MATLAB implementation
of the 3GPP Spatial Channel Model
(3GPP TR 25.996)

Implementation Documentation

Version:	1.21 (scm-05-07-2006)
Date:	July 5, 2006
File:	scm-05-07-2006.zip

Requires MATLAB 6.5.0 (R13) or later

Authors: Jari Salo (formerly TKK)
Giovanni Del Galdo (TUI, giovanni.delgaldo@tu-ilmenau.de)
Jussi Salmi (TKK, jussi.salmi@tkk.fi)
Pekka Kyösti (EBIT, pekka.kyosti@elektrobit.com)
Marko Milojevic (TUI, marko.milojevic@tu-ilmenau.de)
Daniela Laselva (Nokia, daniela.laselva@nokia.com)
Christian Schneider (TUI, christian.schneider@tu-ilmenau.de)

TKK = Helsinki University of Technology
TUI = Technical University of Ilmenau
EBIT = Elektrobit

Document History:	
14.05.2004	Document created
24.05.2004	First draft with chapters: Introduction, High level structure and Model input/output interface
27.05.2004	Chapter added.
28.05.2004	(Giovanni) Small changes in the text.
29.05.2004	(Jari) Added examples, small changes in the text.
23.07.2004	(Pekka) Added/modified input parameters in Tables 1 to 3.
26.07.2004	(Jari) Added Sections 2,5,6 + some editing work
8.12.2004	(Jari) Edited the original WINNER internal document for public release.
14.12.2004	(Jari) Added Section 6
4.1.2005	(Jari) Modified polarization power normalization and Section 7.
6.3.2005	(Jari) Corrected Table 5 and Table 6. Small editorial changes.
5.7.2006	(Jari, Jussi) Corrected a minor bug in scm_mex_core.c, channel coefficient computation remains unaffected. Minor/cosmetic changes in some MATLAB functions.

Table of Contents

1. Introduction

This document describes a MATLAB implementation of the 3GPP Spatial Channel Model (SCM) [3GPP03]. The accompanying MATLAB code implements the system level channel model described in Section 5 of [3GPP03], except the “far scatterer clusters” feature from Section 5.5.2 and the intercell interference feature from Section 5.7. These features were left out of the implementation because they make certain restricting assumptions about the network layout. The current implementation has no built-in assumptions on network geometry and is hence usable for a variety of simulation purposes.

The channel model takes the Multiple-Input Multiple-Output (MIMO) radio link parameters, model configuration parameters, and antenna parameters as inputs, and outputs the MIMO channel matrices. Channel matrices can be generated for multiple BS-MS links with one function call. The output is a multi-dimensional array which contains the channel impulse responses for a pre-defined number of radio links.

Mapping of MS-BS distances, array orientations and MS mobility parameters into the SCM input format is left to the system simulator program. To make using of the model easier default (random) parameters can be used. Channel convolution and other related operations are beyond the scope of the implemented channel model.

The work presented in this report has been in most parts carried out in Workpackage 5 of the WINNER[1] project [WIN].

2. Installation

The code has been created and tested using Matlab version 6.5.0 (Release 13). Older versions might cause unexpected problems. While the software was primarily developed on Windows operating systems (2000Pro, XP), Linux was also used in both development and testing.

The SCM package installs as a MATLAB mini-toolbox. The installation proceeds as follows:

1. Unzip the files into a directory called, for example, ‘winner’ and add it to MATLAB path. See MATLAB command `addpath` for information on how to add a directory to MATLAB path.
2. Type ‘help winner’ at MATLAB prompt to get started.

To compile the ANSI-C functions (optional) MATLAB’s mex compiler must be properly configured. Compilation of the ANSI-C interpolation functions also requires that GNU Scientific Library [GSL] is properly installed and configured in your system.

[1] Wireless World Initiative New Radio

3. Description of files included in the package

Table 1 lists the MATLAB and C-files included in the package. A short description of each file is also given.

Table 1. MATLAB and ANSI-C files included in the distributed package

Filename	Description
scm.m	This the main function called by the user to generate channel matrices. It calls the auxiliary functions that generate bulk parameters, do antenna pattern interpolation and compute the actual channel matrices.
scmparset.m	Helper function for setting the default parameter for the first input struct
linkparset.m	Helper function for setting the default parameter for the second input struct
antparset.m	Helper function for setting the default parameter for the third input struct
pathloss.m	Function for computing the default pathloss according to [3GPP03, Table 5.1]
interp_gain.m	Function for antenna pattern interpolation using MATLAB's interpolating functions
scm_core.m	Function for computing "the big for loop" that generates the channel matrices
generate_bulk_par.m	Function for generating the random "bulk" parameters for macro and micro cells
interp_gain_c.m	MATLAB front end that calls interp_gain_mex.c
scm_mex_core.m	Help text file for scm_mex_core.c
scm_mex_core.c	Channel matrix generation implemented in ANSI-C for faster computation
interp_gain_mex.c	Antenna field pattern interpolation using GNU Scientific Library (GSL) interpolating functions. This is faster than interp_gain.m but requires that GSL is installed in the system. See [GSL] for further information.
cas.m	A utility function for computing the circular angle spread as defined in [3GPP03, Annex A]. This function is not necessary for channel matrix generation.
ds.m	A utility function for computing rms delay spread. This function is not necessary for channel matrix generation.
dipole.m	A utility function that generates the pattern of a slanted dipole for vertical and horizontal polarizations. This is useful for creating some simple MIMO antenna configurations.

In addition to the files listed in Table 1 the distribution package also includes this document, a readme text file and a license.txt file.

4. Model input/output interface

The full syntax for the SCM function is ([.] indicates optional arguments):

[CHAN, [DELAYS], [FULLOUTPUT]] = SCM(SCMPAR, LINKPAR, ANTPAR, [INITVALUES]).

Some quick comments:

- All input arguments are MATLAB structs. The first three input arguments are mandatory. A helper function will be supplied so that their default values can be set easily.
- The fourth input argument is optional. When given, SCM does not generate the channel parameters randomly, but uses the user-supplied initial channel values.
- The first output argument is a 5D-array containing the MIMO channel matrices for all links over a specified number of time samples.
- The second output argument includes multipath delays for all links. The path delays are given in seconds.
- The third output argument is a MATLAB struct containing the randomly generated link parameters and the final phases of the complex sinusoids. This MATLAB struct can be used as INITVALUES in subsequent function calls.

The input and output parameters are explained in the following sections.

4.1 Input parameters

There are four input arguments, all of which are MATLAB structs. The first three arguments are mandatory. Tables 1-4 describe the fields of the input structs.

Table 2: General channel model parameters. Common for all links, MATLAB struct SCMPAR.

Parameter name	Definition	Default value	Unit
NumBsElements	The number of elements in the BS array. This parameter is ignored if antenna patterns are defined in the input struct ANTPAR. In this case the number of BS elements is extracted from the antenna definition.	2	
NumMsElements	The number of elements in the MS array. This parameter is ignored if antenna patterns are defined in the input struct ANTPAR. In this case the number of BS elements is extracted from the antenna definition.	2	
Scenario	Selected SCM channel,scenario (‘suburban_macro’, ‘urban_macro’ or ‘urban_micro’)	’urban_micro’	-
SampleDensity	Time sampling interval of the channel. A value greater than one should be selected if Doppler analysis is to be done.	2	samples/half wavelength
NumTimeSamples	Number of channel samples (impulse response matrices) to generate per link.	100	-
UniformTimeSampling	If this parameter has value ‘yes’, the time sampling interval of the channel for each user will be equal. Sampling interval will be calculated from the SampleDensity and the highest velocity found in the input parameter vector MsVelocity. If this parameter has value ’no’, then the time sampling interval for each link will be different, if MSs have different speeds (see linkpar.MsVelocity). Setting this parameters ‘yes’ may be useful in some system-level simulations where all simulated links need to be sampled at equal time intervals, regardless of MS	‘no’	

	speeds.		
BsUrbanMacroAS	BS mean angle spread for urban macro environment. Possible values are ‘eight’ and ‘fifteen’ degrees. This variable is ignored if ‘Scenario’ is not ‘urban_macro’.	‘eight’	-
NumPaths	Number of paths (channel taps). Path delays are drawn from the delay distribution specified in [3GPP03] regardless of the number of paths set.	6	-
NumSubPathsPerPath	Number of sub-paths per path. The only value supported in the SCM specification is 20 subpaths, see [3GPP03, Table 5.2].	20	-
CenterFrequency	Carrier center frequency. Affects path loss and time sampling interval.	2E9	Hz
ScmOptions	SCM options ('none','polarized','los','urban_canyon'). The options are mutually exclusive.	‘none’	-
DelaySamplingInterval	Delay sampling interval (delay resolution). The default corresponds to Tc/16, where Tc=1/3.84e6 [3GPP03, Sec. 5.3.1].	1.6276e-8	sec
XpdIndependentPower	With this set to ‘yes’ the power of the elements of the channel matrix (without pathloss) is normalized to a constant, that does not depend on the XPD ratios. See Section 6.2.	‘no’	-
PathLossModelUsed	Path loss included in the channel matrices yes/no (if ‘no’, PL is calculated and returned in the third output argument, but not multiplied with the channel matrices)	‘no’	-
ShadowingModelUsed	Shadow fading included in the channel matrices yes/no (if ‘no’ shadow fading is still computed and returned in the third output argument, but not multiplied with the channel matrices). Note that if both path loss and shadowing are switched off the average power of the channel matrix elements will be one (with azimuthally uniform unit gain antennas).	‘no’	-
PathLossModel	The name of the path loss function. Function ‘pathloss’ implements the default SCM path loss model. If the default is used, center frequency is taken from the parameter CenterFrequency. One can define his/her own path loss function. For syntax, see PATHLOSS.	‘pathloss’	-
AnsiC_core	Use optimized computation yes/no. With ‘yes’ faster C-function is used instead of m-function. Note the C-function SCM_MEX_CORE.C must be compiled before usage. For more information, see SCM_MEX_CORE.M.	‘no’	-
LookUpTable	If optimized computation is used, complex exponentials are taken from a look-up table to speed up computation or calculated explicitly. This parameter defines the lookup table size. Value 0 indicates that lookup table is not used, value –1 uses the default table size 2^{14} =16384. The size of the table must be a power-of-two. If AnsiC_core = ‘no’ this parameter is ignored.	0	integer
RandomSeed	Random seed for fully repeatable channel generation (if empty, seed is generated randomly). Note that even if RandomSeed is fixed, two channel realizations may still not be the same due to potential differences between random number generators in different MATLAB versions. Note also that one must also use the same link and antenna parameters.	<empty>	integer

All parameters in this MATLAB struct are vectors of length K, where K is the number of links. The values, if not specified in [3GPP03] are randomly generated; they are not based on any specific network geometry or user mobility model and are provided for easier usage of the model. For a brief example of link parameter configuration, see Section 6.1.

Table 3: Link-dependent parameters, MATLAB struct LINKPAR.

Parameter name	Definition	Default value	Unit
MsBsDistance	Distance between BS and MS	Users are approximately uniformly distributed in a circular cell over distances of [35,500] meters	m
ThetaBs	θ_{BS} (see Figure 1)	U(-180,180)	deg
ThetaMs	θ_{MS} (see Figure 1)	U(-180,180)	deg
OmegaBs	Ω_{MS} (see Figure 1), this parameter is not currently used.	NaN	deg
OmegaMs	Ω_{MS} (see Figure 1), this parameter is not currently used.	NaN	deg
MsVelocity	MS velocity	10*ONES(1,K)	m/s
MsDirection	θ_v (see Figure 1)	U(-180,180)	deg
MsHeight	Height of MS. Possibly needed for path loss computation.	1.5*ONES(1,K)	m
BsHeight	Height of BS. Possibly needed for path loss computation.	32*ONES(1,K)	m
MsNumber	Index number (positive integer) of the MS for each simulated link. This parameter is needed for generating shadow fading values with inter-site correlation. Shadow fading is correlated for links between a single MS and multiple BSs (inter-site correlation). There is no correlation between shadow fading between different MSs.	[1:K]	-

The following parameters characterize the antennas. In this SCM implementation, only linear arrays with dual-polarized elements are supported. The antenna patterns do not have to be identical. The complex field pattern values for the randomly generated AoDs and AoAs are interpolated.

Table 4. Antenna parameters, MATLAB struct ANTPAR

Parameter name	Definition	Default value	Unit
BsGainPattern	BS antenna field pattern values in a 4D array. The dimensions are [ELNUM POL EL AZ] = SIZE(BsGainPattern), where ELNUM is the number of antenna elements in the array. The elements may be dual-polarized. POL – polarization. The first dimension is vertical polarization, the second is horizontal. If the polarization option is not used, vertical polarization is assumed (if both are given). EL – elevation. This value is ignored. Only the first element of this dimension is used. AZ – complex-valued field pattern in the azimuth dimension given at azimuth angles defined in	1	

	BsGainAnglesAz. If NUMEL(BsGainPattern)=1, all elements are assumed to have uniform gain defined by the value of BsGainPattern over the full azimuth angle, and the number of BS antenna elements is defined by NumBsElements. This speeds up computation since field pattern interpolation is not required.		
BsGainAnglesAz	Vector containing the azimuth angles for the BS antenna field pattern values. These values are assumed to be the same for both polarizations. This value is given in degrees over the range (-180,180) degrees. If NUMEL(BsGainPattern)=1, this variable is ignored.	linspace(-180,176,90)	deg
BsGainAnglesEl	Vector of elevation angles for definition of BS antenna gain values. This parameter is for future needs only; its value is ignored in this implementation (SCM does not support elevation).	-	-
BsElementPosition	Element positions for BS linear array in wavelengths. This parameter can be either scalar or vector. If scalar, uniform spacing equal to the scalar is applied. If vector, it defines antenna element positions on a line. Note that one can place two elements in the same position and, by defining the antenna patterns properly, create dual-polarized arrays.	0.5	wavelength
MsGainPattern	MS antenna field pattern values in a 4D array. The dimensions are [ELNUM POL EL AZ] = SIZE(MsGainPattern), where ELNUM – the number of antenna elements in the array. The elements may be dual-polarized. POL – polarization. The first dimension is vertical polarization, the second is horizontal. If the polarization option is not used, vertical polarization is assumed (if both are given). EL – elevation. This value is ignored. Only the first element of this dimension is used. AZ – complex-valued field pattern in the azimuth dimension given at azimuth angles defined in MsGainAnglesAz. If NUMEL(MsGainPattern)=1, all elements are assumed to have uniform gain defined by the value of MsGainPattern over the full azimuth angle, and the number of MS antenna elements is defined by scmpar.NumMsElements. This speeds up computation since field pattern interpolation is not needed.	1	complex
MsGainAnglesAz	Vector containing the azimuth angles for the MS antenna field pattern values. These values are assumed to be the same for both polarizations. This value is given in degrees over the range (-180,180) degrees. If NUMEL(BsGainPattern)=1, this variable is ignored.	linspace(-180,176,90)	deg
MsGainAnglesEl	Vector of elevation angles for definition of MS antenna gain values. This parameter is for future needs only; its value is ignored in this implementation (SCM does not support elevation).	-	-
MsElementPosition	Element positions for MS linear array in wavelengths. This parameter can be either scalar or vector. If scalar, uniform spacing is applied. If vector, it defines antenna element positions on a line. Note that one can place two elements in the same position and, by defining the antenna patterns properly, create dual-polarized arrays.	0.5	wavelength
InterpFunction	The name of the interpolating function. One can replace this with his own function. For syntax, see interp_gain.m, which is the default function. For faster computation, see interp_gain_c.m	‘interp_gain’	
InterpMethod	The interpolation method used by the interpolating function. Available methods depend on the function. The	‘cubic’	

	default function is based on MATLAB's interp1.m function and supports e.g. 'linear' and 'cubic' (default) methods. Note that some methods, such as 'linear', cannot extrapolate values falling outside the field pattern definition.		

Parameter matrices BsGainPattern and MsGainPattern 2nd dimension is either 1 or 2. If polarization option is in use, the field pattern values have to be given for vertical and horizontal polarizations (in this order). If polarization is not used only the first dimension, i.e. vertical, is used, if both are given.

Note that the mean power of narrowband channel matrix elements (i.e. summed over delay domain) depends on the antenna gains. The default antenna has unit gain for both polarizations. Hence, the mean narrowband channel coefficient power is two for 'polarized' option, and one for all other options.

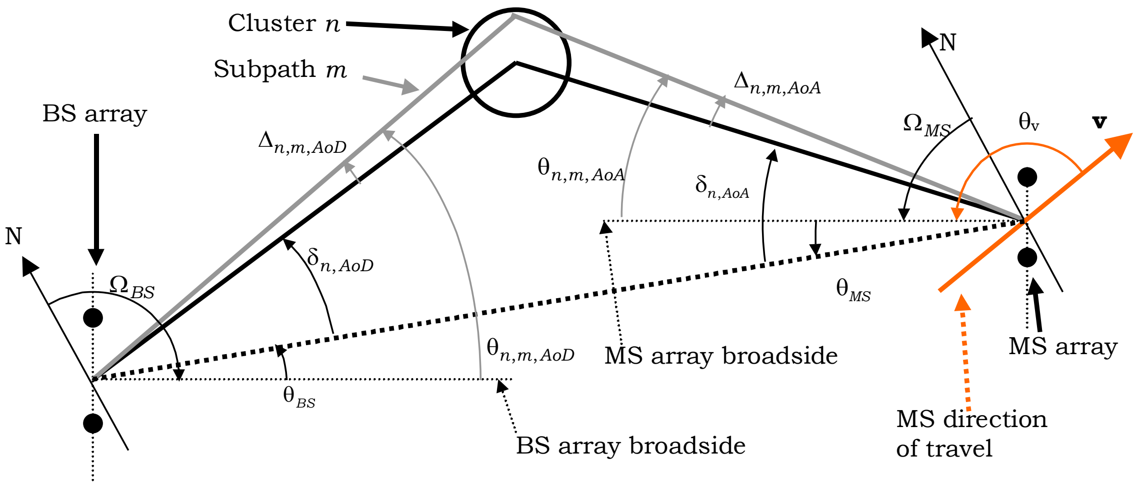

Figure 1: BS and MS angular parameters in SCM specification [3GPP03].

The fourth input argument, which is also a MATLAB struct, is optional. It can be used to specify the initial AoDs, AoAs, phases, path losses and shadowing values when SCM is called recursively, or for testing purposes. If this argument is given, the random parameter generation as defined in SCM is not needed. Only the antenna gain values will be interpolated for the supplied AoAs and AoDs.

The fields of the MATLAB struct are given in the following table. When using the fourth input argument, the names of the fields must be written exactly as given in Table 5. In a typical usage scenario, one first calls the scm function with three output arguments, and then inserts the third output argument as the fourth input argument (possibly with modified field values) in a subsequent function call to the scm function. Notation: K denotes the number of links, N denotes the number of paths, M denotes the number of subpaths within a path.

Table 5. Initial values, fourth optional input argument, a MATLAB struct INITVALUES. Note: the names of the fields must be written as given in the table.

Parameter name	Definition	Unit
delays	A K x N matrix of path delays.	Sec
path_powers	A K x N array of powers of paths.	linear scale

aods	A K x N x M array of angles of departure of subpaths	Degrees
aoas	A K x N x M array of angles of arrival of subpaths	Degrees
subpath_phases	A K x N x M array of initial subpath phases. When polarization option is used, this is a K x P x N x M array, where P=4. In this case the second dimension includes the phases for [VV VH HV HH] polarized components.	Degrees
path_losses	A K x 1 vector of path losses	linear scale
shadow_fading	A K x 1 vector of shadow fading losses	linear scale
xpd	A K x 2 x N array of cross-polarization coupling power ratios. The second dimension is the [V-to-H H-to-V] coupling ratios. This is needed only when 'polarized' option is used.	linear scale

4.2 Output parameters

There are three output arguments: CHAN, DELAYS, FULLOUTPUT. The last two are optional output parameters. Notation: K denotes the number of links, N is the number of paths, T the number of time samples, U the number of receiver elements, and S denotes the number of transmitter elements.

Table 6. Output parameter of the SCM function.

Parameter name	Definition	Unit
CHAN	A 5D-array with dimensions U x S x N x T x K	
DELAYS	A K x N vector of path delay values. Note that delays are, for compatibility with the INITVALUES, also included in FULLOUTPUT.	sec
FULLOUTPUT	A MATLAB struct with the following elements:	
delays	A K x N matrix of path delays. This is identical to the second output argument.	sec
path_powers	A K x N array of powers of paths.	linear scale
aods	A K x N x M array of subpath angles of departure	degrees
aoas	A K x N x M array of subpath angles of arrival	degrees
subpath_phases	A K x N x M array giving the final phases of all subpaths. When polarization option is used, a K x P x N x M array, where P=4. In this case the second dimension includes the phases for [VV VH HV HH] polarized components.	degrees
path_losses	A K x 1 vector	linear scale
shadow_fading	A K x 1 vector	linear scale
delta_t	A K x 1 vector defining time sampling interval for all links.	sec
xpd	A K x 2 x N array of cross-polarization coupling power ratios. The second dimension is the [V-to-H H-to-V] coupling ratios.	linear scale

5. Optimized computation

The computationally heaviest parts of the SCM channel model have been implemented with ANSI-C, namely the computation of the channel coefficients ('core') and interpolation of antenna field patterns. The latter is based on interpolation functions in the GNU Scientific Library [GSL] and can be used only on platforms, where GSL has been installed.

5.1 scm_mex_core.m

To make use of the optimized computation, one must:

1) Compile the ANSI-C function. The simplest way to do this is to type 'mex scm_mex_core.c' at the MATLAB command prompt (provided that MATLAB's C compiler has been configured properly).
2) Set 'scmpar.AnsiC_core='yes''.

There may be considerable differences between C compilers with respect to the resulting performance. It was noticed that Linux's gcc compiler provides consistently good performance, whereas using MATLAB's own lcc compiler may result in somewhat less satisfactory results.

Further performance improvement may be achieved by setting the parameter scmpar.LookUpTable=-1. This activates the lookup table for computing the complex exponential, in the core equation of the channel model; this parameter is only applicable when ANSI-C computation is used. Alternatively, one can set the number of points in the look-up table by e.g. setting scmpar.LookUpTable=1024. The size of the lookup table must be a power-of-two. The default lookup table size (with LookUpTable = -1) is 2^14=16384. Notice that using the lookup table results in distortion in the generated signal. With the default value of 2^14 points the quantization distortion is about -83 dB.

One can also try to tune the compilation process with compiler-dependent code optimization options. This may result in further performance improvement.

5.2 interp_gain_c.m

In some applications, particularly when the SCM function is called repeatedly for a small number of time samples, antenna field pattern interpolation may constitute a large part of computation. For such applications it may be worthwhile to use the ANSI-C written interpolation function interp_gain_c. The function is based on the interpolation functions in GNU Scientific Library (GSL) and supports linear and cubic spline interpolation with periodic boundary conditions. Look-up table -based interpolation is also supported for uniformly sampled field patterns. To compile the function, type

```
mex -lgsl -lgslcblas -lm interp_gain_mex.c
```

at the MATLAB command prompt. The GNU Scientific Library [GSL] must be installed in the system for successful compilation. For list of platforms supported by GSL, see [GSL].

6. Implementation notes

6.1 Configuration of link parameters

In this section a brief example of how to configure the link parameters (input struct linkpar) is given. We assume the simple network shown in Figure 2. Channel matrices will be generated for a total of four links will be simulated. We assume that the first two links correspond to MS #1, and the latter two to MS #2. This is configured by setting

```
linkpar.MsNumber = [1 1 2 2]
```

which tells the simulation program that the first two links are for MS #1 and the latter two links are for MS #2. The indexing is arbitrary (any positive integers will do), but it is advisable to select consecutive integers for computational efficiency[2]. The MS indexing affects the inter-site shadow fading correlation. In the example, the shadow fading values between the first two links (MS #1) and between the last two links (MS #2) would be correlated. However, there would be no shadow fading correlation, for example, between the second and the fourth link; although in Figure 2 these are the two links connected to BS #2 there are no dependencies between any of the channel parameters associated with *different* MSs. In other words [3GPP03, Sec. 5.6] describes the generation of angle spread, delay spread and shadowing deviation parameters for a single MS connected to multiple BSs. The procedure therein should be repeated for each individual MS.

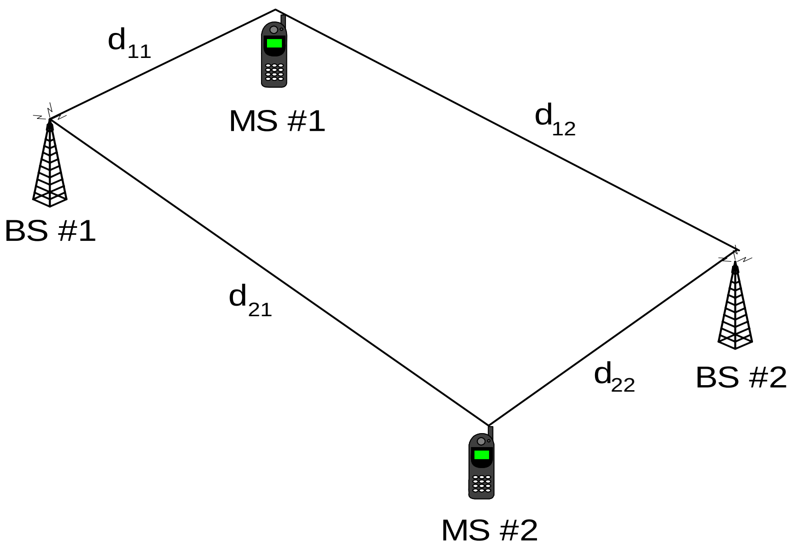

Figure 2. Link parameter configuration example

User needs also to determine the array orientations θ_{BS} and θ_{MS} with respect to the line drawn between the BS and MS arrays, see Figure 1. Similarly, MS-BS distances, the MS velocities and the directions of MS movement need to be set. For example, denoting the distance between the ith MS and the jth BS with d_{ij}, one needs to set

```
linkpar.MsBsDistance = [{d11} {d12} {d21} {d22}]
```

If the link parameters are unknown, one can use the random default values generated by the `linkparset` function.

6.2 Normalization of the channel matrices with the polarization option

The power normalization of the channel matrices in the polarization option [3GPP03, Sec. 5.5.1] takes into account that path loss depends on polarization. Because of this, when the polarization option is switched on, the elements of the channel matrix do not have unit power. Rather, the channel matrix elements' powers depend on the random cross-polarization discrimination (XPD) ratio values. The reason is that the pathloss can be correctly applied only with respect to V-V (or same polarization) transmission. However, in some applications one may not be interested in the path loss, and would

[2] The program currently generates max(linkpar.MsNumber) shadow fading values. Therefore setting `linkpar.MsNumber = [1 1 100 100]` would generate 100 shadow fading coefficients, although only two of them are used.

prefer to normalize the power of the channel matrix elements to a constant[3], i.e., independent of the XPD ratios. In the current implementation this can be achieved by setting scmpar.XpdIndependentPower='yes'. This setting makes the following change in the equation in [3GPP03, page 24] :

$$\begin{bmatrix} \exp\left(j\Phi_{n.m}^{(v,v)}\right) & \sqrt{r_{n1}}\exp\left(j\Phi_{n.m}^{(v,h)}\right) \\ \sqrt{r_{n2}}\exp\left(j\Phi_{n.m}^{(h,v)}\right) & \exp\left(j\Phi_{n.m}^{(h,h)}\right) \end{bmatrix} \to \begin{bmatrix} \sqrt{\frac{\mathrm{xpd}_1}{1+\mathrm{xpd}_1}}\exp\left(j\Phi_{n.m}^{(v,v)}\right) & \sqrt{\frac{1}{1+\mathrm{xpd}_1}}\exp\left(j\Phi_{n.m}^{(v,h)}\right) \\ \sqrt{\frac{1}{1+\mathrm{xpd}_2}}\exp\left(j\Phi_{n.m}^{(h,v)}\right) & \sqrt{\frac{\mathrm{xpd}_2}{1+\mathrm{xpd}_2}}\exp\left(j\Phi_{n.m}^{(h,h)}\right) \end{bmatrix}.$$

Here $XPD_1=1/r_{n1}$ and $XPD_2=1/r_{n2}$. Note that the power of the channel matrix elements depends also on the antenna properties. With the default antennas, which have unit gain on both polarizations over the entire 360 degree azimuth angle, the actual mean narrowband power (summed over all path delays) with the polarization option is two.

6.3 Sectorized Base-Stations[4]

The following text from [3GPP03] describes how to employ sectorized base-stations in the SCM model:

> "The composite AS, DS, and SF shadow fading, which may be correlated parameters depending on the channel scenario, are applied to all the sectors or antennas of a given base. Sub-path phases are random between sectors. The AS is composed of 6 x 20 sub-paths, and each has a precise angle of departure which corresponds to an antenna gain from each BS antenna. The effect of the antennas gain may cause some change. The SF is a bulk parameter and is common among all the BS antennas or sectors."

This can be achieved with the present implementation as follows. Assume the sectorized cell layout of Figure 3. The channels of all "A" sectors are obtained first by employing the scm code as described earlier in this document. Thus we set scmpar, linkpar and antpar to describe the channels between the users and all "A" sectors and get the channels of the "A" sectors HA as

```
>> [HA delays bulkpar]=SCM(scmpar,linkpar,antpar);
```

Then we re-randomize the subpath phases and rotate the the DoD:s in accordance with the direction of the "B" and "C" sectors to get the channels HB and HC corresponding to "B" and "C" sectors as

```
%% Randomize subpath phases for B.
>> bulkpar.subpath_phases=360*rand(size(bulkpar.subpath_phases));
%% Rotate DoDs to go from “A” to “B” antenna.
>> bulkpar.aods=rem(bulkpar.aods-120+180,360)-180;
%% Get “B” channels.
>> [HB delays bulkpar]=SCM(scmpar,linkpar,antpar,bulkpar);
%% Randomize subpath phases “C”.
>> bulkpar.subpath_phases=360*rand(size(bulkpar.subpath_phases));
%% Rotate DoDs to go from “B” to “C” antenna.
>> bulkpar.aods=rem(bulkpar.aods-120+180,360)-180;
%% Get “C” channels.
>> [HC delays bulkpar]=SCM(scmpar,linkpar,antpar,bulkpar);
```

[3] Usage of this option is entirely an application-dependent issue. If one wants to compare system performance (e.g. BER) of a V-V and dual-polarized system, then the path loss difference is important and this option should *not* be used.

[4] Sections 6.3 and 6.4 are due to Per Zetterberg.

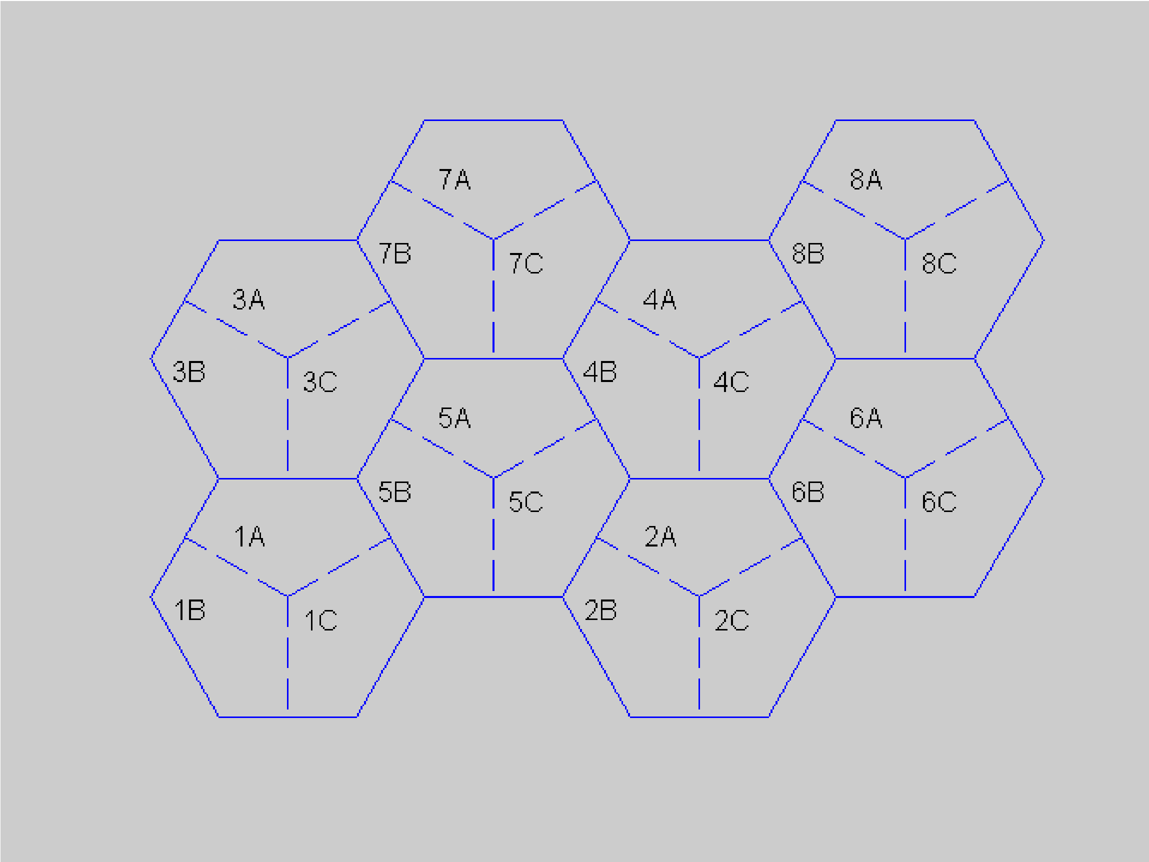

Figure 3: Illustration of three-sector cellular network.

6.4 General Array Configurations

General array configurations (array manifolds) can be supported with a trick. Set

```
>> antpar.BsElementPosition=zeros(1, scmpar.NumBsElements)
```

then specify the phase and amplitude of the antenna elements in antpar.BsGainPattern. Note that the phase of the elements should now also include the phase offsets between the antenna elements which is a function of element position and angle of departure.

7. Examples

7.1 Basic examples of channel matrix generation

```
% Matrix generation for 100 MS-BS links
>> H=scm(scmparset,linkparset(100),antparset);
% Setting and modifying default input parameters
>> scmpar=scmparset;
>> linkpar=linkparset(10);  % 10 links
>> antpar=antparset;
>> scmpar.NumTimeSamples=10;  % 10 time samples per link
>> scmpar.ScmOptions='urban_canyon';  % urban canyon option
>> scmpar.Scenario='urban_macro';  % urban macro scenario
>> [H1 delays out]=scm(scmpar,linkpar,antpar);
% using final conditions as initial conditions in next function
% call
>> [H2 delays out]=scm(scmpar,linkpar,antpar,out);
```

7.2 Antenna patterns and a dual-polarized 2x4 MIMO system

```
>> scmpar.NumTimeSamples=1e5;  % 100000 time samples per link
>> scmpar=scmparset;
>> scmpar.ScmOptions='polarized';  % polarization option
>> linkpar=linkparset(1);  % one link
>> antpar=antparset;
>> az=antpar.BsGainAnglesAz;
>> bs_array=zeros(4,2,1,length(az));  % four outputs
>> bs_array(1,:,:,:)=dipole(az,45);   % 45 degree slanted dipole
>> bs_array(2,:,:,:)=dipole(az,-45);  % -45 degree slanted
>> bs_array(3,:,:,:)=dipole(az,45);
>> bs_array(4,:,:,:)=dipole(az,-45);
>> antpar.BsGainPattern=bs_array;
% feed positions (in wavelengths), two dual-polarized elements
>> antpar.BsElementPosition=[0 0 10 10];
>> az=antpar.MsGainAnglesAz;
>> ms_array=zeros(2,2,1,length(az));
% MS array is a “cross-dipole”, i.e. with V and H feeds
>> ms_array(1,:,:,:)=dipole(az,0);
>> ms_array(2,:,:,:)=dipole(az,90);
>> antpar.MsGainPattern=ms_array;
>> antpar.MsElementPosition=[0 0];  % dual-polarized elements
>> [H delays]=scm(scmpar,linkpar,antpar);
>> mean(abs(sum(H,3)).^2,4)  % mean narrowband power of the 2x4
% system

ans =

    0.5977    0.5472    0.5970    0.5466
    0.2887    0.3138    0.2888    0.3139

% Note that narrowband matrix power is not one because of the
% antenna definition and random xpds. Change to uniform pattern,
% and eliminate the effect of XPD ratios on the power
% normalization.
>> scmpar.XpdIndependentPower='yes';
>> H=scm(scmpar,linkpar,antparset);  % use default antennas
>> mean(abs(sum(H,3)).^2,4)   % mean narrowband power is now two
% because mean powers of polarizations are summed. The default
% antennas have unit gain on both polarizations.

ans =

    2.0392    2.0392
    2.0410    2.0408

>> scmpar.ScmOptions='los';   % change to los option
>> H=scm(scmpar,linkpar,antparset);
>> mean(abs(sum(H,3)).^2,4)

ans =

    1.0004    1.0004
    1.0002    1.0002

% the mean matrix element power is one for all other options
% with the default antenna parameters
```

7.3 Another example about polarization option and xpd

```
% Let's illustrate the xpd with a 2x2 system
>> clear all
>> scmpar=scmparset;linkpar=linkparset;antpar=antparset;
>> scmpar.NumTimeSamples=1e5;
>> scmpar.NumPaths=1;   % frequency-flat channel
>> g=zeros(2,2,1,90);
% create a two-element array with one V and H feed (ideal xpd)
>> g(1,1,1,:)=1; g(2,2,1,:)=1;
% the channel matrix for the system is [VV VH;HV HH]
>> antpar.MsGainPattern=g;antpar.BsGainPattern=g;
>> scmpar.ScmOptions='polarized'; scmpar.AnsiC_core='yes';
>> scmpar.LookUpTable=-1;     % to speed up computation
>> scmpar.XpdIndependentPower='yes';
>> [H delays full]=scm(scmpar,linkpar,antpar);
>> P=mean(mean(abs(sum(H,3)).^2,4),4)

P =

    0.8269    0.0814
    0.1732    0.9187

% compare measured and theoretical, xpd1=VV/HV, xpd2=HH/VH
>> [P(1,1)/P(2,1) P(2,2)/P(1,2); full.xpd]

ans =

    4.7743   11.2893
    4.7741   11.2878
```

8. Licensing and how to cite the work

The software is licensed under the GNU General Public License [GPL]. Basically, you can use the software for any purpose, provided that any programs or modifications you make and distribute are also licensed under the GNU GPL. See the license.txt file included in the distribution package.

Absolutely no guarantees or warranties are made concerning the suitability, correctness, or any other aspect of these MATLAB and ANSI-C routines.

If you use the channel model software, or its modified version, in scientific work you can cite this report as follows (IEEE style):

J. Salo, G. Del Galdo, J. Salmi, P. Kyösti, M. Milojevic, D. Laselva, and C. Schneider. (2005, Jan.) MATLAB implementation of the 3GPP Spatial Channel Model (3GPP TR 25.996) [Online]. Available: http://www.tkk.fi/Units/Radio/scm/

Below is a general BiBTeX entry for citing this work.

```
@MISC{WinnerScmImplementation,
author =          "Jari Salo and
                  Giovanni {Del Galdo} and
                  Jussi Salmi and
                  Pekka Kyösti and
                  Marko Milojevic and
                  Daniela Laselva and
                  Christian Schneider",
title =           "{MATLAB} implementation of the {3GPP Spatial
                  Channel Model (3GPP TR 25.996)} ",
howpublished =    "On-line",
year =            "2005",
month =           jan,
note =            "http://www.tkk.fi/Units/Radio/scm/",
}
```

Below is a BiBTeX entry that can be used with the IEEEtran.bst style file.

```
@ELECTRONIC{WinnerScmImplementationIEEETranbst,
author =          "Jari Salo and
                  Giovanni {Del Galdo} and
                  Jussi Salmi and
                  Pekka Kyösti and
                  Marko Milojevic and
                  Daniela Laselva and
                  Christian Schneider",
title =           "{MATLAB} implementation of the {3GPP Spatial
                  Channel Model (3GPP TR 25.996)}",
year =            "2005",
month =           jan,
url =             "http://www.tkk.fi/Units/Radio/scm/",
}
```

9. Acknowledgement

The WINNER project is acknowledged for supporting this work. Comments of Mats Bengtsson are gratefully acknowledged. Special thanks go to Per Zetterberg and Achilles Kogiantis for their help in interpreting [3GPP03].

10. References

[WIN] WINNER project home page, https://www.ist-winner.org/

[3GPP03] "Spatial channel model for Multiple Input Multiple Output (MIMO) simulations", 3GPP TR 25.996 V6.1.0 (2003-09)

[GSL] GSL – GNU Scientific Library, http://www.gnu.org/software/gsl/

[GPL] The GNU General Public License (GPL), http://www.gnu.org/copyleft/gpl.html