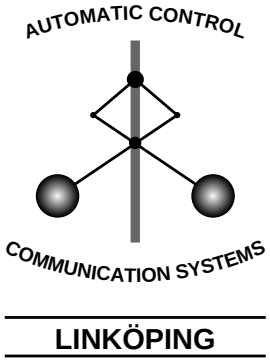
Linköping Studies in Science and Technology
Thesis No. 981

Event Based Sampling with Application to Spectral Estimation

Niclas Persson



Division of Control & Communication
Department of Electrical Engineering
Linköpings universitet, SE–581 83 Linköping, Sweden
WWW: http://www.control.isy.liu.se
Email: persson@isy.liu.se

Linköping 2002

Event Based Sampling
with
Application to Spectral Estimation

Department of Electrical Engineering,
Linköpings universitet,
SE–581 83 Linköping,
Sweden.

ISBN 91-7373-466-7
ISSN 0280-7971
LiU-TEK-LIC-2002:54

Printed by UniTryck, Linköping, Sweden 2002

To Amanda & Pernilla

Abstract

In this thesis event based sampling, estimation of a single resonance frequency and the application indirect tire pressure monitoring system are studied.

Event based sampling is an alternative to traditional uniform or equidistant sampling. In event based sampled systems, signals are sampled only when certain pre-defined events take place. Event based sampling occurs naturally in many contexts such as motion systems where angles and positions are measured. In this thesis special attention is paid to rotating axles where the position is measured with a rotational speed sensor. Due to production tolerances such a sensor has a non-ideal performance. This causes a periodic error in the event based sampled signal which in many cases cannot be neglected. Two different methods for elimitating the sensor errors are proposed.

To be able to use standard tools for analyzing the signal from the rotational speed sensor, the signal is interpolated to a uniformly sampled signal. A desirable feature of the interpolation is to incorporate a lowpass filter to avoid aliasing. We investigate and compare three different methods, all based on a non-parametric regression.

A survey of different methods to estimate a single resonance frequency in the rotational speed signal is presented including aspects of estimation accuracy and computational complexity.

As an application of event based sampling and spectral estimation, a *Tire Pressure Monitoring System (TPMS)* is designed and analyzed. The TPMS is based on event based sampled measurements from a wheel speed sensor. The tire is modeled as a spring-damper system. The idea is to monitor the resonance frequency caused by the spring-damper system. This frequency is correlated with the inflation pressure, and by monitoring the resonance frequency, it is possible to detect inflation pressure changes. The system is implemented and tested in a real vehicle with promising results.

Acknowledgments

First of all I would like to thank my academic supervisor Prof. Fredrik Gustafsson for guidance, inspiring discussions and support throughout my research. I would also like to thank Nils Jönsson for recruiting me to NIRA Automotive AB, and hereby giving me the opportunity to combine academic research with real applications. During the later part of my work Dr. Urban Forssell at NIRA Dynamics AB continued the work Nils started.

I am also grateful to the people in the Control & Communication group for the inspiring spirit they provide. Dr. Fredrik Tjärnström, Ola Härkegård and my room mate Jonas Jansson deserve a special thanks, both for helping me with proofreading the thesis and for the inspiring and fruitful discussions.

This work has been carried out in collaboration with NIRA Dynamics AB. I also acknowledge my colleagues at NIRA Dynamics AB and especially Markus Drevö whom I started this project with and Dr. Tony Gustafsson for interesting discussions and contributing ideas.

Finally I would thank my fiancé Pernilla and my daughter Amanda, for always supporting me in my work. And Amanda, if you ever read this, I would like to thank for all the joy and happiness you have brought into my life.

Linköping, November 2002

Niclas Persson

Contents

Notation

Symbols

t_k	Measured time stamps
t_k^0	True time stamps
$\omega(t)$	Continuous wheel speed
k	Sample index
n	Sample index for decimated signal
ω_k^0	True discrete-time wheel velocity
ω_k	$\int_{t_{k-1}}^{t_k} \omega(t)\, dt$
ω_k^c	Directly computed wheel speed
$\hat{\omega}_k$	Corrected wheel speed
$\hat{\omega}_k^0$	Computed mean wheel speed for one revolution
$\tilde{\omega}_k^0$	Difference between true and computed angular velocity
L	Number of teeth
q_k	Quantization error, $t_k = t_k^0 + q_k$
α	Nominal tooth angle, $\frac{2\pi}{L}$
δ_i	Sensor error i
$T_{s,Clock}$	Sample period for internal clock

Δt_k	Time difference for two consecutive samples, $t_k - t_{k-1}$
$\Delta t_{k,rev}$	Time difference for one revolution, $t_{k+L/2} - t_{k-L/2}$
$\mathbf{C_k}$	Vector with internal counter clock values, c_k
f	Frequency in Hz
f_{res}	True resonance frequency
$\hat{f}_{res}$	Estimated resonance frequency
N	Number of samples
$\bar{V}$	Loss function
R	Covariance function

Operators

q^{-1}	delay operator, $q^{-1}u(t) = u(t-1)$.
$U(a,b)$	Uniform probability density with mean $(a+b)/2$ and variance $(b-a)^2/12$.
$\mathrm{E}(x)$	Expected value of the random variable x
$\mathrm{Std}(x)$	Standard deviation of the random variable x
$\mathrm{Var}(x)$	Variance of the random variable x
$\mathrm{Mean}(x)$	Sample mean of the deterministic variable x, $\frac{1}{N}\sum_{k=1}^{N} x_k$
$\mathrm{RMSE}(x)$	Root mean square error of the deterministic variable x, $\sqrt{\frac{1}{N-1}\sum_{k=1}^{N}\left(x_k - \mathrm{Mean}(x)\right)^2}$

Abbreviations and Acronyms

AC	Auto Calibration
AR	AutoRegressive
DFT	Discrete Fourier Transform
DTFT	Discrete Time Fourier Transform
FFT	Fast Fourier Transform
FS	Fourier Series expansion
HP	HighPass filter
LMS	Least Mean Squares
LP	LowPass filter
RLS	Recursive Least Squares
TPI	Tire Pressure Indicator
TPMS	Tire Pressure Monitoring System

1

Introduction

What is *event based sampling*? The answer is not obvious since there is no established definition for event based sampling. In Åström and Bernhardsson (1999) event based sampling is defined as: *"the system is sampled when the output has changed a specific amount"* and in Powell et al. (1998) it is defined as: *"all the control systems will use a sample rate that is synchronous with the crank shaft motion"*. In this thesis event based sampling is similar to (Åström and Bernhardsson, 1999): *the sample instants are sampled every time the amplitude of the signal passes certain pre-defined levels.* In Figure 1.1 a comparison between the event-domain and time-domain is visualized. Event based sampling is also related to non-uniform sampling and will be further discussed in Chapter 2.

Many applications rely upon event based sampling. Probably the most common application is motion systems where angles and positions are measured. Event based sampling is for example natural in engine control where the crank shaft position determines the control strategy (see Hendricks et al., 1994; Ribbens and Rizzoni, 1989). Event based sampling is also used in vehicle control, where the angular velocities of the wheel axles are measured at regular angular intervals to control the slip between the tire and road, (Rittmannsberger, 1988). Biomedical engineering is another application area with event based sampling. Cardiovascular parameters are evaluated on a beat-to-beat basis yielding a event based sampled time-series (van Steenis and Tulen, 1991).

For some applications, such as automatic control, event based systems may

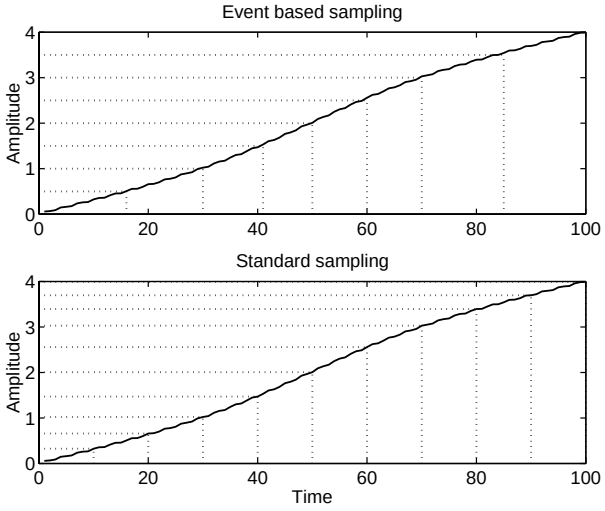


Figure 1.1 *Event based sampling in comparison with standard (uniform) sampling.*

be advantageous because control is not executed unless it is necessary (Kopetz, 1993). In other applications the available sensor implies event based sampling (Galvan et al., 1994).

In this thesis we will treat both theoretical and practical aspects of event based sampling. The thesis also includes analysis and design of attenuating event based sensor errors. In most cases the dynamics of the underlying physical system is time-based. There are two alternatives when the data is event based sampled. Either the dynamic model of the system is converted to the event-domain or the event based data is transformed to the time-domain. Here, we will only treat the latter.

In the thesis a *Tire Pressure Monitoring System (TPMS)* is studied. In a TPMS the main information sources are the wheel speed sensors. The dicretization of a continuous wheel speed sensor signal is a typical example of event based sampling. This includes spectral estimation and especially estimation of an overloaded resonance frequency in the wheel speed.

1.1 Introductory Example: TPMS

One of the most important parameters for the handling properties of a vehicle is the tire-road contact patch. For the contact patch to be optimal the inflation pressure of the tire also needs to be optimal. When the inflation pressure decreases the

handling properties deteriorate. Therefore, it is very important that the tires are correctly inflated. To support the driver TPMSs have been developed (Bishop, 2002; Fennel et al., 2002; Gustafsson et al., 2000b; Hakanen, 2002).

One way of continuously monitoring the tire pressure is to estimate the frequency of a resonance in the wheel speed. The resonance frequency is correlated with the inflation pressure and decreases when the inflation pressure decreases.

Consider the continuous time signal model of the wheel speed

$$\omega(t) = \omega^0(t) + s(t) + e(t) \tag{1.1}$$

where $\omega^0(t)$ is the nominal wheel speed, $s(t)$ is a signal with a pronounced resonance frequency (typically located around 45 Hz) and $e(t)$ is a noise term. $s(t)$ originates from the tire modeled as a spring-damper system, (see Figure 1.2(b)). The wheel speed signal (1.1) is event based sampled by the rotational speed sensor. The sensor measures the time elapsed for the wheel to rotate one L:th of a revolution (see Figure 1.2(a) with $L = 7$). Rotational sensors are further discussed in Section 2.2. This means that the sampling instants are determined by the rotational velocity and the sample period varies from sample to sample. Unfortunately

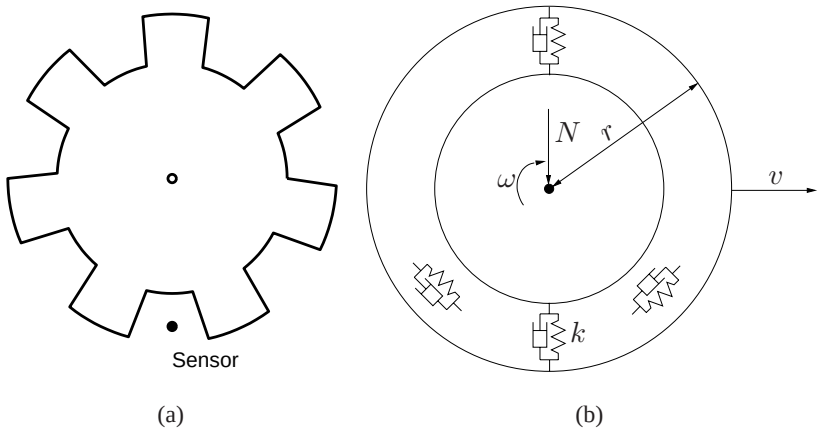


Figure 1.2 *In (a) the sensor device is shown and in (b) the tire modeled as a spring-damper system.*

rotational speed sensors are also subject to sensor errors. In Figure 1.3(a) the periodogram of the event based sampled wheel speed signal is shown. The resonance frequency can be observed around 45 Hz, but unfortunately the energy in the peaks

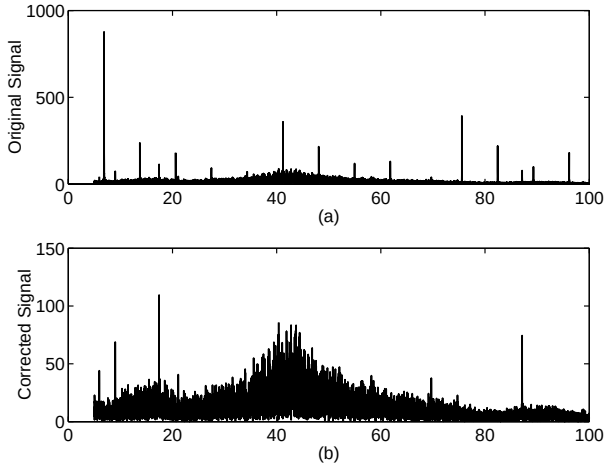


Figure 1.3 *In (a) the power spectrum of the original wheel speed signal is plotted and in (b) the sensor errors are attenuated.*

originating from the sensor errors are many times larger than the energy in the resonance frequency. The sensor error signal needs to be attenuated.

In Chapter 2, methods for attenuating the sensor errors are described. If the sensor errors are attenuated the periodogram in Figure 1.3(b) is obtained. Now the resonance frequency is visible around $40 - 50$ Hz.

It is also interesting to see what happens with the resonance frequency when the tire inflation pressure changes. In Figure 1.4 the smoothed periodogram for three different tire inflation pressures are shown. As can be seen in the figure the resonance frequency is correlated with the tire inflation pressure. The issue is now to estimate the resonance frequency as accurately as possible (Chapter 4). To be able to use classical tools for estimating the resonance frequency, the event based sampled signal is converted to an uniformly sampled time-series. This is treated in Chapter 3. A more detailed description of the Tire Pressure Monitoring System is found in Chapter 5.

1.2 Thesis Outline

The thesis outline is best described by Figure 1.5. The figure also represents the signal flow in the TPMS or some other system aiming at monitoring the spectrum or spectral parameters from event based sampled data.

Chapter 2: Contains theoretical aspects of event based sampling and sensor de-

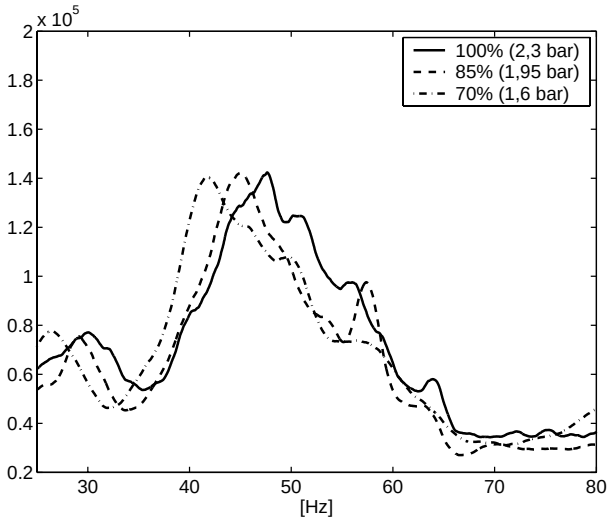


Figure 1.4 *Smoothed periodogram for three different tire inflation pressures: Normal pressure (solid), 85% of nominal pressure (dashed) and 70% of nominal pressure (dash-dotted). The resonance frequency is correlated with the tire inflation pressure.*

vices. It also proposes two different methods to improve the quality of the sensor signal, which makes the rest of the signal processing chain feasible.

Chapter 3: Discusses different methods, and their properties, for converting event domain data to time domain data.

Chapter 4: Proposes three different methods for extracting the resonance frequency from time domain data. Two of the methods include pre-filtering of the data.

Chapter 5: Discusses an indirect TPMS from the continuous time tire model to estimated resonance frequency correlated with the tire inflation pressure, i.e., the whole chain in Figure 1.5. It also contains results from real measurements.

Chapter 6: Concludes the thesis by summarizing the work in terms of performance and applicability. Future prospects for improving the theoretical results are given. There are also some suggestions for future work to enhance the performance of the application TPMS.

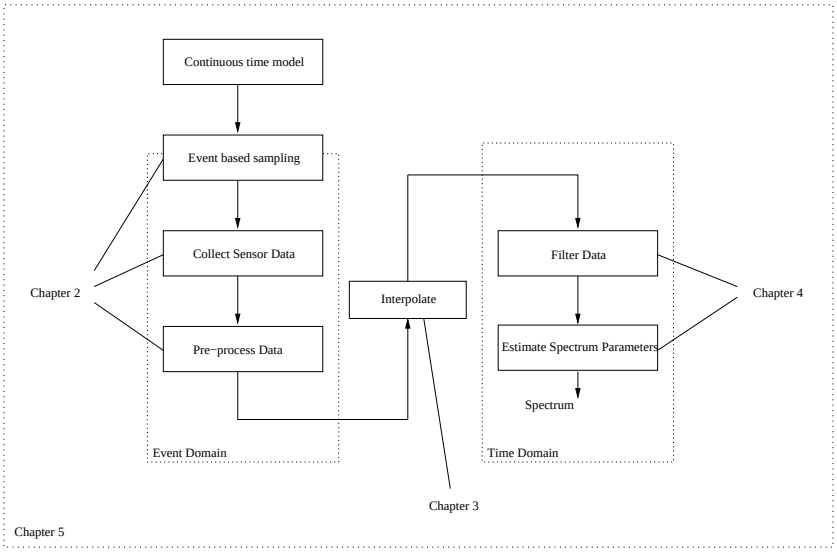


Figure 1.5 *Schematic overview of the thesis or system architecture for a system based on event based sampling.*

Chapters 2-4 use only simulation data while Chapter 5 deals with real measurements. Chapter 5 uses the techniques developed in Chapters 2-4 to extract the resonance frequency, which is used to continuously monitor the tire pressure.

1.3 Readers guide

The thesis is divided into two parts. The first part, Chapters 2-4, contains theoretical aspects of event based sampled systems and spectral estimation and the second part, Chapter 5, contains an application (TPMS) based on the first part. Even though the aim is to keep the first part as general as possible it is "colored" by the applicaton TPMS.

Throughout the thesis x^0 is used to denote the true or nominal value of x and $\hat{x}$ denotes the estimate of x. Sometimes the superscript c is used to denote a computable quantity.

In the thesis the frequency content of a signal is often discussed. Some people often use ω to denote the frequency in rad/s. Here, f is used to denoted the frequence in Hz while ω is used to denote angular velocity.

1.4 Contributions

The main contributions of this thesis are:

- The ideas in Section 2.3 and 2.4 on how to estimate sensor errors in a rotational sensor and correct the signal with respect to the estimated errors.
- The convolution interpolation idea used to transform event-domain data to the time-domain.
- The algorithms to monitor the tire inflation pressure continuously by monitoring a resonance frequency in the wheel speed which is correlated with the tire inflation pressure.

Parts of this thesis have been published previously. The estimation of sensor errors in Chapter 2 originates from

> N. Persson and F. Gustafsson. Event based sampling with application to vibraton analysis in pneumatic tires. In *Proceedings of IEEE Int Conf on Acoustics, Speech, and Sig Proc 2001 in Salt Lake City, USA*, volume 6, pages 3885 –3888, 2001.

The estimation properties in Chapter 4 are evaluated in

> N. Persson. Estimation properties for a tire pressure monitoring system. In *Proceedings of Reglermöte 2002 in Linköping, Sweden*, pages 108–113, 2002:

The TPMS application is presented in

> N. Persson, F. Gustafsson, and M. Drevö. Indirect tire pressure monitoring using sensor fusion. In *Proceedings of SAE 2002 in Detroit, USA*, number 2002-01-1250, 2002
>
> N. Persson, F. Gustafsson, S. Ahlqvist, and U. Forssell. Low tire pressure warning system using sensor fusion. In *Proceedings of ATTCE 2002 in Barcelona, Spain*, number 01ATT-272, 2001
>
> F. Gustafsson, M. Drevö, U. Forssell, M. Löfgren, N. Persson, and H. Quicklund. Virtual sensors of tire pressure and road friction. In *Proceedings of SAE 2001 in Detroit, USA*, number 2001-01-0796, 2001a
>
> F. Gustafsson, N. Persson, and M. Drevö. Tire pressure estimation. Patent no. WO 0187647, 2000b. NIRA Dynamics AB

2

Event Based Sampling

As mentioned in the introduction, there exist more than one definition of event based sampling. Here, event based sampling is defined as:

DEFINITION 2.1 (EVENT BASED SAMPLING)
The sample instant is sampled every time an event occurs.

The definition of an event is:

DEFINITION 2.2 (AN EVENT)
An event is when the amplitude of the signal passes a pre-defined level.

What is the need for a signal sampled at pre-defined levels? In event based sampling the information is stored in the time instants rather than the amplitude of the signal at the sampling times. This is in contrast to both uniform sampling where the sample period is constant and non-uniform sampling where sample period varies (see Marvasti, 1996). In uniform or classical sampling the amplitude is measured at regular pre-defined time intervals and the information is stored in the measured amplitudes. In non-uniform sampling the sample instants are determined by the user or when the system is excited. Both the time instants and the amplitude of the sampled signal contains information.

The signal processing problem, though, is similar no matter which sampling principle used, i.e., separate the signal from noise. In contrast to uniform sampling, which is well explored, there are no standard tools for design and analysis in event

based sampled systems. The design and analysis also needs to be done both in the event and time domain.

In this chapter techniques for analyzing signals and attenuating noise in the event domain will be discussed. This chapter will also treat theoretical aspects of event based sampling and applications where event based sampling can be useful. Sensors for measuring angular velocity will also be discussed.

2.1 Comparing Event Based Sampling and Time Based Sampling

The classical sampling technique measures the amplitude of a continuous time signal $y(t)$ at regular time intervals T_s

$$y_k = y(kT_s), \qquad k = 1, 2, ..., N. \tag{2.1}$$

In event based sampling the signal is sampled every time the amplitude of the signal passes certain pre-defined levels

$$y(t_k) = y_k, \qquad k = 1, 2, ..., N \tag{2.2}$$

where y_k is the pre-defined amplitudes recorded at time instants t_k. Equation (2.2) implies problems when the continuous time signal is not monotonously increasing, i.e., several $y(t_i)$ can be equal to y_k. In many practical applications the continuous time signal is monotonously increasing and the pre-defined levels are uniformly distributed so (2.2) is then formulated as

$$y(t_k) = kY_s, \qquad k = 1, 2, ..., N \tag{2.3}$$

where Y_s is the amplitude sample period. Equations (2.2) and (2.3) will be referred to as the *event domain* while (2.1) is the classical *time domain*.

It is also possible to consider the event domain signal as a time domain signal with varying sample period. One problem is that classical signal processing techniques are usually not applicable when the sample period varies. There is for example no corresponding version of the sampling theorem available for event based sampled signals. Another difference is where the disturbances enter the system. In for example an automotive application where the wheel velocity is measured, road roughness is a typical time domain disturbance while sensor errors, (see Section 2.2) is a typical event domain disturbance. The event based sampled system can be seen as consisting of three different components (2.4).

$$y_k = s_k + e_{k,event} + e_{k,time} \tag{2.4}$$

Here s_k is the information signal, $e_{k,event}$ is a disturbance in the event domain, while $e_{k,time}$ is the disturbance in the time domain. The noise attenuation of a disturbance needs to be done in the domain where it enters the system, i.e., sensor errors need to be attenuated in the event domain. The attenuation of sensor errors will be further discussed in Sections 2.3 and 2.4.

2.1.1 The Sampling Theorem

The sampling process is fundamental in all digital signal processing, since it establishes a link between continuous time and discrete time. An important aspect for time domain sampled signals is the sample period. The sampling theorem states that if the bandwidth of the continuous time signal is less than half of the sample rate, $f_s = 1/T_s$, the continuous time signal can be reconstructed exactly, (see Stoica and Moses, 1997). This also guarantees that the frequency content of the sampled signal will be equal to the continuous time signal. Frequencies above the Nyquist frequency,

$$f_N = f_s/2 \tag{2.5}$$

will be aliased into the interval $[0, f_N]$. In classical signal processing a fixed sample period is used and it is easy to evaluate if the sampling theorem is fulfilled or not. If the signal is event based sampled, the time period between samples will vary and the sampling theorem cannot be used. Unfortunately there is no corresponding version of the sample theorem for event based sampled data available in the literature.

2.1.2 Spectral Analysis

In many cases it is very useful to describe a time domain signal by its frequency content. Sometimes spectral analysis provides an insight into the properties of the system that otherwise would not be possible. A more thorough treatment of spectral analysis can be found in (Gustafsson et al., 2001b; Hayes, 1996; Stoica and Moses, 1997). A continuous time signal can be described in the frequency domain by using the Fourier transform. Mathematically the continuous time Fourier transform is defined as

$$Y(f) = \int_{-\infty}^{\infty} y(t) e^{-i2\pi ft}\, dt. \tag{2.6}$$

To compute the discrete-time Fourier transform *DTFT* of a discrete-time signal, (2.6) can be approximated with a Riemann sum with step size equal to the sample

period, T_s

$$Y(e^{i2\pi fT_s}) = T_s \sum_{k=-\infty}^{\infty} y_k e^{-i2\pi fkT_s}. \tag{2.7}$$

The DTFT is a complex-valued function of the continuous frequency variable f. In both (2.6) and (2.7) it is assumed that the time domain signal is known for all time instants. In practical applications this is not true. If we focus on the DTFT and restricts the number of samples to N, (2.7) needs to be modified with respect to this constraint. One way of achieving the constraint is to consider all samples outside the interval $[0,\ N-1]$ as zero (this is called truncating). Using truncation in (2.7) yields

$$Y(e^{i2\pi fT_s}) = T_s \sum_{k=0}^{N-1} y_k e^{-i2\pi fkT_s}. \tag{2.8}$$

The truncation of the DTFT introduces a distortion which will appear as leakage. Leakage implies that signals with sharp resonance peaks become diffuse, and nearby resonance peaks can be hard to distinguish from each other. Another way of achieving the constraint is to consider the signal as periodic with period NT_s. The theory for *Fourier series* states that a periodic signal can be described by trigonometric functions with the period $1/T_s$. Then we can apply the *Discrete Fourier Transform (DFT)*

$$Y(n) = \sum_{k=0}^{N-1} y_k e^{-i2\pi kn/N},\ n = 0, 1, ..., N-1. \tag{2.9}$$

This is perhaps the most common form of transforming a discrete time signal to the frequency domain. The reason for the wide use of the DFT is the effective numerical algorithm *Fast Fourier Transform (FFT)* available for computing the DFT. The DFT of y_k is equal to the DTFT sampled at N frequencies f equally spaced between 0 and $1/T_s$.

In both (2.8) and (2.9) the sample period T_s is assumed to be constant. Since event based sampling implies a varying sample period none of these methods are directly applicable. The sample instants need to be taken into consideration. Since the frequencies at which the DFT is evaluated are determined by the sample period T_s and number of samples N to receive the efficient implementation FFT the more general (2.8) is used here. To modify (2.8) with respect to the sample instant is

quite straight forward and results in a truncated *Event Based DTFT*

$$\begin{aligned} T_s y_k &\rightarrow y_k \Delta t_k \\ kT_s &\rightarrow t_k \\ Y(f) &= \sum_{k=0}^{N-1} y_k \Delta t_k e^{-i2\pi f t_k}. \end{aligned} \tag{2.10}$$

This method is also supported by van Steenis and Tulen (1991). A comparison between the DFT, (2.9), and the event based DFT, (2.10), is shown in Figure 2.1. In the simulation the signal

$$s(t) = \sin(2\pi 0.3t) \tag{2.11}$$

is used without disturbances. For the DFT the signal is sampled with sample period $T_s = 1$ and for the event based DFT

$$\begin{aligned} e_k &\sim U[-T_s/2, T_s/2] \\ t_k &= kT_s + e_k \\ s_k &= \sin(2\pi 0.3 t_k) \end{aligned} \tag{2.12}$$

is used for sampling. As can be seen in the figure two phenomena can be observed. First, the varying sample period introduces virtual frequencies similar to white noise. Second, the symmetric properties of the DFT is lost. In the figure this can be observed by the missing resonance peak at 0.7 Hz.

2.2 Rotational Speed Sensors

A typical example where event based sampling occurs is when rotational speed sensors are used to measure the angular velocity of rotating axles. The wheel axles in vehicles, the cam shaft axle in engines and motor axles in robots are some examples of applications where these sensors are used. The standard sensor in these applications is an inductive sensor, (BOSCH, 1996; Nwagboso, 1993). In this sensor configuration, a permanent magnet (inductive sensor) with a coil wound around itself is placed closely to a ferromagnetic ring (toothed wheel), see Figure 2.2. When a tooth passes the sensor it generates a voltage directly proportional to the periodic variations in the magnetic flux. The rotational speed is determined using the time elapsed between the zero transition points. Another common method is to use the Hall effect. It is also possible to detect tooth passings by using an optical sensor, but in this case a slotted disc is usually used instead of a toothed wheel.

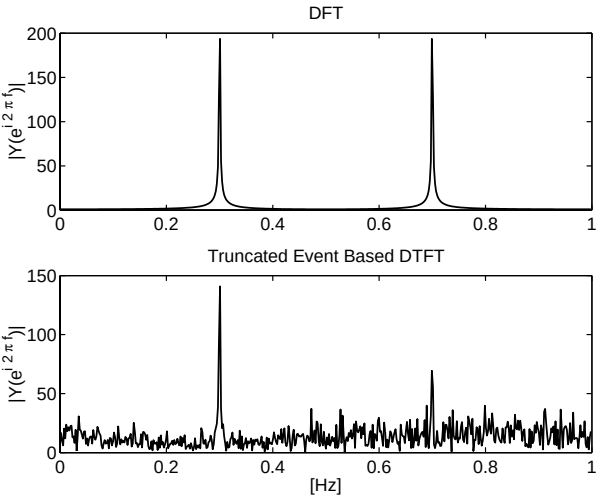


Figure 2.1 *DFT (top) and truncated event based DTFT (bottom) of the signal* (2.11) *sampled with constant sample period and according to* (2.12)*, respectively.*

Rotational speed sensors apply to the event based sampling procedure described in (2.3) where Y_s is replaced by the angular distance α. Figure 2.3 illustrates the principles of event domain sampling, (2.3), and time domain sampling (2.1).

No matter which type of sensor that is used, a continuous signal (voltage) needs to be discretized. This is done by connecting a microprocessor to the sensor checking for zero crossings. The microprocessor consists of an internal counter and clock with clock period $T_{s,Clock}$ and the process can be described with the pseudo code in algorithm 2.1.

ALGORITHM 2.1 Discretization of a continuous signal from a rotational speed sensor

1 Wait $T_{s,Clock}$

2 Flip counter value.

3 Check for zero crossings.

4 IF zero crossings THEN store counter value ELSE goto 1.

❑

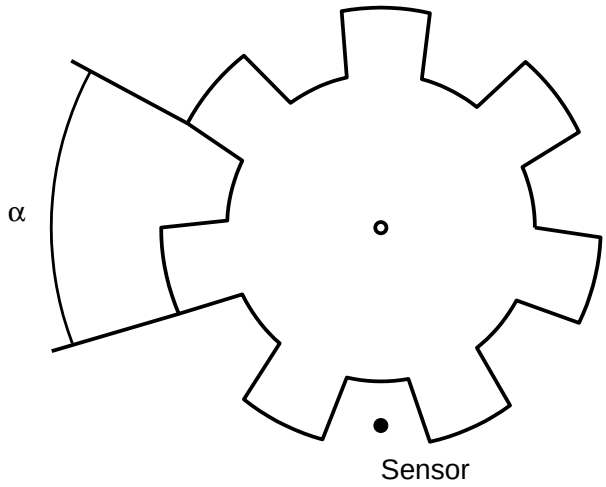


Figure 2.2 *Rotational speed sensor device used to measure the angular velocity for rotating axles.*

The time instants received from the discretization process is a flow of counter values logged in a measurement vector, $\mathbf{C_k} = [c_k, c_{k-1}, ...]$. To get the absolute time $\mathbf{C_k}$ needs to be scaled with the clock period $T_{s,Clock}$

$$\mathbf{t_k} = [t_k, t_{k-1}, ...] = \mathbf{C_k} T_{s,Clock}. \tag{2.13}$$

The event in the sampling process is of course when a tooth passes the sensor, i.e., when the amplitude of the angle signal exceeds $\alpha = 2\pi/L$ where L is the number of teeth. Equation (2.13) enlightens the fact that for event based systems the information is stored in the time instants. An implicit way of defining the sampling instant, t_k, given t_{k-1} is obtained as follows

$$\alpha \triangleq \frac{2\pi}{L} = \int_{t_{k-1}}^{t_k} \omega(t)\, dt \tag{2.14}$$

where $\omega(t)$ is the continuous time angular velocity. The discrete and computable velocity is defined as

$$\omega_k^c \triangleq \frac{\alpha}{t_k - t_{k-1}} = \frac{\alpha}{\Delta t_k} \tag{2.15}$$

which can be seen as the average speed in the interval $[t_{k-1}, t_k]$ or a Riemann approximation of (2.14). If $\omega(t)$ is piece-wise constant for $t_{k-1} < t < t_k$ then $\omega_k^c = \omega(t)$, in the interval. In most applications the angular velocity is the primary

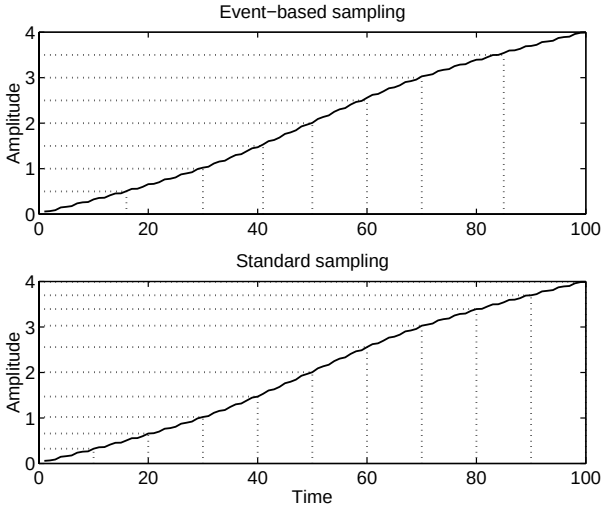


Figure 2.3 *Above the signal is sampled with equidistant amplitude and below with equidistant time.*

signal and (2.15) can be used for computation of the sampled angular velocity. The sample period of the disretized angular velocity ω_k^c is proportional to the current velocity. This means that the signal from the wheel speed sensor will not have a fixed sample rate if the velocity varies, which is most likely in practical applications.

2.2.1 Quantization Error in the Rotational Speed Sensor

One important parameter in the discretization of the voltage signal is the clock frequency, $T_{s,Clock}$. The clock frequency determines the time interval between checking for events (tooth passing the sensor). The more often it is checked the more correct will the time instants be logged. Obviously, there is a quantization error present which is due to the limited clock frequency

$$t_k = t_k^0 + q_k \tag{2.16}$$

where t_k^0 is the true time instant for the event and q_k is an uncorrelated sequence of random variables with distribution $U(0, T_{s,Clock})$. This is not completely true, for situations when q_k is known it is possible to compute $q_{k+i},\ i > 0$ if $\omega(t)$ is known, i.e., q_k is not a random variable. In reality $\omega(t)$ is not known and there are also unknown disturbances, which makes the assumption of uncorrelated quantization

noise to a good working approximation. The statistical properties of q_k are

$$\mathrm{E}(q_k) = \frac{T_{s,Clock}}{2} \tag{2.17}$$

$$\mathrm{Var}(q_k) = \frac{(T_{s,Clock})^2}{12} \tag{2.18}$$

The distribution of the random variable is independent of the current velocity and its influence on ω_k^c in (2.15) will thus be more pronounced when the velocity increases, i.e., the sample interval Δt_k decreases. Let us analyze the influence of the quantization error when the angular velocity is computed with (2.15).

$$\omega_k^c = \frac{\alpha}{t_k - t_{k-1}} = \frac{\alpha}{t_k^0 + q_k - t_{k-1}^0 - q_{k-1}} = \frac{\alpha}{\Delta t_k^0 + q_k - q_{k-1}} \tag{2.19}$$

Using a first-order Taylor approximation, (2.19) can be approximated with

$$\omega_k^c = \omega_k^0 \frac{1}{1 + \frac{q_k - q_{k-1}}{\Delta t_k^0}} \approx \omega_k^0 \left(1 - \frac{q_k - q_{k-1}}{\Delta t_k^0}\right) \tag{2.20}$$

where ω_k^0 is the angular velocity without quantization error. The first and second order statistical properties for ω_k^c are

$$E(\omega_k^c) \approx \omega_k^0 \tag{2.21}$$

$$Std(\omega_k^c) \approx \frac{T_{s,Clock}}{\sqrt{6}} \frac{\omega_k^0}{\Delta t_k^0} = \frac{T_{s,Clock}}{\sqrt{6}} \frac{(\omega_k^0)^2}{\alpha} \tag{2.22}$$

Equations (2.21)-(2.22) can also be found by applying Gauss' approximation formula for two stochastic variables (see Gut, 1995). The computed velocity is unbiased, but the standard deviation is quadraticly proportional to the true velocity and linearly proportional to the clock interval, $T_{s,Clock}$.

2.2.2 Minimum Velocity

In the previous section it was stated that it is desirable to use as high clock frequency as possible. There is also a drawback in using a very high clock frequency. The counter used in the microprocessor has a limited number of counter values, C_{max}. When the counter reaches its maximum value it starts again from zero. If the clock frequency is too high the counter may complete more than one counter cycle, i.e., the difference between two consecutive logged counter values is

$$c_k - c_{k-1} = \Delta c_k^0 + nC_{max} \tag{2.23}$$

where Δc_k^0 is the true difference and n an arbitrary number. To avoid this situation there is a minimum velocity implying that at least one counter value is logged each counter cycle. This minimum velocity is given by

$$\omega(t) \geq \frac{2\pi}{L \cdot C_{max} \cdot T_{s,Clock}}. \tag{2.24}$$

We see that there is a contradiction between minimizing the quantization error and allowing as low velocity as possible. The clock frequency becomes a design parameter to the user.

Example 2.1 Minimum Velocity

To get an idea of the minimum velocity and the statistical properties of ω_k due to the quantization a simple counter example is used. Assume that $L = 48$, $C_{max} = 2^{16}$ and $T_{s,Clock} = 400 \cdot 10^{-9}$ s. The minimum velocity is then $\omega_k = 5$ [rad/s] and $Std(\omega_k) \approx 10^{-7}(\omega_k^0)^2$. ❑

2.2.3 Unideal Toothed Wheel

Equation (2.15) proposes a way to compute the angular velocity. Let us take one step back and consider the continuous time wheel speed signal $\omega(t)$ and for simplicity assume that it contains two components, true angular velocity $\omega^0(t)$ and noise $e(t)$,

$$\omega(t) = \omega^0(t) + e(t). \tag{2.25}$$

An implicit way of defining the sampling instants, t_k, is to consider an integral of $\omega(t)$ from t_{k-1} to t_k, (2.26). If the toothed wheel is ideal this integral is equal to $2\pi/L$. In practice, in most cases this is not true. There are small errors, δ_i (see Figure 2.4) in the air gap or in the dimension, i.e., in the distance between teeth i and $i-1$. This error is due to production tolerances and wear and tear, (Poisson, 1992). To highlight the periodic behavior of the mechanical errors the notation $\delta_{k\text{mod}L}$ is used.

$$\frac{2\pi}{L} + \delta_{k\text{mod}L} = \int_{t_{k-1}}^{t_k} \omega(t)\, dt \tag{2.26}$$

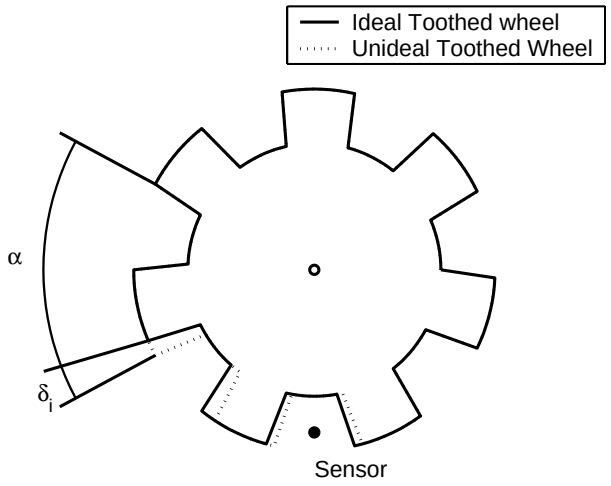


Figure 2.4 *Rotational speed sensor device with sensor errors δ_i.*

Define

$$\omega_k \triangleq \frac{1}{t_k - t_{k-1}} \int_{t_{k-1}}^{t_k} \omega(t)\, dt \tag{2.27}$$

$$\omega_k^0 \triangleq \frac{1}{t_k - t_{k-1}} \int_{t_{k-1}}^{t_k} \omega^0(t)\, dt \tag{2.28}$$

$$e_k \triangleq \frac{1}{t_k - t_{k-1}} \int_{t_{k-1}}^{t_k} e(t)\, dt \tag{2.29}$$

where ω_k, ω_k^0 and e_k are the event domain sampled signals of $\omega(t)$, $\omega^0(t)$ and $e(t)$, respectively. One way to write (2.26) is now

$$\omega_k = \omega_k^0 + e_k = \frac{2\pi/L + \delta_k \text{mod} L}{\Delta t_k} = \omega_k^c + \frac{\delta_k \text{mod} L}{\Delta t_k} \tag{2.30}$$

where $\Delta t_k = t_k - t_{k-1}$ and ω_k^c is the angular velocity computed as in (2.15). Usually δ_i is unknown and (2.15) is used for computing the angular velocity, which will cause a periodic error

$$\omega_k^c = \omega_k - \frac{\delta_k \text{mod} L}{\Delta t_k} \tag{2.31}$$

In Example 2.2 the periodic errors in the angular velocity, ω_k^c, using (2.15) are illustrated.

Example 2.2 Periodic error

Assume a constant noise free angular velocity $\omega_k = 56$ and a toothed wheel with $L = 5$. The largest error, δ_i, is about 4% of the true angle, α. In Figure 2.5 the computed angular velocity, ω_k^c, using (2.15) is shown. In the figure the periodic disturbance originating from δ_i is obvious. ❑

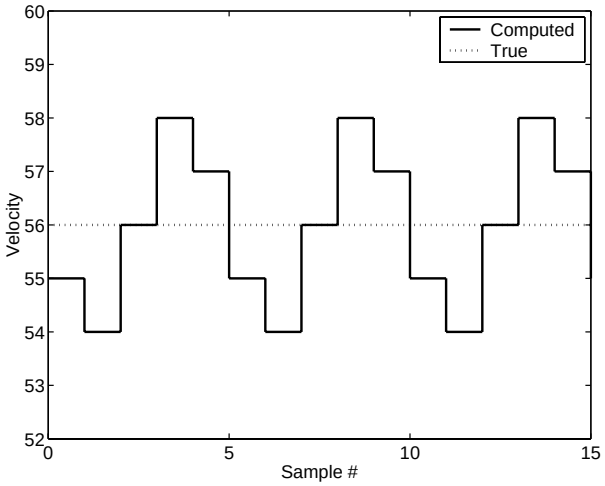


Figure 2.5 *True, ω_k, and computed, ω_k^c, angular velocity for an unideal toothed wheel.*

In the example the magnitude of the errors was a little bit exaggerated to visualize the effect of the periodic disturbance and even though the errors are small in real applications, they can not be neglected. In practical applications the sensor errors are typically less than 1% of the distance between two consecutive teeth, i.e., $|\delta_i| < \alpha/100$.

In Figure 2.6 the periodogram for ω_k^c from a simulated event based sampled measurement with sensor errors is shown. In the figure frequencies below 2 Hz are not shown due to the large energy in this frequency range (DC component). The periodic disturbance causes harmonics in the frequency domain. The fundamental frequency of the harmonics is equal to the revolution frequency of the wheel, i.e., $f_{fund} = \omega^0/(2\pi)$. In Figure 2.6 the true angular velocity was $\omega^0 = 90\ rad/s$ resulting in a fundamental frequency of 14.3 Hz. All the harmonics are multiples

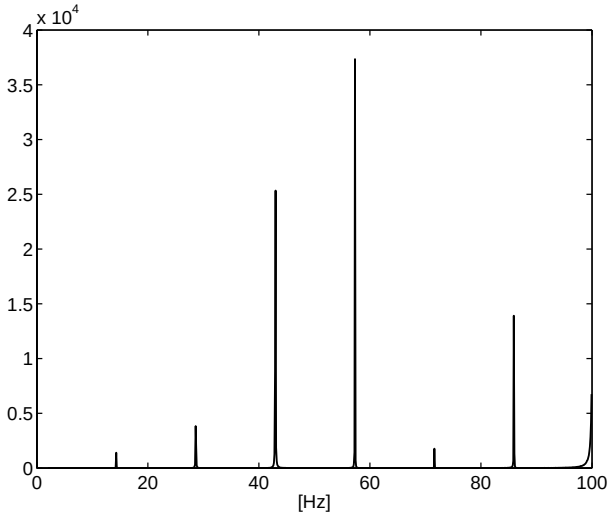


Figure 2.6 *Periodogram for a simulated wheel speed signal* (2.15) *with an unideal toothed wheel.*

of the basic frequency as can be seen in the figure. It is easy to understand that the performance is unacceptable and methods to attenuate the periodic disturbance are needed.

In Example 2.2 the angular velocity was constant and the periodic disturbances were clear harmonics in the frequency domain. The frequency of the harmonics can be computed using

$$f_m = m\omega^0/(2\pi), \quad m = 1, 2, ..., L/2. \tag{2.32}$$

The amplitude of the harmonics will be determined by the magnitude of the sensor errors. In (2.32) there is no DC component, $m = 0$, and this is because the sum of the sensor errors is zero. This can also be seen by computing the DC component of the event based DFT and assuming that the angular wheel speed is constant, i.e., $\Delta t_k \approx \Delta t$

$$Y(0) \approx \sum_{k=0}^{N-1} \omega_k^c = \sum_{k=0}^{N-1} \left(\omega_k + \frac{\delta_{k \bmod L}}{\Delta t} \right) = N\omega_k + \underbrace{\sum_{k=0}^{N-1} \frac{\delta_{k \bmod L}}{\Delta t}}_{r}. \tag{2.33}$$

Because the sum of the sensor errors is zero, $r \ll N\omega_k$ when N increases, and the contribution from r to $Y(0)$ can be neglected. If the angular velocity varies the situation becomes even worse than in Figure 2.6. With varying angular velocity the influence of the errors is distributed in the frequency domain. Therefore the

errors need to be attenuated in order to be able to extract the information in the signal. Two methods will be proposed for attenuating the sensor errors

1. Estimate the sensor errors and use it for an improved computation of the angular velocity.

2. Use a Fourier series to identify the harmonics caused by the periodic error in the event domain.

2.3 Estimation of Sensor Errors in the Event Domain

In the event based sampled signal there are components from both the event and time domain. The signal of our interest is originating from the time domain and we need to know the properties of the signal when it is event based sampled. We also need to know how the attenuation of sensor errors in the event domain affects the signal.

If we start with the continuous time model (2.25) and add a signal component, $s(t)$, it can be written as

$$\omega(t) = \omega^0(t) + s(t) + e(t) \tag{2.34}$$

where $\omega^0(t) \gg s(t)+e(t)$. The distinction between $\omega_0(t)$ and $s(t)$ is that $s(t)$ can be seen as a disturbance with some useful information while $\omega^0(t)$ is the velocity of the axle or wheel. Let us recall the implicit definition of the sampling instants (2.26) and apply it to the signal $s(t)$ to define

$$s_k \triangleq \frac{1}{t_k - t_{k-1}} \int_{t_{k-1}}^{t_k} s(t)\, dt. \tag{2.35}$$

The continuous time model for the angular velocity above can now be discretized to the event domain according to (2.27)–(2.29)

$$\frac{2\pi}{L} + \delta_{k \bmod L} = \int_{t_{k-1}}^{t_k} \omega(t)\, dt \triangleq \omega_k \Delta t_k = \Delta t_k(\omega_k^0 + s_k + e_k). \tag{2.36}$$

An estimate of ω_k^0 can be accomplished by utilizing one complete wheel revolution

$$\begin{aligned} \hat{\omega}_k^0 &= \frac{2\pi}{\Delta t_k^{rev}} \\ \Delta t_k^{rev} &= t_{k+L/2} - t_{k-L/2}. \end{aligned} \tag{2.37}$$

This can be seen as the average velocity in the interval $[t_{k-L/2}, t_{k+L/2}]$. By using a complete revolution the sum of the sensor errors will be zero and do not affect the estimation of the mean velocity. The estimation of $\hat{\omega}_k^0$ is non-casual causing a time delay and the reason for choosing a non-casual revolution time is described in Section 2.3.1. For time critical applications the revolution time can be chosen as $\Delta t_k^{rev} = t_k - t_{k-L}$. The properties of $\hat{\omega}_k^0$ will be further discussed in Section 2.3.1. Collecting the computable signals from (2.36) and using (2.37) as an estimation of ω_k^0 results in

$$y_k = \Delta t_k(\omega_k^c - \hat{\omega}_k^0) \approx \Delta t_k(s_k + e_k) - \delta_{k \bmod L}. \tag{2.38}$$

The objective is to estimate the sensor errors $\delta_i, i = 1, \ldots, L$ and use the estimates to correct the computed angular velocity, (Persson and Gustafsson, 2001). There is no correlation between the sensor errors except for the fact that

$$\sum_{i=1}^{L} \delta_i = 0 \tag{2.39}$$

i.e., the sum of all sensor errors is equal to zero. The estimation of sensor errors therefore treats the errors more or less individually. In practice this means that the signal from the rotational speed sensor is divided into L individual subsequences, one for each tooth pair

$$[y_k, y_{k-1}, ...] = \begin{cases} y^1 &= [y_k, y_{k-L}, ...] \\ y^2 &= [y_{k-1}, y_{k-1-L}, ...] \\ \vdots & \vdots \quad \vdots \\ y^L &= [y_{k-L+1}, y_{k-L+1-L}, ...]. \end{cases} \tag{2.40}$$

Dividing the signal from the sensor according to (2.40) eliminates the periodic behavior of the sensor errors. In each subsequence the sensor error can be treated as an offset. The sample period in each y^i will now be equal to the time for one complete wheel revolution.

The attenuation of sensor errors needs to be done in the event domain and the question is what happens with the time domain signal component $s(t)$ when it is converted to the event domain s_k and divided in L subsequences as in (2.40), i.e., decimated with a factor L.

Example 2.3 Conversion from time domain to event domain

Let us use signal model (2.34) with $\omega^0(t) = 50$ rad/s, i.e., $\Delta t_k = (2\pi/L)/\omega^0(t)$ and a signal component $s(t) = \sin(2\pi f t)$ where $f = 45$. We want to illustrate

what happens with s_k when it is decimated as in (2.40). The continuous time signal is event based sampled without sensor errors and the computable signal is denoted y_k

$$y_k = \omega_k^c - \hat{\omega}^0 = s_k + e_k. \tag{2.41}$$

Every L:th sample of y_k is picked and the resulting sampling frequency is approximately $f_s = \omega^0(t)/(2\pi) = 50/(2\pi) \approx 7.96$ Hz. Since the signal component is not fulfilling the sampling theorem it will be folded into the frequency interval $[0,\ 3.98]$ Hz. An effect of the sampling theorem is that frequencies in the interval $[3.98,\ 7.96]$ Hz will be folded backwards and frequencies in the interval $[7.96,\ 11.94]$ Hz forwards and so on. If the sample period is constant this can also be seen in Poissons-sum formula

$$Y_k(e^{i2\pi f/f_s}) = \sum_{r=-\infty}^{\infty} Y(i(2\pi f + r2\pi f_s)) \tag{2.42}$$

where Y_k denotes the DTFT of y_k and Y the Fourier transform of $y(t)$. The signal component of 45 Hz will then be folded to the frequency 2.75 Hz. In Figure 2.7 the

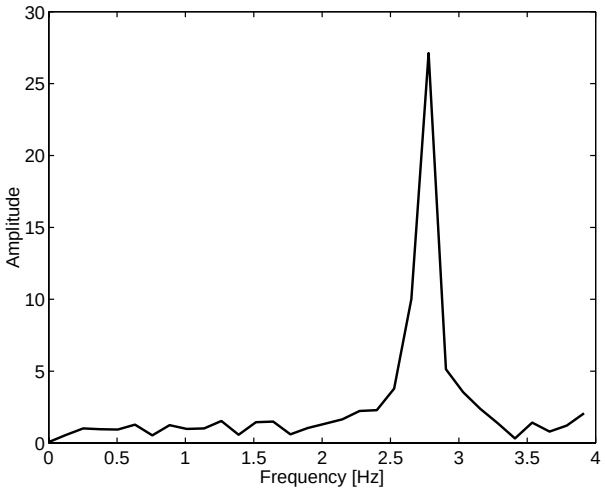


Figure 2.7 *Periodogram for the time domain rotational speed signal* (2.34) *without sensor errors. The continuous time signal is sampled and decimated in the event domain,* (2.41)

periodogram of the decimated signal y_k in the event domain is shown. The figure verifies that the signal component is folded to the frequency computed above. ❑

From the example we can see that folding of the signal component $s(t)$ depends on the current angular velocity, or in other words the sample period for an individual tooth. This means that the folded signal component can be folded to any frequency between zero and the Nyquist frequency. One particular problem occurs when the signal component is folded close to zero.

Depending on the current velocity $\omega(t)$ the signal $s(t)$ is folded into the frequency range determined by the velocity, $[0,\ \frac{\omega(t)/(2\pi)}{2}]$ Hz. Obviously, there are velocities where the folded signal s_k and the sensor errors intersect. The example below illuminates the occurrence of this problem.

Example 2.4 Intersection of s_k and sensor errors

For simplicity let us assume that $s(t)$ is a sinusoidal with $f = 45$, ($s(t) = \sin(2\pi f t)$) and $\omega^0(t)$ is constant. The angular velocity range is $\omega^0(t) \in [10, 180]$ rad/s. Figure 2.8 visualizes the folding of $s(t)$ into the event domain s_k. We see

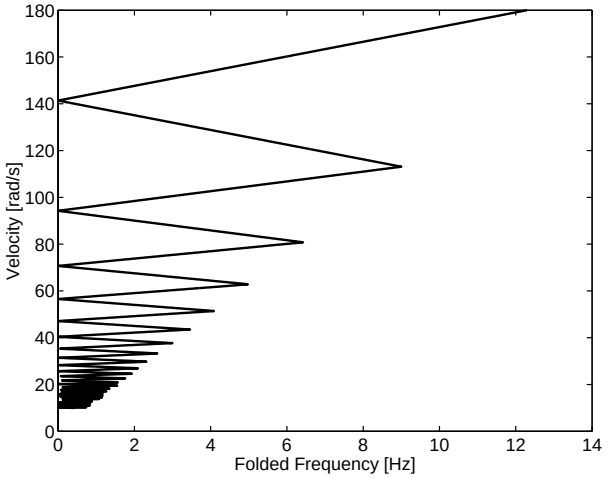


Figure 2.8 *Folded frequency of $s(t)$ as a function of the angular velocity when $s(t)$ is sampled once per revolution in the event domain.*

that $s(t)$ is folded to the low frequency range numerous of times when the angular velocity varies from $10 - 180$ rad/s. The velocity interval between these occurrences is proportional to the velocity itself. In low velocities (up to 60 rad/s) the interval is less than 10 rad/s. ❑

In the event domain the sensor error is low frequent and the signal component and sensor error intersect. When attenuating the sensor error the signal component can be attenuated as well.

2.3.1 Estimation Properties

The sensor errors are estimated individually, i.e., the signal is divided into L subsequences according to (2.40). All estimation procedures are equivalent and we focus on one subsequence y^i. Let the index n_i denotes samples in subsequence i

$$n_i = i + kL + 1, \quad \begin{cases} k & = 0, 1, 2, \dots \\ i & = 0, 1, \dots L-1. \end{cases} \tag{2.43}$$

For notational simplicity the subscript i will not be used in the sequel. Including the error in the estimation of ω_n^0 and collecting the computable quantities on the left hand side (compare with (2.38)) the measurement for one subsequence can be written as

$$\begin{aligned} \alpha - \hat{\omega}_n^0 \Delta t_n &= -\delta_n + \Delta t_n (s_n + e_n) + \Delta t_n (\omega_n^0 - \hat{\omega}_n^0) \\ \underbrace{\alpha - \frac{2\pi}{\Delta t_n^{rev}} \Delta t_n}_{y_n} &= -\delta_n + \Delta t_n s_n + \Delta t_n \underbrace{(\omega_n^0 - \hat{\omega}_n^0)}_{\tilde{\omega}_n^0} + \Delta t_n e_n = -\delta_n + r_n \end{aligned} \tag{2.44}$$

In (2.44) Δt_n denotes the time difference between the two consecutive time instants in the original signal ($t_k - t_{k-1}$) and Δt_n^{rev} denotes the revolution time ($t_{k+L/2} - t_{k-L/2}$) used for estimating ω_n^0. δ_n denotes the sensor error in subsequence i. δ_n can be treated as a constant and is sometimes denoted without index.

The measurement y_n is the difference between the ideal angle $\alpha = 2\pi/L$ and the measurement of the real angle, i.e., a measurement of the sensor error δ. The measurement of the real angle is computed by using the mean velocity of the last revolution (sum of the sensor errors is zero) and the time elapsed between the two last teeth, Δt_n. From the measurements in the left hand side of (2.44) the idea is to estimate $\hat{\delta}_n$ as the "level of y_n". In the right hand side of (2.44) we can see that y_n is not only a measurement of the sensor error but also of $r_n = \Delta t_n (s_n + \tilde{\omega}_n^0 + e_n)$, which will affect the estimation of $\hat{\delta}_n$.

Let us begin with considering $\Delta t_n s_n$ which is defined according to (2.35). Assume that the continuous time signal is a sinusoidal, $s(t) = A \sin(2\pi f t)$. Eval-

uating the integral in (2.35) results in

$$\begin{aligned}\Delta t_n s_n &= \frac{A}{2\pi f}(\cos(2\pi f t_{n-1}) - \cos(2\pi f t_n)) \\ &= \frac{A}{\pi f}\sin(2\pi f \frac{t_n - t_{n-1}}{2})\sin(2\pi f \frac{t_n + t_{n-1}}{2}) \\ &= \frac{A}{\pi f}\sin(2\pi f \frac{\Delta t_n}{2})\sin(2\pi f \bar{t}_n)\end{aligned} \tag{2.45}$$

where $\bar{t}_n = \frac{t_n + t_{n-1}}{2}$. For simplicity, assume that the sample period is given by

$$\Delta t_n \approx \frac{2\pi/L}{\omega_n^0} \tag{2.46}$$

i.e., proportional to the inverse of the angular velocity. This is a good approximation since $\omega_n^0 \gg s_n + e_n$ (see (2.34)). Assume also that the angular velocity is large enough so that the first order approximation

$$\frac{A}{\pi f}\sin(2\pi f \frac{\Delta t_n}{2}) \approx A\Delta t_n \tag{2.47}$$

is valid. If $f = 45$ Hz this is a good approximation for $\omega^0(t) > 45$ rad/s. Equation (2.45) can thus be written as

$$\Delta t_n s_n \approx A\Delta t_n \sin(2\pi f \bar{t}_n) \tag{2.48}$$

From (2.48) $|\Delta t_n s_n|$ is bounded,

$$|\Delta t_n s_n| \leq A\Delta t_n \tag{2.49}$$

The worst case scenario

$$|\Delta t_n s_n| = A\Delta t_n, \quad \forall n \tag{2.50}$$

only occurs when the sample period is constant and f is a multiple of Δt_n. This is exactly the velocities in Figure 2.8 where $s(t)$ is folded to DC. In real applications it is very unlikely that the angular velocity is constant at exactly the velocities where $s(t)$ is folded to DC. For all other velocities and when the velocity varies the mean value of $\Delta t_n s_n$ applies to

$$\frac{1}{N}\sum_{n=1}^{N}\Delta t_n s_n \approx \frac{1}{N}A\sum_{n=1}^{N}\Delta t_n \sin(2\pi f \bar{t}_n) \to 0 \quad \text{when} \quad N \to \infty. \tag{2.51}$$

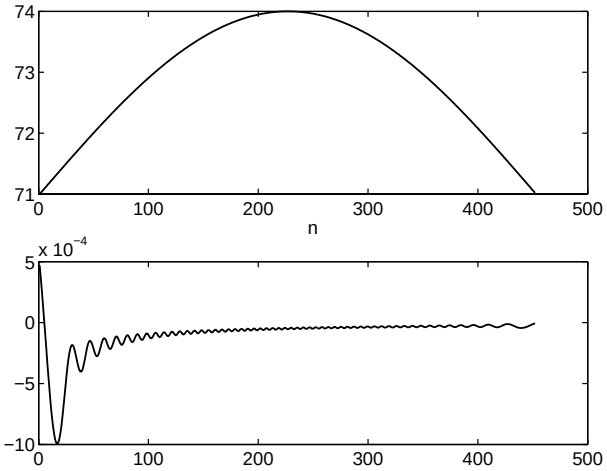


Figure 2.9 *In the top figure the "true" angular velocity ω_n is plotted. In the bottom figure, $\frac{1}{N}\sum_{n=1}^{N}\Delta t_n s_n$ is evaluated as a function of N.*

The convergence rate depends on the sample instants t_n and is beyond the scope of this thesis. An example of the convergence rate is shown in Figure 2.9. The example indicates that the mean value of $\Delta t_n s_n$ is significantly smaller than the sensor error after about $N = 100$ samples (revolutions). Unfortunately, it is not sufficient that the velocity varies for (2.51) to hold. One example is if the velocity is periodic. The remedy is that in real applications where the velocity varies in a random way (2.51) will hold in most cases.

Let us continue by considering the error in estimating the velocity

$$\begin{aligned} \Delta t_n \tilde{\omega}_n^0 &= \Delta t_n(\omega_n^0 - \hat{\omega}_n^0) \\ \hat{\omega}_n^0 &= \frac{2\pi}{\Delta t_n^{rev}}. \end{aligned} \tag{2.52}$$

Assume that the sample period Δt_n is given by (2.46). For constant velocities there is no error in estimating the velocity and $\Delta t_n \tilde{\omega}_n^0 = 0$. This is not exactly true, the signal s_n and e_n should be included in the error term $\Delta t_n \tilde{\omega}_n^0$. To simplify the analysis, s_n and e_n are neglected. This is a good approximation since $\omega_n^0 \gg s_n + e_n$. It should also be noted that s_n and e_n are avaraged over one revolution, which also strenghten the approximation. Next, the velocity can be approximated with piecewise constant acceleration during one revolution

$$\omega_n^0 = \omega^0 + n\dot{\omega}^0 \tag{2.53}$$

where ω^0 is the velocity at $n = 0$ and $\dot{\omega}^0$ is the constant acceleration. When the acceleration is constant and non-zero there will be a bias in the estimate. An

example where the acceleration is about 6 rad/s^2 is shown in Figure 2.10. In the figure the true velocity, ω_n^0, and estimation error, $\Delta t_n \tilde{\omega}_n^0$ are shown as a function of N. From the figure it can be seen that the maximum error in the example is less

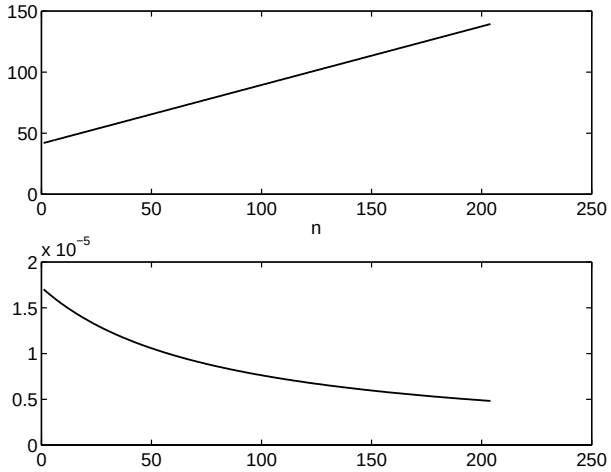


Figure 2.10 *In the top figure the "true" angular velocity ω_n is plotted. In the bottom plot, $\Delta t_n \tilde{\omega}_n^0$ is visualized.*

than $2 \cdot 10^{-5}$, which is approximately less than 1% of the largest sensor error. If the revolution time is chosen from the last revolution $\Delta t_k^{rev} = t_k - t_{k-L}$ the estimation error is approximately 10 times larger than in Figure 2.10. In real applications when the acceleration will be both positive and negative the estimation error

$$\frac{1}{N}\left|\sum_{n=1}^{N} \Delta t_n \tilde{\omega}_n^0\right| < \varepsilon \quad \text{when} \quad N \to \infty \tag{2.54}$$

will be small. As mentioned above ε is approximately less than 1% of the largest sensor error.

The last term in (2.44) is the noise $\Delta t_n e_n$ where e_n is assumed to have zero mean. In this case the mean value

$$\frac{1}{N}\sum_{n=1}^{N} \Delta t_n e_n \to 0 \quad \text{when} \quad N \to \infty. \tag{2.55}$$

will approach zero. Because Δt_n and e_n are mutually independent (2.55) will approach zero for all possible velocity profiles.

Even though the above discussion uses a number of approximations and assumptions it verifies that the sensor error can be estimated as the "level" of y_n in (2.44). Important to remember is that the rest term

$$\frac{1}{N}\sum_{n=1}^{N} r_n = \frac{1}{N}\sum_{n=1}^{N} \Delta t_n(s_n + \tilde{\omega}_n^0 + e_n) \tag{2.56}$$

will not approach zero under all conditions. In real applications though, the assumptions is valid in most cases and the rest term will approach zero.

The sensor errors might vary slowly due to wear and tear. In a signal processing perspective, non-uniformities in the process such as an imbalanced toothed wheel or rotating axle can also be seen as sensor errors. The effect of imbalance is typically dependent on the actual angular velocity. Therefore it is preferable to use a recursive method such as RLS, LMS or a Kalman filter (see Gustafsson, 2000; Hayes, 1996). For the rest term in (2.56) to be significantly smaller than the sensor error δ the number of samples N considered in the estimation needs to be sufficiently large. In a recursive algorithm this is equivalent to choosing the adaptive rate sufficiently small. The linear regression model for estimating all sensor errors is

$$\begin{array}{rcl} y_k & = & 2\pi/L - \hat{\omega}_k^0 \Delta t_k = \varphi_k^T \theta + \varepsilon_k \\ \theta & = & (\delta_1 \delta_2 \cdots \delta_L)^T \\ \varphi_k & = & (0 - 1 \;\; 0 \cdots 0) \quad \text{pos } k \bmod L \text{ is } -1 \\ \hat{\omega}_k^0 & = & \frac{2\pi}{t_{k+L/2} - t_{k-L/2}} \\ \sum_{i=1}^{L} \delta_i & = & 0 \end{array} \tag{2.57}$$

In (2.57) $k = 0, 1, 2, ...$ and φ_k divides the model in L parts. This is a compact way of defining the linear regression model for the sensor errors instead of defining L equivalent models. In (2.57) the sum of all sensor errors is constrained to be zero. To deal with this constraint either a virtual measurement, y_k^v

$$y_k^v = \sum_{i=1, i \neq k \bmod L}^{L} \hat{\delta}_i \tag{2.58}$$

or constrained least squares (Kay, 1993, pp. 251–254) can be used. In Example 2.5 the linear regression model is used for estimation of sensor errors.

Example 2.5 Estimation of sensor errors

The simulation uses a discrete time model approximation of the continuous time model (2.34) with $\omega^0(t) = 70$ rad/s, $s(t)$ is a second order AR model with a

resonance peak at 46.6 Hz, and $e(t)$ is Gaussian noise with standard deviation 0.01.

The RLS estimation is based on the linear regression model (2.57) with a forgetting factor close to 1. The mean value of each recursively estimated sensor error is shown in Figure 2.11 along with the estimation errors. The standard deviation of the estimated sensor errors are small relatively the magnitude of estimation. From

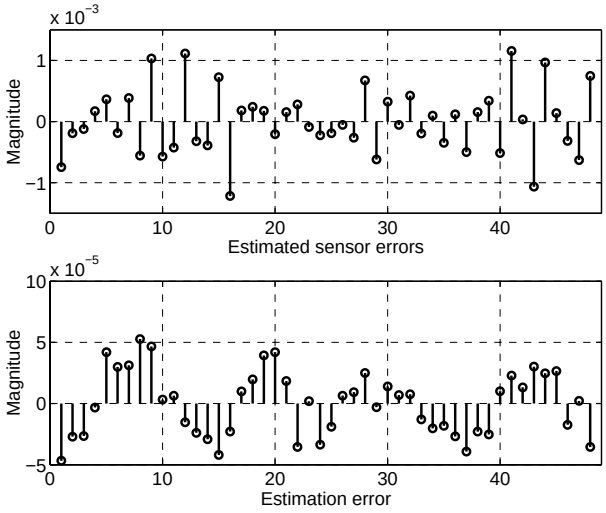


Figure 2.11 *In the top figure the estimated sensor errors $\hat{\delta}_i$ are plotted and below the estimation errors are shown.*

the figure it can be seen that the estimation of sensor errors works quite well and the largest estimation error is less than $5 \cdot 10^{-5}$. ❑

The example indicates that the estimation of sensor errors works well. The idea was to use the estimated sensor errors in the computation of the angular speed. Instead of using the ideal angle, α, to compute the angular velocity as in (2.15) the estimated sensor errors are used as

$$\hat{\omega}_k^c = \frac{\alpha + \hat{\delta}_k}{\Delta t_k} = \frac{2\pi/L + \hat{\delta}_k}{\Delta t_k} \approx \omega_k = \omega_k^0 + s_k + e_k. \tag{2.59}$$

In most applications there is no need to know the magnitude of the sensor errors. What is important is that the influence from the errors in the computed velocity is rejected. In Figure 2.12 the periodograms for the simulation data in Example 2.5

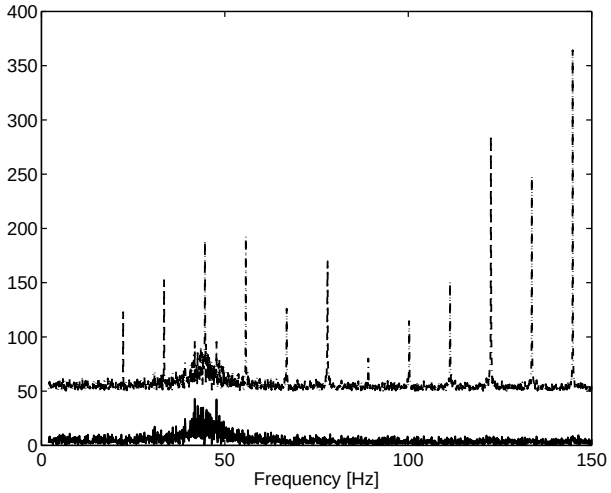


Figure 2.12 *Periodogram for the velocity computed as* (2.15) *(dash dotted) and using the estimated sensor errors as in* (2.59) *(solid). A constant is added to the periodogram of the original signal to separate the periodograms.*

using (2.15) and (2.59) are shown. From the figure it can be seen that the influence of sensor errors are almost completely rejected while the signal component (around 46 Hz) remains unmodified. In Figure 2.13 only frequencies between $[35 - -55]$ Hz are shown.

2.4 Fourier Series Expansion

The second method is similar to noise cancellation methods used in communication systems (Widrow et al., 1975). The idea is to remove only the harmonics that may interfere with the signal $s(t)$. From Equation (2.32) we know that the frequency of the harmonics are

$$f_k(m) = m\omega_k/(2\pi) \approx m/(L\Delta t_k), \quad m = 1, 2, ..., L/2. \tag{2.60}$$

If the energy of $s(t)$ is in the interval $[30,\ 60]$ Hz we want to remove the harmonics only in this interval. For example, if the velocity is $\omega_k = 70$ rad/s, only the harmonics $m = 3, 4, 5$ should be removed. This corresponds to $f_k(m) \approx 33, 44, 55$ Hz, respectively.

The model used for rejection of the harmonics is similar to the one used for

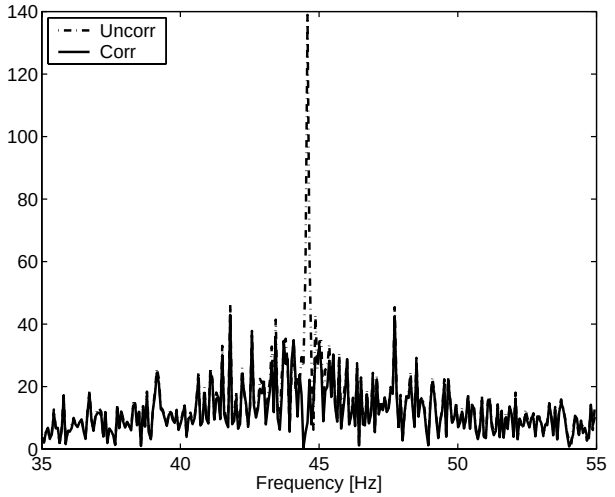


Figure 2.13 *Same periodogram as Figure 2.12, but only frequencies in the interval* $[35 - -55]$ *Hz are plotted*

estimating the sensor errors

$$\underbrace{\omega_k^c - \hat{\omega}_k^0}_{y_k} = \frac{-\delta_k \text{mod} L}{\Delta t_k} + s_k + e_k + \tilde{\omega}_k^0 = \frac{-\delta_k \text{mod} L}{\Delta t_k} + r_k. \tag{2.61}$$

Instead of estimating the sensor errors, they are considered as a time series in the event domain. By using Fourier series expansion , the sensor errors can be written as

$$\delta_k \text{mod} L = a_0 + \sum_{m=1}^{\lfloor L/2 \rfloor} a_m \cos\left(m\frac{2\pi}{L}k\right) + \sum_{m=1}^{\lfloor (L-1)/2 \rfloor} b_m \sin\left(m\frac{2\pi}{L}k\right). \tag{2.62}$$

In (2.62) the coefficient $a_0 = 0$ due to the fact that the sum of the sensor errors is zero. The frequencies $mk2\pi/L$ of the fourier series coefficients corresponds to a time series in the event domain with sample period $T_e = 1$. In Figure 2.14(a) the DFT of the sensor error time series with $L = 7$ is shown. To relate this to a frequency in the time domain consider a constant sample period $\Delta t_k = T_t = 0.01$, i.e., constant angular velocity. In Figure 2.14(b) the frequency grid is corrected with respect to the sample period T_t. According to (2.60) the frequency of the first harmonic ($m = 1$) is approximately 14.3 Hz, which can also be seen in the figure. The event domain frequencies in (2.62) correspond to the frequencies given by (2.60), i.e., the event domain Fourier series coefficients a_m and b_m correspond to the time domain harmonics $f_k(m)$.

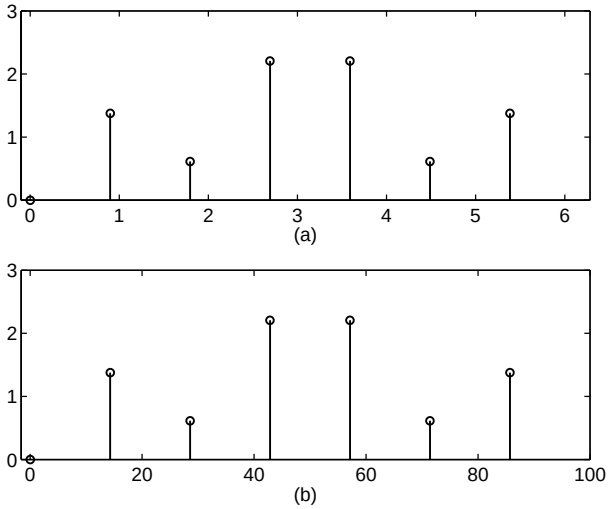


Figure 2.14 *In (a) the DFT of a sensor error time series with frequency grid in the event domain is shown. In (b) the same DFT is shown, but with frequency grid in Hz corresponding to the time domain.*

Rewriting (2.61) using the Fourier series expansion (2.62) results in

$$y_k = \frac{1}{\Delta t_k}\left(\sum_{m=1}^{\lfloor L/2 \rfloor} a_m \cos\left(m\frac{2\pi}{L}k\right) + \sum_{m=1}^{\lfloor (L-1)/2 \rfloor} b_m \sin\left(m\frac{2\pi}{L}k\right)\right) + r_k. \tag{2.63}$$

We are only interested in estimating the coefficients m corresponding to frequency interfering with the signal of interest. Define the index set of these coefficients as $I_m = \{m|\text{interesting coefficients}\}$. The linear regression model for estimating the interesting coefficients is

$$\begin{aligned} \theta &= [a_m, \quad b_m]^T, \quad m \in I_m \\ \varphi_k &= \frac{1}{\Delta t_k}\left[\cos\left(m\frac{2\pi}{L}k\right), \quad \sin\left(m\frac{2\pi}{L}k\right)\right]^T, \quad m \in I_m \\ \varepsilon_k &= y_k - \varphi_k^T\theta_k \\ \theta_{k+1} &= \theta_k + \mu_k\varphi_k\varepsilon_k \end{aligned} \tag{2.64}$$

where the dimension of θ and φ depends on the size of I_m. Any recursive estimation method such as RLS, LMS or Kalman filter can be used to estimate the Fourier series coefficients. Note that the dimension of θ depends on the number

of harmonics to be removed. Important though, is to keep the dimension of θ as small as possible to save both computational complexity and memory.

This method might seem to be similar to the method based on estimation of the sensor errors. The difference is that in this method the frequency component determined by the frequency of the harmonics is directly estimated. This can also be seen when the estimated Fourier series coefficients are used to compute the corrected angular velocity

$$\hat{\omega}_k = \omega_k - \varphi_k^T \theta_k. \tag{2.65}$$

The largest distinction between the methods is that this method is not dividing the signal into L parts, i.e., the discrete signal is not decimated.

To evaluate and visualize the performance the same simulation as in Example 2.5 is used. The angular velocity is 70 rad/s and the harmonics $f_k(m),\ m = 3, 4, 5$ are estimated with *Least Mean Squares (LMS)*. In Figure 2.15 the periodograms with and without rejection of harmonics are shown. In the figure it can

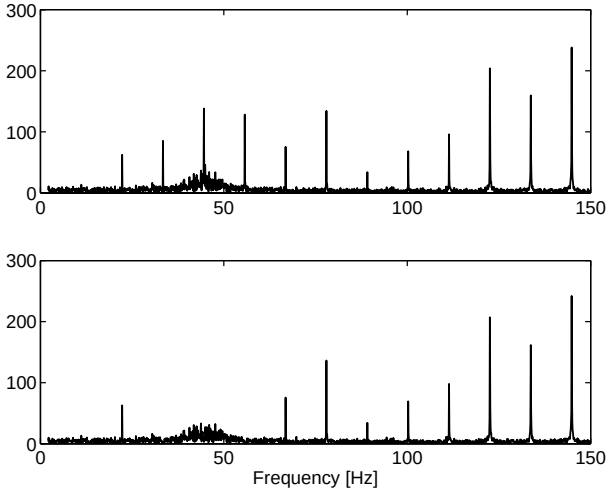


Figure 2.15 *Periodogram for the original velocity* (2.15) *(top) and using frequency rejection* (2.64)-(2.65) *for harmonics* $f_k(m),\ m = 3, 4, 5$ *(below).*

be seen that only the specified harmonics are removed and the rest remains unmodified. In Figure 2.16 only frequencies between $35 - 55$ Hz are shown. We see that almost only the frequencies specified for the harmonics are removed. When the velocity varies the index set I_m needs to be updated in order to fulfill the requirement that harmonics in the interval interfering with s must be removed.

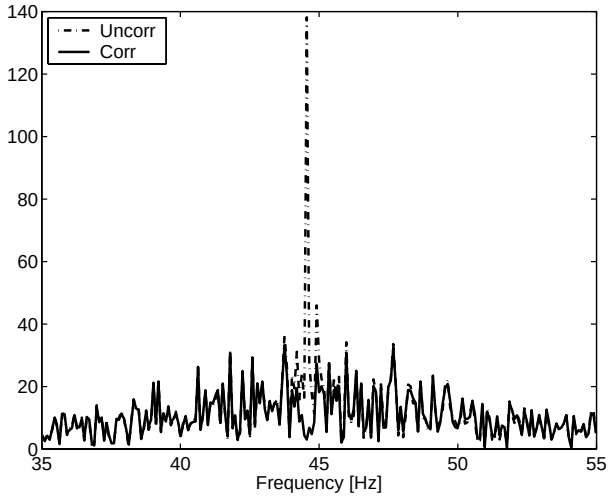


Figure 2.16 *Same periodogram as Figure 2.15, but only frequencies in the interval* $[35 - 55]$ *Hz are plotted.*

2.5 Summary of Attenuating Sensor Errors

In the previous sections two different methods to attenuate sensor errors were proposed. The first method can be seen as an auto calibration method, estimating sensor errors. The second method uses a method similar to noise cancellation in communication systems. Only harmonics that may interfere with the signal of interest are removed.

2.5.1 Properties

Before the performance of the methods are further evaluated the important properties are listed in Table 2.1. In the table *AC* denotes the Auto Calibration method (estimation of sensor errors) and *FS* the Fourier Series method. The FS suffers from some undesirable properties. In FS only the harmonics intersecting with the resonance frequency is attenuated. In Chapter 3 when the event based signal is converted to the time domain the remaining harmonics cause problems in terms of aliasing. The most attractive alternative is to use AC where the computed angular velocity is corrected with respect to the estimated sensor errors.

2.5.2 Computational Complexity and Memory Usage

Another important aspect is the computational complexity and memory usage. These method are supposed to be fitted into an embedded system micro proces-

	AC	**FS**
Estimation of sensor errors	Yes	No
Divides signal into L parts	Yes	No
A corrected velocity as output	Yes	Yes
Harmonics to be removed can be chosen	No	Yes

Table 2.1 *Properties of the two proposed methods for sensor error attenuation. AC = Auto Calibration and FS = Fourier Series*

sor and therefore it is necessary to keep both the computational complexity and memory usage minimal. For comparison *LMS* is applied to estimate the L sensor errors and M harmonics are removed in *FS*. In Table 2.2 memory usage refer to the number of states needed to be stored between each update and *Add* and *Mult* denote the number of additions and multiplications in each update, respectively. In

	AC	**FS**
Memory Usage	$L+1$	$2M$
Add	6	$6M+3$
Mult	4	$12M+4$

Table 2.2 *Computational complexity and memory usage for updating AC and FS for one new sample. AC = Auto Calibration and FS = Fourier Series*

the table it can be seen that the memory usage and number of additions and multiplications depend on the design parameter M in FS, while in AC it is constant.

3

Interpolation

The history of interpolation goes back to the ancient Babylon and Greece (Meijering, 2002). In antiquity, interpolation was used for making predictions concerning astronomical events. This served practical needs for e.g., farmers who based their planting strategies on these predictions. During the years lots of methods such as linear interpolation and spline interpolation, (see Oppenheim and Schafer, 1975; Unser, 1999) have been evolved. The development in the field of image processing has resulted in numerous methods for interpolation, (Hou and Andrews, 1978; Park and Schowengerdt, 1982).

In this chapter we want to transform the event based sampled data to time based sampled data using linear in data interpolation. The objective is to find an interpolation routine which is computationally efficient and fulfills the specifications defined in Section 3.1. In Sections 3.2–3.4 three different interpolation routines are described and in Section 3.5 the proposed routines are evaluated with respect to the performance specifications.

3.1 Problem Formulation

In Chapter 2 event based sampled signals are described. A continuous time signal $x(t)$ is sampled when certain pre-defined events occur. This results in an event based signal $x(t_k)$, where t_k denotes the time instants for the events. The sample period for $x(t_k)$ varies from sample to sample and the objective here is to find an

interpolation routine

$$x(t_k) \overset{\text{Interpolation}}{\Longrightarrow} y(nT_{int}) \tag{3.1}$$

where T_{int} is the sample period for the interpolated signal and $k = 1, 2, 3, \ldots, K$ is the index for the event based signal signal. The number of interpolated samples $(n = 1, 2, \ldots)$ will be given by the interpolation sample period T_{int} and the time instants for the event based signal. The desired performance specifications for the interpolation routine are listed below.

1. Minimum computational complexity.
2. Minimum aliasing of frequencies above the Nyquist frequency $1/(2T_{int})$.
3. Minimum distortion of frequencies below the Nyquist frequency.

The last two items can be interpreted as an ideal lowpass filter. There is a contradiction between item 1 and items 2–3. A reasonable trade-off is needed, which is dependent on the actual application. When the sample period is constant, 2 is usually solved by applying a digital lowpass filter with proper cut-off frequency prior to the decimation. When the signal is event based sampled this is not possible due to the varying sample period. Two alternatives to fulfill 2 are presented in Sections 3.2–3.4. The first alternative, presented in Section 3.2, is to choose an initial sample period, which is small enough to avoid aliasing. Prior to decimation a lowpass filter is applied to avoid aliasing. The second alternative, presented in Sections 3.3 and 3.4, incorporates a lowpass filter in the interpolation routine where the cut-off frequency of the lowpass is user defined.

In this thesis, we will only focus on linear in data interpolation. Here, the problem is to find proper weights $w(i)$ in

$$y(nT_{int}) = \sum_i w(i, n)x(t_i) \tag{3.2}$$

where i is defined by the method used. Sometimes this is also referred to as nonparametric regression, (Rasche et al., 1999; Stenman, 1999; Thevenaz et al., 2000).

From a computational complexity perspective linear in data interpolation is an attractive alternative and is the most common method (Meijering, 2002) to interpolate data. The interpolation (3.2) can be made more or less complicated depending on the application and specifications to be fulfilled. The weights are often constrained according to

$$\sum_i w(i) = 1. \tag{3.3}$$

This will guarantee an unbiased interpolation if $x(t_i)$ is constant.

3.2 Linear Interpolation

The simplest alternative is to use linear interpolation, i.e., use only two weights in (3.2). We want to find the interpolated value $y(nT_{int})$, where nT_{int} fulfills

$$t_{k-1} < nT_{int} < t_k \tag{3.4}$$

by using

$$y(nT_{int}) = \gamma x(t_{k-1}) + (1-\gamma)x(t_k). \tag{3.5}$$

The weights in (3.5) are determined by the distances between the sample instants t_k and t_{k-1} and the interpolation time nT_{int}

$$\gamma = \frac{t_k - nT_{int}}{t_k - t_{k-1}} = \frac{t_k - nT_{int}}{\Delta t_k} \tag{3.6}$$

Because the signal $x(t_k)$ is event based sampled the weights will vary from sample to sample. Two problems will occur using (3.5) for event based sampled data.

First the sample period Δt_k for $x(t_k)$ can be larger than T_{int}, i.e., more than one interpolated sample will be computed in each interpolation step. The interpolation routine then introduces a linear relationship between consecutive samples.

Second when the sample period Δt_k is much smaller than T_{int} information in $x(t_k)$ will be lost using (3.5). Some samples will not be considered in the interpolation and just thrown away.

Typically (3.5) is very computational efficient. The drawback is that it allows aliasing of frequencies above the Nyquist frequency. One remedy is to use a higher interpolation frequency followed by lowpass filtering and decimation. If the interpolation frequency is p times the desired, the routine can be described with

1. Use (3.5) with $T_{int} = T_{desired}/p$.
2. Apply a lowpass filter with normalized cut-off frequency $1/p$ on $y(nT_{int})$.
3. Decimate the filter output by a factor p.

This routine reduces the aliasing, but increases the computational complexity.

The main advantage with linear interpolation is that it is computational efficient. The disadvantage is that the performance in some situations deteriorates.

3.3 Kernel Based Interpolation

To extend the interpolation routine presented in the previous section more weights can be added in (3.2). Instead of a fixed number of weights a surrounding neighborhood in time is fixed.

Assume that we want to compute $y(nT_{int})$ using samples in a surrounding of nT_{int}. In other words, we want to find the weights in (3.2) for all t_i fulfilling

$$T_{int} - d < t_i < T_{int} + d \tag{3.7}$$

where d is defining the surrounding neighborhood of interest. Define the index set

$$I_n = \{i | nT_{int} - d \leq t_i < nT_{int} + d\} \tag{3.8}$$

corresponding to the time instants in the neighborhood of nT_{int}. The weights are chosen from the *Epanechnikov* kernel (Epanechnikov, 1969)

$$w_d(t) = \frac{3}{4d}(1 - \frac{t^2}{d^2})_+ \tag{3.9}$$

modified with respect to the size of the neighborhood d and $(\cdot)_+$ denotes the positive part. The Epanechnikov is depicted in Figure 3.1 for $d = 1$. The interpolation

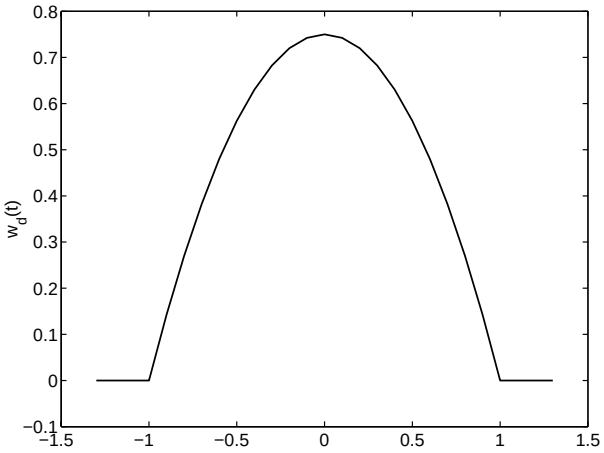


Figure 3.1 *Epanechnikov kernel* (3.9) *with* $d = 1$.

using samples from a neighborhood around nT_{int} and Epanechnikov kernel (3.9) results in

$$y(nT_{int}) = \frac{\sum_{i \in I} w_d(t_i - nT_{int}) x(t_i)}{\sum_{i \in I} w_d(t_i - nT_{int})}. \tag{3.10}$$

Note that the interpolation also includes a normalization to fulfill (3.3). The crucial design parameter is the size of the surrounding neighborhood, d. Sometimes this is also referred to as the bandwidth. A large d tends to "over-smooth" and a small d tends to "under-smooth". In the first case important information might be lost and in the latter the interpolation might be noisy. In Figure 3.2 the frequency function for two different choices of the design parameter d are illustrated. In the figure it can be seen that the Epanechnikov kernel is a lowpass filter, where d is some sense determines the cut-off frequency. The advantage with kernel interpolation is

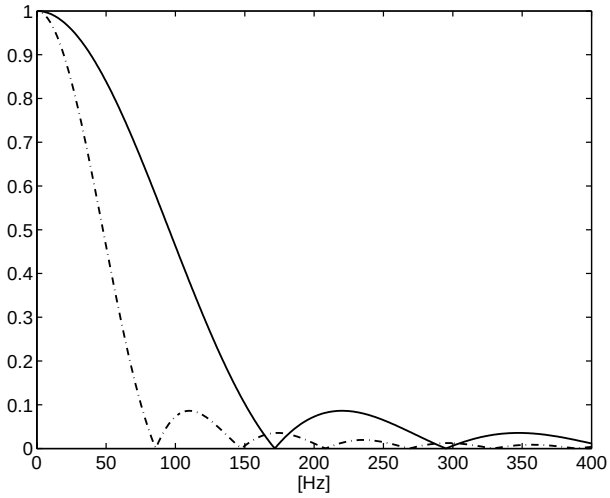


Figure 3.2 *Frequency function of* (3.9) *for* $d = 1/240$ *(solid) and* $d = 1/120$ *(dash-dotted)*

that it is based on a simple kernel, which implicitly incorporates a lowpass filter. The disadvantage is that the computational complexity increases when the number of samples in the surrounding neighborhood increases. This happens when the angular velocity of the rotational sensor increases.

3.4 Filter Based Interpolation

On the analogy of pre-filtering and sampling in AD-converters the idea is to apply an "analog" filter to the event based data in the interpolation routine. It is, of course, not possible to apply a continuous time filter to discrete data, but an approximation can be used.

Why use a time continuous filter in the interpolation? Incorporating a con-

tinuous lowpass filter with proper cut-off frequency into the interpolation reduces the aliasing without increasing the interpolation frequency. This can be achieved by finding proper weights in (3.2), but it is desired to create a framework, which finds the weights automatically. Here, we will only consider lowpass filters, but the framework can easily be extended to other filter structures such as high-pass and band-pass.

Let us start with a continuous filtering, which is equivalent with multiplication in the frequency domain

$$Y(s) = H(s)X(s) \tag{3.11}$$

where $X(s)$ is the Laplace transform of $x(t)$ and $H(s)$ is the frequency function of the filter. In the time domain (3.11) is written as a convolution

$$y(t) = \int_{-\infty}^{\infty} h(t-\tau)x(\tau)\, d\tau \tag{3.12}$$

where $x(\tau)$ is the signal to be filtered and $h(\tau)$ is the impulse response of the filter. In (3.12) it is assumed that the signal is known for all t. In practice the signal is only known from $\tau = 0$ to $\tau = t$. (3.12) can then be rewritten as

$$y(t) = \int_{0}^{t} h(t-\tau)x(\tau)\, d\tau. \tag{3.13}$$

In digital applications the continuous time signal is sampled at discrete time instants and (3.13) needs to be modified to be applicable for a discrete time signal. First we assume that $x(t_k)$ is sampled with a constant sample period T_s. By numerical integration (Riemann integration) (3.13) is approximated with

$$\begin{aligned} y(t_k) &= \sum_{i=0}^{k} T_s h(t_k - t_i)x(t_i) = \sum_{i=0}^{k} T_s h((k-i)T_s)x(iTs) \\ &= T_s \sum_{i=0}^{k} h(k-i)x(i) \end{aligned} \tag{3.14}$$

which results in the classical convolution formula.

Example 3.1 Discrete approximation of continuous filtering

Consider a continuous time second order Butterworth filter

$$\begin{aligned} H(s) &= \frac{1}{1 + a_1 \frac{s}{f_0} + a_2 (\frac{s}{f_0})^2} \\ a_1 &= \sqrt{2}, \;\; a_2 = 1 \end{aligned} \tag{3.15}$$

with cut-off frequency $f_0 = 2\pi 50$. The impulse response of (3.15) is

$$\begin{aligned} h(t) &= \frac{\sqrt{2}}{f_0} e^{-\alpha t} \sin(\beta t), \quad t > 0 \\ \alpha &= \beta = \frac{f_0}{\sqrt{2}} = \frac{2\pi 50}{\sqrt{2}}. \end{aligned} \tag{3.16}$$

Inserting the impulse response in (3.14) results in the filtering routine

$$\begin{aligned} y(t_k) &= \sum_{i=0}^{k} T_s \frac{\sqrt{2}}{f_0} e^{-\alpha(t_k - t_i)} \sin(\beta(t_k - t_i)) x(t_i) \\ &= \sum_{i=0}^{k} T_s \frac{\sqrt{2}}{f_0} e^{-\alpha(k-i)} \sin(\beta(k-i)) x(i). \end{aligned} \tag{3.17}$$

❑

In (3.14) and (3.17) the sample period was assumed to be constant. From Chapter 2 we know that the sample period for an event based signal will vary. It is also desirable to include interpolation in the filtering procedure by means of evaluating (3.14) at the time instants defined by the interpolation, nT_{int}

$$y(nT_{int}|t_k) = \int_0^{t_k} h(nT_{int} - \tau) x(\tau)\, d\tau. \tag{3.18}$$

In most cases the time instants of the interpolation will not be equal to any of the original time instants t_k. Two alternatives are possible

1. Use future data, i.e., $t_k > nT_{int}$ (non-casual filter, smoothing)
2. Do not use future data, i.e., $t_k < nT_{int}$ (casual filter, prediction)

Here, we will only focus on the second alternative using casual filters. First, (3.18) needs to be discretized. In the non-casual case the sinc-function can be used, which is ideal if the band-width of the signal is limited. The drawback is that the sinc-function decays very slowly to zero.

To reduce the computational complexity a fixed number of weights, M, will be used, i.e., fixed number of components in the discretized impulse response. The normalized interpolation routine based on a continuous filter is

$$\begin{aligned} y(nT_{int}|t_k) &= \frac{\sum_{i=k-M}^{k} h(nT_{int} - t_i) \Delta t_i x(t_i)}{\sum_{i=k-M}^{k} h(nT_{int} - t_i) \Delta t_i} \\ h(nT_{int} - t_i) &\equiv 0 \ \text{ if } \ nT_{int} - t_i < 0 \end{aligned} \tag{3.19}$$

where $\Delta t_i = t_i - t_{i-1}$. This can be seen as approximating an IIR filter with a FIR filter. Important though, is to choose the design parameter M sufficiently large, which means that $h(nT_{int} - t_{k-M})$ is close to zero. This can also be formalized with

$$|h(nT_{int} - t_{k-m})| < \varepsilon, \quad \forall m > M. \tag{3.20}$$

where ε is sufficiently small. It is also trivial to see that (3.19) is equivalent to (3.2) and (3.3), but with weights corresponding to the normalized impulse response of the continuous filter.

The benefit with filter based interpolation is that the cut-off frequency of the interpolation filter can easily be chosen by the user. This reduces undesirable aliasing. The disadvantage is that when M is large it is not computational efficient.

3.5 Evaluation

The proposed methods are aiming at choosing weights, $w(i)$, in (3.2) such that the specifications are fulfilled. In this section the performance of the proposed interpolation routines are evaluated. For evaluation purposes a continuous time signal $x(t)$ is defined as

$$x(t) = \omega(t) + s(t) + d(t) + e(t) \tag{3.21}$$

where $\omega(t)$ is a slowly varying signal (compare with main rotational velocity), $s(t)$ is the information signal (compare with resonance frequency), $d(t)$ is a high-frequency disturbance and $e(t)$ is white noise. The desired signal is defined as

$$x_d(t) = \omega(t) + s(t). \tag{3.22}$$

The continuous time signal $x(t)$ is sampled non-equidistantly according to the rotational speed sensors described in Section 2.2 and the sample instants, t_k, are determined by solving the integral equation

$$k\alpha = \int_0^{t_k} \omega(t)\, dt \tag{3.23}$$

for $k = 1, \ldots, K$ where α is a predetermined constant (usually a $2\pi/L$). The signals $x(t_k)$ and t_k are the inputs to the proposed interpolation routines.

3.5.1 Performance

By comparing the the desired signal sampled uniformly, $x_d(nT_{int})$, with the interpolated signal $y(nT_{int})$

$$V = \frac{1}{N-\tau} \sum_{n=1}^{N-\tau} (x_d(nT_{int}) - y(nT_{int} + \tau))^2 \tag{3.24}$$

where τ is an integer corresponding to the time delay (phase shift) in the interpolation, a measure of the performance is obtained. The time delay caused by non-casual filtering will not be considered.

Example 3.2 Performance Evaluation

Consider the continuous time signal model (3.21) with

$$\begin{aligned} \omega(t) &= 50 + 5\sin(2\pi 0.05t) \\ s(t) &= \sin(2\pi 40t) \\ d(t) &= \sin(2\pi 90t) \\ \mathrm{Var}(e(t)) &= 1 \end{aligned} \tag{3.25}$$

The continuous time signal $x(t)$ is sampled at the sampling instants given by (3.23). The sample period for each k is visualized in Figure 3.3 The sampling period for the interpolation is $T_{int} = 1/120$, i.e., the disturbance $d(t)$ will be aliased while the desired signal fulfills the sampling theorem. In the kernel based interpolation $d = T_{int}$ is chosen. The filter based interpolation is implemented as a second-order Butterworth filter with cut-off frequency $55\ Hz$ and $M = 10$ coefficients in the impulse response approximation (see Example 3.1). In Table 3.1 the performance measure (3.24) from 20 Monte Carlo simulations, each realization includes $N = 2400$ interpolated date, is used to evaluate the performance of the proposed interpolation routines. ❑

The interpolation routines display similar results even if the design parameters in the continuous time model are changed. The performance of filter based interpolation deteriorates when the constant term in $\omega(t)$ is too large. This is because M is not sufficiently large. The impulse response is too much truncated. There is of course a trade-off between computational complexity and performance. One idea is to let also M depend on the actual sample instants.

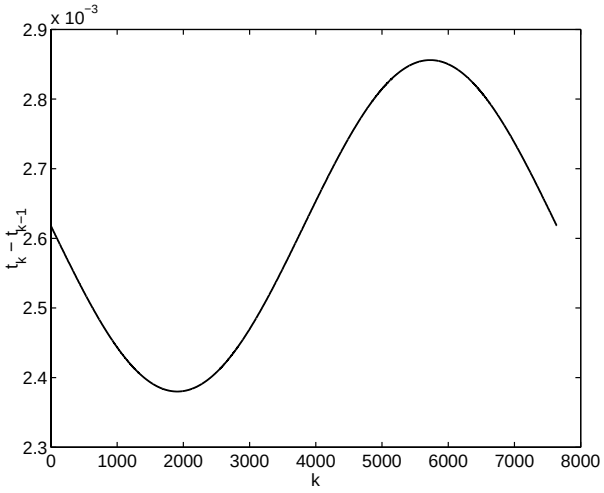


Figure 3.3 *Sample period as a function of the sample number k for the continuous time signal $x(t)$.*

Method	V	τ
Linear Interpolation	4.14	0
Oversampled Linear Interpolation	1.34	1
Kernel Based Interpolation	0.81	0
Filter Based Interpolation	0.65	2

Table 3.1 *Performance measure* (3.24) *for 20 Monte Carlo simulations each of length $N = 2400$.*

3.5.2 Computational Complexity

One of the specifications was to use minimum computational complexity. This is in contradiction with the performance specification evaluated in (3.24). Unfortunately, the computational complexity can not be expressed as the number of additions and multiplications required to compute each interpolated value. In kernel based interpolation the computational complexity is dependent on the number of samples, K, in surrounding neighborhood of the sample point, nT_{int}. In filter based interpolation the computational complexity is dependent of the number of coefficients, M, in the impulse response approximation.

In Table 3.2 the number of additions and multiplications for each interpolation step are presented. For kernel based and filter based interpolation the computa-

tional complexity is expressed in terms of K and M, respectively. As can be seen

Method	**Add**	**Mult**
Linear Interpolation	4	3
Oversampled Linear Interpolation	16	16
Kernel Based Interpolation	4K - 1	3K + 1
Filter Based Interpolation	3M - 1	6M - 1

Table 3.2 *Computational complexity for the proposed interpolation routines.*

in the table, the computational complexity increases quite rapidly for fitler based interpolation when M increases. In the example where $M = 10$, filter based interpolation requires too many computations. For kernel based interpolation the problem is that the computational complexity increases when the sample period $t_k - t_{k-1}$ decreases. For rotational sensors this is equivalent to increased angular velocity, which is an undesirable behavior. To reduce the computational complexity kernel based and filter based interpolation can be implemented with table look-up. In kernel interpolation the values of the Epanechnikov kernel, (3.9) can be implemented with a table and in filter based interpolation the impulse response can be implemented with a table. In Table 3.3, the computational complexity when kernel based and filter based interpolation is implemented with a table look-up, is presented. Especially the number of multiplications in each interpolation step is

Method	**Add**	**Mult**
Kernel Based Interpolation	3K - 1	K + 1
Filter Based Interpolation	3M - 1	2M - 1

Table 3.3 *Computational complexity for kernel based and filter based interpolation when implemented with table look-up.*

significantly reduced. In the example $K \approx \frac{1/T_{int}}{t_k - t_{k-1}} \approx 4$, which makes it more computational efficient than the oversampled linear interpolation, even without table look-up implementation.

3.5.3 Summary

Three different interpolation routines based on (3.2) and (3.3) have been presented. They all aim at fulfilling the specifications in Section 3.1, which are contradictory.

The first method, linear interpolation, fulfills the specification of minimum computation complexity best, but it does not manage to avoid aliasing.

The second method, kernel based interpolation, manage to reduce the aliasing, but it also requires more computations than linear interpolation. The benefit of the kernel interpolation is that it is based on a simple kernel function (Epanechnikov).

The last method, filter based interpolation, shows similar performance as kernel based interpolation. The advantage is that the user can easily choose the cut-off frequency of the lowpass filter incorporated in the interpolation routine. This is at the expense of increased computational complexity.

One way of reducing the computational complexity for kernel based and filter based interpolation is to implement the weights with a table look-up.

4

Spectral Estimation

The area of linear estimation is a mature field (see Ljung, 1999; Söderström and Stoica, 1989) and this chapter is not aiming at contributing new ideas. Instead the aim is to enlighten some common estimation methods and estimation properties of a specific system. The system is modeled as a second order *Auto-Regressive (AR)* model with an additive noise floor. The system is supposed to contain a resonance in the rotational velocity, (Persson, 2002).

Usually identification or estimation aims at finding a model that best fits the real data. Here the focus is somewhat different. The most important issue is to estimate the frequency of the resonance as accurate as possible. The estimate is used to detect changes in the system. This can of course also be done by finding a model that best fits data, but the algorithms are supposed to be fitted into a micro processor with minimal resources. Therefore it is needed to keep the computational complexity and memory usage as low as possible. If the computational complexity is reduced by using methods only focusing on the estimation of the resonance frequency, data fit will be ignored.

Section 4.1 presents the simulation model used throughout the chapter. In Sections 4.2 non-parametric methods to estimate the resonance frequency is discussed, and in Section 4.3 parametric methods are treated. The chapter ends with a summary and proposal for preferred method in Section 4.4.

4.1 Simulation Model

The simulation model in Figure 4.1 is used to generate data similar to a resonance in a real wheel speed signal. The simulation model includes a system, S, representing the resonance in the rotational velocity and an additive noise floor representing all external disturbances. This is a quite simple model, but will enlighten some

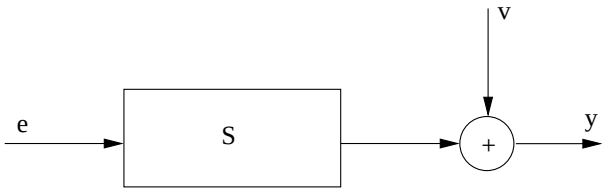


Figure 4.1 *Schematic overview of simulation model.*

of the properties and problems involved when estimating a resonance in a wheel speed signal. Here, a second order AR-model is used to represent the resonance and the model can be described by

$$\begin{aligned} y(t) =& \frac{1}{A(q)} e(t) + v(t) \\ A(q) =& 1 + a_1 q^{-1} + a_2 q^{-2} \\ \mathrm{E}\, e^2 =& \lambda_e, \quad \mathrm{E}\, v^2 = \lambda_v \end{aligned} \tag{4.1}$$

where e and v are mutually independent white Gaussian noise sequences and $a_i,\ i = 1, 2$ are chosen to resemble the spectrum of the real resonance. An example of a real spectrum is shown in Figure 4.2(b). The resonance is located around $40 - 45$ Hz. The simulation model can be converted to an equivalent *Auto Regressive Moving Average* ARMA(2,2) model

$$\begin{aligned} y(t) &= \frac{\tilde{B}(q)}{A(q)} \tilde{e}(t) \\ \tilde{B}(q) &= 1 + \tilde{b}_1 q^{-1} + \tilde{b}_2 q^{-2} \end{aligned} \tag{4.2}$$

with the same poles as (4.1), but for this study we want to keep the original structure. The spectrum of (4.1) is

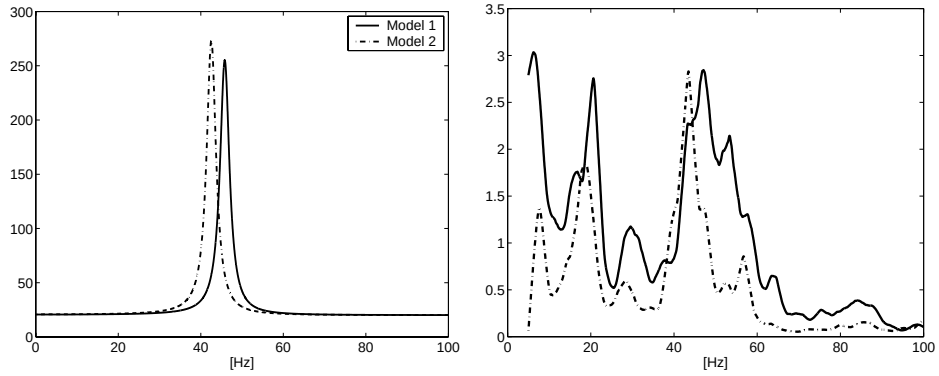
$$\Phi_y(f) = \frac{\lambda_e + \lambda_v \mid A(e^{i2\pi T_s f}) \mid^2}{\mid A(e^{i2\pi T_s f}) \mid^2} \tag{4.3}$$

Two different parameter settings for the model will be used. The first model, *M1*, is supposed to represent a normal functionality while *M2* is a system with erroneous functionality. The aim is to evaluate the possibility to seperate the models in terms of the estimated resonance frequency. The model parameters are chosen to

$$M1: \quad \begin{cases} A^1 &= [-0.7 \quad 0.93] \\ f_{res}^1 &= 45.81\ Hz \end{cases} \tag{4.4}$$

$$M2: \quad \begin{cases} A^2 &= [-0.85 \quad 0.93] \\ f_{res}^2 &= 42.57\ Hz \end{cases} \tag{4.5}$$

where $f_{res}^i,\ i = 1, 2$ are the resonance frequencies for M1 and M2, respectively. $\lambda_e = 1$ will be chosen in all simulations while λ_v will vary to simulate different SNR. The sample frequency used is $T_s = 240$ Hz. $\Phi_y(f)$ for the two simulation models are shown in Figure 4.2(a). In the figure it seems to be a simple task to find



(a) (b)

Figure 4.2 *In (a) system spectrum with $\lambda_v = 20$ for model 1 and 2 is shown. In (b) smoothed periodograms for two real wheel speed signals between are shown.*

the resonance peak, but later we will see that this is not always the case. Comparing with Figure 4.2(b) it can also be seen that replacing all external disturbances with a noise floor is a major model simplification. The reason for this is twofold.

1. We are interested in estimating the resonance in a well-defined case.
2. To be able to perform analytic calculations.

4.2 Non-parametric Spectral Estimation

A very useful tool to analyze the frequency content in a digital signal is to estimate the power spectrum. There are several methods to estimate the spectrum, but in this section we will focus on non-parametric methods (see Hayes, 1996; Stoica and Moses, 1997).

One of the advantages with non-parametric methods is that no prior information about the statistical properties of the signal is necessary. The most common method is to use the *Fast Fourier Transform (FFT)*, which is an computational efficient implementation of the *Discrete Fourier Transform (DFT)*. The DFT and its inverse for a signal, x, with a finite length of N is defined as

$$X(n) = \sum_{k=0}^{N-1} x(k) e^{-i\frac{2\pi n}{N}k} \tag{4.6}$$

$$x(k) = \frac{1}{N}\sum_{n=0}^{N-1} X(n) e^{i\frac{2\pi k}{N}n} \tag{4.7}$$

$$k, n = 0, 1, \ldots, N-1.$$

If the signal $x(k)$ is sampled with period T_s, $X(n)$ will be defined on the frequency points $f(n) = n/(NT_s)$ [Hz].

One way to estimate the location of the resonance frequency is to use

$$\hat{n} = \arg\max_{n\in[0,N-1]} |X(n)|^2 \tag{4.8}$$

$$\hat{f}_{res} = f(\hat{n}) = \frac{\hat{n}}{NT_s}. \tag{4.9}$$

Equations (4.8) – (4.9) are just a way of finding the largest component in the DFT and transform it to the right frequency with respect to the sample period.

Example 4.1 Non-parametric spectral estimation

In the example M1, (4.4), is used to generate $N = 8192$ data with $\lambda_v = 20$. In Figure 4.3 the periodogram for one realization is shown along with the true power spectrum. As can be seen in the figure the estimated power spectrum approximates the true quite well. The prime concern was to estimate the resonance frequency, i.e., the frequency location of the peak. For this realization the estimated resonance frequency is $\hat{f}_{res} = 44.79$ Hz, i.e., a bias $\tilde{f}_{res} = 1.02$ Hz. Even though the estimated spectrum visually is similar to the true, the proposed method to estimate the resonance frequency is sensitive to leakage and resolution. To evaluate the

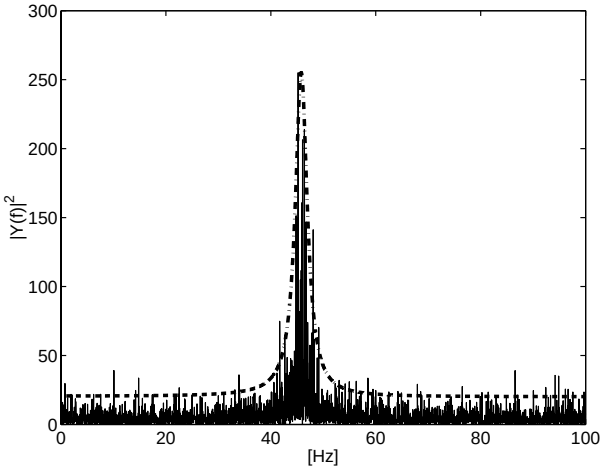


Figure 4.3 *True power spectrum (dash dotted) and estimated power spectrum (solid) for a realization of M1.*

statistical properties of (4.8) – (4.9), Monte Carlo simulations with 1000 realization for both M1 and M2 is performed. The result is presented in Table 4.1. From the

MC	**Model 1**	**Model 2**
f_{res}	45.81	42.57
$\mathrm{Mean}(\hat{f}_{res})$	45.81	42.56
$\mathrm{RMSE}(\hat{f}_{res})$	0.59	0.58
$\min(\hat{f}_{res})$	43.91	40.58
$\max(\hat{f}_{res})$	47.49	44.91

Table 4.1 *Result of* (4.8) – (4.9) *for 1000 Monte Carlo simulations.*

table we see that the proposed method in mean works well, which was expected, but the estimate is quite uncertain. ❏

However, the method is supposed to be fitted into a micro processor with limited amount of memory causing N to be limited. It is also desirable to use recursive estimation of the resonance frequency to attenuate estimation uncertainties instead of increasing the number of data N. It can be utilized by incorporating a forgetting factor, γ, as in

$$\hat{F}_{res}(k) = (1-\gamma)\hat{f}_{res}(k) + \gamma\hat{F}_{res}(k-1) \tag{4.10}$$

where $\hat{F}_{res}(k)$ denotes the filtered estimate from simulations up to k and $\hat{f}_{res}(k)$ is the estimate from simulation k using (4.9). The standard deviation for $\hat{F}_{res}$ assuming a time invariant model is

$$\mathrm{Std}(\hat{F}_{res}) \approx \sqrt{\frac{1-\gamma}{1+\gamma}}\,\mathrm{Std}(\hat{f}_{res}). \tag{4.11}$$

The complete algorithm for estimation of the resonance frequency using a non-parametric method with a forgetting factor is given by

ALGORITHM 4.1 Non-parametric resonance frequency estimation

1. Record N data
2. Compute FFT, (4.6)
3. Find the largest frequency component, (4.8) – (4.9)
4. Update resonance frequency estimate, (4.10).

❑

In Figure 4.4, Algorithm 4.1 is used for 500 realizations, each of $N = 8192$ data, of M1. $\gamma = 0.97$ is used and the $\mathrm{Std}(\hat{F}_{res}) = 0.0051$ is approximately a factor 9 better than in Table 4.1 while (4.11) gives approximately a factor 8. As expected the estimate is still unbiased.

The main disadvantage with the non-parametric algorithm is that it requires a large memory to store at least N data. A trade-off between accuracy and memory usage is needed, but it will not be further treated here.

4.3 Parametric Spectral Estimation

One limitation of non-parametric methods, also one of their advantages, is that they are not designed to incorporate any prior information that may be available about the process. Another disadvantage is that non-parametric methods are designed to work with batches of data rather than recursively. An alternative to non-parametric methods is parametric estimation of the power spectrum, (Ljung, 1999) and (Hayes, 1996). Parametric methods estimate model parameters in a given model structure. Commonly used model structures for time-series are *Auto Regressive (AR)*, *Moving Average (MA)*, and *Auto Regressive Moving Average (ARMA)*. One would hope that incorporating prior knowledge in form of a model structure for the process directly into the estimation increases the accuracy and the resolution.

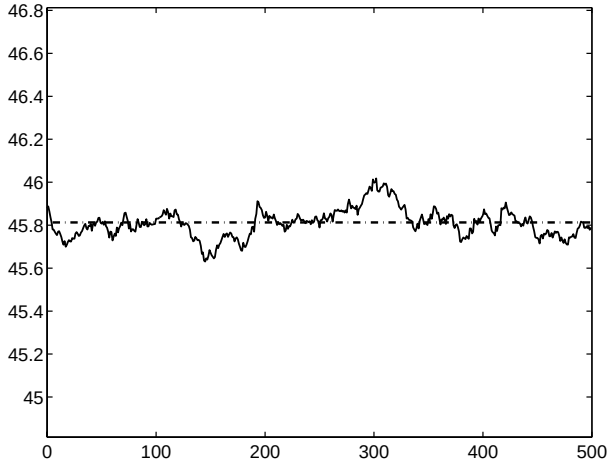


Figure 4.4 *Recursive estimation of resonance frequency using Algorithm 4.1 with* $\gamma = 0.97$.

First a model structure needs to be chosen. This can done by for example examining the power spectrum or by considering physical properties of the process. When a model structure is chosen the next step is to estimate the model parameters. Using least-squares to estimate the model parameters, the interpretation (Ljung, 1999, pp. 202–203) in the frequency domain, assuming the number of data $N \to \infty$, can be seen as

$$\bar{V}_N(\theta) = \int_{-f_s/2}^{f_s/2} \left| \frac{Y(e^{i2\pi T_s f})}{H(e^{i2\pi T_s f}, \theta)} \right|^2 df \tag{4.12}$$

$$\hat{\theta}_N = \arg\min(\bar{V}_N(\theta)) \tag{4.13}$$

where Y is the spectrum of the simulation (true) model and H is the spectrum of the parametric model with parameters θ. If the true model belongs to the model structure, i.e., there exists some θ_0 such that $Y(e^{i2\pi T_s f}) = H(e^{i2\pi T_s f}, \theta_0)$ for almost all f, we get that the limiting estimate corresponds to the true system when $N \to \infty$. The proof and a more thorough treatment can be found in (Ljung, 1999) and (Söderström and Stoica, 1989). When the model parameters are estimated the spectrum can be estimated from

$$\hat{\Phi}_y(f) = \left| H(e^{i2\pi T_s f}, \hat{\theta}_N) \right|^2 \tag{4.14}$$

4.3.1 Using a Proper Model Structure

The prime concern was to estimate the resonance frequency as accurate as possible rather than the power spectrum. For the models used here the location of the resonance frequency only depends on the location of the poles, i.e., the parameters in the $A(q)$ polynomial. The estimated parameters are transformed to a resonance frequency by

$$\hat{f}_{res} = \frac{\arccos\left(-\frac{\hat{a}_1}{2\sqrt{\hat{a}_2}}\right)}{2\pi T_s} \tag{4.15}$$

Example 4.2 Parametric spectrum estimation

Let us use model M1 and M2 to generate $N = 8192$ data with $\lambda_v = 20$. An ARMA(2,2) model structure is chosen,

$$y(t) = \frac{1 + b_1 q^{-1} + b_2 q^{-2}}{1 + a_1 q^{-1} + a_2 q^{-2}} e(t) \tag{4.16}$$

i.e., the true model belongs to the model structure. For one realization the estimated resonance frequency is $\hat{f}_{res} = 46.01$ Hz, i.e., a bias $\tilde{f}_{res} = -0.20$ Hz. To evaluate the statistical properties 100 Monte Carlo simulations for both M1 and M2 are used. In table 4.2 the estimated parameters in the $A(q)$ polynomial using least-squares (4.12) and the resonance frequency using (4.15) are shown. Comparing Table 4.1 with Table 4.2 it can be seen that estimating the resonance

	Model 1	**Model 2**
$\mathrm{Mean}(a_1)$	-0.70	-0.85
$\mathrm{E}(a_2)$	0.93	0.93
f_{res}	45.81	42.57
$\mathrm{Mean}(\hat{f}_{res})$	45.81	42.57
$\mathrm{RMSE}(f_{res})$	0.15	0.15
$\min(f_{res})$	45.31	42.06
$\max(f_{res})$	46.29	43.06

Table 4.2 *Result of* (4.12)–(4.15) *for 1000 Monte Carlo simulations for model 1 and model 2.*

frequency with a parametric model is more accurate than using the non-parametric

method. The estimate from both methods are unbiased, though. This requires that a sufficiently rich model structure is used in the parameter estimation. ❑

4.3.2 Using an Insufficient Model Structure

One interesting question is what happens when the model structure is insufficient, i.e., the true model does not belong to the model structure. In the following sections an insufficient, AR(2), model structure

$$\begin{aligned} y(t) =& \frac{1}{A(q)} e(t) \\ A(q) =& 1 + a_1 q^{-1} + a_2 q^{-2} \end{aligned} \tag{4.17}$$

will be used to estimate the resonance frequency. The reason is that this is (one of the) easiest model structures to estimate with respect to computational complexity and memory usage. The least-squares estimation can also easily be implemented as a recursive estimation using standard methods such as RLS, LMS or a Kalman filter. Another reason is that in reality the "true system" is more complex than an ARMA(2,2) model and to guarantee a sufficient estimation model structure requires a complex high-order model (see Figure 4.2(b)), resulting in increased computation complexity.

Hopefully the AR(2) model is sufficient to find the location of resonance frequency with sufficient accuracy. In the following sections three different methods based on an AR(2) model is used to estimate the resonance frequency of model M1 and model M2, respectively.

4.3.3 Raw Signal Estimation

The simplest approach is to estimate the AR(2) directly from the raw data without any preprocessing. Inserting (4.3) and (4.17) into (4.12) and assuming the number of data $N \to \infty$ the loss function and parameter estimation becomes

$$\begin{aligned} \bar{V}(\theta) &= \int_{-f_s/2}^{f_s/2} \frac{\lambda_e + \lambda_v \left| A(e^{i2\pi T_s f}, \theta_0) \right|^2}{\left| A(e^{i2\pi T_s f}, \theta_0) \right|^2} \left| A(e^{i2\pi T_s f}, \theta) \right|^2 \, df \\ \hat{\theta} &= \arg\min(\bar{V}(\theta)) \\ \theta_0 &= \begin{bmatrix} a_1 & a_2 \end{bmatrix}^T \end{aligned} \tag{4.18}$$

By visual examination of (4.18) it is easy to realize that $\hat{\theta} = \theta^0$ will not minimize the loss function, i.e., there will be a bias caused by the additive noise floor. The

solution of (4.18) is given by the following system of equations

$$\begin{aligned} R_y(0)\hat{a}_1 + R_y(1)\hat{a}_2 &= -R_y(1) \\ R_y(1)\hat{a}_1 + R_y(0)\hat{a}_2 &= -R_y(2). \end{aligned} \tag{4.19}$$

where $R_y(i)$ is the covariance function defined as

$$R_y(i) \stackrel{\Delta}{=} \mathrm{E}\, y(k)y(k-i). \tag{4.20}$$

(4.19) is also called the *Wiener-Hopf equations* and is solved by

$$\hat{\theta} = \begin{bmatrix} \hat{a}_1 \\ \hat{a}_2 \end{bmatrix} = \frac{1}{R_y(0)^2 - R_y(1)^2} \begin{bmatrix} R_y(1)\left(R_y(2) - R_y(0)\right) \\ R_y(1)^2 - R_y(0)R_y(2) \end{bmatrix} \tag{4.21}$$

To transform the estimated model parameters to a resonance frequency (4.15) will be used. The bias is defined as the difference in frequency between the position of the true and the estimated resonance peak

$$\tilde{f}_{res} = f_{res} - \hat{f}_{res} \tag{4.22}$$

rather than difference between true and estimated model parameters.

Example 4.3 Raw signal estimation

Model M1 and M2 are used as simulations models and (4.17) is used as estimation model. In Figure 4.5 the analytic bias in the estimation as a function of the SNR (λ_v) is shown. ❑

Even though it is quite easy to estimate the resonance frequency visually as in Figure 4.2(a) the bias in Figure 4.5 is unacceptable. The degeneration is due to the insufficient model structure and the large frequency distribution of the additive noise floor. The remedy would be to reduce the sample rate. Unfortunately the sample rate is decided from the rotational velocity (see Chapters 2 and 3), and can not be chosen arbitrarily.

4.3.4 Pre-processing Data

The standard procedure to treat the problem with disturbances outside the frequency interval of interest to the system dynamics is to pre-process the data before estimation. Here, we want to isolate the frequency range around the resonance frequency, which can be accomplished by a bandpass filter with appropriate cut-off frequencies.

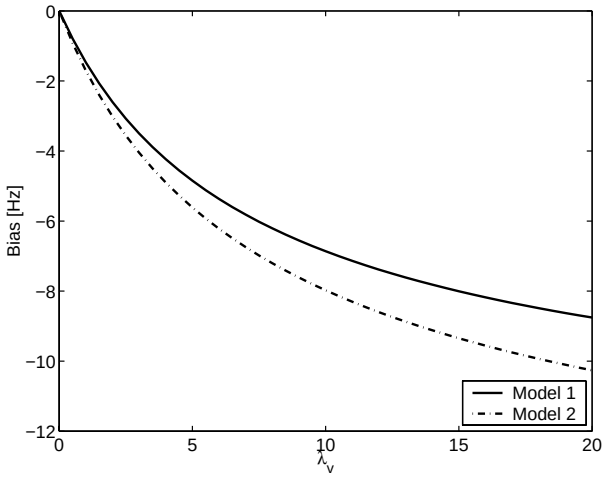


Figure 4.5 *Bias in estimating the resonance frequency.*

From Figure 4.2 it can be seen that the main part of the resonance frequency energy is between 30 and 60 Hz and a bandpass filter with these cut-off frequencies will be used to isolate the resonance frequency in the estimation. If an ideal bandpass filter is used the loss function

$$\begin{aligned} \bar{V}(\theta) &= \int_{30}^{60} \frac{\lambda_e + \lambda_v \mid A(e^{i2\pi T_s f}) \mid^2}{\mid A(e^{i2\pi T_s f}) \mid^2} \mid A(e^{i2\pi T_s f}, \theta) \mid^2 \; df \\ \hat{\theta} &= \arg\min(\bar{V}(\theta)) \\ \theta_0 &= [a_1 \quad a_2]^T \end{aligned} \tag{4.23}$$

only takes the frequencies between 30 and 60 Hz into consideration. Once again we need to solve the Wiener-Hopf equations (4.19) – (4.21), but now y is prefiltered with a bandpass filter. The analytic solution is now much more difficult to find and therefore a numerical solution is used. The covariance function over N data is approximated with

$$\hat{R}_y(i) = \frac{1}{N} \sum_{k=1}^{N} y(k)y(k-i). \tag{4.24}$$

If $y(k)$ is a quasi-stationary signal, which is the case for the simulated data, the approximated covariance function converges to (4.20) w.p.1 when $N \to \infty$, (Ljung, 1999, pp. 33-34). The loss function (4.23) is solved by the Wiener-Hopf equations (4.21), using the approximated covariance function (4.24).

Example 4.4 Parametric spectrum estimation with pre-processed data

1000 Monte Carlo simulations for both model M1 and M2 are performed with $\lambda_v = 20$. A forth order Butterworth bandpass filter was used to pre-filter data. The result is presented in Table 4.4. From the table it can be seen that the estimated

	Model 1	**Model 2**
$\mathrm{Mean}(a_1)$	-0.71	-0.80
$\mathrm{Mean}(a_2)$	0.94	0.94
f_{res}	45.81	42.57
$\mathrm{Mean}(\hat{f}_{res})$	45.68	43.76
$\mathrm{RMSE}(\hat{f}_{res})$	0.13	0.14
$\min(\hat{f}_{res})$	45.32	43.36
$\max(\hat{f}_{res})$	46.07	44.28

Table 4.3 *Statistical properties of estimated model parameters and resonance frequency for 1000 Monte Carlo simulations.*

resonance frequency is biased for both model M1 and M2. The performance is after all much better than without a pre-filter. ❑

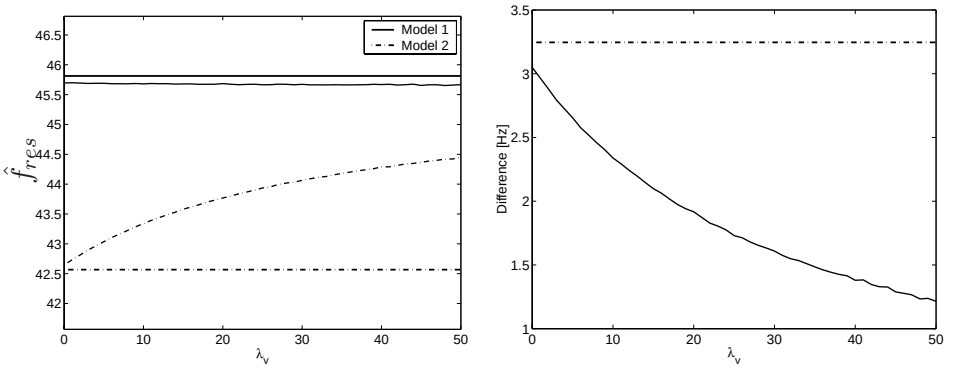
Pre-processing data with a bandpass filter introduces two phenomena.

1. The estimated resonance frequency is pushed to the middle of the passband. This phenomenon gets more pronounced when the SNR becomes worse.

2. The poles of $\hat{A}(q)$ are pushed closer to the unit circle, i.e., the estimated spectrum is more narrow than the true is.

Both these phenomena are due to the large frequency range eliminated by the pre-filter.

Another important property of the estimate is the ability to detect changes during different SNR situations. Here, this is equal to the ability to distinguish the estimated resonance frequency for model M1 and M2 when λ_v is varied. In Figure 4.6(a) the estimated resonance frequency for 1000 Monte Carlo simulations for both model M1 and M2 is shown and in Figure 4.6(b) the difference between the estimates is shown. As can be seen in the figures it becomes more difficult to detect changes in the system when λ_v increases, though it is still possible to separate



(a) (b)

Figure 4.6 *In (a) the mean value over 1000 simulations of estimated resonance frequency with pre-processed data, $\hat{f}^i_{res}$, as a function of λ_v for both M1 and M2 are shown. Also the true resonance frequency is plotted. In (b) the difference $\hat{f}^1_{res} - \hat{f}^2_{res}$ (solid) along with the true difference (dash-dotted) is plotted.*

the two models. The figure also confirms that the estimated resonance frequency moves towards the middle of the passband, 45 Hz, when λ_v increases.

4.3.5 Down sampling

In both approaches studied in Sections 4.3.3 and 4.3.4 the estimation of the model parameters has to consider the frequency content outside the distribution of the resonance frequency. What we ideally want is to only consider the frequency range of the resonance frequency. A way of achieving this is to reduce the sampling rate using down sampling, (see Mitra, 1998; Proakis and Manolakis, 1996). Instead of only decimating data we will also use the alias effect.

Let us assume that the sample rate is 240 Hz for the signal y and we want to consider the frequency interval $[30,\ 60]$ Hz. The algorithm below is used to perform the necessary operations.

ALGORITHM 4.2 Down sampling

1. $y_{ds1} = LP(0.5, y)$, lowpass filtering with normalized cut-off frequency 0.5
2. Decimate y_{ds1} by a factor 2

3. $y_{ds2} = HP(0.5, y_{ds1})$, highpass filtering with normalized cut-off frequency 0.5
4. Decimate y_{ds2} by a factor 2
5. $y_{ds}(k) = (-1)^k y_{ds2}(k)$, reverse spectra

❑

In 1) and 2) the frequency content above 60 Hz is removed and the sample rate is decimated to $f_s = 120$ Hz. In 3) and 4) the frequency content below 30 Hz is removed and the interval $[30,\ 60]$ Hz is aliased backwards to $[0 - 30]$ Hz when the sample rate is decimated to $f_s = 60$ Hz. This means that the true resonance frequency for this special case is folded according to

$$f_{res}^{ds2} = 60 - f_{res}. \tag{4.25}$$

To avoid the backward aliasing the periodic behavior of DFT can be utilized. Because of the periodicy, reversion of spectra can be implemented by frequency shifting the DFT with half the sample frequency

$$Y_{ds}(f) = Y_{ds2}(f - \frac{f_s}{2}) \tag{4.26}$$

where $Y_{ds2}(f)$ is the DFT of y_{ds2} and f_s is the sample frequency, in the example above $f_s = 60$ Hz. In 5) the frequency shift is implemented in the time domain as a simple sign conversion for every second sample

$$y_{ds}(n) = (-1)^n y_{ds2}(n). \tag{4.27}$$

The mapping between the true resonance frequency and the estimated is now

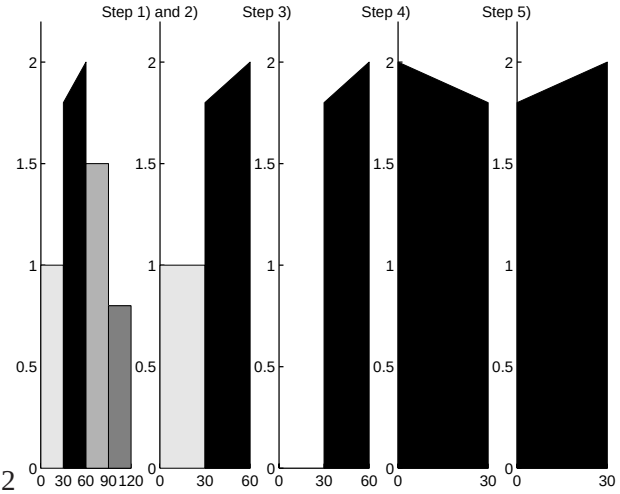
$$f_{res}^{ds} = f_{res} - 30. \tag{4.28}$$

Algorithm 4.2 interpreted in the frequency domain is illustrated in Figure 4.7. The frequencies taken into consideration in the loss function

$$\begin{aligned} \bar{V}(\theta) &= \int_0^{30} \left| Y_{ds}(e^{i2\pi T_s f}) A(e^{i2\pi T_s f}, \theta) \right|^2 \, df \\ \hat{\theta} &= \arg\min(\bar{V}(\theta)) \end{aligned} \tag{4.29}$$

is now $[0 - 30]$ Hz. To solve the Wiener-Hopf equations (4.19) – (4.21) the approximated covariance function is based on the down sampled signal y_{ds}

$$\hat{R}_y(i) = \frac{1}{N} \sum_{k=1}^{N} y_{ds}(k) y_{ds}(k - i). \tag{4.30}$$



22

Figure 4.7 *Illustration of Algorithm 4.2 in the frequency domain.*

Example 4.5 Down sampling

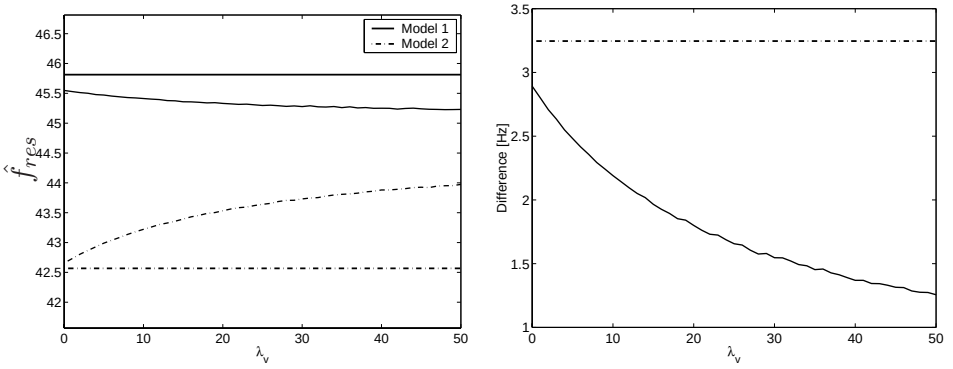
1000 Monte Carlo simulations for both model M1 and M2 are performed with $\lambda_v = 20$. Algorithm 4.2 is used to down sample data and (4.29) is used to estimate the model parameters. Both the lowpass- and highpass filters used in the down sampling algorithm are second order Butterworth filters. To be analogous with the previous methods the estimated resonance frequency is transformed to the original position using (4.28). The estimation result is presented in Table 4.5. In the table

	Model 1	**Model 2**
f_{res}	45.81	42.57
$\mathrm{Mean}(\hat{f}_{res})$	45.33	43.52
$\mathrm{RMSE}(\hat{f}_{res})$	0.15	0.14
$\min(\hat{f}_{res})$	44.87	43.03
$\max(\hat{f}_{res})$	45.75	43.96

Table 4.4 *Statistical properties of estimated model parameters and resonance frequency when down sampled for 1000 Monte Carlo simulations with the noise floor $\lambda_v = 20$.*

it can be seen that the estimated resonance frequency is biased for both model M1 and M2. ❏

In Figure 4.8(a) the estimated resonance frequency for 1000 Monte Carlo simulations for both model 1 and 2 is shown and in Figure 4.8(b) the difference between the estimates is shown. The performance of estimating the resonance fre-



(a) (b)

Figure 4.8 *In (a) the mean value over 1000 simulations of estimated resonanance frequency with downsampled data, $\hat{f}_{res}$, as a function of λ_v for both model M1 and M2 are shown. Also the true resonance frequency is plotted. In (b) the difference $\hat{f}^1_{res} - \hat{f}^2_{res}$ (solid) along with the true difference (dash-dotted) is plotted.*

quency using down sampling is similar to pre-filtering with a bandpass filter. One problem of down sampling is that it truncates the true AR(2) spectra outside the interval $[30 - 60]$ Hz. The advantages are that the model fit is improved and the sample rate is decreased by a factor 4.

4.4 Summary

Five different methods to estimate a single resonance frequency were presented. Even though the non-parametric method seems to have the best performance it suffers from the large memory requirement. N data needs to be recorded to perform the FFT. In an on-line application this also introduces a time delay.

The advantage with using parametric method is apparent when a proper model structure is used. Unfortunately, for the application discussed in Chapter 5 it is impossible to know the proper model structure. Therefore as simple model structure

as possible is desirable. Two methods to isolate the frequency content around the resonance frequency were evaluated. Using a bandpass pre-filter pushes the poles closer to the unit circle and the estimated resonance frequency is pushed towards the middle of the passband. Using down sampling truncates the original spectra by applying filtering and decimation. In the example the sample rate was decreased by a factor 8. The advantage with down sampling is that the algorithms following the down sampling step runs at a slower rate.

5

Application: Tire Pressure Monitoring System

During the last decade there has been an increased interest in intelligent vehicles. More and more intelligence is put into vehicles. In particular, intelligence is used for safety improvements. The first two major intelligent safety systems were the airbag and the Anti-lock Braking System (ABS). Both these systems have been a huge success and is today more or less standard components in new vehicles. Some examples of intelligent safety systems introduced in the last decade are Electronic Stability Program (ESP), Electronic Brake Assistant (EBA), Traction Control Systems (TCS) and Adaptive Head-Light (AHL). Examples of future systems are Collision Avoidance Systems (Jansson, 2002) and Lane Keeping Aid Systems (Kawazoe et al., 2001). Another safety system which has gained popularity in the last couple of years is the *Tire Pressure Monitoring System* (TPMS) (Fennel et al., 2002; Gustafsson et al., 2001a; Hakanen, 2002). It continuously monitors the inflation pressure of the vehicles tires.

The importance of correct inflation pressure in the tires is well known. The handling properties is to a large extent determined by the tire-road contact patch. In Gillespie (1992) the following quote is given ”the critical control forces that determine how a vehicle turns, brakes and accelerates are developed in four contact patches no bigger than a man’s hand”.

When the tire is under-inflated the handling properties deteriorate since the ability to generate forces is reduced, (see Marshek et al., 2002). It can also lead to increased temperature in the tire which may cause tire failure, tire separation or

blow-outs, all of which can cause the driver to lose control of the vehicle.

A recent survey conducted by National Center for Statisics and Analysis (Department of Transportation, 2001) of 6 442 passenger vehicles in the US, found that 36 % of all vehicles were operating with at least one tire tire under-inflated with 20 % below the vehicle manufacturer's recommended cold inflation pressure. About 26 % of the vehicles had at least one tire under-inflated with 25 %.

These statistics indicates a large potential for a TPMS supporting the driver to maintain correct inflation pressure. Besides the safety issues, economical and environmental issues motivate TPMS. Studies have shown that the fuel consumption increases 1.5 % and lifetime decreases 15 % for every 0.2 bar under-inflation, (et. al., 2001).

Motivated by the issues above TPMS have gained popularity in the last couple of years and is now optionally available for some car models (Grygier et al., 2001). The business potential for TPMS is estimated to $2–3 billions annual from the year 2005 according to Lundström and Andersson (2002).

5.1 Tire Pressure Monitoring System

The idea to continuously monitor the tire inflation pressure during driving goes back to the early 70's. Richardsson (1971) used a pair of metallic contact points spaced from each other and mounted inside the tire on the tire wall. When the tire wall is compressed from lack of sufficient air pressure within the tire, the metallic points will be brought into contact with one another and under-inflation is detected. In the last decade the development of TPMS has increased rapidly which can be seen by a patent search on "Tire Pressure". More than 300 European patents (e.g. Nakajima and Isshiki, 1996; Takeyasu et al., 1997; Toshiharu et al., 1999; Yoshira et al., 1996) are found and at least the same number of US patents (e.g. Scott and Alan, 1998; Yuuichi et al., 1996).

One reason for the increasing number of TPMSs is the upcoming regulatory in the US. It states that all vehicles produced from the beginning of 2004 must have a TPMS, (Department of Transportation, 2002). This is due to the growing number of fatal accidents in the US caused by tire failures.

There are mainly two ways of monitoring the tire inflation pressure;

1. One way is to mount a pressure sensor at the rim inside the tire and via a communication-link (usually a radio-link) transfer the sensor value to a central receiver. Systems based on this technique are called *direct TPMS*.

2. The other alternative, called *indirect TPMS*, uses existing sensors (typically wheel speed sensors) and software algorithms, (Gustafsson et al., 2000b;

Persson et al., 2002). The pressure is not directly measured, but instead it is estimated in the software algorithms.

The drawback with a direct TPMS is the requirement of extra hardware in form of sensors, batteries and a central unit. Therfore direct TPMSs is relatively expensive. An indirect TPMS, on the other hand, is only a software program and is very cost-effective. A disadvantage with an indirect TPMS is that it can not estimate the absolute inflation pressure. Instead the deviation from a nominal/calibrated value is estimated. Both indirect and direct TPMS are available on the market (Bishop, 2002), both as options for new vehicles and on the after market.

There are two principles to estimate the tire inflation pressure in an indirect TPMS, vibration analysis and wheel radius analysis. Wheel radius analysis uses the fact that the effective rolling radius decreases when the tire inflation pressure decreases. In Gustafsson et al. (2000a); Nakajima and Isshiki (1996); Persson et al. (2001); Sager and John (1998) wheel radius analysis is further discussed. Vibration analysis uses the fact that the rubber in the tire acts like a spring when excited from road roughness. The spring in the tire generates an overloaded oscillation in the wheel speed with a certain resonance frequency. The resonance frequency is strongly correlated with the tire inflation pressure. The idea is to monitor and detect changes in the resonance frequency.

In this chapter we will discuss the different parts incorporated in a TPMS using vibration analysis. The main source of information to an indirect system is the wheel speed sensor discussed in Section 2.2.

5.2 The Tire

When designing an indirect TPMS it is important to understand the physics of the tire or at least the basic behavior. In most cases the tire is treated/modeled as a part of the vehicle that generates forces by which the vehicle may be controlled in braking, accelerating and turning, (Gillespie, 1992). For the purpose of TPMS, the forces generated by the tire is not the primary issue. Instead, it is more important to know how road roughness and other disturbances propagate through the tire and affect rotational velocity of the wheel axle.

A common and simple model for the tire with respect to vibrations is presented in for example Mancosu et al. (2001) or Wong (1993). The tire can be modeled as a spring-damper system in both torsional and vertical direction, see Figure 5.1. The vertical spring-damper system is the source of two phenomena.

First, the steady state effective rolling radius is affected by the spring constant

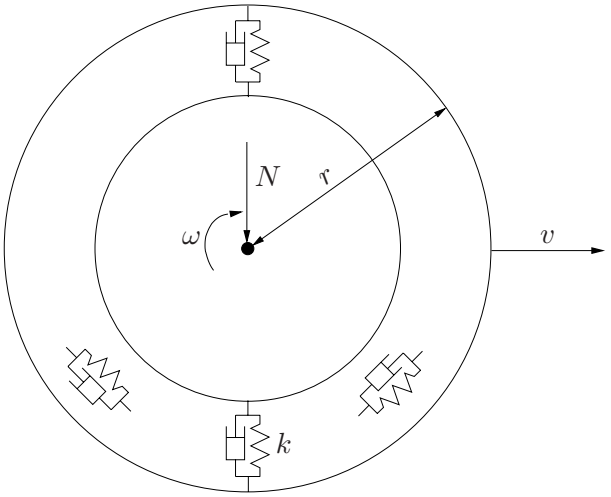


Figure 5.1 *Tire modeled as a spring-damper system in both vertical and torsional direction.*

k and the normal force N

$$\omega = \frac{v}{r(N,k)} = \frac{v}{r_{nom} - N/\Delta k} \tag{5.1}$$

where Δk denotes the deviation from the nominal spring constant value and r_{nom} the nominal tire radii. When the tire inflation pressure decreases the spring constant is decreased with Δk, which means that the effective rolling radius is smaller and forces the rotational velocity of the wheel to increase. This phenomena is used by TPMS's based on wheel radius analysis (see Nakajima, 1998; Randazzo et al., 1998; Sager and John, 1998). The relationship between inflation pressure and rolling radius is sometime called *tire pressure sensitivity*. Most of these systems use a static non-linear relation to detect under inflation. One example is

$$\varepsilon = \frac{\omega_1}{\omega_2} - \frac{\omega_3}{\omega_4} \tag{5.2}$$

where the wheel speeds, ω_i, are enumerated as front left ($i = 1$), front right ($i = 2$), rear left ($i = 3$), and rear right ($i = 4$). A pressure loss is detected when the test statistic, ε, is significantly non-zero. As an inherent disadvantage, this static consistency test cannot detect equal pressure losses on the same axle or the same side of the vehicle. An equal pressure loss in all tires will also be undetected.

Secondly, the tire oscillates in the vertical direction excited by road roughness causing the effective rolling radius to fluctuate with a specific resonance frequency.

Indirectly this results in a fluctuation of the wheel speed at the same frequency. The most significant mode of the vertical vibration is about $10 - 20$ Hz. In case of a pressure loss the spring constant decreases yielding a lower resonance frequency

$$f_{res} = f_{nom} - g(\Delta k) \tag{5.3}$$

where f_{nom} is the nominal resonance frequency and g is a function determining the relationship between Δk and f_{res}. By monitoring the resonance frequency under-inflation can be detected, (Umeno et al., 2001; Vos, 2002).

The torsional spring-damper system is also excited by road roughness and other forces acting on the tire. In contrast to the vertical vibration the torsional vibration affects the rotational velocity directly. As for the vertical vibration, f_{res} drops when the tire loses pressure. The most significant mode of the torsional vibration is about $40 - 50\ Hz$. Due to the fact that the torsional vibration affects the wheel speed directly the resonance frequency is more pronounced than for the vertical vibration. In the rest of this chapter we will describe an indirect TPMS based on the torsional vibration. With some minor modifications the method is applicable to the vertical vibration as well.

5.2.1 Continuous Time Tire Model

In Figure 5.1 the tire was modeled as a spring-damper system in both torsional and vertical direction. This simple model can by no means explain the entire information in the wheel speed signal. The purpose is merely to illustrate the basic properties of the vibrations. In Figure 5.2 a spring-damper system is visualized. A vertical spring-damper system is used for convenience, but the figure and the results are valid for both the vertical and torsional vibrations with respect to the dynamics. y is the position (vertical) or angle (torsional) of the rim, with origin at the static equilibrium. x can be seen as the elevation of the surface profile (or angle). k and b are the spring constant and damping coefficient, respectively. m is the mass of the rim (moment of inertia). By applying Newton's second law to the rim mass, the equation of motion can be obtained (Meriam and Kraige, 1993, pp. 597-600). For vibrations excited by road irregularities, the equations of motion for the rim are as follows

$$\begin{aligned} \ddot{y}(t) + \frac{b}{m}\dot{y}(t) + \frac{k}{m}y(t) &= F(t) \\ \frac{b}{m}\dot{x}(t) + \frac{k}{m}x(t) &= F(t) \end{aligned} \tag{5.4}$$

where $F(t)$ is the excitation acting on the wheel from road surface irregularities. In the equation it is also assumed that the suspension is stiff and the mass of the

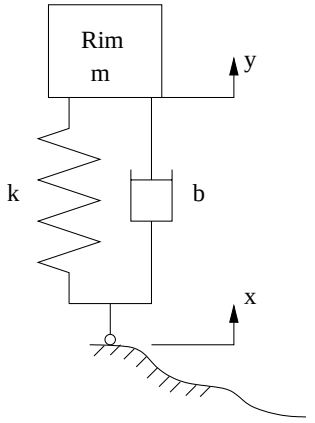


Figure 5.2 *Spring-damper model.*

vehicle only affects the static equilibrium. Equation (5.4) can be rewritten with respect to standard notation for oscillatory motions as

$$\ddot{y}(t) + 2\gamma\dot{y}(t) + f_0^2 y(t) = F(t) \tag{5.5}$$

with

$$\begin{aligned} \gamma &= \frac{b}{2m} \\ f_0 &= \sqrt{\frac{k}{m}}. \end{aligned} \tag{5.6}$$

If equation (5.4) is Laplace transformed it can be written as

$$Y(s) = H(s)F(s) \tag{5.7}$$

where $H(s)$ is the transfer function of the spring damper system defined as

$$H(s) = \frac{1}{s^2 + 2\gamma s + f_0^2}. \tag{5.8}$$

The absolute value and argument of the poles in (5.8) are

$$|s| = \omega_0 = \sqrt{\frac{k}{m}} \tag{5.9}$$

$$\arg(s) = \frac{\gamma}{f_0} = \frac{b}{2\sqrt{mk}}. \tag{5.10}$$

In Wong (1993) it is stated that the spring-constant decreases and damping coefficient increases when the tire inflation pressure decreases. Thus the poles are moved

closer to the origin and the angle to the real axis is decreased, i.e., the resonance frequency will be damped.

From the transfer function (5.8) the spectrum is obtained by the substitution $s = i2\pi f$

$$|H(i2\pi f)|^2 = \frac{1}{\left((2\pi f)^2 - f_0^2\right)^2 + 4\gamma^2(2\pi f)^2}. \tag{5.11}$$

From (5.11) the maximizing argument, i.e., the location of the resonance peak is defined as

$$f_{res} = \frac{1}{2\pi}\sqrt{f_0^2 - 2\gamma^2} = \frac{1}{2\pi}\sqrt{\frac{k}{m} - \frac{b^2}{2m^2}}. \tag{5.12}$$

A problem is that the spectrum, $\Phi_F(w)$ generated by road irregularities is unknown, which implies that $\Phi_y(w) = |H(i\omega)|^2\Phi_F(w)$ also is unknown. Here, the excitation from road irregularities is approximated with white noise, $\Phi_F(w) = \sigma^2$. $|H(i\omega)|^2$ is shown in Figure 5.3 with respect to γ and ω. The figure shows that

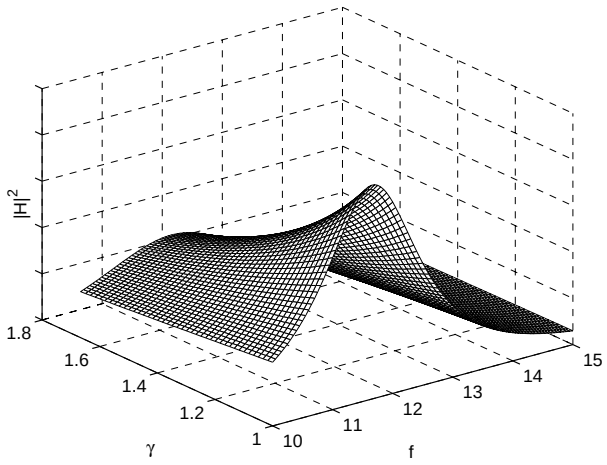


Figure 5.3 *$|H(i\omega)|$ with k, m, b from (Wong, 1993, pp. 64-67).*

the resonance peak gets more and more diffuse for larger values of γ, i.e., larger damping.

5.2.2 Vertical Vibrations

In the vertical vibration the rotational velocity of the wheel is not directly affected by road irregularities. Instead fluctuations in the effective rolling radius cause the

wheel speed to fluctuate. For the vertical vibration, $y(t)$, in Equations (5.4)– (5.5) can be replaced with $\delta_r(t)$. This means that the effective rolling radius, r_{eff} is going to fluctuate around its static equilibrium, r_0. From Figure 5.1 this can be seen as

$$\omega(t) = \frac{v(t)}{r_{eff}} = \frac{v(t)}{r_0 + \delta_r(t)} \tag{5.13}$$

By applying a first order approximation (5.13) can be written as

$$\omega(t) \approx \frac{v(t)}{r_0}\left(1 - \frac{\delta_r(t)}{r_0}\right) = \frac{v(t)}{r_0} - \frac{v(t)}{r_0^2}\delta_r(t). \tag{5.14}$$

In (5.14) it can be seen that the vertical vibration $\delta_r(t)$ is amplitude modulated with the velocity of the vehicle $v(t)$.

5.2.3 Torsional Vibrations

The dynamic Equation (5.5) is applicable for the torsional vibration as well, if the mass of the rim is substituted with the moment of inertia and the spring-damper system is considered as a torsional spring-damper system. The exciting force needs also to be considered as horizontal at the contact patch between the road and the tire instead of horizontal. The main difference is that the rotational velocity is affected directly by road irregularities, i.e., $y(t)$ in equations (5.4)–(5.5) can be replaced with $s(t)$. The angular velocity of the wheel and its transform can be written as

$$\omega(t) = \omega^0(t) + s(t) \tag{5.15}$$

where $\omega^0(t) = v(t)/r_0$. Combining (5.14) and (5.15) the angular velocity affected by both vertical and torsional vibrations can be written as

$$\omega(t) = \omega^0(t) - \frac{\omega^0(t)}{r_0}\delta_r(t) + s(t) \tag{5.16}$$

Usually, a good "rule of thumb" is to consider $\omega^0(t)$ to be band limited to 10 Hz. Most of the energy in the vertical vibration is around 10-20 Hz. In (5.16) the vertical vibration $\delta_r(t)$ is amplitude modulated with $\omega^0(t)$. In the frequency domain this is equivalent to a convolution. This means that most of the energy of the amplitude modulated vertical vibration is below 30 Hz. The resonance frequency of the torsional vibration is around $40 - 50$ Hz and there is little interference with both the vehicle and vertical vibration.

Dynamics	Frequency Range [Hz]
Vehicle motion	0-10
Suspension	1-2
Engine	10-100
Vertical tire vibration (amplitude modulated)	< 30
Torsional tire vibration	40-50

Table 5.1 *Frequency content of the most significant dynamics parts of the vehicle.*

In Table 5.1 the frequency range of the most significant dynamic parts of the vehicle are listed. The vibration from the engine depends on the present motor speed. Even though the engine dynamics seems to interfere with torsional tire vibration, empirical tests have shown that the influence from engine dynamics is neglectable in the TPMS application.

In the sequel the contribution from the vertical vibration is neglected and the transfer function for the angular velocity with a torsional vibration is

$$\begin{aligned} \Omega(s) &= \Omega_0(s) + S(s) \\ S(s) = H(s)F(s) &= \frac{1}{s^2 + 2\gamma\omega_0 s + \omega_0^2} F(s) \end{aligned} \tag{5.17}$$

where $\Omega^0(s)$ is given by Laplace transform of $\omega^0(t)$ and reflects the dynamics of the vehicle, i.e., maximum acceleration and deceleration.

5.2.4 Discrete Time Torsional Vibration Model

The continuous time tire model $H(s)$ is observed in discrete time measured with the wheel speed sensor and the discrete time model for (5.8) is needed. The discretization of the continuous time model adds a zero and time delay in the transfer function

$$H(q) = \frac{b_0 q^{-1} + b_1 q^{-2}}{1 + a_1 q^{-1} + a_2 q^{-2}} \tag{5.18}$$

where q is the shift operator $q^{-1}y(t) = y(t-1)$ and the parameters a, b depend on the sample period and the parameters in the continuous time model. The wheel speed signal is event based sampled which implies that the sample period varies from sample to sample. It also implies that the parameters a, b in (5.18) varies from sample to sample even though the parameters in the continuous time model (5.8) are constant.

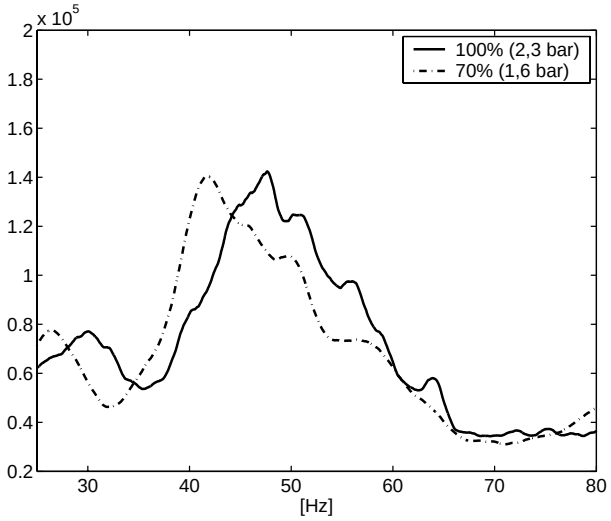


Figure 5.4 *Smoothed speriodogram of the wheel speed for two different tire inflation pressures.*

In Figure 5.4 the smoothed periodogram of the wheel speed for two different tire inflation pressures are plotted. As can be seen in the figure the model in Equation (5.18) will not be sufficient to describe the information in the wheel speed. One solution is to use a higher order model. This implies that the computational complexity is increased. Here, we will use as simple model as possible to minimize the computational complexity. Therefore, a discrete second order AR-model

$$\tilde{H}(q) = \frac{1}{1 + a_1 q^{-1} + a_2 q^{-2}} \tag{5.19}$$

is used for estimating the resonance frequency. The estimation properties for this specific model structure is discussed in Chapter 4.

5.3 An Indirect TPMS Based on Torsional Vibration

The most important task of the TPMS is to identify the resonance frequency of the torsional vibration. It needs to be both robust to different driving scenarios and sensitive to inflation pressure changes. The only information source available is the wheel speed sensor, discussed in chapter 2.2. In Figure 5.5 an overview of the system architecture is shown. This is a simplified picture of the reality, but the most important parts are included. Below a brief description of the different parts are given.

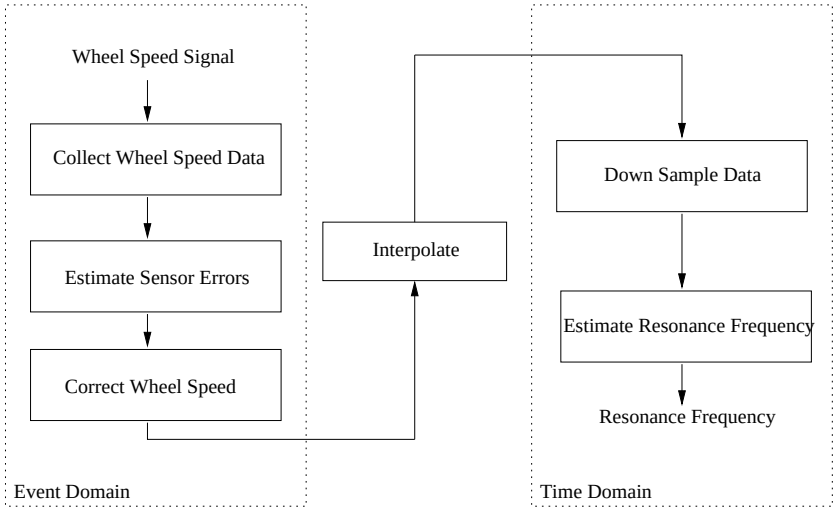


Figure 5.5 *Schematic overview of system architecture.*

Collect Wheel Speed Data: Communicates with the sensor unit (ABS-system) and receives time stamps every time a tooth passes the sensor, as described in Chapter 2.2.

Estimate Sensor Errors: In Chapter 2.3 the influence from the sensor errors in the wheel speed is described. To attenuate this influence the sensor errors are estimated in order to improve the signal quality and make the rest of the signal processing chain feasible.

Correct Wheel Speed: Computes an improved wheel speed with respect to the estimated sensor errors, see Chapter 2.3.

Interpolate: Converts the data from the event domain to the time domain. Described in chapter 3.

Down Sample Data: Removes parts of the signal with no inflation pressure information. Here, the down sampling procedure described in Chapter 4 is used.

Estimate Resonance Frequency: Estimates the resonance frequency by means of estimating parameters in a model structure adapted to the torsional vibration. Described in Chapter 4.

5.3.1 Test Units

The data used to evaluate the performance of the indirect TPMS is collected from real test-drives. The test vehicles are a Volvo V70 (2001) and a Volvo V40 (1997) referred to as testvehicle 1 and 2, respectively. The data acquisition and test vehicles are supplied by NIRA Dynamics AB. The test vehicles are equipped with a central unit communicating with the ABS-system in the vehicle. The specifications of the ABS-system is displayed in Table 5.2. To the central unit it is possible

Property	**Testvehicle 1**	**Testvehicle 2**
Clock sample period, $T_{s,Clock}$	0.4μs	1.66μs
Number of bits in the counter	16	
Maximum counter value, C_{max}	2^{16} = 65536	
Number of teeth, L	48	43
Min. velocity to avoid overflow in counter	5 rad/s	1.5 rad/s
Sensor type	Inductive	

Table 5.2 *ABS specifications*

to connect a PC for easy storage of measurements and on-line monitoring. It is also possible to run the algorithms on-line in the central unit. The absolute value of the tire inflation pressure is not recorded in the measurements. Instead a manual measurement is made both before and after the test-drive. It is possible to simulate a slow leakage. This is done by connecting an extra valve to the original valve. In the extra valve a small opening has been drilled to slowly deflate the tire.

The sample period of the clock (see Table 5.2) can be modified by the user, but to avoid too much quantization noise and still have a low minimum velocity $T_{s,Clock}$ is a relevant choice. Each time a tooth passes the sensor, the counter value in ABS-unit is received, $\mathbf{C_k} = [c_k, c_{k-1}, ...]$. To get the absolute time, $\mathbf{C_k}$ needs to be scaled with the clock period $T_{s,Clock}$

$$\mathbf{t_k} = [t_k, t_{k-1}, ...] = \mathbf{C_k} T_{s,Clock}. \tag{5.20}$$

The time instants $\mathbf{t_k}$ are the input to the TPMS system. The algorithm used to estimate the resonance frequency is recursive, i.e., every time a new sample (time instant) is received it is processed. This is not equivalent with computing an estimate of the resonance frequency every time a new sample is received. The update rate of the resonance frequency estimate is chosen by the user while the sample rate of the time instants is determined by the velocity of the vehicle.

5.3.2 Proposed Methods

In Chapters 2 – 4 some different methods in each step of the signal processing chain to estimate the resonance frequency from the time instants are described. One method needs to be chosen to each step. The algorithm we suggest is summarized below.

Attenuation of sensor errors: For attenuating sensor errors the auto calibration (AC) is used rather than Fourier series expansion (FS). AC is beneficial concerning both performance and computational complexity. In FS all harmonics originating from the sensor errors need to be attenuated to avoid undesirable aliasing in the interpolation. This results in increased computational complexity. It is also natural to estimate the true sensor errors as in AC rather than estimating the time for an ideal sensor as in FS. AC estimates the sensor errors and uses the estimates to compute a corrected wheel speed

$$\hat{\omega}_k^c = \frac{\alpha + \hat{\delta}_k}{\Delta t_k} = \frac{2\pi/L + \hat{\delta}_k}{\Delta t_k} \tag{5.21}$$

In Figure 5.6 the raw computation of the wheel velocity and its spectrum is shown along with the corrected spectrum using (5.21). From the figure it can be seen that the attenuation of sensor errors works well. The remaining disturbances are not originating from sensor errors. In the uncorrected spectrum the basic frequency of the sensor error harmonics is approximately 6.9 Hz. According to (2.32) the basic frequency is equal to the inverse of the revolution time, $\omega/(2\pi)$, if the wheel velocity is constant. From Figure 5.6 the wheel velocity is approximately $\omega = 43$ rad/s which results in a basic frequency of 6.85 Hz. All the other harmonics are multiples of the basic frequency.

Interpolation: The sample period in the interpolation is chosen to 240 Hz. This is to avoid aliasing of high frequency signal components which are not blocked by the interpolation filter. The Epanechnikov kernel described in Chapter 3 is used for interpolation. Filter based interpolation is too computational complex for use in on-line applications.

Estimation of resonance frequency: Before estimating the resonance frequency, the down sampling routine described in Chapter 4 is applied. Down sampling is applied to isolate the resonance frequency in the estimation and also for reducing the sample rate to save computations. Recursive least squares

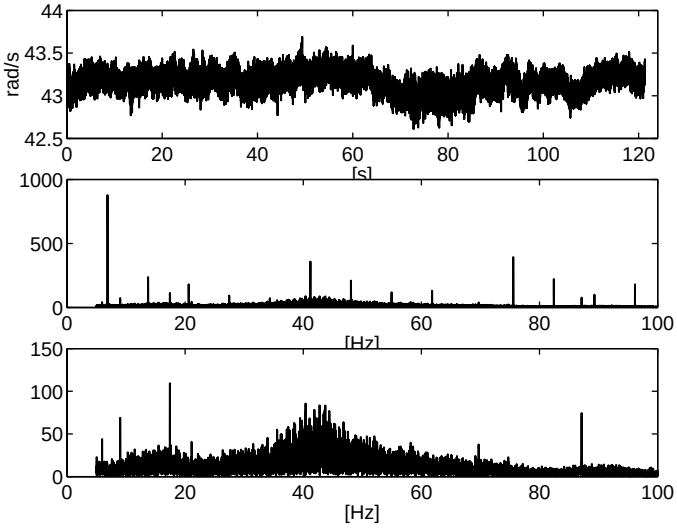


Figure 5.6 *In the top figure the uncorrected wheel speed is plotted. In the lower plots the periodogram of the uncorrected wheel speed (middle) and corrected wheel speed (bottom) are shown.*

is used to estimate the parameters in the second order AR-model

$$\begin{aligned} y(t) =& \frac{1}{A(q)} e(t) \\ A(q) =& 1 + a_1 q^{-1} + a_2 q^{-2}. \end{aligned} \tag{5.22}$$

The estimated parameters are transformed to a resonance frequency by

$$\hat{f}_{res} = \frac{\arccos\left(-\frac{\hat{a}_1}{2\sqrt{\hat{a}_2}}\right)}{2\pi T_s} + 30. \tag{5.23}$$

Adding the factor 30 to the estimated resonance frequency is only to revert the alias effect in the down sampling procedure. In Figure 5.7 the spectrum of the interpolated signal and the estimated resonance frequency are shown. In the figure it can be seen that the transient period is less than 30 seconds. The properties of the estimated resonance frequency for different inflation pressures and when the inflation pressure is slowly decreasing is evaluated in the next section.

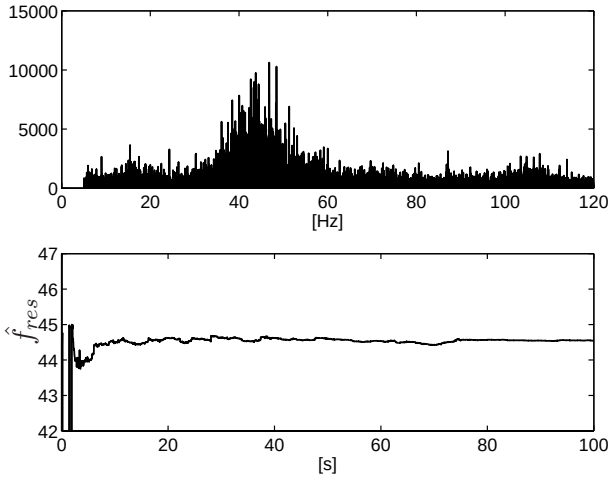


Figure 5.7 *In the upper figure the periodogram of the interpolated wheel speed is plotted and in the lower figure the estimated resonance frequency as a function of time is plotted.*

5.3.3 Test Scenarios

In this section we will look at two different test scenarios. The tests have been performed using the test units and methods described in the previous sections. All tests have been performed at public roads which limits the driving scenario.

First, the tire inflation pressure is constant during the whole test drive. The inflation pressure is varied from test-drive to test-drive. This is for evaluating the possibility to separate different inflation pressure in terms of the resonance frequency.

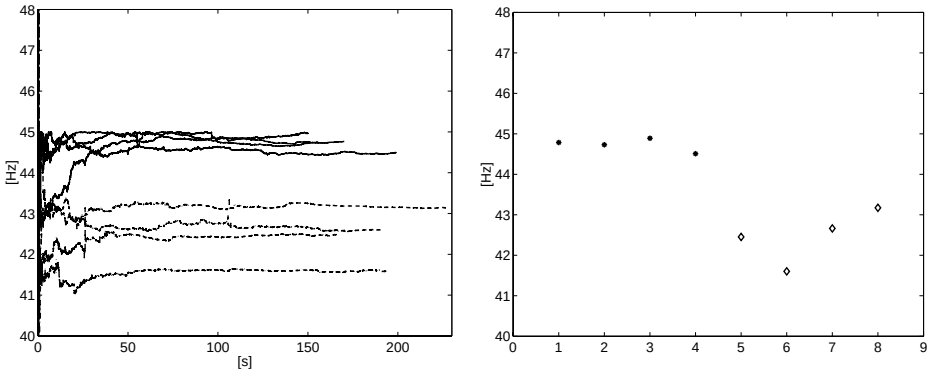
Second, a slow deflation is simulated with the extra valve described in Section 5.3.1. This is for evaluating the possibility to detect real puncture. Also the tracking performance of the system is evaluated.

Another limiting parameter is that the wheel speed signal is not available when the vehicle is driving slower than the minimal velocity. The reason for this is that when the velocity is too low the the algorithms can neither determine the speed of the vehicle nor if it is driving forward or backward. This is of course solvable in a product system, but it is beyond the scope of this thesis.

5.3.4 Constant Tire Pressure

In the first test testvehicle 1 is used. It is equipped with Conti Premium Contact (summer tires) with profile R15 195/65. Eight different test-drives were per-

formed. The speed varies from 50 – 110 km/h and two different tire inflation pressures (100 % and 70 % of nominal inflation pressure) are tested. In Figure 5.8(a) the estimated resonance frequency as a function of time is shown and in Figure 5.8(b) the mean value of the estimated resonance in each test-drive is plotted. From Figure 5.8 it can be seen that the system is able to separate the



(a) (b)

Figure 5.8 *In (a) the estimated resonance frequency as function of time is plotted. The solid lines represent test-drives with nominal pressure (100 %) and the dashed lines represent test-drives with 70 % of nominal inflation pressure. In (b) the mean value of the estimated resonance frequency in each test-drive is plotted. Circles represent the mean value over time of the solid lines in (a) while diamonds represent the mean value of the dashed.*

test-drives with nominal pressure from the test-drives with under-inflation.

To verify that the algorithms work for more than one test vehicle similar test-drives are accomplished with testvehicle 2. It is equipped with Pirelli P6000 (summer tires) with profile R15 205/55. These tires have a lower profile (smaller side wall) than the ones used for testvehicle 1. The speed varies between 60 – 90 km/h and each test-drive is approximately 30 – 40 seconds. In Figure 5.9 the mean value of the resonance frequency from 20 test-drives are plotted. Also in this case it is possible possible to separate the test-drives with nominal pressure from the ones with under-inflation.

By comparing Figure 5.8(b) with Figure 5.9 it can be seen that the nominal pressure does not correspond to the same estimated resonance frequency. For testvehicle 1 it is approximately 45 Hz while it is 47 Hz for testvehicle 2. To

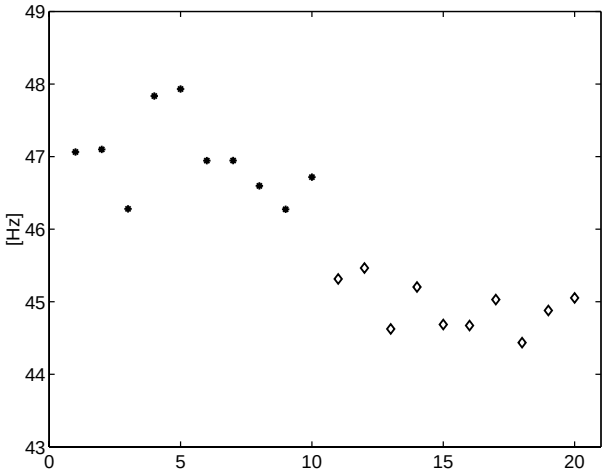


Figure 5.9 *Mean value of estimated resonance frequency in each test-drive for two different inflation pressures. Circles represent nominal pressure (100%) and diamonds under-inflation (70% of nominal pressure).*

detect under-inflation an adaptive threshold is needed. The most common method is to utilize a calibration function trigged by the driver. When the driver presses the calibration button the system "learns" the nominal estimated resonance frequency. Developing a calibration function without driver input is one of the most challenging tasks for indirect TPMS.

5.3.5 Simulation of a slow deflation

For simulation of a slow deflation the extra valve described in Section 5.3.1 is used. Testvehicle 2 is used in the test and is equipped with the tires described in the previous test. Four different test-drives are accomplished. All test-drives start with nominal inflation pressure. After the test-drive the inflation pressure is measured to approximately 60 % of the nominal pressure. The vehicle velocity is between 70 – 80 km/h in all test-drives. In Figure 5.10 the estimated resonance frequency from the test-drives are plotted. As seen in the figure the estimated resonance frequency decreases when the tire pressure decreases. The tracking ability is a trade-off between noise attenuation and tracking. A decent trade-off can only be determined by empirical tests. In most cases the false alarm should be minimized, i.e., good noise attenuation is preferred to tracking ability. This results in a slightly longer detection time.

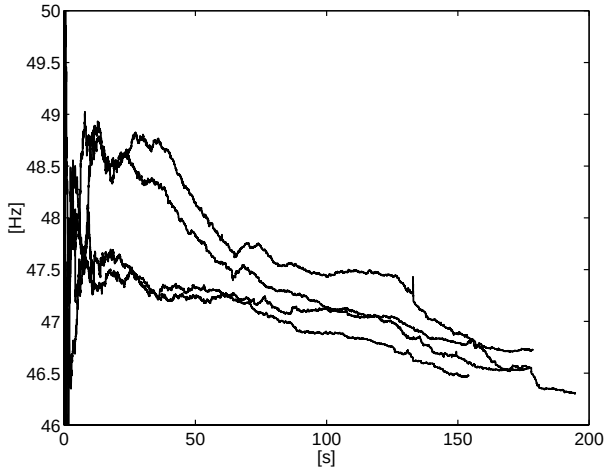


Figure 5.10 *Estimated resonance frequency as a function of time when a deflation is simulated. At the end to the test-drives the inflation pressure is 60 % of the nominal pressure.*

5.3.6 Summary

In the chapter a continuous time tire vibration model is described. It indicates that in an ideal case the tire vibration can be modeled as a second order AR-model. This is used for an indirect *Tire Pressure Monitoring system (TPMS)* to monitor the resonance frequency which is correlated with the tire inflation pressure. The methods to estimate the resonance frequency from wheel speed data, described in Chapters 2 – 4, were applied to real data from test-drives with two test vehicles. The results demonstrated that it is possible to detect under-inflation by monitoring the resonance frequency.

To increase the performance of the indirect TPMS, vibration- and wheel radius analysis should be combined as in (Persson et al., 2002).

6

Conclusions

In this chapter we will summarize the most important topics of the thesis. It will be concluded with some topics for future work.

6.1 Summary

In the thesis event based sampling with application to spectral estimation has been studied. Special attention was paid to a *Tire Pressure Monitoring System (TPMS)*. Parts of the TPMS such as event based sampling and attenuation of senor errors, interpolation and estimation of a resonance frequency have been treated.

The sensor errors were treated as slowly time varying and periodic. By estimating and eliminating the sensor errors the quality of the sensor signal was significantly increased. The estimation of sensor errors is a key part of the the TPMS. All systems relying on wheel speed sensor data such as ABS, ESP etc could benefit from using the improved wheel speed signal.

A linear in data transformation (non-parametric regression) was used to convert event domain data to time domain data. The most straight forward alternative is to use linear interpolation. The other two alternatives discussed, were kernel and filter based interpolation. In kernel based interpolation a weighting function is constructed in the time domain, here an Epanechnikov kernel was used. In filter based interpolation the weighting function is found by discretizing the impulse response of a continuous time filter.

In the last part, estimation of a resonance frequency, the aim was to apply some commonly used estimation methods to a simulation model similar to the tire vibration model and evaluate the estimation properties in terms of accuracy in estimation of the resonance frequency.

As an application of all the above mentioned parts a Tire Pressure Monitoring System was studied. First a continuous time tire vibration model was described, which indicates that a second order Auto Regressive model could be used for estimating the resonance frequency of the torsional vibration of the tire. The proposed methods were evaluated on real test-drives with promising results. Estimation of resonance frequency can be used for continuous monitoring of the tire inflation pressure.

6.2 Future Work

One interesting idea is to develop a TPMS entirely in the event-domain, i.e., estimate the resonance directly from the corrected wheel speed in the event domain. To be able to apply this to TPMS, filtering to isolate the resonance frequency also needs to be done in the event-domain. This can perhaps be utilized by the filter based interpolation discussed in the thesis.

In Chapter 2 the properties of estimating sensor errors were analyzed when a sinusoid were used as disturbance. The analysis should be extended with a larger class of disturbances than just a single sinusoid.

It would also be interesting to study non-uniform sampling in a more general perspective. Especially the “sampling theorem” and spectral analysis for non-uniformly sampled signals.

The filter based interpolation described in Chapter 3 is very interesting. Unfortunately it is too computational complex and the properties of the filter is not completely evaluated/explored. More work needs to be done, but the idea has potential.

One problem that sometimes occurs in the TPMS is when the tire is imbalanced. In a signal processing perspective this is similar to sensor errors. Imbalance, though, is a function of the wheel speed while the sensor errors are almost constant. Wheel imbalance can be interpreted as an additative sinusoid with unknown amplitude and phase to the sensor errors. Perhaps imbalance can be eliminated using Fourier series described in Chapter 2, imbalnce only affects the first coefficients (a_1 and b_1).

Even though the attenuation of sensor errors in Chapter 2 were described as general as possible it would be interesting to verify that the ideas is applicable for more applications than wheel speed sensors. One example is cam-shaft measure-

ments in engines.

In my opinion the bottleneck of the TPMS is the conversion of data from the event-domain to the time-domain. Sometimes high frequency signal components are folded, which degrade the ability to estimate the resonance frequency. This problem needs to be more thoroughly investigated.

Bibliography

K-J Åström and B. Bernhardsson. Comparison of periodic and event based sampling for first-order stochastic systems. In *Preprints of the 14th IFAC World Congress*, 1999.

R. Bishop. Under-inflated ideas. *European Automotive Design*, (June), 2002. Findlay Publications Ltd.

BOSCH. *Automotive Handbook*. Robert Bosch GmbH, 4th edition, 1996.

Department of Transportation. Notice of proposed rulemaking. Docket No: NHTSA-2000-8572-28, 2001. National Highway and Traffic Safety Administration.

Department of Transportation. Tire pressure monitoring systems; controls and displays. Federal Register: 49 CFR Parts 571 and 590, 2002. National Highway and Traffic Safety Administration.

V.A. Epanechnikov. Non-parametric estimation of a multivariate probability density. *Theory of Probability and Its Applications*, 14:153–158, 1969.

W. Hopkins et. al. Transportation recall enhancement, accountability and documentation. Docket No: NHTSA-2000-8572-26, 2001. National Highway and Traffic Safety Administration.

H. Fennel, M. Greisser, and P. Sager. Reifenluftdrucküberwachnung. *System Partners*, May 2002.

E. Galvan, A. Torralba, and L.G. Franquelo. A simple digital tachometer with high precision in a wide speed range. *Industrial Electronincs, Control and Instrumentation*, 2:920–923, Sep 1994.

T.D. Gillespie. *Fundamentals of vehicles dynamics*. Society of Automotive Engineering, Inc., 2nd edition, 1992.

P. Grygier, W.R. Garrott, E.N. Mazzae, J.D. MacIsaac Jr., D. Elsasser R.L. Hoover, and T. A. Ranney. An evaluation of existing tire pressure monitoring systems. Docket Management System: 809 297, 2001. National Highway and Traffic Safety Administration, and Transportation Research Center (TRC) Inc.

F. Gustafsson. *Adaptive Filtering and Change Detection*. John Wiley & Sons, 2000.

F. Gustafsson, M. Drevö, U. Forssell, M. Löfgren, N. Persson, and H. Quicklund. Virtual sensors of tire pressure and road friction. In *Proceedings of SAE 2001 in Detroit, USA*, number 2001-01-0796, 2001a.

F. Gustafsson, L. Ljung, and M. Millnert. *Signalbehandling*. Institutionen för Systemteknik, Linköping, Sweden, 2nd edition, 2001b. In Swedish.

F. Gustafsson, N. Persson, S. Ahlqvist, M. Drevö, and U. Forssell. Adaptive filter model for motor vehicle sensor signals. Patent no. WO 0176925, 2000a. NIRA Dynamics AB.

F. Gustafsson, N. Persson, and M. Drevö. Tire pressure estimation. Patent no. WO 0187647, 2000b. NIRA Dynamics AB.

A. Gut. *An intermediate course in probability*. Springer-Verlag, 1995.

J. Hakanen. Tyre monitoring and safety. In *Proceedings of FISITA 2002 in Helsinki, Finland*, number F02i234, 2002.

M.H. Hayes. *Statistical digital signal processing and modelling*. John Wiley & Sons, Inc, 1996.

E. Hendricks, M. Jensen, A. Chevlier, and T. Vesterholm. Conventional event based engine control. In *Electronic Engine Controls 1994*, number SP-1029. Society of Automotive Engineering, 1994.

H.S. Hou and H.C. Andrews. Cubic splines for image interpolation and digital filtering. *IEEE Trans. Comput.*, ASSP-26:508–517, Dec 1978.

J. Jansson. *Tracking and Decision making for Automotive collision avoidance*. Licentiate thesis 965, Linköping, Sweden, 2002.

H. Kawazoe, T. Murakami, O. Sadano, K. Suda, and H. Ono. Development of a lane-keeping assistance system. In *Proceedings of SAE 2001 in Detroit, USA*, number SAE SP-1593, 2001.

S. M. Kay. *Fundamentals fo Statistical Signal Processing, Estimation Theory*, volume 1. Prentice Hall, 1993.

H. Kopetz. Should responsive systems be event-triggered or time-triggered? *Institute of Electronics, Information, and Communications Engineers Transactions on Information and Systems*, E76-D(11):1325–1332, 1993.

L. Ljung. *System Identificaton Theory for the User*. Prentice Hall, 2nd edition, 1999.

H. Lundström and G. Andersson. Ny lag i USA blir bingo för trådlösa sensorer. *Elektroniktidningen*, (Feb), 2002. Ekonomi & Teknik.

F. Mancosu, C. Savi, P. Brivio, G.C. Travaglio, and I. Ramirez. New dynamic tyre model in multi-body environment. In *Proceedings of SAE 2001 in Detroit, USA*, number 2001-01-0747. Pirelli Pneumatic and SEAT, 2001.

K.M Marshek, J.F. Cuderman II, and M.J. Johnson. Performance of anti-lock braking system - part iii: Braking as a function of tire inflation pressure. In *Proceedings of SAE 2002 in Detroit, USA*, number 2002-01-0306, 2002.

F. Marvasti. Nonuniform sampling theorems for bandpass signals at or below the nyqvist density. *Trans. on Sign. Proc.*, Vol. 44(3):572–576, Mar 1996.

E. Meijering. A chronology of interpolation: From ancient astronomy to modern signal and image processing. *Proceedings of the IEEE*, 90(3):319–342, Mar 2002.

J.L. Meriam and L.G. Kraige. *Dynamics*, volume 2. John Wiley & Sons, Inc, 3rd edition, 1993.

S.K. Mitra. *Digital Signal Processing: A Computer-Based Approach*. McGraw Hill, 1998.

M. Nakajima. Device for determining initial correction factors for correcting rotational angular velocity measurment of vehicle tires. Patent no. EP855597, 1998. Sumitomo Rubber Ind. and Sumitomo Electric Industries.

M. Nakajima and I. Isshiki. Method and device for detecting pressure drop. Patent no. EP716941, 1996. Sumitomo Electric Industries and Sumitomo Rubber Ind.

C. O. Nwagboso. *Automotive sennsory systems*. Chapman & Hall, 1993.

A.V. Oppenheim and R.W. Schafer. *Digital signal processing*. Prentice Hall, 1975.

S.K. Park and R.A. Schowengerdt. Image sampling, reconstruction and the effect of samplescene phasing. *Appl. Opt*, 21(17):3142–3151, 1982.

N. Persson. Estimation properties for a tire pressure monitoring system. In *Proceedings of Reglermöte 2002 in Linköping, Sweden*, pages 108–113, 2002.

N. Persson and F. Gustafsson. Event based sampling with application to vibraton analysis in pneumatic tires. In *Proceedings of IEEE Int Conf on Acoustics, Speech, and Sig Proc 2001 in Salt Lake City, USA*, volume 6, pages 3885 –3888, 2001.

N. Persson, F. Gustafsson, S. Ahlqvist, and U. Forssell. Low tire pressure warning system using sensor fusion. In *Proceedings of ATTCE 2002 in Barcelona, Spain*, number 01ATT-272, 2001.

N. Persson, F. Gustafsson, and M. Drevö. Indirect tire pressure monitoring using sensor fusion. In *Proceedings of SAE 2002 in Detroit, USA*, number 2002-01-1250, 2002.

R.D. Poisson. Measuring relative deflection of interspaced toothed wheels on a less than once per revolution basis. Patent no.US 5327360, 1992. United Technologies Corporation.

J.D. Powell, N.P. Fekete, and C-F. Chang. Observer-based air-fule ratio control. *IEEE Control Magazine*, 18(5):72–83, Oct 1998.

J.G. Proakis and D.G. Manolakis. *Digital Signal Processing: Principles, Algorithms, and Applications*. Prentice Hall, 3:rd edition, 1996.

C. Randazzo, F. Cascio, and S. Fiorentin. A method and apparatus for detecting the presence of an at least partially deflated tyre on a motor vehicle. Patent no. EP844112, 1998. Fiat Auto Spa.

V. Rasche, R. Proksa, R. Sinkus, P. Börnet, and H. Eggers. Resampling of data between arbitrary grid using convolution interpolation. *IEEE Trans. On Medical Imaging*, 18(5), Nov 1999.

W.B. Ribbens and G. Rizzoni. Applications of precise crankshaft position measurements for engine testing, control and diagnosis. In *Proceedings of SAE 1989 in Detroit, USA*, number 890885, 1989.

O.N. Richardsson. Vehicle tire pressure monitoring system. Patent no. US 3593269, 1971.

N. Rittmannsberger. Antilock braking system and traction control. In *International Congress on Transportation and Electronics*, pages 195–202, 1988.

F. Sager and Y.L. John. Mothod for detecting a deflated tire on a vehicle. Patent no. US 5760682, 1998. Bosch Gmbh Robert.

R.D. Scott and B.M. Alan. Low tire warning system. Patent no. US 5721528, 1998. Ford Global Tech Inc.

T. Söderström and P. Stoica. *System Identification*. Prentice-Hall International, 1989.

A. Stenman. *Model on Demand: Algorithms, Analysis and Applications*. PhD thesis 571, Department of Electrical Engineering, Linköping University, Linköping, Sweden, May 1999.

P. Stoica and R. Moses. *Introduction to Spectral Analysis*. Prentice Hall, 1997.

T. Takeyasu, N. Toshiharu, T. Hiromi, I. Yunichi, and F. Shusaku. Tire air pressure detecting device. Patent no. EP783982, 1997. Nippon Denso Co.

P. Thevenaz, T. Blu, and M. Unser. Interpolation revisited. *IEEE Trans. On Medical Imaging*, 19(7), July 2000.

N. Toshiharu, O. Nobuyoshi, and T. Takeyasu. Tyre air pressure estimating apparatus. Patent no. EP925960, 1999. Denso Corp.

T. Umeno, K. Asano, H. Odashi, M. Yonetani, T. Naito, and T. Taguchi. Observer based estimation of parameter varations and its application to tyre pressure diagnosis. In *Control Engineering Practice*, volume 9, pages 639–645, 2001.

M. Unser. Splines, a perfect fit for signal and image processing. *IEEE Signal Processing Magazine*, 16(6):22–38, Nov 1999.

H.G. van Steenis and J.H.M. Tulen. A comparison of two methods to calculate the heart rate spectrum based on non-equidistant sampling. In *Proceedings of the Annual Int. Conf. of the IEEE*, volume 13, pages 670–671. Engineering in Medicine and Biology Society, 1991.

S. Vos. *Eine Indirekte Methode Zur Identifikation Fahrsicherheits- und Komfortrelevanter Reifenkenngrössen*. PhD thesis, Dem Fachberereich Mashinenbau der Gerhard-Mercator-Univerität, Gesamthochschule Duisburg, 2002.

B. Widrow, J.R. Glorer, J. McCool, J.Kaunitz, C.S. Williams, R.H. Hearn, J.R. Zeidler, E. Dong, and R.C. Goodwin. Adaptive noise cancellation: Principles and applications. In *Proceedings IEEE*, volume 63, pages 1692–1716, 1975.

J.Y. Wong. *Theory of ground vehicles*. John Wiley & Sons, Inc, 2nd edition, 1993.

N. Yoshira, H. Ikuo, N. Toshiharu, O. Noboyoshi, and T. Takeyasu. Tire pneumatic pressure detector. Patent no.EP700798, 1996. Nippon Denso Co and Nippon Soken.

I. Yuuichi, H. Ohashi, K. Asano, H. Kaway, T. Naito, N Onogi, T. Umeno, and H. Kojima. Device for estimating air pressure of tire from vibration components of vehicle wheel speed. US 5826207, 1996. Toyota Motor Co Ltd and Toyoda Chuo Kenkyusho KK.