

Software: SimX - Einfuehrung - Elektro-Chaos

Aus OptiYummy

↑

← →

Nichtlinearer elektrischer Schwingkreis (Chaos-Simulation)

Autor: Dr.-Ing. Alfred Kamusella

*Auch ein perfektes Chaos ist etwas Vollkommenes.
- Jean Genet -*

Sommerkurs Elektrotechnik 2012 für Studienanfänger:
"Informatik - Numerische Simulation nichtlinearer Systeme"

Aufbauend auf dem Schulwissen zu den elektrischen Bauelementen erfolgt am Beispiel der Kapazität (Kondensator) und Induktivität (Spule) eine schrittweise Einführung in die numerische Simulation nichtlinearer Systeme. Dabei werden u.a. folgende Fragen behandelt:

1. Wozu dienen numerische Modelle und Simulationen?
2. Was sind numerische Experimente?
3. Wie programmiert man Netzwerk-Elemente?
4. Wie findet man Parameter für nichtlineare Modelle?
5. Was ist das Besondere an nichtlinearen Systemen?
6. Wann entsteht Chaos in nichtlinearen Systemen?

Die benötigte Software ist im benutzten PC-Kabinett vorhanden. Eine Installation auf dem eigenen PC/Notebook wird für das Üben außerhalb der Kurszeiten empfohlen. Die Teilnehmer erhalten die Software vom Kurs-Betreuer. Der eigenständige Download von den Herstellerseiten ist möglich:

1. **SimulationX**-Studentenversion (Systemsimulation)
2. **OptiY**-Trialversion (Multidisziplinäre Analyse und Optimierung)

Übungsabschnitte:

1. Lineare Elemente und Systeme

- . Elektrische Kapazität (Kondensator)
- . Elektrische Induktivität (Luftspule)
- . Elektrischer Schwingkreis
- . Sinus-Oszillator (Schaltungssimulation) -> bei Zeitverzug überspringen!
- . Modellierung eigener Elemente
- . Modellierung und Simulation linearer Systeme (Zusammenfassung)

2. Nichtlineare Elemente und Systeme

- . Dioden-Kapazität mit Datenblatt-Kennlinie
- . Parameter-Identifikation (Spice-Modell der Diode)
- . Nichtlinearer Schwingkreis mit C-Diode (Chaos-Experimente)

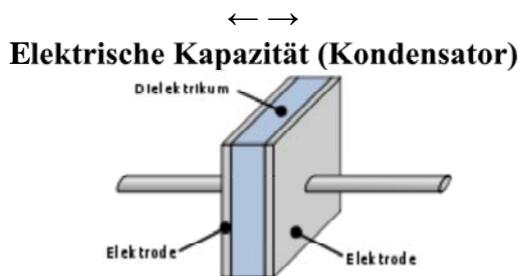
← →

Von „<http://optiyummy.de/index.php?title=Software: SimX - Einfuehrung - Elektro-Chaos>“

Software: SimX - Einfuehrung - Elektro-Chaos - Kondensator

Aus OptiYummy

↑



Ein elektrischer Kondensator besteht aus zwei elektrisch leitenden Flächen in meist geringem Abstand (Elektroden). Dazwischen befindet sich immer ein Bereich mit isolierender Eigenschaft (Dielektrikum). Ein Kondensator kann auf diesen Elektroden elektrische Ladungen speichern. Dadurch entsteht zwischen den Elektroden im Dielektrikum ein elektrisches Feld.

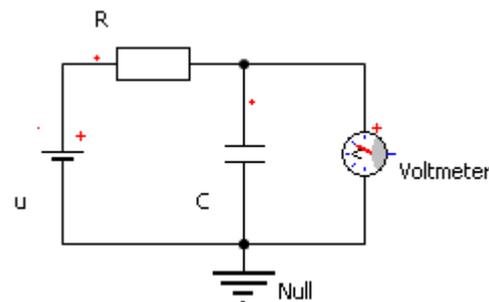
Die Fähigkeit, elektrische Energie in Form von Ladung und daraus resultierendem elektrischen Feld zu speichern, wird als elektrische Kapazität bezeichnet und in der Einheit Farad (**F**) gemessen ($1 \text{ F} = 1 \text{ A} \cdot \text{s/V}$). Ein Kondensator besitzt eine Kapazität von 1 F, wenn sich bei einem konstanten Strom von 1 A die Spannung an den Anschlüssen um 1 V in einer 1 s verändert.

Am Beispiel eines einfachen Experiments mit dem elektrischen Kondensator werden wir zuerst die Nutzung des Simulationssystems *SimulationX* erkunden:

Wir untersuchen die Aufladung eines Kondensators nach dem Anschließen einer Spannungsquelle. Damit es nicht zu sehr "funkelt", lassen wir den Aufladestrom über einen in Reihe geschalteten Widerstand fließen.

Würden wir diese Schaltung als "materiellen" Versuchsstand aufbauen, so könnte man bei einer hinreichend großen Kondensator-Kapazität am Voltmeter beobachten, wie sich der Kondensator bis zur Ausgangsspannung des Netzteils auflädt. Solche Versuche sollten aus dem Physikunterricht bekannt sein.

Nur so erhält man die Chance, ein Gefühl dafür entwickeln, dass es z.B. "funkelt", wenn man den Reihenwiderstand vergisst. Und dass es gewaltig "funkelt", wenn man den aufgeladenen Kondensator mit einem Draht kurzschließt. Außerdem erhöht die Reaktion des Physiklehrers auf solche Ereignisse zusätzlich die soziale Kompetenz des Experimentierenden.



Inhaltsverzeichnis

- . 1 Simulationssoftware
- . 2 Modell des Versuchsstands
- . 3 Konfiguration des Experiments
- . 4 Experimente
- . 5 Zusammenfassung

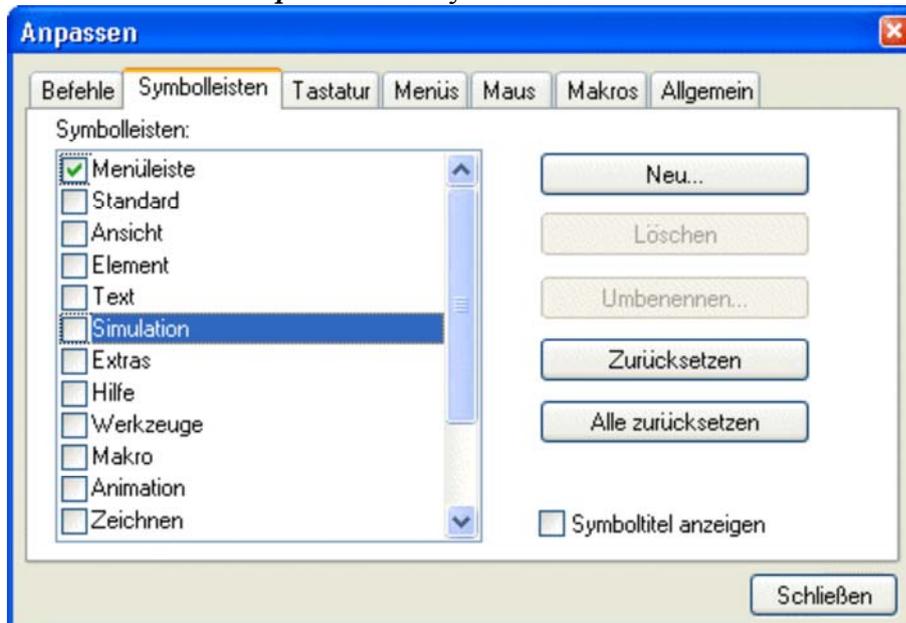
Simulationssoftware

Wir beschränken uns in dieser Übung auf "virtuelle" Experimente mit numerischen Modellen des Kondensators und der restlichen elektronischen Schaltung des Versuchsstandes. Grundlage dafür ist ein geeignetes Simulationssystem. Mit der Studenten-Lizenz von **SimulationX der Firma ITI GmbH Dresden** steht uns ein

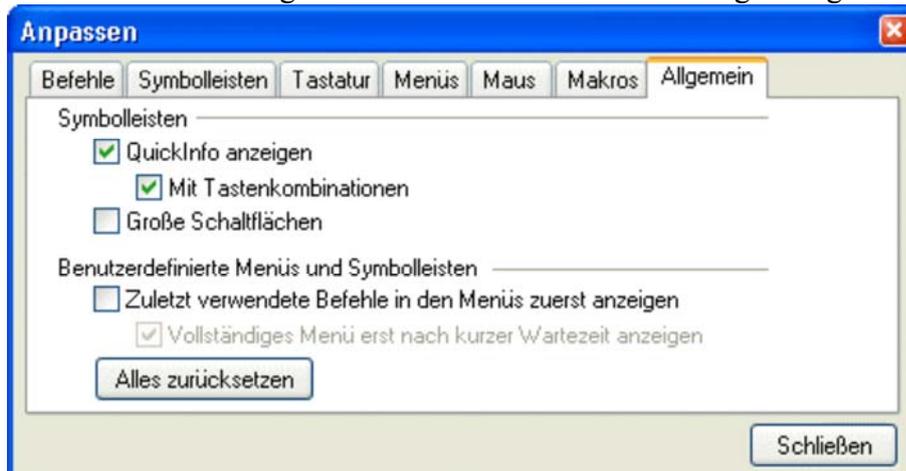
modernes System zum Modellieren, Simulieren und Analysieren von physikalischen Effekten zur Verfügung.

Achtung: Nach dem Start von SimulationX zuerst Benutzeroberfläche anpassen!

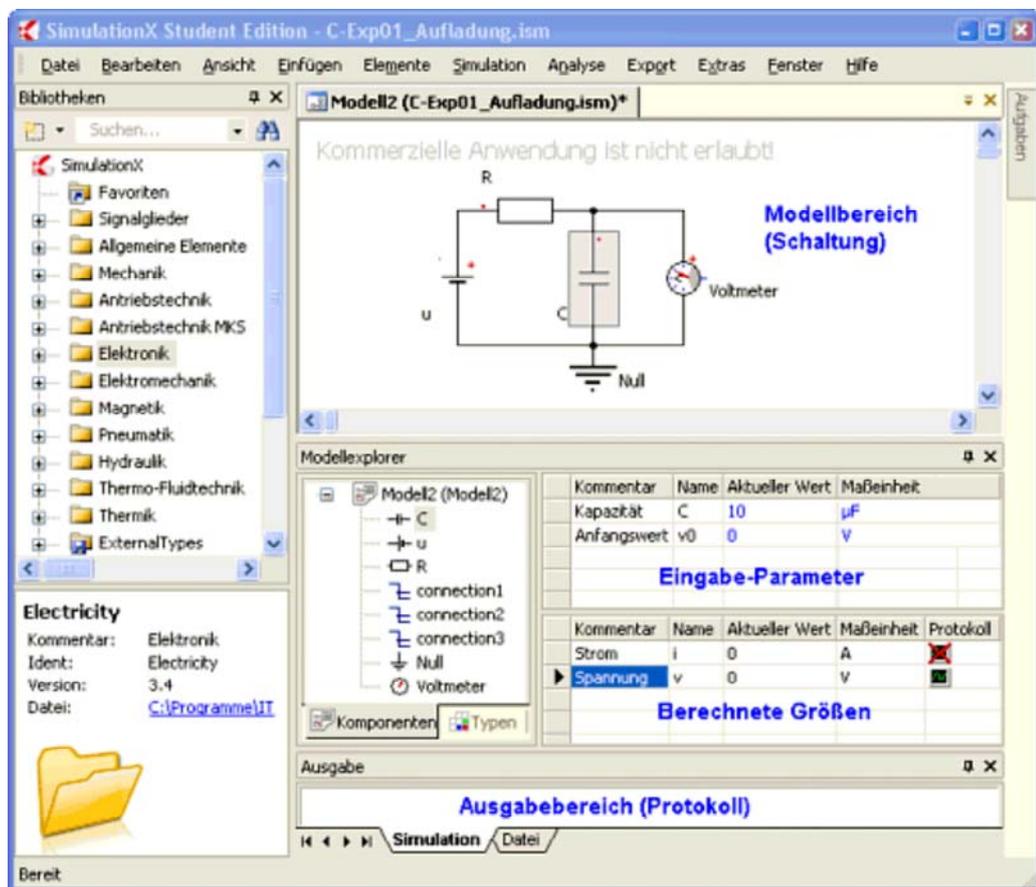
- Hat man noch keine individuellen Anpassungen der Benutzeroberfläche vorgenommen, so wirkt diese infolge einer Vielzahl von Werkzeugleisten sehr überladen.
- Deshalb sollte man mittels **Extras > Anpassen** alle Symbolleisten außer der Menüleiste ausblenden:



- Günstig ist es, in diesem Zusammenhang auch die Menüs immer vollständig anzeigen zu lassen:



- Die Programmoberfläche hat in der "aufgeräumten" Grundeinstellung folgende wesentlichen Bereiche:



In der **Modellbibliothek** findet man ein reichhaltiges Angebot an Modell-Elementen aus allen möglichen Fachgebieten.

In den **Modellbereich** kann man diese Elemente einfach mit dem Cursor ziehen:

Hinweis: Obige Schaltung bauen wir nachher systematisch auf! Wir beschränken uns jetzt auf ein paar spielerische Aktionen.

Wie in einem "echten" Versuchsstand kann man die Elemente miteinander verbinden.

Zum Verbinden nutzt man wieder den Cursor.

Das gelingt aber nur, wenn die Element-Anschlüsse zueinander passen (auch in der wirklichen Welt funktioniert eine elektrische Leitung meist nicht besonders gut als Wasserleitung!)

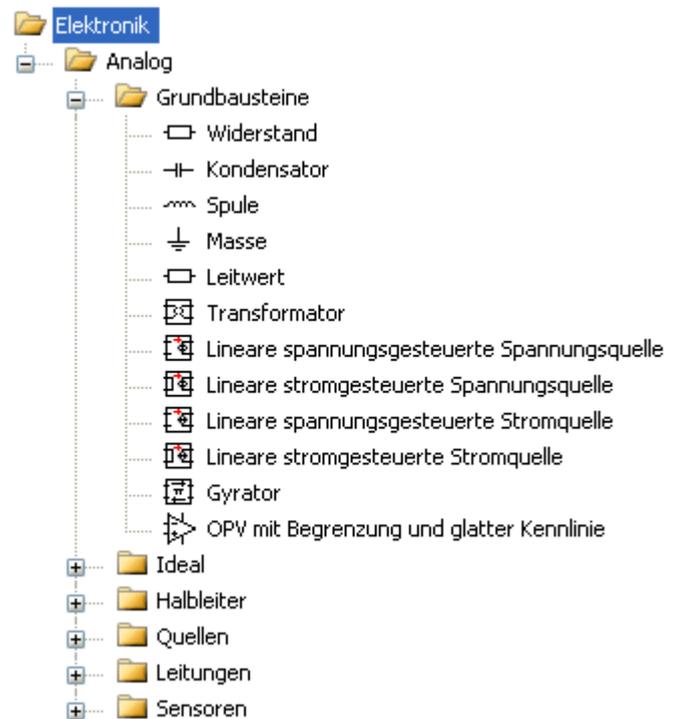
Im **Modellexplorer** erscheint das aufgebaute Modell als Baumstruktur:

Die Form dieser Darstellung entspricht der Ordneransicht im Windowexplorer.

In dieser Baumstruktur werden die Elemente und die Verbindungen abgebildet.

Wenn man ein Element in der Baumstruktur oder in der Modellansicht auswählt, erscheinen die zugehörigen Informationen rechts neben der Baumstruktur.

Im **Ausgabebereich** erscheinen z.B. Fehlermeldungen bei der Arbeit mit dem Modell (hoffentlich nicht!).



Modell des Versuchsstands

Damit es nun richtig losgehen kann, müssen wir zuvor unseren Modell-Ansichtsbereich entrümpeln, falls wir zum Ausprobieren dort ein paar Modell-Element verknüpft hatten:

Die Modell-Elemente kann man einfach löschen.

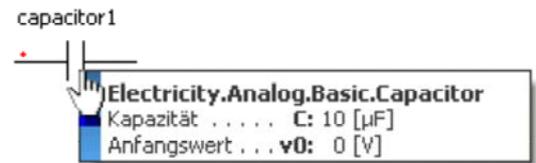
Besser ist es, man schließt das Modell (**Datei > Schließen**) und öffnet ein neues Modell (**Datei > Neu**).

Wir speichern das Modell (*Datei > Speichern unter*) und wählen dafür einen sinnvollen Namen (z.B. **RC-Glied**).

Der Aufbau des numerischen Modells einer elektronischen Schaltung erfolgt prinzipiell genauso, wie bei einer realen Schaltung -> man holt sich die benötigten Bauelemente mit den richtigen Kennwerten und verbindet sie miteinander:

Wir beginnen mit dem Kondensator, den wir aus der Bibliothek (*Elektronik > Analog > Grundbausteine*) in den Modellbereich ziehen.

Der Kondensator besitzt noch nicht den richtigen Bezeichner **C** und ist um 90° zur gewünschten Lage verdreht. Seine aktuelle Kapazität erkennt man, wenn man den Cursor darüber bewegt:



Namensänderung durch Mausklick in das Namensfeld und Editieren.

Drehen des Symbols in der Schaltung z.B. mittels *Elemente > Nach rechts drehen*. Für ungepolte Bauelemente ist die Einbaurichtung in die Schaltung egal. Die Markierung einer "positiven" Seite hat nur Auswirkungen auf das Vorzeichen von Strom und Spannung des Elements, ändert aber nichts an der Funktion.

Kapazitätswert nach Doppelklick auf das Symbol erscheint der Eigenschaftsdialog. In der Registerkarte *Parameter* erhöhen wir den Wert für **C** auf **1000 µF**. Der Anfangswert **v0=0 V** definiert einen entladenen Kondensator.



Wichtig:

Jede physikalische Domäne (z.B. Elektronik, Mechanik, Magnetik) benötigt ein definiertes Nullpotential. Damit man dieses Nullpotential nicht vergisst, sollte man das zugehörige Element möglichst frühzeitig im Modell definieren.

Eine günstige Lage für die Beschriftung eines Elements kann man im *Eigenschaftsdialog > Allgemein* festlegen:



Das Verbinden der Bauelemente-Anschlüsse kann man intuitiv durch Ziehen mit dem Cursor erledigen (linke Maustaste dabei drücken!).

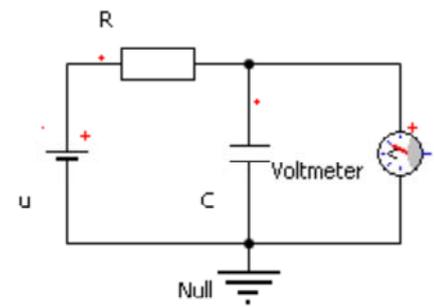
Wir vervollständigen nun die Schaltung mit folgenden Bauelementen:

1. Netzteil $u = 10 \text{ V}$ (*Konstantspannungsquelle*)
2. Widerstand $R = 1 \text{ k}\Omega$

3. Voltmeter (*Spannungssensor*)

Konfiguration des Experiments

Wir wollen den Aufladevorgang des entladenen Kondensators nach dem Anschließen der Spannungsquelle untersuchen:



Im realen Experiment könnte man das Ansteigen der Kondensator-Spannung am Voltmeter beobachten (wenn die Spannungsänderung hinreichend langsam erfolgt). Diese Beobachtung wäre zwar anschaulich, aber nur qualitativ, da man es kaum schafft, zu bestimmten Zeitpunkten die zugehörigen Spannungswerte zu notieren.

Deshalb benutzt man in der Praxis ein digitales Oszilloskop, um für alle interessierenden Stellen an einer Schaltung beliebig schnelle Spannungsverläufe aufzuzeichnen.

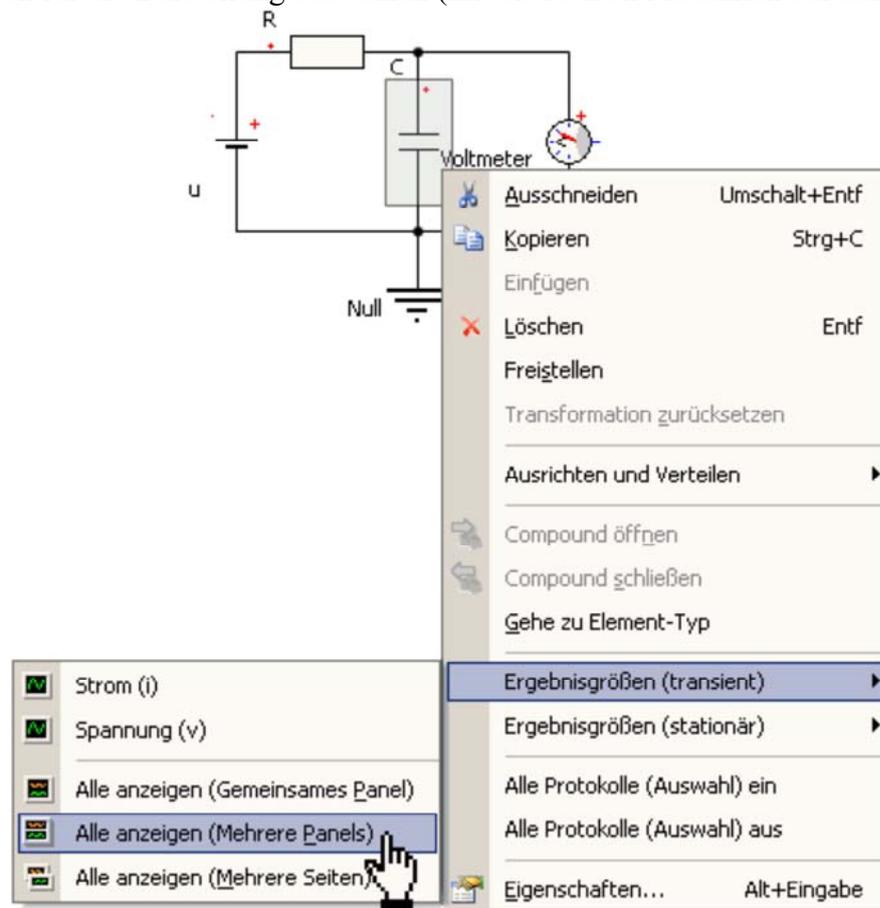
Ein Simulationsprogramm kann nicht nur das zeitliche Verhalten des numerischen Modell unserer elektronischen Schaltung berechnen. Diese Software bietet auch Möglichkeiten, alle berechneten Signalverläufe genauso wie ein digitales Oszilloskop aufzuzeichnen.

Wenn wir im Beispiel den Strom durch den Kondensator und die Spannung am Kondensator aufzeichnen möchten, so müssen wir diese Funktion zuvor aktivieren (**Kondensator > Eigenschaften > Ergebnisgrößen**):



Es werden nur die Signale aufgezeichnet, welche man dafür freigeschaltet hat.

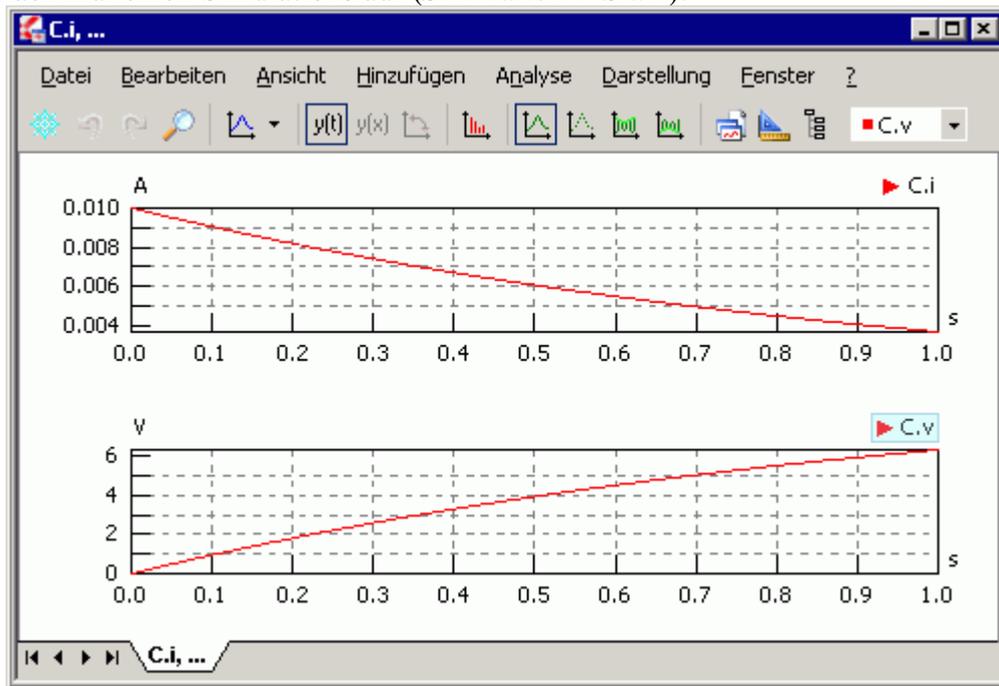
Damit man den Verlauf aufgezeichneter Signalgrößen bei der Simulation betrachten kann, muss man zuvor diese Ergebnisgrößen für die Darstellung auswählen (mittels rechten Mausklick auf den Kondensator):



Eine günstige Form ist hier die Darstellung beider Signale in einem Fenster auf einer Seite, aber jeweils in einem eigenen Panel.

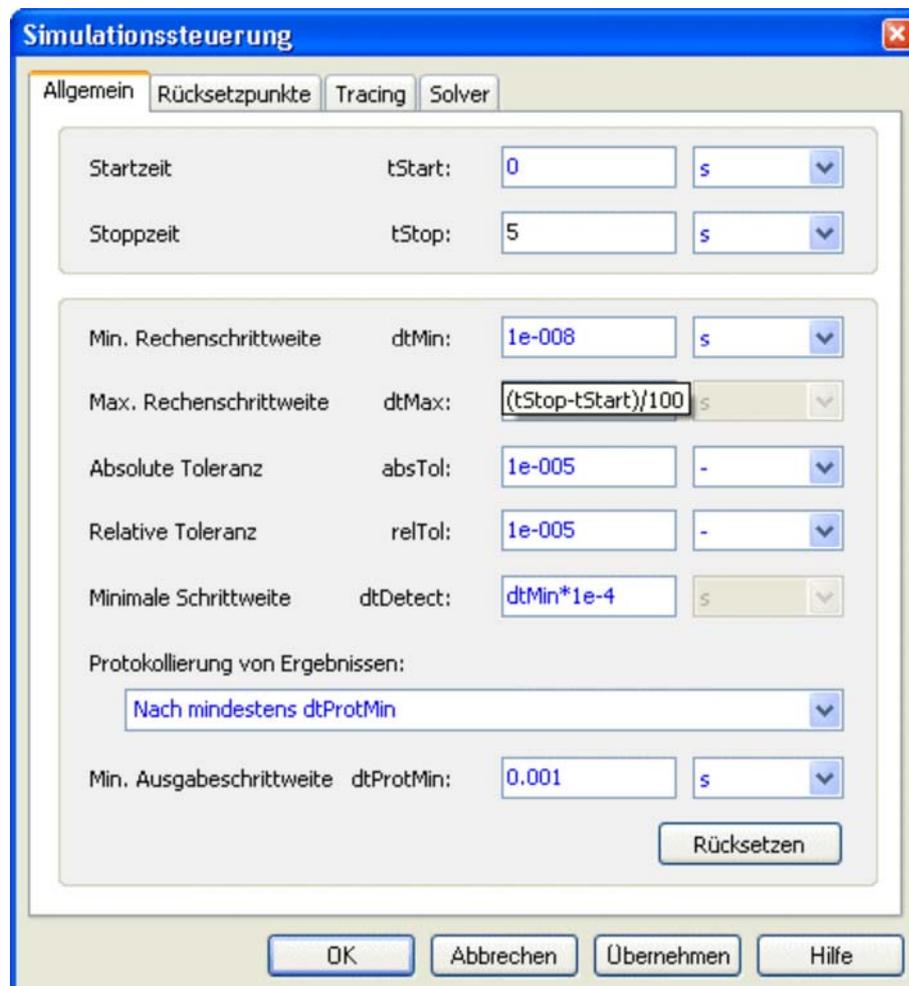
Da wir noch keine Simulation gestartet hatten, erscheint zunächst ein Fenster mit zwei leeren Signal-Panelen.

Wir starten einfach mal einen Simulationslauf (*Simulation > Start*):



Standardmäßig wird ein Zeitbereich von **tStart=0 s** bis **tStop=1 s** simuliert. In unserem Beispiel ist in dieser Zeit der Aufladevorgang des Kondensators noch nicht beendet.

Wir erhöhen mit der Simulationssteuerung (*Simulation > Einstellungen Transient*) die Simulationszeit auf **tStop=5 s**:



Dabei werfen wir gleich noch einen Blick auf einige andere Parameter für einen Simulationslauf:

Die Simulation im Zeitbereich erfolgt mit einer variablen Rechenschrittweite **dt**. Diese wird automatisch so angepasst, dass sie immer möglichst groß ist und trotzdem noch eine hinreichend genaue Simulation erfolgt:

dtMin ist die kleinste zulässige Schrittweite. Bei stark unetigen Modellen kann man diesen Bedarf bis ca. $1e-20$ s verringern. Wir verringern die Standardvorgabe von $1e-8$ s auf **dtMin=1e-12 s**.

dtMax ist die größte zulässige Schrittweite. Diese sollte man auf einen Wert in Abhängigkeit vom Zeitbereich begrenzen. Im Beispiel ist gewährleistet, dass der gesamte Zeitbereich immer durch mindestens 100 berechnete Punkte abgebildet wird (**dtMax=(tStop-tStart)/100**). Signalverläufe werden durch Geraden zwischen den berechneten Punkten approximiert. Gekrümmte Signalverläufe erscheinen bei zu großen Rechenschrittweiten als eckig, obwohl die einzelnen Punkte hinreichend genau berechnet werden.

dtProtMin ist die kleinste Ausgabeschrittweite. Standardmäßig wird hier ein Absolutwert $dtProtMin=0.001$ s vorgegeben. Das stört für den gewählten Zeitbereich nicht. Bei einem Zeitbereich von z.B. 5 ms würden jedoch nur 5 Berechnungspunkte dargestellt! Deshalb sollte man $dtProtMin$ grundsätzlich wie $dtMax$ auf den Zeitbereich beziehen: z.B. gewährleistet **dtProtMin=(tStop-tStart)/500** unabhängig vom Zeitbereich meist eine hinreichend exakte Darstellung der Signalverläufe bei moderater Punktdichte.

Experimente

Die Zeitkonstante des RC-Gliedes beträgt

$$\tau = R \cdot C$$

Der Anstieg der Spannung erfolgt bei der Ladung des Kondensators mit

$$u(t) = U_{\max} \cdot (1 - e^{-\frac{t}{\tau}})$$

Markante Näherungswerte für die erreichte Aufladespannung sind dabei:

$$\begin{aligned} 1 \cdot \tau &\approx 63,2 \% \\ 3 \cdot \tau &\approx 95,0 \% \\ 5 \cdot \tau &\approx 99,3 \% \end{aligned}$$

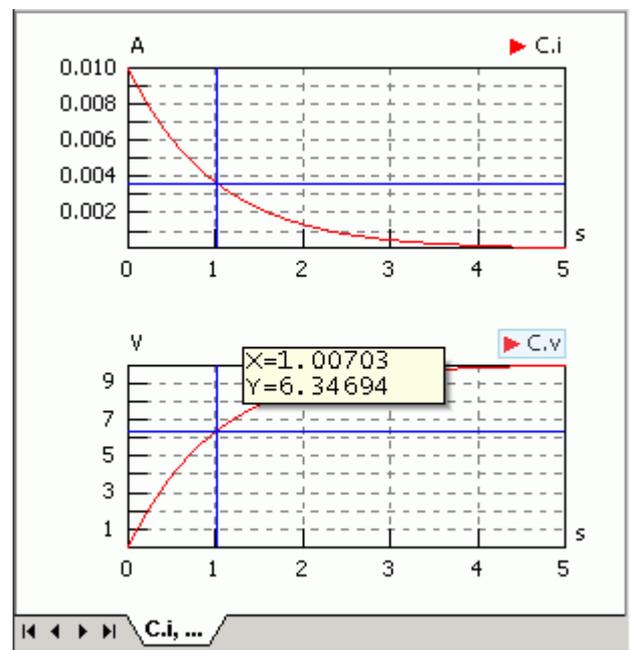
Nach der 5-fachen Zeitkonstante ist ein Kondensator praktisch vollständig aufgeladen.

Mit unseren konkreten Werten

$$\begin{aligned} R &= 1 \text{ k}\Omega = 1000 \text{ V/A} \\ C &= 1000 \text{ }\mu\text{F} = 0.001 \text{ A}\cdot\text{s/V} \end{aligned}$$

ergibt sich eine Zeitkonstante von

$$\tau = 1 \text{ s}$$



Das hatten wir auf Grund des Signalverlaufs der Kondensatorspannung bereits vermutet. Wir überprüfen nun die markanten Näherungswerte innerhalb des Signalverlaufs durch Messen mit dem Cursor:

Mit gedrückter linker Maustaste bewegen wir uns dazu im Signal-Panell **C.v**.

Es erscheint ein Hinweis: *"Halten Sie die Strg-Taste gedrückt, um nur die tatsächlich berechneten Punkte zu vermessen"*. Diesem Rat werden wir folgen.

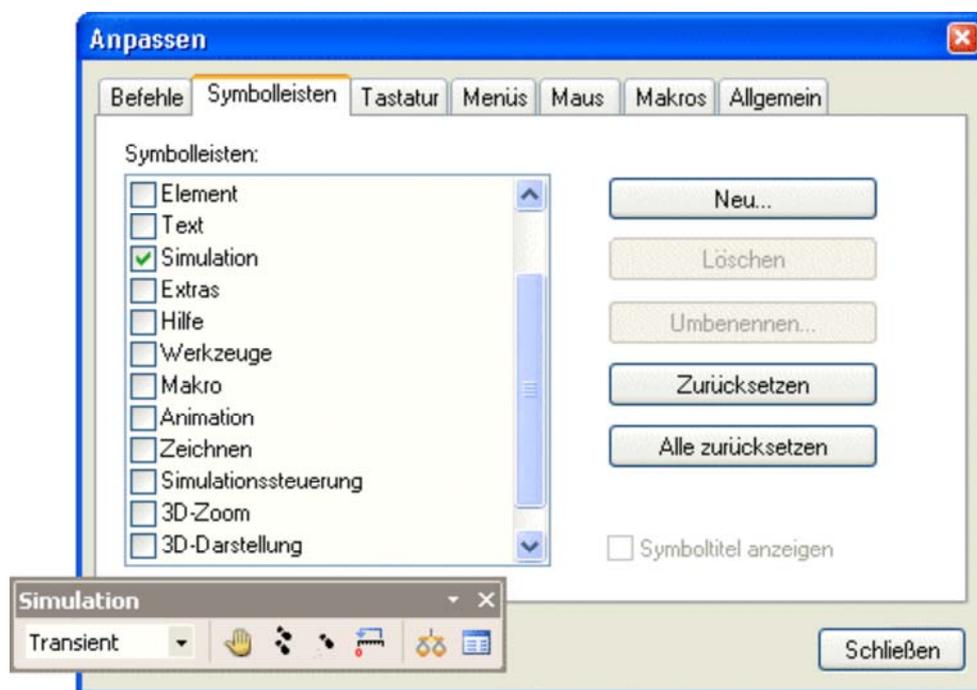
Die zugehörigen X- und Y-Werte erscheinen in einem eingeblendeten Info-Fenster.

Dabei werden wir feststellen, dass wir gewünschte "exakte" Zeitpunkte (z.B. 1 s) mit dem Cursor nicht ansteuern können! Auf Grund der variablen Simulationsschrittweite dt ist das auch sehr unwahrscheinlich. Solche Zeitpunkte müsste man innerhalb des Modells zuvor als "Ereignisse" definieren.

Wir nutzen unser bisheriges Wissen nun, um durch eine Veränderung der Modellparameter den Entladevorgang des Kondensators zu untersuchen:

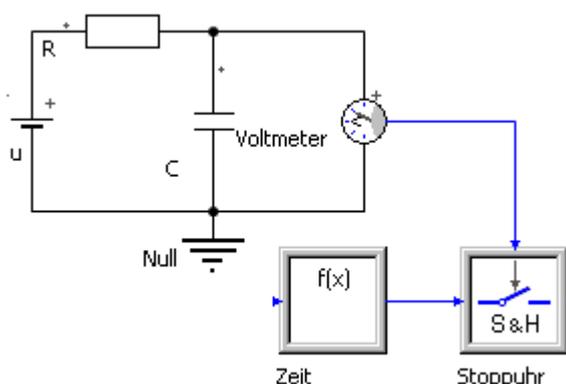
- Zum Zeitpunkt t_{Start} soll der Kondensator eine Anfangsspannung von **10 V** besitzen.
- Die Ausgangsspannung des Netzteils soll **0 V** betragen.
- Ein Rücksetzen und Starten der Simulation wird nun den Entladevorgang zeigen.

Hinweis: Da wir nun häufiger Simulationsläufe starten, sollten wir den Toolbar für die Simulation benutzen (**Extras > Anpassen > Symbolleisten > Simulation**). Nicht den Toolbar "Simulationssteuerung" benutzen!



Für einen vorgegebenen Grenzwert der Kondensator-Spannung soll der "exakte" Zeitpunkt erfasst werden. Manuell durch Ausmessen der Signalverläufe gelingt das nicht:

- **Stoppuhr:** Wir erweitern unser Modell um ein Ereignisgesteuertes Abtastglied (*Bibliotheken > Signalglieder > Spezielle Signalglieder*). Dieses Signalglied speichert den Wert des Eingangssignals x , wenn das Steuersignal s einen vorgegebenen Grenzwert erreicht.
- Das Ausgangssignal des Voltmeters benutzen wir als Steuersignal s der Stoppuhr.
- **Zeit:** Als Eingangssignal x der Stoppuhr benötigen wir die aktuelle Simulationszeit t . Dafür benutzen wir ein Signalglied $f(x)$ mit $F=t$.
- Wir konfigurieren die neuen Modellelemente so, dass ein vorgegebener Spannungsgrenzwert unabhängig vom Auf- oder Entladen von der Stoppuhr erfasst wird.



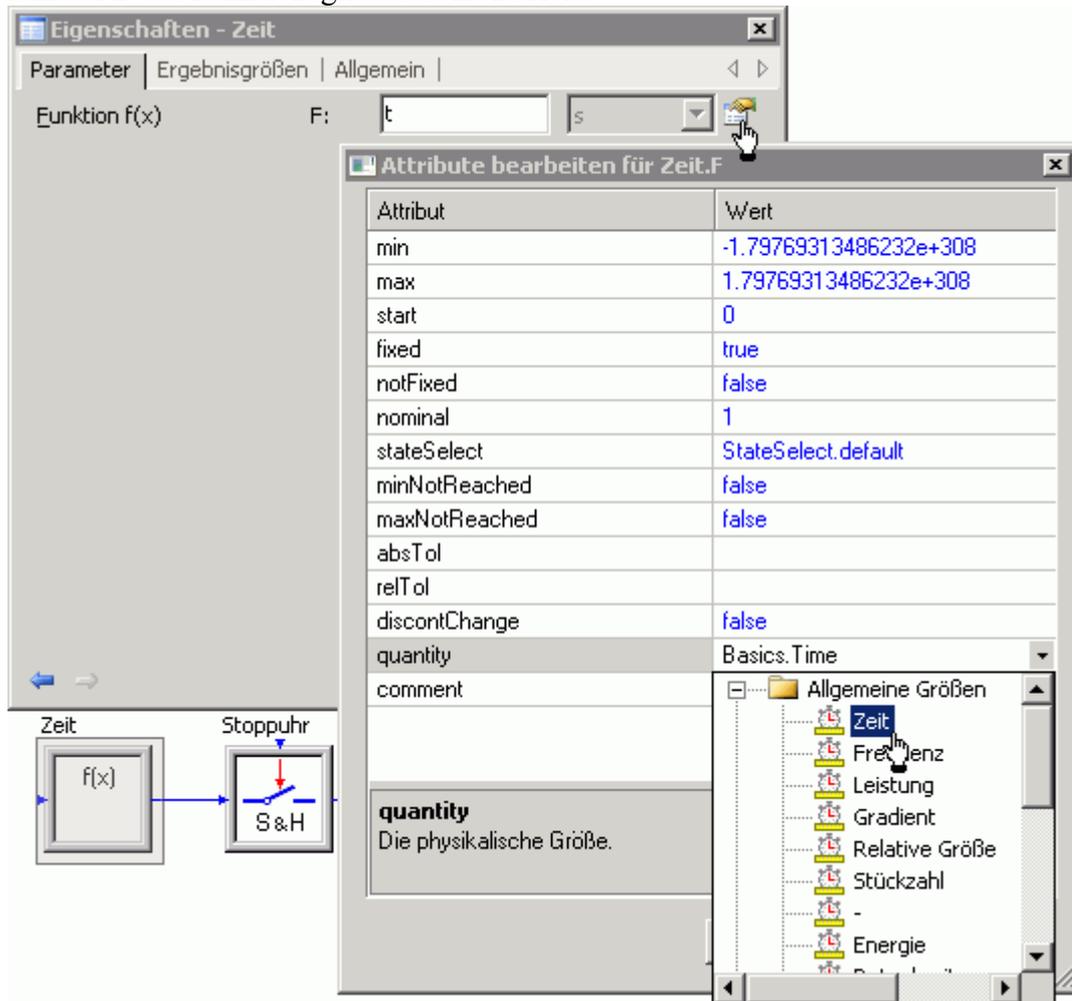
Signalglieder verwenden standardmäßig keine Maßeinheiten, sondern verarbeiten nur die reinen Signal-Zahlenwerte. Das hat mehrere Nachteile:

- Dargestellte Signalverläufe von Signalgliedern besitzen ebenfalls keine Einheiten.
- Es kann keine automatische Umrechnung der physikalischen Größen stattfinden (z.B. Sekunden in Milisekunden).
- Bei der Auswertung der Ergebnisse kann es häufiger zu Fehlinterpretationen kommen.

Deshalb sollte man es sich angewöhnen, physikalische Größen in Simulationsmodellen grundsätzlich mit den richtigen physikalischen Einheiten zu versehen:

Die Attribute der Parameter und Ergebnisgrößen der Funktionsglieder kann man bearbeiten.

"quantity" enthält die zugeordnete physikalische Größe. Im folgenden Bild wird das Zuweisen der "Zeit" zum Parameter **F** des Funktionsgliedes demonstriert:



In der gleichen Weise versehen wir die Ergebnisgröße **y** mit der Einheit der Zeit.

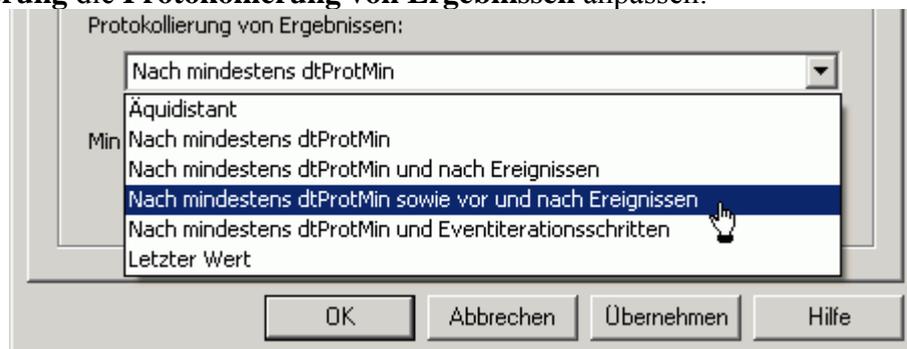
Der Grenzwert **a** der Stoppuhr soll die Einheit der elektrischen Spannung erhalten.

Der Anfangswert **y₀** für den Signalausgang **y** der Stoppuhr besitzt die Einheit der Zeit.

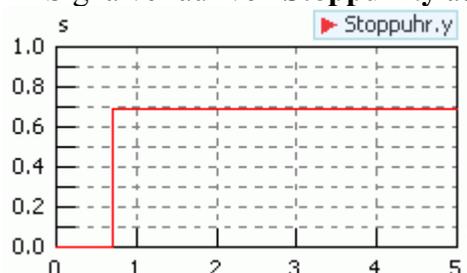
Gleiches gilt für den Signalausgang **y** der Stoppuhr.

Hinweise:

Damit der exakte Zeitwert auch in den Signalverläufen protokolliert wird, muss man in der **Simulationssteuerung** die **Protokollierung von Ergebnissen** anpassen:



Den Ereigniszeitpunkt könnte man im Signalverlauf von **Stoppuhr.y** ausmessen:



Einfacher ist es bei einem konstantem Endwert, diesen im Modellexplorer abzulesen:

Kommentar	Name	Aktueller Wert	Maßeinheit
Richtung	direction	beide Richtungen	
Nur erstes Ereignis...	firstevent	nein	
Grenzwert	a	5	V
Anfangswert	y0	0	s

Kommentar	Name	Aktueller Wert	Maßeinheit	Protokoll
Signalausgang	y	0.693144180646716	s	<input checked="" type="checkbox"/>

Der dargestellte Zeitpunkt entspricht einer Entladung auf 50%!

Frage 1:

Zu welchem Zeitpunkt t wird für das Beispiel exakt eine Kondensatorspannung von **75%** des Anfangswertes beim Entladen erreicht?

Zusammenfassung

Am Beispiel von Auf-/Entladevorgängen eines elektrischen Kondensators haben wir Erfahrungen zum Prozess der Dynamiksimulation elektrischer Netzwerke gesammelt.

Daraus kann man folgende Verallgemeinerungen ableiten:

1. **Numerische Modelle** nutzt man als Ersatzobjekte, um damit Erkenntnisse über die Originalobjekte zu gewinnen.
2. **Simulationen** führt man durch, um mit Modellen die gewünschten Erkenntnisse zu gewinnen.
3. **Modell-Versuchsstände** enthalten das Modell des zu untersuchenden realen Objekts und ein Modell der Experiment-Umgebung (z.B. zur automatisierten Experiment-Auswertung).
4. **Modell-Experimente** benutzen speziell konfigurierte Modell-Versuchsstände, um damit zielgerichtet bestimmte Fragen zu beantworten.

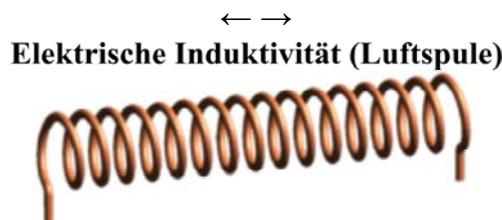
← →

Von „[http://optiyummy.de/index.php?title=Software: SimX - Einfuehrung - Elektro-Chaos - Kondensator](http://optiyummy.de/index.php?title=Software:_SimX_-_Einfuehrung_-_Elektro-Chaos_-_Kondensator)“

Software: SimX - Einfuehrung - Elektro-Chaos - Luftspule

Aus OptiYummy

↑



Eine elektrische Induktivität wird meist durch einen stromdurchflossenen Leiter realisiert. Infolge des Stromflusses wird ein magnetisches Feld um den Leiter aufgebaut. Eine Induktivität speichert die eingespeiste elektrische Energie in Form der Energie des resultierenden magnetischen Feldes.

Da der Mensch magnetische Felder nicht direkt wahrnehmen kann, sind sie "unanschaulicher" als elektrische Felder, deren Wirkung bestimmt schon jeder in irgendeiner Form gespürt hat.

Magnetfelder sind uns neben beobachtbarer Kraftwirkungen vor allem über ihre Wechselwirkung mit den elektrischen Feldern zugänglich. Ein Effekt der Induktivität ist z. B. die Selbstinduktionsspannung entlang des Leiters, wenn sich der durchfließende Strom ändert. Daraus leitet sich auch die Einheit Henry (**H**) ab ($1 \text{ H} = 1 \text{ V}\cdot\text{s}/\text{A}$). Eine Spule besitzt eine Induktivität von 1 H, wenn sich bei einer konstanten Stromänderung von 1 A/s ein Spannungsabfall von 1 V an den Spulenanschlüssen ergibt.

Der Wert der Induktivität einer Spule ist abhängig von der Permeabilität des Materials, in welchem sich das Magnetfeld aufbaut. Einen wesentlichen Einfluss auf die Vergrößerung der resultierenden Induktivität haben dabei nur ferromagnetische Stoffe. Da die Permeabilität dieser Stoffe von der Stärke des Magnetfeldes abhängt und zusätzlich eine Hysterese aufweist, führt dies jedoch zu nichtlinearen Induktivitäten. Deren Induktivitätswert ist abhängig von der Stärke des fließenden Stromes und der Vorgeschichte.

Wir beschränken uns in dieser Übung auf lineare Induktivitäten, deren Wert sich nicht in Abhängigkeit von den Einsatzbedingungen ändert. Das ist bei Luftspulen gegeben, weil sich die Permeabilität von Luft unabhängig von Feldstärken, Temperatur, Feuchtigkeit und Luftdruck praktisch nicht ändert (im Bereich von $1e-5$).

Einschalt-Experiment

Analog zum Auflade-Experiment mit dem RC-Glied untersuchen wir den Anstieg des Spulenstroms nach Anlegen einer Spannung an ein RL-Glied. Dazu schließen wir unsere Modelldatei **RC-Glied.ism** und beginnen die Arbeit an einer neuen Datei **RL-Glied.ism**.

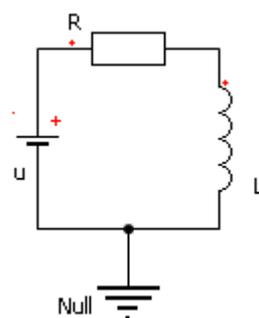


Um die Symbole beim Aufbau der Modellschaltung zu drehen, aktivieren wir zuvor die Element-Symbolleiste.

Die Zeitkonstante des RL-Glieds ist der Quotient aus der Induktivität und dem Widerstand:

$$\tau = \frac{L}{R}$$

Der Anstieg des Stromes erfolgt dabei analog zur Aufladespannung am Kondensator mit:



$$i(t) = \frac{U}{R} \cdot (1 - e^{-\frac{t}{\tau}})$$

τ gibt die Zeit an, nach welcher der Strom durch die Induktivität ca. 63,2 % des Endstromes erreicht hat. Nach $5 \cdot \tau$ hat der Strom etwa 99,3 % seines Endwertes erreicht, was praktisch dem Endstrom entspricht. Der Strom wird durch das ohmsche Gesetz auf $i=U/R$ begrenzt.

Mit Luftspulen ist es praktisch nicht möglich, eine Zeitkonstante von 1 s zu erreichen:

- . Die Induktivität steigt zwar quadratisch mit der Windungszahl. Trotz tausender Windungen sind Luftspulen mit 1 H technisch wegen ihrer Größe kaum realisierbar.
- . Der für die Spulenwindungen benötigte Draht besitzt einen ohmschen Widerstand, der als Bestandteil des Reihenwiderstand R zu betrachten ist. Es ist ohne die Nutzung von Supraleitung nicht möglich, Luftspulen hoher Induktivität mit einem Spulenwiderstand von wesentlich kleiner als 1 Ω zu realisieren.

Analog zum Auf- und Entlade-Experiment eines Kondensators führen wir selbständig die Experimente zum Ein- und Ausschalten einer Induktivität durch. Die interessierende dynamische Zustandsgröße ist jetzt die Spulenstrom. Wir gehen in folgenden Schritten vor:

1. Aufbau des Schaltungsmodells lt. obiger Skizze.
2. Dimensionierung des Widerstands R für eine Zeitkonstante von $\tau = 1 \text{ ms}$ bei einer Induktivität **L =1 mH**.
3. Konfiguration der Simulation für eine Zeitbereich von **5 ms**.
4. Überprüfung der markanten Zeitpunkte auf richtige Stromänderung durch Ausmessen der Signale.
5. Erweiterung des Modells um einen Stromsensor.
6. Automatische Erfassung des Zeitpunktes für das Erreichen eines vorgegebenen Stromwertes.

Frage 2:

Zu welchem Zeitpunkt **t** wird für das Beispiel exakt ein Stromwert von **90%** des Maximalwertes beim Einschalten erreicht?

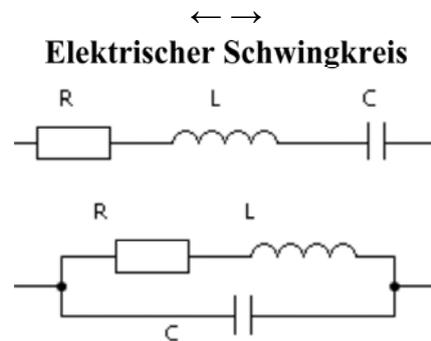
← →

Von „[http://optiyummy.de/index.php?title=Software: SimX - Einfuehrung - Elektro-Chaos - Luftspule](http://optiyummy.de/index.php?title=Software:_SimX_-_Einfuehrung_-_Elektro-Chaos_-_Luftspule)“

Software: SimX - Einfuehrung - Elektro-Chaos - Schwingkreis

Aus OptiYummy

↑



Ein elektrischer Schwingkreis ist eine resonanzfähige elektrische Schaltung aus einer Spule (**L**) und einem Kondensator (**C**), die elektrische Schwingungen ausführen kann. In realen Schwingkreisen muss man immer eine ohmsche Verlustleistung berücksichtigen, welche überwiegend aus dem ohmschen Widerstand (**R**) des Spulendrahtes resultiert.

Je nach Anordnung der Induktivitäten und Kapazitäten unterscheidet man zwischen Reihenschwingkreis und Parallelschwingkreis. Wir untersuchen zuerst die Resonanz eines Reihenschwingkreises, um danach mit den gesammelten Erfahrungen die analogen Experimente an einem Parallelschwingkreis durchzuführen.

Die Resonanzfrequenz eines idealen verlustfreien (**R=0**) Schwingkreises mit konstanten Werten für **L** und **C** berechnet sich zu

$$f_0 = \frac{1}{2\pi\sqrt{LC}}$$

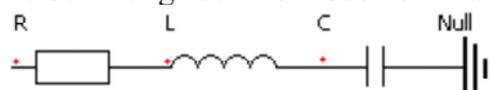
Da zumindest die Spule in realen Schwingkreisen immer einen ohmschen Widerstand **R** des Drahtes aufweist (Ausnahme: Supraleitfähigkeit), ist die reale Resonanzfrequenz **f_r** etwas geringer, als die nach obiger Thomsonschen Schwingungsgleichung berechnete Frequenz **f₀**:

$$f_r = \frac{1}{2\pi} \sqrt{\frac{1}{LC} - \frac{R_L^2}{L^2}}$$

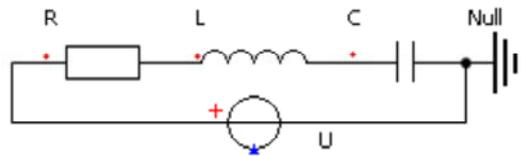
Reihenschwingkreis

Wir werden im SimulationX das Modell eines Serienschwingkreis aufbauen, welcher mit einer Frequenz **f₀=1 kHz** schwingt (Modell-Datei: **Reihenschwingkreis.ism**). Dazu steht uns eine Luftspule mit einer Induktivität **L=1mH** zur Verfügung. Die Spule besitzt einen Drahtwiderstand **R=1,3 Ω**. Den erforderlichen Kondensator berechnen wir mit der Thomsonschen Schwingungsgleichung.

Wir beachten, dass eine elektronische Schaltung auch als Modell ein Nullpotential benötigt:



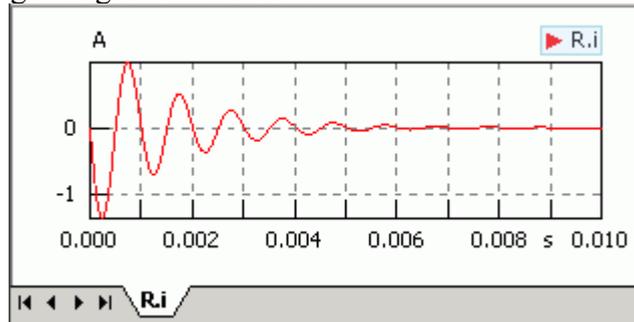
In obiger Schaltung kann es noch nicht zu Schwingungen kommen, weil eine Anregung fehlt und der Kreis noch nicht geschlossen ist. Um später beliebige Anregungen generieren zu können, benutzen wir eine Spannungsquelle **U** mit Signaleingang:



Beachte: Der Innenwiderstand dieser idealisierten Spannungsquellen ist gleich Null. Praktisch bedeutet dies, dass sich die Quelle bei $u=0$ wie ein durchgehender Leiter verhält.

Diese Schaltung kann man ohne äußere Anregung zum Schwingen bringen, wenn man einen aufgeladenen Kondensator einbaut:

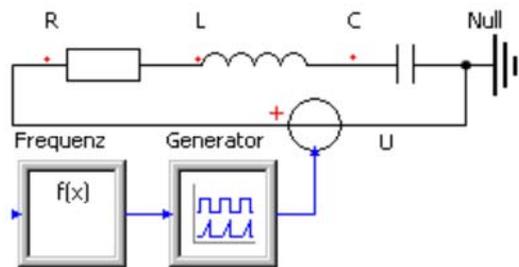
1. Wir setzen die Anfangsspannung $C.v_0=10\text{ V}$.
2. Da durch alle in Reihe geschalteten Bauelemente der gleiche Strom fließt, können wir das Stromsignal eines beliebigen Bauelements (außer des Null-Leiters) betrachten.
3. Wir simulieren einen Zeitbereich von 10 ms und konfigurieren die Rechen- und Protokollschrittweiten so, dass saubere Sinusschwingungen abgebildet werden:



Wenn eine komplette Sinusperiode $T=1\text{ ms}$ erfordert, so haben wir damit den zuvor berechneten Wert für die elektrische Kapazität bestätigt. Die Frequenz beträgt dann $f=1/T=1\text{ kHz}$.

Anderenfalls haben wir uns verrechnet und müssen den geforderten Kondensatorwert noch richtig ermitteln.

Frage 3: Was ist der erforderliche Wert für die Kapazität C?



Wir werden im Folgenden den Schwingkreis mit einer variablen Sinusfrequenz anregen, um die Erhöhung der Schwingungsamplitude im Resonanzfall zu beobachten:

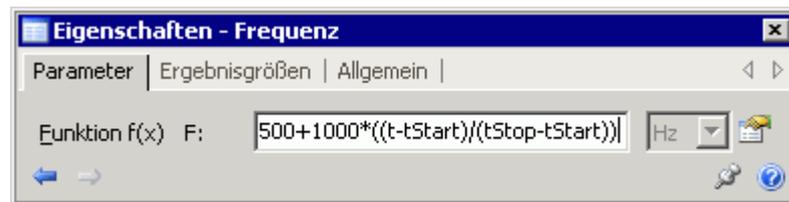
Wir nutzen nun wieder einen am Anfang entladenen Kondensator ($C.v_0=0\text{ V}$).

Wir ergänzen das Modell um einen Signalgenerator (Signalglieder→Quellen), dessen Frequenz wir über seinen Signaleingang **self.x** vorgeben:

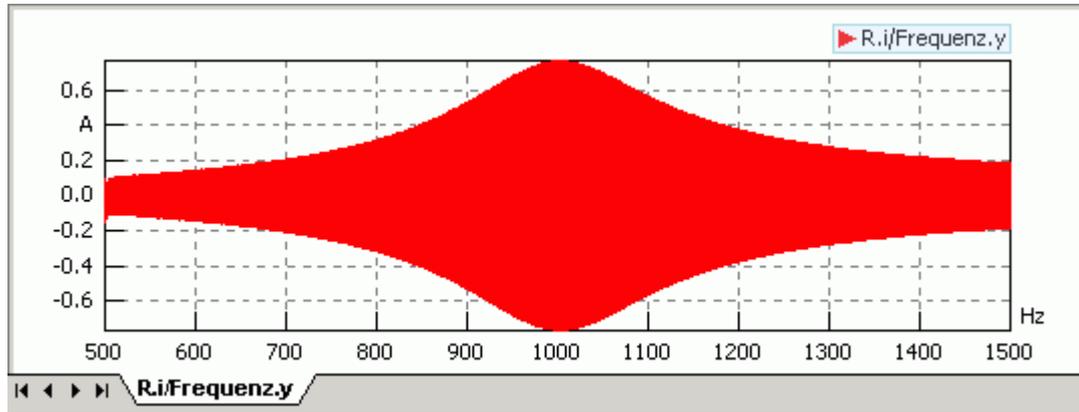


Mittels eines **f(x)**-Signalgliedes programmieren wir im simulierten Zeitbereich eine lineare Erhöhung der

Generator-Frequenz ausgehend von einer unteren Frequenz $f_u=500$ Hz um den Frequenzbereich $f_B=1000$ Hz:

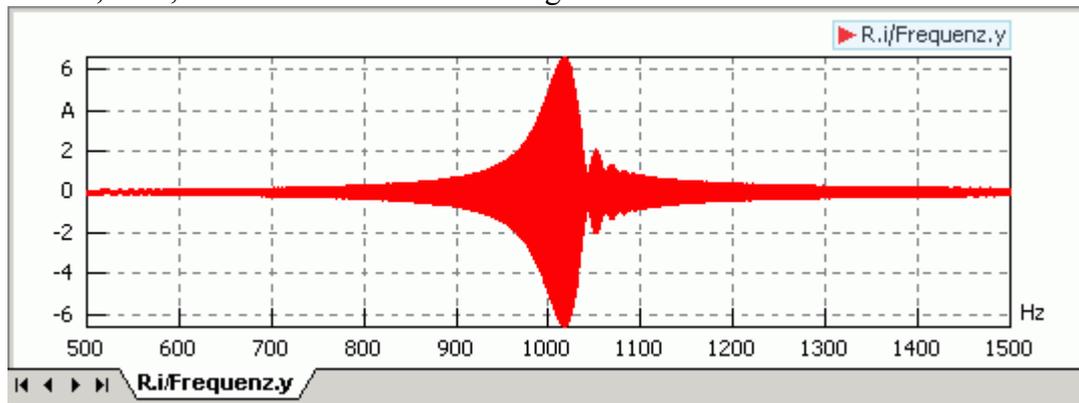


Es genügt eine Simulationszeit von $t_{\text{Stop}}=1$ s, um den gesamten Frequenzbereich hinreichend langsam zu scannen. Die $y(x)$ -Darstellung des Stromes i über die Anregungsfrequenz sollte das folgende Verhalten ergeben:



$y(x)$ -Darstellung: Beide Signale in ein gemeinsames Ergebnisfenster-Panel ziehen und von $y(t)$ - auf $y(x)$ -Darstellung umschalten. Die Zuordnung zu den Achsen kann nachträglich vertauscht werden.

Die Resonanzüberhöhung dieses Schwingkreises ist zwar deutlich, aber nicht sehr groß. Ursache ist die Dämpfung infolge des Spulenwiderstands von $R=1,3 \Omega$. Verwendet man eine Spule mit einem Zehntel dieses Wertes $R=0,13 \Omega$, so fällt Resonanzüberhöhung wesentlich markanter aus:



Hinweise: Die Einbrüche in der Hüllkurve nach Überschreiten der Resonanzfrequenz resultieren aus der relativ schnellen Änderung der Anregungsfrequenz. Vergrößert man die Simulationszeit auf z.B. 5 s, so verschwindet dieser Effekt. Auch wird die Lage des Resonanz-Maximum etwas genauer berechnet. Leider vergrößert sich dadurch wesentlich die Berechnungszeit. Da uns die Einbrüche in der Hüllkurve für die weitere Bearbeitung nicht stören, können wir den Wert $t_{\text{Stop}}=1$ s behalten.

Reihenschwingkreis - Signalverarbeitung

Frage 4: Welche Kreisgüte Q ergibt sich bei einem Drahtwiderstand von $R=1,3 \Omega$?

Der aus der Simulation durch Ausmessen der Grenzfrequenzen (Bandbreite B) zu ermittelnde Güte-Wert des Serienschwingkreises:

$$Q = \frac{f_0}{B}$$

ist mit dem analytisch bestimmten Wert zu vergleichen:

$$Q = \frac{1}{R} \sqrt{\frac{L}{C}}$$

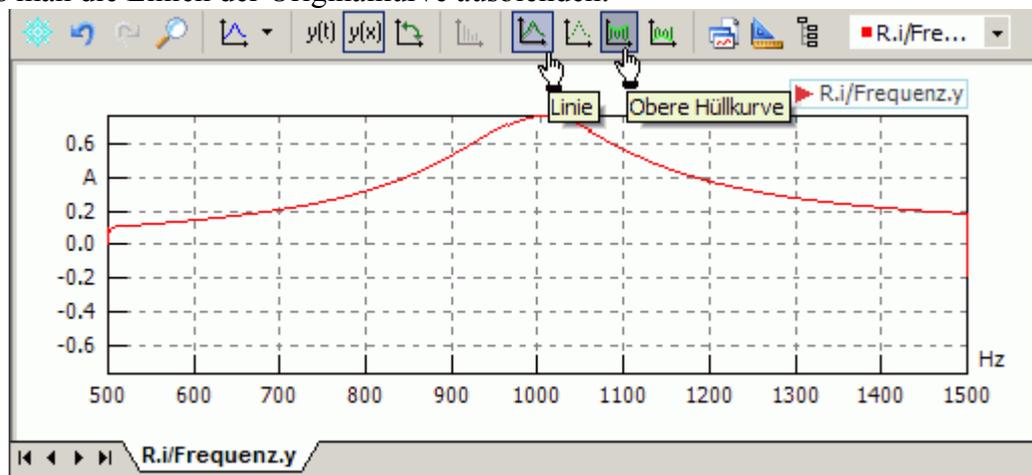
f_0 setzt sich aus dem geometrischen Mittel

$$f_0 = \sqrt{f_1 f_2}$$

der oberen f_2 und der unteren Grenzfrequenz f_1 zusammen. Die Grenzfrequenzen f_1 bzw. f_2 sind diejenigen Frequenzen, bei denen die Spannung U bzw. der Strom I auf den $1/\sqrt{2} \approx 0,707$ -fachen Wert des Maximalwertes zurückgehen. An dieser Stelle ist die Leistung im Schwingkreis nur noch halb so groß wie bei der Resonanzfrequenz.

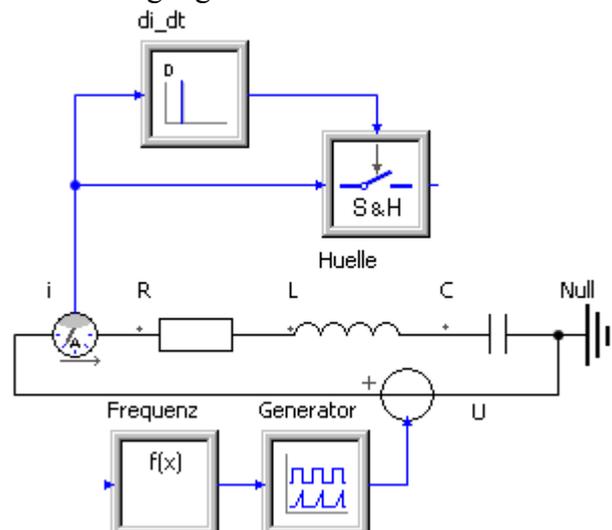
Das direkte Ausmessen der Amplituden der dargestellten sinusförmigen Signalverläufe mit dem Cursor ist kaum praktikabel. Sinnvoll wäre hierfür die Verwendung z.B. der oberen Hüllkurve:

In den Ergebnisfenstern kann z.B. die obere Hüllkurve eingeblendet werden. Damit man die Hüllkurve erkennt, muss man die Linien der Originalkurve ausblenden:



Leider funktioniert das Messen mit eingeblendeter Hüllkurve nicht besser als ohne. Der Messcursor folgt weiterhin dem unsichtbaren Originalsignal (oszilliert in vertikaler Richtung).

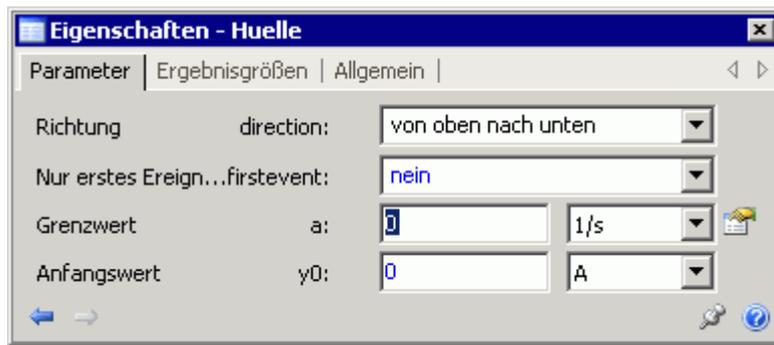
Wir ergänzen deshalb unseren Simulationsversuchsstand um eine Signalverarbeitung, welche die Hüllkurve der Strom-Maxima direkt als Modellgröße liefert. Die erforderlichen Bausteine für die Signalverarbeitung stehen in der Modellbibliothek zur Verfügung:



1. Ein analoger **Stromsensor** liefert uns die aktuellen Stromwerte.
2. Das spezielle Signalglied "**Ereignisgesteuertes Abtastglied**" (*Sample&Hold*) soll möglichst exakt die Maximalwerte der einzelnen Sinusschwingungen erfassen (*Sample*) und bis zum nächsten Maximalwert den Wert als Ausgangssignal bereitstellen (*Hold*).
3. Die Zeitpunkte der Sample-Triggerimpulse werden durch die Nulldurchgänge der zeitlichen Stromableitung

bestimmt (Siehe: Kurvendiskussion). Dazu verwenden wir das lineare Signalglied **Differenzierer (D-Glied)** mit Verstärkung **G=1** und Vorhaltezeit **Td=1 s**:

Entscheidend für die richtige Erfassung der Hüllkurve ist die korrekte Konfiguration des ereignisgesteuerten Abtastgliedes:

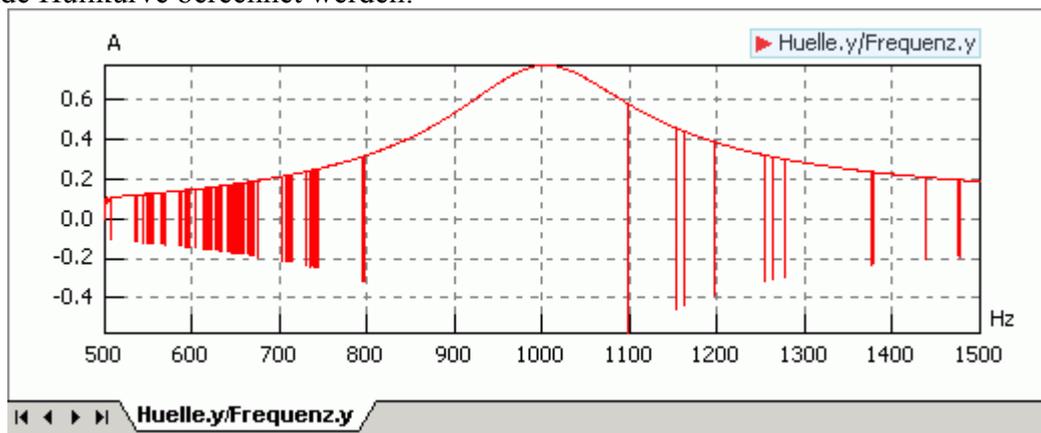


Das Erfassen des aktuellen Stromwertes soll bei $di/dt=0$ A/s erfolgen. Die "Auslösung" des Triggerimpulses wird über den **Grenzwert $a=0$ A/s** für das Steuersignal definiert. (**Hinweis:** In Ermangelung der Einheit A/s im SimulatinX wurde der **Gradient 1/s** als Ersatz verwendet!)

Wir wollen nur die obere Hüllkurve erfassen. Diese wird durch die Maxima des Stromsignals gebildet. Die 2. Ableitung des Stromes ist in den Maxima negativ, was durch **direction=von oben nach unten** definiert wird.

Da alle Maxima des Stromsignals zu erfassen sind, muss **firstevent=nein** gewählt werden.

Im Ergebnis der Simulation sollte mit **tStop=1 s**, **dtMin=1e-12 s** und "exakter" Darstellung der Ereignispunkte ungefähr folgende Hüllkurve berechnet werden:



Es werden einzelne Punkte erfasst, welche zur unteren Hüllkurve (den Minima) gehören.

Die Simulation im Zeitbereich erfolgt mit einer sich automatisch anpassenden Zeitschrittweite **dt**. Diese Schrittweite wird so geregelt, dass die Berechnung mit einer hinreichenden Genauigkeit erfolgt.

Zusätzlich versucht der Solver mit der sogenannten Ereignisbehandlung, markante Zeitpunkte möglichst exakt anzusteuern. Zu diesen "markanten" Zeitpunkten gehören die Auslösezeitpunkte für die Sample-Triggerimpulse.

Hierbei erfolgen augenscheinlich Fehlauflösungen, weil die Simulationssteuerung noch nicht richtig konfiguriert ist. Im Beispiel wurden folgende Werte für die allgemeinen Einstellungen verwendet:

Für numerisch unkritische Simulationen genügt meist die Einstellung des gewünschten Zeitbereiches und eine Anpassung der Schrittweitenparameter, um optisch "schöne" Signalverläufe zu erhalten.

Treten jedoch unvorhergesehene Schwierigkeiten auf, wie wir sie bei der Hüllkurvenfassung beobachten, ist ein gewisses Erfahrungswissen erforderlich. Das ist ähnlich wie beim Autofahren: je kritischer die Straßenverhältnisse sind, desto feinfühlinger muss man lenken, Gas geben und bremsen.

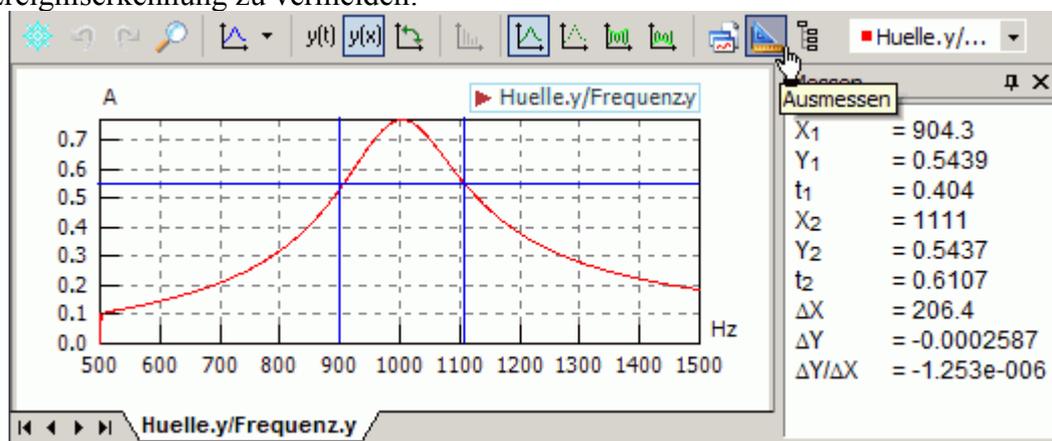
Für unser Beispiel wird hier eine Lösung vorgeschlagen, die hoffentlich bei allen funktioniert:

Im Normalfall stört ein sehr kleiner Wert für **dtMin** nicht, da eine automatische Schrittweitenregelung für **dt** erfolgt.

Das ändert sich jedoch, wenn eine Ereignisbehandlung stattfinden muss. Nachdem ein Ereigniszeitpunkt möglichst exakt angesteuert wurde, beginnt die Simulation nach dem Ereignis wieder mit der Schrittweite **dt=dtMin**.

Wie wir sehen, kann es durch numerische Ungenauigkeiten dazu kommen, dass der soeben detektierte Nulldurchgang von **di/dt** erneut detektiert wird. Infolge des Signalrauschens um den Nullwert kann der Nulldurchgang dabei auch mit umgekehrten Vorzeichen erfasst werden.

Im Beispiel genügte eine Verringerung der minimalen Schrittweite auf **dtMin=1e-7 s**, um die sporadisch fehlerhafte Ereigniserkennung zu vermeiden:

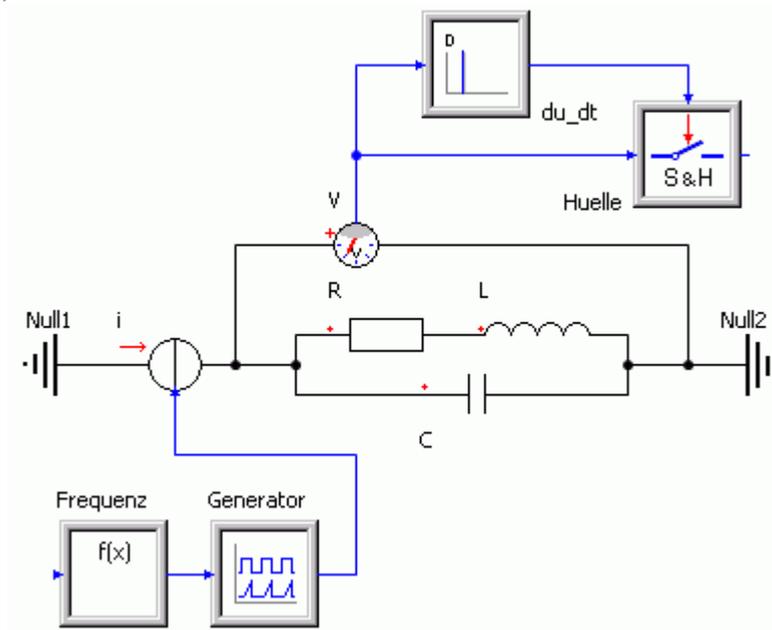


Folgt man den Anweisungen im Fenster für das Ausmessen, so kann man problemlos an der Hüllkurve die erforderliche Absolutwerte und Differenzen zur Berechnung der Schwingkreisgüte ermitteln. Damit kann **Frage 4** beantwortet werden.

Parallelschwingkreis

Mit den am Modell des Serienschwingkreises gewonnenen Erfahrungen werden wir nun ein neues Modell zur Bestimmung der Resonanzgüte eines Parallelschwingkreises aufbauen (Modell-Datei:

Parallelschwingkreis.ism):



Wir benutzen die gleichen Bauelemente, wie für den Serienschwingkreis. Den Drahtwiderstand verringern wir auf $R=1 \Omega$.

Die Spule bilden wir als Reihenschaltung von Induktivität L und Drahtwiderstand R nach.

Anstatt mit einer Spannungsquelle erregen wir den Parallelschwingkreis mit einer Stromquelle.

Im Resonanzfall ergibt sich ein Maximum der Spannungsamplitude über dem Kondensator. Da die Amplitude des Erregerstroms konstant bleibt, bedeutet dies, dass der Wechselstromwiderstand des Parallelschwingkreises im Resonanzfall am größten ist:

Frage 5:

Welche Kreisgüte Q ergibt sich bei einem Drahtwiderstand von $R=1 \Omega$? Der aus der Simulation ermittelte Wert ist mit dem analytisch berechneten Wert zu vergleichen. Die Änderung der Kreisgüte im Vergleich zum höheren Drahtwiderstand des Serienschwingkreises ist zu erläutern.

Hinweise zur Lösung:

Siehe <http://de.wikipedia.org/wiki/Schwingkreis>

Für die analytische Berechnung der Kreisgüte

$$Q = R_p \sqrt{\frac{C}{L}}$$

ist der "echte" Drahtwiderstand R_L zuvor in einen Ersatzwiderstand R_p umzurechnen:

$$R_p = \frac{L}{R_L C}$$

← →

Von „[http://optiyummy.de/index.php?title=Software: SimX - Einfuehrung - Elektro-Chaos - Schwingkreis](http://optiyummy.de/index.php?title=Software:_SimX_-_Einfuehrung_-_Elektro-Chaos_-_Schwingkreis)“

Software: SimX - Einfuehrung - Elektro-Chaos - Oszillator

Aus OptiYummy

↑

← →

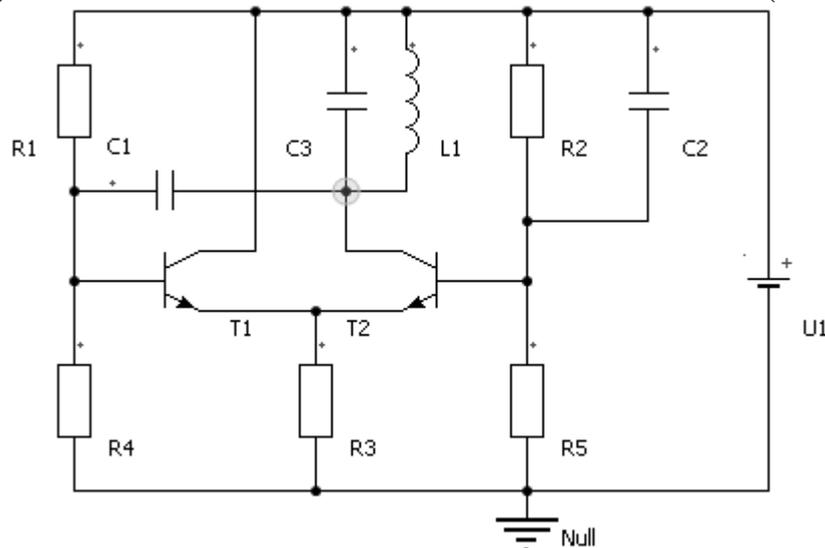
Sinus-Oszillator (Schaltungssimulation) (Bei Zeitverzug überspringen!)

Um einem LC-Schwingkreis eine dauerhafte Sinusschwingung zu entlocken, muss man ihm mittels einer elektronischen Schaltung zu jedem Zeitpunkt exakt die Energie zuführen, welche durch die Verluste im Schwingkreis in Wärme umgewandelt wird (Siehe: Oszillator).

Eine mögliche Schaltung verwendet einen Differenzverstärker mit zwei Transistoren und zeichnet sich durch sehr gutmütiges Verhalten aus. Für die Schaltungssimulation benutzt man im Normalfall Spezialprogramme, z.B. die weit verbreitete SPICE-Software.

In Simulatoren zur System-Simulation, zu denen SimulationX gehört, kann man einfache elektronische Schaltungen ebenfalls problemlos simulieren. Man hat dann zwar den Nachteil fehlender Bauelement-Datenbanken. Jedoch kann man die Schaltungsmodelle direkt mit anderen physikalischen Domänen koppeln (z.B. mit elektro-mechanischen Aktuatoren).

Wir bauen die Schaltung eines Sinus-Oszillators als neues SimulationX-Modell auf (**Oszillator.ism**):



Die Frequenz wird durch den Parallel-Schwingkreis aus **L1** und **C3** bestimmt:

- Wir verwenden eine Luftspule mit einer Induktivität **L1=0.01 H**.
- Es soll eine Frequenz von **f₀=1 kHz** erzeugt werden. Dafür ist die Kapazität **C3=? F** zu ermitteln (verlustarmer Klasse-1-Keramik-Kondensator):

$$f_0 = \frac{1}{2\pi\sqrt{LC}}$$

Für die restlichen Bauelemente verwenden wir folgende Parameter:

- Die Kapazitätswerte der anderen Kondensatoren sollen den gleichen Wert erhalten: **C1=C2=C3**.
- Alle Kondensatoren müssen am Anfang entladen sein: **v0=0** (mit gesetzter "Reißzwecke!").
- Die Widerstände besitzen folgende Werte:

R1=R2=330 k Ω

R4=R5=100 k Ω

R3=4.7 k Ω

Die Transistoren behalten ihre Vorgabewerte.

Die Betriebsspannung betragt **U1=12V**.

Die generierte Sinus-Spannung greifen wir am Kollektor von **T2** z.B. mittels eines zusatzlichen Spannungssensors ab (Potentialpunkt in obiger Schaltung markiert). Diese Ausgangsspannung kann im Bereich von **0..24 V** schwingen und soll in einem Signalfenster dargestellt werden..

Wir konfigurieren die Simulation fur einen Zeitbereich von **tStop=1 s** und mit einer hinreichend kleinen Protokollsrittweite **dtProtMin**:

Nach dem Start der Simulation werden wir mit groer Wahrscheinlichkeit sofort mit einer Fehlermeldung uberrascht:

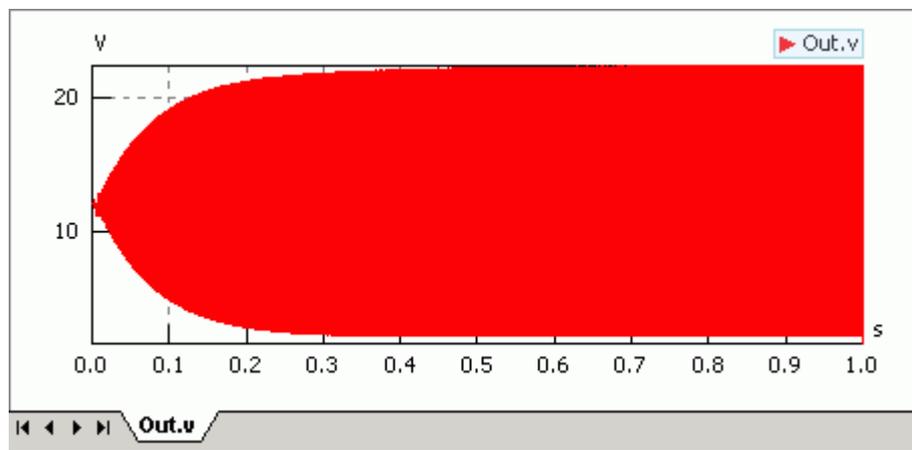


Solche Probleme haben wir fur eine "harmlose" Sinusschwingung nicht erwartet!

Die Sinusschwingung ergibt sich jedoch erst aus der Wechselwirkung insbesondere der beiden Transistoren, welche sehr kleine Zeitkonstanten besitzen. Damit wird insbesondere der Einschwingvorgang unmittelbar nach dem Einschalten sehr kritisch.

Dieses Problem kann man mit Erfahrungswissen durch Verringern der minimalen Zeitschrittweite losen, z.B. **dtMin=1E-12 s**:

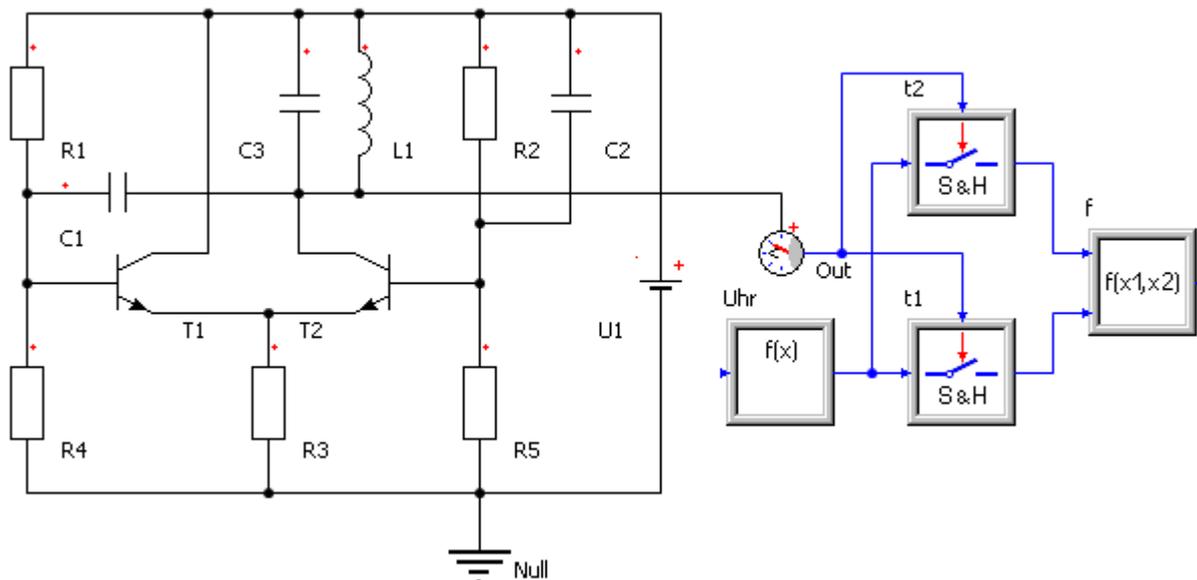
Danach sollte die Simulation ordentlich starten und plausible Ergebnisse berechnen:



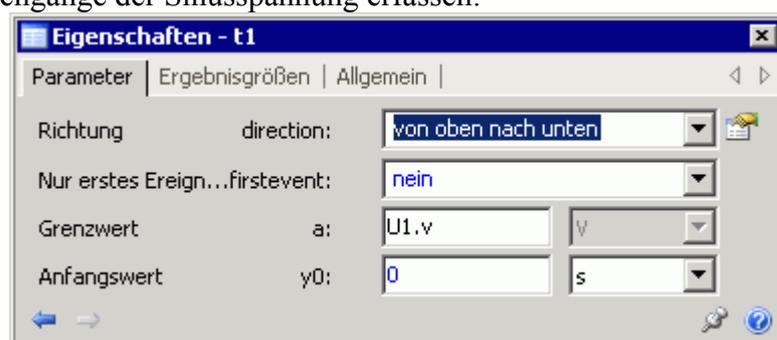
Signalverarbeitung

Spätestens jetzt sollte man überprüfen, ob die mit dem analytisch dimensionierten Schwingkreis erzeugte Frequenz den richtigen Wert besitzt:

- Hier könnte man im Signalfenster die Dauer einer Schwingungsperiode ausmessen. Das ist etwas umständlich und fehleranfällig.
- Mit unseren Erfahrungen zur Signalverarbeitung werden wir uns eine Auswerte-Elektronik modellieren, welche die aktuelle Frequenz automatisch ermittelt:



- Die **Uhr** zum Messen der aktuellen Simulationszeit t wird durch ein Signalglied **Function** nachgebildet. Als Parameter **F** des Function-Blockes verwenden wir die globale Variable t , welche den aktuellen Wert der Simulationszeit enthält.
- Das Ausgangssignal der Uhr speisen wir in **2 ereignisgesteuerte Abtastglieder**. Diese sollen die Zeitpunkte der Nulldurchgänge der Sinusspannung erfassen:



- Die "Null-Linie" der Sinusspannung entspricht dem Wert der Betriebsspannung. Deshalb wird **U1.V** als Grenzwert für die Auslösung des Sample-Vorgangs definiert.

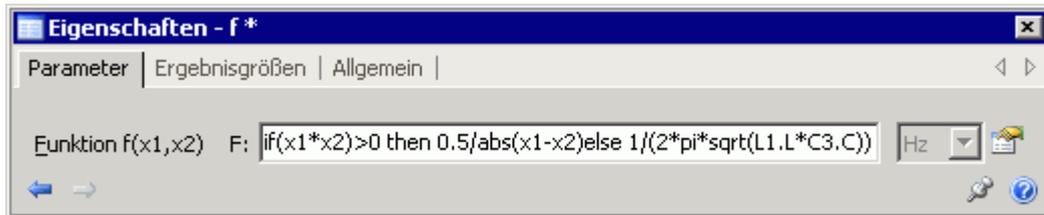
t_1 und t_2 erfassen jeweils die Zeitpunkte entgegengesetzter "Nulldurchgänge" der Sinusspannung. Somit sind die Ausgangs-Zeitwerte beider Abtastglieder immer um die Dauer der letzten Halbwelle versetzt.

Ein Signalblock **Function2** erfasst mit seinen beiden Eingängen x_1 und x_2 die beiden gesampelten Zeitwerte und berechnet daraus die aktuelle Frequenz **F: $0.5/abs(x_1-x_2)$**

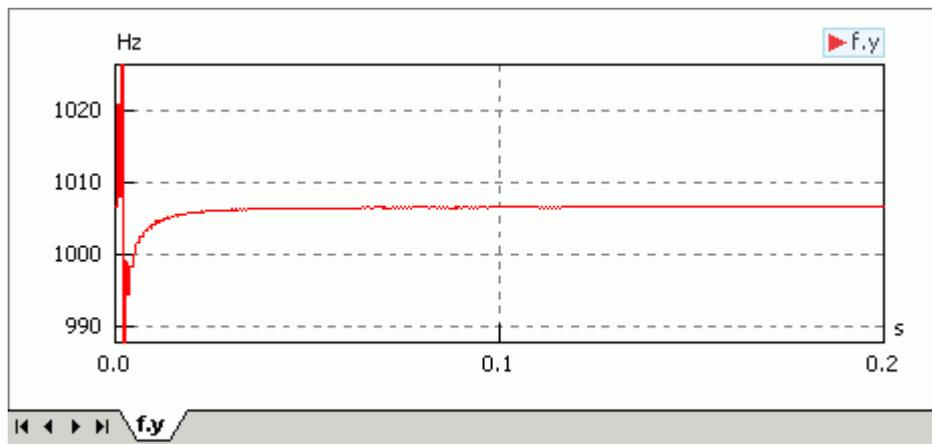
Leider stürzt die Simulation damit sofort nach dem Start mit "Division durch Null" ab, weil für t_1, y und t_2, y jeweils nur der Anfangswert $y_0=0$ vorliegt.

Deshalb ist bei der Frequenzberechnung dieser Fall separat zu behandeln. Es genügt dafür die Bedingung $x_1*x_2>0$.

Für die Zeit der ersten Periode wollen wir aus den Schwingkreis-Parametern analytisch den Wert der Resonanzfrequenz berechnen:



Der eingeschwungene Zustand wird im Beispiel nach ca. 0.2 s erreicht. Für diesen Zeitbereich betrachten wir nun den Frequenzverlauf des Oszillators:



Bereits nach wenigen Perioden ist die endgültige Frequenz erreicht. Die erzeugte Frequenz liegt im Beispiel wenige Hertz oberhalb von 1 kHz. Ursache ist die Rundung des Kapazitätswertes für C_3 auf 2 Ziffernstellen.

Frage 6: Welcher "exakte" Kapazitätswert (gerundet auf 4 Ziffernstellen) müsste bei dem vorgegebenem "exakten" Induktivitätswert $L_1=0.01$ H für eine Frequenz von 1 kHz verwendet werden?

Verlustbehafteter Parallel-Schwingkreis

Bisher funktioniert unsere Oszillator-Schaltung recht gut. Dabei kann man schnell vergessen, dass hier nicht die Schaltung selbst funktioniert, sondern ein Modell der Schaltung! Das Verhalten kann in der Realität ganz anders aussehen, weil wir einen wichtigen Effekt im Modell vergessen haben:

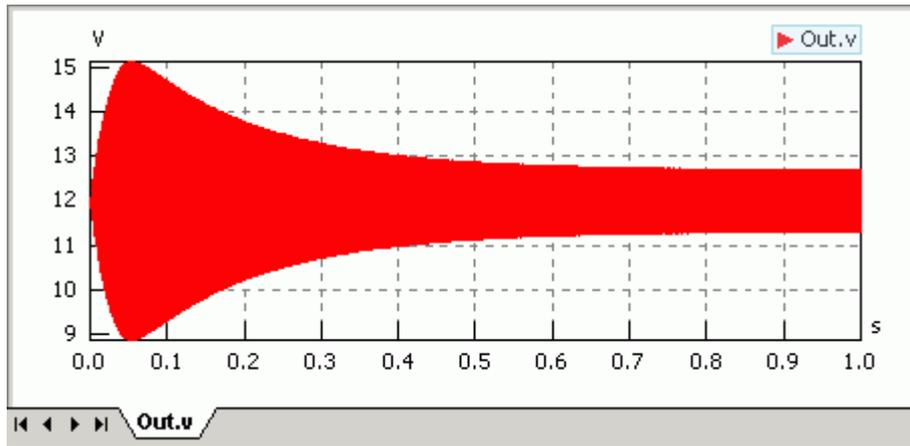
Eine Spule besitzt immer einen ohmschen Widerstand infolge des Spulendrahtes, wenn man nicht den Spezialfall der Supraleitung betrachtet. Das führt zu den bereits untersuchten gedämpften Schwingungen des passiven LC-Schwingkreises.

Auch der Kondensator arbeitet nicht verlustfrei. In unserem Beispiel können wir aber seine Verluste im Vergleich zu den Spulenverlusten vernachlässigen.

Wir ergänzen in unserem Modell in Reihe zur Induktivität einen Widerstand R_{Spule} , für den wir wieder einen Wert von 1.3Ω annehmen.

Hinweis: In der *Student Edition* von *SimulationX* kann man nur 5 Widerstandselemente verwenden. Wir benutzen deshalb ein Leitwert-Element mit $G=1/R$ als Serienwiderstand zur Induktivität.

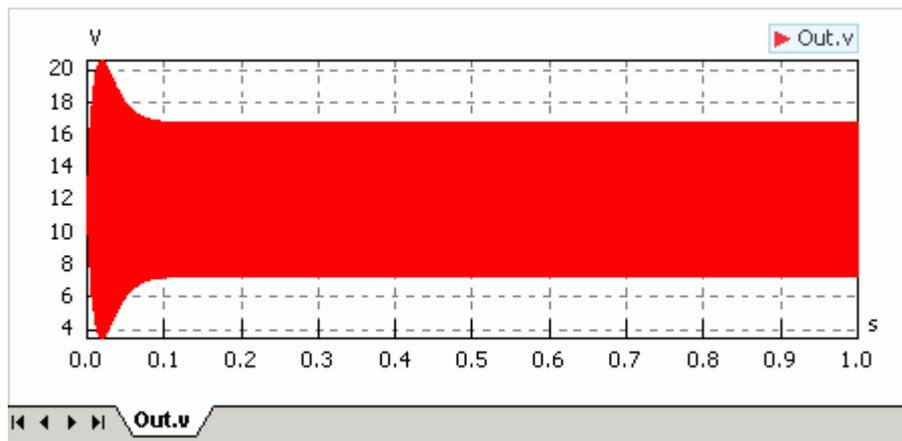
Nach dem Ergänzen des Spulenwiderstands ändert sich das simulierte Verhalten beträchtlich:



- Die Schwingungsamplitude beträgt ca. nur noch 1/10 im Vergleich zum idealen, verlustfreien Schwingkreis.
- Die Frequenz bleibt jedoch praktisch gleich.

Diese starke Abhängigkeit vom konkreten Spulenwiderstand zeigt, dass die Schaltung noch nicht optimal dimensioniert wurde. Mit Hilfe von Simulationsexperimenten werden wir die Schaltung so dimensionieren, dass wir bis zu einem Spulenwiderstand von 2Ω eine Signal-Amplitude von mindestens 10 V (bezogen auf den Mittenwert von 12 V) erzeugen können:

- Sicher kann man eine exakte analytische Dimensionierung dieser Schaltung in der Literatur finden. Bei genügend Expertenwissen, kann man die erforderlichen Gleichungen auch selbst herleiten.
- Wir wählen hier jedoch den pragmatischen Weg der "wohlüberlegten" Modifikation.
- Die Widerstände der Schaltung sind ziemlich hochohmig. Dadurch ist die Schaltung nicht in der Lage, die Verluste in der Spule im erforderlichen Maße zu kompensieren.
- Das kann man sehr schnell überprüfen, indem man z.B. alle Widerstände (außer R_{Spule}) auf ein Zehntel ihres bisherigen Wertes reduziert. Der Erfolg für den bisherigen Wert $R_{\text{Spule}} = 1.3 \Omega$ ist offensichtlich:



Es ist umständlich, durch iterative Variation aller Widerstände eine optimale Widerstandskonfiguration zu finden. Deshalb werden wir diese Experimente etwas rationalisieren, indem wir mittels eines Funktionsblockes $f(\mathbf{x})$ im Modell einen globalen Faktor kR definieren. Mit diesem Faktor verändern wir dann alle Widerstände der Schaltung gleichzeitig:



Die Funktion **F** wurde so gewählt, dass im Zeitbereich bis **tStop=1 s** sich der Reduktionsfaktor von 1 auf 100 erhöht.

Wir setzen alle Widerstandswerte auf ihren ursprünglichen Wert zurück. Als Maßeinheit wählen wir **Ohm**, weil nach dem späteren Ergänzen einer Gleichung mit dem Faktor **kR** der Wert physikalischer Größen in der SI-Einheit berechnet wird:

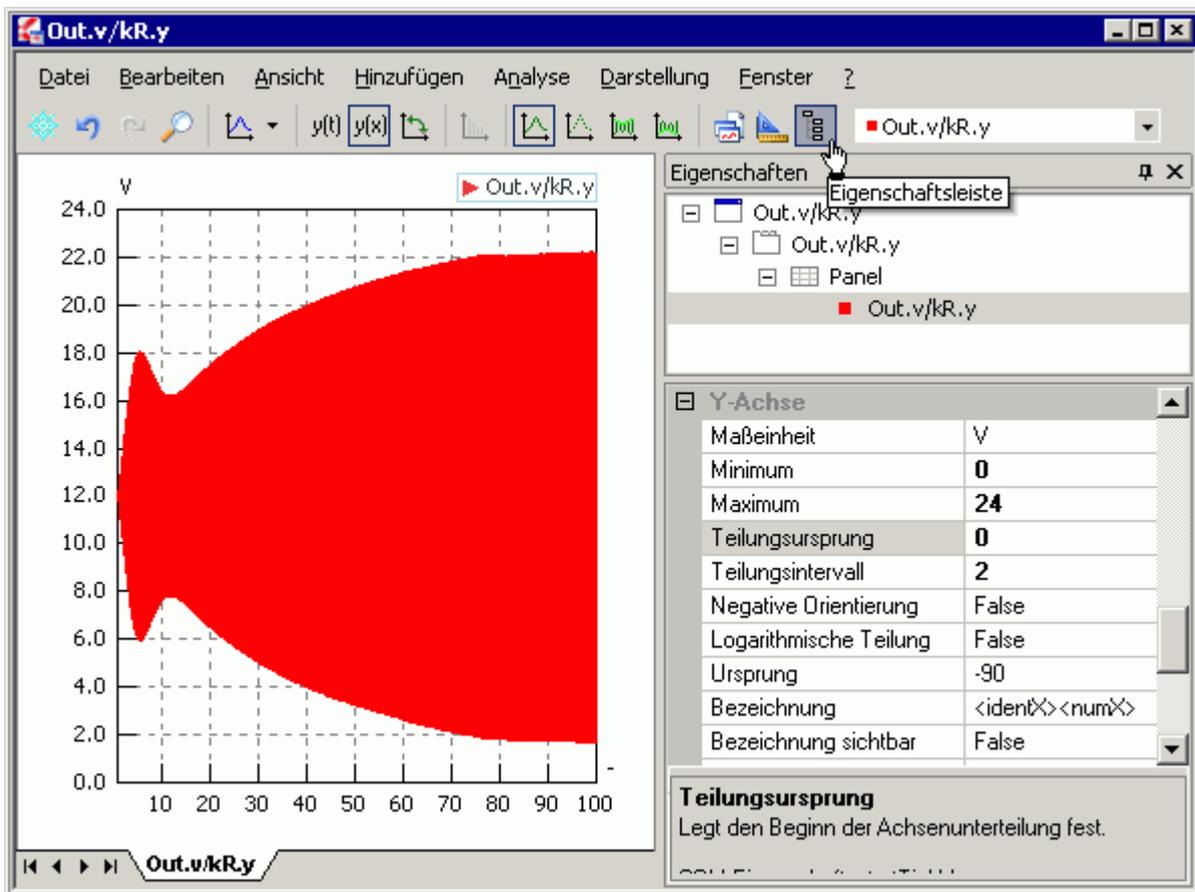


Danach ergänzen wir für die in Ohm angegebenen Widerstandswerte den Reduktionsfaktor:

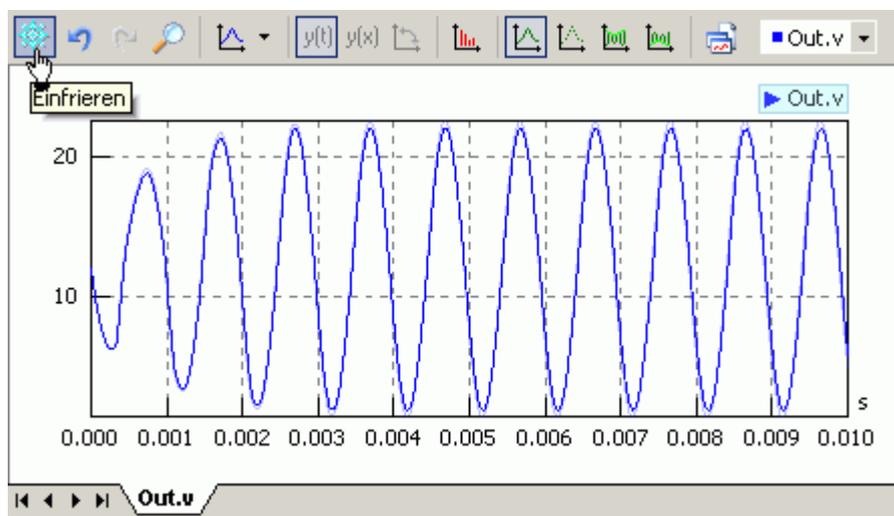


Während der Simulation werden die Widerstandswerte kontinuierlich verringert und damit erhöht sich stetig die Amplitude der Signalspannung bis zu ihrem Maximalwert.

Bei sinnvoller Beschriftung der Y-Achse des Signalfensters braucht man sich jetzt nur noch für den kleinst möglichen Reduktionsfaktor **kR** entscheiden (Konfiguration über Eigenschaftsleiste - für Y-Achse Autoskalierung deaktivieren und manuell konfigurieren):



Im Beispiel konnte mit $R_{\text{Spule}} = 2 \Omega$ ein fast identisches Schwingungsverhalten erzielt werden, wie mit einer idealen verlustfreien Spule (blasser, eingefrorener Kurve mit etwas höherer Amplitude). Der eingeschwungene Zustand wird mit dieser optimierten Dimensionierung nun wesentlich schneller erreicht:



Frage 7: Welche effektive thermische Verlustleistung wird in der optimierten Schaltung am Widerstand R_3 erzeugt?

Hinweise: Die Verlustleistung $P = u \cdot i$ wird in jedem Widerstandselement als Ergebnisgröße PI berechnet. Zeigt man diese berechnete Leistung in einem Signalfenster an, kann man als Hilfslinie den "Effektivwert" einblenden (**Hinzufügen** > **Horizontale Linie** > **Effektivwert**).

← →

Von „[http://optiyummy.de/index.php?title=Software: SimX - Einfuehrung - Elektro-Chaos - Oszillator](http://optiyummy.de/index.php?title=Software:_SimX_-_Einfuehrung_-_Elektro-Chaos_-_Oszillator)“

Software: SimX - Einfuehrung - Elektro-Chaos - Eigene Elemente

Aus OptiYummy

↑

← →

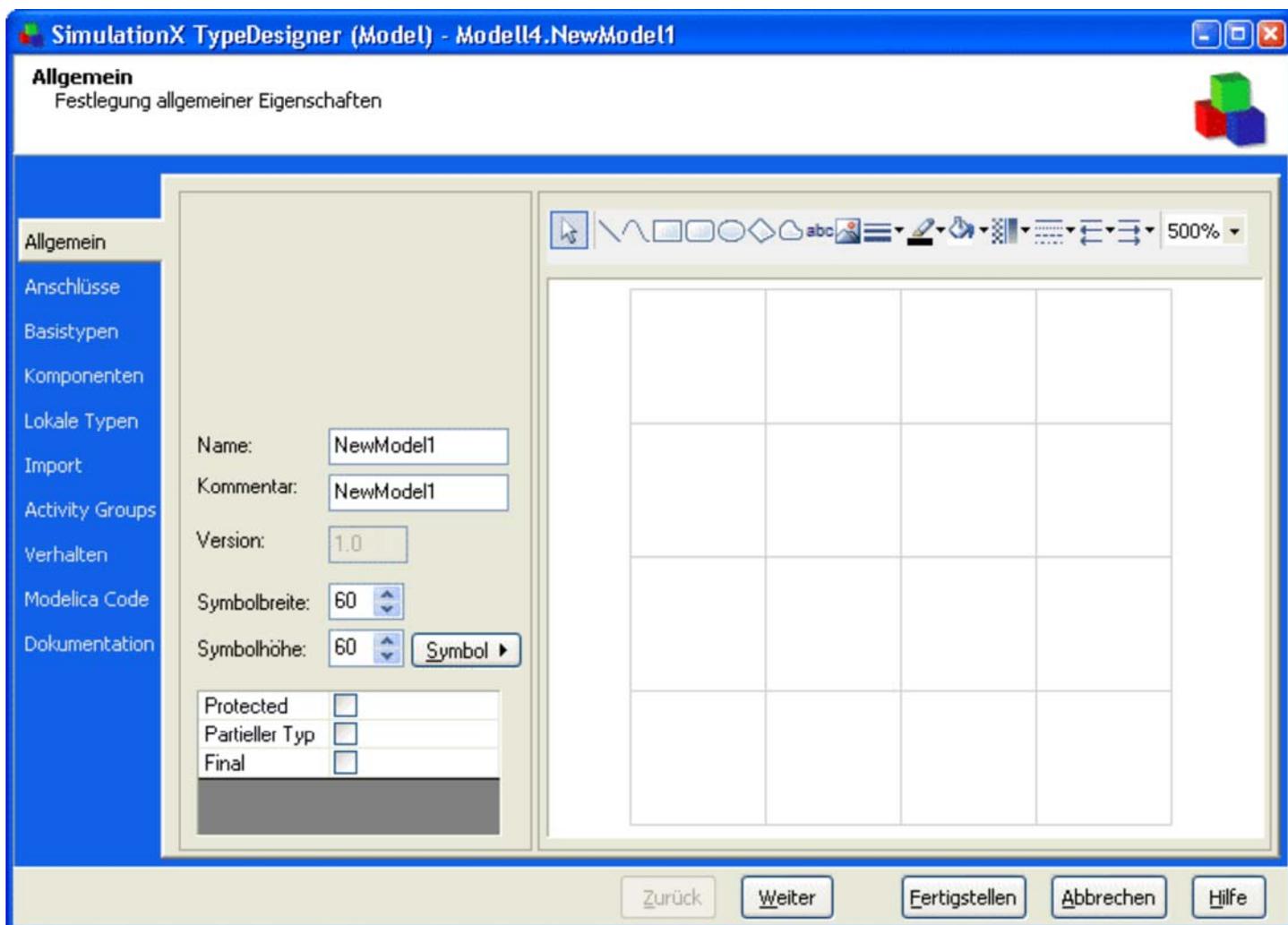
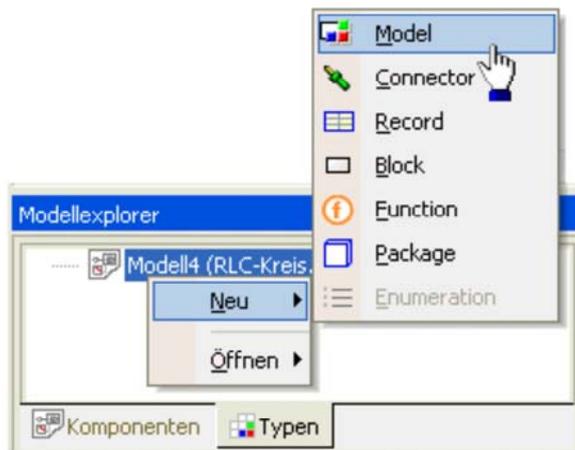
Modellierung eigener Elemente

Bisher haben wir fertige Modell-Elemente aus der SimulationX-Bibliothek verwendet. Am Beispiel der elektrischen Elemente Widerstand, Induktivität und Kapazität werden wir diese Bibliothekselemente zu Übungszwecken nachbauen und am Beispiel des Serienschwingkreises in Betrieb nehmen:

- Dazu erstellen wir zuerst eine Kopie vom Modell des Reihenschwingkreises unter dem Namen **RLC-Kreis.ism**
- Mit diesem funktionierenden Modell können wir dann testen, ob die selbst erstellten Elemente genauso gut funktionieren, wie die Bibliothekselemente.

Neue Elementtypen erstellt man mit dem sogenannten **TypeDesigner** von SimulationX:

- Man wählt im Modellexplorer die Seite **Typen**.
- Dort wählt man mit dem Kontextmenü des Wurzeleintrags (rechte Maustaste) **Neu > Model**.
- Damit bearbeitet man einen lokalen Elementtyp, der als Bestandteil der Modelldatei gespeichert wird:



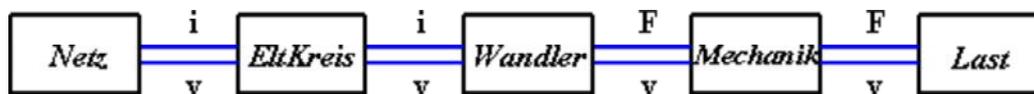
Inhaltsverzeichnis

Datenfluss in Modell-Elementen

Bevor wir ein konkretes Modell-Element "programmieren", werfen wir einen Blick auf die Einbindung dieser n-Pole in die Struktur des Gesamtmodells und den daraus resultierenden Datenfluss durch die Elemente:

- In einem physikalischen System kann man die Energieübertragung als kontinuierlichen Vorgang im Zeitbereich betrachten, solange man sich nicht auf das Niveau der Quantenphysik beigt.
- Ein System besteht aus Elementen, welche sich meist wechselseitig beeinflussen. Diese wechselseitige Beeinflussung erfolgt durch die Energieübertragung zwischen den Elementen.
- Bestimmend für die übertragene Energie \mathbf{W} ist der jeweilige Momentanwert der Leistung $\mathbf{P}=\mathbf{dW}/\mathbf{dt}$.

Betrachtet man ein technisches System (z.B. einen elektro-magnetischen Antrieb) mit seinen unterschiedlichen physikalischen Domänen (z.B. Elektrik, Magnetik, Mechanik), so kommt man sehr anschaulich zu einer verallgemeinerten Netzwerk-Theorie:



- Die Energieübertragung zwischen den Elementen wird über energetische Verbindungen (*Connection*) realisiert. Dazu besitzen die Elemente energetische Anschlüsse (*Connectoren*).
- Jede physikalische Domäne ist durch eine spezielle Art der Energie gekennzeichnet (z.B. elektrische Energie, Energie des magnetischen Feldes, mechanische Energie). Für jede physikalische Domäne gibt es deshalb spezielle energetische Anschlüsse und Verbindungen (z.B. elektrische, magnetische, mechanische).
- Betrachtet man die elektrische Domäne eines Systems, so kann an dem als Verbindung dienenden "Draht" ein Spannungspotential v in Bezug auf den Massepunkt der Schaltung messen. Durch den Draht fließt ein Strom i . Die aktuell durch den Draht übertragene elektrische Leistung ist $\mathbf{P_e}=\mathbf{v}\cdot\mathbf{i}$.
- In der mechanischen Domäne eines Systems berechnet man die mechanische Leistung $\mathbf{P_m}=\mathbf{v}\cdot\mathbf{F}$ als Produkt aus Geschwindigkeit und Kraft der mechanischen starren Verbindung. In Analogie zur Spannung in der Elektrik ist in der Mechanik die Geschwindigkeit die Potentialgröße gemessen im Bezugskoordinatensystem. Die Kraft wird als Flussgröße betrachtet.

Verallgemeinerung für beliebige energetische Netzwerke:

- energetische Verbindungen zwischen den Elementen besitzen Potential- und Flussgrößen
- die aktuell auf einer Verbindung übertragene Leistung ist das Produkt aus Flussgröße und Potentialgröße

Die Wechselwirkung zwischen den Element-Schnittstellen findet in der Natur augenblicklich statt ("Parallelverarbeitung"). Der Mensch definiert dafür jedoch aus Gründen der Anschaulichkeit gern kausale Abfolgen, z.B.

Kraft als Ursache der Bewegungsänderung einer trägen Masse

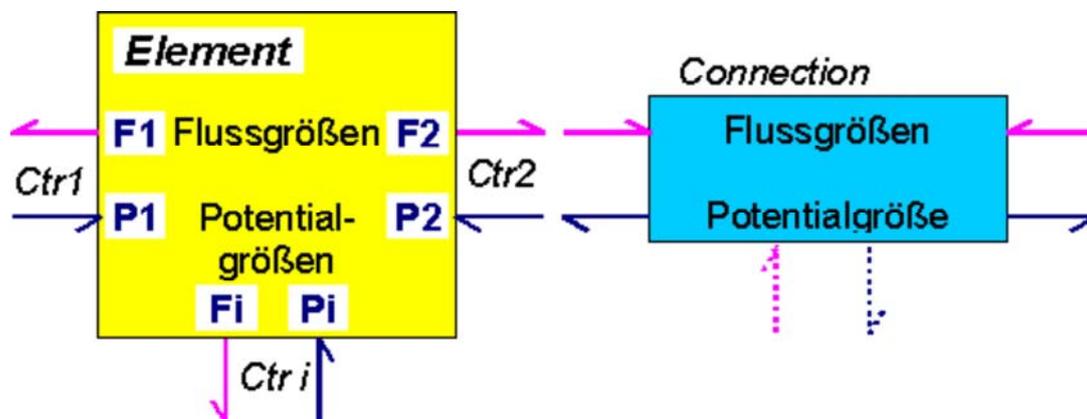
- Die umgekehrte Kausalfolge

Bewegungsänderung einer trägen Masse als Ursache einer Kraftwirkung

ist zwar gewöhnungsbedürftig, aber genauso brauchbar. Dies widerspiegelt sich vereinfacht in der Gleichung $\mathbf{F}=\mathbf{m}\cdot\mathbf{a}$, die beliebig umgestellt werden kann, ohne an Gültigkeit zu verlieren.

Moderne Simulationssysteme bilden diese Gleichzeitigkeit durch die Generierung und Lösung eines ganzheitlichen Gleichungssystems ab. Es ist jedoch in Hinblick auf die Anschaulichkeit sinnvoll, bei der Modellbildung kausale Wirkungsrichtungen als Datenflüsse abzubilden:

- Die Richtung des Datenflusses durch die Modell-Elemente wird hierbei als kausale Wirkungsrichtung interpretiert.
- Die Werte der Potentialgrößen P_i werden in die Element-Anschlüsse eingespeist. Aus der Sicht eines Elements (n-Pol) stehen also sämtliche Potentialwerte seiner Anschlüsse als Werte zur Verfügung.
- Aus den Potentialgrößen (=Ursache) P_i werden in jedem Element die Werte der Flussgrößen (=Wirkung) F_i berechnet und über die Anschlüsse nach außen gegeben.



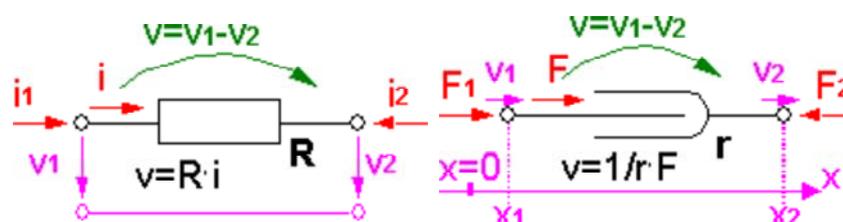
- Jede Connection besitzt eine Potentialgröße, deren Wert sie in die mit ihr verbundenen Element-Anschlüsse einspeist.
- Nach dem Knotensatz gilt für jede Connection $\sum F_n = 0$, wobei die Werte der F_n von den verbundenen Elementen in die Connection eingespeist werden.
- Für jeden berechneten Zeitschritt verändert der Solver iterativ die Werte aller Potentialgrößen solange, bis hinreichend genau für alle Verbindungen der Knotensatz $\sum F_n = 0$ erfüllt ist und eventuell definierte weitere Zwangsbedingungen eingehalten werden.
- Die Potentialgrößen stellen praktisch die Unbekannten des zu lösenden Gleichungssystems dar.

Zusammenfassung für eine günstige Modellierungsmethode:

- Bei der Programmierung eines Modell-Elements betrachtet man die Potential-Größen der Anschlüsse zusätzlich zu den Element-Parametern als vorgegebene Werte.
- Aus den Parametern und Anschluss-Potentialen versucht man mittels eines sequentiellen Algorithmus alle Anschluss-Flussgrößen zu berechnen.
- Existiert aus der lokalen Sicht eines Elements kein solcher Algorithmus, beschreibt man die Zusammenhänge zwischen Potentialgrößen und Flussgrößen mittels zusätzlicher Gleichungen (=Zwangsbedingungen). Diese zusätzlichen Zwangsbedingungen werden vom Solver bei der Ermittlung der aktuellen Potentialwerte berücksichtigt.

Da auf Basis der Modell-Elemente ein Baustein-System für Modelle angestrebt wird, müssen bei der Festlegung der positiven Richtungen an den Schnittstellen bestimmte Konventionen eingehalten werden:

- Im Sinne einer einfachen Übertragung von Modellen zwischen unterschiedlichen Simulationssystemen werden dabei die Vereinbarungen der Netzwerk-Theorie genutzt. Beispielhaft werden hier die positiven physikalischen Richtungen von Fluss- und Potentialgrößen an den Schnittstellen der "Verlust-Elemente" ohmscher Widerstand und translatorische mechanische Dämpfung gegenübergestellt:



physikalische Größe: positive Richtung / Bezugspunkt

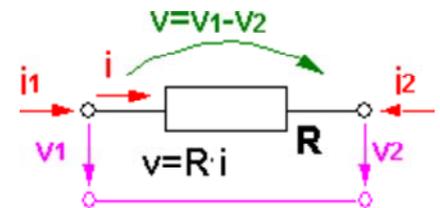
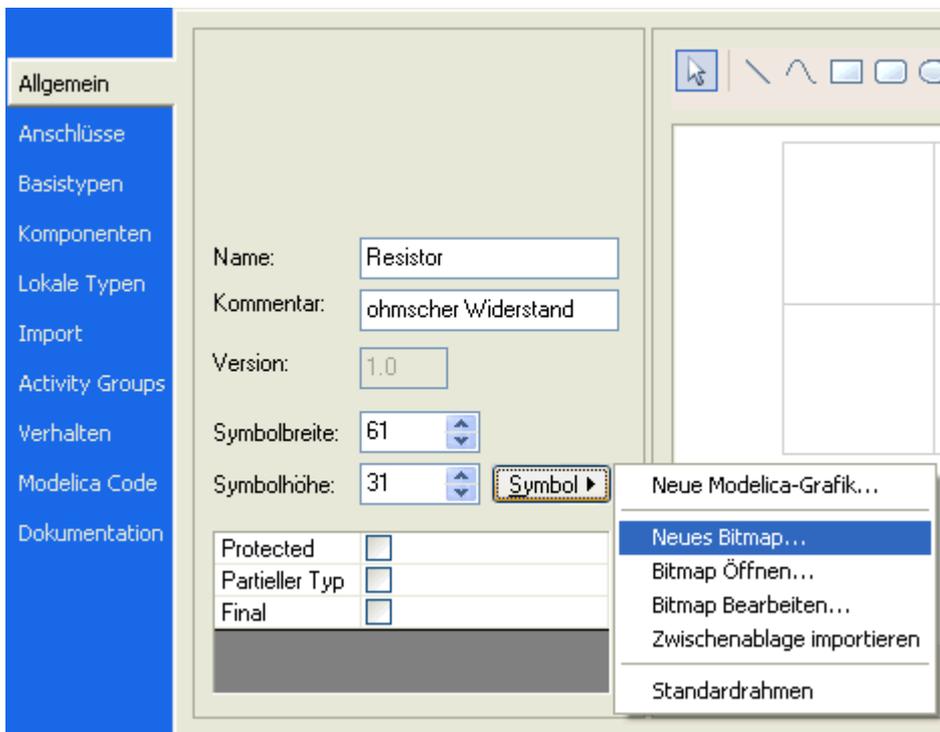
Kraft: in das Element hinein!
 Geschwindigkeit: "zeigt" in positive Wegrichtung;
 v-Tensor pos.=Elementverkürzung;
 Wegkoordinate: Wert auf zugehöriger Achse X,Y,Z des globalen Raumkoordinatensystems
 (bezogen auf Koordinaten-Ursprung)
 Elektrischer Strom: in das Element hinein!
 Elektrische Spannung: vom Potentialpunkt gegen Nullpotential

Die physikalisch positive Richtung (Pfeilrichtungen an obigem Widerstand und Dämpfer) muss man unterscheiden von der kausalen Datenflussrichtung. Zur Unterscheidung wird die Datenflussrichtung deshalb durch Halbpfeile abgebildet:

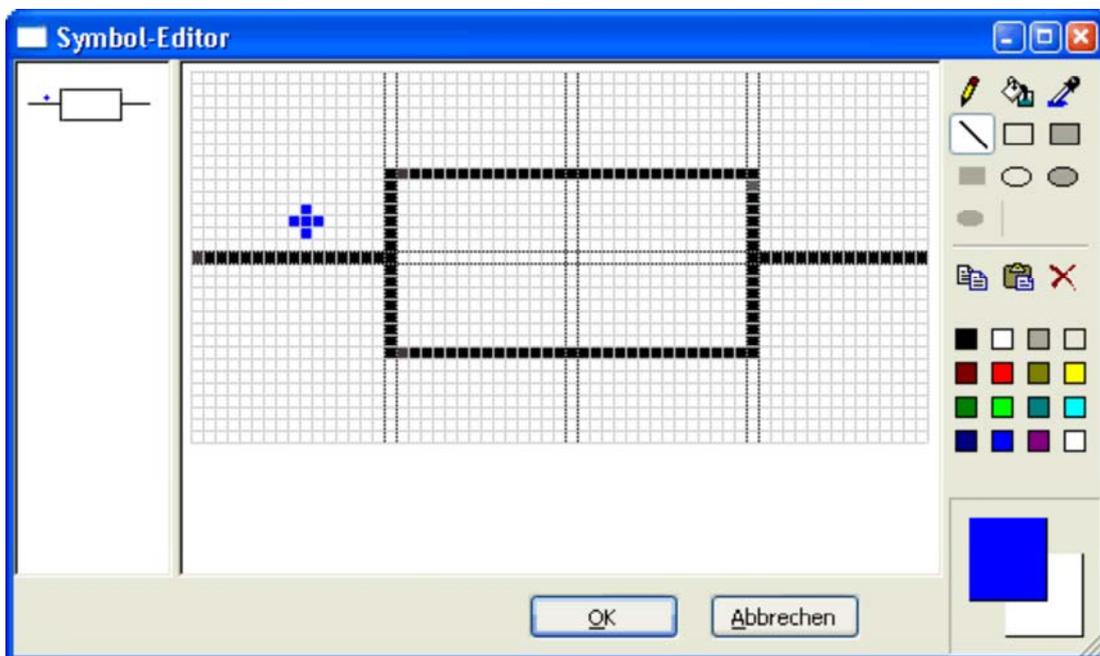
- Die Halbpfeile der Potentialgrößen zeigen grundsätzlich in die Elemente (unabhängig von der physikalischen Wirkungsrichtung).
- Die Halbpfeile der Flussgrößen zeigen grundsätzlich von den Elementen weg (unabhängig von der physikalischen Wirkungsrichtung).

Ohmscher Widerstand

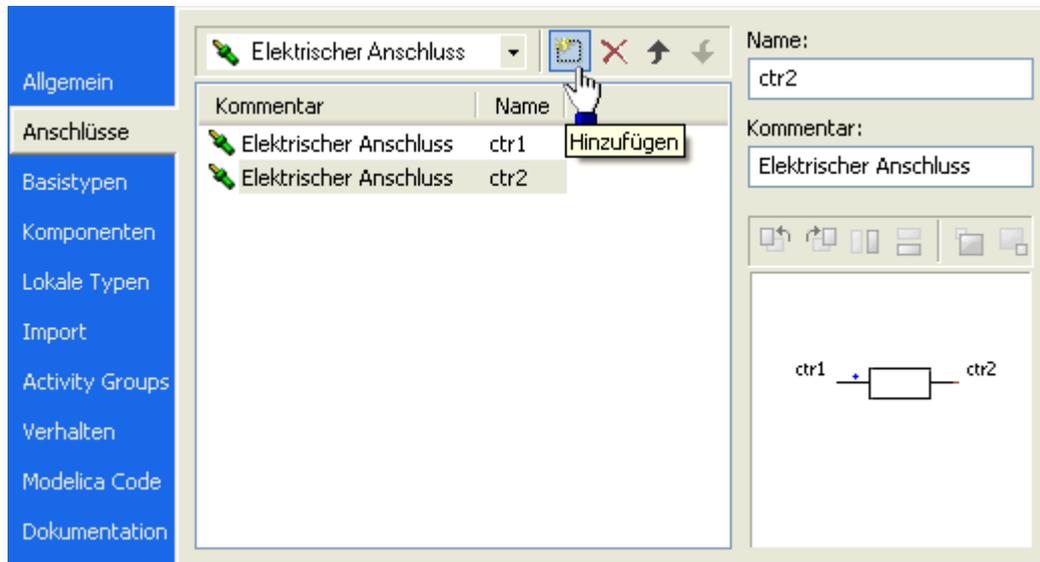
Wir beginnen mit dem Elementtyp **Ohmscher Widerstand**. Der Abschnitt **Allgemein** des *SimulationX-TypeDesigners* für die Erstellung eines lokalen Elementtyps ist geöffnet, wie bereits beschrieben wurde:



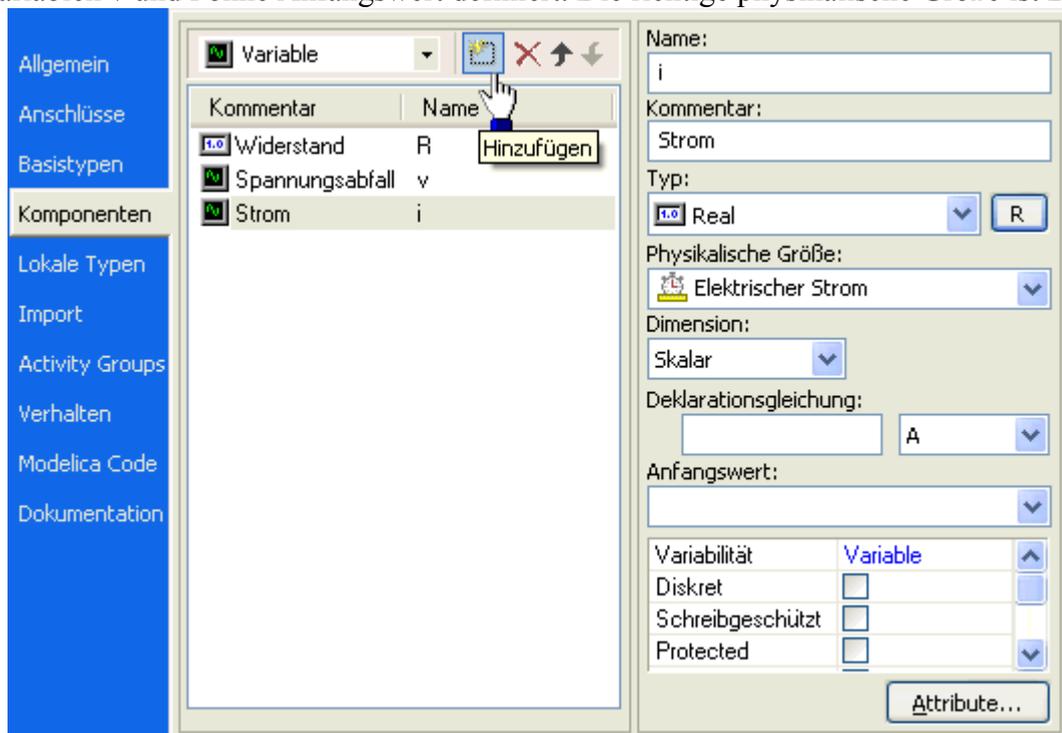
- Wir vergeben einen sinnvollen Namen und einen aussagekräftigen Kommentar für den neuen Elementtyp.
- Das Symbol erstellen wir als **Neues Bitmap** der Größe **61x31 Pixel** mit dem Symbol-Editor des *TypeDesigners*.
- Die linke Seite (Seite 1) ist bei den Bibliothekselementen mit einem rotem Plus gekennzeichnet. Wir verwenden zur Unterscheidung ein kleines blaues Plus:



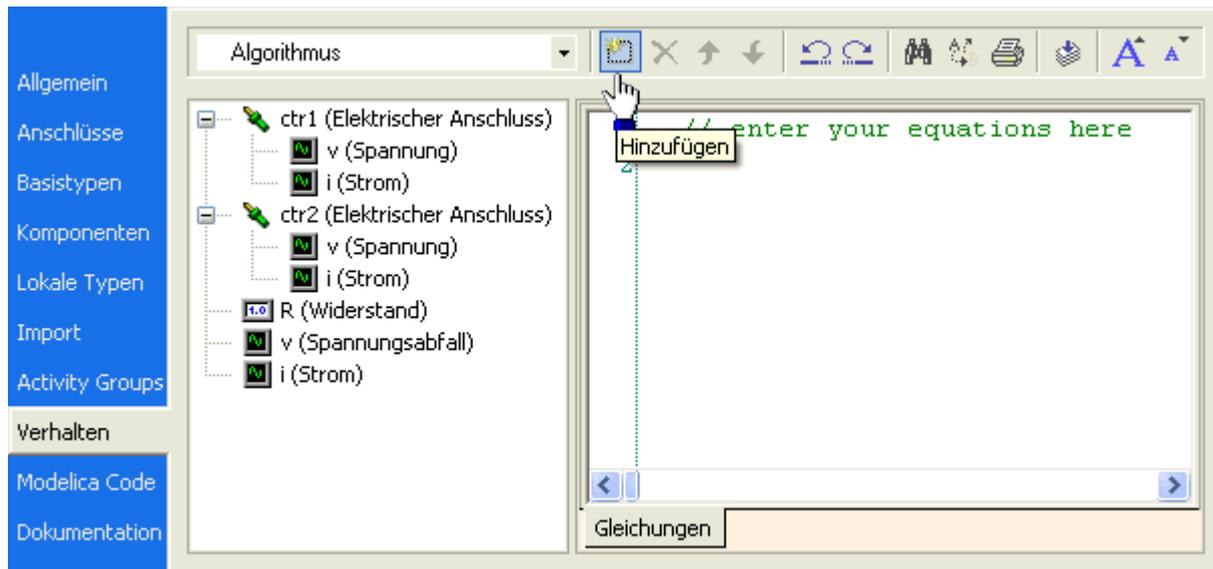
Wir definieren im nächsten Abschnitt zwei elektrische Anschlüsse. Die Standardbezeichner behalten wir bei:



Als Komponenten des Elements werden im Beispiel der Parameter **R** mit der **Standardbelegung 1 Ohm**, sowie die Variablen **v** und **i** ohne Anfangswert definiert. Die richtige physikalische Größe ist zuzuweisen:



Im Abschnitt "Verhalten" werden die im Modell-Element berücksichtigten physikalischen Effekte definiert. Wir berücksichtigen nur das ohmsche Gesetz:



Standardmäßig enthält der Verhaltensabschnitt nur eine Registerkarte für "Gleichungen". Mit diesen Gleichungen kann man die physikalischen Zusammenhänge in Form von einzuhaltenden Zwangsbedingungen beschreiben. Die Intelligenz der symbolischen Analyse des Simulationssystems kümmert sich dann um die Generierung der erforderlichen Modellalgorithmen.

Für den Einsteiger wird es als günstig erachtet, den Algorithmus für Modell-Elemente selbst zu schreiben. Das fördert das Verständnis für die numerische Simulation und gewährleistet auch, dass man alle erforderlichen mathematischen Zusammenhänge im Verhaltensabschnitt definiert.

Wir fügen deshalb dem Verhaltensabschnitt eine Registerkarte "Algorithmus" hinzu.

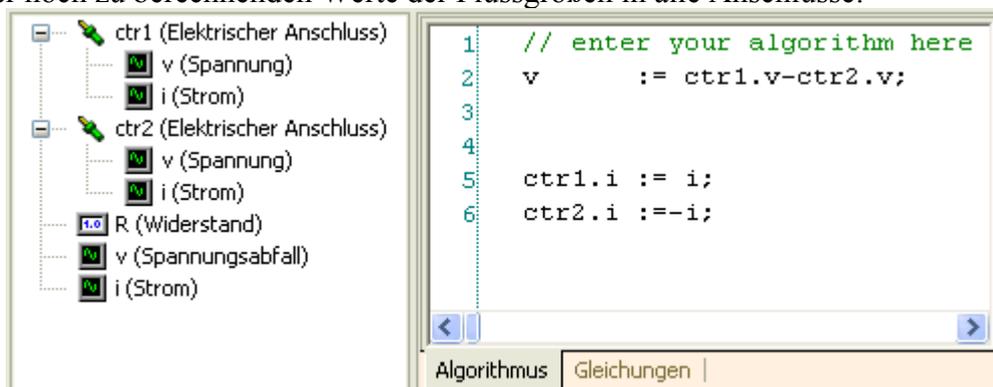
Hinweis:

Für das Erstellen des Verhaltensquelltextes kann man auf alle links im Editor aufgelisteten Elementgrößen per **Drag&Drop** zugreifen.

Achtung:

Man sollte sich bei der Programmierung des Algorithmus folgende Vorgehensweise angewöhnen:

1. Berechnung des Differenz-Potentials zwischen den Anschlüssen (Benutzung der von außen eingespeisten Potentialwerte).
2. Einspeisung der noch zu berechnenden Werte der Flussgrößen in alle Anschlüsse:



3. Schrittweise Ergänzung von Anweisungen "**von unten nach oben**", bis alle Flussgrößen sequentiell "**von oben nach unten**" aus den eingespeisten Parametern und Potentialgrößen eindeutig berechnet werden können:

Zuerst schreiben wir **i:=**.

Dann suchen wir einen Zusammenhang, wie man den Strom **i** aus anderen physikalischen Größen berechnen kann.

Im Beispiel ist es das ohmsche Gesetz (Proportionalität zwischen Strom **I** und Spannung **U** am

konstanten Widerstand R). Für das ohmsche Gesetz gibt es drei gleichberechtigte Gleichungsschreibweisen:

$$R = \frac{U}{I}; \quad U = R \cdot I; \quad I = \frac{U}{R}$$

Wir verwenden natürlich die Schreibweise, bei welcher der Strom allein auf einer Seite der Gleichung steht. Daraus formulieren wir die Zuweisung $i:=v/R$:

The screenshot shows the SimX software interface. On the left, a circuit model is displayed with two electrical connection points (ctr1 and ctr2), a resistor (R), and associated voltage (v) and current (i) variables. On the right, an algorithm editor window is open, showing the following code:

```

1 // enter your algorithm here
2 v := ctr1.v-ctr2.v;
3
4 i := v/R;
5 ctr1.i := i;
6 ctr2.i :=-i;

```

The editor has tabs for 'Algorithmus' and 'Gleichungen'.

Nun muss man überprüfen, ob in der zuletzt notierten Anweisung alle Größen auf der rechten Seite der Zuweisung wertmäßig bekannt sind. In unserem Beispiel ist das gewährleistet. R ist ein Element-Parameter und v haben wir bereits berechnet. Wir sind also fertig, denn alle Anweisungen in diesem Algorithmus können sequentiell von oben nach unten berechnet werden.

Existierte noch eine Größe mit unbekanntem Wert, würde man diese oberhalb in der nächsten Ergibt-Anweisung notieren und einen weiteren physikalischen Zusammenhang für ihre Berechnung suchen.

Wir können unser Element nun **fertigstellen** (hoffentlich ohne Fehlermeldung). Danach sollte man das Modell **RLC-Kreis.ism** speichern, damit die bisherige Arbeit auf der Festplatte gesichert ist.

Ob das Modell-Element richtig funktioniert, können wir im Vergleich zum R-Bibliothekselement einfach testen, da wir über ein lauffähiges Modell für einen Reihenschwingkreis verfügen:

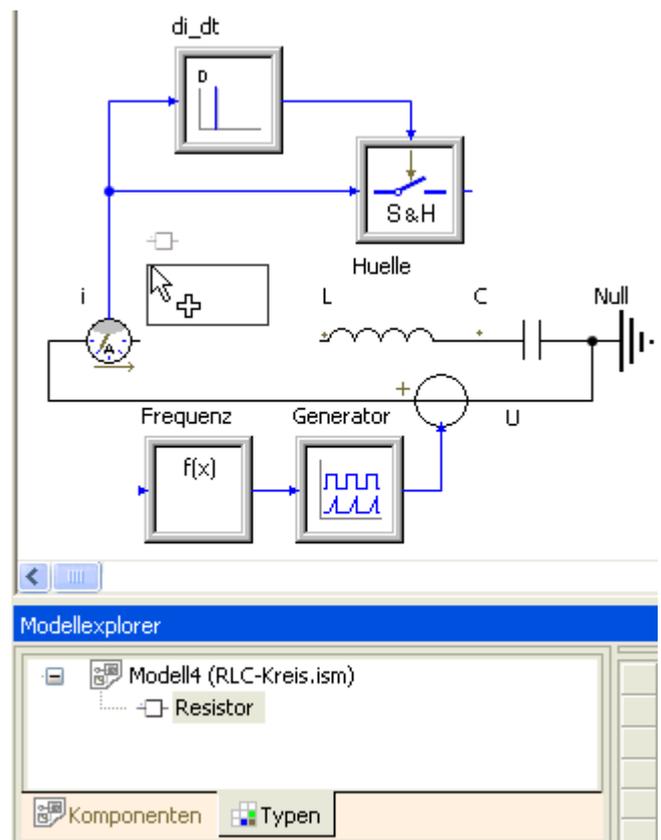
Wir löschen das bisherige Widerstandselement in der Modellstruktur.

Per **Drag&Drop** ziehen wir stattdessen unser Resistor-Element in das Modell und verbinden es über seine Anschlüsse.

Der alte Name R sollte wieder benutzt werden und man darf nicht vergessen, den Parameter R auf **1.3 Ohm** zu setzen.

Hinweis: Ergebnis-Fenster, welche Bezug auf gelöschte Modell-Elemente nehmen, werden automatisch mit gelöscht. Das Fenster für den Stromverlauf $R.i$ als Funktion von **Frequenz.y** müssen wir deshalb neu konfigurieren.

Wenn keine Flüchtigkeitsfehler passiert sind, sollte unserer eigenes Widerstandselement genauso funktionieren, wie das Bibliothekselement!

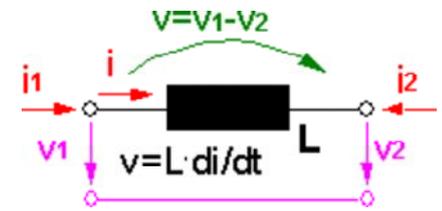


Elektrische Induktivitaet

Eine elektrische Induktivität ist ein elektrischer 2-Pol, wie der ohmsche Widerstand:

Der wesentliche Unterschied im Verhalten beider Elemente besteht darin, dass bei konstanter Induktivität der Spannungsabfall proportional zur Stromänderungsgeschwindigkeit $\dot{i} = di/dt$ ist.

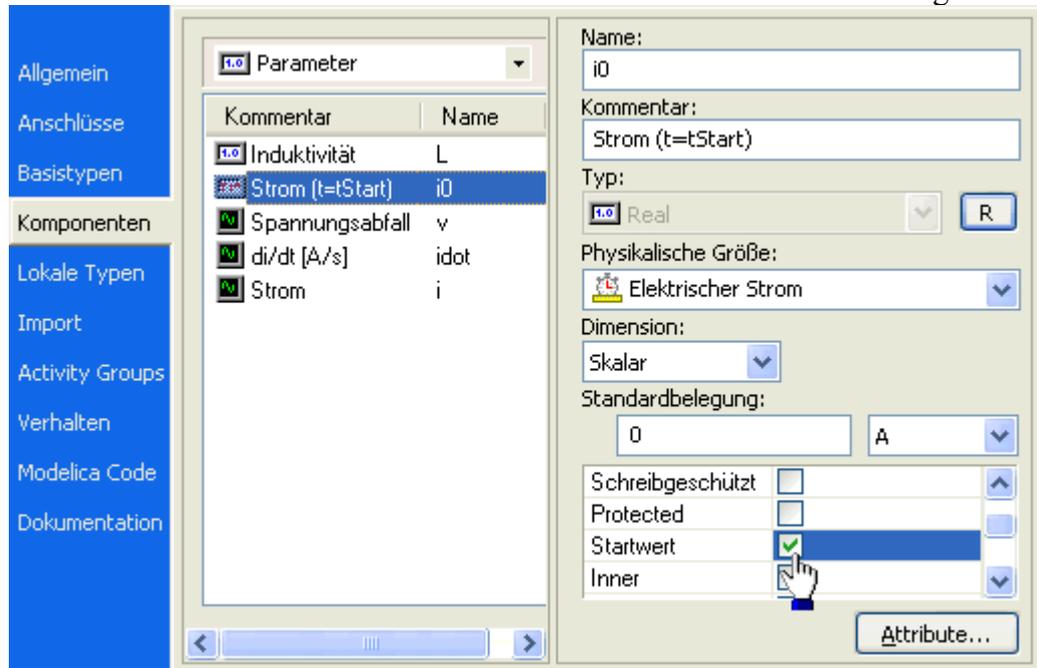
- Den aktuell fließenden Strom i kann man durch ein Zeit-Integral über idot berechnen.
- Für den Zeitpunkt t_{Start} benötigt der Strom i als Integralgröße einen Anfangswert i_0 .



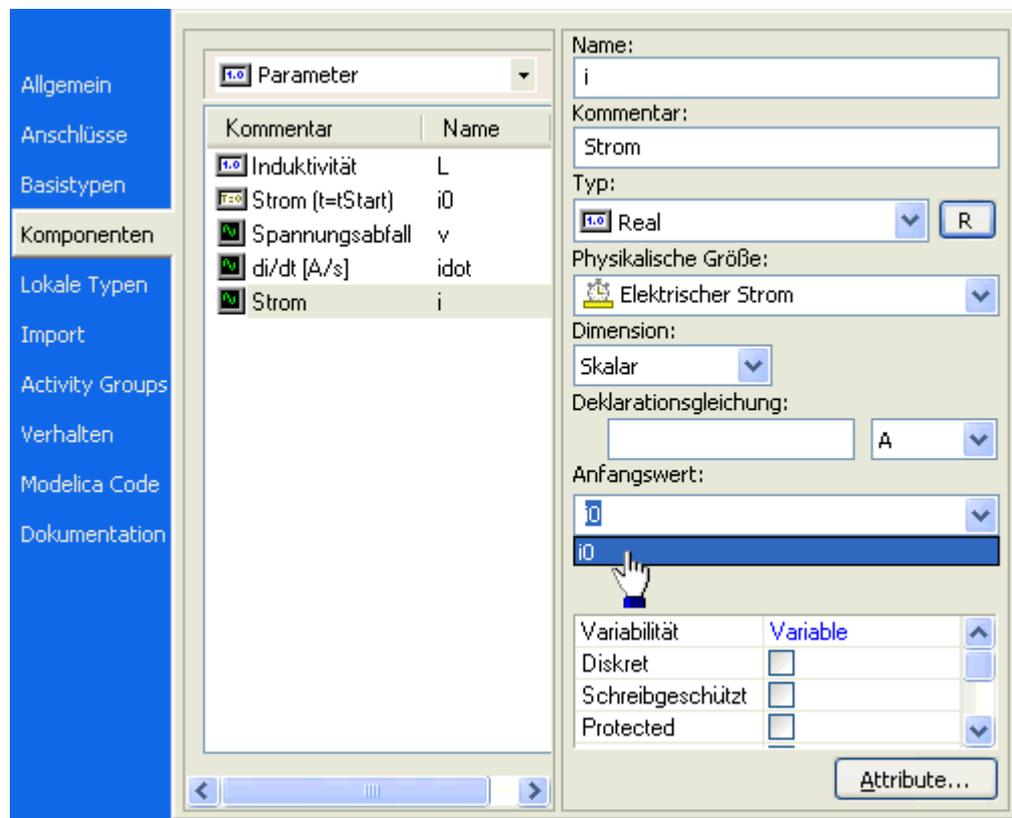
Die Vorgehensweise für die Bildung eines lokalen Elementtyps (Name=**Inductor** /Kommentar=**mit konstantem L**) ist analog der zu dem Widerstandselement. Es werden deshalb im Folgenden nur die Unterschiede beschrieben:

- Das Spulensymbol der Größe 61x31 Pixel ist mit dem Symboleditor etwas umständlich zu zeichnen. Es geht einfacher:
 - Man markiert das Spulensymbol in der Modellstruktur (Hervorhebung als graues Feld)
 - Mittels <Alt><Druck> kopiert man das komplette SimX-Fenster als Bild in die Zwischenablage.
 - Mit einem geeigneten Bildeditor produziert man daraus ein Bild der Größe 61x31 Pixel.
 - Im *TypeDesigner* holt man dann das Symbol aus der Zwischenablage oder lädt eine Bilddatei.
- Man benötigt 3 Komponenten für den Strom:

- Startwert i_0 :** Dieser wird als normaler Parameter definiert und dann als Startwert gekennzeichnet:



- Ableitung idot :** Wird als Variable definiert und benötigt die Maßeinheit **A/s**. Diese steht im SimX nicht als physikalische Größe für die Elektrotechnik zur Verfügung. In diesem Fall sollte man als Maßeinheit "-" vergeben und die Einheit als Teil des Kommentars notieren. In unserem speziellen Fall existiert die Einheit **A/s** jedoch in der Magnetik als "**Zeitliche Ableitung des magnetischen Potentials**". Diese Einheit kann man dann auch benutzen.
- Integralgröße i :** Wird als Variable definiert. Als Anfangswert wählt man den zuvor definierten Startwert i_0 :



Den Algorithmus-Abschnitt für das Verhalten definieren wir, wie es für das Widerstandselement beschrieben wurde:

1. Zuerst notieren wir die Übernahme der Potentialgrößen und die Belegung der Flussgrößen.
2. Die Berechnung des Stromes durch zeitliche Integration erstreckt sich über zwei Anweisungen:

```
idot := v/L;
i := integral (idot,i0);
```

Wir testen das Spulenelement wieder durch Ersatz des Bibliothekselements.

Elektrische Kapazitaet

Der Aufbau eines lokalen Elementtyps für die elektrische Kapazität (Name=**Capacitor** /Kommentar=**mit konstantem C**) sollte nun kein Problem mehr darstellen:

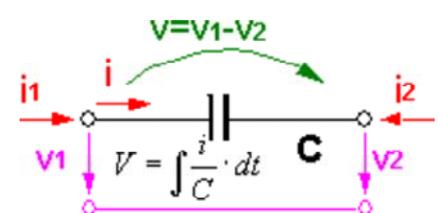
Die Integralgröße für die elektrische Kapazität ist die Spannung:

$$u(t) = \frac{1}{C} \cdot \int i(t) dt$$

Die elektrische Spannung ist im Algorithmus jedoch durch die an den Anschlüssen anliegende Spannung vorgegeben. Deshalb benötigen wir die Differentialform der Integralgleichung, um damit den Wert des Stromes berechnen zu können:

$$i(t) = C \cdot \frac{du(t)}{dt}$$

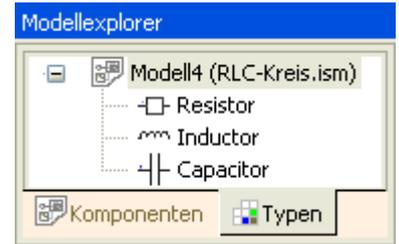
Im Verhaltensabschnitt widerspiegelt sich das in den beiden Anweisungen:



```
vdot := der(v);  
i     := C*vdot;
```

Achtung: Obwohl im Algorithmus kein Integral auftaucht, handelt es sich bei der Spannung v trotzdem um eine Integralgröße. Diese benötigt einen Anfangswert v_0 , den man im Komponenten-Abschnitt als Parameter (Startwert) definieren und der Variablen v als Anfangswert zuweisen muss.

Nach dem sicher erfolgreichen Test unseres Kondensator-Elements haben wir ein Gefühl dafür entwickelt, wie man physikalische Zusammenhänge als Algorithmus in einen Elementtyp einbindet.



← →

Von „http://www.optiyummy.de/index.php?title=Software:_SimX_-_Einfuehrung_-_Elektro-Chaos_-_Eigene_Elemente“

Software: SimX - Einfuehrung - Elektro-Chaos - Modellierung und Simulation

Aus OptiYummy

↑

← →

Modellierung und Simulation linearer Systeme (Zusammenfassung)

In den vorgehenden Abschnitten haben wir gelernt, wie man auf dem Gebiet elektrischer Netzwerke numerische Modelle aufbaut und damit Simulationen durchführt. Mit zielgerichteten Experimenten haben wir Antworten für bestimmte Fragestellungen gefunden. Verallgemeinert für beliebige Systeme sollen im Folgenden die wesentlichen Erkenntnisse zusammengefasst werden:

Modelle

1. Was ist ein "Modell"?

Ein Modell ist ein Ersatzobjekt beliebiger Natur, das man zur Gewinnung von Erkenntnis über ein Originalobjekt benutzt:

- . Ein Modell ist immer ein Abbild von einem "Original" (d.h. von "etwas anderem"). Bei einem "Original" kann es sich auch um ein Modell handeln!
- . Ein Modell bildet nur die Eigenschaften des Originals ab, die dem Modellbenutzer wichtig erscheinen:
 - . Bestimmte Eigenschaften eines Modells werden als Eigenschaften des Originals interpretiert (z.B. mathematische Formel als physikalischer Effekt).
 - . Es gibt immer Eigenschaften eines Modells, die keinen Bezug zum Original besitzen (z.B. Diskretisierung der Zeit durch den Solver).
- . Ein Modell wird pragmatisch und zweckorientiert angewendet:
 - . "Minimalmodell" - mit möglichst wenig Modell hinreichend viel Erkenntnis für aktuelle Fragestellungen (z.B. Kennlinie anstatt vollständiger Berechnung aller physikalischen Zusammenhänge).
 - . Erlaubt ist, was im Sinne des angestrebten Erkenntnisgewinns nützt und niemandem schadet (z.B. sind Tiere und Menschen nur eingeschränkt als Modelle nutzbar!)

2. Was ist ein "mathematisches Modell"?

Mathematische Modelle bilden die Eigenschaften des Originals auf ein mathematisches System ab (z.B. Lineare Algebra, Gewöhnliche Differentialgleichungen, Fuzzy-Logik):

- . Die Nutzung zum Erkenntnisgewinn ist möglich, weil Eigenschaften mathematischer Systeme als Aspekte der realen Welt interpretierbar sind.
- . Algorithmen der Computer-Algebra und der numerischen Mathematik ermöglichen die Behandlung in informationsverarbeitenden Systemen (z.B. Computer).
- . Wird ein mathematisches Modell in einem informationsverarbeitenden System implementiert, so spricht man von einem **numerischen Modell**.

Simulation

Simulation ist allgemein die Nutzung eines Modells. In der Wissenschaft und Technik werden Simulationen als Experimente mit Modellen durchgeführt, um Erkenntnisse über ein "Original" zu gewinnen. Dabei bedient man sich unterschiedlichster Modellformen:

- . Gedankenexperimente mit den Modell-Vorstellungen über die reale Welt sind das wesentliche Merkmal der

Gattung Mensch.

- . Physikalische Experimente mit materiellen Modellen, wenn mathematische oder numerische Modelle noch nicht existieren bzw. noch nicht hinreichend genau sind. Diese Form der Simulation ist meist zeit- und kostenaufwändig!
- . Varianten-Berechnungen mit "Bleistift/Taschenrechner und Papier" bei analytisch einfach lösbaren mathematischen Modellen. Diese Form ist fehleranfällig!
- . Numerische Experimente mit mathematischen Modellen, welche als numerische Modelle implementiert wurden (Computersimulationen). Diese bieten die Basis, um Prozesse des Erkenntnisgewinns zu automatisieren.

Lineare Systeme

Der Begriff "System" wird häufig mit verschiedenen Zusätzen benutzt:

- . lineares bzw. nichtlineares System
- . chaotisches System
- . offenes, geschlossenes, abgeschlossenes System
- . sequentielles und paralleles System
- . emergentes System;

Die vielfältigen Definitionen und Erläuterungen zu Systemen kann man auf folgende Aspekte reduzieren:

- . Ein System besitzt eine Abgrenzung zu seiner Umgebung.
- . Ein System enthält Komponenten, die Wirkung aufeinander auswirken.
- . Es entsteht ein Systemverhalten, welches eine neue Qualität im Vergleich zu den Elementen aufweist (z.B. Spule und Kondensator).
- . Die resultierenden Systemgesetze engen den Freiheitsgrad der Komponenten ein! (Versklavung der Komponenten).

Lineare Systeme müssen folgende zwei Merkmale aufweisen:

1. Alle Elemente besitzen unabhängig von ihrer Belastung konstante Eigenschaften, z.B.:

- . die "Kapazität" eines Energiespeichers ist unabhängig vom "Füllstand" (Kondensator mit konstantem C , Induktivität mit konstantem L , Feder mit konstanter Federkonstante c)
- . Übertragungselemente zwischen den Speichern besitzen einen konstanten "Verlustfaktor" (Widerstand mit konstanten R , Reibelement mit konstantem μ).

2. Die in das System eingespeisten Signale sind aus der Sicht des Systems vorhersagbar:

- . konstante Signale (z.B. Gleichspannung)
- . periodische Signale (z.B. sinusförmige Wechselspannung)
- . Einzelimpuls (Dirac-Impuls, Einheitssprung)

Für die Behandlung linearer Systeme existiert ein umfangreiches mathematisches Methoden-Inventar. Das Verhalten linearer Systeme kann analytisch durch Anwendung der entsprechenden mathematischen Verfahren auf der Basis linearer Gleichungssysteme berechnet werden. Wenn sich die Eingangssignale des Systems auf ein Spektrum harmonischer Schwingungen zurückführen lassen, so kommt man zu den linearen Gleichungssystemen, indem man eine Transformation in einen Bildbereich vornimmt:

- . Sinus (Bildbereich der komplexen Zahlen),
- . Sprungfunktion und DIRAC-Impuls (Laplace-Transformation),
- . beliebige periodische Vorgänge (Z-Transformation).

Analytische Lösungen sind exakte Lösungen einer Gleichung. D.h., man kann alle Ergebnisgrößen direkt mit jeweils einer Formel ausrechnen. In der Praxis nutzt man "manuelle" analytische Lösungen nur für einfache lineare Systeme (z.B. einfache RLC-Glieder):

- . Das Umstellen der Formeln nach den zu berechnenden Größen ist fehleranfällig und zeitaufwändig.
- . Das Gleiche gilt für die Berechnung der Formeln mit dem Taschenrechner.
- . Die numerische Simulation linearer numerischer Modelle liefert zwar keine exakte Lösung, ist aber hinreichend genau, schnell, zuverlässig und flexibel.

← →

Von „http://www.optiyummy.de/index.php?title=Software:_SimX_-_Einfuehrung_-_Elektro-Chaos_-_Modellierung_und_Simulation“

Software: SimX - Einfuehrung - Elektro-Chaos - Dioden-C mit Kennlinie

Aus OptiYummy

↑

← →

Dioden-Kapazität mit Datenblatt-Kennlinie

Jede Halbleiterdiode besitzt eine Sperrschichtkapazität. Der Wert dieser Kapazität ist abhängig von der anliegenden Sperrspannung. Dies nutzt man bei der sogenannten Kapazitätsdiode gezielt aus, um z.B. die Resonanzfrequenz eines Schwingkreises mittels einer veränderlichen Spannung einzustellen.

Die Hersteller der Kapazitätsdioden stellen für jeden Diodentyp Datenblätter mit den wichtigsten Parametern zur Verfügung. Wir beziehen uns im weiteren auf die **Diode BB 512** von SIEMENS. Das zugehörige Datenblatt findet man z.B. auf <http://www.datasheetcatalog.com> als **PDF-Datei zum Download**:

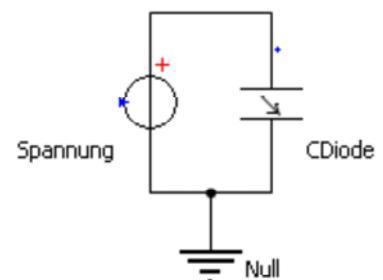
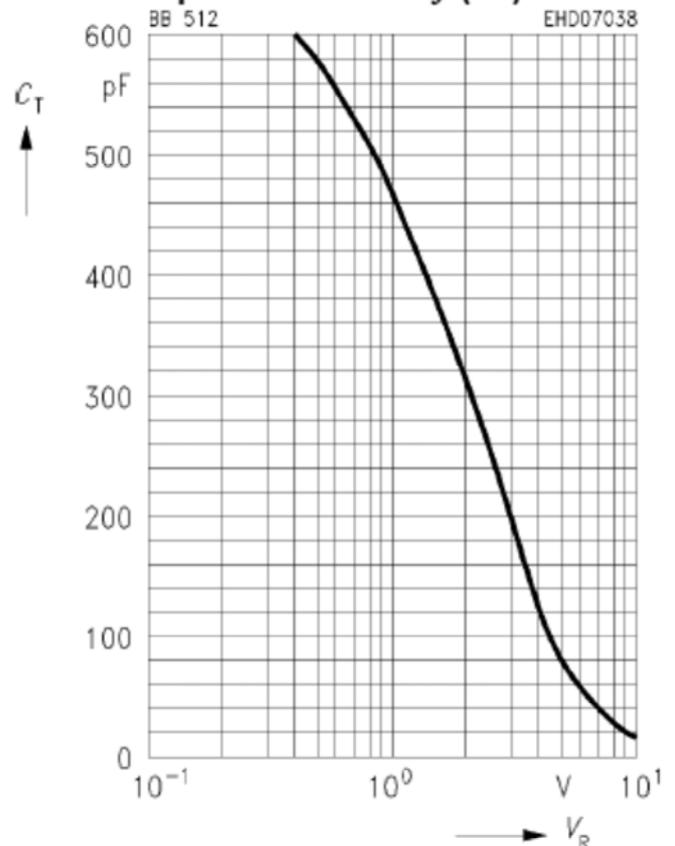
- Uns interessiert auf Seite 3 des Datenblatts die Kennlinie für die Abhängigkeit der Kapazität von der Sperrspannung $C_T = f(V_R)$ (Siehe: Bild rechts).
- Diese Kennlinie werden wir anstatt eines konstanten C-Wertes in einem neu zu gestaltendem Kapazitätselement benutzen.

Die Benutzung einer spannungsabhängigen C-Kennlinie in einem Kondensator-Modell ist nicht trivial. Im Sinne der Fehler-Minimierung gehen wir hierbei behutsam und schrittweise vor:

Neuer Elementtyp

- Wir beginnen mit einem neuen SimX-Modell, das wir unter dem Namen **C-Kennlinie.ism** speichern.
- Darin definieren wir einen neuen lokalen Modell-Elementtyp (Name=**VarCapacitor** / Kommentar=**C-Kennlinie**) mit einem Symbol 61x31 Pixel, welches eine variable Kapazität andeutet.
- Das Element erhält zwei elektrische Anschlüsse.
- Als Komponenten definieren wir den Spannungsabfall und den Strom.
- Wir übernehmen im Verhaltensalgorithmus wie üblich die Spannungsdifferenz der Anschlüsse und speisen die Anschluss-Ströme mit dem Strom.
- Den Strom durch das Element setzen wir vorläufig auf Null. D.h. es liegt eine Spannung am Element an, aber es fließt dabei kein Strom.
- Zur Überprüfung dieses Verhaltens benutzen wir im Modell eine Spannungsquelle mit Signaleingang. Als Spannungswert v_{scr} verwenden wir die Simulationszeit t anstatt des Eingangs **in1**. Eine Simulation bis **tStop=10 s** sollte die Spannung an der Kapazität linear von **0...10 V** ändern.

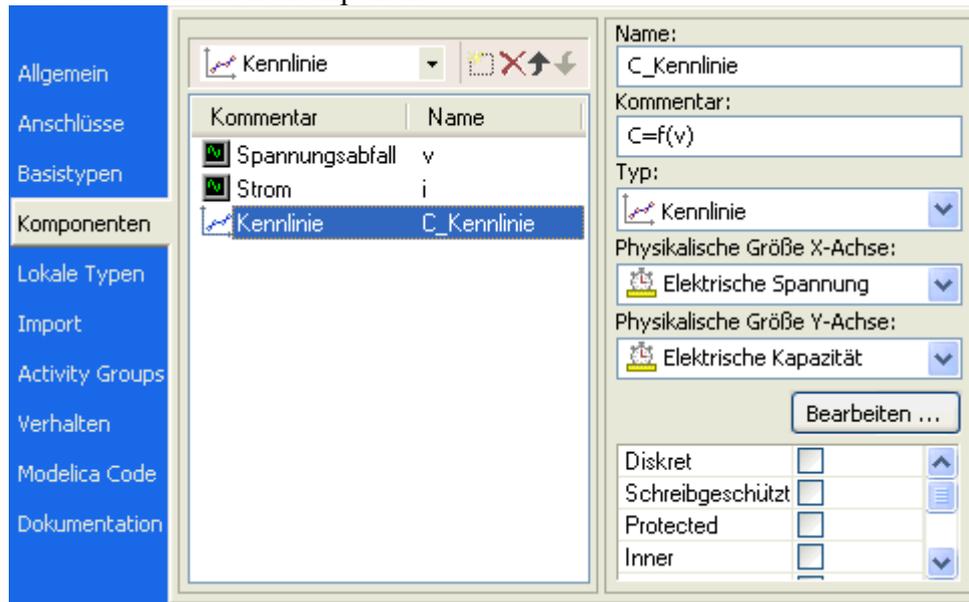
Diode capacitance $C_T = f(V_R)$



C-Kennlinie

Funktioniert die Spannungsänderung an der Kapazität, so implementieren wir als Nächstes die Werte der C-Kennlinie und lesen sie in Abhängigkeit von der anliegenden Spannung aus:

Dazu benötigen wir eine Kennlinien-Komponente:



Im Datenblatt ist die Kennlinie zwischen 0.4 V und 10 V dargestellt. Für diesen Bereich muss man versuchen, die Werte möglichst exakt abzulesen. Das ist sehr subjektiv. Es wird später noch erläutert, wie man dabei zu den eigentlich nicht ablesbaren drei Zifferstellen kommt. Die C-Werte sind in der Basiseinheit **F** angegeben, um Umrechnungsfehler im Element-Algorithmus zu vermeiden:

```

x[V]      y[1][F]
0.4      6e-010
0.5      5.77e-010
0.6      5.53e-010
0.7      5.3e-010
0.8      5.08e-010
0.9      4.88e-010
1        4.69e-010
2        3.1e-010
3        2e-010
4        1.29e-010
5        8.37e-011
6        5.73e-011
7        3.9e-011
8        2.72e-011
9        2e-011
10       1.5e-011

```

Es wurden Werte oberhalb von 10 V so ergänzt, dass auch bei höherer Spannung keine extrapolierten negativen C-Werte entstehen:

```

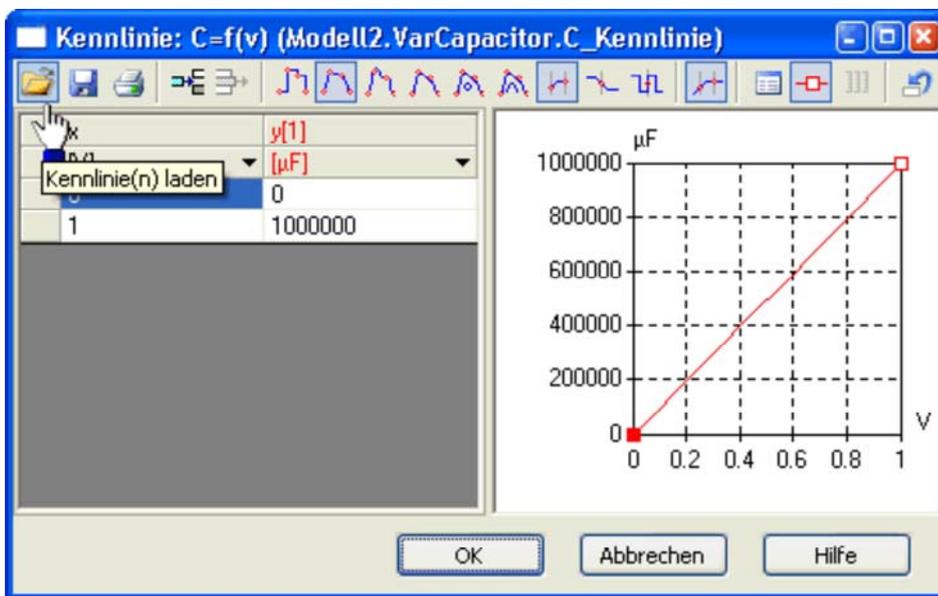
11       1.2e-011
12       1.06e-011
13       1e-011
14       9.5e-012
15       9.395e-012

```

Achtung:

Im Rahmen der Übung werden die Werte in Form einer archivierten Textdatei bereitgestellt (**C_Kennlinie.zip**):

Nach dem Betätigen des **Bearbeiten**-Button für die Kennlinien-Komponente öffnet sich der Kennlinien-Editor. Darin beginnen wir mit "Kennlinie laden":



Nach Wahl der Text-Datei öffnet sich der Importassistent, in dem man die Struktur der Daten-Tabelle beschreiben kann. Im Beispiel wurde die Tabelle im vorgegebenen Default-Format gespeichert. Der Import muss jedoch mit **Zeile 1** beginnen:

Zeile	Wert
0.4	6e-010
0.5	5.77e-010
0.6	5.53e-010
0.7	5.3e-010
0.8	5.08e-010
0.9	4.88e-010
1	4.69e-010
2	3.1e-010
3	2e-010

Die Schritte 2 und 3 des Import-Assistenten kann man im Beispiel einfach Bestätigen:

Importassistent (Seite 2 von 3)

Argumente (X-Achse)

Spalte:

Bezeichnung:

Phys. Größe:

Maßeinheit:

x	y[1]
0.4	6e-010
0.5	5.77e-010
0.6	5.53e-010
0.7	5.3e-010
0.8	5.08e-010
0.9	4.88e-010
1	4.69e-010
2	3.1e-010

Importassistent (Seite 3 von 3)

Funktionswerte (Y-Achse)

Spalte:

Bezeichnung:

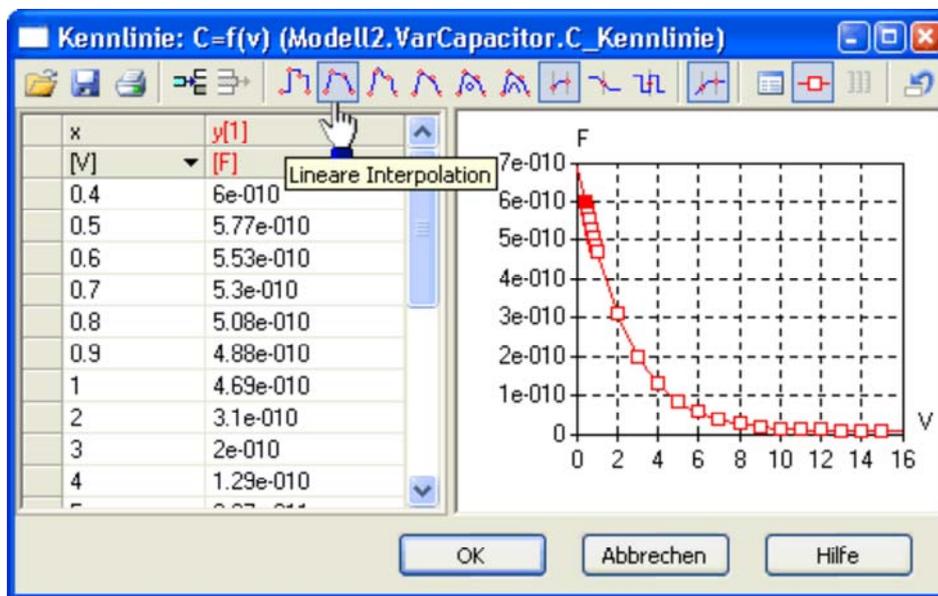
Phys. Größe:

Maßeinheit:

x	y[1]
0.4	6e-010
0.5	5.77e-010
0.6	5.53e-010
0.7	5.3e-010
0.8	5.08e-010
0.9	4.88e-010
1	4.69e-010
2	3.1e-010

Wir verwenden eine **lineare Interpolation**, benutzen **keine Sonderbehandlung** und lassen eine **lineare Extrapolation** zu.

Abgesehen von der linearen Teilung der X-Achse sollte die eingelesene Kennlinie den gleichen Verlauf besitzen, wie die Kennlinie des Datenblattes:



Bevor wir eine Variable für das Auslesen der C-Kennlinie definieren, noch einen Hinweis zur Bedeutung dieser Kennlinie:

Es handelt sich hierbei nicht um den absoluten Kapazitätswert, welcher definiert ist durch die Ladungsmenge, welche bei einer bestimmten Kondensatorspannung im Kondensator gespeichert ist:

$$C = \frac{Q}{U}$$

Solche Dioden benutzt man z.B. für die Abstimmung von Schwingkreisen. Deren Wechselspannung ist dann wesentlich kleiner als die Sperrschichtspannung! Es interessiert hier nur das Kleinsignalverhalten im aktuellen Arbeitspunkt. Die bereitgestellte Kennlinie beschreibt deshalb die sogenannte differentielle Kapazität:

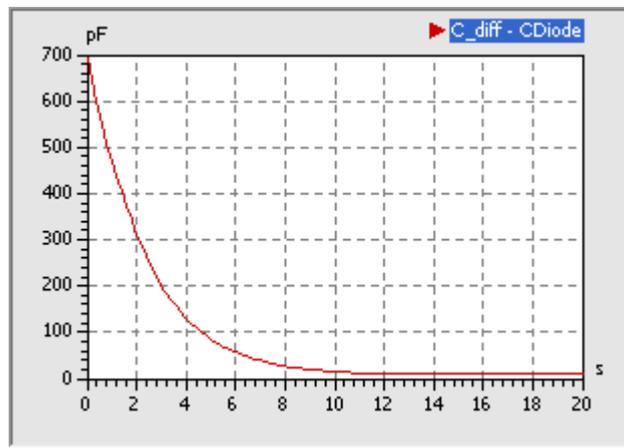
$$C_D = \frac{dQ}{dU}$$

Die differentielle Kapazität C_D entspricht der Tangente (=Anstieg) an der $Q=f(U)$ -Kennlinie der Diode im aktuellen Arbeitspunkt (U, Q).

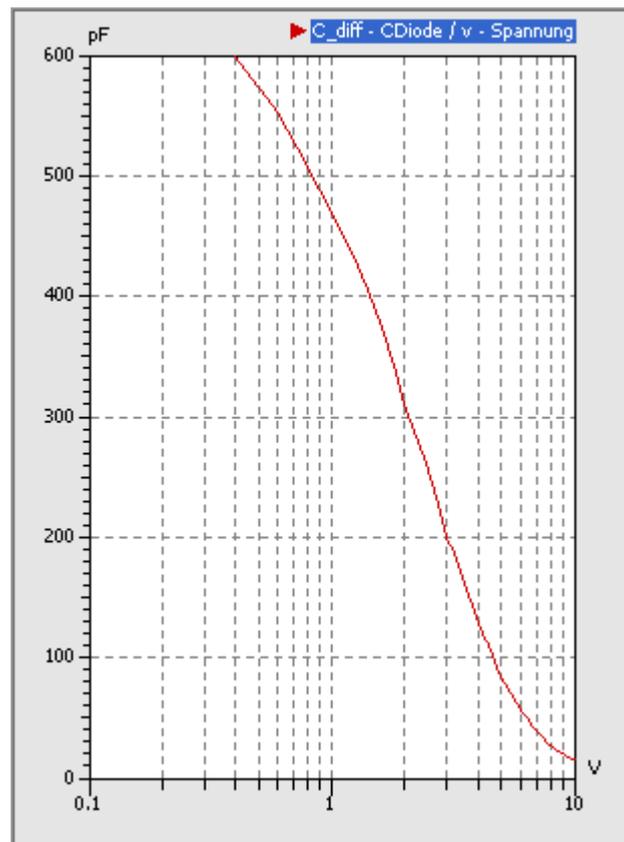
Wir definieren deshalb als Komponente eine Variable C_{diff} und weisen diese im Verhaltensalgorithmus den aktuellen Kennlinienwert zu. Dabei benutzen wir den Absolutwert der Spannung, um für den negativen Spannungsbereich numerische Probleme zu vermeiden:

```
C_diff:=C_Kennlinie(abs(v));
```

Eine Simulation bis $t_{Stop}=20$ s ergibt die folgende Kennlinie für die differentielle Kapazität, wobei $1 \text{ s} = 1 \text{ V}$ entspricht:



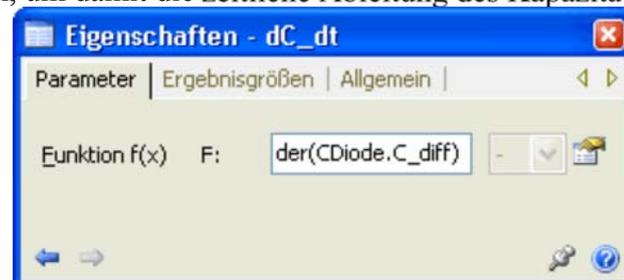
Die Abhängigkeit der Dioden-Kapazität von der anliegenden Spannung kann man durch geeignete Konfiguration eines y(x)-Signalfensters in der gleichen Form darstellen, wie im zugrunde gelegten Datenblatt:



Wenn man genau hinschaut, sieht man in der Kennlinie einige kleine "Knicke".

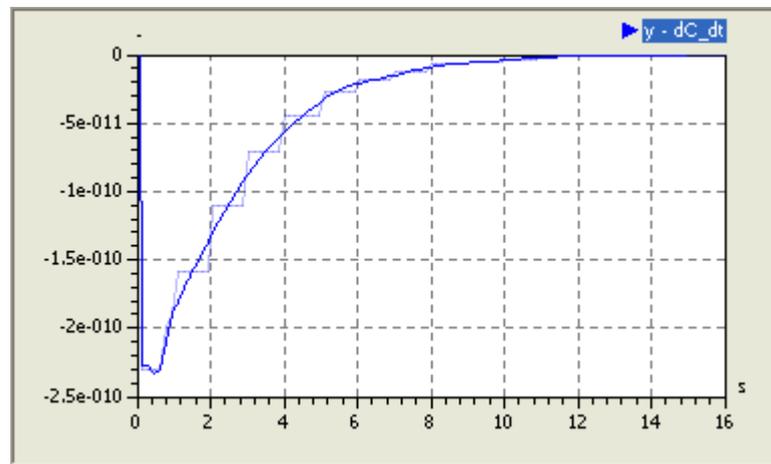
Es kann zwar sein, dass dies auf die Genauigkeit der späteren Simulation keinen Einfluss hat. Aber natürlich sollte man versuchen, eine nichtlineare Kennlinie möglichst gut nachzubilden.

Ein Hilfsmittel hierfür ist die Analyse der Ableitung der nachzubildenden Kennlinie. Dazu ergänzen wir im Modell einen Function-Block, um damit die zeitliche Ableitung des Kapazitätsverlaufes zu bilden:



Die Ableitung ist stufenförmig (eingefrorene blaue Kurve). Dies ist richtig, denn wir benutzen für die Kennlinie bisher eine lineare Interpolation.

Über **Eigenschaften > Bearbeiten** des Elements **CDiode** können wir für die Kennlinie z.B. die Spline-Interpolation wählen, welche wesentlich stetigere Verläufe ergibt:



Andere mögliche Interpolationsarten ergeben schlechtere Approximationen der Kennlinie.

Die in der Text-Datei für diese Übung bereitgestellten Kennlinien-Punkte wurden mit der Spline-Interpolation gewonnen:

- Die ursprünglich anhand des Datenblattes "geschätzten" Punkte ergaben eine sehr wellige Ableitung.
- Diese ursprünglichen Y-Werte wurden solange iterativ um "winzige" Beträge verschoben, bis die resultierende Ableitung der Kennlinie den jetzigen "unwelligen" Verlauf erreichte.
- Dieses Verfahren ist sehr aufwändig. Im Interesse einer hohen Modellqualität sollte man jedoch darauf achten, dass importierte Funktionsverläufe einschließlich ihrer Ableitungen möglichst stetig und ohne Welligkeit sind.

Modell der Dioden-Kapazität

Der resultierende Strom wurde bisher auf Null gesetzt. Um im Algorithmus das dynamische Verhalten der spannungsabhängigen differentiellen Kapazität abzubilden, müssen wir nun schrittweise **von unten nach oben** die Berechnung des Stroms vornehmen. Dazu sind einige Vorbetrachtungen erforderlich:

Die differentielle Kapazität ist laut Definition:

$$C_D = \frac{dQ}{dU}$$

Der Zusammenhang zwischen elektrischen Strom und Ladung ist:

$$I = \frac{dQ}{dt}$$

Stellt man diese Gleichung nach **dQ** um und setzt diesen Ausdruck in die Definition der differentiellen Kapazität ein, so erhält man:

$$C_D = \frac{I \cdot dt}{dU}$$

D.h., wir können im Algorithmus den Strom sehr einfach berechnen mit:

```

vdot := der(v);
i     := C_diff*vdot;

```

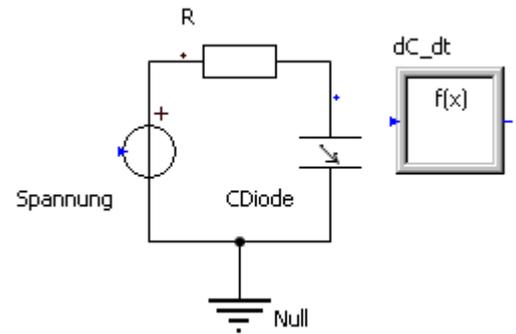
Die erforderliche Variable **vdot** für die zeitliche Ableitung der Spannung müssen wir als zusätzliche Komponente definieren.

Außerdem ist die Kondensator-Spannung eine Integralgröße und erfordert deshalb die Zuweisung eines Anfangswertes!

Nach diesen Ergänzungen sollte das Modell noch genauso rechnen, obwohl wir keinen Vorwiderstand verwenden,

was physikalisch unmöglich ist. Das funktioniert nur, weil wir von Null beginnend die Spannung langsam erhöhen.

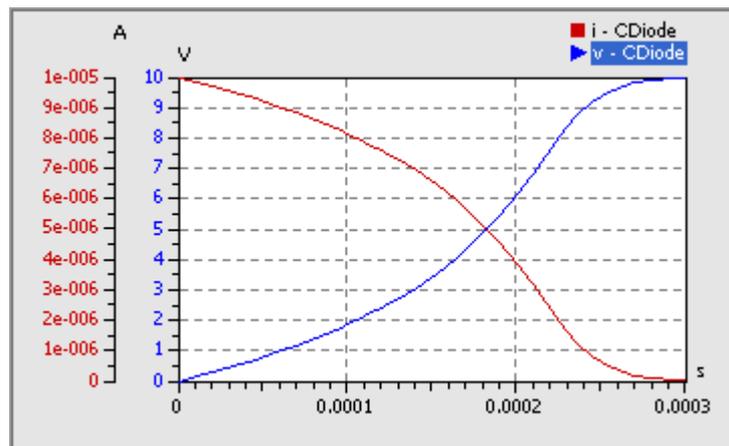
Im nächsten Schritt wollen wir eine konstante Betriebsspannung von **10 V** vorgeben und den Kondensator über einen Vorwiderstand **R** aufladen:



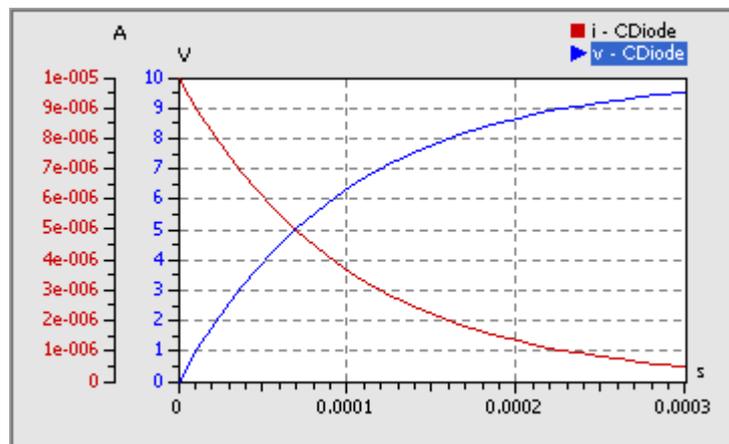
Da die maximale Kapazität ungefähr **1 nF** beträgt, werden wir den Vorwiderstand so dimensionieren, dass wir damit eine Zeitkonstante von **1 ms** erreichen. Die tatsächliche Zeitkonstante wird sich natürlich durch die sich verringernde Kapazität verändern.

Wir müssen in der Simulationsteuerung die Simulationszeit und die Schrittweiten anpassen.

Es ergeben sich folgende Strom- und Spannungsverläufe für den Aufladevorgang:



Diese Verläufe unterscheiden sich wesentlich vom Aufladevorgang einer konstanten Kapazität von z.B. **100 pF**, welche hier zum Vergleich dargestellt ist:



In unserem Elementtyp werden wir zusätzlich noch den Wert der absoluten Kapazität berechnen. Dies ist für die Simulation des Verhaltens nicht erforderlich, aber sicher von Interesse:

Die Kapazität C eines Kondensators ist definiert als

$$C = \frac{Q}{U}$$

Da die anliegende Spannung im Element bekannt ist, wird nur noch der Wert der aktuell im Kondensator gespeicherten Ladung benötigt. Der Wert der Ladung ergibt sich durch die zeitliche Integration des Kondensatorstroms:

$$Q(t) = Q(t_0) + \int_{t_0}^t I(t) dt$$

Nach Definition der erforderlichen Komponenten (Parameter, Variable) genügen zwei Anweisungen für die Berechnung der absoluten Kapazität:

```
q := integral(i,q0);
C_abs := q/v;
```

Leider kommt es sofort nach dem Start der Simulation sofort zu einer Division durch Null. Das ist verständlich, weil die Kondensatorspannung v am Anfang Null ist.

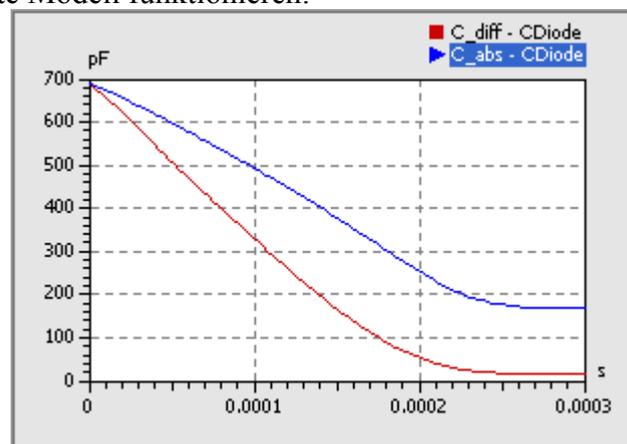
Es lässt sich auch kaum vermeiden, dass während des Betriebs die Spannung am Kondensator auch mal den Wert Null annimmt. Wir müssen also eine Möglichkeit finden, dass dabei trotzdem ein sinnvoller Wert für C_{abs} berechnet wird:

Im Beispiel kommt uns zu Hilfe, dass für eine Spannung von Null die Tangente und die Sekante zum Nullpunkt an der Funktion $Q=f(U)$ den gleichen Anstieg haben. D.h., absolute und differentielle Kapazität sind dann gleich.

Das muss man nur noch durch eine geeignete bedingte Anweisung formulieren. Wir ersetzen die bisherige Berechnung von C_{abs} durch:

```
C_abs := (if v=0 then C_diff else q/v);
```

Das wird für das konfigurierte Modell funktionieren:



Bei maximaler Spannung ist trotz kleiner wirksamer differentieller Kapazität der Absolutwert der Kapazität $C_{abs}=Q/U$ wesentlich größer.

Leider ist die Programmierung von bedingten Anweisungen in den Elementtypen kein triviales Problem, weil dabei die sogenannte Ereignisbehandlung wirksam wird:

Die numerische Integration versucht, den Umschalt-Zeitpunkt der formulierten Bedingungen möglichst exakt zu treffen.

Es wird dabei einen Zeitpunkt angesteuert, der ganz dicht hinter dem tatsächlichen Ereigniszeitpunkt liegt. Das wird dann als Eintreten des Ereignisses interpretiert!

Erst wenn das Ereignis damit "eingetreten" ist, erfolgt ein Umschalten der Bedingung (TRUE/FALSE). Damit ist es möglich, dass der eigentlich verbotene Zweig der Bedingung beim Ansteuern des Ereigniszeitpunktes trotzdem abgearbeitet wird. Es kann mit obiger **if**-Anweisung also trotzdem zur Division durch Null kommen!

Abhilfe schafft hier der **noEvent**-Operator. Damit kann man die Ereignisbehandlung z.B. für die bedingte Anweisung abschalten:

```
C_abs := noEvent(if v==0 then C_diff else q/v);
```

Erst dann erfolgt die Auswertung des bedingten Ausdrucks direkt und es kann zu keiner Division durch Null kommen.

Wir haben damit einen eigenen Elementtyp für eine kennliniengesteuerte differentielle Kapazität erstellt. Im Prinzip funktioniert die Kapazität wie die einer C-Diode. Allerdings haben wir nur den Sperrbereich der Diode nachgebildet. Im Durchlassbereich benutzen wir die gleiche Funktion wie im Sperrbereich, was natürlich falsch ist!

Achtung: Speichern der Modelldatei **C-Kennlinie.ism** nicht vergessen!

← →

Von „http://www.optiyummy.de/index.php?title=Software:_SimX_-_Einfuehrung_-_Elektro-Chaos_-_Dioden-C_mit_Kennlinie“

Software: SimX - Einfuehrung - Elektro-Chaos - Parameter-Identifikation

Aus OptiYummy

↑

← →

Parameter-Identifikation (Spice-Modell der Diode)

Das Dioden-Modell in der SimulationX-Bibliothek bildet das Verhalten einer Halbleiter-Diode bedeutend besser ab, als unser einfaches Kennlinien-Modell für die Sperrschichtkapazität. Deshalb sollte man in einer elektronischen Schaltung möglichst das Bibliothekselement nutzen, wenn man eine C-Diode benötigt.

Leider gibt es keine Möglichkeit, eine beliebige C-Kennlinie direkt in dieses Dioden-Modell einzulesen. Die Kennlinie wird darin durch eine vorgegebene Funktion nachgebildet, deren Parameter man einstellen kann.

Inhaltsverzeichnis

- . 1 Modell mit Guete-Kriterium
- . 2 Finden optimaler Parameter mit numerischer Optimierung
 - . 2.1 Experiment-Workflow
 - . 2.1.1 Entwurfsparameter (Nennwerte)
 - . 2.1.2 Guetekriterium
 - . 2.2 Experiment-Konfiguration
 - . 2.2.1 Optimierungsverfahren
 - . 2.2.2 Visualisierung
 - . 2.3 Experiment-Durchfuehrung
 - . 2.3.1 Lokale Suche
 - . 2.3.2 Globale Suche

Modell mit Guete-Kriterium

Das Ziel der nächsten Experimente soll die Parameter-Findung für diese Kennlinien-Funktion unter Nutzung der numerischen Optimierung sein. Dazu erstellen wir aus unserem Modell **C-Kennlinie.ism** eine Kopie **Parameterfindung.ism**. Dieses Modell werden wir im Folgenden nutzen:

- . Das Element **dC_dt** für die Bildung der Ableitung können wir in der Modellstruktur löschen.
- . Wir holen uns eine Diode aus der Modell-Bibliothek und öffnen den Eigenschaftsdialog. Hier kann man zwischen verschiedenen Typen von Dioden-Modellen umschalten:



- . Das Spice-ähnliche Diodenmodell ist das, welches unter anderem auch die Sperrschicht-Kapazität in Abhängigkeit von der Spannung berücksichtigt. Ruft man innerhalb des Eigenschaftsdialogs mit **<F1>** die Hilfe auf, so erhält man einen Überblick über die implementierten physikalischen Effekte und die zugehörigen Modell-Parameter:

The **dynamic behaviour** of the diode in the blocking state is determined by the junction capacity

$$C_j = C_{j0} \begin{cases} \frac{1}{\left(1 - \frac{v_i}{V_j}\right)^M} & \text{for } v_i \leq F_c V_j \\ 1 - F_c(1 - M) + \frac{M v_i}{V_j} & \text{else} \end{cases}$$

Unsere Kennlinie ist durch den oberen Zweig der Formel beschrieben. Der untere Teil beschreibt den Durchlassbereich der Diode.

Für den oberen Zweig muss man die Parameter **Cj0** und **M** so wählen, dass eine möglichst gute Übereinstimmung zu unserer Kennlinie der Diode **BB 512** erreicht wird.

Das könnte man mit etwas Mühe auch noch manuell machen. Wir werden diesen Prozess jedoch mit Hilfe der numerischen Optimierung automatisieren.

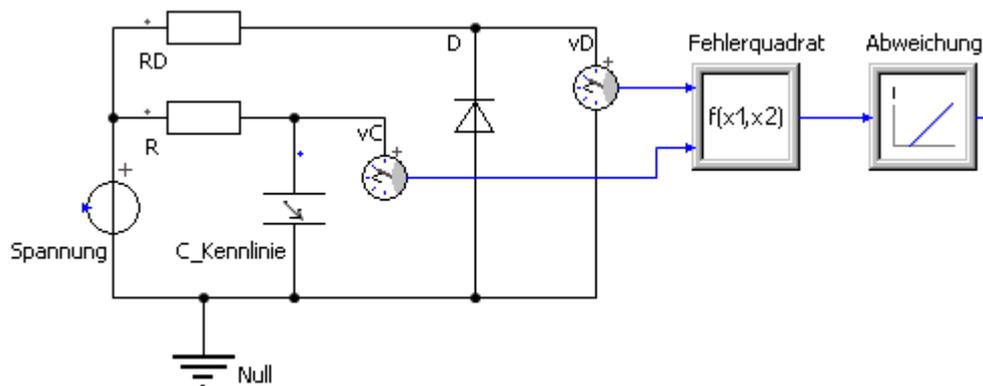
Im Dioden-Modell wird der aktuelle Wert der Kapazität nicht als Variablen-Wert berechnet. Deshalb entfällt der direkte Vergleich der Kapazitäten von Diode und Kennlinien-Modell als Kriterium für die Optimierung. Man könnte mit etwas Aufwand die Berechnung der Dioden-Kapazität ergänzen, aber wir gehen einen anderen Weg der Parameter-Identifikation:

Wenn unsere Kennlinien-Kapazität das gleiche Verhalten beim Aufladen zeigt, wie die Sperrschicht-Kapazität der Diode, dann sollte die Abhängigkeit der Kapazität von der Spannung in beiden Elementen den gleichen Verlauf haben.

Wir konfigurieren deshalb unser Modell so, dass die C-Diode und unsere Kennlinien-Kapazität gleichzeitig unter gleichen Bedingungen aufgeladen werden.

Die Abweichung zwischen den Signalen der Kondensatorspannungen sollen minimal werden. Deshalb wird diese Abweichung im Modell als Gütekriterium berechnet.

Die dafür erforderliche Modellstruktur ist ziemlich einfach:



Das eigene Kennlinien-Element wurde jetzt **C_Kennlinie** genannt.

Man muss darauf achten, dass die Diode **D** in Sperr-Richtung betrieben wird.

Der Zeitbereich muss so gewählt werden, dass mit dem Vorwiderstand **R** die Kennlinien-Kapazität bis zur Spannung von **10 V** aufgeladen wird.

Der Vorwiderstand **RD** der Diode **D** besitzt den gleichen Wert, wie der Vorwiderstand **R** des kennliniengesteuerten Kondensators.

Die Ausgangssignale der beiden Spannungssensoren werden in einen **Function2**-Block eingespeist. Dieser bildet das Fehlerquadrat der aktuellen Spannungsdifferenz: $(x1-x2)^2$.

Die Abweichung zwischen den Spannungsänderungen wird als **Fehlerquadratsumme** durch Integralbildung berechnet.

Es müssen nun die **dynamischen Parameter** der Diode **Cj0** und **M** gefunden werden, welche die kleinste Abweichung beim Aufladevorgang beider Kapazitäten ergeben.

Cj0 ist der Wert der Sperrschicht-Kapazität für die Spannung Null. Dieser Wert wird also etwas größer als 600 pF sein. Mit **cj0=1000 pF** haben wir einen realistischen Anfangswert.

M ist ein Koeffizient, welcher die Krümmung der C-Funktion beschreibt. Hier könnte man manuell nach

günstigen Werten suchen, aber wir behalten den Vorgabewert $M=0,5$ und überlassen die Parametersuche der numerischen Optimierung!

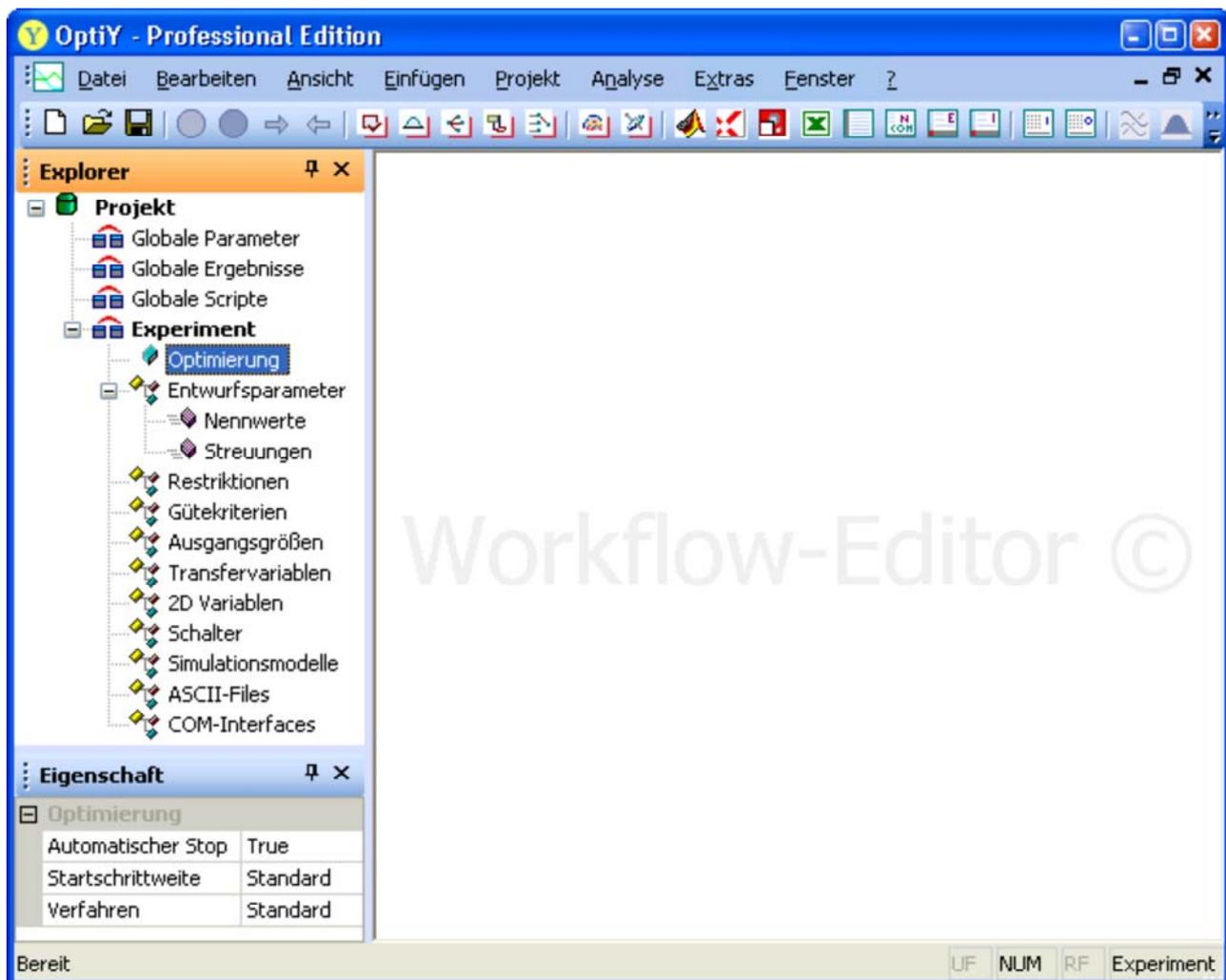
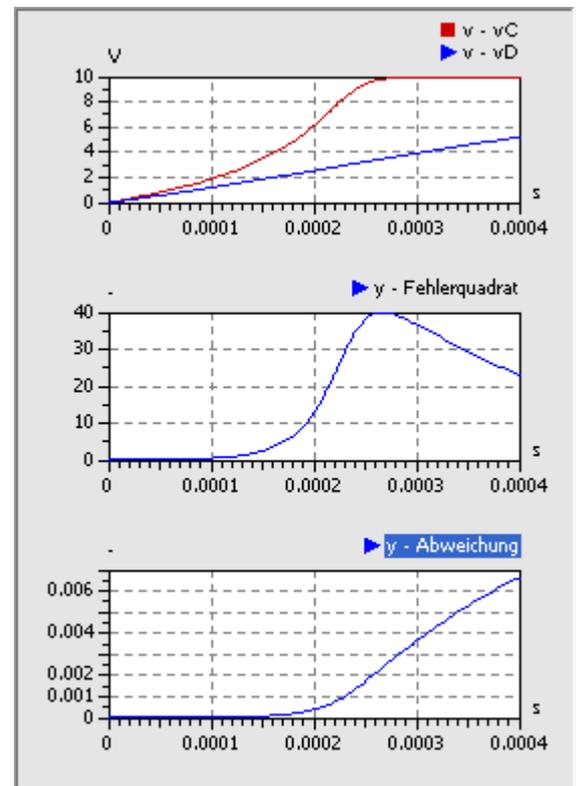
Für die restlichen Dioden-Parameter benutzen wir ebenfalls die Vorgabewerte.

Das für die Parameterfindung konfigurierte Modell sollte das nebenstehende Verhalten zeigen.

Finden optimaler Parameter mit numerischer Optimierung

Als Optimierungstool nutzen wir **OptiY**. Die *Trial Edition* ist ausreichend, welche unter www.optiy.de/Download.html bereitgestellt wird. Im Rahmen dieser Übung wird die Konfiguration des erforderlichen Optimierungsexperiments für den Einsteiger beschrieben. Vorkenntnisse zum Programm **OptiY** sind also nicht erforderlich:

Nach der wahrscheinlich problemlosen Installation unter Windows kann das Programm **OptiY** gestartet werden:

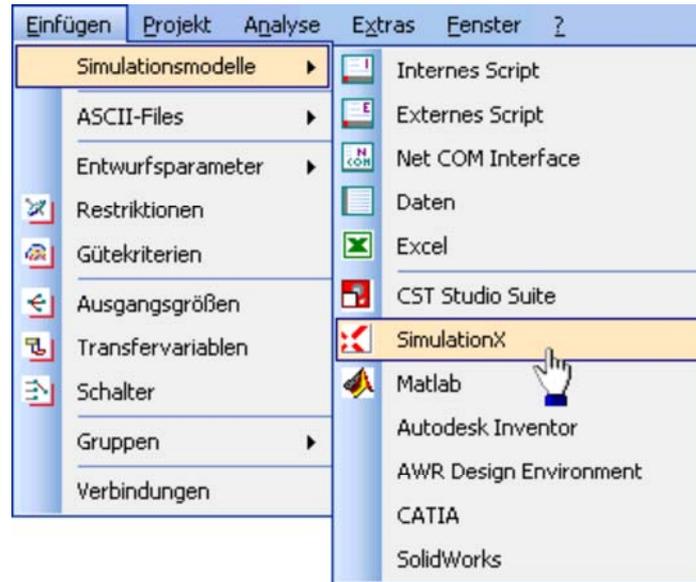


Hinweis: Das Modell **Parameterfindung.ism** im SimulationX sollte geöffnet sein, bevor man dieses Modell in den OptiY-Workflow einbindet.

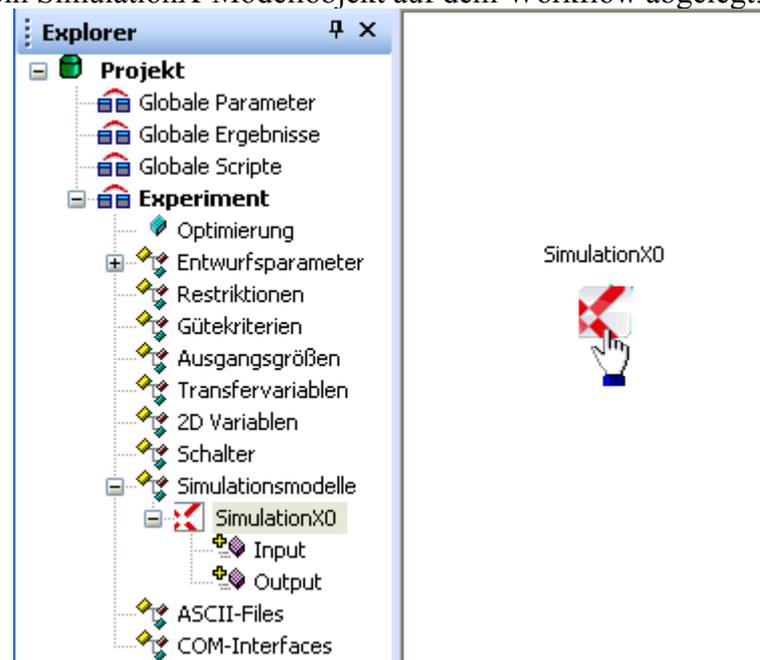
Experiment-Workflow

OptiY stellt über den *Workflow-Editor* Schnittstellen zu unterschiedlichsten Programmsystemen bereit, u.a. zu SimulationX:

Der Menüpunkt **Einfügen > Simulationsmodelle** ermöglicht die Wahl des unterstützten Modell-Typs *SimulationX*:

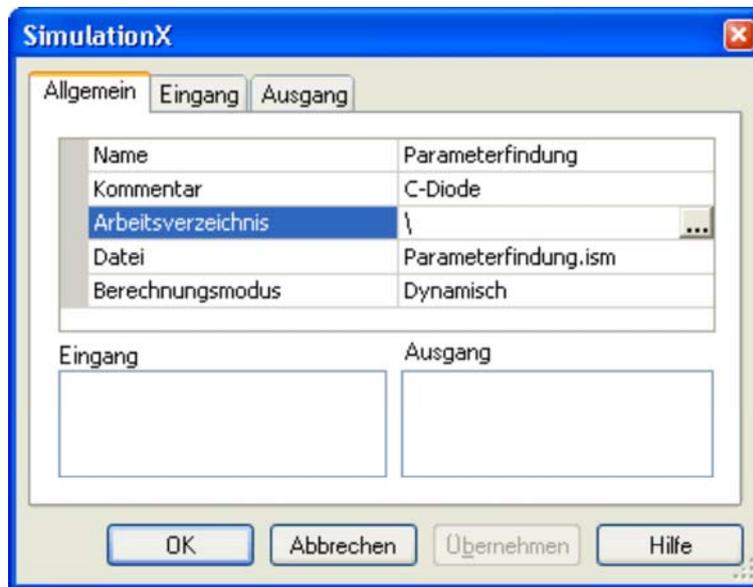


Welches konkrete Modell sich dahinter verbirgt, wird vorläufig nicht spezifiziert, sondern es wird mittels Mausclick vorerst nur ein SimulationX-Modellobjekt auf dem Workflow abgelegt:



Jedes Modellobjekt enthält Input- und Output-Container (Parameter- und Ergebnis-Interface), welche in der Baumstruktur des Experiments erscheinen.

Die Eigenschaften eines Modellobjekts kann man definieren, nachdem man mittels Doppelklick auf das grafische Symbol im Workflow in den entsprechenden Dialog gelangt:



Name und Kommentar sollte man im OptiY-Workflow mit sinnvollen Begriffen versehen.

Wichtig:

Wir ersetzen den Arbeitsverzeichnis-Pfad durch das Zeichen \. Damit wird das Modell im gleichen Pfad erwartet, wie die OptiY-Projektdatei. Die Experimente sind damit zwischen unterschiedlichen Computern portabel!

Das Zuordnen von Eingängen und Ausgängen ist noch nicht möglich, weil auf dem Workflow noch keine Datenobjekte definiert wurden.

Das Modell erscheint unter seinem Namen (hier "Parameterfindung") im OptiY-Explorer.

Konfiguration speichern:

Mittels *Datei* > *Speichern* sollte man den aktuellen Zustand des Projektes sichern.

Dies erfolgt in eine Datei mit der Extension *.opy*, welche im gleichen Ordner zu speichern ist, wie das Simulationsmodell.

Den Dateinamen sollte man im Beispiel möglichst vom Modell übernehmen (*Parameterfindung.opy*). Nur so erkennt man später auch noch die Zuordnung zwischen Simulationsmodell und Optimierungsprojekt.

Ein späteres Öffnen der *.opy*-Datei stellt die gesicherte Konfiguration des Projektes wieder her.

Entwurfparameter (Nennwerte)

Entwurfparameter sind diejenigen Parameter des Modells, an denen das Optimierungsprogramm Veränderungen vornehmen soll, um eine optimale Lösung für die Optimierungsaufgabe zu finden. Wir suchen die **Nennwerte** zweier Entwurfparameter:

Cj0: Sperrschichtkapazität bei $v=0$ V

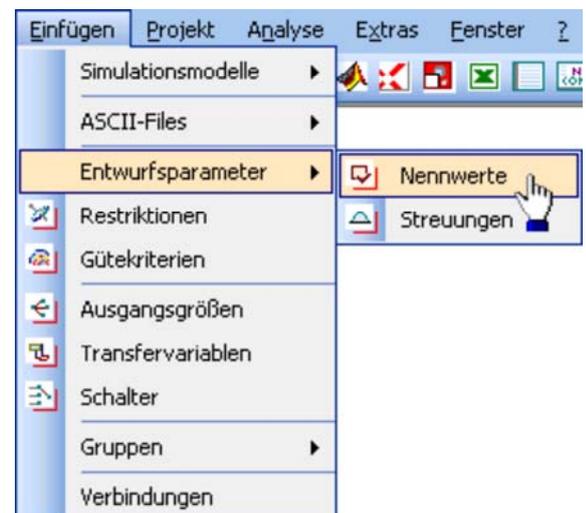
M: Kapazitätskoeffizient

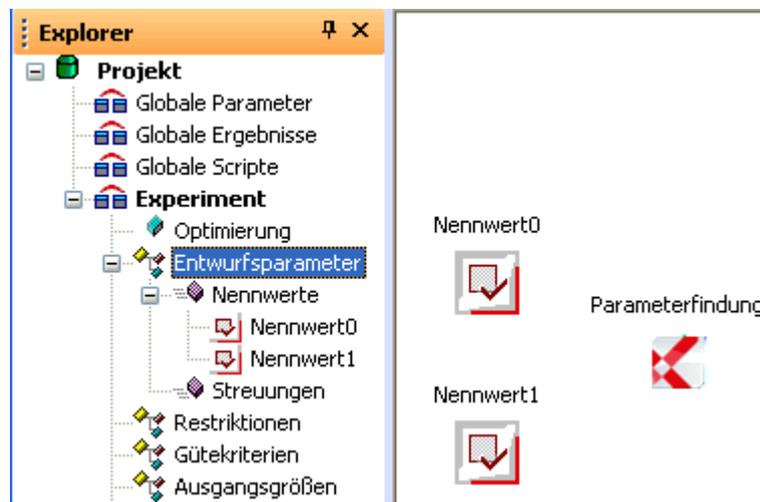
Wir definieren die zwei Entwurfparameter im *Workflow-Editor* durch *Einfügen* > *Entwurfparameter* > *Nennwerte*:

In einem ersten Schritt erzeugt man dafür mit dem Workflow-Editor abstrakte Daten-Objekte. Diese Nennwert-Objekte kann man mittels Mausklick auf dem Workflow-Desktop ablegen:

· Hält man dabei die **Strg**-Taste gedrückt, so kann man mehrere nacheinander ablegen.

· Nach dem Platzieren des letzten Nennwerts beendet man den Vorgang mit der **ESC**-Taste.





Markierte Symbole kann man auf dem Workflow verschieben, so dass eine übersichtliche Anordnung entsteht.

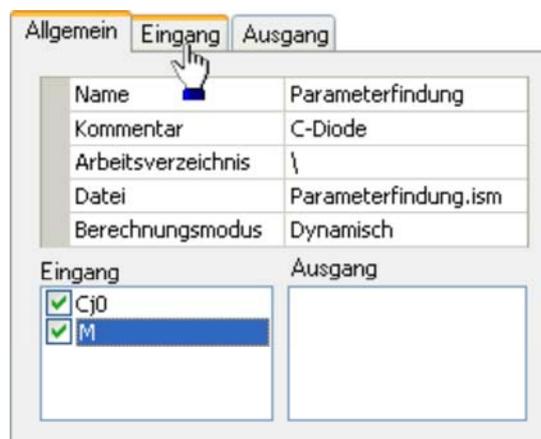
Die Standardbezeichner *Nennwert_x* sollte man durch sinnvolle Namen versehen. Ein zusätzlicher Kommentar erleichtert die spätere Deutung:

- Eigenschaftsdialog über Doppelklick auf Symbol im Workflow-Editor
- oder über das Eigenschaftsfeld des Nennwerts im OptiY-Explorer

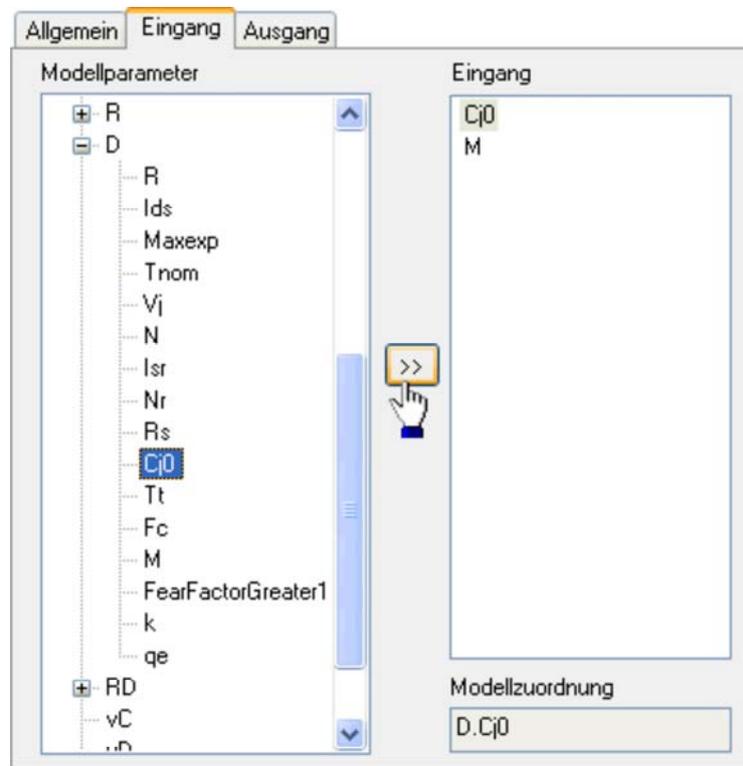


Hinweis: Die Einheiten und Werte der Entwurfparameter werden später automatisch bei der Zuordnung zu den konkreten Parametern des SimulationX-Modells ergänzt!

Die "abstrakten" Entwurfgrößen sollen nun im Workflow den konkreten Modellparametern zugeordnet werden. Nach Doppelklick auf das SimulationX-Objekt erscheint wieder der Dialog zu den Modell-Eigenschaften. Darin markieren wir beide möglichen Eingangparameter und wechseln danach in die Registerkarte **Eingang**:

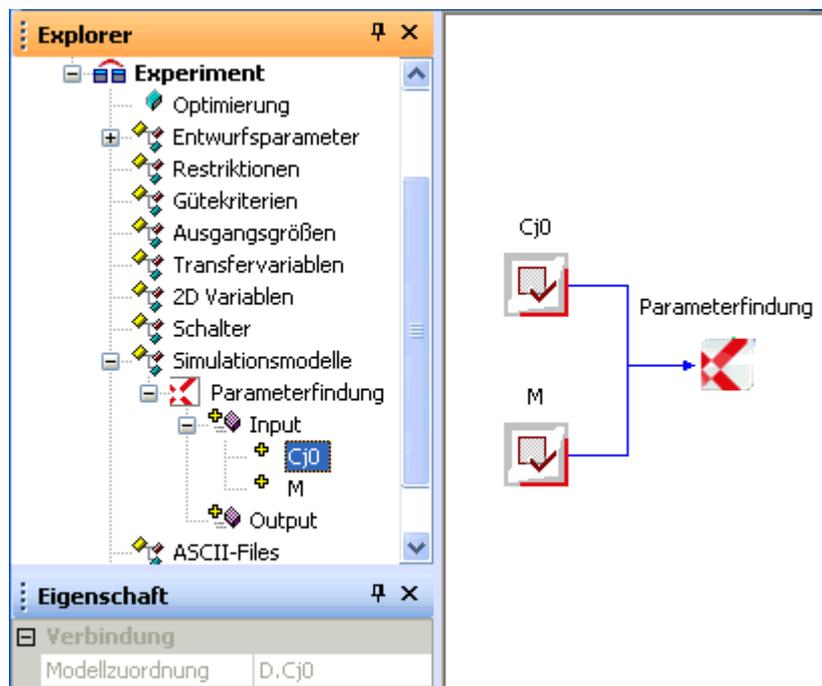


Dort wählt man nacheinander jeweils die zusammengehörenden Paare aus Modell-Parameter und Eingang und stellt die Verbindung durch Betätigen des >>-Button her:

**Achtung:**

Man sollte alle markierten Eingangsgrößen konsequent mit Modellparametern verbinden, bevor man **OK** drückt. Fehlt die Modellzuordnung für eine Verbindung, so führt das später zu "unerklärlichem" Verhalten bei der Optimierung!

Nach Abschluss der Zuordnung erscheinen die Verbindungen im Workflow und können als Eigenschaft der Input-Größen angezeigt werden. Hier sollte man überprüfen, ob alle Zuordnungen korrekt erfolgt sind:



Die Eigenschaften der Entwurfparameter werden auf Basis der aktuellen Modellwerte mit Standard-Annahmen versehen:

Startwert:

Ist der jeweils im Simulationsmodell für die Ausgangslösung eingestellte Wert in der benutzten physikalischen Einheit:

Eigenschaft	
Nennwert Daten	
Name	Cj0
Einheit	pF
Kommentar	Sperrschichtkapazität bei $v=0$
Werte	
Wert	1000
Untergrenze	800
Obergrenze	1200
Genauigkeit	0
Typ	Variable

Grenzwerte:

Die Grenzwerte beschreiben den zulässigen Bereich für die jeweilige Entwurfsgröße.

Unter- und Obergrenze werden von OptiY automatisch um 20% vom Startwert entfernt platziert.

Diese Vorgaben muss man meist durch technisch-physikalisch sinnvolle Werte ersetzen. Die Grenzen sollte man nicht zu eng wählen, damit die optimale Lösung innerhalb des Bereiches liegt. Für die Kapazität **Cj0** kann man auf Grundlage des Datenblattes den möglichen Bereich recht gut abschätzen:

Eigenschaft	
Nennwert Daten	
Name	Cj0
Einheit	pF
Kommentar	Sperrschichtkapazität bei $v=0$
Werte	
Wert	1000
Untergrenze	500
Obergrenze	1500
Genauigkeit	0
Typ	Variable

Für den Kapazitätskoeffizienten **M** fehlt uns noch das Gefühl. Wir übernehmen einfach den vorgegebenen Wert als Startwert, der aber die Krümmung bei weitem nicht nachbildet. Wir nehmen z.B. einen Bereich von **0.1 bis 10** an und hoffen, dass dies ausreicht. Wenn nicht, kann man das später immer noch ändern:

Eigenschaft	
Nennwert Daten	
Name	M
Einheit	-
Kommentar	Kapazitätskoeffizient
Werte	
Wert	0.5
Untergrenze	0.1
Obergrenze	10
Genauigkeit	0
Typ	Variable

Guetekriterium

Als Bewertungsgröße wird in unserem Simulationsmodell die Verhaltensabweichung zwischen der Datenblatt-Kennlinie und der C-Diode als Fehlerquadratsumme berechnet:

Die Abweichung der Spannungsanstiege benutzen wir als Gütekriterium für die Optimierung.

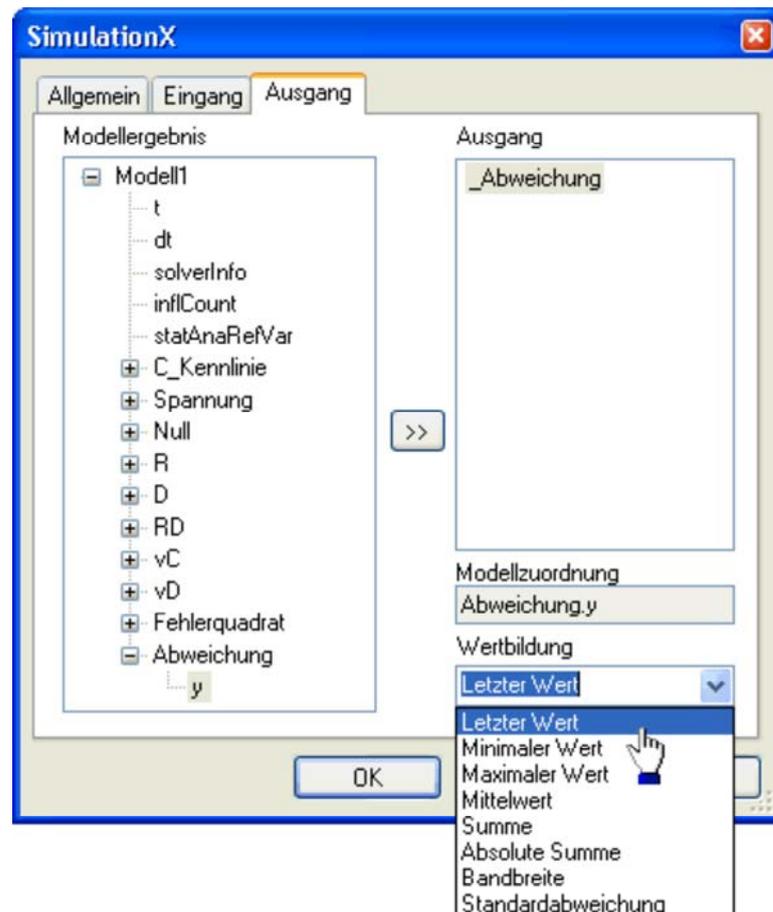
Wir haben die bestmöglichen Parameter für die C-Diode gefunden, wenn die Abweichung nicht mehr verkleinert werden kann.

Wir fügen ein **Gütekriterium** in den Experiment-Workflow ein und nennen es **Abweichung**.

Bewertungsgrößen (hier Gütekriterium) kann man nicht direkt den konkreten Modellgrößen zuordnen. Dafür benötigt man jeweils eine zusätzliche Ausgangsgröße (**Einfügen** > **Ausgangsgrößen**):

Eigenschaft	
Nennwert Daten	
Name	M
Einheit	-
Kommentar	Kapazitätskoeffizient
Werte	
Wert	0.5
Untergrenze	0.1
Obergrenze	10
Genauigkeit	0
Typ	Variable

Da Workflow-Objekte nicht gleiche Namen besitzen sollten, ist es günstig bei den Ausgangsgrößen z.B. in Hinblick auf die zugehörigen Bewertungsgrößen einen Unterstrich voranzustellen. Die Ausgangsgrößen kann man nun in Analogie zu den Entwurfsparametern mit dem Modell-Objekt verknüpfen. Für die Wertbildung einer Ausgangsgröße stehen vielfältige Möglichkeiten zur Verfügung. Wir benötigen hier die Standard-Vorgabe "Letzter Wert", um das Ergebnis des aktuellen Simulationslaufes bewerten zu können:



Die Verknüpfung des Gütekriteriums mit der Ausgangsgröße erfolgt über den Eigenschaftsdialog des Gütekriteriums (Doppelklick auf das Symbol) und Aufruf des Editors:

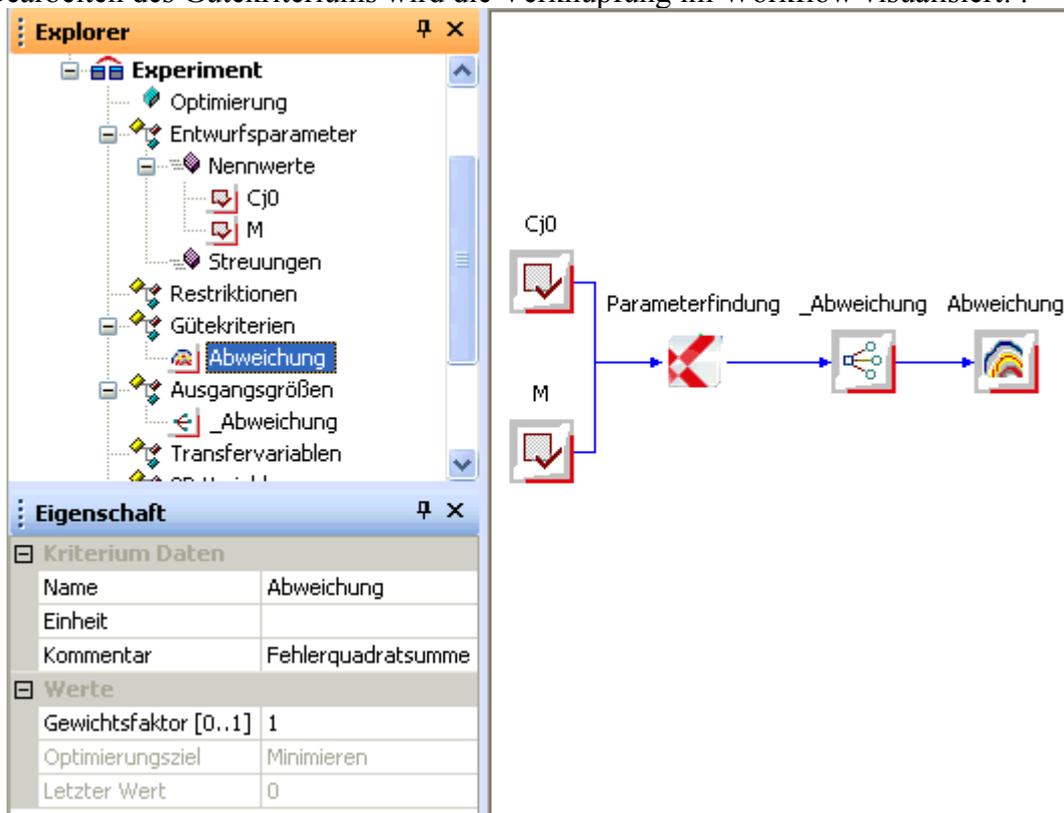


Der Wert einer Bewertungsgröße kann durch mathematische Verknüpfung aller Input- und Output-Größen des Workflows gebildet werden. Diese Verknüpfung wird in Form eines Ausdrucks notiert.

Den Ausdruck kann man jedoch nicht direkt eintragen, sondern man muss ihn mit einem bereitgestellten "Rechner" Editieren:



Nach dem Bearbeiten des Gütekriteriums wird die Verknüpfung im Workflow visualisiert: :



Experiment-Konfiguration

Optimierungsverfahren

Wir haben über den Experiment-Workflow von OptY unser Problem der Parameterfindung komplett in eine Optimierungsaufgabe transformiert. Damit der Computer nun automatisch unser "Problemchen" löst, fehlt nur noch die Konfiguration der Numerik des Optimierungsverfahrens:

OptiY bietet für den unerfahrenen Nutzer ein Standard-Verfahren an. Welches Verfahren sich dahinter verbirgt, ist abhängig von der Anzahl der Entwurfparameter P und Gütekriterien K :

• **Evolutionsstrategie** bei $K > 1$ oder $P > 9$,

• **Hook-Jeeves-Verfahren** für $K < 2$ und $P < 10$

In unserem Beispiel würde also das Hook-Jeeves-Verfahren genutzt. Dieses tastet sich auf der Zielfunktion von der Anfangslösung schrittweise in Richtung kleinerer Werte "bergab" und endet dort in der Talsohle - in der Hoffnung, dass dies das globale Optimum ist:

- Die Option "Automatischer Stop=True" soll die Suche beenden, wenn das Minimum erreicht wurde. Das dafür implementierte Kriterium funktioniert leider nicht immer zuverlässig!
- Um das Optimum zu finden, sollte man "Automatischer Stop=False" setzen. Die Anzahl der Optimierungsschritte kann man auf einen großen Wert (z.B 1000) setzen:

Eigenschaft	
Optimierung	
Automatischer Stop	False
Optimierungsschritte	1000
Startschrittweite	Standard
Verfahren	Standard

Man kann die Optimierung jederzeit manuell stoppen, wenn keine weitere Verbesserung der Lösung zu erwarten ist.

Um an dieser Stelle zumindest einen Einblick in die Konfiguration eines Optimierungsverfahren zu erhalten, wählen wir das Hook-Jeeves-Verfahren:

Eigenschaft	
Optimierung	
Automatischer Stop	False
Optimierungsschritte	1000
Startschrittweite	Standard
Verfahren	Standard
<ul style="list-style-type: none"> Standard Hooke-Jeeves-Verfahren Evolutionäre Algorithmen Adaptive Antwortfläche Raster-Suche Simulation 	

Die individuelle Abtast-Schrittweite für jeden Entwurfparameter ist in der Standardeinstellung u.a. von deren vorgegebenem Variationsbereich [Untergrenze, Obergrenze] abhängig. Da dies numerisch nicht immer günstig sein muss, sollte man sich den Zugriff auf diese Abtast-Schrittweite ermöglichen, indem man *Startschrittweite=manuell* setzt.

Die Startschrittweite erscheint dann als zusätzliche Eigenschaft nach Wahl des jeweiligen Entwurfparameters im OptiY-Explorer (Wert =1/100 des Startwert der Entwurfsgröße). Dieser Vorgabewert ist, wie auch in unserem Fall, meist nutzbar.



Eigenschaft	
Nennwert Daten	
Name	M
Einheit	-
Kommentar	Kapazitätskoeffizient
Werte	
Wert	0.5
Startschrittweite	0.005
Untergrenze	0.1
Obergrenze	10
Genauigkeit	0
Typ	Variable

Visualisierung

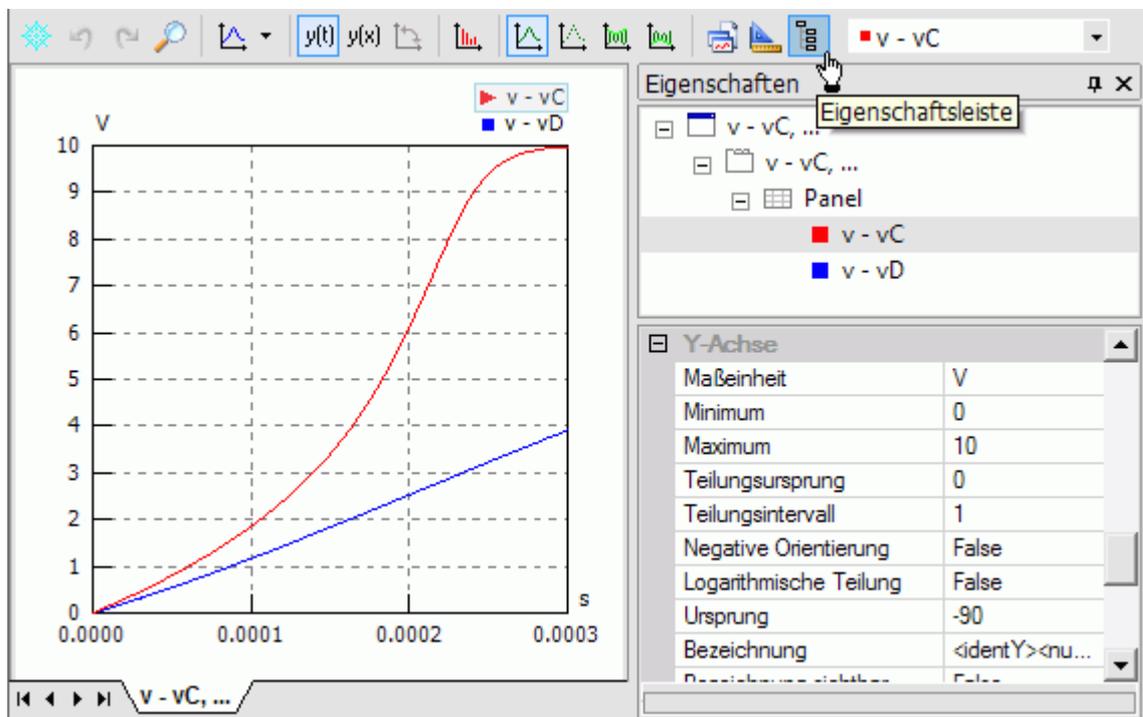
Um das Optimierungsexperiment verfolgen und beurteilen zu können, muss man alle dafür wesentlichen Informationen in geeigneter Form visualisieren. Dazu gehören auch Signalverläufe des Simulationsmodells:

1. SimulationX-Modell

Während des Optimierungsexperiment werden die Entwurfparameter für das SimulationX-Modell in Abhängigkeit vom konfigurierten Optimierungsverfahren verändert.

Im OptiY kann man nur die Output-Größen des Modells als Ergebniswerte des aktuellen Simulationslaufes darstellen. Ob das zeitliche Verhalten des Modells wirklich sinnvoll ist, wird damit nicht unbedingt sichtbar.

Im SimulationX sollte man möglichst ein Ergebnisfenster konfigurieren, welches alle zur Beurteilung des Modellverhaltens relevanten Signalverläufe enthält. Das sind z.B. die beiden Spannungssignale:



Alle nicht mehr benötigten Signalfenster sollte man schließen (wegen der Übersichtlichkeit und dem Zeitbedarf für das Zeichnen!).

Für die zur Experiment-Visualisierung gewählten Signale sollte man mittels der "Eigenschaftsleiste" feste Grenzen für die Koordinatenachsen einstellen, um ein ständiges Flackern der Darstellung zu vermeiden:

Eigenschaften > *Panel* > *Y-Achse* > *Automatische Anpassung=False*

 Für die gemeinsame **Y-Achse** der Signale das **Minimum und Maximum** eintragen (**0..10 V**)

 Eigenschaftsleiste danach wieder schließen

Das Fenster mit den Simulationssignalen ist so auf dem Bildschirm zu platzieren, dass es neben dem OptiY-Fenster sichtbar bleibt.

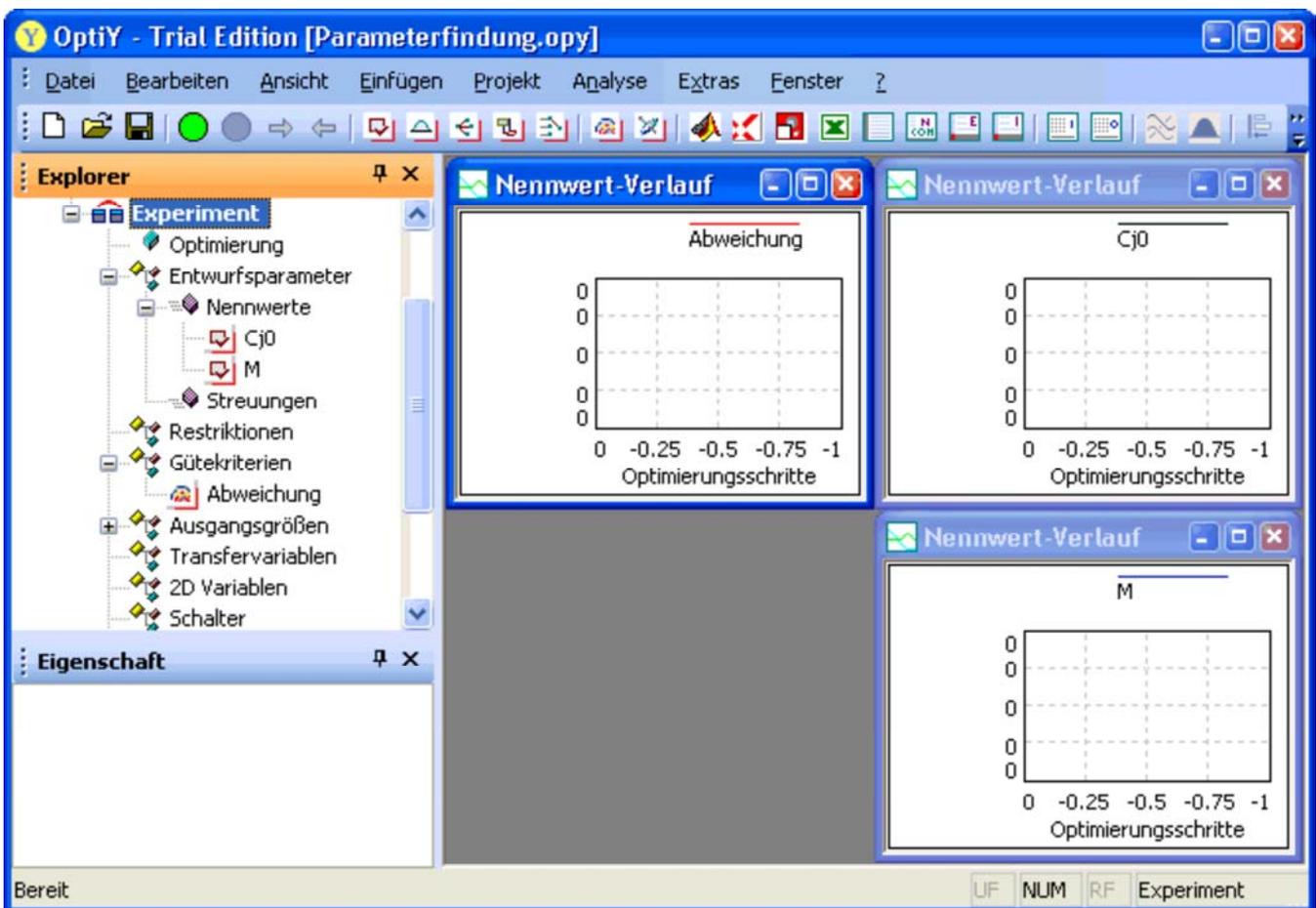
2. OptiY-Experiment

Die Optimierung dauert eine Weile. Deshalb sollte man alle wichtigen Informationen zum Optimierungsfortschritt dabei im Grafik-Bereich darstellen.

Zumindest bei einer überschaubaren Anzahl von Experiment-Größen kann man alle variablen *Entwurfsparameter* und die Bewertungsgrößen (*Restriktionen* und *Gütekriterien*) per *Drag & Drop* vom Explorer in den OptiY-Grafik-Bereich ziehen.

Fenster > **Anordnen** ordnet alle Fenster im Grafik-Bereich nebeneinander an (auch das Workflow-Fenster).

Bei drei Signalfenstern ist es z.B. günstig, das Workflow-Fenster erst nach dem Anordnen zu schließen:



3. Konfiguration speichern

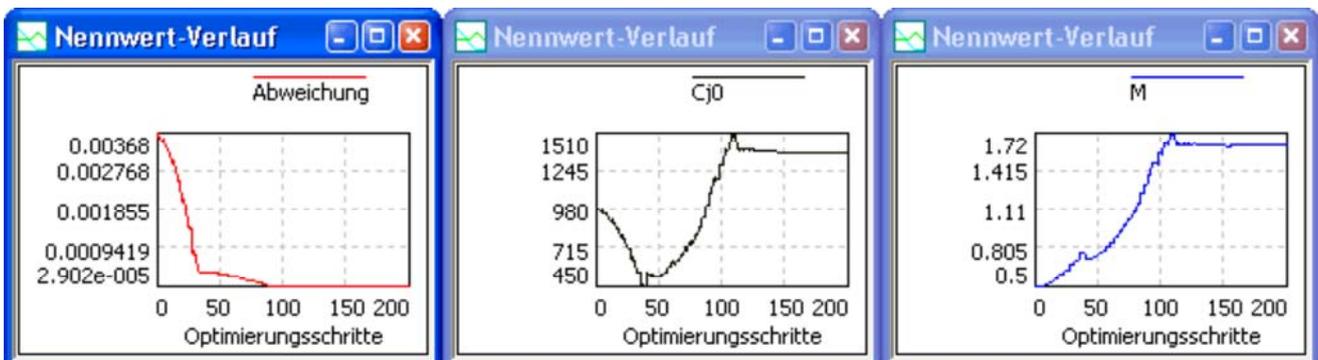
Spätestens jetzt sollte man die aktuelle Konfiguration des OptiY-Versuchsstandes mal wieder als **.opy**-Datei speichern!

Experiment-Durchfuehrung

Lokale Suche

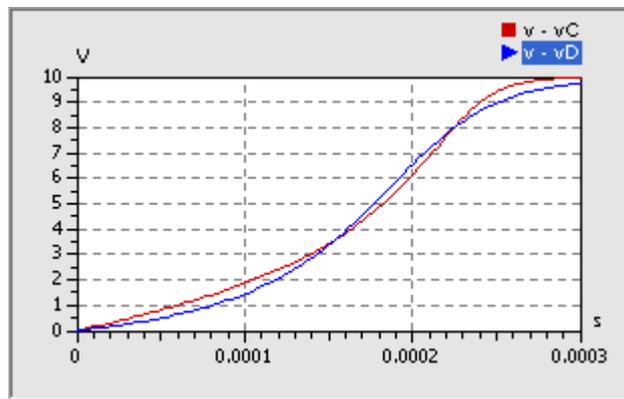
Nach soviel Vorbereitung wollen wir nun beherzt den Start-Button  für das Experiment drücken und hoffen, dass die optimalen Dioden-Parameter gefunden werden:

Wenn keine Verbesserung der Ergebnisse mehr zu erwarten ist, können wir das Experiment stoppen :



Das Optimum wird relativ schnell gefunden. Wir können die Anzahl der Optimierungsschritte deshalb auf 200 verringern.

Das Verhalten der C-Diode hat sich im Vergleich zur Anfangslösung schon wesentlich verbessert:



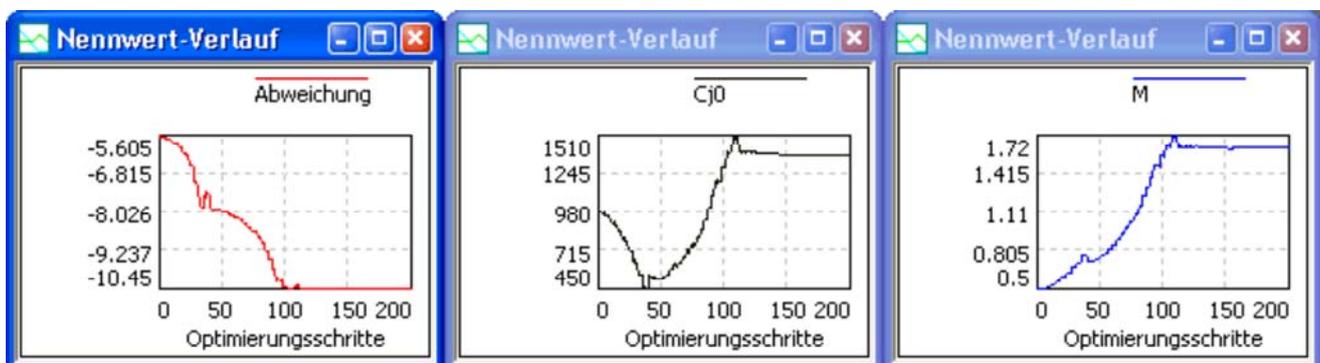
Da die Spannungssignale noch gewisse Abweichungen zeigen, werden wir untersuchen, ob noch weitere Verbesserungen möglich sind. Dazu betrachten wir zuerst den Optimierungsverlauf:

Ungünstig bei der Optimierung ist es, wenn Entwurfparameter an Grenzen stoßen. Dadurch wird die Suche an dieser Stelle gestört, weil der zulässige Bereich der Entwurfparameter nicht verlassen werden darf. In unserem Beispiel betrifft das die Grenzen von C_{j0} . Physikalisch-technisch sind kleinere Werte als 500 pF und größere als 1500 pF zulässig. Wir ändern die Grenzwerte auf **100 pF** und **2000 pF**.

In der Nähe der optimalen Lösung erkennt man optisch keine Veränderung des Gütewertes, obwohl weiterhin eine Veränderung der Entwurfparameter erfolgt. Für Gütekriterien, welche gegen Null streben, ist der Verwendung einer Logarithmusfunktion sinnvoll. Wir benutzen den natürlichen Logarithmus (**ln**):



Für das Optimierungsergebnis ist im Beispiel keine Verbesserung zu spüren, weil die vorherige Kollision mit den Grenzwerten anscheinend den Optimierungsverlauf kaum beeinflusst haben:



Die Änderung des Gütewertes ist infolge der Logarithmusfunktion wesentlich besser zu verfolgen. Am Ergebnis der Optimierung hat sich jedoch noch nichts verbessert!

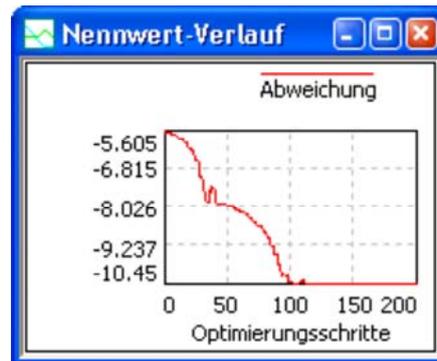
Das Hooke-Jeeves-Verfahren konvergiert durch ständige Abwärtsbewegung auf der Gütefunktion zum nächstgelegenen Minimum:

Es handelt sich zumindest um ein lokales Minimum auf der Gütefunktion.

Den erreichten Bestwert kann man sich anzeigen lassen (**Analyse > Bestwert > Parameter anzeigen**):

Name	Werte	Einheit	Kommentar
Cj0	1381.41	pF	Sperrschichtkapazität bei $v=0$
M	1.62719	-	Kapazitätskoeffizient
Abweichung	-10.4476		Fehlerquadratsumme

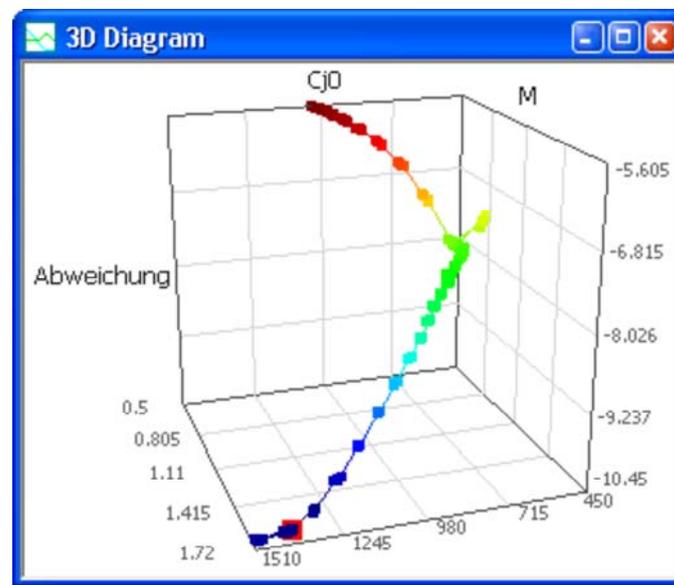
Man kann sich dabei jedoch nie sicher sein, ob nicht doch noch Kombinationen für die Entwurfsparameter existieren, welche zu einer besseren Lösung führen (möglichst zum globalen Minimum der Gütefunktion). Es wäre günstig, wenn man die Topografie der Gütefunktion in Analogie zu einer Landkarte kennen würde. Die Gütefunktion wird in unserem Beispiel nur durch das Gütekriterium **Abweichung** gebildet. Während der Optimierung wird als Nennwert-Verlauf praktisch ein Höhenprofil des auf der Oberfläche der Gütefunktion zurückgelegten Pfades abgebildet:



Der Wert der Abweichung ist eine Funktion zweier Entwurfsparameter (**Cj0 und M**). Man kann den Suchpfad im durch diese beiden Entwurfsparameter aufgespannten Suchraum in einem 3D-Diagramm abbilden (*Analyse > Darstellung > 3D-Darstellung*). Wir belegen die X- und Y-Achse mit den Entwurfsparametern und die Z-Achse mit dem Gütekriterium. Ist dieses 3D-Diagramm aktiv, so kann man die Eigenschaften der 3D-Darstellung konfigurieren:

Eigenschaft	
3D Darstellung	
Rahmen	True
Punkte	True
Linien	True
Fläche	False
Pareto Anzahl	10
Auto-Skalierung	True

Die Position des Bestwertes im 3D-Diagramm wurde markiert, indem in den Nennwertverläufen ein Wert zum Ende der Optimierung durch Doppelklick ausgewählt wurde:



Globale Suche

OptiY bietet die Möglichkeit, für ein Projekt mehrere Experimente zu verwalten. So können wir die bisherigen Einstellungen und Ergebnisse beibehalten und als Ausgangspunkt für ein neues Experiment nutzen:

- Nach dem **Duplizieren** des Experiments erscheint im OptiY-Explorer die Kopie als **Experiment2**.
- Dieses weitere Experiment besitzt am Anfang die gleiche Konfiguration wie das Original. Nur die Anzeigefenster muss man neu definieren.
- Verwaltet man mehrere Experimente in einem Versuchsstand, so sollte man mittels **Umbenennen** dafür sinnvolle Namen vergeben:
 - Das bisherige Experiment könnte man z.B. nach dem verwendeten Optimierungsverfahren **Hooke-Jeeves** nennen.
 - Für das neue Experiment wählen wir die Bezeichnung **Rastersuche 2D**:

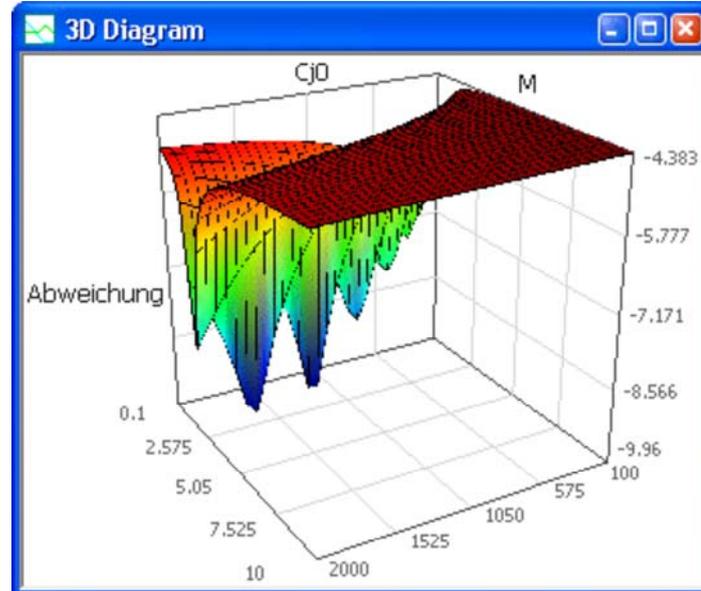


Leider sieht man bei der lokalen Suche nur die Punkte der Gütefunktion, welche durch das Verfahren abgetastet werden. Wie die komplette Gütefunktion im Suchraum aussieht, bleibt uns dabei verborgen:

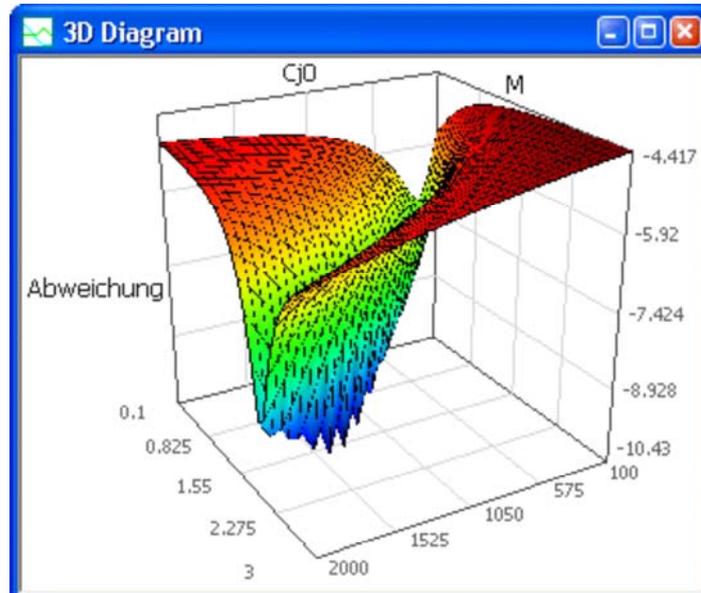
- Es ist bei praktischen Problemstellungen meist nicht möglich, die komplette Gütefunktion z.B. durch hinreichend dichte Stützstellen zu erfassen. Insbesondere bei mehreren Entwurfsgrößen und langsam rechnenden Modellen würde man Tage oder sogar Jahre für die Berechnung benötigen.
- Unser Simulationsmodell rechnet zum Glück sehr schnell (Rechenzeit ca. 0.1 s) und wir müssen nur zwei Entwurfsgrößen berücksichtigen.
- Wir benutzen als Optimierungsverfahren die Raster-Suche.
- Mit 900 Optimierungsschritten ergibt sich ein Raster von 30x30 Abtastpunkten.
- Ein 3D-Diagramm ermöglicht die grafische Abbildung der Gütefunktion (**Analyse > Darstellung > 3D-Darstellung**).
- Wir belegen die X- und Y-Achse wieder mit den Entwurfsparametern und die Z-Achse mit dem Gütekriterium.
- Ist dieses 3D-Diagramm aktiv, so kann man die Eigenschaften der 3D-Darstellung als Fläche konfigurieren:

Eigenschaft	
3D Darstellung	
Rahmen	True
Punkte	False
Linien	True
Fläche	True
Pareto Anzahl	10
Auto-Skalierung	True

Nach der etwas zeitintensiven Berechnung aller Stützstellen sollte sich folgende Gütefunktion ergeben:

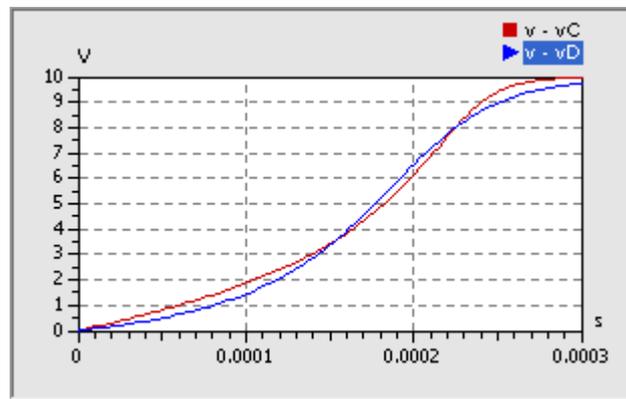


Hinweis: Falls der Flächenteil mit großen M-Werten nicht richtig berechnet wird, kann man dieses numerische Problem meist durch eine Verringerung von $dt_{Min}=1e-20$ s in der Simulationssteuerung lösen. Man erkennt mehrere lokale Minima in einem engen Graben. Trotz der dichten Rasterung kann man Darstell-Effekte bei der Raster-Suche nie ausschließen. Da der obere Grenzwert für den Parameter **M** wesentlich zu groß gewählt wurde, führen wir eine erneute Rastersuche mit einem verringerten Grenzwert von **3** durch. Damit wird die Senke mit dem Optimum dichter abgetastet:



Mit großer Wahrscheinlichkeit existiert ein globales Optimum. Die Vielzahl lokaler Minima wird in der Grafik nur durch die Rasterung der Abtastung vorgetäuscht.

Der Bestwert aus dem Hooke-Jeeves-Verfahren entspricht also wirklich dem globalen Optimalwert:



Schlussfolgerungen:

1. Mit der in der Diode implementierten Funktion für die Sperrschicht-Kapazität ist es nicht möglich, die Kennlinie des Datenblattes exakt nachzubilden. Das ist bei derartigen Modellen jedoch normal!
2. Das resultierende Verhalten der Sperrschichtkapazität wird jedoch qualitativ recht gut nachgebildet.
3. Bei der Nutzung des Diodenmodells mit den identifizierten Parametern muss man sich bewusst sein, dass es gewisse Abweichungen zum realen Verhalten gibt. Die Größe des Fehlers muss man bei Bedarf untersuchen.

Frage 8:

Wie groß ist die differentielle Sperrschicht-Kapazität der C-Diode bei Verwendung der identifizierten Parameter bei einer Spannung von 10 V? Um wieviel Prozent weicht dieser Wert von dem der Datenblatt-Kennlinie ab?

Hinweise:

Die obige Formel für die Berechnung der Sperrschicht-Kapazität der Diode ist im SimulationX-Modell in einem separates Function-Element zu berechnen. Der Zugriff auf die Parameter des Dioden-Elements erfolgt in der Formel über die Bezeichner (z.B. **D.M** oder **D.cj0**). In *Hilfe > Inhalt > Modelica in SimulationX > Operatoren und Ausdrücke* findet man eine Beschreibung zur Syntax.

← →

Von „http://www.optiyummy.de/index.php?title=Software:_SimX_-_Einfuehrung_-_Elektro-Chaos_-_Parameter-Identifikation“

Software: SimX - Einfuehrung - Elektro-Chaos - Schwingkreis mit C-Diode

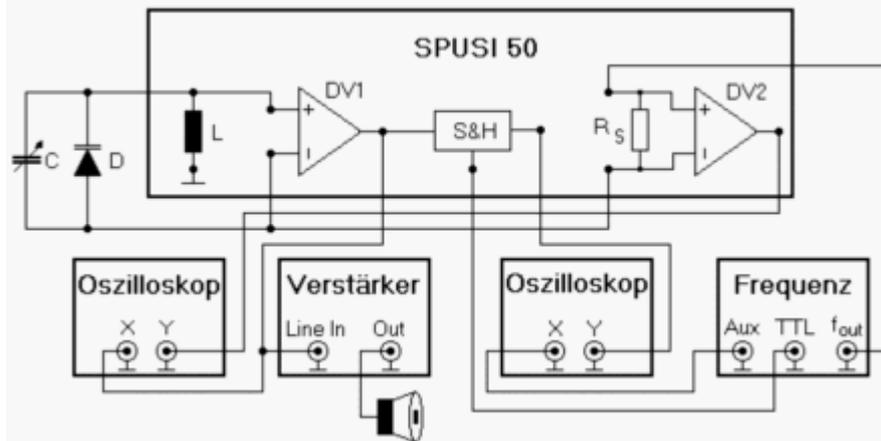
Aus OptiYummy

↑

← →

Nichtlinearer Schwingkreis mit C-Diode (Chaos-Experimente)

Nach einigen theoretischen Betrachtungen soll versucht werden, reale physikalische Chaos-Experimente mittels der numerischen Simulation nachzuvollziehen. Dabei benutzen wir die Veröffentlichung **Experimente zur Untersuchung linearer und nichtlinearer elektrischer Serienschwingkreise** von Burkhard Wolf und Hans-Josef Patt, welche als **PDF-Datei** abrufbar ist. Die Kapazitätsdiode wird hier als Bestandteil eines Serienschwingkreises benutzt. Das folgende Bild aus der PDF-Datei zeigt die Original-Schaltung:



Inhaltsverzeichnis

- . 1 Nichtlineare Systeme
- . 2 Chaotische Systeme
- . 3 Der Phasenraum eines Reihenschwingkreises
 - . 3.1 Linearer Schwingkreis
 - . 3.2 Fremderregter linearer Schwingkreis
 - . 3.3 Fremderregter nichtlinearer Schwingkreis
- . 4 Frequenzspektrum des nichtlinearen Schwingkreises
 - . 4.1 Frequenzanalyse
 - . 4.2 Konfiguration der Anregung
 - . 4.3 Einfluss der Anregungsamplitude
- . 5 Bifurkation
 - . 5.1 Bifurkationsdiagramm
 - . 5.2 Der Weg zum perfekten Chaos

Nichtlineare Systeme

Alle Systeme, welche nicht als lineare Systeme behandelt werden können, sind nichtlineare Systeme:

- . Nichtlineare Systeme enthalten Elemente, deren Parameterwerte abhängig von der Belastung des Elements sind (z.B. Kapazität als Funktion der anliegenden Spannung) oder erhalten "zufällige" Eingangssignale (z.B. zeitabhängige Frequenzänderung der Betriebsspannung).
- . Für nichtlineare Systeme existieren (im Normalfall) keine analytischen Lösungen.
- . Nichtlinearität ist in Natur und Technik der Regelfall. Die linearen Näherungen der klassischen Natur- und Technikwissenschaften beschreiben nur die grundlegenden Effekte (kausalen Zusammenhänge) der jeweiligen physikalisch-technischen Domäne (E-Technik, Mechanik, Fluidtechnik, usw.). Sie vermitteln damit das grundlegende Verständnis, welches man auch zur Modellierung nichtlinearer Systeme anwenden kann.
- . Das Verhalten nichtlinearer Systeme kann der "gesunde Menschenverstand" nicht immer vorhersagen.

Ähnliche Startbedingungen führen häufig zu stark verändertem Verhalten.

Chaotische Systeme

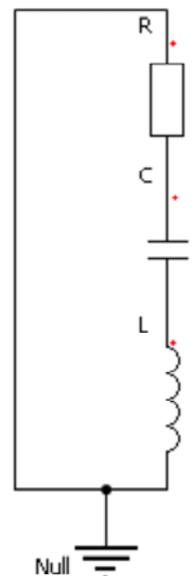
Damit ein nichtlineares System chaotisches Verhalten zeigt, müssen zusätzlich zur "Nichtlinearität" weitere Kriterien erfüllt sein:

- . Es muss mindestens eine Stelle starker sensitiver Abhängigkeit von den Anfangsbedingungen existieren. Umgangssprachlich ist dies als Schmetterlingseffekt bekannt. Kleinste Änderungen der Startparameter führen langfristig zu völlig anderem Verhalten. Man spricht hierbei von deterministischem Chaos.
- . Das System muss hinreichend viele Freiheitsgrade für die Bewegung im sogenannten Phasenraum besitzen:
 - . Der Zustand eines Systems wird zu jedem Zeitpunkt durch einen Punkt in einem Raum dargestellt werden, dessen Koordinatenachsen durch den Satz der unabhängigen Zustandsgrößen (meist Integralgrößen) des Systems gegeben sind. Die Dynamik des Systems lässt sich damit als Bahn (Trajektorie) dieses Punktes im Phasenraum interpretieren. So wird beispielsweise der Phasenraum eines LCR-Schwingkreises durch den Spulenstrom und die Kondensatorspannung aufgespannt. Die ungedämpfte Schwingung entspricht einer geschlossenen Kurve um den Koordinatenursprung.
 - . Die Trajektorie eines Systems kann sich innerhalb des Phasenraums nicht schneiden, weil dies der Eindeutigkeit der Systementwicklung ausgehend von einem Punkt im Phasenraum widersprechen würde. In der Ebene kann es deshalb nur zu spiralähnlichen Kurven kommen. Als Spezialfall ist für ungedämpft oszillierende Systeme die geschlossene Trajektorie möglich.
 - . Erst durch einen zusätzlichen Freiheitsgrad ist es im Beispiel des LCR-Schwingkreises möglich, dass von der Spiralform abweichende Trajektorien entstehen, weil "Schnittpunkte" durch Ausweichen in der 3. Koordinatenachse vermieden werden können. Der zusätzliche Freiheitsgrad wird durch die äußere Anregung des Schwingkreises mit einer zeitveränderlicher Spannung geschaffen.
- . Eine maßvolle Dämpfung muss für eine dosierte Energieabfuhr sorgen, damit das System weder "explodiert" noch "erstarrt".

Der Phasenraum eines Reihenschwingkreises

Ein LCR-Schwingkreis besitzt zwei Energiespeicher mit jeweils einer energetischen Zustandsgröße (Spulenstrom und Kondensatorspannung):

- . Der Wert einer energetischen Zustandsgröße beschreibt eindeutig die aktuell im zugehörigen Speicher enthaltene Energiemenge.
- . Ausgehend von einem gegebenen Ladezustand der Energiespeicher entwickelt sich das System determiniert im Zeitbereich, weil zwischen den Speichern ein Energiefluss über die vorhandenen Verbindungselemente erfolgt.
- . Die Entwicklung eines Systems in der Zeit wird durch die aktuellen Werte aller Zustandsgrößen eindeutig beschrieben.
- . Sehr anschaulich sieht man die Systementwicklung durch die Trajektorie des Systemzustands im Phasenraum.
- . Beim schrittweisen Aufbau des Chaos-Experiments werden wir den Phasenraum detailliert betrachten.



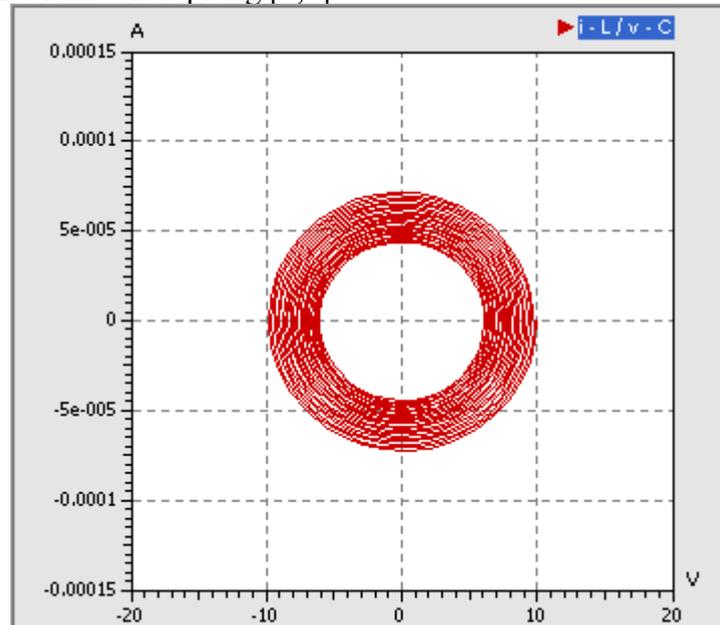
Linearer Schwingkreis

Wir beginnen mit dem Aufbau des Serienschwingkreises (Bild rechts) aus der durch einen Gyrotor simulierten Spule (0,1...50 H), einem Drehkondensator (10..530 pF) und einem Serienwiderstand (50 Ω oder 1 k Ω). Laut Experimentieranleitung verwenden wir zuerst folgende Werte für die Bauelemente:

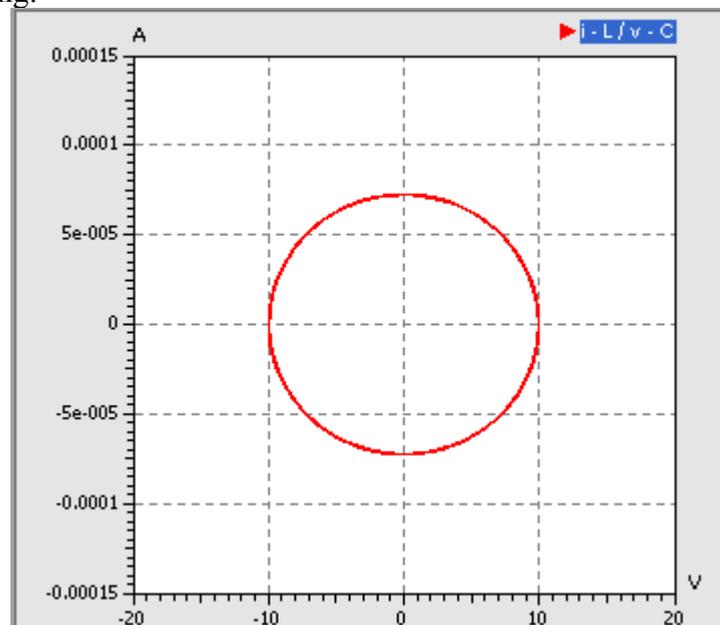
- . **C = 530 pF**
- . **L = 10 H**
- . **R = 1000 Ω**

Wir beobachten mehrere Schwingungen (z.B. **tStop=20 ms**) ausgehend von einem auf **10 V** aufgeladenen Kondensator:

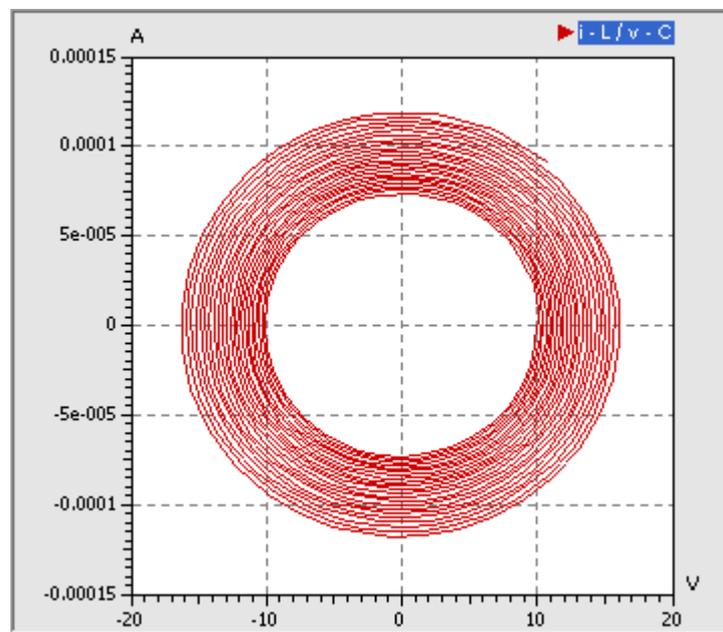
Im Phasenraum widerspiegelt sich die durch den Serienwiderstand gedämpfte Schwingung als Spiral-Trajektorie in Richtung Koordinatenursprung $[0,0]$:



Verringert man den Widerstand auf Null, so ergibt sich der Spezialfall der geschlossenen Trajektorie einer ungedämpften Schwingung:



Realisiert man den Serienwiderstand durch eine aktive Schaltung, so lassen sich auch negative ohmsche Widerstände realisieren. So können wir mit $R=-1000 \Omega$ die letzte mögliche Form der Trajektorie im zweidimensionalen Phasenraum beobachten - die sich ins Unendliche entwickelnde Spirale:



Für die weiteren Untersuchungen setzen wir den Serienwiderstand auf $R=50 \Omega$. Damit ergibt sich eine nur leicht gedämpfte Schwingung.

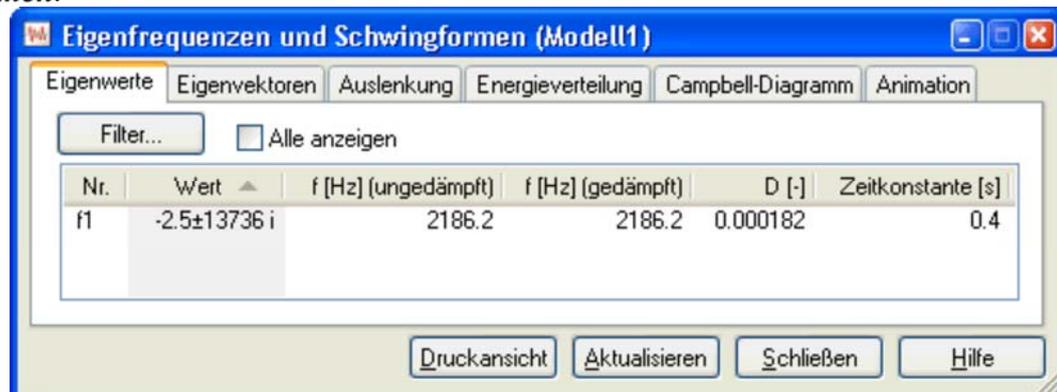
Merke:

Die Trajektorie darf sich nicht kreuzen, ansonsten gäbe es ausgehend von einem Zustand mehr als eine weitere Entwicklungsmöglichkeit. Das ist in einem deterministischen System nicht möglich!

Fremderregter linearer Schwingkreis

Bevor wir einen Sinusgenerator an den Schwingkreis ankoppeln, um diesen in der Nähe seiner Resonanzfrequenz anzuregen, wollen wir die Schwingfrequenz unseres Kreises überprüfen:

Dazu muss man im SimulationX keine zusätzliche Signalverarbeitung in das Modell einbinden, wie wir dies zuvor zu Übungszwecken durchführten. SimulationX enthält ein Tool zur **Analyse > Eigenfrequenzen und Schwingformen**:



Die angezeigte Frequenz von 2186,2 Hz steht im Widerspruch zu den ungefähr 1400 Hz, welche sich laut Experimentieranleitung bei dieser Konfiguration ergeben sollen!

Wenn wir für die eingestellten Parameter die Resonanzfrequenz mit der Thomsonschen Schwingungsgleichung berechnen, werden sich 2186,16 Hz ergeben:

$$f_0 = \frac{1}{2\pi\sqrt{LC}}$$

Was ist hier schiefgelaufen?

Die Genauigkeit eines Luftdrehkondensators wird durch die Toleranzen des geometrischen Aufbaus bestimmt. Damit dürfte die Kapazitätstoleranz im Bereich weniger Prozent liegen. Damit ist die große Abweichung der Resonanzfrequenz nicht erklärbar.

Der Induktivitätswert wird mittels einer Gyrotorschaltung aus einem Kapazitätswert gewonnen, wobei auch

noch die aktuellen Werte von ohmschen Widerständen mit einfließen. Hier können größere Fehler entstehen!

Frage 9:

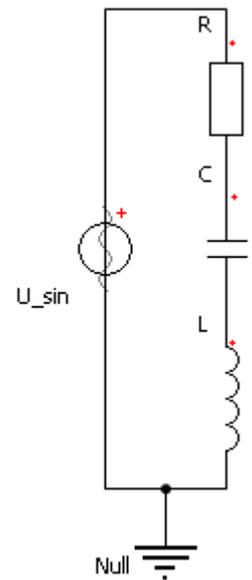
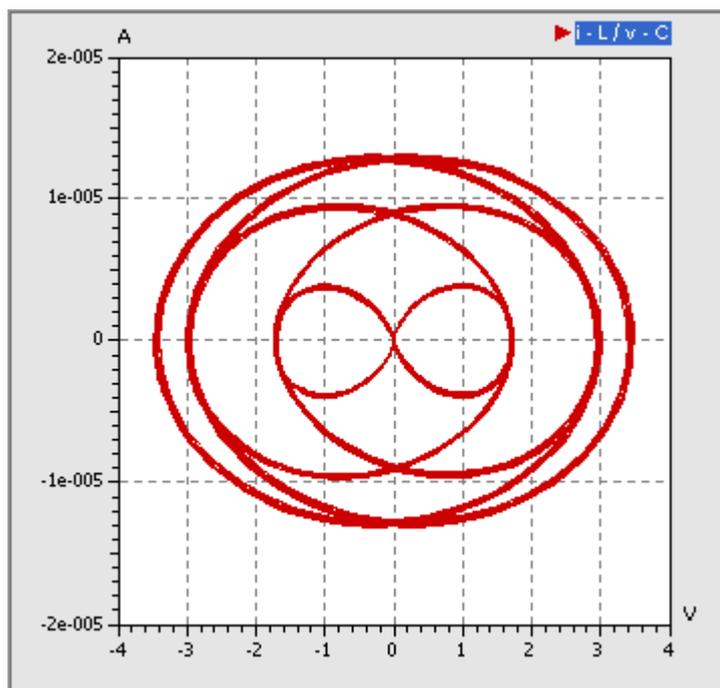
Wie groß ist die erforderliche Induktivität von L für den vorgegeben Wert $C=530 \text{ pF}$, damit eine Resonanzfrequenz $f_0=1400 \text{ Hz}$ entsteht. Dieser Wert ist mittels der Analyse der Eigenfrequenzen zu überprüfen.

Den ermittelten Induktivitätswert behalten wir für die weiteren Experimente bei:

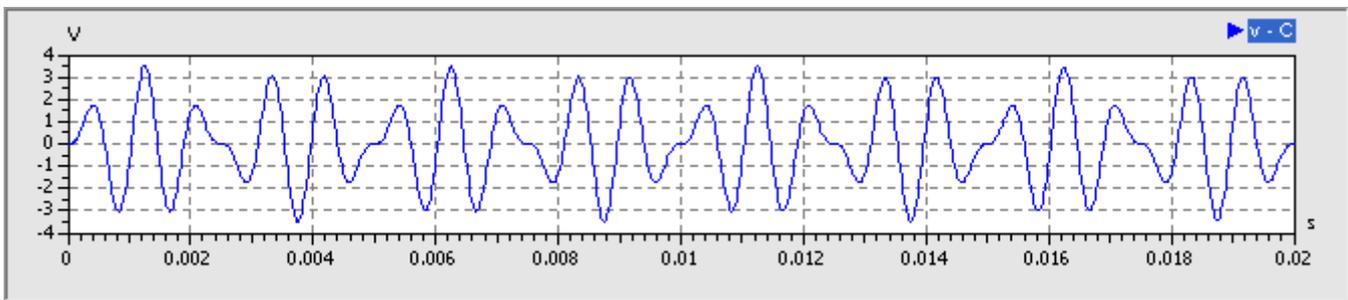
- Man kann davon ausgehen, dass im realen Experiment auch dieser Wert verwendet wurde, da sowohl die Frequenzmessung, als auch der Kapazitätswert des Drehkondensator nur mit geringen Toleranzen behaftet sind.
- Wahrscheinlich wurden am Zeiger der Gyrator-Induktivität 10 H gewählt und diesem Wert wurde ohne analytische Überprüfung vertraut!

Würden wir den Schwingkreis exakt mit der Resonanzfrequenz anregen, so gäbe es auf Grund der hohen Kreisgüte eine sehr große Resonanzüberhöhung:

- Den Kondensator versehen wir wieder mit der Anfangsspannung $v_0=0 \text{ V}$.
- Wir ergänzen eine **sinusförmige Spannungsquelle** aus der Modellbibliothek und benutzen eine Spannungsamplitude $U_{\text{sin}}=1 \text{ V}$.
- Um das Umschlagen der Systemqualität anhand der Trajektorie im Phasenraum zu verdeutlichen, wählen wir eine Erregerfrequenz weit unterhalb der Resonanzfrequenz ($U_{\text{sin}}.f=1000 \text{ Hz}$):



- Infolge der Fremderregung wird der Zustand des Gesamtsystems jetzt auch durch den aktuellen Wert der Betriebsspannung bestimmt. Es gibt also eine dritte Zustandsgröße, deren Koordinatenachse man sich im obigen Diagramm in der Tiefe vorstellen kann.
- Auch jetzt gibt es keine Kreuzungspunkte der Trajektorie. Wir sehen nur Kreuzungspunkte in der 2D-Projektion des Phasenraums, in Wirklichkeit verläuft die Trajektorie dort in unterschiedlichen Tiefen!
- Der zeitliche Verlauf der Kondensatorspannung ist nun nicht mehr sinusförmig, zeigt jedoch eine gewisse Periodizität:



Diese Periodizität widerspiegelt sich auch in den Schleifen der Trajektorie im Phasenraum.

Fremderregter nichtlinearer Schwingkreis

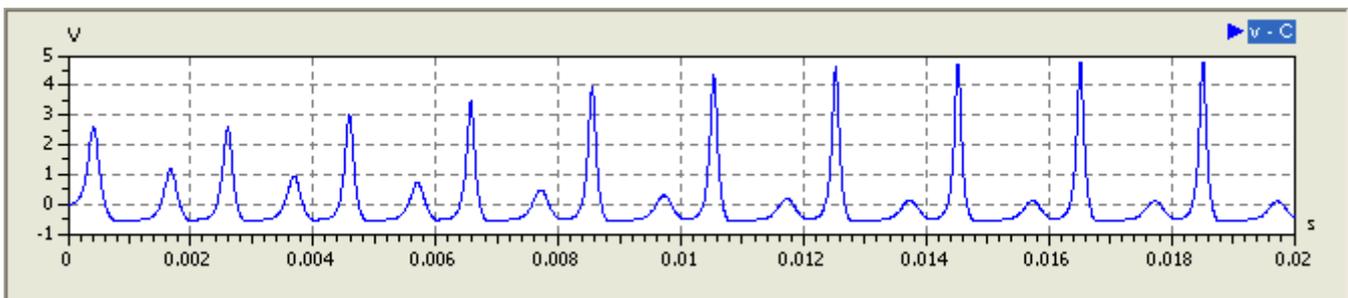
Im nächsten Schritt ergänzen wir laut Anleitung die Diode **D** parallel zum Drehkondensator:

Den Drehkondensator setzen wir auf seinen kleinsten Kapazitätswert **C=10 pF**.

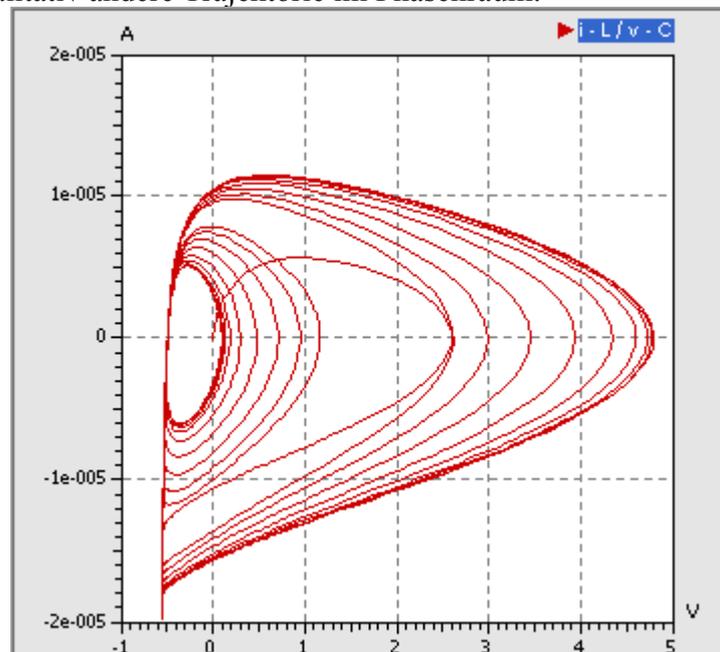
Die Diode erhält die von uns ermittelten Parameter **cj0** und **M**, welche die Kapazitätskennlinie des Diodentyps **BB 512** möglichst gut nachbilden.

Wir verändern die Sinusspannung der Erregung vorläufig nicht (**1 V** bei **1000 Hz**).

Wie zu erwarten war, wird infolge des Gleichrichter-Effekts der Diode eine Halbwelle auf ca. 0,5 V begrenzt. Im Beispiel betrifft die die negative Halbwelle:



Daraus resultiert eine qualitativ andere Trajektorie im Phasenraum:



Wir werden später noch sehen, dass die Trajektorie in Abhängigkeit von der Frequenz und der Amplitude der Erregerspannung unterschiedliche Muster bildet. Diese Muster widerspiegeln die Amplitudencharakteristik des zeitlichen Verlaufs der Diodenspannung.

Frequenzspektrum des nichtlinearen Schwingkreises

Frequenzanalyse

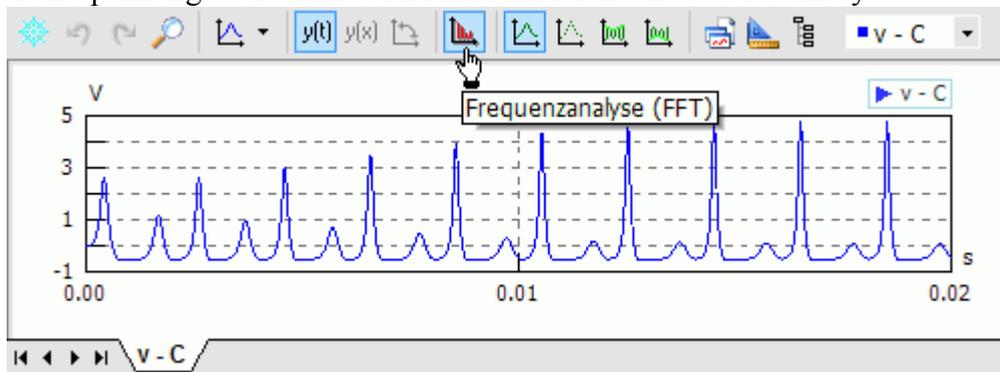
Solange ein Schwingkreis ohne Fremderregung eine nur leichtgedämpfte Sinusschwingung ausführt, besteht das Frequenzspektrum nur aus seiner Resonanzfrequenz. Diese Frequenz konnten wir mittels **Analyse > Eigenfrequenzen und Schwingformen** ablesen. Dieses Tool ist jedoch sinnvoll nur für lineare Systeme nutzbar, da es sich um eine "Lineare Systemanalyse" handelt. Bei nichtlinearen Systemen könnte man nur die Schwingfrequenzen für den aktuellen Zustand sehen, für den das Modell linearisiert wird.

SimulationX bietet jedoch in den Signalfenstern die **Analyse > Frequenzanalyse (FFT)** :

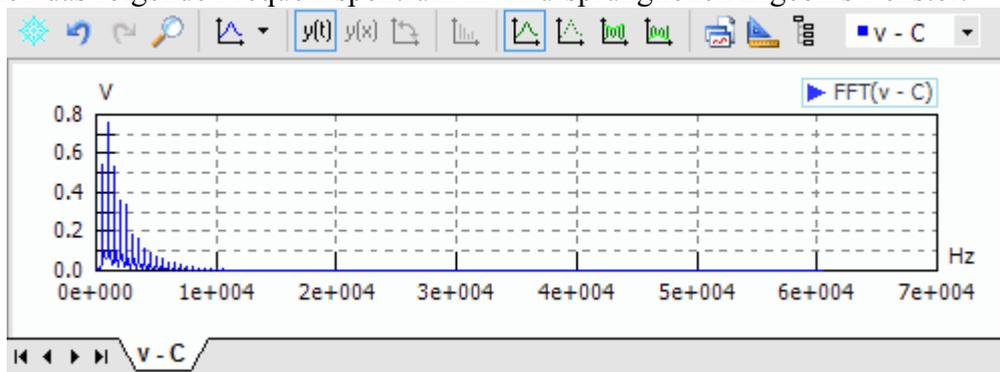
- FFT ist die Abkürzung für *Fast Fourier Transform* (Schnelle Fourier-Transformation).
- Ein beliebiges zeitliches Signal lässt sich immer als Summe von Sinusschwingungen unterschiedlicher Frequenz und Amplitude hinreichend genau abbilden (Reihenentwicklung der Fourier-Analyse).

Die FFT setzt eine konstante Zeitschrittweite für das zu analysierende Signal voraus:

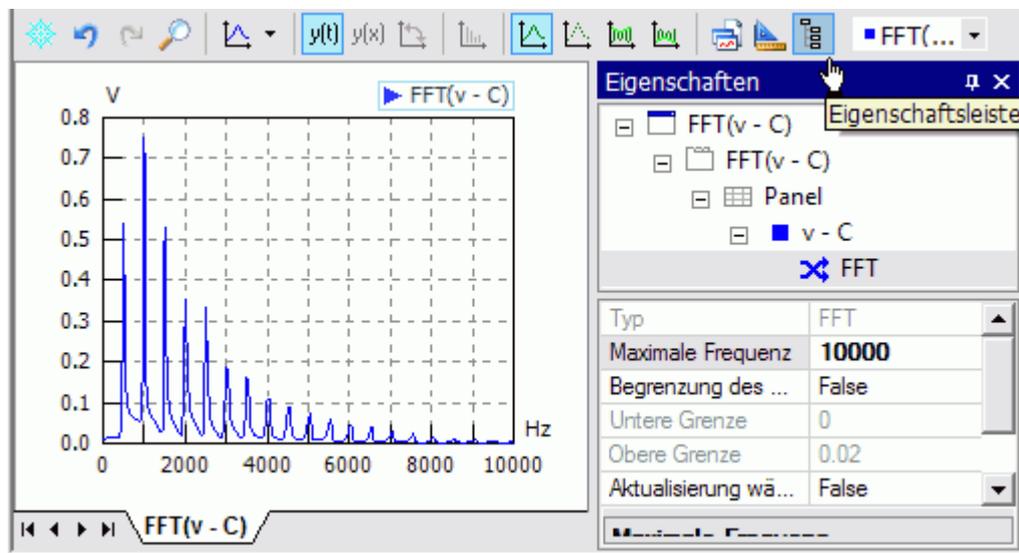
- Wir arbeiten bei der Aufzeichnung der Signale mit variablen Schrittweiten.
- Vor der FFT-Analyse erfolgt im SimulationX eine automatische Überführung in ein Signal mit konstanter Zeitschrittweite.
- Die Wahl dieser Schrittweite erfolgt automatisch auf Grundlage der kleinsten Aufzeichnungsschrittweite.
- Damit wird gleichzeitig der obere Grenzwert für das zu berechnende Spektrum festgelegt.
- Für den simulierten Spannungsverlauf an der Diode aktivieren wir die FFT-Analyse:



- Damit ergibt sich das folgende Frequenzspektrum in im ursprünglichen Ergebnis-Fenster:



- Wird ein zu breites Spektrum berechnet, so begrenzen wir über die Eigenschaftsleiste diese Berechnung des Spektrums auf 10000 Hz, was der 10-fachen Erregerfrequenz entspricht:



Man erkennt die Erregerfrequenz von 1000 Hz als Maximalanteil im Signal, gefolgt von der halben Frequenz.

Alle Frequenzanteile sind Vielfache von 500 Hz. Der Anteil höherer Frequenzen im Spektrum sinkt stetig, das Spektrum endet praktisch bei ca. 9 kHz.

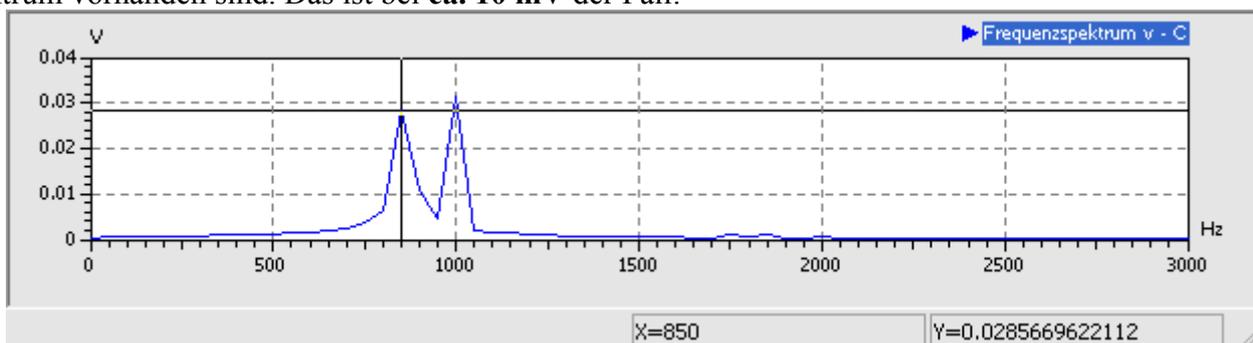
Die Abtastschrittweite des Signalverlaufs muss hinreichend klein sein, um auch hohe Frequenzen noch erfassen zu können. Laut Abtasttheorem muss die Abtastfrequenz mindestens doppelt so hoch sein, wie die zu erfassende Frequenz. Das Abtasttheorem wird vom SimulationX bei der automatischen Ermittlung einer maximal möglichen Frequenz berücksichtigt.

Für Spektrum-Anteile von 10 kHz benötigt man einen Signalwert mindestens aller 50 μ s, was jedoch bereits zu sehr eckigen Signalverläufen führen würde.

Konfiguration der Anregung

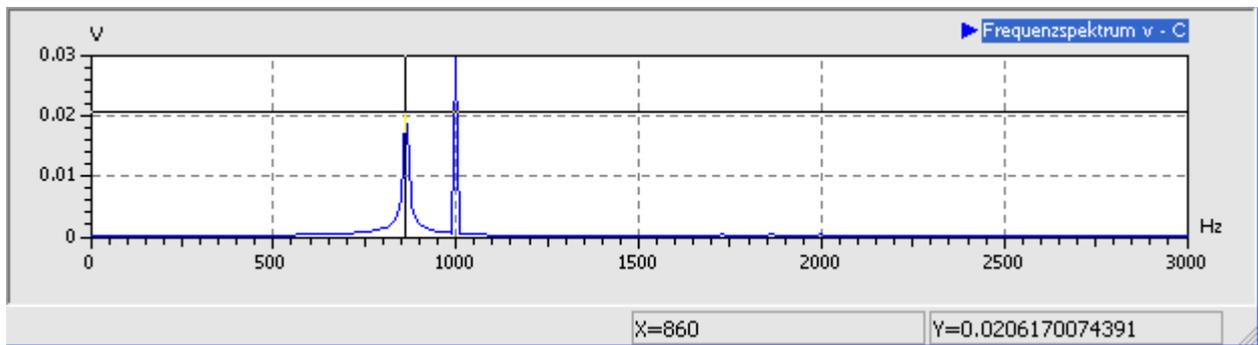
Die Vielzahl von Frequenzanteilen resultiert aus dem Kappen einer Halbwelle, wenn die Durchlassspannung der "Gleichrichter"-Diode überschritten wird. Es entstehen "Knicke" im Spannungsverlauf, welche hohen Frequenzanteilen entsprechen.

Wir verringern nun die Amplitude der Erregerspannung soweit, dass keine höherfrequenten Frequenzanteile mehr im Spektrum vorhanden sind. Das ist bei **ca. 10 mV** der Fall:

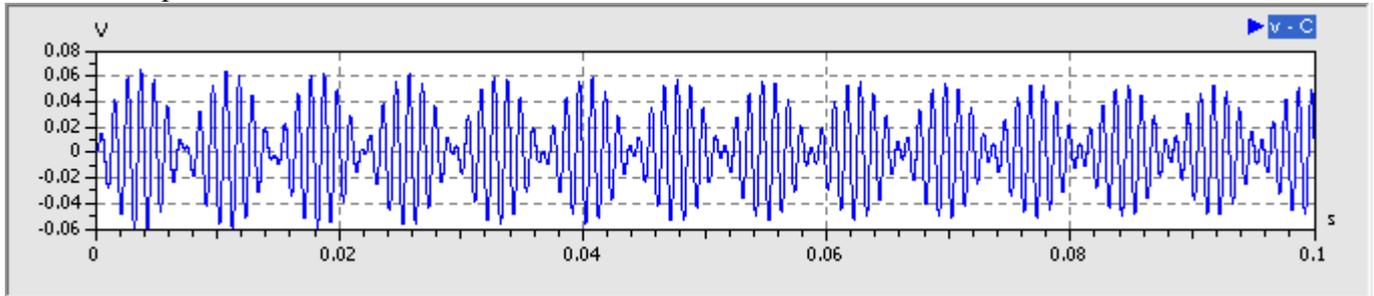


Zusätzlich zur Erregerfrequenz von 1 kHz erscheint die Resonanzfrequenz von ca. 850 Hz. Diese ergibt sich aus der Sperrschicht-Kapazität, die sich bei sehr kleinen Sperrspannungen einstellt und welche ungefähr **Cj0** entspricht:

Das Frequenzspektrum wird genauer berechnet, wenn man mehr Perioden der Schwingung simuliert. Mit **tStop=0.1 s** ergibt sich z.B. eine wesentlich feinere Auflösung, erkennbar an schmalen Frequenz-Peaks. Die Resonanzfrequenz wird jetzt im Beispiel mit 860 Hz ermittelt



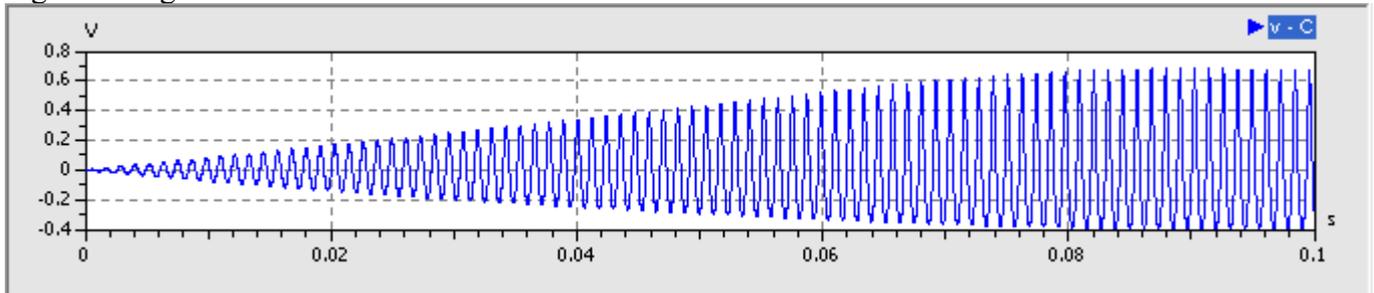
Der zugehörige Zeitverlauf der Diodenspannung zeigt die Schwebung von 140 Hz zwischen Erreger- und Resonanzfrequenz:



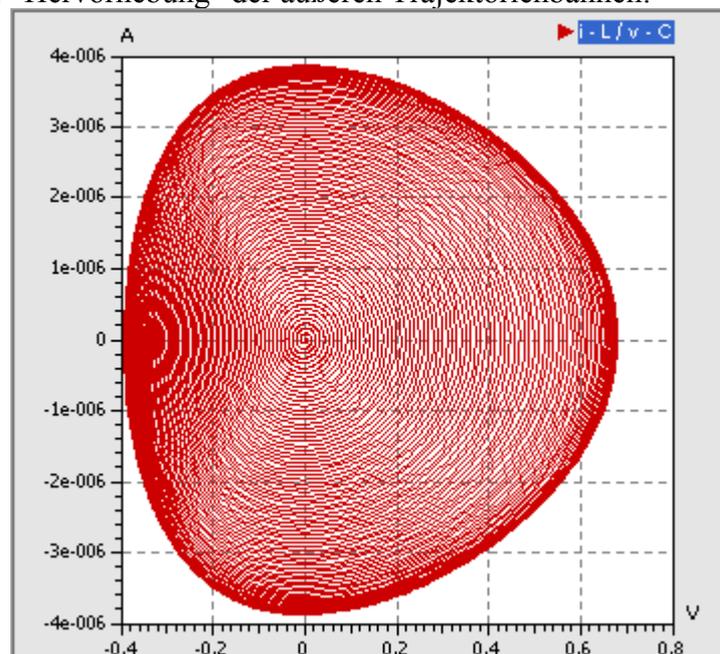
Wir erregen im Folgenden den Schwingkreis mit der zuletzt ermittelten Resonanzfrequenz von z.B. **860 Hz**:

Nun kommt es zu einem Aufschaukeln der Schwingung, bis infolge der vorhandenen Dämpfung in Reihenwiderstand und Diode ein Gleichgewicht erreicht ist (Resonanz-Überhöhung).

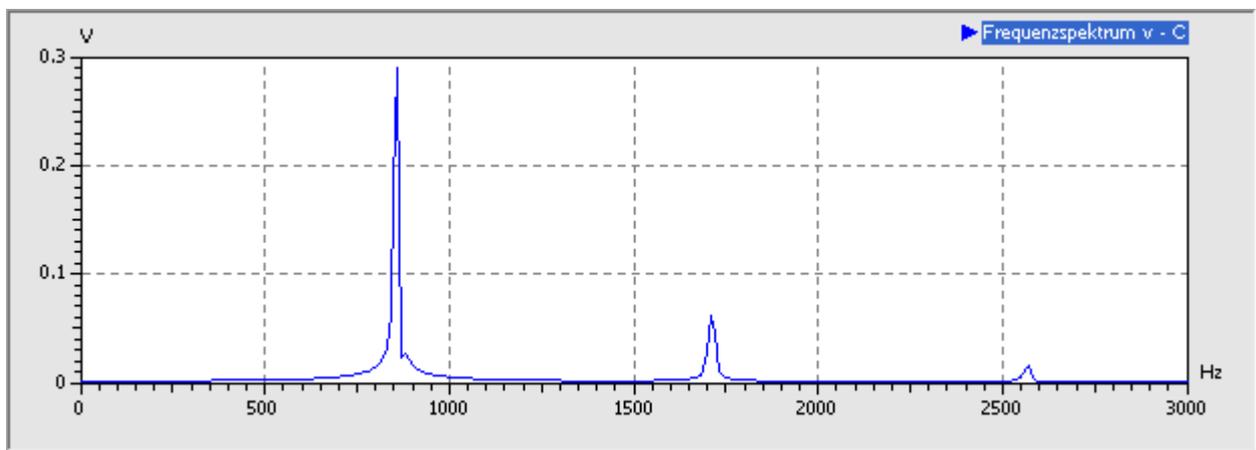
Wir verringern die Erregungsamplitude soweit, dass innerhalb von 100 ms ein "sanfter" Übergang in den eingeschwingenen Zustand stattfindet:



Die Konvergenz zu einem Endzustand gleicher Schwingungsamplitude widerspiegelt sich im 2D-Phasendiagramm in einer "Hervorhebung" der äußeren Trajektorienbahnen:



Das Frequenzspektrum zeigt neben der Erregerfrequenz, welche nun der Resonanzfrequenz entspricht, kleine Amplituden der 1. und 2. Oberwelle:



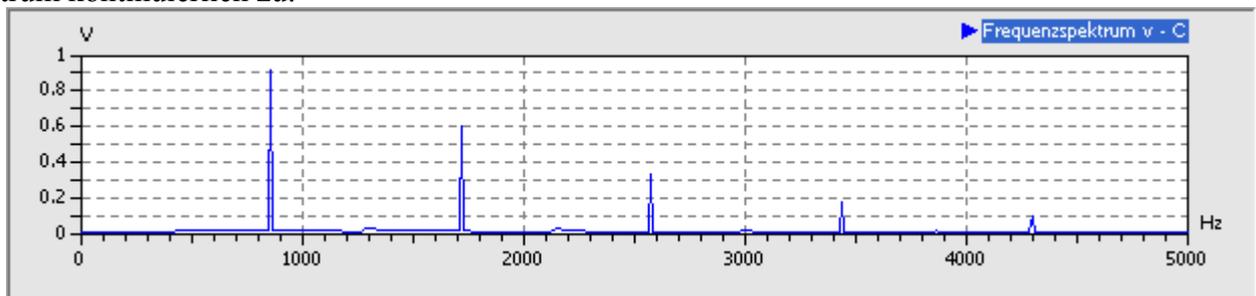
Die Oberwellen existieren, weil infolge der Amplitudenbegrenzung die Schwingung leicht von der Sinusform abweicht. Nur der reine Sinus besitzt keine Oberwellen.

Wenn man sich die Grundfrequenz im Spektrogramm genauer anschaut, erkennt man einen kleinen Peak wenige Hz oberhalb der Erregerfrequenz. Durch die erhöhte Spannungsamplitude an der Diode verringerte sich deren Sperrschicht-Kapazität und damit verschob sich die Resonanzfrequenz etwas nach oben.

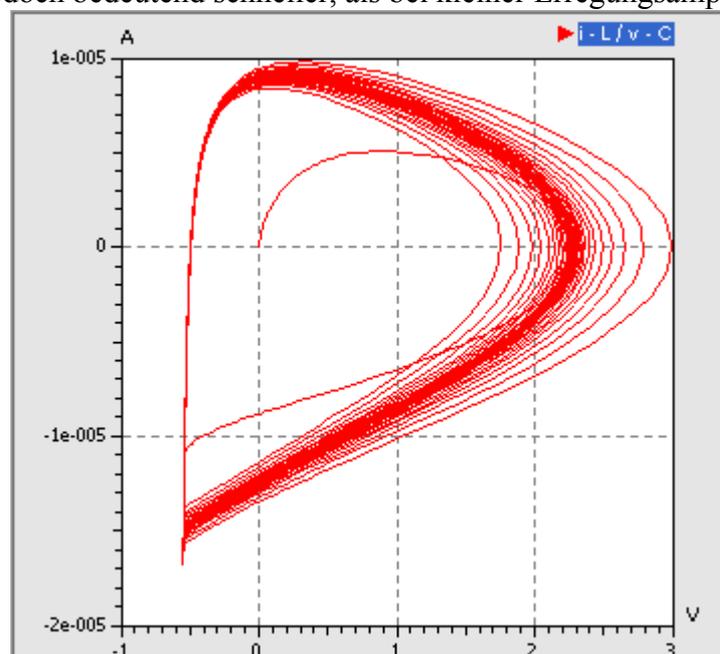
Einfluss der Anregungsamplitude

Wenn wir im Folgenden die Amplitude der Sinus-Erregung in kleinen Schritten vorsichtig erhöhen, so kommt der Gleichrichtungseffekt der Diode zur Wirkung. Das Kappen einer Halbwelle beginnt, wenn sich im Schwingkreis die Amplitude an der Diode auf über 0,5 V erhöht:

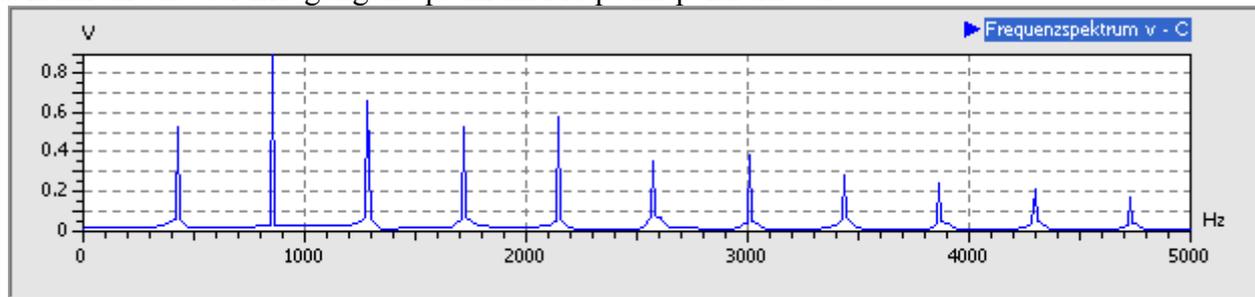
Das Verhalten des Schwingkreises ändert sich qualitativ nicht bis zu einer bestimmten Erreger-Amplitude. Im Beispiel lag dieser Wert bei ungefähr 1 V. Bis dahin nimmt nur die Amplitude der Oberwellen im Spektrum kontinuierlich zu:



Die Trajektorie konvergiert im 2D-Phasendiagramm weiterhin zu einer Trasse (Attraktor), weil sich die Amplitude der Schwingungen auf einen Endwert einschwingt. Das Einschwingen auf die Maximal-Amplitude erfolgt jetzt jedoch bedeutend schneller, als bei kleiner Erregungsamplitude:

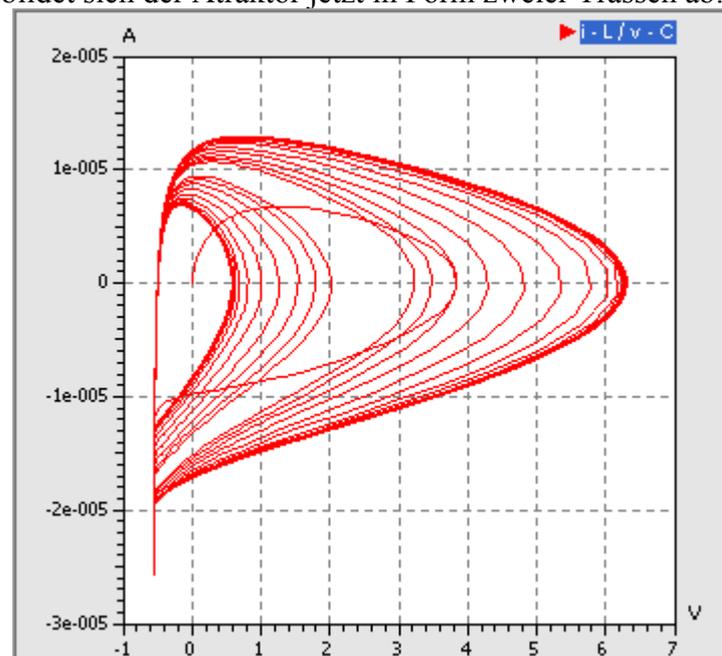


Eine neue Qualität des Verhaltens entsteht, indem bei einer weiteren Erhöhung der Erregungsamplitude plötzlich eine Perioden-Verdopplung stattfindet. Im Beispiel war dies bei 1.3 V Erregungsamplitude der Fall. Es tritt dann die 1. Subharmonische der Erregungsfrequenz im Frequenzspektrum auf:

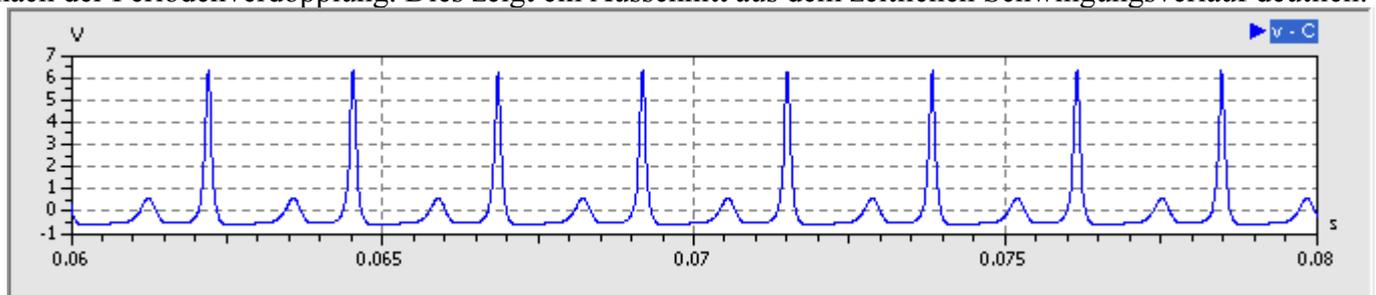


Von dieser 1. Subharmonischen leiten sich zusätzliche Obertöne im Frequenzspektrum ab. Das Spektrum erstreckt sich bis über 10 kHz (im Bild beschnitten).

Im 2D-Phasendiagramm bildet sich der Attraktor jetzt in Form zweier Trassen ab:



Ursache für die 2 Attraktor-Trassen ist die unterschiedliche Schwingungsamplitude der beiden Perioden nach der Periodenverdopplung. Dies zeigt ein Ausschnitt aus dem zeitlichen Schwingungsverlauf deutlich:



Bifurkation

Nichtlineare Systeme, deren Verhalten von einem Parameter (hier der Anregungsamplitude) abhängt, können bei einer Änderung des Parameters ihr Verhalten plötzlich ändern. Wie im Beispiel kann ein System, das zuvor einem Grenzwert zustrebte, nun zwischen zwei Werten hin und her springen, also zwei Häufungspunkte aufweisen. Dies nennt man eine Bifurkation.

Bifurkationsdiagramm

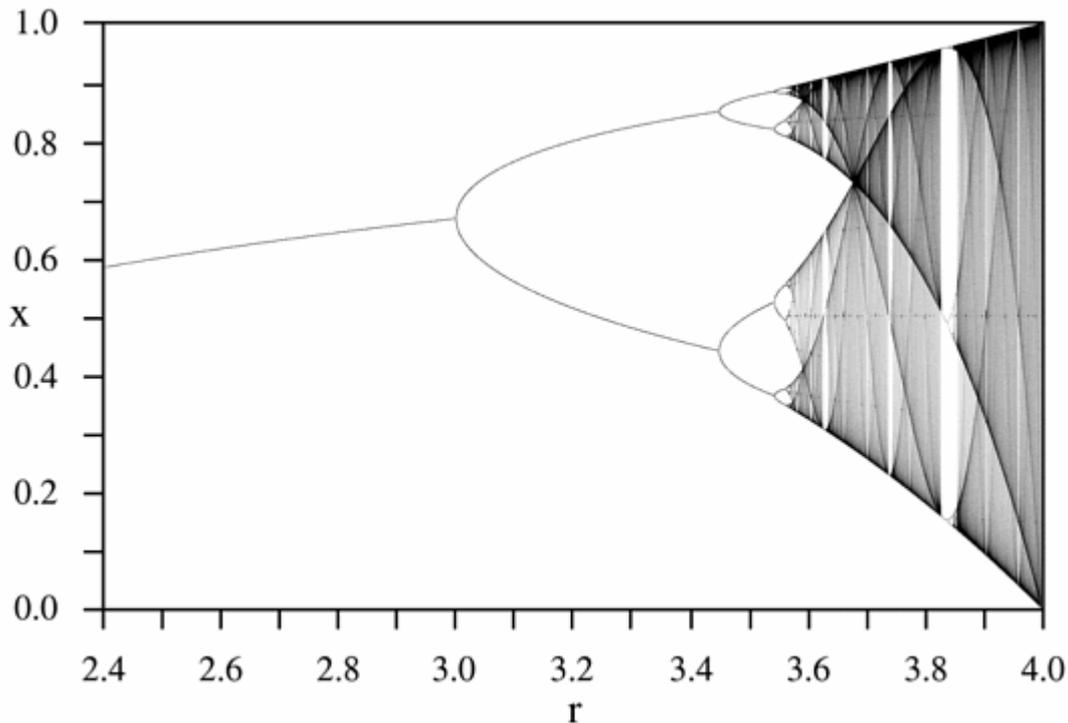
Solche Verzweigungen des Systemverhaltens (Bifurkationen) lassen sich in Bifurkationsdiagrammen graphisch darstellen, was insbesondere bei der Abhängigkeit von nur einem Parameter sehr anschaulich ist:

Die X-Achse wird mit dem veränderliche Parameter belegt. Das sollte bei uns die Amplitude der

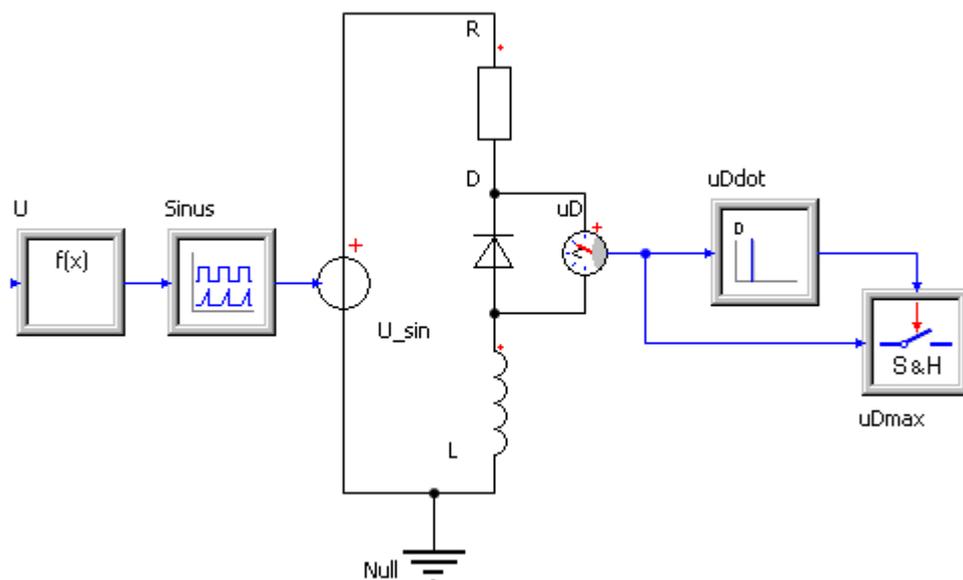
anregenden Sinus-Spannung sein.

Die Y-Achse sollte eine markante Systemgröße darstellen, welche die angestrebten Grenzwerte repräsentiert. In unserem Beispiel ist das die Amplitude der Diodenspannung, welche nach der Verzweigung zwischen den beiden Wert schwankt.

Ein markantes Beispiel ist das Bifurkationsdiagramm der logistischen Gleichung (Bildquelle: Wikimedia Commons):



Wir werden unser Modell so erweitern, dass wir den Verhaltensumschlag infolge der Verdopplung der Periodendauer in solch einem Bifurkationsdiagramm darstellen können. Dafür sind größere Umbauten im Modell erforderlich. Deshalb erzeugen wir eine Kopie unserer bisherigen Modelldatei und arbeiten mit dieser Kopie weiter:



Hinweis: Die Parallelkapazität zur Diode von 10 pF musste entfernt werden, weil es ansonsten zu numerischen Problemen mit dem ereignisgesteuerten Abtastglied kommt. Das Verhalten ändert sich dadurch praktisch nicht!

Leider kann man das bisherige Modell-Element der sinusförmigen Spannungsquelle nur mit einer konstanten Amplitude betreiben. Wir wollen im Simulationszeitbereich die Amplitude jedoch in einem vorgegebenem Bereich variieren:

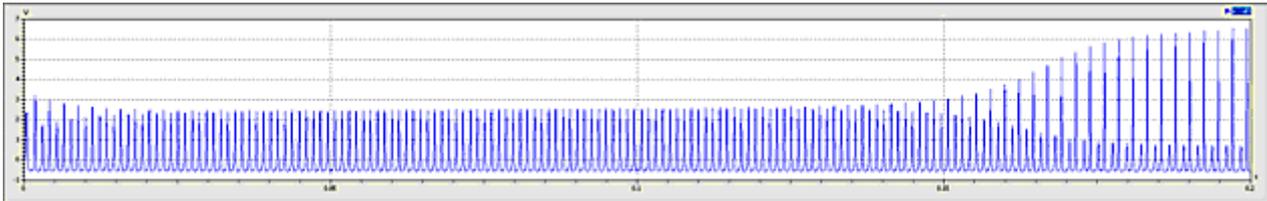
Wir müssen deshalb eine **Spannungsquelle mit Signaleingang** verwenden.

Diese speisen wir mit dem Ausgang eines Signalgenerators, dessen Amplitude wir über seinen Eingang x steuern.

Die Steuerung für einen vorläufigen Amplitudenbereich der Sinus-Amplitude von 0,9...1,3 V übernimmt ein Function-Block mit

$$0.9 + 0.4 * (t - t_{\text{Start}}) / (t_{\text{Stop}} - t_{\text{Start}})$$

Damit der Schwingkreis genügend Zeit hat, der Spannungsänderung zu folgen, erhöhen wir die Simulationszeit auf $t_{\text{Stop}}=0,5$ s. Damit ergibt sich ein Verlauf der Diodenspannung, welcher deutlich die Aufspaltung in zwei Amplitudengrenzwerte zeigt:

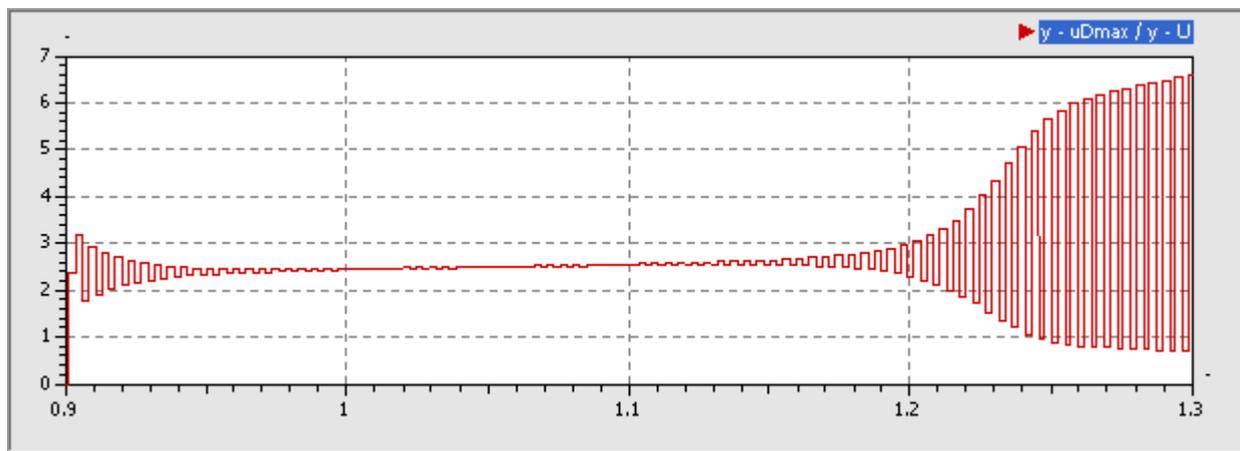


Das Ergänzen der **uDmax**-Erfassung sollte mit unseren bisherigen Erfahrungen auch kein Problem sein:

Mit einem Differenzierer bilden wir die zeitliche Ableitung **uDdot.y** der gemessenen Kondensatorspannung.

Die Nulldurchgänge von **uDdot.y** "von oben nach unten" kennzeichnen die Maxima der Kondensatorspannung **uD.v**. An diesen Stellen wird der Wert der Kondensatorspannung als **uDmax.y** übernommen.

Die Darstellung **uDmax.y=f(U.y)** ergibt das gewünschte Bifurkationsdiagramm:



Am Anfang erkennt man den Einschwingvorgang infolge des Einschaltens der Sinusspannung von 0,9 V. Für den eingeschwungenen Betrieb bei 0,9 V ist jedoch der eine Grenzwert typisch.

Erst nach dem Überschreiten von 1,2 V kommt es zum Aufspalten in zwei Grenzwerte.

Der Weg zum perfekten Chaos

Mit etwas Geduld und Experimentierfreude gelangt man sicher durch systematisches Probieren irgendwann zum perfekten Chaos. Um im Rahmen der knapp bemessenen Übungszeit die gewünschten Effekte beobachten zu können, sollen hier einige Erkenntnisse des Autors einfließen:

Beim Durchfahren der Anregungsamplitude von nahe Null bis zu einem Maximalwert müssen hinreichend viele Periodenverdopplungen stattfinden, damit Chaos entsteht. In der Experiment-Anleitung wurde von 4 Periodenverdopplungen gesprochen. Leider wird dies mit der eingestellten Erregerfrequenz wahrscheinlich nicht gelingen.

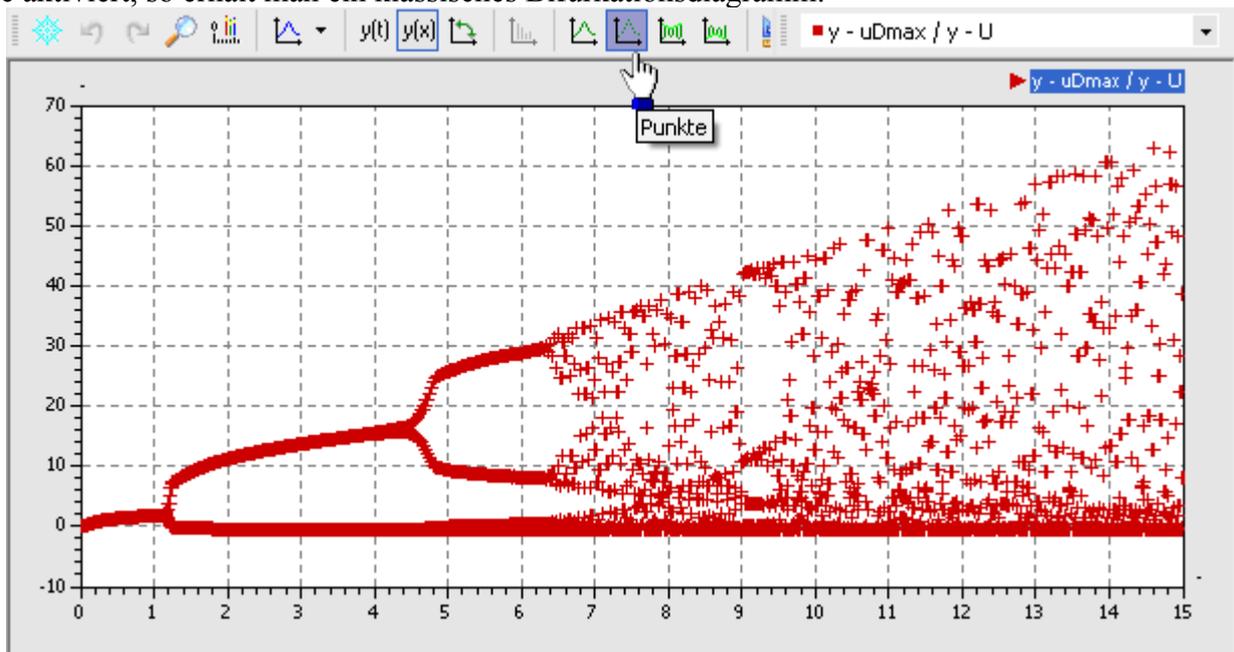
Es ist günstig, im bisher betrachteten Bereich möglichst schnell zu Periodenverdopplungen zu kommen.

Das gelingt, indem man die Anregungsfrequenz etwas erhöht. Im Beispiel war das bei einer Erhöhung um

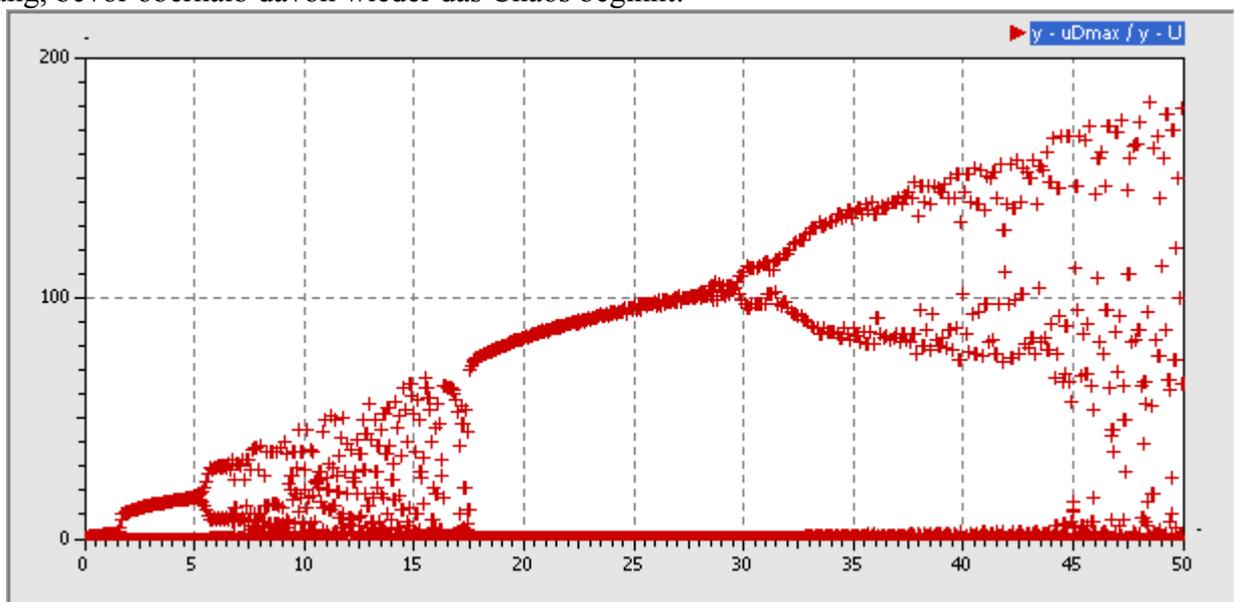
ca. 20% von 860 Hz auf 1000 Hz der Fall.

Man durchfährt eine Anregungsamplitude von **0...15 V** in einer Simulationszeit von **2 s**.

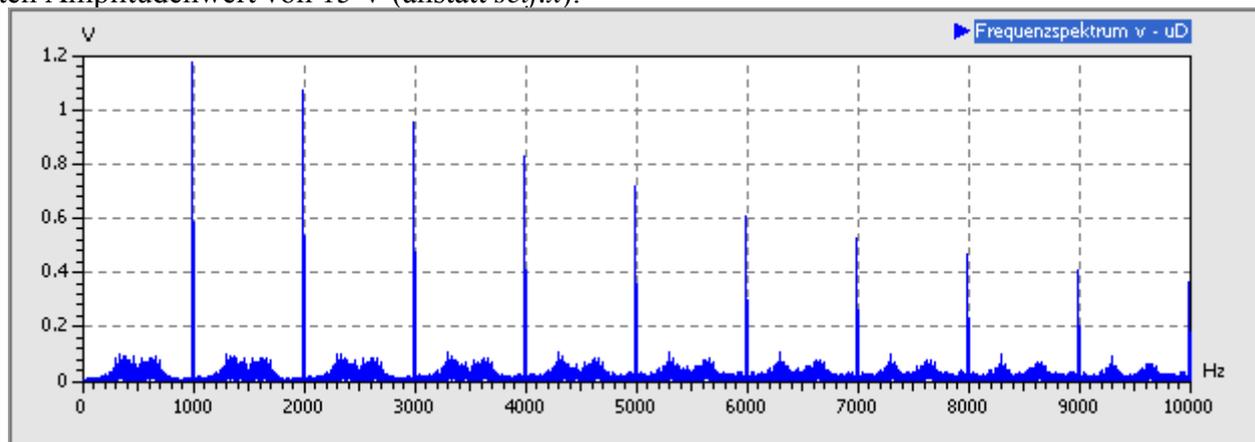
Da die jeweiligen Perioden-Maxima im Ergebnisfenster standardmäßig mit Linien verbunden werden, kann man die Bifurkationen nur schlecht erkennen. Wenn man anstatt der Liniendarstellung die Darstellung der Punkte aktiviert, so erhält man ein klassisches Bifurkationsdiagramm:



In der Hoffnung, dass die Sperrschicht der Diode nicht durchschlägt, kann man die Anregungsamplitude z.B. bis auf 50 V ansteigen lassen. Man erkennt dann im Bereich von 20...30 V ein Gebiet relativer Ordnung, bevor oberhalb davon wieder das Chaos beginnt:



Um das Verhalten z.B. bei 15 V Anregungsamplitude zu untersuchen, verwendet man im Generator einen konstanten Amplitudenwert von 15 V (anstatt *self.x*):

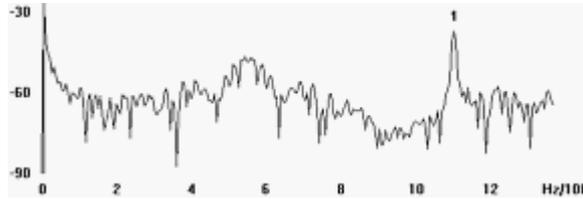


Unterhalb der Anregungsfrequenz und zwischen ihren Harmonischen ist nun im Spektrum Rauschen sichtbar, d.h. alle Frequenzen sind vertreten.

Das obige Spektrum kann man als Beginn des chaotischen Verhaltens interpretieren.

"Perfektes Chaos" wäre durch ein intensives Rosa Rauschen gekennzeichnet. Leider kann man das wahrscheinlich mit unserem Simulationsmodell nicht so einfach oder gar nicht erreichen.

Im realen Experiment (Siehe PDF-Datei) ist es jedoch durch feinfühliges Abstimmen möglich, solch einen Zustand zu erreichen. Dies zeigt ein Bild mit dem Frequenzbereich unterhalb der dort verwendeten Erregerfrequenz von ca. 1100 Hz:



Der Weg war das Ziel - auch der zum perfekten Chaos!

← →

Von „http://optiyummy.de/index.php?title=Software:_SimX_-_Einfuehrung_-_Elektro-Chaos_-_Schwingkreis_mit_C-Diode“
