

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?

Antworten auf obige Fragen erhalten wir bei der Bearbeitung der im Folgenden aufgelisteten Übungsabschnitte:

0. Software

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:

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

1. Lineare Elemente und Systeme → PDF (Stand 13.09.2012)

- . 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 → PDF (Stand 16.09.2012)

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

3. Einsendung von Lösungen

- . Teilnehmer des Sommerkurses senden fertige Modell-Dateien umgehend als Anhang einer **E-Mail** mit dem **Betreff "Name, Vorname: Modellname"** an den Betreuer.
- . Die Modelldatei ist in einem Archiv (z.B. ZIP) zu packen, die zum Modell gestellten Fragen sind im Text der Mail zu beantworten.
- . E-Mail-Adresse: **a.kamusella@ifte.de**

← →

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.

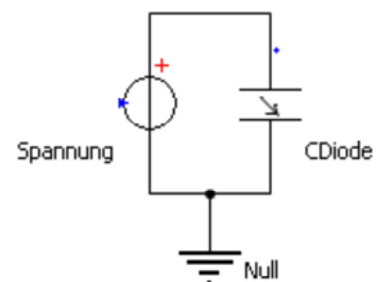
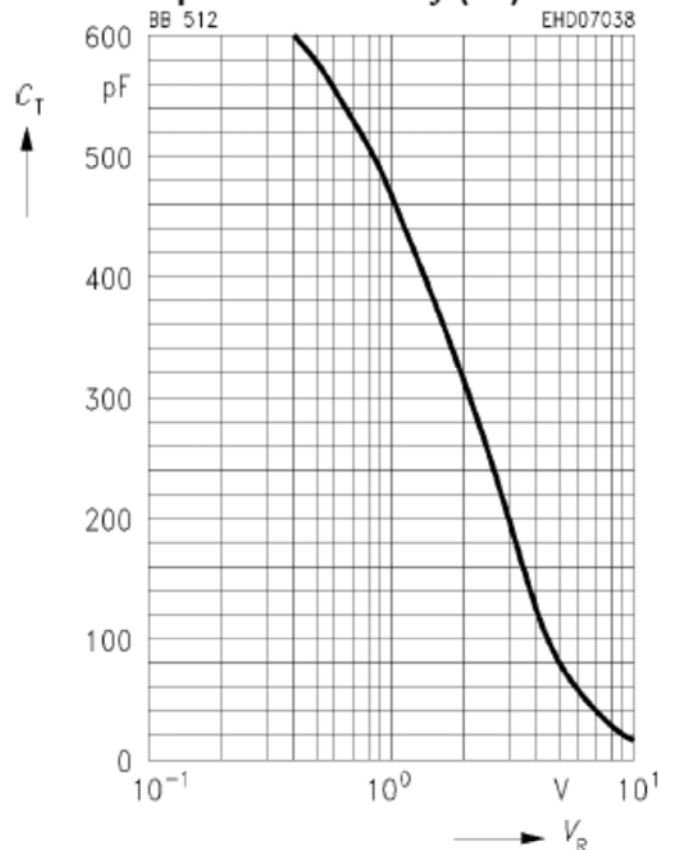
Wir übernehmen im Verhaltensalgorithmus wie üblich die Spannungsdifferenz der Anschlüsse und speisen die Anschluss-Ströme mit dem Strom.

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.

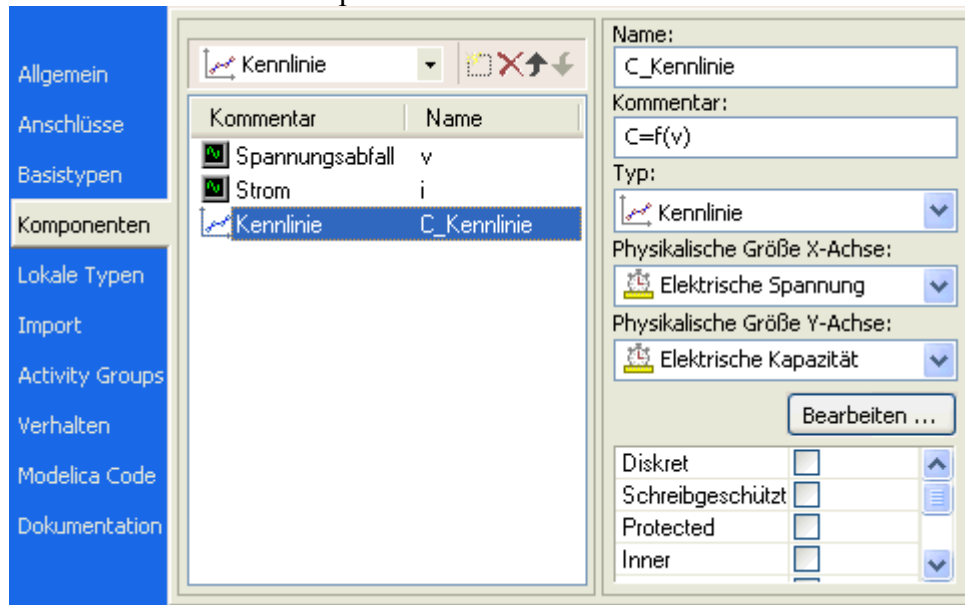
Diode capacitance $C_T = f(V_R)$



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:

```

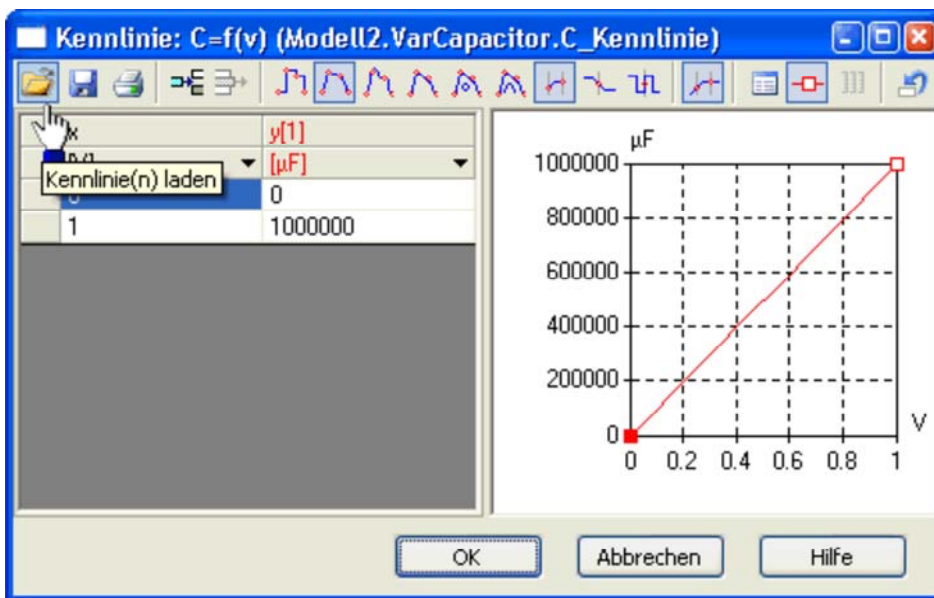
11       1.2e-011
12       1.06e-011
13       1e-011
14       9.5e-012
15       9.395e-012

```

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:

V	C [µF]
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:

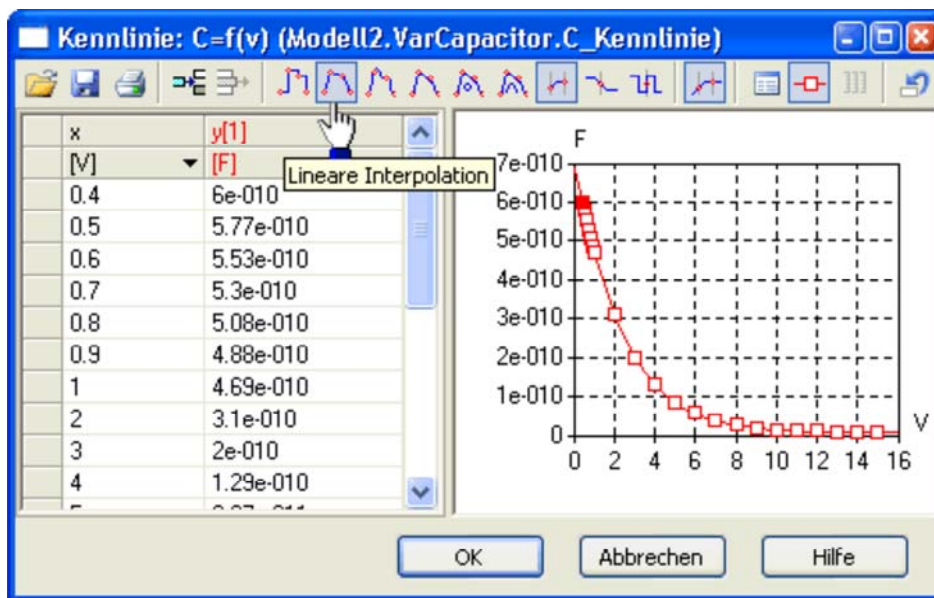
Phys. Größe:

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:

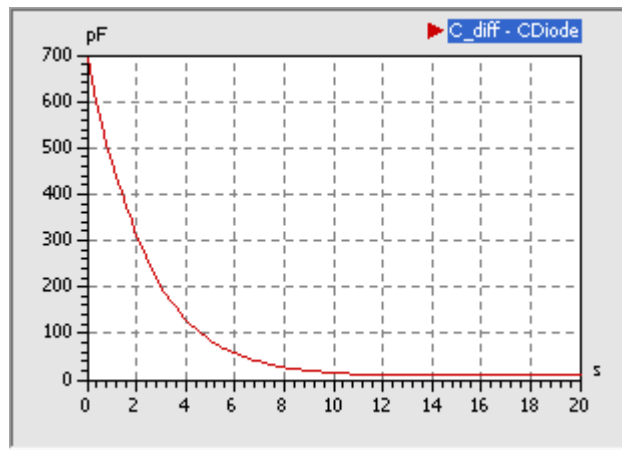
$$C_D = \frac{dQ}{dU}$$

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

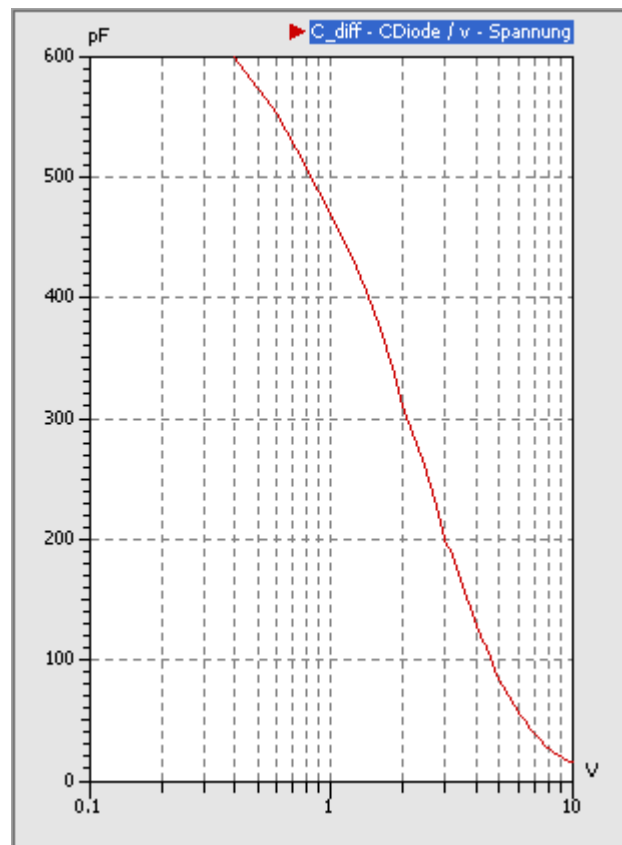
Wir definieren deshalb als Komponente eine Variable C_{diff} und weisen diese im Verhaltensalgorithmus den aktuellen Kennlinienwert zu. Dabei benutzen wir den Absolutwert der Spannung, um für den negativen Spannungsbereich numerische Probleme zu vermeiden:

```
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:



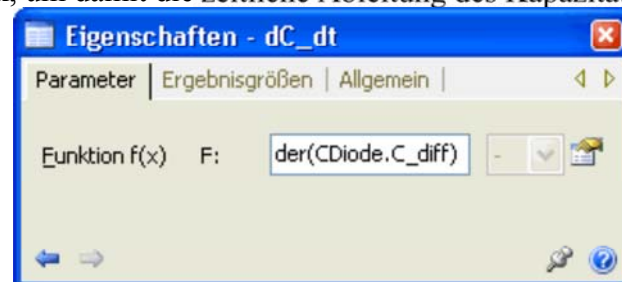
Die Abhängigkeit der Dioden-Kapazität von der anliegenden Spannung kann man durch geeignete Konfiguration eines y(x)-Signalfensters in der gleichen Form darstellen, wie im zugrunde gelegten Datenblatt:



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

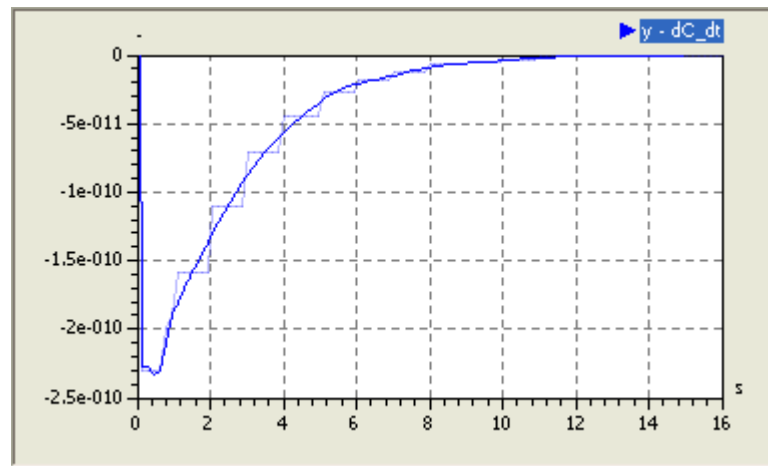
Es kann zwar sein, dass dies auf die Genauigkeit der späteren Simulation keinen Einfluss hat. Aber natürlich sollte man versuchen, eine nichtlineare Kennlinie möglichst gut nachzubilden.

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 blasse 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;

```

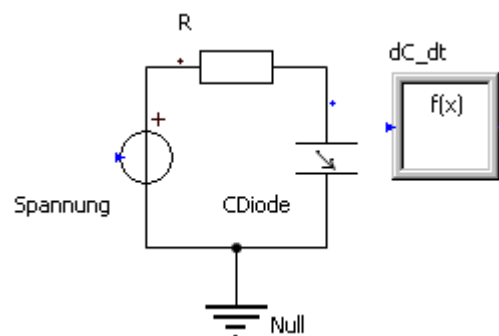
Die erforderliche Variable **vdot** für die zeitliche Ableitung der Spannung müssen wir als zusätzliche Komponente definieren.

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.

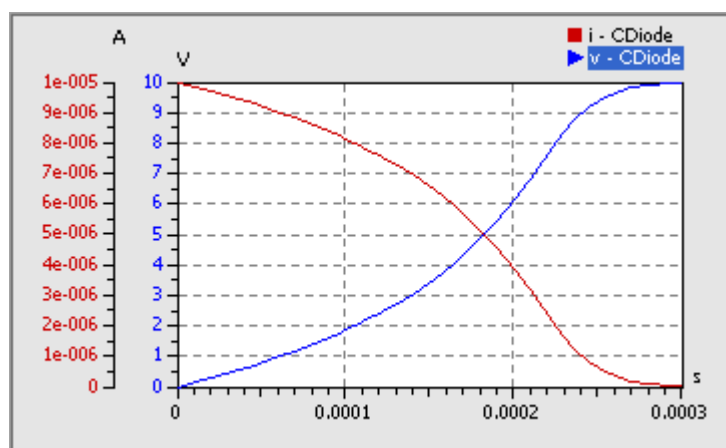
Im nächsten Schritt wollen wir eine konstante Betriebsspannung von **10 V** vorgeben und den Kondensator über einen Vorwiderstand **R** aufladen:



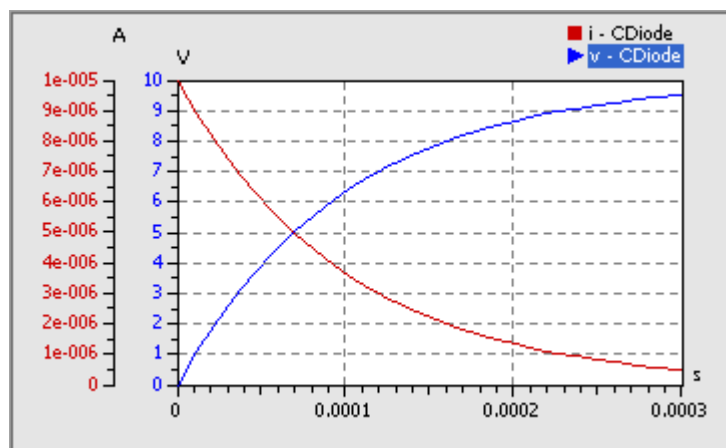
Da die maximale Kapazität ungefähr **1 nF** beträgt, werden wir den Vorwiderstand so dimensionieren, dass wir damit eine Zeitkonstante von **1 ms** erreichen. Die tatsächliche Zeitkonstante wird sich natürlich durch die sich verringernde Kapazität verändern.

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.

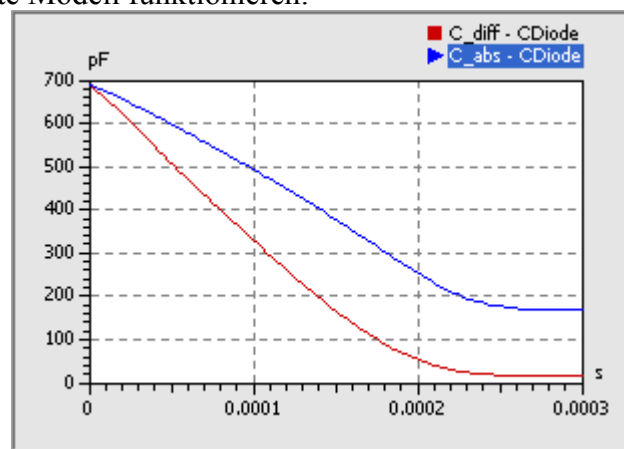
Es lässt sich auch kaum vermeiden, dass während des Betriebs die Spannung am Kondensator auch mal den Wert Null annimmt. Wir müssen also eine Möglichkeit finden, dass dabei trotzdem ein sinnvoller Wert für C_{abs} berechnet wird:

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.

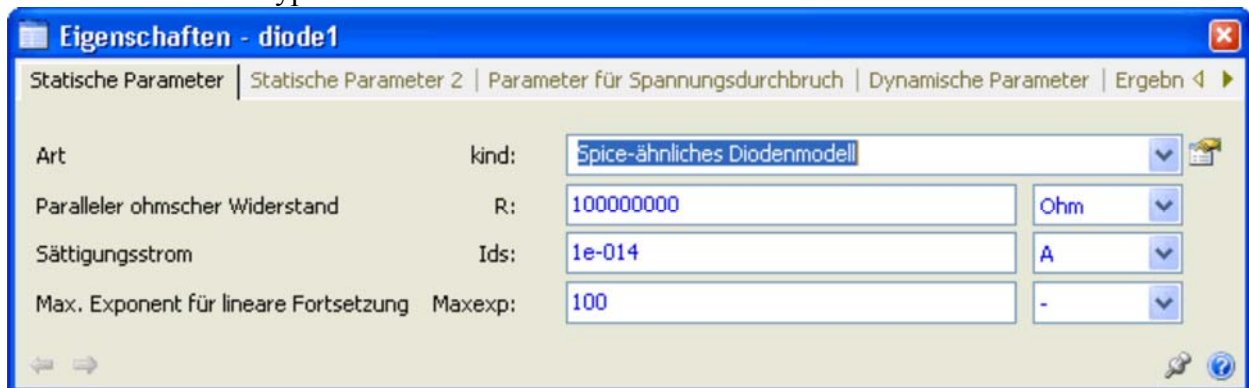
Inhaltsverzeichnis

- . 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.

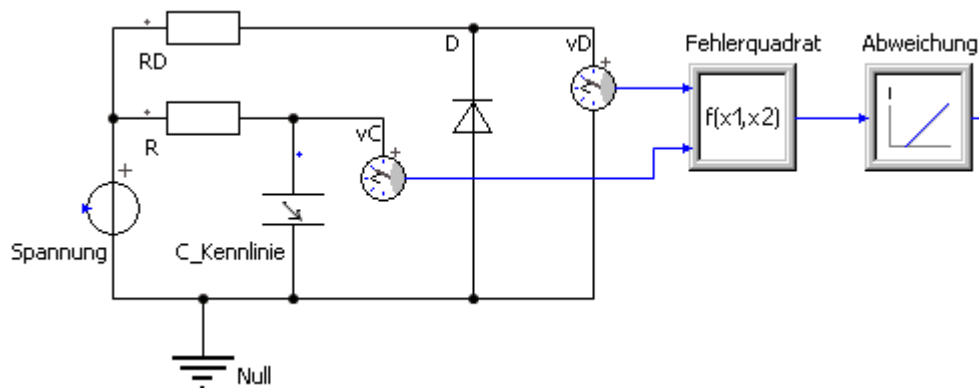
Im Dioden-Modell wird der aktuelle Wert der Kapazität nicht als Variablen-Wert berechnet. Deshalb entfällt der direkte Vergleich der Kapazitäten von Diode und Kennlinien-Modell als Kriterium für die Optimierung. Man könnte mit etwas Aufwand die Berechnung der Dioden-Kapazität ergänzen, aber wir gehen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

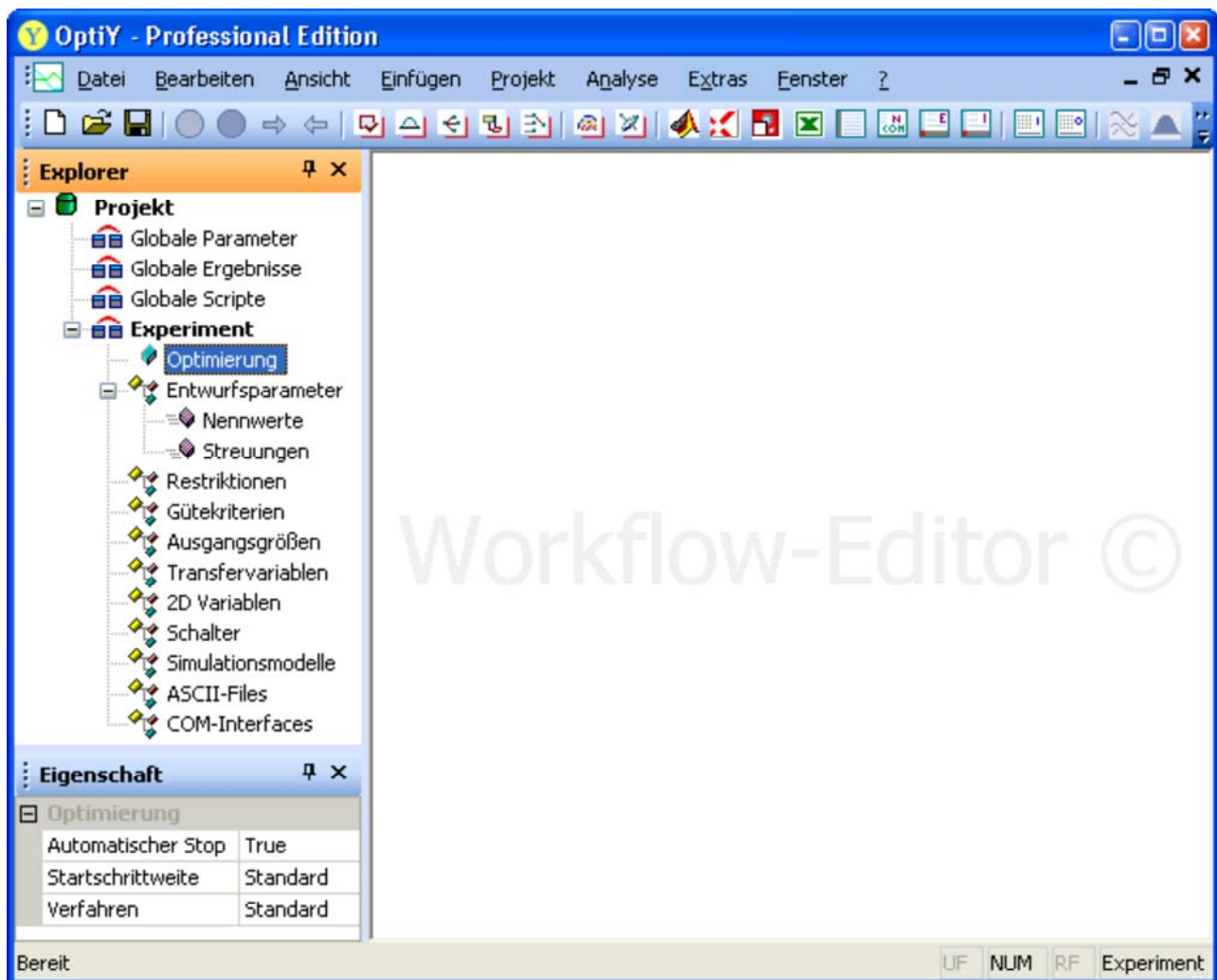
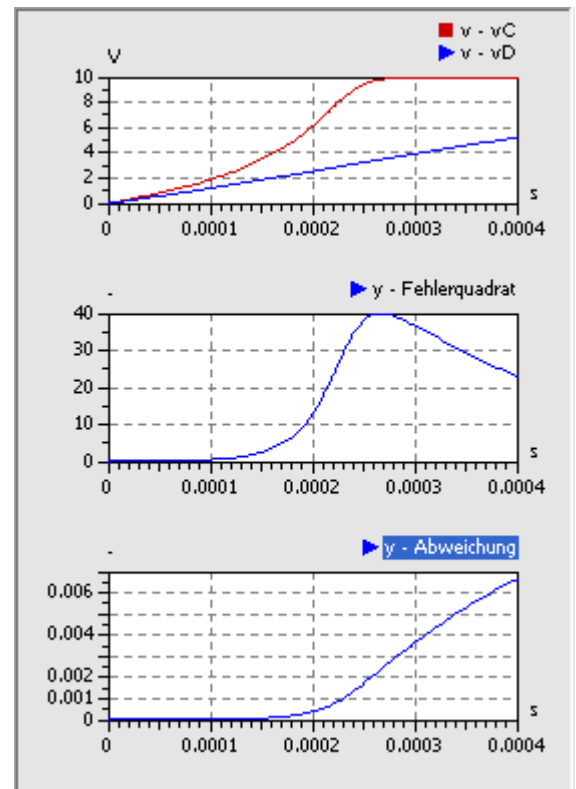
Für die restlichen Dioden-Parameter benutzen wir ebenfalls die Vorgabewerte.

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:

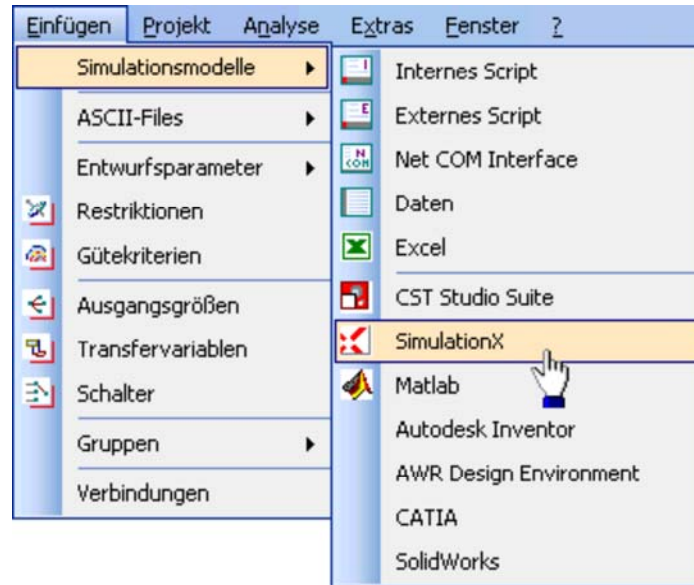


Hinweis: Das Modell **Parameterfindung.ism** im SimulationX sollte geöffnet sein, bevor man dieses Modell in den OptiY-Workflow einbindet.

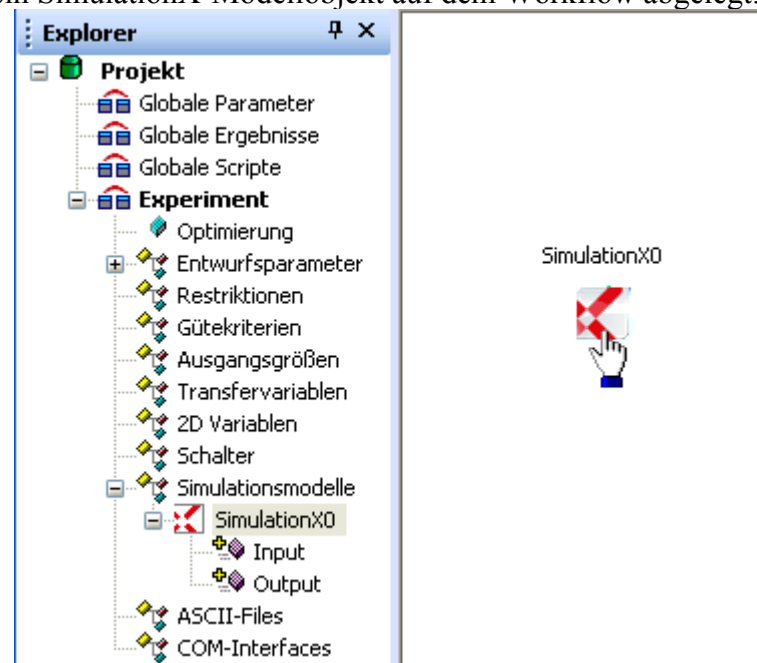
Experiment-Workflow

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*:

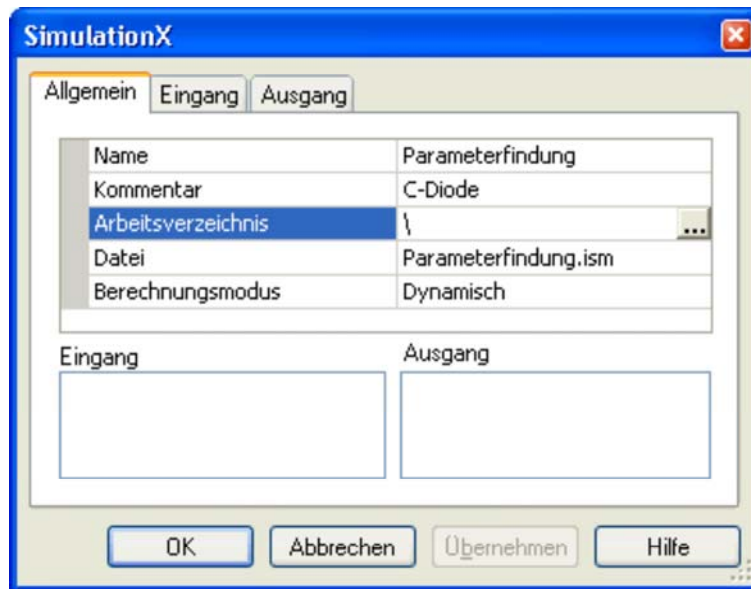


Welches konkrete Modell sich dahinter verbirgt, wird vorläufig nicht spezifiziert, sondern es wird mittels Mausclick vorerst nur ein SimulationX-Modellobjekt auf dem Workflow abgelegt:



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

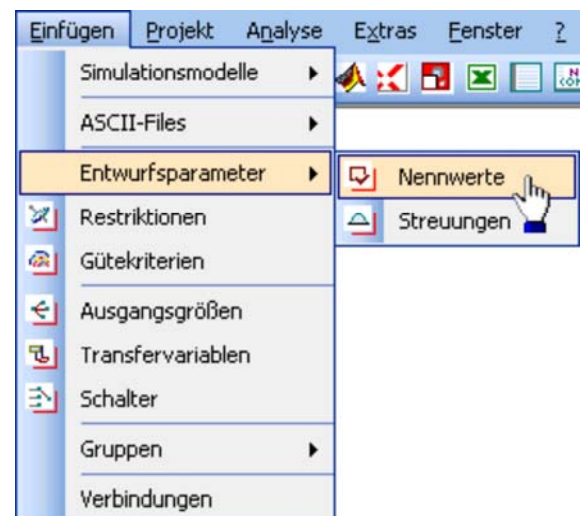
M: Kapazitätskoeffizient

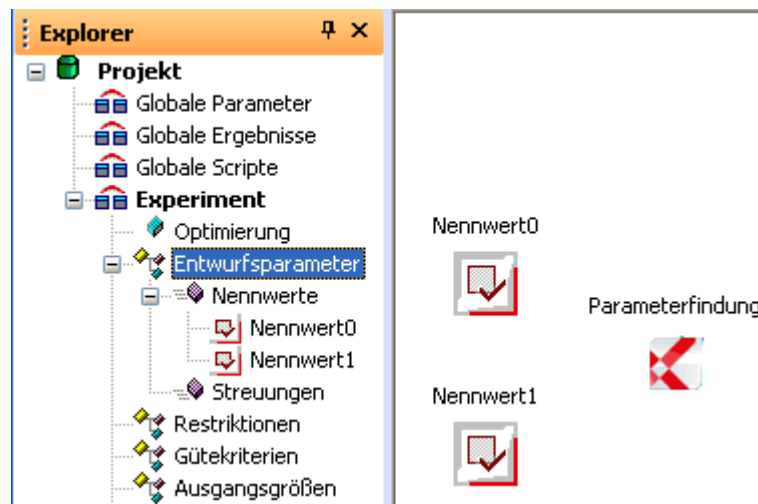
Wir definieren die zwei Entwurfparameter im *Workflow-Editor* durch *Einfügen* > *Entwurfparameter* > *Nennwerte*:

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.

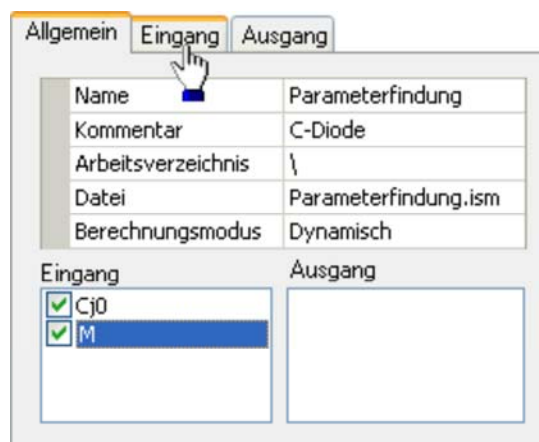
Die Standardbezeichner *Nennwert_x* sollte man durch sinnvolle Namen versehen. Ein zusätzlicher Kommentar erleichtert die spätere Deutung:

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

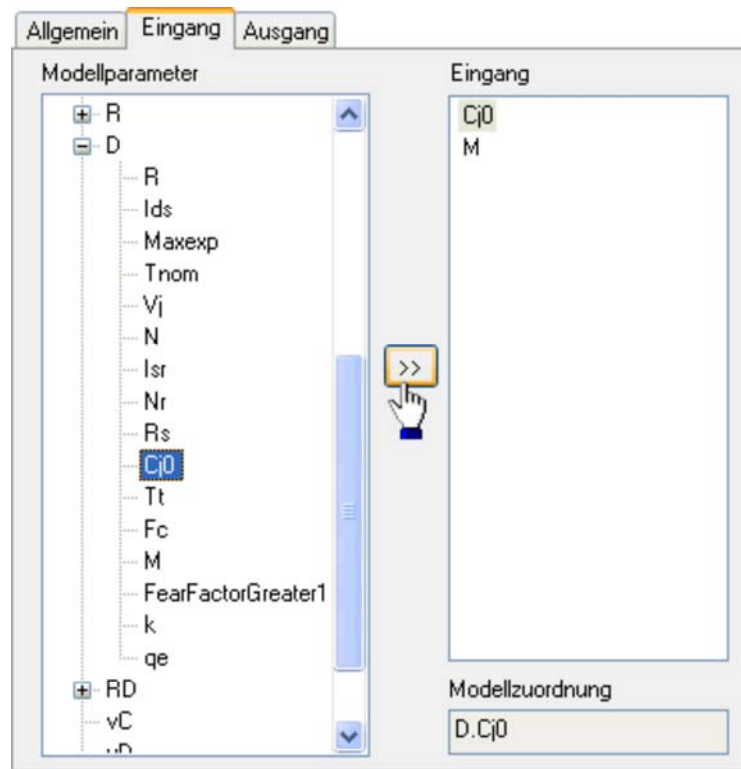


Hinweis: Die Einheiten und Werte der Entwurfparameter werden später automatisch bei der Zuordnung zu den konkreten Parametern des SimulationX-Modells ergänzt!

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 Eingangsparameter und wechseln danach in die Registerkarte **Eingang**:

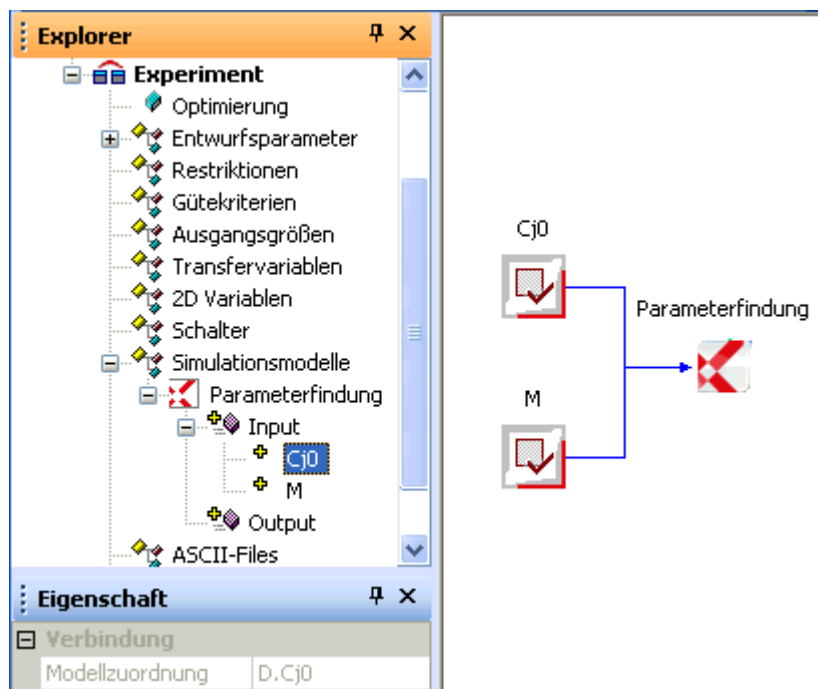


Dort wählt man nacheinander jeweils die zusammengehörenden Paare aus Modell-Parameter und Eingang und stellt die Verbindung durch Betätigen des >>-Button her:

**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.

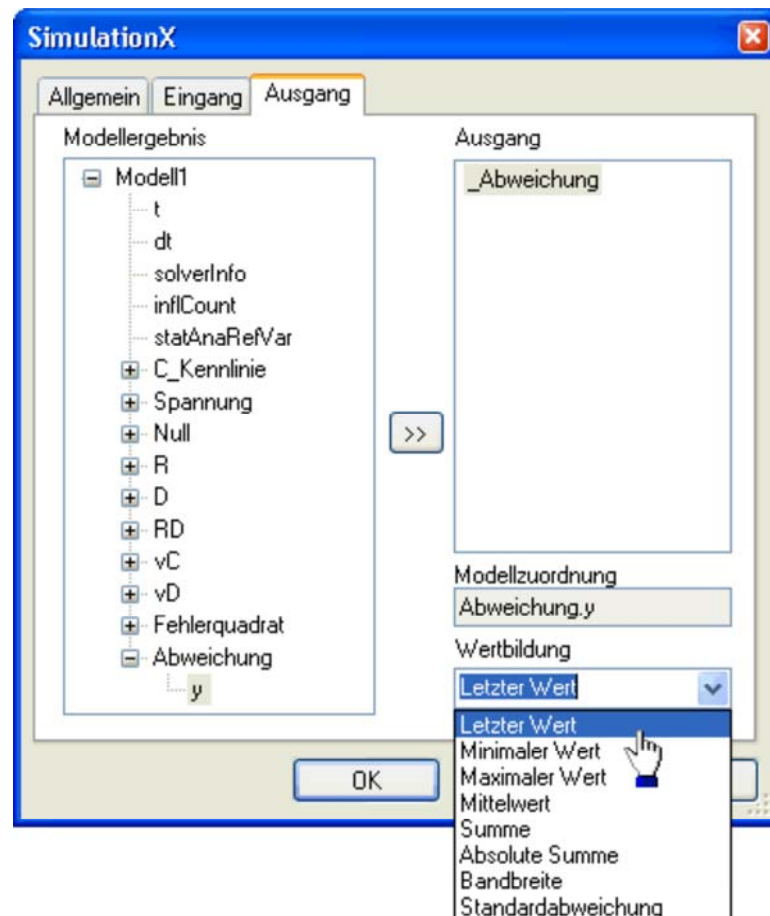
Wir haben die bestmöglichen Parameter für die C-Diode gefunden, wenn die Abweichung nicht mehr verkleinert werden kann.

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

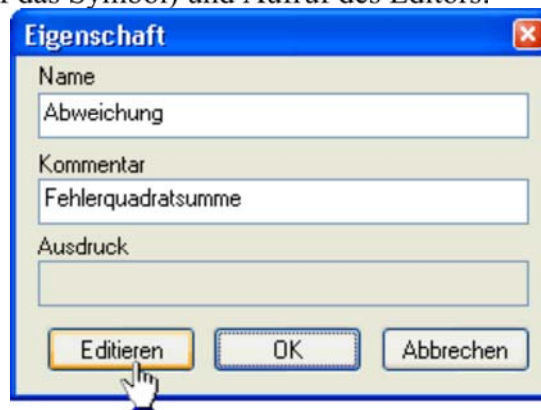
Bewertungsgrößen (hier Gütekriterium) kann man nicht direkt den konkreten Modellgrößen zuordnen. Dafür benötigt man jeweils eine zusätzliche Ausgangsgröße (**Einfügen** > **Ausgangsgrößen**):

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:

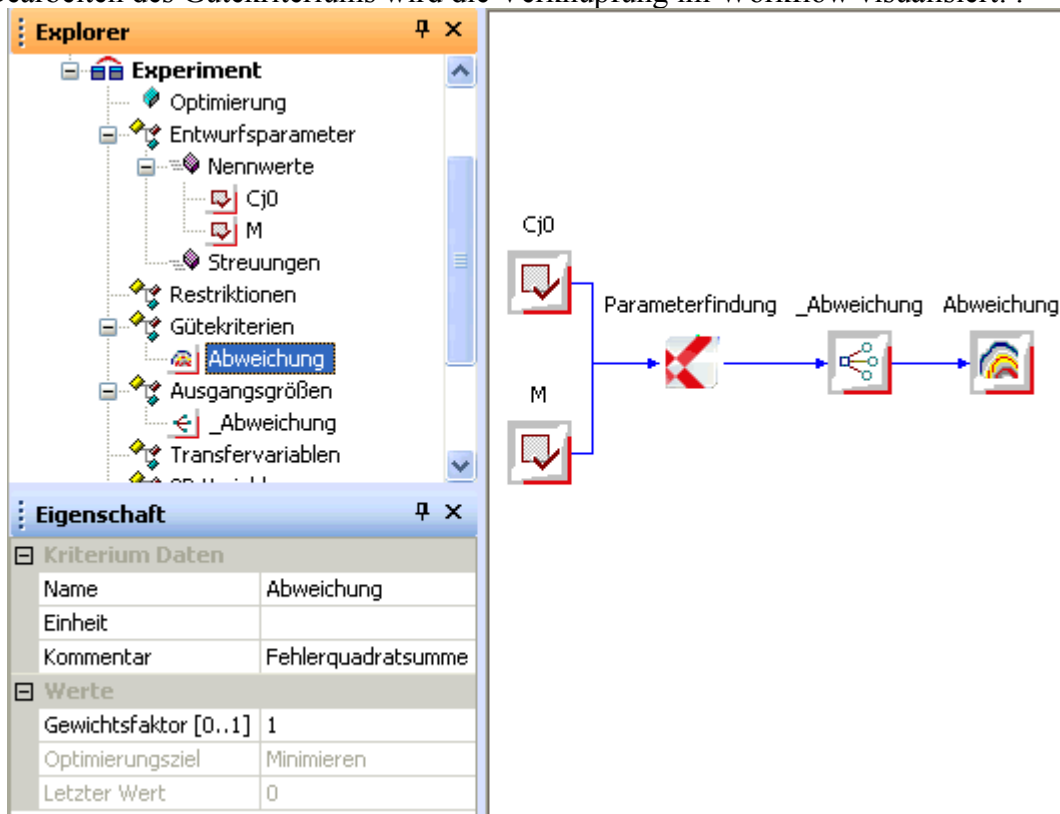


Der Wert einer Bewertungsgröße kann durch mathematische Verknüpfung aller Input- und Output-Größen des Workflows gebildet werden. Diese Verknüpfung wird in Form eines Ausdrucks notiert.

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

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.

Explorer	
Experiment	
Optimierung	
Entwurfparameter	
Restriktionen	
Gütekriterien	
Ausgangsgrößen	
Transfervariablen	
2D Variablen	
Schalter	
Simulationsmodelle	
ASCII-Files	
COM-Interfaces	

Eigenschaft	
Optimierung	
Automatischer Stop	True
Startschrittweite	Standard
Verfahren	Standard

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

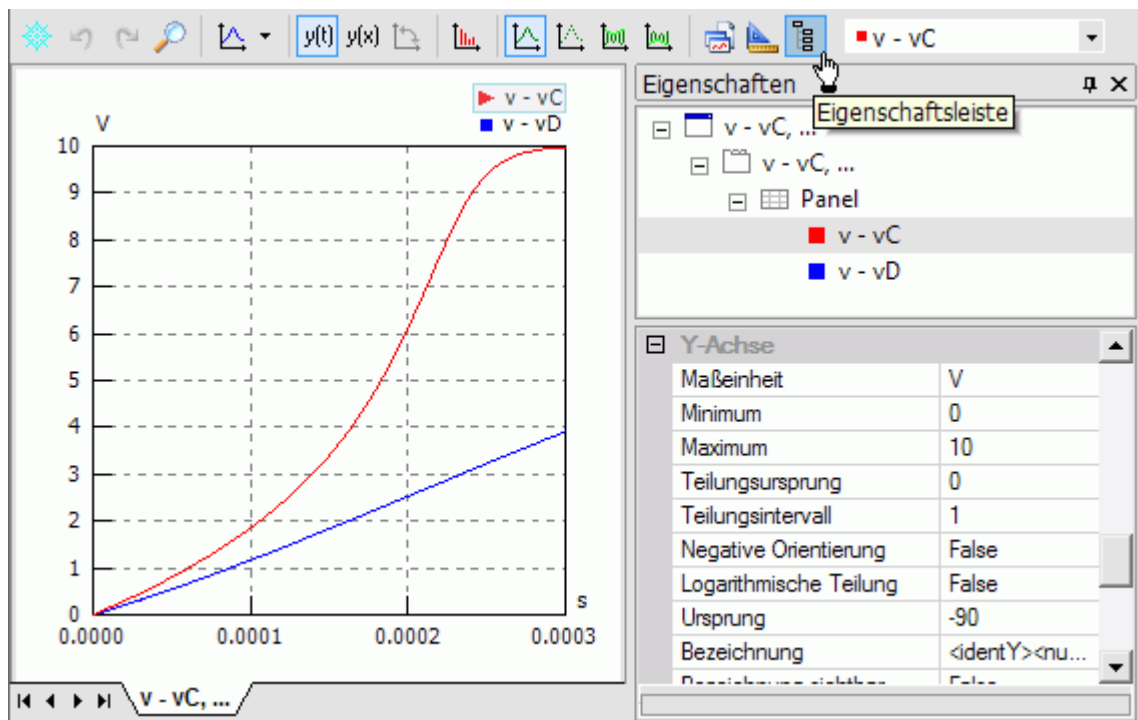
Um das Optimierungsexperiment verfolgen und beurteilen zu können, muss man alle dafür wesentlichen Informationen in geeigneter Form visualisieren. Dazu gehören auch Signalverläufe des Simulationsmodells:

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.

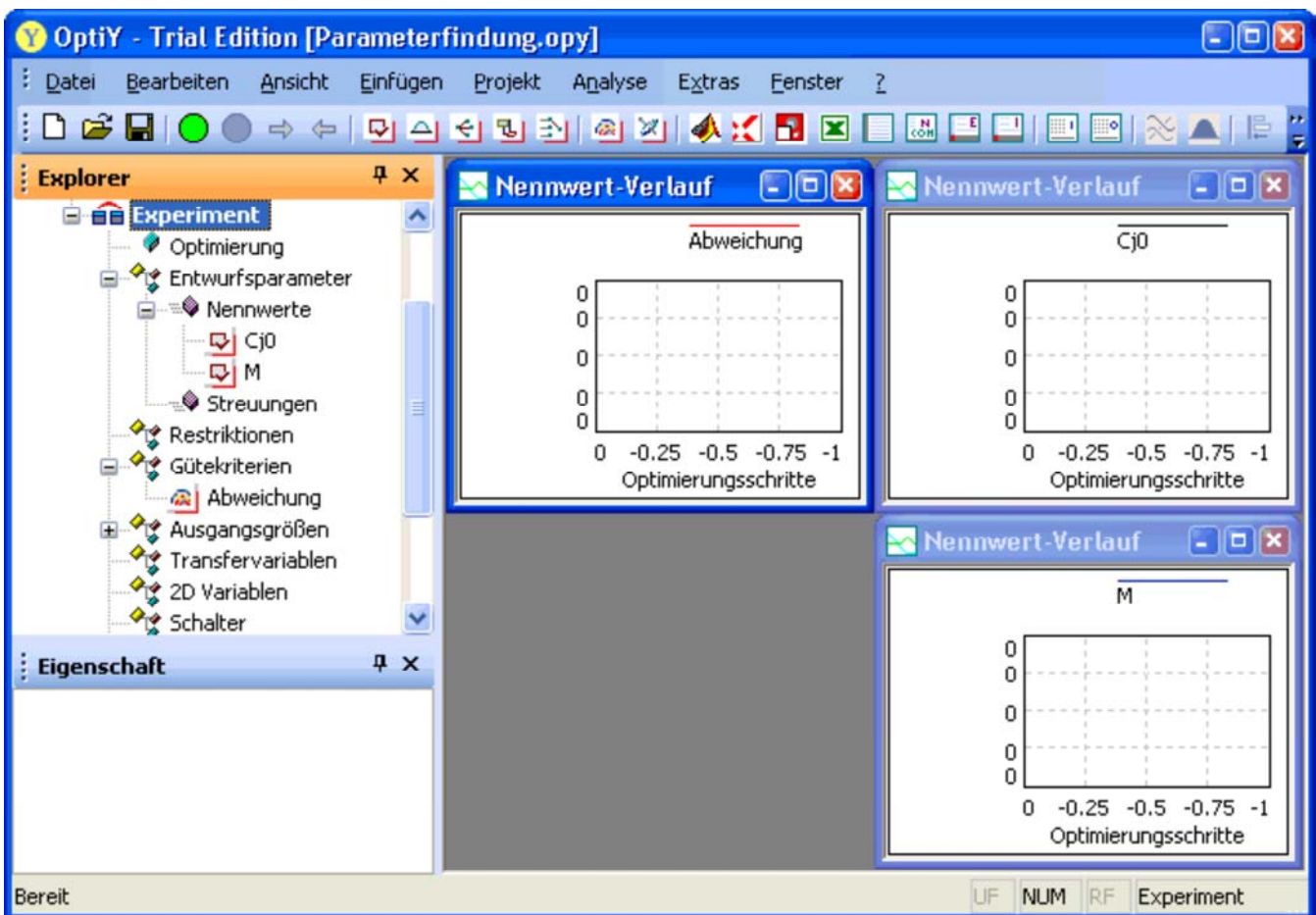
2. OptiY-Experiment

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

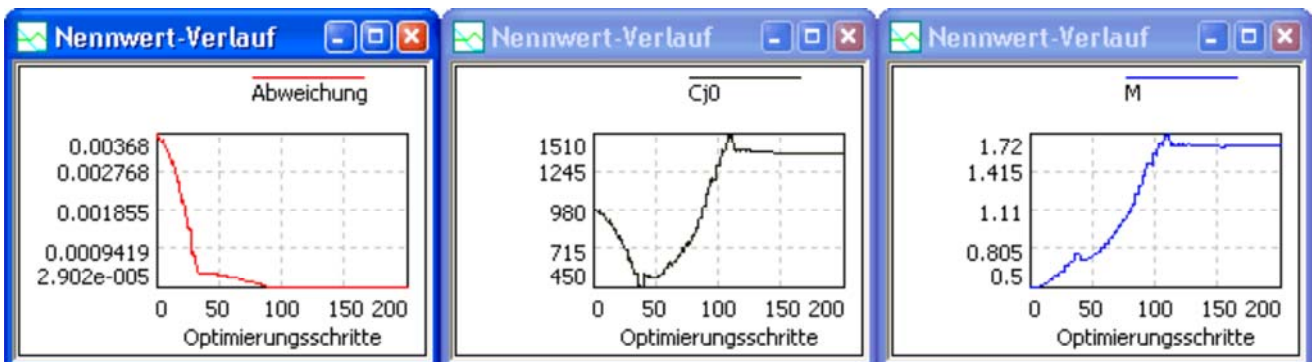
Spätestens jetzt sollte man die aktuelle Konfiguration des OptiY-Versuchsstandes mal wieder als **.opy**-Datei speichern!

Experiment-Durchfuehrung

Lokale Suche

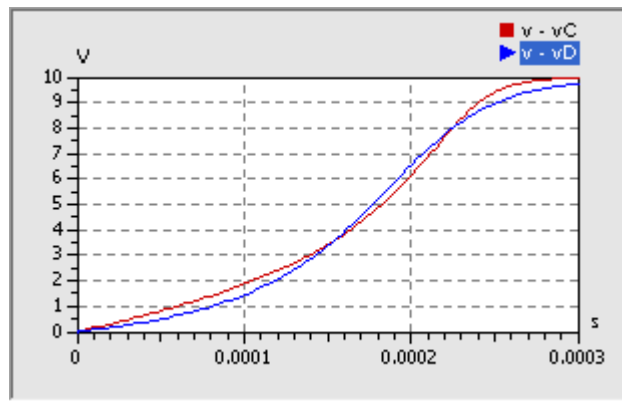
Nach soviel Vorbereitung wollen wir nun beherzt den Start-Button  für das Experiment drücken und hoffen, dass die optimalen Dioden-Parameter gefunden werden:

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:



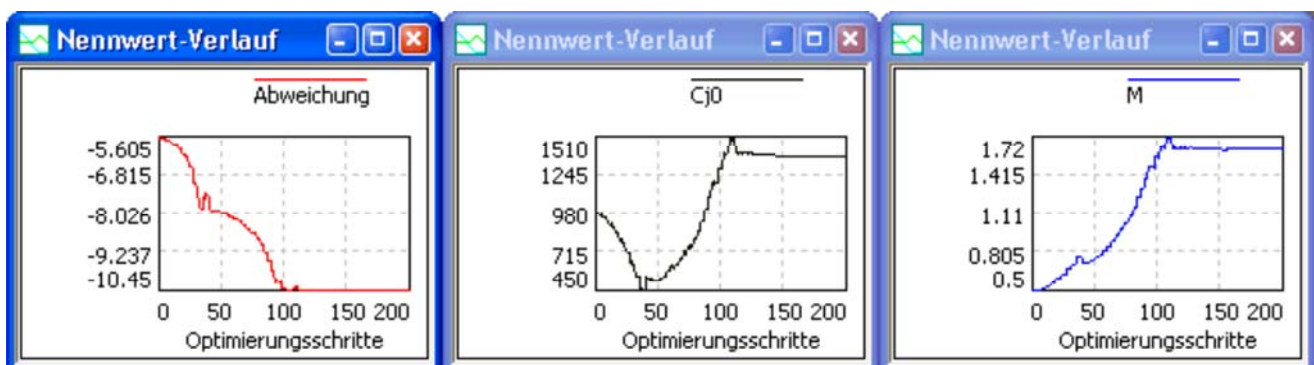
Da die Spannungssignale noch gewisse Abweichungen zeigen, werden wir untersuchen, ob noch weitere Verbesserungen möglich sind. Dazu betrachten wir zuerst den Optimierungsverlauf:

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!

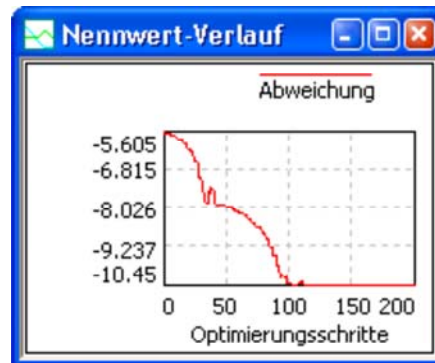
Das Hooke-Jeeves-Verfahren konvergiert durch ständige Abwärtsbewegung auf der Gütefunktion zum nächstgelegenen Minimum:

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

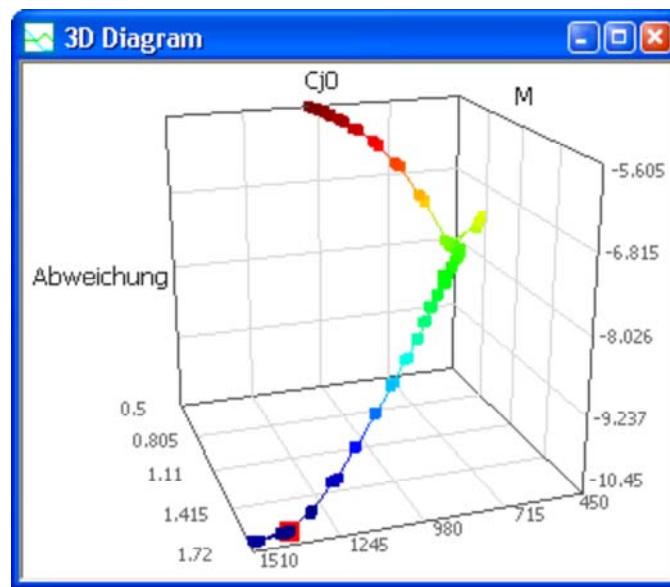
Man kann sich dabei jedoch nie sicher sein, ob nicht doch noch Kombinationen für die Entwurfsparameter existieren, welche zu einer besseren Lösung führen (möglichst zum globalen Minimum der Gütefunktion). Es wäre günstig, wenn man die Topografie der Gütefunktion in Analogie zu einer Landkarte kennen 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**:

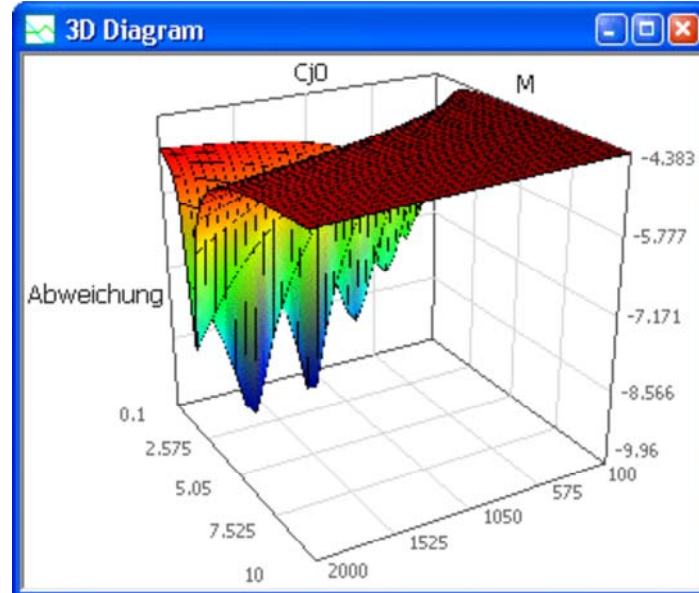


Leider sieht man bei der lokalen Suche nur die Punkte der Gütefunktion, welche durch das Verfahren abgetastet werden. Wie die komplette Gütefunktion im Suchraum aussieht, bleibt uns dabei verborgen:

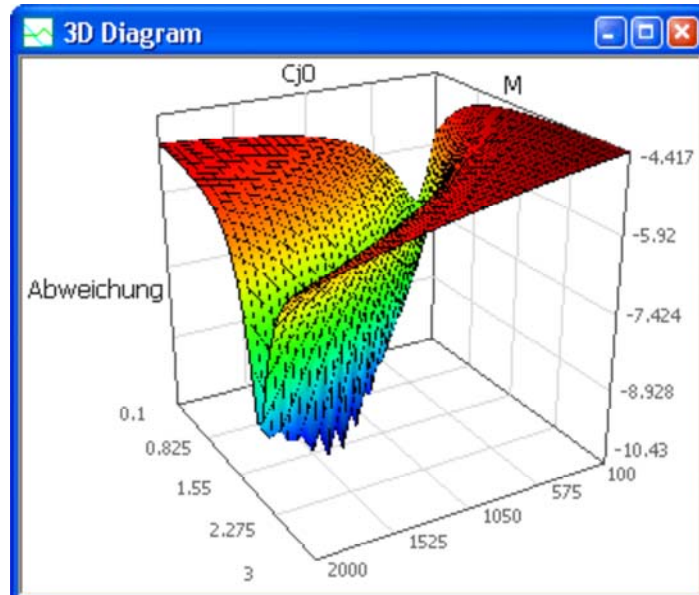
- Es ist bei praktischen Problemstellungen meist nicht möglich, die komplette Gütefunktion z.B. durch hinreichend dichte Stützstellen zu erfassen. Insbesondere bei mehreren Entwurfsgrößen und langsam rechnenden Modellen würde man Tage oder sogar Jahre für die Berechnung benötigen.
- Unser Simulationsmodell 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:

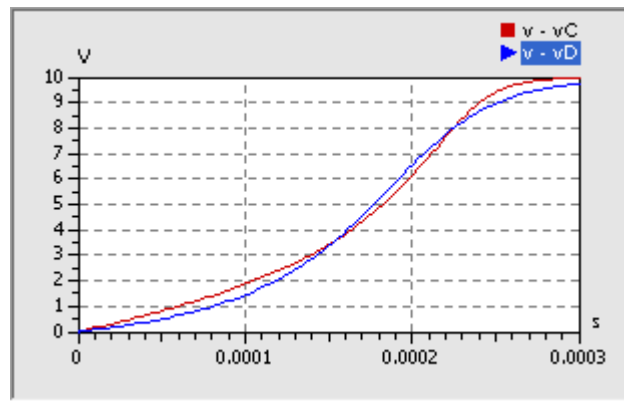


Hinweis: Falls der Flächenteil mit großen M-Werten nicht richtig berechnet wird, kann man dieses numerische Problem meist durch eine Verringerung von $dt_{Min}=1e-20$ s in der Simulationssteuerung lösen. Man erkennt mehrere lokale Minima in einem engen Graben. Trotz der dichten Rasterung kann man Darstell-Effekte 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

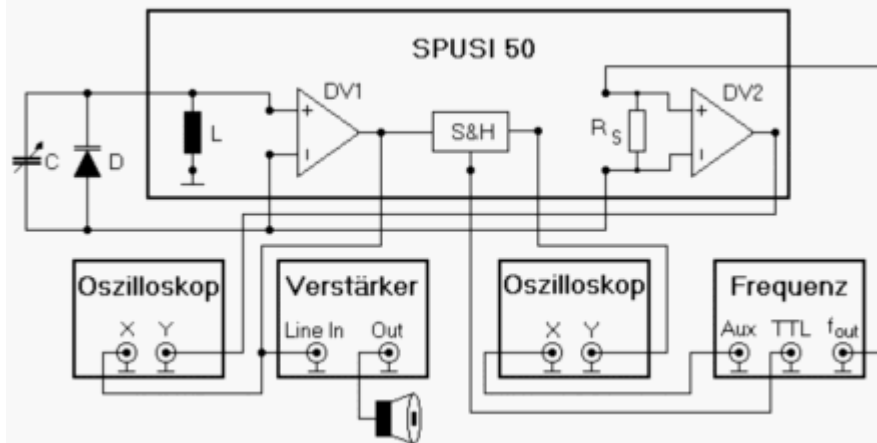
Aus OptiYummy

↑

← →

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

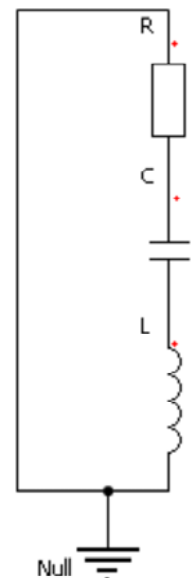
Damit ein nichtlineares System chaotisches Verhalten zeigt, müssen zusätzlich zur "Nichtlinearität" weitere Kriterien erfüllt sein:

- . 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.



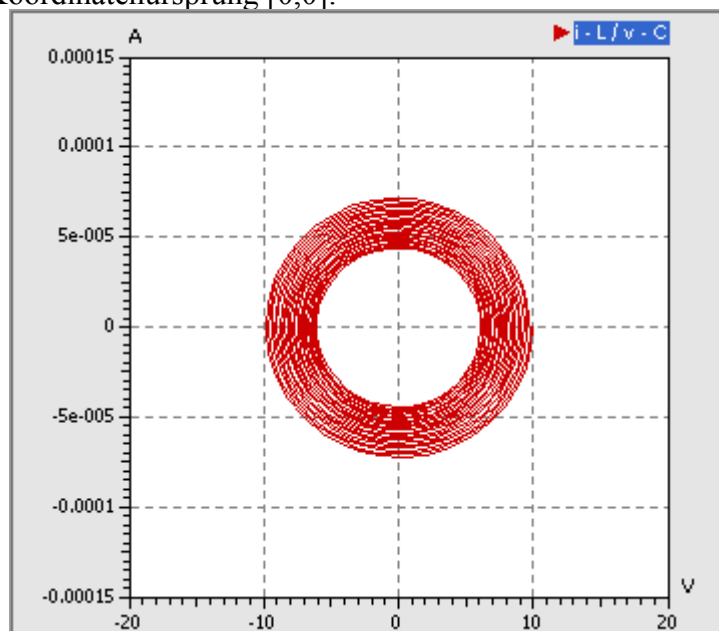
Linearer Schwingkreis

Wir beginnen mit dem Aufbau des Serienschwingkreises (Bild rechts) aus der durch einen Gyator 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:

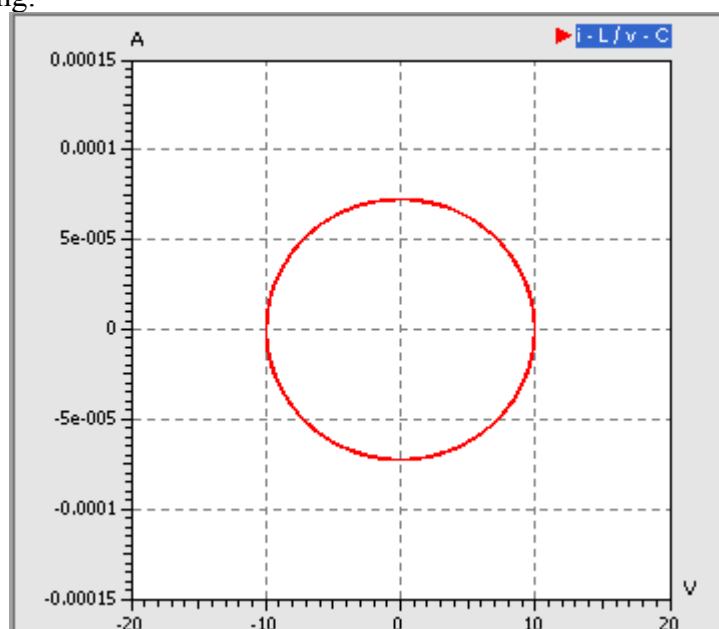
- . **C = 530 pF**
- . **L = 10 H**
- . **R = 1000 Ω**

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

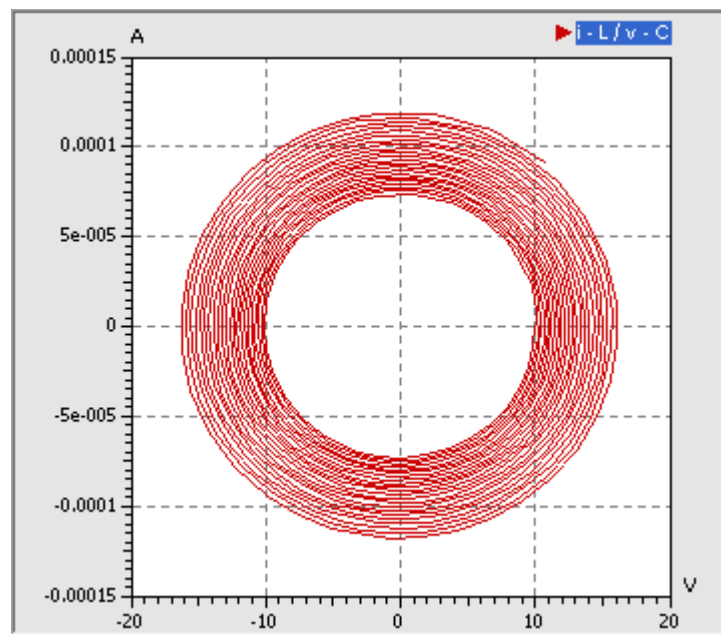
Im Phasenraum widerspiegelt sich die durch den Serienwiderstand gedämpfte Schwingung als Spiral-Trajektorie in Richtung Koordinatenursprung $[0,0]$:



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.

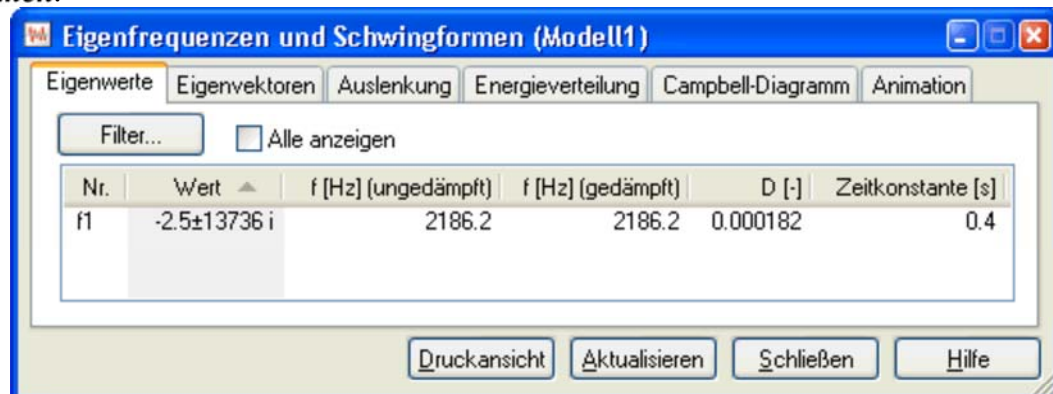
Merke:

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:

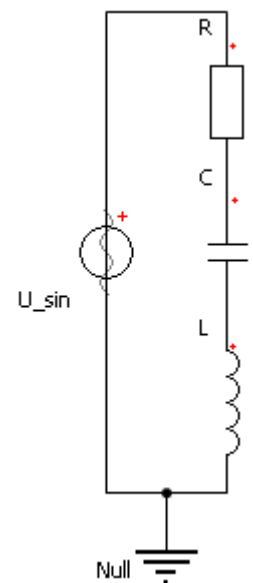
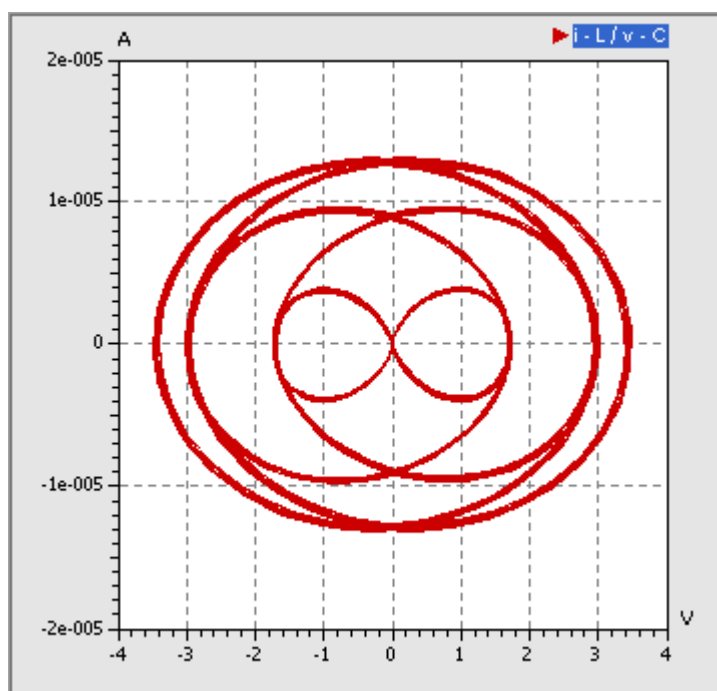
Wie groß ist die erforderliche Induktivität von L für den vorgegeben Wert $C=530 \text{ pF}$, damit eine Resonanzfrequenz $f_0=1400 \text{ Hz}$ entsteht. Dieser Wert ist mittels der Analyse der Eigenfrequenzen zu überprüfen.

Den ermittelten Induktivitätswert behalten wir für die weiteren Experimente bei:

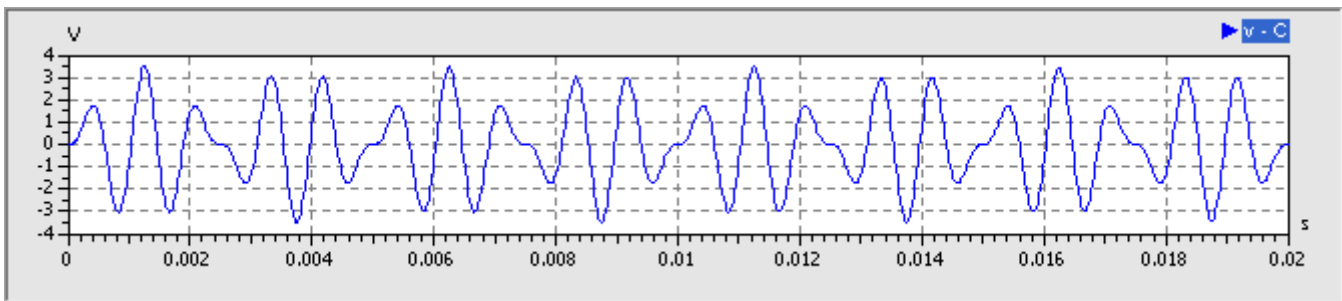
- Man kann davon ausgehen, dass im realen Experiment auch dieser Wert verwendet wurde, da sowohl die Frequenzmessung, als auch der Kapazitätswert des Drehkondensator nur mit geringen Toleranzen behaftet sind.
- Wahrscheinlich wurden am Zeiger der Gyrator-Induktivität 10 H gewählt und diesem Wert wurde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

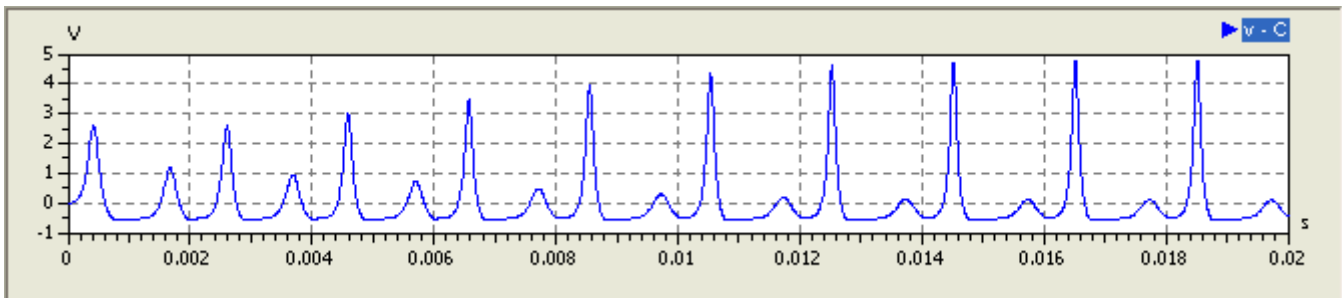
Im nächsten Schritt ergänzen wir laut Anleitung die Diode **D** parallel zum Drehkondensator:

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

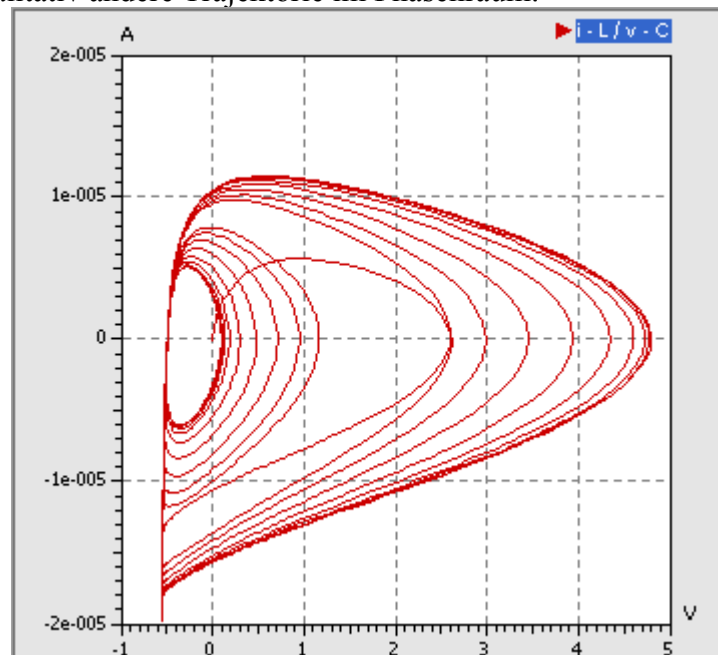
Die Diode erhält die von uns ermittelten Parameter **cj0** und **M**, welche die Kapazitätskennlinie des Diodentyps **BB 512** möglichst gut nachbilden.

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).

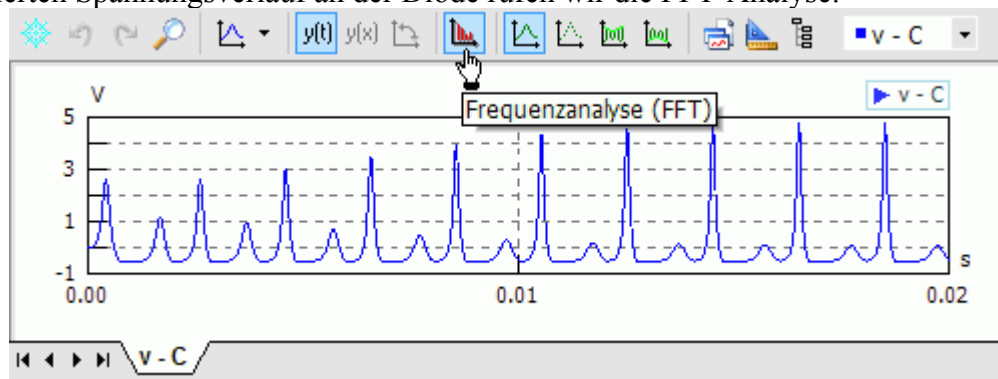
Ein beliebiges zeitliches Signal lässt sich immer als Summe von Sinusschwingungen unterschiedlicher Frequenz hinreichend genau abbilden (Reihenentwicklung der Fourier-Analyse).

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

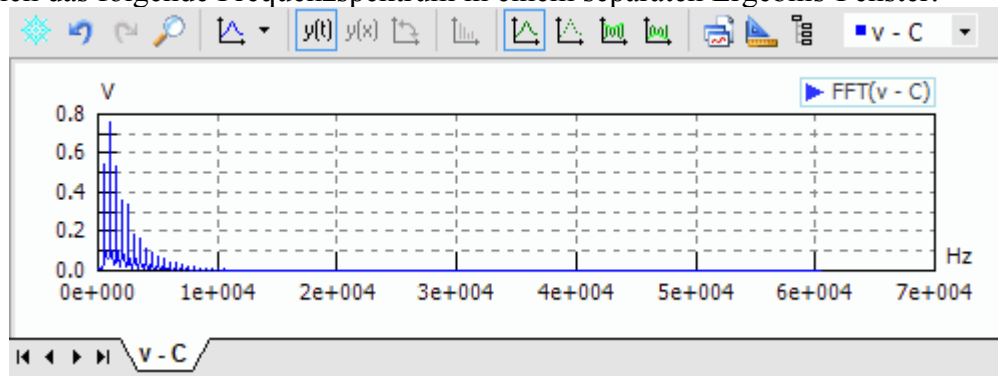
Bisher haben wir bei der Aufzeichnung der Signale mit variablen Schrittweiten gearbeitet.

Das geht im Prinzip auch, weil SimulationX daraus nach Abfrage einer maximalen Frequenz eine automatische Überführung in ein Signal hinreichend kleiner konstanter Zeitschrittweite vornimmt.

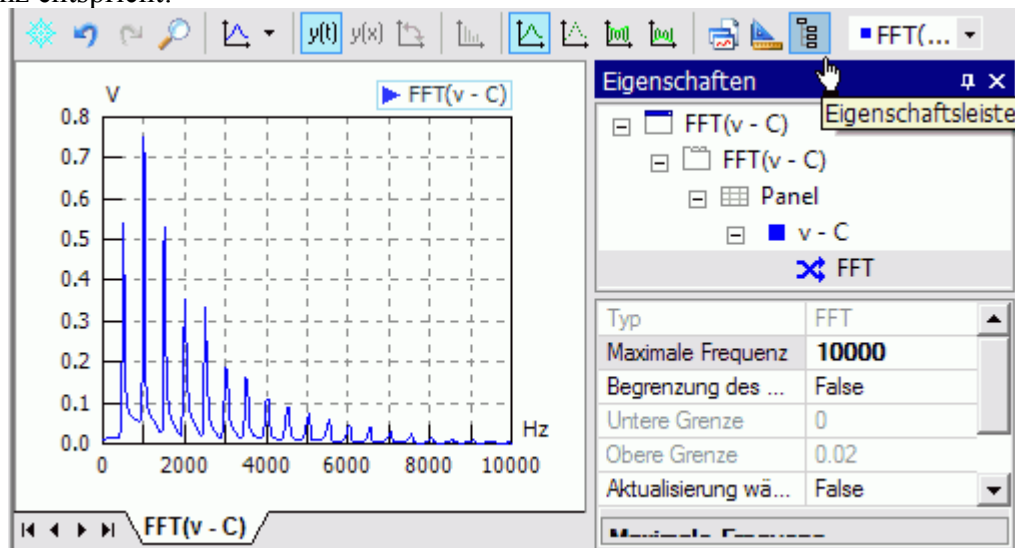
Für den simulierten Spannungsverlauf an der Diode rufen wir die FFT-Analyse:



Damit ergibt sich das folgende Frequenzspektrum in einem separaten Ergebnis-Fenster:



Wir begrenzen über die Eigenschaftsleiste die Berechnung des Spektrums auf 10000 Hz, was der 10-fachen Erregerfrequenz entspricht:



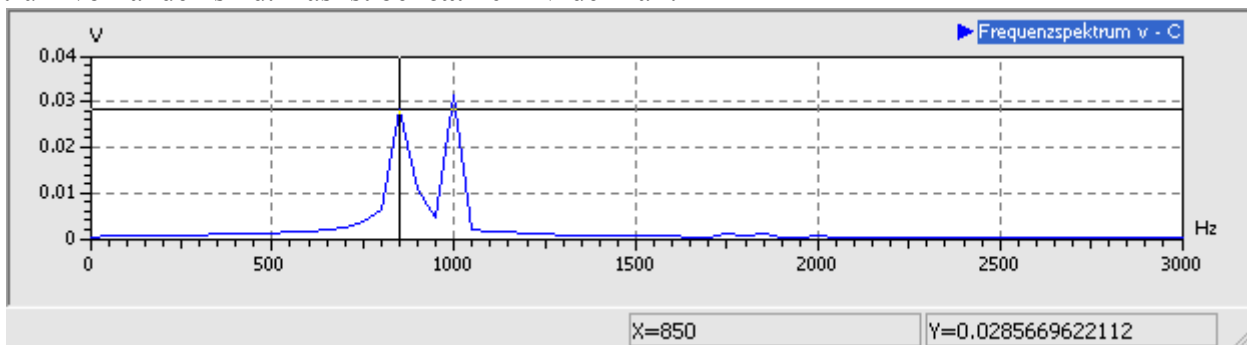
- Man erkennt die Erregerfrequenz von 1000 Hz als Maximalanteil im Signal, gefolgt von der halben Frequenz.
- Alle Frequenzanteile sind Vielfache von 500 Hz. Der Anteil höherer Frequenzen im Spektrum sinkt stetig, das Spektrum endet praktisch bei ca. 9 kHz.
- Die Abtastschrittweite des Signalverlaufs 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 μs . Nutzt man dies als gleichförmige Ausgabeschrittweite, so ergeben sich jedoch sehr eckige Signalverläufe!
- Mit einer **äquidistanten Protokollierung** der Ergebnisse unter Verwendung von **dtProtMin=10 μs** sollte man auf der sicheren Seite sein (Konfiguration in der **Simulationssteuerung**).

Hinweis: Leider erfolgt bei konstanter Signalschrittweite keine Abfrage zur maximalen Frequenz. Das Spektrum wird bis zur maximal möglichen Frequenz berechnet und dargestellt. Man muss dann manuell in dem Spektralfenster durch Konfiguration der X-Achse die Darstellung auf einen sinnvollen Bereich eingrenzen!

Konfiguration der Anregung

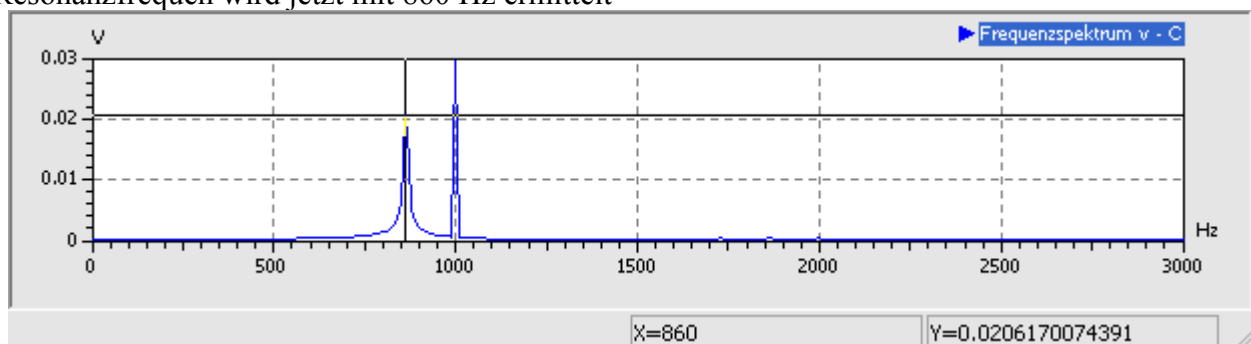
Die Vielzahl von Frequenzanteilen resultiert aus dem Kappen einer Halbwelle, wenn die Durchlassspannung der "Gleichrichter"-Diode überschritten wird. Es entstehen "Knicke" im Spannungsverlauf, welche hohen Frequenzanteilen entsprechen.

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:

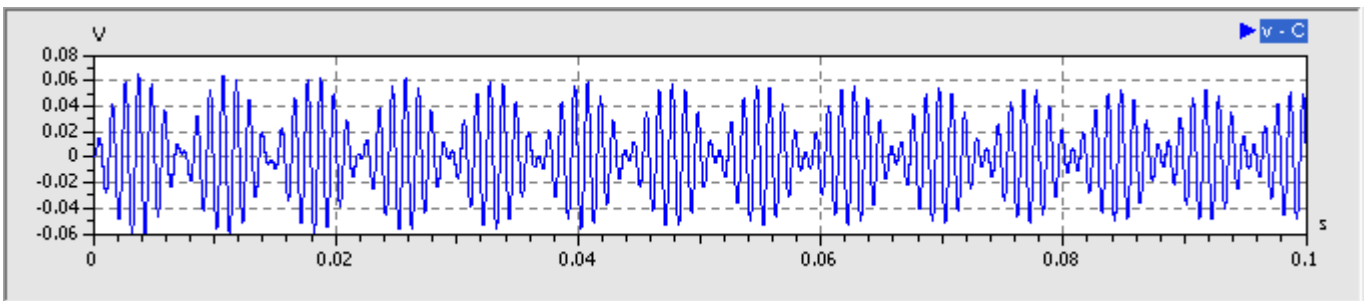


Zusätzlich zur Erregerfrequenz von 1 kHz erscheint die Resonanzfrequenz von 850 Hz. Diese ergibt sich aus der Sperrschicht-Kapazität, die sich bei sehr kleinen Sperrspannungen einstellt und welche ungefähr **Cj0** entspricht:

- 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 schmalere Frequenz-Peaks. Die Resonanzfrequenz wird jetzt mit 860 Hz ermittelt



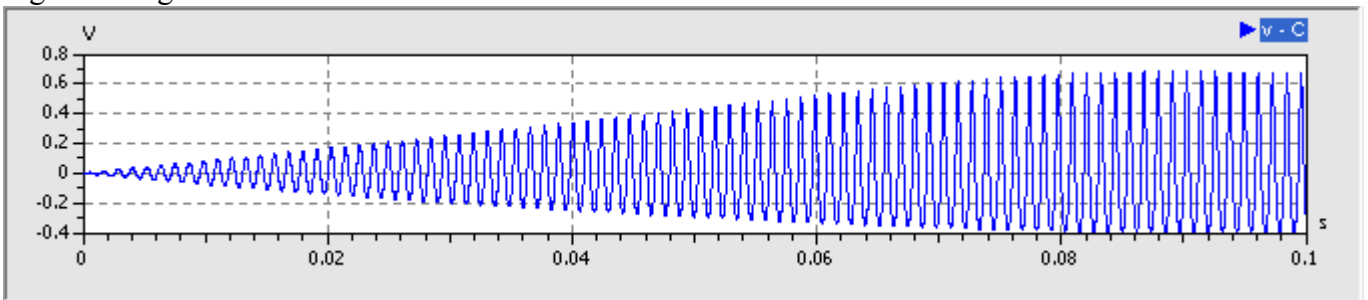
- Der zugehörige Zeitverlauf der Diodenspannung zeigt die Schwebung von 140 Hz zwischen Erreger- und Resonanzfrequenz:



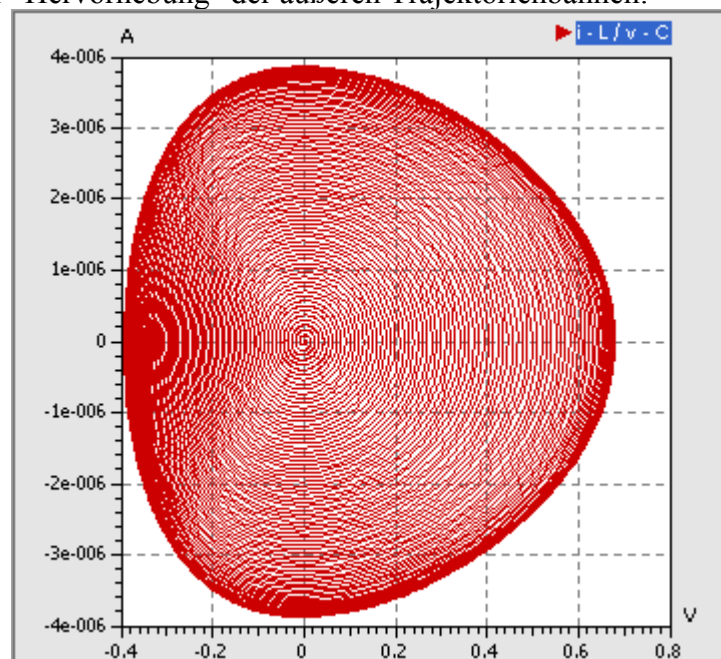
Wir erregen im Folgenden den Schwingkreis mit der zuletzt ermittelten Resonanzfrequenz von **860 Hz**:

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

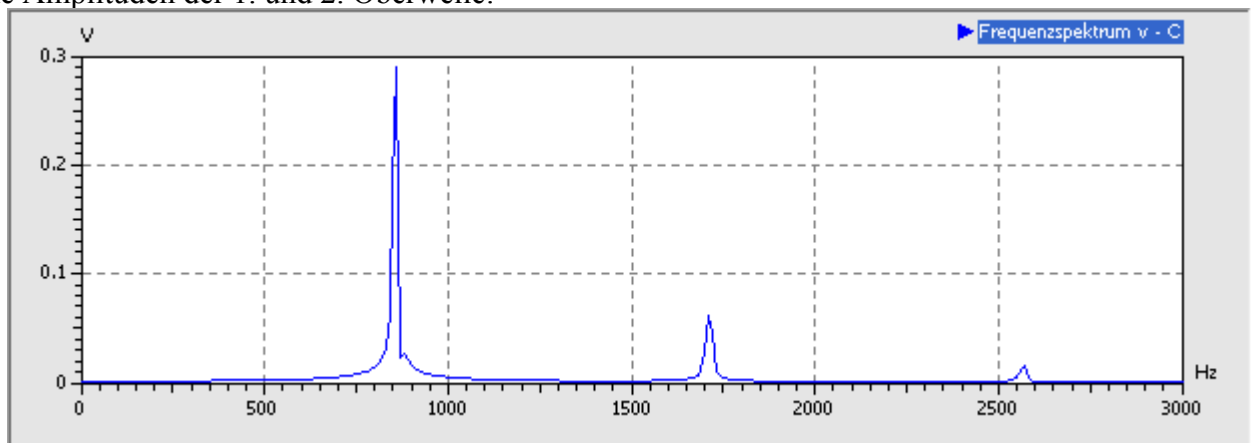
Wir verringern die Erregungsamplitude soweit, dass innerhalb von 100 ms ein "sanfter" Übergang in den eingeschwungenen Zustand stattfindet:



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:



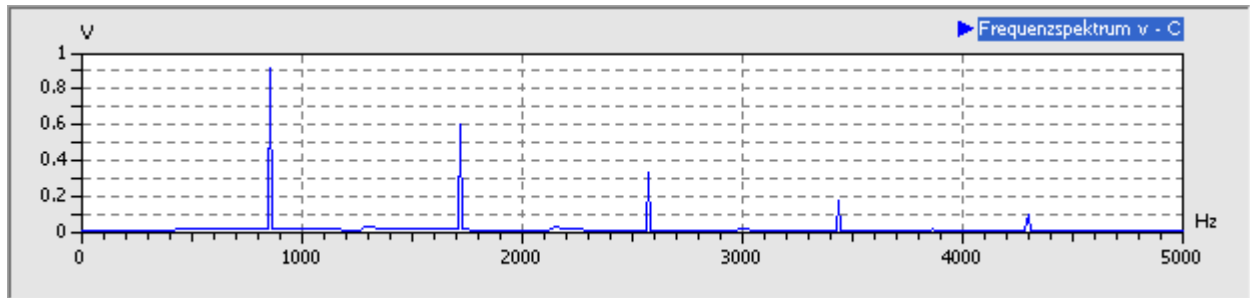
Die Oberwellen existieren, weil infolge der Amplitudenbegrenzung die Schwingung leicht von der Sinusform abweicht. Nur der reine Sinus besitzt keine Oberwellen.

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.

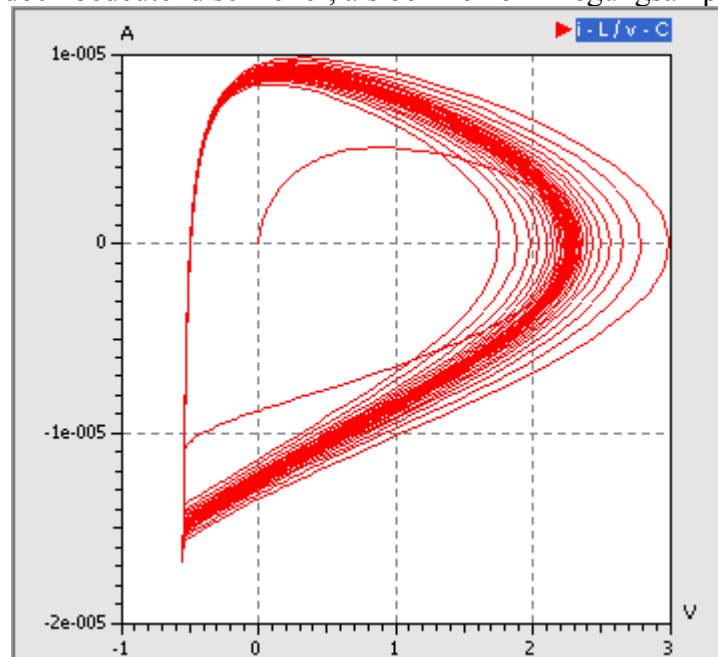
Einfluss der Anregungsamplitude

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:

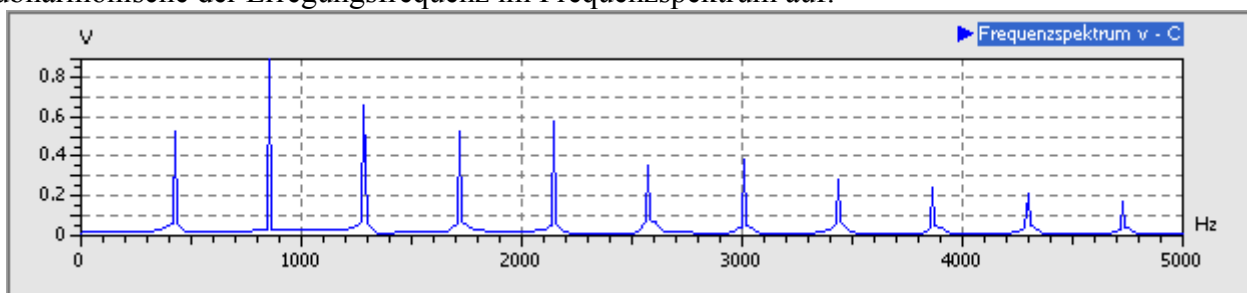
Das Verhalten des Schwingkreises ändert sich qualitativ nicht bis zu einer bestimmten Erreger-Amplitude. Im Beispiel lag dieser Wert bei ungefähr 1 V. Bis dahin nimmt nur die Amplitude der Oberwellen im Spektrum kontinuierlich zu:



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:

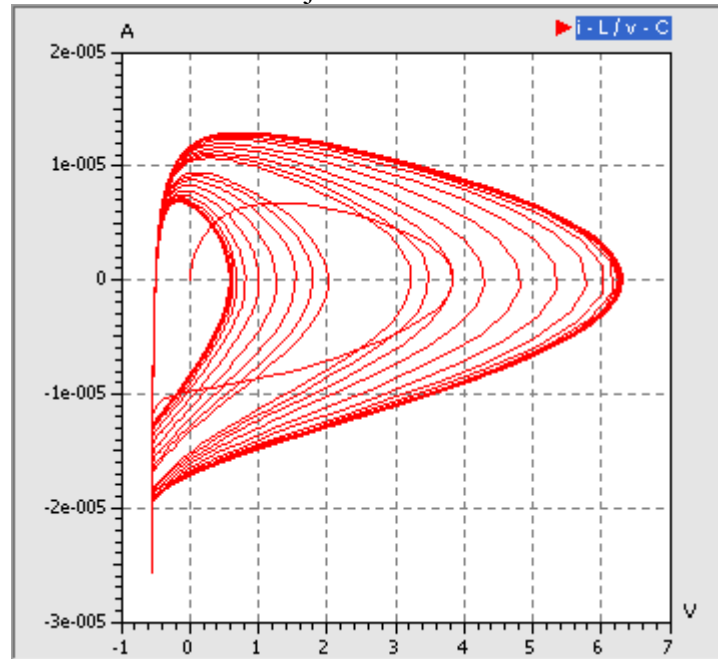


Eine neue Qualität des Verhaltens entsteht, indem bei einer weiteren Erhöhung der Erregungsamplitude plötzlich eine Perioden-Verdopplung stattfindet. Im Beispiel war dies bei 1.3 V Erregungsamplitude der Fall. Es tritt dann die 1. Subharmonische der Erregerfrequenz im Frequenzspektrum auf:

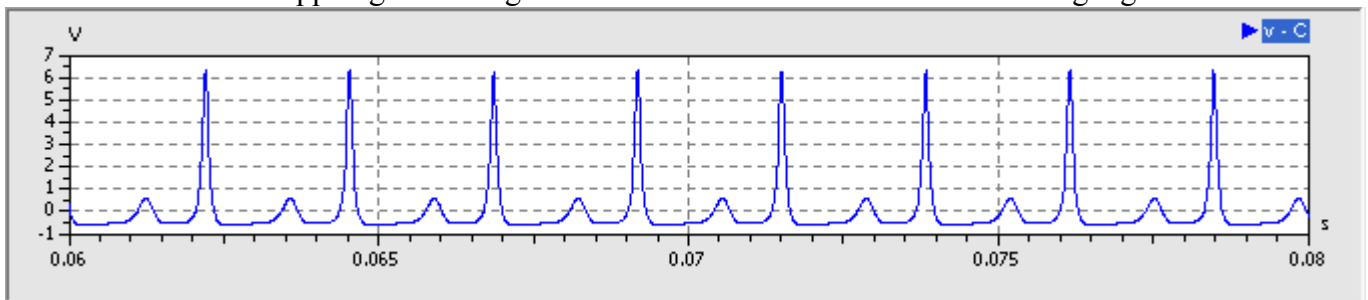


Von dieser 1. Subharmonischen leiten sich zusätzliche Obertöne im Frequenzspektrum ab. Das Spektrum erstreckt sich bis über 10 kHz (im Bild beschnitten).

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.

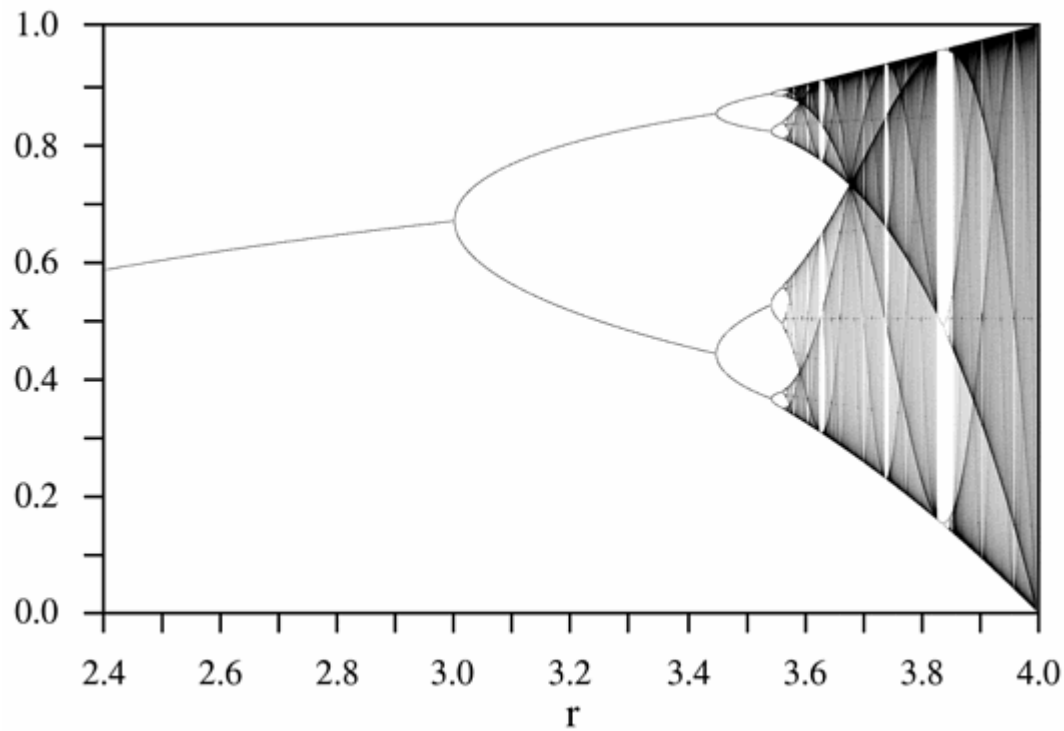
Bifurkationsdiagramm

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

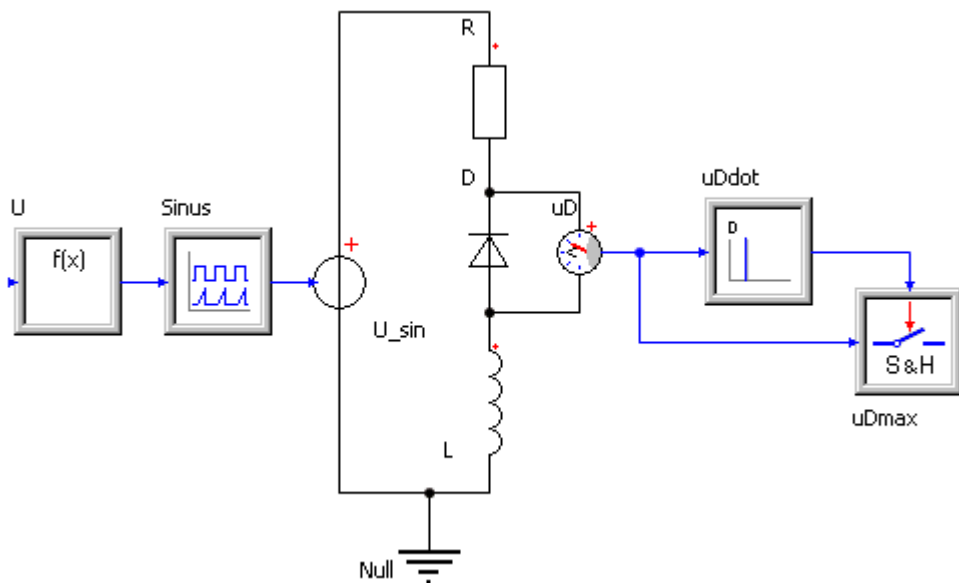
Die X-Achse wird mit dem veränderliche Parameter belegt. Das sollte bei uns die Amplitude der anregenden Sinus-Spannung sein.

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:



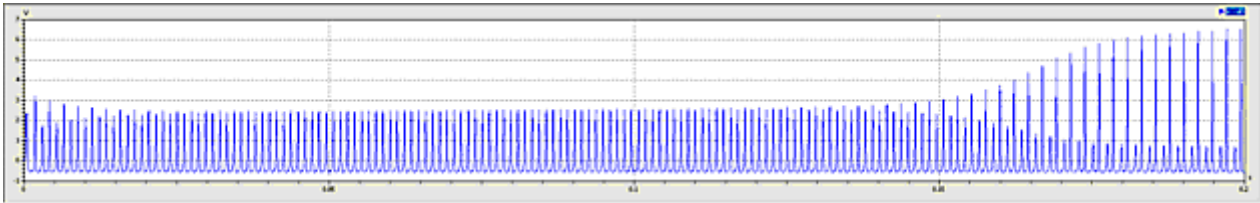
Hinweis: Die Parallelkapazität zur Diode von 10 pF musste entfernt werden, weil es ansonsten zu numerischen Problemen mit dem ereignisgesteuerten Abtastglied kommt. Das Verhalten ändert sich dadurch praktisch nicht!

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

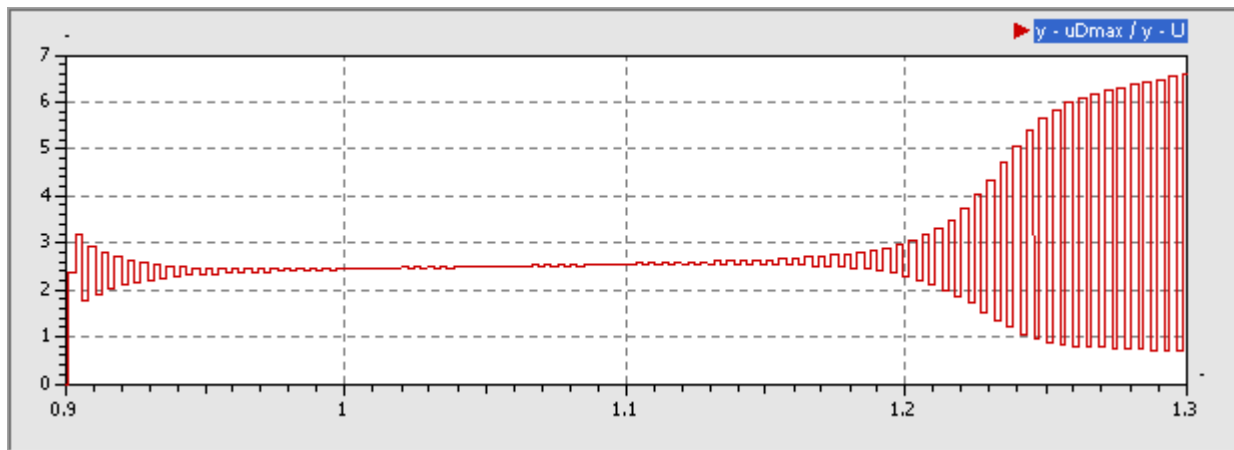
```
0.9+0.4*(t-tStart)/(tStop-tStart)
```

Damit der Schwingkreis genügend Zeit hat, der Spannungsänderung zu folgen, erhöhen wir die Simulationszeit auf **tStop=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:

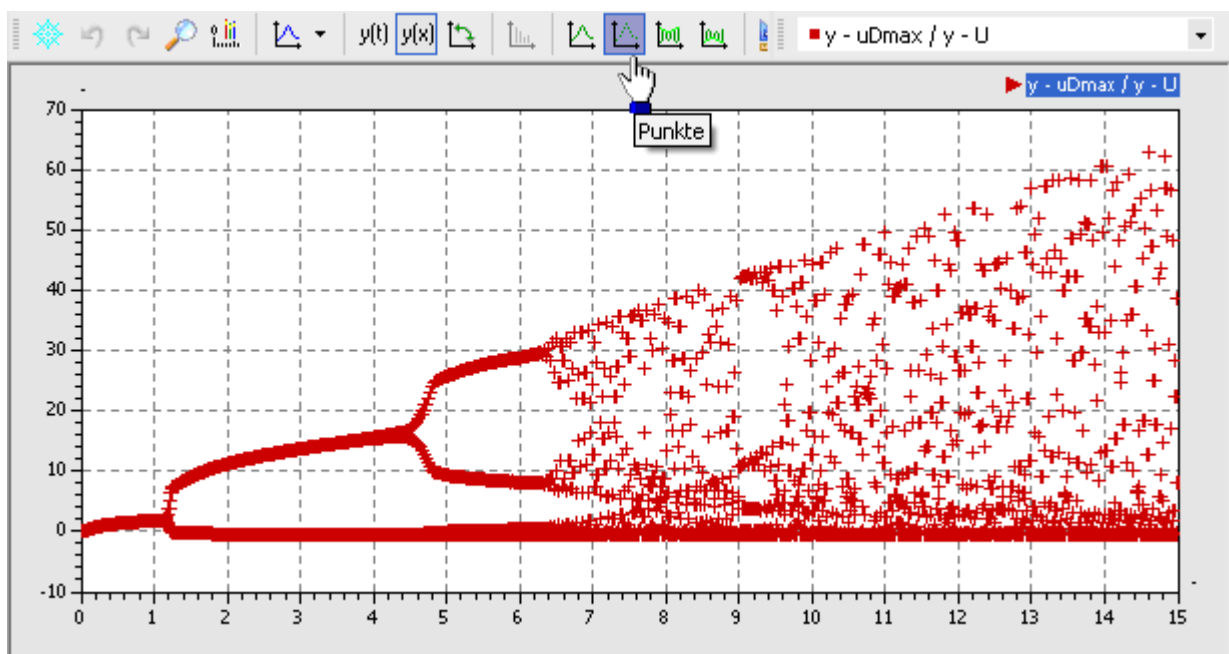


- Am Anfang erkennt man den Einschwingvorgang infolge des Einschaltens der Sinusspannung von 0,9 V. Für den eingeschwungenen Betrieb bei 0,9 V ist jedoch der eine Grenzwert typisch.
- Erst nach dem Überschreiten von 1,2 V kommt es zum Aufspalten in zwei Grenzwerte.

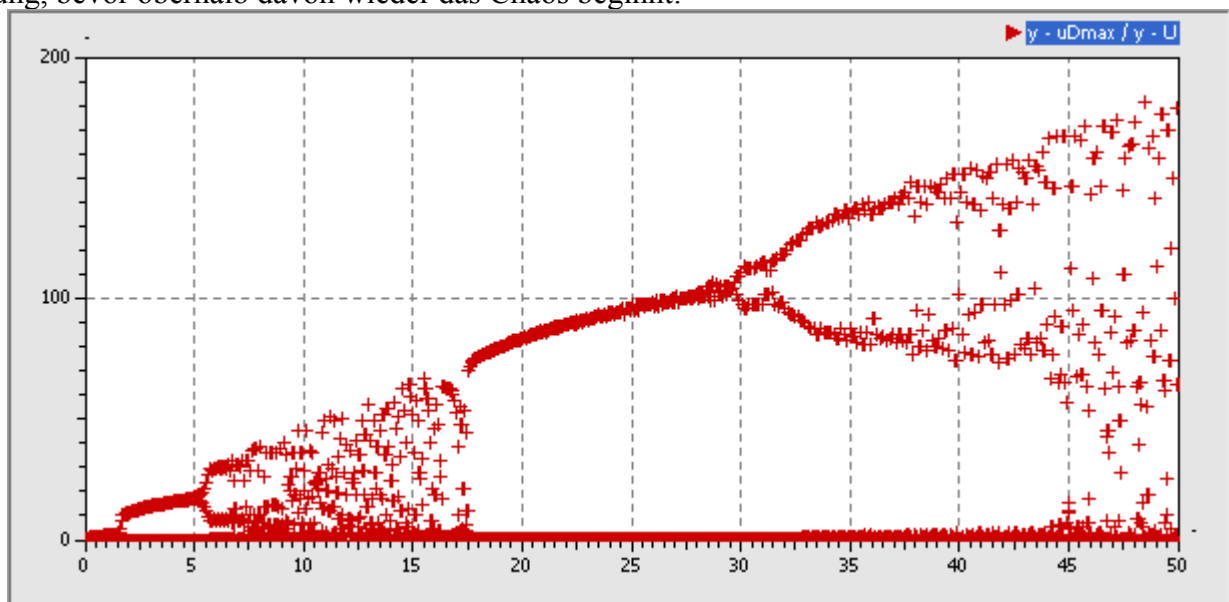
Der Weg zum perfekten Chaos

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:

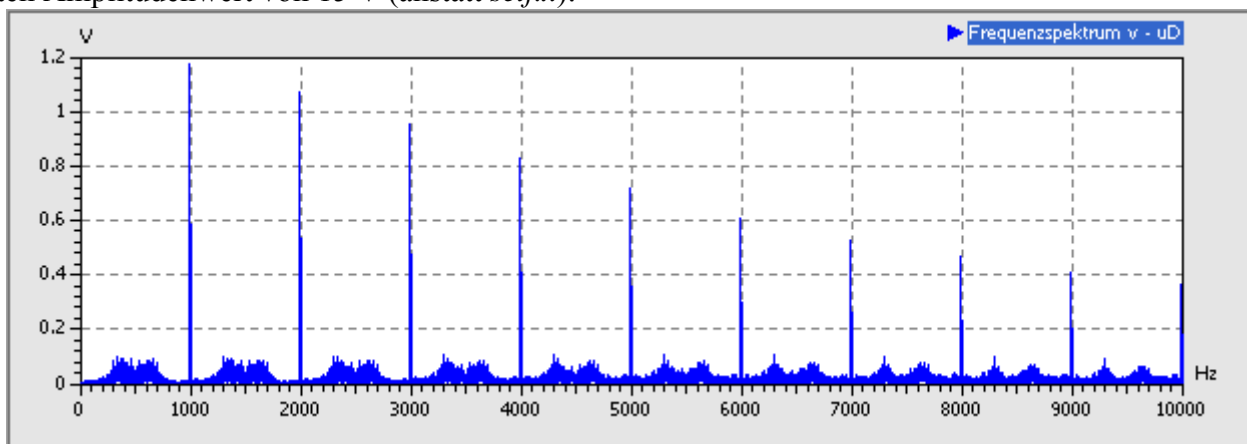
- Beim Durchfahren der Anregungsamplitude von nahe Null bis zu einem Maximalwert müssen hinreichend viele Periodenverdopplungen stattfinden, damit Chaos entsteht. In der Experiment-Anleitung wurde von 4 Periodenverdopplungen gesprochen. Leider wird dies mit der eingestellten Erregerfrequenz wahrscheinlich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



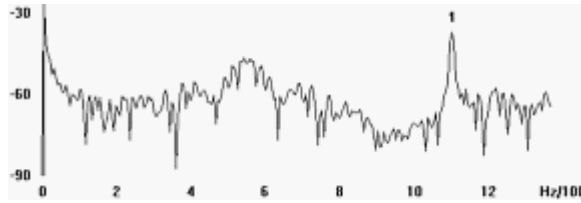
Unterhalb der Anregungsfrequenz und zwischen ihren Harmonischen ist nun im Spektrum Rauschen sichtbar, d.h. alle Frequenzen sind vertreten.

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