

# **Labor CAE - Einfuehrung in die Simulation elektrischer Schaltungen mit LTSpice (6 UE)**

---

Andreas Schaefer - Matthias Holetzko - Franz Smagacz-Allramseder

15.03.2020

Duale Hochschule Baden-Württemberg

-

Fakultät Technik

Studiengang Mechatronik

# Introduction to LTSpice

---

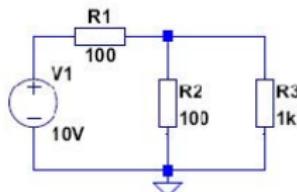
## Schaltungssimulation mit LTSpice

1. Viele Programme zur elektrischen Schaltungssimulation beruhen auf einem Spice Kernel,
2. SPICE bedeutet: Simulation Program with Integrated Circuit Emphasis,
3. SPICE wurde 1973 an der University of California at Berkeley entwickelt,
4. SPICE ist eine “open source” Software (kostenlos) Software.

# Was ist LTSpice - Wie funktioniert eine Simulation?

In einem klassischen SPICE Program erfolgt die Eingabe der Schaltung über eine Netzliste.

Schaltung:



SPICE-Netzliste

```
* Spannungsteiler1
V1 N001 0 10V
R1 N001 N002 100
R2 N002 0 100
R3 N002 0 1k
.end
```

Eine Netzliste repräsentiert eine Schaltung entsprechend Knoten- und Maschenregel,

Viele moderne SPICE Simulationsprogramme bieten dem Nutzer jedoch eine grafische Bedienoberfläche. So auch das in diesem Labor verwendete Programm, **LTSpice XVII**.

# Was ist LTSpice - Installation & Features

1. Hier findet Ihr alle Infos vom Hersteller, **analog devices**.
2. Installer fuer MacOs  
<http://ltspice.analog.com/software/LTspice.dmg>.
3. Installer fuer MacOs  
<http://ltspice.analog.com/software/LTspice.dmg>.

# Ziel dieser Laborübung

1. Erarbeiten der Grundlagen zur Simulation mit LTSpice XVII,
2. Eigenständige Simulation von Schaltungen für Studien-/Abschlussarbeiten,
3. Verständnis theoretischer Inhalte mit empirischen Simulationen ergänzen.

## nächste Schritte

- Übungen zum Einstieg in die Simulationswelt,
- Schrittweise Erarbeitung eines Kleinprojektes  
**Simulation einer Temperaturmessbrücke,**
- Simulation digitaler Schaltungen und Bauelemente

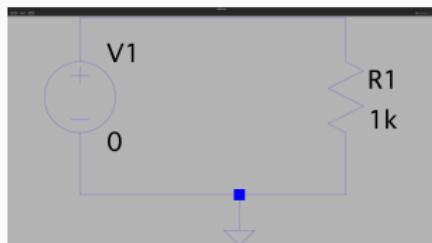
# LTS defense XVII - Bitte öffnet alle das Programm

Die Benutzeroberfläche von LTS defense XVII besteht aus den folgenden Elementen.

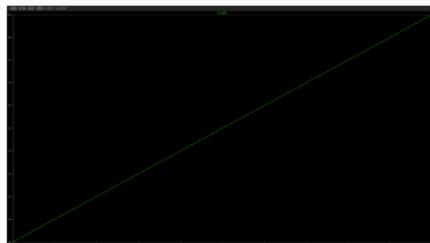
**Hinweis:** Die Benutzeroberfläche von LTS defense auf MacOS und Windows unterscheidet sich leicht. Die Hauptmenuleiste ist für MacO nicht sichtbar.



(a) Hauptmenuleiste



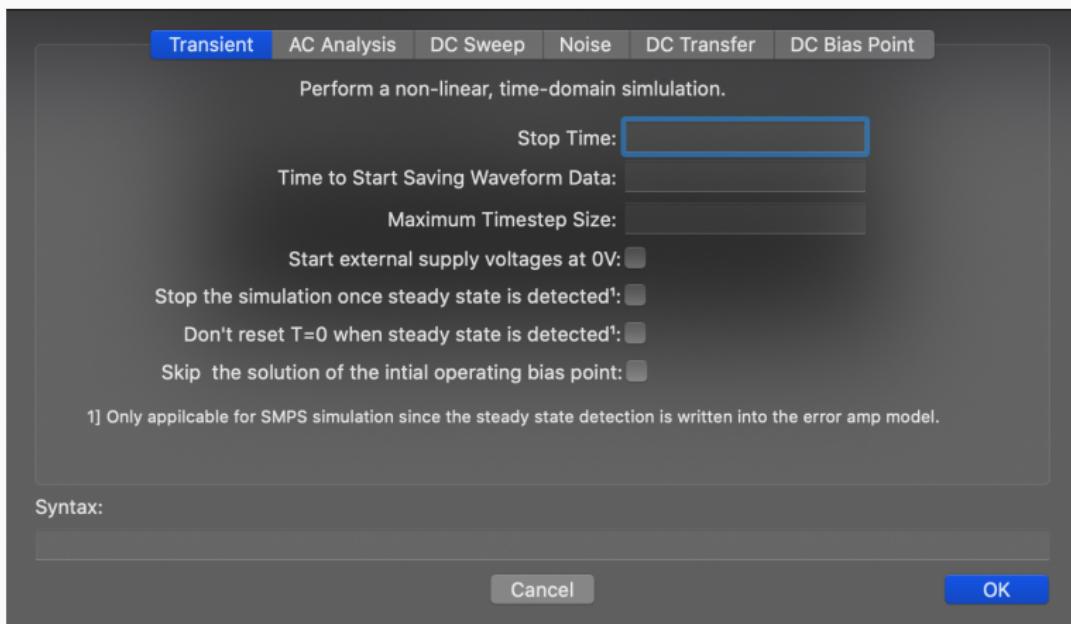
(b) schematic



(c) waveform

# LTS spice XVII - Überblick Simulationsarten

Die Simulationsart kann im Menu **Simulation Command** konfiguriert werden.



# LTS defense XVII - Überblick Simulationsarten

---

## **Transient**

Analyse einer Schaltung über die Zeit, z.B. zur Analyse von Einschwingvorgängen

---

## **AC Analysis**

Analyse einer Schaltung unter Variation der Frequenz,  
z.B. zur Analyse von Grenzfrequenzen

---

## **DC Sweep**

Analyse einer Schaltung unter Variation der Spannung,  
z.B. zur Analyse von Bauteilkennlinien

---

## **Operation Point**

Analyse des Arbeitspunktes einer Schaltung ohne Variation

---

## **Noise**

Analyse von Rauschverhalten, Bauteilfehlern oder z.B. EMV Einflüssen

---

## **DC Transfer**

Nicht Teil des Labors

---

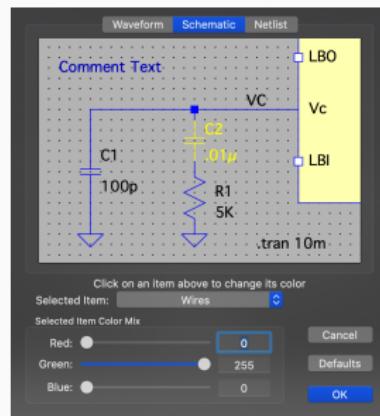
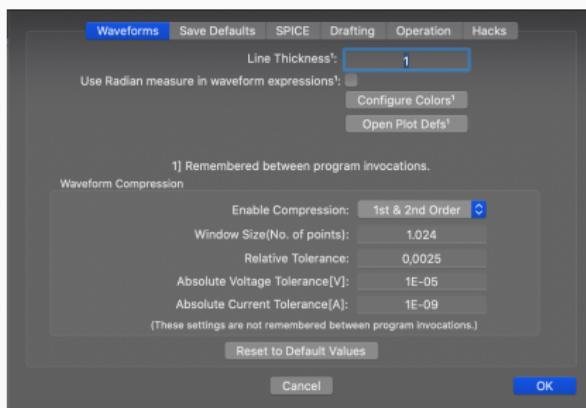
## **DC Bias Point**

Nicht Teil des Labors

---

# Nützliche Voreinstellungen

Die Farbdarstellung von LTSpice kann jederzeit konfiguriert werden. Geht dazu über das **control panel** Feld,  in die Farbeinstellungen (**configure colors**). Hier könnt ihr flexibel die Farbdarstellung konfigurieren.



# Shortkeys

Es gibt eine gute Übersicht von HotKeys die die Arbeit mit LTSpice deutlich effektiver macht (nachfolgend sind Hyperlinks embedded).

- Windows ShortKeys.
- MacOS ShortKeys.
- LTSpice getting started guide.

- Auswählen von Bauteilen aus der **Bibliothek**: F2 
- Verdrahtung zeichnen: F3 
- Verschieben von Bauteilen: Verschiebemodus aktivieren mit F7  oder F8 
- Rotieren von Bauteilen: Strg + R  (ggf. vorher mit F7 oder F8 + linker Mausklick auswählen)
- Spiegeln von Bauteilen: Strg + E  (ggf. vorher mit F7 oder F8 + linker Mausklick auswählen)
- Löschen von Bauteilen: F5 (→ Scherensymbol  ) + linker Mausklick auf Bauteil
- Kopieren von Bauteilen: F6 (→ Kopiersymbol  ) + linker Mausklick auf Bauteil, dann verschieben
- Aktion rückgängig machen: F9 (Undo) 
- Ansicht vergrößern: Strg + Z 
- Ansicht verkleinern: Strg + B 
- Schaltungsgröße an Fenstergröße anpassen: Leertaste
- Verlassen eines Bearbeitungsmodus: Esc-Taste

## Nützliche Hinweise

- Zahlenwerte immer mit **Dezimalpunkt statt Komma eingeben! 3.5 anstatt 3,5!**
- SPICE arbeitet nach dem Knotenpotenzialverfahren - **Es muss immer ein Bezugsknoten (Masse, Ground) angegeben werden**, sonst treten Simulationsfehler auf (Achtung: Es erfolgt keine Fehlermeldung!).

# **Hands-on LTSpice - Das Labor beginnt**

---

# Vorgehensweise

Diese Übungen sind zur interaktiven Arbeit mit LTSpice gedacht.  
Sie führen von der Aufgabe über den Lösungsansatz bis hin zur fertigen Analyse.

Bitte geht nach den folgenden Schritten vor, um die Übungen strukturiert zu bearbeiten.

1. **Schematic modellieren**
2. **Simulationsart analysieren**
3. **Simulationsart konfigurieren**
4. **Ergebnis mit der Vorgabe abgleichen**

Sollte es zu Problemen kommen, bitten wir euch uns direkt einen Hinweis zu geben!

## Übungen

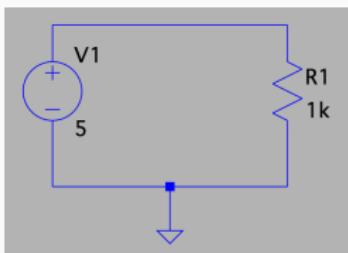
---

# ohmscher Widerstand

## Ziel - Darstellung von Bauteilkennlinie

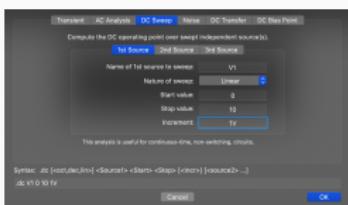
Das ohmsche Gesetz  $U = RI$  lässt sich mit Hilfe eines kleinen Schaltungsexperimentes gut visualisieren und nachvollziehen. Hierzu werden wir die folgende Schaltung aufbauen und den Stromverlauf über dem Widerstand darstellen.

### Erstellung des Schaltplans



- Erstellt ein neues schematic (File –> new schematic)
- Speichert es direkt als neue Datei ab (File –> save as)
- Öffnet den Bauteileditor (**F2**) und für eine Spannungsquelle hinzufügen (voltage)
- Öffnet erneut den Bauteileditor und für einen Widerstand hinzufügen (resistor bzw. EuropeanResistor fuer die bekannte Box-Darstellung)
- Fügt einen Masseknoten als Bezugsknoten hinzu (**F4**).
- Verdrahtet die Schaltung (**F3**)

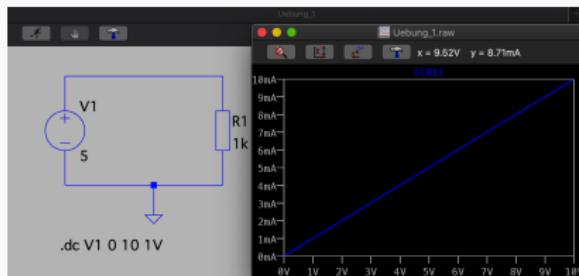
### Konfiguration der Simulation



- Im Menu Simulation, wählt **Edit simulation command** und wählt den DC Sweep.
- Unser Ziel ist es den Stromverlauf über dem Widerstand zu simulieren
- Daher konfigurieren wir die Spannungsquelle **V1** mit einem **linearen Sweep** von 0 bis 10 V mit einer **Schrittweite** von 1V.
- Bestätigt mit **OK** und fügt die Sumulationsansweisung dem schematic hinzu

# ohmscher Widerstand

## Simulation und Analyse



- Klickt auf (run) und LTSpice startet die Simulation
- Der waveform viewer öffnet sich - ihr könnt auf zwei Arten Strom, Spannung und Leistungen aus eurer Schaltung anzeigen lassen.

**Im schematic** könnt ihr über ein Bauteil mit der Maus fahren und es erscheint eine Stromzange

**Im schematic** könnt ihr über ein Bauteil mit der Maus fahren, Shift gedrückt halten und es erscheint eine Leistungsmessanzeige

**Im schematic** könnt ihr über einen Knoten mit der Maus fahren und es erscheint ein Spannungsmesser

**Im waveform viewer** könnt ihr über rechten Mausklick — > add trace die verfügbaren Messtellen direkt auswählen

- Wir wählen **I(R1)**, den Strom durch unseren Widerstand R1.

## Ergebnis und Auswertung

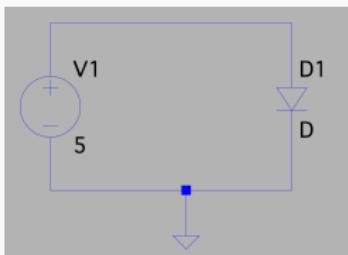
Wie zu erwarten liefert dieses einfache Beispiel den Zusammenhang zwischen Strom, Spannung und Widerstand. Probiert an dieser Stelle gerne die Spannung, Schrittweite zu variieren oder weitere Werte im waveform viewer anzueigen zu lassen.

# Diode

## Ziel - Darstellung von Bauteilkennlinie

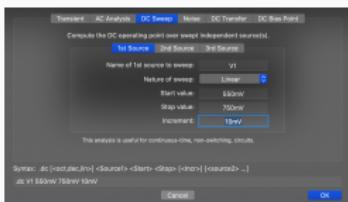
Eine Diode wird oft durch Ihre Durchlassspannung klassifiziert. Dies ist die Spannung ab der die Diode leitend wird. Leitend ist Sie, wenn ein Strom fließt. Dieses Verhalten möchten wir in einem kleinen simulativen Experiment herausarbeiten.

### Erstellung des Schaltplans



- Speichert das Projekt direkt als neue Datei ab (File —> save as )
- Löscht den Widerstand heraus (**F5**)
- Öffnet erneut den Bauteileditor (**F2**) und für eine einfache Diode hinzufügen (sucht nach Diode ...)
- Verdrahtet die Schaltung wieder vollständig (**F3**)

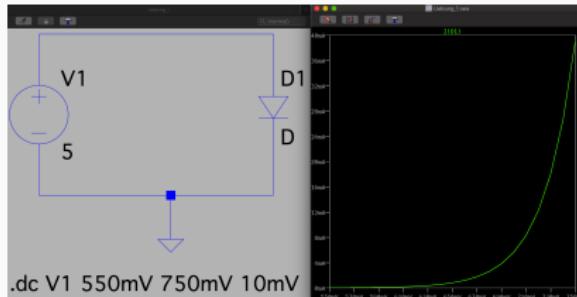
### Konfiguration der Simulation



- Im Menu Simulation, wählt **Edit simulation command** und bleibt beim DC Sweep.
- Unser Ziel ist es den Stromverlauf durch die Diode zu messen. Dafür müssen wir die Spannung erhöhen. Wir wissen, dass die notwendige Spannung im Bereich 600 - 700 mV liegen muss.
- Daher konfigurieren wir die Spannungsquelle **V1** mit einem **linearen Sweep** von 550 bis 750 mV mit einer **Schrittweite** von 10 mV.
- Bestätigt mit **OK** und fügt die Sumulationsansweisung dem schematic hinzu

# Diode

## Simulation und Analyse



- Klickt auf (run) und LTSpice startet die Simulation
- Wir wählen **I(D1)**, den Strom durch die Diode.

## Ergebnis und Auswertung

Wie zu erwarten liefert dieses einfache Beispiel den Zusammenhang zwischen Strom, Spannung und Widerstand. Probiert den Spannungsbereich des DC-Sweep von 550 - 750 mV auf 550 mV - 2V zu erhöhen.

- Was fällt euch auf?
- Könnt ihr euch herleiten, warum man eine Diode immer mit einem Vorwiederstand betreiben sollte?

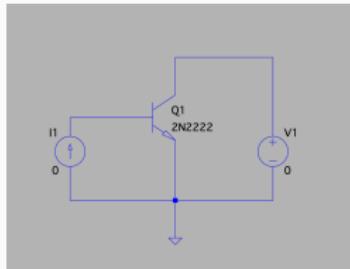


Mit einem rechten Mausklick auf die Diode könnte ihr die Dioden Typen variieren. (**PICK new diode**) Zum Beispiel von einer idealen aus dem obigen Beispiel zu einer beliebigen realen entsprechend dem Modell des Herstellers, schaut euch die Unterschiede an.

## Ziel - Darstellung des Ausgangskennlinienfeld

Das Ausgangskennlinienfeld eines npn Transistors beschreibt den Zusammenhang von Kollektorstrom  $I_c$  und der Spannung an der Kollektor-Emitter Strecke  $U_{ce}$ . Das Kennlinienfeld wird für verschiedene Basisströme  $I_b$  angegeben.

### Erstellung des Schaltplans



- Startet mit einem neuen schematic
- Speichert das Projekt direkt als neue Datei ab (File –> save as )
- Fügt eine ideale Stromquelle (**F2 ... current**) hinzu und dreht dieser (**STRG+R**)
- Fügt einen npn-Transistor(**F2 ... npn**) hinzu.
- Per rechtem Mausklick könnt ihr vom idealen npn analog zur Diode im letzten Beispiel zum 2N2222 wechseln.
- Verdrahtet die Schaltung wieder vollständig (**F3**)

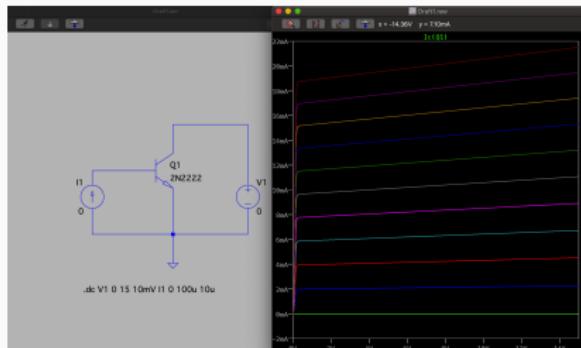
### Konfiguration der Simulation



- Im Menu Simulation, wählt **Edit simulation command** und bleibt beim DC Sweep.
- Unser Ziel ist es das Ausgangskennlinienfeld des 2N2222 simlativ zu bestimmen. Dazu verwenden wir einen DC-Sweep mit einer ideal Stromquelle (I1) die verschiedene Basisströme  $I_b$  simuliert und eine Spannungsquelle (V1) die die Spannung  $U_{ce}$  simuliert.
- V1 soll von 0 - 15V in 1V Schritten simuliert werden, I1 von 0 - 100u in 10u Schritten.
- Bestätigt mit **OK** und fügt die Simlationsansweisung dem schematic hinzu

# NPN-Transistor

## Simulation und Analyse



- Klickt auf (run) und LTSpice startet die Simulation
- Wir wählen **IC(Q1)**, den Kollektorstrom.

## Ergebnis und Auswertung

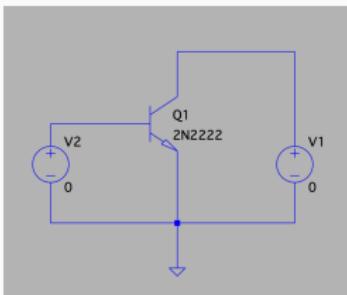
Jeder Graph steht für einen simulierten Basisstrom. Per rechtem Mausklick –> View –> Steps könnt ihr einzelne Graphen zur detaillierten Analyse auswählen.

Achtet darauf, welche Quelle ihr im `.dc...` simulation command zuerst wählt. **Quelle 1 ergibt im Diagramm die Abszisse, die Quelle 2 die Ordinate.**

# NPN-Transistor

Wenn ihr den Zusammenhang zwischen der Basisspannung  $U_{be}$  und dem Kollektorstrom  $I_c$  simulativ herausfinden wollt, müsst ihr die Schaltung leicht variieren. Dazu werden wir das schematic wie folgt anpassen.

## Erstellung des Schaltplans



- Speichert das Projekt direkt als neue Datei ab (File –> save as )
- Löscht die Stromquelle (**F5**) und fügt eine Spannungsquelle hinzu.
- Verdrahtet die Schaltung wieder vollständig (**F3**)
- Super simple dieses mal :)

## Konfiguration der Simulation

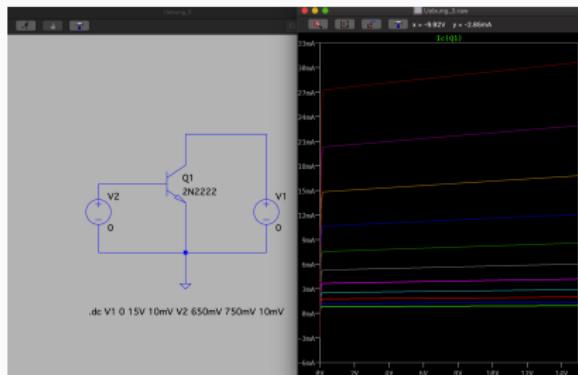


- Im Menu Simulation, wählt **Edit simulation command** und bleibt beim DC Sweep.
- Unser Ziel ist es das Ausgangskennlinienfeld des 2N2222 simlativ zu bestimmen. Dazu verwenden wir dieses mal einen DC-Sweep mit einer Spannungsquelle, die die Basis-Emitterspannung  $U_{be}$  simuliert und eine Spannungsquelle (V1) die die Spannung  $U_{ce}$  simuliert.
- V1 soll von 0 - 15V in 1V Schritten simuliert werden, V2 von 650 - 750 mV in 10mV Schritten.
- Bestätigt mit **OK** und fügt die Simulationsansweisung dem schematic hinzu

# NPN-Transistor

---

## Simulation und Analyse



- Klickt auf (run) und LTSpice startet die Simulation
- Wir wählen **IC(Q1)**, den Kollektorstrom.

---

## Ergebnis und Auswertung

Jeder Graph steht für eine simulierte  $U_{be}$ . Per rechtem Mausklick –> View –> Steps könnt ihr einzelne Graphen zur detaillierten Analyse auswählen.

Achtet darauf, welche Quelle ihr im `.dc...` simulation command zuerst wählt. **Quelle 1 ergibt im Diagramm die Abszisse, die Quelle 2 die Ordinate.**

# OPV Schaltungen - transient, ideal

## Ziel - Simulation eines invertierenden Verstärkers

Wir wollen in einem einfachen simulativen Experiment die Funktionalität eines invertierenden Verstärkers nachvollziehen.



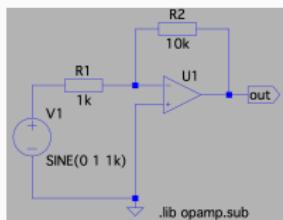
$$V_{out} = -\frac{R2}{R1} V_1 \quad (1)$$

Wenn man nur daran interessiert ist die grundsätzliche Funktionalität einer Schaltung zu verifizieren, kann man ideale Bauelemente aus der LTSpice Bibliothek verwenden. Wie im obigen Beispiel benötigt z.B. der Operationsverstärker entgegen der Realität dann keine Spannungsversorgung. Für einen idealen OPV bietet LTSpice das Bauelement **opamp** an, welches Ihr im Bauteileditor findet direkt über die Suche findet.

**Wichtig:** Ihr müsst als Spice-Direktive jedoch noch ein zu verwendendes Model einbinden. (.lib opamp.sub)

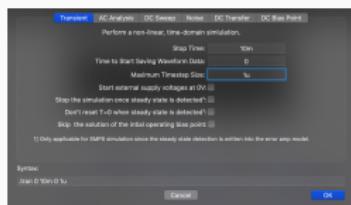
### Erstellung des Schaltplans

- Baut den Schaltplan mit dem o.g. opamp als OPV auf.
- Über einen rechten Mausklick kommt ihr ins **Advanced** Menü der Spannungsquelle. Hier könnt ihr Sie als Signalgenerator konfigurieren. Wir wählen einen **Sinus mit 1V Amplitude und der Frequenz 1kHz**
- Die library kann direkt als spice directive eingebunden werden (siehe Folie 5, .op)
- Ihr könnt über die Funktion **Label Net (F4)** einen Knoten umbenennen und ihm mit einem Symbol für In-/Output versehen. Nennt den Ausgang der Schaltung z.B. *out*.
- **Hinweis:** Wenn ihr einen Knoten benennt, dann kann liegt unter diesem Namen überall im Schaltplan das selbe! Potential an. Dadurch könnt ihr den Plan übersichtlicher gestalten.



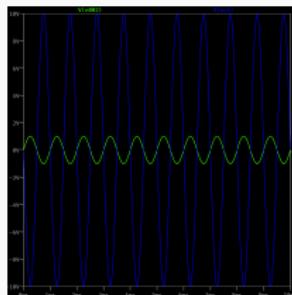
# OPV Schaltungen - transient, ideal

## Konfiguration der Simulation



- Im Menu Simulation, wählt **Edit simulation command** und wählt eine **Transient Analyse**.
- Unser Ziel ist es am Ausgang eine Verstärkte Spannung entsprechend des Verstärkungsfaktors der Schaltung zu messen.
- Da die Schaltung ideal kein Einschwingverhalten zeigt, starten wir die Simulation mit der Aufzeichnung und simulieren 10ms mit einer Schrittweite von 10us.
- Bestätigt mit *OK* und fügt die Sumlationsansweisung dem schematischen hinzu

## Simulation und Analyse



- Klickt auf (run) und LTSpice startet die Simulation
- Fügt nun die Eingangsspannung sowie die Ausgangsspannung der Schaltung als Messpunkte hinzu.

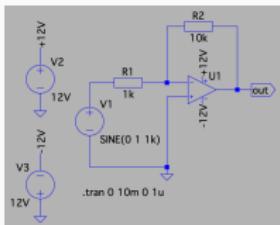
## Ergebnis und Auswertung

Verifiziert, ob die Verstärkung sowie das invertierende Verhalten zu eurem Erwartungswert (10) passt.

# OPV Schaltungen - transient, nicht ideal

Im nächsten Experiment wollen wir die gleiche Schaltung mit einem "nicht idealen" Verstärker simulieren und die Verwendung von Labels verdeutlichen.

## Erstellung des Schaltplans



- Tauscht den opamp gegen das Bauteil **UniversalOpamp2** aus dem Bauteileditor (**F2**).
- Fügt zwei neue Spannungsquellen für die Versorgung des OPV's hinzu.
- Achtet hierbei auf darauf, dass die Label wirklich auf dem Potential +12V und -12V liegen.
- Erstellt die Labels +12V und -12V (**Label Net (F4)**).

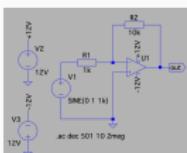
## Ergebnis und Auswertung

Es muss das selbe Ergebnis herauskommen wie im vorherigen Experiment unter Verwendung des idealen OPV's **opamp**.

# OPV Schaltungen - Analyse der Grenzfrequenz einer Schaltung

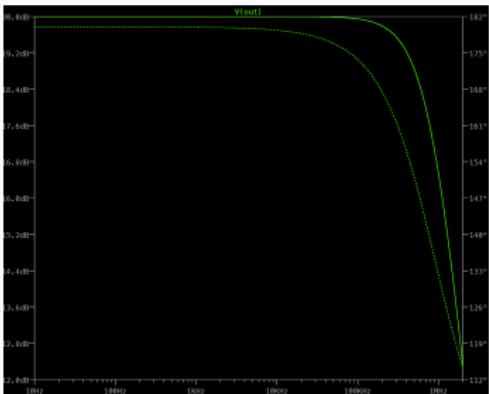
Eine Kennzahl von Tief-/Hochpassfiltern ist ihre 3dB Grenzfrequenz  $f_g$ . Die Grenzfrequenz kann man analytisch, jedoch auch simulativ über LTSpice bestimmen. Hierzu bietet LTSpice die Möglichkeit die Frequenz einer Schaltung zu variieren. Dies wird AC-Sweep genannt. Im Bode-Diagramm kann man den logarithmischen Verlauf der Amplitude über der Frequenz in LTSpice darstellen und somit einfach grafisch die Grenzfrequenz bestimmen indem man den Punkt heraussucht, bei dem die Amplitude um 3dB abgefallen ist.

## Erstellung des Schaltplans + Konfiguration der Simulation



- Wir verwenden die Schaltung aus dem vorherigen Beispiel!
- Wichtig ist hierbei, dass wir Spannungsquelle für den AC-Sweep konfigurieren. Hierzu geht ihr per rechtem Mausklick in das advanced menu von V1 und stellt das Kleinsignal Verhalten (AC) auf **Amplitude 1V und Frequenz 1kHz**.

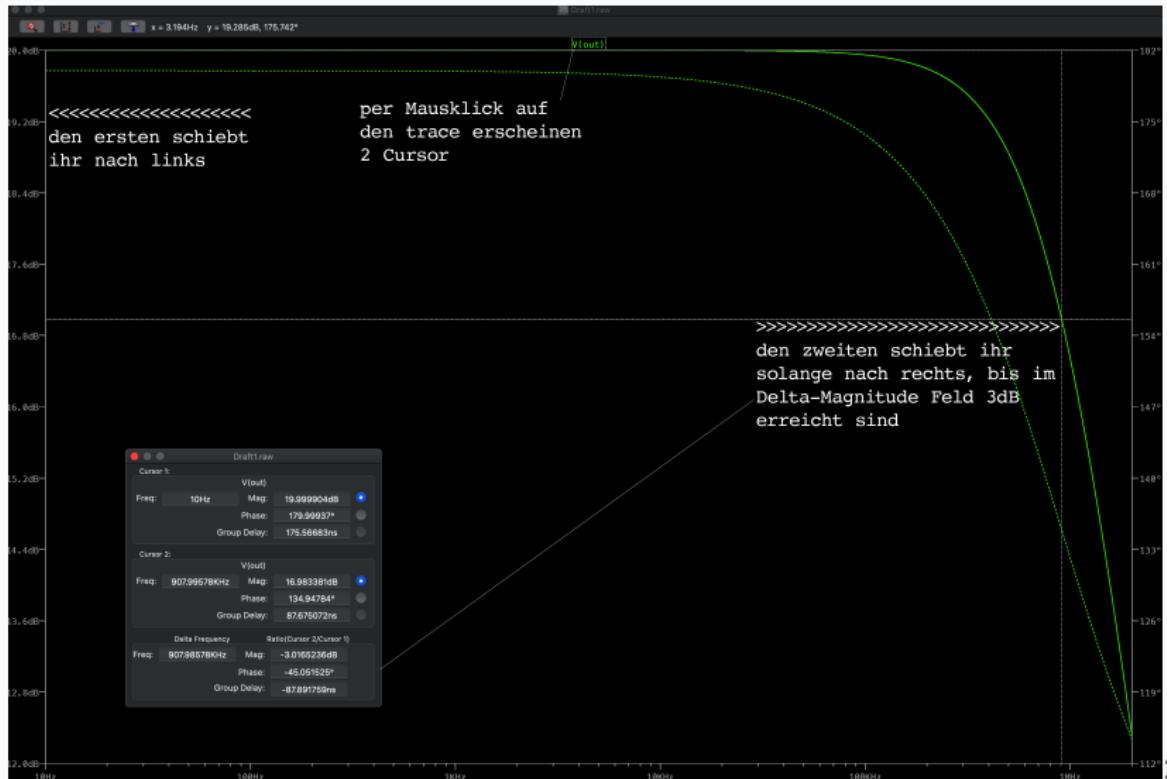
## Simulation und Analyse



- Klickt auf (run) und LTSpice startet die Simulation
- Fügt nun die Ausgangsspannung der Schaltung als Messpunkt hinzu.
- Im waveform viewer solltet ihr das Bode-Diagramm mit Amplitude und Phase sehen.

# OPV Schaltungen - Analyse der Grenzfrequenz einer Schaltung

## Simulation und Analyse

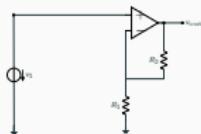


# OPV Schaltungen - nicht-invertierender Verstärker

## Ziel - Anwendung der Kenntnisse

Nachfolgend könnt ihr die Schaltung für einen nicht-invertierenden Verstärker in einer OPV Schaltung sehen.

1. Baut die Schaltung auf, dimensioniert sie so, dass Sie einen Verstärkungsfaktor
2. Verwendet die UniversalOpamp2 aus der vorherigen Übung mit einer Versorgungsspannung von +/-12V
3. Verifiziert eure Schaltung und das **nicht-invertierende** Verhalten mit einer transienten Simulation von 0 - 50ms.
4. Wählt dabei eine ausreichend kleine Schrittweite
5. Wechselt die Simulationsart zum AC-Sweep und ermittelt die Grenzfrequenz



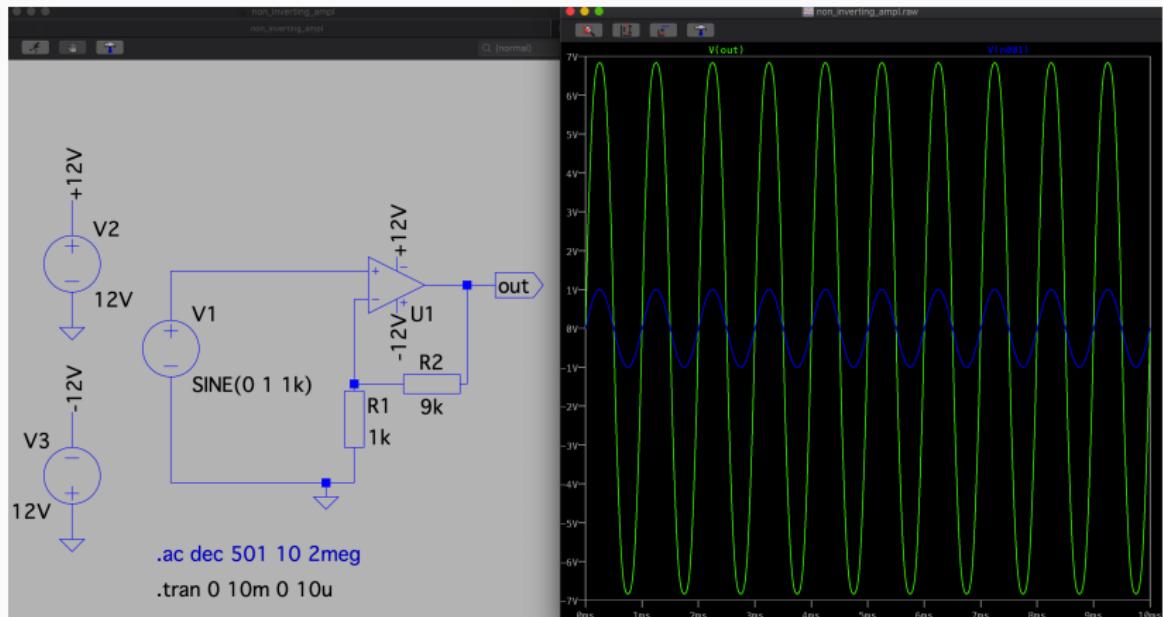
$$V_{out} = \left(1 + \frac{R_2}{R_1}\right) V_1 \quad (2)$$

Spoiler! - Die Lösungen folgen auf den folgenden Folien. Wenn sie sich nicht sicher sind, schauen Sie einfach nach.

Info! - Besonders die Analyse von Grenzfrequenzen ist für Ihre weitere Vorlesungszeit (Filter höherer Ordnung, EMV) hilfreich, da Sie neben der simulativen Bestätigung Ihrer Schaltungen auch theoretische Rechenaufgaben simulieren und so Ihre Rechnung verifizieren können.

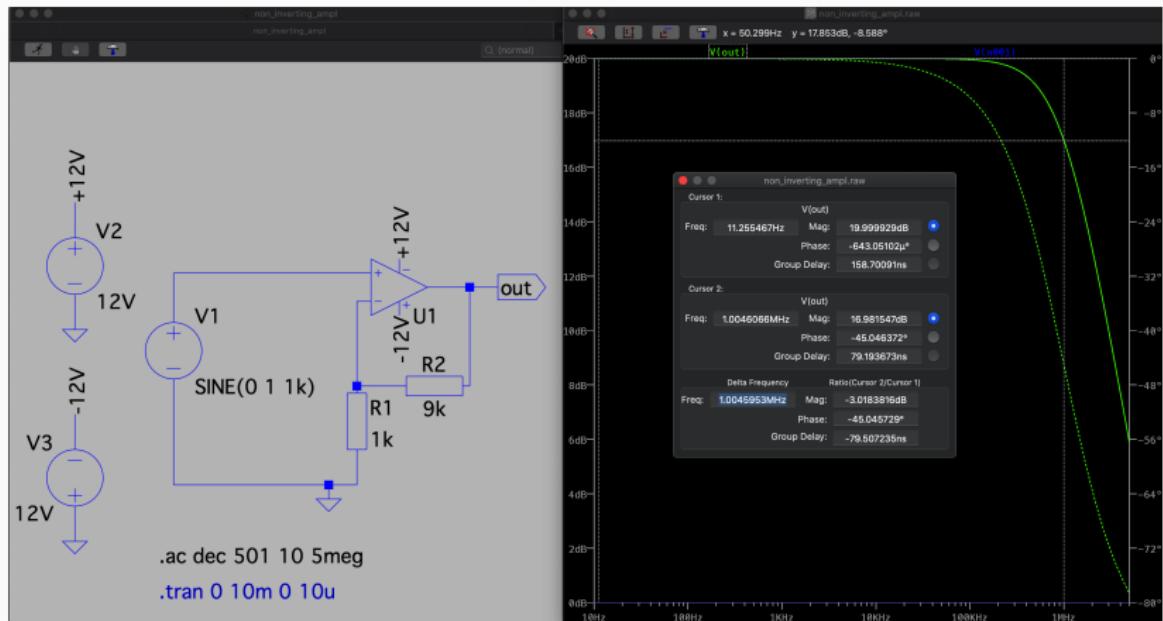
# OPV Schaltungen - Lösung transient

## Simulation und Analyse



# OPV Schaltungen - Lösung Grenzfrequenz

## Simulation und Analyse



## **Projekt - Temperaturmessbrücke**

---