Outlines

- Horn Modeling (Trumpet)
- Piecewise Conical Bore Modeling
- Truncated Infinite Impulse Response (TIIR) Filters

Inverse scattering applied to pulse-reflectometry data to fit piecewise-cylindrical model (like LPC model)

Bore profile reconstruction is reasonable up to bell

The bell is not physically equivalent to a piecewise-cylindrical acoustic tube, due to

- complex radiation impedance,
- conversion to higher order transverse modes
Trumpet-Bell Impulse Response computed from estimated piecewise-cylindrical model

- From pulse reflectometry on trumpet with no mouthpiece
- Bore profile is reconstructed, smoothed, and segmented
- Impulse response of “bell segment” = “ideal filter”
- At $f_s = 44.1$ kHz, filter length is $\approx 400$ to $600$ samples

- A length 400 FIR bell filter is too expensive!
- Convert to IIR? Hard because
  - Phase (resonance tunings) must be preserved
  - Magnitude (resonance Q) must be preserved
  - Rise time $\approx 150$ samples
  - Phase-sensitive IIR design methods perform poorly
FIR to IIR Conversion Attempts

Bell Impulse Response (dB) Before Truncation

- 561 samples gives cut-off around -60 dB relative to maximum
- This length 561 FIR filter can be reduced to a lower-order IIR filter by minimizing some norm of the impulse-response error

Hankel Norm Method

- Hankel norm minimization should always work in theory

Eigenvalues of Hankel Matrix (dB)
Largest Eigenvalues of Hankel Matrix (dB)

- Order 15 is a “sweet spot” in the eigenvalues plot
- Hankel Norm is the only phase-sensitive IIR error norm we know which can always be reliably minimized in principle
• Norm is sensitive to *linear* magnitude error, not dB
• This bell filter is too “bright” and fit is generally poor
• Initial time-domain match is reasonable, but it can’t “hold on” until the main reflection
• Numerical failure is a likely (in Matlab/PentiumII doubles)

Order 8 Hankel-Norm IIR Fit to Length 561 FIR
(Evidence of Numerical Failure in Previous Example)

- Halving the order actually looks better ("can’t happen")
- Error plot indicates numerical troubles here as well
• An order $P$ IIR filter is made using $P$th eigenvector of the $561 \times 561$ Hankel matrix (condition number = $51751075$)
• Numerical failure occurs at the higher orders we need
• Slow rise time of impulse response causes “numerical stress” on all phase-sensitive IIR design methods when the IIR order is much less than the rise time

• All poles concentrated at low frequencies
• Little attention to high frequencies
• Internal “equation-error” weighting
• Numerical ill-conditioning warning printed by Matlab
SM-10 Amplitude Response Fit

Magnitude Fit over Entire Nyquist Band

Another Measured Trumpet Bell Reflectance

Measured Trombone Bell Reflectance
Idea!

- Break up impulse response into exponential or polynomial segments
- Exponential and polynomial impulse-responses can be designed using Truncated IIR (TIIR) Filters
Exp-4 Impulse Response Fit (Repeated)

Two Exponentials Connected by a Cubic Spline
Measured Trumpet Data (Exp2-S3)

Exp-4 Slope Fit

Exp2-S3 Slope Fit
Exp2-S3 Impulse Response Fit

Exp2-S3 Phase Response Fit
Exp2-S3 Amplitude Response Fit

Exp2-S3 Group Delay Fit

Exp2-S3 Low-Frequency Zoom

Exp2-S3 Phase Delay Fit
Exp2-S3 Impulse Response Fit

Exp2-S3 Slope Fit

Two Exponentials Followed by a 6th-Order IIR Filter Designed by Steiglitz McBride Algorithm (Exp2-SM6)
Exp2-SM6 Impulse Response Fit

Exp2-SM6 Slope Fit

Exp2-SM6 Phase Response Fit
Exp2-SM6 Amplitude Response Fit

Exp2-SM6 Group Delay Fit

Exp2-SM6 Low-Frequency Zoom

Exp2-SM6 Phase Delay Fit
Results for Measured Trumpet Data Using Two Offset Exponentials and Two Biquads

- Bell model filter complexity comparable to order 8+ IIR
- Offset exponentials were fit using `fmins()` in Matlab
- Two biquads were fit as a single fourth-order filter using the Steiglitz-McBride algorithm (`stmcb()` in Matlab)
Measured Trumpet Bell Amplitude Response

TIIR Trumpet Bell Amplitude Response

Trumpet Bell Phase Delay Fit

Input Impedance of Complete Bore + Bell Model
Comparison to Measurements

The next two pages of plots compare the measured impulse response with that produced by the final digital waveguide model consisting of a trumpet bore + bell (but no mouthpiece).

• Comparison 1: two offset exponentials and two biquads to model the bell impulse response
• Comparison 2: two offset exponentials and three biquads to model the bell impulse response
Measured Impulse Response

Synthesized Impulse Response, Order 6 Tail

Piecewise Conical Acoustic Tube Modeling

Simple Example: Cylinder with Conical Cap

Physical Outline of Cylinder and Cone:

Digital Waveguide Model (DWM) for Pressure Waves:

Reduced DWM for Maximum Computational Efficiency:

where

\[ R(z) = \left( \frac{1}{99} \right) \left( \frac{1 + z^{-1}}{1 - \frac{101}{99} z^{-1}} \right) \]

\[ T(z) = \left( \frac{100}{99} \right) \left( \frac{1 - z^{-1}}{1 - \frac{101}{99} z^{-1}} \right) = 1 + R(z) \]
• Problem: Reflection filter $R(z)$ and transmission filter $T(z)$ are unstable (pole at $z = 101/99$)

• Overall system is passive $\Rightarrow$ unstable pole is canceled

Implementation Idea

Apply TIIR “alternate and reset” idea to the unstable conical subsystem

• Cone is not truly FIR $\Rightarrow$ $t_{60}$ replaces FIR length

• When cylinder is closed-ended, cone traveling-wave components increase without bound $\Rightarrow$ must switch out and reset the entire cone assembly (scattering-junction filter $R(z)$ and cone’s entire delay line)

• According to simulations thus far, cylinder waves are well behaved and do not need to be reset (no general proof yet)

Basic Principle

Periodically reset any subsystem containing a canceled unstable pole at intervals greater than or equal to the $t_{60}$ for that subsystem

Interesting Paradox at DC

DC Steady State: Closed-End Cylinder

• $R(1) = -1$ (dc response of reflection filter inverts)

• $T(1) = 0$ (dc does not transmit through the junction)

• Physically obvious dc solution (constant pressure offset) is not possible in either the cone or the cylinder model!

• Simulated impulse responses agree with the literature

• A final constant dc offset is observed in the simulations
Solution to Paradox

- It turns out the reflection transfer function looking into the cone from the cylinder has two poles and two zeros at dc.
- The dc poles and zeros cancel and leave a dc cone reflectance equal to +1 (the physically obvious answer).
- We can’t just set the reflection filter to its dc equivalent to figure out the dc behavior of the overall model.
- Instead, a more careful limit must be taken.

In the $s$ plane, the conical cap pressure reflectance, seen from the cylinder, can be derived to be

$$H(s) \triangleq \frac{1 + R(s)(1 + 2st_x)}{2st_x - 1 - R(s)}$$

where $t_x$ is the time (in seconds) to propagate across the cone, and

$$R(s) = -e^{-2st_x}$$

is the reflectance of the cone at its entrance. We have

$$\lim_{s \to 0} R(s) = -1$$

$$\lim_{s \to 0} H(s) = +1$$

Truncated Infinite Impulse Response (TIIR) Digital Filters

An FIR filter can be constructed as the difference of two IIR filters:

**General FIR filter**

- Coefficients:  \{\(h_0, \ldots, h_N\)\}
- Implementation (convolution):

$$y(n) = (h * x)(n) = \sum_{m=0}^{N} h_m x(n - m)$$

- Transfer function:

$$H_{\text{FIR}}(z) \triangleq h_0 + h_1 z^{-1} + \ldots + h_N z^{-N} = z^{-N} C(z),$$
where \( C(z) \) is the \( N \)-th degree polynomial in \( z \) formed by the \( h_k \).

**General \( P \)-th order IIR filter**

- **Difference equation**
  \[
  y(n) = -\sum_{k=1}^{P} a_k y(n-k) + \sum_{\ell=0}^{P} b_\ell x(n-\ell)
  \]

- **Transfer function**
  \[
  H_{IIR}(z) = \frac{b_0 + b_1 z^{-1} + \ldots + b P z^{-P}}{1 + a_1 z^{-1} + \ldots + a P z^{-P}} = \frac{b_0 z^P + b_1 z^{P-1} + \ldots + b P}{z^P + a_1 z^{P-1} + \ldots + a P}
  \]

  \[
  \Delta = \frac{B(z)}{A(z)} = h_0 + h_1 z^{-1} + h_2 z^{-2} + \ldots,
  \]

  where
  \[
  A(z) \overset{\Delta}{=} z^P + a_1 z^{P-1} + \ldots + a P \quad \text{(monic)}
  \]
  \[
  B(z) \overset{\Delta}{=} b_0 z^P + b_1 z^{P-1} + \ldots + b P
  \]

**TIIR Construction: A One-Pole Example**

Consider an FIR filter having a truncated geometric sequence \( \{h_0, h_0 p, \ldots, h_0 p^N\} \) as an impulse response. This filter has the same impulse response for the first \( N + 1 \) terms as the one-pole IIR filter with transfer function

\[
H_{IIR}(z) = \frac{h_0}{1 - p z^{-1}}.
\]

Subtracting off the tail of the impulse response gives

\[
H_{FIR}(z) = h_0 + h_0 p z^{-1} + \ldots + h_0 p^N z^{-N} - \left\{ h_0 p^{N+1} z^{-(N+1)} + h_0 p^{(N+2)} z^{-(N+2)} + \ldots \right\}
\]

\[
= \frac{h_0}{1 - p z^{-1}} - p^{N+1} z^{-(N+1)} \frac{h_0}{1 - p z^{-1}}
\]

\[
= \frac{h_0}{1 - p^{N+1} z^{-(N+1)}}
\]

The time-domain recursion for this filter is

\[
y[n] = \sum_{k=0}^{N} h_0 p^k x[n-k] = p y[n-1] + h_0 \left( x[n] - p^{N+1} x[n - (N + 1)] \right)
\]
Complexity Notes

- Direct FIR filter implementation requires $N + 1$ multiplies and $N$ adds
- TIIR implementation requires 3 multiplies and 2 adds, independent of $N$
- No savings in memory

Note that there is a pole-zero cancellation in the TIIR transfer function

$$H(z) = h_0 \frac{1 - p^{N+1}z^{-(N+1)}}{1 - pz^{-1}} = h_0 + h_0pz^{-1} + \cdots + h_0p^Nz^{-N}$$

- If $|p| < 1$, no problem since the canceled pole is stable
- If $|p| \geq 1$, imperfect pole-zero cancellation due to numerical rounding leads to exponentially growing round-off error

Basic Idea: Since the overall TIIR filter is FIR(N), alternate between two instances of each unstable one-pole, starting each new one from the zero state $N$ samples before it is actually used. (Apparently first suggested by T. Fam at Asilomar-'87 for the case of distinct poles.)

Extension to Higher-Order TIIR Sequences

We can extend this idea from the one-pole case to any rational filter $H(z) = B(z)/A(z)$. The general procedure is to find the "tail filter" $H_{IIR}'(z)$ and subtract it off:

$$H_{FIR}(z) = H_{IIR}(z) - H_{IIR}'(z)$$

Multiply $H_{IIR}(z)$ by $z^N$ to obtain

$$z^NH_{IIR}(z) = h_0z^N + \cdots + h_{N-1}z + h_N + h_{N+1}z^{-1} + h_{N+2}z^{-2} + \cdots$$

$$\Delta = C(z) + H_{IIR}'(z)$$

$$\Delta = \frac{z^NB(z)}{A(z)}$$

- $B'(z)$ is the unique remainder after dividing $z^NB(z)$ by $A(z)$ using "synthetic division" ($z^NB(z) \equiv B'(z) \pmod{A(z)}$)
- We may assume $\text{Deg} \{B'(z)\} = \text{Deg} \{A(z)\} - 1$
- $B'(z)$ gives us our desired "tail filter" for forming $H_{FIR} = H_{IIR} - H_{IIR}'$:

$$H_{IIR}'(z) = \frac{B'(z)}{A(z)}$$
Higher-Order TIIR Filters

We have
\[ H_{\text{FIR}}(z) = H_{\text{IIR}}(z) - z^{-N} H'_{\text{IIR}}(z) \]
\[ = \frac{B(z) - z^{-N} B'(z)}{A(z)} \]

The corresponding difference equation is
\[ y[n] = -\sum_{k=1}^{P} a_k y[n - k] + \sum_{\ell=0}^{P} b_{\ell} x[n - \ell] \]
\[ - \sum_{m=0}^{P-1} b'_m x[n - m - (N + 1)] \]

Since the denominators of \( H_{\text{IIR}}(z) \) and \( H'_{\text{IIR}}(z) \) are the same, the dynamics (poles) can be shared:

**Complexity and Storage-Cost**

\[ H_{\text{FIR}}(z) = \frac{B(z) - z^{-N} B'(z)}{A(z)} \]
\[ N = \text{FIR order and let } P = A(z) \text{ order (\#poles)} \]

- The computational cost of the general truncated \( P \)-th order IIR system is \( 3P + 1 \) multiplies and \( 3P - 2 \) adds, independent of \( N \)
- Net computational savings is achieved when \( N > 3P \)

**Storage Requirements**

- \( P \) output samples for the IIR feedback dynamics \( A(z) \)
- \( N \) input samples of the FIR filter (main delay line)
- \( P \) input samples for \( B(z) \) (normally in delay line)
- \( P \) input samples for \( B'(z) \) (also possibly in delay line)

Thus, we need a total of at least \( N + P \) input delay samples, of which only \( 2P \) are accessed, and \( P \) output delay samples. This is between \( P \) and \( 2P \) more than a direct FIR implementation.
Example

We wish to truncate the impulse response of

\[ H^+(z) = \frac{B^+(z)}{A^+(z)} = \frac{1}{1 - 1.9z^{-1} + 0.98z^{-2}} \]

after \( N = 300 \) samples to obtain a length 301 FIR filter \( H_{FIR}^+(z) \)

Steps:

1. Perform synthetic division on \( z^{300}B^+(z) \) by \( A(z) \) to obtain the remainder

\[ B'^+(z) = -0.162126z + 0.139770 \]

2. Form the TIIR filter as

\[
H_{FIR}^+(z) = \sum_{k=0}^{N} h_k^+ z^{-k} = \frac{B^+(z) - z^{-N}B'^+(z)}{A^+(z)}
= \frac{1 + 0.162126z^{-299} - 0.139770z^{-300}}{1 - 1.9z^{-1} + 0.98z^{-2}}
\]
• At time $n = 301$, the tail of the response is subtracted off, and the impulse-response magnitude drops by about 115 dB.

• Due to quantization errors, there is a residual response.

• Poles are all stable, so error decays.
• Again, impulse-response tail is subtracted off at time $n = 301$, giving around 115 dB attenuation

• Additionally, state variables are cleared every 300 samples

• Residual response completely canceled at time $n = 600$

• System has truly finite memory

---

**Unstable Example**

To form a linear phase TIIR filter based on the previous example, we need also the “flipped” impulse response generated by

$$H_{\text{FIR}}(z) = \frac{-0.139770z^2 + 0.162126z - z^{-300}}{0.98z^2 - 1.9z + 1} = \frac{-0.142622z^2 + 0.165435z - 1.020408z^{-300}}{z^2 - 1.938776z + 1.020408}$$

where the last equation is normalized by 0.98 to make the denominator monic.

This system has two unstable hidden modes.
- Tail is canceled with about 125 dB attenuation
- Due to the unstable canceled poles, quantization noise grows without bound
- By time 1500 samples, the quantization noise dominates
- (Arithmetic = double-precision floating point with single-precision state variables)
Impulse Response of TIIR Implementation With Resets

- State-variable resets zero-out the quantization noise before it becomes significant
- Overall system has truly finite memory
### Synthetic Division Algorithm

Algorithm for performing synthetic division to generate the tail-canceling polynomial $B'(z)$:

```c
int i, j;
double *w = (double *)malloc((P + 1) * sizeof(double));
/*** load the numerator coefficients for B(z) ***/
for (i = 0; i < P + 1; i++) {
    w[i] = b[i];
}
/*** do synthetic division ***/
for (i = 0; i <= N; i++) {
    factor = w[0];
    for (j = 0; j < P; j++) {
        w[j] = w[j + 1] + factor * a[j];
    }
    w[P] = 0;
    /*** The remainder after the i-th step is in w[0..(P-1)] ***/
}
/*** copy the result to the output array ***/
for (i = 0; i < P; i++) {
    bb[i] = w[i];
}
```

### A One-Pole (Almost) TIIR Filter

- Generates truncated exponentials or constants
- Filter complexity on average $\approx$ one pole
- Shared delay line
- Shared dynamics
Offset Exponentials

Use two one-pole TIIRs, to make an offset exponential:

\[ h(n) = \begin{cases} \frac{a}{1 - z^{-1}} + b, & n = 0, 1, 2, \ldots, N - 1 \\ 0, & \text{otherwise} \end{cases} \]

- The constant portion \( b \) requires only one multiply (by \( b \)) since the pole for this TIIR filter is at \( z = 1 \)
- Resets for pure integrators are needed less often than for growing exponentials
- Using a cascade of digital integrators, any polynomial impulse response is possible
- A cubic-spline impulse response requires four integrators