



February 2011
RF System DesignGuide

© **Agilent Technologies, Inc. 2000-2011**

5301 Stevens Creek Blvd., Santa Clara, CA 95052 USA

No part of this documentation may be reproduced in any form or by any means (including electronic storage and retrieval or translation into a foreign language) without prior agreement and written consent from Agilent Technologies, Inc. as governed by United States and international copyright laws.

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. FLEXIm 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>

SuperLU_MT version 2.0 - Copyright © 2003, The Regents of the University of California, through Lawrence Berkeley National Laboratory (subject to receipt of any required approvals from U.S. Dept. of Energy). All rights reserved. SuperLU Disclaimer: THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF 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).

RF System QuickStart Guide	7
Using DesignGuides	7
Basic Procedures	8
Transmitter/Receiver Design	9
Transmitter/Receiver Topologies	11
Co-Simulation Topologies	12
Communication Examples	12
Display Simulation Results	13
RF System DesignGuide Reference	14
Using the RF System DesignGuide	14
Transmitter/Receiver Design	15
Additional Analog/RF Sources	23
cdma2000	24
WCDMA (3GPP)	40
GSM/EDGE (3GPP)	50
WLAN	60
Transmitter/Receiver Topologies	68
Co-Simulation Topologies	69
Communication Examples	71
Reference	73
User Defined Functions	75
Usage and Operation	75

RF System QuickStart Guide

This QuickStart Guide is intended to help you get started using the RF System Design Guide effectively. For detailed reference information, refer to *RF System DesignGuide Reference* (dgrfsys).

The RF System DesignGuide has many simulation set-ups and data displays that are very useful for designing a communication system. The simulation setups are categorized by the System configuration, simulation technique, and type of topology. This DesignGuide is not a complete solution for all types of measurements and topologies, but covers the most common approaches. Subsequent releases of this DesignGuide will include an expanded range of features.

Note

This documentation is written describing and showing access through the selection dialog box method. If you are running the program through the cascading menu preference, the appearance and interface will be slightly different.

Using DesignGuides

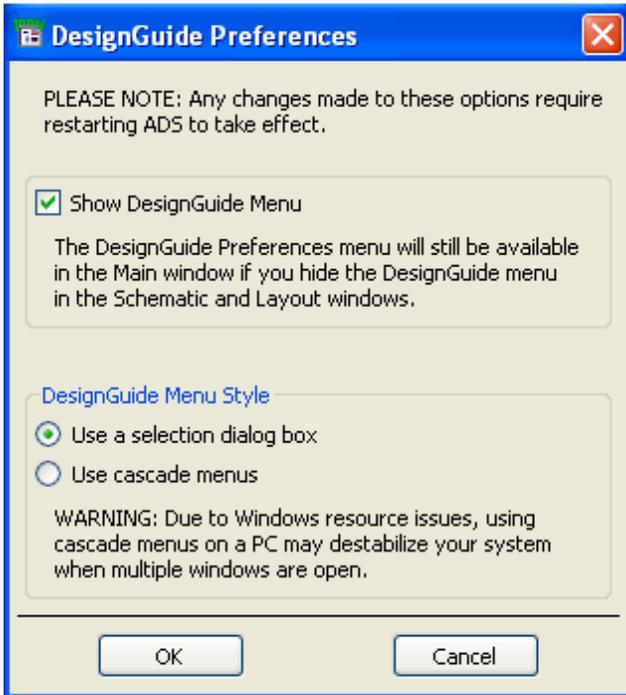
All DesignGuides can be accessed in the Schematic window through either cascading menus or dialog boxes. You can configure your preferred method in the Advanced Design System Main window.

Select the **DesignGuide** menu from Advanced Design System Main window.

The commands in this menu are as follows:

- **DesignGuide Developer Studio > Developer Studio Documentation** is only available on this menu if you have installed the DesignGuide Developer Studio. It brings up the DesignGuide Developer Studio documentation. Another way to access the Developer Studio documentation is by selecting **Help > Topics and Index > DesignGuides > DesignGuide Developer Studio** (from any ADS program window).
- **DesignGuide Developer Studio > Start DesignGuide Studio** is only available on this menu if you have installed the DesignGuide Developer Studio. It launches the initial Developer Studio dialog box.
- **Add DesignGuide** brings up a directory browser in which you can add a DesignGuide to your installation. This is primarily intended for use with DesignGuides that are custom-built through the Developer Studio.
- **List/Remove DesignGuide** brings up a list of your installed DesignGuides. Select any that you would like to uninstall and choose the *Remove* button.
- **Preferences** brings up a dialog box that allows you to:

- Disable the DesignGuide menu commands (all except Preferences) in the Main window by unchecking this box. In the Schematic and Layout windows, the complete DesignGuide menu and all of its commands will be removed if this box is unchecked.
- Select your preferred interface method (cascading menus vs. dialog boxes).



Close and restart the program for your preference changes to take effect.

Note
On PC systems, Windows resource issues might limit the use of cascading menus. When multiple windows are open, your system could become destabilized. Thus the dialog box menu style might be best for these situations.

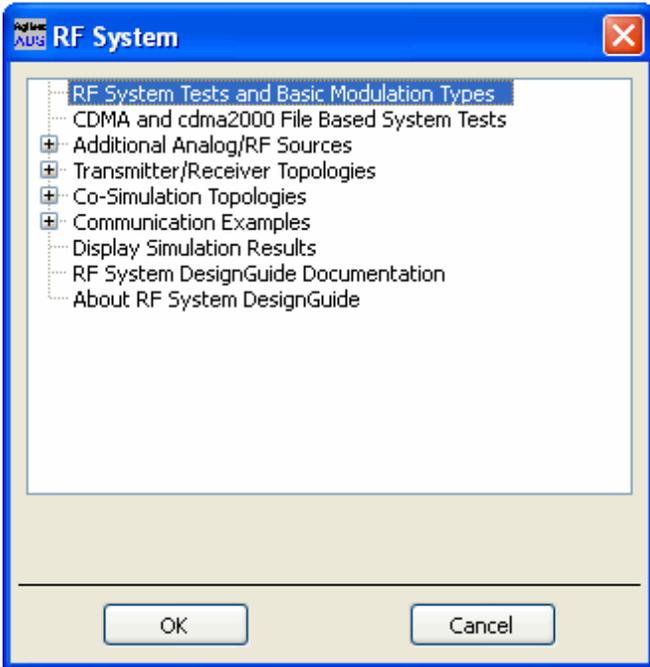
Accessing the Documentation

To access the documentation for the DesignGuide, select either of the following:

- **DesignGuide > RF System > RF System DesignGuide Documentation** (from ADS Schematic window)
- **Help > Topics and Index > DesignGuides > RF System** (from any ADS program window)

Basic Procedures

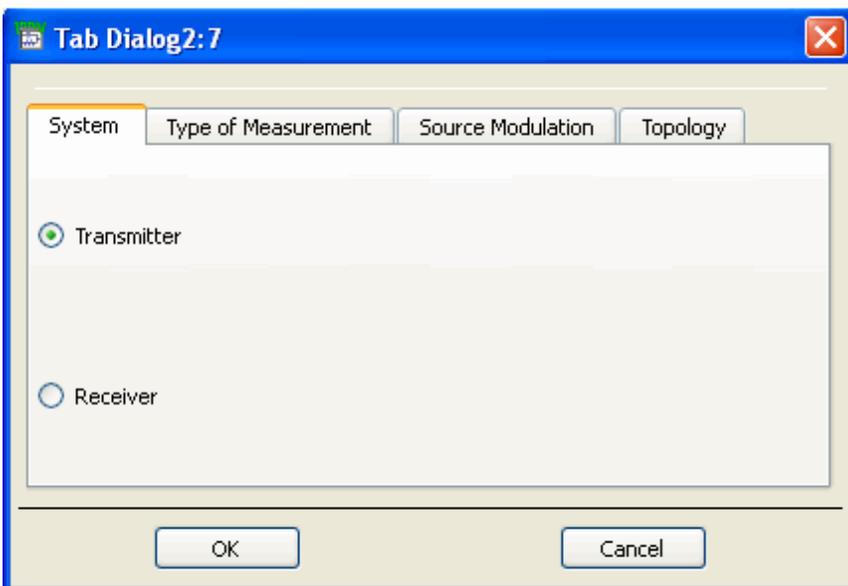
The features and content of the RF System DesignGuide are accessible from the **DesignGuide menu** in the ADS Schematic window.



Transmitter/Receiver Design

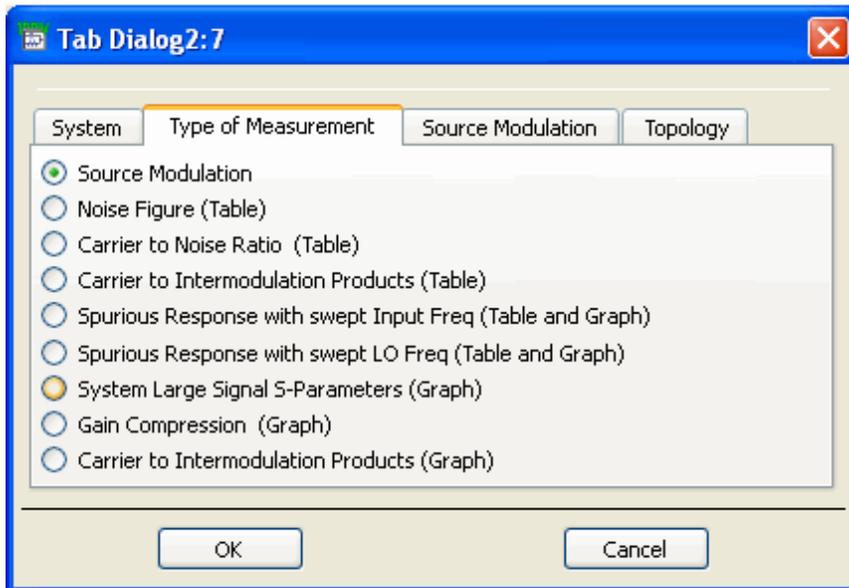
Using a dialog box labeled Transmitter/Receiver Design on the menu, you select your desired communication configuration, as shown here.

The first step is to select one of the two available system configurations as shown in below dialog box.



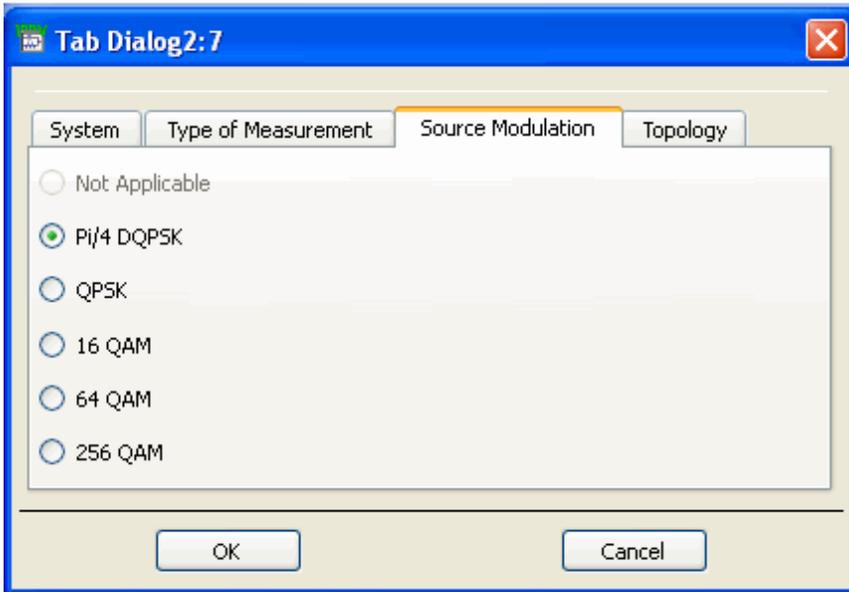
Having identified the type of communication system, you then select one of the nine simulations available from the Simulation tab, as shown in the illustration that follows. The simulations include

- Source Modulation (you choose the type of modulation on the next tab)
- Noise Figure (Tabular Format)
- Carrier to Noise Ratio (Tabular Format)
- Carrier to Intermodulation Distortion (Tabular Format)
- Spurious Response (Table and Graph)
- Spurious Response as a function of the LO frequency (Table and Graph)
- S Parameters (Graphical Format)
- Gain Compression (Graphical Format)
- Carrier to Intermodulation Distortion with swept RF input power (Graphical Format)

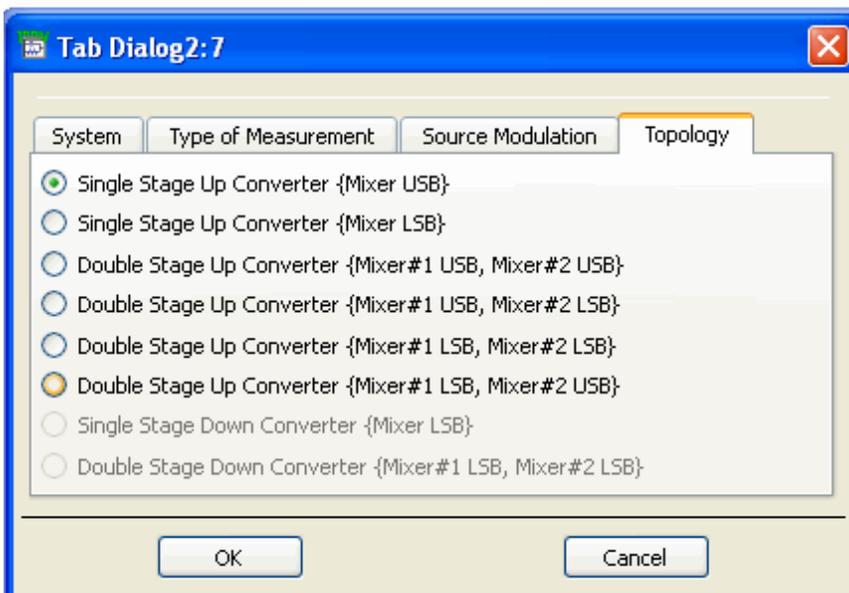


Having selected Source Modulation on the previous tab, you then have a choice of modulations. If you have developed your own source modulation, you can replace the source on the schematic. The **Not Applicable** box will be highlighted if you have not chosen source modulation on the previous tab.

The selection box for source modulations is shown here.



Shown here is the selection box for system topology. The grayed-out selections are not available with the previous tab selections. Click one of the available selections. For a detailed description of the topology selections, refer to *RF System DesignGuide* (dgrfsys). If the communication system is a transmitter, you have a choice of 6 topologies. If the communication system is a receiver, then you have a choice of 2 topologies. The topologies are functions of whether the upper sideband or lower sideband is used for each mixer.

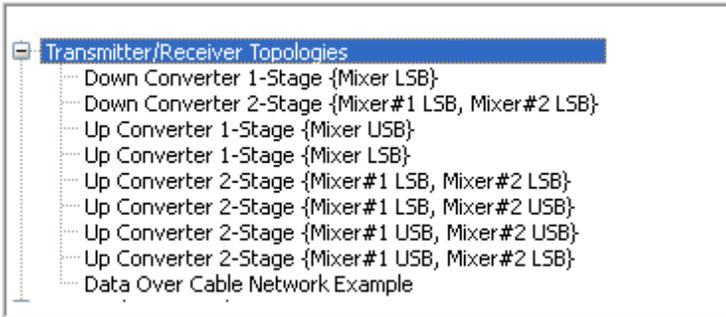


Transmitter/Receiver Topologies

The first step in most system designs is to select a topology. Typically a receiver uses

either one or two stages of frequency conversion. Most receiver systems use the lower sideband of the mixer output at each frequency translation and therefore there are only two topologies. Whereas, for transmitters, the configurations are more varied, this DesignGuide has topologies for each of the four double stage up converters, as well as the two possible single stage up converters. There is a Transmitter example included to demonstrate how the pre-defined topology can be modified to your configuration

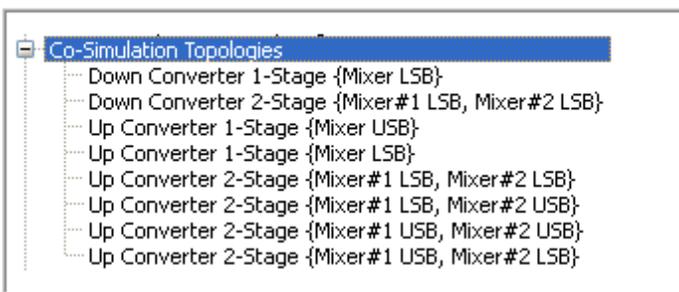
From the Schematic window, select **DesignGuide > RF System > Transmitter/Receiver Topologies**.



Co-Simulation Topologies

To use the system topology within a Ptolemy simulation, there are co-simulation topology modules to facilitate the process. The topology that has been designed and tested with the Type of Measurement schematics can then be entered into these modules for further testing in the Ptolemy environment.

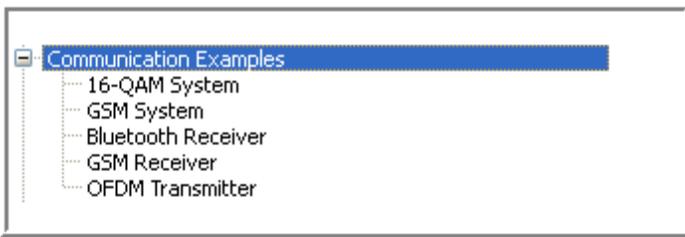
From the Schematic window, select **DesignGuide > RF System > Co-Simulation Topologies**.



Communication Examples

Some example files show how the co-simulation modules are inserted into the Ptolemy schematics.

From the Schematic window, select **DesignGuide > RF System > Communication Examples**.



Display Simulation Results

The **Display Simulation Results** menu selection is only present as a means of opening secondary data displays that correspond to a particular schematic. Most of the simulations with modulated signals have a second data display that shows the spectrum, power, and ACPR calculations. This secondary data display can only be opened via this menu selection. However, if you rename the schematic, the menu selection will not work anymore. For this reason, we recommend that you open the corresponding data display before renaming a schematic. After renaming schematics, you must open data displays using one of the standard ADS methods.

RF System DesignGuide Reference

The RF System DesignGuide is integrated into Agilent EESof's Advanced Design System environment, working as an interactive handbook for the creation of useful designs. The Guide contains many templates to be used in the ADS software environment. These templates can assist the system developer in designing a transmitter/receiver to meet performance specifications.

You can use the Transmitter/Receiver Topologies templates to define a system configuration, then proceed to characterize the performance with respect to quantities such as Noise Figure, Carrier-to-Noise Ratio, Carrier-to-Intermodulation Distortion, etc. The final performance measurement is then observed as the given modulation is applied and quantities such as ACPR and EVM are measured. If more sophisticated sources are required, the Co-Simulation Topologies can be used to easily interface with either a user-defined Ptolemy source or one of the available libraries. The DesignGuide provides a complete tool kit to interactively explore Transmitter/Receiver systems at the top level as part of an integrated design process.

In addition to the requirements of the Agilent EESof ADS software, the RF System DesignGuide requires approximately 60 MBytes of additional storage space.

Note

This document assumes that you are familiar with all of the basic ADS program operations. For additional information, refer to the *Schematic Capture and Layout* documentation. For access to the complete set of ADS online documents, Choose *Help > Topics and Index* from an ADS program window.

Using the RF System DesignGuide

The key features of the RF System DesignGuide follow.

- Predefined system characterization templates and data displays:
 - Noise Figure
 - Carrier to Noise Ratio
 - Carrier to Intermodulation Distortion
 - Spurious Response
 - Spurious Response as a function of the LO frequency
 - S Parameters
 - Gain Compression
 - Carrier to Intermodulation Distortion with swept RF input power

Predefined templates and data displays for various modulation sources:

- pi/4 DQPSK
- QPSK
- 16 QAM, 64 QAM, 256 QAM
- IS95 and CDMA2000 (reverse and forward link)
- Co-Simulation Links to Ptolemy Libraries

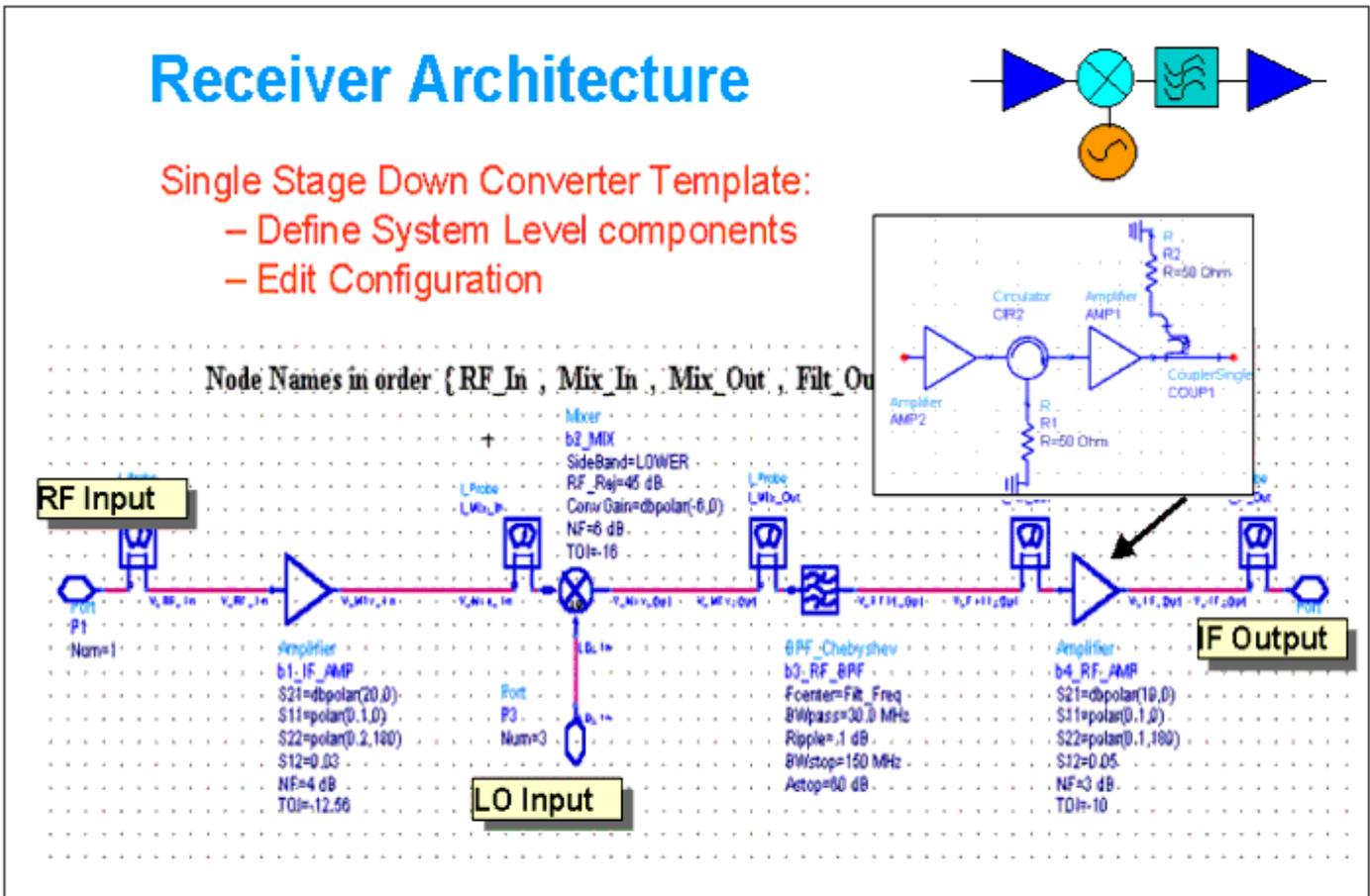
- GSM, OFDM and BlueTooth Ptolemy examples
- Predefined Transmitter and Receiver Topologies:
 - Single Stage Down Converter (mixer LSB)
 - Double Stage Down Converter (mixer #1 LSB, mixer #2 LSB)
 - Single Stage Up Converter (mixer LSB)
 - Single Stage Up Converter (mixer USB)
 - Double Stage Up Converter (mixer #1 LSB, mixer #2 LSB)
 - Double Stage Up Converter (mixer #1 LSB, mixer #2 USB)
 - Double Stage Up Converter (mixer #1 USB, mixer #2 LSB)
 - Double Stage Up Converter (mixer #1 USB, mixer #2 USB)
- Co-Simulation Topologies:
 - Single Stage Down Converter (mixer LSB)
 - Double Stage Down Converter (mixer #1 LSB, mixer #2 LSB)
 - Single Stage Up Converter (mixer LSB)
 - Single Stage Up Converter (mixer USB)
 - Double Stage Up Converter (mixer #1 LSB, mixer #2 LSB)
 - Double Stage Up Converter (mixer #1 LSB, mixer #2 USB)
 - Double Stage Up Converter (mixer #1 USB, mixer #2 LSB)
 - Double Stage Up Converter (mixer #1 USB, mixer #2 USB)
- Communication Examples:
 - 16 QAM complete with Transmitter, Receiver and LOS Link
 - GSM complete with Transmitter, Receiver and LOS Link and co-simulator
 - BlueTooth Ptolemy example with Receiver and co-simulator
 - GSM Ptolemy example with Receiver and co-simulator
 - OFDM Ptolemy example with Receiver and co-simulator

Transmitter/Receiver Design

The general design flow for a transmitter or receiver consists of a three-step process before a prototype is developed. The first step is the creation of the frequency converter topology; this is followed by performing system measurements to ensure that the specifications can be met. The system tests usually involve Carrier to Noise Ratio, Carrier to Intermodulation, Noise Figure, Spurious Response, and others. This initiates an iterative process between the choice of system architecture and the system measurements. Once a candidate design has been optimized, measurements can be performed using the required signal modulation. These tests may involve Adjacent Channel Power Ratio, Error Vector Magnitude, Peak to Average ratio and others.

The system designer identifies a candidate topology from which he then proceeds to define the individual component specifications. Many designers focus initially on the mixers and filter requirements. The mixers can play a significant role in setting the system Dynamic range and reducing costs. Minimizing the number of components at the higher frequencies will typically reduce the system costs. Filters are placed at strategic points in the line-up to eliminate spurious signals and control the CNR. Attention is given to the filters to ensure that their individual order is not excessive, which would increase the loss and size. The gain stages can then be placed throughout the receiver to minimize the system Noise Figure (increase sensitivity) and maintain a good CNR throughout the chain. As one gets closer to a candidate topology, the individual component specifications can be

filled out. This is an iterative process in which the system designer converges on a suitable transmitter/receiver design that meets all his criteria.

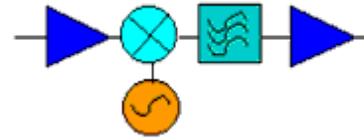


Having pre-configured topologies that can be readily modified to user configurations will aid the designer. Nodes and probe components are identified to facilitate the measurement process. The topologies all have assigned node and probe names that need to be maintained for the simulation to work properly. The nodes and probe names can be placed at various points along the chain, but all names must be included. (Be careful with the syntax). The more advanced user can alter the names as well as add additional names but the schematics as well as the data displays will need to be modified.

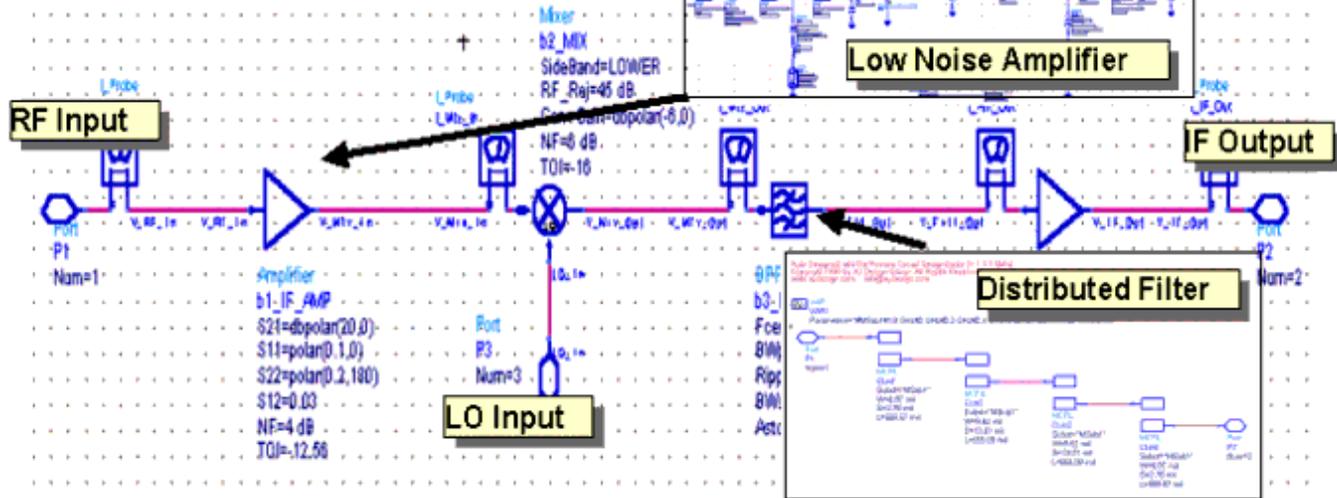
After a suitable review, the system designer then passes the individual component specifications on to the component designers. The component designers then proceed to develop a CAD representation of the individual components. These designs can be inserted in place of the system level building blocks to investigate their overall impact on the performance. This allows the assembly of a complete CAD transmitter/receiver design, which can be further tested.

Receiver Architecture

Single Stage Down Converter Template:
– Insert Designed components



Node Names in order { RF_In , Mix_In , Mix

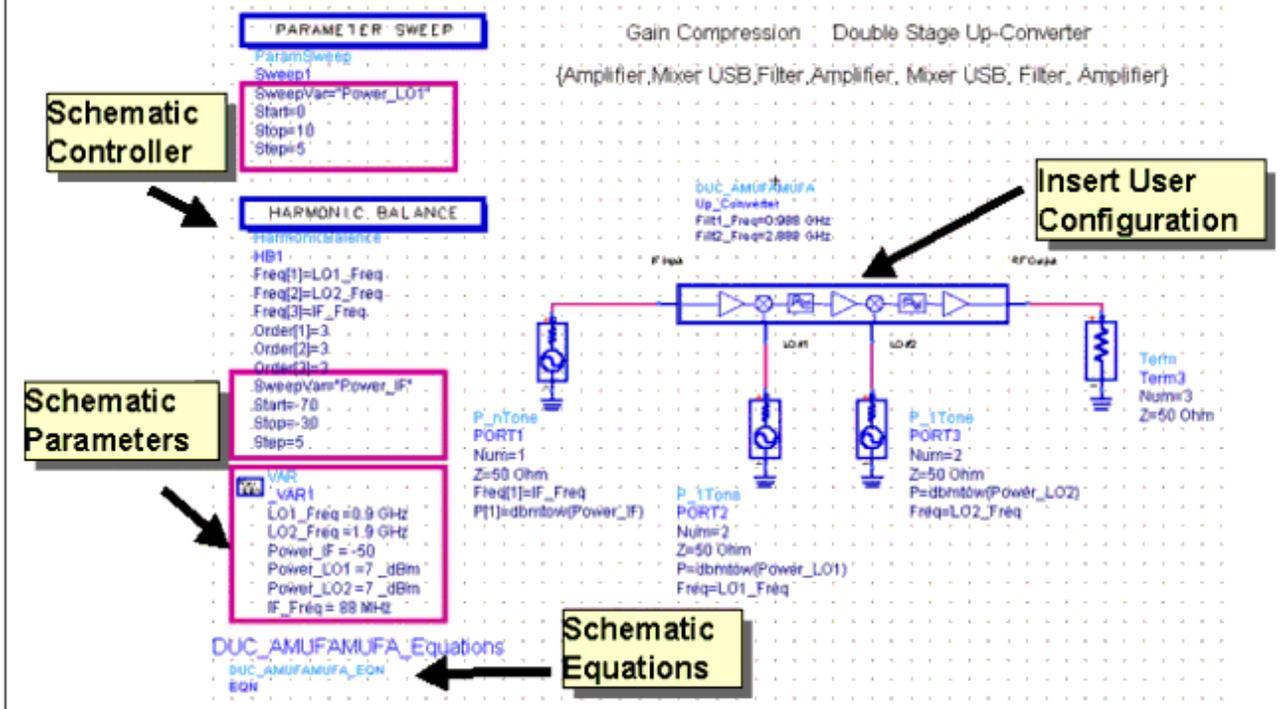


The designed transmitter/receiver performance then needs to be evaluated. Pre-configured schematics and data displays that you can readily modify for your system will speed up the design process. Replace the place-holder with your candidate transmitter or receiver design. The pre-configured schematics are constructed for a particular system measurement. Then edit the schematic parameters and parameter sweep ranges to reflect the desired system. Schematic equations are included to aid the extraction of key measurements such as power, noise, spurious levels, gain, ripple, etc.

Various schematic controllers are used wherever appropriate to reduce the simulation time. Some Schematic Controllers have sweep variables that allow you to observe the sensitivity to various parameters. The Schematic Parameters are central to the design. The Frequencies and Powers of the various sources are typical. These parameters need to be adjusted for your design. The user-designed transmitter/receiver topology needs to be inserted in place of the existing preassigned topology (Insert User Configuration). The Schematic Equations are used to extract measurements pertinent to the simulation.

RF System Measurements

Pre-Configured Schematic



An example of an important measurement is the carrier to noise ratio. The noise limitations may arise from the noise floor of the individual components or it could be from the excessive phase noise of the local oscillators. You also have the flexibility of entering an input noise source.

RF System Measurements

CNR Data Display

Single Stage Down Converter

Measurement: Carrier to Noise Ratio

Configuration: Amplifier-Mixer-Filter-Amplifier

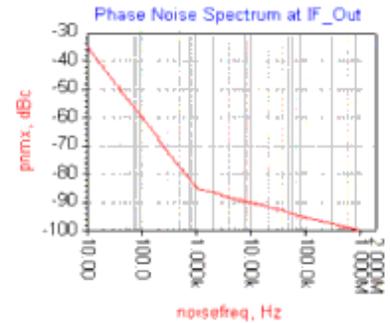
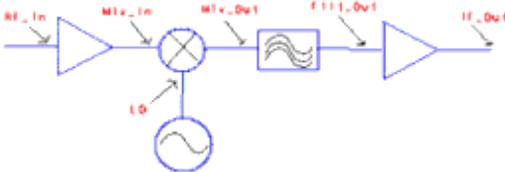
1. The system output power at various nodes in the down converter is listed.
2. The carrier to noise ratio at various nodes is also listed.
3. The phase noise spectrum is shown in the figure.

Power at Nodes (dBm)

RF_In_dBm	Mix_In_dBm	Mix_Out_dBm	Filt_Out_dBm	IF_Out_dBm
-47.062	-27.363	-33.772	-34.784	-24.888

CNR at Nodes (dB)

CNR_RF_In	CNR_Mix_In	CNR_Mix_Out	CNR_Filt_Out	CNR_IF_Out
137.114	133.904	78.756	78.627	79.499



```

200 Noise_IF_Out[2]NO1DownConverter3_IF_Out[noise] "200
201 CNR_IF_Out[2]CNR1DownConverter3_IF_Out[3]
202
203 Noise_Mix_Out[2]NO1DownConverter3_Mix_Out[noise] "203
204 CNR_Mix_Out[2]CNR1DownConverter3_Mix_Out[3]
205
206 Noise_Mix_In[2]NO1DownConverter3_Mix_In[noise] "206
207 CNR_Mix_In[2]CNR1DownConverter3_Mix_In[3]
    
```

Phase Noise at IF Output

There are two factors that limit the transceiver system dynamic range:

- Carrier-to-Noise Floor Ratio
- Carrier-to- Intermodulation Level

The C/IMD performance is limited by the third order intercept point (TOI) of the individual components. The TOI at various nodes in the chain can be estimated as a function of the power levels of the fundamental and intermodulation signals. Or the actual C/IMD levels can be plotted as a function of the input power level. This is a useful means for determining the dynamic range limiting components.

RF System Measurements

Intermodulation Distortion Data Display

Double Stage Up Converter

Measurement: Carrier to Intermodulation Distortion

Configuration: Amplifier-Mixer USB-Filter-Amplifier-Mixer LSB-Filter-Amplifier

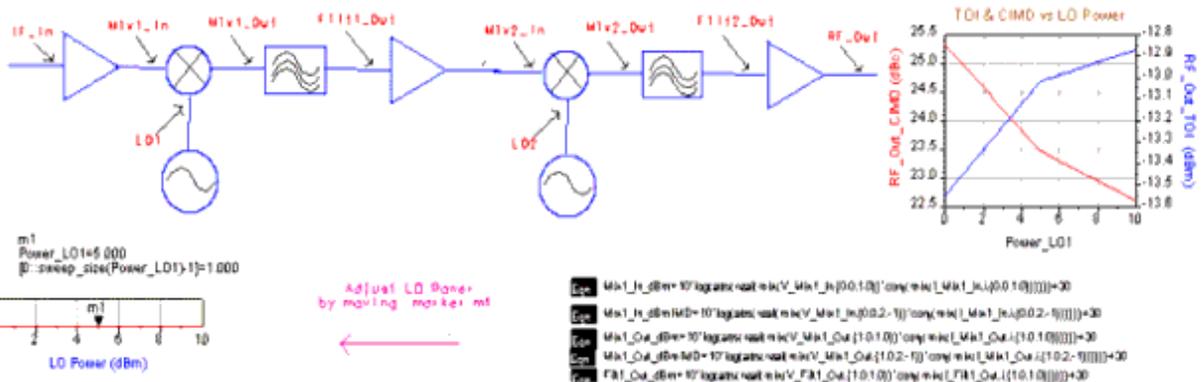
1. The third order intercept point at the identified system nodes is shown.
2. The carrier to third order intermodulation distortion is shown.

3rd Order Intercept Point

Mix1_In_TOI[m1]	Mix1_Out_TOI[m1]	Flt1_Out_TOI[m1]	Mix2_In_TOI[m1]	Mix2_Out_TOI[m1]	Flt2_Out_TOI[m1]	RF_Out_TOI[m1]
-10.404	-19.302	-20.378	-12.286	-20.320	-21.345	-13.016

Node Carrier to IMD

Mix1_In_CIMD[m1]	Mix1_Out_CIMD[m1]	Flt1_Out_CIMD[m1]	Mix2_In_CIMD[m1]	Mix2_Out_CIMD[m1]	Flt2_Out_CIMD[m1]	RF_Out_CIMD[m1]
34.397	31.265	31.294	28.020	25.948	25.948	25.473



The introduction of nonlinear components into the transmitter/receiver generates unwanted spurious signals that need to be controlled. Spurious signals that fall in-band will degrade the BER performance for the given channel. Spurs that fall out-of-band may contaminate other channels or they may be aliased into the given channel. The data display below is useful for identifying the levels of the various spurs as well as assisting in tracking the component from which they originate. The data display can have variable parameters such as the RF input frequency, power levels, LO frequencies, etc. This can aid in the optimization of the system chain.

RF System Measurements

Spurious Response Data Display

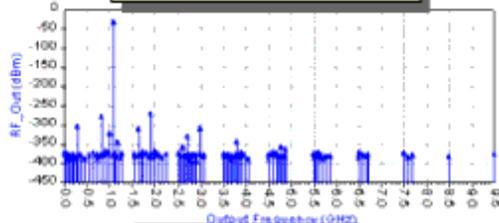
Double Stage Up Converter

Measurement: Spurious Response

Configuration: Amplifier-Mixer-LSB-Filter-Amplifier-Mixer-LSB-Filter-Amplifier

1. This plot shows the spectrum near the RF when the IF input frequency is at the location indicated by marker m1 in the lower right plot.
2. Move the marker to a different IF frequency, and the lower left plot and table will be updated.
3. Set harmonic order, filter bandwidth, and mixing terms of Spur.

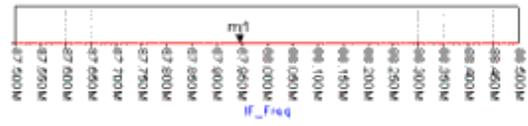
Output Spectrum



List of Spurs

RF_Harmonic Index	Spur Out (dBc)
25	0.7005000
41	-240.177
17	-248.761
9	-271.833
65	-277.408
33	-279.218
23	-291.987
66	-297.343
76	-310.710
28	-314.506
92	-320.432
53	-327.787
95	-330.994
21	-337.506
45	-338.238
37	-338.770

Tunable Parameters



Adjust IF Frequency by moving marker m1

Fundamental Output Power (dBm)

-23.257

Select harmonic

E.g. Index=8

Frequency at Index

275.9MHz

Output Spurs

Mixing Terms at Index

LO1 Mult	LO2 Mult	RF Mult
-2	1	2

Once a candidate design has been developed and has been tested against the standard system level specifications, you can introduce a modulated source. The modulated signal can either be created using the Analog/RF components or in Ptolemy, with a co-simulation used to translate between Ptolemy and the Analog/RF transmitter or receiver.

Applying the modulated source to the transmitter/receiver will allow for more fundamental measurements such as BER, EVM, Average Power Output, Peak to Average Ratio, ACPR, etc. The testing based on a two tone (3dB Peak/Average Ratio) will not give the appropriate measurement of the ACPR when, for example, a CDMA signal is applied. The BER measurement can take a significant amount of time to simulate. Therefore, a measurement of the Error Vector Magnitude can serve as an intermediate evaluation of the performance of the system. The ultimate goal would be to run a Ptolemy simulation for a long enough period of time to evaluate the BER.

System Modulation Measurements

Pi/4 DQPSK Data Display

Single Stage Up Converter

Measurement: Pi/4 DQPSK Modulation

Configuration: Amplifier-Mixer-USB-Filter-Amplifier

1. The input and output spectral response are shown as a function of offset frequency.
2. The peak to average power is displayed as a function of time.
3. The Output power as well as the ACPR is listed.
4. The ACPR mask limits can be adjusted.

Spectrum calculations

```

Spectrum_in@recto\Fwd_in...:kaiser()
Spectrum_out@recto\Fwd_out...:kaiser()
    
```

Adjacent-channel power calculations

```

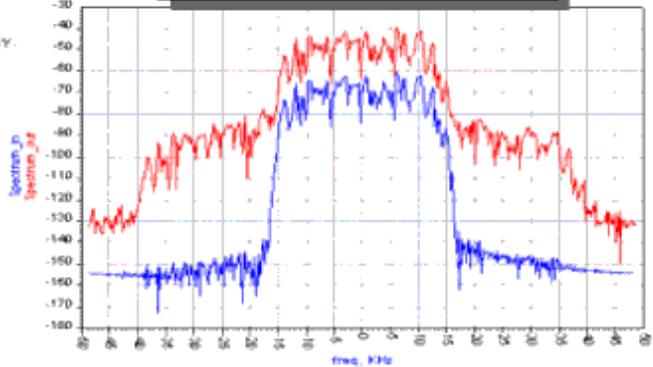
mask_lower()=10*log(10)
mask_upper()=10*log(10)
TmaskACPR_lower@recto\Fwd_in...:kaiser()
TmaskACPR_upper@recto\Fwd_out...:kaiser()
    
```

Mean power, peak power, and peak-to-average power calculations

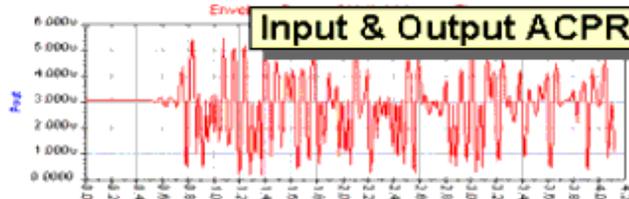
```

PoutPower=mean(Pout)
Pout_avg=10*log(mean(Pout))
Pout_dBm=10*log(mean(Pout))+30
    
```

Spectral Response



Input & Output ACPR



Pout & Peak/Avg Power

Lower Channel ACPR (Input) dBc	-92.55	Upper Channel ACPR (Input) dBc	-62.43
Lower Channel ACPR (Output) dBc	-44.66	Upper Channel ACPR (Output) dBc	-44.42
Pout (dBm)	-25.49	Peak-to-Ave (linear)	1.940
		Peak-to-Ave (dB)	2.879

The Error Vector Magnitude measurement would provide you with an initial indication of the performance achievable by the transmitter/receiver. The constellation diagrams help in giving insight into the contribution of the distortion. As an example, if compression and phase rotation were observed on the constellation, this could be traced back to the overdriving of a particular stage in the chain.

cdma2000

ADS Designs ()

- cdma2000ForwardSR1RC12Src
- cdma2000ForwardSR1RC35Src
- cdma2000ForwardSR3RC69Src
- cdma2000ReverseSR1RC12Src
- cdma2000ReverseSR1RC34Src
- cdma2000ReverseSR3RC56Src
- TestFwdSR1RC12
- TestFwdSR1RC35
- TestFwdSR3RC69
- TestRevSR1RC12
- TestRevSR1RC34
- TestRevSR3RC56

Data Display Files (.dds)

- TestFwdSR1RC12
- TestFwdSR1RC35
- TestFwdSR3RC69
- TestRevSR1RC12
- TestRevSR1RC34
- TestRevSR3RC56

Specifications

All reference to cdma2000 specification sections, figures and tables, are to:
[1]3GPP2 C.S0002-1 Version 6.0 (February 8, 2002) " *Physical Layer Standard for cdma2000 Spread Spectrum Systems - Release A* ".

Reverse Link

For reverse link transmission eight different physical channels are defined. For reference, the reverse physical channels are set out in the following table. The *italicized* physical channels are *NOT* supported by the ADS uplink source.

Reverse Physical Channel	Acronym	Section of [1]
Fundamental Channel	R-FCH	2.1.3.7
Dedicated Control Channel	R-DCCH	2.1.3.6
Supplemental Code Channel	R-SCCH	2.1.3.9
Supplemental Channel	R-SCH	2.1.3.8
Access Channel	R-ACH	2.1.3.3
<i>Common Control Channel</i>	<i>R-CCCH</i>	<i>2.1.3.5</i>
Pilot Channel	R-PICH	2.1.3.2
<i>Enhanced Access Channel</i>	<i>R-EACH</i>	<i>2.1.3.4</i>

The different data rates supported by each of the reverse link physical channels are detailed in the preceding and following tables. The *italicized* data rates *ARE* supported by the ADS reverse link sources.

Reverse Link Physical Channel Data Rates for SR1

Reverse Channel	Data Rates \ (kbps)
Access	4.8
Enhanced Access	Header 9.6 Data 38.4 \ (5, 10, or 20 ms frames) 19.2 \ (10 or 20 ms frames), or 9.6 \ (20 ms frames)
Common Control	38.4 \ (5, 10, or 20 ms frames) 19.2 \ (10 or 20 ms frames), or 9.6 \ (20 ms frames)
Dedicated Control	RC3 9.6 RC4 14.4 \ (20 ms frames) or 9.6 \ (5 ms frames)
Fundamental	RC1 <i>9.6, 4.8, 2.4, or 1.2</i> RC2 <i>14.4, 7.2, 3.6, or 1.8</i> RC3 9.6, 4.8, 2.7, or 1.5 \ (20 ms frames) or 9.6 \ (5 ms frames) RC4 14.4, 7.2, 3.6, or 1.8 \ (20 ms frames) or 9.6 \ (5 ms frames)
Supplemental Code	RC1 9.6 RC2 <i>14.4</i>
Supplemental	RC3 307.2, 153.6, 76.8, 38.4, 19.2, 9.6, 4.8, 2.7, or 1.5 \ (20 ms frames) 153.6, 76.8, 38.4, 19.2, 9.6, 4.8, 2.4, or 1.35 \ (40 ms frames) 76.8, 38.4, 19.2, 9.6, 4.8, 2.4, or 1.2 \ (80 ms frames) RC4 230.4, 115.2, 57.6, 28.8, 14.4, 7.2, 3.6, or 1.8 \ (20 ms frames) 115.2, 57.6, 28.8, 14.4, 7.2, 3.6, or 1.8 \ (40 ms frames) 57.6, 28.8, 14.4, 7.2, 3.6, or 1.8 \ (80 ms frames)

Reverse Link Physical Channel Data Rates for SR3

Reverse Physical Channel		Data Rates \ (kbps)
Access		4.8
Enhanced Access	Header	9.6
	Data	38.4 \ (5, 10, or 20 ms frames) 19.2 \ (10 or 20 ms frames), or 9.6 \ (20 ms frames)
Common Control		38.4 \ (5, 10, or 20 ms frames) 19.2 \ (10 or 20 ms frames), or 9.6 \ (20 ms frames)
Dedicated Control	RC5	9.6
	RC6	14.4 \ (20 ms frames) or 9.6 \ (5 ms frames)
Fundamental	RC5	_9.6, 4.8, 2.7, or 1.5 \ (20 ms frames) or 9.6 \ (5 ms frames)
	RC6	14.4, 7.2, 3.6, or 1.8 \ (20 ms frames) or 9.6 \ (5 ms frames)
Supplemental	RC5	614.4, 307.2, 153.6, 76.8, 38.4, 19.2, 9.6, 4.8, 2.7, or 1.5 \ (20 ms frames) 307.2, 153.6, 76.8, 38.4, 19.2, 9.6, 4.8, 2.4, or 1.35 \ (40 ms frames) 153.6, 76.8, 38.4, 19.2, 9.6, 4.8, 2.4, or 1.2 \ (80 ms frames)
	RC6	1036.8, 460.8, 230.4, 115.2, 57.6, 28.8, 14.4, 7.2, 3.6, or 1.8 \ (20 ms frames) 518.4, 230.4, 115.2, 57.6, 28.8, 14.4, 7.2, 3.6, or 1.8 \ (40 ms frames) 259.2, 115.2, 57.6, 28.8, 14.4, 7.2, 3.6, or 1.8 \ (80 ms frames)

Forward Link

Like the reverse link, a series of physical channels are defined for the forward link. For reference, the forward link physical channels are set out in the following table. The *italized* physical channels *ARE* supported by the ADS downlink sources.

Forward Physical Channel	Acronym	Section of [1]
<i>Fundamental Channel</i>	<i>F-FCH</i>	<i>3.1.3.11</i>
<i>Dedicated Control Channel</i>	<i>F-DCCH</i>	<i>3.1.3.10</i>
<i>Supplemental Code Channel</i>	<i>F-SCCH</i>	<i>3.1.3.13</i>
<i>Supplemental Channel</i>	<i>F-SCH</i>	<i>3.1.3.12</i>
Paging Channel	F-PCH	3.1.3.4
Quick Paging Channel	F-QPCH	3.1.3.6
Common Control Channel	F-CCCH	3.1.3.9
Pilot Channel	F-PICH	3.1.3.2
Dedicated Auxiliary Pilot Channel	F-APICH	3.1.3.2
Transmit Diversity Pilot Channel	F-TDPICH	3.1.3.2
Auxiliary Transmit Diversity Pilot Channel	F-ATDPICH	3.1.3.2
Sync Channels	F-SYNCH	3.1.3.3
Common Power Control Channel	F-CPCCH	3.1.3.7
Common Assignment Channel	F-CACH	3.1.3.8
Broadcast Control Channel	F-BCCH	3.1.3.5

Usage and Operation

Six cdma2000 ADS modulated source components have been developed - three for the reverse link and three for the forward link. The design files of the source components and their associated sub-components are listed in [ADS Designs \(\)](#). Each of the three source components for the reverse link, namely, *cdma2000ReverseSR1RC12Src*, *cdma2000ReverseSR1RC34Src*, and *cdma2000ReverseSR3RC56Src*, implements two of the 6 reverse link radio configurations. The name of the component identifies which radio configurations it implements, as well as the spreading rate of the radio configurations. Similarly, the three forward link source components, namely, *.cdma2000ForwardSR1RC12Src*, *cdma2000ForwardSR1RC35Src*, and *cdma2000ForwardSR3RC69Src* implement the nine radio configurations of the forward link. The forward link components implement radio configurations 1 and 2, 3 through 5, and 6 through 9, respectively.

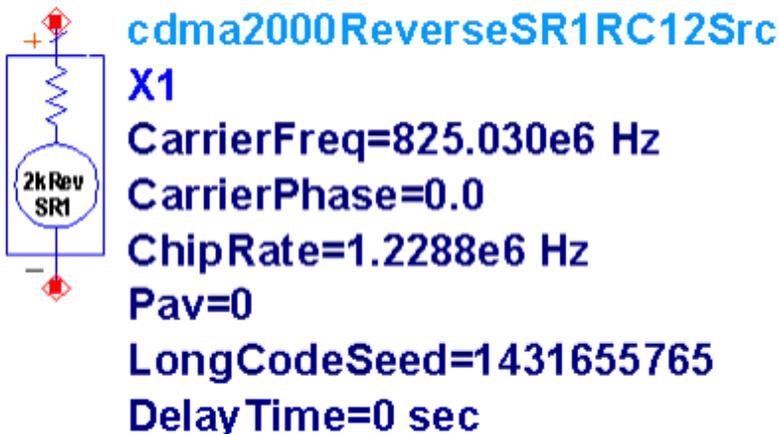
The file list of [ADS Designs \(\)](#) also includes six *Test...* designs. Each *Test...* design contains a simple simulation example of one of the sources.

Reverse Link

Simulation Domain

cdma2000ReverseSR1RC12Src

The *cdma2000ReverseSR1RC12Src* source is shown in the following figure.



ADS cdma2000 Reverse Link Signal Source for RC1 and RC2

The *cdma2000ReverseSR1RC12Src* component generates a continuously modulated single carrier DS-CDMA signal based upon the transport of:

- An RC1 or RC2 reverse traffic channel, (i.e., a reverse fundamental channel or a reverse supplemental code channel), operating at 9.6 kbps or 14.4 kbps respectively, or
- An access channel for SR1.
The channel structure for the reverse fundamental channel and reverse supplemental code channel for RC1 and RC2 are shown in Figures 2.1.3.1.1.1-6 and 2.1.3.1.1.1-7 of [1], respectively. The component parameters of the source are discussed below, where for each parameter a brief description is provided, as well as its numeric type and the range of values that it can take on, respectively:
- **CarrierFreq** : The carrier frequency of the output DS-CDMA RF modulated mobile station signal, real, $(ChipRate / 2) < Carrier_Freq$.
- **CarrierPhase** : The phase of the RF carrier, real, degrees.
- **ChipRate** : The direct sequence chip rate, real, $0 < ChipRate$. The source is intended to operate at a chip rate of 1.2288 Mcps.
- **Pav** : Relative average output power of the source, real, $0 < Pav$.
- **LongCodeSeed** : The initial state of the long spreading code LFSR, integer, $0 < LongCodeSeed < 2^{231} - 1$.
- **DelayTime** : The first chip generated by the source is applied to the baseband pulse shaping filters at $DelayTime$, real, $0 < DelayTime$.

The *CarrierPhase* parameter is necessary because the cdma2000 specification requires that the carrier phase of each reverse supplemental code channel be offset from that of a fundamental channel in accordance with the following table. [1](#)

Reverse Supplemental Code Channel	Carrier Phase Offset (radians)
1	$\pi/2$
2	$\pi/4$
3	$3\pi/4$
4	0
5	$\pi/2$
6	$\pi/4$
7	$3\pi/4$

The *cdma2000ReverseSR1RC12Src* component only supports a simulation time step that is equal to four time steps (samples) per chip. The simulation time step is limited in this manner because the coefficients of the baseband pulse shaping filters are hard coded. The pulse shaping filters are implemented using an FIR filter, with the coefficients being taken directly from the cdma2000 specification. [2](#)

The number of samples per chip depends upon the value of the *ChipRate* parameter and the simulation time-step specified in the *Envelope* simulation controller. Although the *ChipRate* parameter is included in the reverse link source's parameter list, it should remain fixed at its default value, (i.e., the cdma2000 specified chip rate of 1.2288 Mcps). As a result of the chip rate being fixed, the number of samples per chip will depend only upon the simulation time-step. It is up to the designer to set the simulation time step so as to produce the four samples per chip. The *TestRevSR1RC12* design provides an example of how to set the simulation time step.

cdma2000ReverseSR1RC34Src

The *cdma2000ReverseSR1RC34Src* source is shown in the following figure.

**cdma2000ReverseSR1RC34Src****X1****CarrierFreq=825.030e6 Hz****ChipRate=1.2288e6 Hz****Pav=0****DelayTime=0 sec****GainRPC=1.0****GainRFC=0.0****GainRSC1=0.0****GainRSC2=0.0****GainRDCC=0.0****ModSymRateRSC1=38.4e3 Hz****ModSymRateRSC2=307.2e3 Hz****ROut=50 Ohm****ADS cdma2000 Reverse Link Signal Sources for RC3 and RC4**

The *cdma2000ReverseSR1RC34Src* component generates a continuously modulated single carrier DS-CDMA mobile station signal using a chip rate of 1.2288 Mcps. The parameters of the *cdma2000ReverseSR1RC34Src* component source are discussed below, where for each parameter a brief description is provided, as well as its numeric type and the range of values that it can take on, respectively:

- **CarrierFreq** : The carrier frequency of the output DS-CDMA RF modulated mobile station signal, real, $(\text{ChipRate} / 2) < \text{Carrier_Freq}$.
- **ChipRate** : The direct sequence *chip rate* , real, $0 < \text{ChipRate}$. The source is intended to operate at a chip rate of 1.2288 Mcps.
- **Pav** : Relative average output power of the source, real, $0 \leq \text{Pav}$.
- **DelayTime** : The first chip generated by the source is applied to the baseband pulse shaping filters at *DelayTime* , real, $0 \leq \text{DelayTime}$.
- **GainRPC** : Relative gain of the reverse pilot channel, real, $0 \leq \text{GainRPC}$.
- **GainRFC** : Relative gain of the reverse fundamental channel, real, $0 \leq \text{GainRFC}$.
- **GainRSC1** : Relative gain of reverse supplemental channel 1, real, $0 \leq \text{GainRSC1}$.
- **GainRSC2** : Relative gain of reverse supplemental channel 2, real, $0 \leq \text{GainRSC2}$.
- **GainRDCC** : Relative gain of the reverse dedicated control channel, real, $0 \leq \text{GainRDCC}$.
- **ModSymRateRSC1** : The symbol rate of reverse supplemental channel 1, real, either 19.2, 38.4, 76.8, 153.6, 307.2, or 614.4 ksym/s.
- **ModSymRateRSC2** : The symbol rate of reverse supplemental channel 2, real, either 19.2, 38.4, 76.8, 153.6, or 307.2 ksym/s.

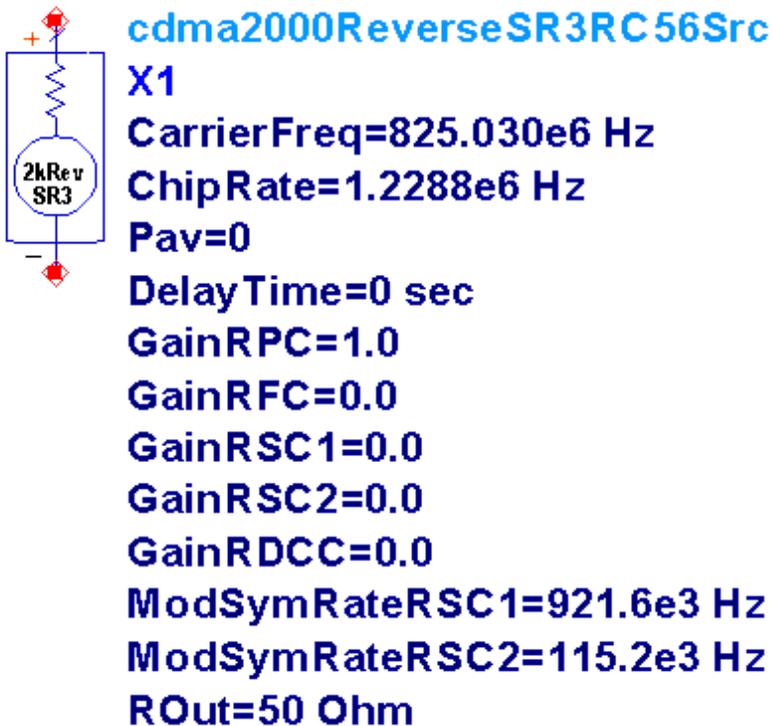
- **ROut** : The output resistance of the source, real, 0.01 *ROut*.

The *cdma2000ReverseSR1RC34Src* component only supports a simulation time step that is equal to four samples per chip. The simulation time step is limited in this manner because the coefficients of the baseband pulse shaping filters are hard coded. The pulse shaping filters are implemented using an FIR filter, with the coefficients being taken directly from the cdma2000 specification. [3](#)

The number of samples per chip depends upon the value of the *ChipRate* parameter and the simulation time-step specified in the *Envelope* simulation controller. Although the *ChipRate* parameter is included in the source's parameter list, it should remain fixed at its default value, i.e. the cdma2000 specified chip rate of 1.2288 Mcps. As a result of the chip rate being fixed, the number of samples per chip will depend only upon the simulation time-step. It is up to the designer to set the simulation time step so as to produce four samples per chip. The *TestRevSR1RC34* design provides an example of how to set the simulation time step.

cdma2000ReverseSR3RC56Src

The *cdma2000ReverseSR3RC56Src* component is shown in the following figure.



ADS cdma2000 Reverse Link Signal Sources for RC5 and RC6

The *cdma2000ReverseSR3RC56Src* component generates a continuously modulated single carrier DS-SS-SSMA mobile station signal using a chip rate of 3.6864 Mcps. Radio configurations 5 and 6 are very similar to radio configurations 3 and 4, with the primary difference between them being the system chip rate - moving from 1.2288 Mcps

to 3.6864 Mcps. Along with a higher chip rate comes higher supported data rates, as well as a wider signal. The manner in which the data of each reverse link physical channel is channelized, the resultant chips aggregated, and the complex spreading applied is the same for both spreading rates. [4](#)

As expected, the operation of the *cdma2000ReverseSR3RC56Src* component is essentially identical to that of the *cdma2000ReverseSR1RC34Src* component. The primary difference between the two sources is the chip rate and range of valid supplemental channel symbol rates. Much of the text describing the operation of the *cdma2000ReverseSR1RC34Src* component from [ADS cdma2000 Reverse Link Signal Source for RC1 and RC2](#) is repeated here.

Data Display

A simple example showing each reverse link source in an envelope simulation is provided in *TestRevSR1RC12*, *TestRevSR1RC34* and *TestRevSR3RC56*, where the name of each design corresponds to the reverse link source it tests. Each of the example simulations is associated with a Data Display file of the same name. The Data Display files contain a series of pages containing equations and plots for the analysis of results from simulations based upon the reverse link sources. The *TestRevSR1RC12* Data Display file contains 6 pages, while the *TestRevSR1RC34* and *TestRevSR3RC56* of files contain 5 pages. The 5 pages of the latter pair are each quite similar to a corresponding page in the former. The additional page in the *TestRevSR1RC12* file is entitled Equivalent BER. All 6 pages are discussed below under a separate heading that corresponds to the title of each page in the Data Display environment. Any differences between the five common pages across the three files are discussed.

In the case of the *TestRevSR1RC12* Data Display file, which process signals based upon RC1 and RC2, it is assumed that the reverse link signal transports a single physical channel only.

Variable Assignment

For the post-processing contained in the Data Display files to be carried out, certain results from the simulation domain are expected to be collected and subsequently assigned to particular variables in the Data Display domain. The assignment of data variables from the simulation domain to Data Display variables is done within the Variable Assignment page.

During the course of a simulation based upon a reverse link, the envelope of the signal generated by the reverse link source (the "reference envelope") and the envelope of the output signal from the device under test (the "distorted envelope") are expected to be collected. In the example simulation designs, the reference envelope is assigned to *VRefOut* and the distorted envelope to *VDistOut*. In addition, two other pieces of information are required from the simulation domain, namely, the chip rate and the time step. In the simulation examples both the chip rate and time step are collected and assigned to *chip_rate_out* and *tstep_out*, respectively.

Spectrums and ACPR

For discussion purposes the Spectrums and ACPR page can be divided into upper and lower halves. The upper half of the page displays the estimated power spectral density of the reference and distorted envelopes, calculated over the entire simulation period, and the lower half of the page contains ACPR calculations.

The estimated power spectral density of each envelope is calculated using the ADS *fs* function and is plotted in the two large plots. The *fs* function includes a parameter for changing the window that is applied to the time domain data prior to determining the estimated power spectral density.

The ACPR for a selected channel can be determined and displayed using the formulas and tables in the lower half of the page. The ACPR calculations employ the ADS *acpr_vr* function. The ACPR is determined for the channels defined by the *mainlimits*, *UpChlimits* and *LoChlimits* variables.

Power Calcs.

The primary aim of the Power Calcs. page is to determine the peak-to-average power ratio (PAR) and the complementary cumulative distribution function (CCDF) for either the reference or distorted envelope over a specified length of time.

The reference or distorted envelope is selected for processing by setting the variable *VEnv* equal to *VRef* or *VDist*, respectively. With the desired envelope selected, the power of the envelope over the entire simulation run-time is determined and plotted. The designer controls the start and end times of the time window over which the PAR and CCDF are calculated by adjusting the *Start* and *Stop* markers, with the time and power that each marker corresponds to being shown to the left of the "Envelope Power Versus Time" plot.

Based upon the time window specified by the *Start* and *Stop* markers, the average power and PAR of the desired envelope are calculated and displayed. Note that the absolute value ("linear") and dB value of the average power and PAR are displayed.

The CCDF of the desired envelope over the selected time window is determined and displayed in the lower half of the page. The CCDF results may look odd if the instantaneous power of the envelope drops to zero within the selected time window because the instantaneous-to-average power ratio at these points is negative infinity. Although the x-axis of the CCDF plot begins at -2 dB, the x-axis actually extends to the smallest instantaneous-to-average power ratio that is encountered over the span of the time window. Hence, if the smallest instantaneous-to-average power ratio is at negative infinity, the width of the bins of the CCDF will be very large, which produces the odd looking CCDF. This issue can be partially overcome by adding a small amount of power (say 1e-15) in the *Pout* calculation.

EVM, Constellation & Trajectory

The *EVM, Constellation & Trajectory* page is designed to extract and display the trajectory and constellation of the reference and distorted envelopes over the length of the simulation period. In addition, the error vector magnitude (EVM) and related statistics are calculated and displayed for:

- The distorted constellation relative to the reference constellation.
- The distorted constellation relative to an ideal constellation.
- The reference constellation relative to an ideal constellation.

The plots pertaining to the reference envelope have red points/traces, while those pertaining to the distorted envelope have blue points/traces.

The reference and distorted envelopes are processed by the *const_evm_oqpsk_ref_v2* function in the case of *RC1* and *RC2*, and by the *const_evm_hpsk_ref_v2* function for *RC3* through *RC6*. The reference and distorted envelopes are processed by the *const_evm ...* functions to extract the trajectory and constellation data of each envelope. In addition, using the constellation data, the EVM (and associated measures) of the distorted constellation relative to the reference constellation is also calculated by the functions.

The clock offset fine tunes the timing for the extraction of the chip samples from the reference envelope. The timing is intended to be adjusted by the user so that the EVM of the reference constellation relative to an ideal constellation is minimized. The timing is adjusted by moving the *ClockOffset* marker located within the small plot located immediately below the tables displaying the EVM results. Located within this same plot is the *PathOffset* marker. The *PathOffset* should be added to the path delay time (*kpathdelay*) so that the timing for the symbol samples extracted from the reference envelope can be adjusted by the user so that EVM of the distorted constellation relative to the reference constellation is also minimized.

The extracted trajectory data of the reference and distorted envelopes is plotted in the two upper most plots on the page. The *ktrajectorysymbols* variable located above the trajectory plots enables the designer to set the total number of chip periods traversed by the trajectory data that is plotted. Located immediately below the trajectory plots are the extracted constellations from the reference and distorted envelopes. The EVM and associated statistics generated by the *const_evm ...* functions is tabulated in the box entitled "Distorted constellation relative to the Reference constellation".

To generate the results tabulated in the "Reference constellation relative to an Ideal constellation" and "Distorted constellation relative to an Ideal constellation" boxes the reference and distorted envelope constellation data produced by the *const_evm ...* functions is measured against an ideal constellation.

The following should be noted regarding all of the EVM and associated statistics in the three sets of tabulated results:

- The results are calculated over all constellation points for which constellation data exists for the two constellations that form the basis of the EVM calculations;
- The EVM results are calculated after adjusting for any phase difference, dc offset and

gain/compression difference between the two constellations that form the basis of the EVM calculations; and,

- The *RMS EVM (%)*, *Peak EVM (%)* and *Mag Error (%)* are all given relative to the average normalized magnitude of the ideal constellation points, in the case of the tabulated data of the leftmost and centre boxes, and relative to the average normalized magnitude of the reference constellation points, in the case of the tabulated data of the rightmost box.

The *TestRevSR1RC12* Data Display file includes a series of equations at the bottom of the page that are applicable to the Equivalent BER page and which does not appear in the *TestRevSR1RC34* and *TestRevSR3RC56* files.

Envelope Plots

The Envelope Plots page contains 4 plots that are quite straight forward - the amplitude of the in-phase and quadrature components of the reference and distorted envelopes versus time.

Equivalent BER

Based upon the EVM figure of the reference and distorted constellations relative to the ideal constellation, an equivalent BER can be derived using the analytic result for OPQSK in an additive white Gaussian noise channel. The complete BER curve from the analytic result is plotted along with that derived for the reference and distorted constellations.

Forward Link

Simulation Domain

All of the forward link sources, *cdma2000ForwardSR1RC12Src* , *cdma2000ForwardSR1RC12Src* and *cdma2000ForwardSR1RC12Src* , generate a continuously modulated DS-CDMA signal that transports a *single* forward traffic channel.

cdma2000ForwardSR1RC12Src

The *cdma2000ForwardSR1RC12Src* source is shown in the following figure.



cdma2000ForwardSR1RC12Src

X1

CarrierFreq=880.0e6 Hz

ChipRate=1.2288e6 Hz

Pav=0

DelayTime=0 sec

WalshCode=1

ADS cdma2000 Forward Link Signal Source for RC1 and RC2

The parameters of the *cdma2000ForwardSR1RC12Src* component are discussed below, where for each parameter a brief description is provided, as well as its numeric type and the range of values that it can take on, respectively:

- **CarrierFreq** : The carrier frequency of the output DS-CDMA RF modulated base station signal, real, $(ChipRate / 2) < Carrier_Freq$.
- **ChipRate** : The direct sequence chip rate, real, $0 < ChipRate$. The source is intended to operate at a chip rate of 1.2288 Mcps.
- **Pav** : Relative average output power of the source, real, $0 Pav$.
- **DelayTime** : The first chip generated by the source is applied to the baseband pulse shaping filters at *DelayTime* , real, $0 DelayTime$.
- **WalshCode** : Index of the Walsh code/function employed for orthogonal spreading, $1 WalshCode 63$.

The *cdma2000ForwardSR1RC12Src* component generates a continuously modulated single carrier DS-CDMA signal having a chip rate of 1.2288 Mcps that is equivalent to the transport of a forward traffic channel using RC1 or RC2. The forward traffic channel structure is shown in Figures 3.1.3.1.1.1-13 and 3.1.3.1.1.1-14 of [1]. The spreading process and quadrature modulator for RC1 and RC2 is shown in Figure 3.1.3.1.1.1-20 of [1].

cdma2000ForwardSR1RC35Src

The *cdma2000ForwardSR1RC35Src* component is shown in the following figure.



cdma2000ForwardSR1RC35Src

X1

CarrierFreq=880.0e6 Hz

ChipRate=1.2288e6 Hz

Pav=0

DelayTime=0 sec

WalshLength=64

WalshCode=1

QPSKRate=9.6e6 Hz

ADS cdma2000 Forward Link Signal Source for RC3 through RC5

The parameters of the *cdma2000ForwardSR1RC35Src* component are discussed below, where for each parameter a brief description is provided, as well as its numeric type and the range of values that it can take on, respectively:

- **CarrierFreq** : The carrier frequency of the output DS-CDMA RF modulated base station signal, real, $(ChipRate / 2) < Carrier_Freq$.
- **ChipRate** : The direct sequence chip rate, real, $0 < ChipRate$. The source is intended to operate at a chip rate of 1.2288 Mcps.
- **Pav** : Relative average output power of the source, real, $0 Pav$.
- **DelayTime** : The first chip generated by the source is applied to the baseband pulse shaping filters at *DelayTime* , real, $0 DelayTime$.
- **WalshLength** : The length of the family of Walsh functions/codes from which to select the orthogonal spreading Walsh function/code, integer, $WalshLength = 64$ or 128 .
- **WalshCode** : Index of the Walsh code/function employed for orthogonal spreading, $1 WalshCode WalshLength$ 1.
- **QPSKRate** : The modulation symbol rate of the modulator, real, $QPSKRate = 2.4, 4.8, 9.6, 19.2, 38.4, 76.8, 153.6$ or 307.2 ksym/s.

The *cdma2000ForwardSR1RC35Src* component generates a continuously modulated single carrier DS-CDMA signal having a chip rate of 1.2288 Mcps that is equivalent to the transport of a forward traffic channel using RC3, RC4 or RC4. The forward traffic channel structure is shown in Figures 3.1.3.1.1.1-10 through 3.1.3.1.1.1-12, and 3.1.3.1.1.1-15 through 3.1.3.1.1.1-18 of [1]. The spreading process and quadrature modulator for RC3 through RC5 is shown in Figure 3.1.3.1.1.1-20 of [1].

cdma2000ForwardSR3RC69Src

The *cdma2000ForwardSR3RC69Src* component is shown in the following figure.



cdma2000ForwardSR3RC69Src

X1

CarrierFreq=880.0e6 Hz

CarrierDelta=1.25e6 Hz

ChipRate=1.2288e6 Hz

GainLower=1.0

GainCentre=1.0

GainUpper=1.0

Pav=0

WalshLength=128

WalshCode=1

QPSKRate=28.8e3 Hz

DelayTime=0.0 sec

ADS cdma2000 Forward Link Signal Source for RC6 through RC9.

The parameters of the *cdma2000ForwardSR3RC69Src* component are discussed below, where for each parameter a brief description is provided, as well as its numeric type and the range of values that it can take on, respectively:

- **CarrierFreq** : The centre frequency of the middle channel of the output multicarrier DS-CDMA RF modulated base station signal, real, $(ChipRate / 2) < Carrier_Freq$.
- **CarrierDelta** : The carrier frequency spacing of the output multicarrier DS-CDMA RF modulated base station signal, real, $(ChipRate / 2) < CarrierDelta$.
- **ChipRate** : The direct sequence chip rate, real, $0 < ChipRate$. The source is intended to operate at a chip rate of 1.2288 Mcps.
- **GainLower** : Relative gain of the lower carrier, real, $0 < GainLower$.
- **GainCentre** : Relative gain of the centre carrier, real, $0 < GainCentre$.
- **GainUpper** : Relative gain of the upper carrier, real, $0 < GainUpper$.
- **Pav** : Relative average output power of the source, real, $0 < Pav$.
- **DelayTime** : The first chip of each carrier that is generated is applied to the baseband pulse shaping filters at *DelayTime* , real, $0 < DelayTime$.
- **WalshLength** : The length of the family of Walsh functions/codes from which to select the orthogonal spreading Walsh function/code, integer, $WalshLength = 128$ or 256 .
- **WalshCode** : Index of the Walsh code/function employed for orthogonal spreading, integer, $1 < WalshCode < WalshLength$.
- **QPSKRate** : The aggregate modulation symbol rate of the modulator, real, $QPSKRate = 3.6, 7.2, 14.4, 28.8, 57.6, 115.2, 230.4, 460.8, \text{ or } 921.6$ ksym/s.

The *cdma2000ForwardSR3RC69Src* component generates a continuously modulated multicarrier DS-CDMA signal that is equivalent to the transport of a forward traffic channel using RC6, RC7, RC8 or RC9. Each carrier has a chip rate of 1.2288 Mcps. The forward traffic channel structure is shown in Figures 3.1.3.1.1.2-7 through 3.1.3.1.1.2-15 of [1]. The spreading process and quadrature modulators for forming the multicarrier signal is shown in Figure 3.1.3.1.1.2-17 of [1].

Data Display

A simple example showing the forward link sources in an envelope simulation is provided in *TestFwdSR1RC12* , *TestFwdSR1RC35* , and *TestFwdSR3RC69* where the name of each design corresponds to the forward link source it tests. The three example simulations are each associated with a Data Display file of the same name. The pages of the *TestFwdSR1RC12* , *TestFwdSR1RC35* , and *TestFwdSR3RC69* files are essentially the same as those pages with a corresponding name in the Data Display files of the reverse link.

Variable Assignment

In the case of *TestFwdSR3RC69*, the multicarrier DS-CDMA signal of RC6 through RC9 demands that complex impulse chip stream that is produced following quadrature spreading be collected for each carrier. The complex impulse symbol streams are assigned to *kideal1_out*, *kideal2_out* and *kideal3_out* in the simulation domain. In addition, the channel spacing is collected and assigned to *ChannelSpacing_out*. These simulation domain variables are assigned to the Data Display variables *lower_symbols*, *centre_symbols*, *upper_symbols* and *frequency_spacing*, respectively. Refer to [Variable Assignment](#) for more information.

Spectrums and ACPR

See [Spectrums and ACPR](#).

Power Calcs.

See [Power Calcs.](#)

EVM, Constellation & Trajectory

See [EVM, Constellation & Trajectory](#).

TestFwdSR1RC12

TestFwdSR1RC12

With respect to extracting the reference and distorted constellations and trajectory data, as well as calculating their relative EVM, there is a difference between the *TestFwdSR1RC12* and *TestRevSR1RC12* files. The difference arises because RC1 and RC2 use an OQPSK signal in the reverse link but a QPSK signal in the forward link.

TestFwdSR3RC69

The multicarrier DS-CDMA signal of RC6 through RC9 is processed by selecting one of the carriers to analyze and isolating that carrier through filtering if necessary. The additional processing that is required is isolated in the lower left-hand portion of the page and is controlled through the *kchannel* and *kfilter* variables. The *kchannel* variable is set by the designer to either "Lower", "Upper", or "Centre" to select which of the three carriers is to be analyzed. The *kfilter* variable is set by the designer to either "Yes" or "No" to control whether the selected carrier of the reference and distorted envelopes is to be filtered prior to extracting the constellation and trajectory data, and performing the EVM calculations.

The first step is to shift the frequency spectrum of the reference and distorted envelopes to centre the carrier to be analyzed about 0 Hz. The spectrum shift of both envelopes is performed using the *kmm_spectrum_shift* function.

After shifting the frequency spectrums, if desired, a filter is applied to the resultant envelopes to isolate the selected carrier. The filter is applied using the *kmm_filt* function, with the filter's coefficients being drawn from the series of filter coefficients that are compiled within the *kmm_filtcoef* function. The *kmm_filt* function performs a convolution of the two swept variables that are passed to it.

Following shifting of the frequency spectrums and filtering, if desired, the reference and distorted envelope of the carrier to be analyzed are assigned to the *VRef_s* and *VDist_s* variables, respectively. Post processing to extract the reference and distorted constellations and trajectory data, as well as calculate EVM results, is performed on *VRef_s* and *VDist_s* in the same manner as that described in [EVM, Constellation & Trajectory](#). In addition, the estimated power spectral density of *VRef_s* and *VDist_s* is calculated using the ADS *fs* function and displayed in the two additional plots on the far right hand side of the page.

Envelope Plots

See [Envelope Plots](#).

Equivalent BER

See [Equivalent BER](#).

WCDMA (3GPP)

ADS Designs ()

- WCDMADownlink_Src
- WCDMAUplink_DPCCH_DPDCH_Src
- Test_Downlink
- Test_Uplink

Data Display Files (.dds)

- Test_Downlink
- Test_Uplink

Specifications

The designs are based upon Release 5 of the following WCDMA (FDD) specifications:

- [1] 3GPP TS 25.101 V5.2.0 (2002-03) "UE Radio Transmission and Reception (FDD)"
- [2] 3GPP TS 25.104 V5.2.0 (2002-03) "BS Radio Transmission and Reception (FDD)"
- [3] 3GPP TS 25.201 V5.0.0 (2001-12) "Physical Layer - General Description"
- [4] 3GPP TS 25.211 V5.0.0 (2002-03) "Physical Channels and Mapping of Transport Channels Onto Physical Channels"
- [5] 3GPP TS 25.212 V5.0.0 (2002-03) "Multiplexing and Channel Coding (FDD)"
- [6] 3GPP TS 25.213 V5.0.0 (2002-03) "Spreading and Modulation (FDD)"
- [7] 3GPP TS 25.214 V5.0.0 (2002-03) "Physical Layer Procedures (FDD)"

All specifications are available from: <http://www.3gpp.org>

Uplink

For uplink transmission five different physical channels are defined. The five physical channels are either of type dedicated or common. For reference, the uplink physical channels are set out in the following table. The *italicized* physical channels are *NOT* supported by the ADS uplink source.

Uplink Physical Channels

Type	Channel	Acronym	Section of [4]
Dedicated	Dedicated Physical Data Channel	DPDCH	5.2.1
	Dedicated Physical Control Channel	DPCCH	5.2.1
	Dedicated Control Channel associated with the High Speed Downlink Shared Channel \ (HS-DSCH)	HS-DPCCH	5.2.1
Common	<i>Physical Random Access Channel</i>	<i>PRACH</i>	5.2.2.1
	<i>Physical Common Packet Channel</i>	<i>PCPCH</i>	5.2.2.2

Downlink

Like the uplink, a series of physical channels are defined for the downlink. For reference, the downlink physical channels are set out in the following table. The *italicized* physical channels are *NOT* supported by the ADS downlink source.

Downlink Physical Channels

Type	Channel	Acronym	Section of [4]
Dedicated	Dedicated Physical Channel	DPCH	5.3.2
Common	Common Pilot Channel	CPICH	5.3.3.1
	Primary Common Control Physical Channel	P-CCPCH	5.3.3.3
	Secondary Common Control Physical Channel	S-CCPCH	5.3.3.4
	Synchronization Channel	SCH	5.3.3.5
	<i>Physical Downlink Shared Channel</i>	<i>PDSCH</i>	5.3.3.6
	<i>Acquisition Indicator Channel</i>	<i>AICH</i>	5.3.3.7
	<i>Common Packet Channel \ (CPCH) Access Preamble Acquisition Indicator Channel</i>	<i>AP-AICH</i>	5.3.3.8
	<i>Common Packet Channel \ (CPCH) Collision Detection/Channel Assignment Indicator Channel</i>	<i>CD/CA - ICH</i>	5.3.3.9
	Paging Indicator Channel	PICH	5.3.3.10
	<i>_Common Packet Channel \ (CPCH) Status Indicator Channel_</i>	<i>CSICH</i>	5.3.3.11
	<i>Shared Control Channel</i>	<i>HS-SCCH</i>	5.3.3.12
<i>High Speed Physical Downlink Shared Channel</i>	<i>HS-PDSCH</i>	5.3.3.13	

Usage and Operation

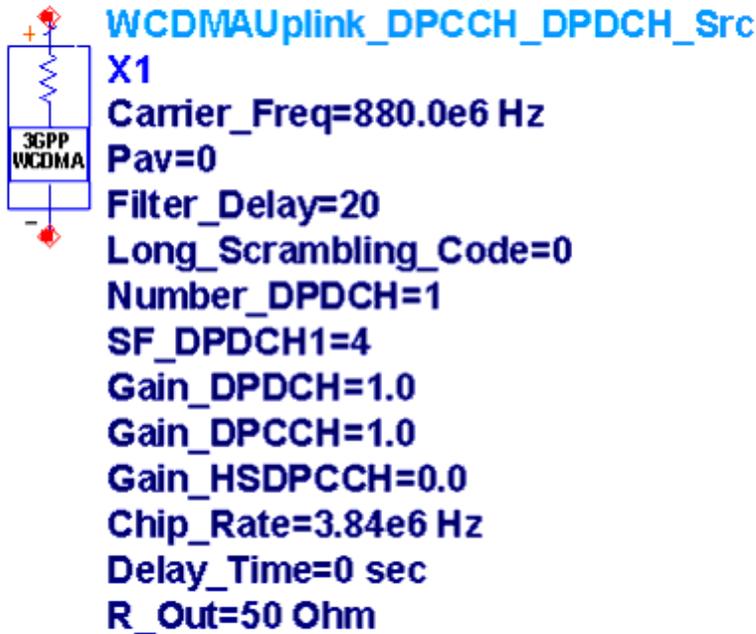
A WCDMA uplink source and a downlink source have been developed - *WCDMAUplink_DPDCH_DPCCH_Src* and *WCDMADownlink_Src*, respectively. The design files of the source components and their associated subcomponents are set out in [ADS Designs \(\)](#). The file list in ADS Designs () also includes two *Test..._* designs. Each *Test..._* design contains a simple simulation example of one of the sources.

Uplink

Simulation Domain

Component Parameters

The *WCDMAUplink_DPCCH_DPDCH_Src* component is shown in the following figure.



ADS WCDMA (FDD) Uplink Modulated Signal Source

The uplink source generates a continuously modulated single carrier WCDMA user signal. The *WCDMAUplink_DPCCH_DPDCH_Src* parameters are discussed below, where for each parameter a brief description is provided, as well as its numeric type and the range of values that it can take on, respectively.

- **Carrier_Freq** : The carrier frequency of the output composite WCDMA RF modulated user signal, real, $(\text{ChipRate} / 2) < \text{Carrier_Freq}$.
- **Pav** : Relative average output power of the source, real, $0 < \text{Pav}$.
- **Filter_Delay**: Delay (in chips) of the baseband inphase and quadrature square root raised cosine pulse shaping filters, integer, $0 < \text{Filter_Delay}$.
- **Long_Scrambling_Code** : The seed value for the long scrambling code, integer, $0 < \text{Long_Scrambling_Code} < 16777215$ (224 - 1).
- **Number_DPDCH** : Number of dedicated physical data channels to transmit, integer, $0 < \text{Number_DPDCH} < 6$.
- **SF_DPDCH1** : If *Number_DPDCH* = 1, the spreading factor of the single dedicated physical data channel that is transmitted, integer, 2k, where k = 2, 3, 4, ... 8. The

value of *SF_DPDCH1* is ignored if *Number_DPDCH* 1.

- **Gain_DPDCH** : Relative gain of the dedicated physical data channel(s), real, 0 *Gain_DPDCH* .
- **Gain_DPCCH** : Relative gain of the dedicated physical control channel, real, 0 *Gain_DPDCCH* .
- **Gain_HSPDCCH** : Relative gain of the dedicated physical control channel associated with the high speed downlink shared channel (HS-DSCH), real, 0 *Gain_HSPDCCH*.
- **Chip_Rate** : The direct sequence chip rate, real, $0 < \text{Chip_Rate}$. The source is intended to operate at a chip rate of 3.84 Mcps.
- **Delay_Time** : The first chip generated by the source is applied to the baseband pulse shaping filters at *Delay_Time* , real, 0 *Delay_Time*.
- **R_Out** : The output resistance of the source, real, 0.01 *_R_Out_* .

The WCDMA uplink source supports different simulation step sizes, that is, the number of samples per chip is not fixed. Support for a variable number of samples per chip enables the simulation bandwidth to be adjusted and thereby facilitates a balance between the competing priorities of simulation accuracy and execution speed.

The number of samples per chip depends upon the value of the *Chip_Rate* parameter and the simulation time-step specified in the *Envelope* simulation controller. Although the *Chip_Rate* parameter is included in the uplink source's parameter list, it should remain fixed at its default value, equal to the WCDMA specified chip rate of 3.84 Mcps. As a result of being fixed, the number of samples per chip will depend only upon the simulation time-step. It is up to the designer to set the simulation time step so as to produce the desired number of samples per chip. The *Test_Uplink* design provides an example of how to set the simulation time step.

Limitations

The uplink source does not implement compressed mode signaling, nor the short scrambling code, an alternative scrambling code to the long scrambling code. In addition, random data is generated for transport by the DPCCH, thus, no pilot, transport format indicator, feedback information or transmit power control bits are specifically generated and inserted into the DPCCH data symbol stream. Similarly, random data is generated for transport by the HS-DPCCH, thus no hybrid-ARQ acknowledgement or channel quality indicator bits are specifically generated.

Data Display

A simple example showing the uplink source in an envelope simulation is provided in *Test_Uplink*. The example simulation is associated with a Data Display file of the same. The Data Display file contains a series of pages containing equations and plots for the analysis of results from simulations based upon the uplink source. The *Test_Uplink* file contains 5 pages.

Variable Assignment

For the post-processing contained in the Data Display files to be carried out, certain results from the simulation domain are expected to be collected and subsequently assigned to particular variables in the Data Display files. This assignment of data variables from the simulation domain to Data Display variables is carried in the Variable Assignment page.

During an uplink source simulation the envelope of the signal generated by the uplink source (the "reference envelope") and the envelope of the output signal from the device under test (the "distorted envelope") are expected to be collected during simulation. In the example simulation design, the reference envelope is assigned to *VRefOut* and the distorted envelope to *VDistOut*. It is also necessary to collect the complex impulse chip stream that is output from the HPSK encoder and fed to the baseband pulse shaping filters. The complex impulse symbol stream is collected and assigned to *ImpOut*. In addition, two other pieces of information are required from the simulation, namely, the chip rate and the time step. In the simulation example both the chip rate and time step are collected and assigned to *chip_rate_out* and *tstep_out*, respectively.

Spectrums and ACPR

The Spectrums and ACPR page can be roughly divided into upper and lower halves. The upper half of the page displays the estimated power spectral density of the reference and distorted envelopes, calculated over entire period of simulation, and the lower half of the page contains ACPR calculations.

The estimated power spectral density of each envelope is calculated using the ADS *fs* function and is plotted in the two large plots. The *fs* function includes a parameter for changing the window that is applied to the time domain data prior to determining the estimated power spectral density.

The ACPR for a selected channel can be determined and displayed using the formulas and tables in the lower half of the page. The ACPR calculations employ the ADS *acpr_vr* function. The ACPR is determined for the channels defined by the *mainlimits*, *UpChlimits* and *LoChlimits* variables.

Power Calcs.

The primary aim of the Power Calcs. page is to determine the peak-to-average power ratio (PAR) and the complementary cumulative distribution function (CCDF) for either the reference or distorted envelope over a specified length of time.

The reference or distorted envelope is selected for processing by setting the variable *VEnv* equal to *VRef* or *VDist*, respectively. With the desired envelope selected, the power of the envelope over the entire simulation run-time is determined and plotted. The designer controls the start and end times of the time window over which the PAR and CCDF are calculated by adjusting the *Start* and *Stop* markers, with the time and power that each

marker corresponds to being shown to the left of the "Envelope Power Versus Time" plot.

Based upon the time window specified by the *Start* and *Stop* markers, the average power and PAR of the desired envelope are calculated and displayed. Note that the absolute value ("linear") and dB value of the average power and PAR are displayed.

The CCDF of the desired envelope over the selected time window is determined and displayed in the lower half of the page.

EVM, Constellation & Trajectory

The EVM, Constellation & Trajectory page is designed to extract and display the trajectory and constellation of the reference and distorted envelopes over the length of the simulation period. In addition, the error vector magnitude (EVM) and related statistics are calculated and displayed for:

- The distorted constellation relative to the reference constellation.
- The distorted constellation relative to an ideal constellation.
- The reference constellation relative to an ideal constellation.

The plots pertaining to the reference envelope have red points/traces, while those pertaining to the distorted envelope have blue points/traces.

The clock offset fine tunes the timing for the extraction of the chip samples from the reference envelope. The timing is intended to be adjusted by the user so that the EVM of the reference constellation relative to an ideal constellation is minimized. The timing is adjusted by moving the *ClockOffset* marker located within the small plot located immediately below the tables displaying the EVM results. Located within this same plot is the *PathOffset* marker. The *PathOffset* should be added to the path delay time (*kpathdelay*) so that the timing for the symbol samples extracted from the reference envelope can be adjusted by the user so that EVM of the distorted constellation relative to the reference constellation is also minimized.

The extracted trajectory data of the reference and distorted envelopes is plotted in the two upper most plots on the page. The *ktrajectorysymbols* variable located above the trajectory plots enables the designer to set the total number of chip periods traversed by the trajectory data that is plotted. Located immediately below the trajectory plots are the extracted constellations from the reference and distorted envelopes. The EVM and associated statistics generated by the *const_evm_hpsk_ref_v2* function is tabulated in the box entitled "Distorted constellation relative to the Reference constellation".

To generate the results tabulated in the "Reference constellation relative to an Ideal constellation" and "Distorted constellation relative to an Ideal constellation" boxes the reference and distorted envelope constellation data produced by the *const_evm_hpsk_ref_v2* function is measured against an ideal constellation. The ideal HPSK constellation will vary in shape and size according to the number of active uplink channels and their relative power. As a result of the variety of HPSK constellations that may be encountered, it is impractical to derive the ideal constellation from the

reference and distorted constellations through slicing. As an alternative to deriving the ideal constellation from the reference and distorted envelopes, during the course of a simulation the ideal constellation points that are generated by the uplink source are collected and stored in the dataset. Recall that within the Variable Assignment page, the complex signal that drives the pulse shaping filters was collected and stored in the dataset, *ImpOut*, and was assigned to the Data Display variable *bb_symbol_impulse_stream*.

Before the desired EVM calculations can be carried out, the ideal constellation must be extracted from the impulse symbol stream. The extraction of the ideal symbols is performed manually by adjusting the sweep indexing scheme used in the equation that assigns data to the *ideal_syms* variable. The impulse symbol stream has the following properties:

- It's collected over the entire duration of the simulation;
- The symbol impulses (ideal constellation points) are spaced apart from one another by the number of samples per chip used for simulation.

With the aid of the *ksamplesperchip* variable, set to the number of samples per chip, and the tables listing the data of the *bb_symbol_impulse_stream*, *ideal_syms* and *RefOut_const* (extracted constellation points of the reference envelope) variables that are located at the bottom of the page, the limits of the indexing scheme can be adjusted to extract the ideal constellation. The first ideal constellation point extracted should reflect the delay introduced by the *ktransientduration* variable that is part of the call to the *const_evm_hpsk_ref_v2* function.

The EVM and associated statistics of the reference and distorted envelope constellations relative to the ideal constellation are calculated using the *kmm_evm_hpsk_v2* function. The resulting EVM and associated statistics are tabulated in the leftmost and centre boxes located in the upper part of the page.

The following should be noted regarding all of the EVM and associated statistics in the three sets of tabulated results:

- the results are calculated over all constellation points for which constellation data exists for the two constellations that form the basis of the EVM calculations;
- the EVM results are calculated after adjusting for any phase difference, dc offset and gain/compression difference between the two constellations that form the basis of the EVM calculations; and,
- the *RMS EVM (%)* , *Peak EVM (%)* and *Mag Error (%)* are all given relative to the average normalized magnitude of the ideal constellation points, in the case of the tabulated data of the leftmost and centre boxes, and relative to the average normalized magnitude of the reference constellation points, in the case of the tabulated data of the rightmost box.

Envelope Plots

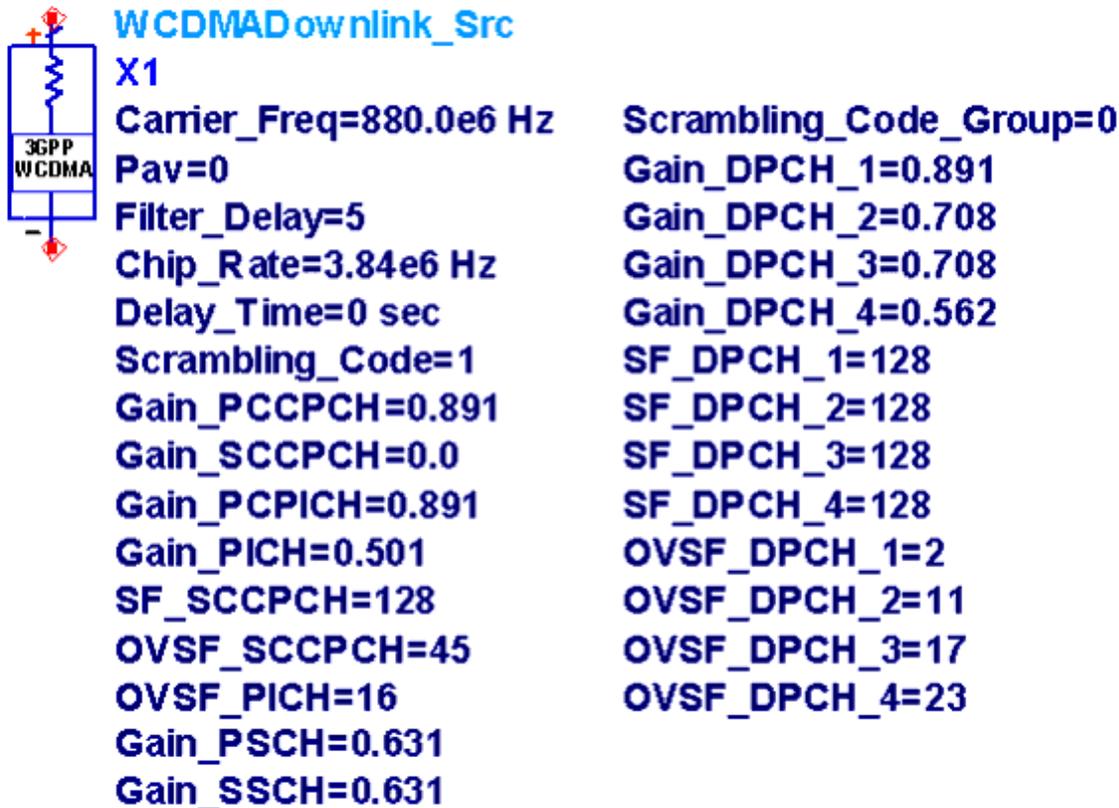
The Envelope Plots page contains 4 plots that are quite straight forward - the amplitude of

the in-phase and quadrature components of the reference and distorted envelopes versus time.

Downlink

Simulation Domain

The *WCDMADownlink_Src* component is shown in the following figure. Not all of the parameters of the *WCDMADownlink_Src* component are shown.



ADS WCDMA (FDD) Downlink Modulated Signal Source

The *WCDMADownlink_Src* component generates a continuously modulated single carrier multiuser WCDMA base station signal.

Component Parameters

The parameters of the *WCDMADownlink_Src* component are discussed below, where for each parameter a brief description is provided, as well as its numeric type and the range of values that it can take on, respectively.

- **Carrier_Freq** : The carrier frequency of the output composite RF modulated WCDMA base station signal, real, $(ChipRate / 2) < Carrier_Freq$.
- **Pav** : Relative average output power of the source, real, $0 < Pav$.
- **Filter_Delay** : Delay (in chips) of the baseband inphase and quadrature square root raised cosine pulse shaping filters, integer, $0 < Filter_Delay$.
- **Chip_Rate** : The direct sequence chip rate, real, $0 < Chip_Rate$. The source is intended to operate at a chip rate of 3.84 Mcps.
- **Delay_Time** : The first chip generated is applied to the baseband pulse shaping filters at $Delay_Time$, real, $0 < Delay_Time$.
- **Scrambling_Code** : The seed value for the scrambling code generator, integer, $0 < Scrambling_Code < 262143$ (218 - 1).
- **Scrambling_Code_Group** : The index of the scrambling code group for the secondary synchronization channel, integer, $0 < Scrambling_Code_Group < 63$.
- **Gain_PCCPCH** : Relative gain of the primary common control physical channel, real, $0 < Gain_PCCPCH$.
- **Gain_SCCPCH** : Relative gain of the secondary common control physical channel, real, $0 < Gain_SCCPCH$.
- **Gain_PCPICH** : Relative gain of the primary common pilot channel, real, $0 < Gain_PCPICH$.
- **Gain_PICH** : Relative gain of the paging indicator channel, real, $0 < Gain_PICH$.
- **SF_SCCPCH** : The spreading factor of the secondary common control physical channel, integer, $2k$, where $k = 2, 3, 4, \dots 8$.
- **OVSF_SCCPCH** : The index of the OVSF code employed for the secondary common control physical channel, integer, $0 < OVSF_SCCPCH < (SF_SCCPCH - 1)$.
- **OVSF_PICH** : The index of the OVSF code employed for the paging indicator channel, integer, $0 < OVSF_PICH < 255$.
- **Gain_PSCH** : Relative gain of the primary synchronization channel, real, $0 < Gain_PSCH$.
- **Gain_SSCH** : Relative gain of the secondary synchronization channel, real, $0 < Gain_SSCH$.
- **Gain_DPCH_x** : Relative gain of dedicated physical channels one through sixteen, real, $0 < Gain_DPCH_x$.
- **SF_DPCH_x** : The spreading factor of dedicated physical channels one through sixteen, integer, $2k$, where $k = 2, 3, 4, \dots 9$.
- **OVSF_DPCH_x** : The index of the OVSF code employed for dedicated physical channels one through sixteen, integer, $0 < OVSF_DPCH_x < (SF_DPCH_x - 1)$.
- **R_Out** : The output resistance of the source, real, $0.01 < R_Out$.
- The *WCDMADownlink_Src* component supports different simulation step sizes, i.e. the number of samples per chip is not fixed. Support for a variable number of samples per chip allows for the simulation bandwidth to be adjusted and thereby facilitates a balance to be achieved between the competing priorities of simulation accuracy and execution time.
- The number of samples per chip depends upon the value of the *Chip_Rate* parameter and the simulation time-step specified in the *Envelope* simulation controller. Although the *Chip_Rate* parameter is included in the downlink source's parameter list, it should remain fixed at its default value - 3.84 Mcps, equal to the WCDMA specified chip rate. As a result of being fixed, the number of samples per chip will only depend upon the time step of the simulation. It is up to the designer to set the simulation time step to result in the desired number of samples per chip. The *Test_Downlink* design provides

an example of how to set the simulation time step.

Limitations

- Beyond the limitations described above, the *WCDMADownlink_Src* component does not implement:
 - compressed mode signaling [5](#)
 - discontinuous transmission [6](#),
 - insertion of transport format control and pilot bits [7](#),
 - insertion of DTX bits in the place of Layer 1 control information in the case of multicode transmission [8](#), and
 - space time diversity [9](#).

Data Display

A simple example showing the downlink source in an envelope simulation is provided in *Test_Downlink* . All 5 pages of the *Test_Downlink* file are essentially identical to its corresponding page in the uplink Data Display file. Only the differences between the downlink and uplink pages are discussed below.

Variable Assignment

See [Variable Assignment](#).

Spectrums and ACPR

See [Spectrums and ACPR](#).

Power Calcs.

See [Power Calcs](#).

The PAR of a base station signal should be significantly greater than that of a mobile signal since more physical channels should be active.

EVM, Constellation & Trajectory

See [EVM, Constellation & Trajectory](#). The constellation and trajectory data becomes increasingly difficult to interpret visually as more physical channels are activated.

Envelope Plots

See [Envelope Plots](#).

GSM/EDGE (3GPP)

ADS Designs ()

- GSM_EDGE_Downlink_8PSK_Src
- GSM_EDGE_Downlink_GMSK_Src
- GSM_EDGE_Uplink_8PSK_Src
- GSM_EDGE_Uplink_GMSK_Src
- Test_Downlink_8PSK
- Test_Downlink_GMSK
- Test_Uplink_8PSK
- Test_Uplink_GMSK

Data Display Files (.dds)

- Test_Downlink_8PSK
- Test_Downlink_GMSK
- Test_Uplink_8PSK
- Test_Uplink_GMSK

Specifications

The designs are based upon Release 5 of the following GSM/EDGE specifications:

[1] 3GPP TS 45.001 V5.3.0 (2002-04) "Physical Layer On the Radio Path"

[2] 3GPP TS 45.002 V5.5.0 (2002-04) "Multiplexing and Multiple Access On the Radio Path"

[3] 3GPP TS 45.003 V5.5.0 (2002-04) "Channel Coding"

[4] 3GPP TS 45.004 V5.0.0 (2001-11) "Modulation" (Note: this document states that it is from Release 4, however, its version indicates that it is from Release 5, and the 3GPP document status table states that it is from Release 5)

[5] 3GPP TS 45.005 V5.3.0 (2002-04) "Radio Transmission and Reception"

A helpful document for interpreting the above specifications is:

[6] 3GPP TR 21.995 V5.3.0 (2002-03) "Vocabulary for 3GPP Specifications"

All specifications are available from: <http://www.3gpp.org>

Usage and Operation

Four different GSM/EDGE sources have been developed - two for the downlink and two for the uplink - all denoted by the suffix `_Src` in the above list of design files. Each pair of sources has one source that outputs a GMSK modulated signal and one source that outputs an 8-PSK modulated signal. The four `Test_` designs contain a simple simulation example for each of the sources.

Uplink

Simulation Domain

The `GSM_EDGE_Uplink_GMSK_Src` and `GSM_EDGE_Uplink_8PSK_Src` components are shown in the following images.



GSM_EDGE_Uplink_GMSK_Src

X2

Carrier_Freq=880.0e6 Hz

Pav=0

Slot_0=1

Slot_1=0

Slot_2=0

Slot_3=0

Slot_4=0

Slot_5=0

Slot_6=0

Slot_7=0

TSC_Num=0

Symbol_Rate=270.83333333e3 Hz

Frequency_Hop="No"

Mobile_Allocation=1

Channel_Spacing=1

Hopping_Sequence_Num=1

Mobile_Allocation_Offset=0

Delay_Time=0 sec

R_Out=50.0 Ohm

ADS GSM/EDGE Uplink GMSK Modulated Source

**GSM_EDGE_Uplink_8PSK_Src****X2****Carrier_Freq=880.0e6 Hz****Pav=0****Slot_0=1****Slot_1=0****Slot_2=0****Slot_3=0****Slot_4=0****Slot_5=0****Slot_6=0****Slot_7=0****TSC_Num=0****Symbol_Rate=270.83333333e3 Hz****Frequency_Hop="No"****Mobile_Allocation=1****Channel_Spacing=1****Hopping_Sequence_Num=1****Mobile_Allocation_Offset=0****Delay_Time=0 sec****R_Out=50.0 Ohm****ADS GSM/EDGE Uplink 8-PSK Modulated Source**

The *GSM_EDGE_Uplink_GMSK_Src* and *GSM_EDGE_Uplink_8PSK_Src* components (the "uplink sources") both generate a *TDMA GMSK/8-PSK MS* modulated signal. The operation of the uplink sources is virtually identical. The primary difference between the two is the modulation format of the output signal. Using the parameters of each component, the user is able to control the number of slots utilized for uplink transmission, as well as the frequency hopping scheme followed by the output signal.

Each of the uplink sources is controlled by an identical set of parameters. The parameters are discussed below, where for each parameter a brief description is provided, as well as its numeric type and the range of values that it can take on, respectively.

- **Carrier_Freq:** The centre frequency of the power spectral density of the output modulated signal (if frequency hopping is not used, *Carrier_Freq* is the carrier frequency of the modulated signal, real, $0 < \text{Carrier_Freq}$).
- **Pav:** Relative average output power of the source, real, $0 \leq \text{Pav}$.
- **Slot_x:** Specifies what type of traffic, if any, is to be transmitted within slot *x*, where $x = 0$ to 7 . A value of 1 results in full-rate traffic, 2 results in half-rate traffic on subchannel 0 , 3 results in half-rate traffic on subchannel 1 , and 0 results in no traffic, integer.
- **TSC_Num:** The training sequence code that is inserted into each data burst, integer, $0 \leq \text{TSC_Num} \leq 7$.
- **Symbol_Rate:** The modulation symbol rate. The uplink sources and their sub-components expect a symbol rate of $270.8333\dots$ ksym/s, real, $0 < \text{Symbol_Rate}$.
- **Frequency_Hop:** Enables or disables frequency hopping, string, "Yes" or "No".
- **Mobile Allocation:** Specifies the number of 200 kHz bandwidth RF channels utilized

for frequency hopping, integer, 1 *Mobile_Allocation* 64.

- **Channel_Spacing**: Specifies the frequency separation (*Channel_Spacing* x 200 kHz) between each successive centre frequency of the RF channels utilized for frequency hopping, integer, 1 *Channel_Spacing*.
- **Hopping_Sequence_Num**: The hopping sequence number, integer, 0 *Hopping_Sequence_Num* 63.
- **Mobile_Allocation_Offset**: The RF channel index offset for this mobile, integer, 0 *Mobile_Allocation_Offset* (*Mobile_Allocation*- 1).
- **Delay_Time**: The first TDMA frame begins precisely 6 modulation symbol periods after *Delay_Time*, real, 0 *Delay_Time*.
- **R_Out**: The output resistance of the source, real, 0.01 *_R_Out_*.

An important aspect of the uplink sources is their ability to operate with 4, 8 or 16 samples per modulated symbol. The ability to operate with a different number of samples per modulated symbol enables the user to vary the simulation bandwidth and thereby balance the competing priorities of simulation accuracy and simulation time. The number of samples per modulated symbol depends upon the value of the *Symbol_Rate* parameter and the simulation time-step specified in the *Envelope* simulation controller. Although the *Symbol_Rate* parameter is described as specifying the symbol rate of the output modulated signal, in practice, it is not a parameter at all and should remain fixed at its default value, equal to the GSM/EDGE specified symbol rate of 270.83333... ksym/s. As a result, the number of samples per modulated symbol will depend only upon the simulation time-step. It is up to the user to set the simulation time step so as to produce the desired number of samples per modulated symbol. The *Test_Uplink* designs provide an example of how to set the simulation time step.

Data Display

A simple example showing each of the uplink sources in an envelope simulation is provided in two design files - *Test_Uplink_GMSK* and *Test_Uplink_8PSK*. These two design files are associated with two Data Display files of the same name. The two Data Display files contain a series of pages containing equations and plots for the analysis of results from simulations based upon the upstream sources. Each of the 5 pages of both files is discussed below under a separate heading that corresponds to the title of each page in the Data Display environment.

Variable Assignment

For the post-processing contained in the Data Display files to be carried out, certain results from the simulation domain are expected to be collected and subsequently assigned to particular variables in the Data Display files. This assignment of data variables from the simulation domain to Data Display variables is carried in the Variable Assignment page.

Spectrums and ACPR

The Spectrums and ACPR page is essentially the same for both uplink sources. Some of the signals and data that are displayed are color coded, with red corresponding to signals and data that are related to the reference envelope and blue corresponding to signals and data that are related to the distorted envelope.

The power of the reference and distorted envelopes is calculated and displayed in the two large lower plots. The bursty nature of GSM/EDGE signaling should be apparent in these plots. The user controls the start and end time of the time window over which the power spectral density of each envelope is estimated by adjusting the *StartSpecR* and *StartSpecD* markers, to set the start of the time window, in addition to the *EndSpecR* and *EndSpecD* markers, to set the end of the time window. The markers can be selected and moved using the mouse, with fine adjustment of the markers being possible using the left and right arrow keys (after the desired marker has been selected). The time and power that each marker corresponds to is shown to the left of each power versus time plot.

The ACPR for a selected channel can be determined and displayed using the formulas and tables on the right half of the page based upon the estimated power spectral density of each envelope. The ACPR calculations are based upon 200 kHz bandwidth channels. With the aid of the estimated power spectral density displayed in the large upper plots, the user sets the centre frequency of the channel for which the ACPR is desired to be calculated by setting the *RefCarrierFreq* and *DistCarrierFreq* variables appropriately. Using the user defined channel centre frequency, the lower and upper channel power ratio for each envelope is calculated and displayed. The ACPRs are determined using the ADS *acpr_vr* function.

Power Calculations

The primary aim of this page is to determine the peak-to-average power ratio (PAR) and the complementary cumulative distribution function (CCDF) for either the reference or distorted envelope over a specified length of time.

The reference or distorted envelope is selected for processing by setting the variable *VEnv* equal to *VRef* or *VDist*, respectively. With the desired envelope selected, the power of the envelope over the entire simulation run-time is determined and plotted. Like the Spectrums and ACPR page, the user controls the start and end times of the time window over which the PAR and CCDF are determined by adjusting the *Start* and *Stop* markers, with the time and power that each marker corresponds to being shown to the left of the plot.

Based upon the time window specified by the *Start* and *Stop* markers, the average power and PAR of the desired envelope are calculated and displayed. Note that the absolute value ("linear") and dB value of the average power and PAR are displayed. The CCDF of the desired envelope over the selected time window is determined and displayed in the lower half of the page.

Trajectories, Constellations and EVM

The Constellations, Trajectories and EVM page is designed to extract and display the trajectory and constellation of the reference and distorted envelopes for specified physical channels over the length of the simulation period. In addition, the error vector magnitude (EVM) and related statistics are calculated and displayed for:

- the distorted constellation relative to the reference constellation,
- the distorted constellation relative to an ideal constellation, and
- the reference constellation relative to an ideal constellation.

Similar to the Spectrums and ACPR page, the plots pertaining to the reference envelope have red traces, while those pertaining to the distorted envelope have blue traces. The post-processing steps necessary to determine the constellations and calculate EVM are slightly different for the *Test_Uplink_GMSK.dds* and *Test_Uplink_8PSK.dds* files. These differences will be highlighted as the page is discussed as it pertains to both files.

The first post-processing step needed to be carried out to extract the trajectory and constellation data is to "dehop" the reference and distorted envelopes. Dehopping of the envelopes is accomplished by the function *kmm_dehop*. The *kmm_dehop* function reverses any frequency hopping impressed upon an envelope by shifting the its spectrum over the course of each TDMA frame so that the spectrum is always centered about 0 Hz.

Following dehopping, the reference and distorted envelopes are further processed by, either, the function *kmm_gsm_gmsk_const_evm_v2*, in the case of GMSK modulation, or the function *kmm_gsm_8psk_const_evm_v2*, in the case of 8-PSK. These two functions only differ in the type of modulation expected upon the reference and distorted envelopes. The dehopped reference and distorted envelopes are processed by *kmm_gsm_gmsk_const_evm_v2* and *kmm_gsm_8psk_const_evm_v2* on a slot-by-slot basis in order to extract the trajectory and constellation data of the desired slot(s). Using the constellation data, the EVM (and associated measures) of the distorted constellation relative to the reference constellation is also calculated.

The filter selection parameter only has an effect when processing 8-PSK modulated data using the *kmm_gsm_8psk_const_evm_v2* function. Although the parameter is part of the parameter list for *kmm_gsm_gmsk_const_evm_v2*, it is not used when processing GMSK modulated data. Two alternative FIR filters are available, either an ISI suppression filter, or a raised-cosine filter. The filter coefficients are automatically adjusted for 4, 8 or 16 samples per symbol. The ISI suppression or raised-cosine filter is selected by setting the filter selection parameter to a value of 1 or 2, respectively. If the parameter is set to any other value than 1 or 2, no filtering will take place. The ISI suppression filter is based upon the *EDGE_RxFilter* component and the raised cosine filter on the *EDGE_RC_Window_RCFilter* component (see section 4.6.2 of [5]), both from the ADS GSM/EDGE component library.

Only an estimate of the start time of the first TDMA frame is required by the *kmm_gsm_gmsk_const_evm_v2* and *kmm_gsm_8psk_const_evm_v2* functions because they both seek out the training sequence symbols embedded within the first transmitted data burst to use as a timing reference (granularity of a symbol period) for the TDMA frames (and slots of each frame). The estimated start time of the first TDMA frame need only be specified within half a slot of the actual start time. This parameter is easily set if the start time of the first TDMA frame is known for the *kmm_dehop* function.

The clock offset fine tunes the timing for the symbol samples extracted from the reference envelope. The timing is intended to be adjusted by the user so that the EVM of the reference constellation relative to an ideal constellation is minimized. The timing is adjusted by moving the *ClockOffset* marker located within the small plot located immediately below the tables displaying the EVM results. Located within this same plot is the *PathOffset* marker. The *PathOffset* should be added to the path delay time so that the timing for the symbol samples extracted from the reference envelope can be adjusted by the user so that EVM of the distorted constellation relative to the reference constellation is

minimized.

The trajectory data for the series of extracted bursts from the (dehopped) reference and distorted envelopes are plotted in the upper two most plots on the page. The *kFrameStart* and *kFrameEnd* variables located above the trajectory plots enable the user to set the TDMA frame range that is used to generate the trajectory plots. That is, only the trajectory data from the slots selected by the *kslotusage* variable that fall within the TDMA frame range is plotted (the first TDMA frame, or portion thereof, is frame 0).

Located immediately below the trajectory plots, are the extracted constellations from the reference and distorted envelopes. These two plots are constructed from the constellations of the slots selected by the *kslotusage* variable and are over all TDMA frames, or part thereof, generated during the entire simulation period. The constellation plots include transition lines between the successive constellation points of each burst.

The EVM and associated statistics generated by the *kmm_gsm_gmsk_const_evm_v2* and *kmm_gsm_8psk_const_evm_v2* functions is tabulated in the box entitled "Distorted constellation relative to the Reference constellation".

To generate the results tabulated in the "Reference constellation relative to an Ideal constellation" and "Distorted constellation relative to an Ideal constellation" boxes the (dehopped) reference and distorted envelope constellation data produced by the *kmm_gsm_gmsk_const_evm_v2* and *kmm_gsm_8psk_const_evm_v2* functions is measured against an ideal constellation.

The following should be noted regarding all of the EVM and associated statistics in the three sets of tabulated results:

- The results are calculated over the constellation points from all of the slots specified by the *kslotusage* parameter and over all of the TDMA frames, or part thereof, of the simulation;
- The EVM results are calculated after adjusting for any phase difference, dc offset and gain/compression difference between the two constellations that form the basis of the EVM calculations; and,
- The *RMS EVM (%)*, *Peak EVM (%)* and *Mag Error (%)* are all given relative to the average normalized magnitude of the ideal constellation points, in the case of the tabulated data of the leftmost and centre boxes, and relative to the average normalized magnitude of the reference constellation points, in the case of the tabulated data of the rightmost box.

The constellation returned by the *kmm_gsm_gmsk_evm* and *_kmm_gsm_evm _* functions is a compensated version of the non-ideal constellation passed to either function. The returned constellation is described as compensated because the non-ideal constellation passed to either function is "corrected" by removing the phase, dc offset and gain/compression difference between it and the ideal constellation.

Envelope Plots

The Envelope Plots page contains 4 plots that are quite straight forward - the amplitude of the in-phase and quadrature components of the reference and distorted envelopes versus

time.

Downlink

The *GSM_EDGE_Downlink_GMSK_Src* and *GSM_EDGE_Downlink_8PSK_Src* components are shown in [ADS GSM/EDGE Downlink GMSK Modulated Source|#1105643] and [ADS GSM/EDGE Downlink 8-PSK Modulated Source](#), respectively.



GSM_EDGE_Downlink_GMSK_Src

X2

Carrier_Freq=880.0e6 Hz

Pav=0

BCCH_Channels=1

BCCH_Gain=0.8

C0_Channel_Number=0

SlotUsage_BCCH_Tr0=makearray(1,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0)

SlotUsage_Tr1=makearray(1,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0)

SlotUsage_Tr2=makearray(1,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0)

SlotUsage_Tr3=makearray(1,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0)

TSC_Num=0

Symbol_Rate=270.83333333e3 Hz

Frequency_Hop="No"

Mobile_Allocation=1

Channel_Spacing=1

Hopping_Sequence_Num=1

Delay_Time=0 sec

R_Out=50.0 Ohm

ADS GSM/EDGE Downlink GMSK Modulated Source



GSM_EDGE_Downlink_8PSK_Src

X2

Carrier_Freq=880.0e6 Hz

Pav=0

BCCH_Channels=1

BCCH_Gain=1.0

C0_Channel_Number=0

Slot_Usage_BCCH_Tr0=makearray(1,0,0,0,0,0,0,0,0)

Slot_Usage_Tr1=makearray(1,0,0,0,0,0,0,0,0)

Slot_Usage_Tr2=makearray(1,0,0,0,0,0,0,0,0)

Slot_Usage_Tr3=makearray(1,0,0,0,0,0,0,0,0)

TSC_Num=0

Symbol_Rate=270.83333333e3 Hz

Frequency_Hop="No"

Mobile_Allocation=1

Channel_Spacing=1

Hopping_Sequence_Num=1

Delay_Time=0 sec

R_Out=50.0 Ohm

ADS GSM/EDGE Downlink 8-PSK Modulated Source

The *GSM_EDGE_Downlink_GMSK_Src* and *GSM_EDGE_Downlink_8PSK_Src* components (the "downlink sources") can both generate a typical GSM/EDGE BS signal. A BS signal is a multicarrier signal that is a composite of one or more TDMA GMSK/8-PSK modulated signals. The operation of the two downlink sources is virtually identical. The primary difference between them is the modulation format of the traffic channels generated by each. Using the parameters of each component, the user is able to control:

- The number of carriers in the output multicarrier signal;
- The allocation of traffic to the slots of each carrier;
- The number of physical channels allocated for control information; and,
- The frequency hopping scheme for the carriers.

Each of the downlink sources is controlled by an identical set of parameters. The parameters are discussed below, where for each parameter a brief description is provided, as well as its numeric type and the range of values that it can take on, respectively.

- **Carrier_Freq:** The centre frequency of the power spectral density of the output multicarrier signal, real, $0 < \text{Carrier_Freq}$
- **Pav:** Relative average output power of the source, real, $0 < \text{Pav}$
- **BCCH_Channels:** The number of physical channels allocated for transport of BCCH logical channels ("BCCH physical channels" or "BCCH channels"). The BCCH physical channels are always allocated for transport using the C0 RF carrier (BCCH carrier) and utilize, in order, slot 0, 2, 4 and then 6, depending upon the number of BCCH physical channels, integer, $0 < \text{BCCH_Channels} < 4$.
- **BCCH_Gain:** Relative gain of the BCCH physical channel(s) and dummy bursts on the C0 RF carrier, real, $0.0 < \text{BCCH_Gain}$
- **C0_Channel_Number:** Channel index of the C0 RF carrier, integer, $0 < \text{C0_Channel_Number} < (\text{Mobile_Allocation} - 1)$.
- **Slot_Usage_BCCH_Tr0:** An 8 element array that specifies whether full-rate traffic is to be transmitted within the slots of the *Tr0* group that are free to be allocated traffic. The first element of the array corresponds to slot 0, the second to slot 1, and so on. An element having a value other than zero will result in the associated slot being allocated traffic and having a relative gain given by the value entered. The *Tr0* group of traffic channels is always associated with the BCCH channels, if any; thus, slots 0, 2, 4, and 6 may only be allocated traffic if they are not otherwise allocated as BCCH channels - slots 1, 3, 5 and 7 are always able to be allocated traffic. 8 element real array, 0.0 each array element.
- **Slot_Usage_Trx:** An 8 element array that specifies whether full-rate traffic is to be transmitted within the slots of the *Trx* group of traffic channels. The first element of the array corresponds to slot 0, the second to slot 1, and so on. An element having a value other than zero will result in the associated slot being allocated traffic and having a relative gain given by the value entered. Note: *Slot_Usage_Tr4*, *Slot_Usage_Tr5*, *Slot_Usage_Tr6* and *Slot_Usage_Tr7* are selected to not be displayed. 8 element real array, 0.0 each array element.
- **TSC_Num:** The training sequence code that is inserted into each data burst, integer, $0 < \text{TSC_Num} < 7$.
- **Symbol_Rate:** The modulation symbol rate. The downlink sources and their sub-components expect a symbol rate of 270.8333... ksym/s, real, $0 < \text{Symbol_Rate}$

- **Frequency_Hop**: Enables or disables frequency hopping, string, "Yes" or "No"
- **Mobile_Allocation**: Specifies the number of 200 kHz bandwidth RF channels utilized for frequency hopping, integer, 1 *Mobile_Allocation* 64.
- **Channel_Spacing**: Specifies the frequency separation (*Channel_Spacing* x 200 kHz) between each successive centre frequency of the RF channels utilized for frequency hopping, integer, 1 *Channel_Spacing*
- **Hopping_Sequence_Num**: The hopping sequence number used by all traffic channels, integer, 0 *Hopping_Sequence_Num* 63.
- **Delay_Time**: The first TDMA frame begins precisely 6 modulation symbol periods after *Delay_Time*, real, 0 *Delay_Time*.
- **R_Out**: The output resistance of the source, real, 0.01 *_R_Out_*.

Like the uplink sources, the downlink sources are able to operate with 4, 8 or 16 samples per modulated symbol. The ability to operate with a different number of samples per modulated symbol enables the user to vary the simulation bandwidth and thereby balance the competing priorities of simulation accuracy and simulation time. The number of samples per modulated symbol depends upon the value of the *Symbol_Rate* parameter and the simulation time-step specified in the *Envelope* simulation controller. Although the *Symbol_Rate* parameter is described as specifying the symbol rate of the output modulated signal, in practice, it is not a parameter at all and should remain fixed at its default value, equal to the GSM/EDGE specified symbol rate of 270.83333... ksym/s. As a result, the number of samples per modulated symbol will depend only upon the simulation time-step. It is up to the user to set the simulation time step so as to produce the desired number of samples per modulated symbol. The *Test_Downlink...* designs provide examples of how to set the simulation time step.

Data Display

A simple example showing each of the downlink sources in an envelope simulation is provided in two design files - *Test_Downlink_GMSK* and *Test_Downlink_8PSK*. The two, respective, Data Display files contain a series of 4 pages containing equations and plots for the analysis of results from simulations based upon the downlink sources. Each of these 4 pages is very similar to a corresponding page in the uplink Data Display files. The downlink files do not include a page that is similar to the Trajectories, Constellations and EVM page of the uplink files. Only the differences between the downlink and uplink pages are discussed below.

Variable Assignment

Whether an downlink source simulation is based upon GMSK or 8-PSK, both, the envelope of the output signal of the downlink source (the "reference envelope") and the envelope of the output signal of the device under test (the "distorted envelope"), are expected to be collected during simulation. Unlike the uplink Data Display files, since the downlink files do not derive and display EVM results, the complex impulse symbol stream(s) that drives the pulse-shaping filters in the 8-PSK downlink source need not be collected during simulation.

Spectrums and ACPR

The Spectrums and ACPR page of both downlink Data Display files is akin to its corresponding page in the uplink files.

When two or more traffic groups assign traffic to the same slot, the power spectral density taken over that slot will clearly show power associated with two or more channels. Furthermore, when interpreting results, keep in mind the following:

- Dummy bursts are always inserted onto the unused slots of the C0 carrier; and,
- For the 8-PSK downlink source, the BCCH channels and dummy bursts are GMSK modulated.

The dummy bursts inserted onto the C0 carrier can, in effect, be turned off, in the case where frequency hopping is enabled, by setting the relative power of all unused traffic slots, or in the case where frequency hopping is disabled, by setting the relative power of the unused slots of the *Tr0* traffic group, to a very tiny number (i.e., $1e-20$).

Power Calculations

The Power Calculations page of both downlink Data Display files is akin to its corresponding page in the uplink files.

The PAR for a multicarrier signal should be significantly greater than for a single carrier signal.

Envelope Plots

The Envelope Plots page of both downlink Data Display files is akin to its corresponding page in the uplink files.

WLAN

ADS Designs ()

- WLAN_802p11a_burst_Src
- WLAN_802p11a_Src
- WLAN_802p11b_CCK_burst_Src
- WLAN_802p11b_CCK_Src
- WLAN_802p11b_PBCC_burst_Src
- WLAN_802p11b_PBCC_Src

- Test_802p11a
- Test_802p11a_burst
- Test_802p11b_CCK
- Test_802p11b_CCK_burst
- Test_802p11b_PBCC
- Test_802p11b_PBCC_burst

Data Display Files (.dds)

- Test_802p11a
- Test_802p11a_burst
- Test_802p11b
- Test_802p11b_burst

Specifications

The designs are based upon the following IEEE 802.11 specifications:

[1]IEEE 802.11, 1999 Edition (ISO/IEC 8802-11: 1999) Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications

[2]IEEE 802.11a-1999 (8802-11:1999/Amd 1:2000(E)) Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications - Amendment 1: High-speed Physical Layer in the 5 GHz band

[3]IEEE 802.11b-1999 Supplement to 802.11-1999, Wireless LAN MAC and PHY specifications: Higher speed Physical Layer (PHY) extension in the 2.4 GHz band

[4]IEEE 802.11b-1999/Cor1-2001 Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications-Amendment 2: Higher speed Physical Layer (PHY) extension in the 2.4 GHz band-Corrigendum1

All specifications are available from: <http://standards.ieee.org/getieee802/802.11.html>

Usage and Operation

Six different 802.11 sources have been developed. The sources are denoted by the suffix `_Src` in the list of design files in [ADS Designs \(\)](#). The design files that immediately follow each `_Src` file are a sub-component design of the preceding `_Src` design.

The six sources can be divided into three pairs, where one pair is based upon the 802.11a standard (sources i and ii, above) and the other two pairs are based upon the 802.11b standard (sources iii through vi). Each pair of sources is comprised of a source that generates a time gated modulated signal, i.e. generates bursts of modulated data, and a (non-burst) source that generates a continuously modulated signal. The sources of each pair have the same name, but for the inclusion of *burst* in the name of the source that generate a time gated modulated signal.

Along with the six source designs shown in the file list in [ADS Designs \(\)](#) are six *Test_*

designs. Each *Test_* design corresponds to one of the sources and contains a simple simulation example of the source.

Simulation Domain

Two pairs of sources are based upon the 802.11b standard (the 802.11b sources). The *WLAN_802p11b_CCK_Src* and *WLAN_802p11b_CCK_burst_Src* components (the *CCK* sources) are shown in [ADS 802.11b CCK Modulated Sources](#). The *WLAN_802p11b_PBCC_Src* and *WLAN_802p11b_PBCC_burst_Src* components (the *PBCC* sources) are shown in [ADS 802.11b PBCC Modulated Sources](#):



WLAN_802p11b_CCK_burst_Src
X4
 Carrier_Freq=2412.0e6 Hz
 Pav=0
 Preamble_Header_Type="Long"
 Payload_Size=64
 Data_Rate=1e6
 Interburst_Time=20e-6 sec
 Chip_Rate=11.0e6 Hz
 Delay_Time=0 sec
 R_Out=50.0 Ohm



WLAN_802p11b_CCK_burst_Src
X4
 Carrier_Freq=2412.0e6 Hz
 Pav=0
 Preamble_Header_Type="Long"
 Payload_Size=64
 Data_Rate=1e6
 Interburst_Time=20e-6 sec
 Chip_Rate=11.0e6 Hz
 Delay_Time=0 sec
 R_Out=50.0 Ohm

ADS 802.11b CCK Modulated Sources



WLAN_802p11b_PBCC_Src
X5
 Carrier_Freq=2412.0e6 Hz
 Pav=0
 Data_Rate=1e6
 Chip_Rate=11.0e6 Hz
 Delay_Time=0 sec
 R_Out=50.0 Ohm



WLAN_802p11b_PBCC_burst_Src
X6
 Carrier_Freq=2412.0e6 Hz
 Pav=0
 Preamble_Header_Type="Long"
 Payload_Size=64
 Data_Rate=1e6
 Interburst_Time=20e-6 sec
 Chip_Rate=11.0e6 Hz
 Delay_Time=0 sec
 R_Out=50.0 Ohm

ADS 802.11b PBCC Modulated Sources

All four of the *802.11b* sources generate a quadrature modulated single carrier signal and support transmission at a rate of 1, 2, 5.5 and 11 Mbps. The two pairs of 802.11b sources are essentially identical, with the only material difference between them being the modulation format employed for 5.5 and 11 Mbps transmission - either CCK or PBCC. The *burst* source of each pair generates bursts of modulated data consistent with the burst structure specified in the 802.11b standard, with the modulator output power being ramped down to zero between each data burst. Both of the non-burst 802.11b sources generate a continuously modulated signal employing the same modulation format that is used for the payload portion of a burst produced by the source's associated *burst* source. In other words, the non-burst sources generate modulated payload data where the

payload size is infinitely large.

The characteristics of the modulated RF signal generated by both of the non-burst sources are controlled by an identical set of parameters. Similarly, the characteristics of the modulated RF data bursts generated by both burst sources are controlled using an identical set of parameters. Moreover, the parameter set of the non-burst sources is a subset of the parameter set of the burst sources.

As between the four 802.11b sources, identically named parameters have the same effect on the signal generated by each of the sources. Thus, all of the parameters of the 802.11b sources are discussed below, where for each parameter a brief description is provided, as well as its numeric type and the range of values that it can take on, respectively.

- **Carrier_Freq** : The RF carrier frequency of the output modulated signal, real, ($Chip_Rate / 2$) < *Carrier_Freq* .
- **Pav** : Relative average output power of the source, real, 0 *Pav* .
- **Preamble_Header_Type** : Format of the preamble and header, string, "Long" or "Short".
- **Payload_Size** : Size of the payload of each burst in bytes (octets), integer, 4 *Payload_Size* .
- **Data_Rate** : Transmission bit-rate of the payload, real, either: 1, 2, 5.5 or 11 Mbps.
- **Interburst_Time** : Time gap between consecutive bursts, real, 20 ms *Interburst_Time* .
- **Chip_Rate** : The DSSS chip rate for the preamble and header, as well as the payload when *Data_Rate* equals 1 or 2 Mbps. The symbol rate of the payload when *Data_Rate* equals 5.5 or 11 Mbps. All of the 802.11b sources are intended to operate at a chip rate of 11 Mcps.
- **Delay_Time** : The centre of the first chip of the preamble of the first burst generated occurs at *Delay_Time* plus the delay of the pulse shaping filter, real, 0 s *Delay_Time* .
- **R_Out** : The output resistance of the source, real, 0.01 *R_Out* .

The 802.11b sources are able to support different simulation step sizes, i.e. the number of samples per chip/symbol is not fixed. Support for a variable number of samples per chip/symbol allows for the simulation bandwidth to be adjusted and thereby facilitates a balance between the competing priorities of simulation accuracy and execution time. The number of samples per chip/symbol depends upon the value of the *Chip_Rate* parameter and the simulation time-step specified in the *Envelope* simulation controller. Although the *Chip_Rate* parameter is included in the 802.11b sources' parameter lists, it should remain fixed at its default value - 11 Mcps (Msyms/s), equal to the 802.11b specified chip/symbol rate. As a result of being fixed, the number of samples per chip/symbol will only depend upon the time step of the simulation. It is up to the designer to set the simulation time step to result in the desired number of samples per chip. The *Test...* designs provide examples of how to set the simulation time step. It is advisable that the number of samples per chip be a power of two and greater than or equal to 4.

Data Display

A simple example showing each of the 802.11b burst and non-burst sources in an envelope simulation is provided in *Test_802p11b_CCK* ., *Test_802p11b_CCK_burst* , *Test_802p11b_PBCC* , and *Test_802p11b_PBCC_burst* . The simulation design file names correspond with the 802.11b source being simulated.

The two non-burst source example simulations employ the *Test_802p11b* Data Display file and the two burst source example simulations employ the *Test_802p11b_burst* Data Display file. The two Data Display files contain a series of pages containing equations and plots for the analysis of results from simulations based upon the 802.11b sources. The *Test_802p11b* file contains 5 pages. The 5 pages are discussed below under a separate heading that corresponds to the title of each page in the Data Display environment.

Variable Assignment

For the post-processing contained in the Data Display files to be carried out, certain results from the simulation domain are expected to be collected and subsequently assigned to particular variables in the Data Display domain. This assignment of data variables from the simulation domain to Data Display variables is carried in the Variable Assignment page.

Spectrums

For discussion purposes, the Spectrums page can be divided into upper and lower halves. The upper half of the page displays the estimated power spectral density of the reference and distorted envelopes, calculated over a designer controlled period of time. The lower half of the page calculates and displays the power of each envelope as a function of time. The plots pertaining to the reference envelope have red traces, while those pertaining to the distorted envelope have blue traces.

The designer controls the start and end time of the time window over which the power spectral density of each envelope is estimated by adjusting the *StartSpecR* and *StartSpecD* markers, to set the start of the time window, and the *EndSpecR* and *EndSpecD* markers, to set the end of the time window. The markers can be selected and moved using the mouse, with fine adjustment of the markers being possible using the left and right arrow keys (after the desired marker has been selected). The time and power that each marker corresponds to is shown to the left of each power versus time plot.

Power Calcs.

The primary aim of the Power Calcs. page is to determine the peak-to-average power ratio (PAR) and the complementary cumulative distribution function (CCDF) for either the reference or distorted envelope over a specified length of time.

The reference or distorted envelope is selected for processing by setting the variable *VEnv* equal to *VRef* or *VDist* , respectively. With the desired envelope selected, the power of the envelope over the entire simulation run-time is calculated and plotted. Similar to the Spectrums page, the designer controls the start and end times of the time window over which the PAR and CCDF are determined by adjusting the *Start* and *Stop* markers. The

time and power that each marker corresponds to is displayed to the left of the plot. Based upon the time window specified by the *Start* and *Stop* markers, the average power and PAR of the desired envelope are calculated and displayed. Both, the absolute value ("linear") and dB value, of the average power and PAR are determined. The CCDF of the desired envelope over the selected time window is calculated and displayed in the lower half of the page.

EVM, Constellation & Trajectory

The EVM, Constellation & Trajectory page is designed to extract and display the trajectory and constellation of the reference and distorted envelopes over the length of the simulation period. In addition, the error vector magnitude (EVM) and related statistics are calculated and displayed for:

- The distorted constellation relative to the reference constellation,
- The distorted constellation relative to an ideal constellation, and
- The reference constellation relative to an ideal constellation.

The plots pertaining to the reference envelope have red points/traces, while those pertaining to the distorted envelope have blue points/traces.

The reference and distorted envelopes are processed by *const_evm_hpsk_ref_v2* to extract the trajectory and constellation data of each envelope. In addition, using the constellation data, the EVM (and associated measures) of the distorted constellation relative to the reference constellation is also calculated.

The clock offset fine tunes the timing for the extraction of the symbol samples from the reference envelope. The timing is intended to be adjusted by the user so that the EVM of the reference constellation relative to an ideal constellation is minimized. The timing is adjusted by moving the *ClockOffset* marker located within the small plot located immediately below the tables displaying the EVM results. Located within this same plot is the *PathOffset* marker. The *PathOffset* should be added to the path delay time (*kpathdelay*) so that the timing for the symbol samples extracted from the reference envelope can be adjusted by the user with the aim of minimizing the EVM of the distorted constellation relative to the reference constellation.

The following should be noted regarding all of the EVM and associated statistics in the three sets of tabulated results:

- The results are calculated over all constellation points for which constellation data exists for the two constellations that form the basis of the EVM calculations of each set of results;
- The EVM results are calculated after adjusting for any phase difference, dc offset and gain/compression difference between the two constellations that form the basis of the EVM calculations; and,
- The *RMS EVM (%)* , *Peak EVM (%)* and *Mag Error (%)* are all given relative to the average normalized magnitude of the ideal constellation points, in the case of the tabulated data of the leftmost and centre boxes, and relative to the average normalized magnitude of the reference constellation points, in the case of the

tabulated data of the rightmost box.

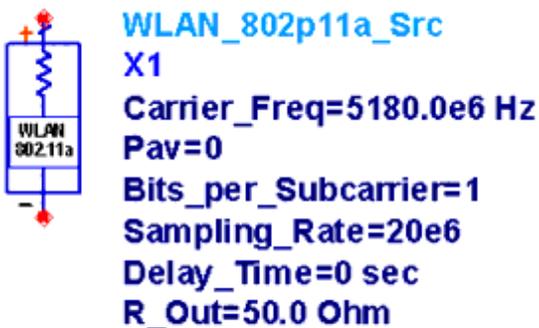
Envelope Plots

The Envelope Plots page contains 4 plots that are quite straight forward - the amplitude of the in-phase and quadrature components of the reference and distorted envelopes versus time.

802.11a

Simulation Domain

The *WLAN_802p11a_Src* and *WLAN_802p11a_burst_Src* components (the 802.11a sources) are shown in the following figure.



ADS 802.11a Modulated Sources

The 802.11a sources generate an OFDM (multicarrier) signal and supports transmission at a gross rate of 12, 24, 48 and 72 Mbps. The *burst* source generates bursts of OFDM modulated data consistent with the burst structure specified in the 802.11a standard, with the modulator output power being ramped down to zero between each data burst. The non-burst source generates a continuous OFDM modulated signal that employs the same subcarrier modulation format used for the payload portion of a burst produced by the burst source. As in the case of the 802.11b sources, the 802.11a non-burst source generates modulated payload data where the payload size is infinitely large.

From [ADS 802.11a Modulated Sources](#), it is apparent that the parameter set of the non-burst source is a subset of the burst source parameter set. The identically named parameters have the same effect on the signal generated by each of the sources. All of the parameters of the 802.11a sources are discussed below, where for each parameter a brief description is provided, as well as its numeric type and the range of values that it can take on, respectively.

- **Carrier_Freq** : The RF carrier frequency of the output modulated signal, real, ($Chip_Rate / 2$) < $Carrier_Freq$.
- **Pav** : Relative average output power of the source, real, 0 Pav .
- **Bits_per_Subcarrier** : Number of payload data bits allocated to each of the 48 data subcarriers per symbol period, integer, 1, 2, 4, or 6.
- **Payload_Size** : Size of the payload of each burst in bytes (octets), integer, ($Bits_per_Subcarrier$ 6) $Payload_Size$.
- **Interburst_Time** : Time gap between consecutive bursts, real, 20 ms $Interburst_Time$.
- **Sampling_Rate** : The sampling rate of the modulator (equal to the inverse of the simulation time step), real, 20 or 40 MHz.
- **Delay_Time** : The first sample of the first burst generated is output at $Delay_Time$, real, 0 s $Delay_Time$.
- **R_Out** : The output resistance of the source, real, 0.01 R_Out .

Data Display

A simple example showing the 802.11a non-burst and burst source in an envelope simulation is provided in *Test_802p11a* and *Test_802p11a_burst* , respectively. Each of the example simulation design files is associated with a Data Display file of the same name. The two Data Display files contain a series of four pages containing equations and plots for the analysis of results from simulations based upon the 802.11b sources. Currently, the two files are identical to one another and neither contains a page containing EVM calculations, constellations or trajectory plots.

Each of the four pages of the *Test_802p11a* and *Test_802p11a_burst* Data Display files is essentially identical to its corresponding page in the 802.11b Data Display files. Reference should be made to [Data Display](#), where the pages are discussed in more detail.

Variable Assignment

See [Variable Assignment](#).

Spectrums and ACPR

See [Spectrums](#).

Power Calcs.

See [Power Calcs.](#)

The PAR of a an 802.11a envelope should be significantly greater than that of an 802.11b envelope.

Envelope Plots

See [Envelope Plots](#).

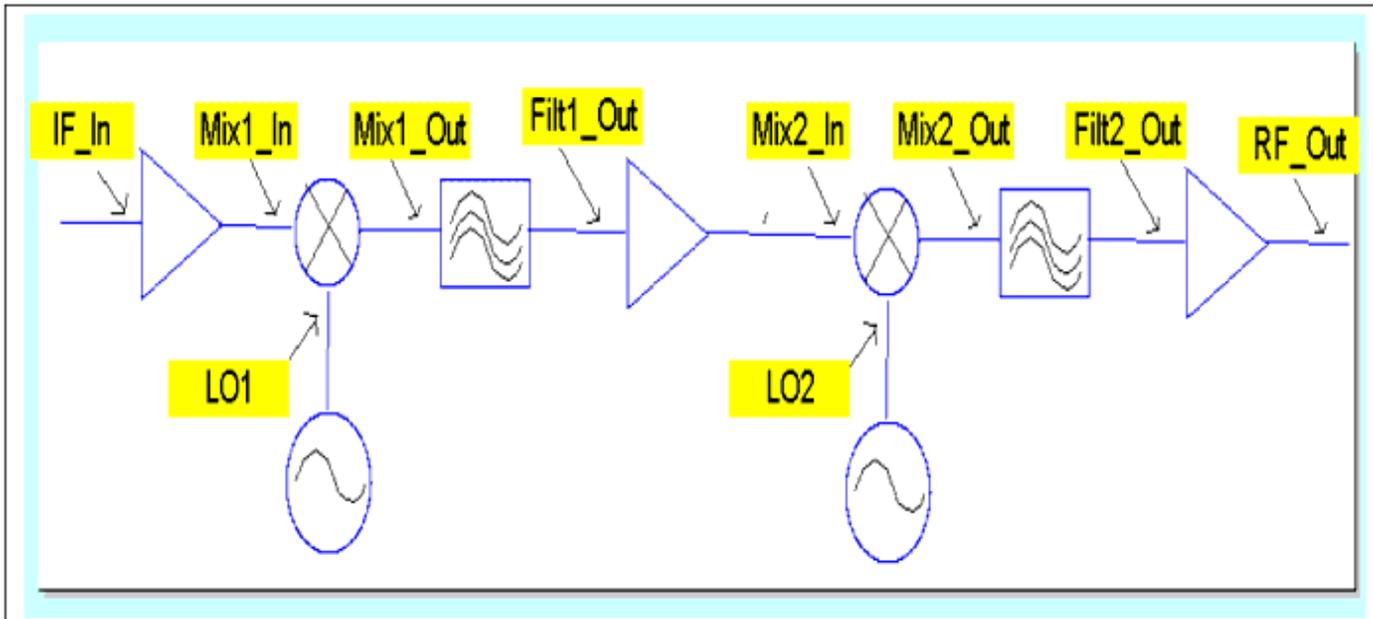
Transmitter/Receiver Topologies

There are 8 predefined topologies that have corresponding measurement Schematics and Data Displays. They are listed below:

- Single Stage Down Converter (mixer LSB)
- Double Stage Down Converter (mixer #1 LSB, mixer #2 LSB)
- Single Stage Up Converter (mixer LSB)
- Single Stage Up Converter (mixer USB)
- Double Stage Up Converter (mixer #1 LSB, mixer #2 LSB)
- Double Stage Up Converter (mixer #1 LSB, mixer #2 USB)
- Double Stage Up Converter (mixer #1 USB, mixer #2 LSB)
- Double Stage Up Converter (mixer #1 USB, mixer #2 USB)

You would select one of the given topologies, then proceed to modify it for your system requirements. The components in the schematic are just placeholders and you can replace, add, or expand on the given configurations. The important elements are the node and probe names and the frequency conversion of the mixers (i.e., whether it is LSB or USB). The identification of the LSB (lower sideband) or USB (upper sideband) is important for insuring that the simulation and data displays will give the correct results.

An important point to remember when performing frequency translation is that the USB component is the complex conjugate of the LSB component. So depending on which sideband component you choose, the fundamental output signal V_{fund_out} may need to be compared with V_{fund_in} or $\text{conj}(V_{fund_in})$. This is obvious by looking at the EVM results of the data display. Therefore in the `const_evm()` function, you need to replace the reference voltage signal V_{fund_in} with $\text{conj}(V_{fund_in})$.

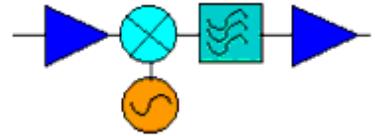


Co-Simulation Topologies

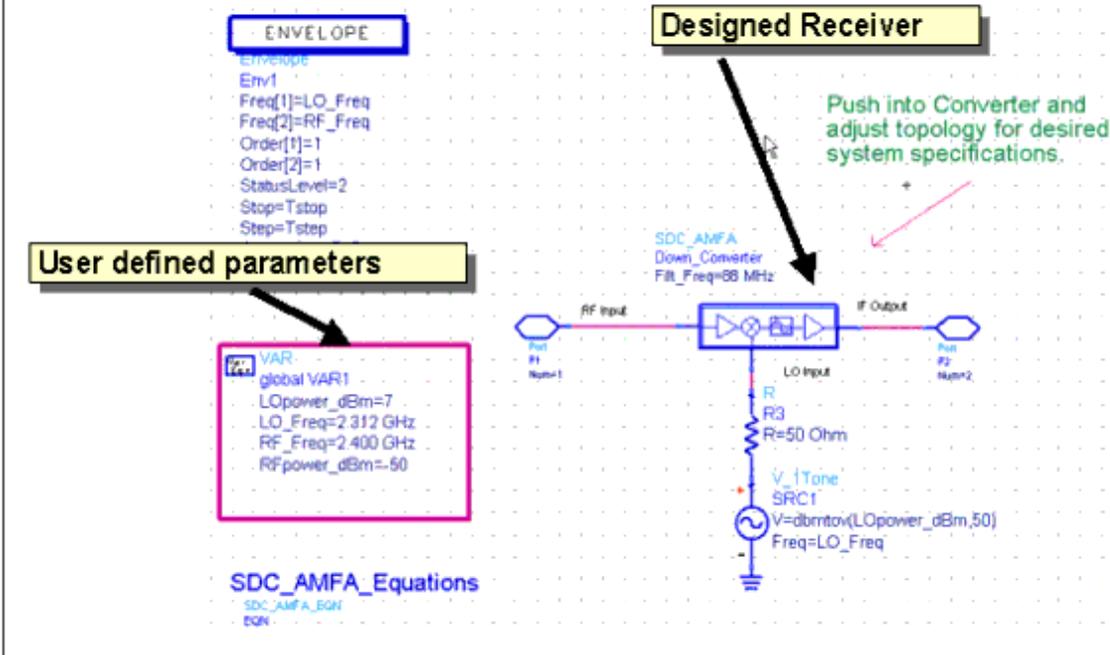
Ultimately, you would like to integrate the transmitter or receiver with the DSP portion of the communication system. Ptolemy can be used to generate a wide variety of baseband modulation signals. This allows you to perform more extensive tests on the system. The designed transmitter/receiver can then be integrated into Ptolemy using the co-simulator, allowing efficient translation between the DSP world and the Analog/RF world.

This figure shows a template for integrating your receiver design into the Ptolemy simulation. Again, the user-defined parameters would be inserted into the schematic. Notice that the envelope simulator has been used.

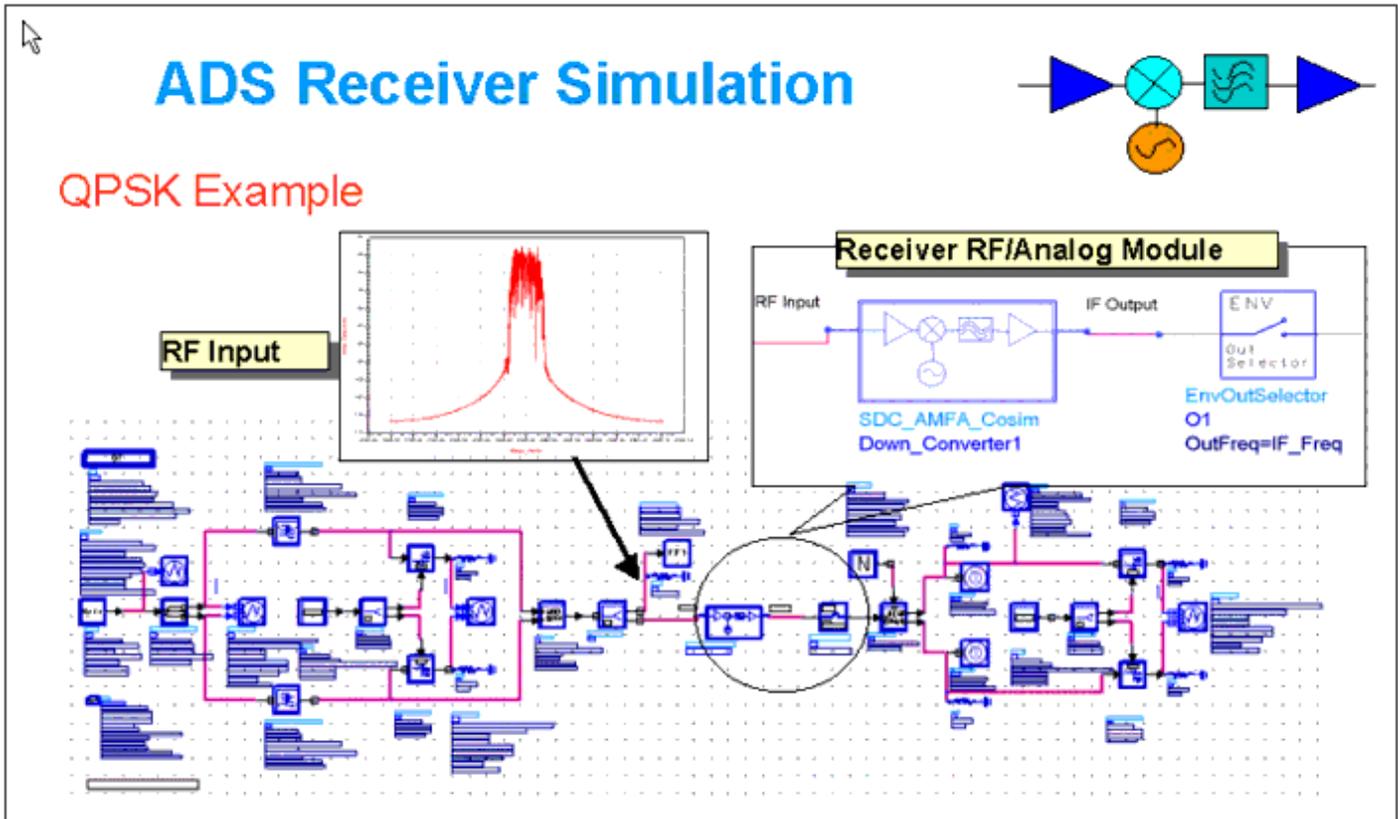
ADS Receiver Simulation



Pre-Configured Co-Simulation

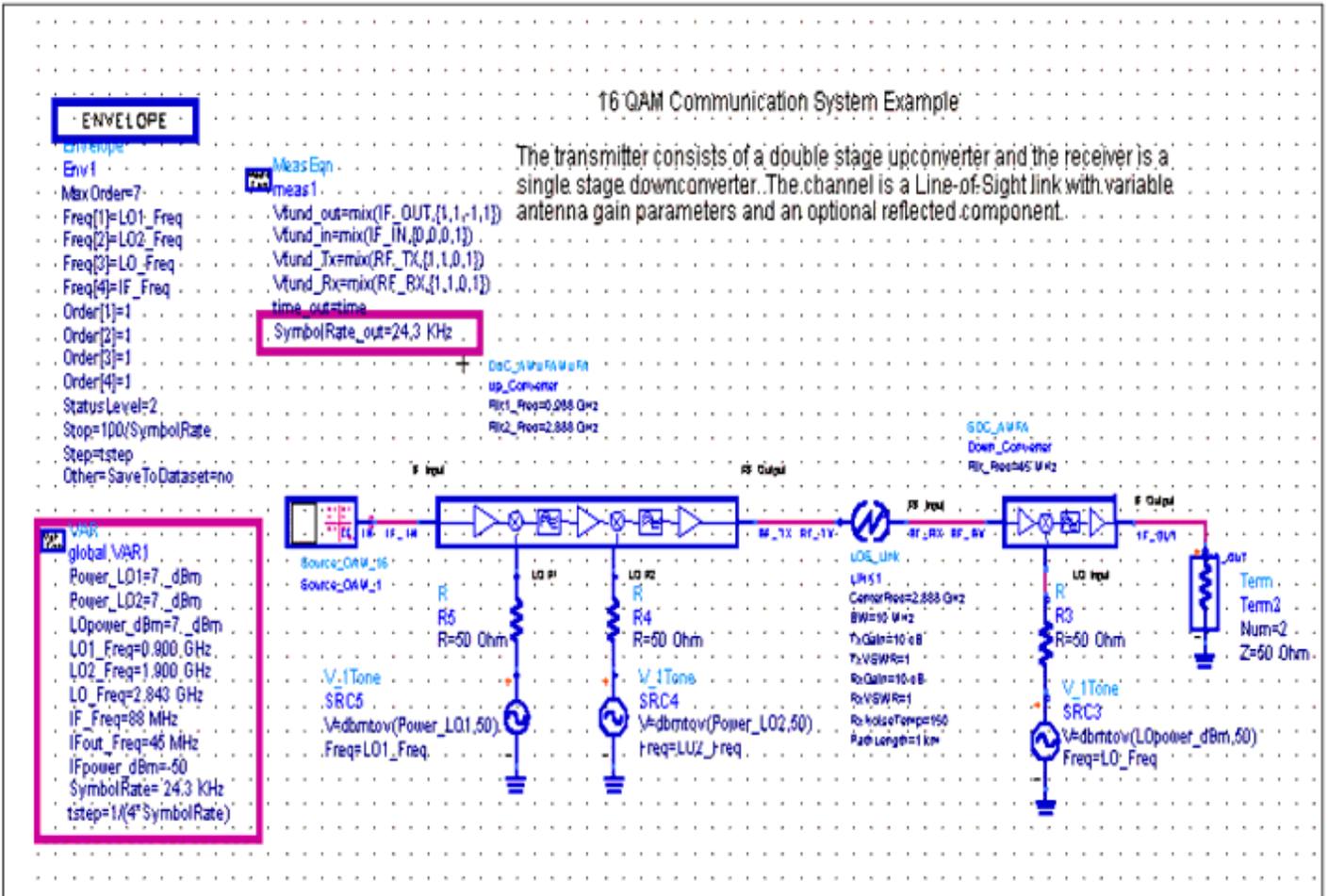


An example of the insertion of the receiver schematic template from the one created appears in the preceding figure. Also we have added the co-simulation block for translation from Analog/RF to Ptolemy.

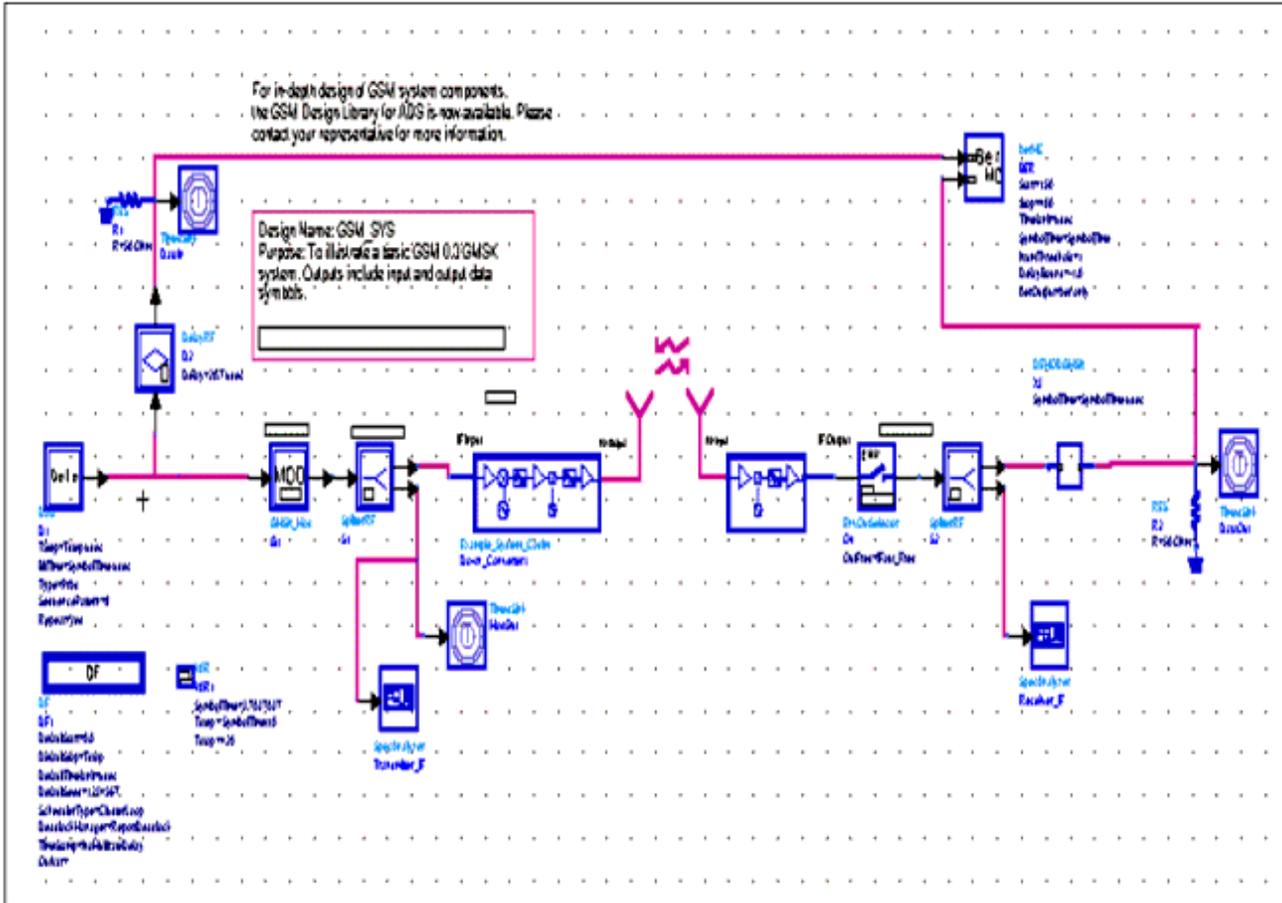


Communication Examples

The transmitter/receiver design dialog box assists you in creating up and down converters. The example below shows how the designed Transmitter and Receiver can be used in a communication system. This example uses a Line-of-Sight Link where the antenna gains, vswr, distance, etcetera can be specified. The example is simulated in the Analog/RF environment using a 16 QAM source.



Another example of a GSM communication system is shown here. The transmitter and receiver were created using the Transmitter/Receiver Design dialog box. These topologies were then inserted into a schematic set up for simulation, along with the identified channel and are identified via the up and down converter symbol. The schematic set up for simulation was then placed in the Ptolemy environment along with the GSM modulator and demodulator for a complete DSP to Analog/RF to DSP simulation.



Reference

1. W.P.Robins, *Phase Noise in Signal Sources* , IEE Telecommunications Series 9, 1982, Peter Peregrinus Ltd.
2. Digital Modulation in Communications Systems, Agilent Application Note 1298, literature number 5965-7160E.
3. Testing and Troubleshooting Digital RF Communications Transmitter Design, Agilent Application note number 1313.
4. Testing and Troubleshooting Digital RF Communications Receiver Designs, Agilent Application note number 1314, literature number 5968-3579E.
5. Using Vector Modulation Analysis in the Integration, Troubleshooting and Design of Digital RF Communications Systems, Agilent Product Note 89400-8, literature number 5091-8687E.
6. Understanding CDMA Measurements for Base Stations and Their Components, Agilent Application Note 1312, literature number 5968-0953E.
7. Voelker, Kenneth M. "Apply Error Vector Measurements in Communication Design", *Microwaves & RF*, December 1995, pp. 143-152.

¹ Table 2.1.3.1.14-1 of [1]

² Table 2.1.3.1.13.1-1 of [1]

³ Table 2.1.3.1.13.1-1 of [1]

⁴ Compare Figures 2.1.3.1.1.1-10 and 2.1.3.1.1.2-7 of [1]

⁵ Section 4.4 of [5]. Also see section 5.3.2 of [4].

⁶ Section 4.2.9 of [5].

⁷ Section 5.3.2 of [4].

⁸ Section 5.3.2 of [4].

⁹ See, for example, sections 5.3.3.3.1, 5.3.3.4.1, 5.3.3.5 and 5.3.3.10 of [4].

User Defined Functions

- `const_evm_oqpsk_ref_v2`
- `const_evm_hpsk_ref_v2`
- `kmm_evm_qpsk_v2`
- `kmm_evm_hpsk_v2`
- `const_evm_qpsk_ref_v2`
- `kmm_spectrum_shift`
- `kmm_dehop`
- `kmm_gsm_gmsk_const_evm_v2`
- `kmm_gsm_8psk_const_evm_v2`
- `kmm_gsm_gmsk_evm`
- `kmm_gsm_8psk_symbol_extract`
- `kmm_evm_gsm`
- `kmm_evm_qpsk`
- `const_evm_qpsk`

Usage and Operation

`const_evm_oqpsk_ref_v2` and `const_evm_hpsk_ref_v2` functions

The reference and distorted envelopes are processed by the `const_evm_oqpsk_ref_v2` function in the case of RC1 and RC2, and by the `const_evm_hpsk_ref_v2` function for RC3 through RC6. The reference and distorted envelopes are processed by the `const_evm...` functions to extract the trajectory and constellation data of each envelope. In addition, using the constellation data, the EVM (and associated measures) of the distorted constellation relative to the reference constellation is also calculated by the functions. The following parameters are passed to both `const_evm...` functions (the corresponding Data Display variable is set out in brackets following each parameter description):

- The reference envelope (*VRef*);
- The distorted envelope (*VDist*);
- The chip rate (*ksymbolrate*);
- A clock offset value to fine tune the chip sampling time for the reference envelope. (*ksampledelay*);
- The start time of the first chip period for which valid envelope data exists (for instance, the transient time associated with a filter) (*ktransientduration*); and,
- The time delay between the point in the simulation where the reference envelope is extracted to the point where the distorted envelope is extracted, i.e., the path delay (*kpathdelay*).

The `const_evm_oqpsk_ref_v2` function is also passed the time delay introduced into the quadrature component in the OQPSK modulator (*koffset*).

The clock offset fine tunes the timing for the extraction of the chip samples from the reference envelope. The timing is intended to be adjusted by the user so that the EVM of the reference constellation relative to an ideal constellation is minimized. The timing is

adjusted by moving the *ClockOffset* marker located within the small plot located immediately below the tables displaying the EVM results. Located within this same plot is the *PathOffset* marker. The *PathOffset* should be added to the path delay time (*kpathdelay*) so that the timing for the symbol samples extracted from the reference envelope can be adjusted by the user so that EVM of the distorted constellation relative to the reference constellation is also minimized.

The *const_evm...* functions return the trajectory, constellation and EVM data in a 10 element list (*klist1*). The 10 elements of the list are each assigned to a Data Display variable.

The extracted trajectory data of the reference and distorted envelopes is plotted in the two upper most plots on the page. The *ktrajectorysymbols* variable located above the trajectory plots enables the designer to set the total number of chip periods traversed by the trajectory data that is plotted. Located immediately below the trajectory plots are the extracted constellations from the reference and distorted envelopes. The EVM and associated statistics generated by the *const_evm...* functions is tabulated in the box entitled "Distorted constellation relative to the Reference constellation".

To generate the results tabulated in the " *Reference constellation relative to an Ideal constellation* " and " *Distorted constellation relative to an Ideal constellation* " boxes the reference and distorted envelope constellation data produced by the *const_evm...* functions is measured against an ideal constellation. The method of deriving the ideal constellation is different for the RC1 and RC2 envelopes versus those of RC3 through 6.

Since the envelopes are an OQPSK signal in the case of RC1 and RC2, the ideal constellations are readily derived by quantizing the in-phase and quadrature components of the chips (points) extracted from the reference and distorted constellations to produce an ideal constellation (that is based upon only 4 points, namely: (1,1), (1,-1), (-1,1) and (-1,-1)). This process of quantizing a constellation is known as "[slicing](#)"¹. The slicing operation, in effect, divides the I-Q plane into four quadrants and assigns one of the ideal constellation points to all of the points of the reference and distorted constellations falling into the same quadrant. Armed with an ideal constellation, the EVM and associated statistics are readily determined.

In the case of RC3 through RC6, the envelopes are an HPSK signal, thus the shape and size of the constellation that is employed will vary according to the number of active uplink channels and their relative power. As a result of the variety of HPSK constellations that may be encountered, it is impractical to derive the ideal constellation from the reference and distorted constellations through slicing. As an alternative to slicing, during the course of a simulation the ideal constellation points that are generated by the reverse link source are collected and stored in the dataset. Recall that within the Variable Assignment page the complex signal that drives the pulse shaping filters was collected and stored in the dataset, *ImpOut*, and was assigned to the Data Display variable *bb_symbol_impulse_stream*.

To perform the desired EVM calculations it is first necessary to extract the ideal constellation from the impulse symbol stream. The extraction of the ideal symbols is performed manually by adjusting the sweep indexing scheme used in the equation that

assigns data to the *ideal_syms* variable. The impulse symbol stream has the following properties:

- It's collected over the entire duration of the simulation;
- The symbol impulses (ideal constellation points) are spaced apart from one another by the number of samples per chip used for simulation.

With the aid of the *ksamplesperchip* variable, set to the number of samples per chip, and the tables located at the bottom of the page listing the data of the *bb_symbol_impulse_stream*, *ideal_syms* and *RefOut_const* (extracted constellation points of the reference envelope) variables, the limits of the indexing scheme can be adjusted to extract the ideal HPSK constellation. The first ideal constellation point extracted should reflect the delay introduced by the *ktransientduration* variable that is part of the call to the *const_evm_hpsk_ref_v2* function .

The slicing method has the benefit of making it unnecessary to store the chips/symbols generated during simulation in the dataset so as to be able to form an ideal constellation during post-processing. This saves simulation time and reduces the size of the dataset. Slicing does have a limitation, however, it will not result in an exact replica of the ideal constellation points when a transmitted symbol is sufficiently corrupted (by noise, interference and/or distortion, for example) such that it falls into the wrong quadrant.

kmm_evm_qpsk_v2 and **kmm_evm_hpsk_v2** functions

The EVM and associated statistics of the reference and distorted envelope constellations relative to an ideal constellation are calculated using the *kmm_evm_qpsk_v2* and *kmm_evm_hpsk_v2* functions. Both functions are part of the AEL user library. One parameter is passed to the *kmm_evm_qpsk_v2* function, namely, the reference or distorted constellation, *RefOut_const* or *DistOut_const* . The *kmm_evm_qpsk_v2* function will slice the passed constellation to derive the ideal constellation. Two parameters are passed to the *kmm_evm_hpsk_v2* function - the ideal constellation, *ideal_syms* , and the reference or distorted constellation, *RefOut_const* or *DistOut_const* . Both *kmm_evm...* functions return a list comprised of 6 elements (*klist2* in the case of the reference constellation and *klist3* in the case of the distorted constellation). The 6 elements of each list are each assigned to a Data Display variable. The resulting EVM and associated statistics are tabulated in the leftmost and centre boxes located in the upper part of the page.

const_evm_qpsk_ref_v2 function

The parameters are the same as with the *const_evm_oqpsk_ref_v2* function , but for the lack of the *koffset* parameter in the *const_evm_qpsk_ref_v2* function call.

kmm_spectrum_shift function

The multicarrier DS-CDMA signal of RC6 through RC9 is processed by selecting one of the carriers to analyze and isolating that carrier through filtering if necessary. The additional processing that is required is isolated in the lower left-hand portion of the page and is controlled through the *kchannel* and *kfilter* variables. The *kchannel* variable is set by the designer to either "Lower", "Upper", or "Centre" to select which of the three carriers is to

be analyzed. The *kfilter* variable is set by the designer to either "Yes" or "No" to control whether the selected carrier of the reference and distorted envelopes is to be filtered prior to extracting the constellation and trajectory data, and performing the EVM calculations.

The first step is to shift the frequency spectrum of the reference and distorted envelopes to centre the carrier to be analyzed about 0 Hz. The spectrum shift of both envelopes is performed using the *kmm_spectrum_shift* function from the user defined library. The parameters that are passed to the *kmm_spectrum_shift* function are:

- The time signal whose frequency spectrum is to be shifted (*VRef* or *VDist*),
- The amount by which the spectrum is to be shifted (*fshift*), and
- An initial phase for the complex exponential used to perform the shift (0.0).

After shifting the frequency spectrums, if desired, a filter is applied to the resultant envelopes to isolate the selected carrier. The filter is applied using the *kmm_filt* function, with the filter's coefficients being drawn from the series of filter coefficients that are compiled within the *kmm_filtcoef* function. The *kmm_filt* and *kmm_filtcoef* functions are both in the user defined library. The *kmm_filt* function performs a convolution of the two swept variables that are passed to it.

Following shifting of the frequency spectrums and filtering, if desired, the reference and distorted envelope of the carrier to be analyzed are assigned to the *VRef_s* and *VDist_s* variables, respectively. Post processing to extract the reference and distorted constellations and trajectory data, as well as calculate EVM results, is performed on *VRef_s* and *VDist_s* in the same manner as that described in *EVM, Constellation & Trajectory* (dgrfsys). In addition, the estimated power spectral density of *VRef_s* and *VDist_s* is calculated using the ADS *fs* function and displayed in the two additional plots on the far right hand side of the page.

kmm_dehop function

The first post-processing step needed to be carried out to extract the trajectory and constellation data is to 'dehop' the reference and distorted envelopes. Dehopping of the envelopes is accomplished by the user defined AEL function *kmm_dehop*. The *kmm_dehop* function reverses any frequency hopping impressed upon an envelope by shifting the its spectrum over the course of each TDMA frame so that the spectrum is always centered about 0 Hz. The following parameters must be passed to the *kmm_dehop* function, respectively (the corresponding Data Display variable is set out in brackets following each parameter description):

- The envelope to be dehopped (*VRef* or *VDist*);
- The GMSK/8-PSK symbol rate (*symbolrate*);
- The time corresponding to 4 symbol periods prior to the start of the first (zero'th) frame. If no additional delay is added to the start of an uplink source simulation (i.e. *Delay_Time* = 0 seconds) the point in time that is 4 symbol periods prior to the start of the first frame is equal to 2 symbol periods. ($2.0 / symbolrate$);
- The mobile allocation (*kMobileAllocation*);
- The channel spacing (*kChannelSpacing*);
- The hopping sequence number (*kHoppingSequenceNumber*); and
- The mobile allocation offset (*kMobileAllocationOffset*).

The *kmm_dehop* function assumes that the first frame to be processed is frame number zero. The dehopped reference and distorted envelopes are assigned to the *VRef_dehop* and *VDist_dehop* variables, respectively.

kmm_gsm_gmsk_const_evm_v2 or **kmm_gsm_8psk_const_evm_v2** functions

Following dehopping, the reference and distorted envelopes are further processed by, either, the function *kmm_gsm_gmsk_const_evm_v2*, in the case of GMSK modulation, or the function *kmm_gsm_8psk_const_evm_v2*, in the case of 8-PSK. These two functions are part of the user defined AEL library and only differ in the type of modulation expected upon the reference and distorted envelopes. The dehopped reference and distorted envelopes are processed by *kmm_gsm_gmsk_const_evm_v2* and *kmm_gsm_8psk_const_evm_v2* on a slot-by-slot basis in order to extract the trajectory and constellation data of the desired slot(s). Using the constellation data, the EVM (and associated measures) of the distorted constellation relative to the reference constellation is also calculated. The following parameters must be passed to the *kmm_dehop* function, respectively (the corresponding Data Display variable is set out in brackets following each parameter description):

- The dehopped reference envelope (*VRef_dehop* , or *VRef* , if no frequency hopping is implemented in the uplink source);
- The dehopped reference envelope (*VDist_dehop* , or *VDist* , if no frequency hopping is implemented in the uplink source);
- The desired filter to apply to each dehopped envelope prior to trajectory and constellation extraction and EVM calculation. (*kfiltype*);
- The GMSK/8-PSK symbol rate (*symbolrate*);
- The estimated start time of the first TDMA frame. (*kestframestart*);
- A clock offset value to fine tune the symbol sampling time for the reference envelope. (*ksampledelay*);
- The time delay between the point in the simulation where the reference envelope is extracted to the point where the distorted envelope is extracted, i.e. the path delay. (*kpathdelay*);
- The slots of each frame from which the trajectory and constellation data is to be extracted. (*kslotusage*); and,
- The training sequence number of the training sequence that is embedded within each burst transmitted (*kTSCnum*).

The filter selection parameter only has an effect when processing 8-PSK modulated data using the *kmm_gsm_8psk_const_evm_v2* function. Although the parameter is part of the parameter list for *kmm_gsm_gmsk_const_evm_v2*, it is not used when processing GMSK modulated data. Two alternative FIR filters are available, either an ISI suppression filter, or a raised-cosine filter. The filter coefficients are automatically adjusted for 4, 8 or 16 samples per symbol. The ISI suppression or raised-cosine filter is selected by setting the filter selection parameter to a value of 1 or 2, respectively. If the parameter is set to any other value than 1 or 2, no filtering will take place. The ISI suppression filter is based upon the *EDGE_RxFilter* component and the raised cosine filter on the *EDGE_RC_Window_RCFilter* component (see section 4.6.2 of [5]), both from of the ADS Ptolemy GSM/EDGE component library.

Only an estimate of the start time of the first TDMA frame is required by the *kmm_gsm_gmsk_const_evm_v2* and *kmm_gsm_8psk_const_evm_v2* functions because they both seek out the training sequence symbols embedded within the first transmitted data burst to use as a timing reference (granularity of a symbol period) for the TDMA frames (and slots of each frame). The estimated start time of the first TDMA frame need only be specified within half a slot of the actual start time. This parameter is easily set if the start time of the first TDMA frame is known for the *kmm_dehop* function.

The slots of each frame from which trajectory and constellation data is extracted are specified using an 8 element swept integer variable where each successive element corresponds, in order, to slot 0 through 7. A slot will be processed when its associated element in the swept variable has a non-zero value. Note that the *kmm_gsm_gmsk_const_evm_v2* and *kmm_gsm_8psk_const_evm_v2* functions do not process half-rate traffic channels, they only process full-rate traffic channels. Typically, the same slots that are selected to carry full-rate traffic within the simulation that generated the current dataset will be selected for post-processing.

kmm_gsm_gmsk_const_evm_v2 and **kmm_gsm_8psk_const_evm_v2** functions

These functions return the trajectory, constellation and EVM data in a 10 element list (*klist1*). The 10 elements of the list are each assigned to a Data Display variable (see the formulas located in the box located below the *kmm_gsm_gmsk_const_evm_v2* and *kmm_gsm_8psk_const_evm_v2* functions).

The trajectory data for the series of extracted bursts from the (dehopped) reference and distorted envelopes are plotted in the upper two most plots on the page. The *kFrameStart* and *kFrameEnd* variables located above the trajectory plots enable the user to set the TDMA frame range that is used to generate the trajectory plots. That is, only the trajectory data from the slots selected by the *kslotusage* variable that fall within the TDMA frame range is plotted (the first TDMA frame, or portion thereof, is frame 0).

Located immediately below the trajectory plots, are the extracted constellations from the reference and distorted envelopes. These two plots are constructed from the constellations of the slots selected by the *kslotusage* variable and are over all TDMA frames, or part thereof, generated during the entire simulation period. The constellation plots include transition lines between the successive constellation points of each burst.

kmm_gsm_gmsk_evm function

In the case of GMSK (*Test_Uplink_GMSK.dds*), the *kmm_gsm_gmsk_evm* function, contained in the AEL user library, is used to determine the EVM and associated statistics for both, the reference and distorted constellations, relative to an ideal constellation. The ideal constellations are readily derived by quantizing the in-phase and quadrature components of the symbols (points) of the reference and distorted constellations to produce an ideal constellation that is based upon only 4 points, namely: (1,1), (1,-1), (-1,1) and (-1,-1). This process of quantizing a constellation is known as "[slicing](#)"¹. The slicing operation, in effect, divides the I-Q plane into four quadrants and assigns one of the ideal constellation points to all of the points of the reference and distorted constellations falling into the same quadrant. Armed with an ideal constellation, the EVM and associated statistics are readily determined. The *kmm_gsm_gmsk_evm* function is

passed only one parameter, the reference or distorted constellation, as the case may be (*RefOut_const* or *DistOut_const*). Like the *kmm_gsm_gmsk_const_evm_v2* function, the *kmm_gsm_gmsk_evm* function returns a list, in this case, comprised of 7 elements (*klist2* for the case of the reference constellation and *klist3* for the case of the distorted constellation). The 7 elements of the list are each assigned to a Data Display variable (see the formulas located in the box located below the *kmm_gsm_gmsk_evm* functions). The resulting EVM and associated statistics are tabulated in the leftmost and centre boxes located in the upper centre of the page. In addition to the EVM and associated statistics, the list returned by the *kmm_gsm_gmsk_evm* function also contains a compensated reference or distorted constellation, as the case may be. Constellation compensation for the purposes of EVM results is discussed below.

kmm_gsm_8psk_symbol_extract function

Before the desired *EVM* calculations can be performed, the desired ideal symbols must be extracted from the impulse symbol stream to construct the ideal constellation. The extraction of the ideal symbols and the construction of the ideal constellation is performed by the *kmm_gsm_8psk_symbol_extract* function, part of the *AEL* user library of functions. The *kmm_gsm_8psk_symbol_extract* function is required to be passed the following parameters, respectively (the corresponding Data Display variable is set out in brackets following each parameter description):

- The impulse symbol stream that drives the 8-PSK pulse-shaping filters (*bb_symbol_impulse_stream*);
- The 8-PSK symbol rate (*symbolrate*);
- The estimated start time of the first TDMA frame. (*kestframestart*);
- The slots of each frame for which the constellation data is desired. (*kslotusage*);
- and,
- The training sequence number of the training sequence that is embedded within each burst transmitted (*KTSCnum*).

The impulse symbol stream has the following properties:

- It's collected over the entire duration of the simulation;
- It has no frame or slot delimiters; and,
- The symbol impulses are spaced 4, 8 or 16 samples apart, depending upon the number of samples per symbol used for simulation.

Due to these characteristics of the impulse symbol stream, frame and symbol synchronization must first be performed before the ideal symbols of the desired slots (*kslotusage*) can be extracted. The symbol synchronization is performed by first determining the average power of a set of equally spaced samples for all possible time-step offsets (4, 8 or 16). The time-step offset of the set that has the highest average power (in fact, the average power of all sets except for the desired set should be zero) provides for symbol synchronization. Frame synchronization is performed by correlation using the training sequence code that is embedded within each normal burst transmitted. Once frame and symbol synchronization is achieved, the ideal symbols of the desired slots are extracted and the ideal constellation is constructed.

The ideal constellation returned by the *kmm_gsm_8psk_symbol_extract* function is

assigned to the *ideal_syms* Data Display variable. The ideal constellation, along with phase transitions, is shown in a small plot located in the lower centre of the page.

kmm_gsm_evm function

To determine the EVM and associated statistics of the (dehopped) reference and distorted envelope constellations relative to the ideal constellation returned by the *kmm_gsm_8psk_symbol_extract* function, the *kmm_gsm_evm* function, from the AEL user library, is invoked twice. The *kmm_gsm_evm* function is passed two parameters, the ideal constellation, *ideal_syms*, and the reference or distorted constellation, as the case may be (*RefOut_const* or *DistOut_const*). Like the *kmm_gsm_gmsk_evm* function, the *kmm_gsm_evm* function returns a list comprised of 7 elements (*klist2* for the case of the reference constellation and *klist3* for the case of the distorted constellation). The 7 elements of each list are each assigned to a Data Display variable (see the formulas located in the box located below the *kmm_gsm_evm* functions). The resulting EVM and associated statistics are tabulated in the leftmost and centre boxes located in the upper centre of the page. In addition, the list returned by the *kmm_gsm_evm* function also contains a compensated reference or distorted constellation, as the case may be.

[1](#) This is minimum distance decoding.