Perturbation Analysis for Word-length Optimization

George A. Constantinides
Imperial College, London

May 20, 2003

Word-length Optimization

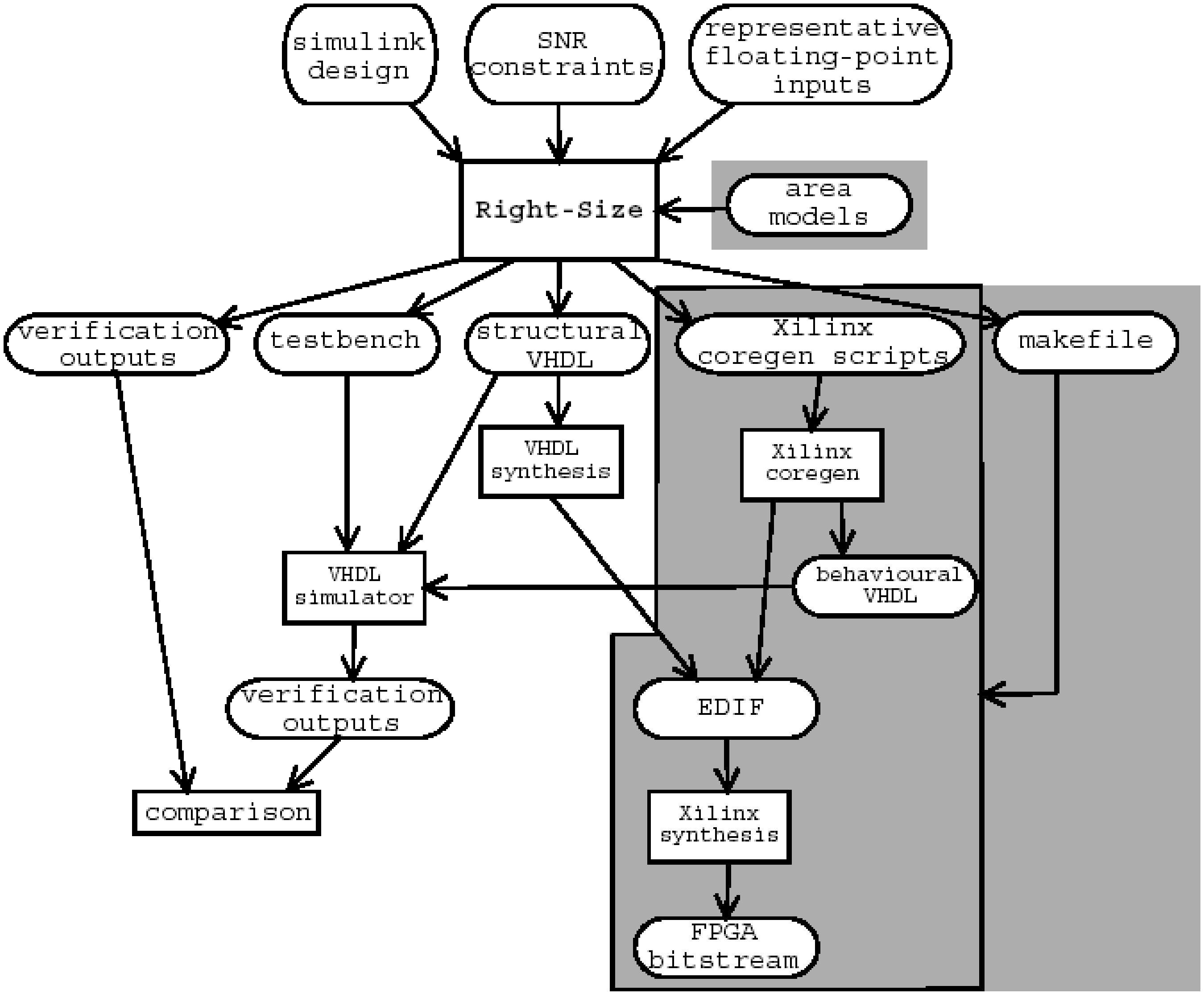
- It’s hard.
  - For linear time-invariant (LTI) systems, it’s proven NP-hard
- It’s necessary.
  - Infinite-precision datapaths do not exist
  - Changing representation can drastically affect area, power, speed, quality of output
- It’s not supported in the vendor toolchains.
  - Someone has to do it by hand

- It’s just not fun.

Contributions

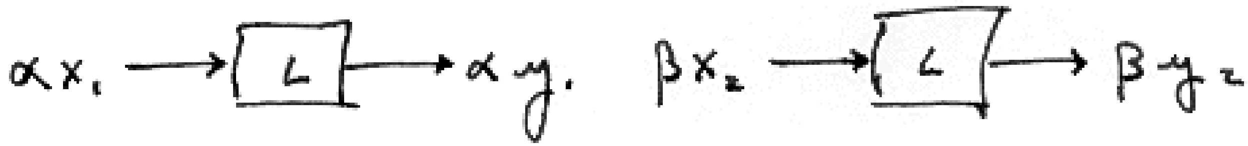
- Uses a high-level design tool (Simulink) for input specification
- Eliminates specifying bit-true hardware in the design process
- Semi-automatic tradeoff of area, power, and speed against user-specified SNR
  - Tool called Right-Size
- Extends previous work on LTI system modeling to non-linear systems

Design Flow

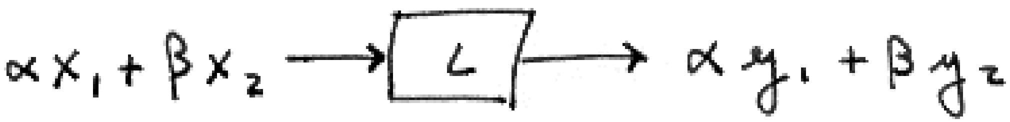


simulink design
SNR constraints
representative floating-point inputs
Right-Size
area models
verification outputs
testbench
structural VHDL
Xilinx coregen scripts
makefile
VHDL synthesis
Xilinx coregen
behavioural VHDL
VHDL simulator
verification outputs
EDIF
comparison
Xilinx synthesis
FPGA bitstream

LTI Systems

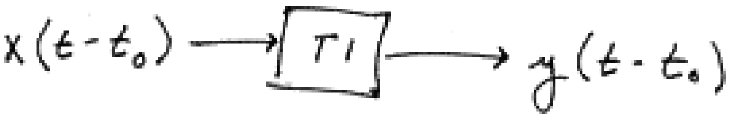


(a)



(b)

(a)



(b)

Linearizing Systems

- **Assumption:**
  - **Quantization errors induced by rounding or truncation are sufficiently small to not affect the macroscopic behavior of the system**

$$Y[t] = f(X_1[t], X_2[t], \ldots, X_n[t])$$

$$x_i[t] \text{ is small perturbation on } X_i[t]$$

$$y[t] \approx x_1[t]\frac{\partial f}{\partial X_1} + \ldots + x_n[t]\frac{\partial f}{\partial X_n}$$

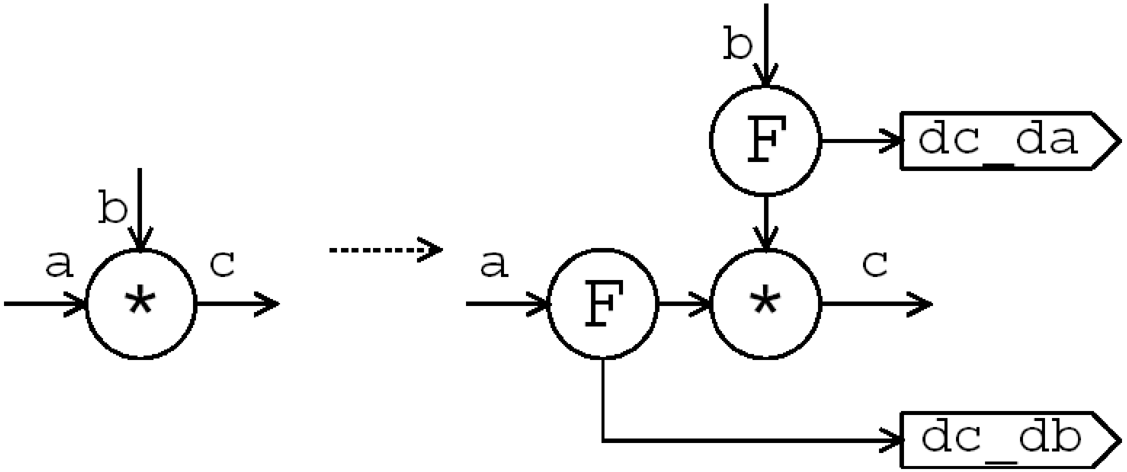
- **Thus, each non-linear component can be locally linearized**
  - **Replaced by its “small-signal equivalent”**
  - **Now, output can be predicted as a linear, time-varying system**

Multiply: Derivative Monitors

$$f(X_1, X_2) = X_1 X_2$$

$$\frac{\partial f}{\partial X_1} = X_2, \frac{\partial f}{\partial X_2} = X_1$$

- Evaluate the differential coefficients of the Taylor series model during simulation
- Coefficients written out to file

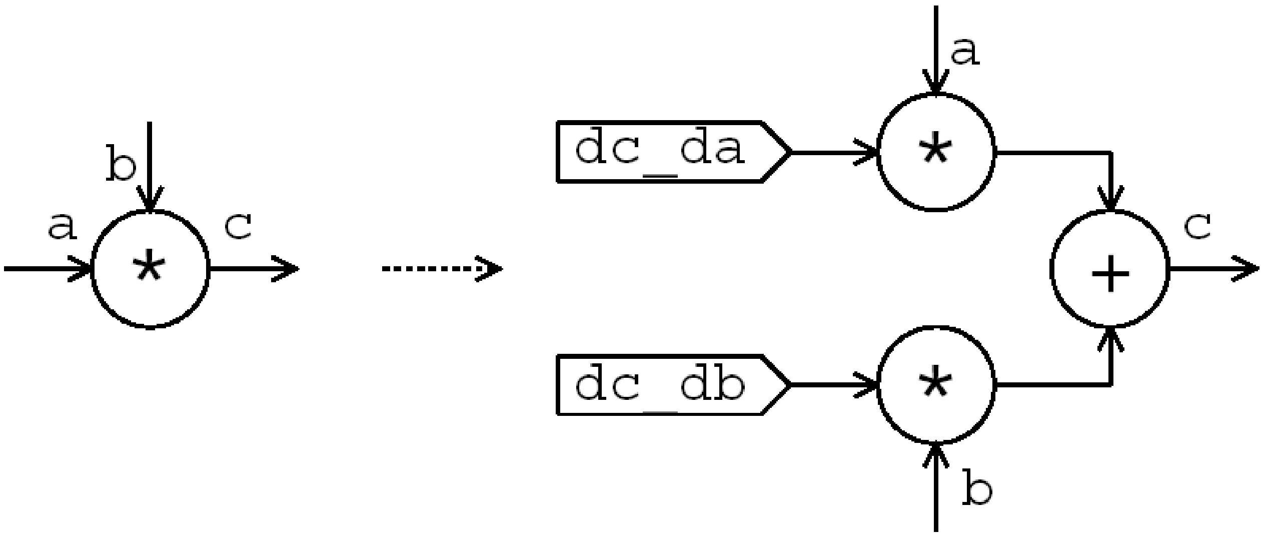


(a) multiplier node

(b) with derivative monitors

Multiply: Linearization

- Replace nonlinear component with Taylor model
- Read Taylor coefficients from previous step

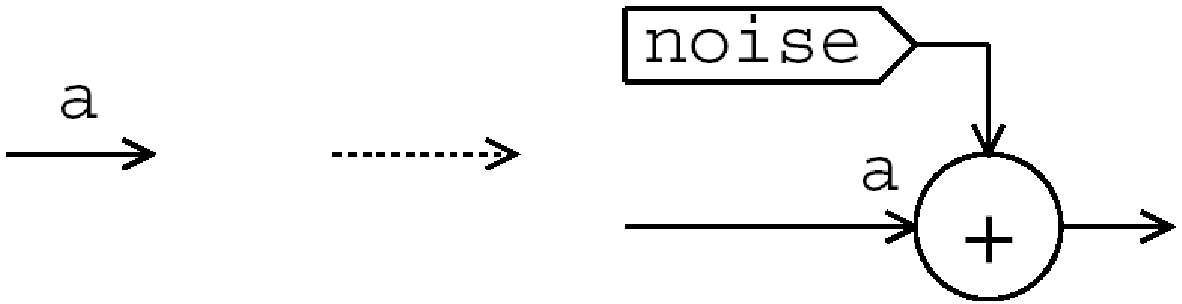


(a) multiplier node

(b) its first-order Taylor model

Multiply: Noise Injection

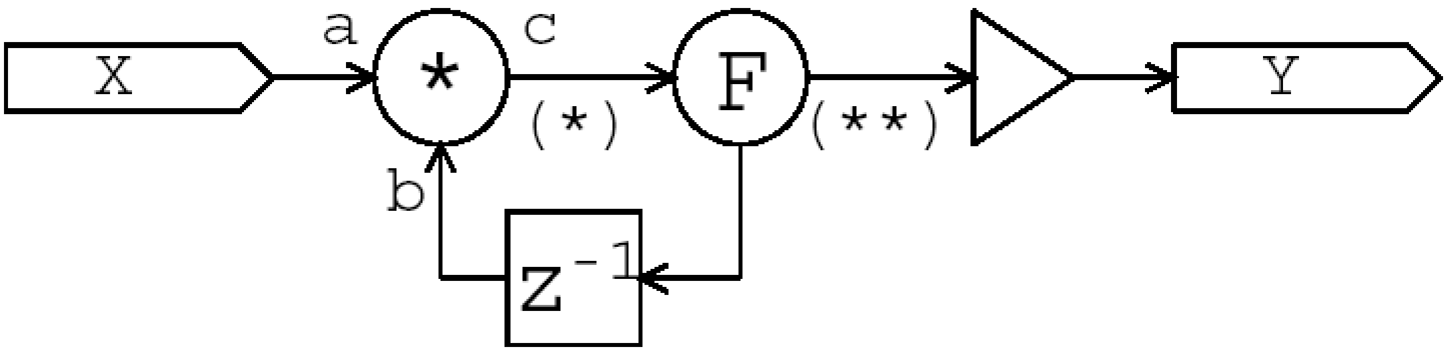
- Noise is the quantization error
- We can predict the sensitivity of a linear system to this additive noise (perturbation)
- Apply this transformation to each signal, propagating zeros from primary inputs



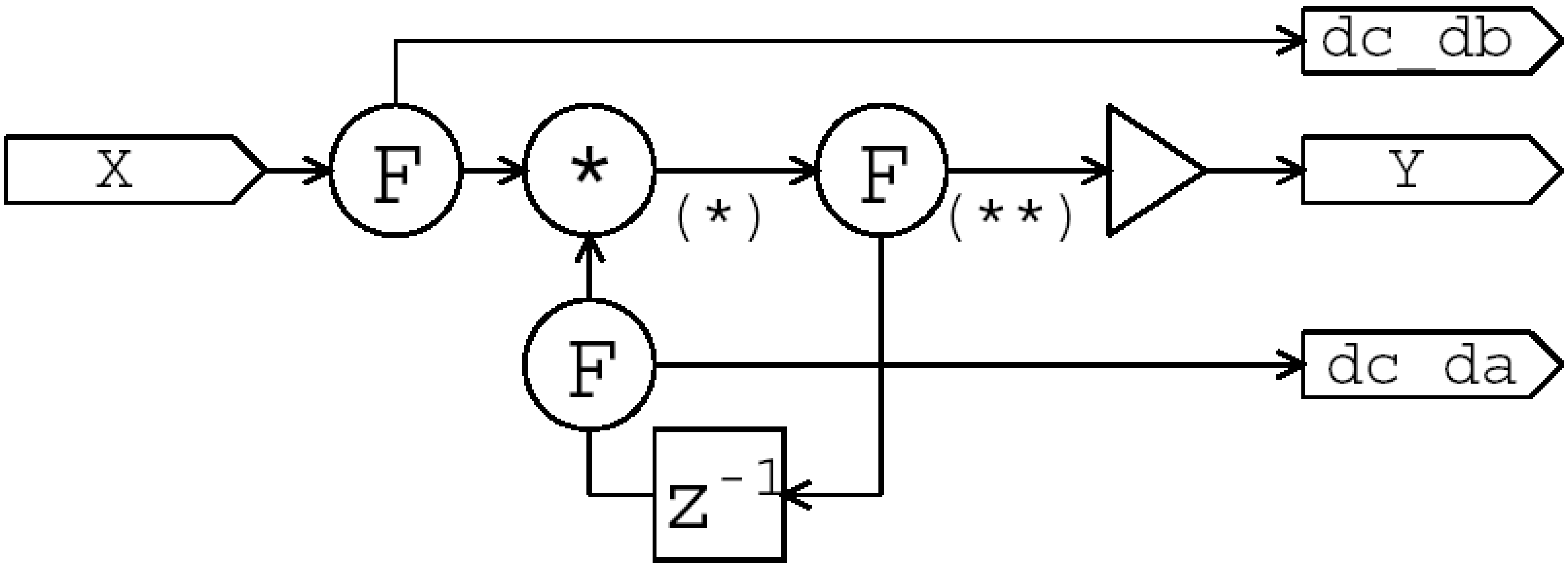
(a) original signal

(b) with noise injection

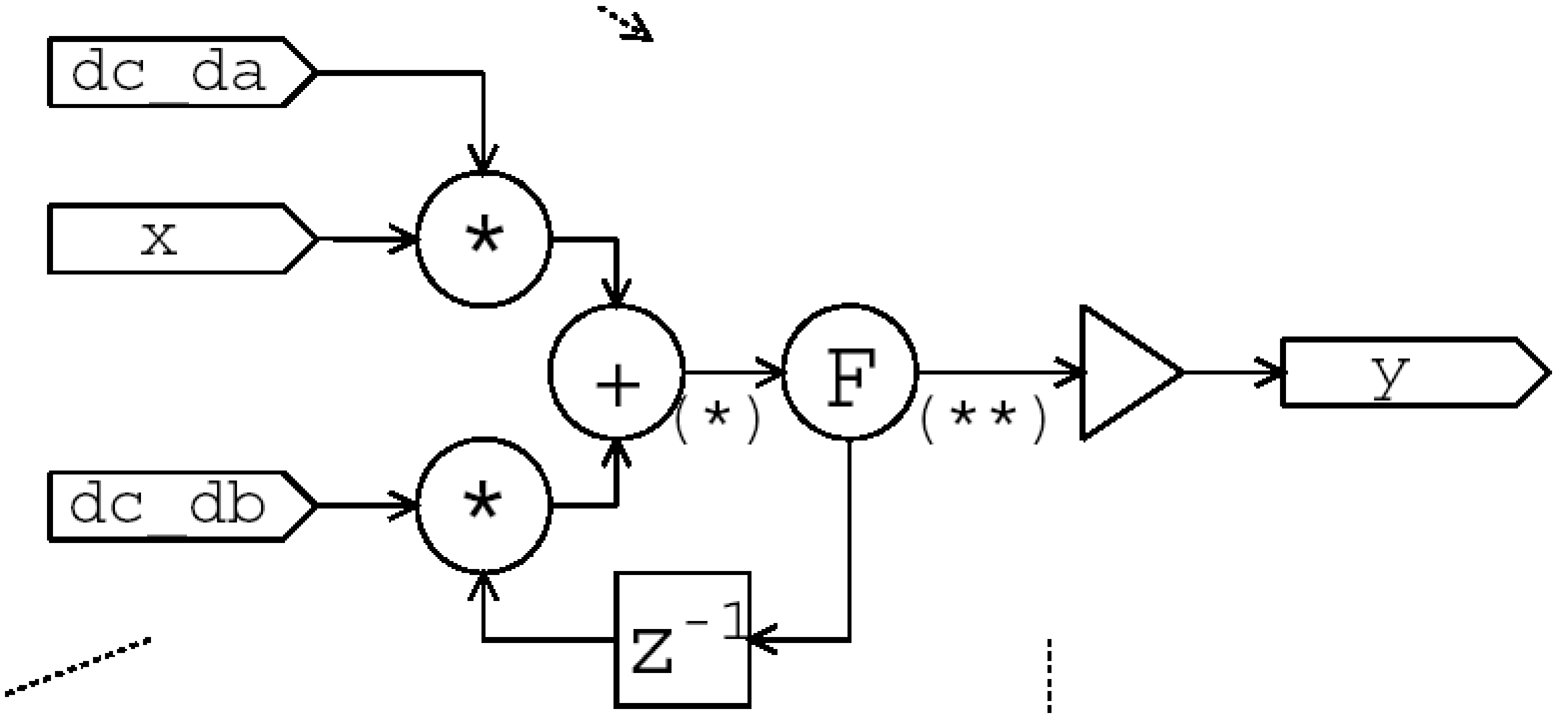
Example



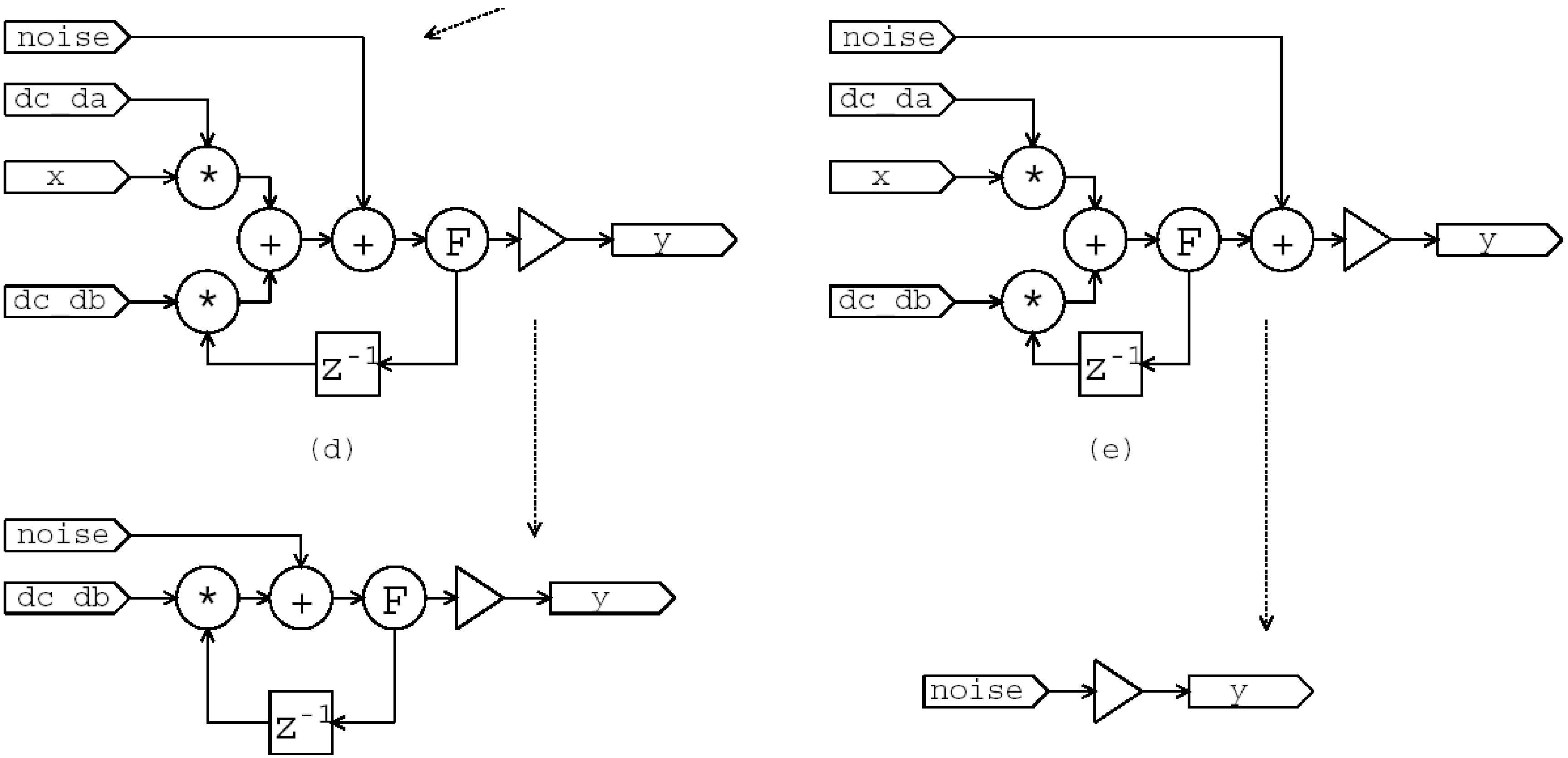
Insert Derivative Monitors



Linearized DFG



Adding Noise (Quantization)



(d)

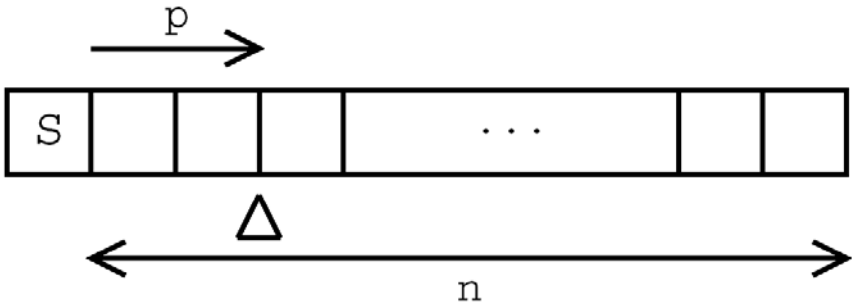
(e)

(f)

(g)

Scaling Analysis

- Each signal in a multiple word-length system has two parameters
  - Word-length (n)
  - Scaling (p)



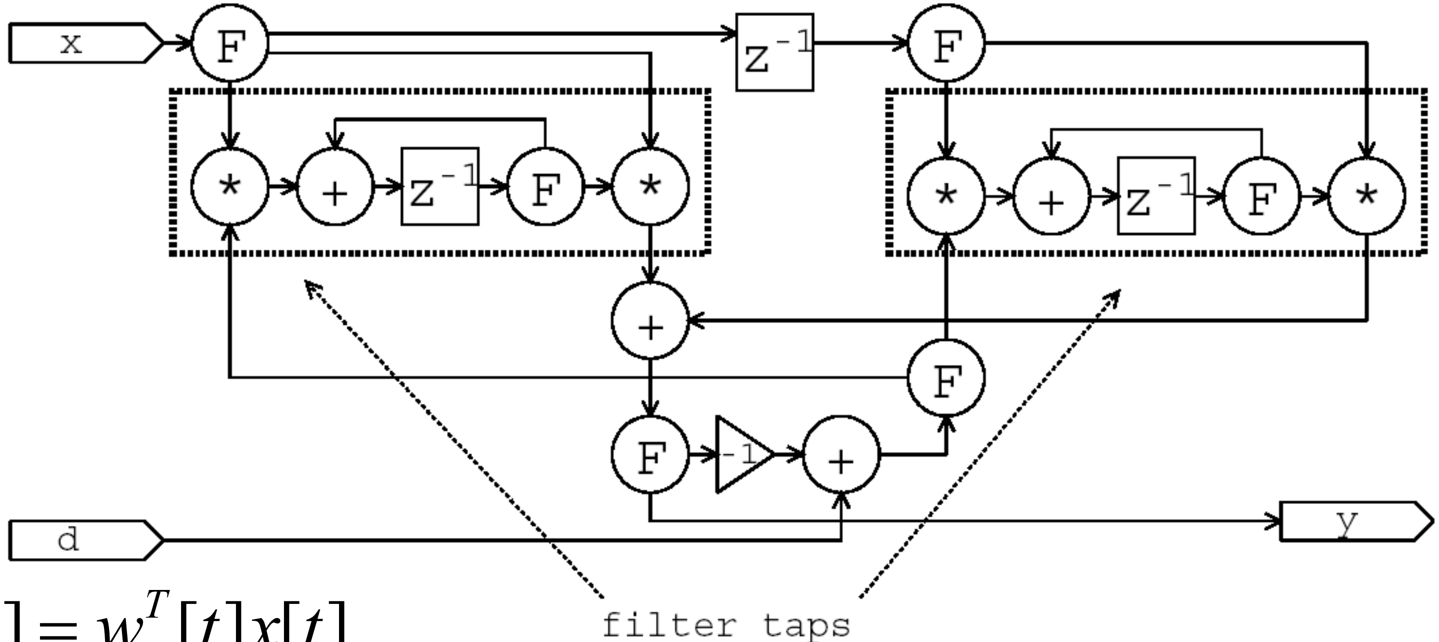
- Perform simulation to find peak signal value reached by each signal
- Scale up by a safety factor (guard bits)

Word-length Optimization

- Two-stage algorithm
  - Compute an optimal uniform wordlength
    - Minimize area under user-defined maximum allowable error
  - Use heuristic to find smaller wordlengths for individual signals
    - Scale up optimal uniform wordlength by a fixed factor
    - Greedily reduce wordlength of individual signals bit by bit according to area pay-off
- All built using area models of Xilinx Coregen arithmetic units

Case Study: Adaptive Filtering

- **Least-mean-square (LMS) filter**
- **Adapts filter coefficients for system identification**



$$y[t] = w^T[t]x[t]$$

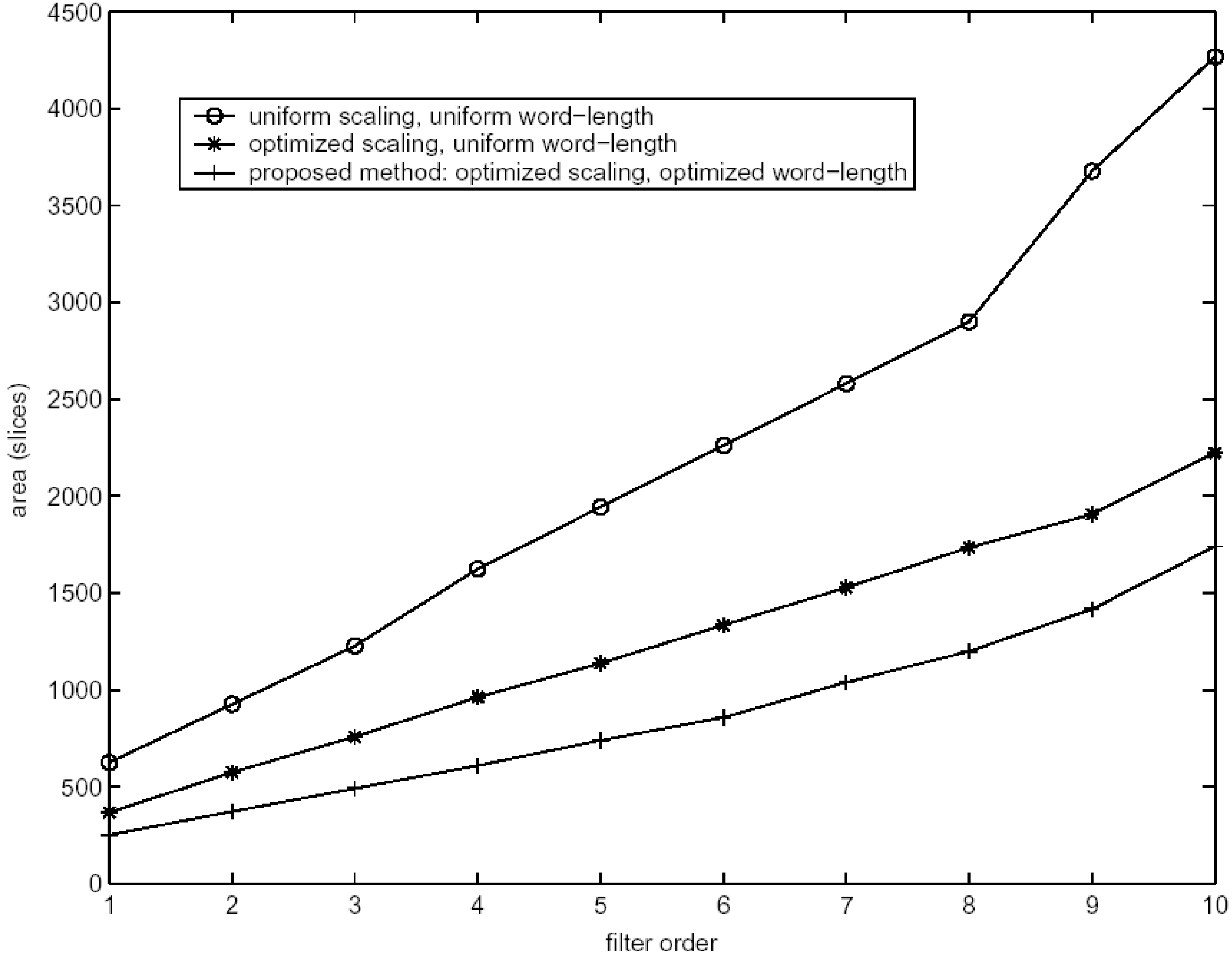
$$e[t] = d[t] - y[t]$$

$$w[t+1] = w[t] + \mu x[t]e[t]$$

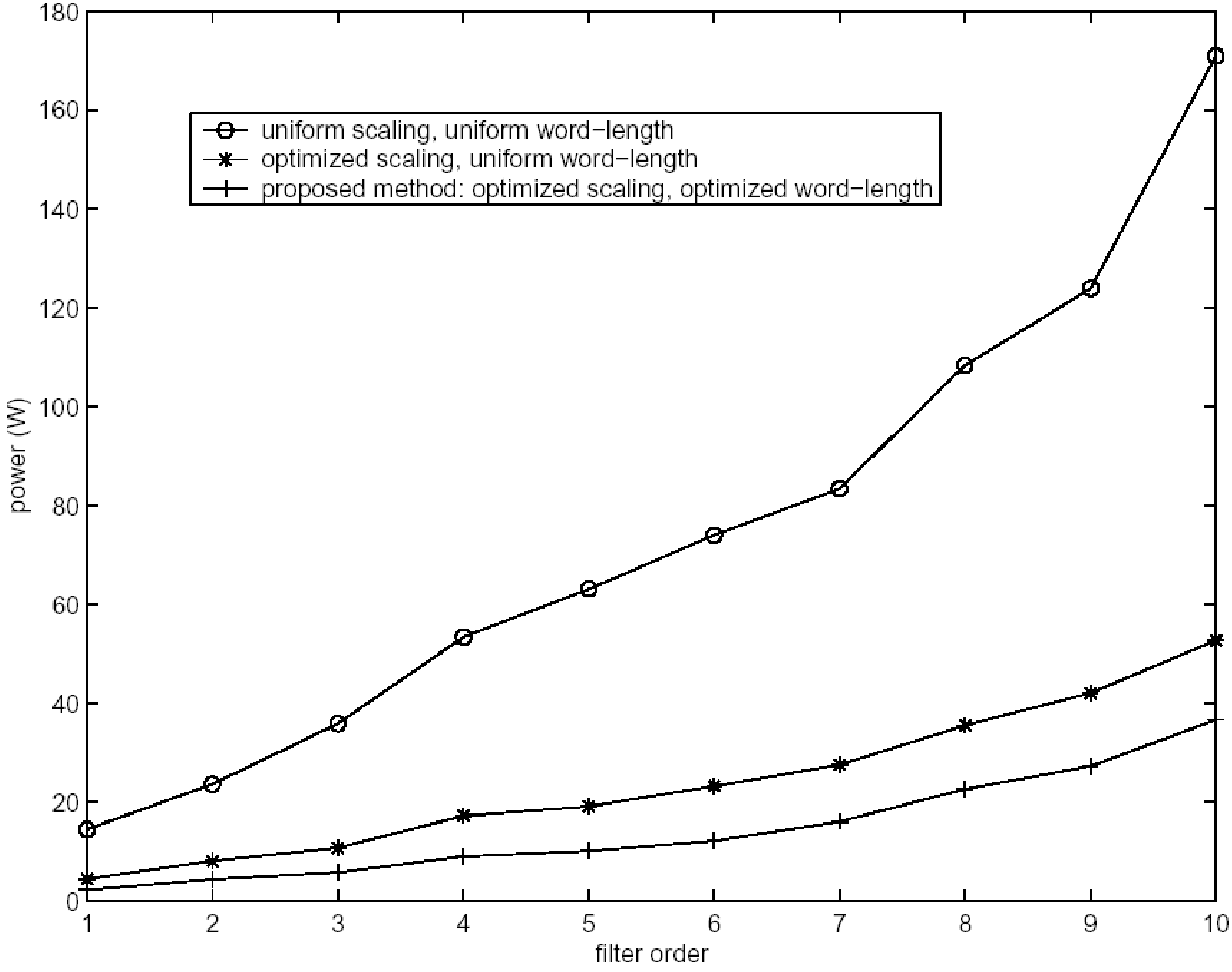
Results

- 90 filters between 1st and 10th order constructed
- Three designs synthesized
  - Uniform scaling and optimum uniform word-length
  - Scaling individually optimized for each signal and optimum uniform word-length
  - Individually optimized scaling and word-length
- Lower bound of output fixed at 34dB

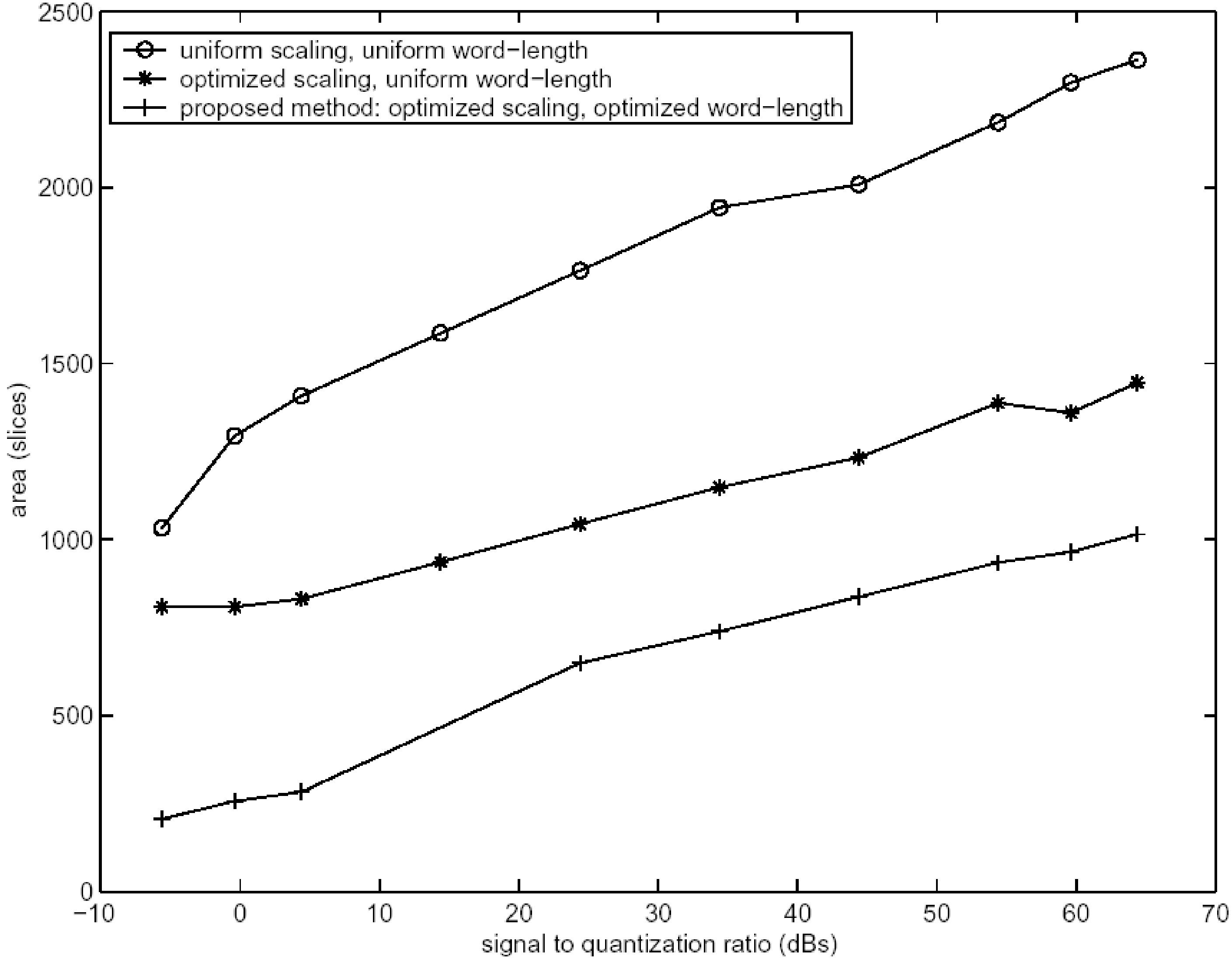
Area vs. filter order



Power vs. filter order



Area vs. SNR bound



Power vs. SNR bound