Chapter 4

Partial Tracking

In the previous chapter, we saw how to estimate parameters of sinusoids with polynomial phase. While theoretically applicable to signals of arbitrary length, for reasons of flexibility and efficiency, we usually estimate the local parameters of the signal under a low-order model and connect multiple estimations to form a partial. We will call these local estimations "analysis points" or "parameter sets".

This chapter presents an interpretation of the *peak matching* procedure of McAulay and Quatieri [36], a classical approach to discovering partials. Our interpretation allows for the specification of an arbitrary cost function measuring the plausibility that a set of analysis points form the path of a partial. With this path interpretation, we were able to design a technique that finds the optimal set of paths under a constraint on the number of paths. The chapter concludes with an example of partial tracking on a synthetic signal.

Typically the DTSTFT is computed for a block of contiguous samples, called a *frame* and these frames are computed every H samples, H being the *hop-size*. We will denote the M sets of parameters at local maxima in frame h as $\theta_0^h, \ldots, \theta_{M-1}^h$ and the N in frame $h+1$ as $\theta_0^{h+1}, \ldots, \theta_{N-1}^{h+1}$ where h and $h+1$ refer to adjacent frames. We are interested in paths that extend across K frames where each path touches only one parameter set and each parameter set is either exclusive to a single path or is not on a path.

4.1 A greedy method

In this section, we present the McAulay-Quatieri method of peak matching. It is conceptually simple and a set of short paths can be computed quickly, but it can be sensitive to

spurious peaks and is optimal only in the sense that the set of paths computed contains the best path possible — the quality of the other paths may be compromised under this criterion.

In [36, p. 748] the peak matching algorithm is described in a number of steps; we summarize them here in a way comparable with the linear programming formulation to be presented in the sequel. In that paper, the parameters of adjacent frames h and $h+1$ are the instantaneous amplitude, phase, and frequency and are indexed by frequency as $\omega_0^h, \ldots, \omega_{M-1}^h$ and $\omega_0^{h+1}, \ldots, \omega_{N-1}^{h+1}$ but we will allow for arbitrary parameter sets. Define a distance function $\mathcal{D}(\theta_i, \theta_j)$ that computes the similarity between $K = 2$ sets of parameters. We will now consider a method that finds L pairs of parameters that are closest.

We compute the cost matrix $\boldsymbol{C}$

$$\boldsymbol{C} = \theta^h \otimes_{\mathcal{D}} \theta^{h+1}$$

so that the ith row and jth column contain $C_{i,j} = \mathcal{D}\left(\theta_i^h, \theta_j^{h+1}\right)$. For each $l \in [0 \ldots L-1]$, find the indices i_l and j_l corresponding to the shortest distance, then remove the i_lth row and j_lth column from consideration and continue until L pairs have been determined or the distances exceed some threshold Δ. This is summarized in Algorithms 1

Algorithm 1: A generalized McAulay-Quatieri peak-matching algorithm.

Input: the cost matrix $\boldsymbol{C}$

Output: L pairs of indices Γ_i and Γ_j

$\Gamma_i \leftarrow \varnothing$;

$\Gamma_j \leftarrow \varnothing$;

for $l \leftarrow 0$ *to* $L-1$ **do**

  $i_l, j_l = \underset{i \in [0,\ldots,M-1] \setminus \Gamma_i, j \in [0,\ldots,M-1] \setminus \Gamma_j}{\arg\min} C_{i,j}$;

  if $C_{i_l,j_l} > \Delta$ **then**

    return Γ_i, Γ_j

  end

  $\Gamma_i \leftarrow \Gamma i \cup i_l$;

  $\Gamma_j \leftarrow \Gamma i \cup j_l$;

end

return Γ_i, Γ_j

This is a greedy algorithm because on every iteration the smallest cost is identified and its indices are removed from consideration. Perhaps choosing a slightly higher cost in one iteration would allow smaller costs to be chosen in successive iterations. This algorithm does not allow for that. In other terms, the algorithm does not find a set of pairs that represent a globally minimal sum of costs. Another drawback of the algorithm is that it only works between two successive frames. The cost function could be extended to consider K frames (K arbitrary) of parameter sets, constructing a K-dimensional tensor instead of a matrix, but assuming equal numbers of parameter sets in all frames, the search space would grow exponentially with K. Nevertheless, the method is simple to implement, computationally negligible when K is small, and works well with a variety of signals encountered in audio [36] [57].

4.2 An optimal method

There is a way to find a set of paths over multiple frames ($K > 2$) having the lowest total cost if we restrict the search to exactly L paths. Instead of indexing parameters by their frame number h, we make h part of the parameter set so that it can be used by the distance function $\mathcal{D}$. Assume that over K frames there are M total parameter sets. In this context we will consider them as nodes in a graph. We define the vector $\boldsymbol{c} \in \mathbb{R}^{M^2}$ where the entry $\boldsymbol{c}_{i+Mj} = \mathcal{D}(\theta_i, \theta_j)$. If we have a set of connections $\Gamma_{i,j}$ we can calculate the total cost of these connections by defining the vector

$$\boldsymbol{x}_{i+Mj} = \begin{cases} 1 & \text{there is a connection between } i \text{ and } j \\ 0 & \text{otherwise} \end{cases}$$

and then forming the inner product

$$c_{\text{total}} = \langle \boldsymbol{c}, \boldsymbol{x} \rangle$$

Note that a node cannot be connected to itself. The question is how to find $\boldsymbol{x}^*$ so that c_{total} is minimized. If no constraints are placed on $\boldsymbol{x}$, the solution is trivial, but not useful. How do we constrain $\boldsymbol{x}$ to give us a solution to the partial tracking problem? Let us consider an example.

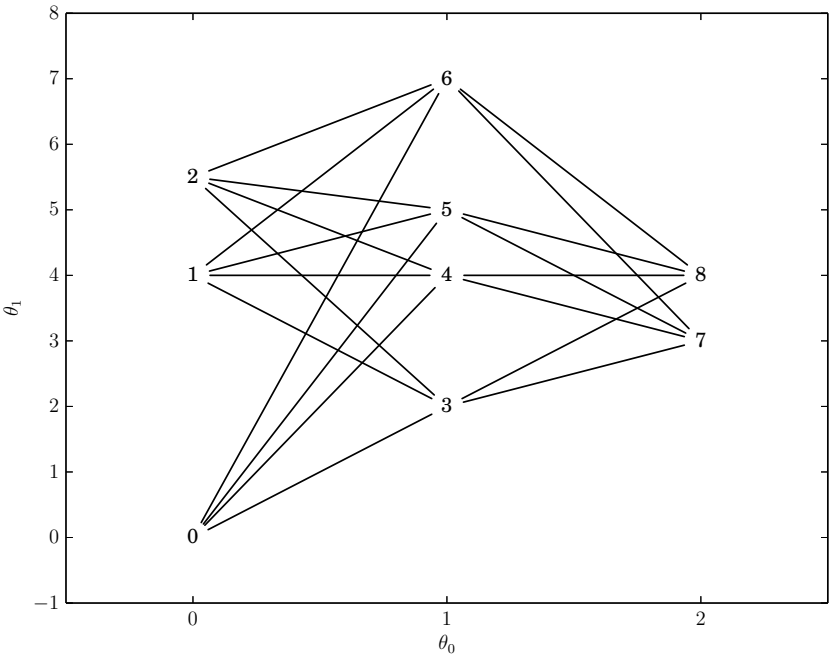

Fig. 4.1: **Possible graph connections**

In Figure 4.1 we have an example of a simple graph or lattice. Such a graph represents a plausible partial tracking situation: vertically aligned nodes are parameter sets estimated from the same analysis frame and we would like to connect these parameter sets between frames. The numbers are indices of nodes in the graph and the possible connections between them are indicated by lines, or *edges*. Imagine that we would like to find the two shortest paths. We will now examine the resulting paths from two algorithms using different criteria for shortness.

In Figure 4.2 we find the paths using an algorithm similar to Algorithm 1 but search instead over a tensor of distances $C \in \mathbb{R}^{3\times4\times2}$ whose entry $C_{i,j,h}$ represents the cost of travelling on the path connecting the ith node in layer 0, the jth node in layer 1 and the hth node in layer 2. This cost is the sum of the Euclidean distances giving the lengths of the connections. This is the greedy method of searching for the best paths whose optimality criterion is to find the set of best paths containing the absolute best path. We see in Figure 4.2 that the absolute shortest path, $1 \to 4 \to 8$, is discovered, followed by the second shortest path not using the nodes of the first path, $2 \to 5 \to 7$.

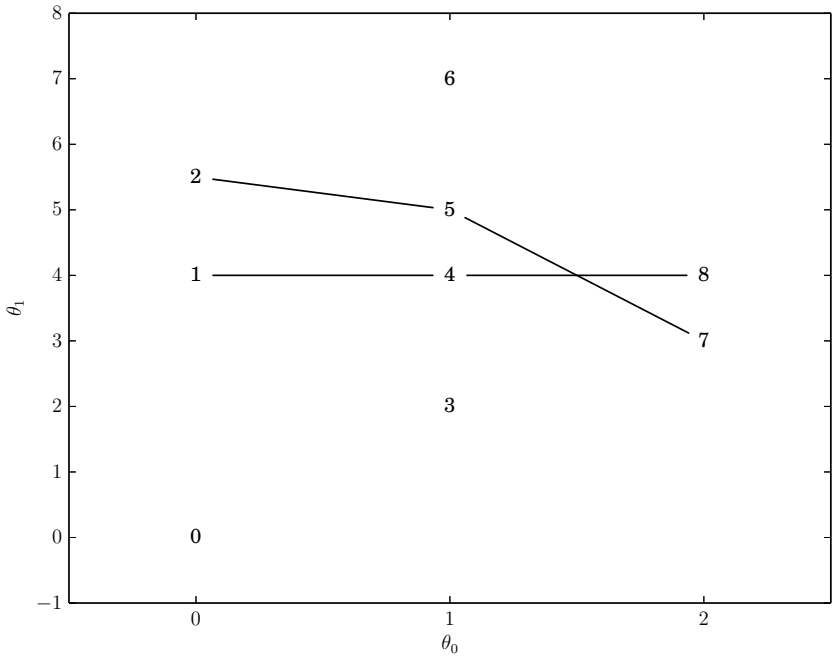

Fig. 4.2: **Two shortest paths using the greedy method**

4.2.1 L shortest paths via linear programming

To find a set of paths minimizing the total cost, we instead search for total solutions $\boldsymbol{x}$ that describe all paths in the graph. Assume for now that we can guarantee that the entries of $\boldsymbol{x}$ will be either 0 or 1. To find a set of constraints for our search, we consider the structure of a valid solution $\boldsymbol{x}^*$. To maintain that paths not overlap, a valid solution's nodes are only allowed to have one edge entering — coming from a node in a previous frame — and one edge leaving — going to a node in a successive frame. To translate this into a constraint, consider the node i and its possible R_i successive connecting nodes $j_0 \dots j_{R_i-1}$. Define the vector[1]

$$a^{\mathrm{s},i}_{i+Mj_r} = \begin{cases} 1 & \forall j_r \in [j_0 \dots j_{R_i-1}] \\ 0 & \text{otherwise} \end{cases}$$

[1]The superscript s stands for "successive".

As all the entries of $\boldsymbol{x}$ are either 0 or 1, we have

$$0 \leq \left\langle \boldsymbol{a}^{\mathrm{s},i}, \boldsymbol{x} \right\rangle \leq 1$$

so we can make this a constraint to ensure that a node has at most one path leaving. Similarly, if we consider the node j and its possible R_j previous connecting nodes $i_0 \ldots i_{R_j-1}$, the vector[2]

$$a^{\mathrm{p},j}_{i_r+Mj} \begin{cases} 1 & \forall i_r \in \left[i_0 \ldots i_{R_j-1}\right] \\ 0 & \text{otherwise} \end{cases}$$

constrains that node j have only one path entering through the constraint

$$0 \leq \left\langle \boldsymbol{a}^{\mathrm{p},j}, \boldsymbol{x} \right\rangle \leq 1$$

A node on a path will also have an edge entering and an edge leaving. To translate this into a constraint, we define a vector that counts the number of edges entering a node and subtracts then the number of edges leaving a node. The result should always be 0 for an equal number of edges entering and exiting a node. If r is the index of the node considered, the vector is simply [3]

$$\boldsymbol{a}^{\mathrm{b},r} = \boldsymbol{a}^{\mathrm{p},r} - \boldsymbol{a}^{\mathrm{s},r}$$

and the constraint

$$\left\langle \boldsymbol{a}^{\mathrm{b},r}, \boldsymbol{x} \right\rangle = 0$$

Finally we want to constrain that there be only L paths. We do this by noticing that if this is true, there will be L edges between frames h and $h+1$. We constrain the number of paths going from edges Γ_h in frame h to Γ_{h+1} by forming the vector [4]

$$\boldsymbol{a}^{\mathrm{c},h} = \sum_{j \in \Gamma_h} \boldsymbol{a}^{\mathrm{s},j}$$

and asserting the constraint

$$\left\langle \boldsymbol{a}^{\mathrm{c},h}, \boldsymbol{x} \right\rangle = L$$

[2]The superscript p stands for "previous".
[3]The superscript b stands for "balanced".
[4]The superscript c stands for "connections".

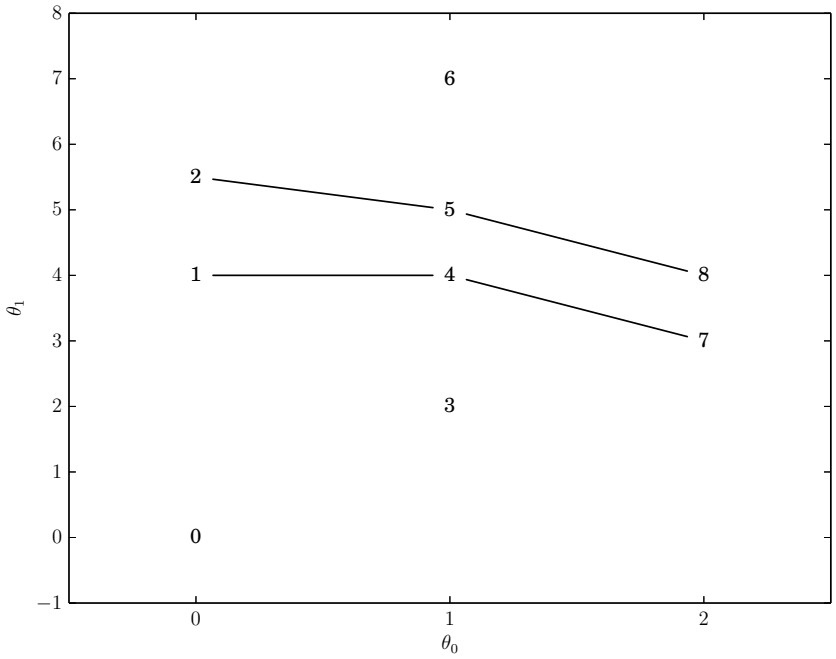

Fig. 4.3: **Two shortest paths using the LP method**

The length of $\boldsymbol{x}$ is M^2 so the total size of all the constraints is not insignificant, but most entries in the constraint vectors will be 0 and therefore the resulting constraint matrices very sparse, so sparse linear algebra routines can be used in computations. Furthermore, the $\boldsymbol{a}^b$ and $\boldsymbol{a}^c$ constraints are derived from $\boldsymbol{a}^p$ and $\boldsymbol{a}^s$, so only the latter need to be stored.

The complete *linear program (LP)* solving the L shortest paths problem is then

$$\min_{\boldsymbol{x}} \langle \boldsymbol{c}, \boldsymbol{x} \rangle$$

subject to

$$\mathbf{0} \leq \begin{bmatrix} \boldsymbol{A}_{\mathrm{s}} \\ \boldsymbol{A}_{\mathrm{p}} \end{bmatrix} \boldsymbol{x} \leq \mathbf{1}$$

$$\begin{bmatrix} \boldsymbol{A}_{\mathrm{b}} \\ \boldsymbol{A}_{\mathrm{c}} \end{bmatrix} \boldsymbol{x} = \begin{bmatrix} \mathbf{0} \\ L\mathbf{1} \end{bmatrix}$$

$$\mathbf{0} \leq \boldsymbol{x} \leq \mathbf{1}$$

where $\boldsymbol{A}_\mathrm{s}$ is the matrix with $\boldsymbol{a}^{\mathrm{s},m}$ as its rows for $m \in [0 \dots M-1]$ and $\boldsymbol{A}_\mathrm{p}$ is the matrix with $\boldsymbol{a}^{\mathrm{p},m}$ as its rows, etc.

The solution of the two best paths using the LP formulation is shown in Figure 4.3 and a comparison of the total costs is shown in Table 4.1

The LP formulation is inspired by a multiple object tracking algorithm for video [23]. A proof that the solution $\boldsymbol{x}^*$ will have entries equal to either 0 or 1 can be found in [44, p. 167]. The theoretical computational complexity of the linear program is polynomial in the number of variables, see [25] for a proof and the demonstration of a fast algorithm for finding its solution. In practice, to extract paths from the solution, we do not test equality with 0 or 1 but rather test if the solution vector's values are greater than some threshold. This may mean that suboptimal solutions may still be close enough. The tolerance of the solutions to suboptimality should be investigated, as if they are tolerant, fewer iterations of a barrier-based algorithm would be required to solve the problem. More information on linear programming and optimization in general can be found in [3].

4.2.2 Complexity

The LP formulation of the L-best paths problem gives results equivalent to the solution to the L-best paths problem proposed in [63]. The complexity of our algorithm is different. Assuming we use the algorithm in [25] to solve the LP, our program has a complexity of $O(M^7B^2)$ where M is the number of nodes (parameter sets) and B is the number of bits used to represent each number in the input. The complexity of the algorithm by Wolf in [63] is equivalent to the Viterbi algorithm for finding the single best path through a trellis whose hth frame has $\binom{N_h}{L}\binom{N_{h+1}}{L}L!$ connections where N_h and N_{h+1} are the number of nodes in two consecutive frames of the original lattice. Therefore, assuming a constant number N of nodes in each frame, its complexity is $O((\binom{N}{L}^2 L!)^2 T)$. If there are few nodes in each frame and a small number of paths are searched, Wolf's formulation is superior as its complexity increases linearly with the number of frames in the lattice. On the other hand,

Table 4.1. Comparison of total costs in Figure 4.3

Greedy	LP
5.354102	4.946461

if each frame has a large number of nodes or many paths are searched, the LP formulation is superior. Informally we have found this to agree with reality — both algorithms were tried when producing the figures in Section 4.3. Indeed the Wolf formulation took prohibitively long to compute when many paths were desired, as did the LP when many frames were considered.

It should be noted that in the special case that only 1 shortest path is searched an algorithm exists that requires on the order of N^2T calculations [46] where N is the number of nodes in each frame and T is the number of frames (assuming the same number of nodes in each frame): this algorithm is known as the Viterbi algorithm [13].

4.3 Partial paths on an example signal

We compare the greedy and LP based methods for peak matching on a synthetic signal. The signal is composed of $K = 6$ chirps of constant amplitude, the kth chirp s at sample n described by the equation

$$s_k(n) = \exp(j(\phi_k + \omega_k n + \frac{1}{2}\psi_k n^2))$$

The parameters for the 6 chirps are presented in Table 4.2.

Two 1 second long signals are synthesized at a sampling rate of 16000 Hz, the first with chirps 0–2, the second with chirps 3–5. We add Gaussian distributed white noise at several SNR to evaluate the technique in the presence of noise.

Table 4.2. Parameters of kth chirp. f_0 and f_1 are the initial and final frequency of the chirp in Hz.

k	ϕ_k	ω_k	ψ_k	f_0	f_1
0	0	0.20	2.45×10^{-6}	500	600
1	0	0.39	4.91×10^{-6}	1000	1200
2	0	0.59	7.36×10^{-6}	1500	1800
3	0	0.27	-7.36×10^{-6}	700	400
4	0	0.55	-1.47×10^{-5}	1400	800
5	0	0.82	-2.21×10^{-5}	2100	1200

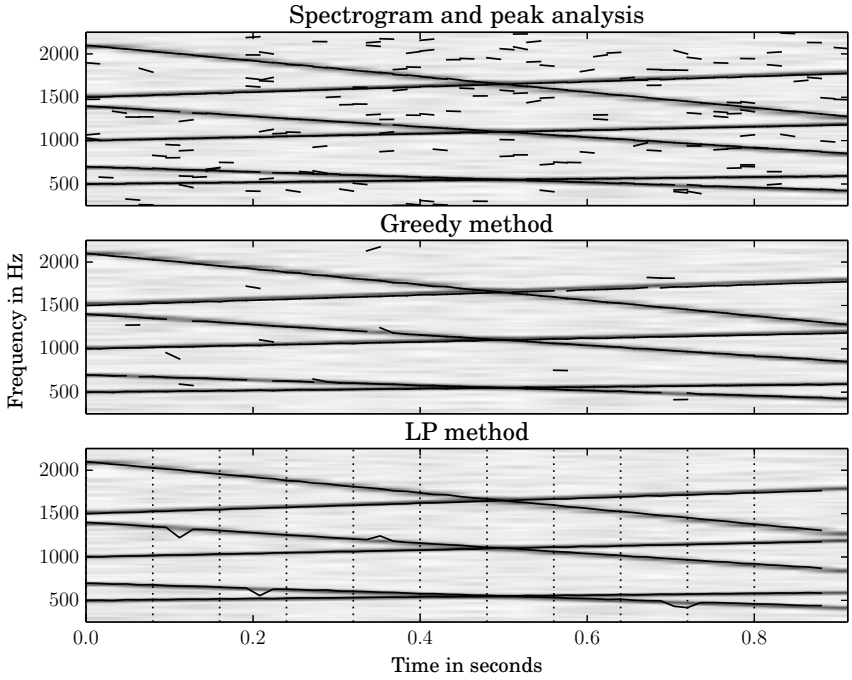

Fig. 4.4: **Compare greedy and LP partial tracking on chirps in noise, SNR 20 dB.** Line-segments representing the frequency and frequency-slope at local spectrogram maxima. In the bottom two plots the line segments not deemed by the respective algorithms as belonging to a partial path are discarded, revealing the estimated partial trajectories. See Table 4.2 for the chirp parameters.

A spectrogram of each signal is computed with an analysis window length of 1024 samples and a hop-size H of 256 samples. Local maxima are searched in 150 Hz wide bands spaced 75 Hz apart. A local maximum is only accepted if its amplitude is greater than -20 dB. At each local maximum the DDM is used to estimate the local chirp parameters, the ith set of parameters in frame h denoted $\theta_i^h = \left\{\phi_i^h, \omega_i^h, \psi_i^h\right\}$. The results of the analyses of both signals are lumped together and it is on this lumped data that we perform partial tracking.

We search for partial tracks using both the greedy and LP strategies. Both algorithms use the distance metric $\mathcal{D}_{\text{pr.}}$ between two parameters sets:

$$\mathcal{D}_{\text{pr.}}\left(\theta_i^h, \theta_j^{h+1}\right) = \left(\omega_i^h + \psi_i^h H - \omega_j^{h+1}\right)$$

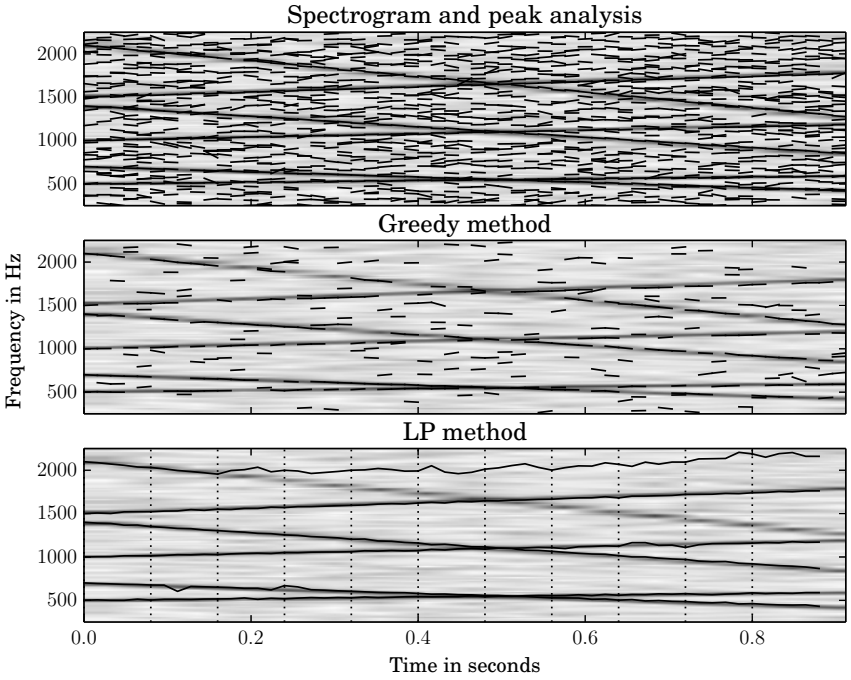

Fig. 4.5: **Compare greedy and LP partial tracking on chirps in noise, SNR 15 dB.** Line-segments representing the frequency and frequency-slope at local spectrogram maxima. In the bottom two plots the line segments not deemed by the respective algorithms as belonging to a partial path are discarded, revealing the estimated partial trajectories. See Table 4.2 for the chirp parameters.

which is the error in predicting jth frequency in frame $h+1$ from the ith parameters in frame h. For the greedy method, the search for partial paths is restricted to one frame ahead like in [36]. For the LP method, to keep the computation time reasonable, we search over 6 frames for 6 best paths[5]. To maintain connected paths, the search on the next frames uses the end nodes of the last search as starting points. For both methods, the search is restricted to nodes between frequencies 250 to 2250 Hz.

Figures 4.4, 4.5, and 4.6 show discovered partial trajectories for signals with a SNR of 20, 15, and 10 dB, respectively. It is seen that while the greedy method begins to perform poorly at a SNR of 15dB, the LP method still gives plausible partial trajectories for SNRs of 10 and 15 dB. At lower SNRs, the LP formulation gives some paths that do not correspond to an underlying partial. These could be filtered out by examining the cost of these paths

[5]The number of paths does not affect the computation time.

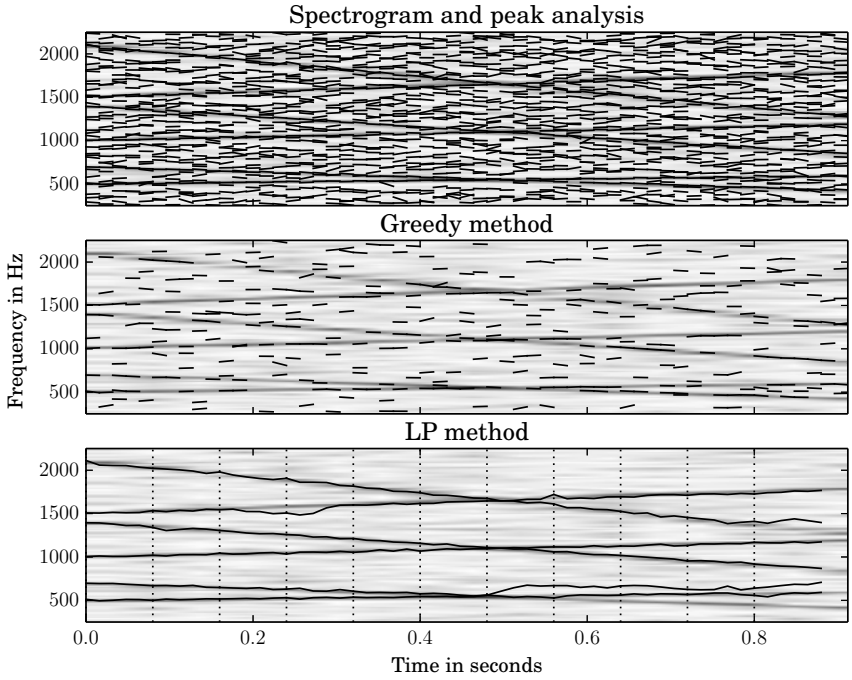

Fig. 4.6: **Compare greedy and LP partial tracking on chirps in noise, SNR 10 dB.** Line-segments representing the frequency and frequency-slope at local spectrogram maxima. In the bottom two plots the line segments not deemed by the respective algorithms as belonging to a partial path are discarded, revealing the estimated partial trajectories. See Table 4.2 for the chirp parameters.

and comparing them to the costs of the others. Those that deviate from a mean cost more than a certain amount should be rejected. This is the strategy used in Chapter 7 and illustrated in Figure 7.6. In any case, the lower SNRs are relatively challenging for any partial tracking technique.

But why did the LP discover a path not present in the underlying signal? This is due to the cost function, which finds a path with minimum prediction error in using the frequency and frequency slope coefficients of one node to predict another node's frequency coefficient. When there are many nodes in the original analysis it is not surprising that some unexpected path exists. An attribute of these erroneous paths is that they are not smooth. To deter the algorithm from finding such paths, regularization could be used like in Section 3.3.1 that minimizes the integral of the squared estimate of the path's second derivative. More on regularization in optimization can be found in [3, ch. 6.3].

4.4 Conclusion

In this chapter we reformulated the classical greedy algorithm of McAulay and Quatieri and showed that it can be seen as a greedy algorithm for finding the L shortest paths in a lattice. An algorithm was then proposed minimizing the sum of the L paths, using a linear programming approach. It was shown on synthetic signals that the new approach finds plausible paths in lattices with a large number of spurious nodes.

There are problems with the proposed approach. As discussed in 4.2, "jagged" paths should be removed using regularization. There are also situations where it is undesirable to have paths extend throughout the entire lattice. Acoustic signals produced by striking media, such as strings or bars, exhibit a spectrum where the upper partials decay more quickly than the lower ones (e.g., see Figure 7.1) — it would be desirable in these situations to have shorter paths for the upper partials, those decaying more quickly. This could be addressed as in [9] where the signal is divided into overlapping sequences of frames and partial paths are connected between sequences.

The proposed algorithm, while faster than algorithms based on the Viterbi algorithm, is still not fast. Assuming the same cost function $\mathcal{D}_{\text{pr.}}$ as in Section 4.3 it would be more efficient to consider narrow bands over which to search for paths when analysing signals with little frequency modulation. However, as we will see in Chapter 6, with different cost functions, the algorithm is useful for solving general L shortest paths problems outside of partial tracking.

Chapter 5

The extended phase and amplitude model

In Chapter 4 techniques were presented for discovering partials in a signal. Each partial is a set of analysis points indexed by time. The information at each analysis point can be used to synthesize a portion of the partial and these are combined to give a signal representing a partial. The various techniques to synthesize these pieces of the signal discussed here differ in the orders used for analysis and those for synthesis. The first technique presented simply synthesizes signals of short duration using the estimated parameters and blends these segments together. We recognize that having multiple estimations of parameters at discrete times within the partial duration allow us to postulate, via interpolation, functions describing the partial of higher-order than those whose parameters were estimated during analysis. It is shown that this strategy is not always to our advantage — interpolants of higher-order than the underlying function can suffer from errors due to over-fitting. In the case of functions that are always better approximated by higher-order polynomials, we will see that there is an advantage to using high-order interpolation. These cases are illustrated through the analysis and synthesis of synthetic signals.

5.1 Partial synthesis

A popular technique for synthesizing partials from a set of analysis points is the *overlap-and-add* procedure [45], [40]. We assume that in the neighbourhood of τ_r the partial's signal is approximately described by the function $x(n) \approx f_{\tau_r}(n - \tau_r)$. To synthesize an

approximation of x we sum windowed f_{τ_r} at multiple locations, windowed by a function w with finite support so the resulting signal has finite energy and the piecewise assumption is maintained. For simplicity we assume the τ_r are equally spaced by H samples, and $\tau_0 = 0$, so we have $\tau_r = rH$. The length of the window function w is $M = VH + 1$ samples, with $V, H \in \mathbb{N}^1$. The approximate signal at sample n is then

$$\tilde{x}(n) = \sum_{l=L_-}^{L^+} w(n - lH) f_{\tau_l}(n - \tau_l)$$

where

$$L_- = \left\lceil \frac{n}{H} \right\rceil - V$$

and

$$L_+ = \left\lfloor \frac{n}{H} \right\rfloor + V$$

Not a big deal, but I think L- and L+ might depend on r (contrary to what is mentioned in Portnoff's paper

This method has some drawbacks. Usually the function f is an approximation $\tilde{f}$ of the true underlying function. In the case of partial tracking, often partials that are too short are discarded or missed. At amplitude transients, these short partials are important for reproducing sharp attacks that are shorter than the window length. If these partials are missing, the resulting signal takes on a transient similar to the window shape. This could be overcome by choosing a window with a shape similar to the overall amplitude envelope in the attack region when resynthesizing an attack transient.

Another drawback is that no attempt is made to interpolate between the functions estimated at τ_r and τ_{r+1} using the model that the underlying sinusoids are non-stationary[2] From Equation 3.3 we know we can estimate a polynomial of arbitrary order for the argument of the complex exponential. We will see that using this additional information can give us an interpolating function closer to the underlying model.

[1] M is always odd so one may wonder how the Fast Fourier Transform can be used to invert F_{τ_r}, the frequency-domain representation of f_{τ_r}. Recall that the DTFT can be interpreted as the coefficients of a Fourier series that give the periodic version of the analysed signal. Also recall that we use window functions that are real and even. In practice the edges of the window are often equal to zero so that the length of the non-zero part is equal to the length of the DTFT N. In the case they are not, simply ensuring that values in the window indexed by integer multiples of N are 0, and that the value at the centre of the window is 1 will ensure proper synthesis [45, p. 244]. In that case, the values outside of the part of the window presented to the DTFT are folded into this region using the window indices modulo-N. See [45] and [40] for more details on this procedure.

[2] This is one of the causes of "pre-echo" when time-stretching using the STFT [49].

5.2 The interpolating analysis-synthesis system

In the following, we investigate the synthesis quality of three interpolating analysis-synthesis systems. The qualifier "interpolating" is used because each system takes the multiple sets of estimated parameters of a smaller-order model and interpolates them with a higher-order model, which is then used for synthesis. This is necessary because we only have values every H samples from the analysis step but require a value for every sample value in the output. The systems will be denoted $\mathscr{S}_{p,q}$ where p is the order of the analysis system and q the order of the synthesis system, e.g., a linear analysis system has $p = 1$, etc.

5.3 $\mathscr{S}_{1,3}$: the McAulay-Quatieri method

5.3.1 Analysis: linear phase and constant amplitude

For the McAulay-Quatieri method, the analysis model is a sinusoid of constant frequency and amplitude (linear phase and constant log-amplitude) in each analysis frame. To estimate the frequency of this sinusoid we find the bin with the most energy and find a refined estimate of the frequency as the maximum of a quadratic interpolating polynomial fit to this bin and its two neighbouring bins. This is a procedure documented in [52, p. 45]. The interpolation is best performed in the log-spectrum and on a spectrum produced using a window, such as a zero-padded Hann window, giving a wide enough main-lobe so that the three points lie on this lobe and not on side-lobes. A refined estimate of the amplitude of the sinusoid is obtained with this procedure as well.

In the original paper by McAulay and Quatieri, they do not use this technique but, as they show an example analysis of a speech signal, instead adjust the analysis window to be a multiple of the period of the glottal pulse. The bins of the DTFT used in the analysis will be integer multiples of the frequency given as the reciprocal of this period. Under the model of the speech signal as harmonically related sinusoids, the best estimate for the frequency is the bin of a local maximum, its amplitude the modulus of the spectrum at this maximum, and the phase the argument.

In our system, we use a fixed window size. To estimate the initial phase then we use Equation 3.4 with

$$\gamma_{\mathrm{MQ}}(n) = \exp(2\pi \frac{k^*}{M} n)$$

where k^* is the bin we have determined to correspond to the frequency of the sinusoid. The initial phase is then $\Im\{c_0\}$.

5.3.2 Synthesis: cubic phase and linear log-amplitude

The phase part

Given two local maxima of the DTSTFT $X(\tau_0, \omega_0)$ and $X(\tau_1, \omega_1)$, where $H = \tau_1 - \tau_0$ we can conjecture a cubic polynomial phase function for the imaginary part of the phase argument

$$\tilde{\phi}(n) = \Im\{c_3\}(n-\tau_0)^3 + \Im\{c_2\}(n-\tau_0)^2 + \Im\{c_1\}(n-\tau_0) + \Im\{c_0\}$$

By noting that we have 2 measurements of the phase and frequency, $\angle\{X(\tau_0, \omega_0)\}$ and $\angle\{X(\tau_1, \omega_1)\}$, and the frequency is the derivative of the phase, we can solve for the coefficients of the polynomial phase function using the following linear system of equations, assuming the DTSTFT was computed using a real and even window

$$\begin{pmatrix} 0 & 0 & 0 & 1 \\ H^3 & H^2 & H & 1 \\ 0 & 0 & 1 & 0 \\ 3H^2 & 2H & 1 & 0 \end{pmatrix} \begin{pmatrix} \Im\{c_3\} \\ \Im\{c_2\} \\ \Im\{c_1\} \\ \Im\{c_0\} \end{pmatrix} = \begin{pmatrix} \angle\{X(\tau_0, \omega_0)\} \\ \angle\{X(\tau_1, \omega_1)\} + 2\pi M \\ \omega_0 \\ \omega_1 \end{pmatrix}$$

We choose M so that

$$\int_0^H \left(\frac{d^2\tilde{\phi}}{dt^2}(t)\right)^2 dt \quad (5.1)$$

Frequency (Note: This is why we use the second derivative of the phase

is minimized in order to have a smooth evolution of phase in the interpolated region. M is necessary because some integer number of periods of a sinusoid will have passed from times τ_0 to τ_1. Informally we choose M so that a polynomial describing the phase evolution between these two times takes a direct route, which is a plausible criterion because a signal with more radical phase variation would unlikely exhibit a spectrum that could be well described by two points in the time-frequency plane, i.e., the signal would exhibit a large bandwidth. See [36, p. 751] for further clarification.

The amplitude part

As only two measurements of the amplitude of the sinusoid are available, $|X(\tau_0, \omega_0)|$ and $|X(\tau_1, \omega_1)|$, the coefficients c_3 and c_2 are purely imaginary and the real parts of c_1 and c_0

are determined as

$$\begin{pmatrix} 0 & 1 \\ H & 1 \end{pmatrix} \begin{pmatrix} \Re\{c_1\} \\ \Re\{c_0\} \end{pmatrix} = \begin{pmatrix} \log(|X(\tau_0, \omega_0)|) \\ \log(|X(\tau_1, \omega_1)|) \end{pmatrix}$$

5.4 $\mathscr{S}_{2,3}$ and $\mathscr{S}_{2,5}$: the DDM-based methods

Here we extend the $\mathscr{S}_{1,3}$ model of McAulay-Quatieri to account for the additional parameters estimated via the DDM. For the $\mathscr{S}_{2,3}$ model, we must introduce additional constraints into the system as we have more estimated parameters than are available in the synthesis model. It would be possible to solve this system via least-squares, but the proposed constraints simplify analytically the expression maximizing the smoothness of the phase and log-amplitude functions, and give satisfactory results. For the $\mathscr{S}_{2,5}$ model the derivation is straightforward as in the $\mathscr{S}_{1,3}$ case — there are the same number of estimated parameters as there are parameters in the model.

5.4.1 Analysis: quadratic phase and log-amplitude

The DDM is used on segments of the signal to estimate the parameters of sinusoid with a complex quadratic polynomial argument. This sinusoid has the form

$$x_a(n) = \exp\left(a_2 n^2 + a_1 n + a_0\right) \tag{5.2}$$

with $a_i \in \mathbb{C}$.

We can estimate the coefficients of Equation 5.2 using the DDM. We write Equation 3.3 in matrix form with $Q = 2$

$$\begin{pmatrix} \langle \mathcal{T}^0 x, \overline{\psi}_1 \rangle & 2\langle \mathcal{T}^1 x, \overline{\psi}_1 \rangle \\ \vdots & \vdots \\ \langle \mathcal{T}^0 x, \overline{\psi}_R \rangle & 2\langle \mathcal{T}^1 x, \overline{\psi}_R \rangle \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} -\left\langle x, \frac{d\overline{\psi}_1}{dn} \right\rangle \\ \vdots \\ -\left\langle x, \frac{d\overline{\psi}_R}{dn} \right\rangle \end{pmatrix}$$

From this we recognize we need to define three functions

$$\langle \mathcal{T}^0 x, \overline{\psi}_k \rangle = \sum_{m=0}^{M-1} w(m) x(m+\tau) \exp(-j2\pi \frac{km}{M})$$

$$\left\langle \mathcal{T}^1 x, \overline{\psi}_k \right\rangle = \sum_{m=0}^{M-1} mw(m)x(m+\tau)\exp(-j2\pi\frac{km}{M})$$

$$\left\langle x, \frac{d\overline{\psi}_k}{dn} \right\rangle = -j2\pi\frac{k}{M}X_{p_1}(\tau,k) + \sum_{m=0}^{M-1}\frac{dw}{dn}(m)x(m+\tau)\exp(-j2\pi\frac{km}{M})$$

Where M is the length of the window and k is the frequency "bin"[3]. We also only consider x at the samples $m = [0,\ldots,M-1]$ because we can always shift the time reference to view an arbitrary contiguous segment of signal with these indices.

We then find k^* such that X is maximum. If multiple components are present in the signal and are sufficiently separated in frequency, we can split the signal up into frequency bands and find local maxima. A technique for doing so is described in [52, p. 42]. To have a system of equations with a unique solution, we take the two adjacent bins $k-1$ and $k+1$ to have enough unique atoms for Equation 3.3. These bins should only contain energy from the component whose parameters we are interested in measuring — this is true if the components are adequately separated in time and frequency. We could choose only two bins to have a non-singular system, and there are many possibilities for choosing different atoms [2, p. 4639]. We choose three from the same frame to have improved estimation accuracy in situations where components are adequately separated in frequency, while avoiding a more sophisticated local peak selection procedure. Then a_2 and a_1 can be determined by solving the linear system

$$\begin{pmatrix} \left\langle \mathcal{T}^0 x, \overline{\psi}_{k-1} \right\rangle & \left\langle \mathcal{T}^1 x, \overline{\psi}_{k-1} \right\rangle \\ \left\langle \mathcal{T}^0 x, \overline{\psi}_{k} \right\rangle & \left\langle \mathcal{T}^1 x, \overline{\psi}_{k} \right\rangle \\ \left\langle \mathcal{T}^0 x, \overline{\psi}_{k+1} \right\rangle & \left\langle \mathcal{T}^1 x, \overline{\psi}_{k+1} \right\rangle \end{pmatrix} \begin{pmatrix} a_1 \\ 2a_2 \end{pmatrix} = \begin{pmatrix} -\left\langle x, \frac{d\overline{\psi}_{k-1}}{dn} \right\rangle \\ -\left\langle x, \frac{d\overline{\psi}_{k}}{dn} \right\rangle \\ -\left\langle x, \frac{d\overline{\psi}_{k+1}}{dn} \right\rangle \end{pmatrix}$$

With a_1 and a_2 determined, we can use Equation 3.4 to estimate a_0. We will write a_i^τ to refer to coefficient i determined at time τ.

5.4.2 Synthesis: cubic order ($\mathscr{S}_{2,3}$)

In this section we describe how to obtain a cubic phase and log-amplitude polynomial from local estimations of the coefficients of a quadratic phase and log-amplitude polynomial.

[3]For tractability, the functions $X(\tau,k)$ are only evaluated at a finite number of frequencies, which are often called "bins" in the signal processing literature.

The phase part

A complex sinusoid with cubic phase has the following form:

$$\beta(n) = \exp\left(j\left(b_3 n^3 + b_2 n^2 + b_1 n + b_0\right)\right) \tag{5.3}$$

with $b_i \in \mathbb{R}$. This sinusoid has magnitude 1 everywhere, only its phase is changing.

Once the $\boldsymbol{a}^\tau$ have been determined at two times τ_0 and τ_1, with $H = \tau_1 - \tau_0$, and these times have between determined as connected (see Chapter 4), we can write a system of equations to determine an interpolating cubic phase polynomial. To avoid numerical instabilities and for simplicity, we shift the time origin so that $\tau_0 = 0$. This means $b_0 = \Im\{a_0^{\tau_0}\}$. To reduce the size of the system, we require that

Should appear on the same line

$$\frac{d\phi}{dn}\left(\frac{H}{2}\right) = \frac{1}{2}\left(\Im\{a_1^{\tau_0}\} + \Im\{a_1^{\tau_1}\}\right)$$

and

$$\frac{d^2\phi}{dn^2}\left(\frac{H}{2}\right) = \frac{1}{2}\left(\Im\{a_2^{\tau_0}\} + \Im\{a_2^{\tau_1}\}\right)$$

i.e., the frequency and first-order frequency modulation in the middle of the segment are the average of the two measured coefficients. Finally we require that the change in phase from time 0 to H correspond to ~~that~~ what was observed, but account for the cycles that were not observed by adding an integer number of 2π radians. If $\phi(n) = j(b_3 n^3 + b_2 n^2 + b_1 n + b_0)$, then

$$\phi(H) = \Im\{a_0^{\tau_1}\} - \Im\{a_0^{\tau_0}\} + 2\pi U^*$$

where $U^* \in \mathbb{Z}$ is determined to minimize Equation 5.1, in this case:

$$\tilde{U} = \arg\min_U \int_0^H \left(6b_3 t + 2b_2\right)^2 dt \tag{5.4}$$

which is then rounded to the nearest integer to give U^*. To summarize we have

$$\begin{pmatrix} H^3 & H^2 & H \\ \frac{3}{4}H^2 & H & 1 \\ 3H & 2 & 0 \end{pmatrix} \begin{pmatrix} b_3 \\ b_2 \\ b_1 \end{pmatrix} = \begin{pmatrix} \Im\{a_0^{\tau_1}\} - \Im\{a_0^{\tau_0}\} + 2\pi U^* \\ \frac{1}{2}\left(\Im\{a_1^{\tau_0}\} + \Im\{a_1^{\tau_1}\}\right) \\ \frac{1}{2}\left(\Im\{a_2^{\tau_0}\} + \Im\{a_2^{\tau_1}\}\right) \end{pmatrix}$$

Solving for $b_1, \ldots, b_3$, we have

$$\begin{pmatrix} b_3 \\ b_2 \\ b_1 \end{pmatrix} = \begin{pmatrix} \frac{4}{H^3}\left(\Im\left\{a_0^{\tau_1}\right\} - \Im\left\{a_0^{\tau_0}\right\} + 2\pi U^*\right) - \frac{2}{H^2}\left(\Im\left\{a_1^{\tau_1}\right\} + \Im\left\{a_1^{\tau_0}\right\}\right) \\ \frac{-6}{H^2}\left(\Im\left\{a_0^{\tau_1}\right\} - \Im\left\{a_0^{\tau_0}\right\} + 2\pi U^*\right) - \frac{3}{H}\left(\Im\left\{a_1^{\tau_1}\right\} + \Im\left\{a_1^{\tau_0}\right\}\right) + \frac{1}{4}\left(\Im\left\{a_2^{\tau_0}\right\} + \Im\left\{a_2^{\tau_1}\right\}\right) \\ \frac{-H}{4}\left(\Im\left\{a_2^{\tau_0}\right\} + \Im\left\{a_2^{\tau_1}\right\}\right) + \frac{3}{H}\left(\Im\left\{a_0^{\tau_1}\right\} - \Im\left\{a_0^{\tau_0}\right\} + 2\pi U^*\right) - \Im\left\{a_1^{\tau_1}\right\} - \Im\left\{a_1^{\tau_0}\right\} \end{pmatrix}$$

and then $\tilde{U}$ is determined using Equation 5.4 to be

$$\tilde{U} = \frac{1}{4\pi}\left[H\left(\Im\left\{a_1^{\tau_1}\right\} + \Im\left\{a_1^{\tau_0}\right\}\right) - 2\left(\Im\left\{a_0^{\tau_1}\right\} - \Im\left\{a_0^{\tau_0}\right\}\right)\right]$$

and then rounded to obtain U^*.

The amplitude part

Solving for the cubic polynomial describing the local log-amplitude function

$$\mu(n) = \exp\left(d_3 n^3 + d_2 n^2 + d_1 n + d_0\right)$$

with $d_i \in \mathbb{R}$, is more straightforward analytically as it does not require solving to maximize the smoothness of resulting polynomial. To require continuity at the end-points of our polynomial, we require

$$\mu(0) = \Re\left\{a_0^{\tau_0}\right\}$$

and

$$\mu(H) = \Re\left\{a_0^{\tau_1}\right\}$$

The first constraint is satisfied simply by setting $d_0 = \Re\left\{a_0^{\tau_0}\right\}$. The second will be accounted for in a constrained least-squares solution for the other coefficients. The other observations are

$$\frac{d\mu}{dn}(0) = \Re\left\{a_1^{\tau_0}\right\}$$

$$\frac{d\mu}{dn}(H) = \Re\left\{a_1^{\tau_1}\right\}$$

$$\frac{d^2\mu}{dn^2}(0) = \Re\left\{a_2^{\tau_0}\right\}$$

$$\frac{d^2\mu}{dn^2}(H) = \Re\left\{a_2^{\tau_1}\right\}$$

The constrained least-squares problem to be solved is then

$$\begin{pmatrix} 0 & 0 & 1 \\ 3H^2 & 2H & 1 \\ 0 & 2 & 0 \\ 6H & 2 & 0 \end{pmatrix} \begin{pmatrix} d_3 \\ d_2 \\ d_1 \end{pmatrix} = \begin{pmatrix} \Re\{a_1^{\tau_0}\} \\ \Re\{a_1^{\tau_1}\} \\ \Re\{a_2^{\tau_0}\} \\ \Re\{a_2^{\tau_1}\} \end{pmatrix}$$

subject to

$$\begin{pmatrix} H^3 & H^2 & H \end{pmatrix} \begin{pmatrix} d_3 \\ d_2 \\ d_1 \end{pmatrix} = \begin{pmatrix} \Re\{a_0^{\tau_1}\} \end{pmatrix}$$

This can be solved using numerical methods, in particular, using a specific interpretation of weighted least-squares [16, p. 266].

5.4.3 Synthesis: quintic order ($\mathscr{S}_{2,5}$)

The phase part

Solving for the coefficients of a quintic phase polynomial is done very similarly to Section 5.4.2. As we have the same number of analysis and synthesis parameters, no constraints have to be introduced to solve the system apart from the value U that maximizes smoothness of the phase function. The quintic phase polynomial is[4]

$$\lambda(n) = \exp\left(j\left(u_5 n^5 + u_4 n^4 + u_3 n^3 + u_2 n^2 + u_1 n + u_0\right)\right) \tag{5.5}$$

with $u_i \in \mathbb{R}$. We have

$$\begin{aligned} \phi(0) &= \Im\{a_0^{\tau_0}\} \\ \frac{d\phi}{dn}(0) &= \Im\{a_1^{\tau_0}\} \\ \frac{d^2\phi}{dn^2}(0) &= \Im\{a_2^{\tau_0}\} \end{aligned}$$

[4]Remember, this sinusoid has constant amplitude of 1 and this function only describes its change of phase.

and solving for the remaining coefficients is done using the linear system of equations:

$$\begin{pmatrix} H^5 & H^4 & H^3 \\ 5H^4 & 4H^3 & 3H^2 \\ 20H^3 & 12H^2 & 6H \end{pmatrix} \begin{pmatrix} u_5 \\ u_4 \\ u_3 \end{pmatrix} = \begin{pmatrix} -\frac{H^2}{2}\Im\left\{a_2^{\tau_0}\right\} - H\Im\left\{a_1^{\tau_0}\right\} + \Im\left\{a_0^{\tau_1}\right\} - \Im\left\{a_1^{\tau_1}\right\} + 2\pi U^* \\ -H\Im\left\{a_2^{\tau_0}\right\} + \Im\left\{a_1^{\tau_1}\right\} - \Im\left\{a_1^{\tau_0}\right\} \\ \Im\left\{a_2^{\tau_1}\right\} - \Im\left\{a_2^{\tau_0}\right\} \end{pmatrix}$$

The smoothness maximizing $\tilde{U}$ is found as

$$\tilde{U} = \frac{1}{80\pi}\left[20H(\Im\left\{a_1^{\tau_0}\right\} + \Im\left\{a_1^{\tau_1}\right\}) + H^2(\Im\left\{a_2^{\tau_0}\right\} - \Im\left\{a_2^{\tau_1}\right\}) + 40(\Im\left\{a_0^{\tau_0}\right\} - \Im\left\{a_0^{\tau_1}\right\})\right]$$

and then rounded to produce U^* as above.

The quintic interpolating phase polynomial has been proposed in a previous paper [15] although they do not directly estimate the frequency slope, choosing instead to derive it using the difference in frequency between two analysis frames.

Still a lot of space; you might use \vspace to tune it properly.

The amplitude part

Solving for the quintic log-amplitude polynomial

$$\rho(n) = \exp\left(v_5 n^5 + v_4 n^4 + v_3 n^3 + v_2 n^2 + v_1 n + v_0\right) \tag{5.6}$$

with $v_i \in \mathbb{R}$, is as follows:

$$\mu(0) = \Re\left\{a_0^{\tau_0}\right\}$$
$$\frac{d\mu}{dn}(0) = \Re\left\{a_1^{\tau_0}\right\}$$
$$\frac{d^2\mu}{dn^2}(0) = \Re\left\{a_2^{\tau_0}\right\}$$

and solving for the remaining coefficients is done using the linear system of equations:

$$\begin{pmatrix} H^5 & H^4 & H^3 \\ 5H^4 & 4H^3 & 3H^2 \\ 20H^3 & 12H^2 & 6H \end{pmatrix} \begin{pmatrix} v_5 \\ v_4 \\ v_3 \end{pmatrix} = \begin{pmatrix} -\frac{H^2}{2}\Re\left\{a_2^{\tau_0}\right\} - H\Re\left\{a_1^{\tau_0}\right\} + \Re\left\{a_0^{\tau_1}\right\} - \Re\left\{a_1^{\tau_1}\right\} \\ -H\Re\left\{a_2^{\tau_0}\right\} + \Re\left\{a_1^{\tau_1}\right\} - \Re\left\{a_1^{\tau_0}\right\} \\ \Re\left\{a_2^{\tau_1}\right\} - \Re\left\{a_2^{\tau_0}\right\} \end{pmatrix}$$

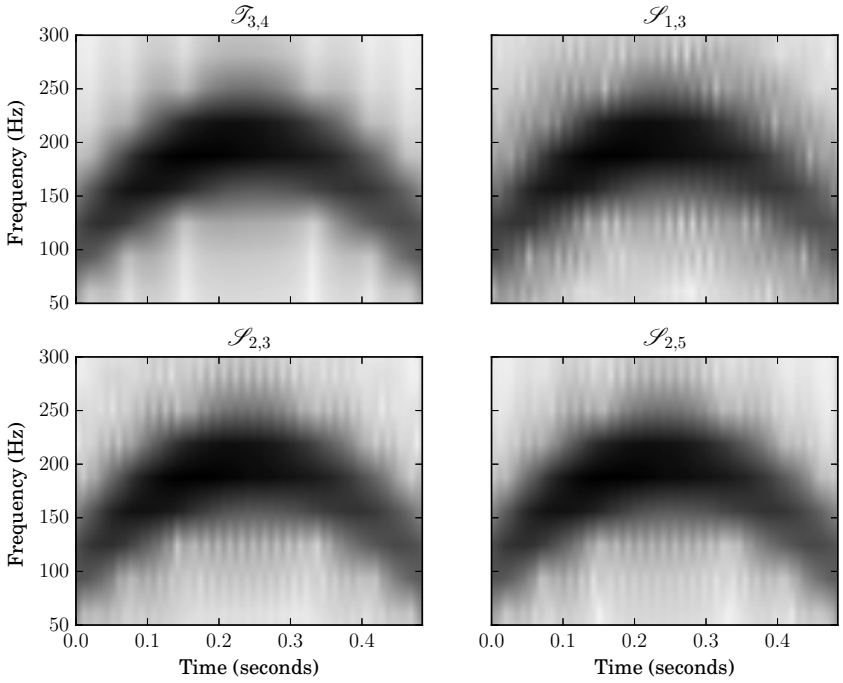

Fig. 5.1: **Spectrograms of $\mathscr{T}_{3,4}$, $\mathscr{S}_{1,3}$, $\mathscr{S}_{2,3}$ and $\mathscr{S}_{2,5}$.** Spectrograms of the true signal and estimated signals for the $\mathscr{T}_{3,4}$ signal.

5.5 Evaluation

We compared the quality of an analysis-synthesis system using the original $\mathscr{S}_{1,3}$ method, $\mathscr{S}_{2,3}$ method, and the $\mathscr{S}_{2,5}$. Frequency- and amplitude-modulated sinusoids were synthesized and then analysed frame-by-frame using the DDM to estimate their initial phase (amplitude), frequency (amplitude slope), and frequency-modulation (amplitude-modulation). Afterwards, the signals were resynthesized using the estimated parameters and compared to the original. We are interested in seeing in what cases higher-order phase and log-amplitude polynomials will improve the accuracy of synthesis.

5.5.1 Evaluation on a sinusoid of cubic phase and quartic log-amplitude

The initial evaluation illustrates how higher-order phase and log-amplitude polynomials will not necessarily improve the quality of synthesis if the underlying phase and log-amplitude

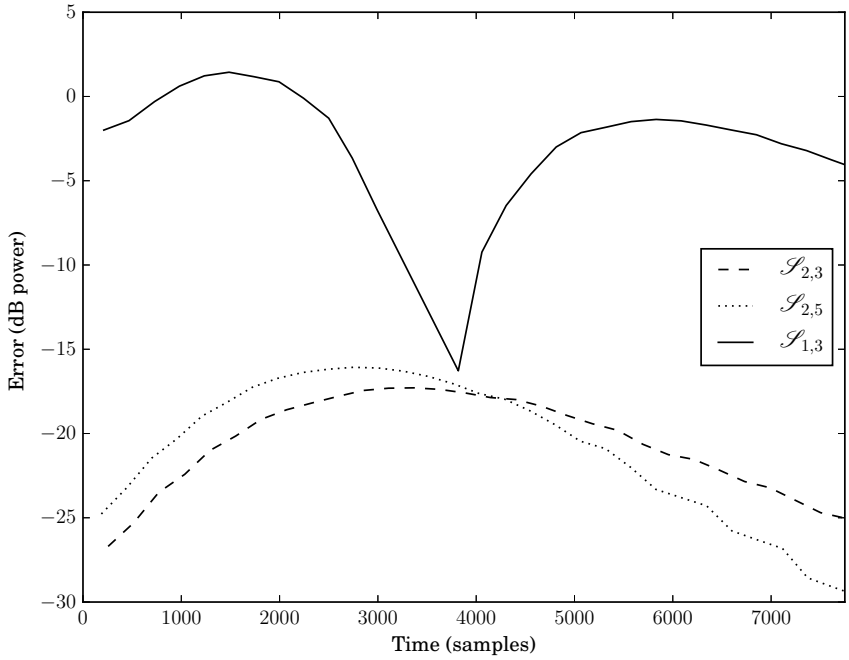

Fig. 5.2: $\mathscr{T}_{3,4}$ **vs.** $\mathscr{S}_{1,3}$**,** $\mathscr{S}_{2,3}$ **and** $\mathscr{S}_{2,5}$**: Upper error bound.** The power of the error when subtracting the original signal from the estimated signal. The local upper bound on the error was produced by connecting the local maxima in the error data.

functions are a polynomial of lower order than the polynomials used for synthesis. As we will see, the estimated phase and log-amplitude functions suffer from "overfitting".

The synthesized signal has 3 frequency break-points and an initial phase, therefore its phase function can be interpolated by a cubic polynomial

$$x_\phi(n) = \exp\left(g_3 n^3 + g_2 n^2 + g_1 n + g_0\right)$$

the frequency break-points are summarized in Table 5.1. The initial phase is 0 radians.

Table 5.1

Time (seconds)	0	0.25	0.5
Frequency (Hz)	100	200	100

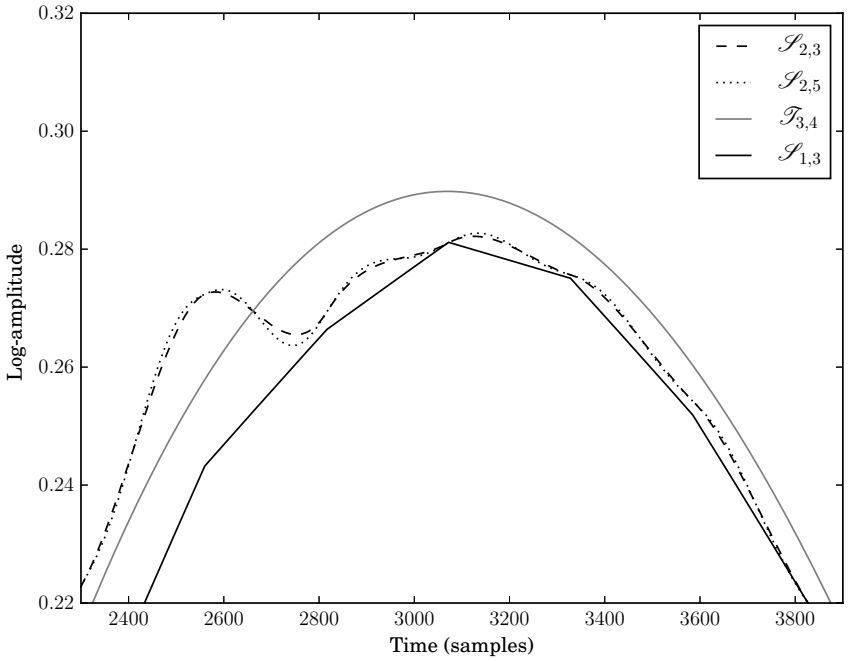

Fig. 5.3: $\mathscr{T}_{3,4}$**,** $\mathscr{S}_{1,3}$**,** $\mathscr{S}_{2,3}$ **and** $\mathscr{S}_{2,5}$**: Log-amplitude functions.** This compares the original log-amplitude function with the interpolated log-amplitude functions. The log-amplitude functions are considered because these are the real part of the polynomial exponents in the complex sinusoid model.

A quartic polynomial is used for the log-amplitude function

$$x_\mu(n) = \exp\left(h_4 n^4 + h_3 n^3 + h_2 n^2 + h_1 n + h_0\right)$$

and its amplitude break-points are summarized in Table 5.2. This will be referred to as the $\mathscr{T}_{3,4}$ signal.

Table 5.2

Time (seconds)	0	0.1	0.3	0.5
Amplitude (dB)	-10	0	0	-10

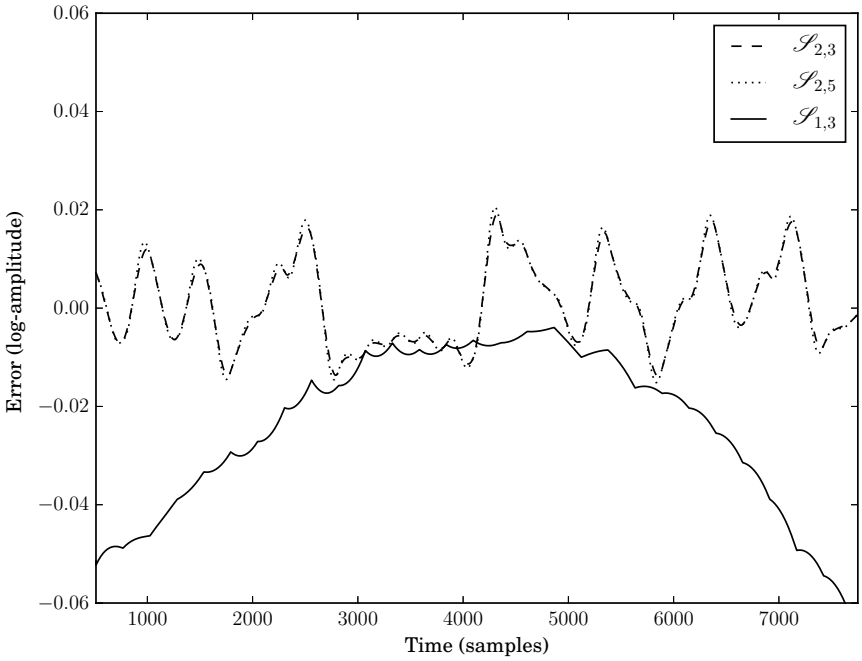

Fig. 5.4: $\mathscr{T}_{3,4}$ **vs.** $\mathscr{S}_{1,3}$**,** $\mathscr{S}_{2,3}$ **and** $\mathscr{S}_{2,5}$**: Log-amplitude function error.** This shows the error of the interpolated log-amplitude functions when compared with the original log-amplitude function for the three proposed methods.

These polynomials are chosen because their orders are greater than or equal to the order of the synthesis model in the $\mathscr{S}_{2,3}$ system and less than the order of the synthesis model in the $\mathscr{S}_{2,5}$ system.

The signal was sampled with a sampling rate of 16000 Hz and was analysed every 256 samples with an analysis window of length 1024 samples. For the DDM method, the $\mathcal{C}^1$ 4-Term Blackman-Harris was used (see Section 3.5).

The results of the evaluation are presented in Figures 5.1 through 5.5. In Figure 5.1 we see informally that the synthesis quality is good for all model orders even though the $\mathscr{S}_{1,3}$ model assumes linear phase in its analysis. Figure 5.2 shows how accurately the different model orders reconstruct the original signal. We see that the $\mathscr{S}_{2,5}$, although of higher-order, is not systemically a better interpolator of the true underlying signal. Figure 5.3 shows the estimated log-amplitude functions along with the original and Figure 5.3 their respective errors in approximating the true log-amplitude function. Finally, Figure 5.5 shows that

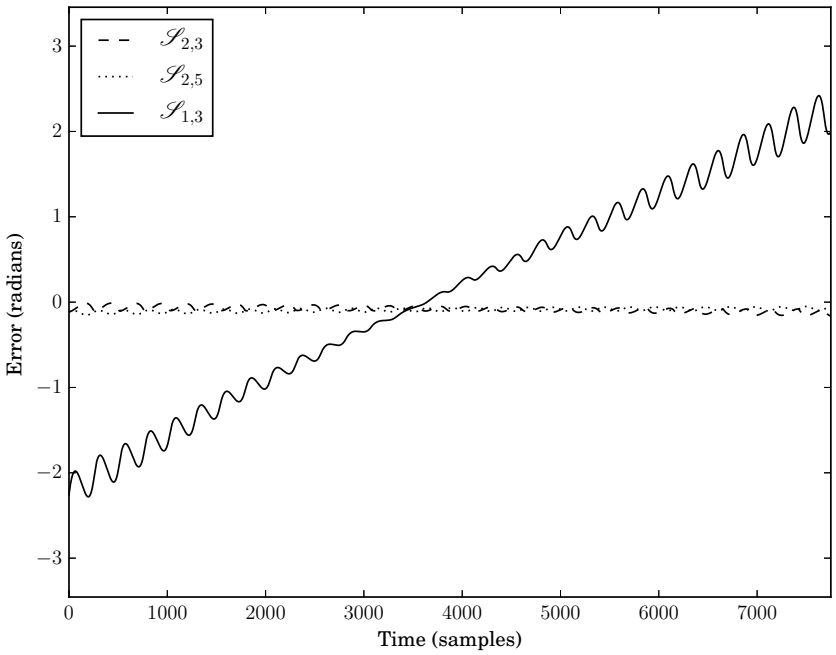

Fig. 5.5: $\mathscr{T}_{3,4}$ **vs.** $\mathscr{S}_{1,3}$**,** $\mathscr{S}_{2,3}$ **and** $\mathscr{S}_{2,5}$**: Phase function error.** This compares the theoretical phase function with the interpolated phase functions. The phase functions are considered because these are the imaginary part of the polynomial exponents in the complex sinusoidal model. The errors are "wrapped" to lie between $-\pi$ and π. The errors stem from both the estimation of the phase and the interpolation of phase between analysis points. As a frequency modulated sinusoid is considered, it is not surprising that the stationary frequency assumption of the $\mathscr{S}_{1,3}$ model exhibits the most errors.

the models incorporating the non-stationary assumption in their analysis perform better at approximating the true phase function.

5.5.2 Evaluation on sinusoid of exponential phase

The previous evaluation of this analysis-synthesis system was on a sinusoid with small-order polynomial phase and log-amplitude. In the cases where we observed overfitting, the polynomial used for synthesis was of higher-order than the true underlying one — the interpolating polynomials were more times differentiable than the true polynomial. We propose evaluating the system on an infinitely differentiable and analytic phase function. The rationale behind this stems from the definition of an analytic function: one whose

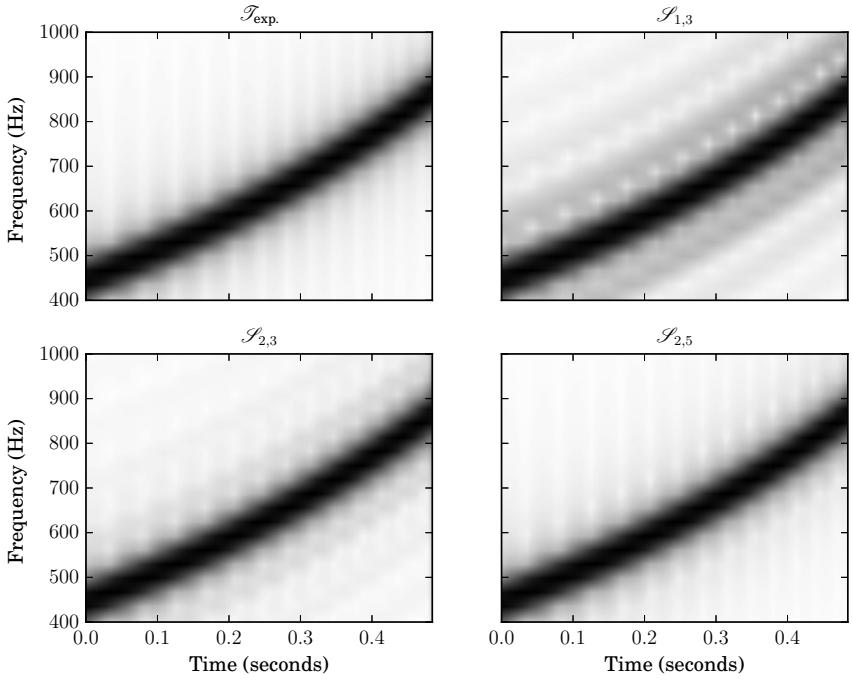

Fig. 5.6: **Spectrograms of $\mathscr{T}_{\text{exp.}}$, $\mathscr{S}_{1,3}$, $\mathscr{S}_{2,3}$ and $\mathscr{S}_{2,5}$.** Spectrograms of the true signal and estimated signals for the exponential phase signal.

power series representation (a polynomial) converges to the function as the number of terms approaches infinity. What this means is, in the region of convergence, the larger the number of terms in the approximating polynomial, the better the approximation to the true underlying function. The exponential function

$$y = \exp(x), x, y \in \mathbb{R}$$

is one such function whose power series is

$$\exp(x) = \sum_{n=0}^{\infty} \frac{x^n}{n!}$$

To find the radius of convergence, we use the ratio test

$$\lim_{n\to\infty} \frac{|1/n!|}{|1/(n+1)!|} = \lim_{n\to\infty} n = \infty$$

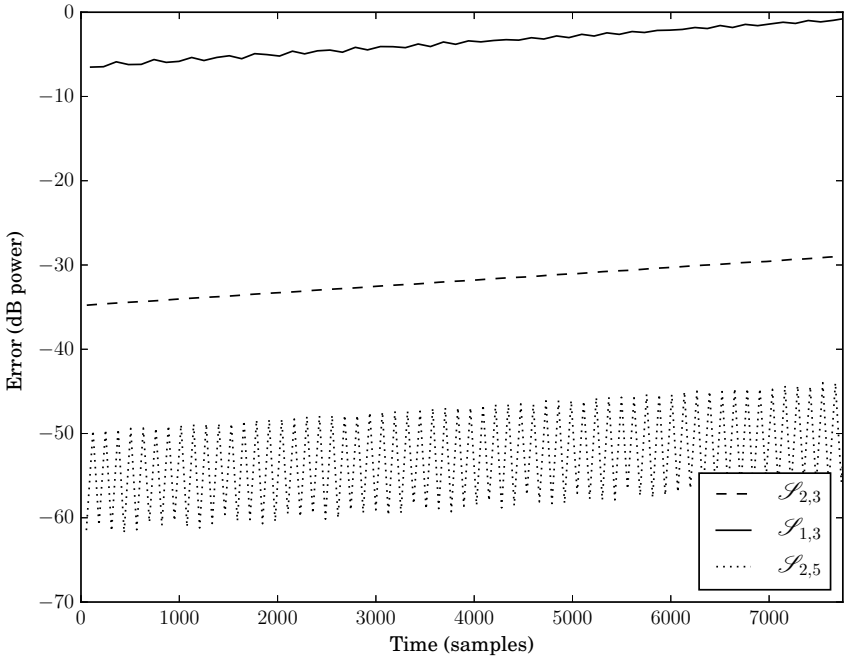

Fig. 5.7: $\mathscr{T}_{\text{exp.}}$ **vs.** $\mathscr{S}_{1,3}$, $\mathscr{S}_{2,3}$ **and** $\mathscr{S}_{2,5}$**: Upper error bound.** The power of the error when subtracting the original signal from the estimated signal for the signals of exponential phase. The local upper bound on the error was produced by connecting the local maxima in the error data.

i.e., the power series of the exponential function converges everywhere and so using more terms of its power series will improve its approximation for any $x \in \mathbb{R}$.

The exponential function arises in music. In 12-tone equal temperament tuning, to find the frequency f_1 of a pitch b-semitones away from the frequency f_0 we compute

$$f_1 = f_0 2^{\frac{b}{12}} = f_0 \exp(\log(2)\frac{b}{12})$$

So a linear transition from pitch b_0 to b_1 (in semitones) is an exponential change in frequency. This could be observed in recordings of performances of the *portamento* gesture.

We synthesize a sinusoid of length N samples with exponential phase and use the same analysis system as in Section 5.5.1 to evaluate the synthesis accuracy for piece-wise

interpolating polynomials of cubic and quartic order for phase. The signal x is defined

$$x(n) = \exp(j\frac{2\pi f_0}{c_1}\exp(c_1 n + c_0))$$

with

$$c_0 = \log(2)\frac{b_0}{12}$$

and

$$c_1 = \log(2)\frac{b_1}{12N}$$

i.e., a signal starting at pitch b_0 with frequency f_0 and arriving at pitch b_1 in N samples. We keep the amplitude of the signal constant in this evaluation as we are interested in the accuracy of the phase reconstruction. The same procedure as in Section 5.5.1 is used to estimate the parameters of piece-wise interpolating phase polynomials. We can see in Figure 5.7 that the resynthesis accuracy is greater for higher-order polynomials. Spectrograms of the original and estimated signals are plotted in Figure 5.6.

5.6 Conclusion

5.6.1 Polynomial phase and log-amplitude function

Out of the three proposed methods it appears that the modified cubic interpolation method works superiorly for the signal model considered. We observe overfitting by the higher-order quintic model in Figure 5.3, compromising the accuracy of resynthesis. Even the proposed cubic model shows some overfitting in this case. This is consistent with the results of [15]. From Figure 5.5 it is clear that the DDM-based methods provide superior estimation of the phase function — this is not the case for the log-amplitude function. Depending on the underlying signal, perhaps better results can be obtained by postulating a lower-order log-amplitude function and higher-order phase function. The possibility of errors arising from numerical accuracy when evaluating the quintic polynomials has been ruled out. We evaluated these polynomials using an implementation of Horner's method that keeps track of the error bound [22, p. 95]: the errors are negligible, see Figure 5.8 for the results.

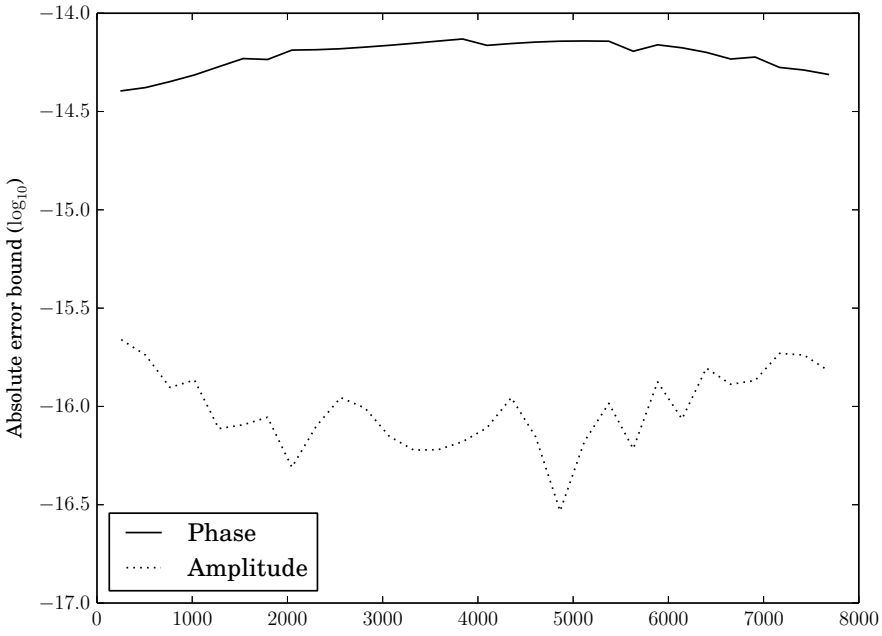

Fig. 5.8: $\mathscr{S}_{2,5}$ **evaluation error bound.** The error bound in evaluating the quintic log-amplitude and phase polynomials of $\mathscr{S}_{2,5}$ using Horner's method. This plot was produced by plotting only the local maxima of the error bound data in order to reduce the plot's range.

5.6.2 Exponential phase function

The quintic interpolation $\mathscr{S}_{2,5}$, the polynomial of highest order, performs the most accurate resynthesis. This is consistent with the analytic property of the exponential function and an encouraging result as it suggests analytic phase functions can be approximated with arbitrary accuracy simply by increasing the order of the interpolating polynomials. Many models of musical gestures involve such functions, apart from the portamento gesture modeled by an exponential phase function, vibrato can be modeled as a sinusoid with sinusoidal phase [32]. The DDM-based analysis system combined with the higher-order polynomial phase and log-amplitude synthesis system presented here allows for accurate modeling of these gestures.