

Knihovna pro modelování procesů

TXV 003 44.01
čtvrté vydání
září 2012
změny vyhrazeny

Historie změn

Datum	Vydání	Popis změn
červen 2008	1	První verze
říjen 2008	2	Vygenerována nápověda pro prostředí Mosaic
Prosinec 2010	3	Opravy v popisech bloků
Září 2012	4	Verze knihovny 1.3 – bloky fbSimplePID a fbStepControl

Obsah

1 Knihovna pro modelování procesů.....	3
1.1 Typy.....	3
1.2 Funkční bloky.....	4
1.2.1 fbIntegrator.....	4
1.2.2 fbDerivator.....	6
1.2.3 fbLimIntegrator.....	8
1.2.4 fbDelay.....	10
1.2.5 fbFirstOrder.....	12
1.2.6 fbSecondOrder.....	14
1.2.7 fbSecondOrderOsc.....	16
1.2.8 fbSimplePID.....	19
1.2.9 fbStepControl.....	24
1.2.10 fbGenerator.....	28

1 Knihovna pro modelování procesů

Knihovna ModelLib slouží k modelování a simulaci procesů, které lze popsat diferenciálními rovnicemi.

Knihovna nabízí následující funkční bloky

fbIntegrator	Integrátor
fbDerivator	Derivátor
fbLimIntegrator	Integrátor s omezením
fbDelay	Zpoždění
fbFirstOrder	Filtr prvního řádu
fbSecondOrder	Filtr druhého řádu
fbSecondOrderOsc	Filtr druhého řádu kmitavý
fbSimplePID	Jednoduše ovladatelný PID regulátoru
fbStepControl	Třístavové ovládání servopohonu
fbGenerator	Generátor průběhů

1.1 Typy

Knihovna ModelLib.mlb definuje následující typy proměnných:

<i>Typ</i>	<i>Popis</i>	<i>Základní typ</i>
GenericArrayForDelay	Pole pro uchování vzorků bloku simulujícího dopravní zpoždění	ARRAY[0..1023] OF REAL

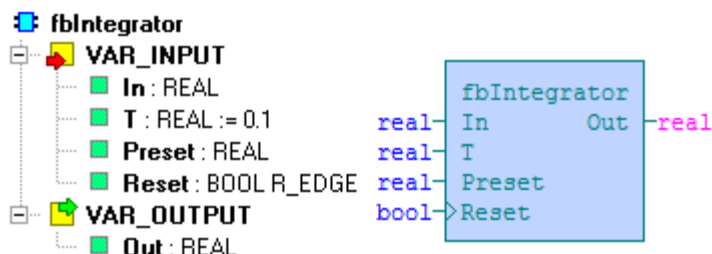
Význam hodnot enumerací:

TGeneratorSignalType - Typ produkovaného signálu		
0	gst_Sin	Sinusoida
1	gst_Square	Obdélníkový průběh
2	gst_Saw	Pilový průběh
3	gst_DblSaw	Symetrický trojúhelníkový průběh

1.2 Funkční bloky

1.2.1 fbIntegrator

knihovna: *ModelLib*



Obr. 1 struktura funkčního bloku *fbIntegrator*

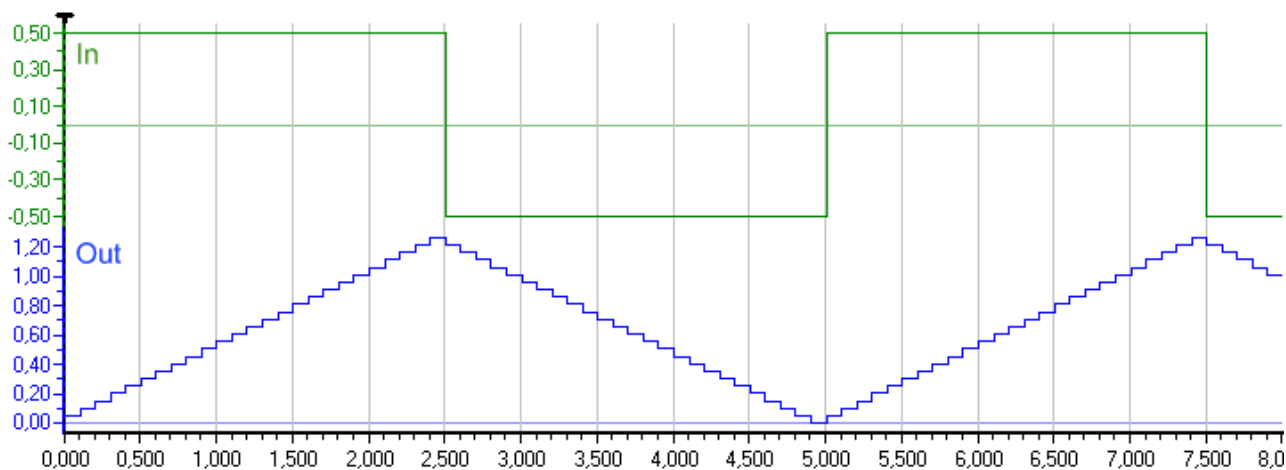
Funkční blok slouží pro simulaci integrace vstupního signálu, dle vzorce

$$Out(t) = \int_{t_0}^t In(t) dt + Preset, \text{ kde } t \text{ značí čas}$$

Výstup je vyhodnocován v rastru periody T ze vztahu získaného levou obdélníkovou aproximací






$$Out(k) = Out(k-1) + T \cdot In(k), \quad Out(0) = Preset, \text{ kde } k \text{ značí krok výpočtu.}$$

Příklad výstupu při výchozích parametrech:



Obr. 2 výstup funkčního bloku *fbIntegrator*

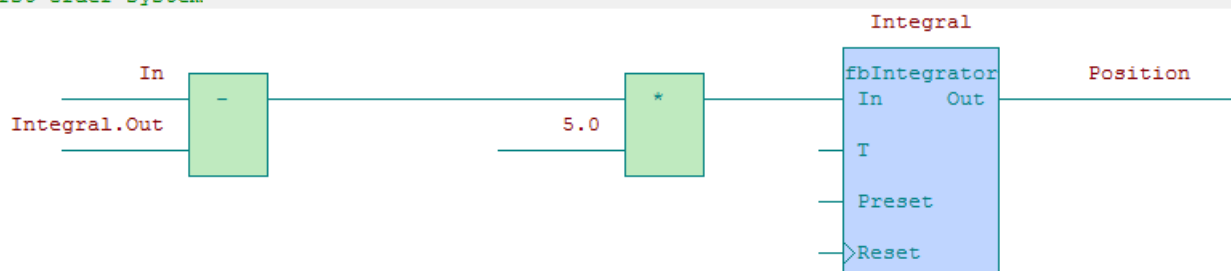
Popis proměnných:

	<i>Proměnná</i>	<i>Typ</i>	<i>Význam</i>
VAR_INPUT			
	<i>In</i>	REAL	Vstupní signál
	<i>T</i>	REAL	Perioda vzorkování v sekundách
	<i>Preset</i>	REAL	Počáteční hodnota
	<i>Reset</i>	BOOL R_EDGE	Reset výstupu na počáteční hodnotu
VAR_OUTPUT			
	<i>Out</i>	REAL	Integrál vstupního signálu

Příklad volání bloku pro řešení soustavy prvního řádu dle diferenciální rovnice

$$\frac{d Out(t)}{dt} = \frac{In(t) - Out(t)}{5} \Rightarrow Out(t) = \int_{t_0}^t \frac{In(t) - Out(t)}{5} dt$$

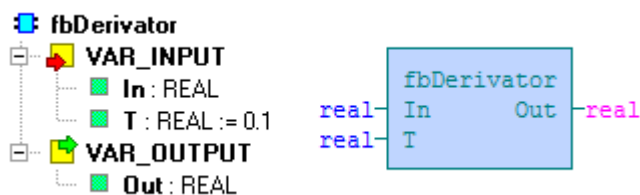
First order system



Obr. 3 Příklad volání funkčního bloku fbIntegrator

1.2.2 fbDerivator

knihovna: *Modellib*



Obr. 4 struktura funkčního bloku fbDerivator

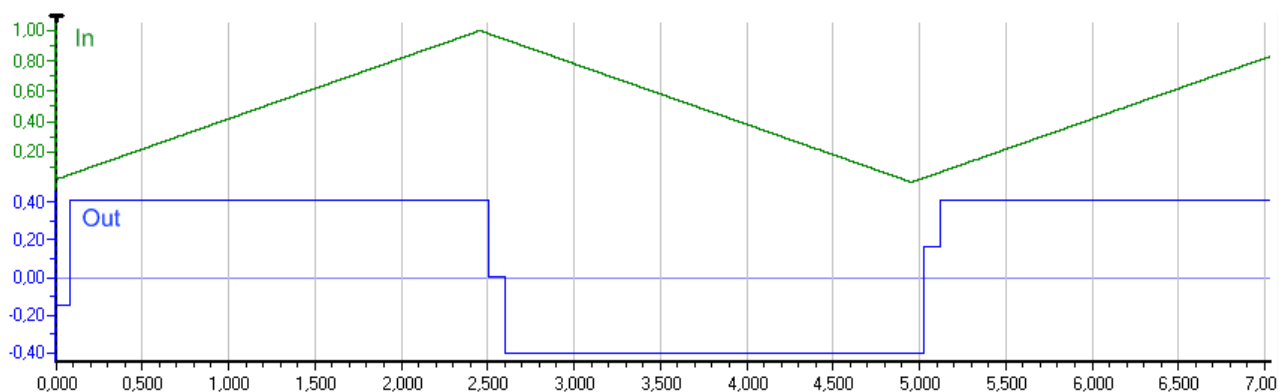
Funkční blok slouží pro simulaci derivace vstupního signálu, dle vzorce

$$Out(t) = \frac{d In(t)}{dt}, \text{ kde } t \text{ značí čas}$$

Výstup je vyhodnocován v rastru periody T ze vztahu získaného lichoběžníkovou aproximací

$$Out(k) = \frac{In(k) - In(k-1)}{T}, \text{ kde } k \text{ značí krok výpočtu.}$$

Příklad výstupu při výchozích parametrech:

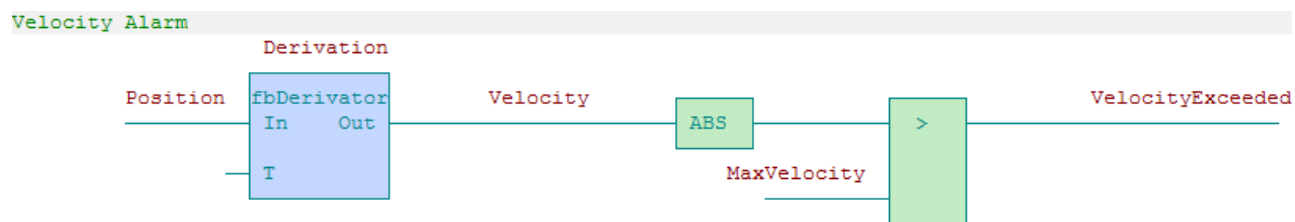


Obr. 5 výstup funkčního bloku fbDerivator

Popis proměnných:

	Proměnná	Typ	Význam
VAR_INPUT			
	<i>In</i>	REAL	Vstupní signál
	<i>T</i>	REAL	Perioda vzorkování v sekundách
VAR_OUTPUT			
	<i>Out</i>	REAL	Derivace vstupního signálu

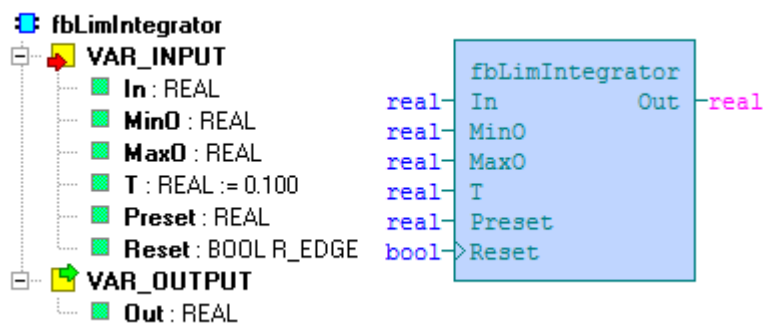
Příklad volání bloku při hlídání maximální rychlosti změny polohy



Obr. 6 Příklad volání funkčního bloku fbDerivator

1.2.3 fbLimIntegrator

knihovna: *ModelLib*



Obr. 7 struktura funkčního bloku *fbLimIntegrator*

Funkční blok slouží pro simulaci integrace vstupního signálu s omezeným výstupem, dle vzorce

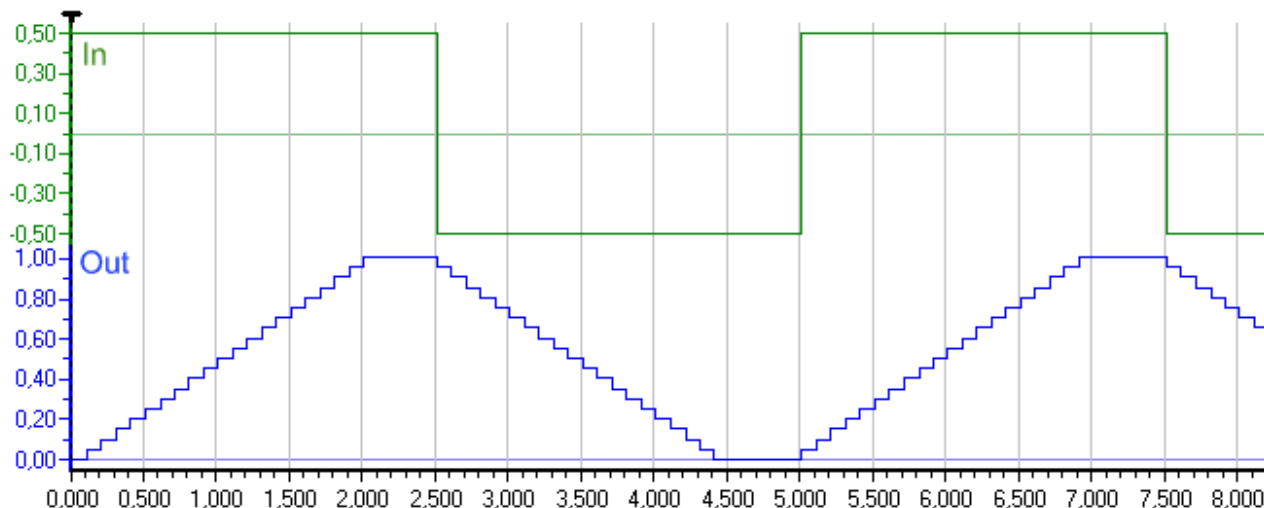
$$Out(t) = \int_{t_0}^t \begin{cases} In(t) & \text{pro } Out \in \langle MinO, MaxO \rangle \\ 0 & \text{pro } Out \notin \langle MinO, MaxO \rangle \end{cases} dt + Preset, \text{ kde } t \text{ značí čas}$$

Výstup je vyhodnocován v rastru periody T ze vztahu získaného levou obdélníkovou aproximací

$$Out(k) = \min(MaxO, \max(MinO, Out(k-1) + T \cdot In(k))), \quad Out(0) = Preset,$$

kde k značí krok výpočtu.

Příklad výstupu při výchozích parametrech a $MinO = 0.0$, $MaxO = 1.0$:

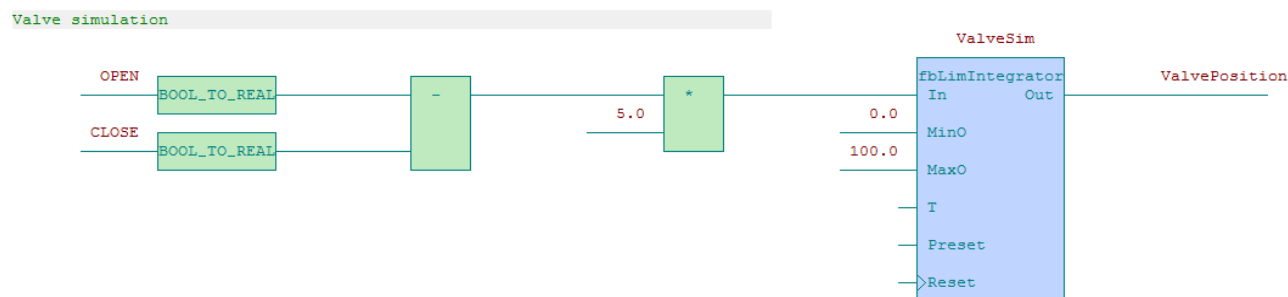


Obr. 8 výstup funkčního bloku *fbLimIntegrator*

Popis proměnných:

	<i>Proměnná</i>	<i>Typ</i>	<i>Význam</i>
VAR_INPUT			
	<i>In</i>	REAL	Vstupní signál
	<i>MinO</i>	REAL	Minimální výstup
	<i>MaxO</i>	REAL	Maximální výstup
	<i>T</i>	REAL	Perioda vzorkování v sekundách
	<i>Preset</i>	REAL	Počáteční hodnota
	<i>Reset</i>	BOOL R_EDGE	Reset výstupu na počáteční hodnotu
VAR_OUTPUT			
	<i>Out</i>	REAL	Integrál vstupního signálu

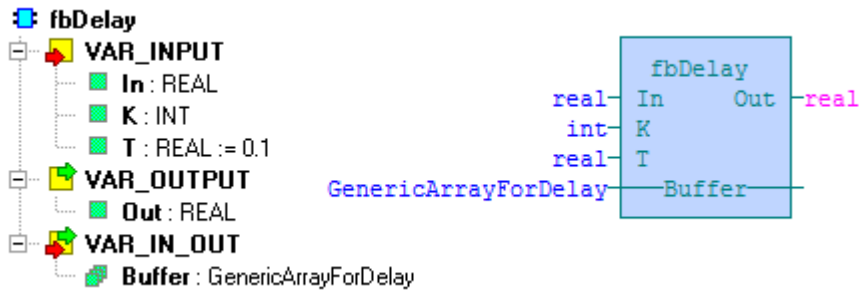
Příklad volání bloku použitého jako simulátor polohy ventilu s dobou přejezdu 20 sekund, řízeného digitálními signály *OPEN* a *CLOSE*.



Obr. 9 Příklad volání funkčního bloku *fbLimIntegrator*

1.2.4 fbDelay

knihovna: *Modellib*



Obr. 10 struktura funkčního bloku `fbDelay`

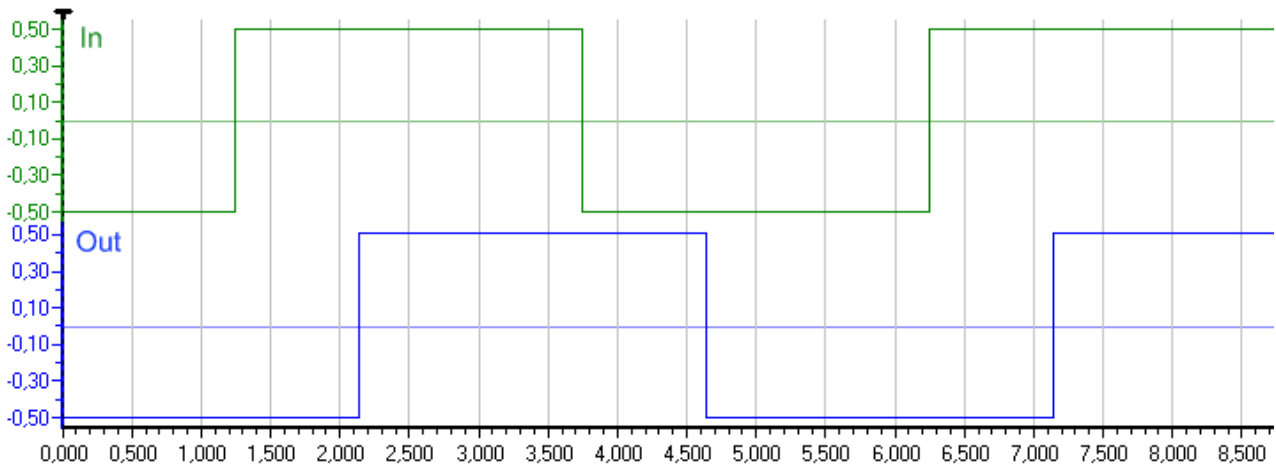
Funkční blok slouží pro simulaci dopravního zpoždění, dle vzorce

$$Out(t) = In(t - \tau), \tau = K \cdot T, \text{ kde } t \text{ značí čas}$$

Výstup je vyhodnocován v rastru periody T ze vztahu získaného levou obdélníkovou aproximací






$$Out(k) = In(k - K), \text{ kde } k \text{ značí krok výpočtu.}$$

Příklad výstupu při výchozích parametrech a $K = 10$



Obr. 11 výstup funkčního bloku `fbDelay`

Popis proměnných:

	<i>Proměnná</i>	<i>Typ</i>	<i>Význam</i>
VAR_INPUT			
	<i>In</i>	REAL	Vstupní signál
	<i>K</i>	INT	Dopravní zpoždění vyjádřené počtem period Maximum je dáno velikostí zásobníku <i>Buffer</i>
	<i>T</i>	REAL	Perioda vzorkování v sekundách
VAR_OUTPUT			
	<i>Out</i>	REAL	Zpožděný vstupní signál
VAR_IN_OUT			
	<i>Buffer</i>	GenericArrayForDelay	Zásobník minulých vzorků. Může odkazovat na libovolné pole typů REAL do velikosti 1024 prvků

Příklad volání bloku pro simulaci dopravního zpoždění 10 sekund

```

PROGRAM ExampleDelay
  VAR_INPUT
    Input : REAL;
  END_VAR
  VAR
    Delay : fbDelay;
    DelayBuffer : GenericArrayForDelay;
  END_VAR
  VAR_OUTPUT
    Output : REAL;
  END_VAR
  VAR_TEMP
  END_VAR

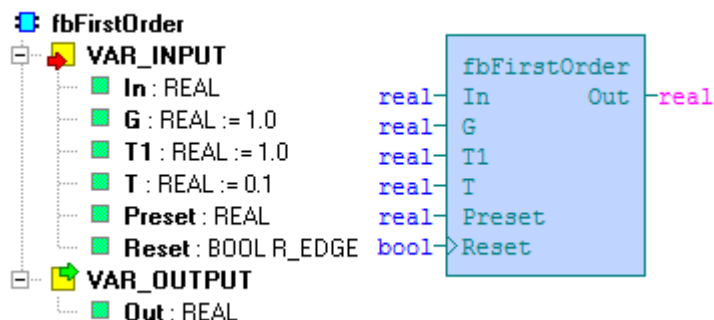
  Delay(In := Input, K := 100, T := 0.1, Buffer := DelayBuffer);
  Output := Delay.Out;

END_PROGRAM

```

1.2.5 fbFirstOrder

knihovna: *Modellib*



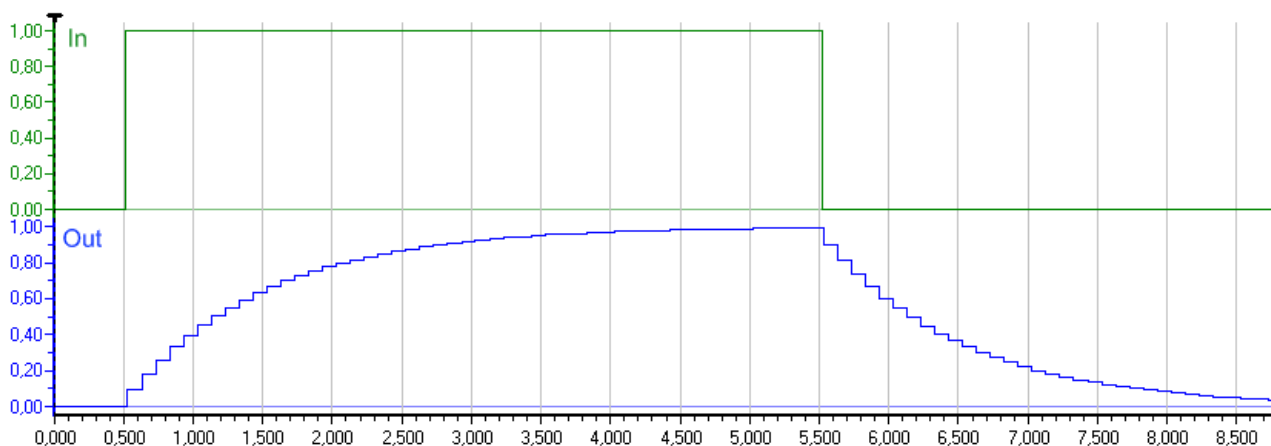
Obr. 12 struktura funkčního bloku fbFirstOrder

Funkční blok slouží pro simulaci lineárního diferenciálního systému prvního řádu se statickým zesílením G a časovou konstantou T_1 dle vzorce

$$T_1 \cdot \frac{d \text{Out}(t)}{dt} + \text{Out}(t) = G \cdot \text{In}(t) \quad , \text{ kde } t \text{ značí čas}$$

Výstup je vyhodnocován v rastru periody T pomocí aproximace získané shodou přechodové charakteristiky.

Příklad výstupu při výchozích parametrech:



Obr. 13 výstup funkčního bloku fbFirstOrder








Přechodová charakteristika soustavy prvního řádu je typická tím, že za čas odpovídající jedné časové konstantě dosáhne výstup 63,2% vstupu, za dobu 3 časových konstant 95,0% a po 5 časových konstantách 99,3%, tedy téměř ustálený stav.

Matematicky se dá přechodová charakteristika popsat:

$$\text{Out}(t) = K \cdot \left(1 - e^{-\frac{t}{\text{Tau}}} \right) \quad ,$$

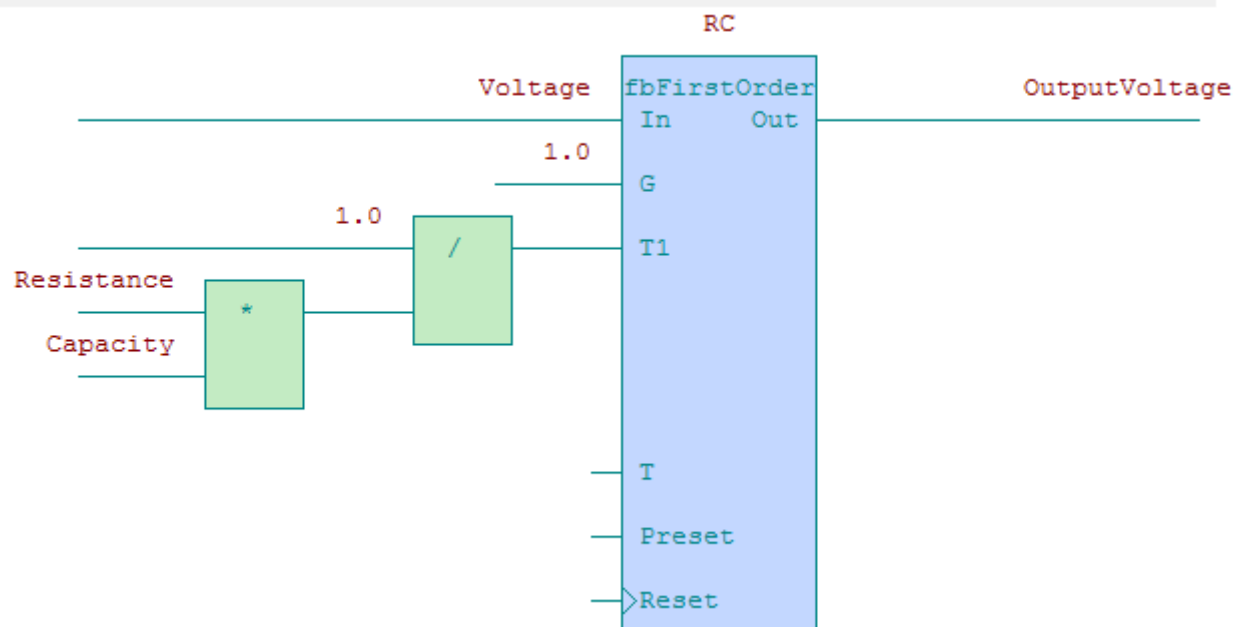
kde K je velikost skokové změny vstupu In .

Popis proměnných:

	<i>Proměnná</i>	<i>Typ</i>	<i>Význam</i>
VAR_INPUT			
	<i>In</i>	REAL	Vstupní signál
	<i>G</i>	REAL	Statické zesílení
	<i>Tl</i>	REAL	Časová konstanta v sekundách
	<i>T</i>	REAL	Perioda vzorkování v sekundách
	<i>Preset</i>	REAL	Počáteční hodnota
	<i>Reset</i>	BOOL R_EDGE	Reset výstupu na počáteční hodnotu
VAR_OUTPUT			
	<i>Out</i>	REAL	Výstupní signál

Příklad volání bloku pro simulaci dynamického chování RC obvodu

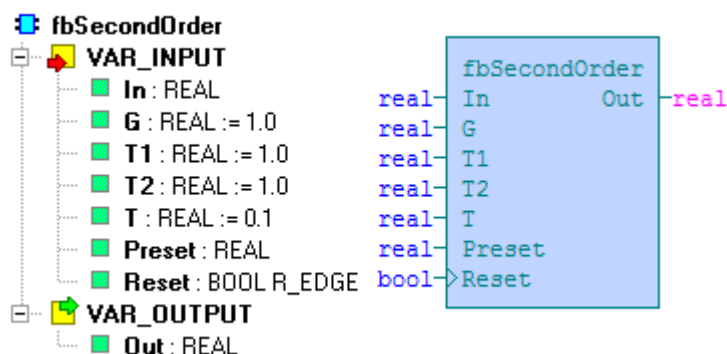
RC circuit



Obr. 14 Příklad volání funkčního bloku fbFirstOrder

1.2.6 fbSecondOrder

knihovna: *Modellib*



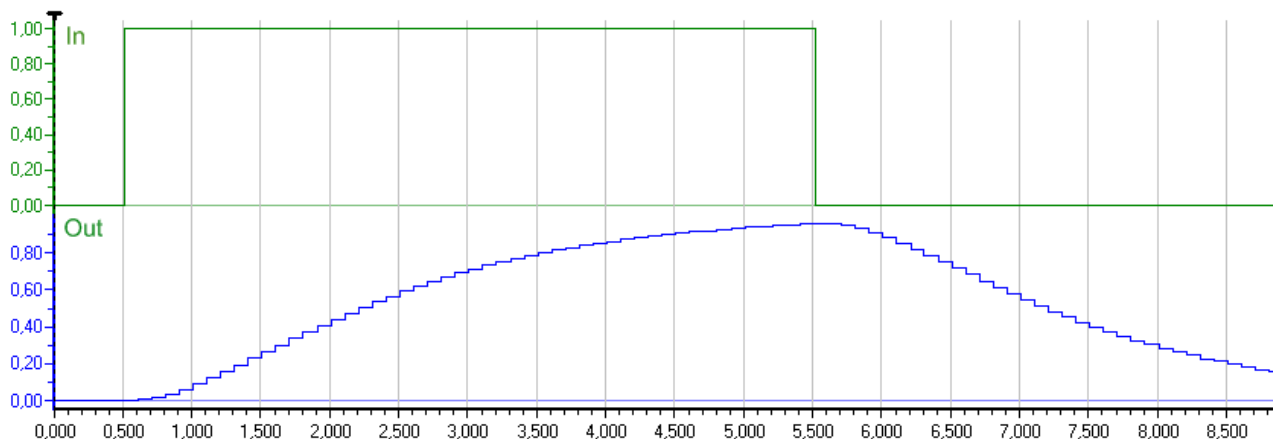
Obr. 15 struktura funkčního bloku fbSecondOrder

Funkční blok slouží pro simulaci lineárního diferenciálního systému druhého řádu se statickým zesílením G a časovou konstantou T_1 a T_2 dle vzorce

$$(T_1 \cdot T_2) \cdot \frac{d^2 Out(t)}{dt^2} + (T_1 + T_2) \cdot \frac{d Out(t)}{dt} + Out(t) = G \cdot In(t) \quad , \text{ kde } t \text{ značí čas}$$

Výstup je vyhodnocován v rastru periody T pomocí aproximace získané shodou přechodové charakteristiky.

Příklad výstupu při výchozích parametrech:



Obr. 16 výstup funkčního bloku fbSecondOrder

Chování bloku fbSecondOrder je shodné s chování dvou bloků fbFirstOrder zapojených do série s časovými konstantami Tau odpovídajícími parametrům $Tau1$ a $Tau2$.

Pro $Tau1 = Tau2$ platí pro přechodovou charakteristiku, že za dobu odpovídající jedné časové konstantě dosáhne výstup 26,4% hodnoty vstupu, za 3 časové konstanty dosáhne 80,1% a za 5 časových konstant 96,0%.

Matematicky je přechodová charakteristika pro $Tau1 = Tau2$ popsána vztahem:









$$Out(t) = K \cdot \left(1 - e^{-\frac{t}{Tau}} - \frac{t}{Tau} e^{-\frac{t}{Tau}} \right) \quad , \text{ kde } K \text{ je velikost skokové změny vstupu } In.$$

Pro $\tau_1 \neq \tau_2$ se přechodová charakteristika řídí vztahem:

$$Out(t) = K \cdot \left(1 - \frac{\tau_1}{\tau_1 - \tau_2} e^{-\frac{t}{\tau_1}} - \frac{\tau_2}{\tau_2 - \tau_1} e^{-\frac{t}{\tau_2}} \right),$$

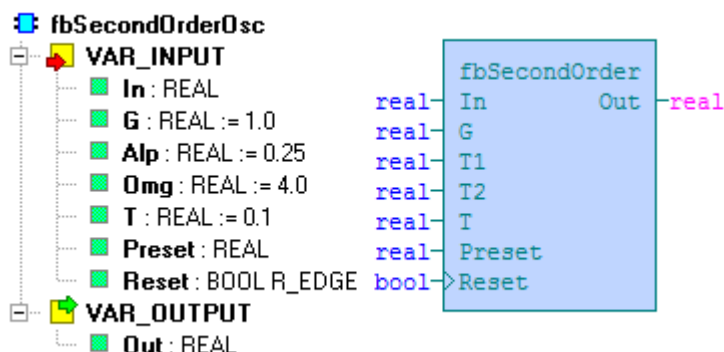
kde K je velikost skokové změny vstupu In .

Popis proměnných:

	Proměnná	Typ	Význam
VAR_INPUT			
	<i>In</i>	REAL	Vstupní signál
	<i>G</i>	REAL	Statické zesílení
	<i>T1</i>	REAL	První časová konstanta v sekundách
	<i>T2</i>	REAL	Druhá časová konstanta v sekundách
	<i>T</i>	REAL	Perioda vzorkování v sekundách
	<i>Preset</i>	REAL	Počáteční hodnota
	<i>Reset</i>	BOOL R_EDGE	Reset výstupu na počáteční hodnotu
VAR_OUTPUT			
	<i>Out</i>	REAL	Výstupní signál

1.2.7 fbSecondOrderOsc

knihovna: *Modellib*



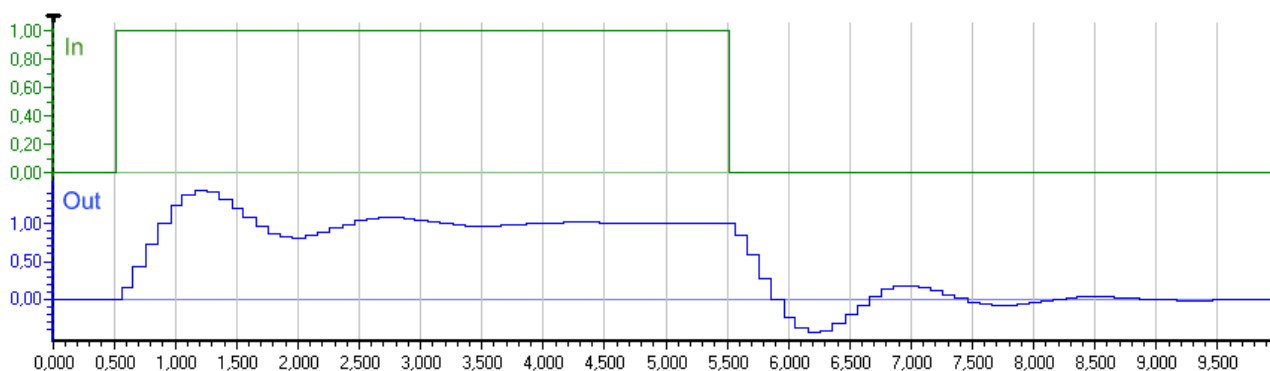
Obr. 17 struktura funkčního bloku `fbSecondOrderOsc`

Funkční blok slouží pro simulaci lineárního kmitajícího diferenciálního systému druhého řádu se statickým zesílením G , dle vzorce

$$\frac{1}{Omg^2} \cdot \frac{d^2 Out(t)}{dt^2} + \frac{2 \cdot Alp}{Omg} \cdot \frac{d Out(t)}{dt} + Out(t) = G \cdot In(t) \quad , \text{ kde } t \text{ značí čas}$$

Výstup je vyhodnocován v rastru periody T pomocí levé obdélníkové aproximace integrace.








Příklad výstupu při výchozích parametrech:



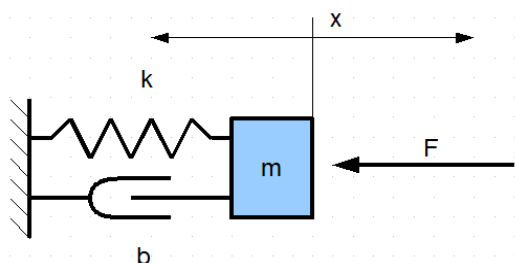
Obr. 18 výstup funkčního bloku `fbSecondOrderOsc`

Chování soustavy se vyznačuje kmitavou odezvou na skokovou změnu vstupu. Chování soustavy je určeno relativním tlumením, vlastní frekvencí a statickým zesílením. Vlastní frekvence se zadává v $\text{rad}^{-1}\text{s}^{-1}$. Perioda kmitů je rovna 2π děleno vlastní frekvencí. Převrácená hodnota relativního tlumení udává počet půlvln po skokové změně vstupu než dosáhne výstup pásma 4% kolem ustálené hodnoty. Pokud je relativní tlumení rovno 0 bude proces netlumeně kmitat.

Popis proměnných:

	<i>Proměnná</i>	<i>Typ</i>	<i>Význam</i>
VAR_INPUT			
	<i>In</i>	REAL	Vstupní signál
	<i>G</i>	REAL	Statické zesílení
	<i>Alp</i>	REAL	Koeficient tlumení
	<i>Omg</i>	REAL	Vlastní úhlová frekvence [$\text{rad}^{-1} \cdot \text{sec}^{-1}$]
	<i>Preset</i>	REAL	Počáteční hodnota
	<i>Reset</i>	BOOL R_EDGE	Reset výstupu na počáteční hodnotu
VAR_OUTPUT			
	<i>Out</i>	REAL	Výstupní signál

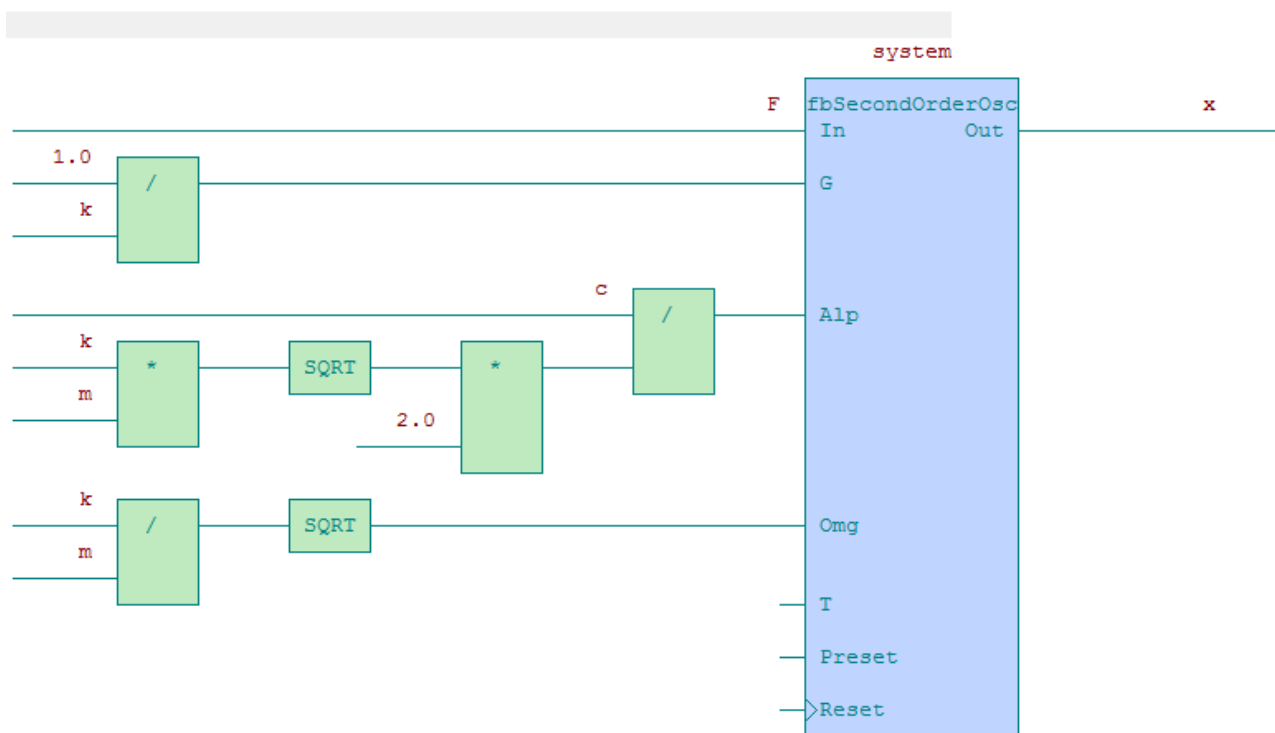
Příklad volání bloku pro simulaci dynamického chování hmoty o hmotnosti m [kg] na pružině o tuhosti k [Nm^{-1}] a tlumiči s tlumením b [Nm^{-1}s] buzené silou F [N] při zanedbání tření a odporu vzduchu.



Obr. 19 Hmota na tlumiči a pružině

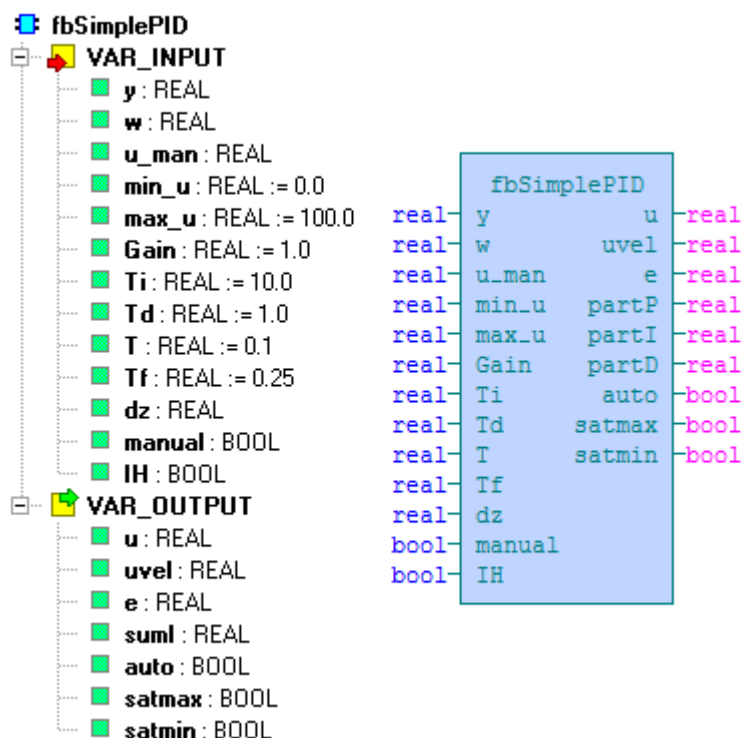
Pro daný systém platí dle 1. Newtonova zákonu pohybová rovnice

$$m \cdot \frac{d^2 x(t)}{dt^2} + c \cdot \frac{d x(t)}{dt} + k \cdot x(t) = F(t) \Rightarrow G = \frac{1}{k}, \text{Alp} = \frac{c}{2\sqrt{km}}, \text{Omg} = \sqrt{\frac{k}{m}}$$



Obr. 20 Příklad volání funkčního bloku `fbSecondOrderOsc`

1.2.8 fbSimplePID

knihovna: *Modellib*Obr. 21 struktura funkčního bloku `fbSimplePID`

Blok zajišťuje funkci PID regulátoru s filtrovanou derivační složkou. Derivační složka reaguje pouze na změny regulované veličiny, což umožňuje rychlé změny žádané veličiny bez nežádoucích akčních zásahů. Algoritmus ošetřuje omezení integrační složky při saturaci výstupu (antiwind-up) a beznárazový přechod z manuálního do automatického režimu.

Měřená veličina se přivádí na vstup `y` a její žádaná hodnota na vstup `w`. Vstup `u_man` slouží jako hodnota výstupu v případě, že je regulátor přepnutý do ručního režimu (vstup `manual` je TRUE). Vstupy `min_u` a `max_u` slouží k omezení výstupu `u`.

Parametry regulačního algoritmu PID jsou předávány na vstupech `Gain` – zesílení (pro opačný směr akčního zásahu je možné použít minusové hodnoty), `Ti` – integrační časová konstanta, `Td` – derivační časová konstanta, `T` – vzorkovací perioda regulátoru a `Tf` – časová konstanta filtrace derivační složky. Všechny parametry kromě zesílení jsou v sekundách.

Algoritmus je doplněn o symetrické pásmo necitlivosti `dz` a pozastavení integrace `IH`. Pokud se regulovaná veličina `y` pohybuje v pásmu $\pm dz$ od žádané hodnoty `w`, je výstup `u` regulátoru konstantní. Pokud je nastaven vstup `IH` na TRUE je integrační část výstupu konstantní. Této vlastnosti jde využít při řízení s pomalými servo pohony, kdy po dobu pohybu pohonu není aktivní integrační část algoritmu, aby se zabránilo kmitání soustavy.

Kromě akční veličiny `u`, která je součtem všech složek PID, blok vrací jednotlivé části akční veličiny pro usnadnění empirického nastavování. **Pokud je třeba třístavové řízení s binárními signály otevřít/zavřít je možné použít blok `fbStepControl`.**

Proporcionální část akčního zásahu je na výstupu $partP$. Tato část je ovlivněna pouze rozdílem žádané hodnoty w a regulované veličiny y a zesílením $Gain$. Rozdíl w a y je vrácen jako regulační odchylka na výstupu e .

$$partP = Gain \cdot e$$

Integrační složka je na výstupu $partI$. Integrační složka je součtem regulačních odchylek e během času. Tato složka je ovlivněna zesílením $Gain$ a integrační konstantou T_i . Čím delší konstanta T_i tím pomalejší je změna integrační složky. Při T_i rovno nula je integrační složka zcela vyřazena z výpočtu. Integrační složka je navíc omezena v případě saturace výstupu nebo je její výpočet zastaven při nastavení vstupu IH na TRUE.

$$partI = \frac{Gain \cdot T}{T_i} \cdot \sum_{k=0}^{k=n} \left(\frac{y_k + y_{k-1}}{2} \right) ,$$

kde k je index kroku regulátoru a n je počet realizovaných kroků.

Derivační částí odpovídá výstup $partD$. Derivační část reaguje na rychlost změny regulované veličiny y . Tato veličina je navíc filtrována filtrem prvního řádu s časovou konstantou T_f . Tento filtr potlačuje nežádoucí akční zásahy způsobené šumem na měřené veličině. Čím větší je tato konstanta, tím více jsou potlačeny změny měřené veličiny, na které derivační složka reaguje. Doporučená hodnota T_f je jedna čtvrtina konstanty T_d . Vlastní derivační složka je ovlivněna zesílením $Gain$ a derivační konstantou T_d . Čím je větší konstanta T_d tím jsou větší akční zásahy od derivační složky. T_d rovno nula zcela derivační složku potlačí.

$$partD = \frac{Gain \cdot T_d}{T} \cdot (y_{f[n]} - y_{f[n-1]}) ,$$

kde n je index aktuálního kroku regulátoru a y_f filtrovaná regulovaná veličina y .

Pokud výstupní veličina u dosáhne saturace (hodnoty max_u respektive min_u) je tento stav signalizován na výstupech $satmax$ respektive $satmin$.

Chování regulátoru mimo saturaci je popsáno následující teoretickou rovnicí, kterou řídicí algoritmus aproximuje

$$u(t) = Gain \cdot \left(e(t) + \frac{1}{T_i} \cdot \int_{t_0}^t e(t) dt - T_d \cdot \frac{d y_f(t)}{dt} \right) , \text{ kde } t \text{ značí čas}$$





y_f je filtrovaný vstup derivační složky. Filtraci zajišťuje filtr prvního řádu, dle rovnice

$$T_f \cdot \frac{d y_f(t)}{dt} + y_f(t) = y(t)$$

Výstup je vyhodnocován v rastru periody T .

Popis proměnných:

	<i>Proměnná</i>	<i>Typ</i>	<i>Význam</i>
VAR_INPUT			
	<i>y</i>	REAL	Měřená veličina
	<i>w</i>	REAL	Žádaná hodnota
	<i>u_man</i>	REAL	Hodnota výstupu v ručním režimu
	<i>min_u</i>	REAL	Minimální hodnota výstupu
	<i>max_u</i>	REAL	Maximální hodnota výstupu
	<i>Gain</i>	REAL	Zesílení
	<i>Ti</i>	REAL	Integrační konstanta v sekundách
	<i>Td</i>	REAL	Derivační časová konstanta v sekundách
	<i>T</i>	REAL	Perioda vzorkování v sekundách
	<i>Tf</i>	REAL	Časová konstanta filtru derivační složky v sekundách
	<i>dz</i>	REAL	Pásmo necitlivosti V pásmu $\pm dz$ kolem žádané hodnoty je výstup regulátoru konstantní
	<i>manual</i>	BOOL	Nastavení ručního režimu 0 – automatický režim, 1 – ruční režim
	<i>IH</i>	BOOL	Zastavení integrace 0 – integrace povolena, 1 – integrace zastavena
VAR_OUTPUT			
	<i>u</i>	REAL	Výstup regulátoru
	<i>uvel</i>	REAL	Rychlost změny výstupu regulátoru za poslední vzorkovací periodu [s^{-1}]
	<i>partP</i>	REAL	Proporcionální část výstupu regulátoru
	<i>partI</i>	REAL	Integrační část výstupu regulátoru
	<i>partD</i>	REAL	Derivační část výstupu regulátoru

	<i>Proměnná</i>	<i>Typ</i>	<i>Význam</i>
	<i>e</i>	REAL	Regulační odchylka
	<i>auto</i>	BOOL	Režim regulátoru 0 – ruční režim, 1 – automatický režim
	<i>satmax</i>	BOOL	Výstup saturován na maximální hodnotě
	<i>satmin</i>	BOOL	Výstup saturován na minimální hodnotě

Příklad volání v jazyku ST

```

TYPE
  T_SimplePID_Param : STRUCT
    Gain : REAL;
    Ti   : REAL;
    Td   : REAL;
    T    : REAL;
    Tf   : REAL;
    dz   : REAL;
  END_STRUCT;
END_TYPE

VAR_GLOBAL RETAIN
  Manual      : BOOL;
  SetPoint    : REAL;
  ManualValue : REAL;
  ParSimplePID1 : T_SimplePID_Param := (Gain := 5.0,
                                         Ti   := 40.0,
                                         Td   := 1.0,
                                         T    := 1.0,
                                         Tf   := 1.0,
                                         dz   := 0.5);
END_VAR

PROGRAM prgExampleSimplePID
  VAR
    SimplePID1 : fbSimplePID;
  END_VAR

  SimplePID1(y      := AnalogInput,
             w      := SetPoint,
             u_man  := ManualValue,
             Gain   := ParSimplePID1.Gain,
             Ti     := ParSimplePID1.Ti,
             Td     := ParSimplePID1.Td,
             T      := ParSimplePID1.T,
             Tf     := ParSimplePID1.Tf,
             dz     := ParSimplePID1.dz,
             manual := Manual,
             u => AnalogOutput);

```

```
IF Manual THEN
  ManualValue := SimplePID1.u;
END_IF;

END_PROGRAM
```

Význam jednotlivých proměnných:

AnalogInput – analogový vstup s regulovanou veličinou

AnalogOutput – analogový výstup ovládající akční člen

SetPoint – proměnná s žádanou hodnotou

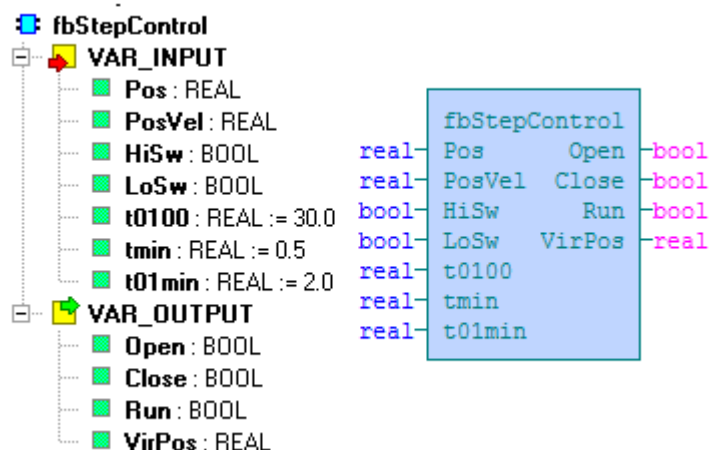
Manual – příznak ručního režimu

ManualValue – hodnota výstupu v ručním režimu

ParSimplePID1 – parametry regulátoru

1.2.9 fbStepControl

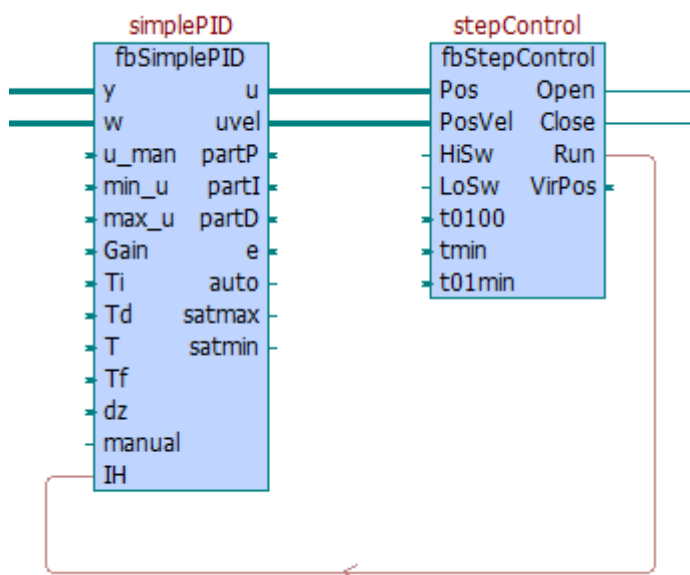
knihovna: *ModelLib*



Obr. 22 struktura funkčního bloku *fbStepControl*

Blok *fbStepControl* zajišťuje převod analogového signálu 0 až 100% na třístavové řízení pomocí binárních signálů otevřít/zavřít. Primárně je blok určen pro spojení s blokem *fbSimplePID*. Žádaná pozice řízeného servo pohonu se předává na vstupu *Pos*. Vstup *PosVel* může nést přidanou informaci o rychlosti změny vstupu *Pos*. Tato hodnota je nepovinná, nicméně zlepšuje predikci budoucí žádané polohy.

Propojení s blokem *fbSimplePID* se realizuje připojením výstupu *u* na vstup *Pos* a výstupu *uvel* na vstup *PosVel*. Pro pomalé servopohony je řízení možno vylepšit propojením výstupu *Run* bloku *fbStepControl* na vstup *IH* bloku *fbSimplePID*. Omezení výstupu bloku *fbSimplePID* by mělo být ponecháno ve výchozích hodnotách *min_u* rovno 0 a *max_u* rovno 100.














Obr. 23 Příklad propojení bloku *fbSimplePID* s blokem *fbStepControl*

Pokud jsou k dispozici koncové snímače polohy servopohonu lze je připojit na vstupy *HiSw* (koncový spínač plně otevřeno) a *LoSw* (koncový spínač plně zavřeno). Tyto vstupy eliminují zbytečné pulsy při dosažení koncových poloh a kalibrují vypočtenou hodnotu polohy pohonu *VirPos* proti skutečné poloze.

Výstupy *Open* – otvírat a *Close* – zavírat jsou ovládány v závislosti na zadaných parametrech pohonu na vstupech *t0100*, *tmin* a *t01min*. Vstup *t0100* udává dobu v sekundách, za kterou servopohon urazí vzdálenost z 0 do 100% otevření. Vstup *tmin* udává v sekundách minimální délku pulzu a dobu mezi pulzy na výstupech *Open* a *Close*. Vstup *t01min* je v sekundách vyjádřená minimální doba mezi pulzy opačné polarity.

Výstup *Run* je logickým součtem výstupů *Open* a *Close*, signalizující požadavek na pohyb

Popis proměnných:

	Proměnná	Typ	Význam
VAR_INPUT			
	<i>Pos</i>	REAL	Žádaná poloha servopohonu
	<i>PosVel</i>	REAL	Rychlost změny žádané polohy servopohonu
	<i>HiSw</i>	REAL	Koncový spínač plně otevřené polohy
	<i>LoSw</i>	REAL	Koncový spínač plně zavřené polohy
	<i>t0100</i>	REAL	Doba přechodu z plně zavřené do plně otevřené polohy v sekundách
	<i>tmin</i>	REAL	Minimální délka pulzu a doba mezi pulzy v sekundách
	<i>t01min</i>	REAL	Minimální doba mezi pulzy opačné polarity v sekundách
VAR_OUTPUT			
	<i>Open</i>	BOOL	Požadavek otvírat
	<i>Close</i>	BOOL	Požadavek zavírat
	<i>Run</i>	BOOL	Pohon v pohybu – logický součet požadavku otvírat a zavírat
	<i>VirPos</i>	REAL	Vypočtená virtuální poloha pohonu

Příklad volání v jazyku ST

```

TYPE
  T_SimplePID_Param : STRUCT
    Gain : REAL;
    Ti   : REAL;
    Td   : REAL;
    T    : REAL;
    Tf   : REAL;
    dz   : REAL;
  END_STRUCT;
END_TYPE

VAR_GLOBAL RETAIN
  Manual       : BOOL;
  SetPoint     : REAL;
  ManualValue  : REAL;
  ParSimplePID1 : T_SimplePID_Param := (Gain := 5.0, Ti := 40.0,
                                         Td := 1.0, T := 1.0,
                                         Tf := 1.0, dz := 0.5);
END_VAR

PROGRAM prgExampleStepControl
  VAR
    SimplePID1 : fbSimplePID;
    StepControl1 : fbStepControl;
  END_VAR

  SimplePID1(y := AnalogInput,
             w := SetPoint,
             u_man := ManualValue,
             Gain := ParSimplePID1.Gain,
             Ti := ParSimplePID1.Ti,
             Td := ParSimplePID1.Td,
             T := ParSimplePID1.T,
             Tf := ParSimplePID1.Tf,
             dz := ParSimplePID1.dz,
             manual := Manual,
             IH := StepControl1.Run);

  StepControl1(Pos := SimplePID1.u,
              PosVel := SimplePID1.uel,
              t0100 := 60.0,
              tmin := 1.0,
              t0lmin := 2.0,
              Open => Output1,
              Close => Output2);

  IF Manual THEN
    ManualValue := SimplePID1.u;
  END_IF;
END_PROGRAM

```

Význam jednotlivých proměnných:

AnalogInput – analogový vstup s regulovanou veličinou

Output1, Output2 – binární výstupy ovládající akční člen

SetPoint – proměnná s žádanou hodnotou

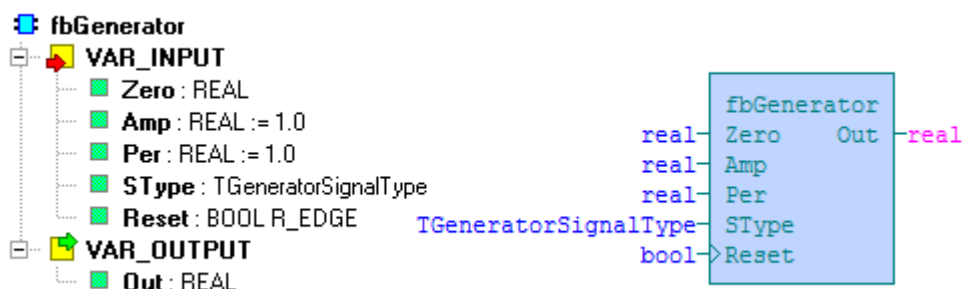
Manual – příznak ručního režimu

ManualValue – hodnota výstupu v ručním režimu

ParSimplePID1 – parametry regulátoru

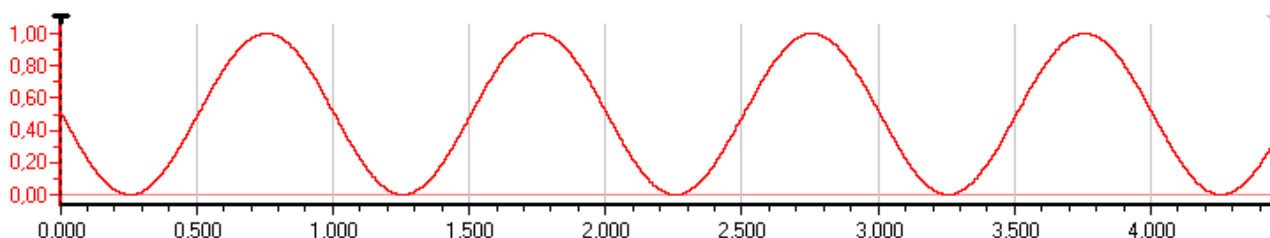
1.2.10 fbGenerator

knihovna: *Modellib*

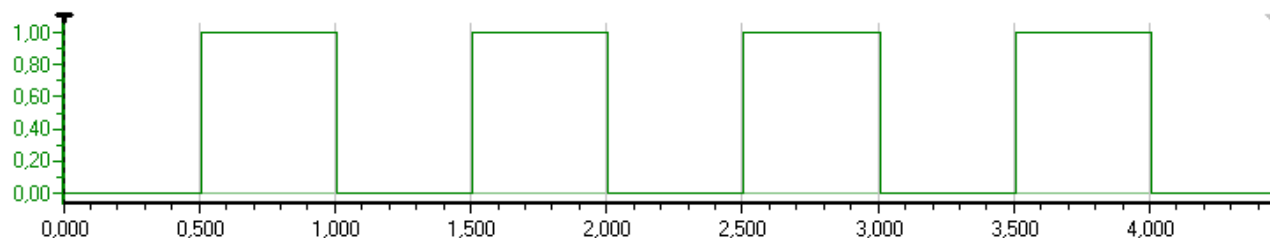


Obr. 24 struktura funkčního bloku `fbGenerator`

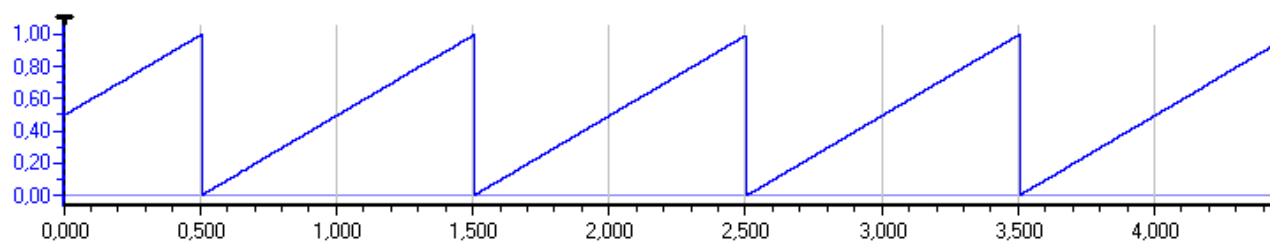
Blok funguje jako generátor signálu. Vstupní proměnná `SType` ovlivňuje tvar výstupního signálu:



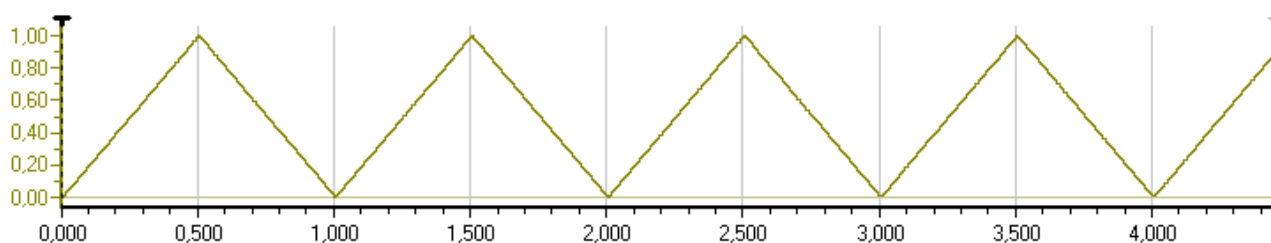
Obr. 25 Výstup bloku `fbGenerator` pro `Zero = 0.0`, `Amp = 1.0`, `Per = 1.0`, `SType = gst_Sin`



Obr. 26 Výstup bloku `fbGenerator` pro `Zero = 0.0`, `Amp = 1.0`, `Per = 1.0`, `SType = gst_Square`









Obr. 27 Výstup bloku `fbGenerator` pro `Zero = 0.0`, `Amp = 1.0`, `Per = 1.0`, `SType = gst_Saw`



Obr. 28 Výstup bloku fbGenerator pro $Zero = 0.0$, $Amp = 1.0$, $Per = 1.0$, $SType = gst_DblSaw$

Popis proměnných:

	Proměnná	Typ	Význam
VAR_INPUT			
	<i>Zero</i>	REAL	Posun nuly generovaného signálu
	<i>Amp</i>	REAL	Amplituda výstupního signálu
	<i>Per</i>	REAL	Perioda výstupního signálu
	<i>SType</i>	REAL	Typ výstupního signálu
	<i>Reset</i>	REAL	Reset generátoru
VAR_OUTPUT			
	<i>Out</i>	REAL	Výstup generátoru



teco

Objednávky a informace:

Teco a. s. Havlíčkova 260, 280 58 Kolín 4, tel. 321 737 611, fax 321 737 633

TXV 003 44.01

Výrobce si vyhrazuje právo na změny dokumentace. Poslední aktuální vydání je k dispozici na internetu www.tecomat.com