Optimal Design of Discrete-Time $\Delta\Sigma$ Modulators

Matthew E. Jackson

Advisor: Peter A. Stubberud, Ph.D.

University of Nevada - Las Vegas

Department of Electrical and Computer Engineering

4505 Maryland Parkway

Las Vegas, Nevada, USA

Overview

- Introduction
  - Motivation: Optimal Design of $\Delta\Sigma$ Modulators
  - Rationale: Global Optimization of Multimodal Performance Surfaces which Characterize Optimal IIR Filters
  - Genetic Algorithms: Structure and Operational Theory
- Hybrid Orthogonal Genetic (HOG) Algorithm
  - Operational Theory and Implementation
  - *Taguchi Method* and *Design of Experiments*
  - Global Optimizer Application Examples
- Optimal Discrete-Time $\Delta\Sigma$ Modulator Design and Results
  - HOG Algorithm Implementation
  - $\Delta\Sigma$ Modulator Modeling, Simulation, and Analysis
  - Results
- Conclusion

Overview

- Introduction
  - Motivation: Optimal Design of $\Delta\Sigma$ Modulators
  - Rationale: Global Optimization of Multimodal Performance Surfaces which Characterize Optimal IIR Filters
  - Genetic Algorithms: Structure and Operational Theory
- Hybrid Orthogonal Genetic (HOG) Algorithm
  - Operational Theory and Implementation
  - *Taguchi Method* and *Design of Experiments*
  - Global Optimizer Application Examples
- Optimal Discrete-Time $\Delta\Sigma$ Modulator Design and Results
  - HOG Algorithm Implementation
  - $\Delta\Sigma$ Modulator Modeling, Simulation, and Analysis
  - Results
- Conclusion

Overview

- Introduction
  - Motivation: Optimal Design of $\Delta\Sigma$ Modulators
  - Rationale: Global Optimization of Multimodal Performance Surfaces which Characterize Optimal IIR Filters
  - Genetic Algorithms: Structure and Operational Theory
- Hybrid Orthogonal Genetic (HOG) Algorithm
  - Operational Theory and Implementation
  - *Taguchi Method* and *Design of Experiments*
  - Global Optimizer Application Examples
- Optimal Discrete-Time $\Delta\Sigma$ Modulator Design and Results
  - HOG Algorithm Implementation
  - $\Delta\Sigma$ Modulator Modeling, Simulation, and Analysis
  - Results
- Conclusion

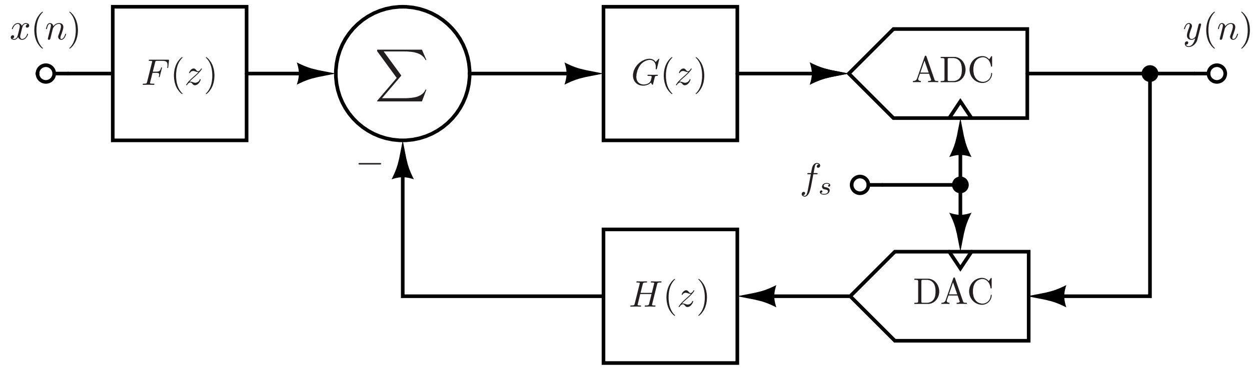
Overview

- Introduction
  - Motivation: Optimal Design of $\Delta\Sigma$ Modulators
  - Rationale: Global Optimization of Multimodal Performance Surfaces which Characterize Optimal IIR Filters
  - Genetic Algorithms: Structure and Operational Theory
- Hybrid Orthogonal Genetic (HOG) Algorithm
  - Operational Theory and Implementation
  - *Taguchi Method* and *Design of Experiments*
  - Global Optimizer Application Examples
- Optimal Discrete-Time $\Delta\Sigma$ Modulator Design and Results
  - HOG Algorithm Implementation
  - $\Delta\Sigma$ Modulator Modeling, Simulation, and Analysis
  - Results
- Conclusion

Introduction

Motivation

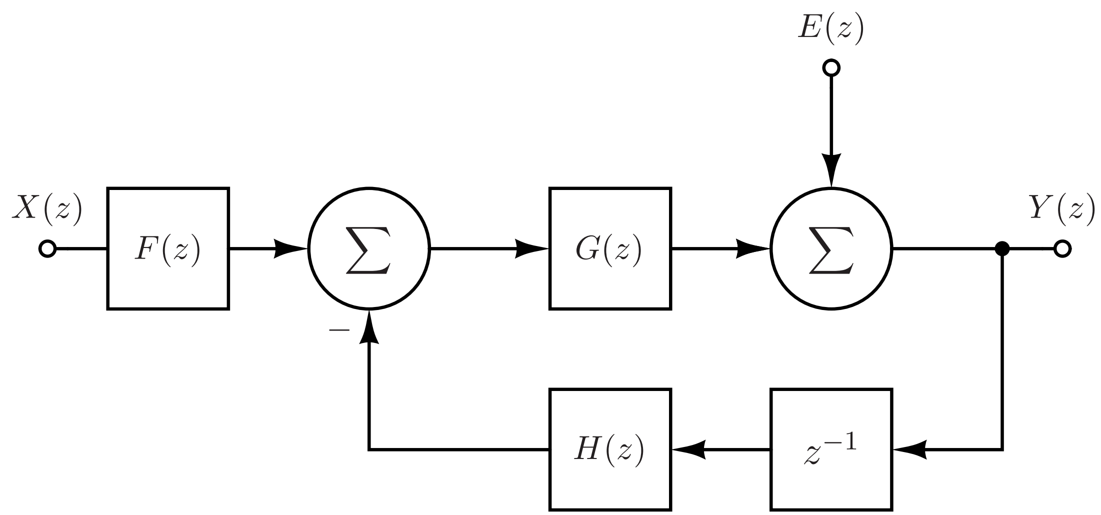
Optimal Design of $\Delta\Sigma$ Modulator Data Converters



- Simple analog circuitry and low order quantizer simplifies hardware requirements
- Utilizes oversampling and noise-shaping to achieve high Signal-to-Noise Ratio (SNR) and improved Dynamic Range (DR)

Motivation

Optimal Design of $\Delta\Sigma$ Modulator Data Converters



$$Y(z) = \text{STF}(z)X(z) + \text{NTF}(z)E(z)$$

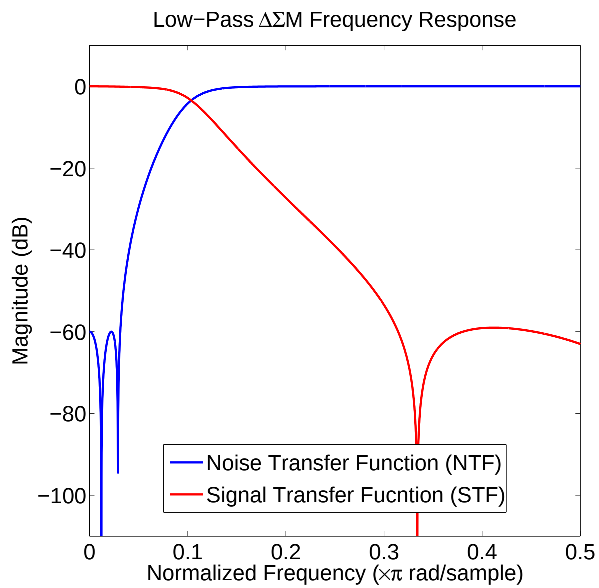
$$\text{STF}(z) = \frac{F(z)G(z)}{1 + z^{-1}G(z)H(z)} \quad \text{NTF}(z) = \frac{1}{1 + z^{-1}G(z)H(z)}$$

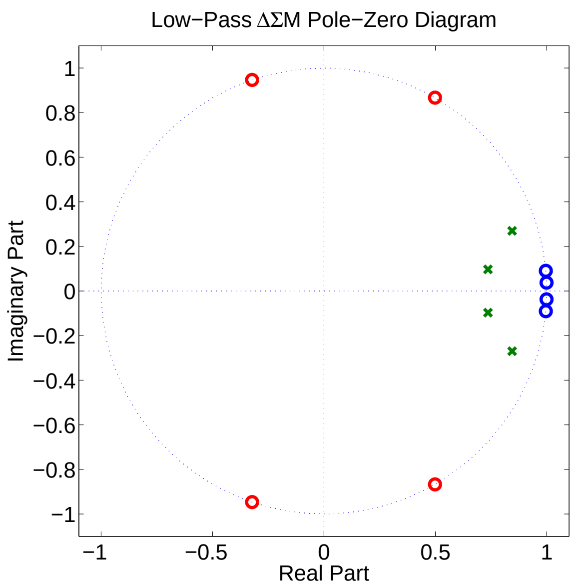
- Linearly modeled as a coupled set of complex transfer functions referred to as the Noise Transfer Function (NTF) and the Signal Transfer Function (STF)

Motivation

Introduction
❖ Motivation
❖ Rationale
❖ Objective Functions
HOG Algorithm
HOG Algorithm Examples
DSM Design and Results
Conclusions

Optimal Design of $\Delta\Sigma$ Modulator Data Converters





- Designed as a complimentary set of system functions such that the STF and NTF share common poles
- Classical Design Techniques: Chebyshev, Bessel, etc.
- EDA Based Methods: MATLAB DelSig Toolbox (R. Schreier)

Rationale

- **Problem Description:**
  Optimal Design of Infinite Impulse Response (IIR) Filters
  - Minimize cost functions representing optimal filter performance
  - Topic of interest since the late 1950s (Linear Programming)
  - Multimodal performance surface
  - Highly non-linear and non-differentiable
- **Solution Techniques:** Constrained Global Optimization
  - Linear / Non-Linear programming techniques
  - Directed search algorithms
  - Traditional genetic algorithms (GAs) and evolutionary strategies (ESs)
  - Memetic GAs which employ second level learning to adaptively modify the algorithm parameters
- **Proposed Solution:** Hybrid Orthogonal Genetic (HOG) Algorithm
  - Numerically reasonable global optimizer
  - Robust for complex performance surfaces
  - Flexible and easily adaptable for differing cost functions

Rationale

- **Problem Description:**
  Optimal Design of Infinite Impulse Response (IIR) Filters
  - Minimize cost functions representing optimal filter performance
  - Topic of interest since the late 1950s (Linear Programming)
  - Multimodal performance surface
  - Highly non-linear and non-differentiable
- **Solution Techniques:** Constrained Global Optimization
  - Linear / Non-Linear programming techniques
  - Directed search algorithms
  - Traditional genetic algorithms (GAs) and evolutionary strategies (ESs)
  - Memetic GAs which employ second level learning to adaptively modify the algorithm parameters
- **Proposed Solution:** Hybrid Orthogonal Genetic (HOG) Algorithm
  - Numerically reasonable global optimizer
  - Robust for complex performance surfaces
  - Flexible and easily adaptable for differing cost functions

Rationale

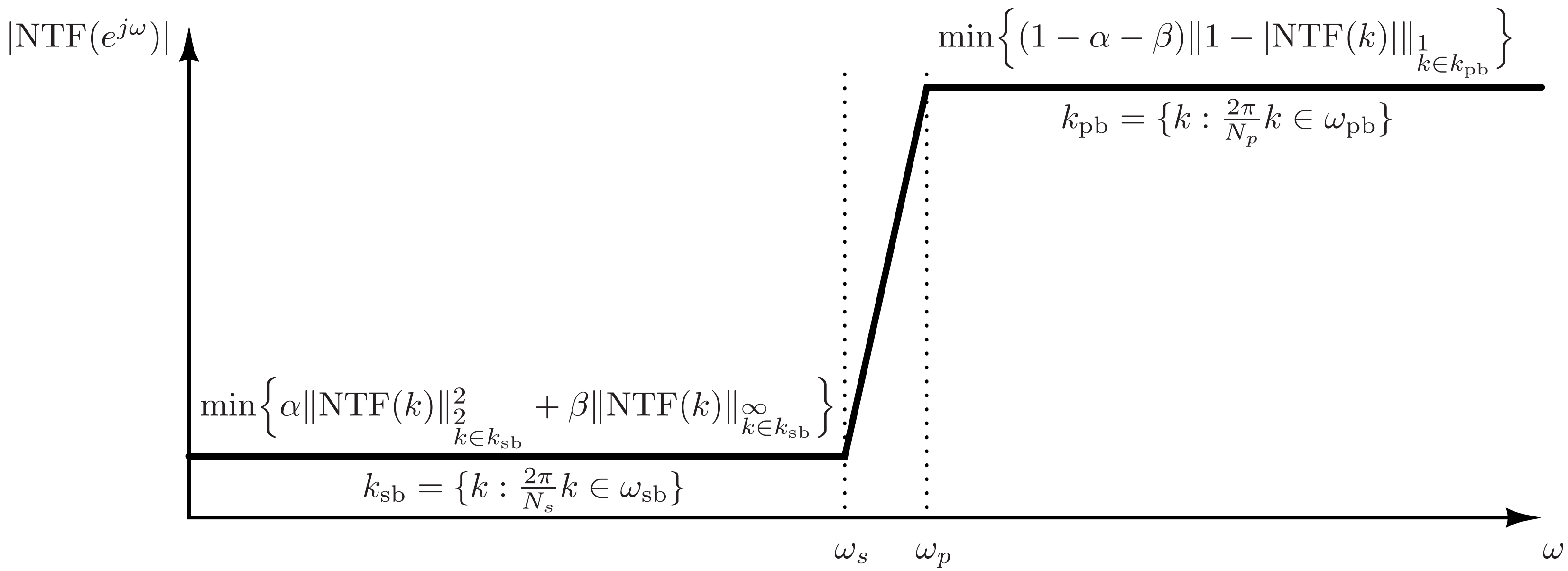
- **Problem Description:**
  Optimal Design of Infinite Impulse Response (IIR) Filters
  - Minimize cost functions representing optimal filter performance
  - Topic of interest since the late 1950s (Linear Programming)
  - Multimodal performance surface
  - Highly non-linear and non-differentiable
- **Solution Techniques:** Constrained Global Optimization
  - Linear / Non-Linear programming techniques
  - Directed search algorithms
  - Traditional genetic algorithms (GAs) and evolutionary strategies (ESs)
  - Memetic GAs which employ second level learning to adaptively modify the algorithm parameters
- **Proposed Solution:** Hybrid Orthogonal Genetic (HOG) Algorithm
  - Numerically reasonable global optimizer
  - Robust for complex performance surfaces
  - Flexible and easily adaptable for differing cost functions

$\Delta\Sigma$ *Modulator Design Objective Functions*

Noise Transfer Function Objective Function

$$J_{\text{NTF}} = \alpha \|\text{NTF}(k)\|^2_{2_{k\in k_{\text{sb}}}} + \beta \|\text{NTF}(k)\|_{\infty_{k\in k_{\text{sb}}}} + (1-\alpha-\beta)\|1-|\text{NTF}(k)|\|_{1_{k\in k_{\text{pb}}}}$$

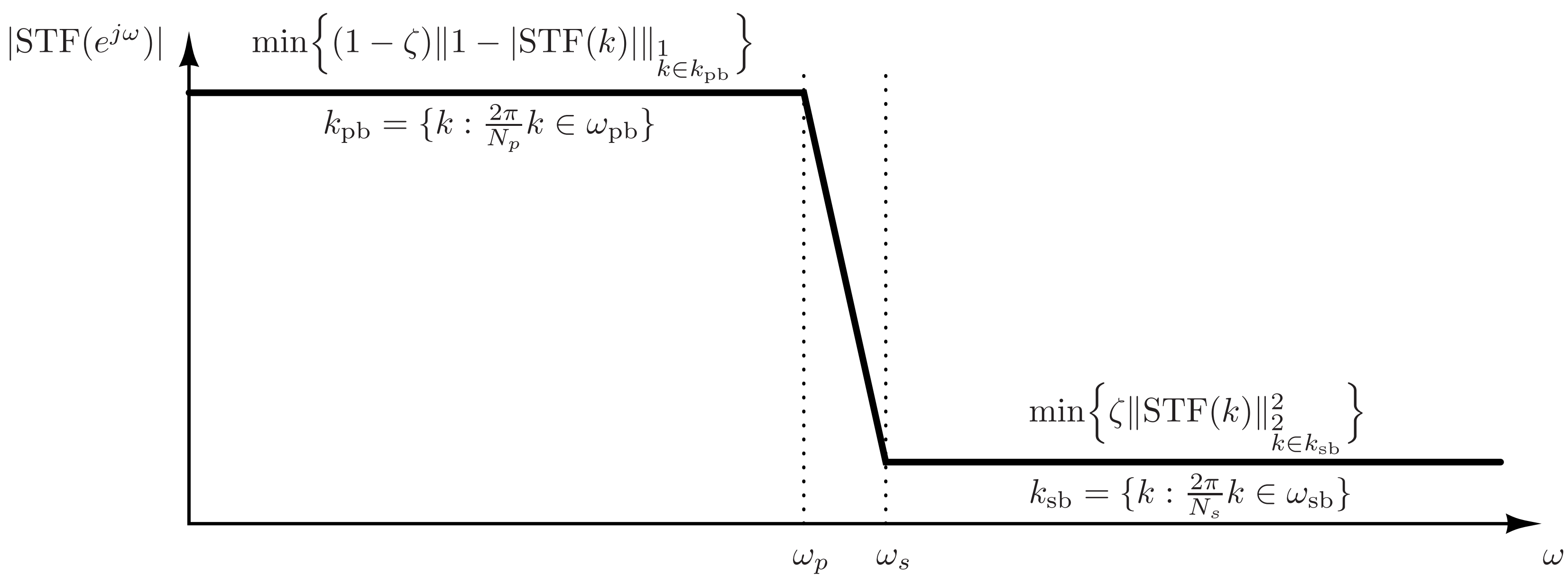
- Approximated by taking the DFT of the NTF frequency response
- Stopband error minimized wrt weighted combination of SNR and DR
- Passband error minimized to eliminate peaking



Signal Transfer Function Objective Function

$$J_{\text{STF}} = \zeta \|\text{STF}(k)\|^2_{2_{k\in k_{\text{sb}}}} + (1-\zeta)\|1 - |\text{STF}(k)|\|_{1_{k\in k_{\text{pb}}}}$$

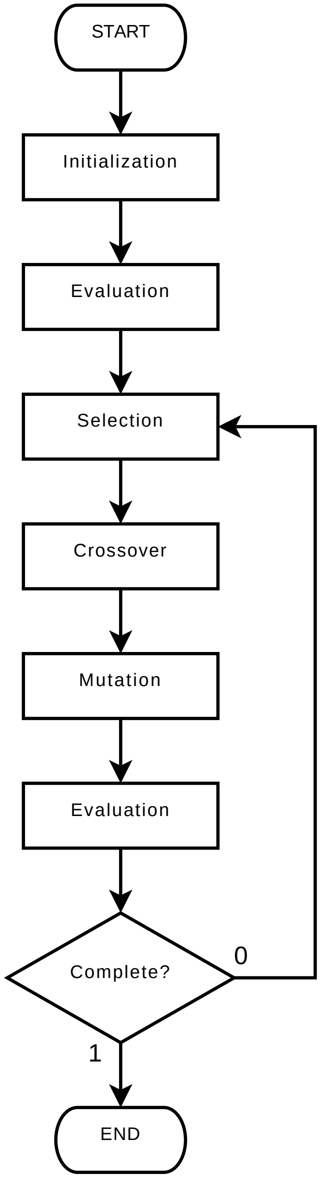
- Approximated by taking the DFT of the STF frequency response
- Stopband error minimized to minimize out of band signal energy
- Passband error minimized to eliminate peaking



Hybrid Orthogonal Genetic (HOG) Algorithm

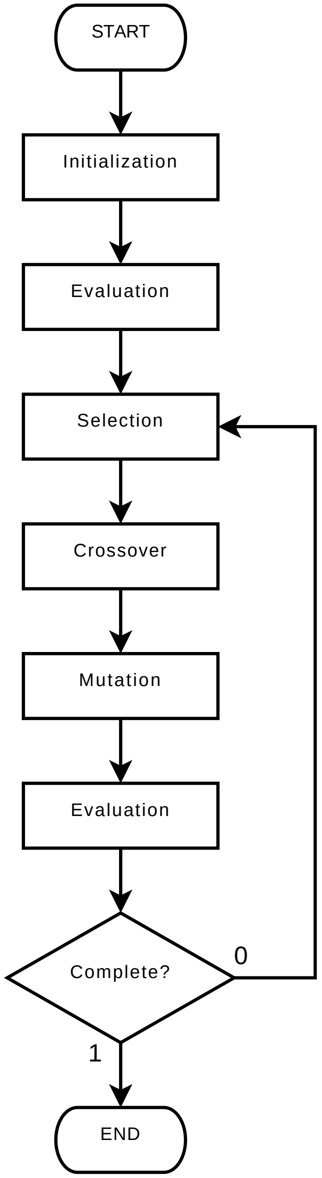
Genetic Algorithms: Overview

- General Theory
- Population Initialization
- Fitness Evaluation
- Progenitor Selection
- Reproduction
- Genetic Diversity
- Convergence



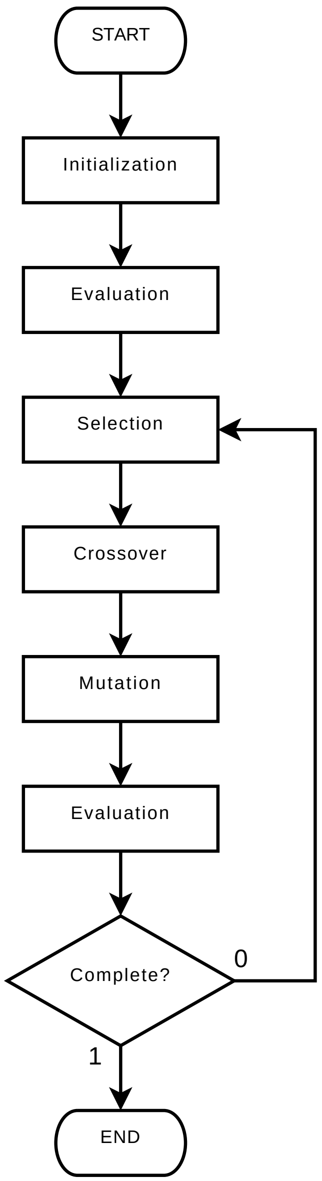
Genetic Algorithms: Overview

- General Theory
- Population Initialization
- Fitness Evaluation
- Progenitor Selection
- Reproduction
- Genetic Diversity
- Convergence



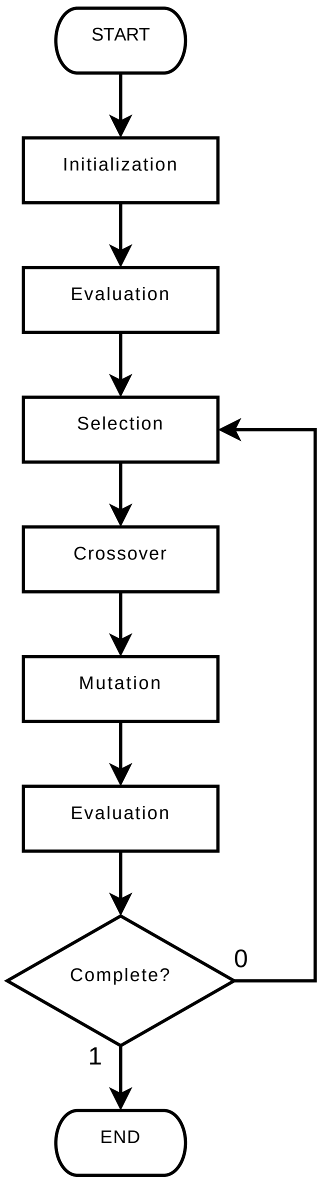
Genetic Algorithms: Overview

- General Theory
- Population Initialization
- Fitness Evaluation
- Progenitor Selection
- Reproduction
- Genetic Diversity
- Convergence



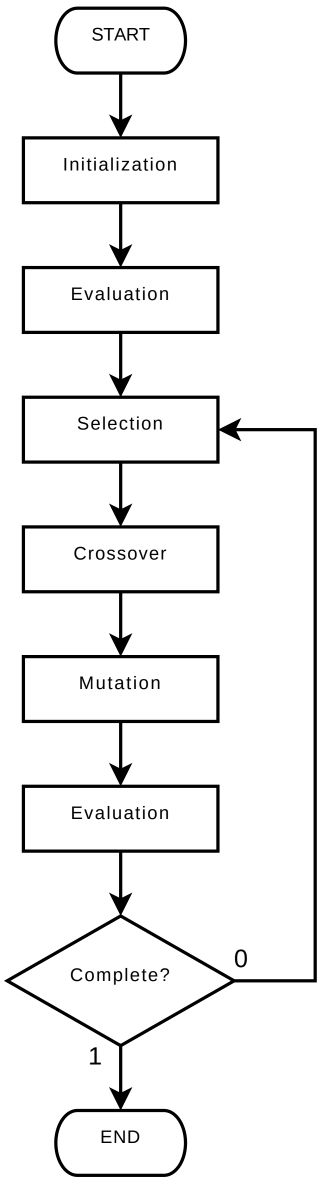
Genetic Algorithms: Overview

- General Theory
- Population Initialization
- Fitness Evaluation
- Progenitor Selection
- Reproduction
- Genetic Diversity
- Convergence



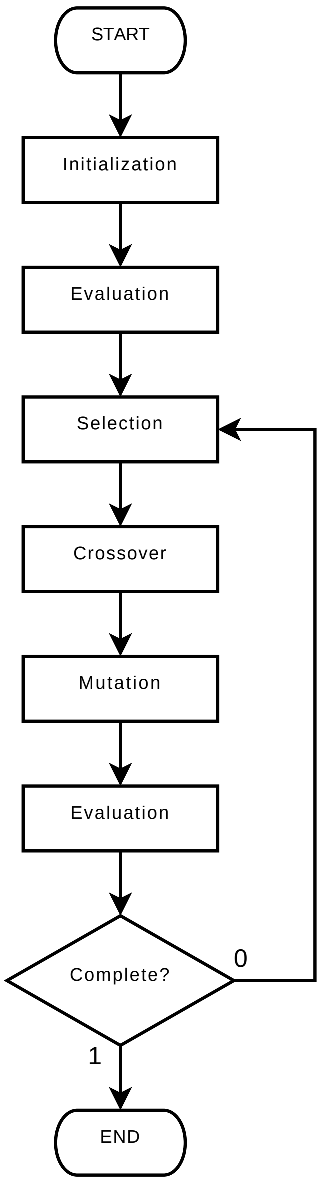
Genetic Algorithms: Overview

- General Theory
- Population Initialization
- Fitness Evaluation
- Progenitor Selection
- Reproduction
- Genetic Diversity
- Convergence



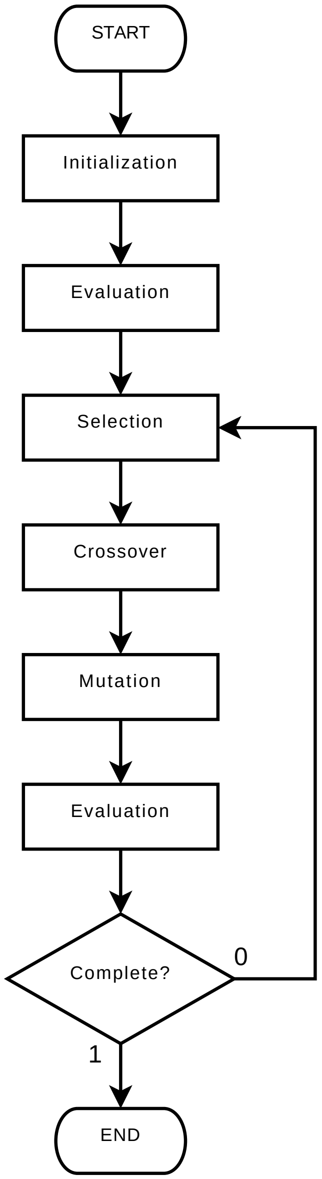
Genetic Algorithms: Overview

- General Theory
- Population Initialization
- Fitness Evaluation
- Progenitor Selection
- Reproduction
- Genetic Diversity
- Convergence



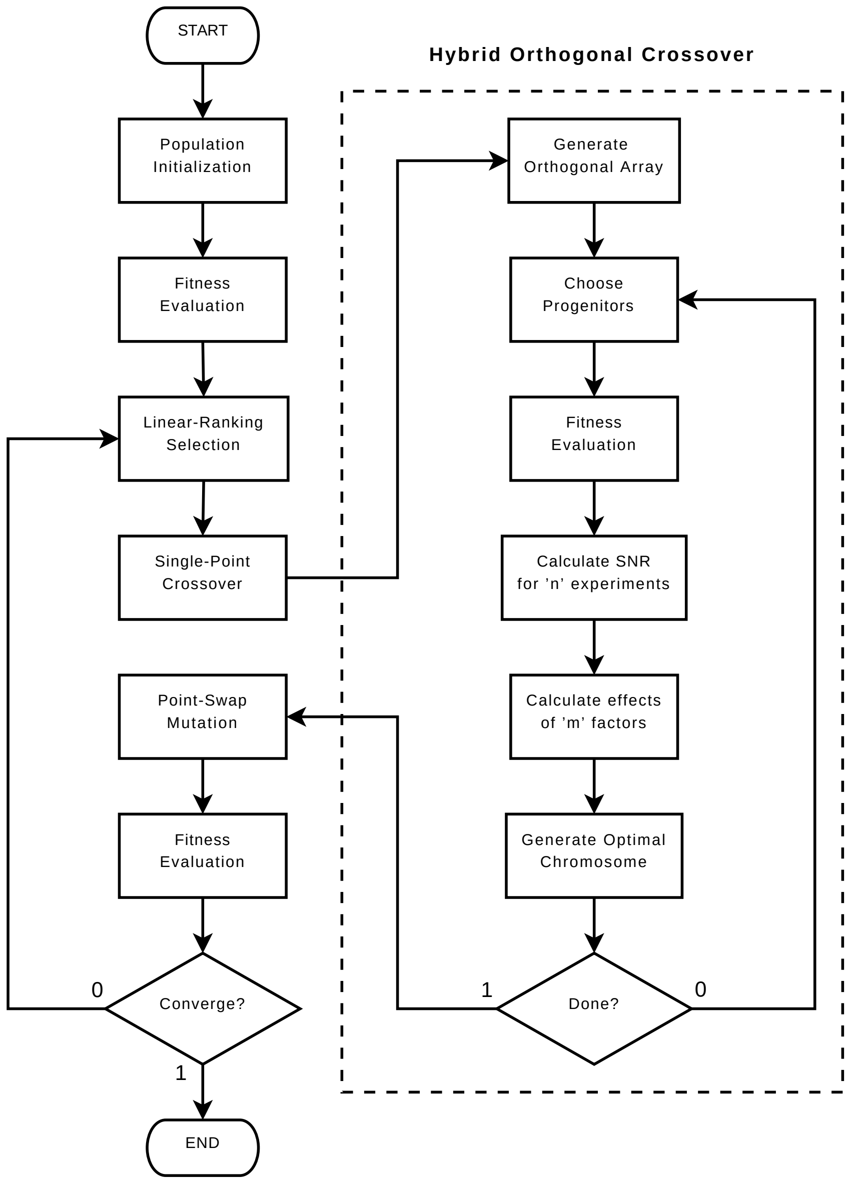
Genetic Algorithms: Overview

- General Theory
- Population Initialization
- Fitness Evaluation
- Progenitor Selection
- Reproduction
- Genetic Diversity
- Convergence



HOG Algorithm Overview

- Linear-Ranking Selection
- Single-Point Crossover
- Orthogonal Crossover
- Point-Swap Mutation



Linear-Ranking Selection

Theory

- Mating eligibility is a function of relative fitness and *selection pressure*
  - Least-fit probability of selection: (η^- / N)
  - Most-fit probability of selection: (η^+ / N)

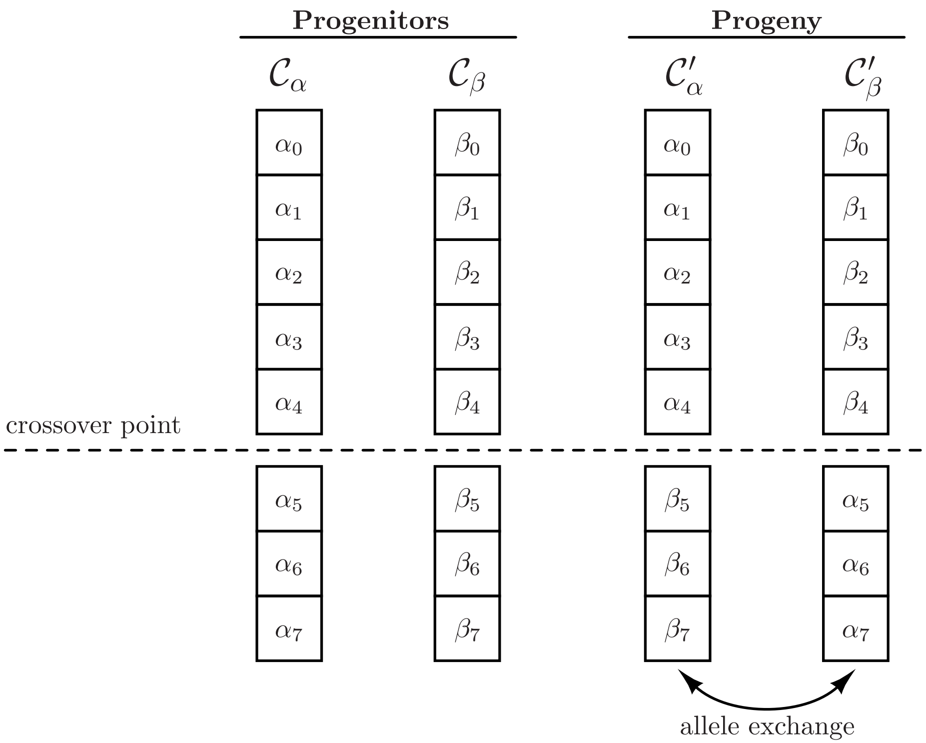
$$\eta^- = 2 - \eta^+$$

Implementation

- Sorted and ranked according to relative fitness from least to greatest
- Assigned an integer index value $i \in [0, N-1]$
- Assigned a probability of selection according to

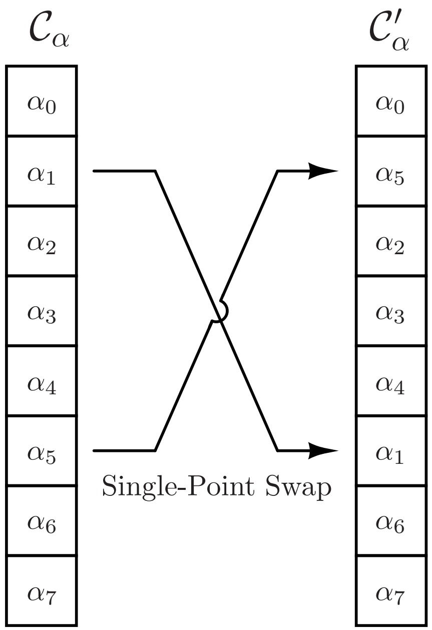
$$p_i = \frac{1}{N}\left(\eta^- + (\eta^+ - \eta^-)\frac{i-1}{N-1}\right)$$

Single-Point Crossover Operator



- Pairs of progenitors are randomly selected from the mating pool and assigned a random number, r, which is uniform over $[0, 1]$
- For $r \geq P_c$: crossover occurs and genetic information is exchanged
- If crossover occurs, a discrete random number, c, uniform over $[0, m]$ is picked to determine the crossover point

Point-Swap Mutation Operator



- Individual chromosomes are randomly selected from the population
- Selected chromosome is assigned a random number, s, which is uniform over $[0, 1]$
- For $s \geq P_m$: mutation occurs and new genetic information is created
- Mutation occurs *unbounded*

Hybrid Orthogonal Crossover via the Taguchi Method

Theory

- Crossover and subsequent fitness evaluation can be viewed as performing experiments on the population
- Design of experiments (DoE) statistical techniques can be implemented to minimize the number of trials required to observe the effects of the experimental factors

Implementation

- Pairs of progenitors are randomly selected from the population whose respective traits become the *experimental factors*
- The factors are mapped to an appropriately sized orthogonal array and the effects for each factor are observed via a metric, S
- Optimal offspring are created from the most beneficial factors

Hybrid Orthogonal Crossover via the Taguchi Method

Orthogonal Arrays (OAs)

- Derived from Latin Squares (e.g. Sudoku)
- Taguchi Method uses 2-Level OAs of the form $L_n(2^{n-1})$
- Number of Trials: n
- Number of Factors: $n-1$

OA Example for 4 trials/3 factors: $$L_4(2^3) = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 2 \\ 2 & 1 & 2 \\ 2 & 2 & 1 \end{pmatrix}$$

- OA elements directly control mapping of factors from a particular progenitor to the experimental matrix
- Each column representing a factor is independent from the other columns

Hybrid Orthogonal Crossover via the Taguchi Method

Taguchi Metric (S_n)

$$S_n = J^2(\boldsymbol{x}_n)$$

- Objective function evaluation squared for n experimental trials (e.g. each row of the 2-level OA)
- $\boldsymbol{x}_n$ is a vector containing the k factors of the nth trial
- Smaller-is-better calculation for global minimization

Optimal Progeny Generation

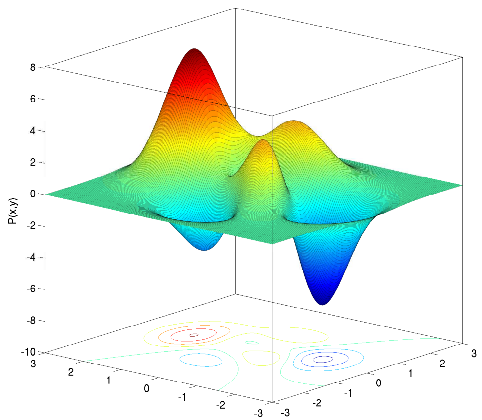
$$E_{x_k P_m} = \sum_{i \in \{k : x_k = P_m\}} S_i$$

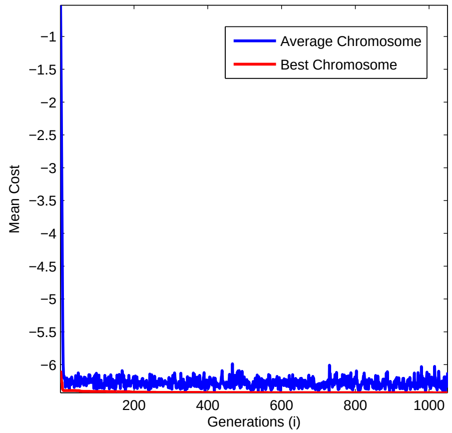
- Effects, E, from all k factors from each progenitor, P_1 and P_2, evaluated based on the metric, S
- Factor with best observed effect populates the optimal progeny
- Generates optimal offspring from the available progenitors

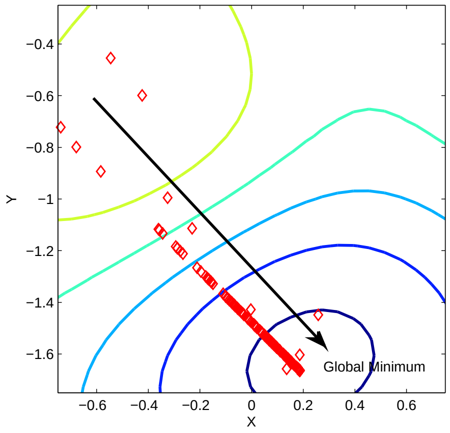
HOG Algorithm Examples

Example: MATLAB Peaks Function

$$P(x,y) = 3\,(1-x)^2\,e^{\left(-x^2-(y+1)^2\right)} - 10\left(\frac{x}{5} - x^3 - y^5\right)e^{\left(-x^2-y^2\right)} - \frac{1}{3}e^{\left(-(x+1)^2-y^2\right)}$$



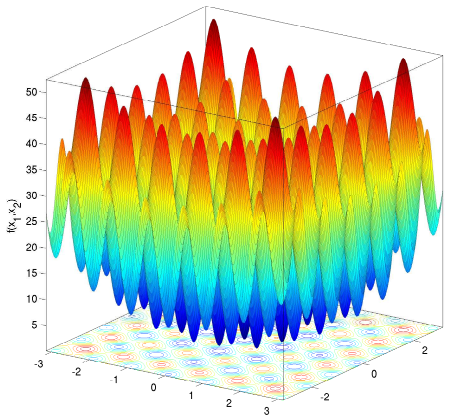


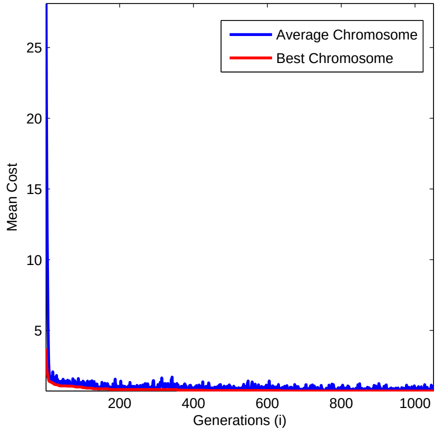


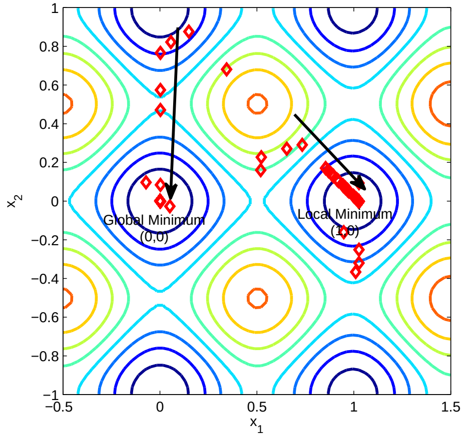
- Obtained by translating and scaling Gaussian distributions
- 2 local minima and 1 global minimum near $(0.2, -1.625)$
- Quick convergence
- Resembles steepest descent convergence path

Example: Rastrigin's Function

$$f(x) = \sum_{i=1}^{N} \left[x_i^2 - 10\cos\left(2\pi x_i\right) + 10 \right] \quad \forall x \in \mathbb{R}^N$$



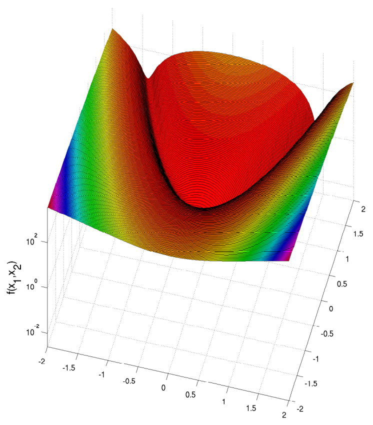


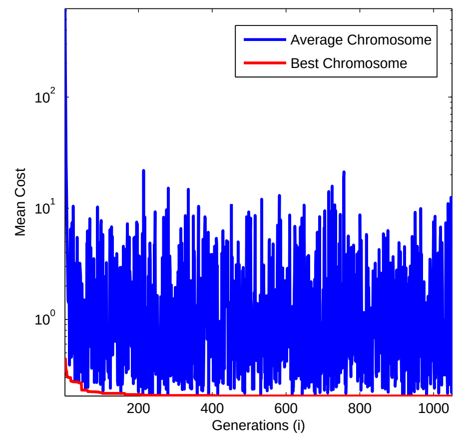


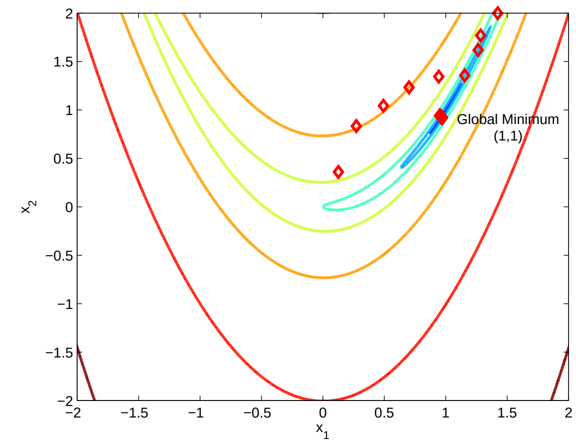
- Widely accepted global optimization algorithm test function
- Characterized by a parabolic egg-crate shape with single lowest point
- Highly multimodal with a global minimum of $f(0,0) = 0$ for $\mathbb{R}^2$
- Numerous local minima provides challenging performance surface

Example: Rosenbrock's Function

$$f(x) = \sum_{i=1}^{N-1} \left[\left(1 - x_i\right)^2 + 100\left(x_{i+1} - x_i^2\right)^2 \right] \quad \forall x \in \mathbb{R}^N$$







- Characterized by a steep parabolic ravine (banana) and gradual descent to global minimum
- Unimodal with a global minimum of $f(1,1) = 0$ for $\mathbb{R}^2$
- Long descent proves challenging for most algorithms

DSM Design and Results

HOG Algorithm Implementation

$$\text{NTF}(z) = \psi_N \left(\frac{1 - a_1 z^{-1}}{1 - b_1 z^{-1}} \right)^m \prod_{n=1}^{M} \frac{\left(1 + c_{1n} z^{-1} + c_{2n} z^{-2}\right)}{\left(1 + d_{1n} z^{-1} + d_{2n} z^{-2}\right)}$$

- Cascade of second-order sections (SOS)
- Reduces coefficient sensitivity to perturbation

$$\mathcal{C}_{\text{NTF}} = \left[a_1, b_1, c_{11}, c_{12}, d_{11}, d_{12}, c_{21}, c_{22}, d_{21}, d_{22}, \ldots, c_{M1}, c_{M2}, d_{M1}, d_{M2}, \psi_N\right]^T$$

$$\mathbb{G}_{\text{NTF}} = \left[\mathcal{C}_{\text{NTF},1} \,\vdots\, \mathcal{C}_{\text{NTF},2} \,\vdots\, \cdots \,\vdots\, \mathcal{C}_{\text{NTF},n} \right]$$

- Population matrix, $\mathbb{G}$, is an aggregate of n chromosomes, $\mathcal{C}$

HOG Algorithm Implementation

$$\text{STF}(z) = \psi_S \left(\frac{1 - \rho_1 z^{-1}}{1 - b_1 z^{-1}} \right)^m \prod_{n=1}^{M} \frac{\left(1 + \nu_{1n} z^{-1} + \nu_{2n} z^{-2}\right)}{\left(1 + d_{1n} z^{-1} + d_{2n} z^{-2}\right)}$$

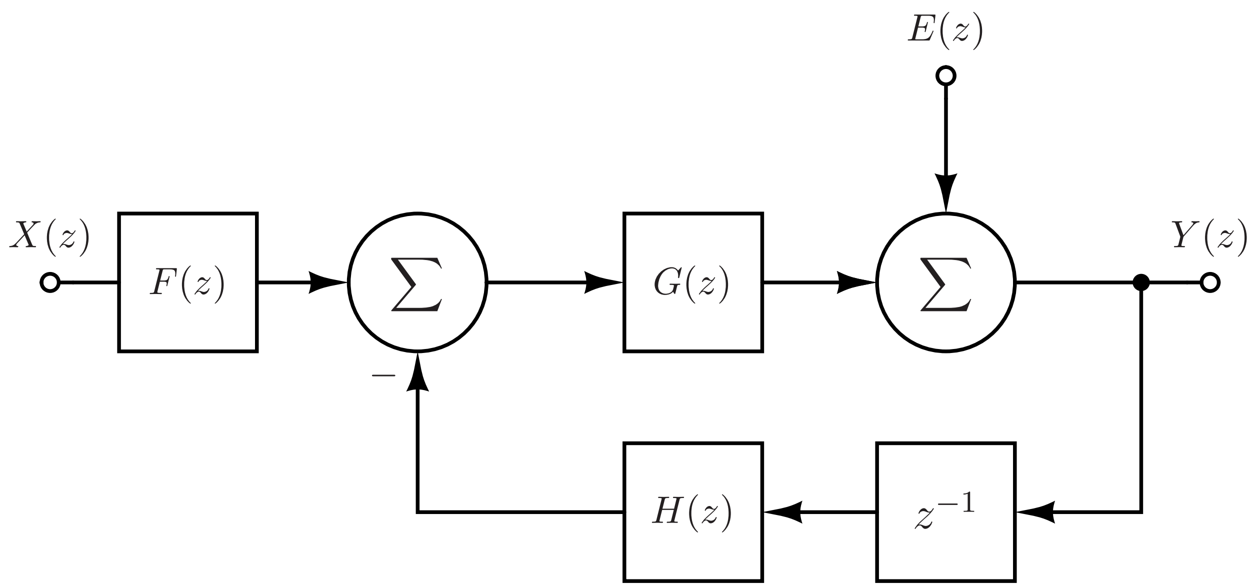
- Cascade of second-order sections (SOS)
- Reduces coefficient sensitivity to perturbation

$$\mathcal{C}_{\text{STF}} = \left[\rho_1, \nu_{11}, \nu_{12}, \nu_{21}, \nu_{22}, \ldots, \nu_{M1}, \nu_{M2}, \psi_S\right]^T$$

$$\mathbb{G}_{\text{STF}} = \left[\mathcal{C}_{\text{STF},1} \,\vdots\, \mathcal{C}_{\text{STF},2} \,\vdots\, \cdots \,\vdots\, \mathcal{C}_{\text{STF},n} \right]$$

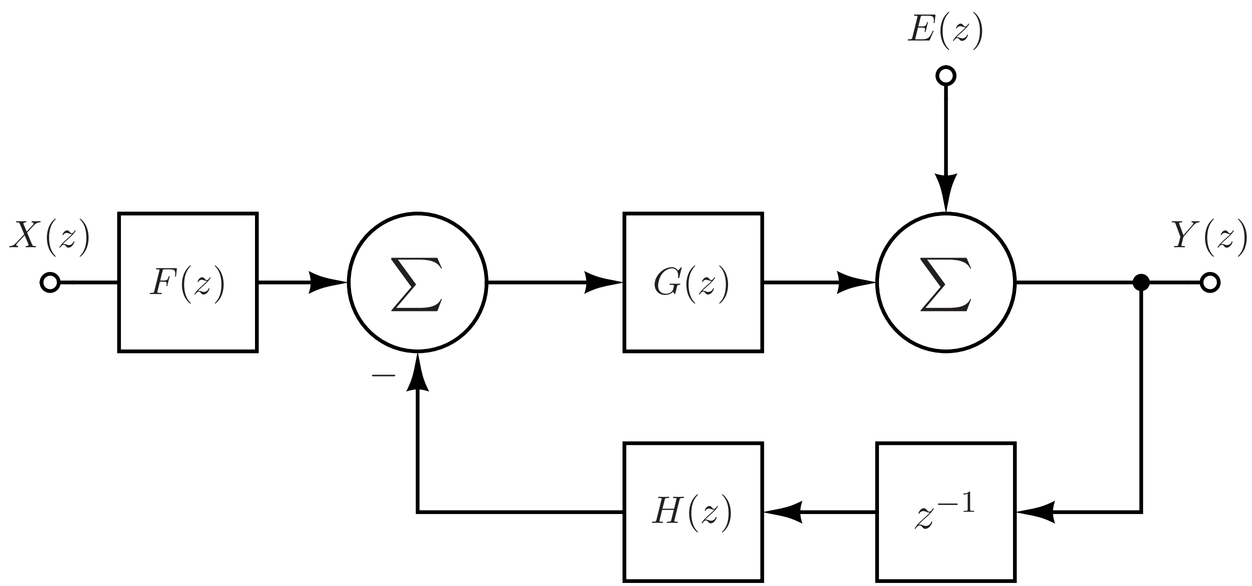
- Population matrix, $\mathbb{G}$, is an aggregate of n chromosomes, $\mathcal{C}$

LTI System Modeling



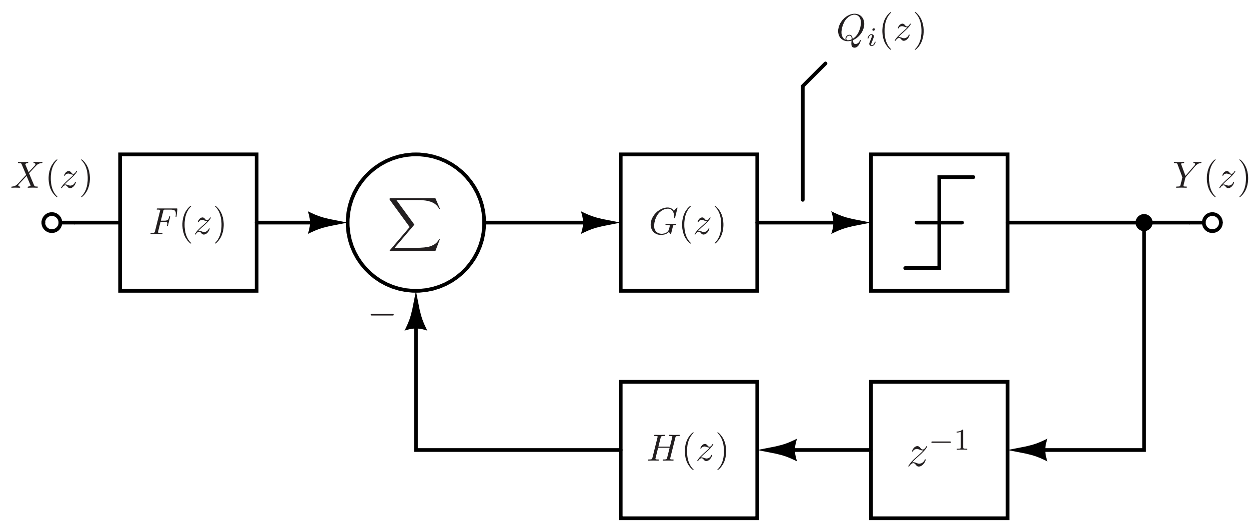
$$\mathrm{STF}(z) = \frac{\sum_{k=0}^{N} \alpha_k z^{-k}}{\sum_{k=0}^{N} \beta_k z^{-k}} \qquad \mathrm{NTF}(z) = \frac{\sum_{k=0}^{N} \gamma_k z^{-k}}{\sum_{k=0}^{N} \beta_k z^{-k}}$$

LTI System Modeling



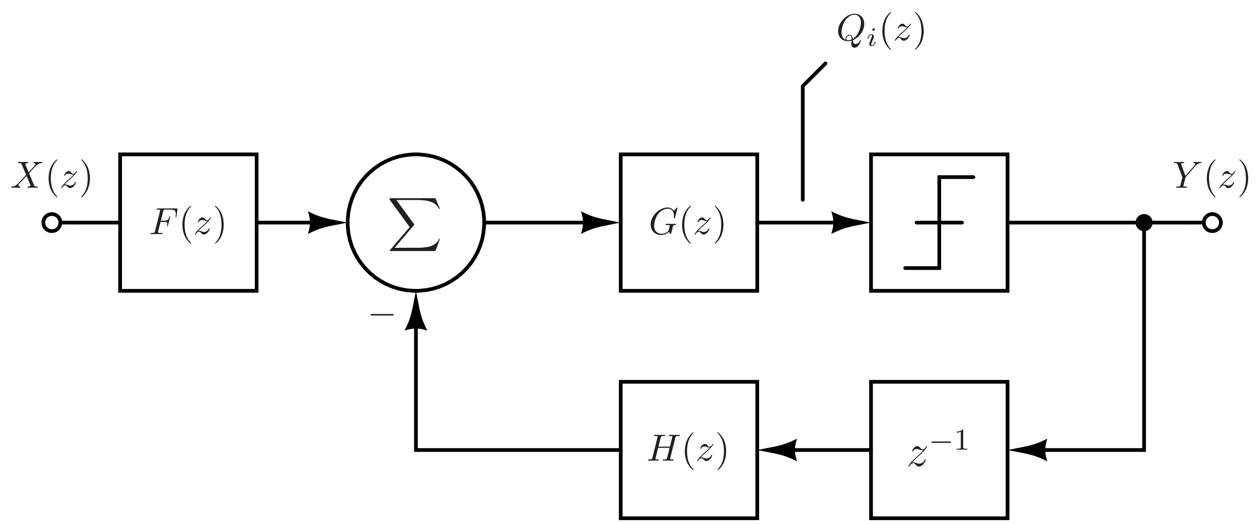
$$F(z) = \sum_{k=0}^{N} \alpha_k z^{-k} \qquad G(z) = \sum_{k=0}^{N} \gamma_k z^{k} \qquad H(z) = \sum_{k=1}^{N} \left(\beta_k - \gamma_k\right) z^{-k+1}$$

Linear Recursive Difference Equation



$$Q_i(z) = G(z)\left(F(z)X(z) - z^{-1}H(z)Y(z)\right)$$

Linear Recursive Difference Equation

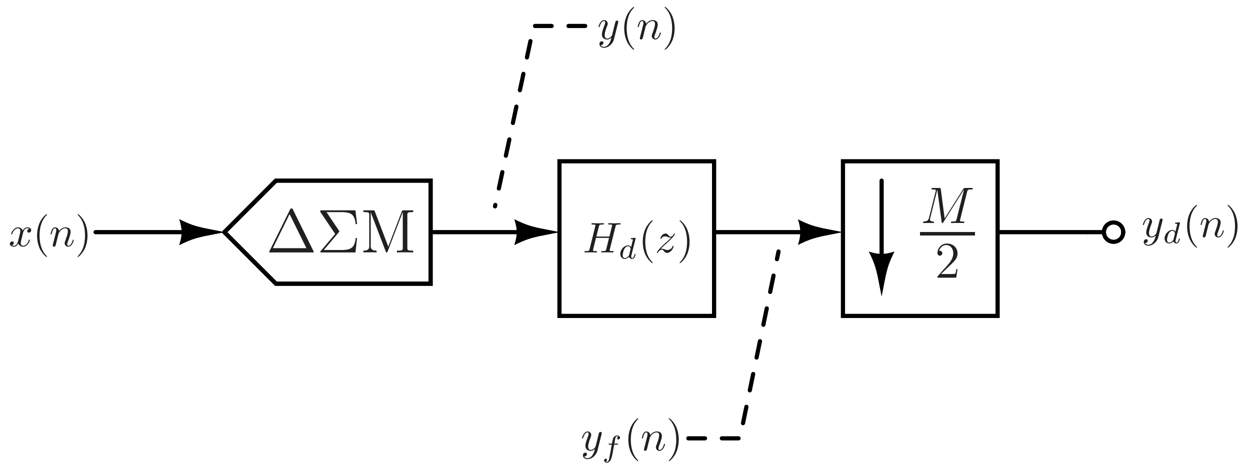


$$q_i(n) = \sum_{k=0}^{N} \alpha_k x(n-k) - \sum_{k=1}^{N} (\beta_k - \gamma_k)\, y(n-k) - \sum_{k=1}^{N} \gamma_k q_i(n-k)$$

$$y(n) = \mathrm{sgn}\left[q_i(n)\right]$$

Frequency Domain Analysis

Output Decimation Filtering



- $\Delta\Sigma$ modulator output, $y(n)$, lowpass filtered by $H_d(z)$
- Filtered output, $y_f(n)$, downsampled by a factor of the OSR, M
- Filtered and downsampled $\Delta\Sigma$ modulator output, $y_d(n)$, is then windowed and analyzed in the frequency domain

Frequency Domain Analysis

Signal-to-Noise Ratio (SNR)

$$\text{SNR}_{\text{dB}} = 10\log\left(\frac{P_s}{P_{n,a}}\right) = 10\log\left(\frac{\sum_{k=K_1}^{K_2} |Y_w(k)|^2}{\sum_{k=0}^{K_1-1} |Y_w(k)|^2 + \sum_{k=K_2+1}^{N/4-1} |Y_w(k)|^2}\right)$$

- K_1 and K_2 are the FFT bins corresponding to the leading and trailing edge of the fundamental lobe
- SNR is the ratio of the power contained in the fundamental lobe to the noise power in the windowed output spectrum

Frequency Domain Analysis

Dynamic Range (DR)

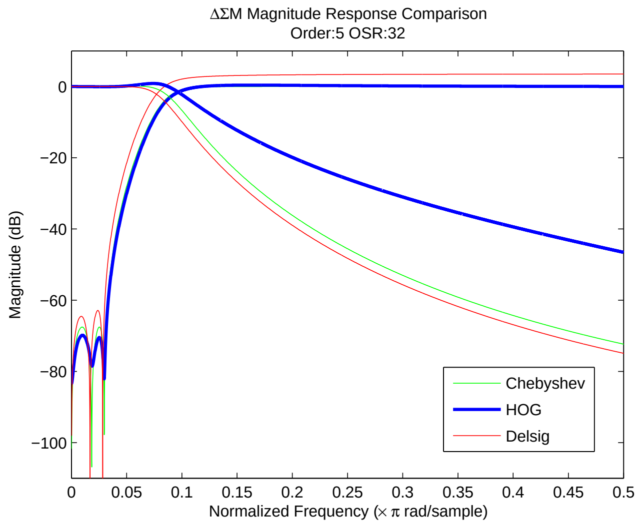
$$\text{DR}_{\text{dB}} = 10\log\left(\frac{P_s}{P_{n,p}}\right) = 10\log\left(\frac{4}{N_p N}\sum_{k=K_1}^{K_2}|Y_w(k)|^2\right)$$

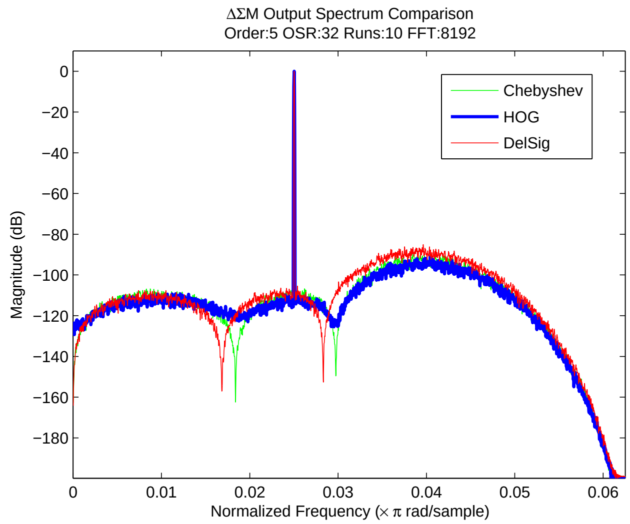
$$N_p = \max\langle|Y_{d,e}(k)|\rangle$$

- K_1 and K_2 are the FFT bins corresponding to the leading and trailing edge of the fundamental lobe
- DR is the ratio of the power contained in the fundamental lobe to an output noise spectrum equal in magnitude to the maximum observed noise component, N_p

ΔΣ Modulator Optimization Results

5th Order ΔΣ Modulator

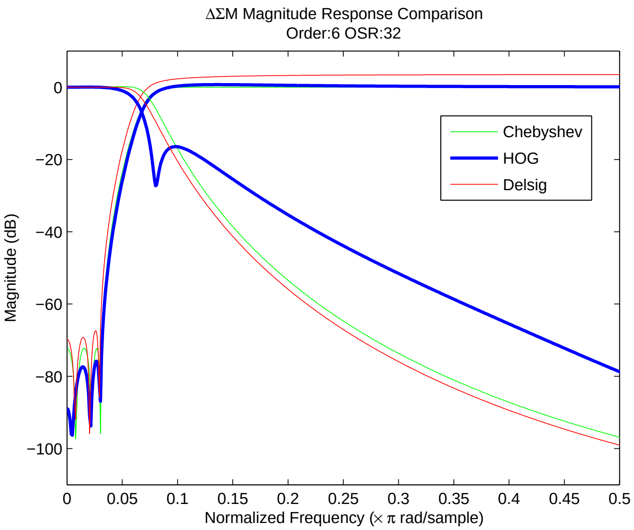


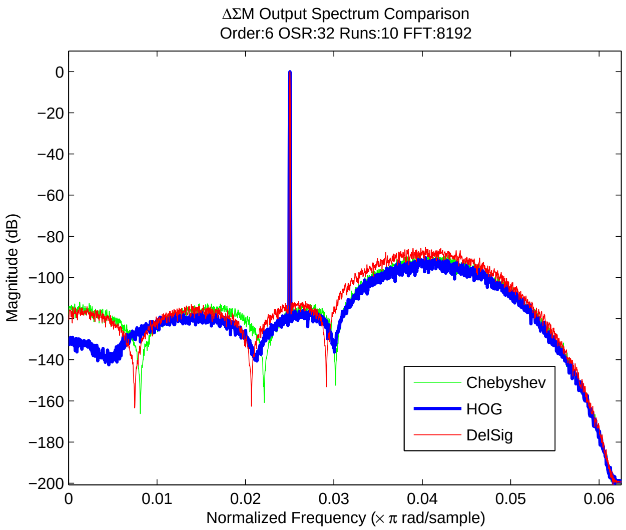


Design Method	SNR_{dB}	DR_{dB}
DelSig Toolbox	83	66
Chebyshev Filter	81	72
HOG Algorithm	87	76

ΔΣ Modulator Optimization Results

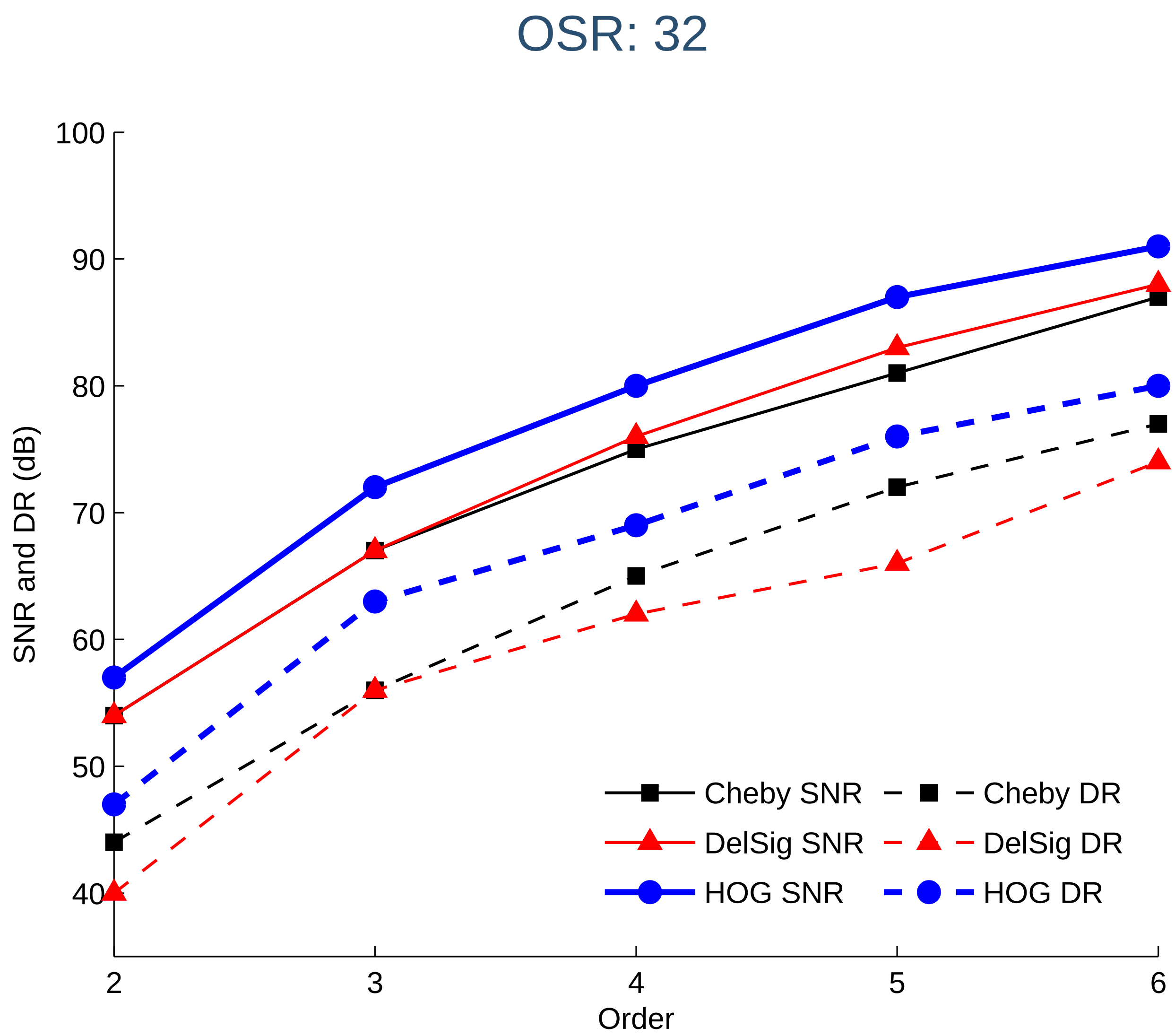
6th Order ΔΣ Modulator



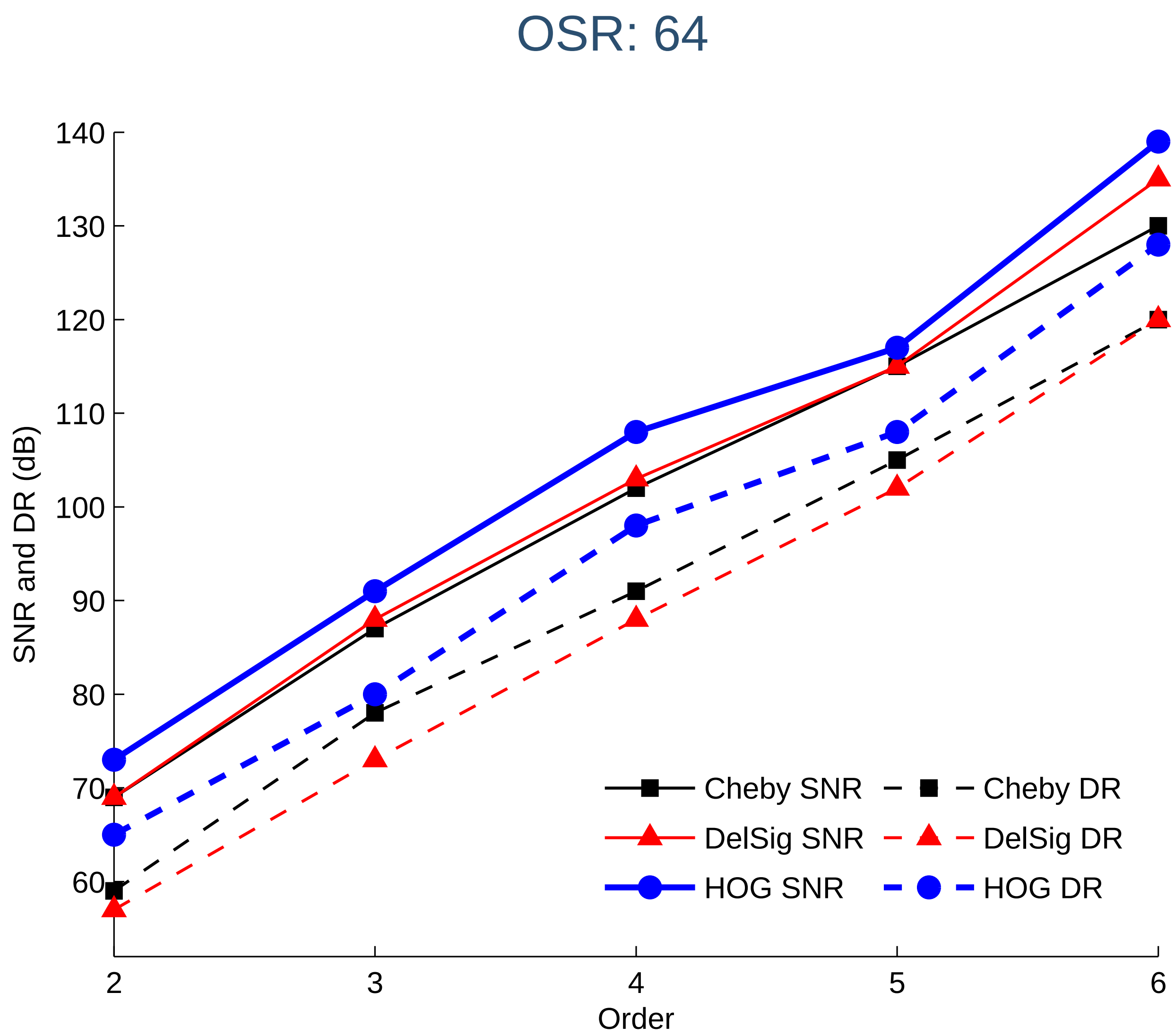


Design Method	SNR_{dB}	DR_{dB}
DelSig Toolbox	88	74
Chebyshev Filter	87	77
HOG Algorithm	91	80

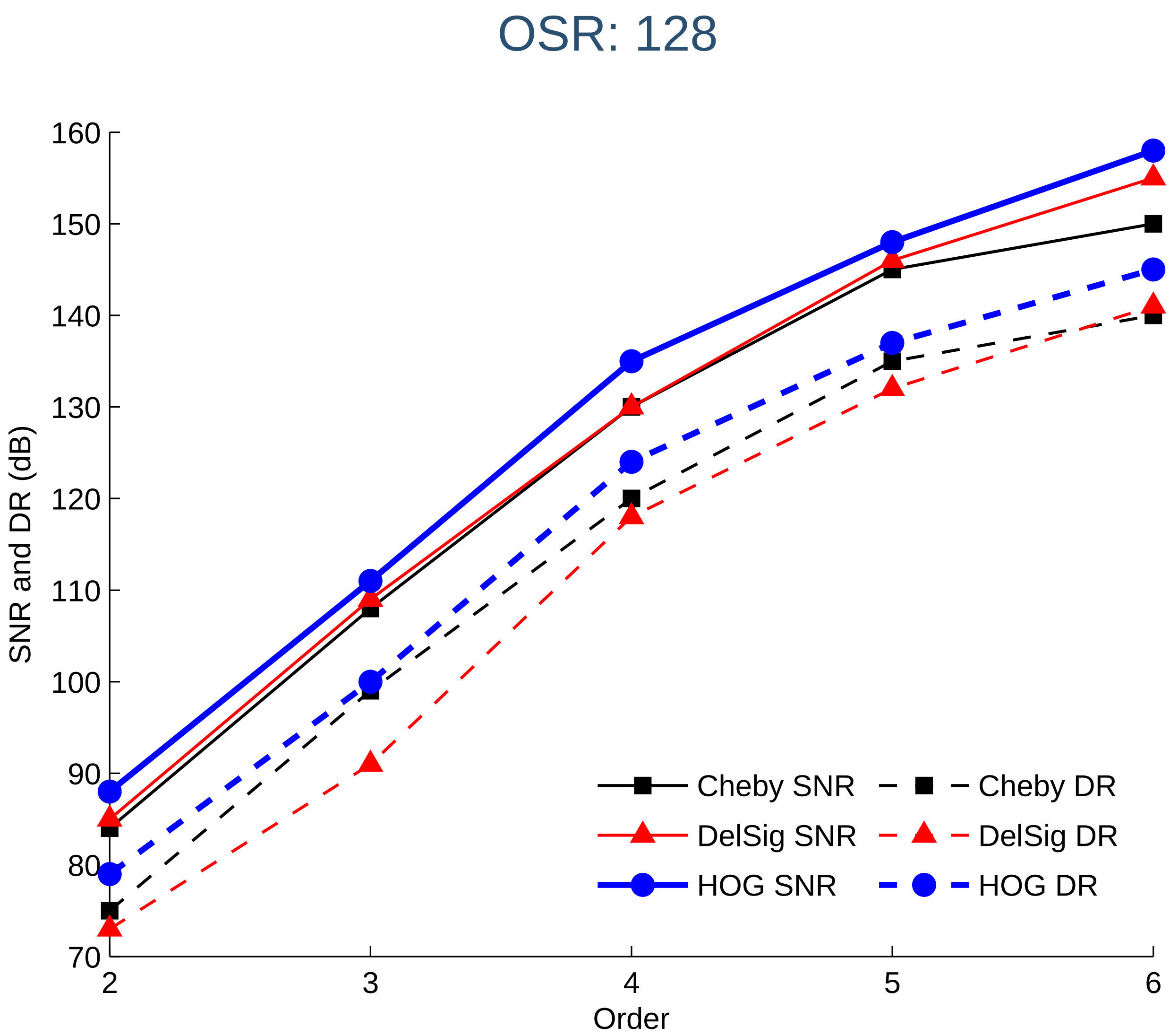
ΔΣ Modulator Optimization Results



ΔΣ Modulator Optimization Results



ΔΣ Modulator Optimization Results



Conclusions

HOG Algorithm Conclusions

- Effectively solves widely accepted benchmark unimodal and multimodal problems across a broad range of dimensions
- Robust methods inherent to the orthogonal crossover operator increase *convergence confidence* by providing repeatable highly-optimized solutions which are equivalent to known optimal solutions
- Algorithm performance (computational load, mean solution, standard deviation, etc) is consistent with similar algorithms

HOG Algorithm Conclusions

- Effectively solves widely accepted benchmark unimodal and multimodal problems across a broad range of dimensions
- Robust methods inherent to the orthogonal crossover operator increase *convergence confidence* by providing repeatable highly-optimized solutions which are equivalent to known optimal solutions
- Algorithm performance (computational load, mean solution, standard deviation, etc) is consistent with similar algorithms

HOG Algorithm Conclusions

- Effectively solves widely accepted benchmark unimodal and multimodal problems across a broad range of dimensions
- Robust methods inherent to the orthogonal crossover operator increase *convergence confidence* by providing repeatable highly-optimized solutions which are equivalent to known optimal solutions
- Algorithm performance (computational load, mean solution, standard deviation, etc) is consistent with similar algorithms

Optimal $\Delta\Sigma$ Modulator Design Conclusions

- Traditional polynomial based design methods shown to perform as good as EDA tool based design techniques for most cases
- When compared to both classical polynomial based design techniques and contemporary EDA based design techniques, the SNR and DR optimization based design was shown to provide increased DR and SNR thereby increasing the $\Delta\Sigma$ modulators effective resolution
- HOG algorithm based method determines optimal NTFs and STFs with respect to SNR and DR

Optimal $\Delta\Sigma$ Modulator Design Conclusions

- Traditional polynomial based design methods shown to perform as good as EDA tool based design techniques for most cases

- When compared to both classical polynomial based design techniques and contemporary EDA based design techniques, the SNR and DR optimization based design was shown to provide increased DR and SNR thereby increasing the $\Delta\Sigma$ modulators effective resolution

- HOG algorithm based method determines optimal NTFs and STFs with respect to SNR and DR

Optimal ΔΣ Modulator Design Conclusions

- Traditional polynomial based design methods shown to perform as good as EDA tool based design techniques for most cases
- When compared to both classical polynomial based design techniques and contemporary EDA based design techniques, the SNR and DR optimization based design was shown to provide increased DR and SNR thereby increasing the ΔΣ modulators effective resolution
- HOG algorithm based method determines optimal NTFs and STFs with respect to SNR and DR

Future Work

- Modify the HOG algorithm mutation operator to introduce genetic information which does not exist in the current population's genotype
- Introduce a check-and-repair operator to monitor mutation with respect to the current objective function
- Augment the HOG algorithm with a gradient based search method to find the optimal value once in the region of the global minimum
- Automate stability analysis to *close the loop* for optimal $\Delta\Sigma$ modulator design
- Extend the application of the HOG algorithm to include the optimal design of continuous-time $\Delta\Sigma$ modulators
- Include non-ideal circuit level implementation considerations such as jitter tolerance, loop delay, and lossy integration in $\Delta\Sigma$ modulator objective function
- Implement the updated HOG algorithm as a multiobjective, constrained global optimizer to determine Pareto optimal system functions including rigid stability criterion

Future Work

- Modify the HOG algorithm mutation operator to introduce genetic information which does not exist in the current population's genotype
- Introduce a check-and-repair operator to monitor mutation with respect to the current objective function
- Augment the HOG algorithm with a gradient based search method to find the optimal value once in the region of the global minimum
- Automate stability analysis to *close the loop* for optimal $\Delta\Sigma$ modulator design
- Extend the application of the HOG algorithm to include the optimal design of continuous-time $\Delta\Sigma$ modulators
- Include non-ideal circuit level implementation considerations such as jitter tolerance, loop delay, and lossy integration in $\Delta\Sigma$ modulator objective function
- Implement the updated HOG algorithm as a multiobjective, constrained global optimizer to determine Pareto optimal system functions including rigid stability criterion

Future Work

- Modify the HOG algorithm mutation operator to introduce genetic information which does not exist in the current population's genotype
- Introduce a check-and-repair operator to monitor mutation with respect to the current objective function
- Augment the HOG algorithm with a gradient based search method to find the optimal value once in the region of the global minimum
- Automate stability analysis to *close the loop* for optimal $\Delta\Sigma$ modulator design
- Extend the application of the HOG algorithm to include the optimal design of continuous-time $\Delta\Sigma$ modulators
- Include non-ideal circuit level implementation considerations such as jitter tolerance, loop delay, and lossy integration in $\Delta\Sigma$ modulator objective function
- Implement the updated HOG algorithm as a multiobjective, constrained global optimizer to determine Pareto optimal system functions including rigid stability criterion

Future Work

- Modify the HOG algorithm mutation operator to introduce genetic information which does not exist in the current population's genotype
- Introduce a check-and-repair operator to monitor mutation with respect to the current objective function
- Augment the HOG algorithm with a gradient based search method to find the optimal value once in the region of the global minimum
- Automate stability analysis to *close the loop* for optimal $\Delta\Sigma$ modulator design
- Extend the application of the HOG algorithm to include the optimal design of continuous-time $\Delta\Sigma$ modulators
- Include non-ideal circuit level implementation considerations such as jitter tolerance, loop delay, and lossy integration in $\Delta\Sigma$ modulator objective function
- Implement the updated HOG algorithm as a multiobjective, constrained global optimizer to determine Pareto optimal system functions including rigid stability criterion

Future Work

- Modify the HOG algorithm mutation operator to introduce genetic information which does not exist in the current population's genotype
- Introduce a check-and-repair operator to monitor mutation with respect to the current objective function
- Augment the HOG algorithm with a gradient based search method to find the optimal value once in the region of the global minimum
- Automate stability analysis to *close the loop* for optimal $\Delta\Sigma$ modulator design
- Extend the application of the HOG algorithm to include the optimal design of continuous-time $\Delta\Sigma$ modulators
- Include non-ideal circuit level implementation considerations such as jitter tolerance, loop delay, and lossy integration in $\Delta\Sigma$ modulator objective function
- Implement the updated HOG algorithm as a multiobjective, constrained global optimizer to determine Pareto optimal system functions including rigid stability criterion

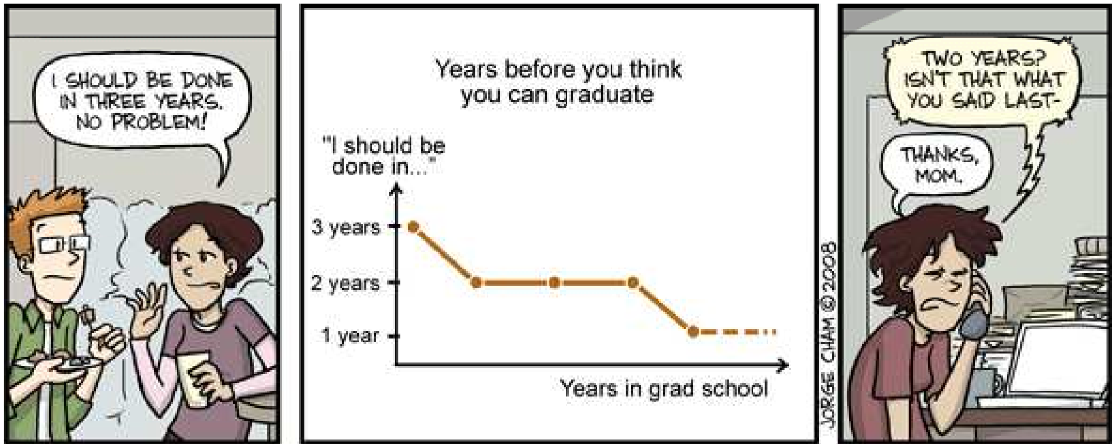
Future Work

- Modify the HOG algorithm mutation operator to introduce genetic information which does not exist in the current population's genotype
- Introduce a check-and-repair operator to monitor mutation with respect to the current objective function
- Augment the HOG algorithm with a gradient based search method to find the optimal value once in the region of the global minimum
- Automate stability analysis to *close the loop* for optimal $\Delta\Sigma$ modulator design
- Extend the application of the HOG algorithm to include the optimal design of continuous-time $\Delta\Sigma$ modulators
- Include non-ideal circuit level implementation considerations such as jitter tolerance, loop delay, and lossy integration in $\Delta\Sigma$ modulator objective function
- Implement the updated HOG algorithm as a multiobjective, constrained global optimizer to determine Pareto optimal system functions including rigid stability criterion

Future Work

- Modify the HOG algorithm mutation operator to introduce genetic information which does not exist in the current population's genotype
- Introduce a check-and-repair operator to monitor mutation with respect to the current objective function
- Augment the HOG algorithm with a gradient based search method to find the optimal value once in the region of the global minimum
- Automate stability analysis to *close the loop* for optimal $\Delta\Sigma$ modulator design
- Extend the application of the HOG algorithm to include the optimal design of continuous-time $\Delta\Sigma$ modulators
- Include non-ideal circuit level implementation considerations such as jitter tolerance, loop delay, and lossy integration in $\Delta\Sigma$ modulator objective function
- Implement the updated HOG algorithm as a multiobjective, constrained global optimizer to determine Pareto optimal system functions including rigid stability criterion

Questions



I SHOULD BE DONE IN THREE YEARS. NO PROBLEM!
Years before you think you can graduate
"I should be done in..."
3 years
2 years
1 year
Years in grad school
JORGE CHAM © 2008
TWO YEARS? ISN'T THAT WHAT YOU SAID LAST-
THANKS, MOM.