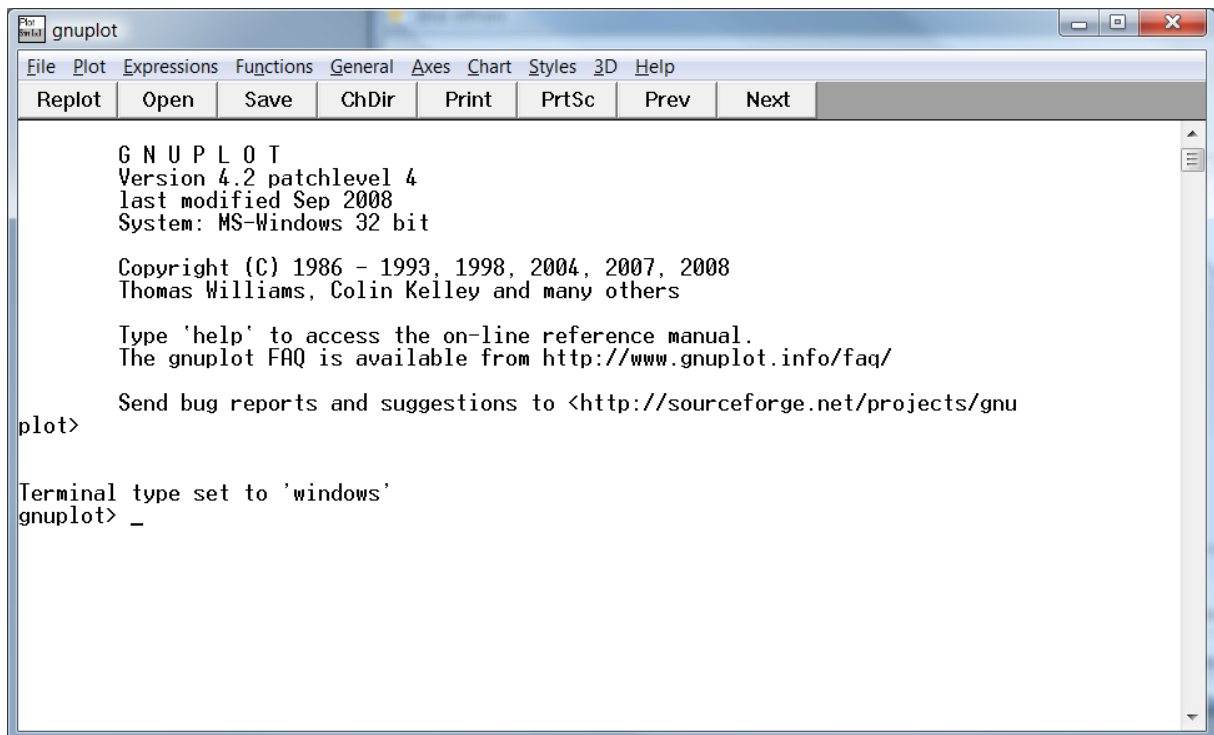


Inhaltsverzeichnis

GNUPLOT – Ein Überblick	2
Bezugsquelle und Installation	2
Erster Kontakt	3
Plots mit Skripten automatisieren	5
Bodediagramm	8
Erweiterte Texteingenschaften	11
Ortskurven	11
Plotten von Messdaten	13
Plotten in eine Datei	15



Die Benutzeroberfläche

GNU PLOT – Ein Überblick

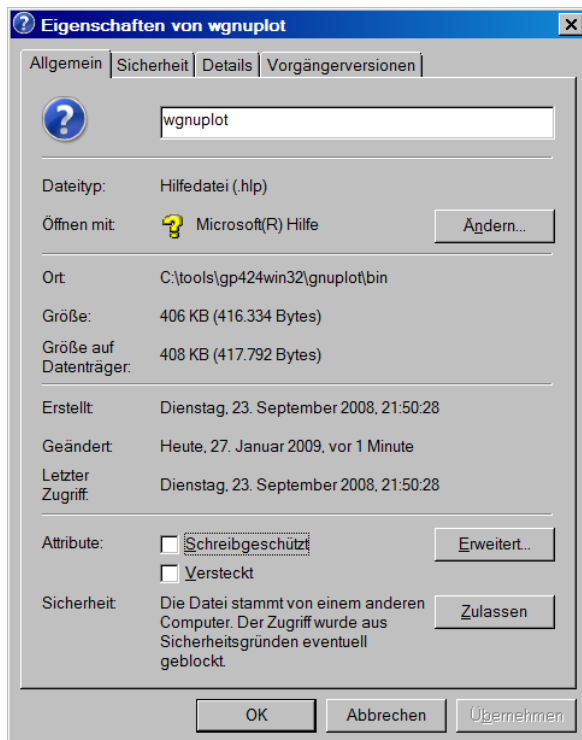
Bei dem aus der Unix-Welt stammenden Freeware Tool GNU PLOT handelt es sich um einen Funktionsplotter der besonderen Art. Auch wenn das Programm über ein umfangreiches Programmmenü verfügt, so wird es im Grunde doch über eine Kommandozeile bedient. Das aufgesetzte Menüsystem dient lediglich dazu, aus den Eingaben des Anwenders Befehle für die Kommandozeile zu generieren. Hat der Anwender erst einmal ein paar Kommandos parat, offenbart sich ihm der Vorteil dieser Art der Programmbedienung und ihm steht fortan ein wirklich mächtiges Werkzeug zur Visualisierung von Funktionen und Messwerten zur Verfügung.

Bezugsquelle und Installation

GNU PLOT ist Freeware. Obwohl der Programmcode offen zur Verfügung steht handelt es sich nicht um ein unter der GPL stehendes Projekt. Ein Umstand auf den die Entwickler auf der Homepage www.gnuplot.info bewusst hinweisen. Dort wird auch die aktuelle Version 4.2.4 für Windows und andere Plattformen zum Download angeboten.

Die Windows Version des Programms muss nicht installiert werden. Es genügt die ZIP Datei (*gp424win32.zip*) an einer beliebigen Stelle zu entpacken. Nach einem Doppelklick auf *wgnuplot.exe* im extrahierten Ordner *bin* erscheint das Programm in schlichtem Gewandt auf dem Desktop. Die gern als zu klein empfundene Schriftgröße kann über *Options* → *Choose Font...* im Systemmenü geändert werden. Soll diese Änderung von Dauer sein, genügt ein anschließender Klick auf *Options* → *Chose* → *Update „LW“:\“Pfad“\update.ini*. Die Datei *update.ini* wird im *AppData\Roaming* Ordner des aktuellen Anwenders abgelegt, so dass alle vorgenommenen Änderungen nur für ihn gelten.

Eine Beschreibung aller Funktionen und Befehle findet sich in der zum Programm gehörenden Hilfe. Sollte Windows Vista statt der Hilfe die sinnige Fehlermeldung „Diese Hilfedatei kann nicht angezeigt werden. Öffnen Sie die Hilfedatei erneut. Wenn diese Nachricht wieder eingeblendet wird, kopieren Sie die Datei auf ein anderes Laufwerk, und ...“ anzeigen, dürfte das ZIP Archiv in einem NTFS Laufwerk entpackt worden sein. Vista versieht Hilfe-Dateien unbekannter Herkunft zur Sicherheit mit einem Marker, der zuverlässig die Anzeige der Hilfe verhindert. Dieser Marker kann im Eigenschaften-Dialog (rechte Maustaste auf *wgnuplot.hlp* → *Eigenschaften*) der Hilfedatei durch einen Klick auf *Zulassen* entfernt werden.

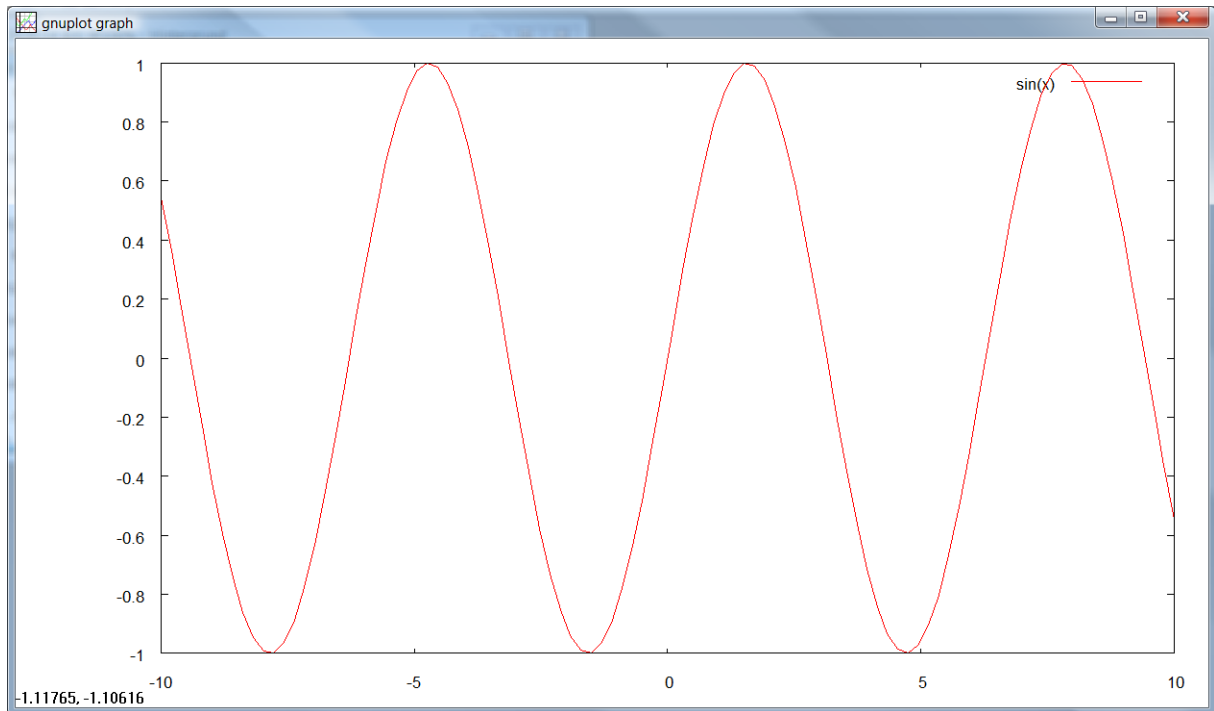


Erster Kontakt

Für einen ersten Test empfiehlt es sich, einfach mal das Kommando `plot sin(x)` direkt auf der Kommandozeile einzugeben.

Alternativ kann das Kommando auch über die aufeinander folgenden Menüaufrufe *Plot* → *Plot* und *Functions* → *Elementary* → *Sine* nach der Eingabe von x als Argument erzeugt werden. In beiden Fällen ist die Eingabe durch Return zu bestätigen.

Es erscheint das neue Fenster *gnuplot graph* mit der Ausgabe des Plots (siehe nächste Seite).

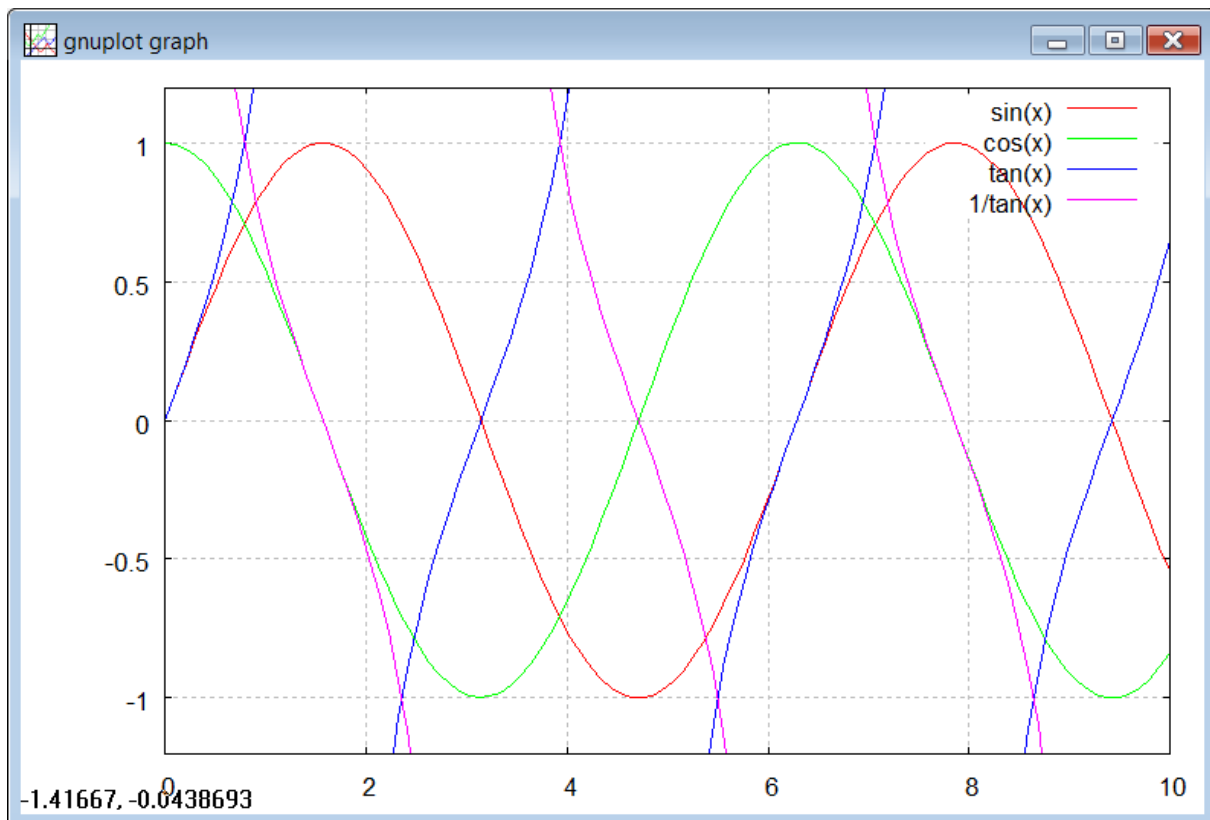


Plot der Funktion $\sin(x)$

Mit der Darstellung nicht zufrieden? Kein Problem. Wie wäre es zur besseren Orientierung mit einem Gitter im Hintergrund. Das Gitter lässt sich über den Menüpunkt *Chart* → *Grid on* im Hauptprogrammfenster einschalten – allerdings ändert sich nach dieser Wahl im Plot nichts. Es erscheint lediglich in der Kommandozeile das Kommando `set grid`, mit dem das Gitter in der Ausgabe weiterer Plots eingeschaltet wird. Erst nach dem Schließen des Ausgabefensters *gnuplot graph* und einem erneuten Plot mit dem Kommando `plot sin(x)` wird das Gitter sichtbar. Übrigens – ein `replot` ohne weitere Parameter wiederholt immer den letzten Plot.

Sollen die Bereiche der Ausgabe festgelegt werden geschieht dies über die Menüpunkte *Axes* → *X Range* bzw. *Y Range* und anschließender Eingabe der Ober- und Untergrenzen. Alternativ und deutlich schneller geht das über die Kommandos `set xrange [0:10]` und `set yrange [-1.2:1.2]`. Nun plottet das Kommando `plot sin(x)` die Sinuskurve in dem soeben definierten Bereich. Ein anschließendes `replot cos(x)` plottet im eben erstellten Plot noch zusätzlich die Kosinusfunktion. Wer möchte kann auch gleich mehrere Funktionen auf einmal plotten lassen - egal ob mit `plot` oder `replot`. So ergänzt ein `replot tan(x), 1/tan(x)` die Grafik noch um den Tangens und Kotangens. Mit dem Kommando `reset` werden alle mit `set` vorgenommenen Einstellungen zurückgesetzt.

Die erneute Eingabe von Kommandos erleichtert die Befehlswiederholung von GNUPLOT. Mit den beiden Tasten `CURSOR HOCH` und `CURSOR RUNTER` besteht die Möglichkeit zwischen den letzten Eingaben zu blättern – auch in denen vergangener Sitzungen. Hierzu werden alle eingegebenen Kommandos in die Datei `gnuplot_histroy` im `AppData\Roaming` Ordner des aktuellen Benutzers geschrieben.



Plot mehrerer trigonometrischer Funktionen

Plots mit Skripten automatisieren

Für Lehrerinnen und Lehrer ist es wichtig Ergebnisse ohne viel Aufwand reproduzieren und ohne viel Mühe verändern zu können. Um sich nun nicht alle eingegebenen Kommandos einzeln notieren zu müssen, besteht die Möglichkeit diese in einer Textdatei zusammenzufassen. Hierzu wird mit einem beliebigen Editor eine Datei mit der Dateiendung *.plt* (z.B. *wechselstromleistung.plt*) erstellt und in dieser die gewünschten Kommandos der Reihe nach eingetragen. Eine solche Datei könnte wie folgt aussehen:

```
# Datei: wechselstromleistung.plt
#
# Autor: H. Bäurle
# Datum: 15.02.2009

# Generelle Einstellungen
reset
set grid
set xrange[0:25]
set yrange[-6:6]
set style fill pattern 5 # Linien zum Ausfüllen
set title "Leistung bei Wechselstrom"
set ylabel "U, I , P"
set xlabel "t in ms"
```

```

# Konstanten
umax = 3 # Scheitelwert der Spannung
imax = 2 # Scheitelwert des Stromes
phi = 2.5 # Phasenverschiebung in ms
T = 20 # Periodendauer in ms

# Funktionen: u(t), i(t) und p(t)
u(x) = umax * sin(2*pi*x/T)
i(x) = imax * sin((2*pi*(x-phi)/T))
p(x) = u(x) * i(x)

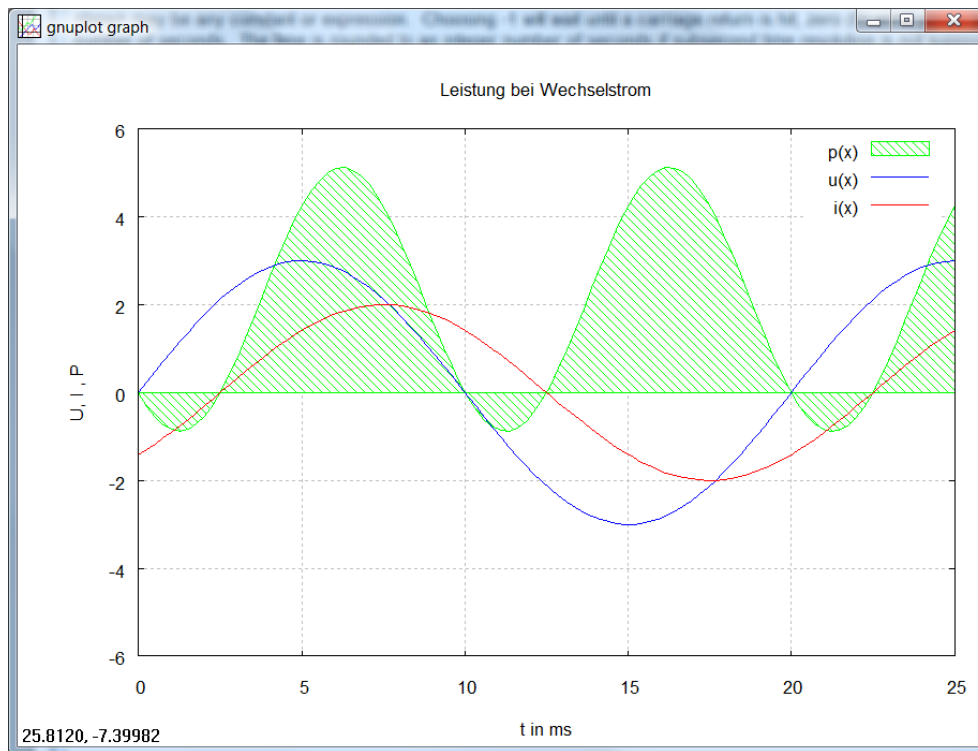
# Ausgabe Spannung und Strom
plot u(x) 3, i(x) 1 # 3: blau, 1: rot
pause -1 "Weiter?" # warten auf Eingabe

# Ausgabe Leistung
plot p(x) with filledcurves y1=0 2
replot u(x) 3, i(x) 1

```

Die plt-Datei wird mit *File* → *Open* ausgewählt und durch Klick auf *Öffnen* geladen und ausgeführt. Der Öffnen-Dialog lässt sich mit einem Klick weniger auch über den Button *Open* unterhalb der Menüleiste öffnen. Wie dem auch sei - auch hier wird letztendlich nur ein Kommando generiert. Konkret handelt es sich um das Kommando *load*, dem in Hochkommas der Dateiname samt Pfadangabe folgt. Alternativ besteht die Möglichkeit zunächst über *Change Directory* aus dem *File* Menü bzw. durch das Kommando *cd* in das gewünschte Verzeichnis zu wechseln. In diesem Fall genügt es an *load* lediglich den Dateinamen ohne Pfadangabe in Hochkommas anzuhängen. Aktuellen Pfad vergessen? Macht nichts - *pwd* (print working directory) gibt gerne Auskunft.

Das obige Beispiel führt zu folgendem Plot:



Wie im Script zu sehen ist, können Konstanten und Funktionen festgelegt werden. Im Beispiel werden zu Beginn des Skriptes die Scheitelwerte für Strom und Spannung, die Periodendauer sowie die Phasenverschiebung als Konstanten definiert. Die anschließenden Funktionen verwenden dann ausschließlich diese Konstanten. Sollen die Kurven für einen anderen Belastungsfall geplottet werden, genügt es dann die Werte der Konstanten im Script entsprechend zu ändern.

Der Verlauf einer sinusförmigen Wechselspannung ist gegeben durch:

$$u(t) = \hat{u} \cdot \sin(\omega \cdot t) = \hat{u} \cdot \sin\left(\frac{2 \cdot \pi \cdot t}{T}\right)$$

In GNUPLLOT wird x und nicht t als Laufvariable verwendet. Die Zeile `u(x) = umax * sin(2*pi*x/T)` definiert $u(t)$ als eine Funktion von x . Diese Funktion kann dann z.B. durch `plot u(x)` geplottet oder bei Bedarf auch innerhalb der Definition neuer Funktionen verwendet werden. Im Beispiel wird die Leistung $p(x)$ als Produkt der beiden Funktionen $i(x)$ und $u(x)$ definiert und kann dann wie jede andere Funktion verwendet werden.

Für eine Präsentation im Unterricht können mit dem Kommando `pause` Pausen zwischen den einzelnen Plots eingefügt werden. Die Zahl hinter dem Befehl `pause` gibt die Dauer der Pause in Sekunden an. Auf Wunsch kann dem Pause Befehl noch ein Text in Gänsefüßchen angehängt werden, der dann auf der Kommandozeile ausgegeben wird. Wird als Zeitwert `-1` verwendet, erscheint ein Dialog, der vor dem Weiterplotten bestätigt werden muss. Steht anstelle einer Zeit das Wort `mouse`, muss zum Fortsetzen mit der Maus in den Plot geklickt werden.

Im Beispiel werden nach der Pause und dem darauf folgenden Plotten der Leistung die beiden Kurven für Strom und Spannung erneut geplottet, da sie von der Leistungskurve überschrieben wurden, aber im Vordergrund zu sehen sein sollen.

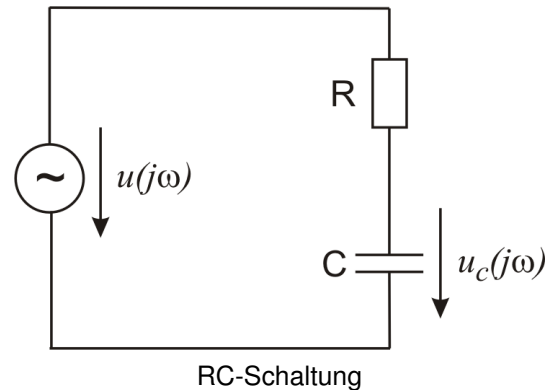
Es gibt zahlreiche Möglichkeiten auf die Gestaltung der Kurven einzuwirken. Die Zahl am Ende der Funktion nach dem `plot` Kommando legt die Farbe fest, mit der diese geplottet wird. So

wird die durch den Befehl `plot p(x) with filledcurves y1=0 2` ausgegebene Leistung in grün (2) gezeichnet. Durch die Option `filledcurves y1=0` wird der Bereich zwischen der X-Achse ($y = 0$) und der Funktion $p(x)$ mit dem durch `set stylefill` festgelegten Füllmuster – hier schraffiert – ausgegeben. Um mehr über die Möglichkeiten zur Gestaltung in Erfahrung zu bringen, lohnt sich ein Blick in die Hilfe. Ungeduldige können auch das Kommando `test` eingeben und sich durch das erscheinende Bild inspirieren lassen.

Alle Kommentare beginnen mit dem #-Zeichen und können auch nach einem Kommando beginnen.

Bodediagramm

Ein Thema von zentraler Bedeutung im Elektrotechnikunterricht ist das Verhalten von RC Schaltungen. Mit den bereits besprochenen Beispielen dürfte es ein leichtes sein, das Verhalten im Zeitbereich (Spannung und Strom beim Laden und Entladen des Kondensators) zu plotten. In der Nachrichtentechnik sowie der Regelungstechnik interessiert zusätzlich das Verhalten im Frequenzbereich.



Die Übertragungsfunktion des RC Gliedes bestimmt das Verhältnis zwischen Ein- und Ausgangsspannung. Im Frequenzbereich wird hierzu die komplexe Übertragungsfunktion $G(j\omega)$ definiert.

$$u_C(j\omega) = \underline{G}(j\omega) \cdot u(j\omega)$$

Daraus ergibt sich für $G(j\omega)$:

$$\underline{G}(j\omega) = \frac{u_C(j\omega)}{u(j\omega)} = \frac{X_C}{R + X_C} = \frac{\frac{1}{j\omega C}}{R + \frac{1}{j\omega C}} = \frac{1}{1 + j\omega R \cdot C}$$

Das Produkt aus R und C ist hierbei die Zeitkonstante τ des RC Gliedes. Aus Sicht der Regelungstechnik handelt sich bei der RC Schaltung um eine Regelstrecke 1. Ordnung – also um ein PT1 Glied. Der Regelungstechniker schreibt an Stelle von τ ein T und erweitert die Gleichung um den Proportionalitätsfaktor K_{PS} , der die Verstärkung der Regelstrecke im eingeschwungenen Zustand angibt.

Allgemein gilt:

$$\underline{G}(j\omega) = \frac{K_{PS}}{1 + j\omega T_1} \quad \text{komplexer Frequenzgang PT1 Glied}$$

Aufgesplittet in Real- und Imaginärteil lässt sich der Frequenzgang auch folgendermaßen darstellen.

$$\underline{G}(j\omega) = \frac{1}{1 + (\omega T_1)^2} - j \frac{\omega T_1}{1 + (\omega T_1)^2}$$

Durch Betragsbildung lässt sich daraus der Amplitudenfrequenzgang ermitteln:

$$|\underline{G}(j\omega)| = \frac{|K_{PS}|}{\sqrt{1 + \omega^2 T_1^2}} \quad \text{Amplitudenfrequenzgang PT1 Glied}$$

Der Phasengang ergibt sich aus dem Arcustangens des Quotienten von Imaginär- und Realteil.

$$\varphi(\omega) = \arctan\left(\frac{-\frac{\omega T_1}{1+(\omega T_1)^2}}{\frac{1}{1+(\omega T_1)^2}}\right) = \arctan(-\omega T_1) \text{ Phasenfrequenzgang PT1 Glied}$$

Die gemeinsame Darstellung des Amplitudenfrequenzganges und des Phasenfrequenzganges erfolgt in einem Bode-Diagramm. Hierbei wird die X-Achse logarithmisch eingeteilt und die Verstärkung wird in dB angegeben.

Das folgende Script erzeugt das Bode Diagramm eines PT1 Gliedes aus den Funktionen für den Amplituden- und Phasenfrequenzgang.

```
# Datei: pt1_bodediagramm.plt
#
# Autor: H. Bäurle
# Datum: 15.02.2009

# Beschriftung
reset # Sicher ist sicher.
set title "PT1-Glied"
set ylabel "Amplitudenfrequenzgang in dB / Phasenfrequenzgang in Grad"
set xlabel "Kreisfrequenz in {/Symbol w}"

# Einstellung (Achsen, etc. )
set xrange [0.01:1000] # X-Achse von 0,01 bis 1000
set yrange [-90:10] # Y-Achse von -90 bis 10 Grad
set logscale x # X-Achse logarithmisch
set angles degrees # von RAD nach DEG
set grid # Gitter einblenden

# Konstanten KPS = 1
T1 = 1

# Funktionen
G(x) = KPS/sqrt(1+(x*T1)**2) # G(jw) **2 bedeutet hoch 2
a(x) = 20*log10(G(x)) # Umrechnung in dB
P(x) = atan(-x*T1)

# Ausgabe
plot a(x)
replot P(x)
```

GNU PLOT ist auch in der Lage, komplex zu rechnen. Das folgende Script erzeugt die gleiche Ausgabe wie das vorhergehende, verwendet aber direkt die komplexe Übertragungsfunktion $G(j\omega)$ des PT1 Gliedes. Für den Plot des Amplitudenfrequenzganges wird mit $abs()$ der Betrag von $G(j\omega)$ gebildet.

```
# Datei: pt1_bodediagramm_komplex.plt
#
# Autor: H. Bäurle
# Datum: 15.02.2009

# Beschriftung
reset # Sicher ist sicher.
set title "PT1-Glied"
set ylabel "Amplitudenfrequenzgang in dB / Phasenfrequenzgang in Grad"
set xlabel "Kreisfrequenz in {/Symbol w}"

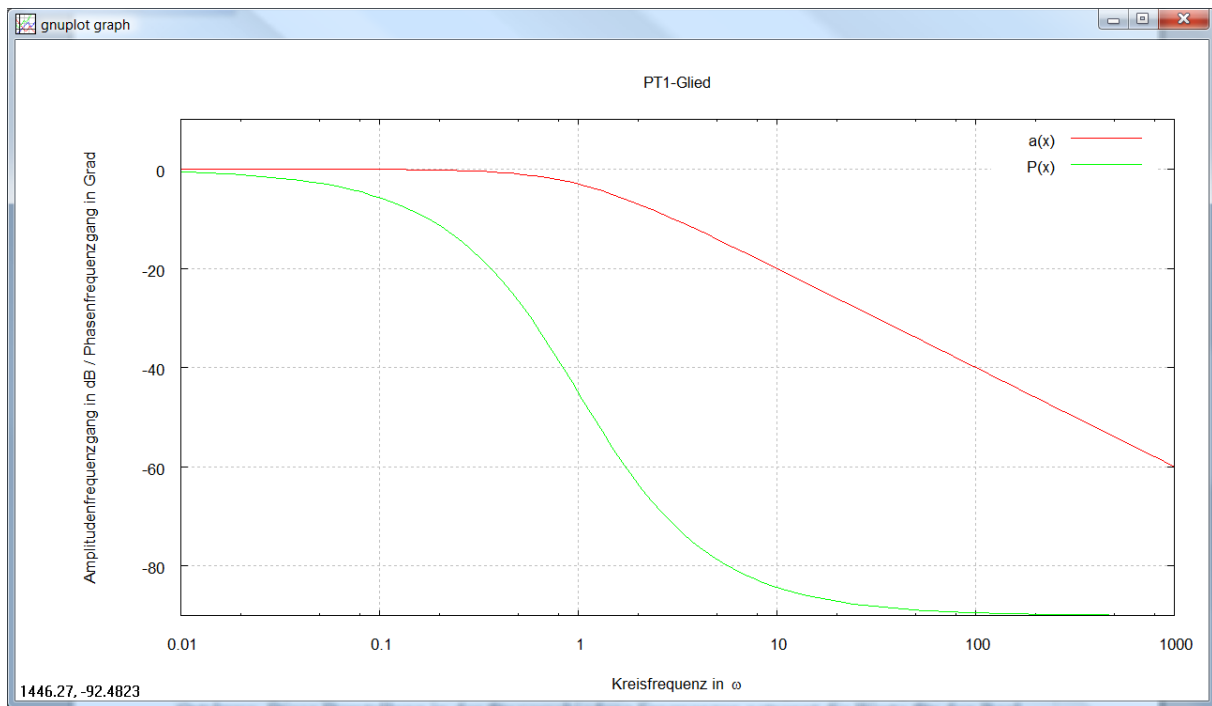
# Einstellung (Achsen, etc. )
set xrange [0.01:1000] # X-Achse von 0,01 bis 1000
set yrange [-90:10] # Y-Achse von -90 bis 10 Grad
set logscale x # X-Achse logarithmisch
set angles degrees # von RAD nach DEG
set grid # Gitter einblenden

# Konstanten
KPS = 1
T1 = 1

# Verwendung komplexer Zahlen definieren
j = {0,1} # Definiert die imaginäre Grundeinheit
# "j" = 0 + j1
# Die komplexe Zahl 2 + j4 wird als {2,4}
# dargestellt.

# Funktionen
G(x) = KPS/(1+j*x*T1) # G(jw)
a(x) = 20*log10(abs(G(x))) # Umrechnung in dB
P(x) = atan(-x*T1)

# Ausgabe
plot a(x)
replot P(x)
```



Frequenzgänge beim PT1-Glied

Erweiterte Texteigenschaften

Prinzipiell ist es GNUPLOT egal, ob es die Ausgabe in eine Datei oder in ein Fenster auf dem Bildschirm schreibt. Nach dem Kommando *show terminal* verrät GNUPLOT dem interessierten Anwender: *terminal type is windows color noenhanced*. Dies bedeutet, dass das Ausgabeterminal ein farbiges Fenster ohne erweiterte Texteingenschaften ist. Diese erweiterten Texteingenschaften erlauben Dinge wie das Hoch- und Tiefstellen von Text, die Auswahl einer Schrift oder die Ausgabe griechischer Buchstaben. Mit einem *set terminal windows color enhanced* werden die erweiterten Möglichkeiten zur Textgestaltung eingeschaltet. Danach wird durch die Anweisung *set xlabel "Kreisfrequenz in ω "* in zukünftigen Plots die x-Achse mit dem Text „Kreisfrequenz in ω “ beschriftet.

Ortskurven

Zur Stabilitätsuntersuchung von Regelkreisen wird gerne die Übertragungsfunktion als Ortskurve dargestellt. Diese Darstellung, in der für verschiedene Frequenzen getrennt die Werte für den Real- und Imaginärteil berechnet und in ein Diagramm eingetragen werden, gehört ebenfalls zum Repertoire von GNUPLOT.

```
# Ortskurve PT1
#
# Autor: Harald Bäurle
# Date: 15.12.2009

# Erst mal alles zurücksetzen. Sicher ist sicher.
reset

# Zeichne die X- und Y-Achse (Linientype -1), Gitter zeichnen
```

```

set xzeroaxis lt -1
set yzeroaxis lt -1
set grid

# Definiere j
j = {0.,1.}
# Konstante
KPS = 1
T1 = 1

# komplexe Fkt. G(omega)
G(w) = KPS / (1 + j*w*T1)

# Plotten von G(omega)
set parametric
set dummy w          # w als Laufvariable bestimmen
set trange[0:1000]   # w von 0 bis 1000 laufen lassen

# Plotbereich, Achsenbeschriftung
set label "Re(G)" at 0.9,0.1 center
set xtics axis -1,0.2,1 # von -1 bis +1 im Abstand von 0,2
                           beschriften
set xrange [-1 : 1]
set label "Im(G)" at 0.05,0.8 rotate
set ytics axis -1,0.2,1
set yrange [-1 : 1]
set title "Ortskurve eines PT1-Gliedes mit KPS = 1 und T1 = 1"

# Plotten
set samples 100000    # Anzahl der zu berechnenden Punkte
plot real(G(w)), imag(G(w))

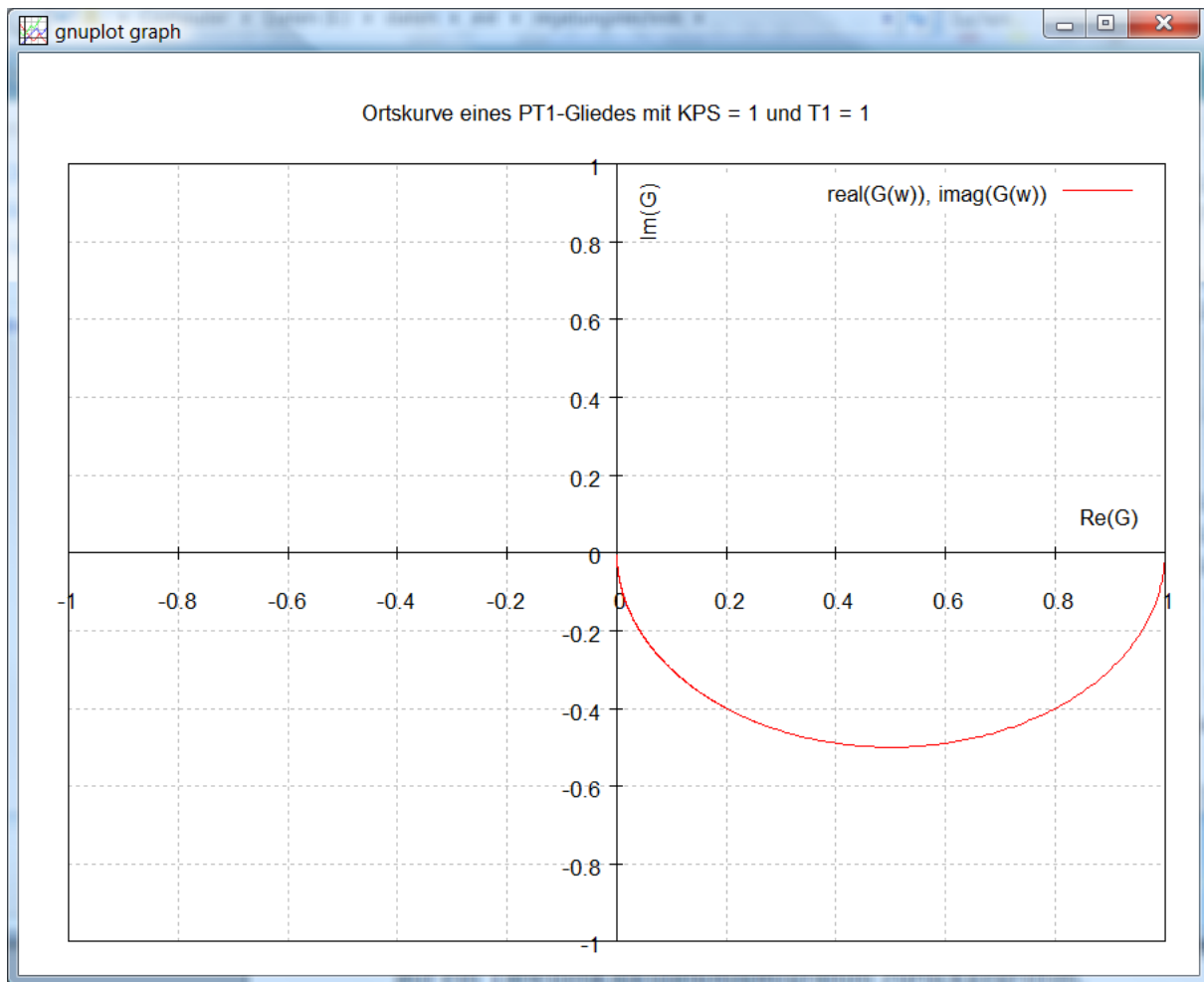
```

Die Funktion wird diesmal nicht als Funktion von x sondern als Funktion von w definiert. Aus diesem Grund muss zunächst w als Laufvariable (*set dummy w*) deklariert werden. Anschließend wird der zu berechnende Wertebereich festgelegt (*set trange[0:1000]*).

Aus dem komplexen Ergebnis von $G(w)$ ermitteln die beiden Funktionen *real()* und *imag()* jeweils den zugehörigen Real- und Imaginärteil und übergeben ihn an das *plot* Kommando, das zu dem durch diese beiden Koordinaten gegebenen Punkt von der aktuellen Position aus eine Linie zeichnet. Die Anzahl der Punkte die berechnet werden wird über das Kommando *set samples* festgelegt.

Für $\omega = 0$ beginnt die Kurve bei $1 + j 0$ und läuft für $\omega \rightarrow \infty$ zum Ursprung. Für eine hinreichend genaue Darstellung ist es wichtig, dass auch für kleine Werte von ω genug Punkte berechnet werden. Um dies zu erreichen, wird mit *set samples* die Anzahl der zu berechnenden Werte auf 100000 festgelegt. Bei einem Wert von nur 1000 liegt bereits der zweite zu berechnende Punkt bei $0,5 - j 0,5$. In Folge dessen wird eine Gerade von $1 + j 0$ zu diesem Punkt gezeichnet.

Mit den Kommandos *set xtics* und *set ytics*, kann die Achseneinteilung den eigenen Wünschen angepasst werden. Hierzu werden die beiden Endwerte sowie dazwischen die gewünschte Schrittweite angegeben.



Ortskurve

Plotten von Messdaten

Die graphische Auswertung von Messergebnissen gehört zu einer immer wiederkehrenden Aufgabe im Unterricht. Gerne wird hier auf Tabellenkalkulationsprogramme und deren Möglichkeit, Daten zu visualisieren, zurückgegriffen. Eine Aufgabe, die mit GNUPLOT ebenfalls einfach gelöst werden kann. Nach Ansicht des Autors liefert GNUPLOT in der Regel qualitativ bessere Ergebnisse. Außerdem sind durch den Einsatz von Skripten die Ausgaben von GNUPLOT leichter zu reproduzieren bzw. Änderungen leichter vorzunehmen.

Im folgenden Beispiel wurde das Verhalten eines Asynchronmotors bei Belastung untersucht. Hierzu wurde ein unter Spannung stehender Motor an einem Motorenprüfstand angeschlossen und seine Drehzahl von 0 bis über die Drehfeldzahl hinaus erhöht und das Drehmoment sowie die Stromaufnahme gemessen. Für die Auswertung wurde der Strom ins Verhältnis zum Nennstrom gesetzt. Die gemessenen und berechneten Werte wurden in die Textdatei *messwerte.txt*, durch Tabulatorsprünge getrennt, eingetragen.

```
# Datei: messwerte.txt
#
# Autor: H. Bäurle
# Datum: 15.02.2009
#
```

```
# Drehzahl in 1/min Drehmoment in Nm I/IN
0 68.18 4.753
450 91.47 4.647
750 114.75 4.329
900 128.05 4.106
1050 140.28 3.741
1140 143.85 3.388
1200 141.36 3.082
1350 103.11 1.906
1425 58.21 1.106
1500 0 0.529
1575 -66.52 1.165
```

Das Skript zur Darstellung der Motorenkennlinie könnte wie folgt aussehen:

```
# Datei: messwerte.plt
#
# Autor: H. Bäurle
# Datum: 15.02.2009
#
# Drehmoment-, Stromkennlinie Asynchronmotor

# Einstellungen
reset
set title "Asynchronmotor"
unset key # Unterdrückt die Legende
set grid
set xrange[0:1600]
set xlabel "n in [1/m_i_n]" # ^ hochstellen, _ tiefstellen
set yrange[-80:150]
set ylabel "M in [Nm]"
set y2range[-3.2:6]
set y2label "I / I_N"
set y2tics '1'

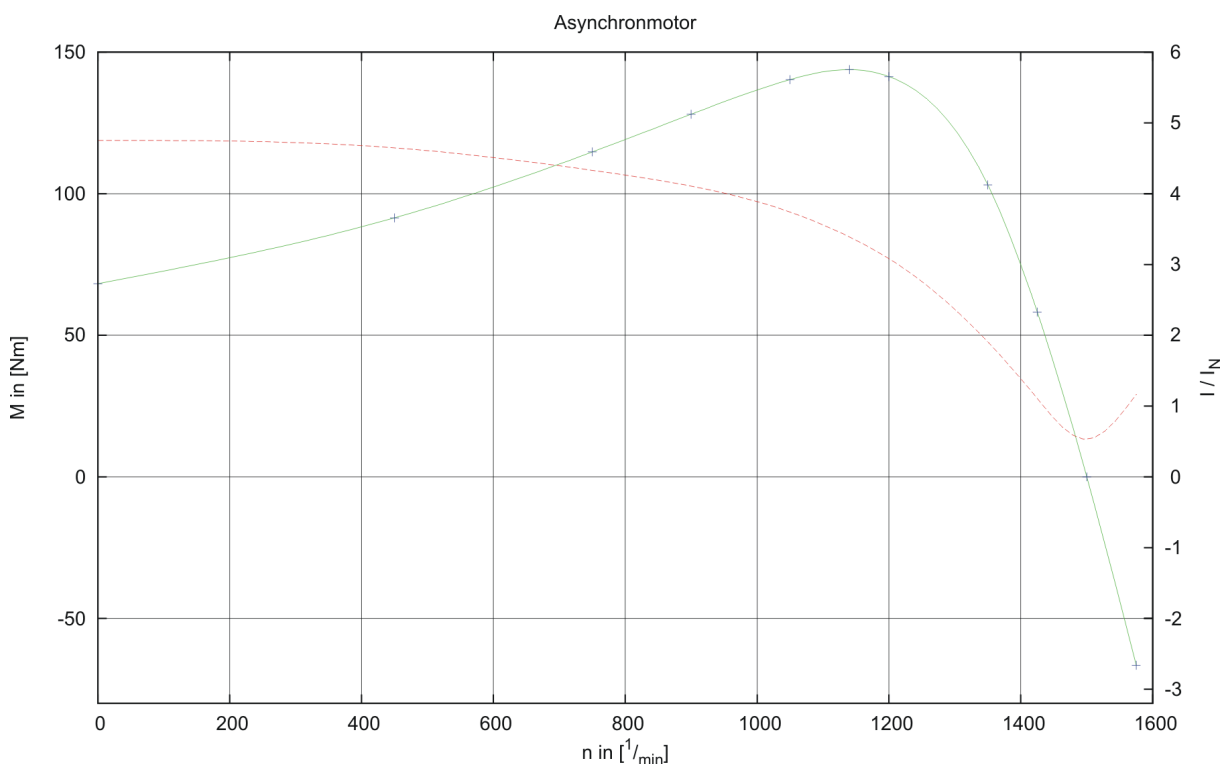
# Ausgabe Kennlinien und Messwerte
# Da in eine Datei geplottet werden soll funktioniert replot nicht.
# Alle plot-Parameter stehen in einer Zeile.
plot 'messwerte.txt' using 1:2 axes x1y1 smooth cspline lc
2, 'messwerte.txt' using 1:3 axes x1y2 smooth cspline lc 1,
'messwerte.txt' using 1:2 axes x1y1 with points
```

Um die Messwerte aus der Textdatei zu plotten, wird deren Dateiname lediglich dem Kommando *plot* angehängt. Ohne Pfadangabe wird die Datei im aktuellen Ordner gesucht. Die Datei *messwerte.txt* enthält drei Spalten. Welche der Spalten zum Plotten verwendet werden sollen, wird dem *plot* Kommando durch die beiden durch : getrennten Zahlen mitgeteilt. Das Kommando *plot 'messwerte.txt' using 1:2 axes x1y1 smooth cspline lc2* verwendet also zum Plotten die erste und zweite Spalte der Datei *messwerte.txt*. Das folgende *x1y1* bestimmt auf welche

Achsen die Werte beider Spalten verteilt werden sollen. In diesem Fall bestimmen die Werte der ersten Spalte (Drehzahl n) die X-Position. Die jeweils zugehörigen Werte der zweiten Spalte (Drehmoment M) werden nach der 1. Y-Achse eingetragen. Durch *smooth cspline* wird der Verlauf der Kurve zwischen den Messpunkten interpoliert.

Für die in der dritten Spalte stehende Verhältnisse von I zu I_N wurde durch *set y2range*, *set y2label*, *set y2tics* eine zweite Y-Achse definiert. Durch *1:3 axes x1y2* wird das Plot Kommando angewiesen für die Werte der dritten Spalte die zweite Y-Achse zu verwenden. Diese Kurve soll ebenfalls interpoliert (*smooth cspline*) und in der Linienfarbe rot (*lc 2*) ausgegeben werden. Den dritten Auftrag, den *plot* in diesem Beispiel erhält, ist es noch einmal die Werte der Drehmoment Messung auszugeben. Allerdings sollen nur die Werte eingezeichnet werden (*with points*).

Plotten in eine Datei



Messdaten als Grafikdatei

Bei den bisherigen Beispielen wurde der Plot auf dem Bildschirm ausgegeben. Wegen der im Vergleich zu Druckern schlechten Auflösung von Monitoren ist es nicht sinnvoll für Printmedien auf Screenshots der Kurven zurückzugreifen. Die Möglichkeit per „copy and paste“ die Kurven in Dokumente einzufügen führt leider in der Regel auch nicht zu den gewünschten Ergebnissen.

Über den schon weiter oben erwähnten *set terminal* Befehl, kann das Ausgabeformat geändert werden. Statt *windows* - für die Ausgabe in einem Fenster - wird dem Kommando als Option der Name des gewünschten Dateiformates z.B. *jpg* oder *gif* angehängt. Welche Dateiformate mit welchen Optionen zur Verfügung stehen, kann der Hilfe entnommen werden.

Für die Ausgabe der Kennlinie des Asynchronmotors soll wegen der guten Qualität die Ausgabe als Encapsulated Postscript erfolgen. Zunächst muss mit *set terminal postscript eps enhanced colour size 16cm,10cm* das Ausgabeformat geändert werden. Die Option *enhanced* schaltet die

erweiterten Textformatierungsmöglichkeiten ein. Über *size XX,YY* wird die Größe der Grafik festgelegt. Ohne Angabe einer Maßeinheit geht GNUPLOT von Zoll (inch) aus.

Zwar wurde mit diesem Kommando das Ausgabeformat geändert, aber noch nicht das Ausgabemedium. Hier merkt man GNUPLOT seine Wurzeln an. Die Ausgabe erfolgt auf der Standardausgabe `STDOUT` – also im Kommandofenster. Damit eine *eps*-Datei entsteht, muss mit *set output "messwerte.eps"* zunächst einmal die Ausgabe auf eine Datei umgeleitet werden. Achtung: Das Zurücksetzen erfolgt nicht über *set output "STDOUT"*, wie auf den ersten Blick anzunehmen ist, sondern nur über *set output*. Im ersten Fall wird in die Datei `STDOUT` geschrieben. Die Datei wird ohne Pfadangabe im aktuellen Ordner angelegt. Mit *show output* wird die aktuelle Einstellung angezeigt.

Wer einfach nur den letzten Plot in eine Datei sichern will, kann diese durch ein einfaches *replot* nach dem Umstellen der Ausgabe wiederholt plotten lassen.

GNUPLOT kann noch viel mehr, als hier dargestellt. Im Internet gibt es unzählige Quellen mit weiteren Beispielen. Also bei Interesse – einfach nur googeln.

Harald Bäurle, Elektronikschule Tett nang

□