USING DIGITAL SIGNAL PROCESSING TECHNIQUES IN LIGHT CONTROLLERS

Henrik Fløe Mikkelsen
Bang & Olufsen,
Struer, Denmark

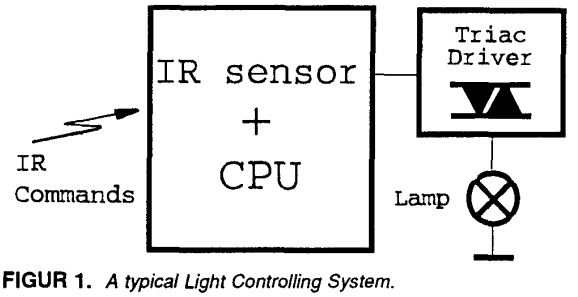
Abstract

In many countries we see that electric power stations - besides supplying the consumers with electricity- also sends code-signals in order to control public timers, syncronising watches, turn on/off the lights on the streets and finally control the speed of the electricity meter which is typically placed in the house of the consumer. This gives the consumer opportunity to only use electricity when it is cheapest - say, at nights. Unfortunately these control signals also disturbs various devices in the homes. One of such "noise" sensitive products are Light Controllers (sometimes called Light Dimmers). In this paper we propose a new noise cancelling method which we have implemented in an low-cost 8-bit microcontroller programmed for Light Controlling purposes.

Introduction

Various methods for controlling the dimming effect from light sources has been suggested and investigated for the past fourty years. Most of these light controllers has basicly been analog circuits.

A typical digital solution for dimming the light is to control the time triggering of a triac coupled direct in series to the light source through the power source (see figur 1). This controlling is done in an open loop configuration where the light intensity is supervised by the consumer. The open loop control function can be calculated in a lowcost micro-controller.



FIGUR 1. *A typical Light Controlling System.*

A triac is functionally an integrated pair of phase-controlled Thyristors connected in anti-parallel. It is a trigger-into-conduction device which can be turned on by a gate current pulse with correct polarity. The triac cannot be turned off again by gate pulses. This commutation is only possible by an AC-line or -load voltage wave.

One of the advantages of using a microcontroller in this application is that it gives us the opportunity to add other features to the Light Controller such as remote-controlling capabilities.

In some countries like Switzerland, Austria and Germany the electric power stations sends out controlling codes modulated onto the net frequency in order to syncronise all the public timers -like the watches on railway stations and to turn the street lights on or off [Ref.1]. Sometimes these signals also controls the counter speed of the electricity meter in peoples houses. The consumer however can take advantage of these signals by only use power when it is cheapest -say at nigths. This code-signal detection can be done automatically so that for instanse the washing machine only starts to wash when it detects that the electrical power is cheap.

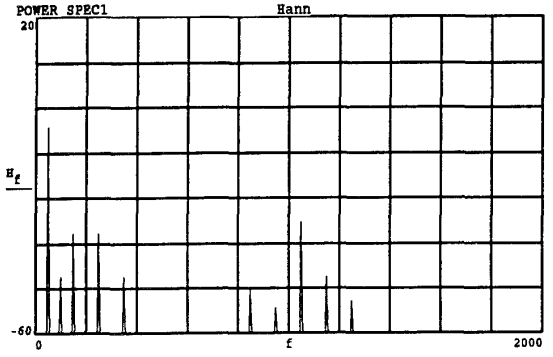
Nevertheless these nice features are detected as modulated noise by the light controllers in the house and the consequences are that the light controlled lamps starts to blink heavily -some light controllers even brakes down caused by the forced displacement of the direct-current due to the modulation noise. In section I these problems will be discussed further.

In the future the new light controllers have to filter out these noisy signal components. We now suggest a digital filtering technique that does not require a dedicated digital signal processor. Only a small lowcost 8-bit processor is nessesary to handle the remote-control commands and at the same time digitally process the signals to the triac-component in order to remove blinking effects and direct-current displacements. The method is discribed in Section II. Section III contains generel results and some suggestions for improvements of the method.

Contributed Paper
Manuscript received March 23, 1993

I. The problem

The noise currupted power signal, see the amplitude spectrum in fig. 2, can be estimated as a linear combination of the primary sinewave -that is the main power signal- and an additiv "noisy" sinewave with relatively unknown and time-varying frequency.



FIGUR 2. *Measurement of the power-net signals in Switzerland.*

Figur 2 shows a typical example of the noise currupted power signal. It's a recording from the power-net in Switzerland.

We could use a traditional analog bandpass filter around the main power frequency, but this would require a complicated filter with many components that will take up a lot of space in the final product layout. A better solution is to implement a digital noise-canceller in the present microprocessor (the main purpose of the processor is to handle infra-red command signals from a remote controlling device). Of course there are also some great advantages in using a digital solution -i.e. the quality of the product is higher due to the use of fewer external components, solderings etc. We choose to model the signal process as shown in equation 1.

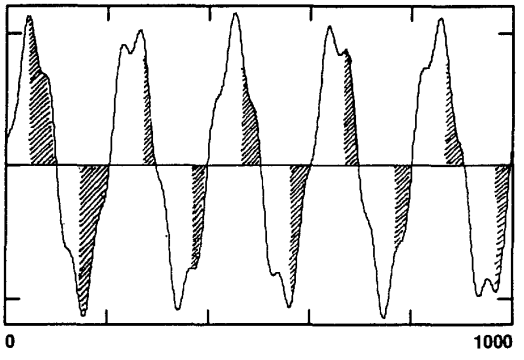
[EQ-1] $x(t) = s(t) + n(t)$

where

$$s(t) = A_o \cdot \sin(\omega_o \cdot t)$$
$$n(t) = A_n(t_o) \cdot \sin[\omega_n(t) \cdot t + \varphi]$$

Notice that $\omega_n(t)$ is a frequency which is time-dependant (indicated by the parameter t) [Ref.4]. This is also the case for $A_n(t)$ where t_o is the starting time of the noise frequency for instans t_o = ten o'clock in the evening. However -we can expect that $A_o >> A_n$. The noise frequency ω_n can be much higher than the main frequency ω_o and at times the noise frequency can be close to the main frequency as well.

We have previously stated that the triac is turned on by a trigger-pulse and turned off again by the forthcomming positive-going zero-crossing in the current. The triggerpulses are generated by the previous negative-going zero-crossing and time-delayed in the microprocessor before it's send to the triac-gate. This means that when the zero-crossings are currupted by noise the trigger pulses will have servere time-jitter. It's obvious that this time-jittering causes a fluctuation of the light intensity in the lampload. This is illustrated in figur 3 as changes in the power signal to the lampload (the gray zones in the figur). Since the signal to the lampload hasn't got a 50% duty cycle around 0 Volts we have to expect a displacement in the direct current value (i.e. the DC voltage).



FIGUR 3. *A typical noisy powersignal in the lampload.*

This DC displacement may course trouble when the load is complex - like the transformers used in low energy consuming lamps. If the displacement drives the transformer to its saturation area we get an increase in the current and an increase in the acoustical noise. Sometimes it even courses the fuse in the light controller to blow.

We want to design a circuit that makes the triac gatepulses insensitive to both frequency and phase modulation -or in other words- the microcontroller have to take these noise contributions into account when it's calculating the trigger time. How this is done is discribed in section II.

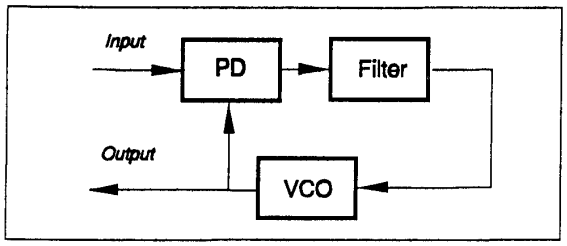
II. The noise cancelling method

As we have seen in section I we have to minimize the time-jittering in the triac gatepulse signal. Furthermore we have to bring the mean square value of the driving powersignal to zero. We can express the problem as shown in equation 2.

[EQ-2] $\omega(t) = \omega_o + G \cdot \omega_n(t)$

As seen in equation 2 we only have to consider the zerocrossings in the powersignal and minimizing the variance of the zerocrossings. If G (the amount of the noisefrequency) in equation 2 is set to zero we have no variance in the zerocrossings and therefore we have no timejitter. Obviously we can't identify the value of G so we have to estimate it from the measurered frequency $\omega(t)$.

Estimating the noisefrequency amount can be done iteratively in a closed loop control system. Since we only have to focus on the zerocrossings it's natural to use the principals known from Phase-Locked Loop theories. A typical Phase-Locked Loop control system is shown in figur 4.



FIGUR 4. *A common PLL circuit.*

The PLL consist of three major functions: A phase detector (PD), a loop filter (F) and a voltage controlled oscillator (VCO). Various textbooks such as Ref. [2],[3] treats the subject PLL very well so we will not go into any details about the PLL-theory. Here we just state that a PLL -among other features -has the ability to "lock" to the predominant frequency in the input signal. When the PLL is in lock the mean square value of the output from the phase detector has reach its minimum. This indicates that the frequency difference between the input and the VCO frequency is minimized. In figur 4 the term 'input' represents $\omega(t)$ and the 'output' ω_0.

Conventional PLL circuits are analog but as stated in the earlier section we are enforced to implement the circuit as a computer programcode - an algorithm. This implies that we have to model the analog PLL in the digital domain which we traditionally call the Z-domain. This is done in chapter IIa in this section.

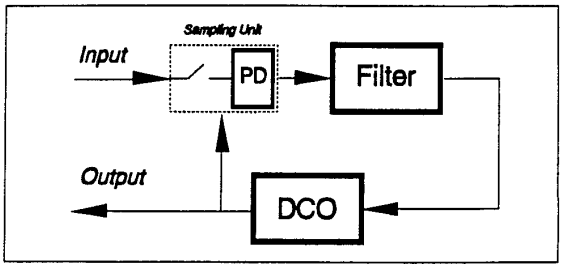
The digital model of the PLL is here called the software implemented PLL or -for short- the SPLL. In the litterature digital PLL's -or DPLL- are infact not really fully digital implementations of the analog PLL. They still use an analog loop filter. The only digital elements are the digital phasedetector and the digital controlled oscillator - sometimes called numerical controlled oscillator. An SPLL - however is a fully digital version of the analog PLL. So our goal is to design a SPLL which models the analog PLL as good as possible.

The input to the SPLL are information about the zerocrossing in the power signal. So we are actually sampling the phase of the signal.

IIa. A Digital modelling of the PLL

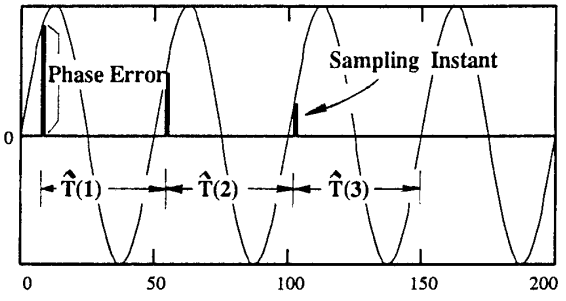
A digital PLL is often realised as a phase-sampler. This technique is called the zerocrossing PLL or ZC-PLL [Ref.5].

A simplified ZC-PLL is shown in figur 5. The sampling unit contains a sampler and a phase-detector. The input signal is now sampled at time instanses given by the controlled oscillator. The sampling frequency is adjusted by calculating and filtering the phase-error.



FIGUR 5. *A simplified ZC-PLL.*

Figur 5a illustrates the principal of ZC-PLL. One can see that when the sampling errors are small there exist nearly a linear relationship between the error value and the sampling correction coming from the oscillator.

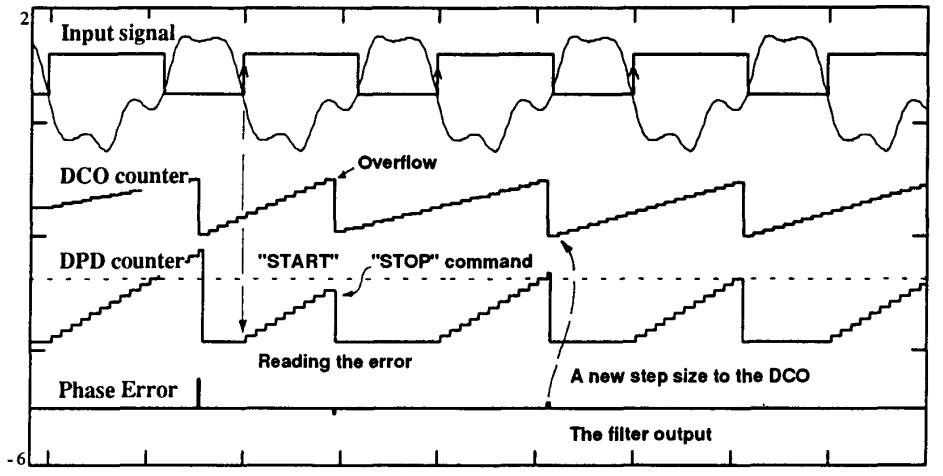


FIGUR 5a. *The ZC-PLL process*

Block Description

The SPLL functional blocks contains a digital phase-detector (DPD), the digital loop filter (DLF) and the digital controlled oscillator (DCO). The DPD is a software implemented J-K master slave flip-flop circuit. The DLF is a differens equation approximating an integrative process. Finally the DCO is a modulus 2^n counter with variable counting value -called stepsize (n is the length of the count registre). The ZC-PLL technique is tranferred to the SPLL as shown in the functional description in figur 6 (see the next page).

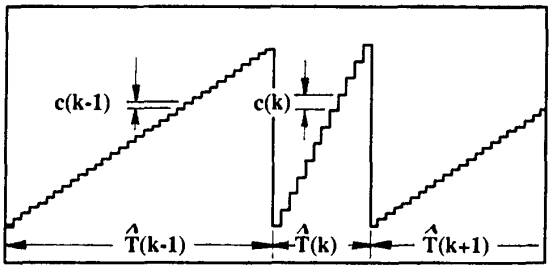
At the top of figur 6 we see the main power signal. A negative-going zerocrossing triggers the sampling unit and starts a counter in the DPD. This phase counter is later stopped by an overflow-flag from the DCO. The now reached counter-value expresses the phase-error between the input signal and the DCO. The loop filter receives the phase-error and calculates a new stepsize for the DCO -and thereby changing the DCO frequency. As seen the functions in the SPLL are highly non-linear but further analysis are



FIGUR 6. *The functions of the ZC-SPLL.*

based on the linearised model of the SPLL. The linearisation is done under the assumption that the phase-errors are small compared to the sampling frequency. Infact -we assume in the following that the SPLL is in lock. Gupta and Gill (Ref.[6]) has investigated the non-linear behavier of the ZC-PLL which is comparable to the SPLL. However -we have experienced that the assumptions made are not unrealistic in this perticularly application.

The Digital Oscillator -The DCO.



FIGUR 7. *The DCO function.*

Within one periode $\hat{T}(k)$ the incremental value ***c(k)*** will be constant. T_{clk} is the programme loop cyclus of the microprocessor -for instans 1 ms. When the counter reaches 2^n it resets and gets a new increment value ***c(k+1)***.

The DCO frequency is thus

[EQ-3] $$f(k) := \frac{C + \delta(k)}{2^n} \cdot \frac{1}{T_{clk}} := f_0 + \frac{\delta(k)}{2^n} \cdot f_{clk}$$

deducted from

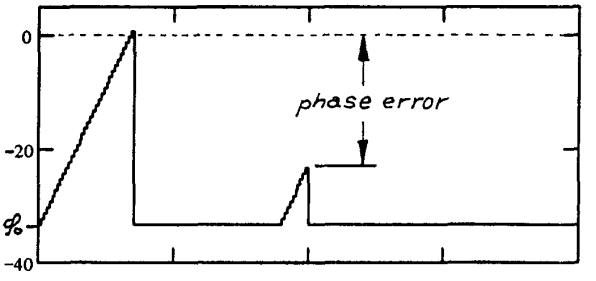
$$f(k) := \frac{1}{\hat{T}(k)} := \frac{c(k)}{2^n} \cdot \frac{1}{T_{clk}}$$

We have devided the increment value into two: a constant increment and a variable. The constant value alone gives us the free-running frequency of the DCO - f_o.

The Digital Phase Detector -The DPD.

The phase counter is incremented at the same timesteps as the DCO namely T_{clk}. But the incremental value is always 1 (see figur 7a). The phase-counter is initialised by a negativ value φ_o.

$$\varphi_0 := -\frac{1}{2} \cdot \frac{f_{clk}}{f_0}$$



FIGUR 7a. *The DPD function.*

Combining the DCO and the DPD.

We mathematically estimate the DPD and the DCO by the transferfunction G(z).

[EQ-4] $$G(z) := \frac{\hat{T}(k)}{E(k)} := K_N \cdot \frac{z^{-1}}{1 - z^{-1}} \Bigg|_{D(z)=1}$$

z^{-1} represent a unit delay in the digital domain so that x(n-1) -which is x(n) delayed one sampleperiod - has the Z-

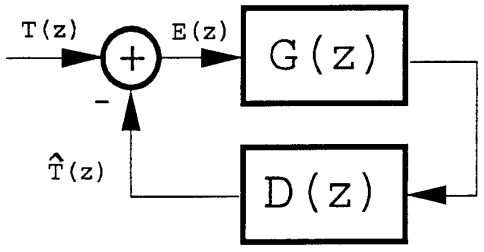
transform $X(z) \cdot z^{-1}$.

Appendix A gives a deduction of G(z). Intuitively one sees that this function has to be right. The term in the denominator of equation 4 is the Z-transformation of an integrator in the s-domain. Since the predicted frequency from the DCO is based on previous error values there must exist a zero-point (z^{-1} in the transferfunction). The constant K_N is a multiplication of the phase-detector constant K_p and the DCO constant K_o.

[EQ-4a] $K_N := K_o \cdot K_p$ $\quad \begin{cases} K_o := 2^{-n} \cdot f_{clk} \\ K_p := \frac{1}{2} \cdot \frac{f_{clk}}{f_o^2} \end{cases}$

The mathematical model of the SPLL.

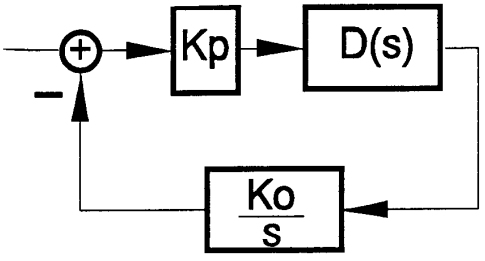
A model of the SPLL function is shown in figur 8.



FIGUR 8. *The SPLL block diagram.*

It would be nice to make use of known analog PLL handrules in designing the digital loop filter. To achieve this we have to transform the analog PLL into the digital domain.

Figur 9 shows a linear model of the analog PLL blockdiagram -it's comparable to figur 4.



FIGUR 9. *The analog PLL.*

The analog PLL has the transfer function:

[EQ-5] $H_a(s) := \frac{\omega_o(s)}{\omega(s)} := \frac{K_p \cdot K_o \cdot D(s)}{s + K_p \cdot K_o \cdot D(s)}$

One of the most popular filtertypes are the proportional-integration filter (PI-filter) given by the function:

[EQ-5a] $D(s) := \frac{1 + \tau_2 \cdot s}{\tau_1 \cdot s}$

τ_1 and τ_2 are the timeconstants of the filter. At low frequencies this filter-type is integrating the phase-error and at high frequencies it contributes as an attenuation. This filtertype results in the closed loop transfer function:

[EQ-5b] $H_a(s) := \frac{2 \cdot \zeta \cdot \omega_n \cdot s + \omega_n^2}{s^2 + 2 \cdot \zeta \cdot \omega_n \cdot s + \omega_n^2}$

$\omega_n := \sqrt{\frac{K_o \cdot K_p}{\tau_1}}$ $\qquad \zeta := \frac{\tau_2}{2} \cdot \omega_n$

ω_n is the loop bandwidth and ζ is the damping factor. Typically $\zeta = 0.5$ because of the resultant transient behavier. ω_n can be chosen freely. If one needs a tracking PLL it should be a high frequency and it should be chosen small if one needs a noise-filtering function. In our case we need both functions - a tracker to get into lockrange quickly and a noise-filter hereafter.

In our research it was found that when transforming $H_a(s)$ into the digital $H_a(z)$ the best practical results was accomplised by using the so-called Forward Euler Transformation (FET).

This gives us:

[EQ-6] $H_a(z) := H_a(s) \Big|_{s := \frac{1}{T} \cdot \left(\frac{1 - z^{-1}}{z^{-1}} \right)}$

$$H_a(z) := \frac{2 \cdot \zeta \cdot \omega_n \cdot T \cdot z^{-1} + \left(\omega_n \cdot T - 2 \cdot \zeta \right) \cdot \omega_n \cdot T \cdot z^{-2}}{1 + \left(2 \cdot \zeta \cdot \omega_n \cdot T - 2 \right) \cdot z^{-1} + \left[1 + \left(\omega_n \cdot T - 2 \cdot \zeta \right) \cdot \omega_n \cdot T \right] \cdot z^{-2}}$$

When using FET one should always check the result for possible instability. Although the s-plane model of the PLL is stable it doesn't neccesarily imply that the digital transform is going to be stable as well. This is not typical to digital transforms. A transform like the bilinear will give a stable function in the digital domaine if the s-model is stable. We doesn't -as already stated - use the bilinear transform.

When matching the analog loop filter D(s) into a digital loop filter we get:

[EQ-7] $D(z) := K_f \cdot \left(\frac{1 - A \cdot z^{-1}}{1 - z^{-1}} \right)$

The closed loop transfer function of the SPLL is found to be

[EQ-7a] $$H_d(z) := \frac{\hat{T}(z)}{T(z)} \Leftrightarrow$$

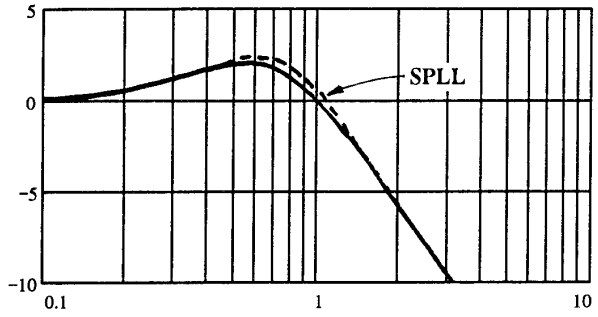
$$H_d(z) := \frac{K_f K_0 K_p z^{-1} - K_f K_0 K_p A z^{-2}}{1 + (K_f K_0 K_p - 2) z^{-1} + (1 - K_f K_0 K_p A) z^{-2}}$$

A coefficient-identification is done by comparing equation 6 and equation 7a. Therefore

[EQ-8] $$K_f := \frac{2 \zeta \omega_n T}{K_0 K_p} \qquad K_p := 1 - \frac{T}{2} \frac{\omega_n}{\zeta}$$

We can evaluate the identification by comparing the frequency characteristics for $H_a(z)$ and $H_d(z)$.

In figur 10 we have the closed loop transfer functions of the two PLL-versions. The frequency characteristics are computed as a Q-transformation of $H_a(z)$ and $H_d(z)$ (the definition of the Q-transformation is shown in appendix).

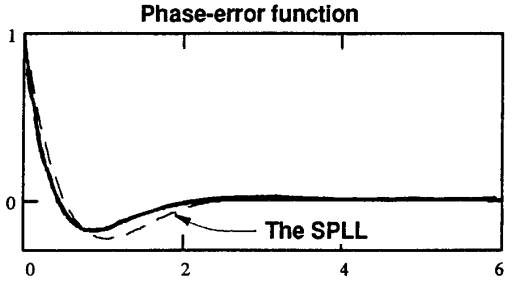


FIGUR 10. *Comparing $H_a(q)$ and $H_d(q)$.*

At low frequencies the two functions acts very similar and at the resonance frequency we observe a slight differens. At higher frequencies however we can expect that the stopband attenuation of the SPLL isn't as good as the analog PLL. This is typical for Forward Euler Transformations. The differens is about 3 dB at ω=100 ω_n and it will not cause any problems. But one should have in mind that the quantization noise of the SPLL contributes negatively to the overall noise floor. So the design criteria for the SPLL have to be stronger than for the analog PLL.

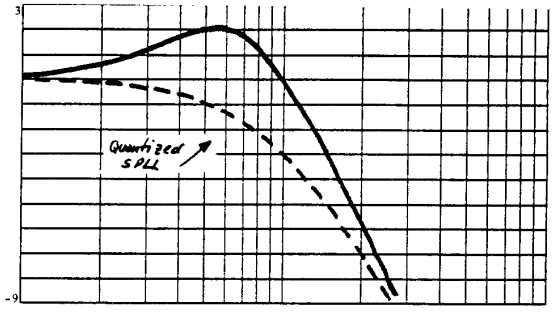
Another noise-parameter one should be aware of is already observable in figur 5a. The SPLL is a non-uniform sampler. The *T(k)*'s is not equally spaced. This makes it difficult to precisely predict the SPLL response upon a noisy input signal. By designing the SPLL with considerable phase and gain margins one avoids the instability risc's.

The respective step-responses of the phase errors are compared in figur 11. The two functions show very similar behavier.



FIGUR 11. *Comparing $E_a(t)$ and $E_d(n \cdot T_o)$.*

The digital error function is based on an 8-bit precision code. It's advisable to use double precision arithmetics when calculating the loop filter. By using 8-bit quantization overall in the SPLL one get a very poor result as shown in



FIGUR 12. *The consequence of using 8-bit quantization in the DLF.*

Now we should be able to design the noise-cancelling algorithm. In the next chapter we setup the design criteria for the SPLL noise-canceller.

IIb. Designing the noise-canceller.

We stated in the previous chapter that it was neccessary to use the SPLL in two modes -a tracking mode and a filtering mode.

The tracking-mode is only used in the up-start of the light controller. The SPLL has to quikly lock onto the main frequency. When the phase-error becomes sufficiently small -numerically speaking -the SPLL switches to the filtering mode. This filter will ignore any sudden changes in the input signal. But if the noise contributes with a constant phase-error the filter will slowly lock onto the new frequency and thereby minimize the error.

In the filtering mode it was found that the bandwidth of the SPLL should be around $\omega_n \approx 3\zeta$. In the tracking mode we found that it was sufficient with a tracking bandwidth of $\omega_{track} = 4\omega_n$.

Our goal was to minimize the time-jittering in the triac-gate signal. This goal is limited by the amplitude quantization noise in the SPLL and by the time-quantization done in computing the SPLL programme-loop. We expect the noise

to be white noise with a squared distribution of the power density. By neglecting the quantization-noise contribution from the other functional blocks (the DPD and the DLF) we can calculate the DCO noise-variance as a function of the loop frequency f_{clk}.

[EQ-9] $$\sigma(T_{clk}) := \sqrt{\frac{1}{12} \cdot T_{clk}^2}$$

In figur 13 we see this function expressed in a plot. At a programme-loop frequency of 3000 Hz one can expect a diviation in the ω_0 of about 95 µs (the indicated box in figur 13). In our application we have given the specification of a triac-trigger resolution greater than 100 µs resulting in a f_{clk}>2800 Hz -which is realistic in the proposed microprocessor.

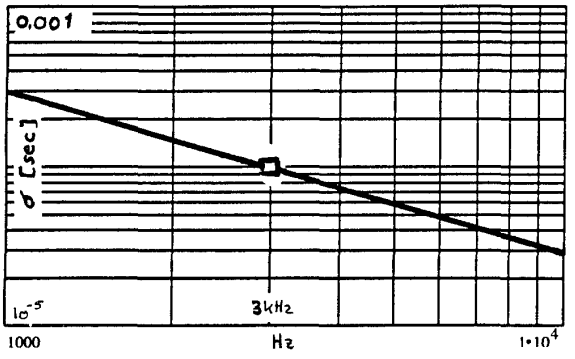


FIG.13. *The standard deviation of f_{clk} due to quantization.*

The implementation of the digital loop filter is a straight-forward procedure. Figur 14 shows a direct form 1 representation of the loop filter. The coefficient values are calculated as 8-bit integers. All multiplications are 16-bit representatives.

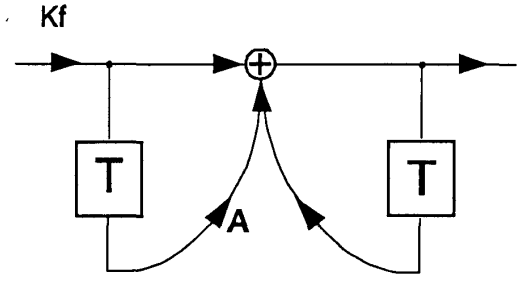


FIG. 14. *The digital loop filter realization.*

III. Results.

The SPLL algorithm is implemented in an 8-bit microcontroller · The microcontroller mainly serves as a light controlling unit (LC for short) with remote controlling and programmable light intensity capabillities. Since the SPLL algorithm isn't the main function of the micro-controller the SPLL may be interrupted at times. In fact the remote control Infra-red sensor alone interrupts every 781 µs. Carefully programme architecture can however prevent any filtrering failure caused by these interruptions.

We have tested the light controller in Austria, Switzerland and Germany with very convincing results. In the tests we compared a light controller without a SPLL (B&O LC-1), a light controller with 2.nd order analog filtrering (B&O LC-1b) and finally the new light controller with the SPLL (B&O LC-2). These tests was carried out in an electric power plant in München, Germany, at Bang & Olufsen, Wien and at Bang & Olufsen, Zürich. In Germany the noise frequencies consist of pulses with frequencies between 180 and 750 Hz. The amplitude is 9-20 Volts. In München -where the German investigation took place- the noise frequency is 283 1/3 Hz. In Switzerland and in Austria the specification of the noise-frequencies wasn't known at the time. The main conclusions from the investigations was that the LC-1 -of course- did not succeed the test. The light intensity fluctuated heavily. The LC-1b and the LC-2 showed a good result. Only at light intensities below 50% a slight variation in the light intensity was observable. However -this was only the case for noise frequencies close to a harmonic of f_0. The over-all performance of the LC-2 was comparable to the LC-1b in all the tests. Originally this was also our goal. No attempt was made to make the LC-2 perform better than the LC-1b. In the final design of the product LC-2 there was no room for an analog filter so we had to design a software solution.

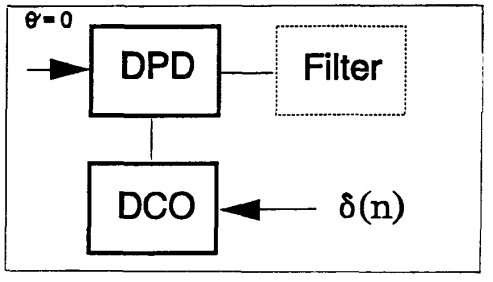
Future improvement of the method of digital noise-filtrering could be to expand the SPLL to be able to control the turn-off angle of the triac-circuit. This requires implementation of a double zero-crossing pll algorithm. The zero-crossings that turns the triac off can at times be displaced in time compared to the triggering zero-crossings. The SPLL is blind to this type of phase-errors. When the light intensity angle of the triac is small these phase-errors can be detected as a little blinking of the light. When detecting both zero-crossings with-in a periode it should be possible to calculate when the triac should turn off to provide minimum phase-errors. However -it is very difficult to turn-off a triac so the turn-off estimate must be transferred to a modified turn-on angle in every half-period of the power signal. An even simpler improvement consist of increasing the order of the loop filter resulting in greater stopband attenuation.

Conclusional remarks.

A new digital noise-filtrering method for light controllers has been implemented in an 8-bit micro-controller. The performance has been investigated in several location around Europe with convincingly good results. This new light controller -the Bang & Olufsen LC2- has gone into production by october 1992.

Appendix A. The Z-transform of the SPLL.

We are considering the impulsresponse of the phase-detector and the DCO based upon zero-phase input (see figur A1). An impulse stimuli set on the DCO input results in a stepresponse on the phase-detector output.



Figur A1.

The period of the DCO based on the impulse has the time-length T_1.

$$T_1 := \frac{2^n}{\gamma + 1} \cdot \frac{1}{f_{clk}}$$

The other periodes has the length T.

$$T := \frac{2^n}{\gamma} \cdot \frac{1}{f_{clk}}$$

So the DCO overflow happens earlier than expected.

$$\Delta T := T - T_1$$

$$\Delta T := \frac{2^n}{f_{clk}} \cdot \left(\frac{1}{\gamma} - \frac{1}{\gamma + 1} \right)$$

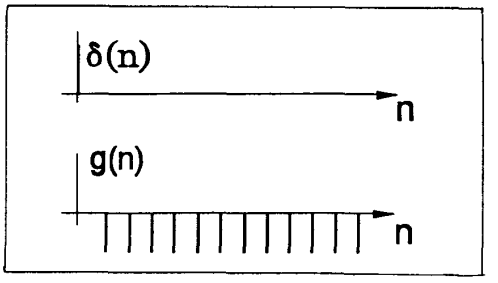
$$\Delta T := \frac{2^n}{f_{clk}} \cdot \frac{1}{\gamma^2} \qquad \text{for } \gamma > 1$$

This results in a phase-error of

$$g := \frac{2^{-n}}{f_{clk}} \cdot \frac{1}{\gamma^2} \cdot f_{clk}$$

$$g := \frac{2^{-n}}{\gamma^2}$$

Figur A2 shows the impulse responses.



Figur A2.

Now we can calculate the loop transferfunction.

$$G(z) := \sum_n g(n) \cdot z^{-n} \qquad n := -\infty \,..\, \infty$$

<=>

$$G(z) := \sum_n -\frac{2^n}{\gamma^2} \cdot z^{-n} \qquad n := 1 \,..\, \infty$$

=>

$$G(z) := -\frac{2^n}{\gamma^2} \cdot z^{-1} \cdot \sum_i z^{-i} \qquad i := 0 \,..\, \infty$$

=>

$$G(z) := -\frac{2^n}{\gamma^2} \cdot z^{-1} \cdot \left(1 + z^{-1} + z^{-2} + ...\right)$$

=>

$$G(z) := -\frac{2^n}{\gamma^2} \cdot \frac{z^{-1}}{1 - z^{-1}}$$

<=>

$$G(z) := K \cdot \frac{z^{-1}}{1 - z^{-1}} \qquad \text{Q.E.D.}$$

Appendix B. The Q-transform.

The Q-transformation is a substitution of

$$q := \frac{2}{T} \cdot \tanh\left(s \cdot \frac{T}{2}\right) := v + j \cdot u$$

If we make this substituion in equation 7a we get

$$H_d(q) := K_{cl} \cdot \frac{\left[(A-1) \cdot T^2 \cdot q^2 + 4 \cdot A \cdot T \cdot q + 4 \cdot (A+1)\right]}{\left[4 - K_{cl} \cdot (A+1)\right] \cdot T^2 \cdot q^2 + 4 \cdot K_{cl} \cdot A \cdot T \cdot q + 4 \cdot K_{cl} \cdot (1-A)}$$

$$K_{cl} := K_o \cdot K_d \cdot K_f$$

The transformation of the analog PLL into the Q-domain gives us

$$H_a(q) := \frac{2 \cdot \zeta \cdot \omega_n \cdot q + \omega_n^2}{q^2 + 2 \cdot \zeta \cdot \omega_n \cdot q + \omega_n^2}$$

The frequency characteristics are calculated by setting q= ju ($u \in \mathbf{R}$).

Appendix C. Calculation of the variance in the DCO due to quantization.

When calculating the DCO frequency f_o in an 8-bit microprocessor we make a quantization-error by rounding the theoretical value into an 8 bit representation. This results in noise-jitter (which can be seen in figur 7 -the overflow of the second periode does not arrive at the same amplitude as the previous overflow). The noise are expected to be equally distributed with-in one time-clock, T_{clk}. The power density of the noise is calculated as the integral

$$\psi_q^2 := \int_{-\infty}^{\infty} q^2 \cdot f_Q(q)\, dq := \frac{1}{12} \cdot T_{clk}^2$$

where ψ^2 is the total quantization power and $f_Q(q)$ is the density function of the quantization-error.

Acknowledgement.

The author is very much indebted to Mr. Jørgen Selmer Jensen, Bang & Olufsen R&D, for the noise-cancelling ideas and for the stimulating discussions and careful review of this manuscript.

References.

[1] Internal Bang & Olufsen documents concerning field tests on LC2. 15th August 1991. *In Danish.*

[2] Technical Compendium on Phase-Locked Loops. The Technical University of Denmark. August 1986. *In Danish.*

[3] Theorie und Anwendungen des Phase-Locked Loops. Roland Best, AT-Verlag, 1981. *In German.*

[4] Internal Bang & Olufsen documents on Pulsetone control systems on the 50 Hz powernet in Germany, Austria and Switzerland. 10th Juli 1990. *In Danish.*

[5] A Survey of Digital Phase-Locked Loops. W.C.Lindsey et. al. Proc. IEEE vol.69 no.4 April 1981.

[6] On Higher Order Discrete Phase-Locked Loops. G.S.Gill and S.C.Gupta. IEEE Trans. on Aerospace and Electronic Systems, Vol. AES-8 no.5 Sept.1972.

Biography of Mr. Henrik Fløe Mikkelsen

In 1989 Henrik F. Mikkelsen received the degree of science in electronic engineering at the Highschool of Engineering in Aarhus, Denmark. His thesis of graduation was "System Identification of Room-acoustics". For a short period of time he worked with the company t.c.electronics in the fields of estimation and ARMA modelling of room-acoustics.
In 1990 he joined the Bang & Olufsen Research Center where his main topics were methods of parametric estimation, adaptive signal processing and control theory. Here he developed the mathematical foundation of the Software Realization of the PLL described in this article.

At present he is involved with developing a Digital Compact Cassette (DCC) tape-recorder.

His main interests at the moment are digital signal processing, adaptive systems, time-delay estimation and control theory.