Novel ray-tracing approach for fast calculation of the impulse response on diffuse IR-wireless indoor channels

Article *in* Proceedings of SPIE - The International Society for Optical Engineering · January 1999

3 authors:

Francisco José López Hernández
Universidad Politécnica de Madrid

Rafael Pérez-JImĕnez
Universidad de Las Palmas de Gran Canaria

Asunción Santamaría
Universidad Politécnica de Madrid

Novel ray-tracing approach for fast calculation of the impulse response on diffuse IR-wireless indoor channels

F.J. López-Hernández [a], R. Pérez-Jiménez [b], A. Santamaría [c]

[a]E.T.S.I.T. Dpto. T. Fotónica. U.P.M. Ciudad Universitaria s/n. 28040 Madrid. Spain
[b]Grupo de T. Fotónica y Com., D. SS.CC., ULPGC. Campus de Tafira, 35017, Las Palmas, Spain
[c]E.T.S.I.T. Dpto. Electrm. y T. Circuitos. U.P.M. Ciudad Universitaria s/n. 28040 Madrid. Spain

ABSTRACT

In this paper, a modified Monte Carlo algorithm for the calculation of the impulse response on infrared wireless indoor channels is presented. This work follows a guideline of studies about the infrared wireless diffuse data communications systems. As is well known, the characteristics of the room where the IR diffuse channel is implemented determine some problems in the communication as can be multipath penalty over the maximum baud rate or hidden station situations. Classical algorithms[1-2] require high computational effort to calculate the impulse response in a regular size room. Monte Carlo offers the possibility of validating the assumptions made for these classic algorithms (basically, the lambertian nature of all reflections) with a computational complexity that is decided by the accuracy desired by the user. It is also an structure that can be easily assumed by a parallel computer architecture. In the other hand, its main drawback is that, for a regular sized room, we need to send much more rays than the components that we receive. This is due to the fact that usually rays are not intercepted by the receiver. We have developed a mixed Monte Carlo-Deterministic algorithm which assures that each ray contributes to the final channel response function each time it rebounds with an obstacle. It increases dramatically the number of contributions and reduces, in the same way, the time required for an accurate simulation.

Extensive simulation results are presented. They are compared both with other simulation methods and with measured values. We will demonstrate that the method presented here is much faster than Monte Carlo classical simulation schemes. It can be used like a method of simulation itself or as a validation algorithm for other comparative studies of pulse broadening.

Keywords: simulation, Monte Carlo, wireless optical transmission, ray-tracing

1. INTRODUCTION:

Multipath dispersion is the most constrictive effect of indoor optical wireless transmission. If good coverage is needed, diffuse transmission is the best choice[3]. This implies the use of one, or several, high power emitters placed on the ceiling or walls. As the signal can take several paths before reaching the receiver, intersymbol interference (ISI) will limit the maximum data rate.

Several deterministic methods have been proposed[1-2], but all of them share the same problem, the intensive calculation effort, especially when time resolution is lower than one nanosecond for a normal sized room. All of the calculations are based on the impulse response of the room, i.e. the time evolution of the signal received by the detector when an infinitely short light pulse is launched by the emitter. Once the impulse response is known, based on the linearity of the optical system, any waveform distortion due to multipath propagation can be calculated. It is noticeable that the impulse response of the room, or its transfer function, depends on the positions and the orientations of the emitter and the receiver, and not only of the characteristics of the room. This implies that for the perfect knowledge of the room effect on an optical transmission, many simulations have to be made. For a normal sized room, the time needed for one simulation, using deterministic models, can vary between several hours or days, so the capacity of prediction of these models is strongly handicapped. Other statistical method has been proposed[4], by it only calculates a rough approximation to the impulse response.

On the other hand, the method proposed takes only several minutes to calculate the room dispersion. There is another problem with deterministic models, they only take into account the scattering when light reaches a wall. For glazing angles there is a strong mirror reflection with a quite different behavior. Of course, if there are polished surfaces such as glasses or mirrors, the reflection is dominant over scattering at any angle. Our model implements both scattering (Lambert) and mirror reflections so its results are closer to real rooms than the deterministic ones.

Part of the SPIE Conference on Optical Wireless Communications II
Boston, Massachusetts • September 1999
SPIE Vol. 3850 • 0277-786X/99/$10.00

2. DESCRIPTION

This is a ray tracing method[7]. Many rays are generated at the emitter position having a distribution probability equal to the emission profile, or angular optical intensity function. When a ray impinges on an obstacle (wall, ceiling, etc), the point where it reaches the obstacle is converted in a new optical source, thus a new ray is generated, and the process continues until the time of flight (counted from the generation on the emitter) reaches the maximum time to simulate (t_{max}).

After every reflection the power of the ray is reduced by the reflection coefficient of the obstacle, so two variables are associated with every ray: the time from its generation, and the power it carries.

On a Monte Carlo simulation, there is a tiny probability for the ray reaching the receiver before the maximum simulation time[5]. When this happens, the power carried by the ray is detected with the delay of the trip. But the probability of this is very low, about 10^{-7}, so many million of rays have to be generated to get a reliable result. The chances are very different if we use the fact that we know the contribution of a scatterer surface over the receiver. In this way, when the ray impinges on an obstacle, not only a new ray is generated, but the reflected power contribution to the receiver is calculated. In this case, we have changed from using only one of many of million of rays, to use every ray generated several times (as many as reflections). Of course, before generating a new ray the direct emitter-to-receiver contribution is added up. The algorithm is presented in table 1.

Table 1. Basic algorithm

Begin

1. Generate a new ray.(t=0, P=1) Calculate direct contribution (same for all the rays)
2. **Loop while** $t<t_{max}$
   Propagate ray until any obstacle (t = t + d/c)
   Reduce power using the reflection coefficient (P=αP)
   Calculate the contribution from that point to the receiver
   Generate ray from the new point.
   End loop
3. Repeat steps 1 and 2 for a number of rays until variance (noise) be acceptable

End

Next we will describe the equation used at every step.

1.1. Emitter

The emitter is defined by its position and orientation ($\vec{r}_e, \hat{n}_e$), and its intensity profile ($I(\theta)$), where cylindrical symmetry around the normal to the emitting surface has been assumed. Every launched ray will have a normalized unit power. After the simulation, this allows the renormalization the power emitted by all of the rays. As the system is linear and time invariant, there is no problem doing in it so. If several emitters with different powers are to be simulated, superposition can be applied.

The main problem with the emitter is generating rays followed a given intensity profile. The most used approximation is the extended Lambertian profile (equation 1.1),

$$I = I_0 \cos^n(\theta) \tag{1.1}$$

where the exponent defines the width of the beam. For n=1, the Lambertian scattering profile is obtained. The most significant characteristic of extended Lambertian is that the maximum intensity direction is always normal to the emitter surface. Several device cases or arrays do not fulfill this condition and this model is useless. So both cases: a) extended Lambertian, and b) general have been studied.

a) Extended Lambertian

This case includes the Lambertian scattering (n=1). The axis definition is given in figure 1, where the emitter surface is the XY plane. As defined in equation (1.1), the main random variable is θ. We use the cumulative probability, i.e. the probability of a ray having a θ value lower or equal to Θ is:

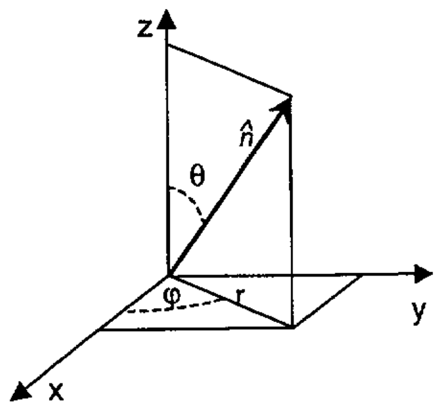

Figure 1. Coordinate system.

$$P(\theta \le \Theta) = \int_0^{2\pi} d\varphi \int_0^{\Theta} I(\theta)\sin(\theta)d\theta$$

Or

$$P(\theta \le \Theta) = 1 - \cos^{n+1}(\Theta) \qquad (1.2)$$

By using the components r and z, which are

$$r = \sin(\theta)$$
$$z = \cos(\theta) \qquad (1.3)$$

As z and θ change monotonically in the opposite way, we have

$$P(z \ge Z) = P(\theta \le \Theta)$$
$$P(z \le Z) = 1 - P(z \ge Z) = Z^{n+1} \qquad (1.4)$$

To get a random unitary vector in which the z component has a cumulative probability given by (1,4), two random numbers are generated having a uniform probability in the range [0,1). Let u and v be these numbers, the sequence is:

$$z = \sqrt[n+1]{u} \qquad r = \sqrt{1-z^2}$$
$$x = r\cos(2\pi v) \qquad y = r\sin(2\pi v) \qquad (1,5)$$

Where x, y and z are the components of a unitary vector relative to a coordinate system normal to emitter surface.

b) General profile

Let be $I(\vartheta)$ a tabulated or analytical function. As before,

$$P(z \le Z) = 1 - 2\pi \int_1^{\operatorname{acos}(Z)} I(\theta)\sin(\theta)d\theta = F(Z) \qquad (1,6)$$

Equation (1,6) produces a function $F(Z)$ which should be normalized to get $F(1)=1$. If the inverse of F(Z) is applied to a random number, generated with uniform density probability in [0,1), the required z component is achieved. The other components, x and y, are generated from z using the last two equation in (1,5).

In both cases, a) and b), the unitary vector is based on a coordinate system normal to emitter surface. It needs to be transformed into room coordinates, this is achieved by using the matrix M_e given in (1,7).

$$\hat{n}_e = \begin{bmatrix} n_{ex} \\ n_{ey} \\ n_{ez} \end{bmatrix}_{room} = \begin{bmatrix} -\sin(\phi) & \cos(\phi) & 0 \\ -\cos(\phi)\cos(\theta) & -\sin(\phi)\cos(\theta) & \sin(\theta) \\ \cos(\phi)\sin(\theta) & \sin(\phi)\sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{emitter} = M_e \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{emitter} \qquad (1,7)$$

We have arbitrarily chosen the x axis of the emitter to be on the XY plane of the room.

1.2. Receiver

The receiver, photodiode plus receiving optics, is defined by its position and orientation ($\vec{r}_r, \hat{n}_r$), detecting surface (A_r) and its field-of-view (FOV), which is the maximum angular deviation from the normal for a ray to be detected.

As stated before, any time a ray impinges on an obstacle, the contribution from that point to the receiver is calculated. If the obstacle is placed at $\vec{r}$ and its surface normal is $\hat{n}$ the contribution is given by

$$P_r = \begin{cases} \frac{A_r}{\pi d^2} \cos\phi \cos\psi \cdot P_{emitted} & ,\psi \le FOV \\ 0 & ,\psi > FOV \end{cases}$$

Where (1,8)

$$\vec{d} = \vec{r}_r - \vec{r} \quad d = \left|\vec{d}\right|$$

$$\phi = angle\,(\vec{n},\vec{d}) \quad \psi = angle\,(\vec{n}_r,\vec{d})$$

The term $cos(\phi)$ is valid for Lambertian scattering, for the direct emitter contribution this term has to be changed according to emitter intensity profile.

The power contribution P_r is delayed by d/c, value to be added to the time accumulated for that ray.

1.3. Obstacles

When a ray impinges on an obstacle two kinds of reflection are possible: mirror (MR) and scattering (SR) reflections. In a general case, both of them take place simultaneously. In the model, only a new ray is generated and the choice between MR or SR is made based on a probability which depends on the surface characteristics and incidence angle. For example, a glass window will have an MR probability equal to one (scattering neglected), while a white area on the ceiling will scatter the light but for glazing incidence angles (probability of MR≈0, for incidence angle lower than 50°).

For every different area in each obstacle surface an MR threshold function is defined. This function depends, in general, on the incidence angle of the ray. When a ray arrives to the obstacle, a random number in the range [0,1) is generated, after comparing this number with MR threshold, SR or MR is used.

The output direction is calculated as follows

Mirror reflection (MR): The incident and reflected rays are on the same plane and form the same angle with the normal to the surface (1st Snell law). If $\hat{n}_i, \hat{n}_o$, and $\hat{n}_s$ are the unitary vectors defining, respectively, the incident, reflected and surface normal directions, the Snell's Law can be expressed as the equation (1,9)

$$\hat{n}_o = \hat{n}_i - 2(\hat{n}_i \cdot \hat{n}_s)\hat{n}_s \qquad (1,9)$$

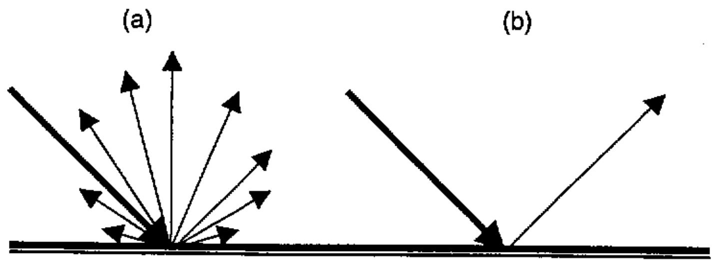

Figure 2. Scattering (a) and mirror (b) reflection

Scattering reflection (SR): The output direction is calculated using equation (1,5), with n=1.

Two reflection coefficients are defined for both SR and MR. The power of the input ray is multiplied by the appropriate coefficient to get the output power. Usually the MR coefficient is much larger than SR.

1.4. Propagation between reflections

Let $\vec{r}_0$ be the starting position of a ray, i.e. the emitter position or where last collision happened, and $\hat{n}$ the direction generated by equation (1,5) or (1,9). The vector equation describing the ray flight is[7]

$$\vec{r}(\lambda) = \vec{r}_0 + \lambda\hat{n} \qquad (1,10)$$

Where λ is the distance the ray has traveled. For several obstacles, (1,10) needs to be solved together with the equations describing them. In general, a set of λ values are obtained, one for every obstacle. It is evident that the small positive value of λ is the first obstacle reached by the ray. The value λ/c has to be added to the accumulated traveling time of the ray.

Using (1,10) the new starting point is calculated, and the obstacle characteristics are used to generate, if needed, a new ray.

As it is possible to calculate the contribution of each rebound on the total impulse response, the contribution of the first steps is presented in figure 4. The vertical units are arbitrary, but the scale is the same for all of them, so their relative importance can be known based on the values in the charts.

It is also noticeable that for high number of rebounds, the noise, for the same number of rays, is larger. Fortunately, their influence in the final result is lower because their low power contribution. Further refinement can be obtained by numerical filtering methods, such as moving averaging.

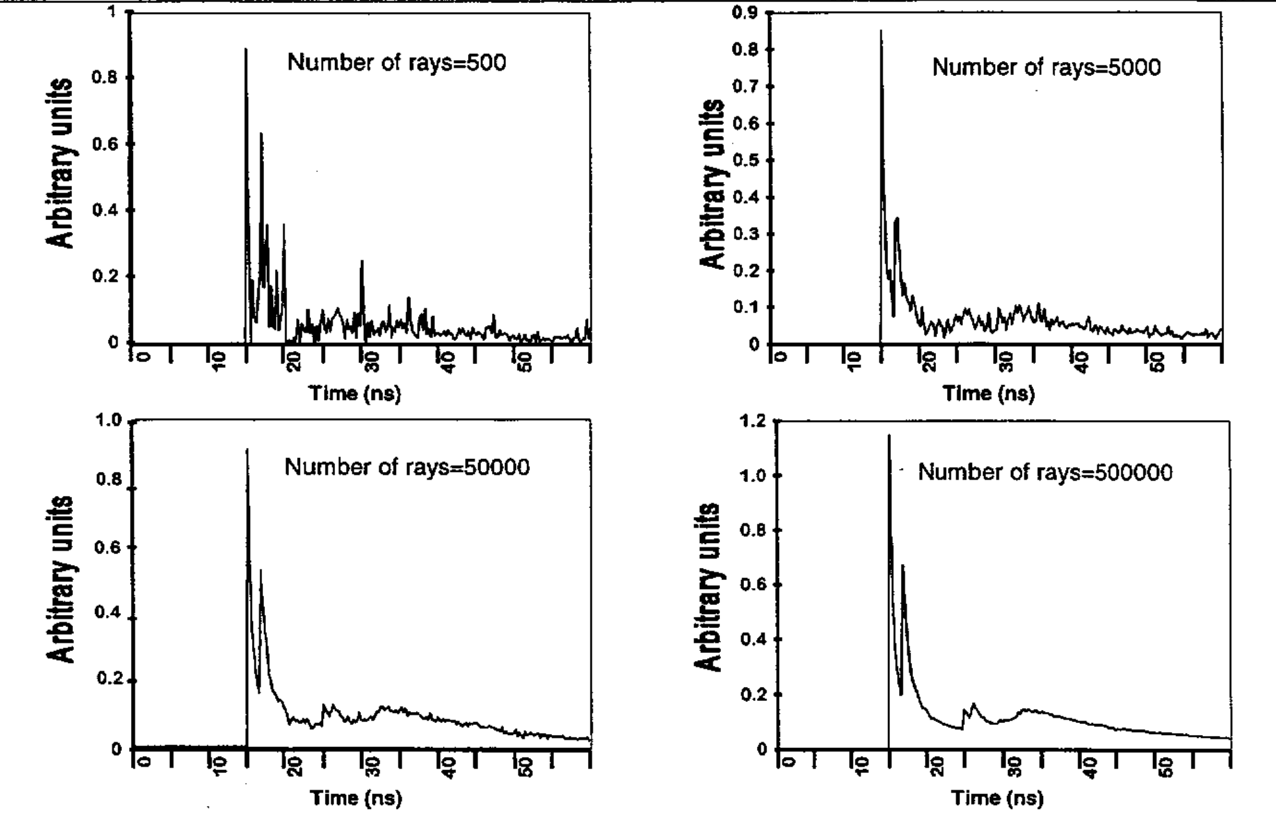

Figure 5. Variance reduction with the number of rays

In figure 5 it is presented the evolution of the impulse response for different number of rays. It can be seen the variance reduction as the number of rays increases. The vertical axis units are arbitrary, the x axis is in nanoseconds, and the graph is from configuration A without line-of-sight (LOS) contribution.

Once the impulse response is found, the room (with the emitter and receiver) is completely characterized and its bandwidth can be calculated by using DFT. By using this procedure, the effect produced by the room on any signal waveform can be obtained, because the room makes the role of a linear and time invariant low pass filter.

4. CONCLUSSIONS

A very fast method for simulating multipath response is presented. Its implementation by using compiled languages will improve its performance, which allows the realization of comparative studies on the effects of surface characteristics optimization of emitter and receiver orientations, etc.

Another important feature is the study of the effect of multipath dispersion on intersymbol interference (ISI) and the penalization on BER in different modulation schemes: PPM, SS-DS or SS-FH.

The use of ray-tracing techniques to solve the ray-obstacle equation system will further improve the calculation time, in complex environments with furniture, curved surfaces, people, textures, etc. Also, the system can take advantage of the capabilities of new 3D cheap accelerating graphics cards, which includes specific hardware to solve ray-tracing and the illumination equations.

The fact of knowing the contribution from any scatterer point on the receiver has allowed us to dramatically improve the speed of the simulation.

5. ACKNOWLEDGEMENTS

This work has been supported by the Spanish CICYT (TIC96-1467-C03-01/03) .

6. REFERENCES

1. J. R. Barry, *Wireless infrared communications*, Chapter 4, Kluwer academic publishers, Boston, 1994.
2. F.J. Lopez-Hernandez, M.J. Betancor, "DUSTIN: a Novel Algorithm for the Calculation of the Impulse Response on IR Wireless Indoor Channels". Electronics Letters, **33**. n° 21, pp. 1804-1805, 1997.
3. A. Santamaria, F.J. Lopez-Hernandez, *Wireless LAN Systems*. Chapter 1, Artech House. Boston. 1994.
4. R. Perez-Jimenez, J. Berges, M.J. Betancor. "Statistical model for the impulse response on infrared indoor diffuse channels", Electronics Letters, **33**. n° 15, pp. 1298-1300, 1997.
5. F.J. Lopez-Hernandez, R. Perez-Jimenez, A. Santamaria, "Monte Carlo calculation of impulse response on diffuse IR wireless indoor channels", Electronics Letters. **34**. n° 12, pp. 1260-1262, 1998.
6. F.J. Lopez-Hernandez, R. Perez-Jimenez, A. Santamaria, "Modified Monte Carlo scheme for high-efficiency simulation of the impulse response on diffuse IR wireless indoor channels". Electronics Letters, **34**. n° 19, pp. 1819-1820, 1998.
7. O.N. Stavroudis. *The Optics of rays, wavefronts, and caustics*, chapter 6, Academic Press, New York, 1972.

3. RESULTS

The algorithm described in section 2 was implemented in a Pentium II PC, using Microsoft Visual Basic. The choice of an interpreted language, instead of FORTRAN or C, is due to the easiness of changing data or code while the execution of the program. Usually this is at the expense of longer execution times. Nevertheless, the algorithm proposed is faster enough to get one simulation in several minutes, which is twenty times shorter than previous methods (implemented in FORTRAN or C).

Two room configurations are presented as examples: configurations A and B. The data used in the examples are in table 2. Basically the scenarios are empty rooms where multiple reflections occur on the walls, ceiling and floor. Only SR are considered, this is to compare the simulations with previous algorithms unable to simulate MR. These examples are the same as those published using other methods[1,2], so they can be compared with them.

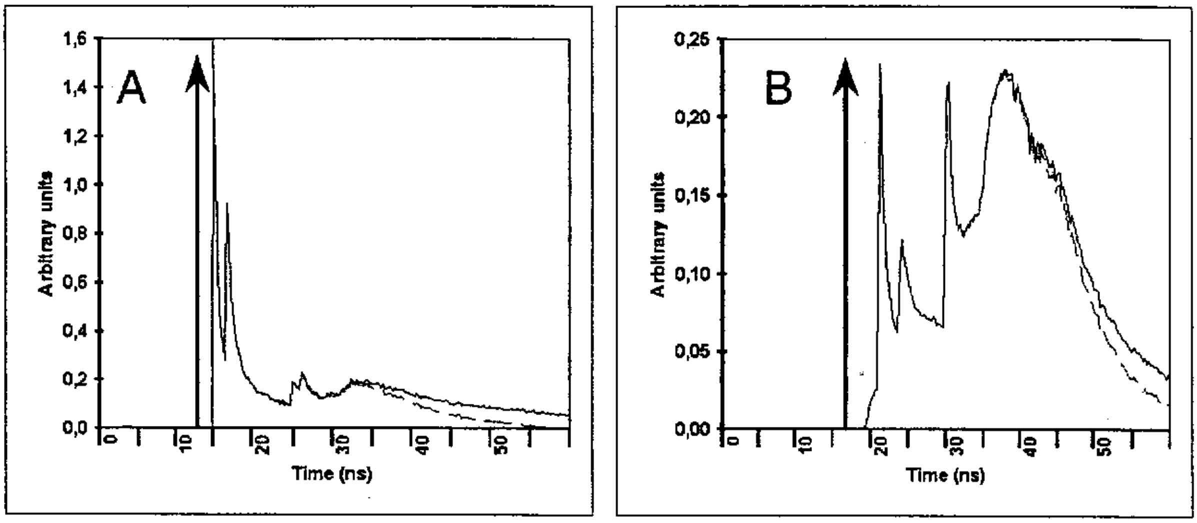

Figure 3. Configurations A and B. Continuous: up to 40 rebounds. Dotted: up to 3 rebounds

Figure 3 presents the impulse responses obtained. It is noticeable that the tails of the impulses are larger than that obtained in [1]. This is because in [1] the number of reflection was limited to 3 and our method is able to calculate as many as needed. In fact, we originally reserved space to 20 reflections, but once a ray reached this limit and the program couldn't cope with it, so to test this, we keep the record of the contribution, not only by the delay, but also by the reflection number.

Parameter			A	B
Room	Length (x)		5m	7.5m
	Width (y)		5m	5.5m
	Height (z)		3m	3.5m
	SR Coefficient	ρ_1	0.8	0.58
		ρ_2	0.8	0.56
		ρ_3	0.8	0.3
		ρ_4	0.8	0.12
		ρ_5	0.8	0.69
		ρ_6	0.3	0.09

Parameter			A	B
Emitter	Beam width		1	1
	Position (r_e)	x	2.5m	5.0m
		y	2.5m	1.0m
		z	3m	3.3m
	Orientation (n_e)	φ	0°	10°
		θ	-90°	-70°

Parameter			A	B
Receiver	Area		1cm²	1cm²
	Position (r_e)	x	0.5m	2.0m
		y	1.0m	4.0m
		z	0m	0.8m
	Orientation (n_e)	φ	0°	0°
		θ	90°	90°
	FOV		85°	70°

Table 2. Data used in simulations

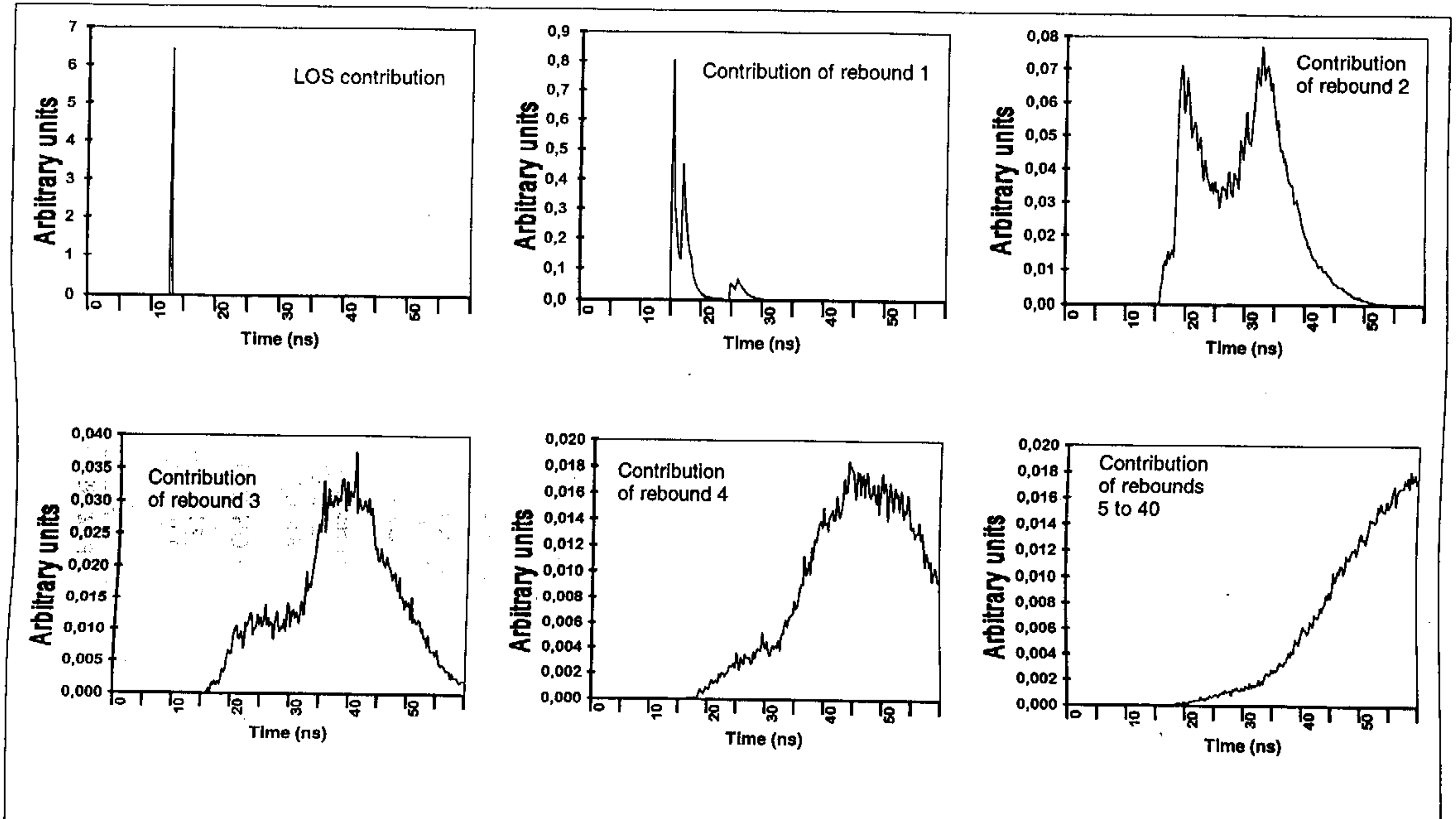

Figure 4. Relative contribution based on the number of rebounds