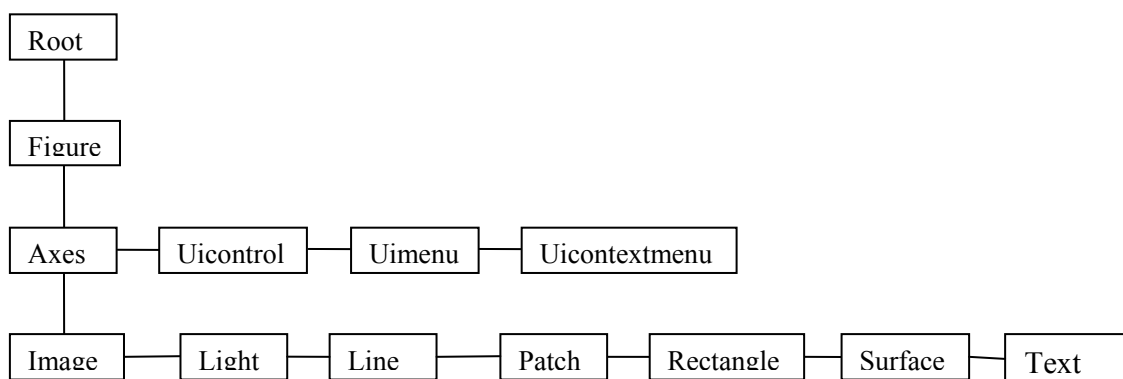
➤ *DoubleBuffer* – vlastnosť obrázku

DoubleBuffer je technika vykresľovania grafiky, ktorá produkuje *flash-free* rendering pre jednoduchú animáciu. Táto vlastnosť možno nastaviť *On* alebo *Off* (predvolené nastavenie). Nastavenie tejto vlastnosti *On* sa používa v kombinácii s nastavením *EraseMode* vlastnosti animovaného objektu na *Normal*.

4.5 Grafické objekty

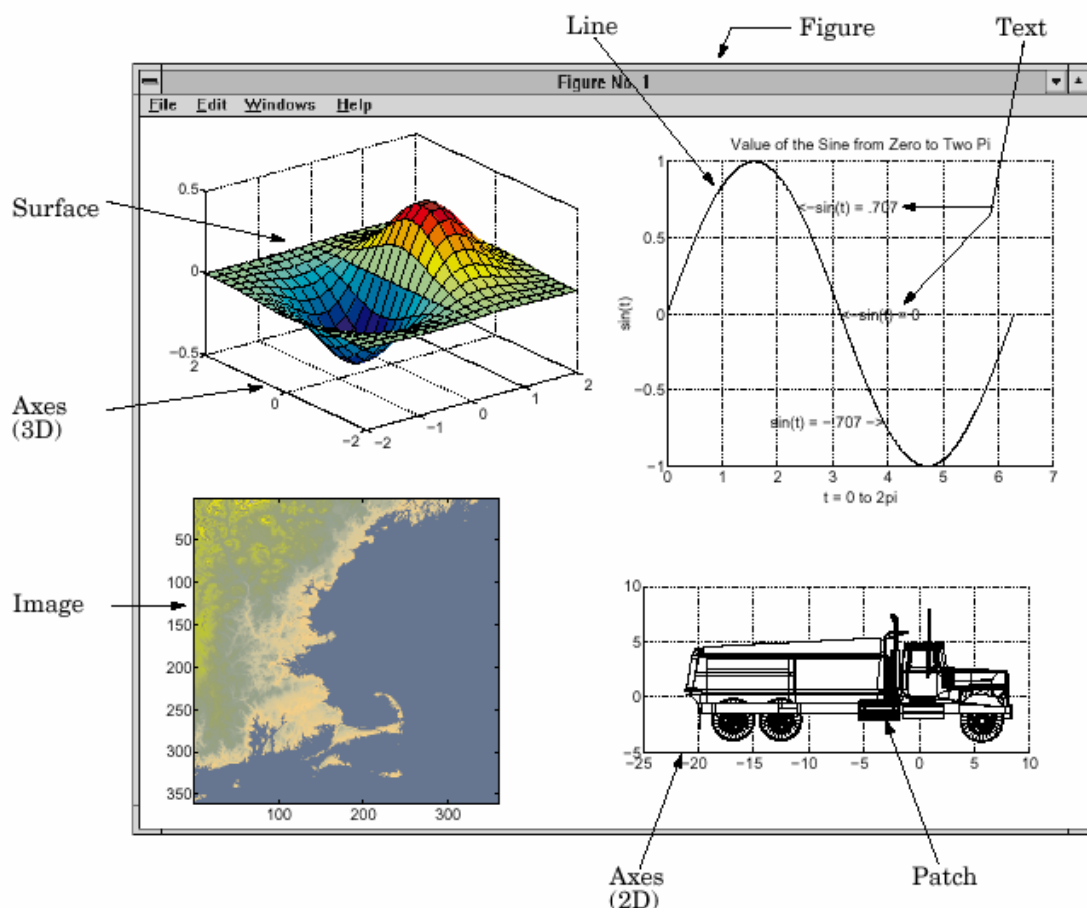
Handle Graphics je objektovo orientovaný grafický systém, ktorý poskytuje jednotlivé komponenty vytvárajúce počítačovú grafiku. Obsahuje príkazy kresliace čiary, mriežky, polygóny a text tak, ako aj interaktívne prostriedky, ako sú menu, tlačidlá a dialógové okienka. *Handle Graphics* umožňuje priamo manipulovať s čiarami, povrchmi a ostatnými grafickými časťami.

Handle Graphics objekty sú základné časti kresby, ktoré MATLAB používa na zobrazenie dát a na vytvorenie grafických užívateľských rozhraní (GUIs). Každému objektu je priradený osobitný identifikátor, ktorý sa nazýva *handle*. Prostredníctvom ukazovateľov môžeme meniť charakteristiky (nazývané vlastnosti objektu) existujúceho grafického objektu.



Obrázok 4.1 Hierarchia grafických objektov

Podstata hierarchie *Handles* objektov spočíva v nezávislosti objektov navzájom. Napríklad na to aby MATLAB nakreslil krivku potrebuje os kde krivka je umiestnená v súradnicovej sústave. A os potrebuje okno na ktorom zobrazí krivku.



Obrázok 4.2: Grafické objekty

Root

Odpovedá obrazovke počítača. Existuje len jeden *root objekt* a ostatné objekty sú od neho odvodené. Užívateľ ho nemôže vytvoriť, vzniká pri spustení MATLABu. Nastavením vlastností *root objektu* ovplyvníte grafické zobrazenie.

Obrázok (Figure)

Figure objekt je individuálne okno kde MATLAB zobrazuje grafiku. Všetky okná sú potomkovia *root objektu* a ostatné grafické objekty sú *figure nasledovníci*.

Uicontrol

Uicontrol objekty sú riadiace prvky užívateľského rozhrania, ktoré pri ich aktivovaní užívateľom volajú programy. Je niekoľko druhov ako napríklad tlačítka, listboxes a sliders. Každý z týchto objektov je navrhnutý prijať od užívateľa určitý druh informácie. *Uicontrol objekty* sú potomkovia obrázku.

Uimenu

Uimenu objekt je menu, ktorý volá príslušný program keď užívateľ zvolí určitú položku menu. MATLAB umiestni *uimenu* do ponukového riadku okna z pravej strany za menu, ktorý je definovaný systémom. *Uimenu objekty* sú potomkovia okna.

Osi (Axes)

Axes objekty definujú oblasť v okne obrázku. Osi sú potomkom obrázku a rodičom pre *image*, *light*, *line*, *patch*, *surface* a *text* objekty.

Image

Image objekt sa skladá z matice dát a farebnej škály (*colormap*). Existujú tri základné typy obrazu, ktoré sa odlišujú v spôsobe akým sú prvky matice dát interpretované ako obrazové prvky farby. *Image objekty* možno zobrazit' len ako 2D objekty.

Light

Light objekty definujú svetelné zdroje ktoré majú vplyv na všetky *patch* a *surface objekty* zobrazených na osi. Definovaním ich vlastností sa určí druh svetelného zdroja, farba, umiestnenie a vlastnosti spoločné pre všetky grafické objekty.

Line

Line objekty sú najzákladnejšie grafické časti, ktoré sa používajú na tvorbu väčšiny 2D a niektorých 3D grafov. *High-level* funkcie ako *plot*, *plot3* atď. vytvárajú *line* objekty.

Patch

Patch objekty sú olemované vyplnené polygóny, ktoré vytvárajú funkcie ako *fill*, *fill3*, a *contour3*.

Rectangle

Rectangle objekty sú vyplnené 2D oblasti, ktorých tvary sú od obdĺžnika až po elipsu. Obdĺžniky sú vhodné pre *flow-chart* kresby.

Surface

Surface objekty sú 3D reprezentácie matíc dát, ktoré sú vytvorené vykreslením každého elementu matice ako výšky nad *x-y* rovinou. *Surface* kresba sa skladá zo

štvoruholníkov, ktorých vrcholy sú určené dátami matice. *Surface objekty* MATLAB vykresľuje sýtymi, alebo interpolovanými farbami, alebo iba mriežkou línií ktoré spájajú jednotlivé body. Tieto objekty vytvárajú *high-level* funkcie ako napr. *pcolor*, *surf* a *mesh*.

Text

Text objekty sú znakové reťazce, ktoré sú vytvárané *high-level* funkciami ako napr. *title*, *xlabel*, *ylabel*, *zlabel*, a *gtext*.

4.5.1 Vlastnosti objektov

Vlastnosti grafických objektov určujú mnohé aspekty ich vzhľadu a správania sa. Zahŕňajú všeobecné informácie ako sú typ objektu, jeho rodiča a potomkov, viditeľnosť a ďalšie špecifické vlastnosti ktoré závisia od triedy objektu.

MATLAB umožňuje zistiť nastavenie ktorejkoľvek vlastnosti a väčšinu z nich je umožnené užívateľom určiť (pretože niektoré nastavenia vlastností sú iba na čítanie). Prestavenie vlastnosti sa týka len daného objektu nikdy nie objektov toho istého typu. Pri vytváraní objektu bez špecifikácie jeho vlastností MATLAB vytvorí objekt s predvolenými vlastnosťami, ktoré je možné kedykoľvek zmeniť.

4.5.2 Object Handles

MATLAB priradzuje ukazovateľa (*handle*) ku každému novovytvorenému objektu. Všetky funkcie vytvárajúce objekty vracajú ich ukazovateľov, ktorých môžeme ukladať do premenných, kedykoľvek je nájsť pomocou funkcie *findobj* alebo v zozname *parent's Children property*.

Handle root objektu je vždy nula. *IntegerHandle* vlastnosť obrázku určuje typ handlu ktorý je obrázku priradený. Ostatné *handles* grafických objektov sú desatinné čísla.

4.5.3 Hľadanie objektov funkciou *findobj*

Funkcia *findobj* prehľadáva hierarchiu objektov a nachádza ukazovatele objektov majúce špecifické vlastnosti. Ak nezadefinujete štartovací objekt *findobj* funkcia prehľadáva objekty od *root* objektu na základe zadefinovaných názvov vlastností s príslušným nastavením.

H = FINDOBJ('NázovVlastnosti', 'NastavenieVlastnosti', ...)

Vracia ukazovatele všetkých grafických objektov, ktoré majú vlastnosti s ich špecifickým nastavením. Je umožnené zadať viacero vlastností s ich nastaveniami, podľa ktorých funkcia *findobj* hľadá objekty.

```

231 % draw liquid entering the left tank
232 - if q ~= 0
233 -     set(findobj('Tag', 'StreamL', 'Visible', 'off'), 'Visible', 'on')
234 -
235 -     if (percF0 > 0 & percF0 <= 0.3)
236 -         X_stream = [bg.pipeline_left2(12,1)*ones(1,2)+2*ones(1,2) bg.pipeline_left2(12,1)*ones(1,2)+9*ones(1,2)];
237 -     elseif (percF0 > 0.3 & percF0 <= 0.7)
238 -         X_stream = [bg.pipeline_left2(12,1)*ones(1,2)+2*ones(1,2) bg.pipeline_left2(12,1)*ones(1,2)+16*ones(1,2)];
239 -     else
240 -         X_stream = [bg.pipeline_left2(12,1)*ones(1,2)+2*ones(1,2) bg.pipeline_left2(12,1)*ones(1,2)+23*ones(1,2)];
241 -     end
242 -     Y_stream = [Yl bg.roof_part3l(4,2) bg.roof_part3l(4,2) Yl];
243 -     if onetime(1) == 0
244 -         fill(X_stream, Y_stream, color_liquid, 'EdgeColor', 'none', 'EraseMode', 'background', 'Tag', 'StreamL');
245 -         onetime(1) = onetime(1)+1;
246 -     else
247 -         set(findobj('Tag', 'StreamL'), 'XData', X_stream, 'YData', Y_stream)
248 -     end
249 - else
250 -     set(findobj('Tag', 'StreamL', 'Visible', 'on'), 'Visible', 'off')
251 - end

```

Obrázok 4.3: Príklad využitia funkcie *findobj* v programe *tanksnew.m*

Funkcia *findobj* sa využíva v kombinácii s funkciou *set*, ktorej syntax je nasledovná

SET(H, 'NázovVlastnosti', 'NastavenieVlastnosti', ...)

Prestavenie vlastnosti objektu, ktorý je určený H-ukazovateľom. Ak parameter H je vektor ukazovateľov, týmto príkazom prestavíme vlastnosti všetkých takto definovaných objektov.

5 Systém zásobníkov kvapaliny

5.1 Definovanie animovaného systému

Systém zásobníkov kvapaliny je deterministický jednorozmerový systém, na ktorý pôsobí z jeho okolia zvolená, vstupná veličina a výsledkom tohto pôsobenia je pozorovateľná, výstupná veličina. Pričom pri rovnakých začiatkových podmienkach k určitej vstupnej veličine je vždy priradená daná výstupná veličina. Nezávisle premenná je čas.

Ide o systém so sústredenými parametrami, čím sú vzťahy medzi vstupnou a výstupnými veličinami opísané obyčajnými diferenciálnymi rovnicami. Závislé premenné sú funkciami času a prípadne iných parametrov.

Zásobníky kvapaliny sú prietočné procesy s jednoduchou akumuláciou hmoty. Patria medzi najrozšírenejšie zariadenie chemickej technológie.

5.2 Matematické modely

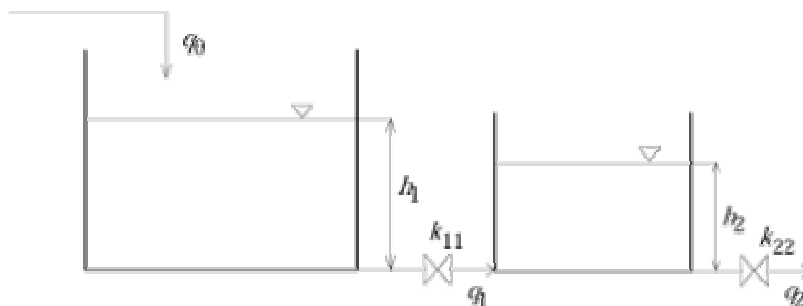
Animovaný objekt je fyzikálne reprezentovaný dvoma nádržami kvapaliny. Do prvého zásobníka kvapalina priteká o prietoku $\dot{q}_0 [\text{m}^3 \cdot \text{s}^{-1}]$ a vyteká o prietoku $\dot{q}_1 [\text{m}^3 \cdot \text{s}^{-1}]$ čo je zároveň vtokom do druhého zásobníka, z ktorého kvapalina vyteká o prietoku $\dot{q}_2 [\text{m}^3 \cdot \text{s}^{-1}]$. Z dynamického hľadiska ide o akumulačný systém 2. rádu. Parametre, ktoré ovplyvňujú dynamické vlastnosti systému sú plochy povrchov hladín kvapaliny $F_1, F_2 [\text{m}^2]$.

Podľa spôsobu sériového zapojenia zásobníkov kvapaliny rozoznávame systémy s interakciou (zásobníky sú umiestnené na rovnakej úrovni) a bez interakcie.

Zjednodušujúce predpoklady:

- ✓ Sústava zásobníkov je otvorená ($dP = 0$)
- ✓ Kvapalina má konštantnú teplotu a hustotu v celom objeme
- ✓ Plocha zásobníkov sa výškou kvapaliny nemení

5.2.1 Dva zásobníky kvapaliny s interakciou



Obrázok 5.1: Dva zásobníky kvapaliny s interakciou.

Odvozenie dynamického matematického modelu vychádza z materiálových rovníc pre každý zásobník osobitne

$$\dot{m}_0(t) = \dot{m}_1(t) + \frac{dm_1(t)}{dt} \quad \dot{m}_1(t) = \dot{m}_2(t) + \frac{dm_2(t)}{dt} \quad (6.)$$

s počiatočnými podmienkami $m_1(0) = m_{01}, \quad m_2(0) = m_{02} \quad (7.)$

Kde hmotnostný prietok \dot{m}_i je daný násobkom objemového prietoku \dot{q}_i a hustoty kvapaliny φ_i $\dot{m}_i = \dot{q}_i \varphi_i \quad \varphi_i = \text{konš.} \quad (8.)$

Závislosť medzi prietokom a výškou hladiny je nelineárna, tvaru v prípade i-teho zásobníka

s interakciou $q_i = k_{i,\text{nel}} \sqrt{h_i - h_{i+1}} \quad (9.)$

bez interakcie $q_i = k_{i,\text{nel}} \sqrt{h_i} \quad (10.)$

Potom nelineárny dynamický matematický model

$$\frac{dh_1(t)}{dt} = \frac{1}{F_1} \dot{q}_0(t) - \frac{k_{11}}{F_1} \sqrt{h_1(t) - h_2(t)} \quad \frac{dh_2(t)}{dt} = \frac{k k_{22}}{F_2} \sqrt{h_2(t)} - \frac{k_{11}}{F_2} \sqrt{h_1(t) - h_2(t)} \quad (11.)$$

Linearizáciou a zavedením odchýlkových veličín získavame odchýlkový lineárny matematický model

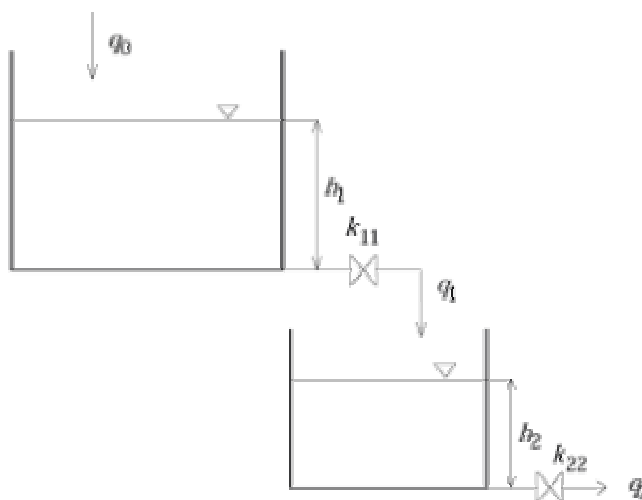
$$\frac{dX_1(t)}{dt} = \frac{1}{F_1} U(t) + \frac{K_1(t)}{F_1} X_2(t) - \frac{K_1(t)}{F_1} X_1(t) \quad (12.)$$

$$\frac{dX_2(t)}{dt} = \frac{K_1(t)}{F_2} X_1(t) - \frac{K_1(t) + K_2(t)}{F_2} X_2(t) \quad (13.)$$

kde konštanty K_1 a K_2 sú vyjadrené vzťahmi

$$K_1 = \frac{k_{11}}{2\sqrt{h_1^S(t) - h_2^S(t)}} \quad K_2 = \frac{k_{22}}{2\sqrt{h_2^S(t)}} \quad (14.)$$

5.2.2 Dva zásobníky kvapaliny bez interakcie



Obrázok 5.2: Dva zásobníky kvapaliny bez interakcie

Odvodenie dynamického matematického modelu vychádza z materiálových rovníc pre každý zásobník osobitne

$$\dot{m}_0(t) = \dot{m}_1(t) + \frac{dm_1(t)}{dt} \quad \dot{m}_1(t) = \dot{m}_2(t) + \frac{dm_2(t)}{dt} \quad (15.)$$

$$\text{s počiatočnými podmienkami} \quad m_1(0) = m_{01}, \quad m_2(0) = m_{02} \quad (16.)$$

Kde hmotnostný prietok \dot{m}_i je daný násobkom objemového prietoku \dot{q}_i a hustoty

$$\text{kvapaliny } \varphi_i \quad \dot{m}_i = \dot{q}_i \varphi_i \quad \varphi_i = \text{konš.} \quad (17.)$$

Závislosť medzi prietokom a výškou hladiny je nelineárna, tvaru

$$q_i = k_{i,\text{nel}} \sqrt{h_i} \quad (18.)$$

Nelineárny dynamický matematický model

$$\frac{dh_1(t)}{dt} = \frac{1}{F_1} \dot{q}_0(t) - \frac{k_{11}}{F_1} \sqrt{h_1(t)} \quad \frac{dh_2(t)}{dt} = \frac{kk_{11}}{F_2} \sqrt{h_1(t)} - \frac{k_{22}}{F_2} \sqrt{h_2(t)} \quad (19.)$$

Linearizáciou a zavedením odchýlkových veličín získavame odchýlkový lineárny matematický model

$$\frac{dX_1(t)}{dt} = \frac{1}{F_1} U(t) - \frac{K_1(t)}{F_1} X_1(t) \quad (20.)$$

$$\frac{dX_2(t)}{dt} = \frac{K_1(t)}{F_2} X_1(t) - \frac{K_2(t)}{F_2} X_2(t) \quad (21.)$$

kde konštanty K_1 a K_2 sú vyjadrené vzťahmi

$$K_1 = \frac{k_{11}}{2\sqrt{h_1^S(t)}} \quad K_2 = \frac{k_{22}}{2\sqrt{h_2^S(t)}} \quad (22.)$$

5.3 Animácia systému zásobníkov kvapaliny

Obrázok animácie možno rozdeliť do dvoch častí:

1. časť: Statické pozadie – je tvorené objektmi *Line* a *Patch*, ktoré vykresľujú zásobníky, potrubia a čerpadlo.
 - **Plot** funkcia vytvára *Line* objekty tvoriace obrysy statických objektov
 - Funkcia **fill** vytvára *Patch* objekty, ktoré farebne vykresľujú jednotlivé statické objekty a kvapalinu v dolnom veľkom zásobníku, z ktorého čerpadlo pumpuje kvapalinu do ľavého zásobníka.
2. časť: V čase sa meniace sa objekty – *Text* a *Patch* objekty.
 - *Text* objekty zobrazujúce aktuálne hodnoty výšok hladín ($h_1(t), h_2(t)$), prietokov ($q_0(t), q_1(t), q_2(t)$) a percentá plnosti zásobníkov.
 - *Patch* objekty zobrazujúce kvapalinu v zásobníkoch a medzi nimi prúdiacu kvapalinu.

Znaky, ktoré animácia zobrazuje

- ✓ *Patch* objekty zobrazujúce výplň zásobníkov kvapalinou, menia svoju výšku v závislosti o aktuálnych výšok vypočítaných z matematického modelu.

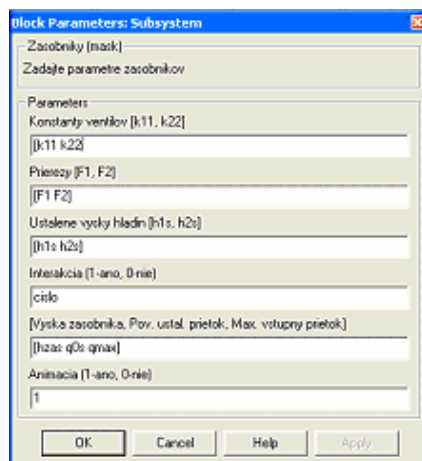
- ✓ *Patch* objekty zobrazujúce prúd kvapaliny:
 - do ľavého zásobníka, sa mení v závislosti od hodnoty akčnej veličiny,
 - do pravého zásobníka (zásobníky bez interakcie), sa mení v závislosti od množstva kvapaliny v ľavom zásobníku,
 - z pravého zásobníka, sa mení v závislosti od množstva kvapaliny v pravom zásobníku.
- ✓ *Text* objekty zobrazujú aktuálne numerické hodnoty výšok hladín, prietokov a plnosť jednotlivých zásobníkov.

Animácia bola navrhnutá pre dva spôsoby použitia.

- 1.) Animácia ako doplňujúci vizualizačný prostriedok simulačnej schémy
 - a. **hs2sanim.mdl** – sledovanie prechodových charakteristík systému
 - b. **hs2riadan.mdl** – sledovanie regulačného pochodu
- 2.) Animácia ako súčasť GUI okna – užívateľ riadi priebeh simulácie prostredníctvom GUI okna; sledovanie dynamiky prechodových charakteristík systému.

5.3.1 Animácia ako doplňujúci vizualizačný prostriedok simulačnej schémy

Animácia je vytváraná S-funkciou nazvanou **tanksnew.m**. Použitie tohto m-súboru je viazané na MATLABovský blok subsystému **Zásobníky**. Dvojitým poklepaním pravým tlačidlom myši na blok **Zásobníky** sa otvorí okno *Block parameters* (Obrázok 5.3), v ktorom užívateľ zadáva hodnoty parametrov pre systém zásobníkov a údaje potrebné pre animáciu. Posledným údajom užívateľ určí či chce simuláciu spolu animáciou.



Obrázok 5.3 Blok parametrov

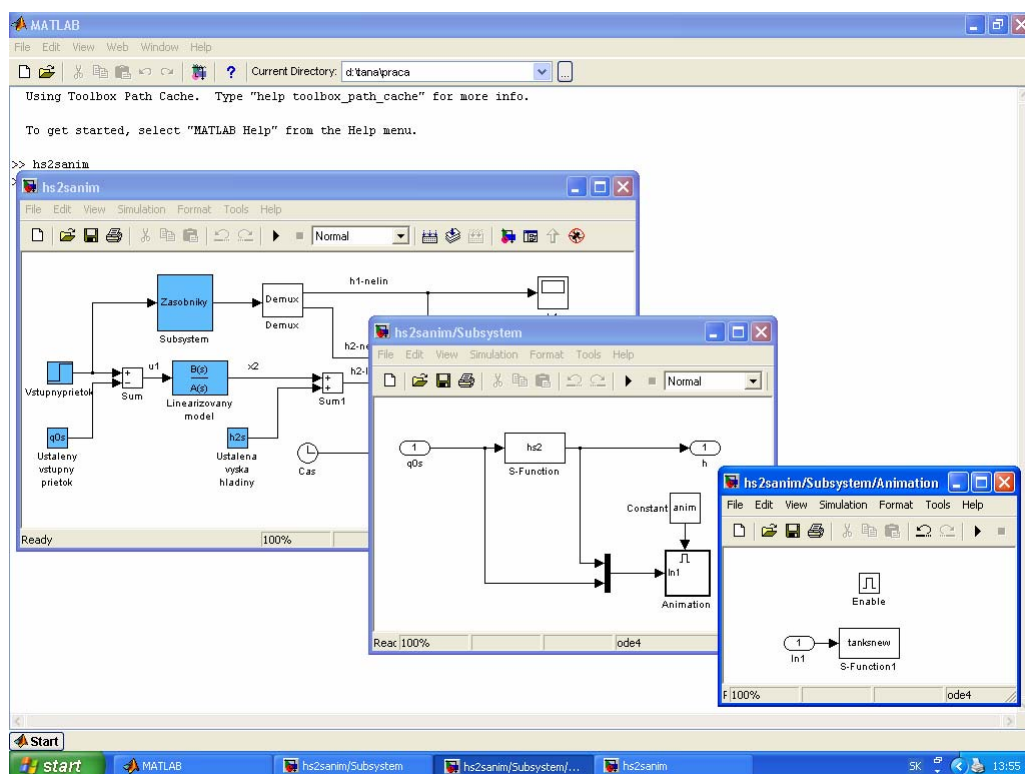
Kliknutím ľavým tlačidlom myši na subsystém **Zásobníky** a výberom položky *Look under mask* sa užívateľovi otvorí okno *hs2sanim/Subsystem*.

V tomto okne sa nachádza:

- ✓ S-funkcia **hs2** - model systému zásobníkov
- ✓ Subsystém **Animation**
- ✓ Konštanta **anim** – ktorej hodnota určuje či simulácia bude aj s animáciou
- ✓ Vstup **q0s** - vstupný prietok
- ✓ Výstup **h** – vektorová veličina obsahujúca obe výšky hladín.

Dvojitým poklepaním pravým tlačidlom myši na blok subsystému **Animation** sa otvorí *hs2sanim/Subsystem/Animation*, ktorý obsahuje:

- ✓ *Enable* blok – ktorého hodnota premennej **anim** určuje či simulácia bude s animáciou alebo bez nej
- ✓ Blok S-funkcie **tanksnew** – tvorba animácie
- ✓ Vstup **In1** - vektor obsahujúci vstup a výstupy z S-funkcie **hs2**



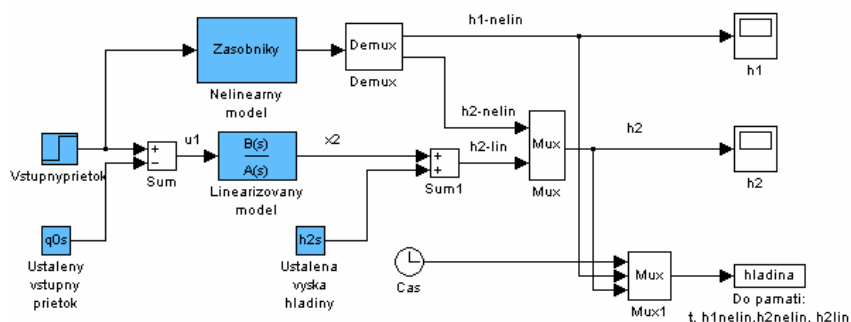
Obrázok 5.4 Štruktúra subsystému **Zásobníky**

Blok subsystému **Zásobníky** s M-súborom **tanksnew.m** je navrhnutý tak aby ich bolo možné používať v simulačných schémach týkajúcich sa systému dvoch zásobníkov kvapaliny.

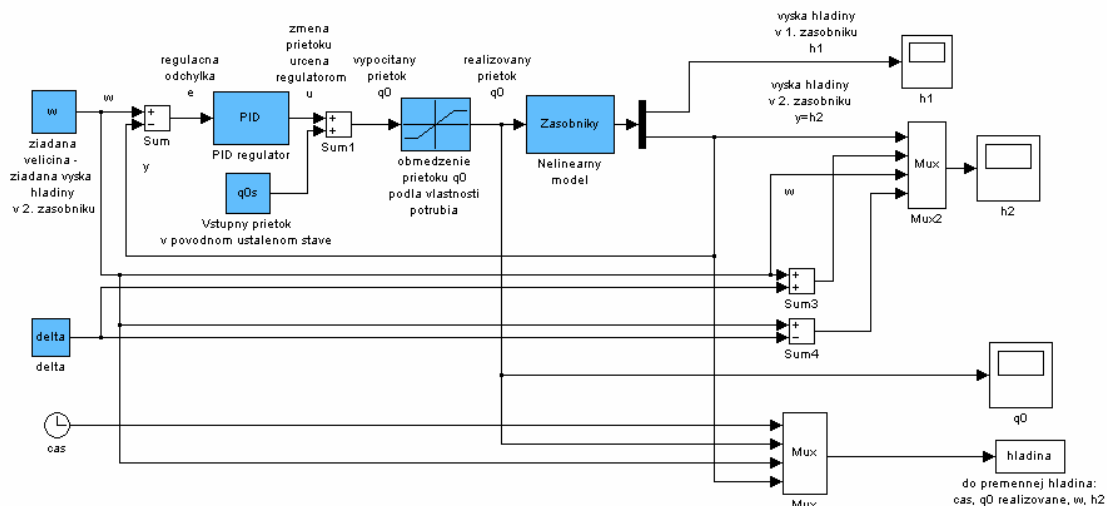
Cieľom cvičení z modelovania zo Základov automatizácie je zvládnuť základy vytvárania matematických modelov dynamických systémov a ich použitia pre simuláciu dynamických vlastností riadených procesov a v ďalšom i pre syntézu regulátorov. Študenti počas cvičení používajú dve simulačné schémy do ktorých bol zavedený hore opísaný subsystém **Zásobníky**.

hs2sanim.mdl – simulačná schéma určená na sledovanie prechodových charakteristík systému zásobníkov kvapaliny

hs2riadan.mdl – simulačná schéma určená na sledovanie priebehu regulačného pochodu

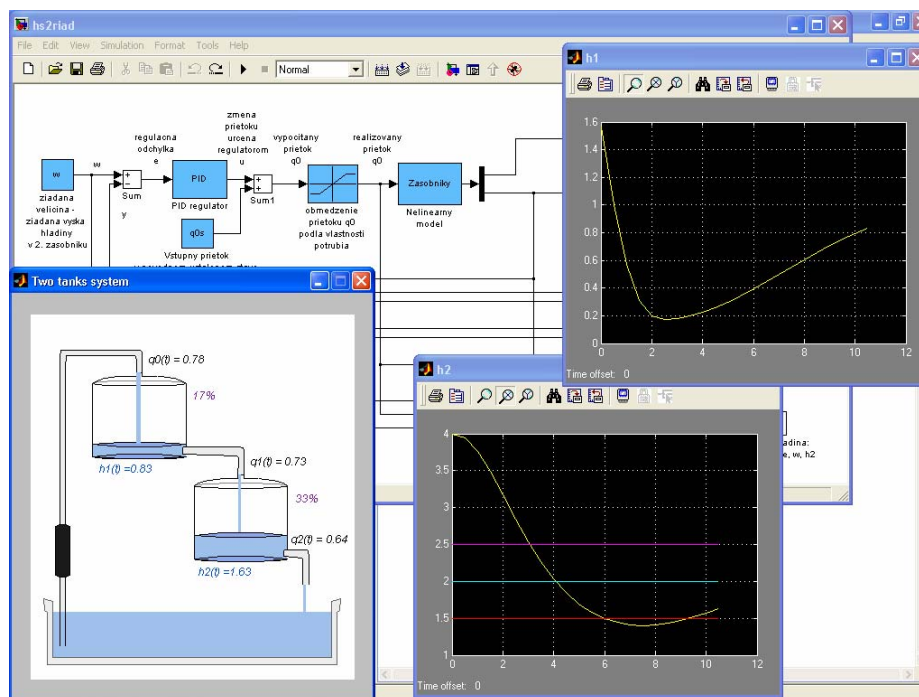


Obrázok 5.5 Simulačná schéma **hs2sanim.mdl**



Obrázok 5.6 Simulačná schéma hs2riadan.mdl

Očakávaným prínosom animácie je že študenti budú schopní ľahšie pochopiť podstatu mechanizmu správania sa zásobníkov, pri prechode z počiatočného ustáleného stavu do druhého i správanie počas regulácie.



Obrázok 5.7: Vizualizácia dát získaných simuláciou pomocou grafov a animácie

S-funkcia Zásobníky (tanksnew.m)**➤ mdlInitializeSizes**

- definovanie systému – počet vstupov, výstupov, stavových veličín atď.
- vytvorenie objektov – objekt okna, objekt osi, objekty textu na zobrazovanie aktuálnych výšok hladín, atď.
- nakreslenie obrázku systému zásobníkov bez kvapaliny

➤ mdlUpdate

- funkcia *createframe* – pri prvom prebehnutí tejto časti programu sa vytvárajú PATCH objekty napr. voda v zásobníkoch, kvapalina tečúca medzi zásobníkmi atď. V ďalších krokoch simulácie sa menia vlastnosti týchto objektov (napr. vlastnosť *Position*, *Visibility*). Text objekty zobrazujú v každom kroku simulácie aktuálne hodnoty výšok hladín, prietokov a percentá plnosti zásobníkov.
- funkcia *draw_background* – zabezpečuje správne poradie prekrývajúcich sa objektov

➤ mdlOutput

- spomalenie animácie

S-funkcia obsahuje ďalšie funkcie:

Funkcia *background* obsahuje súradnice bodov čiar statického obrázku pozadia (zásobníky, potrubia, čerpadlo) a vracia štruktúru **bg** v ktorej sú tieto parametre uložené. Inak povedané táto funkcia je iba zdrojom dát potrebných na vykreslenie obrázku statického pozadia.

Funkcia *draw_background* používa MATLABovské funkcie *plot* a *fill*. *Plot* funkcia vytvára *Line* objekty vytvárajú obrysy statických objektov a funkcia *fill* vytvára *Patch* objekty ktoré farebne vykresľujú jednotlivé statické objekty a kvapalinu v dolnom veľkom zásobníku z ktorého čerpadlo pumpuje kvapalinu do ľavého zásobníka.

5.3.2 Animácia ako súčasť GUI okna

Pomocou GUI (Graphic User Interface) je vytvorené GUI okno prostredníctvom ktorého užívateľ zadáva potrebné parametre systému a riadi animáciu. Animácia je určená na sledovanie prechodových charakteristík systému zásobníkov kvapaliny.

Užívateľ v tomto prípade nemusí mať znalosti ako pracovať s stimulačnými schémami, pretože simuláciu riadi prostredníctvom GUI okna.

Na spustenie GUI okna je potrebné jednu MDL-schému a 8 M-súborov, ktoré by mali byť prekopírované do adresáru v ktorom beží MATLAB.

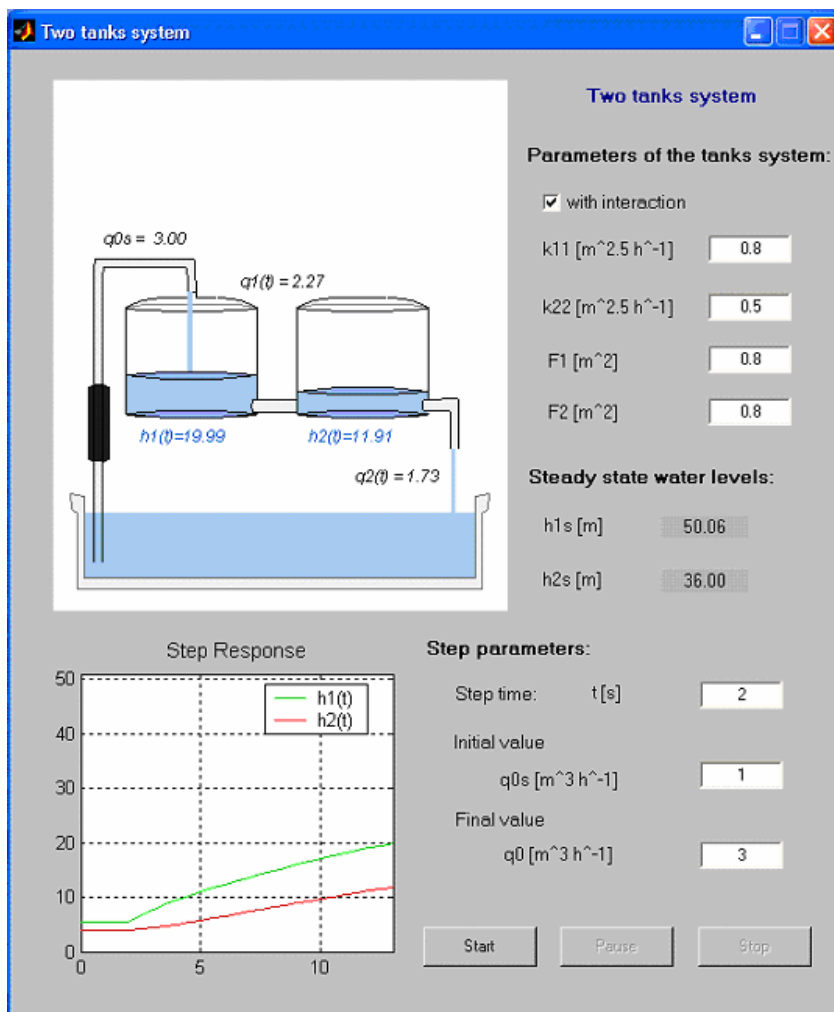
Súbory	Popis
watertanks.mdl	Simulačná schéma.
background.m	Funkcia vracajúca štruktúru bg , ktorá obsahuje súradnice bodov čiar tvoriace statický obrázok pozadia.
draw_background.m	Vytvára obrysy statických objektov objekty ktoré farebne vykresľujú jednotlivé statické objekty a kvapalinu v dolnom veľkom zásobníku, z ktorého čerpadlo pumpuje kvapalinu do ľavého zásobníka.
quifcn.m	Obsahuje príkazy, ktoré sú vykonávané pri aktivovaní editovacích polí alebo tlačidiel.
sfcn.m	S-funkcia opisujúca dynamický systém zásobníkov kvapaliny.
tankeqpoint.m	Výpočet ustálených výšok hladín.
tanklin.m	Výpočet matic lineárneho stavového opisu.
tanks.m	Tvorba animácie.

Tabuľka 5-1: Zoznam súborov

V príkazovom riadku pracovného okna MATLABu po napísaní príkazov:

```
watertanks
animation
```

Príkazmi sa vyvolá simulačné schéma a na obrazovke sa otvorí GUI okno, ktoré je na obrázku 5.8.



Obrázok 5.8: GUI okno

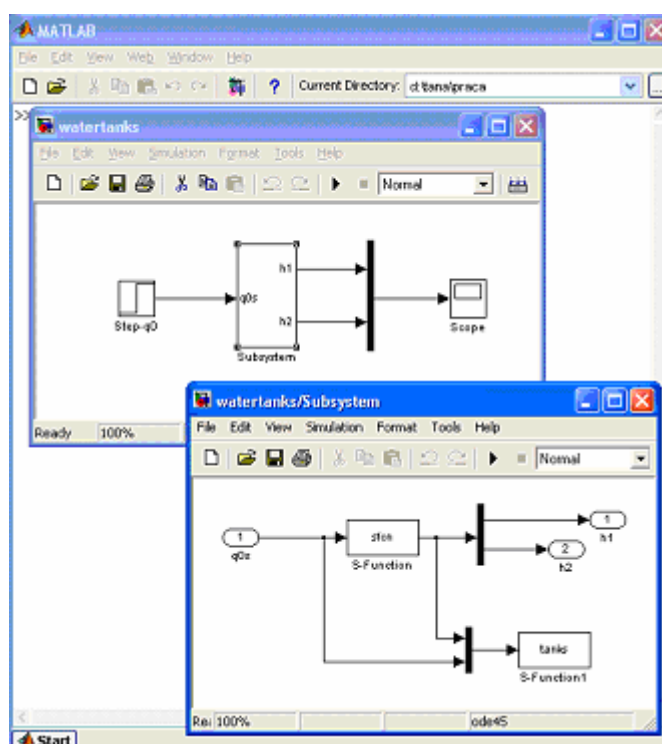
GUI okno pozostáva z nasledujúcich častí:

- ✓ Osi ktoré určujú oblasť zobrazovania animácie
- ✓ Graf znázorňujúci grafické závislosti výšok hladín zásobníkov v čase
- ✓ Editovacie políčka v oblasti **Parameters of the tank system** určené na definovanie parametrov systému
- ✓ Textové políčka v oblasti **Steady state water levels** zobrazujúce hodnoty ustálených výšok hladín
- ✓ Editovacie políčka v oblasti **Step parameters** sú určené na definovanie vstupu vo forme skokovej zmeny prietoku do ľavého zásobníka.
- ✓ Tlačidlá **Start**, **Pause**, **Stop** sú určené na spustenie simulácie, ktorého výsledkom je animácia a grafické zobrazenie výšok hladín v čase.

V príkazovom riadku je možné vyvolať matice lineárneho stavového opisu. Matice sú označené písmenami A, B, C, D.

Simulačná schéma **watertanks.mdl** (obrázok 5.7) obsahuje subsystém v ktorom sa nachádza

- ✓ S-funkcia **sfcn.m** - model systému zásobníkov
- ✓ S-funkcia **tanks.m** - tvorba animácie
- ✓ Vstup – vstupný prietok do ľavého zásobníka, vo forme skokovej zmeny
- ✓ Dva výstupy – výšky hladín kvapaliny v zásobníkoch



Obrázok 5.9: Štruktúra subsystému *Zásobníky*

6 Záver

Podnetom tejto práce bola myšlienka zefektívnenia vyučovania na katedre Informatizácie a riadenia procesov v Bratislave. Doc.Dr.-Ing. Miroslav Fikar navrhol možnosť dynamizovať statické obrazy zásobníkov kvapaliny, ktoré sú predmetom vyučovania Základov automatizácie. Aj napriek tomu, že bolo veľmi náročné získať študijné a pracovné materiály k danej téme podarilo sa nám vytvoriť jednoduchú a efektívnu animáciu.

S prácou na projekte sme začali siedmom semestri štúdiom a získavaním materiálov. Práca na animácii začala v deviatom semestri v programe MATLAB, ktorý je používaný na cvičeniach. Tento program poskytuje široké spektrum možností. Počas môjho pobytu na katedre ATP v Nemecku sme prácu dokončili. V kolektíve Prof.Dr.-Ing. Christiana Schmida sme vyriešili viacero problémov vznikajúcich pri tvorbe animácie. Počas týchto troch mesiacoch som bola v kontakte s vedúcim diplomovej práce s ktorým som konzultovala jednotlivé kroky a smerovanie práce.

V súčasnosti je animácia, ktorú sme vytvorili používaná príliš krátky čas na to aby sme objektívne posúdili jej prínos. Vzhľadom na to že prvé ohlasy na animáciu boli pozitívne predpokladáme, že animácia bude prínosom. Naša práca ukazuje smer vývoja pri tvorbe vyučovacích materiálov. Poukazuje na výhody dynamických proti statickým obrázkom a definuje postupy tvorby animácie. Hlavnú úlohu pri vzdelávaní v budúcnosti preberajú animované multimediálne vzdelávacie programy. Našou úlohou je ich tvoriť a efektívne používať.

7 Literatúra

1. Alojz Mészáros, Ján Danko, Ján Mikleš, Monika Bakošová, Základy automatizácie, Vydavateľstvo STU, 1996
2. Ján Mikleš, Vlastimil Hutla, Teória automatického riadenia, ALFA, 1986
3. Introduction to visualisation and MATLAB Graphics
www.maths.uq.edu.au/~jab/math3203/vis_notes/math3203Visn.htm
4. David M. Auslander, Animating Physics, Mechanical Engineering, University of California at Berkeley, 1999
<http://www.me.berkeley.edu/~dma/AnimatingPhysics/Animation.html>
5. Daniela Perduková, Vizualizácia technologických procesov, Mercury-Smékal, Košice, 2001
6. Pavel Roubal, Počítačová grafika pro úplné začátečníky, Computer Press, Brno, 2003
7. Paul A. Sundberg, "Animation in CALL: Learning to think in the fourth dimension" Educational Psychology, University of Illinois at Urbana-Champaign, 1998
<http://lrs.ed.uiuc.edu/students/p-sundb/homepage/animCALL.html>
8. Richard Lowe, Beyond "eye-candy": improving learning with animation, Faculty of Education, Curtin University of Technology, 2001
www.auc.uow.edu.au/conf/conf01/downloads/AUC2001_Lowe.pdf
9. Animation of Power Electronics and Electrical Drives
www.caspoc.com/manual/menuintroduction.html
10. Animation
<http://athena.fsv.cvut.cz/VG11.2001/Matlab/MatlabManual/tec2.5.html>
11. Michael J. Sanger, Computer Animations in Chemistry: What We Have Learned, Department of Chemistry, University of Northern Iowa, Cedar Falls
<http://faculty.cns.uni.edu/~sanger/Review.htm>
12. Kenneth M. Sochats, Emile Morse, Visualization, Annual Review of Information Science and Technology (ARIST), 1995.
<http://lrl13.exp.sis.pitt.edu/Website/Webresume/VisualInformationProcessingPaper/VisualIP.htm>
13. J. Scott Hamlin, Grafika Animace Kouzla na webu, UNIS Publishing, 2000

14. Animation & Interactivity Tools: Uses in Instruction
<http://wiscinfo.doit.wisc.edu/ltde/ORFI/ais/interactivity.htm>
15. Richard Lowe, Perceptual and Cognitive Challenges to Learning with Dynamic Visualisations, Curtin University
<http://www.iwm-kmrc.de/workshops/visualization/lowe.pdf>
16. Richard Lowe, Animation and learning: selective processing of information in dynamic graphics. Learning and Instruction.
17. Richard E. Mayer, Roxana Moreno, A Cognitive Theory of Multimedia Learning: Implications for Design Principles, University of California, Santa Barbara
<http://www.unm.edu/~moreno/PDFS/chi.pdf>
18. Atlas, R., Cornett, L., Lane, D. M., & Napier, H. A. (1997). The use of animation in software training: Pitfalls and benefits. In M. A. Quinones & A. Ehrenstein (Eds.), *Training for a Rapidly Changing Workplace: Applications of Psychological Research*. (pp. 281-302). Washington: American Psychological Association.
19. Kirsten R. Butcher, Cognitive Processes and Visualization, Visualisation & Enabling Technologies NCAR and the Digital Library for Earth System Education (DLESE), UCAR
http://serc.carleton.edu/files/NAGTWorkshops/visualize04/butcher_essay.pdf
20. Purvi Saraiya, Clifford A. Shaffer, D.Scott McCrickard, Chris North, Effective Features of Algorithm Visualizations, Department of Computer Science, Virginia Tech,
<http://db.grinnell.edu/sigcse/sigcse2004/viewAcceptedSession.asp?sessionType=Paper&sessionNumber=43>
21. Modelling, Simulation and Optimisation
<http://www.chemeng.ed.ac.uk/~jwp/MSO/index.html>
22. Computer Visualization, Graphics Techniques for Scientific and Engineering Analysis, edited by R.S. Gallagher, CRC Press, 1995
23. Matlab 5.3, MathWorks, 1999, buildgui.pdf, graphg.pdf, sfunctions.pdf, using_ml.pdf
24. Monika Bakošová, Miroslav Fikar, Ľuboš Čirka, Základy automatizácie, STU, 2003
25. Lih-Juan Chanlin, Attributes of Animation for LearningScientific Knowleage, Journal of Instructional Psychology, 2000
www.findarticles.com/cf_0/m0FCG/4_27/68998589/print.jhtml