



Publication Year	2024
Acceptance in OA	2024-03-18T11:27:10Z
Title	LFAA functional model user's guide
Authors	CHIARUCCI, Simone
Handle	http://hdl.handle.net/20.500.12386/34972

LFAA functional model user's guide

S. Chiarucci

February 29, 2024

Prepared by	Name	S.Chiarucci	Signature	
	Organization	INAF Osserv. di Arcetri	Date	13-Nov-2017

DOCUMENT SOFTWARE

Application	Version	Filename
Wordprocessing	LaTex	
Math. Simulation	Matlab	

ORGANIZATION DETAILS

Designation	Name	Organization
INAF		Istituto Nazionale di Astrofisica - Osserv. di Arcetri

Contents

1	Model code and user description	5
1.1	GUI applications	5
1.1.1	TPM Parameter Configurator	5
1.1.2	TPM Starter	8
1.2	End-to-End Matlab Model	9
1.3	Parameters Specification	11
1.4	Signal Generator	12
1.5	Quantization block	13
1.6	Signal processing chain	14
1.6.1	Frequency domain beamformer	16
1.7	Data analysis	17
1.8	Results	18

List of Tables

List of Figures

1	TPM configurator main panel	6
2	TPM configurator "Sky model" tab. The sky image in (1) is the map of the selected scenario at 50MHz.	6
3	TPM configurator "Station Beam & Spectra" tab	7
4	TPM secondary panels for the configuration of the polyphase filter	7
5	TPM simulation launcher panel	8
6	MATLAB "tpm_model_main" model	10
7	MATLAB "tpm_model_beamforming" model	10
8	MATLAB "tpm_model_CSP_input" model	11
9	Parameter specifications for tpm simulations. <i>Analysis_full.mat</i> and <i>Analysis_128.mat</i> are not used in the <i>tpm_model_CSP_input</i> model.	12
10	Signal generator model	13
11	The signal quantization block	14
12	Channelization and equalization modules	15
13	Delay-phase correction and calibration modules	15
14	Beamforming process block	16
15	Data analysis module. Applies only for ' <i>tpm_model_beamforming.m</i> ' and ' <i>tpm_model_main.m</i> ' simulation.	17
16	Two different file names are possible. Final results are stored and ready to be analyzed	18

List of acronyms

ADC: Analog to Digital Converter

CSP: Central Signal Processor

INAF: National Institute for Astrophysics

LFAA: Low Frequency Aperture Array Element or Consortium

LNA: Low Noise Amplifier

MATLAB: MATLAB simulation language and application

RFI: Radio Frequency Interference

SKA: Square Kilometre Array

References

- [1] G. Comoretto: "LFAA Tile Beamformer structure", INAF - Osservatorio Astrofisico di Arcetri Internal Report no. 2/2015
- [2] G. Comoretto: "LFAA Station Beamformer structure", INAF - Osservatorio Astrofisico di Arcetri Internal Report no. 5/2016
- [3] G. Comoretto: "Quantization noise, linearity and spectral whitening in the LFAA quantizer", INAF - Osservatorio Astrofisico di Arcetri Internal Report no. 3/2016
- [4] G. Comoretto, S. Chiarucci: "Quantization noise and nonlinearities in the correlation of two Gaussian signals", SKA-CSP-Memo 0016 (2015)
- [5] A. de Oliveira-Costa et al.: "A model of diffuse Galactic Radio Emission from 10 MHz to 100 GHz", DOI: 10.1111/j.1365-2966.2008.13376.x
- [6] A.Ludwig: "The definition of cross polarization", A. (1973). IEEE Transactions on Antennas and Propagation, 21(1), 116-119.

1 Model code and user description

The model describes correctly all the aspects of the LFAA station beamformer. It is an useful tool to analyze variants of the proposed signal processing algorithms or to highlight possible problems. It can also be used to test different calibration strategies and to provide the CSP model with a realistic input signal.

1.1 GUI applications

In this section are briefly explained two MATLAB applications developed to help the user to run the TPM simulations. Their aim is mainly to help the user to set the simulation parameters correctly, avoiding inconsistency during the parameters generation phase which might lead the simulator to raise an exception and stopped the running simulation. For this reason, it is recommended that users do not use the MATLAB script `'tpm_starter.m'` and `'generate_parameters.m'`, unless they are fully aware of the functionality of the simulation model.

1.1.1 TPM Parameter Configurator

The `'TPM_Parameters_configurator'` application generates the configuration file used by the simulator. The file name changes according to the TPM simulation chosen by the user.

In figure 1, the `"Station Configurator"` tab is shown. Here, the user can select the TPM simulation to run(1), the number of antennas to use(2) and one of the 512 stations available. The user can choose among a predefined set of values for these quantities, in order to not compromise the success of the simulation run. In the plots are shown the layout of the selected station(4) and its position compared to the array center(5). Three TPM simulation are available and their signal processing chain model is explained more in depth in the next section.

In the second tab, depicted in figure 2, the user can set the input sky model. Using a simplified version of the Global Sky Model [5], three different scenarios have been selected(1) in order to characterize different possible position of the Galactic center on the sky. For computational time reasons, each scenario takes into account only 5 astronomical sources (circled in the image) and the diffuse Galactic emission component. For each source the sky temperature, declination and right ascension are loaded along with the respective date of observation. These information are summarized into the *Source info* panel. Using the second Ludwig definition [6] the emission from the second point source can be polarized when needed(2). In the plots at the bottom, the antenna temperature(3) and the antenna directivity(4) in the source direction are shown. The latter is calculated from the isolated antenna patterns provided by IDS, considering a zero ohmic losses antenna. The antenna temperature for a source is evaluated as:

$$T_{antenna}(\nu) = T_{sky}(\nu) \cdot D(\nu) \cdot \Delta\Omega/4\pi \quad (1)$$

where D is the directivity value at the source sky coordinates and the solid angle is equal to 0.0035 sr.

In the `"Station Beam & Spectra"` tab, the user can choose the station beam coordinates (right ascension and declination), which are initially set by pointing the second source of the sky scenario selected, and the RFI use case to adopt. Three different use cases are used to represent different RFI environment conditions. The first one represents a standard spectrum, very similar to the one usually observed in the SKA-Low site. In the second use case, two strong tones at the Aircom (120 MHz) and Orbcom (137.5 MHz) characteristic frequencies are added to the standard spectrum. In the last one, the average RMS RFI level is taken into account. Furthermore, a switch allow to remove from the input spectrum the uncorrelated receiver noise, taking into account only the RFI lines. The plot at the bottom can show the power spectra at the LNA input or at the ADC input. The latter involves the effects of the LNA and RX chain on the sky signal measured by the antenna.

In figure 3, the `Polyphase filter configurator` tab is shown. The user can select the file containing the information about the polyphase filter, checking its stop band, bandpass and passband ripple. On the right, filter default values are specified. These values can be changed only when a new polyphase filter file is loaded.

In order to create the configuration file, the user must click on the yellow button in the `"Station configurator"` tab. Then, a summary of the parameter values chosen and saved into the file will pop up into the text area.

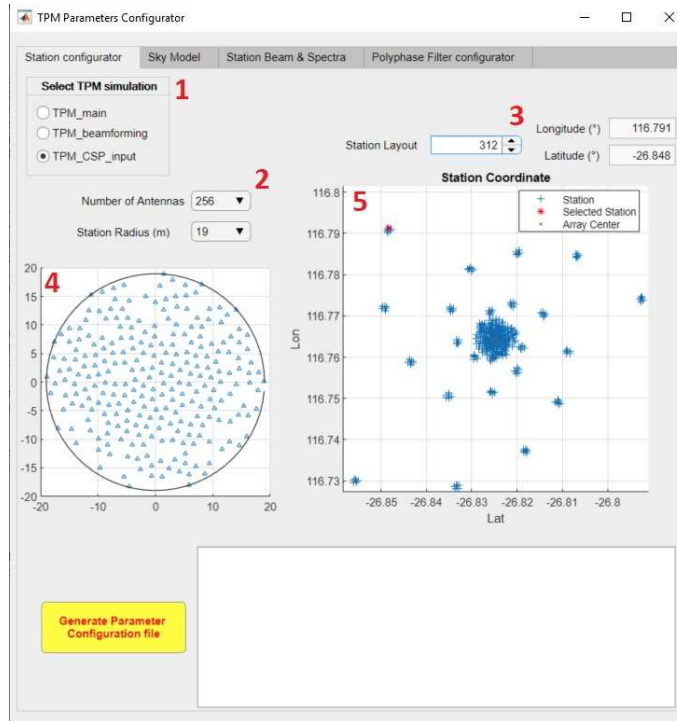


Figure 1: TPM configurator main panel

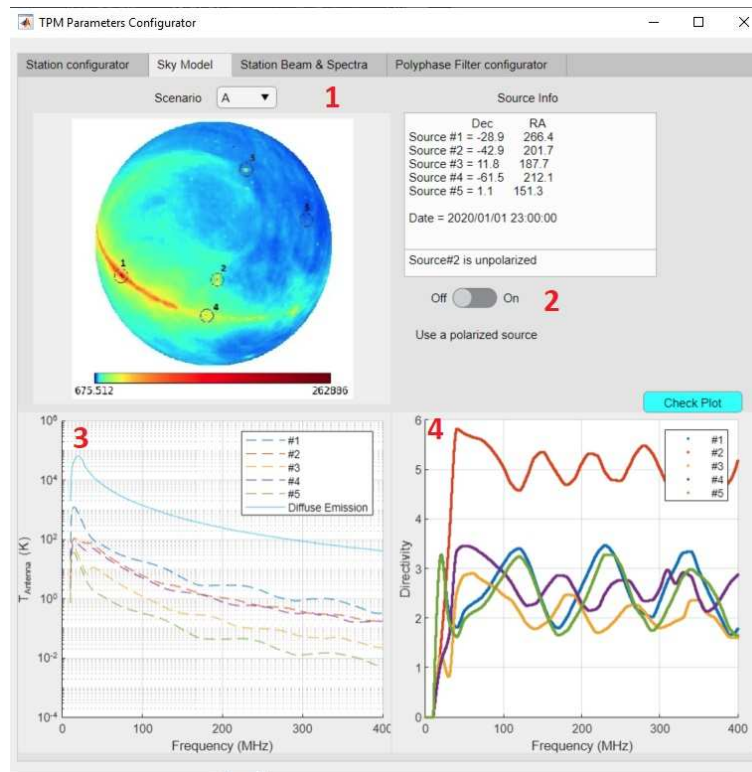


Figure 2: TPM configurator "Sky model" tab. The sky image in (1) is the map of the selected scenario at 50MHz.

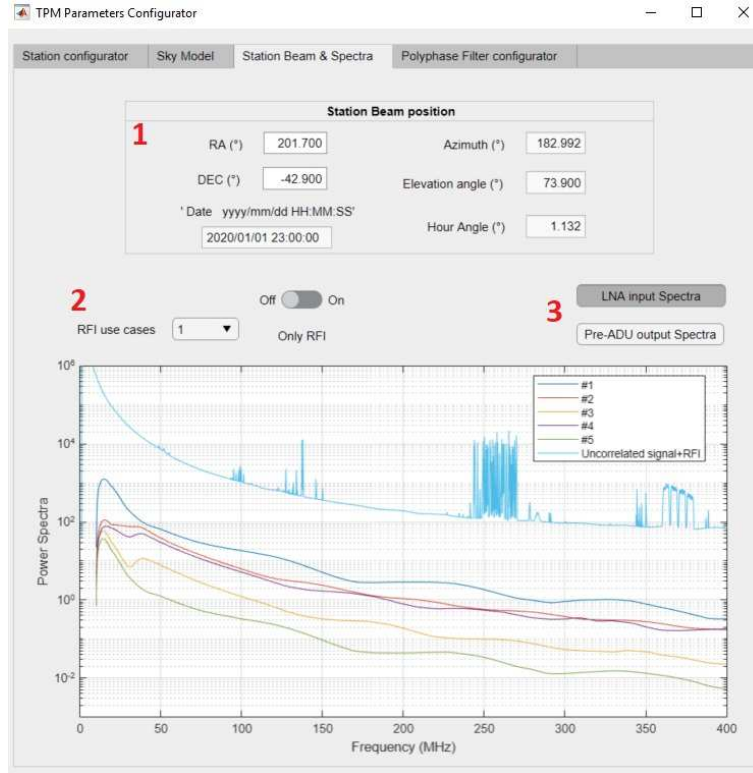


Figure 3: TPM configurator "Station Beam & Spectra" tab

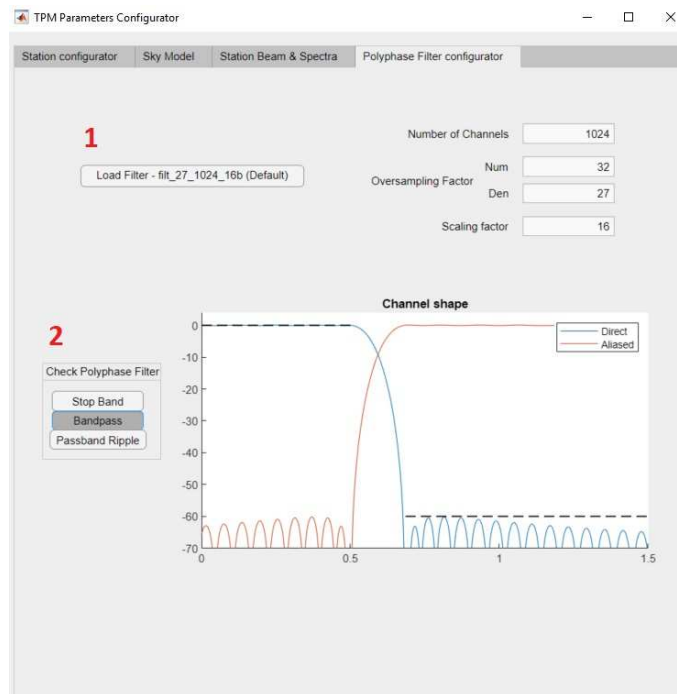


Figure 4: TPM secondary panels for the configuration of the polyphase filter

1.1.2 TPM Starter

In figure 5, the panel of the '*TPM_quick_starter*' application is shown. Here the user can run the TPM simulation by choosing it in the proper menu(1). The user can also select, from a predefined set, the number of bit of the quantization stages or choose to disable one or more of the quantizers(2). The number of antennas, the input signal amplitude in ADC unit and the simulation time used in the simulation can also be selected(3). Furthermore, by checking the *IMDs* box, it is possible to increase the effect of intermodulation products and harmonics provided by the analog signal processing chain model to an equivalent integration time of 1000 s. The RFI switch allows to add an RFI tone to the input signal by setting the RFI tone amplitude, its frequency and the position of the RFI source on the sky(4).

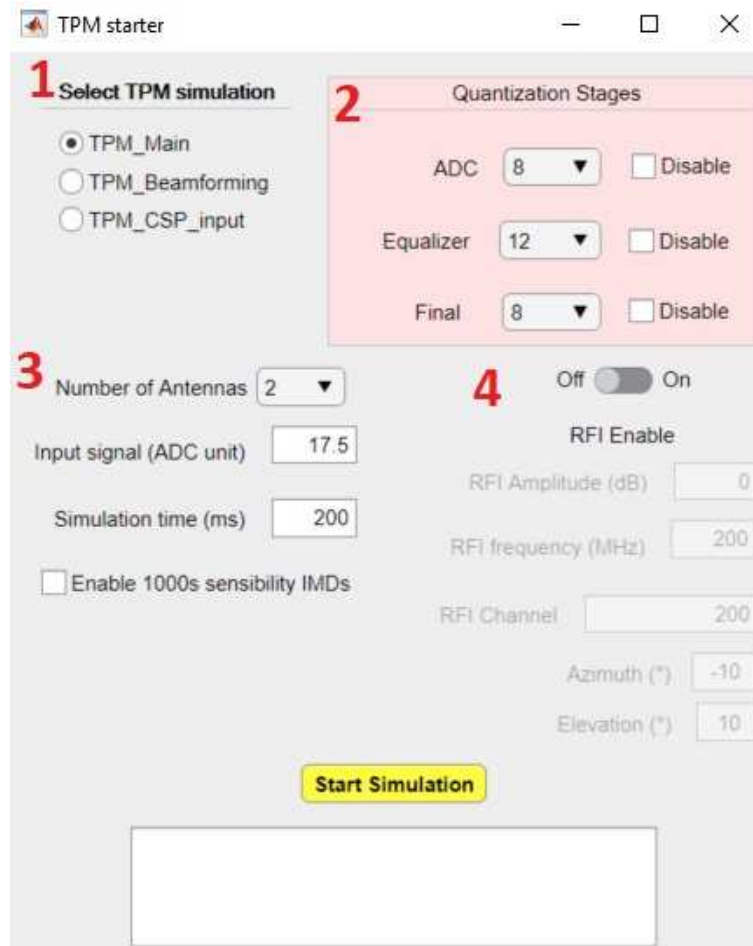


Figure 5: TPM simulation launcher panel

1.2 End-to-End Matlab Model

The diagrams shown in the following pages, illustrate the main components of the signal processing chain model. Since different model features can be analyzed using different parameter values, which may have significant repercussions on the computational time, it is given the opportunity to choose between three different kinds of TPM simulations. For each one, the user can select the value of the most relevant parameters, e.g. number of antennas, integration time and so on, that can affect the algorithm complexity.

The TPM simulations are shown in the next page. They share some of the model elements: the signal generator, the channelizer, the signal equalizer, the delay and phase correction and the calibration stage. Other two components change according to the chosen simulation. The first one performs the frequency domain beamforming while in the other one, the signal is fine channelized and correlated to allow a closer analysis. In this last stage the fine channelizer parameters have been set to reproduce as closely as possible the ones of the actual implemented correlator.

The first TPM simulation is run using '*tpm_model_main.m*' matlab function. Since components up to the beamformer can be analyzed using a small numbers of antennas, only two or three antennas may be used. This allow us to test quantization effects, intermodulation products, channelization effects and more in general the functionality of all stages before the beamforming. In the last stage, the signal is fine channelized in order to compare the autocorrelation and cross-correlation spectra with the expected ones.

The matlab function '*tpm_model_beamforming.m*' describes all the aspects of the LFAA station beamformer. It works with a number of antennas ranging from half a tile (8 antennas) to the whole station (256 antennas). Here, it is possible to estimate the beamforming loss and the final power spectrum as, after the frequency domain beamformer stage, the signal is fine channelized and autocorrelated. The simulation results are saved in a file with a proper format and a set of scripts are provided to plot and evaluate some of the most relevant quantities.

The last TPM simulation function, called '*tpm_model_CSP_input.m*', is designed to provide the proper input to the CPS Low signal processing chain model. This simulator implements the processing chain for an entire station, involving also the last 8-bit quantization after the beamforming process. The beamformed signal is saved as 8-bit integer in a binary file which is updated in each iteration.

Note that the MATLAB model in figure 6 and 7 illustrates only one of two identical signal processing chains, with and without signal quantization stages. The unquantized chain can be obtained by removing the cyan quantization blocks.

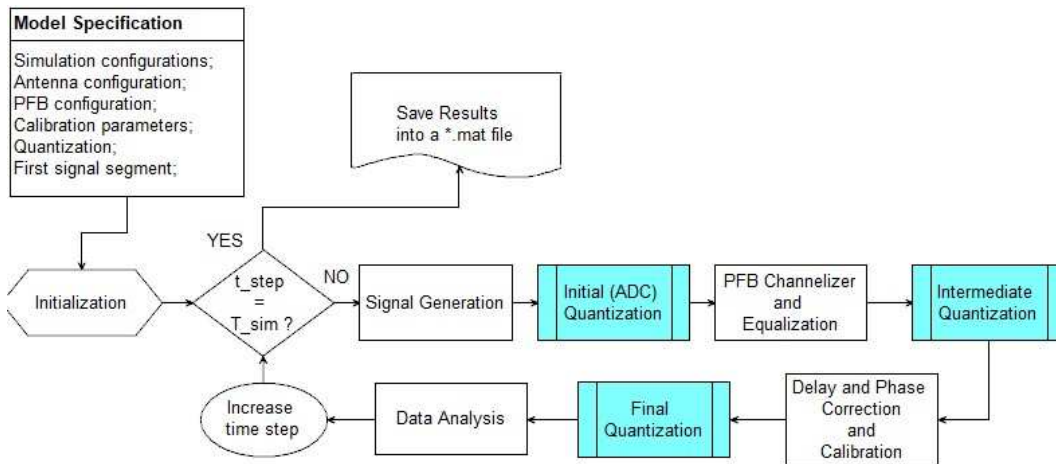


Figure 6: MATLAB "tpm_model_main" model

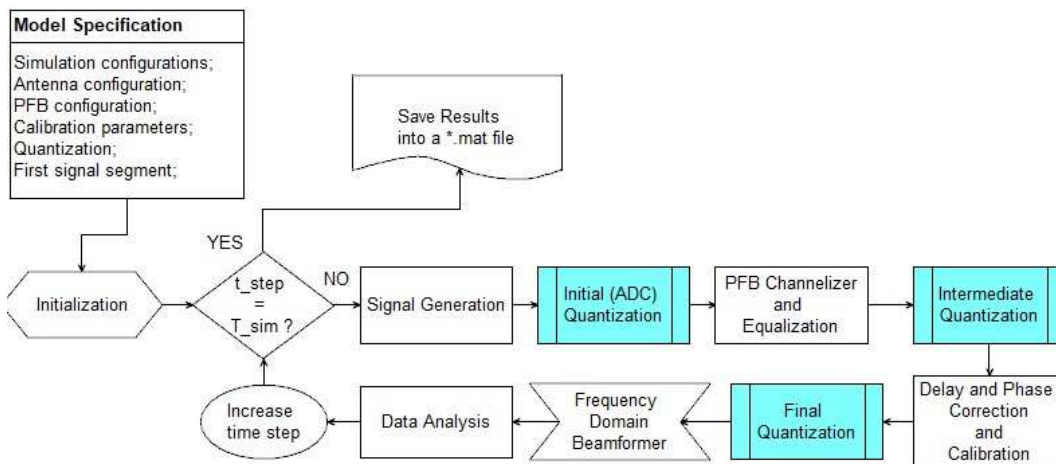


Figure 7: MATLAB "tpm_model_beamforming" model

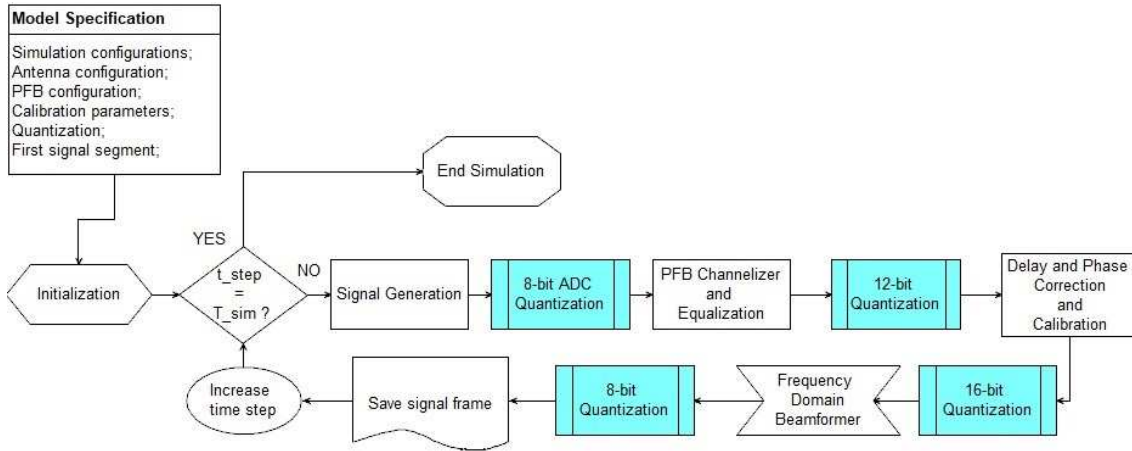


Figure 8: MATLAB "tpm_model_CSP_input" model

1.3 Parameters Specification

The *Model Specification* block takes into account the parameters generation process. Some parameters are chosen by the user, some depend on the TPM simulation selected and the remaining are fixed.

During the initialization phase, three *mat* files are loaded into the model. '*Analysis_full.mat*' and '*Analysis_128.mat*' contain the filter parameters used in the data analysis block. The last *mat* file, contains information about the antenna station, polyphase filter, beamforming process, calibration parameters and input spectra. It must be generated by the user running the script '*generate_parameters.m*' or, more simply, using the MATLAB application described in section 1.1.

Other variables can be set just before start the simulation, by modifying the '*tpm_starter.m*' script or, again, using the MATLAB application '*TPM_quick_starter*'.

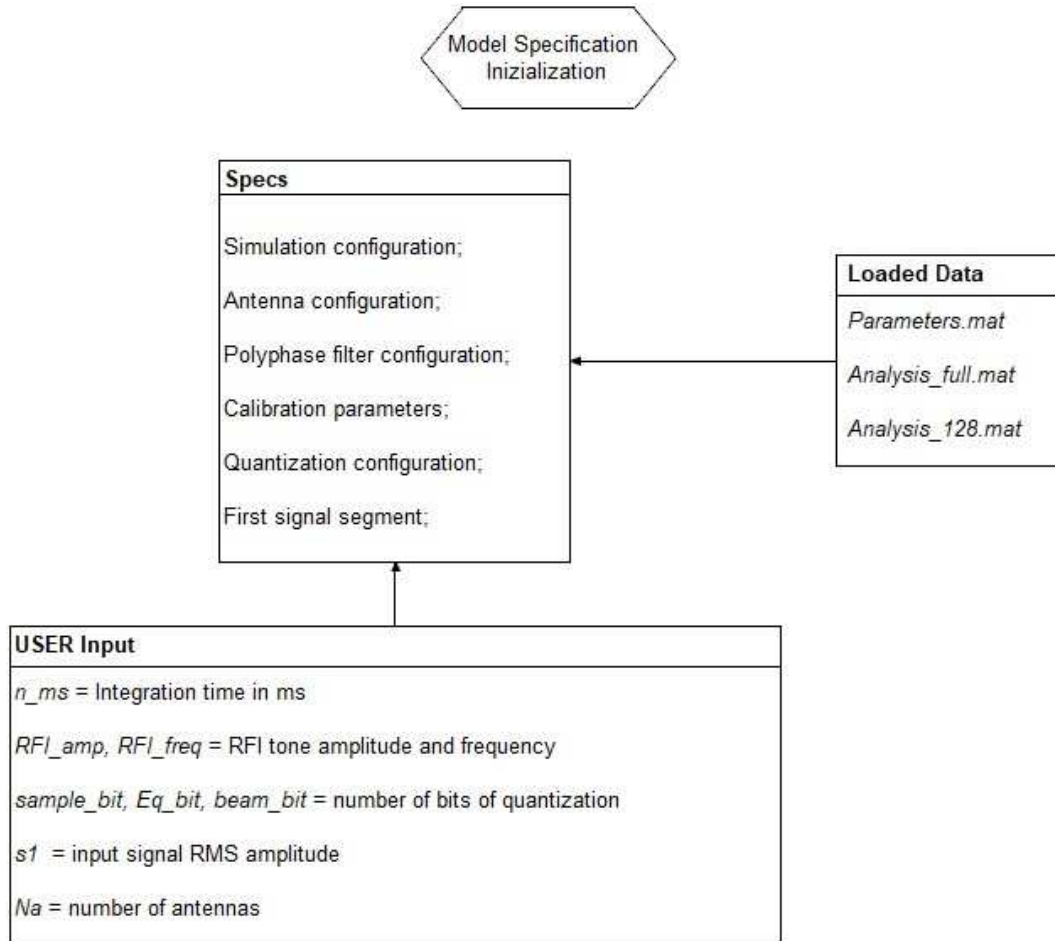


Figure 9: Parameter specifications for tpm simulations. *Analysis_full.mat* and *Analysis_128.mat* are not used in the *tpm_model_CSP_input* model.

1.4 Signal Generator

The input spectrum provided by the initialization phase, is used by the digital model to generate time sample sequences normalized to the RMS amplitude in ADC units. This is accomplished by the function *'tpm_LFAA_signal.m'* which is able to generate a vector of samples with predefined spectrum, correlation function and geometric delay. A single tone or an entire RFI model can be added to the stream of samples using a RFI signal generator. The MATLAB function *'tpm_RFI.m'* is used to generate a RFI signal. This function generates a set of tones with frequencies and amplitudes defined by the user. If the amplitude of the RFI is set to zero, the RFI generator is automatically disabled.

The signal x_Vec represents the signal at the ADC input. Harmonics and intermodulation products of the second and third order are also added to the signal just before the quantization stage.

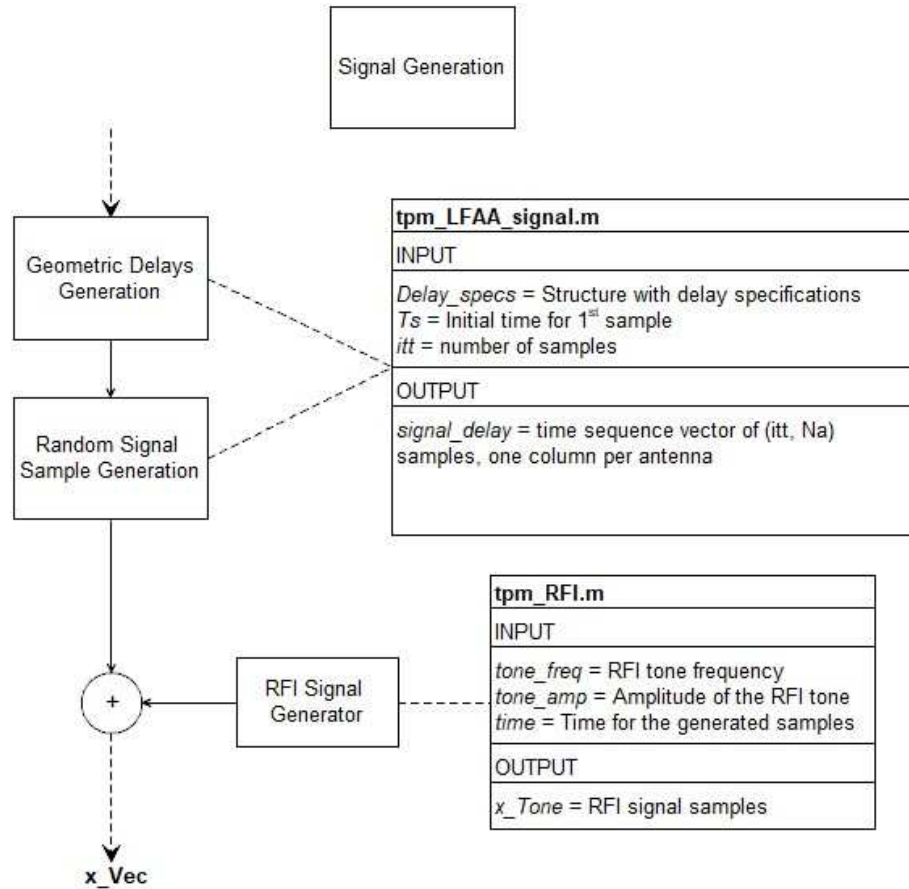


Figure 10: Signal generator model

1.5 Quantization block

As shown in figure 11, the quantization stage is preceded by a normalization block that, in the first iteration, examines the signal level, determines the appropriate scaling, and applies it to all iterations. The scaling factor is applied even if no quantization is performed, and is removed from the final result at the end of each iteration. The same quantization block is used for all instances.

The MATLAB function '*quantization.m*' performs a quantization of the signal. It quantizes the input vector to integers with an even or odd quantization scheme of n_i levels, and with a truncation at $\pm(n_i - 1)/2$. In order to disable a quantization block, the user must simply set the appropriate variable to zero during the parameters initialization.

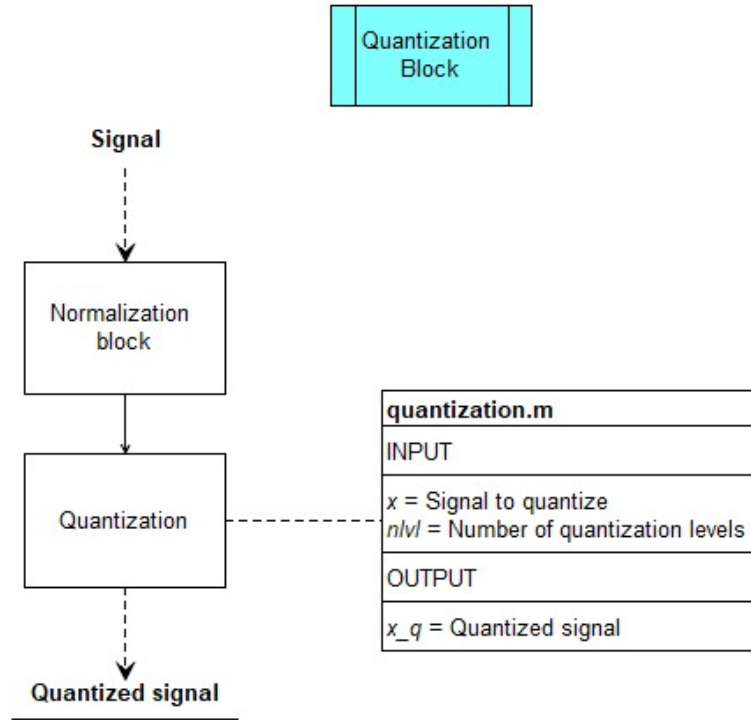


Figure 11: The signal quantization block

1.6 Signal processing chain

The real sample signal (quantized and unquantized) are channelized using a polyphase filter (figure 12). The output signal is a complex sample vector with 512 channels. The function '*pfb_real.m*' simulates the polyphase filter bank with an overlapping between input and output. The function compensates the phase shift due to overlapping, rotating the output samples. In this way, the following FFT time origin is realigned with the filter input samples.

The channelized signal is then equalized. In the equalization block the normalization and calibration coefficients are computed. These values are computed only at the beginning of the simulation, on the first segment of channelized data, and are used for all subsequent data across the simulation. The equalization factors are propagated downstream, to be removed in the data analysis while the normalization factor provides the appropriate level of RMS amplitude for the intermediate and final quantization.

The signal is corrected for the delay and phase and then calibrated (figure 13). The phase correction is computed multiplying the geometric delay by the channel frequency. The delay, in turn, is approximated using a linear function of time from an initial delay and a delay slope.

Note that, from now on, the quantized signal and its path, is explicitly specified in the diagrams, by appending the letter '*q*' to the signal vector name.

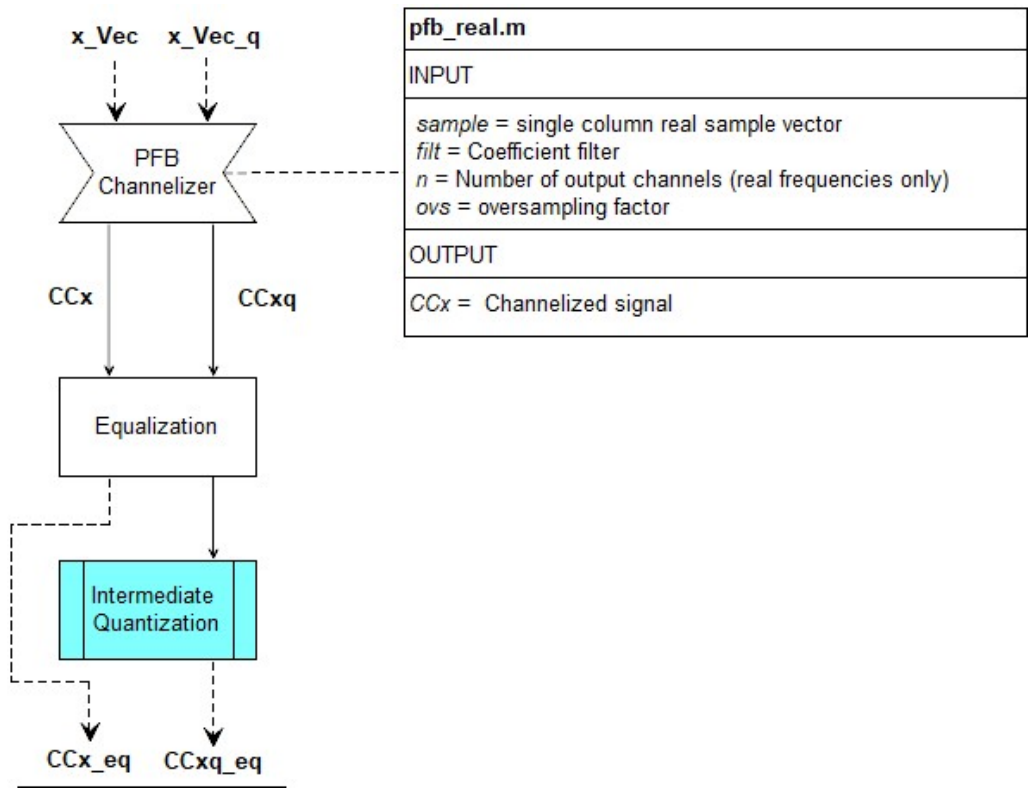


Figure 12: Channelization and equalization modules

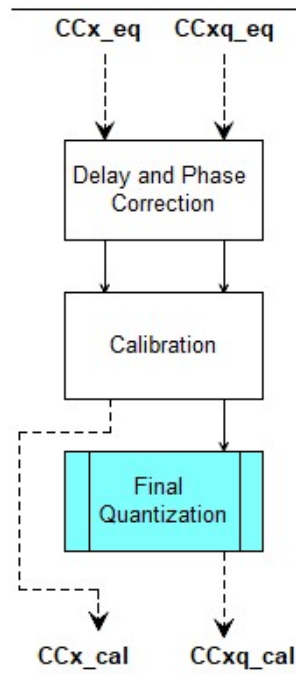


Figure 13: Delay-phase correction and calibration modules

1.6.1 Frequency domain beamformer

According to the tpm simulation chosen, the signal can follow different path. When the *'tpm_model_beamformer.m'* or the *'tpm_model_CSP_input.m'* function is run, the signal is processed in the frequency domain beamformer, as shown in figure 14.

The partial sum block sums the quantized signal vector in group of 8 antennas. For this reason, the number of antennas must be a multiple of 8, otherwise an exception is raised by the running script. This constraint is already fixed if the user starts the simulation using the *'TPM_quick_starter'* application. Then, the signal is rescaled and quantized using a 12-bit quantizer. In the last block, the quantized and unquantized signal are finally beamformed.

After the frequency domain beamformer stage, the *'tpm_model_CSP_input.m'* model performs the last 8-bit quantization and save the real and imaginary part of the working signal frame in two binary files (*'CSP_input_real.bin'* and *'CSP_input_imag.bin'*). In the *'CSP_input_time.bin'* file the time reference for each sample are also saved. This last operation concludes the signal processing chain for the *'tpm_model_CSP_input.m'* model.

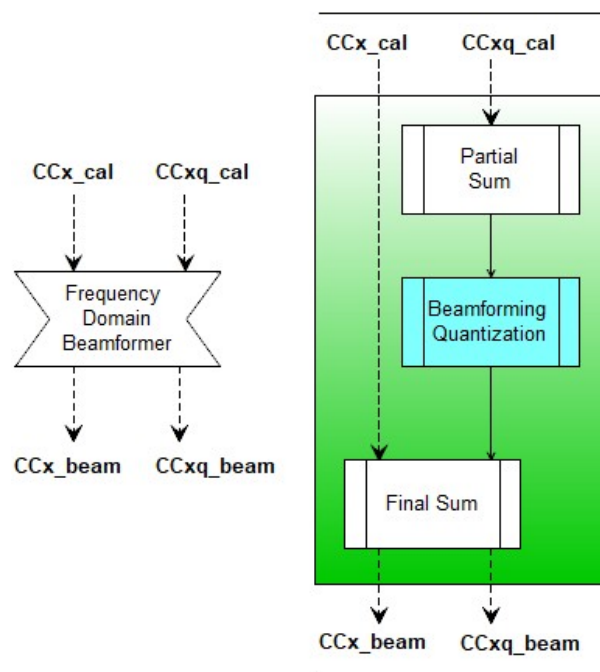


Figure 14: Beamforming process block

1.7 Data analysis

In the data analysis stage the '*composite_spectrum.m*' function computes a fine channelization of the signal, simulating the data processing in the CSP, with a simplified but realistic model of the zoom mode imaging correlator and of the final calibration procedure. All instrumental effects due to signal processing are removed, together with the antenna and receiver gain assumed in signal generation. The rescaling factors used to optimize the quantization are also removed. The final resolution is that of the zoom mode CSP channelizer. The channel outputs are then accumulated until the number of iteration *N_{itt}* is reached.

composite_spectrum.m
INPUT
<i>samp</i> = signal matrix <i>filt_coeff</i> = polyphase filter coefficients <i>norm</i> = normalization factor to come back to calibrated data <i>Analysis_par</i> = filter parameters <i>gain_m</i> = mean amplitude gain for all antennas
OUTPUT
<i>x</i> = matrix containing autocorrelation and crosscorrelation spectra

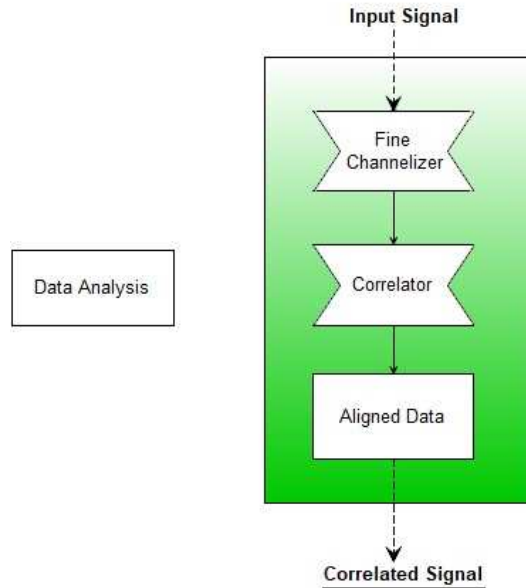


Figure 15: Data analysis module. Applies only for '*tpm_model_beamforming.m*' and '*tpm_model_main.m*' simulation.

1.8 Results

After the simulation, the result is divided by the total number of iterations executed and then it is saved in a MATLAB file. The file name is derived from the TPM simulation chosen and from a few parameters used in the simulation. The list of the results included in the file are given in figure 16.

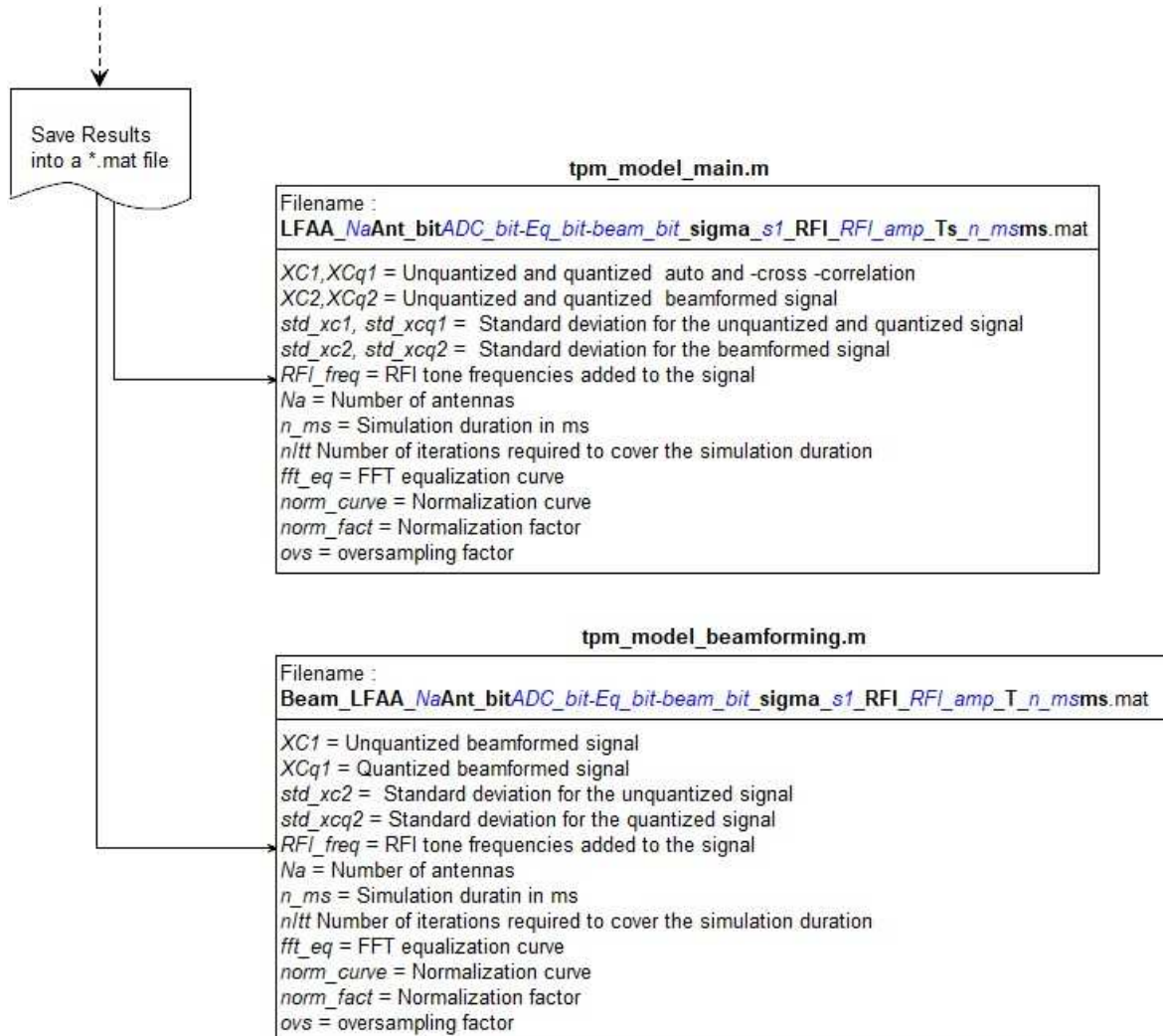


Figure 16: Two different file names are possible. Final results are stored and ready to be analyzed