

UNIVERSITY OF EDINBURGH

# A comparison of virtual analogue Moog VCF models

by

Paul Daly (s1143524)

Final Research Project: Dissertation submitted in partial fulfillment for the  
MSc in Acoustics and Music Technology

in the

College of Humanities and Social Sciences  
School of Arts, Culture and Environment

August 2012

UNIVERSITY OF EDINBURGH

## *Abstract*

College of Humanities and Social Sciences  
School of Arts, Culture and Environment

MSc in Acoustics and Music Technology

by Paul Daly (s1143524)

This report shows how to implement Antti Huovilainen's digital model of the Moog voltage controlled filter. A new model of the Moog is built using an implicit finite difference scheme and Newton-Raphson's method. The models are compared in terms of accuracy, stability and computational expense. It was found that it is possible to create a more accurate version of Huovilainen's model at the cost of stability and computational expense. This report also contains a considerable literature review on the Moog filter which consolidates many sources into one body.

# *Acknowledgements*

The author would like to thank and acknowledge the help of Dr. Stefan Bilbao, who supervised and guided the progress of this dissertation, and the rest of the Acoustics and Fluid Mechanics research group at the University of Edinburgh.

# Contents

<b>Abstract</b>	<b>i</b>
<b>Acknowledgements</b>	<b>ii</b>
<b>List of Figures</b>	<b>v</b>
<b>Abbreviations</b>	<b>vi</b>
<b>Symbols</b>	<b>vii</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Literature review and techniques</b>	<b>3</b>
2.1 The Moog transistor ladder . . . . .	3
2.2 Smith and Stilson’s analysis . . . . .	5
2.3 Huovilainen’s implementation . . . . .	7
2.3.1 Other implementations . . . . .	10
2.4 The bilinear transform . . . . .	11
2.5 Discrete operators . . . . .	11
2.6 Iterative methods . . . . .	13
<b>3 Implementation</b>	<b>16</b>
3.1 Linear models of the Moog VCF . . . . .	16
3.2 Non-linear models of the Moog VCF . . . . .	19
<b>4 Results</b>	<b>22</b>
4.1 Accuracy . . . . .	22
4.2 Stability . . . . .	31
4.3 Computational expense . . . . .	33
4.4 Audio examples . . . . .	34
<b>5 Discussion and conclusions</b>	<b>35</b>
5.1 Accuracy . . . . .	35
5.2 Stability . . . . .	36
5.3 Computational expense . . . . .	37

---

<b>6 Recommendations</b>	<b>38</b>
6.1 Recommendations . . . . .	38
<b>A Tuning Huovilainen's Moog VCF filter</b>	<b>39</b>

# List of Figures

2.1	Schematic of the Moog filter ladder . . . . .	6
2.2	Frequency response of analogue Moog VCF . . . . .	8
2.3	Analogue pole positions of Moog VCF for increasing $k$ . . . . .	9
4.1	The magnitude response of Huovilainen's unit-delay model . . . . .	23
4.2	The magnitude response of Huovilainen's unit-and-a-half delay model . . . . .	24
4.3	The magnitude response of Huovilainen's unit-and-a-half delay model with tuning . . . . .	25
4.4	The magnitude response of the backwards difference model . . . . .	26
4.5	The magnitude response of the centred, backwards difference model . . . . .	27
4.6	The frequency response of Huovilainen's model and the centred, back- wards difference model for high frequencies . . . . .	28
4.7	Harmonic distortion . . . . .	29
4.8	The absolute magnitude error in the magnitude response for low frequencies	30
4.9	The absolute magnitude error in the magnitude response for high frequencies	31

# Abbreviations

<b>dBFS</b>	<b>F</b> ull <b>S</b> cale <b>D</b> ecibels
<b>FFT</b>	<b>F</b> ast <b>F</b> ourier <b>T</b> ransform
<b>NR</b>	<b>N</b> ewton- <b>R</b> aphson
<b>VCF</b>	<b>V</b> oltage <b>C</b> ontrolled <b>F</b> ilter

# Symbols

$F_c$	Filter cut-off frequency	Hz
$k$	Feedback co-efficient for the filter core	dimensionless units
$r$	Feedback co-efficient for a single filter stage	dimensionless units
$y^n$	Sample number $n$ of a general signal $y$	dimensionless units
$\omega_c$	Filter cut-off frequency	rads <sup>-1</sup>



# Chapter 1

## Introduction

In 1965 Dr. Robert Moog presented his paper on a voltage controlled low pass filter that was the first of its kind and would go on to become one of the most acclaimed and influential tools in electronic music.[1] [2] Of course, it was only a matter of time before a digital model of the filter was sought out. In 1996 Smith and Stilson analysed the Moog VCF's circuitry and offered several methods of transforming it to the discrete time domain.[3] Musicians argue that digital filters lack the characteristics of their analogue counterparts, in particular the “warmth” that is attributed to their handling of large scale signals however, it is possible to make digital filters sound “warmer” by embedding a non-linear function within the filter.[4]

In 2004, Antti Huovilainen analysed the Moog VCF with the intent of capturing the inherent non-linearities present in the circuitry and implementing an accurate, non-linear digital model of the filter. The model does not need tuning by ear as it models the circuit directly and the “warmth” of the filter is dependent upon the level of the input signal. His model is built around calculating the solution to four differential equations using a forward difference method in such a way that it can be implemented in real time.[5] The purpose of this project was to implement a model of the Moog VCF that was somehow better than Huovilainen's model, whether it be more accurate (without resorting to many times oversampling) or more stable or with less computational expense.

In this paper a linear version of Huovilainen's model is implemented and its ability to approximate the analogue Moog VCF's frequency response is tested. A centred, backwards difference system is created to solve the four differential system equations simultaneously to create a more accurate model of the Moog VCF. The filter is tested to find that it is not as stable as Huovilainen's and is considerably more expensive. This paper also includes a considerable literature review on the Moog VCF that consolidates

---

articles from Julius Smith and Tim Stilson, Antti Huovilainen, Thomas Hèlie, Federico Fontana, Timothy Stinchcombe and from Robert Moog himself.[\[3\]](#)[\[5\]](#)[\[6\]](#)[\[7\]](#)[\[8\]](#)[\[1\]](#)

Audio examples, MATLAB code and a PDF version of this paper can be found on the accompanying data CD.

## Chapter 2

# Literature review and techniques

### 2.1 The Moog transistor ladder

Extensive analysis of the Moog VCF's transistor ladder has been done, notably by Huovilainen and Stinchcombe, and Robert Moog's original paper and the Moog operation manual remain valuable resource of information.[5][8][1][9] The following literature review contains the information needed to understand the workings of the ladder and implement an accurate digital model. Note that throughout this document the symbols  $f$  and  $\omega$  will both represent frequency: the former is measured in Hertz while the latter in radians per second. Also, "phase" may be measured in the informal units of "degrees" when it aids clarity.

The core of the Moog VCF is a driver; a transistor "ladder" of four identical, buffered stages; and a variable gain feedback path. Each stage consists of a differential pair of NPN-transistors with a capacitor placed in between them. The base-to-emitter resistance of the transistors are used to create four one-pole, voltage controlled, low-pass RC filters. The output current of each stage drives the next; the first stage is driven by a differential transistor pair which draws its current from a control current  $I_c$ . An input signal  $I_s$  enters the core at the bottom of the ladder, passes through each stage and leaves the filter at the top. The output is split between a high impedance, differential amplifier and an inverting feedback path which leads back to the first stage.[5] See Fig. 2.1 for reference. If (and only if) there is no feedback, each stage causes 3 dB of attenuation at the cut-off frequency and the complete core causes 12 dB of attenuation with 24 dB/octave roll-off. The magnitude response of the filter varies considerably as the feedback level is increased. An idiosyncrasy of the Moog is its uncoupled control of the cut-off frequency and its Q-value which are both adjustable by separate dials.[9]

It is possible to derive the differential current in a transistor pair. The current gain of each transistor is great enough that it is reasonable to assume that the base current is zero and that the Early effect can be ignored.[10] Each filter stage is dependent only upon its current state and the current from the preceding stage. Therefore only one isolated filter stage needs to be inspected from which the effects of the complete ladder can be extrapolated.

$V_1$  and  $V_2$  denote the base-emitter voltages of the transistors while  $I_1$  and  $I_2$  represent the collector current where the sub-scripts “1” and “2” denote the left and right hand transistors with respect to Fig. 2.1. As we have assumed that no base current flows, the collector current and emitter current are equal. Notice also the atypical orientation of the four upper differential transistors pairs which is a characteristic of the Moog filter; the bottom transistor driver pair is arranged in the typical fashion of a differential pair.[11]

The ladder output is differential so we aim to find the relationship between  $V_1$ ,  $V_2$  and  $I_1 - I_2$ . Assuming that: the transistors are perfectly matched; transistor gain is great enough to be considered infinite; and that the Early effect is ignored, the current difference in a differential transistor pair<sup>1</sup> is

$$I_1 - I_2 = (I_1 + I_2) \tanh\left(\frac{V_1 - V_2}{2V_t}\right) \quad (2.1)$$

where  $V_t \approx 28.5$  mV is the thermal voltage of a transistor.[12] [13] The two currents  $I_1$  and  $I_2$  sum and subtract such that

$$I_1 + I_2 = I_{ctl} \quad (2.2)$$

$$I_1 - I_2 = I_s - 2I_c \quad (2.3)$$

where  $I_{ctl}$  is the ladder control current,  $I_s$  is the signal current (that is, the input audio) and  $I_c$  is the current through the capacitor. Using the equations above we can establish a new relation

$$2I_c = I_s - I_{ctl} \tanh\left(\frac{V_c}{2V_t}\right). \quad (2.4)$$

Now if we replace  $I_c$  above with the terms from the capacitance equation

$$I_c = C \frac{dV_c}{dt} \quad (2.5)$$

---

<sup>1</sup>This is a standard equation for a differential transistor pair but for a very clear derivation of Eqn. 2.1 with respect to the Moog see [8].

for a capacitor of value  $C$ , then

$$2C \frac{dV_c}{dt} = I_s - I_{ctl} \tanh\left(\frac{V_c}{2V_t}\right). \quad (2.6)$$

However, as each stage is driven by the preceding one, it is possible to replace signal current  $I_s$  to create a general relation

$$\frac{dV_c}{dt} = \frac{I_{ctl}}{2C} \left( \tanh\left(\frac{V_{in}}{2V_t}\right) - \tanh\left(\frac{V_c}{2V_t}\right) \right) \quad (2.7)$$

where  $V_{in}$  is the potential difference across the capacitor (or driver) from the previous stage. For the sake of the brevity, a dimensionless version of the above equation which will be used when implementing the digital model of the Moog will be included here:

$$\frac{dv_i}{dt} = \omega_c (\tanh v_{i-1} - \tanh v_i) \quad (2.8)$$

$$v_0 = v_s - kv_4 \quad (2.9)$$

where  $v_s$  is the input audio signal;  $i \in 1, 2, 3, 4$  labels the stage of the ladder;  $k = [0, 4]$  is the feedback gain and  $\omega_c$  is the filter cut-off frequency in  $\text{rads}^{-1}$ .

## 2.2 Smith and Stilson's analysis

In 1996 while at CCRMA, Julius Smith and Tim Stilson analysed the Moog VCF and explored the best methods for converting it to a discrete time domain.[3] Below is a concise iteration of their work that concerns the frequency response of the Moog VCF.

The transfer function  $G_1(s)$  of a single stage has the form we would expect from a one-pole low pass filter[14]

$$G_1(s) = \frac{1}{1 + s/\omega_c}. \quad (2.10)$$

Each filter stage has a real valued pole when  $s = -\omega_c$ . The cut-off frequency  $\omega_c$  sets the location of the -12 dB point on the green line in Fig. 2.2. The transfer function of the complete core with feedback is

$$H(s) = \frac{Y(s)}{X(s)} = \frac{G_1(s)^4}{1 + kG_1(s)^4} = \frac{1}{k + \left(1 + \frac{s}{\omega_c}\right)^4} \quad (2.11)$$

where  $k \in [0, 4]$  is the feedback gain. Consider the frequency response

$$H(j\omega) = \frac{1}{k + \left(1 + \frac{j\omega}{\omega_c}\right)^4}. \quad (2.12)$$

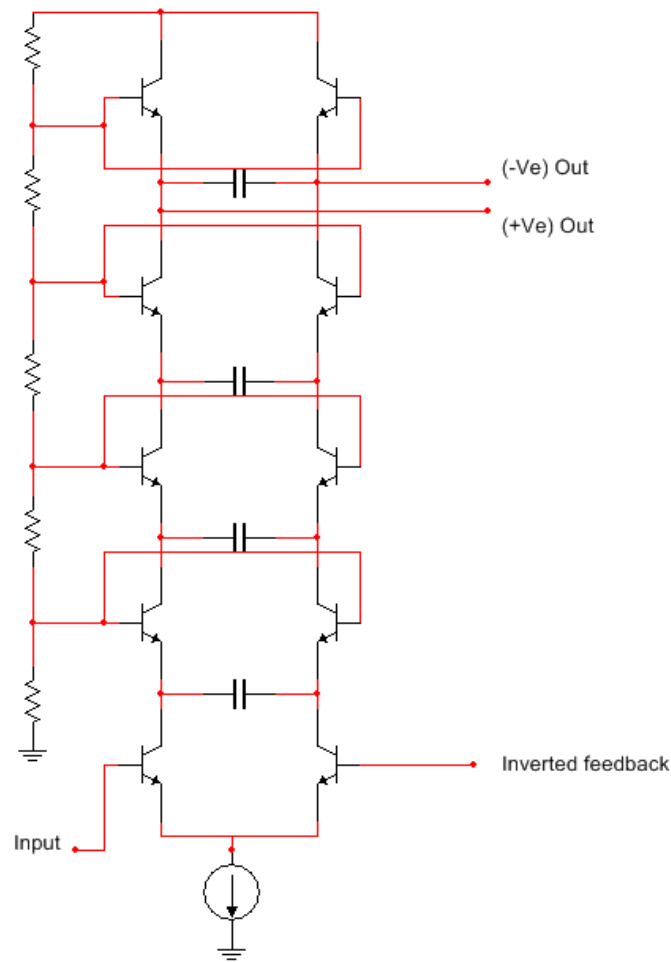


FIGURE 2.1: The core of the Moog filter is a ladder of four stages, each composed of a differential pair of NPN transistors and a capacitor. A signal is fed into the bottom of the ladder and output from the top. An inverted feedback signal path flows from the output back to the first stage of the ladder.

For  $\omega \ll \omega_c$ ,  $|H(j\omega)| \approx \frac{1}{1+k}$  but for  $\omega \gg \omega_c$ ,  $|H(j\omega)| \approx \frac{1}{\omega^4}$ . That is, frequency content below the cut-off frequency is left relatively unattenuated while that which is above is attenuated greatly: these are the characteristics we should expect from a low-pass filter.[15] The upper plot of Fig. 2.2 shows the magnitude response of the analogue filter when  $\omega_c = 100, 1000$  and  $10000$  Hz for feedback values  $k = 0, 2$  and  $3.99$ . Notice that as  $k$  increases the peak becomes increasingly greater and more focused and the pass band is increasingly attenuated.

When  $k = 0$ ,  $|G_1(0)| = 0$  and  $\angle G_1(0) = 0^\circ$ ; and  $|G_1(\infty)| = 0$  and  $\angle G_1(\infty)$  asymptotically approaches  $-360^\circ$ . The complex gain of a single filter stage at  $\omega_c$  is

$$G_1(j\omega_c) = \frac{1}{1+j} = \frac{1}{\sqrt{2}}e^{-j\frac{\pi}{4}} \quad (2.13)$$

such that the complex gain of the four combined filter stages is

$$G_1(j\omega_c)^4 = \frac{1}{4}e^{j\pi} = -\frac{1}{4}. \quad (2.14)$$

So, if an  $f_c$  Hz sinusoid is passed through the ladder the output will be  $\frac{1}{4}$  of the amplitude and completely inverted ( $-180^\circ$  “out of phase”) with respect to the input signal. The feedback signal is then inverted such that it recombines constructively with the input signal causing frequencies around  $f_c$  to be emphasized. As feedback gain  $k$  approaches four, the extent to which frequencies are emphasized increases while the bandwidth of the emphasized frequencies decrease until the filter begins to self-oscillate when  $k = 4$  and an  $f_c$  Hz tone is output<sup>2</sup>. We say that the inverted feedback causes *resonance* around the cut-off frequency and the frequency which is emphasized the most is the *resonant* frequency. Resonance is an important characteristic of the Moog filter’s sound and must be modelled accurately.[9] The lower plot of Fig. 2.2 shows the phase response of the analogue filter when  $\omega_c = 100, 1000$  and  $10000$  Hz for feedback values  $k = 0, 2$  and  $3.99$ .

The analogue frequency response is determined by the positions of the poles in the complex plane and in the case of the Moog VCF it is possible to determine the pole positions analytically.[8] Consider again the denominator in Eqn. 2.11 which is equal to zero at a pole:

$$k + (1 + s')^4 = 0 \quad (2.15)$$

where  $s' = \frac{s}{\omega_c}$  is the *normalized* frequency. The solution is

$$s' = -1 + (-k)^{\frac{1}{4}} \quad (2.16)$$

$$s' = -1 \pm k^{\frac{1}{4}} e^{\pm j\frac{\pi}{4}} \quad (2.17)$$

and a plot on the  $s$ -plane can be seen in Fig. 2.3. As  $k$  increases from zero to four, the position of the poles spread equally outwards in an “X” pattern from the point  $s' = (-1, 0)$ . When  $k = 4$ , the real valued component of the right hand set of poles equals zero and the filter self oscillates at  $f_c$  Hz.

## 2.3 Huovilainen’s implementation

In 2004 at the 7<sup>th</sup> International Conference on Digital Audio Effects (DAFx’04), Antti Huovilainen presented a paper on an implementation of the Moog ladder filter: he analysed the analogue circuit to produce a differential equation that he solved using

<sup>2</sup>On first inspection the range of the feedback co-efficient  $k \in [0, 4]$  may seem unusual but it is hoped now that with an understanding of Eqn. 2.14 its meaning is clear.

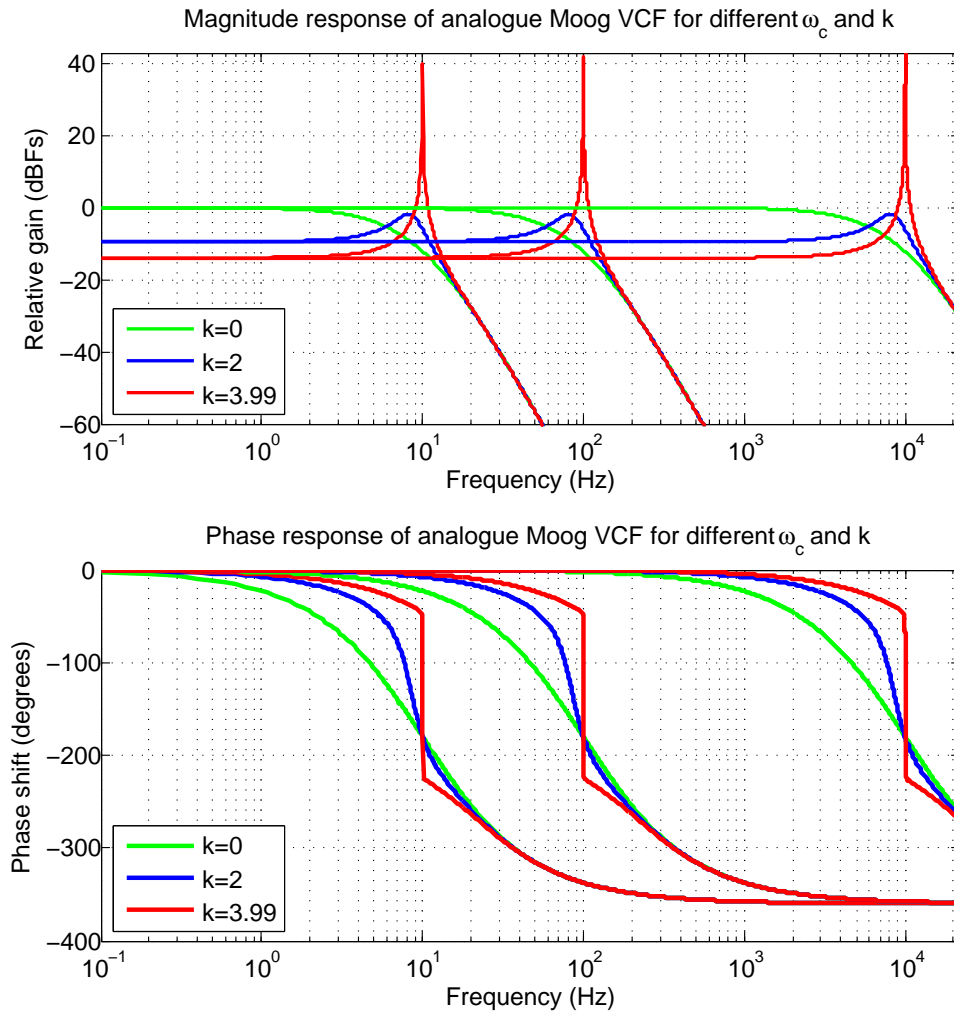


FIGURE 2.2: The upper plot shows the magnitude response for  $\omega_c=100$ , 1000 and 10000 Hz when  $k = 0, 2$  and 3.99. Note how the frequencies around  $\omega_c$  are increasingly emphasized as  $k \rightarrow 4$ . The frequency which is emphasized most for a given  $k$  is the resonant frequency. The lower plot shows the phase response for the same values of  $\omega_c$  and  $k$ . There is no phase shift when  $\omega = 0$  and as  $\omega \rightarrow \infty$  the phase shift approaches  $-360^\circ$ . The phase shift at  $\omega_c$  is  $-180^\circ$ ; the feedback signal is inverted with respect to the input causing constructive interference which causes frequencies around  $\omega_c$  to be emphasized.



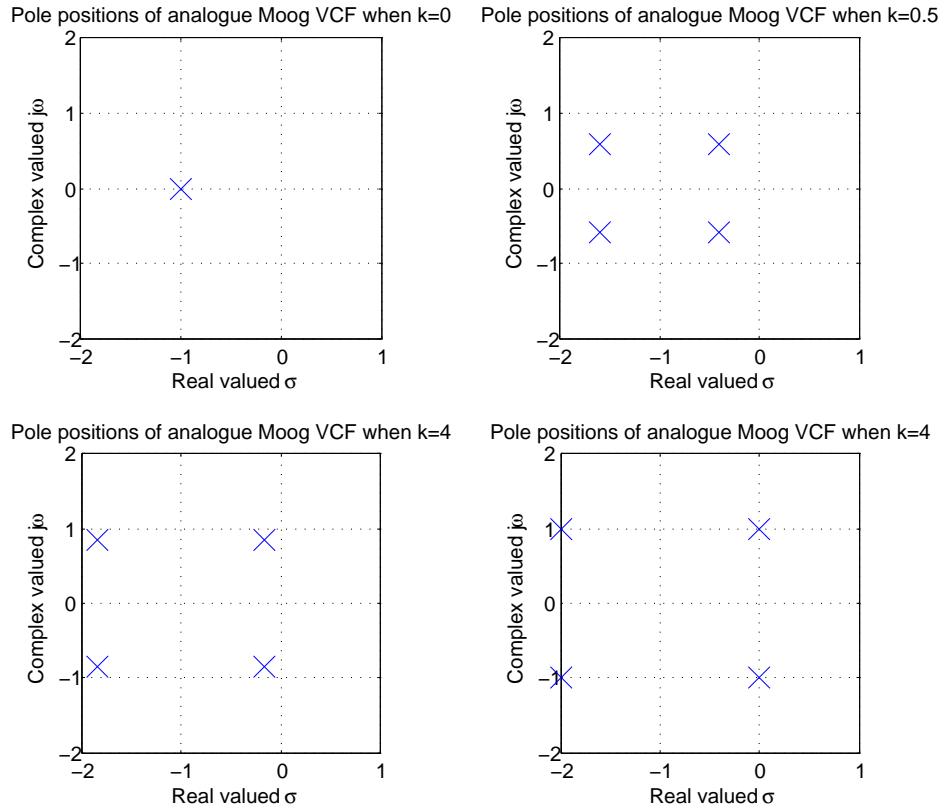


FIGURE 2.3: The complex analogue pole positions of the Moog VCF spread out in an “X” pattern from the point  $(-1,0)$  as  $k$  increases from 0 to 4. When  $k = 4$ , the right set of poles reach the vertical complex axis and the filter self-oscillates at the frequency  $f_c$ .

Euler’s method and showed that the result is equivalent to a cascade of first order IIR filters with embedded non-linear tanh functions; he then modified the filter structure to improve tuning. Huovilainen’s filter requires tuning as the feedback signal is delayed by one sample before it is mixed back into the input signal; this delay considerably affects the frequency response of the filter. Let us look in detail at Huovilainen’s model.[5]

He analyses the Moog VCF’s circuitry to arrive at Eqn. 2.7 which quantifies the change in voltage across the capacitor present in each stage. Huovilainen opts to stick with values that have dimensions but for the sake of clarity the dimensionless Eqn. 2.8 will be used instead. He solves the differential equation at time step  $n$  using Euler’s method

$$v_i^n = v_i^{n-1} + \frac{\omega_c}{F_s} (\tanh v_{i-1}^n - \tanh v_i^{n-1}) \quad (2.18)$$

where  $F_s$  is the sample rate<sup>3</sup>. [16] He notes that at small signal levels, where the tanh

<sup>3</sup>Sample rate, sample time and discrete operations, including Euler’s method, are covered succinctly in Sec. 2.5. It is advisable to be familiar with these concepts before continuing.

function is approximately linear, the above difference equation is that of a digital one pole low-pass filter

$$y^n = y^{n-1} + g(x^n - y^{n-1}) \quad (2.19)$$

where  $g$  is a weighting co-efficient.[14] He then performs a scaled impulse invariant transform<sup>4</sup> on the difference equation to calculate  $g$  in this case[17]

$$g = 1 - e^{-\omega_c/F_s}. \quad (2.20)$$

Huovilainen explains that in the analogue filter, an  $f_c$  Hz sinusoidal signal incurs a  $-90^\circ$  phase shift for each filter stage it passes through resulting in a  $-180^\circ$  phase shift over all. In his model a signal of frequency  $f$  undergoes the analogue phase shift  $p_{single}(f, F_c)$  per stage plus an additional, unwanted phase shift such that the resulting phase shift is

$$p = 4 \times p_{single}(f, F_c) + 180 \frac{f}{F_s} \quad (2.21)$$

in degrees. The unwanted shift offsets the resonant frequency from its required value and deviates the attenuation properties from the analogue filter. The result is that the feedback amount required to produce resonance varies with the frequency of the input signal.[5] Huovilainen opts to alter the filter structure itself. He proposes that a “unit-and-a-half-delay” feedback signal, that is the average of two previous output values, will improve the model’s frequency response. Huovilainen asserts that the remaining frequency response error can be eliminated using a tuning table.

### 2.3.1 Other implementations

Since Huovilainen published his Moog VCF model in 2004 two other models have been presented publicly. Thomas Hélie presented his model which made use of Volterra series at DAFx’06: he transforms the system of differential equations into an infinite set of algebraic equations from which Volterra kernels can be deduced.[6] Volterra series can capture the history of non-linear systems, notably the memory effects of the capacitors in the transistor ladder.[18] At ICM’07 Federico Fontana presented his delay free loop simulation where he “directly solves the contribution of the states of first-order lowpass filters to the signal.”[7] His simulation replicates the structure of the analogue filter to produce an accurate, weakly non-linear model.[19]

<sup>4</sup>The impulse invariant transform is scaled such that gain at DC is one.

## 2.4 The bilinear transform

The bilinear transform is a mapping between the  $s$ -plane and the  $z$ -plane that can preserve desirable properties of an analogue filter.[20] It is defined for values of  $s$  and  $z$

$$s = c \frac{1 - z^{-1}}{1 + z^{-1}} \quad (2.22)$$

$$z^{-1} = \frac{1 - s/c}{1 + s/c} \quad (2.23)$$

where  $s$  and  $z$  are the complex numbers related to the Laplace and  $Z$ -transform respectively. The positive constant  $c$  defines the mapping between  $s$  and  $z$ . When modelling an analogue filter, it is set such that the analogue and digital frequencies are equivalent.

The bilinear transform results in a one-to-one mapping between the analogue frequency axis  $s = j\omega_a$  and the digital frequency axis  $z = e^{j\omega_d/F_s}$ . The relative amplitude response is unchanged by the transform but there is a *frequency warping*: equal steps up the  $j\omega$  axis in the  $s$ -plane cause increasingly smaller increments around the unit circle in the  $z$ -plane.[21] The effects becomes prevalent at high frequencies and tends to “squash” the digital frequency response. The relationship between analogue frequency  $\omega_a$  and digital frequency  $\omega_d$  can be derived from the bilinear transformation in Eqn. 2.22 to show that

$$\omega_a = c \tan(\omega_d T_s / 2). \quad (2.24)$$

The compensated relationship between analogue and digital frequency is[7]

$$\omega_a = \frac{2}{T_s} \tan(\omega_d T / 2). \quad (2.25)$$

The constant  $c$  is chosen such that as  $\omega_d$  approaches zero and the tan function is nearly linear,  $\omega_a \approx \omega_d$  as required.

## 2.5 Discrete operators

Generally speaking, a digital audio signal is a discretized and quantized approximation to a continuously variable function of time. Audio sampling is the process of storing the values of a continuous time function in a time series. Values are evaluated and stored every  $T_s$  seconds.  $T_s$  is called the sample time and  $F_s = \frac{1}{T_s}$  is called the sample rate. If  $y^n$  is the  $n^{\text{th}}$  sample of a digital audio signal it corresponds to the value  $y(nT_s)$  of a continuous time function. By the Nyquist-Shannon sampling theorem the sample rate must be at least twice the value of the largest frequency that is to be sampled otherwise alias frequencies can appear in the digital signal. It may be beneficial to oversample

a signal, that is to increase the sample rate significantly higher than twice the highest frequency to be represented; this is of particular importance when working with non-linear systems. These concepts will now be assumed; for more information consult.[22]

The derivative of a function  $f$  at point  $x$  is defined by the limit

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}. \quad (2.26)$$

It is possible to approximate differential equations in a discrete domain using finite difference methods. If  $h$  is a non-zero, positive number, then the forward, backwards and central finite difference schemes respectively approximate the differential  $f'(x)$

$$f'(x) \approx \frac{f(x+h) - f(x)}{h} \quad (2.27)$$

$$f'(x) \approx \frac{f(x) - f(x-h)}{h} \quad (2.28)$$

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}. \quad (2.29)$$

The forward difference scheme approximates the differential equation at point  $x + \frac{1}{2}h$ ; the backwards difference scheme at point  $x - \frac{1}{2}h$  and the central difference scheme at point  $x$ . The error<sup>5</sup> scales quadratically with  $h$  in the central difference scheme but linearly in the forwards and backwards difference methods. In digital audio,  $h$  is the sample time  $T_s$ . By decreasing  $T_s$  not only do we increase accuracy, we increase the bandwidth of the system and decrease the risk of aliasing.[23]

We can use finite difference schemes to approximate solutions to a differential equation. Consider the problem  $y'(t) = f(t, y(t))$ . The forward difference equation approximates a solution

$$y(t+h) \approx y(t) + hy'(t) \quad (2.30)$$

$$y(t+h) \approx y(t) + hf(t, y(t)) \quad (2.31)$$

and similarly the backwards difference equation an approximation

$$y(t) \approx y(t-h) + hy'(t) \quad (2.32)$$

$$y(t) \approx y(t-h) + hf(t, y(t)). \quad (2.33)$$

---

<sup>5</sup>The error in this sense, is the difference between the differential and the finite difference approximation.

and the central difference equation an approximation

$$y(t+h) \approx y(t-h) + 2hy'(t) \quad (2.34)$$

$$y(t+h) \approx y(t-h) + 2hf(t, y(t)) \quad (2.35)$$

Each of these approximations can be implemented as recursive equations suitable for use in a programmable loop:

$$y^{n+1} = y^n + T_s f(nT_s, y(nT_s)) \quad (2.36)$$

$$y^n = y^{n-1} + T_s f(nT_s, y(nT_s)) \quad (2.37)$$

$$y^{n+1} = y^{n-1} + 2T_s f(nT_s, y(nT_s)) \quad (2.38)$$

for the forward, backward and central difference methods respectively.

Generally, speaking the backwards method is very stable but it is an implicit method. Its function can not be defined explicitly but instead depends upon the algebraic relationship between variables, and must be solved using a fixed point iterative method like Newton-Raphson's method<sup>6</sup>.[\[24\]](#) Newton-Raphson's method is discussed in Sec. [2.6](#).

In finite difference schemes it can be beneficial to replace a value with that of an averaged value, particularly in implicit schemes.[\[23\]](#) The averaging sum

$$y(n) = \frac{y(n) + y(n-1)}{2} \quad (2.39)$$

essentially acts as a low-pass filter and attenuates any high frequency fluctuations.[\[25\]](#) Furthermore, it can increase the accuracy of an approximation to a derivative. The Eqns. [2.36](#) and [2.37](#) approximate a solution to the function  $f(nT_s, y(nT_s))$  at time step  $n$  but they themselves are centred about time step  $n + \frac{1}{2}$  and  $n - \frac{1}{2}$  respectively. It is more accurate to solve functions  $f((n + \frac{1}{2})T_s, y((n + \frac{1}{2})T_s))$  or  $f((n - \frac{1}{2})T_s, y((n - \frac{1}{2})T_s))$ .

## 2.6 Iterative methods

Implicit equations cannot be solved analytically but their solutions *may* be approximated using the iterative Newton-Raphson method, a type of root finding algorithm which is part of the more general Householder methods.[\[26\]](#) An initial approximation is used in some formula to obtain a new approximation that is more accurate; the process is repeated until the approximation converges about the solution value with sufficient

<sup>6</sup>To make the problem clear, Eqn. [2.37](#) aims to find the value of  $y^n$  but this variable is used on *both* sides of the equation.

accuracy. Generally, this method is only successful when the initial approximation is in the region of the actual solution.

Loosely speaking, we can use Newton-Raphson's method to answer the question "where does a polynomial curve equal zero?": we make our best estimation as to where the curve crosses the horizontal axis; calculate the curve's value at this point; then calculate the equation of the line tangential to the curve at this point. We then calculate where the *tangential line* crosses the horizontal axis: generally, this value is a more accurate solution to the question than our initial approximation. The method can then be repeated until the tangential lines converge about one point. Now in a mathematical sense.

Consider an implicit polynomial equation  $f(x) = 0$  to be solved in  $x$ : the Taylor series expansion of  $f(x)$  about  $x = x_0 + \epsilon$ , where  $x_0$  is the initial approximation,

$$f(x_0 + \epsilon_0) = f(x_0) + f'(x_0)\epsilon_0 + \frac{1}{2}f''(x_0)\epsilon_0^2 + \dots \quad (2.40)$$

is approximately

$$f(x_0 + \epsilon_0) \approx f(x_0) + f'(x_0)\epsilon_0 \quad (2.41)$$

to first order accuracy. Eqn. 2.41 is the equation of the line tangential to  $f(x)$  at the point  $(x_0, f(x_0))$  and this line intercepts the  $x$ -axis at the point  $(x_1, 0)$  where  $x_1$  is our new, improved approximation. The horizontal "step" between consecutive approximations is

$$\epsilon_0 = -\frac{f(x_0)}{f'(x_0)} \quad (2.42)$$

such that  $x_1 = x_0 + \epsilon_0$ . The process is repeated iteratively such that

$$x_{n+1} = x_n + \epsilon_n \quad (2.43)$$

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}. \quad (2.44)$$

until the approximation has converged when  $\epsilon_n$  has reached zero or a sufficiently small value. When working with digital audio, the iteration can stop when  $\epsilon_n$  is less than the resolution of the system e.g. in a 24-bit system  $\epsilon_n$  need not be less than  $2^{-24}$ .

It is possible to apply Newton-Raphson's method to linear systems; We can find solutions when the number of unknowns in the system equals the number of equations. Consider

a system of  $n$  functions and  $n$  unknowns

$$f_1(x_1, \dots, x_n) = 0 \quad (2.45)$$

$$\vdots \quad (2.46)$$

$$f_n(x_1, \dots, x_n) = 0 \quad (2.47)$$

$$(2.48)$$

(or in vector notation  $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ ) where we would like to find the vector  $\mathbf{x} = (x_1, \dots, x_n)$ . We make an initial approximation to the vector  $\mathbf{x}_0$  based on our knowledge of the system, and make a linear approximation to  $\mathbf{f}(\mathbf{x})$  at  $\mathbf{x}_0$

$$\mathbf{f}(\mathbf{x}) \approx \mathbf{f}(\mathbf{x}_0) + \mathbf{Df}(\mathbf{x}_0)(\mathbf{x} - \mathbf{x}_0) \quad (2.49)$$

where  $\mathbf{Df}(\mathbf{x}_0)$  is a Jacobian matrix

$$\mathbf{Df}(\mathbf{x}_0) = \begin{bmatrix} \frac{\partial f_1(\mathbf{x}_0)}{\partial x_1} & \cdots & \frac{\partial f_1(\mathbf{x}_0)}{\partial x_n} \\ \vdots & \ddots & \vdots \\ -\frac{\partial f_n(\mathbf{x}_0)}{\partial x_1} & \cdots & \frac{\partial f_n(\mathbf{x}_0)}{\partial x_n} \end{bmatrix}. \quad (2.50)$$

What we aim to do is find a better approximation to  $\mathbf{x}$ ,  $\mathbf{x}_1$  such that

$$\mathbf{f}(\mathbf{x}_0) + \mathbf{Df}(\mathbf{x}_0)(\mathbf{x}_1 - \mathbf{x}_0) = \mathbf{0} \quad (2.51)$$

now, provided that matrix  $\mathbf{Df}(\mathbf{x}_0)$  is non-singular then

$$\mathbf{x}_1 = \mathbf{x}_0 + \Delta \mathbf{x} \quad (2.52)$$

$$\mathbf{x}_1 = \mathbf{x}_0 - (\mathbf{Df}(\mathbf{x}_0))^{-1} \mathbf{f}(\mathbf{x}_0). \quad (2.53)$$

Note the similar form to the single variable Newton-Raphson's method. In practice, we do not calculate the matrix inverse explicitly in each calculation but use some automated linear system solver to calculate  $\Delta \mathbf{x}$  from

$$\mathbf{Df}(\mathbf{x}_0) \Delta \mathbf{x} = -\mathbf{f}(\mathbf{x}_0). \quad (2.54)$$

The subsequent iterative steps are to solve

1. Solve  $\mathbf{Df}(\mathbf{x}_i) \Delta \mathbf{x} = -\mathbf{f}(\mathbf{x}_i)$  for  $\Delta \mathbf{x}$
2. Calculate  $\mathbf{x}_{i+1} = \mathbf{x}_i + \Delta \mathbf{x}$

## Chapter 3

# Implementation

The aim of the project was to improve on Huovilainen's Moog VCF model in terms of accuracy, stability or computational expense. The first step was to implement Huovilainen's model. The model was initially implemented as a strictly linear system so that frequency domain analysis was not based upon input signal level. The backwards finite difference model was constructed and the two models were tested in terms of ability to approximate the frequency response analytically derived by Smith and Stilson. Stability conditions and expense were also investigated.

Note that this project was carried out in MATLAB and certain functions, operations and structures (particularly vectors and matrices) will be referred to without detailing their exact implementation. Unless otherwise specified  $F_s = 88.2$  kHz with the intention of outputting 44.1 kHz wave audio files.

### 3.1 Linear models of the Moog VCF

The transfer function of the analogue Moog VCF as determined by Smith and Stilson is

$$H(s) = \frac{1}{k + (1 + \frac{s}{\omega_c})^4} \quad (3.1)$$

where  $s = 2\pi j f$  is a complex value and  $f$  is some frequency.  $H(s)$  was calculated for each integer  $f \in [0, 22050]$ . The magnitude response of the analogue filter was determined by plotting  $20 \log_{10} |H(s)|$ <sup>1</sup> across the range of  $f$ . The phase angles  $\angle H(s)$  were extracted and plotted against  $f$  to calculate the phase response in degrees.

---

<sup>1</sup>This type of scale is the full-scale decibel.[27]



Huovilainen’s first implementation of the non-linear Moog VCF is characterised by the recursive relations

$$y_a^n = y_a^{n-1} + g(x^n - ky_d^{n-1} - y_a^{n-1}) \quad (3.2)$$

$$y_b^n = y_b^{n-1} + g(y_a^n - y_b^{n-1}) \quad (3.3)$$

$$y_c^n = y_c^{n-1} + g(y_b^n - y_c^{n-1}) \quad (3.4)$$

$$y_d^n = y_d^{n-1} + g(y_c^n - y_d^{n-1}) \quad (3.5)$$

where  $g = 1 - e^{-\omega_{c,a}/F_s}$ ,  $\omega_{c,a} = \frac{2}{T_s} \tan(T_s \omega_{c,d}/2)$  is the “analogue” cut-off frequency and  $\omega_{c,d}$  is the “digital” cut-off frequency set by the user. The subscript notation  $a, b, c, d$  refers to the output of the first, second, third and fourth stages of the ladder rising from the bottom. It is essential that the calculations are carried out in the order that they are presented above. This implementation of Huovilainen’s model is referred to as the “unit-delay model”.

The input signal  $x$  was a delta signal  $N = 10^{14}$  samples long. The recursive relations were used in a loop  $N$  iterations long and the output variable  $y_d$  was stored for each iteration. The signal  $y_d$  is the filter’s impulse response. The filter’s magnitude response was calculated in dBFS as  $20 \log_{10} |FFT(y_d)|$  where  $FFT$  denotes the fast Fourier transform function. The phase angles of the complex valued impulse response were extracted to calculate the phase response. The magnitude and phase responses were plotted against the frequency bins of the FFT where the  $n^{th}$  bin equals  $F_s n/N$  for  $n \in [0, N - 1]$ .  $N$  was chosen to be so large to increase the resolution of the FFT and the plot.

Huovilainen stated that a “unit-and-a-half” feedback delay improves the phase response of the filter. The new set of recursion relations use the average of two previous output values

$$y_a^n = y_a^{n-1} + g \left( x^n - k \frac{y_d^{n-1} + y_d^{n-2}}{2} - y_a^{n-1} \right) \quad (3.6)$$

$$y_b^n = y_b^{n-1} + g (y_a^n - y_b^{n-1}) \quad (3.7)$$

$$y_c^n = y_c^{n-1} + g (y_b^n - y_c^{n-1}) \quad (3.8)$$

$$y_d^n = y_d^{n-1} + g (y_c^n - y_d^{n-1}). \quad (3.9)$$

Huovilainen argued that a look up tuning table will any eliminate any remaining error in the frequency response<sup>2</sup>. The tuning table was created by altering the cut-off frequency of his model such that its resonant peak was co-incident with that of the analogue frequency response. Table. A.1 shows the results of this tuning operation; it

<sup>2</sup>It is worthy of note that in the introduction to his paper Huovilainen states that no additional filter tuning is necessary as his filter is based on the analogue circuitry alone.[5]

can be implemented using an interpolating function. Note that these values are only functional when  $F_s = 88.2$  kHz. This tuned, unit-and-a-half delay forward difference implementation is referred to as Huovilainen's model.

The backwards difference equations that are constructed from the dimensionless system equation Eqn. 2.8 are

$$y_a^n = y_a^{n-1} + T_s \omega_{c,a} (x^n - k y_d^n - y_a^n) \quad (3.10)$$

$$y_b^n = y_b^{n-1} + T_s \omega_{c,a} (y_a^n - y_b^n) \quad (3.11)$$

$$y_c^n = y_c^{n-1} + T_s \omega_{c,a} (y_b^n - y_c^n) \quad (3.12)$$

$$y_d^n = y_d^{n-1} + T_s \omega_{c,a} (y_c^n - y_d^n) \quad (3.13)$$

where  $\omega_{c,a}$  is the ‘‘analogue’’ cut-off frequency as before. The system of implicit equations were solved in MATLAB by NR's method using matrices as described in Sec. 2.6. The initial approximation matrix  $\mathbf{Y}$  was a four-by-one zero matrix. The system matrix  $\mathbf{f}$  was

$$\mathbf{f} = \begin{bmatrix} y_a^n - y_a^{n-1} - T_s \omega_{c,a} (x^n - k y_d^n - y_a^n) \\ y_b^n - y_b^{n-1} - T_s \omega_{c,a} (y_a^n - y_b^n) \\ y_c^n - y_c^{n-1} - T_s \omega_{c,a} (y_b^n - y_c^n) \\ y_d^n - y_d^{n-1} - T_s \omega_{c,a} (y_c^n - y_d^n) \end{bmatrix} \quad (3.14)$$

and the corresponding Jacobian matrix  $\mathbf{Df}$  was

$$\mathbf{Df} = \begin{bmatrix} 1 + T_s \omega_{c,a} & 0 & 0 & T_s k \omega_{c,a} \\ -T_s \omega_{c,a} & 1 + T_s \omega_{c,a} & 0 & 0 \\ 0 & -T_s \omega_{c,a} & 1 + T_s \omega_{c,a} & 0 \\ 0 & 0 & -T_s \omega_{c,a} & 1 + T_s \omega_{c,a} \end{bmatrix}. \quad (3.15)$$

The approximation error matrix  $\Delta \mathbf{Y}$  was calculated using a linear solve operation which solved the equality  $\mathbf{Df} * \Delta \mathbf{Y} = -\mathbf{f}$ . The approximation for iteration  $p$  was  $\mathbf{Y}^{p+1} = \mathbf{Y}^p + \Delta \mathbf{Y}^p$ . The output of the filter for each sample is value  $\mathbf{Y}_{4,1}$ . The matrices  $\mathbf{F}$ ,  $\mathbf{DF}$  and  $\Delta \mathbf{Y}$  were repeatedly calculated until the maximum value of  $|\Delta \mathbf{F}|$  was less than a threshold equal to  $10^{-16}$ . The threshold was chosen assuming the audio output would be exported as 16-bit wave file. It is advisable to set an upper limit to the number of iterations so that infinite loops are impossible; the author chose fifty. In the same manner as above, the magnitude and phase responses were calculated for this implementation.

The system of equations were then centred about sample point  $n - \frac{1}{2}$ . The function and Jacobian matrices were then

$$\mathbf{f} = \begin{bmatrix} y_a^n - y_a^{n-1} - T_s \omega_{c,a} \left( x^n - k \frac{y_d^n + y_d^{n-1}}{2} - \frac{y_a^n + y_a^{n-1}}{2} \right) \\ y_b^n - y_b^{n-1} - T_s \omega_{c,a} \left( \frac{y_a^n + y_a^{n-1}}{2} - \frac{y_b^n + y_b^{n-1}}{2} \right) \\ y_c^n - y_c^{n-1} - T_s \omega_{c,a} \left( \frac{y_b^n + y_b^{n-1}}{2} - \frac{y_c^n + y_c^{n-1}}{2} \right) \\ y_d^n - y_d^{n-1} - T_s \omega_{c,a} \left( \frac{y_c^n + y_c^{n-1}}{2} - \frac{y_d^n + y_d^{n-1}}{2} \right) \end{bmatrix} \quad (3.16)$$

and

$$\mathbf{Df} = \begin{bmatrix} 1 + \frac{T_s}{2} \omega_{c,a} & 0 & 0 & \frac{T_s}{2} k \omega_{c,a} \\ -\frac{T_s}{2} \omega_{c,a} & 1 + \frac{T_s}{2} \omega_{c,a} & 0 & 0 \\ 0 & -\frac{T_s}{2} \omega_{c,a} & 1 + \frac{T_s}{2} \omega_{c,a} & 0 \\ 0 & 0 & -\frac{T_s}{2} \omega_{c,a} & 1 + \frac{T_s}{2} \omega_{c,a} \end{bmatrix}. \quad (3.17)$$

All other processes concerning Newton-Raphson's method were kept the same. Note that unless otherwise specified this is the (centred) backwards difference model. The magnitude and phase responses were calculated for this model.

The absolute magnitude error of both models were calculated and plotted for a low frequency of 1000 Hz and a high frequency of 10000 Hz while  $k = 3.99$ . The error was calculated by the equation

$$\text{absolute magnitude error} = \frac{H_{dB} - M_{dB}}{H_{dB}} \times 100 \quad (3.18)$$

for a range of frequencies where  $H_{dB}$  is the analogue magnitude response and  $M_{dB}$  is the approximated magnitude response. Both responses were measured in dBFs.

### 3.2 Non-linear models of the Moog VCF

It was simple to transform Huovilainen's model into a non-linear system. The tanh functions were inserted into Eqns. 3.6-3.9 and his non-linear model was characterised by the recursive relations

$$y_a^n = y_a^{n-1} + g \left( \tanh \left( x^n - k \frac{y_d^{n-1} + y_d^{n-2}}{2} \right) - \tanh y_a^{n-1} \right) \quad (3.19)$$

$$y_b^n = y_b^{n-1} + g (\tanh y_a^n - \tanh y_b^{n-1}) \quad (3.20)$$

$$y_c^n = y_c^{n-1} + g (\tanh y_b^n - \tanh y_c^{n-1}) \quad (3.21)$$

$$y_d^n = y_d^{n-1} + g (\tanh y_c^n - \tanh y_d^{n-1}) \quad (3.22)$$

and all other aspects of the linear model were kept the same.

The function matrix  $\mathbf{f}$  of the non-linear, centred, backwards difference model was

$$\mathbf{f} = \begin{bmatrix} y_a^n - y_a^{n-1} - T_s \omega_{c,a} \left( \tanh \left( x^n - k \frac{y_d^n + y_d^{n-1}}{2} \right) - \tanh \frac{y_a^n + y_a^{n-1}}{2} \right) \\ y_b^n - y_b^{n-1} - T_s \omega_{c,a} \left( \tanh \frac{y_a^n + y_a^{n-1}}{2} - \tanh \frac{y_b^n + y_b^{n-1}}{2} \right) \\ y_c^n - y_c^{n-1} - T_s \omega_{c,a} \left( \tanh \frac{y_b^n + y_b^{n-1}}{2} - \tanh \frac{y_c^n + y_c^{n-1}}{2} \right) \\ y_d^n - y_d^{n-1} - T_s \omega_{c,a} \left( \tanh \frac{y_c^n + y_c^{n-1}}{2} - \tanh \frac{y_d^n + y_d^{n-1}}{2} \right) \end{bmatrix} \quad (3.23)$$

and the corresponding Jacobian matrix  $\mathbf{DF}$  was

$$\mathbf{Df} = \begin{bmatrix} 1 + \frac{T_s}{2} \omega_{c,a} Y_a & 0 & 0 & \frac{T_s}{2} k \omega_{c,a} Y_d \\ -\frac{T_s}{2} \omega_{c,a} & 1 + \frac{T_s}{2} \omega_{c,a} Y_b & 0 & 0 \\ 0 & -\frac{T_s}{2} \omega_{c,a} & 1 + \frac{T_s}{2} \omega_{c,a} Y_c & 0 \\ 0 & 0 & -\frac{T_s}{2} \omega_{c,a} & 1 + \frac{T_s}{2} \omega_{c,a} Y_d \end{bmatrix}. \quad (3.24)$$

where

$$Y_a = \left( 1 - \tanh^2 \frac{y_a^n + y_a^{n-1}}{2} \right) \quad (3.25)$$

$$Y_b = \left( 1 - \tanh^2 \frac{y_b^n + y_b^{n-1}}{2} \right) \quad (3.26)$$

$$Y_c = \left( 1 - \tanh^2 \frac{y_c^n + y_c^{n-1}}{2} \right) \quad (3.27)$$

$$Y_d = \left( 1 - \tanh^2 \frac{y_d^n + y_d^{n-1}}{2} \right) \quad (3.28)$$

and all other aspects of the linear model are kept the same. A 100 Hz sinusoid with an amplitude of ten was sent through the filter when  $f_c = 20000$  Hz and  $k = 0$ . The wave profile and frequency content of the input and output were compared.

The forward and backwards difference filters were tested for stability. A 100 Hz square wave was input to the filter and the output waveform plotted. If the signal was unbounded, that is it approached infinity, the filter state was said to be unstable. The threshold of stability was investigated for each model. Two parameters (of cut-off frequency, feedback and input signal amplitude) were held constant while the other was systematically varied. In some cases, the filter would become unstable at a definite point and the threshold was easily recognisable, in other cases the process was gradual and the threshold was approximate.

The computational expense of both filters were compared by counting the number of floating point operations and memory stores required for each sample of audio output. A clock was initiated at the beginning of each filter section for a range of filter states. A 100 Hz square wave of duration 1 s was input to each filter and the clock time of the filter section was noted. The experiment was repeated five times to calculate a mean value.

# Chapter 4

## Results

### 4.1 Accuracy

This section contains frequency response plots which demonstrate the accuracies (and inaccuracies) of each model of the Moog VCF. Each model was progressively improved to demonstrate the effects of centred averaging and the benefits of a tuning table. It was found that both models struggled to produce accurate frequency responses for high values of  $f_c$  and  $k$  and this is where our attention must be focused.

Huovilainen's unit-and-a-half delay model was significantly better at approximating the analogue Moog VCF's magnitude response than his unit-delay model. Consider Fig. 4.1 which shows the magnitude and phase response of the unit delay model for values of  $f_c = 100, 1000$  and  $10000$  Hz while  $k = 3.99$ . Notice that the profile of the analogue magnitude response remains uniform for each value of  $f_c$  and that there is more than a 50 dB difference between the resonant frequency peak and the pass band. Observe that the profile of Huovilainen's magnitude response varies across the range of  $f_c$ . The resonant peak is gradually attenuated, becomes wider and drifts upwards as  $f_c$  increases.

It is useful to remember at this point that the analogue phase response has three characteristics. The phase shift: equals  $0^\circ$  when  $f = 0$ ; equals  $-180^\circ$  when  $f = f_c$  and asymptotically approaches  $-360^\circ$  as  $f \rightarrow \infty$ . When  $f_c = 100$  Hz, Huovilainen's phase response closely matches the analogue phase response from 0 to 10000 Hz. At higher frequencies it drifts away but the filter is attenuating these frequencies by more than 60 dB and so this is quite inconsequential. When  $f_c = 1000$ , Huovilainen's phase response loosely retains the first two characteristics of the analogue phase response but it also fails at higher frequencies. The filter attenuate these frequencies by approximately 40 dB which may be audible in some situations. The phase response of the filter when  $f_c = 10000$  Hz only manages to fulfil the first filter characteristic.

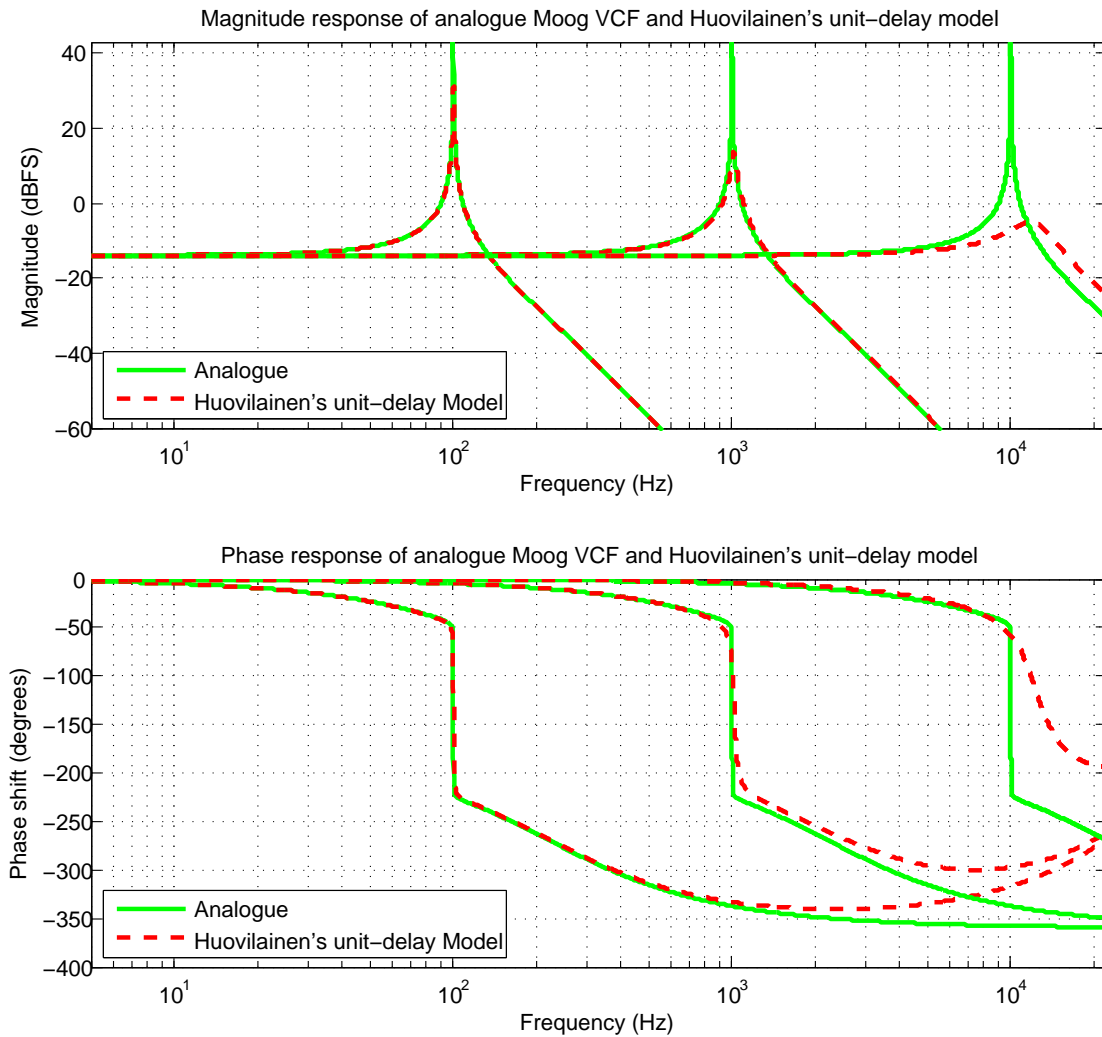


FIGURE 4.1: The upper plot shows the difference between the magnitude response of Huovilainen's unit-delay model of the Moog VCF and the analogue magnitude response. The bottom plot shows the difference in phase response. His model struggles to match the analogue response as the cut-off frequency is increased from 100 to 1000 to 10000 Hz while  $k = 3.99$

Fig. 4.2 shows the frequency response of Huovilainen's unit-and-a-half delay model for the same filter states as above. With comparison to Fig. 4.1, note how the magnitude response profiles remain more consistent as  $f_c$  increases and how the resonant peaks at 100 and 1000 Hz nearly match the analogue magnitude response. The resonant peak when  $f_c = 10000$  Hz is attenuated and shifted downwards with respect to the analogue response but it is considerably better than the one found in Fig. 4.1. A slight dip in the pass band has been incurred due to the changes to the filter's structure.

Similarly to the unit-delay model, the phase response curves closely approximate the desired curves until  $f = 10000$  Hz. As before, this is not so crucial when  $f_c = 100$  and 1000 Hz but is important at higher cut-off frequencies. The curve when  $f_c = 1000$

fulfils the first two filter phases characteristics better than the unit-delay model; in particular, the curve is steeper about the cut-off frequency. Similarly for the curve when  $f_c = 10000$  Hz: despite its failure to match the analogue phase response, it is an improvement due to the steep gradient around the cut-off frequency although the rise above  $0 \text{ rads}^{-1}$  is unfortunate. For  $f > 10000$  Hz, the phase response is wholly inaccurate.

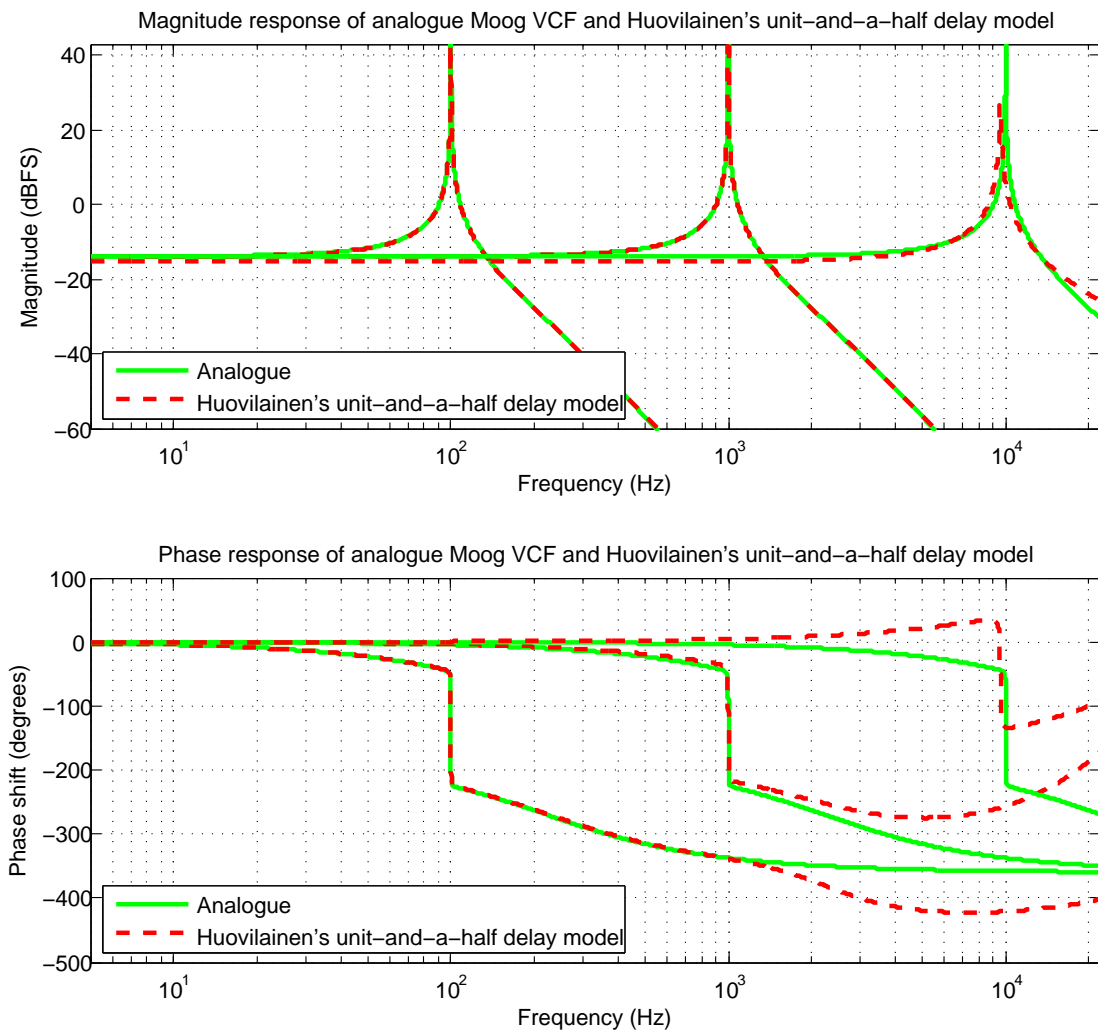


FIGURE 4.2: The upper plot shows the difference between the magnitude response of Huovilainen's unit-and-a-half-delay model of the Moog VCF and the analogue magnitude response. The bottom plot shows the difference in phase response. Both the magnitude and phases responses are improved greatly by the addition of a half-unit delay in the feedback path.

Fig. 4.3 demonstrates the effect of filter tuning. Notice that the third peak when  $f_c = 10000$  Hz is now co-incident with the resonant peak of the analogue filter and similarly for the phase response. Huovilainen's filter was only able to output a bound, stable impulse response up  $f_c = 17000$  Hz which is where the tuning table Tab. A.1 ends.



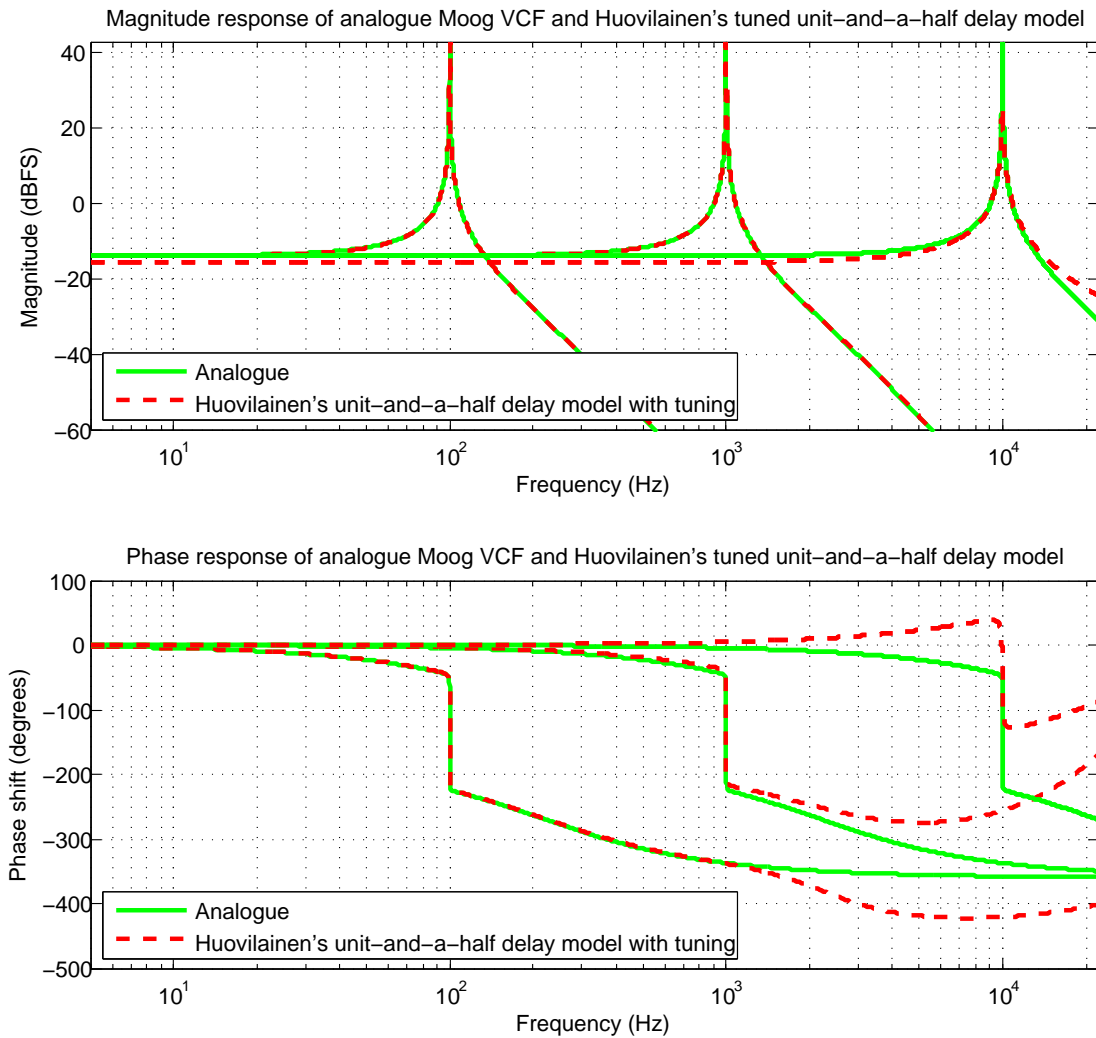


FIGURE 4.3: The upper plot shows the difference between the magnitude response of Huovilainen's tuned unit-and-a-half-delay model of the Moog VCF and the analogue magnitude response. The bottom plot shows the difference in phase response. Tuning the filter has a considerable effect on the magnitude response particularly at higher frequencies.

Fig. 4.4 plots the frequency response of the backwards difference model of the Moog VCF for the same previous filter states. The resonant peaks do not match their analogue counterparts in magnitude for any value of  $f_c$  and become increasingly attenuated as  $f_c$  increases. However, the peaks are in the desired position.

The phase response when  $f_c = 100$  closely matches the analogue response and begins to differ when  $f > 10000$  Hz. The phase response when  $f_c = 1000$  loosely follows the analogue response but does not have the required steep gradient about the cut-off frequency. The third curve corresponding to  $f = f_c$  manages to fulfil the first filter characteristic but begins to falter when  $f = 500$  Hz and makes a far too shallow curve

down to  $f = f_c$ . None of the phase response so far have approach a phase shift of  $-360^\circ$  as  $f \rightarrow \infty$ .

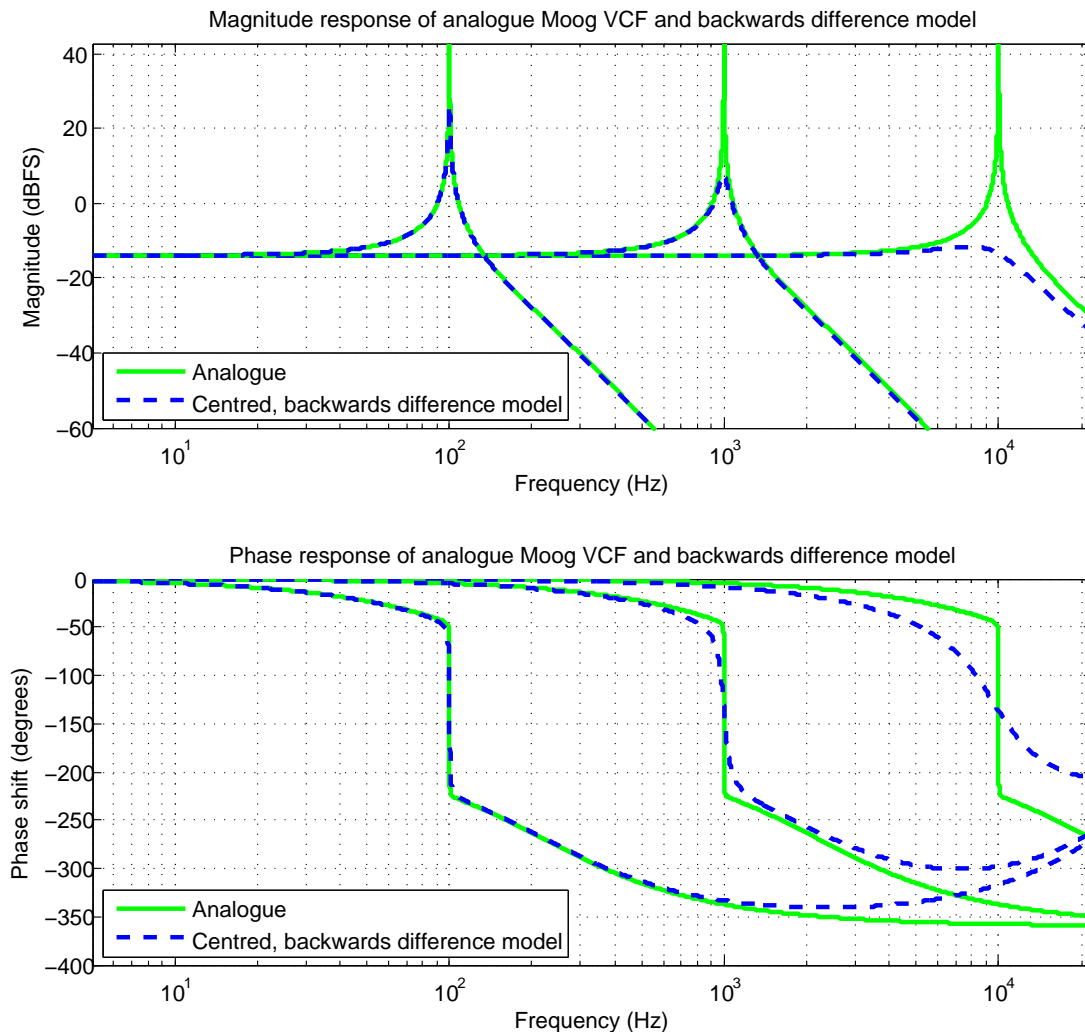


FIGURE 4.4: The upper plot shows the difference between the magnitude response of the backwards difference model of the Moog VCF and the analogue magnitude response. The bottom plot shows the difference in phase response. The peaks of the magnitude response are heavily attenuated but they are in the correct position. The phase response curves loosely follow the analogue curve at lower values of  $f$ .

Fig. 4.5 plots the frequency responses of the centred backwards difference model. The approximation closely matches the resonant peaks of the analogue filter both in terms of magnitude and frequency for each value of  $f_c$ . The peak drifts slightly as  $f_c$  increase but still remains very accurate and is the best approximation of all the models. Note that no tuning table was required to implement this model.

The phase responses of this model are also the most accurate: while none of them fulfil all three requirements entirely, they all handle the steep gradient about the cut-off frequency well and equal  $-180^\circ$  when  $f = f_c$ .

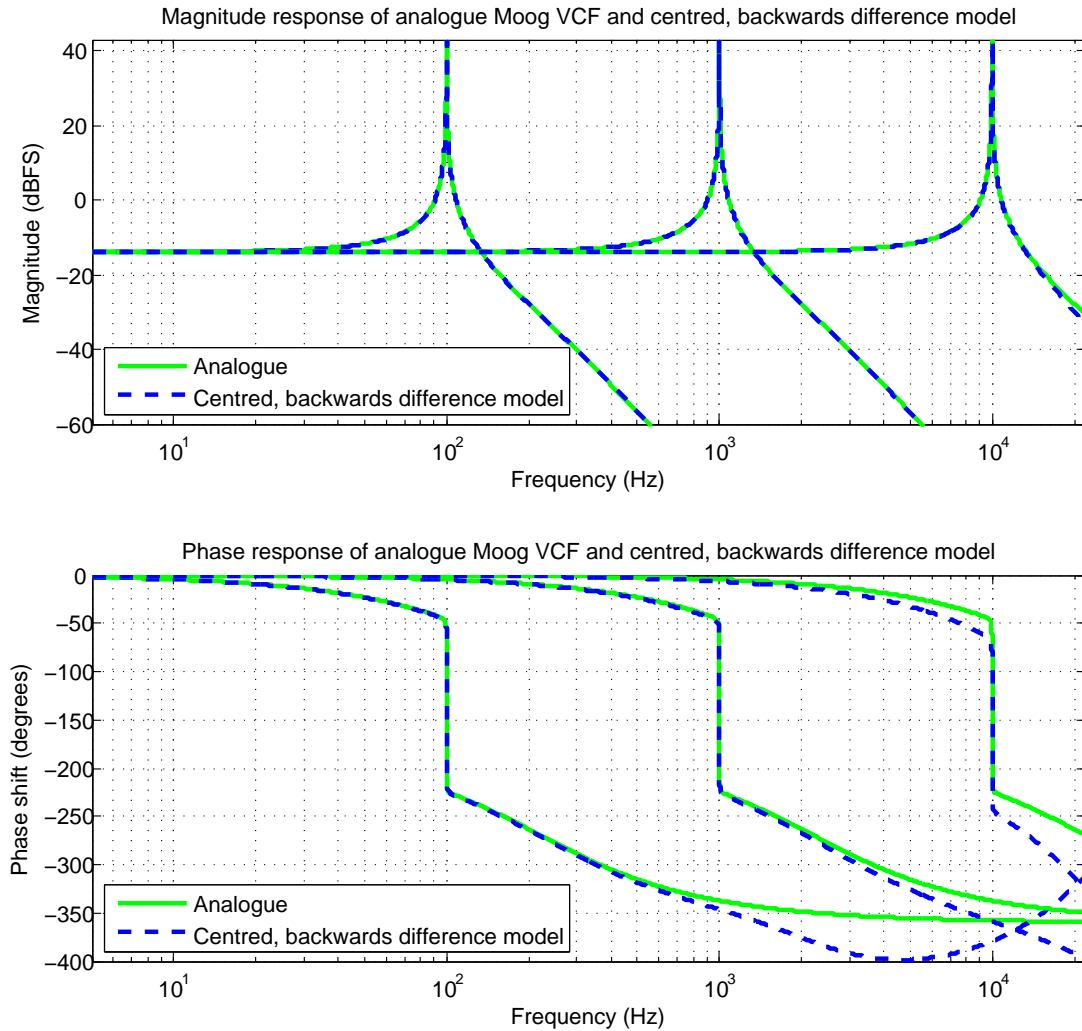


FIGURE 4.5: The upper plot shows the difference between the magnitude response of the centred, backwards difference model of the Moog VCF and the analogue magnitude response. The bottom plot shows the difference in phase response. This response is the most accurate of all the implementations. The phase response does still not approach  $-360^\circ$  as  $f \rightarrow \infty$ .

Fig. 4.6 shows in detail the resonant peak for each model when  $f_c = 14000$  Hz and  $k = 3.99$ . The backwards model closely matches the magnitude response of the analogue Moog VCF but Huovilainen's model has failed quite considerably: its resonant peak is 20 dB less than the analogue model. The phase response also fails to meet the necessary requirements: a positive shift is incurred at low values of  $f$  and the phase shift is  $0^\circ$  at  $f_c$ . This is far from the characteristics of the analogue filter. The backwards model makes a good attempt at approximating the necessary curve and recreates the necessary steep gradient about the cut-off frequency.

Fig. 4.7 demonstrates the non-linearity of the centred, backwards Moog model. The upper left plot shows the waveform of a 100 Hz sinusoid of amplitude 10 which was

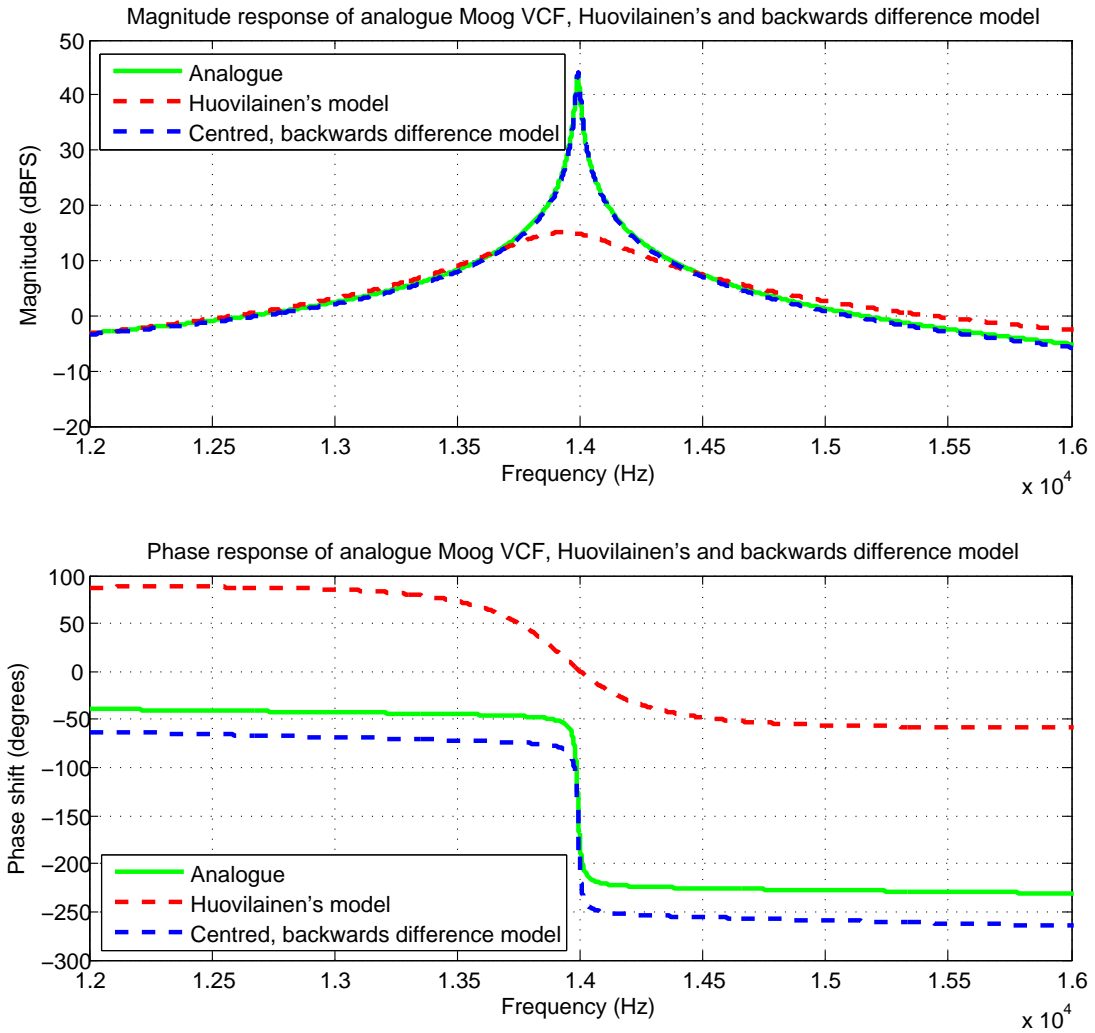


FIGURE 4.6: The frequency response is compared in detail for each model when  $f_c = 14000$  Hz and  $k = 3.99$ . The centred, backwards difference model closely matches the magnitude response peak and the phase response curve. Huovilainen's model is attenuated by more than 20 dB and the phase response does not have any of the required characteristics.

input to the filter. The bottom left plot shows that the sinusoid has only one spectral component focused on 100 Hz. The upper right plot shows the waveform of the filter output: not only has the signal been attenuated its profile has been altered, becoming more like a square wave. The bottom right hand plot shows the output signal has many more spectral components. The 100 Hz component remains but there also other peaks at odd harmonics caused by the non-linear, odd tanh function. The input and output signals can be heard in the audio file entitled "Example of Harmonic Distortion" on the accompanying data CD.

The inaccuracies of Huovilainen's tuned, unit-and-a-half delay filter and the centred,

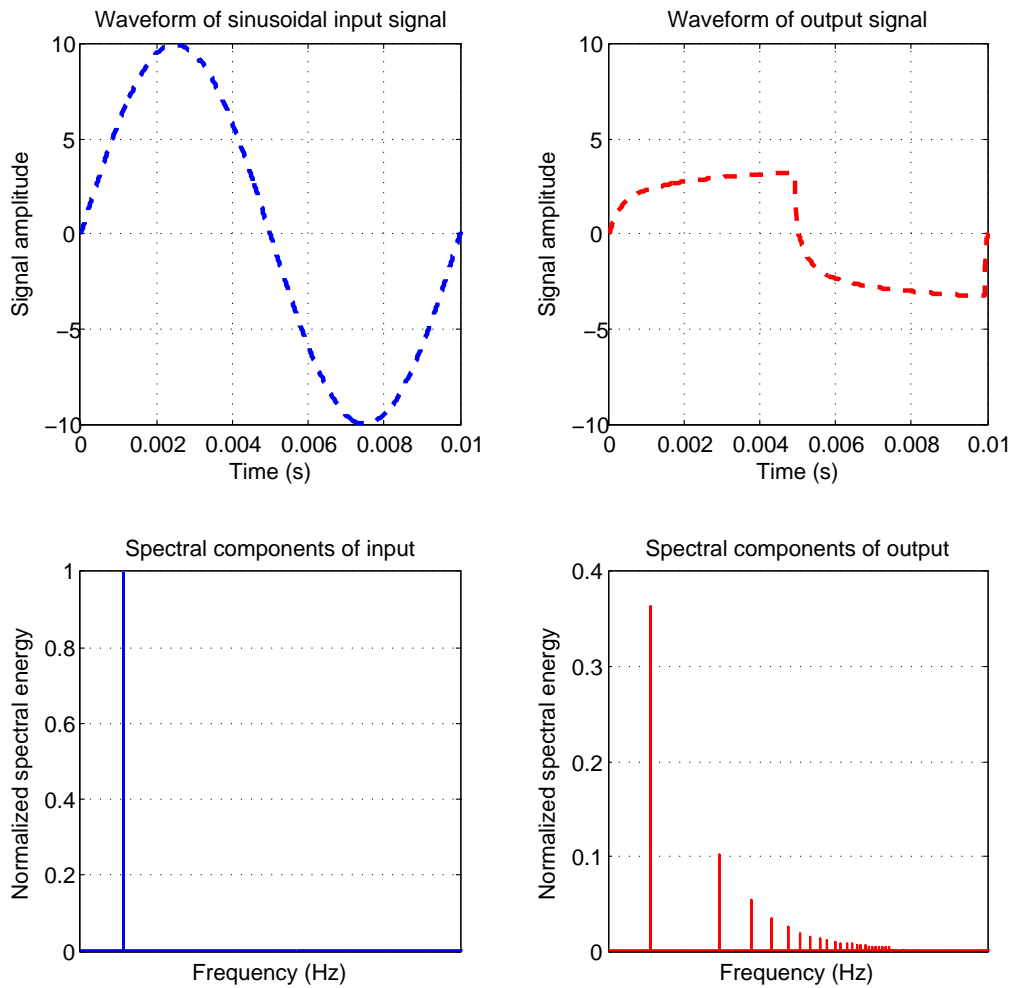


FIGURE 4.7: A 100 Hz sinusoidal signal of amplitude 10 is input to the non-linear centred, backwards Moog model (with  $F_c = 20000$  and  $k = 0$ ) and a square like signal of reduced amplitude is output. The odd tanh function produces odd-order harmonic distortion which causes the “squaring” of the sinusoid.

backwards difference will now be calculated and compared. Note that as all implementations were able to approximate the  $f_c = 100$  Hz magnitude response well then this curve can be ignored.

Fig. 4.8 shows the absolute magnitude error in percent of each magnitude response approximation when  $f_c = 1000$  Hz and  $k = 3.99$ . The upper plot is of Huovilainen’s model and the lower is of the centred, backwards model. This plot was calculated by Eqn. 3.18 for  $f = 100$  to 10000 Hz. Both plots have a constant error from 100 Hz until 900 Hz upon which the error increases massively for a very narrow band of frequencies, settles back down temporarily, then increases hugely at 1100 Hz for another narrow band of frequencies after which it remains constant until the end of the plot at 10000 Hz. In

Huovilainen's model, when the error is constant it is equal to  $-0.03\%$  and the two error peaks equal  $-2292\%$   $-796\%$  respectively. In the backwards model, the constant error is  $-0.06\%$  and the two error peaks equal  $-274\%$  and  $-109\%$ .

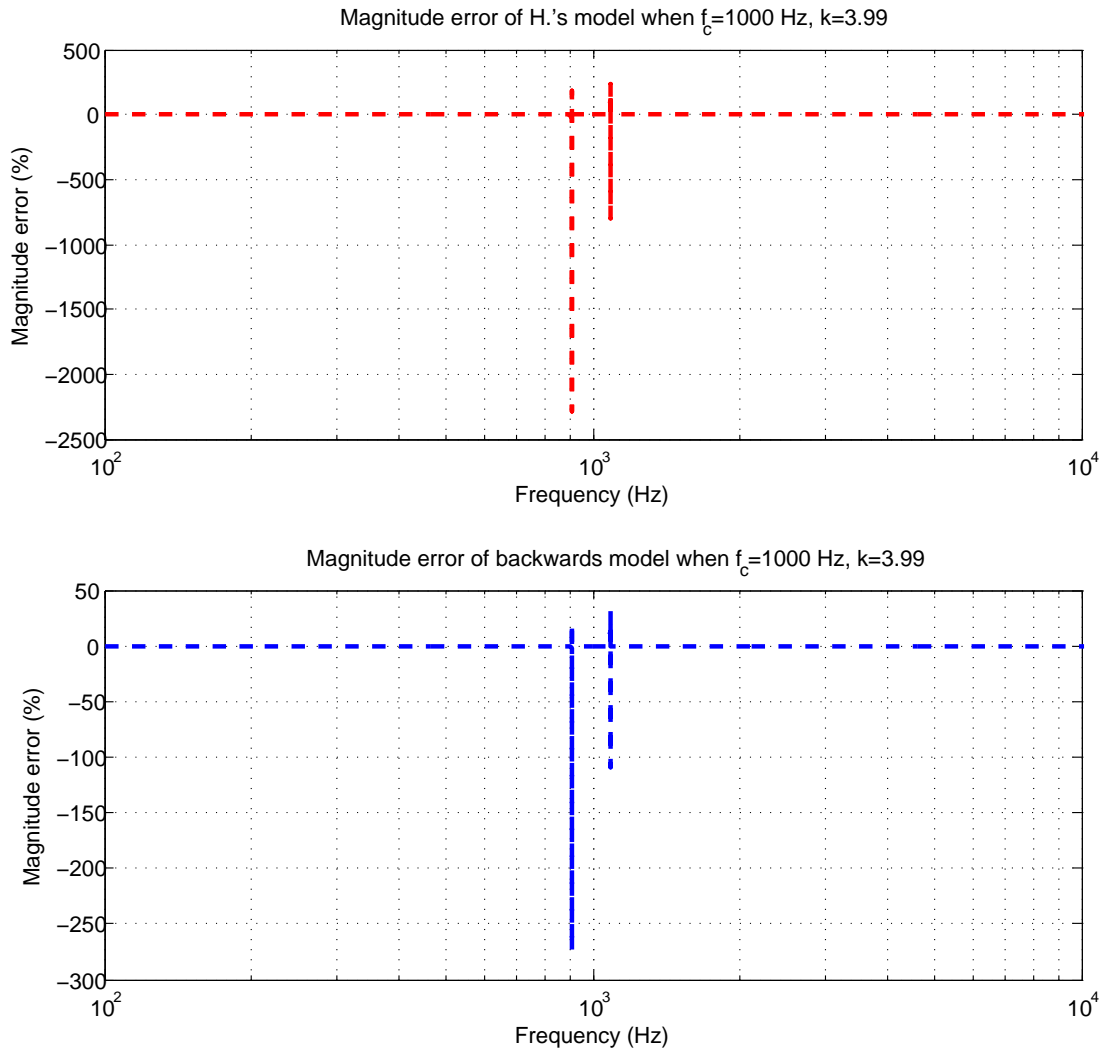


FIGURE 4.8: The upper plots show the magnitude error of Huovilainen's magnitude response when  $f_c = 1000$  Hz. The lower plot is from the centred, backwards model. Both plots display a small, constant error for most of the frequency range and a very large error at  $f = 990$  and  $1100$  Hz.

Fig. 4.9 shows the error in the magnitude response when  $f_c = 1000$  Hz and  $k = 3.99$ . These plots are characterised by a small, constant error for the majority of the frequency band and a large, very narrow error peak close to the cut-off frequency at  $f = 9043$  Hz. In Huovilainen's model, the error across most of the frequency range is  $-11.4\%$  and the error peak equals  $2.05 \times 10^5\%$ . The error in magnitude response for the backwards model is  $-0.6\%$  for the majority of the frequency range and it peaks at  $6.92 \times 10^4\%$  when  $f = 9043$  Hz.

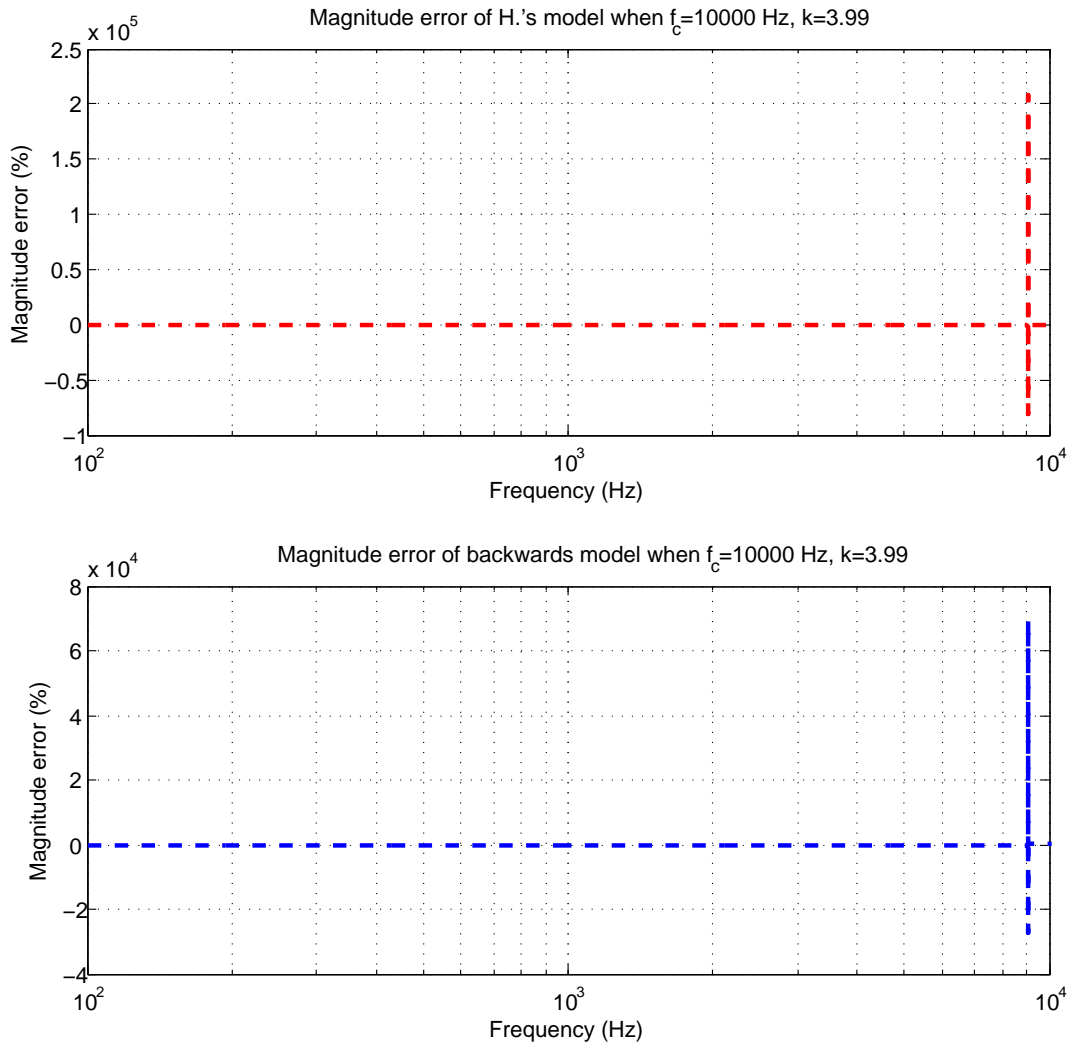


FIGURE 4.9: The upper plots show the magnitude error of Huovilainen's magnitude response when  $f_c = 10000$  Hz. The lower plot is from the centred, backwards model. Both plots display a small, constant error for most of the frequency range and a very large error at  $f = 9043$  Hz.

## 4.2 Stability

The forward difference scheme method employed by Huovilainen is inherently less stable than the backwards difference scheme, his model is resolutely more stable than the centred, backwards model. For all input amplitudes and values of  $k$  tested, Huovilainen's model produced a bounded output for  $f_c \in [0, 44100]$  Hz, that is from the DC to Nyquist frequency. Tab. 4.1 summarizes the settings which just brought about unstable settings, that is these values are just over the *threshold* of stability. Note that accuracy or quality of output is not of concern here but the aim is to find the settings which bring about infinitely large outputs regardless of the scale of the input. It is clear to see that neither

input amplitude nor feedback gain has any effect on stability and that the filter only becomes unstable when  $f_c > 44100$  Hz.

Input signal amplitude	Feedback co-efficient $k$	Cut-off frequency $f_c$ (Hz)
0.1	0	44101
0.1	1	44101
0.1	2	44101
0.1	3	44101
0.1	4	44101
1	0	44101
1	1	44101
1	2	44101
1	3	44101
1	4	44101
2	0	44101
2	1	44101
2	2	44101
2	3	44101
2	4	44101
4	0	44101
4	1	44101
4	2	44101
4	3	44101
4	4	44101
10	0	44101
10	1	44101
10	2	44101
10	3	44101
10	4	44101

TABLE 4.1: Huovilainen’s model is very stable. Neither input signal amplitude or feedback gain has an affect on stability. The filter only becomes unstable when  $f_c > \frac{F_s}{2}$ .

The threshold of stability in the centred, backwards model is variable and not as clearly recognisable as in Huovilainen’s model. Tab 4.2 displays filters states which are unstable. Note that when a value of  $f_c$  is approximated it means to say that the output gradually varied from bounded to infinite around this value.

At low input signal level 0.1 the filter is stable from DC to (almost) the Nyquist frequency. However, as the signal level increases to 1, 2 and 4 the frequency range drops from around 44, to 40, to 35 kHz. When the input amplitude is ten the filter becomes unstable when  $f_c \approx 20$  kHz. The feedback co-efficient  $k$  does affect the stability of the filter although it is hard to define any trend.



Input signal amplitude	Feedback co-efficient $k$	Cut-off frequency $f_c$ (Hz)
0.1	0	44097
0.1	1	44099
0.1	2	44098
0.1	3	44099
0.1	4	44099
1	0	41385
1	1	44078
1	2	44065
1	3	44050
1	4	39697
2	0	$\approx 35800$
2	1	39882
2	2	43299
2	3	41033
2	4	35178
4	0	$\approx 35850$
4	1	$\approx 25250$
4	2	28478
4	3	32732
4	4	30388
10	0	$\approx 35900$
10	1	23728
10	2	$\approx 19000$
10	3	$\approx 16000$
10	4	$\approx 15400$

TABLE 4.2: The values in this table are the threshold of unstable filter states for the backwards difference model of the Moog VCF. Input signal amplitude is the biggest determining factor of its stability.

### 4.3 Computational expense

To implement Huovilainen’s non-linear model, one must carry out 16 multiplications and additions; eight uses of the tanh function; eight stores of floating point numbers and one write-to-output operation per sample of output. The backwards model is considerably more demanding: it requires 95 additions and multiplications; 13 uses of the tanh function; 22 stores of floating point numbers; one linear system solve; one conditional “if” statement and one write-to-output operation per *iteration* of Newton-Raphson’s method.

The number of iterations of Newton-Raphson’s method needed per sample of output is proportional to the input signal amplitude and  $f_c$ <sup>1</sup>. Note that the value of  $k$  has little

<sup>1</sup>This can be easily tested by storing the loop counter “ $p$ ” in the MATLAB code found on the data CD.

effect on computational time. The values can vary from around five iterations for low input signal amplitudes and cut-off frequencies to thirty for high amplitude and large values of  $f_c$  ones. View Tab. 4.3 for a comparison of clock times between Huovilainen’s and the backwards model when filtering a 100 Hz square wave 1 s long. The backwards model is of the order  $10^2$  times slower than Huovilainen’s implementation and in some cases is around 400 times slower.

$f_c$ (Hz)	Input amp.	Filter time (s): Hvlmen’s	Filter time (s): Backwards
100	0.1	0.062	16.658
100	1	0.074	22.030
100	10	0.078	22.833
1000	0.1	0.061	20.561
1000	1	0.079	29.600
1000	10	0.085	31.008
10000	0.1	0.060	24.390
10000	1	0.079	35.789
10000	10	0.085	34.051
20000	0.1	0.061	25.018
20000	1	0.079	37.444
20000	10	0.084	34.692

TABLE 4.3: The time to filter a 100 Hz square wave input signal of 1 s duration varies strongly with cut-off frequency and weakly with input signal amplitude. The backwards model is far slower than Huovilainen’s model.

## 4.4 Audio examples

The accompanying data CD contains five audio examples of sounds processed by the backward difference model of the Moog. The first audio examples “Effects of Resonance” was created by inputting a square wave and gradually ramping the feedback gain up; notice the strong dissonant effect created. The file “Example of Harmonic Distortion” demonstrates the effect of the non-linearity of the filter as the input and output signals are played sequentially. The file “Full Feedback” was created by inputting a recording of a clap to the filter when  $k = 4$ ; the resultant sinusoidal tone was altered in pitch by varying  $f_c$ . The two files “Modulating Cut-off Frequency” and “Modulating Cut-off Frequency 2” are the result of inputting a square wave and varying the value of  $f_c$  over time.

## Chapter 5

# Discussion and conclusions

It can be concluded that it is possible to implement a more accurate version of Huovilainen's non-linear Moog VCF model at the cost of stability and computational expense. Using a centred, backwards difference scheme instead of a forward difference scheme to solve four differential equations, the magnitude error in the frequency response was decreased considerably, especially for high frequencies. Huovilainen's filter is stable for all levels of input signals, feedback gains and values of cut-off frequency from the DC to Nyquist frequency. The backwards difference filter is far less stable: at low level signals the filter will output bound signals while  $f_c$  approaches  $\frac{F_s}{2}$  but as the input signal's amplitude is increased the cut-off frequency must be lowered considerably. Furthermore, the backwards difference implementation is at least 100 times slower than Huovilainen model.

### 5.1 Accuracy

The plots in Sec. 4.1 show approximations to the frequency response of the Moog VCF and this has already been discussed. They also demonstrate the effect of averaging and centring signals. The difference in the magnitude responses seen in Figs. 4.1 and 4.2 is quite remarkable as the only difference between the two models is a two sample average of the feedback signal. The effect of centring an equation is even more so apparent when considering the difference between Figs. 4.4 and 4.5: one of the peaks increases by over 50 dB to match the analogue response. Figs. 4.9 and 4.8 show that for low values of  $f_c$  both models are fairly accurate and that the backwards difference model is considerably more accurate at higher values of  $f_c$ . However, they also show very large spikes of error close to the cut-off frequency. I can not find an explanation for this unusually large error or why it appears in the same place for both models but it is definitely worthy of further

investigation. Generally, the magnitude responses of both models are accurate but the same can not be said for their phase responses.

Figs. 4.1 and 4.2 stress the importance of calculating an accurate phase response. The first plot shows that when  $f_c = 10000$  Hz the small resonant peak is shifted to the right of its desired position and this can be understood by viewing the phase response beneath it. The corresponding phase response curve equals  $-180^\circ$  when  $f \approx 20000$  Hz which is causing the upwards shift of the resonant peak. The more accurate phase response in Fig. 4.2 shifts the resonant peak downwards.

It should be noted that in a number of plots, the magnitude response of Huovilainen's model appear to be vertically offset from the analogue response and that if it were to be moved uni-formally upwards it would result in a more accurate response. The magnitude response was not normalised, not in an attempt to skew results in favour of the backwards difference model but as this scaling was not necessary when considering the backwards difference model and is an example of its improved accuracy.

## 5.2 Stability

Huovilaenen's model is a cascade of four low-pass filters with an embedded non-linear tanh function.[5] I believe that is these tanh functions, characterised by  $\tanh x = 1$  for  $x \rightarrow \infty$ , which keep his filter so stable. I can not explain why the backwards difference filter is so unstable but when calculating the matrix  $\Delta\mathbf{Y}$  using a linear system solve, MATLAB returns the statement that "Matrix is close to singular or badly scaled", meaning that either  $D\mathbf{f}$ ,  $\mathbf{f}$  or  $\Delta\mathbf{Y}$  is singular or very close to being singular.[28]

Without making any changes to the solution methods of the filter, the question of stability is ultimately a balancing act between an increased range of cut-off frequencies or input signal amplitude. A simple resolve of the instability conditions would be to scale the input signal's level appropriately. Tab. 4.2 shows that as the signal level increases from one, to two then four the frequency range drops from around 44, to 40, to 35 kHz. As the ultimate output file is a 44.1 kHz file, perhaps an upper limit of 30 kHz could be set for the cut-off frequency and the input file would be scaled such that it peaks at three.

However, another solution would be to multiply the input signal by a non-linear, possibly tanh-like, function before it reaches the filter. This would in fact make the filter more Moog like. The Moog is quite distinct in that soft clipping happens at the filter input and the clipping itself is filtered along with the signal. Bill Hemsath, one of Moog Music's earliest engineers recalls "Our instrument had punch to it because we inadvertently over

drove the filter like crazy. Jim Scott, did the filter and voltage controlled amplifiers, he made a calculation error, and he over drove the filters by 10, 12, 15 dB, something like that. And nobody knew that until a month or two after we started in production and everybody said, ‘Leave it alone!’.”[29] Whatever the solution, a digital filter must be stable within it’s allowed parameter range and is the cause of further work.

### 5.3 Computational expense

The backwards difference model is of the order  $10^2$  times slower than Huovilaenen’s and as it stands right now is not suitable for real time implementation. However, the script which runs the filter is far from optimized: there are 35 lookups to only nine variables per iteration of Newton-Raphson’s and numerous repetitions of the same calculations. There are also 13 uses of the tanh function which is much slower than a floating point operation; it may be possible to speed the model by using a look up table and an interpolating function to carry out the tanh calculations.

## Chapter 6

# Recommendations

### 6.1 Recommendations

The result of this report can be stated succinctly: it is possible to create a very accurate model of the Moog VCF but at considerable costs to either cut-off frequency range or input signal amplitude and computational expense. Any further investigations should not seek to improve the accuracy of the model but focus on the problems of stability and computational time. Although one of the project aims was to work at a sample rate of 88.2 kHz only, it may prove that further over-sampling may increase the stability of the model and is a possible area of study. It has already been noted that the code is far from optimal as it stands and that many calculations and operations can be consolidated but it may be beneficial to use another iterative solution method. Halley's method is a Householder's method like NR's method. It is of the second order and involves more calculations than NR's method but it converges towards a solution cubically not quadratically.[26] As such, it may result in less computation time. Finally, with a focus on creating a digital musical instrument, it may be desirable to implement a scalable gain stage in the filter's pass band to account for the attenuation as  $k$  increases. Another possible development may be to include non-linear functions that bring about even or even-odd types of harmonic distortion.

## Appendix A

# Tuning Huovilainen's Moog VCF filter

User input $F_c$ (Hz)	Tuned $F_c$ (Hz)
1000	1001
2000	2004
3000	3010
4000	4030
5000	5055
6000	6100
7000	7160
8000	8240
9000	9350
10000	10500
11000	11700
12000	12950
13000	14300
14000	15700
15000	17300
16000	19200
16500	20400

TABLE A.1: This look up tuning table improves the accuracy of Huovilainen's unit-and-a-half delay model when  $F_s = 88.2$  kHz. The left hand column refers to the frequency of the resonant peak in the analogue magnitude response and the right hand column refers to the corresponding peak in Huovilainen's model.

# Bibliography

- [1] Moog, R. *A voltage controlled low-pass high-pass filter for audio processing*. **Audio Engineering Society**, 1965
- [2] Sound On Sound, *Product Review - Moog The Ladder*, <http://www.soundonsound.com/sos/jun12/articles/moog-ladder.htm> (Viewed on: 16<sup>th</sup> August, 2012)
- [3] Stilson, T. and Smith, J., *Analyzing the Moog VCF with Considerations for Digital Implementation*, **International Computer Music Conference, Hong Kong** (1996)
- [4] Rossum, D., *Making digital filters sound “analog”*, **E-mu Systems, Inc.** (1992)
- [5] *Non-linear digital implementation of the Moog ladder filter*, **Proc. of the 7<sup>th</sup> Int. Conference on Digital Audio Effects (DAFx-04)** (October,2004)
- [6] Hèlie, T., *On the use of Volterra series for real-time simulations of weakly non-linear analog audio devices: application to the Moog ladder filter*, **Proc. of the 9<sup>th</sup> Int. Conference on Digital Audio Effects (DAFx-06)** (September,2006)
- [7] Fontana, F. *Preserving the structure of the Moog VCF in the digital domain* **International Computer Music, ICM’07** (2007)
- [8] Stinchcombe, T. *Analysis of the Moog transistor ladder and filter derivative filters*, 2008 [http://www.timstinchcombe.co.uk/synth/Moog\\_ladder\\_tf.pdf](http://www.timstinchcombe.co.uk/synth/Moog_ladder_tf.pdf) (Viewed on: 22<sup>nd</sup> March, 2012)
- [9] Palance, J. *The Minimoog synthesizer: operation manual*, [www.fantasyjackpalance.com](http://www.fantasyjackpalance.com) (Viewed on: 22<sup>nd</sup> March, 2012)
- [10] Rizzoni, G. *Principles and applications of electrical engineering*. International edition. New York City: McGraw-Hill, 2003.



- [11] Lesure, J. *Differential amplifiers and current source*. [http://www.st-andrews.ac.uk/~jcgl/Scots\\_Guide/audio/part1/page3.html](http://www.st-andrews.ac.uk/~jcgl/Scots_Guide/audio/part1/page3.html) (Viewed on: 16<sup>th</sup> August, 2012)
- [12] Hasler, P. *Basics of transconductance: capacitance filters* [http://www.ece.gatech.edu/academic/courses/ece6414/References/Hasler\\_gmCFilter01.pdf](http://www.ece.gatech.edu/academic/courses/ece6414/References/Hasler_gmCFilter01.pdf) (Viewed on: 16<sup>th</sup> August, 2012)
- [13] Carlson, A., Gisser, D. *Electrical engineering: concepts and applications*. World student series edition. Massachusetts: Addison-Wesley, 1981.
- [14] Smith, S., *The scientist & engineer's guide to Digital Signal Processing*, **California Technical Pub.** (1997)
- [15] Park, T. *Introduction to digital signal processing: computer musically speaking*, **World Scientific Publishing, Co., Singapore** (2010)
- [16] Wolfram Mathworld. *Euler Forward Method*.<http://mathworld.wolfram.com/EulerForwardMethod.html> (Viewed on: 16<sup>th</sup> August, 2012)
- [17] Smith, J. *Impulse Invariant Method*, 2010. [https://ccrma.stanford.edu/~jos/pasp/Impulse\\_Invariant\\_Method.html](https://ccrma.stanford.edu/~jos/pasp/Impulse_Invariant_Method.html) (Viewed on: 16<sup>th</sup> August, 2012)
- [18] Rahkonen, T. *Analysis of analog circuits using Volterra series*, 1999. [http://www.ee.oulu.fi/~timor/DCourse/Chp1\\_2.pdf](http://www.ee.oulu.fi/~timor/DCourse/Chp1_2.pdf) (Viewed on: 16<sup>th</sup> August, 2012)
- [19] Zölzer, U. *DAFX: Digital audio effects*, Second edition. New Jersey: John Wiley and sons (2011)
- [20] Roberts, S. *Design of digital filters*, 2000. <http://www.robots.ox.ac.uk/~sjrob/Teaching/SP/16.pdf> (Viewed on: 12<sup>th</sup> August, 2012)
- [21] Smith, J. *Frequency Warping*, 2012. [https://ccrma.stanford.edu/~jos/fp/Frequency\\_Warping.html](https://ccrma.stanford.edu/~jos/fp/Frequency_Warping.html) (Viewed on: 12<sup>th</sup> August, 2012)
- [22] McClellan, J., Schafer, R. and Yoder M. *Signal processing first*, First edition. New Jersey: Prentice Hall, (2003)
- [23] Bilbao, S. *Numerical sound synthesis*, First edition. New Jersey: John Wiley and sons, 2009.
- [24] Wolfram Mathworld. *Implicit Function*. <http://mathworld.wolfram.com/ImplicitFunction.html> (Viewed on: 12<sup>th</sup> August, 2012)

- [25] Smith, J. *Definition of the Simplest Low-Pass*, 2009. [https://ccrma.stanford.edu/~jos/fp/Definition\\_Simplest\\_Low\\_Pass.html](https://ccrma.stanford.edu/~jos/fp/Definition_Simplest_Low_Pass.html) (Viewed on: 16<sup>th</sup> August, 2012)
- [26] Bence, S., Hobson, M. and Riley, K. *Mathematical methods for physics and engineering*, Fifth printing. New York: Cambridge University press, 2009.
- [27] Rane. *Pro-audio reference*. [http://www.rane.com/par-d.html#0\\_dBFS](http://www.rane.com/par-d.html#0_dBFS) (Viewed on: 17<sup>th</sup> August, 2012)
- [28] Mathworks. *Why do I get warning message "Matrix is close to singular or badly scaled."?*, 2012. <http://www.mathworks.co.uk/support/solutions/en/data/1-FA9A48/index.html?solution=1-FA9A48> (Viewed on: 17<sup>th</sup> August, 2012)
- [29] Moog Music Inc., *A brief history of the Minimoog, 1970s* (2010), [http://www.youtube.com/watch?v=sLx\\_x5Fuzp4&list=PL6664A2B79B2BF3A0&index=1&feature=plpp\\_video](http://www.youtube.com/watch?v=sLx_x5Fuzp4&list=PL6664A2B79B2BF3A0&index=1&feature=plpp_video) (Online video.) (Last accessed 5<sup>th</sup> April, 2012)