Spring 4-2015

Analog Filter Design System using Mathematica and System Modeler

Tatjana Samardzic

UNIVERSITY OF SOUTHERN MAINE
PORTLAND • GORHAM • LEWISTON • ONLINE

USM engineering

Analog Filter Design System using *Mathematica* and *System Modeler*

Author: Tatjana Samardzic Advisor: Prof. Dr. Mariusz Jankowski
Department of Engineering

OVERVIEW

Analog filters are an essential part of modern electronics; however, their design, realization and verification can be arduous and time consuming. This project describes a *Mathematica* and *SystemModeler* platform for automated, fast analog filter design and simulation. The platform consists of two key components:

I. *SystemModeler* library of first and second order low-pass, high-pass, band-pass and band-stop filters based on some of the most popular filter topologies: Sallen-Key, Tow-Thomas, State Variable, Multiple Feedback, Bainter and Boctor circuits and
II. *Mathematica* package based on Butterworth, Chebyshev and elliptic frequency response approximations and different filter topologies.

For a given set of design specifications, the program returns overall transfer function, number of first and second order filter stages to be cascaded, and transfer function, gain, cutoff frequency and quality factor for each stage. For any user-selected filter topology, the program builds the circuit model, returns values of resistors and capacitors for each stage, applies them to the model and simulates the circuit to verify the time and frequency responses of the model. We therefore, have a fast and effective means of filter realization and verification, with programmatic control of simulation and filter components.

SYSTEM COMPONENTS AND FUNCTIONALITY

The library named "Filters" is a custom made *SystemModeler* library of first and second order active low-pass, high-pass, band-pass and band-stop filters based on some of the most popular filter topologies: Sallen-Key, Tow-Thomas, State Variable, Multiple Feedback, Bainter and Boctor circuits (Figure 1 and Figure 2.)

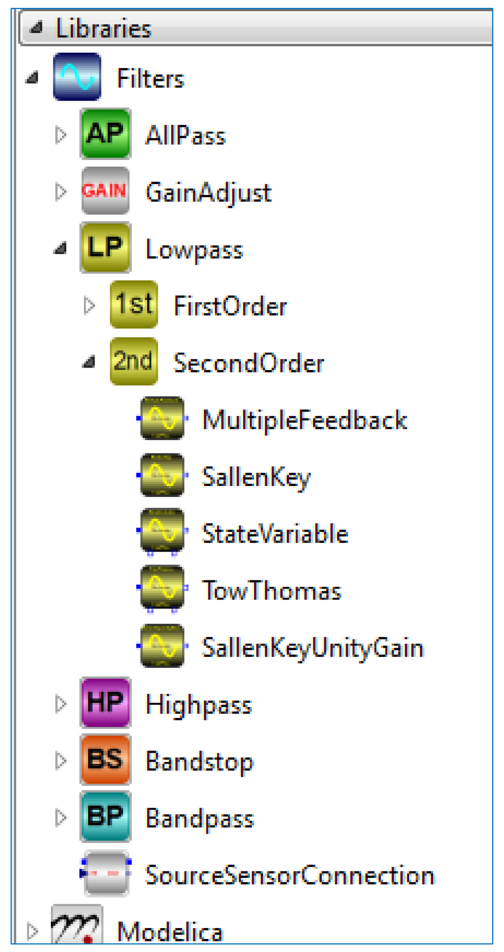


Figure 1. *Screenshot of SystemModeler custom made library "Filters"*

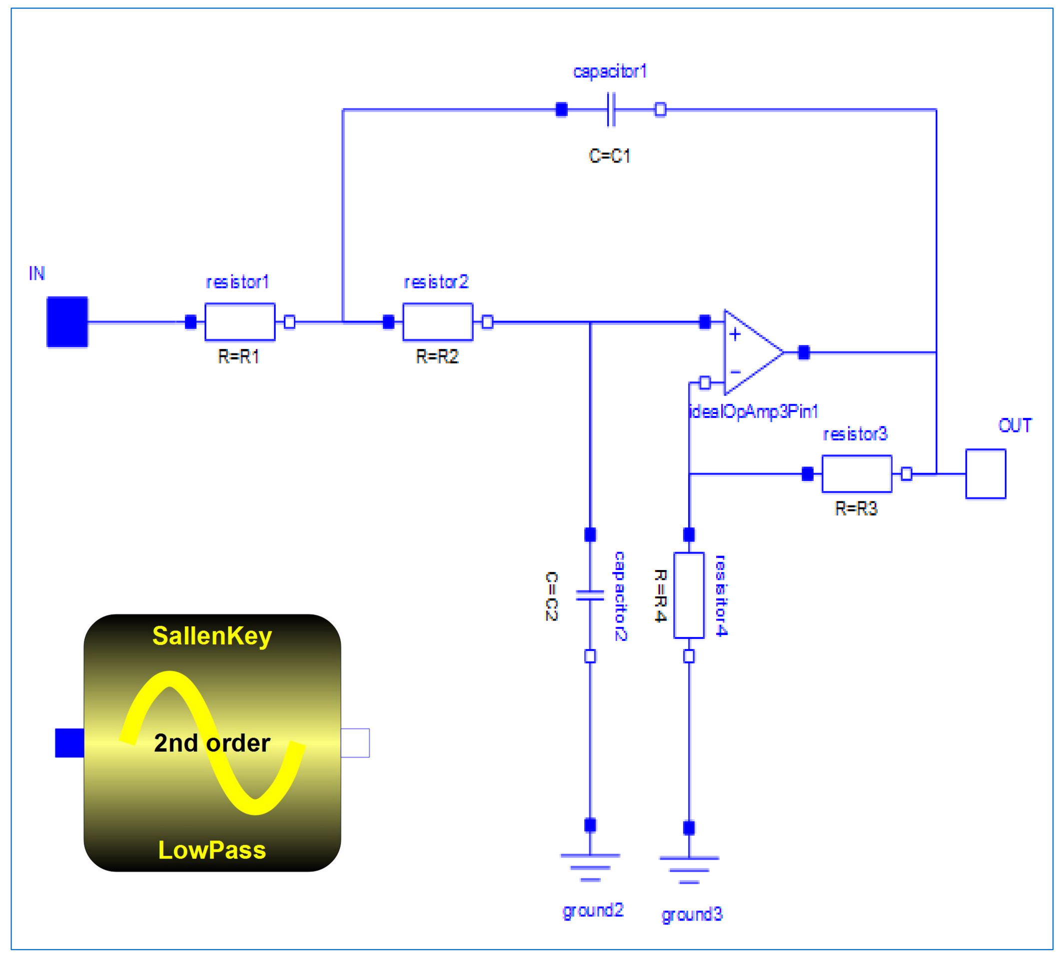
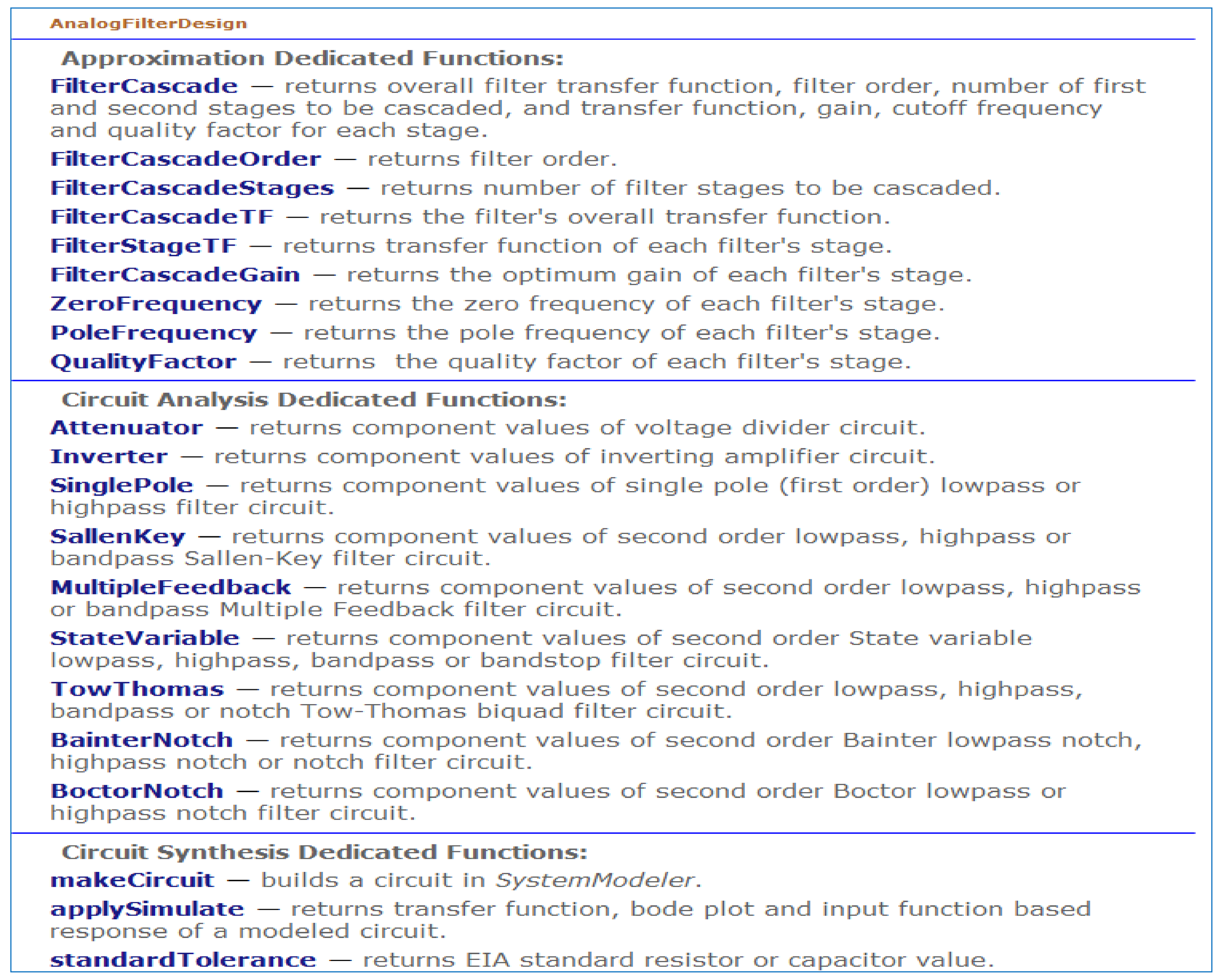


Figure 2. *Diagram view (top image) and icon view (left lower corner) of low-pass second order Sallen-Key circuit.*

AnalogFilterDesign application is a Mathematica based package written in Wolfram Language that consists of three sets of function dedicated to solving three fundamental problems of filter design, namely approximation problem, circuit analysis and circuit synthesis.(Figure 3).


AnalogFilterDesign

Approximation Dedicated Functions:
FilterCascade — returns overall filter transfer function, filter order, number of first and second stages to be cascaded, and transfer function, gain, cutoff frequency and quality factor for each stage.
FilterCascadeOrder — returns filter order.
FilterCascadeStages — returns number of filter stages to be cascaded.
FilterCascadeTF — returns the filter's overall transfer function.
FilterStageTF — returns transfer function of each filter's stage.
FilterCascadeGain — returns the optimum gain of each filter's stage.
ZeroFrequency — returns the zero frequency of each filter's stage.
PoleFrequency — returns the pole frequency of each filter's stage.
QualityFactor — returns the quality factor of each filter's stage.

Circuit Analysis Dedicated Functions:
Attenuator — returns component values of voltage divider circuit.
Inverter — returns component values of inverting amplifier circuit.
SinglePole — returns component values of single pole (first order) lowpass or highpass filter circuit.
SallenKey — returns component values of second order lowpass, highpass or bandpass Sallen-Key filter circuit.
MultipleFeedback — returns component values of second order lowpass, highpass or bandpass Multiple Feedback filter circuit.
StateVariable — returns component values of second order State variable lowpass, highpass, bandpass or bandstop filter circuit.
TowThomas — returns component values of second order lowpass, highpass, bandpass or notch Tow-Thomas biquad filter circuit.
BainterNotch — returns component values of second order Bainter lowpass notch, highpass notch or notch filter circuit.
BoctorNotch — returns component values of second order Boctor lowpass or highpass notch filter circuit.

Circuit Synthesis Dedicated Functions:
makeCircuit — builds a circuit in *SystemModeler*.
applySimulate — returns transfer function, bode plot and input function based response of a modeled circuit.
standardTolerance — returns EIA standard resistor or capacitor value.

Figure 3. *List of functions included in AnalogSystemDesign package*

LOWPASS 5^{th} ORDER ELLIPTIC FILTER – DESIGN EXAMPLE

Design specification : a_p=0.5dB, a_s=66.dB, ωp=2000 rad/s, ωs= 4000 rad/s, gain = 0 dB

spec={{"Lowpass",{2000.,4000.},{0.5,66.}},EllipticFilterModel,0};

Filter order, Number of Stages, Transfer Functions, Gain Distribution, Zero and Pole Frequency, Quality Factor:

FilterCascade[spec]

$$\left\{5,\ 3,\ \left(\frac{1.\ (-6.82521\times10^{15}-5.54497\times10^{8}\,s^2-9.32023\,s^4)}{6.82521\times10^{15}+1.30961\times10^{13}\,s+(1.08605\times10^{10}-4.57216\times10^{-8}\,i)\,s^2+7.89796\times10^{6}\,s^3+2331.36\,s^4+s^5}\right)^{\mathcal{T}},\right.$$

$$\left\{\left(\frac{785.491}{785.491+s}\right)^{\mathcal{T}},\ \left(\frac{0.0499016\,(4.20993\times10^{7}+s^2)}{2.10082\times10^{6}+1161.02\,s+s^2}\right)^{\mathcal{T}},\ \left(\frac{0.237778\,(1.73946\times10^{7}+s^2)}{4.13604\times10^{6}+384.851\,s+s^2}\right)^{\mathcal{T}}\right\},$$

$$\left.\{1.,\ 0.0499016,\ 0.237778\},\ \{\text{none},\ 6488.40,\ 4170.68\},\ \{785.491,\ 1449.42,\ 2033.73\},\ \{0.500000,\ 1.24840,\ 5.28446\}\right\}$$

Based on program output above, Bode plot is obtained for each stage and the overall transfer function (Figure 4.)

Filter Components:

sp=SinglePole[{"Lowpass",FilterCascadeGain[spec,1],PoleFrequency[spec,1],3.3*10^{-7}}]

{{SinglePole1.Rin->3857.85,SinglePole1.Rf->3857.85,SinglePole1.C1->3.3*10^{-7}}}

sv=StateVariable[{"Bandstop",FilterCascadeGain[spec][[2;;3]],ZeroFrequency[spec][[2;;3]],PoleFrequency[spec][[2;;3]],QualityFactor[spec][[2;;3]],3.3*10^{-7},1000.}]

{{StateVariable1.R1->1000.,StateVariable1.R2->1000.,StateVariable1.R3->1000.,StateVariable1.R4->2090.7,StateVariable1.R5->2090.7,StateVariable1.R6->2745.21,StateVariable1.R7->1000.,StateVariable1.R8->20039.4,StateVariable1.R9->1000.,StateVariable1.R10->1000.,StateVariable1.C1->3.3*10^{-7},StateVariable1.C2->3.3*10^{-7}},{StateVariable2.R1->1000.,StateVariable2.R2->1000.,StateVariable2.R3->1000.,StateVariable2.R4->1490.02,StateVariable2.R5->1490.02,StateVariable2.R6->14853.4,StateVariable2.R7->1000.,StateVariable2.R8->4205.61,StateVariable2.R9->1000.,StateVariable2.R10->1000.,StateVariable2.C1->3.3*10^{-7},StateVariable2.C2->3.3*10^{-7}}}

After the component values are obtained,, circuit is synthesized , simulated and verified using functions makeCircuit and applySimulate functions- circuit is simulated for step and chirp input functions. (see Figure 5, 6, and 7).

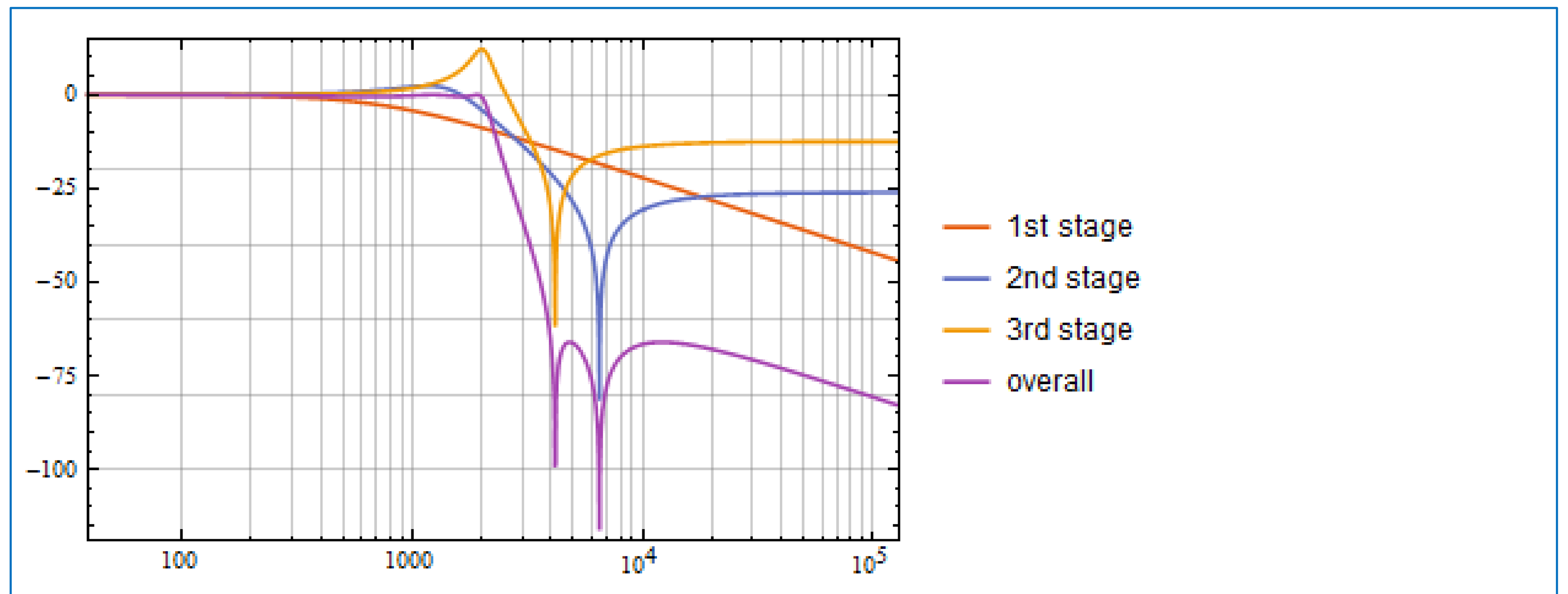


Figure 4. *Bode Plot of the filter and each filter stage.*

Circuit Synthesis

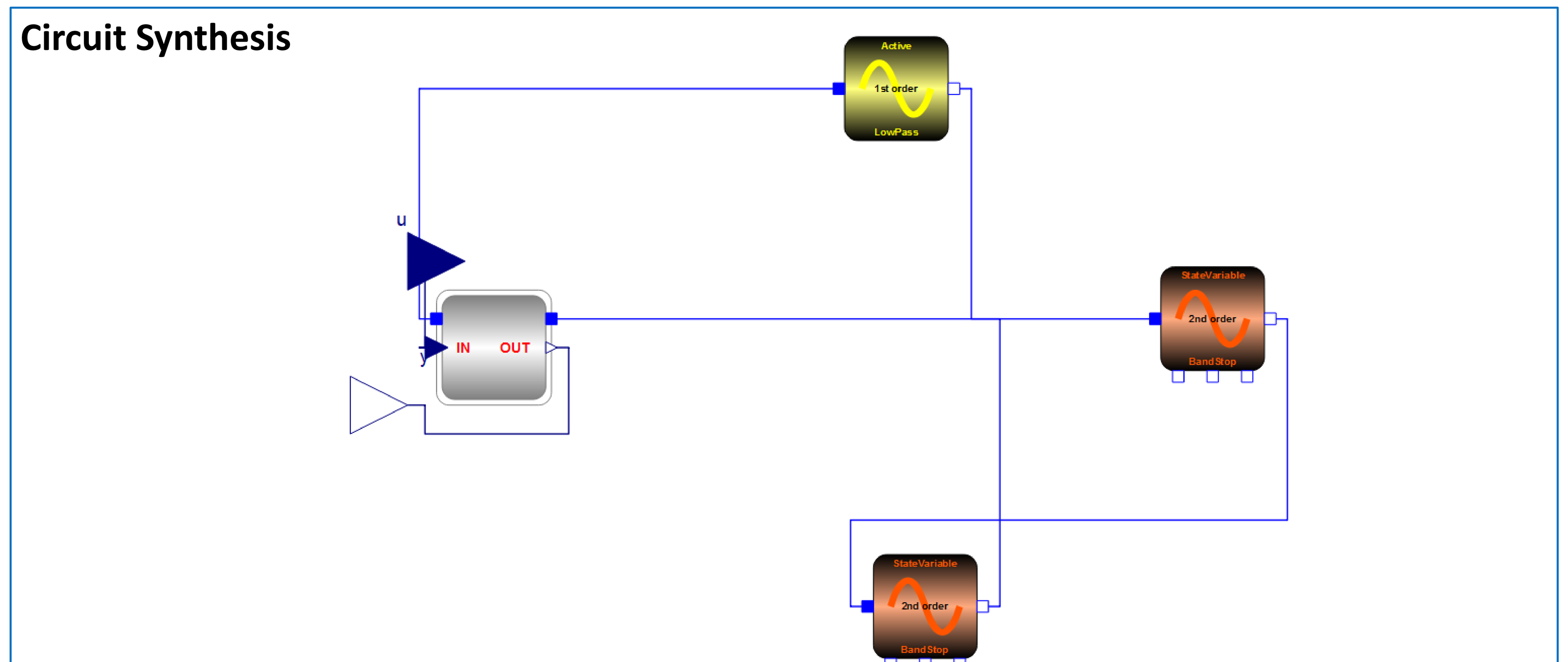


Figure 5. *Schematic of low-pass 5th order elliptic filter created in SystemModeler*

Simulation

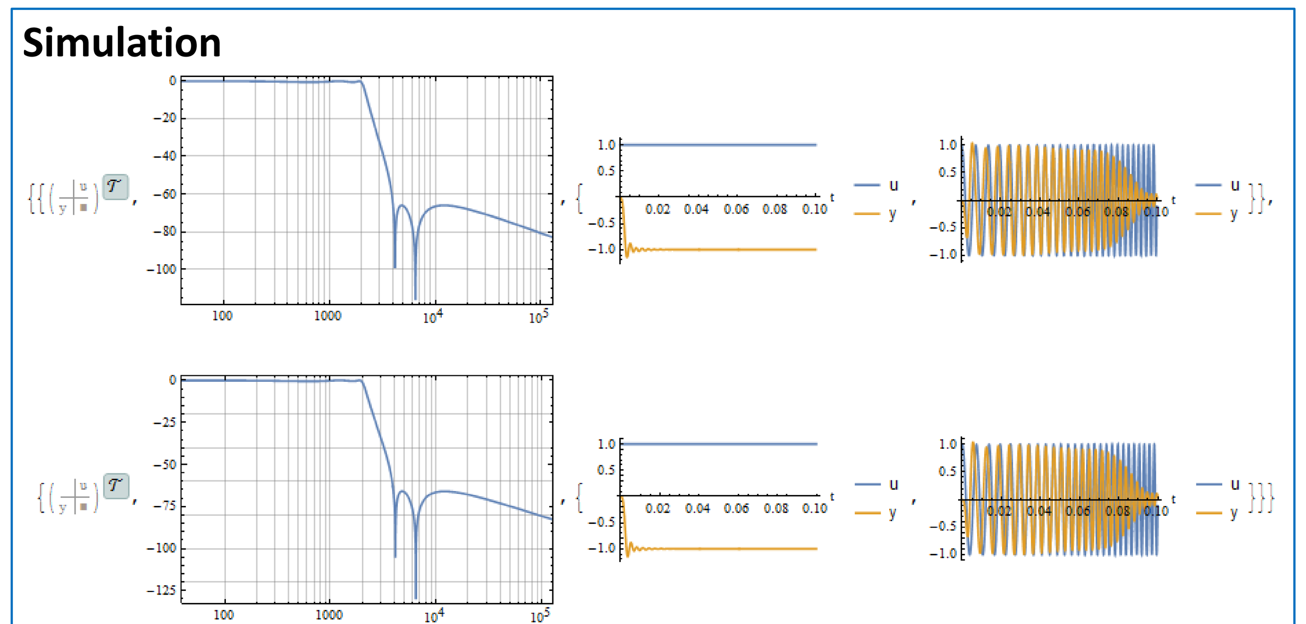


Figure 6. *Transfer functions, Bode plot, Step and Chirp response of designed circuit top row - exact component values, bottom row - 1% standard tolerance components).*

Verification

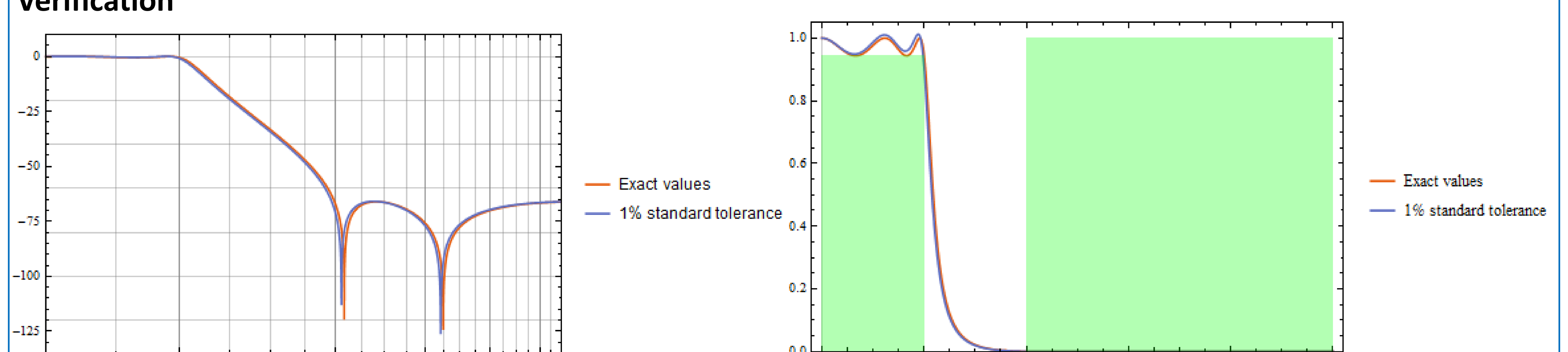


Figure 7. *Magnitude response(log scale left, linear scale right) of both circuits (exact and 1% standard tolerance values). Green rectangles define design specifications boundaries.*