TIME-DOMAIN PHYSICAL MODELING AND REAL-TIME SYNTHESIS USING MIXED MODELING PARADIGMS

Matti Karjalainen

Helsinki University of Technology
Laboratory of Acoustics and Audio Signal Processing
matti.karjalainen@hut.fi

ABSTRACT

Three approaches, often used in discrete-time modeling of physical systems, are digital waveguides (DWG) [1], finite difference time-domain schemes (FDTD) [2], and wave digital filters (WDF) [3]. We have shown it feasible to combine these paradigms to a mixed modeling framework [4, 5], making it possible to utilize the best features of each of them. Based on the theoretical framework we have developed a software tool, called the BlockCompiler [6], which is used for high-level description of model structures, automatically compiled to efficient simulation and real-time synthesis. In this paper we present an overview of the mixed modeling paradigm and the BlockCompiler based on it, as well as 1-D and 2-D physical modeling examples simulated in real time.

1. MIXED MODELING FRAMEWORK

In time-domain discrete-time modeling of physical systems the task is to approximate the underlying (partial) differential equations by difference equations then to be solved numerically. Formulation of the solution as DSP algorithms makes them computable by efficient software tools, even as real-time simulation. We first introduce DSP formulations of digital waveguides (DWG) and finite difference time-domain schemes (FDTD) as well as their combinations. Then we add wave digital filters (WDF) to this modeling framework.

1.1. Wave-based modeling with DWGs

1-D traveling wave formulation is based on the d'Alembert solution of propagation of two opposite direction waves, which properly discretized in time and space [1] yields:

$$y(n,m) = \overrightarrow{y}(n-m) + \overleftarrow{y}(n+m) \tag{1}$$

where y the wave variable and n and m are related to time and position, respectively. It follows that the wave propagation can be computed by updating state variables in two delay lines by

$$\overrightarrow{y}_{k,n+1} = \overrightarrow{y}_{k-1,n} \quad \text{and} \quad \overleftarrow{y}_{k,n+1} = \overleftarrow{y}_{k+1,n} \tag{2}$$

i.e., by simply shifting the samples to the right and left, respectively. Index k denotes now position and n is sample index (time). This kind of discrete-time modeling is called Digital Waveguide (DWG) modeling [1]. Since the physical wave variables are split explicitly into directional wave components, we will call such models *W-models* (wave models).

The next step is to take into account the global physical constraints of continuity by Kirchhoff type of rules. This means to formulate the scattering junctions of interconnected ports, with given impedances and wave variables at related ports. For a scattering junction where the physical variables are sound wave pressure P

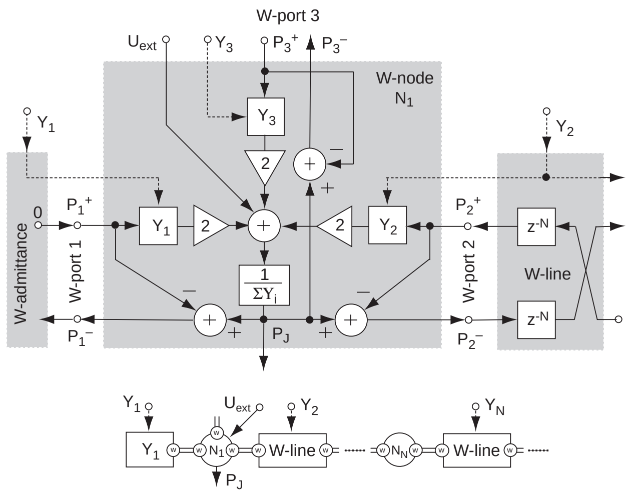

Figure 1: Top: A 3-port scattering junction (W-node N_1). Incoming pressures are P_i^+ and outgoing ones P_i^+. W-port 1 is connected to termination W-admittance Y_1 and port 2 to a two-directional delay line (W-line). Admittance controls are marked by dashed lines. Bottom: Block diagram with abstracted blocks and how they can be connected to form a 1-D DWG waveguide.

and volume velocity U, and for a parallel admittance model[1] of N ports, the Kirchhoff constrains become

$$P_1 = P_2 = \ldots = P_N = P_\mathrm{J} \tag{3}$$

$$U_1 + U_2 + \ldots + U_N + U_\mathrm{ext} = 0 \tag{4}$$

where P_J is the common pressure of coupled branches and U_ext is an external volume velocity to the junction. When port pressures are represented by incoming wave components P_i^+, outgoing wave components by P_i^-, admittances attached to each port by Y_i, and

$$P_i = P_i^+ + P_i^- \quad \text{and} \quad U_i^+ = Y_i P_i^+ \tag{5}$$

the junction pressure P_J can be obtained as:

$$P_\mathrm{J} = \frac{1}{Y_\mathrm{tot}} (U_\mathrm{ext} + 2 \sum_{i=0}^{N-1} Y_i P_i^+) \tag{6}$$

where $Y_\mathrm{tot} = \sum_{i=0}^{N-1} Y_i$ is the sum of all admittances to the junction. Outgoing pressure waves, obtained from Eq. (5), are then $P_i^- = P_\mathrm{J} - P_i^+$. The resulting junction, a *W-node*, is depicted as a DSP structure in the node N_1 of Fig. 1 (top). When admittances Y_i are frequency-dependent, this diagram can be interpreted as a

[1] Models can be formulated as well for impedances instead of admittances and for series connection, and for different physical variable pairs.

filter structure where the incoming pressures are filtered by the corresponding wave admittances Y_i times two, and their sum is filtered further by $1/Y_{\rm tot}$ to get the junction pressure $P_{\rm J}$.

Two special cases can be noticed on the basis of Eq. (6). First, a (passive) loading admittance is the case with Y_i where no incoming pressure wave component P_i^+ is associated. This needs no computation except including Y_i in $Y_{\rm tot}$ because $P_i^+ = 0$, see the left-hand termination, a *W-admittance*, in Fig. 1. Another issue is the external velocity $U_{\rm ext}$ effective to the junction. This is connected directly to the summation at the junction node.

The W-node in Fig. 1 is coupled through *W-ports* to the neighboring elements (port 3 is uncoupled). The right-hand side block is a two-directional delay line, a *W-line*, of admittance Y_2. The bottom part of the figure depicts a block diagram abstraction of the DSP structure. It also characterizes how waveguides are built as structures of W-line elements connected by W-node junctions.

Notice that the admittances in Fig. 1 may be real-valued or frequency-dependent so that Y_i and the impedance $1/\sum Y_i$ can be realized as FIR or IIR filters, or just as real coefficients if all attached admittances are real. The realization of junction nodes, as shown in Fig. 1, is general for any LTI system approximation, also for 2-D and 3-D mesh structures. The delays (see W-line in Fig. 1) between nodes can also approximate fractional delays [7], which is useful particularly with varying length lines. However, delays shorter than a unit delay lead to the problem of delay-free loops, which complicates the situation substantially.

Dashed lines in Fig. 1 are parametric controls for admittances of the network elements. If the DSP blocks are grouped as shown, which is natural in an object-based formulation, the junction W-nodes actually contain most of the computation by implementing wave scattering. The W-node is delegated its admittance parameters through ports from the network elements, W-lines and W-admittances. In a time-varying case the admittance filters (blocks Y_1, Y_2, and Y_3) as well as the inverse of their sum $1/\sum Y_i$ must be updated when the admittance control parameters change.

1.2. Finite difference modeling

In the most commonly used way to discretize the wave equation by finite differences the partial derivatives are approximated by second order finite differences

$$y_{xx} \approx (2y_{x,t} - y_{x-\Delta x,t} - y_{x+\Delta x,t})/(\Delta x)^2 \qquad (7)$$

$$y_{tt} \approx (2y_{x,t} - y_{x,t-\Delta t} - y_{x,t+\Delta t})/(\Delta t)^2 \qquad (8)$$

By selecting the discrete-time sampling interval Δt to correspond to spatial sampling interval Δx, i.e., $\Delta t = c\Delta x$, and using index notation $k = x/\Delta x$ and $n = t/\Delta t$, Eqs. (7) and (8) result in

$$y_{k,n+1} = y_{k-1,n} + y_{k+1,n} - y_{k,n-1} \qquad (9)$$

which is a *K-model*, i.e., based on Kirchhoff type of total variables, not wave components. From form (9) we can see that a new sample $y_{k,n+1}$ at position k and time index $n+1$ is computed as the sum of its neighboring position values minus the value at the position itself one sample period earlier.

The behavioral similarity of DWGs vs. FDTDs [11], although being computationally different formulations (W- vs. K-models), hints to expand Eq. (9) to a FDTD type scattering junction for arbitrary port admittances. For a parallel admittance model, corresponding to Eq. (6), Eq. (9) can be formulated for N ports as

$$P_{{\rm J},n+1} = \frac{2}{Y_{\rm tot}} \sum_{i=0}^{N-1} Y_i P_{i,n} - P_{{\rm J},n-1} \qquad (10)$$

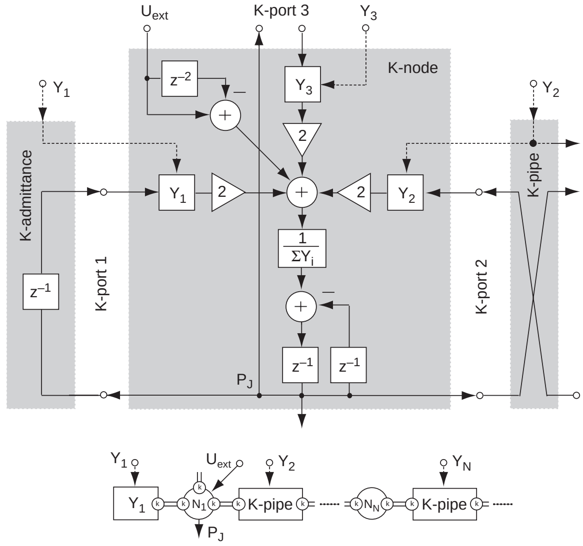

Figure 2: Top: Digital filter structure for finite difference approximation of a 3-port scattering node with port admittances Y_1, Y_2 and Y_3 (port 3 not connected). Only total pressure P_J (K-variable) is explicitly available. Bottom: Block diagram with abstracted blocks to form a 1-D FDTD waveguide.

This is the waveguide mesh formulationas discussed in [8]. Figure 2 depicts a DSP formulation of one such 3-port scattering *K-node* and the way to terminate port 1 by *K-admittance*, Y_1. This corresponds to the W-model in Fig. 1, except that a wave traveling to the left reflects back from Y_1 one unit delay later than in the DWG case. Notice the feedback through a unit delay. The bottom part of Fig. 2 depicts a block diagram abstraction which shows the conformity with the DWG in Fig. 1.

An essential difference between DWGs of Fig. 1 and FDTDs of Fig. 2 is that while DWG junctions are connected through two-directional delay lines (*W-lines*), FDTD nodes have two unit delays of internal memory, and delay-free *K-pipes* connect ports between nodes (see the right-hand side block in Fig. 2). The DWG and FDTD junction nodes and ports are not directly compatible because they use different type of wave variables.

A further difference in K- vs. W-modeling, in addition to algorithmic and computational precision properties, is the way external excitation has to be inserted into a junction node. For a FDTD junction $U_{\rm ext}$ must be fed through filter $H(z) = 1 - z^{-2}$ (see Fig. 2) in order to initiate impulses traveling to opposite directions. An interesting property of the FDTD waveguide is also its inherent numerical integrating ability, utilized in some modeling tasks [9, 10]. If an impulse excitation into a 1-D waveguide node is inserted through $H(z) = 1 + z^{-1}$, there will be step functions propagating to both directions from the excitation point. This integrating property makes it possible to use a pair of variables, such as force and displacement, instead of the physically inherent pair of force and velocity.

1.3. Interfacing DWGs and FDTDs

The next question is the possibility to interface wave-based and FDTD-based submodels. In [5] it was shown how to interconnect a lossy 1-D FDTD waveguide with a similar DWG waveguide into a mixed model using a proper interconnection element (adaptor).

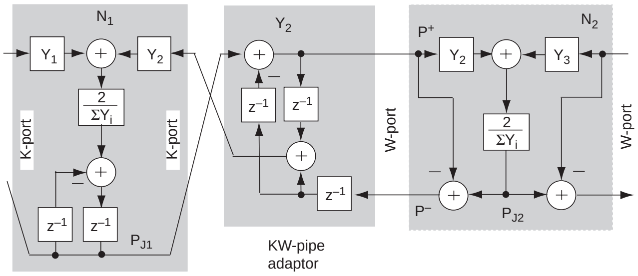

Figure 3: FDTD node (left) and a DWG node (right) forming a part of a hybrid waveguide. Y_i are wave admittances of W-lines, K-pipes, and adaptor KW-pipes between junction nodes. P_{J} are junction pressures, P^+ and P^- are wave components.

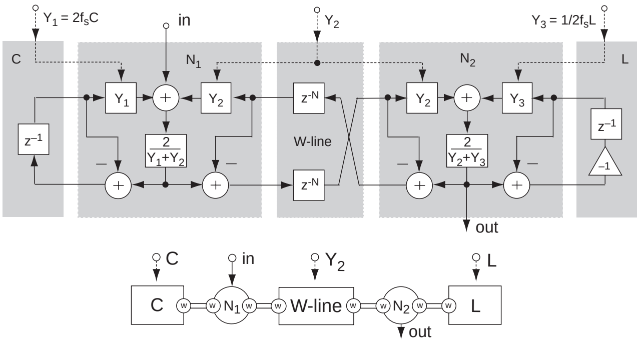

Figure 4: A simple DWG+WDF resonator where a DWG delay line is terminated with a WDF capacitor (left) and inductor (right).

As a generalization, it is possible to make any hybrid model of K-elements (FDTD) and W-elements having arbitrary wave admittances/impedances at their ports.

Figure 3 shows how this can be done in a 1-D waveguide between a K-node N_1 (left) and a W-node N_2 (right). The role of the KW-pipe in the middle of Fig. 3 is to adapt the K-type port of an FDTD node and the W-type port of a DWG node. It is delay-free in left-to-right direction and contains delay in the opposite direction.

Generally the DWG elements are preferable in 1-D modeling due to good numerical properties and possibility of arbitrary (including fractional) delays, while the FDTDs are more efficient in 2-D and 3-D structures, being however more critical in numerical accuracy.

1.4. Interfacing wave digital filters with DWGs

An addition to the mixed modeling above is to adopt Wave Digital Filters (WDF) [3, 11] as discrete-time simulators of lumped parameter elements. Being based on W-modeling (wave variables) they are computationally compatible with the W-type DWGs [11]. A WDF *resistor* is just a real-valued termination (see Y_1) in Fig. 1, but WDF *capacitors* and *inductors* are useful components [3].

As a physically bound choice for the case of this study, a WDF capacitor is realized as a feedback from V^- wave of a port back to V^+ through a unit delay, having a port admittance $2f_{\mathrm{s}}C$. A WDF inductor is a feedback through a unit delay and coefficient -1, having a port admittance $1/2f_{\mathrm{s}}L$. Here C is capacitance, L is inductance, and f_{s} is sample rate (cf. [11]).

Figure 4 shows an example of a model where a DWG delay line is terminated by a WDF capacitor at the left hand side and by a WDF inductor at the right hand side.

A beneficial property of WDF elements is, since their wave admittances are real-valued, that junctions of such ports remain memoryless in the sense of Fig. 1, i.e., Y_i and $1/\sum Y_i$ are real. On the other hand, more flexibility and efficiency may be achieved in practice by higher order approximations of Y_i than by using basic WDF components.

WDF elements, being W-models, are not directly compatible with DFTDs that are K-models. However, the compatibility can be realized through a KW-adaptor element if needed.

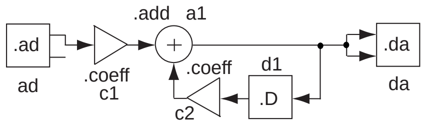

Figure 5: Simple low-pass filtering from sound input to output.

2. REAL-TIME SIMULATION OF MIXED MODELS

The formulation of DWG + FDTD + WDF mixed models above shows directly a way to construct computational physical models based on DSP structures. A software system called the BlockCompiler has been developed that is particularly designed for flexible yet efficient experimentation with physical models [6].

Object-oriented manipulation of blocks and their interactions in the BlockCompiler is implemented in the Common Lisp language. Automatic generation of C source code from interconnected block representation and compilation to a run-time executable provides efficient computation in real time or by sample-by-sample non-realtime simulation.

2.1. BlockCompiler features

The main features and functional principles of the BlockCompiler are briefly described as follows:

• **Block structures and patches**: The basic level of block modeling supports DSP with directed data flow, such as adders, multipliers, delays, and digital filters, see Fig. 5 for a simple low-pass filter example. DSP blocks are instantiated by make-forms such as `(.coeff c1)` for a multiplier, `(.ad)` for sound input and `(.da)` for sound output, `(.d)` for a delay, and `(.add)` for an adder. These are chained by forms `(-> (.ad) ...)` from output to input, output to input, etc. A full model is called a *patch*. The patch of Fig. 5 can be created interactively by scripting

```
(patch ((a (.add)))
   (-> (.ad) (.coeff 0.0666) a (inputs (.da)))
   (-> a (.D) (.coeff -0.8668) (input a 1)))
```

It can be started streaming by command `run-patch`. Audio input is then continuously low-pass filtered to audio output. Inputs and outputs of blocks are normally used for synchronous data flow, but inputs may work also asynchnonously for parametric control.

• **Macro blocks** can be defined as combinations of more elementary blocks. This is a useful abstraction mechanism whereby the details of the the new class objects are hidden.

• **Data types**: The BlockCompiler supports data types {short, long, float, double} and corresponding array types.

• **Multirate support**: Multirate processing is available so that each block can be given a relative sample rate (decimation and interpolation by integer ratio.)

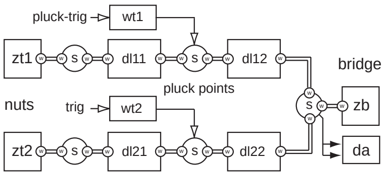

Figure 6: Block diagram of two strings made of delay lines and coupled through a common bridge impedance (z_b). Pluck points can be excited from wavetables (`wt`).

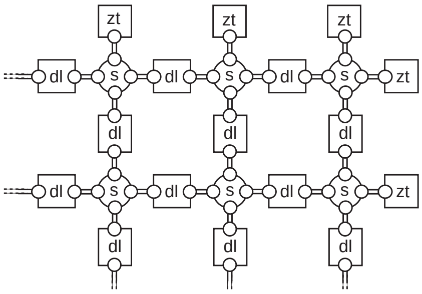

Figure 7: Part of a 2-D rectangular digital waveguide mesh for membrane simulation, composed of delay lines `dl`, series (impedance) nodes, and terminating impedances `zt` at boundaries.

• **Code generation and compilation**: Each block of a patch generates inline C code into a file to make a function with related data and declarations. The resulting C file is then compiled by an automatic call to a C compiler.

2.2. Physical modeling examples

In physical models the elements are coupled through ports of two-directional interaction. Figure 6 shows a block diagram of a 2-string guitar where the strings are coupled through a common bridge impedance (z_b). Each string consists of two delay lines with a pluck point node in between for inserting pluck excitation from a wavetable. The principle of creating a model for the guitar is characterized (only for the upper string) by script:

```
(patch ((zb (.Z :impedance *imped-zb*))
        (zt1 (.Z :impedance *imped-zt1*))
        (wt1 (.wtable *wt-data1*))
        (dl11 (.d-line :delay-length *length11*))
        (dl12 (.d-line :delay-length *length12*))
        ...)
   (-s zt (port dl11 0))
   (-> wt1 (-s (port dl11 1) (port dl12 0)))
   (-> (-s (port dl12 1) (port dl22 1) zb) da)
   ...)
```

Here the blocks are first instantiated (parameter data shown only symbolically) and bound to symbols used in Fig. 6. Then their ports are series connected by forms `(-s ...ports...)` which return the node objects. Excitations and output are wired unidirectionally to and from node objects by `(-> ...)` forms.

Instead of building a full instrument model, such as 6-string guitar, each string with two polarizations, as a single patch from basic elements, a better strategy is to define macro blocks, layer by layer: First a single polarization element, then two of them for a two-polarization string, further on using six of them, a brigde, and a body filter, to finally define a full guitar as a new macro block.

As another example, Fig. 7 depicts a part of a 2-D waveguide mesh in rectangular shape. It can be composed of W- or K-elements, although K-type FDTD elements are more efficient in 2-D and 3-D digital waveguides. Using KW-adaptors the model can mix both types of elements, e.g., for connecting W-type terminators or fractional delays to an FDTD wabeguide mesh. The impedance of each delay can be controlled separately.

3. SUMMARY AND CONCLUSIONS

This paper has presented a physical modeling approach where several paradigms are combined to mixed models, including digital waveguides (DWGs) and second-order finite difference time domain models (FDTDs). Furthermore, wave digital filters (WDFs) are applicable, in addition to DWGs, for wave-based modeling (W-modeling). FDTDs, based on Kirchhoff variables (K-modeling), are made compatible with W-models through an adaptor element. The formulation allows for high flexibility in building 1-D, 2-D, and 3-D physical models from interconnected blocks, supporting both spatially distributed and lumped elements. A software tool is developed where such models can be built and executed efficiently.

4. ACKNOWLEDGMENT

This work is part of EU ALMA project (IST-2001-33059) and of Academy of Finland project "Technology for Audio and Speech Processing" (SA 53537).

5. REFERENCES

[1] J. O. Smith, "Principles of Waveguide Models of Musical Instruments," in *Applications of Digital Signal Processing to Audio and Acoustics*, ed. M. Kahrs and K. Brandenburg, Kluwer Academic Publishers, Boston 1998.

[2] J. Strikverda, *Finite Difference Schemes and Partial Differential Equations*, Wadsworth and Brooks, 1989.

[3] A. Fettweis, "Wave Digital Filters: Theory and Practice," *Proc. IEEE*, 74(2), pp. 270–372, 1986.

[4] C. Erkut and M. Karjalainen, "Virtual strings based on a 1-D FDTD waveguide model," *Proc. AES 22nd Int. Conf.*, pp. 317–323, Espoo, Finland, 2002.

[5] C. Erkut and M. Karjalainen, "Finite Difference Method vs. Digital Waveguide Method in String Instrument Modeling and Synthesis," *Proc. ISMA'02*, Mexico City, 2002.

[6] M. Karjalainen, "BlockCompiler: Efficient Simulation of Acoustic and Audio Systems," *Preprints of AES114th Convention*, Paper 5756, Amsterdam, May 2003.

[7] T. Laakso, V. Välimäki, M. Karjalainen, and U. K. Laine, "Splitting the Unit Delay—Tools for Fractional Delay Filter Design," *IEEE Signal Processing Magazine*, vol. 13, no 1, pp. 30–60, 1996.

[8] S. A. Van Duyne and J. O. Smith, "Physical Modeling with the 2-D Digital Waveguide Mesh," *Proc. ICMC'93*, pp., 40-47, Tokyo, Japan, Sept. 1993.

[9] A. Chaigne, "On the use of finite differences for musical synthesis. Application to plucked stringed instruments", *J. Acoustique,* vol. 5, pp. 181–211, April 1992.

[10] M. Karjalainen, "1-D digital waveguide modeling for improved sound synthesis," *Proc. IEEE ICASSP'2001*, pp. 1869–1872, Orlando, 2002.

[11] S. D. Bilbao, *Wave and Scattering Methods for the Numerical Integration of Partial Differential Equations*, PhD Thesis, Stanford University, May 2001.

[12] A. Sarti and G. De Poli, "Toward Nonlinear Wave Digital Filters," *Proc. IEEE Trans. Signal Processing*, vol. 47, no. 6, pp. 1654–1668, June 1999.