Symbolic Computation in System Simulation and Design

Brian L. Evans

Dept. of Electrical Engineering and Computer Sciences

University of California, Berkeley, CA 94720-1770

`ble@eecs.berkeley.edu`

Ptolemy Mini-Conference, March 10, 1995
University of California, Berkeley, CA

— Introduction —

Role of Symbolic Computation

INPUT		**OUTPUT**
Numeric Signal	$\xrightarrow{\textit{Simulation}}$	Numeric Signal
Symbolic Description	$\xrightarrow[\textit{Transformation}]{}$	Symbolic Description

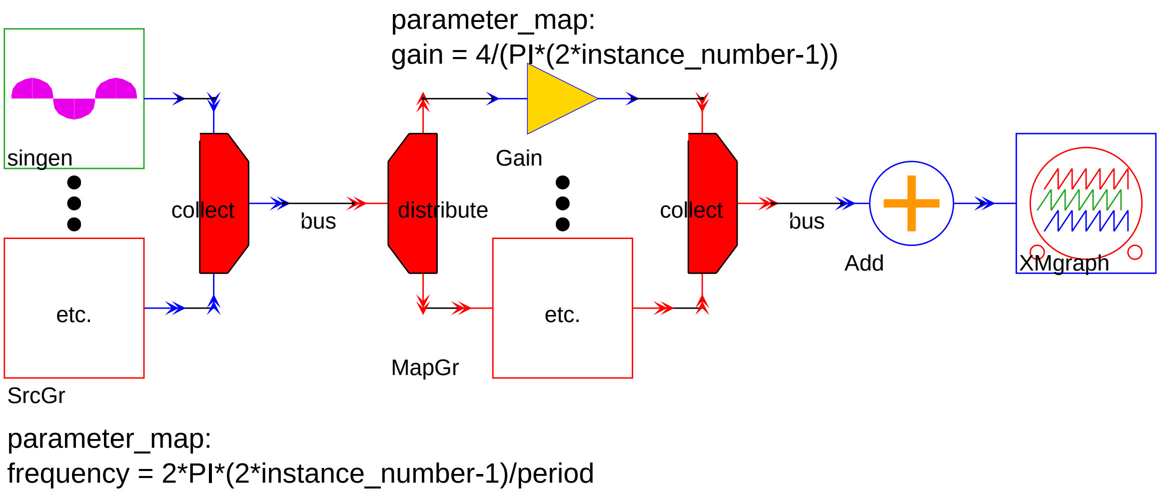
Symbolic Parameter Calculation

Truncated Fourier Series Computation

$$\hat{x}(t) \approx \sum_{m=-N}^{N} C_m \, e^{j2\pi \frac{m}{T} t} \qquad\qquad C_m = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} x(t) \, e^{j2\pi \frac{m}{T} t} \, dt$$

Fixed Fourier Coefficient Formula

Approximate a Square Wave by a
Finite Number of Sinusoids



Compute Fourier Coefficient Formula From $x(t)$

Numeric Parameter Optimization

Optimization of an Existing Filter Design

- Deviation from an ideal magnitude response
- Linear phase response in the passband
- Quality factors of second-order sections
- Peak overshoot in the step response

The Optimization Problem

- Sequential Quadratic Programming
- Differentiable objective functions
- Filter specifications to differentiable constraints

Numeric Parameter Optimization

Code generation

- Define the objective function and constraints
- Compute gradients of both symbolically
- Generate source code (C, Fortran, or Matlab)
- Generate main program (Matlab)

Numeric Parameter Optimization

Example: Fourth-Order All-Pole Filter

- Specifications:

  at $w_p = 20$ rad/sec, $\delta_p = 0.21$

  at $w_s = 30$ rad/sec, $\delta_s = 0.31$

- Initial filter is Butterworth

- Pole locations

initial:	-8.415 ± 20.315	-20.315 ± 8.415
final:	-7.792 ± 22.898	-19.562 ± 0.626

- Objective function

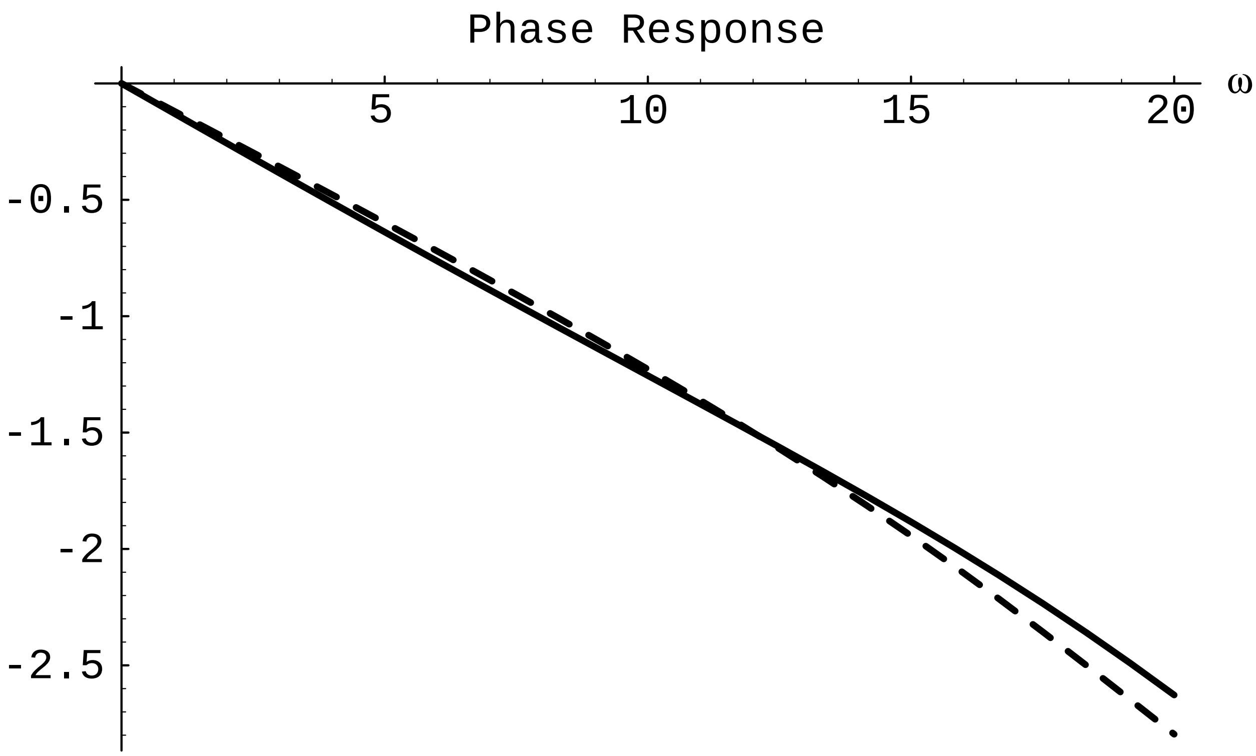
  reduced from 1.17 to 4.7×10^{-5}

- Final gradients of objective function

  3.1×10^{-5}, 4.2×10^{-5}, -2.3×10^{-5}, and -5.5×10^{-6}

Numeric Parameter Optimization

Example: Fourth-Order All-Pole Filter

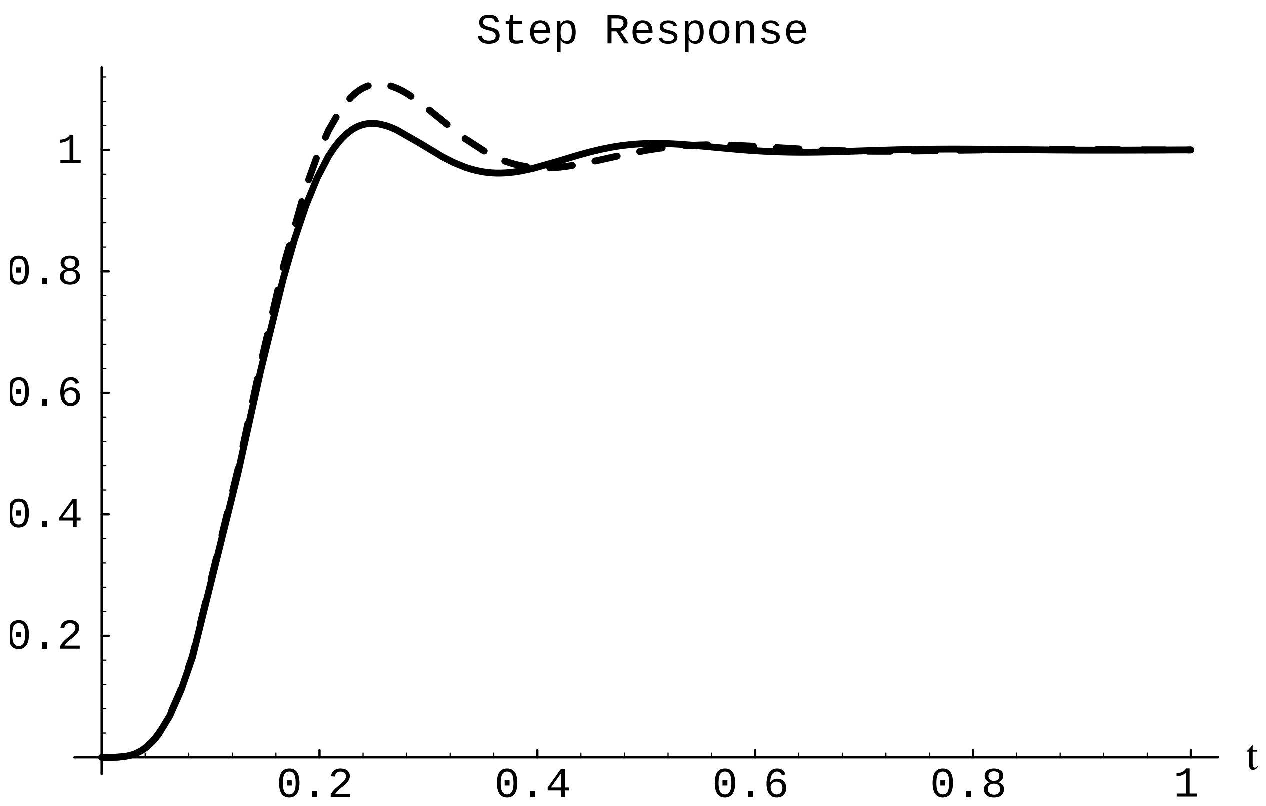


- - - dashed lines represent the initial Butterworth filter

— solid lines represent the filter optimized for linear phase response in the passband and for overshoot of the step response

Numeric Parameter Optimization

Example: Fourth-Order All-Pole Filter

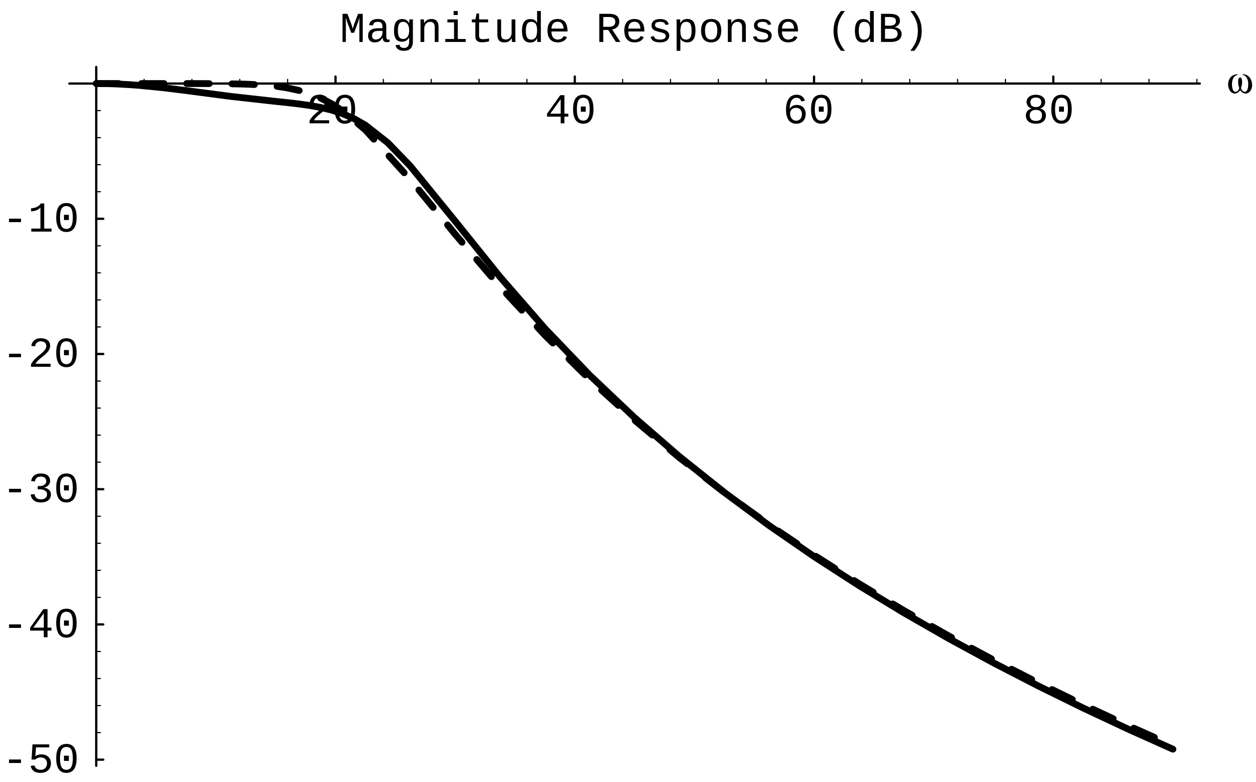


- - - dashed lines represent the initial Butterworth filter

— solid lines represent the filter optimized for linear phase response in the passband and for overshoot of the step response

Numeric Parameter Optimization

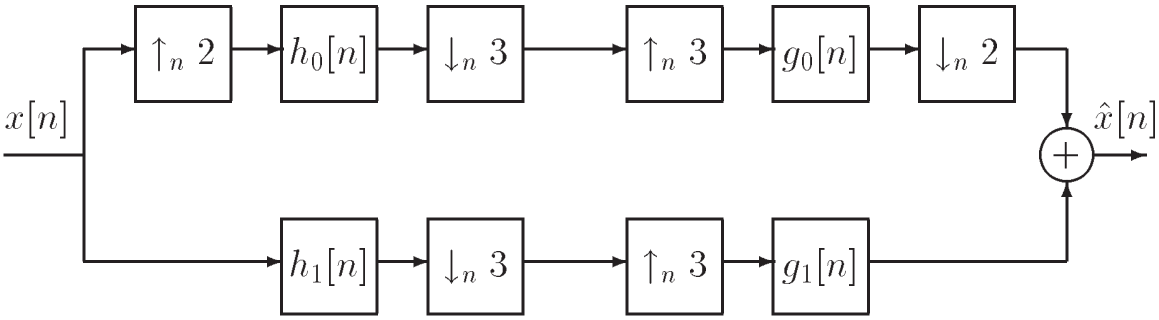
Example: Fourth-Order All-Pole Filter



--- dashed lines represent the initial Butterworth filter

— solid lines represent the filter optimized for linear phase response in the passband and for overshoot of the step response

Non-Uniform Filter Bank



Flow graph of a two-channel non-uniform filter bank

```
upperchannel =
  Downsample[2,n][
    Convolve[n][
      g0[n],
      Upsample[3,n][
        Downsample[3,n][
           Convolve[n][h0[n],
                       Upsample[2,n][x[n]]]]]]]
lowerchannel =
  Convolve[n][
    g1[n],
    Upsample[3,n][
      Downsample[3,n][ Convolve[n][h1[n], x[n]]]]]
```

Algebraic description of the filter bank

— System Design —

Non-Uniform Filter Bank

$$\hat{X}(z) = \tfrac{1}{6}(G_0(-\sqrt{z})H_0(-\sqrt{z}) + G_0(\sqrt{z})H_0(\sqrt{z}) + 2G_1(z)H_1(z))\,X(z) +$$
$$\tfrac{1}{6}\Big(G_0(-\sqrt{z})H_0(e^{\frac{i}{3}\pi}\sqrt{z}) + G_0(\sqrt{z})H_0(e^{\frac{4i}{3}\pi}\sqrt{z}) + 2G_1(z)H_1(e^{\frac{2i}{3}\pi}z)\Big)\,X(e^{\frac{2i}{3}\pi}z) +$$
$$\tfrac{1}{6}\Big(G_0(\sqrt{z})H_0(e^{\frac{2i}{3}\pi}\sqrt{z}) + G_0(-\sqrt{z})H_0(e^{\frac{5i}{3}\pi}\sqrt{z}) + 2G_1(z)H_1(e^{\frac{4i}{3}\pi}z)\Big)\,X(e^{\frac{4i}{3}\pi}z)$$

Symbolic analysis of input-output relationship

```
h0[n] = FIR[n, Hold[ReadList["ptolemy/h0", Number]]];
h1[n] = FIR[n, Hold[ReadList["ptolemy/h1", Number]]];

g0[n] = FIR[n, Hold[ReadList["ptolemy/g0", Number]]];
g1[n] = FIR[n, Hold[ReadList["ptolemy/g1", Number]]];

x[n] = Cos[2 Pi n / 3] Sinc[Pi n / 6] / 3;

PtolemySimulation[ upperchannel + lowerchannel,
                   {n, 1, 100} ] >> "!ptcl"
```

Transformation of algebraic description to Ptcl

Evaluating Alternate Implementations

Rearrangement Rules

- Rules based on interaction between operators
- Based on properties of signals and systems

Cost Functions

- Based on implementation costs
- Require feedback from synthesis tools

Heuristic Searches

- Search through space of alternate implementations

Multidimensional Signal Processing

Multidimensional Signals Defined On Grid of Points

Multidimensional Periodic Signals

$$x[\mathbf{n}] = x[\mathbf{n} + \mathbf{N}\,\mathbf{r}]$$

General Multidimensional DFT

$$X[\mathbf{k}] = \sum_{\mathbf{n}} x[\mathbf{n}] e^{-j\,2\,\pi\,\mathbf{k}^T\,\mathbf{N}^{-1}\,\mathbf{n}}$$

Smith Form Decompositions

$$\mathbf{N} = \mathbf{U}\,\mathbf{\Lambda}\,\mathbf{V} \implies \mathbf{N}^{-1} = \mathbf{V}^{-1}\,\mathbf{\Lambda}^{-1}\,\mathbf{U}^{-1}$$

$$\begin{aligned} X[\mathbf{k}] &= \sum_{\mathbf{n}} x[\mathbf{n}] e^{-j\,2\,\pi\,\left(\mathbf{k}^T V^{-1}\right)\mathbf{\Lambda}^{-1}\left(U^{-1}\mathbf{n}\right)} \\ X[\hat{\mathbf{k}}] &= \sum_{\hat{\mathbf{n}}} x[\mathbf{U}\,\hat{\mathbf{n}}] e^{-j\,2\,\pi\,\hat{\mathbf{k}}^T\,\mathbf{\Lambda}^{-1}\,\hat{\mathbf{n}}} \end{aligned}$$

Multidimensional Rearrangement Rules

$$\rightarrow \boxed{\downarrow U_M} \rightarrow \boxed{\downarrow \Lambda_M} \rightarrow \boxed{\downarrow V_M} \rightarrow \boxed{\uparrow V_L} \rightarrow \boxed{\uparrow \Lambda_L} \rightarrow \boxed{\uparrow U_L} \rightarrow$$

Cascade in Smith Form

$$\rightarrow \boxed{\downarrow U_M} \rightarrow \boxed{\downarrow \Lambda_M} \rightarrow \boxed{\uparrow \Lambda_L} \rightarrow \boxed{\uparrow U_L} \rightarrow$$

Simplified cascade *if* $V_M = V_L$

$$\rightarrow \boxed{\downarrow U_M} \rightarrow \boxed{\uparrow \Lambda_L} \rightarrow \boxed{\downarrow \Lambda_M} \rightarrow \boxed{\uparrow U_L} \rightarrow$$

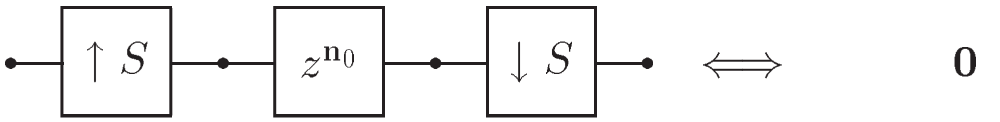
Reversing order of operations *if* Λ_M and Λ_L are coprime

$$\rightarrow \boxed{\uparrow \Lambda_L U_M^{-1}} \rightarrow \boxed{\downarrow \Lambda_M U_L^{-1}} \rightarrow$$

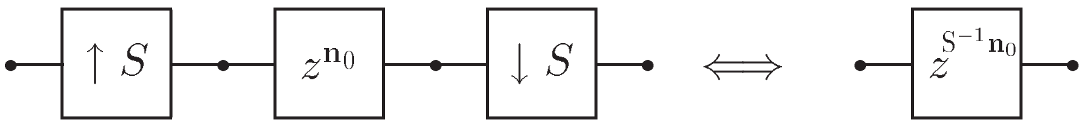
Combining operations

Four Equivalent Forms of a Downsampler and Upsampler in Cascade

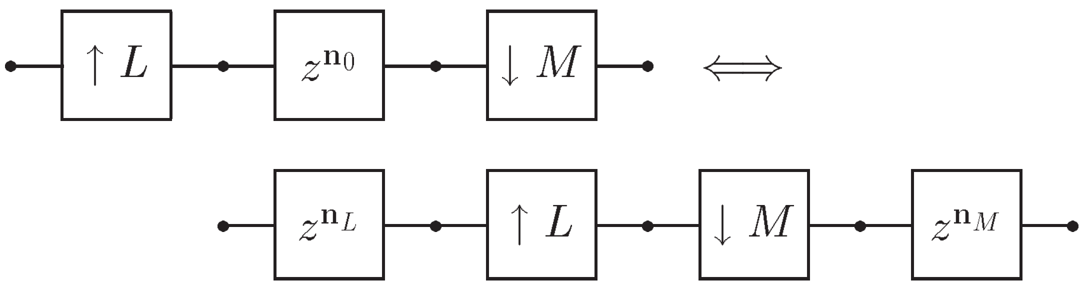
Multidimensional Rearrangement Rules



Up/downsampling by S when the shift vector
$\mathbf{n}_0 \notin$ sublattice(S); i.e., $S^{-1}\mathbf{n}_0$ is not an integer vector



Up/downsampling by S when the shift vector
$\mathbf{n}_0 \in$ sublattice(S); i.e., $S^{-1}\mathbf{n}_0$ is an integer vector



For any L and M, $\mathbf{n}_0$ can be rewritten as $\mathbf{n}_0 = L\mathbf{n}_L + M\mathbf{n}_M$

Interaction between Upsamplers,
Shifters, and Downsamplers in Cascade

Conclusion

System Simulation

- Symbolic parameter calculation
- Numeric parameter optimization

System Design

- Symbolic analysis and transformation
- Evaluating alternative implementations

Future Work

- Allow parameters to calculated symbolically
- Explore optimization of other behavioral models
- Encode Synchronous Dataflow (SDF) system rewriting in the Design Methodology Management (DMM) Domain
- Implement Multidimensional SDF system rewriting in DMM Domain, esp. non-separable resampling operations