

Figure 3.4: The errors of period estimation using the periodogram method. The tests were conducted on a set of ionograms from orbits 3836–4000. The left chart displays the absolute error E_A with the 0.25-quartile = 53 %, median = 68 % and the 0.75-quartile = 79 %. The right chart shows the period corrected error E_P with the 0.25-quartile = 0.8 %, median = 4.5 % and 0.75-quartile = 7.6 %.

To sum it up, the periodogram-based period estimation is not perfect for our task. On the other hand, its results can be utilized by other methods to scale down the search space. One more drawback is it only searches a fixed set of periods to try out.

3.3.2 Sine wave least squares fitting

The idea of fitting a sine wave to the peaks leads us to another algorithm. While the classical least squares fitting methods are suitable only for linear models to fit, there are also ways to do a similar fitting with nonlinear functions. One of these algorithms is the Levenberg-Marquardt algorithm introduced in [31].

We start with defining the model to fit, which is

$$E(y) = f(x; \beta_1, \dots, \beta_k) = f(x, \boldsymbol{\beta}) \quad (3.5)$$

where $E(y)$ is the expected value. We simplify our case to a single variable, whereas [31, p. 431] describes the problem for any number of variables. Denote the peak values we have as (Y_i, X_i) , $i = 1, \dots, N_0$. Then the least squares minimization problem is to find an assignment of β that minimizes the error function

$$\Phi = \sum_{i=1}^n [Y_i - \hat{Y}_i]^2 = \|\mathbf{Y} - \hat{\mathbf{Y}}\|^2, \quad (3.6)$$

where \hat{Y}_i are the estimated values given by $E(y)$ [31, p. 431].

For the algorithm we need the Jacobian matrix

$$J_j = \frac{\partial f(x)}{\partial b_j}, \quad j = 1, \dots, k, \quad (3.7)$$

where b_j are the current estimates of β_j . We also need the vector

$$\mathbf{g} = \left((Y - f(x)) \frac{\partial f(x)}{\partial b_j} \right) = J^T(\mathbf{Y} - f(\mathbf{b})), \quad j = 1, \dots, k. \quad (3.8)$$

Both these notions are defined in [31, p. 433].

With these definitions we can describe the iterative Levenberg-Marquardt algorithm. We select the damping factor $0 < \lambda^{(1)} \leq 1$ (some guides on its selection are provided in [31, p. 437]). The initial parameter vector $\mathbf{b}^{(1)}$ has to be guessed and passed to the algorithm. In every iteration r we solve the set of linear equations

$$(J^{(r)T} J^{(r)} + \lambda^{(r)} I) \boldsymbol{\delta}^{(r)} = \mathbf{g}^{(r)} \quad (3.9)$$

for the difference $\boldsymbol{\delta}^{(r)}$ which is used in the next iteration to construct $\mathbf{b}^{(r+1)} = \mathbf{b}^{(r)} + \boldsymbol{\delta}^{(r)}$. Using $\mathbf{b}^{(r+1)}$ we can update $\Phi^{(r+1)}$. Then we choose new damping factor $\lambda^{(r+1)}$ and proceed to the next iteration (as described in [31, pp. 437–428]). The iterations are repeated until either a maximum number of iterations is exceeded, if Φ converges to 0 or if we encounter a local minimum of Φ .

With a good initial guess, the algorithm converges to the global minimum of Φ . However, the algorithm may often lodge in a local minimum of Φ instead. But by good strategy for choosing λ it should be robust enough [31, pp. 437–428].

In our experiments we used the implementation of the Levenberg-Marquardt (LM) algorithm provided by the library Apache Commons Math 3 [2]. In addition to the above described algorithm, it also supports weighing the data (by simply incorporating the weights in the error function), which is exactly what we want. It also contains a procedure for getting the initial guess on parameters based on computation of several definite integrals and using linear least squares method. This method is described in the source code [3].

In our approach we try to fit a sine wave with unit amplitude against the peaks with unit height. In order to magnify the effect of the weights, we also make square roots of all of them and then normalize them. This way more mid-value weights can affect the fitting, which usually helps, as we have observed.

Since we have both minimum and maximum constraints on the frequency, we have to tell these constraints to the LM algorithm. As the Apache Commons implementation does not support imposing constraints to the optimized variables, we simulate such constraints by returning ± 1 derivative in direction ω for values

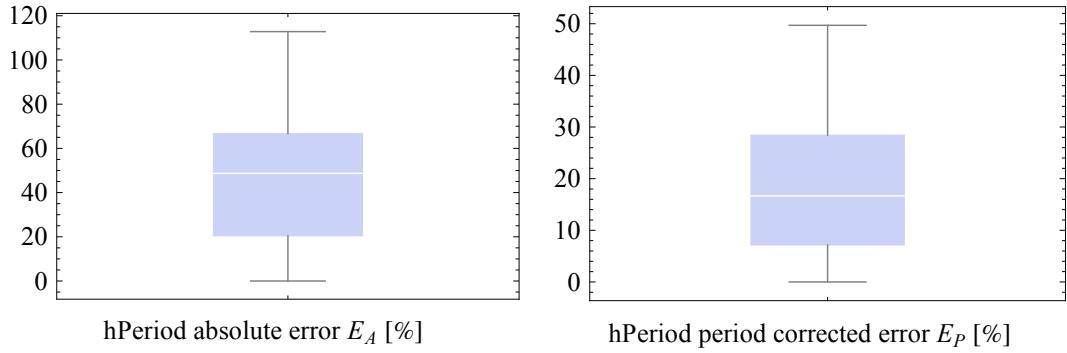


Figure 3.5: The errors of period estimation using the sine wave fitting method. The tests were conducted on a set of ionograms from orbits 3836–4000. The left chart displays the absolute error E_A with the 0.25-quartile = 20.4 %, median = 48.7 % and the 0.75-quartile = 66.7 %. The right chart shows the period corrected error E_P with the 0.25-quartile = 7.1 %, median = 16.7 % and 0.75-quartile = 28.4 %.

outside our boundaries. This seems to be sufficient.

Our implementation can be found on the attached CD in folder `programs/detector-summing`. See Appendix 1 for information on how to run it. The results of the detection can be found also on the CD in subfolders of folder `data/` – they are files named `TRACE_(orbit number)_SUM_FITTING.XML`.

For quality assessment we use the same two metrics as for periodograms – the absolute error E_A and the period corrected error E_P , since the fitting algorithm also suffers from returning “1/integer” multiples of the real period. The reason is the same as for periodograms – with some false peaks, a lower error can be achieved by doubling (or tripling, ...) the frequency.

Final results of our tests are shown in Figure 3.5. Similarly to the periodograms, the absolute error E_A with median 49 % renders the results to be worthless. In contrast to the periodograms, however, the period corrected error with median 17 % and 0.75-quartile 28 % also cannot be treated as a good result. From these results it seems the curve fitting method is not a good solution to our problem. We think it may be due to the false peaks (although they should have small weights). Average computation time (covering also the GE detection detailed in Section 3.5) is about 3 s per ionogram of size 1,012×506 px. It is really slow compared to the other methods, mainly due to the iterative base of this method.

3.3.3 Period averaging

The last proposed way to determine the period from peaks and their weights is a method we call “period averaging”. It utilizes a completely different approach than the previous methods.

On the input we again have the peaks with their weights. We first calculate the “neighbor distances” – distances of the peaks being just one next to the other. This is the base for our approach. Each such distance gets a weight assigned that is equal to the weight of its right peak. In the ideal case without false peaks and with peaks for every harmonic line, simply computing a weighted average of these distances should give us the correct period. As we do not work with ideal data, we devised a method how to overcome the difficulties with false peaks and missing peaks for harmonic lines.

When we would just do a weighted average of the distances, a few missing harmonic lines could completely mislead the result (as well as the false peaks). E.g. we can take a period of 2 units. Then having 4 harmonics spaced correctly at the 2 units distance would give the correct period of 2 units. If these 4 lines were followed by 2 more lines spaced at 4 units (as if they were 4 lines and 2 of them disappeared, which is common in ionograms), then the average period would be $(4 \cdot 2 + 2 \cdot 4) / 6 = 8 / 3 \doteq 2.67$, which is an error of 33.3 %. This is unacceptable. Even the weighing does not help in this case because regular harmonic lines should have large weights (opposite to the false peaks which should have low weights and should not influence the result much).

We investigated the ionograms and we deduced that almost never more than a third of the harmonic lines is missing in the left half of ionograms (which we use in the rows/columns summing algorithm). So if we rule out (rounded off) 35 % of the longest distances, we should not remove many correctly spaced peaks. On the other hand, all the long distances caused by missing harmonics should be eliminated. If there are no harmonics missing, we just eliminate some distances corresponding to the correctly spaced peaks, but since no harmonics are missing there must be several other correct distances left with high weights. As said earlier, false peaks should be smoothed out by their low weights.

So, with the resting 65 % of distances, we re-normalize their weights. And with these weights we just evaluate the weighted arithmetic average of the distances and the harmonics period is found.

Our implementation can be found on the attached CD in folder `programs/detector-summing`. See Appendix 1 for information on how to run it. The results of the detection can be found also on the CD in subfolders of folder `data/` – they are files named `TRACE_(orbit number)_SUM_QUANTILE.XML`.

For quality evaluation we again use the absolute error E_A and the period corrected error E_P . We expect the difference between E_A and E_P to be

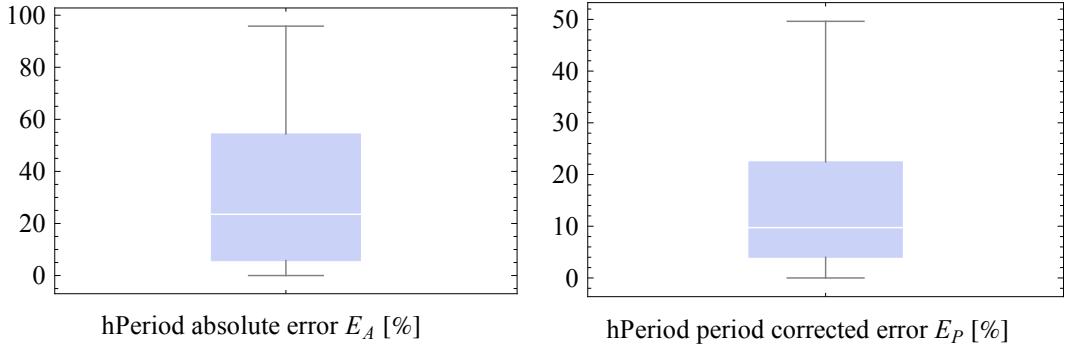


Figure 3.6: The errors of period estimation using the period averaging method. The tests were conducted on a set of ionograms from orbits 3836–4000. The left chart displays the absolute error E_A with the 0.25-quartile = 5.7%, median = 23.5% and the 0.75-quartile = 54 %. The right chart shows the period corrected error E_P with the 0.25-quartile = 4 %, median = 9.9 % and 0.75-quartile = 22.4 %.

significantly lower than in the previous two methods (especially E_A should be lower). That is since the period averaging should not tend the yield fractions of the real periods.

The outcomes of our tests are shown in Figure 3.6. As expected, the absolute error E_A is lower, with median 23.5 % being the best of the methods tested so far. What is a small disappointment for us is a relatively high E_P – with median 9.73 % and 0.75-quartile 22.5 % it is not the best one. Average computation time (covering also the GE detection detailed in Section 3.5) is about 0.56 s per ionogram of size $1,012 \times 506$ px.

3.3.4 Combining the methods

Taking into account the properties of the periodogram-based and average-based methods, there emerges a new, rather powerful and robust method for period estimation. As the periodogram-based method is good at determining the “period corrected” period (denote it p_p), the average-based method well estimates the “magnitude” of the period (denoted p_a). By magnitude we mean that the estimated period is not a fraction of the manually measured period p_m (it is near this value, but still with a rather large error as mentioned in the previous section).

So we have the period p_p about which we know that there exists an integer coefficient c such that $c \cdot p_p \doteq p_m$. And we also know $p_a \doteq p_m$ but with a larger error. So we propose to compute the integer c as $\hat{c} = \text{round}(p_a/p_p)$ and then determine

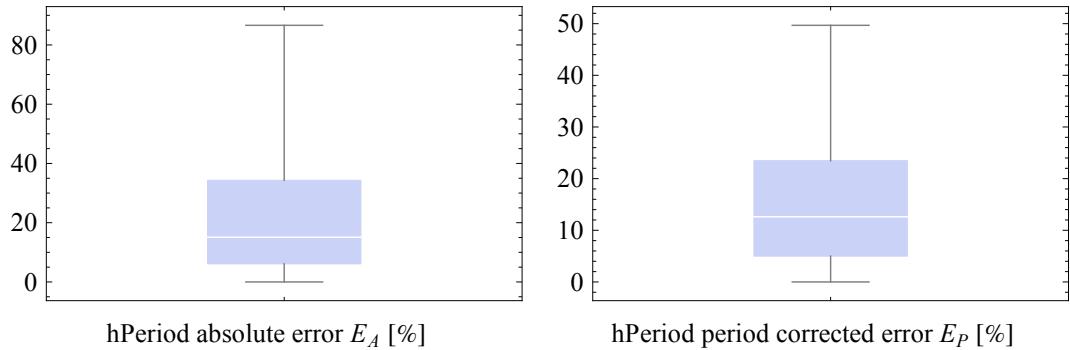


Figure 3.7: The errors of period estimation using the combined periodogram- and averaging-based method. The tests run on a set of ionograms from orbits 3836–4000. The left chart displays the absolute error E_A with the 0.25-quartile = 6.1 %, median = 14.3 % and the 0.75-quartile = 34.6 %. The right chart shows the period corrected error E_P with the 0.25-quartile = 5.2 %, median = 11.7 % and 0.75-quartile = 22.9 %.

the period as $\hat{c} \cdot p_p$.

Because the periodogram-based method uses only a constant set of periods to try, we would be limited by these values, which could lead to unnecessary errors in the result. For this reason we weight both periods into the final result $\hat{p} = 0.5 \cdot p_a + 0.5 \cdot \hat{c} \cdot p_p$.

Our implementation can be found on the attached CD in folder `programs/detector-summing`. See Appendix 1 for information on how to run it. The results of the detection can be found also on the CD in subfolders of folder `data/` – they are files named `TRACE_(orbit number)_SUM_COMBINED.XML`.

The evaluation again consisted of expressing the absolute error E_A and the period corrected error E_P . Our expectation is that the difference between E_A and E_P is insignificant, because there should remain almost no estimates needing the “period correction”. Also E_A should be the lowest observed so long.

The results of our tests are presented in Figure 3.7. Following our expectations, the absolute error E_A is low; with median 14 % it is the winner of this chapter. On the other side, 0.75-quartile 35 % is not as good as we expected. The small difference between E_A and E_P is validated – the median of E_P is 11.7 % (which means that there still remained some results influenced by period correction, which is interesting). Average computation time (covering also the GE detection detailed in Section 3.5) is about 0.60 s per ionogram of size 1,012×506 px.

3.4 Vertical periods detection

All the results presented above refer to the horizontal periods (EPOHs). Since we have no reference data for the vertical periods, we could not conduct such large statistical tests. So we just checked the estimated results with a small set of vertical periods extracted manually by ourselves. The result is not very good. The vertical periods used to be rather large (even a third of height of the ionogram and more), so there are not many echoes and the noise effects become significant.

Also, the often stronger (and mainly longer) EPOHs behave as strong noise in this case – they can even cause filtering out some vertical lines. As the EPOH-caused noise near the top part of the image (where EPOHs are the strongest) may reach rather high values, the real vertical peaks near bottom may fall under the 60 % level used for peaks selection.

To conclude, we can say the detected vertical periods do not have significant relation to the real periods and their detection fails. As long as we do not have some larger set of verification data, solving this problem looks, however, impractical.

3.5 Ground echo detection

This last part of chapter about lateral histograms usage is dedicated to ground echo detection. With the knowledge of the current spacecraft height above surface (computed from the ephemeris data), we can predict the vertical position of ground echoes.

The first thing we do is decide whether a GE is present. This has two steps. The first is rather simple and only takes determining if the height above surface is less than the maximum detectable height. This boundary is limited by the 7.32 ms height of the vertical axis of ionograms. As radio waves spread approximately with the speed of light near Mars, the height corresponding to time delay Δt is $c \cdot \Delta t / 2$ (the $/2$ for the signal path to surface and back, c stands for the speed of light in vacuum). Supplying the maximum time delay to this equation, the maximum height at which a GE may appear is 1,098 km.

The second decision step is based on row sums. We work only with the right half of the ionogram since this is where a GE may occur. We further divide the half of ionogram into two parts – the rows near the possible GE occurrence (up to 20 px under than the possible occurrence for a ionogram sized $1,012 \times 506$ px), and other rows. Computing mean values of the electric field density in these two groups gives us a very good guiding principle for telling if a GE is present or not. If the mean in the group near GE is more than 2 times higher than in the rest, we tell a GE is present. The threshold value 2 was determined experimentally

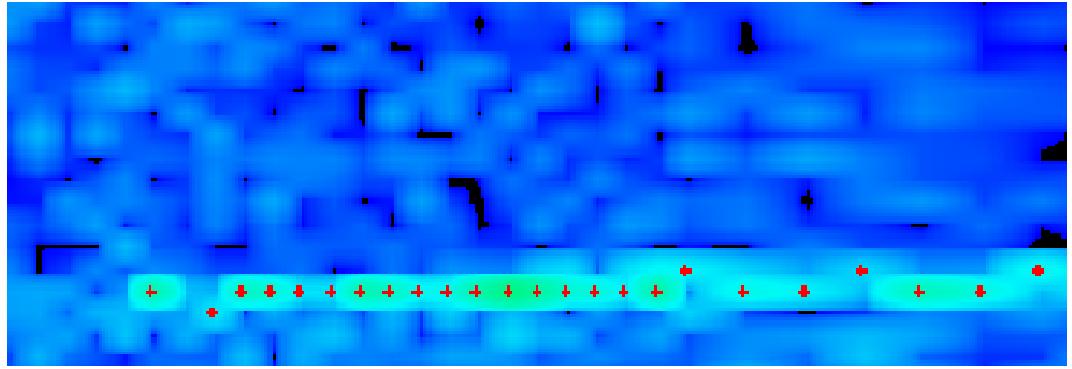


Figure 3.8: An example of detected ground echo (the red dots). A cut of frame nr. 46 from orbit 3874.

and seems to give good performance.

When we know we should look for a GE, we start at the right side of the ionogram and progress to its left (but we stop in the half). We begin on the row corresponding to the predicted GE occurrence. At every column we look at the values in the current row and 10 px under it and pick the highest of them. If this value is higher than the mean of rows near the possible GE as computed earlier, we consider it a part of the GE. If so, we also change the “actual row” to the row with the highest value before proceeding to the next column – this makes the algorithm a tracking algorithm able to track even the cusp that may occur.

The results of the algorithm as described slightly differ from what we have defined as a result of GE detection. We defined the result should be the top edge of the GE and this algorithm instead finds the part with the highest values, which is commonly the centerline of the echo. We do not consider this a big problem because it may be processed further using a simple postprocessing, which, however, needs deeper information on the properties of the edge. As for the shape of the detected echo, it is nearly the same as the shape of the top edge of the GE.

As stated in Section 2.1.2, we have not been provided with GE referential data. So the only assessment method available is visual comparison of a small set of detection results. As far as we can say, we have not found any ground echo with even a little bad shape, all the inspected ones precisely copied the shape of the ground echo. An example of a detected ground echo is given in Figure 3.8. We have also found no false negative detections (false positives may occur, though it is not frequent).

3.6 Summary of lateral histogram methods

We have presented a single method using the lateral histogram approach. This method may utilize one of the four proposed period estimation methods, of which the best is the combined method. As its error of estimation is about 14% we

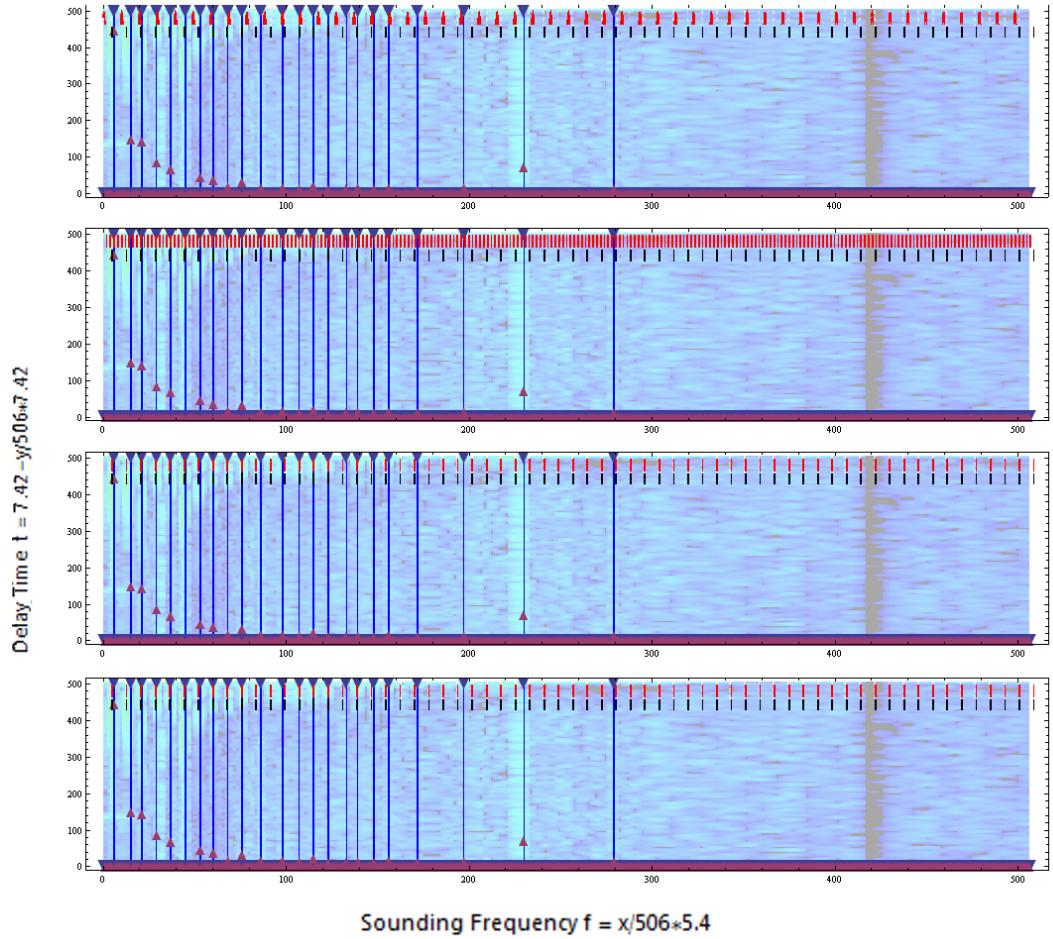


Figure 3.9: A comparison of periods detected by the presented methods. Red ticks are the detected period. Black ticks denote the manually acquired period. Blue triangles as well as the blue lines correspond to peaks detected in the column sums. Purple triangles (with the tip up) are at positions corresponding to peak weights (multiplied by two for better visibility). From the top: the periodogram method, sine wave fitting method, average period method and the combined method. The ionogram in background is frame 0 in orbit 3874. Axis labels are in pixels of the evenly sampled ionogram.

cannot say it is a perfect method (since manual tagging reaches a 1% error level), but it is definitely not worthless. All methods but sine wave fitting can be said to be relatively fast, with computation times about 0.6 s per ionogram. An example comparison of the detected periods is given in Figure 3.9.

We have also stated the the detection of the vertical period is a serious problem for this method. It needs further investigation to discover some really helpful noise-reducing techniques.

At last we have presented a very good way of determining the ground echoes in ionograms. It seems reliable as well as precise. We consider this as a good enough method for the precise measurements, although its detection error could not be validated due to the lack of reference data.

asd