Agilent Technologies

Advanced Design System 2011.01

Feburary 2011
MATLAB Blocks

Acknowledgments
Mentor Graphics is a trademark of Mentor Graphics Corporation in the U.S. and other countries. Mentor products and processes are registered trademarks of Mentor Graphics Corporation. * Calibre is a trademark of Mentor Graphics Corporation in the US and other countries. "Microsoft®, Windows®, MS Windows®, Windows NT®, Windows 2000® and Windows Internet Explorer® are U.S. registered trademarks of Microsoft Corporation. Pentium® is a U.S. registered trademark of Intel Corporation. PostScript® and Acrobat® are trademarks of Adobe Systems Incorporated. UNIX® is a registered trademark of the Open Group. Oracle and Java and registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners. SystemC® is a registered trademark of Open SystemC Initiative, Inc. in the United States and other countries and is used with permission. MATLAB® is a U.S. registered trademark of The Math Works, Inc.. HiSIM2 source code, and all copyrights, trade secrets or other intellectual property rights in and to the source code in its entirety, is owned by Hiroshima University and STARC. FLEXlm is a trademark of Globetrotter Software, Incorporated. Layout Boolean Engine by Klaas Holwerda, v1.7 http://www.xs4all.nl/~kholwerd/bool.html . FreeType Project, Copyright (c) 1996-1999 by David Turner, Robert Wilhelm, and Werner Lemberg. QuestAgent search engine (c) 2000-2002, JObjects. Motif is a trademark of the Open Software Foundation. Netscape is a trademark of Netscape Communications Corporation. Netscape Portable Runtime (NSPR), Copyright (c) 1998-2003 The Mozilla Organization. A copy of the Mozilla Public License is at http://www.mozilla.org/MPL/ . FFTW, The Fastest Fourier Transform in the West, Copyright (c) 1997-1999 Massachusetts Institute of Technology. All rights reserved.

The following third-party libraries are used by the NlogN Momentum solver:

"This program includes Metis 4.0, Copyright © 1998, Regents of the University of Minnesota", http://www.cs.umn.edu/~metis , METIS was written by George Karypis (karypis@cs.umn.edu).

Intel@ Math Kernel Library, http://www.intel.com/software/products/mkl

SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

7-zip - 7-Zip Copyright: Copyright (C) 1999-2009 Igor Pavlov. Licenses for files are: 7z.dll: GNU LGPL + unRAR restriction, All other files: GNU LGPL. 7-zip License: This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version. This library is distributed in the hope that it will be useful,but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA. unRAR copyright: The decompression engine for RAR archives was developed using source code of unRAR program.All copyrights to original unRAR code are owned by Alexander Roshal. unRAR License: The unRAR sources cannot be used to re-create the RAR compression algorithm, which is proprietary. Distribution of modified unRAR sources in separate form or as a part of other software is permitted, provided that it is clearly stated in the documentation and source comments that the code may not be used to develop a RAR (WinRAR) compatible archiver. 7-zip Availability: http://www.7-zip.org/

AMD Version 2.2 - AMD Notice: The AMD code was modified. Used by permission. AMD copyright: AMD Version 2.2, Copyright © 2007 by Timothy A. Davis, Patrick R. Amestoy, and Iain S. Duff. All Rights Reserved. AMD License: Your use or distribution of AMD or any modified version of AMD implies that you agree to this License. This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version. This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA Permission is hereby granted to use or copy this program under the terms of the GNU LGPL, provided that the Copyright, this License, and the Availability of the original version is retained on all copies.User documentation of any code that uses this code or any modified version of this code must cite the Copyright, this License, the Availability note, and "Used by permission." Permission to modify the code and to distribute modified code is granted, provided the Copyright, this License, and the Availability note are retained, and a notice that the code was modified is included. AMD Availability: http://www.cise.ufl.edu/research/sparse/amd

UMFPACK 5.0.2 - UMFPACK Notice: The UMFPACK code was modified. Used by permission. UMFPACK Copyright: UMFPACK Copyright © 1995-2006 by Timothy A. Davis. All Rights Reserved. UMFPACK License: Your use or distribution of UMFPACK or any modified version of UMFPACK implies that you agree to this License. This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version. This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA Permission is hereby granted to use or copy this program under the terms of the GNU LGPL, provided that the Copyright, this License, and the Availability of the original version is retained on all copies. User documentation of any code that uses this code or any modified version of this code must cite the Copyright, this License, the Availability note, and "Used by permission." Permission to modify the code and to distribute modified code is granted, provided the Copyright, this License, and the Availability note are retained, and a notice that the code was modified is included. UMFPACK Availability: http://www.cise.ufl.edu/research/sparse/umfpack UMFPACK (including versions 2.2.1 and earlier, in FORTRAN) is available at http://www.cise.ufl.edu/research/sparse . MA38 is available in the Harwell Subroutine Library. This version of UMFPACK includes a modified form of COLAMD Version 2.0, originally released on Jan. 31, 2000, also available at http://www.cise.ufl.edu/research/sparse . COLAMD V2.0 is also incorporated as a built-in function in MATLAB version 6.1, by The MathWorks, Inc. http://www.mathworks.com . COLAMD V1.0 appears as a column-preordering in SuperLU (SuperLU is available at http://www.netlib.org). UMFPACK v4.0 is a built-in routine in MATLAB 6.5. UMFPACK v4.3 is a built-in routine in MATLAB 7.1.

Qt Version 4.6.3 - Qt Notice: The Qt code was modified. Used by permission. Qt copyright: Qt Version 4.6.3, Copyright (c) 2010 by Nokia Corporation. All Rights Reserved. Qt License: Your use or distribution of Qt or any modified version of Qt implies that you agree to this License. This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version. This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA Permission is hereby granted to use or copy this program under the terms of the GNU LGPL, provided that the Copyright, this License, and the Availability of the original version is retained on all copies.User documentation of any code that uses this code or any modified version of this code must cite the Copyright, this License, the Availability note, and "Used by permission." Permission to modify the code and to distribute modified code is granted, provided the Copyright, this License, and the Availability note are retained, and a notice that the code was modified is included. Qt Availability: http://www.qtsoftware.com/downloads Patches Applied to Qt can be found in the installation at: $HPEESOF_DIR/prod/licenses/thirdparty/qt/patches. You may also contact Brian Buchanan at Agilent Inc. at brian_buchanan@agilent.com for more information.

The HiSIM_HV source code, and all copyrights, trade secrets or other intellectual property rights in and to the source code, is owned by Hiroshima University and/or STARC.

Errata The ADS product may contain references to "HP" or "HPEESOF" such as in file names and directory names. The business entity formerly known as "HP EEsof" is now part of Agilent Technologies and is known as "Agilent EEsof". To avoid broken functionality and to maintain backward compatibility for our customers, we did not change all the names and labels that contain "HP" or "HPEESOF" references.

Warranty The material contained in this document is provided "as is", and is subject to being changed, without notice, in future editions. Further, to the maximum extent permitted by applicable law, Agilent disclaims all warranties, either express or implied, with regard to this documentation and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Agilent shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein. Should Agilent and the user have a separate written agreement with warranty terms covering the material in this document that conflict with these terms, the warranty terms in the separate agreement shall control.

Technology Licenses The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license. Portions of this product include the SystemC software licensed under Open Source terms, which are available for download at http://systemc.org/ . This software is redistributed by Agilent. The Contributors of the SystemC software provide this software "as is" and offer no warranty of any kind, express or implied, including without limitation warranties or conditions or title and non-infringement, and implied warranties or conditions merchantability and fitness for a particular purpose. Contributors shall not be liable for any damages of any kind including without limitation direct, indirect, special, incidental and consequential damages, such as lost profits. Any provisions that differ from this disclaimer are offered by Agilent only.

Restricted Rights Legend U.S. Government Restricted Rights. Software and technical data rights granted to the federal government include only those rights customarily provided to end user customers. Agilent provides this customary commercial license in Software and technical data pursuant to FAR 12.211 (Technical Data) and 12.212 (Computer Software) and, for the Department of Defense, DFARS 252.227-7015 (Technical Data - Commercial Items) and DFARS 227.7202-3 (Rights in Commercial Computer Software or Computer Software Documentation).

About Numeric Matlab

The MATLAB models provide an interface between ADS Ptolemy and MATLAB, a numeric computation and visualization environment from The MathWorks, Inc. All ADS Ptolemy MATLAB Cosimulation models can be found in the *Numeric Matlab* component palette and library on the Digital Signal Processing (DSP) Schematic.

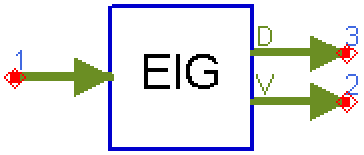
- **Models that interpret a MATLAB script** - The script can contain a function, command, statement or several statements.
    - *MatlabCx_M* (matlabblocks)
    - *MatlabFCx_M* (matlabblocks)
    - *MatlabF_M* (matlabblocks)
    - *MatlabSink* (matlabblocks)
    - *MatlabSinkF* (matlabblocks)
    - *Matlab_M* (matlabblocks)

- **Models that call specific MATLAB built-in functions**
    - *EigCx_M* (matlabblocks)
    - *Norm_M* (matlabblocks)
    - *RandDeintrlv* (matlabblocks)
    - *RandIntrlv* (matlabblocks)
    - *SVD_Cx_M* (matlabblocks)
    - *Erf* (matlabblocks)
    - *Erfc* (matlabblocks)

The components shown in this last section above represent examples of built-in MATLAB functions that are imported for Advanced Design System (ADS) DSP simulation. These imported models demonstrate different methods for interfacing MATLAB blocks with other ADS models. Each model has relevance in communications system analysis and design.

> **Note**
> Use of these models requires that you have MATLAB properly installed and configured for ADS co-simulation. For more information on MATLAB and details on the MATLAB implementation of the various computations used in this document, visit The MathWorks web site at http://www.mathworks.com .

EigCx_M



Description Eigenvalue Decomposition of General Complex Matrix
Library Numeric Matlab

Pin Inputs

Pin	Name	Description	Signal Type
1	X	Input matrix	complex matrix

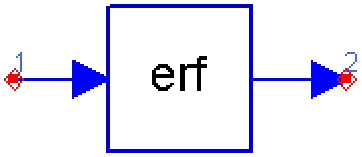
Pin Outputs

Pin	Name	Description	Signal Type
2	V	Eigenvectors	complex matrix
3	D	Eigenvalues	complex matrix

Notes/Equations

1. This model performs the eigenvalue decomposition of a complex matrix using the MATLAB eig function, `[V, D]= eig(A)`. For details on the MATLAB implementation of this computation, visit The MathWorks website at http://www.mathworks.com .
2. The eigenvalue decomposition takes a complex matrix on the input (pin 1) and outputs a complex matrix whose columns represent the eigenvectors of the input (pin 2) and a diagonal matrix whose complex entries represent the eigenvalues of the input (pin 3).
3. In communication system and signal processing analysis, the eigenvalue decomposition is useful for determining the important subspace of a matrix entity. For example, suppose A represents a signal covariance matrix. If we let D0 represent D with all entries whose magnitude is smaller than a threshold set to zero, then VD0V-1 represents an efficient approximation to the covariance. Typically for a covariance, V-1 = VH, where the superscript H represents a conjugate transpose. In this case, only a subset of the columns of V (the subspace) is used to approximate the covariance.

Erf



Description Error Function
Library Numeric Matlab

Pin Inputs

Pin	Name	Description	Signal Type
1	x	Input data	real

Pin Outputs

Pin	Name	Description	Signal Type
2	y	Output data	real

Notes/Equations

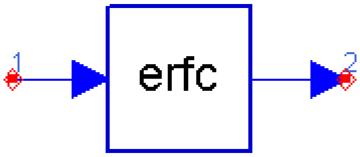
1. This model computes the error function of the input data using the MATLAB erf function, `[y] = erf(x)`. For details on the MATLAB implementation of this computation, visit The MathWorks website at http://www.mathworks.com .

2. The error function is mathematically represented as:

$$y = erf(x) = \frac{2}{\sqrt{\pi}}\int_0^x \exp(-t^2)dt$$

3. The error function is commonly used on communication system analysis to determine the probability of correctly detecting a symbol when the additive noise is described as a Gaussian random process.

Erfc



Description Complementary Error Function
Library Numeric Matlab

Pin Inputs

Pin	Name	Description	Signal Type
1	x	Input data	real

Pin Outputs

Pin	Name	Description	Signal Type
2	y	Output data	real

Notes/Equations

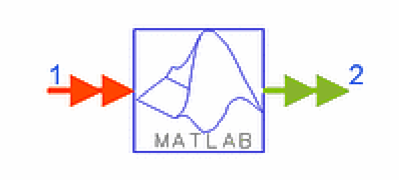
1. This model computes the complementary error function of the input data using the MATLAB erfc function, `[y] = erfc(x)`. For details on the MATLAB implementation of this computation, visit The MathWorks website at http://www.mathworks.com .
2. The complementary error function is mathematically represented as:

$$y = erfc(x) = \frac{2}{\sqrt{\pi}}\int_x^\infty \exp(-t^2)dt$$

3. The complementary error function is commonly used on communication system analysis to determine the probability of correctly detecting a symbol when the additive noise is described as a Gaussian random process. In fact, the Q function commonly used in bit error rate analysis is related to the complementary error function using:

$$Q(x) = \frac{1}{2}erfc\left(\frac{x}{\sqrt{2}}\right)$$

MatlabCx_M



Description Matlab Complex Matrix Output
Library Numeric, Matlab
Class SDFMatlabCx_M
Derived From Matlab_M

Parameters

Name	Description	Default	Type
ScriptDirectory	An optional directory specifying where to find any custom Matlab files.		string
MatlabSetUp	Matlab command to execute during begin method		string
MatlabFunction	Matlab command to execute for each simulation sample		string
MatlabWrapUp	Matlab command to execute during wrapup method		string

Pin Inputs

Pin	Name	Description	Signal Type
1	input		multiple anytype

Pin Outputs

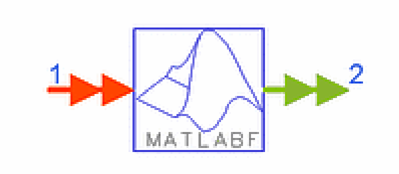
Pin	Name	Description	Signal Type
2	output		multiple complex matrix

Notes/Equations

ℹ MatlabLibLinkCx component has been made obsolete and migrated to MatalbCx_M. However, the setup parameter formatting may need to be adjusted post-migration.

1. MatlabCx_M evaluates Matlab functions on its inputs and outputs complex matrices.
2. ScriptDirectory is an optional directory specifying where to find any custom Matlab files.
3. For more information about Matlab components, refer to *MATLAB Cosimulation Introduction* (ptolemy).
4. For more information regarding numeric matrix component signals, refer to *Numeric Matrix Components* (numeric).

MatlabFCx_M



Description Matlab Complex Matrix Output with Scripts Importing
Library Numeric, Matlab
Class SDFMatlabFCx_M

Derived From MatlabF_M

Parameters

Name	Description	Default	Type
ScriptDirectory	An optional directory specifying where to find any custom Matlab files.		string
MatlabSetUp	Matlab command to execute during begin method		filename
MatlabFunction	Matlab command to execute for each simulation sample		filename
MatlabWrapUp	Matlab command to execute during wrapup method		filename

Pin Inputs

Pin	Name	Description	Signal Type
1	input		multiple anytype

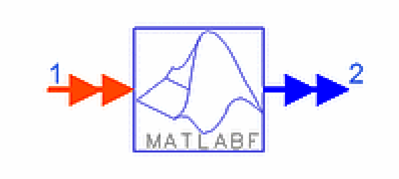
Pin Outputs

Pin	Name	Description	Signal Type
2	output		multiple complex matrix

Notes/Equations

1. MatlabFCx_M evaluates Matlab functions on its inputs and outputs complex matrices.
2. MatlabSetup, MatlabFunction, and MatlabWrapUp inputs accept script files only.
3. ScriptDirectory is an optional directory specifying where to find any custom Matlab files referenced inside MatlabSetup, MatlabFunction, and MatlabWrapUp scripts.
4. For more information about Matlab components, refer to *MATLAB Cosimulation Introduction* (ptolemy).
5. For more information regarding numeric matrix component signals, refer to *Numeric Matrix Components* (numeric).

MatlabF_M



Description Matlab Floating Point Matrix Output with Scripts Importing
Library Numeric, Matlab
Class SDFMatlabF_M
Derived From Matlab

Parameters

Name	Description	Default	Type
ScriptDirectory	An optional directory specifying where to find any custom Matlab files.		string
MatlabSetUp	Matlab command to execute during begin method		filename
MatlabFunction	Matlab command to execute for each simulation sample		filename
MatlabWrapUp	Matlab command to execute during wrapup method		filename

Pin Inputs

Pin	Name	Description	Signal Type
1	input		multiple anytype

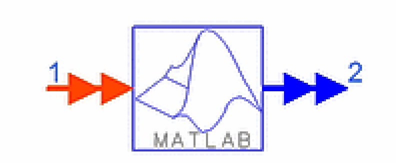
Pin Outputs

Pin	Name	Description	Signal Type
2	output		multiple real matrix

Notes/Equations

1. MatlabF_M evaluates Matlab functions on its inputs and outputs floating-point (real) matrices.
2. MatlabSetup, MatlabFunction, and MatlabWrapUp inputs accept script files only.
3. ScriptDirectory is an optional directory specifying where to find any custom Matlab files referenced inside MatlabSetup, MatlabFunction, and MatlabWrapUp scripts.
4. The MatlabF_M component has parameters called *MatlabSetUp* and *MatlabWrapUp* as shown in the parameter table above. These parameters can refer to a *.m file.
5. For more information about Matlab components, refer to *MATLAB Cosimulation Introduction* (ptolemy).
6. For more information regarding numeric matrix component signals, refer to *Numeric Matrix Components* (numeric).

Matlab_M



Description Matlab Floating Point Matrix Output
Library Numeric, Matlab
Class SDFMatlab_M
Derived From Matlab

Parameters

Name	Description	Default	Type
ScriptDirectory	An optional directory specifying where to find any custom Matlab files.		string
MatlabSetUp	Matlab command to execute during begin method		string
MatlabFunction	Matlab command to execute for each simulation sample		string
MatlabWrapUp	Matlab command to execute during wrapup method		string

Pin Inputs

Pin	Name	Description	Signal Type
1	input		multiple anytype

Pin Outputs

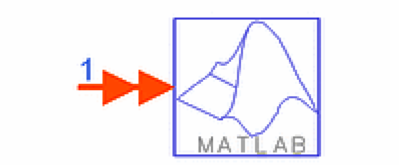
Pin	Name	Description	Signal Type
2	output		multiple real matrix

Notes/Equations

MatlabLibLink component has been made obsolete and migrated to Matalb_M. However, the setup parameter formatting may need to be adjusted post-migration.

1. Matlab_M evaluates Matlab functions on its inputs and outputs floating-point (real) matrices.
2. ScriptDirectory is an optional directory specifying where to find any custom Matlab files.
3. The Matlab_M component has parameters called *MatlabSetUp* and *MatlabWrapUp* as shown in the parameter table above. These parameters can refer to a *.m file.
4. For more information about Matlab components, refer to *MATLAB Cosimulation Introduction* (ptolemy).
5. For more information regarding numeric matrix component signals, refer to *Numeric Matrix Components* (numeric).

MatlabSink



Description Matlab Function
Library Numeric, Matlab
Class SDFMatlabSink
Derived From Matlab_M

Parameters

Name	Description	Default	Type	Range
ScriptDirectory	An optional directory specifying where to find any custom Matlab files.		string	
MatlabSetUp	Matlab command to execute during begin method		string	
MatlabFunction	Matlab command to execute for each simulation sample		string	
MatlabWrapUp	Matlab command to execute during wrapup method		string	
NumberOfFirings	number of invocations during a simulation	1	int	[1, ∞)

Pin Inputs

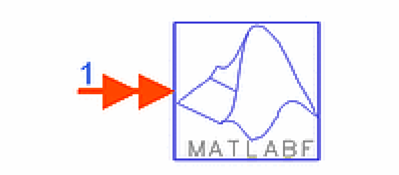
Pin	Name	Description	Signal Type
1	input		multiple anytype

Notes/Equations

1. MatlabSink behaves like other Matlab components except that it does not have any output. The total amount of data collected is determined by NumberOfFirings.

2. ScriptDirectory is an optional directory specifying where to find any custom Matlab files.
3. For more information about Matlab components, refer to *MATLAB Cosimulation Introduction* (ptolemy).
4. For more information regarding numeric matrix component signals, refer to *Numeric Matrix Components* (numeric).

MatlabSinkF



Description Matlab Function with Scripts Importing
Library Numeric, Matlab
Class SDFMatlabSinkF
Derived From MatlabF_M

Parameters

Name	Description	Default	Type	Range
ScriptDirectory	An optional directory specifying where to find any custom Matlab files.		string	
MatlabSetUp	Matlab command to execute during begin method		filename	
MatlabFunction	Matlab command to execute for each simulation sample		filename	
MatlabWrapUp	Matlab command to execute during wrapup method		filename	
NumberOfFirings	number of invocations during a simulation	1	int	[1, ∞)

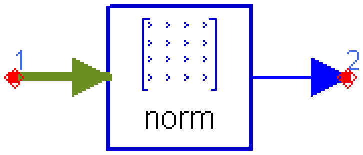
Pin Inputs

Pin	Name	Description	Signal Type
1	input		multiple anytype

Notes/Equations

1. MatlabSinkF behaves like other Matlab components except that it does not have any output. The total amount of data collected is determined by NumberOfFirings.
2. MatlabSetup, MatlabFunction, and MatlabWrapUp inputs accept script files only.
3. ScriptDirectory is an optional directory specifying where to find any custom Matlab files referenced inside MatlabSetup, MatlabFunction, and MatlabWrapUp scripts.
4. For more information about Matlab components, refer to *MATLAB Cosimulation Introduction* (ptolemy).
5. For more information regarding numeric matrix component signals, refer to *Numeric Matrix Components* (numeric).

Norm_M



Description Matrix Norm
Library Numeric Matlab

parameters

Name	Description	Default	Type
p	Indicates whether to use a 1-norm or 2-norm	two_norm	real enum

Pin Inputs

Pin	Name	Description	Signal Type
1	A	Input matrix	complex matrix

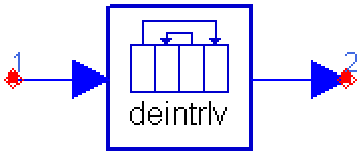
Pin Outputs

Pin	Name	Description	Signal Type
2	n	Matrix norm	real

Notes/Equations

1. This model computes the norm of the input matrix using the MATLAB norm function, `[n]= norm(A, p)`. For details on the MATLAB implementation of this computation, visit The MathWorks website at http://www.mathworks.com .
2. The 1-norm takes the sum of the absolute values of the elements in each column, and returns the maximum over the columns. The 2-norm returns the largest singular value of the input matrix.
3. The matrix norm provides a measure of the magnitude of the matrix elements, and therefore gives an indication of the energy in a matrix (or vector) of signals.

RandDeintrlv



Description Deinterleave Data Interleaved Using Random Permutation
Library Numeric Matlab

Parameters

Name	Description	Default	Type
state	State of the MATLAB random number generator	1234	int
BlockSize	Block size of the interleaved data	128	int

Pin Inputs

Pin	Name	Description	Signal Type
1	idata	Interleaved input data	real

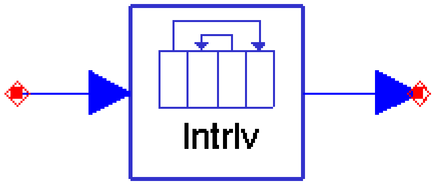
Pin Outputs

Pin	Name	Description	Signal Type
2	data	Deinterleaved data	real

Notes/Equations

1. This model performs deinterleaving of the data at the input using the MATLAB randdeintrlv function, `[data] = randdeintrlv(idata, state)`. It requires a license to the Communications Toolbox. For details on the MATLAB implementation of this computation, visit The MathWorks website at http://www.mathworks.com .
2. This function takes randomly arranged data presented on the input (pin 1) and presents a rearranged version of the data on the output (pin 2).
3. The interleaving is repeatable if the value of the state parameter is unchanged.
4. If the input interleaved data was created with the randinterlv model, then this model can reconstruct the original sequence if it is called with the same value of the state and BlockSize parameters.

RandIntrlv



Description Interleave Input Data Stream Using Random Permutation
Library Numeric Matlab

Parameters

Name	Description	Default	Type
state	State of the MATLAB random number generator	1234	int
BlockSize	Block size of the data to interleave	128	int

Pin Inputs

Pin	Name	Description	Signal Type
1	data	Input data stream	real

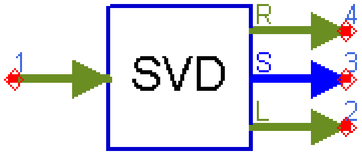
Pin Outputs

Pin	Name	Description	Signal Type
2	idata	Interleaved data stream	real

Notes/Equations

1. This model performs interleaving of the data at the input using the MATLAB randintrlv function, `[idata] = randintrlv(data, state)`. It requires a license to the Communications Toolbox. For details on the MATLAB implementation of this computation, visit The MathWorks website at http://www.mathworks.com .
2. This function takes data presented on the input (pin 1) and presents a randomly rearranged version of the data on the output (pin 2).
3. The interleaving is repeatable if the value of the state parameter is unchanged.
4. To deinterleave the data, use the randdeintrlv model with the same value of the state and BlockSize parameters.
5. Interleaving is used to provide robustness against burst errors. If a burst of interference results in the loss of several sequential data symbols, it becomes difficult to recover the lost data even when error control coding is applied. However, if the data is rearranged, the lost symbols are interspersed in the data stream, resulting in a scenario in which error control coding will be more effective in recovering the lost data.

SVD_Cx_M



Description Singular value decomposition of a general matrix
Library Numeric Matlab

Pin Inputs

Pin	Name	Description	Signal Type
1	X	Input matrix	complex matrix

Pin Outputs

Pin	Name	Description	Signal Type
2	L	Left singular vectors	complex matrix
3	S	Singular values	real matrix
4	R	Right singular vectors	complex matrix

Notes/Equations

1. This model performs the singular value decomposition of a complex matrix using the MATLAB svd function, `[R, S, L] = svd(X)`. For details on the MATLAB implementation of this computation, visit The MathWorks website at http://www.mathworks.com .
2. The singular value decomposition takes a complex matrix on the input (pin 1) and outputs two complex matrices whose columns represent the left and right singular vectors of the input (pins 2 and 4) and a diagonal matrix whose real entries represent the singular vectors of the input (pin 3). It can be applied to any matrix, whether or not it is square.
3. In communication system and signal processing analysis, the singular value

decomposition is useful for determining the important subspace of a general matrix entity. The singular vectors represent a basis for the matrix, while the singular values indicate the relative importance of each basis vector.