

# Design of Parallel Second-order Filters for Instrument Body Modeling

MUMT 618

Patrick Cowden

December 4 , 2020

# 1 Introduction

Because of the desire to represent acoustical phenomena in the digital domain, many researches have focused efforts in general computational acoustic modeling, like the various algorithms and methods that are described in [1] by Julius Smith. These explorations consist of general instrument modeling as well as modeling of audio effects. Most of these modeling techniques utilize complex filters. Within this area of research, Instrument body modeling is an area that has been deeply explored. It is a common area of exploration within musical applications since there is a desire to have instruments and their bodies represented in the digital domain for easy use and manipulation.

Balazs Bank has completed various works in Instrument body modeling as described in [2] and [3]. Most of Bank's efforts have been focused on using second order parallel filters for creating computational acoustic models of instrument bodies. After developing an algorithm for [2], Bank collaborate with other researchers and developed a method for modeling of a guitar loudspeaker cabinet. This research composed of creating the model using the parallel second order filters developed in [2] along with a non-linear component for representing the distortions within the guitar loudspeaker.

For this paper, I focused on implementing the parallel second order filters discussed in [2]. I had originally proposed to additionally implement the guitar speaker model discussed in [3], but since the measurement data required for the model was not publicly available I could not implement or test a model for the guitar speaker cabinet.

This paper contains review and discussion of parallel second order filters and there uses for instrument body modeling. Within the paper, the algorithms and methods/applications are discussed. The design, implementations, and analyses were all completed in MATLAB. A review of the methods, difficulties/failures with implementations, and results of the implementations are also discussed.

# 2 Parallel Filter formulation

The algorithm formulation starts with the defined equation for the parallel second order filter:

$$H(z^{-1}) = \sum_{k=1}^K c_k \frac{1}{1 - p_k z^{-1}} + \sum_{m=0}^M b_m z^{-n} \quad (1)$$

Where K is the number of second order sections, and  $c_k$  and  $p_k$  are the filter coefficients that need to be determined. The second part of the equation is a FIR filter part that is optional for the parallel filter that helps improve the resulting efficacy for modeling of the filter. It consists of a simple FIR delay line with length M.

The sequence for the filter formulation first starts with the impulse response of the instrument body that will be modeled. The impulse response is warped

to an all-pass filter and the resulting filter poles are computed. To warp the input impulse response to the all-pass filter, methods discussed in [4] and [5] are used. After warping the input impulse response, the warped filter poles are found using the prony method.

Once the warped filter poles are found,  $\tilde{p}_k$ , they are converted back to a corresponding linear frequency scale using the following expression:

$$p_k = \frac{\tilde{p}_k + \lambda}{1 + \tilde{p}_k} \quad (2)$$

Where  $\lambda$  is the warping parameter that was used for the warped filter formulation.

Since the prony method is used and the poles are converted to a linear scale, the parallel filter can be split into two separate filter sections. We can compute the resulting impulse response of the model parallel filter by splitting up the undetermined filter zeros and the determined poles into a multiplication of the zero-pole and zero-zero filters. Specifically, this can be represented as:

$$h(n) = \sum_{k=1}^K c_{k,0} u_k(n) + c_{k,1} u_k(n-1) \quad (3)$$

Where  $u_k$  is the impulse response of the zero-pole filter. This can then be represented in matrix form as

$$h = Mp \quad (4)$$

Where  $p$  is the computed poles and  $M$  is a modeling matrix. Using equation 4, the least mean squared error is then found to determine the optimal poles for the parallel filter. This is computed using the expression:

$$p_{opt} = (M^H M)^{-1} M^H h_t \quad (5)$$

Where  $M^H$  is the conjugate transpose of  $M$ .

## 2.1 Implementation

I had extreme difficulty in constructing the algorithm for computing the second order parallel filter for an instrument body modeling. Both of the discussions in [2] and [3] do not go into extreme detail on how to develop the modeling Matrix  $M$ . Additionally there is not a detailed explanation of how to use the optimized poles in order for creating the parallel filter. I found MATLAB examples outlined in supplementary material for [3] which I used as a reference for my implementation development. These examples slightly helped me understand how to develop the algorithm on my own, but the main Modeling matrix multiplication was still very difficult for me to understand. Also, both [2] and [3] do not discuss choosing the appropriate warping parameter impulse all pass filter warping. In examples shown in [3], Bank used a value of 0.65. This seems to be an optimal values so for all tests this value was chosen. Due to the time constraint allotted for the project, I used the helper MATLAB functions for

all of the matrix mathematics that were required for the second order filter formulation. The resulting implementation allows for any impulse response to be inputted into the algorithm, with user defined filter order, resultant impulse response length, and number of FIR sections. The system generates the second order parallel filter coefficients which can then be used for filtering any input audio signal. To test the results, a MIDI piano recording was applied to the resultant filters.

## 2.2 Results

To test the results of the implementation, Instrument body impulse responses were found online in [7]. First, the sample "15: Transition from metal to wood" was used for testing the system. This sample starts with the impulse response of a metallic instrument body. The transition and wood impulse response of the sample was not used. First, the resultant impulse response length was chosen to be 10000 samples long(at 44100 Hz), the warping parameter was chosen to be 0.65, the filter order was chosen to be 200 and the number of FIR sections was chosen to be 1. The FIR component was set to 1 to allow for a simple feed-forward FIR path. After generating the filter coefficients the parallel filter was created in MATLAB and a MIDI piano recording was passed through the filter. The resulting audio sample sounded more metallic than the original MIDI piano sample.

The resulting frequency domain response and time domain response of the input impulse response and model response are shown in figures 1 and 2. In both of these figures the black trace is the original input response and the red trace is the modeled filter response.

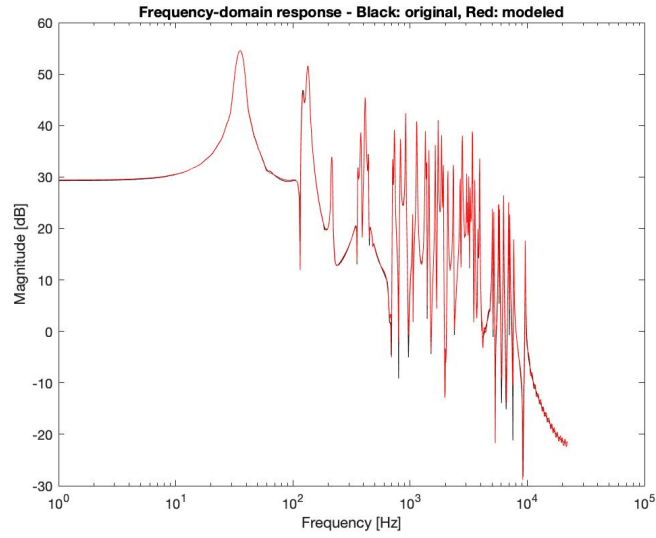


Figure 1: Instrument Body Modeling Frequency Domain Comparison With Filter Order 200

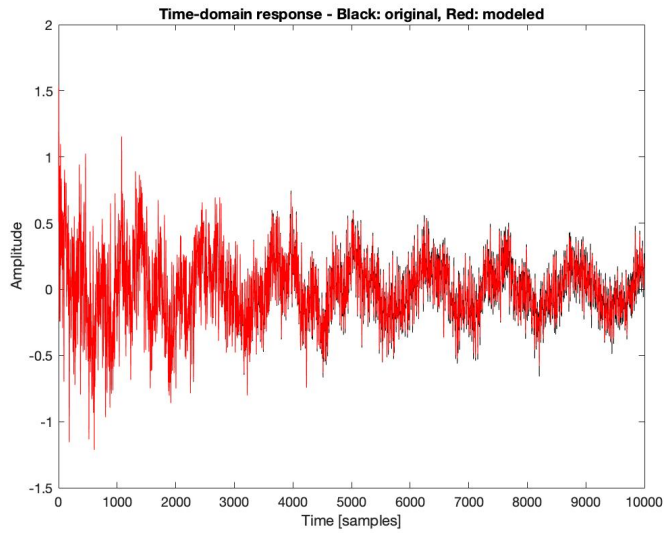


Figure 2: Instrument Body Modeling Time Domain Comparison With Filter Order 200

As a second test, the same input sample was used with a lower filter order. The parameters were the same but the filter order was changed to 50. The

FIR component was set to 1 to allow for a simple feed-forward FIR path. The resulting frequency domain response and time domain response of the input impulse response and model response are shown in figures 2 and 3. In both of these figures the black trace is the original input response and the red trace is the modeled filter response.

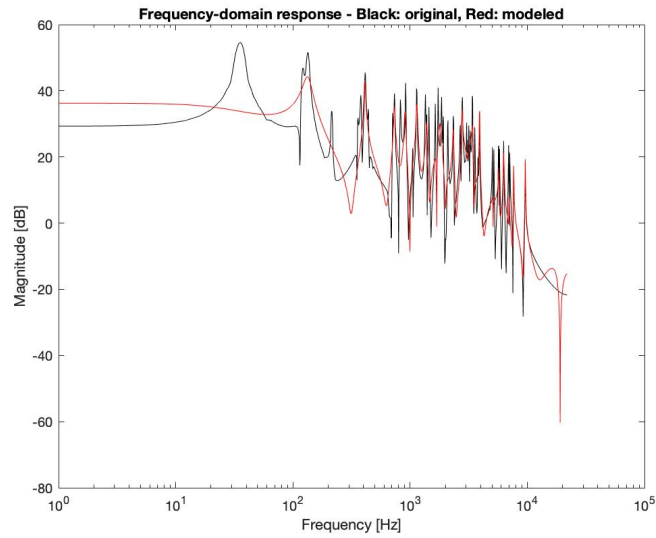


Figure 3: Instrument Body Modeling Frequency Domain Comparison With Filter Order 50

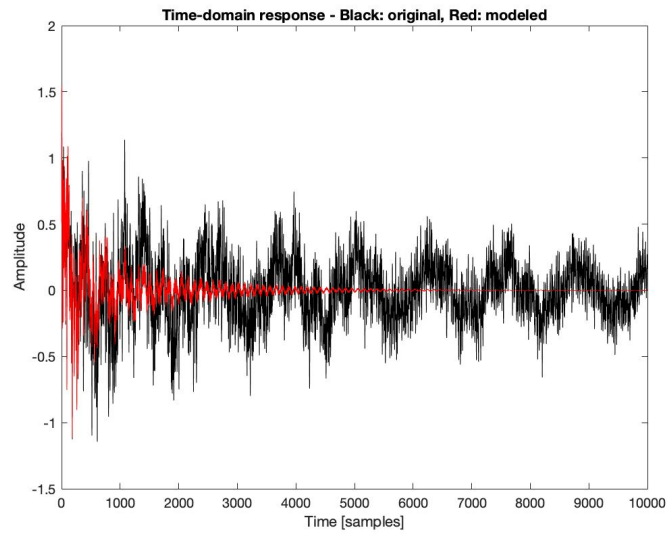


Figure 4: Instrument Body Modeling Time Domain Comparison With Filter Order 50

As a final test, the same input sample was used with the lower filter order. The FIR delay line length was set to 10 to test the effect of the FIR part. section The resulting frequency domain response and time domain response of the input impulse response and model response are shown in figures 2 and 3. In both of theses figures the black trace is the original input response and the red trace is the modeled filter response.

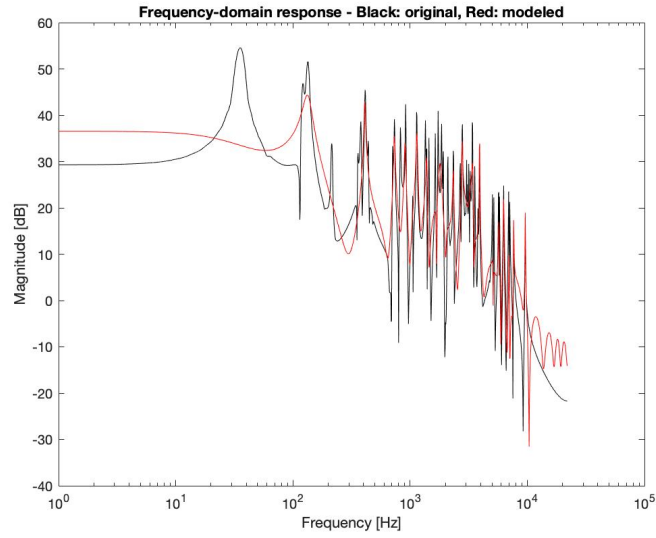


Figure 5: Instrument Body Modeling Frequency Domain Comparison With Filter Order 50 10 Delay Length FIR

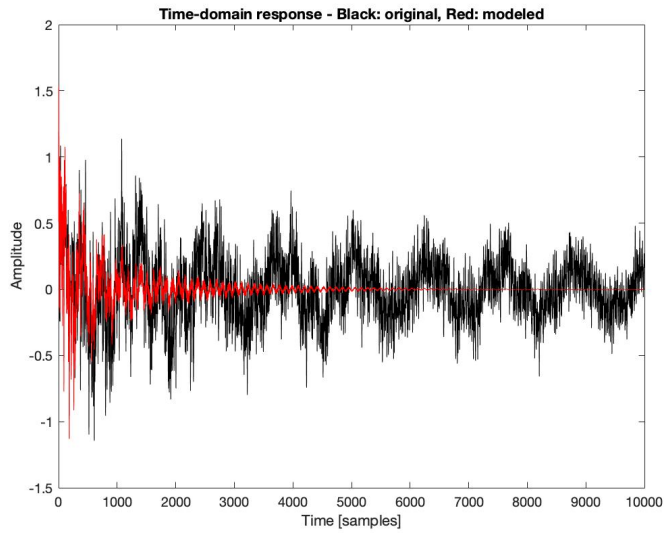


Figure 6: Instrument Body Modeling Time Domain Comparison With Filter Order 50 and 10 Delay Length FIR



## 2.3 Discussion

The results of the above tests show that the second order parallel filter for instrument body modeling is pretty successful but high filter orders are required. If the filter order is not high enough than the resulting filter model frequency and time domain response do not accurately match that of the instrument body.

Based on the results of the tests, the resolution improvement can be seen when the FIR component is included. Increasing the FIR component delay line allows for slight improvement of the frequency and time domain resolution for the models. It is not as dramatic of an increase in performance as increasing the order.

The main issue with increasing the filter order is that it is computationally expensive. Increasing the order causes the computation time to increase. Since the main use for this application is not in real-time this is not an issue.

## 3 Conclusion

Overall, the design and implementation of the second order parallel filter for instrument body modeling was successfully completed in MATLAB. The results from the implementation show that an instrument body can be successfully modeled with a filter order of 200 or more. This algorithm allows for an instrument body impulse response to be inputted to the system and the resulting parallel filter coefficients are calculated. These coefficients can then be used for apply the body model to any audio signal.

Difficulties arose in interpreting and implementing the Matrix mathematics required for computing the parallel filter coefficients that are discussed in [2] and [3]. After extensive research and testing a functional algorithm was unable to be developed on my own. I had to use helper methods provided by Bank in [3].

There are plenty of opportunities to continue researching this topic. I hope to spend further time on better understanding how the algorithm works, specifically the matrix multiplication. Once I have a better understanding of the Matrix multiplication and how it is used for the coefficient generation I believe that I could generate my own version of this model completely from scratch.

## 4 References

- [1] J. O. Smith, *Physical audio signal processing: for virtual musical instruments and audio effects*. Lexington: W3K Publishing, 2011.
- [2] B. Bank, M. Karjalainen, and D. T. Yeh, "Nonlinear Modeling of A Guitar Loudspeaker Cabinet," in *Proc. Of Conf. on Digital Audio Effects (DAFx-08)*, Espoo, Finland, sept 2008.
- [3] B. Bank, "Direct design of parallel second-order filters for instrument body

modeling,” in Int. Comp. Music Conf., Copenhagen, Denmark, Aug. 2007, pp. 458–465, URL: <http://research.spa.aalto.fi/publications/papers/icmc07-parfilt/>

[4] A. Härmä, M. Karjalainen, L. Savioja, V. Välimäki, U. K. Laine, and J. Huopaniemi, “Frequency-warped signal processing for audio applications,” in AES 108th Convention, preprint 5171 (T-5), (Paris, France), Audio Engineering Society, February 2000.

[5] A. Härmä, M. Karjalainen, L. Savioja, V. Välimäki, U. K. Laine, and J. Huopaniemi, “Frequency-warped signal processing for audio applications,” J. Aud. Eng. Soc., November 2000.

[6] A. Härmä, M. Karjalainen, WarpTB, ”Matlab Toolbox for Warped DSP”, Helsinki University of Technology, Department of Signal Processing and Acoustics (former Laboratory of Acoustics and Audio Signal Processing), Espoo, Finland.

[7] M. Aramaki, R. Kronland-Martinet, “Analysis-Synthesis of Impact Sounds by Real-Time Dynamic Filtering,” in IEEE Transactions on Audio, Speech and Language Processing, Volume 14, number 2, pages 695-705, March 2006, URL: [http://www.lma.cnrs-mrs.fr/~kronland/IEEE\\_SAP/sounds.html](http://www.lma.cnrs-mrs.fr/~kronland/IEEE_SAP/sounds.html).