IMPLEMENTATION OF ARBITRARY LINEAR SOUND SYNTHESIS ALGORITHMS BY DIGITAL WAVE GUIDE STRUCTURES

Stefan Petrausch and Rudolf Rabenstein

Multimedia Communications and Signal Processing
University of Erlangen-Nuremberg, Cauerstraße 7, 91058 Erlangen, Germany
{stepe,rabe}@LNT.de

ABSTRACT

The Digital Wave Guide (DWG) method is one of the most popular techniques for digital sound synthesis via physical modeling. Due to the inherent solution of the wave equation by the structure of the DWG method, it provides a highly efficient algorithm for typical physical modeling problems. In this paper it will be shown, that it is possible to use this efficient structure for any existing linear sound synthesis algorithm. By a consequent description of discrete implementations with State Space Structures (SSSs), suitable linear state space transformations can be used to achieve the typical DWG structure from any given system. The proposed approach is demonstrated with two case studies, where a modal solution achieved with the Functional Transformation Method (FTM) is transformed to a DWG implementation. In the first example the solution of the lossless wave equation is transformed to a DWG structure, yielding an arbitrary size fractional delay filter. In another example a more elaborated model with dispersion and damping terms is transformed, resulting in a DWG model with parameter morphing features.

1. INTRODUCTION

Physical Modeling (PM) techniques became an established method for digital sound synthesis in the past decade. They model the sound production mechanisms rather than the sound itself and achieve not only realistic sounds, but also a realistic interaction with the sound. Consequently it is used by most recent synthesizers at least in hybrid forms.

The common starting point for most PM techniques is a set of Partial Differential Equations (PDEs), as given by the laws of physics. However the approach for realization differs a lot between the various PM methods. Currently there are a number of these realizations, Finite Difference Time Domain (FDTD) schemes [1], transfer function techniques like the Functional Transformation Method (FTM) [2, 3], and the Digital Wave Guide (DWG) method [4] to mention just a few ones.

The most popular of the above mentioned techniques is the DWG method. It is based on the analytic d'Alembert solution of the wave equation (see [4]) by forward and backward traveling waves. These waves can be realized by simple delay lines in the discrete-time system, yielding highly efficient algorithms with a typical structure (see [5] for instance). As most physical models for sound synthesis are somehow based on the wave equation, this is the reason for the popularity of the DWG method.

On the other hand all linear systems can be represented by so called State Space Structures (SSSs) [6, 7], and SSSs can be transformed from one form into another form [8, 9]. Therefore in this paper a method for the implementation of arbitrary linear sound synthesis algorithms by DWG structures is presented. It is shown how to transform arbitrary SSSs into typical DWG implementations by usage of linear state-space transformations (also called similarity transformation). In particular three examples are demonstrated, where a FTM implementation is transformed into a DWG structure and vice versa. In doing so, the first example in section 4.1 demonstrates the use of fractional delay filters in DWG implementations. The second example in section 4.2 shows the inverse transformation of the same problem. The third example in section 4.3 yields a DWG implementation of a dispersive string with damping terms, where the physical meaning of the parameters is preserved in the DWG implementation.

The paper is structured as follows. As a first step, in section 2, the typical structure of DWG implementations and the corresponding SSS is introduced. Then in section 3, the required state-space transformations for the transformation into this DWG SSS are given. The example scenarios are demonstrated in section 4, conclusions and an outlook for the usage of the proposed method are discussed in section 5.

2. STRUCTURE OF DIGITAL WAVE GUIDE IMPLEMENTATIONS

First of all, a closer look has to be taken on the principal structure of DWG implementations. Both as a block diagram, as normally given in literature [5], and important in this scope, as a SSS.

2.1. Block Diagram

The typical structure of DWG implementations (see e.g. [5]) is depicted in figure 1 in a simplified scheme. All signals are time discrete values, denoted by the square brackets and the discrete time step k. The input function $v[k]$ is assumed to be suitable pre-filtered, i.e. plucking position and type are included in $v[k]$.

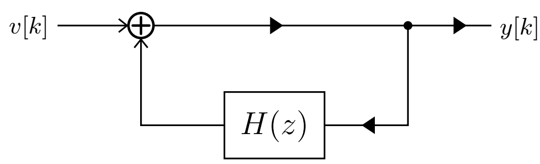

Figure 1: Simplified structure of the DWG-method. The different physical effects are modeled by the filter $H(z)$, which includes at least one sample delay. The input signal $v[k]$ is suitable scaled and filtered.

The delay of the feedback filter $H(z)$ (z is the time discrete frequency variable) determines the pitch of the tone. Although figure 1 is a simplified scheme, it constitutes a quite general model, as there are no restrictions for the feedback filter. In practice $H(z)$ includes a delay line, fractional delay filters, and a model specific allpass filter. Here we describe it in the frequency domain as a division of nominator and denominator polynomial in z (see [6, 7])

$$H(z) = \frac{Z(z)}{N(z)} = \frac{b_{n-1}z^{n-1} + \ldots + b_1 z + b_0}{z^n + a_{n-1}z^{n-1} + \ldots + a_1 z + a_0} \ . \quad (1)$$

2.2. Corresponding State Space Structure

Using this filter description (1) we can derive the transfer function $G(z)$ of the complete system from figure 1 by

$$\begin{aligned} Y(z) &= V(z) + H(z)Y(z) = \frac{1}{1-H(z)}V(z) \\ G(z) &= \frac{Y(z)}{V(z)} = \frac{N(z)}{N(z)-Z(z)} \\ &= \frac{z^n + a_{n-1}z^{n-1} + \ldots + a_1 z + a_0}{z^n + c_{n-1}z^{n-1} + \ldots + c_1 z + c_0} \ , \end{aligned} \quad (2)$$

with the coefficients $c_i := a_i - b_i \ , \ i \in \{0; 1; \ldots ; n-1\}$.

The corresponding SSS is well known in literature (see [9] for instance). For a state space description in the form (bold face notation denotes vectors resp. matrices)

$$\begin{aligned} \mathbf{z}[k+1] &= \mathbf{A}\mathbf{z}[k] + \mathbf{B}v[k] \\ y[k] &= \mathbf{C}\mathbf{z}[k] + \mathbf{D}v[k] \ , \end{aligned} \quad (3)$$

a permissible system matrix $\hat{\mathbf{A}}$ that corresponds to equation (2) is the so called Frobenius or companion matrix. In detail the state space matrices are given by

$$\hat{\mathbf{A}} = \begin{pmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \vdots & & & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & 1 \\ -c_0 & -c_1 & -c_2 & \ldots & -c_{n-1} \end{pmatrix} \quad (4)$$

$$\begin{aligned} \hat{\mathbf{B}} &= \begin{pmatrix} 0 & 0 & \ldots & 0 & 1 \end{pmatrix}^T \\ \hat{\mathbf{C}} &= \begin{pmatrix} a_0 - c_0 & a_1 - c_1 & \ldots & a_{n-1} - c_{n-1} \end{pmatrix} \\ \hat{\mathbf{D}} &= 1 \ . \end{aligned} \quad (5)$$

3. STRUCTURE TRANSFORMATION

With the state space description of the DWG implementation in equation (3) to (5) we can maintain, that the only difference between a DWG implementation and an arbitrary linear and discrete system is in the appearance of the system matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{D}$. The only remaining task to transform the different system descriptions into each other is the transformation of the system matrices.

3.1. Linear state space transformation

This transformation can be performed by the the so called linear state-space transformation, which is well known from systems theory (see [8] for instance). It is based on the linear transformation of the state space vector $\mathbf{z}[k]$ by the non singular transformation matrix $\mathbf{T}$

$$\hat{\mathbf{z}}[k] = \mathbf{T}\mathbf{z}[k] \ . \quad (6)$$

The transformed state-space description is in the same form as the original one (see equation (3))

$$\begin{aligned} \hat{\mathbf{z}}[k+1] &= \hat{\mathbf{A}}\hat{\mathbf{z}}[k] + \hat{\mathbf{B}}v[k] \\ y[k] &= \hat{\mathbf{C}}\hat{\mathbf{z}}[k] + \hat{\mathbf{D}}v[k] \ , \end{aligned} \quad (7)$$

except for the transformed system matrices, which can be obtained by

$$\begin{aligned} \hat{\mathbf{A}} &= \mathbf{T}\mathbf{A}\mathbf{T}^{-1} & \hat{\mathbf{B}} &= \mathbf{T}\mathbf{B} \\ \hat{\mathbf{C}} &= \mathbf{C}\mathbf{T}^{-1} & \hat{\mathbf{D}} &= \mathbf{D} \ . \end{aligned} \quad (8)$$

3.2. Controllability canonical forms

Normally it is not trivial to find suitable transformation matrices. However, fortunately the SSS given by the system matrices (4) to (5) is one of the so called controllability canonical forms. It is proven, that any completely controllable system can be transformed into the controllability canonical form with a system matrix $\mathbf{A}$ in the form of (4) (see [8]). Furthermore, algorithms for this transformation are available.

The first step of the transformation from an arbitrary SSS (3) towards a SSS (7) with the system matrices in form of equation (4) to (5) is the computation of the controllability matrix $\mathbf{K}_c$ by (see [8])

$$\mathbf{K}_c = \left(\mathbf{B}, \mathbf{A}\mathbf{B}, \mathbf{A}^2\mathbf{B}, \ldots , \mathbf{A}^{n-1}\mathbf{B}\right) \ .$$

Then, with $\mathbf{t}_n^T$ denoting the last row of $\mathbf{K}_c^{-1}$, we select the transformation matrix by

$$\mathbf{T} = \begin{pmatrix} \mathbf{t}_n^T \\ \mathbf{t}_n^T\mathbf{A} \\ \vdots \\ \mathbf{t}_n^T\mathbf{A}^{n-1} \end{pmatrix} \ . \quad (9)$$

The application of the matrix transformations from equation (8) leads to the desired form in equation (4) to (5) (see [8] chapter 7.2). The coefficients of the complete transfer function (2) can be directly obtained from the controllability canonical form which leads to the coefficients of the DWG feedback-filter $H(z)$ (see equation (1)) with ease.

3.3. Inverse transformation

The inverse procedure from a DWG system description to other structures is presented in this section. From systems theory it is known, that any system with single eigenvalues λ_i can be transformed to the diagonal form (synonym for the parallel system description) with the matrix of its eigenvectors $\mathbf{v}_i$. The eigenvalues together with their corresponding eigenvectors solve the so called eigenvalue problem

$$\left(\mathbf{A} - \lambda_i \mathbf{I}\right) \mathbf{v}_i = \mathbf{0} \ . \quad (10)$$

For distinct eigenvalues (which is a valid assumption for sound synthesis models), one always achieves n eigenvalues and n linearly independent eigenvectors $\mathbf{v}_i$ for a $n \times n$ system matrix $\mathbf{A}$. These eigenvectors form the transformation matrix $\mathbf{T}$, by

$$\mathbf{T} = (\mathbf{v}_1 | \mathbf{v}_2 | \dots | \mathbf{v}_n) \ . \tag{11}$$

As the system order n normally equals the number of complex harmonics of the resulting sound, one has to find the zeros of e.g. a 200th order (for a normal guitar string) polynomial to find the eigenvalues with equation (10). This is difficult, but not impossible.

However, for the so called Frobenius matrix $\hat{\mathbf{A}}$ from equation (4) fortunately an analytic solution for the transformation matrix $\mathbf{T}$ is known. It is the so called Vandermonde matrix (see [8] for instance)

$$\mathbf{T}_\mathrm{V} = \begin{pmatrix} 1 & 1 & \dots & 1 \\ \lambda_1 & \lambda_2 & \dots & \lambda_n \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_1^{n-1} & \lambda_2^{n-1} & \dots & \lambda_n^{n-1} \end{pmatrix} \ . \tag{12}$$

To summarize the procedure from a DWG implementation to another different implementation (see also [9], figure 2.15), one first has to find the eigenvalues of the DWG implementation with equation (10). Then one can use the Vandermonde matrix (12) to transform the system to the diagonal form. For the second state space transformation one has to find the eigenvectors of the desired system with (10). Application of a state space transformation with the inverse of the transition matrix in (11) yields the desired system description.

4. EXAMPLE SCENARIOS

For the example scenarios, the resulting structure of the Functional Transformation Method (FTM) is used. The frequency approach of the FTM causes an orthogonalization of the problem and in consequence a parallel system description (i.e. a diagonal state matrix $\mathbf{A}$, see [2] for instance), what it is an ideal candidate for the conversion to the DWG-structure.

4.1. FTM to DWG

The first two example scenarios are more illustrative. As a physical model for the string deflection $y(x,t)$ the wave equation is used as given in equation (13), where c is the speed of sound and $f_\mathrm{e}(x,t)$ is an arbitrary excitation distribution

$$\frac{\partial^2}{\partial x^2} y(x,t) - \frac{1}{c^2} \frac{\partial^2}{\partial t^2} y(x,t) = f_\mathrm{e}(x,t) \ . \tag{13}$$

A detailed description of the solution with the FTM is given in [2] or [3] and is not performed here. It is just to mention, that the FTM yields a diagonal system matrix $\mathbf{A}$, whose eigenvalues are the diagonal elements themselves. Furthermore section 3.2 has to be considered, where controllability of the original system is assumed. This is achieved in the FTM system by neglecting the distribution of the excitation $f_\mathrm{e}(x,t)$ and exciting all harmonics identically.

The SSS obtained by the FTM was transformed according to equation (8) with the transition matrix (9). The coefficients a_i and c_i were extracted from the resulting transformed SSS with the Frobenius matrix $\hat{\mathbf{A}}$. By the relation $b_i = a_i - c_i$ (see section 2.2) all coefficients for the feedback filter $H(z)$ of the DWG structure are available.

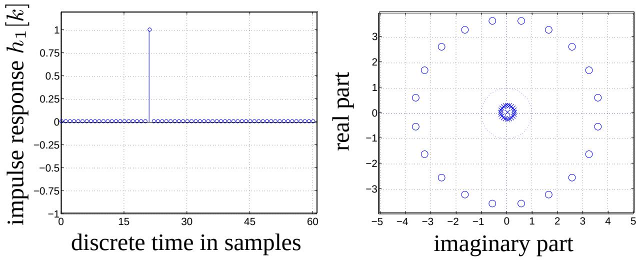

Figure 2: Impulse response $h_1[k]$ and pole zero plot of the first example DWG feedback filters $H_1(z)$. The 20 complex harmonics are uniformly distributed, yielding approximately a perfect integer delay for the feedback filter.

The resulting feedback filter for the first example parameter set is depicted in figure 2. The length of the string l_1 was adjusted to cause uniformly distributed complex harmonics as it can be seen in the pole zero plot on the right side. With T denoting the sampling interval, the length was set to $l_1 = cT \cdot \frac{2}{21}$.

Please note, that figure 2 is a plot of the feedback filter $H_1(z)$. The poles of the complete system $G_1(z)$ are all located on the unit circle, as equation (13) does not include damping effects.

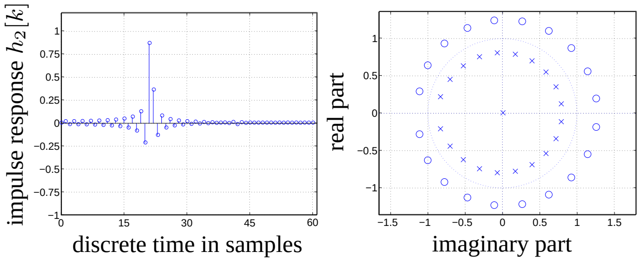

Figure 3: Impulse response $h_2[k]$ and pole zero plot of the second example DWG feedback filters $H_2(z)$. The 20 complex harmonics are non-uniformly distributed, yielding a fractional delay for the feedback filter.

The second example parameter set yields a feedback filter as shown in figure 3. The length was set to $l_2 = cT \cdot \frac{2}{21+\frac{1}{3}}$. This results in the typical fractional delay filter as it is depicted by the impulse response $h_2[k]$ in figure 3.

As it can be seen in figure 2 and 3, we can differentiate between two extreme cases what concerns computational efficiency. The first example is somehow ideal in terms of computational efficiency. The feedback filter of the DWG $H_1(z)$ is simply an integer delay. The second example is only slightly higher in pitch, but it demands more computational power.

To overcome this problem of highly parameter dependent computational efficiency, DWG implementations approximate the fractional delay filter. It is assumed, that the feedback filter $H(z)$ is a serial connection of an integer delay line, and a fractional delay line of fixed length. The integer delay line determines the approximate pitch of the tone, and the fractional delay line is used for the fine tuning of the pitch.

Exactly the same can be done with the proposed approach. The feedback filter $H(z)$ is split into one larger part $H_1(z)$ with

$l_1 = cT \cdot \frac{2}{n+1}$, and a smaller part $H_2(z)$ with length $l_2 = l - l_1$. n is chosen such that $H_2(z)$ has a fixed order. In result $H_1(z)$ is a simple integer delay filter and $H_2(z)$ is a fixed size fractional delay filter. An equivalent implementation is achieved by pure DWG modeling.

4.2. DWG to parallel form

Just for completeness, the inverse direction from a simple DWG implementation to a parallel form is demonstrated in this section. To simulate the lossless wave equation the simple structure for a fractional delay line according to figure 4 is chosen.

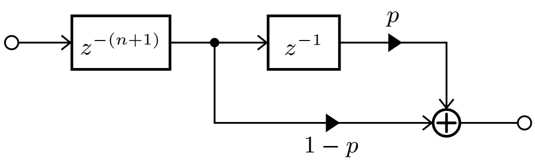

Figure 4: Feedback filter $H_3(z)$ for the third example. It is a simple fractional delay filter by linear interpolation, the parameter p determines the fractional delay.

The feedback filter $H_3(z)$ yields only a few non zero coefficients for the Frobenius matrix $\hat{\mathbf{A}}$ in equation (4), what can be easily reproduced. To achieve the parallel form the procedure from section 3.3 is applied. However, as the desired SSS has diagonal form, it is sufficient to search for the eigenvalues of the Frobenius matrix, use this eigenvalues to formulate the Vandermonde matrix $\mathbf{T}_\text{V}$ (12) and transform the SSS with this Vandermonde matrix.

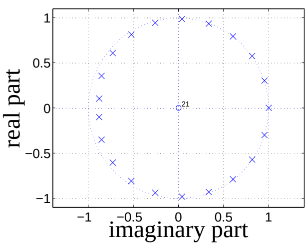

Figure 5: Pole zero plot of the systems transfer function $G_3(z)$.

The transfer function $G_3(z)$ of the resulting system is depicted in figure 5 as an pole zero plot. The poles are not located on the unit circle, as would be expected for the lossless wave equation. Their position inside the unit circle is caused by the lowpass characteristics of the simple fractional delay filter $H_3(z)$ from figure 4.

4.3. Enhanced String Model

The last example scenario demonstrates a more practical problem. The FTM provides analytic solutions also for complex models, for example string models, that includes dispersion (caused by the stiffness of the string) and damping terms. The complete model and its solution can be found in [2]. Here just the result of the transformation procedure is presented, as it can be seen in figure 6.

As both, the application of the FTM and the transformation of the implementation proposed in this paper can be done in an analytic fashion, it is possible to achieve DWG implementations of dispersive strings with a direct link from the physical parameters

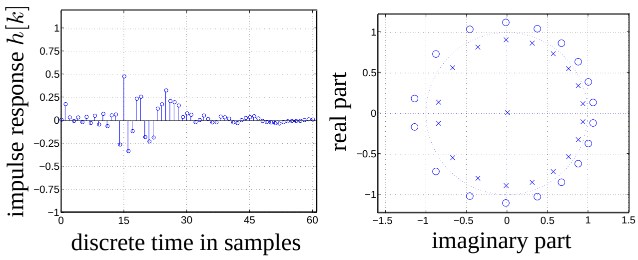

Figure 6: Impulse response $h_4[k]$ and pole zero plot of the fourth example DWG feedback filters $H_2(z)$. The underlying model includes string stiffness and damping terms.

to the parameters of the feedback filter. One can even implement physical parameter morphing.

5. CONCLUSIONS

In this paper a new approach for the implementation of sound synthesis algorithms is presented. By formulating the state space structure description, it is possible to implement any linear discrete algorithm with a DWG structure. The proposed method enables on the one hand new research and simplifies the verification of new algorithms. On the other hand, it is possible to implement sound synthesis algorithms in a consistent form, facilitating cooperation between different techniques.

6. REFERENCES

[1] S. Bilbao and J. O. Smith, "Finite difference schemes for the wave equation: Stability, passivity, and numerical dispersion," *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 3, pp. 255–266, May 2003.

[2] S. Petrausch and R. Rabenstein, "A simplified design of multidimensional transfer function models," in *International Workshop on Spectral Methods and Multirate Signal Processing (SMMSP2004)*, Vienna, Austria, Sept. 2004, pp. 35–40.

[3] Lutz Trautmann and Rudolf Rabenstein, *Digital Sound Synthesis by Physical Modeling using Functional Transformation Models*, Kluwer Academic Publishers, New York, 2003.

[4] J. O. Smith, "Physical modeling using digital waveguides," *Computer Music Journal*, vol. 16, no. 4, pp. 74–91, 1992, Special Issue on Physical Modeling of Musical Instruments, Part I.

[5] M. Karjalainen, V. Välimäki, and T. Tolonen, "Plucked-string models: From the Karplus-Strong algorithm to digital waveguides and beyond," *Computer Music J.*, vol. 22, no. 3, pp. 17–32, 1998.

[6] S. K. Mitra, *Digital Signal Processing, a Computer-Based Approach*, McGraw-Hill, New York, 1998.

[7] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing*, Prentice Hall, Englewood Cliffs, NJ, 1996.

[8] F. Szidarovszy and A. T. Bahill, *Linear Systems Theory*, CRC Press, Boca Raton, Florida, 1992.

[9] B. Girod, R. Rabenstein, and A. Stenger, *Signals and Systems*, John Wiley & Sons Ltd, Chichester, England, 2001.