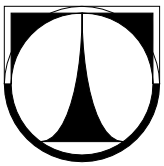
TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky, informatiky a mezioborových studií



BAKALÁŘSKÁ PRÁCE

Liberec 2012 **Jan Kovář**

TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky, informatiky a mezioborových studií

Studijní program: B2612 – Elektrotechnika a informatika

Studijní obor: B2612 – Elektrotechnika a informatika

Virtuální úloha výtahu řízená pomocí měřící karty

Virtual model of lift controlled via measuring card

Bakalářská práce

Autor:	Jan Kovář
Vedoucí práce:	Ing. Petr Školník Ph.D.
Konzultant:	Ing. Lukáš Hubka, Ph.D.

V Liberci 18. 5. 2012

Zadáni list

Prohlášení

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím bakalářské práce a konzultantem.

Datum

Podpis

Poděkování

Děkuji vedoucímu bakalářské práce Ing. Petru Školníkovi, Ph.D. za cenné rady, připomínky a metodické vedení práce.

Abstrakt

Cílem bakalářské práce je vytvořit virtuální model výtahu ovládaný pomocí měřící karty.

Fyzikální vlastnosti virtuálního modelu by měli co nejvěrněji odpovídat reálnému modelu. Bylo potřeba použít vhodný řešící algoritmus pro dané diferenciální rovnice. Pro přesné řízení tohoto modelu v čase bylo potřeba najít co nejpřesnější způsob stanovení reálného času na PC.

Dále byl kladen důraz na vizualizaci modelu i průběžného ukládání aktuálních hodnot výstupních veličin.

Dalším problémem bylo zavedení knihoven Advantech, umožňující komunikaci mezi programem a vstupně výstupní kartou.

Klíčová slova

Měřící karta, Advantech, diferenciální rovnice, virtuální model, výtah, reálný čas

Abstract

The aim of this bachelor work is to create virtual model of lift controlled via measuring card.

Physical properties of the virtual model should correspond closely as possible to the real model. It was necessary to use a suitable algorithm for solving the system of differential equations. For precise control of the model in the time it was needed to find the most accurate method of determining real-time on the PC.

Furthermore, the emphasis was on model visualization and storage of continuous output of current values.

Another problem was the introduction of Advantech libraries, enabling communication between the program and input and output card.

Keywords

Data acquisition card, Advantech, differential equation, virtual model, real time clock

Obsah

Seznam obrázků

Seznam použitých zkratek

UI	User Interface – uživatelské rozhranní
ODE	Ordinary differential equation – obyčejná diferenciální rovnice
AI	Analog input – analogový vstup
AO	Analog output – analogový výstup
AI	Analog input – analogový vstup
DI	Digital output – digitální výstup
CSV	Comma-separated values, hodnoty oddělené čárkami

Úvod

Modelování virtuálních systémů je disciplína, umožňující levnější a rychlejší vývoj reálných systémů.

S virtuálními modely se setkáváme dnes a denně. Počítačové hry, výukové simulátory, vývojové simulaci či jen modely umožňující zobrazit kopii reality širšímu okruhu lidí. Základní myšlenka mé práce spočívá ve vytvoření virtuálního modelu výtahu, který bude možné ovládat pomocí vstupně výstupní karty, či vestavěného demo režimu. Takto bude moci sloužit jako výuková pomůcka, ke které bude možné připojit například studenty naprogramované PLC.

Moje aplikace představuje model běžného výtahu vybaveného lineárním elektromotorem a protizávažím, jaký je možný najít v bytových zástavbách.
Model výtahu se řídí pomocí akční veličiny, představující napětí dodávané elektromotoru. To je také jediný vstup, který měřící karta přijímá.

Sestavení aplikace virtuálního modelu předchází řada problémů. Protože se jedná o simulaci, kde je rozhodující práce ve skutečném čase, je třeba najít co nejpřesnější nástroj pro jeho určení.

Dalším problém je sestavení diferenciálních rovnic daného modelu v explicitním tvaru a jejich řešení v reálném čase s danou akční veličinou.

1 Teoretická část

1.1 Specifikace zadání

Sestavit matematicko - fyzikální rovnice popisující systém výtahu nebylo zájmem mé bakalářské práce. Tyto materiály mi byly dodány vedoucím práce jako výchozí.

Jedná se o soustavu diferenciálních rovnic (1-9) a schéma v prostředí Matlab Simulink na obrázku 2, které jednoznačně určilo, jak má model fungovat. Rovnice vycházejí z práce uvedené [1]. Jako vstupní (akční) veličina je použito napětí motoru výtahu. Výstupní veličinou je délka lana L_2 mezi kabinou a nejvyšším styčným bodem kola spojeného s hřídelí motoru. Pomocí této hodnoty můžeme dále určit vzdálenosti jednotlivých pater, nepracujeme tedy se snímači polohy.

$$m_0 \frac{dv_0}{dt} = T_1 - m_0 g - cv_0 \tag{1}$$

$$m_2 \frac{dv_2}{dt} = m_2 g - T_2 - cv_2 \tag{2}$$

$$\frac{J_1}{R_1^2} \frac{dv_1}{dt} = \frac{\tau_m}{R_1} - T_1 + T_2 \tag{3}$$

$$L_1 \frac{dT_1}{dt} = EA\left(\frac{dL_1}{dt} + v_1\right) - v_1 T_1 \tag{4}$$

$$L_2 \frac{dT_2}{dt} = EA\left(\frac{dL_2}{dt} - v_1\right) - v_2 T_2 + v_1 T_1 \tag{5}$$

$$M_m = \tau_m = k_m i_m \tag{6}$$

$$U_m = R_m i_m + L_m \frac{di_m}{dt} + k_m \frac{d\varphi_m}{dt} \tag{7}$$

$$J_m \frac{d^2\varphi_m}{dt^2} + B_m \frac{d\varphi_m}{dt} = M_M - M_Z \tag{8}$$

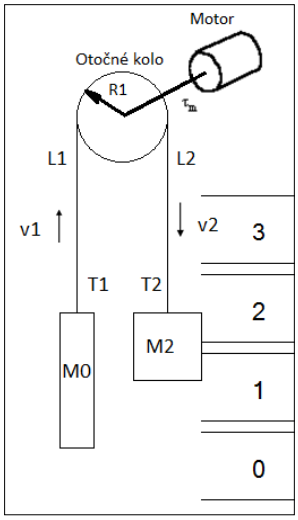
$$M_Z = R_1 (T_2 - T_1) \tag{9}$$

Význam jednotlivých veličin je patrný z tabulky 1. Výchozí hodnota se týká modelu na obrázku 2 i vytvořené aplikace.

Tabulka 1 – přehled veličin

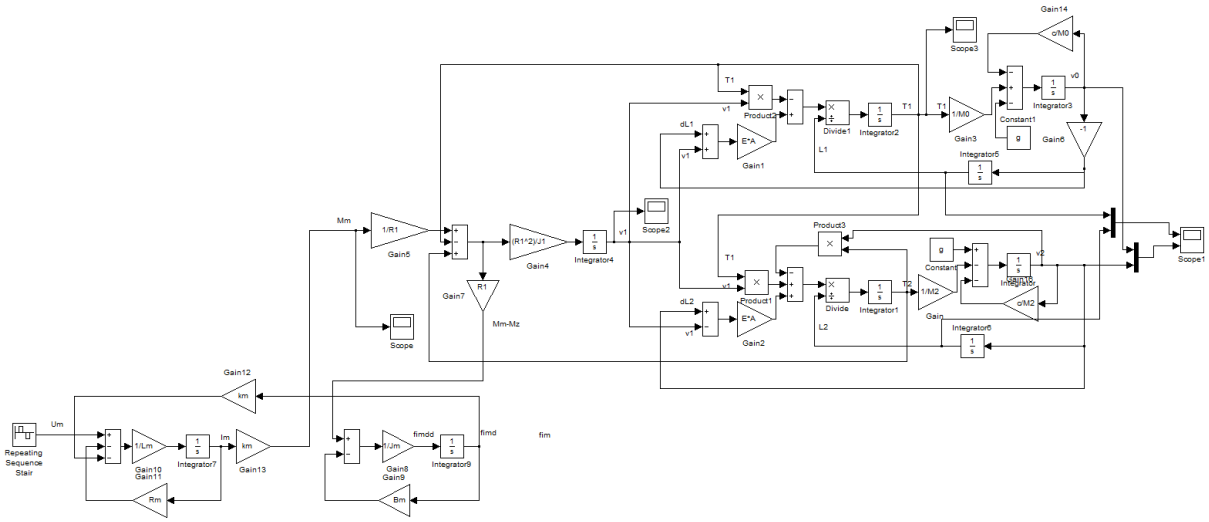
Veličina	Jednotka	Výchozí hodnota v modelu	Určení
m_0	[kg]	650	hmotnost závaží
m_2	[kg]	650	hmotnost kabiny
v_1	[m.s^{-1}]	0	rychlost pohybu závaží
v_2	[m.s^{-1}]	0	rychlost pohybu kabiny
J_1	[Nm]	2.5	moment setrvačnosti otočného kola
R_1	[m]	0.5	poloměr kola
T_1	[N]	0	Tažná síla působící k závaží
T_2	[N]	0	Tažná síla působící k motoru závaží
L_1	[m]	10	délka lana na straně závaží
L_2	[m]	10	délka lana na straně kabiny
E	[Pa]	200*10^9	Modul pružnosti lana v tahu
A	[m^2]	π *(0.03/2)^2	Plocha průřezu lana
M_m	[Nm]	1000	Moment síly motoru
U_m	[V]	200	Napětí motoru
i_m	[A]	0	Proud motoru
J_m	[Nm]	2.5	Moment setrvačnosti motoru
B_m	[T]	4.6*10^{-3}	magnetická indukce motoru
Rm	[Ω]	6	odpor motoru
M_Z	[Nm]	0	moment závaží
φ_M	[WB]	0	magnetický indukční tok smyčkou motoru
k_m		7.2	konstanta motoru
τm	[Nm]	0	moment síly hřídele mezi motorem a otočným kolem
Lm	[H]	0.016	vlastní indukčnost motoru
c	[N.s.m^{-1}]	1000	Tlumení uzpůsobené uložením kabiny

Model sestavy odpovídá obr. 1. V modelu se zanedbává hmotnost lana.



Obrázek 1 – model

Spolu s těmito rovnicemi mi bylo dodáno schéma modelu v prostředí Matlab Simulink. Toto schéma vytvořené pomocí základních bloků představuje funkční model výtahu.



Obrázek 2 - model výtahu sestaven v prostředí Matlab Simulink.

1.2 Přehled vývojových prostředí

1.2.1 Vývojová prostředí dynamických systémů

Matlab – Simulink

Populární programové prostředí vyvíjené firmou MathWorks má uplatnění pro nepřeberné množství aplikací. První verze vydána již roku 1984. [2] Programové prostředí se dodává pro operační systémy Linux, Mac OS X a MS Windows. Pomocí komponenty Simulink, lze simulovat dynamické systémy i bez znalosti programovacího jazyka. Jako nevýhoda Simulinku může být v některých případech jeho signálová orientace.

OpenModelica

Výsledek vývoje nevýdělečné organizace OSMC (Open Source modelica Consorcium). Vývoj software začal jako diplomová práce v roce 1997. První externí verze byla k dispozici v roce 2004 [3] Vychází z objektově orientovaného jazyka Modelica, který je využíván v mnoha komerčních projektech. Obsahuje řadu nástrojů pro vývoj dynamických systémů. Stejně jako předešlá vývojová prostředí je dostupná pro Windows, Linux i Mac. Oproti komerčním není tak propracovaná a uživatelská podpora se spoléhá na open source komunitu. Na druhou stranu s jejím použitím odpadají náklady na pořízení aplikace.

Scisos[4]

Nástroj pro grafické modelování a simulaci dynamických systémů. Stejně jako předchozí aplikace je k dispozici v rámci open source zdarma. Umožňuje kombinaci použití spojitého a diskrétního času v jednom modelu. Lze generovat kód modelu v jazyce C. Velkou výhodou je také možnost simulace v reálném čase s reálnými zařízeními pomocí komponenty Scicos-Hil.

1.2.2 Obecná vývojová prostředí

Microsoft Visual Studio

Integrované vývojové prostředí od firmy Microsoft. Zahrnuje celou řadu nástrojů umožňující vývoj vývoj konzolových i aplikací s grafickým uživatelským rozhraním. Každá verze Visual studia vycházela zároveň s nejnovějším frameworkem Microsoft .NET. Tento

framework je dnes notně zastoupen v mnoha desktopových, internetových i aplikacích na mobilních telefonech. Mezi nejpoužívanější programovací jazyky, které jsou součástí vývojového prostředí Microsoft Visual Studio využívající Microsoft .NET framework patří C# a Visual Basic .NET. Tyto součásti jsou ve verzi EXPRESS zdarma volně dostupné ke stažení i pro komerční účely.

Delphi [5]

Vizuální vývojové prostředí původem od firmy Borland vycházející z jazyka object pascal. První verze byla uvedena na trh roku 1995. Dříve toto prostředí spolu s Borland C++ velice oblíbené. Používá knihoven VCL (Visual Component Library) a CLS (Component Library for Cross Platform). S nástupem Microsoft Visual Studia popularita postupně upadala. Dodnes je Delphi hojně využívané ve školních projektech, ve firemní sféře se používá spíše na rozšiřování starších aplikací.

NetBeans (Java)

Vývojové prostředí firmy Oracle. Umožňuje vytváření desktopových aplikací pomocí grafického rozhranní AWT nebo Swing. Programovací jazyk Java je velice populární díky snadné přenositelnosti mezi různými operačními systémy na různých zařízeních.

1.3 Metody řešení obyčejných diferenciálních rovnic

Diferenciální rovnice (ODE) je taková rovnice $\frac{dx}{dt} = f(t, x(t))$, v níž se vyskytuje neznámá funkce a její derivace. Je proto třeba znát počáteční podmínky $x(0)=x_0$.

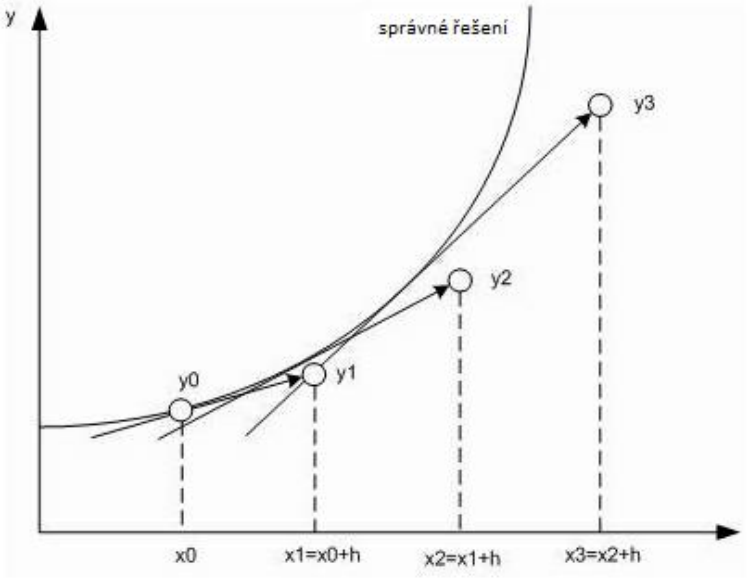
1.3.1 Základní metody pro numerické řešení obyčejných diferenciálních rovnic

Eulerova metoda

Představuje nejjednodušší metodu pro numerické řešení diferenciálních rovnic. Jejím autorem je Leonhard Euler, a poprvé byla zveřejněna roku 1768[6]. Jednoduchost této jednokrokové metody je vykoupena velice malou přesností.

$$y_{n+1} = y_n + hf(t_n, y_n)$$

Vzdálenost mezi dvěma body je definována jako *h*.



Obrázek 3 – znázornění Eulerovy metody[7]

Metoda Runge-Kutta 4. Řádu[8]

Tato metoda byla objevena německými matematiky C.Runge a M.W. Kutta okolo roku 1900. Metoda Runge-Kutta vychází z Eulerovy metody, jak je vidět z prvního členu (1).

$$k_1 = f(t_n, y_n) \tag{1}$$

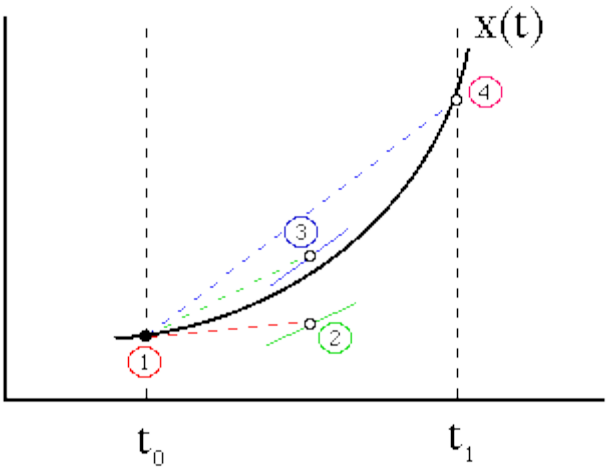
$$k_2 = f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_1\right) \tag{2}$$

$$k_3 = f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_2\right) \tag{3}$$

$$k_4 = f(t_n + h, y_n + hk_3) \tag{4}$$

$$y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{5}$$

Hodnoty *k* reprezentují derivace stavů systému ve speciálních bodech v průběhu intervalu.



Obrázek 4 - znázornění metody RK[9]

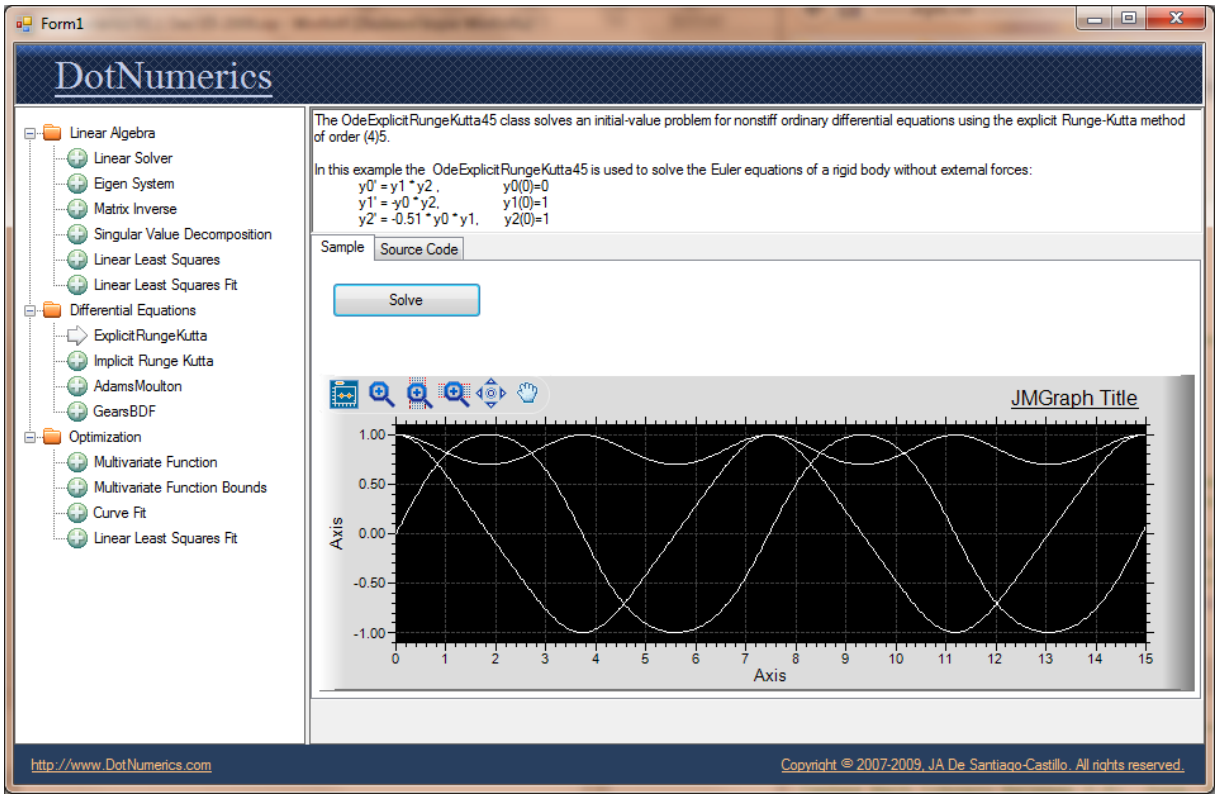
1.3.2 Dynamické knihovny obsahující ODE solver

Alglib[10]

Matematická knihovna volně k dispozici pro nekomerční použití. Zahrnuje dynamické knihovny pro C++, C#, VB.NET, FreePascal, CPython a IronPython. Využívá vysoce přesnou Runge-Kutta-Casp-Karpovu metodu s adaptivním krokem pro numerické řešení soustavy diferenciálních rovnic. RK s adaptivním krokem nejprve vypočítá stav pomocí jednoho a posléze více kroků, pakliže jsou výsledky rozdílnější než umožňuje zadaná tolerance, krok se zjemní. Toto řešení pro náš případ práce s reálným časem není vhodné, protože výpočet nemusí být proveden v daném intervalu časovače.

DotNumerics[11]

Knihovna pro numerické výpočty v prostředí .NET. Obsahuje nástroje pro řešení problémů lineární algebry, numerického řešení diferenciálních rovnic a optimalizace funkcí s více proměnnými. Tato dynamická knihovna obsahuje ODE solver RungeKutta45 s možností nastavit velikost kroku. Díky tomu, lze snadno určit, jak bude výpočet dlouho trvat. Výhodou je také demo režim, demonstrující využití jednotlivých nástrojů a jejich implementaci pomocí zdrojového kódu.



Obrázek 5 – Demo režim knihony DotNumeric

1.4 Problematika reálného času

Pracovat s reálným časem je nezbytné zejména v těchto případech[12]

- Řídící systémy
- Audio / video
- Počítačové hry – virtuální realita
- Simulace
- Přenos dat

1.4.1 Rozdělení reálného času

Hard real time

- Používá se u aplikací, kde je dodržení reálného času stěžejní. Při nedodržení mohou vzniknout velké škody.

- Nemusí se jednat jen o rychlé úlohy. Důležitá je přesná pravidelnost.

Soft real time

- Případné nedodržení časové hranice neznamená nic horšího než snížení kvality služby
- Vhodné pro velice krátké intervaly, kde se případná chyba dá aproximovat.

1.4.2 Určení reálného času v prostředí Microsoft Visual C# [13]

Prostředí Visual Studio obsahuje 3 timery.

System.Windows.Forms.Timer

Základní timer pracující v prostředí Windows Forms. Umožňuje nastavit periodu již od 1 ms. Pracuje pouze v rámci vlákna uživatelského rozhranní programu. Přesnost není vysoká (okolo 65 ms). Vhodné jen pro jednoduché programy a dlouhé časové periody.

System.Timers.Timer

Pokročilý timer optimalizovaný pro více-vláknové použití. Může být synchronizován s uživatelským rozhranním.

System.Threading.Timer

Tento timer umí pracovat jen ve svém pracovním vlákně. Nelze použít na UI.

Multimedia Timer [14]

Speciální timer poskytující mimořádnou přesnost 1 ms. Ideální pro použití v aplikacích vyžadující přesný chod. Vysoká náročnost na výpočetní výkon. Není k dispozici v rámci knihoven .Net.

1.5 Měřící karta Advantech PCI 1733[15]

Tato karta představuje základní model v nabídce firmy Advantech.

Obrázek 6 – měřící karta Advantech PCI 1711

Karta poskytuje 2 analogové a 16 digitálních vstupů. Ve stejném počtu jsou zastoupeny i výstupy. Na kartě je přítomný FIFO zásobník pro výstupní na 1024 vzorků.

digitální vstupy

- Kanálů: 16
- Vstupní napětí: Logická 0: 0.8 V max., Logická 1: 2 V min.
- Kompatibilita s 5 V/TTL

Digitální výstupy

- Kanálů: 16
- Výstupní napětí: Logická 0: 0.8 V, Logická 1: 2 V min.
- Kompatibilita s 5 V/TTL

Analogové vstupy

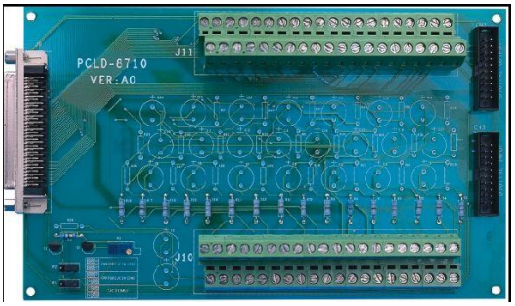
- Kanálů: 2
- Rozlišení: 12 bit
- Maximální vzorkovací frekvence 100 kS/s

- Napěťový rozsah vstupního napětí: volitelný pro každý kanál zvlášť ±10 V, ±5 V, ±2.5 V, ±1.25 V, ± 0.625 V
- Kompatibilita s 5 V/TTL
- FIFO zásobník na 1024 vzorků
- Vstupní impedance 2 MW/5 pF
- Přepěťová ochrana do 30 V

Analogové výstupy

- Kanálů: 2
- Rozlišení: 12 bit
- Výstupní napětí: 0-10 V
- Kompatibilita s 5 V/TTL

Ke kartě je k dispozici terminál Advantech PCDL-8710[16], umožňující snadné propojení jednotlivých kanálů karty Advantech PCI 1711 a připojeného zařízení. Zaujme možností připojit filtry pro filtrování analogových signálů. Pro pevné umístění je možné terminál připojit zadní stranou k nosné DIN liště.

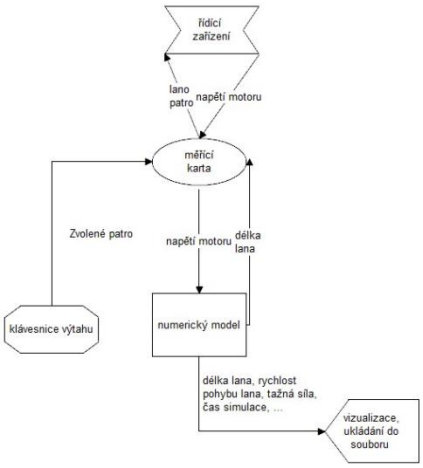


Obrázek 7 – Terminál Advantech PCL-8710

2 Tvorba aplikace

Aplikace byla vytvořena ve vývojovém studiu Visual Studio 2010, pomocí jazyka C#. Bylo tak zvoleno na základě zkušeností s prací v tomto prostředí a dostupnosti verze Ultimate pod akademickou licencí MSDNAA.

Pro správné pochopení pospolitosti jednotlivých částí jsem sestavil diagram činností probíhající v rámci časového kroku.

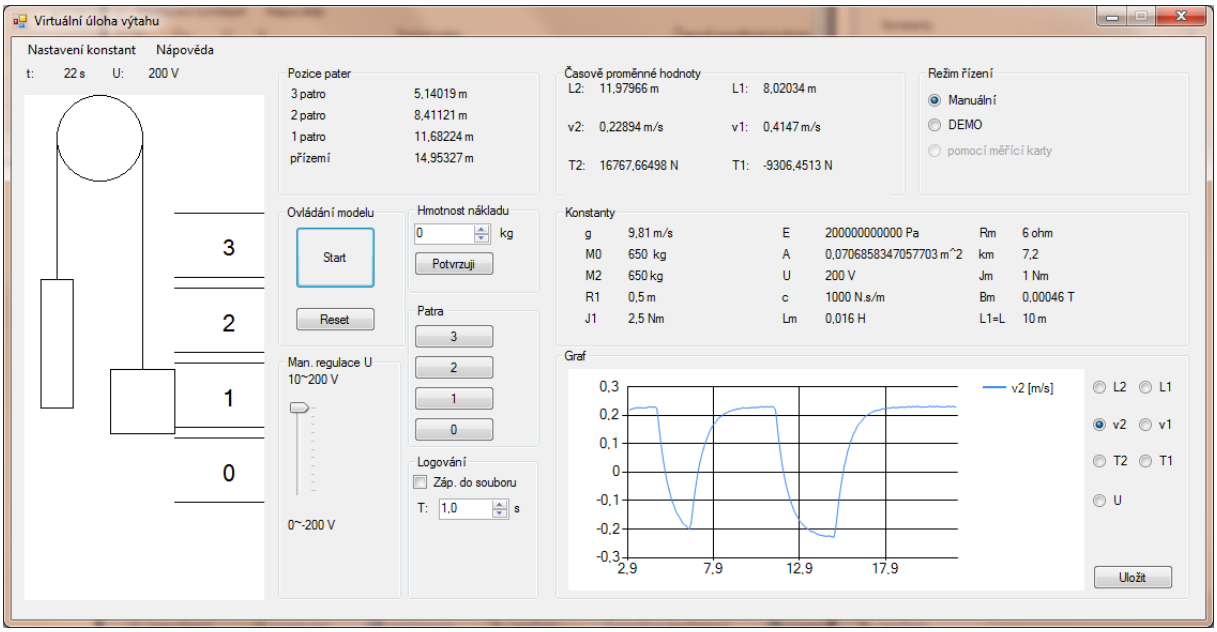


Obrázek 8 – diagram programu

V rámci diagramu předpokládáme již zinicializovanou soustavu prvků. Správné fungování modelu zabezpečuje blok Numerický model. Do tohoto bloku vstupuje hodnota aktuálního napětí motoru ze vstupu měřící karty vedoucí z řídícího zařízení. Výstup numerického modelu dodává měřící kartě informace o délce lana. Dále výstupy numerického modelu využívají vizualizační části a ukládání do souboru.

2.1 Uživatelské rozhranní

Cílem bylo jednoduché, uživatelsky přívětivé rozhranní (UI) s intuitivně uspořádanými prvky. V levé části UI se zobrazuje informace o aktuálním čase a napětí. Pod těmito informacemi je zobrazena vizualizace modelu. Informace o pozici pater je důležitá zejména pro programátora externího řídicího systému. Pod ním jsou tlačítka Start (Stop) a Reset, jež slouží k spuštění (zastavení), či restartování modelu. Následuje prvek umožňující manuální regulaci napětí. Dále je umožněno zadávat hmotnost nákladu v kg. Tlačítka pro volbu patra simulují klávesnici výtahu. Smysl použití mají v demo režimu nebo režimu měřící karty. Níže uložený ovládací prvek checkBox slouží k aktivaci zápisu do souboru s možností nastavit délku periody ukládání. Časově proměnné hodnoty reprezentují informace o aktuálních hodnotách proměnných modelu. Následují konstanty používané v aktuální simulaci. Pod konstantami je zobrazen graf s možností volby veličiny dle prvků radioButton umístěných napravo od grafu.



Obrázek 9 - uživatelské rozhranní programu

Pro snadné pochopení jednotlivých veličin je v UI přítomna kontextová nápověda. Byla využitá například k popisu časově proměnných hodnot. Aktivuje se po najetí kurzorem na daný text.

2.2 Režimy řízení

Virtuální model můžeme řídit třemi způsoby.

2.2.1 Manuální režim

Základní režim, řízení modelu probíhá pomocí komponenty trackBar, která umožní manuálně zvolit napětí. Tento režim umožňuje plnou kontrolu nad modelem.

2.2.2 Demo režim

Tento režim vychází z defaultních hodnot modelu. Výtah je ovládán automaticky na základě informace o zvoleném patře, která byla předána pomocí jednoho z tlačítek pro volbu poschodí. Režim řízení pracuje s informací o délce lana u kabiny. Na základě této hodnoty ovládá napětí motoru. Aby nedocházelo ke zmatkům

2.2.3 Režim měřící karty

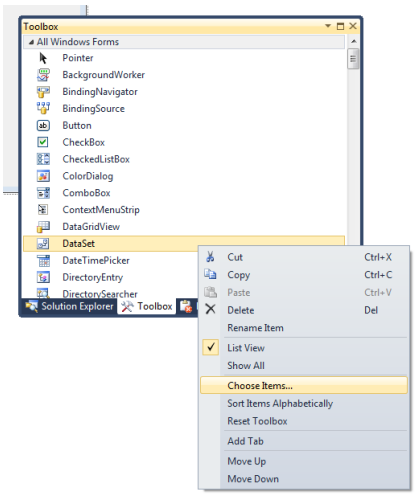
Tento režim umožňuje řízení pomocí měřící karty Advantech PCI 1711. Virtuální model je plně ovládán na základě napětí přivedeného na AI0.

2.3 Řízení času

O řízení reálného času se postaral časovač Multimedia Timer. Tento timer má nejvyšší přesnost.

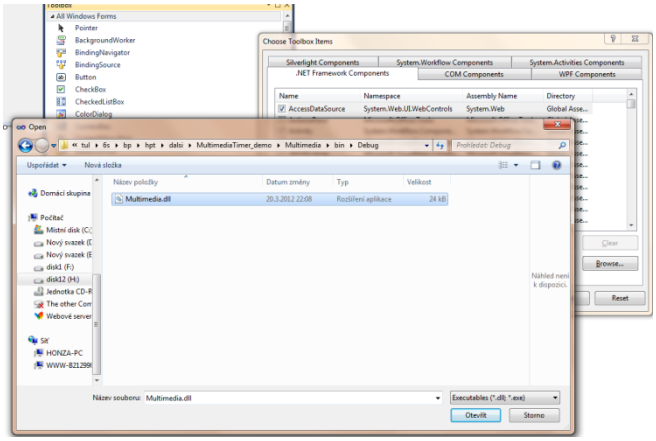
2.3.1 Implementace dynamické knihovny Multimedia Timer

U vytvořeného projektu nejprve klikneme pravým tlačítkem myši na libovolný objekt v okně Toolbox. Zvolíme Choose Items…



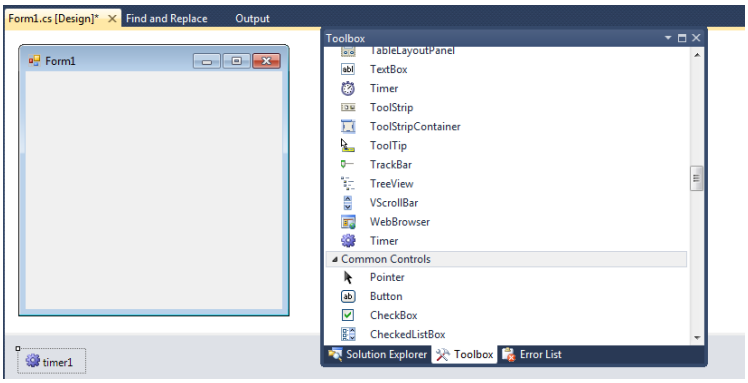
Obrázek 10 – Přidávání objektu

V nabídce Choose Toolbox Items klikneme na tlačítko Browse a vložíme dynamickou knihovnu Multimedia.dll.



Obrázek 11 – Přidávání timeru

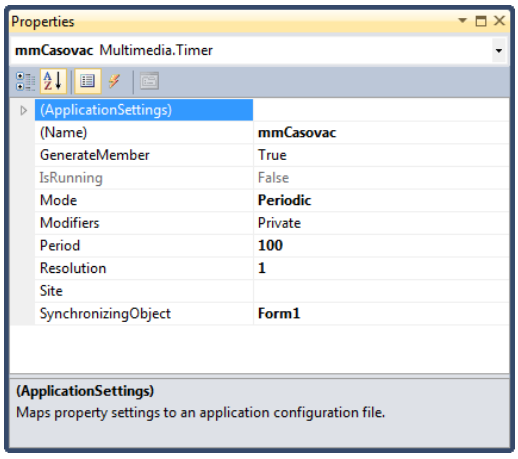
Nyní se v okně Toolbox zobrazí komponenta Timer, kterou pro zakomponování stačí přesunout do okna programu.



Obrázek 12 – Přidaný timer

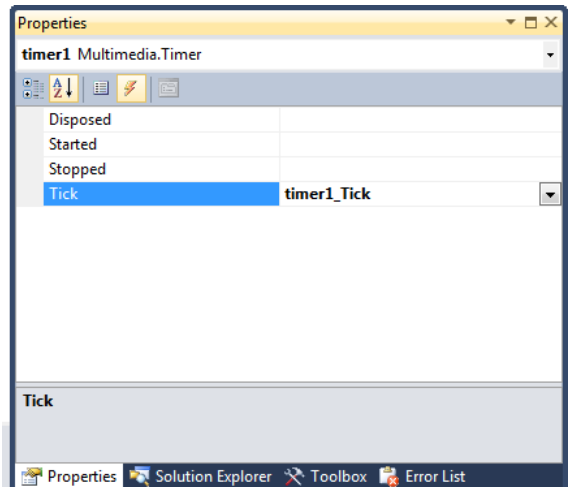
2.3.2 Nastavení vlastností

V okně Properties můžeme měnit výchozí vlastnosti komponenty. V prvé řadě je nutné nastavit synchronizaci. V tomto případě probíhá synchronizace přímo s UI Form1. Tímto se vyhneme mnohdy složitému použití delegátů. Je však nezbytné nastavit vhodně dlouhou periodu, aby mohli proběhnout všechny části cyklu. V případě příliš krátké periody může dojít k přerušení některých procesů cyklu, což může způsobit pád programů. Po důkladném otestování na PC s procesory AMD Athlon II, AMD E-350 a Intel i5 jsem dospěl k hodnotě 100 ms, která se jeví jako optimální z hlediska řízení výtahu i výpočetní náročnosti software.



Obrázek 13 – Nastavení periody

Dvojklikem do volného pole napravo od Tick vytvoříme metodu, jejíž obsah se bude vykonávat během dané periody.



Obrázek 14 – Použití události Tick

2.4 Spolupráce s měřící kartou

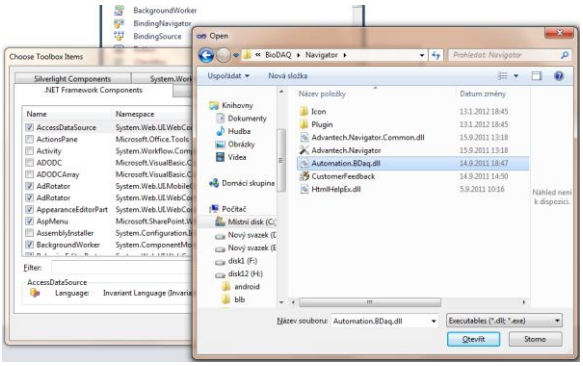
Měřící karta slouží k řízení modelu. Seznam použitých kanálů je zobrazen tabulce 2.

Tabulka 2 – Seznam použitých kanálů

AI0	Vstupní informace o napětí motoru
AO0	Výstupní informace o délce lana
DO0	Logická "1" v případech volby přízemí, 2 nebo 3 patra
DO1	Logická "1" v případech volby 1, 2 nebo 3 patra
DO2	Logická "1" v případech volby 3 patra

2.4.1 Implementace knihoven

Knihovna pro měřící kartu Advantech se přidává stehjně jako v případě časovače. Přidáváme soubor AutomationBdaq.dll.



Obrázek 15 – Přidání knihovny Automation.BDaq.dll

V Toolboxu jsou nyní objekty Advantech.

2.4.2 Použité metody

V programu jsem využil 3 základních objektů pracující s měřící kartou.

InstantAOCtrl

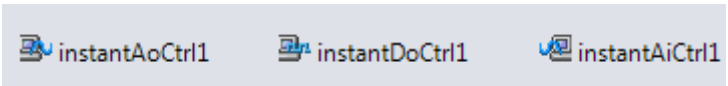
Na analogový výstup vyvolá napětí podle parametrů (port, napětí). Výstup je kontinuální do zavolání jiné funkce přepisující tuto hodnotu.

InstantAICtrl

Čte aktuální hodnotu napětí z analogového výstupu podle parametru (port).

InstantDOCtrl

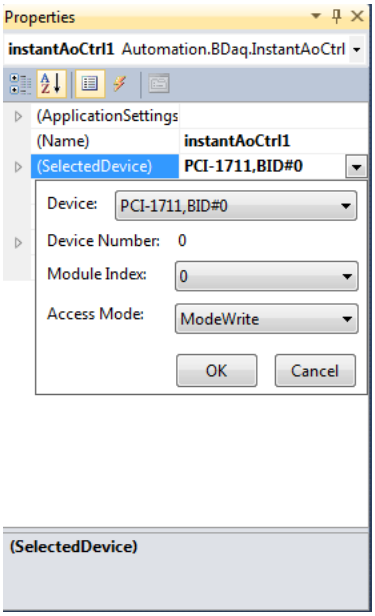
Na digitální výstup vyvolá 1/0 podle parametrů (port, binární hodnota). Oba porty jsou složeny z 8 kanálů. Výstup odpovídá binární soustavě. Výstup je kontinuální do přepsání této hodnoty



Obrázek 16 – objekty Advantech vložené do Form1

2.4.3 Nastavení vlastností

Po přidání objektů je nutné určit zařízení. To se provádí pomocí formuláře vlastností jednotlivých objektů. V případě nedostupnosti karty PCI 1711 je možné použít demo kartu. Demo karta je součástí instalace Advantech BioDAQ SDK.



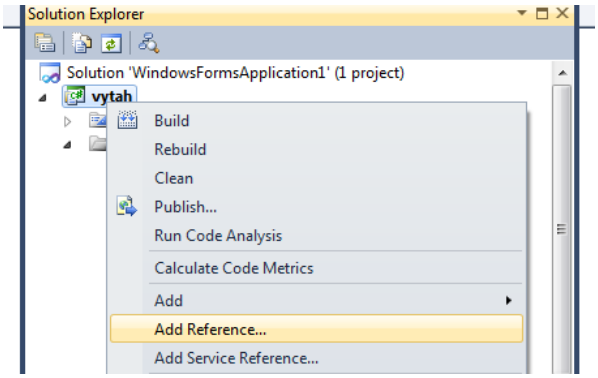
Obrázek 17 – Volba měřící karty

2.5 Simulace

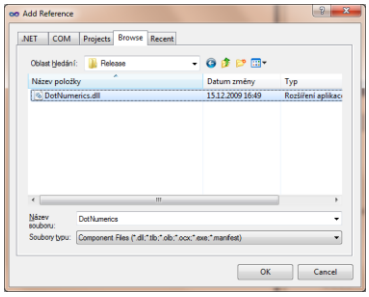
Pro simulaci modelu výtahu je nutné použít metody na numerické řešení soustav obyčejných diferenciálních rovnic v reálném čase. Zvolil jsem knihovnu DotNumeric, umožňující nastavit velikost kroku metody Runge-Kutta.

2.5.1 Implementace knihovny DotNumeric

Implementace této knihovny se provádí pomocí přidání reference a následného přidání direktivy.



Obrázek 18 – přidání reference



Obrázek 19 – Dynamická knihovna

Direktivu přidáme pomocí `using DotNumerics;` na začátek souboru, kde v dané třídě s touto knihovnou operujeme.

2.5.2 Řešící část

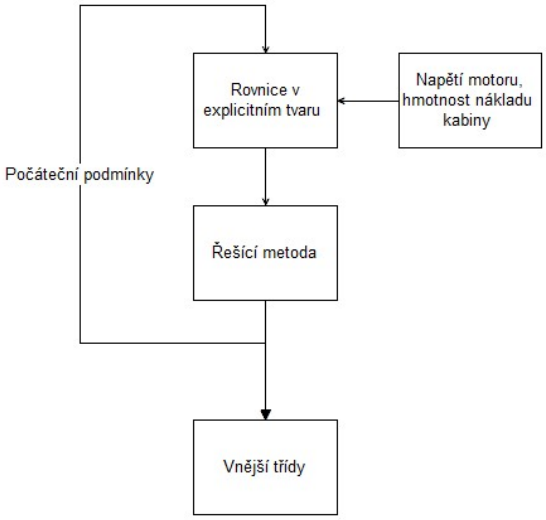
Dalším krokem bylo sestavení dodaných rovnic (123) do explicitního tvaru vhodného pro metodu řešící diferenciální rovnice. Zde jsem vycházel z dodaného schématu vytvořeného v prostředí Matlab Simulink. Rovnice jsem sestavil na základě předchozích dějů každého z integrátoru. Tyto rovnice jsem posléze dosadil do funkce ve třídě model.cs.

```
  private double[] ODEs(double t, double[] y)
        {
  dy[0] = (Um - y[0] * Rm - y[1] * km) / Lm;// integrator7
  dy[1] = (((y[0] * km / R1) - y[3] + y[4]) * R1 - y[1] * Bm) / Jm; // integrator9
  dy[2] = (((y[0] * km) / R1 - y[3] + y[4]) * R1 * R1) / J1; // integrator4
  dy[3] = (((y[2] + (-1) * y[6]) * E * A) - y[3] * y[2]) / y[7]; // integrator2
  dy[4] = ((y[5] - y[2]) * E * A + (y[2] * y[3]) - y[4] * y[5]) / y[8]; // integrator1
  dy[5] = g - y[4] / (M2+Mn) - y[5] * (c / (M2+Mn));// integrator
  dy[6] = y[3] / M0 - (y[6] * c / M0) - g; // integrator3
  dy[7] = y[6] * (-1); // integrator5
```

```
  dy[8] = y[5]; // integrator6

         return this.dy;
        }
```

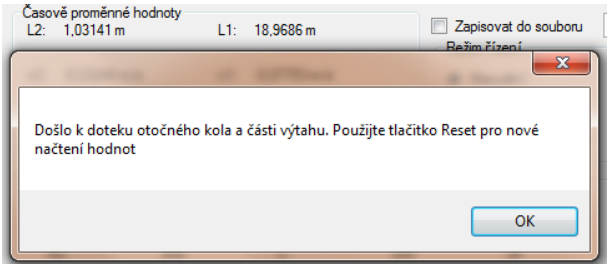
Z obrázku 20 je vidět, jak probíhá numerická výpočetní část simulace. Řešící metoda dostane během každého volání timerem vždy diferenciální rovnice v explicitním tvaru, jejichž počáteční podmínky odpovídají předchozímu výpočtu. Zároveň bude do těchto rovnic dosazena hodnota aktuální hodnota napětí a hmotnosti kabiny. Řěšící metoda používá 10 mezikroků a délka kroku je 100 ms, což odpovídá periodě časovače. Z toho plyne, že metoda může registrovat změny napětí a hmotnosti kabiny jen každých 0.1 s.



Obrázek 20 – Diagram bloku numerický model

Kolize otočného kola a kabiny nebo závaží není žádoucí. Proto bylo potřeba sestavit funkci, která ověřuje, jestli k doteku nedošlo. Pokud nastala kolize, funkce vrátí hodnotu false. Metoda využívající tuto funkci zastaví časovač, zobrazí informaci dle obrázku 20 a opětovné spuštění modelu bude možné až po restartu.

```
  public bool over_dotyk() // ověří, jestli došlo k dotyku otočného kola a kabiny nebo
//závaží
        {
            double hodnota = Math.PI * R1 / 2 + R1;
            if (y[8] <= hodnota || y[7] <= hodnota)
            { return false; }
            else
                return true;
        }
```



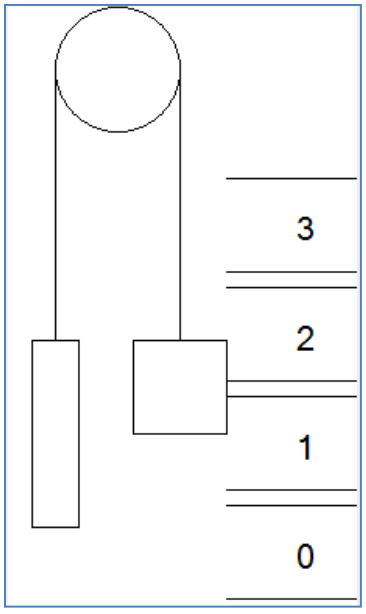
Obrázek 21 – Chybová hláška oznamující kolizi mezi částí výtahu a otočným kolem

2.7 Grafický výstup

2.7.1 Vizualizace modelu

Pro vizualizaci modelu je třeba vytvořit vizuální model v relativních souřadnicích. Relativní souřadnice reprezentují body na obrazovce. Reálné souřadnice odpovídají vypočtené vzdálenosti v modelu.

Vizualizace je řešena pomocí panelu, na kterém je zobrazován model výtahu. Tento panel je vytvořen jako samostatná komponenta s názvem VisualPanel. Při vytváření vyobrazení modelu byla rozhodující čitelnost a rychlost vykreslování.



Obrázek 22 – vizualizace modelu

VisualPanel vykresluje grafické prvky na základě hodnoty o délce lana u kabiny.

```
  public void refresh(double delka) // obnoví relativní souřadnice závaží a kabiny na
//základě délky lana u kabiny
        {
            yv = Convert.ToInt32(214/lano *delka); //souradnice y kabina
            yz = Convert.ToInt32((214/(lano))*(2*lano-delka)); // souradnice y zavazi
        }
```

Pro správné fungování převodu z reálných do relativních souřadnic se před prvním vykreslením definuje celková délka lana. Na základě této hodnoty se určí i souřadnice pro jednotlivá podlaží.

2.7.2 Graf

Pro tvorbu grafu slouží komponenta Chart, která je součástí Visual Studia. Umožňuje dynamickou změnu měřítka os na základě vkládaných hodnot. Součástí je i metoda pro uložení grafu do souboru.

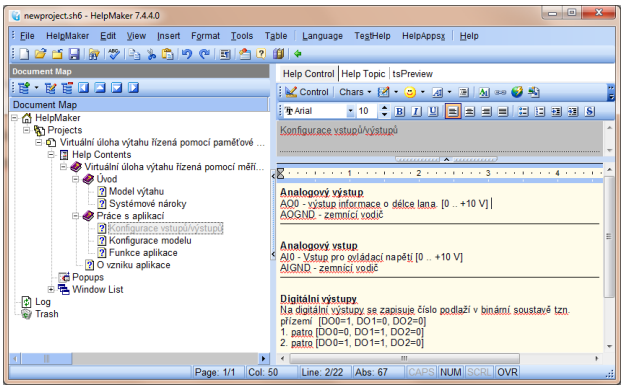
2.8 Logování hodnot

Log je zapisován jako soubor ve formátu CSV. Jedná se o jednoduchý systém zápisu s hodnotami oddělenými středníky, umožňující snadnou editaci a zobrazení např. v MS Excel. Formát ukládání je patrný ze zdrojového kódu. Nejprve se zapíše aktuální čas a čas timeru, pak následuje *L2*, *v2*, *T2*, *v1* a *T1*. Hodnoty se zapisují na základě nastavené periody. Vkládání nových hodnot je prováděno vždy na začátek souboru log.csv, který je obsažen ve složce programu.

```
   // zapise cas mereni;cas timeru;hodnotu napeti; hodnotu L2;v2;T2;L1;v1;T1
                    string zapis = String.Format("{0:yyyy-MM-dd-hh-mm-ss}",
DateTime.Now) + ";" + vytah.lano.ToString() + ";" + vytah.rychlost.ToString() + ";" +
vytah.pnuti.ToString() + ";" +
   vytah.lanoZ.ToString() + ";" + vytah.rychlostZ.ToString() + ";" +
vytah.pnutiZ.ToString();
```

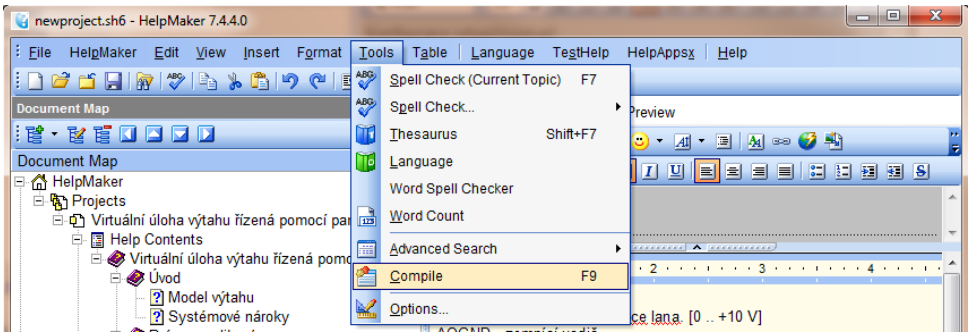
2.9 Tvorba nápovědy

Kvalitní nápověda přispívá k rychlejšímu pochopení uživatelů a snadnému uvedení do problematiky. Pro tvorbu jednoduché nápovědy mi jako vhodný formát připadal Microsoft Compiled HTML Help. Jedná se o shlukované HTML stránky v souboru CHM. Pro vytvoření souboru tohoto typu je nutné mít nainstalovaný kompilátor Microsoft Help Workshop. Pro snadné vytváření podporovaného obsahu je vhodné použít některý z volně dostupných WISIWYG editorů. Osvědčil se mi HelpMaker 7.4.4.0, který má přehledný, logicky uspořádaný layout. Po založení projektu lze v části Document map vytvořit stromovou strukturu jednotlivých stránek nápovědy, které lze po kliknutí dále editovat v pravé části okna.



Obrázek 23 – Okno programu HelpMaker

Pro vytvoření zkompilovaného souboru stačí stisknout klávesu F9 nebo v horní liště zvolit Tools -> Compile.



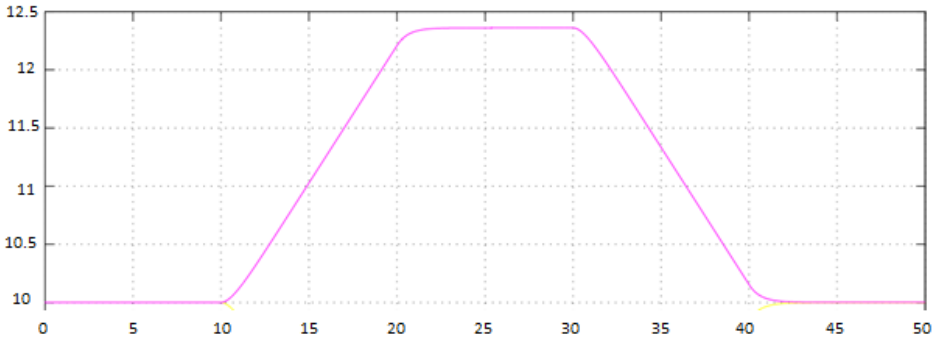
Obrázek 24 – Kompilace nápovědy

Vyvolat nápovědu je možné v horní liště programu. V nápovědě jsou popsány veškeré možnosti aplikace a její nastavení, včetně určení jednotlivých vstupů / výstupů měřící karty. Popsaný model výtahu je rovněž součástí včetně obrázku.

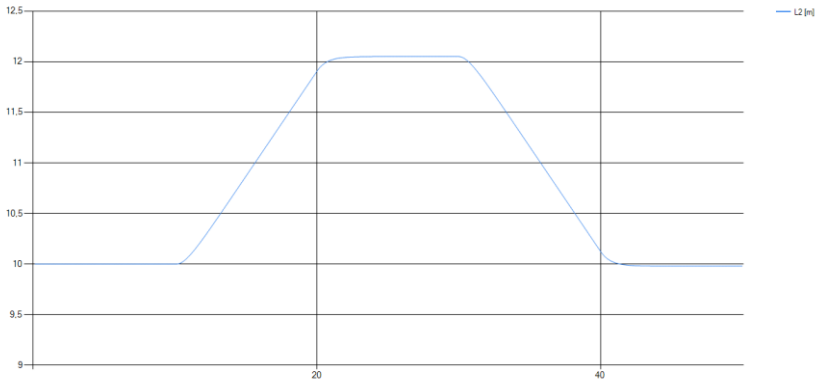
3 Testování

3.1 Porovnání modelů

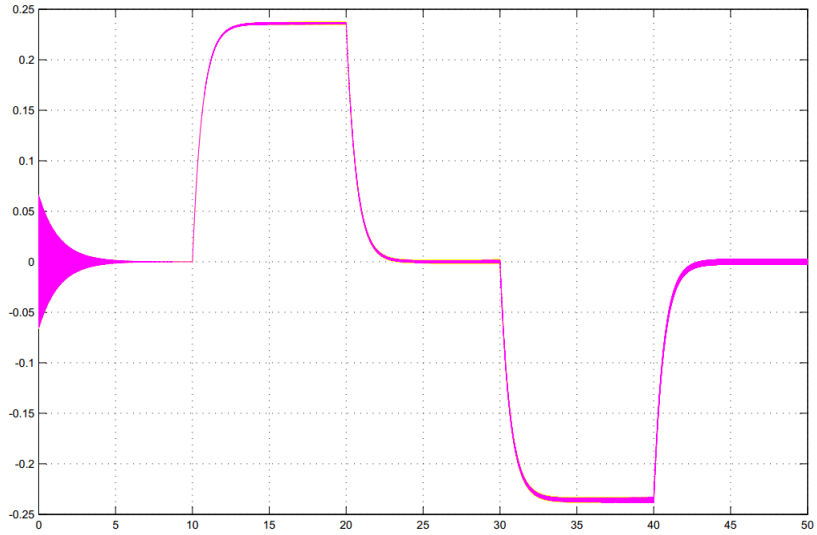
První fází bylo srovnání grafů časových průběhů veličin dodaného modelu vytvořeného pomocí Matlab Simulink a v mnou vytvořené aplikaci. Grafy jsou vytvořené pomocí vestavěných nástrojů. U obou případů jsem zaznamenal průběhy bez větších odlišností. Modely lze tedy považovat za shodné.



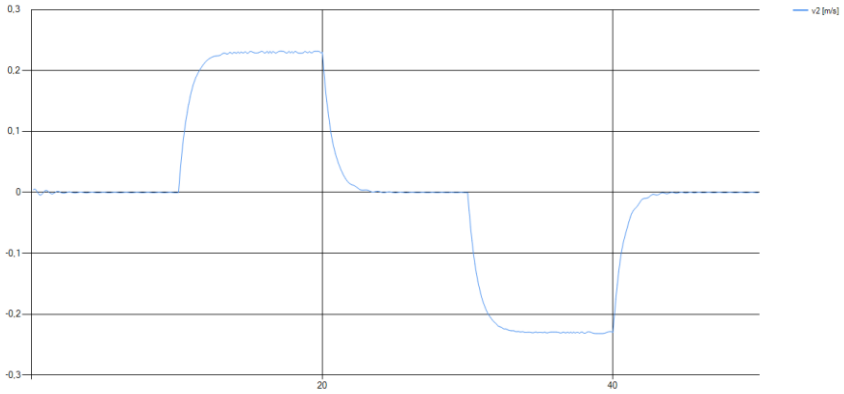
Obrázek 25 – Průběh délky lana kabiny ve výchozím modelu



Obrázek 26 – Průběh délky lana kabiny ve vytvořené aplikaci



Obrázek 27 – Průběh rychlosti pohybu lana u kabiny v prostředí Matlab Simulink



Obrázek 28 – Průběh rychlosti lana u kabiny ve vytvořeném programu

3.2 Testování za využití měřící karty s řízením pomocí zdroje

Funkčnost programu jsem v první fázi testoval pomocí digitálního multimetru UNI-T UT60A a stejnosměrného zdroje UNI-T UTP3703. Karta byla připojena k terminálu Advantech PCLD-8710. Zdroj jsem připojil kladným pólem na analogový výstup AI0 a záporným na AIGnd. Na napěťovém zdroji jsem volil napětí v rozsahu 0 .. 10 V. Multimetr jsem využil pro sledování napětí analogového výstupu AO0 a digitálních výstupů DI0, DI1, DI2.

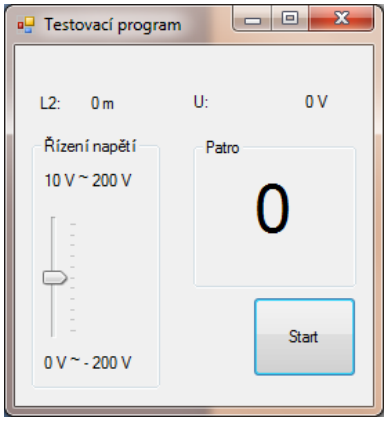
3.3 Testování za využití měřící karty s řízením pomocí druhého PC s měřící kartou

Protože se první testování zdařilo, vytvořil jsem jednoduchý testovací program jako příklad komunikace mezi dvěma zařízeními. K dispozici jsem měl dva PC, kde byl každý vybaven měřící kartou Advantech PCI 1711 a dva terminály Advantech PCDL-8710. Propojení bylo realizováno pomocí 8 vodičů, jejichž zapojení je patrné z tabulky 3.

Tabulka 3 – Tabulka zapojení mezi kartami u virtuálního modelu a řídícího systému

mk / prop.	1	2	3	4	5	6	7	8
Model	AI0	AI GND	AO0	AO GND	DO0	DO1	DO2	D GND
Řídící	AO0	AO GND	AI0	AI GND	DI0	DI1	DI2	D GND

Testovací program je vytvořen pro model výtahu využívající defaultní hodnoty proměnných. V okně programu se zobrazuje aktuální délka lana, zvolené řídící napětí a patro získané z digitálních vstupů karty. Pomocí komponenty trackBar můžeme volit různá napětí a řídit tak model výtahu na druhém PC.



Obrázek 29 – Testovací program

Zvolil jsem periodu časovače 30 ms z důvodu nižších nároků na výpočetní výkon u tohoto programu. Komunikace probíhala bez problému, veškeré hodnoty se zobrazovaly dle očekávání.

Závěr

Cíl práce, vytvoření virtuálního modelu ovládaného měřící kartou, se zdařil. Model je možné ovládat manuálně v programu, automaticky pomocí demo režimu nebo z vnějšího prostředí za využití měřící karty. Správné fungování aplikace potvrzuje úspěšné otestování v laboratoři.

Během vývoje aplikace se objevila řada problémů, jejichž řešení zabralo mnoho času. Jednalo se zejména o problém reálného času a soustavy diferenciálních rovnic. Samotné použití objektů komunikujících s měřící kartou bylo až překvapivě jednoduché.

Aplikaci lze považovat za ucelenou, zlepšení bych viděl v optimalizaci procesů za účelem zmenšení periody časovače.

Seznam literatury

[1] *Elevator Modeling and DC Drive Speed Controller Design*. [online]. [cit. 2012-05-18]. Dostupné z: http://www.apicsllc.com/apics/Sr_7/SR_7.html>

[2] *Evolution of logical arrays in MATLAB*. [online]. [cit. 2012-05-18]. Dostupné z: <http://blogs.mathworks.com/steve/2009/09/18/evolution-of-logical-arrays-in-matlab/>

[3] *Technical Overview of OpenModelica and its Development Environment*. POP, Adrian. ADRIAN POP. [online]. [cit. 2012-05-18]. Dostupné z: <http://www.modprod.liu.se/OpenModelica2012/1.322771/OpenModelica2012-talk10-Adrian-Pop-OpenModelica-current-status.pdf>

[4] *Scicos: Block diagram modeler/simulator*. [online]. [cit. 2012-05-18]. Dostupné z: <http://www.scicos.org>

[5] *Delphi History – from Pascal to Embarcadero Delphi XE 2*. [online]. [cit. 2012-05-18]. Dostupné z: <http://delphi.about.com/cs/azindex/a/dhistory.htm>

[6] *Počítačová cvičení Škola matematického modelování 2012*. [online]. [cit. 2012-05-18]. Dostupné z: http://skomam.vsb.cz/program/cviceni_SKOMAM2012.pdf>

[7] *INTUITIVE UNDERSTANDING OF ODE SOLVERS*. [online]. [cit. 2012-05-18]. Dostupné z: <http://alexkr.com/source-code/127/intuitive-understanding-of-ode-solvers/>

[8] *Runge-Kutta Methods*. [online]. [cit. 2012-05-18]. Dostupné z: <http://pauli.uni-muenster.de/tp/fileadmin/lehre/NumMethoden/SoSe10/Skript/RKM.pdf>

[9] *The Runge-Kutta Method*. [online]. [cit. 2012-05-18]. Dostupné z: <h ttp://www.physics.drexel.edu/courses/Comp_Phys/Integrators/rk4.html>

[10] *ALGLIB*. [online]. [cit. 2012-05-18]. Dostupné z: <http://www.alglib.net/ >

[11] *DotNumerics*. [online]. [cit. 2012-05-18]. Dostupné z: <http://www.dotnumerics.com/>

[12] *Linux jako real-time systém.* [online]. [cit. 2012-05-18]. Dostupné z: http://www.cs.vsb.cz/Files/other_files/mschmidt-realtime.pdf>

[13] *Comparing the Timer Classes in the .NET Framework Class Library*. [online]. [cit. 2012-05-18]. Dostupné z: <http://msdn.microsoft.com/en-us/magazine/cc164015.aspx>

[14] *The Multimedia Timer for the .NET Framework*. [online]. [cit. 2012-05-18]. Dostupné z: <http://www.codeproject.com/Articles/5501/The-Multimedia-Timer-for-the-NET-Framework>

[15] *Advantech PCI-1711/L datasheet*. [online]. [cit. 2012-05-18]. Dostupné z: <http://downloadt.advantech.com/ProductFile/Downloadfile3/1-32A8K9/PCI-1711_DS.pdf>

[16] *Advantech PCL-8710 Data Sheet*. [online]. [cit. 2012-05-18]. Dostupné z: http://downloadt.advantech.com/ProductFile/Downloadfile2/1-2KJIZO/PCLD-8710_DS.pdf