Návrh aplikace a sady příkladů pro podporu výuky automatizace pro prezenční a kombinované studium

Petra Opluštilová

Bakalářská práce
2018

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2017/2018

ZADÁNÍ BAKALÁŘSKÉ PRÁCE
(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Petra Opluštilová**
Osobní číslo:	**A15573**
Studijní program:	**B3902 Inženýrská informatika**
Studijní obor:	**Informační a řídicí technologie**
Forma studia:	**prezenční**
Téma práce:	**Návrh aplikace a sady příkladů pro podporu výuky automatizace pro prezenční a kombinované studium**
Téma anglicky:	**Designing an Application and Set of Examples in Support of the Tuition of Automation for the Full-time and Combined Studies Curricula**

Zásady pro vypracování:

1. **Vypracujte literární rešerši na dané téma.**
2. **Seznamte se s prací v programu Matlab – Simulink.**
3. **Analyzujte již existující podpory výuky automatizace.**
4. **Vytvořte vlastní návrh aplikace pro podporu výuky v automatizaci.**
5. **Vytvořte příklady v Matlabu-Simulinku a umístěte je na DVD.**

Rozsah bakalářské práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. **Perůtka: MATLAB – Základy pro studenty automatizace a inform. technologií, FT UTB Zlín 2005, ISBN 80-7318-355-2.**
2. **Dušek: Matlab a Simulink – úvod do používaní. Univerzita Pardubice 2000, 146 s., ISBN 80-7194-273-1.**
3. **Zaplatílek, B. Doňar: MATLAB pro začátečníky, BEN-Technická literatura 2003, ISBN 80-7300-175-6.**
4. **BALATĚ, J. Automaticke řizeni. 2. přeprac. vyd. Praha : BEN – technicka literatura, 2004. 664 s. ISBN 80-7300-148-9.**
5. **VAŠEK, V. Teorie automatickeho řizeni II. [s.l.] : Rektorat Vysokeho učeni technického v Brně, 1990. 139 s. ISBN 80-214-0115-X.**
6. **PROKOP, Roman, Radek MATUŠŮ a Zdenka PROKOPOVÁ. Teorie automatického řízení: lineární spojité dynamické systémy. Zlín: Univerzita Tomáše Bati ve Zlíně, 2006, 102 s. ISBN 8073183692.**

Vedoucí bakalářské práce: **Ing. Karel Perůtka, Ph.D.**
Ústav řízení procesů

Datum zadání bakalářské práce: **15. prosince 2017**

Termín odevzdání bakalářské práce: **25. května 2018**

Ve Zlíně dne 15. prosince 2017

L.S.

doc. Mgr. Milan Adámek, Ph.D.
děkan

prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové/bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne 24.5. 2018

Oplušťil
....
podpis diplomanta

ABSTRAKT

Bakalářská práce se věnuje návrhem aplikace a sady příkladů pro podporu výuky automatizace. Aplikace je vytvořena v programu MATLAB. Aplikace je vytvořena ve formě hry.

Teoretická část popisuje prostředí MATLAB a základy automatizace. Z oblasti automatizace je teoretická část zaměřena na lineární systémy.

Druhá část bakalářské práce je zaměřena prakticky. Obsahuje popis vývoje hry z pohledu programátora a z pohledu uživatele. Součástí jsou řešené příklady z automatizace.

Klíčová slova: MATLAB, Simulink, Automatizace, Hra

ABSTRACT

The bachelor thesis deals with application design and set of examples to support the teaching of automation technology. The application is created in MATLAB. The application is created as a game.

The theoretical part describes the MATLAB environment and the fundamentals of automation. In automation, the theoretical part is focused on linear systems.

The second part of the bachelor thesis is oriented practically. It contains the description of the development game from the point of view of the programmer and from the point of view of the user. Examples are solved part of the automation.

Keywords: MATLAB, Simulink, Automation, Game

Tímto bych chtěla poděkovat vedoucímu bakalářské práce, panu Ing. Karlu Perůtkovi, Ph.D., za odborné vedení, rady, návrhy a připomínky během vypracování bakalářské práce.

Prohlašuji, že odevzdaná verze bakalářské/diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

OBSAH

ÚVOD

Cílem této bakalářské práce je navrhnout aplikaci a sady příkladů pro podporu výuky automatizace pro prezenční a kombinované studium. Aplikace by měla být na takové úrovni, aby posloužila jako výuková pomůcka. Pomůcka bude určena pro profesory, vyučující a hlavně pro studenty na Fakultě aplikované informatiky Univerzity Tomáše Bati ve Zlíně pro obor Inženýrská informatika.

První část práce popisuje základy programu MATLAB a Simulink. Zabývá se prostředím programu a jeho ovládáním.

Součástí teoretické části jsou základy automatizace. Tato část je zaměřena na lineární systémy a obsahuje podkapitoly: Laplaceova transformace, popis vlastností lineárních systémů a frekvenční přenos a charakteristika. Součástí je analýza již existujících podpor výuky automatizace.

Druhá část práce ja zaměřena prakticky. Aplikace pro podporu výuky automatizace je vytvořena ve formě hry. Praktická část obsahuje specifikaci hry, realizaci hry, hru z pohledu uživatele a řešenou sadu úloh a příkladů z automatizace. Součástí jsou krátké ukázky zdrojových kódů.

I. TEORETICKÁ ČÁST

1 MATLAB

Název MATLAB je zkratkou z anglického spojení *MATrix LABoratory*. Programový balík s názvem MATLAB představila americká společnost The MathWorks v roce 1984.

1.1 O MATLABu

MATLAB je integrované prostředí, díky kterému lze provádět matematické výpočty, modelování a simulace, analýza dat a jejich vizualizace, návrhy řídicích a komunikačních systémů, vývoj algoritmů, zpracování signálů a komunikace atd. Program MATLAB zvládá složité výpočetní operace.

MATLAB obsahuje několik základních částí a mezi ně patří:

- výpočetní jádro,
- grafický subsystém,
- pracovní nástroje,
- toolboxy,
- otevřenou architekturu.

Výpočetní jádro je základem vývojového prostředí. Provádí numerické operace s maticemi reálných nebo komplexních čísel. MATLAB je orientován pro práci s poli a maticemi.

Grafický subsystém zobrazuje výsledky výpočtů v grafické podobě. Práce s grafy je jednoduchá a rychlá. Lze je různě nastavovat, modifikovat a vytvořit je v dvojrozměrném nebo třírozměrném prostředí.

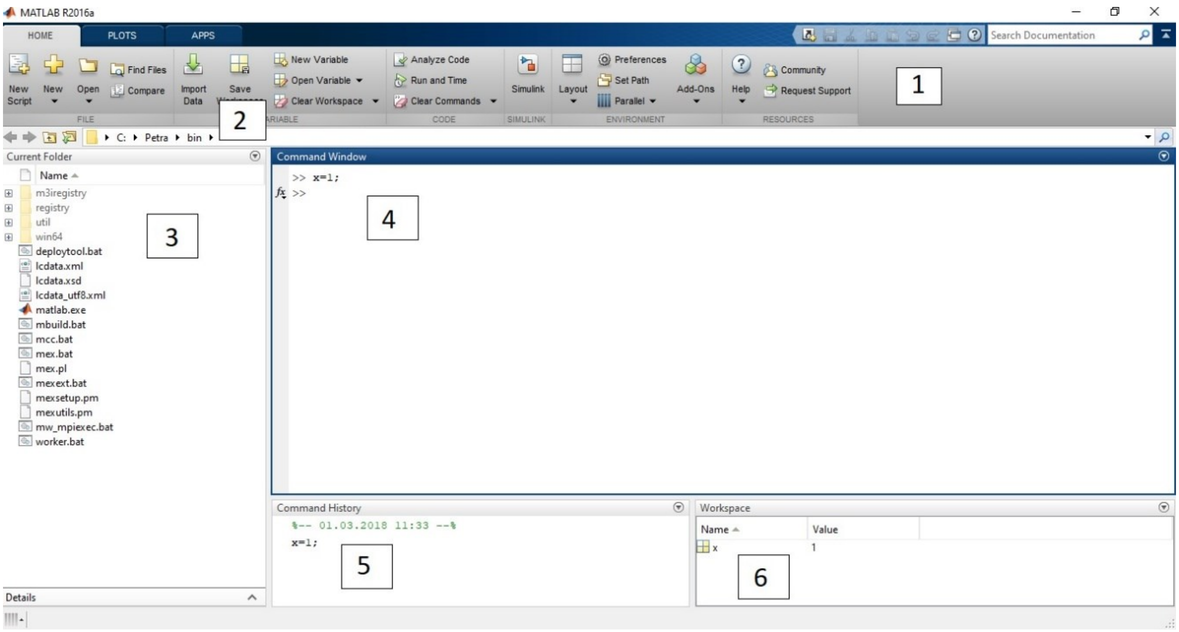
Příkazy lze zadávat jednotlivě do okna *Command Window*.

Toolboxy jsou knihovny funkcí, které jsou součástí jádra. Rozšiřují možnosti programu. Tyto knihovny funkcí jsou pak zaměřené na konkrétní vědní a technické obory. Mezi toolboxy patří např.: Symbolic Math Toolbox, Optimization Toolbox, Control System Toolbox, System Identification Toolbox a mnoho další.

Hlavním rozšířením MATLABu je systém Simulink. Simulink umožňuje pracovat se všemi funkcemi a příkazy jako s grafickými bloky. Tyto bloky je schopný vzájemně propojovat, napojovat na zdroje dat, aj. [9]

1.2 Prostředí MATLAB

Po spuštění programu MATLAB se zobrazí základní okno, které je na následujícím obrázku (Obr. 1)



Obr. 1 Pracovní plocha prostředí MATLAB

Menu (Obr. 1 položka 1) – panel hlavní nabídky se nachází v horní části okna. Tato hlavní nabídky je rozdělena na tři karty. *Home* - slouží pro základní operace se soubory, proměnnými, zobrazení nápovědy atd. *Plots* - pro výběr z několika druhů stylu grafu nebo možnost vytvoření nového stylu grafu. *Apps* – zde je přehled nainstalovaných aplikací a přístup k nim nebo slouží pro stáhnutí různých doplňujících aplikací pro jednodušší práci s MATLABem.

Průzkumník (Obr. 1 položka 2) – okno slouží pro nastavení cesty aktuálního adresáře.

Current Folder (Obr. 1 položka 3) – aktuální složka. V tomto okně jsou vypsány všechny soubory, které se nacházejí v aktuální složce. Tyto soubory můžeme spouštět dvojitým kliknutím myší na danou položku. Podle přípony se soubor otevře v editoru.

Command Window (Obr. 1 položka 4) – příkazové okno. Uživatel zde zadává jednotlivé příkazy, spouští funkce a skripty. Vypisují se zde výsledky vypočítané MATLABem. Používá se i pro ladění a testování skriptů.

Command History (Obr. 1 položka 5) – historie příkazů a funkcí. V tomto okně jsou vypsány veškeré příkazy, které byly použity během práce s MATLABem s datem a časem spuštění.

Dvojitým kliknutím myši na příkaz nebo přesunutím do okna *Command Window* pomocí myši, můžeme daný příkaz z historie použít znovu. Po smazání příkazového okna, se historie uchovává.

Workspace (Obr. 1 položka 6) – hlavní paměťový prostor. Po spuštění programu MATLAB, je toto okno prázdné. Po definování proměnných se zde vypisují spolu s jejich hodnotami. Po dvojitém kliknutí myši na proměnnou, zobrazí se detaily informace – název, rozměr, struktura, atd. Je zde lepší orientace při použití většího množství proměnných než v příkazovém okně. [5, 9]]

1.3 Simulink

Slovo Simulink bylo vytvořeno z anglických slov – SIMUlation and LINK (přeloženo do češtiny jako „Simulace a spojení“). Je to nejznámější a nejpoužívanější nadstavba v programu MATLAB. Používá se pro modelování, simulaci a analýzu dynamických systémů v přehledném grafickém uživatelském prostředí (GUI). V GUI si uživatel může sestavit vlastní model z jednotlivých bloků knihovny, které zastávají různé funkce. Uživateli je poskytnuto i možnosti vlastního návrhu bloku, případně editovat již hotové bloky. Ovládání prostředí Simulinku probíhá přes tzv. click-and-drag operace myši. Což je kliknutí myší na zvolený blok a tažení bloku na určené místo, kde se blok má nacházet.

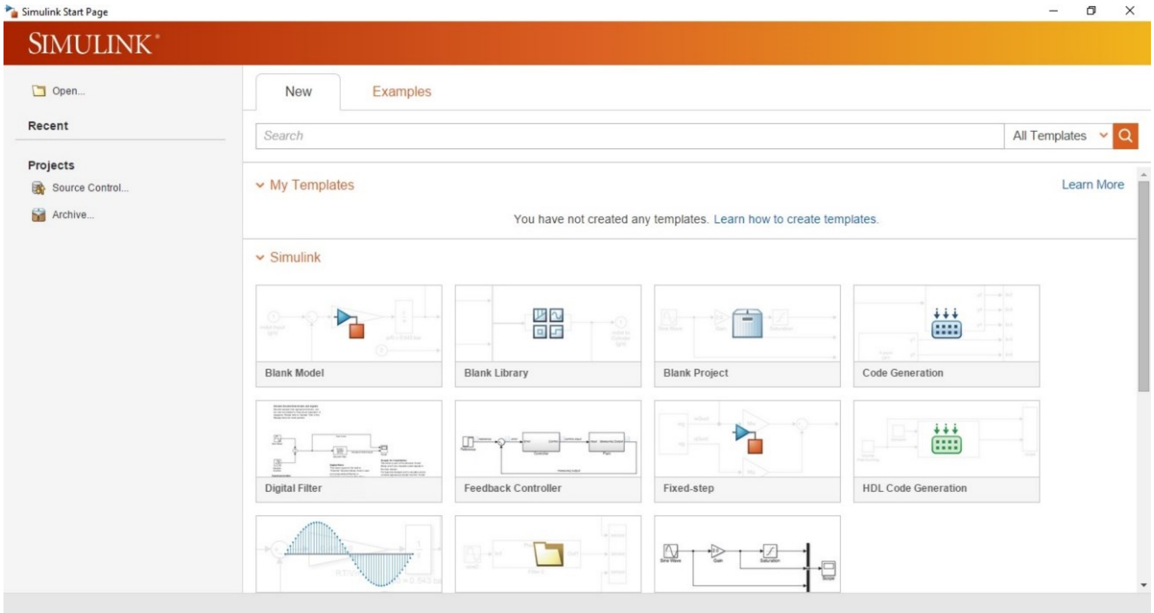
Simulink je určený pro časové řešení – simulaci – chování dynamického systému, jestli známe jeho matematický popis. Pomocí Simulinku můžeme určit časové průběhy výstupních a všech vnitřních veličin v závislosti na časovém průběhu veličin vstupních a počátečním stavu. Matematický popis soustavy může být značně rozsáhlý a složitý. [2, 3]

1.3.1 Spuštění Simulinku

Simulink lze spustit několika způsoby:

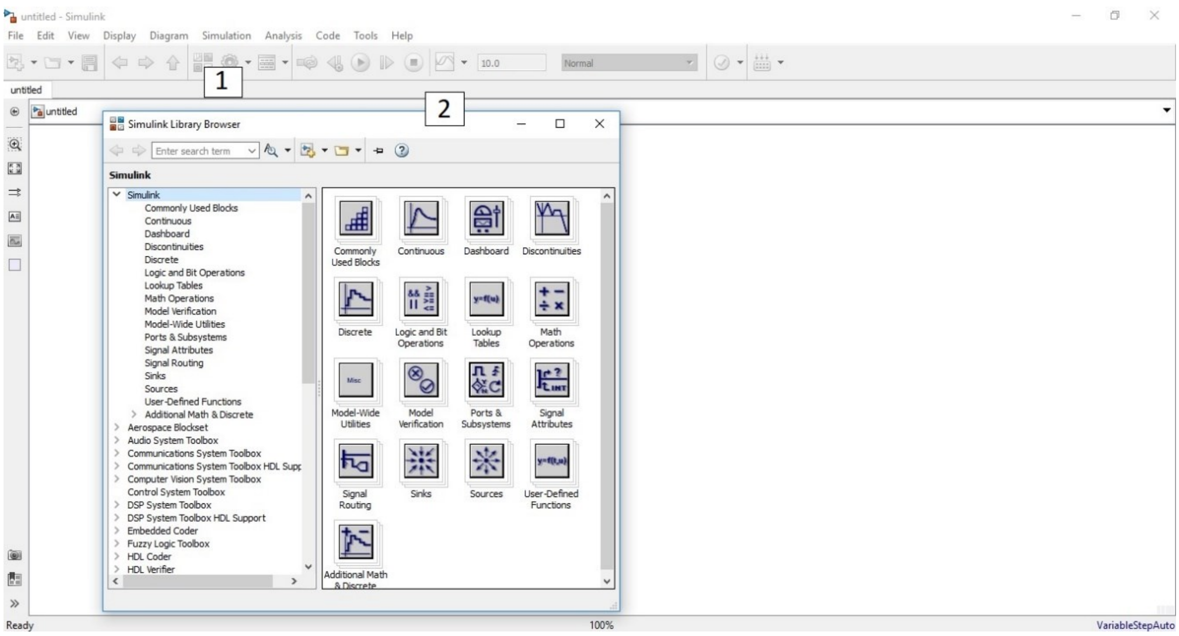
- napsáním klíčového slova „*simulink*“ do příkazového okna Command Window v MATLABu a potvrzením příkazu pomocí ENTER se Simulink spustí
- kliknutím na ikonu Simulink, která se nachází v menu (Obr. 1 – položka 1).

Po spuštění Simulinku se nám zobrazí hlavní nabídka, která je znázorněna na Obr. 2.



Obr. 2 Zobrazení hlavní nabídky po spuštění Simulinku

V hlavní nabídce můžeme otevřít již vytvořené modely. Tyto modely jsou zde i podrobně vysvětleny. Nebo si můžeme vytvořit svůj nový model pomocí nabídky *Blank Model* („prázdný model“). Na toto okno pouze stačí kliknout myší. Zobrazí se nám okno grafického prostředí, ve kterém je možnost přes nabídku *Simulink Library Browser* (Obr. 3 položka 1), otevřít další okno (Obr. 3 položka 2), ve kterém se nachází veškeré standardní knihovny Simulinku. Potřebné knihovny si může uživatel doinstalovat a je zde i možnost vytvoření vlastní knihovny. V knihovnách jsou roztříděny jednotlivé bloky podle toho, do které oblasti spadají.



Obr. 3 Okno Simulinku

Simulink Library Browser je rozdělen na dvě hlavní části. Na levé straně jsou vypsány všechny knihovny, které jsou k dispozici. V pravé části se pak nacházejí bloky, které daná knihovna obsahuje. Jednotlivé bloky se dají hledat buď podle knihovny, do které spadá daný blok, nebo vyhledat v horní části okna v textovém poli *Enter search term*, pokud víme alespoň část názvu bloku. Toto okno obsahuje ještě dvě důležité tlačítka – vytvoření nového modelu a otevření již vytvořeného modelu. Tyto tlačítka se nacházejí v horní části na levé straně.

Do okna grafického prostředí Simulinku lze jednotlivé bloky vložit z knihoven jednoduchým přitažením pomocí myši. Tyto vložené bloky uživatel může mezi sebou propojit a tak sestavit model pro simulaci.

2 AUTOMATIZACE

V dnešní době je největší snahou neustálé zvyšování produktivity práce. Hledají se nové pracovní postupy s minimální spotřebou času a nákladů. Cílem je, aby pracovní úkony byly co nejkratší a nejjednodušší, aby byli vyžádány minima lidských sil. Automatizace výrobních procesů přispívá k tomu procesu.

Člověk se snaží osvobodit od fyzické práce, od jednotvárné práce a od unavující duševní činnosti a to vede člověka k automatizaci. Práci, kterou vykonává člověk, často nyní přebírají automaty, počítače a prvky umělé inteligence. Tento poměrně složitý proces, kdy lidská řídicí činnost při výrobě i mimo výrobní proces je nahrazována činností různých přístrojů a zařízení, se nazývá automatizací. Cílem automatizace je odstranění nebo potlačení vlivu lidského faktoru na výrobní objekt nebo jiný technický objekt.

Neoddělitelným základem automatizace je řízení. Jsou vytvářeny řídicí systémy, kterou jsou bud plně automatické, nebo více či méně automatizované, kde člověk do jinak automaticky řízeného procesu zasahuje způsobem, který je spíše závislý na charakteru řízeného procesu. Teoretickou disciplínou, která se zabývá řízením jako vědní obor, je kybernetika. [4, 6, 7, 8]

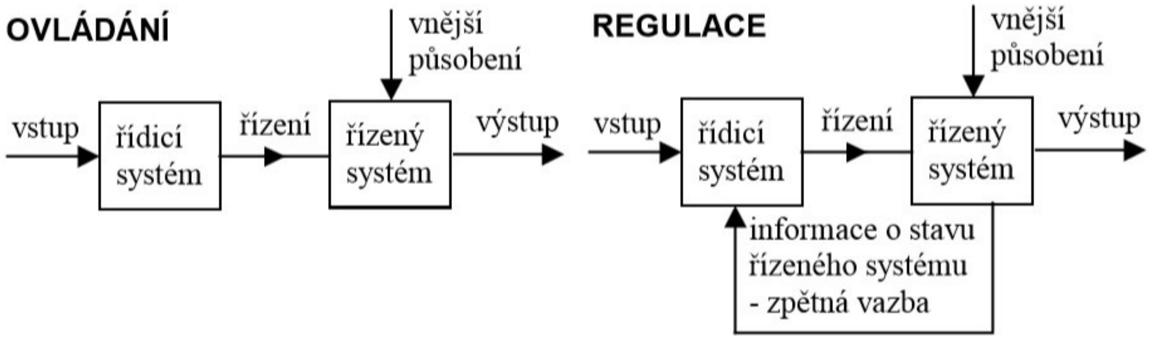
2.1 Základní rozdělení

Řízení je tedy cílené působení na řízený objekt tak, aby se dosáhlo určitého stanoveného cíle. Rozlišuje se automatizace ruční a automatická podle toho, jak se řízení provádí.

Automatické řízení se dělí ještě na řízení přímé a nepřímé. U přímého řízení řídicí proces probíhá bez přívodu energie. U nepřímého řízení řídicí proces probíhá naopak s přívodem energie.

Dalším dělením automatizace je podle hlediska, zda výsledek řízení je anebo není zpětně kontrolován. Jestli probíhá zpětná vazba řízení. Podle toho se rozlišuje ovládání, regulace a vyšší forma řízení.

- ovládání: bez zpětné vazby;
- regulace: řízení se zpětnou vazbou. Při regulaci se udržuje konstantní fyzikální veličina nebo podle nějakého pravidla se měnící hodnotě. Během regulace se zjišťují hodnoty této veličiny. Tyto hodnoty se pak srovnávají s hodnotou, která má být. Podle zjištěných odchylek se zasahuje do regulačního procesu, aby byly odchylky odstraněny či se minimalizovaly;



Obr. 4 Rozdíl mezi oběma druhy řízení – ovládáním a regulací [7, str. 6]

- vyšší formy řízení: optimální řízení, adaptivní řízení, učení a umělá inteligence.
  - optimálním řízení je takové, kdy systém dosáhne požadovaných vlastností;
  - adaptivní řízení je takové, kdy systém je schopen měnit svou strukturu. Systém mění své parametry cíleně, aby proces řízení probíhal optimálně, a to i při změnách parametrů řízeného objektu;
  - nejvyšším stupněm řízení je řízení systémy s umělou inteligencí.

Automatické řízení lze uskutečnit několika způsoby. Tyto způsoby se mezi sebou podstatně liší principem působení řídicího systému na řízený systém. Automatické řízení lze dělit na:

- logické řízení,
- spojité řízení,
- diskrétní řízení.

Logické řízení využívá k řízení dvouhodnotových veličin. Jsou tedy vždy jen dvě možnosti. Také i informace o stavu objektu jsou dvouhodnotové veličiny. Formálně jsou tyto veličiny vyjadřovány hodnotami 0 a 1 a jsou analogické s proměnnými výrokové logiky. Vztahy mezi proměnnými jsou logické funkce a řídicí obvody pracující na tomto principu jsou logické řídicí obvody.

Spojité řízení je prosazováno tam, kde jsou akční zásah a údaje o řízeném systému spojitými veličinami. Mezi vstupy a výstupy je nepřetržitá vazba, kterou vytváří spojitý řídicí systém. Žádná veličina není ani dvouhodnotová ani diskrétní. Všechny veličiny jsou spojitě proměnné v čase.

Diskrétní řízení je v dnešní době nejfrekventovanější. Důsledkem je používání digitálních obvodů a počítačů jako regulátorů. U řídicích počítačů, které neumí zpracovávat spojitý signál, je nutné převádět spojitý signál na diskrétní signál (pomocí A/D). Vztah mezi vstupem a výstupem vytváří diskrétní řídicí systém jako vztah mezi posloupnostmi impulsů, snímaných v časovém sledu daném tzv. vzorkovací periodou. Regulovaná veličina není měřena a ani akční veličina není upravována mezi okamžiky vzorkování.

V dnešní době je spojité řízení spíše na ústupu. Logické i diskrétní řízení lze provádět na jednom a tomtéž přístroji tzv. programovatelném automatu. [4, 6, 7, 8]

3 LINEÁRNÍ SYSTÉMY

Když vyšetřujeme vlastnosti dynamického systému a způsob jeho řízení, je potřebné, aby se zkoumalo, jak působí jedna část systému na jinou část systému. Dále je potřebné zkoumat, jaký vliv mají tyto interakce na chování celého systému v určitém prostředí a jak zpětně působí systém na prostředí nebo jak působí na jin systémy. K tomuto slouží matematický popis reálného systému. Tímto se vytvoří jeho matematický model, který je rovnocenný se sledovaným reálným objektem.

Reálné soustavy disponující zásobníkem některé formy energie, mají zpožděnou odezvu na změnu nějaké veličiny v reálné soustavě. Je to způsobeno dynamickou setrvačností. Z matematického hlediska se toto chování popisuje systémem diferenciálních rovnic. Lineární dynamické soustavy jsou takové, které jsou popsány soustavou lineárních diferenciálních rovnic. O diskrétních lineárních systémech a o diferenčních rovnicích se mluví, když systém je popsán jen v diskrétních časových okamžicích.

Popis chování, tj. analýza a syntéza lineárních spojitých systémů je nejstarší a nejhlouběji studovaná disciplína teorie řízení. Při analýze systémů jde o popis statických a dynamických vlastností systémů, zkoumání vztahů mezi nimi, vyšetřování stability, atd. Pod pojmem syntéza systémů, pak rozumíme stanovení struktury a parametrů regulačního obvodu tak, aby byly splněny požadavky kladené na regulační pochod. Ke zjednodušení úloh spojité lineární regulace nám slouží matematický aparát Laplaceova transformace (L-transformace) a Z-transformace. [4, 6]

3.1 Laplaceova transformace a Z-transformace

3.1.1 Přímá L-transformace a Z-transformace

Přímá L-transformace je definována jako:

$$F(s) = L\{f(t)\} = \int_0^\infty f(t)e^{-st}dt \tag{3.1}$$

kde: L – operátor přímé L-transformace, s – komplexní proměnná, $F(s)$ – spojitý obraz funkce $f(t)$ (funkce komplexní proměnné), t – čas.

Aby časová funkce *f(t)* byla originálem, musí splňovat určité podmínky:

- musí být nulová pro záporný čas *t*. Lze to splnit vynásobením dané časové funkce Heavisideovým jednotkovým skokem, který je definovaný jako:

  $\eta(t) = 1$ pro $t \geq 0$ jinak $\eta(t) = 0$

  $f(t) = \begin{cases} f(t) & \text{pro } t \geq 0 \\ 0 & \text{pro } t < 0 \end{cases}$

- musí být alespoň po částech spojitá funkce
- musí být exponenciálního řádu. Musí vyhovovat nerovnosti:

  $|f(t)| \leq Me^{\alpha_0 t}$

  kde je $M > 0,\ \alpha_0 \in (-\infty, \infty), t \in < 0, \infty)$.

Přímá Z-transformace je definována jako:

$$F(z) = Z\{f(kT)\} = \sum_{k=0}^{\infty} f(kT)z^{-k} \tag{3.2}$$

kde: *Z* - operátor přímé Z-transformace, *z* – komplexní proměnná, *F(z)* – diskrétní obraz funkce *f(kT)* (funkce komplexní proměnné), *kT* – diskrétní čas, *k* – krok, *T* – perioda vzorkování.

Aby diskrétní časová funkce *f(kT)* byla originálem, musí splňovat určité podmínky:

- musí být nulová pro záporné *k*. Lze to splnit vynásobením dané diskrétní časové funkce Heavisideovým jednotkovým skokem, který je definovaný jako:

  $\eta(kT) = 1$ pro $k \geq 0$ jinak $\eta(kT) = 0$

  $f(kT) = \begin{cases} f(kT) & \text{pro } k = 0,1,2,3,\ldots \\ 0 & \text{pro } k = -1,-2,-3,\ldots \end{cases}$

- musí být exponenciálního řádu. Musí vyhovovat nerovnosti:

  $|f(kT)| \leq Me^{\alpha_0 kT}$

  kde je $M > 0,\ \alpha_0 \in (-\infty, \infty), k = 0,1,2,3, \ldots$.

Další způsob jak určit přímou L-transformaci nebo Z-transformaci, je využití transformačního slovníku. [1, 4, 6]

3.1.2 Zpětná L-transformace a Z-transformace

Zpětná L-transformace je definována jako:

$$f(t) = L^{-1}\{F(s)\} = \frac{1}{2\pi j}\oint_C F(s)e^{st}ds \qquad (3.3)$$

kde: L^{-1} – operátor zpětné L-transformace, *s* – komplexní proměnná, *F(s)* – spojitý obraz funkce *f(t)* (funkce komplexní proměnné), *t* – čas, C – kružnice o poloměru *r*, uvnitř které leží všechny singulární body funkce *F(s)*.

Zpětná Z-transformace je definována jako:

$$f(kT) = Z^{-1}\{F(z)\} = \frac{1}{2\pi j}\oint_C F(z)z^{k-1}dz \qquad (3.4)$$

kde: Z^{-1} - operátor zpětné Z-transformace, *z* – komplexní proměnná, *F(z)* – diskrétní obraz funkce *f(kT)* (funkce komplexní proměnné), *kT* – diskrétní čas, *k* – krok, *T* – perioda vzorkování, C – kružnice o poloměru *r*, uvnitř které leží všechny singulární body funkce *F(z)*.

Další způsob jak určit zpětnou L-transformaci nebo Z-transformaci, je uvedeno níže.

Zpětná L-transformace

I) pomocí využití residuí

$$f(t) = \sum_{i=1}^{n} res[F(s)e^{st}] \qquad (3.5)$$

kde: $s = s_i$ – póly funkce $F(s), res[F(s)e^{st}]$ –residua pro jednotlivé póly s_i

- v případě jednonásobných pólů platí: $\underset{s=s_i}{res}[F(s)e^{st}] = \lim_{s\to s_i}[(s - s_i)F(s)e^{st}]$
- v případě vícenásobných pólů platí:

$$\underset{s=s_i}{res}[F(s)e^{st}] = \frac{1}{(n-1)!}\lim_{s\to s_i}\frac{d^{n-1}}{ds^{n-1}}[(s - s_i)^n F(s)e^{st}]$$

II) pomocí rozkladu na parciální zlomky a využití slovníku L-transformace

$$F(s) = \frac{Q(s)}{P(s)} = \frac{q_m s^m + \cdots q_1 s + q_0}{p_n s^n + \cdots p_1 s + p_0} \quad n > m \qquad (3.6)$$

a) nenásobné póly $F(s) = \frac{q_m s^m + \cdots q_1 s + q_0}{p_n(s - s_1)(s - s_2)\dots(s - s_n)}$

b) násobné póly $F(s) = \frac{q_m s^m + \cdots q_1 s + q_0}{p_n(s - s_1)^k(s - s_2)\dots(s - s_{n-k+1})}$

kde: $s_1, \dots, s_{n-k+1}, \dots, s_n$ – póly funkce $F(s)$ (kořeny plonymu jmenovatele $F(s)$), *k* – násobnost

- v případě nenásobných pólů (s_1 je k-násobný pól) $F(s) = \frac{A_1}{s-s_1} + \frac{A_2}{s-s_2} + \cdots + \frac{A_n}{s-s_n}$

  kde: $A_1, \dots, A_n$ - určované hodnoty

- v případě násobných pólů $F(s) = \frac{B_1}{s-s_1} + \frac{B_2}{(s-s_1)^2} + \cdots + \frac{B_k}{(s-s_1)^k} + \frac{A_1}{s-s_2} + \cdots +$

  $+ \frac{A_{n-k}}{s-s_{n-k+1}}$

  kde: $A_1, \dots, A_n, B_1, \dots, B_k$ – určované hodnoty

Zpětná Z-transformace

I) pomocí využití residuí

$f(kT) = \sum_{i=1}^{n} res[F(z)z^{k-1}]\ k = 0,1,2, \dots$ (3.7)

kde: $z = z_i$ – póly funkce $F(z), res[F(z)z^{k-1}]$ –residua pro jednotlivé póly z_i

- v případě jednonásobných pólů platí: $\underset{z=z_i}{res}[F(z)z^{k-1}] = \lim_{z\to z_i}[(z-z_i)F(z)z^{k-1}]$

- v případě vícenásobných pólů platí:

  $$\underset{z=z_i}{res}[F(z)z^{k-1}] = \frac{1}{(n-1)!}\lim_{z\to z_i}\frac{d^{n-1}}{dz^{n-1}}[(z-z_i)^n F(z)z^{k-1}]$$

II) pomocí rozkladu na parciální zlomky a využití slovníku Z-transformace

$F(z) = \frac{Q(z)}{P(z)} = \frac{q_m z^m + \cdots q_1 z + q_0}{p_n z^n + \cdots p_1 z + p_0}\ n > m$ (3.8)

c) nenásobné póly $F(z) = \frac{q_m z^m + \cdots q_1 z + q_0}{p_n(z-z_1)(z-z_2)\dots(z-z_n)}$

d) násobné póly $F(z) = \frac{q_m z^m + \cdots q_1 z + q_0}{p_n(z-z_1)^k(z-z_2)\dots(z-z_{n-k+1})}$

kde: $z_1, \dots, z_{n-k+1}, \dots, z_n$ – póly funkce $F(z)$ (kořeny plonymu jmenovatele $F(z)$), k – násobnost

- v případě nenásobných pólů (z_1 je k-násobný pól) $F(z) = \frac{A_1}{z-z_1} + \frac{A_2}{z-z_2} + \cdots + \frac{A_n}{z-z_n}$

  kde: $A_1, \dots, A_n$ - určované hodnoty

- v případě násobných pólů $F(z) = \frac{B_1}{z-z_1} + \frac{B_2}{(z-z_1)^2} + \cdots + \frac{B_k}{(z-z_1)^k} + \frac{A_1}{z-z_2} + \cdots +$

  $+ \frac{A_{n-k}}{z-z_{n-k+1}}$

  kde: $A_1, \dots, A_n, B_1, \dots, B_k$ – určované hodnoty

IV) pomocí rekurentní formule – Pierců̊v algoritmus

$$F(z) = \frac{Y(z)}{U(z)} = \frac{b_m z^m + \cdots + b_1 z + b_0}{a_n z^n + \cdots + a_1 z + a_0}$$

v záporných mocninách *z* pak obraz *F(z)* je ve tvaru:

$$F(z) = \frac{b_m z^{m-n} + \cdots + b_1 z^{-n+1} + b_0 z^{-n}}{a_n + \cdots + a_1 z^{-n+1} + a_0 z^{-n}} \tag{3.9}$$

dále platí

$$a_n y(kT) + \cdots + a_1 y\big[\big(k-(n-1)\big)T\big] + a_0 y[(k-n)T]$$
$$= b_m u\big[\big(k-(m-n)\big)T\big] + \cdots + b_0 y[(k-n)T]$$

kde: $u[(k-j)T] = 1$ pro $k = j;\ u[(k-j)T] = 0$ pro $k \neq j; y(kT) = 0$ pro$k < 0$.

[6, 7, 8]

3.1.3 Vybrané vlastnosti L-transformace a Z-transformace

Věty, které budou dále uvedeny v textu, jsou bez důkazů. Tyto důkazy lze najít v literatuře.

L-transformace

Věta o linearitě

$$L\{a_1 f_1(t) \pm a_2 f_2(f)\} = a_1 F_1(s) \pm a_2 F_2(s) \tag{3.10}$$

$$L^{-1}\{b_1 F_1(s) \pm b_2 F_2(s)\} = b_1 f_1(t) \pm b_2 f_2(t) \tag{3.11}$$

Věta o posunutí originálu, věta o zpoždění

$$L\{f(t-a)\} = e^{-as} F(s) \tag{3.12}$$

Věta o počáteční a koncové hodnotě

- věta o počáteční hodnotě

  $$f(0) = \lim_{t \to 0} f(t) = \lim_{s \to \infty} sF(s) \tag{3.13}$$

- věta o koncové hodnotě

  $$f(\infty) = \lim_{t \to \infty} f(t) = \lim_{s \to 0} sF(s) \tag{3.14}$$

Věta o derivování originálu

- po první derivaci

  $$L\left\{\frac{df(t)}{dt}\right\} = sF(s) - f(0) \tag{3.15}$$

- pro n-tou derivaci

$$L\{f^{(n)}t\} = s^n F(s) - \sum_{i=1}^{n} s^{n-i} \frac{d^{i-1} f(0)}{dt^{i-1}} =$$

$$= s^n F(s) - s^{n-1} f(0) - s^{n-2} f'(0) - \cdots - f^{(n-1)}(0) \tag{3.16}$$

Věta o integrování originálu

$$L\left\{\int_0^t f(\tau) d\tau\right\} = \frac{1}{s} F(s) \tag{3.17}$$

Z-transformace

Věta o linearitě

$$Z\{a_1 f_1(kT) \pm a_2 f_2(kT)\} = a_1 F_1(z) \pm a_2 F_2(z) \tag{3.18}$$

$$Z^{-1}\{b_1 F_1(z) \pm b_2 F_2(z)\} = b_1 f_1(kT) \pm b_2 f_2(kT) \tag{3.19}$$

Věta o posunutí v časové oblasti

- posunutí vpravo – zpoždění

$$Z\{f[(k-m)T]\} = z^{-m} F(z) m \geq 0 \tag{3.20}$$

- posunutí vlevo - předstih

$$Z\{f[(k+m)T]\} = z^m \left[F(z) - \sum_{i=0}^{m-1} f(iT) z^{-i}\right] m \geq 0 \tag{3.21}$$

Věta o počáteční a koncové hodnotě

- věta o počáteční hodnotě

$$f(0) = \lim_{k \to 0} f(kT) = \lim_{z \to \infty} \frac{z-1}{z} F(z) = \lim_{z \to \infty} F(z) \tag{3.22}$$

- věta o koncové hodnotě

$$f(\infty) = \lim_{k \to \infty} f(kT) = \lim_{z \to 1} \frac{z-1}{z} F(z) = \lim_{z \to 1} (z-1) F(z) \tag{3.23}$$

Věta o derivaci – diference v časové oblasti

- dopředná diference 1.řádu

$$Z\{\Delta f(kT)\} = Z\{f[(k+1)T] - f(kT)\} = z\big(F(z) - f(0)\big) - F(z) =$$

$$= (z-1) F(z) - z f(0) \tag{3.24}$$

- dopředná diference n-tého řádu

$$Z\{\Delta^n f(kT)\} = (z-1)^n F(z) - z \sum_{i=0}^{n-1} (z-1)^{n-i-1} \Delta^i f(0) \tag{3.25}$$

- zpětná diference 1.řádu

  $Z\{\Delta f(kT)\} = Z\{f(kT\} - f[(k-1)T]\} = F(z) - z^{-1}F(z) =$

  $= F(z)(1 - z^{-1})$ (3.26)

- zpětná diference n-tého řádu

  $Z\{\nabla^n f(kT)\} = \left(\frac{z-1}{z}\right)^n F(z) = (1 - z^{-1})^n F(z)$ (3.27)

Integrál – sumace v časové oblasti

- dopředná diference

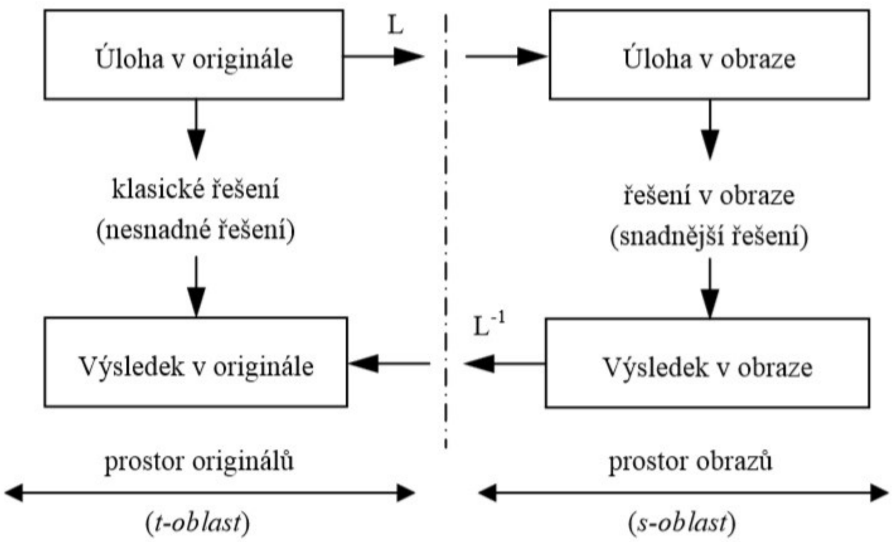
  $Z\left\{\sum_{i=0}^{k} f(iT)\right\} = \frac{1}{z-1} F(z)$ (3.28)

- zpětná diference

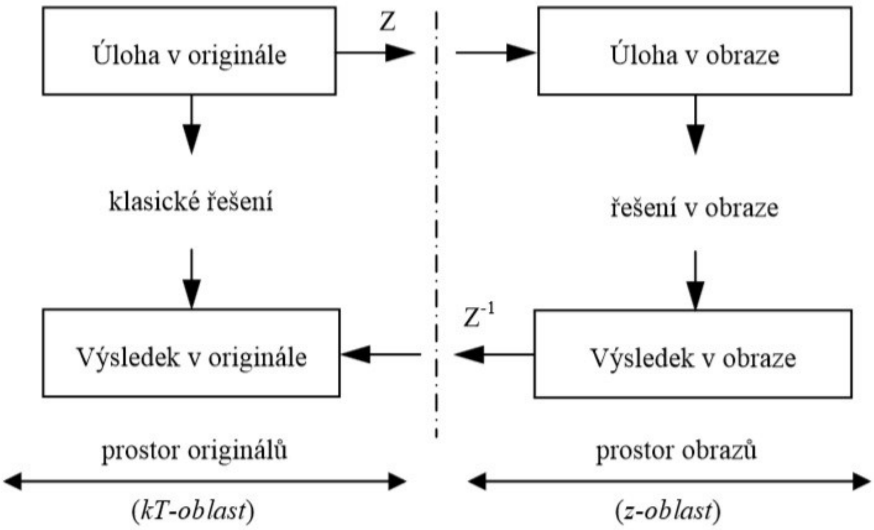
  $Z\left\{\sum_{i=1}^{k} f(iT)\right\} = \frac{z}{z-1} F(z)$ (3.29)

3.1.4 Využití L-transformace a Z-transformace

Pro řešení lineárních diferenciálních rovnic s konstantními koeficienty a pro řešení soustav takových rovnice se využívá L-transformace. Pro řešení lineárních diferenčních rovnic s konstantními koeficienty a pro řešení soustav takových rovnic se využívá Z-transformace. Na obrázcích Obr. 5 a Obr. 6 jsou znázorněné schematicky postupy při užití L-transformace a Z-transformace.



Obr. 5 Kroky řešení při využití L-transformace [4, str. 82]



Obr. 6 Kroky řešení při využití Z-transformace [4, str. 82]

3.2 Popis vlastností lineárních systémů

Dynamické systéme je možné popsat více způsoby. O vnější popis jde tehdy, když k popisu systému se používá pouze relace mezi vstupem a výstupem systému. O vnitřním popisu (stavovém popisu) je řečeno, pokud se k popisu systému využívají další veličiny, které nemusí být na systému přímo měřitelné.

Vnější popis lineárního systému je možno vyjádřit několika způsoby, které jsou si rovnocenné. Mezi tyto způsoby patří:

- lineární diferenciální rovnice a lineární diferenční rovnice,
- přenos systému v L-transformaci a v Z-transformaci,
- rozložení pólů a nul přenosu systému v komplexní rovině,
- impulsní funkce, impulsní charakteristika,
- přechodová funkce, přechodová charakteristika,
- frekvenční přenos, frekvenční charakteristika.

3.2.1 Lineární diferenciální rovnice, lineární diferenční rovnice a přenos

Popis spojitého systému

Spojitý systém, který má jednu vstupní a výstupní veličinu, lze popsat lineární diferenciální rovnicí n-tého řádu. Tuto rovnici lze zapsat ve tvaru:

$$a_n y^{(n)}(t) + a_{n-1} y^{(n-1)}(t) + \cdots + a_1 y'(t) + a_0 y(t) = b_m u^{(m)}(t) + b_{m-1} u^{(m-1)}(t) + \cdots + b_1 u'(t) + b_0 u(t)\ m \leq n \tag{3.30}$$

kde: a_i, b_i – konstantní koeficienty, $u(t), y(t)$ – vstupní a výstupní veličina systému.

Počátečními podmínkami jsou:

$$y(0) = y'(0) = \cdots = y^{(n-1)}(0) = 0; u(0) = u'(0) = \cdots = u^{(m-1)}(0) = 0.$$

Přenos systému

Z diferenciální rovnice lze určit přenos systému, který je definovaný jako poměr Laplaceova obrazu výstupní veličiny k Laplaceovu obrazu vstupní veličiny a to při nulových počátečních podmínkách systému $y(0) = y'(0) = \cdots = y^{(n-1)}(0) = 0$ a vstupního signálu $u(0) = u'(0) = \cdots = u^{(m-1)}(0) = 0$. Přenos systému se určí z diferenciální rovnice použitím L-transformace a s využitím věty o n-té derivaci při splnění výše uvedených podmínek:

$$(a_n s^n + a_{n-1} s^{n-1} + \cdots + a_1 s + a_0) Y(s) = (b_m s^m + b_{m-1} s^{m-1} + \cdots + b_1 s + b_0) U(s).$$

Přenos systému je dán vztahem:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_m s^m + b_{m-1} s^{m-1} + \cdots + b_1 s + b_0}{a_n s^n + a_{n-1} s^{n-1} + \cdots + a_1 s + a_0} \tag{3.31}$$

Pomocí pólů a nulových bodů lze přenos systému také vyjádřit:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_m (s - n_1) \ldots (s - n_j) \ldots (s - n_m)}{a_n (s - p_1) \ldots (s - p_j) \ldots (s - p_n)} \tag{3.32}$$

kde: n – nuly (kořeny čitatele), p – póly (kořeny jmenovatele).

Pomocí časových konstant lze také vyjádřit přenos systému. Pro časové konstanty platí:

- převrácené hodnoty reálných pólů jsou označovány jako časové konstanty jmenovatele

  $$T_i = -\frac{1}{p_i}$$

- převrácené hodnoty reálných nul jsou označovány jako časové konstanty čitatele $\tau_i = \frac{-1}{n_i}$

Pokud jsou nulové body i póly sytému reálné, pak přenos systému lze vyjádřit pomocí časových konstant ve tvaru:

$$G(s) = \frac{b_0}{a_0}\frac{(1+s\tau_1)(1+s\tau_2)\ldots(1+s\tau_m)}{(1+sT_1)(1+sT_2)\ldots(1+sT_n)} \tag{3.33}$$

Poměr $\frac{b_0}{a_0} = k_0$ se nazývá zesílení systému.

Popis diskrétního systému

Diskrétní systém, který má jednu vstupní a výstupní veličinu, lze popsat lineární diferenční rovnicí *n*-tého řádu. Tato rovnice může mít dvojitý tvar.

Diferenční tvar

$$\begin{aligned}\widetilde{a_n}\Delta^n y(kT) + \cdots + \widetilde{a_1}\Delta y(kT) + \widetilde{a_0}y(kT) &= \\ &= \widetilde{b_m}\Delta^m u(kT) + \tilde{b}_{m-1}\Delta^{m-1}u(kT) + \cdots + \tilde{b}_1\Delta u(kT) + \tilde{b}_0\Delta u(kT)\end{aligned} \tag{3.34}$$

kde: $\tilde{a}_{i,}\ \tilde{b}_i$ – konstantní koeficienty systému, $u(kT), y(kT)$ – vstupní a výstupní veličina systému.

Počátečními podmínkami jsou:

$$y(0) = \Delta y(0) = \cdots = \Delta^{n-1}y(0) = 0; u(0) = \Delta u(0) = \cdots = \Delta^{m-1}u(0) = 0.$$

Přenos systému

Z diferenční rovnice lze určit přenos systému, který je definovaný jako poměr Z-obrazu výstupní veličiny k Z-obrazu vstupní veličiny a to při nulových počátečních podmínkách systému $y(0) = \Delta y(0) = \cdots = \Delta^{n-1}y(0) = 0$ a vstupního signálu $u(0) = \Delta u(0) = \cdots = \Delta^{m-1}u(0) = 0$. Přenos systému se určí z diferenční rovnice použitím Z-transformace a s využitím věty o *n*-té diferenci (dopředná diference) při splnění výše uvedených podmínek:

$$(\hat{a}_z z^n + \hat{a}_{n-1}z^{n-1} + \cdots + \hat{a}_1 z + \hat{a}_0)Y(z) = \left(\hat{b}_m z^m + \hat{b}_{m-1}z^{m-1} + \cdots + \hat{b}_1 z + \hat{b}_0\right)U(z).$$

Přenos systému je dán vztahem (v kladných mocninách z):

$$G(z) = \frac{Y(z)}{U(z)} = \frac{\hat{b}_m z^m + \hat{b}_{m-1}z^{m-1} + \cdots + \hat{b}_1 z + \hat{b}_0}{\hat{a}_z z^n + \hat{a}_{n-1}z^{n-1} + \cdots + \hat{a}_1 z + \hat{a}_0} \tag{3.35}$$

Mezi koeficienty $\tilde{a}_{i,}\ \tilde{b}_i$ a $\hat{a}_i, \hat{b}_i$ platí přepočtové vztahy. Ty lze získat při úpravě dané diferenční rovnice s využitím věty o dopředné diferenci obecně *n*-tého řádu při nulových počátečních podmínkách a to srovnáním koeficientů u jednotlivých mocnin.

V případě, kde je diferenční rovnice ve tvaru zpětné diference:

$$\bar{a}_n \nabla^n y(kT) + \cdots + \bar{a}_1 \nabla y(kT) + \bar{a}_0 y(kT) == \bar{b}_m \nabla^m u(kT) + \bar{b}_{m-1} \nabla^{m-1} u(kT) + \cdots + \bar{b}_1 \nabla u(kT) + \bar{b}_0 u(kT) \tag{3.36}$$

Počátečními podmínkami jsou:

$$y(0) = \nabla y(0) = \cdots = \nabla^{n-1} y(0) = 0; u(0) = \nabla u(0) = \cdots = \nabla^{m-1} u(0) = 0.$$

Přenos systému je dán vztahem (v záporných mocninách z):

$$G(z) = \frac{Y(z)}{U(z)} = \frac{b'_m + b'_{m-1} z^{-1} + \cdots + b'_1 z^{-(m-1)} + b'_0 z^{-m}}{a'_n + a'_{n-1} z^{-1} + \cdots + a'_1 z^{-(n-1)} + a'_0 z^{-n}} \tag{3.37}$$

Mezi koeficienty $\bar{a}_{i,}\ \bar{b}_i$ a a'_i, b'_i platí přepočtové vztahy. Ty lze získat při úpravě dané diferenční rovnice s využitím věty o zpětné diferenci obecně *n*-tého řádu při nulových počátečních podmínkách a to srovnáním koeficientů u jednotlivých mocnin.

Rekurentní (normální) tvar

Rekurentní (normální) tvar lze zapsat vztahem:

$$\hat{a}_n y[(k+n)T] + \hat{a}_{n-1} y[(k+n-1)T] + \cdots + \hat{a}_1 [y(k+1)T] + \hat{a}_0 y(kT) =$$
$$= \hat{b}_m u[(k+m)T] + \hat{b}_{m-1} u[(k+m-1)T] + \cdots + \hat{b}_1 u[(k+1)T] + \hat{b}_0 u(kT) \tag{3.38}$$

pro počáteční podmínky (*k*=0), které jsou rovny funkčním hodnotám:

$$y(0), y(T), \dots, y[(n-1)T]; u(0), u(T), \dots, u[(m-1)T]; m \leq n.$$

Přenos systému lze z tohoto tvaru odvodit, jestliže by počáteční podmínky byly nulové. Toto odvození lze pomocí využití věty o posunutí v časové oblasti vlevo (předstih). Výsledný tvar přenosu by odpovídal tvaru uvedeném výše (3.36), tj. rovnici s přenosem v kladných mocninách *z*.

Možné je použít i následující rekurentní tvar:

$$a'_n y(kT) + a'_{n-1} y[(k-1)T] + \cdots + a'_1 y[(k-n+1)T] + a'_0 y[(k-n)T] =$$
$$= b'_m u(kT) + b'_{m-1} u[(k-1)T] + \cdots + b'_1 u[(k-m+1)T] + b'_0 u[(k-m)T] \tag{3.39}$$

pro počáteční podmínky (k=0), které jsou rovny funkčním hodnotám:

$$y(-T), \dots, y[(-n+1)T], y(-nT); u(0), u(-T), \dots, u[(-m+1)T], u(-mT); m \le n.$$

Přenos systému opět lze z tohoto tvaru odvodit, jestliže by počáteční podmínky byly nulové. Toto odvození lze pomocí využití věty o posunutí v časové oblasti vpravo (zpoždění). Výsledný tvar přenosu by odpovídal tvaru uvedeném výše (3.37), tj. rovnici s přenosem v záporných mocninách z.

Diferenční tvary 3.34 a 3.36, které jsou uvedené výše, jsou přímým analogem lineární diferenciální rovnice n-tého řádu, která slouží pro popis spojitého systému. U diskrétních systémů se častěji využívá rekurentní tvar 3.39 z důvodu znalosti počátečních podmínek a vstupního signálu je možné určit řešení přímo rekurentním systémem. [4, 6]

3.2.2 Póly a nuly systému

Ze zadaného přenosu systému, kde polynomy jmenovatele a čitatele jsou rozloženy v součin kořenových činitelů, lze určit póly a nuly, popřípadě poloha pólů a nul systému. Jedním z možných způsobu jak vyjádřit přenos systému, je pomocí pólů a nul systému. Póly mají vliv i na stabilitu systému, který je ovlivněn umístěním kořenů jmenovatele přenosu systému v komplexní rovině „s“.

Spojité systémy

Tvar přenosu spojitého systému pomocí pólů a nul:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_m s^m + b_{m-1} s^{m-1} + \cdots + b_1 s + b_0}{a_n s^n + a_{n-1} s^{n-1} + \cdots + a_1 s + a_0} = \frac{b_m (s-n_1)\dots(s-n_j)\dots(s-n_m)}{a_n (s-p_1)\dots(s-p_j)\dots(s-p_n)} \tag{3.40}$$

Póly a nuly mohou být reálné, komplexně sdružené nebo ryze imaginární. Mohou být jednonásobné i vícenásobné. Aperiodický resp. nekmitavý přechodový děj zapříčiňují reálné póly. Póly, které jsou komplexně sdružené, způsobují naopak periodický resp. kmitavý přechodový děj. Integrační charakter je vytvářen póly v počátku. Derivační charakter je způsoben nulami v počátku. Důležitá je poloha pólů a nul v komplexní rovině vzhledem k imaginární ose. V pravé polorovině, kde póly a nuly mají kladnou reálnou část, jsou nestabilní póly i nuly. V levé polorovině, kde póly a nuly mají zápornou reálnou část, jsou stabilní póly i nuly. Přechodový děj je více tlumen s větší vzdáleností stabilních pólů od imaginární osy. Derivační složka přenosu bude převládat tehdy, kdy nuly budou blíže imaginární ose než póly.

Diskrétní systémy

Tvar přenosu diskrétního systému pomocí pólů a nul:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_m z^m + b_{m-1} z^{m-1} + \cdots + b_1 z + b_0}{a_n z^n + a_{n-1} z^{n-1} + \cdots + a_1 z + a_0} = \frac{b_m (z-n_1) \ldots (z-n_j) \ldots (z-n_m)}{a_n (z-p_1) \ldots (z-p_j) \ldots (z-p_n)} \tag{3.41}$$

Z popisu, který je uvedený výše pro spojité systémy, tak je možné vyjít i pro diskrétní systémy, ale je zde několik rozdílů. Stabilita diskrétních systému je formulována uvnitř jednotkové kružnice. Systém je stabilní, když všechny póly leží uvnitř jednotkové kružnice v komplexní rovině „z“. S využitím transformačního vztahu pro přechod mezi „s“ a „z“, lze odvodit z popisu, který je uvedený u spojitých systémů.

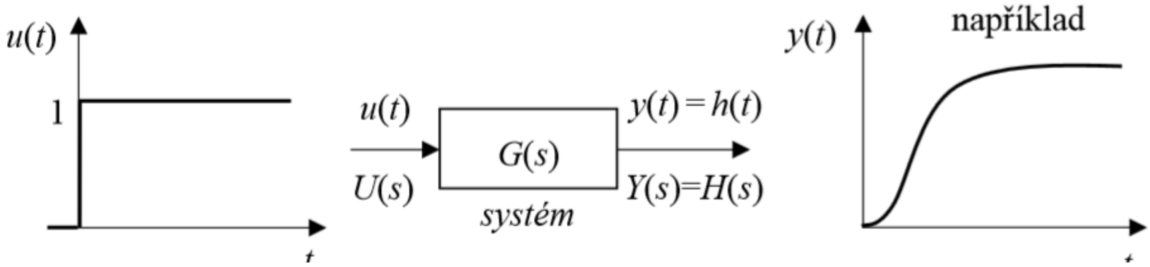
$$z = e^{sT}, s = \frac{1}{T} \ln z \tag{3.42}$$

Reálné záporné póly zapříčiňují periodický resp. kmitavý přechodový děj. Reálné kladné póly způsobují aperiodický resp. nekmitavý přechodový děj. Póly, které jsou komplexně sdružené, způsobují kmitavý přechodový děj. Integrační neboli sumační charakter způsobují póly, které nabývají hodnoty $z = 1$. Derivační neboli diferenční charakter způsobují nuly, které nabývají hodnoty $z = 1$. [4, 6]

3.2.3 Přechodová funkce a charakteristika, impulsní funkce a charakteristika

3.2.3.1 Spojitá verze přechodové funkce a charakteristiky, impulsní funkce a charakteristiky

Spojitá přechodová funkce se označuje $h(t)$. Je to odezva systému na vstupní signál $u(t)$ ve tvaru jednotkového – Heavisideova – skoku při nulových počátečních podmínkách systému.

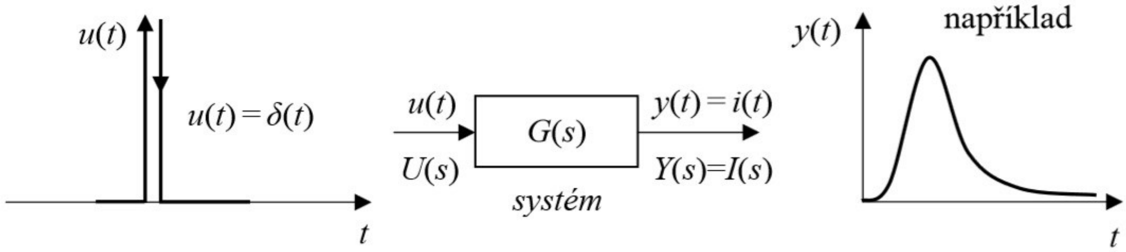


Obr. 7 Odezva systému na jednotkový – Heavisideův-skok [6, str. 101]

Laplaceův obraz jednotkového skoku je $U(s) = \frac{1}{s}$. Laplaceův obraz výstupní funkce $y(t) = h(t)$, tedy $H(s)$ je ve tvaru:

$$Y(s) = G(s)U(s) = G(s)\frac{1}{s} = \frac{G(s)}{s} = H(s) \Rightarrow h(t) = L^{-1}\{H(s)\} \tag{3.43}$$

Impulsní funkce se označuje $i(t)$. Je to odezva systému na vstupní signál $u(t)$ ve tvaru jednotkového – Diracova – impulsu při nulových počátečních podmínkách systému.



Obr. 8 Odezva systému na jednotkový – Diracův – impuls [6, str. 101]

Laplaceův obraz jednotkového impulsu je $U(s) = 1$. Laplaceův obraz výstupní funkce $y(t) = i(t)$, tedy $I(s)$ je ve tvaru:

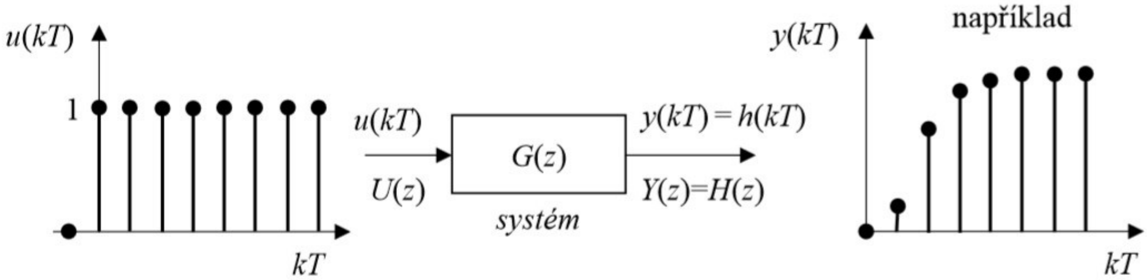
$$Y(s) = G(s)U(s) = G(s)1 = G(s) = I(s) \Rightarrow i(t) = L^{-1}\{I(s)\} \tag{3.44}$$

Pokud známe přechodovou (impulsní) funkcí, tak lze z ní odvodit impulsní (přechodovou) funkci za pomocí následujících vztahů:

$$i(t) = \frac{dh(t)}{dt}, h(t) = \int_0^t i(\tau)d\tau \tag{3.45}$$

3.2.3.2 Diskrétní verze přechodové funkce a charakteristiky, impulsní funkce a charakteristiky

Diskrétní přechodová funkce se označuje $h(kT)$. Je to odezva systému na vstupní signál $u(kt)$ ve tvaru diskrétního jednotkového – Heavisideova – skoku při nulových počátečních podmínkách systému.



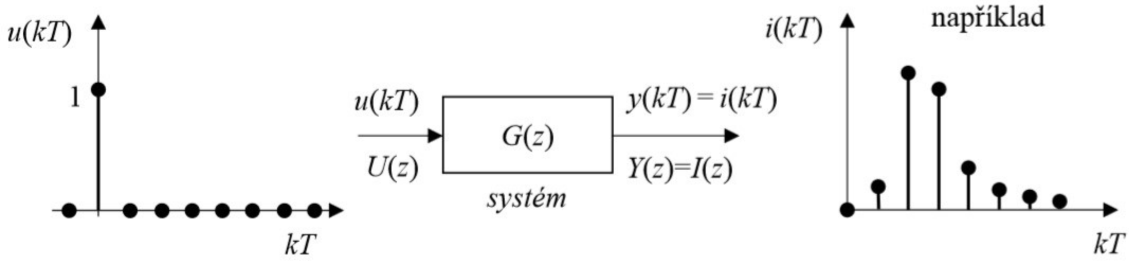
Obr. 9 Odezva systému na diskrétní jednotkový – Heavisideův – skok [4, str. 102]

Z-obraz diskrétního jednotkového skoku je $U(z) = \frac{z}{(z-1)}$. Z-obraz výstupní funkce

$y(kT) = h(kT)$, tedy $H(z)$ je ve tvaru:

$$Y(z) = G(z)U(z) = G(z)\frac{z}{z-1} = H(z) \Rightarrow h(kT) = Z^{-1}\{H(z)\} \qquad (3.46)$$

Diskrétní impulsní funkce se označuje $i(kT)$. Je to odezva systému na vstupní signál $u(kT)$ ve tvaru diskrétního jednotkového – Diracova – impulsu při nulových počátečních podmínkách systému.



Obr. 10 Odezva systému na diskrétní jednotkový – Diracův – impuls [6, str. 102]

Z-obraz diskrétního jednotkového impulsu je $U(z) = 1$. Z-obraz výstupní funkce

$y(kT) = i(kT)$, tedy $I(z)$ je ve tvaru:

$$Y(z) = G(z)U(z) = G(z)1 = I(z) \Rightarrow i(kT) = Z^{-1}\{I(z)\} \qquad (3.47)$$

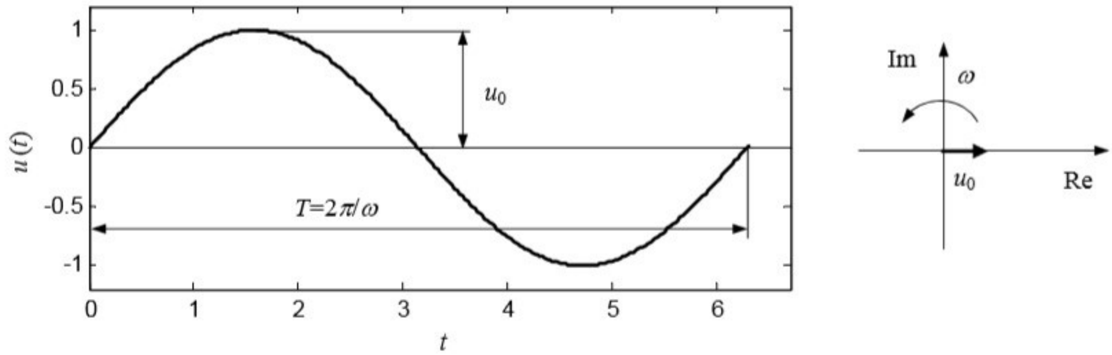
Pokud známe diskrétní přechodovou (impulsní) funkcí, tak lze z ní odvodit diskrétní impulsní (přechodovou) funkci za pomocí následujících vztahů:

$$i(kT) = h(kT) - h[(k-1)T], h(kT) = \sum_{j=0}^{k} i(jT) \qquad (3.48)$$

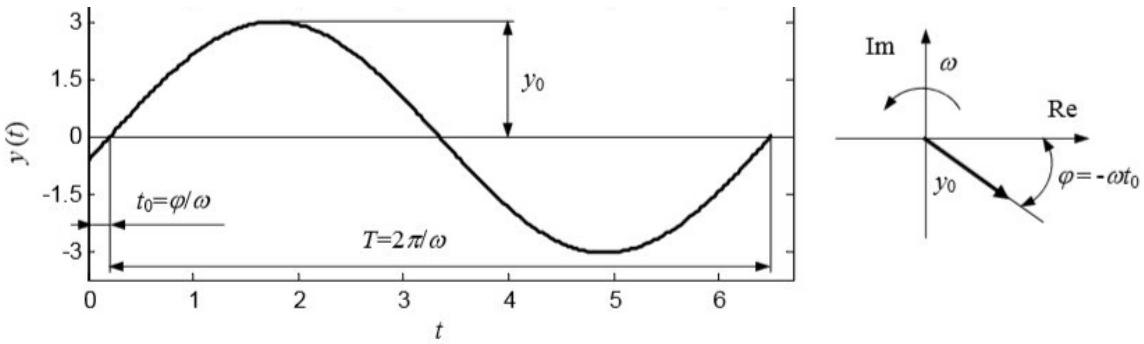
3.2.4 Frekvenční přenos a frekvenční charakteristiky

Frekvenční přenos je možné získat tak, že na vstup systému je přiveden harmonický signál. Z hlediska těchto harmonických signálů frekvenční pojmy zkoumají vlastnosti lineárních spojitých diskrétních systémů. Harmonický signál bývá nejčastěji ve tvaru: $u(t) = u_0 \sin\omega t$.

Pokud je vstupní signál v tomto tvaru, pak výstupní signál má tvar: $y(t) = y_0 \sin(\omega t + \varphi)$.



Obr. 12 Vstupní harmonický signál a jeho souvislost s rotujícím vektorem v komplexní rovině [4, str. 109]



Obr. 11 Výstupní harmonický signál a jeho souvislost s rotujícím vektorem v komplexní rovině [4, str. 109]

Na výstupu y vznikly harmonické kmity, které mají stejnou frekvenci ω, ale mají jinou amplitudu y_0 a vznikl tam určitý fázový posun φ v porovnání se vstupním signálem.

3.2.4.1 Frekvenční přenos

Frekvenční přenos je definovaný ve tvaru:

$$G(j\omega) = G(s)|_{s=j\omega} = \frac{Y(j\omega)}{U(j\omega)} = A(\omega)e^{j\varphi(\omega)} \tag{3.49}$$

kde: $Y(j\omega), U(j\omega)$ – Fourierovy obrazy vstupních a výstupních signálů, *A(ω* – amplituda zesílení, *φ(ω)* – fázový posun.

3.2.4.2 Frekvenční charakteristika v komplexní rovině

Frekvenční charakteristika je grafické vykreslení frekvenčního přenosu $G(j\omega)$ v komplexní rovině pro $\omega \in < 0, \infty)$. Jedná se o amplitudově-fázově frekvenční charakteristiku v komplexní rovině → Nyquistova křivka.

Pomocí úprav frekvenčního přenosu na složkový var komplexního čísla, lze získat frekvenční charakteristiku v komplexní rovině. Složkový tvar komplexního tvaru:

$$G(j\omega) = P(\omega) + jQ(\omega) = \mathrm{Re}G(j\omega) + j\mathrm{Im}G(j\omega) \quad (3.50)$$

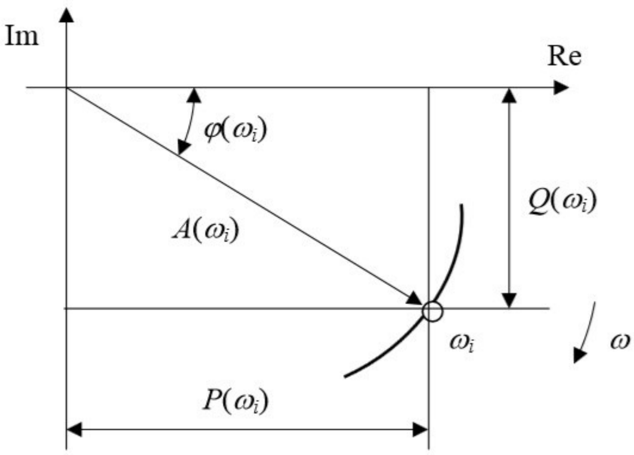
Nebo úprava na exponenciální tvar komplexního čísla:

$$G(j\omega) = A(\omega)e^{j\varphi(\omega)} = |G(j\omega)|e^{j\mathrm{arg}G(j\omega)} \quad (3.51)$$

kde platí:

$$A(\omega) = \mathrm{mod}G(j\omega) = |G(j\omega)| = \sqrt{P^2(\omega) + Q^2(\omega)} \quad (3.52)$$

$$\varphi(\omega) = \mathrm{arg}G(j\omega) = \mathrm{arctan}\frac{Q(\omega)}{P(\omega)} \quad (3.53)$$



Obr. 13 Zobrazení frekvenčního přenosu $G(j\omega)$
v komplexní rovině [4, str. 110]

3.2.4.3 Frekvenční charakteristiky v logaritmických souřadnicích

Tvar frekvenčního přenosu v exponenciálním tvaru:

$$G(j\omega) = A(\omega)e^{j\varphi(\omega)} \quad (3.54)$$

Frekvenční přenos v logaritmických souřadnicích lze vystihnout pomocí dvou charakteristik – Bodeho křivky.

- logaritmická amplitudová charakteristika
  $A(\omega)[dB] = |G(j\omega)|_{dB} = 20 \cdot \log_{10}|G(j\omega)| = 20 \cdot \log_{10} A(\omega) = 20 \cdot \log_{10}(\sqrt{P^2(\omega) + Q^2(\omega)}$ (3.55)

- logaritmická fázová charakteristika

$$\varphi(\omega) = \arg G(j\omega) = \arctan\frac{Q(\omega)}{P(\omega)} \tag{3.56}$$

Výpočet charakteristik složených systémů a jejich sestrojování se velmi zjednoduší díky zavedení logaritmických charakteristik. V logaritmických charakteristikách se násobení přenosů při sériovém zapojení systémů usnadňuje na sčítání charakteristik. Např. pro přenos:

$$G(j\omega) = G_1(j\omega)\cdot G_2(j\omega) = |G_1(j\omega)|e^{j\varphi_1(\omega)}|G_2(j\omega)|e^{j\varphi_2(\omega)} = |G_1(j\omega)|\cdot|G_2(j\omega)|\cdot e^{j(\varphi_1(\omega)+\varphi_2(\omega))} \tag{3.57}$$

Platí pro tento přenos:

$$20\log_{10}|\,G(j\omega)| = 20\log_{10}|G_1(j\omega)| + 20\log_{10}|G_2(j\omega)| \tag{3.58}$$

$$\varphi(\omega) = \arg G(j\omega) = \varphi_1(\omega) + \varphi_2(\omega) \tag{3.59}$$

4 PODPORY VÝUKY AUTOMATIZACE

Existuje několik materiálů pro podporu výuky automatizace. Tyto materiály byly vytvořeny profesory, vyučující nebo samotnými žáky. Bakalářské a diplomové práce věnované automatizaci si může najít každý student UTB FAI na *stagu* v sekci prohlížení kvalifikačních prací.

4.1 Vybrané statě automatizace

Tato práce byla vytvořena panem Ing. Pavlem Navrátilem, Ph.D., který je profesorem na Univerzitě Tomáše Bati ve Zlíně na Fakultě aplikované informatiky.

Tyto statě obsahují rozsáhlou problematiku automatizace. První část práce je věnována základům automatizace. Dále obsahuje logické řízení, lineární systémy, regulační obvody, typy řízených systémů, identifikaci řízených systémů a regulátor a metody jejich nastavení. Poslední část je věnována programu MATLAB a její nadstavby Simulinkem.

Práce obsahuje teoretickou část, ale je prokládána výpočtem a řešením příkladů. Většina kapitol obsahuje i postup řešení příkladů a úloh pomocí programu MATLAB. Jednotlivé příkazy používané v tomto programu, jsou zde popsány.

II. PRAKTICKÁ ČÁST

5 SPECIFIKACE HRY

Kapitola je věnována popisu hry, jejím parametrům a pravidlům.

5.1 Popis a parametry hry

Hra „Kvíz s otázkami z automatizace“ je určena pro všechny studenty, kteří se zabývají automatizací. Je to určeno pro jednoho hráče. Hra obsahuje jednak kvíz s otázkami, ale obsahuje i řešené příklady. Samostatná hra obsahuje 16 políček a každé políčko je věnováno určitému tématu. Po správném zodpovězení otázky, se hráči odkryje dané políčko. Pošpatném zodpovězení na danou otázku, hráči ubyde jeden život. Hráč má celkově tři životy. Pokud hráč během hry neztratí všechny 3 životy, zobrazí se mu obrázek, který je ukrytý pod políčky.

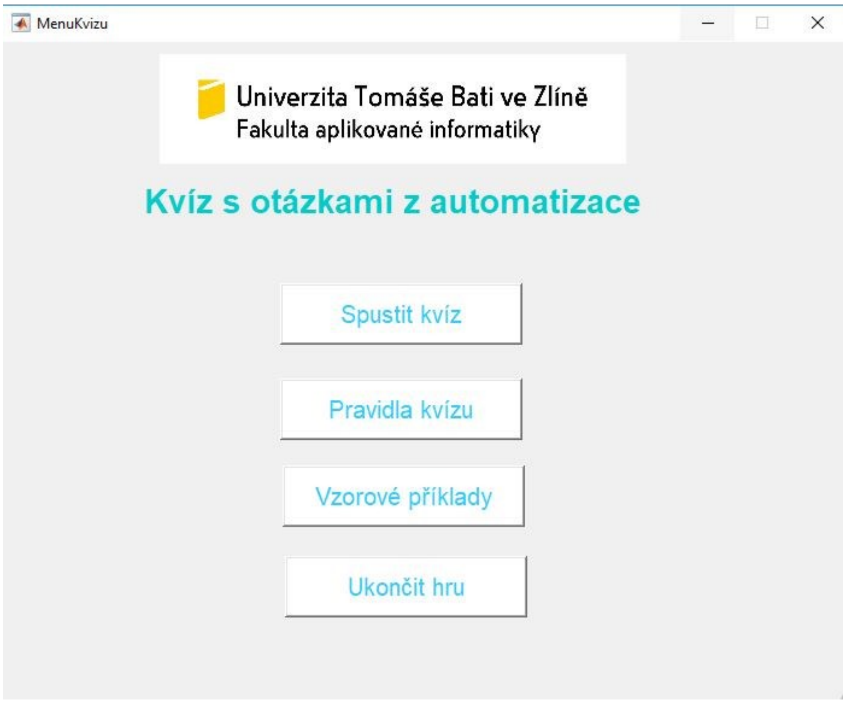
Každá otázka má 4 možnosti výběru odpovědi. Pouze jedna možnost je vždy správná. Některé otázky obsahují možnost „nevím“. Pokud hráč takto zodpoví na otázku, je to bráno jako špatná odpověď a hráč tak ztratí jeden život.

„Kvíz s otázkami z automatizace“ je určeno pro všechny studenty, kteří mají určité znalosti z oboru automatizace, aby znali správné odpovědi na otázky. Ve hře je i možnost si zopakovat jednotlivé kapitoly z předmětu automatizace v sekci „Vzorové příklady“.

Hru lze spustit pomocí příkazu *MenuKvizu* v oblasti *Command Window* v programu MATLAB.

5.2 Menu hry

Po spuštění příkazu *MenuKvizu* se zobrazí menu hry „Kvíz s otázkami z automatizace“. Student si zde může zvolit, zda si chce spustit hru, přečíst si pravidla kvízu, podívat se na vzorové vyřešené příklady, nebo jestli chce hru ukončit.



Obr. 14 Menu hry „Kvíz s otázkami z automatizace

5.3 Pravidla hry

Cílem hry je odpověď na všechny otázky schovávající se pod jednotlivými políčky a zobrazit skrývající se obrázek. Pouze na hráči záleží, v jakém pořadí si bude vybírat políčka s otázkami.

Každé políčko obsahuje určité otázky týkající se určitého tématu. Políčko zmizí po správném zodpovězení hráče na danou otázku. Pokud hráč zodpoví špatně, tak přijde o jeden život, ale políčko zmizí. Hráč má celkem tři životy. Po ztrátě všech životů dojde k ukončení hry a dojde k návratu do hlavního menu hry. Počet životů se zobrazují na pravém boku políček. Pokud hráč dokáže odpověď na všechny otázky, které se skrývají pod všemi 16 políčky a to bez ztráty všech životů, tak se mu zobrazí skrývající se obrázek a vyhrál hru.

Políčka s jednotlivými tématy obsahují většinou 10 otázek. Otázky jsou při každé hře vygenerovány náhodně. Hra obsahuje celkově 150 otázek. Otázky se týkají těchto okruhů:

- přímá L-transformace
- zpětná L-transformace
- vybrané vlastnosti L-transformace
- slovník Laplaceovy transformace
- rozklad funkce na parciální zlomky

- lineární diferenciální rovnice, lineární diferenční rovnice a přenos
- póly a nuly systému
- přechodová funkce a charakteristika, impulsní funkce a charakteristika
- frekvenční přenos a frekvenční charakteristiky.

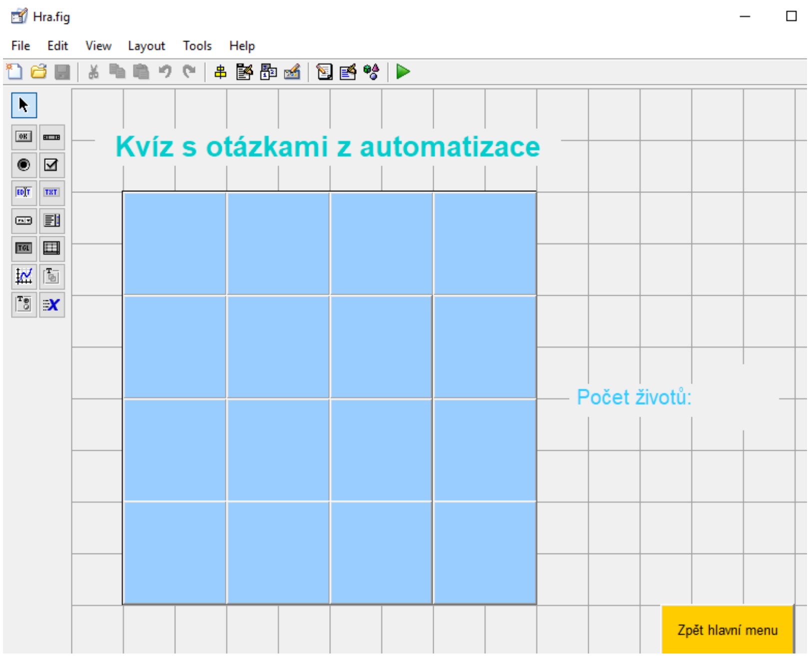
Obr. 15 Popis pravidel přímo ve hře

6 SPECIFIKACE HRY

Kapitola je zaměřená na popis hry z programátorské perspektivy.

6.1 Popis tvorby hry

Nejprve jsem navrhla strukturu hry. Hra byla vyvíjena v programu MATLAB verze R2016a. Pro samostatnou hru „Kvíz s otázkami z automatizace“ jsem vytvořila jeden hlavní GUI soubor. Pro menu, pravidla, vzorové příklady, jednotlivé otázky, atd. jsem vytvořila další GUI soubory. S těmito soubory jsou spjaty M-soubory. Vytvořila jsem jednu novou funkci *kontrola* pro zpřehlednění programu, aby se zdrojový kód několikrát neopakoval.

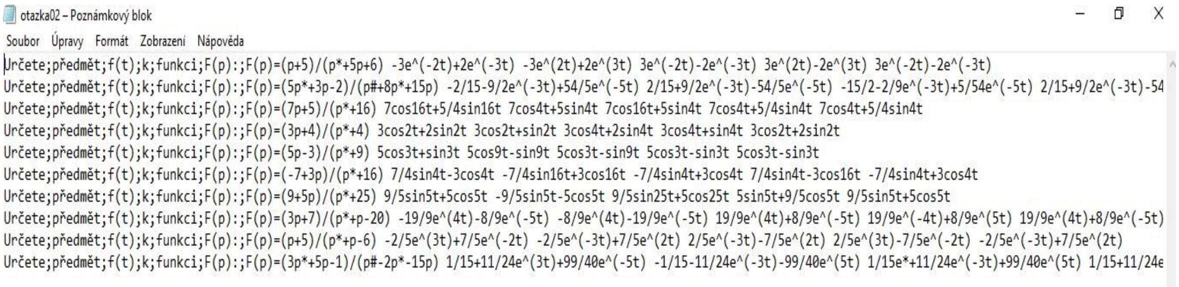


Obr. 16 Grafické okno hry vytvořené pomocí GUIDE

Jednotlivé M-soubory jsou mezi sebou propojeny tak, aby při skončení jednoho M-souboru se spustil druhý, a naopak. M-soubory jsou na sebe navzájem provázány.

6.2 Databáze otázek a odpovědí

Databáze otázek a odpovědí vznikla na základě mých zkušeností, které jsem získala během absolvování předmětu MATLAB a Simulink a také za pomoci doporučené literatury. Pro každé jednotlivé téma jsem vytvořila nový soubor *txt*. V těchto souborech jsou otázky zapsány jako: otázky, čtyři možnosti odpovědí, správné odpovědi. Otázka je zapsána jako jeden řetěz, kde místo mezerníku je použitý znak *;*.



otazka02 – Poznámkový blok

Soubor Úpravy Formát Zobrazení Nápověda

```
Určete;předmět;f(t);k;funkci;F(p):;F(p)=(p+5)/(p*+5p+6) -3e^(-2t)+2e^(-3t) -3e^(2t)+2e^(3t) 3e^(-2t)-2e^(-3t) 3e^(2t)-2e^(3t) 3e^(-2t)-2e^(-3t)
Určete;předmět;f(t);k;funkci;F(p):;F(p)=(5p*+3p-2)/(p#+8p*+15p) -2/15-9/2e^(-3t)+54/5e^(-5t) 2/15+9/2e^(-3t)-54/5e^(-5t) -15/2-2/9e^(-3t)+5/54e^(-5t) 2/15+9/2e^(-3t)-54
Určete;předmět;f(t);k;funkci;F(p):;F(p)=(7p+5)/(p*+16) 7cos16t+5/4sin16t 7cos4t+5sin4t 7cos16t+5sin4t 7cos4t+5/4sin4t 7cos4t+5/4sin4t
Určete;předmět;f(t);k;funkci;F(p):;F(p)=(3p+4)/(p*+4) 3cos2t+2sin2t 3cos2t+sin2t 3cos4t+2sin4t 3cos4t+sin4t 3cos2t+2sin2t
Určete;předmět;f(t);k;funkci;F(p):;F(p)=(5p-3)/(p*+9) 5cos3t+sin3t 5cos9t-sin9t 5cos3t-sin9t 5cos3t-sin3t 5cos3t-sin3t
Určete;předmět;f(t);k;funkci;F(p):;F(p)=(-7+3p)/(p*+16) 7/4sin4t-3cos4t -7/4sin16t+3cos16t -7/4sin4t+3cos4t 7/4sin4t-3cos16t -7/4sin4t+3cos4t
Určete;předmět;f(t);k;funkci;F(p):;F(p)=(9+5p)/(p*+25) 9/5sin5t+5cos5t -9/5sin5t-5cos5t 9/5sin25t+5cos25t 5sin5t+9/5cos5t 9/5sin5t+5cos5t
Určete;předmět;f(t);k;funkci;F(p):;F(p)=(3p+7)/(p*+p-20) -19/9e^(4t)-8/9e^(-5t) -8/9e^(4t)-19/9e^(-5t) 19/9e^(4t)+8/9e^(-5t) 19/9e^(-4t)+8/9e^(5t) 19/9e^(4t)+8/9e^(-5t)
Určete;předmět;f(t);k;funkci;F(p):;F(p)=(p+5)/(p*+p-6) -2/5e^(3t)+7/5e^(-2t) -2/5e^(-3t)+7/5e^(2t) 2/5e^(-3t)-7/5e^(2t) 2/5e^(3t)-7/5e^(-2t) -2/5e^(-3t)+7/5e^(2t)
Určete;předmět;f(t);k;funkci;F(p):;F(p)=(3p*+5p-1)/(p#-2p*-15p) 1/15+11/24e^(3t)+99/40e^(-5t) -1/15-11/24e^(-3t)-99/40e^(5t) 1/15e*+11/24e^(-3t)+99/40e^(5t) 1/15+11/24e
```

Obr. 17 Ukázka souboru s databází otázek, možností výběru a správné odpovědi

Před zobrazením samostatné otázky se nahrají otázky z *txt.* souboru do proměnné *fid.* Po otevření souboru s databází otázek díky příkazu *fopen* jsem pomocí funkce *strsplit* rozdělila řetězec obsahu souboru. Vytvořila jsem si novou proměnnou *C*, která obsahuje náhodnou otázku ze sady všech otázek. V otázce, kde místo mezerníku je použitý znak *;*, jsem použila funkci *strrep*, která vyhledává znak *;* a nahrazuje tento znak mezerníkem. Této funkce jsem využila i později pro zobrazování speciálních znaků, jako např.: ^{2}. V každém souboru otázek, je přibližně deset otázek, které se před zobrazením otázky náhodně vyberou pomocí funkce *randi*. Nakonec soubor s databází otázek je zavřen. Ukázka kódu je uvedena níže.

```
fid = fopen('otazka01.txt');
radek = 0;

while ~feof(fid)
    data = fgetl(fid);
    radek = radek+1;
    NaDruhou = char(178);
    NaTreti = char(179);
    pom = strsplit(data); %funkce pro rozdělení řetězce
    for i = 1:6
        C{radek,i} = pom(i);
        C{radek,i} = C{radek,i}{1};
    end
    %funkce pro nahrazení znaménka - na mezeru
    C{radek,1} = strrep(C{radek,1},';',' ');
    for i = 1:6
        C{radek,i} = strrep(C{radek,i},'*',NaDruhou);
        C{radek,i} = strrep(C{radek,i},'#',NaTreti);
    end
end

setappdata(0,'otazky',C);
radek = randi(10);
fclose(fid);
```

6.3 Hra.m

Tento M-soubor slouží jako hlavní funkce programu. Pomocí tohoto souboru se spouští celá hra. Hlavní funkcí je *Hra_OpeningFcn*, jejímž úkolem je načtení a nastavení obrázku a nastavení viditelnosti objektů, které se mají v okně zobrazit nebo které mají být neviditelné.

Nastavuje se zde globální proměnná pro životy *zivoty* a globální pomocná proměnná *pomocna* pro počet zobrazených otázek. Nastavuje se zde obrázek pomocí funkce *imshow* (*imshow(imread('obrazek.jpg'),'Parent',handles.obrazek)*). Tento soubor obsahuje další funkce, které jsou popsány níže. Obrázek, který je použitý v programu, byl vytvořen v jiném programu – Gimp 2.

otazka1_Callback – funkce, kde se nastavuje objekt na neviditelný. Po zavolání této funkce, se objekt/políčko označováno jako *otazka1* nastaví na neviditelné a obdobně i *figurel* (okno hry se nastaví jako neviditelný). Neviditelnost se nastaví pomocí funkce *set*, kdy se jako první najde požadovaný objekt pomocí funkce *findobj* a poté se zavolá funkce *set.* Tato funkce je stejná i u dalších objektů (od *otazka1_Callback* až po *otazka16_Callback*). Ukázka kódu je uvedena níže.

```
set(handles.otazka1,'Visible','off');
fig = findobj('Type','Figure','Tag','figure1');
set(fig,'Visible','off');
otazka01;
```

ZpetHlavniMenu_Callback – funkce, která se ptá hráče, zda opravdu chce ukončit rozehranou hru. Po zvolení objektu *zpět hlavní menu* se zobrazí dialogové okno s otázkou, zda chce uživatel opustit hru. Toto dialogové okno se zobrazuje pomocí funkce *questdlg*. Pokud hráč zodpoví, že chce ukončit hru, tak se zavře okno s kvízem a zobrazí se menu hry. Jestliže zodpoví „Ne“, tak bude pokračovat ve hře. Ukázka kódu je znázorněna níže.

```
konec = questdlg('Jste si jisti, zda chcete ukončit hru?', ...
    'Ukončení hry', ...
    'Ano','Ne','Ne');
if strcmp(konec, 'Ano')
    delete(Hra);
    MenuKvizu;
else
    set(handles.ZpetHlavniMenu,'Visible','on');
end
```

figure1_CloseRequestFcn – funkce, která se ptá hráče, zda opravdu chce ukončit rozehranou hru. Po označení křížku v pravém horním rohu se zobrazí dialogové okno s otázkou, zda chce uživatel opustit hru. Toto dialogové okno se zobrazuje pomocí funkce *questdlg*. Pokud hráč zodpoví, že chce ukončit hru, tak se zavře okno s kvízem a zobrazí se menu hry. Jestliže zodpoví „Ne“, tak bude pokračovat ve hře.

6.4 otazka01.m

Tento M-soubor je stejný jako další M-soubory *otazka02.m* až *otazka16.m*. V těchto souborech dochází k otevření a načtení jednotlivých otázek ze souborů s databází otázek, které jsou jednotlivě uloženy v souborech typu *txt*. Otázky jsou uloženy v souborech podle daných témat. Tyto soubory budou moci být později využívány z důvodu přidání dalších otázek. Tento soubor obsahuje několik funkcí.

otazka01_OpeningFcn – funkce, ve které dochází k otevření souboru s databází otázek a pozdější načtení těchto dat. Způsob jak se načítají data a jak dochází k náhodnému výběru otázek, bylo zmíněno výše v kapitole Databáze otázek a odpovědí. Po získání dat ze souboru se jednotlivé možnosti odpovědí nastaví do jednotlivých objektů/tlačítek a to pomocí funkce *set*. V této funkci se nastavuje i správná odpověď do proměnné *spravnaodpoved,* která bude sloužit pro vyhodnocení otázky. Uložení správné odpovědi do proměnné funkce *setappdata*, která je uvedena níže.

```
C = getappdata(0,'otazky');
set(handles.otazka,'String',C{radek,1});
set(handles.moznostA,'String',C{radek,2});
set(handles.moznostB,'String',C{radek,3});
set(handles.moznostC,'String',C{radek,4});
set(handles.moznostD,'String',C{radek,5});

spravnaodpoved = C{radek,6};
setappdata(0,'odpoved',spravnaodpoved);
```

moznostA_Callback – funkce, ve které se získávají hodnoty globálních proměnných *zivoty* a *pomocna*. Nastavuje se zde výsledek otázky do proměnné *vysledek* a výběr možnosti (v tomto případě se jedná o objekt/tlačítko označené jako *moznostA*) hráčem do proměnné *vyber*. Ta se pak ve funkci *kontrola(vysledek,vyber,zivoty,pomocna)* vyhodnocuje. Funkce kontrola bude popsána níže v kapitole Kontrola.m. Po výběru tohoto objektu se okno s otázkou zavře. Funkce *moznostB_Callback, moznostC_Callback a moznostD_Callback* jsou podobné, jak je uvedeno níže v kódu.

```
global zivoty;
global pomocna;
zivoty = getappdata(0,'zivoty');
pomocna = getappdata(0,'pomocna');
vysledek = getappdata(0,'odpoved');
vyber = get(handles.moznostA,'String');
closereq;
kontrola(vysledek,vyber,zivoty,pomocna);
```

figure1_CloseRequestFcn - funkce, díky které se zobrazí zprávu, ve kterém je hráč informován o tom, že na zobrazenou otázku musí odpovědět a vybrat si jednu z nabízených možností. Zpráva je zobrazována pomocí funkce *msgbox*, která je uvedena níže.

```
nelze = msgbox('Neodpověděl jsi na otázku! Musíš na ni odpovědět...');
uiwait(nelze);
```

6.5 kontrola.m

Tento M-soubor byl mnou vytvořený z důvodu lepší orientace ve zdrojovém kódu. Část této funkce by se totiž mnohokrát opakovala v programu. Proto jsem využila vytvoření vlastní funkce *kontrola*.

V první části této funkce kontroluji globální proměnnou *vysledek* (v proměnné je uložená správná odpověď) s globální proměnnou *vyber* (v proměnné je uložená možnost zvolená uživatelem). Využívám funkce *if*. Jestliže zvolená možnost hráčem, bude stejná jako správná odpověď, tak se globální proměnná *zivoty* (v proměnné je uložená hodnota životů) nezmění. Pokud *vysledek* se nebude rovnat *vyber*, tak proměnná *zivoty* se sníží o hodnotu 1. V obou případech navyšuji proměnnou *pomocna*, která slouží pro vyhodnocování celkového počtu zodpovězených otázek. Po vyhodnocení otázky, se nastaví proměnná *zivoty* a tato hodnota se zobrazí v příslušném objektu.

V další části se kontroluje, jestli nedošel uživateli počet životů. Pokud se bude hodnota proměnné *zivoty* rovnat nule, tak pomocí funkce *msgbox* vyskočí na obrazovce zpráva o tom, že daný hráč prohrál. Zavře se okno s kvízem a otevře se okno menu hry.

V poslední části funkce *kontrola* se zjišťuje, zda hráč už nezodpověděl na všechny otázky. K tomu slouží globální proměnná *pomocna*. Hodnota této proměnné se navyšuje po každém zodpovězení otázky. Poté co hodnota této proměnné se bude rovnat šestnácti, tak vyskočí okno se zprávou, že uživatel vyhrál hru. Využívá se opět funkce *msgbox*. Zdrojový kód je uveden níže.

```
function kontrola(vysledek,vyber,zivoty,pomocna)

if  strcmp(vysledek,vyber)
    zivoty = (zivoty+0);
    setappdata(0,'zivoty',zivoty);
    pomocna=(pomocna+1);
    setappdata(0,'pomocna',pomocna);
else
    zivoty = (zivoty-1);
    setappdata(0,'zivoty',zivoty);
    pomocna=(pomocna+1);
    setappdata(0,'pomocna',pomocna);
end

zivotPom = findobj('Tag','PocetZivotu');
set(zivotPom,'String',zivoty);

if zivoty == 0
   Prohra = msgbox('Došel Ti počet životů. Bohužel jsi prohrál.');
   uiwait(Prohra);
   delete(Hra);
   MenuKvizu;
else
   fig = findobj('Type','Figure','Tag','figure1');
   set(fig,'Visible','on');
end

if pomocna == 16
   vyhra = msgbox('Vyhrál jsi!');
   uiwait(vyhra);
   delete(Hra);
   MenuKvizu;
   else
   fig = findobj('Type','Figure','Tag','figure1');
   set(fig,'Visible','on');
end
```

6.6 MenuKvizu.m

Tento soubor slouží spíše pro přepínání mezi jednotlivými okny – hra, pravidla hry, řešené příklady, atd. Obsahuje funkce: *MenuKvizu_OpeningFcn*, *SpustitKviz_Callback, UkoncitHru_Callback* atd. Pro zavření okna a otevření druhého okna je použita funkce *close*.

7 HRA Z POHLEDU UŽIVATELE

Pomocí příkazu *MenuKvizu* v programu MATLAB v příkazovém okně *Command Window* spustí uživatel hru. Aplikace se spustí v jakékoliv verzi MATLABu. Po zadání příkazu se spustí hlavní okno hry - menu.

7.1 Menu hry

V menu hry si může uživatel vybrat, zda chce spustit hru, přečíst si pravidla hry, podívat se na vzorové řešené příklady, či hru ukončit.

Uživatel může provádět výběr pouze za pomocí myši. Uživatel najede myší na dané tlačítko a klikne na něj.

7.2 Pravidla hry

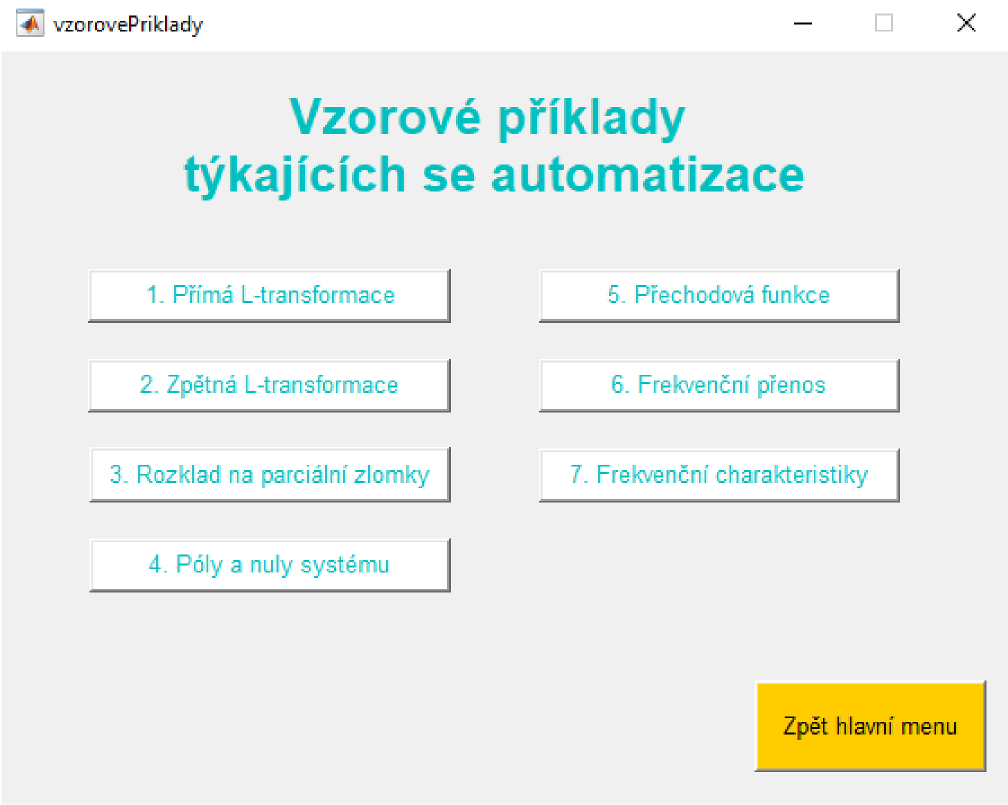
Při výběru *Pravidla hry* se uživateli zobrazí nové okno, kde jsou popsány pravidla hry. Zde je tlačítko *Zpět hlavní menu*, kterým se uživatel dostane nazpět do hlavního menu. Ovládání tohoto tlačítka opět funguje pouze pomocí myší a to najetím na dané tlačítko a kliknutím na něj. Při kliknutí na křížek v horním pravém rohu se okno zavře a ukončí se celá hra.

7.3 Ukončit hru

Při výběru *Ukončit hru*, se hlavní okno hry zavře. Ukončit hru lze i v hlavním menu pomocí křížku v pravém horním rohu.

7.4 Vzorové příklady

Pokud uživatel zvolí *Vzorové příklady*, tak se zobrazí nové okno, kde uživatel má možnost shlédnout postup úlohy a to z několika vypracovaných úloh.

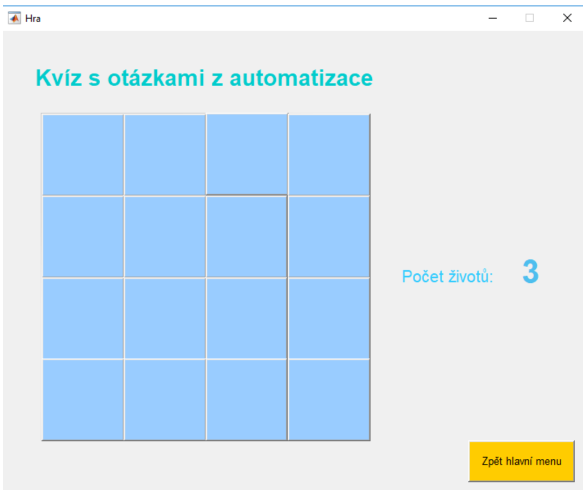


Obr. 18 Vzorové příklady týkající se problematiky automatizace

Ovládání je stejné jak u předchozích oken pomocí myši. Po vybrání z několika možností vzorových příkladů se otevře nové okno, kde se řeší daný problém. Pomocí tlačítka *Zpět hlavní menu* se uživatel dostane nazpět do menu hry.

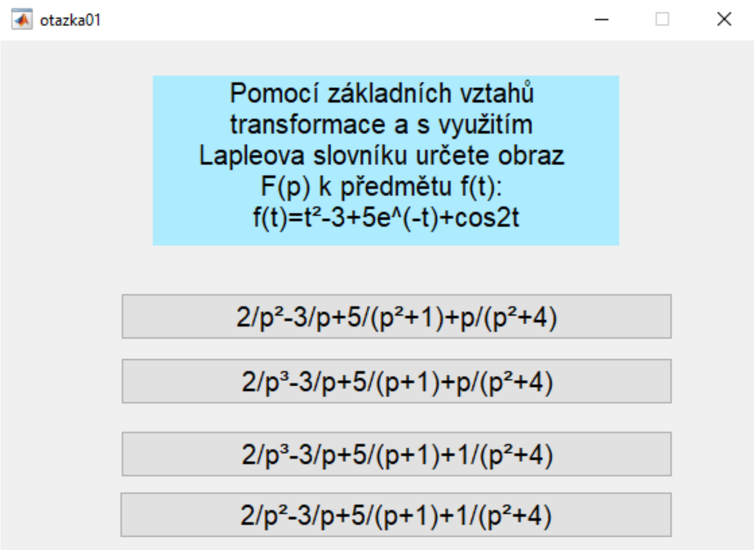
7.5 Hra

Při výběru této volby se uživateli zobrazí okno s nově nahranou hrou.



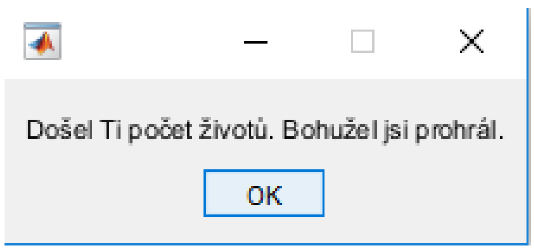
Obr. 19 Okno hry

V pravé části okna je zobrazen počet životů. Pak už je jen na uživateli, které políčko si zvolí. Po označení políčka pomocí myší a to najetím nad dané políčko a kliknutím myši, se zobrazí hráči náhodná otázka. Hráč může vybírat ze čtyř možností. Pouze jedna možnost je vždy správná.



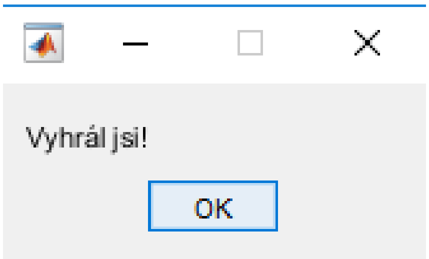
Obr. 20 Okno s vyobrazenou otázkou

Po výběru ze čtyř možností se okno s otázkou zavře a otevře se opět hra s jednotlivými poli. Pokud hráč zodpověděl na danou otázku špatně, tak se mu v pravé části okna odečte život. Po zodpovězení otázky se hráči vždy odkryje část obrázku pod políčky. Pokud hráč přijde o všechny životy, tak se zobrazí okno, ve kterém je hráč informován o prohře. Po označení tlačítka *OK* pomocí myši se hráč dostane nazpět do menu hry.

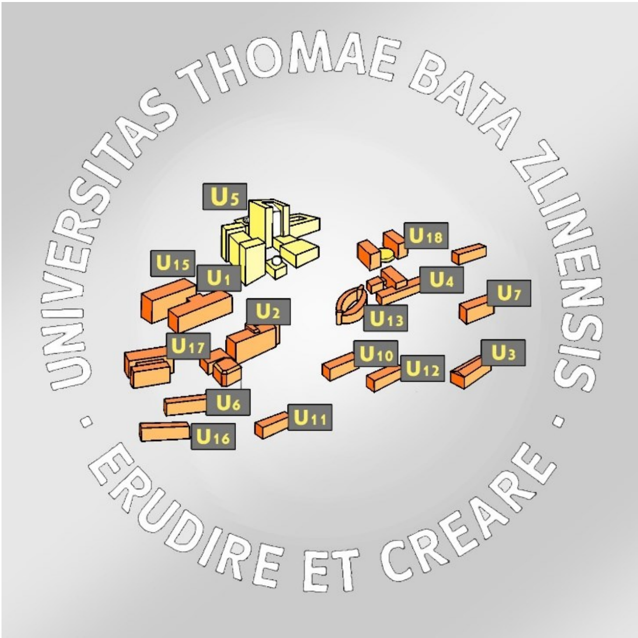


Obr. 21 Informační okno – prohra

Pokud hráč zodpoví na všechny otázky správě bez ztráty všech životů, tak se hráči odkryje skrytý obrázek a zobrazí se informační okno, ve kterém je hráč informován o výhře. Zároveň se mu odkryj obrázek, který byl skrytý pod políčky. Po označení tlačítka *OK* pomocí myši se hráč dostane nazpět do menu hry.

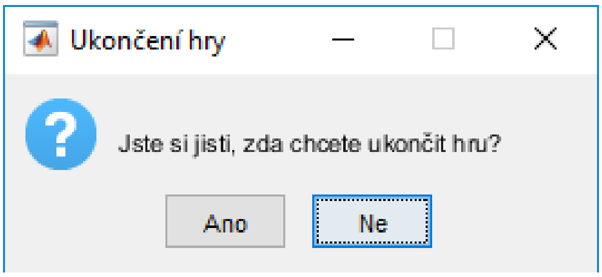


Obr. 22 Informační okno- výhra



Obr. 23 Skrytý obrázek pod políčky

V případě, že hráč zvolí tlačítko *Zpět hlavní menu*, tak se zobrazí dialogové okno, které se hráče ptá, zda opravdu chce opustit hru. Pokud zvolí *Ano*, tak se okno s hrou zavře a otevře se menu hry. Po zvolení tlačítka *Ne*, se dialogové okno zavře a zobrazí se okno s rozehranou hrou.



Obr. 24 Tázací dialogové okno pro ukon-
čení hry

Jestliže hráč během hry zvolí pomocí myši tlačítko křížek v pravém horním rohu, tak se zobrazí dialogové okno, které se hráče ptá, zda opravdu chce opustit hru jako je na Obr. 24.

8 ŘEŠĚNÉ ÚLOHY Z AUTOMATIZACE

Kapitola obsahuje řešené úlohy z automatizace, které se řeší výpočetně a pomocí programu MATLAB a pomocí Simulinku. Tyto řešené příklady jsou obsaženy ve vytvořené aplikaci „Kvíz s otázkami z automatizace“.

8.1 Přímá L-transformace

Pro řešení těchto typů úloh, kdy se hledá obraz *F(s)* k předmětu *f(t)*, se využívá základních vztahů transformace s využitím slovníku Laplaceovy transformace.

Příklad: $f(t) = 2 + 4te^{-2t} - 5t^2e^{-3t}$

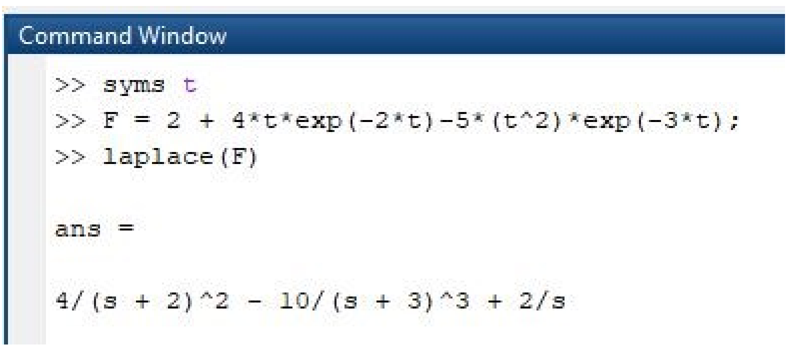
Využiji se linearity pro vztah: $e^{at}f(t) = F(s-a), s_1 = -2, s = -3$,

a vzorec: $1 = \frac{1}{s}, t = \frac{1}{s^2}, t^2 = \frac{2}{s^3}$.

Dosadí se a výsledek bude: $F(s) = \frac{2}{s} + \frac{4}{(s+2)^2} - \frac{2 \cdot 5}{(s+3)^3}$.

Řešení příkladu pomocí příkazu *laplace* v programu MATLAB:

```
syms t
F = 2+4*t*exp(-2*t)-5*(t^2)*exp(-3*t);
laplace(F)
```



Obr. 25 Využití příkazu *laplace* pro hledání obrazu *F(s)*

8.2 Zpětná L-transformace

Pro řešení těchto typů úloh se nejčastěji využívá rozklad na parciální zlomky s využitím slovníku Laplaceovy transformace, kdy se určuje předmět *f(t)* k funkci *F(s)*.

Příklad: $F(s) = \frac{s+3}{s^2+3s-4}$

Rovnice $s^2 + 3s - 4 = (s+4) \cdot (s-1)$ má dva reálné kořeny: $s_1 = -4$ a $s_2 = 1$, tedy se může napsat: $F(t) = \frac{s+3}{s^2+3s-4} = \frac{A}{s-1} + \frac{B}{s+4}$.

Rovnici lze vynásobit jmenovatelem a pro neurčité koeficienty A a B se dostane rovnice:

$s + 3 = A(s + 4) + B(s - 1)$.

Dosadí se hodnoty kořenů do rovnice:

$s = -4{:}\ -4 + 3 = A(-4 + 4) + B(-4 - 1) \Rightarrow B = \frac{1}{5}$;

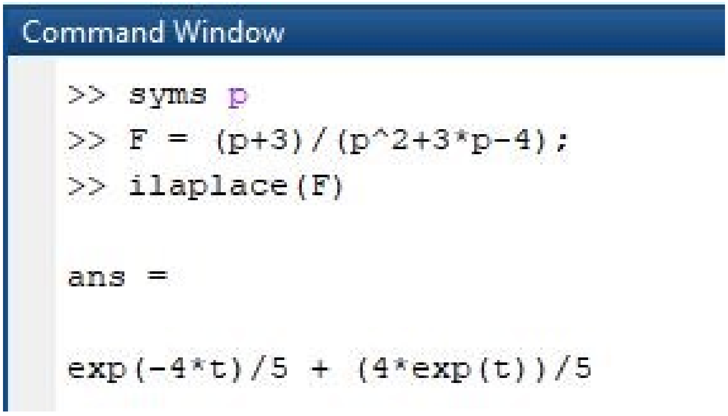
$s = 1{:}\ 1 + 3 = A(1 + 4)B(1 - 1) \Rightarrow A = \frac{4}{5}$.

Zjištěné neurčité koeficienty se dosadí do rovnice: $F(s) = \frac{4}{5} \cdot \frac{1}{(s-1)} + \frac{1}{5} \cdot \frac{1}{s+4}$.

Výsledkem je: $f(t) = \frac{4}{5}e^{t} + \frac{1}{5}e^{-4t}$.

Řešení příkladu pomocí příkazu *ilaplace* v programu MATLAB:

```
syms s
F = (s+3)/(s^2+3*s-4);
ilaplace(F)
```



Obr. 26 Využití příkazu *ilaplace* pro určení předmětu *f(t)*

8.3 Rozklad na parciální zlomky

Příklad: $F(s) = \frac{3s^2-2s+5}{s^3+s^2-6s}$

Rovnice $s^3 + s^2 - 6s = s(s^2 + s - 6) = s(s + 6) \cdot (s - 1)$ má tři reálné kořeny:

$s_1 = 0, s_2 = -1$ a $s_3 = 1$,

tedy se může napsat: $F(s) = \frac{3s^2-2s+5}{s(s+6)\cdot(s-1)} = \frac{A}{s} + \frac{B}{(s+6)} + \frac{C}{(s-1)}$.

Rovnice se vynásobí jmenovatelem a pro neurčité koeficienty A, B a C se dostane rovnice:

$3s^2 - 2s + 5 = A(s+6)\cdot(s-1) + Bs\cdot(s-1) + Cs\cdot(s+6).$

Dosadí se hodnoty kořenů do rovnice a získá se:

$s = 0{:}\ 5 = A(0+6)(0-1) \Rightarrow A = 0,$

$s = -6{:}\ 3\cdot(-6)^2 - 2\cdot(-6) + 5 = B(-6)\cdot(-6-1) \Rightarrow B = -\frac{24}{7},$

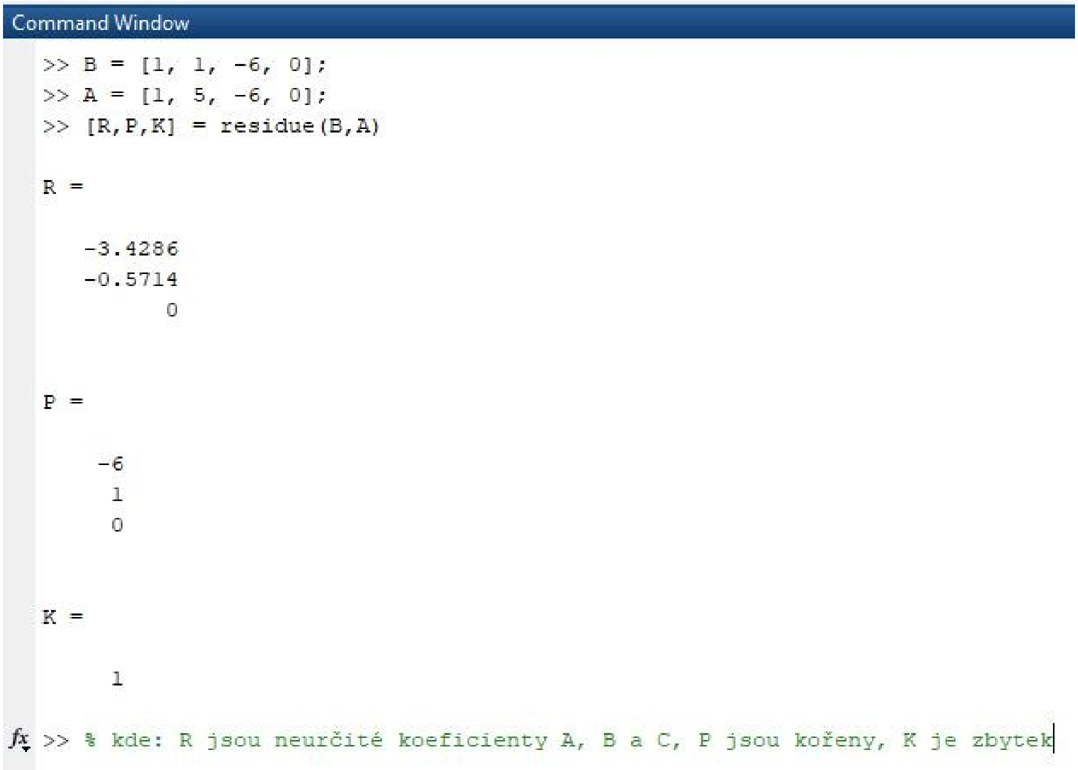
$s = 1{:}\ 3\cdot 1^2 - 2\cdot 1 + 5 = C(1+6) \Rightarrow C = -\frac{4}{7}.$

Zjištěné neurčité koeficienty se dosadí do rovnice:

$F(t) = -\frac{24}{7}\frac{1}{(s+6)} - \frac{4}{7}\frac{1}{(s-1)}.$

Řešení příkladu pomocí příkazu *residue* v programu MATLAB:

```
B = [1, 1, -6, 0];
A = [1, 5, -6, 0];
 [R,P,K] = residue(B,A)
```



Obr. 27 Využití příkazu *residue* pro rozklad na parciální zlomky

8.4 Nuly a póly systému

Příklad: $G(s) = \frac{s^2-3s+2}{s^3-3s^2+5s}$

Rovnici lze zapsat jako: $\frac{s^2-3s+2}{s^3-3s^2+5s} = \frac{(s-1)\cdot(s-2)}{s(s^2-3s+5)}$.

Kořeny rovnice: $s^2 - 3s + 2 = (s-1)\cdot(s-2)$ jsou $s_1 = 1$ a $s_2 = 2$. Zároveň tyto kořeny jsou nulami systému.

Nuly systému jsou $n_1 = 1$ a $n_2 = 2$.

Pomocí diskriminantu lze získat kořeny rovnice: $s(s^2 - 3s + 5)$,

$D = (-3)^2 - 4 \cdot 1 \cdot 5 = 9 - 20 = -11$,

$\sqrt{D} = \sqrt{-11} = \pm\sqrt{11}i$,

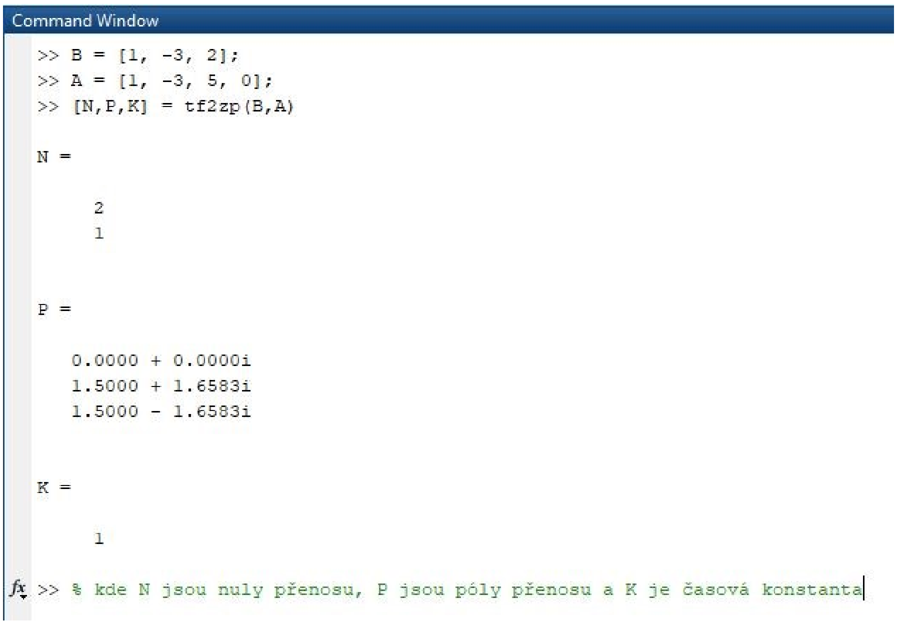
$s_{1,2} = \frac{3\pm\sqrt{11}i}{2} = 1.5 \pm 1.658i$.

Zjištěné kořeny jsou zároveň i póly zadané přenosu.

Póly přenosu $p_1 = 0, p_2 = 1.5 + 1.658i$ a $p_3 = 1.5 - 1.658i$.

Řešení příkladu pomocí příkazu *tf2zp* v programu MATLAB:

```
B = [1, -3, 2];
A = [1, -3, 5, 0];
[N,P,K] = tf2zp(B,A)
```


```
Command Window
>> B = [1, -3, 2];
>> A = [1, -3, 5, 0];
>> [N,P,K] = tf2zp(B,A)

N =

     2
     1


P =

   0.0000 + 0.0000i
   1.5000 + 1.6583i
   1.5000 - 1.6583i


K =

     1

>> % kde N jsou nuly přenosu, P jsou póly přenosu a K je časová konstanta
```

Obr. 28 Využití příkazu *tf2zp* pro zjištění nul a pólů systému

8.5 Přechodová funkce

Příklad: $f(s) = \frac{3s-1}{2s+1}$

Zadaná rovnice se vynásobí $\frac{1}{s}$ pro získání přechodové funkce: $\frac{3s-1}{2s+1} \cdot \frac{1}{s} = \frac{3s-1}{s(2s+1)} = \frac{3s-1}{2s^2+s}$.

Následně se využije rozkladu na parciální zlomky: $\frac{3s-1}{s(2s+1)} = \frac{A}{s} + \frac{B}{(2s+1)}$. Kořeny rovnice $s(2s+1)$ jsou: $s_1 = 0$ a $s_2 = -\frac{1}{2}$.

Rovnice se vynásobí jmenovatelem a pro neurčité koeficienty A a B se získá rovnice

$3s - 1 = A(2s+1) + Bs.$

Dosadí se hodnoty kořenů do rovnice:

$s = 0: 3 \cdot 0 - 1 = A(2 \cdot 0 + 1) \Rightarrow A = -1,$

$s = -\frac{1}{2}: A\left(2\left(-\frac{1}{2}\right) + 1\right) + B\left(-\frac{1}{2}\right) \Rightarrow B = 5.$

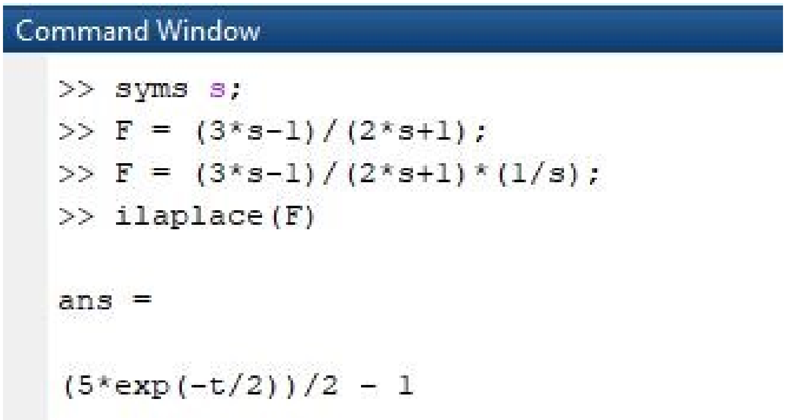
Zjištěné neurčité koeficienty se dosadí do rovnice a dostane se: $F(s) = -\frac{1}{(2s+1)} + 5s,$

následně se využije přímé L-transformace pomocí slovníku Laplaceovy transformace.

Přechodová funkce bude ve tvaru: $F(s) = 5 - \frac{1}{2} e^{-\frac{1}{2}t}.$

Řešení příkladu v programu MATLAB s využitím příkazu *ilaplace*:

```
syms s
F = (3*s-1)/(2*s+1);
F = (3*s-1)/(2*s+1)*(1/s);
ilaplace(F)
```



Obr. 29 Postup výpočtu přechodové funkce pomocí příkazů v MATLABu

8.5.1 Řešení přechodové charakteristiky v programu MATLAB – Simulink

Pro řešení přechodové charakteristiky v nadstavbě programu MATLAB v Simulinku, se využívají bloky *Step*, *Transfer Fcn*, *Mux* a *Scope*. Schéma zapojení je uvedeno níže na Obr. 30.

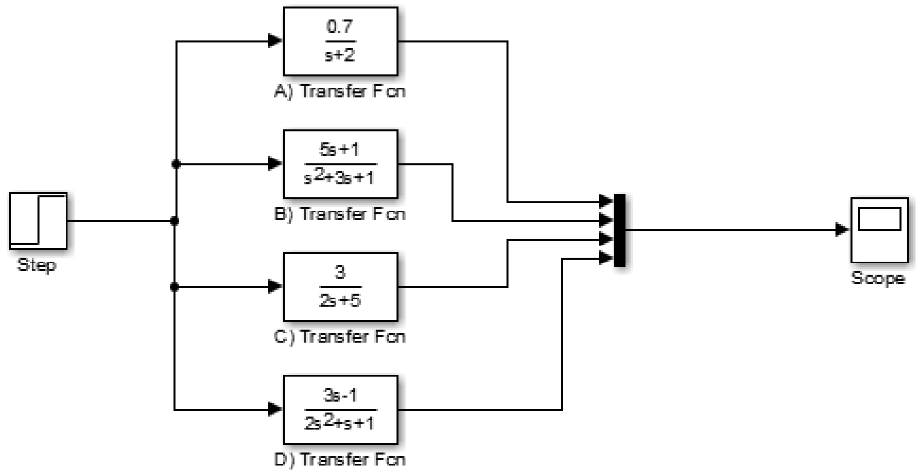
Příklady řešené pomocí Simulinku:

$A)\ G(s) = \frac{0.7}{s+2}$,

$B)\ G(s) = \frac{5s+1}{s^2+3s+1}$,

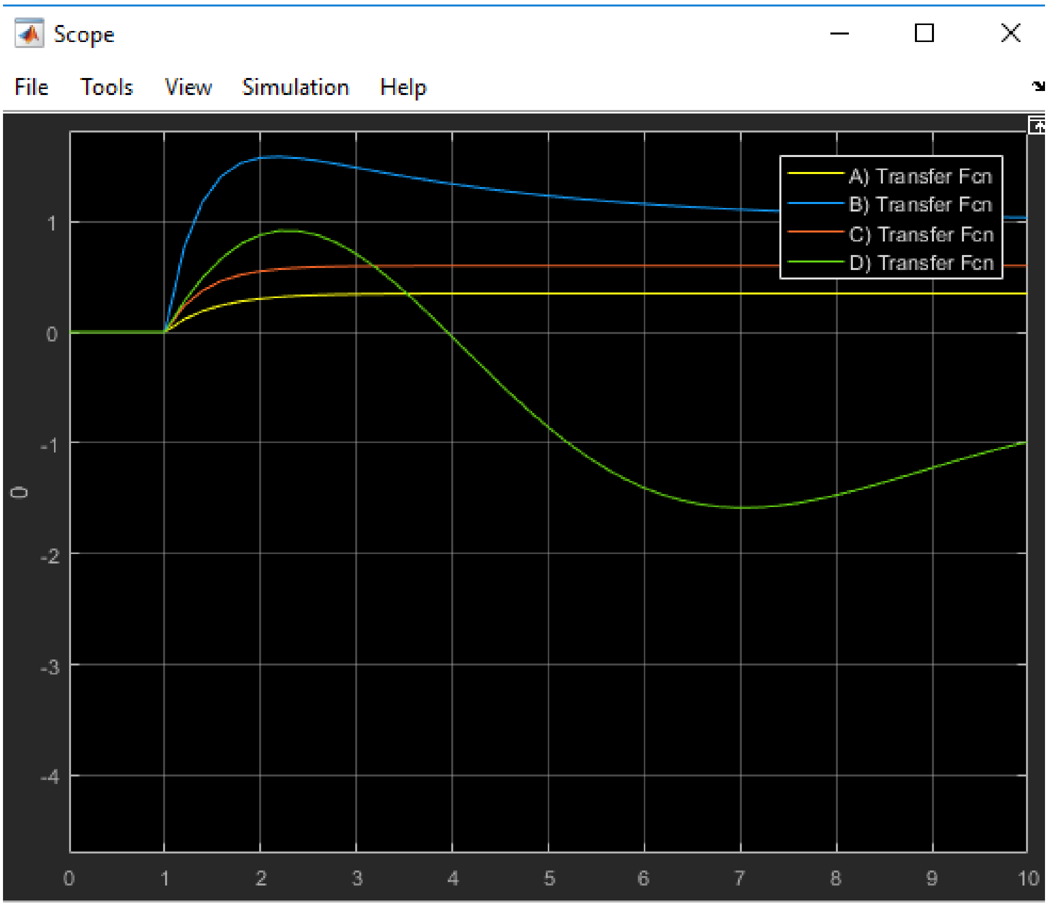
$C)\ G(s) = \frac{3}{2s+5}$,

$D)\ G(s) = \frac{3s-1}{2s^2+s+1}$.



Obr. 30 Schéma zapojení jednotlivých bloků pomocí Simulinku

Výstupem jsou grafy na Obr. 31, které znázorňují průběh přechodových charakteristik.



Obr. 30 Výstup přechodových charakteristik pomocí Simulinku

Pro uživatele, kteří by měli zájem o vykreslení přechodových charakteristik jiných příkladů, než jsou uvedeny zde v této práci, je umožněno v aplikaci „Kvíz s otázkami z automatizace“ ve vzorových příkladech. Pomocí tlačítka *Řešení v Simulinku* se uživateli otevře nové okno se Simulinkem. V okně už jednoduše může měnit jednotlivé funkce pomocí dvojitým kliknutím myši na objekt *Transfer Fcn.*

8.6 Frekvenční přenos

Příklad: $F(s) = \frac{1}{s^2+3s+2}$

Pro určení frekvenčního přenosu se využije dosazení $j\omega$ za s:

$$F(s) = \frac{1}{(j\omega)^2 + 3j\omega + 2} = \frac{1}{2 - \omega^2 + 3j\omega} = \frac{1}{2 - \omega^2 + 3j\omega} \cdot \frac{2 - \omega^2 - 3j\omega}{2 - \omega^2 - 3j\omega} =$$

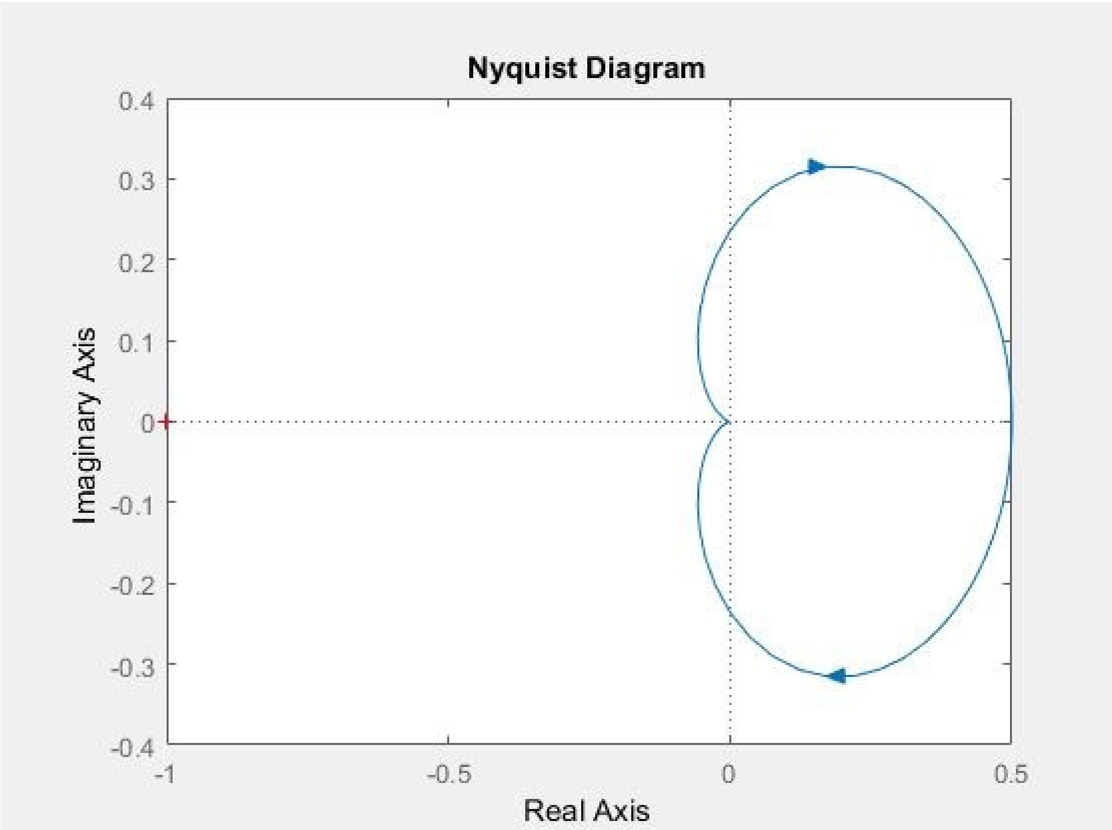
$$= \frac{2-\omega^2}{(2-\omega^2)^2+9\omega^2} \cdot j\frac{-3\omega}{(2-\omega^2)^2+9\omega^2} = \frac{2-\omega^2}{\omega^4+5\omega^2+4} \cdot j\frac{-3\omega}{\omega^4+5\omega^2+4}.$$

Frekvenční přenos vyjádřený pomocí komplexního čísla ve složkovém tvaru:

$$G(j\omega) = \frac{2-\omega^2}{\omega^4+5\omega^2+4} \cdot j\frac{-3\omega}{\omega^4+5\omega^2+4}.$$

Program MATLAB umí zobrazit přenos graficky pomocí funkce *nyquist*, jak je uvedeno níže.

```
B = [1];
A = [1, 3, 2];
sys = tf(B,A);
nyquist(sys)
```



Obr. 31 Zobrazení Nyquistovi křivky pomocí příkazu *nyquist*

8.7 Frekvenční charakteristika

Příklad: $G(j\omega) = \frac{2-\omega^2}{\omega^4+5\omega^2+4} \cdot j\frac{-3\omega}{\omega^4+5\omega^2+4}$

Amplitudově frekvenční charakteristika se získá následovně:

$$A(\omega) = \sqrt{(\frac{2-\omega^2}{\omega^4+5\omega^2+4})^2 + (\frac{-3\omega}{\omega^4+5\omega^2+4})^2} = \frac{\sqrt{(2-\omega^2)^2+(-3\omega)^2}}{\omega^4+5\omega^2+4} =$$

$$= \frac{\sqrt{4-4\omega^2+\omega^4+9\omega^2}}{\omega^4+5\omega^2+4} = \frac{\sqrt{\omega^4+5\omega^2+4}}{\omega^4+5\omega^2+4} = \frac{1}{\sqrt{\omega^4+5\omega^2+4}}.$$

Fázově frekvenční charakteristika:

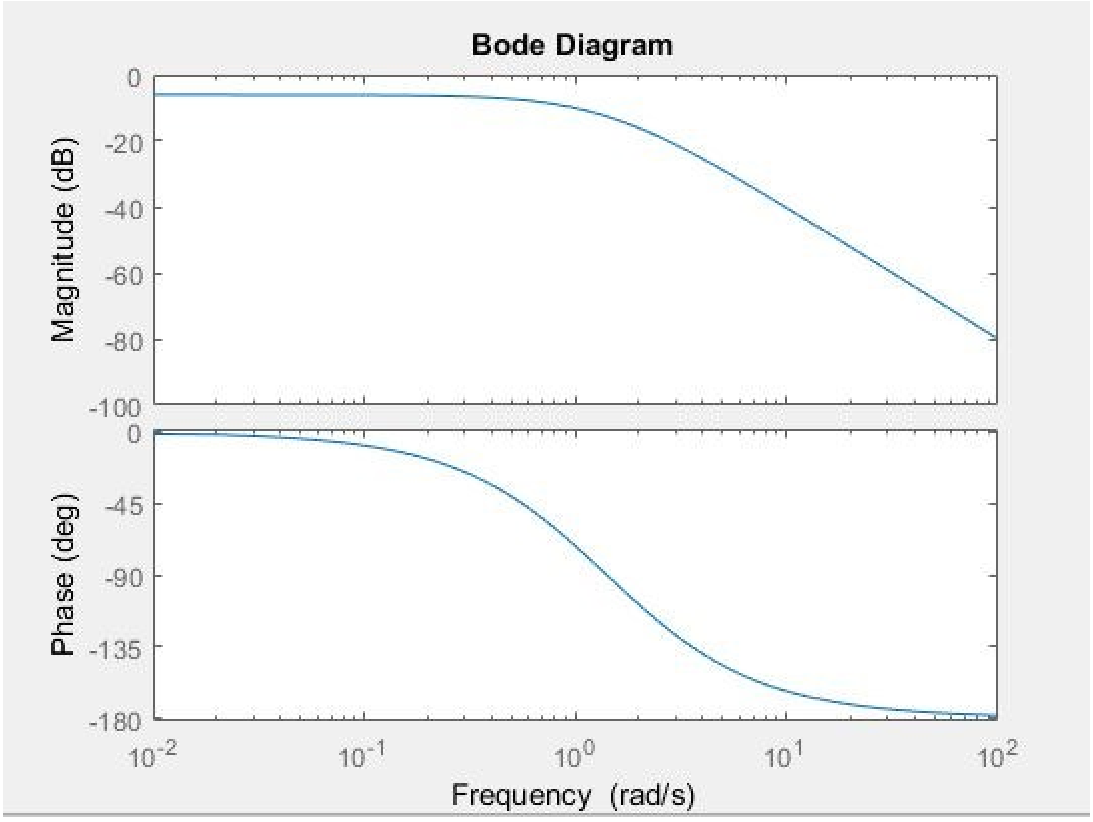
$$\varphi(j\omega) = \mathrm{arctg}\frac{\frac{-3\omega}{\omega^4+5\omega^2+4}}{\frac{2-\omega^2}{\omega^4+5\omega^2+4}} = -\mathrm{arctg}\frac{3\omega}{2-\omega^2}.$$

Frekvenční přenos v exponenciálním tvaru:

$$G(j\omega) = \frac{1}{\sqrt{\sqrt{\omega^4+5\omega^2+4}}} e^{-\mathrm{arctg}\frac{3\omega}{2-\omega^2}}.$$

Program MATLAB zobrazí přenos jako logaritmicko amplitudovou a fázovou charakteristiku graficky pomocí příkazu *bode*, jak je uvedeno níže.

```
B = [1];
A = [1, 3, 2];
sys = tf(B,A);
bode(sys)
```



Obr. 32 Zobrazení Bodeho křivky pomocí příkazu *bode*

ZÁVĚR

Cílem práce bylo vytvoření aplikace pro podporu výuky automatizace. Aplikace měla být vytvořena v programu MATLAB a v její nadstavbě Simulink. Aplikace byla vytvořeno zábavnou formou a dostala název „Kvíz s otázkami z automatizace“. Hra obsahuje šestnáct políček a pod každým políčkem se skrývá otázka. Pod políčky je ukrytý obrázek, který se postupně odkrývá. Hráč je omezen třemi životy.

Hra je pro uživatele, kteří mají základní znalosti automatizace. Protože databáze otázek a odpovědí jsou použity z této oblasti. Díky tomu, že jsou otázky uloženy ve formě databáze v samostatných souborech nezávislých na programu, lze jinými uživateli přidat další otázky nebo otázky obměnit. Součástí hry jsou i vzorové příklady, díky kterým si může uživatel zopakovat učivo.

Teoretická část práce obsahuje úvod k programu MATLAB a jeho rozšíření Simulink. Jsou zde také popsány základy automatizace a lineární systémy.

V praktické části je popsán samotný vývoj aplikace. Aplikace je popsána jak z pohledu programátora, tak i z pohledu uživatele.

Výsledná vytvořená aplikace může posloužit jako pomocné materiály v předmětech týkající se automatizace a programu MATLAB. Pomocí zábavné formy si studenti upevní svoje dosavadní znalosti nebo naopak svoje znalosti rozšíří.

SEZNAM POUŽITÉ LITERATURY

[1] BALATĚ, J. *Automaticke řizeni*. 2. přeprac. vyd. Praha: BEN – technicka literatura, 2004. 664 s. ISBN 80-7300-148-9.

[2] DUŠEK, František. *Simulink – Úvod do používání*. Pardubice: Univerzita Pardubice, 2001. ISBN 80-7194-273-1.

[3] KARBAN, Pavel. *Výpočty a simulace v programech Matlab a Simulink*. Brno: ComputerPress, 2006. ISBN 978-80-251-1448-3.

[4] NAVRÁTIL, Pavel, 2011. *Automatizace: vybrané statě*. Ve Zlíně: Univerzita Tomáše Bati ve Zlíně, 289 s. ISBN 978-80-7318-935-8. Dostupné také z: http://hdl.handle.net/10563/18581

[5] PERŮTKA, Karel. *MATLAB - Základy pro studenty automatizace a informačních technologií*. Zlín: UTB ve Zlíně, 2005. ISBN 80-7318-355-2.

[6] PROKOP, Roman, Radek MATUŠŮ a Zdenka PROKOPOVÁ. *Teorie automatického řízení - lineární spojité dynamické systémy*. Zlín: Univerzita Tomáše Bati ve Zlíně, 2006. ISBN 80-7318-369-2.

[7] ŠVARC, Ivan. *Automatické řízení*. Vyd. 2. Brno: Akademické nakladatelství CERM, 2011. ISBN 978-80-214-4398-3.

[8] VAŠEK, V. *Teorie automatickeho řizeni II*. [s.I.]: Rektorat Vysokeho učeni technického v Brně, 1990. 139 s. ISBN 80-214-0115-X.

[9] ZAPLATÍLEK, Karel a Bohuslav DOŇAR. *MATLAB pro začátečníky*. 2. vydání. Praha: BEN - technická literatura, 2005. ISBN 80-7300-175-6.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

A/D	analogově-digitální převodník
FAI	Fakulta aplikované informatiky
UTB	Univerzita Tomáše Bati
Matlab	Matrix laboratory
Simulink	Simulation link

SEZNAM OBRÁZKŮ

SEZNAM PŘÍLOH