

Multimediální videoprůvodce základy teorie regulace v Matlabu a Simulinku pro předmět "Matlab a Simulink"

Multimedia video-guide of basics of control theory in Matlab and
Simulink for "Matlab and Simulink" subject

Tomáš Zaoral



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2010/2011

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Tomáš ZAORAL**
Osobní číslo: **A08104**
Studijní program: **B 3902 Inženýrská informatika**
Studijní obor: **Informační a řídicí technologie**

Téma práce: **Multimediální videoprůvodce základy teorie
regulace v Matlabu a Simulinku pro předmět
"Matlab a simulink"**

Zásady pro vypracování:

1. Seznamte se s prací v programu Matlab a Simulink.
2. Seznamte se dále s již existujícími multimediálními videoprůvodci základů práce v Matlabu a Simulinku.
3. Vytvořte vlastního multimediálního videoprůvodce pokrývajícího obsahově zbývající část výuky předmětu "Matlab a simulink".
4. Vytvořeného multimediálního videoprůvodce umístěte na DVD s vlastním menu.

Rozsah bakalářské práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. K. Perůtka: **MATLAB – Základy pro studenty automatizace a inform. technologií**, FT UTB Zlín 2005, ISBN 80-7318-355-2.
2. F. Dušek: **Matlab a Simulink – úvod do používání**. Univerzita Pardubice 2000, 146 s., ISBN 80-7194-273-1.
3. P. Karban: **Výpočty a simulace v programech MATLAB a Simulink**, Computer Press 2006, ISBN 978-80-251-1448-3.
4. K. Zaplatílek, B. Doňar: **MATLAB pro začátečníky**, BEN-Technická literatura 2003, ISBN 80-7300-175-6.
5. K. Zaplatílek, B. Doňar: **MATLAB tvorba uživatelských aplikací**, BEN-Technická literatura 2004, ISBN 80-7300-133-0.
6. Kozák Š., Kajan S.: **Matlab – Simulink I**, STU Bratislava 1999, ISBN 80-227-1213-2.
7. Kozák Š.: **Matlab – Simulink II**, STU Bratislava 1999, ISBN 80-227-1235-3.

Vedoucí bakalářské práce:

Ing. Karel Perůtka, Ph.D.

Ústav řízení procesů

Datum zadání bakalářské práce:

25. února 2011

Termín odevzdání bakalářské práce:

7. června 2011

Ve Zlíně dne 25. února 2011

prof. Ing. Vladimír Vašek, CSc.
děkan



prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

ABSTRAKT

Bakalářská práce se zabývá vytvořením multimediálního videoprůvodce pro Matlab a Simulink. Práce navazuje na již existujícího videoprůvodce a doplňuje zbývající část výuky. Videoprůvodce poskytne studentům pomůcku při výuce předmětu Matlab a Simulink pro prezenční i kombinovanou formu studia v podobě krátkých instruktážních videí. Bakalářská práce by měla sloužit studentům jako nezbytný základ pro cvičení v programu Matlab a Simulink, zejména studentům kombinované formy studia při procvičování nebo plnění domácích úkolů.

Klíčová slova: videoprůvodce, Matlab a Simulink

ABSTRACT

This Bachelor thesis deals with creating multimedia video-guide for Matlab and Simulink. Thesis provides students a tool for teaching Matlab and Simulink subject for full-time and combined form study in the form of short instructional videos. This bachelor thesis would make the exercises in program Matlab and Simulink easier for all students, especially combined form students to do their practice or homework.

Keywords: video-guide, Matlab and Simulink

Rád bych poděkoval vedoucím bakalářské práce, Ing. Karlu Perůtkovi, PhD., za odborné vedení, cenné rady a věcné připomínky během řešení této práce a také své rodině a přátelům za podporu.

Motto

„Non schoale sed vitae discimus. - Neučíme se pro školu, ale pro život.“

Seneca

Prohlašuji, že

- beru na vědomí, že odevzdáním 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 bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk 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 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 – bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování 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 bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem 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 bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....
podpis diplomanta

OBSAH

ÚVOD.....	9
I TEORETICKÁ ČÁST.....	11
1 PROSTŘEDÍ MATLAB A SIMULINK.....	12
1.1 HISTORIE.....	12
1.1.1 Počátek	12
1.1.2 Historie vydání po současnost.....	13
1.2 OBLASTI VYUŽITÍ MATLABU	15
1.3 PRACOVNÍ PLOCHA PROSTŘEDÍ MATLAB	15
1.3.1 Vytvoření skriptu v MATLABu.....	17
1.4 ÚVOD K SIMULINKU	18
1.4.1 Jak Simulink funguje.....	19
1.4.2 Spuštění Simulinku	19
1.4.3 Popis nejčastěji používaných bloků Simulinku v oblasti teorie řízení.....	21
1.4.4 Subsystémy a maska subsystému.....	22
2 PRŮZKUM EXISTUJÍCÍCH VIDEOPRŮVODCŮ V OBLASTI TEORIE ŘÍZENÍ.....	23
2.1 ZDROJ WWW.MATHWORKS.COM.....	23
2.2 ZDROJ WWW.YOUTUBE.COM	25
3 METODY POUŽITÉ PŘI VYTVÁŘENÍ VIDEOPRŮVODCE	27
3.1 ZIEGLER-NICHOLSOVA METODA	27
3.1.1 Ziegler-Nicholsova metoda kritických parametrů (2. modifikace).....	27
3.2 ASTRÖMŮV REGULÁTOR.....	28
3.3 RUNGE-KUTTA METODA	29
3.3.1 Runge-Kutta-Fehlberg metoda.....	29
II PRAKTICKÁ ČÁST	31
4 TVORBA VIDEOPRŮVODCE	32
4.1 PROSTŘEDÍ CAMTASIA STUDIO 7.....	32
4.2 NAHRÁVÁNÍ OBRAZOVKY	33
4.3 NAHRÁVÁNÍ KOMENTÁŘŮ	34
4.4 VYTVOŘENÍ DVD MENU	35
4.4.1 Spuštění a práce s DVD	36
5 POPIS JEDNOTLIVÝCH VIDEOPRŮVODCŮ.....	38
5.1 VP 1 – DISKRÉTNÍ PID REGULÁTOR V MATLABU.....	38
5.1.1 Použité funkce nebo příkazy a jejich význam.....	40
5.2 VP 2 – DISKRÉTNÍ PID REGULÁTOR V SIMULINKU.....	40
5.2.1 Použité bloky knihovny Simulinku.....	41
5.3 VP 3 – SPOJITÝ REGULÁTOR V MATLABU	42
5.3.1 Použité funkce nebo příkazy a jejich význam.....	43
5.4 VP 4 – SPOJITÝ PID REGULÁTOR V SIMULINKU	44
5.4.1 Použité bloky knihovny Simulinku.....	45

5.5	VP 5 – EXPORT ZE SCOPE DO GRAFICKÉHO VÝSTUPU	45
5.5.1	Použité funkce nebo příkazy a jejich význam	45
5.6	VP 6 – ZJIŠŤOVÁNÍ ODEZVY SYSTÉMU ODE45	46
5.6.1	Použité funkce nebo příkazy a jejich význam	47
5.7	VP 7 – ZJIŠŤOVÁNÍ ODEZVY SYSTÉMU POMOCÍ VLASTNÍHO ŘEŠITELE DIFERENCIÁLNÍCH ROVNIC	47
5.7.1	Použité funkce nebo příkazy a jejich význam	48
ZÁVĚR		49
ZÁVĚR V ANGLIČTINĚ		50
SEZNAM POUŽITÉ LITERATURY		51
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK		52
SEZNAM OBRÁZKŮ		53
SEZNAM TABULEK		54
SEZNAM PŘÍLOH		55

ÚVOD

Cílem této bakalářské práce je seznámit se s prostředím MATLAB a Simulink a vytvořit multimediálního videoprůvodce, navazující na existujícího videoprůvodce a pokrývající obsahově zbylou část výuky předmětu „Matlab a Simulink“. Tento předmět je vyučován na Fakultě aplikované informatiky Univerzity Tomáše Bati ve Zlíně, obor Inženýrská informatika. Studenti se v předmětu seznamují nejprve se základy práce v MATLABu a později také v Simulinku. Velká část toho předmětu se věnuje oblasti teorie řízení, a proto se tato bakalářská práce věnuje tvorbě multimediálního videoprůvodce, který je zaměřen právě na oblast teorie řízení, a který bude sloužit jako pomůcka při výuce.

Teoretická část práce obsahuje 3 hlavní kapitoly:

- Prostředí MATLAB a Simulink,
- Průzkum existujících videoprůvodce v oblasti teorie řízení, a
- Metody použité při vytváření videoprůvodce.

V kapitole „Prostředí Matlab a Simulink“ jsem se zabýval základním popisem prostředí Matlab a Simulink. Nastínil jsem historii vzniku MATLABu a vedle historie také rok vydání jednotlivých verzí a k nim příslušné verze Simulinku. Dále jsem popsal pracovní plochu prostředí Matlab a vysvětlil jsem jednotlivá pracovní okna, která jsou používána při práci s tímto programem. V této kapitole je dále vysvětleno, co je to Simulink, jak se s ním pracuje a jak se spustí. V poslední části této kapitoly jsou vypsány nejčastěji používané bloky v oblasti teorie řízení pro sestavení modelu v Simulinku. V další hlavní kapitole „Průzkum existujících videoprůvodce v oblasti teorie řízení“ jsem provedl průzkum internetu, ve kterém jsem se zaměřil, jak už z názvu vyplývá, právě na videoprůvodce v oblasti teorie řízení. Zvolil jsem dva videoprůvodce, které jsem detailně popsal. Jeden z nich je řešen v MATLABu a druhý v Simulinku. V poslední kapitole teoretické části s názvem „Metody použité při vytváření videoprůvodce“ se zabývám dvěma metodami použitými u diskrétního PID regulátoru, jedná se o Ziegler-Nicholsovu metodu a Aströmův regulátor. Dále se zabývám numerickou metodou pro řešení obyčejných diferenciálních rovnic, která je použita v sedmém videu videoprůvodce a slouží k vytvoření vlastního řešitele diferenciálních rovnic.

Praktická část obsahuje popis použitého softwaru pro tvorbu videoprůvodce, použité nástroje pro úpravu videa a aplikaci pro vytvoření menu DVD. Pro zachycení obrazovky a následnou úpravu videa a vytvoření menu DVD byl použit program Camtasia Studio 7 v licenci trial od společnosti TechSmith. Jedná se o profesionální program určený pro tvorbu videoprůvodce, kde součástí je i aplikace pro nahrávání obrazovky a komentářů.

I. TEORETICKÁ ČÁST

1 PROSTŘEDÍ MATLAB A SIMULINK

MATLAB je integrované prostředí pro vědeckotechnické výpočty, modelování, návrhy algoritmů, simulace, analýzu a prezentaci dat, paralelní výpočty, měření a zpracování signálu, návrhy řídicích a komunikačních systémů.

Výpočetní systém MATLAB se během uplynulých let stal celosvětovým standardem v oblasti technických výpočtů a simulací ve sféře vědy, výzkumu, průmyslu i v oblasti vzdělávání.

MATLAB poskytuje svým uživatelům nejen mocné grafické a výpočetní nástroje, ale i rozsáhlé specializované knihovny funkcí spolu s výkonným programovacím jazykem čtvrté generace. Knihovny jsou svým rozsahem využitelné prakticky ve všech oblastech lidské činnosti.

Díky své architektuře je MATLAB určen zejména těm, kteří potřebují řešit početné náročné úlohy a přitom nechťejí nebo nemají čas zkoumat matematickou podstatu problémů. Více než milion uživatelů po celém světě využívá možnosti jazyka MATLABu, který je mnohem jednodušší než například Fortran nebo C a který skýtá obrovský potenciál produktivity a tvořivosti. Za nejsilnější stránku MATLABu je považováno mimořádně rychlé výpočetní jádro s optimálními algoritmy, které jsou prověřeny léty provozu na špičkových pracovištích po celém světě. MATLAB byl implementován na všech významných platformách (Windows, Linux, Solaris, Mac).[1]

Název MATLAB vznikl zkrácením slov MATrix LABoratory (volně přeloženo „laboratoř s maticemi“), což odpovídá skutečnosti, že klíčovou datovou strukturou při výpočtech v MATLABu jsou matice. Vlastní programovací jazyk vychází z jazyka Fortran. [2]

1.1 Historie

1.1.1 Počátek

MATLAB vytvořil v roce 1970 Cleve Moler, vedoucí ústavu počítačové techniky na univerzitě v Novém Mexiku. MATLAB se brzy rozšířil na ostatní univerzity a získal silnou oporu v oblasti aplikované matematiky. Během Molerovi návštěvy v roce 1983 na Standfordské univerzitě rozpoznal Jack Little velký komerční potenciál, a následně se

spojil s Molerem a Stevem Bangertem. Přepsali MATLAB do jazyka C a v roce 1984 objevili společnost MathWorks, která pokračovala v jejich vývoji. Tyto přepsané knihovny byly známy jako JACKPAC. V roce 2000 byl MATLAB přepsán pro použití novějších knihoven pro maticové operace, tyto knihovny byly pojmenovány LAPACK.

MATLAB byl nejprve přijat inženýry pro kontrolu řízení, ale rychle se šířil do mnoha jiných sfér. Nyní se využívá v oblasti vzdělávání, zejména při výuce lineární algebry a numerické analýzy, a také je populární mezi vědci zabývajícími se zpracováním obrazu.

1.1.2 Historie vydání po současnost

Následující tabulka (Tabulka 1) zachycuje historický vývoj, tj. verze MATLABu, název daného vydání, verze Simulinku (pokud již daná verze MATLABu Simulink obsahovala) a v poslední řadě také rok vydání.

Verze MATLABu	Název vydání	Verze Simulinku	Rok vydání
MATLAB 1.0	-	-	1984
MATLAB 2	-	-	1986
MATLAB 3	-	-	1987
MATLAB 3.5	-	Simulink 1	1990
MATLAB 4	-		1992
MATLAB 4.2C	R7		1994
MATLAB 5.0	R8	Simulink 2	1996
MATLAB 5.1	R9		1997
MATLAB 5.1.1	R9.1		1997
MATLAB 5.2	R10		1998
MATLAB 5.2.1	R10.1		1998
MATLAB 5.3	R11	Simulink 3	1999
MATLAB 5.3.1	R11.1		1999

MATLAB 6.0	R12	Simulink 4	2000
MATLAB 6.1	R12.1	Simulink 4.1	2001
MATLAB 6.5	R13	Simulink 5	2002
MATLAB 6.5.1	R13SP1	Simulink 5.1	2003
MATLAB 6.5.2	R13SP2	Simulink 5.2	2003
MATLAB 7	R14	Simulink 6.0	2004
MATLAB 7.0.1	R14SP1	Simulink 6.1	2004
MATLAB 7.0.4	R14SP2	Simulink 6.2	2005
MATLAB 7.1	R14SP3	Simulink 6.3	2005
MATLAB 7.2	R2006a	Simulink 6.4	2006
MATLAB 7.3	R2006b	Simulink 6.5	2006
MATLAB 7.4	R2007a	Simulink 6.6	2007
MATLAB 7.5	R2007b	Simulink 7.0	2007
MATLAB 7.6	R2008a	Simulink 7.1	2008
MATLAB 7.7	R2008b	Simulink 7.2	2008
MATLAB 7.8	R2009a	Simulink 7.3	2009
MATLAB 7.9	R2009b	Simulink 7.4	2009
MATLAB 7.10	R2010a	Simulink 7.5	2010
MATLAB 7.11	R2010b	Simulink 7.6	2010

Tabulka 1 Historie vydání MATLABu a Simulinku

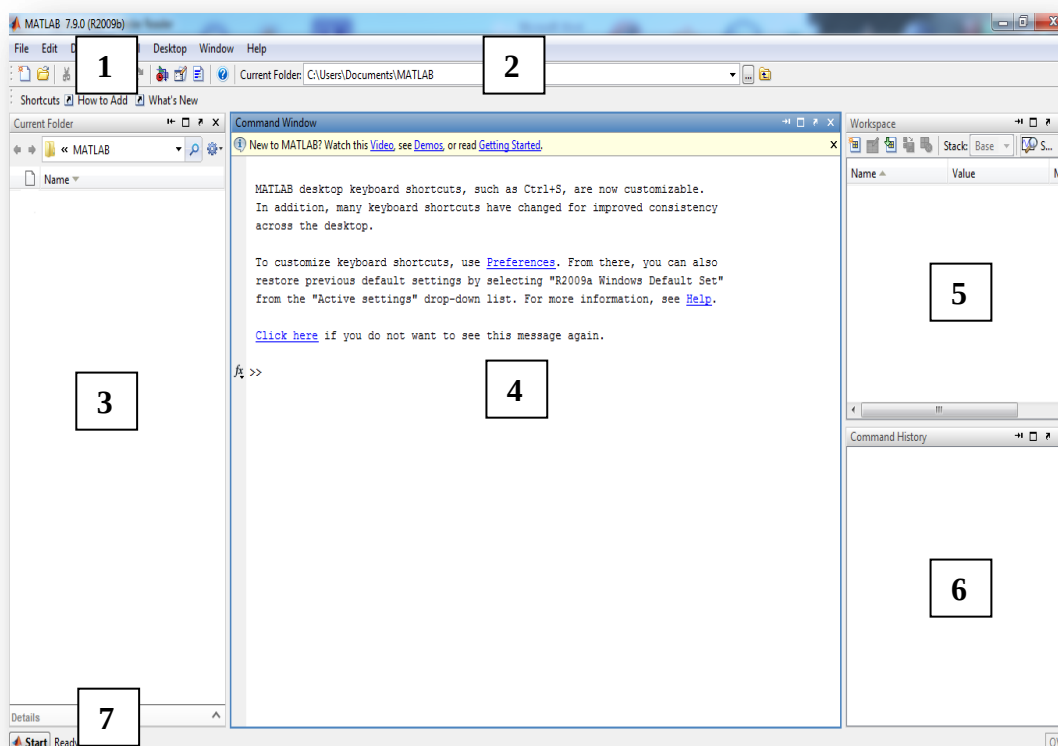
1.2 Oblasti využití MATLABu

Typické oblasti využití:

- aplikovaná matematika,
- automatické řízení a regulace,
- zpracování signálu a komunikace,
- zpracování obrazu,
- měření a testování,
- výpočetní biologie,
- finanční modelování a analýza,
- modelování fyzikálních soustav.

1.3 Pracovní plocha prostředí MATLAB

Na následujícím obrázku (Obrázek 1) je zachycena pracovní plocha MATLABu.



Obrázek 1 Pracovní plocha prostředí MATLAB R2009b

Nyní popíši jednotlivé části pracovní plochy:

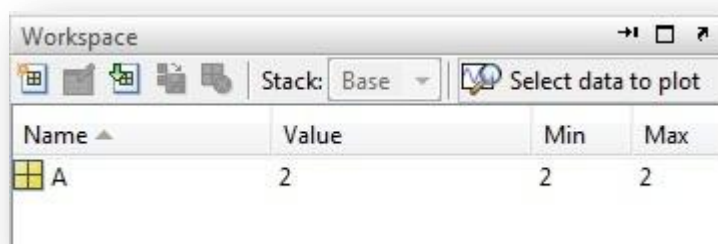
Menu (Obrázek 1, položka 1) – nabídka menu se mění v závislosti na aktuálně používaném nástroji MATLABu.

Aktuální složka (Obrázek 1, položka 2) – slouží pro nastavení cesty k aktuální složce.

Prohlížeč aktuálních složek (Obrázek 1, položka 3) – hlavní nástroj pro práci s MATLAB soubory. Umožňuje přístup operačnímu systému pro správu souborů v rámci MATLABu.

Příkazové okno (Obrázek 1, položka 4) – jeden z hlavních nástrojů pro psaní příkazů, spouštění MATLAB funkcí a ostatních M-File souborů, a zobrazování výsledků.

Workspace (Obrázek 1, položka 5) – v českém překladu „pracovní prostor“, skládá se ze souboru proměnných, které byly použity při práci s MATLABem. Vytvoříme-li proměnnou A s hodnotou 2, uloží se do Workspace (viz. Obrázek 2).



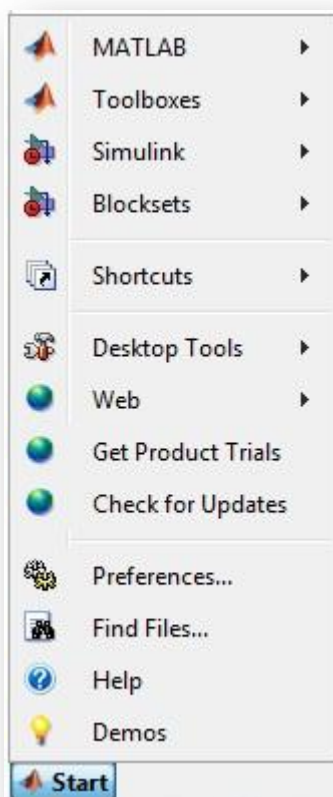
Obrázek 2 Ukázka panelu Workspace při použití příkazu „A = 1 + 1“

Historie příkazů (Obrázek 1, položka 6) – zobrazuje výpis nejčastěji spouštěných příkazů v příkazovém okně. Vypíše se datum a čas použitého příkazu, a samotný příkaz (viz. Obrázek 3).



Obrázek 3 Ukázka panelu historie příkazů

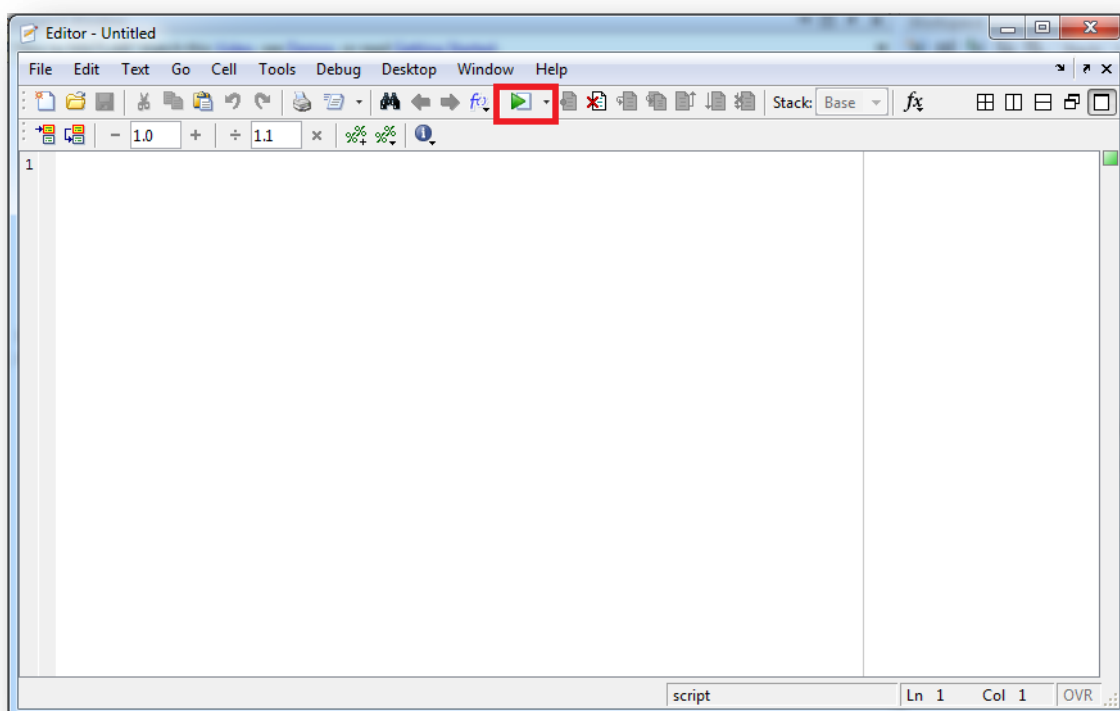
Tlačítko START (Obrázek 1, položka 7) – Slouží obdobně jako tlačítko START v OS Windows, tedy pro rychlý přístup k nástrojům a dalším nabídkám (viz. Obrázek 4).



Obrázek 4 Detail na tlačítko START

1.3.1 Vytvoření skriptu v MATLABu

Po spuštění MATLABu lze zadávat jednotlivé příkazy do příkazového okna nebo lze vytvořit M-soubor. M-soubory slouží k ukládání posloupnosti příkazů (skript) nebo k ukládání uživatelských funkcí. Nový M-soubor otevřeme přes nabídku *File -> New -> Blank M-File*. Po zapsání posloupnosti příkazů do M-souboru a následném uložení, lze spustit tento skript pomocí zelené ikonky spuštění (Run) v panelu nástrojů, který se nachází v horní části M-souboru (Obrázek 5).



Obrázek 5 Okno M-souboru se zvýrazněným tlačítkem spuštění

1.4 Úvod k Simulinku

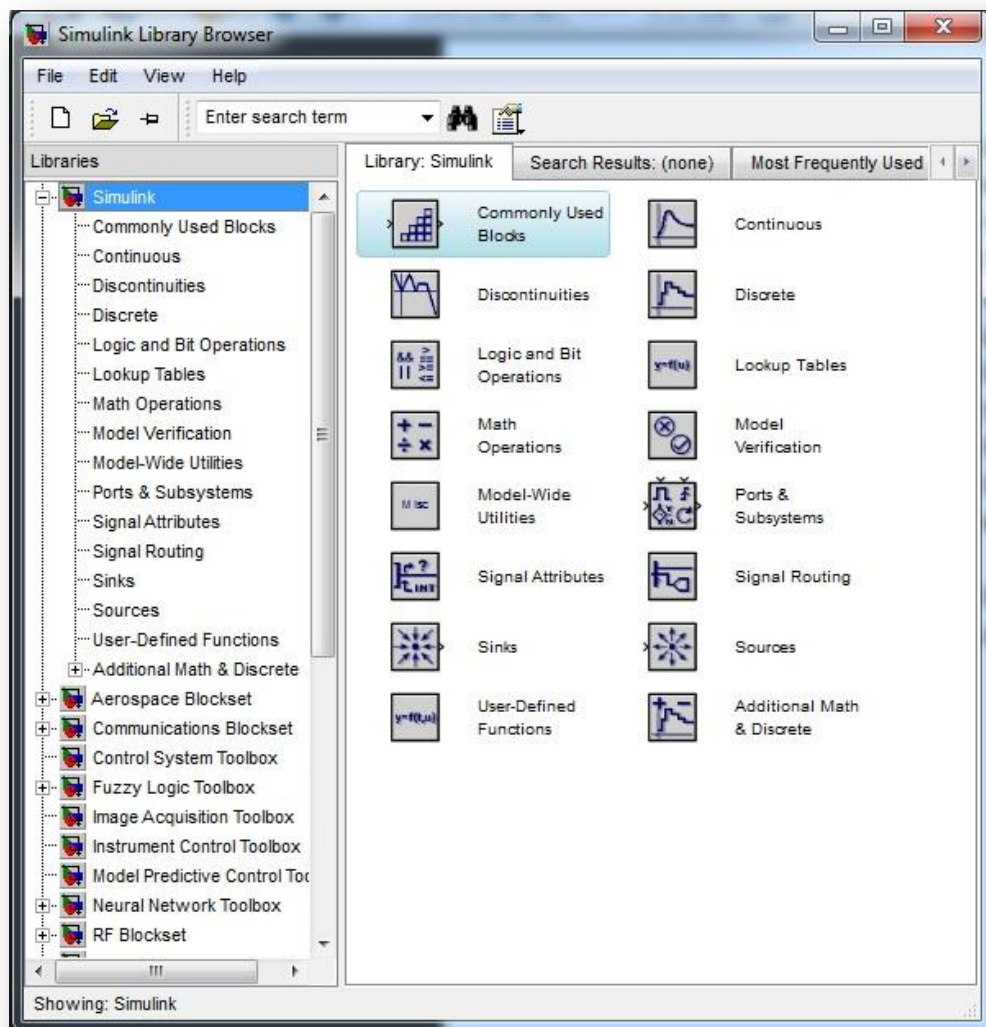
Simulink je softwarový balíček, který umožňuje modelovat, simulovat a analyzovat systémy, jejichž výstupy se v průběhu času mění. Takové systémy jsou často označovány jako dynamické systémy. Pro modelování Simulink poskytuje grafické uživatelské rozhraní (GUI), ve kterém si uživatel sestaví vlastní model z jednotlivých bloků knihovny pomocí tzv. click-and-drag operace myši, což je kliknutí na blok a tažení bloku na místo, kde se blok má v modelu nacházet. Simulink zahrnuje rozsáhlou knihovnu bloků, jako je knihovna zdrojů, knihovna lineárních a nelineárních prvků a mnoho dalších. Dále si uživatel může vytvořit i vlastní bloky. Simulink může být využit ke zkoumání chování celé řady dynamických systémů reálného světa, včetně elektrických, mechanických a termodynamických systémů. První vydání Simulinku obsahovala verze MATLABu 3.5 v roce 1990.

1.4.1 Jak Simulink funguje

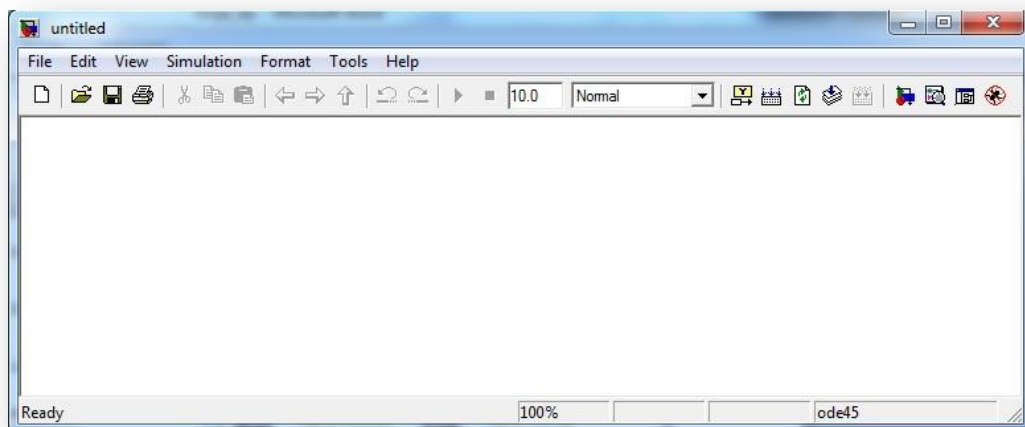
Simulovat dynamický systém se skládá z dvou kroků. Za prvé, uživatel vytvoří blokové schéma s využitím Simulink model editoru, který graficky zobrazuje časové závislé matematické vztahy mezi vstupy, výstupy a stavy. Blokové schéma se sestaví z bloků, které zahrnuje knihovna Simulinku. Po vytvoření blokového schématu nebo modelu uživatel přikáže Simulinku, aby provedl simulaci systému, který je zastoupený modelem za určitých počátečních a koncových hodnot času. Pro zobrazení výsledku simulace se nejčastěji používá blok Scope z knihovny Sinks.

1.4.2 Spuštění Simulinku

Způsobů, jak spustit Simulink, je několik. Popíšeme si dva z nich. Po spuštění MATLABu zapíšeme do příkazového okna (Obrázek 1, položka 4) příkaz „simulink“ nebo se do simulinku dostaneme přes tlačítko *START -> Simulink -> Library Browser*. Otevře se okno nadepsané Simulink Library Browser, které by mělo vypadat podobně jako na obrázku 6. Na levé straně vidíme stromovou strukturu jednotlivých knihoven, které Simulink obsahuje a na pravé straně se po kliknutí na libovolnou knihovnu zobrazí seznam bloků v dané knihovně. Tyto bloky představují základní stavební jednotku Simulinku a abychom mohli sestavit nějaký model, je zapotřebí grafického prostředí. Okno grafického prostředí otevřeme přes nabídku *File -> New -> Model*. Objeví se prosté okno (Obrázek 7), kde můžeme vkládat jednotlivé bloky z knihoven a sestavit tak libovolný model pro simulaci.

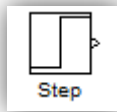


Obrázek 6 Okno Simulink Library Browser (Prohlížeč knihoven)



Obrázek 7 Okno grafického prostředí Simulink

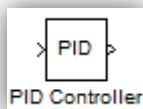
1.4.3 Popis nejčastěji používaných bloků Simulinku v oblasti teorie řízení



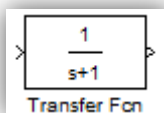
Blok *Step* generuje na výstup krok mezi dvěma definovanými úrovněmi v zadaný čas.



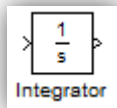
Blok *Repeating sequence* generuje na výstup periodický skalární signál s definovaným průběhem.



Blok *PID controller* je implementovaný PID regulátor, kde se nastavují hodnoty proporcionální, integrační a derivační složky.



Blok *Transfer Function* modeluje lineární systémy pomocí Laplaceho transformace.



Blok *Integrator* integruje vstup v jeho současné době kroku.



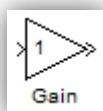
Blok *Derivate* aproximuje derivát jeho vstupu výpočtem, kde du je změna vstupních hodnot a dt změna v čase od předchozího kroku simulace.



Blok *Zero-Order Hold* vzorkuje a udržuje vstup po určenou periodu vzorkování.



Blok *Sum* provádí sčítání nebo odčítání jeho vstupů.



Blok *Gain* násobí vstup o definovanou konstantní hodnotu.



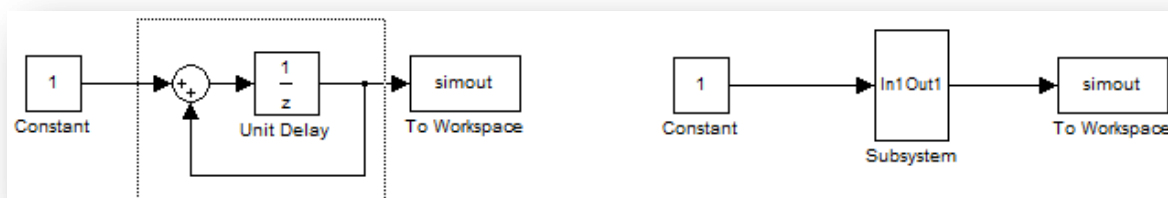
Blok *Mux* kombinuje své vstupy do jednoho vektoru výstupu.



Blok *Scope* zobrazuje svůj vstup s ohledem na simulační dobu.

1.4.4 Subsystémy a maska subsystému

Subsystém reprezentuje sadu bloků, které mohou být nahrazeny jedním jediným blokem. Hlavní výhodou subsystémů je, že jakmile se váš model zvětšuje a stává složitějším, můžete jej jednoduše seskupit do subsystému. Vytvořit subsystém lze tak, že označíme bloky, které chceme sjednotit a po kliknutí pravým tlačítkem myši na tyto bloky zvolíme možnost *Create Subsystem*. Označené bloky se okamžitě sjednotí do subsystému. Po otevření subsystému lze provádět úpravy. Kliknutím pravého tlačítka na subsystém a zvolení možnosti *Mask Subsystem* můžeme vytvořit masku subsystému. Pomocí masky lze měnit vzhled bloku, parametry bloků uvnitř masky a také lze vytvořit dokumentaci nebo nápovědu pro maskovaný blok.



Obrázek 8 Vlevo označení bloků pro vytvoření subsystému, vpravo vytvořený subsystém

2 PRŮZKUM EXISTUJÍCÍCH VIDEOPRŮVODCŮ V OBLASTI TEORIE ŘÍZENÍ

V oblasti teorie řízení jsou nejpoužívanější regulátory, zejména regulátory typu PID. Proto jsem při průzkumu na internetu zvolil videoprůvodce od společnosti MathWorks, která je developer MATLABu, a kde autor řeší ladění PID regulátoru v Simulinku. Ze serveru youtube.com, který patří k největší komunitě pro sdílení videí na světě, jsem zvolil videoprůvodce regulátorů typu PI, PD a PID, řešeného v MATLABu.

2.1 Zdroj www.mathworks.com

Na oficiálních stránkách MathWorks najdeme řadu návodů a rad. Na odkazu níže se nachází videoprůvodce Simulinku s názvem *PID Controller Design for a DC Motor*, který vytvořil Arkadiy Turevskiy, výrobní a reklamní manažer MathWorks:

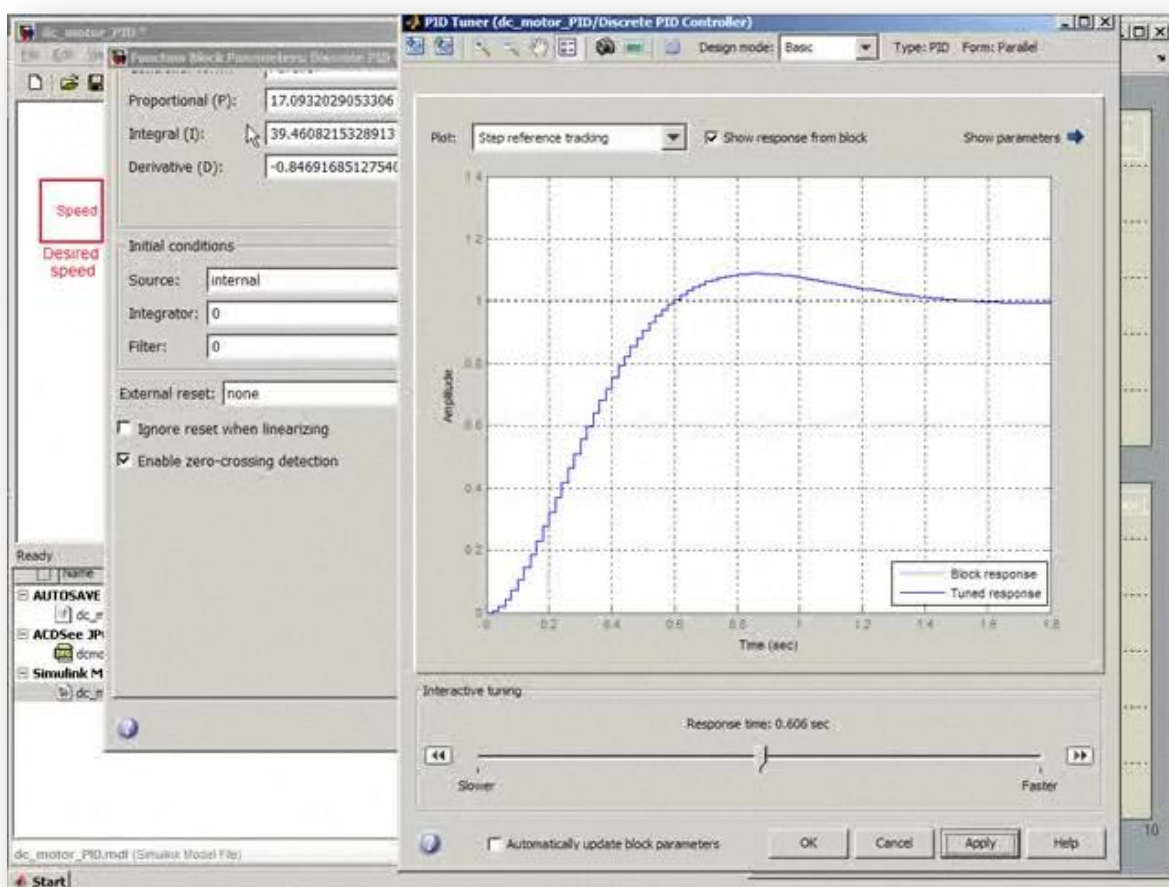
<http://www.mathworks.com/products/simcontrol/description2.html>

Tento multimediální videoprůvodce se skládá z šesti dílčích částí: *Introduction*, *DC Motor Model*, *Control Problem Setup*, *Adding PID Controller*, *Tunning PID Controller* a *Fine-Tunning PID Controller*.

V části *Introduction* autor představuje řešený problém. Následuje část *DC Motor Model*, kde je zobrazena sestava modelu z jednotlivých bloků knihoven Simulinku. Zde stojí za povšimnutí, že autor použil masku subsystému k sjednocení více bloků do jednoho jediného s názvem DC Motor. V části *Control Problem Setup* poukazuje na problémy modelu, které řeší v další části *Adding PID Controller*. V této části autor přidá blok Discrete PID Controller, který lze najít v knihovně Simulinku, v záložce Discrete. Po otevření tohoto bloku autor pokazuje na možnost změny typu použitého regulátoru. Na výběr jsou možnosti PID, PI, PD, P a I. Zde je důležité dodržet dobu vzorkování, která je nastavena v bloku A-D Converter pro správnost simulace. Autor ponechá výchozí hodnoty jednotlivých složek parametrů regulátor a po zavření bloku spustí simulaci, abychom viděli, jaký bude rozdíl mezi požadovanou a naměřenou rychlostí. Rozdíl, který je zde vidět je způsoben nesprávným nastavením regulátoru, a proto se autor v další části

Tuning PID Controller zabývá nastavením parametrů regulátoru, aby bylo docíleno požadované rychlosti. Toto nastavení je jednoduché a spočívá pouze v otevření bloku regulátoru a kliknutí na tlačítko *Tune*. Po spuštění ladění regulátoru se provede linearizování modelu, navrhnutí regulátoru a inicializace ladění. Po dokončení těchto tří bodů se otevře grafické uživatelské rozhraní pomocí kterého lze interaktivně ladit parametry regulátoru. V poslední části s názvem *Fine-Tuning PID Controller* autor poukáže na to, co se stane, pokud zrychlíme dobu odezvy regulátoru. U modelu stejnosměrného motoru bude odezva menší, ale za cenu většího kolísání napětí.

Tento videoprůvodce je přínosem pro uživatele, kteří nemají mnoho zkušeností se Simulinkem, a pro ty, kteří neumí sestavit model diskrétního PID regulátoru. Jednoduchým vložením jednoho předdefinovaného bloku z knihovny Simulinku a následným spuštěním automatického ladění parametrů regulátoru Simulink vše vyřeší za uživatele.



Obrázek 9 Ukázka interaktivního grafického rozhraní pro ladění regulátoru

2.2 Zdroj www.youtube.com

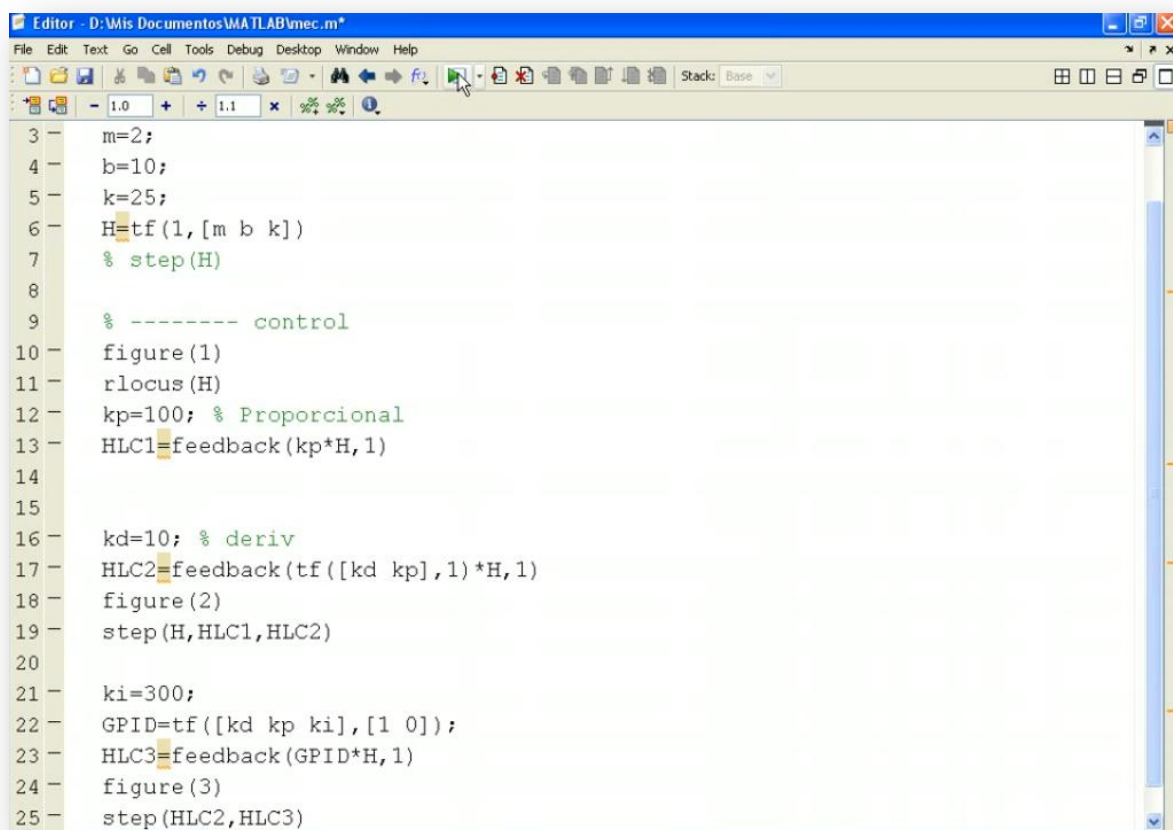
Na serveru Youtube můžeme nalézt videoprůvodce od uživatele lunaticokraken. Tento španělský videoprůvodce nese název *Uso de MATLAB para Control PD, PI y PID* a zabývá se laděním PD, PI a PID regulátoru v MATLABu.

Videoprůvodce se nachází na odkazu:

<http://www.youtube.com/watch?v=TBLebM1CyuM>

Tento multimediální videoprůvodce je rozdělen na čtyři části. Na počátku první části je definován přenos systému pomocí příkazu *tf*, který slouží k vytvoření přenosové funkce. Autor dále pomocí funkce *rlocus* (zobrazení polohy pólů a nul otevřené smyčky a trajektorii jejich pohybu v závislosti na zesílení) zjišťuje přibližné zesílení, které nastaví proporciální složce. Následně zapojí systém do zpětné vazby pomocí funkce *feedback* a vykreslí přechodové funkce systému bez zpětné vazby a se zpětnou vazbou do jednoho grafu. Pokračuje v druhé části videoprůvodce, kde přidá derivační složku a vznikne tak regulátor typu PD. Přibližně v polovině druhé části tohoto videoprůvodce autor ukazuje, jak se mění přenosová funkce přidáním proporciální a derivační složky, a pomocí příkazu *zero* a *pole* si nechá vypočítat nuly a póly systému. Ty se nachází v levé části poloroviny, a tudíž se jedná o stabilní systém. V třetí části videoprůvodce autor přidává integrační složku a vzniká tak regulátor typu PID a v poslední části tohoto videoprůvodce autor mění proporciální složku a poukazuje na to, jak se tato změna projeví v regulačním pochodu regulátoru.

Tento videoprůvodce je veden ve španělském jazyce, což může být značný problém španělsky nehovořící uživatele, ale na druhou stranu už při pohledu na zdrojový kód je jasné, o co se tu jedná.



The image shows a screenshot of a MATLAB editor window. The title bar reads "Editor - D:\Mis Documentos\MATLAB\vnec.m*". The menu bar includes File, Edit, Text, Go, Cell, Tools, Debug, Desktop, Window, and Help. The toolbar contains various icons for file operations, editing, and execution. The script content is as follows:

```
3- m=2;
4- b=10;
5- k=25;
6- H=tf(1,[m b k])
7- % step(H)
8-
9- % ----- control
10- figure(1)
11- rlocus(H)
12- kp=100; % Proporcional
13- HLC1=feedback(kp*H,1)
14-
15-
16- kd=10; % deriv
17- HLC2=feedback(tf([kd kp],1)*H,1)
18- figure(2)
19- step(H,HLC1,HLC2)
20-
21- ki=300;
22- GPID=tf([kd kp ki],[1 0]);
23- HLC3=feedback(GPID*H,1)
24- figure(3)
25- step(HLC2,HLC3)
```

Obrázek 10 Ukázka zdrojového kódu videoprůvodce ze serveru Youtube

3 METODY POUŽITÉ PŘI VYTVÁŘENÍ VIDEOPRŮVODCE

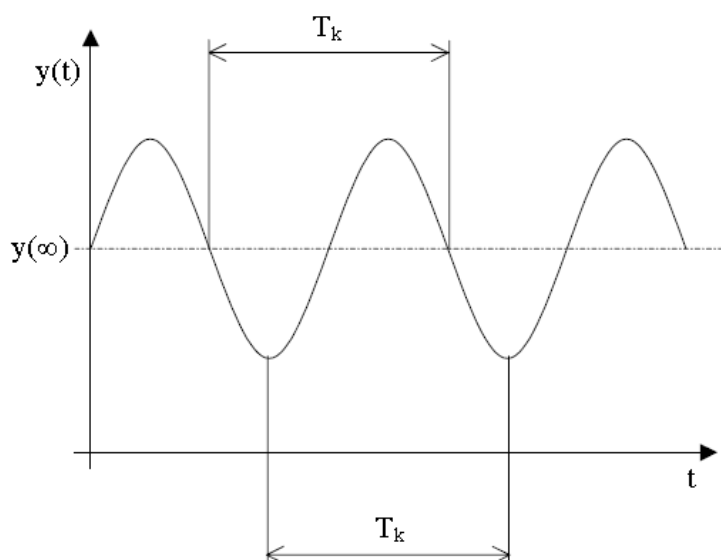
3.1 Ziegler-Nicholsova metoda

Jedná se o heuristickou metodu ladění PID regulátoru, o kterou se zasloužil John G. Ziegler a Nathaniel B. Nichols. Podle příjmení těchto autorů byla tato metoda pojmenována. Metoda má tři modifikace, z nichž 2. modifikace je použita ve videoprůvodci.

3.1.1 Ziegler-Nicholsova metoda kritických parametrů (2. modifikace)

Tato metoda je použitelná pro lineární spojitě i diskrétní regulační obvody. Princip spočívá v tom, že přivedeme obvod do tzv. kritického stavu, tj. na kmitavou mez stability, přičemž regulátor pracuje pouze s proporciální složkou. Integrační a derivační složka je vyřazena.[3]

Dosáhne-li obvod meze stability, hodnota zesílení, při které tento stav nastal, se nazývá kritické zesílení K_{pk} . Perioda kmitu, při které k tomu došlo pak nazveme periodu kritických kmitů T_k .



Obrázek 11 Určení kritické periody

Následující tabulka nám umožní určit hodnoty parametrů regulátoru:

Typ regulátoru	T_P	T_I	T_D
P	$0,5 K_{pk}$		
PI	$0,45 K_{pk}$	$0,83 T_k$	
PD	$0,4 K_{pk}$		$0,5 T_k$
PID	$0,6 K_{pk}$	$0,5 T_k$	$0,125 T_k$

Tabulka 2 Ziegler-Nicholsovy (2. modifikace) vztahy pro výpočet parametrů regulátoru

Více informací najdete v odborné literatuře Základy automatizace, autorů Jiří Tůma, Renata Wagnerová, Radim Farana a Lenka Landryová.

3.2 Aströmův regulátor

Aströmův algoritmus pro výpočet akčního zásahu diskrétního PID regulátoru je dán vztahem

$$u(k) = u_{PI}(k) + u_D(k), \quad (9)$$

kde

$$u_{PI}(k) = K_P \cdot [y(k-1) - y(k)] + \frac{K_P T_0}{2T_I} \cdot [e(k) + e(k-1)] + \beta K_P \cdot [w(k) - w(k-1)] + u_{PI}(k-1), \quad (10)$$

a

$$u_D(k) = K_P \cdot \frac{T_D \alpha}{T_D + T_0 \alpha} \cdot [y(k-1) - y(k)] + \frac{T_D}{T_D + T_0 \alpha} \cdot u_D(k-1). \quad (11)$$

Parametry ve vztazích (10) a (11) se volí v rozmezí $\beta \in (0;1)$, $\alpha \in <3;20>$. Více informací najdete v odborné literatuře Advanced PID Control, autorů K. J. Åström, T. Hägglung.

3.3 Runge-Kutta metoda

Nejpoužívanější metoda pro numerické řešení obyčejných diferenciálních rovnic. Vyznačuje se čtvrtým řádem přesnosti. V každém kroku se počítají čtyři pomocné veličiny. Tato technika byla vyvinuta přibližně kolem roku 1900 německými matematiky C. Runge a M. W. Kutta. Vylepšení metody Runge-Kutta navrhl roku 1970 německý matematik Erwin Fehlberg a tomuto vylepšení se dal název Fehlbergova metoda nebo také Runge-Kutta-Fehlberg metoda.

3.3.1 Runge-Kutta-Fehlberg metoda

Vylepšení spočívá v tom smyslu, že v každém kroku lze řídit velikost kroku a tedy i výsledné chyby použitím dvou metod Runge-Kutta. Fehlberg našel takové dvě varianty metody Runge-Kutta, které při současném použití vyžadují pouze 6 operací. Tato metoda je pátého řádu přesnosti. [4]

Fehlberg používá jednak metodu Runge-Kutta pátého řádu, při níž se hodnota funkce y_{n+1} vyjádří jako

$$y_{n+1} = y_n + \alpha_1 k_1 + \alpha_2 k_2 + \alpha_3 k_3 + \alpha_4 k_4 + \alpha_5 k_5 + \alpha_6 k_6, \quad (1)$$

což můžeme vyjádřit ve tvaru

$$y_{n+1} = y_n + \alpha^T k, \quad (2)$$

přičemž složky α^T jsou

$$\alpha^T = \left[\frac{16}{135}, 0, \frac{6656}{12825}, \frac{28561}{56430}, -\frac{9}{50}, \frac{2}{55} \right], \quad (3)$$

a dále metodu čtvrtého řádu, kdy vyjadřuje y_{n+1}^* ve tvaru

$$y_{n+1}^* = y_n + \beta_1 k_1 + \beta_2 k_2 + \beta_3 k_3 + \beta_4 k_4 + \beta_5 k_5, \quad (4)$$

tedy

$$y_{n+1}^* = y_n + \beta^T k, \quad (5)$$

přičemž

$$\beta^T = \left[\frac{25}{216}, 0, \frac{1408}{2565}, \frac{2197}{4104}, -\frac{1}{5} \right]. \quad (6)$$

Složky vektoru k jsou dány následujícími vztahy:

$$\begin{aligned} k_1 &= h \cdot f(x_n, y_n), \\ k_2 &= h \cdot f\left(x_n + \frac{h}{4}, y_n + \frac{k_1}{4}\right), \\ k_3 &= h \cdot f\left(x_n + \frac{3h}{8}, y_n + \frac{3k_1}{32} + \frac{9k_2}{32}\right), \\ k_4 &= h \cdot f\left(x_n + \frac{12h}{13}, y_n + \frac{1932k_1}{2197} - \frac{7200k_2}{2197} + \frac{7296k_3}{2197}\right), \\ k_5 &= h \cdot f\left(x_n + h, y_n + \frac{439k_1}{216} - 8k_2 + \frac{3680k_3}{512} - \frac{845k_4}{4104}\right), \\ k_6 &= h \cdot f\left(x_n + \frac{h}{2}, y_n - \frac{8k_1}{27} + 2k_2 - \frac{3544k_3}{2565} + \frac{1859k_4}{4104} - \frac{11k_5}{40}\right). \end{aligned} \quad (7)$$

Rozdíl mezi (1) a (4) dá odhad chyby, tedy

$$\varepsilon_{n+1} \approx y_{n+1} - y_{n+1}^* = \frac{k_1}{360} - \frac{128k_3}{4275} - \frac{2197k_4}{75240} + \frac{k_5}{50} + \frac{2k_6}{55}. \quad (8)$$

Více informací najdete v odborné literatuře Numerické metody, autorů M. Navara a A. Němeček.

II. PRAKTICKÁ ČÁST

4 TVORBA VIDEOPRŮVODCE

K vytvoření videoprůvodce byl použit program Camtasia Studio 7 ve verzi Trial. Jedná se o program určený pro tvorbu videoprůvodce, který umožňuje jak nahrání obrazovky, tak úpravu videa a také vytvoření menu DVD.

4.1 Prostředí Camtasia Studio 7

Po spuštění se zobrazí pracovní plocha programu. Pracovní plocha je rozdělena na tři dílčí části (Obrázek 12). V levé části se nachází interaktivní okno, kde se zobrazují aktuálně zvolené nástroje pro úpravu videa. Vpravo je obrazovka, která slouží k úpravě videa a také k jeho prohlížení. Na spodní části plochy se nachází časová osa.



Obrázek 12 Pracovní prostředí programu Camtasia Studio 7

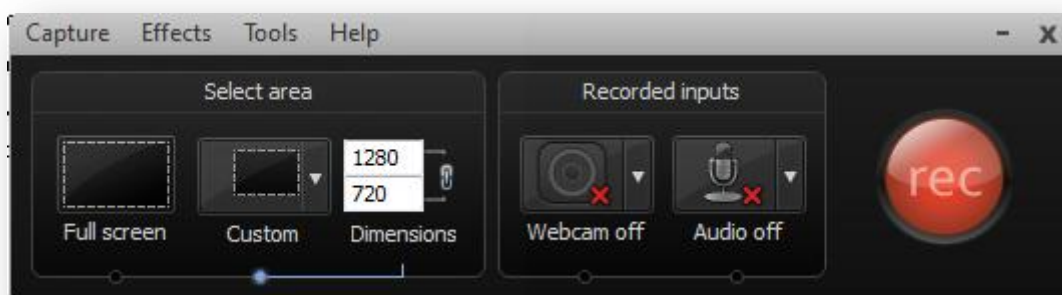
Mezi hlavní nástroje Camtasia Studia 7 patří:

- popisky,
- zoomování,
- přechody,

- nahrávání hlasu,
- efekty kurzoru,
- úprava zvukové stopy,
- titulky, a
- nadpisy klipů.

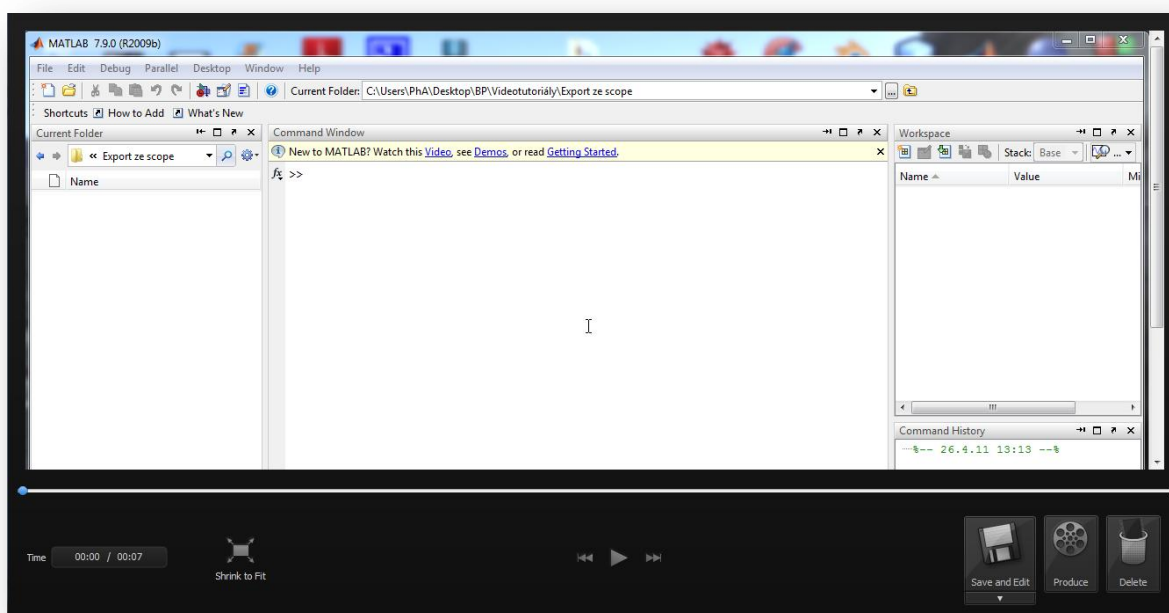
4.2 Nahrávání obrazovky

Pro nahrání obrazovky slouží aplikace Camtasia Recorder (Obrázek 13). Pro spuštění aplikace vybereme položku *Record the screen* nebo také v nabídce *Tools->Studio Tools->Camtasia Recorder*.



Obrázek 13 Camtasia Recorder

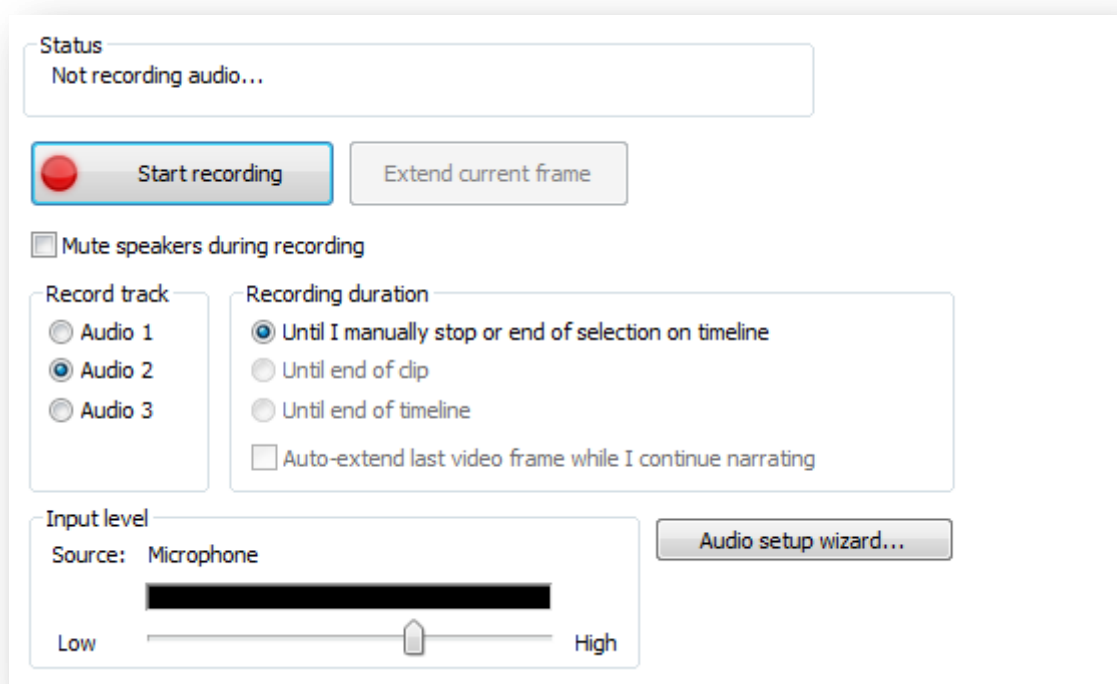
Po spuštění aplikace se zobrazí ovládací panel, který slouží pro nahrávání obrazovky, nahrávání zvuku z mikrofону a popřípadě pro zachycení videa z webkamery. Uživatel si zde zvolí oblast nahrávání z přednastavených možností nebo si může nadefinovat vlastní velikost nahrávací oblasti. Nahrávací oblast, která bude zachycena ve videu, je lemována zelenými přerušovanými čarami a okolí nahrávací oblasti, které zachyceno nebude, je tmavé. Pro ukončení nahrávání obrazovky slouží klávesa F10. Jakmile skončí nahrávání obrazovky, otevře se náhled - Preview (Obrázek 14). Náhled slouží k prohlédnutí nahraného obrazovky a jsou zde na výběr možnosti *Save and Edit*, *Produce* a *Delete*. Při zvolení možnosti *Save and Edit* se nahraná obrazovka uloží a přesune do aktuálně otevřeného projektu k další úpravě. Možnost *Produce* slouží k okamžitému vytvoření videa z nahrané obrazovky a možnost *Delete* slouží k smazání nahraného obrazovky.



Obrázek 14 Camtasia Recorder Preview

4.3 Nahrávání komentářů

Pro nahrání komentářů k videu byl použit nástroj Voice Narration (Obrázek 15). Tento nástroj Camtasia Studia 7 umožňuje efektivní nahrání komentáře na zvolený úsek videa. Stisknutím tlačítka *Start Recording* se spustí nahrávání komentáře, od zvoleného úseku videa. Po celou dobu nahrávání se video přehrává, dokud uživatel neukončí nahrávání tlačítkem *Stop recording*. Po nahrání všech komentářů bylo nutné na zvukovou stopu aplikovat odstranění šumu a zvýšení hlasitosti, které naleznete v nástroji pro úpravu zvukové stopy nebo také v nabídce *Tools->Audio*. Veškeré komentáře se ukládají do zvukového formátu wav.

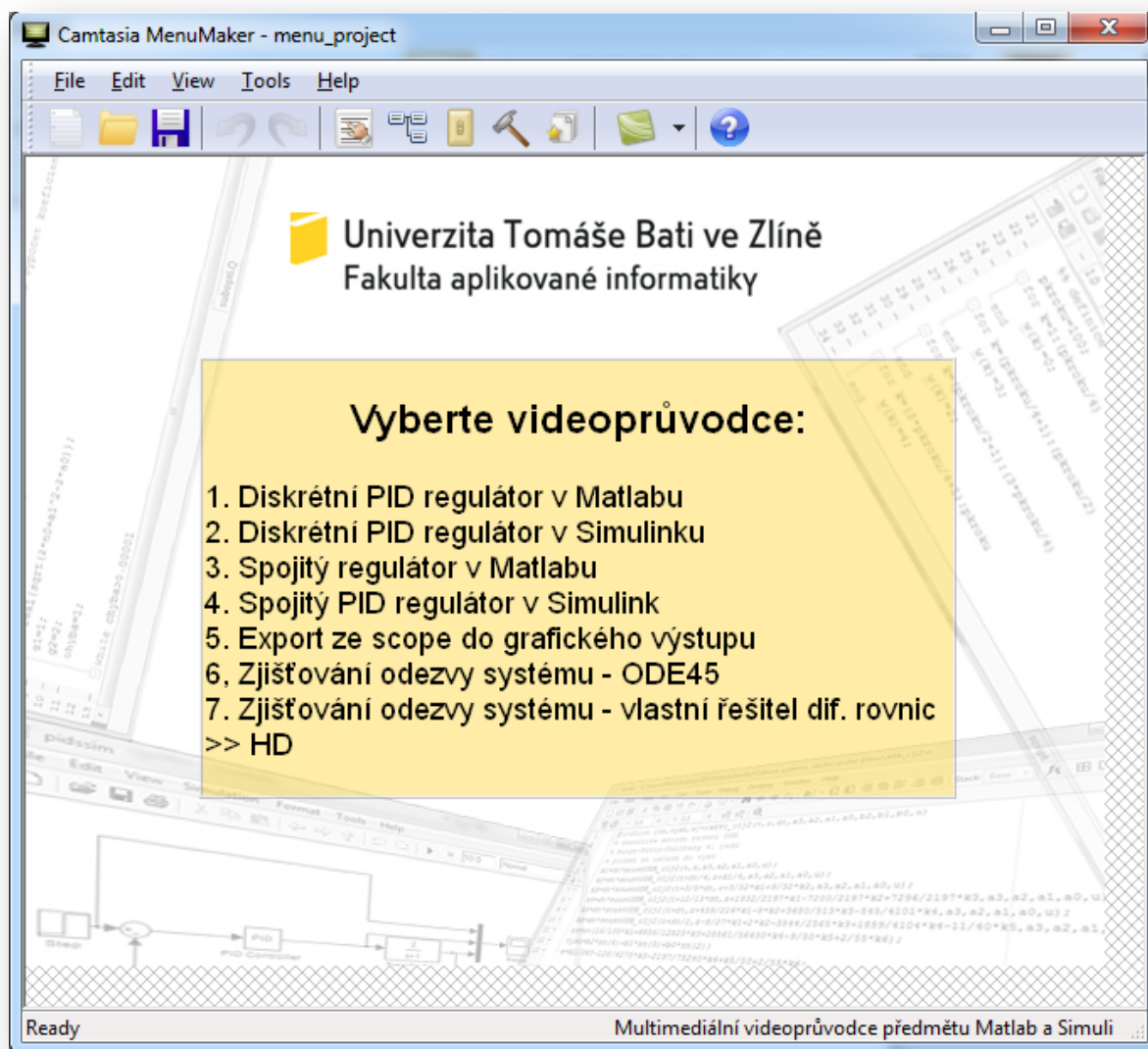


Obrázek 15 Nástroj Voice Narration

4.4 Vytvoření DVD menu

Pro vytvoření menu DVD byla použita aplikace Camtasia MenuMaker (Obrázek 16). Tato aplikace je součástí Camtasia Studio 7 a spouští se přes nabídku menu *Tools -> Studio tools -> Camtasia MenuMaker*. Po spuštění se objeví uvítací dialog, který nabízí možnosti jako vytvoření nového projektu pomocí průvodce, otevření nového projektu, otevření existujícího projektu anebo otevření posledního projektu. Po zvolení první nebo druhé možnosti se otevře knihovna základních 11 šablon. Uživatel si může vybrat jednu z nabízených šablon nebo šablonu nepoužije a zvolí možnost „Prázdná šablona“. Otevře se nový projekt, který lze upravovat pomocí *Menu Properties*. Okno *Menu Properties* obsahuje 4 záložky: *General*, *Cursor*, *List* a *Content*. V záložce *General* lze nastavit titulek menu, barvu pozadí nebo obrázek na pozadí a velikost plátna menu. V záložce *Cursor* lze nastavit vzhled kurzoru, zvuk kurzoru při kliknutí a zvuk kurzoru při pohybu nad objektem. V záložce *List* lze nastavit barva textu (normální, navštívený, zvýrazněný), zarovnání textu, styl rámečku menu a pozici rámečku. Poslední záložka *Content* je

nejdůležitější. Zde se vytvářejí odkazy na video soubory, které chcete zobrazit při kliknutí v menu na daný odkaz.



Obrázek 16 Camtasia MenuMaker

4.4.1 Spuštění a práce s DVD

Menu, které se nachází na DVD, se po vložení do mechaniky počítače automaticky spustí. Pokud se tak nestane, lze toto menu spustit manuálně a to otevřením obsahu DVD a spuštěním souboru *videoprůvodce.exe*.

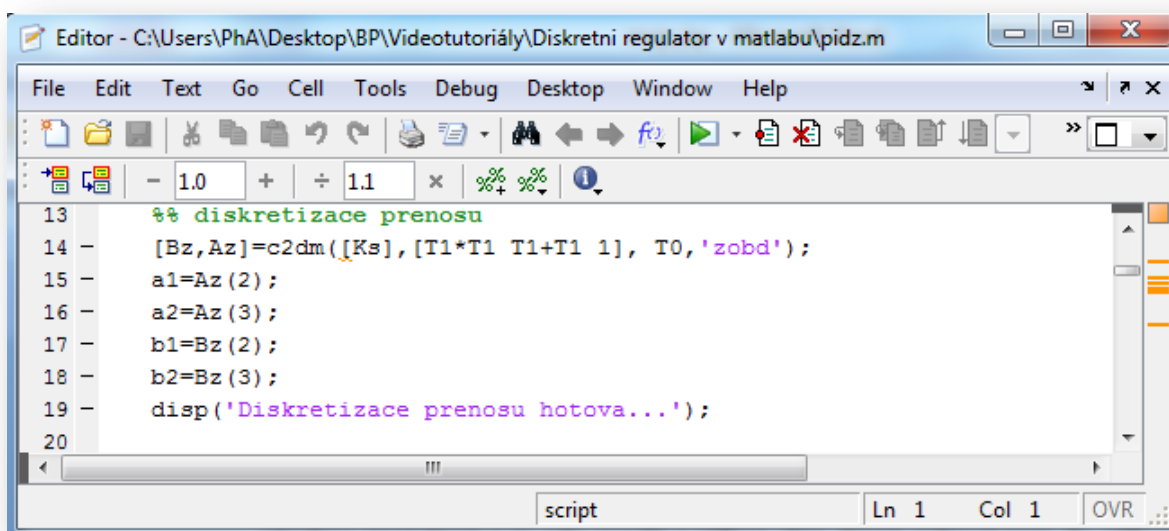
Menu je řešeno jednoduše a intuitivně (Obrázek 16). Obsahuje titulek a celkem sedm odkazů. Odkaz, na který ukazuje kurzor myši, se zvýrazní tyrkysovou barvou. Po kliknutí myši na vybraný odkaz se spustí dané video z videoprůvodce v obrazovém formátu 4:3 s rozlišením 720x480 pixelů. Toto rozlišení je nastaveno z důvodu optimalizace pro širokou škálu monitorů či projektorů. Po kliknutí myši na text HD, který se nachází v poslední části menu, se otevře sub-menu. Toto sub-menu slouží pro spuštění jednotlivých videí z videoprůvodce v širokoúhlém obrazovém formátu 16:9, v rozlišení 1280x720. V širokoúhlém obrazovém formátu je kvalita videa vyšší, než u klasického formátu 4:3.

5 POPIS JEDNOTLIVÝCH VIDEOPRŮVODCŮ

5.1 VP 1 – Diskrétní PID regulátor v MATLABu

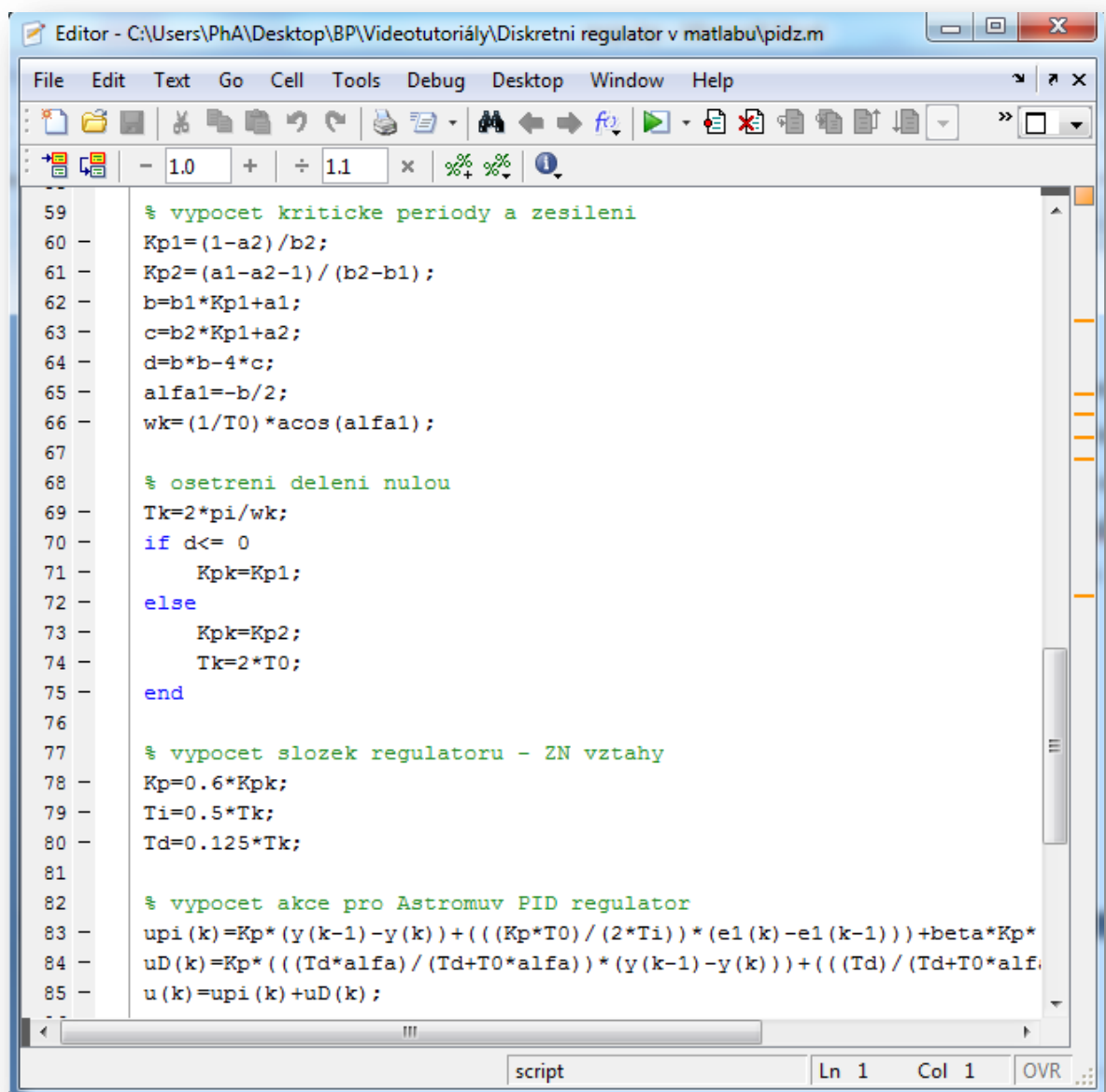
Uživatel se seznámí s problematikou diskrétního PID regulátoru v MATLABu pomocí skriptu, který je rozdělen na šest dílčích částí.

Na počátku je vymazána obrazovka příkazového okna, z *Workspace* jsou vymazány všechny proměnné a je zde použit i příkaz *disp*, který zobrazí text do příkazového okna. Dále je definován přenos systému a perioda vzorkování. Následuje diskretizace přenosu (Obrázek 17) systému pomocí funkce *c2dm* a je zde použita zpětná obdélníková metoda (ZOBD).



Obrázek 17 Ukázka zdrojového kódu – diskretizace přenosu

Definuje se hodnota žádané veličiny pomocí čtyř *for* cyklů a inicializují se proměnné, se kterými se dále pracuje. Následuje nejdůležitější část skriptu pro výpočetní cyklus regulátoru, který je také ve *for* cyklu. Ten obsahuje výpočet kritické periody a kritického zesílení, ošetření dělení nulou, výpočet parametrů regulátoru pomocí Ziegler-Nichols vztahů a výpočet akce pro Aströmův PID regulátor. Poslední část skriptu se stará o vykreslení regulačního pochodu regulátoru.



The image shows a MATLAB script editor window titled "Editor - C:\Users\PhA\Desktop\BP\Videotutoriály\Diskretní regulátor v matlabu\pidz.m". The script contains the following code:

```
59 % vypocet kriticke periody a zesileni
60 Kp1=(1-a2)/b2;
61 Kp2=(a1-a2-1)/(b2-b1);
62 b=b1*Kp1+a1;
63 c=b2*Kp1+a2;
64 d=b*b-4*c;
65 alfa1=-b/2;
66 wk=(1/T0)*acos(alfa1);
67
68 % osetreni deleni nulou
69 Tk=2*pi/wk;
70 if d<= 0
71     Kpk=Kp1;
72 else
73     Kpk=Kp2;
74     Tk=2*T0;
75 end
76
77 % vypocet slozek regulatoru - ZN vztahy
78 Kp=0.6*Kpk;
79 Ti=0.5*Tk;
80 Td=0.125*Tk;
81
82 % vypocet akce pro Astromuv PID regulator
83 upi(k)=Kp*(y(k-1)-y(k))+((Kp*T0)/(2*Ti))*(e1(k)-e1(k-1))+beta*Kp*
84 uD(k)=Kp*((Td*alfa)/(Td+T0*alfa))*(y(k-1)-y(k))+((Td)/(Td+T0*alf
85 u(k)=upi(k)+uD(k);
```

The status bar at the bottom indicates "script", "Ln 1", "Col 1", and "OVR".

Obrázek 18 Ukázka zdrojového kódu – hlavní výpočetní cyklus

5.1.1 Použité funkce nebo příkazy a jejich význam

clc – výmaz příkazového okna

disp – výpis textu do příkazového okna

clear all – výmaz všech proměnných uložených ve Workspace

c2dm – funkce pro diskretizaci přenosu

acos – funkce pro inverzní cosinus

figure – otevře nové grafické okno a učiní jej aktivním

close – uzavření aktuálního grafického okna

hold off – vykreslení grafu s vymazáním předchozího

hold on – vykreslení více grafů do jednoho

plot – funkce pro vykreslení grafického výstupu

legend – funkce pro zobrazení legendy grafu

xlabel – funkce pro zobrazení popisu osy x u grafu

ylabel – funkce pro zobrazení popisu osy y u grafu

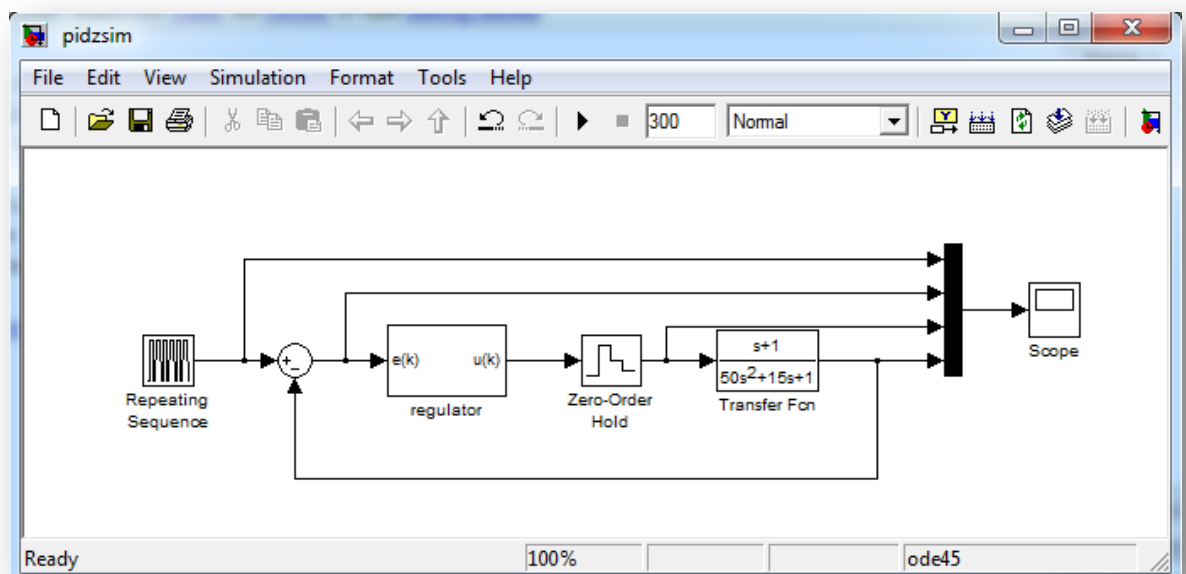
5.2 VP 2 – Diskrétní PID regulátor v Simulinku

V tomto videu se uživatel naučí, jak vytvořit model diskrétního PID regulátoru v Simulinku. Jedná se o první video, ve kterém se pracuje v Simulinku, a proto jsou na začátku ukázány dva způsoby, jak Simulink spustit. Po spuštění Simulinku se vytvoří nové okno a začíná se vytvářet model diskrétního PID regulátoru. Jako první se vytvoří schéma regulátoru a na toto schéma se aplikuje maska subsystému. Po vytvoření subsystému se vkládají další bloky z knihovny Simulinku pro sestavení diskrétního PID regulátoru. V poslední části videa se vymění blok *Step* za blok *Repeating Sequence*, ve kterém se definuje vlastní průběh žádané veličiny.

5.2.1 Použité bloky knihovny Simulinku

Tento videoprůvodce používá tyto bloky:

- step,
- repeating sequence,
- sum,
- unit delay,
- gain,
- zero-order hold,
- transfer fcn,
- mux,
- scope.



Obrázek 19 Model diskrétního PID regulátoru v Simulinku

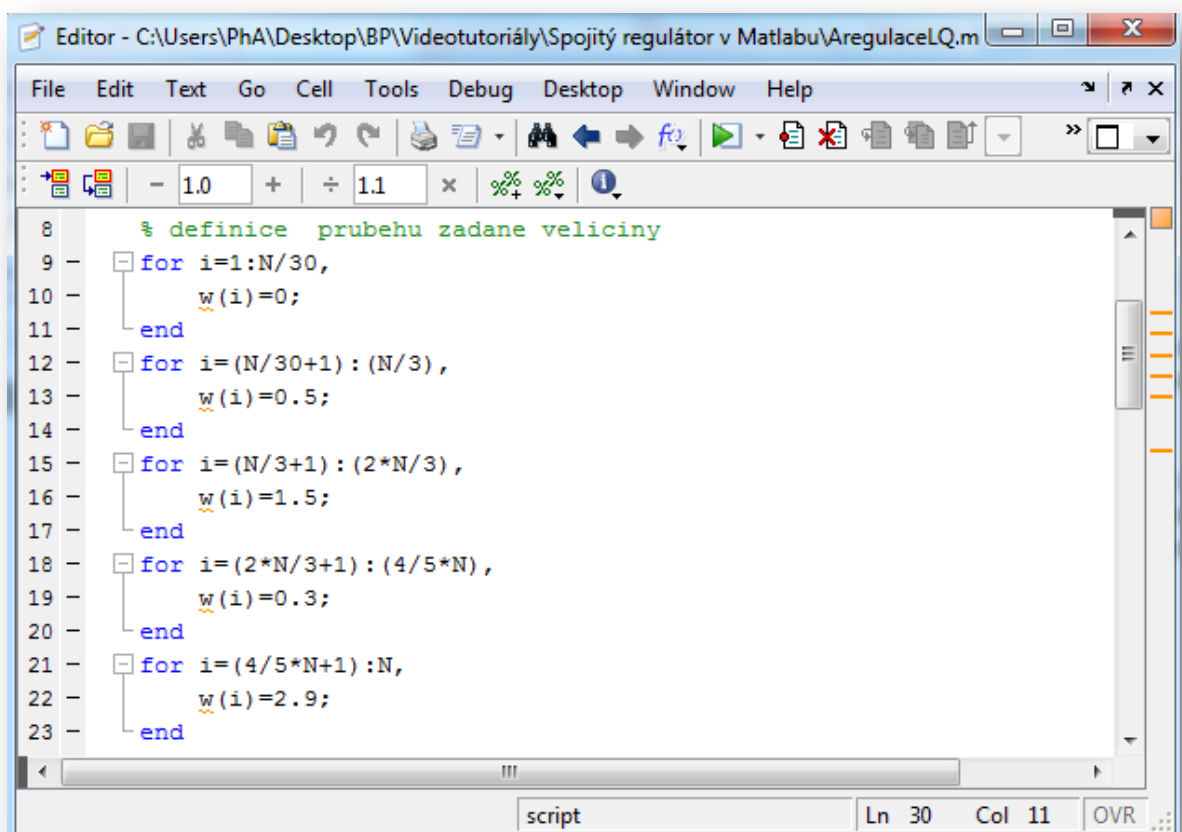
5.3 VP 3 – Spojitý regulátor v MATLABu

Tento videoprůvodce se skládá z pěti M-souborů:

- rkf4r_c2j3contr.m,
- soustODR_c2j3contr.m,
- soustODR_c0j2.m,
- suboptLQ.m, a
- AregulaceLQ.m,

z nichž první čtyři M-soubory jsou funkce, které jsou použity v hlavním skriptu *AregulaceLQ.m*.

Na začátku hlavní skriptu (*AregulaceLQ.m*) se vymažou proměnné ve *Workspace* a nadefinovaly se globální proměnné a doba regulace. Doba regulace je stanovena na hodnotu 1500, ale ve skutečnosti bude celková doba regulace 10x menší, tedy 150 sekund. Je to z důvodu realizace spojitě oblasti. V další části skriptu se nadefinoval čas počátku a konce, časové vektory a časový krok. Poté se inicializují proměnné, se kterými se bude v další části skriptu pracovat. Pro výpočet parametrů regulátoru je nutné nejprve nadefinovat parametry pro inicializaci regulátoru, parametry pro inicializaci soustavy a váhové koeficienty. Poté se použije funkce *suboptLQ* a pomocí této funkce se vypočítají parametry regulátoru. Dále se vypočítá akční zásah, kde je využita funkce *rkf4r_c2j3contr*, která slouží jako vlastní řešitel diferenciálních rovnic. V neposlední řadě se počítá regulační zásah a regulační odchylka. Na konci hlavního skriptu se pomocí funkce *plot* vykreslí regulační pochod regulátoru.



Obrázek 20 Ukázka kódu – definice průběhu žádané veličiny

5.3.1 Použité funkce nebo příkazy a jejich význam

clear all – výmaz všech proměnných uložených ve Workspace

global – definice globální proměnné

linspace – generování lineárně rozložených vektorů

plot – funkce pro vykreslení grafického výstupu

legend – funkce pro zobrazení legendy grafu

xlabel – funkce pro zobrazení popisu osy x u grafu

ylabel – funkce pro zobrazení popisu osy y u grafu

real – funkce, která vrátí reálnou část komplexního čísla

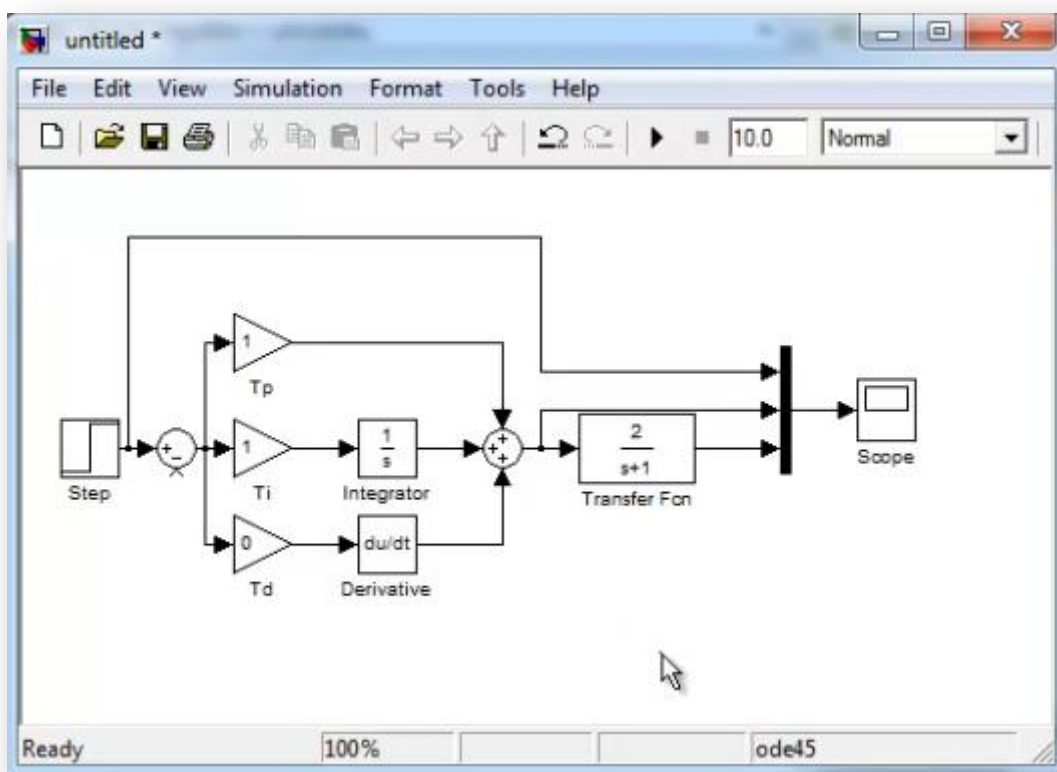
sqrt – funkce, která vrátí odmocninu

zeros – funkce, která vynuluje pole

function – příkaz, který deklaruje funkci

5.4 VP 4 – Spojitý PID regulátor v Simulinku

Ve stejném duchu jako je vytvořeno video 2 (kapitola 5.2) je vedeno i toto video. Uživatel se naučí, jak vytvořit model spojitého PID regulátoru v Simulinku. Model se nejprve sestaví z jednotlivých bloků knihovny Simulinku, poté se aplikuje maska subsystému a v poslední části videoprůvodce se subsystém, který tvoří regulátor, nahradí speciálním blokem *PID Controller*, který nahrazuje PID regulátor.



Obrázek 21 Ukázka modelu spojitého PID regulátoru VYMĚNIT

5.4.1 Použité bloky knihovny Simulinku

V tomto videoprůvodce jsou použity tyto bloky:

- step,
- sum,
- gain,
- integrátor,
- derivative,
- transfer fcn,
- mux,
- scope,
- pid controller.

5.5 VP 5 – Export ze Scope do grafického výstupu

Uživatel se naučí exportovat data z bloku *Scope* do grafického výstupu (figure nebo obrázek).

Na začátku je použit libovolný model vytvořený v Simulinku, zde se jedná konkrétně o model spojitého regulátoru. Videoprůvodce seznámí uživatele s nastavením bloku *Scope* tak, aby se uložily data do proměnné ve *Workspace* (Obrázek 22). S touto uloženou proměnnou se dále pracuje a je použita ve skriptu, kde je graficky pomocí příkazu `plot` vykreslen průběh.

5.5.1 Použité funkce nebo příkazy a jejich význam

`clc` – výmaz příkazového okna

`disp` – výpis textu do příkazového okna

`plot` – funkce pro vykreslení grafického výstupu

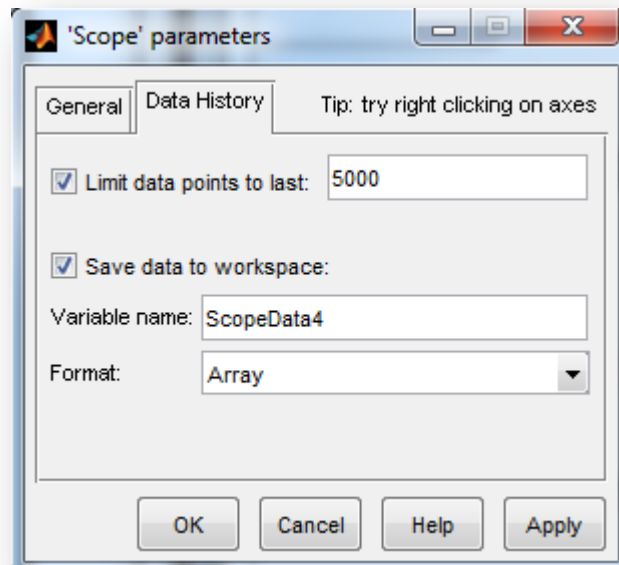
`legend` – funkce pro zobrazení legendy grafu

`xlabel` – funkce pro zobrazení popisu osy x u grafu

ylabel – funkce pro zobrazení popisu osy y u grafu

grid on – zapnutí mřížky v grafu

axis – úprava měřítka os



Obrázek 22 Nastavení bloku Scope

5.6 VP 6 – Zjišťování odezvy systému ODE45

Tento videoprůvodce se skládá ze tří M-souborů:

- odezva.m,
- soustODR_c1j2.m, a
- soustODR_c0j3.m.

Uživatel se naučí, jak zjistit odezvu systému popsaným diferenciální rovnicí a to pomocí vestavěné funkce MATLABu - ODE45. Na začátku hlavního skriptu (odezva.m) se vymažou globální i lokální proměnné, a nadefinují se globální proměnné, se kterými se bude dále pracovat. Pomocí dvou *for* cyklů se nastaví hodnoty výstupního signálu,

nadefinuje se počáteční čas, koncový čas a časové vektory. V další části skriptu se vytvoří koeficienty přenosu systému. Následuje *for* cyklus, ve kterém se počítá soustava 2. řádu s první řádem v čitateli. V tomto *for* cyklu je použita funkce ODE45 a funkce soustODR_c1j2, která řeší pravou stranu diferenciální rovnice. V poslední části skriptu se graficky vykreslí odezva systému pomocí příkazu *plot*. Dále se vytvoří zmiňovaný M-soubor soustODR_c1j2.

V poslední části videoprůvodce se upraví M-soubory pro jiný přenos systému, konkrétně pro soustavu 3. řádu s 1. řádem v čitateli.

5.6.1 Použité funkce nebo příkazy a jejich význam

clear all – výmaz všech proměnných uložených ve Workspace

clear global all – výmaz všech globálních proměnných uložených ve Workspace

linspace – generování lineárně rozložených vektorů

ode45 – řešitel diferenciálních rovnic

figure – otevře nové grafické okno a učiní jej aktivním

plot – funkce pro vykreslení grafického výstupu

5.7 VP 7 – Zjišťování odezvy systému pomocí vlastního řešitele diferenciálních rovnic

Tento videoprůvodce je podobný předchozímu (kapitola 5.6) s tím rozdílem, že zde je použit vlastní řešitel diferenciálních rovnic. Skládá ze tří M-souborů:

- odezva.m,
- soustODR_c1j2.m, a
- rkf4r_c1j2.m

Uživatel se naučí, jak zjistit odezvu systému popsaným diferenciální rovnicí s použitím vlastního řešitele diferenciálních rovnic. Ten se nachází v M-souboru s názvem

rkf4r_c1j2.m a je naprogramován na základě metody numerického řešení diferenciálních rovnic Runge-Kutta-Fehlberg 4. řádu (kapitola 3.2.1). Pravá strana diferenciální rovnice se řeší v dalším M-souboru s názvem soustODR_c1j2.m. Hlavní skript (odezva.m) je totožný se skriptem v předchozím videoprůvodci, s tím rozdílem, že zde je použit vlastní řešitel diferenciálních rovnic na místo vestavěné MATLAB funkce ODE45.

V poslední části videoprůvodce se upraví M-soubory pro jiný přenos systému, konkrétně pro soustavu 4. řádu s 2. řádem v čitateli.

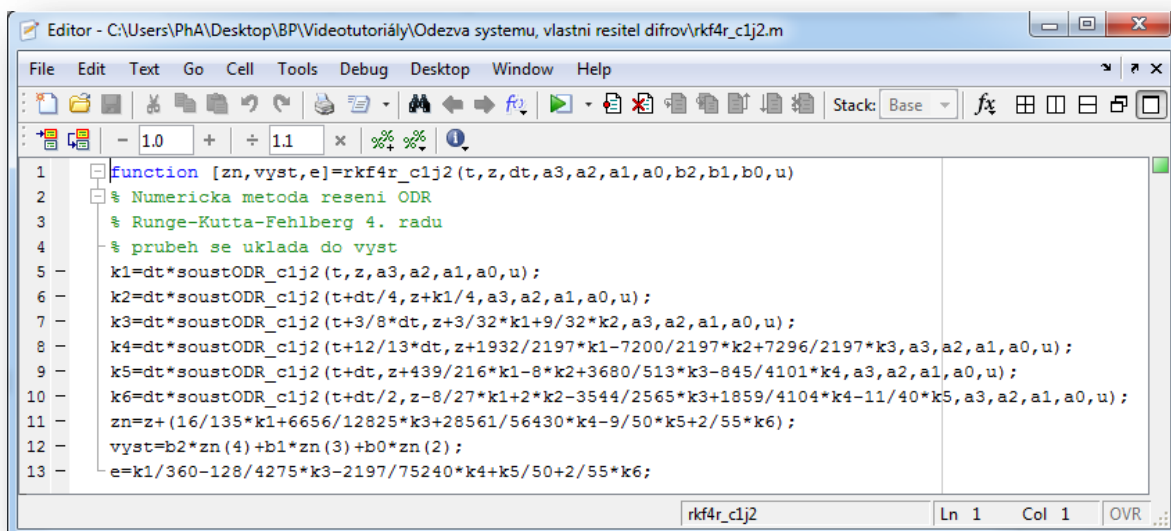
5.7.1 Použité funkce nebo příkazy a jejich význam

clear all – výmaz všech proměnných uložených ve Workspace

linspace – generování lineárně rozložených vektorů

figure – otevře nové grafické okno a učiní jej aktivním

plot – funkce pro vykreslení grafického výstupu



Obrázek 23 Ukázka zdrojového kódu – vlastní řešitel diferenciálních rovnic

ZÁVĚR

Cílem této práce bylo seznámit se s prostředím MATLAB a Simulink a vytvořit multimediálního videoprůvodce z oblasti teorie řízení pro předmět Matlab a Simulink, který je vyučován na Fakultě aplikované informatiky Univerzity Tomáše Bati ve Zlíně. Tento cíl se mi podařilo zvládnout a vytvořil jsem videoprůvodce, který se skládá ze sedmi dílčích videí.

Při tvorbě videí byla snaha o to, aby uživatel, který videoprůvodce shlédne, nejen pochopil danou problematiku, ale aby si danou problematiku i procvičil, a to formou úpravy zdrojového kódu v MATLABu nebo přidáním, odebráním či úpravou bloku v Simulinku. Tento přístup k tvorbě videí jsem zvolil na základě vlastních zkušeností, kdy po vysvětlení probírané látky mi nebylo vše zcela jasné.

Videoprůvodce je uložen na DVD, které je přiloženo k bakalářské práci. DVD obsahuje vlastní menu pro spuštění jednotlivých videí a dále obsahuje složku s názvem *Media*, ve které můžete nalézt jednotlivá videa v normální kvalitě a také v HD kvalitě. Menu je jednoduché a přehledné.

Přínos toho multimediálního videoprůvodce by měl být jak pro studenty, tak pro vyučujícího. Studenti, kteří nepochopili látku dost důkladně při výkladu vyučujícího v hodině, mohou využít tohoto videoprůvodce a osvěžit si paměť nebo přiučit se, co potřebují. Oproti tomu vyučující může šetřit energii při výkladu látky a pustit studentům vybrané video z videoprůvodce. Tento videoprůvodce není pouze pro studenty prezenční formy studia, ale také pro studenty kombinované formy studia. Ti zajisté využijí takové pomůcky při jejich studiu častěji.

ZÁVĚR V ANGLIČTINĚ

Objective of this work was familiar with Matlab and Simulink to create multimedia video-guide in control theory for object Matlab and Simulink, which is taught at the Faculty of Applied Informatics Tomas Bata University in Zlin. I managed to handle this object and I've created video-guide, which consists with seven sub-videos.

When making videos was tendency that the user which viewed a video-guide, not only understand issue, but to make the issue even practiced. It was achieved by modifying the source code in Matlab or by adding, removing or modifying block in Simulink. This access to making video is based on my own experience when, after an explanation of the discussed topics I was confused.

Video-guide is stored on DVD, which is attached to the bachelor thesis. DVD contains a custom menu to start each video and includes a folder called *Media*, which you can find individual videos in normal quality and in HD quality. The menu is simple and easy to use.

The benefits of video-guide should be for both students and teachers. Students, who don't understand well enough subjects in the interpretation of the teacher in class, can use this video-guide and refresh their memory or learn what they need to know. In contrast this, teacher can save energy in the interpretation of the subject and let the students play video from video-guide. This video-guide isn't just for full-time students, but also for students combined study. They certainly appreciate the tools of their studies more often.

SEZNAM POUŽITÉ LITERATURY

Internetové zdroje:

- [1] Humusoft: Matlab & Simulink [online]. [cit. 2011-02-19]. Dostupný z:
<http://www.humusoft.cz/produkty/matlab>
- [2] Wikipedie, otevřená encyklopedie: MATLAB [online]. [cit. 2011-02-19].
Dostupný z: <http://cs.wikipedia.org/wiki/MATLAB>
- [3] Základy automatizace [online]. [cit. 2011-05-10]. Dostupný z:
http://www.elearn.vsb.cz/archivcd/FS/Zaut/Skripta_text.pdf
- [4] Numerické metody I [online]. [cit. 2011-05-15]. Dostupný z:
<http://k315.feld.cvut.cz/download/ape1/numerika3.pdf>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

VP	Videoprůvodce.
DVD	Digital video disc.
GUI	Graphical user interface.
HD	High definition.
ZOBD	Zpětná obdélníková metoda

SEZNAM OBRÁZKŮ

Obrázek 1 Pracovní plocha prostředí MATLAB R2009b	15
Obrázek 2 Ukázka panelu Workspace při použití příkazu „ $A = 1 + 1$ “	16
Obrázek 3 Ukázka panelu historie příkazů	16
Obrázek 4 Detail na tlačítko START	17
Obrázek 5 Okno M-souboru se zvýrazněným tlačítkem spuštění	18
Obrázek 6 Okno Simulink Library Browser (Prohlížeč knihoven)	20
Obrázek 7 Okno grafického prostředí Simulink	20
Obrázek 8 Vlevo označení bloků pro vytvoření subsystému, vpravo vytvořený subsystém	22
Obrázek 9 Ukázka interaktivního grafického rozhraní pro ladění regulátoru	24
Obrázek 10 Ukázka zdrojového kódu videoprůvodce ze serveru Youtube.....	26
Obrázek 11 Určení kritické periody.....	27
Obrázek 12 Pracovní prostředí programu Camtasia Studio 7.....	32
Obrázek 13 Camtasia Recorder	33
Obrázek 14 Camtasia Recorder Preview	34
Obrázek 15 Nástroj Voice Narration	35
Obrázek 16 Camtasia MenuMaker	36
Obrázek 17 Ukázka zdrojového kódu – diskretizace přenosu	38
Obrázek 18 Ukázka zdrojového kódu – hlavní výpočetní cyklus	39
Obrázek 19 Model diskrétního PID regulátoru v Simulinku.....	41
Obrázek 20 Ukázka kódu – definice průběhu žádané veličiny.....	43
Obrázek 21 Ukázka modelu spojitého PID regulátoru VYMĚNIT.....	44
Obrázek 22 Nastavení bloku Scope	46
Obrázek 23 Ukázka zdrojového kódu – vlastní řešitel diferenciálních rovnic	48

SEZNAM TABULEK

Tabulka 1 Historie vydání MATLABu a Simulinku	14
Tabulka 2 Ziegler-Nicholsovy (2. modifikace) vztahy pro výpočet parametrů regulátoru.....	28

SEZNAM PŘÍLOH

**PŘÍLOHA P I: DVD-ROM (OBSAHUJE VIDEOPRŮVODCE
PŘEDMĚTU MATLAB A SIMULINK)**