

1 Direkte Digitale Synthese

In diesem Aufsatz wird die direkte digitale Synthese von periodischen mathematischen Funktionen theoretisch und praktisch am Beispiel eines Sinusgenerators erklärt.

1.1 Theorie

Ein Sinus mit der Frequenz f_0 ist gegeben durch die Zeitfunktion

$$y(t) = A \cdot \sin(2\pi f_0 \cdot t) . \quad (1)$$

Für die digitale Signalverarbeitung muß diese kontinuierliche Darstellung in eine wert- und zeitdiskrete Darstellung umgewandelt werden.

Zeitdiskret bedeutet, daß das Signal nur zu den diskreten Zeitpunkten $k \cdot \Delta T$

$$y[k] = y(k \cdot \Delta T) = A \cdot \sin(2\pi \cdot f_0 \cdot k \cdot \Delta T) \quad (2)$$

besteht, wobei der Zeitschritt ΔT der Kehrwert der Abtast- bzw. Generierungsfrequenz f_T ist. Das ist die Frequenz, mit welcher der Sinus für die Digitalisierung abgetastet wird bzw. mit welcher bei der Generierung die Ausgangswerte erzeugt werden. Mit der Abtastfrequenz selbst ergibt sich

$$= A \cdot \sin\left(2\pi \cdot f_0 \cdot k \cdot \frac{1}{f_T}\right) \quad (3)$$

$$= A \cdot \sin\left(k \cdot 2\pi \cdot \underbrace{\frac{f_0}{f_T}}_{\Omega_0}\right) . \quad (4)$$

Die Größe Ω_0 ist die diskrete Kreisfrequenz des Sinus, eine wichtige Größe in der digitalen Signalverarbeitung, die der Kreisfrequenz ω bei zeitkontinuierlichen Signalen entspricht, etwa bei der Fouriertransformation.

Würde man einen N-Werte-Puffer mit den Werten einer Sinusperiode füllen

$$\text{sinus}_N[n] = A \cdot \sin\left(n \cdot \frac{2\pi}{N}\right), n \in [0, N - 1] \quad (5)$$

und die Werte dieses Puffers zyklisch mit der Frequenz f_T ausgeben, so erhielte man ein diskretes Sinussignal

$$y(k \cdot \Delta T) = A \cdot \sin\left(k \cdot \frac{2\pi}{N}\right) . \quad (6)$$

Ein Vergleich mit Gleichung 4 ergibt die Frequenz des Sinus zu

$$\frac{2\pi}{N} = 2\pi \cdot \frac{f_0}{f_T} \Rightarrow f_0 = \frac{f_T}{N} . \quad (7)$$

Will man eine andere Frequenz f_1 erzielen

$$y[k] = A \cdot \sin\left(2\pi \cdot k \cdot \frac{f_1}{f_T}\right) , \quad (8)$$

so läßt sich dies durch eine Reindizierung des Puffers erreichen

$$= A \cdot \sin \left(k \cdot \frac{2\pi}{N} \cdot \frac{f_1 \cdot N}{f_T} \right). \quad (9)$$

Statt mit einer Schrittweite von 1, wie in Gleichung 6, wird der Puffer nun mit einer Schrittweite $N \cdot \frac{f_1}{f_T}$ adressiert.

Ein Problem tritt auf, wenn die Schrittweite nicht ganzzahlig ist, etwa 1,4. Dann muß ein Wert zwischen zwei Pufferwerten interpoliert werden, z.B. für den Wert 1,4 zwischen den Werten 1 und 2, für den Wert 2,8 zwischen den Werten 2 und 3.

Die Interpolation selbst entspricht einer Rückgewinnung des zeitkontinuierlichen Signals aus seiner zeitdiskreten Version. Das Abtasttheorem von Shannon besagt, daß die Rückgewinnung des zeitkontinuierlichen Signals aus den Abtastwerten bei idealer Abtastung (s. [JN90]) fehlerfrei möglich ist, wenn die Abtastfrequenz größer ist als die doppelte Signalbandbreite.

Für den diskreten Sinus nach Gleichung 6 gilt für die Frequenz f_0 und damit die Bandbreite B die Relation $B = f_0 = \frac{f_T}{N}$ (s. Gleichung 7), die Forderung des Abtasttheorems $B \leq \frac{f_T}{2}$ ist somit für $N \geq 2$ erfüllt.

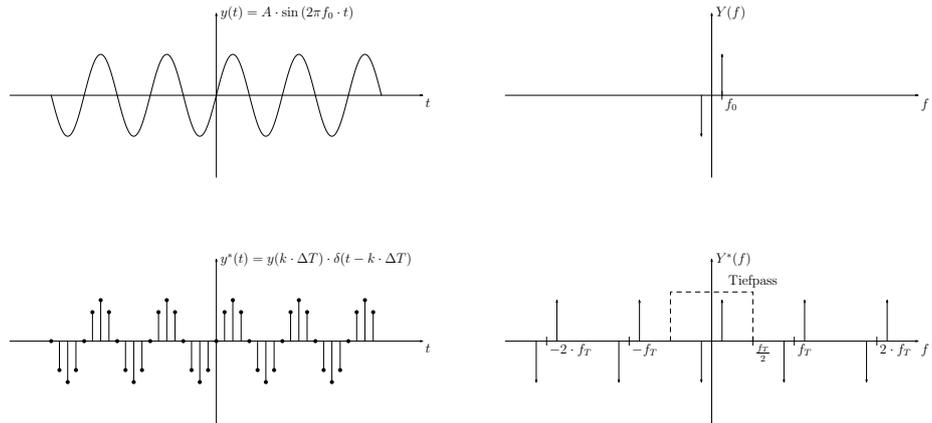


Abbildung 1: Ideale Abtastung eines Sinussignals im Zeit- und Frequenzbereich.

Die Forderung des Abtasttheorems wird aus Abbildung 1 verständlich. Idealerweise läßt sich die Abtastung als Multiplikation mit einem Deltakamm beschreiben

$$y^*(t) = y(t) \cdot \delta_{\frac{1}{f_T}}(t) \quad (10)$$

$$= \sum_{k=-\infty}^{\infty} y(k \cdot \Delta T) \cdot \delta(t - k \cdot \Delta T). \quad (11)$$

Im Frequenzbereich ergibt sich daraus

$$Y^*(j\omega) = \frac{1}{2\pi} \cdot Y(j\omega) * 2\pi f_T \cdot \delta_{2\pi \cdot f_T}(\omega) \quad (12)$$

$$= f_T \cdot \sum_{k=-\infty}^{\infty} Y(j(\omega - k \cdot 2\pi \cdot f_T)). \quad (13)$$

Im Frequenzbereich bewirkt die (ideale!) Abtastung also die spektrale Wiederholung des Spektrums des abgetasteten Signals mit der Abtastfrequenz plus einer Skalierung des Gesamtspektrums mit f_T .

Dies läßt sich mit einem Tiefpaß wieder rückgängig machen, indem man mit diesem einfach die 0. Wiederholung ausschneidet. Bedingung dafür ist, daß sich die spektralen Wiederholungen nicht überlappen. Dies ist gegeben, wenn die Abtastfrequenz mindestens doppelt so hoch ist wie die Signalbandbreite.

Die Verwendung eines idealen Tiefpasses

$$H_{TP,ideal}(j\omega) = \Pi_{2\pi f_T}(\omega) = \begin{cases} 1 & : |\omega| < 2\pi \frac{f_T}{2} \\ \frac{1}{2} & : |\omega| = 2\pi \frac{f_T}{2} \\ 0 & : |\omega| > 2\pi \frac{f_T}{2} \end{cases} \quad (14)$$

wie in Abbildung 1 führt zu einer perfekten Rekonstruktion des zeitkontinuierlichen Signals, da nach der Anwendung des Tiefpasses das Spektrum des gefilterten Signals mit dem Spektrum des zeitkontinuierlichen Signals identisch ist.

Die Anwendung des Tiefpasses entspricht einer Multiplikation der Spektren im Frequenzbereich und einer Faltung mit der Impulsantwort bzw. inversen Fouriertransformierten des Tiefpasses im Zeitbereich.

Diese ergibt sich mit Korrektur der Skalierung mit f_T zu

$$h_{TP,ideal}(t) = \frac{1}{2\pi} \cdot \int_{-\infty}^{\infty} H_{TP,ideal}(j\omega) \cdot e^{j\omega t} d\omega \quad (15)$$

$$= \frac{1}{2\pi} \cdot \int_{-2\pi \cdot \frac{f_T}{2}}^{2\pi \cdot \frac{f_T}{2}} \frac{1}{f_T} \cdot e^{j\omega t} d\omega \quad (16)$$

$$= \frac{1}{2\pi \cdot jt \cdot f_T} \cdot [e^{j\omega t}]_{-\pi \cdot f_T}^{\pi \cdot f_T} \quad (17)$$

$$= \frac{1}{\pi f_T \cdot t} \cdot \frac{1}{2j} \cdot (e^{j\pi f_T \cdot t} - e^{-j\pi f_T \cdot t}) \quad (18)$$

$$= \frac{1}{\pi f_T \cdot t} \cdot \sin(\pi f_T \cdot t), \quad (19)$$

woraus mit $si(x) = \frac{\sin(x)}{x}$ schließlich

$$= si(\pi f_T \cdot t) \quad (20)$$

wird. Eingesetzt in das Faltungsintegral

$$y_{rek.}(t) = \int_{-\infty}^{\infty} y^*(\tau) \cdot h_{TP,ideal}(t - \tau) d\tau \quad (21)$$

ergibt sich mit Gleichung 11

$$= \int_{-\infty}^{\infty} \sum_{k=-\infty}^{\infty} y(k \cdot \Delta T) \cdot \delta(t - k \cdot \Delta T) \cdot si(\pi f_T \cdot (t - \tau)) d\tau, \quad (22)$$

woraus nach Vertauschung von Summation und Integration, Ausnutzung der Symmetrie der $si(x)$ -Funktion und der Ausblendeigenschaft des $\delta(t)$ -Impulses

$$u(T) = \int_{-\infty}^{\infty} u(t) \cdot \delta(t - T) dt \quad (23)$$

schließlich die Interpolationsgleichung

$$y_{rek.}(t) = \sum_{k=-\infty}^{\infty} y(k \cdot \Delta T) \cdot si(\pi f_T \cdot (t - k \cdot \Delta T)), \Delta T = \frac{1}{f_T} \quad (24)$$

wird.

Da die $si(x)$ -Spaltfunktion schwer zu berechnen ist, zeitlich unendlich ausgedehnt ist und auch den ja erst zu berechnenden Sinus selbst enthält, muß ein einfacherer Tiefpaß verwendet werden.

Die einfachste Realisierung eines Tiefpasses im Zeitbereich ist das Halteglied mit der Impulsantwort

$$h_{TP, Halt}(t) = \square_{\Delta T} \left(t - \frac{\Delta T}{2} \right) = \begin{cases} 0 & : t < 0 \\ 1 & : t \in [0, \Delta T] \\ 0 & : t > \Delta T \end{cases} \quad (25)$$

In der Anwendung führt dieses Tiefpaß dazu, daß der letzte Wert solange gehalten wird, bis ein neuer Wert kommt. Der Index aus Gleichung 9 muß somit einfach auf den nächsten ganzzahligen Wert abgerundet werden.

Der Frequenzgang dieses Tiefpasses ergibt sich als Fouriertransformierte der Impulsantwort

$$H_{TP, Halt} = \mathfrak{F} \left\{ \square_{\Delta T} \left(t - \frac{\Delta T}{2} \right) \right\} \quad (26)$$

$$= \Delta T \cdot si \left(\omega \cdot \frac{\Delta T}{2} \right) \cdot e^{-j\omega \frac{\Delta T}{2}} \quad (27)$$

$$= \frac{1}{f_T} \cdot si \left(\frac{\omega}{2f_T} \right) \cdot e^{-j \frac{\omega}{2f_T}} \quad (28)$$

und ist, wie Abbildung 2 unten rechts zeigt, alles andere als ideal.

Neben dem erwünschten Sinus im Basisband entstehen Kosinusanteile als Oberwellen. Um diese zu unterdrücken, gibt es zwei Ansätze:

- Ein nachgeschaltetes analoges Tiefpaßfilter, welches ab $\frac{f_T}{2}$ zu sperren beginnt und/oder
- Eine möglichst hohe Puffergröße N und Verwendung nur geringer Frequenzen; somit bleiben die Spektrallinien der durch die Abtastung hervorgerufenen spektralen Wiederholungen im Bereich der Nulldurchgänge des Tiefpaß-Frequenzganges.

Eine weitere Möglichkeit besteht in der Verwendung eines besseren Interpolationsfilters als dem Halteglied. Eine vom Rechenaufwand noch akzeptable Möglichkeit ist die lineare Interpolation.

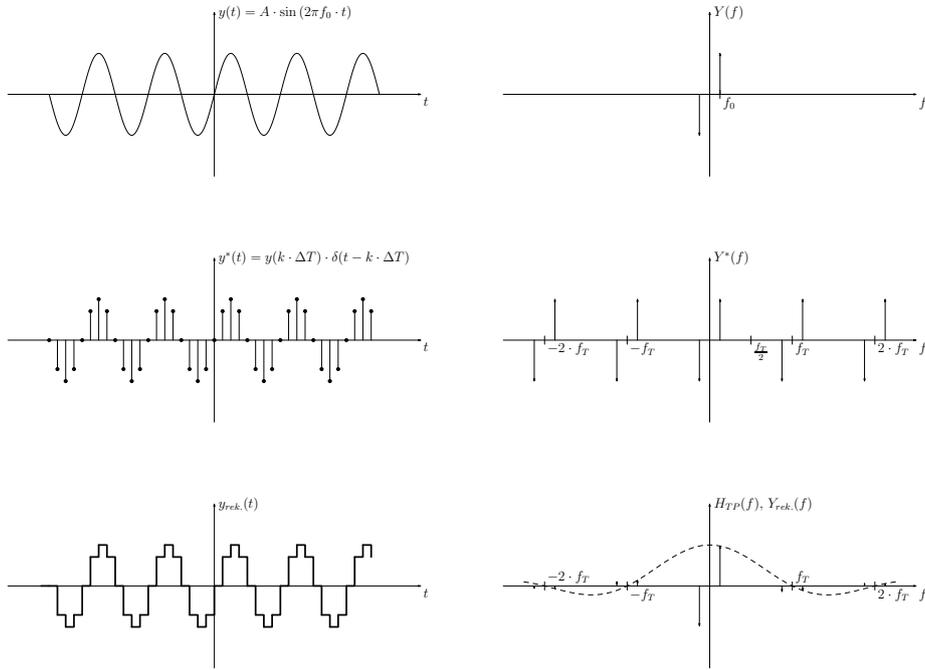


Abbildung 2: Rekonstruktion mit einem Halteglied als Tiefpaß.

Bei dieser wird der Wert $y(k \cdot \Delta T + t)$, $t \in [0, \Delta T]$ zwischen den Abtastwerten $y[k]$ und $y[k + 1]$ gemäß

$$y_{rek., lin.}(k \cdot \Delta T + t) = y[k] \cdot \left(1 - \frac{t}{\Delta T}\right) + y[k + 1] \cdot \frac{t}{\Delta T} \quad (29)$$

interpoliert. Das gesamte interpolierte Signal läßt sich als gewichtete Überlagerung von Dreiecken der Breite $2\Delta T$ beschreiben

$$y_{rek., lin.}(t) = \sum_{k=-\infty}^{\infty} y[k] \cdot \Delta_{2\Delta T}(t - k \cdot \Delta T). \quad (30)$$

Dies läßt sich auch als Faltung der Dreiecksfunktion mit dem ideal abgetasteten Signal beschreiben

$$= \Delta_{2\Delta T}(t) * \sum_{k=-\infty}^{\infty} y[k] \cdot \delta(t - k \cdot \Delta T). \quad (31)$$

Die Dreiecksfunktion ist somit die Impulsantwort des Tiefpaßfilters. Sie selbst läßt sich als das Produkt der Faltung zweier Rechteckfunktionen beschreiben

$$\Delta_{2\Delta T}(t) = h_{TP, lin.}(t) = \frac{1}{\Delta T} \cdot \Pi_{\Delta T}(t) * \Pi_{\Delta T}(t). \quad (32)$$

Daraus ergibt sich der Frequenzgang des Filters zu

$$H_{TP,lin.}(j\omega) = \frac{1}{\Delta T} \cdot \left(\text{sinc} \left(\omega \cdot \frac{\Delta T}{2} \right) \right)^2, \quad (33)$$

woraus mit $f_T = \frac{1}{\Delta T}$

$$= \frac{1}{f_T} \cdot \text{sinc}^2 \left(\frac{\omega}{2f_T} \right) \quad (34)$$

wird.

Ein Vergleich des Frequenzganges des Rekonstruktionstiefpasses bei der linearen Interpolation in Gleichung 34 mit dem des Halteglieds aus Gleichung 28 zeigt, daß ersterer im Vergleich eine quadratische Dämpfung der Oberwellen bewirkt. Dies wird auch aus Abbildung 3 deutlich.

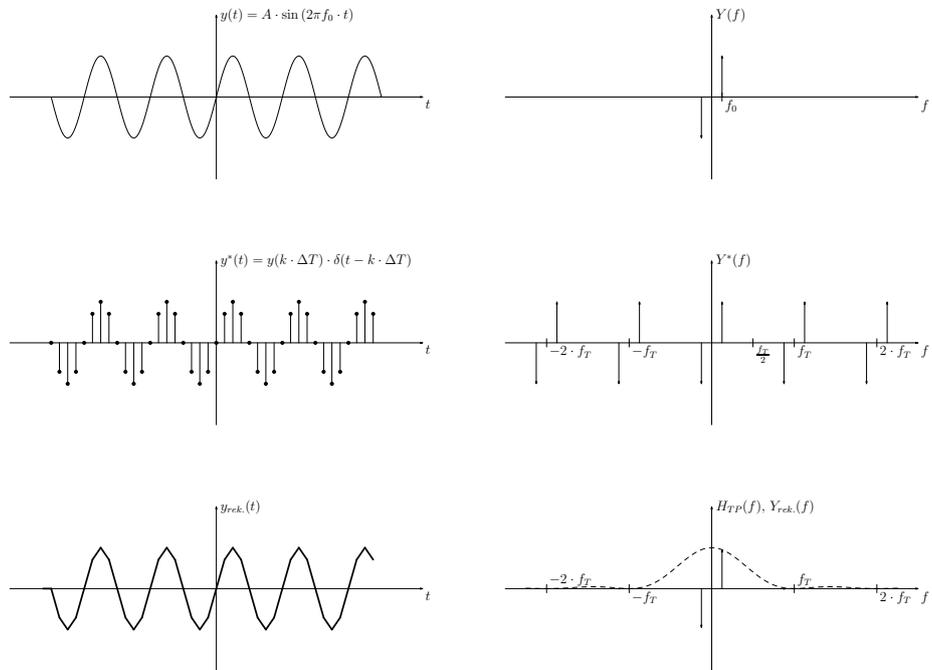


Abbildung 3: Rekonstruktion mittels linearer Interpolation.

1.2 Praktische Umsetzung mit Halteglied

Im folgenden wird die praktische Umsetzung unter Verwendung eines Halteglieds als Interpolationstiefpaß beschrieben.

Die Puffergröße beträgt $N = 256$, für den Pufferindex wurde eine 16-Bit-Festkommazahl mit 8 Bit Nachkommaanteil verwendet. Durch diese Wahl braucht keine Modulo-Operation zur Beschränkung auf den Pufferindexbereich durchgeführt werden, da dies implizit durch Überlauf im oberen Byte geschieht.

Das Abrunden kann durch ausschließliche Verwendung des oberen Bytes zur Adressierung des Puffers geschehen. Auf 8-Bit-Kernen läßt sich dies elegant

lösen, da die 16-Bit-Addition durch zwei sukzessive 8-Bit-Additionen, die zweite mit Carry-Übertrag, erfolgt. Somit braucht nur das die oberen 8 Bit beinhal-
tende Register für die Indizierung verwendet werden.

Für die AVR-8-Bit-Mikrocontroller konnte dies in C++ (avr-g++) durch die Verwendung des in Listing 1 abgedruckten Datentyps umgesetzt werden. Dieser Datentyp erlaubt 16-Bit-Additionen und gleichzeitigen Zugriff auf das untere und obere Byte.

Listing 1: Datentyp.

```
union word_t
{
    unsigned short word;
    struct byte_t
    {
        unsigned char low;
        unsigned char high;
    } byte;
} inkr, index;
```

Somit gestaltet sich die Generierung des Sinus recht einfach (s. Listing 2).

Listing 2: Generierung der Sinuswerte.

```
unsigned char operator () ()
{
    index.word += inkr.word;
    return sinbuffer[index.byte.high];
}
```

Die Berechnung des Inkrements erfolgt in dem Quellcodefragment in Listing 3. Übergeben wird der Methode ein Fließkommawert, welcher die gewünschte Frequenz in Relation zur maximal möglichen Frequenz, der halben Abtast/-Generierungsfrequenz $\frac{f_T}{2}$, angibt. Die Frequenz des generierten Sinus ist somit $f = f_{rel} \cdot \frac{f_T}{2}$, $f_{rel} < 1$.

Listing 3: Berechnung des Inkrements.

```
void setFrequency(double frel)
{
    inkr.word = (unsigned short) ((1 << 15) * frel);
}
```

Diese Quelltextfragmente sind einer C++-Klasse entnommen, welche die Funktionen mit den Pufferdaten, angelegt als statisches Feld, kombiniert. Auf diese Weise kann der Sinusgenerator effizient (keine Übergabe eines Verweises auf den Puffer), komfortabel (mögliche Methodenaufrufe werden von modernen IDE's angezeigt) und mit dem hohen Abstraktionsniveau der objektorientierten Programmierung realisiert werden.

Getestet wurde diese Klasse auf einem ATmega16, der mit $f_Q = 7,3728$ MHz getaktet wurde. Dessen 8-Bit-Zähler Timer0 wurde im nichtinvertierten PWM-Modus für die D/A-Wandlung verwendet (s. Listing 4), der zweite 8-Bit-Zähler Timer2 generierte den Abtasttakt im CTC-Modus gemäß

$$f_T = \frac{f_Q}{Vorteiler_2 \cdot (1 + OCR2)} \quad (35)$$

zu $f_T = 57,6$ kHz. Die Referenzfrequenz für den Sinusgenerator sind somit 28,8 kHz.

Mit der Abtastfrequenz wird die Interruptserviceroutine für den Compare-Interrupt des 2. Zeitgebers aktiviert, in welcher ein neuer Sinuswert berechnet und für die PWM eingestellt wird.

Nicht im Quelltext enthalten ist die Variation der Sinusfrequenz über die externen Interrupts bzw. 2 angeschlossene Taster.

Listing 4: Testprogramm auf einem ATmega16.

```

volatile double f=0.001;
const double df=0.0001;
// Sinus generation object
DigSinGenerator sinus(f);

...

/**
 * @brief Timer 2 Compare A interrupt service routine.
 * Signalgenerierung.
 */
ISR(TIMER2_COMP_vect)
{
    OCR0=sinus();
}

int main()
{
    DDRB = OUTPUT;
    // activate pull-up resistors
    PORTD = TASTER1 | TASTER2;
    // aktiviere INT0/1 interrupt auf fallender Flanke
    GICR = (1<<INT0) | (1<<INT1);
    OCR0 = 0;
    // Timer0: Nichtinvertierte PWM
    // Vorteiler Timer0 = 1
    TCCR0 = (1<<WGM01) | (1<<WGM00) | (1<<COM01) | (1<<CS00);
    OCR2 = 1; //  $f_T/2=28,8$  kHz ( $f_{osc}=7.3728$  MHz)
    // Timer2 im CTC-Modus, Vorteiler = 64
    TCCR2 = (1<<WGM21) | (1<<CS22);
    // Aktiviere Timer2 compare interrupt
    TIMSK = (1<<OCIE2);
    sei();
    while(1)
    {
        set_sleep_mode(SLEEP_MODE_IDLE);
        sleep_mode();
    }
}

```

Selbst mit einem improvisierten RC-Tiefpass ($R = 10$ k Ω , $C = 10$ nF) zur

Unterdrückung der (PWM-)Oberwellenanteile, welcher nicht auf $f_g = \frac{f_x}{2}$ bzw. $f_g = \frac{f_{osc}}{2 \cdot 256 \cdot V_{orteiler_1}}$ abgestimmt war, ergaben sich gute Ergebnisse (s. Abbildung 4).

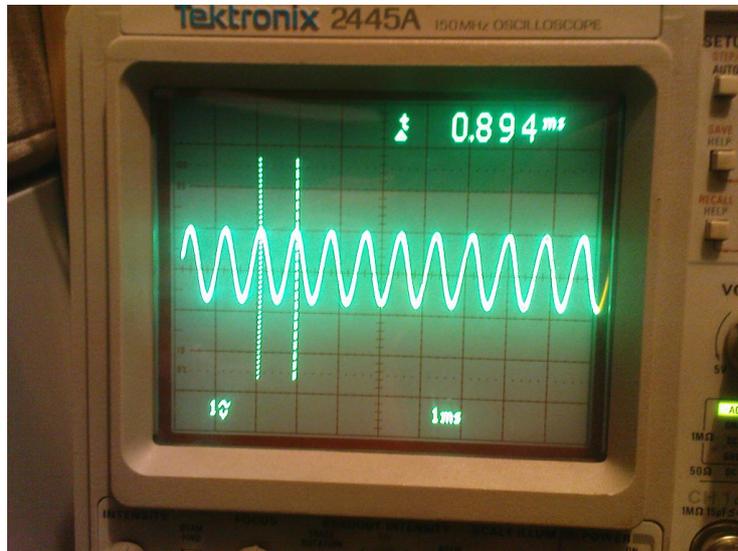


Abbildung 4: Erzeugter Sinus am Oszilloskop.

Literatur

- [JN90] Nuggehally S. Jayant and P. Noll. *Digital Coding of Waveforms: Principles and Applications to Speech and Video*. Prentice Hall Professional Technical Reference, 1990.