ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA ELEKTROTECHNICKÁ

KATEDRA TECHNOLOGIÍ A MĚŘENÍ

BAKALÁŘSKÁ PRÁCE

Freeware simulační a vizualizační nástroje pro GNU/Linux

Roman Pich **2013**

Originál (kopie) zadání BP/DP

Abstrakt

Předkládaná bakalářská práce je zaměřena na vyhledání a zhodnocení volně šiřitelných programů vhodných pro simulaci a vizualizaci elektrotechnických úloh pro začátečnické distribuce OS GNU/LINUX.

Klíčová slova

Linux, Ubuntu, Mandriva, Freemat, Freepascal, Lazarus, Maxima, Octave, Python, Scilab, simulace

Abstract

The present thesis is aiming at correct findings and assessing open source programs suitable for simulation and visualisation of electrical tasks for beginners distribution of OS GNU/Linux.

Key words

Linux, Ubuntu, Mandriva, Freemat, Freepascal, Lazarus, Maxima, Octave, Python, Scilab, simulation

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně, s použitím odborné literatury a pramenů uvedených v seznamu, který je součástí této diplomové práce.

Dále prohlašuji, že veškerý software, použitý při řešení této bakalářské práce, je legální.

V Plzni dne 5.6.2013 Jméno příjmení

……..

Poděkování

Tímto bych rád poděkoval vedoucímu bakalářské práce ing. Martinu Jandovi Ph.D. za odborné vedení a cenné rady při zpracování daného tématu.

Obsah

Úvod

Hlavním úkolem práce je zaměření na vyhledání vhodného volně šiřitelného softwaru k simulaci a vizualizaci elektrotechnických úloh pro začátečnické distribuce operačního systému GNU/Linux včetně popisu jejich instalace a porovnání programů na základě uživatelského komfortu, snadnosti instalace, demonstrace použitelnosti a celkovém zhodnocení softwaru.

Seznam symbolů a zkratek

BIOS ..... basic input-output systém

C++ ..... objektově orientovaný programovací jazyk

CD ..... kompaktní disk

DOS ..... diskový operační systém

DVD ..... digitální video disk

EMACS ..... textový editor

LINUX ..... Unixový operační systém

MATLAB..... Matrix laboratory

MS..... Microsoft

SW ..... software

VIM..... textový editor

1 Volně šiřitelné programy

Svobodný software (freeware) jsou programy, ke kterým je k dispozici také zdrojový kód, spolu s právem software používat, publikovat a dále distribuovat. Vzhledem k rozsahu práv zaručených svobodnou licencí není nabytí svobodné licence podmíněno poskytnutím finančního nebo jiného plnění držiteli autorských práv.[1]

Vyhledání volně šiřitelných simulačních programů vhodných pro elektrotechnické školy v dnešní době není problém. Vybrat ovšem kvalitní simulační program již lze označit za náročný úkol.

Klíčová kritéria, která jsem zvolil pro výběr vhodného simulačního programu:

- ***Snadnost instalace***

Instalace s minimální možností chyb způsobených na straně uživatele. Jestli instalace lze provést pomocí Centra softwaru pro operační systém nebo jinou cestou, například stažením instalačního balíku z internetových stránek poskytovatele programu.

- ***Uživatelský komfort***

Uživatelský komfort hodnotím z pozice uživatele zvyklého na operační systém Microsoft Windows a pouze základními uživatelskými znalostmi o prostředí Linux. Především hodnotím, jestli jsou ovládací prvky umístěny v očekávaném prostoru pracovní plochy daného programu. Dále posuzuji přehlednost a srozumitelnost celé aplikace.

- ***Demonstrace použitelnosti***

Pro demonstraci použitelnosti simulačních programů je vybrán případ regulace proudu v kotvě DC motoru napájeného ze čtyř-kvadrantového pulsního měniče. V této části je obsaženo i kritérium porovnání rychlosti výpočtu zadané simulace.

- ***Celkové hodnocení***

Zde bude komplexní shrnutí vlastností daného programu a posouzení použitelnosti pro zadanou simulaci.

Požadavky výše vyjmenované uplatním jak v základních simulačních nástrojích, taktéž i ve vyšších simulačních nástrojích.

1.1 Linux

Nejdříve všeobecně o operačním systému Linux a jeho bohaté historii. Zakladatel Linus Torvalds začal vyvíjet jádro Linuxu v roce 1991. Mezi hlavní důvody pro vznik právě unixového systému patřil faktor, že Unix je systém, který upřednostňuje jednoduchost a je přednášen na univerzitách. Torvalds byl dále inspirován MINIXem od Andewa Tanenbauma,

který napsal svoji verzi unixovového systému jako doprovodný projekt ke své výuce a knihám o operačních systémech. Na rozdíl od něj však Linux nevyužil svůj projekt komerčně, raději preferoval otevřený vývoj, tedy open source software.[2]

1.1.1 Ubuntu

Verze SW: 12.10

Ubuntu je jedna z nejrychleji se rozvíjejících a současně nejrozšířenější linuxová distribuce na světě. Za svoji krátkou dobu od jejího vzniku, vznikl v roce 2004, si operační systém Ubuntu dokázal získat širokou řadu uživatelů. Nejvíce tomu napomohl zřejmě fakt, že linuxová verze systému pod tímto označením si mezi uživateli vybudovala pověst distribuce vhodnou i pro úplné začátečníky.[3]

Všechny následující programovací jazyky a simulační programy jsem nainstaloval a poté dále testoval v již zmíněném uživatelsky přívětivém prostředí Ubuntu.

Postup instalace Ubuntu

Před instalací operačního systému je v každém případě velice důležitá záloha uživatelských dat.

Systém je možno nainstalovat již do vytvořeného oddílu na pevném disku. Doporučuje se před instalací nebo při instalaci vytvořit nový oddíl určený pouze pro systém.

Možností je i instalace v operačním systému od Microsoft Windows, ale ta je považována za nejméně stabilní, proto ji doporučuji pouze v krajním případě. Název instalace do systému Windows nalezneme pod označením Wubi. Například na vyzkoušení systému, ale samotné vyzkoušení systému již nabízí Ubuntu před instalací.

Nejčastější instalace probíhá prostřednictvím média CD nebo DVD. Je proto nutností zkontrolovat konfiguraci v BIOSu, tedy základním nastavení počítače. Kde zvolíme v sekci nejčastěji označenou Boot, pořadí zaváděcích zařízení a na první místo zvolíme jednotku CD/DVD. Do nastavení systému BIOS se vstupuje ihned po spuštění počítače stisknutím speciální klávesy. Nejčastěji se jedná o klávesu Escape, F1, F2, F10 nebo Delete.

Nyní po uložení změn v nastavení a následném restartu počítače se již vložené DVD automaticky načte a zobrazí instalační nabídku Ubuntu. Zde je na výběr několik možností, ze kterých vybereme položku Nainstalovat Ubuntu.

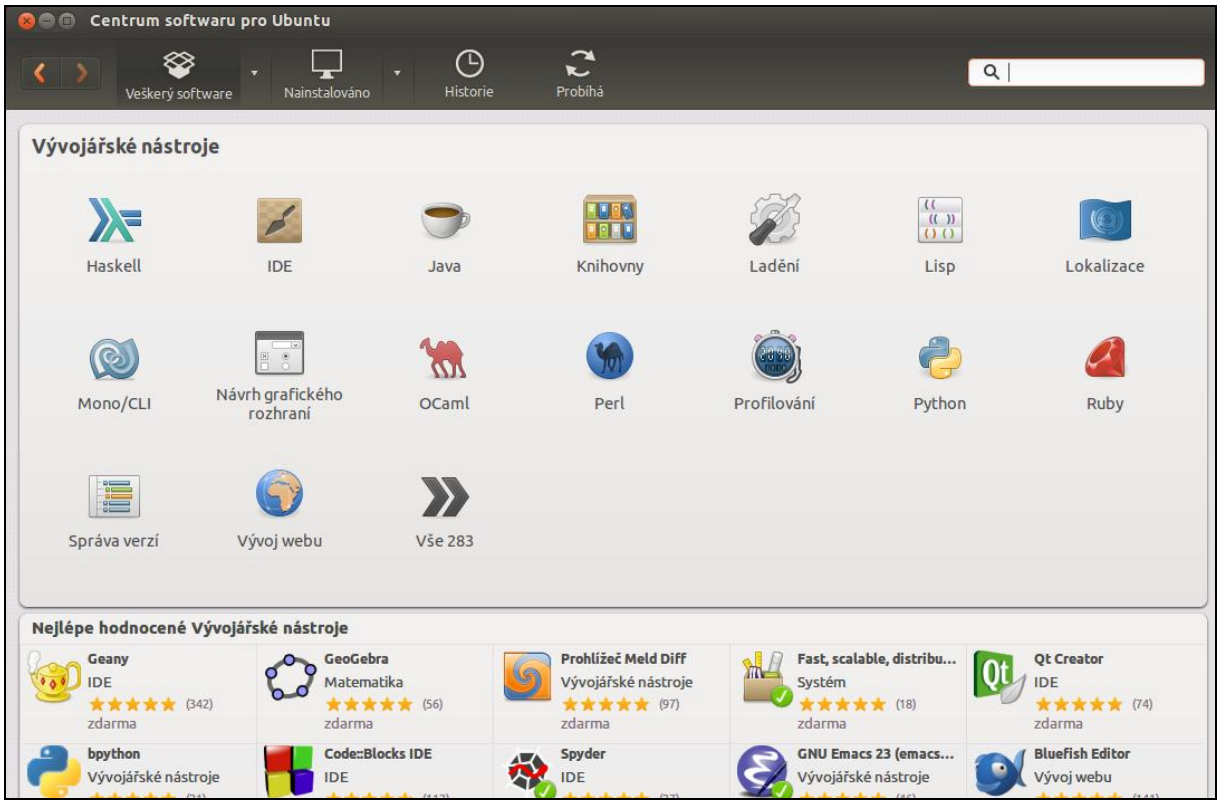
V prvním kroku je nabídnuta volba lokalizace, tedy jazyka, ve kterém bude probíhat celá instalace systému. Dále je na výběr časové pásmo po rozložení klávesnice až po zmíněné rozdělení disku. Poté následuje vytvoření uživatele s nutností vložení hesla, které je nutné při

každé instalaci jakékoliv aplikace nebo přihlášení do systému. U přihlášení do systému lze nutnost vložení hesla dodatečně v nastavení zrušit.[3]

Uživatelský komfort Ubuntu

Plocha po instalaci působí na první dojem stroze, ovšem poskytuje vše co k běžné práci potřebujeme. Dok (Launcher) umístěný vpravo poskytuje jednoduchý přístup k oblíbeným programům. Umožňuje přeskupení, přidávání a třídění programů podle potřeby. Hlavní menu (Dash) poskytuje rychlé a efektivní nalezení souborů nebo programů co hledáme. Pracovní plochy (Workspaces) můžeme různě organizovat a oddělit různě zaměřenou práci od sebe. Centrum softwaru pro Ubuntu umožnuje přístup k tisíci svobodných a hlavně ověřených aplikací, před instalací si již můžeme pročíst jejich hodnocení.[4]

Další samozřejmostí je kvalitní internetový prohlížeč (prohlížeč Firefox). K dispozici je i oblíbený webový prohlížeč Google Chrome ve správci aplikací pod označením Chromium. Ubuntu již má v sobě implementován správce fotografií, hudby a videa.



Obr.č. 1 – Centrum softwaru pro Ubuntu (vlastní zpracování 2013)

1.1.2 Mandriva Linux

Verze SW: 2010.2

Mandriva Linux, dříve pod názvem Mandrake Linux, je operační systém vybaven mnoha aplikacemi pro každodenní použití. Mezi hlavní výhody patří jeho rychlá instalace a snadné základní nastavení. Právě díky vysoké uživatelské přívětivosti patří mezi nejoblíbenější linuxové distribuce na světě. I v České republice patří podle průzkumů mezi nejpoužívanější a nejoblíbenější distribuce.[5]

Postup instalace Mandriva Linux

Nezbytné je záloha všech důležitých uživatelských dat v počítači. Pro instalaci systému musí být BIOS počítače nastaven tak, aby bylo možné zavedení systému z jiného média než z pevného disku.

Před zahájením samotné instalace je vhodné připojit k počítači všechna periferní zařízení, co budeme později využívat, aby došlo k jejich detekci a následné instalaci již během zavedení systému. Nebudeme muset periferie později konfigurovat jednotlivě.

Vložíme instalační DVD nebo CD disk do mechaniky a restartujeme počítač. Po restartu se spustí již úvodní instalační obrazovka, kde je na výběr z několika položek. Vybereme možnost s názvem Install Mandriva Linux Spring on your computer.

Nyní již probíhá instalace v prostředí DrakX, který je hlavní předností operačního systému. K výběru je grafický nebo textový režim instalace. Dále nastavení jazyka, rozdělení disku na oddíly. Další kroky instalace jsou podobné se systémem Ubuntu.[6]

Uživatelský komfort Mandriva Linux

Pracovní plocha vzbuzuje příjemný dojem pro uživatele, zvyklého na systém Windows, protože si je s ním nápadně podobná. Vzhledem podobnost ovšem končí. Mandriva Linux je bezpečný systém, kde nehrozí počítačové viry a spyware. Systém nabízí uspokojivé využití běžným uživatelům i profesionálům, kteří vyžadují po systému vysokou stabilitu a výkon.

1.1.3 Další linuxové distribuce

Zde uvedu pouze průřez dalšími verzemi distribucí pro Linux v abecedním pořadí a jejich stručnou charakteristiku.

Arch Linux - jednoduchá distribuce ve smyslu vnitřní struktury

Debian - jedna z nejstarších verzí, systém vyvíjí hlavně dobrovolníci

Fedora - systém vhodný pro všeobecné použití

Gentoo - distribuce založená na zdrojových kódech

Knoppix - snadná použitelnost a vynikající detekce hardwaru

Suse - víceúčelová distribuce Linuxu

Kubuntu - odvozená od Ubuntu, liší se jen v grafickém prostředí

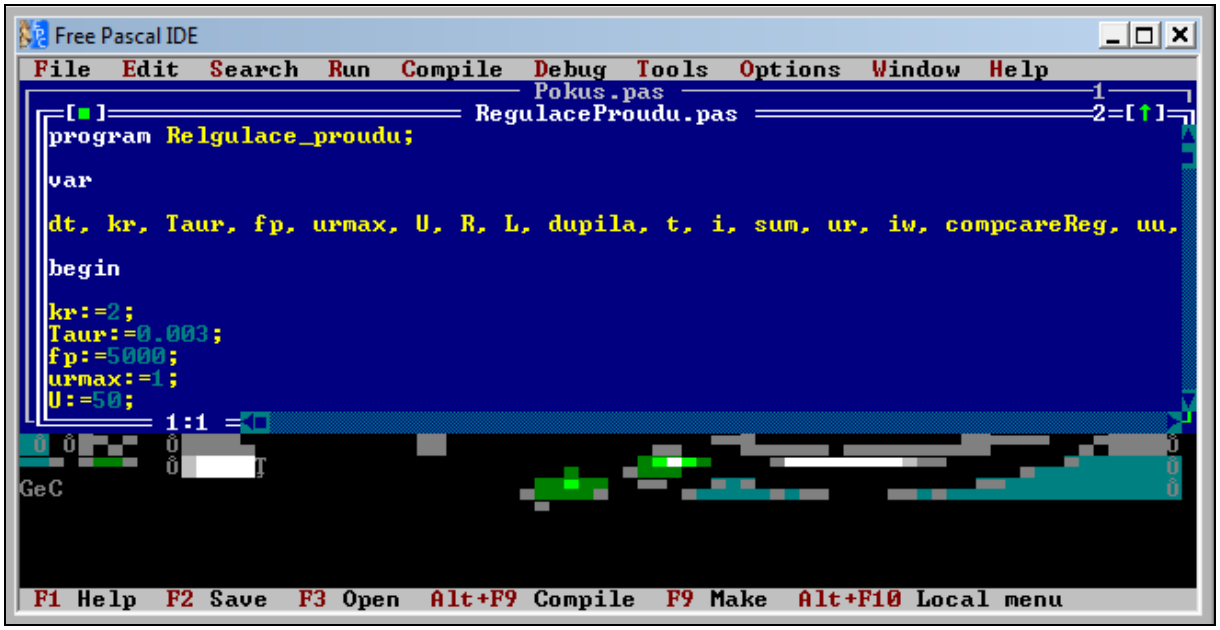
2 Simulační nástroje

2.1 Základní simulační nástroje

2.1.1 Freepascal

Verze SW: 2.6.0

Jedná se o základní programovací jazyk nižší úrovně. Vhodný kompilátor pro zdrojové kódy psané v jazyce Pascal.



Obr.č. 2 – FreePascal (vlastní zpracování 2013)

Popis instalace programu

Program Freepascal nenalezneme v Centru softwaru pro Ubuntu. Aktuální verzi stáhneme na webové adrese http://www.freepascal.org v sekci Download.

Uživatelský komfort

Historický program z dnešního pohledu pro své nízkého textové rozlišení. Problém je složitá deklarace proměnných, která je do značné míry náročná na přesnost. Zdrojový kód neodpustí sebemenší nepřesnost. Z tohoto důvodu je Freepascal dobrý hlavně pro začínající programátory. Chcete-li v průběhu psaní kódu něco odbýt, zpravidla to zabere více času, než to napsat stylisticky správně. Díky tomu si uživatelé vylepšují svůj styl psaní vlastního programu.[7]

Demonstrace použitelnosti

Freepascal se svojí použitelností nepatří k nejjednodušším nástrojům z důvodu vysoké náročnosti k přesnosti deklaraci proměnných. K zobrazení výsledků lze využít Openoffice nebo Libreoffice.

Celkové hodnocení

Textové prostředí v dnešní době již neodpovídá vyžadovaným standardům, přesto stále patří k nejlepším pro jednoduchost a rychlost kompilace. Pro složité elektrotechnické úlohy nevyhovující. Psaní zdrojového kódu nepatří k nejsnadnějším záležitostem.

2.1.2 C/C++

Jazyk vznikl v sedmdesátých letech dvacátého století. Ken Thompson a Dennis Ritchie vytvořili tento programovací jazyk. C patří mezi jazyk nižší úrovně. Není specializovaný na jednu oblast používání, je to univerzální mocný nástroj. Pro mnoho úloh je efektivnější a rychlejší než jiné programovací jazyky.[8]

Postup instalace

V centru pro software Ubuntu nalezneme položku Vim nebo Emacs, jedná se o textové editory, poté vybraný editor nainstalujeme. Dále potřebujeme překladač, nejčastěji se používá Gcc, který bývá součástí všech běžných distribucí Linuxu. Můžeme si vybrat i přívětivé grafické prostředí. Například Adjuntu nebo KDevelop.

GVim editor

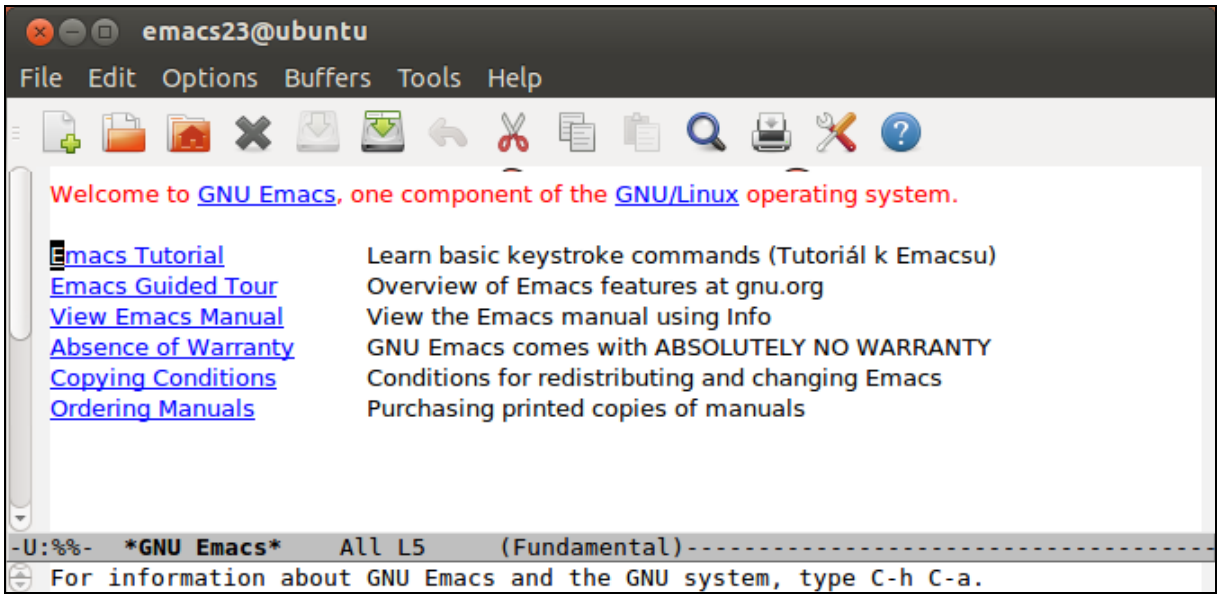
Verze SW: 7.3.547

Velice efektivní víceúčelový textový editor sloužící k přípravě textů. Poskytuje možnost pracovat s více okny najednou. Také je velice zajímavá funkce barevného značení na základě syntaktických pravidel.

Emacs editor

Verze SW: 23.4.1

Textový editor velice specifický, občas je mu vytýkána složitost. Je použitelný jako vývojové prostředí pro mnoho jazyků. V posledních verzích se již nachází přívětivé grafické prostředí.



Obr.č. 3 – Emacs (vlastní zpracování 2013)

KDevelop

Verze SW: 4.4.1

Grafické prostředí primárně zaměřené na vývoj v jazyce C++. Dokáže sám odhalit jednodušší chyby a zvýraznit je, ještě pře spuštěním kompilace. Zajímavostí je označení pro každý příkaz jinou barvou.

Obr.č. 4 – Kdevelop (vlastní zpracování 2013)

Uživatelský komfort

Editor Emacs je velice přívětivý program s velice povedeným grafickým prostředím. Editor Vim patří k povedeným programům. Ovládání má ovšem trochu jiná pravidla, než je běžné. KDevelop velice překvapil detekcí jednoduchých chyb před samotnou kompilací.

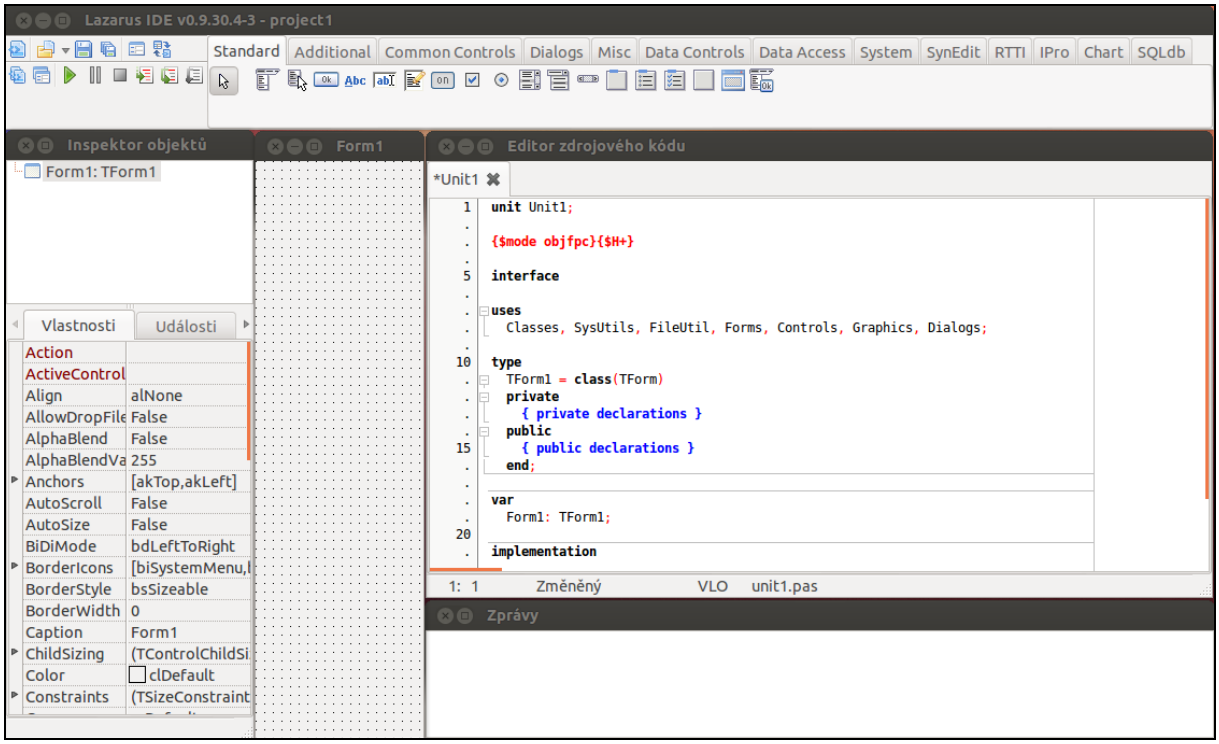
Celkové hodnocení

Práce v KDevelop programu je příjemná. Největší nevýhodou je nutnost použití dalšího programu ke grafické simulaci výsledků.

2.1.3 Lazarus

Verze SW: 0.9.30.4

Program Lazarus, který vychází ze základů programovacího jazyka Pascal je výbornou náhradou klasického Freepascalu nebo Pascalu. Jeho vývoj začal v roce 2001 a hlavním úkolem byla zpětná kompaktibilita s programy Pascal a Deplhi.



Obr.č. 5 – Lazarus (vlastní zpracování 2013)

Postup instalace programu

Do vyhledávacího okna v Centru softwaru pro Ubuntu můžeme zapsat jak název Lazarus, tak i název Pascal, kde se na prvním místě rovněž objeví zmiňovaný program. Instalace rovněž probíhá po vybrání tlačítka instalovat automaticky. Jediné co musíme nastavit je výchozí adresář po prvotním spuštění aplikace po úspěšné instalaci.

Uživatelský komfort

Lazarus jakožto vývojové prostředí pro programovací jazyk Freepascal svojí grafickou stránkou zcela postačuje. Ovšem pro práci a úpravu zdrojového kódu je podle mého názoru složitý a v dnešní době bude pro mnoho studentů nejméně přijatelný programovací jazyk.

Mezi výhody patří rychlá kompilace kódu.

Demonstrace použitelnosti

K výsledné simulaci jsou výsledná data zapsána do souboru a poté data importována ze souboru a následně vyobrazena pomocí Openoffice nebo Liberoffice.

Celkové hodnocení

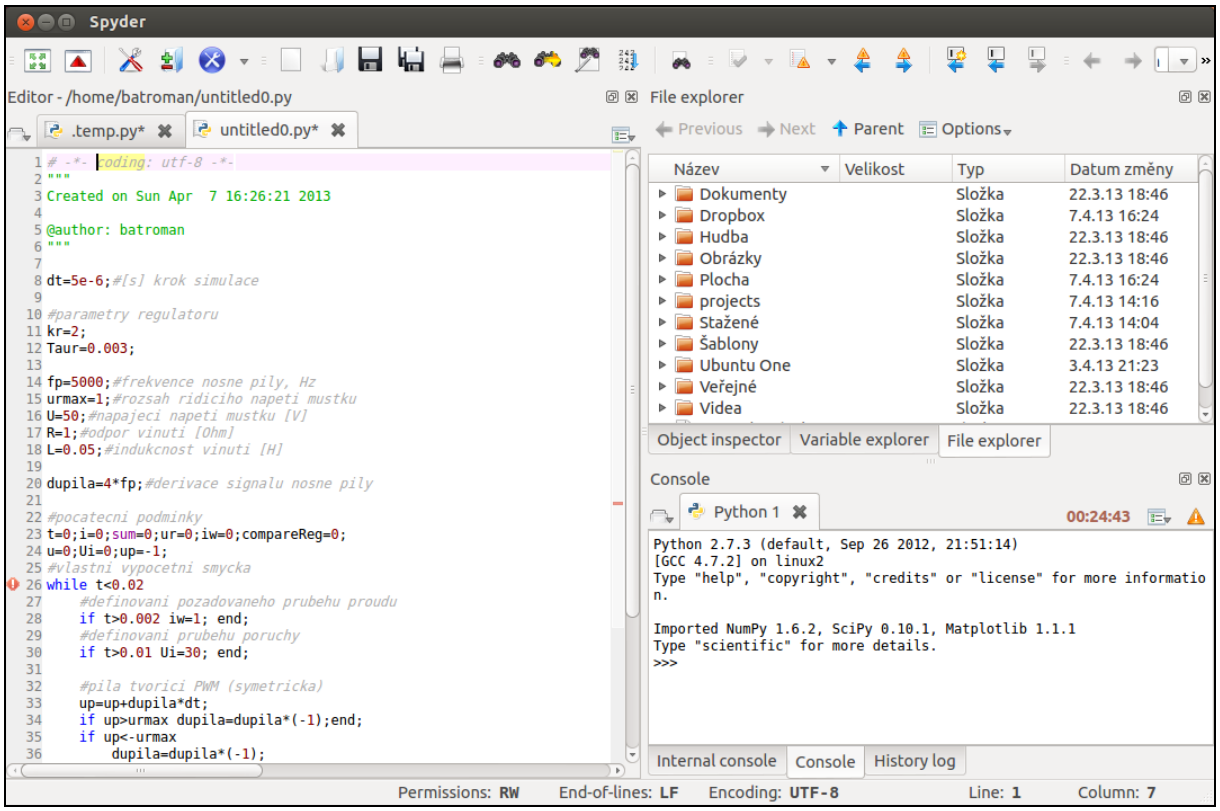
Pro jednoduché úlohy postačující. Složité úlohy jsou v programu Lazarus podle mého názoru, komplikovanou záležitostí.

2.1.4 Python/Spyder

Verze SW: Python 3.3.0 / Spyder 2.1.10

Python je dynamický objektově orientovaný skriptovací programovací jazyk, který v roce 1991 navrhl Guido van Rossum.[9]

Spyder je pokročilé vývojové prostředí pro Python, napsané pomocí frameworku Qt.



Obr.č. 6 – prostředí programu Spyder (vlastní zpracování 2013)

Postup instalace programu

Aplikaci můžeme vyhledat v již zmíněné ikoně Centrum softwaru pro Ubuntu. V nabídce je několik verzí. Po vybrání verze již stačí pouze kliknout na instalaci. Zbytek instalace již proběhne automaticky. Dále nainstalujeme vývojové prostředí Spyder stejným způsobem.

Uživatelský komfort

V programu lze snadno vytvářet nebo upravovat zadaný zdrojový kód, také zde nalezneme barevné označení příkazů, které zvyšuje čitelnost i srozumitelnost celého kódu. Prostředí Spyder je přehledné a logicky uspořádané. Intuitivnost práce a úprava kódu je zde na dobré úrovni. Umístění ikony pro spuštění skriptu není vhodně umístěné.

Celkové hodnocení

Uživatelské prostředí se podle mého názoru méně povedlo oproti propracovanějšímu QtOctave, ale rozložení je podobné. Příkazy jsou zde trochu jiné, to se může často projevovat zmatky v celém kódu. Ve vývojovém prostředí Spyder je jazyk Python vhodný k simulaci jednoduchých i složitých elektrotechnických úloh.

2.2 Zobrazovací programy pro základní simulační nástroje

2.2.1 Openoffice

Verze SW: 3.4.1

Openoffice.org je kancelářský balík obsahující samostatnou sadu programů:

Writer - textový editor

Calc - tabulkový procesor

Impress - nástroj pro prezentace

Draw - grafický editor

Base - databázový editor

Math - matematický nástroj

Vývoj balíku Openoffice.org probíhá pod záštitou Apache Software Foundation. Vzorem se stal komerční balík od společnosti Microsoft s názvem Office. Cílem bylo vytvořit bezplatnou alternativu, která dokáže otevřít i soubory vytvořené v již zmíněném Microsoft Office. Dokonce si dokáže poradit i s poškozenými soubory, které konkurent nedokáže zobrazit.

Postup instalace

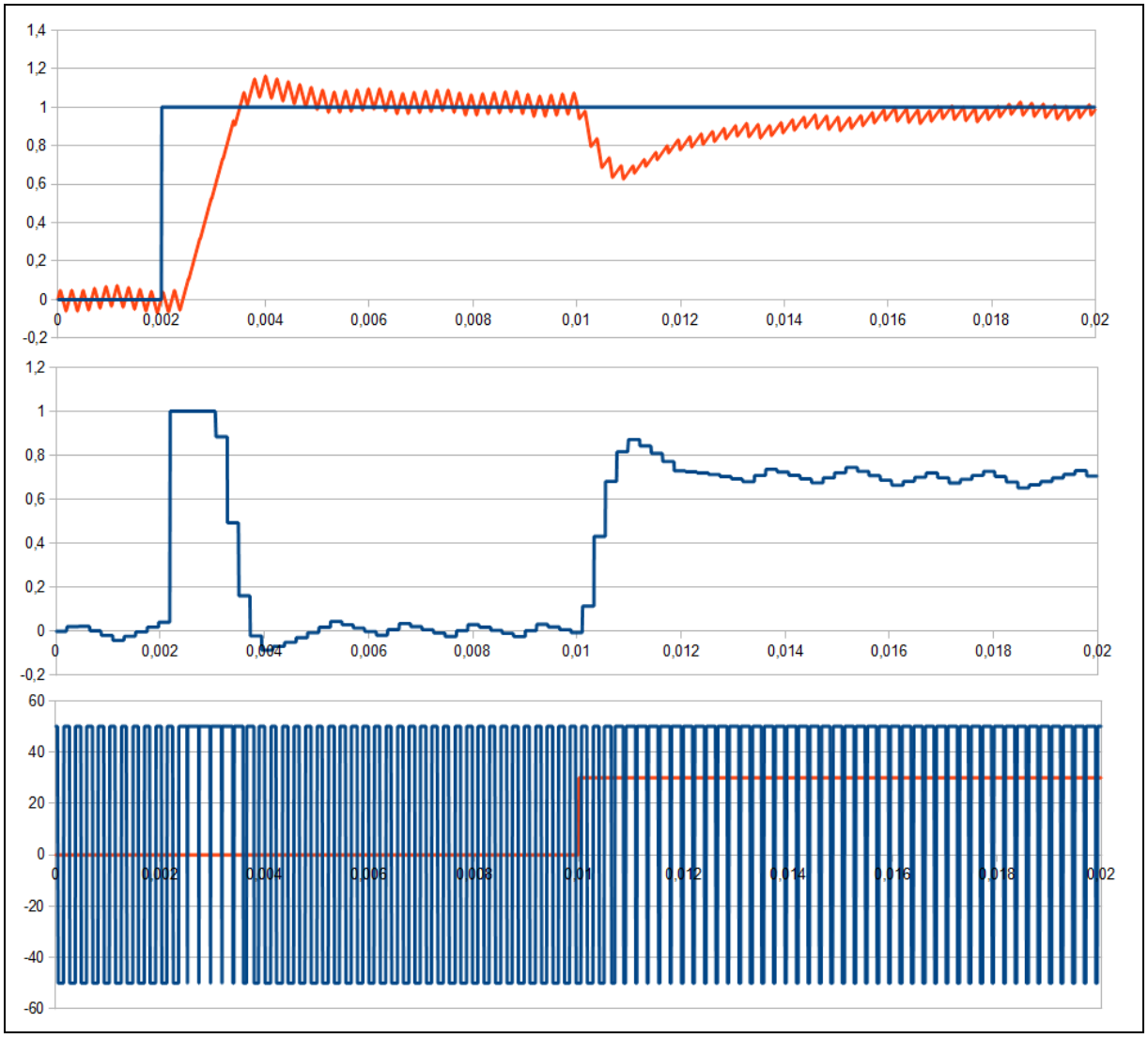
Zde se vyskytl problém podpory Centrum softwaru pro Ubuntu, kde instalační balík již není v nabídce. Musíme se proto obrátit na oficiální stánky vývojářů http://www.openoffice.org, kde v sekci download se nachází verze pro Linux.

Uživatelský komfort

Z celého balíku využiji pouze Calc tabulkový procesor, s jehož pomocí vykreslím výsledky simulací ze základních programovacích jazyků, které nemají grafický výstup implementován v sobě. Aplikace Calc na první pohled zaujme profesionálním vzhledem.

Demonstrace použitelnosti

K zobrazení externích dat jsem použil import dat ze souboru, do kterého jsem předtím zapsal hodnoty ze základního simulačního nástroje.



Obr.č. 7 – demostrace použitelnosti Calc Openoffice (vlastní zpracování 2013)

Celkové hodnocení

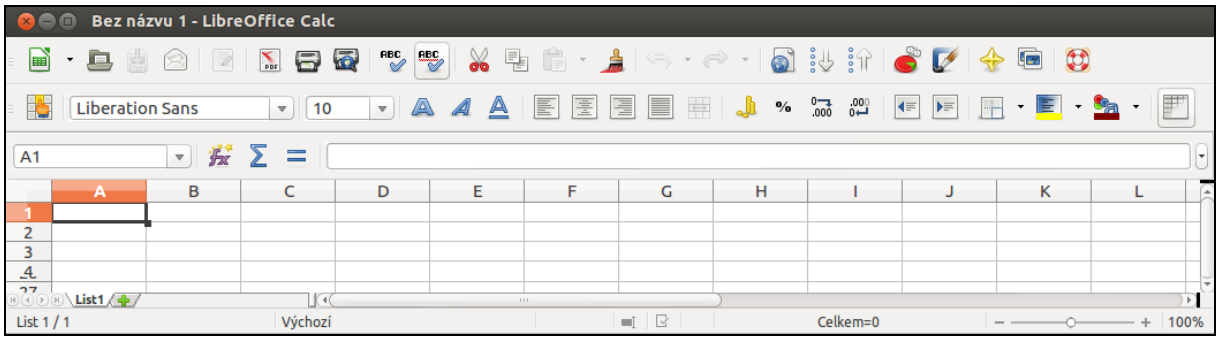
Calc představuje vynikající tabulkový kalkulátor. Splnil základní očekávání a uživatelsky ho považuji za přívětivý. Vykreslení výsledků získaných ze základních simulačních programů je ovšem nekomfortní záležitostí. Pro zobrazení složité simulace je Calc nevhodný nástroj.

2.2.2 Libreoffice

Verze SW: 3.6.2.2

Kancelářský balík, který vznikl poměrně nedávno, přesně v roce 2010 skupina vývojářů opustila projekt Openoffice.org a převzala vývoj kancelářského balíku pod dočasným názvem Libreoffice.[10] Balík obsahuje stejné programy jako Openoffice, tedy Writer, Calc, Impress, Draw, Base a Math.

Pro vykreslení grafického výstupu využiji Calc.



Obr.č. 8 – LibreOffice Calc (vlastní zpracování 2013)

Postup instalace

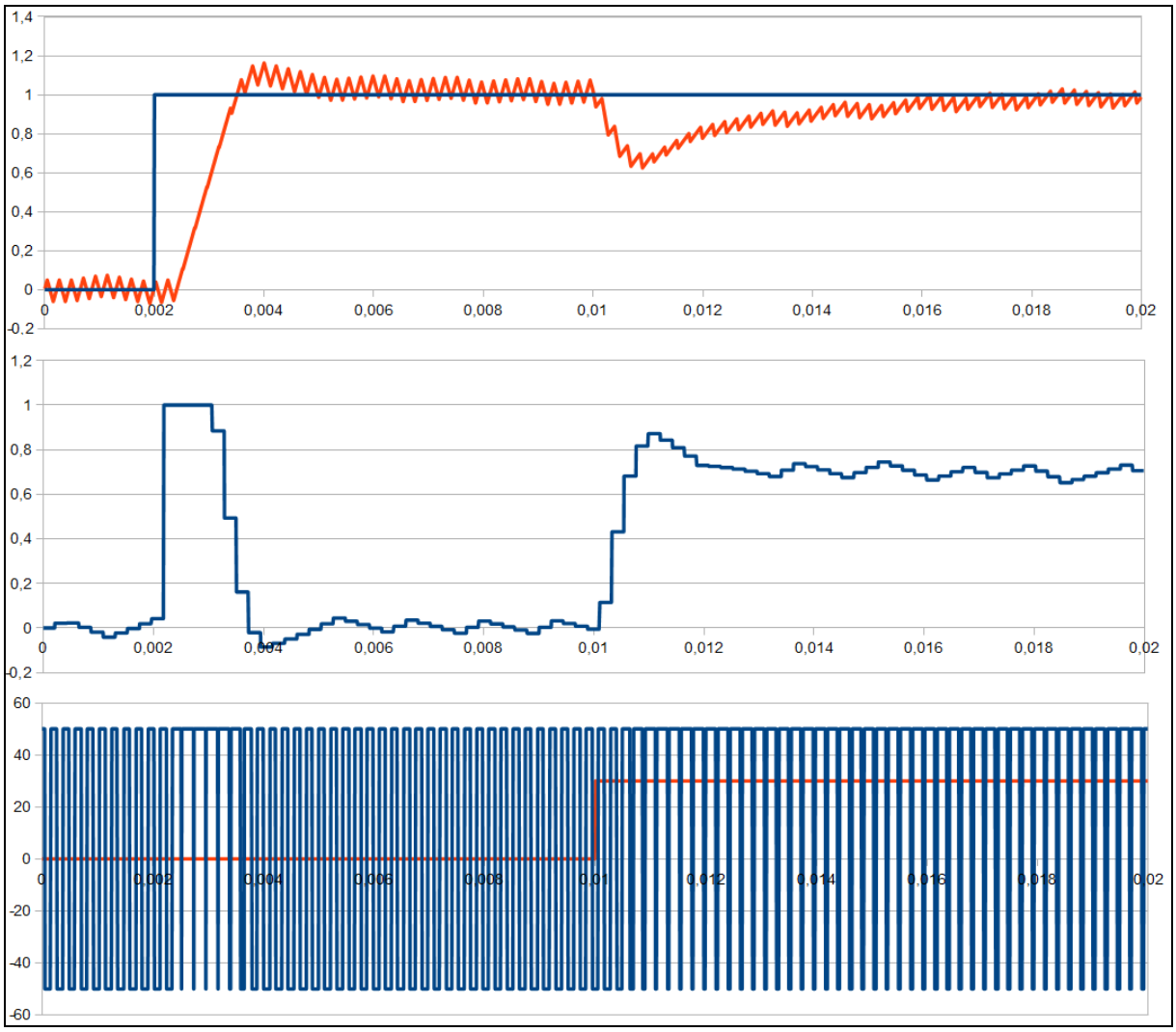
V Centrum softwaru pro Ubuntu nalezneme Libreoffice v záložce Kancelář nebo využijeme vyhledávací okno, kde zadáme celý název programu. Poté opět stačí spustit samotnou instalaci.

Uživatelský komfort

Přehlednost a vysoká efektivnost zdobí tento kancelářský balík. Přehledné ikony programu přispívají k dobré ovladatelnosti celého programu. Průvodce vytvoření grafu je také dobře zpracovaný.

Demonstrace použitelnosti

Při objemném množství dat ke grafickému zpracování dochází k občasnému prodloužení odezvy programu a snižuje se tím komfortnost celé aplikace.



Obr.č. 9 – grafický výstup z aplikace Libreoffice (vlastní zpracování 2013)

Celkové hodnocení

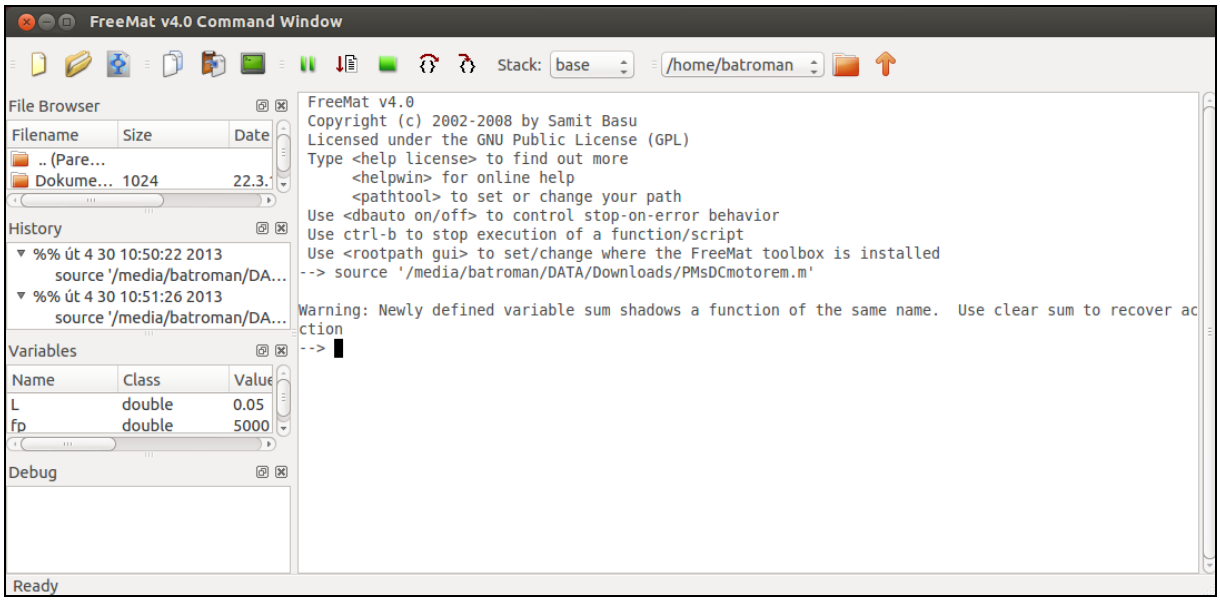
Pro použití k zobrazení výsledků náročné simulace nevhodný program. Použitelný podle mého názoru pouze pro jednoduché simulace.

2.3 Pokročilé simulační nástroje

2.3.1 Freemat

Verze SW: 4.0

Program je vhodný k simulacím a výpočtům složitých matematických operací. Patří do skupiny programů, které se snaží být alternativou k programu Matlab. Uváděna je téměř 95% kompatibilita se zmíněném prostředím Matlab.[11]



Obr.č. 10 – Freemat (vlastní zpracování 2013)

Postup instalace

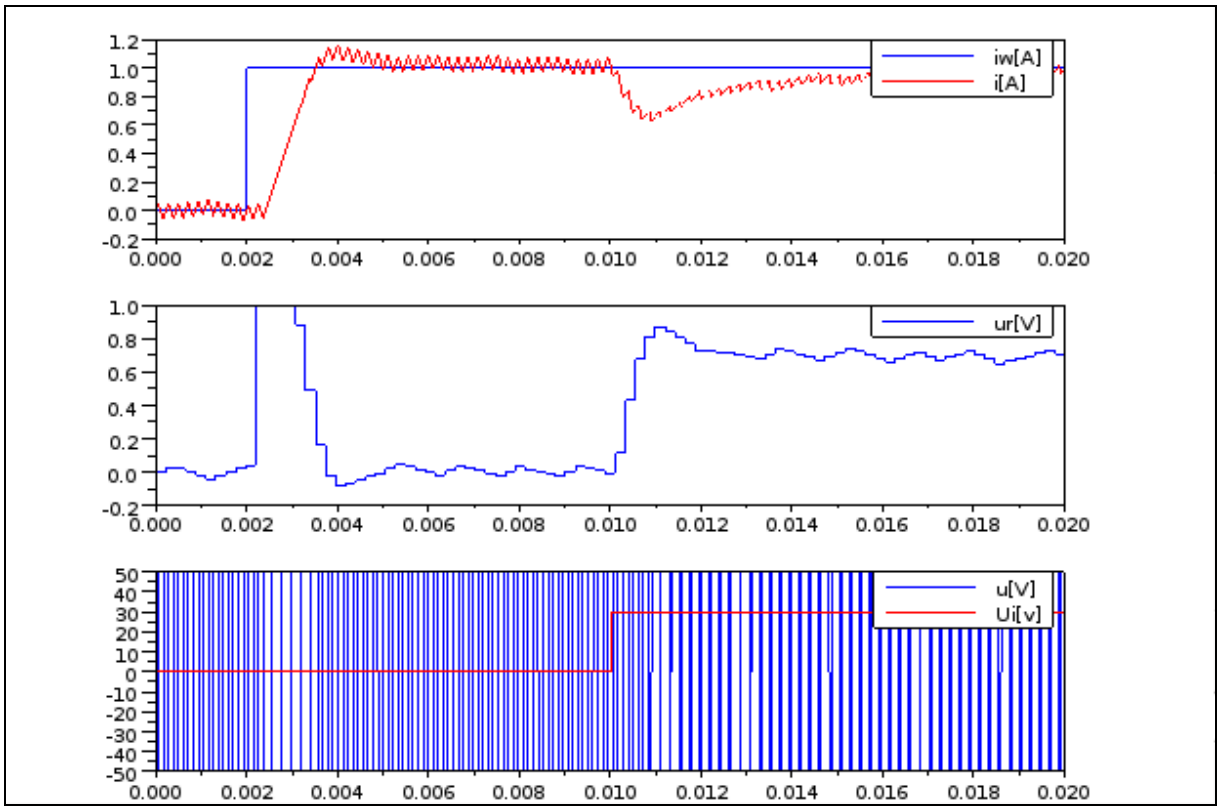
Program se nachází již v nabídce aplikace Centrum softwaru pro Ubuntu. Instalace po potvrzení možnosti nainstalovat již probíhá automaticky.

Uživatelský komfort

Prostředí programu je velice přehledné a logicky uspořádané. Ikony jsou srozumitelné a výborně konfigurované. Orientace v samotném programu nečiní žádné problémy.

Demonstrace použitelnosti

Obrázek číslo 11 deklaruje přímou použitelnost m-file z programu Matlab, bez nutnosti rozsáhlých změn. Pouze popisky os musí být změněny pro správné zobrazení. Při delším nastavení doby simulace se rapidně prodlužuje rychlost výpočtu, která je za daných podmínek mnohonásobně vyšší než u programu Octave nebo Scilab.



Obr.č. 11 – grafický výstup z programu Freemat (vlastní zpracování 2013)

Celkové hodnocení

Delší výpočetní odezva při kompilaci proti programům Octave a Scilab. Dobrý v úpravě a editaci zdrojového kódu. Program celkově vyhovuje pro simulaci jednoduchých i složitých elektrotechnických úloh.

2.3.2 Maxima

Verze SW: 12.04.0

Matematický software vhodný pro řešení symbolických a numerických rovnic s důrazem na symbolické výpočty. Mezi jeho základní vlastnosti patří symbolické integrace a 3D kreslení.

Obr.č. 12 – uživatelské prostředí wxMaxima (vlastní zpracování 2013)

Postup instalace

V centrum softwaru pro Ubuntu program nalezneme pod názvem wxMaxima. Jedná se o verzi, která je vylepšená o přívětivé grafické rozhraní. Instalace po potvrzení položky nainstalovat již probíhá automaticky.

Uživatelský komfort

Prostředí programu působí jednoduše. Je ovšem velice efektivní a pro běžné matematické výpočty dostačující. Rovnice lze jednoduše vkládat a upravovat pomocí příkazové lišty. Velice povedený je editor 2D a 3D grafů.

Demonstrace použitelnosti

Zdrojový kód musí být upraven pro potřeby programu Maxima.

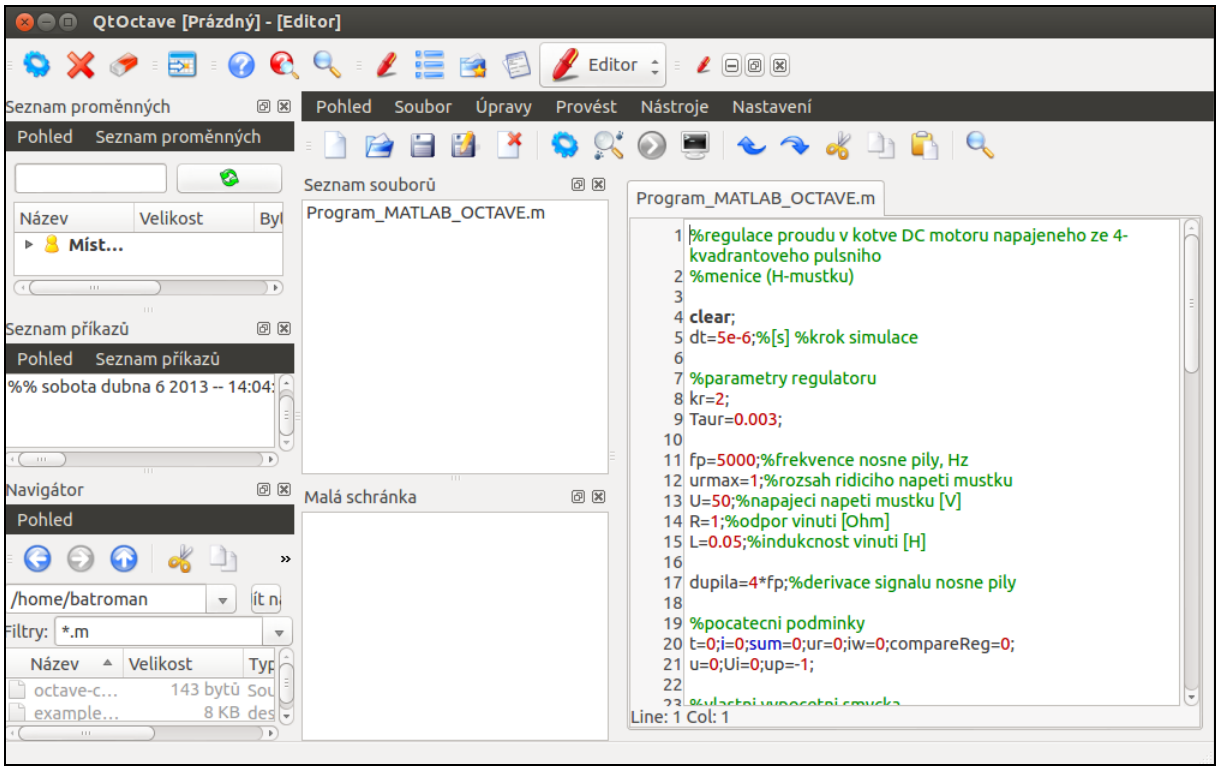
Celkové hodnocení

Pro jednoduché simulace v elektrotechnice vhodný simulační nástroj. Pro složité simulační úlohy není vhodný.

2.3.3 Octave

Verze SW: QtOctave 0.10.1

GNU Octave je vyšší programovací jazyk určený především pro numerické výpočty. Jeho hlavní výhodou je vysoká kompatibilita s programem Matlab. Nejdříve byl projekt Octave použitý pouze jako pomůcka k přednáškám o chemických reakcích. V roce 1992 se začal program upravovat pro uplatnění nejen v oblasti chemie.[12]



Obr.č. 13 – základní prostředí QtOctave (vlastní zpracování 2013)

Postup instalace

Po otevření ikony s názvem Centrum softwaru pro Ubuntu nacházející se na pracovní ploše se objeví nabídka všech dostupných programů pro systém Ubuntu. Nabídka je rozdělena do logických sekcí, podle způsobu použití. Program Octave se nachází v sekci vývojářské nástroje. Nebo pokud víme přímo název programu, rovnou aplikaci můžeme nalézt pomocí vyhledávacího okna umístěného na úvodní stránce Centrum softwaru.

Pro program Octave nainstalujeme dvě části GNU Octave a QtOctave, které je použito pro grafické rozhraní Octave.

Uživatelský komfort

Prostředí je celkově přehledné, protože je rozdělené do jednotlivých sekcí. Výborná je již zmíněná kompaktnost s programem Matlab, tedy odpadá zde občas velice náročná úprava dostupného zdrojového kódu.

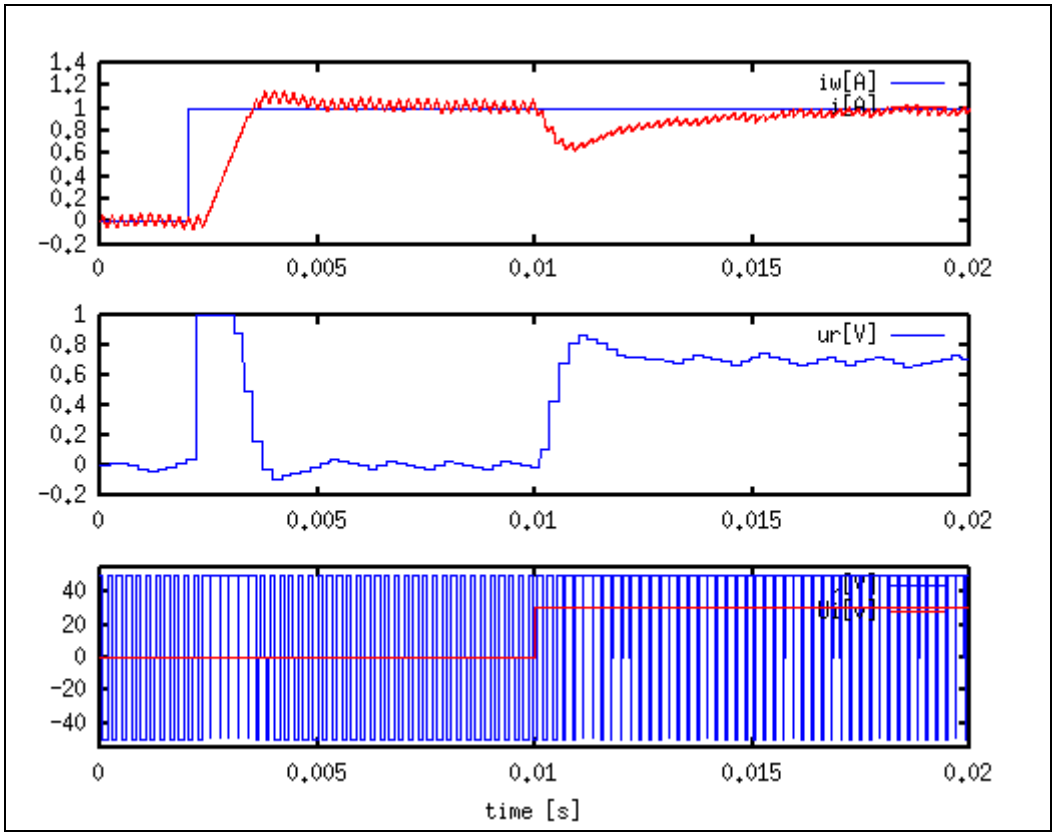
Uspořádání QtOctave vypadá následovně. Pohledy na Seznam proměnných, Seznam příkazů se nachází standardně vlevo spolu s možností úpravy nebo změny pracovní složky s nabídkou filtrování. Uprostřed je umístěn Seznam souborů a pod ním se nachází Malá schránka. Editor se nachází vpravo s možností přepnutí editoru na terminál.

Ikony jsou naprosto srozumitelné a logicky umístěné, takže práce v editoru je velice příjemná a intuitivní.

Odezva při výpočtech u případné korekce je dostatečně rychlá a téměř srovnatelná s prostředím Matlab.

Demonstrace použitelnosti

Na obrázku číslo 14 je zobrazen výsledný grafický výstup z programu, kde je znázorněna regulace proudu v kotvě DC motoru napájeného ze čtyř-kvadrantového pulsního měniče.



Obr.č. 14 – grafický výstup z programu QtOctave (vlastní zpracování 2013)

Celkové hodnocení

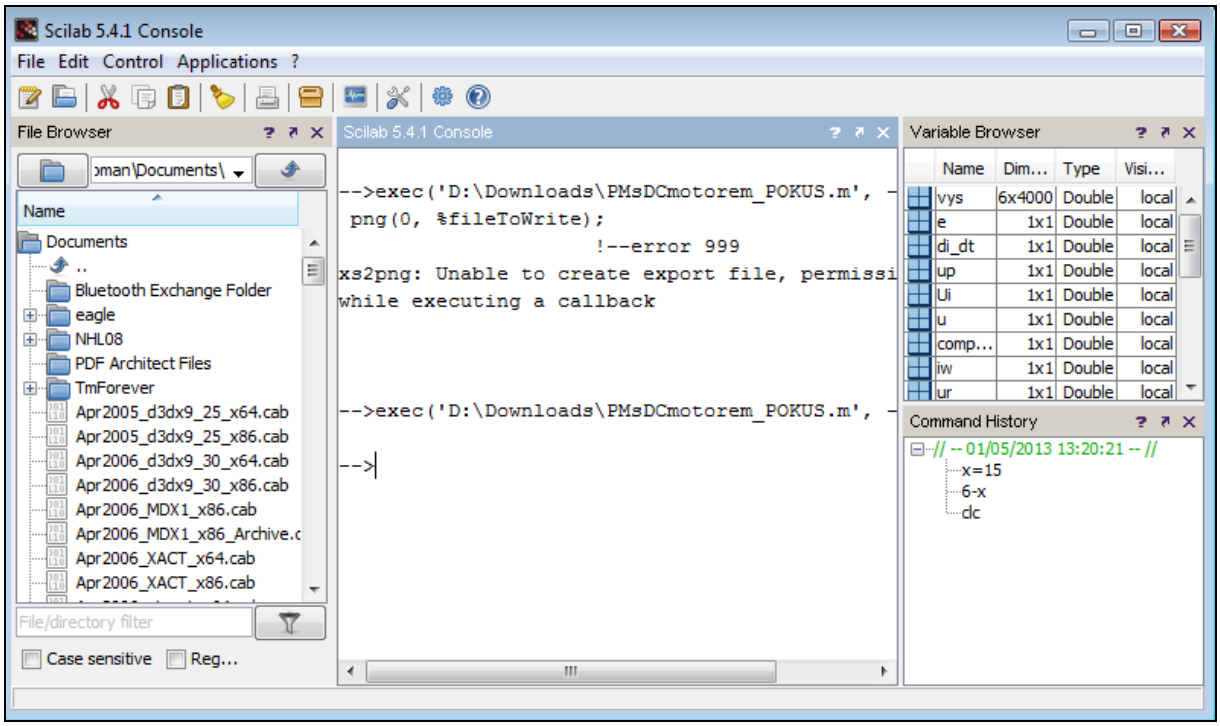
Program Octave podle mého názoru patří k nejlepším programovacím jazykům. Je vhodný také pro úplné začátečníky. Rychlá je výpočetní odezva testovaného zdrojového kódu a upravení kódu je velice snadné.

Výborný simulační program se širokým využitím vhodný pro všechny úlohy v elektrotechnickém odvětví.

2.3.4 Scilab

Verze SW: 5.3

Scilab je vědecký software vhodný pro numerické výpočty 2D i 3D grafů. Vyvíjen je francouzskými institucemi INRA a ENCP. Program je částečně kompatibilní s programem Matlab.



Obr.č. 15 – uživatelské prostředí Scilab (vlastní zpracování 2013)

Postup instalace

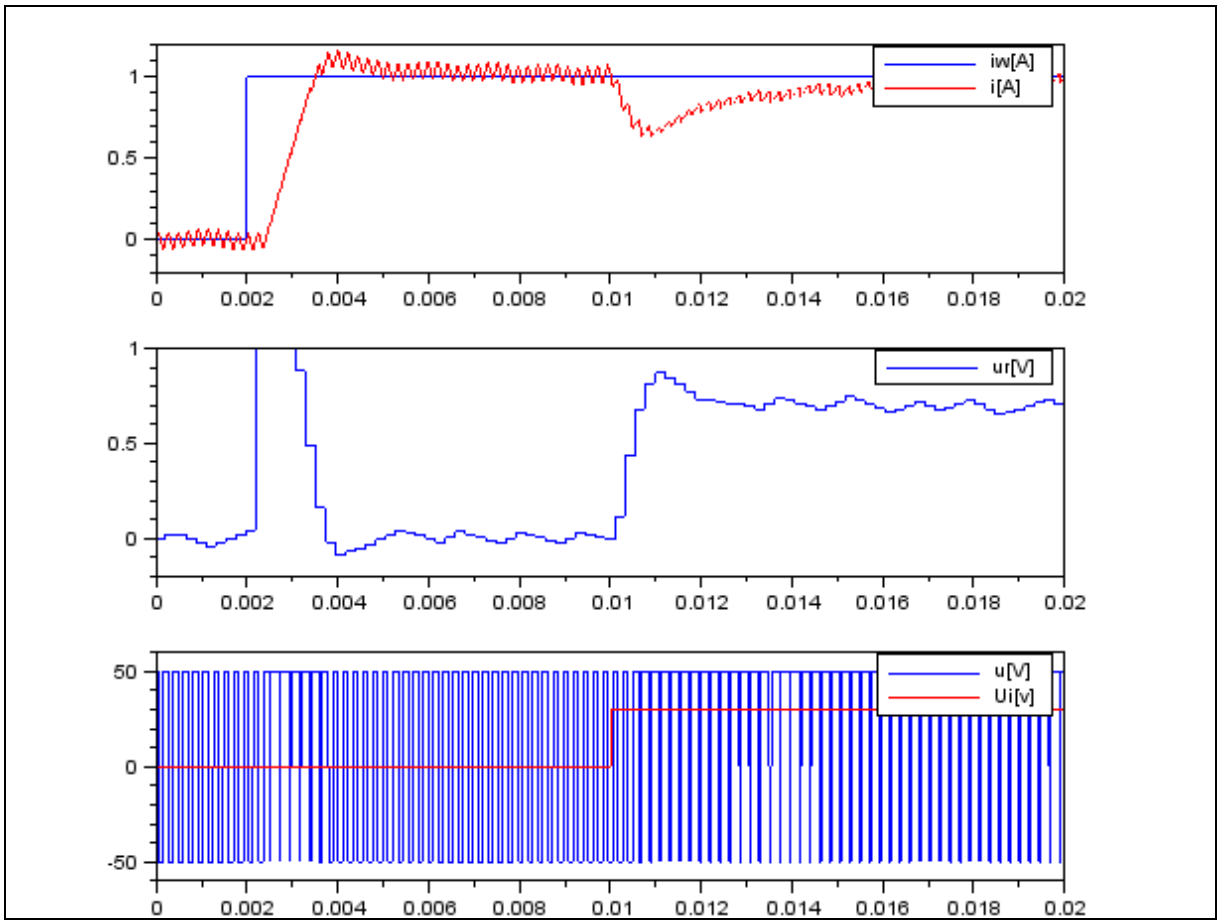
Software je možno nalézt v aplikaci Centrum software pro Ubuntu. Po přečtení požadavků a případných recenzí je možno program jednoduše nainstalovat příkazem o stejném názvu.

Uživatelský komfort

Uživatelské prostředí působí naprosto přehledně, rozdělení do jednotlivých sekcí je na dobré úrovni. Zajímavá je vysoká kompaktnost s programem Matlab. Upravení dostupného zdrojového kódu je jednoduchá záležitost. Vynikající je barevné označení příkazů od ostatního textu zdrojového kódu.

Demonstrace použitelnosti

Na obrázku číslo 16 je znázorněn výsledný grafický výstup simulace regulace proudu v kotvě DC motoru. Pouze u popisek os dochází k neshodě oproti původnímu kódu.



Obr.č. 16 – grafický výstup simulace z programu Scilab (vlastní zpracování 2013)

Celkové hodnocení

Vynikající program pro simulaci elektrotechnických úloh. Výpočetní odezva je poměrně rychlá. Editace ve zdrojovém kódu je velice jednoduchá.

Zhodnocení vlastností simulačních programů a nástrojů

Program		*ovladatelnost programu*	*použitelnost programu*	*přehlednost programu*	*rychlost výpočtu*	*celkové hodnocení*
Freepascal	ano	3	2	4	2	3
C/C++/KDevelop	ano	2	2	2	2	2
Lazarus	ano	3	2	2	2	2
Python/Spyder	ano	2	3	2	2	2
Freemat	ano	1	1	1	4	2
Maxima	ano	2	3	2	2	2
Octave	ano	1	1	2	1	1
Scilab	ano	1	1	2	1	1

Tab. č. 1 – zhodnocení programů (vlastní zpracování 2013)

Stupnice v tabulce číslo 1 odpovídá systému školnímu známkování

 - dostupnost softwaru pro operační systém MS Windows

Závěr

Pro simulaci složitých elektrotechnických úloh jsou podle mého názoru vhodné zejména následující programy Freemat, Octave nebo Scilab. Jejich atributy pro simulaci zadaného příkladu regulace proudu v kotvě DC motoru byly korektní a byla naplněna všechna očekávaná kritéria.

Základní simulační nástroje jsem neshledal vhodným řešením, protože import výsledků do jiného programu umožňující grafický výstup mi připadá složitým řešením. Pouze u příkladů, kde postačují výsledky bez nutnosti u každé simulace použít grafický výstup, lze uvažovat nad jejich využitím.

Seznam literatury a informačních zdrojů

[1] Svobodný software. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-04-14]. Dostupné z: https://cs.wikipedia.org/wiki/Svobodn%C3%BD_software

[2] Linux. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-04-14]. Dostupné z: http://cs.wikipedia.org/wiki/Linux#Historie

[3] BÍBR, Ivan a kol. Ubuntu 8.10 CZ: příručka uživatele Linuxu. Vyd. 1. Brno: Computer Press, 2008. 270 s. ISBN 978-80-251-2332-4

[4] Ubuntu: Ubuntu - systém který jste hledali. [online]. [cit. 2013-04-14]. Dostupné z: http://www.ubuntu.cz/cojeubuntu/predstaveni

[5] Mandriva Linux: Co je Mandriva Linux?. In: [online]. [cit. 2013-04-14]. Dostupné z: http://www.mandrivalinux.cz/o-mandriva-linuxu/

[6] BÍBR, Ivan. *Mandriva Linux 2007.1 CZ: instalační a uživatelská příručka*. Vyd. 1. Brno: Computer Press, 2007. 416 s. ISBN 978-80-251-1664-7

[7] SATRAPA, Pavel. *Pascal pro zelenáče*. Vyd. 3. Praha: Neocortex, 2005, 253 s. ISBN 80-863-3003-6

[8] HEROUT, Pavel. *Učebnice jazyka C*. 6. vyd. České Budějovice: Kopp, 2009, 271, viii s. ISBN 978-80-7232-383-8.

[9] Python. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-04-14]. Dostupné z: http://cs.wikipedia.org/wiki/Python

[10] LibreOffice. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-04-14]. Dostupné z: http://cs.wikipedia.org/wiki/LibreOffice

[11] LIPOVSKÝ, Pavol a Katarína DRAGANOVÁ. FreeMat: Základy práce s programom. Košice: TECHNICKÁ UNIVERZITA V KOŠICIACH, 2010. ISBN 978-80-553-0580-6.

[12] About GNU Octave. In: [online]. [cit. 2013-04-14]. Dostupné z: http://www.gnu.org/software/octave/about.html

Příloha 1 – zdrojový kód určený pro program OCTAVE

```
%regulace proudu v kotve DC motoru napajeneho ze 4-kvadrantoveho pulsniho
%menice (H-mustku)
clear;
dt=5e-6;%[s] %krok simulace
%parametry regulatoru
kr=2;
Taur=0.003;
fp=5000;%frekvence nosne pily, Hz
urmax=1;%rozsah ridiciho napeti mustku
U=50;%napajeci napeti mustku [V]
R=1;%odpor vinuti [Ohm]
L=0.05;%indukcnost vinuti [H]
dupila=4*fp;%derivace signalu nosne pily
%pocatecni podminky
t=0;i=0;sum=0;ur=0;iw=0;compareReg=0;
u=0;Ui=0;up=-1;
%vlastni vypocetni smycka
while t<0.02
   %definovani pozadovaneho prubehu proudu
   if t>0.002 iw=1; end;
   %definovani prubehu poruchy
   if t>0.01 Ui=30; end;
   %pila tvorici PWM (symetricka)
   up=up+dupila*dt;
   if up>urmax dupila=dupila*(-1);end;
   if up<-urmax
      dupila=dupila*(-1);
      compareReg=ur;%do compare registru PWM se priradi vystup regulatoru z minule
periody PWM - doba vypoctu algoritmu Pi regulatoru trva celou 1/fp (nejhorsi pripad)
      e=iw-i; %regulacni odchylka
      %PI (P) regulator
      if abs(ur)<urmax sum=sum+1/Taur*e/fp; end;%integrace reg. odchylky, pokud neni
```

```
vystup omezen
      ur=kr*(e+sum);
      %omezovac regulatoru
      if ur>urmax ur=urmax; end;
      if ur<-urmax ur=-urmax; end;
   end;
   % PWM
   if compareReg>up u=U; else u=-U; end;
   % vypocet diference proudu (z rovnice u=R*i+L*di/dt)
   di_dt=1/L*(u-R*i-Ui);
   % numericka integrace Eulerovou metodou
   i=i+di_dt*dt;
   % zapis vysledku do pole
   if t==0 vys=[t,iw,ur,u,Ui,i]';
   else vys=[vys [t,iw,ur,u,Ui,i]']; end;
   % posun casu na dalsi vypocetni krok
   t=t+dt;
end;

subplot(3,1,1);
plot(vys(1,:),vys(2,:));hold on;
plot(vys(1,:),vys(6,:),'r');hold off;
legend('iw[A]','i[A]');
subplot(3,1,2);
plot(vys(1,:),vys(3,:));
legend('ur[V]');
subplot(3,1,3);
plot(vys(1,:),vys(4,:));hold on;
plot(vys(1,:),vys(5,:),'r');hold off;
legend('u[V]','Ui[v]');
set(gca,'YLim',[-U*1.1,U*1.1]);
xlabel('time [s]');
```

Příloha 2 – zdrojový kód upravený pro program SCILAB

```
dt=5e-6;
kr=2;
Taur=0.003;
fp=5000;
urmax=1;
U=50;
R=1;
L=0.05;
dupila=4*fp;
t=0;i=0;suma=0;ur=0;iw=0;compareReg=0;
u=0;Ui=0;up=-1;
while t<0.02
   if t>0.002 iw=1; end;
   if t>0.01 Ui=30; end;
   up=up+dupila*dt;
   if up>urmax dupila=dupila*(-1);end;
   if up<-urmax
      dupila=dupila*(-1);
      compareReg=ur;
      e=iw-i;
      if abs(ur)<urmax suma=suma+1/Taur*e/fp; end;
      ur=kr*(e+suma);
      if ur>urmax ur=urmax; end;
      if ur<-urmax ur=-urmax; end;
    end;
   if compareReg>up u=U; else u=-U; end;
   di_dt=1/L*(u-R*i-Ui);
   i=i+di_dt*dt;
   if t==0 vys=[t,iw,ur,u,Ui,i]';
   else vys=[vys [t,iw,ur,u,Ui,i]']; end;
   t=t+dt;
end;
```

```
subplot(3,1,1);
plot(vys(1,:),vys(2,:));
plot(vys(1,:),vys(6,:),'r');
legend('iw[A]','i[A]');
subplot(3,1,2);
plot(vys(1,:),vys(3,:));
legend('ur[V]');
subplot(3,1,3);
plot(vys(1,:),vys(4,:));
plot(vys(1,:),vys(5,:),'r');
legend('u[V]','Ui[v]');
gca,'YLim',[-U*1.1,U*1.1];
xlabel('time [s]');
```

Příloha 3 – zdrojový kód upravený pro program Freemat

```
clear;
dt=5e-6;
kr=2;
Taur=0.003;
fp=5000;
urmax=1
U=50;
R=1;
L=0.05;
dupila=4*fp;
t=0;i=0;sum=0;ur=0;iw=0;compareReg=0;
u=0;Ui=0;up=-1;
while t<0.02
   if t>0.002 iw=1; end;
   if t>0.01 Ui=30; end;
   up=up+dupila*dt;
   if up>urmax dupila=dupila*(-1);end;
```

```
   if up<-urmax
      dupila=dupila*(-1);
      compareReg=ur;
      e=iw-i;
      if abs(ur)<urmax sum=sum+1/Taur*e/fp; end;
      ur=kr*(e+sum);
      if ur>urmax ur=urmax; end;
      if ur<-urmax ur=-urmax; end;
    end;
   if compareReg>up u=U; else u=-U; end;
      di_dt=1/L*(u-R*i-Ui);
      i=i+di_dt*dt;
   if t==0 vys=[t,iw,ur,u,Ui,i]';
   else vys=[vys [t,iw,ur,u,Ui,i]']; end;
   t=t+dt;
end;

subplot(3,1,1);
plot(vys(1,:),vys(2,:));hold on;
plot(vys(1,:),vys(6,:),'r');hold off;
legend('iw[A]','i[A]');
subplot(3,1,2);
plot(vys(1,:),vys(3,:));
legend('ur[V]');
subplot(3,1,3);
plot(vys(1,:),vys(4,:));hold on;
plot(vys(1,:),vys(5,:),'r');hold off;
legend('u[V]','Ui[v]');
set(gca,'YLim',[-U*1.1,U*1.1]);
xlabel('time [s]');
```

Příloha 4 – zdrojový kód upravený pro program Maxima

```
dt:5e-6;
kr:2;
Taur:0.003;
fp:5000;
urmax:1;
U:50;
R:1;
L:0.05;
dupila:4*fp;
t:0;i:0;
sum:0;ur:0;iw:0;compareReg:0;
u:0;Ui:0;up:-1;
while t<0.02 then
if t>0.002 then iw:1;
if t>0.01 then Ui:30;
up:up+dupila*dt;
if up>urmax then dupila:dupila*(-1);
if up<-urmax then dupila:dupila*(-1);
compareReg:ur;
e:iw-i;
if abs(ur)<urmax then sum:sum+1/Taur*e/fp;
ur:kr*(e+sum);
if ur>urmax then ur:urmax;
if ur<-urmax then ur:-urmax;
if compareReg>up then u:U; else u:-U;
di_dt:1/L*(u-R*i-Ui);
i:i+di_dt*dt;
if t=0 then vys:[t,iw,ur,u,Ui,i]';
else then vys:[vys [t,iw,ur,u,Ui,i]'];
t:t+dt;

subplot2d(3,1,1);
plot2d(vys(1,:),vys(2,:));hold on;
```

```
plot2d(vys(1,:),vys(6,:),'r');hold off;
legend('iw[A]','i[A]');
subplot2d(3,1,2);
plot2d(vys(1,:),vys(3,:));
legend('ur[V]');
subplot2d(3,1,3);
plot2d(vys(1,:),
vys(4,:));hold on;
plot2d(vys(1,:),vys(5,:),'r');hold off;
legend('u[V]','Ui[v]');
set(gca,'YLim',[-U*1.1,U*1.1]);
xlabel('time [s]');
```

Příloha 5 – zdrojový kód upravený pro program Lazarus

```
program Regulace;
{$mode objfpc}{$H+}
uses
  {$IFDEF UNIX}{$IFDEF UseCThreads}
  cthreads,
  {$ENDIF}{$ENDIF}
  Classes, SysUtils, CustApp
  { you can add units after this };
type
  { Regulace_proudu }
  Regulace_proudu = class(TCustomApplication)
  protected
    procedure DoRun; override;
  public
    constructor Create(TheOwner: TComponent); override;
    destructor Destroy; override;
    procedure WriteHelp; virtual;
  end;
{ Regulace_proudu }
procedure Regulace_proudu.DoRun;
```

```
const pocet=100;
var
  ErrorMsg: String;   dt:real;  vys : array [1..pocet,1..6] of Real;       {2D pole realnych cisel,
(1xPocet,1x3)}
     n : Integer;
     Taur,L,dupila,i,t,sum,ur,iw,compareReg,u,Ui,up,e,didt:real;
     kr,fp,urmax,UU,R:integer;
     c:char;
begin
  // quick check parameters
  ErrorMsg:=CheckOptions('h','help');
  if ErrorMsg<>'' then begin
    ShowException(Exception.Create(ErrorMsg));
    Terminate;
    Exit;
  end;
  // parse parameters
  if HasOption('h','help') then begin
    WriteHelp;
    Terminate;
    Exit;
  end;
  { add your program here }
    dt:=5E-6;
kr:=2;
Taur:=0.003;
fp:=5000;
urmax:=1;
U:=50;
R:=1;
L:=0.05;
dupila:=4*fp;
t:=0;i:=0;sum:=0;ur:=0;iw:=0;compareReg:=0;
u:=0;Ui:=0;up:=-1;
```

```
while t<0.02   do
   if t>0.002 then iw:=1;
   if t>0.01 then Ui:=30;
   up:=up+dupila*dt;
   if up>urmax then dupila:=dupila*(-1);
   if up<-urmax then dupila:=dupila*(-1);
      compareReg:=ur;
      e:=iw-i;
      if abs(ur)<urmax then sum:=sum+1/Taur*e/fp;
      ur:=kr*(e+sum);
      if ur>urmax then ur:=urmax;
      if ur<-urmax then ur:=-urmax;
   if compareReg>up then u:=U else u:=-U;
   didt:=1/L*(u-R*i-Ui);
   i:=i+didt*dt;
   if t=0 then vys:=[t,iw,ur,u,Ui,i]
   else then  vys:=[vys [t,iw,ur,u,Ui,i]];
   t:=t+dt;
subplot(3,1,1);
plot(vys(1,:),vys(2,:));hold on;
plot(vys(1,:),vys(6,:),'r');hold off;
legend('iw[A]','i[A]');
subplot(3,1,2);
plot(vys(1,:),vys(3,:));
legend('ur[V]');
subplot(3,1,3);
plot(vys(1,:),vys(4,:));hold on;
plot(vys(1,:),vys(5,:),'r');hold off;
legend('u[V]','Ui[v]');
set(gca,'YLim',[-U*1.1,U*1.1]);
xlabel('time [s]');
 // stop program loop
 Terminate;
end;
```

```
constructor Regulace_proudu.Create(TheOwner: TComponent);
begin
  inherited Create(TheOwner);
  StopOnException:=True;
end;
destructor Regulace_proudu.Destroy;
begin
  inherited Destroy;
end;
procedure Regulace_proudu.WriteHelp;
begin
  { add your help code here }
  writeln('Usage: ',ExeName,' -h');
end;
var
  Application: Regulace_proudu;
begin
  Application:=Regulace_proudu.Create(nil);
  Application.Title:='regulace';
  Application.Run;
  Application.Free;
end.
```

Příloha 6 – zdrojový kód upravený pro program Freepascal

```
Program Regulace;
Uses Crt;
Const  pocet=4000;
Var   P : array [1..pocet,1..6] of Real;
      n : Integer;
      dt,Taur,L,dupila,i,t,sum,ur,iw,compareReg,u,Ui,up,e,di_dt:real;
      kr,fp,urmax,UU,R:integer;
      c:char;
Begin
```

```
dt:=5E-6;
kr:=2;
Taur:=0.003;
fp:=5000;
urmax:=1;
UU:=50;
R:=1;
L:=0.05;
dupila:=4*fp;
t:=0;i:=0;sum:=0;ur:=0;iw:=0;compareReg:=0;u:=0;Ui:=0;up:=-1;
if t<0.002 then
if t>0.002 then iw:=1;
if t<0.01 then Ui:=30;
up:=up+dupila*dt;
if up>urmax then dupila:=dupila*(-1);
if up<-urmax then dupila:=dupila*(-1);
compareReg:=ur;
e:=iw-i;
if (abs(ur))<urmax then sum:=sum+1/Taur*e/fp;
ur:=kr*(e+sum);
if ur>urmax then ur:=urmax;
if ur<-urmax then ur:=-urmax;
if compareReg>up then u:=UU else u:=-UU;
di_dt:=1/L*(u-R*i-Ui);
i:=i+di_dt*dt;
t:=t+dt;
  For n:=1 to pocet do  {zaplnime pole vypocitanymi hodnotami}
   begin
    P[n,1]:=t;  P[n,2]:=iw;  P[n,3]:=ur;  P[n,4]:=u;  P[n,5]:=Ui;  P[n,6]:=i;
   end;
  For n:=1 to pocet do
   begin
    WriteLn (P[n,1]:3:3,'       ',P[n,2]:3:3,'       ',P[n,3]:3:3,'       ',P[n,4]:3:3,'
',P[n,5]:3:3,'       ',P[n,6]:3:3);
```

```
  end;
 if c='k' then halt(0);
end.
```

Příloha 7 – zdrojový kód upravený pro program KDevelop

```
# include <stdio.h>
# include <stdlib.h>
float dt = 5e-6;
int kr = 2;
float Taur = 0.002;
float fp = 5000.0;
float urmax = 1.0;
float U = 50.0;
float R = 1.0;
float L = 0.05;
float dupila = 4 * fp;
float t = 0.0, i = 0.0, sum = 0.0, ur = 0.0, iw = 0.0, compareReg = 0.0;
float u = 0.0, Ui = 0.0, up = -1.0;
while (t < 0.02)
{
if (t > 0.002)
iw = 1.0;
if (t > 0.01)
Ui = 30.0;
up = up + dupila * dt;
if (up > urmax)
dupila = -1 * dupila;
if (up < -urmax)
{
dupila = -1 * dupila;
compareReg = ur;
float e = iw - i;
if (abs(ur) < urmax)
sum = sum + 1 / Taur * e / fp;
```

```
ur = kr * (e + sum);
if (ur > urmax)
ur = urmax;
if (ur < -urmax)
ur=-urmax;
}
if (compareReg > up)
u = U;
else
u = -U;
float di_dt =( 1 / L * (u - R * i - Ui));
i = i + di_dt * dt;
Vysledek vys;
vys.vys[0] = t;
vys.vys[1] = iw;
vys.vys[2] = ur;
vys.vys[3] = u;
vys.vys[4] = Ui;
vys.vys[5] = i;
vyslpole.append(vys);
t = t + dt;
```

Příloha 8 – zdrojový kód upravený pro program Python/Spyder

```
dt=5e-6;
kr=2;
Taur=0.003;
fp=5000;
urmax=1;
U=50;
R=1;
L=0.05;
iw=1;
dupila=4*fp;
```

```
t=0;
i=0;
suma=0;
ur=0;iw=0;compareReg=0;
u=0;Ui=0;up=-1;
while t<0.02:
if t>0.002: iw=1;
if t>0.01: Ui=30;
up=up+dupila*dt;
if up>urmax: dupila=dupila*(-1);
if up<-urmax:
    dupila=dupila*(-1);
compareReg=ur;
e=iw-i;
if abs(ur)<urmax: suma=suma+1/Taur*e/fp;
ur=kr*(e+suma);
if ur>urmax: ur=urmax;
if ur<-urmax: ur=-urmax;
if compareReg>up: u=U;
else: U=-u;
di_dt=1/L*(u-R*i-Ui);
i=i+di_dt*dt;
if t==0: vys=[t,iw,ur,u,Ui,i]
else: vys=[vys[t,iw,ur,u,Ui,i]]
t=t+dt;
```