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

- 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:
  \[ F^+(s) = T(s)F^+(s) + K(s)F^-R(s) \]
  \[ F^-(s) = T_R(s)F^-R(s) + K(s)F^+(s) \]
**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 \( R(s) \):

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

This is the impedance step over the impedance sum, the usual force-wave reflectance at an impedance discontinuity, but now in the Laplace domain.

**Bilinear Transformation**

To digitize via the bilinear transform, we make the substitution

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

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

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

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

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

**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</td>
<td>↔ unit delay</td>
</tr>
<tr>
<td>ideal mass</td>
<td>↔ unit delay and sign inversion</td>
</tr>
<tr>
<td>ideal dashpot</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 $R_0 = mc$)
  \[ S_m(z) = -z^{-1} \quad \text{(mass reflectance)} \]

- “Wave digital spring” ($R_0 = k/c$)
  \[ S_k(z) = z^{-1} \quad \text{(spring reflectance)} \]

- “Wave digital dashpot” ($R_0 = \mu$)
  \[ S(z) = 0 \quad \text{(dashpot [non-]reflectance)} \]
  (In this case, the interface is the element itself.)

These are the \textit{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 $c$.

Example: “Piano hammer in flight”

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

(a)

\[
\begin{align*}
  f^+(n) &\quad z^{-1} &\quad f^-(n) \\
-1 & & -1 \\

  x(n) &\quad = &\quad x(n)
\end{align*}
\]

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

  when $c = 2/T$ is chosen for the bilinear transform.

(b)

Mass Momentum and Energy

- Above we found the mass \textit{velocity} to be
  \[
  v(n) = \frac{2}{mc} x(n) = \frac{T}{m} x(n)
  \]
  when $c = 2/T$.
- The \textit{momentum} of the mass is therefore
  \[
  p(n) \triangleq m v(n) = \frac{2}{c} x(n) = T x(n)
  \]
  when $c = 2/T$.
- State variable $\left[ x(n) = p(n)/T \right]$ is \textit{mass momentum per sample}.
- Since momentum is conserved, \textit{momentum waves} are good to consider in place of velocity waves.
- The \textit{kinetic energy} of the mass is given by
  \[
  \mathcal{E}_m = \frac{1}{2} m v^2(n) = \frac{p^2(n)}{2m} = \frac{2}{mc^2} x^2(n) \to \frac{[T x(n)]^2}{2m}
  \]
  for $c \to 2/T$.
- The \textit{potential energy} of the mass-in-flight is of course zero ($f(n) \equiv 0$).
**Force Driving a Mass**

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

(a) \hspace{2cm} (b)

![Diagram](image)

Wave digital mass driven by external force \( f(n) \).

**Traveling-Wave View of Driving Force**

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

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

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:

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

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

\[ f(n) = f_m = f_k \]

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: \( \frac{[f^+(n)]^2}{R_0} = \infty \) 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:

\[ f = f_m = f_k \]

Physical Diagram

Equivalent Circuit

- 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)
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 \( x_1(n) \approx f_k(n) \), 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

low-frequency analysis:

- assume sampling rate \( f_s = 1/T \) is large \( \Rightarrow \)
  - 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:

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

we can now see what’s going physically at low frequencies relative to the sampling rate:

expanded wave digital spring-mass system

classic wave variables

we have been using our usual traveling-wave decomposition of force and velocity waves:

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

\[
 v(t) = v^+(t) + v^-(t) = \frac{f^+(t)}{R_0} - \frac{f^-(t)}{R_0}
\]

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

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

inverting these gives

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

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

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

\[
 a(t) = v(t) + R_0 i(t)
\]

\[
 b(t) = v(t) - R_0 i(t)
\]

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):

Reflection-Free Ports

- The symbol \( \perp \) on a WDF adaptor port denotes a reflection-free port (RFP)
- To make a port reflection-free, its wave-impedance must be the
  - parallel combination of the other port impedances for a parallel adaptor, or
  - 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 recalculated up to the root.

Example

See page 42 of David Yeh’s WDF Tutorial

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

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

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

WDF State Space Interpretation

Digital filters can be expressed in state-space form as
\[ \bar{x}(n+1) = A \bar{x}(n) + B u(n) \]
by simply enumerating all delay elements as state variables \( \bar{x}(n) = [x_1(n), x_2(n), \ldots, x_N(n)] \), 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

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)

Nonlinear Wave Digital Filters

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. “Resolving Wave Digital Filters with Multiple/Multiport Nonlinearities”
   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

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


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


