Analoge Schaltungstechnik

Mirco Meiners Tim Ziemann

2025-06-24

Table of contents

Unsyllabus 1

I. Seminar 3

1. Einleitung und Übersicht 5

1.1. Lernziele des Moduls . . . 5
1.2. Voraussetzungen des Moduls . . . 6
1.3. Scientific Computing . . . 6
1.4. EDA Tools . . . 6
1.5. OS-Tools . . . 6
1.6. Code Editors . . . 7
1.7. Data Science . . . 7
1.8. Publish Computational Content . . . 7
1.9. Are you writing or TeXing? . . . 7
1.10. LaTeX Editoren . . . 8
1.11. Bibliography and LaTeX . . . 8
1.12. Labor . . . 8
1.13. Schaltungsanalyse/-entwurf $\neq$ Black Magic . . . 8
1.14. Schöne neue Welt . . . 9
1.15. From Sand to Silicon (Infineon, Dresden) . . . 9
1.16. FinFET (Intel) . . . 9
1.17. TSMC Fab (Next Gen 7/5 nm) . . . 9
1.18. Es war einmal … . . . 10
1.19. Damals und heute . . . 10
1.20. Systemhierarchie . . . 11
1.21. System Assembly . . . 12
1.22. Schnittstellen zur Aussenwelt . . . 13
1.23. Meeting mit einem System . . . 13
1.24. System in a Package (SiP) . . . 14
1.25. Backend Phasen . . . 14
1.26. Sie werden Experte . . . 14
1.27. Evolution von Produkten . . . 15
1.28. EDA Kompetenz . . . 15
1.29. Design-/Entwurfsmethodik . . . 16
1.30. Nachhaltige Elektronik … . . . 16
1.31. Warum es sicht lohnt … . . . 16

1.32. Und ab an den Strand ... 16

Literaturverzeichnis 17

II. Labor 19

2. Experiment 4 - Design of Analog Filters 21
2.1. Brief theory and motivation 21
2.1.1. Transfer funtions of active filters 21
2.2. Specification 23
2.3. Measurements to be taken 24
2.3.1. Steady-state response 24
2.3.2. Frequency response 24
2.4. What you should submit 24
2.4.1. Simulation with KiCad (ngspice) 24
2.4.2. Measurements with Redpitaya STEMlab 25
2.5. Exercise Set 4 30
2.5.1. 3rd order butterworth 30
2.5.2. Notch filter 30

3. Validation of Transfer Functions 31

4. Mathematical derivation of the transfer functions of the biquad 33
4.1. Calculation of $\frac{V_2}{V_i}$ 33
4.2. Transfer-function $\frac{V_3}{V_i}$: 34
4.3. Transfer-function $\frac{V_4}{V_i}$: 34
4.4. Calculation of the transfer function $\frac{V_1}{V_i}$: 35
4.5. Repeated calculation of the Transfer-function $\frac{V_2}{V_i}$ via V_1: 37
4.6. Conclusion 37

Unsyllabus

Name	Beschreibung
Kursus	M 6.9 Analoge Schaltungen (ANS)
Semester	SoSe25
Dozent	Prof. Dr.-Ing. M. Meiners
Seminar	Di., 09:45 Uhr - 13:00 Uhr, E 507

Part I.

Seminar

1. Einleitung und Übersicht

Im Modul **M 6.9 Analoge Schaltungen (ANS)**, im Bachelorstudiengang (B.Eng.) Elektrotechnik (ET) der Fakultät 4 der Hochschule Bremen (HSB), werden aus der Theorie des Schaltungsentwurfs gewonnene Kenntnisse im Labor experimentell erprobt (ASLK Pro Experimente) und mit KiCAD als eigene PCB-Designs umgesetzt. Für die erste Erprobung von Schaltungen wird das ASLK (Analog System Lab Kit) Pro der Firma MikroElektronika eingesetzt.

Zur Simulation werden Schaltungssimulatoren wie LTSpice und ngspice verwendet.

Des Weiteren kommen die MATLAB Campus-Lizenz und Python als leistungsstarke Instrumente zur Modellierung und Verhaltensanalyse von Filtern- und Verstärkern beim Schaltungsentwurf zum Einsatz.

Für die Charakterisierung der Schaltungen stehen im Labor sechs Meßplätze mit gängigen Gerätschaften wie Quellen, Signalgeneratoren, Oszilloskopen und dem integrierten Meßlabors STEMlab von RedPitaya zur Verfügung.

1.1. Lernziele des Moduls

- Ein tieferes Verständnis für das Verhalten von elektronischen Bauelementen und Baugruppen im analogen Schaltungsentwurf entwickeln und Ausblicke auf weiterführende Kurse im Master bekommen.
- Funktionsprinzipien und Charakterisierung von elektronischen Bauelementen
- Fundamentale integrierte analoge Schaltungen
- Operationsverstärker
- Lernen, Grenzen und Tradeoffs analoger Schaltungen zu bewerten
- Entwickeln eines systematischen Entwurfsstils, auch anwendbar für andere Ingenieursdisziplinen
- Lernen, einen Schaltungssimulator für den Entwurf einzusetzen.
- Technisch-wissenschaftliche Dokumentation
- Konsoliderung der oberen Aspekte in Laborprojekten

1.2. Voraussetzungen des Moduls

- Digitaltechnik
- Grundlagen der Halbleiterbauelemente
- Netzwerk- und Systemtheorie
- Regelungstechnik

1.3. Scientific Computing

- Python (Anaconda)
- Matlab (Campus Lizenz)
- Command-line tools

1.4. EDA Tools

- PCB / System Design
  - LTspice
  - KiCad EDA
  - Altium Designer
  - SiemensEDA PCB tools
  - cadence System Design & Analysis
- IC / Silicon Design
  - IIC-OSIC-TOOLS (open-source)
  - SiemensEDA IC tools
  - cadence IC Design & Verification
  - synopsys silicon design (IC)

1.5. OS-Tools

- Microsoft-Terminal
- Microsoft-PowerShell
- MacOS-Terminal
- Linux/MacOS Shell zsh-tools,
- git (Versionskontrolle)

1.6. Code Editors

- Visual Studio Code
- Spyder IDE
- Thonny (Micro-)Python IDE
- Emacs
- Vim

1.7. Data Science

- File system: Files and directories
- Tabular data: Comma/Tab-Separated-Values (CSV/TSV), Spreadsheet (.xlsx, .ods)
- Special formats, e.g. MATLAB mat, HDF5
- Embedded Databases
  - SQL, z.B. SQlite
  - OLAP, z.B. DuckDB

1.8. Publish Computational Content

- Jupyter-Book
- quarto
  - Manuscripts

1.9. Are you writing or TeXing?

- MikTeX (Windows, MacOS, Linux)
- MacTeX (MacOS)
- TeXLive (Linux)

1.10. LaTeX Editoren

- IDE's
  - TeXStudio
  - TeXMaker
- Collaborative Frameworks
  - Overleaf, Online LaTeX
  - CoCalc - Online LaTeX

1.11. Bibliography and LaTeX

- Citavi im Detail > Titel exportieren > Export nach BibTeX
- RefWorks - Library Guide Univ. Melbourne
- Benutzerdefinierte BibTex-Keys mit Zotero | nerdpause
- JabRef - Library Guide Univ. Melbourne
- EndNote - Library Guide Univ. Melbourne

1.12. Labor

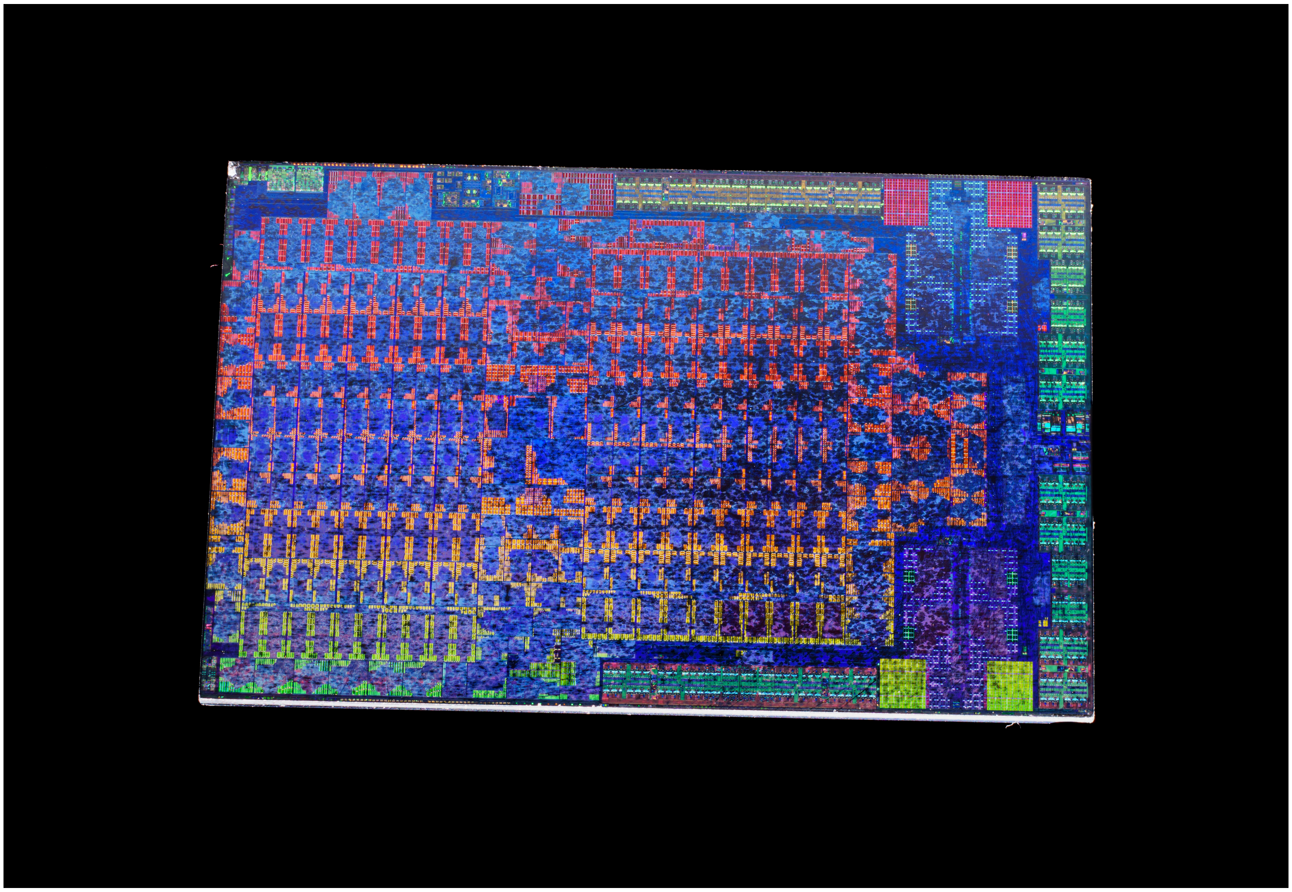
- Anwendung unterschiedlicher Beschreibungsebenen
  - Systemebene (Verhaltensmodell z.B. mit Matlab/Simulink oder Python)
  - Schaltungsebene (SPICE)
  - Charakterisierung (Messungen)
- Analog System Lab Kit – ASLK Pro
- Messautomatisierung Red Pitaya STEMlab 125-14/10

1.13. Schaltungsanalyse/-entwurf $\neq$ Black Magic

Schaltungsanalyse. * ist die Fähigkeit, Schaltungen in handhabbare Teile zu zerlegen * basierend auf einem einfachen, aber hinreichend genauen Modell * "Just-in-time" Modellierung: Verwende kein komplexes Modell, so lange es nicht benötigt wird! * eine Schaltung $\Rightarrow$ eine Lösung

Schaltungsdesign/-entwurf. * ist die Fähigkeit der Schaltungssynthese auf Basis von Erfahrung und intensiver Analyse. * eine Spezifikation $\Rightarrow$ viele Lösungen * Entwurfspraktiken werden am besten durch's "Selbermachen" ausgebildet – "Machen ist wie wollen nur krasser."

1.14. Schöne neue Welt



1.15. From Sand to Silicon (Infineon, Dresden)

https://www.youtube.com/embed/bor0qLifjz4?list=PLO_wT97BGA6xC6hNy9VGtt1bKwVuQXI5B

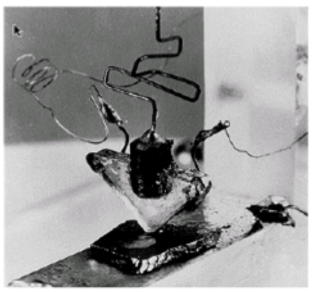
1.16. FinFET (Intel)

https://www.youtube.com/embed/_VMYPLXnd7E

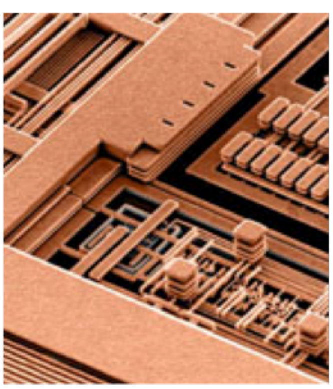
1.17. TSMC Fab (Next Gen 7/5 nm)

https://www.youtube.com/embed/Hb1WDxSoSec

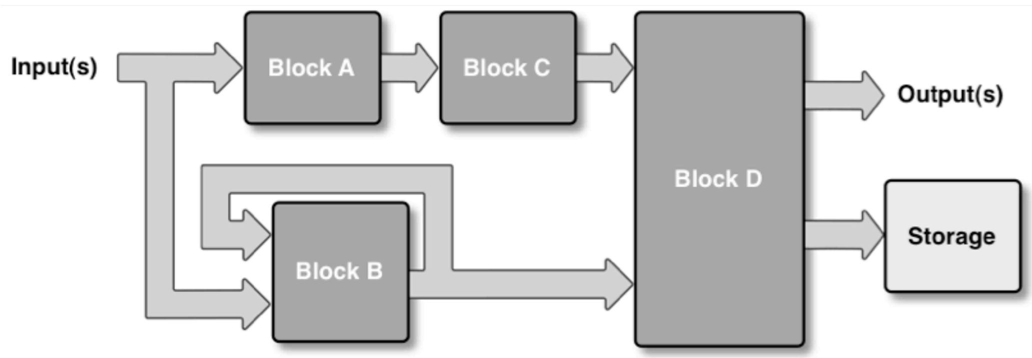
1.18. Es war einmal ...

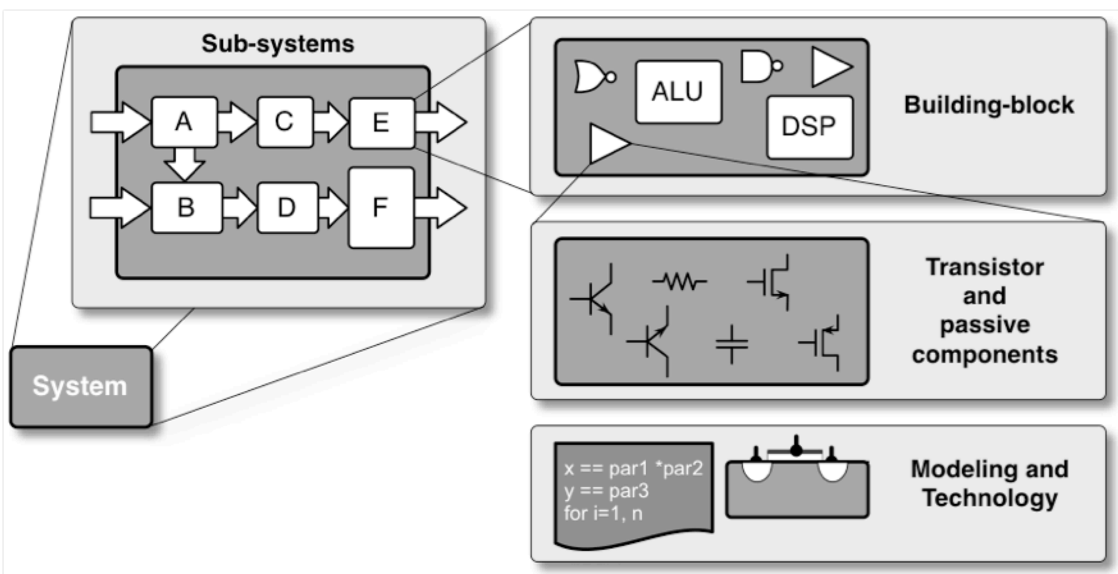


1.19. Damals und heute



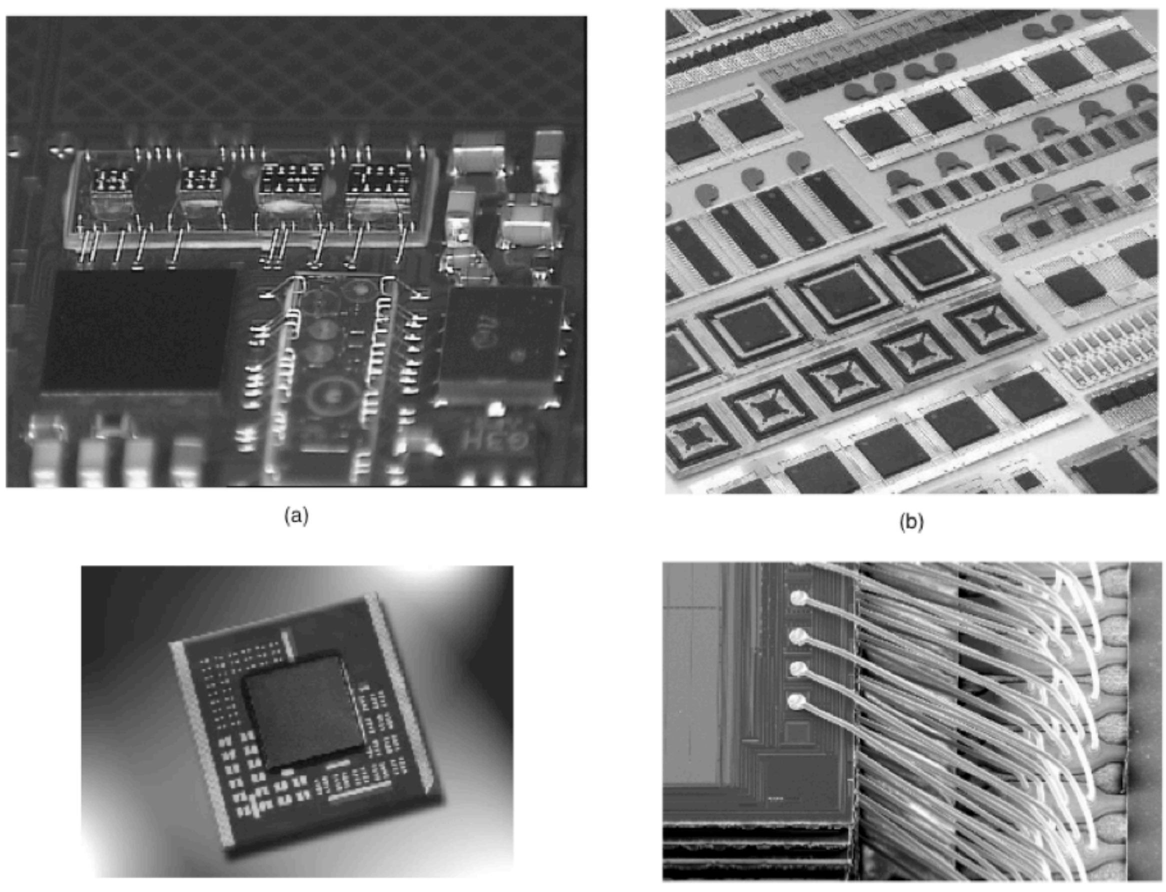
1.20. Systemhierarchie





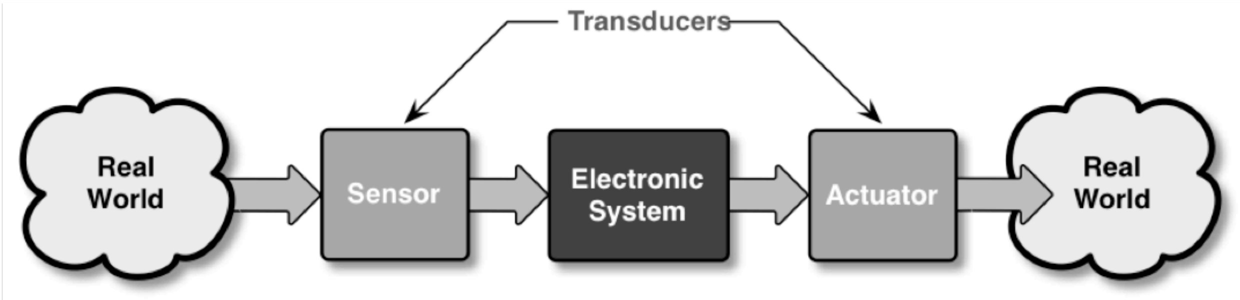
- Nutzen Sie Hierarchien zur Beschreibung komplexer Systeme
- Teile und herrsche

1.21. System Assembly



(a) (b) (c) (d)

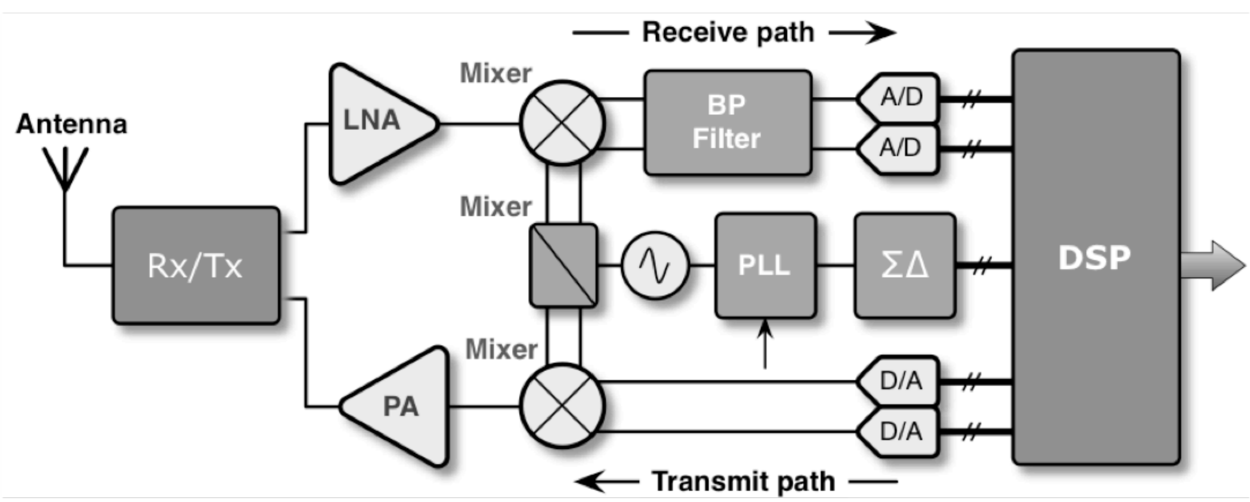
1.22. Schnittstellen zur Aussenwelt



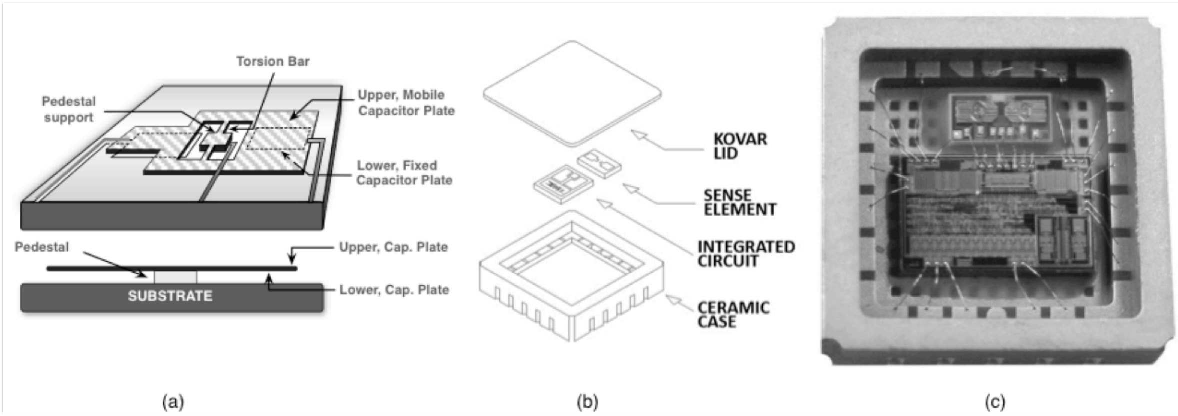
Entire system involving signals of real world.

1.23. Meeting mit einem System

Block diagram of a wireless communication system



1.24. System in a Package (SiP)



(a) Micro structure of an accelerometer.

(b) Assembling diagram of the system-on- package.

(c) Microphotograph. (*Courtesy of Silicon Designs, Inc.*).

1.25. Backend Phasen

- Packaging
- Zuverlässigkeit = Qualität auf Zeit
- Testing auf Wafer Level, known good die (KGD)
  - Burn-in und Accelerated Aging (thermischer Stress, Arrhenius Gesetz)
  - Automatic Test Equipment (ATE)
    - System Probe
    - Interconnect Test
    - Build-in Self-Test (BIST)
- Statistische Datenanalyse und Yield Prognosen
  - Ausfallrate FIT (failure in time)
  - Badewannenkurve

1.26. Sie werden Experte

Leistungsmerkmale. * Hintergrundwissen * Systemverständnis, Architektur, Herstellungsverfahren, Implementation

- Unterbewusste Kompetenz
  - Abgespeicherte Erfahrungen aus Erfolgsgeschichten und Misserfolgen
- Spezialwissen
  - Berufsspezifisches Wissen
- Teamwork Haltung
  - Kommunikationsfähigkeit, Berichtswesen und technische Präsentation
- Kreativität
- Tool-Kenntnisse

1.27. Evolution von Produkten

- Angetrieben durch Technologieverbesserung
  - Kosten (größere Chips, geringere Größe der Merkmale, bessere Ausbeute)
  - Leistung (neue Bauteile, höhere Geschwindigkeit, weniger Stromverbrauch)
- Angetrieben durch Verbesserung der Entwurfsmethodik
  - Architektur (Leistung, Funktionen)
  - CAD (Entwicklungskosten, Time-to-Market)
- Komplexität der Designs verdoppelt sich jedes Jahr (Moore's Gesetz)
- Rolle von CAD
  - Verbesserung der Produktivität von Konstruktionsprozessen
  - Reduzierung der Komplexität für den Konstrukteur
  - Sicherstellung des ordnungsgemäßen Betriebs der Geräte

1.28. EDA Kompetenz

- EDA-Anbieter (Tool-Entwickler)
  - Identifikation von Entwurfsaufgaben, Bedarf an Werkzeugen
  - Entwicklung von Strategien und Algorithmen
  - Implementierung von Software-Werkzeugen
  - Verifikation der Stabilität und Funktionalität der Software-Tools
- IC-Hersteller
  - Entscheidungsplanung, welches Tool die Produktivität steigern könnte
  - EDA-Tool-Manager, Installation und Wartung
  - Experten für Softwareeinsatz, Anwendung in Produktdesign und -entwicklung

- Dozenten und Studenten
  - Jobchancen
  - Notwendigkeit, auf dem Laufenden zu bleiben

1.29. Design-/Entwurfsmethodik

- Full Custom - vollständig manuell: ASIC
  - Überwiegend analoge Schaltungen
  - Einfache digitale Gatter
  - Volle Kontrolle, aber lange Entwicklungszeit (bis zu Jahren)
- Semi-custom: ASIC-Fertigung mit Verwendung von vorgefertigten Teilen
  - Standardzellen, Makrozellen, IP’s
  - Wiederverwendung von vordefinierten Blöcken oder Maskensätzen
  - Eingeschränkte Kontrolle/Flexibilität, aber kürzere Entwicklungszeit (bis zu Wochen)
- Vollständig automatisiert: Keine Fertigung, reprogrammierbare ASICs
  - FPGA, PLA
  - Ausschließlich digitale Schaltungen
  - Schnelles Prototyping

1.30. Nachhaltige Elektronik ...

https://www.youtube.com/embed/7S5IuaKiZIY

Geekchester.

1.31. Warum es sicht lohnt ...

https://www.youtube.com/embed/SwPGxwBZw6I

Circuit Song.

1.32. Und ab an den Strand ...

https://www.youtube.com/embed/ekkJlQf-K4I

Viva la Electronica.

Literaturverzeichnis

Part II.

Labor

2. Experiment 4 - Design of Analog Filters

2.1. Brief theory and motivation

Second order filters (or biquard filters) are important since they are the building blocks in the construction of N^{th} order filters, for $N > 2$. When N is odd, the N^{th} order filter can be realized using $N-1$ second order filters and one first order filter. When N is even, we need $N-1$ second order filters.

Second order filter can be used to construct four different types of filters. The transfer functions for the different filter types are shown in Section 2.1.1, where $f_0 = \frac{1}{2\pi RC}$ and H_0 is the low frequency gain of the transfer function. The filter names are often abbreviated as LPF (lowpass filter), HPF (highpass filter), BPF (bandpass filter), and BSF (bandstop filter).

2.1.1. Transfer funtions of active filters

- Lowpass Filter (LPF):

$$\frac{V_{03}}{V_i} = \frac{H_0}{\left(1 + \frac{s}{\omega_0 Q} + \frac{s^2}{\omega_0^2}\right)}$$

- Highpass Filter (HPF):

$$\frac{V_{01}}{V_i} = \frac{\left(H_0 \frac{s^2}{\omega_0^2}\right)}{\left(1 + \frac{s}{\omega_0 Q} + \frac{s^2}{\omega_0^2}\right)}$$

- Bandpass Filter (BPF):

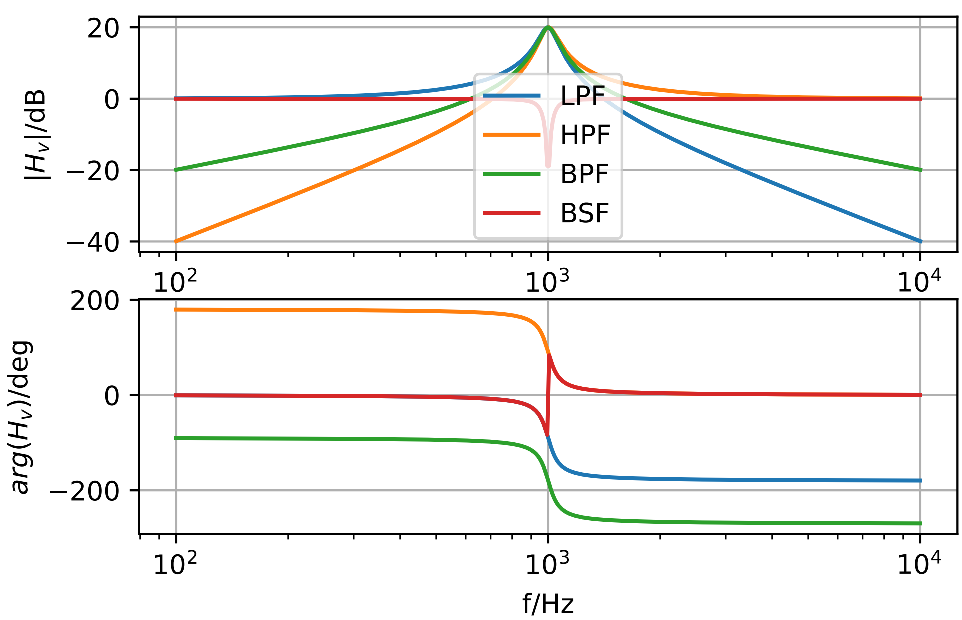
$$\frac{V_{02}}{V_i} = \frac{\left(-H_0 \frac{s}{\omega_0}\right)}{\left(1 + \frac{s}{\omega_0 Q} + \frac{s^2}{\omega_0^2}\right)}$$

- Bandstop Filter (BSF):

$$\frac{V_{01}}{V_i} = \frac{H_0\left(1+\frac{s^2}{\omega_0^2}\right)}{\left(1+\frac{s}{\omega_0 Q}+\frac{s^2}{\omega_0^2}\right)}$$

```
# -*- coding: utf-8 -*-
"""
Magnitude and phase response of active filters

"""

# %% Init
import numpy as np
# https://docs.scipy.org/doc/scipy/reference/signal.html#
import scipy.signal as sig
import matplotlib.pyplot as plt

# %% Definition der TFs
# https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.tf2zpk.html#scipy.signal.t
H0 = 1
Q = 10
f0 = 1e3
w0 = 2*np.pi*f0
f = np.logspace(2, 4, 400)

LPF = sig.lti([H0], [1/w0**2, 1/(w0*Q), 1])
w, magLPF, phaseLPF = LPF.bode(2*np.pi*f)

HPF = sig.lti([H0/w0**2, 0, 0], [1/w0**2, 1/(w0*Q), 1])
w, magHPF, phaseHPF = HPF.bode(2*np.pi*f)

BPF = sig.lti([-H0/w0, 0], [1/w0**2, 1/(w0*Q), 1])
w, magBPF, phaseBPF = BPF.bode(2*np.pi*f)

BSF = sig.lti([H0/w0**2, 0, H0], [1/w0**2, 1/(w0*Q), 1])
w, magBSF, phaseBSF = BSF.bode(2*np.pi*f)

# %% Erzeugen des Bode-Diagramms
plt.subplot(2, 1, 1)
plt.semilogx(f, magLPF, label='LPF')
plt.semilogx(f, magHPF, label='HPF')
plt.semilogx(f, magBPF, label='BPF')
plt.semilogx(f, magBSF, label='BSF')
plt.ylabel(r'$\vert H_v \vert$/dB')
plt.grid()
```

```
plt.legend()
plt.subplot(2, 1, 2)
plt.semilogx(f, phaseLPF, label='LPF')
plt.semilogx(f, phaseHPF, label='HPF')
plt.semilogx(f, phaseBPF, label='BPF')
plt.semilogx(f, phaseBSF, label='BSF')
plt.ylabel(r'$arg(H_v)$/deg')
plt.xlabel('f/Hz')
plt.grid()
plt.show()
```



In this experiment, we will describe a universal active filter, which provides all four filter functionalities. Figure 2.1 shows a second order universal filter realized using two integrators. Note that there are different outputs of the circuit that realize LPF, HPF, BPF and BSF functions.

2.2. Specification

Design a band-pass (BPF) and a band-stop (BSF) filter. * For the BPF, assume $f_0 = 1\,kHz$ and $Q = 1$. * For the BSF, assume $f_0 = 10\,kHz$ and $Q = 10$.

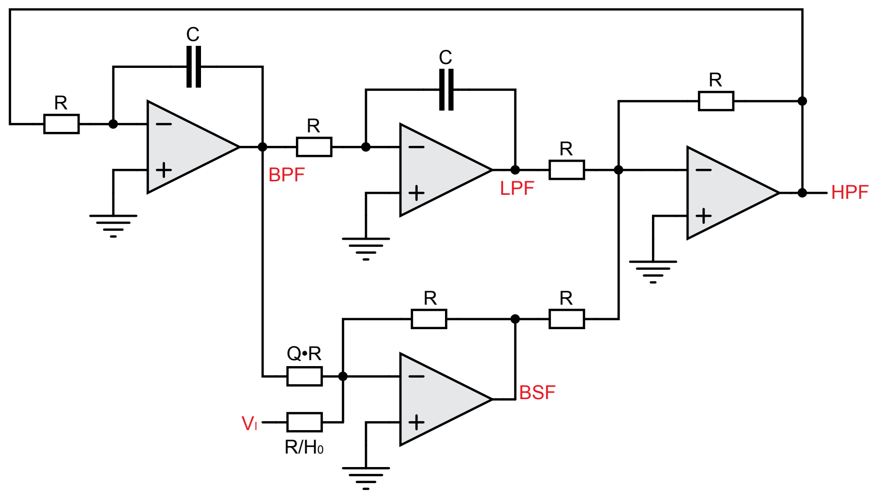


Figure 2.1.: A second-order universal active filter

2.3. Measurements to be taken

2.3.1. Steady-state response

Apply a square wave input (Try f = 1 kHz and f = 10 kHz to both BPF and BSF circuits and observe the outputs.

- Band-pass output will output the fundamental frequency of the square wave multiplied by the gain at the centre frequency. The amplitude at this frequency is given by $\frac{4V_p}{\pi H_0 Q}$, where V_p is the peak amplitude of the input square wave.
- The band-stop filter's output will carry all the harmonics of the square wave, other than fundamental. This illustrates the application of BSF as a distortion analyzer.

2.3.2. Frequency response

Apply a sine wave input and obtain the magnitude and the phase response.

2.4. What you should submit

2.4.1. Simulation with KiCad (ngspice)

Simulate the circuits and obtain the steady-state response and frequency response.

2.4.2. Measurements with Redpitaya STEMlab

Take the plots of the steady-state and frequency response with STEMlab oscilloscope or SCPI and compare it with simulation results. For frequency response, apply a sine wave input and vary its input frequency to obtain the phase and magnitude error.

2.4.2.1. SCPI mit Python

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Measurements for Bode plots
Input signal: DF_IN1
Output signal: DF_IN2

@author: Mirco Meiners (HSB)
"""

# %% Init
import time
# from datetime import datetime
import redpitaya_scpi as scpi
import numpy as np
import pandas as pd
import scipy.io as sio
import matplotlib.pyplot as plt

# %% Connection params
# IP of your STEMlab
IP = '192.168.111.182'
rp_s = scpi.scpi(IP)

# %% Measurement / Data Accquisition
DF_IN1 = pd.DataFrame()
DF_IN2 = pd.DataFrame()

# Parameters
func = 'SINE'
ampl = 0.2
offset = 0.0
freqs = np.arange(500, 2000, 100)
```

```python
for freq in freqs:

    rp_s.tx_txt('GEN:RST')  # Signal Generator reset
    rp_s.tx_txt('SOUR1:FUNC ' + str(func).upper())  # Wave form
    rp_s.tx_txt('SOUR1:VOLT ' + str(ampl))  # Magnitude
    rp_s.tx_txt('SOUR1:VOLT:OFFS ' + str(offset))  # Offset
    rp_s.tx_txt('SOUR1:FREQ:FIX ' + str(freq))  # Frequency
    rp_s.tx_txt('OUTPUT1:STATE ON')  # Output
    rp_s.tx_txt('SOUR1:TRig:INT')
    time.sleep(1)

    # Trigger
    rp_s.tx_txt('ACQ:RST')  # Input reset
    rp_s.tx_txt('ACQ:DEC 64')  # Decimation
    rp_s.tx_txt('ACQ:TRIG:LEV 0.5')  # Trigger level
    rp_s.tx_txt('ACQ:TRIG:DLY 8192')  # Delay
    rp_s.tx_txt('ACQ:START')  # Start measurement
    rp_s.tx_txt('ACQ:TRIG NOW')

    # Input IN1
    time.sleep(0.1)  # in seconds
    rp_s.tx_txt('ACQ:SOUR1:DATA?')  # Readout buffer IN1
    IN1str = rp_s.rx_txt()
    IN1str = IN1str.strip('{}\n\r').replace("  ", "").split(',')
    IN1 = np.array(list(map(float, IN1str)))
    DF_IN1[str(freq)] = IN1

    # Input IN2
    time.sleep(0.1)  # in seconds
    rp_s.tx_txt('ACQ:SOUR2:DATA?')  # Readout buffer IN2
    IN2str = rp_s.rx_txt()
    IN2str = IN2str.strip('{}\n\r').replace("  ", "").split(',')
    IN2 = np.array(list(map(float, IN2str)))
    DF_IN2[str(freq)] = IN2

    rp_s.tx_txt('OUTPUT2:STATE OFF')

# %% Data storage
Data_IN1 = 'data/DF_IN1'  # + str(datetime.now().strftime('%Y-%m-%d_%H_%M'))
Data_IN2 = 'data/DF_IN2'  #+ str(datetime.now().strftime('%Y-%m-%d_%H_%M'))

# %% Store data on disk as comma-seperated-values
DF_IN1.to_csv(Data_IN1 + '.csv', index=False)
DF_IN2.to_csv(Data_IN2 + '.csv', index=False)
```

```
# %% Store data on disk as excel sheet
# with pd.ExcelWriter(Data_IN + '.xlsx') as writer:
#     DF_IN1.to_excel(writer, sheet_name='IN1', index=False)
#     DF_IN2.to_excel(writer, sheet_name='IN2', index=False)

# DF_IN1.to_excel(Data_IN1 + '.xlsx', index=False)
# DF_IN2.to_excel(Data_IN2 + '.xlsx', index=False)

# %% Store data on disk as mat-file
# Ref. https://blog.finxter.com/5-best-ways-to-convert-pandas-dataframe-to-matlab/

# Convert the DataFrame to a dictionary with col names as keys
# dict_IN1 = DF_IN1.to_dict('list')
# dict_IN2 = DF_IN2.to_dict('list')

# Save the dictionary as a .mat file
# sio.savemat(DF_IN1 + '.mat', dict_IN1)
# sio.savemat(DF_IN2 + '.mat', dict_IN2)

# %% Store data on disk as HDF5
# DF_IN1.to_hdf(Data_IN1 + ".h5", "table", append=True)
# DF_IN2.to_hdf(Data_IN2 + ".h5", "table", append=True)

# %% Store data on disk as apache parquet
# DF_IN1.to_parquet(Data_IN1 + ".parquet", index=False)
# DF_IN2.to_parquet(Data_IN2 + ".parquet", index=False)

# %% Store data on disk as apache feather
# DF_IN1.to_feather(Data_IN1 + ".feather")
# DF_IN2.to_feather(Data_IN2 + ".feather")

# %% Test plot
# plt.plot(DF_IN1['900'], label='IN1')
# plt.plot(DF_IN2['900'], label='IN2')
# plt.legend()
# plt.show()
```

2.4.2.2. SCPI mit MATLAB

```
%% STEMlab Measurements for Bode plots
% @author: Mirco Meiners (HSB)
% Input signal: DF_IN1
```

```
% Output signal: DF_IN2
clear;

%% Define Red Pitaya as TCP client object
IP = '192.168.111.183';  % IP of your Red Pitaya ...
port = 5000;
RP = tcpclient(IP, port);
RP.InputBufferSize = 16384*32;

%% Open connection to Red Pitaya
RP.ByteOrder = "big-endian";
configureTerminator(RP, "CR/LF");

flush(RP);

%% Generate continous signal
func = "SINE";  % {sine, square, triangle, sawu, sawd, pwm}
ampl = 0.5;  % Set amplitude
offset = 0.0;  % Set offset
freqs = [500:100:2000];  % Set frequencies


%% Loop to measure multiple tones
for n = 1:length(freqs)
    % Send SCPI command to Red Pitaya to turn ON generator
    writeline(RP,'GEN:RST');  % Reset Generator
    writeline(RP,strcat("SOUR1:FUNC ", func));  % Set function of output signal
    writeline(RP,strcat("SOUR1:VOLT ", num2str(ampl)));  % Set amplitude
    writeline(RP,strcat("SOUR1:VOLT:OFFS ", num2str(offset)));  % Set offset
    writeline(RP,strcat("SOUR1:FREQ:FIX ", num2str(freqs(n))));  % Set frequency
    writeline(RP,'OUTPUT1:STATE ON');  % Turn on output OUT2

    writeline(RP,'SOUR1:TRig:INT');  % Generate trigger

    pause(1);

    % Trigger
    writeline(RP,'ACQ:RST');  % Input reset
    writeline(RP,'ACQ:DATA:FORMAT ASCII')
    writeline(RP,'ACQ:DATA:UNITS VOLTS')
    writeline(RP,'ACQ:DEC 64');  % Decimation 64
    writeline(RP,'ACQ:TRIG:LEV 0.5');  % Trigger level

    % Set trigger delay to 0 samples
    % 0 samples delay sets trigger to the center of the buffer
```

```
% Signal on your graph will have the trigger in the center (symmetrical)
% Samples from left to the center are samples before trigger
% Samples from center to the right are samples after trigger

writeline(RP,'ACQ:TRIG:DLY 8192');  % Delay
writeline(RP,'ACQ:SOUR1:GAIN LV');  % Sets gain to LV/HV (should be the same as jumpers)
writeline(RP,'ACQ:SOUR2:GAIN LV');  % Sets gain to LV/HV (should be the same as jumpers)

% Start & Trigger
% Trigger source setting must be after ACQ:START
% Set trigger to source 1 positive edge

writeline(RP,'ACQ:START');

% After acquisition is started some time delay is needed in order to acquire fresh samples in
pause(1);
% Here we have used time delay of one second, but you can calculate the exact value by taking
% length and sampling rate

writeline(RP,'ACQ:TRIG NOW');  % Instant data aquisition

% Wait for trigger
% Until trigger is true wait with acquiring
% Be aware of the while loop if trigger is not achieved
% Ctrl+C will stop code execution in MATLAB

while 1
   trig_rsp = writeread(RP,'ACQ:TRIG:STAT?');
   if strcmp('TD',trig_rsp(1:2))  % Read only TD
       break;
   end
end

% Read data from buffer
IN1 = writeread(RP,'ACQ:SOUR1:DATA?');
IN2 = writeread(RP,'ACQ:SOUR2:DATA?');

% Convert values to numbers.
% The first character in string is "{" and the last 3 are 2 spaces and "}".

IN1_num = str2num(IN1(1, 2:length(IN1)-3));
DF_IN1(:,n) = IN1_num';

IN2_num = str2num(IN2(1, 2:length(IN2)-3));
DF_IN2(:,n) = IN2_num';
```

```
    % Turn off generator OUT1
    writeline(RP,'OUTPUT1:STATE OFF');
end


%% Close connection to Red Pitaya
clear RP;


%% Save data as mat file
save('./data/IN_INT.mat', 'DF_IN1', 'DF_IN2');
% save('./data/IN1_INT.mat', 'DF_IN1');
% save('./data/IN2_INT.mat', 'DF_IN2');

%% Save data as parquet file
% parquet data is of type table, no matrix operations
% parquetwrite('data/IN1_INT.parquet', array2table(DF_IN1));
% parquetwrite('data/IN2_INT.parquet', array2table(DF_IN2));

%% Save data as excel sheet
% data is table data, no matrix operations
% writematrix(DF_IN1, './data/IN1_UB_VBS_mat.xlsx');
% writematrix(DF_IN1, './data/IN2_UB_VBS_mat.xlsx');
% writematrix(DF_IN1, './data/IN_UB_VBS_VBP_mat.xlsx', 'Sheet', 1);
% writematrix(DF_IN2, './data/IN_UB_VBS_VBP_mat.xlsx', 'Sheet', 2);
```

2.5. Exercise Set 4

2.5.1. 3rd order butterworth

Higher order filters are normally designed by cascading second order filters and, if needed, one first-order filter. Design a third order Butterworth lowpass filter using Python or Matlab and obtain the frequency response as well as the transient response of the filter. The specifications are bandwidth of the filter $f_0 = 10\,kHz$ and gain $H_0 = 10$.

2.5.2. Notch filter

Design a notch filter (band-stop filter) to eliminate the $50\ Hz$ power life frequency. In order to test this circuit, synthesize a waveform $v(t) = \sin(100\pi t) + 0.1\sin(200\pi t)$ Volts and use it as the input to the filter. What output did you obtain?

3. Validation of Transfer Functions

Lorem ipsum

4. Mathematical derivation of the transfer functions of the biquad

First, the individual equations of the operational amplifiers were set up:

$$V_1 = -(V_3 + V_4) \tag{4.1}$$

$$V_2 = -\left(\frac{1}{s}\omega_0 \cdot V_1\right) \tag{4.2}$$

$$V_3 = -\left(\frac{1}{s}\omega_0 \cdot V_2\right) \tag{4.3}$$

$$V_4 = -\left(\frac{V_2}{Q} + H_0 \cdot V_i\right) \tag{4.4}$$

4.1. Calculation of $\frac{V_2}{V_i}$

Insert Equation 4.1 into Equation 4.2 :

$$\begin{aligned} V_2 &= -\frac{\omega_0}{s} \cdot V_1 = -\frac{\omega_0}{s} \cdot (-V_3 - V_4) \\ &= -\frac{\omega_0}{s} \cdot \left(\frac{\omega_0}{s} \cdot V_2 + \frac{1}{Q} \cdot V_2 + H_0 \cdot V_i\right) \end{aligned} \tag{4.5}$$

Summarizing results in:

$$V_2 = -\frac{\omega_0^2}{s^2} \cdot V_2 - \frac{\omega_0}{sQ} \cdot V_2 - \frac{\omega_0}{s} H_0 \cdot V_i \tag{4.6}$$

$$\Rightarrow \frac{\omega_0}{s} H_0 \cdot V_i = -\left(\frac{\omega_0^2}{s^2} + \frac{\omega_0}{sQ} + 1\right) V_2 \tag{4.7}$$

Resolve to $\frac{V_2}{V_i}$:

$$\frac{V_2}{V_i} = \frac{\frac{\omega_0}{s}H_0}{-\left(\frac{\omega_0^2}{s^2} + \frac{\omega_0}{sQ} + 1\right)} = -\frac{\frac{s}{\omega_0}H_0}{1 + \frac{s}{\omega_0 Q} + \frac{s^2}{\omega_0^2}} \tag{4.8}$$

4.2. Transfer-function $\frac{V_3}{V_i}$:

Insert Equation 4.8 into Equation 4.3 :

$$V_3 = -\frac{\omega_0}{s} \cdot V_2 = -\frac{\omega_0}{s} \cdot \left(-\frac{\frac{s}{\omega_0}H_0}{1 + \frac{s}{\omega_0 Q} + \frac{s^2}{\omega_0^2}} \cdot V_i\right)$$

$$V_3 = \left(\frac{H_0}{1 + \frac{s}{\omega_0 Q} + \frac{s^2}{\omega_0^2}}\right) V_i$$

$$\Rightarrow \frac{V_3}{V_i} = \frac{H_0}{1 + \frac{s}{\omega_0 Q} + \frac{s^2}{\omega_0^2}} \tag{4.9}$$

4.3. Transfer-function $\frac{V_4}{V_i}$:

Insert Equation 4.8 into Equation 4.4 :

$$V_4 = -\left(\frac{1}{Q} \cdot V_2 + H_0 \cdot V_i\right) = -\left(\frac{1}{Q} \cdot \left(-\frac{\frac{s}{\omega_0} H_0}{1 + \frac{s}{\omega_0 Q} + \frac{s^2}{\omega_0^2}} \cdot V_i\right) + H_0 \cdot V_i\right)$$

$$= \left(\frac{\frac{s}{\omega_0 Q} H_0}{1 + \frac{s}{\omega_0 Q} + \frac{s^2}{\omega_0^2}} - H_0\right) V_i \tag{4.10}$$

$$= \left(\frac{\frac{s}{\omega_0 Q} - \left(1 + \frac{s}{\omega_0 Q} + \frac{s^2}{\omega_0^2}\right)}{1 + \frac{s}{\omega_0 Q} + \frac{s^2}{\omega_0^2}} \cdot H_0\right) V_i \tag{4.11}$$

$$= -\frac{\left(1 + \frac{s^2}{\omega_0^2}\right) H_0}{1 + \frac{s}{\omega_0 Q} + \frac{s^2}{\omega_0^2}} \cdot V_i \tag{4.12}$$

$$\Rightarrow \frac{V_4}{V_i} = -\frac{\left(1 + \frac{s^2}{\omega_0^2}\right) H_0}{1 + \frac{s}{\omega_0 Q} + \frac{s^2}{\omega_0^2}} \tag{4.13}$$

Note: In the task booklet, the following equation is given here:

$$\frac{V_4}{V_i} = \frac{\left(1 + \frac{s^2}{\omega_0^2}\right) H_0}{1 + \frac{s}{\omega_0 Q} + \frac{s^2}{\omega_0^2}}$$

This corresponds to a negation of my own result.

4.4. Calculation of the transfer function $\frac{V_1}{V_i}$:

Based on equation Equation 4.1 :

$$V_1 = -(V_3 + V_4)$$

Using the transfer-function Equation 4.9 for V_3:

$$\frac{V_3}{V_i} = \frac{H_0}{1 + \frac{s}{\omega_0 Q} + \frac{s^2}{\omega_0^2}}$$

Case 1: Use of the negated form of the transfer function of V_4 from my own derivation:

$$\frac{V_4}{V_i} = -\frac{\left(1 + \frac{s^2}{\omega_0^2}\right) \cdot H_0}{1 + \frac{s}{\omega_0 Q} + \frac{s^2}{\omega_0^2}} \tag{4.14}$$

Inserting Equation 4.9 and Equation 4.14 into Equation 4.1 results in:

$$V_1 = -\left(V_3 + V_4\right)$$

$$V_1 = -\left(\frac{H_0}{D} + \left(-\frac{\left(1+\frac{s^2}{\omega_0^2}\right)H_0}{D}\right)\right)V_i$$

$$V_1 = \left(\frac{\left(\frac{s^2}{\omega_0^2}\right)H_0}{D}\right)V_i \quad \text{where } D = 1 + \frac{s}{\omega_0 Q} + \frac{s^2}{\omega_0^2}$$

$$\Rightarrow \frac{V_1}{V_i} = \frac{\left(\frac{s^2}{\omega_0^2}\right)H_0}{1+\frac{s}{\omega_0 Q}+\frac{s^2}{\omega_0^2}} \tag{4.15}$$

Case 2: Use the positive form of the transfer function of V_4 as specified in the task booklet:

$$\frac{V_4}{V_i} = \frac{\left(1+\frac{s^2}{\omega_0^2}\right)\cdot H_0}{1+\frac{s}{\omega_0 Q}+\frac{s^2}{\omega_0^2}} \tag{4.16}$$

Insert into Equation 4.1 :

$$\begin{aligned} V_1 &= -\left(V_3+V_4\right) = -\left(\frac{H_0}{D} + \frac{\left(1+\frac{s^2}{\omega_0^2}\right)H_0}{D}\right)V_i \\ &= -\left(\frac{\left(2+\frac{s^2}{\omega_0^2}\right)H_0}{D}\right)V_i && (4.17) \\ \Rightarrow \frac{V_1}{V_i} &= -\frac{\left(2+\frac{s^2}{\omega_0^2}\right)H_0}{1+\frac{s}{\omega_0 Q}+\frac{s^2}{\omega_0^2}} && (4.18) \end{aligned}$$

4.5. Repeated calculation of the Transfer-function $\frac{V_2}{V_i}$ via V_1:

Based on the correct function for V_1 Equation 4.15 :

$$\frac{V_1}{V_i} = \frac{\frac{s^2}{\omega_0^2} \cdot H_0}{1 + \frac{s}{\omega_0 Q} + \frac{s^2}{\omega_0^2}}$$

Using the known relationship $V_2 = -\frac{\omega_0}{s} \cdot V_1$:

$$V_2 = -\frac{\omega_0}{s} \cdot V_1$$

$$V_2 = -\frac{\omega_0}{s} \cdot \frac{\frac{s^2}{\omega_0^2} \cdot H_0}{1 + \frac{s}{\omega_0 Q} + \frac{s^2}{\omega_0^2}} \cdot V_i$$

$$V_2 = -\frac{\frac{s}{\omega_0} \cdot H_0}{1 + \frac{s}{\omega_0 Q} + \frac{s^2}{\omega_0^2}} \cdot V_i$$

$$\Rightarrow \frac{V_2}{V_i} = -\frac{\frac{s}{\omega_0} H_0}{1 + \frac{s}{\omega_0 Q} + \frac{s^2}{\omega_0^2}} \tag{4.19}$$

A comparison of the two transfer functions $\frac{V_2}{V_i}$ Equation 4.8 and Equation 4.19 shows that the error is due to equation $\frac{V_4}{V_1}$ since the derivation of $\frac{V_1}{V_i}$ remains correct.

4.6. Conclusion

Only the use of the negated form of $\frac{V_4}{V_i}$ from Equation 4.14 leads to a consistent result for $\frac{V_1}{V_i}$ according to Equation 4.15 . The version from the exercise booklet (Equation 4.16) therefore appears to be incorrect.