

Verification methodology for the IMT-Advanced channel simulator

Kyung-Won Kim¹, Jae-Joon Park², Sungroh Yoon³,
and Seong-Jun Oh^{1a)}

¹ College of Information & Communication, Division of Computer and Communication Engineering, Korea University, Seoul, 136–701, Korea

² Electronics and Telecommunication Research Inst. (ETRI), Daejeon, Korea

³ School of Electrical Engineering, Korea University, Seoul, 136–713, Korea
a) seongjun@korea.ac.kr

Abstract: In order to reflect a realistic wireless channel as closely as possible, the channel model becomes very complicated and is thus prone to error when implemented as a channel simulator. However, there is no rigorous way of verifying the implementation of the channel simulator. We propose a novel verification methodology that aims at implementing a verified channel simulator for the IMT-Advanced channel model. We first derive a systematic description of the channel model, where a random waveform is generated through various deterministic functional blocks and independent random number generators. We then verify the deterministic functional blocks by adopting the vector-based software verification methodology.

Keywords: channel model, channel simulator, communication standard, 4G system, IMT-Advanced, ITU-R

Classification: Science and engineering for electronics

References

- [1] ITU-R Rep. M.2134, *Requirements related to technical performance for IMT-Advanced radio interface(s)*, 2008.
- [2] ITU-R Rep. M.2135, *Guidelines for evaluation of radio transmission technologies for IMT-Advanced*, 2008.
- [3] ITU-R Rep. M.1225, *Guidelines for evaluation of radio transmission technologies for IMT-2000*, 1997.
- [4] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical recipes: the art of scientific computing, Third Edition*, Cambridge University Press, 2007.
- [5] Finland and China (People's Republic of), *Software Implementation of IMT.EVAL Channel Model*, Contribution 5D/313, presented in ITU-R WP 5D #3 in Seoul, Oct. 2008.
- [6] Korea (Republic of), *C-Code for Rep. ITU-R M.2135 – Channel model implementation*, Contribution 5D/476, presented in ITU-R WP 5D #5 in Geneva, June 2009.

1 Introduction

In order to meet the explosive demands of high-speed packet-based wireless communication services, International Telecommunication Union Radiocommunication Sector (ITU-R) has announced a plan for the 4G cellular system standard – International Mobile Telecommunications-Advanced (IMT-Advanced) system. A few candidate radio interface technologies (RITs) or a set of RITs (SRITs) are expected to be proposed to ITU-R as one of the 4G systems. It will then acknowledge one or a few RITs/SRITs as the IMT-Advanced systems, if the proposed technology meets the requirements defined in [1]. For the fair evaluation of various candidate RITs/SRITs, ITU-R has defined a common evaluation methodology, M.2135 [2] for IMT-Advanced systems, where the multi-path fading channel model for the simulation can be found.

In the past, the channel model [3] of ITU-R defined only the statistical characteristics of the random process (i.e., the power delay profile of the channel impulse responses for a few scenarios) assuming that the channel impulse response is a wide sense stationary random process. Therefore, a correct implementation can easily be checked by obtaining the second order statistics of the generated random waveform. However, in order to reflect the real wireless channels in several environments as closely as possible, the channel model recently announced in [2] is described in terms of how to generate a random waveform rather than the statistical characteristics of the random process. This new model is straightforward to implement, but there are many steps involved and many random numbers have to be generated. Therefore, errors in implementing the channel may not be found by observing the random waveform, and they can be critical for the evaluation of a communication technology.

In the vector-based verification, the original design is independently developed by the software under test (SUT) and by the reference software. Then, the same input (i.e., a set of random input vectors) is applied for both to check whether or not the same outputs (i.e., output vectors) are obtained from SUT and from the reference software. The channel implementation process is conducted by separating the deterministic functional blocks and the independent random number generator blocks. Then, we apply the vector-based verification methodology only to the deterministic functional blocks. As the functional block analysis of the channel implementation is done up to a level where only the basic random number generations are needed, we can use the well-known algorithms in [4] or a compiler-built-in random number generator for the basic random number generators without verification.

2 IMT-Advanced channel model

M.2135 [2] provides the simulation methodology and required configurations including the channel model – the IMT-Advanced channel model. Four nominal deployment scenarios are considered, namely indoor hotspot (InH), urban micro-cell (UMi), urban macro-cell (UMa) and rural macro-cell (RMa) sce-

narios.

Fig. 1 shows the overall steps for the channel model implementation. First, the deployment scenario is selected, and the network layout and the antenna configuration are then determined. In the second stage, large-scale (LS) and small-scale (SS) parameters are defined. In the third stage, channel impulse responses (ChIRs) are computed. There are three levels of randomness. First, LS parameters are drawn randomly from the distribution functions. Next, SS parameters are drawn randomly according to the tabulated distribution functions and randomly determined LS parameters. At this stage, all the geometric setups are fixed, and only the leftover randomness occurs in the random initial phases of the scatterers.

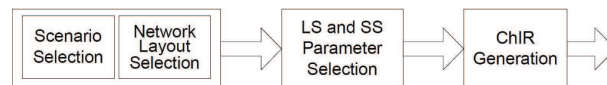


Fig. 1. ChIR Generation Process

As the channel implementation is to instantiate the time-varying impulse response of a linear time-varying system, consider the channel as a linear system whose output at time τ , when the impulse is applied at time t , is represented as

$$H(t; \tau) = \sum_{n=1}^N \left(\sum_{m=1}^M H_{n,m} \cdot \exp(j2\pi\nu_{n,m}t) \right) \cdot \delta(\tau - \tau_n). \quad (1)$$

$H(t; \tau)$ is the summation over the impulse responses of the clusters, where n is the cluster index, N is the number of the clusters, m is the ray index, and M is the number of rays per cluster. The cluster (or path) is defined as the collection of rays with the same delay, and therefore the impulse response of each cluster is again represented by the summation of a number of rays with a common delay. $H_{m,n}$ is the time-invariant part of $H_{m,n}(t)$, and $\nu_{n,m}$ is the Doppler frequency component of ray n of cluster m . All the time-invariant numbers (i.e., $\tau_{m,n}$'s, $H_{m,n}$'s and $\nu_{n,m}$'s) are called the channel coefficients, and they are randomly determined per drop at its initialization.

The channel impulse response of a radio link between a mobile station and a base station is a random waveform, generated according to the steps shown in Fig. 1. Due to the page limitation, in the following sections, we focus only on the SS parameter generation in the next sections. The same separation principle, deterministic functional blocks and independent random number generations, can be applied to other blocks as well.

3 Functional blocks for the SS parameters' generation

After analyzing the steps of generating the SS parameters' in [2], the equivalent block diagram can be drawn as shown in Fig. 2 for a single transmit-receive antenna pair. Note that the deterministic parts, shown in rectangular boxes and the independent random numbers' generations, shown in bubbles,

are separated. As the channel coefficients are obtained per cluster, the same random numbers as the number of clusters should be independently generated. As in (1), denote the number of clusters by N ; then the result depends upon the scenario types and propagation condition.

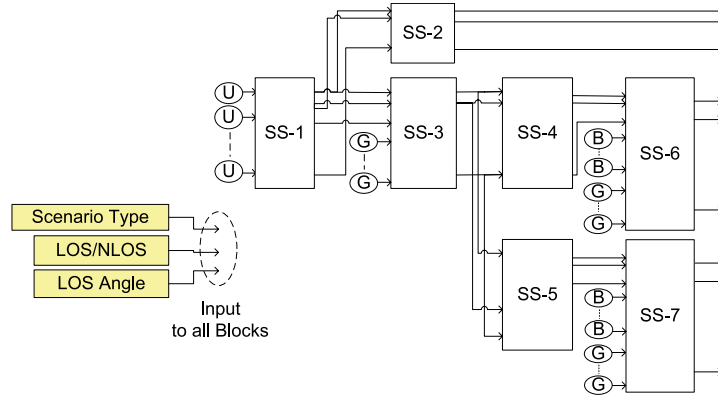


Fig. 2. Equivalent block diagram for generating SS parameters in [2]

SS-1 generates N ordered delays from N uniform $[0, 1]$ random variables by

$$\tau'_n = -r_\tau \sigma_\tau \ln(X_n) \quad \text{and} \quad \tau_n = \text{sort}(\tau'_n - \min\{\tau'_1, \tau'_2, \dots, \tau'_N\}) \quad (2)$$

for $n = 1, 2, \dots, N$, where r_τ is the delay distribution proportionality factor given in Table A1-7 of [2], X_n 's are uniform random variables distributed between 0 and 1 (set by U's), and n is the cluster index. σ_τ is generated by Corr_τ as

$$\sigma_\tau = \mu_{DS} + \text{Corr}_\tau \cdot \sigma_{DS} \quad (3)$$

where μ_{DS} is the expected value and σ_{DS} is the standard deviation of σ_τ . Corr_τ is generated from the correlation generation as a LS parameter, and μ_{DS} and σ_{DS} are given in Table A1-7 of [2] according to the scenario type and propagation condition. τ'_n 's are the unordered delays, and τ_n 's ordered delays with τ_1 being the minimum of zero.

SS-2 divides all the delays from block SS-1 in the case of LOS by D , where

$$D = 0.7705 - 0.0433K_R + 0.0002K_R^2 + 0.000017K_R^3 \quad (4)$$

and K_R is the Ricean K -factor that is generated by

$$K_R = \mu_K + \text{Corr}_K \cdot \sigma_K \quad (5)$$

where μ_K is the expected value and σ_K is the standard deviation of K_R given in Table A1-7 of [2]. For NLOS, the delays are bypassed in SS-2.

SS-3 is the normalized power generation block. It first generates the un-normalized powers for each cluster by

$$P'_n = \exp\left(-\tau_n \frac{r_\tau - 1}{r_\tau \sigma_\tau}\right) \cdot 10^{-\frac{Z_n}{10}} \quad (6)$$

for $n = 1, 2, \dots, N$, where Z_n is the zero-mean Gaussian random variable representing the per-cluster shadowing (in dB), whose variance is defined in Table A1-7 of [2] according to the scenario type and the propagation condition. Other numbers and parameters in (6) are given from SS-1 or are computed according to Table A1-7 of [2]. The powers are normalized so that the summation of all cluster powers is equal to one, and the normalized powers become the outputs of SS-3.

SS-4 and SS-5 generate the arrival angle (φ') spread and departure angle (ϕ') spread, respectively. As the underlying assumptions on the distributions are different for the InH and other scenarios, different equations are used. φ' and ϕ' are generated as

$$\varphi'_n = \frac{2\sigma_\varphi \sqrt{-\ln(P_n/\max(P_m))}}{1.4 \cdot C} \quad \text{and} \quad \phi'_n = \frac{2\sigma_\phi \sqrt{-\ln(P_n/\max(P_m))}}{1.4 \cdot C} \quad (7)$$

for $n = 1, 2, \dots, N$, and for all scenarios, except the InH scenario. For the InH scenario, they are computed as

$$\varphi'_n = -\frac{\sigma_\varphi \ln(P_n/\max(P_m))}{C} \quad \text{and} \quad \phi'_n = -\frac{\sigma_\phi \ln(P_n/\max(P_m))}{C} \quad (8)$$

for $n = 1, 2, \dots, N$, where σ_φ and σ_ϕ in (7) and (8) are the standard deviations of the arrival and departure angles, respectively. 1.4 in (7) is the ratio of Gaussian standard deviation to the corresponding root mean square (RMS) spread. σ_φ and σ_ϕ are generated by using Corr_φ and Corr_ϕ as

$$\sigma_\varphi = \mu_{AoA} + \text{Corr}_\varphi \cdot \sigma_{AoA} \quad \text{and} \quad \sigma_\phi = \mu_{AoD} + \text{Corr}_\phi \cdot \sigma_{AoD} \quad (9)$$

where μ_{AoA} and μ_{AoD} are the expected values, and σ_{AoA} and σ_{AoD} are the standard deviations for the angle of arrival and departure, respectively. They are given in Table A1-7 of [2]. Constant C in (7) and (8) is the scaling factor given in Table A1-4 of [2] according to the scenario type and propagation condition.

SS-6 and SS-7 generate the arrival and departure angles by using the angle spreads obtained from SS-4 and SS-5, respectively. For the case where the LOS component does not exist, the departure and arrival angles are generated as

$$\varphi_n = X_n \varphi'_n + Y_n + \varphi_{LOS} \quad \text{and} \quad \phi_n = X_n \phi'_n + Y_n + \phi_{LOS} \quad (10)$$

and for the case where there is a LOS component, they are generated as

$$\begin{aligned} \varphi_n &= X_n \varphi'_n + Y_n - (X_1 \varphi'_1 + Y_1 - \varphi_{LOS}) \quad \text{and} \\ \phi_n &= X_n \phi'_n + Y_n - (X_1 \phi'_1 + Y_1 - \phi_{LOS}) \end{aligned} \quad (11)$$

where X_n 's are independent and identically distributed (iid) binary random numbers taking +1 or -1 with equal probabilities, and Y_n 's are iid zero-mean Gaussian random variables with variances of $(\frac{\sigma_\varphi}{7})^2$ and $(\frac{\sigma_\phi}{7})^2$, respectively, for AoA and AoD. Note that X_n 's and Y_n 's for AoA and AoD are independently generated. φ_{LOS} and ϕ_{LOS} are LOS directions, but when we consider an omni-directional antenna at the mobile station, φ_{LOS} and ϕ_{LOS} are assumed to be the same as the LOS angle.

4 Verification with a reference code – a case study

The verification of channel simulator is performed as follows. First, our own C-based channel implementation can be developed according to M.2135 [2], and this software is used for SUT. Then, the Matlab code of the same implementation, publicly available from [5], is used as reference software. After identifying the deterministic functional blocks, SS-1 – SS-7 in Fig. 2 in the reference software, the deterministic functional blocks of SUT are verified against the randomly generated input vector. Two output vectors, one from SUT and the other from the reference software, for the same random input vector, should be exactly the same.

After all the deterministic blocks are verified, the probability density functions for each block’s output are compared for the C-based SUT and Matlab code. A correct implementation is justified by checking the probability density functions (pdf). Fig. 3 shows pdf for SS-6 block. The lines indicate the output from C-based SUT, and the dots represent the output from Matlab. The pdf’s from two software implementations are the same; the verification of the C-based SUT is justified.

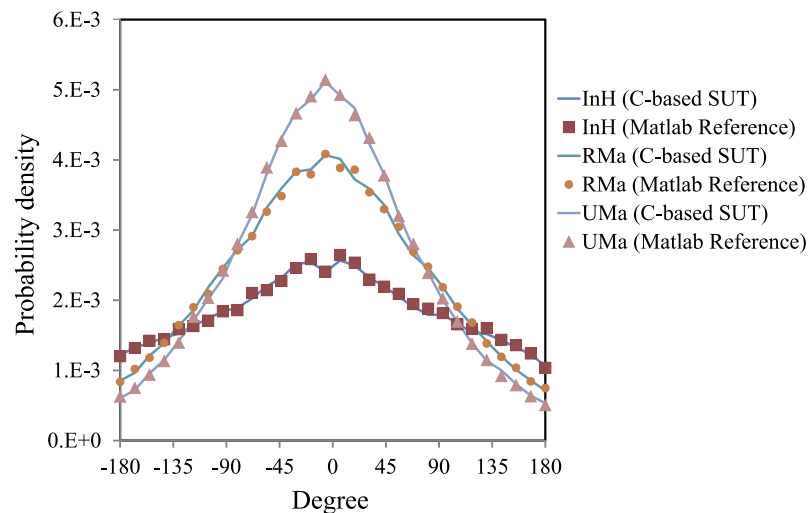


Fig. 3. AoA’s pdf obtained from SS-6 for the LOS case

5 Conclusion

We have proposed a vector-based software verification methodology for the IMT-Advanced channel simulator. It is based on M.2135, an ITU-R report for the evaluation of the candidate technologies for the IMT-Advanced systems.¹ As the channel simulator generates a random waveform, the traditional vector-based verification methodology should be modified by separating the deterministic functional blocks and the random number generators. The vector-based verification should be applied only to the separated deterministic blocks. The channel simulator verified by the proposed methodol-

¹The C-based IMT-Advanced channel simulator verified against [5] is also found as a contribution in ITU-R WP5D [6].

ogy can be used for evaluating various candidate technologies for the IMT-Advanced systems and other wireless communication systems.

Acknowledgments

This research was supported by the MKE (The Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency (NIPA-2011-(C1090-1121-0011))).