Spice-Modell für eine Glühlampe

Die Modellierung temperaturabhängiger Widerstände ist keine einfache Arbeit. Neben der eigentlichen Temperaturabhängigkeit des Widerstandes ist im Modell auch noch der Effekt der Eigenerwärmung und die mit der Wärmekapazität und Abstrahlung einhergehende Zeitkonstante zu berücksichtigen. Hierzu wurden für NTC schon zahlreiche Modelle vorgestellt. Als Spezialfall eines PTC ist die Glühlampe zu sehen. Der Widerstandsverlauf ist wesentlich weniger steil als bei einem Halbleiter-PTC, jedoch gelten dieselben Gesichtspunkte zur Modellierung.
Treibende Kraft für diese Arbeit war ein Artikel in der Zeitschrift EDN [EDN] zur Modellierung einer Glühlampe. Weitere Recherchen zeigten die Arbeit von [MEA88] der Firma Intusoft als Grundlage dieses Artikels. Leider konnte das dort gezeigte Modell nicht wunschgemäss zum Funktionieren gebracht werden. Trotzdem wurden die grundsätzliche Idee der dort gezeigten Lösung für die nachfolgende Arbeit verwendet.

Ziel der nachfolgenden Abschnitte ist die Entwicklung eines parametrisierbaren Modells für eine Glühlampe mit Wolframfaden. Die Parameter des Modells sind die Lampenspannung V_L und die nominale Leistung P_L.

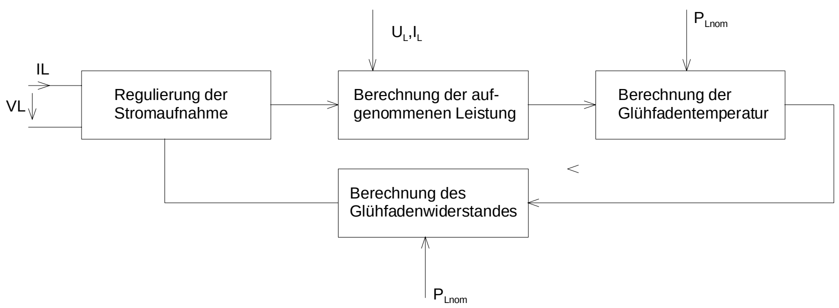


Bild 1: Prinzip des Modells für eine Glühfadenlampe mit Funktionsblöcken.

Die grundsätzliche Arbeitsweise wird mit Funktionsblöcken in Bild 1 dargestellt. Der Strom I_L wird durch den temperaturabhängigen Widerstand des Glühfadens reguliert. Aus der anliegenden Spannung V_L und dem aufgenommenen Strom I_L wird die aktuell aufgenommene Leistung berechnet ($P_L = V_L \cdot I_L$). Die Temperatur des Glühfadens wird bezüglich der Nominalleistung und Fadentemperatur von 2900°K berechnet. Aus der Temperatur wird über eine polynomiale Funktion der Fadenwiderstand bestimmt und für die Einstellung des Stromes I_L verwendet.

1. Erstellen des Modells

In einem ersten Schritt werden die notwendigen Funktionsblöcke auf Schaltplanebene zusammengestellt und getestet. Anschliessend erfolgt eine Umsetzung in ein Modell als textuell beschriebene parametrisierte Subcircuit. Das Erstellen des zugehörigen Editorsymbols mit Template, gehört auch zur Arbeit, wird hier nicht näher beschrieben. Die Kenntnis, wie Symbole und Modelle erstellt werden, wird für die nachfolgenden Ausführungen vorausgesetzt.

Im Modell selbst sind folgende Berechnungen auszuführen:

Gegeben (konstant): V_L, P_{Lnom}

Gemessen: I_L

Zu berechnen: $P_L = V_L \cdot I_L = I_L^2 \cdot R_L$ (1.1)

$$V_L = R_L \cdot I_L \quad (1.2)$$

Andere Ansätze, z.B. anstatt den Widerstand R_L den Leitwert G_L zu verwenden, sind auch möglich. Diese Form ist aber interessant, weil ausschliesslich Produkte zu berechnen sind. Diese können direkt mit gesteuerten POLY(2) Quellen von SPICE berechnet werden. In anderen Fällen müsste ein Quotient gebildet werden. Das wäre zwar mit einem ABM-Block in Pspice auch möglich, jedoch wäre das Modell weniger portabel.

2. Berechnung der Leistung

Die Leistung kann gemäss $P_L = V_L \cdot I_L$ bestimmt werden. Dazu eignet sich die Beschaltung:

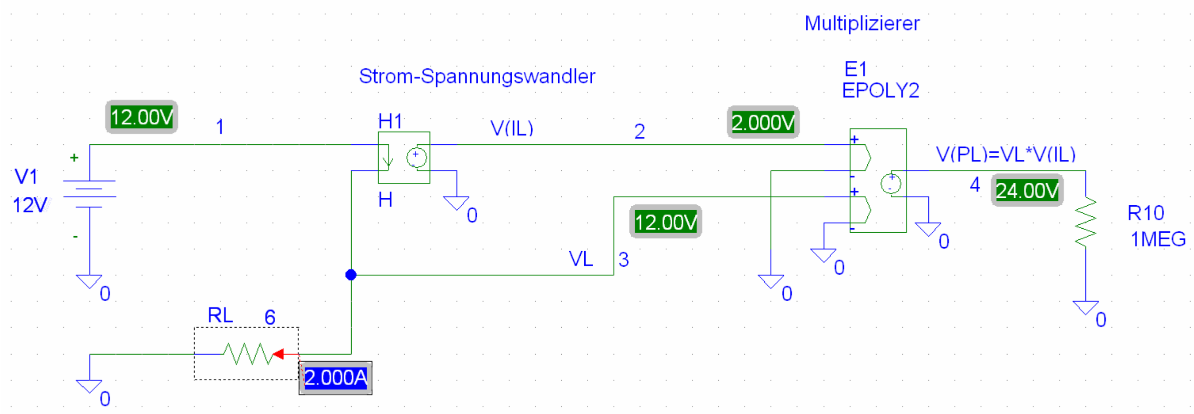


Bild 2: Berechnung der Leistung, hier gezeigt am Beispiel einer Lampe mit P_L=24W, R_L=6Ω.

R_L verkörpert den zunächst noch konstanten Lampenwiderstand. Über einen Stromwandler wird eine Stromproportional Spannung gebildet und mit der Spannung V_L multipliziert. An E_1 steht nun im Knoten 4 eine leistungsproportionale Spannung zur Verfügung. Der Widerstand R_{10} ist nur aus simulationstechnischen Gründen vorhanden.

3. Bestimmung der Glühfadentemperatur

Die Glühfadentemperatur wird aus den Parametern der nominalen Lampenspannung V_L und Lampenleistung P_{Lnom} bestimmt. Bei Nennleistung wird nach [MEA88] die Glühfadentemperatur von 2600°C (2900°K) definiert. Die gesteuerte Stromquelle liefert einen leistungsproportionalen Strom. Dieser bildet über den Widerstand R_{L2} eine temperaturproportionale Spannung in Kelvin ab. Die Umgebungstemperatur wird mit der Versatzspannung V_2 modelliert.

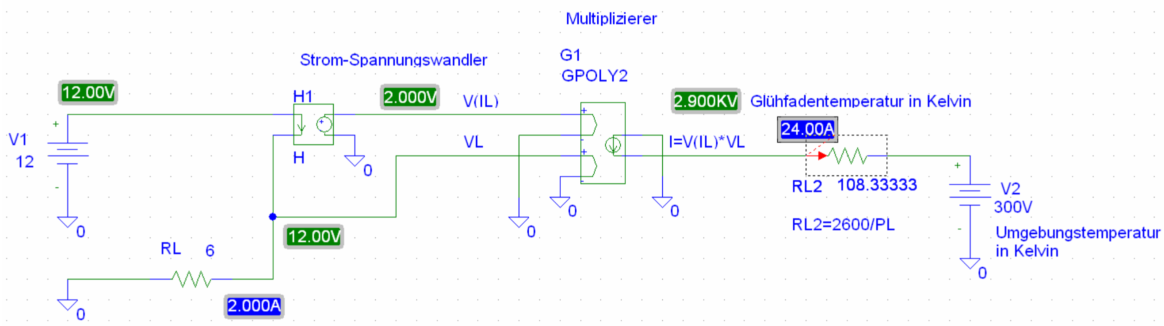


Bild 3: Berechnung der Glüfadentemperatur, hier gezeigt am Beispiel einer Lampe mit P_L=24W, R_L=6Ω.

4. Modellierung des spannungsgesteuerten Widerstandes

Der temperaturabhängige Widerstand des Glühfadens wird mit einer spannungsgesteuerten Spannungsquelle E_1 realisiert. Es gilt in der Masche $V_1 = E_1 = V_L \cdot I_L$. Der Widerstand R_{10} ist aus simulatortechnischen Gründen notwendig, hat aber auf das Resultat keinen Einfluss. Über die Spannung V_3 kann nun spannungsproportional der Widerstand R_L eingestellt werden und an G_1 erscheint der zugehörige Wert der Glühfadentemperatur.

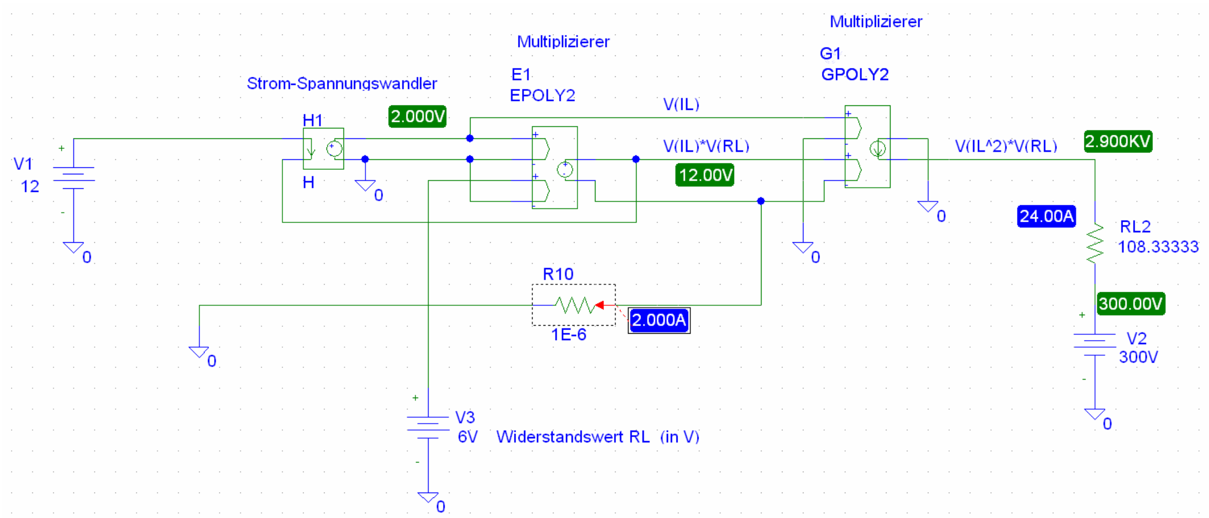


Bild 4: Einstellung des Lampenstromes mit dem Multiplizierer E1. Die Zahlenwerte beziehen sich auf eine Lampe mit P_L=24W, R_L=6Ω.

Abschliessend ist die Umsetzung der Glühfadentemperatur in den Glühfadenwiderstand zu bewerkstelligen. Bezugspunkt ist die Nennleistung P_{Lnom} der Glühlampe einhergehend mit einer Glühfaden-temperatur von 2900°K. Der Widerstandsverlauf ist leicht nichtlinear. Verschiedene Publikationen zeigen eine recht grosse Streuung der Werte. Sicherlich auch, weil teilweise auf sehr altes Material zurückgegriffen wurde.
Die in [DAV04] gezeigte kubische Näherung zeigt vor allem bei grossen Temperaturen eine erhebliche Abweichung als die tabellierten Werte in [ZER01] und [MEA88]. Ferner muss sichergestellt sein, dass die Ausgleichsfunktion ein hinreichend gutes Monotonieverhalten zeigt. Sonst ist keine Konvergenz gewährleistet.

5. Beschreibung des Widerstandsverlaufes

Zur Beschreibung der Ausgleichsfunktion wird das tabellierte Material aus [DAV04] verwendet. Zusätzlich werden die Werte für 0°K, 100°K, 200°K mittels linearer Extrapolation zugefügt. Sie bringen ein Gewicht für Temperaturen < 300°K so, dass keine negativen Resultate erzeugt werden. Da die Ausgleichfunktion einen grossen Wertebereich abdeckt, ist es sinnvoll, das Ausgleichpolynom auf der Basis der minimalen relativen Summe der Fehlerquadrate zu entwickeln [KRU96-IAM], anstatt nach der klassischen Methode der minimalen absoluten Fehlerquadrate. Das so gewonnene Ausgleichspolynom hat zwar eine etwas grössere absolute Summe der Fehlerquadrate. Für die Praxis ist aber bei kleineren Werten die Näherung wesentlich besser.

Grundsätzlich ist die polynomiale Ausgleichsfunktion als Polynomfunktion vom Grad n definiert:

$$y(x)=\sum_{i=0}^{n} a_i x^i = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n \qquad (1.3)$$

Diese Definition beinhaltet auch die lineare Regressionsfunktion wenn n=1 gesetzt wird.

Die Bestimmung der Koeffizienten $a_0,...,a_n$ erfolgt in bekannter Weise durch Minimieren der relativen Fehlerquadratsumme. Für eine Polynomfunktion vom Grad n lautet die relative Fehlerquadratsumme:

$$qr(a_0,a_1,\ldots,a_n)=\sum_{k=1}^{m}\left(\sum_{i=0}^{n}\frac{a_i x_k^i - y_k}{y_k}\right)^2 \qquad \begin{array}{l} m: \text{ Anzahl Datenpunkte} \\ n: \text{ Grad des Ausgleichspolynoms} \end{array} \qquad (1.4)$$

Die Koeffizienten a_i für eine minimale relative Fehlerquadratsumme werden über Nullsetzen der partiellen Ableitungen $\frac{\partial qr}{\partial a_i}$ für $i=0,..,n$ bestimmt.

$$\frac{\partial qr}{\partial a_i}=2\sum_{k=1}^{m} x_k^i\left(\sum_{i=0}^{n}\frac{a_i x_k^i - y_k}{y_k^2}\right)=0 \qquad (1.5)$$

Wir erhalten dann durch Auflösen der i-Summen die Normalgleichungen:

$$\sum_k \frac{x_k^i a_0}{y_k^2}+\sum_k \frac{x_k^{i+1} a_1}{y_k^2}+\cdots+\sum_k \frac{x_k^{i+j} a_j}{y_k^2}+\cdots+\sum_k \frac{x_k^{i+n} a_n}{y_k^2}=\sum_k \frac{x_k^i y_k}{y_k^2} \qquad i:0,...,n \qquad (1.6)$$

$$
\begin{aligned}
&a_0 \sum_k \frac{x_k^0}{y_k^2} + a_1 \sum_k \frac{x_k^1}{y_k^2} + \cdots + a_n \sum_k \frac{x_k^n}{y_k^2} = \sum_k \frac{1}{y_k} \\
&a_0 \sum_k \frac{x_k^1}{y_k^2} + a_1 \sum_k \frac{x_k^2}{y_k^2} + \cdots + a_n \sum_k \frac{x_k^{n+1}}{y_k^2} = \sum_k \frac{x_k}{y_k} \\
&\dots \\
&a_0 \sum_k \frac{x_k^n}{y_k^2} + a_1 \sum_k \frac{x_k^{n+1}}{y_k^2} + \cdots + a_n \sum_k \frac{x_k^{2n}}{y_k^2} = \sum_k \frac{x_k^n}{y_k}
\end{aligned}
\tag{1.7}
$$

Dies ist ein lineares Gleichungssystem mit *n+1* Bestimmungsgleichungen und *n+1* Unbekannten. Die Unbekannten stellen hier die Koeffizienten $a_0,\ldots,a_n$ dar.

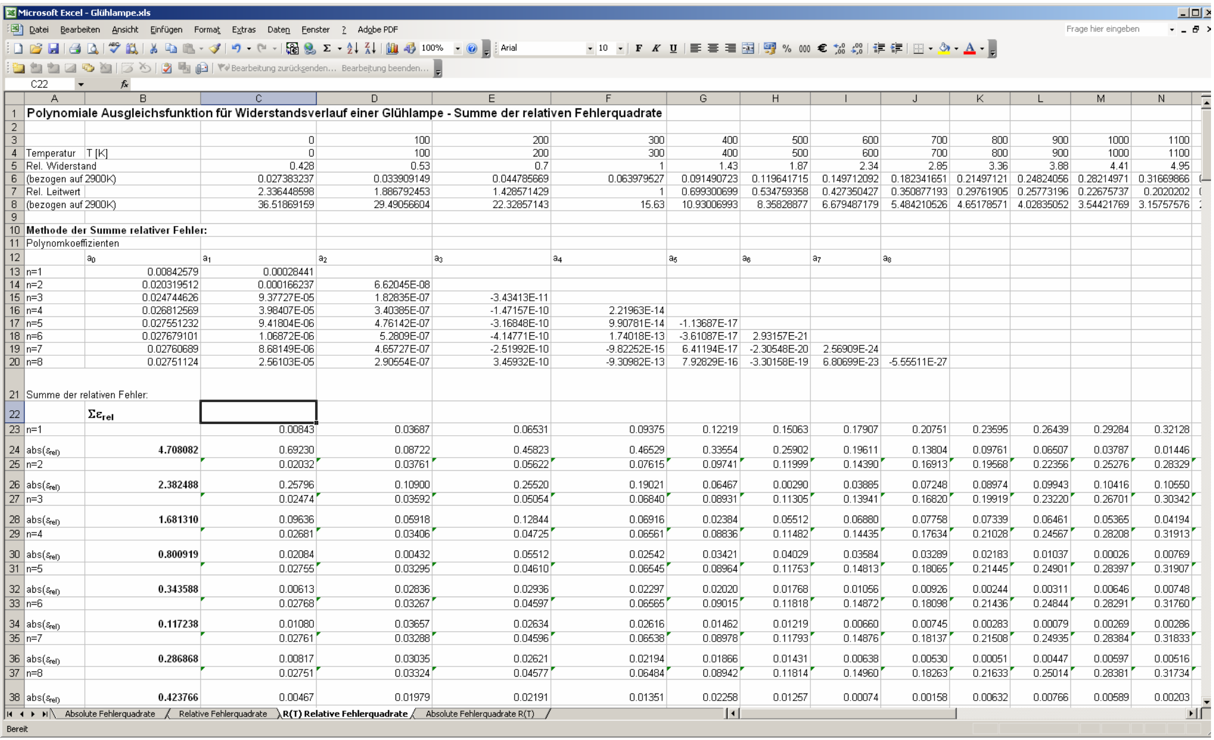
Die Koeffizientenmatrix G hat die Elemente: $$g_{ij} = \sum_k \frac{x_k^{i+j}}{y_k^2} \qquad i,j:0,\ldots,n \tag{1.8}$$

Die Elemente des Konstantenvektors $\underline{c}$ (rechte Seite des Systems): $$c_i = \sum_k \frac{x_k^i}{y_k} \qquad i:0,\ldots,n \tag{1.9}$$

Somit lautet das zu lösende Gleichungssystem:

$$G \cdot \underline{a} = \underline{c} \tag{1.10}$$

Dieses System wird mit den bekannten Verfahren (Matrizenrechnung) gelöst und wir erhalten als Resultat die gesuchten Polynomkoeffizienten. Die hier in EXCEL benutzte programmierte Lösung ist als VBA-Funktion im Anhang beigefügt.


Microsoft Excel - Glühlampe.xls

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Polynomiale Ausgleichsfunktion für Widerstandsverlauf einer Glühlampe - Summe der relativen Fehlerquadrate													
2														
3			0	100	200	300	400	500	600	700	800	900	1000	1100
4	Temperatur	T [K]	0	100	200	300	400	500	600	700	800	900	1000	1100
5	Rel. Widerstand		0.428	0.53	0.7	1	1.43	1.87	2.34	2.85	3.36	3.88	4.41	4.95
6	(bezogen auf 2900K)		0.027383237	0.033909149	0.044785669	0.063979527	0.091490723	0.119641715	0.149712092	0.182341651	0.21497121	0.24824056	0.28214971	0.31669866
7	Rel. Leitwert		2.336448598	1.886792453	1.428571429	1	0.699300699	0.534759358	0.427350427	0.350877193	0.29761905	0.25773196	0.22675737	0.2020202
8	(bezogen auf 2900K)		36.51869159	29.49056604	22.32857143	15.63	10.93006993	8.35828877	6.679487179	5.484210526	4.65178571	4.02835052	3.54421769	3.15757576
9														
10	**Methode der Summe relativer Fehler:**													
11	Polynomkoeffizienten													
12		a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8				
13	n=1	0.00842579	0.00028441											
14	n=2	0.020319512	0.000166237	6.62045E-08										
15	n=3	0.024744626	9.37727E-05	1.82835E-07	-3.43413E-11									
16	n=4	0.026812569	3.98407E-05	3.40385E-07	-1.47157E-10	2.21963E-14								
17	n=5	0.027551232	9.41804E-06	4.76142E-07	-3.16848E-10	9.90781E-14	-1.13687E-17							
18	n=6	0.027679101	1.06872E-06	5.2809E-07	-4.14771E-10	1.74018E-13	-3.61087E-17	2.93157E-21						
19	n=7	0.02760689	8.68149E-06	4.65727E-07	-2.51992E-10	-9.82252E-15	6.41194E-17	-2.30548E-20	2.56909E-24					
20	n=8	0.02751124	2.56103E-05	2.90554E-07	3.45932E-10	-9.30982E-13	7.92829E-16	-3.30158E-19	6.80699E-23	-5.55511E-27				
21	Summe der relativen Fehler:													
22		$\Sigma\varepsilon_{rel}$												
23	n=1		0.00843	0.03687	0.06531	0.09375	0.12219	0.15063	0.17907	0.20751	0.23595	0.26439	0.29284	0.32128
24	abs(ε_{rel})	**4.708082**	0.69230	0.08722	0.45823	0.46529	0.33554	0.25902	0.19611	0.13804	0.09761	0.06507	0.03787	0.01446
25	n=2		0.02032	0.03761	0.05622	0.07615	0.09741	0.11999	0.14390	0.16913	0.19568	0.22356	0.25276	0.28329
26	abs(ε_{rel})	**2.382488**	0.25796	0.10900	0.25520	0.19021	0.06467	0.00290	0.03885	0.07248	0.08974	0.09943	0.10416	0.10550
27	n=3		0.02474	0.03592	0.05054	0.06840	0.08931	0.11305	0.13941	0.16820	0.19919	0.23220	0.26701	0.30342
28	abs(ε_{rel})	**1.681310**	0.09636	0.05918	0.12844	0.06916	0.02384	0.05512	0.06880	0.07758	0.07339	0.06461	0.05365	0.04194
29	n=4		0.02681	0.03406	0.04725	0.06561	0.08836	0.11482	0.14435	0.17634	0.21028	0.24567	0.28208	0.31913
30	abs(ε_{rel})	**0.800919**	0.02084	0.00432	0.05512	0.02542	0.03421	0.04029	0.03584	0.03289	0.02183	0.01037	0.00026	0.00769
31	n=5		0.02755	0.03295	0.04610	0.06545	0.08964	0.11753	0.14813	0.18065	0.21445	0.24901	0.28397	0.31907
32	abs(ε_{rel})	**0.343588**	0.00613	0.02836	0.02936	0.02297	0.02020	0.01768	0.01056	0.00926	0.00244	0.00311	0.00646	0.00748
33	n=6		0.02768	0.03267	0.04597	0.06565	0.09015	0.11818	0.14872	0.18098	0.21436	0.24844	0.28291	0.31760
34	abs(ε_{rel})	**0.117238**	0.01080	0.03657	0.02634	0.02616	0.01462	0.01219	0.00660	0.00745	0.00283	0.00079	0.00269	0.00286
35	n=7		0.02761	0.03288	0.04596	0.06538	0.08978	0.11793	0.14876	0.18137	0.21508	0.24935	0.28384	0.31833
36	abs(ε_{rel})	**0.286868**	0.00817	0.03035	0.02621	0.02194	0.01866	0.01431	0.00638	0.00530	0.00051	0.00447	0.00597	0.00516
37	n=8		0.02751	0.03324	0.04577	0.06484	0.08942	0.11814	0.14960	0.18263	0.21633	0.25014	0.28381	0.31734
38	abs(ε_{rel})	**0.423766**	0.00467	0.01979	0.02191	0.01351	0.02258	0.01257	0.00074	0.00158	0.00632	0.00766	0.00589	0.00203

Absolute Fehlerquadrate / Relative Fehlerquadrate / R(T) Relative Fehlerquadrate / Absolute Fehlerquadrate R(T)

Bild 5: Entwicklung der polynomialen Ausgleichsfunktionen vom Grad 1 bis 8 für die Beschreibung des Widerstandsverlaufes in Abhängigkeit von der Temperatur.

Man erkennt, dass die Summe der relativen Fehlerquadrate mit wachsendem Polynomgrad bis 6 abnehmen, um dann wieder anzusteigen. Der Anstieg ist mit der Schwingneigung Polynome höheren Grades begründet. Je höher der Grad des Polynoms, desto ausgeprägter ist eine Oszillation im Werteverlauf zu beobachten. Deshalb sollte das Polynom kleinst-möglichen Grades verwendet werden, das die Anforderungen hinreichend erfüllt.

Aus Gründen der Konvergenzsicherheit werden nur Polynome geradzahligen Grades verwendet. Polynome ungeraden Grades haben immer einen Nulldurchgang und demzufolge einen Bereich negativer Funktionswerte.

Zur Wahl stehen daher die Polynome aus Bild 5:

$$p_4(T) = 0.026812569 + 3.98407 \cdot 10^{-5} T + 3.40385 \cdot 10^{-7} T^2 - 1.47157 \cdot 10^{-10} T^3 + 2.21963 \cdot 10^{-14} T^4 \tag{1.11}$$

$$p_6(T) = 0.027679101 + 1.06872 \cdot 10^{-6} T + 5.2809 \cdot 10^{-7} T^2 - 4.14771 \cdot 10^{-10} T^3 + 1.74018 \cdot 10^{-13} T^4 - 3.61087 \cdot 10^{-17} T^5 + 2.93157 \cdot 10^{-21} T^6 \tag{1.12}$$

Das Polynom 4. Grades hat einen maximalen relativen Fehler von 10.9% bei 3600°K. Das Polynom 6. Grades °inen Fehler von 3.7% bei 200°K. Die Wahl fällt auf das Polynom 4. Grades. Es beschreibt im Bereich von 300°K-3400°K den Verlauf mit einem maximalen Fehler von 4%.
Die Umsetzung der Temperatur, dargestellt als Spannungswert, erfolgt mit einer spannungsgesteuerten Spannungsquelle des Typs E POLY(1). Die Polynomkoeffizienten werden in aufsteigender Reihenfolge des Grades spezifiziert: (Bezogen auf Bild 6.)

```
E2  7 0 POLY(1) 4 0  0.026812569  3.98407E-05  3.40385E-07 -1.47157E-10  2.21963E-14
```

Eine weiterführende Beschreibung zu POLY(1) und POLY(2) Quellen sind in [HOE85] zu finden.

6. Modellierung der Zeitkonstanten

Die Zeitkonstante wird mit der Kapazität C_1 und R_{L2} realisiert. In der Arbeit von [MEA88] wird für den Kondensatorwert ohne nähere Begründung empfohlen:

$$C_1 = P_{Lnom} \cdot 1.57 \cdot 10^{-3} \frac{F}{W} \tag{1.13}$$

Die Zeitkonstante ist direkt von der Nominalleistung, einhergehend mit der Wärmekapazität, abhängig.
Anmerkung: Meines Erachtens ist dieser Wert eher etwas gross. Dies kann aber mit einer Messreihe am praktischen Objekt verifiziert und angepasst werden.

Für die Arbeitspunktbestimmung der Schaltung wird der Kondensator C_1 mit einem Vorgabewert versehen (Initial Condition, `IC`). Hier wird im Modell 300V (entsprechend 300°K) als Umgebungstemperatur vorgegeben. Im implementierten Simulatormodell wird die Vorgabetemperatur aus der SPICE-Systemvariablen `TEMP` festgelegt.

7. Gesamte Schaltung

Sie dient zur Verifikation des Modells auf Stufe Schaltplan. Eine Fehlersuche auf Stufe Textmodell ist wesentlich mühsamer.

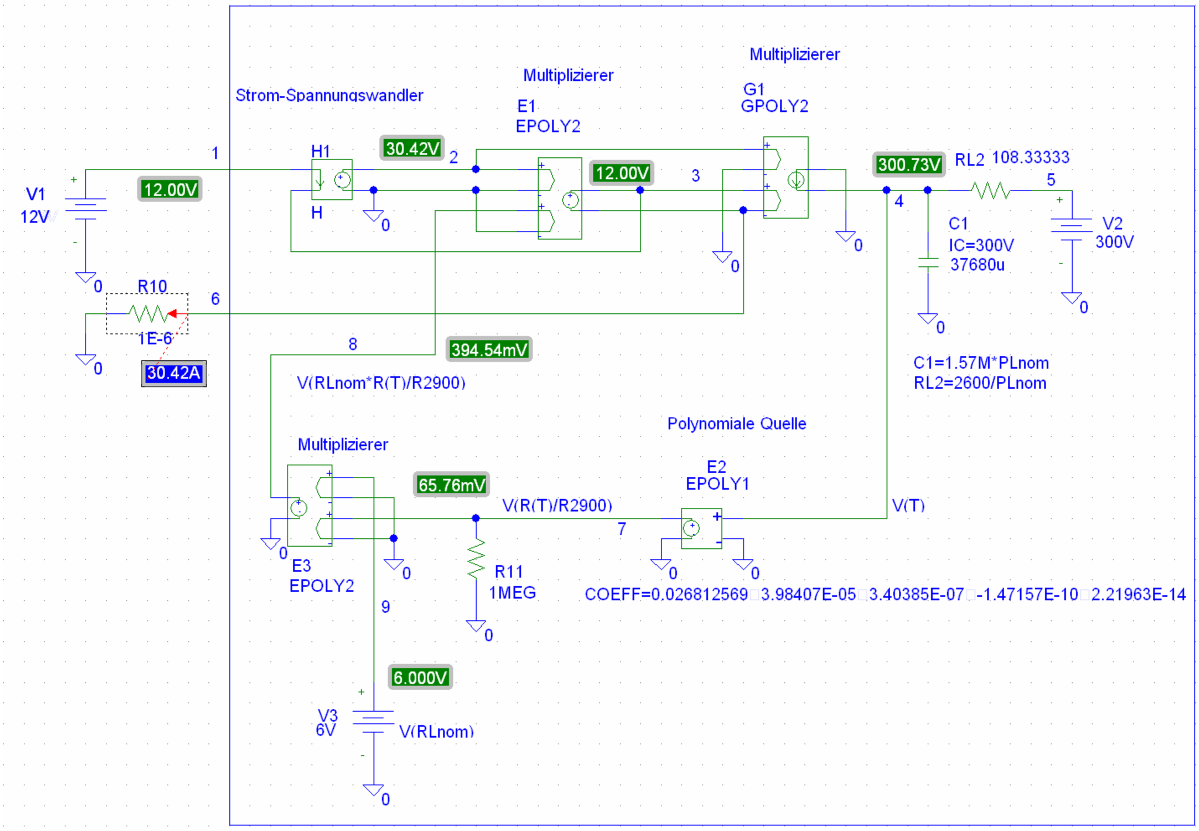


Bild 6: Vollständige Schaltung zur Simulation der Glühlampe. Die Zahlenwerte beziehen sich auf eine Lampe mit P_L=24W, R_L=6Ω. Die Zahlenwerte zeigen das Einschaltstverhalten bei T=300°K.

Anmerkung:
Auf den Multiplizierer E_3 könnte verzichtet werden, indem beim Multiplizierer E_1 die Multiplikation direkt mit dem konstanten Widerstandswert R_{Lnom} erfolgt.

8. Ableiten des Simulatormodells

Für die Umsetzung in ein parametrisiertes Simulationsmodell wird eine *Subcircuit* mit dem Namen `LAMP` gebildet:

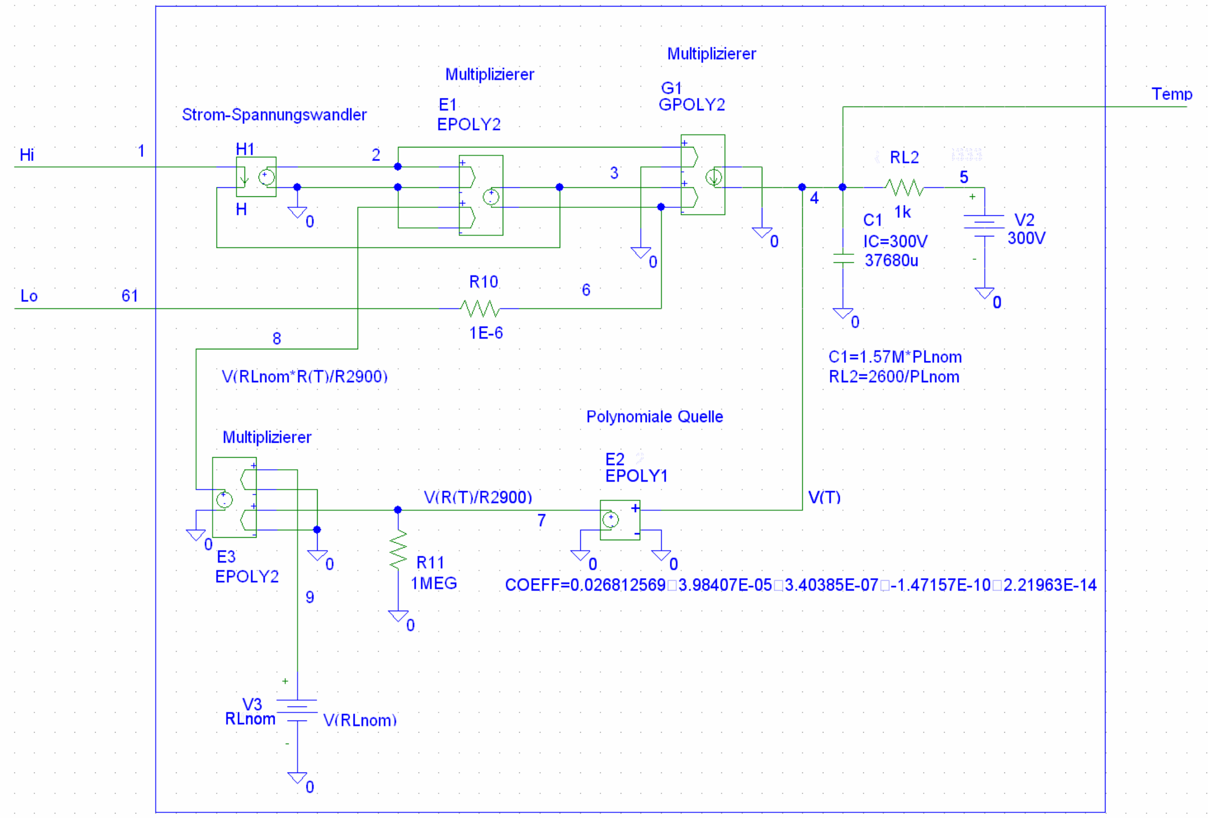


Bild 7: Die aus Bild 6 abgeleitete Teilschaltung (Subcircuit) für das Glühlampenmodell. Das implementierte Textmodell verwendet dieselben Knotennummern und Bezeichnungen.

Die Knoten `1` und `61` verkörpern die Anschlüsse der Glühlampe. Am Knoten 4 steht eine temperaturproportionale Spannung der Glühfadentemperatur zur Verfügung.
Die Werte für V_3, RL_2 und C_1 werden aus den Parametern V_{Lnom}, P_{Lnom} berechnet. Pspice verlangt für alle Parameter im Simulatormodell einen Standardwert, da im Simulatormodell keine Unterscheidung zwischen obligatorischen und fakultativen Parametern gemacht wird.

Die Subcircuit wird als Textfile definiert. Die stromgesteuerte Spannungsquelle `H1` benötigt aus simulationstechnischen Gründen eine Spannungsquelle, hier `VH1`.

```
*
* Tungsten lamp simulator model                                                   File: LAMP.LIB
* Author: Gerhard Krucker, Zaunackerstrasse 9, CH-3113 Rubigen
* 27-SEP-2004
*
*
.SUBCKT LAMP  1  61                  ; Hi   Lo  nodes
+ PARAMS: VLnom=12V  PLnom=0.1W      ; Mandatory parameters: Nominal lamp voltage and power
H1      2 0 VH1 1
VH1     1 3 0V                       ; Control voltage source is needed by H1
V2      5 0 300V                     ; Base temperature for polynomial approximation
RL2     4 5 {2600/{PLnom}}
V3      9 0 {{VLnom}*{VLnom}/{PLnom}}
E1      3 6 POLY(2)  2  0  8  0  0 0 0 0 1 0
G1      0 4 POLY(2)  2  0  3  6  0 0 0 0 1 0
C1      4 0 {1570u*{PLnom}} IC={{TEMP}+273} ; Use system temperature for OP
E3      8 0 POLY(2)  9  0  7  0  0 0 0 0 1 0
R11     0 7 1MEG
R10     61 6  1E-6
E2      7 0 POLY(1) 4 0  0.026812569  3.98407E-05  3.40385E-07  -1.47157E-10  2.21963E-14
.ENDS
```

Das Modell `LAMP2` hat zusätzlich den Knoten 4 herausgeführt. An diesem Anschluss kann die Glühfadentemperatur in K als Spannungswert abgegriffen werden. Dieser Anschluss darf aber nicht belastet werden, da sonst die Rechnung verfälscht wird.

9. Beispiele

Beispiel 1: Einschaltverhalten einer Glühlampe.
Das Einschaltverhalten zweier Glühlampen für 12V soll gezeigt werden. L_1 hat die Leistung P_L=24W, L_2 hat P_L=100W. An L_1 wird zusätzlich der Verlauf der Glühfadentemperatur erfasst.

Lösung:
Für die Simultion des Einschaltverhaltens wird das Modell `LAMP` verwendet. Die Parameter P,V werden nach Aufgabenstellung eingetragen. Soll zusätzlich die Temperatur erfasst werden, wird das Modell `LAMP2` verwendet. Am dritten Pin steht eine zur Glühfadentemperatur (in °K) proportionale Spannung zur Verfügung.

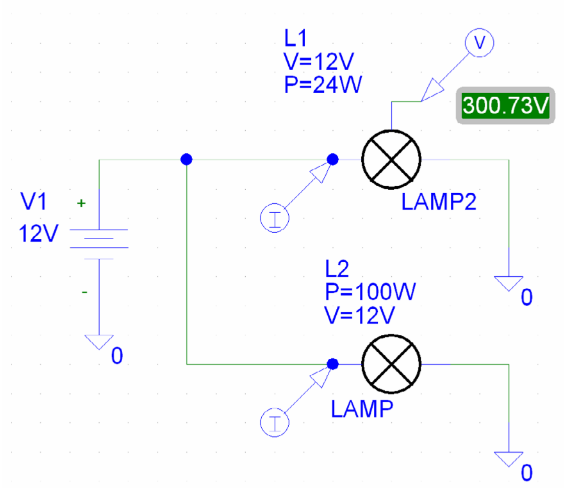


Bild 8: Schaltung zur Simulation des Einschaltverhaltnes zweier Glühlampen in Beispiel 1. Bei L_1 wird am dritten Pin zusätzlich die Glühfadentemperatur erfasst.

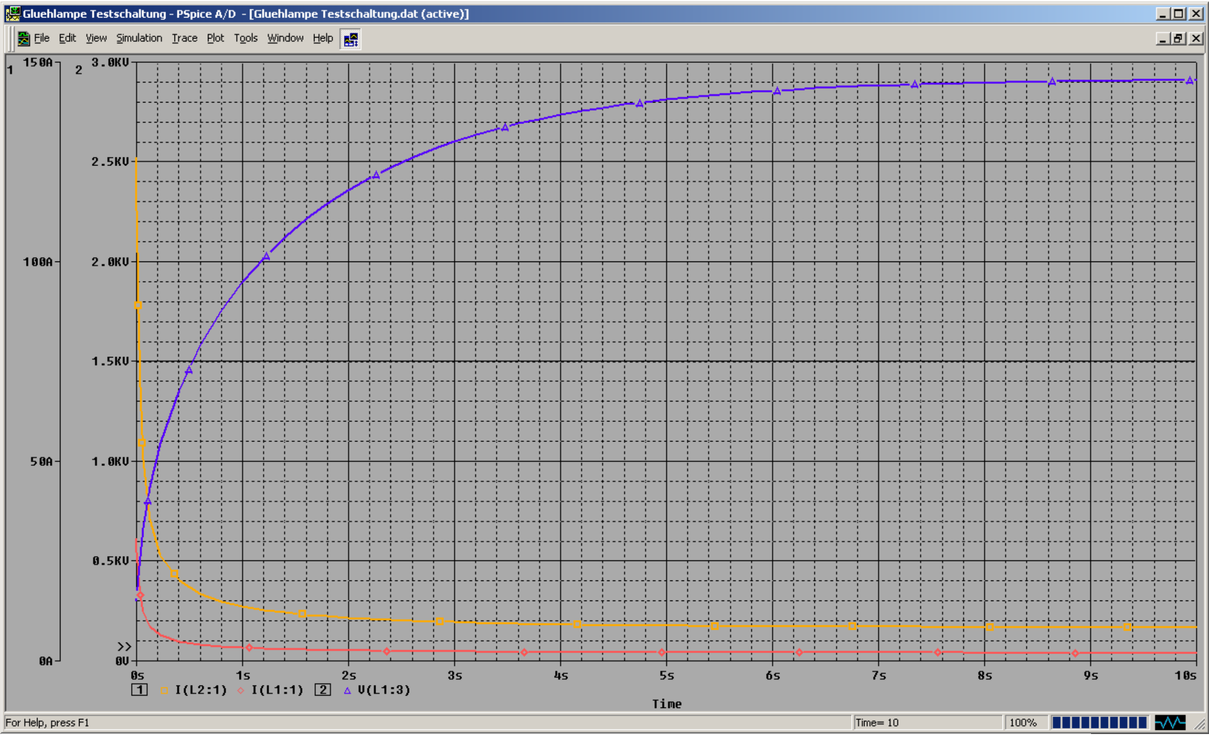


Bild 9: Einschaltstromverlauf der Glühlampen L_1, L_2 in Beispiel 1. Für L_1 wird zusätzlich der Temperaturverlauf aufgezeigt 1kV entspricht der Temperatur von 1000°K.

Beispiel 2: Stabilisierung eines Oszillators mit Wien-Brücke und Operationsverstärker LM741.
Zu realisieren ist die Stabilisierung eines Oszillators mit einer bereits dimensionierten Wien-Brücke für 1kHz nach Bild 10. Die Stabilisierung erfolgt mit einer Micro-Glühlampe mit den Daten 36V, 0.125W. Für den Betrieb ist eine Glühfadentemperatur von ca. 350°K vorzusehen.

Die Schaltung ist im Zeitbereich zu simulieren und die Ausgangsspannung nach einer Einschwingzeit von 1s für 10ms aufzuzeigen.

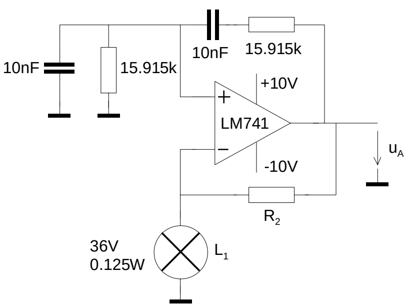


Bild 10: Oszillator mit Wien-Brücke für die Dimensionierung der Stabilisierung mit R_2-L_1 in Beispiel 2 .

Lösung:
Für den Betrieb mit einer Glühfadentemperatur von ca. 350°K beträgt der Widerstand des Glühfadens nach (1.11) etwa 7.5% des Warmwiderstandes. R_2 ist daher so zu dimensionieren, dass bei dieser Fadentemperatur eine Verstärkung von 3 erreicht wird. Vorgabegrössen sind die nominale Lampenleistung P_{Lnom} und Betriebspannung U_{Lnom}:

$$R_2 \approx 2\frac{0.075 \cdot U_{Lnom}^2}{P_{Lnom}} \tag{1.14}$$

Die Dimensionierung von R_2 erfolgt nach(1.14):

$$R_2 \approx 2\frac{0.075 \cdot U_{Lnom}^2}{P_{Lnom}} = 2\frac{0.075 \cdot 36}{0.125} = 1.5552k\Omega$$

Wir setzen den Normwert 1.5kΩ ein.
Um ein sicheres Anschwingen in der Simulation zu gewähleisten, wird eine Speisespannung um 1us zeitversetzt eingeschaltet.

Bemerkung:
Die Dimensionierung von R_2 ist kritisch. Er beinflusst die Schwingfähigkeit, Amplitude und den Klirrfaktor. R_2 wird in der Praxis einstellbar ausgelegt, so dass der Wert etwa um ± 20% variiert werden kann.

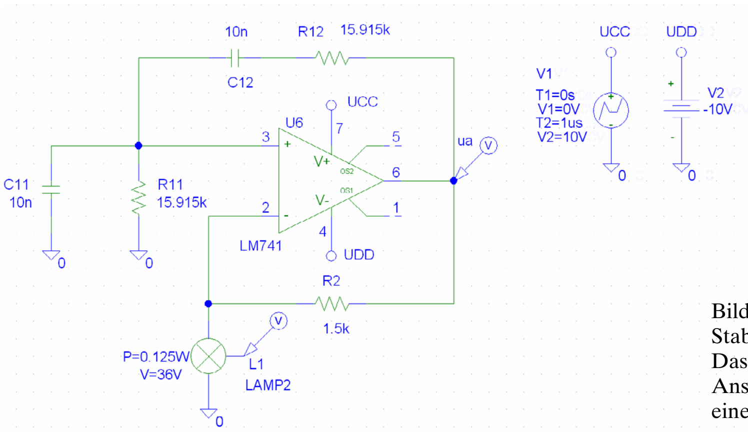


Bild 11: Schaltplan des Wien-Brücken-Oszillators mit PTC-Stabilisierung nach Beispiel 2.
Das hier verwendete Glühlampenmodell erlaubt am dritten Anschluss die Messung der Glühfadentemperatur in Form einer Spannung.

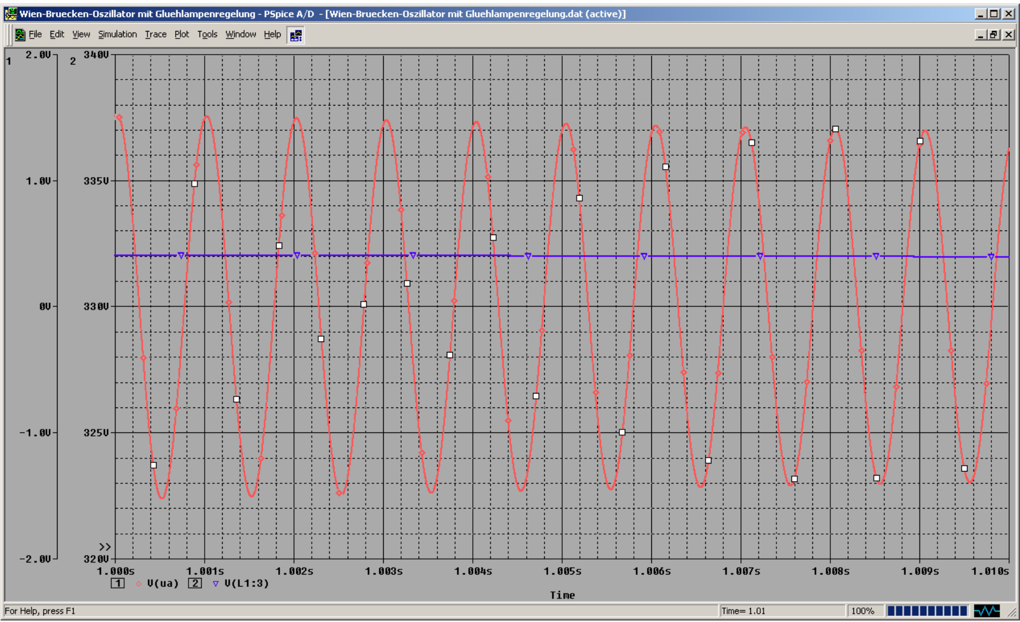


Bild 12: Simulation des Wien-Brücken-Oszillators mit PTC-Stabilisierung nach Beispiel 2.
Die Spannung V(L1:3) zeigt als Wert die Glühfadentemperatur in Kelvin. Man erkennt, dass die Amplitude der Ausgangsspannung immer noch leicht abnehmend, d.h. die Regelung ist auch nach einer Sekunde noch nicht eingeschwungen.

10. Ergänzende Bemerkungen zur Ausgabe vom 29.9.2004

Das Modell wurde mehrfach als PTC-Element zur Stabilisierung von Oszillatoren benutzt. Die gemachten Erfahrungen sind an sich gut, jedoch zeigten sich dabei aber folgende Mängel:

- Die Amplitudenstabilisierung mit diesem Modell neigt zu niederfrequenten Schwingungen (Motorboating). Das kann durch Vergrösserung der Kapazität C_1 vermindert werden.
- Die Strahlungsverluste sind im Modell nicht berücksichtigt.
- Die Wärmekapazität ist nicht genügend berücksichtigt und sollte parametrisierbar sein.

Im Rahmen einer möglichen Erweiterung wird dies vermutlich aufgearbeitet. Für diesbezügliche Hinweise oder konkrete Lösungen bin ich dankbar.

11. Literaturreferenzen

[DAV04] J. W. Davis (US) and S. Fabritsiev (RF).
Pure Tungsten - Electrical Resistivity. ITER Material Properties Handbook.
University of California, San Diego.
http://www-ferp.ucsd.edu/LIB/PROPS/ITER/AM01/AM01-3201.html

[EDN] EDN.com: The design source for electronics engineers and managers,
SPICE models tungsten lamp, DI #1305Z Larry Meares, Intusoft, San Pedro, CA
http://www.reed-electronics.com/ednmag/index.asp?layout=siteInfo&doc_id=31899

[HOE85] SPICE – Analyseprogramm für elektronische Schaltungen, E.E.E. Hoefer/
H. Nielinger, Springer Verlag 1985, ISBN 3-540-15160-5

[KRU96-IAM] Informatik und angewandte Mathematik, Kapitel 3 -
Ausgleichs- und Interpolationsrechnung, Skript zur Vorlesung mit Übungen,
Gerhard Krucker, Hochschule für Technik und Architektur Bern, Herbst 2000.

[MEA88] A Tungsten Lamp Model - Intusoft Newsletter October 1988,
Larry Meares 1988.
http://www.intusoft.com/nlpdf/nl11.pdf

[ZER01] Zerda, T.W. Stefan Boltzmann Law. Texas Christian University. 2001.
http://personal.tcu.edu/~zerda/manual/lab22.htm

Anhang

VBA-Funktion zur Bestimmung des Ausgleichspolynoms

Die hier gezeigte Funktion berechnet das Ausgleichspolynom *n*-ten Grades für eine beliebige Menge von Datenpunkten *m*>1, wobei *m*>=*n*.

```
'Berechnen der Polynomkoeffizienten a0,..,an für eine polynomiale Ausgleichsfunktion
' n-ten Grades für m Datenpunkte nach der Methode der kleinsten Summe der relativen Fehlerquadrate.
' Parameter: x = Array mit x-Werten (Anzahl: m, beliebig)
'            y = Array mit y-Werten (Anzahl: m, beliebig)
'            n = Grad des zu erzeugenden Ausgleichspolynoms
'
' Das Resultat wird als Funktionswert (Arrayfunktion) retourniert
' Autor: Gerhard Krucker
' Datum: 17.8.1995, 22. 9. 1996, 24.9.2004
' Sprache: VBA for EXCEL7, EXCEL XP
'
Function PolynomRegRel(x, y, Polynomgrad)
Dim AnzX, AnzY     ' Anzahl x- und y-Werte
Dim m              ' Anzahl auszugleichender Datenpunkte'
Dim Sxk()          ' Dynamisches Array fuer die Summe der Potenzen von xk '
Dim Sxkyk()        ' Dynamisches Array fuer die Summe der Potenzen von xk * yk '
Dim G(), g1        ' Dynamisches Array fuer die Koeffizientenmatrix G'
Dim c()            ' Dynamisches Array fuer den Konstantenvektor c'
Dim a()            ' Dynamisches Array fuer die Polynomkoeffizienten a0,..,an'
Dim i, j, k

  ' Parameterkontrollen'
  If (Polynomgrad < 1) And Polynomgrad <> "Integer" Then
     MsgBox "Polynomgrad muss eine Ganzzahl >= 1 sein!"
     PolynomRegRel = CVErr(xlErrValue)
     Exit Function
  End If
  AnzX = x.Count  ' Anzahl Datenpunkte in den Arrays bestimmen '
  AnzY = y.Count
  If (AnzX <> AnzY) Then
     MsgBox "Anzahl x-Werte und Anzahl y-Werte muss gleich gross sein"
     PolynomRegRel = CVErr(xlErrValue)
     Exit Function
  End If
  If AnzX <= Polynomgrad Then
     MsgBox "Anzahl Datenpunkte muss > Polynomgrad sein!"
     PolynomRegRel = CVErr(xlErrValue)
     Exit Function
  End If
  m = AnzX

  'Summe der Potenzen xk/yk^2 und xk*yk berechnen und den entprechende Arrays abspeichern'
  ReDim Sxk(Polynomgrad * 2)    ' Arrays auf passende Groesse dimensionieren '
  ReDim Sxkyk(Polynomgrad)      ' Die Arrayindizes laufen von 0..Polynomgrad, resp 0..2*Polynomgrad'
  For i = 0 To 2 * Polynomgrad
     Sxk(i) = 0
     For k = 1 To m            ' Fuer jeden Datenpunkt'
       Sxk(i) = Sxk(i) + x(k) ^ i / y(k) ^ 2
     Next k
  Next i
  For i = 0 To Polynomgrad
     Sxkyk(i) = 0
     For k = 1 To m
       Sxkyk(i) = Sxkyk(i) + x(k) ^ i / y(k)
     Next k
  Next i

  'Koeffizientenmatrix G und Konstantenvektor c erzeugen'
  ReDim G(1 To Polynomgrad + 1, 1 To Polynomgrad + 1)  'Matrix mit Indizes 0..Polynomgrad,0..Polynomgrad dimensionieren'
  ReDim g1(1 To Polynomgrad + 1, 1 To Polynomgrad + 1) 'Matrix fuer die Inverse von G (MINV kann nicht in G
zurueckschreiben)'
  ReDim c(1 To Polynomgrad + 1)
  ReDim a(0 To Polynomgrad)                            'Polynomkoeffizienten a0,..,an (a(0) = a0) '

  For i = 0 To Polynomgrad                             'Koeffizientenmatrix G und Konstantenvektor c aufbauen '
     For j = 0 To i
        G(i + 1, j + 1) = Sxk(i + j)
        G(j + 1, i + 1) = Sxk(i + j)
     Next j
     c(i + 1) = Sxkyk(i)
  Next i

  ' Gleichungssystem G * a = c loesen mit Matrixinversion'
  g1 = Application.MInverse(G)             'Koeffizientenmatrix G invertieren'
  For i = 1 To Polynomgrad + 1      'Matrixmultiplikation a = G1 * c'
    a(i - 1) = 0
    For j = 1 To Polynomgrad + 1
      a(i - 1) = a(i - 1) + g1(i, j) * c(j)
    Next j
  Next i

  PolynomRegRel = a   'Koeffizientenvektor a0,..,an retournieren'

End Function
```

Die Funktion `PolynomRegRel` ist eine sog. *Arrayfunktion* . Der Wert der ersten Resultatzelle kann direkt mit dem Funktionsassistenten oder per Handeingabe erfolgen. Die Eingabe wird normal abgeschlossen und wir erhalten in der Zelle den Koeffizienten a_0. Nun wird das Feld für die Koeffizienten a_0,..,a_n durch Selektieren der nachfolgenden Zellen aufgezogen. Dann wird die Eingabezeile mit der Formel oberhalb des Arbeitsblattes angeklickt und die Zeile mit <CTL><Shift><RET> abgeschlossen. Die restlichen Koeffizienten werden jetzt in die markierten Felder übertragen. Die Formel im Zellenblock wird nachher in geschweiften Klammern dargestellt. Dies zeigt an, dass eine Arrayfunktion die Zellinhalte definiert.
Ein Editieren im zellenblock ist noch möglich, jedoch muss die Eingabe immer mit CTL><Shift><RET> in der Editorzeile unterhalb des Menüs abgeschlossen werden.
Erfolgt das CTL><Shift><RET> in der Zelle selbst, werden die restlichen Koeffizienten nicht angezeigt oder es wird ein Fehler ausgegeben.