Analysis of a discrete time oscillator

Konijn, Victor J.H.

Award date:
1987

EINDHOVEN UNIVERSITY OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING

group measurement and control

ANALYSIS OF A DISCRETE TIME OSCILLATOR

by Victor J.H. Konijn

Report of a Master of Science (M.Sc. / ir.) project
carried out at the Central Application Laboratory CAB
Product Division Elcoma - Nederlandse Philips Bedrijven B.V.
from December 1986 to October 1987

Supervised by Prof.dr.ir. P. Eykhoff, T.U.E.
Coached by Ir. H.A.L. Piceni, T.U.E.
Ir. W.P.G. Crooijmans, C.A.B.

SUMMARY

A discrete time oscillator can be used to synthesize a wave shape with a certain stable frequency out of a (crystal) clock frequency. It can replace a voltage controlled oscillator in a phase lock loop. A model of a circuit is presented which contains behind the DTO a read only memory (ROM), a digital to analogue converter (DAC) and a reconstruction filter.

The model is used to describe the influence of the various parameters on system performance. A computer simulation package has been developed which calculates the exact amplitudes and phases of the sidebands in the DAC output signal and the resulting time jitter in the output signal of a reconstruction filter. Results have been verified by measurements on an existing DTO circuit.

PREFACE

This report describes the project which marks the end of my Master of Science education in electrical engineering at Eindhoven University of Technology. After several years of mainly theoretical education I had the opportunity to carry out this project in an industrial company. The project has therefore been born out of practical interest which has been a clear motivation for me.

The project has been carried out at the Digital Video Processing group of the Central Application Laboratory CAB. This laboratory is part of the Product Division Electronic Components and Materials (Elcoma) of the Nederlandse Philips Bedrijven B.V.. The group is involved in developments concerning the digitizing of components of present TV receivers. Although my project arose from this work, it actually deals with general digital signal processing. For this reason this report will probably and hopefully be useful to others as well.

I would like to thank the Philips company for giving me the opportunity to act like an employee for almost a year. Furthermore, the project would never have been completed the way it has been completed without the advice and assistance of many. I am therefore indebted to my professor, dr.ir. Piet Eykhoff, for his interest throughout the year and his painstaking reading of this report; my university coach, ir. Hans Piceni, for the detailed technical discussions we had resulting from several mid-term reports I overburdened him with; my daily coach and room-mate, ir. Wim Crooijmans, who managed in spite of his own busy working schedule to correct some of my crazy thoughts, to read and comment on all reports I wrote, who forced me to explain my ideas in an understandable way and who helped me to find my way in the Philips company. I am indebted to him and to my other room-mate, Hans van Weersch, for both general technical and all other not-to-describe-in-detail matters we discussed, which provided me with a good environment in which to work. Several discussions I have had with some other CAB employees, Ton Nillesen, Erik Heerkens and Henk Simons, undoubtedly contributed to the eventual results. Especially in the beginning of the project they really helped me to master the details of my job. For those interested there are some mid-term reports available, which contain a detailed description of the work performed, and a popular version of this final report in Dutch, by those persons mentioned above. My gratitude also to Herman Aben who helped me explore the possibilities of the text editor system, to John O'Malley who gave a lot of useful suggestions concerning the English language, and to all CAB employees for numerous discussions, their interest and offered cakes. Finally, something less cheerful, the finishing of this project also marks the end of my student-days. I cannot mention all persons and clubs here who contributed to this happy period. Nevertheless I would like to thank them in general terms, my family, friends, fellow club members and house-mates of the student house 'Sam Sam Tsjat'.

Victor J.H. Konijn
Eindhoven, The Netherlands
October, 1987

TABLE OF CONTENTS

LIST OF SYMBOLS AND ABBREVIATIONS

a	Width of the DTO in number of bits.
c	Address width of the ROM in number of bits ($c \leq a$).
d	ROM and DAC data width in number of bits.
f_c	VCO center frequency in Hz.
f_{cl}	DTO clock frequency in Hz.
f_k	Modulation frequency due to the quantization $c < r$ in Hz.
f_o	Realized DTO output frequency in Hz.
K_d	Phase detector output voltage per radian phase error.
K_o	VCO frequency change in rad/s per volt input voltage.
m	Dividing factor between cycle time of the time discrete output signal and cycle time belonging to the desired output frequency
p	DTO input word, corresponding to the phase increment per clock cycle.
q	DTO overflow value.
Q1	Block in diagram which represents frequency quantization due to finite a.
Q2	Block in diagram which represents phase quantization due to $c < r$.
Q3	Block in diagram which represents amplitude quantization due to finite d.
r	Effective DTO width in number of bits; 2^r is the number of clock cycles in m periods of the time discrete output signal.
T_{cl}	Cycle time of the DTO clock signal in seconds.
T_k	Periodicity of the phase correction phenomena due to c<r in seconds.
T_o	Cycle time of the desired output signal in seconds.
v_d	VCO input control voltage in volts.
α	Loss factor of the PLL reconstruction filter.
β	Argument of Bessel functions.
ζ	Damping factor of the PLL reconstruction filter.
ϕ_e	PLL phase error in radians.
ϕ_i	PLL input phase in radians.
ϕ_o	PLL output phase in radians.
σ	Phase jitter due to two symmetrical sidebands in radians.
μ	Clipping level of a clipped triangular wave in volts.
ω_n	Natural frequency of the PLL reconstruction filter in rad/s.

BPF	Band pass filter
DAC	Digital analogue converter
DTO	Discrete time oscillator
HDMAC	High definition multiplexed analogue components
HDTV	High definition television
LF	Loop filter
LIM	Limiter
LPF	Low pass filter
LSB	Least significant bit
MAC	Multiplexed analogue components
MSB	Most significant bit
PAL	Phase alternating on line
PD	Phase detector
PLL	Phase locked loop
RAM	Random access memory
ROM	Read only memory
VCO	Voltage controlled oscillator
ZOH	Zero order hold

1. INTRODUCTION

The present developments in TV receivers are tending towards the digitizing of basic functions. The main goal of these developments is to realize picture enhancements and some auxiliary features (picture in picture, still picture, etc.). Recently, a family of new transmission standards has been accepted in Europe, mainly developed for satellite communication. These standards (known as Multiplexed Analogue Components: MAC) already result in some minor picture quality improvements. Furthermore, this transmission system offers the feasibility of several digital sound and data channels. The next goal is to develop High-Definition TV-transmission based on these MAC systems (HDMAC) with double the resolution (spatial and temporal) compared to the present PAL system [1]. One of the general problems that arise in digital TV systems is generation of several clock signals, locked to the (varying) so called line frequency of the horizontal picture scanning. Line-locked clock signals are preferable for memory applications [2].

A TV picture is displayed line-sequential. First the even lines are written (even field) then the odd lines (odd field). In this way 25 pictures (50 fields) a second are shown. In the transmission signal the start of a new video line is indicated with a horizontal synchronization pulse. With use of this pulse, clock oscillator signals can be synchronized.

For this purpose in the PAL system a phase lock loop (PLL) is used as shown in figure 1.1 [3].

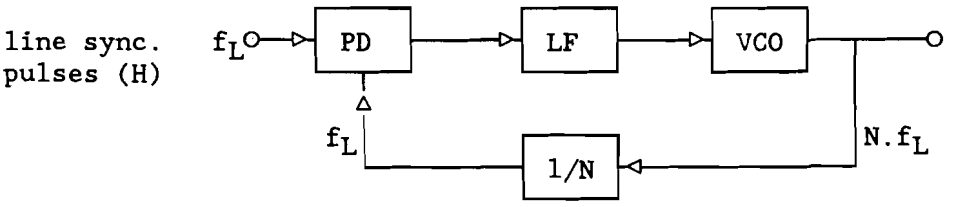


Fig.1.1 Clock signal generation in the PAL system.

In this circuit N.f_L is the least common multiple of the required clock frequencies. In certain TV systems clock signals of 20.25 MHz, 27 MHz and 40.5 MHz are required. Therefore a voltage controlled oscillator (VCO) with a nominal frequency of 81 MHz then has to be used. In practice it turns out that this frequency is too high and its stability insufficient

To avoid this problem an alternative circuit is being developed, which is based on a digital oscillator instead of the VCO. This oscillator is called a Discrete Time Oscillator (DTO). A DTO synthesizes a frequency out of a clock frequency, controlled by a digital input. Because a crystal clock can be used it is possible to synthesize frequencies with crystal stability. The DTO is used in the circuit shown in figure 1.2.

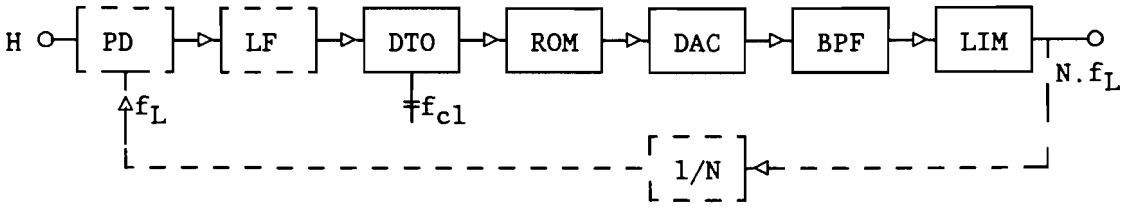


Fig.1.2 Line-locked clock generation based on a DTO.

The VCO in figure 1.1 is replaced by those blocks in figure 1.2 drawn with full lines. The DTO input is a multibit word. The DTO calculates the momentary phase of the desired frequency. This phase is transformed by a read only memory (ROM) to a particular wave shape. After digital to analogue conversion and zero order hold reconstruction data samples are filtered by an analogue band pass reconstruction filter to yield a sine wave of the desired frequency. Finally this sine wave is transformed by a high gain amplitude limiter to a square wave.

The control circuit given in figure 1.2 has been the subject of this M.Sc. research project. Due to time limitations, however, research has been restricted to the blocks drawn with full lines. First some DTO principles and the relation between time and frequency domain analyses are discussed. Then a theoretical model for practical DTO circuits has been developed. This model incorporates the DTO in combination with read only memory, digital to analogue converter and reconstruction filter. The influence of parameter values on system performance is investigated by theoretical analysis. Furthermore a computer simulation package of this circuit part has been developed to check theory and to achieve exact results in case the theory leads to complex expressions. Experiments are performed to verify both theory and simulation results.

Two types of references have been included: those which are actually referred to in the text are mentioned under the heading References. All others which contributed to the analyses described in this report and can provide for a general background are mentioned under the heading Bibliography. Because the research project has been carried out at an industrial company, the structure of this report has been organized as to facilitate reading by people who are not too interested in all theoretical aspects of this project but rather like to recognize the practical implementations. For this reason detailed analyses are printed in a smaller letter size.

2. DTO PRINCIPLES

2.1 CONSTRUCTION OF THE DTO AND ITS OUTPUT SIGNAL

A Discrete Time Oscillator is based on a multibit adder, the output of which is applied to one out of two inputs via D-flip-flops. These flip-flops are clocked with a clock frequency f_{cl}. At each clock cycle the inputs of the flip-flops are transferred to the outputs. At the second input of the adder a multibit word p is applied. This way a counter function is realized. The width of the adder in number of bits is (a), see figure 2.1.

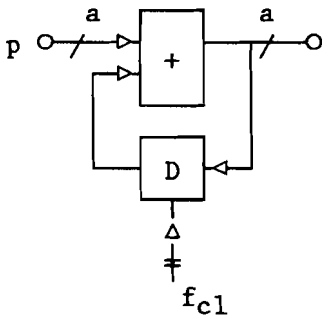


Fig.2.1 Construction of the Discrete Time Oscillator (DTO).

At each clock cycle, the contents of the accumulator is incremented by p until overflow occurs at the value q (= 2^a in figure 2.1). After overflow the next DTO output value will be the previous value plus p, modulo q; this contents of the accumulator is the base for a new adding cycle. In this way a time discrete quantized sawtooth output function is obtained, with a periodicity of $1/f_o$, see figure 2.2.

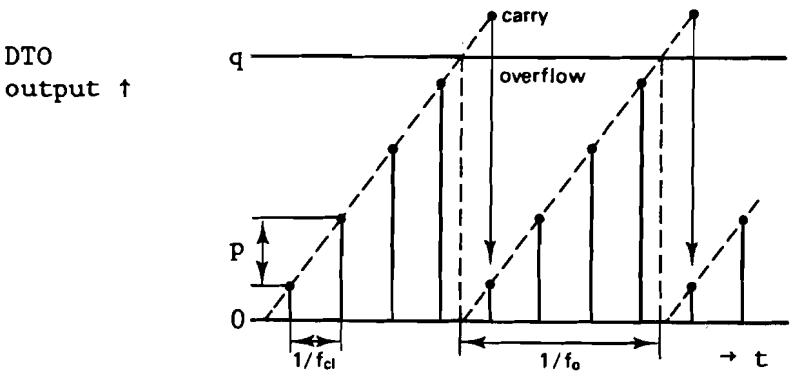


Fig.2.2 Construction of the DTO output signal.

In principle all overflow values q can be used, but the simplest implementation is based on q = 2^a. Although this implementation yields a kind of quantization (see § 2.2), this value of q will be used in further analysis. The way to implement other values of q requires some additional hardware and will be described in § 2.2.

It is quite obvious from figure 2.2 that the ratio between p and q equals the ratio between the two frequencies concerned:

$$f_o/f_{cl} = p/q = p/2^a \qquad (2.1)$$

The desired output frequency f_o is determined by the clock frequency f_{cl}, the DTO input p and the overflow value q. In fact, the DTO integrates the phase increment per clock period p/q to the instantaneous phase. In this definition of the DTO the input p can be any integer between 0 and q. For practical reasons, however, the p range will be more restricted.

2.2 FREQUENCY QUANTIZATION IN THE DTO

A DTO implementation which uses the full adder range from 0 to 2^a restricts the number of possible output frequencies f_o because, in fact, the ratio f_o/f_{cl} has to be approximated:

$$f_o/f_{cl} = (p + \partial)/2^a \qquad \text{with } |\partial| \leq \tfrac{1}{2}$$

This means that the difference between desired and realized output frequency is at most $f_{cl}/2^{a+1}$. Although one should keep this effect in mind, from now on the realized output frequency will be called f_o, according to (2.1).

This frequency quantization can be avoided by implementing other values of q. When realizing other values of q the basic idea is not to lower the overflow value of the accumulator, but to lift the starting value of a new adding cycle. If this starting value is (2^a-q) instead of zero, the range to the overflow value includes q different words, whatever integer (smaller than 2^a) q is. In this way e.g. a ratio $f_o/f_{cl} = 1/3$ can be realized exactly, see figure 2.3.

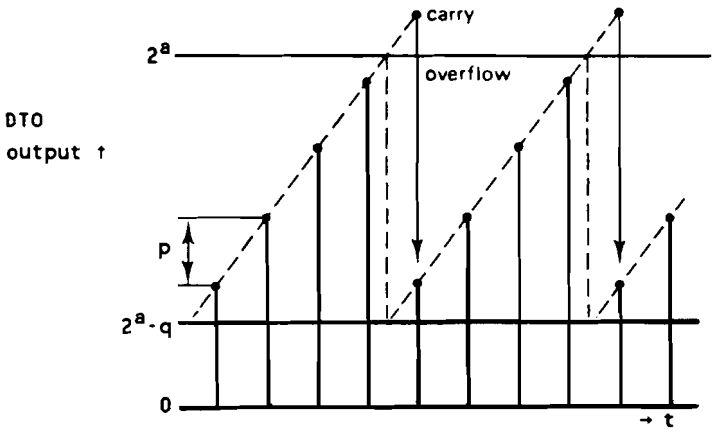


Fig.2.3 DTO output signal (when $q \neq 2^a$) versus time.

This means that when an overflow occurs the new adder input should be $p + 2^a - q$ instead of p. So an extra multiplexer is needed to select the appropriate DTO input. Selection can be controlled by an and-gate and an inverter, see figure 2.4.

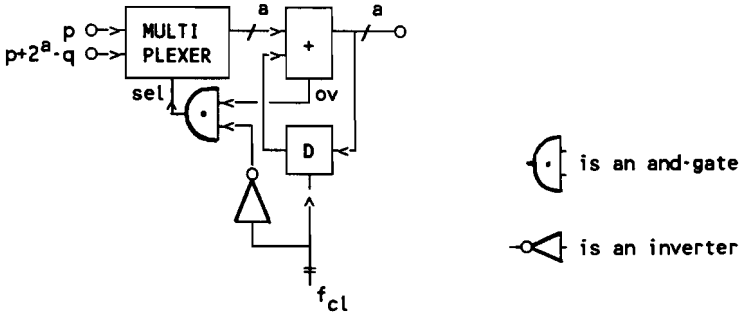


Fig.2.4 Extension of circuit to create an implementation $q \neq 2^a$.

In the first half of a clock cycle p is added to the contents of the D-flip-flops. When then an overflow occurs, in the second half of the clock cycle the output of the and-gate will be high. Then $p+2^a-q$ is added to the old contents of the D-flip-flops. At the next rising edge of the clock signal this new value will be clocked into the registers. When no overflow occurs the select input of the multiplexer is not activated and p will be added to the register contents. A practical limitation of this circuit might be the delay of the accumulator from the input to the overflow output. This delay should be less than half a clock cycle, which limits the clock frequency.

A second effect due to a full adder implementation is the minimum frequency variation when changing p by 1 LSB:

$$\Delta f_{min} = 2^{-a}.f_{cl} = f_o/p \qquad (2.2)$$

So a big relative frequency accuracy is achieved by using a large DTO width (a) and a large input word (p). The latter can be understood by realizing that the required approximation results in a rounding of the word p to an integer. This means a constant increment error at each clock cycle in the DTO. The smaller p is, the more increments are needed before overflow occurs, the bigger the accumulated error will be.

2.3 CONSTRUCTION OF THE SPECTRUM OF THE DTO OUTPUT

As mentioned before, the DTO output function can be seen as a time-discrete quantized sawtooth signal. Therefore the spectrum of the signal will contain more components than only f_o. All spectral components but the one at f_o can be regarded as noise and should contain as little spectral energy as possible.

An analogue sawtooth wave contains spectral components at all harmonics. When sampled, this spectrum will repeat itself around multiples of the clock frequency. Because the clock frequency is not necessarily one of the harmonics of f_o, this sampling operation in

general introduces new spectral components. With aid of formula (2.1) it can be shown at which frequencies that occurs.

For this purpose, it is useful to simplify (2.1) to lowest terms. Then the numerator will be odd:

$$f_o/f_{cl} = p/2^a = \tfrac{1}{2}p/2^{a-1} = \ldots = m/2^r \qquad (2.3)$$

Formula (2.3) shows that 2^r clock periods fit in m output periods. When T_o is the cycle time of a wave with frequency f_o, then the cycle time of the corresponding time discrete function will be $m.T_o$. Figure 2.5 shows an example of the circuit output, assuming that a sine wave shape has been stored in the ROM.

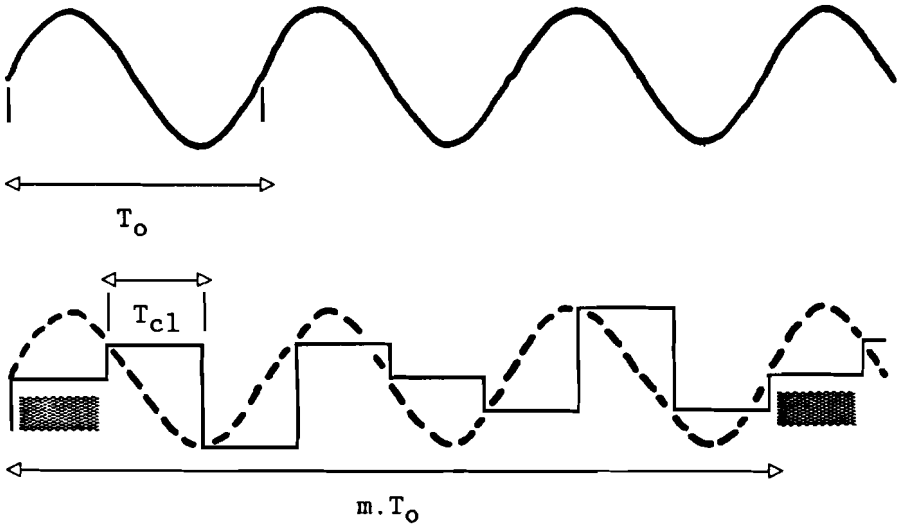


Fig.2.5 Periodicity of the time discrete signal (m = 3).

Since the periodicity of the output signal is $m.T_o$, the spectrum contains in principle frequency components at multiples of f_o/m, see figure 2.6.

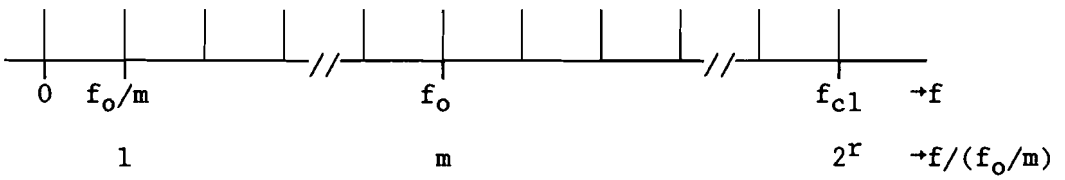


Fig.2.6 Spectrum of the DTO output signal; the signal contains no spectral components outside the marked frequencies. This figure doesn't give any information about the amplitude of the components.

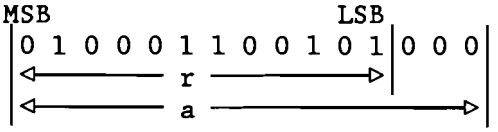
So, in general, the frequency f_o to be synthesized is represented by the m^{th} spectral line, the clock frequency f_{cl} by the $(2^r)^{th}$. The variable r is the effective DTO width; r represents the number of bits which is really needed to generate a frequency f_o out of a clock frequency f_{cl}. If the fraction f_o/f_{cl} is fixed the (a - r) DTO LSBs

will always be zero.

Example: A clock frequency f_{cl} = 24.576 MHz and a 15 bits DTO are available; a frequency f_o = 6.75 MHz is desired.

For this case, it holds for the DTO ratio: $f_o/f_{cl} = 6.75/24.576 = 1125/4096 = 1125/2^{12}$, so m=1125 and r=12. Although a=15, a 12 bits DTO will satisfy for this static conversion. Then, when a=r, it yields that m=p so an input word 1125 is needed. The spectrum contains components at multiples of $f_o/m = 6.75/1125 = 6$ kHz. The minimal frequency change is also (with 2.2) $\Delta f_{min} = 2^{-12}.f_{cl} = 6$ kHz.

When the 15 bits DTO is used, $p = (2^{15-12}).m = 9000$. Of course the spectrum is not influenced, but the minimum frequency change when changing p is now $\Delta f_{min} = 2^{-15}.f_{cl} = 750$ Hz.



This structure of the spectrum is verified by experiments. The details about the test circuit and equipment are mentioned in appendix 1. Photo 2.1 shows an oscilloscope view of the DAC output time function and the corresponding clock signal when $f_o/f_{cl} = 3/16$ so r = 4 and m = 3; c = 3 and a sine wave shape has been stored in the ROM. This photo clearly shows the periodicity of the time discrete function is $3.T_o$. Therefore spectral components are expected at multiples of $f_o/3$. In appendix 2 the corresponding measured spectrum is given. In this appendix also a table is added which contains an expectation of the spectrum based on a computer simulation. This simulation will be discussed in § 3.2 and appendix 3. The theory about the amplitudes of the sidebands will be discussed in chapter 4. Photo 2.2 is added on the next page for comparison. It shows the output function displayed on an oscilloscope when $f_o/f_{cl} = 5/16$ so then the periodicity of the time discrete signal is $5.T_o$.

For the distance between spectral lines in the spectrum it yields:

$$f_o/m = f_o/(p.2^{r-a}) = (f_o/p).2^{a-r}$$

By writing the equation this way it can easily be seen that if the entire DTO width is used, the relative spectral line distance will be f_o/p. Each decrease of the effective DTO width (r) with one bit increases this line distance with a factor two. This might influence the choice of the clock frequency.

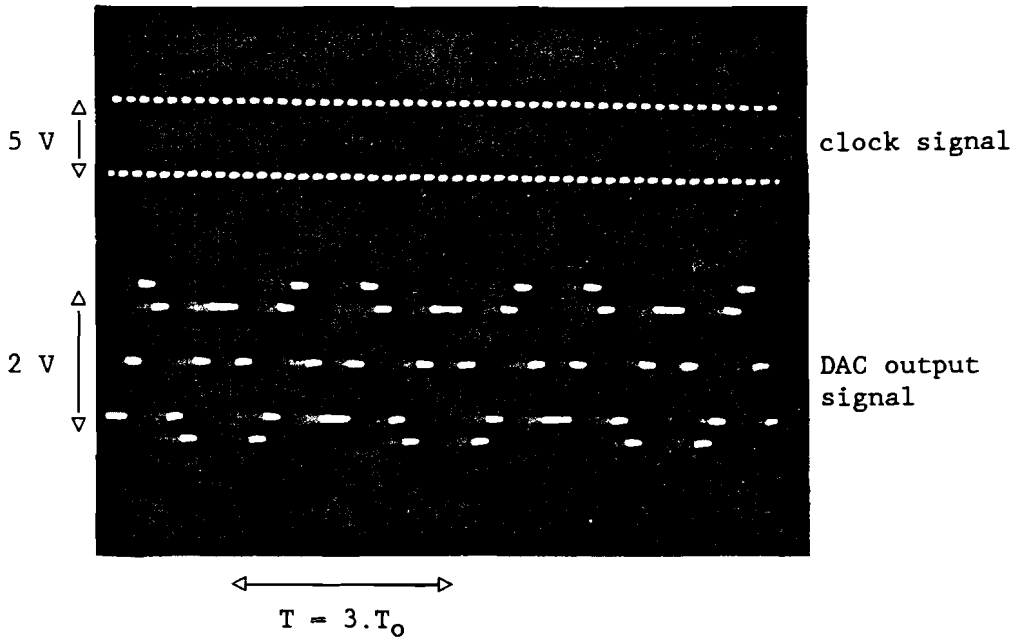


Photo 2.1 Oscilloscope picture of the DAC output time function and corresponding clock signal; $f_o/f_{cl} = 3/16$, c = 3, sine wave shape stored in the ROM. The periodicity of the DAC output signal is $3.T_o$.

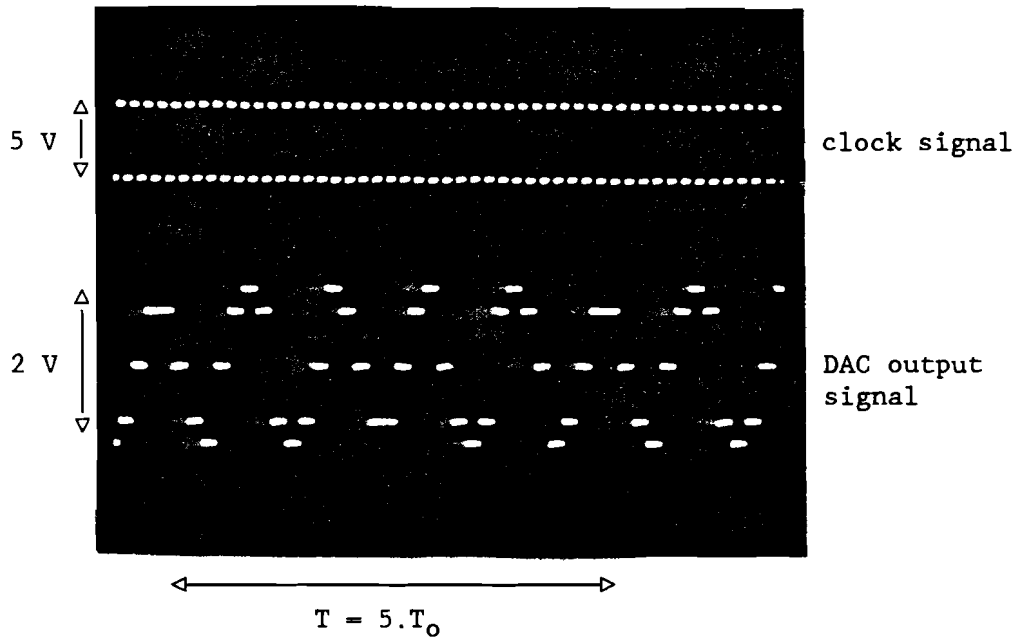


Photo 2.2 As photo 2.1; now $f_o/f_{cl} = 5/16$ so the periodicity of the DAC output signal is $5.T_o$.

3. RELATION BETWEEN TIME AND FREQUENCY DOMAIN

3.1 ANALYSIS OF PHASE JITTER DUE TO SIDEBANDS

As mentioned in chapter 1 an application of this circuit might be generating line-locked clock signals. Then the resulting time or phase jitter in the output signal is relevant for circuit quality. So the influence of various system parameters on circuit performance in the time domain is important. However, the reconstruction filter behind the DAC (see fig.1.2) is defined in the frequency domain. In order to investigate the effects of sidebands in the frequency spectrum on the resulting jitter in the output signal, the contribution of sidebands at one frequency distance is evaluated.

Consider a sine wave of frequency f_o which is disturbed by two sidebands of frequency $f_o \pm z$, see figure 3.1.

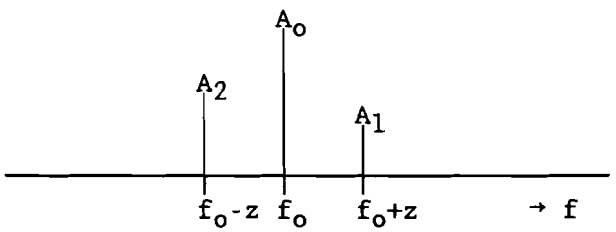


Fig.3.1 Sine wave with amplitude A_o and two equidistant sidebands at a distance of z [Hz] with amplitude A_1 and A_2.

The time function v(t) can be written, with $\omega_o = 2.\pi.f_o$, as:

$$v(t) = A_o.\sin(\omega_o t) + A_1.\sin[(\omega_o+z)t + \phi_1] + A_2.\sin[(\omega_o-z)t + \phi_2]$$

This formula can be rewritten into the form:

$$v(t) = V.\sin(\omega_o t + \sigma(t))$$

For the time dependant phase displacement $\sigma(t)$ it holds:

$$\sigma(t) = (A_1/A_o).\sin(zt+\phi_1) + (A_2/A_o).\sin(zt+\pi-\phi_2) \qquad (3.1)$$

To achieve these results some goniometric rules should be applied:

$$v(t) = A_o.\sin(\omega_o t) + A_1.\sin[(\omega_o+z)t + \phi_1] + A_2.\sin[(\omega_o-z)t + \phi_2]$$

$$= A_o.\sin(\omega_o t) + A_1.\sin(\omega_o t).\cos(zt+\phi_1) + A_1.\cos(\omega_o t).\sin(zt+\phi_1) + A_2.\sin(\omega_o t).\cos(zt-\phi_2) - A_2.\cos(\omega_o t).\sin(zt-\phi_2)$$

$$= \sin(\omega_o t).[\, A_o + A_1.\cos(zt+\phi_1) + A_2.\cos(zt-\phi_2)\,] + \cos(\omega_o t).[\, A_1.\sin(zt+\phi_1) - A_2.\sin(zt-\phi_2)\,]$$

$$= B(t).\sin(\omega_o t) + C(t).\cos(\omega_o t)$$

$$= B(t).[\, \sin(\omega_o t) + (C(t)/B(t)).\cos(\omega_o t)\,] \qquad (B(t) \neq 0)$$

$v(t) = V(t).\sin(\omega_o t + \sigma)$ with $\tan(\sigma) = C(t)/B(t)$ and $V(t) = B(t)/\cos(\sigma)$

$$\tan(\sigma) = \frac{A_1.\sin(zt+\phi_1) - A_2.\sin(zt-\phi_2)}{A_o + A_1.\cos(zt+\phi_1) + A_2.\cos(zt-\phi_2)}$$

When $A_o >> A_1, A_2$ and σ small, then $\tan(\sigma) \approx \sigma$ and then it holds:

$$\sigma(t) \approx (A_1/A_o).\sin(zt+\phi_1) - (A_2/A_o).\sin(zt-\phi_2)$$

$$= (A_1/A_o).\sin(zt+\phi_1) + (A_2/A_o).\sin(zt+\pi-\phi_2)$$

Formula (3.1) can be rewritten into the form:

$$\sigma(t) = R.\sin(zt + \phi)$$

The amplitude R and phase ϕ of the jitter σ can be constructed out of the two vectors in (3.1), see figure 3.2.

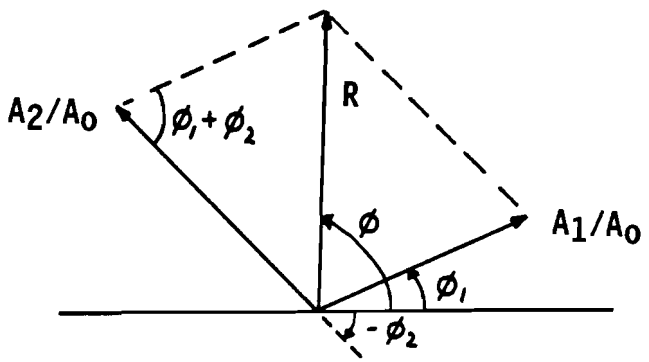


Fig.3.2 Construction of the phase jitter vector.

So the two equidistant sidebands result in a sinusoidal phase jitter σ which has an amplitude R, and a frequency which equals the frequency difference between the sidebands and the base wave. Applying the cosine rule to figure 3.2 yields:

$$R^2 = (A_1/A_o)^2 + (A_2/A_o)^2 - 2.(A_1.A_2/A_o^2).\cos(\phi_1+\phi_2) \qquad (3.2)$$

So both the amplitudes of the sidebands and their relative phases influence the amplitude of the jitter. Two special cases can be distinguished:

$$\text{If } \phi_1+\phi_2 = 0 + 2.k.\pi \text{ then } R = |A_1-A_2|/A_o$$
$$\text{If } \phi_1+\phi_2 = \pi + 2.k.\pi \text{ then } R = (A_1+A_2)/A_o \qquad (3.3)$$

The first case of (3.3) covers amplitude modulation (AM). Then $A_1 = A_2$ and $\phi_1 = -\phi_2$ so $R = 0$. It is known indeed that AM does not cause time jitter in a signal. In table 3.1 the amplitude of phase jitter due to one sideband is given. The relation is also shown in figure 3.3. It should be noticed that the amplitude of the total jitter due to a sum of sidebands is less then the sum of the single jitter amplitudes due

to two effects. First the relative phases of two equidistant sidebands can partly compensate the mutual effects, see (3.2) and (3.3). Second the amplitude of the sum of some sinusoidal waves with different frequencies is mostly less then the sum of all individual amplitudes.

An upper limit for the eventual jitter amplitude of n pairs of equidistant sidebands at a distance of multiples of f_o/m from the center frequency f_o, which each result in a jitter of amplitude R_i and phase ϕ_i according to R and ϕ in figure 3.2, is given by:

$$A = \max\{ \sum_{i=1}^{n} R_i.\sin(i.f_o/m + \phi_i) \} \qquad (3.4)$$

Table 3.1 Amplitude of phase jitter in degrees due to one sideband in the spectrum with amplitude A_1; the fundamental wave is assumed to have amplitude A_o.

A_1/A_o (dB)	max. jitter(deg.)
-10	18.12
-20	5.730
-30	1.812
-40	0.5730
-50	0.1812
-60	0.05730
-70	0.01812

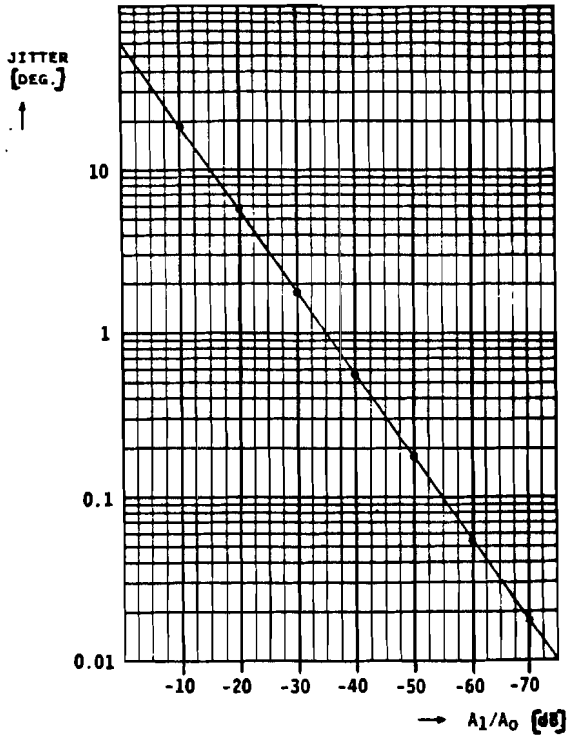


Fig.3.3 Amplitude of the sinusoidal phase jitter in degrees due to one sideband in the spectrum versus relative amplitude of the sideband.

3.2 COMPUTER SIMULATION

It follows from the analysis that both the amplitudes and the relative phases of the sidebands are relevant for the resulting jitter. The complexity of the structure of the spectrum makes it arduous to find an expression for the exact amplitudes and phases of the sidebands. However, a computer simulation of the circuit can assist the designer of a practical DTO circuit.

Using Fourier series it will be possible to achieve frequency components of a signal if the time function is known. The spectrum of a periodic function contains components at multiples of the frequency of the signal. The amplitude of the n^{th} harmonic component of a function with period T is given by [4]:

$$c_n = (1/T) \int_T f(t).e^{-2\pi jnt/T}\, dt \qquad (3.5)$$

Applying this formula for the present case yields a formula which can be used in a computer program.

To derive this formula remember that the period of the DAC output signal is $m.T_o$. One period consists of 2^r clock cycles. During one clock period the output function does not change due to the ZOH device in the DAC (fig.1.2). Therefore it holds for the c_n corresponding to the spectral line at frequency $n.f_o/m$:

$$c_n = (1/mT_o).\sum_{s=0}^{2^r-1} f(sT_{cl}).\int_{s.T_{cl}}^{(s+1).T_{cl}} e^{-2\pi jnf_o t/m}\, dt$$

Developing the integral yields:

$$c_n = (1/\pi n).\sin(\pi n/2^r).\sum_{s=0}^{2^r-1} f(s.T_{cl}).e^{-2\pi jn(s+\frac{1}{2})/2^r}$$

$$c_n = (1/2^r).\mathrm{sinc}(n/2^r).\sum_{s=0}^{2^r-1} f(s.T_{cl}).e^{-2\pi jn(s+\frac{1}{2})/2^r}$$

So for calculating the value of c_n only the clock cycle time T_{cl}, the values $f(s.T_{cl})$ and the effective DTO width r are needed. All values $f(s.T_{cl})$ are determined by the wave shape in the ROM. In general the value of c_n will be complex. The norm $|c_n|$ represents the amplitude of the frequency component, the argument $\arg(c_n)$ its phase. To obtain the relative phase with regard to the component at the desired frequency f_o, the phase of this component should be subtracted.

Some programs are written in Pascal on a VAX computer. The flow charts of these programs are given in appendix 3. The input contains all relevant system parameters, the output includes relative amplitudes and relative phases of all frequency components in a user-defined frequency range. Appendix 3 also contains an example of program output.

It follows from many simulations performed that it holds in a lot of cases for equidistant sidebands that either $\phi_1+\phi_2 = 0$ or $\phi_1+\phi_2 = \pi$. These are just the two extremes with relation to phase jitter as mentioned in § 3.1. It depends on the circuit parameters c, m and r which situation occurs. No general rules have been found; for that reason simulation is needed to characterize a particular circuit implementation.

This simulation can be used to diminish the effects of certain sidebands by altering some parameters. In this way a tool is created that will be useful for a practical designer.

4. PHASE QUANTIZATION IN DTO - ROM CONNECTION

4.1 MODEL

According to the block diagram of figure 1.2 the output of the DTO is connected to a Read Only Memory (ROM). This memory transforms the phase information out of the DTO into a particular wave shape (sine, triangular). This is done by storing amplitude values of discrete phase moments in addresses. The DTO output word assigns an address, the data stored is the circuit output. Main goal of this transformation is to clean up the spectrum of the DTO output signal by reducing or even eliminating harmonics.

When all DTO output lines are connected to the ROM, 2^a adresses can be assigned. Because this is not necessarily true, in the general model it will be assumed that c MSBs out of the DTO will be used to assign the ROM addresses. As long as $c \geq r$, this quantization is not relevant (§ 2.3). When $c < r$, however, a further phase quantization occurs. Whatever c is, all (a) output bits of the DTO will always be applied to the DTO input.

The width of each data word in the ROM is assumed to be d bits. This means an amplitude quantization with 2^d amplitude levels. These d bits are connected to a digital analogue converter (DAC) which transforms the digital input word into analogue data samples. In most practical DACs a zero order hold (ZOH) device is included which makes the actual output to be a stair case function. Figure 4.1 shows the model of the circuit. The DTO frequency quantization (see § 2.2) is also modelled.

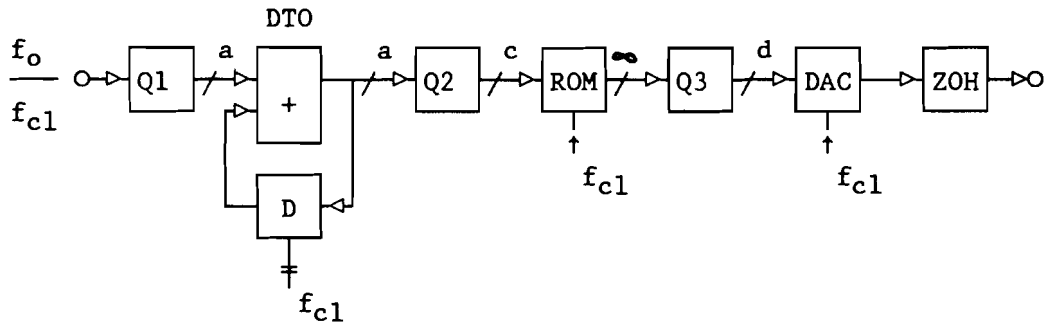


Fig.4.1 Model of the DTO - ROM - DAC circuit.

4.2 TIME DOMAIN ANALYSIS

To facilitate a good understanding of the effects of the DTO-ROM quantization (Q2 in fig.4.1) in time domain first an example will be given.

Example: Consider a circuit implemented with a=4, c=3, p=3. In table 4.1 the successive DTO output words and corresponding DTO- and ROM phases are given:

Table 4.1 Successive DTO output words and corresponding phases.

time	word (p=3)		DTO-phase (a=4)	ROM-phase (c=3)
$0.T_{c1}$	0.p	000 | 0	0	0
$1.T_{c1}$	1.p	001 | 1	3 x $\pi/8$	1 x $\pi/4$
$2.T_{c1}$	2.p	011 | 0	6 x $\pi/8$	3 x $\pi/4$
$3.T_{c1}$	3.p	100 | 1	9 x $\pi/8$	4 x $\pi/4$
$4.T_{c1}$	4.p	110 | 0	12 x $\pi/8$	6 x $\pi/4$
$5.T_{c1}$	5.p	111 | 1	15 x $\pi/8$	7 x $\pi/4$
$6.T_{c1}$	6.p	001 | 0	2 x $\pi/8$	1 x $\pi/4$
$7.T_{c1}$	7.p	010 | 1	5 x $\pi/8$	2 x $\pi/4$

To illustrate the Q2 quantization effect the phase versus time plot is shown in figure 4.2 for three cases: a=c=4 (no quantization), a=4, c=3 (practical situation) and a=c=3 (3 bits DTO, no phase correction of the LSBs).

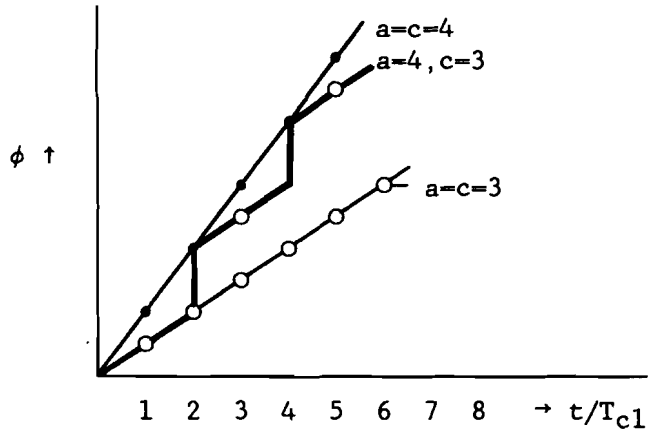


Fig.4.2 Phase at the ROM input versus time.

As can be seen in this figure, in the practical situation, the phase is corrected every two clock cycles so that the ideal curve (a=c=4) can be traced. This correction is caused by applying the (a-c) DTO LSBs to DTO input. That is why the implementation a=c=3 shows the uncorrected curve.

The phase increment of the practical situation is then as given in figure 4.3.

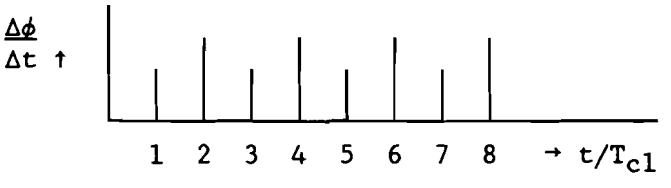


Fig.4.3 Phase increment per clock period when the DTO output is quantized before ROM input.

So the phase increment can have two values. One of them is too low and occurs when there is no carry of the DTO LSBs. The other value is too high and occurs when there is a carry of the LSBs to the c MSBs. This too high a value can correct the momentary phase. The desired phase increment per clock period is in between those two values, here 1.5 ROM address. This phase correction phenomena is periodic with two clock cycles. After each period the momentary phase is correct so within each period the average phase increment is correct.
• End of example.

The effect presented in the previous example occurs in general. If c<r, actually the DTO output word is divided by 2^{r-c} and truncated to an integer. This truncation makes the phase increment sent to the ROM too low because the (r-c) LSBs are assumed to be zero but they are not, by definition of r. For this reason a phase error is created compared to the desired signal.

Because the (r-c) LSBs are applied to the DTO input the information incorporated in those bits is not lost. Once in a while there is a carry of these LSBs to the c MSBs. The address increment is then one address more. In this way the phase lag is decreased. Therefore there is a maximum value for this phase lag. This occurs if all (r-c) LSBs are 1:

$$\Delta\phi_{max} = (2^{-c}-2^{-r}).2.\pi$$

When this maximum is achieved a carry at the next clock cycle will prevent the error to increase; it will decrease instead. When analyzing the DTO output word it can be noticed when the momentary phase will be correct again. First an example will be given as an illustration.

Example: a=8, r=7, c=4, p=38 so m=19.
The successive DTO output words are then (a space is used to mark the c MSBs that are fed to the ROM):

```
0000 0000
0010 0110
0100 1100
0111 0010 : first carry
1001 1000
1011 1110 : maximum phase error (2^-3-2^-6).2.π = (7/64).2.π
1110 0100
0000 1010
0011 0000 : zero phase error
0101 0110
  └──┘ └──┘──▷ phase error
    └──────▷ to ROM
```

It holds in general that after the first clock cycle the r^{th} bit of the DTO output word represents the last 1, by definition of r. Each clock cycle this output word (which equals p) is added to get the new output word. After 2^{r-c} clock cycles this r^{th} bit is shifted (r-c)

bits to the left and is therefore shifted into the c MSBs. Then no phase error results. Therefore the phase correction phenomena is periodic with periodicity T_k:

$$T_k = 2^{r-c}.T_{cl} \qquad (4.1)$$

A clear proof of this periodicity is given by the following. The instantaneous phase error after one clock cycle is given by the last (a-c) bits of the word p. This is the integer (p mod 2^{a-c}). The periodicity given by (4.1) is true when it holds that after $2^{r-c}+1$ clock cycles the same phase errors results:

$$[\ (\ 2^{r-c} + 1\ ).p\ ] \bmod 2^{a-c} = p \bmod 2^{a-c}$$

Use of the general rule (a + b) mod c = [a mod c + b mod c] mod c yields for the left member of the equation exactly the right member, because by definition of r:

$$(\ 2^{r-c}.p\ ) \bmod 2^{a-c} = 0$$

This last equation shows directly that after each period T_k the total phase error left equals zero.

It follows already from the example that the momentary phase is not necessarily correct after each carry correction. Several corrections in one period T_k might be needed to adjust the phase totally.

Because after such a period T_k there is no phase error left, the average phase increment per clock cycle within the period is correct. It is just this <u>average</u> phase increment that corresponds to the desired frequency f_o.

4.3 FREQUENCY DOMAIN ANALYSIS I

Because only the average phase increment per clock cycle at the ROM input is correct, and this phase increment is not constant, the DTO-ROM quantization causes some phase jitter in the signal. This jitter corresponds to some sidebands in the frequency spectrum of the output signal. The exact relation between amplitude of sidebands and resulting phasejitter has been explained in chapter 3. It is clear though that a reconstruction filter should remove as many sidebands as possible and that therefore it should be a band pass filter with center frequency f_o. The filter then averages phase increments in the output signal.

To predict the amplitude and frequency of the sidebands first consider a situation when m=1. This means that 2^r clock periods fit in 1 output period T_o. When c<r each address in the ROM is read several times. Actually all addresses are then read 2^{r-c} times, because after 2^{r-c} clock periods the 1 shifts into the c MSBs and results in an address increment; see the example below.

Example: m=1, c=r-2. Successive DTO output words:

```
00 00
00 01
00 10
00 11      When r-c = 2 each address is read 4 times.
01 00
01 01
  :
```

Because the actual circuit contains a ZOH device at the output, see figure 4.1, the clock frequency cannot be noticed in the output signal any more, see figure 4.4.

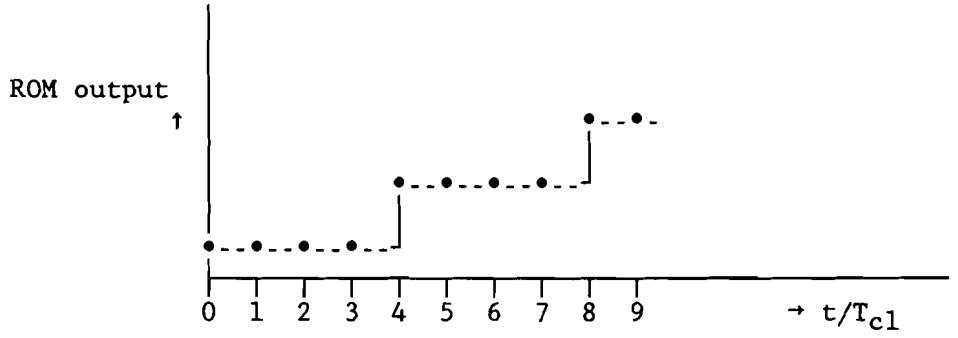


Fig.4.4 Effect of an implementation c=r-2 when m=1.

In general when every address is read 2^{r-c} times the fictitious clock frequency is $f_{cl}/2^{r-c}$. This is the frequency belonging to the periodicity of the phase correction phenomena, cp. (4.1).

$$f_k = 1/T_k = 2^{c-r}.f_{cl} = 2^c.[\ f_{cl}/2^r\ ] = 2^c.[\ f_o/m\ ] \qquad (4.2)$$

Therefore, due to the DTO-ROM quantization the spectrum of the analogue waveshape stored in the ROM will repeat itself around multiples of f_k. So actually the output signal is modulated with a frequency f_k, see figure 4.5.

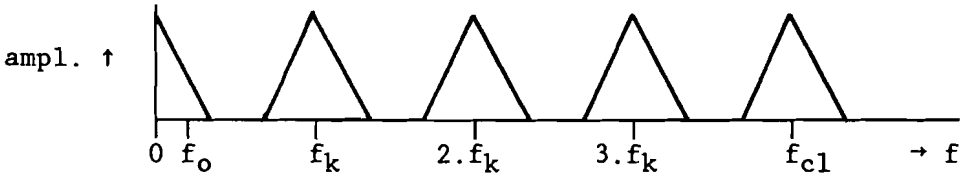


Fig.4.5 Spectrum of the DAC output when m=1, c<r. The spectrum of the analogue wave shape repeats itself at multiples of f_k.

In practice, due to the ZOH device behind the DAC, the entire spectrum should be multiplied by a sinc (= $\sin[\pi x]/\pi x$) function with zeros at multiples of f_k. This causes subsequent modulations to be damped according to a 1/n curve.

This follows from the ZOH transfer function:

$$H(j\omega) = (1-e^{-j\omega T_{cl}})/j\omega$$
$$= (e^{-j\omega \frac{1}{2} T_{cl}}/\frac{1}{2}\omega).(e^{j\omega \frac{1}{2} T_{cl}} - e^{-j\omega \frac{1}{2} T_{cl}})/2j$$
$$= (e^{-j\omega \frac{1}{2} T_{cl}}/\frac{1}{2}\omega).\sin(\pi.f.T_{cl})$$
$$= T_{cl} \cdot e^{-j\omega \frac{1}{2} T_{cl}} \cdot \mathrm{sinc}(f/f_{cl})$$

So this device takes care of a phase shift of half a clock period for all frequencies (constant group delay) and an amplitude damping according to a sinc function of amplitude T_{cl} and with zeros at multiples of f_{cl}. Because now the virtual clock frequency is f_k, T_{cl} and f_{cl} should be replaced by T_k and f_k respectively. The damping of subsequent modulations follows from the fact that

$$\frac{\mathrm{sinc}(x+1)}{\mathrm{sinc}(x)} = \frac{\sin[\pi(x+1)] \,/\, \pi(x+1)}{\sin[\pi x] \,/\, \pi x} = \frac{-x}{x+1}$$

So spectral lines at the same distance of subsequent zeros of the sinc functions are damped according to a 1/n curve. The corresponding phases change with π radians.

In the general case $m \neq 1$ the phase increment per clock period out of the DTO follows a more random function between the two possible values (carry or no carry). This results in not only one virtual clock frequency, but a much more complex sampling signal. In terms of the spectrum this turns out to result in an extra damping of the modulated spectra; the magnitude of the damping of the modulation around $n.f_k$ is determined by a Bessel function $J_n(\beta)$.

Actually the damping factor equals an infinite product series of Bessel functions. This can be understood by analyzing the DAC output function, assuming that no harmonics of f_o appear in the spectrum. Harmonics of f_o only mean a modification of the spectrum. This modification will be dealt with later. For this output function u(t) can be written:

$$u(t) = A.\sin[\, 2.\pi.\int_0^t f(\tau)\, d\tau \,] \cdot \sum_k \delta(kT_{cl}) \qquad (4.3)$$

In this formula, f(t) equals the momentary frequency. As can be seen from figure 4.3 this value can change from clock period to clock period due to the DTO - ROM quantization; f(t) can have one out of two values:

$f(t) = f_o + \Delta f(t)$ with $\Delta f(t) = -\Delta f_1$ if there is no phase correction due to a carry.
$= +\Delta f_2$ if there is a phase correction due to a carry.

Because $\Delta f(t)$ has periodicity T_k (see § 4.2) it can be written as a Fourier series:

$$\Delta f(t) = \sum_w c_w . e^{2\pi j w f_k t} \quad \text{with } f_k = 1/T_k$$

$\Delta f(t)$ Has only two values which can occur only in particular order. This limitation causes the function to be even. Then it can be written as a cosine series:

$$\Delta f(t) = \sum_w a_w.\cos(2\pi w f_k t) \qquad (4.4)$$

Applying (4.4) in (4.3) yields for u(t):

$$u(t) = A.\sin\left[\, 2.\pi.\left\{ f_o.t + \int_0^t \sum_w a_w.\cos(2\pi w f_k \tau)\, d\tau \right\} \right]$$

$$= A.\sin(2\pi f_o t).\cos\left[\sum_w (a_w/w f_k).\sin(2\pi w f_k t) \right] +$$

$$A.\cos(2\pi f_o t).\sin\left[\sum_w (a_w/w f_k).\sin(2\pi w f_k t) \right]$$

It holds [4]:

$$\cos[\beta.\sin(2\pi f_k t)] = J_o(\beta) + \sum_{n\ \text{even}} \{\, 2.J_n(\beta).\cos(n f_k t) \,\}$$

$$\sin[\beta.\sin(2\pi f_k t)] = \sum_{n\ \text{odd}} \{\, 2.J_n(\beta).\sin(n f_k t) \,\} \qquad (4.5)$$

The functions $J_n(\beta)$ are Bessel functions. So u(t) can be written indeed as a series of these functions. $J_o(0)=1$; the norms of all other Bessel functions are less than 1. Therefore the expression for u(t) is approximated by using only the term with w=1. All other terms contain quadratic Bessel functions or even higher orders; these are neglected. This yields:

$$u(t) = A.\sin(2\pi f_o t).\cos\left[(a_1/f_k).\sin(2\pi f_k t) \right] +$$
$$A.\cos(2\pi f_o t).\sin\left[(a_1/f_k).\sin(2\pi f_k t) \right]$$

Then applying (4.5) results in:

$$u(t) = A.\sin(2\pi f_o t).\left[J_o(a_1/f_k) + \sum_{n\ \text{even}} \{\, 2.J_n(a_1/f_k).\cos(2\pi n f_k t) \,\} \right. +$$

$$\left. A.\cos(2\pi f_o t).\left[\sum_{n\ \text{odd}} \{\, 2.J_n(a_1/f_k).\sin(2\pi n f_k t) \,\} \right.\right.$$

$$u(t) = A.J_o(a_1/f_k).\sin(2\pi f_o t) +$$

$$A \sum_{n\ \text{even}} J_n(a_1/f_k).\{\, \sin[2\pi(f_o - n f_k)t] + \sin[2\pi(f_o + n f_k)t] \,\} +$$

$$A \sum_{n\ \text{odd}} J_n(a_1/f_k).\{\, \sin[2\pi(n f_k - f_o)t] + \sin[2\pi(n f_k + f_o)t] \,\} +$$

Taking advantage of the property $J_{-n}(\beta) = (-1)^n.J_n(\beta)$ [4] this equation can be rewritten into the more compact form:

$$u(t) = A \sum_n J_n(\beta).\sin[2\pi(f_o + n f_k)t] \qquad \beta = a_1/f_k \qquad (4.6)$$

This formula shows that the amplitude of the sine wave at $f_o + n.f_k$ is determined by the Bessel function $J_n(\beta)$. Because actually u(t) not only contains f_o but also its harmonics it holds in general that the base spectrum of the wave shape stored in the ROM will be modulated around multiples of f_k. The entire modulation around

$n.f_k$ is damped. The amplitude of the damping is given by the Bessel function $J_n(\beta)$. For the argument β it holds (see 4.6) that it is the main component in the spectrum of $\Delta f(t)$ (see 4.4) multiplied by the cycle time of $\Delta f(t)$. To get a clean spectrum $J_o(\beta)$ should be large and all other Bessel functions, with n>0, should be small as can be seen from (4.6). In figure 4.6 some Bessel function curves are shown.

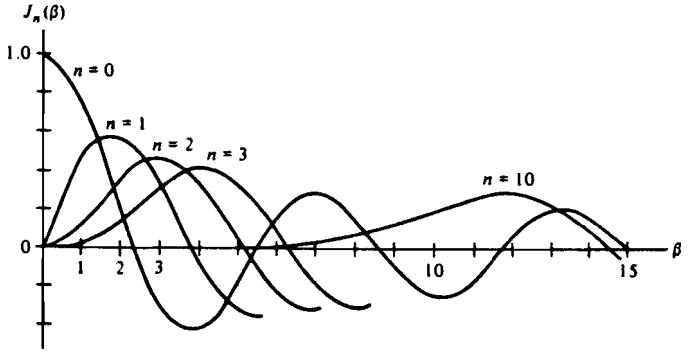


Fig.4.6 Bessel functions of fixed order plotted versus the argument β.

Figure 4.6 shows that by selecting an appropriate value for β the modulation amplitudes can really be influenced. The choice β=2.4 would e.g. not be the best choice! It is more difficult however to find out which β would be preferable. This point is better illustrated in figure 4.7 which gives the amplitudes as a function of the ratio n/β.

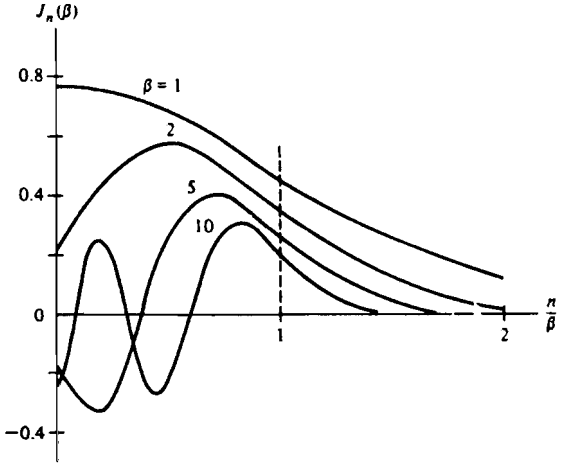
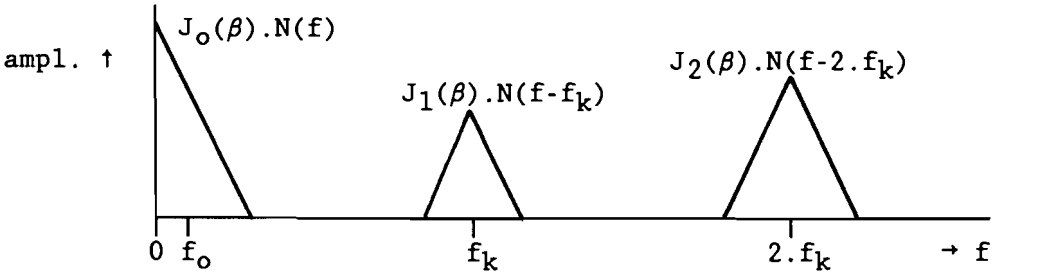


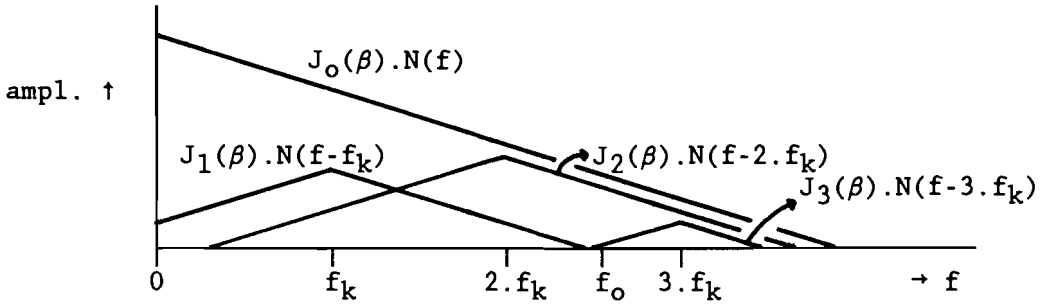
Fig.4.7 Bessel functions of fixed argument β plotted versus n/β with n the order of the function.

This figure shows that if $n>\beta$ Bessel functions are strictly decreasing. Therefore a small β is preferable which can be achieved by realizing a small value of T_k. This result is quite clear because with (4.1) this means that c should be as close to r as possible.

It follows from (4.2) that in dependance of c and m the modulating frequency f_k can be either larger or smaller than f_o. In figure 4.8 the effect of this DTO-ROM quantization is shown for both cases.



a: $f_k > f_o$



b: $f_k < f_o$

Fig.4.8 Due to the implementation c<r the spectrum N(f) of the wave shape stored in the ROM is modulated around multiples of f_k. The modulation around $n.f_k$ is damped with a factor $J_n(\beta)$.

The value r-c determines the modulating frequency f_k. A DTO implementation with a small r enables a small ROM (a small c) without introducing a lot of sidebands in the spectrum. When r is fixed the number of modulations will be halved if c is increased with one bit; see (4.2) and figure 4.9.

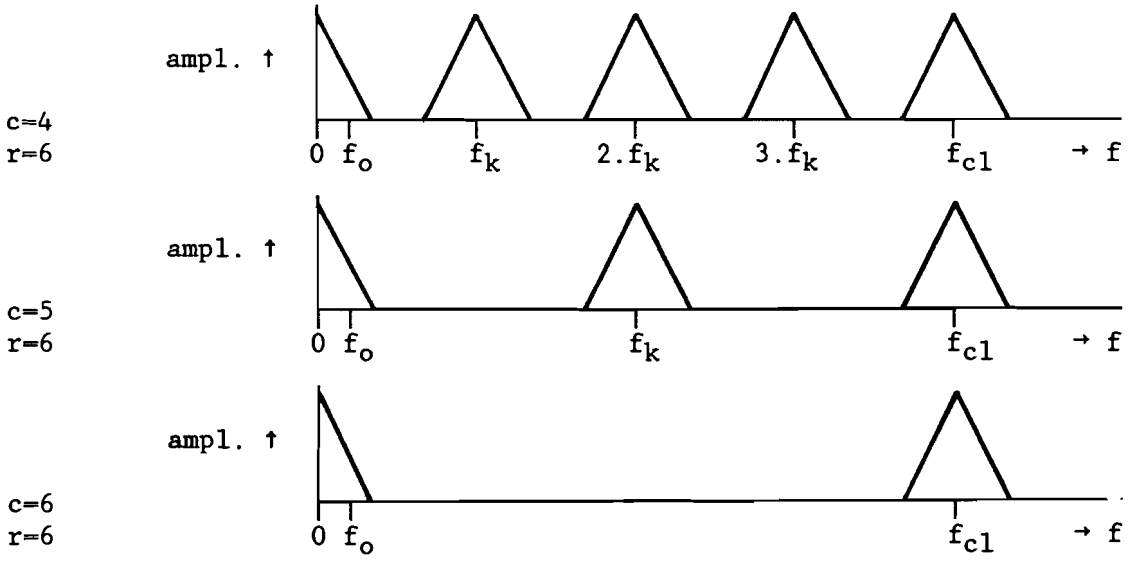


Fig.4.9 All odd modulations vanish if c is increased with one bit.

Because it is possible to determine the origin of a particular sideband in the spectrum it is then possible to calculate the value of c necessary to eliminate the modulation which causes that particular sideband. However, because a sideband can be caused by different modulations the sideband amplitude will be reduced but probably not eliminated by increasing c. The shape of the spectrum contour at a particular modulation is determined by the wave shape in the ROM and the amplitude modulation (see chapter 5). This is a second tool to reduce the amplitudes of particular sidebands.

To distinguish the origin of a particular sideband at frequency f one should just solve the equation

$$f = n.f_k \pm j.f_o = n.f_{cl}/2^{r-c} \pm j.f_o \qquad (4.7)$$

Several combinations of n and j might form a solution of (4.7); that is why it is difficult to really eliminate a sideband. However, the n with lowest j to match is always the most important modulation because, besides the Bessel function damping, all sidebands with j>1 also have a base spectrum damping due to the wave shape (ROM) and amplitude modulation (DAC). When n is found in this equation, the number of times n has to be divided by two to lose its integer state equals the number of bits c has to increase to eliminate the relevant modulation without changing other relevant parameters (m, r, f_{cl}).

By using the simulation program described in § 3.2 and appendix 3 it is possible to calculate the sidebands amplitudes as a function of m, r, c and d. In table 4.1 results are shown for a particular circuit implementation for several values of c. In this table it is clearly shown that an increase of c by one bit strongly decreases sidebands caused by modulations of odd order (n odd).

If c increases with one bit the order of all modulations is halved. As can be seen clearly from table 4.1, the amplitude of a sideband decreases with about 20 dB when halving n makes n to lose its integer state. If c is increased more the amplitude increases again a few dB. This is because of the lower order the modulation gets at each increase of c corresponding to an other Bessel function . When c increases, the amplitudes due to modulations which are not eliminated increase a few decibels. This is because of energy sharing among all sidebands left.

In the example shown in table 4.1 the two sidebands next to f_o cannot be reduced by increasing c because then n is a power of two. This means that those sidebands are caused by the modulation around f_{cl}.

Table 4.1 Sidebands amplitudes for several values of c when f_o/f_{cl} = 85/512, so m=85, r=9, d=c-1, triangular wave in the ROM. Parameters n and j are according to (4.7) in case c=4.

f / f_o/m	n	j	Sideband amplitude (dB) c=4	c=5	c=6	c=7	c=8	c=9
73	-22	+5	-49	-46	-69	-66	-63	-63
75	10	-1	-30	-30	-82	-61	-61	-61
77	42	-7	-56	-46	-63	-57	-56	-56
79	-11	+3	-36	-56	-54	-52	-51	-51
81	21	-3	-33	-53	-46	-45	-45	-45
83	-32	+7	-45	-35	-34	-34	-34	-34
85	0	+1	0	0	0	0	0	0
87	32	-5	-32	-29	-28	-28	-28	-28
89	-21	+5	-38	-46	-43	-42	-42	-42
91	11	-1	-25	-65	-52	-50	-49	-49
93	-42	+9	-55	-49	-60	-56	-55	-55
95	-10	+3	-40	-39	-72	-60	-59	-59
97	22	-3	-41	-41	-74	-64	-62	-62

Only the odd frequency components are shown in table 4.1. This is because theoretically the even components are zero if the spectrum of the wave shape in the ROM does not contain even harmonics, which is the case with a triangular wave, see § 5.1.

This can be proved by first considering that the spectrum in general contains components at multiples of f_o/m, see § 4.2. Furthermore, it is assumed that the continuous analogue signal contains spectral components at only odd harmonics, these are the spectral lines $i.f_o/m$ with i = m, 3m, 5m, 7m, ... This signal is modulated around multiples of $f_k = (2^c/m).f_o$. Therefore, the spectrum will repeat itself around multiples of f_k, resulting in components at $(n.2^c \pm i.m).f_o/m$. It holds:

$n.2^c$: even
i : odd
m : odd, and therefore
i.m : odd. Then, eventually
$n.2^c \pm i.m$: odd. So the DAC output signal then contains only odd harmonics of f_o/m.

One more relevant property of the spectrum construction is the symmetry as can also be seen in table 4.1. The modulations around $n.f_k$ and $-n.f_k$ cause symmetrical sidebands. Because both modulations have the same order they will vanish at the same value of c.

This symmetry follows from rewriting the expression for one sideband. Assumption:

$$f_o + 2.k.f_o/m = n.f_k - j.f_o$$

Now the interesting sideband is the sideband at frequency $f_o - 2.k.f_o/m$. It follows from the assumed equation that

$4.k.f_o/m = 2.n.f_k - 2.(j+1).f_o$

Therefore it holds for the origin of the other sideband:

$$f_o - 2.k.f_o/m = n.f_k - j.f_o - 4.k.f_o/m$$
$$= n.f_k - j.f_o - [\ 2.n.f_k - 2.(j+1).f_o\ ]$$
$$= -n.f_k + (j + 2).f_o$$

This sideband is indeed caused by the modulation around $-n.f_k$.

Experiments are performed of cases c=4 and c=5 of the implementation of table 4.1 in the way described in appendix 1. Spectra of the DAC output signal are given in appendix 4. The expectation of the spectrum amplitudes based on a computer simulation is added. Analyzing these spectra shows that all main sidebands are caused by first harmonics of different modulations (j=1 in 4.7). Comparing the two spectra shown, vanishing of modulations of odd order by increasing c can easily be seen.

4.4 FREQUENCY DOMAIN ANALYSIS II

A second approach to the Q2 quantization phenomenon in frequency domain is also illustrative. This approach can give some insight in the expected values of Bessel functions by analyzing the origin of these functions.

This approach is mainly mathematical and deals with convolutions. In this paragraph a new modelling with several clock signals will be introduced. One should note that the frequency f_{cl} is the only physically existing clock frequency. All other 'clock signals' are just mathematical resources.

To facilitate the understanding of the new modelling, first consider a situation without relevant quantization, so c = r. Reading the ROM with a certain phase increment m can be modelled as sampling the signal stored in the ROM. This stored signal is the signal which would result at the output for m=1. The output signal for m=1 has the spectrum of an analogue sine wave, quantized in d bits, which is sampled with $2^c=2^r$ samples per period.

Actually the operations sampling and amplitude quantization are swapped now compared to the model given in figure 4.1. It can be shown, however, that this is allowed. Consider for that purpose the following functions:

v(t) analogue, continuous wave shape.
$v^*(t)$ sampled wave shape, $v^*(t) = v(kT)$.
w(t) analogue, quantized wave shape; in case of a sine wave then the amplitude transitions are not equidistant.

When the signal v(t) is first sampled (*) then quantized (Q), the following happens successively:

$$v(t) \rightarrow v^*(t) = v(kT) \rightarrow Q[v(kT)] = w(kT)$$

When the signal v(t) is first quantized then sampled this yields:

$v(t) \rightarrow Q[v(t)] = w(t) \rightarrow w^*(t) = w(kT)$

Because both output functions are equal, the sampling and quantizing operations are interchangeable.

In figure 4.10 reading of the ROM is modelled.

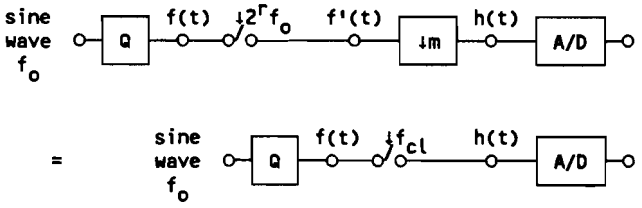


Fig.4.10 Model for reading the ROM with a frequency $f_{cl} = (2^r/m).f_o$,r=c. Reading is regarded as downsampling the signal that would result in case of m=1.

The spectrum of the signal f(t) is the spectrum of a quantized analogue sine wave. It is handled in § 5.2.2. To obtain the spectrum of signal f'(t) this spectrum should be repeated at multiples of $2^r f_o$, due to the sampling operation.

$$f'(t) = f(t).\sum_n \delta(t - n.T_o/2^r)$$

$$F'(f) = F(f) * 2^r.f_o.\sum_n \delta(f - n.2^r.f_o) \qquad (4.8)$$

In the mathematical expression for F'(f), the * stands for a convolution operation. To obtain an expression for the output signal h(t) and its spectrum, implement the downsampling:

$$h(t) = f'(t).\sum_n \delta(t - n.m.T_o/2^r)$$

$$H(f) = F'(f) * (2^r/m).f_o.\sum_n \delta(f - n.2^r.f_o/m)$$

Applying (4.8) yields:

$$H(f) = F(f) * 2^r.f_o.\sum_n \delta(f - n.2^r.f_o) * (2^r/m).f_o.\sum_n \delta(f - n.2^r.f_o/m)$$

Because the first convolution is one of the multiples of the second, the expression for H(f) can be simplified conforming with fig.4.9:

$$H(f) = F(f) * (2^r/m).f_o.\sum_n \delta(f - n.2^r.f_o/m) = F(f) * f_{cl}.\sum_n \delta(f - n.f_{cl})$$

This method of deriving an expression for H(f) can be modified for the case c<r. As is shown in previous paragraphs (see e.g. table 4.1) minimum phase increment per clock cycle increases when less bits are used to assign a ROM address. Actually the quantizing operation c<r means that the phase information sent to the ROM is the phase information out of the DTO with all (r-c) LSBs assumed to be zero, see figure 4.11.

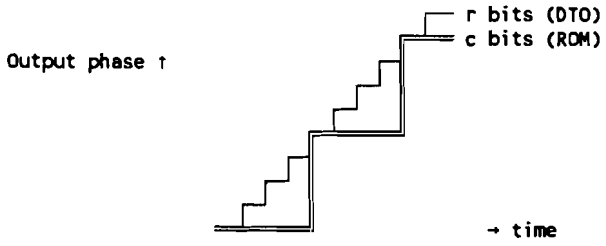


Fig.4.11 DTO output phases (r bits) and ROM input phases (c bits, c=r-2). Applying c bits to the ROM means that all r-c LSBs are supposed to be zero.

The loss of information due to this phase quantization can be modelled in a virtual ROM which contains the same data in groups of 2^{r-c} bits. That means, when this ROM is assigned with r bits, the data out of the ROM is the data belonging to the input word with all (r-c) bits zero so it is the data which would result when the physical ROM would be assigned with c bits.

Because this virtual ROM is assigned with all r bits the address increment per clock cycle is constant. Thus the problem of non-constancy (see table 4.1) is avoided this way; see figure 4.12 for the same case as shown in table 4.1.

ROM address r = 4	virtual ROM data	assigned address m=3	output data
0000	data 0	<—	data 0
0001	data 0		
0010	data 1		
0011	data 1	<—	data 1
0100	data 2		
0101	data 2		
0110	data 3	<—	data 3
0111	data 3		
1000	data 4		
1001	data 4	<—	data 4

Fig.4.12 An implementation c=r-1 can be modelled by loading a ROM with the same data in groups of $2^{r-c}=2$ addresses. This virtual ROM is then assigned with all r bits so the address increment is constant if p is constant.

Again reading the ROM with particular increment m (resulting again in h(t)) can be seen as sampling the signal which would result when all addresses would be read. This signal is called g(t), see figure 4.13.

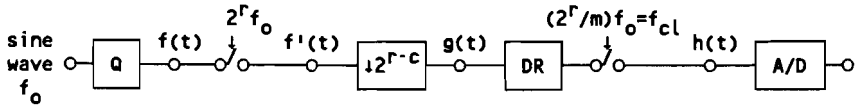


Fig.4.13 Model for reading the ROM; block DR takes care of repeating data samples 2^{r-c} times.

Finally, to get an expression for the output function the model information should just be transformed to formulas:

$$g(t) = f'(t).\sum_n \delta(t - n.T_o/2^c)$$

$$G(f) = F'(f) * 2^c.f_o.\sum_n \delta(f - n.2^c.f_o)$$

Then, with use of (4.8):

$$G(f) = F(f) * 2^r.f_o.\sum_n \delta(f - n.2^r.f_o) * 2^c.f_o.\sum_n \delta(f - n.2^c.f_o)$$

$$= F(f) * 2^c.f_o.\sum_n \delta(f - n.2^c.f_o) \qquad (4.9)$$

The data repeater (DR) is meant just to make sure that sample values are available at all moments $k.T_{cl}$. Because eventually a zero order hold device will be used behind the DAC, a data repeater does not influence the spectrum. Therefore:

$$h(t) = g(t).\sum_n \delta(t - n.m.T_o/2^r)$$

$$H(f) = G(f) * (2^r/m).f_o.\sum_n \delta(f - n.2^r.f_o/m)$$

G(f) is known (4.9), so finally

$$H(f) = F(f) * 2^c.f_o.\sum_n \delta(f - n.2^c.f_o) * (2^r/m).f_o.\sum_n \delta(f - n.2^r.f_o/m)$$

$$H(f) = F(f) * 2^c.f_o.\sum_n \delta(f - n.2^c.f_o) * f_{cl}.\sum_n \delta(f - nf_{cl}) \qquad (4.10)$$

Formula (4.10) shows that reading of the ROM in general results in an output function, the spectrum of which is a double convolution of the spectrum of the quantized analogue wave shape. First, a convolution around $2^c.f_o$ is performed. This results in the sampled signal which is actually stored in the ROM (2^c samples per period). Second this function is convoluted with f_{cl}.

4.5 COMPARISON OF TWO ANALYSES

Comparing the theory of paragraph 4.3 with the theory presented in paragraph 4.4 first consider again the case that m = 1. Then $f_{c1} = 2^r.f_o$ resulting in a special expression for the output spectrum H(f).

Formula (4.10) can now be simplified:

$$H(f) = F(f) * 2^c.f_o.\sum_n \delta(f - n.2^c.f_o) * 2^r.f_o.\sum_n \delta(f - n.2^r.f_o)$$

Because frequency $2^r f_o$ is one of the multiples of frequency $2^c f_o$ this formula again can be simplified:

$$H(f) = F(f) * 2^c.f_o.\sum_n \delta(f - n.2^c.f_o) \qquad (4.11)$$

Evidently in case m=1 (increment one address per clock period) the double convolution degenerates to a single convolution. The base spectrum will then repeat itself at multiples of $2^c f_o$. Compare this result with formula (4.2) and figure 4.5 in § 4.3.

If m≠1 the energy collected in some spectra repeating at multiples of $2^c.f_o$ is apparently spread out by the second convolution (see analysis II, formula 4.10) resulting (see analysis I, § 4.3) in modulations around multiples of $f_k = 2^c.f_o/m$. The less the spreading is, the more energy stays in the modulation around $2^c.f_o = m.f_k$. So, in general, the m^{th} modulation and its multiples are expected to be of higher amplitude, assuming no full spreading occurs.

This effect is tested in an experiment (see for the organization of the experiments appendix 1). Implementing a circuit with r=10, c=5 and m=3 results (see formula 4.2) in $f_k=32.f_o/m$ so a spectrum is expected which contains modulations around multiples of the 32^{nd} spectral component. The modulations are damped by a Bessel function. Due to the effect described above the third modulation is expected to be of higher amplitude. This indeed is the case. The resulting spectrum with simulation output is given in appendix 5.

Figure 4.14 shows once more the effect of the second convolution:

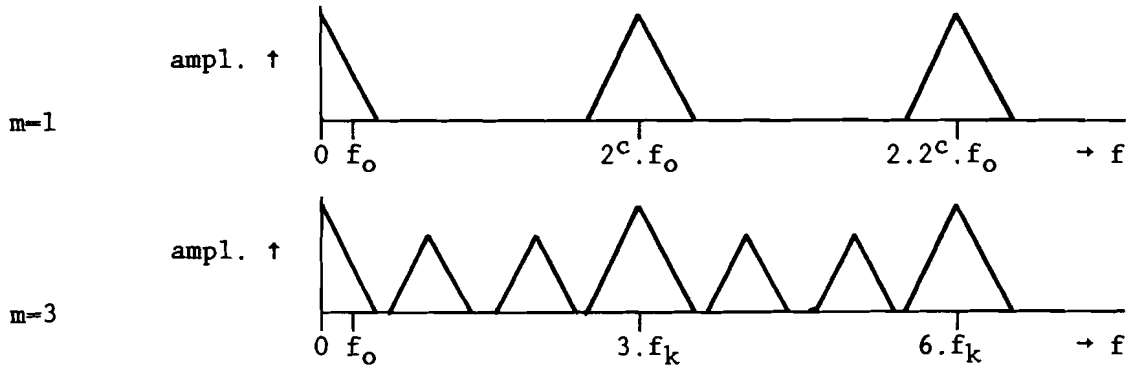


Fig.4.14 If m≠1 the second convolution causes more modulations.

5. ROM : WAVE SHAPE AND AMPLITUDE QUANTIZATION

The phase information out of the DTO is transformed to a particular wave shape by storing the corresponding amplitude values on addresses of a read only memory (ROM). In the model of figure 4.1 the physical ROM is separated into two blocks; the first block only takes care of this wave shape creation; this block will be handled in § 5.1. The amplitude quantization which always occurs due to finite word length of ROM data is put in a second block and will be handled in § 5.2.

5.1 WAVE SHAPE IN THE ROM

Although in the present case a sampled wave will result due to finite ROM length (2^c addresses), spectra of analogue wave shapes will be discussed. Due to sampling the spectrum will be repeated around multiples of the sample frequency.

Each wave shape has its characteristic frequency spectrum. The more components the spectrum contains apart from the fundamental frequency the more the output signal will be disturbed. A sine wave is the only wave shape which spectrum contains only one single component, so a sine ROM would give best results. There might be, however, some cheaper solutions which result in acceptable disturbance.

The spectrum of a wave shape is given by the Fourier series of the time signal. With use of (3.5) these series can be calculated. This is done for some basic wave shapes. Of all wave shapes it is assumed they have periodicity $T_o = 1/f_o$.

In this way for exact formulas for waves of amplitude 1 is found:

sawtooth wave : $f(t) = \frac{1}{2} + \sum_{n} (j/2\pi n).e^{2\pi jnf_o t}$

square wave : $f(t) = \frac{1}{2} - \sum_{n} (-1)^n/[\pi(2n-1)].e^{2\pi j(2n-1)f_o t}$

triangular wave: $f(t) = \frac{1}{2} - \sum_{n} (2/[\pi^2(2n-1)^2].e^{2\pi j(2n-1)f_o t}$

It follows that a sawtooth wave contains all harmonics of f_o decreasing like a 1/n curve. The spectrum of a square wave has same shape, but contains only odd harmonics. A triangular wave also contains only odd harmonics, decreasing like a $1/n^2$ curve.

The triangular wave which out of the three examples given contains least harmonic disturbance, can simply be modified in hardware by clipping its amplitude. This can be an improvement with respect to spectral contents. Whether it is an improvement or not is determined by the clipping level μ which is defined in figure 5.1.

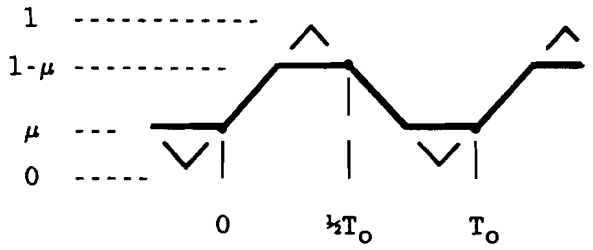


Fig.5.1 Definition of a clipped triangular wave with clipping level μ.

The Fourier coefficients c_n of a clipped triangular wave with clipping level μ as shown in figure 5.1 are given by the following formula:

$$c_n = [\ 1/(2\pi^2 n^2)\ ] \ .\ [\ e^{2\pi j n\mu}.((-1)^n-1) + (-1)^n - 1\ ]$$

Separating the amplitudes of odd and even harmonics result in :

$$c_{2n} = 0$$

$$c_{2n+1} = [\ -1/(\pi^2(2n+1)^2)\ ] \ .\ [\ 1 + e^{2\pi j\mu(2n+1)}\ ] \qquad (5.1)$$

The derivation of these formulas is given in appendix 6. No even harmonics appear in the spectrum for any value of μ. The complex value of the amplitude of odd harmonics is illustrated in figure 5.2.

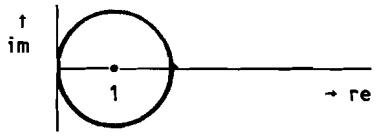


Fig.5.2 The amplitude of odd harmonics is the absolute value of a complex number on the drawn circle multiplied with the factor $[\ -1/(\pi^2(2n+1)^2)\ ]$. In dependence of μ only some particular numbers on the circle are reached.

From the formula for the amplitudes of odd harmonics in (5.1) and from figure 5.2 it follows that in dependence of μ the amplitude of some of these harmonics might be zero. This is actually the case when $e^{2\pi j\mu(2n+1)} = -1$:

$$2\pi\mu(2n+1) = (2k+1).\pi \qquad k = ..,-2,-1,0,1,2,..$$
$$2n+1 = (k+\tfrac{1}{2})/\mu$$

The right member of this equation should therefore be an odd integer. This can just be true for some particular values of μ:

$\mu = 1/6$: $c_{3+6k} = 0$
$\mu = 1/10$: $c_{5+10k} = 0$
$\mu = 1/14$: $c_{7+14k} = 0$

A clipping level $\mu = 1/4$ for example does not reduce the relative amplitude of odd harmonics because only the phase is affected then.

When μ = 1/6 most harmonics vanish. Therefore this seems to be an optimum value. However, other values of μ decrease or even eliminate other harmonics which might be more useful in some cases. Therefore the actual optimum value for μ has to be determined by the application.

The effect of clipping a triangular wave on the spectrum is measured in some experiments and simulations. In appendix 7 an example is shown for μ = 1/6. The analogue wave shape is approximated by choosing a DTO ratio f_o/f_{c1}=1/1024. This approximation, combined with the fact that μ=1/6 cannot be programmed exactly in a ROM with 2^c addresses causes the unexpected odd harmonics (c_3, c_9, c_{15}, etc.) to appear. Nevertheless their amplitude is much lower than other odd harmonics, so the effect is shown anyway.

As told before, in fact no analogue wave shapes are used when applying a ROM. Because the ROM is assumed to have 2^c adresses, the stored wave is actually a wave sampled with 2^c samples per period. This number of samples determines the smallest possible phase increment per clock period. A detailed analysis of the effect of the variable c on circuit performance is given in chapter 4.

Although the model of figure 4.1 contains a ROM, in general this does not mean that a physical ROM is always necessary. Considering the wave shapes above, a sampled sawtooth is just the DTO output, and a square wave can be achieved by taking only the MSB of the DTO output.

> It seems like a paradox arises here. For the MSB contains less information than the entire DTO output, but nevertheless a square wave contains less harmonics than a sawtooth. So system performance is improved by discarding some information available? The solution can be found in realizing that no perfect wave shapes can be realized with a practical implementation. Only at clock transients output bits can change. This makes the signal jitter. Because a more detailed phase determination can be performed when all DTO output bits are used the eventual 'sawtooth' signal will give better performance than the eventual square wave.

To create a triangular wave the DTO output except from the MSB can be used as data word. The MSB then functions as sign bit. When this bit equals '1' the other bits should be inverted. In terms of the model of figure 4.1 this means that c = d + 1. In the next chapter it will be shown that this is even the optimum relation when using a triangular wave. To create a clipped triangular wave a physical ROM can be avoided as well by applying some extra logic circuits between DTO and DAC. If all these solutions result in too much jitter in the final output signal a physical ROM -loaded with e.g. a sine wave- seems inevitable.

5.2 AMPLITUDE QUANTIZATION

Whether a physical ROM is used or not, word length of data sent to the DAC (see fig.4.1) will be finite. When d bits are connected to DAC input, 2^d words are possible. This means that any wave shape has undergone an amplitude quantization in 2^d levels.

In general an amplitude quantization causes higher harmonics to increase in the spectrum of the signal. Because of aliasing components due to sampling operations amplitude quantization can be responsible for higher sidebands near the desired frequency and will therefore enlarge the eventual jitter in the signal, even behind the reconstruction filter.

5.2.1 Signal to noise ratio

Actually the amplitude quantization can be modelled by a noise signal which is added to the continuous wave, see figure 5.3.

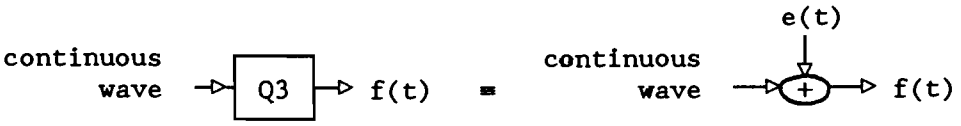


Fig.5.3 Amplitude quantization represented by added noise e(t).

This modelling makes it possible to calculate the resulting signal to noise ratio as a function of the number of quantizing levels by calculating the power in the signal e(t). In figure 5.4 all three involved functions are drawn versus time in case of a sine wave.

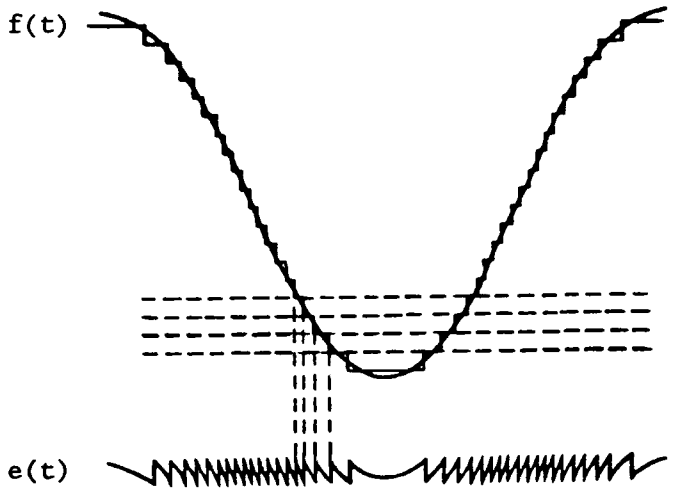


Fig.5.4 Symmetrical quantization of a sine wave; the noise signal e(t) is drawn separately.

Assuming the signal has amplitude 1, it holds for the noise power E^2:

$$E^2 = (1/12).2^{-2d} \qquad (5.2)$$

This formula can be derived in the following way. Consider a continuous wave which varies between -2^{-d-1} and $1-2^{-d-1}$. The output levels are given by $i.2^{-d}$ $(i=0,1,\ldots,2^d-1)$. Now consider a particular interval for the amplitude x of the continuous wave:

$$(i-\tfrac{1}{2})2^{-d} < x < (i+\tfrac{1}{2})2^{-d} \qquad (5.3)$$

When the continuous wave amplitude equals x it holds for the noise signal $e(t) = x - i.2^{-d}$. If d is large enough the quantization step is small compared to the signal amplitude. Then the probability density $p_i(x)$ of e(t) in this interval may assumed to be constant. This yields for the noise power E_i^2 in this interval:

$$E_i^2 = E[e^2(t)] = \int_{(i-\frac{1}{2})2^{-d}}^{(i+\frac{1}{2})2^{-d}} (x-i2^{-d})^2.p_i(x)\,dx$$

$$E_i^2 = p_i(i2^{-d}).(1/3).(x-i2^{-d})^3 \Big|_{(i-\frac{1}{2}).2^{-d}}^{(i+\frac{1}{2}).2^{-d}} = p_i(i2^{-d}).(1/3).\tfrac{1}{4}.2^{-3d}$$

Then, for the probability P_i for x to be in any particular interval it holds:

$$P_i = \int_{(i-\frac{1}{2})2^{-d}}^{(i+\frac{1}{2})2^{-d}} p_i(x)\,dx = 2^{-d}.p_i(i2^{-d})$$

This yields for the noise power in an interval:

$$E_i^2 = 2^d.P_i.(1/12).2^{-3d} = (1/12).P_i.2^{-2d}$$

Finally, the total noise power is the summation of all intervals:

$$E^2 = \sum_i E_i^2 = \sum_i (1/12).P_i.2^{-2d} = (1/12).2^{-2d} \qquad (5.2)$$

The last equality holds because by definition $\Sigma P_i = 1$.

Because signal power S is dependant on wave shape the same holds for the signal to noise ratio $S/N = S/E^2$. In general it holds:

$$S/N = S/[(1/12).2^{-2d}] = 12.S.2^{2d} = 12.S.4^d$$

It is more usual however to express this ratio in decibels:

$$S/N = 10.\log(12S) + 10.d.\log(4) = 10.\log(12S) + 6.02\ .\ d\ (dB) \qquad (5.4)$$

Formula (5.4) shows that each increase of one bit in data representation makes the signal to noise ratio of the output signal increase with about 6 dB.

5.2.2 Effects on the spectrum of a sine wave

The spectrum of an analogue sine wave contains no harmonics at all. This will change when the signal is submitted to amplitude quantization. To investigate to which extent the number of data bits (d) influences the spectrum, the Fourier series of a sine wave quantized in d bits has been derived. The derivation is given in appendix 8. Eventually for the amplitude c_n of the n^{th} harmonic has

$$(i-½)2^{-d} < x < (i+½)2^{-d} \qquad (5.3)$$

When the continuous wave amplitude equals x it holds for the noise signal $e(t) = x - i.2^{-d}$. If d is large enough the quantization step is small compared to the signal amplitude. Then the probability density $p_i(x)$ of e(t) in this interval may assumed to be constant. This yields for the noise power E_i^2 in this interval:

$$E_i^2 = E[e^2(t)] = \int_{(i-½)2^{-d}}^{(i+½)2^{-d}} (x-i2^{-d})^2.p_i(x)\,dx$$

$$E_i^2 = p_i(i2^{-d}).(1/3).(x-i2^{-d})^3 \Big|_{(i-½).2^{-d}}^{(i+½).2^{-d}} = p_i(i2^{-d}).(1/3).¼.2^{-3d}$$

Then, for the probability P_i for x to be in any particular interval it holds:

$$P_i = \int_{(i-½)2^{-d}}^{(i+½)2^{-d}} p_i(x)\,dx = 2^{-d}.p_i(i2^{-d})$$

This yields for the noise power in an interval:

$$E_i^2 = 2^d.P_i.(1/12).2^{-3d} = (1/12).P_i.2^{-2d}$$

Finally, the total noise power is the summation of all intervals:

$$E^2 = \sum_i E_i^2 = \sum_i (1/12).P_i.2^{-2d} = (1/12).2^{-2d} \qquad (5.2)$$

The last equality holds because by definition $\Sigma P_i = 1$.

Because signal power S is dependant on wave shape the same holds for the signal to noise ratio $S/N = S/E^2$. In general it holds:

$$S/N = S/[(1/12).2^{-2d}] = 12.S.2^{2d} = 12.S.4^d$$

It is more usual however to express this ratio in decibels:

$$S/N = 10.\log(12S) + 10.d.\log(4) = 10.\log(12S) + 6.02\ .\ d\ (dB) \qquad (5.4)$$

Formula (5.4) shows that each increase of one bit in data representation makes the signal to noise ratio of the output signal increase with about 6 dB.

5.2.2 Effects on the spectrum of a sine wave

The spectrum of an analogue sine wave contains no harmonics at all. This will change when the signal is submitted to amplitude quantization. To investigate to which extent the number of data bits (d) influences the spectrum, the Fourier series of a sine wave quantized in d bits has been derived. The derivation is given in appendix 8. Eventually for the amplitude c_n of the n^{th} harmonic has

been found:

$$c_{2n} = 0$$

$$c_{2n+1} = (1/j\pi(2n+1)2^d).\{\ 1+ 2.\sum_{i=0}^{2^{d-1}-1} \cos[(2n+1).\arcsin(i.2^{1-d})]\ \} \qquad (5.5)$$

So amplitude quantization never introduces even harmonics. Furthermore, all coefficients of odd harmonics are purely imaginary, so all harmonics have the same phase. To obtain some more insight in the amplitude of odd harmonics formula (5.5) is made into a computer program. The flow chart and results are given in appendix 8. The tables given in appendix 8 are graphically presented in figure 5.5.

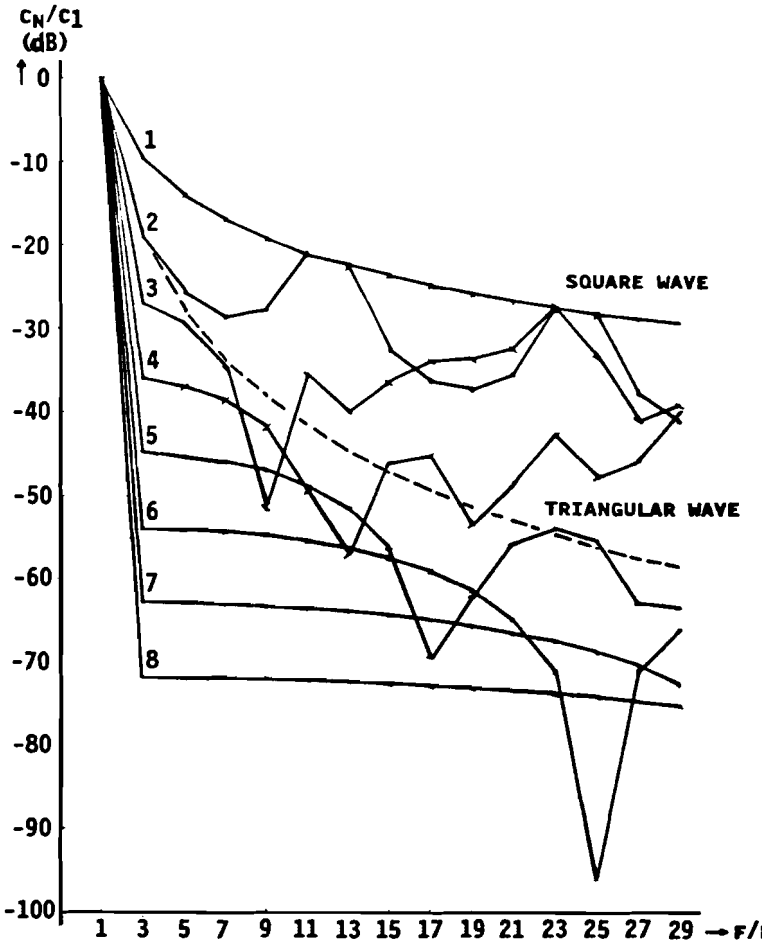


Fig.5.5 Contour of the spectrum of an analogue, quantized sine wave; the number of bits (d) is a parameter. The spectra contain only odd harmonics. The dashed line, representing the spectrum of a triangular wave is given as a reference.

Several remarks can be made referring to figure 5.5.

* Apparently when the number of bits equals 1 a square wave results. Then the spectrum has a contour like a 1/n curve.
* For large d the quantizing effect is negligible; when d=8 (256

amplitude levels) all harmonics have amplitude more than 70 dB below the fundamental frequency f_o.

* For d between 2 and 6 the spectrum is rather capricious. However, some periodicity can be distinguished. The larger d, the larger this periodicity is. Each period contains one or more very low harmonics; the larger d the sharper this negative peak in the spectrum is. Each period ends with a large harmonic which equals the corresponding amplitude when d=1 as far as shown in figure 5.5.

In [5] a model is presented which predicts high harmonics at multiples of $\pi.2^d.f_o$ which is indeed the case in figure 5.5, although spectra in figure 5.5 are more smoothed than the model predicts.

Therefore the contours of the spectra are not strictly decreasing. One reason for this is that, due to vertical edges of a theoretical quantized signal, bandwidth of the signal is infinite.

* Each increase of number of bits from d to d+1 causes the first few harmonics to diminish with about 9 dB.

5.2.3 Effects on the spectrum of a triangular wave

An analogue triangular wave contains besides its own frequency f_o all odd harmonics decreasing like a $1/n^2$ curve. When this signal is quantized in d bits the spectrum is the same as if the signal is sampled with 2^{d+1} samples per period. This means then the spectrum is repeated around multiples of $2^{d+1}.f_o$; see figure 5.6.

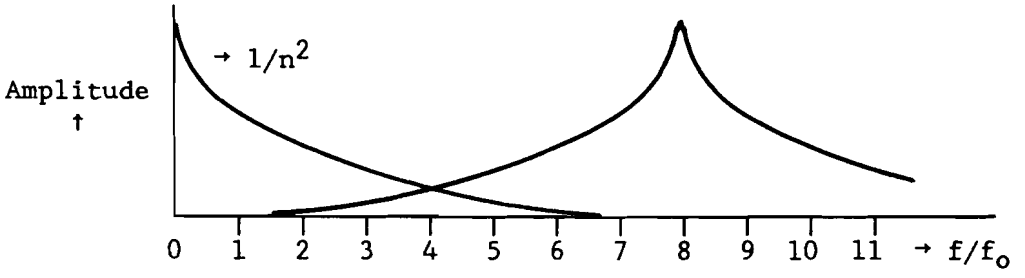


Fig.5.6 Contour of the spectrum of a 2 bits quantized triangular wave (d=2); this spectrum equals the spectrum of a continuous triangular wave sampled with 2^{2+1}=8 samples per period.

Because in the present circuit (figures 1.2 and 4.1) the wave stored in the ROM is a wave sampled with 2^c samples per period the optimum circuit implementation when a triangular ROM is applied is

$$c = d + 1 \qquad (5.6)$$

This relation is found in the following way. When a triangular wave is quantized level transitions occur at equidistant moments due to constant slope of the time function. Apart from a certain phase shift, these moments in the first half of the cycle time will be t=0 and 2^d-1 other moments to achieve 2^d levels. The same holds for the second half of the cycle time. The total number of transitions thus equals 2^{d+1}. One should note, however, that when the wave has cycle time T_o no

real level transition occurs at the moments t=0 (T_o) and t=T_o/2, see figure 5.7.

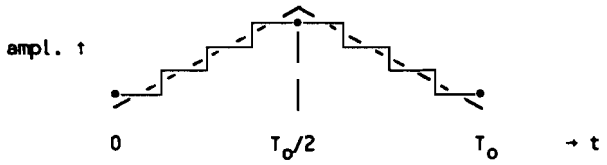


Fig.5.7 Triangular wave quantized in d=2 bits which yields 2^d amplitude levels.

If the signal shown in figure 5.7 is sampled with 2 samples a period (c=1) the sample moments are assumed to be t=0 and t=T_o/2. An increase of this number of samples yields samples at transition moments until all transitions are covered. The number of samples then equals $2^c = 2^{d+1}$. Assuming a zero order hold device behind the sampler a further increasing of c does not change the output function and is therefore of no use.

Assuming a situation when c = d + 1 a further increase of d does not influence the output signal either. Then the stair function in figure 5.7 would be refined, but because of the fixed sample moments combined with the ZOH device this will not affect the output signal.

For the circuit of figures 1.2 and 4.1 this means that when a triangular wave shape is used a DTO - ROM quantization (Q2, c < r) is of no importance as long as c ≥ d + 1. So when e.g. a 14 bits DTO and a 4 bits DAC are applied, it is no use to connect more than 5 bits from the DTO to the ROM. As already mentioned in § 5.1 this actually means that no physical ROM is needed at all. All c bits applied from the DTO can directly be used as data bits except the MSB. This bit functions as a sign bit; when the MSB equals '1' the c-1 data bits have to be inverted.

6. PLL AS RECONSTRUCTION FILTER

The ZOH output signal in the model of figure 4.1 can be filtered by a reconstruction filter according to figure 1.2 to reduce sidebands that cause time jitter. As mentioned in § 4.3 this filter should be a band pass filter (BPF) with center frequency f_o. In the closed loop of figure 1.2 the frequency of the input signal can show large variations in frequency, so the same holds for the synthesized frequency f_o. The bandwidth of a fixed BPF should be large enough to ensure that all possible frequencies f_o are in the pass band. Because all jitter caused by sidebands within the pass band will not be reduced by the BPF, a fixed filter will result in too much jitter. Therefore it is of no use for this application.

A better solution can be a narrow band tracking filter. Then the center frequency traces changes in the input frequency when variations are not too fast. This function can be realized by a phase locked loop (PLL) [6]. In this chapter some relevant PLL characteristics are investigated.

6.1 PLL PRINCIPLES

A PLL consists of three basic components. A phase detector (PD) compares the phases of the input signal and the signal of a local oscillator. The output signal is a measure of the phase difference between both signals. This PD output signal is filtered by a loop filter (LF) to remove high frequency components and then connected to a voltage controlled oscillator (VCO) which operates as local oscillator. By brief frequency variations of the VCO, phase and frequency variations at the input can be traced c.q. averaged. The VCO output signal is the circuit output, see figure 6.1.

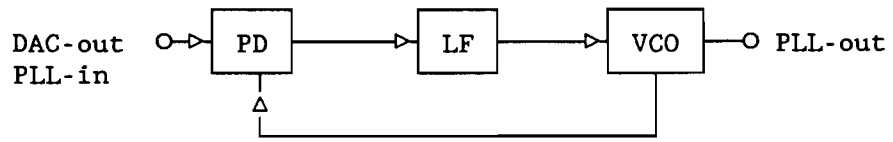


Fig.6.1 Block diagram of a phase locked loop.

The PD can be implemented as a multiplier. When both PD input signals have same frequency the output signal contains a DC term which is proportional to the (sine of) the phase difference and an AC term of double the input frequency.

When both input signals are given by:

$v_i = V_i.\sin(\omega_o t + \phi_i)$

$v_o = V_o.\cos(\omega_o t + \phi_o)$

Then for the output signal it holds :

$$v_d^* = k.v_i.v_o$$
$$= k.V_i.\sin(\omega_o t + \phi_i).V_o.\cos(\omega_o t + \phi_o)$$
$$= \tfrac{1}{2}.k.V_i.V_o.[\sin(\phi_i - \phi_o) + \sin(2.\omega_o.t + \phi_i + \phi_o)]$$

$= K_d.[\sin(\phi_i-\phi_o) + \sin(2.\omega_o.t + \phi_i + \phi_o)]$

So the phase detector constant K_d is proportional to the amplitude of both input signals. When both input signals are square waves, the DC component of the output signal is directly proportional to the time difference between rising or falling edges, so directly proportional to the phase difference. When one of the two input signals is a square wave, only the fundamental frequency of this square wave can contribute to a DC component, so no relevant changes occur compared to the case that two sinusoidal signals are applied.

An other possibility is to use flip-flops to create an edge triggered PD. Then the time difference between rising or falling edges of the input square waves is influencing the duty cycle of the output square wave and with that its DC value. In further analysis in this report, however, a multiplier PD will be assumed because of its simplicity.

The LF then removes the double frequency component in the output of the PD by averaging the signal, so it is a low pass filter (LPF). In most applications a first order LPF will satisfy. To keep the possibility to choose some basic PLL parameters independently and to ensure a possibility of an appropriate damping of the loop a first order LPF with two time constants is applied [6]. In principle both an active and a passive filter are possible. See figure 6.2 for both configurations.

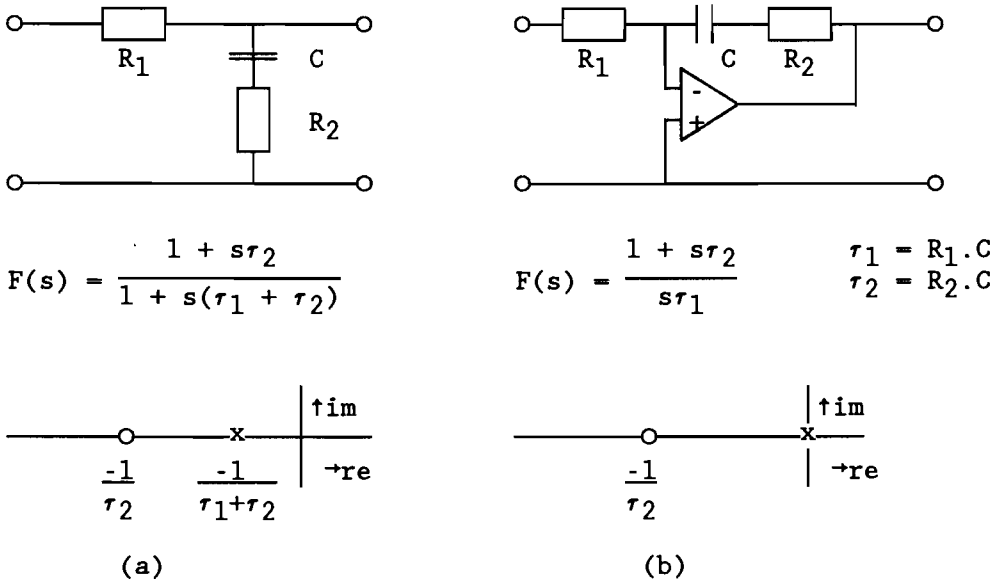


Fig.6.2 Configuration of loop filters with transfer function F(s) and pool zero plot in the s-plane; a: passive, b: active.

Consequently, apart from the ripple the LF output signal when ϕ_e is the phase error is:

$$v_d = K_d.\sin(\phi_i-\phi_o) = K_d.\sin(\phi_e) \simeq K_d.\phi_e \quad (\phi_e \text{ small}) \qquad (6.1)$$

For K_d it therefore holds $[K_d] = V/rad$.

To get an impression of the value of $K_d.\phi_e$ just remember that it can

never exceed the power voltage of the circuit, so v_d is limited to just a few volt. This voltage v_d is applied to the VCO. It holds for the VCO frequency f_o:

$$2.\pi.f_o = 2.\pi.f_c + K_o.v_d$$
$$\omega_o = \omega_c + K_o.v_d \qquad (6.2)$$

The variable f_c in (6.2) is the center frequency of the VCO. This frequency can be adjusted in both directions by v_d. The amplification factor is K_o for which it holds:

$[K_o]$ = rad/V.s

The order of magnitude of K_o can be seen from (6.2). For $K_o.v_d$ should not limit the tracking performance of the PLL (see § 6.5) the value of K_o should be like the value of the center frequency of the PLL in rad/s. The argument of the VCO output signal is the integral of its frequency:

$$\Phi_o = \int \omega_o \, dt = \omega_c.t + K_o.\int v_d \, dt$$

This yields for the phase of the VCO output signal:

$$\phi_o = K_o.\int v_d \, dt = K_o.K_d.\int \phi_e \, dt \qquad (6.3)$$

Equation (6.3) shows that the VCO averages the phase error ϕ_e. The factor $K_o.K_d$ is called loop gain; it has the dimension of a frequency. Following the loop starting with a phase lag of the VCO signal ($\phi_o<\phi_i$) a positive control voltage v_d is created which makes the VCO frequency increase. This way the phase lag is made up which makes the control voltage decrease. When no phase error is left the voltage equals zero.

6.2 TRANSFER FUNCTION

To determine the closed loop transfer function, figure 6.1 is drawn in the Laplace (s) domain with use of (6.1) and (6.3); see figure 6.3.

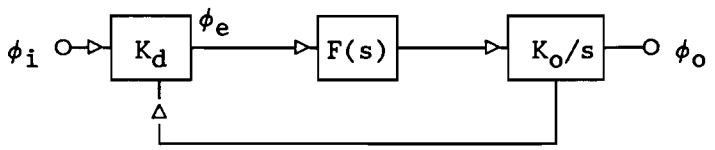


Fig.6.3 Block diagram of a phase locked loop in the s-domain.

From this figure the transfer function can be found directly:

$$H(s) = \frac{\phi_o(s)}{\phi_i(s)} = \frac{K_o.K_d.F(s)/s}{1+K_o.K_d.F(s)/s} \qquad (6.4)$$

It follows from the root locus plot of this transfer function that for both an active and a passive loop filter the loop is stable for all values of K_oK_d.

The characteristic equation is:

$1+K_o.K_d.F(s)/s = 0$

By defining $F(s)/s = T(s)/N(s)$ this yields:

$T(s)/N(s) = -1/(K_o.K_d)$

Therefore the poles of the closed loop start ($K_oK_d = 0$) at the poles of the open loop and end ($K_oK_d \rightarrow \infty$) at the zeros of the open loop. These are the poles and zeros of F(s), see figure 6.2, and an extra pole at s=0 due to the integrating function of the VCO. By applying the root locus theory [7] the root locus plot for increasing K_oK_d can now be drawn, see figure 6.4.

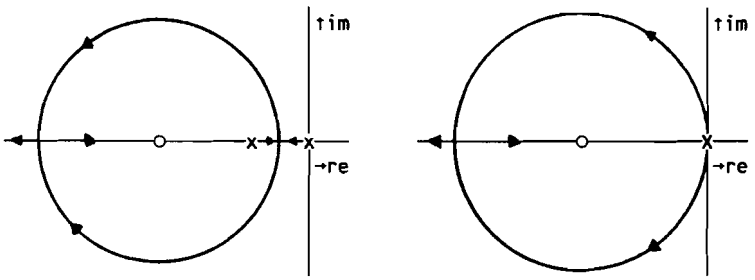


Fig.6.4 Root locus plot in the s-plane for increasing loop gain with loop filters as shown in figure 6.2; a: passive LF, b: active LF.

The break in and break away points of the real axis occur for those values of K_oK_d at which the imaginary part of s along the root locus just equals zero. Because for all values of K_oK_d the poles are in the left hand s-plane the closed loop will never be unstable [7].

Applying the formulas for F(s) given in figure 6.2 yields for H(s) given in (6.4):

$$\text{active LF : } H(s) = \frac{K_o.K_d.(1 + s\tau_2)/\tau_1}{s^2 + s.(K_o.K_d.\tau_2/\tau_1) + K_o.K_d/\tau_1}$$

$$\text{passive LF: } H(s) = \frac{K_o.K_d.(1 + s\tau_2)/(\tau_1 + \tau_2)}{s^2 + s.(K_o.K_d.\tau_2 + 1)/(\tau_1 + \tau_2) + K_o.K_d/(\tau_1 + \tau_2)} \qquad (6.5)$$

Formulas (6.5) show that a first order LF results in a second order PLL. For second order systems it is common practice in circuit and control theory to write equations (6.5) in the so called normalized form [8]:

$$H(s) = \frac{s.\omega_n.(2.\zeta - \alpha) + \omega_n^2}{s^2 + 2.\zeta.\omega_n.s + \omega_n^2} \qquad (6.6)$$

The introduced variables are the natural frequency ω_n, damping factor ζ and loss factor α. By comparison of (6.5) and (6.6) expressions for these variables can be derived. These are given in table 6.1.

Table 6.1 PLL parameter values.

LF	ζ	ω_n	α
active	$\frac{1}{2}.\tau_2.\omega_n$	$(K_o.K_d/\tau_1)^{\frac{1}{2}}$	0
passive	$\frac{1}{2}.(\tau_2 + 1/K_o.K_d).\omega_n$	$(K_o.K_d/(\tau_1+\tau_2))^{\frac{1}{2}}$	$\omega_n/K_o.K_d$

The meaning of ω_n and ζ is illustrated in figure 6.5 which shows the transfer function H(s) when an active LF is applied.

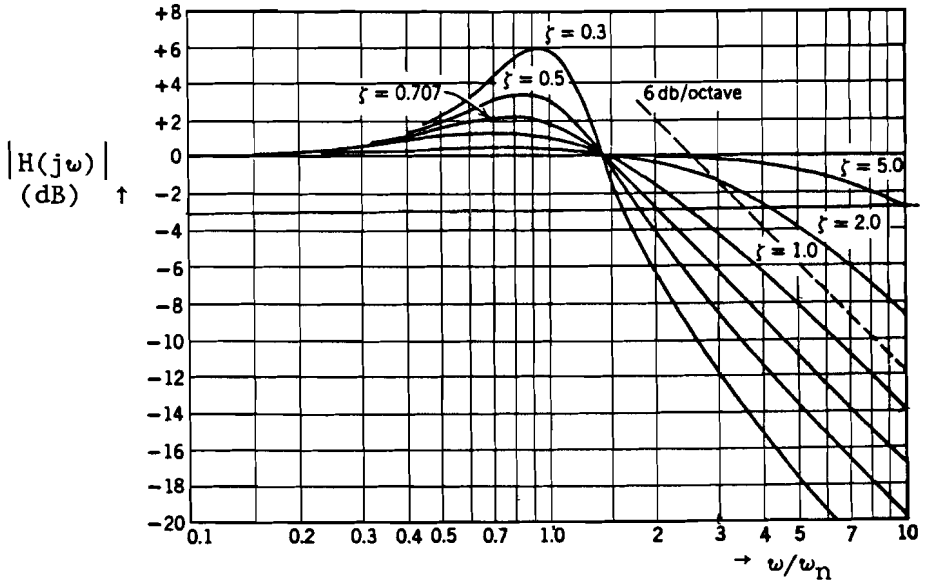


Fig.6.5 Frequency response of a second order PLL with active loop filter.

So as a coarse approximation one can say that input phase variations of a frequency smaller than ω_n will be tracked by the PLL whereas phase variations of higher frequency will be damped (averaged). In this way high frequency noise is filtered and the tracking range of the PLL is limited. The value of ζ determines the flatness of the frequency response. Both parameters also influence other quantities as will be seen later.

Because $\phi_e = \phi_i - \phi_o$ it holds:

$$\frac{\phi_e(s)}{\phi_i(s)} = \frac{\phi_i(s)-\phi_o(s)}{\phi_i(s)} = 1-H(s) = \frac{1}{1 + K_o.K_d.F(s)/s} = \frac{s^2 + \alpha.\omega_n.s}{s^2 + 2.\varsigma.\omega_n.s + \omega_n^2} \qquad (6.7)$$

This is the general expression of a second order high pass filter. Here it means that high frequency input phase variations are passed on to the phase detector output, because then the VCO phase ϕ_o does not track. By the way, when input phase variations are caused by input jitter due to the earlier mentioned quantization in a DTO circuit and not by deterministic frequency change the PLL should not track; then a phase 'error' $\phi_e \neq 0$ is wanted.

In general, a narrow band PLL (ω_n small) filters out a lot of noise but the phase or frequency step response will be slow because only a small part of the spectral contents of this step is passed. This results in bad transient response.

When a constant input word p is applied to a DTO no deterministic steps will occur and all changes in the in put of a following PLL which are not harmonic can be regarded as noise. A change of the DTO input means an output frequency change for the present circuit (fig. 1.2) and therefore a frequency step to the PLL reconstruction filter. In the next paragraph frequency step response will be investigated.

6.3 PHASE ERROR DUE TO A FREQUENCY STEP

A frequency change $\Delta\omega$ (in rad/s) yields for the PLL input phase in the s-domain:

$$\phi_i(s) = \Delta\omega/s^2 \qquad (6.8)$$

Combining (6.8) and (6.7) yields

$$\phi_e(s) = \frac{\Delta\omega.(1 + \alpha.\omega_n/s)}{s^2 + 2.\varsigma.\omega_n.s + \omega_n^2} \qquad (6.9)$$

Applying inverse Laplace transformation to (6.9) gives the time function:

$$\varsigma<1: \frac{\phi_e(t)}{\Delta\omega/\omega_n} = \alpha + \frac{e^{-\varsigma\omega_n t}}{2.\omega_n.(1-\varsigma^2)^{1/2}}.\{-2\alpha(1-\varsigma^2)^{1/2}.\cos[\omega_n(1-\varsigma^2)^{1/2}t]+2(1-\alpha\varsigma).\sin[\omega_n(1-\varsigma^2)^{1/2}t]\}$$

$$\varsigma=1: \frac{\phi_e(t)}{\Delta\omega/\omega_n} = \alpha.(1 - e^{-\omega_n t}) + (1-\alpha).\omega_n.t.e^{-\omega_n t} \qquad (6.10)$$

$$\varsigma>1: \frac{\phi_e(t)}{\Delta\omega/\omega_n} = \alpha + \frac{e^{-\varsigma\omega_n t}}{2.\omega_n.(\varsigma^2-1)^{1/2}}.\{-2\alpha(\varsigma^2-1)^{1/2}.\cosh[\omega_n(\varsigma^2-1)^{1/2}t]+2(1-\alpha\varsigma).\sinh[\omega_n(\varsigma^2-1)^{1/2}t]\}$$

The peak value of the phase error can be found by equating the derivative of the functions (6.10) with respect to time to zero. This yields:

$$\zeta<1: \quad \frac{\phi_{e,max}}{\Delta\omega/\omega_n} = \alpha + (1-2\zeta\alpha+\alpha^2)^{\frac{1}{2}}.\exp\left\{ \frac{-\zeta}{(1-\zeta^2)^{\frac{1}{2}}}.\arctan\left(\frac{(1-\zeta^2)^{\frac{1}{2}}}{\zeta-\alpha}\right) \right\}$$

$$\zeta=1: \quad \frac{\phi_{e,max}}{\Delta\omega/\omega_n} = \alpha + (1-\alpha).\exp\left\{ \frac{-1}{1-\alpha} \right\} \qquad (6.11)$$

$$\zeta>1: \quad \frac{\phi_{e,max}}{\Delta\omega/\omega_n} = \alpha + (1-2\zeta\alpha+\alpha^2)^{\frac{1}{2}}.\exp\left\{ \frac{-\zeta}{(\zeta^2-1)^{\frac{1}{2}}}.\operatorname{arctanh}\left(\frac{(\zeta^2-1)^{\frac{1}{2}}}{\zeta-\alpha}\right) \right\}$$

$$= \alpha + (1-2\zeta\alpha+\alpha^2)^{\frac{1}{2}}.\left[\frac{\zeta-\alpha+(\zeta^2-1)^{\frac{1}{2}}}{\zeta-\alpha-(\zeta^2-1)^{\frac{1}{2}}} \right]^{-\frac{1}{2}\zeta/(\zeta^2-1)^{\frac{1}{2}}}$$

When an active LF is applied is $\alpha = 0$. Formulas (6.11) then really simplify to set (6.12).

$$\zeta<1: \quad \frac{\phi_{e,max}}{\Delta\omega/\omega_n} = \exp\left\{ \frac{-\zeta}{(1-\zeta^2)^{\frac{1}{2}}}.\arctan\left(\frac{(1-\zeta^2)^{\frac{1}{2}}}{\zeta}\right) \right\}$$

$$\zeta=1: \quad \frac{\phi_{e,max}}{\Delta\omega/\omega_n} = \frac{1}{e} \qquad (6.12)$$

$$\zeta>1: \quad \frac{\phi_{e,max}}{\Delta\omega/\omega_n} = \exp\left\{ \frac{-\zeta}{(\zeta^2-1)^{\frac{1}{2}}}.\operatorname{arctanh}\left(\frac{(\zeta^2-1)^{\frac{1}{2}}}{\zeta}\right) \right\} = \left[\frac{\zeta+(\zeta^2-1)^{\frac{1}{2}}}{\zeta-(\zeta^2-1)^{\frac{1}{2}}} \right]^{-\frac{1}{2}\zeta/(\zeta^2-1)^{\frac{1}{2}}}$$

It follows from functions (6.11) and (6.12) that maximum transient phase error is inversely proportional to natural frequency ω_n. In TV applications this transient phase error is an important quantity. A phase shift of the chroma signal after colour decoding results in colour change and therefore picture impairment. Some subjective tests [11,12] have defined tolerable phase errors. When also the time is known which is left after a frequency step before the phase error should be limited, concrete demands for ω_n and ζ can be formulated. It is clear though that phase error demands require a large ω_n which will result in a fast loop. Curves of transient phase error when an active loop filter is applied are given in figure 6.6.

To investigate the desired ω_n realize that the entire circuit of figure 1.2 is actually a PLL itself, so it has its own loop gain $(K_oK_d)_{ent}$. A phase step $\Delta\phi$ at the input of the entire circuit results in a change of the DTO input word p. This means a frequency step $\Delta\omega$ which equals, according to (6.1) and (6.2): $\Delta\omega = \Delta\phi.(K_oK_d)_{ent}$

So when phase steps and loop gain of the entire circuit are known, frequency steps at the PLL reconstruction filter input are known as well. With use of formulas (6.10) the transient phase error can then be investigated.

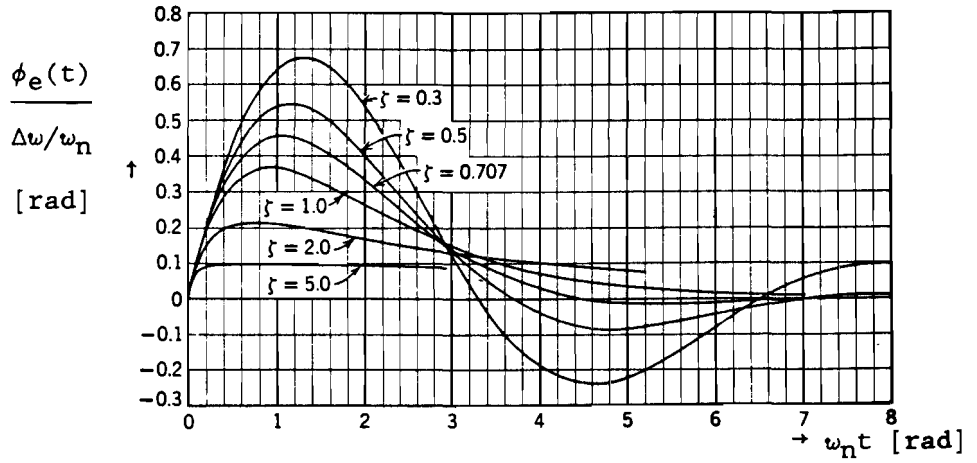


Fig.6.6 Transient phase error due to a step in frequency $\Delta\omega$ of a second order PLL with active loop filter.

To illustrate the effect of the loss factor α on this phase error the maximum value $\phi_{e,max}$ is given in table 6.2 and figure 6.7 for two values of ζ.

When the loss factor α is larger than the damping factor ζ strange effects occur due to the discontinuity of the arctangens function. These effects are not investigated because such a situation is not practical anyway.

Table 6.2 Normalized maximum phase error in radians due to a step in frequency as a function of the loss factor α of the PLL.

α	0	.1	.2	.3	.4	.5	.6	.7	.8	.9	1
$\zeta = 1$ $\phi_{e,max}/(\Delta\omega/\omega_n)$	.368	.396	.429	.468	.513	.568	.633	.711	.801	.900	$\rightarrow 1$
$\zeta = \frac{1}{2}\sqrt{2}$ $\phi_{e,max}/(\Delta\omega/\omega_n)$	.456	.494	.537	.586	.641	.704	.773	.848			

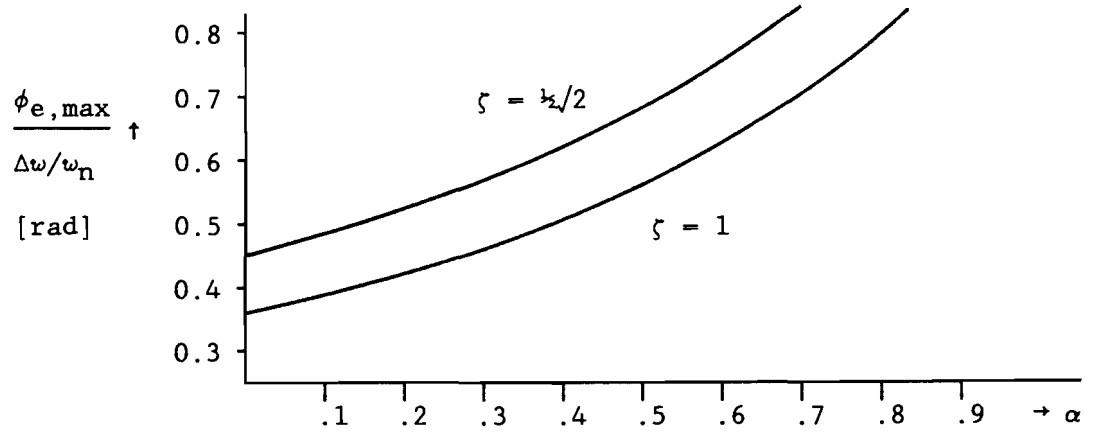


Fig.6.7 Normalized maximum phase error in radians due to a step in frequency versus the PLL loss factor α with the damping factor ζ as parameter.

Figure 6.7 clearly shows that the loss factor influences the maximum phase error in a negative way. Therefore, from this point of view, a small α is preferable. An active LF (α=0) gives best results.

Application of the final value theorem to (6.9) yields for the static phase error in case of a passive LF:

$$\lim_{t\to\infty} \phi_e(t) = \lim_{s\to 0} s.\phi_e(s) = \lim_{s\to 0} \frac{\Delta\omega.(\alpha.\omega_n + s)}{s^2 + 2.\zeta.\omega_n.s + \omega_n^2} = \frac{\Delta\omega.\alpha}{\omega_n} = \frac{\Delta\omega}{K_o.K_d}$$

Again a large loss factor makes circuit performance worse. When an active LF is applied the static phase error after a step in frequency is zero. However, when a certain static phase error is allowed, a particular passive LF might satisfy.

6.4 NOISE BANDWIDTH

To express the amount of noise which will be passed by the PLL, a certain bandwidth has been defined according to (6.13); this quantity is called loop-noise bandwidth [6]:

$$B_n = \frac{1}{2.\pi} \int_0 \frac{|H(j\omega)|^2}{|H(0)|^2} d\omega \qquad (6.13)$$

It is obvious that this quantity should be as small as possible to reduce resulting time jitter in the output signal most. With $H(j\omega)$ according to (6.6) this formula can be transformed to [9]:

$$B_n = (\omega_n/8\zeta).[\ (2\zeta - \alpha)^2 + 1\ ] \qquad (6.14)$$

So the noise bandwidth is proportional to the natural frequency. Therefore a fast loop has a large noise bandwidth and reversed. The influence of the loss factor α is drawn in figure 6.8.

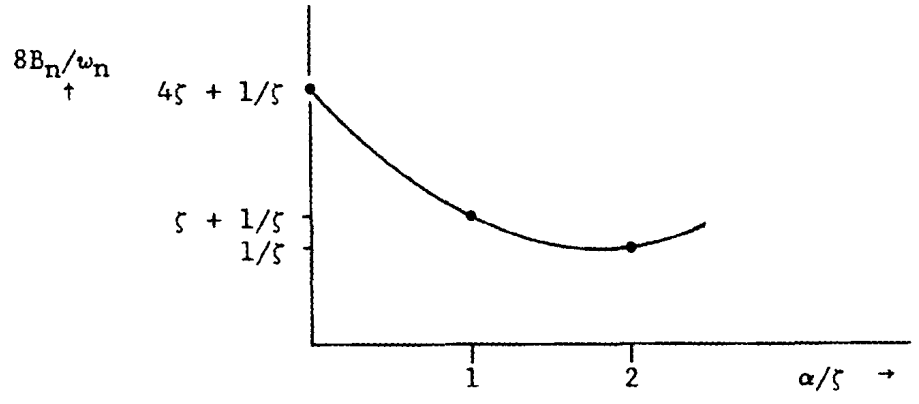


Fig.6.8 Normalized loop noise bandwidth versus α/ζ.

should note that the distance between marked points on the
ical axis in figure 6.8 is dependent on the damping factor.
efore this figure gives only qualitative information. Furthermore,

as told before, in practice the loss factor will be much smaller than the damping factor. Therefore only for small values of α/ζ the curve will be useful. However, main conclusion is that with respect to noise bandwidth some loss can be useful; once the damping factor ζ has been selected, the larger the loss factor, the smaller the noise bandwidth.

To interpret the influence of ζ on noise bandwidth, consider a situation with α=0 (active LF). Then (6.14) simplifies to:

$B_n = (\omega_n/2).(\zeta + 1/4\zeta)$

This expression is plotted in figure 6.9. The optimum is ζ=½ as can be seen in this figure. Differentiation of (6.14) with respect to ζ yields that in general minimum noise bandwidth results when $\zeta = ½./(\alpha^2+1)$.

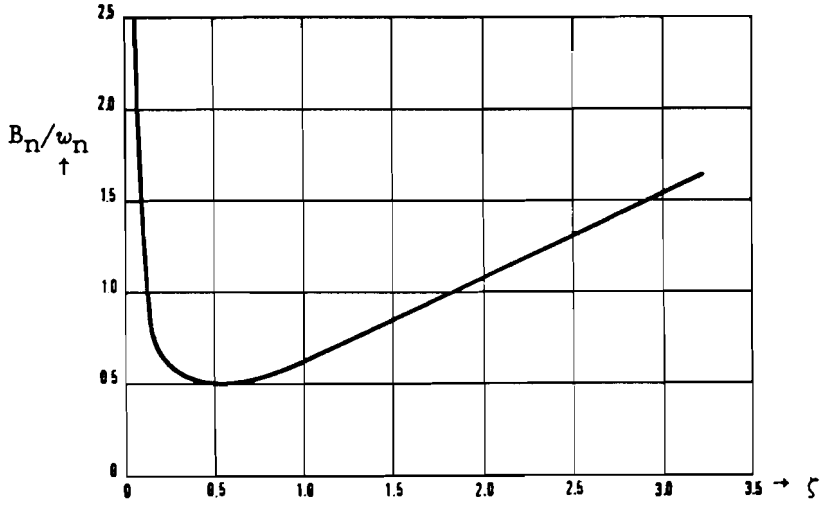


Fig.6.9 Loop noise bandwidth of a second order PLL with active LF versus damping.

6.5 TRACKING PERFORMANCE

When the loop cannot track phase variations at the input because they are of too high a frequency, it is said to be out of lock. In dependance of parameter values a certain difference is possible between input frequency and VCO center frequency before the loop loses its lock state.

For a loop which is in lock (6.2) holds, that means for the frequency difference can be written:
$\Delta\omega = K_o.v_d = K_o.K_d.F(0).\sin(\phi_e) \leq K_o.K_d.F(0)$. So when frequency difference becomes larger than $K_o.K_d.F(0)$, the VCO cannot deliver the right frequency any more and the loop will fall out of lock. This frequency difference is called hold-in range ($\Delta\omega_H$):

$$\Delta\omega_H = K_o.K_d.F(0) \qquad (6.15)$$

When a passive LF is applied, F(0) = 1, and then hold-in range equals loop gain. For an active LF $F(0) \rightarrow \infty$. It is clear though that the lock-in range cannot be infinite. Its practical value is then determined by maximum PD output voltage c.q. maximum VCO frequency deviation.

There is some frequency step limit below which the loop does not skip cycles but remains in lock. In literature this limit is called 'pull-out frequency'. Experiments performed [6] state that its value can be approximated by:

$$\Delta\omega_{po} = 1.8\omega_n.(\zeta + 1) \qquad (6.16)$$

This quantity is relevant when e.g. the output signal is used to clock a second DTO. In digital TV receivers this way subcarrier locked signals are generated out of line locked clock signals [10]. To ensure an appropriate colour decoding the line locked output signal of the first DTO should not then skip cycles.

A loop which is not in lock delivers a sinusoidal control voltage to the VCO. When $\omega_i - \omega_o = \Delta\omega$ it holds for this voltage v_d:

$$v_d = K_d.\sin[\Delta\omega.t].|F(j\Delta\omega)| \qquad (6.17)$$

So the peak deviation of the VCO frequency is $K_o.K_d.|F(j\Delta\omega)|$. When this peak deviation is smaller than $\Delta\omega$, the VCO will reach the right frequency within one cycle. This limit for $\Delta\omega$ is called <u>lock-in range</u> $\Delta\omega_L$:

$$\Delta\omega_L = K_o.K_d.|F(j\Delta\omega_L)|$$

Because the LF is a LPF a safe approximation for $F(j\Delta\omega_L)$ can be found by using the high frequency value for $F(j\omega)$. Using fig.6.2 and table 6.1 then results in

Passive LF: $\Delta\omega_L = K_o.K_d.\tau_2/(\tau_1+\tau_2) = 2.\zeta.\omega_n - \omega_n^2/K_oK_d$

Active LF : $\Delta\omega_L = K_o.K_d.\tau_2/\tau_1 = 2.\zeta.\omega_n$

The lock-in transient occupies a time on the order of $1/\omega_n$ seconds [6]. However, even when the frequency difference exceeds $\Delta\omega_L$ the loop might fall in lock after some skipping cycles. This can be understood as follows. Because the VCO frequency ω_o is modulated by v_d given by (6.17) is $\Delta\omega(t)$ not a constant. It varies harmonically. Therefore $\omega_o(t)$ is not modulated harmonically, but varies in the way presented in figure 6.10 [8]. Because the average value of the control voltage v_d is not zero, the VCO frequency ω_o is pulled in the right direction.

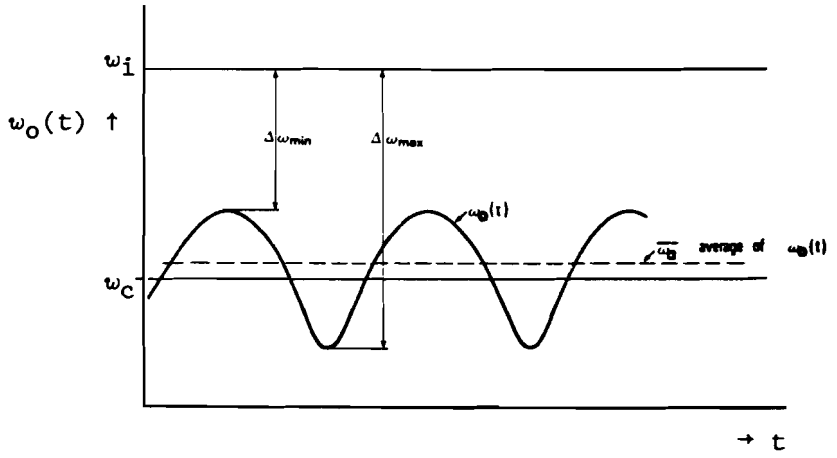


Fig.6.10 In the unlocked state of the PLL the frequency modulation of the VCO output signal is nonharmonic. This causes a pull-in effect; the average value of $\omega_o(t)$ is pulled in the right direction.

The duration of the pull-in process for all second order PLLs turns out to be quadratic in the frequency difference ω_i-ω_o. An approximating formula is given by [8]:

$$T_p \simeq \Delta\omega^2 \,/\, (2\zeta\omega_n^3) \qquad (6.18)$$

Although the pull-in process can occur outside the lock-in range, it only occurs when the initial frequency offset $\Delta\omega$ is smaller than a critical value. This value defines the pull-in range $\Delta\omega_p$:

$$\text{Passive LF: } \Delta\omega_p \simeq \frac{4.K_o.K_d}{\pi}.\left[\frac{2.\tau_2}{\tau_1+\tau_2}\right]^{\frac{1}{2}} = (8/\pi).\sqrt{(\zeta.\omega_n.K_o.K_d - \tfrac{1}{2}.\omega_n^2)} \qquad (6.19)$$

$$\text{Active LF : } \Delta\omega_p \simeq \frac{4.K_o.K_d}{\pi}.\left[\frac{2.\tau_2}{\tau_1}\right]^{\frac{1}{2}} = (8/\pi).\sqrt{(\zeta.\omega_n.K_o.K_d)}$$

Whenever the frequency offset is outside the pull-in range a pull-in process does not take place because then the pulling effect is not regenerative [8]. The pull-in range should be large enough to guarantee that the PLL can cope with all possible input frequencies. In particular static variation of line frequency in VCR systems or transmitting waves define a minimum for the pull-in range. In practice these demands require a wide band PLL.

6.6 LINK TO PRACTICAL IMPLEMENTATION

Translation of the expressions given in previous paragraphs to concrete PLL parameter suggestions is strongly dependant on the application of the circuit. It is therefore impossible to draw general conclusions. Moreover, as follows from previous paragraphs, most demands on circuit performance lead to conflicting parameter demands.

A compromise is therefore inevitable.

Especially the demands on resulting time or phase jitter on the one hand and demands on frequency step response and pull-in range on the other hand are contradictory. The first group requires a narrow band PLL so a small ω_n. A value $\zeta = \frac{1}{2}\sqrt{2}$ then fits best. This value also gives a flat frequency response. However, a narrow band PLL is very slow. This means that a frequency step results in unacceptable phase errors, and it might even be impossible to lock at all appearing line frequencies due to limited pull-in range.

When the desired pull-in range is known, demands on ω_n can be formulated. This will surely result in a proposal for a wide band PLL. Furthermore, phase errors demands will result in a similar proposal but these demands are based on subjective tests and it is therefore hard to formulate them exactly. By applying a larger ζ than the common value, ω_n can be chosen smaller, thus reducing loop-noise bandwidth.

Although a passive LF influences loop-noise bandwidth in a positive way, demands on static phase error after a frequency step strongly require appliance of an active LF. For the VCO it just follows from the analysis presented that parameter K_o should have the same order of magnitude (in rad/Vs) as the desired frequency ω_o (in rad/s). Of course when selecting a practical VCO its own parameters as internal jitter etc. should be investigated. Computer simulations and experiments can simplify decision making.

Assuming an appropriate PLL can be designed, in figure 6.11 two possible configurations are shown to synthesize several frequencies required for digital TV applications. It is possible to generate several clock frequencies out of one DTO synthesis by locking not only on f_o, but also on $f_{cl}-f_o$, $f_{cl}+f_o$, etc. The figure is mainly meant as an illustration to circuit application. In the first configuration given two modified DTOs are used, i.e. $q \neq 2^a$. See for this implementation § 2.2. The second configuration given uses only one DTO, but needs some frequency mixers.

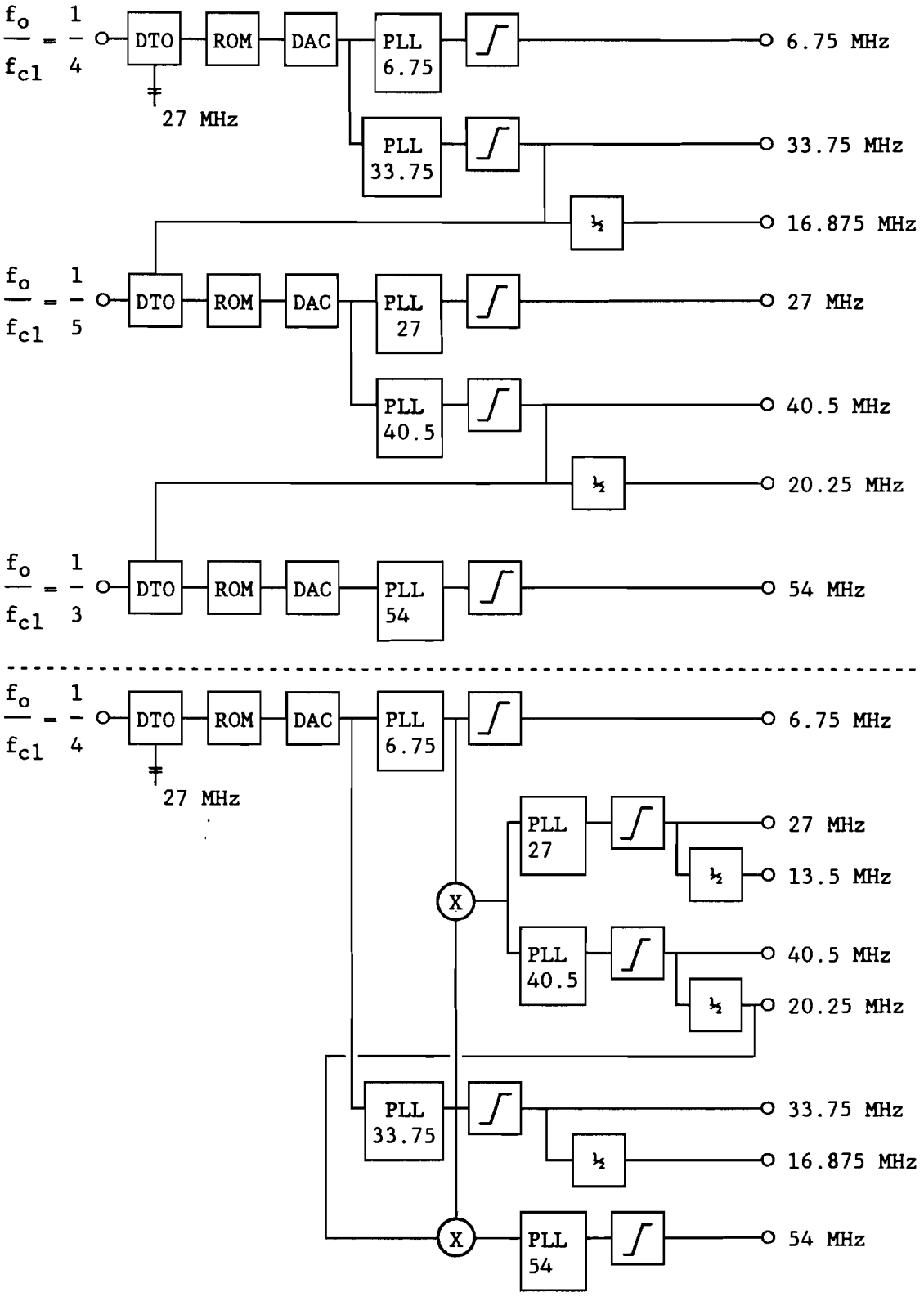


Fig.6.11 Two possible configurations to synthesize several line locked frequencies out of one crystal clock; all these frequencies are required in digital TV applications.

6.7 COMPUTER SIMULATION

Surprising to say here, a PLL has not been simulated. However, as part of the software package described in appendix 3, resulting time jitter after a reconstruction filter implemented as a fixed analogue BPF is calculated. It might be interesting for further analysis to replace this filter by a PLL and then use the same package. For this reason a short description of this simulation is given.

After execution of the program described in § 3.2 data samples of the DAC output signal (figure 1.2) are available. To simulate an analogue reconstruction filter these data samples are used to define a voltage source as a pulse train. All pulses have width T_{cl} while the amplitude equals the instantaneous data value, thus simulating the ZOH output (see fig.4.1). Because this signal is periodic, a continuous source can and is defined. When the periodicity reaches some excessive time period it is approximated using a window to limit declaration time. This source is input to a circuit which contains a BPF with center frequency f_o. The type and order of the filter are user defined parameters. The output signal of this circuit is calculated by the software package Philpac, which is described in [13]. Then cycle times and therefore time jitter of the filter output signal are calculated as time difference between zero crossings with positive slope, using interpolation when necessary. Finally a table with momentary and maximum jitter is available. All relevant flow charts and an output example are given in appendix 3.

7. CONCLUSIONS

A model is presented for a circuit which contains a discrete time oscillator, read only memory, digital to analogue converter and a reconstruction filter. On the basis of this model the circuit has been analyzed. A computer simulation package calculates amplitudes and phases of sidebands of the DAC output signal and the resulting time jitter at the output of an analogue reconstruction filter at a particular circuit implementation. This package can function as a tool for a practical circuit designer.

The analysis can be used for a forward circuit analysis, i.e. the influence of individual parameters on system performance can be examined with a certain implementation; then the resulting performance can be compared to the actual demands. It is quite impossible to fulfill a backward circuit analysis, i.e. to find an optimal implementation given some performance demands. This holds because there is no unique optimum due to exchangeable parameter variations and application dependant technical and economical reasons.

The analysis has not been finished. Particularly some statistical analysis can be done on the resulting jitter due to sidebands. Furthermore in the simulation package the band pass filter can be replaced by a PLL to allow one to simulate the entire practical circuit. Then, for various applications a circuit synthesis can be performed. Finally, the analysis can be extended to a closed loop to examine control effects that then occur.

REFERENCES

[1] Robson, T.S.
Why IBA says MAC for Europe.
Electronics & Power, pp 578-580, September 1982

[2] Nillesen, T.
Line locked digital colour decoding.
Acta Electronica, vol. 27, no. 1-2, pp 101-107, 1985

[3] Davidse, J.
Elektronische beeldtechniek.
Het Spectrum B.V., Prisma-Technica 51, Utrecht/Antwerpen 1973

[4] Carlson, A.B.
Communication Systems.
Mc Graw Hill Inc, Tokyo 1981

[5] Claassen T.A.C.M. and A. Jongepier
Model for the power spectral density of quantization noise.
IEEE Transactions on acoustics, speech, and signal processing,
vol. ASSP-29, no. 4, pp 914-917, August 1981

[6] Gardner, Floyd M.
Phaselock techniques.
John Wiley & sons Inc, New York 1966

[7] Dorf, Richard C.
Modern Control Systems.
Addison-Wesley Publishing Company, Phillipines 1980

[8] Best, Roland E.
Phase Locked Loop, Theory Design & Applications.
Mc Graw Hill Inc, New York 1984

[9] Faatz, J.A.W. et al.
Elektronische Schakelingen: bijzondere onderwerpen.
Lecture notes, Eindhoven University of Technology, Eindhoven 1983

[10] N.N.
Digital Video Signal Processing.
Overhead sheets, Philips Elcoma 1986

[11] Devereux, V.G.
Pulse code modulation of video signals: subjective tests on acceptable limits for timing jitter in the decoded analogue samples.
BBC research department, report no. 1971/42, Kingswood Warren, Tadworth, Surrey, nov 1971

[12] Devereux, V.G. & G.C. Wilkinson
Digital video: effect of PAL decoder alignment on the acceptable limits for timing jitter.
BBC research department, report no. 1973/1, Kingswood Warren, Tadworth, Surrey, feb 1973

[13] Essen, R.F. & D.J. Gibbs
Philpac Reference Manual.
Philips Corporate ISA, Eindhoven 1983

[14] Hellenthal, J.W.
Digitale oscillator.
Graduation report H.T.S. Hilversum, Hilversum 1986

[15] Valks, M.T.W.M.
Dipro, User's manual version 3.1.
Philips Nat. Lab., report no. 5975, Eindhoven 1983

BIBLIOGRAPHY

These works have not been explicitly referred to in the text, but they contributed to the described analyses and can provide for a general background.

1 Heerkens, H.J.
Bemonsterde PLL voor de horizontale synchronisatie van een digitale TV ontvanger.
M.Sc. report Eindhoven University of Technology, Eindhoven 1985

2 Kuo, Benjamin C.
Digital Control Systems.
Holt-Saunders International Editions, Tokyo 1981

3 N.N.
Digitale Video Signalverarbeitung, 1. Teil.
Report Valvo, Hamburg 1986

4 Berkhout, P.J.
D/A en A/D omzetting.
Philips cursus 'Tijddiscrete signaalverwerking', Eindhoven 1983

5 Gersho, A.
Quantization.
IEEE Communication Society Magazine, pp 16-29, sept 1977

6 Lamnabhi, M. & Ph. Salembier
Effects of sampling clock imperfections on digital conversion.
Report Laboratoires D'electronique et de Physique Appliqué, Limeil-Brevannes Cedex, juli 1986

7 Puijenbroek, C. van
Computer simulatie van een number controlled oscillator.
Graduation report H.T.S. Eindhoven, Eindhoven 1985

8 Nührmann, D.
Das groβe Werkbuch Elektronik.
Franzis Verlag GmbH, München 1984

9 Zverev, Anatoli I.
Handbook of filter synthesis.
John Wiley & sons Inc., New York/London/Sydney 1967

10 Tietze U. & Ch. Schenk
Halbleiter Schaltungstechnik - Fünfte Auflage.
Springer Verlag, Berlin 1980

11 Robins, W.P.
Phase noise in signal sources (Theory and applications).
Peter Peregrinus Ltd, London 1982

12 Davidse, J. et al.
Phase Lock Loop.
Post academic course TH Delft, Delft 1980

13 Spiegel, Murray R.
Mathematical handbook of formulas and tables.
Mc Graw Hill Inc, New York 1968

14 Naslin, P.
Les régimes variables dans les systèms linéaires et non-linéaires.
Dunod, Paris 1962

APPENDIX 1 : ORGANIZATION OF EXPERIMENTS PERFORMED

In several chapters some results of experiments performed are presented. The main goal of these experiments is to verify both theory and simulation results. This appendix deals with the structure of these experiments.

All measurements are executed at a circuit board which contained a 24 bits DTO (a=24), a 1k x 8 bits random access memory (RAM) and an 8 bits digital to analogue converter (DAC) with internal zero order hold (ZOH) device. The ZOH output has been used as circuit output. This circuit board is described in detail in [14].

The RAM fulfils the ROM function in the model of figure 4.1. It can be loaded via a personal computer with several wave shapes. A computer program written in Turbo Pascal takes care of this loading action. It offers the possibility to define a 2^c x d memory ($1 \leq c \leq 10$, $1 \leq d \leq 8$). A variable number of addresses is simulated by loading the same data in groups of 2^{10-c} addresses conforming with the theory of § 4.4. A variable number of quantizing levels is achieved by defining integer values as data samples when the wave has amplitude 2^d. The following wave shapes can be loaded this way:

* sine wave
* triangular wave
* clipped triangular wave, μ = 1/4 (see fig.5.1)
* clipped triangular wave, μ = 1/6 (see fig.5.1)
* square wave (DTO - MSB)
* zero function

The complete program is given below:

```
PROGRAM DTOInterface;
  (* This program controls the parallel-Centronics gate and the output of the
     1024*8 RAM memory of the DTO function generator.
                                        last update 20-05-87 V.J.H. Konijn - CAB *)
label  start, stop, cont, leesq;
const  BusPortNr=$03BC;
        InPortNr=$03BD;
       OutPortNr=$03BE;
var    c,d,n,m,f,g: integer;
       q,z:char;
procedure DataOut(value:byte);
  (* This procedure controls the 8 bit data bus of the parallel gate. *)
  begin
    port[BusPortNr]:=value
  end;
procedure ControlOut(store,strobe1,strobe2:integer);
  (* This procedure controls the control-lines STORE, STROBE1 enSTROBE2. In the
     program these lines are active high *)
  begin
    port[OutPortNr]:=strobe1+2*strobe2+8*store
  end;
function acknowledge: boolean;
  (* This function checks the level of the acknowledge control line. High level:
     no acknowledge. Low level: acknowledge *)
```

```
  begin
    acknowledge:=(port[InPortNr] and $80 )=0;
  end;
function funktie(q:char;n,f,g:integer):byte;
  (* This function defines the RAM contents as if a f*g ROM (f=2^c,g=2^d) is loaded
     with data samples of one period of a particular wave selected by the value
     of q; n is the selected address in the RAM, from 0 to 1023 *)
  var h,k:byte;
  begin
    case q of
    '1': begin (* sine *)
           k:=254 div (g-2);
           h:=round(0.5*(g-2+(g-1)*sin(2*pi*n/f)));
           if h = -1 then h := 0;
           funktie := k*h;
         end;
    '2': begin (* triangular *)
           k:=255 div (g-1);
           if n <= f div 2 then funktie:=k*round(-0.5+(g/f)*(2*n-1))
                           else funktie:=k*round(-0.5+g*(1-(2*n-1-f)/f));
         end;
    '3': begin (* 50% clipped triangular *)
           k:= 254 div (g-2);
           if n<f div 4 then funktie:=k*round(-0.5+4*n*((g-1)/f))
                        else if n<= f div 2 then funktie := k*(g-2)
                        else if n<3*(f div 4)-1
                             then funktie:=k*round(-0.5+(g-1)*(1+2*(f-2*n)/f))
                             else funktie :=0;
         end;
    '4': begin (* 33% clipped triangular *)
           k:=254 div (g-2);
           if n<=f div 3 then funktie:=k*round(-0.5+3*n*((g-1)/f))
                         else if n<=f div 2 then funktie := k*(g-2)
                         else if n<=(5*f) div 6 then
                                 funktie:=k*round(-0.5+(g-1)*(1+3*(f div 2-n)/f))
                              else funktie := 0;
         end;
    '5': if n<=f div 2 then funktie:=0 else funktie:=255;
         (* square: DTO-MSB *)
    '6': funktie:=0; (* zero *)
    end; (* of case *)
end; (* of function *)

begin (* main program *)
start:clrscr;
  writeln; writeln;
  writeln('           ***************************************************');
  writeln('           * POSSIBILITY TO ENTER PARAMETERS                 *');
  writeln('           * OF A DTO - RAM - DAC CIRCUIT.                   *');
  writeln('           * DEFAULT VALUES ARE PLACED IN FRONT OF THE COLON *');
  writeln('           ***************************************************');
  writeln;writeln;writeln;writeln;
  writeln('The RAM can be loaded with one of the following waves :');
  writeln;
```

```
      writeln('1: sine');
      writeln('2: triangular');
      writeln('3: 50% clipped triangular');
      writeln('4: 33% clipped triangular');
      writeln('5: square (DTO-MSB)');
      writeln('6: zero');
      writeln;
      writeln('0: exit');
      writeln;
      write('ENTER THE NUMBER OF THE DESIRED ');
      write('FUNCTION                                : ');
    leesq:
      read(kbd,q);
      if q='0' then goto stop;
      if (q='1') or (q='2') or (q='3') or (q='4') or (q='5') or (q='6')
         then goto cont else goto leesq;
    cont:
      writeln(q);
      writeln;write('We are going to create the output signal of ');
      writeln('the DTO-ROM-DAC circuit.');writeln;
      write('ENTER THE ADDRESS WIDTH OF THE ROM IN ');
      write('BITS (MAX 10)                (',c,'): ');
      readln(c);
      writeln;writeln('ENTER THE WIDTH OF THE ROM DATA IN BITS,');
      write('This is the log base 2 of the number of ');
      write('quantizing levels (MAX 8)   (',d,'): ');
      readln(d);
      f:=round(exp(c*ln(2)));g:=round(exp(d*ln(2)));

      repeat ControlOut(1,1,0) until acknowledge;
      ControlOut(1,0,0);
      clrscr;
      writeln;writeln;
      write('Do you want data to be displayed on the screen? (y/n): ');
      read(kbd,z);writeln(z);
      writeln('RAM is loaded now.');
      for n:=0 to 1023 do
      begin
        m:=n div round(exp((10-c)*ln(2))) + 1;
        repeat
          DataOut(funktie(q,m,f,g));
          if (z='y') or (z='Y') then writeln(n,'   ',m,'   ',funktie(q,m,f,g));
          ControlOut(1,0,1);
        until acknowledge;
        ControlOut(1,0,0);
      end;
      repeat ControlOut(0,1,0) until acknowledge;
      ControlOut(0,0,0);
      goto start;
    stop:clrscr;
    end.
```

Around this circuit board a measuring circuit has been built as shown in figure A1.1.

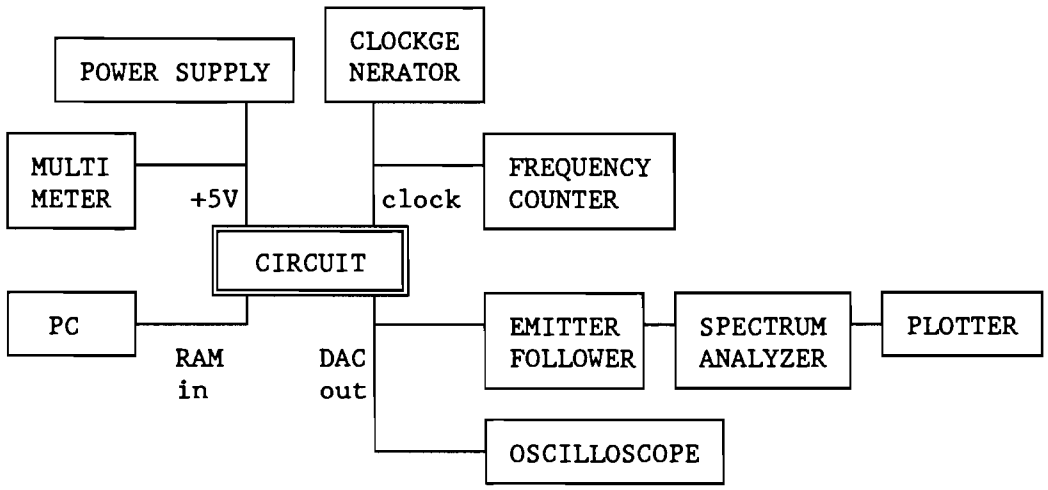


Fig.A1.1 Structure of measuring circuit.

Below the equipment used is given:

* 5V power supply : Philips PE 1540
  DC Power Supply 40V - 3A
* clock generator : Philips PM 5771
  Pulse generator 1 Hz - 100 MHz
* multimeter : Philips PM 2411
* frequency counter : Philips PM 6615
  Universal Counter 1 GHz
* personal computer : Philips P 3102
* spectrum analyzer : Hewlett Packard HP 853A
  0.1 - 1500 MHz
* plotter : Hewlett Packard HP 7470A
* oscilloscope : Philips PM 3264
  risetime 3.5 ns

Some remarks about the measuring circuit:

* As can be seen from figure A1.1, no crystal clock has been used. A common pulse generator delivered the DTO clock signal, therefore no crystal stability has been reached. This generator has an output impedance of 50 Ω. It has been connected to the circuit by a 50 Ω cable. Therefore a 56 Ω terminator has been used.
* The circuit board contains a PC interface as well. Connection to the PC is provided by a flat cable to the parallel centronics gate.
* The spectrum analyzer used has an input impedance of 50 Ω. This is not suitable to the DAC (output impedance about 500 Ω). For adjustment an emitter follower has been used as shown in figure A1.2.

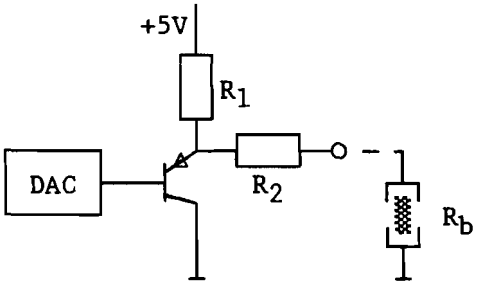
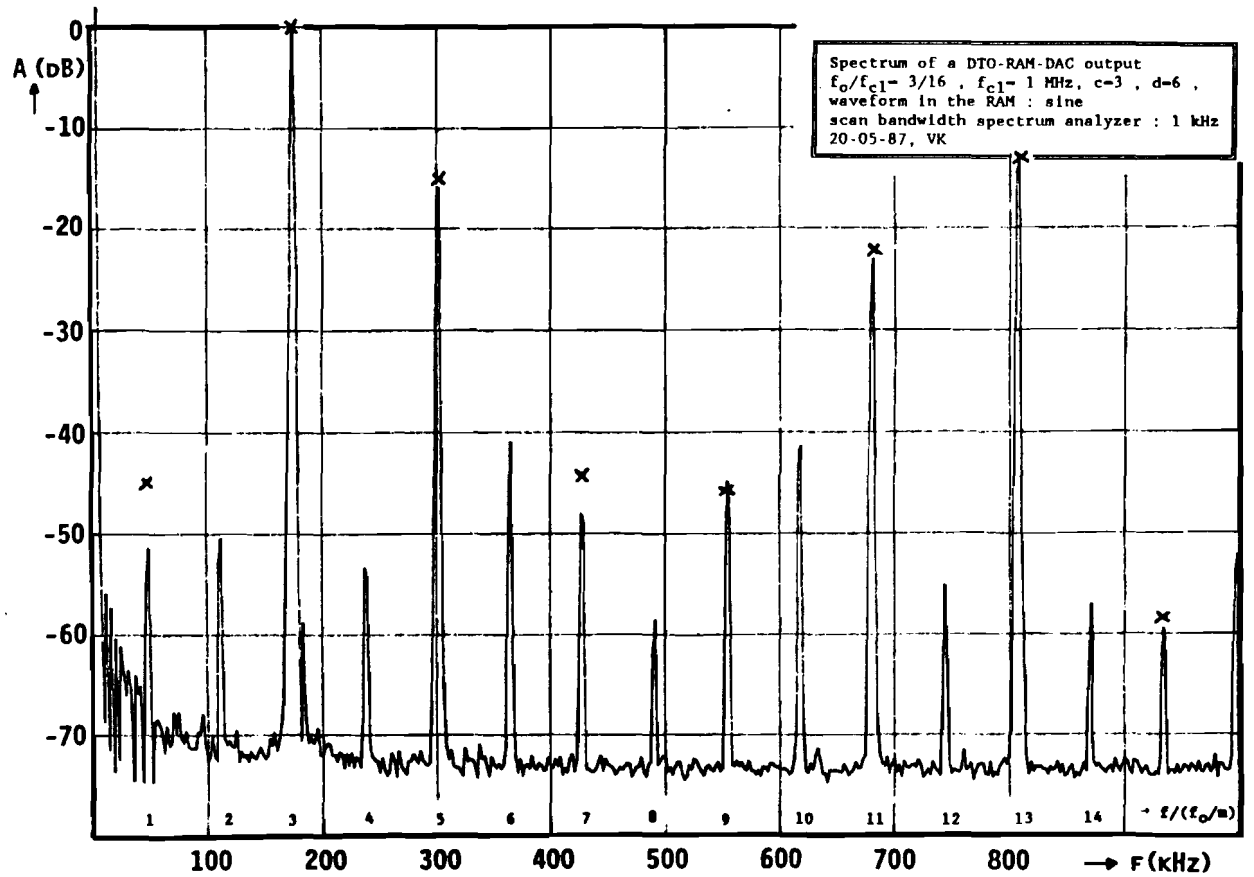


Fig.A1.2 Emitter follower for impedance adjustment.

The transistor is type BC 558. Now the output impedance is about R_2. Therefore R_2 = 47 Ω. The DAC output voltage can vary between 0 and 2 V. Because the emitter voltage should therefore be larger than 2.65 V, R_1 = 82 Ω, given a load of R_b = 50 Ω. The closed circuit current is then about 61 mA, which is large enough to ensure linear performance of the follower.

* The current used by the circuit board is measured with the multimeter; it amounts to 1.4 A .

APPENDIX 2 : EXAMPLE OF MEASURED SPECTRUM AND SIMULATION OUTPUT



* The width of the DTO is 24 bits.
* The DTO clockfrequency is: 16 MHz.
* The realized DTO output frequency is: 3.0 MHz.
* The ROM is loaded with a sine wave.
* The address width of the ROM is 3 bits.
* The data width of the ROM is 6 bits.
* The desired output frequency is the 3rd harmonic of the time discrete signal.

The spectral coefficient of the n^{th} harmonic of the time discrete signal is:

n	freq.(MHz)	norm value	phase (deg.)	ampl.(dB)
1	1	8.191E-02	56.9	-45.05
2	2	2.277E-06	103.9	-136.17
3	3	1.464E+01	0.0	0.00
4	4	3.337E-06	10.4	-132.84
5	5	2.616E+00	179.1	-14.96
6	6	2.625E-06	83.0	-134.93
7	7	8.804E-02	122.2	-44.42
8	8	3.183E-06	89.5	-133.26
9	9	6.847E-02	56.9	-46.60
10	10	1.545E-06	12.6	-139.53
11	11	1.189E+00	0.0	-21.81
12	12	2.075E-06	86.3	-136.97
13	13	3.379E+00	179.1	-12.74
14	14	3.133E-06	35.4	-133.39
15	15	5.461E-03	122.1	-68.57
16	16	8.766E-07	89.5	-144.46
17	17	4.816E-03	56.9	-69.66
18	18	1.177E-06	74.5	-141.90
19	19	2.312E+00	0.0	-16.03
20	20	3.255E-06	69.7	-133.06
21	21	6.228E-01	179.1	-27.43
22	22	7.321E-06	55.6	-126.02
23	23	2.679E-02	122.1	-54.75
24	24	5.440E-06	89.3	-128.60
25	25	2.465E-02	56.9	-55.48
26	26	2.206E-06	12.9	-136.44
27	27	4.844E-01	0.0	-29.61
28	28	2.767E-06	136.0	-134.47
29	29	1.515E+00	179.1	-19.71
30	30	1.276E-06	8.1	-141.19
31	31	2.643E-03	122.1	-74.87
32	32	8.766E-07	89.5	-144.46
33	33	2.481E-03	56.9	-75.42
34	34	4.025E-06	160.1	-131.22
35	35	1.255E+00	0.0	-21.34
36	36	4.285E-07	90.5	-150.67
37	37	3.535E-01	179.1	-32.35

APPENDIX 3 : PACKAGE DEVELOPED FOR COMPUTER SIMULATION

The DTO - ROM - DAC - BPF circuit has been simulated on a VAX computer. Two kind of outputs can be generated using the developed package:

* Spectrum of the DAC output signal (after a ZOH device); both the relative amplitudes and relative phases of the spectral components are calculated.
* Time and phase jitter of the output signal of the reconstruction filter which is implemented here as a fixed BPF.

The package can function both as verification to the theory and as tool to the practical designer. In this appendix the headings and flow charts of the programs are given. Detailed information about this package (programs and command files) has been published in a mid-term report.

```
program
SPECTRUMINVOER(input,output,invoer);
   (* this program enables the user to enter parameters of the
      DTO-ROM-DAC circuit and stores all parameter values in the file
      'invoer.dat' *)
```

For flow chart see figure A3.1.

```
program
SPECTRUM(invoer,romfile,dipfunktie,cn);
   (* this program calculates the time samples of the output signal of
      the DTO-ROM-DAC circuit at certain DTO-width (in bits), DTO-
      clock frequency and desired output frequency. These data samples
      are selected samples out of the ROM. The ROM-data (one
      particular wave) is written into the file 'romfile.dat'; the
      selected samples for one period of the time discrete signal are
      stored in the file 'dipfunktie.dat'. Then the spectral
      coefficients of this signal are calculated, assuming a
      zero-order-hold circuit behind the DAC, and stored in the file
      'cn.dat' *)
```

For flow chart see figures A3.2 and A3.3. An output example (file 'cn.dat') is given behind figure A3.3 on page 70.

```
program
DIPROPLOTINVOER(input,output,invoer,dipfunktie,chtypein,chtypedata);
   (* this program defines an adjusted data input for the 'Dipro'
      package [15] which can be used to draw a plot of the spectrum of
      the DAC output signal. To achieve this the data samples from the
      file 'dipfunktie' are repeated several times to simulate a ZOH
      device in a digital way, to a maximum of 32768 data samples.
      When the input file contains more than 32768 sample values only
      the first 32768 samples are used. The new data set is stored in
      the file 'chtypedata.dat'. All other data needed for the Dipro
      program 'CHTYPE' is stored in the file 'chtypein.dat'. The
      number of input samples (k=2^r) is read from the file 'invoer'.*)
```

For flow chart see figure A3.4.

```
program
PHILPACINVOER(input,output,invoer,dipfunktie,philp,perioddata);
   (* this program takes care of a correct input for the 'Philpac'
      package [13] which will perform a simulation of the analogue
      reconstruction filter. To achieve this, the DAC output samples
      are read from the file 'dipfunktie.dat', for one period of the
      time discrete function. These values are defined as amplitudes
      of pulses for an input source of a fixed band pass filter around
      f0. All relevant circuit parameters are read from the file
      'invoer.dat'. Order and kind of the reconstruction filter can be
      chosen. If periodicity occurs after more than 996 pulses, then a
      certain periodicity of at most 996 pulses is assumed because
      Philpac can handle only 1000 nodes. Besides a circuit
      description, the user defines the time interval and accuracy in
      which the analysis will take place. These declarations are
      stored in the file 'perioddata.dat'. The output file of this
      program, 'philp.dat' is a complete Philpac input file *)
```

For flow chart see figure A3.5.

```
program
DELETEHEADER(pasd1,pasd2);
   (* this program deletes the header of the 'pasd1.dat' file -i.e.
      the converted 'philp.ifp' Philpac output file. Then the output
      file 'pasd2.dat' contains pure data. This program uses some
      procedures from Mr. van Schagen  - Philips-Geldrop; No source
      listings of these procedures are available. The compiled version
      of this program should be linked with 'postproc.obj' and the
      'nag-library' in the following way:
      link delhead,postproc,TIS$DISK4:[NAG.NAGOLB]NAG.OLB/LIB *)
```

For flow chart see figure A3.6.

```
program
PERIODETIJDEN(pasd2,invoer,perioddata,period);
   (* this program calculates the cycle times of the waves of the
      filtered output signal, in a particular time interval as defined
      in the program 'philpacinvoer'. The input file for this program
      is the 'pasd2.dat' file; this is a converted 'philp.ifp' file.
      The cycle times and the resulting time- and phase errors are
      stored in the file 'period.dat'. Some heading information
      concerning the circuit and measuring method is read from the
      files 'invoer.dat' and 'perioddata.dat'. *)
```

For flow chart see figure A3.7. An output example (part of file 'period.dat') is given behind figure A3.7 on page 73.

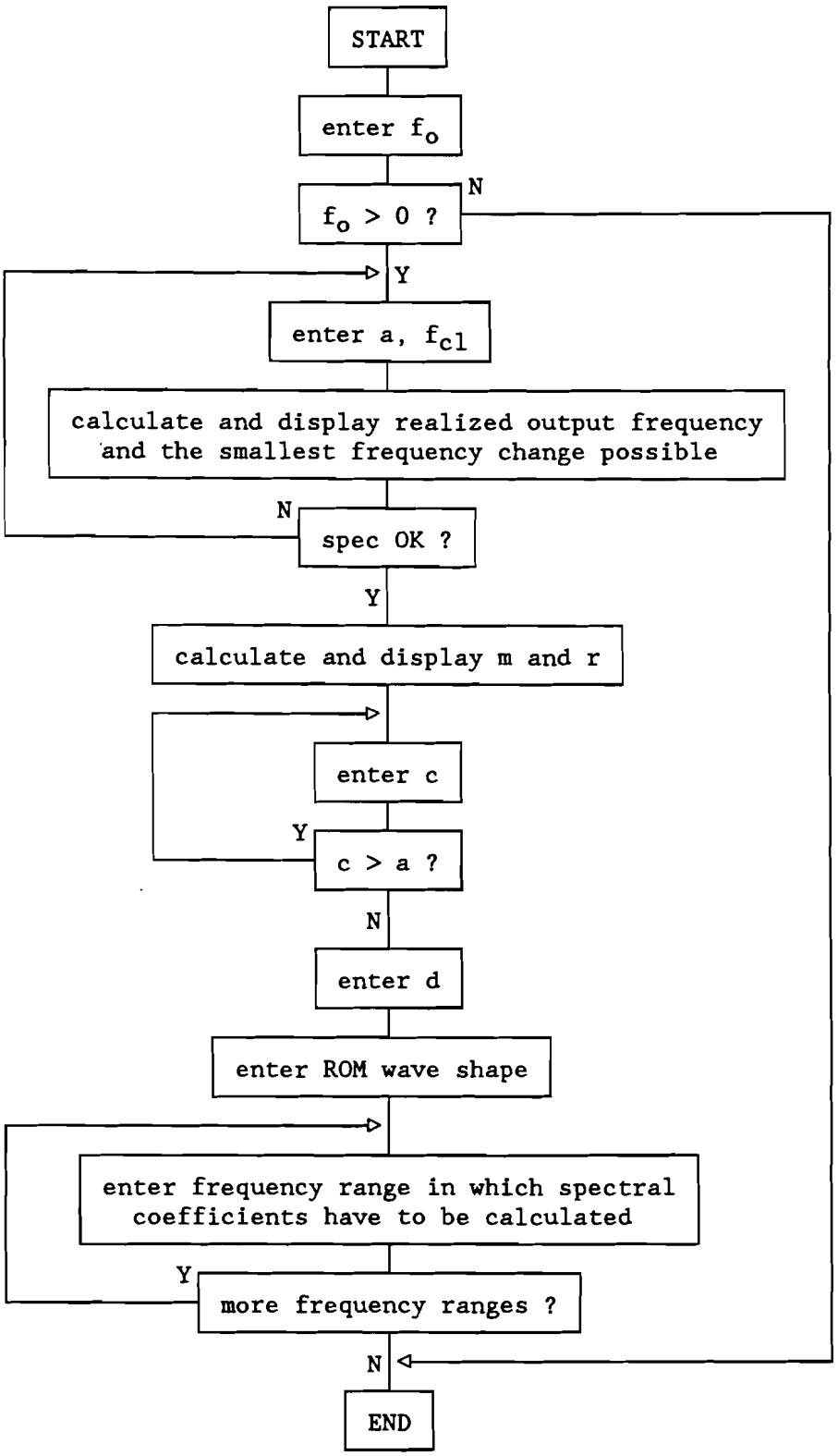


Fig.A3.1 Flow chart of the program 'SPECTRUMINVOER'.

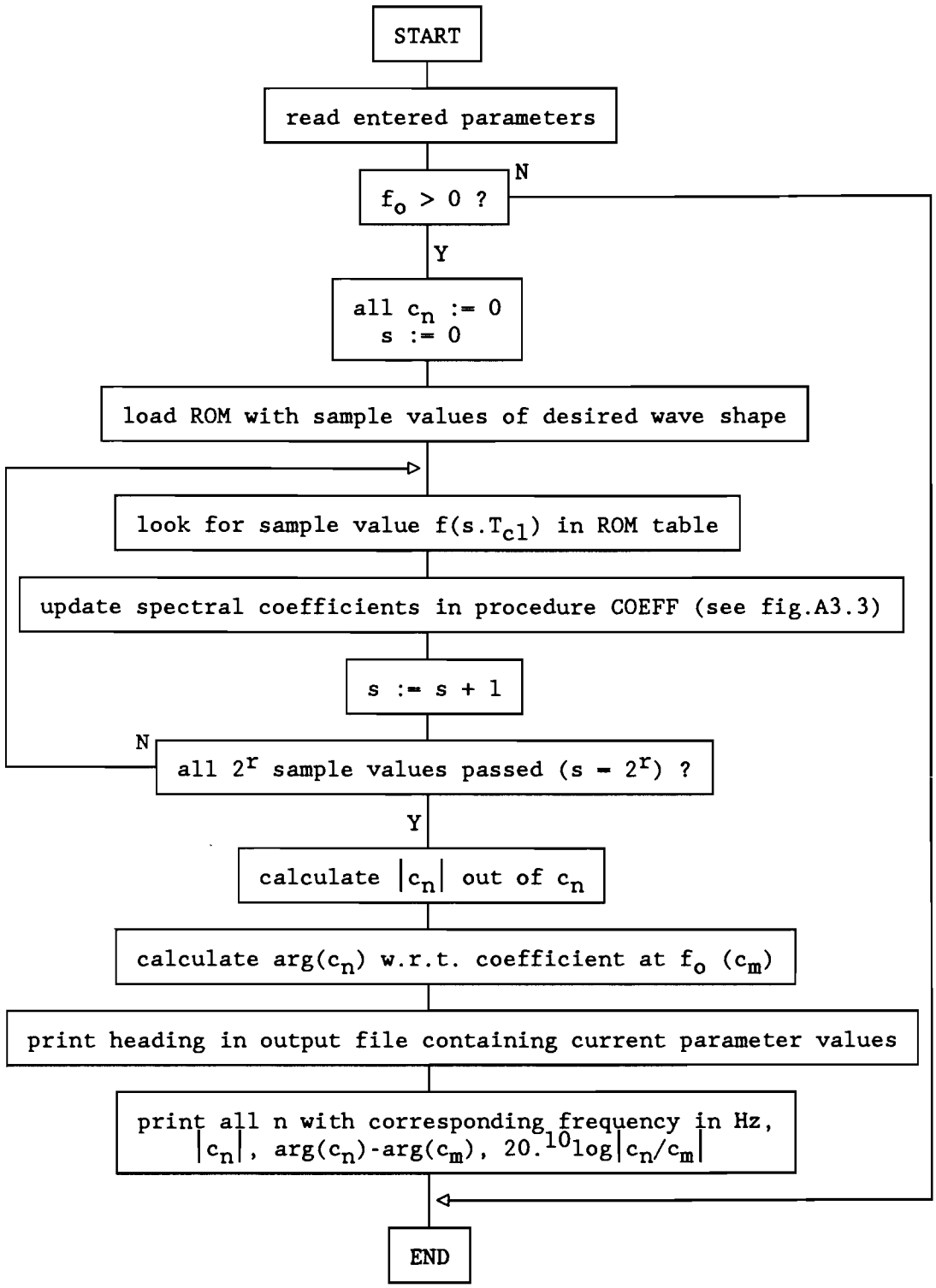


Fig.A3.2 Flow chart of the program 'SPECTRUM'.

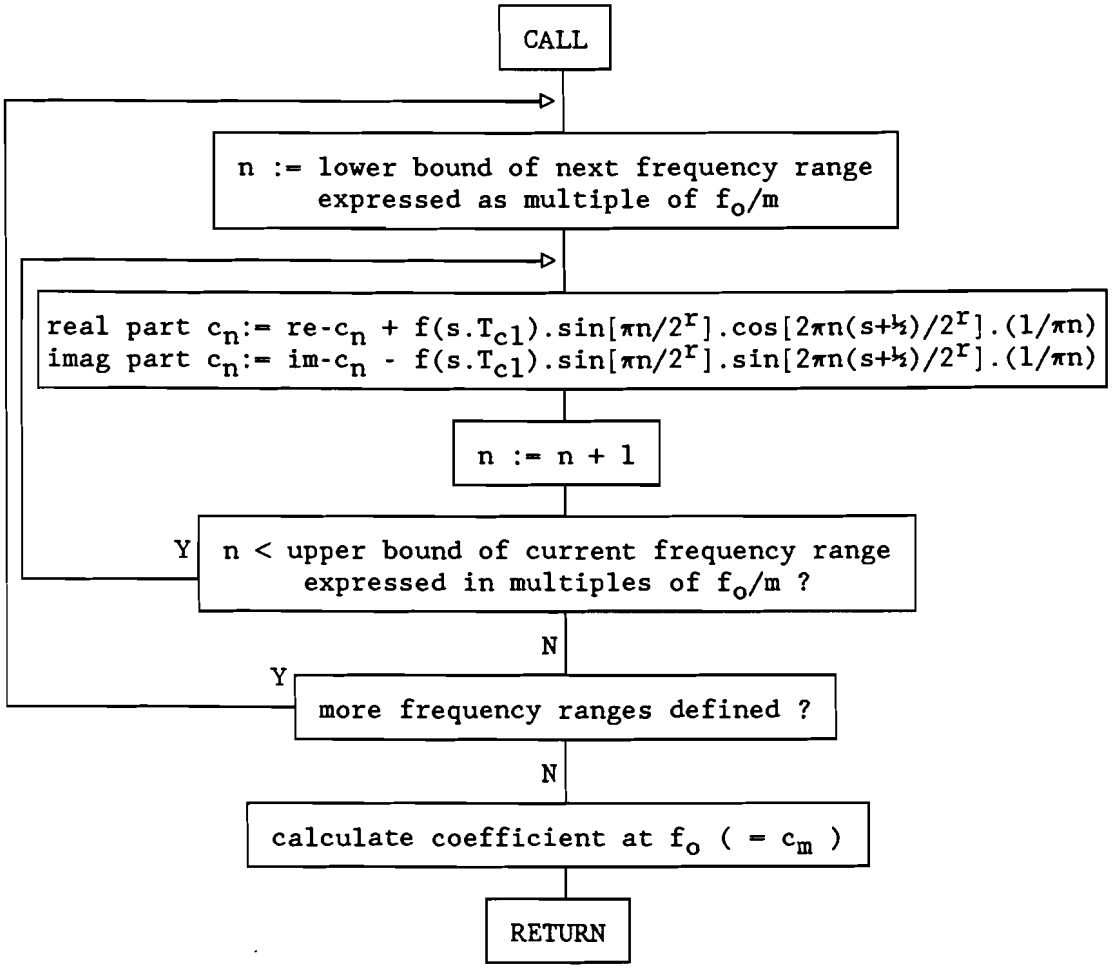


Fig.A3.3 Flow chart of procedure COEFF called in the program SPECTRUM (see figure A3.2).

Output example of the program SPECTRUM:

* The width of the DTO is 15 bits.
* The DTO clock frequency is: 24.576 MHz.
* The realized DTO output frequency is: 6.75 MHz.
* The ROM is loaded with a triangular wave.
* The address width of the ROM is 5 bits.
* The data width of the ROM is 4 bits.
* The desired output frequency is the 1125th harmonic of the DAC output signal.

Spectral coefficient of the nth harmonic of the DAC output signal:

n	freq.(MHz)	norm value	phase (deg.)	amplitude (dB)
1107	6.6420	5.468E-04	-18.0	-74.33
1108	6.6480	9.001E-06	-0.7	-110.00
1109	6.6540	2.061E-05	93.2	-102.80
1110	6.6600	6.843E-06	-1.6	-112.38
1111	6.6660	1.547E-04	47.5	-85.29
1112	6.6720	1.697E-05	7.8	-104.49
1113	6.6780	1.884E-03	-11.9	-63.58
1114	6.6840	1.202E-05	-10.3	-107.48
1115	6.6900	2.319E-03	109.6	-61.78
1116	6.6960	1.769E-05	8.4	-104.13
1117	6.7020	3.474E-04	47.4	-78.27
1118	6.7080	2.755E-05	2.1	-100.28
1119	6.7140	5.732E-04	-6.3	-73.92
1120	6.7200	4.033E-05	2.0	-96.97
1121	6.7260	2.920E-05	96.8	-99.78
1122	6.7320	8.708E-06	-20.0	-110.28
1123	6.7380	3.072E-04	57.6	-79.34
1124	6.7440	5.469E-06	-16.1	-114.33
1125	6.7500	2.845E+00	0.0	0.00
1126	6.7560	1.995E-05	-13.9	-103.09
1127	6.7620	4.716E-04	131.0	-75.61
1128	6.7680	1.048E-05	-4.0	-108.68
1129	6.7740	1.288E-04	71.0	-86.89
1130	6.7800	1.938E-05	37.1	-103.34
1131	6.7860	2.051E-04	4.5	-82.84
1132	6.7920	3.737E-06	53.1	-117.63
1133	6.7980	2.191E-04	126.9	-82.27
1134	6.8040	9.615E-06	-3.8	-109.42
1135	6.8100	7.542E-04	69.0	-71.53
1136	6.8160	6.895E-06	6.9	-112.31
1137	6.8220	3.246E-03	11.9	-58.86
1138	6.8280	9.365E-06	12.0	-109.65
1139	6.8340	2.560E-04	136.9	-80.92
1140	6.8400	8.972E-06	-9.1	-110.02
1141	6.8460	2.527E-05	-35.4	-101.03
1142	6.8520	9.168E-06	-1.7	-109.84
1143	6.8580	3.629E-04	17.3	-77.89

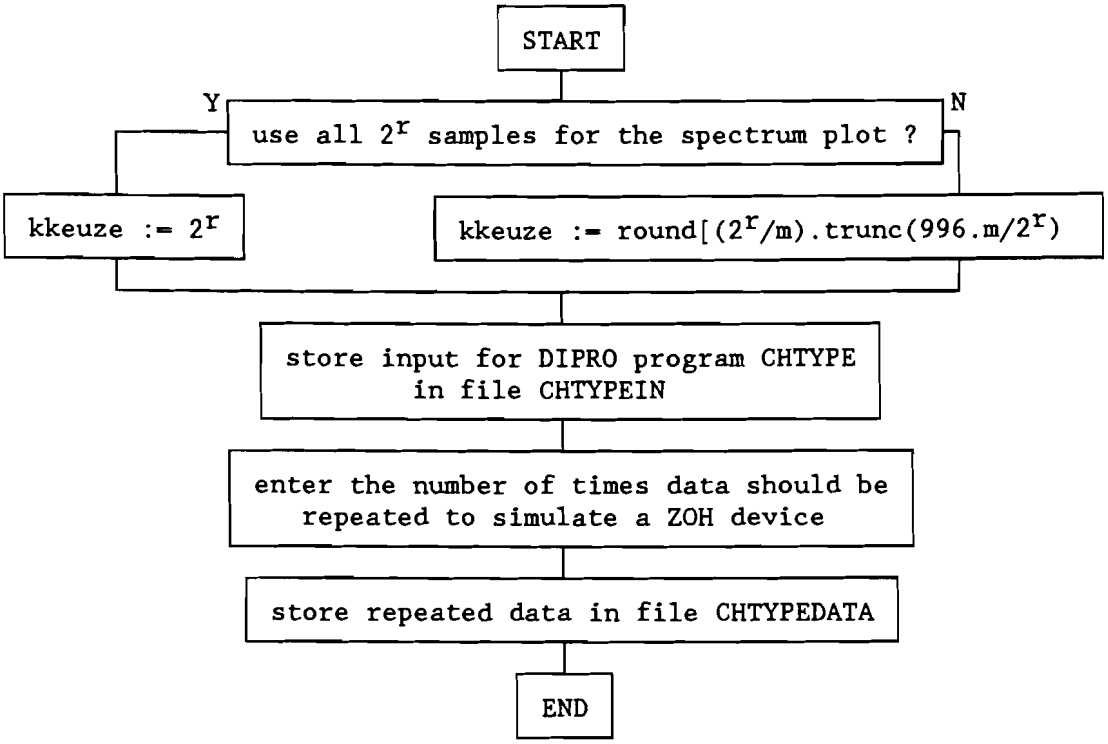


Fig.A3.4 Flow chart of the program 'DIPROPLOTINVOER'.

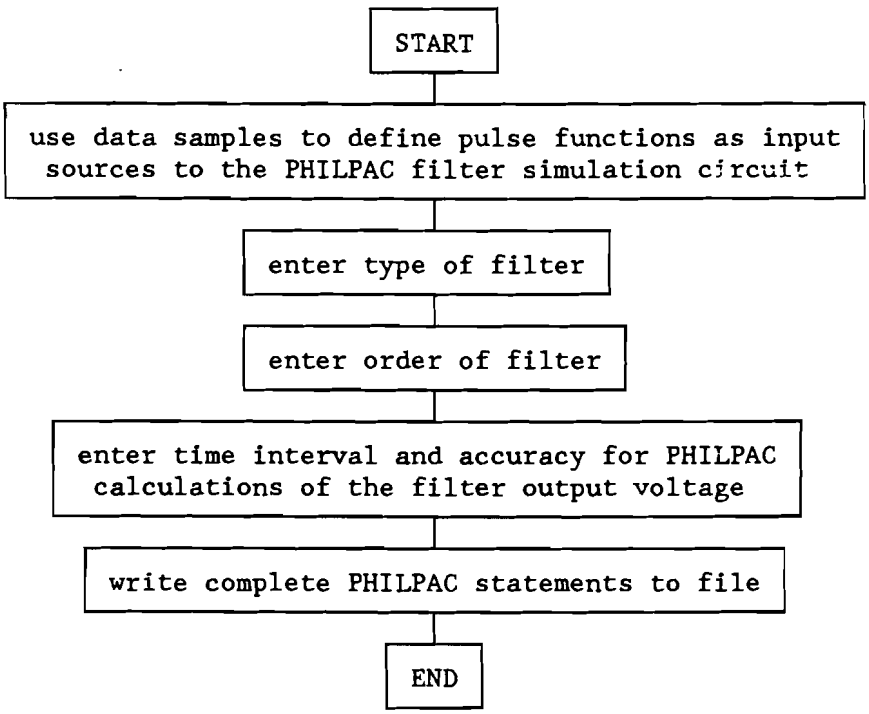


Fig.A3.5 Flow chart of the program 'PHILPACINVOER'.

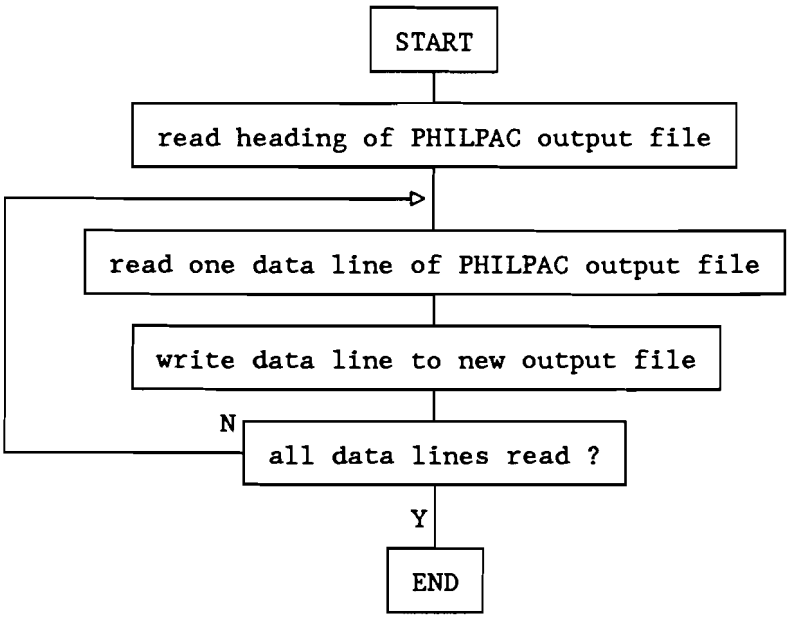


Fig.A3.6 Flow chart of the program 'DELETEHEADER'.

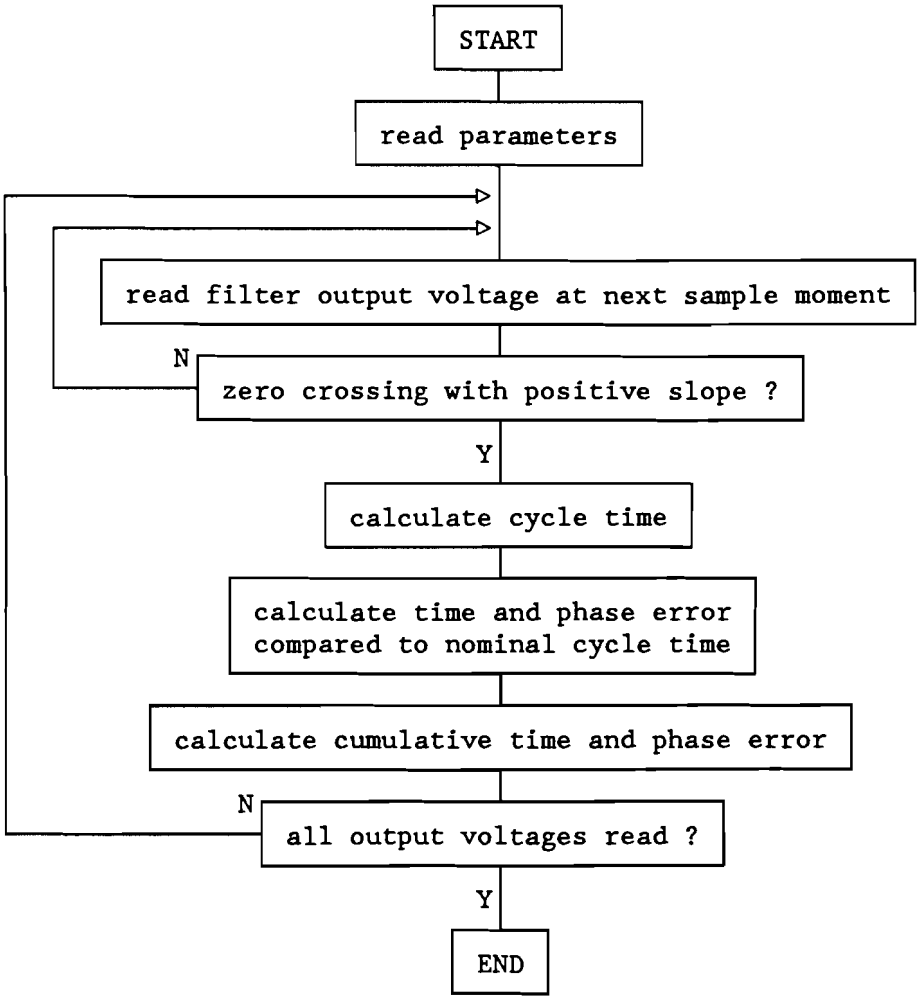


Fig.A3.7 Flow chart of the program 'PERIODETIJDEN'.

Output example of the program PERIODETIJDEN

Circuit parameters: output frequency f_o : 6.75000E+00 MHz
crystal clock freq. f_{cl}: 2.45760E+01 MHz
DTO width (a): 14 bits
ROM address width (c): 5 bits
ROM data width (d): 4 bits
ROM stored wave : triangular
Kind of filter : Chebychev 0.5 dB ripple
Order of filter : 3

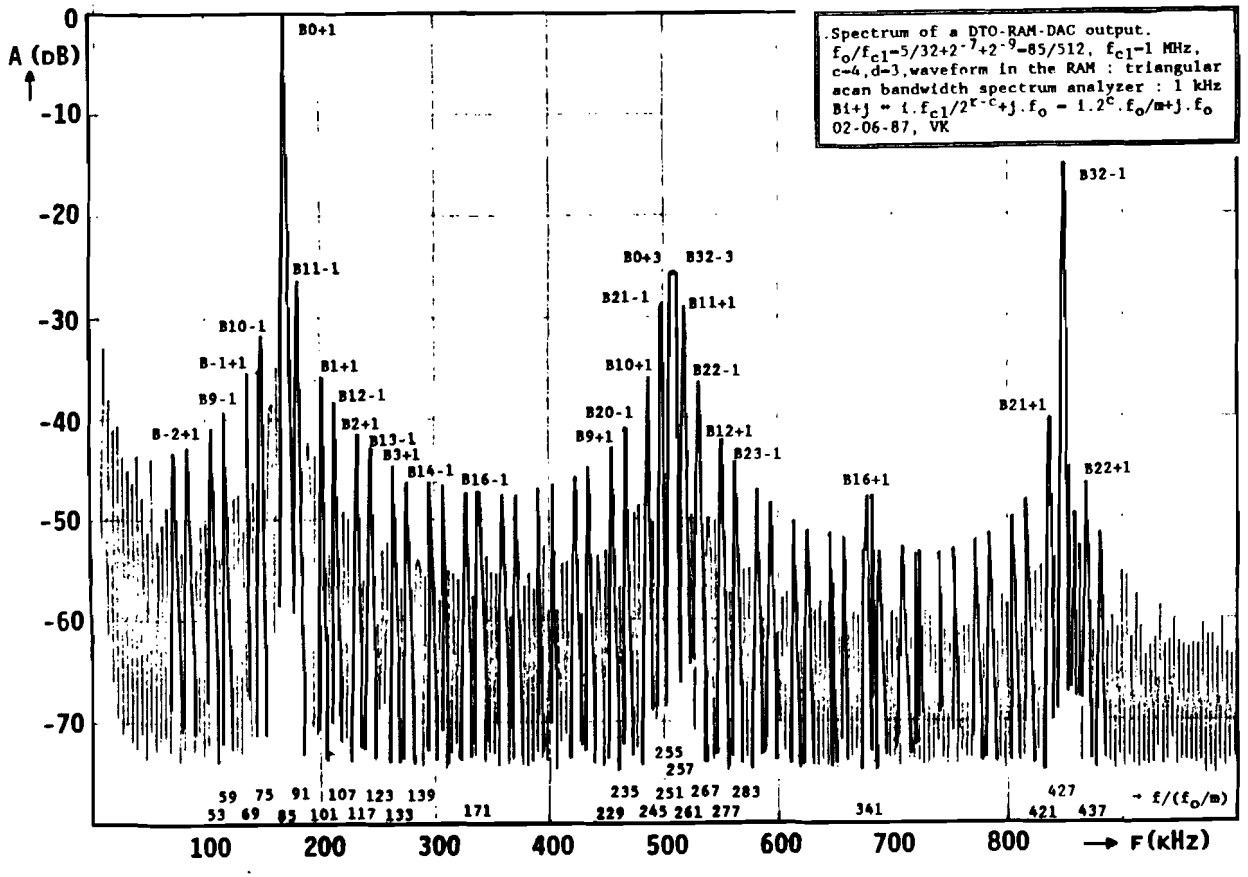
t/T_o	cycle time	time error	phase error	cumulative time error	cumulative phase error
8	1.48051E-07	-9.69607E-11	-0.2	-9.69607E-11	-0.2
9	1.48297E-07	1.48603E-10	0.4	5.16422E-11	0.1
10	1.47764E-07	-3.84361E-10	-0.9	-3.32719E-10	-0.8
11	1.48695E-07	5.46962E-10	1.3	2.14243E-10	0.5
12	1.48191E-07	4.31015E-11	0.1	2.57344E-10	0.6
13	1.47133E-07	-1.01555E-09	-2.5	-7.58206E-10	-1.8
14	1.48364E-07	2.15906E-10	0.5	-5.42300E-10	-1.3
15	1.48575E-07	4.26908E-10	1.0	-1.15392E-10	-0.3
16	1.48282E-07	1.34051E-10	0.3	1.86589E-11	0.0
17	1.48699E-07	5.50600E-10	1.3	5.69258E-10	1.4
114	1.47942E-07	-2.06100E-10	-0.5	-3.63073E-10	-0.9
115	1.48382E-07	2.34095E-10	0.6	-1.28978E-10	-0.3
116	1.48233E-07	8.49383E-11	0.2	-4.40394E-11	-0.1
117	1.48324E-07	1.75888E-10	0.4	1.31848E-10	0.3
118	1.47917E-07	-2.31566E-10	-0.6	-9.97176E-11	-0.2
119	1.47877E-07	-2.71584E-10	-0.7	-3.71301E-10	-0.9
120	1.48848E-07	6.99757E-10	1.7	3.28455E-10	0.8
121	1.47913E-07	-2.35204E-10	-0.6	9.32516E-11	0.2
122	1.48026E-07	-1.22427E-10	-0.3	-2.91749E-11	-0.1
123	1.48557E-07	4.08718E-10	1.0	3.79544E-10	0.9
124	1.47458E-07	-6.89951E-10	-1.7	-3.10408E-10	-0.8

* The nominal cycle time is : 1.48148E-07 seconds.
* The filter input signal is periodic after : 1.66667E-04 seconds
: 1125 output periods
: 4096 clock periods.
* Because the "Philpac" package can only handle
996 pulses periodicity is assumed after : 2.08333E-05 seconds
: 273.01 output periods
: 994 clock periods.

Calculations of the cycle times have started after 500 clock periods.
Calculations have then been performed for 447 clock periods.
The value of the output signal is calculated every 1 nanosecond.
Zero crossings are then determined using linear interpolation.

The largest time error occurring in one period is :
too large : 1.24E-09 seconds.
too small : -1.22E-09 seconds.

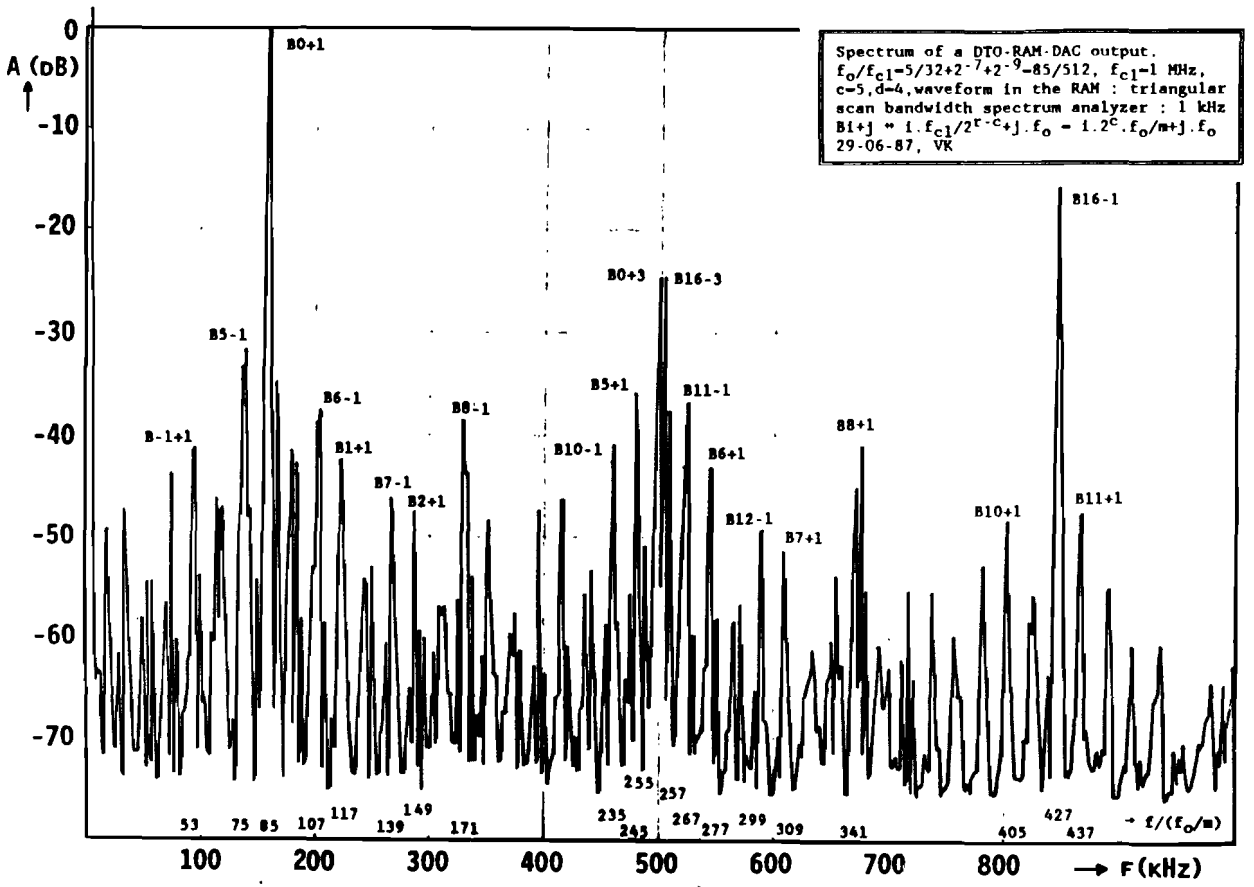
APPENDIX 4 : EXAMPLES OF MEASURED SPECTRUM AND SIMULATION OUTPUT



* The width of the DTO is 9 bits.
* The DTO clockfrequency is: 512 MHz.
* The realized DTO output frequency is: 85 MHz.
* The ROM is loaded with a triangular wave.
* The address width of the ROM is 4 bits.
* The data width of the ROM is 3 bits.
* The desired output frequency is the 85th harmonic of the time discrete signal.

The spectral coefficient of the nth harmonic of the time discrete signal is:

n	freq.(MHz)	norm value	phase(deg.)	ampl.(dB)
67	67	1.142E-03	25.2	-62.53
68	68	1.003E-06	1.8	-123.66
69	69	3.217E-02	22.5	-33.54
70	70	6.236E-07	83.4	-127.79
71	71	4.848E-03	19.7	-49.98
72	72	4.708E-07	23.6	-130.23
73	73	5.606E-03	16.9	-48.71
74	74	1.610E-07	17.0	-139.55
75	75	5.012E-02	14.1	-29.69
76	76	3.356E-07	17.0	-133.17
77	77	2.450E-03	11.2	-55.90
78	78	1.747E-06	5.0	-118.84
79	79	2.416E-02	8.4	-36.03
80	80	1.470E-06	-4.8	-120.34
81	81	3.519E-02	5.6	-32.76
82	82	1.089E-06	17.2	-122.95
83	83	8.660E-03	2.8	-44.94
84	84	7.542E-06	5.5	-106.14
85	85	1.529E+00	0.0	0.00
86	86	1.111E-06	-34.3	-122.78
87	87	4.056E-02	-2.8	-31.53
88	88	4.293E-06	2.3	-111.03
89	89	1.840E-02	-5.6	-38.39
90	90	7.254E-07	93.1	-126.47
91	91	8.948E-02	-8.4	-24.65
92	92	8.171E-07	-15.8	-125.44
93	93	2.615E-03	-11.3	-55.34
94	94	3.396E-06	63.7	-113.07
95	95	1.559E-02	-14.1	-39.83
96	96	9.747E-07	-1.5	-123.91
97	97	1.292E-02	-16.9	-41.46
98	98	2.376E-07	38.5	-136.17
99	99	1.467E-03	-19.7	-60.36
100	100	6.182E-06	74.0	-107.86
101	101	3.236E-02	-22.5	-33.49
102	102	2.623E-06	53.9	-115.31
103	103	3.818E-03	-25.3	-52.05

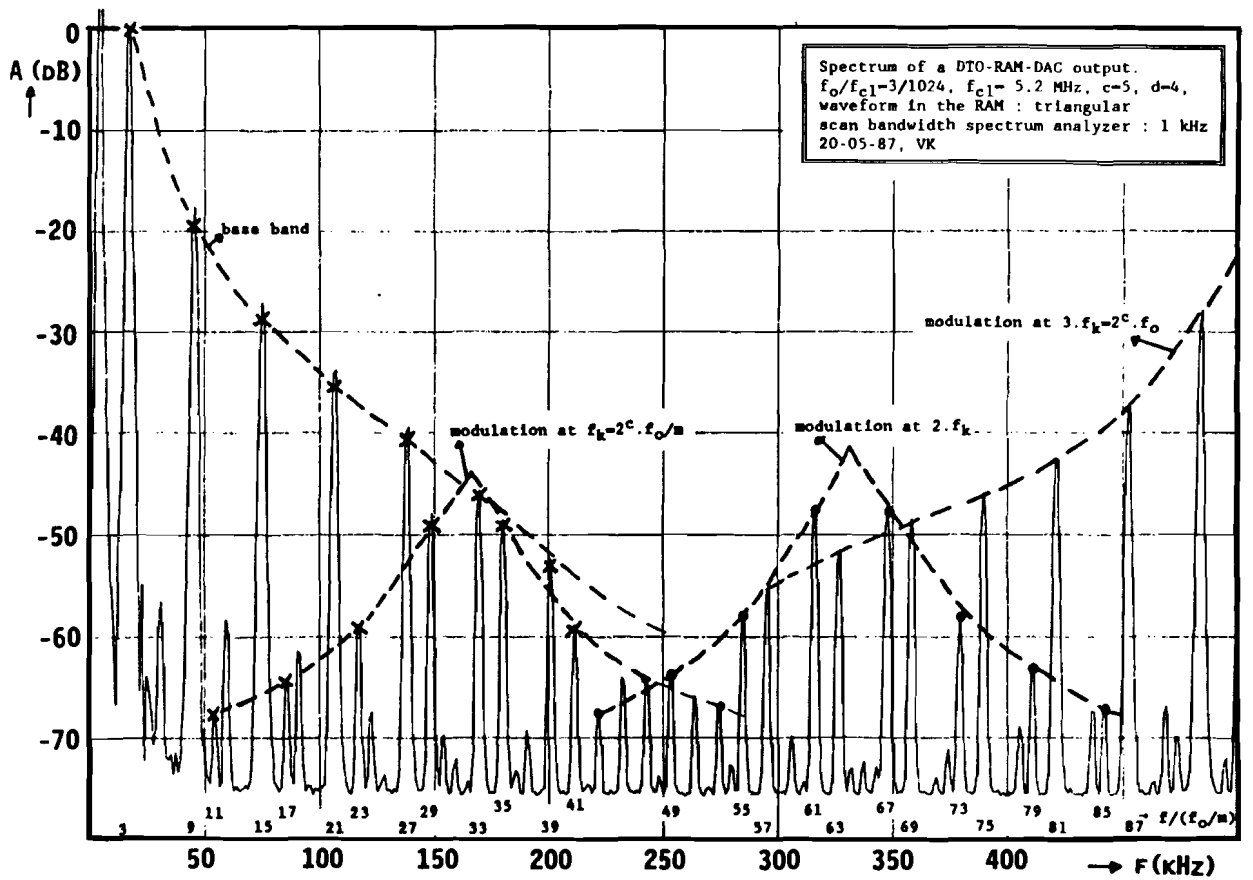


* The width of the DTO is 9 bits.
* The DTO clockfrequency is: 512 MHz.
* The realized DTO output frequency is: 85 MHz.
* The ROM is loaded with a triangular wave.
* The address width of the ROM is 5 bits.
* The data width of the ROM is 4 bits.
* The desired output frequency is the 85$^{\underline{th}}$ harmonic of the time discrete signal.

Spectral coefficient of the n$^{\underline{th}}$ harmonic of the time discrete signal:

n	freq.(MHz)	norm value	phase(deg.)	ampl.(dB)
67	67	4.702E-03	25.3	-56.35
68	68	1.890E-06	2.6	-124.26
69	69	6.306E-04	22.5	-73.80
70	70	1.155E-06	107.3	-128.54
71	71	3.877E-03	19.7	-58.02
72	72	1.003E-06	19.2	-129.76
73	73	1.570E-02	16.9	-45.88
74	74	3.072E-07	3.0	-140.04
75	75	1.012E-01	14.1	-29.69
76	76	7.362E-07	27.5	-132.45
77	77	1.500E-02	11.2	-46.27
78	78	3.164E-06	9.2	-119.79
79	79	4.897E-03	8.4	-55.99
80	80	3.519E-06	-2.4	-118.86
81	81	7.131E-03	5.6	-52.73
82	82	2.234E-06	21.0	-122.81
83	83	5.306E-02	2.8	-35.30
84	84	1.468E-05	7.5	-106.46
85	85	3.087E+00	0.0	0.00
86	86	2.027E-06	-49.0	-123.66
87	87	1.136E-01	-2.8	-28.69
88	88	9.218E-06	1.4	-110.50
89	89	1.472E-02	-5.6	-46.43
90	90	1.869E-06	115.6	-124.36
91	91	1.749E-03	-8.3	-64.93
92	92	1.827E-06	-18.4	-124.56
93	93	1.079E-02	-11.2	-49.13
94	94	7.238E-06	66.8	-112.60
95	95	3.435E-02	-14.1	-39.07
96	96	2.130E-06	-4.5	-123.22
97	97	2.846E-02	-16.9	-40.71
98	98	4.996E-07	60.5	-135.82
99	99	6.050E-03	-19.7	-54.16
100	100	1.309E-05	73.4	-107.45
101	101	6.341E-04	-22.4	-73.75
102	102	5.364E-06	61.2	-115.20
103	103	3.055E-03	-25.3	-60.09

APPENDIX 5 : EXAMPLE OF MEASURED SPECTRUM AND SIMULATION OUTPUT



* The width of the DTO is 24 bits.
* The DTO clockfrequency is: 1024 MHz.
* The realized DTO output frequency is: 3.0 MHz.
* The ROM is loaded with a triangular wave.
* The address width of the ROM is 5 bits.
* The data width of the ROM is 4 bits.
* The desired output frequency is the 3rd harmonic of the time discrete signal.

The spectral coefficient of the n^{th} harmonic of the time discrete signal is:

n	freq.(MHz)	norm value	phase(deg.)	ampl.(dB)
1	1	6.032E-04	-59.8	-74.58
2	2	5.707E-07	87.8	-135.06
3	3	3.232E+00	0.0	0.00
4	4	5.499E-07	49.4	-135.38
5	5	9.224E-04	59.7	-70.89
6	6	2.630E-07	2.0	-141.79
7	7	3.437E-04	-60.5	-79.47
8	8	1.863E-07	-23.1	-144.78
9	9	3.497E-01	-0.7	-19.31
10	10	1.334E-07	30.7	-147.68
11	11	1.365E-03	59.1	-67.49
12	12	1.838E-07	46.3	-144.90
13	13	1.121E-04	-61.1	-89.19
14	14	1.863E-07	34.2	-144.78
15	15	1.191E-01	-1.4	-28.67
16	16	1.538E-07	73.8	-146.45
17	17	2.088E-03	58.4	-63.79
18	18	1.992E-07	17.5	-144.20
19	19	1.122E-04	-61.8	-89.19
20	20	1.147E-07	43.3	-149.00
21	21	5.537E-02	-2.1	-35.32
22	22	1.518E-07	-87.3	-146.57
23	23	3.666E-03	57.7	-58.91
24	24	1.802E-06	14.8	-125.07
25	25	3.471E-04	-62.6	-79.38
26	26	2.653E-07	42.8	-141.72
27	27	2.900E-02	-2.8	-40.94
28	28	9.454E-08	-31.1	-150.68
29	29	1.125E-02	57.0	-49.17
30	30	2.108E-07	4.4	-143.71
31	31	6.133E-04	-63.3	-74.44
32	32	3.361E-07	66.1	-139.66
33	33	1.544E-02	-3.5	-46.41
34	34	6.410E-08	19.9	-154.05
35	35	1.120E-02	56.3	-49.20
36	36	4.509E-07	-56.6	-137.11
37	37	9.447E-04	-64.0	-70.68

APPENDIX 6 : SPECTRUM OF A CLIPPED TRIANGULAR WAVE

In § 5.1 the clipped triangular wave has been introduced, because its spectrum might contain less harmonics than a common triangular wave. In this appendix the spectrum of a clipped triangular wave is derived in dependance of the clipping level μ. This parameter is (as in fig.5.1) defined in figure A6.1.

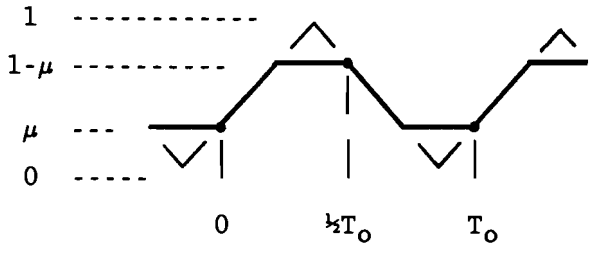


Fig.A6.1 Definition of a clipped triangular wave with clipping level μ.

To calculate the spectrum of the signal first the time function is defined according to figure A6.1:

$$f(t) = \begin{cases} (2/T_o).t + \mu & 0 < t < T_o.(\tfrac{1}{2}-\mu)=t_1 \\ 1 - \mu & T_o.(\tfrac{1}{2}-\mu) < t < \tfrac{1}{2}.T_o \\ 2 - \mu - (2/T_o).t & \tfrac{1}{2}.T_o < t < T_o.(1-\mu)=t_2 \\ \mu & T_o.(1-\mu) < t < T_o \end{cases}$$

Using formula (3.5) the complex spectral coefficients c_n are given by:

$$T_o.c_n = \int_0^{t_1} (\frac{2.t}{T_o} + \mu).e^{-2\pi jnt/T_o}\,dt + (1-\mu)\int_{t_1}^{\frac{1}{2}T_o} e^{-2\pi jnt/T_o}\,dt +$$

$$\int_{\frac{1}{2}T_o}^{t_2} (2 - \mu - \frac{2.t}{T_o}).e^{-2\pi jnt/T_o}\,dt + \mu\int_{t_2}^{T_o} e^{-2\pi jnt/T_o}\,dt$$

$$= I_1 + I_2 + I_3 + I_4$$

$$I_1 = \frac{-2}{2\pi jn}.T_o.(\tfrac{1}{2}-\mu).e^{-2\pi jn(\frac{1}{2}-\mu)} + \frac{T_o}{2\pi^2n^2}.[e^{-2\pi jn(\frac{1}{2}-\mu)}-1] - \frac{\mu T_o}{2\pi jn}.[e^{-2\pi jn(\frac{1}{2}-\mu)}-1]$$

$$\frac{I_1}{T_o} = \frac{-2}{2\pi jn}.\{\ (\tfrac{1}{2}-\mu).e^{-2\pi jn(\frac{1}{2}-\mu)} - \tfrac{1}{2}\mu + \tfrac{1}{2}\mu e^{-2\pi jn(\frac{1}{2}-\mu)}\ \} - \frac{1}{2\pi^2n^2}.[e^{-2\pi jn(\frac{1}{2}-\mu)}-1]$$

$$\frac{I_1}{T_o} = \frac{1}{2\pi jn}.\{\ \mu - (1-\mu).e^{-2\pi jn(\frac{1}{2}-\mu)}\ \} + \frac{1}{2\pi^2n^2}.[e^{-2\pi jn(\frac{1}{2}-\mu)}-1]$$

$$\frac{I_2}{T_o} = \frac{\mu-1}{2\pi jn}.\{\ e^{-\pi jn} - e^{-2\pi jn(½-\mu)}\ \}$$

$$\frac{I_3}{T_o} = \frac{\mu-2}{2\pi jn}.\{\ e^{-2\pi jn(1-\mu)} - e^{-\pi jn}\ \} + \frac{2}{2\pi jn}.\{\ (1-\mu).e^{-2\pi jn(1-\mu)} - ½.e^{-\pi jn}\ \} +$$

$$\frac{1}{2\pi^2n^2}.\{\ e^{-2\pi jn(1-\mu)} - e^{-\pi jn}\ \}$$

$$\frac{I_4}{T_o} = \frac{\mu}{2\pi jn}.\{\ e^{-2\pi jn(1-\mu)} - 1\ \}$$

Combining all formulas yields for c_n:

$$c_n = \frac{1}{2\pi^2n^2}.\{\ e^{-2\pi jn(½-\mu)} - 1 - e^{-2\pi jn(1-\mu)} + e^{-\pi jn}\ \}$$

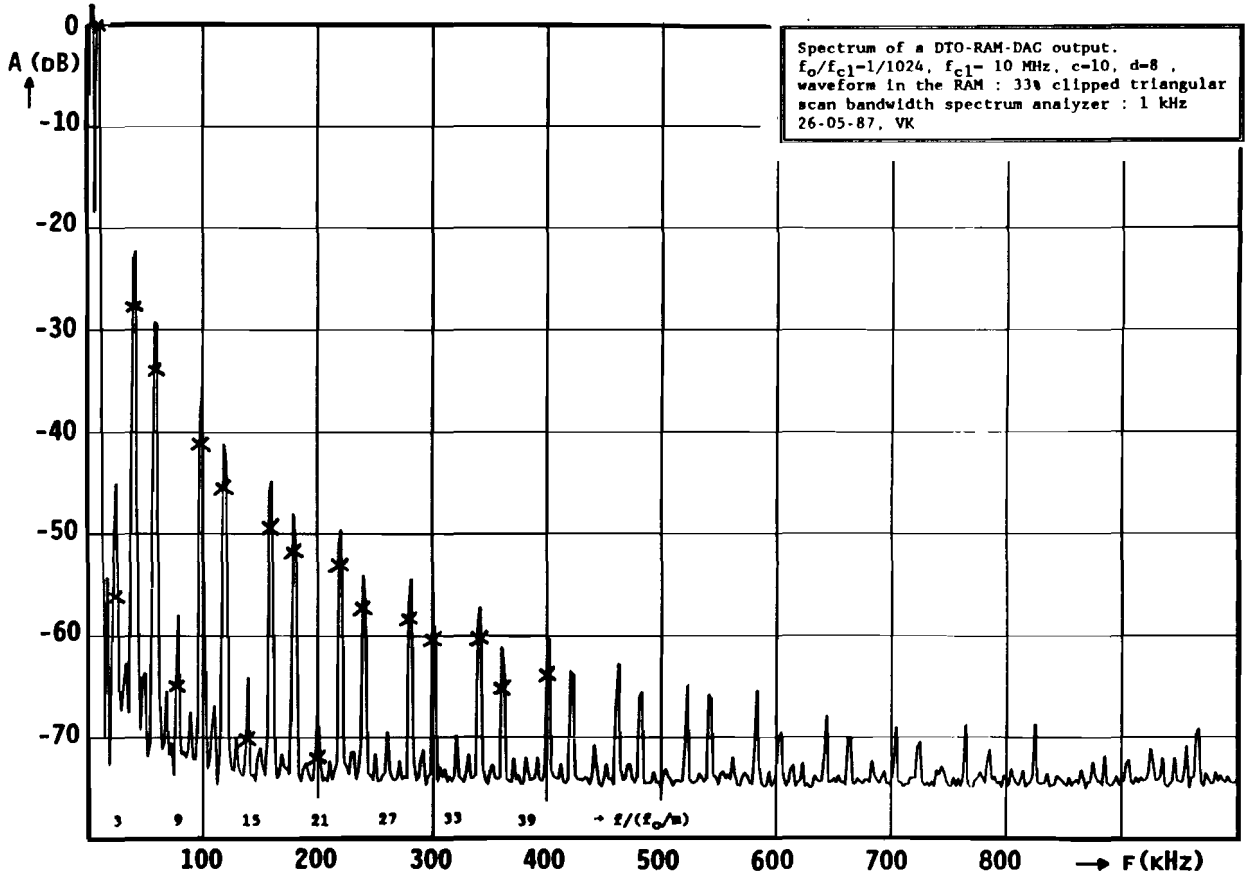
$$c_n = \frac{1}{2\pi^2n^2}.\{\ e^{2\pi jn\mu}.[\ (-1)^n - 1\ ] + (-1)^n - 1\ \}$$

When n is even $(-1)^n = 1$, when n is odd $(-1)^n = -1$. Therefore

n even: $c_n = 0$

n odd : $c_n = \frac{-1}{\pi^2n^2}.\{\ e^{2\pi jn\mu} + 1\ \}$

APPENDIX 7 : EXAMPLE OF MEASURED SPECTRUM AND SIMULATION OUTPUT



* The width of the DTO is 10 bits.
* The DTO clockfrequency is: 1024 MHz.
* The realized DTO output frequency is: 1.0 MHz.
* The ROM is loaded with a 33% clipped triangular wave.
* The address width of the ROM is 10 bits.
* The data width of the ROM is 8 bits.
* The desired output frequency is the $1^{\underline{st}}$ harmonic of the time discrete signal.

The spectral coefficient of the $n^{\underline{th}}$ harmonic of the time discrete signal is:

n	freq.(MHz)	norm value	phase(deg.)	ampl.(dB)
1	1	6.697E+01	0.0	0.00
2	2	1.054E-05	46.2	-136.06
3	3	1.058E-01	-119.7	-56.03
4	4	3.729E-06	-119.8	-145.09
5	5	2.723E+00	-59.3	-27.82
6	6	4.699E-06	-96.8	-143.08
7	7	1.355E+00	1.1	-33.88
8	8	2.202E-06	-8.5	-149.66
9	9	3.569E-02	-118.6	-65.47
10	10	2.728E-06	-85.6	-147.80
11	11	5.940E-01	-58.3	-41.04
12	12	8.849E-07	42.3	-157.58
13	13	3.660E-01	2.2	-45.25
14	14	3.282E-07	0.2	-166.20
15	15	2.089E-02	-117.6	-70.12
16	16	2.395E-06	6.3	-148.93
17	17	2.410E-01	-57.5	-48.88
18	18	2.587E-06	-39.2	-148.26
19	19	1.789E-01	3.7	-51.47
20	20	1.828E-06	-34.5	-151.28
21	21	1.548E-02	-116.4	-72.72
22	22	2.988E-06	-49.1	-147.01
23	23	1.447E-01	-56.3	-53.31
24	24	3.496E-06	-51.1	-145.65
25	25	9.040E-02	4.5	-57.39
26	26	5.058E-06	10.3	-142.44
27	27	1.145E-02	-115.6	-75.34
28	28	4.944E-07	-10.0	-162.64
29	29	8.447E-02	-56.0	-57.98
30	30	1.821E-06	1.9	-151.31
31	31	6.528E-02	6.8	-60.22
32	32	1.599E-05	-47.4	-132.44
33	33	9.961E-03	-114.0	-76.55
34	34	8.110E-06	-38.3	-138.34
35	35	6.617E-02	-54.3	-60.10
36	36	1.534E-06	-32.4	-152.80
37	37	3.720E-02	7.1	-65.11

APPENDIX 8 : SPECTRUM OF A QUANTIZED SINE WAVE

In par. 5.2.2 the effects of amplitude quantization on the spectrum of a sine wave have been investigated, using formula (5.5). This formula describes the spectrum of an analogue sine wave, quantized in d bits. In this appendix this formula is derived. Furthermore the flow charts of a computer program are given which calculates amplitude values of harmonics. These are added in some tables. A graphic representation of the spectrum has already been given in figure 5.5.

An analogue quantized sine wave will be defined according to fig.A8.1.

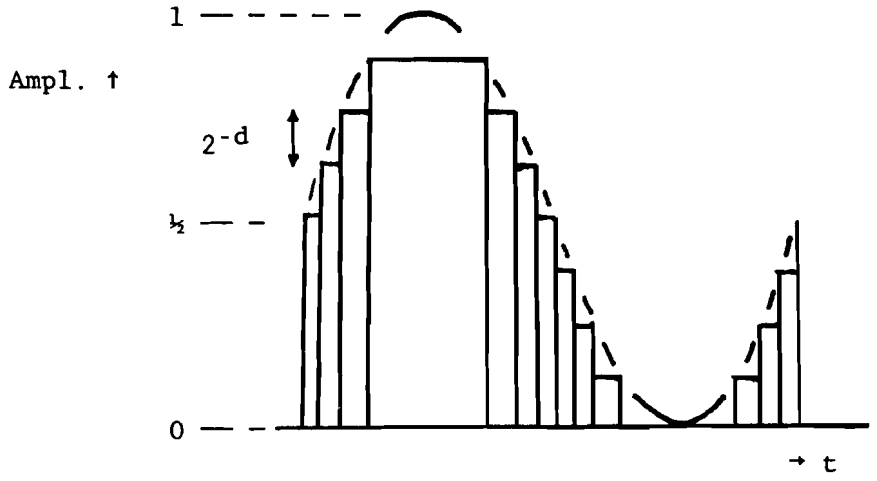


Fig.A8.1 Definition of an analogue quantized sine wave as a pulse train; the width of the pulses varies with the slope of the sine wave, all amplitude values are multiples of 2^{-d}.

The sine wave is assumed to have periodicity T_o. This yields for its frequency $1/T_o = f_o$. All pulses have amplitude A_i, start time t_{i-1} and end time t_i, see figure A8.2.

Fig.A8.2 General definition of a pulse, used to define a quantized sine wave.

To define all time values in terms of arc sine values, the wave is divided in three parts, because the arc sine is only defined in the range $[-\frac{1}{2}\pi, \frac{1}{2}\pi]$.

Part I: $0 < \phi < \pi/2$

t_i: $\frac{1}{2} + \frac{1}{2}.\sin(2\pi f_o t_i) = \frac{1}{2} + i.2^{-d}$
$\Rightarrow$ $t_i = (1/2\pi f_o).\arcsin(i.2^{1-d})$
i : last pulse ends when $t_i = T_o/4$
$\Rightarrow$ $i = 1,2,..,2^{d-1}$
A_i: $A_i = \frac{1}{2} + (i-1).2^{-d}$

Part II: $\pi/2 < \phi < 3\pi/2$

t_i: $t_i - \frac{1}{4}T_o = \frac{1}{4}T_o - \tau$, $\frac{1}{2} + \frac{1}{2}.\sin(2\pi f_o \tau) = 1 - i.2^{-d}$
$\Rightarrow$ $\tau = (1/2\pi f_o).\arcsin(1-i.2^{1-d})$
$\Rightarrow$ $t_i = \frac{1}{2}T_o.[\ 1 - (1/\pi).\arcsin(1-i.2^{1-d})\ ]$
i : last pulse ends when $t_i = 3T_o/4$
$\Rightarrow$ $i = 1,2,..,2^{d}$
A_i: $A_i = 1 - i.2^{-d}$

Part III: $3\pi/2 < \phi < 2\pi$

t_i: $t_i - T_o = \tau$, $\frac{1}{2} + \frac{1}{2}.\sin(2\pi f_o \tau) = i.2^{-d}$
$\Rightarrow$ $\tau = (1/2\pi f_o).\arcsin(i.2^{1-d}-1)$
$\Rightarrow$ $t_i = T_o.[\ 1 + (1/2\pi).\arcsin(i.2^{1-d}-1)\ ]$
i : last pulse ends when $t_i = T_o$
$\Rightarrow$ $i = 1,2,..,2^{d-1}$
A_i: $A_i = (i-1).2^{-d}$

From this definition of the function in the time domain, the complex Fourier coefficients can be found. For each pulse function a Fourier series can be defined. For this case the general formula (3.5) holds:

$$c_n = f_o.\int_{t_{i-1}}^{t_i} A_i.e^{-2\pi jnf_o t}\,dt = \frac{-A_i}{2\pi jn}.\{\ e^{-2\pi jnf_o t_i} - e^{-2\pi jnf_o t_{i-1}}\ \}$$

For the entire function f(t) it holds:

$$f(t) = f_I(t) + f_{II}(t) + f_{III}(t)$$

where the subscripts conform with the part numbers defined before. Combining all formulas so far yields:

$$f_I(t) : c_{I,n} = \sum_{i=1}^{2^{d-1}} \frac{\frac{1}{2} + (i-1).2^{-d}}{-2\pi jn}.\{e^{-jn.\arcsin(i2^{1-d})} - e^{-jn.\arcsin((i-1).2^{1-d})}\}$$

$$f_{II}(t) : c_{II,n} = \sum_{i=1}^{2^{d}} \frac{1-i.2^{-d}}{-2\pi jn}.\{e^{-jn(\pi-\arcsin(1-i2^{1-d})} - e^{-jn(\pi-\arcsin(1-(i-1)2^{1-d})}\}$$

$$f_{III}(t) : c_{III,n} = \sum_{i=1}^{2^{d-1}} \frac{(i-1).2^{-d}}{-2\pi jn}.\{e^{-jn.\arcsin(i2^{1-d}-1)} - e^{-jn.\arcsin((i-1).2^{1-d}-1)}\}$$

As can be seen from figure A8.1, apart from a DC term which only influences c_o the quantized sine wave is an odd function like the continuous sine wave. Therefore all coefficients c_n will be pure imaginary.

Furthermore, for the odd function it holds that $f(x) = -f(x+\pi)$. Therefore only odd harmonics will appear in the spectrum:

This follows from the Fourier series of each function f(x):

$$f(x) = \Sigma\, c_n.e^{jnx}$$

Therefore it holds:

$$\begin{aligned} f(x) + f(x+\pi) &= \Sigma\, c_n.e^{jnx} + \Sigma\, c_n.e^{jn(x+\pi)} \\ &= \Sigma\, c_n.e^{jnx} + \Sigma\, (-1)^n.c_n.e^{jnx} \\ &= c_o + c_{2n}.e^{j2nx} + c_{4n}.e^{j4nx} + \ldots \end{aligned}$$

When $f(x) + f(x+\pi) = 0$ for each x all terms of the right member of the equation above should be zero. Therefore no even harmonics will appear in the spectrum.

So, to obtain an expression for the coefficients c_n of the entire function only the imaginary parts of the coefficients of the subfunctions for odd n should be added. Therefore:

n even:

$$c_n = 0$$

n odd:

$$\begin{aligned} c_n = &\sum_{i=1}^{2^{d-1}} \frac{\tfrac{1}{2} + (i-1).2^{-d}}{-2\pi jn}.\{\cos[n.\arcsin(i2^{1-d})] - \cos[n.\arcsin((i-1).2^{1-d})]\} + \\ &\sum_{i=1}^{2^{d}} \frac{i.2^{-d}-1}{-2\pi jn}.\{\cos[n.\arcsin(1-i2^{1-d})] - \cos[n.\arcsin(1-(i-1)2^{1-d})]\} + \\ &\sum_{i=1}^{2^{d-1}} \frac{(i-1).2^{-d}}{-2\pi jn}.\{\cos[n.\arcsin(i2^{1-d}-1)] - \cos[n.\arcsin((i-1).2^{1-d}-1)]\} \end{aligned}$$

Most cosine functions are part of two successive terms with different signs. Rearranging the formula this way yields:

n odd:

$$\begin{aligned} c_n = &\frac{1}{2\pi jn}.\{\tfrac{1}{2} + 2^{-d}.\sum_{i=1}^{2^{d-1}}\cos[n.\arcsin(i.2^{1-d})]\} + \\ &\frac{1}{2\pi jn}.\{2^{-d}.\sum_{i=1}^{2^{d}}\cos[n.\arcsin(1-i.2^{1-d})]\} + \\ &\frac{1}{2\pi jn}.\{-\tfrac{1}{2} + 2^{-d} + 2^{-d}.\sum_{i=1}^{2^{d-1}}\cos[n.\arcsin(i.2^{1-d}-1)]\} \end{aligned}$$

This formula can be simplified, using the fact that the arc sine function is odd and the cosine function is even. This yields:

n odd:

$$c_n = \frac{1}{2\pi jn}.\{2^{-d}.\sum_{i=1}^{2^{d-1}}\cos[n.\arcsin(i.2^{1-d})] + 2^{-d}.\sum_{i=1}^{2^{d}}\cos[n.\arcsin(i.2^{1-d}-1)] +$$

$$2^{-d} + 2^{-d}.\sum_{i=1}^{2^{d-1}}\cos[n.\arcsin(i.2^{1-d}-1)]\ \}$$

Moreover,

$$\text{because} \sum_{i=1}^{2^{d-1}} i.2^{1-d} = 2^{1-d} + 2.2^{1-d} + .. + (1 - 2^{1-d}) + 1$$

$$\text{and} \sum_{i=1}^{2^{d-1}} (1-i.2^{1-d}) = (1 - 2^{1-d}) + .. + 2^{1-d}$$

and because $\cos[n.\arcsin(1)] = \cos[n.\pi/2] = 0$ when n is odd the first and the third series in the formula for c_n are equivalent.

Then:

$$\sum_{i=1}^{2d}(i.2^{1-d}-1) = (2^{1-d}-1) + (2.2^{1-d}-1) +..+ -2^{1-d} + 0 + 2^{1-d} +..+ (1-2^{1-d}) + 1$$

When the right member equals 1 the cosine term in the expression for c_n will be zero; when the right member equals 0, the cosine term equals 1. All other terms occur in pairs which result in same cosine terms due to the oddness of the arc sine function. Therefore the second term in the expression for c_n equals two times a summation up to 2^{d-1} plus 2^{-d}. Therefore:

n odd:

$$c_n = \frac{2^{-d}}{2\pi jn}.\{\ 2 + 2.\sum_{i=1}^{2^{d-1}}\cos[n.\arcsin(i.2^{1-d})] + 2.\sum_{i=1}^{2^{d-1}}\cos[n.\arcsin(i.2^{1-d}-1)]\ \}$$

Finally, because of the summation equivalence just mentioned the last two terms of the right member in this equation are equal as well:

$$\text{n odd: } c_n = \frac{2^{-d}}{\pi jn}.\{\ 1 + 2\sum_{i=1}^{2^{d-1}}\cos[n.\arcsin(i.2^{1-d})]\ \}$$

This equation equals (5.5) because the last term ($i=2^{d-1}$) equals zero.

The final formula for c_n has been used in an interactive computer program written in Turbo Pascal on a personal computer. When the number of data bits (d) and the largest harmonic nmax are entered the program calculates c_1 to c_{nmax} according to the flow charts given in figures A8.3 and A8.4. Because in Turbo Pascal the arc sine function is not defined, the following equality has been used:

$$\arcsin[x] = \arctan[x/\sqrt{(1-x^2)}]$$

This equality is easily derived from general goniometric formulas:

$x = \sin[y]$, so $y = \arcsin[x]$

$\sin^2[y] + \cos^2[y] = 1$, so $\cos[y] = (1-x^2)^{1/2}$

$\tan[y] = \sin[y]/\cos[y] = x/(1-x^2)^{1/2}$

$y = \arctan[\tan(y)] = \arctan[x/(1-x^2)^{1/2}] = \arcsin[x]$

Therefore, when $h:=2^{d-1}$ then $\arcsin[i/h] = \arctan[i/\sqrt{(h^2-i^2)}]$.

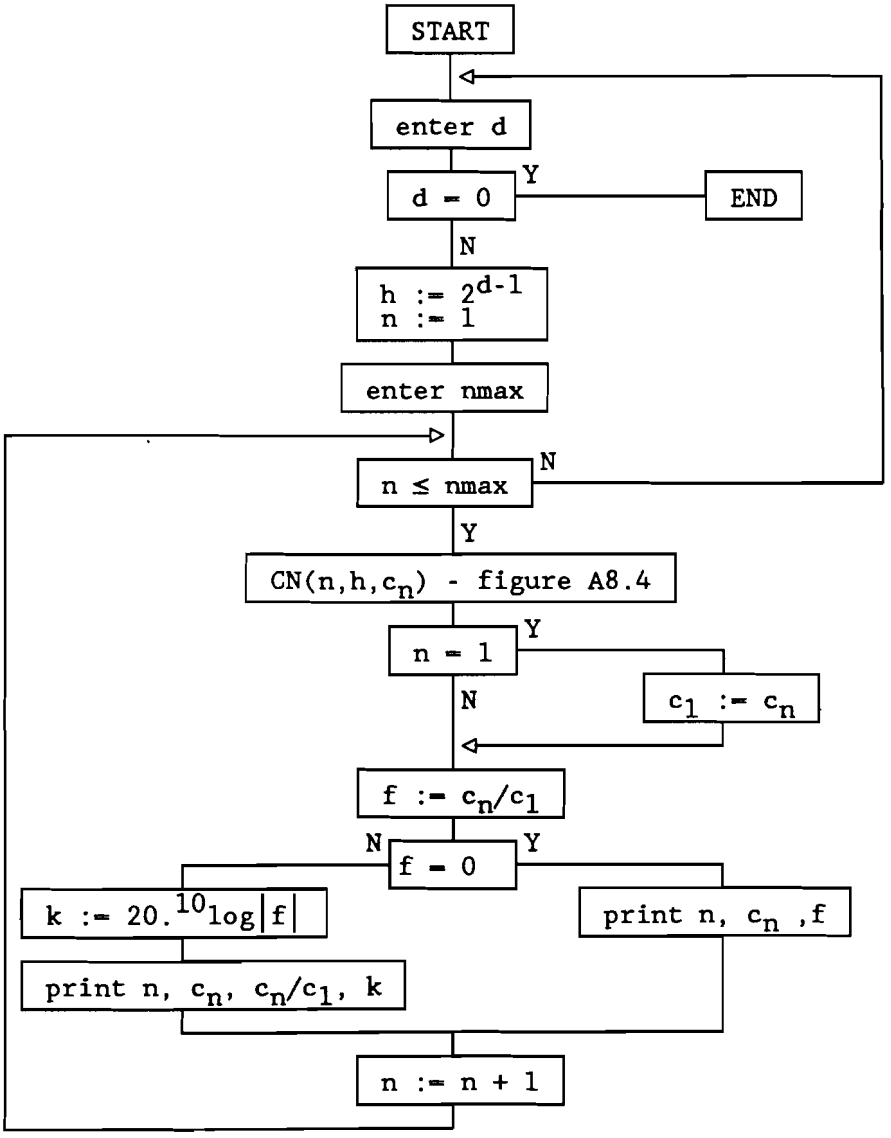


Fig.A8.3 Flow chart of the program used to calculate spectral coefficients of a quantized sine wave.

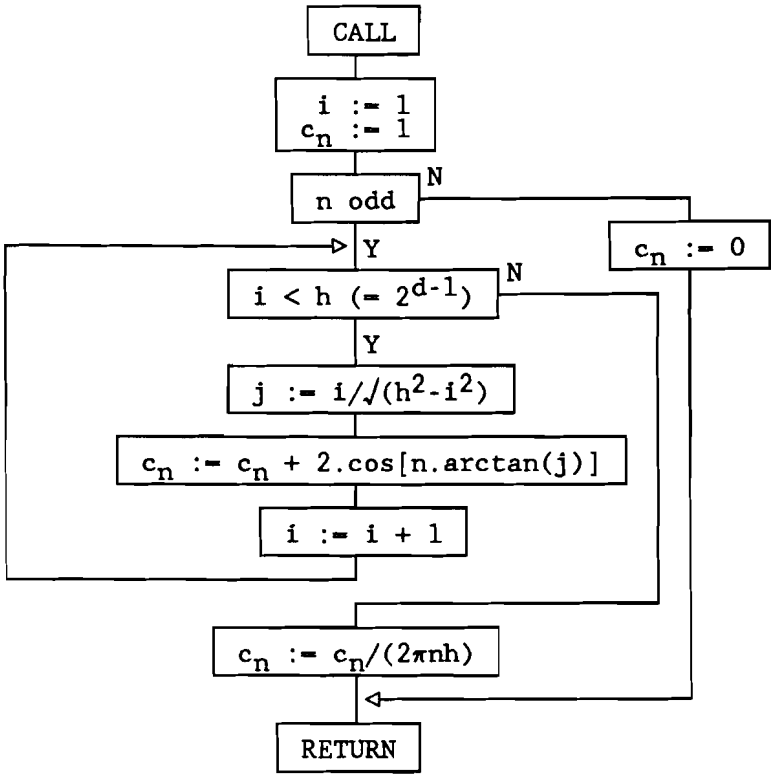


Fig.A8.4 Flow chart of the procedure CN which is called in the program given in figure A8.3.

On the next two pages the program output is shown in case that nmax = 30, for d = 1 to d = 8. A graphic representation of these results is given in chapter 5 (figure 5.5).

Table A8.1 Program output : spectral coefficients c_n of an analogue quantized sine wave

Width of sine ROM in number of bits (d):1				Width of sine ROM in number of bits (d):2			
n	imaginary coefficient	c_1 / c_n (lin)	c_1/c_n (dB)	n	imaginary coefficient	c_1 / c_n (lin)	c_1/c_n (dB)
1	-1.59154943E-01	1.000000000E+00	0.00	1	-2.17409695E-01	1.000000000E+00	0.00
2	0.000000000E+00	0.000000000E+00		2	0.000000000E+00	0.000000000E+00	
3	-5.30516477E-02	3.333333333E-01	-9.54	3	-2.65258238E-02	1.220084679E-01	-18.27
4	0.000000000E+00	0.000000000E+00		4	0.000000000E+00	0.000000000E+00	
5	-3.18309886E-02	2.000000000E-01	-13.98	5	1.165095046E-02	-5.35898385E-02	-25.42
6	0.000000000E+00	0.000000000E+00		6	0.000000000E+00	0.000000000E+00	
7	-2.27364204E-02	1.428571429E-01	-16.90	7	8.322107473E-03	-3.82784561E-02	-28.34
8	0.000000000E+00	0.000000000E+00		8	0.000000000E+00	0.000000000E+00	
9	-1.76838826E-02	1.111111111E-01	-19.08	9	-8.84194128E-03	4.066948931E-02	-27.81
10	0.000000000E+00	0.000000000E+00		10	0.000000000E+00	0.000000000E+00	
11	-1.44686312E-02	9.090909091E-02	-20.83	11	-1.97645178E-02	9.090909091E-02	-20.83
12	0.000000000E+00	0.000000000E+00		12	0.000000000E+00	0.000000000E+00	
13	-1.22426879E-02	7.692307692E-02	-22.28	13	-1.67238227E-02	7.692307692E-02	-22.28
14	0.000000000E+00	0.000000000E+00		14	0.000000000E+00	0.000000000E+00	
15	-1.06103295E-02	6.666666667E-02	-23.52	15	-5.30516477E-03	2.440169359E-02	-32.25
16	0.000000000E+00	0.000000000E+00		16	0.000000000E+00	0.000000000E+00	
17	-9.36205548E-03	5.882352941E-02	-24.61	17	3.426750136E-03	-1.57617172E-02	-36.05
18	0.000000000E+00	0.000000000E+00		18	0.000000000E+00	0.000000000E+00	
19	-8.37657595E-03	5.263157895E-02	-25.58	19	3.066039595E-03	-1.41025891E-02	-37.01
20	0.000000000E+00	0.000000000E+00		20	0.000000000E+00	0.000000000E+00	
21	-7.57880681E-03	4.761904762E-02	-26.44	21	-3.78940341E-03	1.742978113E-02	-35.17
22	0.000000000E+00	0.000000000E+00		22	0.000000000E+00	0.000000000E+00	
23	-6.91978013E-03	4.347826087E-02	-27.23	23	-9.45259545E-03	4.347826087E-02	-27.23
24	0.000000000E+00	0.000000000E+00		24	0.000000000E+00	0.000000000E+00	
25	-6.36619772E-03	4.000000000E-02	-27.96	25	-8.69638782E-03	4.000000000E-02	-27.96
26	0.000000000E+00	0.000000000E+00		26	0.000000000E+00	0.000000000E+00	
27	-5.89462752E-03	3.703703704E-02	-28.63	27	-2.94731376E-03	1.355649644E-02	-37.36
28	0.000000000E+00	0.000000000E+00		28	0.000000000E+00	0.000000000E+00	
29	-5.48810149E-03	3.448275862E-02	-29.25	29	2.008784562E-03	-9.23962732E-03	-40.69
30	0.000000000E+00	0.000000000E+00		30	0.000000000E+00	0.000000000E+00	

Width of sine ROM in number of bits (d):3				Width of sine ROM in number of bits (d):4			
n	imaginary coefficient	c_1 / c_n (lin)	c_1/c_n (dB)	n	imaginary coefficient	c_1 / c_n (lin)	c_1/c_n (dB)
1	-2.38390953E-01	1.000000000E+00	0.00	1	-2.45879995E-01	1.000000000E+00	0.00
2	0.000000000E+00	0.000000000E+00		2	0.000000000E+00	0.000000000E+00	
3	-1.05940717E-02	4.443990651E-02	-27.04	3	-3.94574882E-03	1.604745770E-02	-35.89
4	0.000000000E+00	0.000000000E+00		4	0.000000000E+00	0.000000000E+00	
5	8.247203625E-03	-3.45952878E-02	-29.22	5	3.566199024E-03	-1.45038193E-02	-36.77
6	0.000000000E+00	0.000000000E+00		6	0.000000000E+00	0.000000000E+00	
7	-4.29470033E-03	1.801536624E-02	-34.89	7	-2.93369033E-03	1.193139088E-02	-38.47
8	0.000000000E+00	0.000000000E+00		8	0.000000000E+00	0.000000000E+00	
9	-6.44224440E-04	2.702386274E-03	-51.37	9	2.017747764E-03	-8.20622989E-03	-41.72
10	0.000000000E+00	0.000000000E+00		10	0.000000000E+00	0.000000000E+00	
11	4.083861663E-03	-1.71309423E-02	-35.32	11	-8.62082674E-04	3.506111482E-03	-49.10
12	0.000000000E+00	0.000000000E+00		12	0.000000000E+00	0.000000000E+00	
13	-2.48218552E-03	1.041224716E-02	-39.65	13	-3.46917617E-04	1.410922500E-03	-57.01
14	0.000000000E+00	0.000000000E+00		14	0.000000000E+00	0.000000000E+00	
15	-3.66669511E-03	1.538101619E-02	-36.26	15	1.247013837E-03	-5.07163601E-03	-45.90
16	0.000000000E+00	0.000000000E+00		16	0.000000000E+00	0.000000000E+00	
17	4.901657762E-03	-2.05614252E-02	-33.74	17	-1.39035235E-03	5.654597267E-03	-44.95
18	0.000000000E+00	0.000000000E+00		18	0.000000000E+00	0.000000000E+00	
19	5.011798931E-03	-2.10234443E-02	-33.55	19	5.358825434E-04	-2.17944751E-03	-53.23
20	0.000000000E+00	0.000000000E+00		20	0.000000000E+00	0.000000000E+00	
21	-5.93267913E-03	2.488634343E-02	-32.08	21	9.322689455E-04	-3.79156078E-03	-48.42
22	0.000000000E+00	0.000000000E+00		22	0.000000000E+00	0.000000000E+00	
23	-1.05506301E-02	4.425767830E-02	-27.08	23	-1.85768173E-03	7.555237368E-03	-42.44
24	0.000000000E+00	0.000000000E+00		24	0.000000000E+00	0.000000000E+00	
25	-5.28815125E-03	2.218268428E-02	-33.08	25	1.024663496E-03	-4.16733169E-03	-47.60
26	0.000000000E+00	0.000000000E+00		26	0.000000000E+00	0.000000000E+00	
27	-2.18586235E-03	9.169233645E-03	-40.75	27	1.290218372E-03	-5.24734992E-03	-45.60
28	0.000000000E+00	0.000000000E+00		28	0.000000000E+00	0.000000000E+00	
29	-2.72902769E-03	1.144769822E-02	-38.83	29	-2.57838328E-03	1.048634836E-02	-39.59
30	0.000000000E+00	0.000000000E+00		30	0.000000000E+00	0.000000000E+00	

Table A8.1 (cont.) Program output : spectral coefficients c_n of an analogue quantized sine wave.

Width of sine ROM in number of bits (d):5				Width of sine ROM in number of bits (d):6			
n	imaginary coefficient	c_1 / c_n (lin)	c_1/c_n (dB)	n	imaginary coefficient	c_1 / c_n (lin)	c_1/c_n (dB)
1	-2.48540580E-01	1.000000000E+00	0.00	1	-2.49483524E-01	1.000000000E+00	0.00
2	0.000000000E+00	0.000000000E+00		2	0.000000000E+00	0.000000000E+00	
3	-1.42907817E-03	5.749878639E-03	-44.81	3	-5.11153493E-04	2.048846692E-03	-53.77
4	0.000000000E+00	0.000000000E+00		4	0.000000000E+00	0.000000000E+00	
5	1.365522723E-03	-5.49416407E-03	-45.20	5	5.002488649E-04	-2.00513788E-03	-53.96
6	0.000000000E+00	0.000000000E+00		6	0.000000000E+00	0.000000000E+00	
7	-1.26361016E-03	5.084120107E-03	-45.88	7	-4.83269219E-04	1.937078695E-03	-54.26
8	0.000000000E+00	0.000000000E+00		8	0.000000000E+00	0.000000000E+00	
9	1.117253555E-03	-4.49525609E-03	-46.94	9	4.595420426E-04	-1.84197351E-03	-54.69
10	0.000000000E+00	0.000000000E+00		10	0.000000000E+00	0.000000000E+00	
11	-9.21683940E-04	3.708384122E-03	-48.62	11	-4.28303962E-04	1.716762515E-03	-55.31
12	0.000000000E+00	0.000000000E+00		12	0.000000000E+00	0.000000000E+00	
13	6.767732654E-04	-2.72298900E-03	-51.30	13	3.888303704E-04	-1.55854128E-03	-56.15
14	0.000000000E+00	0.000000000E+00		14	0.000000000E+00	0.000000000E+00	
15	-3.91239136E-04	1.574145904E-03	-56.06	15	-3.40611583E-04	1.365266843E-03	-57.30
16	0.000000000E+00	0.000000000E+00		16	0.000000000E+00	0.000000000E+00	
17	8.691760366E-05	-3.49711921E-04	-69.13	17	2.835763176E-04	-1.13665349E-03	-58.89
18	0.000000000E+00	0.000000000E+00		18	0.000000000E+00	0.000000000E+00	
19	1.986516354E-04	-7.99272439E-04	-61.95	19	-2.18354061E-04	8.752243748E-04	-61.16
20	0.000000000E+00	0.000000000E+00		20	0.000000000E+00	0.000000000E+00	
21	-4.14075136E-04	1.666026274E-03	-55.57	21	1.465532201E-04	-5.87426447E-04	-64.62
22	0.000000000E+00	0.000000000E+00		22	0.000000000E+00	0.000000000E+00	
23	5.035223798E-04	-2.02591617E-03	-53.87	23	-7.10120215E-05	2.846361173E-04	-70.91
24	0.000000000E+00	0.000000000E+00		24	0.000000000E+00	0.000000000E+00	
25	-4.25542014E-04	1.712163117E-03	-55.33	25	-4.04416978E-06	1.621016776E-05	-95.80
26	0.000000000E+00	0.000000000E+00		26	0.000000000E+00	0.000000000E+00	
27	1.790055660E-04	-7.20226718E-04	-62.85	27	7.302746998E-05	-2.92714600E-04	-70.67
28	0.000000000E+00	0.000000000E+00		28	0.000000000E+00	0.000000000E+00	
29	1.731560779E-04	-6.96691374E-04	-63.14	29	-1.29294878E-04	5.182501676E-04	-65.71
30	0.000000000E+00	0.000000000E+00		30	0.000000000E+00	0.000000000E+00	

Width of sine ROM in number of bits (d):7				Width of sine ROM in number of bits (d):8			
n	imaginary coefficient	c_1 / c_n (lin)	c_1/c_n (dB)	n	imaginary coefficient	c_1 / c_n (lin)	c_1/c_n (dB)
1	-2.49817311E-01	1.000000000E+00	0.00	1	-2.49935394E-01	1.000000000E+00	0.00
2	0.000000000E+00	0.000000000E+00		2	0.000000000E+00	0.000000000E+00	
3	-1.81751973E-04	7.275395452E-04	-62.76	3	-6.44404953E-05	2.578286101E-04	-71.77
4	0.000000000E+00	0.000000000E+00		4	0.000000000E+00	0.000000000E+00	
5	1.798542575E-04	-7.19943134E-04	-62.85	5	6.410770506E-05	-2.56497105E-04	-71.82
6	0.000000000E+00	0.000000000E+00		6	0.000000000E+00	0.000000000E+00	
7	-1.76950892E-04	7.083211802E-04	-63.00	7	-6.36034247E-05	2.544794624E-04	-71.89
8	0.000000000E+00	0.000000000E+00		8	0.000000000E+00	0.000000000E+00	
9	1.729769038E-04	-6.92413601E-04	-63.19	9	6.292167622E-05	-2.51751763E-04	-71.98
10	0.000000000E+00	0.000000000E+00		10	0.000000000E+00	0.000000000E+00	
11	-1.67851168E-04	6.718956662E-04	-63.45	11	-6.20547239E-05	2.482830579E-04	-72.10
12	0.000000000E+00	0.000000000E+00		12	0.000000000E+00	0.000000000E+00	
13	1.614815256E-04	-6.46398463E-04	-63.79	13	6.099328409E-05	-2.44036201E-04	-72.25
14	0.000000000E+00	0.000000000E+00		14	0.000000000E+00	0.000000000E+00	
15	-1.53771609E-04	6.155362417E-04	-64.21	15	-5.97267985E-05	2.389689490E-04	-72.43
16	0.000000000E+00	0.000000000E+00		16	0.000000000E+00	0.000000000E+00	
17	1.446296020E-04	-5.78941474E-04	-64.75	17	5.824377719E-05	-2.33035331E-04	-72.65
18	0.000000000E+00	0.000000000E+00		18	0.000000000E+00	0.000000000E+00	
19	-1.33979084E-04	5.363082476E-04	-65.41	19	-5.65322201E-05	2.261873324E-04	-72.91
20	0.000000000E+00	0.000000000E+00		20	0.000000000E+00	0.000000000E+00	
21	1.217719997E-04	-4.87444202E-04	-66.24	21	5.458011981E-05	-2.18376913E-04	-73.22
22	0.000000000E+00	0.000000000E+00		22	0.000000000E+00	0.000000000E+00	
23	-1.08003633E-04	4.323304608E-04	-67.28	23	-5.23760552E-05	2.095583756E-04	-73.57
24	0.000000000E+00	0.000000000E+00		24	0.000000000E+00	0.000000000E+00	
25	9.272918839E-05	-3.71188002E-04	-68.61	25	4.990987808E-05	-1.99691117E-04	-73.99
26	0.000000000E+00	0.000000000E+00		26	0.000000000E+00	0.000000000E+00	
27	-7.60812308E-05	3.045474732E-04	-70.33	27	-4.71734944E-05	1.887427533E-04	-74.48
28	0.000000000E+00	0.000000000E+00		28	0.000000000E+00	0.000000000E+00	
29	5.828681124E-05	-2.33317744E-04	-72.64	29	4.416173778E-05	-1.76692613E-04	-75.06
30	0.000000000E+00	0.000000000E+00		30	0.000000000E+00	0.000000000E+00	