UNIVERSITY OF MIAMI

ALIASING AND HARMONIC DECAY CONTROL OF THE BANDLIMITED STEP METHOD USING PSYCHOACOUSTIC MODELS AND THE GENETIC ALGORITHM

By

Francisco J. Valencia Otalvaro

A THESIS PROJECT

Submitted to the Faculty
of the University of Miami
in partial fulfillment of the requirements for
the degree of Master of Science in Music Engineering Technology

Coral Gables, Florida

April 17, 2015
UNIVERSITY OF MIAMI

A Thesis Project submitted in partial fulfillment of
the requirements for the degree of
Master of Science in Music Engineering Technology

ALIASING AND HARMONIC DECAY CONTROL OF THE BANDLIMITED
STEP METHOD USING PSYCHOACOUSTIC MODELS AND THE GENETIC
ALGORITHM

Francisco J. Valencia Otalvaro

Approved:

Prof. William C. Pirkle
Assistant Professor of Music Engineering

Dr. Shannon de l’Etoile
Associate Dean of Graduate Studies

Dr. Colby N. Leider
Associate Professor of Music Engineering

Dr. Michael Scordilis
Associate Professor of Electrical Engineering
The issue of reducing or completely eliminating the aliasing caused by the discontinuities found in trivial pitched oscillators used in Virtual Analog (VA) synthesis has brought a lot of research. Standard methods, commonly referred as Ideally Bandlimited, such as: additive synthesis, Discrete Summation Formulae (DSF), and Feedback Delay Loop (FDL), are very expensive computationally. In order to have a implementation that is real time efficient, other algorithms have been developed that use less computational resources but allow some aliasing, specially at high frequencies. The cost of reducing the aliasing is seen at the harmonic envelope, losing amplitude mainly at higher frequencies near the Nyquist limit. Such methods are categorized as Quasi-Bandlimited Algorithms and include: the Bandlimited Impulse Trains (BLIT) and the Bandlimited Step Functions (BLEP), the latter being the focus of study in this document. Most recent methods allow aliasing in the complete audible band, but it is suppressed in comparison to the trivial methods. These are referred as Alias-Suppressing methods and include: the Differentiated Parabolic Waveform (DPW) and the Polynomial Transition Regions (PTR). These last two categories take advantage of the human hearing threshold and the effect of frequency masking to hide the aliasing from the human ear.

It has been established that even though the BLEP method is more expensive computationally than the other methods available, due to its look up table implementation, it has the best performance in terms of sound quality. The purpose of this research is to find, using a genetic algorithm that has a perceptual model as feedback, an optimal input for the BLEP method based on the $sinc$ function and its applied window (Hanning, Hamming, Blackman, Blackman-Harris, Welch, etc.). The goals are: first, to eliminate the most aliasing overall while keeping a fairly good harmonic envelope. And second, to optimize the size of the table and the number of BLEP points of correction.
ACKNOWLEDGMENTS

Thanks to my advisor Will for his incredible guidance, advice and of course the original idea for this research. It is much easier to be interested and enjoy a subject when your teacher is so passionate about it and dedicated to it. Your way of teaching is inspiring and I hope it continues making this program so great and outstanding for many years.

Thanks to my committee member and department director Colby for his support and input, not only during this research, but during the entire program. These two years of learning have been an incredible and unforgettable experience. Your enthusiasm for music and engineering has made this department one of the best learning places in the world.

Thanks to professor Scordillis, my other committee member, and Fabian at Aalto University for their experienced input in this research.

Thanks to my friends and family — especially my mom, dad, and nana — for their incredible constant support during all the long days and more importantly, all the long nights; not only during this research, but all the years of my career. This is the culmination of almost 20 years of studies, or at least a pause, during which the experience and engineering skills of my father and the flawless ethic and teaching capabilities of my mother guided me and helped me achieve all my goals.
DEDICATION

To my family. Because they are the most important people in life.
**PREFACE**

But other paths also led to the same stochastic crossroads—first of all, natural events such as the collision of hail or rain with hard surfaces, or the song of cicadas in a summer field. These sonic events are made out of thousands of isolated sounds; this multitude of sounds, seen as a totality, is a new sonic event. This mass event is articulated and forms a plastic mold of time, which itself follows aleatory and stochastic laws... The statistical laws of these events, separated from their political or moral context... are the laws of the passage from complete order to total disorder in a continuous or explosive manner. They are stochastic laws. [Xenakis, 1992]
TABLE OF CONTENTS

ACKNOWLEDGMENTS .................................................. iii
DEDICATION ............................................................. iv
PREFACE ................................................................. v
LIST OF TABLES ......................................................... xii
LIST OF FIGURES ......................................................... xiii

CHAPTER

1 Introduction ......................................................... 1

2 Theoretical Background ............................................. 6
   2.1 Virtual Analog Synthesis ....................................... 6
      2.1.1 Subtractive Synthesis ................................... 6
      2.1.2 Issues ..................................................... 7
   2.2 Source Signal Generation ....................................... 7
      2.2.1 Aliasing .................................................... 8
      2.2.2 Masking and Hearing Threshold ......................... 9
   2.3 Types of Aliasing Reduction Methods ......................... 11
      2.3.1 Ideally Bandlimited Algorithms ....................... 11
      2.3.2 Quasi-Bandlimited Algorithms ....................... 13
      2.3.3 Alias-Suppressing Algorithms ....................... 23
      2.3.4 Ad-hoc Methods ....................................... 26
   2.4 Comparison of Most Relevant Methods ....................... 26
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.5</td>
<td>Perceptual Evaluation of Antialiasing Oscillator Algorithms</td>
<td>29</td>
</tr>
<tr>
<td>2.5.1</td>
<td>Samples Used for Testing</td>
<td>31</td>
</tr>
<tr>
<td>2.5.2</td>
<td>Listening Test</td>
<td>31</td>
</tr>
<tr>
<td>2.5.3</td>
<td>Results</td>
<td>35</td>
</tr>
<tr>
<td>2.5.4</td>
<td>Conclusion</td>
<td>39</td>
</tr>
<tr>
<td>2.6</td>
<td>Choosing the BLEP Method as Focus of Research</td>
<td>40</td>
</tr>
<tr>
<td>2.7</td>
<td>Restrictions in BLEP Documentation</td>
<td>41</td>
</tr>
<tr>
<td>3</td>
<td>Proposed Method</td>
<td>42</td>
</tr>
<tr>
<td>3.1</td>
<td>Description and Overview</td>
<td>42</td>
</tr>
<tr>
<td>3.1.1</td>
<td>Block diagram</td>
<td>42</td>
</tr>
<tr>
<td>3.2</td>
<td>Input for the BLEP Method</td>
<td>43</td>
</tr>
<tr>
<td>3.2.1</td>
<td>The Sinc Function</td>
<td>44</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Windowing</td>
<td>46</td>
</tr>
<tr>
<td>3.3</td>
<td>Creation of the BLEP Table</td>
<td>48</td>
</tr>
<tr>
<td>3.4</td>
<td>Implementation of the Trivial Waveform</td>
<td>52</td>
</tr>
<tr>
<td>3.4.1</td>
<td>Issues at High Fundamental Frequencies</td>
<td>53</td>
</tr>
<tr>
<td>3.5</td>
<td>Applying BLEP to the Trivial Waveform</td>
<td>54</td>
</tr>
<tr>
<td>3.5.1</td>
<td>Number of Zero Crossings vs. Number of Correction Points Paradigm</td>
<td>57</td>
</tr>
<tr>
<td>3.6</td>
<td>DFT for Oscillator Algorithms</td>
<td>59</td>
</tr>
<tr>
<td>3.6.1</td>
<td>FFT Size and bins</td>
<td>61</td>
</tr>
<tr>
<td>3.6.2</td>
<td>Choosing the Right Fundamental Frequencies</td>
<td>63</td>
</tr>
<tr>
<td>3.6.3</td>
<td>Finding Harmonics and Aliasing Peaks</td>
<td>65</td>
</tr>
<tr>
<td>3.6.4</td>
<td>The Harmonic Envelope Decay</td>
<td>67</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>----------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>3.7</td>
<td>Additive Synthesis</td>
<td>69</td>
</tr>
<tr>
<td>3.7.1</td>
<td>Additive Synthesis based on Trivial Waveform Harmonics</td>
<td>71</td>
</tr>
<tr>
<td>3.7.2</td>
<td>Additive Synthesis based on BLEP Waveform Harmonics</td>
<td>73</td>
</tr>
<tr>
<td>3.8</td>
<td>Time Domain Analysis</td>
<td>74</td>
</tr>
<tr>
<td>3.8.1</td>
<td>BLEP and Additive Synthesis Similarities</td>
<td>76</td>
</tr>
<tr>
<td>3.9</td>
<td>Analysis of Only the Harmonic Envelope</td>
<td>76</td>
</tr>
<tr>
<td>3.9.1</td>
<td>Maximum Harmonic Absolute Error (MaxAbsHarmErr)</td>
<td>77</td>
</tr>
<tr>
<td>3.9.2</td>
<td>Root Mean Squared Error (RMSE)</td>
<td>78</td>
</tr>
<tr>
<td>3.10</td>
<td>Error in Harmonics and Aliasing Amplitudes (A-Weigthed)</td>
<td>79</td>
</tr>
<tr>
<td>3.10.1</td>
<td>Aliasing Only</td>
<td>79</td>
</tr>
<tr>
<td>3.10.2</td>
<td>Harmonic Decay Only</td>
<td>80</td>
</tr>
<tr>
<td>3.11</td>
<td>Hearing Threshold and Spread Masking (Bosi)</td>
<td>80</td>
</tr>
<tr>
<td>3.12</td>
<td>Noise-to-Mask Ratio (NMR)</td>
<td>85</td>
</tr>
<tr>
<td>3.13</td>
<td>Perceptual Evaluation of Audio Quality (PEAQ)</td>
<td>87</td>
</tr>
<tr>
<td>3.14</td>
<td>Comparison of Analysis Strategies</td>
<td>88</td>
</tr>
<tr>
<td>3.15</td>
<td>Genetic Algorithm for Obtaining the Optimal BLEP Residual</td>
<td>89</td>
</tr>
<tr>
<td>3.15.1</td>
<td>Genetic Algorithm (GA) Overview</td>
<td>89</td>
</tr>
<tr>
<td>3.15.2</td>
<td>Fitness Function</td>
<td>90</td>
</tr>
<tr>
<td>3.15.3</td>
<td>Input Function Creation</td>
<td>94</td>
</tr>
<tr>
<td>3.15.4</td>
<td>Bounds</td>
<td>100</td>
</tr>
<tr>
<td>3.15.5</td>
<td>Restrictions</td>
<td>102</td>
</tr>
<tr>
<td>3.15.6</td>
<td>Obtaining the Results for Regular Windows (Starting Point)</td>
<td>104</td>
</tr>
</tbody>
</table>
4 Evaluation  
4.1 Results from GA Phase 0 (Pre-Processing)  
4.2 Results from GA Phase I and Phase II (Aliasing Optimization Only)  
4.2.1 Phase I  
4.2.2 Phase II  
4.2.3 Input Functions Obtained  
4.2.4 Trade off between Aliasing and Harmonic Decay  
4.2.5 BLEP Table for Perceptually Alias-Free MIDI Keyboard (Piano)  
4.2.6 Further Analysis of Results  
4.2.7 Comparison of Input Functions with Equal Bosi Scores  
4.3 Listening Test  
4.3.1 Test Conditions and subjects  
4.3.2 Samples to Evaluate and Scenarios  
4.3.3 Normalization of the Loudness of the Samples  
4.3.4 Results  
4.4 Results from GA Phase III (Aliasing and Harmonic Decay Optimization)  
5 Discussion  
5.1 Findings in Computational and Perceptual Results  
5.1.1 Type of Input Functions Found with GA Phase II
<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Comparison of Oscillator Algorithms in Terms of Quality</td>
<td>27</td>
</tr>
<tr>
<td>2</td>
<td>Comparison of Oscillator Algorithms in Terms of Highest Alias-Free Fundamental Frequency</td>
<td>28</td>
</tr>
<tr>
<td>3</td>
<td>Grading Scale for the ABC/HR Listening Test</td>
<td>33</td>
</tr>
<tr>
<td>4</td>
<td>MIDI Table for Piano Keyboard of 88 Notes</td>
<td>64</td>
</tr>
<tr>
<td>5</td>
<td>Grading Scale for the PEAQ Objective Difference Grade (ODG)</td>
<td>88</td>
</tr>
<tr>
<td>6</td>
<td>Comparison of Evaluation Strategies</td>
<td>89</td>
</tr>
<tr>
<td>7</td>
<td>List of Window Pool</td>
<td>96</td>
</tr>
<tr>
<td>8</td>
<td>Specifications of the Variables of the GA</td>
<td>101</td>
</tr>
<tr>
<td>9</td>
<td>Highest Fundamental Frequency Perceptually Alias-Free of Regular Windows</td>
<td>112</td>
</tr>
<tr>
<td>10</td>
<td>Highest Fundamental Frequency Perceptually Alias-Free Comparison</td>
<td>117</td>
</tr>
<tr>
<td>11</td>
<td>Factor Coefficients for Input Functions Results of GA</td>
<td>119</td>
</tr>
<tr>
<td>12</td>
<td>Power Coefficients for Input Functions Results of GA</td>
<td>120</td>
</tr>
<tr>
<td>13</td>
<td>Samples Used in Listening Test</td>
<td>173</td>
</tr>
<tr>
<td>14</td>
<td>List of Scenarios in Listening Test</td>
<td>174</td>
</tr>
<tr>
<td>15</td>
<td>Ranking of Results of Listening Test</td>
<td>183</td>
</tr>
<tr>
<td>16</td>
<td>Highest Fundamental Frequency Perceptually Alias-Free Comparison (Phase III)</td>
<td>184</td>
</tr>
<tr>
<td>17</td>
<td>Factor Coefficients for Input Functions Results of GA (Phase III)</td>
<td>184</td>
</tr>
<tr>
<td>18</td>
<td>Power Coefficients for Input Functions Results of GA (Phase III)</td>
<td>184</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Aliasing in the Reverse Sawtooth Waveform</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>Unipolar Impulse Train for BLIT</td>
<td>15</td>
</tr>
<tr>
<td>3</td>
<td>Block Diagram for BLIT</td>
<td>16</td>
</tr>
<tr>
<td>4</td>
<td>Bandlimited Step Function (BLEP)</td>
<td>19</td>
</tr>
<tr>
<td>5</td>
<td>Bandlimited Step Function (BLEP) Residuals</td>
<td>20</td>
</tr>
<tr>
<td>6</td>
<td>PolyBLEP Residuals</td>
<td>21</td>
</tr>
<tr>
<td>7</td>
<td>Block Diagram for DPW</td>
<td>25</td>
</tr>
<tr>
<td>8</td>
<td>Comparison of Several Oscillator Algorithms</td>
<td>29</td>
</tr>
<tr>
<td>9</td>
<td>Moog sawtooth</td>
<td>31</td>
</tr>
<tr>
<td>10</td>
<td>Listening Test for Ideal Bandlimited Focused Results</td>
<td>36</td>
</tr>
<tr>
<td>11</td>
<td>Listening Test for BLEP Focused Results</td>
<td>38</td>
</tr>
<tr>
<td>12</td>
<td>Block Diagram of Proposed Method</td>
<td>42</td>
</tr>
<tr>
<td>13</td>
<td>Impulse Response of LPF</td>
<td>44</td>
</tr>
<tr>
<td>14</td>
<td>Input for the BLEP Method</td>
<td>44</td>
</tr>
<tr>
<td>15</td>
<td>The sinc Function</td>
<td>45</td>
</tr>
<tr>
<td>16</td>
<td>The Blackman Window</td>
<td>46</td>
</tr>
<tr>
<td>17</td>
<td>Windowed sinc Function</td>
<td>47</td>
</tr>
<tr>
<td>18</td>
<td>Integral of the Windowed sinc Function</td>
<td>49</td>
</tr>
<tr>
<td>19</td>
<td>BLEP Residual Table</td>
<td>50</td>
</tr>
<tr>
<td>20</td>
<td>Inverted BLEP Residual Table</td>
<td>50</td>
</tr>
<tr>
<td>21</td>
<td>Block Diagram for Creation of BLEP Table</td>
<td>51</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td>------</td>
</tr>
<tr>
<td>22</td>
<td>Sawtooth Waveform implemented with the Modulo</td>
<td>53</td>
</tr>
<tr>
<td>23</td>
<td>Trivial Sawtooth Waveform implemented with the Modulo (Peak Amplitude Issues)</td>
<td>54</td>
</tr>
<tr>
<td>24</td>
<td>BLEP Sawtooth Waveform</td>
<td>56</td>
</tr>
<tr>
<td>25</td>
<td>KORG BLEP Illustration</td>
<td>58</td>
</tr>
<tr>
<td>26</td>
<td>Trivial Sawtooth FFT with Lobing</td>
<td>60</td>
</tr>
<tr>
<td>27</td>
<td>Trivial Sawtooth FFT</td>
<td>61</td>
</tr>
<tr>
<td>28</td>
<td>BLEP Sawtooth FFT</td>
<td>62</td>
</tr>
<tr>
<td>29</td>
<td>Trivial Sawtooth FFT with Specified Aliasing and Harmonics</td>
<td>65</td>
</tr>
<tr>
<td>30</td>
<td>BLEP Sawtooth FFT Specified Aliasing and Harmonics</td>
<td>66</td>
</tr>
<tr>
<td>31</td>
<td>Trivial vs. BLEP Sawtooth FFT (Blackman) - Higher Amplitude in Trivial</td>
<td>68</td>
</tr>
<tr>
<td>32</td>
<td>Trivial vs. BLEP Sawtooth FFT (Narrower Win.) - Higher Amplitude in BLEP</td>
<td>69</td>
</tr>
<tr>
<td>33</td>
<td>BLEP Sawtooth with Additive Synthesis based on Trivial Amplitudes</td>
<td>71</td>
</tr>
<tr>
<td>34</td>
<td>BLEP Sawtooth FFT with Additive Synthesis based on Trivial Amplitudes</td>
<td>72</td>
</tr>
<tr>
<td>35</td>
<td>BLEP Sawtooth with Additive Synthesis based on BLEP Amplitudes</td>
<td>73</td>
</tr>
<tr>
<td>36</td>
<td>BLEP Sawtooth FFT with Additive Synthesis based on BLEP Amplitudes</td>
<td>74</td>
</tr>
<tr>
<td>37</td>
<td>Time Domain Analysis</td>
<td>75</td>
</tr>
<tr>
<td>38</td>
<td>Time Domain Analysis</td>
<td>77</td>
</tr>
<tr>
<td>39</td>
<td>Frequency Domain Analysis</td>
<td>78</td>
</tr>
<tr>
<td>40</td>
<td>Bosi Method in General</td>
<td>82</td>
</tr>
<tr>
<td>Figure</td>
<td>Page</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>83</td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>84</td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>97</td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>97</td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>98</td>
<td></td>
</tr>
<tr>
<td>46</td>
<td>98</td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>114</td>
<td></td>
</tr>
<tr>
<td>48</td>
<td>121</td>
<td></td>
</tr>
<tr>
<td>49</td>
<td>121</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>122</td>
<td></td>
</tr>
<tr>
<td>51</td>
<td>123</td>
<td></td>
</tr>
<tr>
<td>52</td>
<td>124</td>
<td></td>
</tr>
<tr>
<td>53</td>
<td>124</td>
<td></td>
</tr>
<tr>
<td>54</td>
<td>125</td>
<td></td>
</tr>
<tr>
<td>55</td>
<td>126</td>
<td></td>
</tr>
<tr>
<td>56</td>
<td>127</td>
<td></td>
</tr>
<tr>
<td>57</td>
<td>127</td>
<td></td>
</tr>
<tr>
<td>58</td>
<td>128</td>
<td></td>
</tr>
<tr>
<td>59</td>
<td>129</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>130</td>
<td></td>
</tr>
<tr>
<td>61</td>
<td>130</td>
<td></td>
</tr>
<tr>
<td>62</td>
<td>131</td>
<td></td>
</tr>
<tr>
<td>63</td>
<td>132</td>
<td></td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>64</td>
<td>Optimal Input Functions Optimized for Harmonic Decay</td>
<td>134</td>
</tr>
<tr>
<td>65</td>
<td>Optimal BLEP Residuals Optimized for Harmonic Decay</td>
<td>135</td>
</tr>
<tr>
<td>66</td>
<td>NMR (Trade Off)</td>
<td>137</td>
</tr>
<tr>
<td>67</td>
<td>A-Weighted (Aliasing) Error (Trade Off)</td>
<td>138</td>
</tr>
<tr>
<td>68</td>
<td>RMSE (Trade Off)</td>
<td>139</td>
</tr>
<tr>
<td>69</td>
<td>Maximum Absolute Harmonic Error (Trade Off)</td>
<td>140</td>
</tr>
<tr>
<td>70</td>
<td>A-Weighted (Harmonics) Error (Trade Off)</td>
<td>141</td>
</tr>
<tr>
<td>71</td>
<td>NMR Evaluation</td>
<td>147</td>
</tr>
<tr>
<td>72</td>
<td>NMR Evaluation (MIDI Keyboard Case)</td>
<td>149</td>
</tr>
<tr>
<td>73</td>
<td>PEAQ Evaluation</td>
<td>150</td>
</tr>
<tr>
<td>74</td>
<td>PEAQ Evaluation (MIDI Keyboard Case)</td>
<td>152</td>
</tr>
<tr>
<td>75</td>
<td>A-Weighted (Aliasing Only) Evaluation</td>
<td>153</td>
</tr>
<tr>
<td>76</td>
<td>A-Weighted (Aliasing Only) Evaluation (MIDI Keyboard Case)</td>
<td>154</td>
</tr>
<tr>
<td>77</td>
<td>RMSE Evaluation</td>
<td>156</td>
</tr>
<tr>
<td>78</td>
<td>RMSE Evaluation (MIDI Keyboard Case)</td>
<td>157</td>
</tr>
<tr>
<td>79</td>
<td>Maximum Absolute Harmonic Error Evaluation</td>
<td>158</td>
</tr>
<tr>
<td>80</td>
<td>Maximum Absolute Harmonic Error Evaluation (MIDI Keyboard Case)</td>
<td>159</td>
</tr>
<tr>
<td>81</td>
<td>A-Weighted (Only Harmonics) Evaluation</td>
<td>160</td>
</tr>
<tr>
<td>82</td>
<td>A-Weighted (Only Harmonics) Evaluation (MIDI Keyboard Case)</td>
<td>161</td>
</tr>
<tr>
<td>83</td>
<td>NMR (Trivial Reference) Evaluation (Optimized for Aliasing)</td>
<td>163</td>
</tr>
<tr>
<td>84</td>
<td>NMR (Trivial Reference) Evaluation (Optimized for Harmonic Decay)</td>
<td>164</td>
</tr>
<tr>
<td>85</td>
<td>PEAQ (Trivial Reference) Evaluation (Optimized for Aliasing)</td>
<td>166</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>86</td>
<td>PEAQ (Trivial Reference) Evaluation (Optimized for Harmonic Decay)</td>
<td>167</td>
</tr>
<tr>
<td>87</td>
<td>NMR Evaluation (Equal Bosi Output)</td>
<td>168</td>
</tr>
<tr>
<td>88</td>
<td>Maximum Absolute Harmonic Error Evaluation (Equal Bosi Output)</td>
<td>169</td>
</tr>
<tr>
<td>89</td>
<td>Hidden Reference Absolute Statistics</td>
<td>176</td>
</tr>
<tr>
<td>90</td>
<td>Hidden Reference Absolute Comparison</td>
<td>177</td>
</tr>
<tr>
<td>91</td>
<td>Test Object Absolute Statistics</td>
<td>178</td>
</tr>
<tr>
<td>92</td>
<td>Test Object Absolute Comparison</td>
<td>179</td>
</tr>
<tr>
<td>93</td>
<td>Difference Grade Statistics</td>
<td>180</td>
</tr>
<tr>
<td>94</td>
<td>Difference Grade Comparison</td>
<td>181</td>
</tr>
<tr>
<td>95</td>
<td>Optimal Input Function for GA Phase III</td>
<td>185</td>
</tr>
<tr>
<td>96</td>
<td>Optimal BLEP Residual for GA Phase III</td>
<td>186</td>
</tr>
<tr>
<td>97</td>
<td>NMR Evaluation for GA Phase III</td>
<td>187</td>
</tr>
<tr>
<td>98</td>
<td>PEAQ Evaluation for GA Phase III</td>
<td>188</td>
</tr>
<tr>
<td>99</td>
<td>RMSE Evaluation for GA Phase III</td>
<td>189</td>
</tr>
<tr>
<td>100</td>
<td>Maximum Absolute Harmonic Error Evaluation for GA Phase III</td>
<td>190</td>
</tr>
<tr>
<td>101</td>
<td>Comparison of Best Inputs Windowing for Aliasing</td>
<td>193</td>
</tr>
<tr>
<td>102</td>
<td>Comparison of Best Inputs Windowing for Harmonic Decay</td>
<td>195</td>
</tr>
<tr>
<td>103</td>
<td>Comparison of Best Inputs Windowing for Aliasing and Harmonic Decay</td>
<td>196</td>
</tr>
<tr>
<td>104</td>
<td>Bosi Output for Listening Test Results (GA 8)</td>
<td>200</td>
</tr>
<tr>
<td>105</td>
<td>Bosi Output for Listening Test Results (GA 13)</td>
<td>201</td>
</tr>
<tr>
<td>106</td>
<td>Bosi Output for Blackman (Aliasing Distribution)</td>
<td>203</td>
</tr>
<tr>
<td>107</td>
<td>Bosi Output for GA (Aliasing Distribution)</td>
<td>204</td>
</tr>
</tbody>
</table>
1

Introduction

Virtual Analog (VA) synthesis has been an active field of study and research for the last couple of decades. The idea of the virtualization of many, even all, real synthesizers is appealing to any member of the music industry. Nowadays, plugins are a great percentage of the instruments or tools used in making music. Therefore, having the possibility to replicate almost exact sounds from famous synthesizers is of great value to all interested parties.

Developing programming code based oscillators is not a simple task. Mathematically perfect implementations, know as the trivial implementations, contain discontinuities in the waveform, which in the frequency domain, generate horrible aliasing that increases with frequency. Antialiasing oscillator algorithms have been an important research field in VA synthesis. Several methods have been used in order to smooth the discontinuities found in trivial oscillators.

The general solution to this problem is additive synthesis, but is very computationally expensive and because of this it cannot be implemented in real time. Additive synthesis uses a sum of sinusoids of proper harmonic frequencies and amplitudes, resulting in a waveform with a perfect frequency spectrum, only the harmonics are present with no aliasing. Along with additive synthesis, in the category known as Ideally Bandlimited methods are the Discrete Summation Formulae (DSF) and the Feedback Delay Loop (FDL), also with high
computational costs. Other more efficient approximations have been developed based on subtractive synthesis which use less computational resources, in order to be able to have a real time implementation. The first category is referred as Quasi-Bandlimited oscillators [Välimäki and Huovilainen, 2007]. These methods allow aliasing at certain frequencies, where human hearing is less sensitive to aliasing distortion, mainly at high frequencies [Pekonen and Holters, 2012]. This category includes: the Bandlimited Impulse Train (BLIT) and the Bandlimited Step Function (BLEP); the latter being the focus of this research. More recently, a second category has been developed know as Alias-Suppressing methods, which using some strategies, allow aliasing in the whole audible band, but it is suppressed in comparison to the trivial method. These methods have even lower computational costs than the previous category. The Differentiated Parabolic Waveform (DPW) and the Polynomial Transition Regions (PTR) are the most renounced examples. Apart from these three main categories, approaches made to meet specific needs have also been developed, known as ad-hoc methods. A further description of all of the mentioned methods above is given in the background chapter. The reason for this was to have the most VA oscillator algorithms information documented in just one research. It also includes a listening test that was conducted previous this research to corroborate the results found in literature on actual human subjects when comparing the different methods, but mainly focused on the BLEP method performance.

The Quasi-Bandlimited and the Alias-Suppressing methods have in
common that they both allow some aliasing, while benefiting of the effects of frequency masking and the hearing threshold of the human ear. These two effects combine, allowing aliasing over the entire audible range, specifically at higher frequencies, that is not perceived by the human hearing system. These methods are not considered Alias-Free like the Ideally Bandlimited category, but rather Perceptually Alias-Free.

The BLEP method is currently used in the many designs of VA synthesis pitched oscillators. Out of all the methods mentioned before, BLEP is the one that shows a greater performance in terms of quality of the synthesized sound, making it the closest Quasi-Bandlimited method to the Ideally Bandlimited category. This is the main reason for which the BLEP method has been chosen as the focus of this research. Giving the great advances in memory and processing power in processors, chips and consumer computers, the computational load is not such a negative factor for this method anymore. A very detailed description on the method is given in the Proposed Method chapter. The reason for this is that since its introduction [Brandt, 2001], not many details have been documented on how the regular BLEP method is implemented and how all the different variables affect the output waveform in terms of frequency domain performance.

The original idea was to use a \textit{sinc} function as the BLEP input. This is not the most efficient solution because it still allows a lot of aliasing and introduces a significant loss of gain in the harmonic peaks at high frequencies.
The improved idea was to apply a window before inputing the function. If the sinc is windowed by a common symmetric window (i.e. Blackman, Hamming, Hanning, Blackman-Harris, Welch, etc.), the results improve considerably, but clearly, there is room for better and optimized results. The idea developed in this thesis is to find an optimal input function based on the windowed sinc for the BLEP method, which surpasses the performance of previous combinations in both aliasing and harmonic envelope. The solution will also be optimized to obtain the best results in terms of the size of the look up table and how many times it has to be used, meaning how many samples are corrected by the method.

The search for the desired input function will be through a genetic algorithm. The algorithm will have a number of variables that modify the input function and a fitness function to evaluate the modified input. This fitness function will be based in a perceptual evaluation developed by [Bosi, 2003], which analyzes the waveform in the frequency domain and compares it against the effects of frequency masking and hearing threshold for a specific fundamental frequency. This not only finds the aliasing peaks that surpass the limit, but also the harmonic peaks that fall below the limit and are lost due to the inevitable introduced harmonic decay. The goal is to obtain a BLEP table which is perceptually alias-free at certain frequency and below. The desired frequency is the highest note in the piano, so that a VA synthesizer can be implemented with no alias in all its range. The starting point of optimization is the windowed sinc using Blackman, which currently delivers the best performance out of the BLEP
method. There are three different ways to optimize and eventually solve this problem: improving only aliasing, improving only harmonic decay, or improving both. These three options are all considered in this research. The ideal solution would be to improve the two issues at the same time in order to get a perfect BLEP table for implementing the oscillator.

Different options will be analyzed and compared to see which one delivers the best results. The analysis will be done with perceptual evaluation based on both, computational models and actual human subjects listening tests. For the computational evaluation, time and frequency domain analysis will be done, along with different calculations of error such as maximum absolute harmonic error and root mean squared error (RMSE). The main perceptual model used is the implementation of Bosi. The alternate models are the Perceptual Evaluation of Audio Quality (PEAQ) and the Noise to Mask Ratio (NMR). A different and easy to implement method is introduced where the error of aliasing and harmonic decay is analyzed separately using the A-Weighted curve for the human hearing dependency on frequency. Finally, to corroborate the results of the computed perceptual models, an ABC/Hidden Reference listening test is conducted on human subjects.
2

Theoretical Background

2.1 Virtual Analog Synthesis

VA synthesis is the part of the audio engineering field which attempts to emulate the sounds found in early analog synthesizers, especially those from the 1960s and 1970s [Pekonen and Holters, 2012]. This concept was introduced by Clavia, a Swedish company in 1995, when they introduced to the market the NordLead synthesizer [Pekonen and Välimäki, 2011]. This synth was the first of many to have a digital implementation of analog subtractive sound synthesis.

2.1.1 Subtractive Synthesis

VA synthesis is focused mainly on subtractive sound synthesis. It is defined as the filtering, using a time-varying lowpass filter (LPF), of a source signal rich in spectral properties, which is usually a periodic classical waveform (i.e. sawtooth, square, triangular or a mix), in order to obtain a desired specific waveform [Pekonen and Holters, 2012].

Subtractive synthesis was the basis technology for most synths of the 1960s and 1970s until it was overtaken by other techniques [Pekonen and Välimäki, 2011]. The desire of the people for the warmness of the sound produced by this type of synths has become the main reason of the spike in VA development over the last three decades, adding to it the possibility to have many synth models in a single machine or plug-in.
2.1.2 Issues

Designing and implementing VA synthesizers is not an easy task. The main reason for this is that emulating analog electronics with digital processing (programming language code) is much harder than it appears to be. One of the main problems is the aliasing caused by sampling an analog waveform which has rapid changes, because its spectra has infinitely high frequencies, therefore is not bandlimited. The other main problem is that analog electronics are not linear, but linearity is kept on digital electronics. This issue has to be solved by implementing this nonlinearity in the digital world [Välimäki and Huovilainen, 2006].

2.2 Source Signal Generation

In audio synthesis, the low-frequency oscillators (LFOs) usually act as controllers, they are allowed to use mathematically perfect waveforms which contain a lot of aliasing, because they are not being used to render musical notes. On the other side, a big part of VA synthesis is generating the waveforms to be later manipulated, just as it is on physical analog synthesizers. This is the function of the pitched oscillators in the physical world, which translates to the oscillator algorithms in the virtual world.

The oscillators which do render pitched notes cannot be mathematically perfect, in fact, they have to mitigate the aliasing. Trivial oscillator algorithms which generate periodic classical waveforms have several issues in the VA world. The most relevant problem is that the discontinuities in the waveform or its
derivative create an aliasing distortion which produces, in theory, an infinite number of harmonic components [Pekonen and Holters, 2012].

2.2.1 Aliasing

The aliasing problem cannot be observed in the time domain, only on the frequency domain. The discontinuities in the trivial classical waveforms create many undesired spectral peaks at frequencies that differ from the fundamental and its original harmonics (integer multiples of the fundamental). This error will corrupt the sound quality. The aliasing of a reverse sawtooth waveform is observed in Figure 1; in the ideal case, only the peaks at the fundamental and the harmonics would appear, all the other peaks are the aliasing. Three main audible disturbances are cause by aliasing and are described below [Välimäki and Huovilainen, 2007].

![Aliasing in the Reverse Sawtooth Waveform](image)

**Figure 1: Aliasing in the Reverse Sawtooth Waveform.** Time domain (left) and FFT spectrum (right) of reverse sawtooth waveform showing aliasing [Pirkle, 2014].

**Aliasing Effects**

Inharmonicity is the first effect of aliasing. Since the aliased components fall between the desired harmonics (spaces where the fundamental does not divide the sampling), the harmony is lost. The most audible components are the ones below the fundamental frequency.
The beating effect on the other hand, occurs when one of the aliased components appears in the vicinity of a desired harmonic component (less than 10Hz apart), and their amplitude is similar. The sum of those components generates the beating effect. If the aliased component is right on the harmonic, the summing of the two would cause amplitude distortion.

The last effect to introduce is heterodyning. Sometimes, when vibrato or glissando is used, the fundamental frequency is varied. So when the frequency of some aliased components are changing in the opposite direction, heterodyning can be heard.

2.2.2 Masking and Hearing Threshold

The masking effect is a psychoacoustic property where the human hearing system cannot differentiate two different sounds, so it becomes just one. There are mainly two types of masking: time and frequency dependent. The effects of temporal masking are not of big importance to this research, given that the purpose is to synthesize pure tones, therefore temporal effects do not have an noticeable impact.

The frequency masking is the effect that affects or aids the problem of aliasing directly. Each element of the frequency domain covers an asymmetric range, higher frequencies more than lower, where only that particular frequency is heard, even though other components may be in this area. Luckily, these other components can be the aliasing components, so that the harmonics can mask the aliasing near them. The anatomical explanation for this effect is the manner the
inner hair cells work inside the cochlea, where the physical distance between each characteristic place for a certain frequency is too small to differentiate from another characteristic place corresponding to a close frequency.

The human ear sensitivity is also frequency-dependent, which is referred as the human hearing threshold. This means that different levels of pressure are required for each frequency, for it to commence being perceived by the ear. The low and high frequencies are the ones with the highest minimum required levels, while the mid frequencies are the one more easily perceived.

**Perceptually Alias-Free Effects**

In order to obtain a cleaner and more desirable sound from the VA synthesizers, it is necessary to implement oscillator algorithms different from the trivial ones. Several techniques have been developed to completely eliminate the aliasing, but they are very expensive in computation terms. Other techniques take advantage of the perceptual hearing effects of the human beings described in the previous section, and use them to hide the aliasing.

The aliasing components have to be kept to mainly high frequencies (above the fundamental frequency), where it becomes almost inaudible, specially near the Nyquist frequency. This is due to the effect of the human Hearing Threshold only. The rest of the aliasing that is not close to Nyquist has to be distributed over the audible range in a manner that the auditory masking phenomenon hides it from the human hearing without actually eliminating it.

When the hearing threshold and the masking effect successfully hide the
aliasing to the human ear, the sound is referred as Perceptually Alias-Free and has no difference with an alias free sound in the ears of a person.

2.3 Types of Aliasing Reduction Methods

Since the 1990s, several discrete-time methods for oscillators that generate signals with none or less aliasing than the trivial have been developed. Mainly, three classes of antialiasing oscillator algorithms currently exist: Ideal Bandlimited, Quasi-Bandlimited and Alias-Suppressing. A separate class referred as ad-hoc methods is considered as well.

2.3.1 Ideally Bandlimited Algorithms

An ideally bandlimited method generates a predefined number of harmonics components, generating a waveform which of course is completely alias-free. These methods basically simulate the sampling of ideally low-pass filtered classical waveforms [Pekonen and Holters, 2012].

Discrete Summation Formulae (DSF)

In the 70s, Winham and Steiglitz introduced the first approach to use additive synthesis in sound signal generation [Winham, 1970]. Based on this, a couple of years later the Discrete Summation Formulae (DSF) concept was introduced [Moorer, 1976].

This approach, using only properties of trigonometric functions, reduces a sum of harmonically related sinusoids into a fraction of two trigonometric functions. It allows the generation of $N$ harmonics of a fundamental
simultaneously. It is based on the closed form expression of a geometric series
(See Equation 1).
\[ 1 + r + r^2 + \ldots + r^{N-1} = \frac{1 - r^N}{1 - r} \]  

(1)

The implementation is alias-free. It produces only the harmonics below the Nyquist limit. This can be achieved by the simple Equations 2 and 3.

\[ y_{bl}(n) = -\frac{2}{\pi} \sum_{k=1}^{K} \frac{\sin\left(\frac{2\pi k f_0 n}{f_s}\right)}{k} \]  

(2)

\[ K = \left\lfloor \frac{f_s}{2f_0} \right\rfloor \]  

(3)

where n is the sample index.

This method was active in research until a decade ago, with its most recent development being in [Lazzaro and Wawrzynek, 2004].

**Additive Synthesis**

A bandlimited simulation of analog synthesizers using additive synthesis was developed [Chaudhary, 1998]. This is a wider concept, which includes the previous described DSF method. This is what is known as the general concept of Additive Synthesis.

There are three main categories. The first one is the Oscillator Bank Synthesis, it is basically a bank of sinusoidal oscillators, which uses one oscillator for each partial.

The second category is the Wavetable Synthesis, it was the defacto standard in the audio synthesis business for many years. It is basically a predetermined number of harmonically related sinusoids stored on a table where
one cycle of an oscillation is played. Changing the rate of the look-up table changes the fundamental frequency of the waveform. Wavetable synthesis has some limitations when the fundamental frequency is desired to vary over a wide range [Massie, 1998].

And the last category is Inverse FFT Synthesis, where a bandlimited oscillator is implemented by frequency-domain synthesis [Deslauriers and Leider, 2009]. It specifies the spectral content of the classical waveforms in the frequency domain and then applies the inverse fast Fourier transform (IFFT) to the synthesized spectrum in order to get the time waveform.

**Feedback Delay Loop (FDL)**

The newest contribution to the Ideally Bandlimited algorithm field was made in 2009. It is by using a Feedback Delay Loop (FDL) to implement alias-free VA oscillators. This approach basically generates an impulse train using a feedback delay loop. Using a leaky integrator, the classical waveforms are then obtained. The output of this approach is perceptually harmonic, even though it is not exactly periodic. When time varying pitch is used, some attenuation in amplitude appear only at high frequencies, but it can be adjusted. This method uses less computational costs than other Ideally Bandlimited algorithms [Nam et al., 2009].

### 2.3.2 Quasi-Bandlimited Algorithms

Stilson and Smith were the first ones to introduce the concept of Quasi-Bandlimited waveform synthesis in 1996. Their idea behind it is that by
differentiating a continuous waveform with respect to time, a sequence of impulse functions comes out.

A Quasi-Bandlimited method allows certain aliasing at high frequencies, where human hearing is less sensitive to aliasing distortion than at low or mid frequencies. These methods basically sample lowpass-filtered continuous-time waveforms, but unlike the Ideally Bandlimited methods, this lowpass filter is not ideal, it is a realizable filter that has a non-infinitesimal transition band [Pekonen and Holters, 2012].

**Bandlimited Impulse Train (BLIT)**

The first type of Quasi-Bandlimited algorithms is the BLIT method. Since Stilson and Smith in 1996 published their paper on the subject, it opened a new field of research in oscillator algorithms which as of 2014 is still very active. This method uses only the original idea of Quasi-Bandlimited waveform synthesis. In order to obtain a sawtooth wave you have to [Pekonen and Holters, 2012]:

1. Differentiate a continuous-time classical waveform with respect to time. A sequence or train of impulse functions (BLIT) with a DC offset is obtained.

2. Filter the impulse train with a lowpass filter. Every impulse is replaced with the impulse response (IR) of the lowpass filter.

3. Compute the impulse response values so the BLIT can be synthesized in the digital domain.

4. The resulting BLIT is numerically integrated to obtain the bandlimited waveform.
The impulse train obtain previous to the waveform can be observed in Figure 2. When the lowpass filter is ideal, the IR is the $sinc$ function (See Equation 4) and the method would produce the same results as the Ideally Bandlimited oscillator algorithms. But since the $sinc$ function is infinitely long, the BLIT method cannot be implemented, therefore several realizable approximations have been developed.

![Figure 2: Unipolar Impulse Train for BLIT. Time domain of a unipolar bandlimited impulses train equally spaced [Pirkle, 2014].](image)

**Bandlimited Impulse Train - Sum of Windowed Sincs (BLIT-SWS)**

In the original research of BLIT, the actual method possible to implement was called BLIT-SWS (Sum of Windowed Sincs). This is the earliest realizable approximation of the BLIT method, introduced in the same paper where the original BLIT method was developed [Stilson and Smith, 1996]. This process is summarized in the block diagram in Figure 3. They proposed that since the $sinc$ function cannot be implemented, then a windowed version of it would work as a basis function for the BLIT method. The method would be as follows:

1. Differentiate a continuous-time classical waveform with respect to time. A sequence or train of impulse functions (BLIT) with a DC offset is obtained.

2. The windowed sinc function is sampled and tabulated. See the basis
function in Equation 4.

\[
h_{id}(t) = 2f_c \text{sinc}(2f_c t) = \frac{\sin(2\pi f_c t)}{\pi t}
\]  

3. Every impulse in the BLIT is replaced with the IR value stored in the table.

4. The resulting BLIT is numerically integrated to obtain the bandlimited waveform. This can be done for the sawtooth in Equations 5 and 6.

\[
s_{\text{saw}}(t) = -\frac{1}{2} + \int_{0}^{t} \delta(\tau) - C_1 d\tau 
\]  

\[
C_1 = \int_{0}^{T} \frac{\delta(\tau)}{T} d\tau
\]

Figure 3: **Block Diagram for BLIT.** Block Diagram for BLIT algorithm used to produce a sawtooth wave [Pirkle, 2014].

**Bandlimited Impulse Train with Modified Frequency Modulation (BLIT-FM)**

A different approach to the BLIT method using a technique based in Frequency Modulation (FM) synthesis was developed. The paper explains the equations required for bandlimited pulse generation using this modified FM synthesis method [Timoney and Lazzarini, 2008].

The subject on BLIT-FM was expanded and referred as Modified Frequency Modulation (BLIT-modFM). It provides a more smoothly evolving
spectrum with respect to variations in the modulation index [Lazzarini and Timoney, 2010].

**Bandlimited Impulse Train Low-Order Fractional Delay Filters (BLIT-FDF)**

The first approach to the idea of implementing Fractional Delay Filters (FDF) in the BLIT method was in a Masters thesis research [Pekonen, 2007]. It is based on the approach by Nam et al in 2009 of the FDL approach. The general FDF concept was later introduced [Nam et al., 2010]. They found that the 3rd order B-Spline FDF was the most efficient for the BLIT method. The B-spline interpolation is described in Equation 9. During the same year, a variable version was developed [Pekonen et al., 2010].

**Bandlimited Impulse Train Non-Linear Phase Function (BLIT-NLP)**

This is the most recent contribution to the Quasi-Bandlimited methods for VA synthesis, next to its corresponding BLEP implementation [Pekonen and Holters, 2012]. The approximations of the Ideally Bandlimited basis functions are usually linear-phase, this paper proposes a non-linear approach. The algorithm transforms the impulse or step response of an analog filter to a digital filter using a modified impulse invariance transformation method which applies arbitrary (sub-sample) offsets to the response of the filter. The result is a set of parallel first and second order IIR filters. Analysis is pending in whether the nonlinearity of the phase response has audible effects.
Bandlimited Impulse Train with Sampled Infinite Impulse Responses of Analog Filters (BLIT-IIR)

The latest advancement in the BLIT field was by including sampled Infinite Impulse Responses (IIR). Most BLIT implementations are based on Finite Impulse Responses (FIR), but the paper shows how to implement it using IIR filters which have a better stop-band rejection performance. The BLIT is obtained by emulating the sampling at an arbitrary sampling rate of an impulse train filtered by an analog IIR filter [Tassart, 2013].

Bandlimited Step Function (BLEP)

The BLEP method was first introduced as an improved version of the BLIT method [Brandt, 2001]. The BLIT approach has some issues from the numerical integration. When the IR values are not exact, the output comes out with an undesired DC component, thus the BLEP method is an improved BLIT method that does not need to perform integration at run-time, but rather it integrates the BLIT basis function in advance.

The ideal BLEP basis function would then be the integral of the sinc function (See Equation 7) which is infinitely long, just like the sinc function.

\[
h_{\text{tid}}(t) = \int_{-\infty}^{t} 2f_c \text{sinc}(2f_c\tau)d\tau = \frac{1}{2} + \frac{1}{\pi} \text{Si}(2\pi f_c t) \quad (7)
\]

The basis function shown in 7 is infinitely long, therefore it need to be approximated to be implemented. Because of this, the BLEP method cannot be implemented, therefore several realizable approximations have been developed.
Minimum Phase Bandlimited Step Function (mBLEP)

This is the earliest realizable approximation of the BLEP method, introduced in the same paper by Brandt where the original BLEP method was introduced [Brandt, 2001].

The idea is to accumulate a minimum-phase representation of the windowed sinc function to produce an approximation of the bandlimited step function. This approximation uses a non-linear phase basis function.

Bandlimited Step Function using a Look-up Table (BLEP-LUT)

This implementation of the LUT-BLEP method was done at Korg in 2006. They were awarded the U.S. Patent 7,589,272 for their implementation of the BLEP method [Leary and Bright, 2009]. The steps to follow are [Pirkle, 2014]:

1. Generate Impulse Response (IR) of a lowpass filter (LPF), which is a sinc function. See Figure 4 for a visualization of the function in the time domain.

2. The windowed sinc function is integrated and converted to bipolar. See Equation 7. See Figure 4 for a visualization of the integrated bipolar function in the time domain.
3. The resulting BLEP is subtracted by the regular step function to find the BLEP residual.

4. The BLEP residual is stored in a table, normal for rising edges and inverted for falling edges. See Figure 5 for both residuals.

![Figure 5: Bandlimited Step Function (BLEP) Residuals. Normal BLEP residual for the rising edge (left) and inverted BLEP residual for the falling edge (right) [Pirkle, 2014].](image)

5. The trivial sawtooth waveform is generated using a bipolar modulo counter.

6. The points stored in the BLEP residual table are modified one by one in the trivial sawtooth around the discontinuity.

This method is the focus of the research of this thesis. A more detailed description and implementation of this method is given in the Proposed Method chapter.

**Polynomial Bandlimited Step Function (PolyBLEP)**

The issue of using trigonometric function to implement the BLEP-LUT method gave a new interpretation using a polynomial as the basis function instead of the \( \text{sinc} \) function [Välimäki and Huovilainen, 2007]. The idea is to not have to use a table, which eliminates the look-up delay. It instead, approximates
the BLEP residual function to a simple polynomial (See Equation 8). This polynomial is obtained from integrating a unipolar triangle (See Figure 6.a) and then converting it to bipolar (See Figure 6.b). The results of this method are very good and close to the BLEP method and its implementation is much simpler. The residual obtained is shown in Figure 6.c and a comparison to the original BLEP method is observed in Figure 6.d.

$$p_{PolyBLEP}(t) = \begin{cases} 
\frac{t^2}{2} + t + \frac{1}{2} & \text{when} \ -1 \leq t \leq 0 \\
-t^2 - \frac{1}{2} & \text{when} \ 0 < t \leq 1
\end{cases} \tag{8}$$

Figure 6: **PolyBLEP Residuals.** a) Unipolar triangle. b) Integrated unipolar triangle converted to bipolar. c) PolyBLEP residual. d) Comparison of PolyBLEP and BLEP residuals [Pirkle, 2014].
Bandlimited Step Function using Low-Order Fractional Delay Filters (BLEP-FDF)

A corresponding BLEP version of the previously BLIT modified method with fractional delay filters was also developed [Välimäki et al., 2012]. They found that the 4th order B-Spline FDF was the most efficient for the BLEP method. It is basically a polynomial approximation of the BLEP method based on integrated B-spline interpolation which does not require the use of a table or oversampling. The method uses low-order polynomials, which correspond to shorter correction functions than when a table is used.

The B-spline interpolation is described in Equation 9.

\[
h_{IB}(n) = \int h_B(n)dD + E_B(n) \tag{9}
\]

B-spline interpolation was chosen over Lagrange because its spectra decays faster, implying that it can suppress aliasing more effectively. The 4th order version was chosen because the odd-order polynomials, in this case 3rd, are as computationally complex as the higher even-order.

This idea was later tested [Pekonen et al., 2011]. In this paper, a Moog synth sawtooth waveform is simulated using a fundamental frequency dependent oscillator based on the 4th order B-Spline BLEP approach and it is compared statistically to other main methods like additive synthesis, BLIT-FDF and DPW. It was found that the 4th order B-Spline BLEP approach performed better than all the other approximations and the spectral error was reduced to almost being negligible. This will be the base for the listening test described later in the
chapter.

**Bandlimited Step Function with Non-Linear Phase Function (BLEP-NLP)**

This method is the most recent contribution to the Quasi-Bandlimited methods for VA synthesis, next to its BLIT implementation [Pekonen and Holters, 2012]. The previous methods propose linear-phase approximations of the impulse or unit step response of the ideal lowpass filter. Unlike the previous methods, this one proposes a non-linear phase approximation of the ideal filter responses. For a wider explanation, please refer to the BLIT-NLP section above.

### 2.3.3 **Alias-Suppressing Algorithms**

An Alias-Suppressing method possesses aliasing distortion in the complete audible band, but using some strategies, this distortion is suppressed in comparison to the trivial approach [Pekonen and Holters, 2012]. These methods are the least expensive computationally. They sample a signal that has the same spectral content as the desired waveform but with a steeper roll of rate (tilt) which lowers the aliasing in the whole audible band.

**Filtering a Full-Wave Rectified Sine Wave (Lane)**

Back in the 90s, the first alias-suppressing approach was developed [Lane et al., 1997]. It generates a waveform where the harmonic partials fall off rapidly with harmonic number $m$. It then has to be passed through a low-pass filter (LPF) and its output is processed by a first-order high-pass tracking filter
\[ H(z) = \frac{\alpha(1 - z^{-1})}{(1 - \gamma z^{-1})} \] (10)

Differentiated Parabolic Wave (DPW)

The most recognized method of this category is the Differentiated Parabolic Waveform (DPW), which produces the sawtooth by differentiating a piecewise parabolic waveform. The concept of DPW, basically, first evaluates an integrated polynomial version of the signal and then differentiates and scales it [Välimäki, 2005].

\[ y(n) = \left( P_0 \right)^{N-1} \nabla_{S_N}^{N-1}(n) / 2^{N-1}N! \] (11)

The DPW interpretation, analysis, and research was later expanded [Välimäki and Huovilainen, 2006]. In this method, the basic idea is to reduce the aliasing of a sawtooth waveform by modifying the spectral tilt of the signal to be sampled [Pekonen et al., 2011]. The block diagram in Figure 7 summarizes the method. These are the steps to implement the DPW algorithm:

1. The trivial sawtooth waveform is generated using a bipolar modulo counter.

2. The waveform is raised to the second power.

3. The signal is differentiated with a first difference filter with transfer function \( 1 - z^{-1} \).

4. The obtained waveform is scaled by factor \( c = \frac{f_s}{4f(1-f/f_s)} \).
Differentiated Parabolic Waveform Variation (DPW-2X)

Valimaki also included a slightly improved version of the original DPW algorithm called DPW-2X. While the regular DPW only differentiates a piecewise parabolic waveform, the DPW-2X oversamples and decimates the parabolic wave with a simple filter prior to differentiation. This version has better results, due mainly to the oversampling.

High Order DPW (DPW-HO)

An improvement to the original DPW method was introduced where higher orders were implemented [Välimäki et al., 2010]. It proposes two new classes of polynomial waveforms that can be differentiated one or more times to improve the output of the DPW approach. It provides a better alias-suppression because it extends the method to higher polynomial orders.

Polynomial Transition Regions (PTR)

In 2012, an extension of the DPW method was developed, although it is quite different from it. The disadvantages of the DPW method were mainly the scaling of the output signal and the requirement for differentiable polynomial waveforms. The PTR algorithm processes each discontinuity specifically without...
modifying all other signal samples which is unnecessary [Kleimola and Välimäki, 2012]. PTR replaces the samples on a finite region around each discontinuity with values taken from a smooth polynomial, the results obtained have 40% less operations in comparison to DPW.

\[
y(n) = \begin{cases} 
p_W(n) - c_{dc} & \text{when } \phi(n) < W T_0 \\
x(n) - c_{dc} & \text{when } \phi(n) \geq W T_0 
\end{cases}
\]  

(12)

The PTR is much less expensive computationally than DPW because outside the transition region, which is wider than \( W = N - 1 \) samples, the method reduces to a bipolar modulo counter with an offset. PTR is considered a time-reinterpretation of the DPW algorithm. The computational costs of PTR are lower than those of PolyBLEP and BLIT-FDF. A lot of research is left in this field for optimization of alternative polynomial functions for the PTR method.

2.3.4 Ad-hoc Methods

The ad-hoc methods produce similar periodic classical waveforms to the trivial ones by using simple non-linear processing techniques [Pekonen and Holters, 2012]. These methods are developed for very specific applications, for which they are not of interest in this research.

2.4 Comparison of Most Relevant Methods

All the methods described in this chapter have very similar implementation and basic ideas, but each has different sound quality performances as well as computational complexity and amount of memory used.
It is very important to have a graphic comparison of the most relevant oscillator algorithms, as well as a table comparison of their properties and a ranking for the highest fundamental frequency which is still perceptually alias-free classified by the Bosi method.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Sound Quality</th>
<th>Computations</th>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trivial</td>
<td>Poor</td>
<td>Very Low</td>
<td>None</td>
</tr>
<tr>
<td>Lane</td>
<td>Good</td>
<td>Average</td>
<td>Very Small</td>
</tr>
<tr>
<td>DPW</td>
<td>Good</td>
<td>Very Low</td>
<td>Very Small</td>
</tr>
<tr>
<td>DPW2X</td>
<td>Very Good</td>
<td>Low</td>
<td>Very Small</td>
</tr>
<tr>
<td>PTR</td>
<td>Very Good</td>
<td>Low</td>
<td>Very Small</td>
</tr>
<tr>
<td>PolyBLEP</td>
<td>Very Good</td>
<td>Low</td>
<td>Very Small</td>
</tr>
<tr>
<td>Wavetable</td>
<td>Very Good</td>
<td>Above Average</td>
<td>Very Large</td>
</tr>
<tr>
<td>BLIT-SWS</td>
<td>Very good</td>
<td>Above Average</td>
<td>Small</td>
</tr>
<tr>
<td>BLIT-FDF</td>
<td>Very good</td>
<td>Above Average</td>
<td>Small</td>
</tr>
<tr>
<td>BLEP-FDF</td>
<td>Excellent</td>
<td>Above Average</td>
<td>Small</td>
</tr>
<tr>
<td>BLEP</td>
<td>Excellent</td>
<td>Average</td>
<td>Small</td>
</tr>
<tr>
<td>Additive Synthesis</td>
<td>Excellent</td>
<td>Very High</td>
<td>Very Small</td>
</tr>
</tbody>
</table>

Table 1: **Comparison of Oscillator Algorithms in Terms of Quality.** Comparison of most relevant oscillator algorithms in terms of sound quality, computational complexity and memory consumption adapted from [Välämäki and Huovilainen, 2007].

Table 1 shows a comparison of some of the most relevant oscillator algorithms explained in the chapter, ranked by sound quality. The BLEP method which is the interest of this study is more computationally expensive than some of the other algorithms, but the research chooses this method mainly, because of its good alias elimination and improvement of the harmonic envelope in comparison to the others. The BLEP method is the one closest to the quality of Additive Synthesis.

In previous research, implementations of the oscillator algorithms have been evaluated and a ranking has been obtained for the highest fundamental
Algorithm | Fundamental frequency
---|---
BLEP | 5135 Hz
BLIT-FDF | 4593 Hz
BLIT-SWS | 1203 Hz
DPW-2X | 1136 Hz
DPW | 600 Hz
Lane | 600 Hz

Table 2: **Comparison of Oscillator Algorithms in Terms of Highest Alias-Free Fundamental Frequency**. Highest fundamental frequency perceptually free of aliasing for sawtooth oscillator algorithms [Nam et al., 2010].

frequency perceptually free of aliasing when generating a regular sawtooth waveform. This research is based on the Bosi analysis of frequency spectrum. Table 2 shows this classification and it can be observed that the BLEP method has the best performance amongst all the algorithms. This is another reason of why this method was the focus of this document.

The main methods implemented from the previous sections are compared side by side in the time and frequency domains (See Figure 8). The idea of this graph is to compare the effects of the methods in each domain. Additive synthesis has a perfect spectrum with only the harmonics and no aliasing present; but its waveform is very different from a trivial one, and the sinusoidal rippling can be observed around the discontinuity. The waveform is not that important, but the spectrum is. This is the reason additive synthesis is one of the best method to implement alias-free sounds. The BLEP, BLIT and DPW have similar waveforms, where the discontinuities are smoothed in a polynomial shaped manner, rather than the usual straight line found in the trivial version. The DPW method has the most noticeable effects over the time domain.
waveform. Looking at the spectrum of each of these methods, the BLEP method is the one that eliminates the more aliasing, and only few aliased components are seen near the highest frequencies. The BLIT follows after with slightly more aliasing, and the DPW is the one that reduces the least aliasing amount. All this analysis reflects the information found on Tables 1 and 2.

Figure 8: **Comparison of Several Oscillator Algorithms.** Time domain (left) and FFT spectrum (right) of sawtooth implemented by (top to bottom): additive synthesis, BLEP, BLIT and DPW at fc=1049 Hz.

### 2.5 Perceptual Evaluation of Antialiasing Oscillator Algorithms

Previous to the development of the actual thesis research, an evaluation of several methods for reducing aliasing was done to corroborate the information found the documentation available from other researchers. The comparison found in the previous section points to the BLEP method as the best option of the
Quasi-Bandlimited methods. A listening test was conducted to confirm this information and be able to move forward more safely with further research.

VA synthesis has focused over the past two decades to develop antialiasing oscillator algorithms based on subtractive synthesis. Several outputs have come out from this research, including Ideally Bandlimited, Quasi-Bandlimited and Alias-Suppressing methods as described in this chapter. These have been evaluated through several techniques such as spectrum analysis like the Bosi method, Noise-to-Mask Ratio (NMR), and Perceptual Evaluation of Audio Quality (PEAQ). The idea being developed here is to perform a listening test on experienced listeners to have a non-computed modeled perceptual evaluation.

Comparison of these methods has been done before [Pekonen and Välimäki, 2011], but not many listening tests have been performed or at least published or documented. The BLEP method has been found, in previous research, to have the closest performance to additive synthesis in terms of aliasing and envelope gain, specially at high frequencies, near to Nyquist. Followed by the BLIT and DPW. The idea of this test is to obtain a perceptual evaluation of these methods through a listening test, hoping to find the same results as the analytical evaluations previously done. The main four methods will be: Additive Synthesis, BLEP, BLIT and DPW. In order to have a real-world implementation of the test, the samples will not be a plain sawtooth, but an approximation to the “Moog” sawtooth, which is considered to have a desirable and unique sound. A complete analysis in this particular waveform has been
conducted for several methods [Pekonen et al., 2011].

2.5.1 Samples Used for Testing

As mentioned above, in order to make the sound more real-world and not plain, an approximation to the Moog sawtooth is used, where the ramp portion is not straight, but rather sigmoid shaped. This approximation have been implemented before using several algorithms and filters. A recorded version of the Minimoog Voyager as a reference for the implementation was plotted (see Figure 9).

![Moog Sawtooth waveform](image)

Figure 9: **Moog Sawtooth.** Waveform of the Moog sawtooth recorded from a Minimoog Voyager at fc=220.62 Hz [4].

2.5.2 Listening Test

The basis of this research is to perform a listening test to semi-experienced/experienced listeners to corroborate the previous research that
states that the BLEP method performs much better than any other antialiasing method, even though this is at a computational cost. Four main methods will be used: Additive Synthesis (referred from here on as Ideally Bandlimited), BLIT-SWS (referred as BLIT), BLEP-LUT (referred as BLEP) and DPW. The Ideally Bandlimited method is considered to be a reference, since it is of course alias free. Six audio samples were obtained for each method, covering the audible range. The idea is to do two different and independent experiments:

1. Ideally Bandlimited (alias-free reference) vs. BLEP, BLIT, DPW.
2. BLEP (reference best alias performance) vs. BLIT, DPW.

**Test Type**

Since the only waveform being tested is sawtooth, and the methods generate similar waveforms, it is necessary to implement a listening test designed for small impairments. Tests, such as the MUltiple Stimuli with Hidden Reference and Anchor (MUSHRA), are design for more noticeable changes, therefore they do not apply to this case. The International Telecommunications Union (ITU) gives Recommendation ITU-R BS.1116-2, Methods for Subjective Assessment of Small Impairments in Audio Systems [International Communications Union ITU, 2014]. The idea of this test is to compare two sounds: the reference (A) and a modified sample. The subject listens to three samples A, B and C. Where B and C are randomly chosen where one is the reference (hidden reference) and the other the modified sample. This test is referred as ABC with a Hidden Reference (ABC/HR).
ABC/HR is comparing only two samples, one modified and one unmodified, but it has a scale and two different questions, not to be confused with the ABX test which only has a discrete difference/no difference answer and only one question (X is equal to A or B). In ABC/HR two questions are asked: 1. How does A compare to B? 2. How does A compare to C?

In order to answer those questions a predefined scale is provided, going from Imperceptible to very annoying. The annoyance does not necessarily imply an undesirable sound, but rather a case where the impairment is very perceivable. This can be observed in Table 3.

<table>
<thead>
<tr>
<th>Impairment</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>Imperceptible</td>
<td>5.0</td>
</tr>
<tr>
<td>Perceptible, but not annoying</td>
<td>4.0</td>
</tr>
<tr>
<td>Slightly annoying</td>
<td>3.0</td>
</tr>
<tr>
<td>Annoying</td>
<td>2.0</td>
</tr>
<tr>
<td>Very annoying</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Table 3: *Grading Scale for the ABC/HR Listening Test*. Grading scale for ABC/HR listening test [International Communications Union ITU, 2014].

So, based on Table 3; in the case that B is the hidden reference, the ideal response for question 1 would be imperceptible (5.0) and the response for question 2 can be any value from 1.0 to 5.0 and there is no wrong or right answer.

**Experimental Conditions**

In order to obtain good and constant results from all the subject listeners on the test it is necessary to have headphones that perform well in the whole audible frequency range, that have a good noise cancellation technology and that
have a good IR. The headphones model HDJ-2000 [Pioneer Electronics, 2014] were chosen, because they meet all these requirements. The theoretical frequency range is 5 Hz to 30 KHz, with an impedance of 36 Ohms for a maximum input of 3.5 W, and sensitivity of 107 dB/mW. Impartial third-party charts of the headphones state that the frequency response is fairly plain from 70Hz - 2000Hz, extending this range from 30Hz - 10000Hz to be a good approximation. Its isolation behaves well from 30Hz to 10000Hz, after that decays slowly, and its impulse response stabilizes at 0.9 ms.

The conditions of the listening tests have to be constant, this is the reason that the test was performed on the same computer, with the same set-up/settings, on the same room, on the same position inside the room and with the same headphones. The room chosen was the Weeks Studio Control Room at the Frost School of Music, inside the Coral Gables campus of the University of Miami. When these conditions are met and the subjects are experienced listeners, experience shows that data from 20 subjects is enough for obtaining appropriate conclusions from the test. The computer used was a Windows 8.1 machine with a Realtek HD audio card.

The tests have two main sections, the training phase and the testing phase. The training phase is required for the subject to familiarize with the test, how to play the sound samples, how are the questions formulated and how are the questions answered. Both tests were performed by 21 subjects. All of them were undergraduate/graduate students of the Frost School of music with a
listening experience ranging from medium to high. The first test consists of 18
different scenarios. This number belongs to six scenarios per each of the three
methods: BLEP, BLIT, DPW to be compared to Ideally Bandlimited. Each of
those six scenarios corresponds to a different frequency (Hz) [98.1, 220.62, 495.44,
1049, 2096, 6256]. This frequencies were determined by the recorded samples of
the Minimoog Voyager synthesizer.

The second test consists of 12 different scenarios. This number belongs to
six scenarios per each of the two methods: BLIT and DPW to be compared to
BLEP. Same frequencies apply in this case. Each scenario has only 2 samples,
one for the reference and one for the method to be compared. The subject listens
to the samples in this order A-B-A-C. Subjects can reproduce the samples again
as many times a desired. The instructions have to be very clear for the test to go
well. A page with instructions was printed and handed to each subject to read
carefully before performing the test. This was done according to the ITU
recommendations. Finally, the implementation of the test was done in the
MATLAB environment. The reason for this is because it is cross-platform and
the data introduced/obtained can be directly analyzed inside the software. This
implementation was adapted from a more general implementation of listening
tests, which included other types of testing such as MUSHRA
[Vasquez Giner, 2013].

2.5.3 Results

The results for each of the listening test are presented separately.
Listening Test 1 - Ideally Bandlimited Focused

A general average of the absolute grades was calculated for each of the methods respect to the frequency (See Figure 10). This analysis is the first step and it is not meaningful enough to draw any conclusions.

The results of the test were analyzed using the Analysis of Variance (ANOVA) model. One subject was excluded from the test, because the data was significantly different compared to the rest of the subjects. The standard deviation (variance) for question 1 (reference) was statically small. For question 2, was statistically greater than for question 1.

A hypothesis test was done, and it was proved that the average of the
Scenario 6 \((f_c = 6256Hz)\), is different to the average of the other scenarios for the BLEP, with a 95% confidence level. The same test was done on the other 5 scenarios and their values were considered statistically equal or not different.

A new hypothesis test was done, and it was proved that the average of the Scenario 12 \((f_c = 6256Hz)\), is different to the average of the other scenarios for the BLIT, with a 95% confidence level. Same occurred with the Scenario 11 \((f_c = 2096Hz)\) and Scenario 10 \((f_c = 1049Hz)\). The same test was done on the other 3 scenarios and their averages could be considered statistically equal or not different. When the test was done for Scenarios 12 and 11, the average of 12 is statistically greater than Scenario 11. When comparing Scenario 10 with 11 and 12, no significant results were found.

A final hypothesis test was done, and it proved that the average of the Scenario 18 \((f_c = 6256Hz)\), is different to the average of the other scenarios for the DPW, with a 95% confidence level. Same for the Scenario 17 \((f_c = 2096Hz)\). The same test was done on the other 4 scenarios and their averages could be considered statistically equal. When the test was done for Scenarios 18 and 17, the average of 18 is statistically greater than Scenario 17. The DPW method performed better than the BLIT method in some of the cases. This contradicts pervious research, but the statistical evidence is not enough to draw any further conclusions. This might have been cause due to large variance in those sets of data.
Listening Test 2 - BLEP Focused

The same procedure for the Listening Test 1 was followed to analyze the results of Listening Test 2. An absolute grade average was calculated for each of the methods respect to the frequency (See Figure 11). As mentioned before, these absolute grade results are not a complete or sufficient analysis and ANOVA is done for further evaluation.

Figure 11: Listening Test for BLEP Focused Results. Average (Absolute grade) results from the listening test with the BLEP as reference. Blue: Question 1 (ref) - BLIT, red: Question 2 - BLIT , green: Question 1 (ref) - DPW, black: Question 2 - DPW.

The standard deviation (variance) for question 1 (reference) was statically small. For question 2, was statistically greater than for question 1. The same as in the first test. The first hypothesis test was done, and it was proved that the average of the Scenario 5 ($f_c = 6256Hz$), is different to the average of the other scenarios for the BLIT with a 95% confidence level. Same for Scenario 6
(\(f_c = 2096\, Hz\)). The same test was done on the other 4 scenarios and their averages could be considered statistically equal.

The second hypothesis test implemented proved that the average of the Scenario 12 (\(f_c = 6256\, Hz\)), is different to the average of the other scenarios for the DPW, with a 95% confidence level. Same for Scenario 11 (\(f_c = 2096\, Hz\)) and Scenario 10 (\(f_c = 1049\, Hz\)). The same test was done on the other 3 scenarios and their averages could be considered statistically equal. When the test was done for Scenarios 12 and 11, the average of 12 is statistically greater than Scenario 11. It was also found that Scenario 12 was significantly greater than Scenario 10. When comparing Scenario 10 with 11, not significant results were found.

The BLIT method performed better than the DPW method in some of the cases. This contradicts the first test, but corroborates previous research. In comparison to the results from the first test, the standard deviation (variance) was significantly smaller. Therefore, the results behaved in a less extreme manner. The maximum and minimum values are closer.

2.5.4 Conclusion

The results from the tests were mostly consistent with the previous research. The only contradiction was that the DPW performed better than the BLIT method in the first test. The statistical evidence was not sufficient to draw any conclusions. In the second test, the BLIT performed better than the DPW, which agrees with previous research. In all cases, it was evident that the aliasing increases as the fundamental frequency also increases. There was no statistical
evidence that showed a decaying line with respect to frequency. The ANOVA model draws several of the desired conclusions, but it also did not find many differences between the frequencies above the highest perceptually alias free frequency. The tests results showed the effects of frequency masking and hearing threshold over the aliasing. It was present, but the subjects did not hear it, statistically speaking. The main conclusion, which is the reason of this research, is that the BLEP method performs better than any other antialiasing method in terms of quality of sound. this includes the elimination of aliasing.

2.6 Choosing the BLEP Method as Focus of Research

Based on the research presented on this chapter and the listening test conducted, there is plenty of evidence that indicates that the BLEP method is the best in comparison to all the other studied algorithms in terms of quality of the sound and in terms of the highest perceptually alias-free fundamental frequency it can achieve. It might be more computationally expensive than others, but it is the one that gets the closest to an Ideally Bandlimited method response. These are the main reasons for choosing this only method as the focus of this research. A lot of potential is observed in this method, where optimization of certain parameters such as: the input function, the size of the table, the number of correction points, or the number of zero crossings, can take the BLEP method to even better results.

The idea is to improve this method only, so no further comparison with the other methods will be conducted. This is an assumption that if BLEP is
better than all the others, then an improved version of BLEP would also be
to all the other methods. This research will have the goal to improve the
original BLEP method, considered the worst case scenario, and try to obtain the
closest results to the best case scenario which is the Additive Synthesis method.

2.7 Restrictions in BLEP Documentation

One of the main issues with the subject of this thesis research is that the
field of VA synthesis is very narrow respect to the number of researches dedicated
to it. The main two research groups are the one directed by professor Valimaki at
the Aalto University in Finland and the one directed by Julius Smith at Stanford
University in California.

The number of papers of BLEP is not very large. There are currently a
total 24 publications on Quasi-Bandlimited methods, including 20 papers, one
patent and three thesis’s. Only five of these documents go into details of the
BLEP method. The fact that the BLEP method was developed in 2001, causes
an issue for this research because there are not current updated implementations
of it, due to the programming languages and computational advances in the past
14 years. This means that the all code has to be written from scratch which adds
a time factor to the research, but it also gives the opportunity to develop efficient
tools for the entire oscillator algorithms field that can be used by others in the
future.
3

Proposed Method

3.1 Description and Overview

The method developed for this research is the implementation of a genetic algorithm which based on a perceptual model, obtains the optimal input function for the BLEP method in order to have the highest Perceptually Alias-Free fundamental possible, with the smallest table size and the fewer look up times over each period. The evaluation of the results will be based on regular time domain and frequency domain analysis, and perceptual models like Bosi, PEAQ and NMR. The best results obtained will be corroborated by a listening test conducted on human subjects.

3.1.1 Block diagram

![Block Diagram of Proposed Method](image)

Figure 12: **Block Diagram of Proposed Method.** Overall description of the proposed method.

The block diagram from Figure 12 shows the main three blocks of the proposed method. Each block is very complex and will be described on its own in
the following sections. Based on some restrictions previously established, the GA generates a set of variables used to create the input function, which is later evaluated by the fitness function. This value or grade is the feedback the GA obtains to then modify the variables and provide a better or optimized result. This system runs in a loop until it finds the optimal input function that gives perceptually alias-free notes (waveforms) over all the desired frequency range.

The generalized purpose is to obtain at the end of this research a complete system that optimizes the performance of the BLEP method for some given conditions, which are defined by the restrictions. These options are not all required, some of them can be implemented as a free variable that can take any value, also referred as unrestricted. This includes the size of the look up table (N), the number of points to correct in each period (m) and the desired highest fundamental that is perceptually alias-free.

3.2 Input for the BLEP Method

The original input of the BLEP method is the impulse response of a low-pass filter (LPF). This idea can be graphically seen in Figure 13. A discrete impulse is fed to a LPF and the response is a symmetric periodic waveform that starts with one period of a sin function, but decays as it gets further from its center. This is also known as the sinc function. In the original method, the input is the sinc function as is. In the modified look-up table (LUT) method, this input is then windowed and processed to obtain the BLEP residual. This idea is explained in 14.
3.2.1 The Sinc Function

The definition of the sinc function is given in Equation 13.

\[ \text{sinc}(t) = \frac{\sin(f_c t)}{f_c t} \]  

(13)

In Figure 15, the sinc function can be observed with a total of \( n = 8 \) zero crossings, 4 on each side. One of the most important properties of this function is that the portion between two continuous zero crossings represents one cycle period of the waveform, except for the first zero crossing on each side of its center, where the period is reached at the crossing with the Y axis at 1 (the maximum absolute amplitude, same as the sin function).

The sinc function is calculated depending on the number of points \( N \) of the desired BLEP table. The reason for this is that no resampling or downsampling is desired over the this function and the BLEP residual all
Figure 15: The sinc Function. With \( n = 8 \) zero crossings, 4 on each side and \( N = 4,096 \) samples.

Together. In order to achieve this, it is necessary to manipulate the period of the sinc function as shown in Equation 14. This is the reason why the number of samples in Figure 15 is \( N = 4,096 \), no matter what the number of zero crossings \( n \) is.

\[
T = \frac{N - 1}{f_s n}
\]

In Equation 14, \( T \) is the period of the sinc function, \( N \) is the size of the table, \( n \) is the number of zero crossings desired, and \( f_s \) is the sampling rate used to sample the sinc function. Note that this \( f_s \) has no relation to the sampling rate of the final waveform.
3.2.2 Windowing

Windowing the sinc function before giving it as an input to the BLEP method has a positive effect in the frequency domain of the BLEP final waveform. The window has to be symmetric, because the sinc function is also symmetric and its properties must be untouched. Most common windows can be applied to the sinc function, but some have proven to have better performance than others. For instance, the Blackman window observed in Figure 16, gives one of the best results. An analysis of why the Blackman results are considered better than any other window is explained over the next few sections.

![Blackman Window with N = 4096 samples](image)

Figure 16: The Blackman Window. Using $N = 4,096$ samples.

The output of the Blackman windowing (shown in Figure 16) applied to the sinc function (shown in Figure 15) is observed in Figure 17. This windowing
is done in the discrete time domain by simple multiplication of the two waveforms. Note that in order to multiply vector in the time domain, these must be the same length. This is why the sinc function has a fixed number of samples \( N \), which corresponds to the same number of samples of the window. This way, a correct multiplication is achieved and no modification to the samples has to be performed, which might cause losses or repetitions of information.

![Windowed Input Function with N = 4096 samples](image)

Figure 17: **Windowed sinc Function.** Using the Blackman window with \( n = 8 \) total zero crossings, size of \( N = 4,096 \) samples.

The effects of windowing seen in the time domain of Figure 17 are not very meaningful. The main center component is kept almost intact, and the smaller side components of the sinc are diminished. The real effects are seen in the frequency domain, which will be explained later in the chapter.
3.3 Creation of the BLEP Table

The process to create the BLEP Table has several steps. The size of the table (N), is kept unmodified throughout the entire process. First the input has to be integrated. An integration in the continuous time is equivalent to a sum over time in the discrete domain. This is described by Equation 15, where \( x[k] \) represent the input function of the method, originally the windowed sinc function.

\[
x_{\text{integrated}}[k] = \sum_k x[k]
\]  

(15)

Next, the integral of the input function has to be converted to bipolar, given that the sinc function is a function that oscillates around 0 and has a maximum value of 1. The desired out has to cover the range -1 to 1 in order to correctly apply the method to a bipolar waveform. Because the sinc lower points are below 0, the range covered actually passes the -1 to 1 limits at some points, and this is not a negative effect (Refer to the BLEP vs. Additive Synthesis section). The integral of the windowed sinc function from the previous section is observed in Figure 18. In this figure, the size of the table is maintained as \( N = 4,096 \), as well as the number of total zero crossings \( n = 8 \), and the Blackman window is still used.

\[
x_{\text{bipolar}}[k] = 2x_{\text{integrated}}[k] - 1
\]  

(16)

After the signal is converted to bipolar, a step function is subtracted from it. Notice the step function also has to be bipolar in order to work. The bipolar step function equation is defined as in Equation 17.
The result obtained corresponds to the BLEP residual. The regular BLEP residual is used to correct certain number of points (samples) in a regular sawtooth on each cycle. For this case, an inverted BLEP residual is used because and inverse sawtooth is the one studied in this research. The regular BLEP residual and the inverted BLEP residual can be observe in Figures 19 and 20 respectively.

The complete process can be observed in Figure 21. This is part of the method that was mentioned as INPUT CREATION in the main block diagram.

\[
x_{\text{step}}[k] = \begin{cases} 
-1 & \text{when } k < 0 \\
1 & \text{when } k \geq 0 
\end{cases}
\]  

(17)
Figure 19: **BLEP Residual Table.** Using the Blackman window with $n = 8$ total zero crossings, size of $N = 4,096$ samples.

Figure 20: **Inverted BLEP Residual Table.** Using the Blackman window with $n = 8$ total zero crossings, size of $N = 4,096$ samples.
The rest of the block is explained in the Input Function Creation on the GA Section. Note that from here on, the Inverted BLEP Residual will be the one used because the studied sawtooth is the actual Reverse Sawtooth, so for simplicity, it will be simply referred as BLEP Residual.

The size of the table (N) should always be an even number. This means that the value from the middle of the table, which is the same as the discontinuity sample value is missing from it. The reason for this is that all the values in the first half of the table correspond to the left side of the discontinuity, and the values for the second half would correspond to the right side of the discontinuity. This can be checked by looking at the sign of the sample values right before and right after the discontinuity, being for the rising edge positive and negative, respectively. And negative and positive, respectively, for the falling edge.

The exact fundamental frequency to correctly implement and analyze this method depends on the size of the FFT, as described in the DFT for Audio Synthesis section.
3.4 Implementation of the Trivial Waveform

In order to apply the BLEP method to the trivial waveform, this has to be implemented as well. Since the application is real-time focused, the trivial waveform will be implemented in real time as well as the application of the BLEP correction. Note that the waveform studied during this entire research is actually referred as the Reverse Sawtooth, which is the one that increases from 0 to 1 over the time domain. The Regular Sawtooth is the one that actually starts at 1 and decays all the way to 0. In a purpose of simplicity, for the rest of this document the Reverse Sawtooth will be referred just as Sawtooth.

For this to work, a modulo has to be established to keep track of each sample in a period of the waveform. The increment of this modulo is defined in Equation 18.

\[ \text{inc}_{\text{modulo}} = \frac{f_c}{f_s} \]  

(18)

The actual sawtooth waveform is just the same modulo value incremented at each sample, discounted by the unit when it reaches a value greater than 1. This sawtooth waveform is unipolar, so it has to be converted to bipolar using Equation 16. The length of the waveform in samples is defined as a number that is a power of two. This way, the DFT will be more efficient when running the algorithm (Refer to the DFT for Analysis section). The sawtooth waveform can be appreciated in Figure 22.
3.4.1 Issues at High Fundamental Frequencies

Note that as the fundamental frequency of the waveform gets higher, the modulo increment also get larger. This is a problem because bigger steps are taken and when a step adds to a value greater than the unit, its discount by 1 will have an impact of the maximum and the minimum value (peak values) of the waveform as seen on Figure 23. This problem has a negative impact on the BLEP method and it is the main reason of why it is harder to eliminate aliasing at higher frequencies. The simple solution for this is oversampling, but the sampling rate of sounds is usually defined by the entire synthesizer system, so the desired optimization will be while keeping a same sampling rate over the entire audible range, in this research it will be of 48kHz.
3.5 Applying BLEP to the Trivial Waveform

It is necessary to know the center of the table to find the exact BLEP point to replace the trivial sample. The center of the table corresponds to the last sample before the discontinuity. Remember that the discontinuity point is not included in this table. For example for a Table of size \(N = 4,096\), the center is 2,048, counting that the vector is formed from index 1 to index 4,096. The exact location of the sample to be changed is found using this center \((c)\) and the Equation 19 [Pirkle, 2014]. The variable \(m\) refers to the number of points to be modified at each side. This notation will be kept the same during the entire
\[ \text{index} = \begin{cases} c(1 + \frac{\text{mod} - 1}{\text{inc mod} m}) & \text{at the left side of the discontinuity} \\ c(\frac{\text{mod}}{\text{inc mod} m}) + c + 1 & \text{at the right side of the discontinuity} \end{cases} \] (19)

There is a limiting factor here and it is the number of points to be corrected at high frequencies. In the case of a \( f_s = 48kHz \), when the frequency reaches a quarter of the Nyquist frequency, the number of points for each period is roughly 8, which gives only 4 points on each side to be modified. After this frequency, less than 4 points should be corrected on each side, to obtain a higher perceptually alias free fundamental. It is important to note that after making the first correction, a second one is possible, making the final value even more accurate. This double correction improves the aliasing even more but reduces the efficiency due to its computational costs, specially in real time implementations. More than two corrections are also possible, referred as multi-BLEP.

The BLEP waveform often surpasses the -1 and 1 limits, so it has to be normalized accordingly in order to avoid clipping, specially when handling WAV files. The BLEP waveform with 8 modified points total, 4 on each side, using the previous trivial waveform (from Figure 22) and the inverted BLEP residual (from Figure 20) is shown in Figure 24.

This waveform is achieved by running an algorithm that runs in real-time as the trivial waveform is created. The tracking of the index location is done using the same modulo for the trivial. The idea is to check every sample to see if the area of discontinuity (limited by the number of correction points of each side)
is reached. Once the modulo is in this area, the corresponding correction is applied by looking up the index from Equation 19 in the BLEP Residual Table and giving the sample the value at that index. Note that the mentioned computational costs in the previous chapter correspond to the checking of every sample to see if they are inside the discontinuity region. This is something that cannot be eliminated if the method used uses the idea of changing the waveform in real-time as the trivial waveform is constructed.

It is also important to note on this research that the commonly known waveform obtained from the BLEP method, where the discontinuity is smoothed by shaping the left side downwards, and the right side upwards is not the actual result of the BLEP method. This only occurs when having one zero crossing on
each side of the \textit{sinc} function and changing a very large number of points. A real or possible implementation of BLEP will look like a piecewise linear approximation of this decay because of its discrete nature and the small number of points that are actually corrected.

This waveform also shows a tilting and even incremented values on the left side and reduced values of the right side of the discontinuity when more zero crossings are included. Opposite to the commonly know waveform. This is due to negative rippling values of the side components of the \textit{sinc} function, giving it sometimes peak values greater than 1, thus having to be normalized to avoid clipping. Figure 24 shows the normalized (no clipping) version with the upwards components on the left and downwards on the right side of the discontinuity. A graph showing the commonly know graphical understanding of BLEP can be found in the KORG patent [Leary and Bright, 2009]. See Figure 25.

\textit{3.5.1 Number of Zero Crossings vs. Number of Correction Points Paradigm}

Note that in this implementation of BLEP, the number of correction points is referred as \( m \), and it is not the same number of zero crossings \( n \). In the original method, one zero crossing in the sinc function, corresponds to exactly one point of correction while constructing the BLEP waveform. One of the hypotheses introduced here is that this might be not necessarily the ideal situation in all cases. During this research this paradigm is broken and all possible combinations of \( m \) and \( n \) are analyzed. This was based on the first trials of the method when sometimes better solutions where found having \( m \) and \( n \)
Figure 25: **KORG BLEP Illustration.** Illustration of the BLEP implementation by KORG [Pirkle, 2014]. Originally adapted from [Leary and Bright, 2009].

a) Sawtooth waveform. The sample in grey is to be corrected.

b) The BLEP residual. The sample in grey is the correction value.

c) Sawtooth waveform. The arrows show the correction value and direction.

d) BLEP sawtooth waveform. The samples in grey were corrected.
being different, than with the commonly used approach.

3.6 DFT for Oscillator Algorithms

The Discrete Time Fourier Transform (DFT) is implemented through the Fast Fourier Transform (FFT) algorithm of Matlab®, `fft`. This is actually a DFT because the signal is sampled, rather than continuous. In order for the FFT to be correctly analyzed, the size of the FFT (`NFFT`) plays an important part in the selection of the length of the waveform and the fundamental frequency. In order to obtain a more accurate result, which eliminates the lobing (See Figure 26) and selects only the harmonic and the aliased peaks, it is necessary to window the signal in the time domain before doing the FFT. This can be achieved using any type of symmetric window. In this case the Dolph-Chebyshev window was used because it gave a good visualization of the FFT spectrum and the peaks were narrower.

If this window is not used, a large amount of information goes missing, specially at the peaks of the harmonics and at the lower dB levels. And this means that the information processed for further analysis would not be accurate.

After applying the Dolph-Chebyshev window and doing the FFT, the obtained spectrum is unnormalized, the values accumulated during the sample by sample transform. So it has to be divided by the size `NFFT` to obtain a normalized version. After this, the correct manner to obtain the amplitude is to only analyze the first half of the FFT, which corresponds to the positive frequencies and multiply by 2 to compensate for the other half, due to its
symmetry. A conversion to dB is necessary to appreciate the small changes.

Figures 27 and 28 show the FFT spectrum for the trivial sawtooth and the
BLEP corrected sawtooth respectively. Note the improvement in comparison to
the FFT without windowing in Figure 26.

Note that the harmonics at high frequencies are diminished in the BLEP
waveform in comparison to the trivial waveform. The exact phenomenon that
occurs here is not only that they are diminished, but the first harmonics are
boosted. Due to normalization this cannot be observed. For an analysis of this
refer to the Effects of BLEP on Harmonic Decay section.
Figure 27: **Trivial Sawtooth FFT.** Fundamental frequency of $f_c = 2,988.2813 \text{Hz}$, $f_s = 48 \text{kHz}$, and $NFFT = 32,768$.

### 3.6.1 FFT Size and bins

The correct FFT size to choose $NFFT$ is based on the length of the time domain waveform being transformed. In order to avoid zero padding at the time of doing the FFT, the same length of the waveform and FFT size are chosen. The FFT is much more efficient when using a size that is a power of two, for this reason the length of the samples will be that same power of two. A $NFFT = 131,072$ would give the same number of samples in the waveform, which at a $f_s = 48 \text{kHz}$ correspond to around 2.73 seconds of sound.

The number of bins is half of the size of the FFT. The reason for this is that half of the FFT corresponds to negative frequencies, and the other half to positive frequencies. But these two halves are equal in terms of amplitude, so the
Figure 28: **BLEP Sawtooth FFT.** Fundamental frequency of $f_c = 2,988.2813\,Hz$, $f_s = 48\,kHz$, and $NFFT = 32,768$. With $m = 8$ points modified with the BLEP method.
second half can be ignored in further analysis. The size of each bin in Hz is defined by Equation 20.

\[ L_{bin} = \frac{f_s}{NFFT} \text{(Hz)} \]  

(20)

For the case in Figures 27 and 28, the bin size is 1.46 Hz. This value is obtained with an \( NFFT = 32,768 \) and a \( f_s = 48k\text{Hz} \).

### 3.6.2 Choosing the Right Fundamental Frequencies

In order to have a good analysis in the frequency domain, the fundamental frequency chosen has to be a factor of this size of each bin, so that when the FFT is calculated, the harmonics and the fundamental can be clearly analyzed. This means, the fundamental lands right inside a bin, as well as all the harmonics. If a fundamental that is not a multiple of the bin size is used, it can throw away some information in the process due to not landing in an exact bin.

A MIDI table has to be calculated for each sampling rate and each FFT size. To differentiate between the lowest notes on the MIDI table, the FFT size has to be at least 32,768, which is the smaller power of 2 possible. This is due because the lower MIDI notes have a difference of less than 2 Hz, which a smaller table like 16,384 will not differentiate due to a bin size of about 2.92 Hz. Table 4 for the values of the MIDI Table in the case of 48 Hz sampling rate and 32,768 FFT size.

Note that the frequency \( f_c = 2,998.2813 \) for Figures 27 and 28 corresponds to a multiple of the bin size of 1.46 Hz.
<table>
<thead>
<tr>
<th>MIDI</th>
<th>Freq. (Hz)</th>
<th>MIDI</th>
<th>Freq. (Hz)</th>
<th>MIDI</th>
<th>Freq. (Hz)</th>
<th>MIDI</th>
<th>Freq. (Hz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>27.83203125</td>
<td>43</td>
<td>98.14453125</td>
<td>65</td>
<td>348.632813</td>
<td>87</td>
<td>1245.117</td>
</tr>
<tr>
<td>22</td>
<td>29.296875</td>
<td>44</td>
<td>104.003906</td>
<td>66</td>
<td>370.65469</td>
<td>88</td>
<td>1318.359</td>
</tr>
<tr>
<td>23</td>
<td>30.76171875</td>
<td>45</td>
<td>109.8672815</td>
<td>67</td>
<td>392.578125</td>
<td>89</td>
<td>1397.461</td>
</tr>
<tr>
<td>24</td>
<td>32.2265625</td>
<td>46</td>
<td>117.1875</td>
<td>68</td>
<td>416.01625</td>
<td>90</td>
<td>1479.492</td>
</tr>
<tr>
<td>25</td>
<td>35.15625</td>
<td>47</td>
<td>123.046875</td>
<td>69</td>
<td>439.453125</td>
<td>91</td>
<td>1567.383</td>
</tr>
<tr>
<td>26</td>
<td>36.62109375</td>
<td>48</td>
<td>130.371034</td>
<td>70</td>
<td>465.820313</td>
<td>92</td>
<td>1661.133</td>
</tr>
<tr>
<td>27</td>
<td>39.55078125</td>
<td>49</td>
<td>139.160156</td>
<td>71</td>
<td>493.65235</td>
<td>93</td>
<td>1759.277</td>
</tr>
<tr>
<td>28</td>
<td>41.015625</td>
<td>50</td>
<td>146.484375</td>
<td>72</td>
<td>522.94919</td>
<td>94</td>
<td>1864.746</td>
</tr>
<tr>
<td>29</td>
<td>43.9453125</td>
<td>51</td>
<td>155.273438</td>
<td>73</td>
<td>553.710938</td>
<td>95</td>
<td>1976.074</td>
</tr>
<tr>
<td>30</td>
<td>46.875</td>
<td>52</td>
<td>165.527344</td>
<td>74</td>
<td>587.40234</td>
<td>96</td>
<td>2093.262</td>
</tr>
<tr>
<td>31</td>
<td>48.33984375</td>
<td>53</td>
<td>174.316406</td>
<td>75</td>
<td>622.55859</td>
<td>97</td>
<td>2217.773</td>
</tr>
<tr>
<td>32</td>
<td>51.26953125</td>
<td>54</td>
<td>184.570313</td>
<td>76</td>
<td>659.17968</td>
<td>98</td>
<td>2349.609</td>
</tr>
<tr>
<td>33</td>
<td>55.6640625</td>
<td>55</td>
<td>196.289063</td>
<td>77</td>
<td>698.730469</td>
<td>99</td>
<td>2488.77</td>
</tr>
<tr>
<td>34</td>
<td>58.59375</td>
<td>56</td>
<td>208.00813</td>
<td>78</td>
<td>739.74094</td>
<td>100</td>
<td>2636.719</td>
</tr>
<tr>
<td>35</td>
<td>61.5234375</td>
<td>57</td>
<td>219.726563</td>
<td>79</td>
<td>783.691406</td>
<td>101</td>
<td>2793.457</td>
</tr>
<tr>
<td>36</td>
<td>65.91796875</td>
<td>58</td>
<td>232.910156</td>
<td>80</td>
<td>830.56604</td>
<td>102</td>
<td>2960.449</td>
</tr>
<tr>
<td>37</td>
<td>68.84765625</td>
<td>59</td>
<td>247.558594</td>
<td>81</td>
<td>880.371094</td>
<td>103</td>
<td>3136.23</td>
</tr>
<tr>
<td>38</td>
<td>73.2421875</td>
<td>60</td>
<td>262.207031</td>
<td>82</td>
<td>931.640625</td>
<td>104</td>
<td>3322.266</td>
</tr>
<tr>
<td>39</td>
<td>77.63671875</td>
<td>61</td>
<td>276.85469</td>
<td>83</td>
<td>987.304688</td>
<td>105</td>
<td>3520.02</td>
</tr>
<tr>
<td>40</td>
<td>82.03125</td>
<td>62</td>
<td>292.96875</td>
<td>84</td>
<td>1045.8984</td>
<td>106</td>
<td>3729.492</td>
</tr>
<tr>
<td>41</td>
<td>87.890625</td>
<td>63</td>
<td>310.546875</td>
<td>85</td>
<td>1108.88672</td>
<td>107</td>
<td>3950.684</td>
</tr>
<tr>
<td>42</td>
<td>92.28515625</td>
<td>64</td>
<td>329.589844</td>
<td>86</td>
<td>1174.80469</td>
<td>108</td>
<td>4186.523</td>
</tr>
</tbody>
</table>

Table 4: **MIDI Table for Piano Keyboard of 88 Notes.** Frequency of MIDI Table for an 88 key piano with an \( f_s = 48k\text{Hz} \) and a \( \text{NFFT} = 32,768 \).
3.6.3 Finding Harmonics and Aliasing Peaks

In the worst scenarios, the amplitude of the aliasing peaks gets very close to the amplitude of the harmonics in the higher frequencies. The reason for this is that the higher frequencies harmonics have a very small amplitude and the aliasing from previous (lower) harmonics travels all the way to the higher frequencies, since these lower harmonics have a larger amplitude, so will their aliased components. This is a main problem, given that the harmonic can be easily mistaken by an aliasing peak that is sitting very close to it. Refer to Figure 29, where the 7th harmonic peaks has an aliased component at its right that has almost the same amplitude.

![Sawtooth FFT with Specified Aliasing and Harmonics](image)

Figure 29: Sawtooth FFT with Specified Aliasing and Harmonics. Fundamental frequency of $f_c = 2988.2813\, Hz$, $f_s = 48\, kHz$, and $NFFT = 32,768$. 8 harmonic peaks.
Figure 30: **BLEP Sawtooth FFT Specified Aliasing and Harmonics.** Fundamental frequency of $f_c = 2,988.2813\,Hz$, $f_s = 48kHz$, and $NFFT = 32,768$. 8 harmonic peaks. $m = 8$ points modified with the BLEP method.
Figures 29 and 30 show the respective selection of harmonic peaks and aliasing peaks for the trivial case and the BLEP modified case. In order to correctly choose or identify the harmonic peaks in the FFT spectrum it is necessary to look at the amplitude of the harmonics of the original trivial signal. In the specific case of the sawtooth these are identify looking at amplitudes that are of nature of $1/N$, being $N$ the number of the harmonic, with $N = 1$ being the fundamental frequency and a normalized amplitude of 1. So when the normalized amplitude is equal to $1/N$, this is set to be the exact $N$th harmonic peak. The total number of harmonics is determined by Equation 21.

$$n_{\text{harmonics}} = \left\lfloor \frac{f_s}{2f_c} \right\rfloor$$  \hspace{1cm} (21)

Notice that Equation 21 includes the last harmonic, which in some cases is not included given that it might land right at the Nyquist frequency. But this is not a problem for this implementation, given the fundamental frequency values analyzed will never have a harmonic right at the Nyquist frequency.

Due to the specific limitations of this implementation and considering the sampling rate and that the modulo counter that generates the trivial sawtooth rarely hits the amplitude exact to 1, the harmonics are not always an exact multiple of the fundamental, sometimes they have a margin error of a few Hertz.

3.6.4 The Harmonic Envelope Decay

As seen in Figure 29 and 30, the normalized FFT of the trivial version only defers of the FFT of the BLEP modified waveform in a decay on the
harmonics but not the fundamental. This is not entirely true. For the normalized case, you only see a difference in the reduction of aliasing and a small decay only at high frequencies. The real effect of the BLEP method is that it modifies the relation of the harmonic peaks amplitude in a manner that depends on the input used, and it also affects the amplitude of the fundamental, sometimes making it higher and sometimes lower. See the comparison using the regular Blackman window in the linear domain in Figure 31. In this case, the fundamental, as well as all the harmonics peaks have a lower amplitude compared to its trivial version.

Figure 31: **Trivial vs. BLEP Sawtooth FFT (Blackman) - Higher Amplitude in Trivial.** Linear domain with a fundamental frequency of $f_c = 2,988.2813$, $f_s = 48kHz$, and $NFFT = 32,768$. With $m = 8$ points modified with the BLEP method. If you look at this comparison for a different, narrower window than the Blackman, you actually see that the BLEP modified version has some higher
peaks at the fundamental and the first harmonics, and much more reduced peaks at the highest frequency harmonics. See Figure 32. This effect is thoroughly analyzed in the next chapter of Evaluation.

![Trivial vs. BLEP Frequency Domain (Spectrum) for fc = 2988.2813 Hz](image)

Figure 32: **Trivial vs. BLEP Sawtooth FFT (Narrower Win.) - Higher Amplitude in BLEP.** Linear domain with a fundamental frequency of $f_c = 2,988.2813$, $f_s = 48\, kHz$, and $NFFT = 32,768$. With $m = 8$ points modified with the BLEP method. The window used is narrower than the Blackman in its genera shape.

### 3.7 Additive Synthesis

Once the harmonics are obtained, it is very simple to implement an alias free additive synthesis version of the waveform. Two options are considered, a version using the diminished high frequencies amplitudes and a version with the correct trivial undiminished amplitudes.

The idea of additive synthesis in the discrete domain is to have an infinite sum of one sinusoid for each of the harmonics of the sound desired. A theoretical
analog sound would have infinite harmonics. The human hearing goes up to 20kHz in the best cases, especially young people, and with a sampling rate of 48kHz, a range of up to 24kHz is obtained. This range is sufficient to create an alias free perfect version of the sawtooth waveform, having reduced an infinite problem which could not be implemented to a finite sum of sines which definitively can be. This idea was first introduced in [Chaudhary, 1998]. This can be achieved using Equation 22.

\[ x_{\text{additive}}[k] = \sum_{i=1}^{n_{\text{harmonics}}} A_i \sin(2\pi f_i k + \phi_i) \]  

(22)

Where \( A_i \) is the amplitude of the harmonic \( f_i \), and \( \phi_i \) is its phase. These values are obtained from the DFT of the signals directly through the complex value the FFT delivers for each specific frequency. The correct values are chosen with the method explained in the previous section on how to differentiate the harmonic peaks from the aliasing peaks. The amplitude is the double of the absolute value of the complex number, this is to compensate for the second half of the FFT corresponding to the negative frequencies. The phase is the same phase of the complex number, there is no need for compensation in this case. Only the harmonics are incorporated in this sum. In additive synthesis is very important to have a defined approximation of the \( \sin \) function. Matlab® has a fairly good approximation of this, so no \( \sin \) table is needed in this case.
3.7.1 Additive Synthesis based on Trivial Waveform Harmonics

If the additive synthesis method is based in the trivial waveform harmonics, it means that it is using the amplitude and phase of the textbook sawtooth of $1/N$. For further analysis, this means that the comparison between the BLEP waveform and its additive version will not only include the level of aliasing but also the reduction or amplification in the amplitude peaks of the harmonics, specially at high frequencies. The location of the harmonics are extracted from the FFT of the BLEP waveform to not introduce an error to the analysis other than the aliasing effects and harmonic decay.

![Trivial Additive Synthesis Time Domain for fc = 2988.2813 Hz](image)

Figure 33: **BLEP Sawtooth with Additive Synthesis based on Trivial Amplitudes**. Fundamental frequency of $f_c = 2988.2813\,Hz$ and $f_s = 48kHz$, using the amplitudes of the trivial sawtooth. $m = 8$ points of correction for the BLEP method.

Figure 33 shows that the reconstructed trivial signal does not look the
same as the original trivial signal. There is some clear rippling in the ramp, that
gets higher as it reaches the discontinuity on both sides. This is due to the sum
of sines with which the additive synthesis reconstructed the signal.

Figure 34: **BLEP Sawtooth FFT with Additive Synthesis based on Trivial Amplitudes.** Fundamental frequency of $f_c = 2988.2813\text{Hz}$, $f_s = 48\text{kHz}$, and $N_{FFT} = 32,768$. Using the amplitudes of the trivial sawtooth. $m = 8$ points of correction for the BLEP method.

Figure 34 shows the perfect harmonics of the sawtooth which correspond
to an amplitude of $1/N$ for each of the harmonics. Given that the graph scale is
in dB, the relationship is not $1 : 1/N$, but the effect is clearly seen. This is the
reason why the last harmonics appear to have very similar amplitudes. This case
is used for the analysis of both the harmonic envelope and the aliasing errors.
3.7.2 Additive Synthesis based on BLEP Waveform Harmonics

The second alternative is to implement an additive synthesis version based on the amplitude and phase of the BLEP modified waveform. This is mainly done to avoid the error of the harmonic envelope, if just considering the aliasing is the goal. In this approach, the location of the harmonics are also extracted from the FFT of the BLEP waveform, as in the previous method.

![BLEP Additive Synthesis Time Domain for fc = 2988.2813 Hz](image)

Figure 35: **BLEP Sawtooth with Additive Synthesis based on BLEP Amplitudes.** Fundamental frequency of $f_c = 2988.2813\,Hz$, $f_s = 48kHz$, and $NFFT = 32,768$. Using the amplitudes of the BLEP sawtooth. $m = 8$ points of correction for the BLEP method.

Figure 35 shows the reconstructed signal for the BLEP modified waveform. This version has the same amplitude as the original, the only missing component is the aliasing. The phase and amplitude of the harmonics is the same, and this is the reason why the two waveforms are very similar. This does
not occur in the trivial case.

Figure 36: **BLEP Sawtooth FFT with Additive Synthesis based on BLEP Amplitudes.** Fundamental frequency of $f_c = 2988.2813\, Hz$, $f_s = 48kHz$, and $NFFT = 32,768$. Using the amplitudes of the BLEP sawtooth. $m = 8$ points of correction for the BLEP method.

Figure 36 shows that the FFT of the reconstructed signal in fact presents no aliasing at all, leaving only the harmonics and having a perfect sawtooth waveform. This case is used for analysis of aliasing effects only, because it is keeping the same harmonic decay error introduce with the BLEP method.

### 3.8 Time Domain Analysis

The analysis over the time domain is not very important in this research, given that the goal is to improve the frequency response of the oscillator. Nevertheless, it is important to observe what the BLEP method does to the trivial sawtooth and how it compares to its corresponding additive synthesis.
version.

Figure 37: **Time Domain Analysis.** Comparison of Trivial and BLEP waveforms (top), Trivial and its Additive Synthesis Version (bottom).

Figure 37, at the top, is comparing the trivial sawtooth and the BLEP modified version. It is clear to see that the BLEP waveform only changes around the discontinuity and it is inserting a line that is shaped like the BLEP residual. This section replaces the hard and sudden change from 1 to -1, for a more subtle change. Discussion on this waveform was explained in the previous Implementing the BLEP waveform section. At the bottom, the trivial case and its additive synthesis version is observed. You can see the rippling effect from the sum of sines.
3.8.1 BLEP and Additive Synthesis Similarities

Notice that the BLEP version and the additive synthesis are very similar around the discontinuities. This is the explanation on time domain of why the BLEP method reduces the aliasing, it is basically emulating the sum of sins effects in the time domain. The more points the BLEP corrects, the more it looks like the additive trivial sawtooth waveform. This can be seen in Figure 38. At the bottom, the BLEP waveform is compared to the additive textbook version, which is the same as the trivial additive version. Notice that the BLEP waveform is more similar to the additive trivial than the actual trivial waveform. This is why the aliasing is reduced when the BLEP method is used. At the top, the BLEP is compared to its additive version using its same peak amplitudes and not the trivial ones. The reconstruction is almost perfect. The more the BLEP looks like its corresponding additive version, the less aliasing there is. The more the BLEP looks like the trivial additive version, the less aliasing and the better harmonic envelope performance there are.

3.9 Analysis of Only the Harmonic Envelope

In order to improve the results of the BLEP method, a frequency domain analysis must be made to evaluate and compare the performance of the different ways for improvement. The most common way to do this is the maximum absolute harmonic error. Another option is the RMSE (Root Mean Squared Error). A new way is presented here to evaluate the entire audible range of the frequency response in the next section.
3.9.1 Maximum Harmonic Absolute Error (MaxAbsHarmErr)

The maximum harmonic absolute error corresponds to the maximum difference of the harmonic amplitudes of the trivial (textbook) waveform and the BLEP waveform. As seen in Figure 39, the BLEP method introduces a harmonic envelope decay at the upper harmonics while sometimes boosting the lower harmonics. In dB, this effect converts to only decay in the upper harmonics when normalization is applied. Each harmonic amplitude is compared with its corresponding textbook version, and the maximum of these differences is the maximum harmonic absolute error. The smaller the error is, the better the BLEP method performance is in terms of harmonic envelope decay.

Figure 38: Time Domain Analysis. Comparison of BLEP and its Additive Synthesis Version (top) and BLEP and its Additive Synthesis Textbook Version (bottom).
3.9.2 Root Mean Squared Error (RMSE)

The RMSE is yet another way to measure the error in the harmonic peaks between the BLEP corrected version and the trivial case. It is very similar to the maximum absolute harmonic error in that it uses the same input values, but the RMSE has a completely different behavior. Rather than just finding the maximum error, it makes an entire sum of the squared difference between the trivial and the BLEP harmonic amplitudes. It then is normalized by the number of elements and its squared root is the final RMSE value. It is noted in dB as well.
3.10 Error in Harmonics and Aliasing Amplitudes (A-Weighted)

A new approach is introduced in this research to evaluate the output of the BLEP method. The idea is to develop a very efficient method that requires a minimum number of calculations and gives at least a general idea of the performance of the BLEP modified waveform in the frequency domain. Given that the aliasing is spread over the entire audible range, the total aliasing perceived by the human ear is not the same as the actual aliasing that is occurring. Tones with the same amplitude at different frequencies have a different perceived loudness, which makes it frequency dependent. The A-Weighted scale is obtained from the Hearing threshold curves of Fletcher-Munson. This gives a more accurate value for the loudness of each aliased component. The A-Weighting factor is described in Equation 23. With this equation you can convert an unweighted amplitude to an A-weighted amplitude directly.

\[ A_{A\text{-Weighted}} = A_{Unweight} \cdot \frac{12200^2 f^4}{(f^2 + 20.6^2)(f^2 + 107.8^2)(\sqrt{f^2 + 737.9^2})(f^2 + 12200^2)} \cdot 10^{0.1} \]  

(23)

Note that this model does not include the effects of frequency masking. It was found that the method described above has a similar output to the PEAQ value (ODJ), even though this last one does include masking effects.

3.10.1 Aliasing Only

After the correct weighted amplitude for each aliased component is calculated, it is then added over the entire audible range and a total aliased value is obtained. This is a linear value, it is necessary to convert to dB in order to
have a correct analysis. After the values is calculated, it is then compared to the same value obtained for the trivial case, which is the worst aliasing scenario for each fundamental frequency. The absolute value of the difference is obtained, and the smaller the value in dB is, the more successful the reduction in aliasing was while using the BLEP method.

3.10.2 Harmonic Decay Only

The harmonic decay error based on the A-Weighting factor is calculated in the same way as with the aliasing. The only components that are corrected and the added are the harmonic peaks. The total is compared to the trivial version total and the value in dB is obtained. The smaller this value is, the closer harmonic performance is obtained with that particular BLEP method to the trivial or textbook envelope.

3.11 Hearing Threshold and Spread Masking (Bosi)

A method to evaluate the performance in the frequency domain that included the hearing threshold effects, as well as the frequency masking effects was developed [Bosi, 2003]. The idea is to calculate a curve of the maximum values between the hearing threshold and the spread masker for each harmonic. Any component above this curve will be heard by the human ear, and any component under will not, due to either loudness or masking effects.

The frequency values of \( f \) in Hz were converted to Bark using the Equation 24.

\[
bark = 13 \tan^{-1}(0.00076 f) + 3.5 \tan^{-1}\left(\frac{f}{7500}\right)
\]  
(24)
The human hearing threshold curve can be approximated by Equation 25 for each fundamental frequency \( f_c \) [Terhardt, 1979].

\[
SPL_{dB} = 3.64\left(\frac{f_c}{1000}\right)^{0.8} - 6.5 \exp\left(-0.6\left(\frac{f_c}{1000} - 3.3\right)^2 + 10^{-3}\left(\frac{f_c}{1000}\right)^{-4}\right)
\]  

(25)

In order for the spread masking to work, the calculation has to be done in the Bark domain rather than in the frequency (Hz) domain. The asymmetric spread masking function for each fundamental frequency can be obtained by Equations 26, 27 and 28.

\[
SM_{i,j}(dB) = A_i(dB) - L_{downshift}(dB)
\]

\[
+ (-27 + 0.37 \max((A_i(dB) - L_{downshift}(dB)) - 40, 0)\theta_{\Delta Z})|\Delta Z|
\]  

(26)

\[
\Delta Z = bark_i - bark_j
\]  

(27)

\[
\theta_{\Delta Z} = \begin{cases} 
0 & \text{when } \Delta Z < 0 \\
1 & \text{when } \Delta Z \geq 0 
\end{cases}
\]  

(28)

Where \( i \) refers to the bark frequency of the masker, in this case all the harmonic peaks for a particular fundamental frequency, therefore \( A_i \) is the amplitude of each harmonic peak. The index \( j \) refers to the maskee which is the sweep of the entire audible range in bark scale. It is important to note that the tone has to be emitted at a 96dB loudness. This means that the fundamental component has to have a 96 dB amplitude in order to correctly analyze the hearing threshold loudness. This 96 dB comes from the amplitude of the common sine tone oscillating between 1 and -1 [Nam et al., 2010]. It is also important to state that the masker has a -10 dB downshift factor corresponding to \( L_{downshift} \).
for pure tones which is what the pitched oscillator creates [Lagrange and Marchand, 2001].

This method delivers a matrix $SM_{i,j}$ which is then compared with the Hearing Threshold curve to find the maximum value between the two. Note that the spread masking function covers all the audible range for each harmonic, meaning that one curve is calculated for each harmonic, explaining why a matrix and not a vector. Which means that the overall masking curve with only its maximum values, from each individual harmonic, is the one compared to the hearing threshold levels to obtain the final Bosi curve.

Two important outputs are obtained from this method. First is a figure

![Bosi Spectrum for fc = 2988.2813 Hz](image)

**Figure 40: Bosi Method in General.** Fundamental frequency $f_c = 2,988.2813 Hz$, $f_s = 48kHz$, and $NFFT = 32,768$. Comparison of Hearing Threshold ( - - black), Maximum masking curve ( – green), the minimum curve between the two (– blue), and harmonic peaks (green circle)
Figure 41: **Bosi Method for Trivial Case.** Fundamental frequency $f_c = 2,988.2813 \, \text{Hz}$, $f_s = 48 \, \text{kHz}$, and $NFFT = 32,768$. Where the harmonic peaks (green), the aliased components (red), the Bosi curve (blue), the harmonic peaks (green circle), and the aliased peaks (red cross) are compared.
Figure 42: **Bosi Method for BLEP.** Fundamental frequency $f_c = 2,988.2813\,Hz$, $f_s = 48kHz$, and $NFFT = 32,768$. Where the harmonic peaks (green), the aliased components (red) and the Bosi curve (blue), the harmonic peaks (green circle), and the aliased peaks (red cross) are compared.
where you can see how the aliasing is distributed over the entire audible range and how the hearing threshold and the masking effects cover it. The idea is to not have any aliasing peaks above this curve, something that can be graphically observed. The second output is that based on the first output, a maximum fundamental frequency is obtained where the alias is perfectly covered by the curve and therefore, it is perceptually alias free, given a performance grade for each different implementation of BLEP.

It is possible that more fundamental frequencies above are perceptually alias free, but the maximum is calculated based on the first (lowest) fundamental frequency where aliasing occurs above the curve. The method behaves similarly to the NMR method described in the next section.

3.12 Noise-to-Mask Ratio (NMR)

The Noise-to-Mask ratio is a measure commonly used while performing perceptually evaluation on audio systems. The idea is very similar to the one from Bosi. It uses a spread masking function to introduce the effects of frequency masking and uses a frame by frame analysis [Brandenburg, 1987]. The main difference with the Bosi method, is that instead of giving a lot of information. It only delivers a single value in dB, which behaves like a noise ratio, the lower, the better the signal is.

The NMR method takes two inputs: the reference signal and the test signal. The method considers the test signal to always be of less quality than the reference signal. The error signal is calculated as the difference between the two
inputs. The second step is to calculate the FFT, of size of 1,024 using the Hanning window, of the error signal and the reference signal. The FFT spectrum is divided in critical bands specific for each sampling rate (in this research a 48 kHz sampling rate is maintained throughout all the analysis). The next step is to calculate the average energy of each critical band. Afterwards, it is scaled by dividing it by its size in bins and finally converted to dB. The following step is to introduce the masking effect, this is done using a spreading function, also specific to each sampling rate, which copies a fraction of the energy from each critical band to all its neighboring bands. The hearing threshold is added to the energy of each band. The total error is obtained adding up all its critical bands, so finally the NMR value in dB is the ratio of the error to the mask threshold. Any NMR value that is below -10 dB is considered to be good, given that a ratio of at least that amount perceptually hides the noise (aliasing effects) very well.

The first implementation of the NMR evaluation method only includes the aliasing effects. This is because the reference sample chosen is the additive version of the BLEP waveform based on its own harmonic amplitudes, therefore will have the same harmonic envelope. This one will be referred as NMR.

The next implementation includes both aliasing and harmonic envelope effects. The reference used here is the additive version of the BLEP based on the textbook waveform which has the original harmonic envelope. This last one will be referred as NMR-TR (Trivial or Textbook Reference).
3.13 Perceptual Evaluation of Audio Quality (PEAQ)

The PEAQ method is a well-known audio evaluation standard ITU-R BS.1387-1 [International Communications Union ITU, 1998]. It has been since then a commonly used reference for perceptual evaluation of sound. The PEAQ method mixes several models, including the previously mentioned NMR.

It takes two signals as input: the reference signal and the test signal. The idea is to find how much difference is there between the reference signal, which is the one known to be good, and the test signal, which is the one known to have a change respect to the reference signal. The method considers the test signal to always be of less quality than the reference signal.

It is very complex and involves a lot of different grade values obtained from different models such as: Disturbance Index (DIX), Noise-to-Mask ratio (NMR), Objective Audio Signal Evaluation (OASE), Perceptual Audio Quality Measure (PAQM), PERCeptual EVALuation (PERCEVAL) and Perceptual Objective Measurement (POM). For this reason, in this research, an implemented model at the McGill University in Canada is used [Kabal, 2003]. Since this implementation is from 2002, it had to be updated to the current Matlab® standards, some functions have to be modified or replaced. Implementing the PEAQ method varies with the sampling rate of the input signals, just like the NMR. The implementation was done making sure that it worked with sampling rate of the BLEP and Trivial samples (described in previous sections), so there was no need to resample or down-sample and avoid any error in the calculations.
Table 5 show the final output of the PEAQ method, which is called the Objective Difference Grade (ODG). It is an average of the output of all the different models inside the PEAQ model. A ODG of 0, means the testing sample has no difference with the reference sample. The lower the grade is, the worst the comparison is perceptually, between the testing and the reference samples.

<table>
<thead>
<tr>
<th>Impairment</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>Imperceptible</td>
<td>0.0</td>
</tr>
<tr>
<td>Perceptible, but not annoying</td>
<td>-1.0</td>
</tr>
<tr>
<td>Slightly annoying</td>
<td>-2.0</td>
</tr>
<tr>
<td>Annoying</td>
<td>-3.0</td>
</tr>
<tr>
<td>Very annoying</td>
<td>-4.0</td>
</tr>
</tbody>
</table>

Table 5: Grading Scale for the PEAQ Objective Difference Grade (ODG). Grading scale for PEAQ output (ODG) [International Communications Union ITU, 1998].

Just as in the NMR model, there are two ways to implement the PEAQ model. The one using the regular BLEP amplitudes for the additive synthesis version will be called PEAQ, and the one using the textbook or trivial additive synthesis version will be called PEAQ-TR (Trivial or Textbook Reference).

### 3.14 Comparison of Analysis Strategies

Table 6 shows a comparison of all the evaluation methods used in this research to find the optimal input for the BLEP method. All of them are useful and all of them are utilized in different ways to obtain the best solution possible. The comparison is made in terms of the type of analysis it includes and the type of psychoacoustical effects it can simulate.
<table>
<thead>
<tr>
<th>Strategy</th>
<th>Analysis</th>
<th>Effects</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Aliasing</td>
<td>Harm. Envelope</td>
</tr>
<tr>
<td>MaxAbsHarmErr</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>RMSE</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>A-Weighted</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Bosi</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>PEAQ</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>PEAQ-TR</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>NMR</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>NMR-TR</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

Table 6: **Comparison of Evaluation Strategies.** In terms of analyzing aliasing error, harmonic envelope error or both; and simulating hearing threshold effects, masking effects, or both.

### 3.15 Genetic Algorithm for Obtaining the Optimal BLEP Residual

The main innovative idea of this research is to develop an algorithm which finds the best BLEP residual table to work at a certain fundamental frequency or below where the alias is perceptually negligible. As an option, the table size, the number of zero crossings of the original sinc function, and the number of BLEP points to correct can be set as an input argument for the algorithm.

#### 3.15.1 Genetic Algorithm (GA) Overview

The Genetic Algorithm (GA) is a method of heuristic search, based on natural selection. It can solve constrained and unconstrained optimization problems that other optimization algorithms cannot. Different than a regular algorithm, it generates several points at each iteration instead of only just one. The best of these points is one step closer to the optimal solution. It then advances to the next iteration by selecting the next population with random number generators, instead of using deterministic computation like regular
3.15.2 Fitness Function

The fitness function is the most important element of the GA. It is the function that evaluates the modified input by the GA and gives the corresponding output. It is essentially the feedback the GA obtains to choose a better set of variables that get closer to the optimal solution. The GA is only as good as its fitness function.

A fitness function has to be efficient because it is run many times on each iteration to the GA. This function also has to be able to give a really good approximation or estimation of the evaluation of the input. The purpose of the GA is to minimize the output of the fitness function to an optimal solution.

For this particular case, the fitness function can be implemented based on different evaluation methods. In order for the function to be time efficient and have more clear result, it has to be related to only one evaluation model. Having several methods of evaluation inside one fitness function is not time efficient and the GA would not be as effective. The question is which of the available options is a better fit for the GA. There are four main options: fit it to the A-weighted, PEAQ or PEAR-TR, NMT or NMR-TR, or the Bosi method.

Fit to A-Weighted

For the first testing phase of the GA, the fitness function used the A-weighting method. This approach gave good results and it reduced the A-weighted aliasing factor considerably, but when the optimal input was
submitted to the other evaluation methods, it did not perform very well. This is the reason why this approach was not chosen, even though a reduced aliasing was found, the distribution of it in the audible range did not meet the expectations of improvement when tested the other methods. Given that the other evaluations have proved to be good indicators, and are based in more advanced perceptual models, it can be said that the output of this fitness function is not as revealing or efficient as the ones the other methods would give. The limit of the optimization of this approach is the A-weighted value of the aliasing of the Trivial case, if it is reduced completely, then the signal would be optimal and perfect. This value varies for each fundamental frequency, which is a problem for running the algorithm throughout all the audible range.

**Fit to PEAQ**

The behavior of the A-Weighted fitness function and the PEAQ fitness function are very similar, when comparing the outputs of both approaches, the rank of the output in terms of performance were very close, almost equal. The PEAQ-TR behaved in the same manner but with a lower grade, in a range of 1 to 2 grades lower than the regular PEAQ. This was something expected as it includes also the effect of the decay of the harmonic envelope. The optimal value for this approach would be 0, which is the best grade, so the function had to be inverted in order to minimize this value, taking -4 as 4, and 0 as the minimum desired limit. One of the advantages of this method is that its close response from 4 to 0 is better than the open response of the A-weighted approach. The
disadvantage of the PEAQ output is that the value 0 is never reached, because it
would mean that the aliasing was eliminated completely and there was no
harmonic decay in the case of the PEAQ-TR. This never happens, given the
nature of the Quasi-Bandlimited approach of the BLEP. For these reasons and
for some inconsistencies explained in the Evaluation chapter, this method was
not optimal for the GA either.

**Fit to NMR**

The NMR based function was the third approach in testing before
implementing the GA. The results of this approach proved to fit accordingly to
the other evaluation methods, given a better response than the PEAQ and the
A-weighted approaches. The NMR-TR behaved very similar to the NMR, but it
had a grade around 5 to 15 dB lower. This is due to the error introduced in the
harmonic envelope decay.

The main disadvantage of the NMR is that we do not know the optimal
value desired. A NMR below 10 dB is considered good, but there could be a
better solution which is even lower, so there is no way to know when or how to
stop the GA. The main reason for not choosing this method, is that the fitness
function based on the Bosi curve, explained in the next section had better results
and constraints.

**Fit to Bosi**

The last approach on the list was to fit the fitness function to the Bosi
method. This was the more successful approach out of all four, hence, it is the
one used in the final implementation of the GA. This approach solves all the problems of the previous approaches, in the terms that its output behaves well with the other methods, and the most important reason is that it is very easy to find the optimal value of the fitness function and stop the GA. This optimal value is 0, referring to 0 aliasing peaks above the curve. When this occurs the signal is considered perceptually alias free, thus it is an optimal solution.

The only worrying issue is that the GA would have to make more calculations because the output it is trying to minimize is discrete (integers) only, corresponding to number of peaks, so the minimization would jump (i.e from 3, to 2, to 1, to 0), loosing all the optimization in between.

The implementation of this fitness function was based on choosing a particular fundamental frequency $f_c$ and iterate until it finds an optimal solution with 0 aliasing peaks above the curve. In order to construct this function, the following steps have to be taken:

1. Given the modified input provided by the GA, the BLEP table is implemented. The variables necessary for this are only the sampling frequency $f_s$ of the sinc function and the size of the table $N$. The number of zero crossings $n$ is sometimes used as a variable for the table.

2. The trivial waveform at that particular $f_c$ has to be implemented with the modulo technique. This step is equal for all the iterations, so it can be performed only once for efficiency. The length of the output is the same as the $NFFT$ size. This step uses the sampling rate for the final waveform $f_s$. 
3. That trivial waveform is then modified with the BLEP method as described above. For this it is necessary to have the table from step 1, the trivial waveform from step 2, and the number of points to correct $m$.

4. The BLEP waveform is then transformed to the Frequency domain using the DFT method described in the previous sections. It uses the same $NFFT$ as the length of the waveform. In order for the Bosi method to function correctly, the FFT output has to be in the linear domain, not in the dB scale. It is also important that the raw FFT values are used. When modified results are used, it will affect the results of the Bosi method.

5. The harmonic peaks and aliasing peaks are separated in the FFT, with their corresponding frequency locations.

6. The results from step 5 are the only input for the Bosi method. It then calculates the spreading masking function for that specific set of harmonics and the hearing threshold for the entire audible range.

7. The evaluation value, or output, of the fitness function is the total number of aliasing peaks above the Bosi curve.

3.15.3 Input Function Creation

The final modified input consists of two main parts: the window creation and the $sinc$ function portion chosen. This input, was set up to have 35 variables minimum, given by the GA. The explanation of each part is described below.

The idea is to modify the BLEP method input, which is the $sinc$ function with a modified window of different sizes, covering the $sinc$ function at exact zero
crossing locations on each side (i.e. 1, 2, 3, ...), and a modified shape.

**Window creation**

The shape of the window is created by mixing 17 different windows with a factor coefficient and a power coefficient for each, finally normalizing the result so it has the properties of a regular symmetric window. This final window has a maximum amplitude of 1 at the y-axis crossing. Equation 29 shows the combination of each of the 17 windows into the final window.

\[
w_{\text{mixed}}[k] = \sum_{i=1}^{17} a_i (w_i[k])^{b_i}
\] (29)

Table 7 shows the ID number of each window and what type of window it is. It also shows the location of the coefficients inside the GA resulting vector positions from 1 to 34.

All the windows are symmetric and have their maximum value at the center with an amplitude of 1 or 100%. The first and last points for most of the windows is 0, but some of them do not decay all the way to zero like the rectangular window, the Gaussian window, the Hamming window, the Kaiser window. The Taylor window does not start or end at 0, and it also does not have 1 as its center maximum value. The rectangular window basically acts as an offset. Some of the windows have even negative effects in between the first sample and the center sample like the flattop window. The Kaiser window, the Taylor window, and the Gaussian window have an extra parameter which can be modified to change their shape. All these properties can be observed in Figures
<table>
<thead>
<tr>
<th>ID</th>
<th>Window Type</th>
<th>$a_i$</th>
<th>$b_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Bartlett</td>
<td>1</td>
<td>18</td>
</tr>
<tr>
<td>2</td>
<td>Bartlett-Hanning</td>
<td>2</td>
<td>19</td>
</tr>
<tr>
<td>3</td>
<td>Blackman</td>
<td>3</td>
<td>20</td>
</tr>
<tr>
<td>4</td>
<td>Blackman-Harris</td>
<td>4</td>
<td>21</td>
</tr>
<tr>
<td>5</td>
<td>Bohman</td>
<td>5</td>
<td>22</td>
</tr>
<tr>
<td>6</td>
<td>Dolph-Chevyshev</td>
<td>6</td>
<td>23</td>
</tr>
<tr>
<td>7</td>
<td>Flattop</td>
<td>7</td>
<td>24</td>
</tr>
<tr>
<td>8</td>
<td>Gaussian</td>
<td>8</td>
<td>25</td>
</tr>
<tr>
<td>9</td>
<td>Hamming</td>
<td>9</td>
<td>26</td>
</tr>
<tr>
<td>10</td>
<td>Hanning</td>
<td>10</td>
<td>27</td>
</tr>
<tr>
<td>11</td>
<td>Kaiser</td>
<td>11</td>
<td>28</td>
</tr>
<tr>
<td>12</td>
<td>Nuttall</td>
<td>12</td>
<td>29</td>
</tr>
<tr>
<td>13</td>
<td>Parzen</td>
<td>13</td>
<td>30</td>
</tr>
<tr>
<td>14</td>
<td>Rectangular</td>
<td>14</td>
<td>31</td>
</tr>
<tr>
<td>15</td>
<td>Taylor</td>
<td>15</td>
<td>32</td>
</tr>
<tr>
<td>16</td>
<td>Triangular</td>
<td>16</td>
<td>33</td>
</tr>
<tr>
<td>17</td>
<td>Tukey</td>
<td>17</td>
<td>34</td>
</tr>
</tbody>
</table>

Table 7: **List of Window Pool.** Comparison of windows and its coefficients locations inside the GA vector. The factor coefficients go from 1 to 17, while the power coefficients go from 18 to 34.
Figure 43: **Comparison of Several Widows (Part 1)**. Windows compared: Bartlett (blue), Bartlett-Hanning (orange), Blackman (yellow), and Blackman-Harris (purple).

Figure 44: **Comparison of Several Widows (Part 2)**. Windows compared: Bohman (blue), Dolph-Chebyshev (orange), Flattop (yellow), and Gaussian (purple).
Figure 45: **Comparison of Several Widows (Part 3)**. Windows compared: Hamming (blue), Hanning (orange), Kaiser (yellow), and Nuttall (purple).

Figure 46: **Comparison of Several Widows (Part 4)**. Windows compared: Parzen (blue), Rectangular (orange), Taylor (yellow), Triangular (purple), and Tukey (green).
43, 44, 45, and 46, where the windows are plotted separately in groups of four or five in order to be able to appreciate each particle window.

Having more windows means having a greater pool of where to find the answer. This translates to more computational processing, but also to a greater possibility of finding an optimal solution. Different pool sizes were considered including just some of the windows mentioned in Table 7. One of these many considerations was to modify only the best windows that fit the BLEP method which were Blackman, Blackman-Harris, and Dolph-Chebyshev. This solution gave good results but the pool of 17 windows was even better. This is because each window delivers a different and unique factor to the optimal answer.

**Modifying the sinc function**

After creating the mixed window, it is then applied to the sinc function. It can have many zero crossings. This number has to be symmetric, meaning that the same number of zero crossings are taken on each side of the sinc Y axis.

When this number varies, it affect mainly the number of ripples in the modified input for the BLEP. This rippling is kept during the entire BLEP process, translating to the BLEP residual and finally to the BLEP modified waveform.

The number of zero crossings \( n \) is considered to be another variable of the GA, given that there was not a clear relationship of this number with any other constant parameter like the size of the table \( N \) of the number of modified points \( m \) for this particular implementation of the BLEP method. Further discussion can be found in the Number of Zero Crossings vs. Number of BLEP Correction.
Points section. The resulting vector from the GA has the variable $n$ as its variable 35.

**Case with more variables**

When the number of points to correct $m$ or the size of the table $N$ are maintained constant during the run of Phase I of the GA, it optimizes the input function of the BLEP for those particular values. Sometimes, the desired result does not depend on these variables, so a more freely GA is ran in Phases II and III, with 37 orders instead of 35. The variables correspond to the position 36 and 37, respectively, of the GA resulting vector. Note that when only one of these variables is constant and the other is not, the latter becomes variable 36 for the GA result.

3.15.4 **Bounds**

When using non-integer values, like numbers of type double, the search space becomes much wider. This results in a longer time to run for the GA and gives different result depending on the time constraints. This might cause sometimes problems compared to having integer values for the variables in the easiest cases. On the other hand, when a more critical solution is required, having integers reduces the search space, and consequently might not get to the optimal solution. For this case a type double for the variables might be more suitable, but the time to run the GA would be of the order of hours or even days. Having only integer values makes the algorithm much more efficient for the simple cases.

For the coefficients of the window, in the case of the factor coefficients, it
was proven that having negative and positive values gave better responses than having only positive values; independently of being of type integer or double.

The chosen bounds were the ones that gave the more efficient solution, as well as the less consuming time implementation. A mixed integer (all integers) solution was optimal for all variables, ranging as show in Table 8.

<table>
<thead>
<tr>
<th>GA Pos.</th>
<th>Variable</th>
<th>Type</th>
<th>Lower</th>
<th>Upper</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:17</td>
<td>Factor Coefficient</td>
<td>Integer</td>
<td>-100</td>
<td>100</td>
</tr>
<tr>
<td>18:34</td>
<td>Power Coefficient</td>
<td>Integer</td>
<td>1</td>
<td>30</td>
</tr>
<tr>
<td>35</td>
<td># of Zero Crossings (Each Side)</td>
<td>Integer</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>36</td>
<td>Table Size</td>
<td>Integer Vector</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>37</td>
<td># of Correction Points (Each Side)</td>
<td>Integer</td>
<td>1</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 8: **Specifications of the Variables of the GA.** Classification of the properties of each variable of the GA by its position in the variables vector, its type, and its lower and upper bounds constraints.

The factor coefficients shown in table 8 translate to a percentage idea of including a 100% of the window properties, or 0%, or even -100% of it. An order of 30 degrees was chosen for the power, because in comparison to fewer or more degrees it showed the better performance. The number of correction points for the method was set to a maximum of 5, this is because at high frequencies, there are less points to modify on each cycle, this depends on the sampling rate $f_s$, and it is restricted by Equation 30.

$$n_{points/period} = \frac{f_s}{f_c}$$  \hspace{1cm} (30)

For a sampling rate $f_s$ of 48 kHz, a maximum of 8 points are found in the BLEP waveform on each period cycle with a fundamental $f_c$ of 6 kHz. This means that at its maximum, the BLEP correction is modifying all 8 sampled
points in each cycle with 4 points of correction on each side. Five points were chosen because it was close to the number of points in the highest note of the MIDI keyboard.

3.15.5 Restrictions

The GA is a very complex algorithm, and because of this, restrictions must be established in order to obtain an optimal result in a finite amount of time, or even know when to stop the algorithm when it has not found an optimal solution in a period of time, or number of calculations.

Population

The population consists of all the set of values to evaluate at each iteration. Out of this population, a set of parents is chosen to be the seed of the next iteration. The size of the population for mixed integer problems is defined in Equation 31. For this particular case, the value is always set to 100 given the large number of variables, totaling 37 in the most complex case.

\[ n_{\text{population}} = \min(\max(10n_{\text{variables}}, 40), 100) \] (31)

Generations

The generations are the number of iterations the GA does in each run, its limit is infinite, it can run forever trying to optimize the minimum value. It is then necessary to set a maximum number of generations for the GA to stop, even if it has not found the optimal solution. The commonly used value is 100 times the number of variables. In this particular case, it would vary from 3500 to 3700.
The fitness function chosen takes a few seconds (2-3) to run each time, which then accumulates over time when running the GA. The maximum number of generations was set to 350, which translates to about 4 hours of calculations using 6 cores in parallel (See the Parallel Computing section).

**Crossover**

The crossover value represents the fraction of the current population at the next generation. It is always a value between 0 and 1. For this case, 0.8 was chosen. A greater crossover value, means a greater number of calculation on each iteration, which accumulates for a greater number of total calculations over one run.

**Fitness Limit**

The fitness limit is the second way to stop the GA, the other one being the number of generations. It represents the optimal value desired if known for the minimal solution. If it is unknown, then negative infinity would be the limit. In the case of the fitness function implemented based of the Bosi model, the limit would be 0. So when the GA solution reaches the number 0, it stops, given this is an optimal solution. Not that this solution is only one of many possible solutions for the same particular case, this is due to the stochastic nature of the GA.

**Stall Limit**

The stall limit is the third option to stop the GA. It is a specific number of generations over which the optimal solution has not been reduced by a
minimum quantity. In the general case, this value is very small, of the order of $10^{-6}$. In this particular case, since the Bosi output only moves in integer steps, the minimum limit would be 1. In this particular case, a limit of 50 generations is chosen, this means that if the solution has not been minimized by at least 1 over 50 generations, the algorithm should stop. The reason for this is that the algorithm would most likely not find a solution after this, so a new random seed should be generated to try and find the optimal solution, which means, running the algorithm again. Sometimes, without modifying the parameters an optimal solution is found, sometimes is not. Since the seed is random for the populations, it is uncertain if there will be or not an optimal solution in a different run.

3.15.6 Obtaining the Results for Regular Windows (Starting Point)

To obtain the results for all the regular windows and the case with no window, a method was developed to be followed exactly like this:

0. Follow all the same steps described for the fitness function to calculate the output of the Bosi method.

1. Calculate the number of peaks above the Bosi curve for a certain window or with no window staring at the first note in the MIDI Table.

2. Do the same procedure for the next value in the MIDI table.

3. If the value is above 0, then we know that the highest frequency is between the current MIDI note and the one before.

4. Go back to the previous MIDI note and start incrementing the fundamental frequency by only one bin size until the value for the aliasing peaks
goes above 0.

5. Finally, the highest frequency is the one previous to the current frequency.

6. Repeat steps 0 to 5 for all combinations of \( N, n \) and \( m \), keeping a constant \( f_s \) and \( NFFT \).

A different value is found for each different FFT size, but they are very close, the largest difference is only related to the greater bin size between the two FFT sizes. This method was implemented to optimize the time and not having to sweep through all the frequency bins in the audible range.

This process has to be ran for all table sizes (\( N \)), all number of zero crossings (\( m \)), and all of the windows in the pool. Remember that in the original case, the number of BLEP correction points is equal to the number of zero crossings (\( n \)). Since a wider analysis was wanted in this research, the process was run for all the different combinations of \( n \) and \( m \). The best results are the taken from this large amount of results, and a best stating point is set for each particular combination of \( N \) and \( m \).

Given the nature of this approach, a lot of calculations are needed. A parallel version of the algorithm was implemented (See the Parallel Computing section), where a different core obtained the results of each different table size. The \texttt{parfor} command was used to run through the for loop of each table size in a parallel manner.
3.15.7 Description of the Method to run the GA based on the Bosi Approach

The first approach was to have the size of the table and the number of points to correct in the BLEP method also as variables to see their behavior. This always showed that the largest table size was preferred and that the number of points to correct depended on the sampling rate. When using a sampling rate of 48 kHz, there are around 9 sampled points per period cycle, at high frequencies (i.e. 5.1 kHz), the best number was to modify 4 samples on each side for a total of 8 samples. Since the idea is to optimize the size of the table, and the number of correction points, this approach is not good enough, even though it does give optimal solutions. This is the case of 37 variables for the GA.

The second approach was to have a fixed table size and number of points to correct with BLEP. The idea is to optimize to the maximum the highest frequency perceptually alias free of the best window available at that specific table size. This is the case of 35 variables for the GA output.

The third approach was to have a fixed table size but a variable number of points to correct. This indirectly includes the second approach, in that the number of correction points of it, is included in this approach as well. This case has 36 variables total.

The implemented approach is a mixture of the second and third approaches. The idea is to get the coefficients for the window for each table size and its corresponding number of points to correct. Remember that the solution depends on the sampling rate. For this case of study a sampling rate of 48 kHz
The chosen implementation or method to run the GA has to be very efficient, in terms of finding an optimal solution for an specific set of table size \((N)\) and correction points \((m)\). It is very simple to obtain an output for a particular case, but this solution is not guaranteed to be good for all the frequencies above, or more importantly, below.

The implementation of the GA was used in Matlab\(^\text{®}\) using the \texttt{ga} function provided in the Global Optimization Toolbox. It is a very efficient and fast code that fits all the requirements for this particular approach to synthesis.

The final implementation of the GA has several steps. They are detailed below:

\textbf{Phase 0 (Pre-Processing)}

1. Obtain the original window results. These results are in a pre-calculated table where only the best results for each \(N\) and \(m\) combination is described. The result corresponds to the highest frequency possible that is alias free obtained using a common window and a common \texttt{sinc} function. This value is the starting point of the optimization, meaning that any improvement to this highest frequency is a better solution than the original BLEP implementation. Refer to the Obtaining the Results for Regular Windows (Starting Point) section.

2. A particular combination of table size \((N)\) and correction points \((m)\) is chosen. This combination already has a value from step 1 which is the highest
frequency alias free. The stating point will be one bin size above this frequency.

**Phase I**

3. The actual first phase of the running starts with the highest fundamental frequency perceptually alias free of all of the different number of points to correct in the BLEP method \((m)\) is chosen and with that same \(m\), the GA is run increasing one bin per each run to find an optimal solution for that new fundamental frequency. When the limit is reached, the GA stops.

**Phase II**

4. The second phase of the GA running is to keep increasing the previous fundamental frequency but changing the number of points to correct \(m\). In some cases, no better results might be found in this stage.

5. A corroborating phase is needed to verify the highest frequency this particular optimal result can get to that is perceptually alias free. The same method used for calculating the highest frequency of the original windows is used (See Starting Point section). The result of this step is not always the same as the ones from step 4, some aliasing might be present at some location before reaching that fundamental frequency, which reduces the actual performance of the input.

6. Run steps 2 to 5 for all \(N\) and \(m\) combinations.

7. Choose the best results out of all the results from step 6 that have the better performance respect to each other and to the original no window and best window cases.
Phase III

8. The output of the GA fitness function is modified. The output count for the aliased components above the curve stays the same. Additionally, when a harmonic peak that was originally above the Bosi curve in the trivial case, and is below the curve in the BLEP modified case is counted as a negative contribution to the output value. This allows the harmonic error decay to also be optimized.

9. Different from Phase I and II, the algorithm is looped through all the MIDI frequencies on each run. This only optimizes the result to the highest MIDI note of the 88-key piano.

10. The best result is the one where no aliased peaks are above the curve and no more harmonic peaks are below than the ones from the trivial case. This solution might not always be found, but it can be reduced to have a very small value of only 1 or 2 undesired peaks, either above or below the curve.

Parallel Computing

Since the GA usually evaluates the fitness function hundreds or thousands of times on each run, it is necessary to have a good computing system set up to have a shorter running time. Evaluating the exact function many times is an ideal case for parallel computing given that the results do not depend on each other, so they can be running at the same time, in parallel.

Fortunately, Matlab® has a Parallel Computing Toolbox that can help with the set up of this. The ga function is also designed to work in conjuncture with the parpool function of the Parallel Toolbox.
4

Evaluation

NOTE: All the results obtained on this section correspond to a fixed sampling rate $f_s = 48kHz$ and a FFT size $NFFT = 16,384$ to produce faster calculations. The sampling rate was chosen to be 48 kHz, because it is a common value in the audio industry and also, the most important analysis of the results are perceptually based. This $f_s$ covers a range of up to 24 kHz which is enough to cover the human audible range (20-20,000 Hz). Larger FFT sizes were also implemented during research, but the results are not displayed as thoroughly as the 16,384 case.

4.1 Results from GA Phase 0 (Pre-Processing)

The first item necessary for the development of the algorithm are the stating points, these are based on the original windows (Refer to the Obtaining the Results for Regular Windows (Starting Point) section in Chapter 3). These are set by the highest alias-free fundamental frequency obtained with a regular window and a regular sinc function.

In order to obtain the best results, the calculations were made for each table size, and each number of correction points using all of the available windows from Table 7 and the original implementation of the BLEP method, where the variables $m$ and $n$ are equal. The best results are obtained for each particular combination of $N$ and $m$ out of all windows tested. The results can be
seen in Table 9. The table sizes chosen were inside a vector containing 8 different sizes formed by blocks of 4,096 points, adding up to 32,768 points. These values were chosen as multiples of 1,024 which corresponds to 1 kB in terms of memory. So the analysis would be from 4 kB to 32 kB.

The Table 9 corroborates the results of the original BLEP method. It indicates that a larger table gives better results. It also shows that a higher correction points number has better results in most of the cases, meaning that at high frequencies, a smaller \( m \) might give better results because fewer samples are available to be modified. These will be the starting points for the runs of the GA. At these particular frequencies and above, the GA will run and it will try to find an optimal solution. Running the GA to find optimal solutions for fundamental frequencies below is not important, because they are not considered an improvement to the original BLEP method. But keep in mind, that the solution found does have to be backwards compatible with all the lower frequencies, this way it is really considered Perceptually Alias-Free. The highest frequency observed in Table 9 is of 3,336.91 Hz and corresponds to a table size of \( N = 32,768 \) with \( m = 4 \) correction points and it is obtained using the Blackman window. This is not higher than the last note on the piano (4,186 Hz approximately), so it would not be possible to implement the complete 88 key MIDI piano being perceptually alias-free.

In an overall view of the best results from Table 9, the Blackman window is the current best fit for the BLEP method, this corroborates the results found
<table>
<thead>
<tr>
<th>N</th>
<th>m</th>
<th>Win ID</th>
<th>$f_c$ (Hz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4096</td>
<td>1</td>
<td>1</td>
<td>331.0547</td>
</tr>
<tr>
<td>4096</td>
<td>2</td>
<td>2</td>
<td>495.1172</td>
</tr>
<tr>
<td>4096</td>
<td>3</td>
<td>2</td>
<td>878.9063</td>
</tr>
<tr>
<td>4096</td>
<td>4</td>
<td>2</td>
<td>878.9063</td>
</tr>
<tr>
<td>4096</td>
<td>5</td>
<td>2</td>
<td>832.0313</td>
</tr>
<tr>
<td>8192</td>
<td>1</td>
<td>1</td>
<td>331.0547</td>
</tr>
<tr>
<td>8192</td>
<td>2</td>
<td>2</td>
<td>495.1172</td>
</tr>
<tr>
<td>8192</td>
<td>3</td>
<td>2</td>
<td>1242.188</td>
</tr>
<tr>
<td>8192</td>
<td>4</td>
<td>2</td>
<td>1400.391</td>
</tr>
<tr>
<td>8192</td>
<td>5</td>
<td>2</td>
<td>1125.000</td>
</tr>
<tr>
<td>12288</td>
<td>1</td>
<td>1</td>
<td>331.0547</td>
</tr>
<tr>
<td>12288</td>
<td>2</td>
<td>2</td>
<td>533.2031</td>
</tr>
<tr>
<td>12288</td>
<td>3</td>
<td>2</td>
<td>2106.445</td>
</tr>
<tr>
<td>12288</td>
<td>4</td>
<td>2</td>
<td>2217.773</td>
</tr>
<tr>
<td>12288</td>
<td>5</td>
<td>3</td>
<td>1429.688</td>
</tr>
<tr>
<td>16384</td>
<td>1</td>
<td>1</td>
<td>331.0547</td>
</tr>
<tr>
<td>16384</td>
<td>2</td>
<td>2</td>
<td>533.2031</td>
</tr>
<tr>
<td>16384</td>
<td>3</td>
<td>2</td>
<td>2217.773</td>
</tr>
<tr>
<td>16384</td>
<td>4</td>
<td>2</td>
<td>2106.445</td>
</tr>
<tr>
<td>16384</td>
<td>5</td>
<td>3</td>
<td>1798.828</td>
</tr>
<tr>
<td>20480</td>
<td>1</td>
<td>1</td>
<td>331.0547</td>
</tr>
<tr>
<td>20480</td>
<td>2</td>
<td>2</td>
<td>533.2031</td>
</tr>
<tr>
<td>20480</td>
<td>3</td>
<td>2</td>
<td>2217.773</td>
</tr>
<tr>
<td>20480</td>
<td>4</td>
<td>2</td>
<td>2642.578</td>
</tr>
<tr>
<td>20480</td>
<td>5</td>
<td>2</td>
<td>2106.445</td>
</tr>
<tr>
<td>24576</td>
<td>1</td>
<td>1</td>
<td>331.0547</td>
</tr>
<tr>
<td>24576</td>
<td>2</td>
<td>2</td>
<td>533.2031</td>
</tr>
<tr>
<td>24576</td>
<td>3</td>
<td>2</td>
<td>3043.945</td>
</tr>
<tr>
<td>24576</td>
<td>4</td>
<td>2</td>
<td>2645.508</td>
</tr>
<tr>
<td>24576</td>
<td>5</td>
<td>2</td>
<td>2217.773</td>
</tr>
<tr>
<td>28672</td>
<td>1</td>
<td>1</td>
<td>331.0547</td>
</tr>
<tr>
<td>28672</td>
<td>2</td>
<td>2</td>
<td>533.2031</td>
</tr>
<tr>
<td>28672</td>
<td>3</td>
<td>2</td>
<td>3134.766</td>
</tr>
<tr>
<td>28672</td>
<td>4</td>
<td>2</td>
<td>3193.359</td>
</tr>
<tr>
<td>28672</td>
<td>5</td>
<td>3</td>
<td>3002.930</td>
</tr>
<tr>
<td>32768</td>
<td>1</td>
<td>1</td>
<td>331.0547</td>
</tr>
<tr>
<td>32768</td>
<td>2</td>
<td>2</td>
<td>533.2031</td>
</tr>
<tr>
<td>32768</td>
<td>3</td>
<td>2</td>
<td>3137.695</td>
</tr>
<tr>
<td>32768</td>
<td>4</td>
<td>2</td>
<td>3336.914</td>
</tr>
<tr>
<td>32768</td>
<td>5</td>
<td>3</td>
<td>3146.484</td>
</tr>
</tbody>
</table>

Table 9: **Highest Fundamental Frequency Perceptually Alias-Free of Regular Windows**. The Win ID value corresponds to Rectangular (1), Blackman (2), and Blackman-Harris (3).
in previous papers [Nam et al., 2010]. Further research will be compared to this window only, being the best available option, and any results better than this will be a truly improvement to the method.

4.2 Results from GA Phase I and Phase II (Aliasing Optimization Only)

The GA was explained in the Implemented Approach of the GA section. It is separated into two phases, explained in detail over the next sections.

4.2.1 Phase I

The GA is run for each particular fundamental frequency. This is because it would not be efficient at the first stage to run it for every frequency below the fundamental due to the time consumption of the Bosi analysis for each frequency. The algorithm was ran a total of 150 times during Phase I, for different $N$ and $m$ combinations. The combinations were chosen by the best values of the starting points. The best values were chosen to be combinations of smaller table sizes, smaller number of correction points and the highest fundamental frequencies from Bosi. For example, the tables with $N = 24,576$ has its best performance with only $m = 3$, rather than $m = 4$ or $m = 5$, giving that it gives a higher Bosi value of $f_c = 3,043.945 Hz$. Observe these results in Table 9.

Not all of the combinations were tried, only the ones yielding the best performance. These results are not displayed here because they are very large and rather not important to the final outcome. The results from Phase II contain the important information obtained from the GA and will be explained
thoroughly in the next section. Additionally, all the results from Phase I were later optimized in Phase II, so they are implicitly included.

4.2.2 Phase II

After all the best combination were finished, the GA was ran again for each $N$ constant, varying the $m$ to see if an even higher fundamental was found, in all cases it did, even if it was by a only a few Hertz. Out of the 214 times ran during Phase II, 102 were successful to find an optimal solution. The running time added up to around 80 hours of computing time. One iteration of the genetic algorithm is shown in Figure 47.

![Figure 47: Output of a run of the GA.](image)

Figure 47 shows the output value, which is the number of aliasing peaks
that are above the Bosi curve. The desired value is 0, so the algorithm stops when this value is reached, at the 40th generation in this particular case. The number of generations varies for every run, the higher the fundamental frequency is, the higher the number of generations is because it is harder to find the optimal solution.

Each of these 102 optimal solutions was then corroborated using the method developed to find the highest fundamental perceptually alias-free (Refer to the Starting Points section in Chapter 3). This checks that all the frequencies below are also perceptually alias-free. In the case that one of them is not, then the lowest frequency is chosen to be the new Bosi value (highest fundamental). The main output of Phase II is a unique highest fundamental for each of the 102 solutions.

**Highest Perceptually Alias Free Fundamental Frequencies**

A total of 88, out of the original 102 optimal solutions from Phase II, had an improvement over the Blackman window. The next step to trim the number of optimal solutions was to find the performance of each solution with all of the available table sizes and number of corrections points available. The method to do this is to find the optimal combination(s) of $N$ and $m$ for each particular solution. The idea behind this is that having certain $N$ and $m$, which is perceptually alias-free in all frequencies below, could only improve (higher fundamental) with a different optimal $N$ and $m$ combination. In the worst case scenario, the current optimal combination will be the best, so it will be the final
optimal solution.

Some of the solutions performed better at lower $N$ and lower $m$, these solutions were considered better than the larger table sizes and correction points, which cuts down the number of optimal solutions even more.

Finally, some solutions overlapped. It was found that a particular solution would have a better performance with an specific $N$ and $m$ than most of the other solutions while keeping these values constant, so all the other solutions were thrown away. This way, the final number of optimal inputs for the BLEP method obtained was 14, all of them with a meaningful improvement over the Blackman window. Out of these, only 12 are unique given that two different inputs gave the same performance in two particular cases. The list can observed in Table 10, which contains the most important information of Phases I and II. Improvements over the Blackman window varied from 33\% to up to 168\%. The most important improvements, were the ones that pushed the BLEP method from being unable to construct the entire MIDI keyboard, to actually being able to have a perceptually alias-free implementation. Solutions 8, 13 and 14 met this requirement.

4.2.3 Input Functions Obtained

The solutions from Table 10 work for a particular table size and number of correction points, but notice that some solutions have the same ID input, but different $N$ and $m$ values. This means that the same input was optimal with two or more different combinations. At the end, only 9 optimal unique inputs are found in the 14 final optimal solutions.
<table>
<thead>
<tr>
<th>N</th>
<th>m</th>
<th>Rect.</th>
<th>Blackman</th>
<th>GA</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>345.70</td>
<td>495.12</td>
<td>43%</td>
<td>788.09</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>413.09</td>
<td>878.91</td>
<td>113%</td>
<td>1872.07</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>413.09</td>
<td>878.91</td>
<td>113%</td>
<td>1872.07</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>413.09</td>
<td>2106.45</td>
<td>410%</td>
<td>3518.55</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>413.09</td>
<td>2106.45</td>
<td>410%</td>
<td>3518.55</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>413.09</td>
<td>2217.77</td>
<td>437%</td>
<td>3969.73</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>413.09</td>
<td>3043.95</td>
<td>637%</td>
<td>4045.90</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>413.09</td>
<td>3134.77</td>
<td>659%</td>
<td>4192.38</td>
</tr>
<tr>
<td>9</td>
<td>4</td>
<td>413.09</td>
<td>878.91</td>
<td>113%</td>
<td>2358.40</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>495.12</td>
<td>1400.39</td>
<td>183%</td>
<td>3169.92</td>
</tr>
<tr>
<td>11</td>
<td>4</td>
<td>495.12</td>
<td>2217.77</td>
<td>348%</td>
<td>3231.45</td>
</tr>
<tr>
<td>12</td>
<td>4</td>
<td>495.12</td>
<td>2642.58</td>
<td>434%</td>
<td>3969.73</td>
</tr>
<tr>
<td>13</td>
<td>4</td>
<td>495.12</td>
<td>3193.36</td>
<td>545%</td>
<td>4447.27</td>
</tr>
<tr>
<td>14</td>
<td>5</td>
<td>495.12</td>
<td>2217.77</td>
<td>348%</td>
<td>4204.10</td>
</tr>
</tbody>
</table>

Table 10: **Highest Fundamental Frequency Perceptually Alias-Free Comparison.** The first column is the number of identification of each solution with its correspondent \( m \) and \( N \), referred as GA 1-14. The fourth, fifth and seventh columns are the highest fundamental frequencies perceptually alias-free for the Rectangular, Blackman and GA solutions respectively, all indicated in Hz. The last column corresponds to the input function ID independent of \( N \) and \( m \). The sixth column corresponds to the improvement of using the Blackman window over using no window at all or Rectangular. The eighth column is the improvement percentage of the GA solution over no window. The ninth column is the improvement of the GA solution over the Blackman window. The solutions are sorted first by number of correction points \( m \) and secondly by table size \( N \). Note that the GA 11 is outperformed by GA 4 and GA 5, but it is included for comparison due to the repetition of input ID 28 in several optimal solutions. Also note that GA 12 is outperformed by GA 6, but it was included for comparison, due that it is very close to the highest note in the piano.
The output of the GA is a vector of 35 variables (See the Input Function Creation section in Chapter 3), with which the input is constructed and then processed to the BLEP residual. In order to construct the input function, the factor and power coefficients are obtained from the GA for each particular solution without considering the table size or the number of correction points. Table 11 shows the factor coefficients for the optimal 9 inputs, while 12 gives the ones for calculating the power (order). Remember, these 9 unique solutions are combined with different optimal combinations of $N$ and $m$ to find the 14 final solutions from Table 10. The 35th variable of the GA output vector corresponds to the number of zero crossings $n$. This value was found optimal at 1 for all solution, for further reasoning about this refer to the Trade Off between Harmonic Decay and Aliasing section.

Figures 48 and 49 show the input functions and the BLEP residuals respectively, for each of the 9 unique solutions. Their behavior is very similar in the time domain, maintaining a similar shape to those of the original windows like Blackman or Blackman-Harris. But their effects on the frequency spectrum are very different as seen in the next section. These solutions are independent of table size or correction points, so they are all shown with a simple window size of 4,096 points.

In order to appreciate the properties of each of the input solutions on its own, the next sections show the optimal solutions for the input function separately for each of the table sizes.
<table>
<thead>
<tr>
<th>ID</th>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$a_3$</th>
<th>$a_4$</th>
<th>$a_5$</th>
<th>$a_6$</th>
<th>$a_7$</th>
<th>$a_8$</th>
<th>$a_9$</th>
<th>$a_{10}$</th>
<th>$a_{11}$</th>
<th>$a_{12}$</th>
<th>$a_{13}$</th>
<th>$a_{14}$</th>
<th>$a_{15}$</th>
<th>$a_{16}$</th>
<th>$a_{17}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>45</td>
<td>32</td>
<td>-68</td>
<td>54</td>
<td>8</td>
<td>-40</td>
<td>-27</td>
<td>56</td>
<td>6</td>
<td>125</td>
<td>69</td>
<td>-47</td>
<td>36</td>
<td></td>
<td></td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>-89</td>
<td>17</td>
<td>15</td>
<td>60</td>
<td>-61</td>
<td>-98</td>
<td>-47</td>
<td>-29</td>
<td>0</td>
<td>2</td>
<td>96</td>
<td>1</td>
<td>19</td>
<td>42</td>
<td>12</td>
<td>19</td>
<td>92</td>
</tr>
<tr>
<td>28</td>
<td>0</td>
<td>2</td>
<td>96</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>93</td>
<td>97</td>
<td>1</td>
<td>99</td>
<td>100</td>
<td></td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>55</td>
<td>-39</td>
<td>-10</td>
<td>-18</td>
<td>42</td>
<td>89</td>
<td>-21</td>
<td>-41</td>
<td>64</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>57</td>
<td>-38</td>
<td>-85</td>
<td>-13</td>
<td>-39</td>
<td>12</td>
<td>19</td>
<td>92</td>
<td>26</td>
<td>74</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>59</td>
<td>-9</td>
<td>46</td>
<td>-93</td>
<td>100</td>
<td>-41</td>
<td>51</td>
<td>21</td>
<td>-82</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>63</td>
<td>-33</td>
<td>-40</td>
<td>-34</td>
<td>68</td>
<td>54</td>
<td>-38</td>
<td>40</td>
<td>-82</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>72</td>
<td>-8</td>
<td>-68</td>
<td>-10</td>
<td>-20</td>
<td>99</td>
<td>-78</td>
<td>-57</td>
<td>42</td>
<td>41</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>73</td>
<td>52</td>
<td>-17</td>
<td>-48</td>
<td>-84</td>
<td>-94</td>
<td>65</td>
<td>82</td>
<td>51</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>52</td>
<td>48</td>
<td>23</td>
<td>-54</td>
<td>-54</td>
<td>19</td>
<td>2</td>
<td>-64</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-50</td>
<td>-28</td>
<td>-24</td>
<td>0</td>
<td>64</td>
<td>-18</td>
<td>-93</td>
<td>28</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 11: **Factor Coefficients for Input Functions Results of GA**. The coefficients corresponds to only the 9 unique solutions and not the 14 solution involving different table size or correction points number.
Table 12: Power Coefficients for Input Functions Results of GA. The coefficients corresponds to only the 9 unique solutions and not the 14 solution involving different table size or correction points number.

<table>
<thead>
<tr>
<th>ID</th>
<th>b₁</th>
<th>b₂</th>
<th>b₃</th>
<th>b₄</th>
<th>b₅</th>
<th>b₆</th>
<th>b₇</th>
<th>b₈</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>20</td>
<td>17</td>
<td>27</td>
<td>14</td>
<td>19</td>
<td>17</td>
<td>17</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>27</td>
<td>14</td>
<td>12</td>
<td>20</td>
<td>2</td>
<td>24</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>11</td>
<td>19</td>
<td>5</td>
<td>7</td>
<td>16</td>
<td>14</td>
<td>24</td>
<td>10</td>
<td>21</td>
</tr>
<tr>
<td>28</td>
<td>15</td>
<td>25</td>
<td>8</td>
<td>24</td>
<td>9</td>
<td>20</td>
<td>6</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>30</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>55</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>30</td>
<td>28</td>
<td>2</td>
<td>29</td>
</tr>
<tr>
<td>57</td>
<td>2</td>
<td>16</td>
<td>8</td>
<td>15</td>
<td>26</td>
<td>29</td>
<td>12</td>
<td>22</td>
</tr>
<tr>
<td>59</td>
<td>18</td>
<td>17</td>
<td>17</td>
<td>26</td>
<td>3</td>
<td>29</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td>63</td>
<td>7</td>
<td>22</td>
<td>26</td>
<td>22</td>
<td>16</td>
<td>5</td>
<td>24</td>
<td>6</td>
</tr>
<tr>
<td>72</td>
<td>22</td>
<td>13</td>
<td>6</td>
<td>16</td>
<td>24</td>
<td>8</td>
<td>10</td>
<td>7</td>
</tr>
<tr>
<td>73</td>
<td>12</td>
<td>13</td>
<td>12</td>
<td>24</td>
<td>23</td>
<td>7</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>17</td>
<td>9</td>
<td>23</td>
<td>26</td>
<td>6</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>27</td>
<td>13</td>
<td>9</td>
<td>9</td>
<td>26</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>17</td>
<td>22</td>
<td>5</td>
<td>11</td>
<td>17</td>
<td>8</td>
<td>15</td>
</tr>
<tr>
<td>73</td>
<td>25</td>
<td>12</td>
<td>3</td>
<td>30</td>
<td>5</td>
<td>3</td>
<td>19</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>22</td>
<td>10</td>
<td>8</td>
<td>23</td>
<td>19</td>
<td>11</td>
<td>29</td>
</tr>
<tr>
<td>73</td>
<td>13</td>
<td>20</td>
<td>7</td>
<td>11</td>
<td>6</td>
<td>10</td>
<td>10</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>7</td>
<td>16</td>
<td>7</td>
<td>21</td>
<td>14</td>
<td>11</td>
<td>14</td>
</tr>
<tr>
<td>73</td>
<td>13</td>
<td>20</td>
<td>7</td>
<td>11</td>
<td>6</td>
<td>10</td>
<td>10</td>
<td>13</td>
</tr>
</tbody>
</table>
Figure 48: **Optimal Input Functions.** All 9 different input functions indicated by its original ID in Phase II. \( N = 4,096 \) points.

Figure 49: **Optimal BLEP Residuals.** All 9 different BLEP residuals indicated by its original ID in Phase II. \( N = 4,096 \) points.
Table size of 4,096

The smallest table size had 4 optimal solutions: GA 1, 2, 3, and 9, each with different correction points ranging from 2 to 4. \( m = 3 \) and \( m = 4 \) gave the best performance. Also, two different inputs (7 and 11) with the same \( m = 3 \), gave the exact same highest fundamental. The final inputs can be seen in Figure 50. Note that this table had a highest fundamental limited at \( f_c = 2,358.40 \) Hz in the optimal input which uses the most correction points \( m = 4 \).

Figure 50: **Optimal Input Functions for Table Size of 4,096.** GA output for Optimal Input Functions of Table N = 4,096 points.

Observe that the GA 2 input function, which corresponds to ID 7 with \( m = 3 \) has a small dent in its center. This means it does not behave like the other functions, but has a steeper slope around its center. This is not a problem because the integration eliminates that difference and the final BLEP residual
obtained is regular, like all the others. These can be observed in Figure 51.

![Figure 51: Optimal BLEP Residuals for Table Size of 4,096. GA output for Optimal BLEP Residuals of Table N = 4,096 points.](image)

**Table size of 8,192**

The size $N = 8,192$ only had one optimal solution, GA 10, which uses $m = 4$ correction points. It behaves as all the previous solutions and its function input and BLEP residual can be observed in Figure 52 and 53 respectively. The highest fundamental was found to be $f_c = 3,169.92$, an improvement of 126% over the Blackman window using this same parameters.

**Table size of 12,288**

This table had 3 optimal input functions: GA 4, 5, and 11, two of which are unique. As it happened with the $N = 4,096$ table, two solutions (GA 4 and
Figure 52: **Optimal Input Functions for Table Size of 8,192.** GA output for Optimal Input Functions of Table N = 8,192 points.

Figure 53: **Optimal BLEP Residuals for Table Size of 8,192.** GA output for Optimal BLEP Residuals of Table N = 8,192 points.
5) yielded the same result of a highest fundamental of $f_c = 3.518.55$. These two solutions marked as GA 4 and GA 5 had very similar windows, only varying in one decaying faster to 0 than the other. This can be observed in Figure 54. The BLEP residuals are also very similar and can be seen in Figure 55. The GA 11 solution is not as good as the other two, with a lowest fundamental and you can see a much wider window, letting more components pass through. This solution was considered optimal, because it uses function ID 28, which had other three optimal solutions, making it interesting for further evaluation.

Figure 54: **Optimal Input Functions for Table Size of 12,288.** GA output for Optimal Input Functions of Table $N = 12,288$ points.
Figure 55: **Optimal BLEP Residuals for Table Size of 12,288.** GA output for Optimal IBLEP Residuals of Table N = 12,288 points.

**Table size of 16,384**

This table had only one optimal solution, GA 6, but it had a very important growth in frequency. It went from $f_c = 2,217.77\, \text{Hz}$ to almost making the highest note on the MIDI keyboard at $f_c = 3,969.73\, \text{Hz}$. The input function and BLEP residuals are displayed in Figures 56 and 57 respectively. They do not differ considerably from the previously found results.

**Table size of 20,480**

The table with $N = 20,480$ points used in GA 12 had an important growth over the Blackman window implementation, but its frequency is the same as the obtained with the previously discussed GA 6 with the $N = 16,384$ table
Figure 56: **Optimal Input Functions for Table Size of 16,384.** GA output for Optimal Input Functions of Table N = 16,384 points.

Figure 57: **Optimal BLEP Residuals for Table Size of 16,384.** GA output for Optimal BLEP Residuals of Table N = 16,384 points.
and fundamental of $f_c = 3,969.73 H z$. The reason for including this result as well, is that it is also closest to the highest note on the MIDI keyboard, which means that the next optimal solutions will probably get to the desired fundamental to cover the entire piano range. The input solution for this table can be observed in Figure 58, with its respective BLEP residual in Figure 59.

Figure 58: Optimal Input Functions for Table Size of 20,480. GA output for Optimal Input Functions of Table N = 20,480 points.

Table size of 24,576

This table size became the first to obtained a fundamental frequency above the desired highest note on the MIDI keyboard. Its two optimal solutions, GA 7 and GA 14, had very interesting improvements. The GA 7 had a highest fundamental of $f_c = 4,045.90 H z$, almost reaching the 88th key. The GA 14 was
Figure 59: Optimal BLEP Residuals for Table Size of 20,480. GA output for Optimal BLEP Residuals of Table N = 20,480 points.

the first result to be higher than the MIDI keyboard range and it was the second highest fundamental obtained in the entire Phase II with $f_c = 4,204.10Hz$, making it just barely above the desired limit. Figures 60 and 61 show the input functions and the BLEP residual respectively. Notice that GA 14 has a narrower window than GA 7. This results start pointing to the conclusion that a narrower window reduces the aliasing more effectively.

Table size of 28,672

This table was the last one to obtain optimal results, having generated two inputs which both overcame the desired limit of the highest piano note. GA 8 and GA 13, which both had very good performance. GA 8 uses only $m = 3$
Figure 60: **Optimal Input Functions for Table Size of 24,576.** GA output for Optimal Input Functions of Table $N = 24,576$ points.

Figure 61: **Optimal BLEP Residuals for Table Size of 24,576.** GA output for Optimal BLEP Residuals of Table $N = 24,576$ points.
correction points and gets to a $f_c = 4,192.38$, making it the third best solution of Phase II. GA 13 uses one more correction point but delivers $f_c = 4,447.27\,Hz$ perceptually alias-free, making it the best solution of Phase II. Notice in Figure 62 that the GA 13 has a narrower windowing factor than GA 8, this starts confirming the previous conclusion from the $N = 24,576$ tables that a narrower window will effectively reduce more aliasing. The BLEP residual is shown in Figure 63, having also a steeper decay to 0 in the GA 13 solution.

![Windowed Input Functions Obtained With GA - Table N = 28672](image)

Figure 62: **Optimal Input Functions for Table Size of 28,672.** GA output for Optimal Input Functions of Table $N = 28,672$ points.

**Table size of 32,768 and larger**

Given the chosen sampling rate of 48 kHz, a larger table size is not optimal. This is because larger tables gave the same performance as a smaller
Figure 63: **Optimal BLEP Residuals for Table Size of 28,672.** GA output for Optimal BLEP Residuals of Table N = 28,672 points.

Table at the limit where the number of correction points is close to the number of samples per each period cycle. At these frequencies, the problem becomes dependent only on the number of correction points and the table size is not as important. Tables of 32,768 points were tried with no better results than the tables of size 28,672.

### 4.2.4 Trade off between Aliasing and Harmonic Decay

There was an interesting finding in this research and it was that a trade off between aliasing and harmonic decay was found in the number of zero crossings (n) of the original sinc function, while maintaining the same number of BLEP correction points (m) and the same table size (N). This translates to the
rippling of the BLEP residual which simulates the additive synthesis effects. The more rippling there is (number of zero crossings), the more the harmonic envelope is preserved in comparison to the original trivial one.

When you have the less possible number of zero crossings, one on each side, you have no rippling at all in the BLEP residual, but you do have a larger and smoother transition around the discontinuity. This is the best option to reduce the more aliasing, and thus achieve the highest perceptual alias-free fundamental frequency.

This finding applies to the Blackman window, as well as the GA obtained results. Given that the GA was based on the Bosi method, the optimal solutions it found corresponded to this last case, where the free variable of zero crossings always was found optimal at one on each side with all different number of correction points (from 2-5) and table sizes (4,096 to 28,672). These solutions of course, were the ones that gave the better performance in the Bosi analysis. Because of this, the trade off between aliasing and harmonic decay is inevitable, in both the GA results and the Blackman windowing technique.

So it is not correct to compare the results of a GA result optimized for alias-free with its Blackman correspondent version in the Harmonic Decay analysis because, the one with less aliasing will always have greater harmonic decay.

In order to make a good analysis, the solutions found in the GA will be optimized for the Harmonic Decay as well, and this was found to be done with a
very simple change on the number of zero crossings $n$ from the optimal 1 for Aliasing to an optimal 4 for harmonic decay. This solutions will be referred as Optimized for Aliasing and Optimized for Harmonic Decay respectively. So in conclusion, it would be correct to compare only the harmonic decay optimized versions in the Harmonic decay analysis, as well as only the aliasing optimized versions in the aliasing analysis.

The Harmonic optimized versions will have the rippling necessary as seen on Figure 64. It will also add the rippling to the final BLEP residual, which can be observed in Figure 65.

![Windowed Input Functions Obtained With GA - 4 Zero Crossings](image)

Figure 64: **Optimal Input Functions Optimized for Harmonic Decay**. Table of $N = 4,096$ points for all (independent of $m$ and $N$), with $n = 4$ zero crossing for augmented rippling.

The next two sections explain better how the trade off is between the
Figure 65: **Optimal BLEP Residuals Optimized for Harmonic Decay.** Table of $N = 4,096$ points for all (independent of $m$ and $N$), with $n = 4$ zero crossing for augmented rippling.
harmonic envelope and the aliasing, using the NMR and A-Weighted evaluation
techniques to see the effects over the aliasing and the RMSE, Maximum Absolute
Harmonic Error, and A-Weighted methods to see the trade off in terms of
harmonic decay.

**Trade off with Aliasing**

Figures 66 and 67 display the NMR values and A-Weighted Aliasing error
respectively, for 4 total scenarios. The scenarios are based on the Blackman
window implementation, but as discussed before, it yields the same conclusion for
the GA outputs. The 4 scenarios all use $m = 4$ correction points. The idea is to
see what happens in the GA solutions and in the Blackman case when the table
size or the number of zero crossing is modified. So for each of the two table sizes
of $N = 4,096$ and $N = 24,576$, two different NMR have been calculated: the first
one with no rippling at all ($n = 1$), and the second with some ($n = 4$). You can
see that for both cases, the NMR is higher (worst) in the case including the
rippling, compared to its corresponding version with no rippling. This is the
negative effect of adding rippling, a lower NMR, which translates to a lower
highest fundamental frequency alias-free and in general more aliasing. This trade
off has a positive effect on the harmonic envelope decay as seen in the next
section. The effect of the table size translate to a lower NMR (better) with a
larger table at high frequencies. Note that at low frequencies, the NMR is almost
the same no matter the table size, only the rippling added.

For the A-Weighted Aliasing Error, Figure 67 shows similar results to the
Figure 66: **NMR (Trade Off)**. Blackman case. Table sizes of $N = 4,096$ and $N = 24,576$, both with $m = 4$ correction BLEP points and each table with a different zero crossing number of $n = 1$ and $n = 4$. 
NMR. Larger number of zero crossings \( n \) have a larger error which translates to more aliasing. The effect of the size of the table are seen more clearly, where the larger table have slightly better results, considering the big jump of a table 6 times bigger.

Figure 67: \textbf{A-Weighted (Aliasing) Error (Trade Off)}. Blackman case. Table sizes of \( N = 4,096 \) and \( N = 24,576 \), both with \( m = 4 \) correction BLEP points and each table with a different zero crossing number of \( n = 1 \) and \( n = 4 \).

\textbf{Trade off with Harmonic Envelope}

As it was seen in the previous section, the adding of rippling in the BLEP residual, which translates to having a higher number of zero crossing in the original \( \text{sinc} \) function, causes a negative effect in aliasing, but it improves the harmonic decay. As seen in Figures 68, 69, and 70, the case when more rippling is present, both the GA solution and the Blackman cases, have a better
Looking first at the RMSE evaluation, it is clear that this method only depends on the number of zero crossings $n$. This is proved for only this range of table sizes, maybe some larger or smaller sizes will behave differently. A larger $n$ corresponds to a smaller error in the harmonic envelope, opposite of what occurs with aliasing. It is seen that the error decays around $15dB$ from $n = 1$ to $n = 4$. Observe the results in Figure 68.

The maximum absolute harmonic error can be observed in Figure 69. It is clear that for this range of sizes, the table size is not important as in the RMSE evaluation. A more drastic reduction of harmonic envelope decay is observed in
the case of \( n = 4 \), where the error is almost reduced to 0.1\( dB \) from a 1.8\( dB \) with \( n = 1 \).

Finally, the A-Weighted Harmonic Error gives similar results to the RMSE and Maximum Absolute Harmonic Error evaluations. Figure 70 shows the same results, but it adds some information, where in the case of \( n = 1 \) zero crossings, the table size that matters, and a 10\( dB \) difference is observed between the two sizes, where the smaller table has a smaller overall error.

**General Effects of Size of Table**

From the previous analysis, it can also be extracted that the table size is not very important at low frequencies, but it is at high frequencies, especially
Figure 70: **A-Weighted (Harmonics) Error (Trade Off)**. Blackman case. Table sizes of $N = 4,096$ and $N = 24,576$, both with $m = 4$ correction BLEP points and each table with a different zero crossing number of $n = 1$ and $n = 4$. 
above the highest fundamental perceptually alias-free. A discussion about this finding will be done in the next Chapter.

4.2.5 **BLEP Table for Perceptually Alias-Free MIDI Keyboard (Piano)**

The highest note on the MIDI keyboard for a piano is the MIDI note 108, which corresponds to an approximated fundamental frequency of

\[ f_c = 4,186.0 \text{Hz} \]

This note is the last key (88th) on the conventional piano and it represents the note C in the 8th Octave. There are two ways to solve this problem: one is optimizing the size of the table, for memory purposes; the other is to optimize the number of correction points for computational purposes. The latter one being preferred.

**Optimized for Number of Correction Points**

The best table to implement a MIDI keyboard based on the fewest number of points is the Solution GA 8. It covers the entire piano frequency range alias-free and it only corrects 3 points on each side. The table size is a bit large, 28,672, which exactly 28 kB (kilo-Bytes), but this is not an issue for current memory performance. The processing is minimum, having to only look up the table value 6 times per period cycle. This is considered to be a better solution because of its efficiency in computing costs, which can be crucial in real-time synthesis where memory is not a major issue.

The second best solution in terms of computing performance, and overall, is GA 13, where only 2 more points are added in total, bringing the total to 8 points per cycle period. These two solutions will be the ones tested in the
Listening test developed in the next section.

**Optimized for Size of Table**

The other optimization technique was in terms of memory. The smaller table to be perceptually-alias free over the entire piano notes is of 24,576 points, which corresponds to 24 kB. This corresponds to GA 14. The issue with this optimization is that it has to correct 10 points per cycle, which is 4 more than the best current option from the previous section, which translates to higher computational costs, specially at higher frequencies, close to 8th Octave C note. This is also an issue because at this frequency, only a bit more than 10 points are sampled per cycle, which means it would be correcting almost all of the points in the waveform. This solution is considered not as efficient as GA 8 or GA 13 and will not be included in the listening test.

### 4.2.6 Further Analysis of Results

The next step after having all the results from Phase II of the GA is to analyze them with several evaluation methods that measure aliasing, harmonic decay or both. In order to have this analysis, a procedure was followed because it was found to be efficient in calculations and order.

The general idea is to analyze only the best results from Phase II by calculating and plotting all the evaluation methods, while doing a frequency sweep over the entire MIDI table. Comparison will be made between all the GA results, the trivial, and the original windowed BLEP method.

The method to do the evaluation of the computational results is described
below:

0. Follow the same steps for calculating the input for the Bosi method, which where the harmonic peaks and aliasing peaks were separated in the FFT (See Section Fitness Function based on Bosi).

1. Do a analysis of the trivial case parallel to the BLEP modified case. This is necessary to obtain the results of the methods for analysis. This trivial analysis is also the same as the calculation of the input for the Bosi method (See Section Fitness Function based on Bosi).

2. Based on the harmonic peaks of the BLEP waveform, do additive synthesis to find the additive synthesis version of the waveform. This will work for the PEAQ and NMR methods.

3. Based on the harmonic peaks of the Trivial waveform, do additive synthesis to find the additive synthesis textbook version of the waveform. This will work for the PEAQ-TR and NMR-TR methods.

4. Find the NMR and PEAQ values based on the BLEP and the BLEP additive version waveforms.

5. Find the NMR-TR and PEAQ-TR values based on the BLEP and the BLEP additive textbook (trivial) waveforms.

6. Calculate the RMSE and the maximum absolute harmonic error based on the harmonic peaks converted to dB and normalized to 0 dB and below.

7. Calculate the A-Weighted errors for only aliasing and only harmonic decay based on the linear harmonic peaks. Afterwards, convert to dB and
normalize to 0 dB and below.

8. Repeat steps 0 to 7 for all input functions found to be optimal with the GA.

9. Repeat steps 0 to 7 for all Blackman implementations corresponding to each input function of the GA in a matter of table size $N$ and number of correction points $m$.

Given the trade off found between the aliasing elimination and the harmonic decay, and because the effects apply to the Blackman window, as well as the GA optimal solutions found, further analysis of the results will be done with the GA obtained inputs and Blackman versions optimized for each case: only aliasing and only harmonic decay.

**Perceptually Alias-Free Solutions Analysis**

For the analysis based on just the aliasing, three different measurements can be calculated: NMR, PEAQ and A-Weighted Aliasing Error. The first two are comparing the aliasing of the BLEP modified waveform to its Ideally Bandlimited waveform, implemented through additive synthesis. This last one keeps the same amplitude and phase of the harmonic peaks of the BLEP waveforms and reconstructs an identical version, just without any aliasing. The last one, calculates an A-Weighted total sum of the aliasing over the the audible frequency range. These measurements are only comparing aliasing, but not the harmonic decay the BLEP method naturally introduces.

Remember from the Trade Off section, that the aliasing is improved in
larger table sizes, specially at higher frequencies. This was confirmed in all three measurements for aliasing. So the Best Case Scenario for the comparison to the Blackman implemented BLEP will be the larger table size of $N = 28,672$. The worst case scenario will be the $N = 4,096$ table. The best case will also have the less correction points $m = 2$, which has proved to give better aliasing improvement, and the worst case will have the large number of correction points $m = 5$, which has less improvement in aliasing noise. These values were chosen because the original method has the same $n$ and $m$ values, when $n$ is smaller, the better the aliasing improvement is.

NMR

The NMR is a basic measurement of the noise compared to the masking. The signal is compared with a completely alias-free version. The masking is implemented by an asymmetrical spreading function. Figure 71 shows an overall comparison of the NMR of all the 14 solutions found for the GA. Each solution has a different number of BLEP points and size of the table, therefore they are going to have very different NMR values. Logically, the best NMR would be given by the larger $N$, and this holds true. In this case, the best scenario is GA 14, because it is the one that holds steady longer in a low NMR, until the highest fundamental alias-free.

The idea is to compare the results of the GA with the current implementation of BLEP. For this, the Blackman windowed implementation is compared with a particular GA solution. Keep in mind, that the table size and
the number of BLEP points have to be the same for both cases. With this thought, a comparison between each GA solution and its corresponding Blackman implementation was done. Figure 72 shows the solution for the particular case of a table size of 28,576 points and 3 BLEP correction points (GA 8). For this case, the NMR value of the GA solution is below the Blackman one, which is what we are looking for; a lower aliasing.

Note also that the NMR stays constant for low frequencies, it then starts growing rapidly after a certain frequency, which corresponds to the highest fundamental that is perceptually alias-free. For the GA case, this frequency is higher than for the Blackman case which is the same results found with the Bosi
analysis. After certain point, at a high frequency, the Blackman and the GA solutions have varying NMR, where sometimes one of the solutions is better than the other. This is not a worrying situation, because sounds are synthesized up to MIDI 108, and the GA responds well even up to $f_c = 4,192.383\,Hz$, which is just enough.

The general conclusion is that at the MIDI keyboard range, the GA performs better than the Blackman window, and after a certain high frequency, which is above the highest frequency on the keyboard, there is no clear statement on which solution performs better. This analysis was made for all 14 solutions and the same results were found, in each of the cases, the GA would perform better than the Blackman case over the keyboard frequency range. This confirmed the results found based on the Bosi method.

**PEAQ**

The PEAQ standard has been omitted from previous researches because its results do not coincide with the NMR or Bosi values [Välimäki et al., 2012]. However, this research implemented PEAQ to corroborate these findings. Figure 73 shows a comparison of all the 14 GA outputs, the trivial and the Blackman worst and best cases, same as with NMR. The noticeable disagreement with NMR is that the worst case of Blackman and the Trivial version seem to perform better than all other measurements from the first MIDI note $27.5\,Hz$ up to about $70\,Hz$. This is the opposite of what it was expected to be found. Even without considering the GA outputs being tested, the Blackman best case is known to
Figure 72: NMR Evaluation (MIDI Keyboard Case). Comparison of NMR value for GA 8 (light blue circle) and its corresponding Blackman case (black x). GA 8 was the optimal implementation found to construct an perceptually alias-free MIDI keyboard.
have better performance than the trivial at all frequencies. After this range of no clear results, the PEAQ value behaves inside the expectations. The best Blackman case performs much better than the trivial and the worst case. The GA outputs also behaves accordingly to the NMR values. More of this discussion about the results from PEAQ can be found in the next chapter.

Figure 73: **PEAQ Evaluation.** Comparison of PEAQ value for Trivial (red +), the worst case of Blackman $N = 4,096$ and $m = 5$ (black x), the best case of Blackman $N = 28,673$ and $m = 2$ (blue x) and all 14 GA outputs (color circles).

When comparing the GA 8 output and its corresponding Blackman window, the results become clearer. The ODJ of PEAQ for GA 8 is always higher than for Blackman, which was the same results found on the NMR analysis. The other effect observed in PEAQ as well as NMR, was the interchangeable performance after the breakpoint of the highest perceptually
alias-free fundamental. After this point, the behavior of both the Blackman and the GA is unexpected, where one is better than the other in a random manner. This effect can be observed in Figure 74. When the trivial reference is not present, the analysis of this results become easier to interpret. The only range which is definitely out of context is the one in the lower frequencies, where a smaller PEAQ value is registered. Previous knowledge of how the BLEP method works indicates that at the lower fundamentals, it is easier to hide or eliminate the aliasing components from the human hearing system. After this range the behavior agrees with the results from NMR and A-Weighted Aliasing Error (refer to the following section). All other comparisons of the GA and their corresponding Blackman version showed the same results as in this particular case, this is the reason why only one of them is shown.

**A-Weighted Aliasing Error**

The A-Weighted Aliasing Error was one of the evaluations introduced on this research as an alternative measurement, which was design to be very cheap computationally in comparison to the other perceptual models. Figure 75 shows the accumulated aliasing for each case respect to the trivial reference. The lower the error is, the more improvement there was in reducing the aliasing. This values agree with the NMR measurement in all senses, but the one where the Blackman best case performs worst in the lower frequencies than the worst case, but better at the higher ones. All the GA outputs show a lower aliasing noise than both Blackman cases, this is due to having an only aliasing optimization technique.
Figure 74: **PEAQ Evaluation (MIDI Keyboard Case)**. Comparison of PEAQ value for GA 8 (light blue circle) and its corresponding Blackman case (black x). GA 8 was the optimal implementation found to construct an perceptually alias-free MIDI keyboard.
during Phase II. Same behavior is seen at the higher fundamentals, where there is no clear conclusion on whether the Blackman are better than the GA solutions.

Figure 75: **A-Weighted (Aliasing Only) Evaluation.** Comparison of A-Weighted Aliasing Error for Trivial (red +), the worst case of Blackman $N = 4,096$ and $m = 5$ (black x), the best case of Blackman $N = 28,673$ and $m = 2$ (blue x) and all 14 GA outputs (color circles).

Figure 76 shows the comparison of only GA 8 with its corresponding Blackman version. As seen in Figure 75, the GA has a much lower error than the Blackman case over the entire piano frequency range. At really high fundamentals, no clear outcome is seen on which performs better. All other 13 GA outputs obtained very similar results, thus giving the same conclusion.

**Harmonic Decay Optimization Analysis**

The main factor that affects all of the harmonic decay evaluation options is the number of zero crossing in the $sinc$ function $n$. The higher this number is,
Figure 76: **A-Weighted (Aliasing Only) Evaluation (MIDI Keyboard Case).** Comparison of A-Weighted Aliasing Error for GA 8 (light blue circle) and its corresponding Blackman case (black x). GA 8 was the optimal implementation found to construct an perceptually alias-free MIDI keyboard.
the lower the errors are. This section includes the RMSE, Maximum Absolute Harmonic Error and the A-Weighted value for only Harmonic Error. Remember that all the GA outputs being analyzed on this section where optimized for rippling, modifying the $n$ from 1 to 4.

As seen on the Trade Off section, sometimes a worst case for the Harmonic Decay is seen in a larger table, specially where evaluating the maximum absolute harmonic error. The results presented on this section are compared to the worst case scenario of the Blackman input as the larger table, and the best one as the smaller table. This is also affected by the number of correction points. A Blackman with more correction points has a better harmonic envelope because it also has more zero crossings of the $sinc$. This is why the worst case scenario not only has the larger table $N = 28,672$ but also the less correction points $m = 2$. While the best case has the more correction points $m = 5$ and the smaller table size $N = 4,096$. Remember that this thought is based on the original Blackman case, where the $n$ is the same $m$.

**RMSE**

The first measurement of Harmonic behavior is the RMSE. In Figure 77, all of the GA outputs have a smaller RMSE value than the Blackman versions. Of course, only some are better than the best Blackman implementation. The best Blackman case for each GA solution are different. For this reason a comparison of each output has to be done with its corresponding Blackman case. In conclusion, the RMSE shows an improvement for the GA over the Blackman
versions, as it was found in the aliasing only evaluation.

Figure 77: **RMSE Evaluation.** Comparison of RMSE value for the best case of Blackman $N = 4,096$ and $m = 5$ (blue x), the worst case of Blackman $N = 28,673$ and $m = 2$ (black x) and all 14 GA outputs (color circles).

Figure 78 shows the comparison of the best GA case for the MIDI keyboard implementation, GA 8. It is shown only against its Blackman version. Over all the frequency range, GA shows a better performance. Same procedure was done for each of the GA outputs; all of them demonstrated improvement against the original Blackman implementation.

**Maximum Absolute Harmonic Error**

The second measurement for Harmonic Decay is the maximum absolute harmonic error. Figure 79 shows the results for all the GA outputs and the Blackman worst, and best versions. The overall view is that the GA also
Figure 78: RMSE Evaluation (MIDI Keyboard Case). Comparison of RMSE value for GA 8 (light blue circle) and its corresponding Blackman case (black x). GA 8 was the optimal implementation found to construct a perceptually alias-free MIDI keyboard.
performs better in this evaluation.

![Graph showing Maximum Absolute Harmonic Error Analysis]

Figure 79: **Maximum Absolute Harmonic Error Evaluation.** Comparison of Maximum Absolute Harmonic Error for the best case of Blackman $N = 4,096$ and $m = 5$ (blue x), the worst case of Blackman $N = 28,673$ and $m = 2$ (black x) and all 14 GA outputs (color circles).

In order to correctly analyze each of the GA outputs, a comparison with only its corresponding version was made. As seen in Figure 80, the GA 8 solution performed well below its corresponding Blackman version, more than $3dB$ below. All 14 GA outputs obtained the same results, which agrees with the previously found results from the RSME evaluation.

**A-Weighted Harmonic Error**

The last measurement for the Harmonic Decay was the A-Weighted method, which was used for the first time in this research. Figure 81 shows the same behavior as RMSE. All of the GA solutions behave well below the
Figure 80: **Maximum Absolute Harmonic Error Evaluation (MIDI Keyboard Case)**. Comparison of Maximum Absolute Harmonic Error for GA 8 (light blue circle) and its corresponding Blackman case (black x). GA 8 was the optimal implementation found to construct an perceptually alias-free MIDI keyboard.
As it is correct doing, each GA output was compared to its corresponding Blackman option. Only the GA 8 case is shown in Figure 82. The error is considerably lower than the one found with Blackman. All other 13 cases behaved in the same way, confirming the greater performance of the GA against the original Blackman version in the Harmonic Decay error.

**Combined Aliasing and Harmonic Envelope Analysis**

The other two measurements being tested in this research are based on comparing the NMR and PEAQ values to the textbook trivial additive synthesis version without any harmonic decay, instead of using the BLEP additive version.

---

**Figure 81: A-Weighted (Only Harmonics) Evaluation.** Comparison of A-Weighted Harmonic Error for the best case of Blackman $N = 4,096$ and $m = 5$ (blue x), the worst case of Blackman $N = 28,673$ and $m = 2$ (black x) and all 14 GA outputs (color circles).
<table>
<thead>
<tr>
<th>Frequency (Hz)</th>
<th>A-Weighted Error (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>100</td>
<td>0.1</td>
</tr>
<tr>
<td>1000</td>
<td>0.2</td>
</tr>
<tr>
<td>10000</td>
<td>0.3</td>
</tr>
</tbody>
</table>

Figure 82: **A-Weighted (Only Harmonics) Evaluation (MIDI Keyboard Case)**. Comparison of A-Weighted Harmonic Error for GA 8 (light blue circle) and its corresponding Blackman case (black x). GA 8 was the optimal implementation found to construct an perceptually alias-free MIDI keyboard.
that does have the BLEP introduced harmonic envelope error. This would include the effect of both harmonic decay and aliasing together.

**NMR-TR (Optimized for Aliasing)**

Figure 83 shows the NMR-TR for the optimized aliasing version. The same problem occurs as in PEAQ where in the lower frequencies, the trivial case seems to perform better, even though it is know this cannot be possible. The Blackman references are not very clear in this case. For the optimized aliasing version the same references were chosen as the ones for the Aliasing Analysis section, but because of the trade off, having a worst aliasing correction will have a smaller harmonic decay error. This is why in Figure 83, the Blackman version have a better performance than the GA outputs. This evaluation method shows that it has a priority of harmonic decay over aliasing correction, having almost the same ranking as any of the Harmonic Decay evaluation methods (RMSE, Maximum Absolute Harmonic Error or A-Weighted Harmonic Error). Given the issues with this measurement, it will not be brought into further investigation.

**NMR-TR (Optimized for Harmonic Decay)**

Figure 84 shows the comparison for the harmonic optimized decay versions. The same Blackman references from the previous Harmonic Decay Only analysis were chosen. In this case, as in the previous section, these references are not very clear because the measurement mixes both aliasing and harmonic decay. The good finding in this case is that the error at low frequencies where the trivial apparently performed better is eliminated and all the GA outputs behave in a
Figure 83: NMR (Trivial Reference) Evaluation (Optimized for Aliasing). Comparison of NMR-TR (Optimized for Aliasing) value for Trivial (red +), the worst case of Blackman $N = 4,096$ and $m = 4$ (black x), the best case of Blackman $N = 28,673$ and $m = 3$ (blue x) and all 14 GA outputs (color circles).
similar manner; same occurs with the Blackman references. Here, the trade off
between aliasing are harmonic decay becomes clearer. There is a mixture of
performance between all GA outputs that is not directly related to those of only
aliasing or only harmonics. This seems to be a good indicator of a middle point
for finding an optimal solution for both issues. Further analysis of this
measurement is done when evaluating results from Phase III.

Figure 84: NMR (Trivial Reference) Evaluation (Optimized for Harmonic
Decay). Comparison of NMR-TR (Optimized for Harmonic Decay) value for Trivial
(red +), the best case of Blackman $N = 4,096$ and $m = 4$ (black x), the worst case of
Blackman $N = 28,673$ and $m = 3$ (blue x) and all 14 GA outputs (color circles).

PEAQ-TR (Optimized for Aliasing)

The other measurement proposed in this research is the PEAQ evaluation
based on the trivial amplitudes, in order to include harmonic decay as well. This
PEAQ-TR was calculated only over the optimized versions of the GA and
Blackman for aliasing. Figure 85 shows the graphical output of this method. One of the positive outcomes of this method is that it does not include the error seen in the regular PEAQ or the NMR-TR (Optimized for Harmonic Decay). The trivial seems to have a worst performance over the entire frequency sweep as it should definitely have. The references of the Blackman version are not very clear either, considering both effect come into play here. But an overall performance seems to be a good indicator to define this method as a measurement of an efficient input for both harmonic decay and aliasing. This will be included and expanded in the Phase II evaluation, along with NMR-TR (Optimized for Harmonic Decay) where both parameters are optimized. Note that in this method, the two Blackman references seem to have almost the same exact performance. This is another reason for arguing why the Blackman references are not very clear for these cases of double parameter measurement.

PEAQ-TR (Optimized for Harmonic Decay)

This measurement has the same issue as regular PEAQ and NMR-TR for optimized aliasing. The trivial case seems to perform better at very low frequencies. This is something not wanted in an evaluation, given that it is proven to be false. After that low frequency range, the behavior is more or less normal, but the previous measure of PEAQ-TR (For aliasing optimization) gave clearer results. Note that as in the previous section, the Blackman references have almost the exact same performance. In conclusion, this measurement as well as NMR-TR (Optimized for Aliasing) are not considered to be meaningful and
Figure 85: **PEAQ (Trivial Reference) Evaluation (Optimized for Aliasing).** Comparison of PEAQ-TR (Optimized for Aliasing) value for Trivial (red +), the worst case of Blackman $N = 4,096$ and $m = 4$ (black x), the best case of Blackman $N = 28,673$ and $m = 3$ (blue x) and all 14 GA outputs (color circles).
will be dropped from further investigation. Figure 86 shows these results.

![PEAQ-TR Analysis](image)

Figure 86: **PEAQ (Trivial Reference) Evaluation (Optimized for Harmonic Decay)**. Comparison of PEAQ-TR (Optimized for Harmonic Decay) value for Trivial (red +), the best case of Blackman $N = 4,096$ and $m = 4$ (black x), the worst case of Blackman $N = 28,673$ and $m = 3$ (blue x) and all 14 GA outputs (color circles).

### 4.2.7 Comparison of Input Functions with Equal Bosi Scores

There were two different sets of input solutions that had the same Bosi score, one was GA 2 and 3 with a $N = 4,096$ table and $m = 3$ BLEP correction points. The other pair was GA 4 and 5 with $N = 12,288$ table size and $m = 3$. The idea is to compare these two pairs of solutions that seem to be equal using the Bosi method, and find out which two are actually better using other evaluation algorithms. The NMR and Maximum Absolute Harmonic Error have proven to yield good results, so these will be the ones used for both aliasing and harmonic decay comparison.
Figure 87 shows the NMR performance of all the four GA outputs. Looking at GA 2 and 3 together, the NMR overall performance for GA 2 is better, lower than for GA 3. This effect is not very clear at low frequencies, but at high ones it is very noticeable. Comparing GA 4 and 5, there is almost no difference in the NMR value, it is almost equal. From this evaluation, no conclusion is obtained for GA 4 and 5. For GA 2 and 3, a lower NMR is definitely better. Given these solutions were optimized for aliasing, GA 2 is chosen over GA 3.

On the other hand, Figure 88 shows the Maximum Absolute Harmonic Error of the 4 outputs. GA 4 and 5 did not have any variation in NMR, so the harmonic decay will be the defining factor to chose one over the other. The lower
harmonic error is given by GA 5, so this is the one chosen from the second pair. The first pair is more complicated. The NMR value favored GA 2, but because of the trade off problem, GA 3 has a lower harmonic error. As discussed before, given these solutions were aliasing optimized, the lower NMR has a bigger weight over the decision so GA 2 is the one chosen.

![Maximum Absolute Harmonic Error Analysis](image)

**Figure 88: Maximum Absolute Harmonic Error Evaluation (Equal Bosi Output).** Comparison of Maximum Absolute Harmonic Error value for Trivial (red +), Blackman (black x), and GA outputs.

### 4.3 Listening Test

The listening test will evaluate the performance of only two of the best GA results available, selected in terms of the previous analysis (NMR, RMSE, etc.). The test will also include a Blackman windowed BLEP version corresponding to the specifications of each GA result. The results chosen are GA
8, which has a $N = 28,672$ table and changes $m = 3$ points only, with its correspondent Blackman version (referred as BLK 8). This result was chosen because it was the most efficient way to implement the MIDI table for a piano.

The second result to test will be GA 13, which was the highest frequency obtained with a table of $N = 28,672$ changing $m = 4$ BLEP points on each side, its corresponding Blackman version is BLK 13.

Each of the four methods will be compared to their respective Ideally Bandlimited signal obtained using additive synthesis. This purpose of this test is to corroborate the findings in the aliasing only analysis from the previous section.

The chosen type of test is the ABC/HR, which corresponds to recommendation ITU-R BS.1116-2 [International Communications Union ITU, 2014]. This test in designed for only small impairments which is the case here. Theoretically, in the frequencies below their highest perceptually alias-free fundamental, there should not be an impairment perceived by the subject. At higher frequencies, there should be aliasing perceived, but not as noticeable as in the trivial case, which makes this a test to only detect very small changes. Remember that this test has a continuous grade from 5.0 being Imperceptible to 1.0 being Very Annoying. This can be seen in Table 3. This is the same type of test was implemented for comparing the BLEP method to BLIT and DPW. Because of this, this type of test will not be discussed further in this section. More information on this type of test can be obtained in the Listening Test section in Chapter 2.
4.3.1 Test Conditions and subjects

A total of 20 subjects was used to perform the test, which is the minimum required for this type of test. The conditions of the test must be constant with all subjects, that includes the environment and the equipment. For this test, the Weeks Studio at the University of Miami was used. The same set up location inside the studio was used for all the subjects.

The idea is to evaluate a specific parameter of audio quality; aliasing. Because of this, the subjects must have a background in music performance and/or music education. All 20 subjects were students of the Frost School of Music, with different audio backgrounds and number of years of experience. Some of the students were from the undergraduate programs and some from the graduate school.

The equipment used was a PC computer running Windows 8.1 with a Realtek HD audio card connected to a Pioneer HDJ-2000 headset. The choosing of the headphones was based on their performance over the frequency range of the samples being test, it has a nearly flat response over the entire range. The implementation of the test was by using an implementation with an open-license for public use [Vasquez Giner, 2013]. Same set up was used for the Listening Test section in Chapter 2, please refer there for further information.

This test consists of two questions on each scenario. The questions are randomly presented to the listener. The first question asks how perceptible is the difference between the Reference and the hidden reference, this will be referred as
Question 1. The theoretical correct answer would be that no difference was perceived, so a grade of 5.0 would be adequate.

The second question asks how perceptible is the difference between the Reference and the Object being tested, this will be referred as Question 2. The answer to this question is completely unique for each subject, the grade could be any value from 5.0 to 1.0. This is because the subjective quantification of the aliasing heard can vary from person to person.

4.3.2 Samples to Evaluate and Scenarios

The samples to evaluate were chosen to cover a range from where aliasing was not present to where it was proved to present, based on the results from the perceptual computed models. Frequencies near the highest fundamental perceptually alias-free were chosen to see the change around those points. In total 5 MIDI notes were evaluated, spread around from 2 kHz to 6 kHz. Only 5 frequencies were tested because there is a maximum of 20 scenarios before having negative effects of listening fatigue in the test results. The notes corresponded to the ones in Table 13. No higher frequencies were chosen because they are considered to be annoying for most people, and this would definitely be a bias for the test.

The samples were obtained at a 48 kHz sampling rate in .WAV raw files, with a vector length of 131,072. This corresponds to the samples being 2.73 seconds long. A total of 20 scenarios is reached when evaluating each sample of the 5 frequencies and the 4 methods to its corresponding additive synthesis
### Table 13: Samples Used in Listening Test

<table>
<thead>
<tr>
<th>MIDI</th>
<th>Frequency (Hz)</th>
<th>Aliasing Present (Bosi)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>2,637.02</td>
</tr>
<tr>
<td>2</td>
<td>105</td>
<td>3,520.00</td>
</tr>
<tr>
<td>3</td>
<td>109</td>
<td>4,434.92</td>
</tr>
<tr>
<td>4</td>
<td>113</td>
<td>5,587.65</td>
</tr>
<tr>
<td>5</td>
<td>115</td>
<td>6,271.92</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BLK 8</th>
<th>GA 8</th>
<th>BLK 13</th>
<th>GA 13</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

Table 13: **Samples Used in Listening Test.** Showing whether alias is present (X) or not in the Bosi method output at each particular frequency.

version. The total order of the scenarios can be observed in Table 14.

### 4.3.3 Normalization of the Loudness of the Samples

All the samples had to be normalized in loudness, so the results would not be affected. Given each pair of waveforms being compared are the same frequency and waveform, a timbre or pitch normalization is not required. The samples obtained from the BLEP methods all had a different loudness depending on the implementation the method. Both the Blackman version and the GA solutions, introduced a small error in the harmonic amplitudes.

In order to solve this problem, the best way found to normalized the .WAV files was to use the most common recommendation ITU-R BS.1770-3 to calculate perceptual loudness [International Communications Union ITU, 2011]. The method to match the loudness was doing an exhaustive search for a factor that scaled on of the samples to match the exact perceptual loudness of the other one with a certainty of ±0.02. The results from using this standard were very good. The samples could not be differentiated by any other aspect other than aliasing.
<table>
<thead>
<tr>
<th>Scenario</th>
<th>Test Sample</th>
<th>Frequency (Hz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>BLK 8</td>
<td>2,637.02</td>
</tr>
<tr>
<td>2</td>
<td>BLK 8</td>
<td>3,520.00</td>
</tr>
<tr>
<td>3</td>
<td>BLK 8</td>
<td>4,434.92</td>
</tr>
<tr>
<td>4</td>
<td>BLK 8</td>
<td>5,587.65</td>
</tr>
<tr>
<td>5</td>
<td>BLK 8</td>
<td>6,271.92</td>
</tr>
<tr>
<td>6</td>
<td>GA 8</td>
<td>2,637.02</td>
</tr>
<tr>
<td>7</td>
<td>GA 8</td>
<td>3,520.00</td>
</tr>
<tr>
<td>8</td>
<td>GA 8</td>
<td>4,434.92</td>
</tr>
<tr>
<td>9</td>
<td>GA 8</td>
<td>5,587.65</td>
</tr>
<tr>
<td>10</td>
<td>GA 8</td>
<td>6,271.92</td>
</tr>
<tr>
<td>11</td>
<td>BLK 13</td>
<td>2,637.02</td>
</tr>
<tr>
<td>12</td>
<td>BLK 13</td>
<td>3,520.00</td>
</tr>
<tr>
<td>13</td>
<td>BLK 13</td>
<td>4,434.92</td>
</tr>
<tr>
<td>14</td>
<td>BLK 13</td>
<td>5,587.65</td>
</tr>
<tr>
<td>15</td>
<td>BLK 13</td>
<td>6,271.92</td>
</tr>
<tr>
<td>16</td>
<td>GA 13</td>
<td>2,637.02</td>
</tr>
<tr>
<td>17</td>
<td>GA 13</td>
<td>3,520.00</td>
</tr>
<tr>
<td>18</td>
<td>GA 13</td>
<td>4,434.92</td>
</tr>
<tr>
<td>19</td>
<td>GA 13</td>
<td>5,587.65</td>
</tr>
<tr>
<td>20</td>
<td>GA 13</td>
<td>6,271.92</td>
</tr>
</tbody>
</table>

Table 14: **List of Scenarios in Listening Test.** A total of 20 scenarios. Five frequencies for each method tested. Four methods tested.
4.3.4 Results

There are several ways to analyze the results of this type of listening test (ABC/HR). The easier and less meaningful is the Absolute Grades for each question. This provides a general idea of how the test went. The second is the Difference Grades between questions 1 and 2, which provide a clearer behavior of the data. And the further analysis is done with the results from the Difference Grades but seeing the behavior between special cases in order to define a ranking or overall evaluation of the data. This analysis was performed as suggested in the standard [International Communications Union ITU, 2014].

Absolute Grades

The first step for analyzing the results of the listening test statistically was to do ANOVA to obtain the absolute grades. That is the hidden reference results apart from the test object results.

The idea is to do an evaluation in order to see if the results from the hidden reference case (Question 1) do not have a significant difference, which means they have correctly identified the reference in most cases. Figure 89 shows the results for the ANOVA run on the results from the listening test. Out of the 20 scenarios, 11 have no significant difference; therefore it could be likely that the hidden reference was correctly identified. This is a good number, considering the nature of small impairments being tested. The mean of each of those 12 scenarios is close to 5. The other 4 scenarios could not have been correctly identified as the reference. Further analysis is required to verify these assumptions.
Figure 89: **Hidden Reference Absolute Statistics.** The plot contains a box for each scenario. The mean is the central mark (red -). The edges of the box are the 25th and 75th percentiles (blue). The extreme data points are the whiskers (black -). The outliers (red +).
Figure 90 shows the comparison between all absolute grades of Question 1. There is a clear partition in two groups, seen by the gray dotted line. This observation agrees with Figure 89. There are 9 scenarios that seem to not have been easily correctly identifies, and 11 that seemed to have been. Looking at Table 14 and Table 13, the 11 samples that apparently were easily identified all had aliasing present. This is a first good indicator that the test is going to have meaningful results.

![Comparison Mean Absolute Grades from Q1](image)

Figure 90: **Hidden Reference Absolute Comparison.** The reference point is Scenario 1 (blue), all gray scenarios are not significantly different. All red scenarios are significantly different.

The ANOVA was also run separately on the objects tested (Question 2). See Figure 91 for reference. The behavior of the statistics is not as clear as the ones for Question 1. But they do seem to behave in a similar manner. Scenarios
2, 3, 4, 5, 9, 10, 13, 14, 15, and 20 seem to have a much lower mean than all other scenarios. Referring to Tables 13 and 14, all these samples have aliasing present. The only missing sample was sample 12, which did have aliasing present but has a higher grade than all the others.

![Absolute Grades from Q2](image.png)

Figure 91: **Test Object Absolute Statistics.** The plot contains a box for each scenario. The mean is the central mark (red -). The edges of the box are the 25th and 75th percentiles (blue). The extreme data points are the whiskers (black -). The outliers (red +).

It is clearer to see a comparison for all scenarios of only the Question 2 results. Figure 92 shows an easier way to identify what is happening than when having all the other statistics included, like in Figure 91. There are about 7 different groups. The first group on the right, includes Scenarios 1, 6, 11 and 16. None of these scenarios have aliasing present and they have the higher grade, which is a good indicator the results from the listening test are behaving like the
Bosi method indicated.

![Comparison Mean Absolute Grades from Q2](image)

Figure 92: **Test Object Absolute Comparison.** The reference point is Scenario 1 (blue), all gray scenarios are not significantly different. All red scenarios are significantly different.

All the results on this section were evaluated to be significant with a p value below 0.05. Even though this results are going in the same direction as the perceptual models from the previous section, the absolute grades are not a trustworthy basis for detailed analysis. Further analysis will be done with the difference grades.

**Difference Grades**

To improve the basis of a further analysis, the difference grades (Q1-Q2) were obtained using ANOVA. These are the results of the difference between the value of the hidden reference and the object questions. In an ideal case, if the
impairment was imperceptible, a difference grade of 0.0 (5.0-5.0) is obtained and if it was extremely perceptible, the difference grade would be 4.0 (5.0-1.0).

Figure 93 shows the statistics of all 20 scenarios for the difference grades. There are 10 scenarios with an obvious lower mean than the rest. This indicates, that in those scenarios, the subjects (in average) did not perceive aliasing. In order to have a correct comparison of the performance of all scenarios separately a further analysis was done.

**Further Analysis**

This is the most important analysis of the results of the listening test. This is a set of comparisons between all scenarios, and indicates with a p value of
less than 0.05, if there is any statistical difference between them. All other comparisons and graphs are good indicators, but Figure 94 is the best evaluation for the test.

![Comparison Mean Difference Grades](image)

Figure 94: **Difference Grade Comparison.** The reference point is Scenario 1 (blue), all gray scenarios are not significantly different. All red scenarios are significantly different.

There are clearly 8 groups. The members of each group have no significant difference between them. The first group is conformed by scenarios 1, 6, 11, and 16. Notice that none of these scenarios presented aliasing in the Bosi evaluation. They also correspond to the same frequency of $f_c = 2637.02\,Hz$. This concludes that there is no significant difference between all the methods at this frequency. Since there is no difference, they all the methods are ranked equal (1st). For this ranking see Table 15.
This process was repeated for each frequency, finding a ranking of all the methods. The case of $f_c = 3,520.00\, Hz$ showed that scenarios 7, 12, and 17 had no difference, while 2 had a significant larger mean. This concludes that 7, 12, and 17 ranked the same (1st), while 2 ranked worst (2nd).

The case of fundamental $f_c = 4,434.92\, Hz$ had a similar behavior. Scenarios 3 and 13 had no difference. Scenarios 8 and 18 also had no difference. But 3 and 13 had a difference with 8 and 18. This ranks 8 and 18 first, lower grade, and 3 and 13 second, larger grade.

In the fundamental $f_c = 5,587.65\, Hz$, there were three different rankings. The worst were Scenarios 4 and 14, which had no difference in between them, they were ranked 3rd. Followed by scenario 9, which was significantly lower, ranked 2nd. Finally the best scenario had a significant lower grade than all the other three scenarios and was number 19, being ranked 1st.

For the last case, $f_c = 6,271.92\, Hz$, four clear difference were established, putting only one method per rank. The 4th place was the higher significant grade, scenario 5. The 3rd place performed better than 5, being Scenario 15. The 2nd place performed significantly better than 5 and 15, which was scenario 10. The 1st place performed better statistically than scenarios 5, 10 and 15, which was scenario 20. Table 15 shows the entire ranking of all the scenarios and samples.

The conclusion of the listening test is that the behavior was very close to the evaluation done with the Bosi method. There were only two main
Table 15: **Ranking of Results of Listening Test**. Comparison of evaluation strategies in terms of analyzing aliasing error, harmonic envelope error or both. And simulating hearing threshold effects, masking effects, or both.

<table>
<thead>
<tr>
<th>MIDI</th>
<th>Frequency (Hz)</th>
<th>BLK 8</th>
<th>GA 8</th>
<th>BLK 13</th>
<th>GA 13</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>2,637.02</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>105</td>
<td>3,520.00</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>109</td>
<td>4,434.92</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>113</td>
<td>5,587.65</td>
<td>3</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>115</td>
<td>6,271.92</td>
<td>4</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

discrepancies. The scenarios 8 and 18 had aliasing present in Bosi, but were ranked in the listening test as if there was not any. These two cases correspond to the GA 8 and GA 13. All other samples behaved as expected. Explanation of why this occurred will be found in the next Chapter.

### 4.4 Results from GA Phase III (Aliasing and Harmonic Decay Optimization)

The GA was then run in its third phase. The idea was to obtain an output that was optimized for both the aliasing and the harmonic envelope decay. The GA uses a lot of memory and processing power running this phase, given that it is evaluating the fitness function 88 times for each variable set. That is to verify all of the 88 MIDI notes are perceptually alias free. The GA was run several times, but the results were not as good as expected. The harmonic envelope decay was very hard to optimize, having always a few peaks below the threshold curve. There was only one optimized solution for this phase, it can be observed in Table 16. This solution has only one peak below the hearing threshold curve (compared to trivial solution) at only one fundamental frequency,
\[ f_c = 2,794.9\,Hz, \] which corresponds to MIDI 101. The trivial had only 2 harmonic peaks below the threshold, while the GA solution had 3.

<table>
<thead>
<tr>
<th>GA</th>
<th>N</th>
<th>m</th>
<th>n</th>
<th>Rect.</th>
<th>Blackman</th>
<th>GA</th>
</tr>
</thead>
<tbody>
<tr>
<td>31</td>
<td>32,768</td>
<td>5</td>
<td>4</td>
<td>495.12</td>
<td>3,336.91</td>
<td>574%</td>
</tr>
</tbody>
</table>

Table 16: **Highest Fundamental Frequency Perceptually Alias-Free Comparison (Phase III).** The first column is the number of identification of the solution with its correspondent \( m \) and \( N \), referred as GA 31. The fifth, sixth and eighth columns are the highest fundamental frequencies perceptually alias-free for the Rectangular, Blackman and GA solutions respectively, all indicated in Hz. The seventh column corresponds to the improvement of using the Blackman window over using no window at all or Rectangular. The ninth column is the improvement percentage of the GA solution over no window. The tenth column is the improvement of the GA solution over the Blackman window.

As with the previous optimal solutions from GA Phase II, the coefficients for the solutions from Phase III can be observed in Tables 17 and 18 for the factor and power respectively.

<table>
<thead>
<tr>
<th>GA</th>
<th>( a_1 )</th>
<th>( a_2 )</th>
<th>( a_3 )</th>
<th>( a_4 )</th>
<th>( a_5 )</th>
<th>( a_6 )</th>
<th>( a_7 )</th>
<th>( a_8 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( a_9 )</td>
<td>( a_{10} )</td>
<td>( a_{11} )</td>
<td>( a_{12} )</td>
<td>( a_{13} )</td>
<td>( a_{14} )</td>
<td>( a_{15} )</td>
<td>( a_{16} )</td>
</tr>
<tr>
<td>31</td>
<td>-52</td>
<td>-85</td>
<td>64</td>
<td>-78</td>
<td>-63</td>
<td>68</td>
<td>71</td>
<td>-42</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>-93</td>
<td>41</td>
<td>-75</td>
<td>-73</td>
<td>12</td>
<td>51</td>
<td>88</td>
</tr>
</tbody>
</table>

Table 17: **Factor Coefficients for Input Functions Results of GA (Phase III).** The coefficients corresponds to only the 9 unique solutions and not the 14 solution involving different table size or correction points number.

<table>
<thead>
<tr>
<th>GA</th>
<th>( b_1 )</th>
<th>( b_2 )</th>
<th>( b_3 )</th>
<th>( b_4 )</th>
<th>( b_5 )</th>
<th>( b_6 )</th>
<th>( b_7 )</th>
<th>( b_8 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( b_9 )</td>
<td>( b_{10} )</td>
<td>( b_{11} )</td>
<td>( b_{12} )</td>
<td>( b_{13} )</td>
<td>( b_{14} )</td>
<td>( b_{15} )</td>
<td>( b_{16} )</td>
</tr>
<tr>
<td>31</td>
<td>19</td>
<td>9</td>
<td>26</td>
<td>10</td>
<td>13</td>
<td>11</td>
<td>28</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>23</td>
<td>12</td>
<td>17</td>
<td>21</td>
<td>17</td>
<td>11</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 18: **Power Coefficients for Input Functions Results of GA (Phase III).** The coefficients corresponds to only the 9 unique solutions and not the 14 solution involving different table size or correction points number.

Figure 95 shows the input function obtained with Phase III. The graphs
shows the rippling it has due to its $n = 4$ zero crossings. It also shows that the window is narrower than its Blackman correspondent. The rippling is less noticeable in the GA solution because the windowing has a faster decay to zero. Figure 96 shows the corresponding BLEP residual; observe the rippling as well.

![Windowed Input Functions Obtained With GA](image)

**Figure 95:** *Optimal Input Function for GA Phase III.* GA output for Optimal Input Functions of GA Phase III.

In order to evaluate the output from Phase III, all the same measurements from Phase II were used. Only the NMR, PEAQ, RMSE, and Maximum Absolute Harmonic Error showed clear results. Figure 97 shows the comparison for the GA solution 31. It can be observed that it performs the same as the Blackman solution for the lower frequencies. Around its highest fundamental alias-free, the Blackman starts to have a higher NMR than the GA solution. At higher frequencies, the performance of the GA becomes clearly better than the
Figure 96: **Optimal BLEP Residual for GA Phase III.** GA output for Optimal BLEP Residuals of GA Phase III.

Blackman case.

Figure 98 shows the PEAQ evaluation for GA 31. Remember that this measurement has an issue at the lower frequencies where the trivial waveform wrongly performs better. Above this range, the PEAQ measurement shows an overall improved performance of the GA over its Blackman version. The area of interest, which is higher frequencies, shows an even greater difference between the two methods. Both NMR and PEAQ show an improvement of the GA over Blackman, confirming the Bosi evaluation for this solution and giving conclusion that the aliasing was reduced in a more appropriate manner than when Blackman is used.

For the harmonic error, RMSE and Maximum Absolute Harmonic Error
Figure 97: **NMR Evaluation for GA Phase III.** Comparison of NMR value for GA 31 (light blue circle), trivial case (red x), and its corresponding Blackman case (black x). GA 31 was the optimal implementation found to construct an perceptually alias-free MIDI keyboard found in GA III.
Figure 98: **PEAQ Evaluation for GA Phase III.** Comparison of PEAQ value for GA 31 (light blue circle), trivial case (red x), and its corresponding Blackman case (black x). GA 31 was the optimal implementation found to construct an perceptually alias-free MIDI keyboard found in GA III.
are used. Figure 99 shows the RMSE performance of the Blackman case and the GA 31. As shown before, the RMSE is very similar for all cases. In this case, the highest fundamental frequency is notable because the RMSE for the GA case is slightly lower stating at that value and going higher. At higher frequencies, the difference becomes more noticeable, giving an overall improvement of the GA over Blackman.

![RMSE Analysis](image)

Figure 99: **RMSE Evaluation for GA Phase III.** Comparison of RMSE value for GA 31 (light blue circle) and its corresponding Blackman case (black x). GA 31 was the optimal implementation found to construct a perceptually alias-free MIDI keyboard found in GA III.

Figure 100 shows the Maximum Absolute Harmonic Error for the GA 31 solution compared to its Blackman version. This solution has an overall improvement over Blackman that is very noticeable, specially at lower frequencies. The error was reduced from an average of 16.5 dB to 10.5 dB, that is
around 5 dB of improvement. Even at higher frequencies, the improvement is very clear. This proves that the optimization for the harmonic decay is working well without affecting or trading off the aliasing.

In conclusion, the solution optimized for both aliasing and harmonic error, performed better than its Blackman version in all measurements of both aliasing and harmonic decay. This confirms that the solution optimized both aspects, making this the best solution so far in this research in terms of performance. Solutions from Phase II are also very good when used for each aspect separately. The NMR-TR, PEAQ-TR measurements did not provide clear information, given the previously discussed error at low frequencies appeared again.
5

Discussion

There were many discoveries and discussions in the research that are worth mentioning. Because of this, it is better to organize them in main sections. The first one will cover all the findings in the computational and perceptual evaluation part of the research. The second section will cover the developed tools and methodology during the project that are useful for similar investigations in the future. The third section will cover how the improvements of the research can be easily implemented in a real-time synthesis environment. Finally, in the last section, all the unexplored paths and ideas, that due to time reasons were not able to be implemented in this research, will be discussed; as well as future work that might improve or help in any way the findings of this or any other related investigation.

5.1 Findings in Computational and Perceptual Results

In general, the findings during the computational analysis and the listening test coincide with previous research. There were some introductions of methodology in this research like the A-Weighted Error and the NMR-TR and PEAQ-TR. There was also some behavior that had not been reported in other research like breaking the paradigm of the “one zero crossing per one correction point”. All of these findings and any other relevant information that was obtained from this research will be thoroughly discussed in this section.
5.1.1 Type of Input Functions Found with GA Phase II

The first idea to discuss is the type of windowing and final input functions the GA found optimal during its Phase II, where only aliasing was the optimized variable. Most of the input functions seemed to be narrower around its center than commonly used options like the regular Blackman windowing.

Narrower Windowing Improves Aliasing

The best 3 results from the GA Phase II were of course the ones that were able to pass the limit of the highest note of the piano with no perceivable aliasing. These were the GA 8, 13 and 14. In order to have a clear view of what is going on with the windowing part of the input, Figure 101 has a comparison of these three GA solutions with the Blackman window. Note that all three windows are narrower than Blackman and all of them had better aliasing results in the Bosi, NMR, PEAQ, and A-Weighted Aliasing Error evaluations. The GA 13 was the one with the best result going up to $f_c = 4,472.6 Hz$, which compared to GA 8 has a narrower profile. GA 14 has a narrower profile than GA 13, but it has $m = 5$ correction points instead of 4, which shortens the highest fundamental frequency alias-free because of the issue of changing more points at higher fundamentals. In the overall aliasing evaluation, the NMR value of GA 14 was slightly lower than GA 13 at lower fundamentals, which means that its performance in eliminating the aliasing was better.

The conclusion of this section is that a narrower window will give a better aliasing elimination. The reason for this is that a narrower window tapers the
Figure 101: **Comparison of Best Inputs Windowing for Aliasing.** Narrower windows showed a better performance in reducing the aliasing.

Rippling more than a wider one, refer to the Trade off between Aliasing and Harmonic Envelope Decay section to see why this improves the aliasing performance. This is of course limited at high frequencies, where the number of correction points restricts the performance to get a higher fundamental perceptually alias-free; but this only happens when doing a single correction BLEP. One way to evade this limitation is to use multiple correction. When the number of possible points of correction is small, correcting the same discontinuity area several times improves the performance of the BLEP, but it also requires more computations.

The linear harmonic amplitude has been proved to have a direct relationship with the aliasing or harmonic optimization. If the first harmonics,
including the fundamental, have a larger amplitude than its trivial version, the alias is reduced more. These type of effects were found to be produced only by the BLEP methods with narrower windows. The boosting of the amplitude of the first harmonics introduces a larger error than the reducing factor would have in the case of a wider window, which will be explained in the next section.

**Wider Windowing and Rippling Improve Harmonic Amplitude**

On the other hand, the performance in Harmonic Decay is exactly the opposite of what happens with aliasing in the windowing analysis. A wider window will give a better performance in the harmonic amplitudes having a smaller error in the evaluations of Maximum absolute harmonic Error, RMSE and A-Weighted Harmonic Error. The reason for this is that the wider window will let more rippling pass through. Refer to the Trade off between Aliasing and Harmonic Envelope Decay section to see why this improves the harmonic envelope performance.

The wider windows like the GA 1 in comparison to Blackman and all the other wider GA inputs functions always showed having a very small error in the harmonic decay. Observe the windowing factor in Figure 102. The explanation for this is that the wider windows always introduce less harmonic amplitude in all of the harmonics in comparison to the trivial, instead of boosting the first harmonics like the narrower windows. The amplitude difference is minimal in this cases and the total error is reduced.

The important finding here is that also the narrower windows showed an
improvement over the Blackman, even though the latter is wider. This means that the trade off between harmonic decay and aliasing error is not that discrete. The GA solutions were optimized for Harmonic Decay, by adding more rippling to it, keeping the same windowing factor and they performed better than Blackman in the harmonic decay analysis.

Figure 103 shows an example of the case when a narrower window improves the harmonic envelope as well as the aliasing. The GA solution 31 of Phase III has been optimized for both cases. It gave a better performance in aliasing and harmonic decay than its Blackman equivalent. The harmonic decay or the aliasing were not as optimized as if it had been done separately, but it did reach a frequency higher than the 88th key in the MIDI Piano and it had a
better NMR, PEAQ, RMSE and Maximum Absolute Harmonic Error than its correspondent Blackman version.

Observing that the GA solutions with a wider windowing factor gave better performance in harmonic decay means that it does depend on this factor, but not as heavily. The conclusion of this discussion is that the Harmonic decay depends more on the rippling than in the windowing factor.

5.1.2 The One Correction Point per Zero Crossing Paradigm

Before getting into the trade off between the aliasing and harmonic envelope decay, it is necessary to discuss the problem of being restricted to using the same number of zero crossings $n$ and correction points $m$. Remember that
the results of this research apply for this exact implementation of the method, it might not be the same on different variations of BLEP. The idea of having the option to modify a different number of points, delivers many possible combinations that an algorithm, like the GA, can use as an advantage and obtain an optimal solution. The direct relationship is based on that the $sinc$ function has a zero crossing per each cycle period, being able to produce a good rippling on the BLEP residual when taking one point of each of these cycles. But when windowing is applied, this rippling is not so clear or marked anymore. It usually tapers off on both ends, and the points far from the center will have no rippling at all, with amplitudes of almost zero, that in reality will not modify significantly the BLEP waveform. This is why when trying all the different inputs, this paradigm was broken, and having all these different combinations, the GA was able to find optimal solutions for both aliasing and harmonic envelope decay.

5.1.3 Trade off between Aliasing and Harmonic Envelope Decay

The number of zero crossings has a direct relationship with the trade off between aliasing and harmonic envelope decay described above.

The Original BLEP method windowed with Blackman was found to perform better for the harmonic decay than for aliasing, when having a large $m$. It of course relates to the fact that the number of correction points is the same as the number of zero crossings of the $sinc$ function. So when having a large $m$, it would also have a large $n$ of up to 4 or 5, which delivers a lot more rippling to the final waveform.
For the GA method, the finding was that having an independent number of \( n \) and \( m \), the harmonic envelope presented a lower decay in the cases of a large \( n \), more rippling, and a smaller \( m \), along with the smaller table size \( N \). The best combination would be the smaller table \( N = 4,096 \), with a big rippling \( n = 4 \) (optimized for harmonic decay), and the fewer possible corrections \( m = 2 \), which corresponded to GA 1.

On the other hand, the best performance of the GA for aliasing reduction, was found to be efficient at: the larger table size \( N \), a larger \( m \), and the minimum rippling \( n \) possible. This solution was found to be GA 14, which had a larger \( N = 28,672 \), correcting \( m = 5 \) points, having no rippling at all \( n = 1 \), being optimized for aliasing only.

**Rippling vs. No Rippling**

The results found on the analysis using several evaluation techniques always pointed to the fact that the more rippling there is, the less decay there is in the harmonic envelope. It also proved than the less rippling there is, the more aliasing reduction occurs. This is the exact definition of the trade off. The optimal solution for both cases (mixed) could be to find an input which improved the aliasing at least in the piano frequency range to get the highest note to be perceptually alias-free, and that it had a very small effect in the harmonic envelope. This is a compromise that can be done. It would not be the best optimized cases separately, but together could give very good results, at least perceptually and inside the piano range.
Perceived Harmonic Decay of BLEP

The main issue when trying to improve the harmonic decay is that none of the harmonic peaks that are perceived in the trivial case, fall down to an amplitude where they become imperceivable. The harmonics are more likely to not be perceived at higher frequencies, due to higher levels of the hearing threshold in that range. This is a main issue and it is why optimizing the harmonic decay error is so important. A very small decay in the amplitude of a high frequency harmonic peak can change it from being perceived to not being perceived at all.

5.1.4 Effects of Table Size with Constant Rippling

The table size is not important at low frequencies, the same NMR, and aliasing reduction in general, was found for tables of different sizes and same number of correction points. The difference becomes noticeable at high frequencies, where the larger table size always showed to have a highest fundamental perceptually alias-free and an overall lower NMR value at higher frequencies than the perceptual limit. This applies for both, the case with no rippling at all (optimized for aliasing) and the case with rippling (optimized for harmonic decay).

It is also important to note that the RMSE and the A-Weighted Aliasing Error, both showed no clear dependence on the table size, only in the number of zero crossings (rippling or not rippling). The case for the Maximum Absolute Harmonic Error is not that simple or discrete. When a small table is compared
to a larger table, both optimized for only aliasing (no rippling), the larger table will have a slightly larger error in this measurement. When there is rippling, optimized for harmonic decay, the table size seems to have no effect on the error.

5.1.5 Aliasing Perceptibility on Fundamentals Close to the Highest Fundamental Perceptually Alias-Free

In the listening test, note that in two of the scenarios (8 and 18), aliasing was not identified, even though it was present on the Bosi method. The reason for this is that the aliasing present in all cases was minimum. This can be observed in the Bosi graphs in Figures 104 and 105.

Figure 104: Bosi Output for Listening Test Results (GA 8). Scenario 8 with a fundamental $f_c = 4,435.5469$ Hz. BLEP GA 8 method using $N = 28,672$ and $m = 3$.

Figure 104 shows the case of Scenario 8 at the frequency where no aliasing was identified by the subjects in the listening test. There is only one peak of
aliasing above the hearing threshold curve. It is placed at lower frequency than the fundamental. The explanation for the subjects not identifying this aliasing is that its amplitude barely makes it above the curve, so only a very well-trained ear would be able to hear or differentiate that small aliased peak. Not that the GA 8 solution has a very good performance on hiding the aliasing.

Figure 105: Bosi Output for Listening Test Results (GA 13). Scenario 18 with a fundamental $f_c = 5,586.9141\, Hz$. BLEP GA 13 method using $N = 28,672$ and $m = 4$. The second case of Scenario 18 is similar. Figure 105 shows the Bosi output for GA 13. This case is even harder to identify than the one for Scenario 8. The amplitude of the only aliased peak above the curve is even smaller and is practically sitting on top of the curve. Even for a well-trained ear, it would be really hard to identify this aliasing correctly. The reason for this is that one more point is being corrected in GA 13 than in GA 8, which delivers an even better
alias reduction.

The conclusion of this is that the GA solutions work very well in hiding the aliasing inside the hearing threshold and masking limits. This indicates, that even if Bosi identifies aliasing at frequencies slightly higher than its highest perceptually alias-free fundamental, actual human subjects might not hear this aliasing. The reason is, because there would only be a few number of peaks, sometimes just one; and they will have a very small insignificant amplitudes, sitting close to the curve limit.

5.1.6 Aliasing Below and Near the Fundamental Frequency

The aliasing with the more issues is the one that is close or below the fundamental frequency. This is observed in Figures 104 and 105.

The results found with the GA have a better distribution of aliasing in its highest fundamental than other methods, according to its better Bosi values. The conclusion here is that the reduction of the entire aliasing is not as important as how it is distributed, specially at lower frequencies. This corroborates previous research that states that the most critical aliasing is the one near the fundamental [Lehtonen et al., 2012].

The most simple way to look at this issue is graphically. Figures 106 and 107 draw a new conclusion as it that the few aliasing peaks that are first heard are always around the fundamental frequency. This area has a lower hearing threshold and almost no masking.

Figure 106 shows the Bosi output for the Blackman original BLEP
Figure 106: **Bosi Output for Blackman (Aliasing Distribution)**. Fundamental frequency of $f_c = 5,000.9766\,Hz$. BLEP Blackman method using $N = 28,672$ and $m = 4$. 
implementation at a frequency of around 5 kHz. The aliasing that is above the
curve is mostly near the fundamental, specially sitting at a lower frequency than
it. The Blackman method used a $N = 28,672$ and $m = 4$ correction points. This
case has a highest fundamental perceptually alias-free of $f_c = 3,193.35 Hz$. It
hides successfully the aliasing at high frequencies, but it is not able to hide the
more critical one at lower. On the other hand, the corresponding implementation
of GA 13, with the same $N$ and $m$ as the Blackman case shows no aliased peaks
above the limit curve, see Figure 107. Even though its highest perceptually
alias-free fundamental is only $f_c = 4,447.26 Hz$; at the 5kHz example, it spreads
the aliasing in a much more efficient manner than the Blackman implementation,
hiding it completely for the human hearing system.

Figure 107: Bosi Output for GA (Aliasing Distribution). Fundamental frequency
of $f_c = 5,000.9766 Hz$. BLEP GA 13 method using $N = 28,672$ and $m = 4$. 
The conclusion is that the solution found with the GA for optimized aliasing have a better distribution of the aliasing, specially around or before the fundamental, where it is more critical due to the lower hearing thresholds.

5.1.7 Aliasing Near the Nyquist Frequency

The aliasing present near the Nyquist frequency is not crucial to the perceptual models and neither is to the real human subjects. The reason for this is that the hearing threshold and masking levels are very high at higher frequencies, so none of the component are perceived, even the harmonic components in some cases. This effect can be seen in Figures 106 and 107. The aliasing that is crucial is the one found below of near the fundamental peak as explained in the previous section. This is also explained in the previous Perceived Harmonic Decay of BLEP section, just as harmonics are easily turned from being perceived to not being perceived at all, so are aliasing.

5.1.8 A-Weighted Evaluation for Harmonics and Aliasing Errors

A method to evaluate the performance of an oscillator algorithm that has not been used before in this area was introduced in this research, the A-Weighted measurement. It is a very computational inexpensive method.

This evaluation measurement proved to be a good indicator of the performance of the oscillator over the audible frequency range. It also showed to have a very similar behavior to those of NMR, RMSE and Maximum Harmonic Absolute Error. It is not as good as the other models, because it is only base on hearing threshold levels and does not include masking, but it does give a good
approximation.

The measurement indicates actually two different values. The first one is the error of only the aliasing components, which can be compared to NMR or PEAQ. The second one is the error of the harmonic peaks only, comparable to RMSE or Maximum Absolute Harmonic Error.

5.1.9 PEAQ Evaluation

As in previous research [Välimäki et al., 2012], the PEAQ method showed a wrong behavior at lower frequencies. The trivial performs better than all the methods, even though this is proven to be impossible. This makes this evaluation not as trustworthy as NMR or Bosi. At higher frequencies, its behavior seemed to be normal and the results seem to be related to NMR and the A-Weighted Error for Aliasing only. There is not enough evidence to conclude if this method is not useful at all, or if the information above the estrange behavior zone can be actually used.

5.1.10 PEAQ and NMR including Trivial Harmonic Decay

The measurements for PEAQ and NMR when being compared to the trivial harmonic decay were also introduced in this research. The evaluation phase showed that the NMR-TR was a good measurement when the methods being tested were optimized for aliasing only. If the methods were optimized for harmonic decay, this method did not yield any meaningful results and had the same issue the regular PEAQ has at lower frequencies.

The PEAQ-TR had the complete opposite success as NMR-TR. It did
yield good indications only when the method tested were optimized for harmonic decay.

Given that these two methods are designed for evaluation of both aliasing and harmonic decay effects, the best way to test them is to used them in the results of the Phase III of the GA, where both problems are optimized.

Sadly, the results from these measurements using the results from Phase III had the same issues at low frequencies. The conclusion is that these methods provide some information that is useful, specially at high frequencies, but they are not a good indicator given their many issues. These evaluation techniques could be developed further to obtain more relevant, useful, and trustworthy information from them.

5.1.11 Normalization of Perceptual Loudness

In order to do a correct perceptual analysis, specially the listening test, of the samples obtained in this research, the audio had to be normalized based on the Normalizing Perceptual Loudness in Chapter 4. This normalization showed having very little effect on the perceptual computed models. It was so small, that the general results did not seem to be affected at all.

So focusing on only human subjects, whom are the ones that actually listen the waveforms produced by the synthesizer, the conclusion is that each different BLEP method implementation does change the perceptual loudness of the waveform in a unique manner. The more or less constant maximum harmonic absolute error during the MIDI keyboard frequency range implies that the
harmonic difference is around the same for all the fundamental frequencies, and
it only depends on the method applied.

This gives an idea, that for each method there should exist a correction
factor that works for all fundamental frequencies and it is basically trying to fix
the harmonic envelope decay introduced by the BLEP method. This would be a
new parameter for each BLEP table along its size and number of correction
points. Having this parameter calculated would have a smoother change when
switching between different types of BLEP implementation in a continuos form,
specially for real time synthesis. The goal is to maintain around the same
perceptual loudness for all different implementations of the BLEP method in a
single synthesizer. An example of this would be having a synthesizer with a knob
that switches between different BLEP implementations to obtain slightly
different sounds. This might even be a good idea for a synthesizer, where
different and very perceivable aliasing can be used in a positive manner.

5.2 Computational Complexity and Memory Used

This research was optimizing the BLEP method in order to get a better
performance of high frequency fundamental perceptually alias-free, but it also did
it while trying to optimize the computational costs and the amount of memory
involved.

Note that if comparing two methods with these values constant, the
performance depends only on the best result and not on the memory or
processing required. This is why all the Blackman corresponding references for
each of the GA output had to have the same $N$ and $m$ in order to make a fair comparison.

Computational complexity is based on the number of points to correct or look up table times. When there are more points to correct, the method has to process information more time in each cycle. When the fundamental frequency is high, the method has to do the same number of calculations in an even shorter time. Reducing the number of correction points is crucial for an efficient implementation, specially if it is in real-time synthesis.

Memory consumption is based on the size of the table only. The largest table found in Phase II of the GA was of a size of 28kB. This is not a very large memory amount, specially for the available computing resources nowadays. Because of this, the memory consumption is considered less important than the computational costs, and an optimal solution with smaller $m$ will be chosen without the $N$ being much considered.

5.2.1 Cutting the Memory Used in Half

Note also that the BLEP residual stored in the table is symmetrical, so it is only necessary to store one half of the table, and the other one can be calculated by a simple multiplication. This might be of great value if the BLEP is being implemented in a chip that has some restrictions in memory or if there is a need to implement several versions of the BLEP method inside the same system.
5.3 Developed Tools and Methodology

A large amount of useful tools were developed over this entire research. These were mainly Matlab implementations of all the algorithms discussed over this document. Many of these methods so not have an implementation available online. Some of them do have available implementations but are not coded for Matlab or are obsolete due to the constant updates of the integrated Matlab functions. One of the purposes of this research was to develop this set of tools in a useful manner so that any person in the future can use them for their own investigation. All the code will be included in the Appendix of this document.

The innovative implementations are the algorithms that: create the BLEP input, create the BLEP residual table, create the BLEP and trivial waveforms, do additive synthesis using BLEP or trivial amplitudes, do the Bosi analysis, find the highest perceptually alias-free fundamental, match the perceptual loudness based in ITU standard, and many more. The updated methods were the ones for calculating PEAQ and NMR.

An important contribution was the methodology for the running process of the GA for particular synthesis problems. The implementation of the GA can be used with any other type of oscillator synthesis. So any other investigation that is trying to improve a different method than BLEP can take advantage of the coded GA algorithm as well.
5.3.1 Parallel Computing

All the algorithms developed during this research were adapted to be used in parallel computing using the Parallel Computing Toolbox of Matlab. This is a very important factor, given that specially the Bosi and PEAQ methods, as well as the additive synthesis take long times to run in normal processing. The most important use of parallel computing was for the GA, because it evaluates many variables every iteration, than each additional core is cutting the time by a factor of $1/\text{number}_{\text{cores}}$.

5.4 Integration with Real-Time VA Synthesis

An important part of the oscillator synthesis research is mainly focus on getting the algorithms to work efficiently in real-time. The reason for this is that almost all sounds coming out of a synth are requested in real time by a plugin User Interface (UI) or by an external MIDI controller. Therefore, the algorithms implemented have to be tested to work well, so no clipping or undesired delays are introduce to the synth.

5.4.1 RackAFX Implementation

The software RackAFX is a free platform used for creating plug-ins in Windows. It can produce plugins for different APIs such as: RackAFX, VST2, VST3 and AU [Pirkle, 2014]. The program has several internal synthesizer plugins that implement the BLEP method. The results of this research can be easily introduced into this software for real-time testing. The only modifications needed are to store the BLEP residual table obtained from the GA and to
indicate the desired number of correction points.

5.5 Unexplored Options and Future Work

Due to restrictions of time during this research, many options were not tested that could contribute to even better results. This section mentions just a few of the more important unexplored paths that might be of interest in a future to continue developing this research.

5.5.1 Optimizing the Algorithm

There is definitely more room for optimizing the algorithm. The GA efficiency depends mostly in the efficiency of its fitness function. Having a very good, fast, and efficient fitness algorithm will deliver faster and more adequate results. The implementation of the fitness function depends of many smaller category algorithms. Each can be optimized individually and as a group. Doing this would let the GA run more freely, in a larger search space, thus increasing the possibilities of finding more optimal solutions.

5.5.2 A Different Approach for Creating the Input Function

Several manners to create the input function were tested. Finally the one implemented was the mixture of the 17 windows and the sinc function with different number of zero crossings. This is definitely not the only option available. The BLEP method has some restrictions due to the shape of the optimal input an its required symmetry, but there could be an infinite amount of ways to create input functions that might or might not be more effective that the
one developed in this research.

The findings suggest that the window should have a Gaussian shape. Triangular or rectangular windows produce the worst results. Wide windows are also a bad starting point. In the case of the function, it definitely has to be sinusoidal. The rippling of the sinc function produced the best performance because of its tapered shape. Narrower main lobes are a good starting point, they keep the discontinuity as steep as possible.

5.5.3 Interpolation

Synthesis based on tables have proved to give better results when interpolation is used [Välimäki et al., 2012]. When a larger table is used, interpolation might not be as effective. This research did not implemented any interpolation in neither the smaller nor the larger tables. This is one of the main options that should definitely be tested because the GA could benefit in a great manner from a feedback including an improvement due to interpolation.

5.5.4 Multiple Correction

The research was focused on only performing a single correction on every discontinuity area. When the fundamental frequency increases the number of possible correction points becomes smaller, making it more difficult for the BLEP method to reduce the aliasing. In this case, multiple correction is often used, also know as Multi-BLEP. This method corrects the same discontinuity area twice or even more times in order to create a better waveform in terms of aliasing reduction. This is definitely a topic worth exploring using the findings of this
research and better yet, obtain new BLEP tables optimized for different Multi-BLEP methods using the already implemented GA.

5.5.5 Other Waveforms and Discontinuity Types

The main focus of this research was the reverse sawtooth waveform. All the findings in this document can easily be transported into a different waveform that has discontinuities such as the square wave. All waveforms with discontinuities will have aliasing problems when being implemented in the trivial mathematical manner. It is very important to corroborate that all the theories proposed during this research apply to other types of discontinuities as well. Once this is proved, the BLEP method developed here would be robuster and more ready to make it to an implementation in a real end-user application.

Both the square and the sawtooth waveforms have discontinuities of type $C1$. This discontinuity only occurs when the slopes before and after the discontinuity are the same. Unfortunately, this is the only case the BLEP method can correct, making it a limitation to correct other types of discontinuities like the hard synced triangle waveform, which has an inverse slope at one side.

An area of improvement for the BLEP technique would be to obtain a different input function which could correct these other types of discontinuities. This becomes a more complex task given the input cannot be symmetrical in most cases.
5.5.6 Another Listening Test

Another listening test is considered; compare the results from the GA based on the Bosi model and the results from a GA based on the NMR or any other model. Even though Bosi has proven to be a really good indicator of the human perceptual hearing, perhaps results optimized with other method would yield different or even better results when actually being tested in human subjects. The implementation of the GA based on the other methods is discussed in Chapter 3. Only slight modifications have to be made in order for the fitness function to work with any other computed perceptual model.

5.5.7 Manipulating the Aliasing

The presence of aliasing is not desired in most cases, but it might be wanted for specific applications. The developed GA is very open to the modification of the parameters used to constraint the research. Modifying the GA to find optimal solutions for problems where aliasing is desired would be simple. The idea would be to create BLEP tables that allow aliasing at specific frequencies or overall distributed under a curve. This implementation would be a great tool to use in real time synthesizers. An example would be the development of a synth plugin which has different aliasing conditions.
LIST OF REFERENCES


function [DFTdBFS, DFTFS, DFTdB, DFT, freqArray] = calculateDFT (fs, ... 
    input, NDFT, plot)
% CALCULATEDFT Calculates Discrete Fourier Transform (DFT) of an audio signal.
% [DFTdBFS, DFTFS, DFTdB, DFT, freqArray] = calculateDFT (fs, ... 
    input, NDFT, plot)
% Copyright 2015.
% Francisco J. Valencia
% Music Engineering and Technology
% University of Miami
% INPUTS
% fs (double): sampling rate of input audio signal.
% input (1xn double array): input vector of audio signal (MONO).
% NDFT (double): size of DFT.
% plot (bool): true for drawing plots.
% OUTPUTS
% DFTdBFS (1xNDFT/2 double array): DFT in dB Full Scale (0dB=max).
% DFTFS (1xNDFT/2 double array): DFT in linear Full Scale (1=max).
% DFTdB (1xNDFT/2 double array): DFT in dB raw results.
% DFT (1xNDFT/2 double array): DFT in linear raw results.
% freqArray (1xNDFT/2 double array): frequency array from 0 to Nyquist (fs/2) Hz.
% [a,"] = size(input);
% if a ~= 1
%    error('Audio signal must be MONO (1-channel)')
% end
L = length(input);
lowerLimitdB = -96;
% Use Dolph-Chebyshev Windowing
window = chebwin(L);
inputWin = input.*window';
% DFT
input2FFT = fft(inputWin,NDFT);
inputFFT = input2FFT/L;
DFT = 2*abs(inputFFT(1:NDFT/2));
% DFTdB
DFTdB = 20.*log10(DFT);
% DFTdB
maxinputFFT = max(DFT);
inputFFTAbs = DFT/maxinputFFT;
DFTFS = inputFFTAbs;
% DFTdBFS
DFTdBFS = 20.*log10(DFTFS);
% Frequency Array
freqArray = fs/2*linspace(0,1,NDFT/2);
if plot
    figure()
    plot(f/1000,DFT)
    title ('Frequency Domain (Spectrum) − Linear')
    xlabel 'Frequency (kHz)'
    ylabel 'Magnitude'
function [harmonicPeaks, harmonicLocs, nonHarmonicPeaks, ...
    nonHarmonicLocs] = findHarmonicPeaks (peaks, locs, fs, fc)
% FINDHARMONICPEAKS Separates the harmonic and non harmonic peaks of the
% power spectrum of a signal.
%
% [harmonicPeaks, harmonicLocs, nonHarmonicPeaks, ...
% nonHarmonicLocs] = findHarmonicPeaks (peaks, locs, fs, fc)
% Copyright 2015.
% Francisco J. Valencia
% Music Engineering and Technology
% University of Miami
%
% INPUTS
% peaks (1xn double array): array with the magnitudes of the peaks of the
% power spectrum of a signal.
% locs (1xn double array): array with the frequencies of the peaks of the
% power spectrum of a signal.
% fs (double): sampling rate of input audio signal.
% fc (double): fundamental frequency of waveform.
%
% OUTPUTS
% harmonicLocsBLEP (1xn double array): array with the frequencies of the
% harmonic peaks of a BLEEP signal.
% harmonicPeaksBLEP (1xn double array): array with the magnitudes of the
% harmonic peaks of a BLEEP signal.
% nonHarmonicLocsBLEP (1xm double array): array with the frequencies of
% the non harmonic peaks of a BLEEP signal.
% nonHarmonicPeaksBLEP (1xm double array): array with the magnitudes of
% the non harmonic peaks of a BLEEP signal.

nHarmonics = floor(fs/(2*fc));
nonHarmonicPeaks = peaks;
nonHarmonicLocs = locs;
harmonicPeaks = zeros(1, nHarmonics);
harmonicLocs = zeros(1, nHarmonics);
% Original harmonics
originalHarmonics = zeros(1, nHarmonics);
for i=1:1:nHarmonics
    originalHarmonics(i) = i*fc;
end
% Assign 0.0 to all undesired data
for k=1:1:nHarmonics
    absValues = abs(locs - originalHarmonics(k));
    [~, index] = min(absValues);
    harmonicPeaks(k) = peaks(index);
    harmonicLocs(k) = locs(index);
    nonHarmonicPeaks(index) = 0.0;
    nonHarmonicLocs(index) = 0.0;
end
% Eliminate all 0.0
nonHarmonicPeaks(nonHarmonicPeaks == 0.0) = [];
nonHarmonicLocs(nonHarmonicLocs == 0.0) = [];

function AWeigthFactor = calcAWeightFactor (freq)
% CALCAWEIGHTFACTOR Calculates the A-weighted factor correspondent to a
% frequency in Hz. Multiply this factor by the linear
% power spectrum of the signal at the particular
% frequency. As described in ANSI Standards S1.4–1983
% and S1.42–2001.

% [AWeigthFactor] = calcAWeightFactor (freq)
Copyright 2015.
Francisco J. Valencia
Music Engineering and Technology
University of Miami

% INPUTS
freq (1xn double array): frequency in Hz.

% OUTPUTS
AWeigthFactor (1xn double array): A-weighting factor.

AWeigthFactor = ((12200^2*freq^4)/((freq^2+20.6^2)*((freq^2+107.8^2)*...
(freq^2+737.9^2)*(1/2)+(freq^2+12200^2)))*10^(-2/20));

end

function [maxAbsError] = calcMaxAbsError (ref, test)
% CALCMAEXABERROR Calculates the maximum absolute error of two signals.

% [maxAbsError] = calcMaxAbsError (ref, test)
Copyright 2015.
Francisco J. Valencia
Music Engineering and Technology
University of Miami

% INPUTS
ref (1xn double array): reference signal.
test (1xn double array): signal to be tested.

% OUTPUTS
maxAbsError (1xn double array): maximum absolute error.
function [RMSE] = calcRMSE (ref, test)
% CALCRMSE Calculates the root mean squared error (RMSE) of two signals.
% [RMSE] = calcRMSE (ref, test)
% Copyright 2015.
% Francisco J. Valencia
% Music Engineering and Technology
% University of Miami
% INPUTS
% ref (1xn double array): reference signal.
% test (1xn double array): signal to be tested.
% OUTPUTS
% RMSE (1xn double array): root mean squared error.
RMSE = sqrt(sum((ref(:)−test(:)).ˆ2)/numel(ref));
end

function [output] = fixClipping (input)
% FIXCLIPPING Fixes clipping in an audio signal.
% [output] = fixClipping (input)
% Copyright 2015.
% Francisco J. Valencia
% Music Engineering and Technology
% University of Miami
% INPUTS
% input (1xn double array): clipped signal.
% OUTPUTS
% output (1xn double array): corrected signal.
fromMax = max(abs(input));
[output] = input/fromMax;
end

function [bark] = freq2bark (freq)
% FREQ2BARK Calculates the bark correspondent to a frequency in Hz based
% "Analytical expressions for critical band rate and critical
% bandwidth as a function of frequency," J. Acoust. Soc. Am. 68,
% 1523–1524.
% INPUTS
% freq (1xn double array): frequency in Hz.
function [MIDI] = freq2MIDI (freq)
% FREQ2MIDI Calculates the MIDI correspondent value to a frequency in Hz.
% 
% [bark] = freq2bark (freq)
% Copyright 2015.
% Francisco J. Valencia
% Music Engineering and Technology
% University of Miami
% 
% INPUTS
% freq (1xn double array): frequency in Hz.
% 
% OUTPUTS
% MIDI (1xn double array): frequency in MIDI.
% 
MIDI = 69 + 12.*log2(freq./440);
end

function [freq] = MIDI2freq (MIDI)
% MIDI2freq Calculates the frequency in Hz correspondent to a MID value.
% 
% [freq] = MIDI2freq (MIDI)
% Copyright 2015.
% Francisco J. Valencia
% Music Engineering and Technology
% University of Miami
% 
% INPUTS
% MIDI (1xn double array): frequency in MIDI.
% 
% OUTPUTS
% freq (1xn double array): frequency in Hz.
% 
freq = 2.^((MIDI−69)./12).*440;
end

function [hearingThresholdBSPL] = hearThres (freq)
% 
% [hearingThresholdSPLdB] = hearingThreshold (freq)
% Copyright 2015.
% Francisco J. Valencia
% Music Engineering and Technology
% University of Miami
% 
% INPUTS
% freq (1xn double array): frequency in Hz.
% OUTPUTS
% hearingThresholdSPLdB (1xn double array): hearing threshold in dB-SPL.

hearingThresholdBSPL = 3.64.*(freq./1000).^(-0.8) - ...
6.5.*exp(-0.6.*(freq./1000 - 3.3).^2) + (10.^(-3)).*(freq./1000).^4;
end
Appendix B - Psychoacoustic Models

```matlab
function [aliasPresent, nonHarmonicsAbove, harmonicsBelow] = BosiPAM ...
    (harmonicPeaks, harmonicLocs, nonHarmonicPeaks, nonHarmonicLocs, ...
    type, plot)
% BOSIPAM Applies the Bosi Psychoacoustic Model (PAM) to an audio signal
% [aliasPresent, nonHarmonicsAbove, harmonicsBelow] = BosiPAM ...
% (harmonicPeaks, harmonicLocs, nonHarmonicPeaks, nonHarmonicLocs, ...
% type, plot)
% Copyright 2015.
% Francisco J. Valencia
% Music Engineering and Technology
% University of Miami
%
% INPUTS
% harmonicPeaks (1xn double array): array with the magnitudes of the harmonic peaks of a signal.
% harmonicLocs (1xn double array): array with the frequencies of the harmonic peaks of a signal.
% nonHarmonicPeaks (1xm double array): array with the magnitudes of the non harmonic peaks of a signal.
% nonHarmonicLocs (1xm double array): array with the frequencies of the non harmonic peaks of a signal.
% type (string): 'dB' or 'linear' magnitudes of the peaks.
% plot (bool): true for drawing plots.
%
% OUTPUTS
% aliasPresent (bool): Determines if there is aliasing.
% nonHarmonicsAbove (1xp double array): array of frequencies of aliased peaks above the PAM curve
% harmonicsBelow (1xk double array): array of frequencies of harmonic peaks below the PAM curve
%
% Convert to bark
harmonicLocsBark = freq2bark(harmonicLocs);
nonHarmonicLocsBark = freq2bark(nonHarmonicLocs);
%
% Factors for downshift and scaling
downshift = 10; % dB for BL oscillator that contain nearly tonal maskers
d_refSPLdB = 96; % dB for the first harmonic
d_refSPL = 10^(d_refSPLdB/20); % Linear for the first harmonic
%
% Pre-processing on dB
if (strcmpi(type, 'dB'))
    if (isempty(nonHarmonicPeaks))
        minOffset = min(harmonicPeaks);
    else
        minHarmonicdB = min(harmonicPeaks);
        minNonHarmonicdB = min(nonHarmonicPeaks);
        minOffset = min(minHarmonicdB, minNonHarmonicdB);
    end
    harmonicPeaks = harmonicPeaks - minOffset; % Offset for negative numbers
    nonHarmonicPeaks = nonHarmonicPeaks - minOffset;
    maxHarmonicdB = max(harmonicPeaks);
    harmonicPeaks = harmonicPeaks.*d_refSPLdB./maxHarmonicdB;
    nonHarmonicPeaks = nonHarmonicPeaks.*d_refSPLdB./maxHarmonicdB;

% Pre-processing on Linear
elseif (strcmpi(type, 'linear'))
    maxHarmonic = max(harmonicPeaks);
    harmonicPeaks = harmonicPeaks.*d_refSPLd.MaxHarmonic;
end
```
nonHarmonicPeaks = nonHarmonicPeaks*refSPL/maxHarmonic;
harmonicPeaks = 20*log10(harmonicPeaks);
nonHarmonicPeaks = 20*log10(nonHarmonicPeaks);
else
eroerror('Must enter dB or linear')
end

% Hearing Threshold calculation
f = zeros(1, 22050−19);
for i = 20:1:22050
f(1, i−19) = i;
end
hearingThres = hearThres(f);
b = freq2bark(f);
S = zeros(length(b));

% Implement the masking curve
for i=1:1:length(harmonicLocsBark) %masker
for j=1:1:length(b) %maskee
 dz = b(j) − harmonicLocsBark(i);
 if dz < 0
  thetadz = 0;
 else
  thetadz = 1;
 end
 S(i, j) = (harmonicPeaks(i)−downshift) + (-27 + ...
 0.37*max((harmonicPeaks(i)−downshift)−40, 0)*thetadz)*abs(dz);
end

% Draw the masking curve
masking = S(1, :);
for i=1:1:length(b)
 for j=2:1:length(harmonicLocsBark)
  if(S(j,i) > masking(i))
   masking(i) = S(j, i);
  end
 end

% Find the maximum of Hearing Threshold Curve and the Masking Curve
curve = hearingThres;
for i=1:1:length(b)
 for j=1:1:length(harmonicLocsBark)
  if(S(j,i) > curve(i))
   curve(i) = S(j, i);
  end
 end

% Plot the curve, harmonic peaks, and non−harmonic peaks
if plot
 figure()
 hold on
 plot(b, curve, 'b−−')
 hold on
 plot(b, hearingThres, 'k−−')
 hold on
 plot(b, masking, 'g−−')
 hold on
 stem(harmonicLocsBark, harmonicPeaks,'BaseValue',−20, 'Color', 'g')
 hold on
 stem(nonHarmonicLocsBark, nonHarmonicPeaks,'BaseValue',−20, ... 'Color', 'r', 'Marker', 'x')
title('Bosi PAM Spectrum')
xlabel('Frequency (Bark)')
function [harmonicDecaydB, aliasingdB] = AWeightedPAM ...
\( \text{harmonicLocsBLEP}, \text{harmonicPeaksBLEP}, \text{nonHarmonicLocsBLEP}, \ldots \)
\( \text{nonHarmonicPeaksBLEP}, \text{harmonicLocsTrivial}, \text{harmonicPeaksTrivial}, \ldots \)
\( \text{nonHarmonicLocsTrivial}, \text{nonHarmonicPeaksTrivial} \)
\% AWEIGHTEDPAM Applies the a simple A-weighted Psychoacoustic Model (PAM)
\% to an audio signal and compares the harmonic decay and
\% aliasing levels to those of the trivial version.
\% Copyright 2015.
\% Francisco J. Valencia
\% Music Engineering and Technology
\% University of Miami
\% INPUTS

ylabel('Magnitude (dB)')
if(isempty(nonHarmonicLocs))
    legend('PAM Curve', 'Hearing Threshold Curve', ...
            'Masking Curve', 'Harmonics')
else
    legend('PAM Curve', 'Hearing Threshold Curve', ...
            'Masking Curve', 'Harmonics', 'Non-Harmonics')
end
axis tight
ylim([-20 100]);
end

% Check for first non-harmonic peak above the curve
nonHarmonicsAbove = nonHarmonicLocs;
for i=1:length(nonHarmonicLocs)
    % Find closest b value for the location
    absBark = abs(b - nonHarmonicLocsBark(i));
    [~, index] = min(absBark);
    if (nonHarmonicPeaks(i) < curve(index))
        nonHarmonicsAbove(i) = 0.0; %Hz
    end
end
nonHarmonicsAbove(nonHarmonicsAbove == 0.0) = [];

% Check for first harmonic peak below the curve
harmonicsBelow = harmonicLocs;
for i=1:length(harmonicLocs)
    % Find closest b value for the location
    absBark = abs(b - harmonicLocsBark(i));
    [~, index] = min(absBark);
    if (harmonicPeaks(i) > curve(index))
        harmonicsBelow(i) = 0.0; %Hz
    end
end
harmonicsBelow(harmonicsBelow == 0.0) = [];

% Check if there is any aliasing above the curve
if isempty(nonHarmonicsAbove)
    aliasPresent = false;
else
    aliasPresent = true;
end
end

end
% harmonicLocsBLEP (1xn double array): array with the frequencies of the
% harmonic peaks of a BLEP signal.
% harmonicPeaksBLEP (1xn double array): array with the magnitudes of the
% harmonic peaks of a BLEP signal.
% nonHarmonicLocsBLEP (1xm double array): array with the frequencies of
% the non harmonic peaks of a
% BLEP signal.
% nonHarmonicPeaksBLEP (1xm double array): array with the magnitudes of
% the non harmonic peaks of a
% BLEP signal.

% OUTPUTS
harmonicDecaydB (double): overall difference of harmonic decay between
the BLEP and the trivial waveforms (dB).
aliasingdB (double): overall difference of aliasing between the BLEP
and the trivial waveforms (dB).

% Find total sum of Aliasing
nonHarmonicPeaksTrivialWeigthed = nonHarmonicPeaksTrivial;
for i=1:1:length(nonHarmonicPeaksTrivial)
    AWeigthedFactor = calcAWeigthFactor(nonHarmonicLocsTrivial(i));
    nonHarmonicPeaksTrivialWeigthed(i) = ...
    nonHarmonicPeaksTrivial(i)*AWeigthedFactor;
end
sumNonHarmonicPeaksTrivial= sum(nonHarmonicPeaksTrivialWeigthed);
if(isempty(nonHarmonicPeaksBLEP))
    sumNonHarmonicPeaks = 0;
else
    nonHarmonicPeaksBlepWeigthed = nonHarmonicPeaksBLEP;
    for i=1:1:length(nonHarmonicPeaksBLEP)
        AWeigthedFactor = calcAWeigthFactor(nonHarmonicLocsBLEP(i));
        nonHarmonicPeaksBlepWeigthed(i) = ...
        nonHarmonicPeaksBLEP(i)*AWeigthedFactor;
    end
    sumNonHarmonicPeaks = sum(nonHarmonicPeaksBlepWeigthed);
end
% Compare total aliasing to the trivial sawtooth
aliasingdB = 20*log10(sumNonHarmonicPeaks) ...
20*log10(sumNonHarmonicPeaksTrivial);

% Find total sum of Harmonic Decay
harmonicPeaksTrivialWeigthed = harmonicPeaksTrivial;
for i=1:1:length(harmonicPeaksTrivial)
    AWeigthedFactor = calcAWeigthFactor(harmonicLocsTrivial(i));
    harmonicPeaksTrivialWeigthed(i) = ...
    harmonicPeaksTrivial(i)*AWeigthedFactor;
end
sumHarmonicPeaksTrivial= sum(harmonicPeaksTrivialWeigthed);
harmonicPeaksBlepWeigthed = harmonicPeaksBLEP;
for i=1:1:length(harmonicPeaksBLEP)
    AWeigthedFactor = calcAWeigthFactor(harmonicLocsBLEP(i));
    harmonicPeaksBlepWeigthed(i) = ...
    harmonicPeaksBLEP(i)*AWeigthedFactor;
end
sumHarmonicPeaks = sum(harmonicPeaksBlepWeigthed);
% Compare total harmonic decay to the trivial sawtooth
harmonicDecaydB = 20*log10(sumHarmonicPeaks) ...
20*log10(sumHarmonicPeaksTrivial);
Appendix C - BLEP Implementation

function [window] = createWindow (N, a, b, plot)
% CREATEWINDOW Creates a window based on the 35 parameters of the Genetic
% Algorithm (GA).
% 
% [window] = createWindow (N, a, b, plot)
% Copyright 2015.
% Francisco J. Valencia
% Music Engineering and Technology
% University of Miami
% 
% INPUTS
% N (double): size of BLEP table.
% a (1x17 double array): multiplicative coefficients from GA.
% b (1x17 double array): power coefficients from GA.
% plot (bool): true for drawing plots.
% 
% OUTPUTS
% window (1xN double array): window created.

originalWindows = {@bartlett, @barthannwin, @blackman, @blackmannharris, ...
@bohmanwin, @chebwin, @flattopwin, @gausswin, @hamming, @hann, ...
@kaiser, @nuttallwin, @parzenwin, @rectwin, @taylorwin, ...
@triang, @tukeywin};
nOriginalWindows = length(originalWindows);
errorVector = zeros(1, nOriginalWindows);
if a == errorVector
  error('Window is vector 0')
end
windows = zeros(nOriginalWindows, N);
% Original Windows
for i=1:1:nOriginalWindows
  windows (i, :) = ((originalWindows{i}(N)).ˆb(i))*(a(i));
end
window = sum(windows);
%Normalize
maxWin = max(window);
window = window/maxWin;
if plot
  samples = 1:1:N;
  figure()
  plot(samples,window)
  title ('Window for N = ' num2str(N) ' samples')
xlabel 'Samples'
ylabel 'Amplitude'
axis tight
end
end

function [bleptable] = doBLEPTable(fs, input, window, N, n, plot)
% DOBLEPTABLE Creates a BLEP table for an specific input and window
% 
% [bleptable] = doBLEPTable(fs, input, window, n, m)
% Copyright 2015.
% Francisco J. Valencia
% Music Engineering and Technology
% University of Miami
% 
% INPUTS
% fs (double): sampling rate of input audio signal.
% input (handle): handle of the input function (i.e. '@sinc').
% window (1xN double array): window in an array.
% N (double): size of BLEP table.
% n (double): number of zero crossings on each side of function.
% plot (bool): true for drawing plots.

% OUTPUTS
% bleptable (1xn double array): BLEP table.

Ts = 1/fs;
T = ((N-1)*Ts)/(2*n);
f = 1/T;
t = −n*T:Ts:n*T;

% 1. DEFINE INPUT
impulse = input(fc*t);
if plot
    figure()
    plot(t/Ts,impulse)
    title(['Input Function with N = ' num2str(N) ' samples'])
    xlabel('Samples');
    ylabel('Amplitude');
    axis tight

    figure()
    plot(t/Ts,window)
    title(['Window with N = ' num2str(N) ' samples'])
    xlabel('Samples');
    ylabel('Amplitude');
    axis tight
end

% 2. APPLY WINDOW
windowed = impulse.*window;
if plot
    figure()
    plot(t/Ts,windowed)
    title(['Windowed Input Function with N = ' num2str(N) ' samples'])
    xlabel('Samples');
    ylabel('Amplitude');
    axis tight
end

% 3. INTEGRATE AND CONVERT TO BIPOLAR
sum = 0;
sum2 = 0;
integral = zeros(1,length(t));
for i = 1:length(t)
    sum = sum+windowed(i);
    integral(i)=sum;
    sum2 = sum2 + integral(i);
end
bipolar = integral/sum;
bipolar = bipolar.*2−1;
if plot
    figure()
    plot(t/Ts,bipolar)
    title(['Bipolar Integrated Input Function with N = ' num2str(N) ' samples'])
    xlabel('Samples');
    ylabel('Amplitude');
    axis tight
end
% 4. SUBTRACT STEP FUNCTION
step = zeros(1,length(t));
for i = 1:length(t)
    if t(i)<0
        step(i) = -1;
    else
        step(i) = 1;
    end
end
residual = bipolar - step;
invresidual = residual.*-1;
bleptable = residual;
if plot
    figure()
    plot(t/Ts, residual)
    title(['BLEP Residual with N = ' num2str(N) ' Samples'])
    xlabel('Samples');
    ylabel('Amplitude');
    axis tight
    figure()
    plot(t/Ts, invresidual)
    title('BLEP Residual Inverted')
    title(['BLEP Residual Inverted with N = ' num2str(N) ' Samples'])
    xlabel('Samples');
    ylabel('Amplitude');
    axis tight
end
end

function [blepWaveform, trivialWaveform, timeArray] = doBLEPWaveform ...
% DOBLEPWAVEFORM Creates a BLEP waveform for an specific BLEP table base
% on the method on Pirkle, W. (2014). Designing Software
% Synthesizer Plug-Ins in C++: For RackAFX, VST3, and
% Audio Units. Focal Press.
% [blepWaveform, trivialWaveform, timeArray] = doBLEPWaveform ...
% (fc, fs, bleptable, m, NDFT, plot)
% Copyright 2015.
% Francisco J. Valencia
% Music Engineering and Technology
% University of Miami
% INPUTS
% fc (double): fundamental frequency of waveform.
% fs (double): sampling rate of input audio signal.
% bleptable (1xN double array): BLEP table.
% m (double): number of correction points on each side.
% NDFT (double): size of DFT.
% plot (bool): true for drawing plots.
% OUTPUTS
% blepWaveform (1xn double array): BLEP corrected waveform.
% trivialWaveform (1xn double array): trivial waveform.
% timeArray (1xn double array): time array for the waveforms.
% Define time array
Ts = 1/fs;
timeArray = 0:Ts:(NDFT/fs)-Ts;
% Create sawtooth
modulo = 0.5;
inc = fc/fs;
sampleschanged = 0;
center = length(bleptable)/2.0 - 1.0;
trivialWaveform = zeros(1, length(timeArray));
blepWaveform = zeros(1, length(timeArray));
for i = 1:length(timeArray)
    if (modulo >= 1.0)
        modulo = modulo - 1.0;
    end
trivialWaveform(i) = 2.0*modulo - 1.0;
blepWaveform(i) = trivialWaveform(i);
% Apply BLEP % Left side of discontinuity -1 < d < 0
for j = 1:m
    if (modulo > (1.0 - j*inc))
        d = (modulo - 1.0)/(m*inc);
        index = (1.0 + d)*center;
        index = floor(index);
        blepvalue = bleptable(index + 1.0);
        blepvalue = blepvalue - 1.0;
        blepWaveform(i) = trivialWaveform(i) + blepvalue;
        sampleschanged = sampleschanged+1;
        break
    end
end % Right side of discontinuity 0 <= d < 1
for j = 1:m
    if (modulo < j*inc)
        d = modulo/(m*inc);
        index = d*center + (center+1.0);
        index = floor(index);
        blepvalue = bleptable(index + 1.0);
        blepvalue = blepvalue - 1.0;
        blepWaveform(i) = trivialWaveform(i) + blepvalue;
        sampleschanged = sampleschanged+1;
        break
    end
endmodulo = modulo + inc;
if plot
cyclesToShow = 3;
samplesToShow = ceil(fs*cyclesToShow/fc);
figure()
pot(timeArray(1:samplesToShow)/Ts, trivialWaveform(1:samplesToShow))
title (['Sawtooth Trivial - Time Domain for fc = ' num2str(fc) ' Hz'])
xlabel 'Samples'
ylabel 'Amplitude'
axis tight
figure()
pot(timeArray(1:samplesToShow)/Ts, blep(1:samplesToShow))
title (['Sawtooth BLEP - Time Domain for fc = ' num2str(fc) ' Hz'])
xlabel 'Samples'
ylabel 'Amplitude'
axis tight
figure()
plot(timeArray(1:samplesToShow)/Ts, trivialWaveform(1:samplesToShow))
hold on
plot(timeArray(1:samplesToShow)/Ts, blep(1:samplesToShow))
title (['Time Domain Comparison for fc = ' num2str(fc) ' Hz'])
legend('Sawtooth Trivial', 'Sawtooth BLEP')
function [additiveSynthWaveform] = createAdditiveVersion (fc, fs, ...
    input, NDFT, peaksLocs, peaksLocsIndex, harmonicLocs, plot)
    % CREATEADDITIVEVERSION Creates an additive version of a BLEP waveform
    % which only include the harmonic components, but
    % keeping the same harmonic decay.
    %
    % [additiveSynthWaveform, DFT] = createAdditiveVersion (fc, fs, ...
    % input, NDFT, peaksLocs, peaksLocsIndex, harmonicLocs, plot)
    % Copyright 2015.
    % Francisco J. Valencia
    % Music Engineering and Technology
    % University of Miami
    %
    % INPUTS
    % fc (double): fundamental frequency of waveform.
    % fs (double): sampling rate of input audio signal.
    % input (1xn double array): BLEP corrected waveform.
    % NDFT (double): size of DFT.
    % peaksLocs (1xn double array): array with the frequencies of the peaks
    % of the DFT of the input.
    % peaksLocsIndex (1xn double array): array with the index (inside
    % frequencies array) of the
    % peaks of the DFT of the input.
    % harmonicLocs (1xn double array): array with the frequencies of the
    % harmonic peaks of the input.
    % plot (bool): true for drawing plots.
    %
    % OUTPUTS
    % additiveSynthWaveform (1xn double array): Additive synthesis waveform.
    %
    L = length(input);
    Y = fft(input, NDFT)/L;

    fHarmonics = zeros(1, length(harmonicLocs));
    aHarmonics = zeros(1, length(harmonicLocs));
    for i=1:length(harmonicLocs)
        for j=1:length(peaksLocs)
            if (harmonicLocs(i) == peaksLocs(j))
                fHarmonics(i) = harmonicLocs(i);
                aHarmonics(i) = ...
                2*abs(Y(peaksLocsIndex(j)))*exp(1i*angle(Y(peaksLocsIndex(j))));
                break
            end
            end
    end

    Ts = 1/fs;
    t = 0:Tsa:(nFFT/fs)-Ts;

    % Additive Synthesis
    additiveSynthWaveform = zeros(size(t));
    for k=1:length(fHarmonics)
        additiveSynthWaveform = additiveSynthWaveform + ...
            aHarmonics(k)*exp(1i*2*pi*fHarmonics(k)*t);
    end
    additiveSynthWaveform = real(additiveSynthWaveform);
if plot
    cyclesToShow = 3;
samplesToShow = ceil(fs*cyclesToShow/fc);
figure()
plot(t(1:samplesToShow)/Ts, input(1:samplesToShow))
hold on
plot(t(1:samplesToShow)/Ts, additive(1:samplesToShow))
xlabel 'Samples'
ylabel 'Amplitude'
title ('Additive Synthesis for fc = ' num2str(fc) ' Hz')
legend ('Original BLEP', 'Ideal BL')
axis tight
end
end
Appendix D - Genetic Algorithm (GA)

```matlab
function [output, eval] = runGAPhaseIII (NDFT, fs, N)
% RUNGAPHASEIII Runs the GA Phase III and finds an optimal solution for a
% specific table size N.
% 
% [output, eval] = runGAPhaseIII (NDFT, fs, N)
% Copyright 2015.
% Francisco J. Valencia
% Music Engineering and Technology
% University of Miami
% 
% INPUTS
% NDFT (double): size of DFT.
% fs (double): sampling rate of input audio signal.
% N (double): size of BLEP table.
% 
% OUTPUTS
% output (1x36 double array): (1:17) multiplicative coefficients from GA.
% (18:34) power coefficients from GA.
% (35) number of zero crossings at each side
% of sinc function (n).
% (36) number of BLEP cocrrection points at each side (m).
% eval (double): Bosi PAM value for the output.
% 
parpool % Parallel computing
tic;

% Boundaries
lb = [-100 -100 -100 -100 -100 -100 -100 -100 -100 -100 -100 -100 -100 ... 
    -100 -100 -100 -100 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1];
ub = [100 100 100 100 100 100 100 100 100 100 100 100 100 ... 
    100 30 30 30 30 30 30 30 30 30 30 30 30 30 30 8 5];
intCon = 1:36;

% GA
options = gaoptimset('FitnessLimit',0,'StallGen',125,...
    'PlotFcns',@gaplotbestf,...
    'UseParallel', true, 'Vectorized', 'off');

[output,eval] = ga(@(x)fitnessFunction(x, fs, NDFT, ... 
    N),36,[],[],[],lb,ub,[],intCon,options);
toc;
delete(gcp)
end
```

```matlab
function [output] = fitnessFunction (x, fs, NDFT, N)
% FITNESSFUNCTION Evaluates an output of the GA based on an evaluation
% function. It verifies all the MIDI notes from 84 to 108.
% 
% [output] = fitnessFunction (x, fs, NDFT, N)
% Copyright 2015.
% Francisco J. Valencia
% Music Engineering and Technology
% University of Miami
% 
% INPUTS
% x (1x36 double array): GA output.
% 
% output (1:17) multiplicative coefficients from GA.
% (18:34) power coefficients from GA.
% (35) number of zero crossings at each side
% of sinc function (n).
```
% (36) number of BLEP correction points at each side (m).

% fs (double): sampling rate of input audio signal.
% NDF (double): size of DFT.

% OUTPUTS
% output (double): Bosi PAM value for the output.

MIDITable = 84:1:108;
freqTable = MIDI2freq(MIDITable);

% Establish the fundamental frequency array (fixed to bins)

factorsFundamental = 1:1:floor(4200/(fs/NDFT)); %MIDI 127
fundamentalsFixedBins = (fs/NDFT)*factorsFundamental;
freqTableFixedBins = freqTable;

for i=1:1:length(freqTable)
    [~, indexFundamental] = min(abs(fundamentalsFixedBins - freqTable(i)));
    freqTableFixedBins(i) = fundamentalsFixedBins(indexFundamental);
end

fc = freqTableFixedBins;

%34 Variables doubles
a = x(1:17);
b = x(18:34);
nZeroCrossingEachSide = 1:1:16;
n = nZeroCrossingEachSide(x(35));
mBlepEachSide = 1:1:16;
m = mBlepEachSide(x(36));
window = createWindow (N, a, b, false);

% Harmonics below Bosi curve for original trivial MIDI 84 to 108
trivialHarmBelow = [7,7,6,6,5,5,4,4,3,3,2,2,2,2,2,1];
output = 0;

for i=1:1:length(fc)
    value = evaluationFunction (fc(i), fs, N, n, m, NDFT, window);
    output = output + value - trivialHarmBelow(i);
end

end

function [output] = evaluationFunction (fc, fs, N, n, m, NDFT, window)
% EVALUATIONFUNCTION Evaluates an output of the GA based on the Bosi PAM.
% [output] = evaluationFunction (fc, fs, n, m, m2, nFFT, window)
% Copyright 2015.
% Francisco J. Valencia
% Music Engineering and Technology
% University of Miami

% INPUTS
% fc (double): fundamental frequency of waveform.
% fs (double): sampling rate of input audio signal.
% N (double): BLEP table size.
% n (double): number of zero crossing on each side of the sinc function.
% m (double): number of correction points on each side.
% NDFT (double): size of DFT.
% window (1xN double array): window array.

% OUTPUTS
% output (double): Bosi PAM value for the output.

% Create the BLEP tables for window
bleptable = doBLEPTable(fs, @sinc, window, N, n, false);
% Create the Trivial Sawtooth and the BLEP–Sawtooths
[blep, ~, ~] = doBLEPWaveform(fc, fs, bleptable, m, NDFT, false);
% Fix Clipping and extend to [-1, 1]
blep = fixClipping (blep);
% Obtain DFT of BLEPS
[~, ~, fftRawNormBlep, freqArray] = calculateDFT(fs, blep, NDFT, false);
% Get the harmonic and non–harmonic locations and amplitudes
[peaksBlepRaw, locsBlepRaw] = findpeaks(fftRawNormBlep, freqArray);
[harmonicPeaksBlepRaw, harmonicLocsBlepRaw, nonHarmonicPeaksBlepRaw, ...
    nonHarmonicLocsBlepRaw] = findHarmonicPeaks (peaksBlepRaw, ...
    locsBlepRaw, fs, fc);
% Apply Bosi PAM
[~, nonHarmonicsAboveBlepRaw, harmonicsBelowBlepRaw] = BosiPAM ...
    {harmonicPeaksBlepRaw, harmonicLocsBlepRaw, ...
    nonHarmonicPeaksBlepRaw, nonHarmonicLocsBlepRaw, 'linear', false);
output = length(nonHarmonicsAboveBlepRaw) + length(harmonicsBelowBlepRaw);
end