IEEE International Workshop on Acoustic Echo and Noise Control, Darmstadt, Germany, Sep. 2001

CHARACTERIZATION OF ACOUSTICAL ENVIRONMENTS BY NUMERICAL SIMULATION

S. Spors and R. Rabenstein

Telecommunications Laboratory
University of Erlangen-Nuremberg
Cauerstrasse 7, 91058 Erlangen, Germany
E-mail: {spors, rabe}@LNT.de

1. INTRODUCTION

The state of the art to solve problems of active noise control and acoustic echo cancellation is to use adaptive control systems [1]. For acoustic echo cancellation, adaptive algorithms have to provide a correct estimate of the room impulse response of loudspeaker-enclosure-microphone systems during operation. Test and evaluation of such algorithms requires either a real-time implementation or measured appropriate impulse responses. However, these impulse responses are only valid for one special room setup and cannot be applied to other situations. Simulations of acoustical environments can help to overcome this problem. Impulse responses computed from realistic room models can replace measured ones for test purposes.
The different methods used for computational modeling of room acoustics can be divided into three groups [2]: *Statistical models*, *Ray-based models* and *Wave-based models*. The propagation of sound waves in the air is governed by the wave equation. Unfortunately the wave equation can only be solved analytically for special cases like free field conditions or three-dimensional enclosures with very simple geometries. Therefore the solution must be approximated using more simple models for the sound propagation. *Statistical models* try to model the statistical properties of the sound intensity and are therefore not useful in our context. *Ray-based models* suppose that the sound behaves like optical rays. As a result the effects caused by the wave nature of sound, like diffraction, cannot be handled by these methods. *Wave-based* methods try to find numerical simulations for the wave equation. Because they use distributed parameter models, they are able to handle all relevant physical effects, namely wave propagation, reflection, transmission and diffraction.
To meet the high simulation quality requirements of the proposed application, *wave-based* methods are the only ones suitable here. Among various other methods developed in the last two decades, we present here a direct method to computational acoustics, which leads from the partial differential equations to a state space description of the simulation algorithm.

2. SIMULATION ALGORITHM

The propagation of sound waves in air is governed by the equation of motion and the equation of continuity for the acoustic pressure $p(\mathbf{x}, t)$ and the acoustic fluid velocity vector $\mathbf{v}(\mathbf{x}, t)$ [2],

$$\rho_0 \frac{\partial}{\partial t}\mathbf{v}(\mathbf{x}, t) + \text{grad}\, p(\mathbf{x}, t) = \mathbf{e}_s(\mathbf{x}, t) \qquad (1a)$$

$$\frac{1}{\rho_0 c^2}\frac{\partial}{\partial t} p(\mathbf{x}, t) + \text{div}\, \mathbf{v}(\mathbf{x}, t) = j_s(\mathbf{x}, t) \qquad (1b)$$

where t denotes time and $\mathbf{x}$ the vector of space coordinates x, y, z. ρ_0 is the static density of the air and c is the speed of the sound. $\mathbf{e}_s$ and j_s are appropriate source terms. These two physical principles form a set of two partial differential equations (PDEs) describing the propagation of sound waves. For our purposes a symmetric form of these equations is advantageous. This is achieved by introduction of the normalization constant $r_0 = \sqrt{3}\,\rho_0 c$ and combining (1) into one matrix equation

$$\underbrace{\begin{bmatrix} \rho_0 D_t & 0 & 0 & r_0 D_x \\ 0 & \rho_0 D_t & 0 & r_0 D_y \\ 0 & 0 & \rho_0 D_t & r_0 D_z \\ r_0 D_x & r_0 D_y & r_0 D_z & 3\rho_0 D_t \end{bmatrix}}_{\mathbf{Z}} \underbrace{\begin{bmatrix} i_1 \\ i_2 \\ i_3 \\ i_4 \end{bmatrix}}_{\mathbf{i}(\mathbf{x},t)} = \underbrace{\begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{bmatrix}}_{\mathbf{e}(\mathbf{x},t)} \qquad (2)$$

where the operators D_t, D_x, D_y, D_z denote partial derivation with respect to time and to the components x, y, z of $\mathbf{x}$. The components of $\mathbf{v}$ are denoted by i_κ, $\kappa = 1 \ldots 3$ and $i_4 = p/r_0$. Similarly, the components of $\mathbf{e}_s$ are denoted by e_κ, $\kappa = 1 \ldots 3$ and $e_4 = r_0 j_s$. This vector PDE is the starting point for the derivation of our simulation algorithm. It is essentially based on the multidimensional wave digital principle. However, a more direct access is given in [3], based on a

four-dimensional discrete-time and discrete-space state space description.
The derivation of this discrete system according to the state space approach starts from the normalized vector PDE (2). After a series of intermediate steps, the state space representation of a discrete-time and discrete-space algorithm is obtained. The following subsections cover these steps in detail.

2.1. Separation into spatial components

Inspection of (2) shows, that this PDE for three spatial components can be broken down into three PDEs with only one spatial component each. For example, the PDE for the x-component has the form

$$\bar{\mathbf{Z}}_1 \ \bar{\mathbf{i}}_1 = \bar{\mathbf{u}}_1 \ . \tag{3}$$

The matrix $\bar{\mathbf{Z}}_1$ is obtained from $\mathbf{Z}$ in (2) by elimination of the second and the third row and column, which contain only y and z components. Similarly, the first element in $\bar{\mathbf{u}}_1$ is equal to the first element in $\mathbf{e}$.

2.2. Numerical solution for each component

The numerical solution of the separate spatial components follows through a series of steps, which are shown in Fig. 1. The procedure is explained for the spatial direction x. The starting point is the partial differential operator $\bar{\mathbf{Z}}_1$ from (3). Since it involves both time and space differentiation, it is not very suitable for direct numerical integration. A decoupled form with simple differentiation operators would be more desirable. This is achieved by two measures: First, a variable transformation, which decouples the differentiation operators, or in other words, a diagonalization of the operator matrix $\bar{\mathbf{Z}}_1$. Second, a coordinate transformation, such that each entry in the diagonal operator matrix contains only differentiation with respect to a single coordinate. The resulting decoupled form allows a numerical integration by the trapezoidal rule. Care has to be taken to avoid delay free loops, which would call for an iterative solution. This problem is circumvented by another transformation of the variables. It leads to the so called wave quantities $\bar{\mathbf{a}}_1$ and $\bar{\mathbf{b}}_1$ [4]. Now, the spatial components can be integrated numerically, though in transformed coordinates and variables. Therefore, the decoupling steps have to be reversed, to arrive at a discrete-time, discrete space wave quantity formulation of the original problem. A more detailed description of the coordinate and variable transformations shown in Fig. 1 is given in [3]. The matrix $\bar{\mathbf{D}}_1$ in Fig. 1 contains shift operators in x-direction and delay operators in time direction. As a difference operator matrix it corresponds to the differential operator matrix $\bar{\mathbf{Z}}_1$.

2.3. State space formulation

The final form of the algorithm can be formulated in terms of a state space description. The state $\mathbf{z}$ is associated with one of the wave quantities (see [3] for details). The state space model consists of the state equation and the output equation

$$\mathbf{z} \ = \ \mathcal{D}\left[\mathcal{A}\mathbf{z} + \mathcal{B}\mathbf{e}\right], \tag{4a}$$
$$\mathbf{i} \ = \ \mathcal{C}\mathbf{z} + \mathcal{F}\mathbf{e}, \tag{4b}$$

The state equation (4a) follows from condensing the procedure outlined in Fig. 1 into one matrix equation. The output equation (4b) represents the conversion of the wave quantities back to acoustic variables pressure and velocity. The concise matrix formulation of the discrete model as a state space description allows the direct implementation of (4) in a software algorithm.

2.4. Boundary conditions

The operator matrix $\mathcal{D}$ in the state equation (4a) contain shifts in both directions of each spatial dimension. This requires the knowledge of the previous states in all adjacent points. However, if a point lies at the boundary of the spatial domain, e.g. at the wall of an enclosure, then one or more of the adjacent points are beyond the boundary, where the PDE is no more valid. In this case, the state of these points has to be determined from boundary conditions rather than from the PDEs (1). A detailed presentation of the incorporation of various types of boundary conditions is beyond the scope of this paper. Only a short outline of the general approach is given here.

The idea is to split the state vector $\mathbf{z}$ into two components: the interior states $\mathbf{z}_i$ and the boundary states $\mathbf{z}_b$. The interior states follow from a state equation similar to (4a). The boundary states follow from the interior states and the boundary conditions. The state space representation has to consider both types of states appropriately. Its general form is given by

$$\mathbf{z}_i \ = \ \left(\mathbf{T}_i^T\mathcal{D}\right)\left[\mathcal{A}\mathbf{z} + \mathcal{B}\mathbf{e}\right], \tag{5a}$$
$$\mathbf{z}_b \ = \ \mathcal{A}_b\mathbf{z}_i + \mathcal{B}_b\mathbf{e}, \tag{5b}$$
$$\mathbf{z} \ = \ \mathbf{T}_i\mathbf{z}_i + \mathbf{T}_b\mathbf{z}_b, \tag{5c}$$
$$\mathbf{i} \ = \ \mathcal{C}\mathbf{z} + \mathcal{F}\mathbf{e}. \tag{5d}$$

The matrices $\mathbf{T}_i$ and $\mathbf{T}_b$ contain only ones and zeros. They depend on the geometry and describe whether a state is an interior state $\mathbf{z}_i$ or a boundary state $\mathbf{z}_b$. Equation (5a) is very similar to the state equation (4a), except that it delivers only the interior states. The boundary states are computed in (5b) from the interior states and the boundary conditions, which determine $\mathcal{A}_b$ and $\mathcal{B}_b$. Both interior and boundary states are merged into the complete state vector $\mathbf{z}$ in (5c). It is used to deliver the output quantities in (5d) and to update the interior states in (5a).

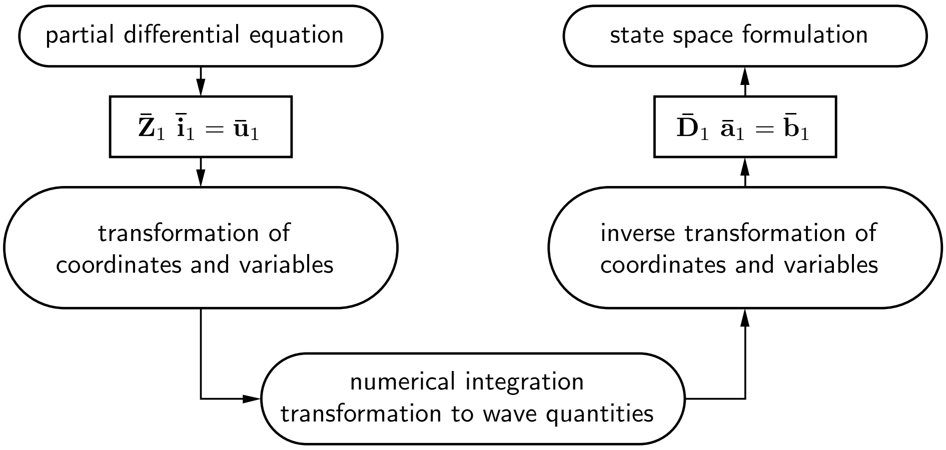


Figure 1: Transformation of the partial differential operator for one spatial component into a state space formulation

3. IMPLEMENTATION

The algorithm described above has been implemented in `C++` in an object oriented fashion. This implementation is based on a multidimensional systems library which was developed at our laboratory. In the current version, objects with rectangular shape and analytical objects of 2nd degree (e.g. ellipsoids) with given surface reflexion factor can be modeled. Available sources include point sources, loudspeaker arrays and horn loudspeakers. The respective sound pressure of the wavefield can be captured at any point within the spatial grid.

4. RESULTS

To show the performance of the algorithm described above, we conducted some experiments with two different room setups. The first room has the size $2.0 \times 2.0 \times 1.8$ m (w × d × h). The ceiling and floor have a surface reflection factor of $r = 0$ and are therefore absorbing, the remaining four walls have a surface reflection factor of $r = 0.7$. The source is placed in the middle of a reflecting wall at 20 cm distance, the microphone at the middle of the opposite wall in 40 cm distance. Figure 2 shows the impulse response of the empty room. The direct sound and the first reflections can be seen clearly. The second room setup consists of the same boundaries as the first one, but has some walls built within. Again the surface of the inserted walls have an reflection factor of $r = 0.7$. Figure 3 shows snapshots of the wavefield for the second room setup at $t = 4.1$ ms and $t = 6.7$ ms after excitation with an Gaussian impulse at $t = 0$ at the shown source position Q. Figure 4 shows the impulse responses of two different microphone positions M_1 and M_2 as shown in Figure 3. The upper plot shows the impulse response recorded at the microphone M_1, the lower plot the one recorded at the microphone M_2. The difference in terms of a higher reverberation is clearly visible in the upper one.

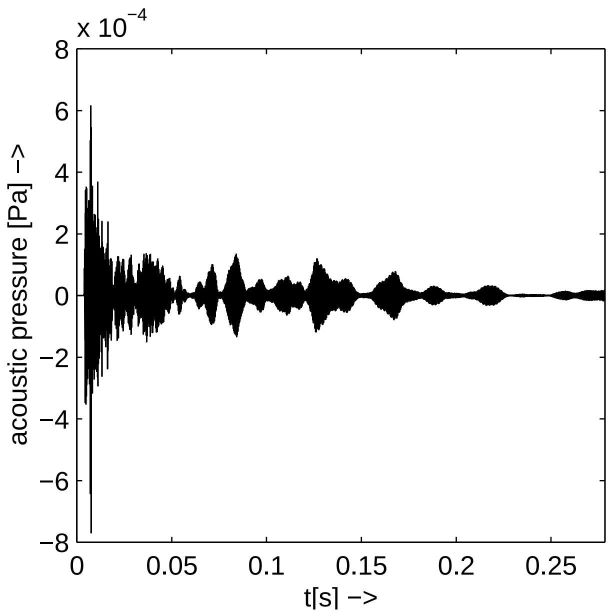
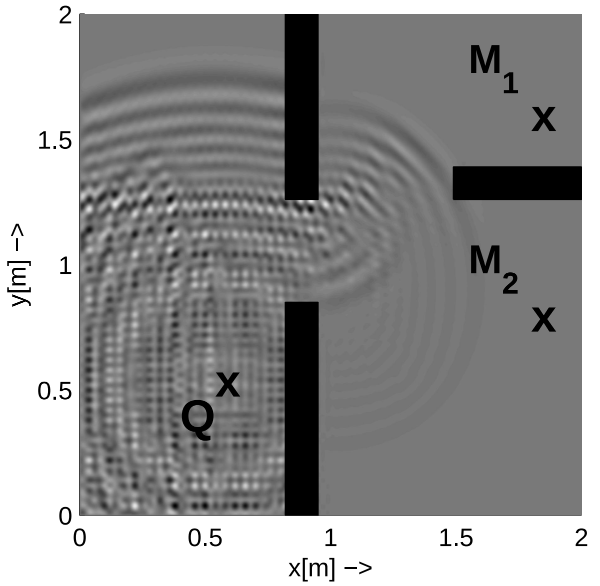


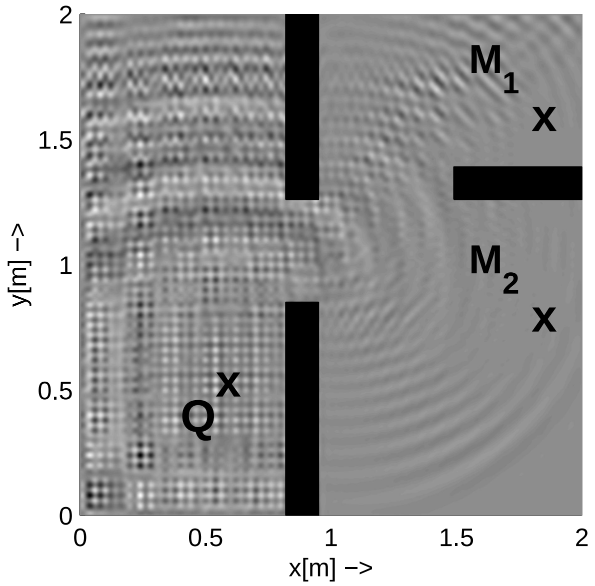
Figure 2: Impulse response of an empty rectangular room

5. CONCLUSION

The simulations presented here show, that a numerical calculation of room impulse responses is an alternative to room response measurements. It is the only possibility if test data is required for virtual acoustical environments. Due to its solid physical foundation, the presented method handles all relevant acoustical effects (propagation, reflection, diffraction) and does



(a) $t = 4.1$ ms



(b) $t = 6.7$ ms

Figure 3: Snapshots of the wavefield after excitation with an Gaussian impulse at the source position Q. The microphone positions M_1 and M_2 where used to record the impulse responses shown in Figure 4

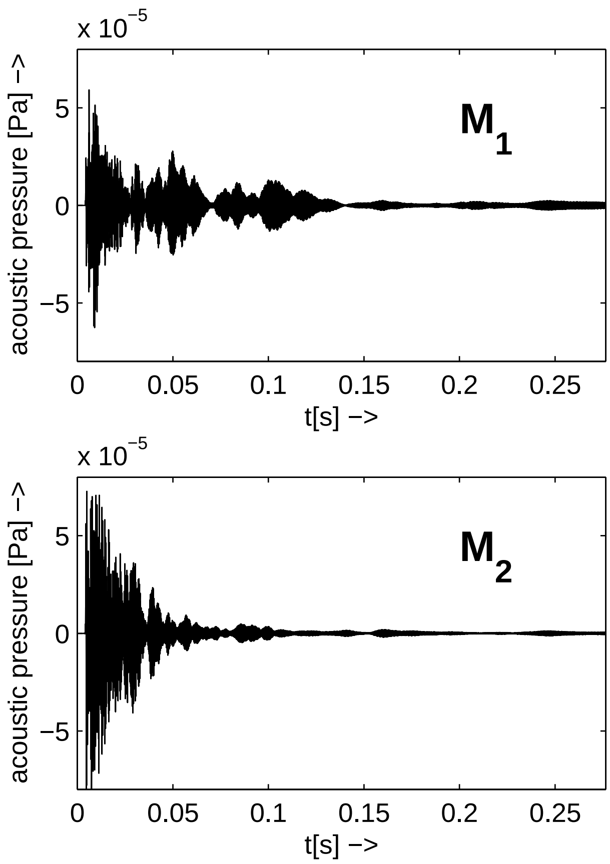


Figure 4: Impulse responses recorded at the microphone positions M_1 and M_2 as shown in Figure 3

not require simplifying assumptions. Applications for the derived algorithm include test data generation for echo cancellation, noise control and source separation algorithms. Another application field includes virtual acoustics. The algorithm can be applied to predict room acoustics and calculate characteristic parameters like the sound decay time T_{60} and MPEG4 perceptual parameters [5] based on an geometrical description of the scene.

6. REFERENCES

[1] C. Breining et al., "Acoustic echo control: An application of very-high-order adaptive filters," *IEEE Signal Processing Magazine*, vol. 16, no. 4, pp. 42–69, 1999.

[2] K.H. Kuttruff, "Sound in enclosures," in *Handbook of Acoustics*, M.J. Crocker, Ed. John Wiley and Sons, Inc., 1998.

[3] R. Rabenstein and A. Zayati, "Sound field simulation by computational acoustics. Part I: Simulation algorithm," *Int. Journal of Adaptive Control and Signal Processing*, vol. 14, pp. 663–680, 2000.

[4] A. Fettweis, "Multidimensional wave-digital principles: From filtering to numerical integration," in *Proc. Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP 94)*. IEEE, April 1994, pp. VI–173 – VI–181.

[5] ISO/IEC 14496-1, MPEG-4 Standard, "Information technology - coding of audiovisual objects," 2000, Part1 (Systems).