Mixed Physical Modeling Techniques Applied to Speech Production

Matti Karjalainen

Helsinki University of Technology
Laboratory of Acoustics and Audio Signal Processing
P.O. Box 3000, FIN-02015 HUT, Espoo, Finland
matti.karjalainen@hut.fi

Abstract

The Kelly-Lochbaum transmission-line model of the vocal tract started the discrete-time modeling of speech production. More recently similar techniques have been developed in computer music towards a more generalized methodology. In this paper we will study the application of mixed physical modeling to speech production and speech synthesis. These approaches are Digital Waveguides (DWG), Finite Difference Time-Domain schemes (FDTD), and Wave Digital Filters (WDF). The equivalence and interconnectivity of these schemes is shown and flexible real-time synthesizers for articulatory type of speech production are demonstrated.

1. Introduction

The development of practical speech synthesis has recently been directed towards data-driven solutions such as concatenative (sample-based) synthesis and utilization of speech corpora. From a basic reaserch and fundamental understanding point of view, however, modeling of speech production remains a challenging topic where detailed treatment of speech acoustics and motory functions is essential.

A physical (i.e., acoustic) modeling approach was important in the early development of speech synthesis and studies on speech production [1]. The Kelly-Lochbaum transmission-line model of the vocal tract started the discrete-time modeling approach [2]. Many similar methods and techniques have been applied since then but the current trend in speech synthesis has reduced the interest in the acoustic modeling direction.

The development in sound synthesis and computer music has been more the opposite. So called physical modeling techniques have become increasingly popular in musical instrument studies and model-based sound synthesis. Among these approaches are the digital waveguides (DWG) [3, 4], the finite difference time-domain schemes (FDTD) [5, 6, 7], and also the wave digital filters (WDF). [8, 4]. Recently there has been investigations to unite the theory and modeling practice by studying the relationships between these approaches and how to mix them [9, 10, 11, 12]. These methods and techniques are as well applicable to studies in speech sciences. In this paper we will explore particularly the applicability of mixed physical modeling to speech production and speech synthesis as well as applying a modeling tool called BlockCompiler to these cases.

2. Discrete-time physical modeling

In time-domain discrete-time modeling of physical systems the task is to convert the underlying (partial) differential equations into approximating difference equations then to be solved. Formulation of the solution as digital signal processing algorithms makes them computable by existing efficient software tools, even as real-time simulation. In this paper we next introduce DSP formulations of the two main approaches of interest, the digital waveguides (DWG) and the finite difference time-domain schemes (FDTD).

In a one-dimensional lossless medium the wave equation is written

$$y_{tt} = c^2 y_{xx} \tag{1}$$

where y is a wave variable, subscript tt refers to second partial derivative in time t, xx second partial derivative in place variable x, and c is speed of wavefront in the medium of interest.

2.1. Wave-based modeling

The traveling wave formulation is based on the d'Alembert solution of propagation of two opposite direction waves, i.e.,

$$y(t,x) = \overrightarrow{y}(t - x/c) + \overleftarrow{y}(t + x/c) \tag{2}$$

where the arrows denote the right-going and the left-going components of the total waveform. Assuming that the signals are bandlimited to half of sampling rate, we may sample the traveling waves without losing any information by selecting T as the sample interval and X the position interval between samples so that $T = X/c$. Sampling is applied in a discrete time-space grid in which n and m are related to time and position, respectively. The discretized version of Eq. (2) [3] becomes:

$$y(n,m) = \overrightarrow{y}(n - m) + \overleftarrow{y}(n + m) \tag{3}$$

It follows that the wave propagation can be computed by updating state variables in two delay lines by

$$\overrightarrow{y}_{k,n+1} = \overrightarrow{y}_{k-1,n} \quad \text{and} \quad \overleftarrow{y}_{k,n+1} = \overleftarrow{y}_{k+1,n} \tag{4}$$

i.e., by simply shifting the samples to the right and left, respectively. This kind of discrete-time modeling is called Digital Waveguide (DWG) modeling [3].

The next step is to take into account the global physical constraints of continuity by Kirchhoff 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 and volume velocity U, and a parallel admittance model of N ports is utilized, the Kirchhoff constrains become

$$P_1 = P_2 = \ldots = P_N = P_{\mathrm{J}} \tag{5}$$

$$U_1 + U_2 + \ldots + U_N + U_{\mathrm{ext}} = 0 \tag{6}$$

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

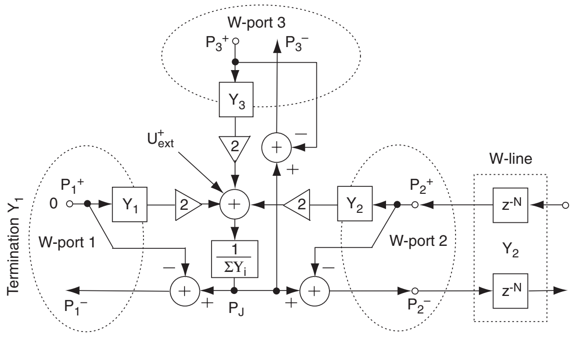

Figure 1: N-port scattering junction (three ports are shown) of ports with admittances Y_i. Incoming pressures are P_i^+ and outgoing pressures P_i^+. W-port 1 is terminated by admittance Y_1 and port 2 by a two-directional delay line (W-line).

$$P_i = P_i^+ + P_i^- \quad \text{and} \quad U_i^+ = Y_i P_i^+ \tag{7}$$

the junction pressure P_J can be obtained as:

$$P_\mathrm{J} = \frac{1}{Y_\mathrm{tot}}(U_\mathrm{ext} + 2\sum_{i=0}^{N-1} Y_i P_i^+) \tag{8}$$

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. (7), are then $P_i^- = P_\mathrm{J} - P_i^+$. The result is illustrated in Fig. 1. When admittances Y_i are frequency-dependent, this diagram can be interpreted as a filter structure where the incoming pressures are filtered by the corresponding wave admittances Y_i times two, and their sum is filtered further by $1/Y_\mathrm{tot}$ to get the junction pressure P_J.

Two special cases can be noticed on the basis of Eq. (8). First, a (passive) loading admittance is the case with Y_i where no incoming pressure wave P_i^+ is associated. This needs no computation except including Y_i in Y_tot because $P_i^+ = 0$, see the left-hand termination in Fig. 1. Another issue is the external velocity U_ext effective to the junction. This is connected directly to the summation at the junction as shown in Fig. 1.

The wave variables and admittances at ports attached to a junction can be specified in any proper transform domain, but we are here interested in z-domain formulations for practical discrete-time computation. Notice that the admittances in Fig. 1 can 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. In the latter case, if we skip the external velocity U_ext of Eq. (8), we may write the equation using scattering parameters α_i as $P_\mathrm{J} = \sum_{i=0}^{N-1} \alpha_i P_i^+$, where $\alpha_i = 2Y_i/Y_\mathrm{tot}$. This and other special forms of scattering are efficient computationally when admittances are real-valued, but in a general case it is practical to implement computation as shown in Fig. 1 so that the term $1/\sum Y_i$ is a common filter.

The freedom to use any impedance in a digital filter formulation allows also for applying numerical (e.g., measured) data, such as lip radiation admittance, as a part of a model. The passivity condition is, as for a scattering junction in general, that Y_i must be positive real. Notice also that the realization of junction nodes as shown in Fig. 1 is general for any linear and time invariant system approximation, also for 2-D and 3-D mesh structures. The delays (see Fig. 1) between nodes can also approximate fractional delays [13], which is useful with varying length tubes. However, delays less than a unit delay generally lead to the problem of delay-free loops, which complicates the situation substantially.

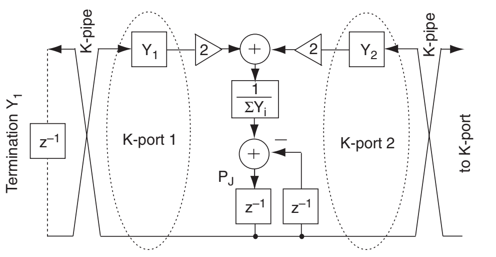

Figure 2: Digital filter structure for finite difference approximation of a two-port scattering node with port admittances Y_1 and Y_2. Only total pressure P_J (K-variable) is explicitly available.

2.2. Finite difference modeling

In the most common way to discretize the wave equation by finite differences the partial derivatives in Eq. (1) are approximated by second order finite differences

$$y_{xx} \approx (2y_{x,t} - y_{x-\Delta x,t} - y_{x+\Delta x,t})/(\Delta x)^2 \tag{9}$$

$$y_{tt} \approx (2y_{x,t} - y_{x,t-\Delta t} - y_{x,t+\Delta t})/(\Delta t)^2 \tag{10}$$

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. (9) and (10) result in

$$y_{k,n+1} = y_{k-1,n} + y_{k+1,n} - y_{k,n-1} \tag{11}$$

which is a special case of multidimensional meshes as an FDTD formulation [3, 14]. From form (11) 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 equivalence of digital waveguides and FDTDs [4], although being computationally different formulations, is also applicable to expand Eq. (11) to a scattering junction with arbitrary port admittances. Figure 2 depicts one scattering node of a 1-D FDTD waveguide and the way to terminate one port by admittance Y_1. There can be any number of ports attached also here as for a DWG junction.

An essential difference between DWGs of Fig. 1 and FDTDs of Fig. 2 is that while DWG junctions are connected through 2-directional delay lines (*W-lines*), FDTD nodes have two unit delays of internal memory and delay-free *K-pipes* connecting ports between nodes. The DWG and FDTD junction nodes and ports are thus not directly compatible (see next subsection). One further difference, in addition to algorithmic and computational precision properties, is the possibility of 'spurious' responses in FDTDs, i.e., an initial state of finite energy may generate waves of infinitely expanding energy in them [5]. This 'non-physical' behavior needs extra computational elements in DWGs to get a similar behavior.

2.3. Interfacing of DWGs and FDTDs

The next question is the possibility to interface wave-based and FDTD-based submodels. In [11] it was shown how to interconnect a lossy 1-D FDTD waveguide with a similar DWG waveguide into a hybrid model using a proper interconnection element (adaptor). In a similar way, 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 is done in a one-dimensional modeling case between a K-node and a W-node. The left-hand node in Fig. 3 is an FDTD waveguide node, and the right-hand part

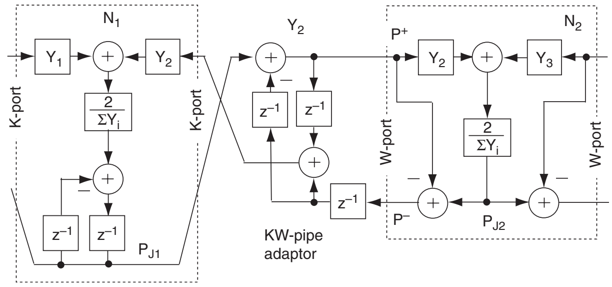

Figure 3: FDTD node (left) and a DWG node (right) forming a part of a hybrid waveguide. Y_i are wave admittances of W-lines, K-pipes, and adaptor KW-pipes between junction nodes. P_{J} are junction pressures, P^+ and P^- are wave components.

is a similar formulation for a wave-based model (DWG). The function of the KW-pipe in the middle of Fig. 3 between the FDTD node N_1 and DWG element N_2 is to adapt the K-type port of an FDTD node and the W-type port of a DWG node, and it is delay-free in one direction.

The proper functioning of the adaptor can be shown by testing the propagation of a left- and a right-traveling impulse through the adaptor. The equivalence and interfacing rules of wave-based and K-variable based (FDTD) models allow now for implementing mixed models where either of the approaches can be applied, depending on which one is more useful in the problem at hand. Generally the DWG elements are preferable in 1-D modeling due to good numerical properties and possibility of arbitrary (including fractional) delays, while the DWGs are more efficient in 2-D and 3-D structures, being however more critical in numerical precision.

2.4. Including lumped and nonlinear elements

A useful additional formalism is to adopt Wave Digital Filters (WDF) [8, 4] as discrete-time simulators of lumped parameter elements. Based on wave variables, they are computationally fully compatible with the structures described above. A WDF *resistor* does not add much to the cases above, but WDF *capacitors* and *inductors*, as well as *ideal transformers* and *gyrators*, etc., are useful components [8].

As a physically bound choice for the case of this study, a WDF capacitor is a feedback from V^- wave of a port back to V^+ through a unit delay, and having a port admittance $2f_{\mathrm{s}}C$. A WDF inductor is a feedback through a unit delay and coefficient -1, and having a port admittance $1/2f_{\mathrm{s}}L$. Here C is capacitance, L is inductance, and f_{s} is the sample rate (cf. [4]). A beneficial property of these 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 is achieved in practice by higher order approximations of Y_i than by using basic WDF components. The WDF formulation helps more essentially in a problem that appears when nonlinearities or fast parametric changes in a system are to be modeled, where delay-free loops may appear, requiring special solutions [15]. This problem is, however out of the scope of this paper.

3. Real-time modeling in BlockCompiler

An experimental software platform has been developed for efficient yet highly flexible development of physical models, called the *BlockCompiler* [12]. Here we just list its most important features, as utilized in modeling speech production.

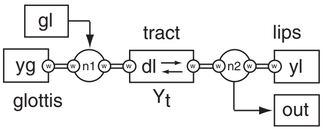

Figure 4: Homogeneous vocal-tract model: Blocks and interconnections for BlockCompiler realization.

Table 1: Model script for the simple vocal tract model in Fig. 4.

```
(patch ((gl (.glottis :pitch 120.0))
        (yg (.admittance *yglott*))
        (dl (.delay-line :delay-time 0.0005
               :admittance *ytract*))
        (yl (.admittance *ylips*))
        (out (.da)))
  (-> gl (-p (port yg) (port dl 0)))
  (-> (-p (port dl 1) (port yl)) (inputs out)))
```

- The best features of two programming languages were combined in the implementation of the Block Compiler: Common Lisp and CLOS object system was used as a high-level symbolic processing environment and the C language for most efficient numeric computation.
- A complete model is called a *patch*, which consists of interconnected *block* items. An example, shown in Fig. 4 and scripted in Table 1, shows how compactly a simple simulator for speech production can be created. The 'patch' macro first instantiates the computational blocks. Local variables 'gl', 'yg', 'dl', 'yl', and 'out' refer to the blocks of glottis oscillator, glottal admittance, vocal tract delay line, lip radiation admittance, and sound output (D/A), respectively. Then the functions '->' and '-p' are used (in nested functional forms) to make parallel port junctions (by '-p') and signal routing from glottis and to sound output (by '->') for real-time streaming.
- Multirate processing is available so that each block can be given a relative sample rate.
- Macro blocks can be defined as containers of more elementary blocks. Parameters can be passed to specify the details of a macro block during model creation.
- Data types {short,long,float,double} and corresponding array types are available for signal data. Data is normally transferred between blocks in (multi-rate) synchronous dataflow. Parameter control flow is available that can be asynchronous.
- When a given model specification is evaluated as a Lisp script, an object-based patch is created and memory is allocated. Generation of executable code is done in the next steps.
- The patch is scheduled by walking the hierarchical structure and ordering the elementary operations. If there are delay-free loops or illegal structures, an error is reported.
- Each block writes inline C code into a file to create a single function with related data and declarations. (C code generation of each block class has been defined in 'pseudo-C' which looks like C code but data references are to Lisp.) The resulting C file is then compiled by an automatic call to a C compiler.
- The function pointer of the compiled code is taken and connected to the sample stream of the sound driver for real-time processing, or it can be called in a step-by-step mode.
- While real-time streaming is running, the patch is fully controllable from Lisp level, allowing for highly flexible control and inspection of the model behavior.

The physical modeling principles introduced above can be applied to form arbitrary networks, also multidimensional ones. Presently there is no graphic editor of block structures, since textual scripting is found more powerful in research work.

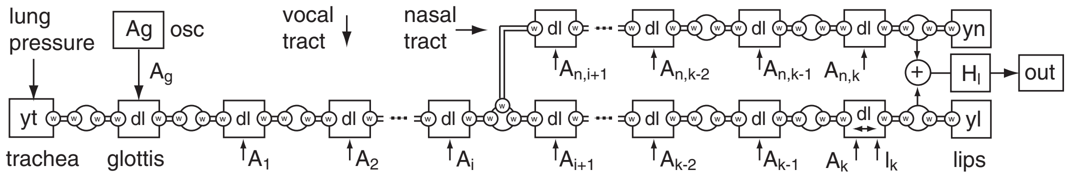

Figure 5: DWG transmission-line speech production model, including nasal-tract, made of (mostly) constant length sections.

4. Modeling of speech production

The flexible properties of the BlockCompiler make it easy to create effcient simulation models for speech production. We discuss two intresting cases using block diagrams (scripting code is not shown because its detailed explanation is too verbose for a short paper).

Figure 5 illustrates an advanced transmission-line model, which however follows traditional guidelines. The tract is divided into a set of constant length sections whereby acoustic admittances $Y(i)$ can be controlled according to their cross-sectional area dependency $Y(i) = A(i)/\rho c$, where ρ is air density and c is speed of sound. Parametric control of vocal tract shape can be based for example on mapping from articulatory parameters, or by contextual lookup and interpolation in time. Fine-tuning of the tract length can be made at the lips by a single controllable fractional delay line section.

The glottis is realized as a single section of vocal tract with varying area, controlled by a glottal waveform oscillator. Lung pressure makes the volume flow through the glottis in relation to its opening. More advanced nonlinear models of self-oscillation can also be experimented easily.

The termination in the model of Fig. 5 at lips and nostrils includes a filter H_l for detailed lip pressure to far-field radiation function. Other functionalities that can be experimented relatively easily are for example generation of turbulence frication and bursts in constrictions and during the opening of occlusion.

Another type of solution is depicted in Fig. 6 where the vocal tract is divided into variable length sections. 4–5 such sections can approximate the tract in an articulatory approach so that the sections inherently correspond to moving parts of the articulators, such as the tongue body. This principle was introduced originally in [16], and it is computationally efficient enough due to flexible control of fractional delay lines in the BlockCompiler. With additional impedances at junctions the principle can simulate also tract models with conical sections.

Concluding remarks

Generalization of physical modeling methods is developed in this paper in a form applicable to the modeling of speech production and articulatory speech synthesis. A software tool, the BlockCompiler, is described shortly as an embodiment of the mixed modeling principles. The compactness and flexibility to create real-time speech production models on a systematic basis is demonstrated. This allows for future investigation of new and increasinly detailed models.

5. Acknowledgments

The study is part of the Academy of Finland projects 53005 and 53537 as well as the EU project "ALMA" (IST-2001-33059).

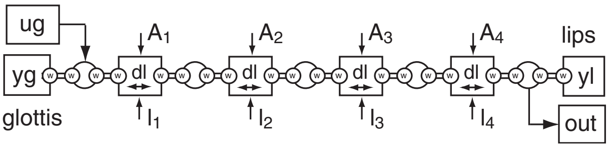

Figure 6: Speech production using vocal-tract sections of variable length (l_i) and cross-area (A_i). (No nasal tract is shown.)

6. References

[1] G. Fant, *Acoustic Theory of Speech Production*, Mounton and Co., 1960.

[2] J. L. Kelly and C. C. Lochbaum, "Speech Synthesis," *Proc. 4th Int. Congr. Acoust.*, paper G42, 1962.

[3] 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.

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

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

[6] L. Hiller and P. Ruiz, "Synthesizing Musical Sounds by Solving the Wave Equation for Vibrating Objects: Part I, Part II," *J. Audio Eng. Soc.* vol. 19, nr. 6 and nr. 7, pp. 452–470 and 542–551, 1971.

[7] 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.

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

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

[10] 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.

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

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

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

[14] L. Savioja, *Modeling Techniques for Virtual Acoustics.* PhD thesis, Helsinki Univ. of Tech., Espoo, Finland, 1999.

[15] 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.

[16] V. Välimäki, M. Karjalainen, and T. Kuisma, "Articulatory Control of a Vocal Tract Model Based on Fractional Delay Waveguide Filters," *Proc. IEEE ISSIPNN*, pp. 571-574, Hong Kong, April 1994.