

PRÍLOHA P 1:
PREHLAD VŠETKÝCH BLOKOV

OBSAH

1.1	CONTINUOUS TIME SYSTEMS	7
1.1.1	CLINDUMMY_f	7
1.1.2	CLR	7
1.1.3	CLSS	7
1.1.4	DERIV	8
1.1.5	INTEGRAL_f	9
1.1.6	INTEGRAL_m	9
1.1.7	PID	10
1.1.8	TCLSS	11
1.1.9	TIME_DELAY	12
1.1.10	VARIABLE_DELAY	12
1.1.11	PDE	13
1.2	DISCRETE TIME SYSTEMS	15
1.2.1	DELAY_f	16
1.2.2	DLR	17
1.2.3	DLSS	17
1.2.4	DOLLAR, DOLLAR_f, DOLLAR_m	18
1.2.5	SAMPHOLD_m	19
1.2.6	REGISTER	19
1.3	DISCONTINUITIES	20
1.3.1	BACKLASH	20
1.3.2	DEADBAND	20
1.3.3	HYSTHERESIS	21
1.3.4	RATELIMITER	22
1.3.5	QUANT_f	22
1.3.6	SATURATION	23
1.4	LOOKUP TABLES	24
1.4.1	INTRPLBLK_f	24
1.4.2	INTRP2BLK_f	24
1.4.3	LOOKUP_f	25
1.5	EVENT HANDLING	26
1.5.1	CLOCK_c	26
1.5.2	SampleCLK	26
1.5.3	VirtualCLK0	27
1.5.4	ANDBLK	28
1.5.5	ANDLOG_f	28
1.5.6	CEVENTSCOPE	28
1.5.7	CLKFROM	29
1.5.8	CLKGOTO	30
1.5.9	CLKGoto TagVisibility	30
1.5.10	CLKSOMV_f	31
1.5.11	EDGE_TRIGGER	31
1.5.12	ENDBLK	32
1.5.13	END_c	32
1.5.14	ESELECT_f	33
1.5.15	EVTDLY_c	33

1.5.16	EVTGEN_f	34
1.5.17	EVTVARDLY	35
1.5.18	HALT_f.....	35
1.5.19	IFTHEL_f.....	36
1.5.20	M_freq.....	36
1.5.21	freq_div	37
1.6	MATHEMATICAL OPERATIONS	38
1.6.1	ABS_VALUE	38
1.6.2	BIGSOM_f.....	38
1.6.3	COSBLK_f.....	39
1.6.4	EXPBLK_m.....	39
1.6.5	GAINBLK, GAINBLK_f, GAIN	39
1.6.6	INVBLK.....	40
1.6.7	LOGBLK_f	40
1.6.8	MATMAGPHI	41
1.6.9	MATZREIM	42
1.6.10	MAXMIN.....	42
1.6.11	MAX_f.....	43
1.6.12	MIN_f.....	43
1.6.13	POWBLK_f.....	44
1.6.14	PRODUCT	44
1.6.15	PROD_f.....	45
1.6.16	SIGNUM.....	45
1.6.17	SINBLK_f.....	45
1.6.18	SQRT.....	45
1.6.19	SUMMATION	46
1.6.20	SUM_f.....	47
1.6.21	TANBLK_f	47
1.6.22	TrigFun.....	47
1.7	MATRIX.....	48
1.7.1	CUMSUM.....	48
1.7.2	EXTRACT	48
1.7.3	EXTTRI.....	49
1.7.4	MATBKSL.....	50
1.7.5	MATCATH.....	50
1.7.6	MATCHTV	51
1.7.7	MATDET	51
1.7.8	MATDIAG.....	52
1.7.9	MATDIV.....	52
1.7.10	MATEIG	53
1.7.11	MATEXPM.....	54
1.7.12	MATINV.....	54
1.7.13	MATLU.....	55
1.7.14	MATMUL	55
1.7.15	MATPINV.....	56
1.7.16	MATRESH.....	57
1.7.17	MATSING.....	57
1.7.18	MATSUM	58
1.7.19	MATTRAN	59

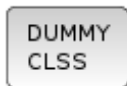
1.7.20	MATCONJ.....	59
1.7.21	RICC	60
1.7.22	ROOTCOEF.....	60
1.7.23	SUBMAT	61
1.8	ELECTRICAL	62
1.8.1	CCS	62
1.8.2	CVS	62
1.8.3	Capacitor	62
1.8.4	ConstantVoltage	63
1.8.5	CurrentSenzor	63
1.8.6	Diode	64
1.8.7	Ground.....	64
1.8.8	Gyrator	65
1.8.9	IdealTransformer.....	65
1.8.10	Inductor	66
1.8.11	NMOS	67
1.8.12	NPN.....	68
1.8.13	OpAmp.....	70
1.8.14	PMOS.....	70
1.8.15	PNP	71
1.8.16	PotentialSensor.....	72
1.8.17	Resistor.....	73
1.8.18	SineVoltage	73
1.8.19	Switch.....	74
1.8.20	VVsourceAC	74
1.8.21	VariableResistor	75
1.8.22	VoltageSensor	75
1.8.23	VsourceAC.....	75
1.9	PORT & SUBSYSTEMS.....	77
1.9.1	CLKINV_f	77
1.9.2	CLKOUTV_f	77
1.9.3	IN_f	78
1.9.4	INIMPL_f.....	78
1.9.5	OUTIMPL_f.....	79
1.9.6	OUT_f	79
1.10	ZERO CROSSING DETECTION	81
1.10.1	NEGTOPOS_f.....	81
1.10.2	POSTONEG_f.....	81
1.10.3	ZCROSS_f	81
1.11	SIGNAL ROUTING	82
1.11.1	DEMUX, DEMUX_f.....	82
1.11.2	EXTRACTOR.....	82
1.11.3	FROM	83
1.11.4	FROMMO	83
1.11.5	GOTO.....	84
1.11.6	GOTOMO	84
1.11.7	GotoTagVisibility	85
1.11.8	GotoTagVisibilityMO	85
1.11.9	ISELECT_m.....	86

1.11.10	MUX, MUX_f.....	87
1.11.11	M_SWITCH.....	87
1.11.12	NRMSOM_f.....	88
1.11.13	RELAY_f.....	88
1.11.14	SELECT_m.....	89
1.11.15	SWITCH2_m.....	90
1.11.16	SWITCH_f.....	90
1.11.17	SELF_SWITCH.....	91
1.12	INTEGER.....	92
1.12.1	BITCLEAR.....	92
1.12.2	BITSET.....	92
1.12.3	CONVERT.....	93
1.12.4	DFLIPFLOP.....	94
1.12.5	DLATCH.....	94
1.12.6	EXTRACTBITS.....	95
1.12.7	INTMUL.....	96
1.12.8	JKFLIPFLOP.....	96
1.12.9	LOGIC.....	97
1.12.10	SHIFT.....	98
1.12.11	SRFLIPFLOP.....	99
1.13	ANNOTATIONS.....	101
1.13.1	TEXT_f.....	101
1.14	SINKS.....	102
1.14.1	AFFICH_m.....	102
1.14.2	BARXY.....	103
1.14.3	CANIMXY.....	104
1.14.4	CANIMXY3D.....	105
1.14.5	CFSCOPE.....	106
1.14.6	CMAT3D.....	107
1.14.7	CMATVIEW.....	108
1.14.8	CMSCOPE.....	109
1.14.9	CSCOPE.....	110
1.14.10	CSCOPXY.....	111
1.14.11	CSCOPXY3D.....	112
1.14.12	TOWS_c.....	114
1.14.13	TRASH_f.....	114
1.14.14	WRITEAU_f.....	115
1.14.15	WRITEC_f.....	115
1.15	SOURCES.....	117
1.15.1	CONST, CONST_m, CONST_f.....	117
1.15.2	CURV_f.....	117
1.15.3	Counter.....	118
1.15.4	FROMWSB.....	119
1.15.5	GENSIN_f.....	120
1.15.6	GENSQR_f.....	121
1.15.7	INIMPL_f.....	121
1.15.8	IN_f.....	122
1.15.9	Modulo_Count.....	122
1.15.10	RAMP.....	123

1.15.11	RAND_m	123
1.15.12	READAU_f	124
1.15.13	READC_f	125
1.15.14	RFILE_f	127
1.15.15	SAWTOOTH_f	128
1.15.16	STEP_FUNCTION	128
1.15.17	PULSE_SC	129
1.15.18	Sigbuilder	130
1.15.19	TIME_f	130
1.15.20	TKSCALE	131
1.16	USER-DEFINED FUNCTION	132
1.16.1	CBLOCK	132
1.16.2	EXPRESSION	133
1.16.3	MBLOCK	133
1.16.4	SUPER_f	135
1.16.5	c_block	136
1.16.6	fortran_block	136
1.16.7	generic_block3	137
1.16.8	scifunc_block_m	138

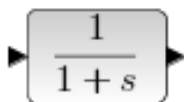
1.1 Continuous time systems

1.1.1 CLINDUMMY_f

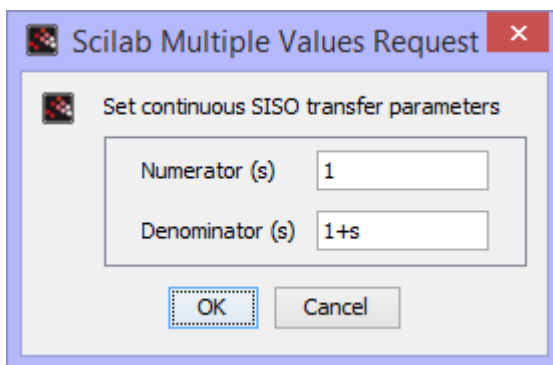


Tento blok by mal byť umiestnený v každom funkčnom diagrame, ktorý obsahuje blok typu „zero-crossing“, a zároveň neobsahuje blok typu „continuous-time state“. Dôvodom je možnosť nájdenia „zero-crossing“ prostredia.

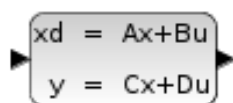
1.1.2 CLR



Tento blok slúži na realizáciu SISO lineárnych systémov, reprezentovaný jeho racionálnym prenosom funkcie „Numerator/Denominator“ (Čitateľ/Menovateľ). Racionálna funkcia musí byť správna (stupeň menovateľa musí byť väčší alebo rovný stupňu čitateľa).

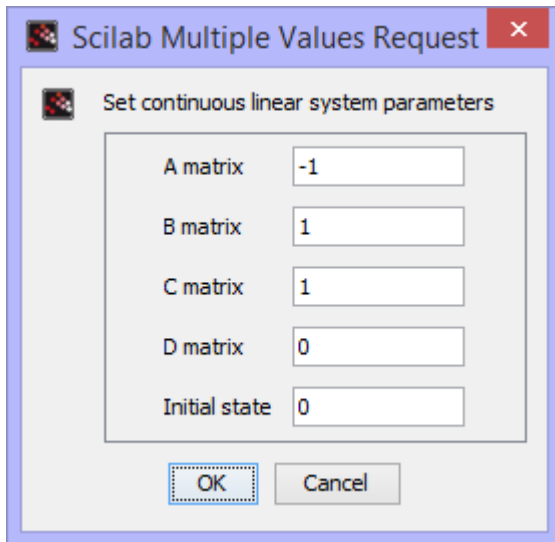


1.1.3 CLSS



Tento blok realizuje stavový popis spojitého lineárneho systému.

System je definovaný maticami (A, B, C, D) a počiatocným stavom X0. Dimenzie musia byť zhodné.



- **A matrix** (matica A): štvorcová matica.
- **B matrix** (matica B): ak systém nemá výstup, je matica prázdna "[]".
- **C matrix** (matica C): ak systém nemá výstup, je matica prázdna "[]".
- **D matrix** (matica D): ak systém nemá matematický člen D, je matica prázdna "[]".
- **Initial state** (počiatočný stav): vektor počiatocného stavu systému.

1.1.4 DERIV



Blok zabezpečujúci deriváciu vstupu podľa základného vzťahu:

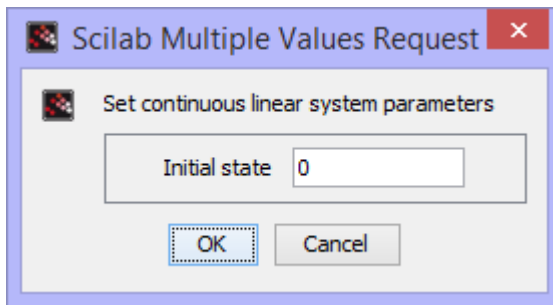
$$\frac{\Delta u}{\Delta t}$$

Tento blok nemá žiadne nastavenia a jeho počiatocná hodnota na výstupe je nula.

1.1.5 INTEGRAL_f



Tento blok pracuje ako integrátor. Jeho výstup je integrálom vstupu.



- **Initial Condition:** vektor, ktorý udáva počiatočný stav.

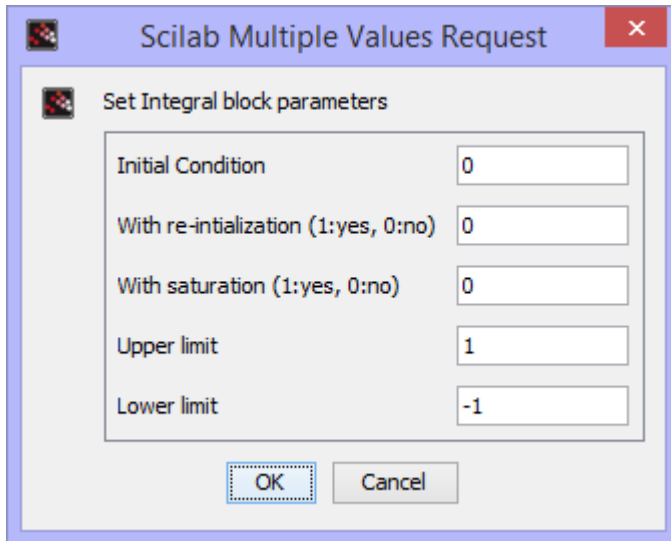
1.1.6 INTEGRAL_m



Blok pracujúci ako integrátor. Jeho výstup y je integrálom vstupu u v aktuálnom časovom kroku t .

$$y(t) = \int_{t_0}^t u(t)dt + y_0$$

- y_0 je počiatočná hodnota a t_0 je zodpovedajúci čas

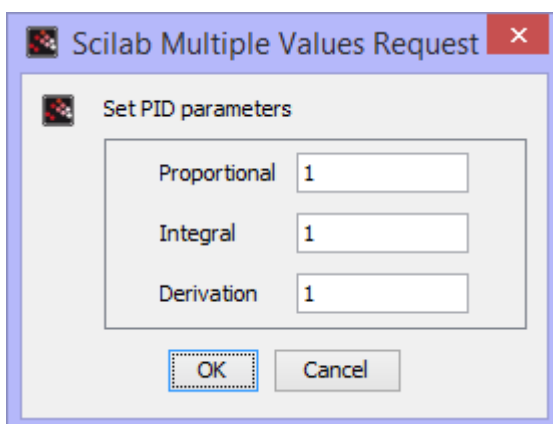


- **Initial Condition:** počiatočná hodnota, ktorá definuje dátový typ na vstupe a výstupe. Môže byť reálneho alebo komplexného typu.
- **With re-initialization:** resetovanie stavu do špecifickej počiatočnej hodnoty založenej na výbere externého signálu 1.
- **With saturation:** ak je zvolená „1:yes“, obmedzuje stavy na hodnotu medzi horným a dolným limitom nasýtenia („Upper limit“ a „Lower limit“).
- **Upper limit:** horný limit pre integrál.
- **Lower limit:** dolný limit pre integrál.

1.1.7 PID



Blok implementujúci PID regulátor. Vypočítava „chybu“ ako rozdiel medzi nameranou a požadovanou veličinou.

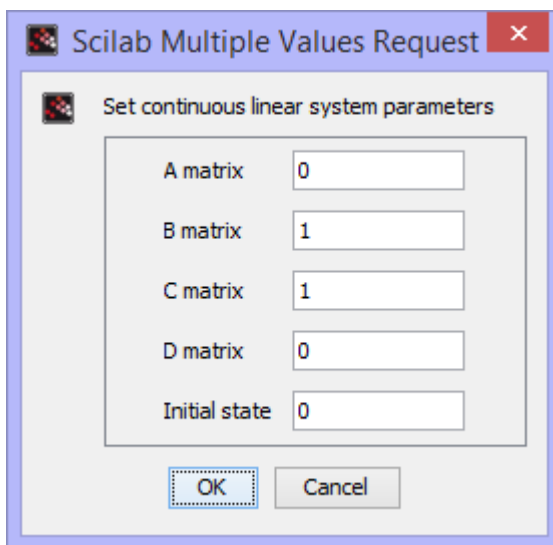


- **Proportional:** získaná hodnota ako násobok chyby.
- **Integral:** hodnota integrálu v čase chyby (1/Integral).
- **Derivation:** hodnota derivácie v čase chyby.

1.1.8 TCLSS

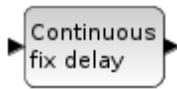


Tento blok realizuje stavový popis lineárneho systému s možnosťou skoku medzi stavmi. Blok obsahuje dva vstupné porty, kde prvý je regulárny vstup lineárneho systému. Druhý vstup nesie novú hodnotu stavu, ktorá sa skopíruje do stavu vo chvíli, keď prichádza udalosť na unikátnom vstupnom porte bloku. To znamená, že stav systému skočí na hodnotu prezentovanú na druhom vstupnom porte. Tento systém je definovaný pomocou matíc a počiatkovej podmienky. Rozmery jednotlivých matíc musia byť kompatibilné. Veľkosť vstupných a výstupných portov je nastavená automaticky.

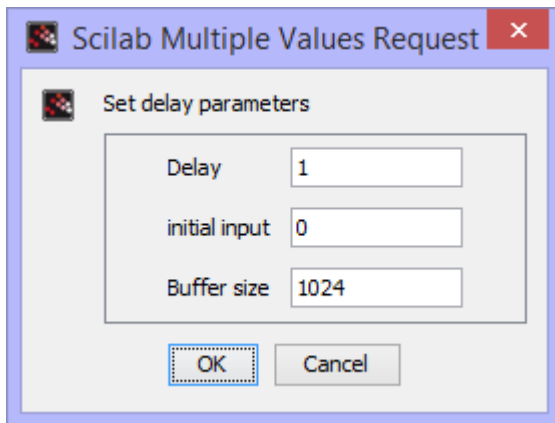


- **A matrix** (matica A): štvorcová matica.
- **B matrix** (matica B): ak systém nemá vstup, je matica prázdna "[]".
- **C matrix** (matica C): ak systém nemá výstup, je matica prázdna "[]".
- **D matrix** (matica D): ak systém nemá matematický člen D, je matica prázdna "[]".
- **Initial state** (počiatočný stav): vektor počiatkovej stavu systému.

1.1.9 TIME_DELAY

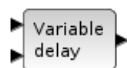


Blok zabezpečujúci meškanie hodnoty na vstupe o určitý čas. Používa na simuláciu časového oneskorenia. Počas pozastavenia sa nastaví hodnota, ktorá bude dočasne na výstupe.

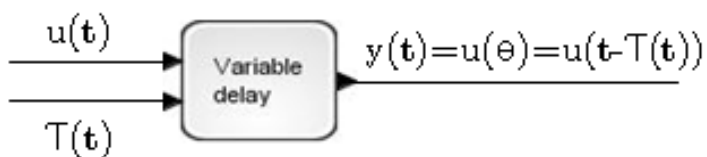


- **Delay:** hodnota, ktorá udáva dĺžku pozastavenia predtým ako sa vstupná hodnota začne šíriť na výstup.
- **Initial input:** hodnota, ktorá bude na výstupe počas pozastavenia.
- **Buffer size:** počiatočná alokácia v pamäti pre určitý počet hodnôt.

1.1.10 VARIABLE_DELAY



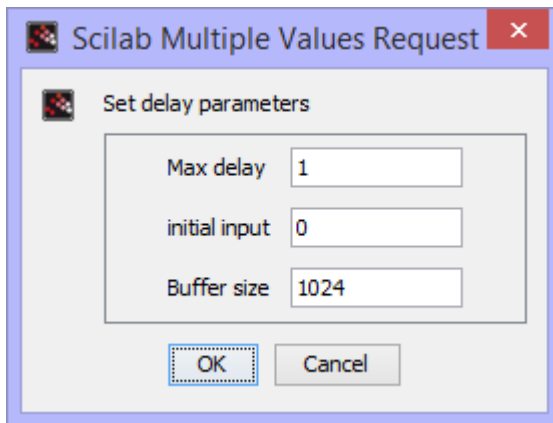
Blok môže byť využitý na simulovanie premenného časového oneskorenia medzi akciou a následným efektom. Blok tiež môže byť použitý napríklad na modelovanie systému s potrubím, kde rýchlosť pumpovania tekutiny v potrubí je premenlivá. Blok má dva vstupné a jeden výstupný port. Prvý vstup je signál $u(t)$, ktorý prechádza skrz blok a druhý vstup je premenná časového oneskorenia $\tau(t)$.



V danom časovom okamihu t , blok vysiela oneskorené dáta zo vstupu $u(\theta)$, kde θ je aktuálny simulačný čas t mínus časové omeškanie τ .

Príklad výstupného signálu môže vyzerat' nasledovne:

$$y(t) = y(\theta) - u(t - \tau(t))$$



- **Max delay:** je najvyššia možná hodnota, ktorú môže mať časové omeškanie. Hodnota nemôže byť záporná.
- **Initial input:** výstup generovaný blokom po dobu kým nedosiahne hodnotu časového omeškania.
- **Buffer size:** počiatková alokácia v pamäti pre určitý počet hodnôt.

1.1.11 PDE



Tento blok je implementáciou niekoľkých numerických schém (metóda konečných prvkov (1. a 2. rádu), metóda konečných diferencií (1. a 2. rádu), metóda konečných objemov (1. rádu)) na riešenie jednorozmerných PDE (Partial Differential Equation – parciálne diferenciálne rovnice) v rámci Xcos. Matematický priestor v rámci PDE sa obmedzuje maximálne na skalár 2. rádu v čase a priestore. Systém zvolí najúčinnjšie numerické schéma v závislosti od typu PDE a spustí riešiteľ'a.

PDE_IHM

File Tools Edit ? Documentation Quitter

PDE_IHM

Domain:
 a:
 b:

PDE block

PDE Expression: Display

<input type="checkbox"/> $\frac{\partial^2 u}{\partial t^2}$	<input type="text"/> a1(x)	<input type="text"/> b1(t) IN EDP1(t)	<input type="checkbox"/> $\frac{\partial u}{\partial x}$	<input type="text"/> a5(x)	<input type="text"/> b5(t) IN EDP5(t)
<input type="checkbox"/> $\frac{\partial^2 u}{\partial x^2}$	<input type="text"/> a2(x)	<input type="text"/> b2(t) IN EDP2(t)	<input type="checkbox"/> u	<input type="text"/> a6(x)	<input type="text"/> b6(t) IN EDP6(t)
<input type="checkbox"/> $\frac{\partial u}{\partial t}$	<input type="text"/> a3(x)	<input type="text"/> b3(t) IN EDP3(t)	<input type="checkbox"/> f	<input type="text"/> a7(x)	<input type="text"/> b7(t) IN EDP7(t)
<input type="checkbox"/> $\frac{\partial^2 u}{\partial t \partial x}$	<input type="text"/> a4(x)	<input type="text"/> b4(t) IN EDP4(t)			

Discriminant type:
 Constant discriminant
 Variable discriminant: positive
negative
null

Measurement points

u(xi): The measurement points

Add
Remove

Spatial discretization method:

Choice: Automatic Manual

Type: Finite difference Finite elements

Ordering: degree: Number of nodes:

Display step value

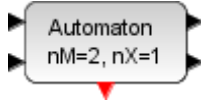
Initial conditions:
 $u(x, t_0) =$ $\frac{du}{dt}|_{t_0} =$

Validate
Cancel

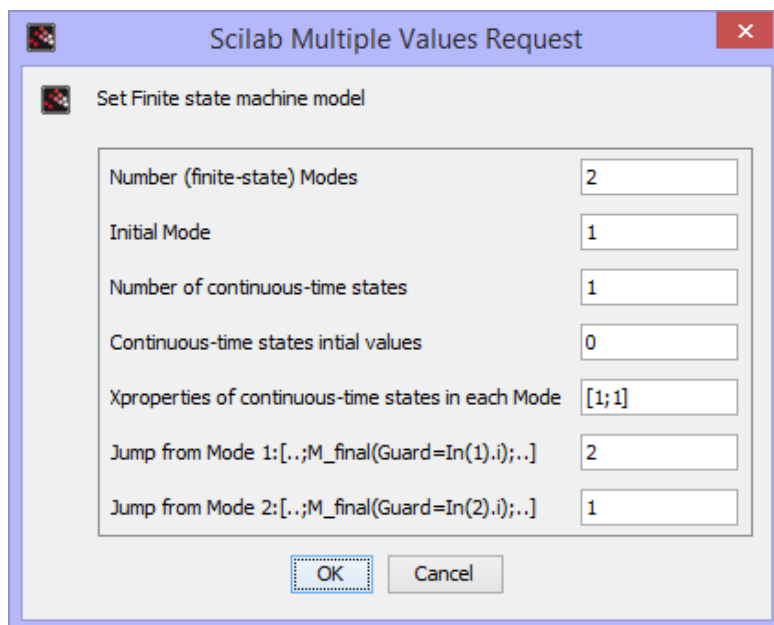
<p>Boundary condition on L: Type: Dirichlet Expression = <input type="text"/> IN_CL1(t)</p>	<p>Boundary condition on R: Type: Dirichlet Expression = <input type="text"/> IN_CL2(t)</p>
--	--

1.2 Discrete time systems

1.2.1 AUTOMAT



Tento blok umožňuje zostrojiť hybridný automat, tj. hybridný systém, ktorého diskretná časť je definovaná ako pomocou režimov a spojitá časť je definovaná pomocou diferenciálnych rovníc. Blok poskytuje spínací mechanizmus medzi subsystémov a priradeným riadiacim módom automatu.



Scilab Multiple Values Request

Set Finite state machine model

Number (finite-state) Modes	2
Initial Mode	1
Number of continuous-time states	1
Continuous-time states initial values	0
Xproperties of continuous-time states in each Mode	[1;1]
Jump from Mode 1:[.;;M_final(Guard=In(1).i);..]	2
Jump from Mode 2:[.;;M_final(Guard=In(2).i);..]	1

OK Cancel

- **Number (finite-state) Modes:** počet režimov v automatizácií.
- **Initial Mode:** režim aktivovaný na začiatku simulácie.
- **Number od continuous-time states:** počet spojitých stavov v režimoch. Počet spojitých stavov bude pre všetky režimy rovnaký.
- **Continuous-time states initial values:** počiatková hodnota spojitých stavov na začiatku simulácie.
- **Xproperties of continuous-time states in each Mode:** v tomto poli sú uvedené typy stavov v režime. Stav s indexom 1 môže byť diferenciálny alebo algebraický. Vektor je zapísaný ako matica $M * N$, kde M je počet režimov a N je počet stavov. Keď je v režime diferenciálny stav, (i,j) -ty prvok tejto matice by mal byť nastavený ako „+1“, v opačnom prípade ako „-1“.

- **Jump from Mode 1:[..;M_final(Guard=In(1).i);..]:** pole udáva informáciu o režime prechodu. Predpokladajme, že všetky riadiace režimy sú označené od 1 do M. Potom oblasť, ktorá zodpovedá riadiacemu režimu, je definovaná vo vektore. Potom j-ty prvok tohto vektora udáva cieľový režim, keď j-ta funkcia preskočí (stáva sa pozitívna).

1.2.2 DELAY_f

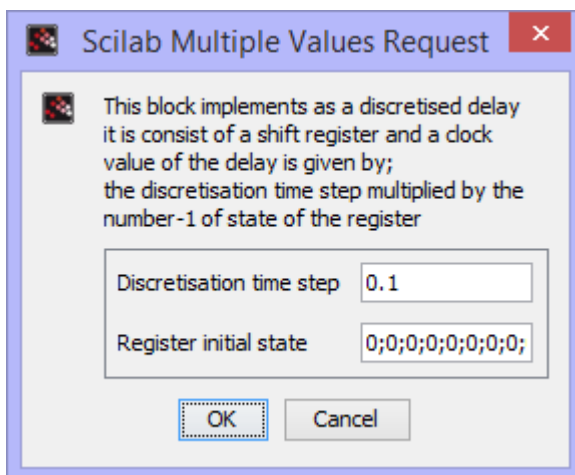


Tento blok implementuje diskkrétne oneskorenie. To sa prevádza pomocou posuvného registra a času. Hodnota omeškania je daná premennou „Discretisation time step“ a vynásobená číslom -1 stavu registra.

$$T_0 = \text{length}(\text{Register})$$

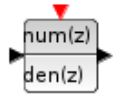
$$y(t) = \text{Register}(n) \quad \text{keď } t < nT_0$$

$$y(t) = x(n * (T - T_0)) \quad \text{keď } t \in [n; n + 1] * (T - T_0)$$

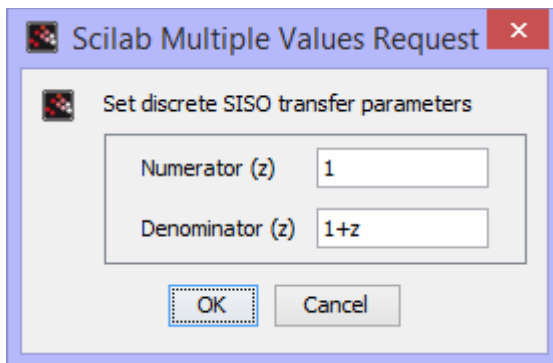


- **Discretization time step:** nastaví periódu integrovaných hodín.
- **Register initial state:** nastaví dĺžku a počiatočné podmienky registru.

1.2.3 DLR

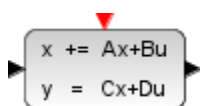


Blok realizujúci SISO lineárne systémy reprezentované prenosovou funkciou.

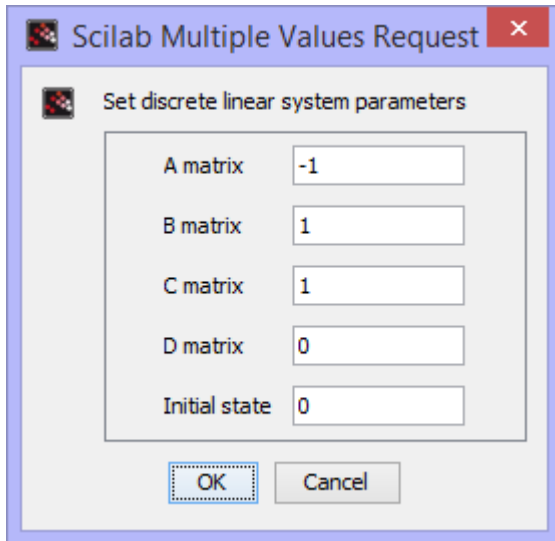


- **Numerator (z):** tento parameter nastavuje čitateľ prenosovej funkcie. Musí to byť polynóm.
- **Denominator (z):** tento parameter nastavuje menovateľ prenosovej funkcie. Musí to byť polynóm.

1.2.4 DLSS



Tento blok realizuje nespojitý lineárny systém cez stavový popis. Systém je definovaný pomocou matíc a počiatkovej podmienky. Rozmery musia byť kompatibilné. Pri príchode vstupnej udalosti na unikátny vstupný port je stav aktualizovaný.

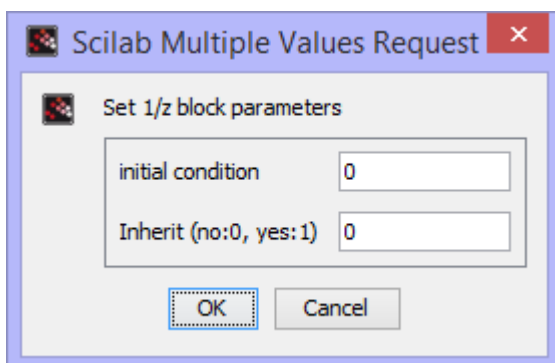


- **A matrix** (matica A): štvorcová matica.
- **B matrix** (matica B): ak systém nemá vstup, je matica prázdna "[]".
- **C matrix** (matica C): ak systém nemá výstup, je matica prázdna "[]".
- **D matrix** (matica D): ak systém nemá matematický člen D, je matica prázdna "[]".
- **Initial state** (počiatočný stav): vektor počiatočného stavu systému.

1.2.5 DOLLAR, DOLLAR_f, DOLLAR_m



Spomaľuje vstupný signál o dané obdobie. Vstup a výstup môže byť buď skalár alebo vektor.

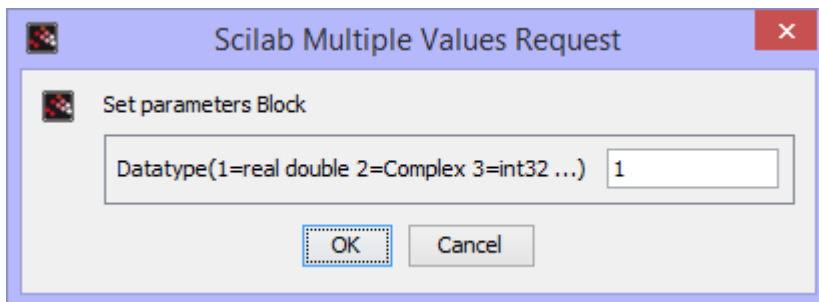


- **Initial condition:** doba o ktorú bude výstupný signál oneskorený oproti vstupnému.
- **Inherit:** ak je zadaná 1 (yes), tak blok dedí jeho vstup.

1.2.6 SAMPHOLD_m



Zakaždým, keď príde udalosť na unikátny vstupný port, blok skopíruje údaj zo vstupu na výstup. Tam ho podrží tak dlho kým na unikátny vstup nepríde ďalšia udalosť. Pre pravidelné vzorkovanie musí byť pripojený blok „Clock“.

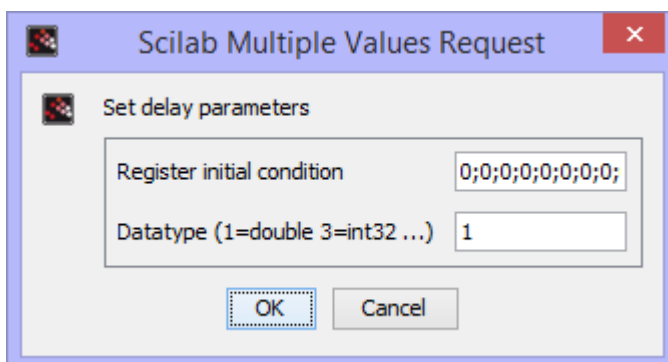


- **Datatype(1=real double 2=Complex 3=int32 ...)** : dátový typ na výstupe. Blok podporuje všetky dátové typy.

1.2.7 REGISTER



Tento blok realizuje posúvanie v registri. Zakaždým, keď príde do bloku nejaká udalosť, hodnota registru je posunutá o jeden krok.



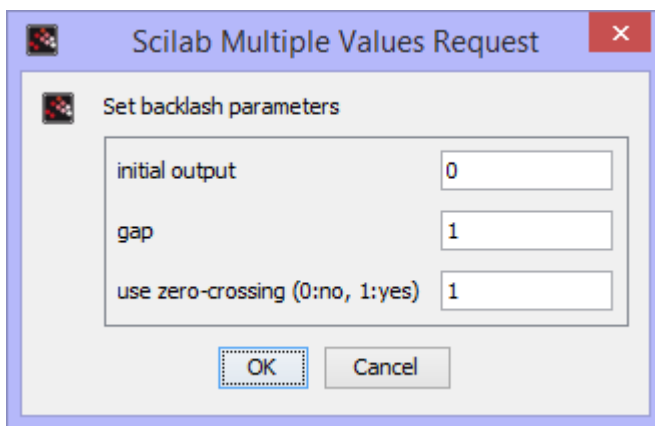
- **Register initial condition:** stĺpcový vektor. Obsahuje počiatočný stav registru.
- **Datatype:** dátový typ. Tento blok podporuje všetky dátové typy okrem komplexných čísel.

1.3 Discontinuities

1.3.1 BACKLASH



Blok zaznamenáva zmenu a vstupnom porte (zvyšovanie/znižovanie) a danú zmenu predáva na výstupný port. Keď sa na vstupe zmení smer, systém sa za moment odpojí (dochádza k hysterézii alebo medzere). Na začiatku vstup nemá žiadny vplyv na výstup preto ostáva konštantný.

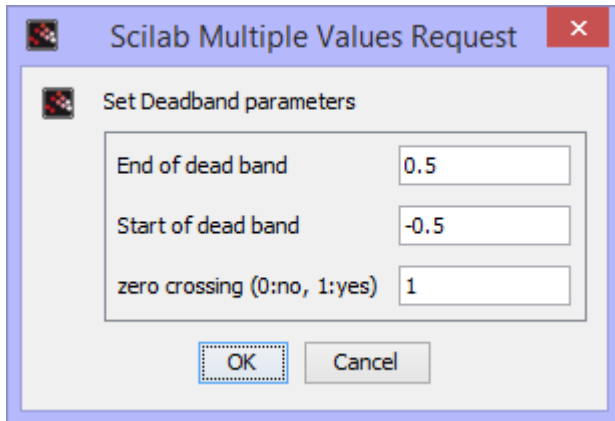


- **Initial output:** počiatočná výstupná hodnota
- **Gap:** šírka mŕtveho pásma
- **Use zero-crossing:** povolenie detekcie prechodu cez nulu na detekciu dolného a horného limitu.

1.3.2 DEADBAND



Blok DEADBAND poskytuje oblasť s nulovým výstupom. Je to oblasť na vstupe, pre ktorú ostáva výstup nezmenený. Mimo tejto oblasti je vzťah medzi mŕtvou zónou (oblasť necitlivosti) vstupu $u(k)$ a výstupu $v(k)$ čisto lineárny.

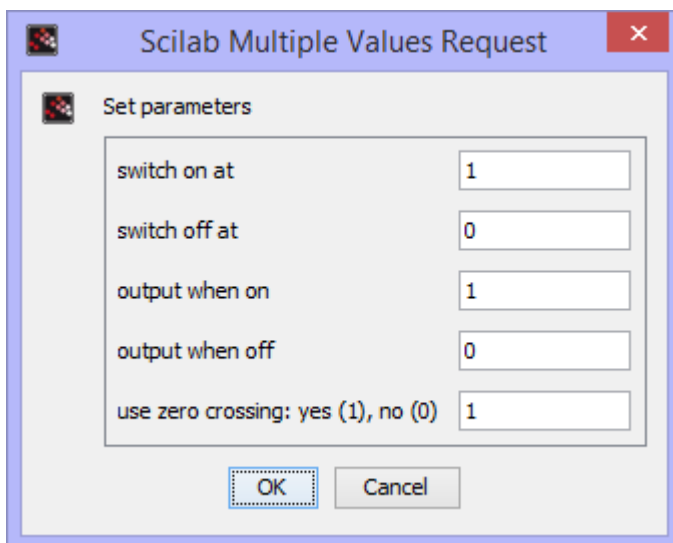


- **End of dead band:** horný limit mŕtveho pásma.
- **Start of dead band:** dolný limit mŕtveho pásma.
- **Zero crossing:** povolenie detekcie prechodu nulou, keď sú dosiahnuté zadané limity.

1.3.3 HYSTHERESIS

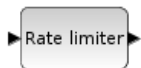


Blok umožňuje prepínanie medzi dvoma zadanými hodnotami na výstupe. Pracuje na princípe relé. Keď je spínač zapnutý (on) drží si svoj stav ak kým hodnota neklesne pod hodnotu vypnutia. Ak je spínač vypnutý (off), zostane vypnutý až kým hodnota vstupu neprekročí hranicu pre zapnutie.

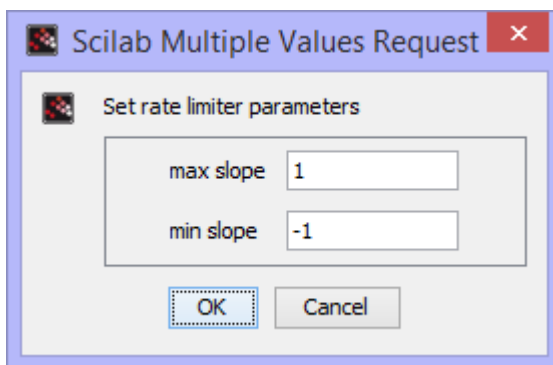


- **Switch on at:** hranica zapnutia spínaču. Daný parameter je prevedený na vstupný dátový typ v režime offline využitím metódy „round-to-nearest“ a saturácie.
- **Switch off at:** hranica vypnutia spínaču. Daný parameter je prevedený na vstupný dátový typ v režime offline využitím metódy „round-to-nearest“ a saturácie.
- **Output when on:** výstup keď je zapnutý.
- **Output when off:** výstup keď je vypnutý.
- **Use zero crossing: yes (1), no(0):** povolenie prechodu nulou na detekciu zapnutého/vypnutého stavu spínaču.

1.3.4 RATELIMITER



Obmedzovač prvej derivácie signálu, ktorý prechádza skrz blok. Zmeny na výstupe nie sú rýchlejšie než stanovený limit.

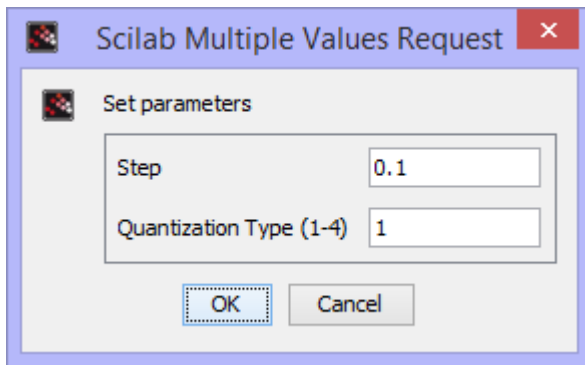


- **Max slope:** limit derivácie stúpajúceho vstupného signálu.
- **Min slope:** limit derivácie klesajúceho vstupného signálu.

1.3.5 QUANT_f



Výstup tohto bloku je vzorkovaný vstupný signál podľa voľby metódy.

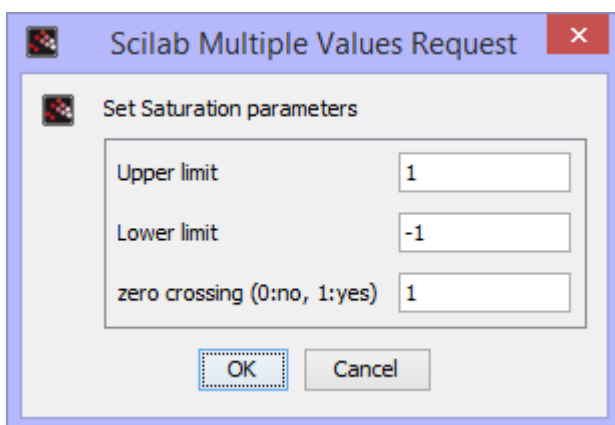


- **Step:** skalár, vzorkový krok
- **Quantization Type (1-4):** skalár s možnými hodnotami od 1 do 4.

1.3.6 SATURATION



Blok, ktorý orezáva vstupný signál medzi určenou hornou a dolnou hranicou. Ak signál nepresahuje dané medze, prechádza cez blok bez zmeny. Ak je horný aj dolný limit nastavený na rovnakú hodnotu, výstup z bloku bude práve táto hodnota.



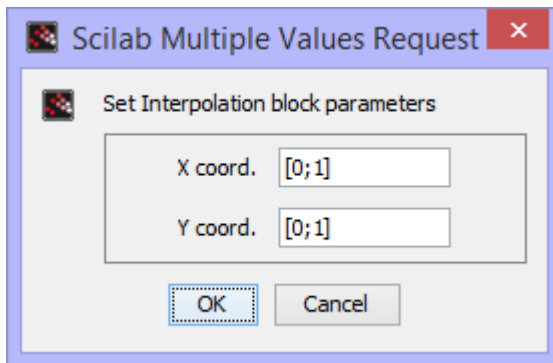
- **Upper limit:** špecifikovaná horná hranica vstupného signálu. Ak je hodnota vstupného signálu väčšia ako táto hranica, signál je na výstupe orezaný na danú hodnotu.
- **Lower limit:** špecifikovaná dolná hranica vstupného signálu. Ak je hodnota vstupného signálu menšia ako táto hranica, signál je na výstupe orezaný na danú hodnotu.
- **Zero crossing:** povolenie detekcie prechodu nulou.

1.4 Lookup Tables

1.4.1 INTRPLBLK_f

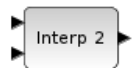


Výstupom tohto bloku je funkcia získaná lineárnou interpoláciou vstupu.

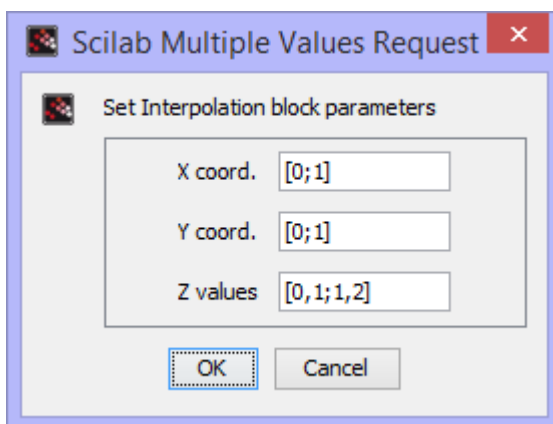


- **X coord.:** vektor, striktno rastúci.
- **Y coord.:** vektor, musí mať rovnaký rozmer ako X coord.

1.4.2 INTRP2BLK_f



Výstupom tohto bloku je funkcia získaná bilineárnou interpoláciou vstupu. Blok má dva skalárne vstupy a jeden skalárny výstup.



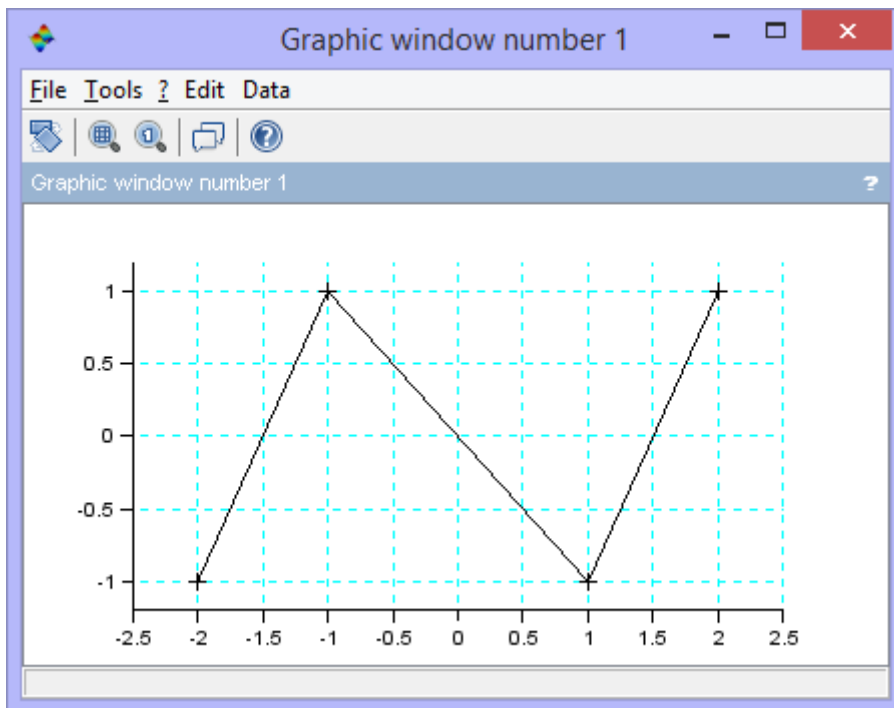
- **X coord.:** n-vektor, striktno rastúci.
- **Y coord.:** m-vektor, striktno rastúci.

- **Z values:** matica $m * n$.

1.4.3 LOOKUP_f



Blok realizujúci nelineárne funkcie, ktoré sú definované užívateľom v grafickom editore.

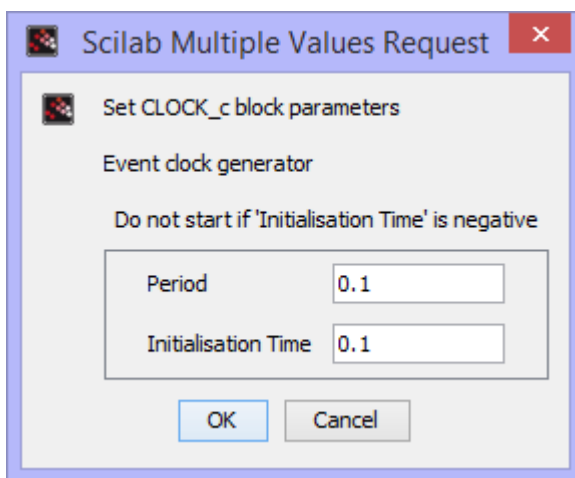


1.5 Event handling

1.5.1 CLOCK_c



Unikátny výstup bloku generujúci pravidelný sled udalostí, ktoré sú plánované parametrom Period (v sekundách). Čas začiatku možno nastaviť cez parameter Initialisation Time (v sekundách).



- **Period:** čas, ktorý oddeľuje dve výstupné udalosti. Je to inverzná frekvencia času.
- **Initialization Time:** čas, kedy sa začne udalosť. Hodnota nesmie byť záporná, pretože by operácia nikdy nezačala.

1.5.2 SampleCLK



Rozdiel medzi SampleCLK a CLOCK_c je, že všetky bloky SampleCLK využívané v diagramoch sú synchronné. Synchronizácia sa vykonáva z dôvodu dvoch rôznych metód výpočtu vo fáze kompilácie.

Prvá metóda spočíva vo výpočte hodín, ktorá je rýchlejšia ako všetky SampleCLK pripojenia na register, ktorý aktivuje udalosť vybraného bloku.

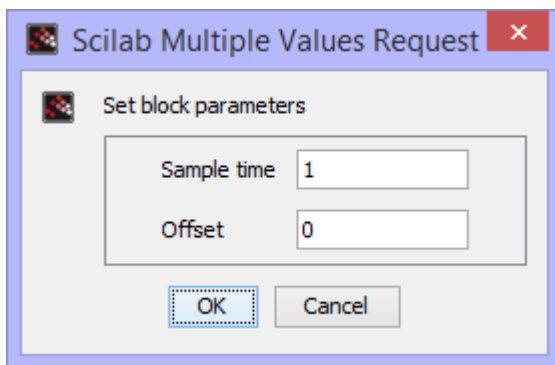
Hodiny sú počítané na základe nasledovného pravidla:

Ak majú všetky bloky rovnaký ofset, potom frekvencia hodín je najväčším spoločným deliteľom zo vzorkovacích časov, a ofset hodín je rovný ofsetu.

Ak sú ofsety rôzne, potom je frekvencia hodín najväčším spoločným deliteľom vzorkovacích časov a ofset, a ofset hodín je rovný 0.

Počítadlo počíta od 1 do najmenšieho spoločného násobku (NSN) vzorkovacieho času.

Druhá metóda využíva multifrekvenčný blok na generovanie udalosti len na určitý čas. Udalosti v tejto metóde nie sú periodicky generované ako v prvom prípade.



- **Sample time:** hodnota vzorkovacieho času.
- **Offset:** hodnota ofsetu.

1.5.3 VirtualCLK0



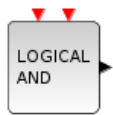
Tento blok je považovaný za virtuálny blok kompilátoru. Je využívaný v Super bloku (sub-systém) na spúšťanie vždy aktívnych blokov a v pod úrovni. Bloky sú prakticky pripojené na vstup tohto bloku.

1.5.4 ANDBLK



Daný blok kombinuje skupinu signálov reprezentovaných jedným riadkom blokového diagramu. To umožňuje zníženie počtu riadkov pre smerovanie signálov z jednej časti diagramu do inej. Robí ho prehľadnejším.

1.5.5 ANDLOG_f



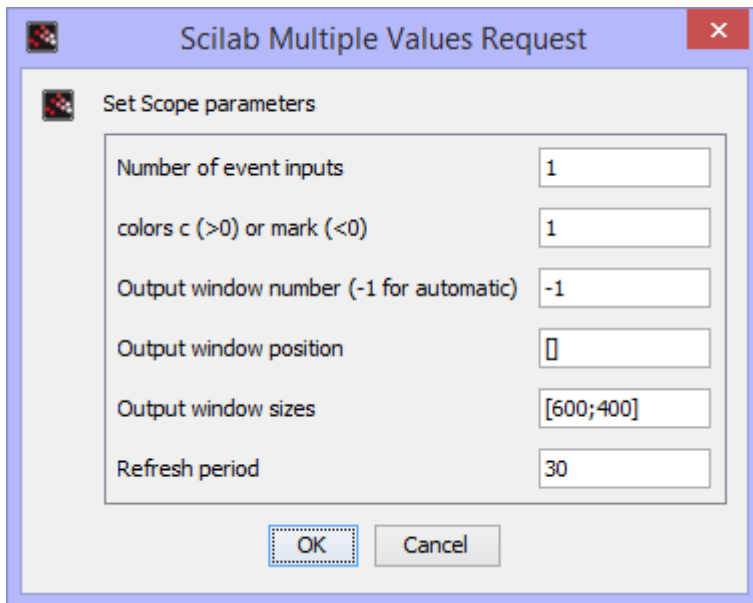
Tento blok obsahuje dva unikátne vstupné porty a jeden výstupný port. Výstup nadobúda hodnoty +1 alebo -1 v závislosti od vstupných udalostí.

- +1: ak sú na oboch vstupných portoch prítomné udalosti.
- -1: ak je dostupná len jedna udalosť.

1.5.6 CEVENTSCOPE



Daný blok realizuje vizualizáciu signálov vstupných udalostí.

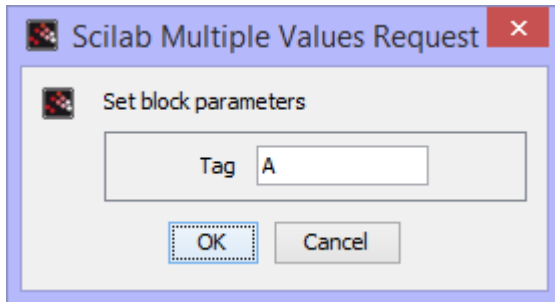


- **Number of event inputs:** celé číslo, ktoré udáva počet vstupným portov farieb.
- **Colors c (>0) or mark (<0):** celé číslo. Ku každému číslu je priradená určitá farba.
- **Output window number (-1 for automatic):** udáva počet grafických okien používaných na zobrazenie. To sa často využíva pre zobrazenie vysokých hodnôt, aby nedochádzalo ku konfliktom s paletami a oknami Super bloku.
- **Output window position:** vektor s 2 hodnotami, ktorý špecifikuje súradnice horného ľavého rohu grafického okna.
- **Output window sizes:** vektor s 2 hodnotami, ktorý špecifikuje výšku a šírku grafického okna.
- **Refresh period:** maximálna hodnota na ose X (čas). Grafický výstup sa prekreslí vždy, keď čas dosiahne násobok tejto hodnoty.

1.5.7 CLKFROM



Tento blok je používaný na pripájanie unikátnych portov medzi subsystémami. V blokoch CLKFROM a CLKGOTO musí byť rovnaké označenie aby fungovala komunikácia.

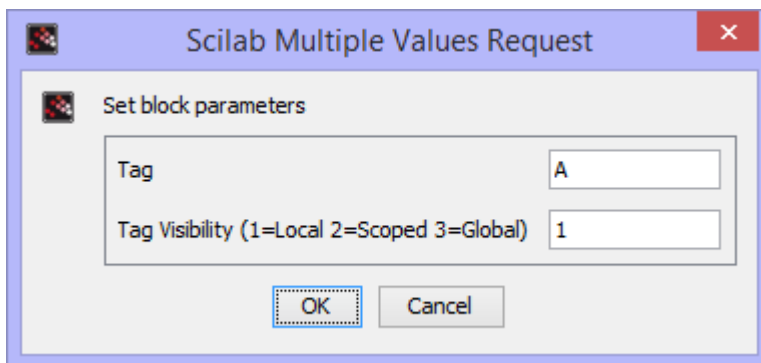


- **Tag:** označenie výstupu z bloku.

1.5.8 CLKGOTO



Tento blok je používaný na pripájanie unikátnych portov medzi subsystémami. V blokoch CLKFROM a CLKGOTO musí byť rovnaké označenie aby fungovala komunikácia.

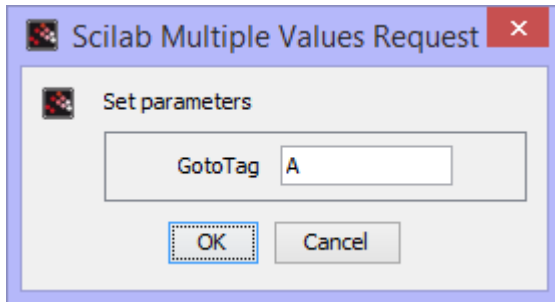


- **Tag:** označenie vstupu na spárovanie s blokom CLKFROM.
- **Tag Visibility:** určuje oblasť do ktorej môže blok zasahovať. To môže byť local (1), scoped(2), global(3).

1.5.9 CLKGoto TagVisibility



Blok je využívaný v prípade, že existuje nejaká udalosť.



- **GotoTag**: je značka bloku Goto, ktorého umiestnenie je definované v tomto bloku.

1.5.10 CLKSOMV_f

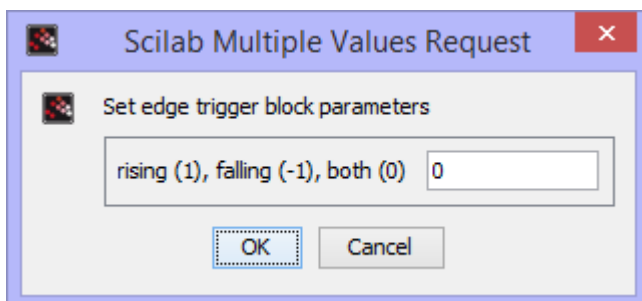


Je to doplnkový blok pre udalosti s tromi vstupnými portami. Výstup reprezentuje udalosti zo všetkých vstupných portov. Presnejšie povedané, daný blok nie je blokom Xcos, pretože je zanedbávaný vo fáze kompilácie. Vstupy a výstupy sú synchronizované.

1.5.11 EDGE_TRIGGER



Blok generuje udalosti na základe vzostupnej, zostupnej alebo na obe hrany vstupného signálu (v závislosti parametru bloku).

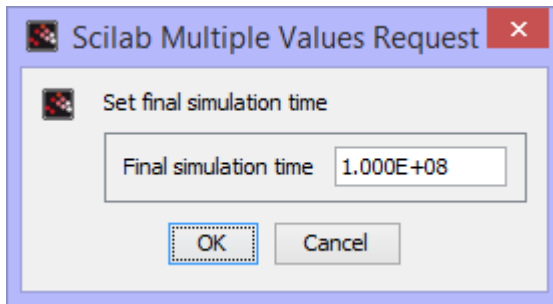


- **rising (1), falling (-1), both (0)**: nastaví hodnotu na ktorú má blok reagovať. Vzostupná hrana (1), zostupná hrana (-1), obe súčasne (0).

1.5.12 ENDBLK



Používa sa na nastavenie času, kedy má byť ukončená simulácia. Parameter môže byť numerická hodnota alebo symbolická premenná definovaná v rámci Scilabu.

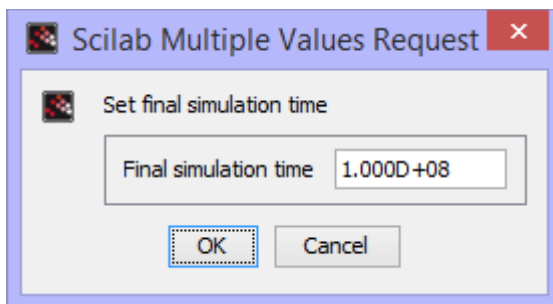


- **Final simulation time:** nastaví konečný simulácie.

1.5.13 END_c



Tento blok postupne zvyšuje aktuálny čas na finálny čas simulácie.

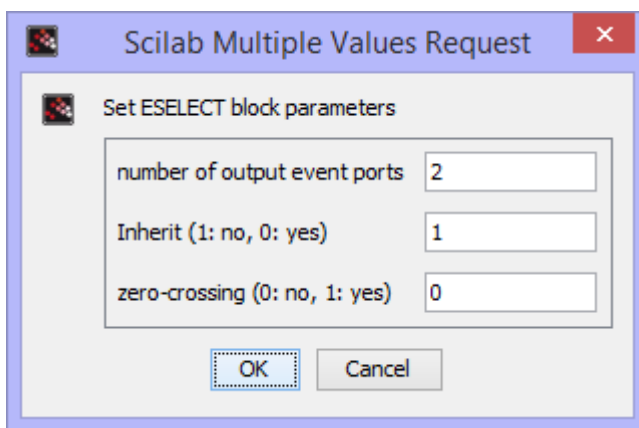


- **Final simulation time:** parameter dátumu pre počiatočné výstupné udalosti. Využitím spätnej väzby z výstupu na vstup môže blok sám ukončiť simuláciu v definovanom čase.

1.5.14 ESELECT_f



Špeciálny blok podobný If_Then_Else. Vstupy a výstupy sú synchronizované. Prichádzajúca udalosť je presmerovaná na jeden z výstupných portov na základe hodnoty pravidelného vstupu. Napríklad, ak je vstupná hodnota medzi 0 a 1, riadiaci vstup je presmerovaný do prvého výstupu. Ak je hodnota vstupu medzi 1 a 2, riadiaci vstup je presmerovaný do druhého výstupu atď.



- **Number of output event ports:** skalár udávajúci počet výstupných portov.
- **Inherit (1: no, 0: yes):** ak je zadané „no“, tak udalosť sa dedí z unikátneho vstupného portu. Ak je zadané „yes“, potom je udalosť aktivovaná pravidelne vstupným portom.
- **Zero-crossing (0: no, 1: yes):** povolenie detekcie prechodu nulou.

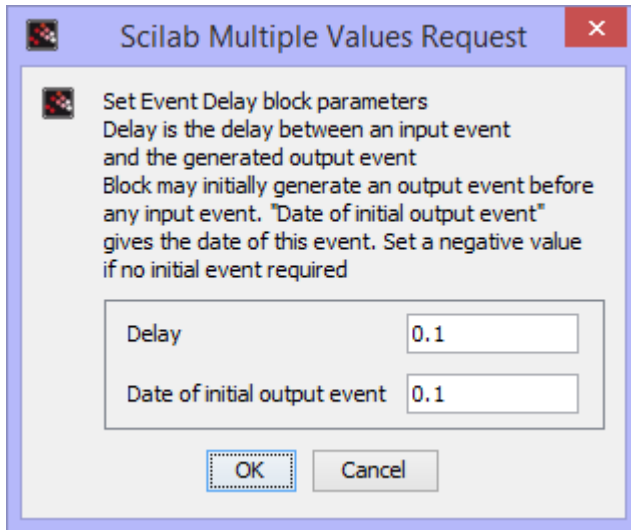
1.5.15 EVTDLY_c



Udalosť je generovaná v časovom odstupom podľa parametru „Delay“. Blok tiež môže generovať počiatočnú výstupnú udalosť. Dátum udalosti je počítaný podľa vzťahu:

$$t_i = t_{init} + i * T_{delay},$$

kde t_{init} je dátum počiatočnej výstupnej udalosti, T_{delay} je omeškanie a i je celočíselné diskkrétne počítadlo.

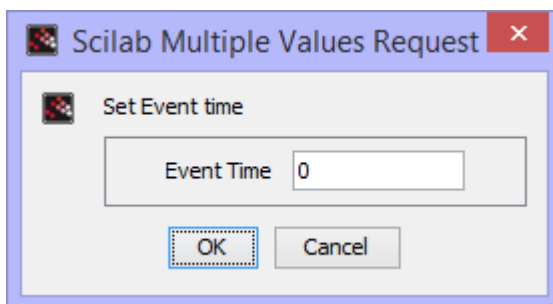


- **Delay:** skalár, ktorý udáva časové omeškanie medzi vstupom a výstupom.
- **Date of initial output event:** skalár, udávajúci čas generovania počiatkovej výstupnej udalosti.

1.5.16 EVTGEN_f



Udalosť je generovaná na unikátnom výstupnom porte ak je hodnota „Event Time“ väčšia alebo rovná 0. Ak je menšia ako 0, udalosť nebude generovaná.

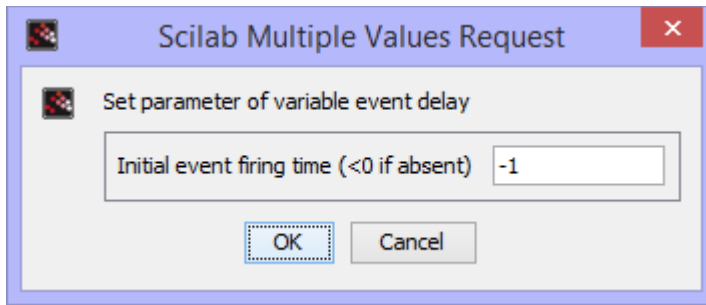


- **Event Time:** skalár udávajúci čas počiatkovej udalosti.

1.5.17 EVTWARDLY



Udalosť je generovaná s omeškaním oproti času kedy prišla udalosť na unikátny vstupný port. Blok tiež môže generovať počiatočnú udalosť na výstupnom porte.

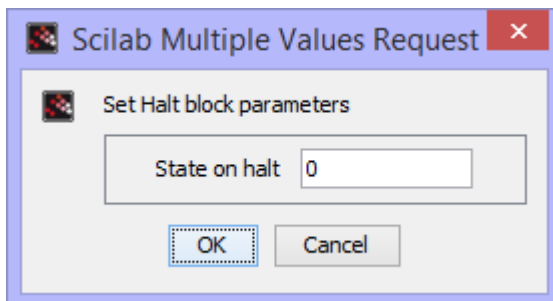


- **Initial event firing time (<0 if absent):** udalosť je generovaná na unikátnom výstupnom porte ak je daný parameter väčší alebo rovný 0. Ak je menší, tak udalosť nebude generovaná.

1.5.18 HALT_f

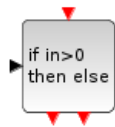


Blok má unikátny vstupný port. Po príchode udalosti na tento blok sa simulácia zastaví a aktivuje sa hlavné okno Xcos. Simulácia môže byť reštartovaná alebo obnovená (tlačidlo „Run“).

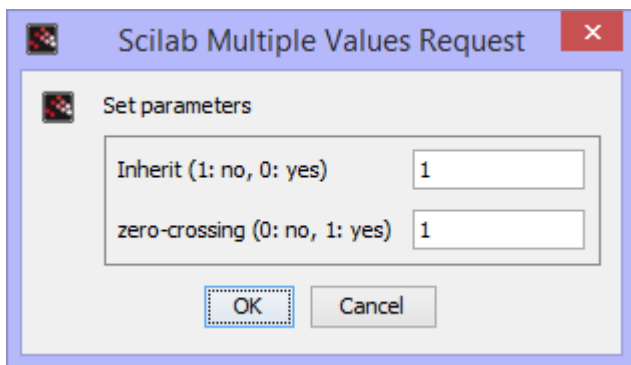


- **State on halt:** skalárna hodnota, ktorá sa využíva pri ladení. Umožňuje rozlišovanie medzi rôznymi zastaveniami (Halt).

1.5.19 IFTHEL_f



Udalosť je generovaná na unikátnom výstupnom porte keď príde udalosť na unikátny vstupný port. V závislosti na prichádzajúcej hodnote je výstup generovaný buď na prvom alebo druhom výstupnom porte. Vstupy a výstupy sú synchronizované.



- **Inherit (1: no, 0:yes):** ak je zadané „no“, tak udalosť sa dedí z unikátneho vstupného portu. Ak je zadané „yes“, potom je udalosť aktivovaná pravidelne vstupným portom.
- **Zero-crossing (0: no, 1:yes):** povolenie detekcie prechodu nulou.

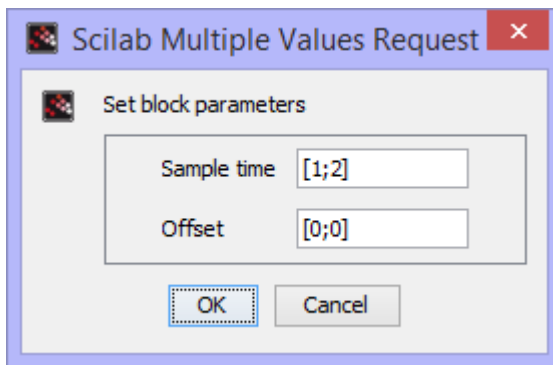
1.5.20 M_freq



Tento blok generuje udalosť na určitú časovú dobu pri danom simulačnom čase. Doba vzorky je uvedená v poli „Sample Time“ a offset je v poli „Offset“. Blok má jeden vstup na organizáciu akcií, počet výstupov závisí od počtu rôznych vzorkovacích časov. Napríklad v prípade, že vzorkovací čas je [1 1 2] a offset je [0 .5 0] potom má blok 7 výstupov.

- Prvý výstup je aktivovaný keď je simulačný čas zhodný s násobkom prvej vzorky času plus prvý offset.
- Druhý výstup je aktivovaný keď je simulačný čas zhodný s násobkom druhej vzorky času plus druhý offset.
- Tretí výstup je aktivovaný keď sú aktívne oba prípady (prvý a druhý).

- Štvrtý výstup je aktivovaný keď simulačný čas je zhodný s násobkom tretej vzorky času plus tretí ofset.
- Piaty výstup je aktivovaný keď je aktívny prvý a štvrtý prípad súčasne.
- Šiesty výstup je aktivovaný keď je aktívny druhý a štvrtý prípad súčasne.
- Siedmy výstup je aktivovaný keď je aktívny tretí a štvrtý prípad súčasne.

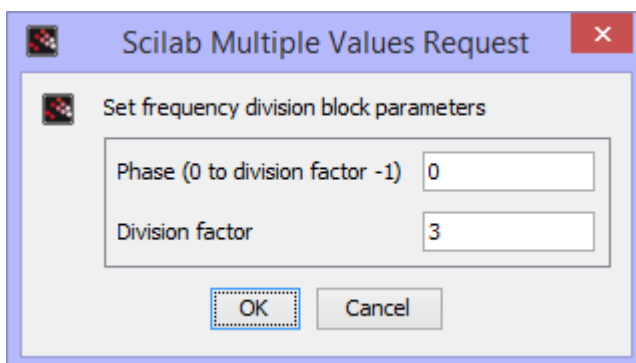


- **Sample time:** vektor vzorkovacieho času.
- **Offset:** vektor hodnoty ofsetu. Musí mať rovnakú veľkosť ako „Sample time“ a zároveň každá hodnota daného vektoru musí byť menšia ako zodpovedajúca hodnota vektoru vzorkovacieho času.

1.5.21 freq_div



Tento blok je Super blok. Vstupná udalosť je posielaná na výstup n-krát za určitý čas.



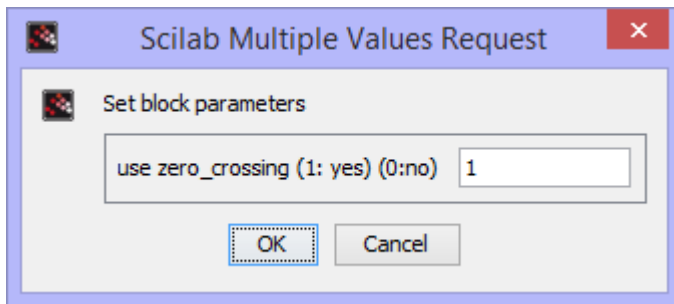
- **Phase (0 to division factor -1):** kladný skalár.
- **Division factor:** celé číslo väčšie ako 1.

1.6 Mathematical Operations

1.6.1 ABS_VALUE



Vypočíta absolútnu hodnotu prvkov vstupného vektoru.

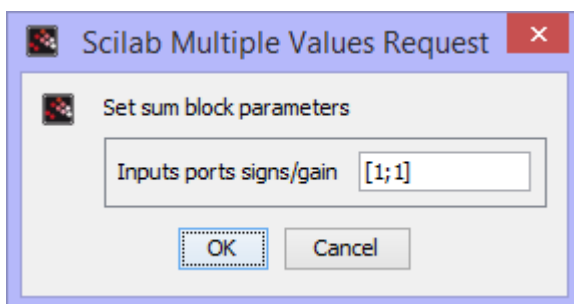


- **Use zero_crossing (1: yes) (0: no):** povolenie detekcie prechodu nulou.

1.6.2 BIGSOM_f



Blok vykonáva sčítanie alebo odčítanie skalárnych alebo vektorových vstupov. Užívateľ môže nastaviť zosilnenie pre každý vstup s parametrom „Inputs ports signs/gain“.



- **Inputs ports signs/gain:** nastaví zosilnenie pre každý vstupný signál.

1.6.3 COSBLK_f

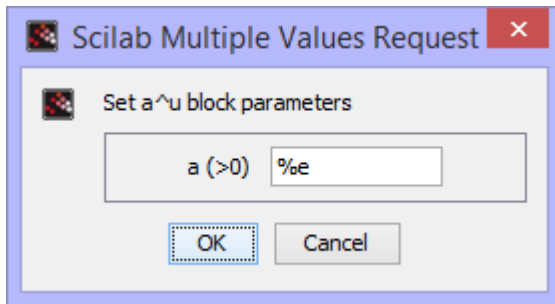


Počíta kosínus prvkov vstupného vektora.

1.6.4 EXPBLK_m



Výstup tohto bloku je vektor $y(i)=a^{u(i)}$, kde parameter a je kladný skalár a u je vstupný vektor. Veľkosti vstupných a výstupných portov sú určené kompilátorom.

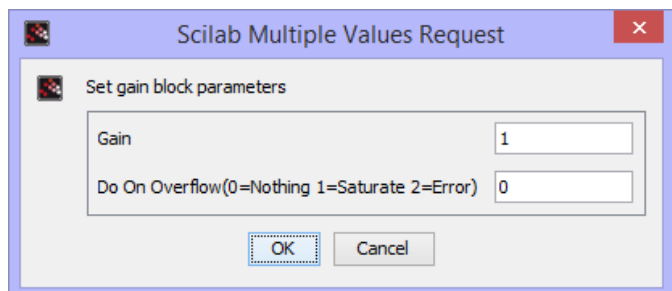
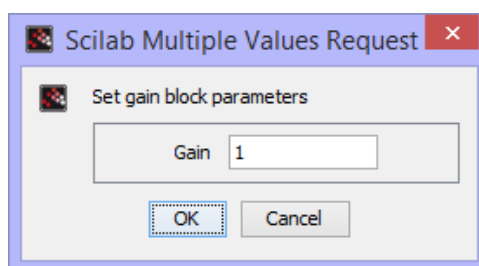


- **a (>0)**: reálny kladný skalár.

1.6.5 GAINBLK, GAINBLK_f, GAIN



Bloky počítajú štvorcové matice A podľa vstupného vektora U , kde počet riadkov/stĺpcov sa rovná počtu riadkov vektora U . Matica A je nastavovaná parametrom „Gain“.



- **Gain:** parameter definuje štvorcovú maticu A.
- **Do On Overflow (0=Nothing, 1=Saturate, 2=Error):** keď nastane pretečenie môžu nastať tri stavy:
 - 0: nič sa nedeje, výsledok nie je nasýtený,
 - 1: výsledok je nasýtený.
 - 2: simulácia ohlási chybové hlásenie.

1.6.6 INVBLK

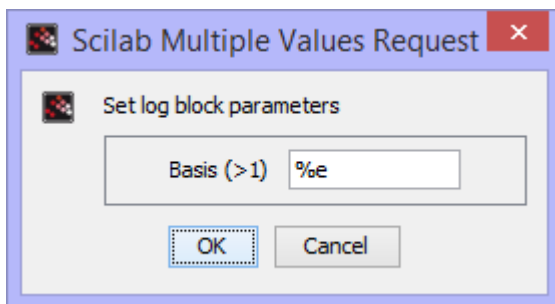


Tento blok počíta výstupný vektor $y(i) = 1/u(i)$, kde u je vstupný vektor. Veľkosť vstupu a výstupu sa určuje podľa kontextu.

1.6.7 LOGBLK_f



Počíta logaritmus zo vstupných prvkov vektoru. Užívateľ môže opraviť základ cez parameter „Basis“. V predvolenom nastavení blok vypočíta prirodzený logaritmus. Veľkosť vstupného a výstupného portu je určená podľa kontextu.



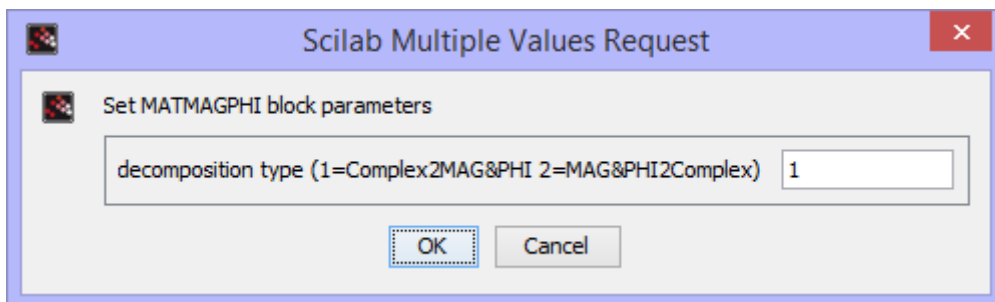
- **Basis (>1):** reálny skalár väčší ako 1.

1.6.8 MATMAGPHI



Daný blok vypočíta veľkosť a uhol matice komplexných čísel alebo poskladá maticu komplexných čísel z matice veľkostí a matice uhlov s ohľadom na hodnotu parametra „decomposition type“. Nasledujúca tabuľka popisuje správanie bloku:

Typ prevodu	Vstupy (dátový typ)	Výstupy (dátový typ)	Operácia
1	Jeden (complex alebo double)	Dva (double)	Konvertuje komplexné číslo na veľkosť (1. výstup) a uhol v radiánoch (2. výstup). Ak je vstup typu double, uhol bude nulový alebo PI a veľkosť bude rovná absolútnej hodnote vstupu.
2	Dva (double)	Jeden (complex)	Výstup z bloku je matica komplexných čísel pozostávajúca z veľkosti a uhlu na vstupoch respektíve z prvého a druhého vstupu.



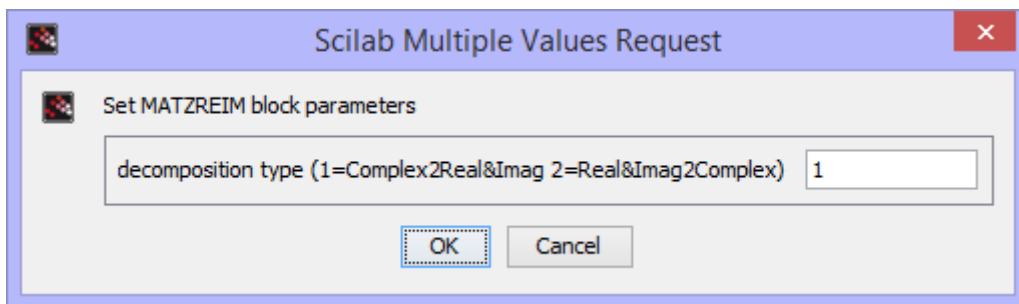
- **decomposition type (1=Complex2MAG&PHI 2=MAG&PHI2Complex):** nastavuje typ konvertovania.

1.6.9 MATZREIM



Blok rozloží maticu komplexných čísel na reálu a imaginárnu časť alebo poskladá maticu komplexných čísel spojením dvoch častí na základe parametra „decomposition type“. Nasledujúca tabuľka popisuje správanie bloku:

Typ prevodu	Vstupy (dátový typ)	Výstupy (dátový typ)	Operácia
1	Jeden (complex)	Dva (double)	Rozklad: reálna a imaginárna časť vstupu sú budú k dispozícii v tomto poradí na prvom a druhom výstupnom porte.
2	Dva (double)	Jeden (complex)	Zloženie: výstupom je matica komplexných čísel zložená z reálnych a imaginárnych častí, ktoré prichádzajú z prvého a druhého vstupu.



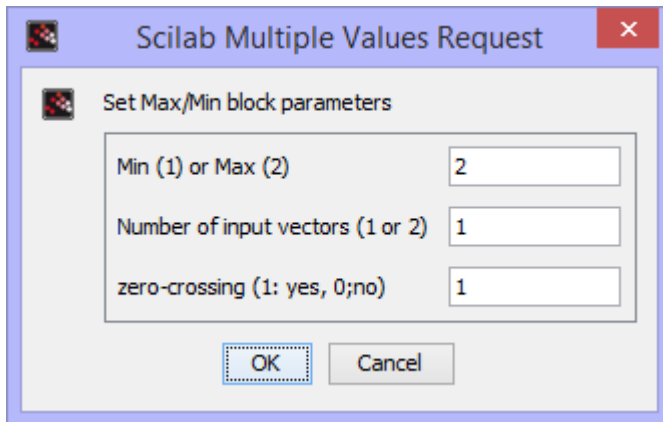
- **decomposition type (1=Complex2Real&Imag 2=Real&Imag2Complex):** označuje použitý typ konverzie.

1.6.10 MAXMIN



Blok vyhľadá minimálnu/maximálnu hodnotu a prijíma jeden alebo dva vstupy na základe parametra „Number of input vectors“. Pri jednom vstupe musí byť vstupná hodnota vektor.

Pri dvoch vstupoch to môžu byť buď skaláry alebo vektory. Ak ide o vektory tak musia mať rovnakú veľkosť.



- **Min (1) or Max (2):** funkcia aplikovaná na výstup.
- **Number of input vectors (1 or 2):** udáva počet vstupov bloku.
- **Zero-crossing (1: yes, 0: no):** aktivácia detekcie prechodu nulou pri detekcii minimálnej a maximálnej hodnoty.

1.6.11 MAX_f



Blok nájde prvok s najväčšou hodnotou zo vstupného vektoru.

1.6.12 MIN_f

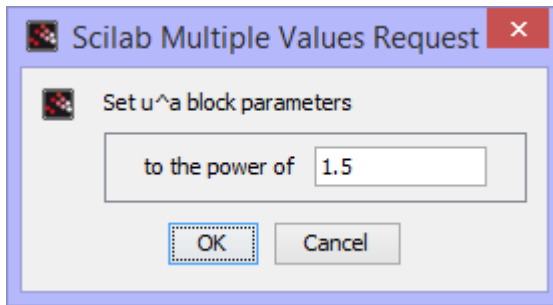


Blok nájde prvok s najmenšou hodnotou zo vstupného vektoru.

1.6.13 POWBLK_f



Počíta výstupný vektor $y(i)=u(i)^a$, kde a je reálne číslo dané parametrom „to the power of“ a u je vstup. Veľkosť vstupného a výstupného portu sú určené prekladačom podľa veľkostí pripojených blokov.

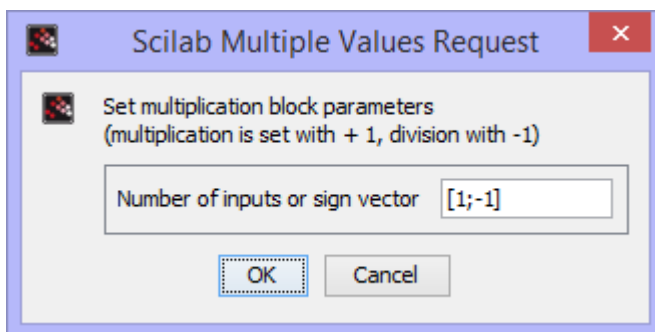


- **To the power of:** reálne číslo udávajúce hodnotu exponentu.

1.6.14 PRODUCT



Tento blok počíta násobenie alebo delenie svojich vektorových vstupov.



- **Number of inputs or sign vector:** vektor udávajúci počet a typ vstupov. Násobenie (1), delenie (-1).

1.6.15 PROD_f



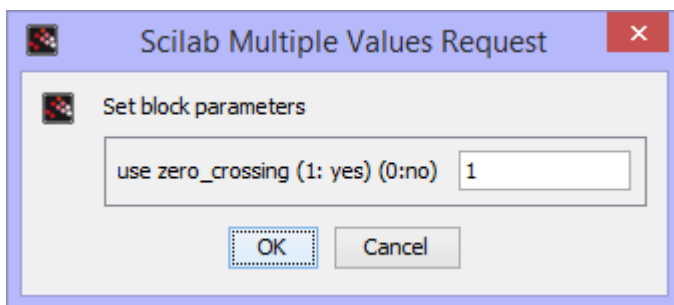
Výstupom tohto bloku je spoločný vektor zložený z dvoch vstupov.

1.6.16 SIGNUM



Výstup bloku je vektor príznakových prvkov zo vstupného vektora. Každému prvku vstupného vektora odpovedá prvok výstupného vektora:

- 1 keď je prvok väčší ako nula,
- 0 keď je prvok rovný nule,
- -1 keď je prvok menší ako nula.



- **Use zero_crossing (1: yes) (0: no):** povolenie detekcie prechodu nulou.

1.6.17 SINBLK_f

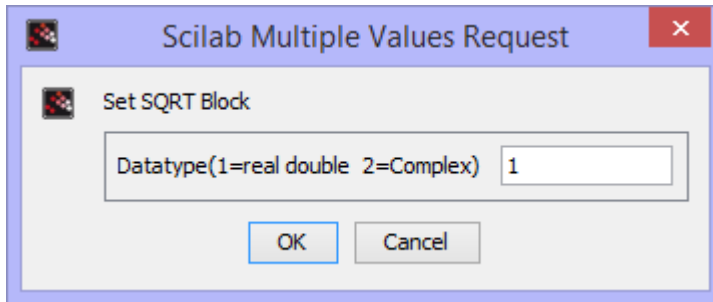


Blok počíta funkciu sínus z prvkov vstupného vektora.

1.6.18 SQRT



Daný blok počíta druhú odmocninu každého prvku vstupnej matice.

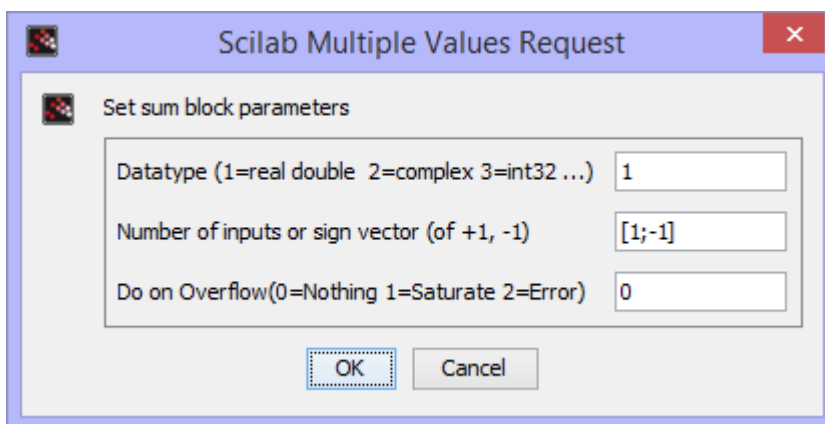


- **Datatype(1=real double 2=Complex):** udáva dátový typ výstupu. Podporuje iba dva dátové typy double (1) a complex (2).

1.6.19 SUMMATION



Tento blok vykonáva sčítavanie alebo odčítavanie skalárnych, vektorových alebo maticových vstupov. Vektory/matice musia mať rovnaký rozmer.



- **Datatype (1=real double 2=comple 3=int32...):** nastaví dátový typ vstupu/výstupu. Podporuje všetky dátové typy, ale musia byť v rozsahu 1 až 8.
- **Number of inputs or sign vector (of +1, -1):** nastaví počet a typ vstupov.
- **Do on Overflow (0=Nothing 1=Saturate 2=Error):** ak je nastavená 0, výsledok je podobný normálnemu súčtu dvoch celočíselných matic. Ak je nastavená 1, tak v prípade pretečenia nastaví výsledok na hodnotu nasýtenia. Ak je nastavená 2, tak v prípade pretečenia zobrazí okno s chybovým hlásením. Ak je dátový typ double alebo complex, tento parameter nie je použitý.

1.6.20 SUM_f



Blok môže sčítavať tri vstupy do jedného. Podporuje skalárny alebo vektorový vstup.

1.6.21 TANBLK_f

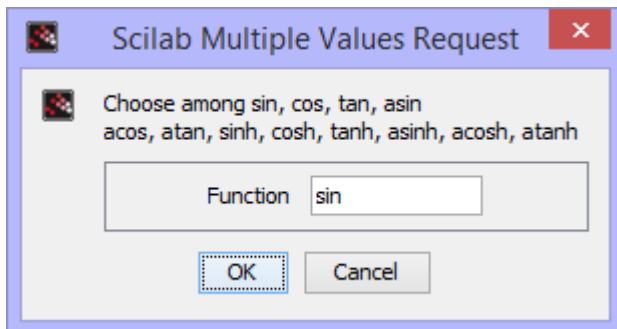


Blok počíta funkciu tangens z prvkov vstupného vektora.

1.6.22 TrigFun



Funkčný blok vykonáva mnoho bežných goniometrických funkcií. Podporuje nasledujúce funkcie: $\sin()$, $\cos()$, $\tan()$, $\text{asin}()$, $\text{acos}()$, $\text{atan}()$, $\text{atan2}()$, $\text{sinh}()$, $\text{cosh}()$, $\text{tanh}()$. Vstup do bloku môže byť skalár alebo vektor.



- **Function:** udáva určitú goniometrickú funkciu (\sin , \cos , \tan , asin , acos , atan , atan2 , sinh , cosh , tanh).

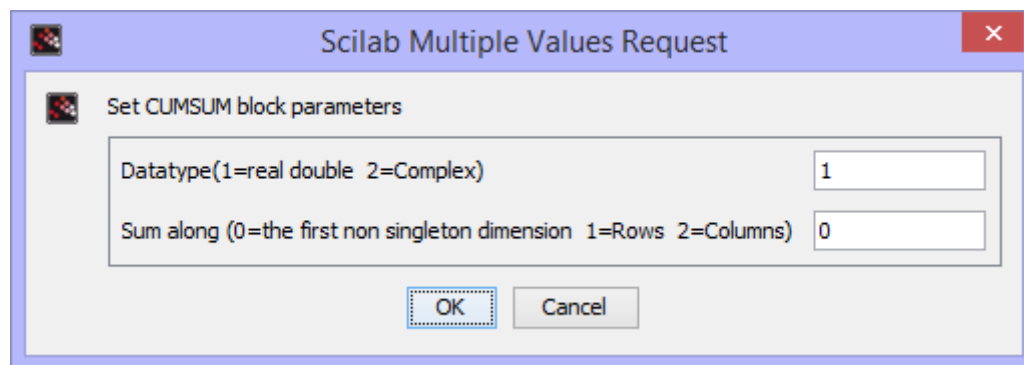
1.7 Matrix

1.7.1 CUMSUM



Blok sčítava vstupnú maticu U o rozmeroch $M \times N$ buď po riadkoch, stĺpcoch alebo cez prvý nejednotkový rozmer. Keď je parameter „Sum along“ nastavený na:

- 1: výstupom bude matica $M \times 1$,
- 2: výstupom bude matica $1 \times N$,
- 0: výstupom bude skalár.

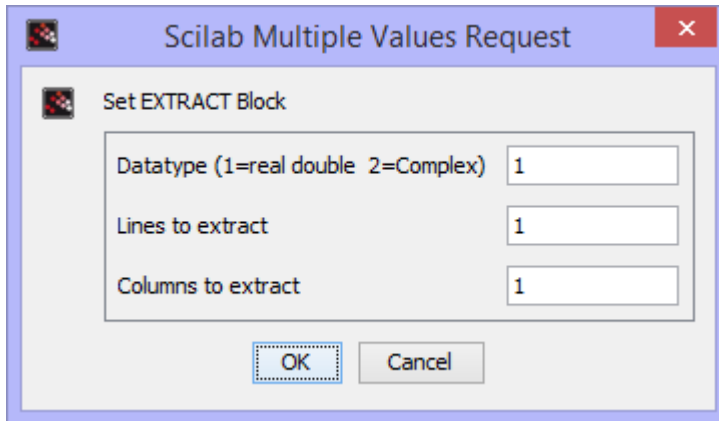


- **Datatype(1=real double 2=Complex):** udáva dátový typ výstupu. Podporované sú dátové typy double (1) a complex (2).
- **Sum along (0=the first non singleton dimension 1=Rows 2=Columns):** udáva či sa má sčítavať po riadkoch, stĺpcoch alebo prvý nejednotkový rozmer.

1.7.2 EXTRACT



Blok extrahuje niektoré vybrané prvky zo vstupnej matice. Veľkosť výstupu závisí od počtu extrahovaných riadkov a stĺpcov.



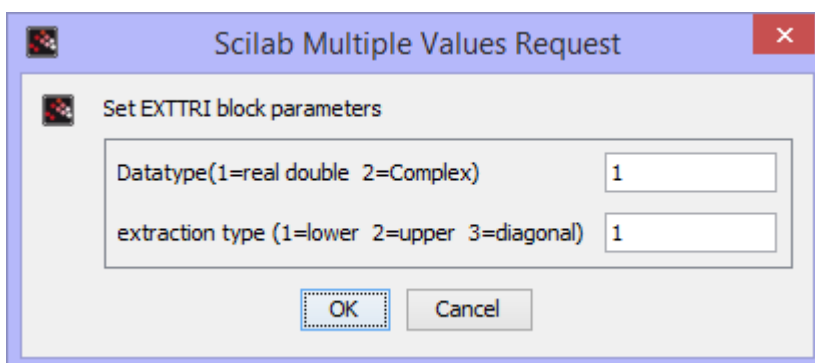
- **Datatype (1=real double 2=Complex):** udáva dátový typ výstupu. Podporované sú dátové typy double (1) a complex (2).
- **Lines to extract:** vektor udávajúci indexy riadkov, ktoré majú byť extrahované.
- **Columns to extract:** vektor udávajúci indexy stĺpcov, ktoré majú byť extrahované.

1.7.3 EXTTRI



Blok extrahuje niektoré prvky vstupnej matice U . Veľkosť výstupu je rovnaká ako táto matica. Ak je parameter „extraction type“ nastavený na:

- 1 (horný trojuholník): blok skopíruje prvky nad hlavnou diagonálou na výstup a ostatné sú vynulované,
- 2 (dolný trojuholník): blok skopíruje prvky pod hlavnou diagonálou na výstup a ostatné sú vynulované,
- 3 (uhlopriečka): blok skopíruje prvky na hlavnej diagonále na výstup a ostatné sú vynulované.

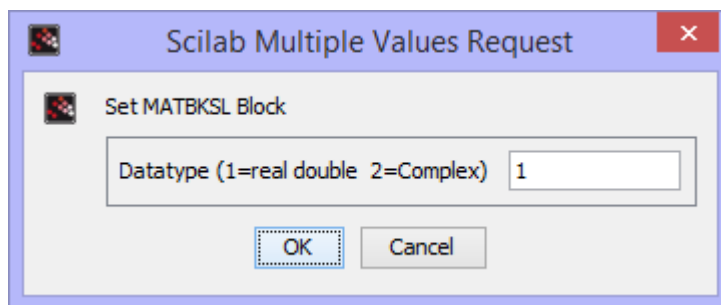


- **Datatype(1=real double 2=complex):** udáva dátový typ výstupu. Podporované sú dátové typy double (1) a complex (2).
- **Extraction type (1=lower 2=upper 3=diagonal):** udáva formu výstupnej matice. To môže byť horný trojuholník (2), dolný trojuholník (1) alebo diagonálna matica (3).

1.7.4 MATBKSL



Blok má dva vstupné porty pre matice A a B. Riadi sa vzťahom $A * x = B$, kde x je výstupná hodnota. Čím väčšia je matica A, tým je matica B menšia. Ak matica A má rozmer $M \times N1$, tak matica B musí mať rozmer $M \times N2$, kde $N1$ a $N2$ môžu byť rôzne alebo rovnaké. Výstupná premenná x bude mať rozmer $N1 \times N2$.

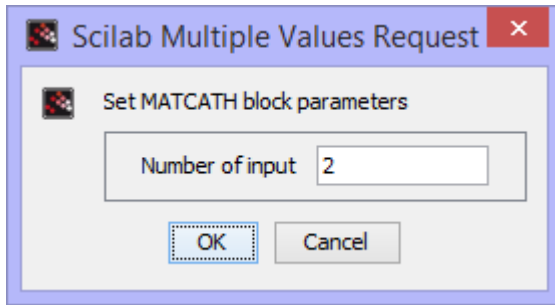


- **Datatype(1=real double 2=complex):** udáva dátový typ výstupu. Podporované sú dátové typy double (1) a complex (2).

1.7.5 MATCATH

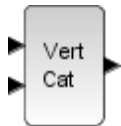


Blok prevádza horizontálne zret'azenie viacerých matíc. Vstupné matice $U1, U2, \dots, Un$ musia mať rovnaký počet riadkov

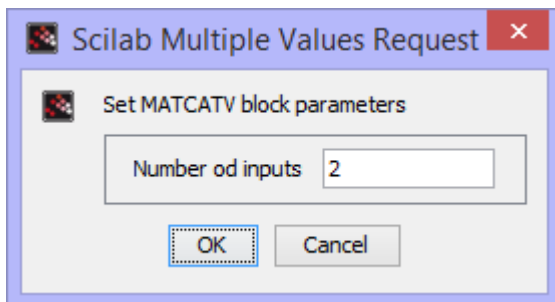


- **Number of inputs:** udáva počet vstupných matíc.

1.7.6 MATCHTV



Blok prevádza vertikálne zrežazenie viacerých matíc. Vstupné matice U_1, U_2, \dots, U_n musia mať rovnaký počet stĺpcov.

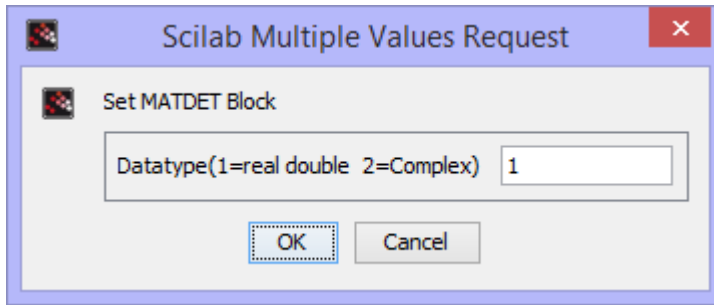


- **Number of inputs:** udáva počet vstupných matíc.

1.7.7 MATDET



Daný blok vypočíta determinant štvorcovej vstupnej matice.

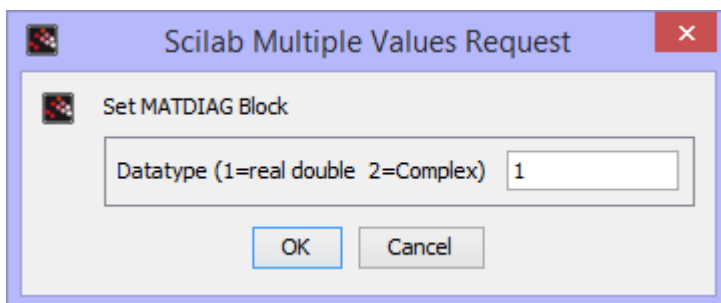


- **Datatype(1=real double 2=complex):** udáva dátový typ výstupu. Podporované sú dátové typy double (1) a complex (2).

1.7.8 MATDIAG



Vytvorí diagonálnu maticu z 1D vektoru. Ak je vstupný vektor $M \times 1$, tak výstupom bude matica $M \times M$.

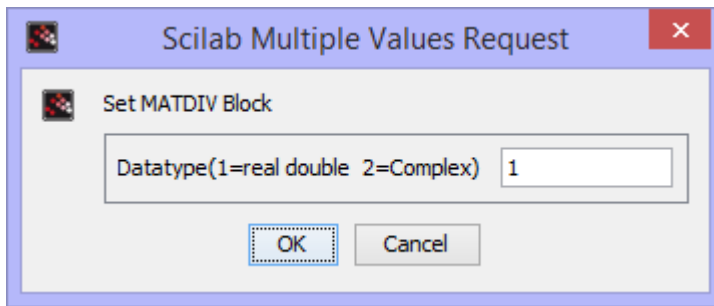


- **Datatype(1=real double 2=complex):** udáva dátový typ výstupu. Podporované sú dátové typy double (1) a complex (2).

1.7.9 MATDIV



Blok má dva vstupné porty pre matice A a B. Riadi sa vzťahom $x * B = A$, kde x je výstupná hodnota. Čím väčšia je matica A, tým je matica B menšia. Ak matica A má rozmer $M1 \times N$, tak matica B musí mať rozmer $M2 \times N$, kde $M1$ a $M2$ môžu byť rôzne alebo rovnaké. Výstupná premenná x bude mať rozmer $M1 \times M2$.



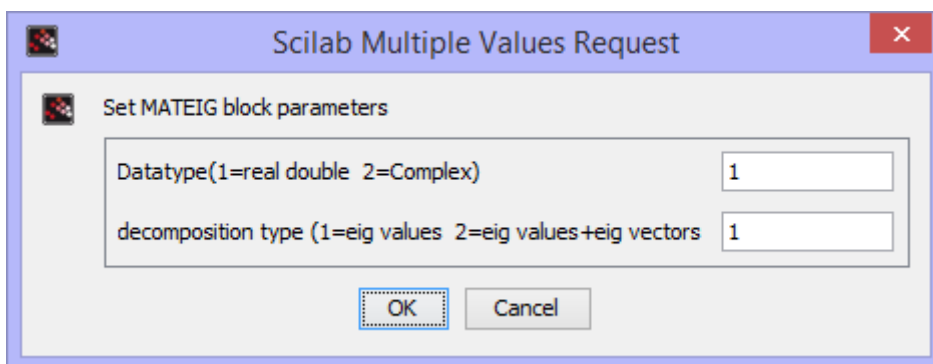
- **Datatype(1=real double 2=complex):** udáva dátový typ výstupu. Podporované sú dátové typy double (1) a complex (2).

1.7.10 MATEIG



Tento blok počíta vlastné čísla a vlastné vektory zo vstupnej štvorcovej matice U . Ak je parameter „decomposition type“ nastavený na:

- 1: výstupná hodnota bloku budú vlastné čísla vo vektorovej podobe. V prípade, že vstupná matica má rozmer $M \times M$ potom výstup bude vektor $M \times 1$.
- 2: výstupom budú dve matice. V prípade, že vstupná matica je $M \times M$, tak prvý výstup bude diagonálna matica $M \times M$ zložená z vlastných čísel, a druhá je matica $M \times M$, kde stĺpce sú vlastné vektory.

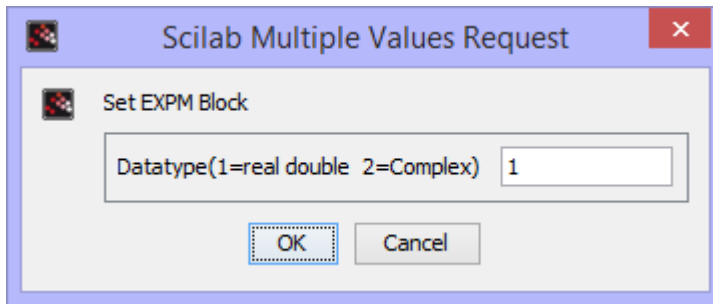


- **Datatype(1=real double 2=complex):** udáva dátový typ výstupu. Podporované sú dátové typy double (1) a complex (2).
- **Decomposition type (1=eig values 2=eig values+eig vectors):** udáva formu výstupu

1.7.11 MATEXPM



Výstupná exponenciálna matica je rátaná zo vstupnej štvorcovej matice pomocou Padeho rozvoja. Výstupom je štvorcová matica rovnakej veľkosti.

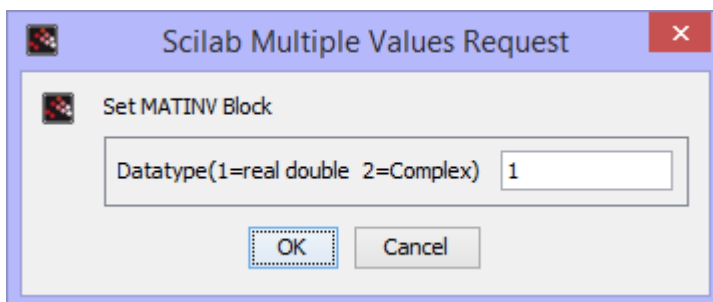


- **Datatype(1=real double 2=complex):** udáva dátový typ výstupu. Podporované sú dátové typy double (1) a complex (2).

1.7.12 MATINV



Blok vyráta inverznú štvorcovú maticu pomocou LU faktorizácie.

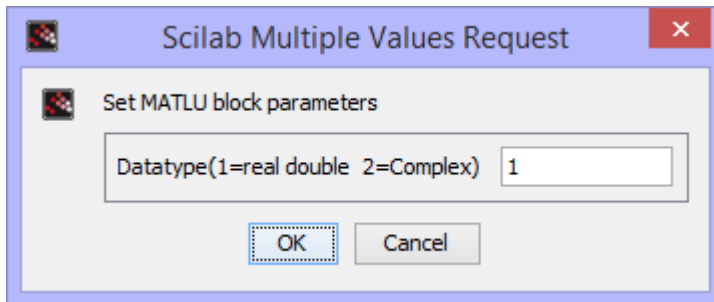


- **Datatype(1=real double 2=complex):** udáva dátový typ výstupu. Podporované sú dátové typy double (1) a complex (2).

1.7.13 MATLU



Výstupom sú dve matice L a U po LU faktorizácii vstupnej štvorcovej matice. Ak A je vstupná matica, potom $E \cdot A = L \cdot U$, kde E je permutácia matice, U je horný trojuholník matice a L je dolný trojuholník matice.



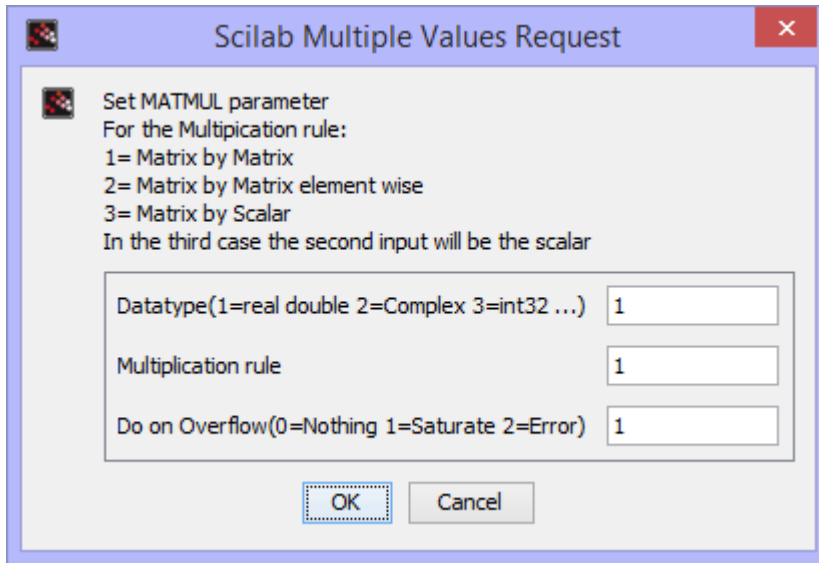
- **Datatype(1=real double 2=complex):** udáva dátový typ výstupu. Podporované sú dátové typy double (1) a complex (2).

1.7.14 MATMUL



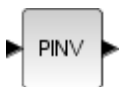
Blok násobí prvú vstupnú maticu s druhou vstupnou maticou/skalárom. Keď parameter „Multiplication rule“ je nastavený na:

- 1: blok vypočíta maticu klasickým násobením dvoch matíc. Počet riadkov prvej matice musí byť rovnaký z počtom stĺpcov druhej matice. Výstup je matica, kde počet riadkov je rovnaký ako u prvej vstupnej matice a počet stĺpcov je rovnaký ako u druhej matice.
- 2: blok vypočíta maticu pomocou Hadamardoveho súčinu. Matice musia mať rovnaké rozmery. Výstupná matica bude mať rovnaký rozmer ako vstupné matice.
- 3: blok vypočíta výstupnú maticu násobením vstupnej matice a skaláru. Výstupná matica má rovnaký rozmer ako vstupná.

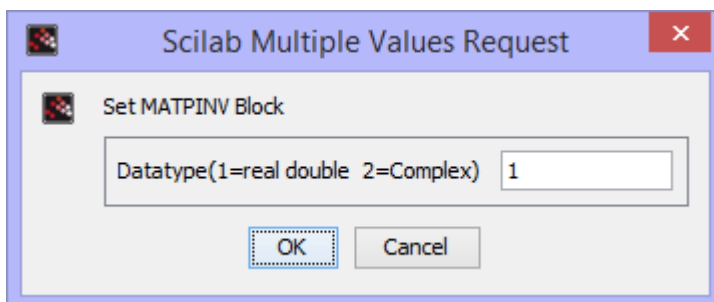


- **Datatype(1=real double 2=complex):** udáva dátový typ výstupu. Podporované sú dátové typy double (1) a complex (2).
- **Multiplication rule (1=* 2=.* 3=scalar):** výber metódy násobenia.
- **Do on Overflow (0=Nothing 1=Saturate 2=Error):** ak je nastavená 0, výsledok je podobný normálnemu súčtu dvoch celočíselných matic. Ak je nastavená 1, tak v prípade pretečenia nastaví výsledok na hodnotu nasýtenia. Ak je nastavená 2, tak v prípade pretečenia zobrazí okno s chybovým hlásením. Ak je dátový typ double alebo complex, tento parameter nie je použitý.

1.7.15 MATPINV



Výstup bloku bude inverzná matica z neštvorcovej matice pomocou teórie singulárneho rozkladu. Keď bude vstupná matica $M \times N$, potom výstupná matica bude $N \times M$.

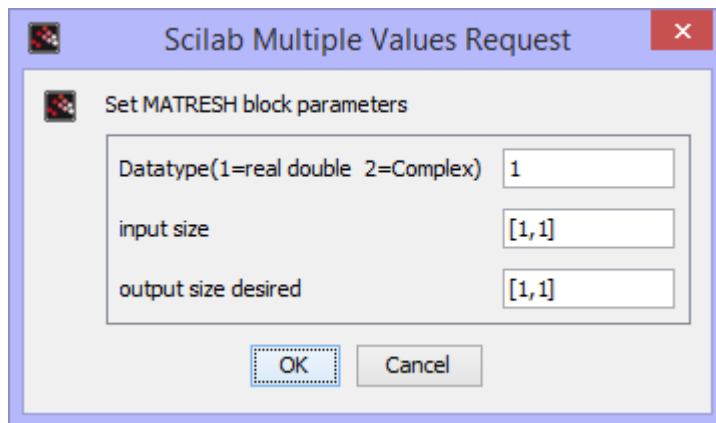


- **Datatype(1=real double 2=complex):** udáva dátový typ výstupu. Podporované sú dátové typy double (1) a complex (2).

1.7.16 MATRESH



Daný blok mení rozmery matice alebo vektoru na rozmery stanovené užívateľom. Veľkosť výstupu musí byť menšia alebo rovnaká ako veľkosť vstupu.

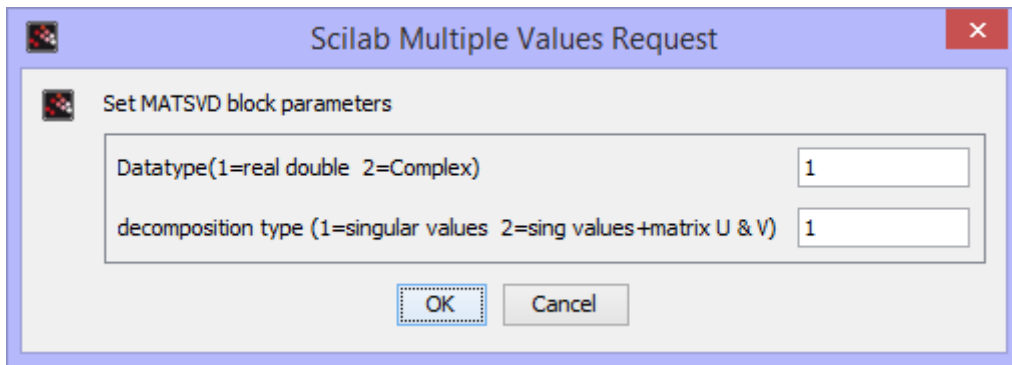


- **Datatype(1=real double 2=complex):** udáva dátový typ výstupu. Podporované sú dátové typy double (1) a complex (2).
- **Input size:** nastaví veľkosť vstupnej matice.
- **Output size desired:** nastaví požadovanú veľkosť výstupu.

1.7.17 MATSING



Blok počíta ekonomickú veľkosť teórieu singulárneho rozkladu z matice A o rozmeroch $M \times N$ aby našiel U, S a V. $A=U*S*V$



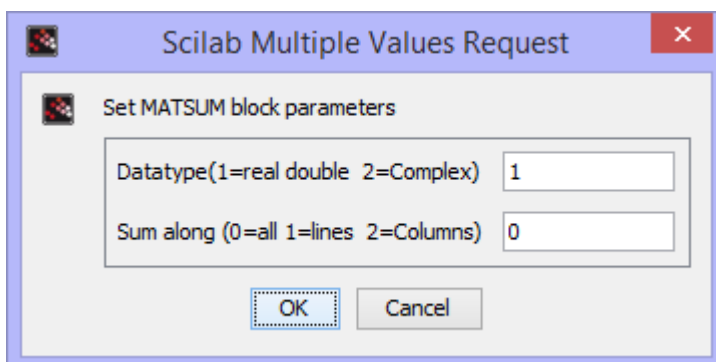
- **Datatype(1=real double 2=complex):** udáva dátový typ výstupu. Podporované sú dátové typy double (1) a complex (2).
- **Decomposition type (1=singular values 2=sing values+matrix U & V):** udáva formu výstupu. Keď je nastavená 1, výstup bude vektor singulárnych hodnôt. Keď je nastavená 2, dostaneme 3 výstupy U, S a V, kde S je diagonálna matica zložená zo singulárnych hodnôt a ostatné dva výstupy sú jednotkové matice U a V.

1.7.18 MATSUM



Blok vracia súčet prvkov vstupnej matice/vektoru. Keď je parameter „Sum along“ nastavený na:

- 0 (všetky): blok vypíše súčet všetkých prvkov matice. Výstupom je skalár,
- 1 (riadky): blok sčíta jednotlivé riadky do seba. Výstupom je riadkový vektor,
- 2 (stĺpce): blok sčíta jednotlivé stĺpce do seba. Výstupom je stĺpcový vektor.



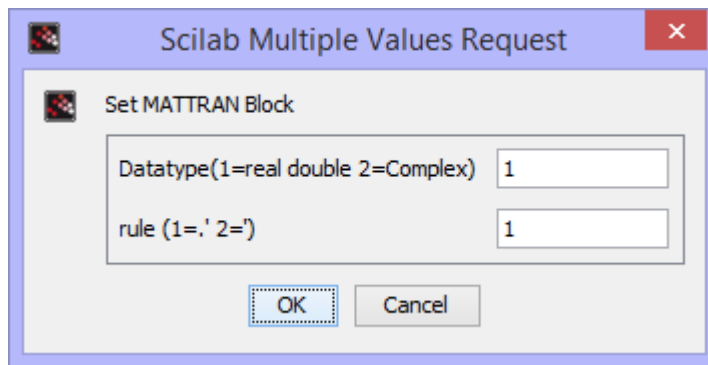
- **Datatype(1=real double 2=complex):** udáva dátový typ výstupu. Podporované sú dátové typy double (1) a complex (2).

- **Sum along (0=all 1=lines 2=Columns):** nastaví metódu, ktorá sa má využiť na získanie sumy.

1.7.19 MATTRAN



Tento blok transponuje maticu $M \times N$ na maticu $N \times M$.



- **Datatype(1=real double 2=complex):** udáva dátový typ výstupu. Podporované sú dátové typy double (1) a complex (2).
- **Rule (1=,' 2=')**: udáva pravidlo pre transponovanie. Ak je zadaná 1, na výstupe bude transponovaná matica. Ak je zadaná 2, na výstupe bude transponovaná matica z konjugovanej matice.

1.7.20 MATCONJ

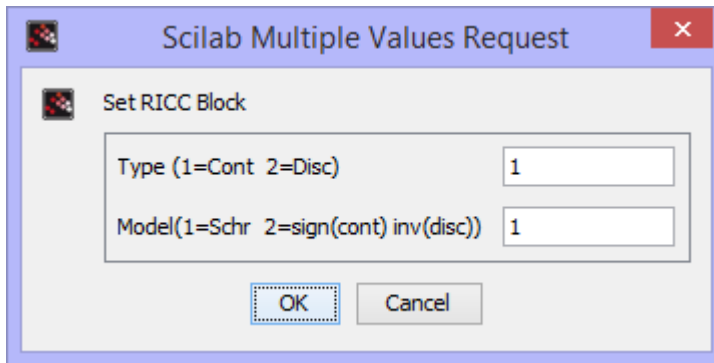


Blok počíta maticu obsahujúcu konjugovanú časť z prvkov vstupnej komplexnej matice.

1.7.21 RICC



Tento blok rieši Riccatiho rovnice použitím rôznych metód, ako pre spojité tam pre diskrétné systémy.

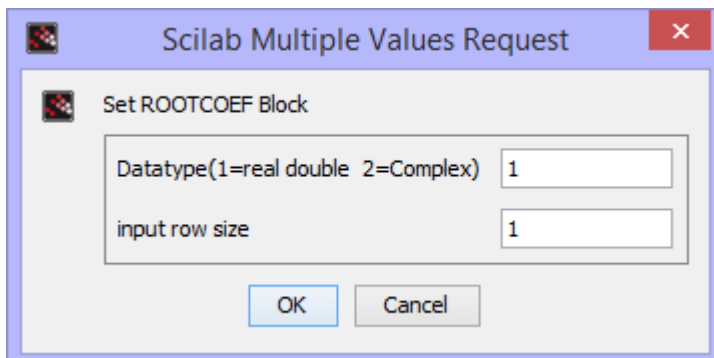


- **Type (1=Cont 2=Disc):** 1 pre spojité signál, 2 pre diskrétné signály.
- **Model(1=Schr 2=sign(cont) inv(disc)):** pre použitie Schurovej metódy (spojité a diskrétné systémy) na výpočet je nutné zvoliť 1. Voľba 2 zaistí, že blok bude využívať metódu maticovej funkcie pre spojité systémy a voľne obrátenú metódu spektrálneho rozkladu pre diskrétné systémy.

1.7.22 ROOTCOEF



Blok počíta koeficienty polynómu vzhľadom na hodnotu koreňa. Korene sú dané na vstupe vo forme stĺpcového vektora. Dĺžka vektorového koreňa musí byť špecifikovaná parametrom „input row size“.

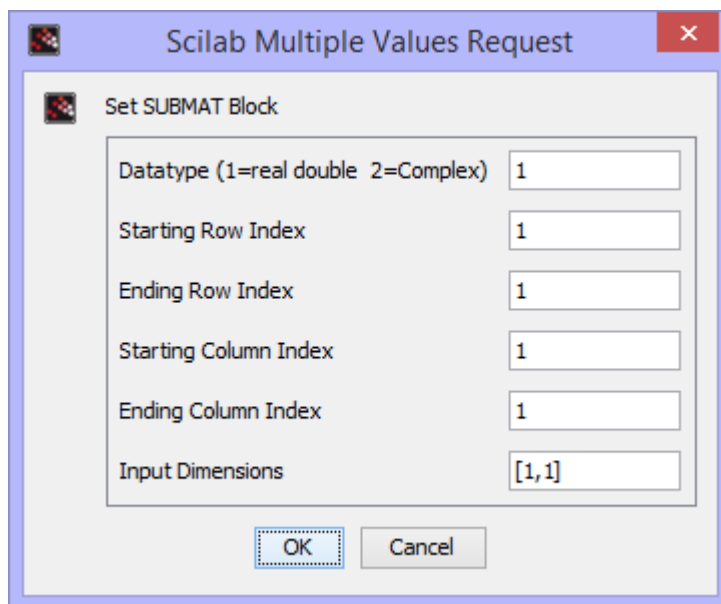


- **Datatype(1=real double 2=complex):** udáva dátový typ výstupu. Podporované sú dátové typy double (1) a complex (2).
- **Input row size:** veľkosť vstupného riadku.

1.7.23 SUBMAT



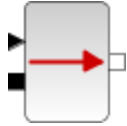
Výstupom bloku je submatica zo vstupnej matice. Podľa parametrov užívateľ presne určí ktorú časť matice chce mať na výstupe.



- **Datatype(1=real double 2=complex):** udáva dátový typ výstupu. Podporované sú dátové typy double (1) a complex (2).
- **Starting Row Index:** prvý riadok submatice.
- **Ending Row Index:** posledný riadok submatice.
- **Starting Column Index:** prvý stĺpec submatice.
- **Ending Column Index:** posledný stĺpec submatice.
- **Input Dimension:** rozmer vstupnej matice.

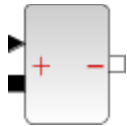
1.8 Electrical

1.8.1 CCS



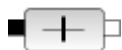
Blok pracuje ako ideálny zdroj prúdu. Aktuálna hodnota je riadená skrz explicitný vstup bloku (pripojenie ako pri štandardných Xcos blokoch). Napätie bloku je nezávislé na aktuálnej hodnote.

1.8.2 CVS



Blok pracuje ako ideálny zdroj napätia. Hodnota napätia je riadená skrz explicitný vstup bloku (pripojenie ako pri štandardných Xcos blokoch). Prúd prechádzajúci blokom je nezávislý na napätí na svorkách.

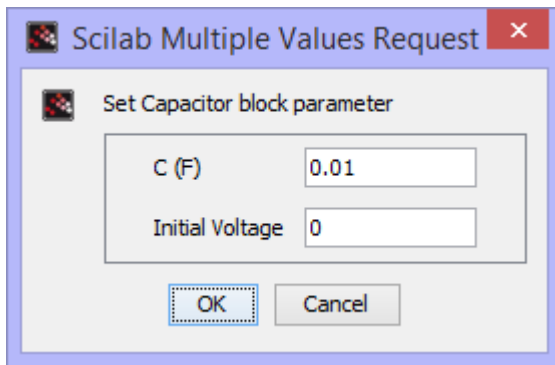
1.8.3 Capacitor



Blok simulujúci kondenzátor. To je elektrická súčiastka, ktorá môže uchovávať energiu v elektrickom obvode. Vzťah medzi napätím v kondenzátore a kapacitou je daný ako:

$$I = C \frac{dV}{dt}$$

Kondenzátory tiež môžu byť použité na rozlišovanie medzi vysokofrekvenčným a nízko-frekvenčným signálom, to ich robí užitočnými v elektrických filtroch.

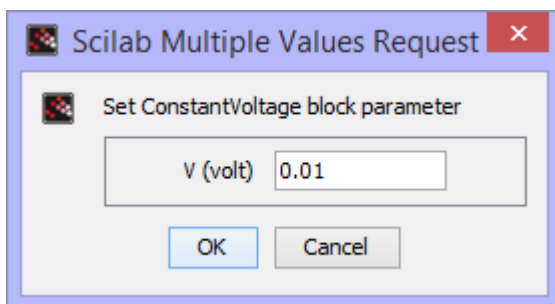


- **C (F)**: parameter udávajúci kapacitu.
- **Initial Voltage**: počiatočné napätie.

1.8.4 ConstantVoltage



Táto súčiastka sa používa pre akýkoľvek zariadenie alebo systém, ktorý podporuje konštantný elektromotorický zdroj medzi jeho portom. Výstupné napätie tohto jednosmerného zdroja napätia je definované užívateľom. Čierny port zaznamenáva kladné napätie. Ohmic-ký odpor tohto zdroja jednosmerného napätia je nulový.



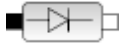
- **V (volt)**: výstupné napätie.

1.8.5 CurrentSenzor



Tento blok sa zapája do série v elektrickom obvode na meranie aktuálneho prúdu prechádzajúceho danou súčiastkou. Zvyčajne prúd, ktorý tečie do čierneho portu sa považuje za pozitívny. Ohmic-ký odpor tohto bloku je nulový.

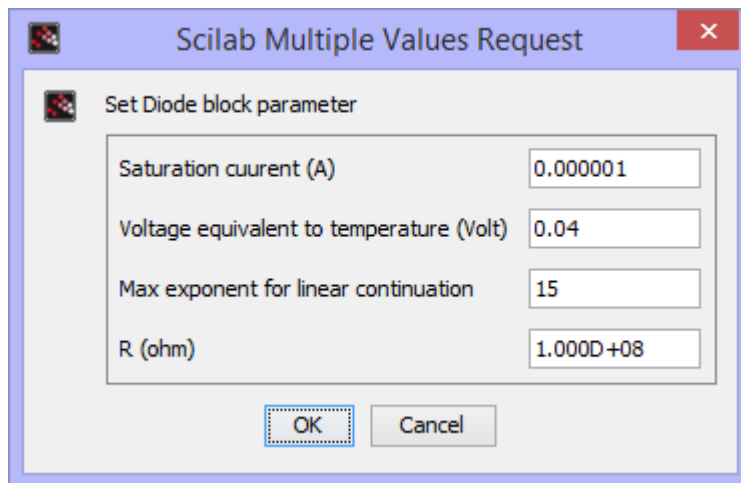
1.8.6 Diode



Táto súčiastka je zložená z jednoduchej diódy a ohmického odporu (R). Prúd prechádzajúci týmto blokom je definovaný ako funkcia napätia:

$$i = I_{ds} \left(\exp^{\frac{u}{V_t}} - 1 \right) + \frac{v}{R}$$

kde I_{ds} a V_t sú saturovaný prúd a napätie zodpovedajúce teplote. Ak exponent dosiahne určitý zadaný limit (Max exponent for linear continuation), VA charakteristika diódy sa stáva lineárnou aby sa predišlo pretečeniu.



- **Saturation current (A):** prúd nasýtenia.
- **Voltage equivalent to temperature (Volt):** napäťový ekvivalent teploty.
- **Max exponent for linear continuation:** maximálny exponent pre lineárny spojitý priebeh.
- **R (ohm):** paralelná ohmická odolnosť.

1.8.7 Ground

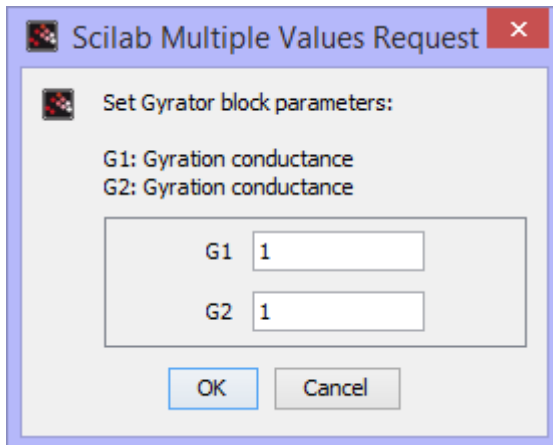


Súčasťou prvku je jeden port poskytujúci referenčné napätie v elektrických obvodoch. Potenciál na uzemňovací uzol je nulový. Každý elektrický obvod musí obsahovať aspoň jeden uzemňovací prvok.

1.8.8 Gyrator



Blok s dvomi párami portov definovaných nasledujúcimi vzťahmi: $i_1 = G_2 * V_2$ a $i_2 = -G_1 * v_1$, kde konštanty G_1 a G_2 sú vodivosti prúdenia.

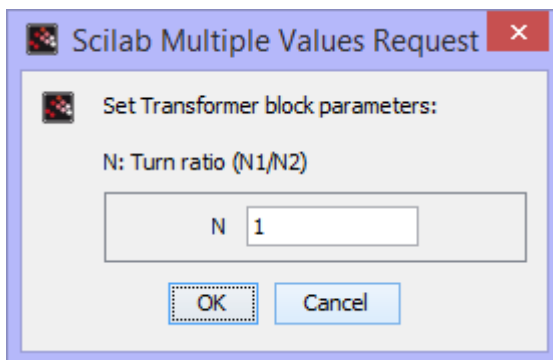


- **G1:** vodivosť prúdenia
- **G2:** vodivosť prúdenia

1.8.9 IdealTransformer



Ideálny transformátor je ideálne odporový obvod cez dva páry portov, ktoré sú charakterizované vzťahmi: $v_1 = n * v_2$ a $i_2 = -n * i_1$, kde n je reálne číslo označované ako počet otáčok.



- **N:** pomer otáčok (N_1/N_2).

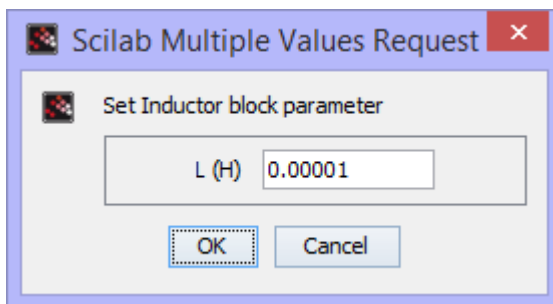
1.8.10 Inductor



Cievka je elektrická súčiastka, ktorá môže uchovávať energiu v elektrickom obvode. Vzťah medzi napätím na cievke, indukčnosťou a prechádzajúcim prúdom je daný ako:

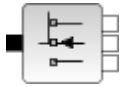
$$U = L \frac{dI}{dt}$$

Cievky môžu byť tiež použité na rozlišovanie medzi vysokofrekvenčným a nízkofrekvenčným signálom. Zároveň vykazuje vysokú impedanciu pre vysokofrekvenčné signály.



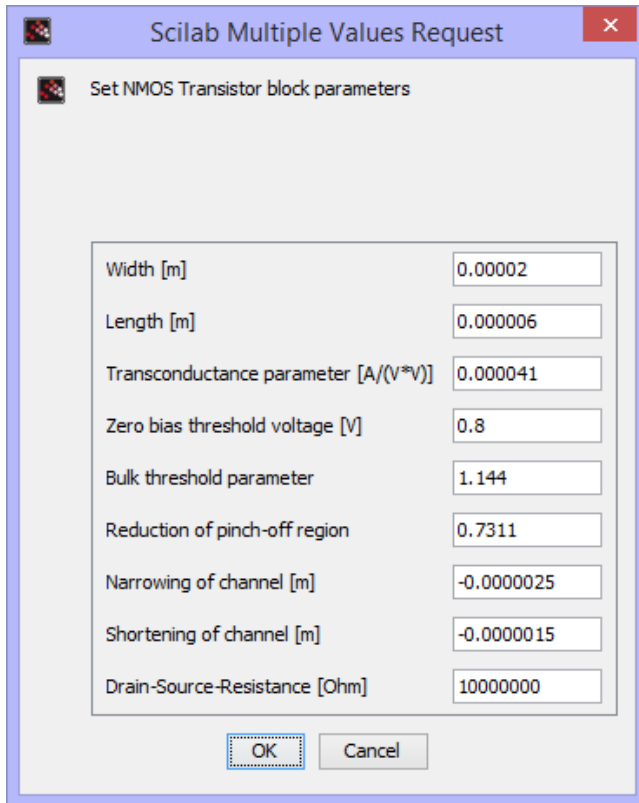
- **L (H)**: hodnota indukčnosti.

1.8.11 NMOS



Daný blok je jednoduchým modelom n-kanálového MOSFET tranzistoru. Model neuvažuje o kapacitách.

W [m]	L [m]	Beta [1/V ²]	Vt [V]	K2	K5	DW [m]	DL[m]	
12.e-6	4.e-6	0,062	-4,5	0,24	0,61	-1.2e-6	-.9e-6	vyprázdnenie
60.e-6	3.e-6	0,048	0,1	0,08	0,68	-1.2e-6	-.9e-6	zvýšenie
12.e-6	4.e-6	0,0625	-0,8	0,21	0,78	-1.2e-6	-.9e-6	nula
50.e-6	8.e-6	0,0299	0,24	1,144	0,7311	-5.4e-6	-4.e-6	
20.e-6	6.e-6	0,041	0,8	1,144	0,7311	-2.5e-6	-1.5e-6	
30.e-6	9.e-6	0,025	-4	0,861	0,878	-3.4e-6	-1.74e-6	
30.e-6	5.e-6	0,031	0,6	1,5	0,72	0	-3.9e-6	
50.e-6	6.e-6	0,0414	-3,8	0,34	0,8	-1.6e-6	-2.e-6	vyprázdnenie
50.e-6	5.e-6	0,03	0,37	0,23	0,86	-1.6e-6	-2.e-6	zvýšenie
50.e-6	6.e-6	0,038	-0,9	0,23	0,707	-1.6e-6	-2.e-6	nula
20.e-6	4.e-6	0,06776	0,5409	0,065	0,71	-.8e-6	-.2e-6	
20.e-6	4.e-6	0,06505	0,6209	0,065	0,71	-.8e-6	-.2e-6	
20.e-6	4.e-6	0,05365	0,6909	0,03	0,8	-.3e-6	-.2e-6	
20.e-6	4.e-6	0,05365	0,4909	0,03	0,8	-.3e-6	-.2e-6	
12.e-6	4.e-6	0,023	-4,5	0,29	0,6	0	0	vyprázdnenie
60.e-6	3.e-6	0,022	0,1	0,11	0,65	0	0	zvýšenie
12.e-6	4.e-6	0,038	-0,8	0,33	0,6	0	0	nula
20.e-6	6.e-6	0,022	0,8	1	0,66	0	0	

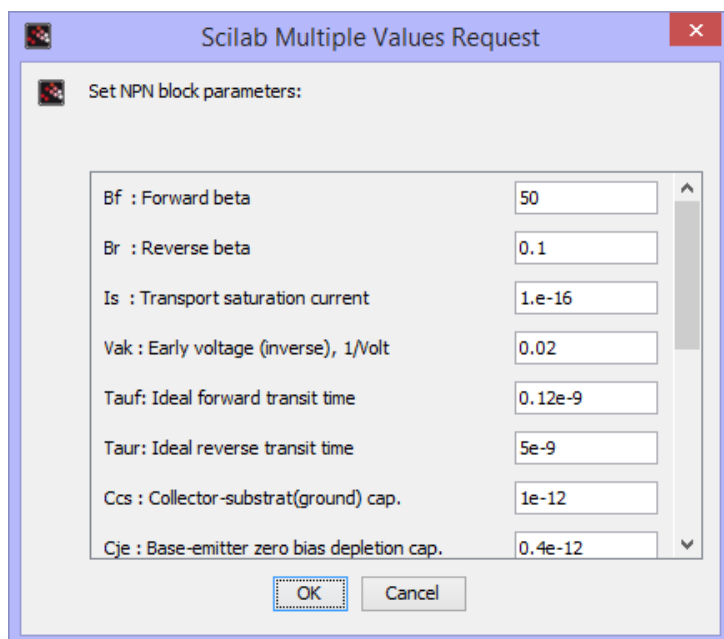


- **Width [m]:** šírka W .
- **Length [m]:** dĺžka L .
- **Transconductance parameter [A/(V*V)]:** parameter strmosti $Beta$.
- **Zero bias threshold voltage [V]:** nulové prahové napätie V_t .
- **Bulk treshold parameter:** parameter prahového rozsahu K_2 .
- **Reduction of pinch-off region:** redukcia oblasti „pinch-off“ K_5 .
- **Narrowing of channel [m]:** zúženie kanála dW .
- **Shortening of channel [m]:** skrátenie kanála dL .
- **Drain-Source-Resistance [Ohm]:** R_{DS} .

1.8.12 NPN

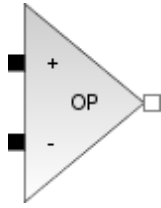


Toto je jednoduchý model bipolárneho NPN plošného tranzistoru.



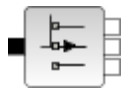
Parameter	Základná hodnota	Popis
Bf	50	Predná beta
Br	0,1	Reverzná beta
Is	1.e-16	Dopravné omeškanie prúdu [A]
Vak	0,02	Skoré napätie (inverzné), 1/Volt [1/V]
Tauf	0.12e-9	Ideálny popredný transportný čas [s]
Taur	5,00E-09	Ideálny reverzný transportný čas [s]
Ccs	1,00E-12	Kolektor-substrát(zem) [F]
Cje	0.4e-12	Báza-emitor nulové prahové vyčerpanie kapacity [F]
Cjc	0.5e-12	Báza-kolektor nulové prahové vyčerpanie kapacity[F]
Phie	0,8	Báza-emitor difúzne napätie [V]
Me	0,4	Báza-emitor stupňujúci exponent
Phic	0,8	Báza-kolektor difúzne napätie [V]
Mc	0,333	Báza-kolektor stupňujúci exponent
Gbc	1,00E-15	Báza-kolektor vodivosť [S]
Gbe	1,00E-15	Báza-emitor vodivosť [S]
Vt	0,02585	Napätie zodpovedajúce teplote [V]
EMin	-100	ak $x < E_{Min}$, funkcia $\exp(x)$ je linearizovaná
EMax	40	ak $x > E_{Max}$, funkcia $\exp(x)$ je linearizovaná

1.8.13 OpAmp



Ideálny operačný zosilňovač s dvomi portami. Ľavé porty sú pevne nastavené na $v_1=0$ a $i_1=0$. Na pravej strane sa mení napätie v_2 a prúd i_2 .

1.8.14 PMOS



PMOS predstavuje jednoduchý model p-kanálového MOSFET tranzistoru. Model neuvažuje o kapacitách.

Parameter	Value
Width [m]	0.00005
Length [m]	0.000006
Transconductance parameter [A/(V*V)]	0.0000105
Zero bias threshold voltage [V]	-1
Bulk threshold parameter	0.41
Reduction of pinch-off region	0.839
Narrowing of channel [m]	-0.0000025

- **Width [m]:** šírka W .
- **Length [m]:** dĺžka L .
- **Transconductance parameter [A/(V*V)]:** parameter strmosti $Beta$.
- **Zero bias threshold voltage [V]:** nulové prahové napätie V_t .
- **Bulk threshold parameter:** parameter prahového rozsahu K_2 .
- **Reduction of pinch-off region:** redukcia oblasti „pinch-off“ K_5 .
- **Narrowing of channel [m]:** zúženie kanála dW .

- **Shortening of channel [m]:** skrátenie kanála dL .
- **Drain-Source-Resistance [Ohm]:** RDS .

W [m]	L[m]	Beta [$1/V^2$]	Vt [V]	K2	K5	DW [m]	DL [m]
50.e-6	8.e-6	0,0085	-0,15	0,41	0,839	-3.8e-6	-4.0e-6
20.e-6	6.e-6	0,0105	-1	0,41	0,839	-2.5e-6	-2.1e-6
30.e-6	5.e-6	0,0059	-0,3	0,98	1,01	0	-3.9e-6
30.e-6	5.e-6	0,0152	-0,69	0,104	1,1	-.8e-6	-.4e-6
30.e-6	5.e-6	0,0163	-0,69	0,104	1,1	-.8e-6	-.4e-6
30.e-6	5.e-6	0,0182	-0,69	0,086	1,06	-.1e-6	-.6e-6
20.e-6	6.e-6	0,0074	-1	0,4	0,59	0	0

1.8.15 PNP



Tento blok predstavuje jednoduchý model bipolárneho PNP plošného tranzistoru.

Scilab Multiple Values Request

Set PNP block parameters:

Bf : Forward beta	50
Br : Reverse beta	0.1
Is : Transport saturation current	1.e-16
Vak : Early voltage (inverse), 1/Volt	0.02
Tauf: Ideal forward transit time	0.12e-9
Taur: Ideal reverse transit time	5e-9
Ccs : Collector-substrat(ground) cap.	1e-12

OK Cancel

Parameter	Základná hodnota	Popis
Bf	50	Predná beta
Br	0,1	Reverzná beta
Is	1.e-16	Dopravné omeškanie prúdu [A]
Vak	0,02	Skoré napätie (inverzné), 1/Volt [1/V]
Tauf	0.12e-9	Ideálny popredný transportný čas [s]
Taur	5,00E-09	Ideálny reverzný transportný čas [s]
Ccs	1,00E-12	Kolektor-substrát(zem) [F]
Cje	0.4e-12	Báza-emitor nulové prahové vyčerpanie kapacity [F]
Cjc	0.5e-12	Báza-kolektor nulové prahové vyčerpanie kapacity[F]
Phie	0,8	Báza-emitor difúzne napätie [V]
Me	0,4	Báza-emitor stupňujúci exponent
Phic	0,8	Báza-kolektor difúzne napätie [V]
Mc	0,333	Báza-kolektor stupňujúci exponent
Gbc	1,00E-15	Báza-kolektor vodivosť [S]
Gbe	1,00E-15	Báza-emitor vodivosť [S]
Vt	0,02585	Napätie zodpovedajúce teplote [V]
EMin	-100	ak $x < E_{Min}$, funkcia $\exp(x)$ je linearizovaná
EMax	40	ak $x > E_{Max}$, funkcia $\exp(x)$ je linearizovaná

1.8.16 PotentialSensor



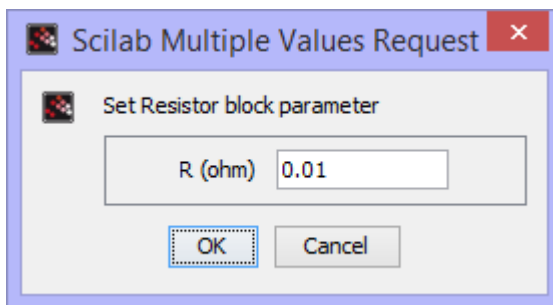
Blok sa využíva na meranie napätia vzhľadom k referenčnému napätiu (uzemňujúci blok) v elektrickom obvode. Napätie je dané na výstupnom porte.

1.8.17 Resistor



Odpor je elektrická súčiastka, ktorá odoláva elektrickému prúdu (I) tým, že produkuje úbytok napätia (U) medzi jej svorkami podľa Ohmového zákona.

$$R = \frac{U}{I}$$

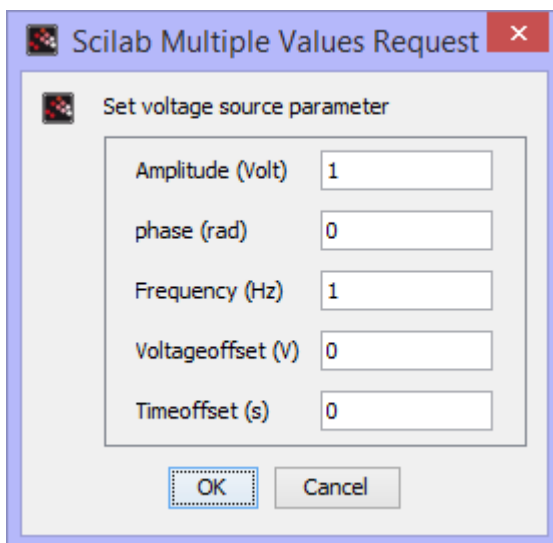


- **R (ohm):** odpor rezistoru.

1.8.18 SineVoltage



Súčiastka generuje napäťový signál v tvare funkcie sínus. Vnútorňý odpor je nulový.



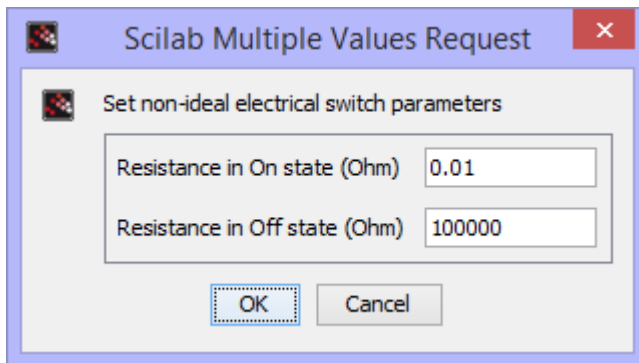
- **Amplitude (Volt):** sínusová amplitúda napätia.
- **Phase (rad):** fázové posunutie sínusového napätia.

- **Frequency (Hz):** sínusová frekvencia napätia.
- **Voltageoffset (V):** ofset napätia sínusového napätia.
- **Timeoffset (s):** čas pred začiatkom. Počas tejto doby je hodnota signálu daná parametrom „Voltageoffset“.

1.8.19 Switch



Blok nie je ideálnym dvojpólovým spínačom. V uzavretom stave má na výstupe inú hodnotu ako v otvorenom stave. Tieto hodnoty sú nastaviteľné užívateľom.

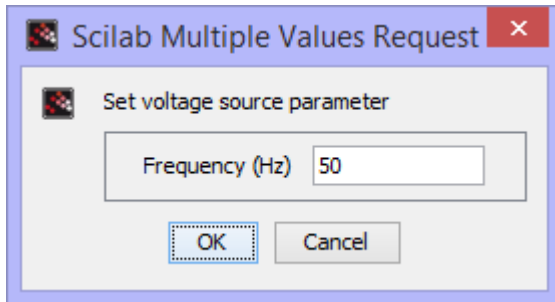


- **Resistance in On state (Ohm):** odpor na výstupe, keď je spínač uzavretý.
- **Resistance in Off state (Ohm):** odpor na výstupe, keď je spínač otvorený.

1.8.20 VVsourceAC

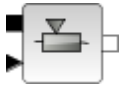


Model pracuje ako zdroj striedavého napätia. Blok poskytuje sínusové napätie na jeho portoch. Amplitúda výstupného napätia je riadená explicitným vstupným portom a frekvencia je definovaná užívateľom. Ohmický odpor bloku je nulový.



- **Frequency (Hz):** frekvencia výstupného sínusového napätia.

1.8.21 VariableResistor



Tento blok predstavuje premenný ohmický odpor. Odpor (R_x) je riadený pomocou explicitného vstupného portu.

$$R_x = \frac{U}{I}$$

1.8.22 VoltageSensor



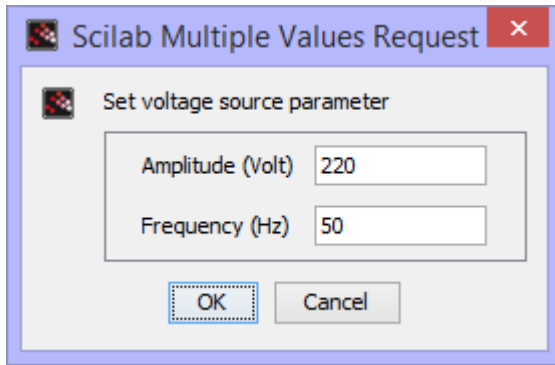
Daný blok sa využíva na meranie rozdielu napätia medzi dvomi uzlami v elektrickom obvode. Výstupný signál je rozdiel medzi napätím v čiernom a bielom porte.

$$U_{výstup} = U_{čierny} - U_{biely}$$

1.8.23 VsourceAC



Pracuje ako zdroj striedavého napätia so sínusovým výstupom napätia. Amplitúda a frekvencia výstupného napätia je nastavovaná užívateľom. Ohmický odpor bloku je nulový.



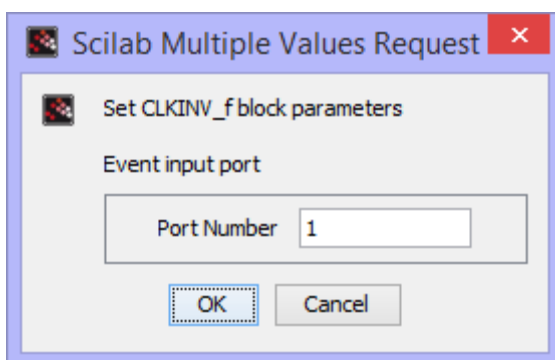
- **Amplitude (Volt):** amplitúda výstupného sínusového signálu napätia.
- **Frequency (Hz):** frekvencia výstupného sínusového signálu napätia.

1.9 Port & Subsystems

1.9.1 CLKINV_f



Blok predstavujúci aktivačný vstupný port, ktorý musí byť použitý len vo vnútri „Xcos Super Blok“ a pripojený na aktivačný vstup v subsysteme. Tento je odkaz na udalosť z vonka systému do subsystemu. V super bloku musia byť všetky aktivačné porty očíslované od 1. Na očíslovanie slúži parameter „Port Number“.

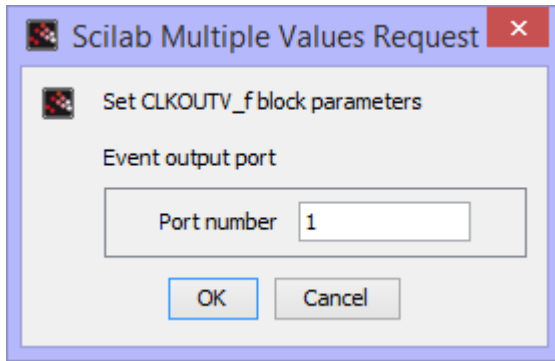


- **Port number:** celé číslo definujúce číslo portu.

1.9.2 CLKOUTV_f



Blok reprezentujúci aktivačný výstupný port. Musí byť použitý výhradne iba vo vnútri Super bloku. Zároveň funguje ako odkaz na udalosť z hlavného diagramu do subsystemu. V Super bloku musia byť aktivačné výstupné porty očíslované od 1 do celkového počtu aktivačných výstupov.

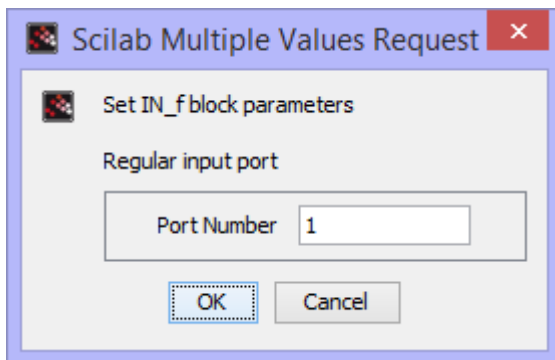


- **Port number:** celé číslo definujúce číslo portu.

1.9.3 IN_f



Blok predstavuje regulárny vstupný port. Musí byť použitý iba vo vnútri „Xcos Super Block“ a pripojený na regulárny vstup v subsysteme. Tento blok je zároveň odkaz na signál z vonku do subsystemu. Všetky porty v Super bloku musia byť očíslované od 1 do celkového počtu portov.

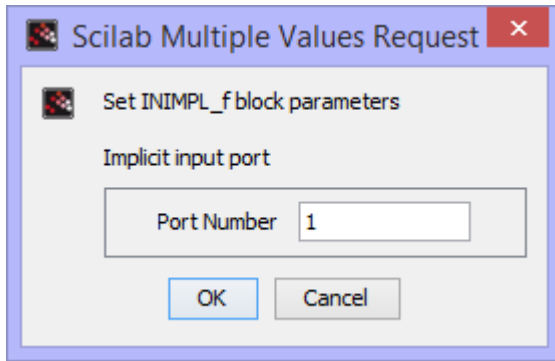


- **Port number:** celé číslo definujúce číslo portu.

1.9.4 INIMPL_f



Daný blok predstavuje implicitný vstupný port. Ten musí byť použitý iba vo vnútri „Xcos Super Block“ a pripojený na implicitný vstup v subsysteme. Tento blok je zároveň odkaz na signál z vonku do subsystemu. Všetky porty v Super bloku musia byť očíslované od 1 do celkového počtu portov.

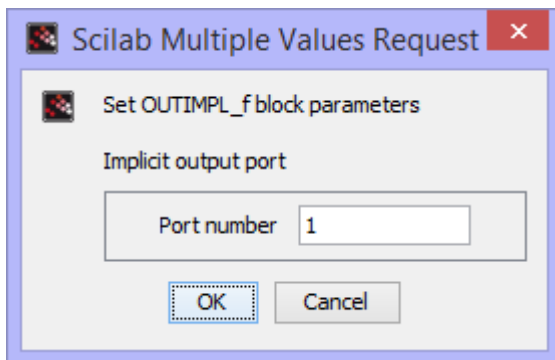


- **Port number:** celé číslo definujúce číslo portu.

1.9.5 OUTIMPL_f



Tento blok reprezentuje regulárny implicitný port, ktorý musí byť použitý iba vo vnútri „Xcos Super Block“ a pripojený na implicitný výstup v subsysteme. Tento blok je zároveň odkaz na signál z vonku do subsystemu. Všetky porty v Super bloku musia byť očíslované od 1 do celkového počtu portov.

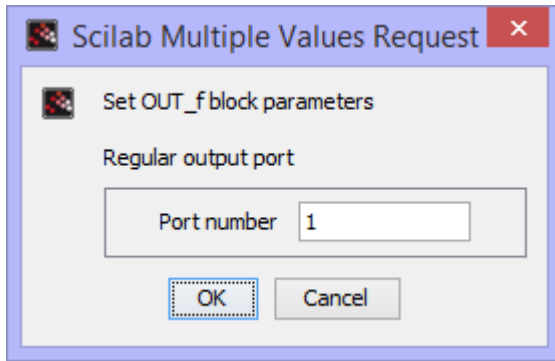


- **Port number:** celé číslo definujúce číslo portu.

1.9.6 OUT_f



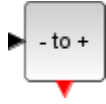
Tento blok reprezentuje regulárny výstupný port, ktorý musí byť použitý iba vo vnútri „Xcos Super Block“ a pripojený na regulárny výstup v subsysteme. Tento blok je zároveň odkaz na signál z vonku do subsystemu. Všetky porty v Super bloku musia byť očíslované od 1 do celkového počtu portov.



- **Port number:** celé číslo definujúce číslo portu.

1.10 Zero crossing detection

1.10.1 NEGTOPOS_f



Výstupná udalosť bloku je generovaná keď hodnota na vstupnom porte zmení zápornú hodnotu na kladnú.

1.10.2 POSTONEG_f

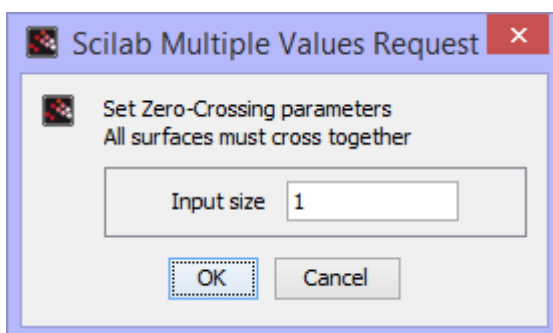


Výstupná udalosť bloku je generovaná keď hodnota na vstupnom porte zmení kladnú hodnotu na zápornú.

1.10.3 ZCROSS_f



Výstupná udalosť bloku je generovaná keď všetky vstupné signály (viac ako jeden) prechádzajú cez nulu.



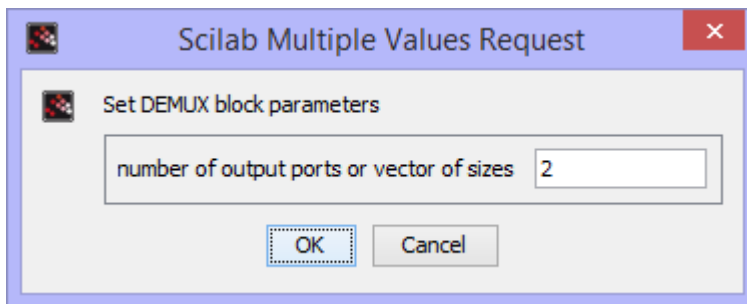
- **Input size:** kladné celé číslo označujúce veľkosť vstupu.

1.11 Signal Routing

1.11.1 DEMUX, DEMUX_f



Blok posiela (kopíruje) hodnoty vstupného vektora na všetky dostupné výstupné porty.

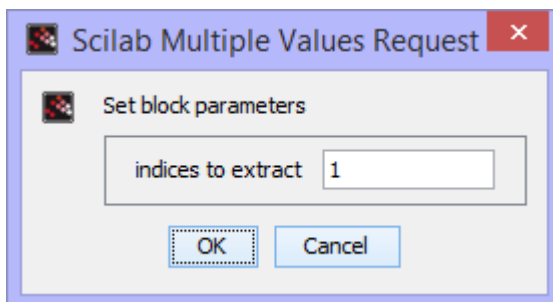


- **Number of input ports or vector of sizes:** kladné celé číslo menšie alebo rovné 8. Udáva počet výstupných portov bloku.

1.11.2 EXTRACTOR



Extrahuje vybraný vstupný port.

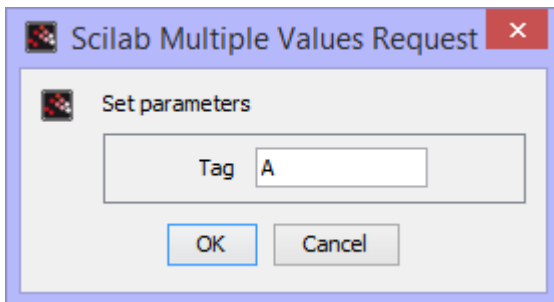


- **Indices to extract:** indexy vstupných portov, ktoré majú byť extrahované.

1.11.3 FROM

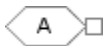


Hlavnou úlohou blokov „GOTO/FROM“ je dopraviť signál z jedného bloku do iného bez fyzického pripojenia. Blok „FROM“ transportuje prijaté dáta na svoj výstup. Blok „FROM“ môže prijímať dáta len z jedného bloku „GOTO“, zatiaľ čo blok „GOTO“ môže posilať dáta viacerým blokom „FROM“. Dané bloky sú prepojené parametrom „Tag“ a podporujú všetky dátové typy.

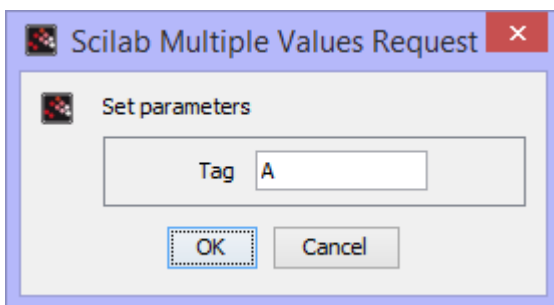


- **Tag:** značka označujúca blok „FROM“, z ktorého sa má priviesť signál na tento blok.

1.11.4 FROMMO



Tento blok sa používa na prepojenie s blokmi „Modelica“. Pracuje na princípe ako blok „FROM“.



- **Tag:** značka označujúca blok „FROMMO“, z ktorého sa má priviesť signál na tento blok.

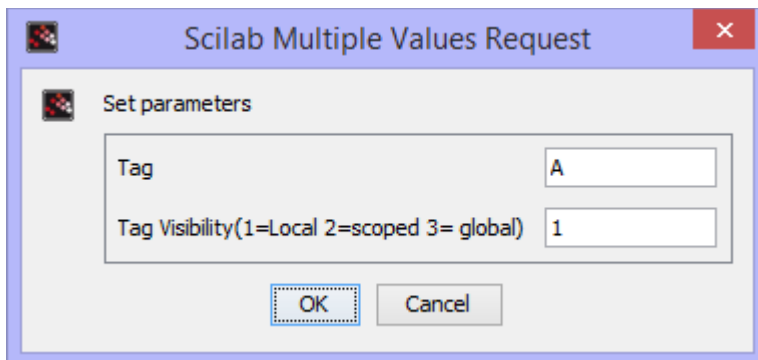
1.11.5 GOTO



Hlavnou úlohou blokov „GOTO/FROM“ je dopraviť signál z jedného bloku do iného bez fyzického pripojenia. Blok „FROM“ transportuje prijaté dáta na svoj výstup. Blok „FROM“ môže prijímať dáta len z jedného bloku „GOTO“, zatiaľ čo blok „GOTO“ môže posilať dáta viacerým blokom „FROM“. Dané bloky sú prepojené parametrom „Tag“ a podporujú všetky dátové typy.

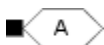
Viditeľnosť tohto bloku je daná parametrom „Tag Visibility“, ktorý môže nadobudnúť tri rôzne stavy:

- Local: bloky „GOTO“ a „FROM“ musia byť v rovnakom subsysteme,
- Scoped: bloky „GOTO“ a „FROM“ musia byť v rovnakom subsysteme alebo v ktoromkoľvek subsysteme pod blokom „GotoTagVisibility“ v modelovej hierarchii,
- Global: bloky „GOTO“ a „FROM“ môžu byť kdekoľvek v modeli.

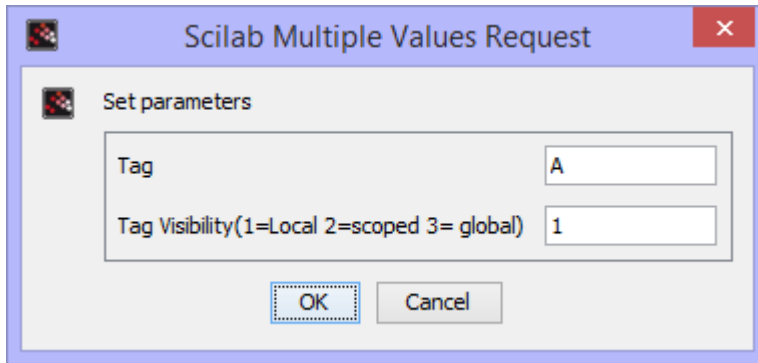


- **Tag:** značka označujúca blok „GOTO“.
- **Tag Visibility(1=Local 2=scoped 3=global):** parameter udávajúci viditeľnosť bloku.

1.11.6 GOTOMO



Tento blok sa používa na prepojenie s blokmi „Modelica“. Pracuje na princípe ako blok „GOTO“.

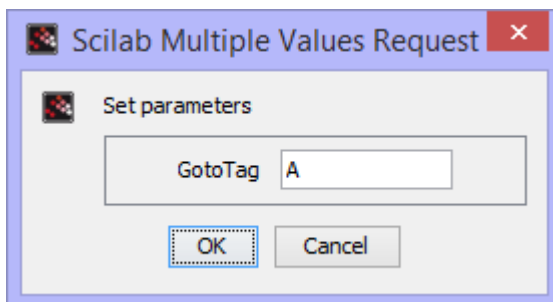


- **Tag:** značka označujúca blok „GOTO“.
- **Tag Visibility(1=Local 2=scoped 3= global):** parameter udávajúci viditeľnosť bloku.

1.11.7 GotoTagVisibility



Tento blok definuje prístupnosť bloku „GOTO“, keď je nastavený na „scoped“. Blok „FROM“ zodpovedajúci určitému bloku „GOTO“ musí byť v tom istom subsystéme ako tento blok alebo v subsystémoch pod ním v modelovej hierarchii.

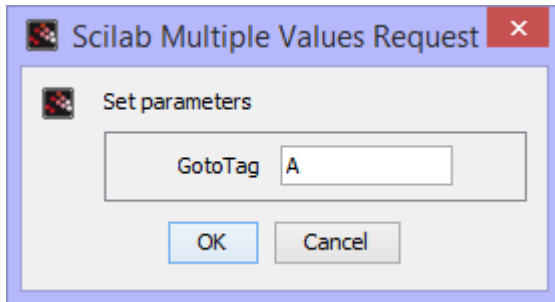


- **GotoTag:** značka, ktorá definuje viditeľnosť umiestnením tohto bloku.

1.11.8 GotoTagVisibilityMO



Daný blok sa používa v prípade, že existuje prepojenie v blokmi „Modelica“. Inak pracuje ako blok „GotoTagVisibility“.

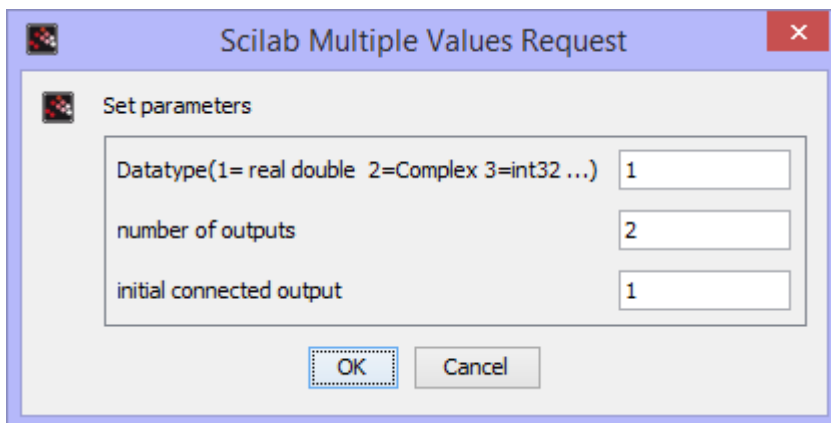


- **GotoTag:** značka, ktorá definuje viditeľnosť umiestnením tohto bloku.

1.11.9 ISELECT_m



Blok vyberie signál z prichádzajúcich udalostí.

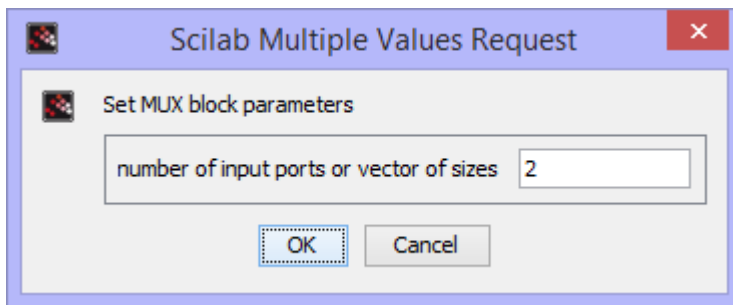


- **Datatype(1=real double 2=Complex 3=int32 ...):** nastaví dátový typ vstupu/výstupu. Podporuje všetky dátové typy.
- **Number of outputs:** skalár udávajúci počet regulárnych a aktivačných výstupných portov.
- **Initial connected output:** celé číslo udávajúce počet vstupov.

1.11.10 MUX, MUX_f



Tento blok spája všetky vstupné vektory do jedného výstupného.

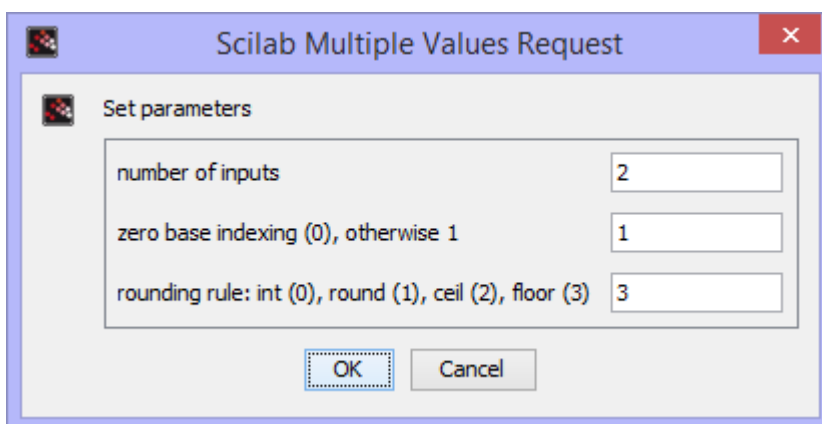


- **Number of input ports or vector of sizes:** celé číslo väčšie alebo rovné 1 a menšie ako 8. Udáva počet vstupných portov.

1.11.11 M_SWITCH



Spínací blok s viacerými vstupmi, kde prvý (horný) vstup je nazývaný ako riadiaci vstup, zatiaľ čo ostatné sú dátové vstupy. Hodnota riadiaceho vstupu určuje, ktorá vstupná hodnota sa dostane na výstup.



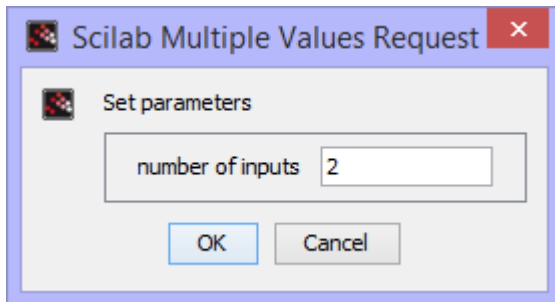
- **Number of inputs:** parameter udávajúci počet vstupných portov.
- **Zero base indexing (0), otherwise (1):** ak je vybraná 0, blok používa indexovanie od 0. Ak je vybraná 1, blok používa indexovanie od 1.
- **Rounding rule: int(0), round(1), ceil(2), floor(3):** voľba spôsobu zaokrúhľovania.

1.11.12 NRMSOM_f



Bus creator

Tento blok spája vstupy do jedného výstupu

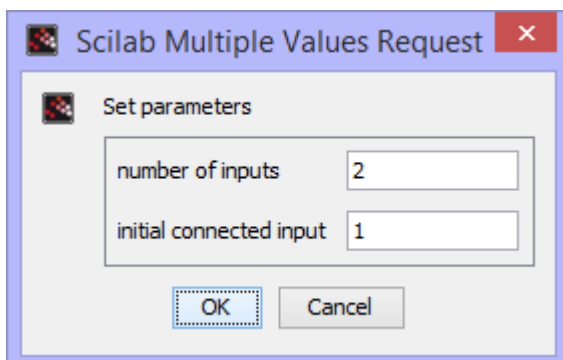


- **Number of inputs:** udáva počet vstupných portov, ktoré majú byť zlúčené.

1.11.13 RELAY_f



Tento blok prioritne smeruje na výstup jeden z regulárnych vstupov. Po príchode ďalších ríadiacich udalostí sa posielajú na výstup aj ostatné vstupy.

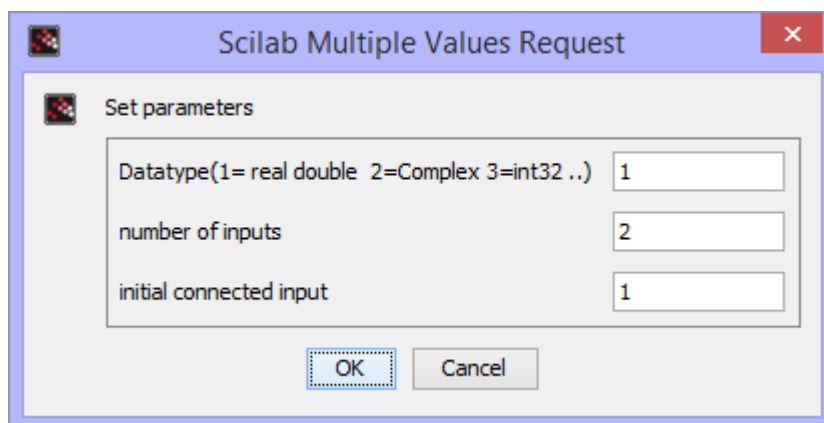


- **Number of inputs:** skalár, ktorý udáva počet regulárnych a riadiacich výstupov.
- **Initial connected input:** celé číslo, ktoré udáva číslo vstupu, ktorý má byť pripojený na výstup ako prvý. Musí to byť číslo od 1 do celkového počtu vstupov.

1.11.14 SELECT_m

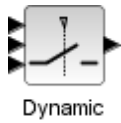


Blok akceptuje buď vektorové alebo maticové signály na vstupe. Tieto vstupy sú blokom vyhodnocované a posielané na výstup. Podľa typu vstupu sa mierne líši spôsob vyhodnocovanie vstupov.

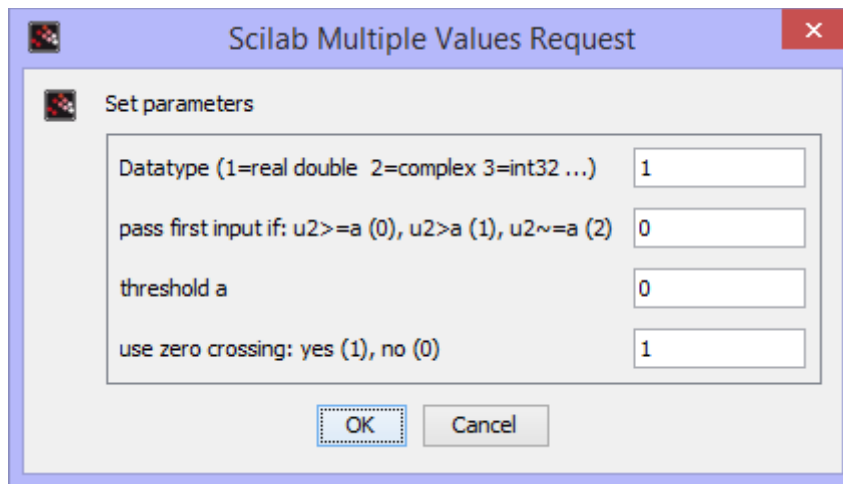


- **Datatype(1=real double 2=Complex 3=int32 ...):** nastaví dátový typ vstupu/výstupu. Podporuje všetky dátové typy.
- **Number of inputs:** udáva počet regulárnych vstupov.
- **Initial connected input:** celé číslo, ktoré udáva číslo vstupu, ktorý má byť pripojený na výstup ako prvý. Musí to byť číslo od 1 do celkového počtu vstupov.

1.11.15 SWITCH2_m



Daný blok vyberá medzi prvým a tretím vstupom na základe hodnoty druhého vstupu (riadiaceho).

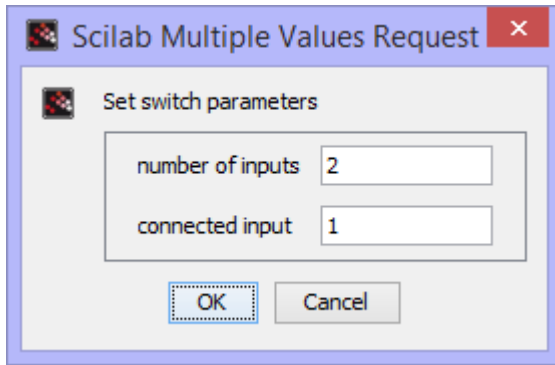


- **Datatype (1=real double 2=complex):** skalár, ktorý udáva dátový typ vstupov a výstupov.
- **Pass first input if: $u_2 \geq a$ (0), $u_2 > a$ (1), $u_2 \sim a$ (2):** vyberie podmienky, za ktorých prejde prvý vstup. Je možné vytvoriť blok či kontrolný vstup, ktorý je väčší alebo rovný hodnote a alebo je nenulový. Ak riadiaci vstup spĺňa podmienku tohto parametra, potom je prvý vstup odovzdaný na výstup. V opačnom prípade je odovzdaný tretí vstup.
- **Threshold a:** parameter určujúci, prahovú hodnotu a .
- **Use zero crossing: yes (1), no (0):** povolenie detekcie prechodu nulou.

1.11.16 SWITCH_f



Toto je manuálny prepínač. Na výstup privedie len výstup, ktorý je definovaný parametrom „Connected input“.



- **Number of inputs:** udáva počet vstupných portov.
- **Connected input:** celé číslo, ktoré musí byť od 1 do celkového počtu vstupov.

1.11.17 SELF_SWITCH



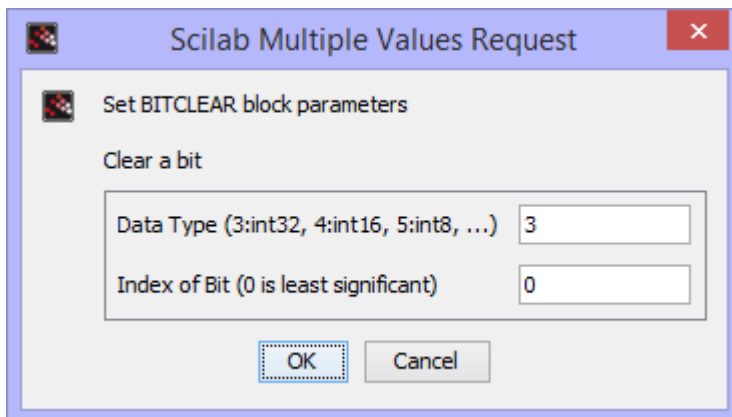
Pre povolenie alebo zablokovanie prechodu signálu cez daný blok stačí 2x kliknúť a prepínač zmení svoj stav na vypnutý/zapnutý.

1.12 Integer

1.12.1 BITCLEAR



Tento blok nastavuje určitý vybraný bit vstupu na 0.

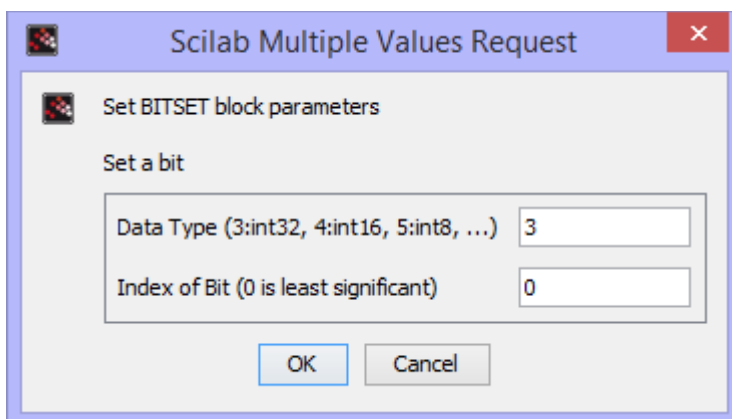


- **Data Type (3:int32, 4:int16, 5:int8, ...):** definuje celočíselný dátový typ vstupu/výstupu. Parameter môže nadobúdať hodnoty 3 až 8.
- **Index of Bit (0 is least significant):** indikuje index bitu, ktorý má byť vynulovaný.

1.12.2 BITSET



Tento blok nastavuje určitý vybraný bit vstupu na 1.

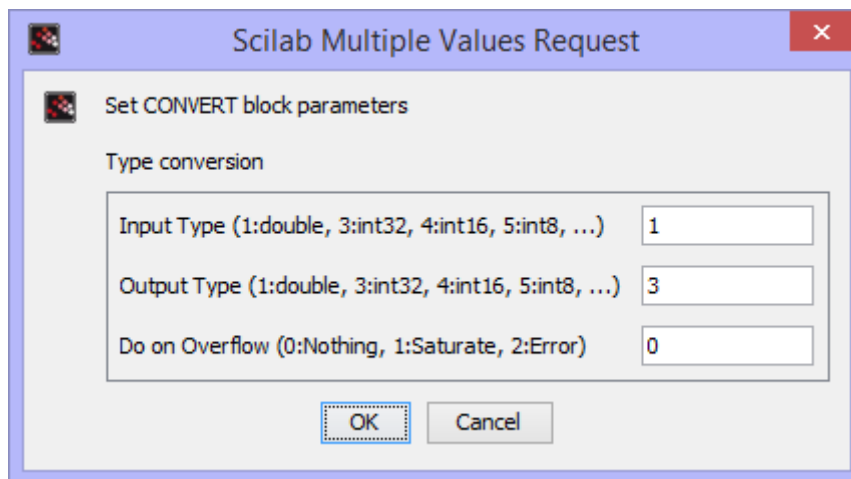


- **Data Type (3:int32, 4:int16, 5:int8, ...):** definuje celočíselný dátový typ vstupu/výstupu. Parameter môže nadobúdať hodnoty 3 až 8.
- **Index of Bit (0 is least significant):** indikuje index bitu, ktorý má byť vynulovaný.

1.12.3 CONVERT

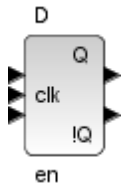


Daný blok konvertuje vstupný signál typu „real double“ alebo „integer“ na typ „integer“ alebo „real double“. To je užitočné hlavne keď je nutné aby komunikovali dva rôzne bloky s rôznymi dátovými typmi.



- **Input Type (1:double, 3:int32, 4:int16, 5:int8, ...):** definuje vstupný dátový typ. To môže byť „double“ alebo „integer“.
- **Output Type (1:double, 3:int32, 4:int16, 5:int8, ...):** definuje výstupný dátový typ. To môže byť „double“ alebo „integer“.
- **Do on Overflow (0:Nothing, 1:Saturate, 2:Error):** keď nastane pretečenie môžu nastať tri stavy:
 - 0: nič nedeje, výsledok nie je nasýtený,
 - 1: výsledok je nasýtený.
 - 2: simulácia ohlási chybové hlásenie.

1.12.4 DFLIPFLOP

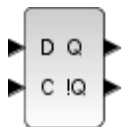


Na výstup Q prichádza vstupný port D , keď je povolený vstup en a zároveň reaguje len na vzostupnú hranu hodinového vstupu clk . Výstup $!Q$ je logická negácia Q .

Pravdivostná tabuľka tohto bloku:

clk	en	D	Q_n	$!Q_n$
↑	0	X	Q_{n-1}	$!Q_{n-1}$
↑	1	0	0	1
↑	1	1	1	0
nerastúci	X	X	Q_{n-1}	$!Q_{n-1}$

1.12.5 DLATCH



Tento blok kopíruje vstupný stav (D) na výstup (Q), keď je povolený vstup (C) je väčší a v nastavení sa prejaví ako transparentný. Výstup $!Q$ je logická negácia Q . Keď povolený vstup dosiahne nízkej úrovne, výstup udrzuje predchádzajúci stav a pôsobí ako pamäť.

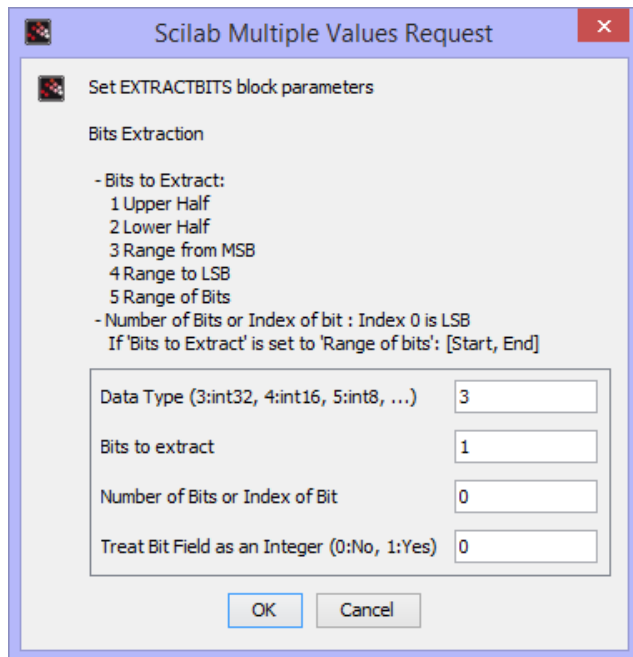
Pravdivostná tabuľka tohto bloku:

C	D	Q_n	$!Q_n$	
0	X	Q_{n-1}	$!Q_{n-1}$	žiadna zmena
1	0	0	1	reštart
1	1	1	0	nastav

1.12.6 EXTRACTBITS



Pre celočíselný vstup bloku je výstupom súvislý výber bitov.

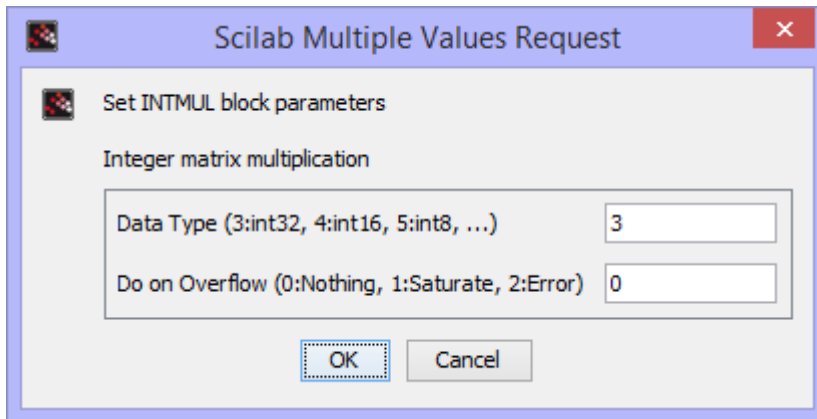


- **Data Type (3:int32, 4:int16, 5:int8, ..):** indikuje celočíselný dátový typ na vstupe.
- **Bits to Extract:** voľba režimu na extrakciu bitov zo vstupných dát (1: horná polovica, 2: dolná polovica, 3: rozsah od bitu s najvyššou hodnotou, 4: rozsah od bitu s najnižšou hodnotou, 5: rozsah bitov).
- **Number of Bits or Index of Bit:** keď je parameter „Bits to Extract“ nastavený na:
 - 1 alebo 2: tento parameter je ignorovaný,
 - 3 alebo 4: parameter sa používa na stanovenie počtu bitov na extrakciu.
 - 5: parameter sa používa na stanovenie rozsahu bitov na extrakciu, musí to byť v tvare vektoru [*Start*, *End*]. *Start* musí byť menší ako *End*. Maximálna hodnota závisí od veľkosti daného celočíselného dátového typu.
- **Treat Bit Field as an Integer (0:No, 1:Yes):** nastavuje či má blok pracovať s bitovým poľom ako s celočíselným dátovým typom.

1.12.7 INTMUL

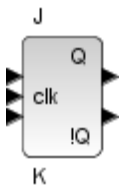


Blok násobí dve vstupné matice celých čísel. Počet riadkov prvej matice musí byť rovnaký ako počet stĺpcov druhej matice.



- **Data Type (3:int32, 4:int16, 5:int8, ..):** indikuje celočíselný dátový typ na vstupe.
- **Do on Overflow (0:Nothing, 1:Saturate, 2:Error):** keď nastane pretečenie môžu nastať tri stavy:
 - 0: nič nedeje, výsledok nie je nasýtený,
 - 1: výsledok je nasýtený.
 - 2: simulácia ohlási chybové hlásenie.

1.12.8 JKFLIPFLOP



Daný blok je najuniverzálnejší z kategórie „flip-flops“. Má dva vstupy označované ako **J(Set)** a **K(Reset)**.

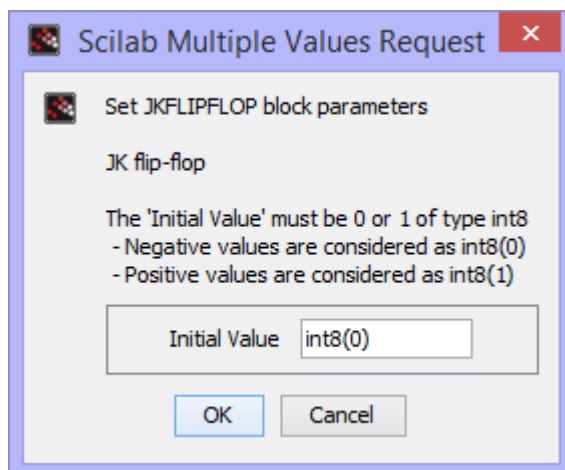
- Keď sú tieto vstupy rozdielne, výstup **Q** nadobúda hodnotu **J** hneď pri ďalšej zostupnej hrane.
- Keď sú oba vstupy nízke, na výstupe nenastane žiadna zmena.

- Keď sú oba vstupy vysoké ako výstup Q je možné prepínať z jedného stavu do druhého. Je možné vykonávať funkcie **set/reset (SR)** a zároveň nemôže dôjsť k nejednoznačným stavom.

Výstup $!Q$ je logická negácia Q .

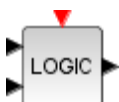
Pravdivostná tabuľka bloku:

J	K	Q_n	$!Q_n$	
0	0	Q_{n-1}	$!Q_{n-1}$	zadržanie
0	1	0	1	reštart
1	0	1	0	nastavenie
1	1	$!Q_{n-1}$	Q_{n-1}	prepínač



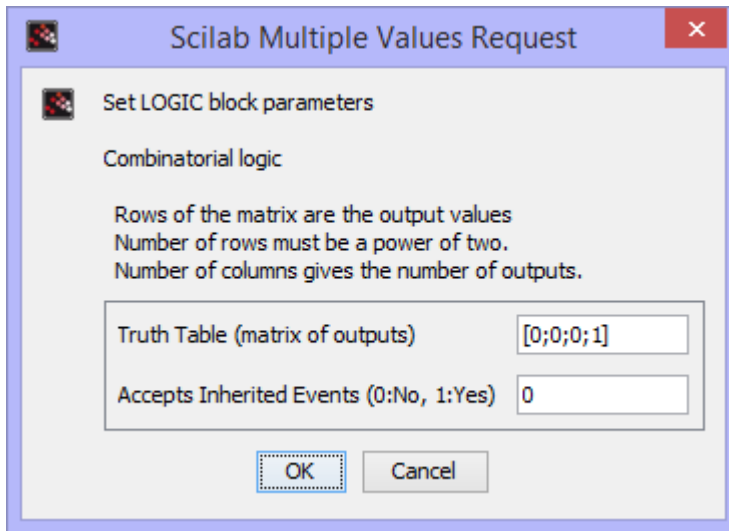
- **Initial Value:** počiatočný stav výstupu Q .

1.12.9 LOGIC



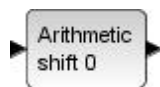
Blok implementuje štandardnú pravdivostnú tabuľku pre programovacie pole, digitálny obvod a akýchkoľvek iných booleovských výrazov. Užívateľ musí zadať maticu, ktorá definuje všetky možné výstupy bloku. Preto počet riadkov musí byť mocnina dvoch. Každý riadok matice obsahuje logickú kombináciu vstupných prvkov. Blok podporuje všetky

celočíselné dátové type (integer). Zároveň kladné čísla sa zapíšu ako logická 1 a záporné alebo 0 sa zapíšu ako logická 0.

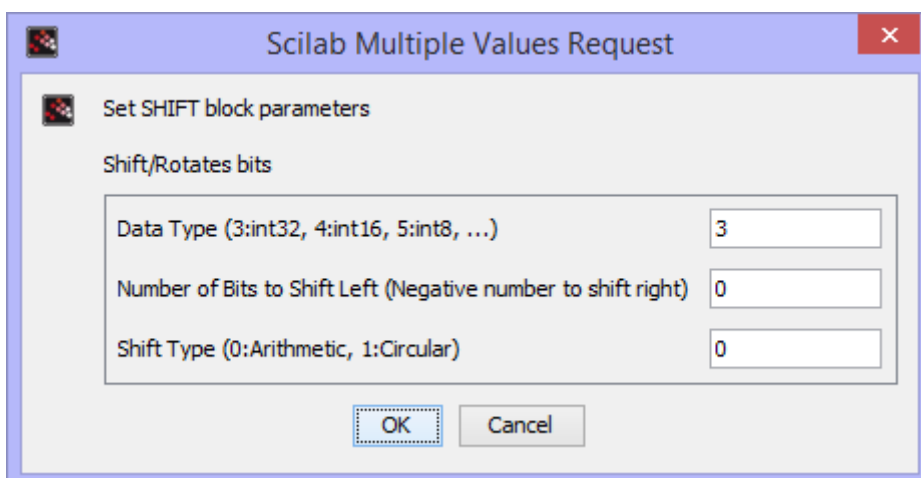


- **Truth Table:** maticový výstup, jednotlivé prvky musia byť 0 alebo 1.
- **Accepts Inherited Events (0:No, 1:Yes):** určí či sa majú dediť hodiny alebo nie.

1.12.10 SHIFT



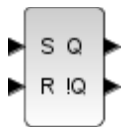
Tento blok posúva bity vstupného signálu. V tejto operácii sú bity posunuté doprava alebo doľava. Pravidlo pre posun bitov môže užívateľ meniť.



- **Data Type (3:int32, 4:int16, 5:int8, ...):** udáva dátový typ vstupu/výstupu.

- **Number of Bits to Shift Left (Negative number to shift right):** parameter udávajúci počet bitov vstupného signálu, ktoré sú posunuté/rotované. Kladná hodnota značí posun doľava, záporná hodnota posun doprava.
- **Shift Type (0:Arithmetic, 1:Circular):** nastavuje pravidlo, ktoré sa má použiť na posun bitov. Pre 0 je to aritmetický posun a pre 1 je to kruhový posun.

1.12.11 SRFLIPFLOP

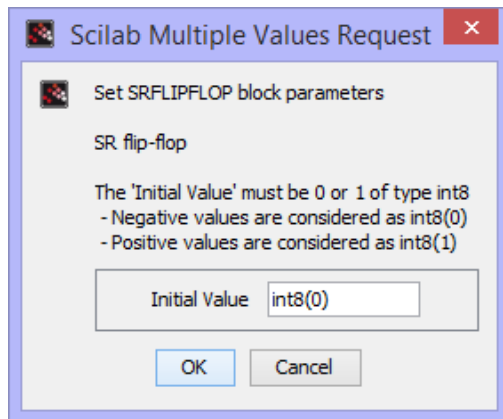


Tento blok je najjednoduchší z kategórie blokov „flip-flop“. Výstup Q závisí od vstupov S a R . Výstup $!Q$ je logická negácia Q .

- Ak je hodnota S (Set) vysoká, zatiaľ čo R je nízka, potom Q nadobúda vysokej hodnoty a zostáva vysoká aj keď S vráti nízku.
- Ak je hodnota R (Reset) vysoká, zatiaľ čo S je nízka, potom Q nadobúda nízku hodnotu a zostáva nízka aj keď R vráti nízku.
- Keď S a R sú obe vysoké, Q aj $!Q$ nadobúdajú vysokú alebo nízku hodnotu. Stav je nestabilný a preto je prakticky zakázaný.

Pravdivostná tabuľka bloku:

S	R	Q_n	$!Q_n$	
0	0	Q_{n-1}	$!Q_{n-1}$	Zadržanie
0	1	0	1	Reštart
1	0	1	0	Nastavenie
1	1	U	U	Zakázaný stav



- **Initial Value:** počiatočná hodnota stavu Q . Musí byť dátového typu int8.

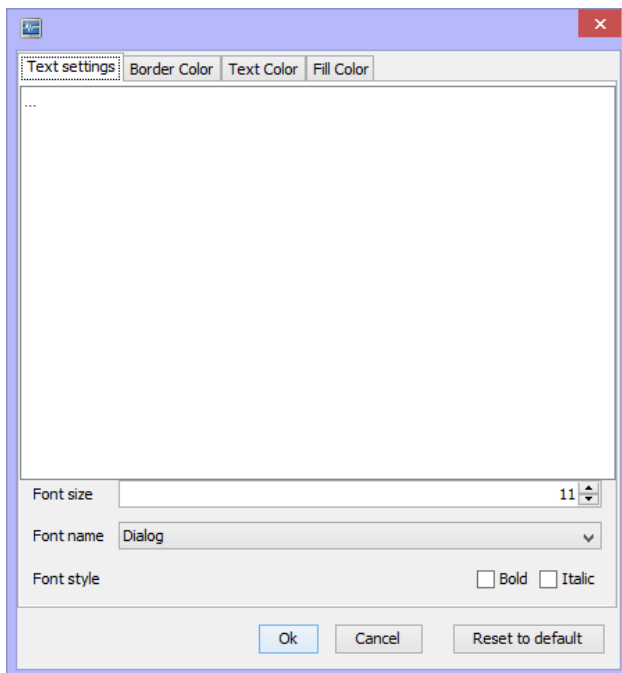
1.13 Annotations

1.13.1 TEXT_f



Tento špeciálny blok slúži na pridávanie textu kdekoľvek v diagrame, pritom nemá žiadny vplyv na simuláciu. Po vložení stačí dvojklikom aktivovať zmenu bloku a potom je možné dopisovať text. Pre viac možností formátovania textu stačí na blok kliknúť pravým, zobrazí sa rolovacie menu. Ďalej vyberieme „Format“ -> „Edit“. Otvorí sa okno, kde sú parametre pre rôzne nastavenie textu.

Týmto spôsobom možno meniť aj vzhľad ostatných blokov.

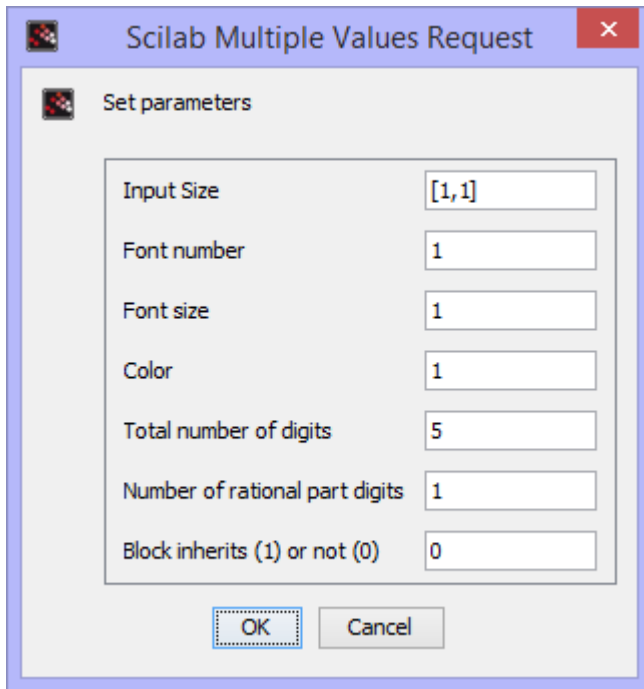


1.14 Sinks

1.14.1 AFFICH_m



Tento blok zobrazuje hodnotu riadiaceho vstupu na svojej maske priamo počas simulácie.

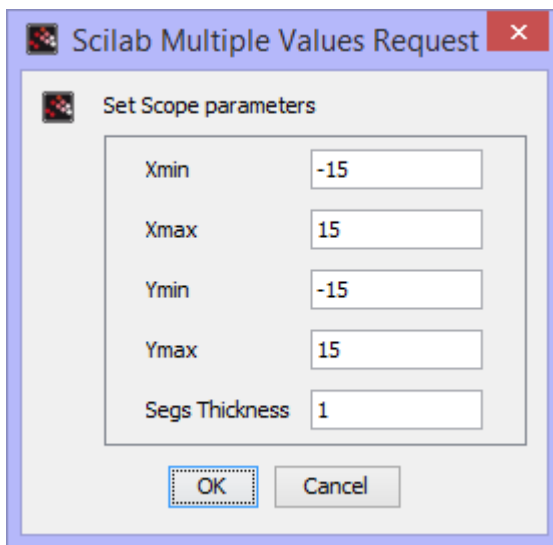


- **Input Size:** explicitne definovaná veľkosť vstupného portu.
- **Font number:** celé číslo, ktoré určuje typ písma.
- **Font size:** celé číslo, ktoré určuje veľkosť písma.
- **Color:** celé číslo, určujúce farbu textu.
- **Total number of digits:** celé číslo väčšie ako 3. Udáva maximálny počet číslic, ktoré sa majú zobrazovať.
- **Number of rational part digits:** celé číslo väčšie alebo rovné 0. Je to číslo udávajúce počet číslic reprezentujúcich racionálnu časť čísla.
- **Block inherits (1) or not (0):** možnosť dedenia udalosti z regulárneho vstupu alebo z riadiaceho vstupu (0).

1.14.2 BARXY



Blok realizuje vizualizáciu vývoja dvoch pravidelných vstupných signálov vykreslením rovných čiar medzi dvoma po sebe nasledujúcimi vstupmi, ktoré predstavujú súradnice x a y .

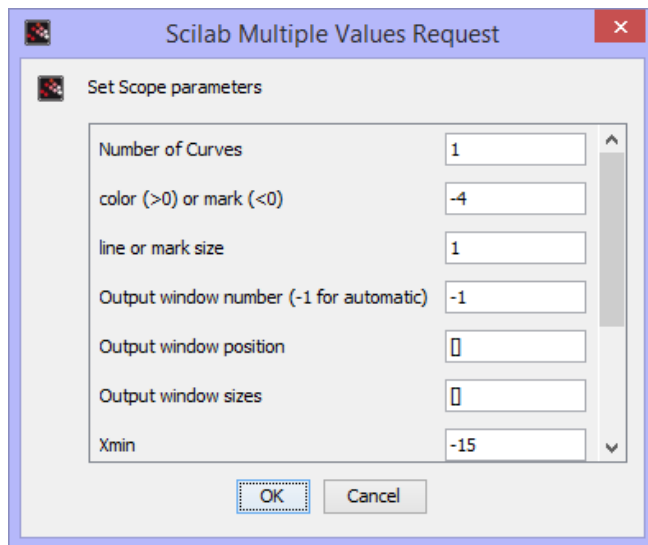


- **Xmin:** minimálna hodnota prvého vstupu. Zároveň definuje osu X v zobrazovacom okne.
- **Xmax:** maximálna hodnota prvého vstupu. Zároveň definuje osu X v zobrazovacom okne.
- **Ymin:** minimálna hodnota druhého vstupu. Zároveň definuje osu Y v zobrazovacom okne.
- **Ymax:** maximálna hodnota druhého vstupu. Zároveň definuje osu Y v zobrazovacom okne.
- **Segs Thickness:** hrúbka čiary, ktorá sa bude zobrazovať v okne.

1.14.3 CANIMXY



Realizuje vizualizáciu vývoja dvoch pravidelných vstupných signálov pomocou vykresľovania druhého vstupu ako funkcie prvého v okamihoch vstupu riadiacich signálov na riadiaci vstupný port. Keď je bol vykreslený na obrazovku, zostane tam až kým vyrovnávacia pamäť nedosiahne maximálny limit. Tento priestor je vhodný na jednoduché animácie.



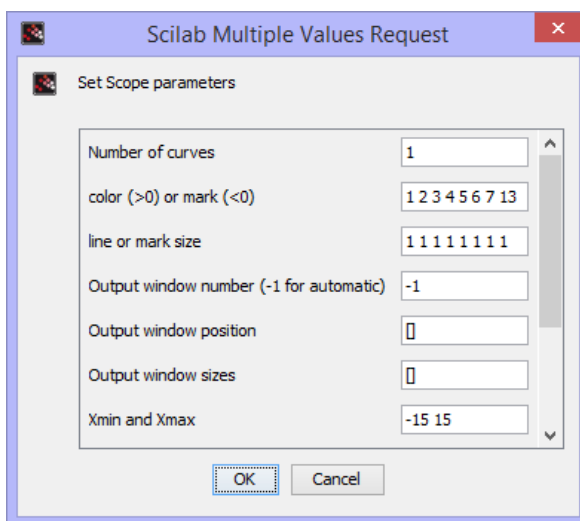
- **Number of Curves:** nastaví počet kriviek.
- **Color (>0) or mark (<0):** celé číslo, ktoré udáva číslo farby (≥ 0) alebo typ značky (< 0), používa na vykreslenie vývoja signálu vstupného portu.
- **Line or mark size:** celé číslo.
- **Output window number (-1 for automatic):** udáva počet grafických okien. Zvlášť vhodné pri vysokých hodnotách, aby nedochádzalo ku konfliktom.
- **Output window position:** vektor s dvoma prvkami, špecifikujúci súradnice ľavého horného rohu grafického okna. [] pre základné zastavenie.
- **Output window sizes:** vektor s dvoma prvkami, špecifikujúci výšku a šírku grafického okna. [] pre základné nastavenie.
- **Xmin:** minimálna hodnota druhého vstupu. Zároveň definuje osu X v zobrazovacom okne.
- **Xmax:** maximálna hodnota prvého vstupu. Zároveň definuje osu X v zobrazovacom okne.

- **Ymin:** minimálna hodnota druhého vstupu. Zároveň definuje osu Y v zobrazovacom okne.
- **Ymax:** maximálna hodnota druhého vstupu. Zároveň definuje osu Y v zobrazovacom okne.
- **Buffer size:** celočíselná hodnota, ktorá udáva veľkosť vyrovnávacej pamäte.

1.14.4 CANIMXY3D



Blok realizujúci vizualizáciu vývoja troch pravidelných vstupných signálov pomocou vykresľovania tretieho vstupu ako funkciu ostatných dvoch vstupov v okamihu prítomnosti udalosti na riadiacom porte. Keď je bol vykreslený na obrazovku, zostane tam až kým vyrovnávacia pamäť nedosiahne maximálny limit. Tento priestor je vhodný na jednoduché animácie.



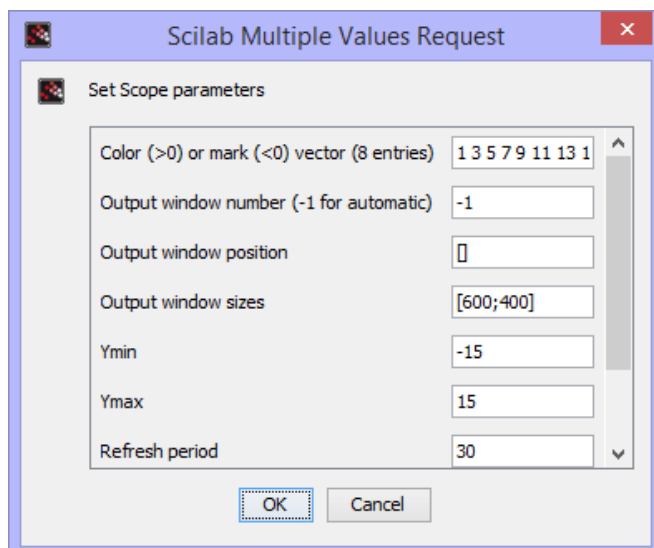
- **Number of Curves:** nastaví počet kriviek.
- **Color (>0) or mark (<0):** celé číslo, ktoré udáva číslo farby (≥ 0) alebo typ značky (< 0), používa na vykreslenie vývoja signálu vstupného portu.
- **Line or mark size:** celé číslo.
- **Output window number (-1 for automatic):** udáva počet grafických okien. Zvlášť vhodné pri vysokých hodnotách, aby nedochádzalo ku konfliktom.

- **Output window position:** vektor s dvoma prvkami, špecifikujúci súradnice ľavého horného rohu grafického okna. [] pre základné zastavenie.
- **Output window sizes:** vektor s dvoma prvkami, špecifikujúci výšku a šírku grafického okna. [] pre základné nastavenie.
- **Xmin and Xmax:** minimálna a maximálna hodnota druhého vstupu. Zároveň definuje osu X v zobrazovacom okne.
- **Ymin and Ymax:** minimálna a maximálna hodnota druhého vstupu. Zároveň definuje osu Y v zobrazovacom okne.
- **Zmin and Zmax:** minimálna a maximálna hodnota pre tretí vstup. Zároveň definuje osu Z v zobrazovacom okne.
- **Alpha and Theta:** nastaví hodnoty *Alpha* a *Theta* pre 3D zobrazenie.
- **Buffer size:** celočíselná hodnota, ktorá udáva veľkosť vyrovnávacej pamäte.

1.14.5 CFSCOPE



Jedná sa o blok zobrazujúci priebeh riadiaceho signálu privedeného na jeho vstupný port.



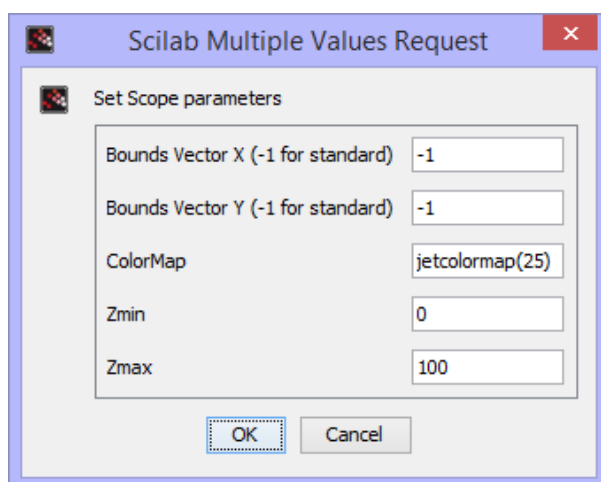
- **Color (>0) or mark (<0):** celé číslo, ktoré udáva číslo farby (≥ 0) alebo typ značky (< 0), používa na vykreslenie vývoja signálu vstupného portu.
- **Output window number (-1 for automatic):** udáva počet grafických okien. Zvlášť vhodné pri vysokých hodnotách, aby nedochádzalo ku konfliktom.

- **Output window position:** vektor s dvoma prvkami, špecifikujúci súradnice ľavého horného rohu grafického okna. [] pre základné zastavenie.
- **Output window sizes:** vektor s dvoma prvkami, špecifikujúci výšku a šírku grafického okna. [] pre základné nastavenie.
- **Ymin:** minimálna hodnota druhého vstupu. Zároveň definuje osu Y v zobrazovacom okne.
- **Ymax:** maximálna hodnota druhého vstupu. Zároveň definuje osu Y v zobrazovacom okne.
- **Refresh period:** maximálna hodnota na ose X (čas). Keď hodnota dosiahne násobku tohto parametru, priebeh funkcie sa prekreslí.
- **Buffer size:** celočíselná hodnota, ktorá udáva veľkosť vyrovnávacej pamäte.
- **Links to view:** parameter povoľujúci odkazovať na sa daný výstup špeciálnym odkazom.

1.14.6 CMAT3D



Blok zobrazujúci hodnoty matice v rozsahu súradnicového systému (xy).



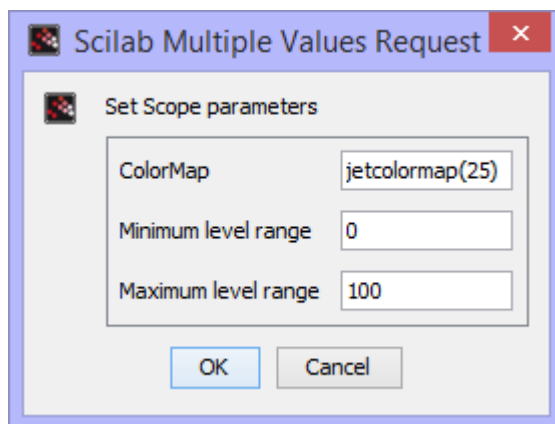
- **Bounds Vector X (-1 for standard):** ak ostane parameter -1 hodnoty vektoru by mali byť v rozsahu 0 až 1. V opačnom prípade je možné zadať vlastný vektor.

- **Bounds Vector Y (-1 for standard):** ak ostane parameter -1 hodnoty vektoru by mali byť v rozsahu 0 až 1. V opačnom prípade je možné zadať vlastný vektor.
- **ColorMap:** je to rozsah farieb odkazujúci na výstupné okno pre zobrazovanie. Možno zadať parametre „jetcolormap“, „hotcolormap“, „graycolormap“ alebo vlastnú farebnú mapu.
- **Zmin:** minimálna hodnota na ose Z.
- **Zmax:** maximálna hodnota na ose Z.

1.14.7 CMATVIEW



Zobrazuje hodnoty matice v určitej farebnej škále.

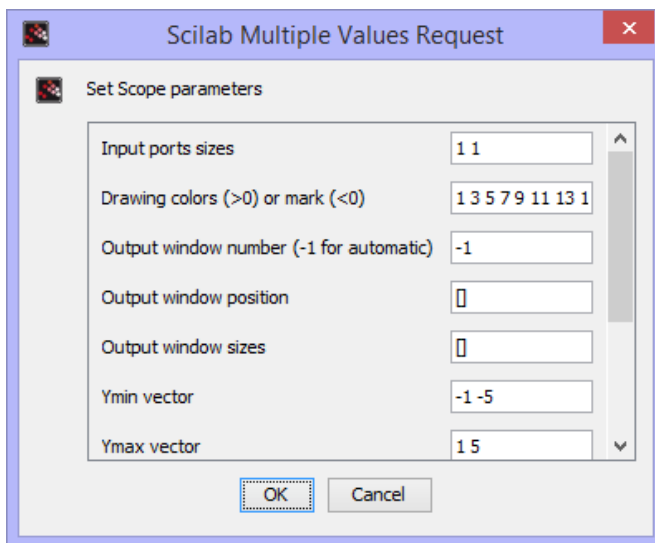


- **ColorMap:** je to rozsah farieb odkazujúci na výstupné okno pre zobrazovanie. Možno zadať parametre „jetcolormap“, „hotcolormap“, „graycolormap“ alebo vlastnú farebnú mapu.
- **Minimum level range:** minimálna hodnota, ktorá prichádza na regulárny vstupný port. To môže odkazovať na „cold value (chladnú hodnotu)“ farebnej škály.
- **Maximum level range:** maximálna hodnota, ktorá prichádza na regulárny vstupný port. To môže odkazovať na „hot value (horúcu hodnotu)“ farebnej škály.

1.14.8 CMSCOPE



Keď je simulácia aktívna, systém spustí tento blok. Blok zobrazuje priebeh vstupného signálu s ohľadom na simulačný čas. Blok podporuje viacero osí (jedna na každý port). Všetky osi majú spoločný časový rozsah s nezávislými osami Y.



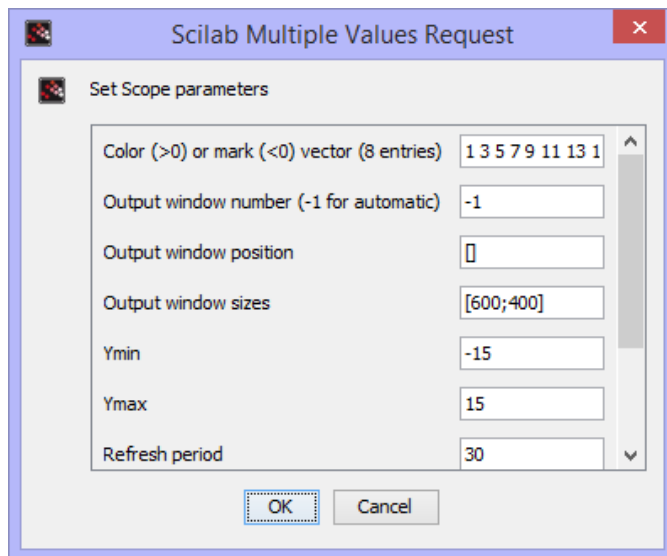
- **Input ports sizes:** definuje počet vstupných portov.
- **Drawing colors (>0) or mark (<0):** celé číslo, ktoré udáva číslo farby (>0) alebo typ značky (<0), používa na vykreslenie vývoja signálu vstupného portu.
- **Output window number (-1 for automatic):** udáva počet grafických okien. Zvlášť vhodné pri vysokých hodnotách, aby nedochádzalo ku konfliktom.
- **Output window position:** vektor s dvoma prvkami, špecifikujúci súradnice ľavého horného rohu grafického okna. [] pre základné zastavenie.
- **Output window sizes:** vektor s dvoma prvkami, špecifikujúci výšku a šírku grafického okna. [] pre základné nastavenie.
- **Ymin vector:** minimálna hodnota vstupu. Zároveň definuje osu X v zobrazovacom okne.
- **Ymax vector:** maximálna hodnota vstupu. Zároveň definuje osu Y v zobrazovacom okne.
- **Buffer size:** celočíselná hodnota, ktorá udáva veľkosť vyrovnávacej pamäte.
- **Accept inherited events 0/1:**
 - ak je 0, blok vykreslí nový bod iba vtedy, keď príde riadiaci signál.

- ak je 1, blok vykreslí nový bod iba vtedy, keď príde riadiaci signál alebo keď príde k zmenám v dôsledku riadiaceho signálu v inom predchádzajúcom bloku.
- **Name of Scope:** označenie/názov bloku.

1.14.9 CSCOPE



Blok zobrazuje signál zo vstupného portu s ohľadom na simulačný čas. Obe osi majú spoločný rozsah. Blok umožňuje nastaviť dobu a rozsah zobrazovaných vstupných hodnôt.



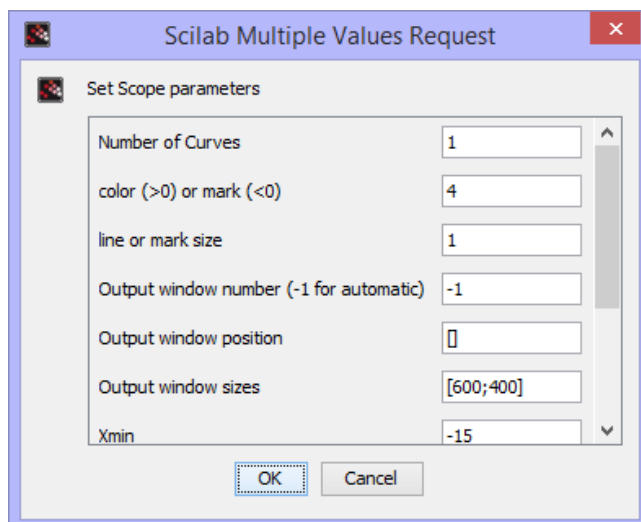
- **Color (>0) or mark (<0):** celé číslo, ktoré udáva číslo farby (≥ 0) alebo typ značky (< 0), používa na vykreslenie vývoja signálu vstupného portu.
- **Output window number (-1 for automatic):** udáva počet grafických okien. Zvlášť vhodné pri vysokých hodnotách, aby nedochádzalo ku konfliktom.
- **Output window position:** vektor s dvoma prvkami, špecifikujúci súradnice ľavého horného rohu grafického okna. [] pre základné zastavenie.
- **Output window sizes:** vektor s dvoma prvkami, špecifikujúci výšku a šírku grafického okna. [] pre základné nastavenie.
- **Ymin:** minimálna hodnota druhého vstupu. Zároveň definuje osu Y v zobrazovacom okne.

- **Ymax:** maximálna hodnota druhého vstupu. Zároveň definuje osu Y v zobrazovacom okne.
- **Refresh period:** maximálna hodnota na ose X (čas). Keď hodnota dosiahne násobku tohto parametru, priebeh funkcie sa prekreslí.
- **Buffer size:** celočíselná hodnota, ktorá udáva veľkosť vyrovnávacej pamäte.
- **Accept inherited events 0/1:**
 - ak je 0, blok vykreslí nový bod iba vtedy, keď príde riadiaci signál.
 - ak je 1, blok vykreslí nový bod iba vtedy, keď príde riadiaci signál alebo keď príde k zmenám v dôsledku riadiaceho signálu v inom predchádzajúcom bloku.
- **Name of Scope:** označenie/názov bloku.

1.14.10 CSCOPXY



Blok realizuje vizualizáciu vývoja dvoch regulárnych vstupných signálov vykreslením druhého vstupu ako funkciu prvého v okamihu príchodu riadiaceho signálu. Keď je bod vykreslený v danom okne, zostane tam kým sa neukončí simulácia.

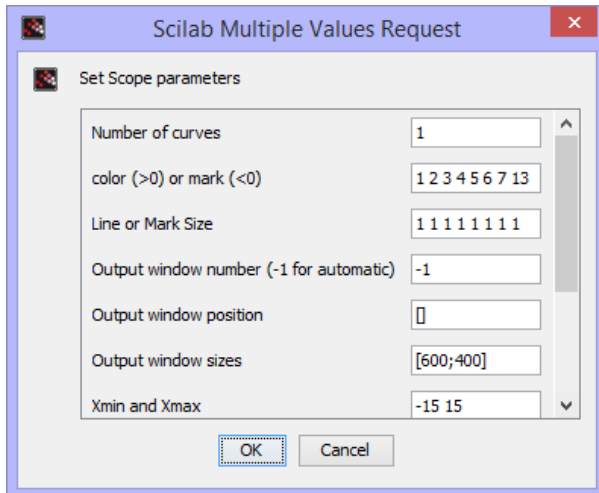


- **Number of Curves:** nastaví počet kriviek.
- **Color (>0) or mark (<0):** celé číslo, ktoré udáva číslo farby (>=0) alebo typ značky (<0), používa na vykreslenie vývoja signálu vstupného portu.
- **Line or mark size:** celé číslo.
- **Output window number (-1 for automatic):** udáva počet grafických okien. Zvlášť vhodné pri vysokých hodnotách, aby nedochádzalo ku konfliktom.
- **Output window position:** vektor s dvoma prvkami, špecifikujúci súradnice ľavého horného rohu grafického okna. [] pre základné nastavenie.
- **Output window sizes:** vektor s dvoma prvkami, špecifikujúci výšku a šírku grafického okna. [] pre základné nastavenie.
- **Xmin:** minimálna hodnota druhého vstupu. Zároveň definuje osu X v zobrazovacom okne.
- **Xmax:** maximálna hodnota prvého vstupu. Zároveň definuje osu X v zobrazovacom okne.
- **Ymin:** minimálna hodnota druhého vstupu. Zároveň definuje osu Y v zobrazovacom okne.
- **Ymax:** maximálna hodnota druhého vstupu. Zároveň definuje osu Y v zobrazovacom okne.
- **Buffer size:** celočíselná hodnota, ktorá udáva veľkosť vyrovnávacej pamäte.

1.14.11 CSCOPXY3D

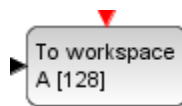


Tento blok realizuje vizualizáciu vývoja troch vstupných portov vykreslením tretieho vstupu ako funkciu ostatných dvoch vstupov v okamihu príchodu riadiaceho signálu na riadiaci port. Keď je bod vykreslený v danom okne, zostane tam kým sa neukončí simulácia.

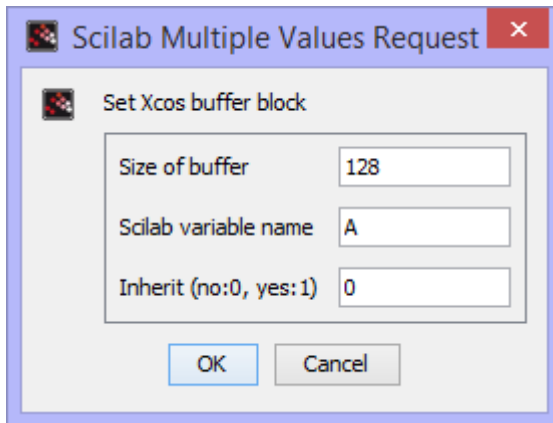


- **Number of Curves:** nastaví počet kriviek.
- **Color (>0) or mark (<0):** celé číslo, ktoré udáva číslo farby (≥ 0) alebo typ značky (<0), používa na vykreslenie vývoja signálu vstupného portu.
- **Line or mark size:** celé číslo.
- **Output window number (-1 for automatic):** udáva počet grafických okien. Zvlášť vhodné pri vysokých hodnotách, aby nedochádzalo ku konfliktom.
- **Output window position:** vektor s dvoma prvkami, špecifikujúci súradnice ľavého horného rohu grafického okna. [] pre základné zastavenie.
- **Output window sizes:** vektor s dvoma prvkami, špecifikujúci výšku a šírku grafického okna. [] pre základné nastavenie.
- **Xmin and Xmax:** minimálna a maximálna hodnota druhého vstupu. Zároveň definuje osu X v zobrazovacom okne.
- **Ymin and Ymax:** minimálna a maximálna hodnota druhého vstupu. Zároveň definuje osu Y v zobrazovacom okne.
- **Zmin and Zmax:** minimálna a maximálna hodnota pre tretí vstup. Zároveň definuje osu Z v zobrazovacom okne.
- **Alpha and Theta:** nastaví hodnoty *Alpha* a *Theta* pre 3D zobrazenie.
- **Buffer size:** celočíselná hodnota, ktorá udáva veľkosť vyrovnávacej pamäte.

1.14.12 TOWS_c



Blok sa využíva na presun dát do pracovného priestoru Scilab. Každý záznam obsahuje čas a hodnotu.



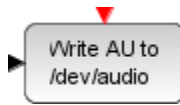
- **Size of buffer:** nastaví veľkosť vyrovnávacej pamäte. To udáva celkový počet vzoriek zaznamenaných počas simulácie.
- **Scilab variable name:** nastaví meno premennej v Scilabe. Simulácia musí byť ukončená aby bola daná premenná získaná.
- **Inherit (no:0, yes:1):** možnosť výberu dedenia udalosti z regulárneho vstupu alebo z riadiaceho vstupu (0).

1.14.13 TRASH_f

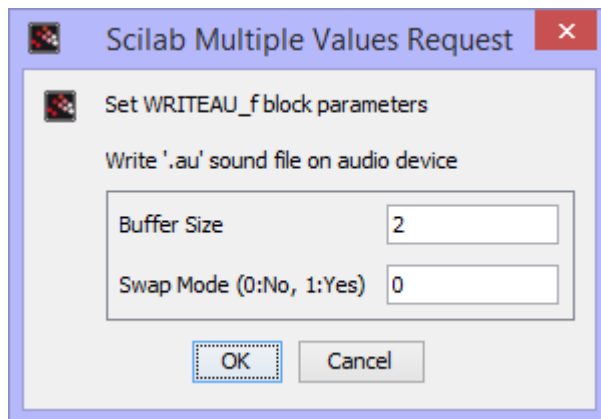


Toto je končový blok. Všetky prichádzajúce dáta budú stratené.

1.14.14 WRITEAU_f

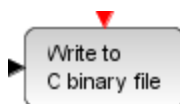


Tento blok zapisuje zvukový súbor definovaný príponou *.au. Jeden dátový kanál by mal zodpovedať jednému stĺpcu. Amplitúdové hodnoty mimo rozsah (-1;1) sú orezané pred zapísaním do súboru. Zápis tohto typu zvukového súboru podporuje viackanálové dáta pre 8 a 16 bitové formáty a 8 bitový „mu-law“ formát.

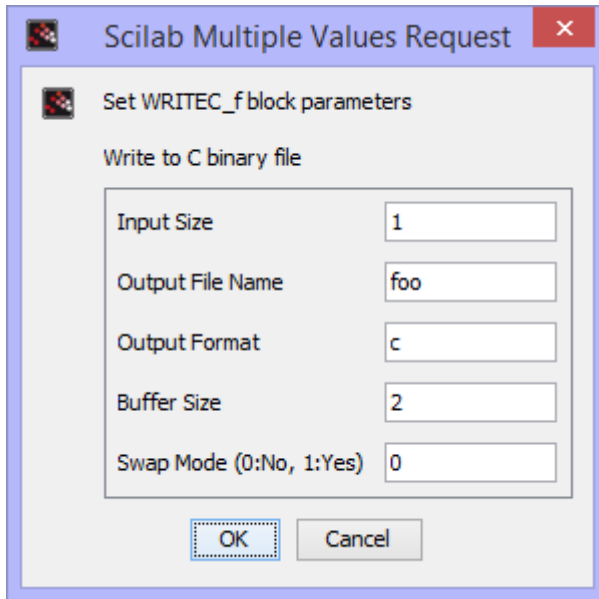


- **Buffer Size:** pre zvýšenie účinnosti je možné vstupné dáta dočasne uložiť do vyrovnávacej pamäte.
- **Swap Mode (0:No, 1:Yes):** v móde 1 je súbor kódovaný v „little endian IEEE“ formáte a v prípade potreby je formát zmenený aby sa zhodoval s formátom IEEE. Ak je nastavený mód 0 automatická zmena dát je vypnutá.

1.14.15 WRITEC_f



Tento blok umožňuje užívateľovi zapísať dáta do binárneho súboru v jazyku C s určitým definovaným názvom.



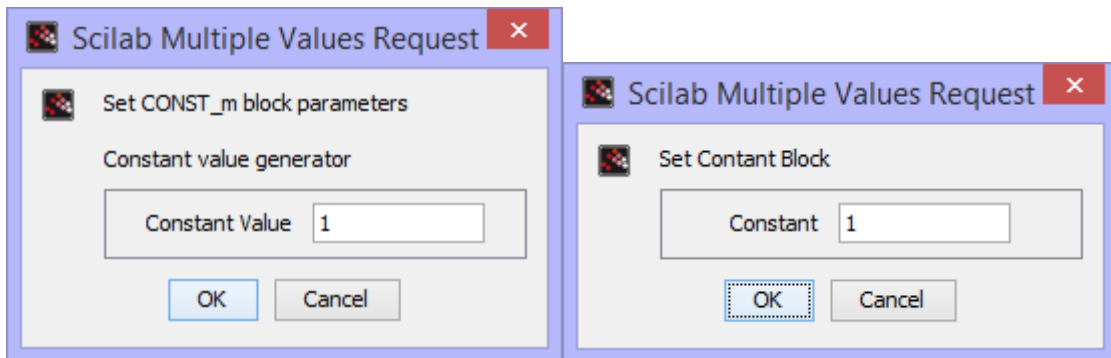
- **Input Size:** skalár udávajúci veľkosť vstupného stĺpcového vektoru.
- **Output File Name:** znakový reťazec definujúci cestu a názov súboru kde má byť súbor vytvorený.
- **Output Format:** znakový reťazec definujúci formát použitia. Použité sú reťazce: „l“, „i“, „s“, „ul“, „ui“, „us“, „d“, „f“, „c“, „uc“ respektíve ako dátové typy int32, int16, int8, uint32, uint16, uint8, double, float, char or unsigned char.
- **Buffer Size:** pre zvýšenie účinnosti je možné vstupné dáta dočasne uložiť do vyrovnávacej pamäte.
- **Swap Mode (0:No, 1:Yes):** v móde 1 je súbor kódovaný v „little endian IEEE“ formáte a v prípade potreby je formát zmenený aby sa zhodoval s formátom IEEE. Ak je nastavený mód 0 automatická zmena dát je vypnutá.

1.15 Sources

1.15.1 CONST, CONST_m, CONST_f

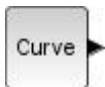


Tento blok pracuje ako generátor konštanty, ktorú si užívateľ zvolí cez parameter „Constant Value / Constant“.



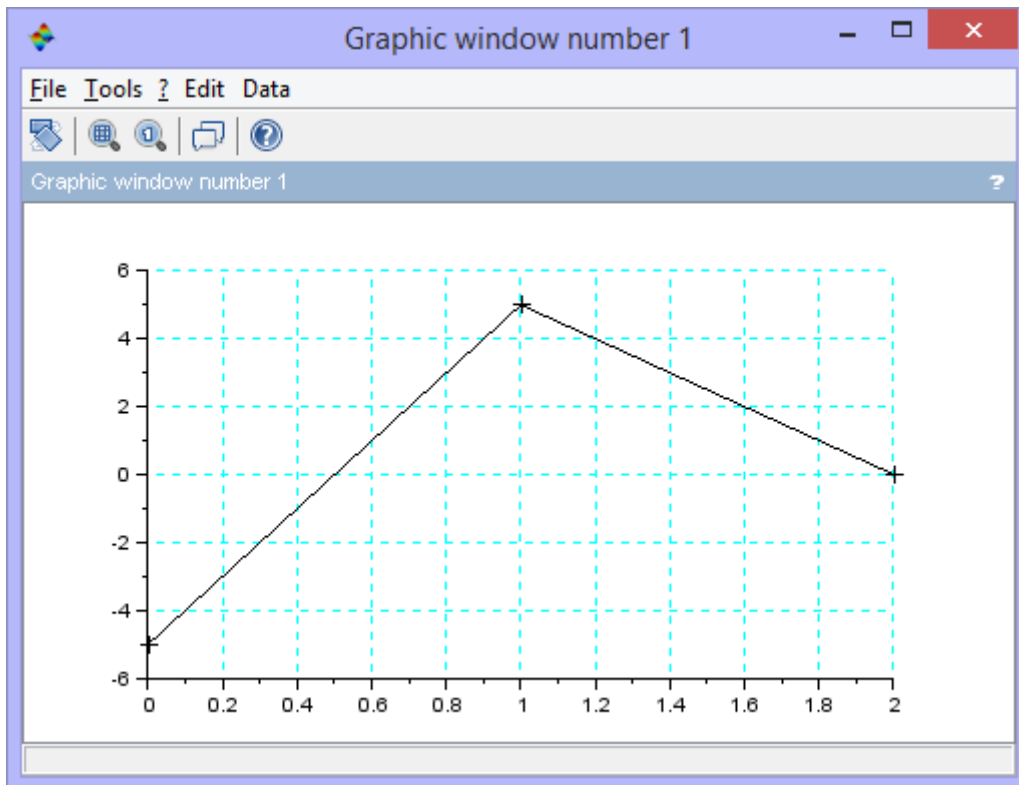
- **Constant value / Constant:** je to matica akéhokoľvek typu. Dátový typ bloku de- dia aj ostatné bloky.

1.15.2 CURV_f

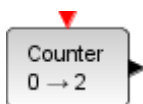


Blok definujúci časové funkcie v tabuľke. V jednotlivých bunkách blok vykonáva lineárnu interpoláciu. Výstupom v tomto bloku je posledná tabuľková hodnota. Užívateľ si môže definovať tabuľku s funkciou využitím editoru krivky.

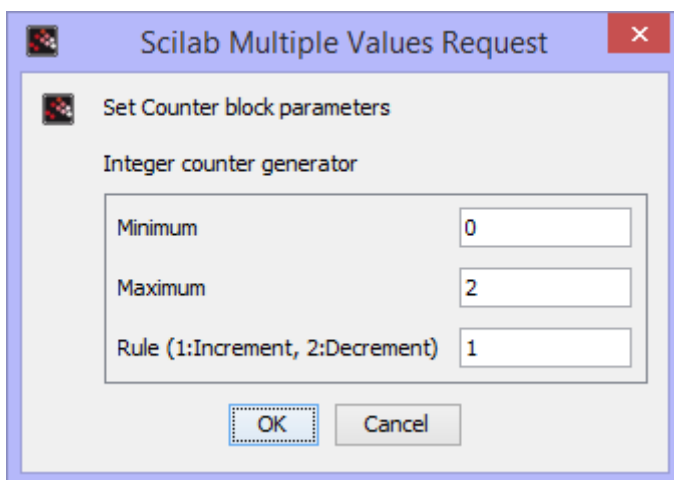
Jednotlivé body sa presúvajú kliknutím na bod, posunutím a následným znovu kliknutím. Nový bod sa pridáva dvojklikom.



1.15.3 Counter



Tento blok zvyšuje / znižuje danú hodnotu vždy o 1 v rozmedzí veličín „Minimum“ a „Maximum“.



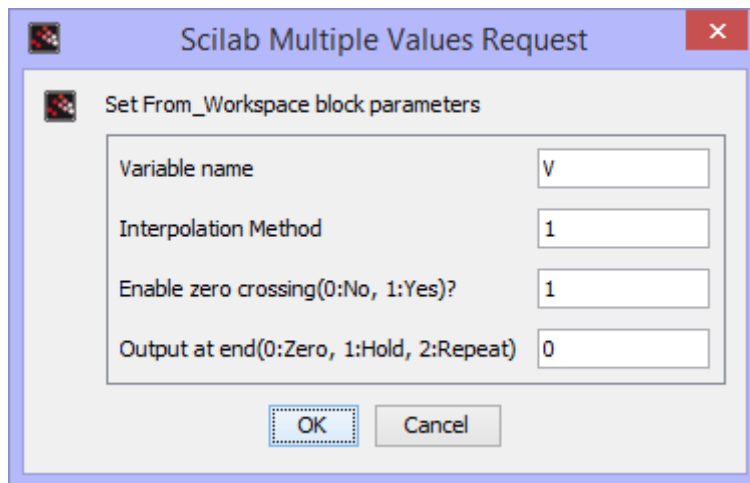
- **Minimum:** spodná hranica počítadla.
- **Maximum:** horná hranica počítadla.

- **Rule:** pri možnosti 1 (Increment) je hodnota zvyšovaná od spodnej hranice až po hornú, naopak pri možnosti 2 (Decrement) je hodnota znižovaná od hornej hranice až po spodnú.

1.15.4 FROMWSB



Blok slúži na získavanie dát, definovaných v pracovnom priestore Scilab a ich použitie v Xcos. Údaje by mali mať pole s časom a pole s hodnotou.



- **Variable name:** táto premenná je definovaná v Scilabe a mala byť zložená z dvoch polí, t.j. pole času o veľkosti $N \times 1$ (stĺpcový vektor) a pole hodnoty o veľkosti $N \times M$. Pole času by malo byť reálneho typu, zatiaľ čo pole hodnoty môže byť rôzneho dátového typu.
- **Interpolation Method:** premenné čítané cez Xcos sú údaje namerané pri diskretných okamihoch danej časovej oblasti. Táto voľba spôsobí, že blok interpoluje v časových krokoch, kde pre existujúce dáta neexistuje pracovný priestor. K dispozícii sú štyri rôzne metódy interpolácie.
 - 0 („Zero order method“): táto metóda generuje konštantný signál po častiach. Je prístupná pre všetky dátové typy.
 - 1 („Linear method“): táto metóda generuje lineárny signál po častiach. Je prístupná pre všetky dátové typy okrem double a complex. Pre tieto dátové typy môže byť použitá lineárna interpolácia, ale finálny výstup bude interpolovaný do pôvodného dátového typu.

- 2 („NATURAL method“): táto metóda je k dispozícií iba pre reálne a komplexné dátové typy.
- 3 („NOT_A_KNOT method“): táto metóda je k dispozícií iba pre reálne a komplexné dátové typy.
- **Enable zero crossing** (0: nie, 1: áno): Keď sú vybrané interpolačné metódy, budú údaje výstupného signálu prerušované v určitých časových okamihoch. To môže spôsobovať problémy pre numerické riešenia. Aby bolo možné spoľahlivo používať numerické integrácie, je nutné použiť túto metódu (1: Yes). Ak sú dáta použité na výstupe bloky numerickým riešením v nespojitých miestach, je generovaná diskretná simulácia a numerické riešenie je chladno reštartované.
- **Output at end:** táto voľba určuje metódu pre generovanie výstupu po poslednom časovom bode, ktorý bol k dispozícií z pracovného priestoru.
 - 0: výstup je nastavený na nulu.
 - 1: výstup je zadržaný.
 - 2: výstup sa opakuje z pracovného priestoru.

1.15.5 GENSIN_f

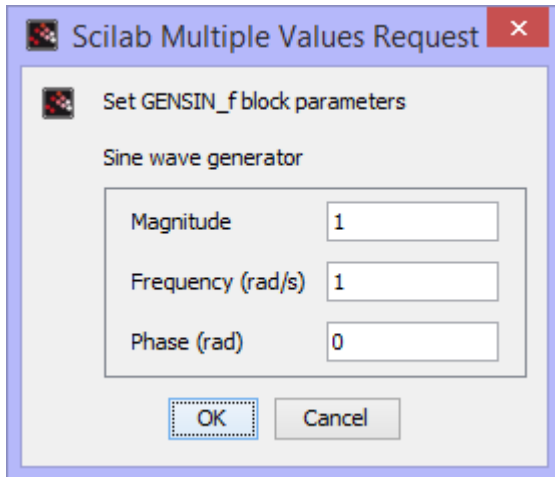


Blok generujúci funkciu sínus, ktorá je definovaná rovnicou:

$$Output = M \sin(F \cdot t + P)$$

Možnosti nastavenia:

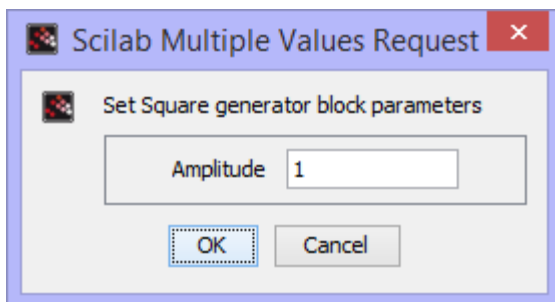
- Veľkosť M je nastavená parametrom „Magnitude“.
- Frekvencia F je nastavená parametrom „Frequency“ (rad/s).
- Počiatočná fáza P je nastavená parametrom „Phase“ (rad).



1.15.6 GENSQR_f



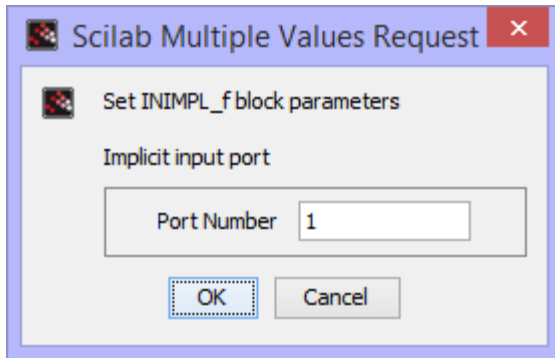
Tento blok generuje obdĺžnikový signál. Výstup nadobúda hodnoty $-M$ a $+M$. Zakaždým keď je na vstupnom porte prijatá úloha, výstup sa prepne z $-M$ do M alebo z M do $-M$.



1.15.7 INIMPL_f



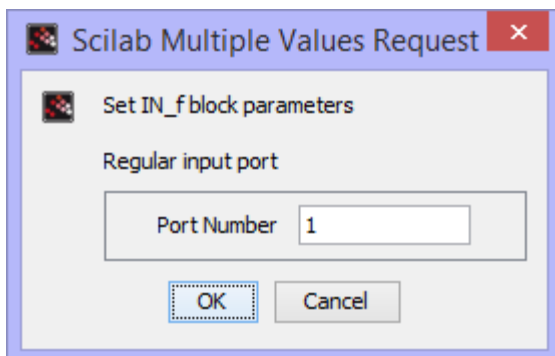
Blok predstavujúci implicitný vstupná port, ktorý musí byť použitý vo vnútri „Xcos Super Block“ a spojený s implicitným vstupom v subsysteme. Predstavuje odkaz na signál z vonku systému do subsystemu. Implicitné vstupné porty musia byť očíslované od čísla 1, to sa nastavuje parametrom „Port Number“.



1.15.8 IN_f



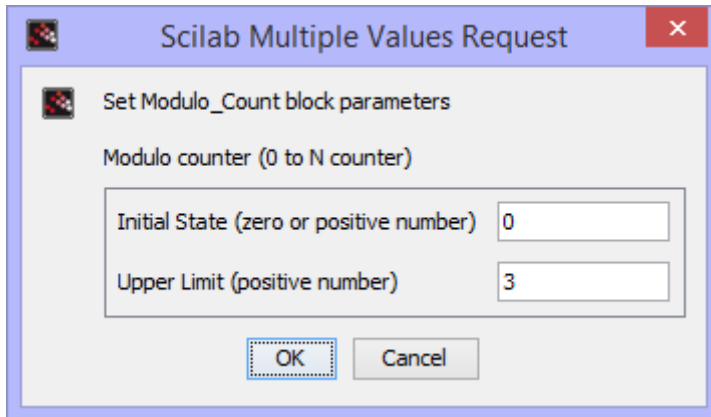
Blok predstavujúci implicitný vstupná port, ktorý musí byť použitý vo vnútri „Xcos Super Block“ a spojený s implicitným vstupom v systéme. Predstavuje odkaz na signál z vonku systému do subsystému. Implicitné vstupné porty musia byť očíslované od čísla 1, to sa nastavuje parametrom „Port Number“.



1.15.9 Modulo_Count



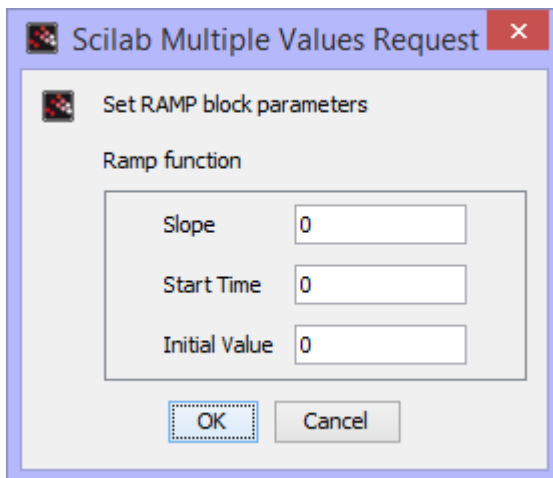
Je to diskretný blok. Výstupom z bloku je periodický skalárny čiastočne konštantný signál. Signál začína na hodnote „Initial State“. Výstupná hodnota je zvyšovaná o 1, keď na aktívny vstup príde úloha, až pokiaľ nedosiahne hodnotu v premennej „Upper Limit“. Po dosiahnutí tejto úrovne je výstupná hodnota resetovaná na 0.



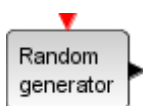
1.15.10 RAMP



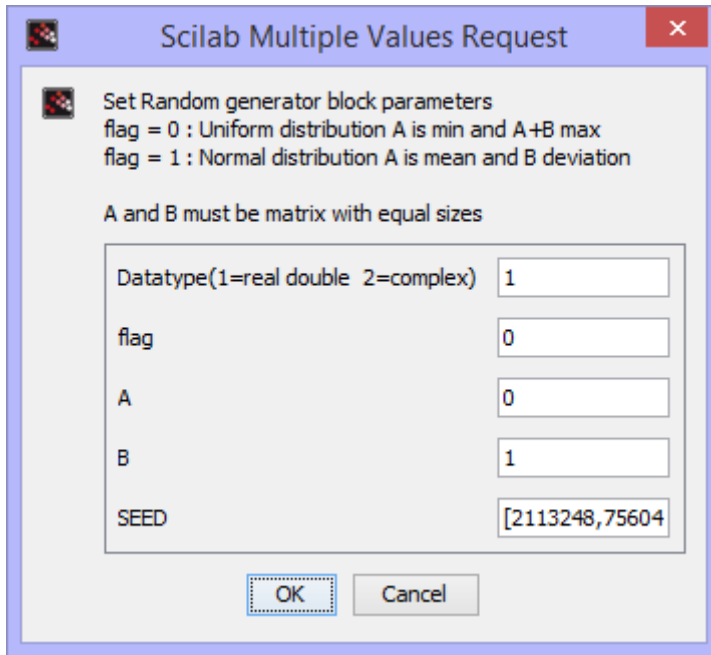
Blok generujúci signál, ktorý začína v učený čas definovaný premennou „Start Time“ a „Initial Value“ a zvyšuje sa o stanovenú mieru (parameter Slope).



1.15.11 RAND_m

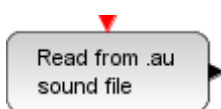


Tento blok generuje náhodný signál. Zakaždým keď na aktivačný vstup príde úloha, výstupom sa priradia nové nezávislé hodnoty. Veľkosť portu je daná veľkosťou matice A a B.

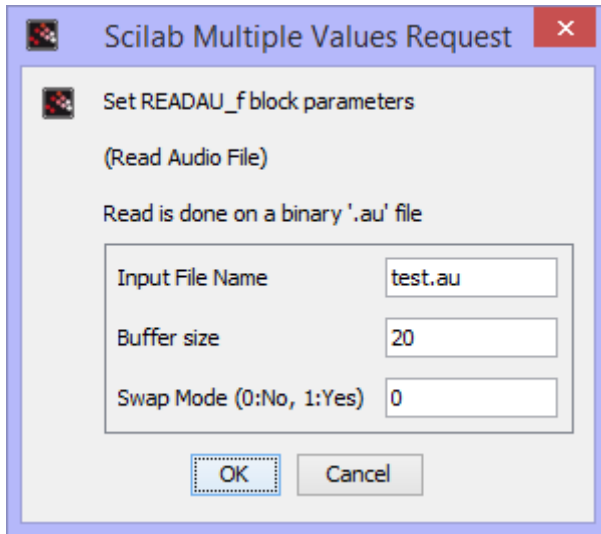


- **Datatype:** tento blok podporuje iba dátový typ double(1) alebo complex(2).
- **flag:** možno zvolit' 0 alebo 1. 0 pre jednotné rozloženie [A,A+B], 1 pre normálne rozloženie.
- **SEED:** hodnota pre sekvenciu náhodného čísla. Prvé číslo je reálna časť a druhé je imaginárna časť.

1.15.12 READAU_f

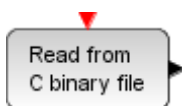


Blok načíta zvukový súbor s príponou *.au. Hodnoty sú v rozsahu -1 až +1.

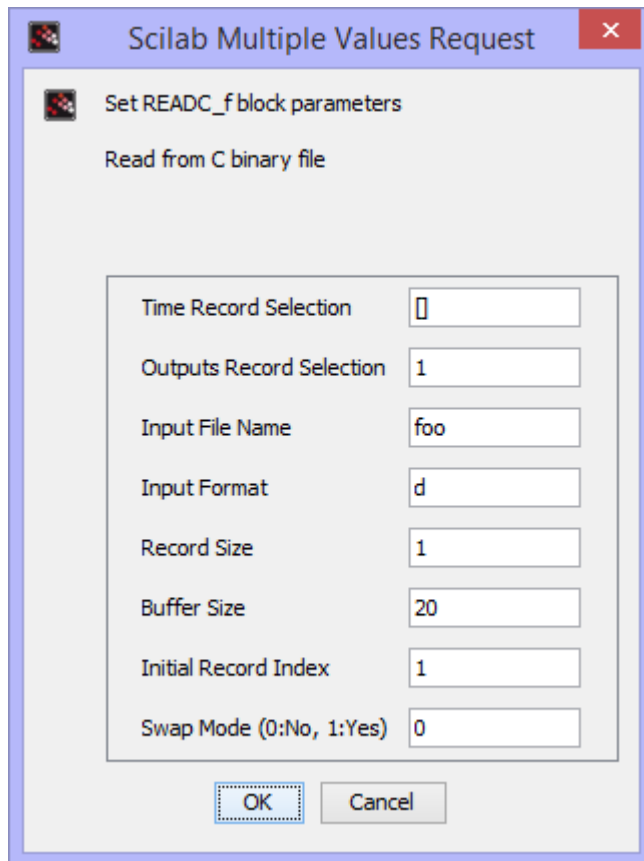


- **Input file name:** textový reťazec definujúci cestu k súboru.
- **Buffer size:** pre zvýšenie účinnosti je možné vstupné dáta dočasne uložiť do vyrovnávacej pamäte.
- **Swap Mode:** v móde 1 je súbor kódovaný v „little endian IEEE“ formáte a v prípade potreby je formát zmenený aby sa zhodoval s formátom IEEE. Ak je nastavený mód 0 automatická zmena dát je vypnutá.

1.15.13 READC_f

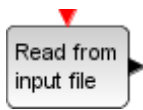


Tento blok umožňuje užívateľovi čítať dáta z C súboru. "Output record selection" a "Time-record Selection" umožňujú užívateľovi vyberať dáta zo záznamu súboru. Každé volanie k záznamu bloku zodpovedá jednému záznamu v súbore.

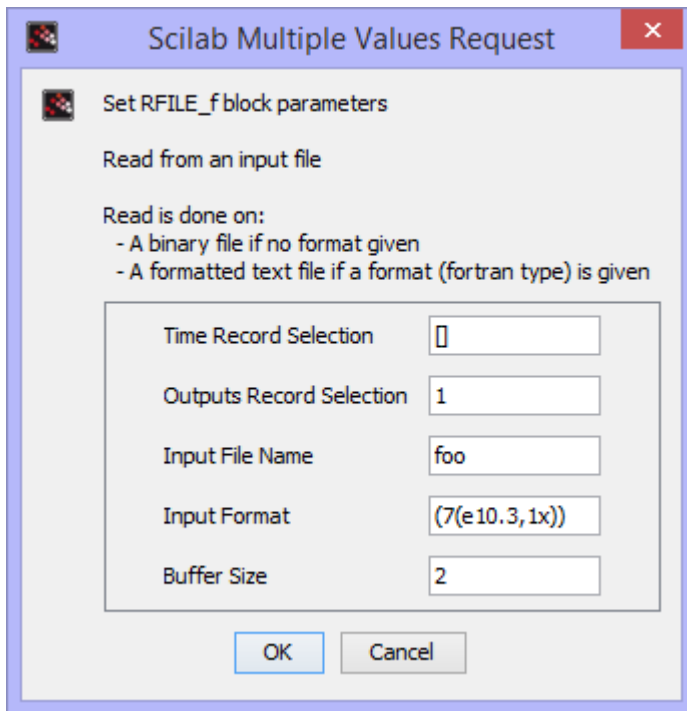


- **Time record selection:** prázdna matica alebo kladné celé číslo. Ak je číslo i brané ako i -tá časť čítaného záznamu, tak sa jedná o čas výstupnej udalosti.
- **Output Record Selection:** vektor celých kladných čísel. $[k_1, \dots, k_n]$, k_i -tá časť čítaného záznamu vracia hodnotu i -teho výstupu.
- **Input File Name:** charakteristický reťazec definujúci cestu k súboru.
- **Input Format:** charakteristický reťazec definujúci použitý formát (d – double, f – float, l – long, i – integer, s – short, c – character,...).
- **Record Size:** súbor je tvorený sekvenciami dát s rovnakým formátom. Tieto dáta sú usporiadané do sekvencií záznamov, kde každý záznam nesie informáciu veľkosti definovanú v „Record Size“.
- **Buffer Size:** veľkosť vyrovnávacej pamäte, ktorá sa pridáva kvôli zvýšeniu efektivity.
- **Initial record index:** vektor, ktorý opravuje prvý použitý záznam súboru.
- **Swap Mode:** v móde 1 je súbor kódovaný v „little endian IEEE“ formáte a v prípade potreby je formát zmenený aby sa zhodoval s formátom IEEE. Ak je nastavený mód 0 automatická zmena dát je vypnutá.

1.15.14 RFILE_f



Blok umožňujúci užívateľovi čítať dáta zo súboru, ktorý je definovaný parametrom „Input File Name“, v textovom alebo binárnom formáte. Súbor je zaznamenávaný v sekvenciách. Každý záznam má dĺžku definovanú ako „Record Size“



- **Time record selection:** prázdna matica alebo kladné celé číslo. Ak číslo i je brané ako i -ta časť čítaného záznamu, tak potom sa jedná o čas výstupnej udalosti.
- **Output Record Selection:** vektor celých kladných čísel. $[k_1, \dots, k_n]$, k_i -tá časť čítaného záznamu vracia hodnotu i -teho výstupu.
- **Input File Name:** charakteristický reťazec definujúci cestu k súboru.
- **Input Format:** charakteristický reťazec využívajúci formát „Fortran“ alebo žiadny pre neformátovaný (binárny) zápis. Ak je pridelený formát, musí začať ľavou zátvorkou a skončiť pravou zátvorkou. *Príklad:* $(10e3)$.
- **Buffer size:** veľkosť vyrovnávacej pamäte, ktorá sa prideluje kvôli zvýšeniu efektivity.

1.15.15 SAWTOOTH_f



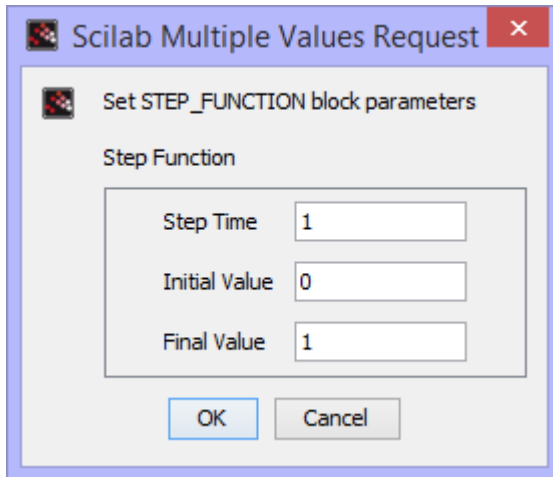
Blok generujúci trojuholníkový signál. Prvá udalosť na aktivačnom vstupe, lineárne zvyšuje výstupnú hodnotu z 0 pričítavaním 1. Pri ďalšej udalosti sa výstup zresetuje na 0 a začne sa opäť zvyšovať a potom stále dookola. Výsledná amplitúda výstupu je rovná dobe signálu udalosti.

1.15.16 STEP_FUNCTION



Blok prevádza funkciu zmeny počiatkovej hodnoty na finálnu hodnotu v danom čase. Hodnoty je možné definovať ako vektor alebo maticu. V tomto prípade sa blok riadi nasledovnou tabuľkou:

Veľkosť počiatkovej hodnoty	Veľkosť finálnej hodnoty	Výstupná veľkosť	Vzťah
1×1	1×1	1×1	
1×1	$1 \times N$	$N \times 1$	
$1 \times N$	$L \times M$	$N \times 1$	$N = L \times M$
$L \times M$	$1 \times N$	$N \times 1$	$N = L \times M$
$1 \times N$	$L \times M$	$N \times 1$	$N = L \times M$

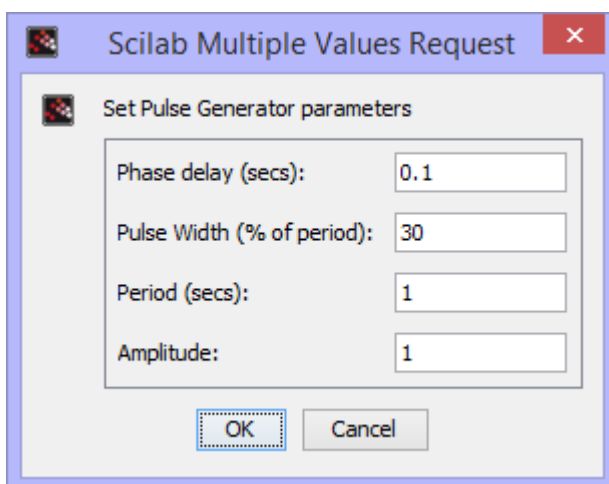


- **Step Time:** čas v sekundách, definujúci kedy má výstupný signál zmeniť počiatočnú hodnotu na finálnu hodnotu.
- **Initial Value:** počiatočná hodnota, ktorá je na výstupe kým neuplynie čas do skoku.
- **Final Value:** finálna hodnota, ktorá bude na výstupe keď uplynie čas skoku.

1.15.17 PULSE_SC



Blok ako generátor impulzu. Generuje impulz v pomere $1/P$, kde P je perióda signálu.

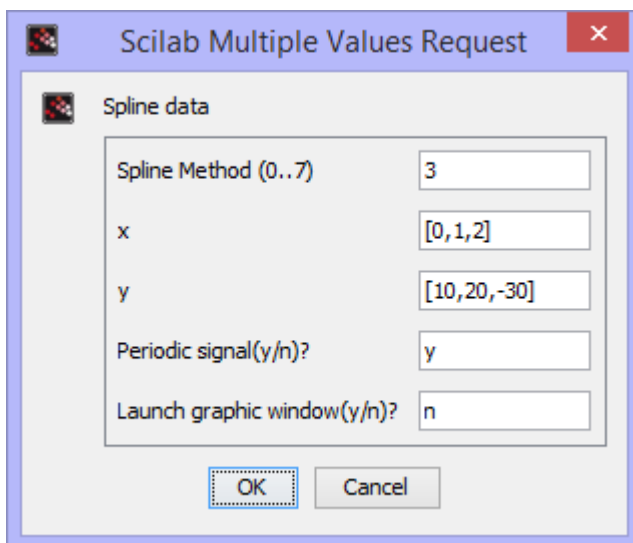


- **Phase delay (secs):** fázový posun. Musí byť menší ako $Frequency * (1 - (Pulse_width/100))$.
- **Pulse Width (% of period):** veľkosť pulzu v percentách.
- **Period (secs):** perióda signálu.
- **Amplitude:** veľkosť amplitúdy. Podporuje všetky Xcos dátové typy.

1.15.18 Sigbuilder



Daný blok je zložený ako Super blok. Obsahuje blok, ktorého výstupný riadiaci port je pripojený na vstupný riadiaci port. Táto spätná väzba poskytuje možnosť generovať body nespojitého signálu. Vytvorené riadiace udalosti automaticky reštartujú numerického riešiteľa a vyhýba sa numerickým problémom. Generovaná udalosť je tiež k dispozícii pre užívateľa.

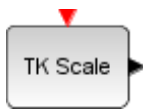


1.15.19 TIME_f

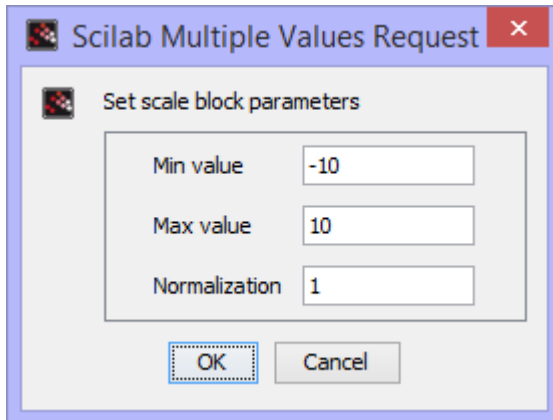


Blok časového generátoru. Špeciálny výstupný port dáva aktuálny čas.

1.15.20 TKSCALE



Tento blok generuje čiastočne konštantný signál, ktorého hodnota môže byť nastavená interaktívne pomocou „widgetu“ v priebehu simulácie.



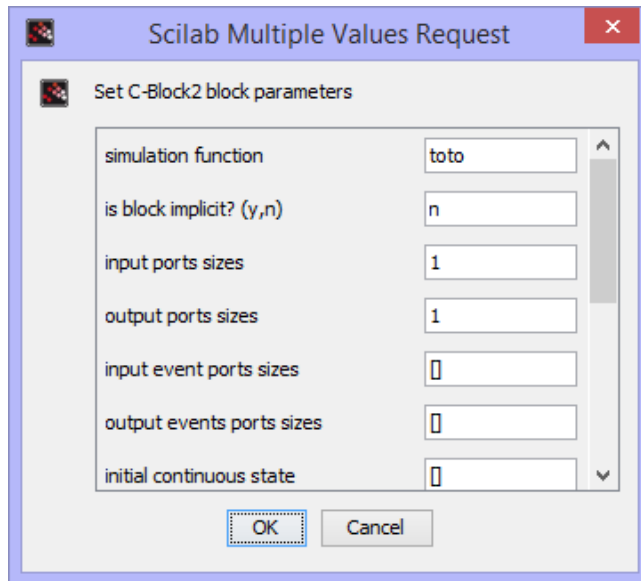
- **Min value:** celé číslo špecifikujúce minimálnu hodnotu rozsahu.
- **Max value:** celé číslo špecifikujúce maximálnu hodnotu rozsahu.
- **Normalizatiaon:** Výstup bloku je celočíselná hodnota daná zadaná posuvníkom a vydelená týmto parametrom.

1.16 User-Defined Function

1.16.1 CBLOCK



Blok tvoriaci základnú kostru pre funkcie v jazyku C. Tiež vytvorí súbory knižníc a objektové súbory.

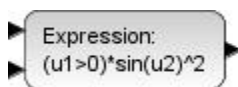


- **Simulation function:** názov generovanej funkcie.
- **Is block implicit?:** ak je zvolené áno (y), je zavolaný implicitný riešiteľ. Ak je zvolené nie (n), je zavolaný explicitný riešiteľ.
- **Input ports sizes:** počet regulárnych vstupných portov.
- **Output ports sizes:** počet regulárnych výstupných portov.
- **Input event ports sizes:** počet riadiacich vstupných portov.
- **Output events ports sizes:** počet riadiacich výstupných portov.
- **Initial continuous state:** počiatočná podmienka spojitého stavu.
- **Number of zero crossing surfaces:** povolenie detekcie prechodu nulou.
- **Initial discrete state:** počiatočná podmienka diskrétného stavu.
- **Real parameters vector:** vektor reálnych čísel, ktorý prijíma funkciu.
- **Integer parameters vector:** celočíselný vektor, ktorý prijíma funkciu.
- **Initial firing vector:** vektor, ktorého veľkosť je rovnaká ako počet riadiacich výstupov. Hodnota i-tého vstupu, špecifikuje čas predprogramovanej udalosti smeru-

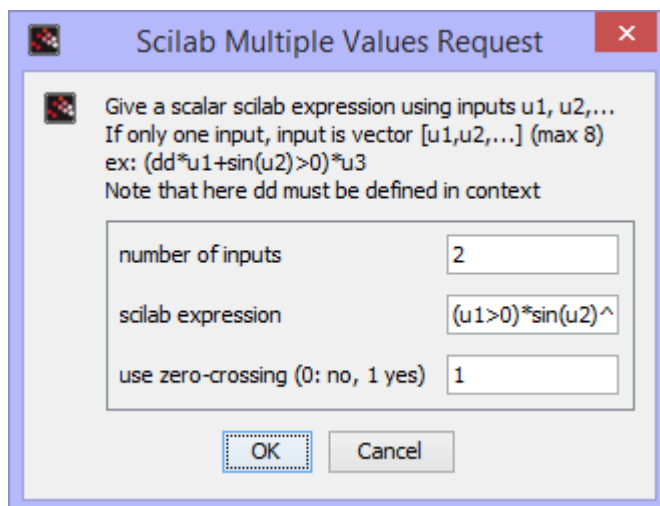
júcej na i-tý výstupný riadiaci port. Ak je menší ako nula, udalosť nie je predprogramovaná.

- **Direct feedthrough:** vstup bloku v aktuálnom čase určuje výstup bloku v aktuálnom čase. To prinúti vstup premeniť do výstupu, ako keby bol v prevádzke neustále.
- **Time dependence:** vytvorí signál, ktorý špecifikuje časovú závislosť.

1.16.2 EXPRESSION



Daný blok využije zadané Scilab výrazy na jeho výstupe.

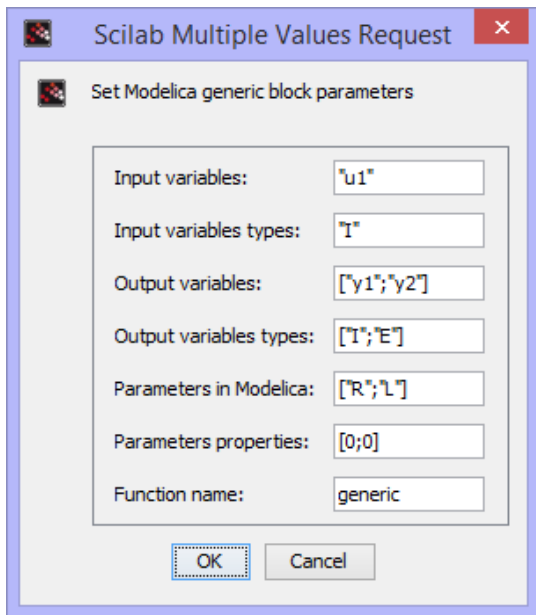


- **Number of inputs:** vektor alebo skalár udávajúci počet vstupov.
- **Scilab expression:** Scilab výraz aplikovaný na vstupe.
- **Use zero-crossing:** povolenie detekcie prechodu nulou.

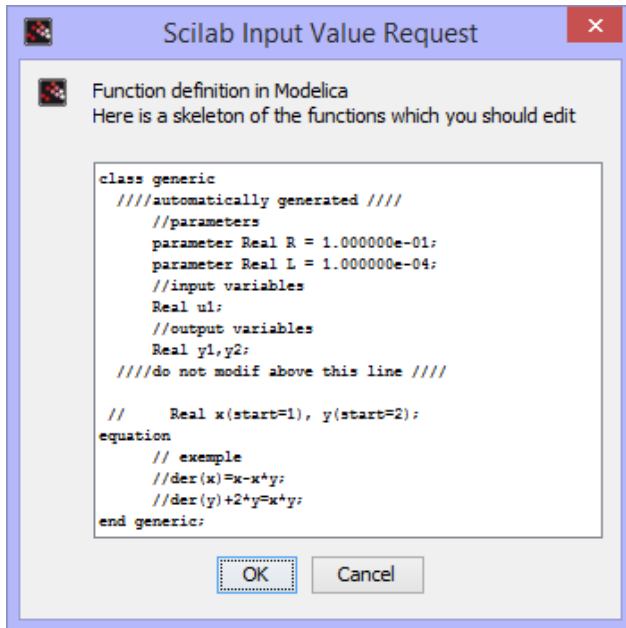
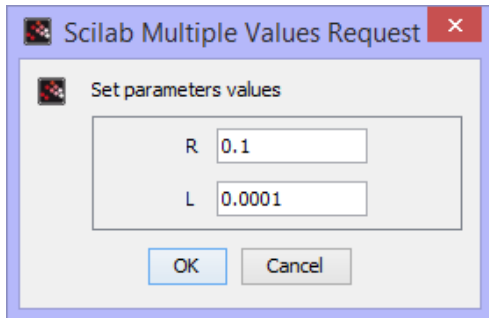
1.16.3 MBLOCK



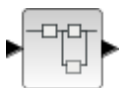
Blok, ktorého správanie určuje program Modelica. Pomocou neho môže užívateľ písať a kompilovať v Modelice. Daný blok môže byť kombinovaný s ostatnými blokmi.



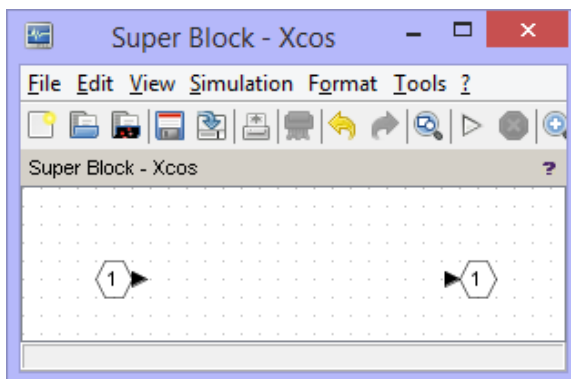
- **Input variables:** vektor, definujúci vstupné porty bloku. Premenná v Modelice by mala byť definovaná ako dátový typ „Real“.
- **Input variables types:** špecifikácia vstupných premenných, kde „I“ sú implicitné porty a „E“ sú explicitné porty. Veľkosť „Input variables“ a tohto parametru by mali byť rovnaké.
- **Output variables:** platí to čo pri „Input variables“ akurát sa jedná o výstup.
- **Output variables types:** platí to čo pri „Input variables types“ akurát sa jedná o výstup.
- **Parameter in Modelica:** definovanie parametrov v Modelice. Môžu by aj preťažované.
- **Parameter properties:** typ Modelica parametrov. Definované môžu byť tri typy parametrov:
 - 0: parameter je nastavený ako Modelica parameter. Môže byť skalár alebo vektor,
 - 1: parameter je nastavený ako počiatočná podmienka Modelica stavu. Môže byť skalár alebo vektor.
 - 2: parameter je nastavený ako počiatočná podmienka Modelica stavu s vlastnosťou „fixed=true“. Môže byť skalár alebo vektor.
- **Function name:** udáva názov triedy bez cesty, ktorú má blok interaktívne otvoriť ako nové okno. Ak blok nenájde zadaný súbor s príponou *.mo, tak bude vytvorený nový súbor s príponou *.mo. Keď už je súbor vytvorený, je možné ho upravovať v textovom editore.



1.16.4 SUPER_f



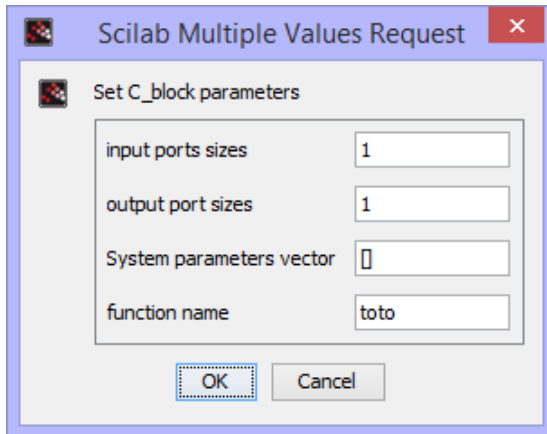
Tento blok otvorí nové dialógové okno na editáciu novej blokovej schémy. Táto nová schéma pracuje ako subsystém. Môže mať vstupné a výstupné porty a môžu byť buď regulárne alebo riadiace.



1.16.5 c_block



Blok tvoriaci základnú kostru pre funkcie v jazyku C. Tiež vytvorí súbory knižníc a objektové súbory.

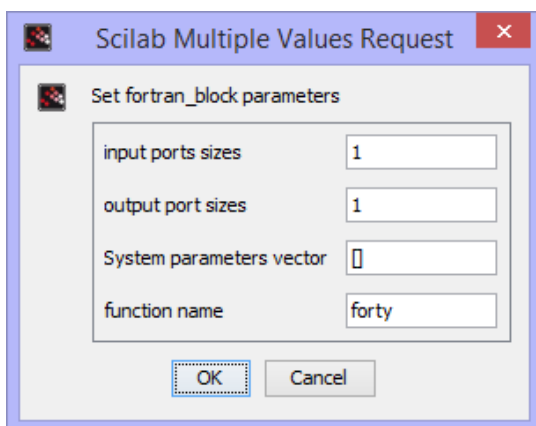


- **Input ports sizes:** počet regulárnych vstupných portov.
- **Output port sizes:** počet regulárnych výstupných portov.
- **System parameters vector:** počet parametrov, ktoré akceptujú danú funkciu.
- **Function name:** názov generovanej funkcie.

1.16.6 fortran_block



Blok tvoriaci základnú kostru pre funkcie v jazyku FORTRAN. Tiež vytvorí súbory knižníc a objektové súbory.

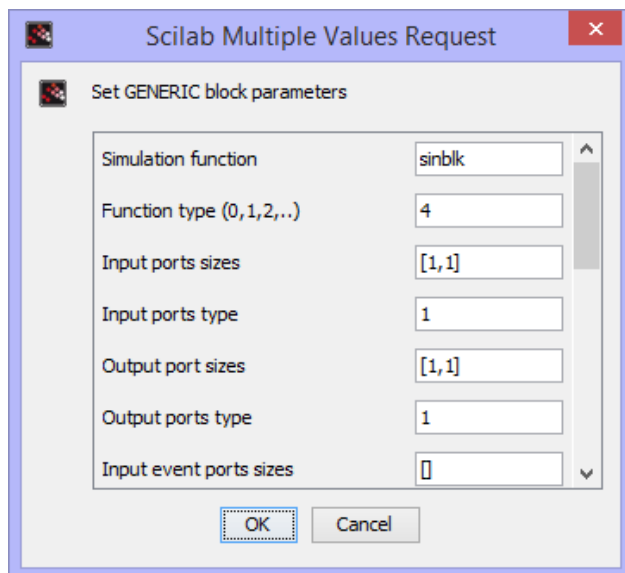


- **Input ports sizes:** počet regulárnych vstupných portov.
- **Output port sizes:** počet regulárnych výstupných portov.
- **System parameters vector:** počet parametrov, ktoré akceptujú danú funkciu.
- **Function name:** názov generovanej funkcie.

1.16.7 generic_block3



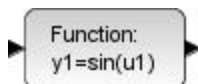
Blok poskytuje všeobecné funkčné rozhranie, ale je potrebné aby funkcie jednotlivých jazykov boli od seba oddelené.



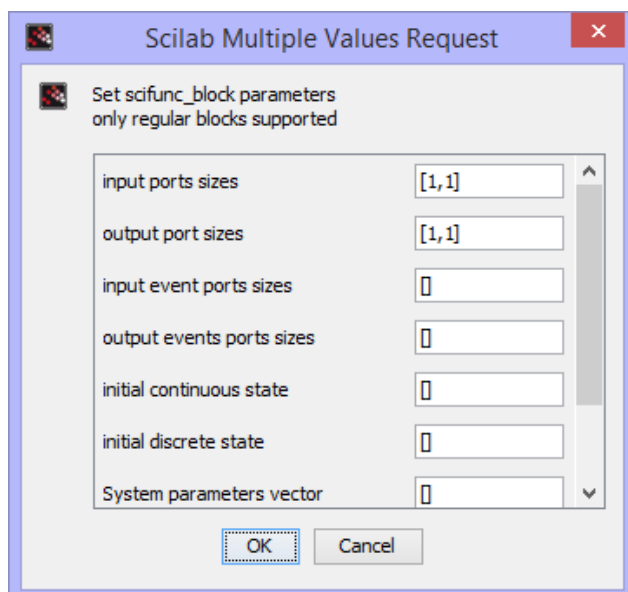
- **Simulation function:** názov načítanej funkcie.
- **Function type:** typ výpočtovej funkcie podporovanej nadstavbou Xcos.
- **Input ports sizes:** počet regulárnych vstupných portov.
- **Input ports type:** nastaví dátový typ regulárnych vstupných portov.
- **Output ports sizes:** počet regulárnych výstupných portov.
- **Output ports type:** nastaví dátový typ regulárnych výstupných portov.
- **Input event ports sizes:** vektor veľkostí vstupných riadiacich portov. Veľkosť vektora udáva počet riadiacich vstupov.
- **Output events ports sizes:** vektor veľkostí výstupných riadiacich portov. Veľkosť vektora udáva počet riadiacich výstupov.
- **Initial continuous state:** stĺpcový vektor počiatočných spojitéch stavov.
- **Initial discrete state:** stĺpcový vektor počiatočných diskretných stavov.

- **Real parameters vector:** riadkový vektor. Parametre reálneho typu použité v tomto bloku môžu byť definované tu ako riadkový vektor.
- **Integer parameters vector:** riadkový vektor. Parametre celočíselného typu použité v tomto bloku môžu byť definované tu ako riadkový vektor.
- **Object parameters list:** Scilab zoznam, ktorý definuje zoznam objektových parametrov.
- **Number of modes:** počet pravostranných funkcií v systéme.
- **Number of zero_crossing:** počet prechodov nulou.
- **Initial firing vector:** vektor, ktorý zodpovedá počtu riadiacich výstupov. Hodnota i-teho vstupu určuje čas predprogramovanej udalosti idúcej na i-ty riadiaci výstup. Ak je menšia ako nula, žiadna udalosť nie je naprogramovaná.
- **Direct feedthrough:** znak „y“ alebo „n“ určuje, či má blok priamy vplyv na výstupnej prechodke.
- **Time dependence:** časová závislosť. Znak „y“ alebo „n“ určuje, či je výstup explicitne závislý na čase.

1.16.8 scifunc_block_m



Tento blok môže realizovať akýkoľvek typ Xcos bloku. Funkcia bloku je definovaná interaktívne pomocou dialógového boxu a Scilab jazyka. Počas simulácie sú tieto inštrukcie interpretované Scilabom. Simulácie diagramov, ktoré zahŕňa tento blok, sú pomalšie.



- **Input ports sizes:** skalár udávajúci počet regulárnych vstupných portov.
- **Output port sizes:** skalár udávajúci počet regulárnych výstupných portov.
- **Input event ports sizes:** skalár udávajúci počet riadiacich vstupných portov.
- **Output events ports sizes:** skalár udávajúci počet riadiacich výstupných portov.
- **Initial continuous state:** stĺpcový vektor.
- **Initial discrete state:** stĺpcový vektor.
- **System parameters vector:** reťazec: c alebo d (CBB alebo DBB), ostatné typy nie sú podporované.
- **Initial firing vector:** vektor, ktorého veľkosť sa zhoduje s počtom riadiacich výstupov. Hodnota i-teho vstupu určuje čas predprogramovanej udalosti idúcej na i-ty riadiaci výstup. Ak je menšia ako nula, žiadna udalosť nie je naprogramovaná.
- **Is block always:** ďalšie dialógy sú otvorené nepretržite, kde môže byť použitý vstupný Scilab kód spojený s potrebnými výpočtami (inicializácia bloku, výstupy, spojité a diskkrétne stavy, čas výstupných udalostí, koniec bloku).