Dpto. Ingeniería de Comunicaciones. University of Málaga

Params. Description

Wireless Mobile Simulator (WiMo-SIM)

ltesim@ic.uma.es
24/02/2016

1. Introduction

This document gives a brief description of all the parameter associated to the simulation. The parameters are organized in tabs and boxes. Each tab contains several boxes that are somewhat related. In the list presented, the type of each parameter is placed between brackets on the left of it. Additionally, there could appear the units associated to the parameter between parentheses on the right of it.

2. Parameters Description

2.1 CONFIG

This tab contains parameters that control the simulation length, the saved results associated to the scheduling process in order to monitor the behavior of the scheduler and the deep of the trace level. Each parameter is associated to a different box as depicted below. The boxes are '*Simulation'*, *'Monitoring'*, '*Traces'* and *'Scenario'*.

2.1.1 Simulation

This box control general features of the simulation to be performed.

- *[string] simName*: This parameter is used to build the name of the folder which contains the obtained results in the form of a Matlab file with extension *.mat*. The actual name of the results folder also includes the name of the variable parameters defined in the tab *VAR_PARAM* (see *Section 2.9*) and the date associated to the performed simulation with format *yymmdd_hhmm*.
- *[int] iterationsNumber (subframes)*: This parameter sets the simulated time. It indicates the number of subframes of 1ms to be transmitted from the base station (BS).
- *[int] seed*: This parameter determines the generation of the random variables. If this parameter is set to '0', the actual seed used in order to generate the random variables is itself random. By contrast, if the value used is different to '0' all the pseudorandom generators use this seed.
- *[bool] saveHistograms*: : In case of being 'true', the associated delay and number of transmissions of each packet sent is stored. The resulting matrixes are useful so as to represent the packet delay and packet number of transmission histogram.

2.1.2 Monitoring

This box has parameters that control whether some variables are stored in the results folder in order to monitor the behavior of the scheduler in the system.

- *[bool] monitorMAC*: if ‘true’, a set of variables related to MAC layer are stored in the results folder in a file with a name starting with the string *'monitorMAC'*.
- *[int] subFrameIniMAC (subframes)*: It indicates the first subframe that is going to be monitored.

- *[int] subFrameStepMAC (subframes)*: Each '*subFrameStepMAC* ' subframes the set of variables to be monitored is saved.
- *[int] subFrameNumMAC (subframes*): It indicated the overall number of subframes to be monitored.

2.1.3 Traces

This box has parameters related to the degree of information that is being displayed or saved. The first two parameters fix the trace level of saved results in debug and release mode when running the executable from the Visual Studio toolbox so this parameter are superfluous when running simulation directly from the .exe file.

- *[int] minTraceLevel*: Minimum trace level (superfluous for the free version of WiMo-SIM)
- *[int] maxTraceLevel*: Maximum trace level (superfluous for the free version of WiMo-SIM)
- *[scalar] refreshPercent (% simulation)*: When a simulation is being performed, the console associated to the simulation displays the percentage of iterations (subframes) that have been carried out. The parameter '*refreshPercent'* indicates the degree of updating of this percentage.

2.1.4 Scenario

This box has parameters related to the definition of the static scenario considered in order to run the simulation.

- *[int] dummyTxNbr*: Number of dummy transmitter which will be considered in each simulated cell. The information these transmitters send is got from a *.mat* file, so that they act as interference with no additional computational cost in the simulation.
- *[string] dummyTxFile*: It determines the name of the file (with extension *.mat*) which contain the symbols (filtered by the channel) associated to dummy transmitters.
- *[bool] getDummyTxFile*: if 'true', the symbols transmitted during the simulation run (after filtered by the channel) are saved in the file indicated in the parameter *dummyTxFile*.
- *[scalar] loadRateOCI*: Percentage of interference of the non-simulated cells considered. The value of this parameter must be in the range [0 - 1].
- *[string] multiCellFile*: It determines the name of the file (without extension *.mat*) with the information which defines the static characteristics of the scenario considered (like number of simulated cells, number of simulated users per cell, etc.) This file is generated using the *networkPlacement* application. It should be mentioned that the named indicated in this parameter must not contain the information related to realizations and variations, i.e. if the name of the file to be

used is *multiCellParams_1c_4users_iR1_iV1.mat*, the value which must be used in the parameter is '*multiCellParams_1c_4users'*.

- *[int] realization*: Number of realization which are taken into account in the simulation. It should be the same as that indicated in the *mainParam* Matlab file (see [1] for more details).
- *[int] varMatlabValue*: Number of variations which are taken into account in the simulation. It should be the same as that indicated in the *mainParam* Matlab file (see [1] for more details).

2.2 CHANNEL

This tab contains parameters that control the configuration of the channel model employed in the simulation and the interference considered. Two boxes can be found: '*ChConfig'* and '*Interferences*'.

2.2.1 ChConfig

This box has parameters that determine the configuration of the channel.

The channel model assumed is Rayleigh/Ricean fading. Moreover, the Kronecker model is used if a MIMO scenario is simulated.

- *[int] numAntenasTx (antennas)*: Number of antennas in each transmitter.
- *[int] numAntenasRx (antennas)*: Number of antennas in each receptor.
- *[matrix] correlationTx* : It contains the correlation matrix in transmission written by-row. A white space should be used to separate each element of the matrix. The total number of elements must be (*numAntenasTx* x *numAntenasTx*). In a SISO scenario, the value must be 1.
- *[matrix] correlationRx* : It contains the correlation matrix in reception written by-row. A white space should be used to separate each element of the matrix. The total number of elements must be (*numAntenasRx* x *numAntenasRx*). In a SISO scenario, the value must be 1.
- *[bool] efficientMode:* If 'true', an efficient way is adopted to generate the channel in order to reduce the simulation time. This option is only valid when the frequency model is selected (i.e. *frequencyModel = true*).
- *[bool] frequencyModel:* If 'true', the transmited signal is filtered in the frequency domain and, therefore, the received signal is in the frequency domain too.
- *[string] metodoCorrTemporal*: It determines the model used to generate the temporal correlation between channel samples. Two options can be selected:
    - *Jakes* → a Filter-based approach is used to generate the channel samples [2]
    - *SOS* → a Sum Of Sinusoids-based approach is used to generate the channel samples [3].

- *[int] numSinusoides (sinusoids):* If the model used to generate the temporal correlation between channel samples is the SOS model, this parameters determines the number of sinusoids used in this method. A value less than 10 is not recommended.
- *[string] predefChannel*: Three options can be selected:
    - *NONE* → The channel simulated is an AWGN channel.
    - *SELECTIVE_RAYLEIGH* → The channel simulated is Rayleigh fading and it has the power delay profile of a Typical Urban Channel (TU). The concrete values are detailed in Table 1.

Attenuation	***delay [samples]***
1	*0*
2.15	*1*
3.14	*3*
4.29	*5*
5.4	*10*

Table 1. Power Delay Profile associated to SELECTIVE_RAYLEIGH option

    - *FLAT_RAYLEIGH* → *The* channel simulated is flat Rayleigh fading (only one tap is generated)

  It should be noted that this parameter is only considered if the parameter *customChannel* is 'false'.
- *[bool] customChannel:* If 'true', the power delay profile associated to the channel is determined by the values associated to the parameters *attenuation* and *delay*. If 'false', the channel is generated according to the option selected in the parameter *predefChannel*.
- *[vect] attenuation:* It contains as elements as the number of path associated to the channel. The *i*th value represents the amplitude associated to the *i*th path.
- *[vect] delay (samples):* It contains as elements as the number of path associated to the channel. The *i*th value represents the delay (in number of samples) associated to the *i*th path.
- *[scalar] kRicean (dB):* Power of the direct path expressed in dB. If a value less than -80 dBs is used, the channel model is Rayleigh fading since no direct path is generated. Otherwise, the channel model is Ricean.
- *[scalar] aoaRicean (degrees)*: If a Ricean model is selected, it determines the angle of arrival of the direct path.

- *[scalar] mobileSpeed (km/h):* Velocity of the Users Equipment (UEs) expressed in kilometer per hour.

2.2.2 Interferences

This box has parameters that determine if a source of interferences is considered in the simulation. The interference considered affects to each subcarrier and the concrete value is load from a file.

- *[bool] FMinterference*: If 'true', a source of interference is considered in each subcarrier.
- *[string] InterferenceFile*: Name of the file used to load the interference.
- *[int] idAntenaInterferente*: It identified the antenna which will suffer the effect of the interference. Possible values: [1 – *numAntenasTx*].

2.3 CONTROL

This tab controls functionalities associated to control channels of the return link which conveys the CSI reported from the receptors.

2.3.1 Signalling

- *[bool] directMuxSignal*: It activates direct signaling from receptors to transmitters by the return link.

2.3.2 Feedback

This box contains parameters that model the return channel between each receptor and each transmitter.

- *[int] chReportDelay (subframes)*: This parameter models a delay in the return link associated to the channel information (i.e. CQI and PMI) of '*chReportDelay*' subframes.
- *[int] harqReportDelay (subframes)*: This parameter models a delay in the return link associated to ACK/NACK information of the HARQ channels of '*harqReportDelay*' subframes.

2.3.3 Reports

- *[int] cqiReportingRate (subframes)*: It indicates the reporting rate of the CQI index in subframes.
- *[int] pmiReportingRate (subframes)*: It specifies the reporting rate of the PMI index in subframes.
- *[int] cqiChunk (PRBs)*: Both PMI and CQI are reported for a band of frequencies. Such band is determined by the parameter *cqiChunk* which is given in number of PRB.

2.4 PHY

This tab contains parameters associated to the LTE physical layer.

2.4.1 OFDM

This box contains parameters to configure the bandwidth transmission.

[int] nCarriers (subcarriers): It defines the number of physical subcarriers for the whole bandwidth, i.e. it defines the transmission bandwidth including data subcarriers and the guards bands. The accepted LTE-A values are:

nCarriers	*Bandwidth[MHz]*
128	*1.4*
256	*3*
512	*5*
1024	*10*
1536	*15*
2048	*20*

Table 2. Number of physical subcarriers for LTE bandwidths

However, higher bandwidth sizes can be used if CA with adjacent CCs is used.

- *[int] dataCarriers (subcarriers):* It specifies the number of allocable subcarriers. Its value depends on the value of *nCarriers* as can be seen in Table 3:

nCarriers	*DataCarriers*
128	*85*
256	*180*
512	*300*
1024	*600*
1536	*900*
2048	*1200*

Table 3. Value of DataCarriers according to the value of nCarriers.

- *[int] PRBCarriers (subcarriers):* It sets the number of subcarriers per PRB. The typical value is 12.

- *[int] cPrefixLength (samples):* It specifies the length of the cyclic prefix. This parameter will be taken into account for the temporal channel model.
- *[int] PRBsymbols (OFMD symbols):* Number of OFDM symbols that form a PRB.
- *[int] pilotDataRatio:* It indicates the lineal power relationship between pilots and data.
- *[scalar] samplingFrec (Hz)*: It indicates de sampling frequency. Its value depends on the transmission bandwidth (see Table 4).

nCarriers	***samplingFrec [Hz]***
128	*1.92e6*
256	*3.84e6*
512	*7.68e6*
1024	*15.36e6*
1536	*23.04e6*
2048	*30.72e6*

Table 4. Sampling frequency for different values of nCarriers parameter

- *[scalar] carrierFrec (Hz)*: It indicates the frequency of the central carrier.

2.4.2 MIMO

This box contains parameters to configure the MIMO transmission mode.

- *[string] txMode:* It indicates the MIMO transmission mode. The possible values are:
  - *SISO* → Single Input Single Output. One antenna at transmitter and receiver.
  - *BF* → Beamforming, i.e. one layer spatial multiplexing MIMO mode.
  - *MRC* → Maximal Ratio Combining. Uses more than one antenna at reception.
  - *MUX* → Two layers spatial multiplexing MIMO mode.
  - *SFBC* → Space Frequency Block Code. Uses more than one antenna at transmission and reception.

- *Precoded MUX* → Precoded two layers spatial multiplexing MIMO mode.
- *Rank Adaptation* → When this value is used, the transmitter adapts the number of layers in order to achieve the maximum throughput.

- *[int] nLayers (layers):* This parameter indicates the number of layers of the transmission mode. When the transmission mode *Rank Adaptation* is activated, the number of layers is adaptively chosen.
- *[string] precoding:* It indicates whether the precoding is ideal or codebooks are used. This parameter only is taken into account when the transmission mode is *BF, Precoded MUX* or *Rank Adaptation*.
- *[string] codebookFile:* If codebooks are selected in the parameter *precoding*, the value of these codebooks are taken from the file specified in this parameter. If this field is empty, codebooks values are taken from the parameter *predefCodebooks*.
- *[string] predefCodebooks:* It indicates the value of the codebooks to be used for precoded MIMO mode. There are two types of codewords:
  - *LTE – 4words/6words*: LTE standard codebooks of 4 or 6 values.
  - *Grassmanian – 6words/8words*: Grassmanian alphabet of 6 or 8 values.

2.4.3 UE_Rx

This box contains parameters that configure the physical layer of the User Equipment.

- *[int] bandwidth (MHz)*: It indicates the reception bandwidth.
- *[string] chEstimationMethod:* This parameter indicates the frequency channel estimation method. Except for the ideal estimation, the frequency channel response is estimated only in the OFDM symbol with reference signals:
  - *IDEAL* → Frequency channel response is taken directly from the mobile channel for a whole subframe.
  - *ZHAO* → The Zhao method is carried out in the whole bandwidth.
  - *ZHAO_LINEAL* → The Zhao method is carried out in central frequencies while a lineal interpolation is carried out at the edges.
  - *LOW-PASS* → A low pass filtering of the received reference signals is carried out.
  - *LINEAL* → A lineal interpolation of the reference signals value is carried out.

- *LINEAL_FILTER* → An exponential filter is first applied to the each received reference signal and the previous reference signals at the same frequency. Next, a lineal interpolation of these filtered values is carried out.

- *[string] timeInterMeth:* It indicates the channel estimation method for the OFDM symbols with no reference signals when no ideal frequency estimation method is carried out.
  - *LINEAL* → A lineal interpolation of the estimated value of a subframe is carried out.
  - *RETENTION* → Values estimated at the OFDM symbols of a subframe are average out.

- *[int] nCalcs:* Number of SNR values calculated for PRB to be used in the turbo decoded. The highest *nCalcs* is, the more accuracy in the turbo decoding process when fading channels are used.

- *[string] mimoDetectionMethod:* It indicates the MIMO detection mode used. Several methods have been implemented based on Zero Forcing (ZF) or on Minimum Mean Square Error (MMSE).

- *[string] snrEstimationMethod:* It indicates the method used to estimate the SNR:
  - *IDEAL* → The SNR value is taken from the set value in the *Params* file.
  - *Error Base* → The SNR value is calculated from the difference between the transmitted reference signal value and the received reference signal value.

- *[int] maxStepsSOVA:* Maximum number of steps in the turbo decoding process. Its value has a different meaning according to the channel decoding algorithm chosen:
  - If SOVA algorithm, it indicates the maximum number of steps of the updating process.
  - If BCJR algorithm, it indicates the size of the decoding and updating window.

- *[string] sisoDecodingAlgorithm:* It indicates the algorithm for the channel decoding:
  - *SOVA* → Soft Output Viterbi Algorithm
  - *BCJR* → Bhal, Cocke, Jelinek and Raviv

2.5 MAC

In this tab the parameters related with the Medium Access Control are configured.

2.5.1 AM

This box configures the parameters for the adaptive modulation and coding:

- *[bool] adaptiveModulation:* If true, adaptive modulation to an instantaneous target BLER is carried out. If channel coding is active, adaptive coding is also carried out.
- *[string] thresholdsFileName:* This parameter specifies the *.mat* file where the value of the thresholds used for the adaptive modulation and coding.
- *[bool] onlyEvenNumBits:* It indicates if modulations can use odd number of bits (false) or even (true).
- *[bool] txDuringOutage:* If true, transmitting data is always allowed although the adaptive modulation and coding process determines than the instantaneous BLER is higher than the target BLER. In that case, the transmission is carried out using the lowest modulation and coding scheme.
- *[int] maxbits (bits):* If adaptive modulation is activated (*adaptiveModulation* = true), it indicates the maximum number of bits that can be used for modulation. If not (*adaptiveModulation* = false), it indicates the number of bits used for modulation.
- *[scalar] maxBler:* This parameter sets the value of the instantaneous target BLER for de adaptive modulation.
- *[string] linkAdaptMethod:* It determines the method used by the UE to carried out the link adaptation when the adaptive modulation is activated:
    - *LinkAdaptForAWGN* → CQI values are obtained by mapping the estimated SNR value directly to a CQI value from look up tables whose values have been obtained by simulations.
    - *OLLA* → Outer Loop Link Adaptation algorithm is applied to the estimated SNR value to ensure the target BLER value.
- *[scalar] stepOLLA:* if the OLLA algorithm is chosen to carried out the link adaptation at the UE, this parameter indicates the size of the step of this algorithm.

2.5.2 HARQ

In this box activation of the Hybrid Automatic Repeat Request and its configuration is carried out.

- *[bool] codingActive:* If true, the channel coding is activated using the algorithm established in *sisoDecodingAlgorithm.*
- *[bool] harqActive:* If true, the Hybrid Automatic Repeat Request is activated.
- *[int] harqCqiType:* This parameter determines if the transmitter modifies the CQI value reported by the UE when a retransmission is carried out.
    - *harqCqiType = 0* → Keeps the CQI value used in the first transmission of these TB.
    - *harqCqiType = 1* → Reduces the CQI value for a value of the parameter *CQI_Reduction.*
    - *harqCqiType = 2* → Uses the CQI value reported by the UE.
- *[int] CQI_Reduction:* This parameter indicates the value for which the value of the received CQI from the UE has to be reduces if *harqCqiType = 1.*
- *[bool] incrementalRedundancy:* If true, the incremental redundancy technique is performed during retransmissions.
- *[bool] chaseCombining*: If true, the chase combining technique is performed during retransmissions.
- *[int] harqChNo (channels):* This parameter indicates the number of HARQ channels per user (typically 8 in LTE/ LTE-A).
- *[int] maxHarqReTxNo (reTx):* This parameter indicates the maximum number of HARQ retransmissions allowed (the typical value of LTE/LTE-A is 3).
- *[string] minAssignSizeForReTx:* This parameter sets the initial value of number of PRBs from which the scheduler stars searching an allocable number of PRBs for a retransmission.
    - *sameAsNewTx* → The initial value is the number of PRBs used for the previous new transmission.
    - *minAssignSize* → The initial value is the value of the parameter *minAssignSize.*

2.6 Scheduler

This tab makes possible to determine the scheduling algorithm used in the simulation as well as the parameters necessary to configure it.

- *[string] muxAlgorithm* : It determines the scheduling algorithm used in the simulation. The following options are possible:

- *RR* → Round Robin
- *PF* → Proportional Fair
- *hardPriority* → Opportunistic Hard Priority
- *softPriority* → Channel Dependent Earliest Deadline Due (CD-EDD)
- *weightPriority* → CD-EDD with postponed EDD
- *PowerControl* → Power Control

- *[int] assignSize (PRB)*: This parameter, together with *assignSizeComputation* and *adjustTbSize,* determines the number of resources allocate to each user.

  If *assignSizeComputation* = fixed → the number of resources allocate to each user will be $\lceil \text{assignSize}/\text{minAssignSize} \rceil \cdot \text{minAssignSize}$ [1] or less, depending on the value of the parameter *adjustTbSize* (as it will be seen).

  Otherwise, the number of resources allocate to each user depends on the value of the parameters *assignSizeComputation* and *adjustTbSize* (as it will be seen).

- *[string] assignSizeComputation:* This parameter determines the algorithm which will be used in order to determine the number of resources assigned to each user. Three algorithms have been implemented:
  - *fixed:* The number of resources assigned to each user is determined by the parameter *assignSize*, using the equation $\lceil \text{assignSize}/\text{minAssignSize} \rceil \cdot \text{minAssignSize}$.
  - *loadAware:* The number of resources assigned to each user is computed each TTI based on the number of active users in this TTI, i.e. the number of users who can do a new transmission or have a pending retransmission in this TTI. The maximum number of resources is computed using the following equation

    $\lceil (\text{Total num PRBs}/\text{num active users})/\text{minAssignSize} \rceil \cdot \text{minAssignSize}$.
  - *maximum*: The number of resources assigned to each user in each TTI will be the total number of PRBs available (depending on the bandwidth configuration selected).

[1] The value of minAssignSize depends on the simulation bandwidth selected in number of PRBs

BW (num PRBs)	minAssigSize (num PRBs)
≤10	1
10<BW<27	2
26<BW<64	3
63<BW<110	4

It should be noted that the minimum number of resources allocated will be determined by *minAssignSize* in any case.

- *[bool] adjustTbSize:* If 'true', if the user has not enough information on his queue to be sent in a group of *N* PRBs, the number of PRBs is reduced according to the size of his queue.
  Otherwise,
    if *codingActive* is 'true', the information is sent in the group of *N* PRBs and the *free* PRBs will be filled in with extra coding.
    Otherwise, the group of *N* PRBs will not be assigned to the user.
  The number *N* is computed based on the algorithm selected in the parameter *assignSizeComputation*.
- *[int] typeOfCQI:*This parameter determines the way to compute the CQI associated to a group of PRBs. The possible values are:
  - *0* → The CQI will be computed as the mean value.
  - *1* → The CQI will be the minimum value of CQI in the group of PRBs
  - *2* → The CQI will be the CQI associated to the PRB located in the center of the group, e.g. in a group of 5 PRBs, the CQI of the group will coincide with the CQI associated to the PRB number 3.
- *[string] meanRateComputation*: This parameter is only used when the scheduling algorithm selected is not 'RR' or 'PowerControl'. It determines the way to compute the mean rate associated to each user. Two values are possible:
  - *meanRateCte* → The mean rate will be the rate corresponding to the mean fading level associated to each user according to the constraints established in LTE, i.e. it will depend on the mean SINR.
  - *meanRateSlidingWindow* → The mean rate of each user will be compute as the rate actually given to each user and measured over a certain *sliding window.*
- *[int] window (TTIs):* This parameter is only valid when the value of the *meanRateComputation* selected is *meanRateSlidingWindow.* It determines the size of the *sliding window* expressed in number of TTIs. A different value can be assigned to each user (see Section 2.8).
- *[int] weightedDelay:* Not used in the current version.
- *[int] typeDelayMeasure:* Not used in the current version.
- *[int] delayThreshold (TTIs):* This parameter is associated to the scheduling algorithm: *hardPriority*, *softPriority*, *weightPriority* or *PowerControl*. If other scheduling algorithm is used, the value associated to it must be 0.
  It determines the maximum admissible associated delay of the head of line (HOL) packet in the queue of a user. Packets whose delay exceeds the *delayThreshold* are discarded. A different value can be assigned to each user (see Section 2.8).
- *[int] delayGuard (TTIs):* This parameter is only used when the following scheduling algorithm are selected: Opportunistic Hard Priority or CD-EDD with postponed EDD.
  It is a delay tolerance, i.e. how close to the *delayThreshold* established can be the delay of the HOL packet before the algorithm acts.

- *[bool] singleUser:* This parameter is only used when a two codewords MIMO mode is selected.

  If 'true', multiple antennas Tx send data to multiple Rx antennas located at the same UE (Single User MIMO mode). E.g. if PRBs 1-4 of codeword 1 are assigned to user 1, PRBs 1-4 of codeword 2 must be assigned to user 1, too.

  Otherwise, separate data streams are sent to spatially separated UEs over the same sub-channel (Multi User MIMO mode). E.g. if PRBs 1-4 of codeword 1 are assigned to user 1, PRBs 1-4 of codeword 2 must be assigned to other user.
- *[string] tbSizeFileName:* Name of the *.mat* file where the LTE recommended transport block size as a function of the number of PRBs assigned and the CQI associated to them are stored.
- *[bool] fixedCodRate:* if 'true', the coding rate will be determined by the value in the parameter *codRate*.
- *[scalar] codRate:* Value of the coding rate that will be applied if the parameter *fixCodRate* is 'true'.
- *[vect] semiPersUser:* Only two values are valid:
  - '1' → semi-persistent scheduling is used.
  - '0' → semi-persistent scheduling is not used.

  It is a vector so that its number of position must be equal to the number of users.
- *[vect] spsTTI:* It is a vector so that its number of position must be equal to the number of users. In position *i*, active period associated to the *i*th user, i.e. TTIs in which the BS has to calculate the Modulation Coding Scheme (MCS) and numbers of PRBs to transmit information based on the average channel quality of the *i*th user will be a multiple of this value.
- *[vect] spsPeriod:* It is a vector so that its number of position must coincide with the number of users. In position *i*, new transmission period associated to the *i*th user, i.e. TTIs in which user *i* can do a new transmission based on SPS will be a multiple of this value. It should be noted that *spsTTI>spsPeriod* always. Otherwise, an error message will be generated.
- *[vect] trafficClass:* Not used in the current version.

2.7 RLC

This tab contains parameters associated with the Radio Link Control layer

2.7.1 Queues

This box contains parameters referred to the queues of the RLC layer

- *[int] sizeOfQueues (bits):* This parameter indicates the maximum number of bits that can be queued before serving. In case of exceeding this value, the incoming bits are discarded.

2.7.2 Source

This box contains the parameters associated to the source of RLC packets.

- *[int] sourceModel:* This parameter set the type of RLC source model:
  - *1* → Full Buffer: It simulates the transmission of an infinite RLC packet. Thus, queues are always full. For this source model packet measurement such as packet delay can not be carried out.
  - *2* → Streaming: It generates RLC packets based on the Truncated Pareto distribution for both the packet size and the inter arrival packet time. The configuration of this source model is carried out using parameters of this box. The Probability Density Function (PDF) of the Truncated Pareto distribution is:

$$f_x(x) = \begin{cases} \alpha \dfrac{x_m^{\alpha}}{x^{\alpha+1}} \text{ when } x > x_m \\ 0 \text{ when } x < x_m \end{cases}$$

  - *3 and 4* → It generates traffic based on traces. When these values are set, the packet size and the inter arrival time between packets are read from a *.mat* file. Thus, it can be simulated any type of traffic model.
- *[string] gamingSourceModel:* This parameter contains the name of the Matlab file from which traces are read, when parameter *sourceModel* is set to 3.
- *[string] ipCameraSourceModel:* This parameter contains the name of the Matlab file from which traces are read, when parameter *sourceModel* is set to 4.
- *[scalar] alphaParetoPacket:* This parameter sets the *α* parameter of the Truncated Pareto distribution for the packet size.
- *[scalar] meanParetoPacket (bytes):* This parameter sets the mean value of the packet size for the Truncated Pareto distribution.
- *[scalar] maxParetoPacket (bytes):* This parameter sets the maximum value allowed for the packet size when the Truncated Pareto distribution is used.
- *[scalar] alphaParetoTime:* This parameter sets the *α* parameter of the Truncated Pareto distribution for the inter arrival time between packets.
- *[scalar] meanParetoTime (ms):* This parameter sets the mean value of the inter arrival time between packets for the Truncated Pareto distribution.
- *[scalar] maxParetoTime (ms):* This parameter sets the maximum value allowed for the inter arrival time between packets when the Truncated Pareto distribution is used.

- *[scalar] incrementalFactorPacket:* The value of this parameter is added to the value of the *meanParetoPacket* parameter It is useful to set this parameter as a variable parameter (see Section 2.9).

- *[scalar] incrementalFactorTime:* The value of this parameter is added to the value of the *meanParetoTime* parameter. It is useful to set this parameter as a variable parameter (see Section 2.9).

2.7.3 Description

In this box it is shown the possible values for the *sourceModel* parameter and their meanings. It should be mentioned that this parameter does not establish the source model which will be used in the simulation. It is only used as help in order to remember the meaning of the number which identifies each source model.

- *[string] sourceModel_des:*
    - *1→ Full Buffer*
    - *2→ Streaming*
    - *3→Gaming*
    - *4→ Camera*

2.8 MULTI_USER

This tab allows specifying some parameters for each specific UE. The way to indicate the value of each parameter for each UE can be done in a deterministic manner, or in a random manner. The random manner involves a random dispersion of the selected parameters with one of the following distribution: log-normal, Gaussian or uniform. If a random manner is selected the mean and standard deviation (log-normal and Gaussian), or the minimum and maximum value (uniform) must be added.

Specifically, let consider a multiuser parameter (i.e. any parameter contained in the multiuser tab except *sourceModel*). It should be noticed that the types of all the multiuser parameter is '*vect*'. The first element of the vector determines whether the distribution of the parameter will be deterministic (first element is '0') or random (first element is not '0').

In case of deterministic manner, the selected multiuser parameter must be a vector of length *nUsers+1*. The first element must be '0', while the rest of elements specify the value of that parameter associated to each user in increasing order. For instance, consider that we want to place 5 UEs moving at different velocities. If we set '*mobileSpeed = 0 5 30 60 5 150*', UE 1 will move at 5km/h whereas UE 5 for instance will move at 150 km/h.

In case of random distributions, the desired values for each UE cannot be specified. Instead of that, a common value of the parameter for all UEs and then the statistics that characterize the chosen distribution must be specified. WiMo-SIM internally will generate as many random values as UEs are considered with that probability distribution. Afterwards, the common parameter value is added to each random value to get the actual parameter value used for each UE.

Notice that if the '*seed*' parameter of *CONFIG* tab is set to '0' (random seed), each time you run a simulation a random multiuser parameter will take different values for each UE; whereas if the seed is the same between different simulations, a random multiuser parameter will take exactly the same values.

The available random distributions are log normal (first element is set to '1'), Gaussian (first element is set to '2') and uniform (first element is set to '3'). In uniform distributions, the specified statistics are minimum and maximum value instead of mean and variance. The next table summarizes the structure of the vector associated to a random multiuser parameter.

	First element	***Second element***	***Third element***	***Fourth element***
LOG-NORMAL	1	Common param. value	Mean	Standard dev.
GAUSSIAN	2	Common param. value	Mean	Standard dev.
UNIFORM	3	Common param. value	Minimum value	Maximum value

Table 5. Summary of the vector structure associated to random multiuser parameters

The multiuser parameters that appear in the MULTI_USER tab are listened below. The parameter *randomDistribution* at the bottom of the tab does not select the user distribution, (this is determined by the first element of each multiuser parameters) and it appears only for the sake of recalling the possible distribution types and the associated numbers (0-3). Finally, notice that if you do not write any multiuser parameter, all the UEs will have the same value in all parameters (i.e. same SNR, mobile speed, etc.).

2.8.1 CHANNEL

- *[vect] SNR (dB)*
- *[vect] mobileSpeed (km/h)*

2.8.2 PHY

- *[vect] bandwidth (MHz)*

2.8.3 MAC

- *[vect] maxBer*
- *[vect] priority*
- *[vect] window (TTIs)*
- *[vect] weightedDelay*

- *[vect] delayThreshold*
- *[vect] delayBudget*
- *[vect] mTTI*
- *[vect] semiPersUser*
- *[vect] spsTTI*
- *[vect] spsPeriod*
- *[vect] trafficClass*

2.8.4 SOURCE

- *[vect] sourceModel*
- *[vect] sourceRate*
- *[vect] alphaParetoPacket*
- *[vect] meanParetoPacket (bytes)*
- *[vect] maxParetoPacket (bytes)*
- *[vect] alphaParetoTime*
- *[vect] meanParetoTime (ms)*
- *[vect] maxParetoTime (ms)*
- *[vect] incrementalFactorPacket*
- *[vect] incrementalFactorTime*

2.8.5 Description

- *[string] randomDistribution*:
  - 0. OFF (DETERMINISTIC)
  - 1. LOG_NORMAL
  - 2. GAUSSIAN
  - 3. UNIFORM

2.9 VAR_PARAM

This tab allows running simulations for different values of the selected parameters. Up to 3 different variable parameters can be configured. In the result folder, it will appear as many results files as combinations of the different variable parameters exist.

2.9.1 Parameter_1

- *[string] nameVar1*: This parameter gives the name of the parameter that will vary. This means that the simulation will be repeated for each given value this parameter takes as specified in this box.
- *[scalar] iniVar1*: It determines the initial value of the variable parameter 1.
- *[scalar] stepVar1*: It gives the step between consecutives values of this variable parameter.
- *[int] numVar1*: It specifies the number of values the variable parameter 1 will take.

2.9.2 Parameter_2

- *[string] nameVar2*: Analogous to ' *nameVar1*' but for parameter 2
- *[scalar] iniVar2*: Analogous to ' *iniVar1'* but for parameter 2
- *[scalar] stepVar2*: Analogous to ' *stepVar1*' but for parameter 2
- *[int] numVar2*: Analogous to ' *numVar1*' but for parameter 2

2.9.3 Parameter_3

- *[string] nameVar3*: Analogous to ' *nameVar1*' but for parameter 3
- *[scalar] iniVar3*: Analogous to ' *iniVar1'* but for parameter 3
- *[scalar] stepVar3*: Analogous to ' *stepVar1*' but for parameter 3
- *[int] numVar3*: Analogous to ' *numVar1*' but for parameter 3

For instance consider the following setting for the *VAR_PARAM* tab as Figure 1 illustrates.

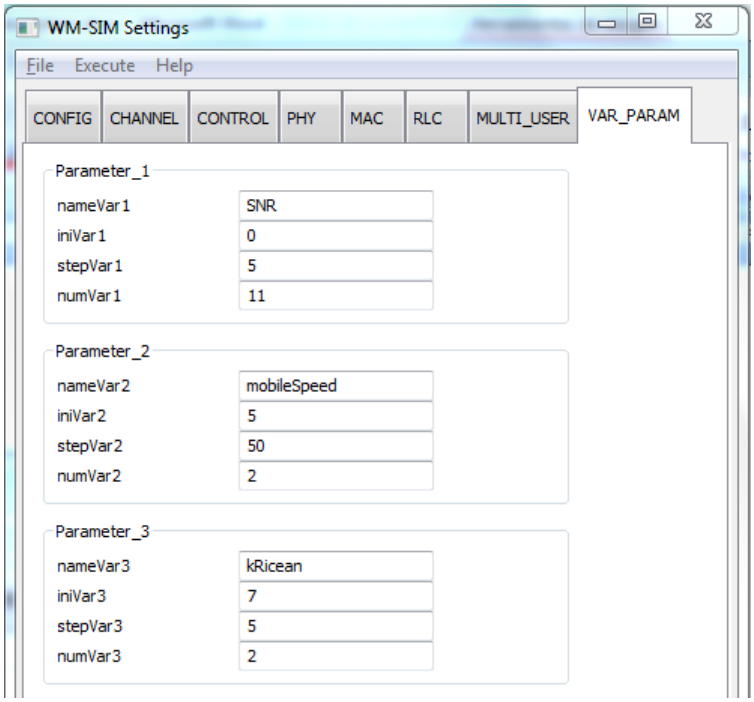

Figure 1. Example of variable parameters

This means that the simulation will be repeated 11x2x2 = 44 times and thus 44 results files will appear in the results folder. The folder name will be a concatenation of the *simName* parameter, the names of the variable parameters and the date. If *simName* is set to *sim*, the results folder will be named *sim_kRicean_mobileSpeed_SNR_yymmdd_hhmm* . The result Matlab file names will range from:

simulation_7_kRicean_5_mobileSpeed_0_SNR.mat to
simulation_12_kRicean_55_mobileSpeed_50_SNR.mat

3. Types

This section briefly shows the different types associated to the parameters.

- [string] : Vector of chars in order to store names of files, folders, parameters, etc.
- [bool] : Boolean value that can be true or false.
- [int] : Integer value.
- [scalar] : Real value.
- [vect] : Vector of real values. Each value must be separated by a blank space.
- [matrix]: A two dimension vector of real values. A matrix is written row wise as a vector in the parameters XML file.

4. Acronyms

AWGN	Additive White Gaussian Noise
BCJR	Bahl-Cocke-Jelinek-Raviv Algorithm
BF	Beamforming
BLER	Block Error Rate
BS	Base Station
CD-EDD	Channel Dependent Earliest Deadline Due
CQI	Channel Quality Indicator
HARQ	Hybrid Automatic Repeat Request
HOL	Head of Line
LTE-A	Long Term Evolution Advanced
MAC	Medium Access Control
MCS	Modulation and Coding Scheme
MIMO	Multiple Input Multiple Output
MMSE	Minimum Mean Square Error
MRC	Maximal Ratio Combining
OFDM	Orthogonal Frequency Division Multiplexing
OLLA	Outer Loop Link Adaptation

PDF	Probability Density Function
PF	Proportional Fair
PMI	Precoding Matrix Index
PRB	Physical Resource Block
RLC	Radio Link Control
RR	Round Robin
Rx	Receptor
SFBC	Space Frequency Block Code.
SISO	Single Input Single Output
SNR	Signal to Noise Ratio
SOS	Sum Of Sinusoids
SOVA	Soft-Output Viterbi Algorithm
SPS	Semi-Persintent Scheduling
TB	Transport Block
TTI	Transmission Time Interval
TU	Typical Urban
Tx	Transmitter
UE	User Equipment
XML	eXtensible Markup Language
ZF	Zero Forcing

5. References

[1] Network Placement_Description_v2.1.pdf

[2] F. Ruiz, J. Baños, "Implementación de un simulador software de desvanecimiento multicamino para comunicaciones móviles", Actas Congreso Español URSI, September 1994, pags. 788-792

[3] Y. R. Zheng and C. S. Xiao, “Improved models for the generation of multiple uncorrelated Rayleigh fading waveforms”, IEEE Commun. Lett., vol. 6, no. 6, pp. 256-258, June 2002.