Master of Science Thesis

# Bumblebee Killer

Per Rosengren Andreas Nilsson

November 1999

Master of Science in Electrical Engineering 1999
University of Karlskrona/Ronneby
Department of Applied Signal Processing
Examiner: Professor Ingvar Claesson, University of Karlskrona/Ronneby
Supervisors: Professor Ingvar Claesson, University of Karlskrona/Ronneby
Per Ljungberg, Johan Udén and Gabor Lakatos, Ericsson Mobile Communications
Thesis: MME 99-07

# Bumblebee Killer

Per Rosengren Andreas Nilsson

November 1999

**Abstract**

A common problem in GSM terminals is an interfering signal nicknamed the "Bumblebee". This interference is generated by the switching nature of TDMA cellular telephony, the radio circuits are switched on and off with the radio access rate. In GSM, this frequency is approximately 217 Hz. This frequency and its harmonics get into the analog microphone signal, and produce a very annoying periodic noise in the uplink speech.

This thesis contains a study of four different software solutions to suppress this interference. These four different methods are denoted Notch filter, Matched filter, Correlators and Interpolation in frequency. It is shown that the best result is achieved with the Correlators.

Since the frequency components of the Bumblebee are well known, it is possible to estimate the phase and the amplitude of these with Correlators. This is done by correlating the microphone signal with sinusoids having the same frequencies as the Bumblebee, hence the name Correlators. By generating sinusoids with these phase and amplitude estimates, and then subtracting them from the microphone signal, the Bumblebee is suppressed. Since it is subtracted in the time domain, it is also possible to consider a recurring pause in the interference caused by the so called idle frame in the transmission.

# Acknowledgments

We would like to thank everyone who has helped us carry out this Master thesis. We would especially like to thank our supervisor at the department of applied signal processing at the University of Karlskrona/Ronneby Professor Ingvar Claesson for his support, generous contribution of knowledge, and his constructive criticism. We would also like to express our gratitude to our supervisors at the department of Voice & Audio Technology at Ericsson Mobile Communications in Lund, Per Ljungberg, Johan Udén and Gabor Lakatos who have given us a lot of guidance, ideas and support. Especially, we are grateful to Per Ljungberg for the help he has given us with the report.

Further, we also wish to express our appreciation to the whole department of Voice & Audio Technology at Ericsson Mobile Communications in Lund, including the other students, for both help and nice companionship.

Finally, we would like to thank our parents, brothers and sisters, relatives and all our friends and lovers for putting up with us during our years of study.

# Contents

# Sound files

These sound files can be found by looking up the *Bumblebee killer* report at http://www5.hk-r.se/fou/. All sound files except [S1] are saved before and after.

[S1] Interfering signal, *Bumblebee*, recorded in a silent room.

[S2] Cancellation of the Bumblebee with Notch filter. No speech.

[S3] Cancellation of the Bumblebee with Notch filter in speech.

[S4] Cancellation of the Bumblebee with Correlators where the idle frame has been considered. No speech.

[S5] Cancellation of the Bumblebee with Correlators in speech where VAD and idle have been considered.

[S6] Cancellation of the Bumblebee with Correlators in the Half Rate case where the idle frame has been considered. No speech.

[S7] Cancellation of the Bumblebee with Correlators in speech in the Half Rate case where VAD and idle have been considered.

[S8] Cancellation of the Bumblebee with interpolation using a 256 FFT. No speech.

[S9] Cancellation of the Bumblebee with interpolation in speech using a 256 FFT where VAD has been considered.

# Acronyms

| | |
|---|---|
| $\mu$C | Micro Controller |
| BTS | Base Transceiver Station |
| DAI | Digital Audio Interface |
| Downlink | Transmission from the BTS to the MS |
| DSP | Digital Signal Processor |
| DTX | Discontinuous transmission |
| EFR | Enhanced Full Rate |
| FACCH | Fast Associated Control CHannel |
| FFT | Fast Fourier Transform |
| FN | Frame Number |
| FR | Full Rate |
| GMSK | Gaussian Minimum Shift Keying |
| GSM | Global System for Mobile communication |
| HR | Half Rate |
| IFFT | Inverse Fast Fourier Transform |
| MAC | Multiply ACcumulate operation |
| MACS | Multiply ACcumulates per Second |
| MIPS | Millions of Instructions Per Second |
| MOPS | Millions of Operations Per Second |
| MS | Mobile Station |
| PCM | Pulse Code Modulation |

| | |
|---|---|
| ROM | Read Only Memory |
| RSS | Radio SubSystem |
| SACCH | Slow Associated Control CHannel |
| SDCCH | Stand alone Dedicated Control CHannel |
| SDR | Signal-Distortion-Ratio |
| SID | SIlence Descriptor |
| SNR | Signal-to-Noise-Ratio |
| TCH | Traffic Channel |
| TDMA | Time Division Multiple Access |
| TN | Time slot Number |
| Uplink | Transmission from the MS to the BTS |
| VAD | Voice Activity Detector |
| WGN | White Gaussian Noise |

# Chapter 1

# Introduction

In GSM mobile stations it is a common problem that an interfering signal is introduced in the microphone signal when the mobile station is transmitting. This interfering signal is transmitted along with the speech signal to the receiver. Due to the sound of the interfering signal it is commonly denoted the *Bumblebee*, which will be used throughout this report [S1].

The Bumblebee is generated by the switching nature of TDMA cellular telephony, where the radio circuits are switched on and off. During the time the radio is switched on, denoted a time slot, the Mobile Station transmits its information by sending electro magnetic impulses. These impulses are induced in the microphone path and generate the interference. The switching rate is approximately 217 Hz, according to the GSM standard [5].

Depending on the power level the mobile station is transmitting, how it is held and if one uses portable hands-free equipment or not, the amplitude will fluctuate. When the mobile station changes time slot, i.e., during a hand over, the phase will also change.

Earlier solutions of this problem have been different hardware constructions, i.e., different placement of the components, usage of special microphones, reconstruction of analog parts, etc. However, this is expensive, time demanding and becomes harder when the mobile stations become smaller.

The presented solution to the problem in this report is to make use of the knowledge that the Bumblebee disturbance, after a Fourier series expansion, consists of a sum of sinusoids with accurately well known frequencies. Different software solutions to attenuate these frequency components are evaluated. It is shown that the best result is achieved by estimating the phase and the amplitude of the different sinusoids with correlators, and then subtract these sinusoid estimates from the microphone signal.

# Chapter 2

# Problem

In this chapter a detailed study of the interfering signal is presented. Both theoretical and practical aspects are considered.

Some knowledge of GSM is assumed. The reader not familiar with GSM is referred to Appendix A.

## 2.1 Theoretical outline

The Bumblebee is a result of the transmitting technique that is used in GSM, i.e., Time Division Multiple Access (TDMA). The mobile station (MS) is sending its information during a time slot that it is assigned. Eight time slots make one TDMA-frame, where the time slots are numbered $0 - 7$. The MS is using the same time slot in every TDMA-frame until the network orders it to another time slot, i.e., a handover. The duration of a time slot is $3/5200$ seconds, and the periodicity of the TDMA-frames is $8 \cdot 3/5200$ seconds. During the assigned time slot the MS is transmitting its information by sending electromagnetic impulses. These impulses are induced in the analog microphone path and produce a very annoying periodic interference in the uplink speech. The period is $1/(8 \cdot (3/5200)) \approx 217$ Hz in Full Rate (FR), see Section A.2.2.

### 2.1.1 Special case - Half Rate Transmission

There is a special case that is not so common but still interesting to evaluate, Half Rate transmission (HR). This case is described in section A.2.2 where it can be seen that the radio access pattern differs from FR. The period of the interference in this case is $1/(8 \cdot 2 \cdot (3/5200)) \approx 108$ Hz which is half the frequency of the FR, since the MS is only transmitting during every other time slot.

### 2.1.2 Special case - Discontinuous Transmission

In some mobile networks a feature called Discontinuous Transmission (DTX) is supported. DTX is a mechanism which allows the radio transmitter to be switched off most of the time during speech pauses. During these pauses the background noise is averaged and sent to the receiver as Silent Descriptor frames, SID frames, see Section A.3. The SID frames contain no Bumblebee because of the averaging, consequently the algorithm does not have to run during DTX.

## 2.2 Analysis of the Bumblebee

A typical recorded Bumblebee signal in a silent room can be seen in Figure 2.1. The interfering signal is periodic but not simple since, in the FR case,

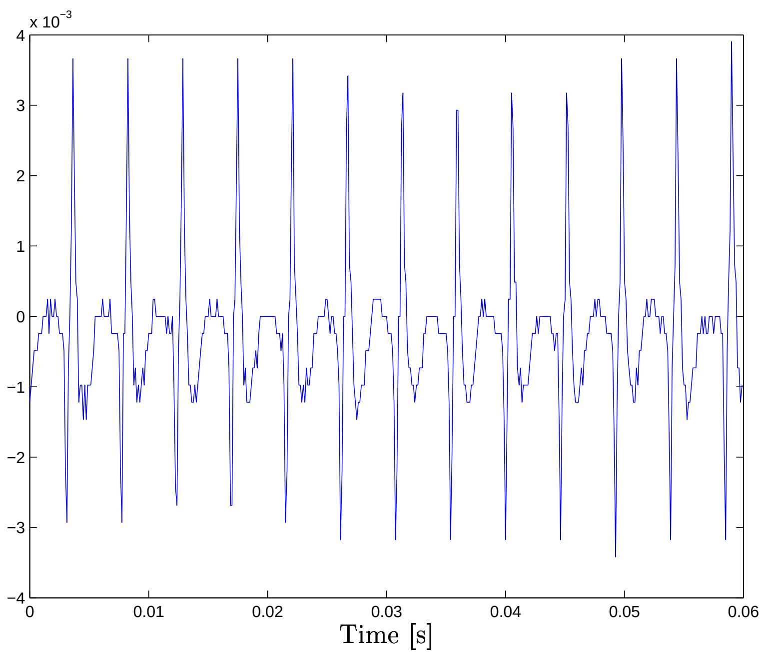



Figure 2.1: *Interfering signal recorded in a silent room.*

the absence of transmission when the MS is listening to other base stations. This occurs once every 26 TDMA-frames and is denoted *idle* frame. The idle frame is illustrated in Figure 2.2. Neither in the HR case the Bumblebee is simple and periodic although for a different reason, see Section A.2.2. Two different HR channels, even and odd, are used depending on which frames that are used for traffic, see Figure A.8. This results in two different radio access patterns. The radio access pattern for an even HR channel can be seen in Figure 2.3. The idle frame is necessary to consider when eliminating

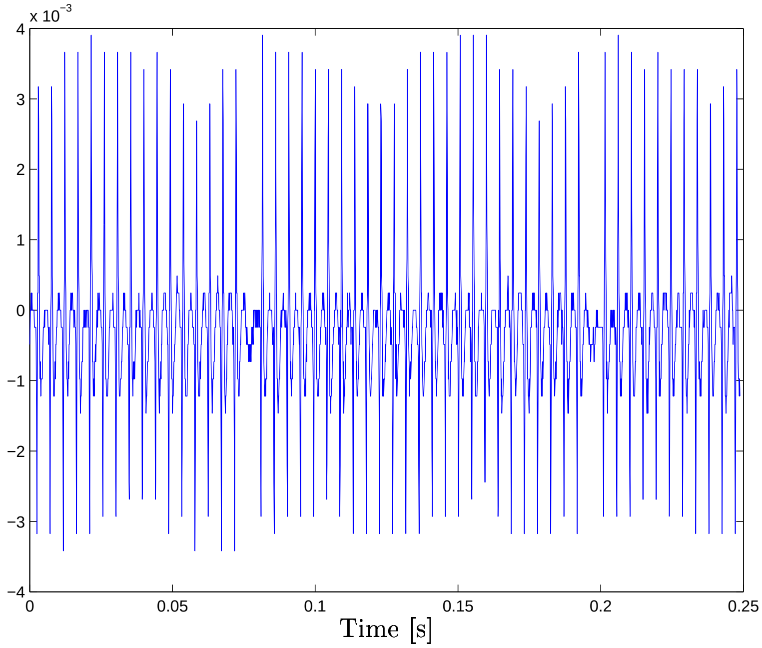



Figure 2.2: *Interfering signal recorded in a silent room,* **Full Rate.**

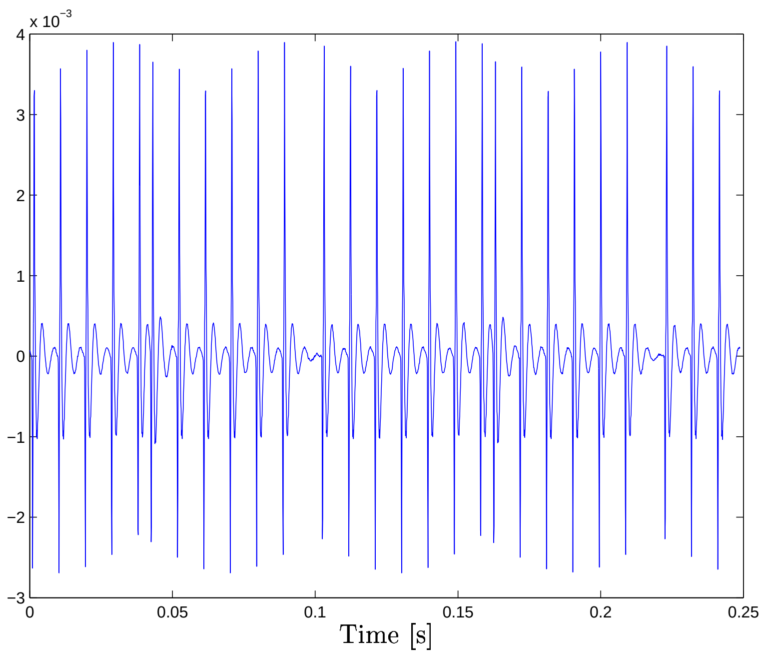



Figure 2.3: *Interfering signal recorded in a silent room,* **Half Rate.**

the interference.

### 2.2.1 Fourier series expansion of the Bumblebee

Since the Bumblebee is periodic, it can be viewed as a Fourier series expansion, see appendix B.

$$x(n) = \sum_{k=1}^{K} A_k \sin(2\pi k \left(\frac{f_0}{f_s}\right) n + \Phi_k), \quad K = \infty \tag{2.1}$$

In equation 2.1 $K-1$ is the number of harmonics, $f_s$ is the sample frequency, and $f_0$ is the frequency of the fundamental tone. As previously mentioned this frequency is approximately 217 Hz.

One period of the Bumblebee, for FR, is simplified as $x_{simp.}(t)$ shown in Figure 2.4. The Fourier series coefficients of the simplified Bumblebee are

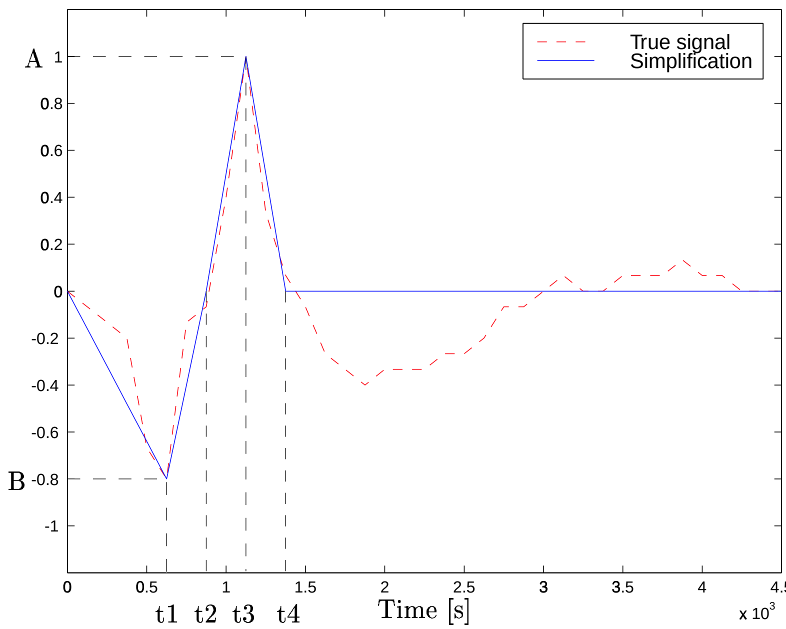



Figure 2.4: *Simplification, $x_{simp.}(t)$, of one period of the Bumblebee.*

calculated, in continuous time, as

$$C_k = \frac{1}{T_0} \int_0^{T_0} x_{simp.}(t) e^{-jk\omega_0 t} \, dt. \tag{2.2}$$

This will result in

$$
\begin{aligned}
C_k &= \frac{1}{T_0(k\omega_0)^2}\left[\frac{B}{t_1} - \frac{2B}{t_1}e^{-jk\omega_0 t_1} + \left(\frac{B}{t_2 - t_1} - \frac{A}{t_3 - t_2}\right)e^{-jk\omega_0 t_2}\right. \\
&\quad \left. + \frac{2A}{t_3 - t_2}e^{-jk\omega_0 t_3} - \frac{A}{t_4 - t_3}e^{-jk\omega_0 t_4}\right],
\end{aligned} \tag{2.3}
$$

where A, B and $t_1$ to $t_4$ are the values of the simplified Bumblebee, $x_{simp.}(t)$, which can be seen in Figure 2.4.

After calculation of the Fourier series coefficients, $\hat{x}(t)$ can be written as a sum of sinusoids

$$
\begin{aligned}
x_{simp.}(t) &= C_0 + \sum_{k=1}^{\infty} |C_k| \cos\left(k\omega_0 t + \arg(C_k)\right) \\
&= C_0 + \sum_{k=1}^{\infty} |C_k| \sin\left(k\omega_0 t + (\arg(C_k) + \frac{\pi}{2})\right).
\end{aligned} \tag{2.4}
$$

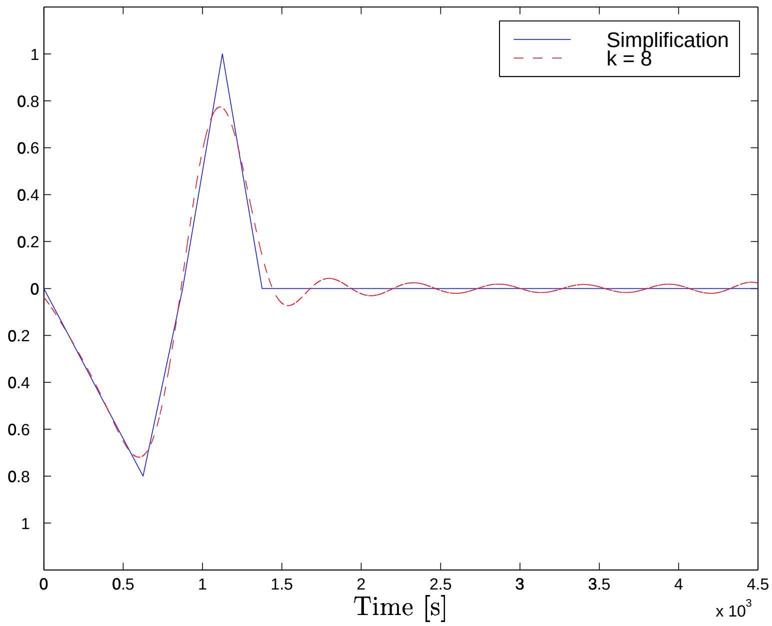
The result of using Equation 2.4 to generate an approximation, $\hat{x}_{simp.}(t)$ of the simplified Bumblebee $x_{simp.}(t)$ is shown in Figure 2.5 where two different values of $k$ have been used.

It can be seen in Equation 2.3 that the harmonics will decay as $1/f^2$. Because of this maybe not all of the harmonics will necessarily have to be eliminated to get a satisfying result.

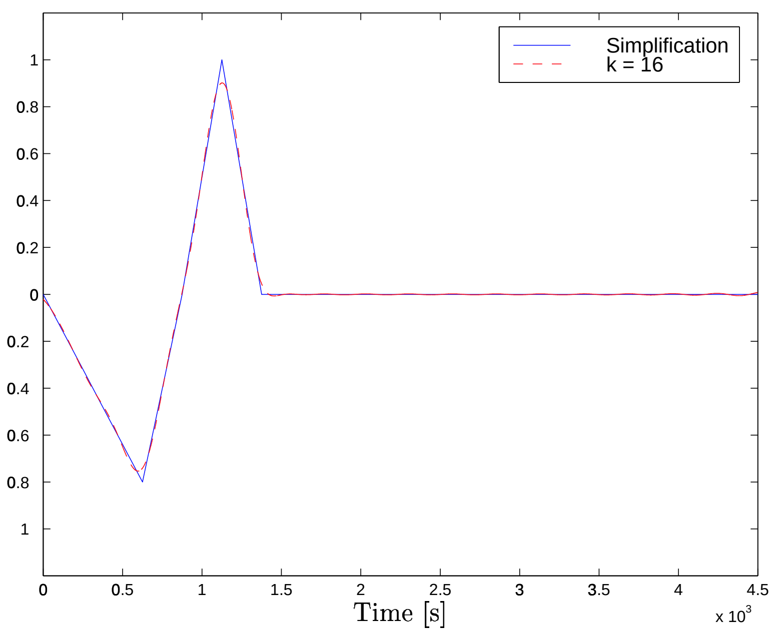
In Figure 2.6 the absolute value of $C_k$ is shown. Comparing this figure with the frequency characteristic of the Bumblebee in Figure 2.7 it can be seen that the frequency characteristic of the approximation, $\hat{x}_{simp.}(t)$, is close to the real signal. This shows that the Bumblebee can be seen as a sum of sinusoids, which can be used when the Bumblebee shall be eliminated.

The number of harmonics $K$ that need to be represented depends on the sampling rate of the signal, which is 8 kHz. Consequently, the interfering signal will after sampling only consist of frequencies below 4 kHz if aliasing is avoided. The filters in connection with the A/D converter and the speech coder band limit the Bumblebee even more to approximately $300 - 3400$ Hz. Hence, the fundamental tone and the 15:th harmonic will be severely attenuated. In Figure 2.7 this can be seen, but since the signal is recorded before the speech coder, it is not fully attenuated.

A similar Fourier series expansion can be done for the HR case but is omitted in this thesis. This, however, would show that the fundamental frequency, $f_0$, equals approximately 108.5 Hz with harmonics at $n \cdot f_0$.




(a)




(b)

Figure 2.5: $x_{simp.}(t)$ *approximated with, for (a) k=8 and for (b) k=16, sinusoids*

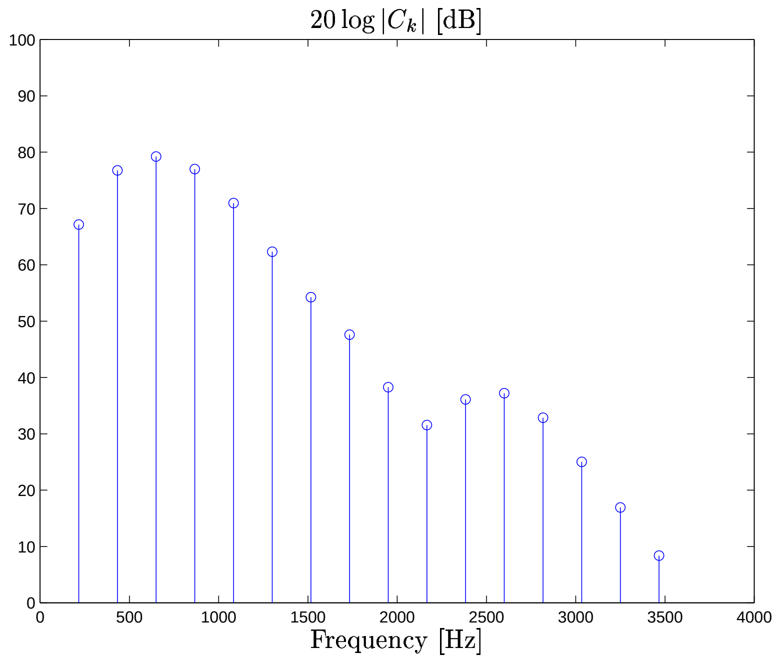



Figure 2.6: *Absolute value of the $C_k$ coefficients in log scale.*

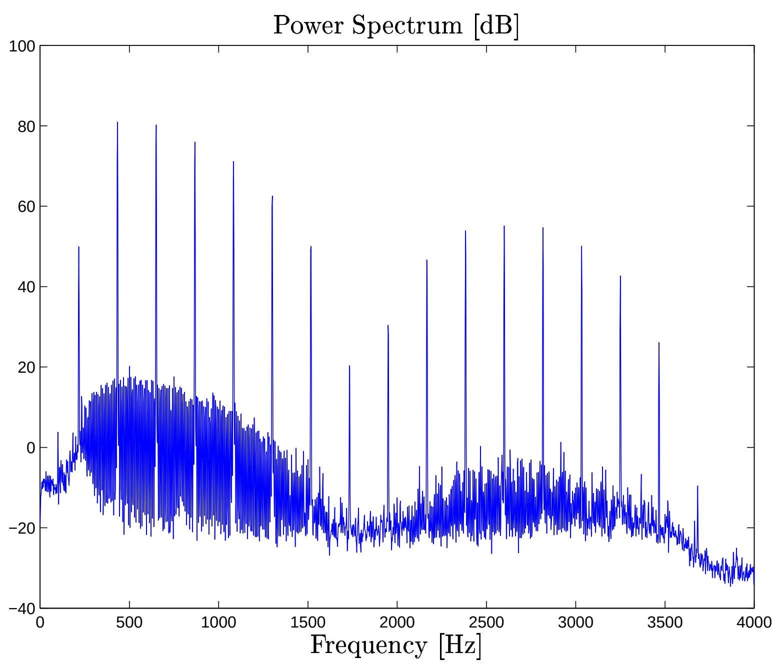



Figure 2.7: *Spectrum of the Bumblebee plus random noisy background.*

### 2.2.2 Property of the Bumblebee

If it shall be possible to subtract an estimate of the Bumblebee it is necessary that it is additive. To test this hypothesis, recordings have been made on the interfering signal along with different levels of white Gaussian noise (WGN). These tests show that the interfering signal is added onto the WGN, see Figure 2.8. Observe that the spectrum seems to increase linearly along with the WGN power.

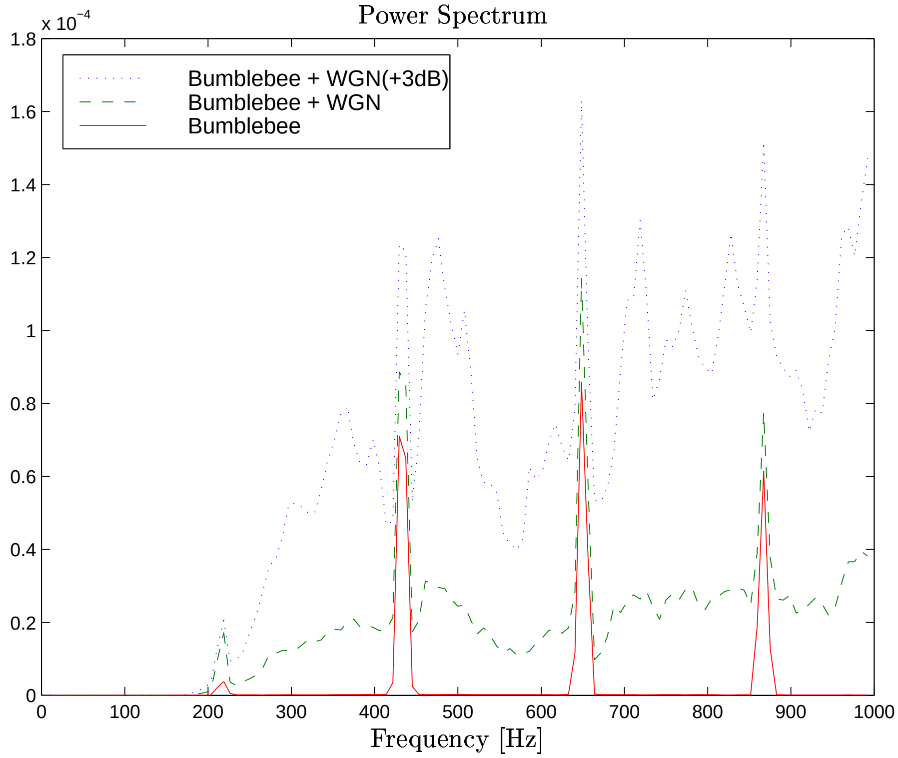



Figure 2.8: *Interfering signal with two different levels of WGN shows that the interfering signal is additive.*

# Chapter 3

# Possible solutions

Four different methods to eliminate the Bumblebee have been evaluated. These methods are Notch filter, Correlators, Matched filter and Interpolation, which are explained in this chapter.

## 3.1 Notch filter

A notch filter consists of a number of deep notches, or ideally nulls, in its frequency response, see Figure 3.1. Such a filter is useful when specific frequency components of known frequencies must be eliminated. To eliminate the frequencies at $(\omega_n, n = [1, \ldots, N])$ a pair of complex-conjugated zeros are placed on the unit circle at the angle $\omega_n$,

$$z_{n_{1,2}} = r_b e^{\pm j\omega_n}, \qquad r_b = 1 \tag{3.1}$$

This results in an FIR notch filter with the system function

$$H(z) = B(z) = b_o \prod_{n=1}^{N} (1 - r_b e^{j\omega_n} z^{-1})(1 - r_b e^{-j\omega_n} z^{-1}). \tag{3.2}$$

The $b_0$ constant is chosen as

$$b_0 = \frac{1}{\sum_{n=1}^{N} b_n} \tag{3.3}$$

to avoid an amplification between the notches. However, these filter notches have a relatively large bandwidth, which means that the frequencies around $\omega_n$ will also be relatively much attenuated, see Figure 3.1. To reduce the bandwidth of the notches, poles are placed at the same angle as the zeros, but with a slightly smaller magnitude, see Figure 3.2. Thus the position of the poles are

$$p_{n_{1,2}} = r_a e^{\pm j\omega_n}, \qquad 0 \ll r_a < r_b. \tag{3.4}$$

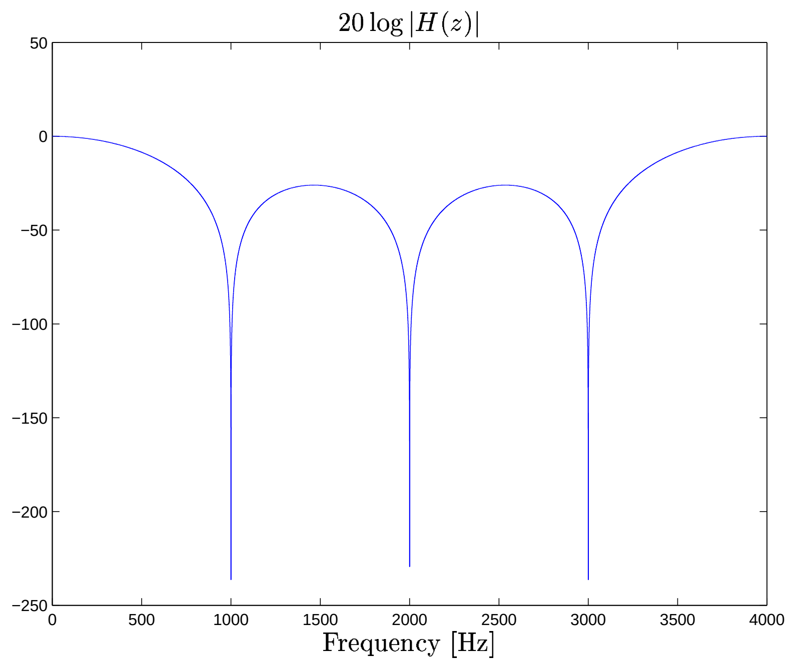



Figure 3.1: *Frequency response of an FIR notch filter with* $r_b = 1,\ N = 3$ *and* $\omega_n = n \cdot 2\pi \cdot (1000/8000)$

Consequently, the system function of the resulting notch filter is

$$H(z) = \frac{B(z)}{A(z)} = b_o \prod_{n=1}^{N} \frac{(1 - r_b e^{j\omega_n} z^{-1})(1 - r_b e^{-j\omega_n} z^{-1})}{(1 - r_a e^{j\omega_n} z^{-1})(1 - r_a e^{-j\omega_n} z^{-1})} \tag{3.5}$$

where

$$b_0 = \frac{\sum_{n=1}^{N} a_n}{\sum_{n=1}^{N} b_n}. \tag{3.6}$$

The frequency response of the filter in Equation (3.5) is plotted in Figure 3.3.

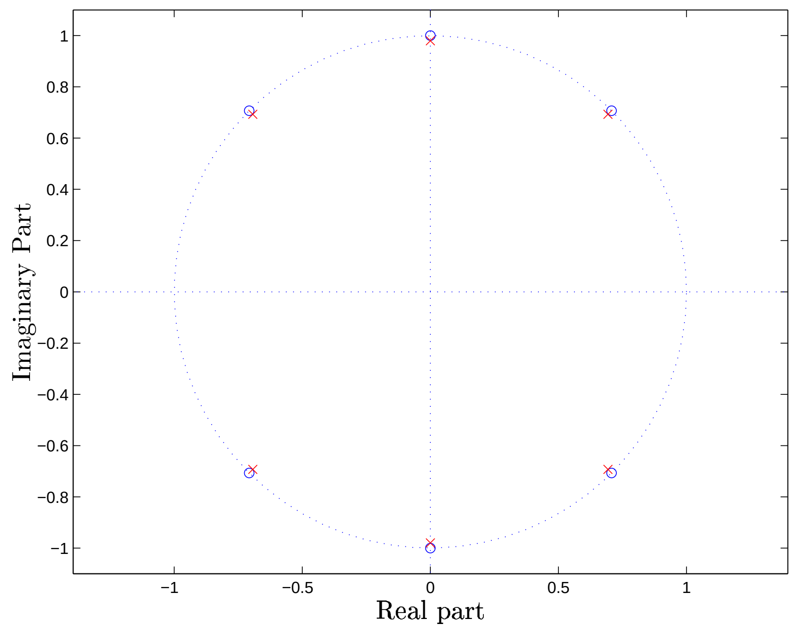



Figure 3.2: *Zero-pole plot of a notch filter with* $r_b = 1,\ r_a = 0.98,\ N = 3$ *and* $\omega_n = n \cdot 2\pi \cdot (1000/8000)$

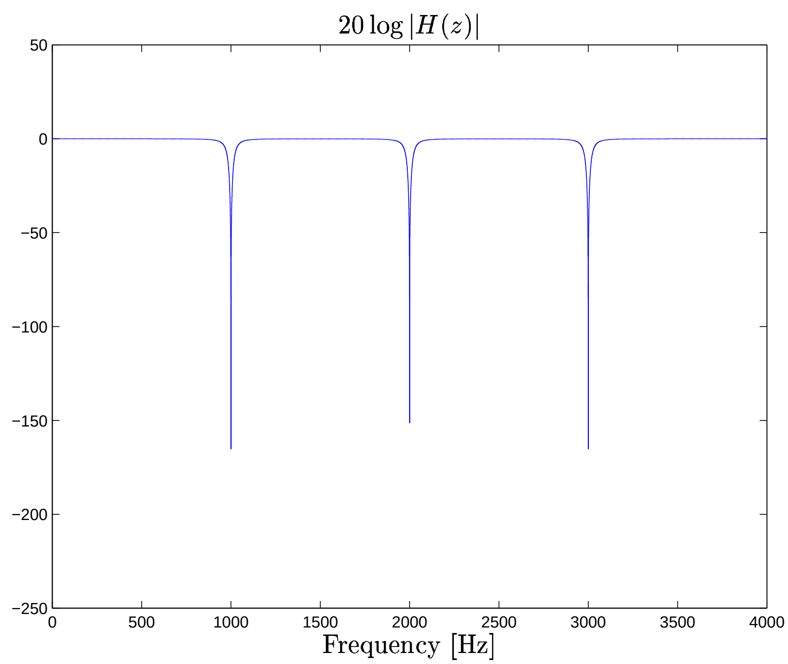



Figure 3.3: *Frequency response of a notch filter with* $r_b = 1,\ r_a = 0.98,\ N = 3$ *and* $\omega_n = n \cdot 2\pi \cdot (1000/8000)$

## 3.2 Orthogonal Correlators

In this evaluation of the correlators, continuous time is used, since later a parallel will be drawn between correlators and matched filters that requires continuous time.

If correlators are used to detect a signal in noise the scheme is shown in Figure 3.4. It consists of a bank of N product-integrators, or correlators.

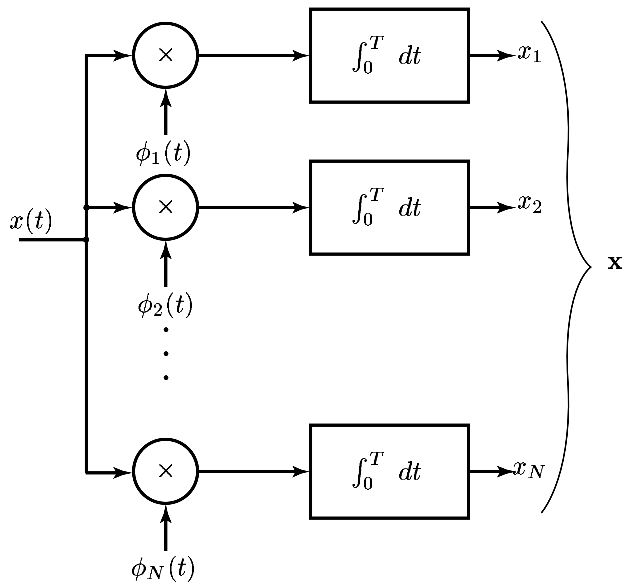



Figure 3.4: *Detector with correlators*

They are supplied with a corresponding set of reference signals or orthonormal basis functions $\phi_1(t)$, $\phi_2(t)...\phi_N(t)$ that are generated locally. This bank of correlators operate on the received signal $x(t)$, $0 \leq t \leq T$, to produce the observation vector $\mathbf{x} = [x_1, x_2, ..., x_N]$ where

$$\int_0^T x(t)\phi_n \, dt = x_n. \tag{3.7}$$

The observation vector will represent the received signal in the space that is defined by the orthonormal basis functions $\phi_1(t)$, $\phi_2(t)...\phi_N(t)$.

### 3.2.1 Estimation of Phase and Amplitude with correlators

If the desired signal equals

$$C\sin(2\pi f_i t + \theta) \qquad , 0 \leq t \leq T, \tag{3.8}$$

where C is the unknown amplitude, and $\theta$ is the unknown phase that is considered to be a random variable uniformly distributed between 0 and $2\pi$

radius. The received signal will be $x(t)$, which is the desired signal added with noise,

$$x(t) = C\sin(2\pi f_i t + \theta) + w(t) \qquad ,0 \leq t \leq T, \tag{3.9}$$

where $\omega(t)$ is the sample function of a white Gaussian noise process of zero mean and power density $\mathcal{N}_0/2$. By using the trigonometric identity $x(t)$ can be rewritten in the expanded form

$$x(t) = C\sin(2\pi f_i t)\cos(\theta) + C\cos(2\pi f_i t)\sin(\theta) + w(t). \tag{3.10}$$

The received signal $x(t)$ is applied to a pair of orthonormal correlators; where one correlator is supplied with the reference signal $\frac{2}{T}\sin(2\pi f_i t)$, and the other is supplied with the reference signal $\frac{2}{T}\cos(2\pi f_i t)$. For both correlators, the observation interval is $0 \leq t \leq T$. In the absence of noise the first correlator output equals $C\cos(\theta)$ and the second correlator output equals $C\sin(\theta)$. The dependence on the unknown phase $\theta$ may be removed by summing the squares of the two correlator outputs, and then taking the square root of the sum. Thus the result of these operations are $C$, which is independent of the unknown phase $\theta$.

$$\sqrt{(C\cos(\theta))^2 + (C\sin(\theta))^2} = \hat{C} \tag{3.11}$$

It is also possible to get an estimate of $\theta$ by solving

$$\arg\left(C\sin(\theta), C\cos(\theta)\right) = \hat{\theta}. \tag{3.12}$$

These solutions is the result of maximizing the likelihood function [4][6]. Figure 3.5 shows the model with blocks for estimation.

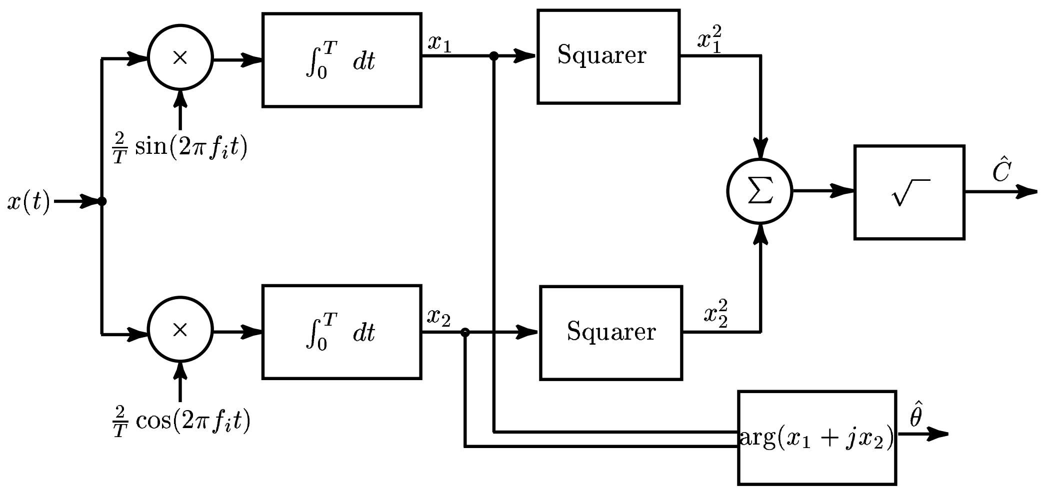



Figure 3.5: *Detection and estimation with correlators where* $x_1 = C\cos(\theta)$ *and* $x_2 = C\sin(\theta)$

## 3.3 Matched filter

Another way to find a signal with known form in noise is to use a filter that maximizes the output-signal-to-noise ratio. In the case where different symbols are transmitted with the symbol rate $1/T$ the filter output is sampled at $t = T$. If the input consists of a sum of a Fourier-transformable deterministic signal $x(t)$ of known form and continuous noise $n(t)$, the output signal and noise is denoted by $x_o(t)$ and $n_o(t)$.Then the output-signal-noise rate that should be maximized is

$$\left(\frac{\hat{S}_o}{N_o}\right) = \frac{|x_o(t_0)|^2}{E[n_o^2(t)]} \tag{3.13}$$

where

$$\hat{S}_o = |x_o(t_o)|^2 \tag{3.14}$$

is the output signal power at time $t_o$ and

$$N_o = E[n_o^2(t)] \tag{3.15}$$

is the output average noise power [2].

If $H(\omega)$ defines the transfer function of the system, the output signal $x_0(t)$ at any time $t$ is

$$x_o(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(\omega)H(\omega)e^{j\omega t}d\omega. \tag{3.16}$$

The output average noise power can be written in the form

$$N_o = E[n_o^2(t)] = \frac{1}{2\pi}\int_{-\infty}^{\infty} \Phi_{nn}(\omega)|H(\omega)|^2 d\omega, \tag{3.17}$$

where $\Phi_{nn}(\omega)$ is the power density spectrum of the random process, denoted $N(t)$, that represents the input noise $n(t)$. By use of 3.16 at time $t_o$ and 3.17, 3.13 can be written as

$$\left(\frac{\hat{S}_o}{N_o}\right) = \frac{\left|\frac{1}{2\pi}\int_{-\infty}^{\infty} X(\omega)H(\omega)e^{j\omega t}d\omega\right|^2}{\frac{1}{2\pi}\int_{-\infty}^{\infty} \Phi_{nn}(\omega)\left|H(\omega)\right|^2 d\omega}. \tag{3.18}$$

To find the optimum filter, $H(\omega)$, that maximizes 3.18 Schwartz inequality is used [3]. If the input noise is white with power density $\mathcal{N}_o/2$, this results in the optimum filter

$$H_{opt} = Kx^*(\omega)e^{-jwt_o} \tag{3.19}$$

where $K = 1/\pi S\mathcal{N}_o$ is an arbitrary constant, and where $S$ is an arbitrary constant scaling factor resulting from Schwartz inequality. Equation 3.19

specifies the matched filter in the frequency domain. The impulse response denoted $h_{opt}(t)$ of the optimum filter is the inverse Fourier transform of $H_{opt}(\omega)$. From 3.19, it is found that

$$h_{opt} = Kx^*(t_0 - t) \tag{3.20}$$

which for a real signal $x(t)$, reduces to

$$h_{opt} = Kx(t_0 - t). \tag{3.21}$$

Equation 3.21 shows that the impulse response of the optimum filter is a time reversed and delayed version of the input signal $x(t)$.

Consider, for example, a matched filter with the impulse response $h$ and with the received signal $x(t)$ used as the filter input. The resulting filter output $y_j(t)$ is then defined by the convolution integral:

$$y_j(t) = \int_{-\infty}^{\infty} x(\tau)h_j(t-\tau)d\tau \tag{3.22}$$

If the impulse response is set to

$$h_j(t) = \phi_j(T - t), \tag{3.23}$$

the resulting filter output is

$$y_j(t) = \int_{-\infty}^{\infty} x(\tau)\phi_j(T - t + \tau)d\tau. \tag{3.24}$$

Sampling this output at time $t = T$ results in

$$y_j(T) = \int_{-\infty}^{\infty} x(\tau)\phi_j(\tau)d\tau. \tag{3.25}$$

and since $\phi_j(t)$ is zero outside the interval $0 \leq t \leq T$

$$y_j(T) = \int_{0}^{T} x(\tau)\phi_j(\tau)d\tau. \tag{3.26}$$

When $y_j(T) = x_j$ this method is equivalent to using correlators. This can be seen in Figure 3.4, where $x_j$ is the the $j$th correlator output produced by the received signal $x(t)$. The implementation of matched filters is shown in Figure 3.6.

### 3.3.1 Estimation of Phase and Amplitude with matched filter

The estimation of phase and amplitude can, because of the result above, be done in the same way as for the correlators in section 3.2.1 that are illustrated in Figure 3.7.

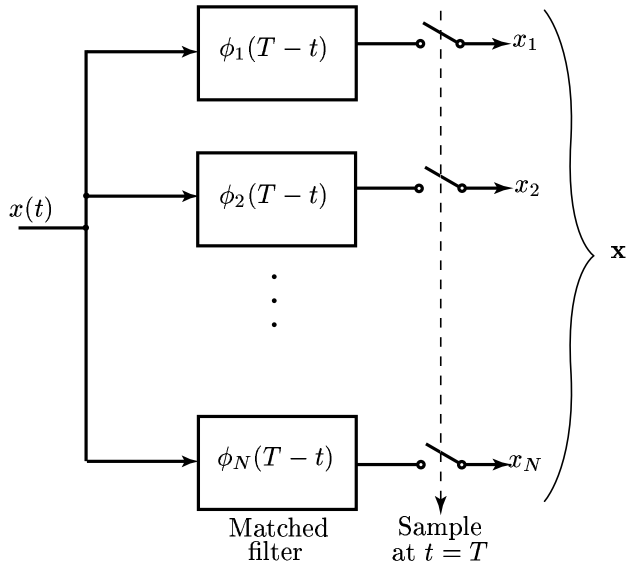



Figure 3.6: *Detector with matched filter.*

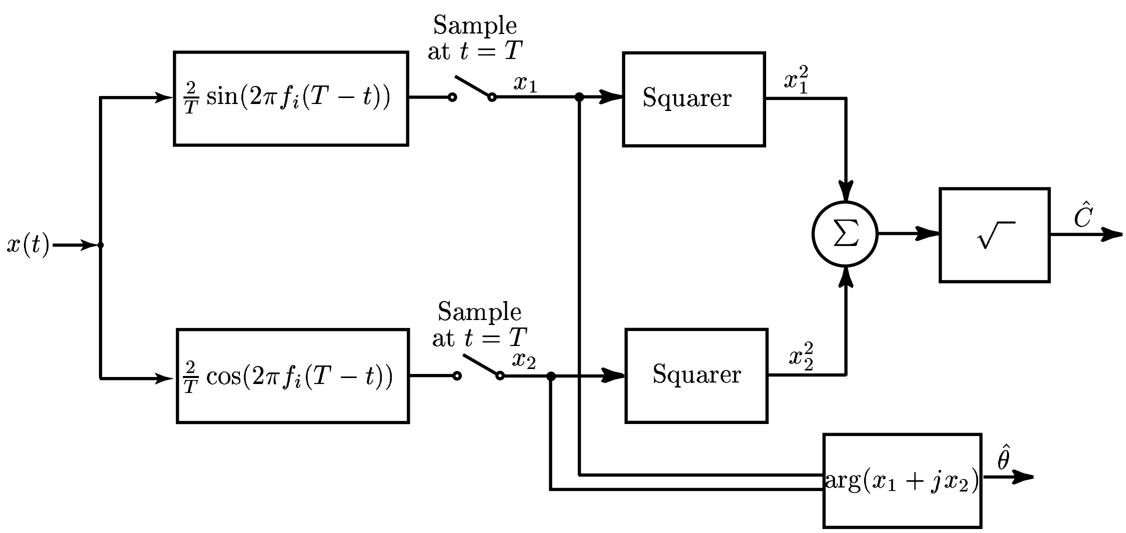



Figure 3.7: *Detection and estimation with matched filter.*

## 3.4 Interpolation in frequency domain

In the evaluation of the Interpolation discrete time is used again. Assume that the problem can be seen as a signal with an interference, where the interference is a sum of sinusoids with known frequencies. If a Fast Fourier Transform (FFT) is performed on the signal, it is possible to suppress the interference in the frequency domain. The suppression of the FFT bins that contain the interfering frequency components is done by subtracting the difference between the FFT bin in point and the mean of the two neighboring FFT bins. This gives an approximation of how much the interference adds to the specific bin. To decrease the variance, of the estimated amount to subtract with, the difference can be averaged over an arbitrary number of data blocks to avoid musical artefacts. By performing an Inverse Fast Fourier Transform (IFFT) the signal is transformed back to the time domain.

# Chapter 4

# Simulations

In Chapter 3 four different methods to eliminate the bumblebee were presented: Notch filter, Correlators, Matched filter and Interpolation. It was shown in Section 3.3 that Correlators and Matched filters, for this application, gave the same result. Because of this only correlators, of those two, has been evaluated.

In the following simulations, data recorded from the Digital Audio Interface (DAI) in an Ericsson MS have been employed. The DAI is the interface after the A/D-converter where the signal is Pulse Code Modulated. This is the signal that enters the DSP, where the algorithm is supposed to be implemented.

The problem with the Bumblebee is not just to eliminate it, but eliminate it without degrading the speech quality. This makes it difficult to find a fair expression of the enhancement.

The frequencies that will be attenuated in the tests are:

$$k \cdot \omega_0, \qquad k = [1, \ldots, K] \tag{4.1}$$

where $K = 16$ and

$$\omega_0 = 2\pi \frac{1}{8 \cdot \frac{3}{5200} \cdot 8000} \approx 2\pi \cdot 217 \text{ rad/s.}$$

$\omega_0$ is the fundamental tone of the Bumblebee, see section 2.2.1. With $K = 16$ the fundamental tone and 15 of its harmonics will be eliminated. This will cover a range up to 3467 Hz which is the total frequency range of the audio band (see section 2.2.1).

## 4.1 Notch filter

Since it is well known which frequencies the Bumblebee consists of, it is possible to use a notch filter (see Chapter 3.1) to reduce this interference.

### 4.1.1 Implementation

The notches are made as deep as possible to, in the ideal case, totally eliminate the frequencies in question, which results in the following system function:

$$H(z) = \frac{B(z)}{A(z)} = \frac{\sum_{k=1}^{16} a_k}{\sum_{k=1}^{16} b_k} \prod_{k=1}^{16} \frac{(1 - r_b e^{jk\omega_0} z^{-1})(1 - r_b e^{-jk\omega_0} z^{-1})}{(1 - r_a e^{jk\omega_0} z^{-1})(1 - r_a e^{-jk\omega_0} z^{-1})} \quad (4.2)$$

The simulations are done recursively on the whole data set. This will result in a convergence period at the start up and when a handover occurs, because of the phase shift.

### 4.1.2 Tests

The first test was performed on data recorded in a silent room. This case is shown in Figure 4.1 [S2]. The second test is made on data with speech and

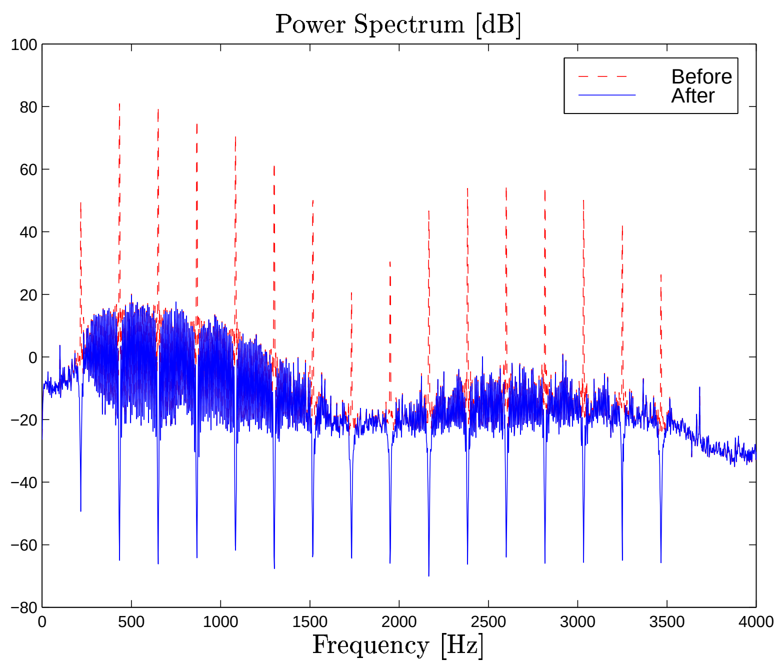



Figure 4.1: *Cancellation of the Bumblebee with* **notch filter**. *The Bumblebee was recorded in a silent room* [S2]. **Full Rate, no speech**.

is shown in Figure 4.2 [S3].

### 4.1.3 Evaluation

It can be seen that the Bumblebee is severely attenuated. However, this solution does not give a satisfying result because too much of the speech is

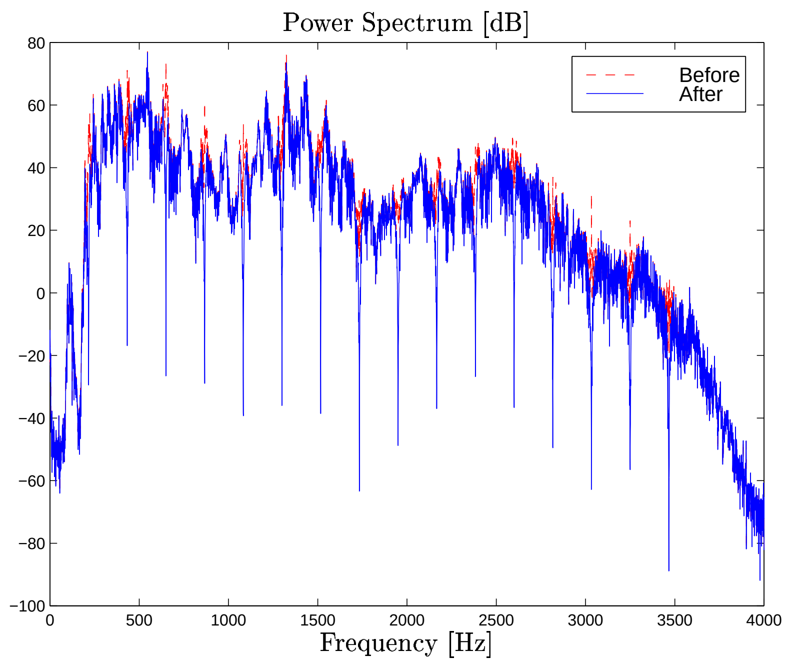



Figure 4.2: *Cancellation of the Bumblebee with* **notch filter** *in speech* [S3]. **Full Rate, with speech**.

also attenuated, which results in a metallic sound. This can be seen in Figure 4.2. Another problem with this solution is that the periodic idle frame can not be considered and therefore results in a new periodic interference, see Figure 4.3. The reason for this is that the Notch filter consists of poles which give a feedback of the output signal ($y(t)$). Consequently, the Bumblebee is added during the idle frame.

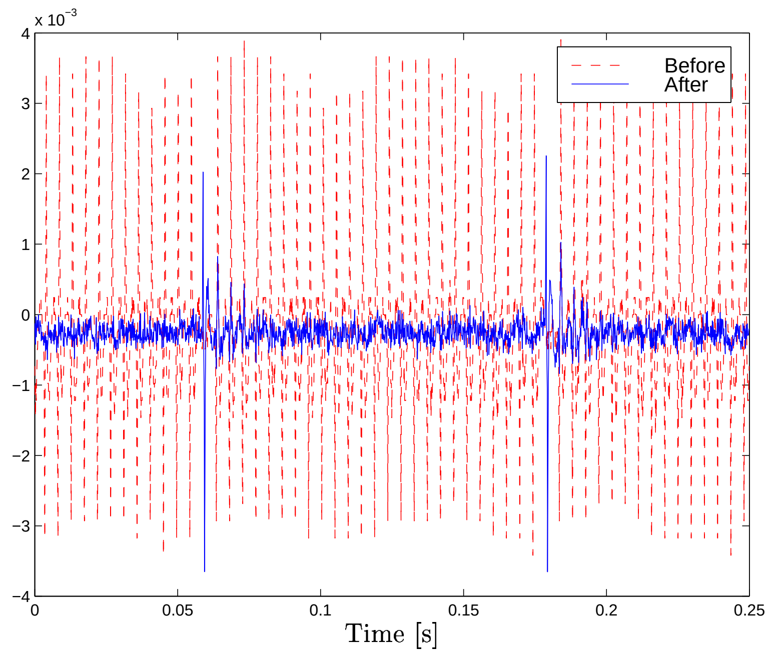



Figure 4.3: *Time signal of the notched Bumblebee.*

## 4.2 Correlators

As was shown in Section 2.2.1, the Bumblebee can be seen as a sum of sinusoids. This makes it possible to subtract the Bumblebee in time. To do this the phase and amplitude of the different sinusoids must be estimated. This estimation is done block-wise with correlators.

### 4.2.1 Implementation

The correlators were explained in continuous time in Section 3.2. To implement this technique in a DSP in discrete time, the received signal, corresponding to Equation 3.9, becomes

$$x(n) = \sum_{k=1}^{\infty} C_k \sin(nk\omega_0 + \theta_k) + w(n). \tag{4.3}$$

The the bank of correlators becomes

$$\phi_k(n) = \frac{2}{T} \sin(nk\omega_0) \tag{4.4}$$

and

$$\varphi_k(n) = \frac{2}{T} \cos(nk\omega_0). \tag{4.5}$$

This makes it possible to, in discrete time, correlate the received signal with the basis functions of the correlators. Then the estimates of the phase and the amplitude are calculated in the same way as in Section 3.2. These estimates are subsequently used to generate sinusoids that are subtracted from the received signal.

#### Voice Activity Detector (VAD)

If the estimation is done during speech, the estimate of the Bumblebee will be incorrect, since the speech contains energy at the same frequencies as the Bumblebee. This is solved by only estimating during speech pauses, which requires a Voice Activity Detector (VAD). There is already a VAD in the MS, which can be used but in these tests this has been done by hand.

#### Consideration of the idle frame

The idle frame is also considered. This is done by not subtracting during idle, which is possible because it is known when the idle frame occurs. In the MS this is done by using the knowledge about the structure of the frames that is shown in Figure A.8. The MS Micro Controller ($\mu$C) is sending information (code commands) to the DSP about how to speech code the 160 last received PCM samples. The MS $\mu$C then requires a reply from the DSP containing the speech coded data. The code commands arrive to the DSP on the average every 20 ms. However, the exact time arrivals follow the pattern (18.465 ms, 18.465 ms, 23.070 ms). In order for the DSP to synchronize its PCM buffers properly the code commands contain information, the "syncInfo", on the time to the next code command. This information can take six different values, and is carried in the code commands in a three bit field. The six different values of syncInfo correspond to certain positions in the 120 ms multiframe structure (26-multiframe). Thus, given the "syncInfo" information, it is possible to calculate the position of the idle frame. After the reception of a code command with the correct "syncInfo" the PCM samples that enter the DSP are counted. Since it is known when the idle frame occurs with respect to a certain "syncInfo" the samples that are received during the idle frame are marked.

### Block size decision

Another thing to consider is the amount of data, in each block, that is used for the estimation. **To avoid a bias, the estimation must be done over an integer number of periods**. The calculation of the block size is done on the fundamental tone since this has the lowest frequency and thereby requires most samples to fulfill this requirement. In the FR case the frequency of the fundamental tone is $1/(8 \cdot (3/5200) \cdot 8000) \approx 108$ Hz and the sample rate is 8 kHz. This leads to 480/13 samples to represent one period. To fulfill the requirement 13 periods are required, which gives a block size of 480 samples.

### HR case

In the HR case the same method, with the code command, is used to synchronize the data samples with the 26-multiframe structure. As mentioned in Section 2.2 there are two different HR channels, even and odd, which result in two different radio access patterns. This has to be considered when the samples are divided into blocks. The MS $\mu$C has information about which type of channel that is used. To be able to attenuate the fundamental tone of the HR Bumblebee a block size of 960 samples should be needed. However, because of the band limiting that comes with the filters in connection with the A/D converter and the speech coder, the fundamental tone does not have to be considered. This results in that a block size of 480 samples is enough. However, to get an integer number of periods the even number of harmonics must be estimated on only 320 samples of the data set. To cover the same frequency range for HR as for FR, $K$ in Equation 4.1 must be set to 32, where for HR

$$\omega_0 = 2\pi \frac{1}{2 \cdot 8 \cdot \frac{3}{5200} \cdot 8000} \approx 2\pi \cdot 108 \text{ rad/s}.$$

### Using the MS VAD

A program that is working equivalent to one of the VAD algorithms that are present in the MS has also been tested. This VAD algorithm is working on 160 sample blocks, and set an output bit to one if speech is present. To accept an estimation on three blocks (480 samples) these three plus the next coming block must all have VAD=0. The reason for this is that even if VAD=0, there may be the beginning of speech at the end of the block that would destroy the estimation. Those tests with the VAD algorithms are not presented because they gave the same result as when the VAD was done by hand.

To avoid a delay of the speech it is possible to estimate on one block delayed samples but subtract on the present, see Figure 4.4. This results in

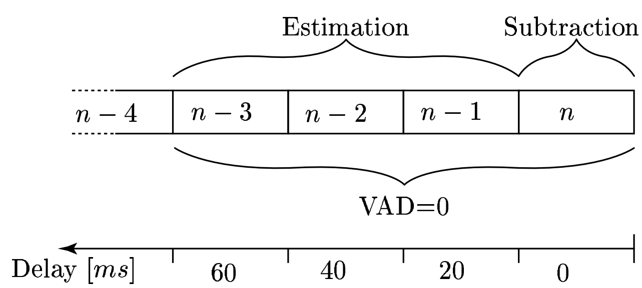



Figure 4.4: *Estimation and subtraction when the VAD algorithm is used*

that the estimate always will be delayed, but this will not cause any major problem. More important is that the speech not will be delayed.

## 4.2.2 Tests

The data set that has been used is identical to when testing the Notch filter. That is, the first test is done on data recorded in a silent room. This data is shown in Figure 4.5 [S4]. The second is made on data with speech and is shown in Figure 4.6 [S5].

Also two tests of the HR case have been done, the first is presented in Figure 4.7, where the Bumblebee is recorded in a silent room [S6]. The second test for the HR case is made on data with speech, and is shown in Figure 4.8 [S7].

## 4.2.3 Evaluation

Even if the results seem to be similar with the results of the Notch, there is a big difference in speech quality. Since a VAD has been used and the idle has been considered, the metallic sound and the new periodic interference

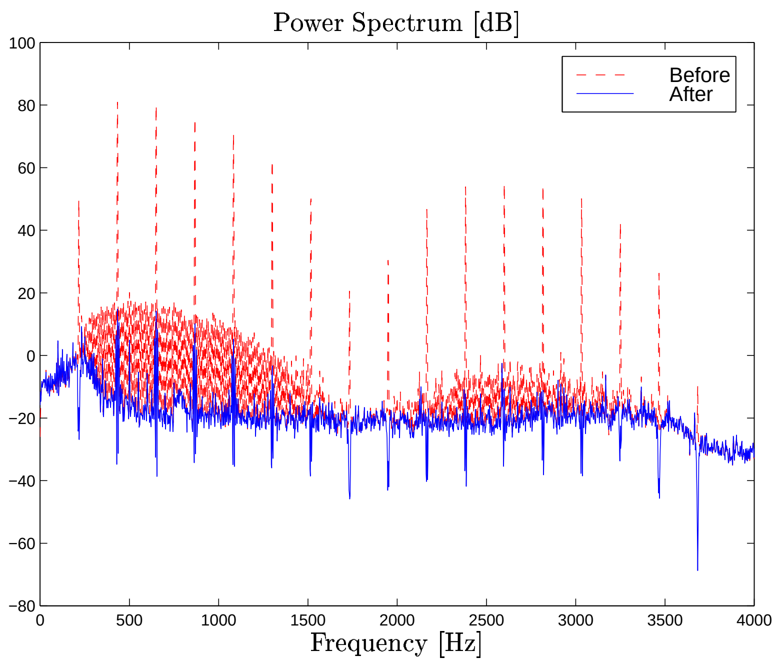



Figure 4.5: *Cancellation of the Bumblebee with* **correlators** *where idle frame has been considered. The Bumblebee was recorded in a silent room* [S4]. **Full Rate, no speech**.

that appeared in the Notch tests are avoided. This gives a very satisfying result. Observe in Figure 4.6 that only the Bumblebee is attenuated.

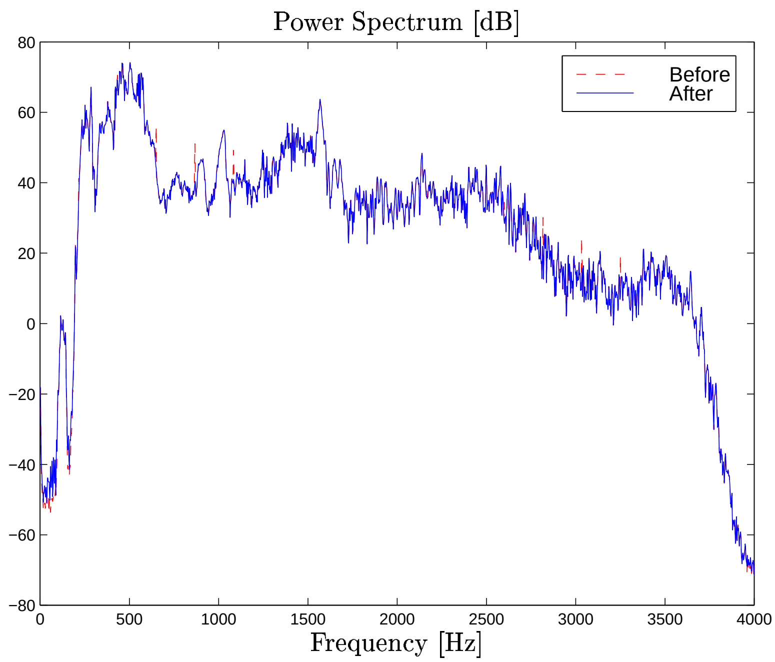



Figure 4.6: *Cancellation of the Bumblebee in speech with* **correlators** *where VAD and idle have been considered* [S5]. **Full Rate, with speech**.

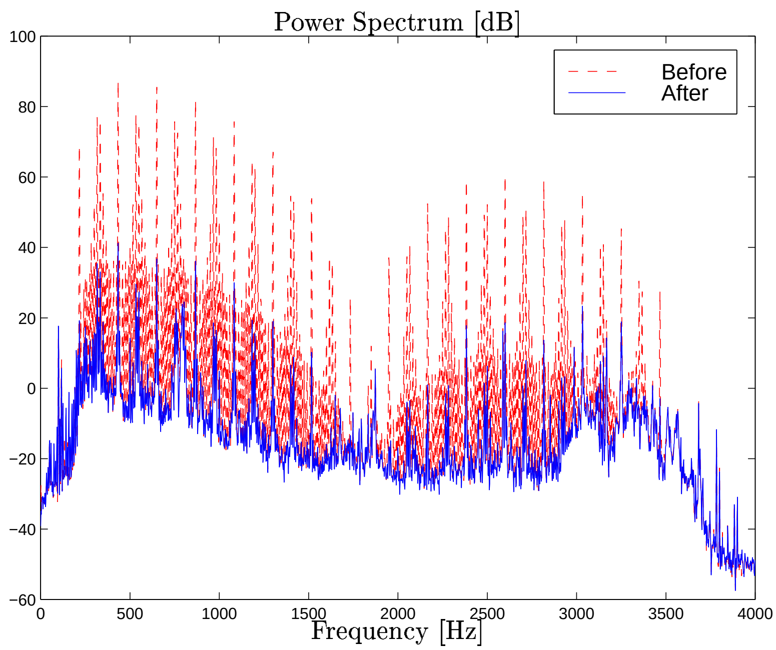



Figure 4.7: *Cancellation of the Bumblebee with* **correlators** *in the* **Half Rate** *case where the idle frame has been considered. The Bumblebee was recorded in a silent room* [S6]. **No speech**.

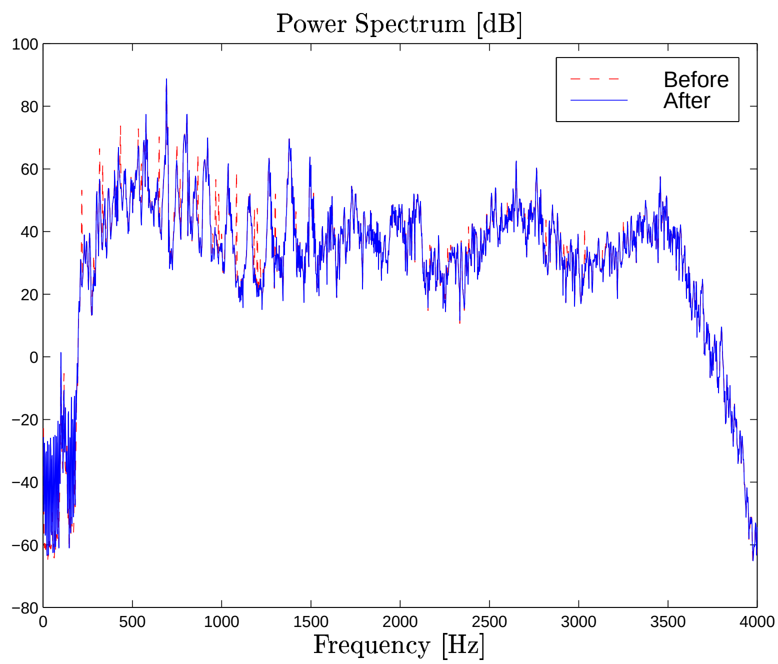



Figure 4.8: *Cancellation of the Bumblebee in speech with* **correlators** *in the* **Half Rate** *case where the VAD and idle frame have been considered* [S7]. **With speech**.

## 4.3 Interpolation in frequency domain

In this simulation a 256 bin FFT has been used considering what is reasonable in the MS DSP.

### 4.3.1 Implementation

Performing a 256 bin FFT on the signal results in a distance of 30 Hz between the FFT bins, when the signal is sampled at 8 kHz. The FFT, IFFT and the data blocks all must have the same size, in this case 256. The approximation of the Bumblebee is only done during speech pauses. If it is done during speech, the approximation will be incorrect since the speech contains energy at the same frequencies. This is solved by using a VAD. In the MS there is already a VAD that can be used, but in these tests this has been done manually.

### 4.3.2 Tests

These tests are done on the same data that has been used for the previous tests. The first test is done on data recorded in the silent room, see Figure 4.9 [S8]. The second is made on data with speech and is shown in Figure 4.10 [S9].

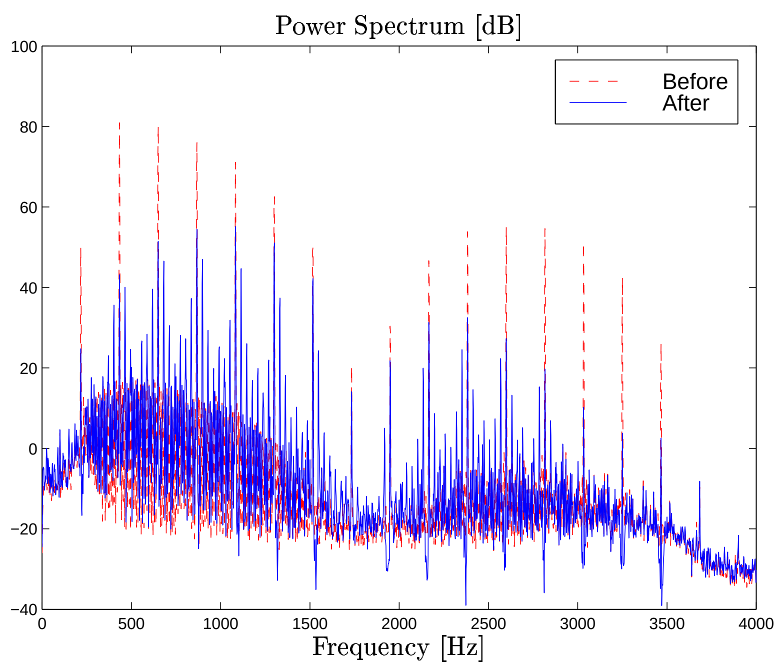



Figure 4.9: *Cancellation of the Bumblebee with* **interpolation**. *The Bumblebee was recorded in a silent room* [S8]. **Full Rate, no speech**.

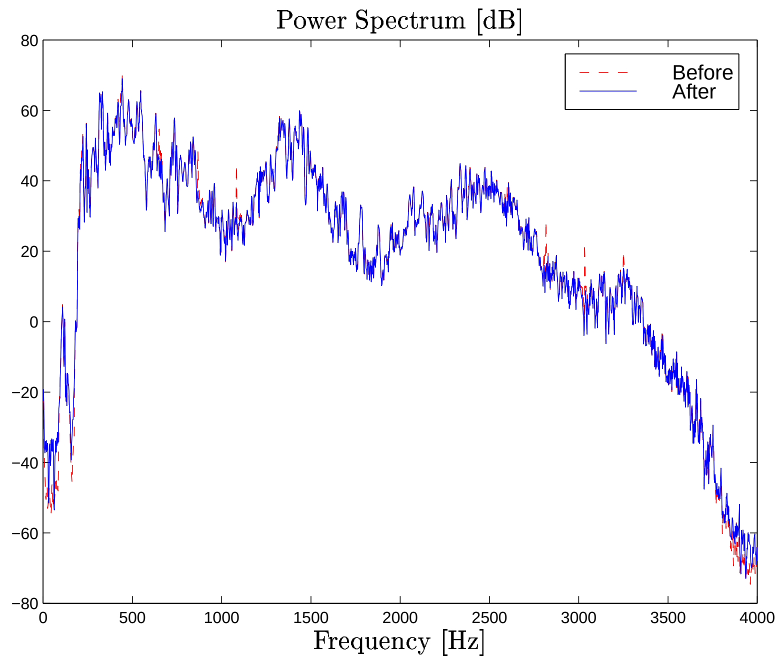



Figure 4.10: *Cancellation of the Bumblebee in speech with* **interpolation** *where VAD has been considered* [S9]. **Full Rate, with speech**.

### 4.3.3 Evaluation

In Figure 4.9 it can be seen that the result of using Interpolation is not very satisfying. The Bumblebee is not completely attenuated and it has leaked out in frequency. One of the reasons for this is that the bandwidth of the FFT bins is too large. Using a larger FFT to decrease the bandwidth would result in a larger block size, which would make it difficult to find blocks that do not contain any speech. Another reason is that the bins of a 256 FFT do not match in frequency with the fundamental tone of the Bumblebee and its harmonics.

# Chapter 5

# Conclusions

In this chapter a further study of the results has been done. There is also a discussion of what is possible to do in the Digital Signal Processor, and a calculation of the complexity of one proposed algorithm. Finally, there are some ideas of further work.

## 5.1 Results

It is difficult to find a fair expression that describes the speech quality after attenuation of the Bumblebee and makes it possible to compare the different algorithms, since the result is influenced by many variables. It is especially difficult if an objective value of the SNR improvement is desired.

Instead different plots of the power spectrum have been done on the results for the Notch filter and the correlators, which were the algorithms that gave the best results. In Figure 5.1 $\frac{P_{Correlators}}{P_{org.sign.}}$ is illustrated, which shows the attenuation in dB and width of the attenuation when correlators are used. Figure 5.2 illustrates the corresponding plot for the Notch filter, $\frac{P_{Notchfilter}}{P_{org.sign.}}$.

It can be seen when the figures are compared that the Notch filter gives a both deeper and wider attenuation. This explains the metallic sound when the Notch filter is used.

## 5.2 What is possible in the DSP

Digital Signal Processors (DSP) are microprocessors designed to perform mathematical operations on digitally represented signals. The performance of todays DSPs is approximately 100-200 MHz. For higher performance it is also possible to have several DSPs working in parallel.

Most DSPs share some common basic features designed to support high performance, repetitive, numerical intensive tasks. The most often cited of these features is the ability to perform a Multiply ACcumulate operation

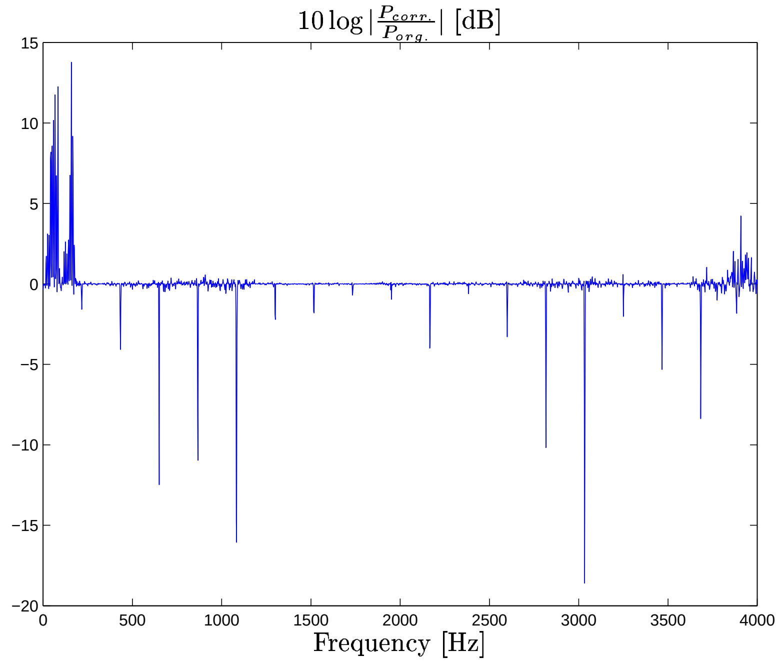



Figure 5.1: *Cancellation of the Bumblebee with correlators. The Bumblebee was recorded in a silent room.*

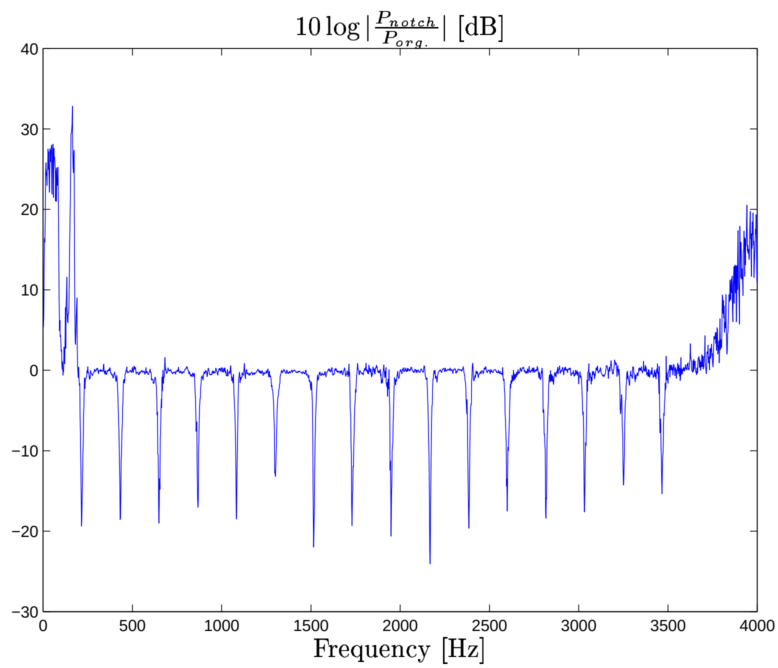



Figure 5.2: *Cancellation of the Bumblebee with notch filter. The Bumblebee was recorded in a silent room.*

("MAC") in a single instruction cycle. The MAC operation is useful in DSP calculations such as the vector dot product used for digital filtering. The dot product, derived by summing the products of vector element pairs, is efficiently calculated with repeated MAC operations.

The number of instructions to calculate trigonometric functions and square roots depends on the desired precision.

## 5.3 Complexity

Complexity estimates have only been made for the correlators, since this solution gave the best result. The most commonly used unit when performing complexity estimates is MIPS (Millions of Instructions Per Second). However, this is misleading because of the varying amounts of work done by an instruction. That is, an instruction on one processor may accomplish far more work than an instruction on another processor. This is especially true for DSP processors, which often have highly specialized instruction sets. Similarly, MOPS (Millions of Operations Per Second) suffer from related problems, what counts as an operation and the number of operations needed to accomplish useful work vary greatly from processor to processor. A third performance unit that can be used is MACS (Multiply ACcumulates per Second). Most DSP processors can complete one MAC per instruction cycle, making this unit equivalent to MIPS for DSPs. Furthermore, MAC estimates disregard the important data movement and processing required before and after multiply accumulates.

The complexity calculations are based on the attenuation of 16 sinusoids. The estimation is performed on 480 samples, and the subtraction of the estimated signal on 160 samples. This is how it has to be done in the MS to avoid a delay. The sinusoids and the cosinusoids are stored in a Read Only Memory (ROM) as a table. Another solution could be to use a digital sinusoidal oscillator, see Appendix C. This solution does not require as much ROM memory as the table approach, but has a much larger complexity and does not generate the sinusoids and the cosinusoids perfectly correct.

To build up the 480 samples long sinusoids and cosinusoids the table have to contain an integer number of periods of every frequency. That is $480/k$ samples except for the frequencies stated in Table 5.1. If $K = 16$, a ROM of 6452 words, is required and the complexity is approximately 1.3 MIPS, see Table 5.2. Control code and data transfers will also be needed. A conservative estimation of the total complexity is 2 MIPS.

## 5.4 Future work

As mentioned in Chapter 2.2 the fundamental tone (and the first harmonic for HR) are already severely attenuated because of the filters in connection

| k | Samples needed |
|---|---|
| 7 | 480 |
| 9 | 160 |
| 11 | 480 |
| 13 | 480 |
| 14 | 240 |

Table 5.1: Samples needed for the frequencies $k \cdot f_0$

| Task | Instructions / 20ms | MIPS |
|---|---|---|
| Correlation | $16 \times 2 \times 480$ | 0.768 |
| Building $\hat{b}$ | $16 \times 2 \times 2 \times 160$ | 0.512 |
| Subtracting | 160 | 0.008 |
| **Total** | **25760** | **1.288** |

Table 5.2: Complexity of the Table approach

with the A/D converter and the speech coder. This should make it possible to ignore these tones without degrading the result. Then only 320 samples are required to satisfy the requirement to have an integer number of periods of the represented sinusoids. However, for the odd harmonics only 240 samples of the data set must be used to get an integer number of periods, see Section 4.2.1.

Also an evaluation of the number of harmonics that are required to attenuate to get a satisfying result should be performed. For such an evaluation a subjective test could be interesting.

For the Notch filter and the Interpolation there are different methods that would probably tune up the result. For the Notch filter the number of poles and the placement of these could be evaluated. The Interpolation could be evaluated for different sizes of the FFT, i.e. a 480 bin FFT would match in frequency with the fundamental tone of the Bumblebee and its harmonics. However, for these methods the idle frame would still be a problem.

It is difficult to find a fair expression that describes the speech quality after attenuation of the Bumblebee, which would make it possible to compare the different algorithms. One way could be to make a faked test signal

$$x(n) = s(n) + b(n)$$

where $s(n)$ is a known speech signal and $b(n)$ is the Bumblebee recorded in a silent room. The algorithms could then be applied to this signal which would result in

$$\hat{s}(n) = s_o(n) + b_o(n)$$

where $s_o(n)$ is the remaining speech and $b_o(n)$ is the remaining Bumblebee. If a N bins FFT is performed on $x(n)$, $s(n)$, $b(n)$ and $\hat{s}(n)$ they could be used to get a curve that compares the Signal-Distortion-Ratio (SDR) with the Signal-to-Distortion-Noise-Ratio (SDNR) improvement. This curve would show the pros and cons with different methods, and also what result different changes in a method would give.

$$SDR = 10\log\left[\frac{\sum_{n=1}^{N}|S(n)|^2}{\sum_{n=1}^{N}|(S(n)-\hat{S}(n))|^2}\right],$$

$$SDNR_{improvement} = SDNR_{after} - SDNR_{before}$$

where

$$SDNR_{before} = 10\log\left[\frac{\sum_{n=1}^{N}|S(n)|^2}{\sum_{n=1}^{N}|B(n)|^2}\right]$$

and

$$SDNR_{after} = 10\log\left[\frac{\sum_{n=1}^{N}\hat{|}\hat{S}(n)|^2}{\sum_{n=1}^{N}|S(n)-\hat{S}(n)|^2}\right].$$

In these equations $n$ represent the FFT bins. It is also possible to look at the frequency components of the Bumblebee one by one.

# Appendix A

# An introduction to GSM

To get a profound knowledge of the problem it is necessary to understand how the GSM is working. This chapter contains an explanation of the radio access parts that are interesting for the Bumblebee problem.

## A.1 Digital Radio Transmission

### A.1.1 General

The GSM system utilizes digital radio transmission. This means that a series of symbols, ones and zeros, are transmitted from one point to another. Since speech is an analog, continuous wave, it has to be transformed and described in digital terms (A/D-conversion). This is done using a process called PCM, Pulse Code Modulation, which is a common principle used in telecommunication systems.

PCM involves three main steps: sampling, quantization and coding. The process of PCM, including sampling at 8kHz and performing quantization and coding using 13 bits, produces a bit rate of $8000 \times 13 = 104$ kbits/s. The digitized speech is then divided into segments of 20 ms, which are fed into the speech coder for reduction of the bit rate to 13 kbit/s. The next step is channel coding that increases the bit rate to 22.8 kbit/s in order to improve the error resilience. Finally, after interleaving, ciphering and burst formatting, the rate is 33.8 kbit/s. The last step is to modulate the bit stream on a carrier and to transmit the signal on a link. In order to use the link more efficiently several channels are multiplexed onto the same link. The technique utilized in GSM is called TDMA, Time Division Multiple Access, and means that several channels share the same link. Each channel uses the link during a certain amount of time, a time slot.

On the receiving side the corresponding procedure is performed. The difference between the mobile station side and the network side is mainly that the speech is not A/D- or D/A-converted on the network side. The network will transmit the digital signal through the network, while the mobile station has to convert it to understandable speech directly.

If data is transmitted instead of speech, no A/D- or D/A-conversion is performed on the mobile side, nor is the data fed through the speech coder. There is also another type of channel coding in this case since data is much more sensible to transmission errors.

### A.1.2 Transmission

The solution and how the transmission is carried out is shown in Figure A.1. Figure A.1 shows the signal processing blocks very schematically. It can been seen that signal processing is performed both in the mobile station and in the network.

#### Speech coding

First the analog speech is digitized by the A/D-converter (analog-to-digital) using a PCM process. It is then divided into 20ms segments, which are fed into the speech coder for further compression. Since it is necessary to have a low bit rate in the speech coding, it is not possible to transmit the speech itself, what is done is that information about the speech is transmitted. This is done by employing a model of how speech is created from the diaphragm, through the lungs and into the vocal tract, where we have the vocal cords and the tongue, see Figure A.2. With this knowledge a parametric model of speech is created.

The speech can be separated into voiced and unvoiced sounds. The voiced speech sounds, vowels, are produced when the vocal cords vibrate. Unvoiced speech sounds are produced when the vocal cords do not vibrate, i.e., "s" and "f" sounds.

The voiced speech production can be modeled with the vocal tract filter being excited by a pulse source, while the excitation in unvoiced speech can be considered to be noise.

Since the speech organs are slowly moving, it is a fair approximation to say that the filter parameters representing the speech organs are constant for 20ms.

On the transmitting side there is a model or a filter H with the inverse characteristics compared to the filter model for creation of speech, see Figure A.3 An analysis function in the speech coder calculates the filter parameters for H so that the output signal will be as close as possible to a white noise signal. The speech coder also estimates the frequency of the vocal cords. What is then transmitted over the air is the filter parameters (so it is possible to establish an inverse filter 1/H) and information about the "excitation

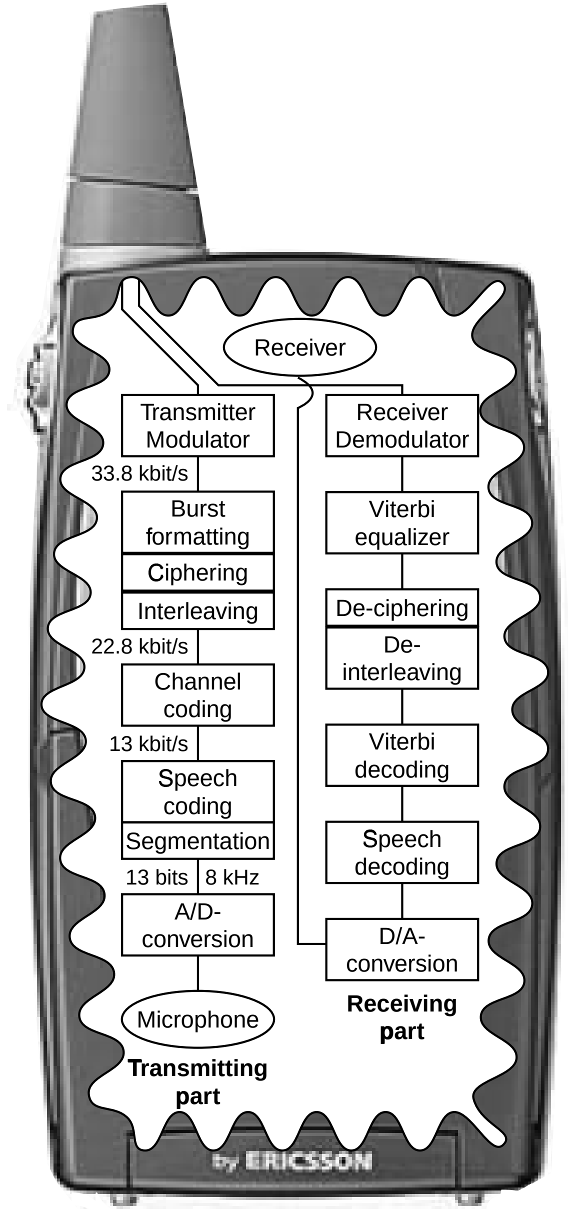



Figure A.1: *The signal processing blocks for the transmission between mobile station and base station.*

sequence" (=output from H). The output on the receiving side should be understandable speech of "good quality". The result of the speech coder is a compression of the speech to 260 bits for each 20 ms frame of speech. This

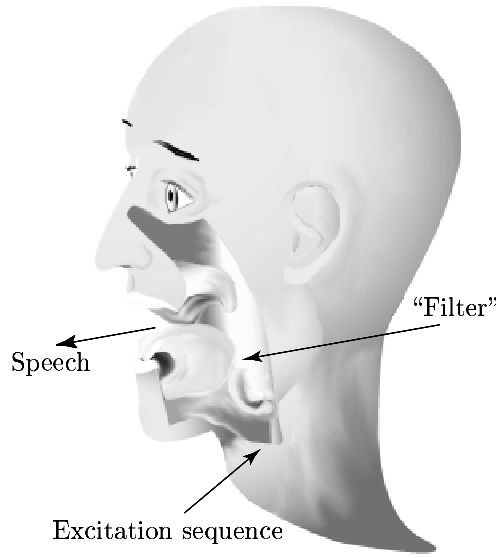



Figure A.2: *The human speech process.*

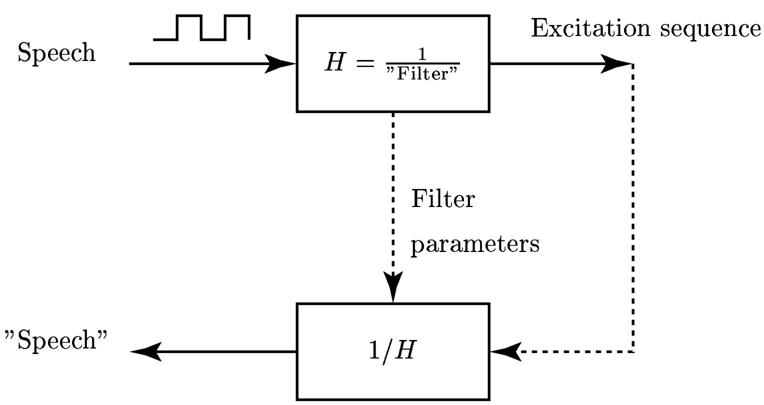



Figure A.3: *The speech transmission model.*

is the case for a Full Rate coder. There are also two other speech coders defined in GSM, EFR and HR, but these are not considered in this report.

### Channel coding

The next step is channel coding where the 260 speech coded bits are divided into

- 50 very important bits
- 132 important bits and
- 78 not so important bits

To the 50 bits, three parity bits are added (block coding). These 53 bits together with the 132 important bits and 4 tail bits are convolutionally encoded to 378 bits (rate 1:2). The remaining bits are not protected. This result is that the channel coder provides 456 bits every 20ms, corresponding to a bit rate of 22.8 kbits/s

### Interleaving

The 456 bits are interleaved into 8 groups of 57 bits. In a normal burst, there is space for 2×57 bits of coded speech information. This would result in a 25% loss if one burst was lost and 2×57 bits was sent in that burst. This is too much for the channel coding to cope with. Instead another level of interleaving is added between two speech frames, see FigureA.4. This will introduce a small delay in the system. On the other hand, it is now possible to lose one whole burst since the loss only affects 12.5% of the bits of each speech frame, which the channel coder can correct.

### Ciphering and Burst formatting

Ciphering of the speech is to protect the call from eavesdropping and in Burst formatting start and stop bits, flag etc. are added.

### Modulation

The last step is to modulate the bit stream on a carrier and then to transmit the signal. The modulation method used in GSM is Gaussian Minimum Shift Keying (GMSK). The modulator can be viewed as a phase modulator. The carrier changes phase depending on the information bits sent into the modulator. GMSK includes the desirable feature of a constant envelope modulation within a burst. To obtain a smooth curve shape when changing the phase, the baseband signal is filtered with a Gaussian shaped filter in passband. GMSK has a more narrow bandwidth as compared to ordinary MSK. The price paid for this is less resistance against noise.

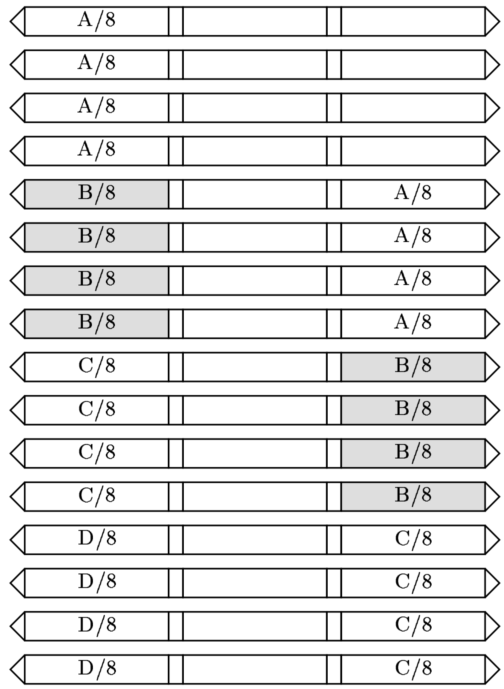



Figure A.4: *Second level of interleaving.*

### A.1.3 Timeslots and TDMA frames

The relationship between bursts and frames is shown in Figure A.5. A time slot has a duration of 3/5200 seconds($\approx$577$\mu$s). Eight time slots form a TDMA frame. The time slots within a TDMA frame are numbered from 0 to 7, and a particular time slot is referred to by its Time slot Number (TN).

The TDMA frames are then numbered by a Frame Number (FN). The FN is cyclic and has a range of 0 to FNMAX, where FNMAX=(26$\times$51$\times$2048)-1=2715647. The FN is incremented at the end of each TDMA frame.

The complete cycle of TDMA frame numbers from 0 to FNMAX is defined as a hyperframe. A hyperframe consists of 2048 superframes, where a superframe is defined as 26$\times$51 TDMA frames. A 26-multiframe, comprising 26 TDMA frames, is used to support traffic and associated control channels. A 51-multiframe comprising 51 TDMA frames, is used to support broadcast, common control and stand alone dedicated control (and their associated control) channels. Hence, a superframe may be considered to consist of 51 traffic/associated control multiframes, or 26 broadcast/common control multiframes. A 52-multiframe, comprising two 26-multiframes is used to support packet data traffic and control channels.

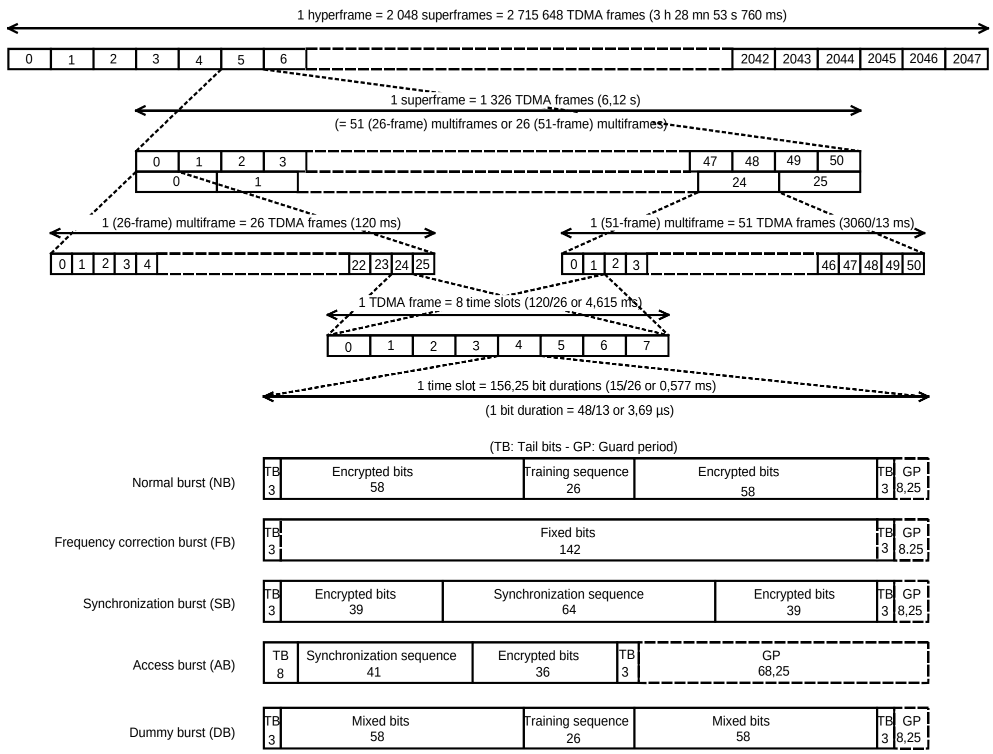



Figure A.5: *Bursts and frames*

### A.1.4 Bursts

The information contained in one time slot on the TDMA frame is called a burst, see Figure A.5. There are five different types of bursts:

- **Normal Burst (NB)** is used to carry information on traffic and control channels.
- **Frequency correction Burst (FB)** is used for frequency synchronization of the mobile.
- **Synchronization Burst (SB)** is used for frame synchronization of the mobile.
- **Access Burst (AB)** is used for random and handover access.
- **Dummy Burst (DB)** is used when no other channel requires a burst to be sent and carries no information.

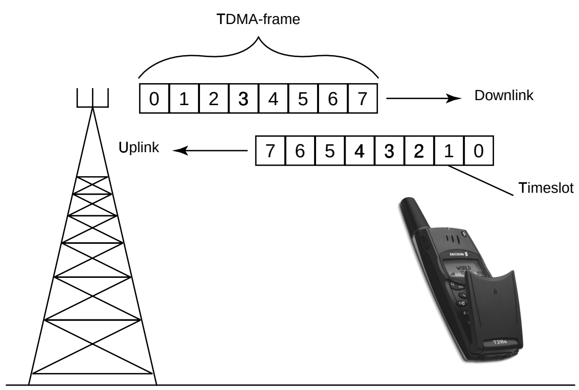



Figure A.6: *The TDMA channel concept.*

## A.2 The Channel concept

### A.2.1 Physical channels

One time slot on a TDMA frame on one carrier is referred to as a physical channel. Consequently, there are eight physical channels on each carrier frequency, channel 0 to 7 corresponding to time slot 0 to 7. The information sent during one time slot is called a burst. Figure A.6 shows the TDMA channel concept.

### A.2.2 Logical channels

A variety of information is transmitted between the Base Transceiver Station, BTS, and the MS. Different logical channels are used depending on the type of information being transmitted. Each logical channel is used for a specific purpose such as paging, call setup or speech. These logical channels are mapped onto the physical channels. Several logical channels share the same physical channel. There are two types of logical channels:

- traffic channels
- control channels

For example, speech is sent on the traffic channel. The control channels are used, e.g., to send signaling information used at call setup. Figure A.7 shows the different logical channels.

#### Stand alone Dedicated Control CHannel (SDCCH)

The call setup procedure is performed on the SDCCH, as well as the transmission of text messages (Short Message and cell Broadcast) in idle mode.

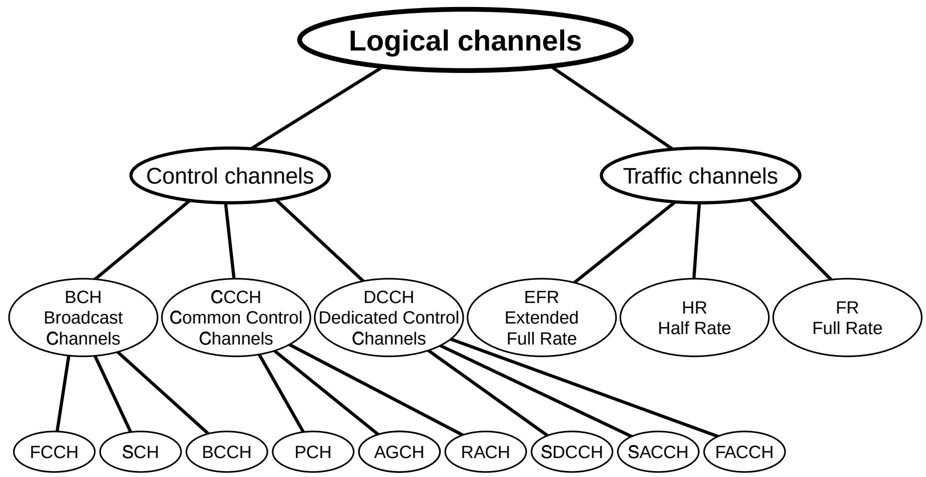



Figure A.7: *The different logical channels.*

SDCCH is transmitted on both up and down link and point to point. When call setup is performed, the MS is ordered to switch to a Traffic CHannel (TCH) defined by a carrier frequency and a time slot.

### Slow Associated Control CHannel (SACCH)

Within certain time intervals on the SDCCH and on the TCH, information is transmitted on the SACCH. On the uplink, the MS sends averaged measurements on the currently used base station (signal strength and quality), as well as neighboring base stations (signal strength). On the downlink, the MS receives information on what transmitting power to use and an instruction on the timing advance. SACCH is transmitted on both up and down link and point to point.

### Fast Associated Control CHannel (FACCH)

If a handover must be performed suddenly during the conversation, the FACCH is used. FACCH works in stealing mode, meaning that one 20 ms segment of speech is exchanged for signaling information necessary for the handover. To compensate this the speech coder repeats the previous speech block.

### Traffic CHannels (TCH)

There are, at the moment, three types of TCHs:

- Full Rate (FR)

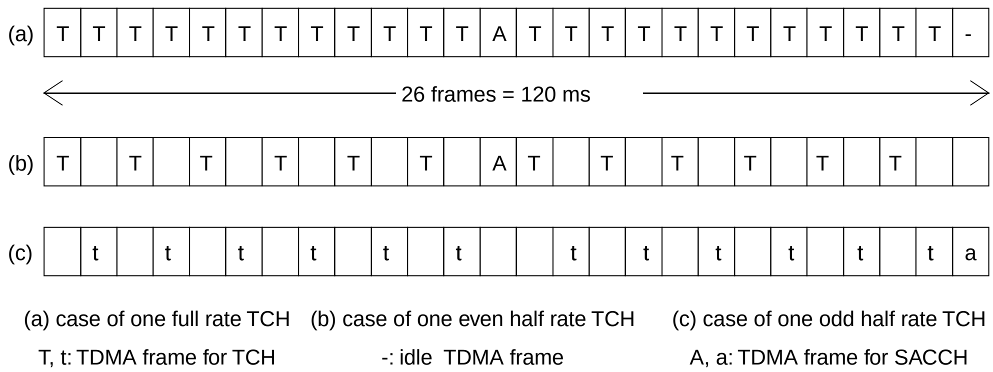



Figure A.8: *Traffic channel organization*

- Enhanced Full Rate (EFR)
- Half Rate (HR)

One FR TCH occupies one physical channel (one TS on a carrier), while two HR TCHs can share one physical channel. Enhanced Full Rate also occupies one physical channel but result in a enhanced speech quality.

The channel organization for the traffic channels (TCH), FACCHs and SACCH uses the 26-frame multiframe. It is organized as described in Figure A.8, where only one time slot per TDMA frame is considered

## A.3 Discontinuous transmission (DTX)

Discontinuous transmission is a mechanism which allows the radio transmitter to be switched off most of the time during speech pauses. This serves two purposes:

- to save power in the mobile station
- to reduce the overall interference level on the air interface.

The DTX mechanism requires the following three functions:

- a Voice Activity Detector (VAD) on the transmit side
- evaluation of the background acoustic noise on the transmit side, in order to transmit characteristic parameters to the receive side
- generation on the receive side of a similar noise, called comfort noise, during periods where the radio transmission is cut.

A frame characterized by information that describes averaged background acoustic noise is called a SIlence Descriptor (SID) frame. The DTX handler continuously passes traffic frames individually marked by a flag SP to the radio system. A speech frame is indicated by SP=1 and a SID frame by SP=0.

The VAD indicates whether speech is present in the frame or not (VAD=1 or VAD=0, respectively). Whenever VAD=1, the speech encoder output frame shall be passed directly to the the radio subsystem, marked with SP=1.

At the end of a speech burst (transition VAD=1 to VAD=0), it takes $N+1$ consecutive frames to make a new updated SID frame available. Normally, the first $N$ speech encoder output frames after the end of the speech burst shall therefore be passed directly to the radio subsystem, marked with SP=1 (hangover period). The first new SID frame is then passed to the Radio SubSystem (RSS) as frame $N+1$ after the end of the speech burst, marked with SP=0, see fig A.9.

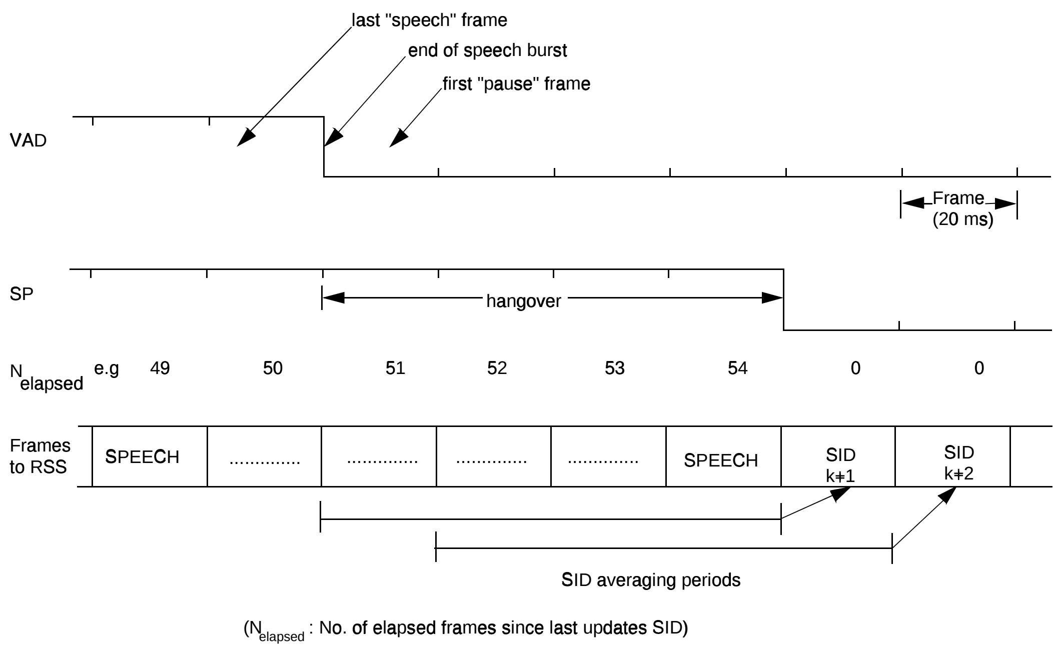



Figure A.9: *Update of SID frame with hangover period*

If, however, at the end of the speech burst, less than 24 frames have elapsed since the last SID frame was computed and passed to the RSS, then this last SID frame shall repeatedly be passed to the RSS, until a new updated SID frame is available (after $N+1$ consecutive frames marked with VAD=0). This reduces the activity on the air in cases where short background noise spikes are taken for speech, by avoiding the "hangover" waiting for the SID frame computation, see fig A.10.

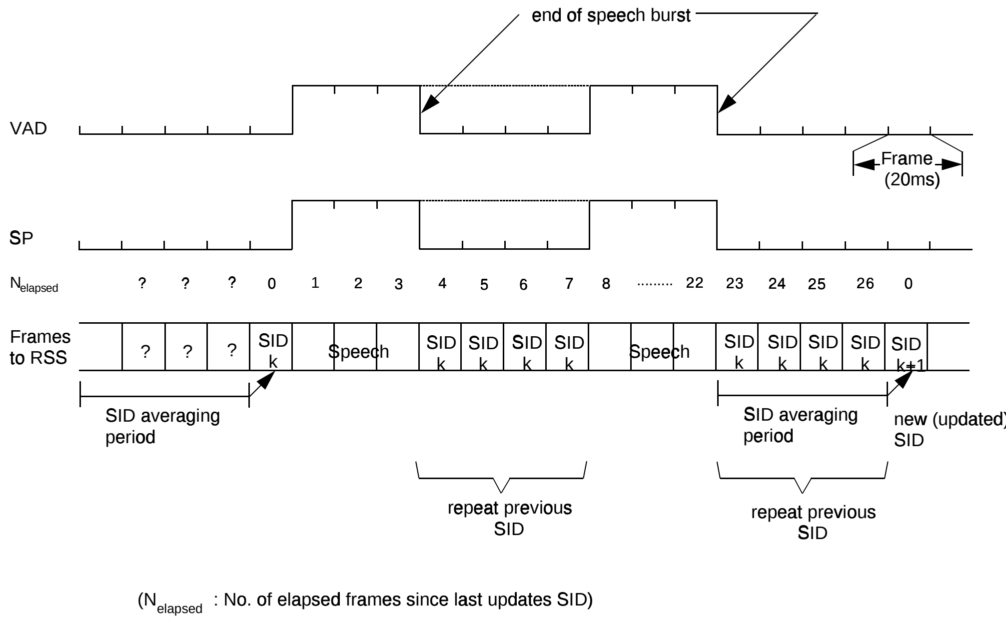



Figure A.10: *Update of SID frames when the number of elapsed frames since last update* $< 24$

Once the first SID frame after the end of a speech burst has been computed and passed to the RSS, the DTX Handler continuously computes and passes updated SID frames to the RSS, marked with SP=0 as long as VAD=0.

During DTX, not all TDMA frames is transmitted. However, the subset stated in Table A.1 is always transmitted. These TDMA frames are occupied by the SID information, SACCH and FACCH.

| **Type of channel** | **TDMA frame subset always to be transmitted TDMA frame number modulo 104** |
|---|---|
| Fullrate TCH | 52, 53, 54, 55, 56, 57, 58, 59 |
| Halfrate TCH, subchannel 0 | 0, 2, 4, 6, 52, 54, 56, 58 |
| Halfrate TCH, subchannel 1 | 14, 16, 18, 20, 66, 68, 70, 72 |

Table A.1: Transmission during DTX

# Appendix B

# Fourier series

The occurrence of periodic functions in technical applications are manifold. Some common examples are Alternating Current (A.C). and mechanical vibrations in machines.

**Definition:** A function $f(t)$ is said to be periodic with the period $T$ if $f(t+T) = f(t)$ for all $t$.

Common examples of periodic functions are $\sin t$ and $\cos t$, which have the period $T = 2\pi$.

All periodic functions can be expressed as

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos(n\Omega t) + b_n \sin(n\Omega t), \tag{B.1}$$

where $\Omega = 2\pi/T$. This representation of $f(t)$ is called the Fourier series expansion of $f(t)$. Let $c$ be an arbitrary real constant, then the Fourier coefficients $a_n$ and $b_n$ are

$$a_n = \frac{2}{T}\int_c^{c+T} f(t)\cos n\Omega t dt$$

$$b_n = \frac{2}{T}\int_c^{c+T} f(t)\sin n\Omega t dt$$

By using Euler's formulas

$$\sin(t) = \frac{e^{jt} - e^{-jt}}{2j}$$

$$\cos(t) = \frac{e^{jt} + e^{-jt}}{2}$$

the Fourier series can be written in its complex form, which is the most commonly used form in signal processing. Equation B.1 then becomes

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos(n\Omega t) + b_n \sin(n\Omega t)$$

$$
\begin{aligned}
&= \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \frac{e^{jn\Omega t} + e^{-jn\Omega t}}{2} + b_n \frac{e^{jn\Omega t} - e^{-jn\Omega t}}{2j} \\
&= \frac{a_0}{2} + \sum_{n=1}^{\infty} \left(\frac{a_n}{2} + \frac{b_n}{2j}\right) e^{jn\Omega t} + \left(\frac{a_n}{2} - \frac{b_n}{2j}\right) e^{jn\Omega t} \\
&= \frac{a_0}{2} + \sum_{n=1}^{\infty} c_n e^{jn\Omega t} + \bar{c}_n e^{jn\Omega t}
\end{aligned}
$$

where $c_n = \left(\frac{a_n}{2} + \frac{b_n}{2j}\right)$. If $c_{-n} = \bar{c}_n$ and $c_0 = \frac{a_0}{2}$ then

$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{jn\Omega t}$$

which is the complex Fourier series of $f(t)$. In practice the $c_n$ coefficients are computed directly without computing $a_n$ and $b_n$. Due to the definition $c_n = \left(\frac{a_n}{2} + \frac{b_n}{2j}\right)$

$$
\begin{aligned}
c_n &= \frac{1}{2}\frac{2}{T}\int_a^{a+T} f(t)\cos(n\Omega t)dt + \frac{1}{2j}\frac{2}{T}\int_a^{a+T} f(t)\sin(n\Omega t)dt \\
&= \frac{1}{T}\int_a^{a+T} f(t)\frac{e^{jn\Omega t} + e^{-jn\Omega t}}{2}dt\frac{1}{jT}\int_a^{a+T} f(t)\frac{e^{jn\Omega t} - e^{-jn\Omega t}}{2j}dt \\
&= \frac{1}{T}\int_a^{a+T} f(t)e^{-jn\Omega t}dt
\end{aligned}
$$

where a is an arbitrary real constant.

**Example:** Half-wave rectification

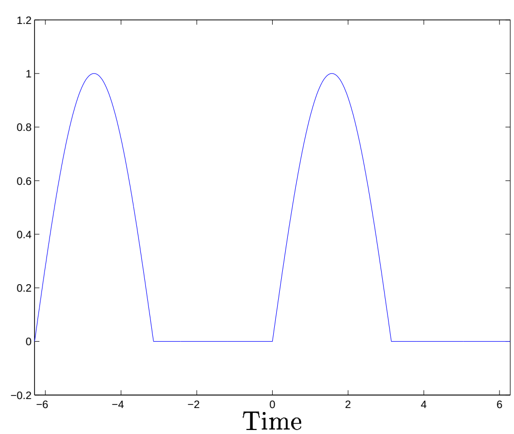



Figure B.1: *Half-way rectified sine signal.*

Sometimes an A.C. is rectified by passing it through a diode. Figure B.1 shows a half-wave rectified sine signal. The Fourier coefficients of this signal are

$$\begin{aligned} a_n &= \frac{1}{\pi}\int_0^{\pi} \sin t \cos ntdt + \frac{1}{\pi}\int_{\pi}^{2\pi} 0dt = \{\text{the formula for } \sin(a+b)\} \\ &= -\frac{1}{\pi}\frac{\cos \pi n + 1}{-1+n^2} = \frac{-2}{\pi(n^2-1)} \end{aligned}$$

when $n$ is even and 0 when $n$ is odd. The coefficient of the sine term is

$$b_n = \frac{1}{\pi}\int_o^{\pi} \sin t \sin ntdt + \frac{1}{\pi}\int_{\pi}^{2\pi} 0dt = \frac{1}{\pi}\int_0^{\pi} \sin t \sin ntdt = \frac{1}{\pi}\cdot\frac{\pi}{2} = \frac{1}{2}$$

when $n = 1$ and 0 otherwise. Consequently, the Fourier series of the signal is

$$f(t) = \frac{1}{\pi} + \frac{1}{2}\sin t + \sum_{n=1}^{\infty} \frac{-2}{\pi(4n^2-1)} \cos 2nt$$

Figure B.2 shows the half-wave rectified signal and the approximation consisting of just one constant, one sine and one cosine term. The corresponding $c_n$ coefficients of the complex Fourier series is

$$c_0 = \tfrac{1}{\pi}, \quad c_1 = \bar{c}_{-1} = \tfrac{1}{4j}, \quad c_2 = \bar{c}_{-2} = -\tfrac{1}{3\pi}.$$

This results in the complex Fourier series

$$f_t = \frac{1}{\pi} + \frac{1}{4j}e^{jt} - \frac{1}{4j}e^{-jt} - \frac{1}{3\pi}e^{2jt} - \frac{1}{3\pi}e^{-2jt}$$

which is shown in Figure B.2.

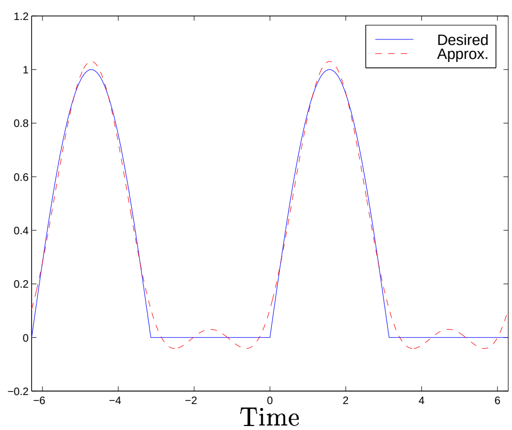



Figure B.2: $\frac{1}{\pi} + \frac{1}{2}\sin t - \frac{2}{3\pi}\cos 2t$ *approximates a half-wave rectified sine.*

# Appendix C

# Digital Sinusoidal Oscillator

This is a method with low complexity that can be used to generate sinusoids, with arbitrary phase and amplitude. A second order IIR filter with a pair of complex-conjugated poles on the unit circle is used. Thus, the position of the poles are

$$p = e^{\pm j\omega_0},$$

which results in the following system function:

$$\begin{aligned} H(z) &= \frac{b_0}{(1 - e^{j\omega_0} \cdot z^{-1})(1 - e^{-j\omega_0} \cdot z^{-1})} \\ &= \frac{b_0}{1 - 2\cos\omega_0 \cdot z^{-1} + z^{-2}} \,. \end{aligned} \tag{C.1}$$

By taking the inverse z-transform of the system function, the unit sample response is obtained

$$h(n) = \frac{b_0}{\sin\omega_0} \sin((n+1)w_0) \cdot u(n),$$

where $u(n)$ is the unit step function. If $b_0$ is set to $A\sin\omega_0$, then

$$h(n) = A\sin((n+1)\omega_0) \cdot u(n).$$

Thus, the impulse response of the filter is a sinusoid and such a filter is therefore called a digital sinusoidal oscillator.

The block diagram representation of the system function given by Equation (C.1) is illustrated in Figure C.1. The difference equation of this system is

$$y(n) = -a_1 y(n-1) - a_2 y(n-2) \qquad n = [2, 3, \ldots], \tag{C.2}$$

where the parameters are $a_1 = -\cos\omega_0$ and $a_2 = 1$. The initial conditions are $y(0) = A\sin\phi$ and $y(1) = A\sin(\omega_0 + \phi)$. The initial conditions comes from the function of the desired output:

$$y(n) = A\sin(n\omega_0 + \phi)$$

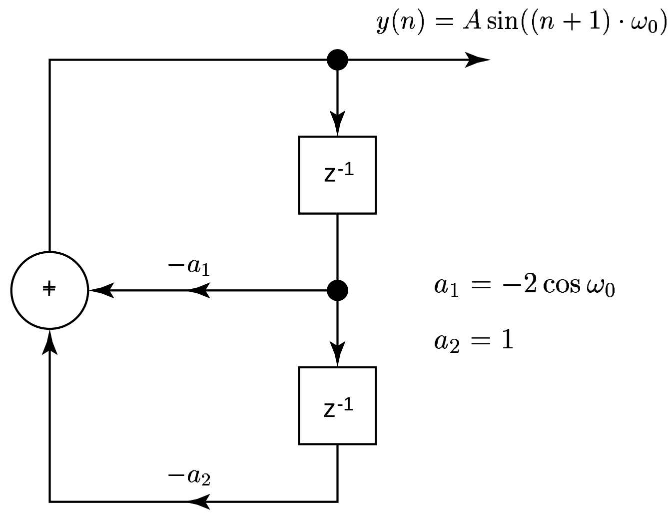



Figure C.1: *Block diagram representation of the system function given by eq.* (C.1)

To reduce the computational burden, four look-up tables can be used that contains $\sin(\omega_0)$, $\sin(\phi)$, $\cos(\omega_0)$ and $\cos(\phi)$. With those values available all parameters and initial conditions, which are different for different phases and frequencies, can be calculated. To be able to calculate $y(1)$ it has to be rewrited as $y(1) = A(\sin(\omega_0)\cos(\phi) + \cos(\omega_0)\sin(\phi))$.

By iterating equation (C.2) a sinusoidal oscillation will be obtained. The oscillation is self sustaining because the system has no attenuation.

# Bibliography


[1] Proakis, J.G. and Manolakis, D.G. *Digital signal processing*, 1996, Prentice-Hall Inc.

[2] Peyton Z. Peebles, Jr. *Probability, random variables, and random signal principles*, 1993, McGraw-Hill Inc.

[3] Simon Haykin *Digital communicatins*, 1988, John Wiley & Sons Inc.

[4] Steven M. Kay *Fundamentals of statistical signal processing: Estimation Theory*, 1993, Prentice-Hall Inc.

[5] GSM Standard (GSM 05.01 version 7.0.0 Release 1998) *Digital cellular telecommunications system (Phase 2+), Physical layer on the radio path (General description).*

[6] Per Eriksson *On estimation of the amplitude and the phase function (Technical Report TR-148)*, 1981, University of Lund / SWEDEN.