Multi-Criteria Analog IIR Filter Optimization

Prof. Brian L. Evans

http://www.ece.utexas.edu/~bevans

work performed with

Niranjan Damera-Venkata

Electrical and Computer Engineering Department
The University of Texas at Austin
Austin, TX 78712-1084

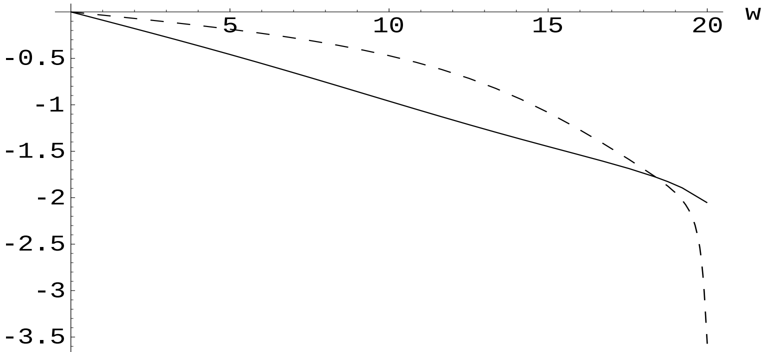
http://anchovy.ece.utexas.edu

Outline

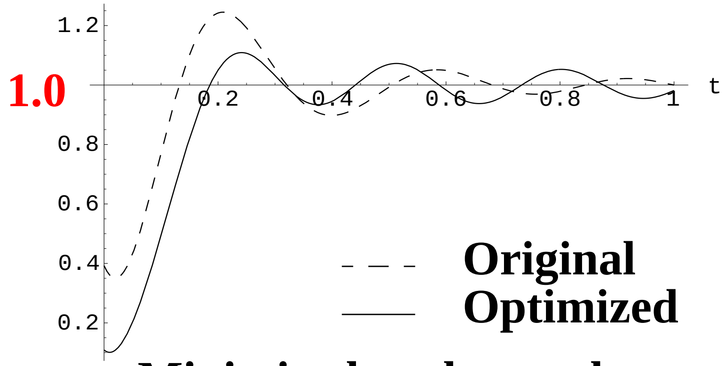
- **Introduction**
- **Modeling**
- **Objective Measures for Properties**
- **Distance Measures for Properties**
- **Design Example**
- **Validation**
- **Conclusion**

Introduction

- ***Problem*: Optimize multiple analog filter behavioral and implementation characteristics at the same time**
- ***Goal*: Develop an extensible, automated framework**
- ***Solution*: Filter Optimization Packages for Mathematica**
  - **Constrained non-linear optimization as Sequential Quadratic Programming: converges to global optimum & robust when closed-form gradients provided.**
  - **Program Mathematica to derive formulas for cost function, constraints, and gradients, and convert the formulas to Matlab programs to run optimization.**
  - **Example: linearize phase and minimize peak overshoot of an elliptic filter; constraining Q_{max} to 10 reduced Q_{max} from 61 to 10 (filter easier to build)**



Linearized phase in passband



Minimized peak overshoot

http://www.ece.utexas.edu/~bevans/projects/syn_filter_software.html

Modeling

- **Free Parameters: Locations of Poles and Zeros**
  - **List of *n* conjugate pole-pairs $a_k \pm j\, b_k$ (*and multiplicities*)**
  - **List of *r* conjugate zero-pairs $c_k \pm j\, d_k$ (*and multiplicities*)**
  - **Can be combined into a cascade of second-order sections**
- **Properties**
  - ***Behavioral*: magnitude, phase, and step responses**
  - ***Implementation*: quality factors**
  - **All properties are real-valued**
- **Formulate Optimization Problem**
  - ***Objective measures* of the properties as functions of the free parameters**
  - ***Distance measures* for deviation of the actual and desired property values**
  - ***Cost function* as a weighted combination of distance measures**
  - ***Constraints* on the values of the free parameters**

Objective Measures for Properties

Objective Measures for the All-Pole IIR Filter Case

- **Magnitude response (*with polynomials in Horner's form*)**

$$|H(j\omega)| = \prod_{k=1}^{n} \frac{a_k^2 + b_k^2}{\sqrt{(\omega^2 + 2(a_k^2 - b_k^2))\omega^2 + (a_k^2 + b_k^2)^2}}$$

- **Unwrapped phase response**

$$\angle H(j\omega) = \sum_{k=1}^{n} \left(\operatorname{atan}\frac{\omega - b_k}{a_k} + \operatorname{atan}\frac{\omega + b_k}{a_k} \right)$$

- **Quality factors**
  - **For *k*th second-order section, use standard formula**
  - **Q_k >= 0.5, where Q_k = 0.5 corresponds to a real pole and Q_k of infinity corresponds to an imaginary pole**
  - **Effective quality factor Q_{eff} is a combination of the second-order quality factors: we chose the *geometric mean***

$$Q_k = \frac{\sqrt{a_k^2 + b_k^2}}{-2a_k}$$

Objective Measures for Properties

Measuring Peak Overshoot in the Step Response

- **Partial fractions decomposition**

$$\frac{H(s)}{s} = \frac{1}{s}\sum_{k=1}^{n} \frac{C_k s + D_k}{s^2 - 2a_k s + a_k^2 + b_k^2}$$

- **Step response, where** $y_k = C_k\,(a_k^2 + b_k^2)\,/\,D_k$**,**

$$h_{step}(t) = \sum_{k=1}^{n} \frac{D_k}{a_k^2 + b_k^2}\left(1 - e^{a_k t}\left(\cos(b_k t) - \frac{a_k + \gamma_k}{b_k}\sin(b_k t)\right)\right)$$

- **Time when peak overshoot occurs in each section**

$$t_{peak}^{k} = -\frac{1}{b_k}\left(\operatorname{atan}\left(\frac{\gamma_k b_k}{a_k^2 + \gamma_k a_k + b_k^2}\right) + \pi\right)$$

  - **For computing gradients only, approximate peak overshoot time as a constant times the average of second-order peak overshoot times**

Distance Measures for Properties

- **Deviation in Magnitude Response**
  - **Euclidean distance over each passband, stopband, and transition band**
- **Deviation from Linear Phase Response**
  - **Measure deviation from linear phase over the passband**

$$\sigma_{phase} = \int_{\omega_2}^{\omega_1} (\angle H(j\omega) - m_{lp}\omega)^2 \; d\omega$$

  - **Optimal slope of the phase, m_{lp}, is a function of the passband interval (ω_1, ω_2) as well as the pole and zero pairs**

$$m_{lp} = \frac{\int_{\omega_2}^{\omega_1} \angle H(j\omega)\omega d\omega}{\int_{\omega_2}^{\omega_1} \omega^2 d\omega} = \frac{3}{2(\omega_2^3 - \omega_1^3)} \sum_{k=1}^{n} (f_k(\omega_1) - f_k(\omega_2))$$

  - **Calculated by computer algebra software**
- **Deviation for Peak Overshoot: $(h_{step}(t_{peak}) - 1)^2$**
- **Deviation for Quality Factors: $Q_{eff} - 0.5$**

Design Example

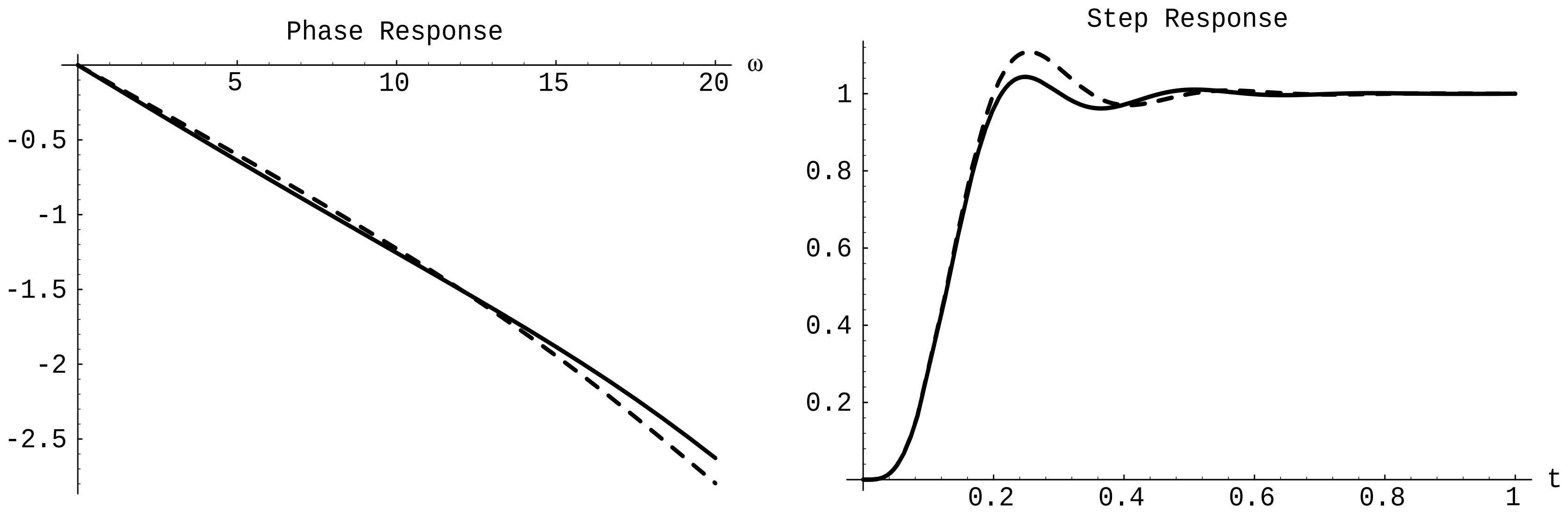
Optimizing for Linear Phase and Peak Overshoot

Table 1: Fourth-Order Lowpass Filter

	Initial	Final
Pole Pair 1	- 8.4149 ± *j* 20.3153	-7.7918 ± *j* 22.8984
Pole Pair 2	-20.3153 ± j 8.4149	-19.5623 ± *j* 0.6255
Cost Function	1.17	0.000047
Peak Overshoot	0.16%	0.08%

- ***initial value*****: fourth-order lowpass Butterworth filter**
- ***final value*****: a hybrid filter**
- **phase response in passband became *nearly linear***
- **one second-order section more sensitive to perturbations**
- **quality factors: {0.541, 1.31} -> {0.500, 1.55}**

Design Example (continued)

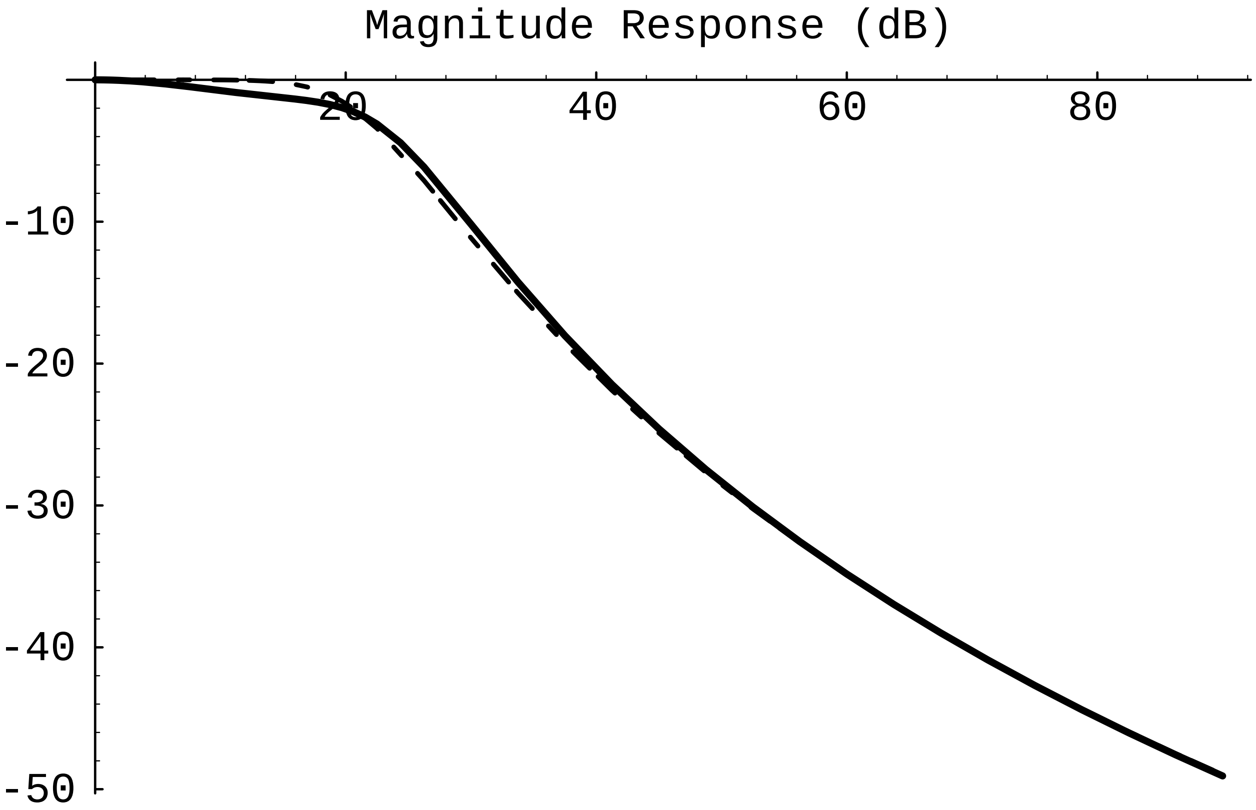


---- initial filter

—— optimized filter

Design Example (continued)

Trade-off Magnitude Response for Step and Phase Response



---- initial filter

—— optimized filter

Validation of the Automated Framework

Algebraic Verification of Formulas

- **Formula for partial fractions decomposition**
- **Formula for step response**
- **Optimal slope for linear phase**

Numerical Validation of Formulas

- **Magnitude and phase formulas**

Validation of Synthesized Code

- **Plot Mathematica and MATLAB formulas**
- **SQP MATLAB *constr* routine checks symbolic gradients**

Conclusion

Automating the Solution

- Enter objective measures, distance measures, and constraints in computer algebra environment
- Choose an optimization technique
- Transform optimization problem to fit the technique
- Synthesize transformed problem into software
- Export solution to a system-level design environment

Advantages

- Abstract design specification to a higher level
- Avoid errors in performing algebra and calculus
- Avoid errors in converting equations to source code