

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Simulace chování systému v reálném čase
Martin Horák

Bakalářská práce
2009

Univesity of Pardubice
Faculty of Electrical Engineering and Informatics

Simulation of system behavior in real time

Martin Horák

Bachelor work

2009

Prohlašuji:

Tuto práci jsem vypracoval/a samostatně. Veškeré literární prameny a informace, které jsem v práci využil/a, jsou uvedeny v seznamu použité literatury.

Byl/a jsem seznámen/a s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na náhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně Univerzity Pardubice.

V Pardubicích dne 10.8. 2009

Martin Horák

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Katedra informačních technologií
Akademický rok: 2008/2009

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Martin HORÁK**
Studijní program: **B2646 Informační technologie**
Studijní obor: **Informační technologie**
Název tématu: **Simulace chování systému v reálném čase**

Z á s a d y p r o v y p r a c o v á n í :

Cíl: vytvořit aplikaci v PROMOTICu, která simuluje dynamické chování zadaného systému v reálném čase
Teoretická část: a) numerické řešení diferenciálních rovnic metodou Runge-Kuta b) nastudovat, tvorbu aplikace v prostředí PROMOTIC, použití VBS pro výpočty
Praktická část: vytvořit aplikaci v PROMOTICu zobrazující simulované chování zadané soustavy v reálném čase
Tvorba aplikace zahrnuje: a) převod zadané soustavy (diferenciální rovnice) do diskrétního výpočetního tvaru b) realizace výpočetní části v VBS (Visual Basic Script) c) vytvoření aplikací umožňující v průběhu simulace měnit parametry a průběžně zobrazovat časové průběhy vstupních a výstupních veličin

Rozsah grafických prací:

Rozsah pracovní zprávy:

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

Seznam odborné literatury:

Rektorys, K. Přehled užití matematiky či jiná literatura zabývající se numerickými metodami dokumentace k systému PROMOTIC

Vedoucí bakalářské práce:

doc. Ing. František Dušek, CSc.
Katedra řízení procesů

Datum zadání bakalářské práce: **15. ledna 2009**

Termín odevzdání bakalářské práce: **15. května 2009**



doc. Ing. Simeon Karamazov, Dr.
děkan



Ing. Lukáš Čegan
vedoucí katedry

V Pardubicích dne 31. března 2009

Rád bych touto cestou poděkoval vedoucímu své bakalářské práce doc. Ing. Františku Duškovi CSc. za zadání, odborné vedení a cenné rady při vypracování bakalářské práce. Dále bych chtěl poděkovat své rodině za jejich duševní i materiální podporu a rodinné zázemí po celou dobu mého studia.

Souhrn

Tato bakalářská práce se zabývá vytvořením aplikace v SCADA / HMI nástroji PROMOTIC, která simuluje dynamické chování zadaného systému v reálném čase. Simulovaný systém představuje uzavřený regulační obvod se soustavou popsanou lineární diferenciální rovnicí max. 5. řádu a s diskretním regulátorem se dvěma stupni volnosti popsaným diferenční rovnicí max. 5. řádu.

Klíčová slova

PROMOTIC, diferenciální rovnice, Runge-Kutta, simulace, regulace

Summary

Create application in SCADA / HMI PROMOTIC tool, which simulates dynamical behavior system in real time. The simulated system is a control closed loop. The controlled system is described by a linear differential equation with maximum of 5th order and the discrete controller with two degree of freedom is described by a difference equation with maximum of 5th order.

Keywords

PROMOTIC, differential equation, Runge-Kutta, simulation, control

Obsah

1.	Úvod	10
2.	Teoretická část	12
2.1	Regulace	12
2.2	Dynamický systém	12
2.2.1	Diferenciální rovnice	13
2.2.2	Řešení ODE pomocí numerických metod	15
2.2.3	Metoda Runge-Kutta 4. řádu	15
2.3	Číslicový regulátor	16
2.4	Aditivní šum a porucha na vstupu	16
2.5	Časování a intervaly výpočtů	17
2.5.1	Synchronizace výpočtu s reálným čas	18
3.	Praktická část	19
3.1	SCADA systémy / HMI	19
3.2	PROMOTIC	19
3.2.1	Použité PROMOTIC objekty	21
3.3	Uživatelské rozhraní aplikace	22
3.3.1	Aplikační logika	23
3.4	Formát souborů s konfigurací	24
3.5	Algoritmy v PROMOTICu	25
3.5.1	Jazyk VBScript	25
3.5.2	Implementace výpočtů a časování	25
3.6	Grafické znázornění průběhů veličin	27
4.	Závěr	29
5.	Literatura	30

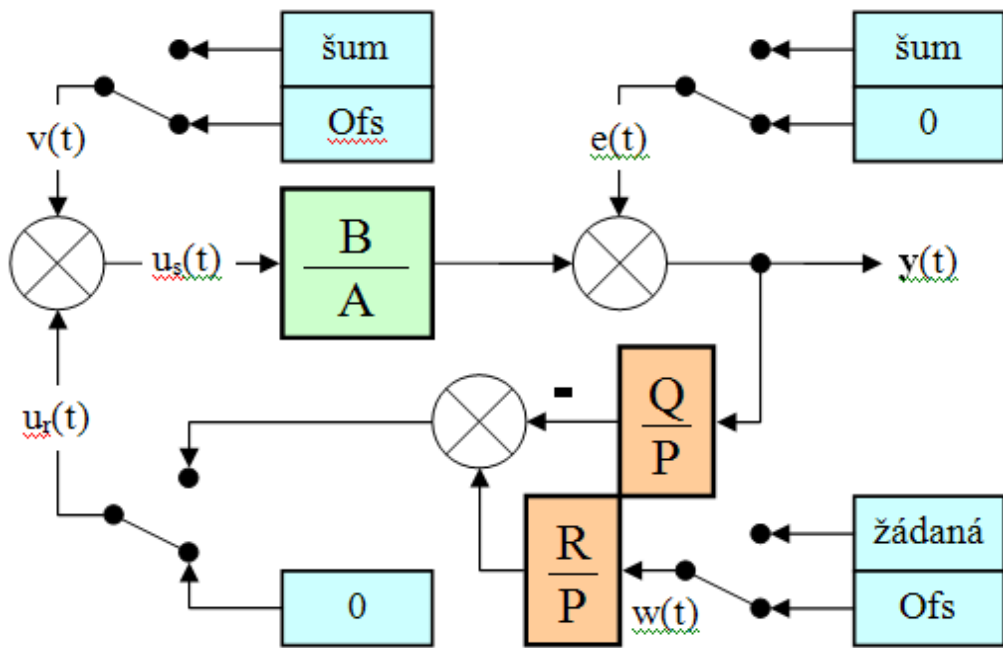
1. Úvod

Bakalářská práce se zabývá simulací spojitého dynamického lineárního systému, s poruchou na vstupu a aditivním šumem na výstupu, řízeného číslicovým lineárním regulátorem v reálném čase. Systém, jehož základní schéma je na Obr. č. 1, obsahuje regulovanou soustavu popsanou formou obecné lineární diferenciální rovnice maximálně 5. řádu, číslicový regulátor se dvěma stupni volnosti popsaný diferenční rovnicí maximálně 5. řádu, dvěma bloky pro generování poruchové veličiny na vstupu a šumu na výstupu soustavy a blokem generování žádané hodnoty.

Cílem práce je poskytnout možnost sledování a zaznamenávání časových průběhů vybraných veličin lineárního dynamického systému včetně možnosti nastavení, uložení a obnovení všech parametrů systému. Aplikaci je možno použít jako simulační program pro ověření chování navržených diskrétních regulátorů a získání představy o jejich chování za ideálních podmínek. Uživatel může měnit chování simulovaného systému kromě parametrů diferenciální rovnice také změnou průběhu poruchové veličiny na vstupu a aditivního šumu na výstupu. Také je možné měnit časový průběh žádané hodnoty regulátoru. Práce se skládá ze dvou částí.

V první části práce jsou uvedeny teoretické informace a základní pojmy související s jednotlivými částmi blokového schématu.

Druhá část je zaměřena na tvorbu aplikace v SCADA / HMI systému PROMOTIC, který je použit nejen pro tvorbu uživatelského rozhraní a vizualizaci ale i pro numerický výpočet diferenciálních a algebraických rovnic popisující chování celého systému včetně synchronizace výpočtu s reálným časem.



Obr. č. 1 Blokové schéma simulovaného systému

2. Teoretická část

2.1 Regulace

Abychom lépe porozuměli blokovému schématu, je nutné si vysvětlit pojem regulace, která je zvláštním druhem řízení.

Řízení je cílevědomá činnost, při níž se hodnotí a zpracovávají informace o řízeném objektu i informace o dějích vně tohoto procesu a podle nich se ovládají příslušná zařízení tak, aby bylo dosaženo jistého předepsaného cíle. Je-li řízení uskutečňováno samočinně nějakým zařízením nebo systémem, mluvíme o automatickém řízení.

Regulace je udržování určitých fyzikálních veličin (regulované veličiny) na předem stanovených hodnotách (žádané hodnoty). Při tom se v průběhu regulace zjišťují hodnoty regulovaných veličin a srovnávají se s žádanými hodnotami. Podle zjištění odchylek (regulační odchylky), které jsou mírou přesnosti regulace, se zasahuje do regulovaného procesu změnou akčních veličin, aby regulované veličiny co nejlépe sledovaly průběh žádaných hodnot.

Zařízení, které má být regulováno, nazýváme regulovanou soustavou a zařízení, které samočinně provádí regulaci, nazýváme regulátorem. Obě tyto části tvoří regulační obvod. Protože soustava i regulátor se skládají z jednotlivých členů, které samy o sobě jsou většinou složitými jednotkami (měřicí přístroje, zesilovače, různé stroje a zařízení), používáme pro schematické znázorňování regulačních obvodů blokových schémat.[1]

2.2 Dynamický systém

V našem blokovém schématu blok s označením B/A představuje regulovanou soustavu. Jedná se o matematický model, který reprezentuje fyzikální strukturu reálného objektu, s jasně definovanými vstupy a výstupy a se známými počátečními podmínkami [2]. Tento model je nejčastěji popsán soustavou diferenciálních a algebraických rovnic.

V našem případě soustavy s jedním vstupem $u(t)$ a jedním výstupem $y(t)$, je model popsán lineární diferenciální rovnicí řádu n ($1 \leq n \leq 5$) s derivací pravé strany maximálně řádu o stupeň nižšího, dopravním zpožděním T_d ve vstupní veličině $u(t)$, popisující průběh výstupní veličiny $y(t)$ v závislosti na časovém průběhu vstupních veličin a počátečních podmínkách.

$$\begin{aligned}
& y^{(5)}(t) + a_4 y^{(4)}(t) + a_3 y^{(3)}(t) + a_2 y^{(2)}(t) + a_1 y^{(1)}(t) + a_0 y(t) = \\
& = b_4 u^{(4)}(t - T_d) + b_3 u^{(3)}(t - T_d) + b_2 u^{(2)}(t - T_d) + b_1 u^{(1)}(t - T_d) + b_0 u(t - T_d) \\
& y^{(4)}(t = t_0) = y_{40} \quad u^{(4)}(t = t_0) = u_{40} \\
& y^{(3)}(t = t_0) = y_{30} \quad u^{(3)}(t = t_0) = u_{30} \\
& y^{(2)}(t = t_0) = y_{20} \quad u^{(2)}(t = t_0) = u_{20} \\
& y^{(1)}(t = t_0) = y_{10} \quad u^{(1)}(t = t_0) = u_{10} \\
& y(t = t_0) = y_{00} \quad u(t = t_0) = u_{00}
\end{aligned} \tag{1a}$$

kde $a_4, \dots, a_0, b_4, \dots, b_0$ jsou volitelné parametry

$y_{40}, \dots, y_{00}, u_{40}, \dots, u_{00}$ jsou počáteční podmínky v čase t_0

V oblasti řízení je zvykem psát popis soustavy ve formě přenosové funkce $F(s)$, která je ekvivalentní diferenciální rovnici až na to, že se při jejím použití předpokládají nulové počáteční podmínky. Symbol s je komplexní operátor integrální Laplaceovy transformace.

$$\frac{Y(s)}{U(s)} = F(s) = \frac{B(s)}{A(s)} = \frac{b_4 s^4 + b_3 s^3 + b_2 s^2 + b_1 s + b_0}{s^5 + a_4 s^4 + a_3 s^3 + a_2 s^2 + a_1 s + a_0} e^{-T_d s} \tag{1b}$$

2.2.1 Diferenciální rovnice

Diferenciální rovnice je matematická rovnice, ve které jako proměnné vystupují derivace funkcí. Matematická teorie diferenciálních rovnic se zabývá existencí řešení, jednoznačností (čili zda je řešení jen jedno), závislostí řešení na počátečních a okrajových podmínkách. Ve fyzice a dalších aplikacích je zajímavé zejména získání analytického řešení, tedy funkce $u(t)$, která rovnici řeší. Pokud taková funkce nejde analyticky vyjádřit, vstupuje do hry numerické řešení diferenciálních rovnic [3]. Analytické řešení homogenní lineární diferenciální rovnice existuje vždy. V našem případě speciální pravé strany (derivace neznámého signálu) je nutné numerické řešení. Jedna z nejpoužívanějších numerických metod je jednokroková metoda Runge-Kutta 4.řádu, která je aplikována i pro tento případ. Abychom ji mohli použít pro diferenciální rovnice vyšších řádů, je nutné původní diferenciální rovnici (1a) převést na ekvivalentní soustavu lineárních diferenciálních rovnic řádu prvního. Převod není jednoznačný. Použil jsem variantu popsanou např. Ogatou [4] a její aplikace je následující:

$$\begin{aligned}
\frac{dx_1}{dt} &= -a_4x_1 + x_2 + b_4u(t - T_d) & x_1(t=0) &= x_{10} \\
\frac{dx_2}{dt} &= -a_3x_1 + x_3 + b_3u(t - T_d) & x_2(t=0) &= x_{20} \\
\frac{dx_3}{dt} &= -a_2x_1 + x_4 + b_2u(t - T_d) & x_3(t=0) &= x_{30} \\
\frac{dx_4}{dt} &= -a_1x_1 + x_5 + b_1u(t - T_d) & x_4(t=0) &= x_{40} \\
\frac{dx_5}{dt} &= -a_0x_1 + b_0u(t - T_d) & x_5(t=0) &= x_{50}
\end{aligned} \tag{2}$$

Pro takto vyjádřenou soustavu je potřeba přepočít počátečních podmínek původní diferenciální rovnice vyššího řádu na počáteční podmínky soustavy diferenciálních rovnic. Přepočít je pak realizován podle vztahů (3b):

$$\begin{aligned}
y(t) &= x_1(t) \Rightarrow y_0 = x_{10} \\
\frac{dy}{dt} &= \frac{dx_1}{dt} - a_4y + x_2 + b_4u \\
\frac{d^2y}{dt^2} &= -a_4 \frac{dy}{dt} + \frac{dx_2}{dt} + b_4 \frac{du}{dt} = -a_4 \frac{dy}{dt} - a_3y + x_3 + b_3u + b_4 \frac{du}{dt} \\
\frac{d^3y}{dt^3} &= -a_4 \frac{d^2y}{dt^2} - a_3 \frac{dy}{dt} + \frac{dx_3}{dt} + b_3 \frac{du}{dt} + b_4 \frac{d^2u}{dt^2} = -a_4 \frac{d^2y}{dt^2} - a_3 \frac{dy}{dt} - a_2y + x_4 + b_2u + b_3 \frac{du}{dt} + b_4 \frac{d^2u}{dt^2} \\
\frac{d^4y}{dt^4} &= -a_4 \frac{d^3y}{dt^3} - a_3 \frac{d^2y}{dt^2} - a_2 \frac{dy}{dt} + \frac{dx_4}{dt} + b_2 \frac{du}{dt} + b_3 \frac{d^2u}{dt^2} + b_4 \frac{d^3u}{dt^3} = -a_4 \frac{d^3y}{dt^3} - a_3 \frac{d^2y}{dt^2} - a_2 \frac{dy}{dt} - a_1y + x_5 + b
\end{aligned} \tag{3a}$$

$$\begin{aligned}
x_{10} &= y_0 \\
x_{20} &= y_1 + a_4y_0 - b_4u_0 \\
x_{30} &= y_2 + a_4y_1 + a_3y_0 - b_3u_0 - b_4u_1 \\
x_{40} &= y_3 + a_4y_2 + a_3y_1 + a_2y_0 - b_2u_0 - b_3u_1 - b_4u_2 \\
x_{50} &= y_4 + a_4y_3 + a_3y_2 + a_2y_1 + a_1y_0 - b_1u_0 - b_2u_1 - b_3u_2 - b_4u_3
\end{aligned} \tag{3b}$$

Stejný postup se pak aplikuje i na diferenciální rovnice nižších řádů a tím získáme následující soustavy rovnic:

4. řád

$$\begin{aligned}
x_{10} &= y_0 \\
x_{20} &= y_1 + a_3y_0 - b_3u_0 \\
x_{30} &= y_2 + a_3y_1 + a_2y_0 - b_2u_0 - b_3u_1 \\
x_{40} &= y_3 + a_3y_2 + a_2y_1 - b_1u_0 - b_2u_1 - b_3u_2
\end{aligned} \tag{4}$$

3. řád

$$\begin{aligned}x_{10} &= y_0 \\x_{20} &= y_1 + a_2 y_0 - b_2 u_0 \\x_{30} &= y_2 + a_2 y_1 + a_1 y_0 - b_1 u_0 - b_2 u_1\end{aligned}\tag{5}$$

2. řád

$$\begin{aligned}x_{10} &= y_0 \\x_{20} &= y_1 + a_1 y_0 - b_1 u_0\end{aligned}\tag{6}$$

2.2.2 Řešení ODE¹ pomocí numerických metod

Používají se v případech, kdy analytické řešení neexistuje nebo je příliš složité a to za použití pouze aritmetických a logických operací. Výsledky numerických metod jsou diskrétní hodnoty a nepřesnost vzniklá při výpočtu se udává jako odhad chyby [5].

Při řešení diferenciálních rovnic se používá řada numerických metod, vyvíjených od počátku minulého století. Metody lze rozčlenit podle mnoha kritérií do různých tříd a studovat přitom jejich efektivitu z hlediska přesnosti a numerické náročnosti [6].

2.2.3 Metoda Runge-Kutta 4. řádu

Jedná se o jednokrokovou metodu, což znamená, že stav systému v čase t_i vypočítáme na základě předcházejícího stavu t_{i-1} , tzn. nový stav systému odvodíme z předcházejícího stavu [7]. U metody Runge-Kutta 4. řádu je aproximace² stanovena nejen na základě hodnot předcházejícího stavu, ale i na základě hodnot ležících mezi stavy t_i a t_{i-1} , což vede k přesnějším výsledkům. Metoda má pro soustavu lineárních diferenciálních rovnic ve tvaru

$$\begin{aligned}\frac{d\mathbf{x}(t)}{dt} &= \mathbf{M}\mathbf{x}(t) + \mathbf{n}u(t - T_d) \\y(t) &= \mathbf{c}\mathbf{x}(t)\end{aligned}\tag{7}$$

následující podobu:

¹ ODE (Ordinary differential equation) – obyčejná diferenciální rovnice

² Aproximace je odhad hodnoty řešené v následujícím časovém kroku.

$$\begin{aligned}
\mathbf{k}_1 &= \mathbf{M}\mathbf{x}(t_{i-1}) + \mathbf{n}u(t_{i-1} - T_d) \\
\mathbf{k}_2 &= \mathbf{M}\left[\mathbf{x}(t_{i-1}) + \frac{1}{2}h\mathbf{k}_1\right] + \mathbf{n}u\left(t_{i-1} + \frac{1}{2}h - T_d\right) \\
\mathbf{k}_3 &= \mathbf{M}\left[\mathbf{x}(t_{i-1}) + \frac{1}{2}h\mathbf{k}_2\right] + \mathbf{n}u\left(t_{i-1} + \frac{1}{2}h - T_d\right) \\
\mathbf{k}_4 &= \mathbf{M}\left[\mathbf{x}(t_{i-1}) + h\mathbf{k}_3\right] + \mathbf{n}u(t_{i-1} + h - T_d) \\
\mathbf{x}(t_i) &= \mathbf{x}(t_{i-1}) + \frac{1}{6}h[\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4] \\
y(t) &= \mathbf{c}\mathbf{x}(t)
\end{aligned} \tag{8}$$

kde h je časový krok výpočtu $h=t_i-t_{i-1}$

2.3 Číslicový regulátor

Je také nazýván diskretní (nespojité), protože vyhodnocuje veličiny pouze v určitých časových okamžicích. V praxi bývá realizován číslicovým počítačem.

Regulátor v pravidelných časových intervalech T snímá výstupní hodnoty $y(t_k=kT)$ sledovaného systému. Zápis diskretního času $t_k=kT$ se zjednodušuje a nahrazuje pouze symbolem pořadového čísla měření k . Zjištěné hodnoty spolu s hodnotami zaznamenanými v předchozích snímáních $y(k-n)$, $u(k-n)$ sledovaného systému a hodnotami žádané $w(k-n)$, regulátor použije k výpočtu nových hodnot regulačního výstupu $u(k)$. Tyto hodnoty mohou působit jako zpětná vazba na vstup sledovaného systému, s cílem ovlivnit systém, aby jeho výstupní hodnoty byly udržovány na předepsaných hodnotách $w(k)$. Výsledný diskretní regulátor je pak obecně popsán diferenční rovnicí (9), její řád a parametry jsou předmětem návrhu regulátoru.

$$\begin{aligned}
u(k) + p_1u(k-1) + \dots + p_nu(k-n_u) &= r_0w(k) + r_1w(k-1) + \dots + r_nw(k-n_w) + \\
&\quad - q_0y(k) - q_1y(k-1) - \dots - q_ny(k-n_y)
\end{aligned} \tag{9}$$

kde

$w(k)$	žádaná hodnota
$y(k)$	regulovaná veličina
$u(k)$	výstupní veličina regulátoru (akční)
p, r, q	parametry regulátoru

2.4 Aditivní šum a porucha na vstupu

Reálné systémy se obvykle nechovají ideálně podle předpokládaného matematického modelu. Odchytky od reálných průběhů signálů jsou způsobeny mnoha důvody. Mezi hlavní patří nedokonalost matematického popisu, působení signálů, které

nejsou do modelu zahrnuty (poruchy) a chyby měření (šum). Pro přiblížení chování simulovaného regulačního obvodu se tyto vlivy obvykle do simulace také zahrnují. V této práci je zahrnuta aditivní porucha $v(t)$ na vstupu soustavy se sinusovým průběhem a volitelnými parametry (amplituda A , perioda P a střední hodnota O) podle vztahu

$$v(t) = A \cdot \sin\left(\frac{2\pi}{P} \cdot t\right) + O \quad (10)$$

a aditivní diskretní šum měření $e(k)$ na výstupu soustavy s normalizovaným¹ normálním rozložením a volitelným posunem O a vahou A podle vztahu

$$e(k) = A \cdot N(1,0) + O \quad (11)$$

2.5 Časování a intervaly výpočtů

Interval časovače T (reálný čas)

- pevně zvolená hodnota (1 s), ostatní intervaly budou celočíselné násobky či podíly této hodnoty

Interval výpočtu diferenciální rovnice T_s

- celočíselný (n_s) podíl $T_s = T/n_s$ je použitý pro numerický výpočet spojité diferenciální rovnice, dopravní zpoždění je zaokrouhloveno na hodnotu nejbližší celočíselnému násobku intervalu T_s tj. $nT_d = \text{int}(T_d/T_s)$

Interval realizace číslicového regulátoru T_r

- určí se jako volitelný celočíselný násobek intervalu výpočtu diferenciální rovnice $T_r = n_r \times T_s$.

Interval výpočtu poruchy T_p a interval výpočtu žádané hodnoty T_w

- stejný jako interval výpočtu diferenciální rovnice

Interval výpočtu šumu T_e

- určí se jako volitelný celočíselný násobek n_e intervalu výpočtu diferenciální rovnice $T_e = n_e \times T_s$

¹ tj. s nulovou střední hodnotou a jednotkovým rozptylem

Interval zápisu historie pro grafické zobrazení T_h je stejný jako časovač T

2.5.1 Synchronizace výpočtu s reálným čas

Protože interval T je pevně zvolen (1 s) a intervaly výpočtů mohou být i v jiných časových intervalech než násobky této hodnoty, je nutné určit, kolikrát se dané výpočty v tomto časovém intervalu provedou. Pokud budeme mít intervaly $T_s=T_p=T_w=0,1$ s, $T_r=0,2$ s, $T_e=0,2$ s, $T_d=0$ s, budou výpočty probíhat následovně podle schématu (Obr. č. 1): základní interval bude T_s , protože ostatní intervaly jsou jeho celočíselné násobky (tzn., že jednotlivé časové kroky budou 0,1 s až do hodnoty 1 s).

Nejdříve generujeme aktuální hodnoty všech částí schématu v čase $t=0$ s a z počátečních podmínek zjistíme první hodnotu $y(t)$. Jestliže počítáme i s dopravním zpožděním, tak všechny hodnoty u_s posunuté o časový interval T_d , nastavíme na u_s použité při výpočtu první hodnoty $y(t)$.

V prvním kroku bude aktuální čas $t=0,1$ s. Vygenerují se hodnoty $v(t)$ a $u(t)$, jejichž součet u_s se následně použije pro výpočet spojitého systému. Pokud by byl interval $T_d>0$, použily bychom hodnoty u_s vygenerované o daný časový interval v minulosti. Dále vezmeme aktuální (v tomto kroku negenerujeme novou) hodnotu šumu $e(t)$ a přičteme ji k výstupu systému $y(t)$. Následuje vygenerování žádané hodnoty $w(t)$. Výpočet regulátoru neprovádíme. V dalším kroku bude aktuální čas $u_t=0,2$ s. Vše proběhne stejně jako v prvním kroku, s tím rozdílem, že vygenerujeme novou hodnotu šumu $e(t)$ a také provedeme výpočet regulátoru $u_t(t)$. Je to proto, že aktuální čas $t=0,2$ s je násobkem intervalu T_e a T_r . Další kroky provádíme identicky, proto je zde nebudeme uvádět. Tento postup aplikujeme každou sekundu. Jeho implementace v systému PROMOTIC bude popsána v kapitole Implementace výpočtů a časování.

3. Praktická část

Pro tvorbu aplikace byl vybrán systém PROMOTIC, který spadá pod tzv. SCADA systémy. Systém PROMOTIC obsahuje spoustu nastavení a možností pro tvorbu aplikací. Zde si vysvětlíme pouze základní princip tvorby aplikací a zaměříme se pouze na některé použité objekty. Pro podrobnější popis systémů odkazují na jeho nápovědu, která je k dispozici v češtině.

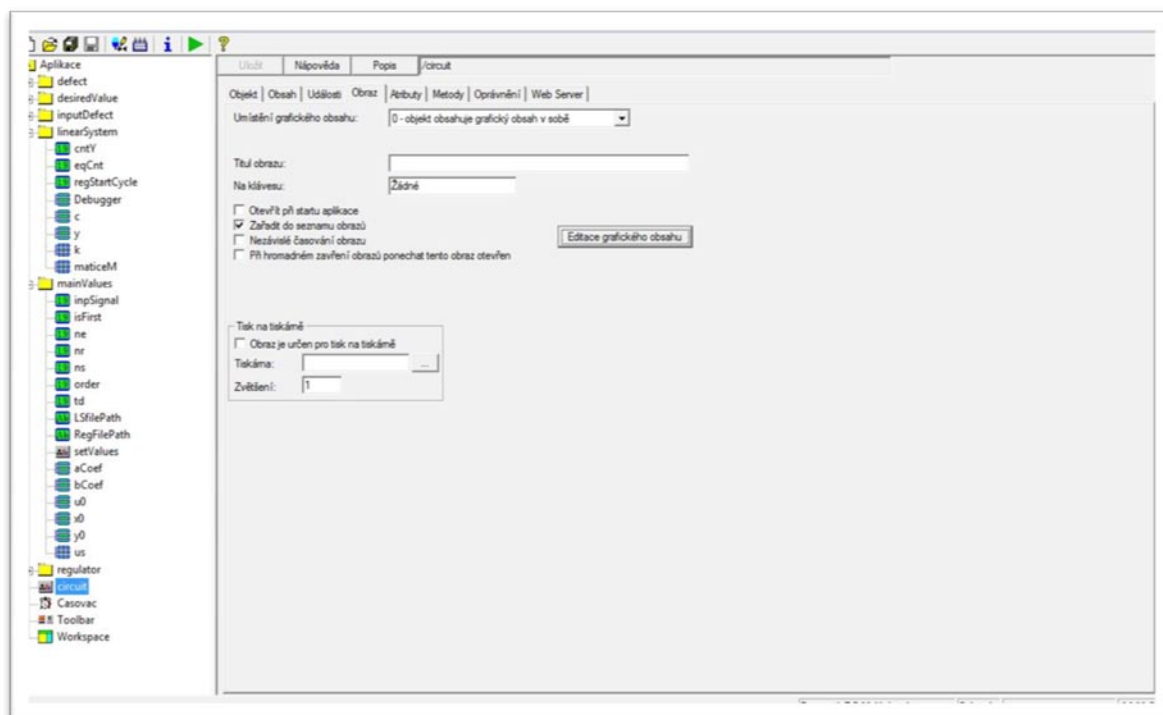
3.1 SCADA systémy / HMI

Jde o anglickou zkratku Supervisory Control And Data Acquisition, což se dá přeložit jako systémy pro řízení a získávání dat. Je doplňována dodatkem HMI (Human-Machine Interface), který upřesňuje, že se jedná o řešení operátorského rozhraní. U nás se spíše používá pojmenování vizualizační, řídicí a informační systémy. Pod tímto pojmem se rozumí nástroj, ve kterém je možné vytvořit program pro monitorování jednotlivých veličin (např. objem produkce, kvalita, atd.) a grafické zobrazení procesu výroby v nejrůznějších odvětvích průmyslu pro zviditelnění aktuálního stavu, umožňuje jeho ovládání a změnu parametrů a umožňuje implementovat spoustu dalších funkcí [8].

3.2 PROMOTIC

Za tímto systémem stojí firma MICROSYS, která vydala první verzi již v roce 1994. Od té doby se stále vyvíjí, aby se zlepšoval komfort a usnadňovaly práce při vývoji aplikací. Také se zlepšuje integrace a komunikace s ostatními moderními technologiemi, což z něj dělá systém se zcela otevřenou architekturou. Jako příklad podporovaných technologií zmiňme např. podnikové databáze (MS Sql, Oracle, MySQL, Access), zabudovaná rozhraní XML, ActiveX. Pro komunikaci s decentralizovanými zařízeními jsou k dispozici např. komunikační rozhraní s protokoly TCP/IP, OPC nebo HTTP. PROMOTIC je pro vývoj aplikací rozdělen do dvou částí [9].

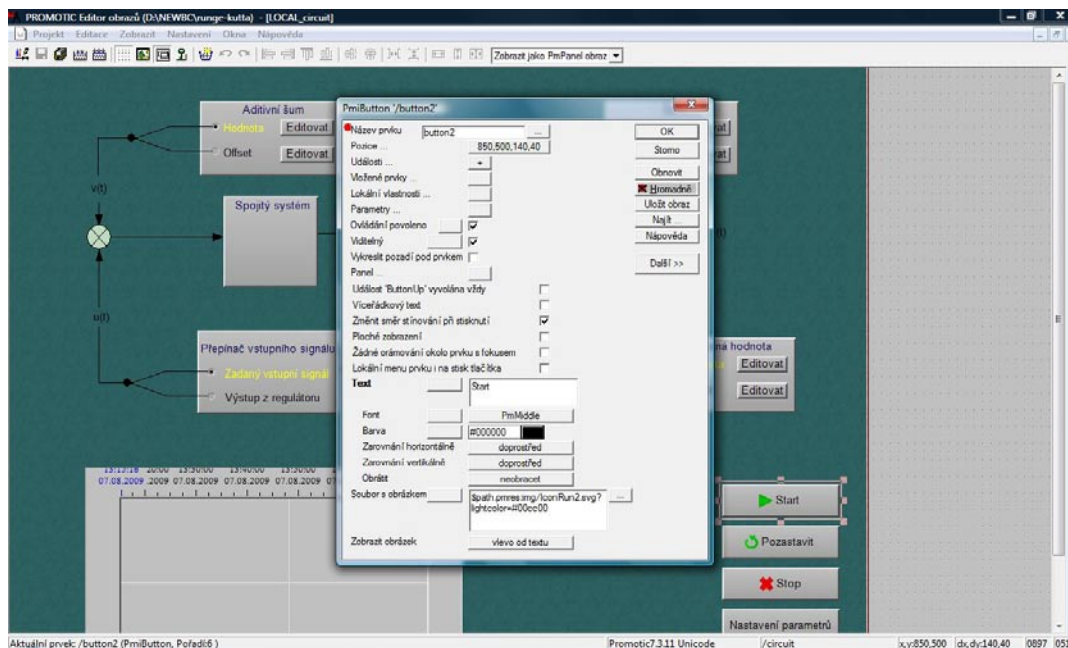
První část se nazývá editor aplikace (Obr. č. 2), kde vytváříme jednotlivé PROMOTIC objekty do stromové struktury. Tyto objekty reprezentují základní datové typy jako integer, float, string i některé speciální např. timer, trend, panel a další [9]. Níže si popíšeme jen některé důležité objekty, které jsou použity v naší aplikaci. Detailní popis všech objektů najdeme v nápovědě systému PROMOTIC.



Obr. č. 2 Editor aplikace

Druhá část se nazývá editor obrazů (Obr. č. 3) a slouží pro objekt PmPanel k vytváření grafických obrazů aplikace. Ty se skládají z libovolného počtu grafických prvků, které vytvoří projektant podle svých představ výběrem z palety předdefinovaných prvků. V těchto prvcích lze zadávat jejich statické vlastnosti, popř. tyto vlastnosti napojit datovou vazbou a oživit tak vytvořenou vizualizaci. Napojit datovou vazbou¹ lze všechny důležité vlastnosti grafických prvků: barva, poloha, velikost, hodnota a mnoho dalších.

¹ Datová vazba umožňuje spojení, kdy např. čtení z proměnné způsobí přečtení hodnoty na kterou je proměnná vázaná.



Obř. ř. 3 Editor obrazů

Pro zápis uživatelských algoritmů nebo pro přístup k metodám a vlastnostem objektů slouží zabudovaný jazyk VBScript se syntaxí Visual Basic [9].

3.2.1 Použité PROMOTIC objekty

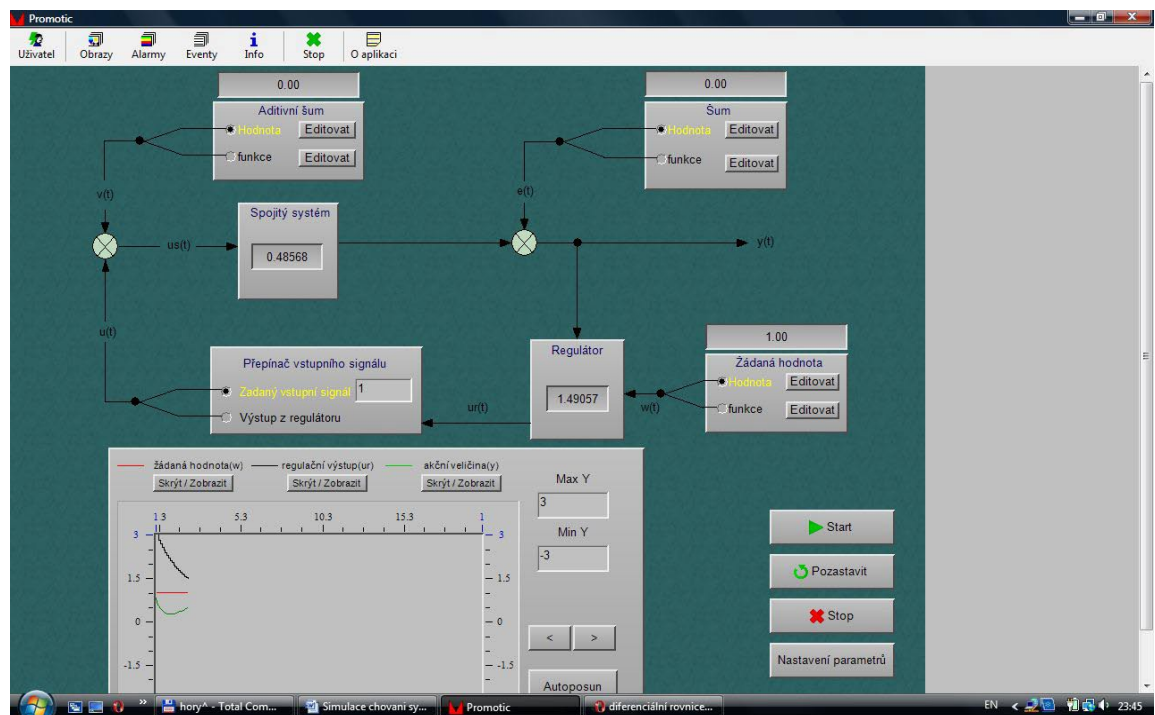
- PmToolbar a PmWorkspace – základní objekty, které jsou automaticky vytvořeny s každou novou aplikací. PmToolbar je nástrojová lišta, na kterou lze umístit tlačítka s volitelně definovanými funkcemi. PmWorkspace je pracovní plocha, na kterou se umísřují další součásti uživatelského rozhraní (vřetně PmToolbar)
- PmFolder – jde o složku, do které se mohou umísřovat další PROMOTIC objekty. Výhodou je pak stromová struktura aplikace, což vede k větří přehlednosti a lepší orientaci
- PmTimer – objektu se nastaví perioda a ten pak hlídá její uplynutí. S danou periodou se spustí metoda TimerTick, která obsahuje uživatelem definovaný VBScript
- PmData – slouží k definici libovolného počtu proměnných různých datových typů v jednom objektu
- PmDataTable – data jsou rozdělena do řádků a sloupců. Jedná se o obdobu databázové tabulky. Velká výhoda spočívá v přidávání i odebírání jak

řádků, tak sloupců a to přímo za běhu aplikace. Její užití se nabízí v případech, kdy při inicializaci neznáme přesný počet hodnot

- PmPanel – vizualizační objekt, jehož obsah se tvoří v editoru obrazů. PROMOTIC obsahuje množství předdefinovaných vizualizačních objektů a možnost jejich napojení na data v aplikaci pomocí datových vazeb.
- ActiveX – z ActiveX objektů byl použit TrendsView, který slouží pro grafické zobrazení dat

3.3 Uživatelské rozhraní aplikace

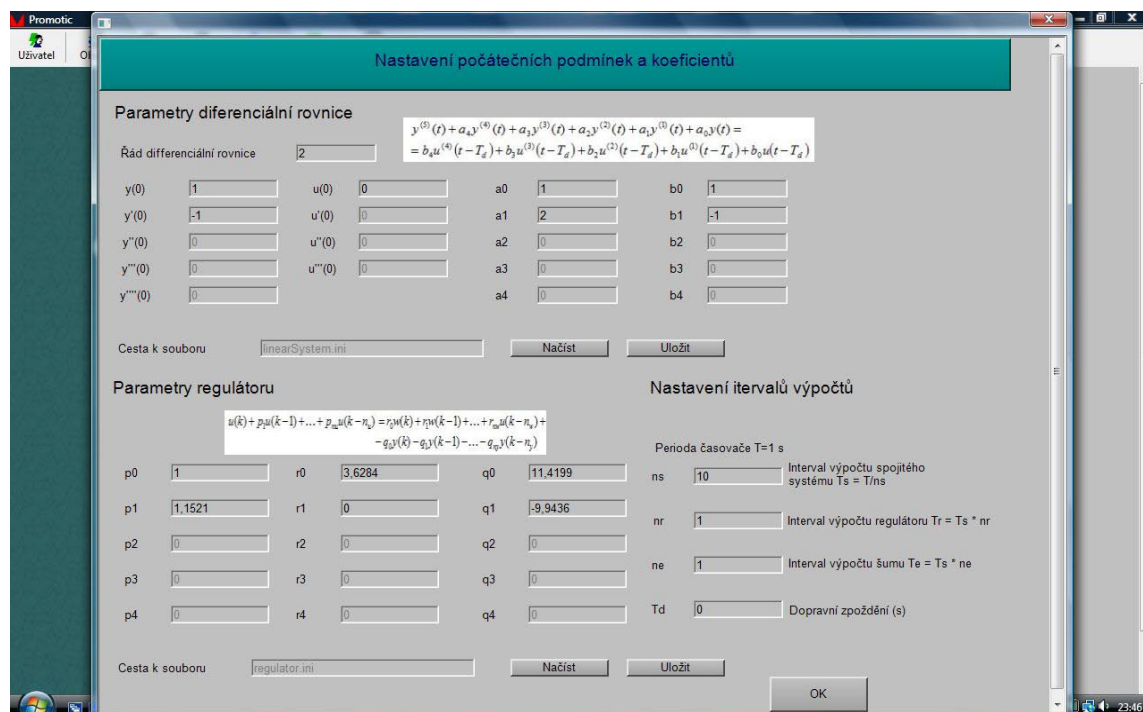
Po spuštění aplikace máme před sebou blokové schéma (Obr. č. 4) spojitého systému, abychom získali základní představu o jeho zapojení. Také zde již můžeme nastavovat hodnoty šumu, poruchy, žádané hodnoty a vybrat si, zda regulátor bude působit zpět na soustavu nebo se bude generovat zadaný vstupní signál. Všechny výše popsané možnosti je možné měnit za běhu aplikace.



Obr. č. 4 Hlavní obrazovka aplikace

Další nastavení se skrývá pod tlačítkem „nastavení parametrů“ (Obr. č. 5), kde zadáme údaje o našem spojitém systému, regulátoru, jednotlivé intervaly a časování výpočtů. U spojitého systému jsou to řád diferenciální rovnice, její koeficienty a

počáteční podmínky. Regulátor pak definujeme pomocí jeho parametrů p , q a r . Nastavené parametry regulátoru a spojitého systému si můžeme uložit do vlastně pojmenovaného souboru pro pozdější použití.



Obř. ř. 5 Modální okno s nastavením parametrů

3.3.1 Aplikační logika

Abychom zajistili správný běh aplikace, je nutné zabránit uživateli v provádění určitých akcí za běhu programu nebo mu umožnit zadávat hodnoty jen v určitých mezích.

U nastavování diferenciální rovnice je její řád omezen na rozsah 1-5 celočíselně. U parametrů pro intervaly časování jsou to pak celočíselné hodnoty 1-10. Kontrolu pro rozsah hodnot nebo pro správný datový typ nabízí přímo objekt PmWEdit (editační pole), takže zvětšení rozsahu nebo změny kontroly datového typu jde kdykoliv snadno upravit.

Další omezení se vztahuje na ovládací tlačítka (start, stop, pozastavit/pokračovat, nastavení parametrů) aplikace a je realizováno pomocí několika jednoduchých skriptů. Před spuštěním aplikace je aktivní jen tlačítko „start“ a „nastavení parametrů“, kterým je možno vyvolat modální okno s nastavením. Po stisknutí tlačítka „start“ se ale stává

neaktivním a nelze na něj kliknout. Naopak se aktivují tlačítka „stop“ a „pozastavit/pokračovat“. Po stisknutí tlačítka stop se nastaví všechny potřebné proměnné i tlačítka do počátečního stavu. Významy jednotlivých tlačítek jsou pochopitelné z jejich názvů, proto jejich popis vynecháme.

3.4 Formát souborů s konfigurací

Pro ukládání nastavení byl vybrán formát INI souborů systému Windows, který se skládá ze sekcí a klíčů. Název sekce v *.ini souboru je vždy uveden v hranatých závorkách. Každá sekce potom obsahuje několik klíčů (klíč musí být vždy přiřazen k sekci, která je uložena v souboru před tímto klíčem). Obsah jedné sekce končí označením další sekce. Název klíče je vždy na samostatném řádku a znakem = má přiřazen obsah [9].

Uživatel má možnost změny umístění a názvu souboru pomocí metody `Pm.inputBoxForFilePath`, která vyvolá okno s prohlížením obsahu na disku. Soubory s konfigurací jsou dva.

První soubor obsahuje údaje o diferenciální rovnici a parametry pro výpočet intervalu časování pro diferenciální rovnici a aditivního šum (Obr. č. 6).

```
[order]
value=2
[y0 - y1]
y0=1
y1=-1
[a0 - a1]
a0=1
a1=2
[b0 - b1]
b0=1
b1=-1
[u0 - u0]
u0=0
[Intervaly]
ns=10
ne=1
```

Obr. č. 6 Konfigurační soubor s parametry diferenciální rovnice

Druhý soubor pak obsahuje údaje o regulátoru včetně parametru pro výpočet jeho časování a řád diferenciální rovnice, pro kterou byl navržen (Obr. č. 7).

```
[order]
value=2
[p0 - p1]
p0=1
p1=1,1521
[q0 - q1]
q0=11,4199
q1=-9,9436
[r0 - r1]
r0=3,6284
r1=0
[Interval]
nr=1
```

Obr. č. 7 Konfigurační soubor s parametry regulátoru

Hlavní výhody formátu jsou dobrá podpora ze strany PROMOTICu, kde jsou k dispozici metody pro jeho manipulaci, ale i přehledný formát pro uživatele, který má možnost soubor snadno editovat z jakéhokoliv textového editoru.

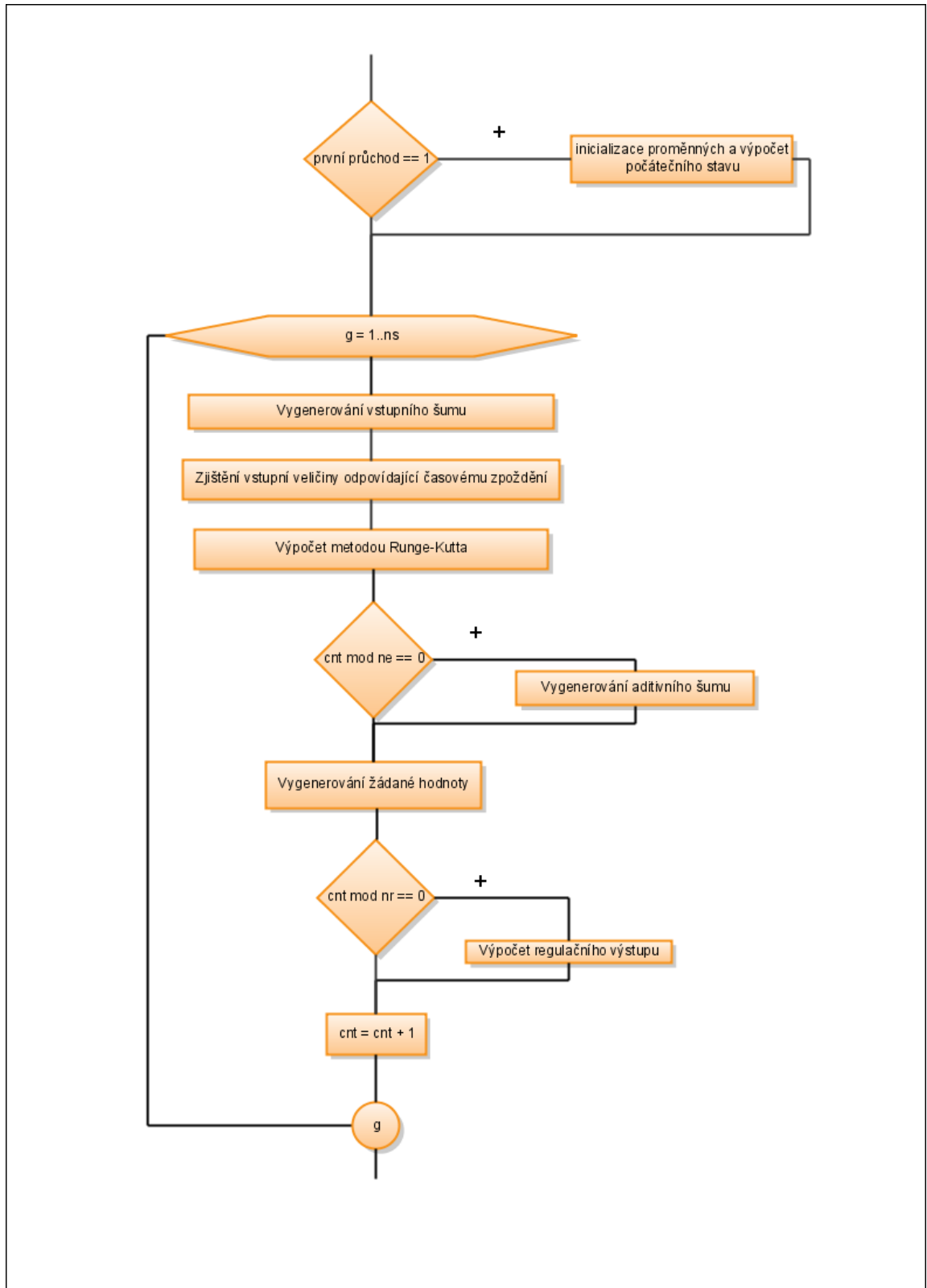
3.5 Algoritmy v PROMOTICu

3.5.1 Jazyk VBScript

Jazyk, v němž se píše algoritmy v systému PROMOTIC je Visual Basic (přesněji jeho skriptové vydání), často označovaný jako VBScript. Jedná se o jednoduchý jazyk, který se dá snadno naučit a přitom je výkonný a optimalizovaný. Kvůli bezpečnosti nemůže přistupovat k paměti počítače ani vytvářet nebo pracovat se soubory na pevném disku. Ve VBScriptu však můžeme používat objekty. Lze tak používat i speciální objekt Pm, který obsahuje např. metody pro přístup na disk. Takto se nám rozšiřuje jazyk VBScriptu za jeho hranice [9].

3.5.2 Implementace výpočtů a časování

Pro lepší představu je přiložena část vývojového diagramu (Obr. č. 8), který zjednodušeně popisuje algoritmus napsaný v jazyce VBScript, volaného v události TimerTick objektu PmTimer (časovač).



Obr. č. 8 Vývojový diagram časování obvodu

Podle nastaveného řádu diferenciální rovnice se pomocí konstrukce switch provede příslušný algoritmus pro výpočet diferenciální rovnice metodou Runge-Kutta.

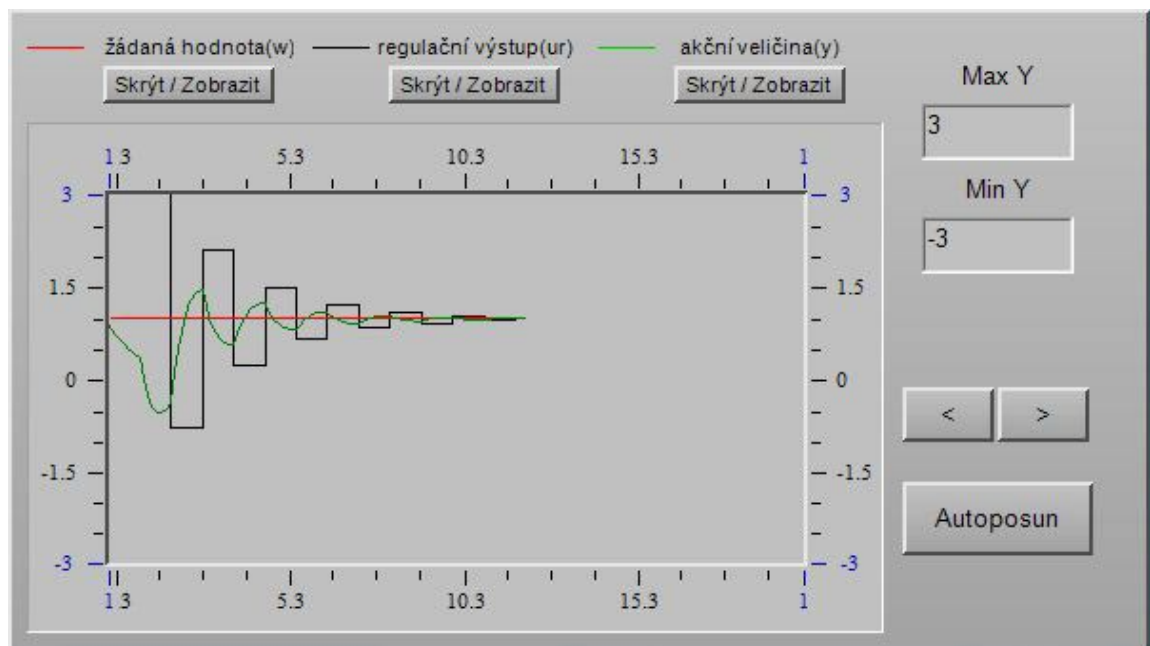
Nejprve se zjistí, zda se jedná o začátek aplikace a pokud ano, provede se inicializace proměnných a výpočet prvních hodnot z počátečních podmínek. Dále následuje cyklus, který provádí výpočet spojitého systému. Počet výpočtů je určen podle parametru n_s . Další výpočty se provádějí uvnitř tohoto cyklu, protože intervaly jejich výpočtů jsou násobky intervalu výpočtu diferenciální rovnice (spojitý systém). Určíme, zda máme v daném cyklu vygenerovat hodnotu aditivního šumu. Určení provedeme pomocí operátoru zbytek po celočíselném dělení (mod) celkového počtu (cnt) výpočtů diferenciální rovnice a parametru n_e . Pokud se zbytek po dělení rovná nule, provedeme vygenerování příslušné hodnoty. Stejný postup použijeme i v případě výpočtu regulačního výstupu, pouze namísto parametru n_e použijeme parametr n_r . Ostatní výpočty jsou shodné s intervalem výpočtu diferenciální rovnice, proto se provedou při každém průchodu cyklu.

Aktuální čas t lze vypočítat jako podíl celkového počtu výpočtů diferenciální rovnice cnt a parametru n_s .

3.6 Grafické znázornění průběhů veličin

Pro znázornění průběhů veličin byl použit ActiveX objekt TrendsView. V PROMOTICu je možné přes trendy jednoduše graficky znázornit hodnotu zobrazované veličiny v aktuálním čase (Obr. č. 9). Zde jsem ovšem narazil na problém uchování času s přesností větší než 1 s. Z tohoto důvodu byl pro zobrazování dat použit vlastní skript, který v objektu TrendsView místo časové osy používá osu numerickou (reálná čísla s desetinnou částí), která představuje čas v sekundách od začátku běhu programu.

Graf zobrazuje grafické průběhy 3 základních veličin, kterými jsou akční veličina y , regulační výstup u_r a žádaná hodnota w . Jelikož veličiny mohou mít dosti odlišné hodnoty, je možné zadat rozsah osy y . U grafu je také možnost vypnutí automatického posunu, procházení do historie nebo vypnout zobrazení zvolené veličiny.



Obr. č. 9 Graf zachycující průběh základních veličin

4. Závěr

Cílem práce bylo v systému PROMOTIC vytvořit aplikaci, která simuluje chování zadaného systému v reálném čase. Stěžejní bylo nastudování řešení diferenciálních rovnic pomocí metody Runge-Kutta 4. řádu. Dále pak rozklad zadané rovnice do diskrétního výpočetního tvaru tak, aby šel realizovat v jazyce VBScript a nastudovat potřebné funkce a vlastnosti systému PROMOTIC. Výsledkem je aplikace, která odpovídá základním požadavkům na funkčnost.

PROMOTIC se ukázal jako vhodný nástroj pro tvorbu zadané práce pro jeho zabudované funkce jako je např. napojení na datovou vazbu objektu, která se automaticky postará o aktualizaci hodnot, které jsou na ni napojeny. Zobrazování hodnot a grafů je díky editoru obrazů, kde máme velké množství předpřipravených objektů, jednoduché a rychlé. Pokud bychom měli stejnou aplikaci navrhnout a realizovat v některém běžném programovacím jazyce jako je např. Java, stálo by to nespočetně více úsilí a hodin práce, než bychom vyřešili jen tyto základní věci.

Jako u každé aplikace i zde bychom našli mnoho způsobů jak ji vylepšit, např. zobecnění výpočtu pro rovnice vyšších řádů, ale kvůli časovým důvodům nebylo již toto možné implementovat do výsledné aplikace.

5. Literatura

1. KUBÍK, Stanislav, KOTEK, Zdeněk, ŠALAMON, Miroslav. Teorie regulace : I. Lineární regulace. Praha : SNTL-NAKLADATELSTVÍ TECHNICKÉ LITERATURY, 1974. 272 s. ISBN 04-518-74.
2. www.337.vsb.cz/materialy/experimentalni_mechanika/DYNAM_SYST.pdf
3. http://cs.wikipedia.org/wiki/Diferenciální_rovnice
4. OGATA, Katsuhiko. Modern control Engineering. 2. japonský. [s.l.], PRENTOCE HALL, 1990. 963 s. ISBN 0-13-589128.
5. http://www.kvd.zcu.cz/cz/materialy/numet/_numet.html#_Toc501178900
6. <http://www.fce.vutbr.cz/studium/materialy/Dynsys/kap7/kap7.htm>
7. VITÁSEK, Emil. Základy teorie numerických metod pro řešení diferenciálních rovnic. Praha : Academia, 1994. 408 s. ISBN 80-200-0281-2.
8. <http://www.automatizace.cz/article.php?a=1193>
9. <http://www.promotic.eu/cz/pmdoc/WhatIsPromotic/WhatIsPromotic.htm>

Údaje pro knihovnickou databázi

Název práce	Simulace chování systému v reálném čase
Autor práce	Martin Horák
Obor	Informační technologie
Rok obhajoby	2009
Vedoucí práce	doc. Ing. František Dušek CSc.
Anotace	Tato bakalářská práce se zabývá vytvořením aplikace v SCADA / HMI nástroji PROMOTIC, která simuluje dynamické chování zadaného systému v reálném čase. Simulovaný systém představuje uzavřený regulační obvod se soustavou popsanou lineární diferenciální rovnicí max. 5. řádu a s diskrétním regulátorem se dvěma stupni volnosti popsaným diferenční rovnicí max. 5. řádu.
Klíčová slova	PROMOTIC, diferenciální rovnice, Runge-Kutta, simulace, regulace