

Praktikum für Fortgeschrittene

Versuch 252

Digitale Filter

Analoge Signale werden heute zunehmend durch den Einsatz digitaler Signalverarbeitung gemessen und bearbeitet: sie werden in Zahlenformen (digitale Signale) umgewandelt und auf Digitalrechnern zur Weiterverarbeitung und Auswertung gespeichert. Dieser Versuch führt in die Grundlagen der digitalen Signalverarbeitung ein. Es wird ein schneller *Signalprozessor* verwendet, um die digitalen Signale so schnell zu verarbeiten, daß die Ergebnisse sofort wieder in analoge Signale zurückgewandelt werden können (*Echtzeit-Verarbeitung*). Zunächst werden Fehler und Grenzen bei der Umwandlung analoger Signale in digitale Signale untersucht (Samplingtheorem, Quantisierungsrauschen) und anschließend digitale Filter maximal 2. Ordnung realisiert und ausgemessen.

Zubehör

Signalgenerator (Sinus) für Frequenzen von 0 bis 20 kHz

Oszilloskop

RMS-Röhrenvoltmeter mit dB-Skala

Filter 2. Ordnung mit einstellbaren Koeffizienten (in einem Gehäuse mit AD/DA-Wandlern)

Literatur

- [1] A. Lacroix: *Digitale Filter*. München/Wien: Oldenbourg, 1980
- [2] H.W. Schüssler: *Digitale Systeme zur Signalverarbeitung*. Berlin/Heidelberg/New York: Springer, 1973
- [3] S.D. Stearns: *Digitale Verarbeitung analoger Signale*. München/Wien: Oldenbourg, 1979
- [4] S.A. Azizi: *Entwurf und Realisierung digitaler Filter*. München/Wien: Oldenbourg, 1983
- [5] A.V. Oppenheim, R.W. Schäfer: *Digital Signal Processing*. Englewood Cliffs (NJ): Prentice Hall, 1978
- [6] L.R. Rabiner, B. Gold: *Theory and Application of Digital Signal Processing*. Englewood Cliffs (NJ): Prentice Hall, 1975

Gang des Versuchs

1. Quantisierungsrauschen

Messen der Leistung des Quantisierungsfehlers bei $f_s = 25$ kHz und einem sinusförmigen Eingangssignal von 20 Hz für verschiedene Wortbreiten des AD-Wandlers.

2. Rekursives Filter 1. Ordnung

$$\alpha_1 = \alpha_2 = \beta_2 = 0; \quad \alpha_0 = 1$$

$$\beta_1 = 0.5; \quad 0.9; \quad 0.98$$

$$f_s = 10 \text{ kHz}$$

Messung des Frequenzgangs von 20 Hz bis 20 kHz.

3. Rekursives Filter 1. Ordnung

$$\alpha_0 = \alpha_2 = 0$$

$$\alpha_1 = r \sin \gamma; \quad \beta_1 = 2r \cos \gamma; \quad \beta_2 = -r^2$$

$$r = 0.9; \quad 0.95; \quad 0.99; \quad \gamma = 30^\circ$$

$$f_s = 10 \text{ kHz}$$

Messung des Frequenzgangs von 20 Hz bis 5 kHz.

Messung der Impulsantwort.

4. Nichtrekursives Filter

$$\beta_1 = \beta_2 = 0$$

$$\alpha_0 = \alpha_1 = \alpha_2 = 1$$

$$f_s = 10 \text{ kHz}$$

Messung des Frequenzgangs von 20 Hz bis 5 kHz.

Außerdem dasselbe für eine weitere, beliebige Nullstelle im oder auf dem Einheitskreis (die Parameter α_i entsprechend selbst wählen).

5. Tiefpaßfilter 2. Ordnung

Butterworth: $f_g = 1$ kHz

$$[\alpha_1 = 2; \quad \alpha_0 = \alpha_2 = 1] \times 1/16$$

$$\beta_1 = 1.143; \quad \beta_2 = -0.428$$

$$f_s = 10 \text{ kHz}$$

Tschebyscheff: $f_g = 1$ kHz; 3 dB Welligkeit

$$[\alpha_1 = 2; \quad \alpha_0 = \alpha_2 = 1] \times 1/16$$

$$\beta_1 = 1.532; \quad \beta_2 = -0.7094$$

$$f_s = 10 \text{ kHz}$$

Messung des Frequenzgangs jeweils von 20 Hz bis 5 kHz.

Auswertung

1. Auftragung des RMS-Wertes des Quantisierungsrauschens über der Wortbreite des AD-Wandlers. Abschätzung des nutzbaren Dynamikbereichs.
2. Zeichnen des Frequenzgangs und Bestimmung der Grenzfrequenz (3 dB - Abfall, "full width"). Bestimmung der Abtastfrequenz. Vergleich mit der Theorie.
3. Zeichnen des Frequenzgangs. Bestimmung der Resonanzfrequenz und der Halbwertsbreiten. Bestimmung der Frequenz und des logarithmischen Dekrements aus der Impulsantwort. Vergleich der Meßwerte aus Frequenzgang und Impulsantwort untereinander. Vergleich der Meßwerte mit der Theorie.
4. Zeichnen des Frequenzgangs. Bestimmung der Resonanzfrequenz (=Sperrfrequenz) und Vergleich mit der Theorie.
5. Zeichnen der Frequenzgänge. Bestimmung der Grenzfrequenzen (3 dB - Abfall relativ zu $H(\omega \rightarrow 0)$). Bestimmen der Welligkeit beim Tschbyscheff-Tiefpaß [$H_{max} - H(\omega \rightarrow 0)$]. Vergleich von Butterworth und Tschbyscheff.

Bemerkungen:

Es ist darauf zu achten, daß der gemeinsame Faktor für α_i so eingestellt wird, daß keine Überläufe auftreten ($H(z)$ muß immer kleiner als 2 sein).

Zu 3.:

$$H(z) = \frac{(r \sin \gamma)z}{z^2 - (2r \cos \gamma)z + r^2}$$

ist Z-Transformierte von

$$h_n = r^n \sin(n\gamma).$$

Für den Zusammenhang zwischen logarithmischem Dekrement

$$\Lambda = \ln \frac{x_{max}^{(i)}}{x_{max}^{(i+1)}},$$

Halbwertsbreite Δf und Resonanzfrequenz f_{res} gilt näherungsweise

$$\Delta f = \frac{1}{\pi} \Lambda f_{res}.$$

1 Theorie

1.1 Quantisierungsrauschen

Um analoge Signal digital verarbeiten zu können, müssen diese zunächst in binäre Zahlen umgewandelt werden. Dies geschieht mit Hilfe von Digital-Analog-Wandlern. Die Genauigkeit dieser Umwandlung hängt hauptsächlich von der Wortbreite ab, d.h. der Anzahl Bits, mit welcher die Zahlen dargestellt werden können. Ein 12-Bit-Wandler hat z.B. $2^{12} = 4096$ Zahlen zur Darstellung der analogen Werte zur Verfügung. Die Kennlinie dieser Quantisierung entspricht einer Treppenfunktion mit der Stufenhöhe q (Quantisierungsstufe), wobei

$$q = 2^{-W}A \quad (1)$$

mit der Wortbreite W und der Vollaussteuerungsamplitude A .

Der beim Quantisieren auftretende Fehler e hängt wie folgt von der Quantisierungsstufe q ab:

$$-\frac{q}{2} \leq e \leq \frac{q}{2}. \quad (2)$$

Im allgemeinen kann dieser Quantisierungsfehler als gleichverteilte Zufallsgröße angenommen werden, deren Verteilung $p(e)$ vom analogen Eingangssignal unabhängig ist: $p(e) = 1/q$. Wegen der statischen Eigenschaften des Fehlers spricht man von *Quantisierungsrauschen*.

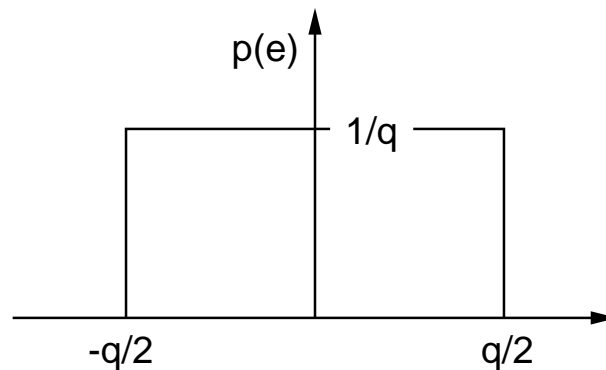


Abb. 1: Verteilungsfunktion des Quantisierungsfehlers.

Die Varianz des Quantisierungsfehlers ist seiner Leistung proportional und berechnet sich als

$$\sigma_e^2 = \int_{-q/2}^{q/2} p(e)(e - \langle e \rangle)^2 de = \int_{-q/2}^{q/2} \frac{1}{q}(e - 0)^2 de = \frac{q^2}{12}. \quad (3)$$

Um den Einfluß des Quantisierungsrauschens abschätzen zu können, interessiert man sich für das Verhältnis der Varianz des Nutzsignals σ_s^2 zur Varianz des Quantisierungsrauschens σ_e^2 . Dieses entspricht dem Verhältnis der Leistungen von Signal und Rauschen und wird daher als *Signal-Rausch-Abstand* (SNR = Signal to Noise Ratio) bezeichnet. Ein voll ausgesteuerter Sinus hat die Varianz

$$\sigma_s^2 = \frac{1}{T} \int_{-T/2}^{T/2} \cdot \left(\frac{A}{2} \sin(2\pi ft) \right)^2 dt = \frac{A^2}{8} = \frac{1}{8} q^2 2^{2W}. \quad (4)$$

Daraus erhalten wir:

$$\text{SNR} = \frac{\sigma_s^2}{\sigma_e^2} = \frac{12}{8} 2^{2W}, \quad (5)$$

$$\text{SNR} = 10 \log_{10} \frac{12}{8} \text{ dB} + 10 \cdot 2W \log_{10} 2 \text{ dB} = W \cdot 6 \text{ dB} + 1.8 \text{ dB}. \quad (6)$$

Somit erhält man für jedes zusätzliche Bit der Wortbreite eine Verbesserung des SNR von 6 dB. Zum Beispiel haben 12-Bit-Wandler einen (maximalen) SNR von 74 dB, 16-Bit-Wandler von 96 dB.

Bei der bisherigen Betrachtung wurde noch kein Einfluß der endlichen Abtastrate (Samplingfrequenz f_s), also kein Rauschen durch Quantisierung der Zeit, berücksichtigt. Man kann sich überlegen, daß eine Erhöhung der Wortbreite bei festem f_s dann keine Verbesserung der Qualität mehr bringt, wenn die Änderung des Eingangssignals während einer Abtastperiode $T_s = 1/f_s$ bereits größer ist als eine Quantisierungsstufe q . Für einen vollausgesteuerten Sinus der Frequenz f erhält man als Abschätzung der maximalen Wortbreite, bevor dieser Effekt auftritt:

$$\left| \frac{d}{dt} \left(\frac{A}{2} \sin(2\pi ft) \right) \right|_{\max} \cdot T_s \leq q = A \cdot 2^{-W} \quad (7)$$

$$\Rightarrow \pi f T_s \leq 2^{-W} \quad \Rightarrow \quad W \leq \frac{\ln(f_s/\pi f)}{\ln(2)}. \quad (8)$$

Oberhalb von $W \approx \ln(f_s/\pi f)/\ln(2)$ erwartet man daher keine signifikante Verminderung des Rauschens mehr.

1.2 Abtasttheorem

Bei der digitalen Signalverarbeitung wird einerseits die Größe des Eingangssignals durch Quantisierung diskret gemacht und andererseits der Zeitverlauf des Signals diskretisiert, indem das Signal in festen Abständen

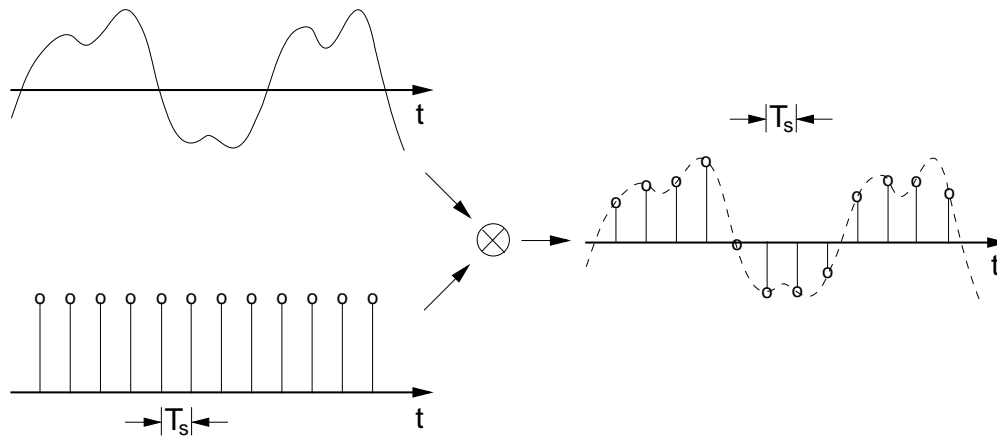


Abb. 2: Beschreibung der Abtastung durch einen Dirac-Kamm.

abgetastet wird. Daher stellt sich die Frage, wie ein Zeitsignal durch äquidistante Abtastung verfälscht wird. Um diese Frage zu beantworten, muß man das Spektrum der abgetasteten Funktion betrachten.

Formal wird die Abtastung durch die Multiplikation eines analogen Zeitsignals mit einem Dirac-Kamm beschrieben:

$$y_n = y(t) \cdot \text{comb}(t, \tau) \quad (9)$$

$$\text{comb}(t, \tau) = \sum_{k=-\infty}^{\infty} \delta(t - k\tau). \quad (10)$$

Das Spektrum der abgetasteten (digitalen) Funktion y_n ergibt sich demnach aus der Faltung des Spektrums der ursprünglichen Zeitfunktion $y(t)$ mit dem Spektrum des Dirac-Kamms. Das Spektrum einer Dirac-Kammfunktion ist wiederum ein Dirac-Kamm mit der Periode $f_s = 1/\tau$. Wenn wir mit \mapsto die Fouriertransformierte bezeichnen, gilt also für das Spektrum des abgetasteten Signals:

$$\begin{aligned} y(t) &\mapsto Y(f) \\ \text{comb}(t, \tau) &\mapsto \frac{1}{\tau} \text{comb}(f, f_s) \\ y_n(t) &\mapsto Y_n(f) = \frac{1}{\tau} \sum_{k=-\infty}^{\infty} Y(f - kf_s). \end{aligned} \quad (11)$$

Mit anderen Worten: Das Spektrum des Originalsignals wiederholt sich im Frequenzbereich periodisch mit der Abtastfrequenz f_s . Besitzt nun das Spektrum der Zeitfunktion $Y(f)$ Frequenzen oberhalb der halben Abtastfrequenz f_s , so kann das digitale Signal in das zugehörige analoge Zeitsignal

zurückgewandelt werden, indem man es digital-analog wandelt und mit einem Tiefpaß mit einer Abschneidefrequenz von $f_s/2$ die periodischen hochfrequenten Anteile herausfiltert. Bei der digitalen Signalverarbeitung muß also das abzutastende Signal mit $f_s/2$ bandbegrenzt sein.

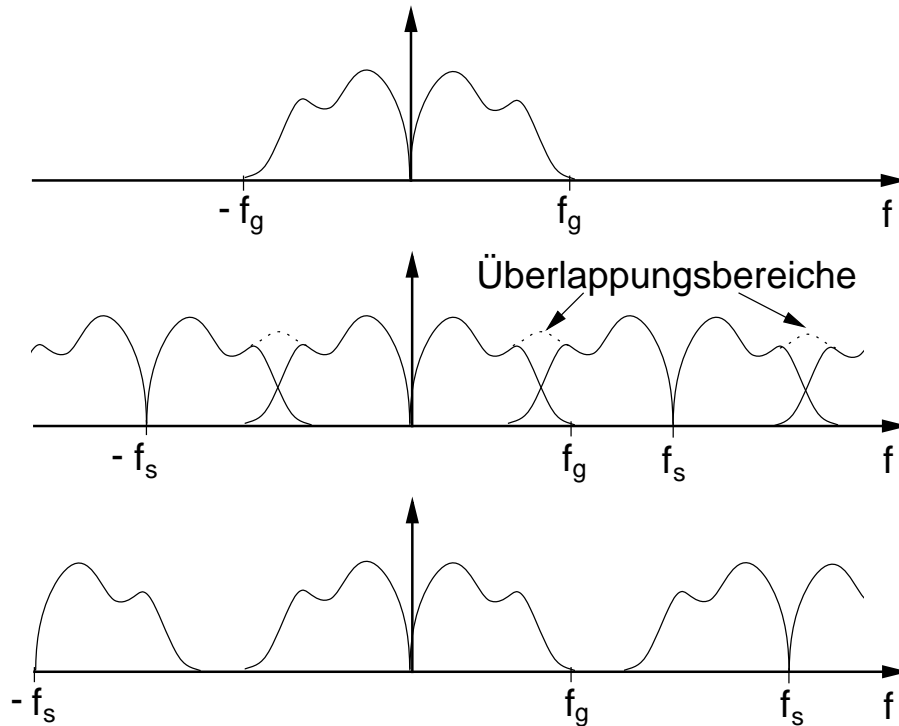


Abb. 3: Zum Abtasttheorem. Oben: Spektrum der Zeitfunktion. Mitte: Spektrum der abgetasteten Funktion mit $f_s < 2f_g$. Unten: dasselbe mit $f_s > 2f_g$.

1.3 Digitale Systeme

Nachdem wir die Grundlagen der Beziehung zwischen einem analogen und einem digitalen Signal besprochen haben, interessieren uns die Operationen, die mit den digitalen Signalen z.B. in einem Digitalrechner durchgeführt werden können. Die wichtigsten sind hierbei die linearen digitalen Systeme, die sich durch folgende Differenzgleichung beschreiben lassen:

$$y_n = \sum_{k=0}^M \alpha_k x_{n-k} + \sum_{k=1}^N \beta_k y_{n-k}. \quad (12)$$

Der aktuelle Ausgangswert y_n berechnet sich also aus N vergangenen Ausgangswerten (y_{n-1}, \dots, y_{n-N}) und $M + 1$ vergangenen Eingangswerten (x_n, \dots, x_{n-M}).

Ein lineares digitales System kann somit aus den drei Elementen *Addierer*, *Multiplizierer* und *Verzögerer* aufgebaut werden, die sich mit einem Digitalrechner realisieren lassen.

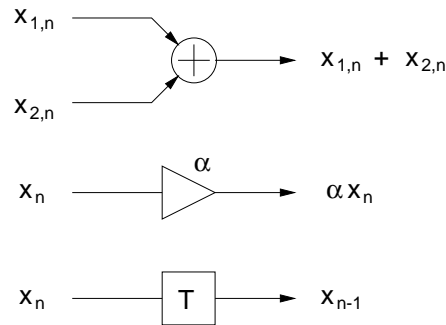


Abb. 4: Die Elemente Addierer, Multiplizierer und Verzögerer.

Eine wichtige Größe zur Beschreibung eines digitalen Systems ist die *Einheitsimpulsantwort*. Dies ist die Ausgangsfolge des Systems bei Anregung mit einer Eingangsfolge, die aus einer Eins und sonst nur aus Nullen besteht. Aus der Einheitsimpulsantwort h_n läßt sich die Ausgangsfolge y_n des Systems bei einer beliebigen Eingangsfolge x_n berechnen:

$$y_n = \sum_{k=-\infty}^{\infty} h_k x_{n-k}. \quad (13)$$

Diese spezielle Summe bezeichnet man als *Faltungssumme*.

Als Beispiel betrachte man das folgende System 1. Ordnung:

$$y_n = x_n + \frac{1}{2}y_{n-1}. \quad (14)$$

Wie man leicht nachrechnet ist die Einheitsimpulsantwort h_n

$$h_n = 1, \quad \frac{1}{2}, \quad \frac{1}{4}, \quad \frac{1}{8}, \dots \quad (15)$$

Im weiteren interessieren wir uns für das Frequenzverhalten dieses Systems, d.h. für Betrag und Phase der Ausgangsfunktion bei einer harmonischen Eingangsfunktion $e^{i\omega t}$ mit der variablen (Kreis-)Frequenz ω . Die diskrete Eingangsfunktion x_n errechnet sich dann zu

$$x_n = e^{i\omega\tau n} \quad \tau = T_s : \text{Abtastperiode.} \quad (16)$$

Weil es sich um ein lineares System handelt, ist die Ausgangsfunktion ebenfalls eine harmonische Funktion mit der Frequenz ω , so daß gilt:

$$y_{n-1} = e^{i\omega\tau} y_n$$

$$y_n = x_n + \frac{1}{2}e^{-i\omega\tau} y_n \quad (17)$$

$$y_n = \frac{1}{1 - \frac{1}{2}e^{-i\omega\tau}} x_n.$$

Die Übertragungsfunktion $H(\omega)$ ist nun der Quotient aus Eingangs- und Ausgangsfunktion bei harmonischer Anregung:

$$H(\omega) = \frac{y_n}{x_n} = \frac{1}{1 - \frac{1}{2}e^{-i\omega\tau}}. \quad (18)$$

In unserem Beispiel kann nun für beliebige reelle Signale die Übertragungsfunktion $H(\omega)$ nach Betrag und Phase ausgewertet werden: Für tiefe Frequenzen ($\omega \approx 0$) ist der Betrag maximal, und für $\omega = \pi/\tau$ ist der Betrag minimal, so daß sich unterhalb dieser Frequenz eine Tiefpaßcharakteristik ergibt.

Im allgemeinen ist die Übertragungsfunktion als das Verhältnis der Fouriertransformierten von Ausgangsfolge ($F(y_n)$) und der Eingangsfolge ($F(x_n)$) definiert:

$$H(\omega) = \frac{F(y_n)}{F(x_n)} = F(h_n). \quad (19)$$

Die Berechnung der Übertragungsfunktion nach dieser Formel kann sehr aufwendig sein. Der oben angewandte "Trick" zur einfachen Berechnung der Übertragungsfunktion durch harmonische Eingangs- und Ausgangsfunktionen kann jedoch auch für ein beliebiges lineares digitales System benutzt werden. Ausgehend von der Differenzgleichung

$$y_n = \sum_{k=0}^M \alpha_k x_{n-k} + \sum_{k=1}^N \beta_k y_{n-k} \quad (20)$$

läßt sich bei harmonischem Eingangssignal $x_n = e^{i\omega\tau n}$ das Ausgangssignal ausdrücken als

$$y_n = \sum_{k=0}^M \alpha_k e^{-i\omega\tau k} x_n + \sum_{k=1}^N \beta_k e^{-i\omega\tau k} y_n, \quad (21)$$

also

$$\frac{y_n}{x_n} = \frac{\sum_{k=0}^M \alpha_k e^{-i\omega\tau k}}{1 - \sum_{k=1}^N \beta_k e^{-i\omega\tau k}} = H(\omega, \tau) = H(e^{i\omega\tau}). \quad (22)$$

Die Übertragungsfunktion läßt sich also direkt durch die Koeffizienten α_k und β_k ausdrücken. Für ein digitales System ist es daher sinnvoll, anstelle der Abhängigkeit von ω und τ nur die Abhängigkeit von der zusammengesetzten Größe $z = e^{i\omega\tau}$ zu betrachten:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^M \alpha_k z^{-k}}{1 - \sum_{k=1}^N \beta_k z^{-k}}. \quad (23)$$

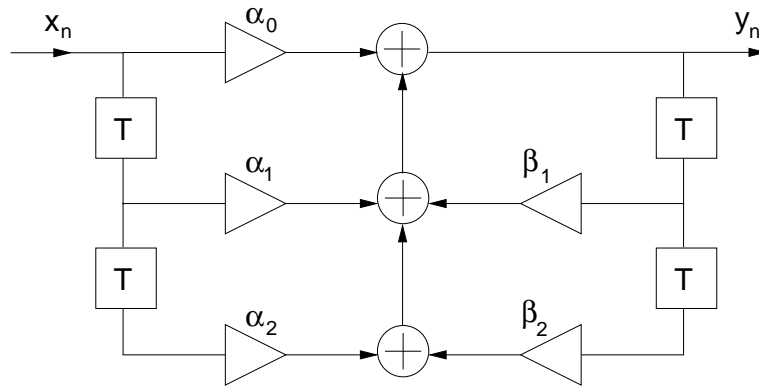


Abb. 5: Digitales System 2. Ordnung

Die Rechenvorschrift, einer zeitvariablen Folge x_n eine frequenzabhängige Funktion $X(z)$ zuzuordnen, wird als *Z-Transformation* bezeichnet. Sie ist mit der Fourier-Transformation (bzw. deren Verallgemeinerung, der Laplace-Transformation) identisch, wenn die richtige Variablenbeziehung $z = e^{i\omega\tau}$ berücksichtigt wird. Für digitale Systeme ist die Z-Transformation jedoch formal einfacher, weil die Koeffizienten der Zeitfolge (Differenzgleichung) mit den Koeffizienten der Z-Transformierten übereinstimmen, die Abhängigkeit von ω und τ in einer Variablen vereinigt ist und die aus der Periodizität der e -Funktion resultierende Mehrdeutigkeit eliminiert wird. Allgemein ist die Z-Transformation einer Zahlenfolge $\{y_k\}$ folgendermaßen definiert:

$$Z\{y_n\} = Y(z) = \sum_{k=0}^{\infty} y_k z^{-k}. \quad (24)$$

In unserem Beispiel errechnete sich die Übertragungsfunktion nach (18) als

$$H(\omega) = \frac{1}{1 - \frac{1}{2}e^{-i\omega\tau}} = \frac{1}{1 - \frac{1}{2}z^{-1}}. \quad (25)$$

Wir können sie aber auch mit Hilfe von (24) aus der Einheitsimpulsantwort h_n berechnen, da $H(z) = Z\{h_n\}$ gilt:

$$h_n = \left(\frac{1}{2}\right)^n, \quad n \geq 0, \quad (26)$$

$$H(z) = \sum_{n=0}^{\infty} h_n z^{-n} = \sum_{n=0}^{\infty} \left(\frac{1}{2}z^{-1}\right)^n = \frac{1}{1 - \frac{1}{2}z^{-1}}. \quad (27)$$

1.4 Pol-Nullstellen-Darstellung der Systemfunktion

Man betrachte das folgende System 2. Ordnung:

$$y_n = \alpha_0 x_n + \alpha_1 x_{n-1} + \alpha_2 x_{n-2} + \beta_1 y_{n-1} + \beta_2 y_{n-2}. \quad (28)$$

Dies läßt sich durch eine Schaltung wie in Abbildung 5 realisieren. Die Übertragungsfunktion lautet

$$H(z) = \frac{\alpha_0 + \alpha_1 z^{-1} + \alpha_2 z^{-2}}{1 - \beta_1 z^{-1} - \beta_2 z^{-2}} = \frac{\alpha_0 z^2 + \alpha_1 z + \alpha_2}{z^2 - \beta_1 z - \beta_2}. \quad (29)$$

Die Polynome der Systemfunktion können auch in Produktform geschrieben werden:

$$H(z) = A \frac{(z - z_{1,0})(z - z_{2,0})}{(z - z_{1,\infty})(z - z_{2,\infty})}. \quad (30)$$

Die Nullstellen des Zählerpolynoms ($z_{1,0}, z_{2,0}$) werden als Nullstellen, die Nullstellen des Nenners ($z_{1,\infty}, z_{2,\infty}$) als Pole des Systems bezeichnet.

Aus den Polen und Nullstellen lassen sich mit einfachen geometrischen Methoden Betrag und Phase der Übertragungsfunktion des Systems herleiten. Den Betrag erhalten wir, indem wir auf dem Einheitskreis in der komplexen Ebene ($z = e^{j\omega T}$) "entlangfahren" und die Abstände zwischen dem jeweiligen Punkt auf dem Einheitskreis und den Nullstellen multiplizieren. Dieses Produkt muß noch durch die Abstände zwischen dem jeweiligen Punkt und den Polen dividiert werden. Die Phase erhält man entsprechend durch Addition der Winkel zu den Nullstellen und Subtraktion der Winkel zu den Polen.

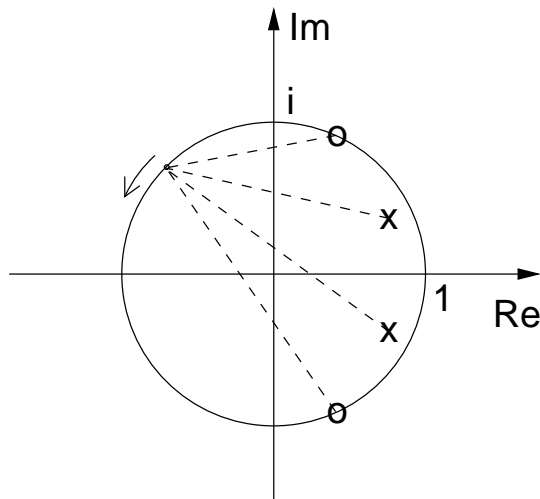


Abb. 6: Beispiel zur geometrischen Abschätzung einer Übertragungsfunktion aus den Nullstellen (o) und Polen (x).

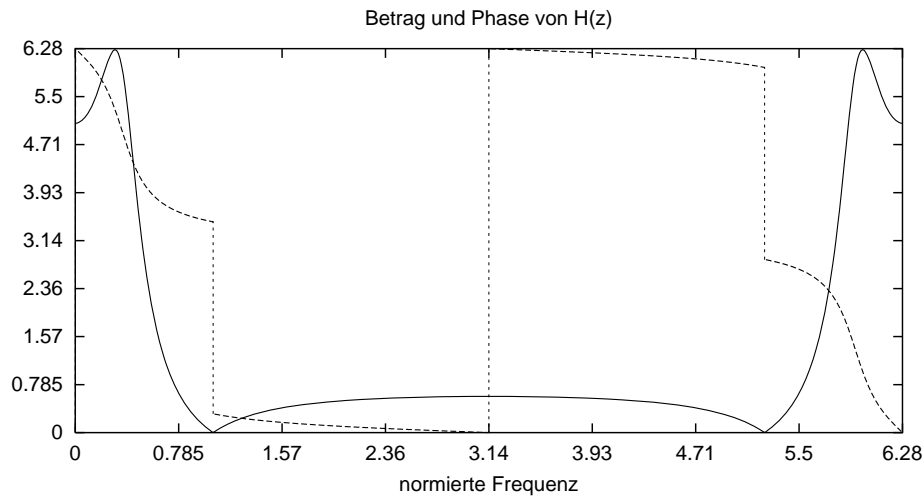


Abb. 7: Verlauf von $|H(z)|$ (normiert auf 2π , durchgezogene Linie) und der Phase von $H(z)$ ($\arg[H(z)]$, gestrichelte Linie) zu dem Beispiel in Abb. 6.

1.5 Spezielle Filter

Üblicherweise werden Filter in der komplexen p -Ebene, d.h. im Definitionsbereich der Laplace-Transformation entworfen. Als Beispiel betrachte man den analog realisierten Tiefpaß 2. Ordnung, den Abb. 8 zeigt.

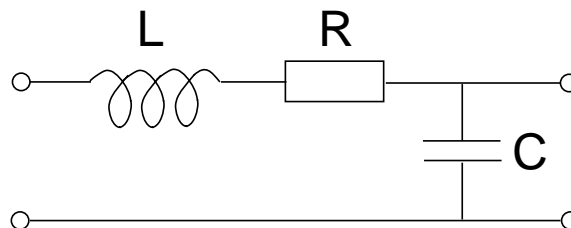


Abb. 8: Analoger RCL-Tiefpaß.

Dieser Tiefpaß hat die Übertragungsfunktion

$$H(\omega) = \frac{1}{1 + RCi\omega - LC\omega^2}, \quad (31)$$

oder bei Verallgemeinerung der rein imaginären Variablen $i\omega$ auf eine komplexe Variable p :

$$H(p) = \frac{1}{1 + RCp + LCp^2}. \quad (32)$$

Man erkennt, daß sich die Übertragungsfunktion eines solchen Tiefpasses durch ein Nennerpolynom 2. Ordnung auszeichnet. Durch geeignete Wahl der Koeffizienten lassen sich Tiefpaßfilter mit unterschiedlichen Eigenschaften herstellen.

Um ein äquivalentes Filter digital zu realisieren, muß man eine Variablentransformation durchführen, die die p -Ebene auf die z -Ebene abbildet. Bei dieser Transformation muß die i -Achse auf den Einheitskreis abgebildet werden. Dies leistet die sogenannte *Bilineartransformation*:

$$p \rightarrow \frac{2(1-z)}{\tau(1+z)}. \quad (33)$$

Betrachten wir wieder einen Tiefpaß:

$$H(p) = \frac{1}{1 + b_1 p + b_2 p^2}. \quad (34)$$

Die Bilineartransformation liefert:

$$H(z) = A \frac{1 + 2z + z^2}{1 + \alpha_1 z + \alpha_2 z^2}. \quad (35)$$

Diese Systemfunktion besitzt zwei Nullstellen bei $z = -1$ und zwei Pole, deren Lage die Abschneidefrequenz, die Flankensteilheit und die maximale Dämpfung des Filters beschreiben.

Auf einfache Weise erhalten wir aus einem gegebenen Tiefpaß einen Hochpaß, indem wir z mit -1 multiplizieren. Dies entspricht der Transformation $\omega \rightarrow \pi f_s - \omega$, d.h. einer Spiegelung des Frequenzbereichs an der halben Abtastfrequenz, und liefert in unserem Fall zwei Nullstellen bei $z = 1$. Andere Transformationen, z.B. Tiefpaß-Bandpaß-Transformationen, sind möglich. Auf sie wird hier nicht weiter eingegangen, sondern auf die Literatur verwiesen.

Der konkrete Entwurf eines digitalen Filters sei am Beispiel eines *Butterworth*-Tiefpasses 2. Ordnung gezeigt. Bei diesem Filtertyp versucht man, den Frequenzgang im Durchlaßbereich möglichst lange konstant zu halten. Erst ab einer Grenzfrequenz ω_g wird gesperrt. Dies erreicht man durch die Übertragungsfunktion

$$|H(P)|^2 = \frac{1}{1 + P^{2n}} \quad (36)$$

mit der normierten Variablen $P = \frac{i\omega}{\omega_g}$ und der Ordnungszahl n . Für $\omega = \omega_g$ wird das Betragsquadrat $|H(P)|^2 = \frac{1}{2}$. Für ein Filter braucht man jedoch die

komplexe Übertragungsfunktion. Bei $n = 2$ ist diese gegeben durch

$$H(P) = \frac{1}{1 + \sqrt{2}P + P^2}. \quad (37)$$

Die Bilineartransformation (33) verursacht eine tangensförmige Frequenzverzerrung

$$\omega = \frac{2}{\tau} \tan\left(\omega' \frac{\tau}{2}\right), \quad (38)$$

wobei ω' die analoge Frequenz bezeichnet. Damit die Grenzfrequenz im analogen und im digitalen Fall gleich ist, muß man wie folgt korrigieren:

$$\omega = k \frac{2}{\tau} \tan\left(\omega' \frac{\tau}{2}\right) \quad (39)$$

mit

$$k = \omega_g \frac{\tau}{2} \cot\left(\omega_g \frac{\tau}{2}\right). \quad (40)$$

Dies liefert dann die folgende modifizierte Bilineartransformation:

$$P \rightarrow \cot\left(\omega_g \frac{\tau}{2}\right) \frac{1-z}{1+z}. \quad (41)$$

Setzt man etwa $f_g = 1$ kHz und $f_s = 1/\tau = 10$ kHz, so erhält man das folgende digitale *Butterworth*-Filter:

$$H(z) = 0.06743 \frac{1 + 2z + z^2}{0.428 - 1.143z + z^2}. \quad (42)$$

SETTING		CTL4	0	0	0	0	1	1	1	1
		CTL5	0	0	1	1	0	0	1	1
CTL1	CTL2	CTL6	0	1	0	1	0	1	0	1
		CTL3								
0	0	0	1M	100 K	10 K	1 K	100	10	1	1/10
0	0	1	100 K	10 K	1 K	100	10	1	1/10	1/100
0	1	0	500 K	50 K	5 K	500	50	5	1/2	1/20
0	1	1	333.3K	33.3 K	3.3 K	333.3	33.3	3.33	1/3	1/30
1	0	0	250 K	25 K	2.5 K	250	25	2.5	1/4	1/40
1	0	1	200 K	20 K	2 K	200	20	2	1/5	1/50
1	1	0	166.6K	16.6 K	1.6 K	166.6	16.6	1.6	1/6	1/60
1	1	1	83.3 K	8.3 K	833.3	83.3	8.3	0.83	1/12	1/120

CTL1 CTL2 CTL3 CTL4 CTL5 CTL6
 ○ ○ ○ ○ ○ ○

Einstellung der Abtastfrequenz: z.B. 10 kHz 0 0 1 0 0 1

Abb. 9: Tabelle der Schalterstellungen für verschiedene Sample-Frequenzen.

Nächste zwei Seiten:
 Schalterstellungen für die Koeffizienten α_i, β_j .

1 0 0 0 0 0 0 0	-2.00000	1 1 0 0 0 0 0 0	-1.00000
1 0 0 0 0 0 0 1	-1.98438	1 1 0 0 0 0 0 1	-0.98438
1 0 0 0 0 0 1 0	-1.96875	1 1 0 0 0 0 1 0	-0.96875
1 0 0 0 0 0 1 1	-1.95312	1 1 0 0 0 0 1 1	-0.95312
1 0 0 0 0 1 0 0	-1.93750	1 1 0 0 0 1 0 0	-0.93750
1 0 0 0 0 1 0 1	-1.92188	1 1 0 0 0 1 0 1	-0.92188
1 0 0 0 0 1 1 0	-1.90625	1 1 0 0 0 1 1 0	-0.90625
1 0 0 0 0 1 1 1	-1.89062	1 1 0 0 0 1 1 1	-0.89062
1 0 0 0 1 0 0 0	-1.87500	1 1 0 0 1 0 0 0	-0.87500
1 0 0 0 1 0 0 1	-1.85938	1 1 0 0 1 0 0 1	-0.85938
1 0 0 0 1 0 1 0	-1.84375	1 1 0 0 1 0 1 0	-0.84375
1 0 0 0 1 0 1 1	-1.82812	1 1 0 0 1 0 1 1	-0.82812
1 0 0 0 1 1 0 0	-1.81250	1 1 0 0 1 1 0 0	-0.81250
1 0 0 0 1 1 0 1	-1.79688	1 1 0 0 1 1 0 1	-0.79688
1 0 0 0 1 1 1 0	-1.78125	1 1 0 0 1 1 1 0	-0.78125
1 0 0 0 1 1 1 1	-1.76562	1 1 0 0 1 1 1 1	-0.76562
1 0 0 1 0 0 0 0	-1.75000	1 1 0 1 0 0 0 0	-0.75000
1 0 0 1 0 0 0 1	-1.73438	1 1 0 1 0 0 0 1	-0.73438
1 0 0 1 0 0 1 0	-1.71875	1 1 0 1 0 0 1 0	-0.71875
1 0 0 1 0 0 1 1	-1.70312	1 1 0 1 0 0 1 1	-0.70312
1 0 0 1 0 1 0 0	-1.68750	1 1 0 1 0 1 0 0	-0.68750
1 0 0 1 0 1 0 1	-1.67188	1 1 0 1 0 1 0 1	-0.67188
1 0 0 1 0 1 1 0	-1.65625	1 1 0 1 0 1 1 0	-0.65625
1 0 0 1 0 1 1 1	-1.64062	1 1 0 1 0 1 1 1	-0.64062
1 0 0 1 1 0 0 0	-1.62500	1 1 0 1 1 0 0 0	-0.62500
1 0 0 1 1 0 0 1	-1.60938	1 1 0 1 1 0 0 1	-0.60938
1 0 0 1 1 0 1 0	-1.59375	1 1 0 1 1 0 1 0	-0.59375
1 0 0 1 1 0 1 1	-1.57812	1 1 0 1 1 0 1 1	-0.57812
1 0 0 1 1 1 0 0	-1.56250	1 1 0 1 1 1 0 0	-0.56250
1 0 0 1 1 1 0 1	-1.54688	1 1 0 1 1 1 0 1	-0.54688
1 0 0 1 1 1 1 0	-1.53125	1 1 0 1 1 1 1 0	-0.53125
1 0 0 1 1 1 1 1	-1.51562	1 1 0 1 1 1 1 1	-0.51562
1 0 1 0 0 0 0 0	-1.50000	1 1 1 0 0 0 0 0	-0.50000
1 0 1 0 0 0 0 1	-1.48438	1 1 1 0 0 0 0 1	-0.48438
1 0 1 0 0 0 1 0	-1.46875	1 1 1 0 0 0 1 0	-0.46875
1 0 1 0 0 0 1 1	-1.45312	1 1 1 0 0 0 1 1	-0.45312
1 0 1 0 0 1 0 0	-1.43750	1 1 1 0 0 1 0 0	-0.43750
1 0 1 0 0 1 0 1	-1.42188	1 1 1 0 0 1 0 1	-0.42188
1 0 1 0 0 1 1 0	-1.40625	1 1 1 0 0 1 1 0	-0.40625
1 0 1 0 0 1 1 1	-1.39062	1 1 1 0 0 1 1 1	-0.39062
1 0 1 0 1 0 0 0	-1.37500	1 1 1 0 1 0 0 0	-0.37500
1 0 1 0 1 0 0 1	-1.35938	1 1 1 0 1 0 0 1	-0.35938
1 0 1 0 1 0 1 0	-1.34375	1 1 1 0 1 0 1 0	-0.34375
1 0 1 0 1 0 1 1	-1.32812	1 1 1 0 1 0 1 1	-0.32812
1 0 1 0 1 1 0 0	-1.31250	1 1 1 0 1 1 0 0	-0.31250
1 0 1 0 1 1 0 1	-1.29688	1 1 1 0 1 1 0 1	-0.29688
1 0 1 0 1 1 1 0	-1.28125	1 1 1 0 1 1 1 0	-0.28125
1 0 1 0 1 1 1 1	-1.26562	1 1 1 0 1 1 1 1	-0.26562
1 0 1 1 0 0 0 0	-1.25000	1 1 1 1 0 0 0 0	-0.25000
1 0 1 1 0 0 0 1	-1.23438	1 1 1 1 0 0 0 1	-0.23438
1 0 1 1 0 0 1 0	-1.21875	1 1 1 1 0 0 1 0	-0.21875
1 0 1 1 0 0 1 1	-1.20312	1 1 1 1 0 0 1 1	-0.20312
1 0 1 1 0 1 0 0	-1.18750	1 1 1 1 0 1 0 0	-0.18750
1 0 1 1 0 1 0 1	-1.17188	1 1 1 1 0 1 0 1	-0.17188
1 0 1 1 0 1 1 0	-1.15625	1 1 1 1 0 1 1 0	-0.15625
1 0 1 1 0 1 1 1	-1.14062	1 1 1 1 0 1 1 1	-0.14062
1 0 1 1 1 0 0 0	-1.12500	1 1 1 1 1 0 0 0	-0.12500
1 0 1 1 1 0 0 1	-1.10938	1 1 1 1 1 0 0 1	-0.10938
1 0 1 1 1 0 1 0	-1.09375	1 1 1 1 1 0 1 0	-0.09375
1 0 1 1 1 0 1 1	-1.07812	1 1 1 1 1 0 1 1	-0.07812
1 0 1 1 1 1 0 0	-1.06250	1 1 1 1 1 1 0 0	-0.06250
1 0 1 1 1 1 0 1	-1.04688	1 1 1 1 1 1 0 1	-0.04688
1 0 1 1 1 1 1 0	-1.03125	1 1 1 1 1 1 1 0	-0.03125
1 0 1 1 1 1 1 1	-1.01562	1 1 1 1 1 1 1 1	-0.01562

0 0 0 0 0 0 0 0	0.00000	0 1 0 0 0 0 0 0	1.00000
0 0 0 0 0 0 0 1	0.01562	0 1 0 0 0 0 0 1	1.01562
0 0 0 0 0 0 1 0	0.03125	0 1 0 0 0 0 1 0	1.03125
0 0 0 0 0 0 1 1	0.04688	0 1 0 0 0 0 1 1	1.04688
0 0 0 0 0 1 0 0	0.06250	0 1 0 0 0 1 0 0	1.06250
0 0 0 0 0 1 0 1	0.07812	0 1 0 0 0 1 0 1	1.07812
0 0 0 0 0 1 1 0	0.09375	0 1 0 0 0 1 1 0	1.09375
0 0 0 0 0 1 1 1	0.10938	0 1 0 0 0 1 1 1	1.10938
0 0 0 0 1 0 0 0	0.12500	0 1 0 0 1 0 0 0	1.12500
0 0 0 0 1 0 0 1	0.14062	0 1 0 0 1 0 0 1	1.14062
0 0 0 0 1 0 1 0	0.15625	0 1 0 0 1 0 1 0	1.15625
0 0 0 0 1 0 1 1	0.17188	0 1 0 0 1 0 1 1	1.17188
0 0 0 0 1 1 0 0	0.18750	0 1 0 0 1 1 0 0	1.18750
0 0 0 0 1 1 0 1	0.20312	0 1 0 0 1 1 0 1	1.20312
0 0 0 0 1 1 1 0	0.21875	0 1 0 0 1 1 1 0	1.21875
0 0 0 0 1 1 1 1	0.23438	0 1 0 0 1 1 1 1	1.23438
0 0 0 1 0 0 0 0	0.25000	0 1 0 1 0 0 0 0	1.25000
0 0 0 1 0 0 0 1	0.26562	0 1 0 1 0 0 0 1	1.26562
0 0 0 1 0 0 1 0	0.28125	0 1 0 1 0 0 1 0	1.28125
0 0 0 1 0 0 1 1	0.29688	0 1 0 1 0 0 1 1	1.29688
0 0 0 1 0 1 0 0	0.31250	0 1 0 1 0 1 0 0	1.31250
0 0 0 1 0 1 0 1	0.32812	0 1 0 1 0 1 0 1	1.32812
0 0 0 1 0 1 1 0	0.34375	0 1 0 1 0 1 1 0	1.34375
0 0 0 1 0 1 1 1	0.35938	0 1 0 1 0 1 1 1	1.35938
0 0 0 1 1 0 0 0	0.37500	0 1 0 1 1 0 0 0	1.37500
0 0 0 1 1 0 0 1	0.39062	0 1 0 1 1 0 0 1	1.39062
0 0 0 1 1 0 1 0	0.40625	0 1 0 1 1 0 1 0	1.40625
0 0 0 1 1 0 1 1	0.42188	0 1 0 1 1 0 1 1	1.42188
0 0 0 1 1 1 0 0	0.43750	0 1 0 1 1 1 0 0	1.43750
0 0 0 1 1 1 0 1	0.45312	0 1 0 1 1 1 0 1	1.45312
0 0 0 1 1 1 1 0	0.46875	0 1 0 1 1 1 1 0	1.46875
0 0 0 1 1 1 1 1	0.48438	0 1 0 1 1 1 1 1	1.48438
0 0 1 0 0 0 0 0	0.50000	0 1 1 0 0 0 0 0	1.50000
0 0 1 0 0 0 0 1	0.51562	0 1 1 0 0 0 0 1	1.51562
0 0 1 0 0 0 1 0	0.53125	0 1 1 0 0 0 1 0	1.53125
0 0 1 0 0 0 1 1	0.54688	0 1 1 0 0 0 1 1	1.54688
0 0 1 0 0 1 0 0	0.56250	0 1 1 0 0 1 0 0	1.56250
0 0 1 0 0 1 0 1	0.57812	0 1 1 0 0 1 0 1	1.57812
0 0 1 0 0 1 1 0	0.59375	0 1 1 0 0 1 1 0	1.59375
0 0 1 0 0 1 1 1	0.60938	0 1 1 0 0 1 1 1	1.60938
0 0 1 0 1 0 0 0	0.62500	0 1 1 0 1 0 0 0	1.62500
0 0 1 0 1 0 0 1	0.64062	0 1 1 0 1 0 0 1	1.64062
0 0 1 0 1 0 1 0	0.65625	0 1 1 0 1 0 1 0	1.65625
0 0 1 0 1 0 1 1	0.67188	0 1 1 0 1 0 1 1	1.67188
0 0 1 0 1 1 0 0	0.68750	0 1 1 0 1 1 0 0	1.68750
0 0 1 0 1 1 0 1	0.70312	0 1 1 0 1 1 0 1	1.70312
0 0 1 0 1 1 1 0	0.71875	0 1 1 0 1 1 1 0	1.71875
0 0 1 0 1 1 1 1	0.73438	0 1 1 0 1 1 1 1	1.73438
0 0 1 1 0 0 0 0	0.75000	0 1 1 1 0 0 0 0	1.75000
0 0 1 1 0 0 0 1	0.76562	0 1 1 1 0 0 0 1	1.76562
0 0 1 1 0 0 1 0	0.78125	0 1 1 1 0 0 1 0	1.78125
0 0 1 1 0 0 1 1	0.79688	0 1 1 1 0 0 1 1	1.79688
0 0 1 1 0 1 0 0	0.81250	0 1 1 1 0 1 0 0	1.81250
0 0 1 1 0 1 0 1	0.82812	0 1 1 1 0 1 0 1	1.82812
0 0 1 1 0 1 1 0	0.84375	0 1 1 1 0 1 1 0	1.84375
0 0 1 1 0 1 1 1	0.85938	0 1 1 1 0 1 1 1	1.85938
0 0 1 1 1 0 0 0	0.87500	0 1 1 1 1 0 0 0	1.87500
0 0 1 1 1 0 0 1	0.89062	0 1 1 1 1 0 0 1	1.89062
0 0 1 1 1 0 1 0	0.90625	0 1 1 1 1 0 1 0	1.90625
0 0 1 1 1 0 1 1	0.92188	0 1 1 1 1 0 1 1	1.92188
0 0 1 1 1 1 0 0	0.93750	0 1 1 1 1 1 0 0	1.93750
0 0 1 1 1 1 0 1	0.95312	0 1 1 1 1 1 0 1	1.95312
0 0 1 1 1 1 1 0	0.96875	0 1 1 1 1 1 1 0	1.96875
0 0 1 1 1 1 1 1	0.98438	0 1 1 1 1 1 1 1	1.98438