Project Number:	IST-4-027756
Project Title:	Wireless World Initiative New Radio – WINNER II
Document Type:	I (Internal)

Document Number:	
Document Title:	**Matlab SW documentation of WIM2 model**
Source Activity:	<insert>
Editor:	Milan Narandžić, Pekka Kyösti
Authors:	Milan Narandžić, Pekka Kyösti, Lassi Hentilä, Martin Käske
Status / Issue:	1.0
Date Last changes:	19.08.2008
File:	WIM2 Matlab documentation V1.0.doc

Abstract:	This is intended for the abstract

Keywords:

Document History:	
19.08.08 ver 0.1	(Pekka) Document created by modifying Phase I Matlab SW Documentation of the WIM model.doc (Authors: Jari Salo, Pekka Kyösti, Daniela Laselva, Giovanni DelGaldo)
9.11.07 ver 0.2	(Pekka) Document updated to Phase II final model
13.11.2007 ver 0.3	(Pekka) Input parameter tables refined.
09.04.2008 ver 0.4	(M. Narandzic) Integration of 3D-Antenna-Array description into this document. Major changes.
19.08.2008 ver 1.0	(Lassi) Final version. Minor changes.

Table of Contents

1. Introduction and scope

The channel modeling work package WP1 provides a Matlab implementation of the WINNER Phase II Model (referred as WIM this onward). The main purpose of the WIM channel models is to generate a radio channel realisations for a link and system level simulations. The implementation is based on the earlier implementations of WINNER II interim model [2], WINNER I (interim) model [4], 3GPP/3GPP2 Spatial Channel Model (SCM) [5] and SCM(E) Extension models. For details concerning the WIM model, please refer to [1]. This document describes the SW implementation structure of WIM, its input/output interface and its functionality.

The channel model takes the user defined parameters, the MIMO radio link parameters and antenna parameters described in [1] as an input. 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 the given radio links. In addition, the randomly drawn channel parameters for each link will be given as an output.

The channel convolution and other related operations are beyond the scope of the implemented channel model.

2. Installation

The WIM package installs as a MATLAB mini-toolbox. Unzip the files in their own directory, e.g. 'winner'. Add the directory to MATLAB path. Type 'help winner' at MATLAB command to get started.

The package includes the following modules:

```
% WINNER Phase II channel model
% Version 1.2, Apr. 09, 2008
%
%% Channel model functions
%   wim                        - WINNER channel model (D1.1.2)
%     wim_core                 - Channel coefficient computation for a geometric channel model
%       scm_mex_core.m         - SCM computations
%   scenpartables              - Set WIM parameters for WINNER scenarios
%   pathloss                   - Pathloss models for 2GHz and 5GHz
%
%% Helper functions for model Initialization
%   wimparset                  - Model parameter configuration for WIM
%   layoutparset               - Layout parameter configuration for WIM
%
%% Determination of link-level parameters
%   layout2link                - Computes and converts layout to link parameters. The following
helper functions are used:
%     StationDistXY            - Distance between stations in XY plane
%     StationDirectionXY       - Link angular direction in Global-Coordinate-System, w.r.t Y-axis
%     StationVelocityXY        - MS velocity
%
%% Utility functions
%   StationNumElements         - Retrieve number of array elements
%   NTlayout                   - Visualisation of network layout
%
%% Generation of structural parameters
%   generate_bulk_par          - Generation of RANDOM WIM bulk parameters
%     LScorrelation.m          - Correlation of Large-Scale-Parameters (LSP)
%     LOSprobability.m         - Probability of LoS condition
%     offset_matrix_generation.m - Generation of fixed angle offsets
%     ScenarioMapping.m        - Retrive WINNER scnario label
%     struct_generation.m      - Manipulation with bulk_parameters
%   fixedAoas.m                - FIXED AoAs for CDL model
%   fixedAods.m                - FIXED AoDs for CDL model
%   fixedPdp.m                 - FIXED PDP for CDL model
%
%% 3D Antenna Array Model functions
%   AntennaArray.m             - 3D-AA model construction
%     ArrayPreprocess.m        - construction w. element field-pattern rotatation in ACS
%   AntennaResponse.m          - Computation of array response
%
%   Aperture_Calc.p            - EADF representation of radiation pattern
%     BP2Aperture.m            - Interface toward Aperture_Calc.p for 2D field pattern
%     BP2Aperture1D.m          - Interface toward Aperture_Calc.p for 1D field pattern
%     G_Calc3D_simple.p        - Calculation for 2D field pattern (3D-AA)
%     G_Calc1D_simple.p        - Calculation for 1D field pattern
%   antenna_pol_vect.m         - Computation of polarization vectors
%
%   mycart2sph.m               - Transfmation of carthesian into spherical coordinates
%   mysph2cart.m               - Transfmation of spherical into carthesian coordinates
%   rotate_vector.m            - 3D rotation
%   unrotate_vector.m          - 3D inverse rotation
%
%% Miscellaneous functions
%   cas                        - Circular angle spread (3GPP TR 25.996)
%   ds                         - RMS delay spread
%   dipole                     - Field pattern of half wavelength dipole
%
%% Examples
%   example_EADF_approx.m      - Representation of radiation pattern
%   example_syntetic_arrays.m  - Array struct generation
%   example_channel_matrix.m   - Generation of channel matrix
```

Note! The code is created and tested on Matlab versions 6.5, 7.1 (R14) and 7.5 (R2007b). Some other version might cause problems.

3. Model features

3.1 Basic features

WINNER MIMO radio channel model enables system level simulations and testing. This means that multiple links are to be simulated (evolved) simultaneously. System level simulation may include multiple base stations, multiple relay stations, and multiple mobile terminals as in Figure 3.1. Link level simulation is done for one link, which is shown by blue dashed ellipse. The short blue lines represent channel segments where large scale parameters are fixed. System level simulation consists of multiple links. Both link level and system level simulations can be done by modelling multiple segments, or by only one (CDL model).

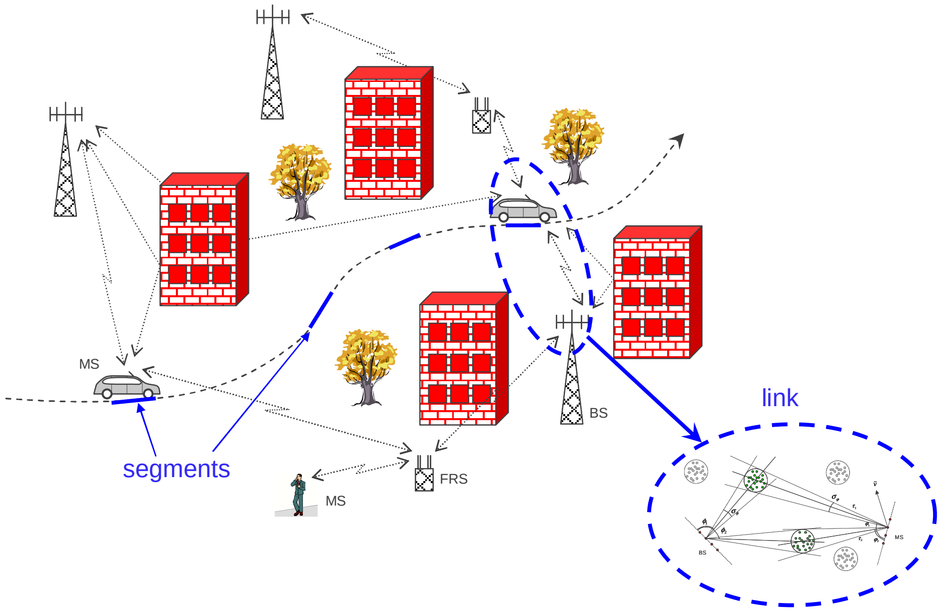


Figure 3.1. System level approach, several segments (drops).

A single link model is shown in Figure 3.2. The parameters used in the models are also shown in the figure. Each circle with several dots represents scattering region causing one cluster. The number of clusters varies from scenario to another.

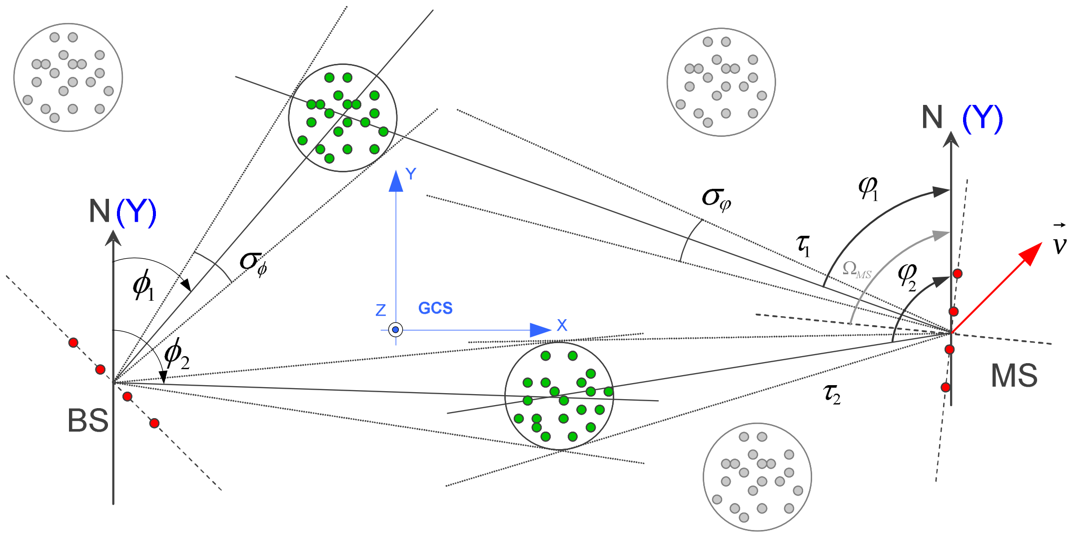


Figure 3.2. Single link.

The number of paths is scenario dependent and varies from 8 to 24 (see [1]). The number of rays (subpaths) is fixed to 20 for all scenarios.

3.2 3D-Antenna-Array model

3D-Antenna-Array model allows complete 3D description of polarization, directional filtering and spatial displacement. It supports arbitrary array geometries (not limited to Uniform Linear Arrays). Each mobile or base station can use different antenna array, where radiation pattern of each element can be independently defined.

Detailed reasoning for chosen antenna array representation is given in [1]. In order to fully describe positions and directivity of antenna elements in 3D space, 2 coordinate systems will be used (Figure 3.3):

- GCS – Global Coordinate System (used to define radio-network system layout, and as a reference system for polarization)
- ACS – Array Coordinate System - describes array geometry and rotated radiation patterns of antenna elements.

Relative position of CS are described by orientation and translation in 3D space.

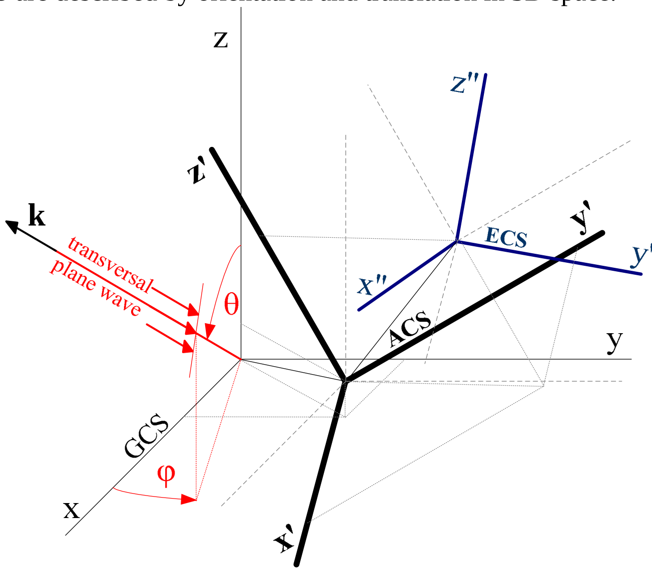


Figure 3.3 Coordinate system definitions: GCS, ACS and ECS

Usage of the third co called Element-Coordinate-System (ECS) to represent radiation pattern of each antenna element is not suitable since it increases simulation complexity. In [1] it was shown that representation of field patterns (of all elements in array) in ACS requires lower complexity.

Additionally, if field patterns are represented with its Effective-Aperture-Density-Functions (EADF, 2D-Fourier transform of the field pattern) that will reduce memory storage requirements and provide effective way of interpolating the field pattern for arbitrary (θ, φ).

It was therefore concluded that most suitable representation for element field patterns is EADF defined for all elements in the array in respect to common ACS.

3.2.1 Conceptual changes related to GCS and 3D-AA

- All angles used in simulations (AoA/AoD) are now defined relative to the GCS (Figure 3.2) and not to the arrays broadside. This change is following evolution of link-level to system-level model: single station can be considered in multiple links – therefore same reference is necessary.
- All field patterns are polarized patterns. In the case where only single polarization of field pattern is known, the other polarization dimension has to be filled with zeros. This is necessary to support arbitrary array rotation.

3.3 CDL model option

Deliverable D1.1.2 [1] contains fixed channel models called Clustered Delay Line (CDL) models with tabulated spatio-temporal channel parameters. They are fully deterministic despite random initial phases of the rays. Doppler is not explicitly defined, because it results from the angular properties. These models might be useful e.g. in the simulation system calibration simulations. With CDL models there is no randomness in the large scale parameters.

CDL models can be selected by setting parameters:
`wimpar.FixedPdpUsed =‘yes’` and `wimpar.FixedAnglesUsed =‘yes’`.

3.4 Supported propagation scenarios

For details about the scenarios definition see [1], sect 2.3. Spatial channel model parameters, derived mostly based on the WINNER measurement campaigns [1], characterize each channel scenario. The scenarios-dependent parameters are currently supported at center frequency of 2-6 GHz and at bandwidth of 100 MHz.
Fixed parameters, defining every parameterized WINNER scenario are provided in the function ScenParTables.m.

Table 3.1. Supported propagation scenarios.

Scenario	Definition	LOS/ NLOS	Mob. km/h	Frequency (GHz)	CG	Note
A1 In building	Indoor small office / residential	LOS/ NLOS	0–5	2 - 6	LA	
A2	Indoor to outdoor	NLOS	0–5	2 - 6	LA	AP inside UT outside. Outdoor environment urban
B1 Hotspot	Typical urban micro-cell	LOS NLOS	0–70	2 - 6	LA, MA	Typical distance ranges. Actual ranges depend on frequency and antenna heights
B2	Bad Urban micro-cell	LOS/ NLOS	0–70	2 - 6	MA	Same as B1 + long delays
B3 Hotspot	Large indoor hall	LOS	0–5	2 - 6	LA	
B4	Outdoor to indoor. -Outdoor typical urban. -Indoor A1	NLOS	0–5	2 - 6	MA	B1 or C2 to the wall/window.
B5a Hotspot Metropol	LOS stat. feeder, rooftop to rooftop	LOS	0	2 - 6	MA	Same channel model for hot spot and metropol.
B5b Hotspot Metropol	LOS stat. feeder, street-level to street-level	LOS	0	2 - 6	MA	Typical distance ranges. Actual ranges depend on frequency and antenna heights
B5c Hotspot Metropol	LOS stat. feeder, below-rooftop to street-level	LOS	0	2 - 6	MA	Typical distance range. Actual range depends on frequency and antenna heights
B5d Hotspot Metropol	NLOS stat. feeder, above rooftop to street-level	NLOS	0	2 - 6	MA	Extended C2
B5f	Feeder link BS -> FRS. Approximately RT to RT level.	LOS/ OLOS/ NLOS	0	2 - 6	WA	Desired link: LOS or OLOS, Interfering links: LOS/(OLOS) /NLOS FRS -> MS = B1*
C1 Metropol	Suburban	LOS/ NLOS	0–120	2 - 6	WA	
C2 Metropol	Typical urban macro-cell	NLOS	0–120	2 - 6	MA WA	
C3	Bad Urban macro-cell	LOS/ NLOS	0–70	2 - 6	-	Same as C2 + long delays
D1 Rural	Rural macro-cell	LOS/ NLOS	0–200	2 - 6	WA	
D2	a) Moving networks: BS – MRS, rural	LOS	0 –350	2 - 6	WA	Very large Doppler variability.
	b) Moving networks: MRS – MS, rural	LOS / OLOS/ NLOS	0 – 5	2 - 6	LA	

4. Model usage

The model is used for generation of multidimensional channel matrix H, containing time-variant Channel-Impulse-Responses (CIR) between all transmitter and receiver antenna combinations of MIMO system. Generation of matrix is performed by MATLAB function wim.m. Apart from the matrix H full set of parameters characterizing physical propagation channel can be retrieved as well. Detailed description of input and output arguments of wim.m function is given in section 5.2. In this section preparation of network layout description will be discussed, and particularly construction of antenna array simulation model. Additionally, some examples will be given for both usage of model in simulation, and construction of the antenna array models.

4.1 Construction of antenna array model (Preprocessing phase)

Although WINNER channel model is constructed as antenna independent model, antenna array model is necessary to obtain signals at the output of the radio-channel. This model is deterministic and permanent - from the viewpoint of the simulation, and can be created independently from channel model simulations. Therefore certain type of array requires only single construction, which should be performed independently from WIM simulations - in pre-processing phase. It is not good strategy to construct arrays each time when WIM is used, instead defined antenna arrays should be stored and retrieved when needed. This becomes particularly important when large number of operations is performed in this phase, i.e. when 3D field patterns are rotated from ECS to ACS.

WINNER phase II channel model uses structure *Array* (subsection 5.2.2.2) for representation 3D-AA model. For construction of *Array* structure MATLAB function AntennaArray.m is provided. If one would like to verify created 3D-AA model, this can be performed by AntennaResponse.m function.

In order to make *Array* it is necessary to define its geometry (positions and rotation of elements), and to provide element field patterns. The arguments provided to AntennaArray .m are always processed in predefined way: first array geometry is created and after that field patterns are assigned.

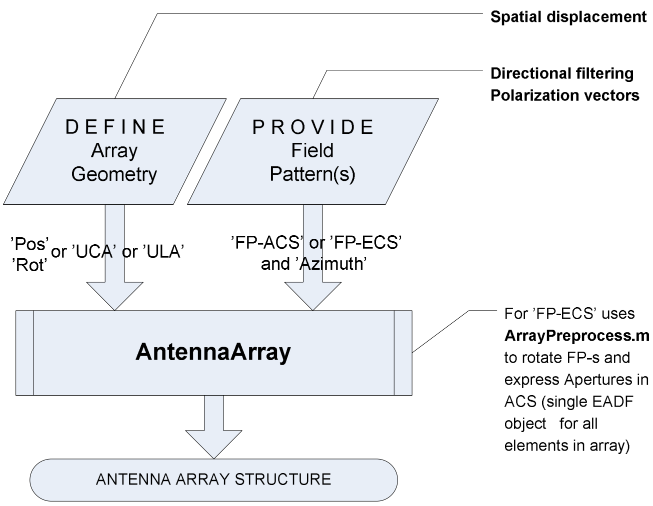


Figure 4.1 Construction of antenna array

Supported options for a creation of *Array* structure are illustrated in the following examples:

```
ArrayGeometryExample1=AntennaArray('Pos',Position,'Rot',Rotation, 'FP-ECS',
                        FieldPattern);
ArrayGeometryExample2=AntennaArray('UCA',N,r, 'FP-ECS', FieldPattern);
ArrayGeometryExample3=AntennaArray('ULA',N,d, 'FP-ACS', FieldPattern);
```

4.1.1.1 Array Geometry (AG)

Geometry can be defined explicitly using 'Pos' and 'Rot' arguments followed by ELNUMx3 matrix, where ELNUM is the number of elements. Alternatively, common array types 'UCA' and 'ULA' could be defined with few parameters only. For UCA elements are placed starting from x-axis (phi=0) every $\Delta\varphi = \frac{2\pi}{N}$, and n[th] element is rotated for $(n-1)\Delta\varphi$ in positive mathematical direction. ULA elements are placed along x-axis is such a way that the center of the array is at [0;0;0] (for even N there is no antenna element at [0;0;0]).

Default geometry:

If there are no parameters defining geometry default is single antenna positioned at centre of ACS, without rotation.

Table 4.1. Input parameters for AntennaArray.m function.

Parameter Name	Group	Definition	Default value	Unit	Note
Position	AG	ELNUMx3 matrix, where n-th row contains [x, y, z] position of n-th antenna element in ACS.	-	m	
Rotation	AG	ELNUMx3 matrix, where n-th row contains [Rot_x;Rot_y;Rot_z] rotation of n-th antenna element afound axes of ACS.	$\mathbf{0}_{Nx3}$	rad	
ELNUM	AG	Number of physical antenna elements in array. Used as 1st argument for 'UCA' and 'ULA' options. Implicitly defines the first dimension of Position, Rotation and FieldPattern.	-	-	
r	AG	Radius of UCA.	1	m	Optional
d	AG	Distance between antenna elements in ULA.	1/ELNUM	m	Optional
FieldPattern	FP	4D array containing field patterns of antenna elements. The dimensions of FieldPattern are [ELNUM POL EL AZ] = SIZE(FieldPattern)			Same format is used for both 'FP-ECS' and 'FP-ACS' options
POL	FP	Number of polarizations used to characterize FieldPattern. The first dimension in FieldPattern is used to store vertical polarization, the second for horizontal. Missing polarization dimensions of FieldPattern are substituted with zeros.	2		
EL	FP	Number of equidistant FieldPattern samples taken over elevation angle.	-	-	
AZ	FP	Number of equidistant FieldPattern samples taken over azimuth angle.	-	-	
Elevation	FP	Vector of elevation angles corresponding to 3rd dimension of FieldPattern.	-	deg	
Azimuth	FP	Vector of azimuth angles corresponding to 4rd dimension of FieldPattern.	-	deg	

4.1.1.2 Field pattern (FP) representation

The field patterns of individual array elements are described using the EADF[1] defined in ACS. This was done because EADF has proven to be superior in terms of memory requirements and interpolation errors. EADF representation is created from a sampled field pattern (FP) that is provided as an input to *AntennaArray*. The number of provided field patterns (#FP) for 'FP-ACS' and 'FP-ECS' should be equal to number of elements in array (ELNUM), however:

a) if #FP is equal to 1, same FieldPattern will be apply to all antennas.
b) if 1<#FP< ELNUM the error will be issued
c) if #FP> ELNUM take only first N.

NOTE: When single FP defined in ECS is shared by all elements, EADF calculation will be performed twice: i) calculation of CommonAperture in ECS and ii) during FP rotation to ACS.

Reference coordinating system

[1]Effective Aperture Distribution Function

In order to distinguish between FPs that are expressed in ECS and ACS it was necessary to define different argument types: ‘FP-ECS’ and ‘FP-ACS’. However, in both cases input argument FP has the same structure.

Sampling grid

In order to calculate 2D Fourier transformation of FP equidistant sampling points are required. If non-uniform samples are provided, additional interpolation (e.g. spline) of FP, before calculating its EADF, is necessary. This may introduce significant errors in the FP representation and therefore we would assume that uniform sampling is always used. This would mean that the azimuth angles must be equidistant from -180 to (but not including) 180 degrees.

Polarization

Since 3D-Antenna-Array-Model generally allows arbitrary rotation of antenna elements, it is necessary to calculate projections to referent polarization vectors in GCS. Using of wimpar.PolarizedArrays=’no’ option in wimparset.m (WIM1) with 3D-Antenna Array is not directly applicable since the representation of the polarization for single-polarized but rotated antenna still requires two orthogonal reference polarization vectors – what is equivalent to previous polarized option (wimpar.PolarizedArrays=’yes’). Otherwise, only single (vertical or horizontal) projection of polarization vector will be handled, what is obviously wrong. Therefore control parameter wimpar.PolarizedArrays can not be used (in general) to reduce complexity – channel coefficients will be always based on polarized FP (after rotation).

Default field pattern:

If neither ‘FP-ACS’ nor ‘FP-ECS’ are defined use isotropic, vertically polarized antenna with XPD=∞.

4.1.1.3 Examples

Helper function `arrayparset.m` is provided to illustrate construction of `Arrays`. It shows construction of ULA 2/4/8 and UCA 4/8 with synthetic field pattern defined in ECS. Slanted dipole and isotropic radiation patterns are used for that purpose.

This section provides some additional examples and explanations about antenna array model creation.

Example 1:

Geometry: Unifom-Circular-Array (UCA) with 16 elements, and radius of 5 cm

Field pattern for each element is defined in its own ECS. This happens if each array element is measured separately or if analytical description (i.e. mathematical expression/function) is used to define of the patterns.

```
Array=AntennaArray('UCA',16,0.05,’FP-ECS’,FP);
```

FP should have dimensions $\begin{bmatrix} ELNUM & POL & EL & AZ \\ 16 & 2 & 1 & \#AS \end{bmatrix}$:

- First dimension should fit to number of elements required for UCA geometry, alternatively can be 1 what would mean that same FP will be applied to all array elements.
- Both polarizations must be provided for FP. It only single polarization is available other dimension must be filled with zeros.
- Since model is still 2D only single (zero) elevation is used, and third dimension of FP is equal to 1.
- Typicaly FP is sampled once per angular degree, giving total number of azimuth samples #AS= 360. It is assumed that FP samples are taken from uniform grid. If optional ‘Azimuth’ argument is provided it should have #AS elements.

Example 2:

Custom Geometry is defined with `Position` matrix.

Field pattern for all elements are defined in the ACS. When field patterns of array elements are measured after array creation it is preferable to define same measurement grid in ACS for all elements.

```
Array=AntennaArray('Pos',Position,’FP-ACS’,FP,’Azimuth’,Az);
```

Comments:

- Number of rows in `Position` matrix will determine number of elements in array, and each row contains x, y and z position (in [m]) of antenna element inside array.
- Since ‘Rot’ argument is not defined, there will be no rotation: ECS axis will be align with ACS axis.

- FP argument to *AntennaArray* function has equivalent form like in previous example. Difference is that in this case internal rotation of FRs from ECSs to ACS is not longer necessary.
- Az vector must contain equidistant azimuth angles from -180 to (but not including) 180 degrees. Length of this vector must correspond to 4^{th} dimension of FP.

4.2 Description of the network layout

WIM implementation supports a multi-base station and multi-mobile station network layout. The network layout includes information about: the number and locations of MSs and BSs in the GCS; the number of sectors in a BS (in case of a multi-cell network); the array broad side orientations at both MS and BS; the coupling of an active radio link from a MS to a certain sector of a BS (or vice-versa); and the directions of the MSs movement.
More detailed guidelines for system-level simulations could be found in [1]. That document describes handover, multi-cell, multi-user, multi-hop and relaying simulations.

4.2.1 Construction of semi-random layout

Supported version of layoutparset.m is used to define type of the antenna array that is assigned to each station, and therefore layout is not completely random. Desired number of K links is formed by random BS-MS pairing (number of BSs and MSs is given implicitly during antenna selection). Function layoutparset.m. generates random positions for all stations, and assigns random scenario and propagation conditions to all links. MSs and BSs locations are randomly generated within e.g. the 500x500m^2 cell area. Default height of 32 m is used for BSs and 1.5 m for MSs.

```
layoutpar=layoutparset(MsAAIdx, BsAAIdxCell, K, Arrays)
```

Table 4.2. Input parameters for layoutparset.m function.

Parameter Name	Group	Definition	Default value	Unit	Note
Arrays	-	Vector of *Array* definitions (see subsection 5.2.2.2) containing all array types that should be used in the simulation	-	-	
MsAAIdx	-	Vector of `Arrays` indices; single `Array` index defines type of AA used by that MS.	-	-	The total number of MS `MsNum = length(MsAAIdx);`
BsAAIdxCell	-	Cell array of `Arrays` indices, where single cell contains vector that describe multi-sector-BS. Vector entries are `Arrays` indices, showing wich of available AA is assigned to particular sectors. Multi-sector-BSs could have arbitrary number of sectors. Indexing of sectors will be taken according to the order of their occurrence in vector.	-	-	The total number of multi-sector-BS `BsNum = length(BsAAIdxCell);` The total number of sectors (one-sector-BS) is: `BsSectNum = length([BsAAIdxCell{:}]);`
K	-	Number of links formed by random BS-MS pairing	-	-	

WIM model assumes that different sectors of multi-sector-BS are closely located and therefore links between MS and different sectors exhibit full correlation: the same LoS/NLoS conditions and LSPs (AS, DS) values. Links from mobile to different sectors are still not identical due to the specific array orientation and directional filtering, and because they use different low-level parameters (MPC delays, angles etc.). The second assumption is that sectors of the other BSs are located “very far away”, so that there is no considerable correlation between links from single MS toward sectors belonging to different BSs.

4.2.1.1 Example

If e.g. *MsAAIdx* and *BsAAIdxCell* is defined like this:

```
>> MsAAIdx = [1 1 2 3];
>> BsAAIdxCell = {[1 3]; [2]; [1 1 2]};
```

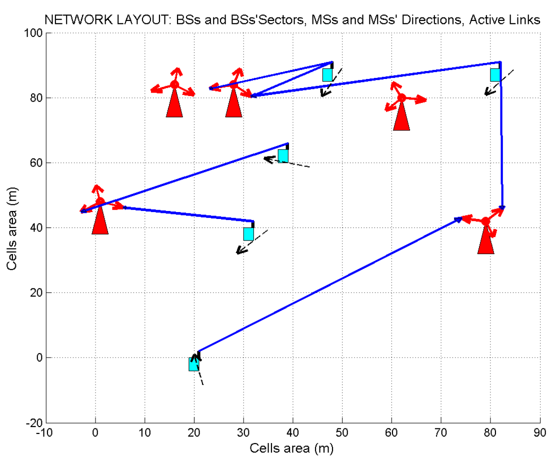
It means that a total of 4 MS are present: The first two will use array type defined in `Arrays(1)` the third MS will use `Arrays(2)` and the fourth `Arrays(3)`. Created layout will have 3 multi-sector-BSs. The first of them has two sectors, that are using `Arrays(1)` and `Arrays(3)`, the second sector is one-sector-BS with `Arrays(2)`, while the third has three sectors: two of them are using `Arrays(1)` and one is using `Arrays(2)`. Please check the file *ArraysExample.m* for a more complex example.

4.2.2 Manual editing

It is also possible to define the network layout manually by directly editing all or part of the layout parameters in the `layoutpar` structure. Starting from the previous semi-random layout wanted links (i.e. (BS,Ms) pairs) could be defined by modifying LAYOUTPAR.Pairing. Position and orientation of each station could be manually adjusted using LAYOUTPAR.Station.Pos/Rot parameters. Change of per-link scenario and propagation conditions can be modified through LAYOUTPAR.ScenarioVector/PropagConditionVector. For more details please refer to section 5.2.2 where these parameters are explained.

4.2.3 Layout setup visualization

A rudimental visualization of the network layout is implemented as helper utility in order to visualize the layout of the generated channel; the network layout with 5 BSs and MSs, and 7 active links looks like as in the following figures. In the figures red arrows denote the BS sector array orientation, black arrows denote the MS direction of motion, and blue arrows denote the active (modelled) links.



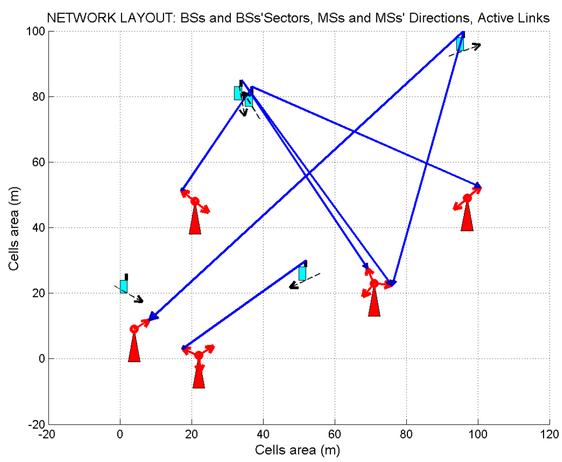


Figure 4.2: Examples of the network layout: a) 5 BSs (three sectors in any BS), 5 MSs and 7 active links; b) 5 BSs (number of sectors varies from 1-3 in different BSs), 5 MSs and 7 active links

The utility function NTlayout.m is used the following way:

```
>> NTlayout(layoutpar)
```

By setting the MSs and BSs coordinates and selecting the Pairing matrix properly it is possible to simulate many system-level cases (see next paragraph).

4.3 Channel matrix generation

4.3.1 Simple initialization

```
%% Matrix generation for 10 MS-BS links (semi-random layout)
Arrays=arrayparset;
```

```
MsAAIdx = [1 1 2 3];
BsAAIdxCell = {[1 3]; [2]; [1 1 2]};
layoutpar=layoutparset(MsAAIdx, BsAAIdxCell,10,Arrays);

wimpar= wimparset;

H=wim(wimpar,layoutpar);
```

4.3.2 Initialization with arbitrary parameters

Instructions for the generation of arbitrary network layout and antenna array models are given in previous sections. Control of the general model features (for all links), simulation parameters and output formats could be adjusted in wimpar structure. In general, wimpar structure can be created from the scratch, or one can initialize parameters to default values (by calling wimparset.m) and then modify selected parameters.

```
%% Matrix generation with modified parameter settings
% Setting and modifying wimpar and layoutpar input stuctures
Arrays=arrayparset;

MsAAIdx = [1 1 2 3];
BsAAIdxCell = {[1 3]; [2]; [1 1 2]};
layoutpar=layoutparset(MsAAIdx, BsAAIdxCell,10,Arrays);
layoutpar.PropagConditionVector=zeros(1,10);  % (NLOS=0/LOS=1)
layoutpar.ScenarioVector=3*ones(1,10);  % B1 scenario

wimpar=wimparset;
wimpar.NumTimeSamples=1000;     % 100 time samples per link

% Generate channel realisations
[H1,delays,out]=wim(wimpar,layoutpar);
```

4.3.3 Initialization of the structural model parameters

This kind of initialization assumes complete knowledge of model structure. This option is provided to enable consequtive calls of wim.m functions, without (default) random initialization of structural parameters. This means that structural parameters obtained after one simulation run could be used to initialize new run, preserving in that way previous channel conditions – what somehow means continuation of the previous simulation run. This enables performing of seamless channel simulation is several simuation runs. In example bellow it is assumed that input argument `out` is provided during some previous call of wim.m, e.g. by example from the previous section.

```
% using final conditions as initial conditions in next function call
[H2,delays,out]=wim(wimpar,layoutpar,out);
```

5. Technical description of the SW

5.1 SW high level structure

The high-level WIM implementation structure is shown in the block diagram given in Figure 5.1. It is assumed that the user mobility model, which is not specified in [1] is external to the channel matrix generation routine. The path loss model is also implemented as a separate user-supplied function. The default path loss function, complying with [1], is [pathloss.m]. Interpolation of antenna field patterns is also required since AoD/AoAs can be any values over (-180,180) degrees. In WIM, field pattern interpolation is based on EADF representation - function [ArrayResponse.m].

The WIM computation consists of two main parts:

- the random user parameter generation and
- the actual channel matrix computation.

Input and output arguments are defined in more detail in the next section.

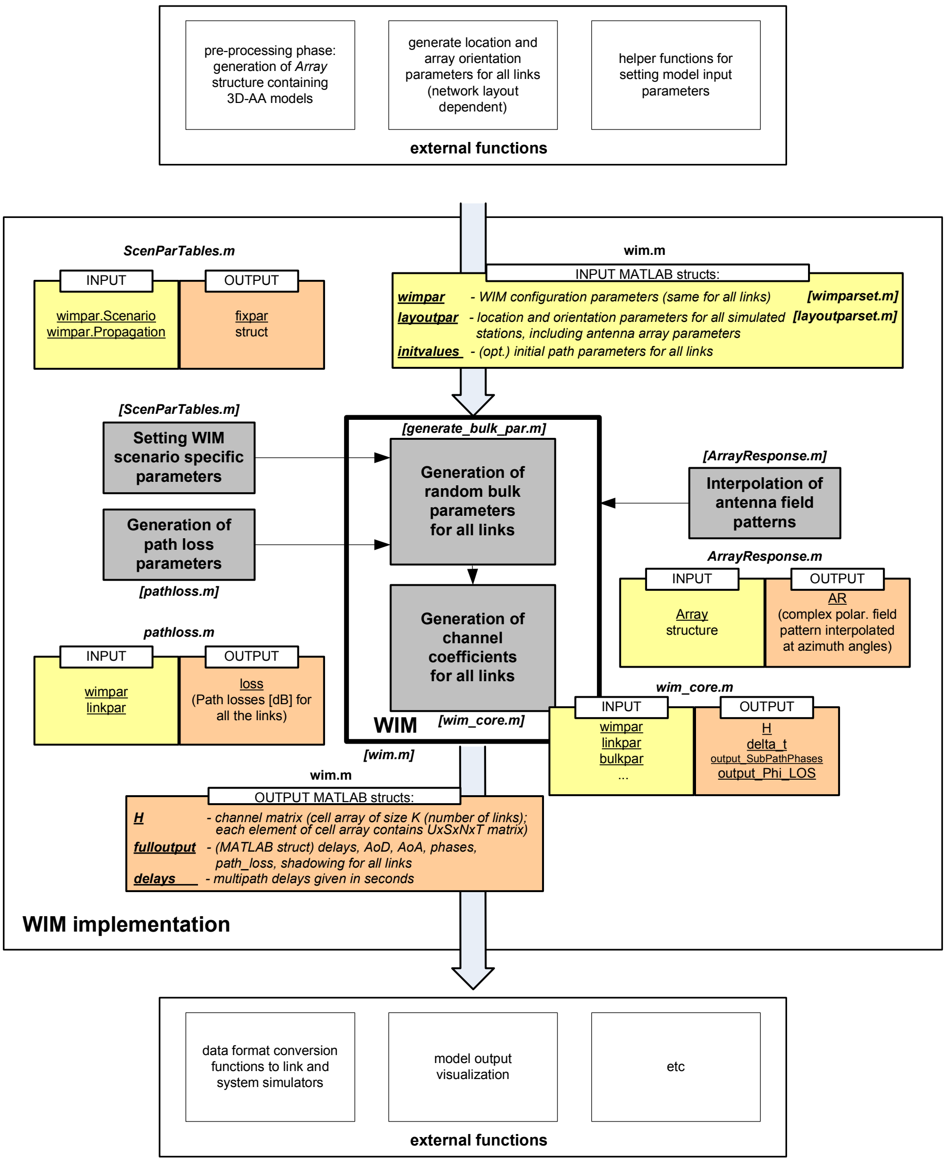


Figure 5.2. High-level description of the WIM computation. The actual WIM model is in the box labeled 'WIM'.

5.2 Model input/output interface

The full syntax for the WIM function is ([.] indicate optional arguments):

[H, [DELAYS], [FULL_OUTPUT]] = WIM (WIMPAR, LAYOUTPAR, [INITVALUES]).

Table 5.1 Short overview of input and output aruments for WIM function

Argument name	Type	Description	Helper functions for structure generation:	Note
WIMPAR	input	General simulation parameters.	wimparset.m	
LAYOUTPAR	input	Defines position of terminal stations, their assigned antenna arrays and gives links of interest for simulation.	layoutparset.m	The function *linkparset* should not be used anymore. Instead WIM should always be initialized with a *layoutpar* structure. This is necessary to keep consistency of data defined in *layoutpar* structure. Internally *layout2link* is used to create missing link parameters.
INITVALUES	input	Parameters of the propagation channel. When this parameter is given WIM does not generate the channel parameters randomly, but uses the supplied initial channel values.	-	Optional
H	output	MIMO channel 5D-matrix is collection of time-variant CIRs (=f(t,τ)) between all (Tx,Rx) pairs, for all links defined in LINKPAR.	-	Calculated for a specified number of time samples
DELAYS	output	Multipath delays for all links, given in [s].	-	Optional
FULL_OUTPUT	output	Stores 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 to generate time continuous channel realizations with separate function calls.	-	Optional

NOTE: All input and output arguments of WIM function are MATLAB structures/cells.

5.2.1 Global simulation parameters (WIMPAR)

Global simulation parameters could be classified into two major groups:

- MDL – parameters defining model structure (also includes system dependant parameters like CenterFrequency). Change of parameters from this group causes different model behavior.
- SIM –simulation control parameters. These parameters control sampling in time and delay, parameter initialization mode (random vs. manual), and format of output parameters (e.g. inclusion/exclusion of path-loss in channel matrix H).

Table 5.2: MATLAB struct <u>WIMPAR</u> (set in wimparset.m). General channel model parameters, common for all links.

Parameter Name	Group	Definition	Default value	Unit	Note
CenterFrequency	MDL	The carrier center frequency. Center frequency affects path loss and time sampling interval.	5.25e9	Hz	

Parameter Name	Group	Definition	Default value	Unit	Note
NumTimeSamples	SIM	Number of time samples	100	-	
SampleDensity	SIM	Oversampling factor, number of time samples per half wavelength. For successful Doppler analysis, one should select SampleDensity > 1. The time sample interval is calculated from CenterFrequency and MsVelocity (see LAYOUTPARSET and LAYOUT2LINK) according to wavelength/(MsVelocity*SampleDensity). The calculated time sample interval for each link is included in the optional output argument of WIM.	2	-	
UniformTimeSampling	SIM	If UniformTimeSampling is 'yes' all links will be sampled at simultaneous time instants. In this case, the time sample interval is the same for all links it is calculated by replacing MsVelocity with MAX(MsVelocity), where the maximum is over all links. If 'no' all the links are time sampled with different rate depending on MsVelocity.	'no'	-	'Yes' setting could be useful in some system-level simulations where all simulated links need to be sampled at equal time intervals, regardless of MS speeds.
IntraClusterDsUsed	MDL	If 'yes' the two strongest clusters in power are divided in delay into three subclusters. Fixed delays are [0 5 10] ns, fixed powers are [10 6 4]/20. For details see [1], section 4.2. Number of delay tap grows by four. If 'no', the clusters are not spread in delay.	'yes'	-	
NumSubPathsPerPath	MDL	Number of rays (i.e. complex sinusoids, plane waves) per cluster. It is not possible to change this value from 20 without modifying the code.	20	-	This is a fixed value.
FixedPdpUsed	MDL	If 'yes' the power and delay parameters are not drawn randomly, but taken from the CDL parameter tables [1], table 6-1..26. In the default mode 'no', the parameters are random variables.	'no'	-	
FixedAnglesUsed	MDL	If 'yes' the angle parameters are not drawn randomly, but taken from the CDL parameter tables [1], table 6-1..26 and the random pairing of AoDs and AoAs is not used. In the default mode 'no', the parameters are random variables.	'no'	-	
TimeEvolution	MDL	If TimeEvolution='yes', the transition between adjacent channel segments is enabled. Transition from segment to segment is carried out by replacing clusters of the "old" segment by the clusters of the "new" segment, one by one. The route between adjacent channel segments is divided to number of sub-intervals equal to maximum number of clusters within the channel segments. During each sub-interval the power of one old cluster ramps down and one new cluster ramps up. Power ramps are linear. Clusters from the old and new segments are coupled based on their power. If number of clusters is different in the channel segments the weakest clusters are ramped up or down without a pair from other cluster. See [1], Sec 3.4.	'no'	-	Implem-entation is no complete. This option is not supported.
DelaySamplingInterval	SIM	DelaySamplingInterval determines the sampling grid in delay domain. All path delays are rounded to the nearest grid point. It can also be set to zero.	5e-9	sec	

Parameter Name	Group	Definition	Default value	Unit	Note
PathLossModelUsed	SIM	When PathLossModelUsed is 'no' the path losses are still computed for each link but they are not multiplied into the channel matrices. Path loss is given only as an output parameter. If 'yes', path loss is multiplied to channel matrices.	'no'	-	
UseManualPropCondition	SIM	If 'yes' the propagation condition (los/nlos) setting is defined manually in LAYOUTPARSET in PropagConditionVector. If 'no', the propagation condition is drawn from LOS probabilities in [1], table 4-7.	'yes'	-	
ShadowingModelUsed	SIM	When ShadowingModelUsed is 'no' the shadowing coefficients are still computed for each link but they are not multiplied into the channel matrices. Shadowing is given only as an output parameter. If 'yes', shadowing is multiplied to channel matrices.	'no'	-	
PathLossModel	MDL	The path loss model function name. Path loss model is implemented in a separate function, whose name is defined in PathLossModel. For syntax, see PATHLOSS. The default function is PATHLOSS, which complies with [1].	'pathloss'	-	
PathLossOption	MDL	A wall material option for A1 NLOS path loss calculation with default formulas from [1]. Options are ['CR_light', 'CR_heavy', 'RR_light', 'RR_heavy']. Here CR denotes Corridor-Room condition and RR denotes Room-Room nlos condition. Light wall loss is 5dB and heavy wall loss is 12dB. Light wall material is e.g. plaster board, heavy wall material is e.g. brick or concrete. Note, in [1], table 4-4 default A1 NLOS case is CR.	'CR_light'	-	
RandomSeed	SIM	Sets random seed for Matlab random number generators. The default value is empty. Even fixing the random seed may not result in fully repeatable simulations due to differences in e.g. MATLAB versions.	[ ]	-	see Matlab help
range	MDL	Path loss parameter for scenario B5b. In B5b the path-loss ranges 1, 2 and 3 are defined, see [1], table 6-20.	1	-	B5 specific
end_time	SIM	Observation end time for B5 scenarios time points are taken as: wimpar.TimeVector=linspace(0,wimpar.end_time,T);	1	sec	B5 specific

5.2.2 Parameters defining network layout (LAYOUTPAR)

A network layout is defined with positions of terminal stations, where each station has assigned antenna array. The coordinate system where placement of stations takes place is called Global-Coordinate-System (GCS). Given 3D-AA representation allows rotation of the antenna array (or equivalently station) around any axis of the GCS.
The monitored links are given as a collection of terminal station pairs. Propagation condition and radio environment are defined on link-level.
It is therefore possible to identify several groups of parameters:

- ENV – parameters for selection of propagation conditions and radio environment description,
- POS – parameters defining positions of terminals (also includes mobility model – in this case "MsVelocity" and "MsDirection").
- ANT – antenna array parameters.

Table 5.3. MATLAB struct LAYOUTPAR (set in layoutparset.m): network layout parameters.

Parameter	Group	Definition	Default	Unit	Note

Name			value		
Stations	POS + ANT	Parameters of the stations being included into system simulations.	see Table 5.5	-	MATLAB structure
Pairing	-	A 2xK matrix whose k-th colum contains indices of Stations constituting k-th link.	*see* layoutparset.m	-	To define pairing Stations are ordered: first all BS Sectors, than MS (1 AA is assumed per MS)
NofSect	-	A parameter defining the number of sectors in each of the BSs.		-	*see* layoutparset.m
ScenarioVector	ENV	A 1xK vector mapping scenarios to links. Scenarios are [1=A1, 2=A2, 3=B1, 4=B2, 5=B3, 6=B4, 7=B5a, 8=B5c, 9=B5f, 10=C1, 11=C2, 12=C3, 13=C4, 14=D1, 15=D2a].	ones(1,K)	{1,2, …,15}	
PropagConditionVector	ENV	A 1xK vector mapping propagation condition (NLOS/LOS) to links. If WIMPAR UseManualPropCondition = 'yes', link propagation conditions (NLOS=0/LOS=1) are defined by this vector.	zeros(1,K)	{0,1}	Possible values 0=NLOS and 1=LOS.
StreetWidth	ENV	A parameter for B1 and B2 path loss model. Average width of the streets, same for all users.	20	m	
NumFloors	ENV	A parameter for A2/B4 path loss model. NumFloor is the floor number in which the indoor MS/BS is located. E.g. in A2 scenario NumFloors is 5 if BS is located on the 5th floor. On ground floor (=street level) NumFloor = 0.	1	-	
NumPenetratedFloors	ENV	A parameter for A1 NLOS path loss model [1], table 4-4. Number of penetrated floors between BS and MS.	0	-	
Dist1	POS	Distance definition for B1 and B2 path loss model. Dist1 is a distance from BS to the "last line-of-sight point", typically street crossing, see [1], fig 4-3. Default value is NaN, which denotes random distance determination in PATHLOSS function.	NaN	-	

5.2.2.1 Link-level parameters (LINKPAR)

Based on parameters in LAYOUTPAR structure necessary link parameters are calculated. They should not be modified directly in system level simulations to prevent inconsistencies with chosen network layout. Please note that some of the link parameters are kept only to minimize changes in respect to previous versions of the source code.

Both the distance and line of sight (LOS) direction information of the radio links are calculated w.r.t. XY plane of GCS, i.e. without consideration of z dimension. The distance between the BS_i and MS_k (MsBsDistance) is

$$d_{BS_i,MS_k} = \sqrt{(x_{BS_i} - x_{MS_k})^2 + (y_{BS_i} - y_{MS_k})^2}.$$

In order to express link orientation and array broadside direction, Y axes of GCS, being aligned with North direction, is used as a reference the zero angle (Figure 5.3). The positive direction of the angles is the clockwise direction.

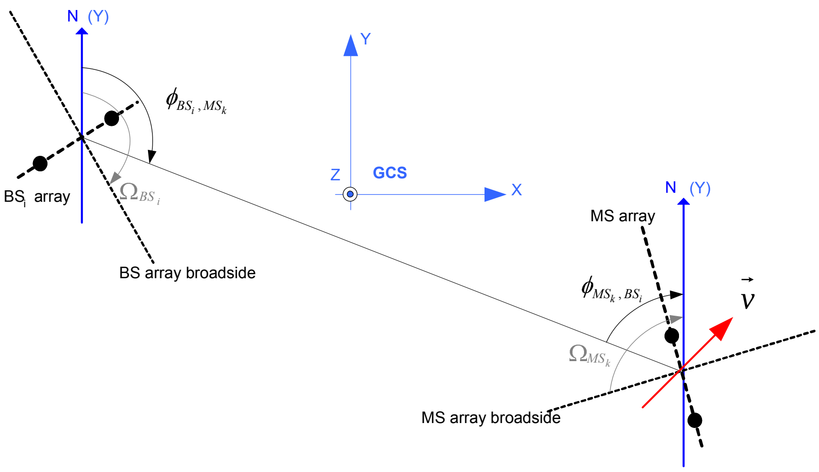


Figure 5.3: Link and BS and MS antenna array orientations.

The LOS direction from BS_i to MS_k with respect to BS antenna array broad side is

$$\theta_{BS_i,MS_k} = -\arctan\left(\frac{y_{MS_k} - y_{BS_i}}{x_{MS_k} - x_{BS_i}}\right) + 90^\circ$$

Table 5.4: Internally used MATLAB struct <u>LINKPAR</u> (calculated by layout2link.m).

Parameter Name	Group	Definition	Unit	Note
MsBsDistance	POS	A 1xK vector defining distance between BS and MS for K links.	m	calculated pairwise from *Stations* (as defined in *Pairing*) StationDistXY.m
BsHeight	POS	A 1xK vector defining BS height from ground level.	m	*Stations.Pos(3)*
MsHeight	POS	A 1xK vector defining MS height from ground level.	m	*Stations.Pos(3)*
ThetaBs	POS	θ_{BS} (see Figure 5.3)	deg	calculated pairwise from *Stations* (as defined in *Pairing*) StationDirection.m
ThetaMs	POS	θ_{MS} (see Figure 5.3)	deg	
MsVelocity	POS	MS velocity	m/s	calculated from *Stations.Velocity*
MsDirection	POS	$\theta\varpi$ (see Figure 5.3)	deg	

5.2.2.2 *Stations* (*Array*) structure

Properties of the "Stations" are defined by its position and antenna array characteristics. The antenna array description is created by making a copy of the *Array* structure[2]. The desired collection of *Array* structures, intended for use in simulation, should be created only once - in preprocessing phase, independently from channel simulations (section 4.1). The *Array* structure contains definitions of array geometry and radiation patterns. Remaining parameters for position, orientation and velocity, should be adjusted according to targeted network layout.

Table 5.5. *<u>Stations (Array)</u>* parameters.

Parameter Name	Group	Definition	Default value	Unit	Note

[2] *Array* structure is not provided as input argument to WIM function since necessary *Array* definitions are embedded into Station definition, being part of *layoutpar*.

Pos	POS	[x;y;z] position in GCS		m	
Rot	POS	[Rot_x;Rot_y;Rot_z] rotation of array by respective axes of GCS		rad	
Element	ANT	vector of array elements			
CommonAperture	ANT	EADF common to all elements (if empty each element has to provide its own pattern). This parameter is used only during Array construction when provided field patterns are defined in ECS. In this case rotation of ECS 3D field patterns is performed to calculate their EADF representation in ACS.			Temporary parameter used only in preprocessing phase
Aperture	ANT	if preprocessing is used, this field contains EADF for the whole array			structure
Velocity	POS	[v_x;v_y;v_z] velocity of this station			

Since velocity vector is assigned to this structure, in the new WIM implementation MS and one-sector-BS can be treated in the same way. The fundamental difference between MS/BS is that BS has zero velocity. In the new model this is achieved by setting *Station.Velocity* to zero.

5.2.2.3 Antenna element parameters

Each antenna element in characterized with its position in resprect to ACS:

Table 5.6. *Element* parameters.

Parameter Name	Group	Definition	Default value	Unit	Note
Pos	POS	[x;y;z] position in ACS		m	
Rot	POS	[Rot_x;Rot_y;Rot_z] rotation of element by respective axes of ACS			
Aperture	ANT	EADF for this element. This parameter is used only during Array construction when provided field patterns are defined in ECS. In this case rotation of ECS 3D field patterns is performed to calculate their EADF representation in ACS.			Temporary parameter used only in preprocessing phase

The geometry of the array is defined in meters (as well as station positions in GCS), not in normalized distance (number of wavelengths) as in the previous model versions.

Note that the mean power of narrowband channel matrix elements (i.e. summed over delay domain) depends on the antenna gains

5.2.3 External initialization of structural model parameters (INITVALUES) - optional

The fourth input argument, which is also a MATLAB struct, is optional. It can be used to specify the initial AoDs, AoAs, cisoid phases, path losses and shadowing values when WIM is called recursively, or for testing purposes. If this argument is given, the random parameter generation as defined in [1] 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. Note! The fourth input argument can be directly the output structure FULLOUTPUT defined in Table 5.8 or the structure defined in Table 5.7.

Table 5.7. MATLAB struct INITVALUES: initial values, fourth optional input argument.

Parameter name	Definition	Unit
InitDelays	A K x N matrix of path delays.	Sec
InitSubPathPowers	A K x N x M array of powers of the subpaths.	-
InitAods	A K x N x M array	Degrees
InitAoas	A K x N x M array	Degrees
InitSubPathPhases	A complex-valued K x N x M array. When polarization option	Degrees

	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.	
InitPathLosses	A K x 1 vector	Decibel
InitShadowLosses	A K x 1 vector	Decibel
InitIndOfSpreadClust	A K x 2 matrix. Index to two strongest clusters. These clusters are spread to three delay positions if parameter IntraClusterDsUsed = 'yes'	

5.2.4 Output parameters

There are three output arguments: H, DELAYS, FULLOUTPUT. The last two are optional.

Table 5.8. MATLAB output parameters.

Parameter name	Definition	Unit	Note
H	cell array of size K (number of links). Each element of this cell array contains a U x S x N x T matrix.		See Appendix 1 for terminology
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 path powers.	linear	
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	
path_losses	A K x 1 vector	linear scale	
MsBsDistance	1 x K vector of MS-BS distances	m	
shadow_fading	A K x 1 vector	linear scale	
sigmas	A K x 4 vector of per link large scale parameters (ASD ASA DS SF)	-	
propag_condition	A K x 1 vector indicating LOS / NLOS condition (0=NLOS, 1=LOS)	-	
Kcluster	A K x 1 vector defining narrowband K-factors of links.	linear scale	Only with LOS
Phi_LOS	Final phases for LOS paths, K x 1 array.	deg	Only with LOS
scatterer_freq	A K x N x M array of subpath Doppler frequecies	Hz	Only with B5 CDL
subpath_phases	A complex-valued 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	
delta_t	A K x 1 vector defining time sampling interval for all links.	sec	
IndexOfDividedClust	A K x 2 matrix. Index to two strongest clusters. These clusters are spread to three delay positions if parameter IntraClusterDsUsed = 'yes'	-	
xpr	A K x N x M array of cross-polarization coupling power ratios.	linear scale	Only with PolarisedArrays case.

5.3 Implementation notes

5.3.1 Speed optimization

A performance improvement may be achieved by setting wimpar. LookUpTable=-1. This activates the lookup table for computing the complex exponential, in the core equation of the channel model. Alternatively, one can set the number of points in the look-up table by e.g. setting wimpar. LookUpTable=1024. The default lookup table size (with LookUpTable = -1) is 2^14=16384.

5.3.1.1 Compatibility issues

In the previous implementation of WIM implementation ANSI-C support was provided for some critical functions: scm_mex_core.dll and interp_gain_c.dll. The new WIM version uses EADF-based antenna pattern interpolation, and modified version[3] of scm_mex_core.m, and therefore *.dll files from previous implementations can not be used for speed optimization.

5.3.2 Output interpolation

Channel sampling frequency has to be finally equal to the simulation system sampling frequency. To have feasible computing time it is not reasonable to generate channel realisations on the sampling frequency of the system to be simulated. The channel realisations can be generated on some lower sampling frequency and then interpolated to the desired frequency. A practical solution is e.g. to generate channel samples with sample density (i.e. over-sampling factor) two, interpolate them accurately to sample density 64 and to apply zero-order-hold interpolation to the system sampling frequency. Channel impulse responses can be generated during the simulation or stored on a file before the simulation on low sample density. Interpolation can be done during the system simulation.

6. References

[1] IST-WINNER II, D1.1.2 “WINNER II Channel Models”, ver 1.0, Sep 2007, https://www.ist-winner.org/WINNER2-Deliverables/.

[2] IST-WINNER II, D1.1.1 “WINNER II Interim Channel Models”, ver 1.2, Feb 2007, https://www.ist-winner.org/WINNER2-Deliverables/.

[3] D5.4, “Final Report on Link Level and System Level Channel Models”, ver 1.4 , November 2005.

[4] D5.3, “Interim Channel Models”, April 2005.

[5] “Spatial channel model for Multiple Input Multiple Output (MIMO) simulations” 3GPP TR 25.996 V6.1.0

[6] M. Narandžić, M. Käske, C. Schneider, M. Milojević, M. Landmann, G. Sommerkorn, and R.S. Thomä, „3D-Antenna Array Model for IST-WINNER Channel Simulations,” *Proc. of IEEE VTC2007-Spring*, Dublin, Ireland, April 23 - 25, 2007.

[3] scm_mex_core.dll created for previous version does not support intra-cluster delay spread option.

Appendix A: Terminology and notation

'Link' = one-directional (downlink or uplink) BS-MS connection. The term 'link' is sometimes, but not always, interchangeable with 'user', i.e. MS.

'Channel realisation' = a sequence of channel matrices over a pre-defined number of time samples (in WIM, each element of the channel matrix has six channel clusters in delay domain).

Cluster $\equiv$ tap $\equiv$ path
Ray $\equiv$ subpath
K number of links
N number of paths
M number of subpaths within a path
U number of receiver elements
S number of transmitter elements
T number of time samples