

CHALMERS



Design and implementation of an adaptive
harmonic controller:

Active noise control in an intake system

Master's Thesis in the Master Degree programme, Sound and Vibration

LARS HANSSON

Department of Civil and Environmental Engineering
Division of Applied Acoustics
Vibroacoustics Research Group
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2012
Master's Thesis 2012:125

MASTER'S THESIS 2012:125

Design and implementation of an adaptive harmonic controller:
Active noise control in an intake system

LARS HANSSON

Division of Applied Acoustics
Vibroacoustics Research Group
Department of Civil and Environmental Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2012

Design and implementation of an adaptive harmonic controller:

Active noise control in an intake system

Lars Hansson

Master's Thesis 2012:125

Division of Applied Acoustics

Vibroacoustics Research Group

Department of Civil and Environmental Engineering

CHALMERS UNIVERSITY OF TECHNOLOGY

SE-412 96 Göteborg

Sweden

Phone : +46-(0)31-772 2200

Fax : +46-(0)31-772 2212

Cover: none

© Lars Hansson, 2012

Göteborg, Sweden 2012

Design and implementation of an adaptive harmonic controller:

Active noise control in an intake system

Lars Hansson

Chalmers University of Technology

Department of Civil and Environmental Engineering

Division of Applied Acoustics

ABSTRACT

When passive noise control techniques cannot achieve a target reduction, within a certain cost- or technology envelope, one must sometimes resolve to active methods. A relatively new technique—first patented in 1934 by Paul Lueg—which has been widely adopted, more recently due the increase in performance for low-cost computing devices.

In engine compartments there is a very limited amount of space; this together with the short distance to the receiver, makes it hard to use conventional passive methods. The application considered is a single-channel active noise control system in an intake system of an engine compartment. A previous investigation have shown that there is higher order harmonics of the fundamental rotational frequency of the engine, with high sound pressure levels, present in the intake system. The range of interest for that application was 300 to 600 Hz, and the same scenario is considered for the application under study.

The proposed controller in this thesis make use of feed-forward information from the engine, using a tachometer to tap the momentary rotational speed of the engine, consisting of a pulse signal. The proposed controller uses the adaptive LMS-algorithm, and is modified for harmonic disturbances, and to accommodate the effect of the tachometer, and the present control path.

The study of this master's thesis is to investigate how such an adaptive harmonic can be designed and implemented; and how the underlying parameters in the extended controller, effect performance, stability and model complexity. The controller and its comprising internal parts are tested in a virtual setup in MATLAB/Simulink, as well as in a simplified experimental setup.

Keywords: active noise control, intake system, adaptive, harmonic control, tachometer, LMS, single channel feed-forward, driving cycle

NOMENCLATURE

\bar{x}	complex number x
\hat{x}	amplitude of the complex number x
x^*	estimate of x
\mathbf{x}	vector x
X	matrix x
\tilde{X}	magnitude of frequency response function x
d	disturbance signal
e	error signal
e_r	residual error signal
x_{ref}	feed-forward reference signal
y	control output signal
β	performance limit gain factor
μ	step-size factor of the steepest-descent, and LMS, algorithm
μ_0	step-size factor for normalised case
G	control transfer function
P	primary path transfer function
S	secondary path transfer function
ANC	active noise control
ARMA	auto-regressive moving average
DSP	digital signal processor
FIR	finite impulse response
IIR	infinite impulse response
LMS	least mean square algorithm
SNR	signal to noise ratio

ACKNOWLEDGEMENTS

This master of science thesis was written during the spring of 2012 as part of the Master's programme in Sound and Vibrations at the Division of Applied Acoustics at Chalmers University of Technology, and as a part of the Green City Car project. The work has been supervised by Wolfgang Kropp.

I would like to thank all of my colleagues at the division, and especially my supervisor Wolfgang Kropp for his help and support. In addition I would like to express my gratitude to Per Sjösten and to Börje Wijk for all the technical support.

Gothenburg June 2012

Lars Hansson

TABLE OF CONTENTS

1.	INTRODUCTION	1
1.1.	Background.....	1
1.2.	Purpose and scope.....	1
1.3.	Aims and objectives.....	1
2.	THEORY	3
2.1.	A short introduction to Active Noise Control.....	3
2.1.1.	System description of a single-channel ANC-system.....	3
2.1.2.	Adaptive control algorithm.....	5
2.1.3.	Method of steepest-descent.....	5
2.1.4.	The LMS-algorithm.....	6
2.2.	Harmonic control.....	7
2.2.1.	Harmonics.....	7
2.2.2.	Transfer path filtering.....	9
2.2.3.	Performance control.....	9
2.2.4.	μ -normalisation and FY-LMS.....	10
2.3.	Decoding of the tachometer signal.....	11
3.	IMPLEMENTATION	12
3.1.	Introduction.....	12
3.2.	Parametric study.....	12
3.2.1.	Decoding filtering.....	12
3.2.2.	Performance control.....	14
3.2.3.	Sampling frequency.....	15
3.3.	Experimental setup.....	16
3.3.1.	Introduction.....	16
3.3.2.	Secondary path and control output normalisation.....	16
3.3.3.	Step size.....	16

3.4.	Virtual setup in MATLAB/Simulink.....	19
3.4.1.	Introduction.....	19
3.4.2.	Primary path.....	19
3.4.3.	Setup.....	19
3.5.	Multiple tonal disturbances.....	20
4.	RESULTS.....	21
4.1.	Simulation results.....	21
4.1.1.	Decoding filtering.....	21
4.1.2.	Overshoot factor.....	23
4.1.3.	Driving-cycle variations.....	25
4.1.4.	Step size.....	26
4.1.5.	Sampling frequency.....	28
4.1.6.	Performance control.....	29
4.1.7.	Transfer path.....	30
4.2.	Experimental results.....	32
4.2.1.	Without control output normalisation.....	32
4.2.2.	With control output normalisation.....	34
5.	DISCUSSION AND CONCLUSION.....	38
5.1.	Decoder.....	38
5.2.	Controller parameters.....	39
5.2.1.	Sampling frequency.....	39
5.2.2.	Experimental intake-system and control output normalisation.....	39
5.2.3.	Step size.....	40
5.3.	Complete application.....	40
6.	REFERENCES.....	42

1. INTRODUCTION

1.1. Background

When passive noise control techniques cannot achieve a target reduction, within a certain cost- or technology envelope, one must sometimes resolve to active methods. A relatively new technique—first patented in 1934 by Paul Lueg—which has been widely adopted more recently, due to the increase in performance for low-cost computing devices.

In engine compartments there is a very limited amount of space; this together with the short-distance to the receiver, make it hard to use conventional passive methods. The application considered is a single-channel active noise control system in an intake system of an engine compartment. A previous investigation of a specific engine compartment have shown that there is higher order harmonics of the fundamental rotational frequency of the engine, with high sound pressure levels, present in the intake system.

1.2. Purpose and scope

The purpose of this thesis is to design and implement an adaptive harmonic active noise controller in an intake system, and to briefly study the underlying general parameters for performance and stability conditions. The rotation speed of the engine provide information about the fundamental frequency of the engine noise. By using a tachometer signal from the engine, the controller is provided by a reference signal for the frequency of the fundamental harmonic.

The scope of the thesis is to assess the filter design of such a harmonic controller; any hardware related issues is outside the scope of the thesis, such as driver design and DSP performance. Specific broadband noise issues related to active noise control are not considered in the design and implementation of the harmonic controller. The controller should take advantage of the tachometer signal provided from the engine.

1.3. Aims and objectives

The aim of this thesis is to present a design strategy and necessary filter parameters for an adaptive feed-forward harmonic controller, in order to achieve a reduction of around 10 dB in an intake system; using a virtual setup and a simplified experimental setup, for harmonic tones in the range of 300–600 Hz. The proposed controller should include the design of the tachometer-signal decoder.

Real-time target

The controller should be an online application operating in real-time, in order to adopt to the varying environment inside the engine compartment, and outside the vehicle. The driving cycle behaves almost stochastically, and the controller must thus be adopted for such a system. This behaviour is suited for the LMS-algorithm, which is not reliant on a long time-history for its predictions of the disturbance. Its simplicity also makes it suitable for this study, and its behaviour is well-documented.

2. THEORY

2.1. A short introduction to Active Noise Control

2.1.1. System description of a single-channel ANC-system

The disturbance signal, d , represents a signal to be controlled by the means of adding a control output signal, y , to the system, see figure 1. The resulting signal, e , is called the error signal of the control system. In order to achieve control of the system, a control law function, g , could be defined as

$$y = g(d) \quad 1$$

The error signal is the sum of d and y

$$e = d + y = d + g(d) \Rightarrow g = e - d \quad 2$$

The control law is usually defined by a desired error signal, and typically the error signal is to be minimised. Generally the disturbance is not completely known, and therefore making g difficult to acquire/formulate directly. Different design strategies to formulate a control law, for a desired error signal, are therefore needed. Two fundamental strategies¹ for the single-channel case are:

- Feed-forward active control, figure 2.
- Feed-back active control, figure 3.

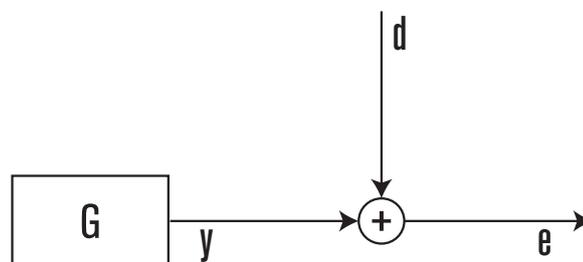


figure 1. Simplified block diagram of an single-channel ANC-system.

¹ Nelson, P.A, Elliott, S.J. (1992) *Active Control of Sound*. London: Academic Press

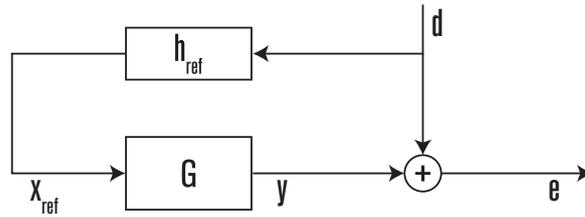


figure 2. Simplified block diagram of feed-forward control.

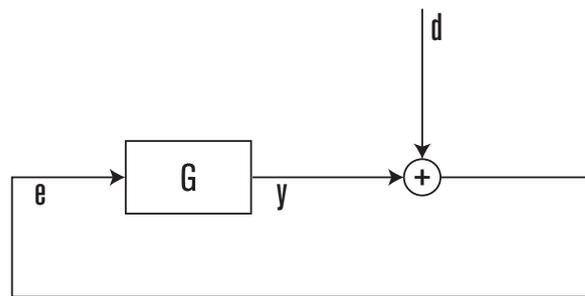


figure 3. Simplified block diagram of feed-back control.

The feed-forward controller:

The feed-forward controller make use of devices, such as detection microphones and tachometers, in order to provide a reference signal, x_{ref} , to the controller G , generalised as

$$\mathbf{x}_{ref} = \mathbf{h}_{ref}^{-1} * \mathbf{d} \tag{3}$$

Where h_{ref} represent the effect of an intermediate subsystem present, e.g. the response of the measuring microphone.

The reference signal could be said to be a representation of the disturbance signal, if h_{ref} is sufficiently known. The control law could therefore be defined as

$$\mathbf{g} = \mathbf{e} - \mathbf{d} = \mathbf{e} - \mathbf{x}_{ref} * \mathbf{h}_{ref}^{-1} \approx \mathbf{e} - \mathbf{x}_{ref} * \mathbf{h}_{ref}^{s-1} \tag{4}$$

Where h_{ref}^s is the estimate of h_{ref}

The adaptive harmonic controller described in this thesis will make use of the feed-forward approach to active noise control.

The feed-back controller:

The feed-back controller uses a *feed-back* path, by tapping e , to relate y to d , and could be written as, for this simplified system:

$$\mathbf{y} = g(\mathbf{e}) \Leftrightarrow \mathbf{g} = \mathbf{e} - \mathbf{d} \Leftrightarrow \mathbf{e} = \mathbf{d} + g(\mathbf{e}) \tag{5}$$

However as a system consists of more subsystems, the control law is of a more complex nature for a real-world control system. The feed-back controller is not discussed further in this thesis.

Electrical error signal:

Typically the error signal is filtered through the measuring equipment, and therefore the true error signal is unknown, unless the transfer function \mathbf{h}_e is known:

$$\mathbf{e}_e = \mathbf{h}_e * \mathbf{e} + \mathbf{e}_n \tag{6}$$

If the effect of \mathbf{h}_e is small and the electrical noise \mathbf{e}_n neglectable, the estimated error signal, i.e. electrical error signal, \mathbf{e}_e , is a sufficient estimator of \mathbf{e} , i.e. $\mathbf{h}_e \approx \delta$. In general, the error signal are to be minimised, and therefore an approximation could be made as:

$$\dot{\mathbf{e}}_e \rightarrow \mathbf{0} \Leftrightarrow \dot{\mathbf{e}} \rightarrow \mathbf{0} \tag{7}$$

2.1.2. Adaptive control algorithm

As seen in equation 4, the control law is a function, g , of the transformed reference signal. In order to successfully achieve a desired error $\mathbf{e}_{desired}$, the transfer function \mathbf{h}_{ref} , must be sufficiently estimated. If the estimate \mathbf{h}_{ref}^* is a perfect copy, for all possible scenarios, then the control law could easily be defined *off-line*, i.e. before/during installation, and the controller could be left to its own devices. But if the estimate could not be described beforehand, but instead is highly dependant on unforeseeable circumstances, another approach to a control law formulation must be deployed.

The adaptive approach use methods to continuously update the control law function, $g(x_{ref}(t)) = y(t)$, to achieve a desired error signal. According to equations 2 and 3, the controller must know \mathbf{h}_{ref} in order to be able to achieve a desired error signal. Therefore it would be a good idea to continuously update the estimated reference transfer function, $\mathbf{h}_{ref}^* \rightarrow \mathbf{h}_{ref}$, within the function g , to get:

$$\mathbf{d} + g \circ h_{ref}^{-1}(\mathbf{x}_{ref}) = \mathbf{e}_{desired} \Leftrightarrow \lim_{k \rightarrow k_{end}} h_{ref_k}^* = h_{ref} \tag{8}$$

Where k is the number of iterations in the adaptive algorithm.

The algorithm for finding the true estimate, could not be defined arbitrarily. It exists a number of tools to use for this kind of problem; the method used in this thesis is defined in section 2.2. Note that \mathbf{h}_{ref} must be non-casual², or act as a *pass-through*, in order to achieve a casual³ control law for the feed-forward case. The adaptive algorithm can be used *off-line*, as well as *on-line*, i.e. in real-time.

2.1.3. Method of steepest-descent

The minimisation of the disturbance/electrical noise in an ANC-system could be described as a cost-function minimisation problem. Typically the mean-square value of the electrical noise, e_e , are to be minimised. A cost-function to be minimised, J , could be defined as a function, f , of a generalised variable, x :

$$f(x) = J | x \in [x_i, x_n] \tag{9}$$

Note that $f(x)$ is continuously differentiable for $\forall x \in [x_i, x_n]$.

2 Related to future input values.
3 Related to present, or past, input values.

A minima of $f(x)$ could be found in the end-points of the restricted interval, $[x_l, x_h]$, or in a local minima, $x \in (x_l, x_h)$. A minima at x_{min} could be defined by:

$$\frac{df}{dx}(x_{min}) = 0 \wedge \frac{d^2f}{dx^2}(x_{min}) > 0 \quad 10.$$

The method of steepest-descent⁴ is an adaptive algorithm, i.e. iterative, in the sense that x_{min} is found using a series of iterations, i . The algorithm could be described as

$$x_i = x_{i-1} - \mu \frac{df}{dx}(x_{i-1}) \quad 11$$

Where μ is a step-size factor.

As one can see, the algorithm will be most likely to find a local minima at x_{min} , if there exist no maxima in the interval $[x_0, x_{min}]$:

$$\lim_{i \rightarrow \infty} x_i = x_{min} \Leftrightarrow \neg \exists x \in [x_0, x_{min}]: \frac{df}{dx} = 0, \frac{d^2f}{dx^2} < 0 | x_{min}, x_0 \in [x_0, x_{min}], \mu \ll 1 \quad 12.$$

Where x_0 is the initial value for the algorithm, and i is the number of iterations. μ is sufficiently small.

If eq. 12 does not hold, the algorithm will be less likely to find x_{min} , and might instead converge to an adjacent minima, or an end-point in the interval—which may, or may not, be the global minima of the function in the interval. It can be seen as a rule of thumb for converging, as the true convergence is dependant on the shape of f and the factor μ .

2.1.4. The LMS-algorithm

The controller, G , can be seen as a FIR-filter, with filter coefficients \mathbf{w}^5 . By using equation 11 the filter coefficients can be updated as

$$\mathbf{w}_i = \mathbf{w}_{i-1} - \mu \frac{\partial \mathbf{J}}{\partial \mathbf{w}_{i-1}} \quad 13.$$

The error signal could be described, using equation 2, as

$$\mathbf{e} = \mathbf{d} - \mathbf{y} = \mathbf{d} - \mathbf{w}^T \mathbf{X}_{ref} \quad 14.$$

The cost-function J could be, for this case, defined as the error signal squared, which represents the energy in the error.

$$\frac{\partial \mathbf{J}}{\partial \mathbf{w}} = \frac{\partial \mathbf{e}^2}{\partial \mathbf{w}} = -2\mathbf{X}_{ref} \mathbf{e} \quad 15.$$

Equations 13 and 15 give the LMS-algorithm

$$\mathbf{w}_i = \mathbf{w}_{i-1} + 2\mu \mathbf{x}_{ref,i-1} \mathbf{e}_{i-1} \quad 16.$$

4 Fedoryuk, M.V. (2001) *Encyclopedia of Mathematics*. Springer

5 Nelson, P.A, Elliott, S.J. (1992) *Active Control of Sound*. London: Academic Press

The algorithm, eq. 16, for updating the controller becomes quite simple, and requires little amount of calculations per iteration⁶, compared to other algorithms—such as the RLS-algorithm.

2.2. Harmonic control

For this application it is important to study the behaviour of the present harmonic disturbances, in order to understand how control can be achieved. The present engine noise in the intake system consist of many overtones related to the fundamental frequency of the engine, but only a few of those are considered disturbances in an ANC sense. In order to achieve control, the order of the unwanted overtones must be known, and by knowing the fundamental frequency at any given time, the frequencies of the disturbances could be obtained. How to obtain the fundamental frequency from the tachometer signal is discussed in section 2.3.

There exist other methods for identifying the disturbances, e.g. using FFT-analysis; but these other methods are not considered in this thesis.

2.2.1. Harmonics

A harmonic tone, \bar{z} , could be described in the complex plan, $\bar{z} \in \mathbb{C}$, using Euler's formula, as:

$$\bar{z} = \hat{z}(t)e^{j\theta(t)} \wedge \Re(\bar{z}) = \bar{z}_R = \cos\theta(t) \wedge \Im(\bar{z}) = \bar{z}_I = \sin\theta(t) \quad 17.$$

Where $\theta(t)$ is the angle of \bar{z} at time t , and \hat{z} is the amplitude of \bar{z} . $\theta'(t)$ is usually denoted as the angular frequency ω , and the time-invariant part of θ is usually denoted as the phase, φ , of \bar{z} .

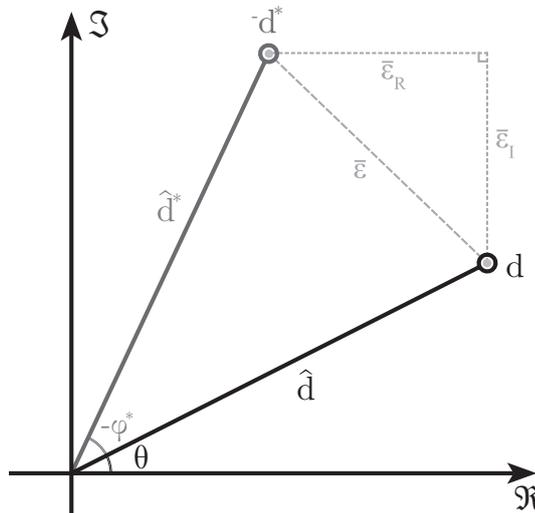


figure 4. Plot of the disturbance signal, control output signal and the corresponding error.

6 Elliott, S. (2001) *Signal processing for active control*. London: Academic Press

For this section lets consider a single harmonic disturbance, d , with an angular function θ , see figure 4. For simplicity the energy in the error signal, e^2 , are to be minimised. Equations 2 and 17 give

$$\bar{d}(t) = \hat{d}(t)e^{j\theta(t)} \Leftrightarrow \bar{e}(t) = \bar{d}(t) + \bar{y}(t) \quad 18.$$

Where \bar{y} is the complex control output signal.

If the error, $e^2 \rightarrow 0$, lets call the control output signal the negative estimate of the disturbance, \bar{d}^* . The residual error, e_r , could therefore be re-written as

$$e_r(t) = \bar{d}(t) - \bar{d}^*(t) \equiv \mathbf{e}_r = \bar{\mathbf{d}} - \bar{\mathbf{d}}^* \quad 19.$$

$$\mathbf{e}_r = \hat{\mathbf{d}}e^{j\theta} - \hat{\mathbf{d}}^*e^{j(\theta^* - \varphi^*)} \quad 20.$$

Here we can see that if ω is given, from the reference input to the controller, and therefore adequately determined, $\omega^* = \omega$, the residual error can be minimised if the amplitude \hat{d}^* and phase φ^* can assume the right values:

$$\theta^* - \varphi^* = \theta \wedge \hat{\mathbf{d}}^* = \hat{\mathbf{d}} \quad 21.$$

As shown in section 2.1.2 these values can not be determined if the system h_{ref} is not known. However, by using the method of steepest-descent (eq. 11 and 13), these values can be obtained by an iterative process, i.e. the LMS-algorithm. The cost-function to be minimised is the square value of the residual error, and the coefficients to be determined are the amplitude and the phase. In the real-world the harmonic z is described as $\Re(\bar{z})$; the real-valued residual error is similarly described. The cost-function is therefore

$$\Re(e_r) = e_r \wedge J = e_r^2 \quad 22.$$

The algorithms could be written as

$$\hat{d}_i^* = \hat{d}_{i-1}^* - \mu_1 \frac{\partial e_R^2}{\partial \hat{d}_{i-1}^*} \quad 23.$$

$$\varphi_i^* = \varphi_{i-1}^* - \mu_2 \frac{\partial e_R^2}{\partial \varphi_{i-1}^*} \quad 24.$$

The gradient could be described using equations 17 and 20—which give a simple expression for the gradients.

$$\frac{\partial e_R^2}{\partial \hat{d}^*} = \frac{\partial (\hat{d} \cos \theta - \hat{d}^* \cos(\theta^* - \varphi))^2}{\partial \hat{d}^*} = (\hat{d} \cos \theta - \hat{d}^* \cos(\theta^* - \varphi)) \cdot -2 \cos(\theta^* - \varphi) = -2e_r \Re(-\hat{d}^*) \quad 25.$$

$$\frac{\partial e_R^2}{\partial \varphi^*} = \frac{\partial (\hat{d} \cos \theta - \hat{d}^* \cos(\theta^* - \varphi))^2}{\partial \varphi^*} = (\hat{d} \cos \theta - \hat{d}^* \cos(\theta^* - \varphi)) \cdot -2 \hat{d}^* \sin(\theta^* - \varphi) = -2e_r \Im(-\hat{d}^*) \quad 26.$$

Implementing these gradients in equations 23 and 24 give

$$\hat{d}_i^* = \hat{d}_{i-1}^* + 2\mu_1 e_R \Re(-\hat{d}_{i-1}^*) \quad 27$$

$$\varphi_i^* = \varphi_{i-1}^* + 2\mu_2 e_R \Im(-\hat{d}_{i-1}^*) \quad 28$$

Equations 27 and 28 is the LMS-algorithm used for the proposed adaptive harmonic controller. The implementation of this algorithm is described in section 4.

2.2.2. Transfer path filtering

As seen in section 2.2.1, the algorithm uses the estimate of the disturbance signal, and for a simple ideal ANC-system it is no problem. Generally, however, the control output signal is filtered through a intermediate system, called the secondary path S (see figure 5):

$$\mathbf{d}^* = \mathbf{y} * S \quad 29$$

The secondary path could be simplified as a phase- and amplitude-shift of the control output signal, y . As discussed by Elliott⁷ and Morgan⁸ the phase-difference between y and d^* must be less than 90° to maintain stable control. If S is known, y could be filtered as

$$y = y^* * S^{*-1} \Rightarrow d^* = (y^* * S^{*-1}) * S \approx y^* \quad 30$$

To obtain stability, only the phase error is important, thus only the phase shift of S , must be compensated:

$$y^*(\omega) = y(\omega) \cdot \hat{S}^*(\omega) \approx d^*(\omega) \quad 31$$

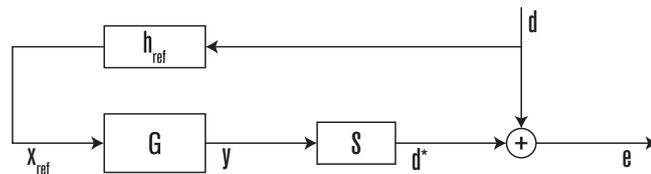


figure 5. Block diagram of feed-forward control with secondary path.

2.2.3. Performance control

For some applications it is desirable to be able to control the reduction/performance achieved by the controller, e.g. in sound design applications. It can also be used to maintain control for a stochastic system, where stability can be difficult to achieve. Such a case could be an undamped system, where it is impossible for the controller to achieve total control of extreme amplitudes of sound and/or vibrations.

7 Elliott, S. (2001) *Signal processing for active control*. London: Academic Press

8 Morgan, D. R. (1980) An analysis of multiple correlation cancellation loops with a filter in the auxiliary path. *IEEE Transactions on acoustics, speech, and signal processing*, vol. 28, no. 4, pp. 454-467.

The LMS-algorithm in section 2.2.1 minimises the cost function and is therefore not suited for such a control situation. However, there is a technique that can be used as a workaround for this problem, as proposed by Oliveira et al⁹. See figure 6.

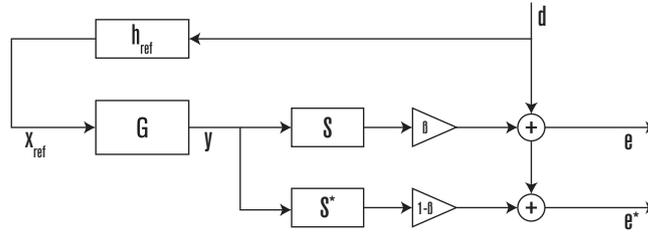


figure 6. Block diagram of feed-forward control with performance control.

The control output signal is sent through both the real system, S , and an estimate of the system, S^* , with a applied gain of β and $1-\beta$, respectively. Thus when adding those signal together with the desired signal, the same error signal will be seen by the controller, as without any *performance-control* scheme added to the system. Therefore can the error signal, e^* , be minimised, while the true error signal, e , can fulfil the desired error.

2.2.4. μ -normalisation and FY-LMS

Equations 27 and 28 give the basic adaptive control algorithm for the investigated application, but the step-size factor is still unknown. As seen in equation 12, a badly chosen μ , would lead to instability, as the algorithm will not converge to the true value—even with convergence, a optimal, or even good, convergence speed is not guaranteed. By normalising the step-size, so it becomes a function of the input energy, equation 32¹⁰, or a function of the secondary/forward path, equation 33¹¹, the algorithm can be optimised for a given signal¹².

$$\mu = \frac{\mu_0}{x^T x} \quad 32.$$

$$\mu = \frac{\mu_0}{|\hat{S}|^2} \quad 33.$$

where \hat{S} is the magnitude of the secondary path frequency response function

- 9 Oliveira, Leopoldo P.R. de, et al. (2010) NEX-LMS: A novel adaptive control scheme for harmonic sound quality control. *Mechanical Systems and Signal Processing*, vol. 24, no. 6, pp. 1727–1738.
- 10 Larsson, Erik. G. (2009) Enhanced-Convergence Normalized LMS Algorithm. *IEEE Signal Processing Magazine*, vol. 26, no. 3, pp. 81–82, 95.
- 11 Sjösten, Per. (2003) *Active Noise Control of Enclosed Sound Fields, Optimising the performance*. Gothenburg: Chalmers University of Technology. (PhD Thesis, Division of Applied Acoustics)
- 12 Elliott, S. (2001) *Signal processing for active control*. London: Academic Press

The constant μ_0 is of importance for stability and convergence rate. The stability criterion is discussed by Johansson et al¹³, and could be simplified as:

$$\mu_0 \in [0, LRH] \quad 34$$

where L is the number of secondary sources, R is the number of noise sources and H is the number of controlled harmonics from each noise source

Control output normalisation / filtered y

If the magnitude of the secondary path response function is normalised, the step-size function could simply be defined using an optimised scalar value, μ_0 . However this is unlikely to occur by default; the control output, y , must therefore be filtered to achieve a normalised output in the range of interest. As shown in section 2.2.2, the residual secondary path phase response function must be known, and the output normalisation filter must be included in the internal phase response model.

2.3. Decoding of the tachometer signal

The reference signal, x_{ref} , is given as a tacho-signal, representing the fundamental frequency of the engine. It is given as a series of pulses—one pulse per revolution. For this application, the pulse signal is reconstructed into an amplitude signal, where the amplitude represent the present frequency. This fundamental frequency is then multiplied to obtain the desired overtones, and to identify the disturbance signal.

$$\omega_{0_i} = 2\pi \frac{f_s}{k_i} \quad 35$$

where k is the number of samples between each pulse, at a sampling rate of f_s .

The true fundamental frequency is continuous, whereas the obtained frequencies are *sampled* at a low rate—the fundamental frequency itself. To increase the precision in the estimated frequency between pulses, the signal needs to be filtered; a zero-phase low-pass filter would be suitable, but, due to the realtime target, such filters can not be used. A prediction model must therefore be used, as there is no additional information in the pulse signal to extract. Taylor approximation provides a simple model for function estimation:

$$f^*(x) = \sum_{i=0}^n \frac{f^{(i)}(x_0)}{i!} (x - x_0)^i \quad 36$$

where f^* is the approximation of f around x_0 , and n is the order of the Taylor series

The true derivate is unknown, but could be estimated as

$$\frac{df}{dx} \approx \frac{f(x) - f(x_0)}{x - x_0} \quad 37$$

The implementation of this section is discussed in 3.2.1.

13 Johansson, S. et al. (2000) *A Novel Multiple-Reference Algorithm for Active Control of Propeller-Induced Noise in Aircraft Cabins*. Karlskrona: Blekine Institute of Technology (Research report no. 16/00, Department of Telecommunications and Signal Processing)

3. IMPLEMENTATION

3.1. Introduction

This chapter covers the structure of the proposed controller, how it is implemented, and how the considered parameters are investigated, regarding convergence speed, stability, overall performance and sensitivity to variations in drive cycles, i.e. variations in the tachometer-signal.

The *complete* adaptive harmonic controller is tested using two different methods:

- Computer based simulations in MATLAB/Simulink, covered in section 3.4.
- Experimental setup, covered in section 3.3.

3.2. Parametric study

3.2.1. Decoding filtering

The fundamental frequency of the engine is captured using a tachometer; in order to acquire this frequency the tachosignal must be decoded, so that x_{ref} is a sufficient frequency estimate of the disturbance signal, d . The information is incoming at a low rate, and must therefore be low-pass filtered; effectively it will act as smoothing filtering with upsampling.

The filter design is simplified, in order to maintain control of the complete ANC chain, and to make it easier to study the effects of the filter parameters—thus making more *exotic* filters unavailable for this study. Using simple textbook filters¹⁴, as well as using a filter functioning as described in section 2.3, the performance of the controller is investigated.

The filters selected for further investigation are (see figure 7 and figure 8):

- MA FIR — 25 ms window—with a low amount of overshoot
- IIR, 1:st order Butterworth—with a medium amount of overshoot
- Taylor-approximation filter, 1st order—with a low amount of overshoot

The filters chosen, are not optimal, but provide good attenuation for quite low frequencies, with adequate step delay. The selected filters should show the differences in overall performance, when using different filters in the decoding stage, and could therefore be suited for future in-depth investigation.

14 Lyons, Richard. G. (2010) *Understanding Digital Signal Processing, 3:rd Edition*, New Jersey: Prentice Hall

For simplicity, all filters are evaluated using a time-delay of 4 ms between engine and disturbance; other than that, the true transfer function of the primary path is unknown to the controller. The transfer function of the secondary path is known for this study, thus making the results easier to interpret in a decoding filter design sense.

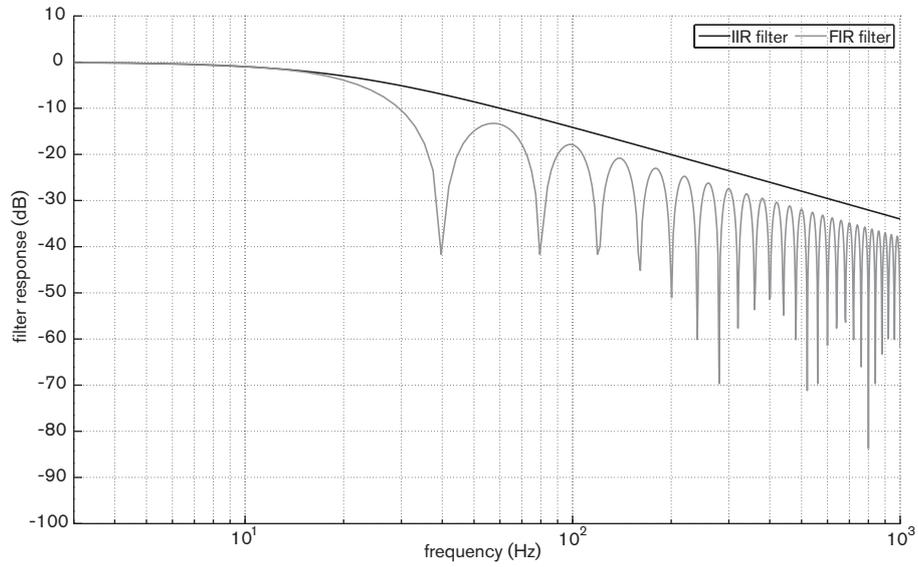


figure 7. Frequency response of the filters used in the decoder

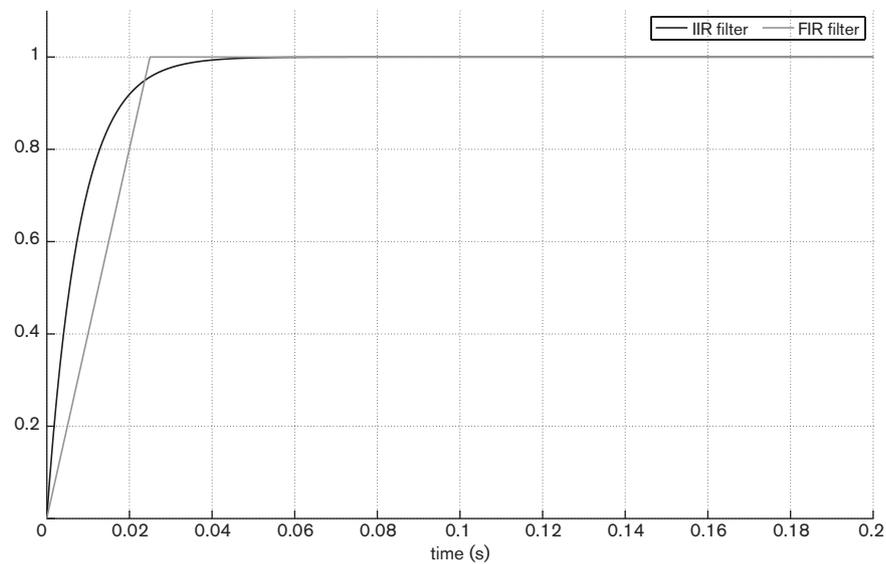


figure 8. Step response of the filters used in the decoder

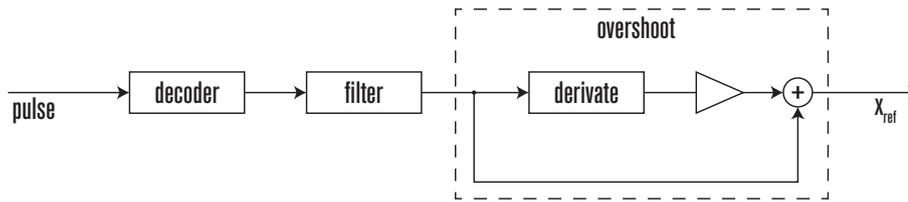


figure 9. Block diagram of decoder filter stages

Setup

A signal¹⁵, representing the true fundamental frequency of the engine, is created in a Matlab script. A binary pulse signal is extracted from that signal by creating a pulse for every revolution—this artificial pulse signal represents a perfect tachometer signal in the real-world application. As the speed-signal is a representation of the driving-cycle, care must be taken to ensure consistency throughout this investigation, as well as real-world applicability.

For this thesis the main approach to the driving-cycle design is to use simple cycles, and instead study the effect of variations in basic parameters:

- Start frequency
- End frequency
- Run-up-, and run-down duration
- Linear, logarithmic and sinusoidal run-up *shapes*.

By using these parameters, consistency as well a wide variety in driving-cycles can be achieved. As seen in figure 8, the step response is not adequately fast, as it falls well outside the chosen 4 ms¹⁶ time-window. To get around this problem, a simple prediction model (see figure 9) is applied (overshooting). Note that, even though it is similar to the approximation described in section 2.3, they are not applied identically. The Taylor-approximation filter is supposed to mimic a zero-phase low-pass filter¹⁷; whereas the overshoot function is supposed to predict future values—acting somewhat as a time-reversing filter¹⁸.

In figure 10, the true speed-signal and the unfiltered extracted speed-signal can be seen. As the pulses represents the *mean* frequency in the interval, the extracted signal will always be *delayed*.

3.2.2. Performance control

A performance control function is implemented in the controller, as shown in section 2.2.3. The controller uses the measured secondary path impulse response as the internal model. An adaptive splitter, applies gain to the external- and the internal control output signal as a function of frequency and/or desirable attenuation.

15 Referred to as the speed-signal.

16 4 ms is equal to an transmission path, in air, of 1.36 m.

17 Mitra, S.K. (2001) *Digital Signal Processing*, McGraw-Hill

18 Elliott, S.J. (1998) Filtered reference and filtered error LMS algorithms for adaptive feedforward control. *Mechanical Systems and Signal Processing*, vol. 12, no. 6, pp. 769–781.

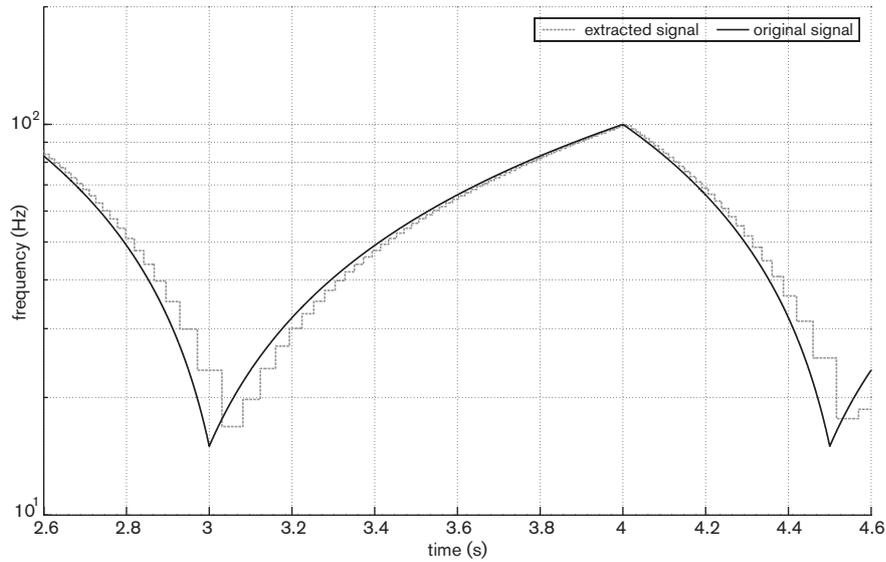


figure 10. Driving-cycle: original speed-signal and the unfiltered extracted signal: 900 rpm - 6000 rpm with a 1 s run-up, and 0.5 s run-down

3.2.3. Sampling frequency

The proposed adaptive algorithm¹⁹ is working in discrete time, and its iteration frequency is referred to its sampling frequency, from now on. All parts in the extended controller²⁰ can run at independent frequencies, however for simplicity all parts are working at the same frequency. The effects of varying the global sampling frequency is the subject to a small parametric study. Basically the performance of the controller is studied in the virtual setup²¹, for various sampling frequencies:

- 3 kHz
- 5 kHz
- 30 kHz
- 300 kHz
- 3 MHz

As suggested in section 2.2.4 there is for every instance an optimal step size, and therefore a least amount of iterations required, independent of sampling frequency. The optimal convergence should therefore be reduced in time for higher sampling frequencies.

¹⁹ Section 2.2.1

²⁰ The complete decoder, filter stage, controller and AD-DA conversion system.

²¹ Section 3.4

3.3. Experimental setup

3.3.1. Introduction

The control system comprising an engine²², an intake system and the controller, is reduced in complexity, in order to investigate the properties of the controller in its more isolated state. The engine, responsible for the disturbance signal, is replaced by a loudspeaker, known as the primary source. The intake system is simplified as a duct system. The control output source is replaced by a second loudspeaker, mounted to the side of the tube, on the opposite end of the primary source—which is mounted in the duct direction. By making these simplifications, the disturbance signal could more easily be controlled²³, and thus monitored. It should be noted that even though the disturbance signal is artificial, it is unknown to the controller. The controller, as well as the primary source, is programmed using Matlab/Simulink²⁴, and run on-line on an outboard DSP²⁵. A simplified schematic of the setup can be seen in figure 11.

3.3.2. Secondary path and control output normalisation

In order to maintain control, the estimate of the secondary path must be adequately accurate, as discussed in 2.2.2. For the experimental setup the transfer path between the secondary source and the error microphone is measured—see appendix A.2 for details regarding measurements, instrumentation and signal processing. The measured transfer function is also used in the virtual setup, see 3.4.

As seen in figure 12 there is a troublesome area between 300– and 400 Hz, with a magnitude drop of around 35 dB, and a phase shift of $\sim 150^\circ$. The secondary path can also be said to be somewhat resonant in the frequency range of interest, 300–600 Hz. By applying a small amount of damping to the system, and using a custom 30th-order IIR-filter, the secondary path is normalised as discussed in section 2.2.4; thus eliminating the problematic anti-resonant behaviour, see figure 13.

3.3.3. Step size

As described in section 2.2.4, the step-size is normalised according to the energy in the secondary path system. The factor μ_0 is important for stability and convergence speed; the valid range of μ_0 for this system is therefore investigated.

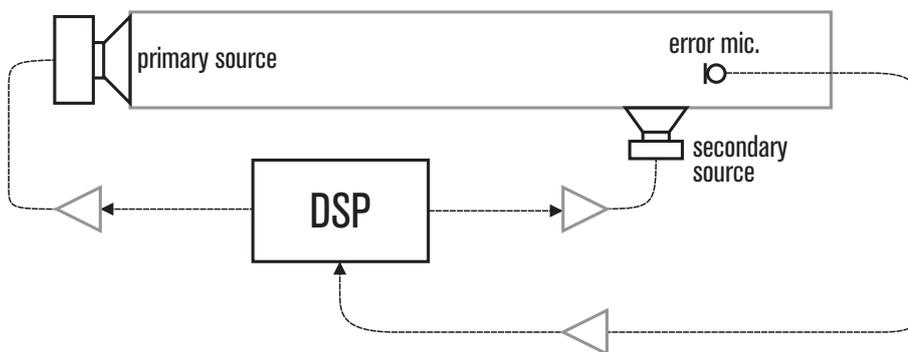


figure 11. Schematic of experimental setup

- 22 Used in an extended sense, and thus include all components in the engine compartment involved in the disturbance signal.
 23 The disturbance signal is a product of the primary output and the primary path transfer function.
 24 Using the same basic setup as in the virtual setup, see 3.4.
 25 See appendix A.2.1.

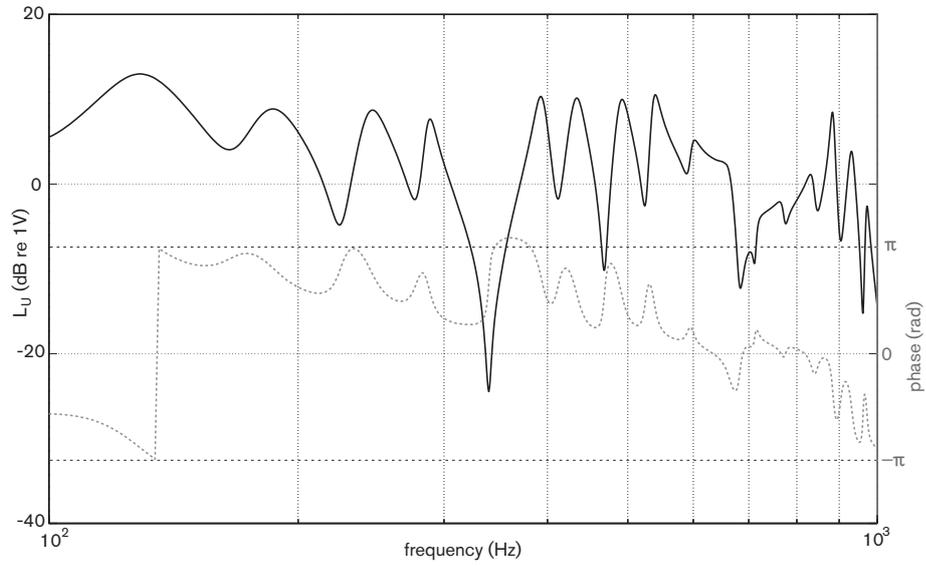


figure 12. Magnitude- and phase response of the secondary path, S. Before correction and filtering.

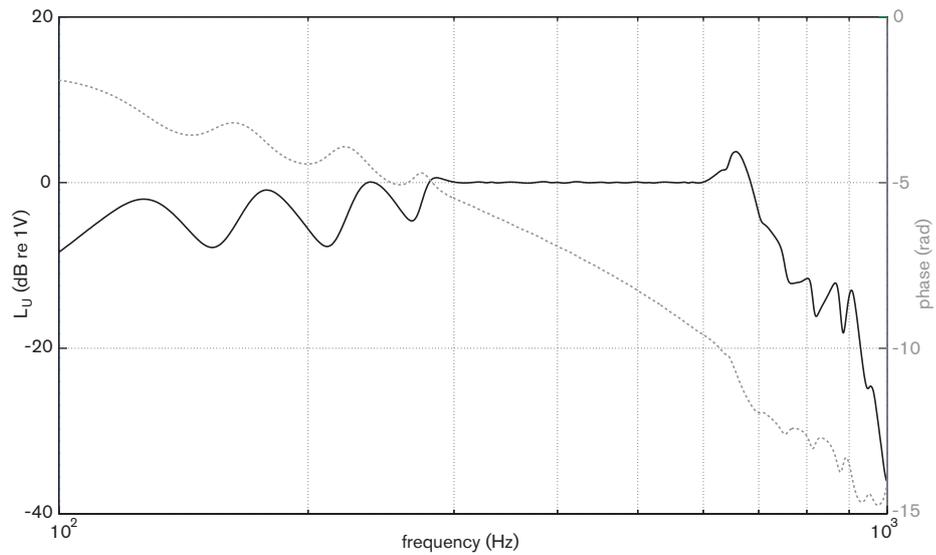


figure 13. Magnitude- and phase response of the secondary path, S. After correction and filtering.

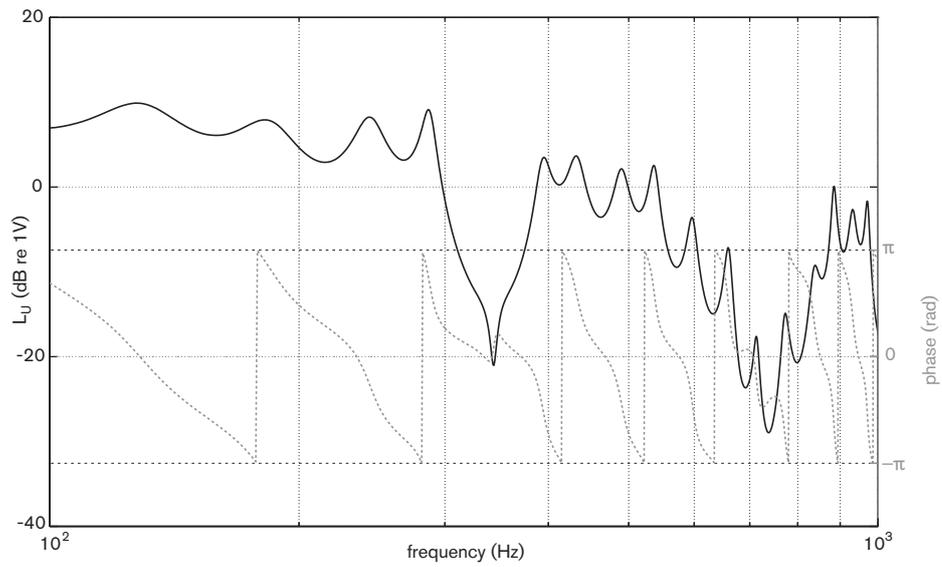


figure 14. Magnitude- and phase response of the primary path, P. Before correction of secondary path.

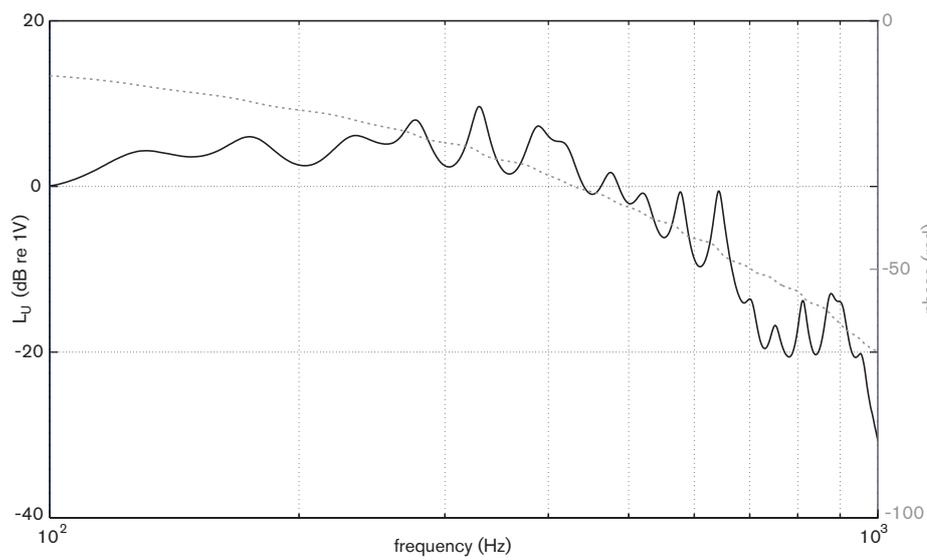


figure 15. Magnitude- and phase response of the primary path, P. After correction of secondary path.

3.4. Virtual setup in MATLAB/Simulink

3.4.1. Introduction

While the experimental setup provide a good approximation of the real-world application, a virtual setup has the advantage of added convenience, speed and control²⁶. Small parts of the controller can be monitored and recorded for analysis purposes, whereas in the experimental setup such probing would require much effort. The complete controller should therefore be suited for testing in the experimental setup, while the comprising components should be suited for virtual testing.

Another advantage provided in an virtual setup is the exclusion of parts such as inaccuracies in transfer path estimations, the disturbance signal and the decoding stage—which can be useful whenever such inaccuracies provide an analysis disturbance. As the decoder is a true digital application, the performance could solely be analysed using a virtual setup. The controller is a function of the real-world, making the virtual behaviour, only a rough estimation of the true behaviour. For improved accuracy, and for validation purposes, the virtual setup make use of the measured primary- and secondary paths. The virtual setup is mostly tested using no normalisation filter in the control output stage.

3.4.2. Primary path

The primary path transfer function is measured similar to the secondary path (3.3.2). It is used in the virtual setup to simulate the experimental setup. The measured frequency response of the primary path can be seen in figure 14. Note that the phase response is wrapped, for comparison reasons. The dip between 300- and 400 Hz is similar to the secondary path. The primary path compensation is simplified as a pure delay, as the primary path is unknown for the real-world application. The error in the estimated delay, and its effects on overall performance is assessed.

3.4.3. Setup

Two signals are created as described in 3.2.1. They are pre-stored in memory and fetched by a Simulink model, incoming at the set global sampling frequency. The Simulink model is a representation of the control system—i.e. the virtual setup—the model contains (see figure 16):

- Decoder, with internal filter stages.
- Controller, as defined in equations 27, 28 and 31.
- Secondary source, with secondary path.
- Primary source, with primary path.
- Error path (although for most simulations negligible).

The true frequency signal is sent to the primary source model, where optional random- and tonal noise can be applied; the pulse signal is processed in the decoder and used as the reference signal, x_{ref} (see equation 16). The signals of interest, such as the residual error, are recorded and stored in memory, for later analysis in Matlab—but those signals are also monitored in simulation-time²⁷.

26 Morrison, Margaret. (2009) Models, measurement and computer simulation: the changing face of experimentation. *Philosophical Studies*, vol. 143, no. 1, pp. 33-57.

27 Similar to real-time, but using virtual/simulated time instead of actual time.

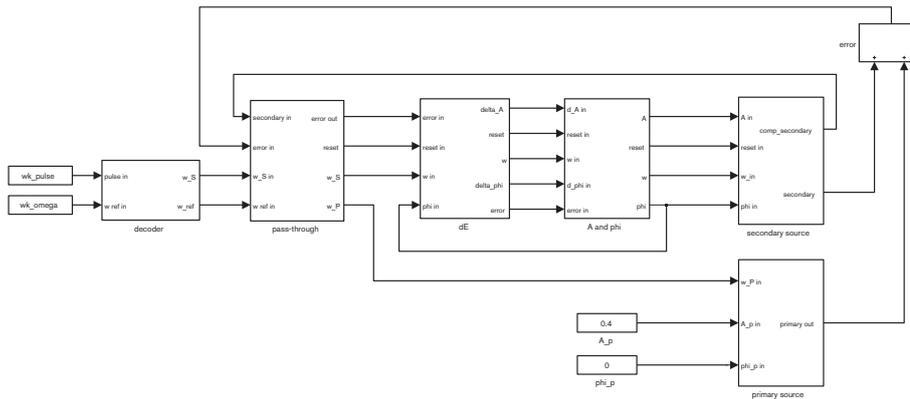


figure 16. Block diagram of virtual setup in MATLAB/Simulink.

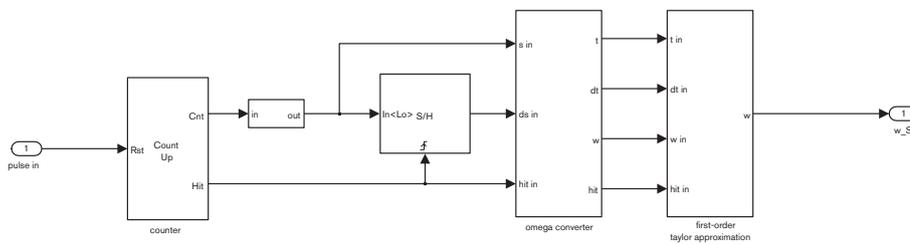


figure 17. Block diagram of decoder in MATLAB/Simulink.

3.5. Multiple tonal disturbances

For multiple tonal disturbances²⁸ in the intake noise, a number of controllers run in parallel—as seen in figure 18. The performance of such a system is subject to a limited study.

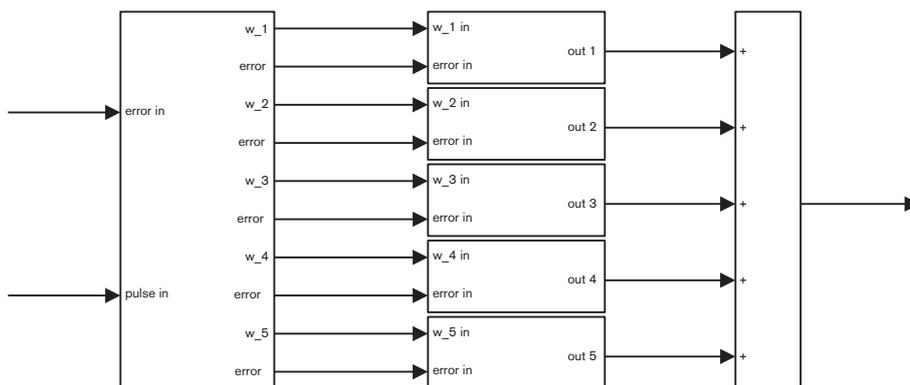


figure 18. Block diagram of multiple controllers in MATLAB/Simulink.

28 All of which are related to the fundamental frequency of the engine.

4. RESULTS

4.1. Simulation results

4.1.1. Decoding filtering

The filters used in the decoder are tested in the virtual setup, with a known secondary path. The resulting difference in the residual error can be seen in figure 19. The driving cycle is a simple sawtooth shaped function with a run-up time of 1 ms, and a run-down time of 0.5 ms, for the frequency range 150–1000 Hz²⁹. The FIR-filter uses a very low amount of overshooting, whereas the IIR- and the Taylor approximation filter uses a tuned amount – to see the effects of variable overshooting. As seen in figure 20, the taylor-approximation is more effective for the range of 150–750 Hz. The inaccuracies due to the low rate of incoming pulses for low rotational speeds can be seen in figure 21. A more detailed plot of the accuracy in the decoding above 750 Hz can be seen in figure 19. In section 5, all results are from using the taylor-approximation filter, unless otherwise noted.

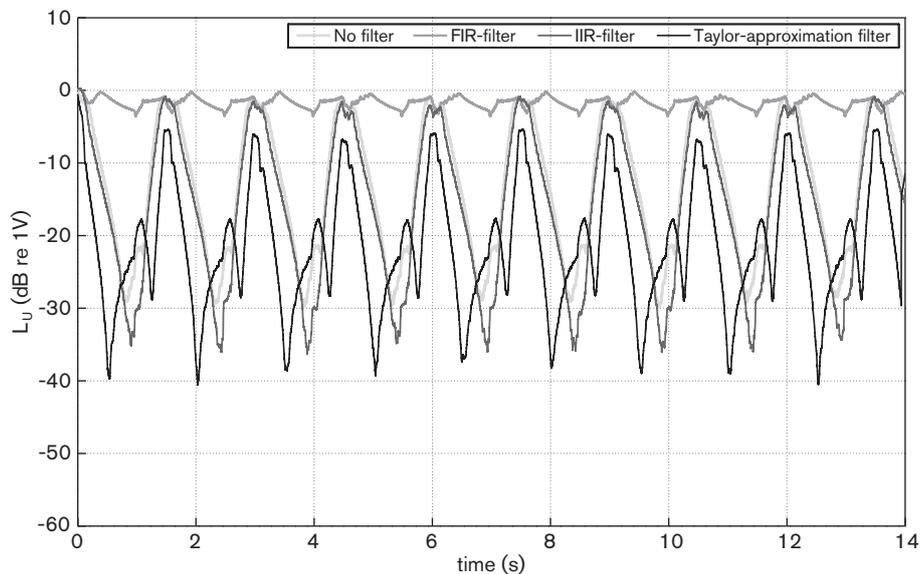


figure 19. Error gain using different decoder filters. 1 s run-up, 0.5 s run-down for a sawtooth shaped driving-cycle. 125 ms integration time.

²⁹ For the 10th overtone of the fundamental frequency, which translates as 900–6000 rpm.

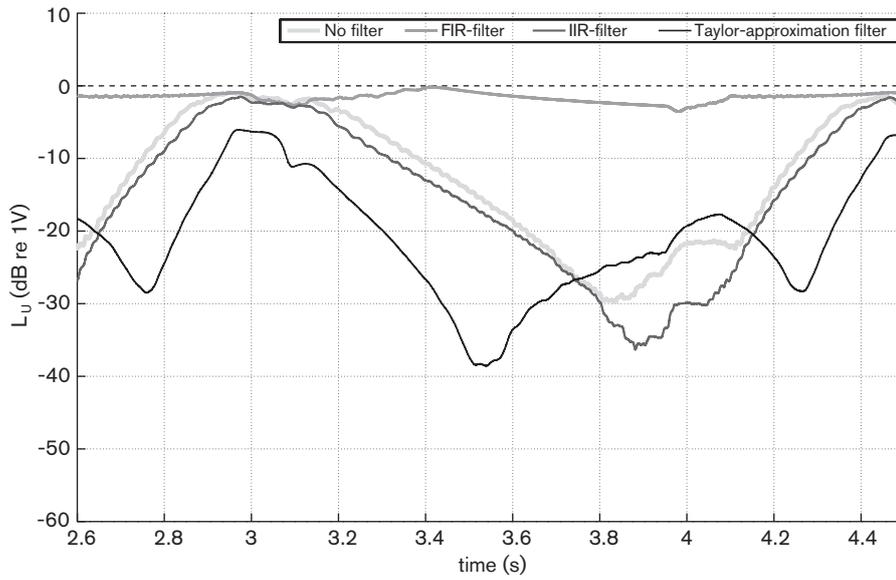


figure 20. Error gain³⁰ using different decoder filters. 1 s run-up, 0.5 s run-down for a sawtooth shaped driving-cycle.

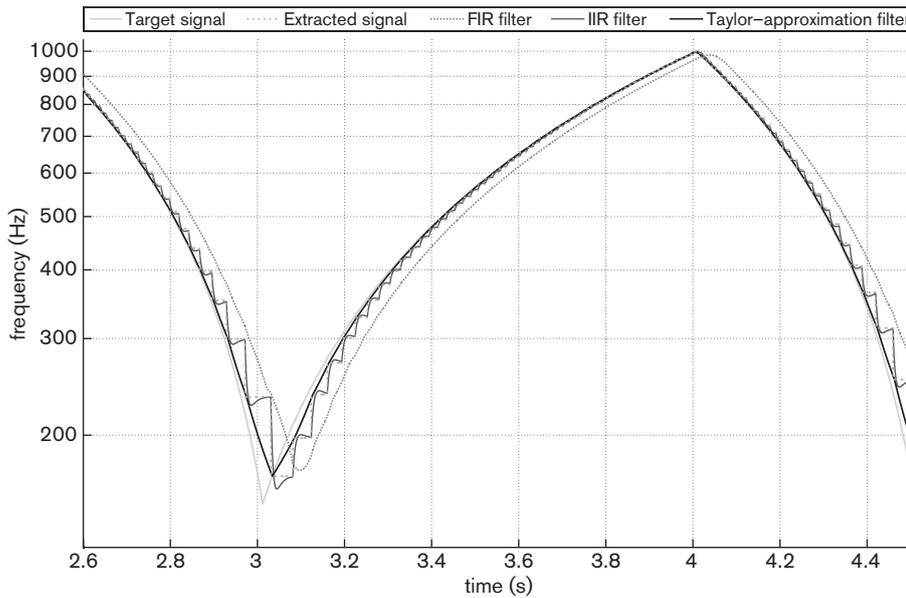


figure 21. Frequency estimation using different decoder filters. 1 s run-up, 0.5 s run-down for a sawtooth shaped driving-cycle.

³⁰ In this section, all error gains is filtered using fast SPL filtering, i.e. 125 ms integration, and all sound levels are referenced to 1 V.

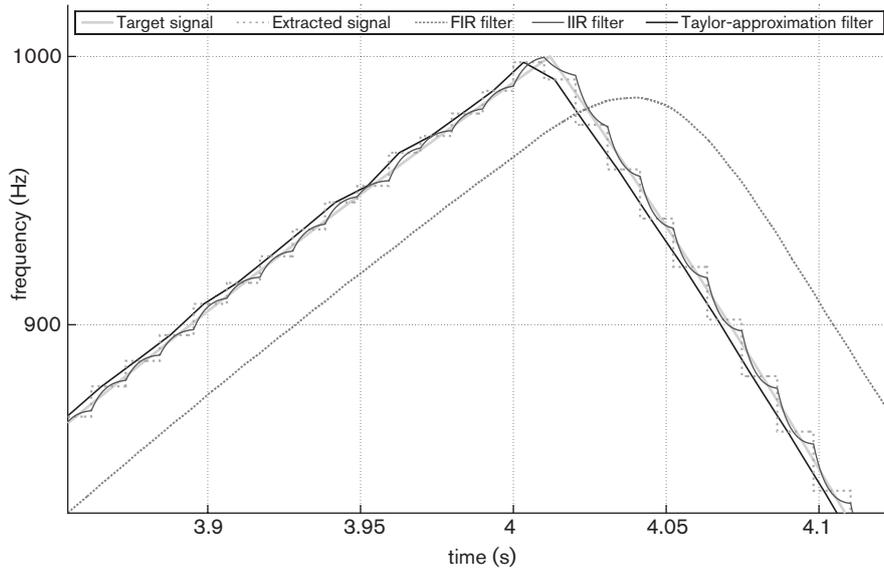


figure 22. Frequency estimation using different decoder filters. 1 s run-up, 0.5 s run-down for a sawtooth shaped driving-cycle. Range: 800–1000 Hz.

4.1.2. Overshoot factor

The overshoot gain factor, as seen in figure 9, is investigated by using different values and looking at the performance of the controller. The primary delay is 4 ms, the secondary path is modelled as a pure delay, and the step size is set at a constant non-optimised value of 0.1, with a frequency range of 150–1000 Hz. In the figures below, one can see the spectrogram of the error gain, for different overshoot gain factors; when the overshoot gain factor is increased, the frequency area of maximum error reduction is lowered—i.e. to achieve a good reduction at lower frequencies, a higher amount of overshoot is needed, compared to when high frequencies disturbances are to be minimised.

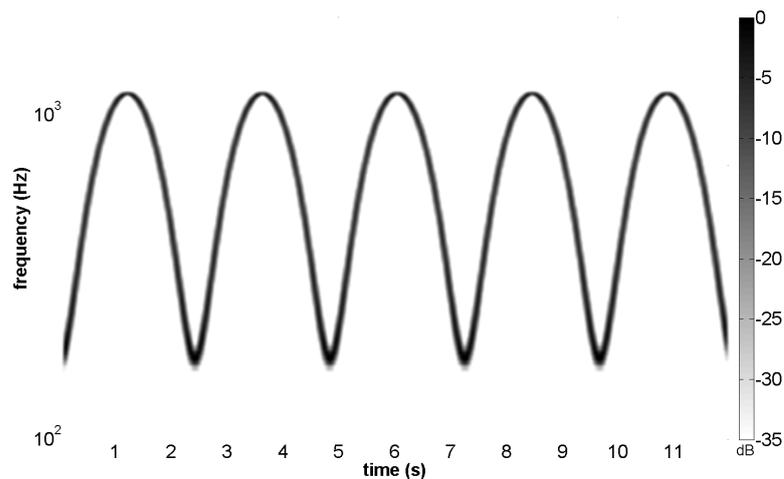


figure 23. Spectrogram of error gain, with disabled controller. Scaling in dB, re 1

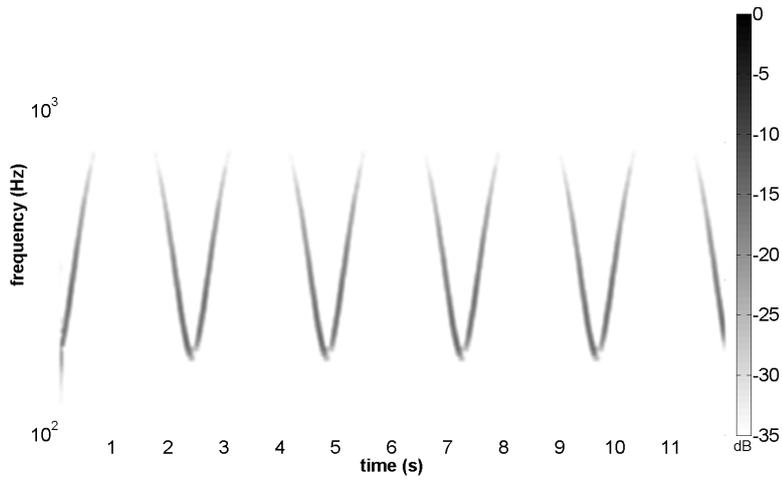


figure 24. Spectrogram of error gain, with a overshoot gain of 0.5%.

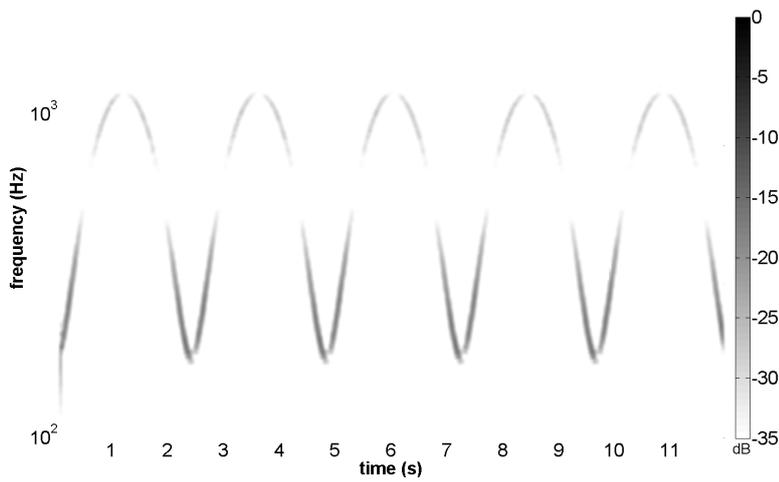


figure 25. Spectrogram of error gain, with a overshoot gain of 0.9%.

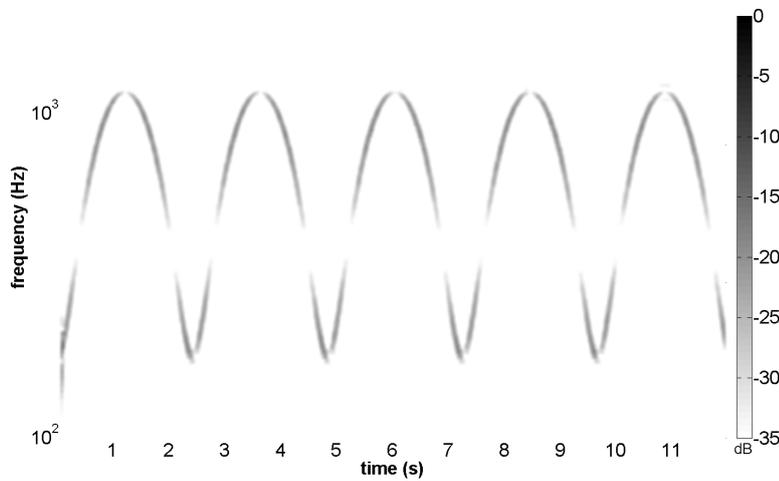


figure 26. Spectrogram of error gain, with a overshoot gain of 1.4%.

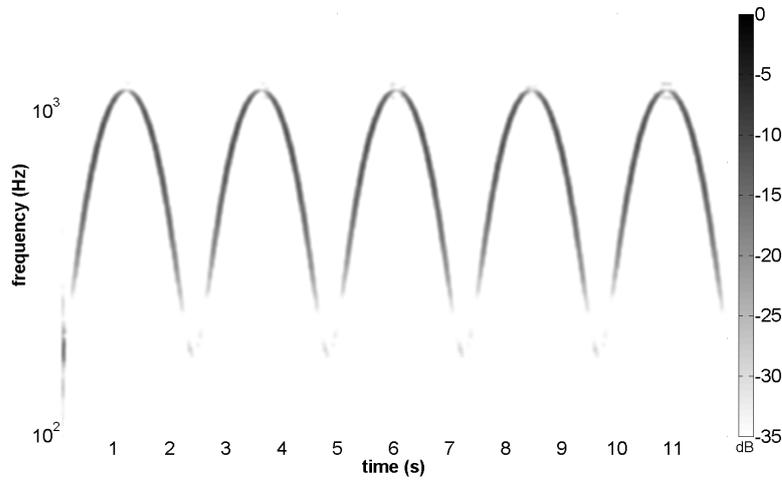


figure 27. Spectrogram of error gain, with a overshoot gain of 2.5%.

4.1.3. Driving-cycle variations

The artificial speed-signal is varied using two basic geometries for run-up and run-down:

- Sinusoidal
- Sawtooth

These shapes include basic linear sections, as well as approximative logarithmic sections. The step size factor is constant at 0.1³¹ and the setup uses artificial path models. In figure 28, the performance of the controller can be seen for different run-up and run-down speeds, for a sinusoidal shaped driving cycle:

- (7, 2.3, 1, 0.5) [s]

In figure 29, the performance of the controller can be seen for different run-up and run-down speeds, for a sawtooth shaped driving cycle:

- (9.3, 5, 2, 1) [s]

It can be seen that the error gain is highly dependant on the rate of change of the engine speed, see the figures below.

31 This value is not optimised, and the absolute performance of the controller is therefore sub-optimal.

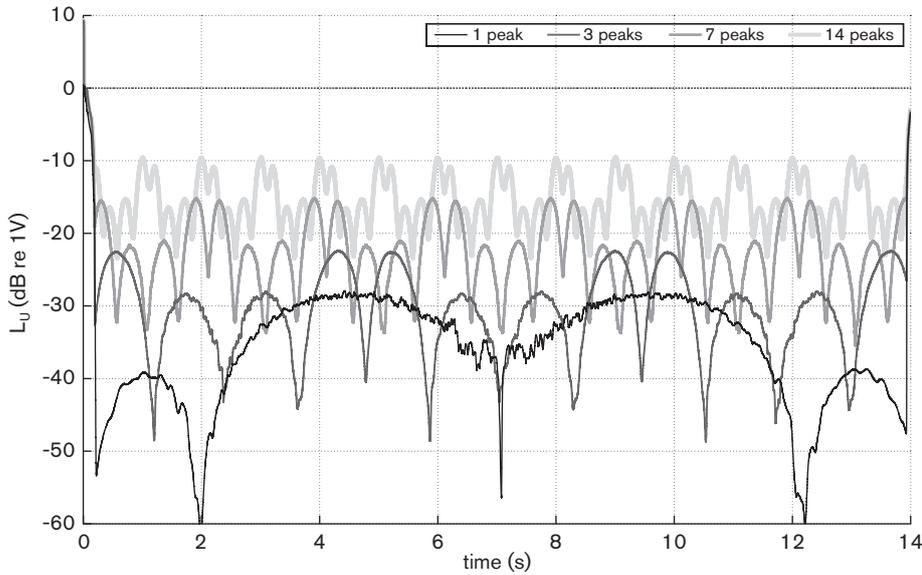


figure 28. Error gain for different run-up times, for a sinusoidal signal. Constant arbitrary step size. Range: 150–600 Hz

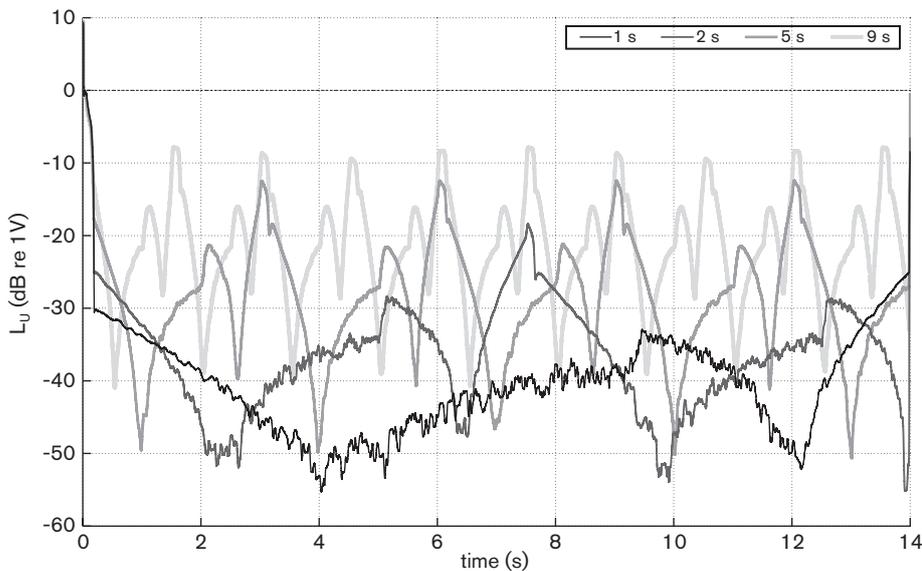


figure 29. Error gain for different run-up times, for a sawtooth signal. Constant arbitrary step size. Range: 150–600 Hz

4.1.4. Step size

The control performance of using step-size normalisation versus the performance of using a constant step-size factor is tested in the virtual setup, using a value of 0.07 for μ and μ_0 . In figure 30, the resulting error gain is plotted for a sinusoidal signal, varying between 300 and 600 Hz in 0.5 s intervals. The setup uses the measured transfer paths from the experimental setup. The global sampling frequency is 51.2 kHz. Other signals show similar differences (not shown).

The results of an assessment of the control performance for different step-size factors can be seen in figure 31. It is seen here with artificial transfer paths for a sinusoidal signal varying between 300 and 600 Hz, with a sampling frequency of 51.2 kHz and a varying overshoot-factor, using the results in 4.1.2. The resulting performance, if too large step-size factors are used, can be seen in figure 32. It should be noted that even though the algorithm is quite unstable, it is kept somewhat stable due to the implemented output limiter³².

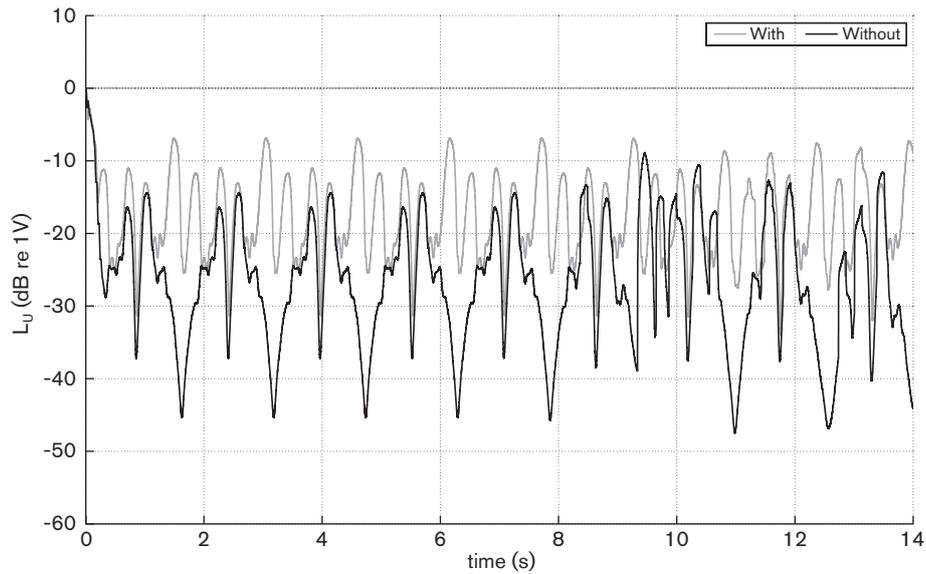


figure 30. Error gain; with, and without, step-size normalisation, for a sinusoidal signal. μ and μ_0 is 0.07. Range: 300–600 Hz

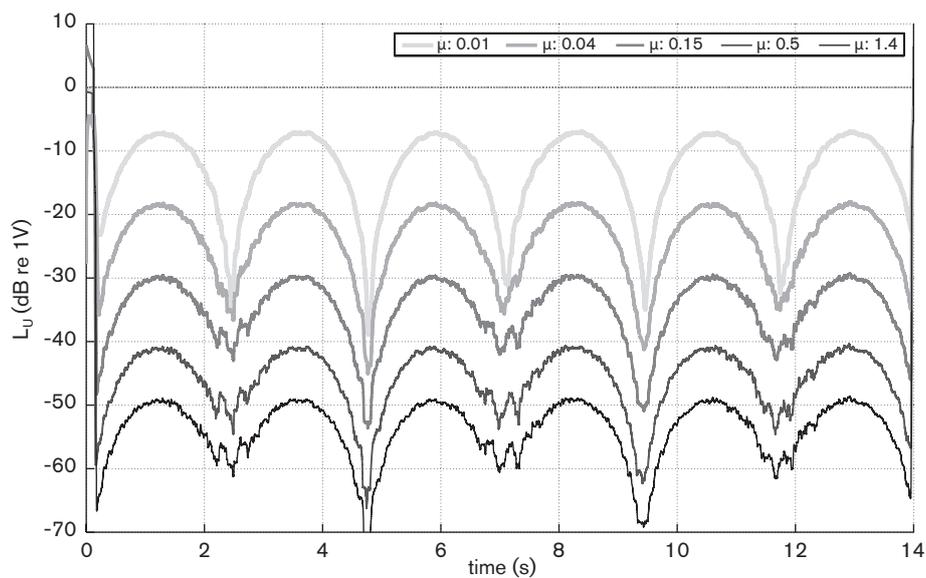


figure 31. Error gain for different step size factors. Sinusoidal signal with a period time of 4.7 s. Range: 300–600 Hz.

32 If the output exceeds the defined amplitude- and phase-thresholds, the algorithm resets itself.

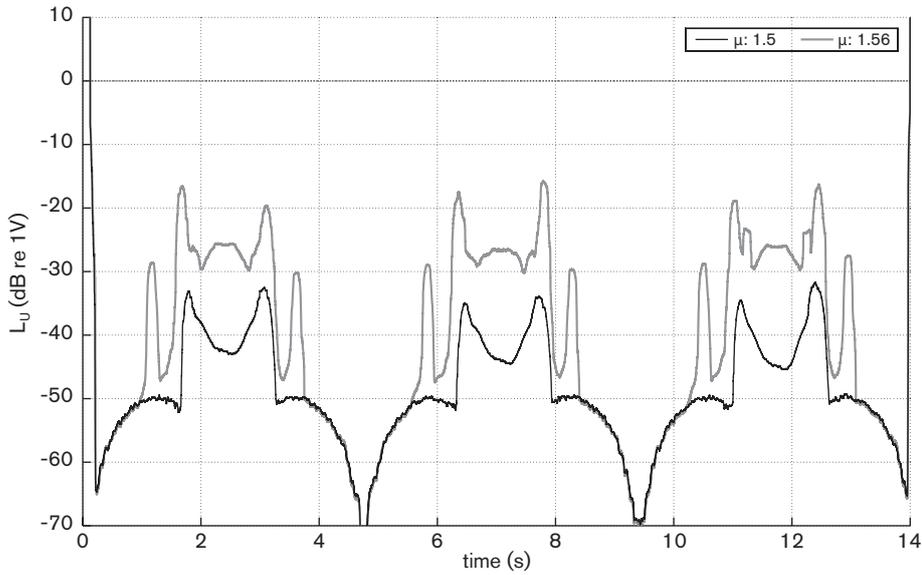


figure 32. Error gain for marginally unstable step size factors. Sinusoidal signal with a period time of 4.7 s. Range: 300–600 Hz.

4.1.5. Sampling frequency

The performance for different global sampling frequencies are shown in figure 33 for a sawtooth driving cycle with a run-up of 2 s, and with a constant step size of 0.1. The tested frequencies are listed in 3.2.3. The performance increases for higher sampling frequencies, however there is very little gain for going above 300 kHz, for this setup.

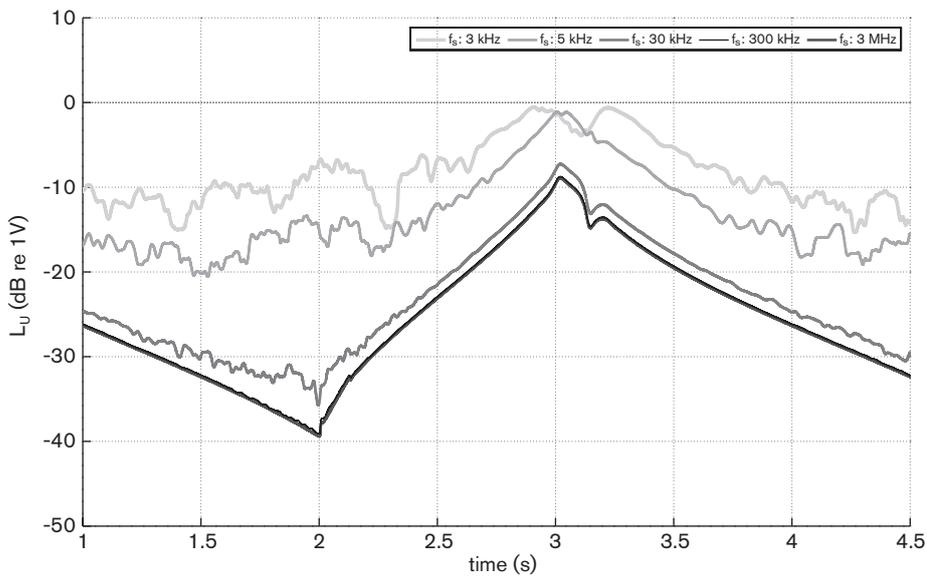


figure 33. Error gain for different global sampling frequencies.

4.1.6. Performance control

Using a target reduction of 20 dB, the controller is evaluated using the measured transfer path responses, for a simple logarithmic run-up of 4 s, and a constant step size—see figure 34. The performance controller can also be used as a sliding on/off function, and thus specifying the working range of the controller itself. The results of such a function can be seen in figure 35.

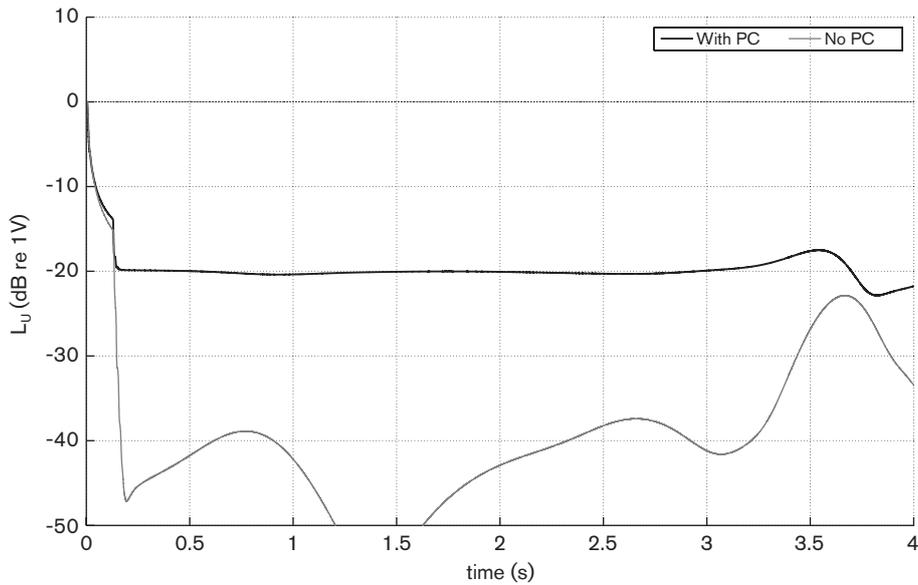


figure 34. Error gain, for performance control with a 20dB reduction target.

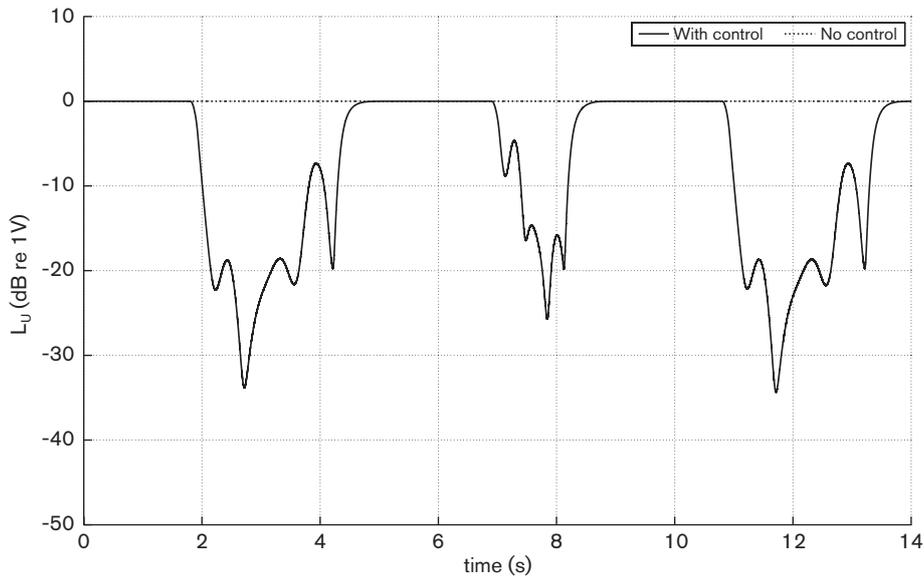


figure 35. Error gain, with a sliding on/off performance control function.

4.1.7. Transfer path

Using no transfer path filtering, the controller cannot maintain stability when passing approx. 300 Hz, as seen in figure 36. A performance control restraint of 20 dB reduction is applied for this simulation, as well as a ~10 dB instability threshold. When the transfer path filter is used, the controller can maintain stable control, as seen in figure 37. In figure 38 the differences in performance when the signal passes the 340 Hz node³³, and when it doesn't, are visible.

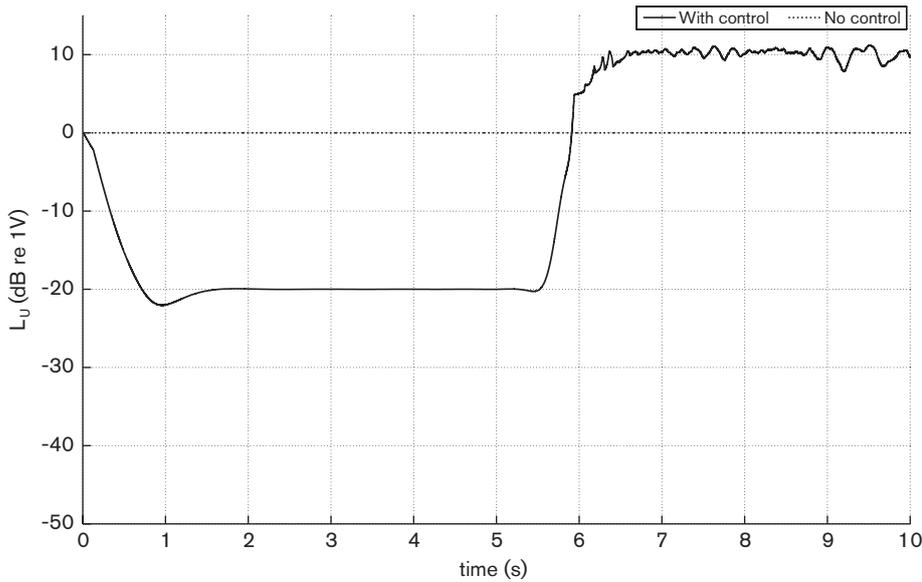


figure 36. Error gain for a 10s run-up with a logarithmic driving-cycle, 150–600 Hz. No transfer path filtering applied.

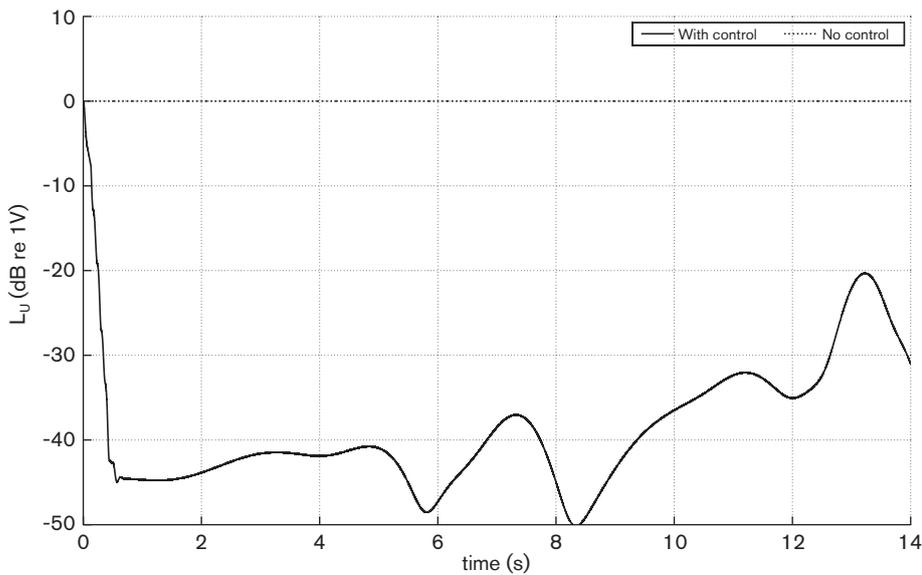


figure 37. Error gain for a 14 s run-up with a logarithmic driving-cycle.

33 Section 3.3.2

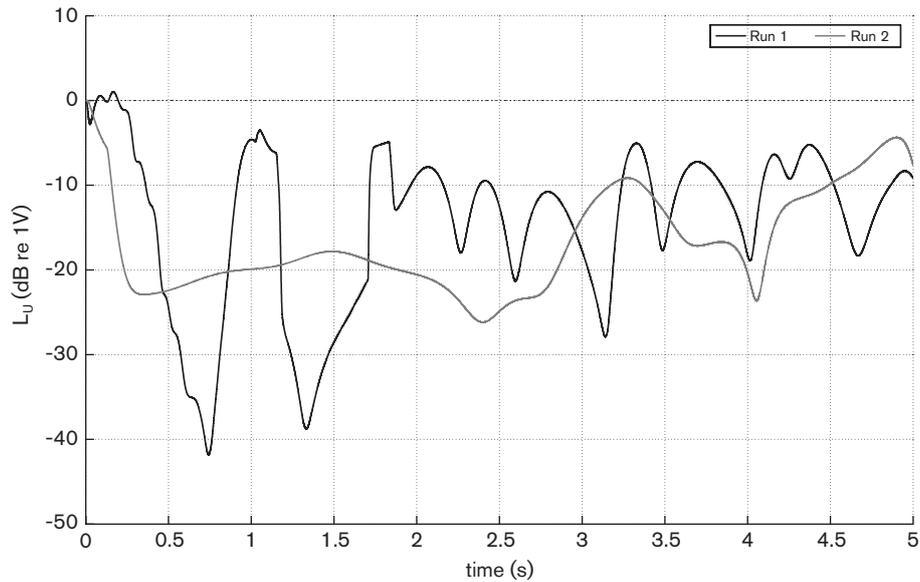


figure 38. Error gain for a 5s run-up with a logarithmic driving-cycle. Run 1 include 340 Hz, run 2 start above 340 Hz, otherwise similar parameters.

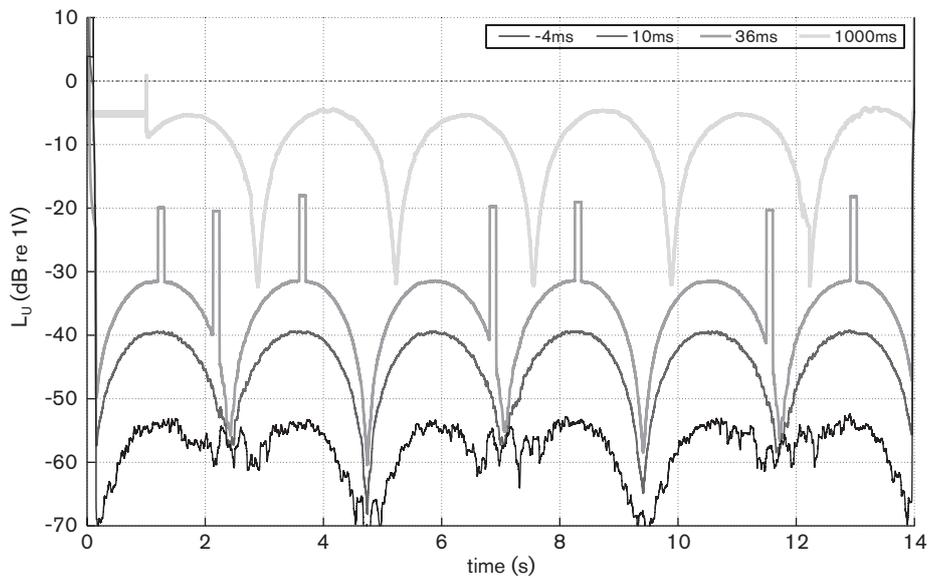


figure 39. Error gain for different deliberate errors in the primary path estimation. Sinusoidal, 300–600 Hz, period time: 4.7s

The controller uses a delay-model for the delay between the tachometer and the disturbance point. In figure 39 the resulting control performance is plotted, when the internal model differs from the real delay (4 ms), for errors between -4 ms and 1000 ms. This simulation uses known artificial primary path and secondary paths, with a step size factor of 1. The controller maintain stability for the tested errors, but the performance is reduced for large errors.

4.2. Experimental results

4.2.1. Without control output normalisation

The performance of the controller is tested in the experimental setup for varying speed signals; the results of some of them can be seen below. The step-size factor is constant, and transfer path filtering is applied, the global sampling frequency is 51.2 kHz. See appendix A.2 for details regarding measurements and instrumentation. No output normalisation filtering is used.

The total error gain can be seen to be dependant on the run-up time. It can also be seen that the error gain is time-dependant, which indicates a frequency dependency, see 3.3 for the unfiltered secondary path response function, and 4.2.2 for the measured error gain when the secondary path is normalised. The error gain, as a function of time, is evaluated using a 125 ms time integration.

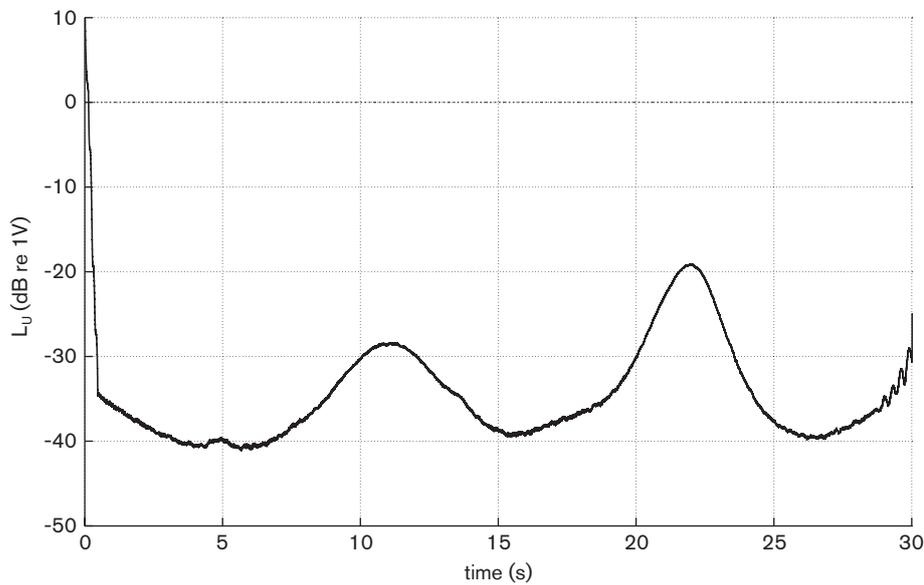


figure 40. Error gain for a 30s run-up with a logarithmic driving-cycle. 100–300 Hz

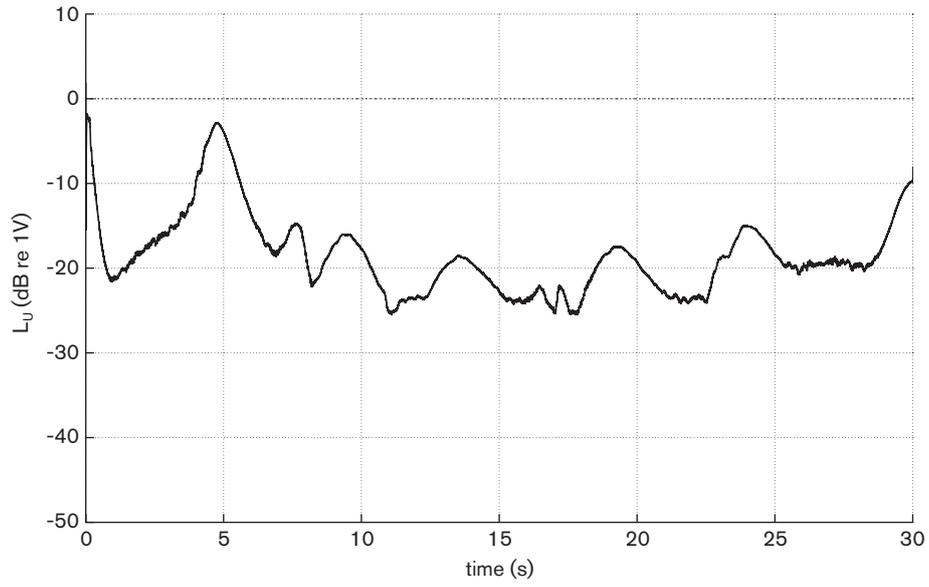


figure 41. Error gain for a 30s run-up with a logarithmic driving-cycle. 300–600 Hz

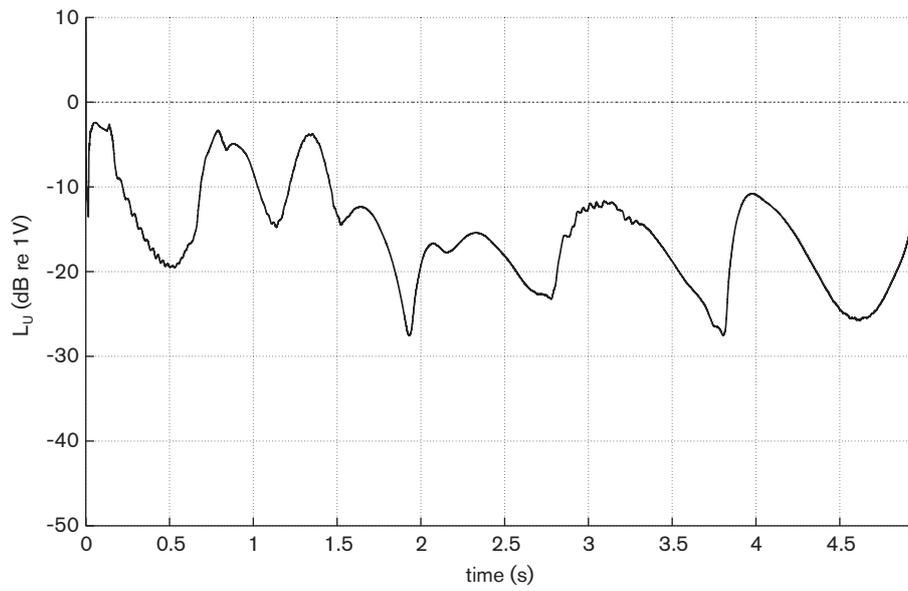


figure 42. Error gain for a 5s run-up with a logarithmic driving-cycle. 300–600 Hz

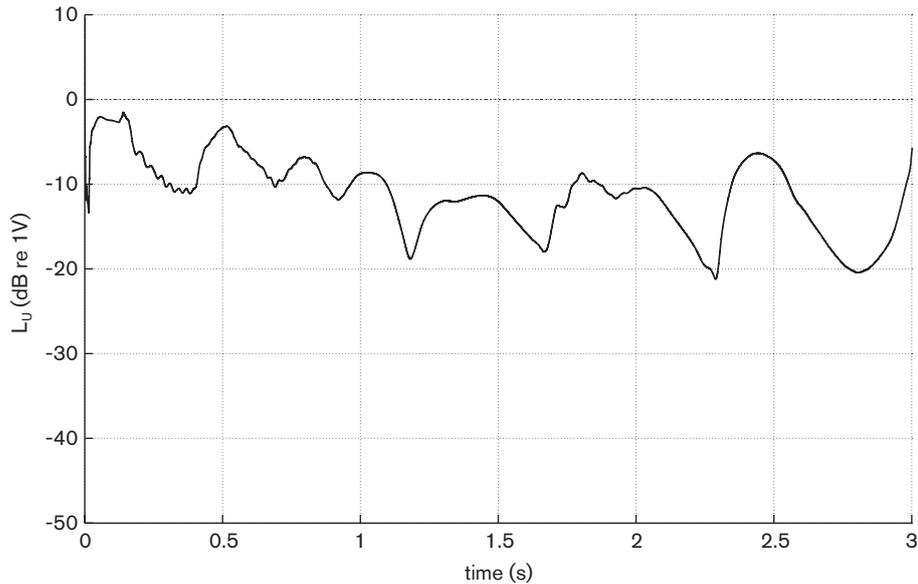


figure 43. Error gain for a 3s run-up with a logarithmic driving-cycle. 300–600 Hz

4.2.2. With control output normalisation

The experimental setup is tested using a control output normalisation filter, as shown in sections 2.2.4 and 3.3.2. The step size factor is constant at 0.02, which is the highest value that gives a stable behaviour for all tested scenarios, in this experimental setup. A custom-designed digital 30th-order autoregressive moving average (ARMA) filter, is used as the output normalisation filter. The global sampling frequency is set to 25.6 kHz, in order to match the measurements used to design the filter, without using resampling. The error gain, as a function of time, is evaluated using a 125 ms time integration.

Below the results of the control performance can be seen for varying run-up time and shapes. Looking at different run-up times, such as 0.5s, see figure 45, and 2s, see figure 46, one can see the same basic behaviour of the controller, but with increasing performance as the run-up time increases. Comparing figure 41 and figure 47, one can note an increase in controller performance, ~20 dB, for the filtered case, even for a slow run-up case, and also notable is the difference in error gain variance—the filtered output controller yield a more steady/consistent error gain. For short run-up/run-down times of 0.125s, there is a quite limited error reduction, ~3 dB; stable control is however maintained, see figure 44.

When a sinusoidal is used as the speed-signal, one can see the same shape in the error gain, e.g. figure 45. This is also seen in the virtual setup, see figure 28. Maximum error reduction is achieved when the amplitude of the derivative of the speed signal is at a minima—as expected.

The error reduction for a steady state 300 Hz tonal disturbance can be seen in figure 48. The 300 Hz tone is reduced by ~110 dB, but some added side-band noise can also be seen. No noise is added to the primary source but the already present environmental noise.

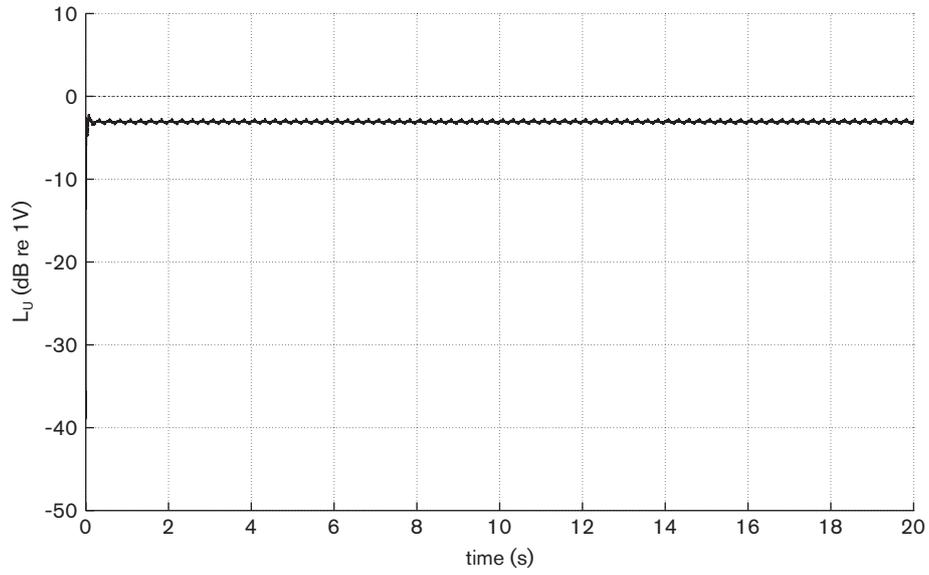


figure 44. Error gain for a 0.125 s run-up/run-down with a sinusoidal driving-cycle. 300–600 Hz

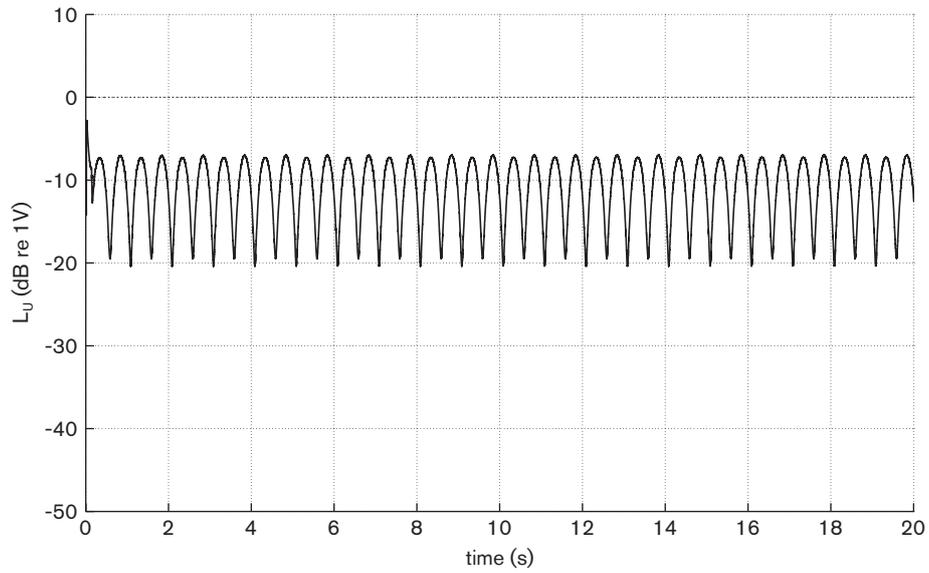


figure 45. Error gain for a 0.5 s run-up/run-down with a sinusoidal driving-cycle. 300–600 Hz

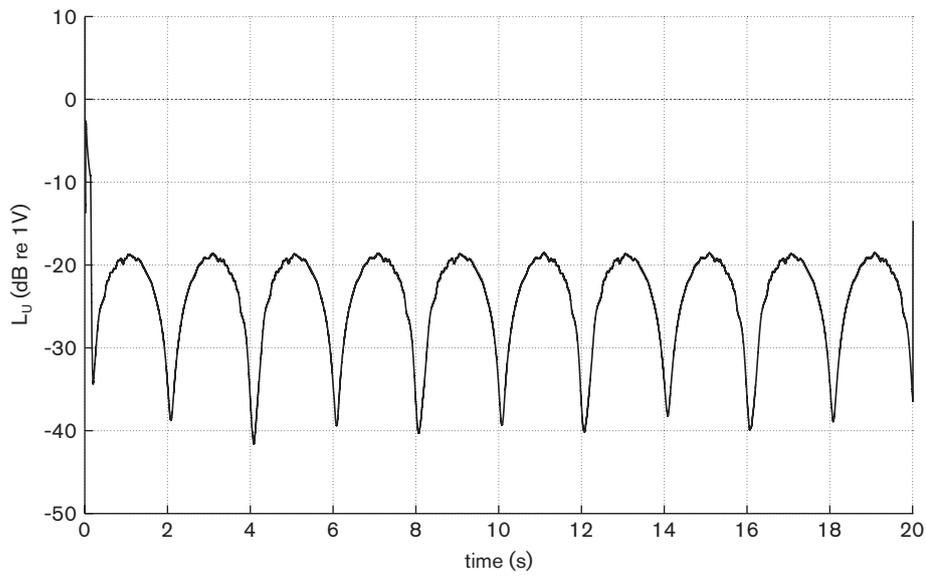


figure 46. Error gain for a 2s run-up/run-down with a sinusoidal driving-cycle. 300–600 Hz

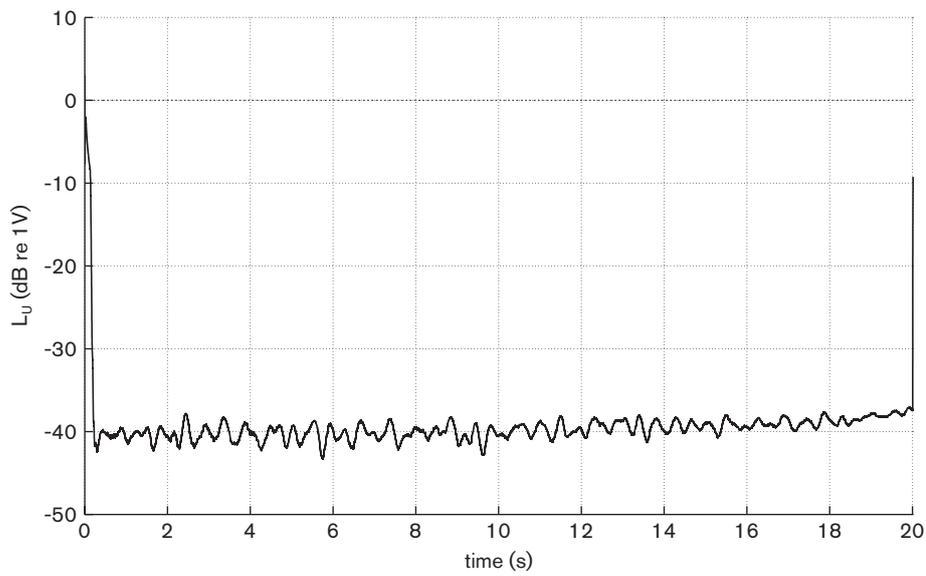


figure 47. Error gain for a 20s run-up with a logarithmic driving-cycle. 300–600 Hz

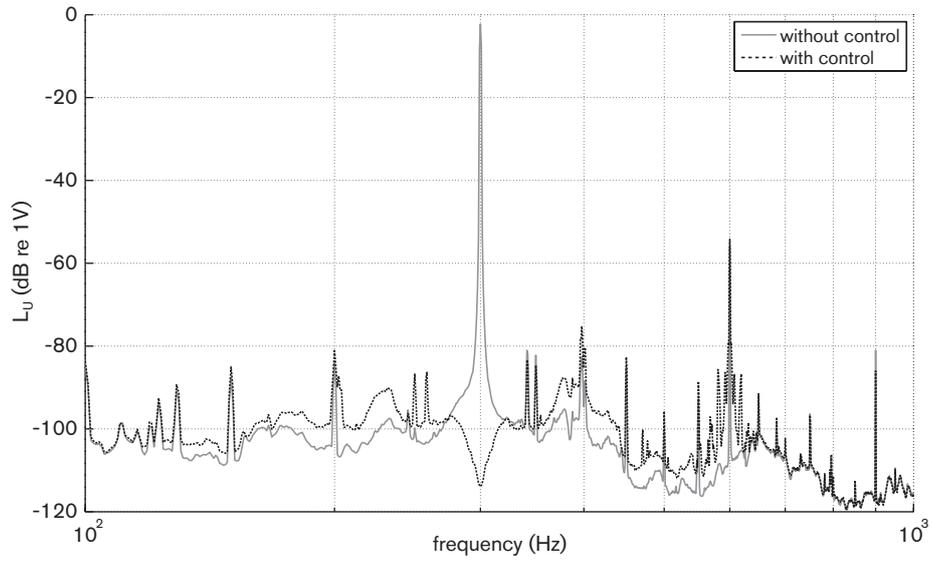


figure 48. Frequency amplitude spectra for a steady state disturbance of 300 Hz.

5. DISCUSSION AND CONCLUSION

5.1. Decoder

The proposed tachosignal decoder provides sufficient performance in most cases, as seen in section 4.1.1. Using different filtering techniques, the Taylor-approximation filter provide the best results for most cases; which is mostly due to its lower delay compared to the other standard low-pass filters tested. As no zero-phase filters can be used, due to the real-time target, an approximate prediction model should be faster, but with an inherent loss in accuracy. To achieve sufficient accuracy, a low-latency low-pass filter³⁴ is applied to the approximated data.

However the default delay is not sufficient for all cases, when optimal performance is needed, as seen in figure 21. It can also be seen that the performance of the decoder is dependant on the driving-cycle. The rate of incoming tacho pulses determine the precision and time-delay of the extracted signal, which is expected, but this effect must be counteracted in order to maintain optimal control. The lower the range, the greater loss in accuracy; but it should also be taken into consideration that the controller achieve better reduction for lower frequencies,³⁵ see figure 40—so this effect could be neglected if, already, the performance target is met.

Using an overshoot factor the decoder can be tuned for a narrow-band range, using a constant gain factor, or a wide-band range, by the use of a adaptive overshoot gain factor. In this case where the range of interest is 300 to 600 Hz, a constant factor, see figure 26, should be sufficient for most cases. When no low-pass filtering is added to the filter stages, the performance decreases, but as they are not optimised for this application, it is uncertain how much the performance can increase.

Future work

The effect of using higher-order Taylor coefficients was not tested, due to the sufficient performance of the simple first-order filter, but in future work it could be interesting to investigate a high-order approximation filter. All filters used are not optimised in its true sense, and should therefore be subject to further investigations. Also the exact behaviour of positive overshooting could be investigated in greater detail, as it is only slightly covered in this thesis. The decoder is also not tested in the real-world application, only in the artificial virtual- and experimental controller.

Overall, the proposed decoder, with its internal taylor-approximation filter, auxiliary low-pass filters and overshooting function, provide good performance for the application under study. This thesis has shown that the performance of the controller is dependant on the accuracy and speed of the decoder, and care must therefore be taken when such a decoder is to be implemented.

34 1:st order Butterworth filter, with a cut-off frequency of 1 kHz.

35 Around 1000 rpm of the engine.

5.2. Controller parameters

5.2.1. Sampling frequency

In figure 33 the differences in performance for different sampling frequencies for a single driving cycle, can be seen. The overall performance pattern is consistent for all driving cycles tested, not shown. It matches the expectations, as there is for every minimisable cost-function an optimal iteration number. However for very high sampling frequencies, >300 kHz, the performance increase start to plateau. whether it is due to simulation limitations, or limitations in the control system, i.e. there is only so much reduction achievable for certain errors in the decoding- and transfer path estimations. The overall pattern suggest that a high as possible sampling frequency should be used, if it meets the other cost-, hardware- and software limitations, for a certain application.

Future work

The behaviour at very high sampling frequencies should be further investigated, along with a more thorough parametric study, in order to get a better understanding of this plateau-effect,

A high global sampling frequency provides a performance gain. The relationship is not linear, and should therefore be considered in cost-optimisation cases. For the application under study, there is little gain in going above 30 kHz; the global sampling frequency mostly used in this study is 51.2 kHz and 25.6 kHz, for simplicity reasons³⁶.

5.2.2. Experimental intake-system and control output normalisation

The limits on what is possible in active control is depending on the control law, hardware/software, processing power/speed, and the physical- and casual limitations of the system itself. For this simple single-channel active noise control system in a duct, it is possible to achieve a good overall attenuation of the harmonic disturbances, as shown in the virtual setup as well in the experimental setup; however there seems to be a problem regarding stability and control performance in the range of 300–400 Hz. As seen in figure 12 and figure 14, there is an anti-resonance present in this range, and it is due to the position of the error microphone, as this behaviour is present in both the primary path and the secondary path. This makes it hard to maintain control in this frequency range.

By applying a small amount of damping (plastic foam) and using a normalisation filter at the control output stage, that problem is removed. This also increases the performance of the controller, while making it more stable and steady, see figure 44 and figure 47 for example. As shown in section 2.2.4, *optimal* control can be achieved using a simple constant step-size factor, if the secondary path is normalised.

Future work

The behaviour of the experimental setup should be further investigated, to show how the control performance can be optimised. The filter used as control output normalisation filter should be subject to further studies.

³⁶ It matches the sampling frequency of measurement system used to measure the transfer path function, and their impulse responses. It should be noted, that they are the same only to avoid re-sampling.

For the application under study there is a resonance problem in the experimental intake-system, giving rise to a stability problem—the stability can be maintained, but comes with the price of low convergence speed, and lower control performance. I suggest that a control output normalisation filter is used, as it improves stability, and control performance with little added cost.

5.2.3. Step size

As seen in equation 11, and in figure 31, the step-size factor, μ , is important for the adaptive algorithm, and therefore for the performance of the controller. The figure shows the performance for a single case, but the relative performance is valid for all tested driving-cycles—not shown. It can be seen that the largest possible step-size yields the best results, however for larger values, see figure 32, the algorithm becomes marginally unstable—for even larger step-size the controller becomes completely unstable. This behaviour is expected, as there is for any controllable system an optimally large step-size³⁷, which is the largest step-size that fulfils equation 12.

The results from using the step-size normalisation shows a degraded performance compared to when a constant step-size is used, as seen in figure 30. This behaviour is present for other signals, as well as for different combinations of μ and μ_0 . Why this is the case is not covered in this thesis, but a possible explanation could be the resonant behaviour of the secondary path; the anti-resonances of the system would give rise to very large momentary step-size. When a control output normalisation filter is used, there is an increase in performance. It also functions as a step-size normalisation, thus making the optimal step-size factor, constant, and therefore easier to determine.

As shown the range of possible step-sizes is a function of the system, and must therefore be optimised for any given application and system.

Future work

Variants of the NLMS-algorithm, such as the one proposed by Choi et al³⁸, were not considered, but should be investigated in future studies. Furthermore the effect of the experimental setup used should be investigated. The stability differences between the standard Fx-LMS and the proposed normalised control output LMS, Fy-LMS, should be investigated.

The step-size is important for the achievable performance of the controller, both in terms of convergence and stability. For a simple case, a constant step-size is sufficient, when used together with a high iteration frequency, i.e. global sampling frequency, and especially with a normalisation control output filter.

5.3. Complete application

As seen in section 4.1.3, the performance of the controller is closely related to the driving cycle, but for all tested variations, the controller can maintain control and stability. The study uses artificial driving cycles, but as the variation in run-up times and shapes of the cycles are quite varied it should be applicable to real-world scenarios. The controller is not *aware* of any non-causal information, and the performance, at the peaks and dips of the signal, should therefore be closely related to the real-worlds varying non-periodic signals. It can be seen

37 Elliott, S. (2001) *Signal processing for active control*. London: Academic Press

38 Choi, Y-S et al. (2006) Robust Regularization for Normalized LMS Algorithms. *IEEE Transactions on circuits and systems – II: Express briefs*, vol.53, no.8, pp. 627-631

that the performance is increased for smoothly changing engine-speeds, and reflect both the performance of the controller, and the decoder. The controller can maintain stable control for fast changing engine speeds, as fast as a run-up, and run-down, time of 0.125s. Which should be sufficient for most cases, however with a limited error reduction of only ~3 dB, see figure 44. When faced with a steady state engine speed, the reduction achieved is as large as ~110 dB, for the studied experimental setup—tested using a 300 Hz tonal disturbance. See figure 48. However such a great reduction is highly dependant on the SNR of the present ANC system. It is also notable that the residual 300 Hz tone is lower than the *noise floor*.

The overall performance is highly dependant on the experimental setup system, and its problematic frequency range; the proposed controller displays weak performance, when faced with a problematic path response, as discussed in 5.2.2. This is visible in both the virtual setup and the experimental setup, showing that a simple cost-effective virtual implementation could be used to investigate such problems. The approximation filter in the decoder seems to be not causing any stability problems, as the time-window of the prediction is quite small—around one engine revolution, as seen in figure 21.

By applying a range limiter, i.e. a sliding on-off performance control function, a specific range of frequencies can be selected. This can be done to achieve a desired error signal, as well as maintain stability of the controller at problematic frequencies. By applying the internal secondary path model, as shown in section 2.2.3, the controller can *slide* into action, with minimal convergence time, see figure 35. Such a function provide a good tool for sound quality control, as discussed by Olivier et al³⁹. This function combined with a optimised overshoot gain function, make is possible to optimise the performance of the controller for the range of interest, without any concerns for control instability outside this range.

It should be added that the controller maintain stability, when random noise are added to the system, not shown. Adding more harmonic tones to this noise does not effect the controller, thus making it possible to stack the number of needed controllers to control many harmonic disturbances—which is tested with maintained results.

Future work

The proposed controller, with its supporting functions, should be further tested using the real-world signals and setup, to validate its performance and behaviour. The adaptive LMS-algorithm used is also not tested against other similar algorithms. It was chosen due to its simplicity and well-documented history. The studied control algorithm achieve the target reduction in the range of interest, but should greater attenuation be required, further investigation and optimisation must be done.

Overall the proposed adaptive harmonic controller achieve the target reduction of 10 dB for most tested scenarios and driving cycles, as validated in a virtual test setup and a experimental setup. By using a performance limiter, the attenuation, and the frequency range, can be tailored for specific applications and needs. The parameters within the controller and decoder, must however be optimised for any given implementation. This thesis has shown that those parameters can be quite easily assessed, for instance with a virtual setup using measured data from the desired system. The performance could be further optimised, and the controller's working range extended, if needed; and the controller can be part of a modular multiple control system, where many harmonic disturbances are to be controlled. By using a control output normalisation filter, the performance and stability of the controller is increased.

39 Oliveira, Leopoldo P.R. de, et al. (2010) NEX-LMS: A novel adaptive control scheme for harmonic sound quality control. *Mechanical Systems and Signal Processing*, vol. 24, no. 6, pp. 1727–1738.

6. REFERENCES

- Choi, Y-S et al. (2006) Robust Regularization for Normalized LMS Algorithms. *IEEE Transactions on circuits and systems—II: Express briefs*, vol.53, no.8, pp. 627–631
- Elliott, S.J. (2001) *Signal processing for active control*. London: Academic Press
- Elliott, S.J. (1998) Filtered reference and filtered error LMS algorithms for adaptive feedforward control. *Mechanical Systems and Signal Processing*, vol. 12, no. 6, pp. 769–781.
- Fedoryuk, M.V. (2001) *Encyclopedia of Mathematics*. Springer
- Johansson, S. et al. (2000) *A Novel Multiple-Reference Algorithm for Active Control of Propeller-Induced Noise in Aircraft Cabins*. Karlskrona: Blekinge Institute of Technology (Research report no. 16/00, Department of Telecommunications and Signal Processing)
- Larsson, Erik.G. (2009) Enhanced-Convergence Normalized LMS Algorithm. *IEEE Signal Processing Magazine*, vol. 26, no. 3, pp. 81–82, 95.
- Lyons, Richard.G. (2010) *Understanding Digital Signal Processing, 3:rd Edition*, New Jersey: Prentice Hall
- Mitra, S.K. (2001) *Digital Signal Processing*, McGraw-Hill
- Morgan, D.R. (1980) An analysis of multiple correlation cancellation loops with a filter in the auxiliary path. *IEEE Transactions on acoustics, speech, and signal processing*, vol. 28, no. 4, pp. 454–467.
- Morrison, Margaret. (2009) Models, measurement and computer simulation: the changing face of experimentation. *Philosophical Studies*, vol. 143, no. 1, pp. 33-57.
- Nelson, P.A, Elliott, S.J. (1992) *Active Control of Sound*. London: Academic Press
- Oliveira, Leopoldo P.R. de, et al. (2010) NEX-LMS: A novel adaptive control scheme for harmonic sound quality control. *Mechanical Systems and Signal Processing*, vol. 24, no. 6, pp. 1727–1738.
- Sjösten, Per. (2003) *Active Noise Control of Enclosed Sound Fields, Optimising the performance*. Gothenburg: Chalmers University of Technology. (PhD Thesis, Division of Applied Acoustics)

A. MEASUREMENTS AND INSTRUMENTATIONS

A.1. Transfer path measurements and signal processing

A.1.1. Instrumentation

The transfer path is measured using the following instruments:

- VXI Data acquisition system E8408A
- Larson Davis 2520 microphone, serial no. 1112
- Brüel & Kjær 2633 microphone cable, serial no. 799474
- Brüel & Kjær 2803 microphone power supply, serial no. 192743
- Brüel & Kjær 2807 microphone power supply, serial no. 427912
- Larson Davis 2200C microphone preamplifier, serial no.0789
- Stageline MPA-102 microphone preamplifier, serial no. W06/001091-02
- Primary source: SEAS H1207-08, 4.5" with sealed box.
- Secondary source: unknown 3.5" driver mounted in a closed box.
- Amplifier: NAD 310, serial no. M783102951
- Software: Trigger Happy 4.0 using Matlab 5.3

A.1.2. Measurement

The transfer path functions is measured by sending white noise in paralell to the source under study, and to the input to the VXI acquisition system. The response at the microphone is amplified by a pre-amplifier and connected to the second input of the VXI acquisition system. It is controlled by the Trigger happy software, which uses a matlab script to calculate the necessary transfer functions and corresponding coherence function. The coherence function is used to validate the significance of the measurement.

A.2. Experimental setup

A.2.1. Instrumentation

The performance of the controller is measured using the following instruments:

- Dspace DS1103 DSP, with I/O panel
- VXI Data acquisition system E8408A
- Larson Davis 2520 microphone, serial no. 1112
- Brüel & Kjær 2633 microphone cable, serial no. 799474
- Brüel & Kjær 2803 microphone power supply, serial no. 192743
- Brüel & Kjær 2807 microphone power supply, serial no. 427912
- Larson Davis 2200C microphone preamplifier, serial no.0789
- Stageline MPA-102 microphone preamplifier, serial no. W06/001091-02
- Primary source: SEAS H1207-08, 4.5" with sealed box.
- Secondary source: unknown 3.5" driver mounted in a semi-closed box.
- Amplifier: NAD 310, serial no. M783102951
- Software: Trigger Happy 4.0 using Matlab 5.3
- Matlab/Simulink R2010a
- Software: HP Daq Express

A.2.2. Measurement

The error gain of the controller in the experimental setup is measured by capturing the full time signal, using HP Daq Express. The error microphone is positioned 15 cm from the edge of the tube, and 34 cm from the secondary loudspeaker, which is mounted to the side of the tube. All gain settings are set to achieve a good SNR without overload/clipping.

