Table of Contents

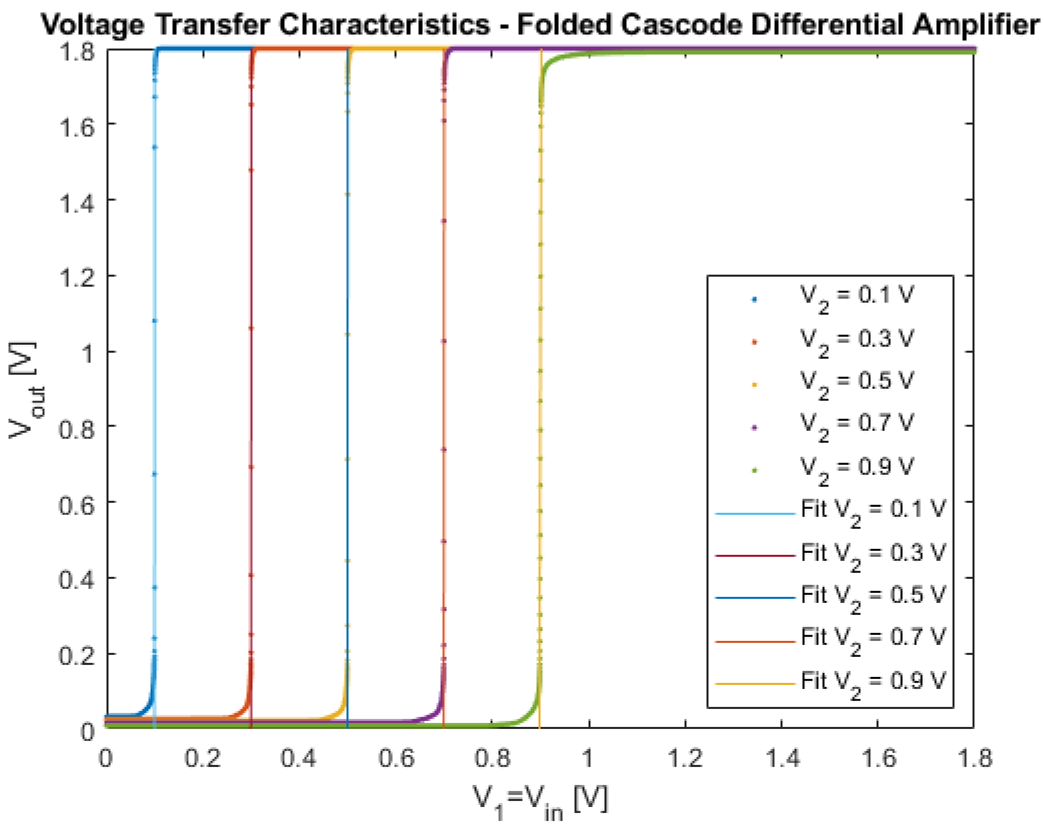

```
% Folded Cascode Differential Amplifier Analysis
% @author: Thomas Jagielski
% March 2021
```

Voltage Transfer Characteristics

order of columns = [Vout, V1, V2]

```
vtc_data = csvread('./folded_cascode_vtc.csv',1);

% Separate Data
vtc_0_1 = vtc_data(1:18001,:);
vtc_0_3 = vtc_data(18002:36003,:);
vtc_0_5 = vtc_data(36004:54005,:);
vtc_0_7 = vtc_data(54006:72007,:);
vtc_0_9 = vtc_data(72008:end,:);

% Lines of best fit
v_in = linspace(0,1.8);

vtc_0_1_fit_params =
 polyfit(vtc_data(998:1002,2),vtc_0_1(998:1002,1),1);
vtc_0_1_fit_line = vtc_0_1_fit_params(1) * v_in +
 vtc_0_1_fit_params(2);
vtc_0_3_fit_params =
 polyfit(vtc_0_3(2997:3001,2),vtc_0_3(2997:3001,1),1);
vtc_0_3_fit_line = vtc_0_3_fit_params(1) * v_in +
 vtc_0_3_fit_params(2);
vtc_0_5_fit_params =
 polyfit(vtc_0_5(4995:5002,2),vtc_0_5(4995:5002,1),1);
vtc_0_5_fit_line = vtc_0_5_fit_params(1) * v_in +
 vtc_0_5_fit_params(2);
vtc_0_7_fit_params =
 polyfit(vtc_0_7(6993:7001,2),vtc_0_7(6993:7001,1),1);
vtc_0_7_fit_line = vtc_0_7_fit_params(1) * v_in +
 vtc_0_7_fit_params(2);
vtc_0_9_fit_params =
 polyfit(vtc_0_9(8970:9012,2),vtc_0_9(8970:9012,1),1);
```

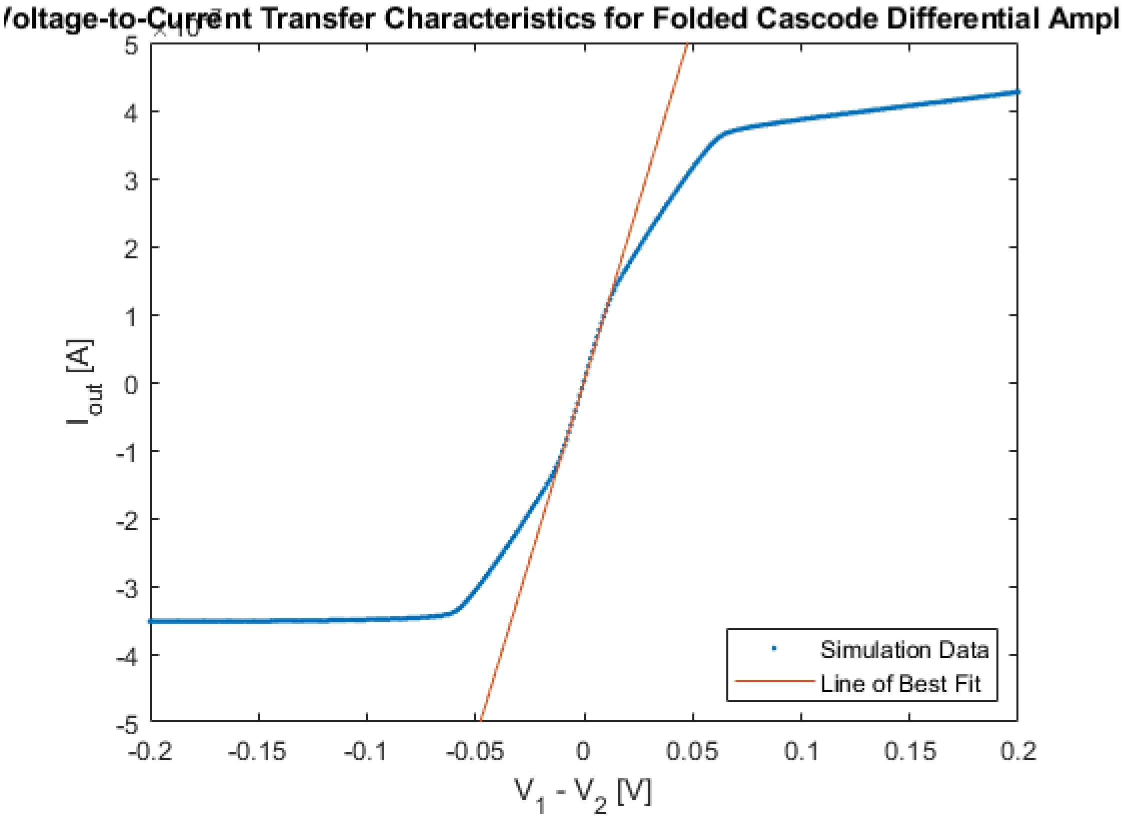
```
vtc_0_9_fit_line = vtc_0_9_fit_params(1) * v_in +
 vtc_0_9_fit_params(2);

% Plot Data
figure
plot(vtc_0_1(:,2), vtc_0_1(:,1),'.')
hold on
plot(vtc_0_3(:,2), vtc_0_3(:,1),'.')
plot(vtc_0_5(:,2), vtc_0_5(:,1),'.')
plot(vtc_0_7(:,2), vtc_0_7(:,1),'.')
plot(vtc_0_9(:,2), vtc_0_9(:,1),'.')
plot(v_in,vtc_0_1_fit_line)
plot(v_in,vtc_0_3_fit_line)
plot(v_in,vtc_0_5_fit_line)
plot(v_in,vtc_0_7_fit_line)
plot(v_in,vtc_0_9_fit_line)
hold off
axis([0 1.8 0 1.8])
title('Voltage Transfer Characteristics - Folded Cascode Differential
 Amplifier')
xlabel('V_{1}=V_{in} [V]')
ylabel('V_{out} [V]')
legend('V_{2} = 0.1 V','V_{2} = 0.3 V','V_{2} = 0.5 V',...
    'V_{2} = 0.7 V','V_{2} = 0.9 V', 'Fit V_{2} = 0.1 V',...
    'Fit V_{2} = 0.3 V','Fit V_{2} = 0.5 V','Fit V_{2} = 0.7 V',...
    'Fit V_{2} = 0.9 V','location', 'Southeast')
```



Voltage-to-Current Transfer Chracteristics

order of columns = [Iout, V1, V2, Vout]

```
voltage_to_current_data = csvread('./
folded_cascode_voltage_to_current.csv',1);

% Separate Data
%voltage_to_current_data.Iout = voltage_to_current_data(:,1);
%voltage_to_current_data.V1 = voltage_to_current_data(:,2);
%voltage_to_current_data.V2 = voltage_to_current_data(:,3);
%voltage_to_current_data.Vout = voltage_to_current_data(:,4);
voltage_to_current_input_diff = voltage_to_current_data(:,2) -...
                                    voltage_to_current_data(:,3);


voltage_to_current_best_fit_parameters =
 polyfit(voltage_to_current_input_diff(19000:20000,:),voltage_to_current_data(1900
best_fit_line_voltage_to_current =
 voltage_to_current_best_fit_parameters(1)*...
    voltage_to_current_input_diff
+voltage_to_current_best_fit_parameters(2);

incremental_transconductance =
 voltage_to_current_best_fit_parameters(1);

figure
plot(voltage_to_current_input_diff(1:100:end),voltage_to_current_data((1:100:end),
title('Voltage-to-Current Transfer Characteristics for Folded Cascode
 Differential Amplifier')
xlabel('V_{1} - V_{2} [V]')
ylabel('I_{out} [A]')
hold on
plot(voltage_to_current_input_diff,best_fit_line_voltage_to_current)
hold off
legend('Simulation Data','Line of Best Fit','Location','Southeast')
axis([-0.2 0.2 -0.5e-6 0.5e-6])
```



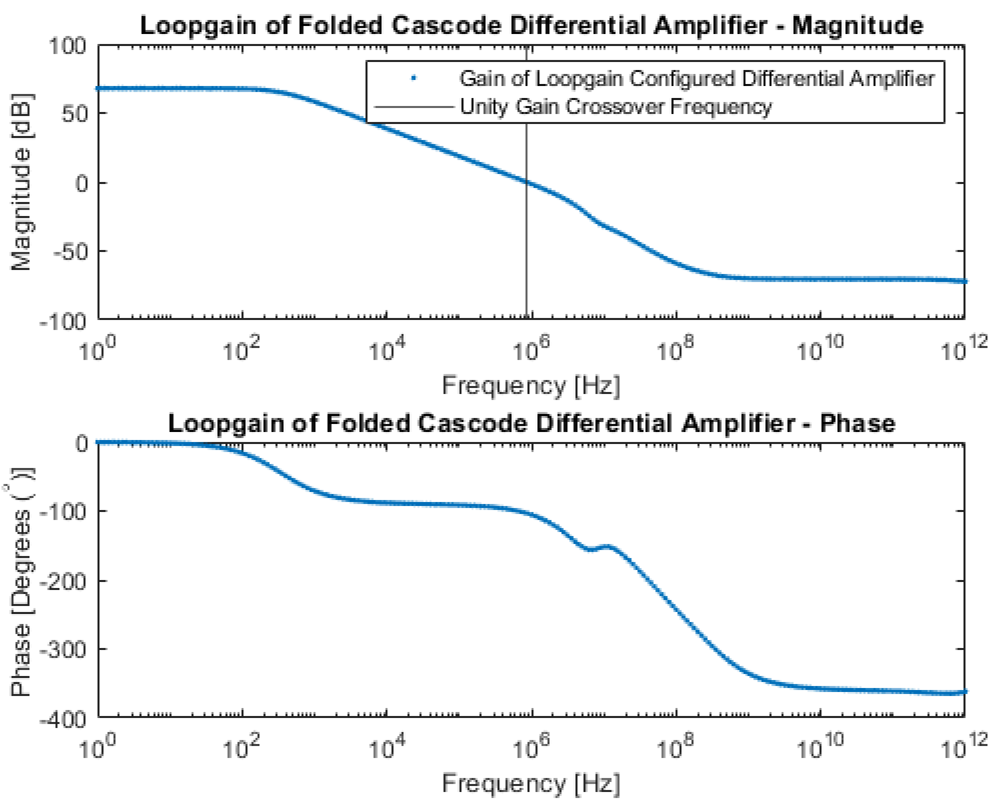
Loopgain

order of columns = Frequency [Hz], Magnitude [dB], Phase [Degrees]

```
loopgain_data = csvread('./folded_cascode_loopgain.csv',1);

dc_gain = 20 * log10(2393.3);

gain_percent_error = (abs(max(loopgain_data(:,2)) - dc_gain) /
 max(loopgain_data(:,2))) * 100;

unity_gain_crossover_frequency_slope = (-0.48941 - 0.54699)/
(891250.938 - 794328.235);
unity_gain_crossover_frequency = 891250.938 + (0.48941 /
 unity_gain_crossover_frequency_slope);

figure
subplot(2,1,1);
semilogx(loopgain_data(:,1),loopgain_data(:,2),'.')
xline(unity_gain_crossover_frequency)
title('Loopgain of Folded Cascode Differential Amplifier - Magnitude')
xlabel('Frequency [Hz]')
ylabel('Magnitude [dB]')
legend('Gain of Loopgain Configured Differential Amplifier','Unity
 Gain Crossover Frequency')
```

```
subplot(2,1,2);
semilogx(loopgain_data(:,1),loopgain_data(:,3),'.')
title('Loopgain of Folded Cascode Differential Amplifier - Phase')
xlabel('Frequency [Hz]')
ylabel('Phase [Degrees (^{\circ})]')
```

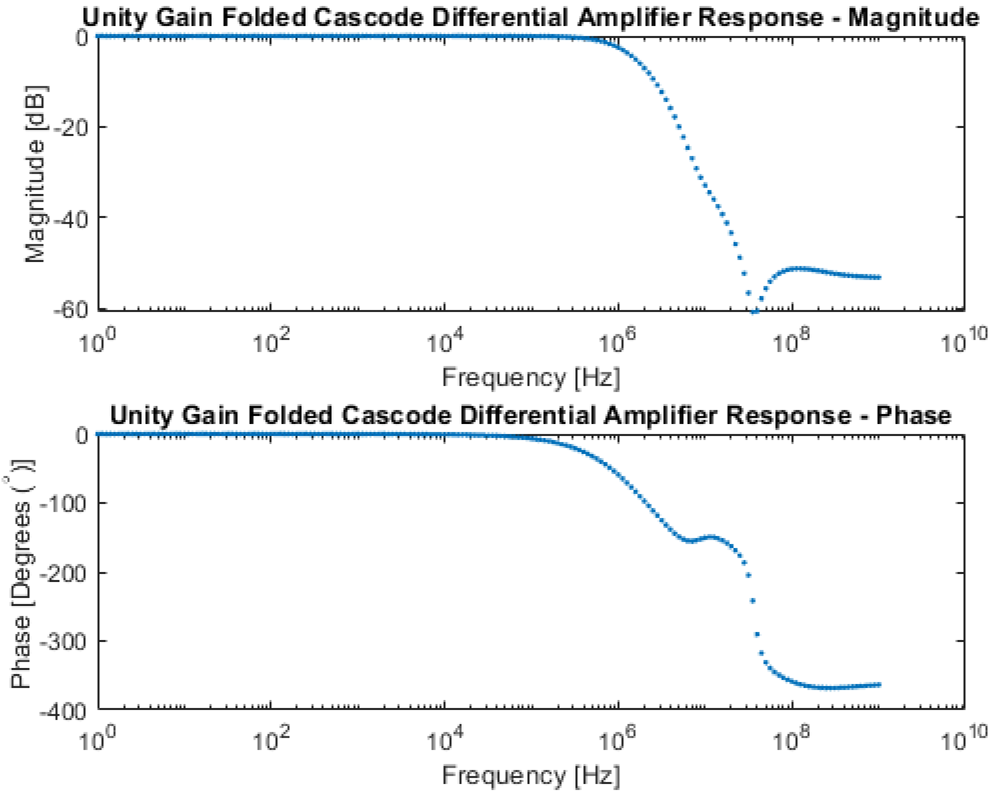
```
ans =

  ConstantLine with properties:

    InterceptAxis: 'x'
            Value: 8.4548e+05
            Color: [0.1500 0.1500 0.1500]
        LineStyle: '-'
        LineWidth: 0.5000
            Label: ''
      DisplayName: ''

  Use GET to show all properties
```



Unity-Gain Follower Freqency Response

order of columns = Frequency [Hz], Magnitude [dB], Phase [Degrees]

```
unity_gain_data = csvread('./folded_cascode_unity_gain.csv',1);

figure
subplot(2,1,1);
semilogx(unity_gain_data(:,1),unity_gain_data(:,2),'.')
title('Unity Gain Folded Cascode Differential Amplifier Response -
 Magnitude')
xlabel('Frequency [Hz]')
ylabel('Magnitude [dB]')

subplot(2,1,2);
semilogx(unity_gain_data(:,1),unity_gain_data(:,3),'.')
title('Unity Gain Folded Cascode Differential Amplifier Response -
 Phase')
xlabel('Frequency [Hz]')
ylabel('Phase [Degrees (^{\circ})]')

unity_gain_corner_frequency = 1050000;

percent_error_corner_frequency = (abs(unity_gain_corner_frequency -...
    900000) / unity_gain_corner_frequency) * 100;
```



Small-Signal Step Response

order of columns = time, vout, v1

```
small_signal_data = csvread('./folded_cascode_small-
signal_step.csv',1);

figure
plot(small_signal_data(:,1),small_signal_data(:,2),'.')
hold on
plot(small_signal_data(:,1), small_signal_data(:,3))
hold off
title('Small-Signal Step Response')
xlabel('Time [S]')
ylabel('V_{out} [V]')
legend('Simulated Data for V_{out}','Step Input (V_{in})')
```

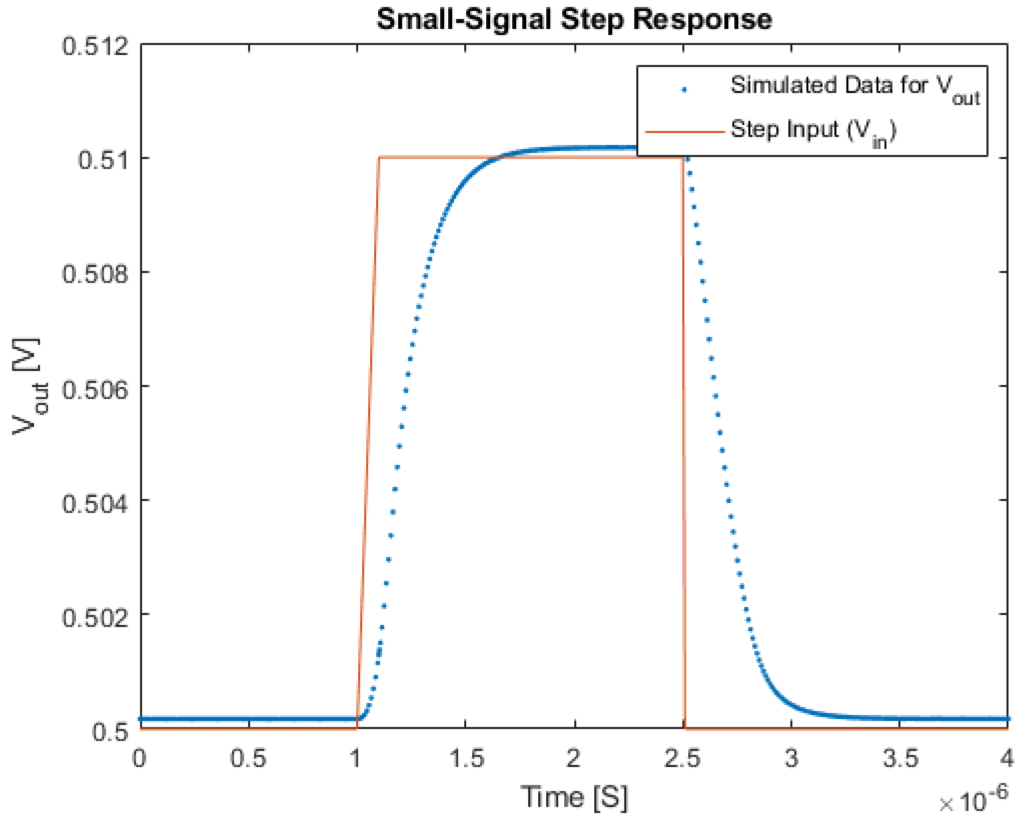
Small Signal Step in Up-going %%%

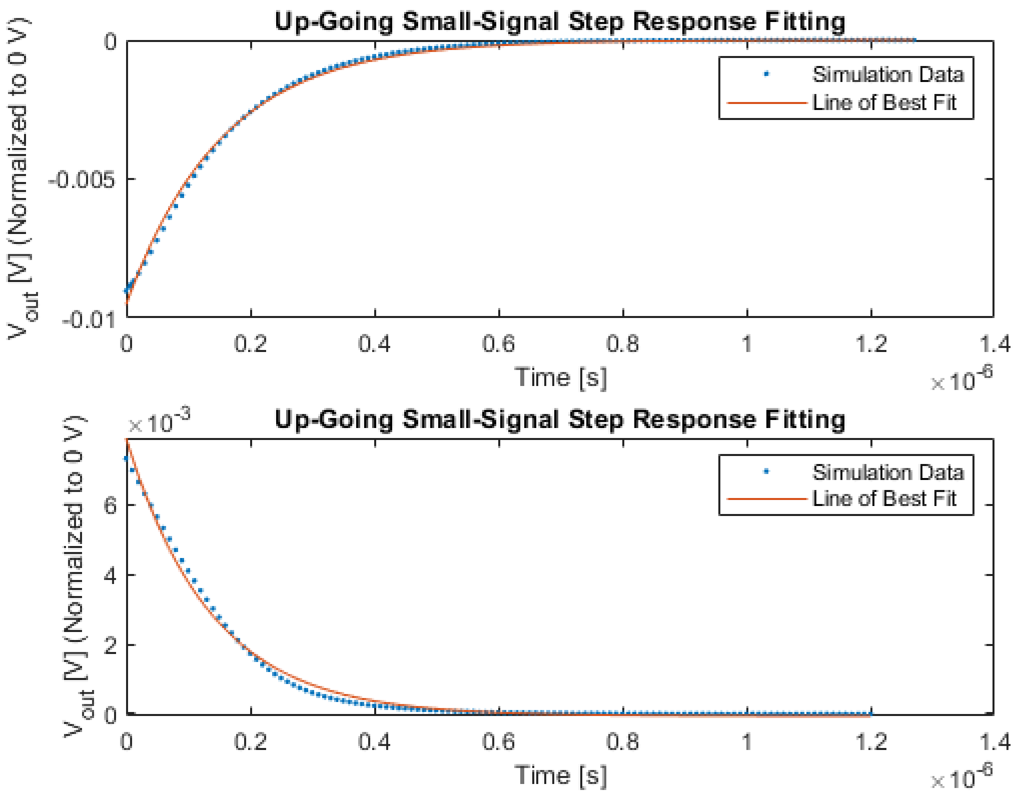
```
small_signal_tau_up_time = small_signal_data(120:250,1) -
 min(small_signal_data(120:250,1));
small_signal_tau_up_Vout = small_signal_data(120:250,2) -
 max(small_signal_data(120:250,2));

% From CFTOOL
a = -0.009524;
tau_up = 1.546e-7;

figure
subplot(2,1,1);
plot(small_signal_tau_up_time, small_signal_tau_up_Vout,'.')
hold on
plot(small_signal_tau_up_time, a*exp(-small_signal_tau_up_time/
tau_up))
hold off
title('Up-Going Small-Signal Step Response Fitting')
xlabel('Time [s]')
ylabel('V_{out} [V] (Normalized to 0 V)')
legend('Simulation Data', 'Line of Best Fit')
```

Small Signal Step in Down-going %%%

```
small_signal_tau_down_time = small_signal_data(280:400,1) -
 min(small_signal_data(280:400,1));
small_signal_tau_down_Vout = small_signal_data(280:400,2) -
 min(small_signal_data(280:400,2));

% From CFTOOL
a = 0.007959;
tau_down = 1.372e-7;
c = -5.859e-5;

subplot(2,1,2);
plot(small_signal_tau_down_time, small_signal_tau_down_Vout,'.')
hold on
plot(small_signal_tau_down_time, a*exp(-small_signal_tau_down_time/
tau_down)+c)
```

```
hold off
title('Up-Going Small-Signal Step Response Fitting')
xlabel('Time [s]')
ylabel('V_{out} [V] (Normalized to 0 V)')
legend('Simulation Data', 'Line of Best Fit')

% Percent Error for tau
percent_error_tau = (abs(tau_up - tau_down) / tau_up) * 100;
```



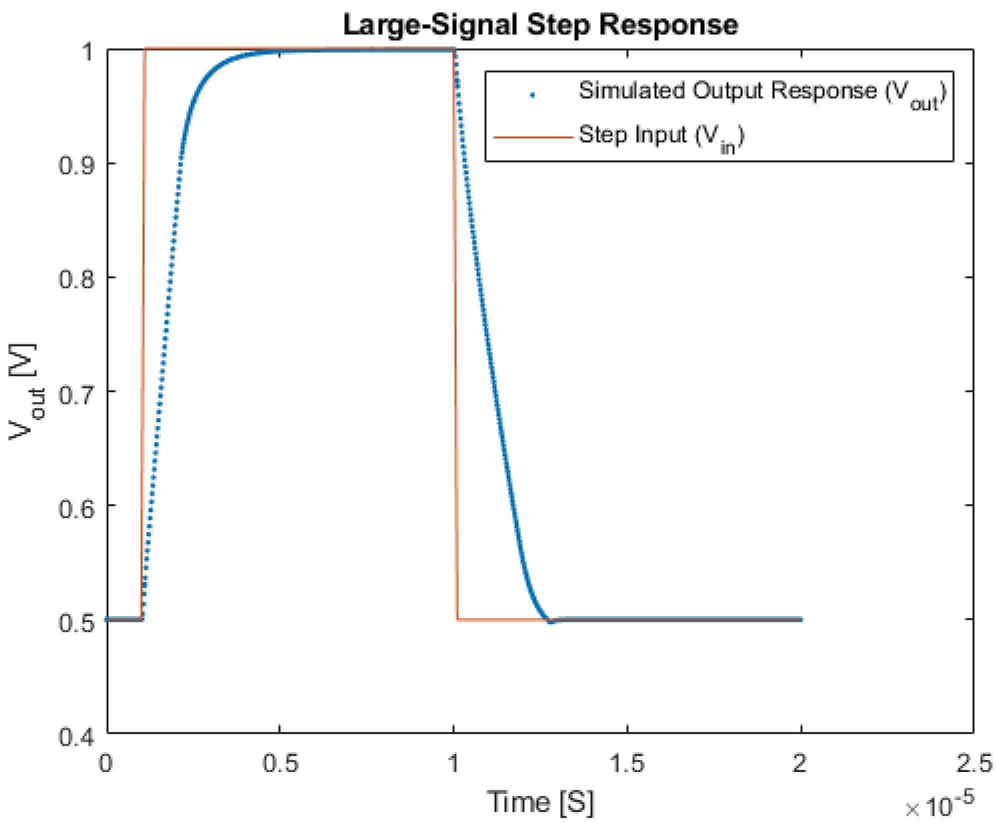


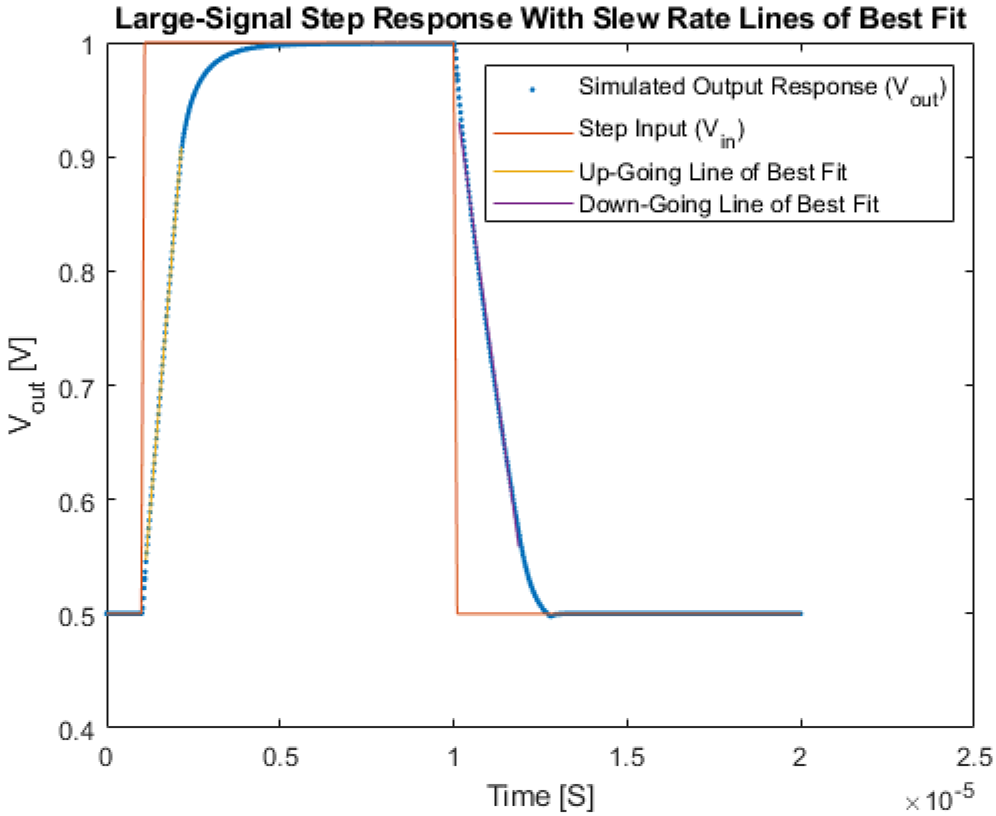
Large-Amplitude Step Response

order of columns = time, vout, v1

```
large_signal_data = csvread('./
folded_cascode_large_amplitude_step.csv',1);

figure
plot(large_signal_data(1:2:end,1),large_signal_data(1:2:end,2),'.')
hold on
plot(large_signal_data(:,1), large_signal_data(:,3))
hold off
title('Large-Signal Step Response')
xlabel('Time [S]')
ylabel('V_{out} [V]')
legend('Simulated Output Response (V_{out})', 'Step Input (V_{in})')

large_step_up_going_best_fit_parameters =
 polyfit(large_signal_data(127:229,1),large_signal_data(127:229,2),1);
large_step_up_going_fit =
 large_step_up_going_best_fit_parameters(1)*...

 large_signal_data(127:229,1)+large_step_up_going_best_fit_parameters(2);

large_step_down_going_best_fit_parameters =
 polyfit(large_signal_data(1035:1207,1),large_signal_data(1035:1207,2),1);
```

```
large_step_down_going_fit =
 large_step_down_going_best_fit_parameters(1)*...

 large_signal_data(1035:1207,1)+large_step_down_going_best_fit_parameters(2);

slew_up = large_step_up_going_best_fit_parameters(1);
slew_down = large_step_down_going_best_fit_parameters(1);

slew_percent_error = ((slew_up - slew_down) / slew_up) * 100;

figure
plot(large_signal_data(1:2:end,1),large_signal_data(1:2:end,2),'.')
hold on
plot(large_signal_data(:,1), large_signal_data(:,3))
plot(large_signal_data(127:229,1), large_step_up_going_fit, '-')
plot(large_signal_data(1035:1207,1), large_step_down_going_fit, '-')
hold off
title('Large-Signal Step Response With Slew Rate Lines of Best Fit')
xlabel('Time [S]')
ylabel('V_{out} [V]')
legend('Simulated Output Response (V_{out})', 'Step Input
 (V_{in})', 'Up-Going Line of Best Fit', 'Down-Going Line of Best
 Fit')
```





Published with MATLAB® R2019a