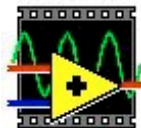


Control for LabVIEW



Radek Votrubec

únor 2000

**Technická univerzita v Liberci
Fakulta mechatroniky a mezioborových inženýrských studií
Katedra řídicí techniky**

1. Úvod

Knihovna Control vznikla seskupením programů a funkcí používaných pro aplikace v oblasti řízení.

2 Popis komponent

2.1 Poly.vi

výpočet charakteristického polynomu matice

symbolicky lze charakteristický polynom vyjádřit jako $f(\lambda): Poly(A) = \det(\lambda E - A)$

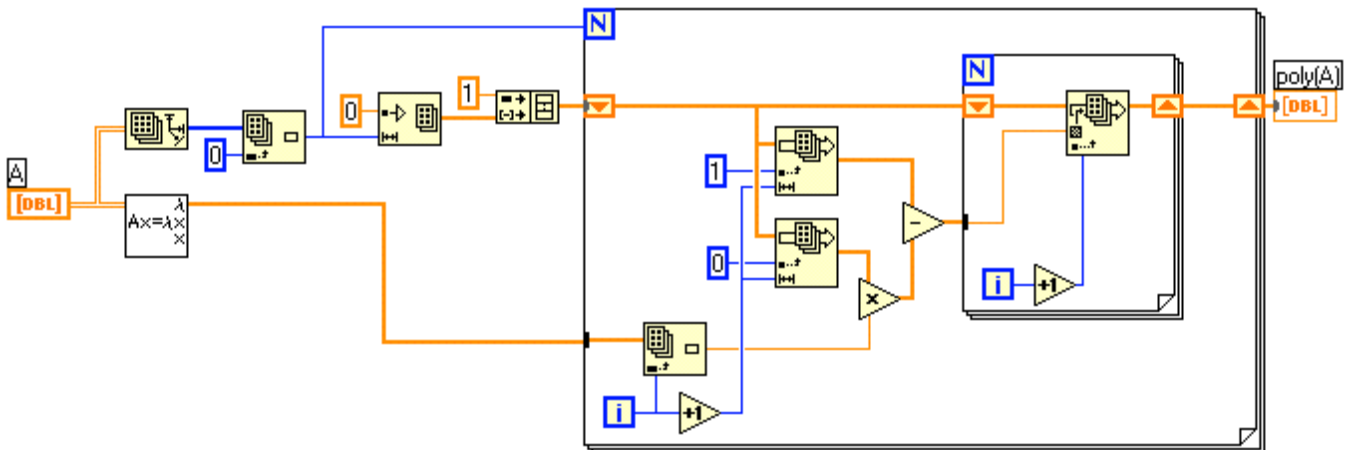
koefficienty charakteristického polynomu lze vypočítat dle algoritmu

A , $(n \times n)$

$poly = [1 \ 0 \ \dots \ 0]$, $(1 \times n+1)$

$poly_{2..j+1} = poly_{2..j+1} - eig(A)_j \cdot poly_{1..j}$, pro $j \in \langle 1; n \rangle$

POLY
[$\lambda E - A$]



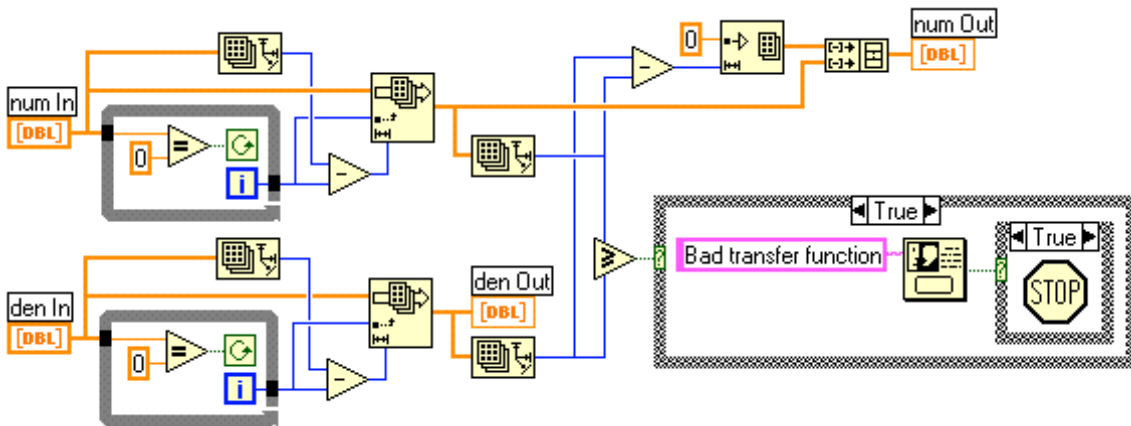
2.2 IsF.vi

?F(s)

test obrazového přenosu

obrazový přenos $F(s) = \frac{b_0 + b_1s + \dots + b_ms^m}{a_0 + a_1s + \dots + a_ns^n}$ je zadán koeficienty čitatele a jmenovatele

nejprve se odstraní nulové počáteční koeficienty, následuje test fyzikální realizovatelnosti tj. není-li řád čitatele menší než řád jmenovatele, nahlásí chybu „Bad Transfer Function“ a ukončí běh programu, jinak vyrovná počet prvků čitatele a jmenovatele tak, že do čitatele na počáteční pozice doplní nuly, což je výhodné pro další použití přenosu



2.3 tf2ss.vi

num	A
den	D

převod obrazového přenosu na stavový popis

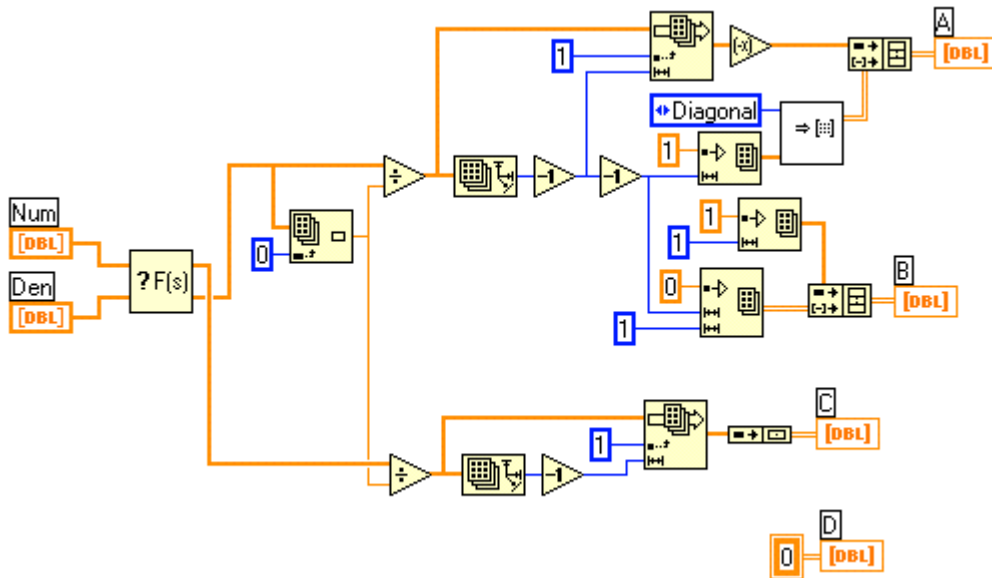
obrazový přenos $F(s) = \frac{b_0 + b_1s + \dots + b_ms^m}{a_0 + a_1s + \dots + a_ns^n}$ nejprve testuje IsF.vi, následuje vydělení

přenosu koeficientem a_0 a sestavení matic stavového popisu $x' = Ax + Bu$
 $y = Cx + Du$

$$A = \begin{bmatrix} -a_1 & -a_2 & -a_3 & \dots & -a_{n-1} & -a_n \\ 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$C = [b_1 \quad b_2 \quad b_b \quad \dots \quad b_{n-1} \quad b_n] \quad D = [0]$$

pro další použití je výhodné, aby všechny matice byly dvourozměrné



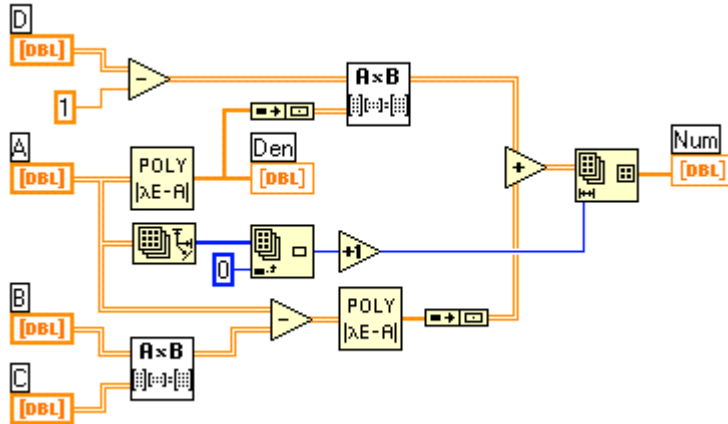
2.4 ss2tf.vi

A	num
B	C
C	den
D	

převod stavového popisu na obrazový přenos

jmenovatel $den = poly(A)$

čítatel $num = poly(A - B \times C) + (D - 1) \cdot den$



2.5 c2d.vi

n(s)	n(z)
d(s)	d(z)
Ts	

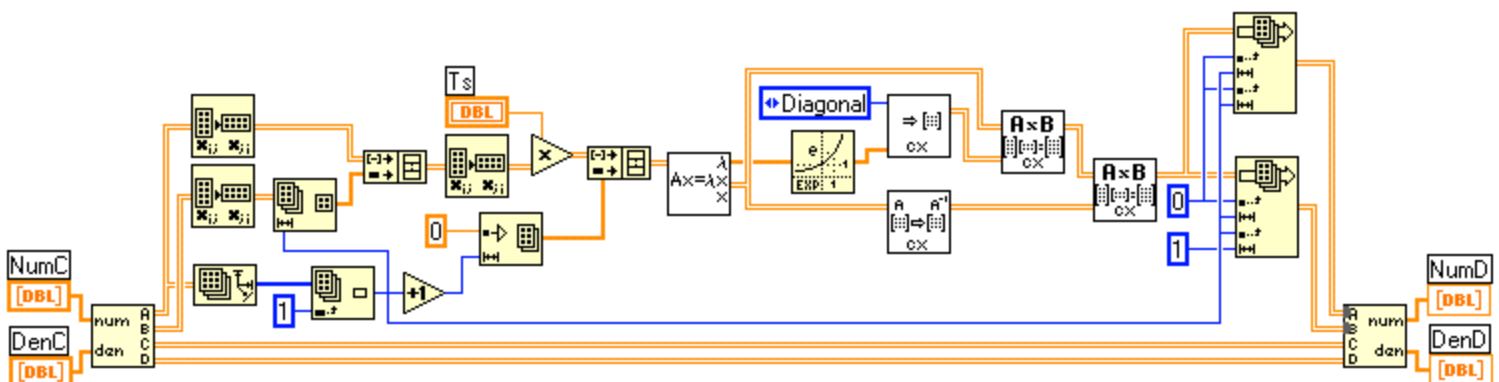
převod spojitého popisu na diskretní

obrazový přenos $F(s) = \frac{b_0 + b_1s + \dots + b_ms^m}{a_0 + a_1s + \dots + a_ns^n}$ nejprve převede na stavový popis,

vytvoří matici $TEMP = e^{\begin{bmatrix} A_c & B_c]Ts \\ 0 & \dots & 0 \end{bmatrix}}$

diskretní stavový popis je pak $A_d = TEMP_{0..n-1,0..n-1}$ $B_d = TEMP_{0..n-1,n}$ $C_d = C_c$ $D_d = D_c$

ten převede na diskretní popis $\frac{b_{d1}z^{-1} + \dots + b_{dm}z^{-m}}{1 + a_{d1}z^{-1} + \dots + a_{dn}z^{-n}}, Ts$



2.6 d2c.vi

$n(z)$
 $d(z)$
 T_s

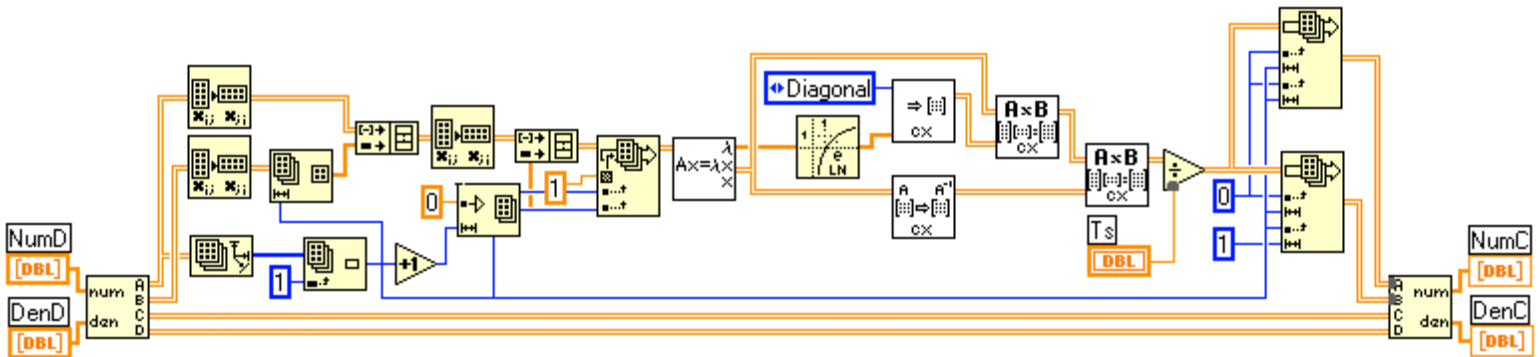
převod diskrétního popisu na spojitý

diskrétní popis $\frac{b_{d1}z^{-1} + \dots + b_{dm}z^{-m}}{1 + a_{d1}z^{-1} + \dots + a_{dm}z^{-n}}$ převede na diskrétní stavový popis

vytvoří matici $TEMP = \log_2 \begin{bmatrix} A_d & B_d/T_s \\ 0 & \dots & 0 \end{bmatrix}$ a vezme pouze reálnou část

spojitý stavový popis je pak $A_c = TEMP_{0..n-1,0..n-1}$ $B_c = TEMP_{0..n-1,n}$ $C_c = C_d$ $D_c = D_d$

a ten převede na obrazový přenos $F(s) = \frac{b_1s + \dots + b_ms^m}{1 + a_1s + \dots + a_ns^n}$



2.7 Hurwitz.vi

$\begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} > 0$
Hurwitz

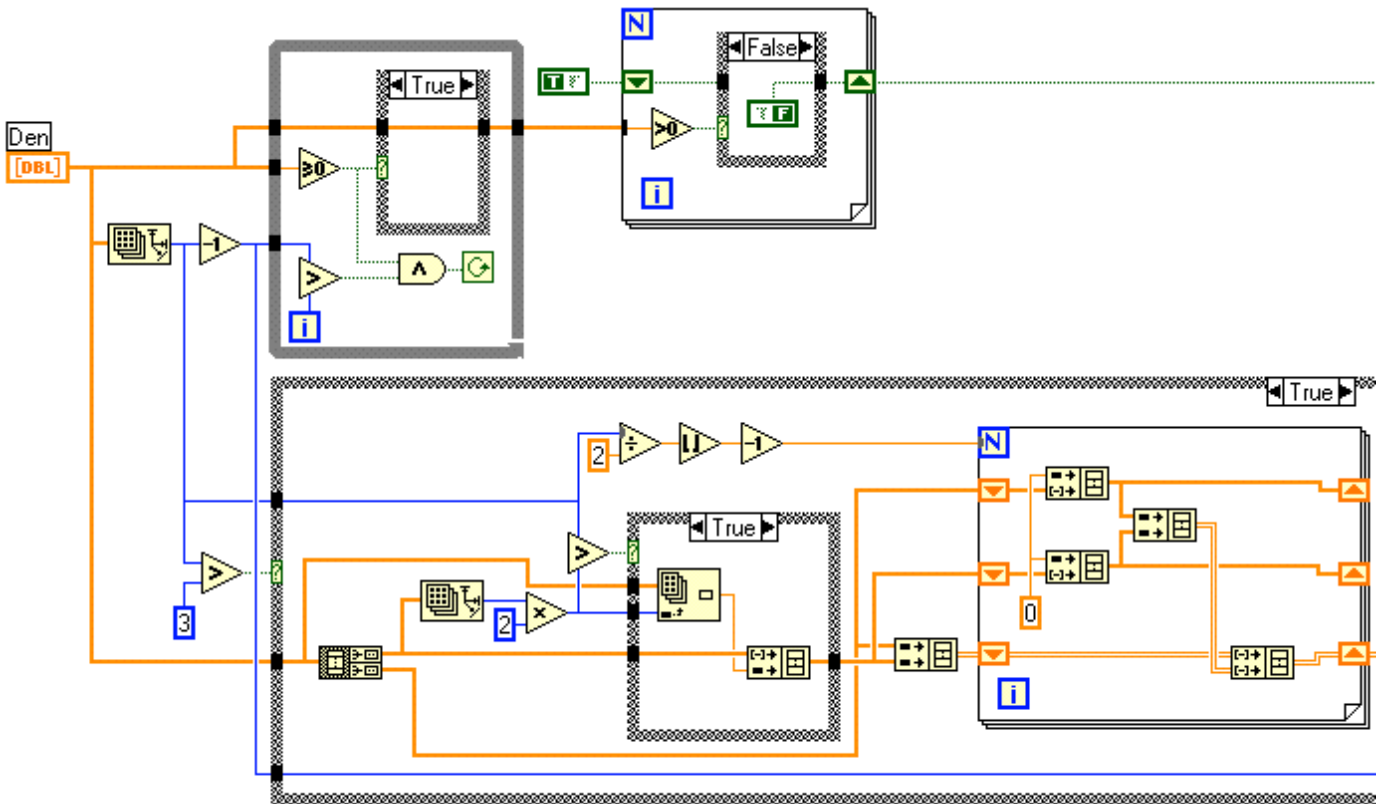
Hurwitzovo kritérium stability

na základě jmenovatele obrazového přenosu rozhoduje o jeho stabilitě

testuje, zda všechny koeficienty mají stejná znaménka a jsou v absolutní hodnotě kladné

pro $n > 3$ sestaví Hurwitzovu matici $H = \begin{bmatrix} a_1 & a_3 & a_5 & \dots & a_n & 0 & \dots & 0 \\ a_0 & a_2 & a_4 & \dots & a_{n-1} & 0 & \dots & 0 \\ 0 & a_1 & a_3 & \dots & a_{n-2} & a_n & \dots & 0 \\ 0 & a_0 & a_2 & \dots & a_{n-3} & a_{n-1} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & a_{n-4} & a_{n-2} & \dots & 0 \\ 0 & \dots & 0 & a_0 & \dots & a_{n-3} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & a_n \end{bmatrix}$

a testuje, zda subdeterminanty $H_{0..j,0..j}$ pro $j = 1..n-1$ jsou kladné

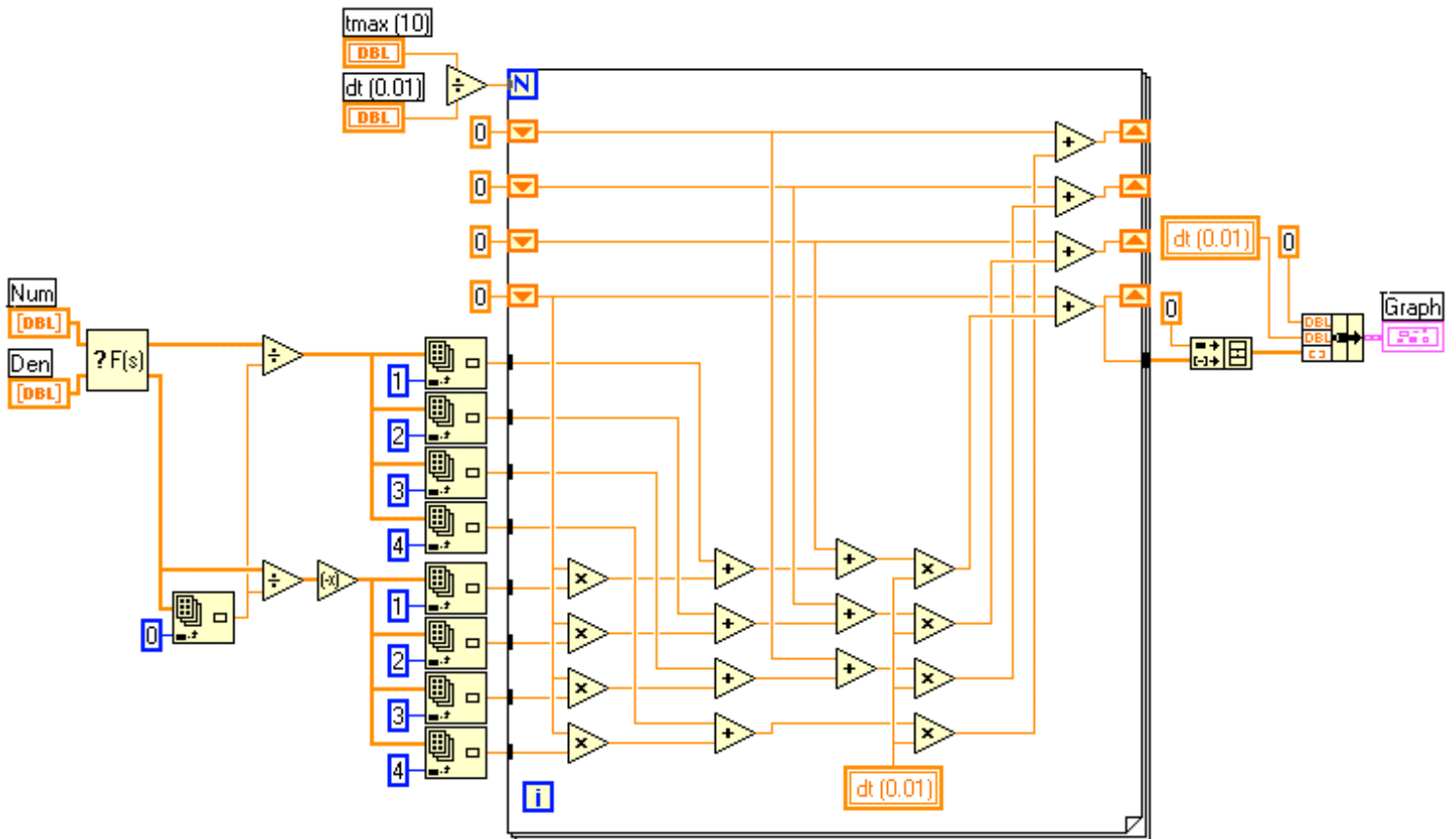


2.8 SimC.vi



spojitá simulace přechodové funkce (vstupní veličina obrazového přenosu = 1)
numericky Eulerovou metodou řeší diferenciální rovnici s $u=1$

$$a_0 y^{(4)} + a_1 y''' + a_2 y'' + a_3 y' + a_4 y = b_1 u''' + b_2 u'' + b_3 u' + b_4 u, \quad p.p. = 0$$



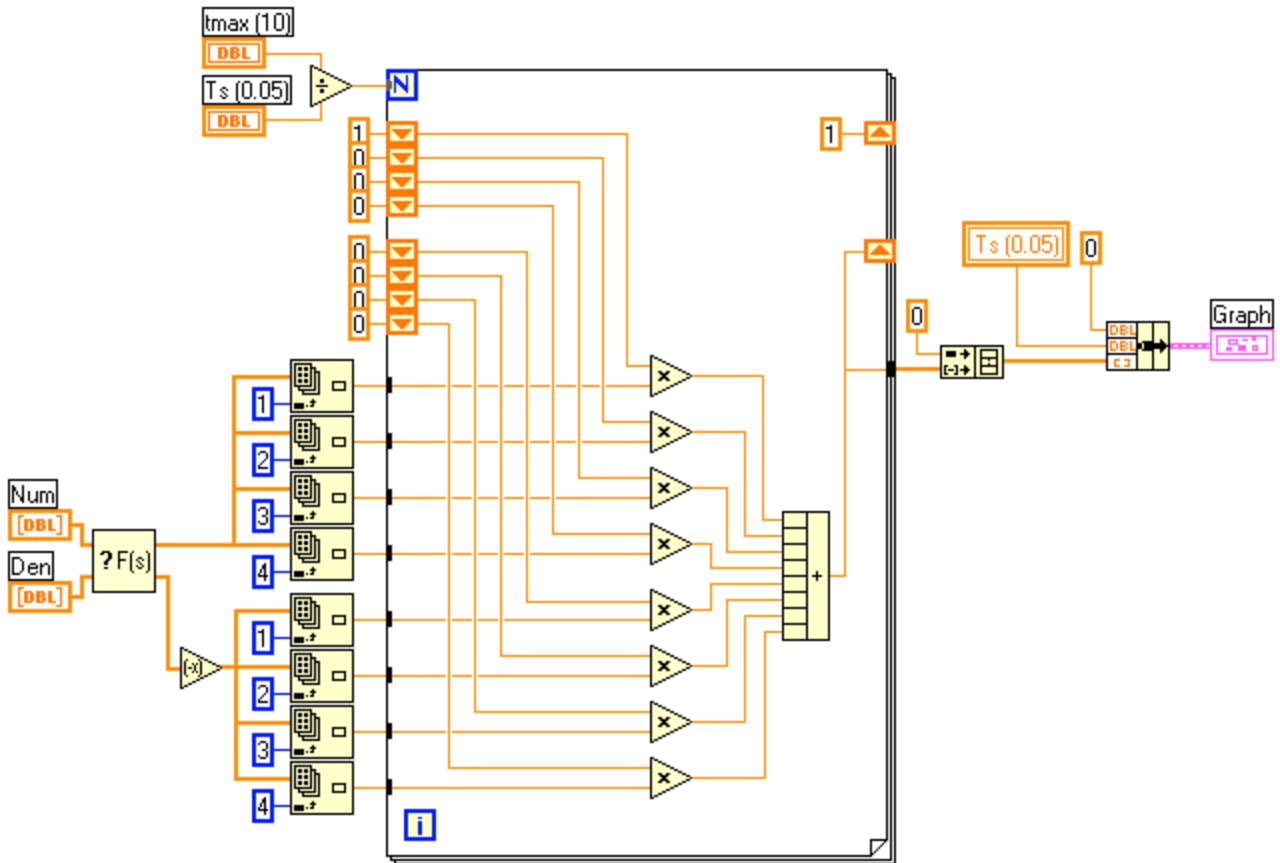
2.9 SimD.vi



diskrétní simulace přechodové funkce (vstupní veličina = 1)

řeší diferenční rovnici pro $u=1$

$$a_0 y(n) + a_1 y(n-1) + a_2 y(n-2) + a_3 y(n-3) + a_4 y(n-4) = b_1 u(n-1) + b_2 u(n-2) + b_3 u(n-3) + b_4 u(n-4)$$



2.10 Simulation.vi

ukázka simulačních metod na soustavě čtvrtého řádu

základní sekvence

- 0: příprava koeficientů
- 1: analytické řešení (pouze pro 4.ř s jednoduchými kořeny)
- 2: spojitá simulace Eulerovou metodou
- 3: diskretní simulace
- 4: spojitá simulace Eulerovou metodou ve stavovém prostoru
- 5: diskretní simulace ve stavovém prostoru
- 6: spojitá simulace metodou Runge-Kutta 4.ř

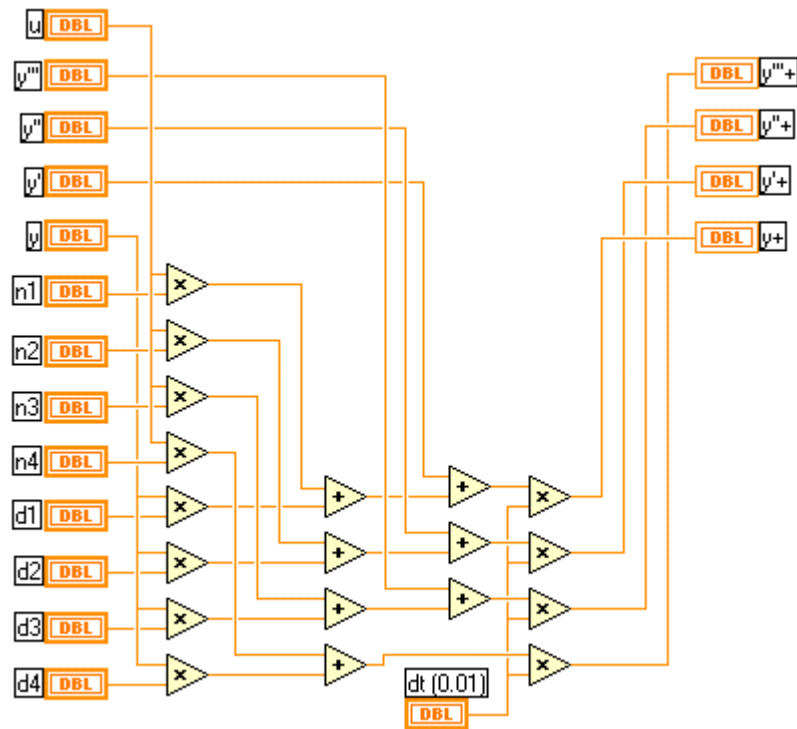
aplikace umožňuje porovnání přesností a rychlostí výpočtu uvedených metod

2.11 SimCuDR.vi



spojitá simulace diferenciální rovnice
 numericky Eulerovou metodou řeší diferenciální rovnici

$$a_0 y^{(4)} + a_1 y''' + a_2 y'' + a_3 y' + a_4 y = b_1 u''' + b_2 u'' + b_3 u' + b_4 u, \quad p.p. = 0$$

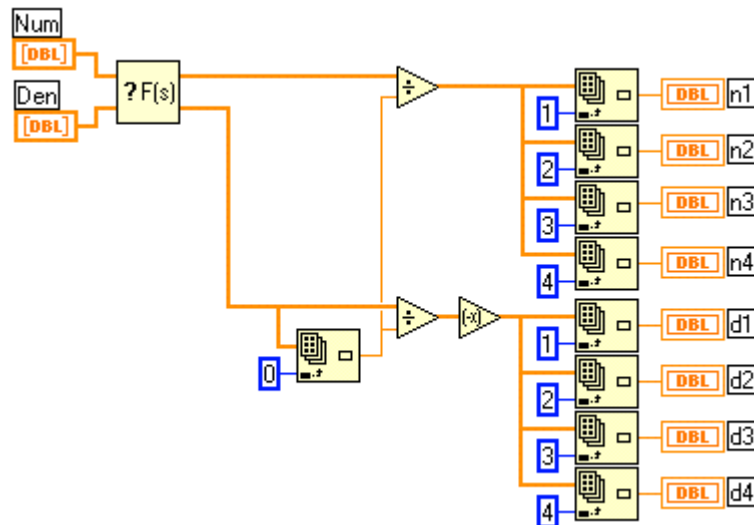


2.12 SimCuAB.vi



koeficienty pro spojitou simulaci diferenciální rovnice
 koeficienty diferenciální rovnice

$$a_0 y^{(4)} + a_1 y''' + a_2 y'' + a_3 y' + a_4 y = b_1 u''' + b_2 u'' + b_3 u' + b_4 u, \quad p.p. = 0$$

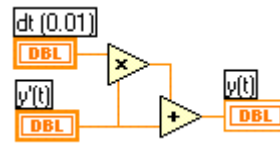


2.13 Euler.vi



výpočet přírůstku při Eulerově metodě

$$y(n+1) = y(n) + y' \cdot dt$$



2.14 RK4.vi



výpočet přírůstku při metodě Runge-Kutta 4. řádu

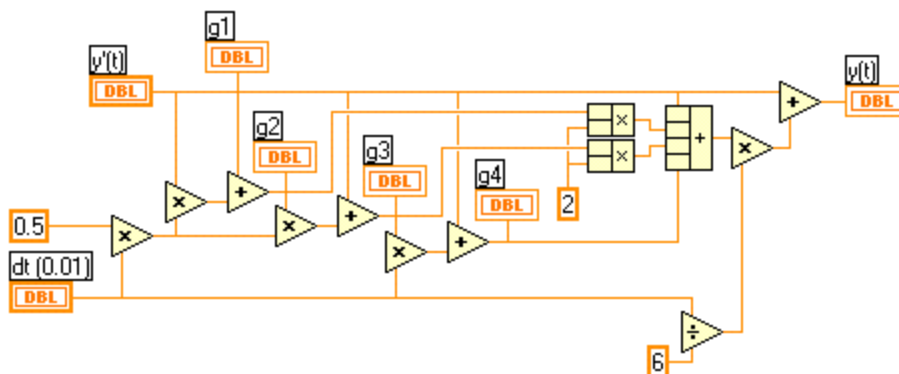
$$y(n+1) = y(n) + \frac{1}{6}(g_1 + 2g_2 + 2g_3 + g_4)dt$$

$$g_1 = f(x(n), u(n))$$

$$g_2 = f(x(n) + 0.5 \cdot dt \cdot g_1, u(n) + 0.5 \cdot dt)$$

$$g_3 = f(x(n) + 0.5 \cdot dt \cdot g_2, u(n) + 0.5 \cdot dt)$$

$$g_4 = f(x(n) + dt \cdot g_3, u(n) + dt)$$

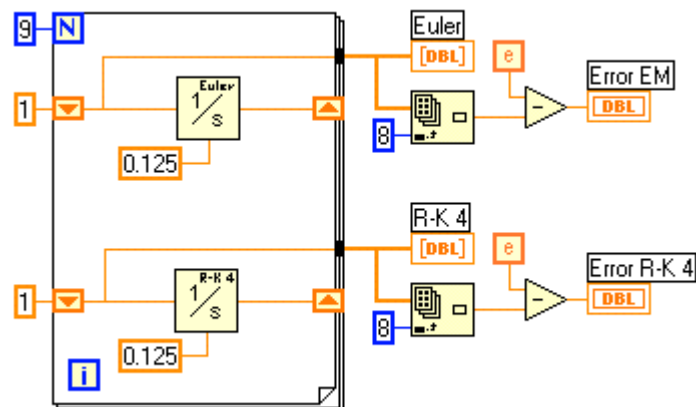


2.15 eApprox.vi

ukázka použití Euler.vi a RK4.vi na aproximaci Eulerova čísla

$$y' = y, y(0) = 1 \text{ s } dt = 0.125$$

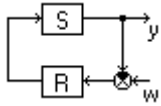
Euler	R-K 4
1.000000	1.000000
1.125000	1.133148
1.265625	1.284025
1.423828	1.454990
1.601807	1.648720
1.802032	1.868244
2.027287	2.116997
2.280697	2.398871
2.565785	2.718277
Error EM	Error R-K 4
0.152497	0.000005



2.16 PID.vi



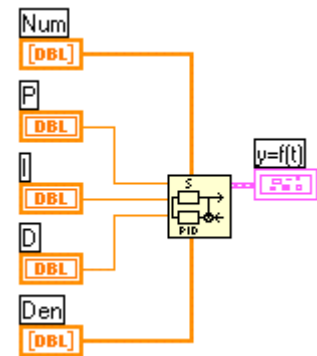
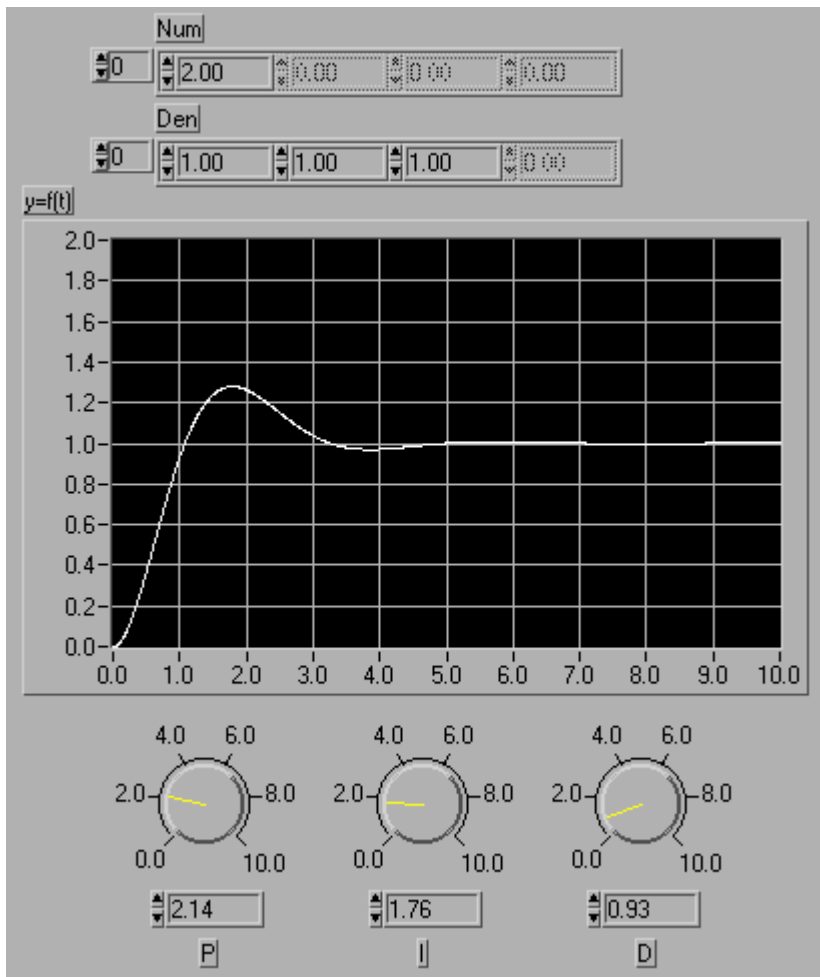
simulace PID regulátoru v záporné zpětné vazbě
aplikace umožňuje nastavit PID regulátor na žádanou hodnotu dané soustavy



$$u = P \cdot e + I \cdot \int e \cdot dt + D \frac{de}{dt}$$

2.17 PIDctrl.vi

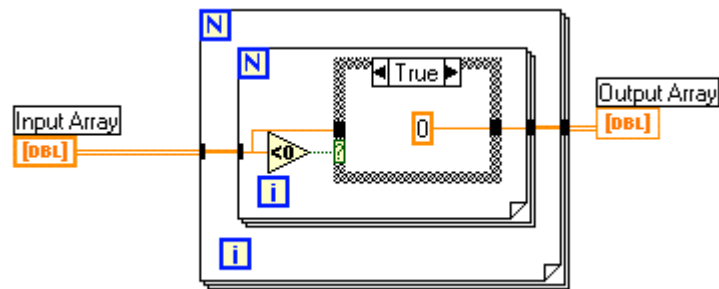
demonstrační aplikace PID.vi



2.18 NegToZero.vi



záporné prvky v matici nahradí nulami

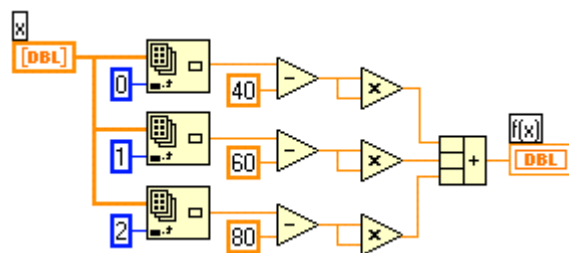


2.19 Simplexf.vi



výpočet funkce více proměnných

příkladem funkce je $f(x) = (x_1 - 40)^2 + (x_2 - 60)^2 + (x_3 - 80)^2$



2.20 Simplex.vi



simplexová metoda hledání minima funkce

Princip metody spočívá v určení směru hledání v n -dimenzionálním prostoru z hodnot funkce v $n+1$ bodech, vrcholech n -rozměrného polyedru, tzv. regulárního simplexu.

Pro vlastní použití se do souboru Simplexf.vi naprogramuje libovolná funkce

$$\text{počáteční simplex: } S = \begin{bmatrix} x_1 & \cdots & x_{n+1} \end{bmatrix} = \begin{bmatrix} 0 & k_1 & k_2 & k_2 & \cdots & k_2 \\ 0 & k_2 & k_1 & k_2 & \cdots & k_2 \\ 0 & k_2 & k_2 & k_1 & \cdots & k_2 \\ \vdots & \vdots & \vdots & \vdots & \ddots & k_2 \\ 0 & k_2 & k_2 & k_2 & k_2 & k_1 \end{bmatrix}$$

$$k_1 = \frac{d}{n\sqrt{2}}(\sqrt{n+1} + n - 1) \quad k_2 = \frac{d}{n\sqrt{2}}(\sqrt{n+1} - 1), d = 1$$

algoritmus: ${}_l x = \min(S) \quad {}_h x = \max(S) \quad {}_{hh} x = \max(S - \{{}_h x\})$

$${}_{n+2} x = \frac{1}{n} \left(\sum_{i=1}^{n+1} i x - {}_h x \right)$$

$${}_{n+3} x = {}_{n+2} x + \alpha ({}_{n+2} x - {}_h x), \alpha = 1$$

$$f({}_{n+3} x) < f({}_{hh} x) \Rightarrow f({}_{n+3} x) < f({}_l x) \Rightarrow {}_{n+4} x = {}_{n+2} x + \gamma ({}_{n+3} x - {}_{n+2} x), \gamma = 2$$

$$f({}_{n+4} x) < f({}_l x) \Rightarrow {}_h x \leftrightarrow {}_{n+4} x, E$$

$$\neq \Rightarrow {}_h x \leftrightarrow {}_{n+3} x, R$$

$$\neq \Rightarrow {}_h x \leftrightarrow {}_{n+3} x, R$$

$$\neq \Rightarrow f({}_{n+3} x) < f({}_h x) \Rightarrow f_m = f({}_{n+4} x)$$

$$\neq \Rightarrow f_m = f({}_h x)$$

$$f_m < f({}_{hh} x) \Rightarrow {}_{n+5} x = {}_{n+2} x + \beta ({}_h x - {}_{n+2} x), \beta = 0.5$$

$${}_h x \leftrightarrow {}_{n+5} x, C$$

$$\neq \Rightarrow i x_{k+1} = 0.5(i x_k + {}_l x_k), i \in \langle 1; n+1 \rangle$$

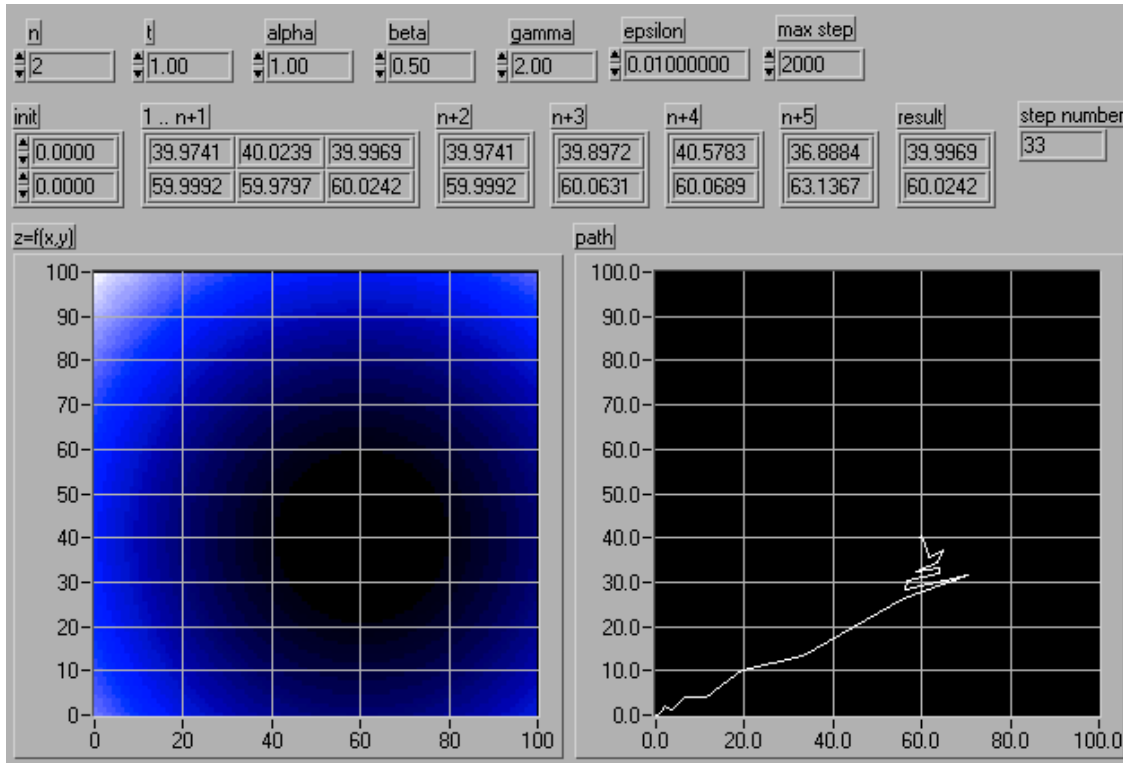
$${}_h x \leftrightarrow 0.5({}_l x + {}_h x), S$$

ukončující podmínka: $\frac{1}{n+1} \sum_{i=1}^{n+1} [f(i x_k) - f({}_{n+2} x)]^2 \leq \varepsilon^2$

2.21 SimplexD.vi

ukázka hledání minima funkce simplexovou metodou

aplikace je obdobou Simplex.vi, hledá minimum fce $f(x) = (x_1 - 40)^2 + (x_2 - 60)^2$
 navíc obsahuje zobrazení funkce a cesty hledání minima



2.22 Simplex3.vi

ukázka hledání minima funkce simplexovou metodou

aplikace je obdobou funkce Simplex.vi, doplněná o krokování se zobrazováním operací

2.23 Interval.vi

identifikace funkce 1.ř metodou prohledávání intervalu

analytické řešení přechodové funkce 1.ř ($K=1$) $y(t) = \frac{1}{T} e^{-\frac{t}{T}}$

simulace Eulerovou metodou $F(s) = \frac{1}{Ts + 1}$

jelikož hledaným parametrem je časová konstanta, vezme se za výchozí interval $(0; t_{max})$, v deseti rovnoměrně rozložených bodech se vypočítá hodnota kvadratického funkcionálu

$$J = \sum_{i=0}^n (a x_i - x_i)^2, \text{ za střed nový interval se vezme } T \Big|_{i=0..n}^{\min(J_i)} \pm 0.1(T_{\max} - T_{\min})$$

2.24 IdentC.vi

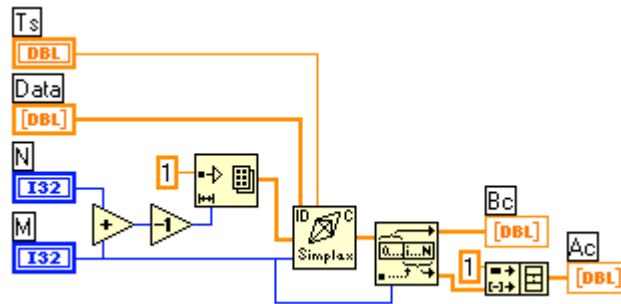


spojitá identifikace soustavy

aplikace identifikuje soustavu na základě vstupních dat simplexovou metodou

hledání minima funkce $J = \sum_{i=0}^n (s x_i - x_i)^2$, využitím IdentSimplexC.vi

m udává řád čitatele, n řád jmenovatele, počet parametrů je tedy $m+n-1$



2.25 IdentSimplexC.vi



podprogram pro spojitou identifikaci soustavy

program je obdobou Simplex.vi, je rozšířen o vstupní data pro minimalizovanou funkci

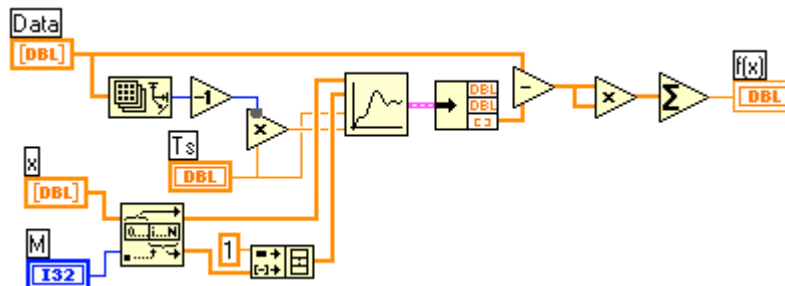
2.26 IdentSimplexC.vi



minimalizovaná funkce pro IdentSimplexC.vi

provede se simulace pro aktuální simplex (SimC.vi), odečte se od identifikovaných dat,

umocní se a sečte, tj. funkce $J = \sum_{i=0}^n (s x_i - x_i)^2$



2.27 IdentD.vi



diskrétní identifikace soustavy
obdoba spojité identifikace, s podprogramem IdentSimplexD.vi

2.28 IdentSimplexD.vi



podprogram pro diskrétní identifikaci soustavy
obdoba spojité identifikace, s minimalizovanou funkcí IdentSimplexD.vi

2.29 IdentSimplexD.vi



minimalizovaná funkce pro IdentSimplexD.vi
obdoba spojité identifikace, simulace SimD.vi

2.30 IdentificationC.vi

spojitá identifikace
 aplikace dovoluje porovnat identifikace přenosy

$$F_{1/2} = \frac{A}{Bs + C}$$

$$F_{1/3} = \frac{A}{Bs^2 + Cs + D}$$

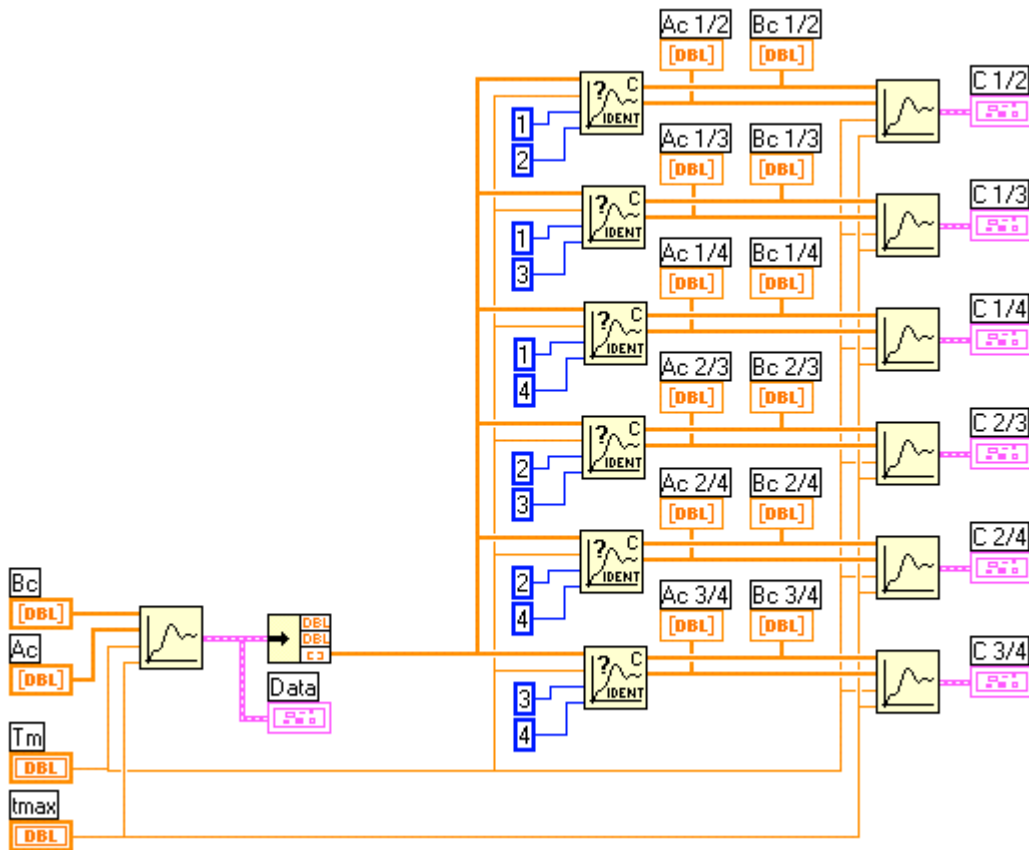
$$F_{1/4} = \frac{A}{Bs^3 + Cs^2 + Ds + E}$$

$$F_{2/3} = \frac{As + B}{Cs^2 + Ds + E}$$

$$F_{2/4} = \frac{As + B}{Cs^3 + Ds^2 + Es + F}$$

$$F_{3/4} = \frac{As^2 + Bs + C}{Ds^3 + Es^2 + Fs + G}$$

a vybrat dostačující řád soustavy



2.31 Identification.vi

spojitá i diskretní identifikace
 obdoba Identification.vi, rozšířeno o diskretní identifikaci
 pozn.: spojitému přenosu $F_{1/3}(s)$ odpovídá diskretní $F_{2/3}(z^{-1})$
 diskretní funkce je velmi proměnlivá a tedy mnohem náchylnější na uvíznutí v lokálním
 minimu

Obsah:

1. Úvod	2
2. Popis komponent	2
2.1 Poly.vi	2
2.2 IsF.vi	3
2.3 tf2ss.vi	4
2.4 ss2tf.vi	5
2.5 c2d.vi	5
2.6 d2c.vi	6
2.7 Hurwitz.vi	6
2.8 SimC.vi	8
2.9 SimD.vi	9
2.10 Simulation.vi	9
2.11 SimCuDR.vi	10
2.12 SimCuAB.vi	10
2.13 Euler.vi	11
2.14 RK4.vi	11
2.15 eApprox.vi	11
2.16 PID.vi	12
2.17 PIDctrl.vi	12
2.18 NegToZero.vi	13
2.19 Simplexf.vi	13
2.20 Simplex.vi	13
2.21 SimplexD.vi	15
2.22 Simplex3.vi	15
2.23 Interval.vi	15
2.24 IdentC.vi	16
2.25 IdentSimplexC.vi	16
2.26 IdentSimplexfC.vi	16
2.27 IdentD.vi	17
2.28 IdentSimplexD.vi	17
2.29 IdentSimplexfD.vi	17
2.30 IdentificationC.vi	18
2.31 Identification.vi	18