Wave Digital Filters

A Wave digital filter (WDF) is a particular kind of digital filter (or finite difference scheme) based on physical modeling principles.

- Developed to digitize lumped electrical circuit elements:
  - inductors
  - capacitors
  - resistors
  - gyrators, circulators, etc., (classical circuit theory)
- Each element is digitized by the bilinear transform
- Wave variables are used in place of physical variables (new), yielding superior numerical properties.
- Element connections involve wave scattering
Wave Digital Filter (WDF) Construction

Wave digital elements may be derived from their describing differential equations (in continuous time) as follows:

1. Express forces and velocities as _sums of traveling-wave components_ ("wave variables"): 
   
   \[ f(t) = f^+(t) + f^-(t) \]
   \[ v(t) = v^+(t) + v^-(t) \]
   
   The actual "travel time" is always _zero_.
   (For historical reasons, WDFs typically use traveling-wave components scaled by 2.)

2. Digitize via the _bilinear transform_ (trapezoid rule)

3. Use _scattering junctions_ ("_adaptors"") to connect elements together in
   - _series_ (common velocity, summing forces), or
   - _parallel_ (common force, summing velocities).
Wave Variable Decomposition

Introduced Infinitesimal Transmission Line

\[ F^+ (s) \rightarrow F(s) \rightarrow F^-(s) \]

- The inserted waveguide impedance \( R_0 \) is arbitrary because it was physically introduced.
- The element now interfaces to other elements by abutting its waveguide (transmission line) to that of other element(s).
- Such junctions involve lossless wave scattering:

\[
\begin{align*}
F_R^+ (s) &= T(s)F^+(s) + K_R(s)F^-(s) \\
F^- (s) &= T_R(s)F_R^-(s) + K(s)F^+(s)
\end{align*}
\]
Element Reflectance

Imposing *physical continuity constraints* across the junction:

\[ F(s) = F_R(s) \]
\[ 0 = V(s) + V_R(s) \]

with

\[ F(s) = F^+(s) + F^-(s) \]
\[ F_R(s) = F_R^+(s) + F_R^-(s) \]
\[ V(s) = V^+(s) + V^-(s) = \frac{F^+(s)}{R_0} - \frac{F^-(s)}{R_0} \]
\[ V_R(s) = V_R^+(s) + V_R^-(s) = \left[ \frac{F_R^+(s)}{R(s)} - \frac{F_R^-(s)}{R(s)} \right] \]

we obtain the *reflection transfer function* ("reflectance") of the element with impedance \( R(s) \):

\[
S_R(s) \triangleq \frac{F^-}{F^+} = \frac{R(s) - R_0}{R(s) + R_0}
\]

This is the *impedance step over the impedance sum*, the usual force-wave reflectance at an impedance discontinuity, but now in the Laplace domain.
Reflectance of Ideal Mass, Spring, and Dashpot

For a mass $m$ kg, the impedance and reflectance are respectively

$$R_m(s) = ms$$

$$\Rightarrow S_m(s) = \frac{ms - R_0}{ms + R_0}$$

This reflectance is a stable first-order allpass filter, as expected, since energy is not dissipated by a mass.

For a spring $k$ N/m, we have

$$R_k(s) = \frac{k}{s}$$

$$\Rightarrow S_k(s) = \frac{k - R_0}{k + R_0}$$

also allpass as expected.

For a dashpot $\mu$ N s/m, we have

$$R_\mu(s) = \mu$$

$$\Rightarrow S_\mu(s) = \frac{\mu - R_0}{\mu + R_0}$$
Bilinear Transformation

To digitize via the bilinear transform, we make the substitution

\[
s = c \frac{1 - z^{-1}}{1 + z^{-1}}
\]

where \( c \) is any positive real constant (typically \( 2/T \)).

For the ideal mass reflectance

\[
S_m(s) = \frac{ms - R_0}{ms + R_0}
\]

the bilinear transform yields

\[
\tilde{S}_m(z) = \frac{p_m - z^{-1}}{1 - p_m z^{-1}}
\]

with

\[
p_m \equiv \frac{mc - R_0}{mc + R_0}
\]

Note that \( |p_m| < 1 \) and \( |\tilde{S}_m(e^{j\omega T})| = 1 \). The stable allpass nature of the digitized mass reflectance is preserved by the bilinear transform, as always.

**Important Observation:**

If we choose \( R_0 = mc \), then \( p_m = 0 \) and

\[
\tilde{S}_m(z) = -z^{-1} \Rightarrow \text{no delay-free path through the mass reflectance}
\]

7
Digitized Reflectances Without Delay-Free Paths

Plan:

1. Fix the bilinear-transform frequency-scaling parameter $c$ once for the whole system (so there is only one frequency-warping)

2. Set the “connector” wave impedance $R_0$ separately for each circuit element to eliminate the delay-free path in its reflectance

3. We will then get scattering when we connect different elements together

This yields the following elementary reflectances:

<table>
<thead>
<tr>
<th>Element</th>
<th>Reflectance</th>
</tr>
</thead>
<tbody>
<tr>
<td>ideal spring (capacitor)</td>
<td>↔ unit delay</td>
</tr>
<tr>
<td>ideal mass (inductor)</td>
<td>↔ unit delay and sign inversion</td>
</tr>
<tr>
<td>ideal dashpot (resistor)</td>
<td>↔ 0</td>
</tr>
</tbody>
</table>

The original element values remain only in the waveguide-interface impedances $R_0 = k/c, mc, \mu$
Wave Digital Elements

In summary, our chosen digital element reflectances (and their connecting wave impedances $R_0$) are

- “Wave digital mass” (interface impedance $R_0 = mc$)
  \[
  \tilde{S}_m(z) = -z^{-1}
  \]  
  (mass reflectance)

- “Wave digital spring” ($R_0 = k/c$)
  \[
  \tilde{S}_k(z) = z^{-1}
  \]  
  (spring reflectance)

- “Wave digital dashpot” ($R_0 = \mu$)
  \[
  \tilde{S}(z) = 0
  \]  
  (dashpot [non-]reflectance)

(In this case, the interface is the element itself.)

These are the discrete-time reflectances of the basic circuit building-blocks as seen from their interface-waveguides.

We still have the usual freedom in choosing our bilinear-transform frequency-scaling constant $c$. 
Elementary Wave Flow Diagrams

Wave digital mass

Wave digital spring

Wave digital dashpot
Example: “Piano hammer in flight”

Mass $m$ at constant velocity, force-wave simulation:

- The reflecting termination on the left corresponds to zero force on the mass
- A nonzero state variable $x(n)$ corresponds to a nonzero velocity for the mass:

\[
v(n) = v^+(n) + v^-(n) = \frac{f^+(n)}{R_0} - \frac{f^-(n)}{R_0} = \frac{f^+(n)}{mc} + \frac{f^+(n-1)}{mc} = \frac{x(n+1) + x(n)}{mc} = \frac{2}{mc}x(n) = \frac{T}{m}x(n)
\]

when $c = 2/T$ is chosen for the bilinear transform
Mass Momentum and Energy

• Above we found the mass velocity to be

\[ v(n) = \frac{2}{mc} x(n) = \frac{T}{m} x(n) \]

when \( c = \frac{2}{T} \) is chosen for the bilinear transform

• The momentum of the mass is therefore

\[ p(n) \triangleq m v(n) = \frac{2}{c} x(n) = T x(n) \]

when \( c = \frac{2}{T} \)

• State variable \( x(n) = p(n)/T \) is mass momentum per sample

• Since momentum is conserved, momentum waves are good to consider in place of velocity waves

• The kinetic energy of the mass is given by

\[ \mathcal{E}_m = \frac{1}{2} m v^2(n) = \frac{p^2(n)}{2m} = \frac{2}{mc^2} x^2(n) \rightarrow \frac{[T x(n)]^2}{2m} \]

for \( c \rightarrow 2/T \)

• The potential energy of the mass-in-flight is of course zero \((f(n) \equiv 0)\)
Force Driving a Mass

\[ f(n) = f^+(n) + f^-(n) \quad \Rightarrow \quad f^+(n) = f(n) - f^-(n) \]

Wave digital mass driven by external force \( f(n) \).
Traveling-Wave View of Driving Force

\[ f_1^+(n) = \frac{f(n)}{2} \]

\[ f_1^-(n) = \frac{f(n)}{2} \]

\[ R_0 = 0 \]

\[ f_2^-(n) = f_m^+(n) \]

\[ f_2^+(n) = f_m^-(n) \]

\[ R_0 = mc \]

\[ x(n) \]

Wave digital mass

- Parallel junction with \( R_0 = 0 \) on the force side and \( R_0 = mc \) on the mass side
- Impedance step over impedance sum is
  \[ R = \frac{(mc - 0)}{(mc + 0)} = 1 \]
- Obviously non-physical (see next page)
Zero Source-Impedances are Non-Physical

We postulated the following driving-source interface:

\[ f_1^+(n) = \frac{f(n)}{2} \]

\[ f_1^-(n) = \frac{f(n)}{2} \]

\[ R_0 = 0 \]

\[ f_2^-(n) = f_m^+(n) \]

\[ f_2^+(n) = f_m^-(n) \]

Non-physical because:

- Velocity transmission is zero \( \Rightarrow \) no power delivered
- There can be no traveling force (voltage) wave in a zero impedance (which would “short it out”)
- Recall power waves: \( [f^+(n)]^2 / R_0 = \infty \) if \( f^+(n) \neq 0 \)
- Zero source-impedances can be a useful idealization, but be careful
- Exercise: Study the case of small \( R_0^+ = \epsilon > 0 \).
Spring-Driven Mass

To keep the model physical, let’s use a pre-compressed spring as our force-source for driving the mass:

\[ f = f_m = f_k \]

\[ v_k(t) + v_m(t) = 0 \]

Physical Diagram

- The mass and spring form a loop, so the connection can be defined as either parallel or series (as determined by the element reference directions)
• We arbitrarily choose a parallel junction, giving the following physical constraints:

\[- f_k(n) = f_m(n) \] (common force)

\[- v_k(n) + v_m(n) = 0 \] (sum of spring-compression-velocity and rightgoing-mass velocity is zero)

• Exercise: Work out the case for a series junction and verify everything comes out the same physically

• Connecting our wave digital spring and mass at a parallel force-wave junction is depicted as follows:

Note the WDF symbol “||” for a parallel adaptor (scattering junction)
Expanded Wave Digital Spring-Mass System

\[ f_k^-(n) \quad x_1(n) \quad z^{-1} \quad f_k^+(n) \quad \]
\[ + s \quad 1 + s \quad \]
\[ f_m^-(n) \quad -s \quad -s \quad f_m^+(n) \quad z^{-1} \quad x_2(n) \quad \]

State variables labeled \( x_1(n) \) and \( x_2(n) \)

**Low-Frequency Analysis:**

- Assume sampling rate \( f_s = 1/T \) is large ⇒
- Bilinear transform constant \( c = 2/T \)
- Frequency warping not an issue
- Physical simulation should be very accurate

The reflection coefficient for our parallel force-wave connection is given as usual by the *impedance step over the impedance sum*:

\[
s = \frac{mc - k/c}{mc + k/c} = \frac{m2/T - kT/2}{m2/T + kT/2} = \frac{m - kT^2/4}{m + kT^2/4} \approx 1
\]

We can now see what’s going physically at low frequencies relative to the sampling rate:
Low-Frequency Spring-Driven-Mass Analysis

Referring to the previous figure:

- We found earlier that \( x_2(n) \approx p_m(n)/T \) where \( p_m(n) \) is the mass momentum at time \( n \), and \( T \) is the sampling interval.

- We similarly find that \( x_1(n) = f_k^-(n) \approx f(n)/2 \), so that the mass sees \((1 + s)f(n)/2 \approx f(n)\) coming in each sample from the summer, i.e.,

\[
\frac{p_m(n)}{T} \approx \frac{p_m(n - 1)}{T} + f(n)
\]

- Multiplying through by \( T \) gives the momentum update per sample:

\[
p_m(n) \approx p_m(n - 1) + f(n)T \triangleq p_m(n - 1) + \Delta p(n)
\]

where \( \Delta p(n) \triangleq f(n)T \) is the momentum transferred to the mass by constant force \( f(n) \) during one sampling interval \( T \).

- This makes physical sense and suggests momentum and momentum-increment samples as an appealing choice of wave variables.
We have been using our usual traveling-wave decomposition of force and velocity waves:

\[
\begin{align*}
f(t) &= f^+(t) + f^-(t) = R_0 v^+(t) - R_0 v^-(t) \\
v(t) &= v^+(t) + v^-(t) = \frac{f^+(t)}{R_0} - \frac{f^-(t)}{R_0}
\end{align*}
\]

where \( R_0 \) is the wave impedance of the medium, or

\[
\begin{bmatrix}
f(t) \\ v(t)
\end{bmatrix} = \begin{bmatrix} R_0 & -R_0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} v^+(t) \\ v^-(t) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ \frac{1}{R_0} & -\frac{1}{R_0} \end{bmatrix} \begin{bmatrix} f^+(t) \\ f^-(t) \end{bmatrix}
\]

Inverting these gives

\[
\begin{align*}
v^+(t) &= \frac{1}{2} \begin{bmatrix} 1/R_0 & 1 \end{bmatrix} \begin{bmatrix} f(t) \\ v(t) \end{bmatrix} \\
v^-(t) &= \frac{1}{2} \begin{bmatrix} 1 & R_0 \end{bmatrix} \begin{bmatrix} f(t) \\ v(t) \end{bmatrix}
\end{align*}
\]

\[
\begin{align*}
f^+(t) &= \frac{1}{2} \begin{bmatrix} 1 & R_0 \end{bmatrix} \begin{bmatrix} f(t) \\ v(t) \end{bmatrix} \\
f^-(t) &= \frac{1}{2} \begin{bmatrix} 1 & -R_0 \end{bmatrix} \begin{bmatrix} f(t) \\ v(t) \end{bmatrix}
\end{align*}
\]

In the WDF literature, the second case is typically used, multiplied by 2, and replacing force and velocity by voltage and current:

\[
\begin{align*}
a(t) &= v(t) + R_0 i(t) \\
b(t) &= v(t) - R_0 i(t)
\end{align*}
\]

where \( v(t) \) is now voltage and \( i(t) \) denotes current.

Thus, \( a(t) = 2v^+(t) \) and \( b(t) = 2v^-(t) \) (doubled voltage traveling-wave components)
Binary Connection Tree

It has become common practice to organize WDF elements into a *Binary Connection Tree* (BCT):

![Binary Connection Tree Diagram]

21
Reflection-Free Ports

• The symbol ⊥ on a WDF adaptor port denotes a reflection-free port (RFP)

• To make a port reflection-free, its wave-impedance must be the
  – \textit{parallel combination} of the other port impedances for a parallel adaptor, or
  – \textit{series combination} of the other port impedances for a series adaptor

This choice of port impedance zeros the impedance step “seen” by waves in the RFP, thus suppressing instantaneous reflection from it

• All ports outgoing from the BCT root must be RFPs, for computability (no delay-free loops)

• Computations propagate (each sample) from the leaves of the tree (delay element outputs) up to the root, where there is a final reflection which then propagates back down to all of the reflection-free ports, thereby updating all of the delay elements (capacitor/spring and inductor/mass states)

• When an element value changes (typically a resistor), RFPs must be \textit{recalculated up to the root}.
Reflection-Free Port Coefficients

For an $N$-port adaptor, with port wave-impedances $R_i$, $i = 1, 2, \ldots, N$, let’s arbitrarily designate port $N$ as the reflection-free port (the one on top). It is convenient to define the port conductances $G_i \triangleq 1/R_i$. To suppress reflection on port $N$, we need, for a parallel adaptor,

$$R_N = R_1 \parallel R_2 \parallel \cdots \parallel R_{N-1} \iff G_N = G_1 + G_2 + \cdots + G_{N-1}$$

and, for a series adaptor,

$$R_N = R_1 + R_2 + \cdots + R_{N-1}.$$

Recall the alpha parameters for an $N$-port series scattering junction, derived from the physical constraints that the velocities be equal and the forces sum to zero at the (series) junction:

$$\alpha_i \triangleq \frac{2R_i}{R_1 + R_2 + \cdots + R_N} = \frac{R_i}{R_N}$$

when port $N$ is reflection free.

Since $\sum_{i=1}^{N} \alpha_i = 2$, we have $\alpha_N = 1$ and $\sum_{i=1}^{N-1} \alpha_i = 1$. 

23
Shockley diode equation ("diode law")

\[ I(t) = I_s \cdot \left( \frac{V_d}{e^{nV_T}} - 1 \right) \]

where

- \( I \) = diode current
- \( I_s \) = diode reverse leakage current
- \( V_d \) = voltage across the diode
- \( n \) = ideality factor (1 for ideal, up to 2 or more otherwise)
- \( V_T \) = thermal voltage \( kT/q \)
- \( k \) = Boltzmann constant
- \( q \) = electron charge
- \( T \) = temperature

See page 42 of David Yeh’s WDF Tutorial[1]

Topology Issues

• Classical WDFs are composed of parallel and series connections of elements

• A Binary Connection Tree (BCT) can represent any such parallel/series network

• $R$-Nodes
  – Some circuits, such as the “bridged T” circuit, cannot be represented using parallel/series connections of elements
  – These circuits are modeled using more general scattering matrices
  – Such circuits are called $R$-Nodes in the overall WDF network graph
  – $R$-Nodes connect naturally to BCT graphs, since all signals are compatible traveling-wave components
  – An open issue is how to minimize the computational complexity of $R$-node scattering matrices
SPQR Decomposition

Every graph can be decomposed into Series (S), Parallel (P), and R (“Rigid”) type subgraphs (Q is the degenerate case consisting of only one graph edge)

- S and P handled by standard WDF methods (BCT)
- R node characterized by its scattering matrix
- Modified Nodal Analysis (MNA) may be used to find the R-node scattering matrix (see Werner et al. reference below)
Digital filters can be expressed in state-space form as

\[ x(n + 1) = A x(n) + B u(n) \]

by simply enumerating all delay elements as state variables \( x^T(n) = [x_1(n), x_2(n), \ldots, x_N(n)] \), and finding the state transition matrix \( A \) by inspection. Any inputs are collected in \( u(n) \) and determine the \( B \) matrix.

- For WDFs, the \( A \) matrix is a scattering matrix.
- The \( A \) matrix is orthogonal (lossless) for reactive elements (masses, springs).
- The state variables are all sampled traveling waves.
- Physical state variables (bilinear transformed) are obtainable by summing (capacitors, springs) or subtracting (inductors, masses) the input and output of the unit delays:

\[ y_k(n) = x_k(n) \pm x_k(n - 1) \]

- In comparison to other state-space models, WDF state-space form has top numerical properties due to its lossless scattering formulation.
A WDF network tree can have a multiport *instantaneous nonlinearity* at its root:

- A typical instantaneous nonlinearity is a *nonlinear resistor* $R(v)$ (such as a diode) or a *dependent source* (as used in transistor models, etc.)
- Because the resistance of a nonlinear resistor depends on the voltage across it, there is no way to avoid an instantaneous reflection in general (no fixed port-impedance can match it for all input conditions)
- The nonlinearity is placed at the *root* of the BCT. A delay-free path is “computable” only there (we get one per tree)
- Each sample, computations propagate up the tree to the root, reflecting instantaneously, then back down to all the reflection-free ports
- The nonlinear reflectance can be pre-computed and stored for fast interpolated table look-up in real time (no iterations)
- If the nonlinearity cannot be placed at the root of the WDF BCT (e.g., because there are two or more
nonlinearities in the circuit) the delay-free-path may be solved iteratively using Newton’s method et al.

• Alternatively, all nonlinearities can be placed at the root of the WDF tree and connect to the BCT through an R-Node. References:

  1. “Wave Digital Filter Adaptors for Arbitrary Topologies and Multiport Linear Elements”[2]
  2. “Resolving Wave Digital Filters with Multiple/Multiport Nonlinearities”[3]

Kurt Werner et al.
Int. Conf. Digital Audio Effects (DAFx-15)
Trondheim, Norway, 2015

**Dynamic Nonlinearities**

Nonlinearities can be *instantaneous* or *dynamic* (having *memory*)

• A dynamic nonlinearity can sometimes be converted into an instantaneous nonlinearity:

• Convert to the physical units in which the nonlinearity is instantaneous

Choice of WDF Topology

Summarizing points above,

- Generally try to make a Binary Connection Tree (BCT) using only three-port adaptors
- At the *root* of the tree, include all
  - nonlinearities
  - non-adaptable elements such as switches
- When everything is linear and adaptable, place a time-varying element at the root, to minimize update propagation when that element changes
- When multiple elements are at the root, or when topology is not merely series + parallel connections, there will generally be at least one $R$ node
Free WDF Software

Real Time Wave Digital Filter Software (DAFx-2016):

- GitHub: RT-WDF
- DAFx16 Paper

Overview and Demo of Various Wave Digital Filter Software (DAFx-2015, KeyNote 2, Part 2):

- Video (YouTube)
- Slides (PDF)

WDF References


