Assignment 2
Feature Extraction for Song Recognition

Carl Holmqvist and Philip Stiff

October 3, 2017

1 Our features

Our feature extractor is based upon the frequency difference between time points in the melody. By checking the difference in Hz between two time points it calculates the number of semitones the tone has changed and lets that number be the feature value of the specific time point. Since this calculation is done in such a manner that the direction of the pitch change (higher or lower) is not preserved, a second feature is included that simply provides that information.

1.1 Definition

As stated in *https://en.wikipedia.org/wiki/Equal_temperament* the relation between 2 adjacent semitones in western music is a scaling factor of $\sqrt[12]{2} \approx 1.06$. By using this knowledge our extractor checks the previous and current pitches and simply multiplies the lower frequency value by this scaling factor until it becomes larger than the higher value and returns the number of times it had to perform the multiplication. To avoid returning the value 0 (since the lab assignment stated that discrete features should only return positive values) a 1 is added to the return value, resulting in the method effectively returning the value $semitoneDist + 1$.

To get the second feature, the direction of the change of pitch, we simply check which of the previous pitch and the current pitch is larger. The second feature returns values according to the following:

$$direction = \begin{cases} 1 & \text{if } currentfrequency < previousfrequency \\ 2 & \text{if } currentfrequency > previousfrequency \\ 3 & \text{otherwise} \end{cases}$$

The motivation behind having this feature is that the lab assignment told us to not use negative numbers for discrete distributions. Therefor we split up the original feature into these two features.

To ensure that we also have an upper bound to the possible discrete values returned by our extractor we do not allow frequencies higher than 20 000 Hz, those that are higher are simply set to 20 000. This results in the highest possible return value being 172 (the maximum number of semitone jumps + 1). Our extractor therefor returns values in the range $[1, 172]$ for the first feature and $[1, 3]$ for the second feature.

1.2 Discussion

Here follows a discussion of how our features handle the main requirements mentioned in the lab assignment.

- *Distinguish between different melodies, i.e., sequences of notes where the ratios between note frequencies may differ.* Different sequences of notes will consist if different lengths of semitone jumps between the notes, this is exactly what our features are recording.
- *Distinguishing between note sequences with the same pitch track, but where note or pause durations differ.* Our extractor records the length of each note/pause since it returns semitone distance 1 (meaning no change) when the frequency does not change, i.e. during a note or a pause.

- *Should be insensitive to transposition.* Except for the first semitone distance our features remain identical no matter how the pitch is transposed. See figure 6

- *In quiet segments, the pitch track is unreliable and may be influenced by background noises in your recordings. This should not affect the features too much, or how they perceive the relative pitches of two notes separated by a pause.* If the noise remains the same throughout the recording our features handles it by recording the jump into the noise and back from the noise, the difference in length between these jumps tells us the next note.

The intuition when looking at the plots of our feature capturing the semitone distance (figure 4) is that when the plot is 1 the pitch of the recording is constant, i.e. we are either in a note or in a pause. Since the pauses of the recordings plotted here are so short they are difficult to point out (they are "inside" the spikes in the plots). A spike in the plot indicates that there was a change of pitch. Based on this one can roughly say that in the plots of this feature each interval of 1:s is a note. The exact change of note is often hard to tell with the naked eye since the jumps to and from the noise are so high, but the information is there for the computer to see.

For our second feature, the direction of the semitone jump, the plots are not very informative to the eye. But we consider this to still be relevant information for the HMM.

Our features do not take the intensity of the recording into account. This results in us getting the same features for the same song being played at different volumes. But a downside to this is that it makes it possible to insert a noise with a low intensity to disturb the feature extraction. So if the noise is in the same frequency as a note our extractor would not notice the pause in the melody. By using this limitation in the features one could record two similar sounding songs that give very different features.

Another issue with our extractor is that it only works well when the noise in a recording is roughly constant. If the noise changes pitch during the recording this would cause the jumps from notes to noise and vice versa to differ in length even though the same note is being played.

2 Plots

Melody 1, 2 and our own are recordings of hummings of the Swedish national anthem while melody 3 is a recording of a humming of the Russian national anthem.

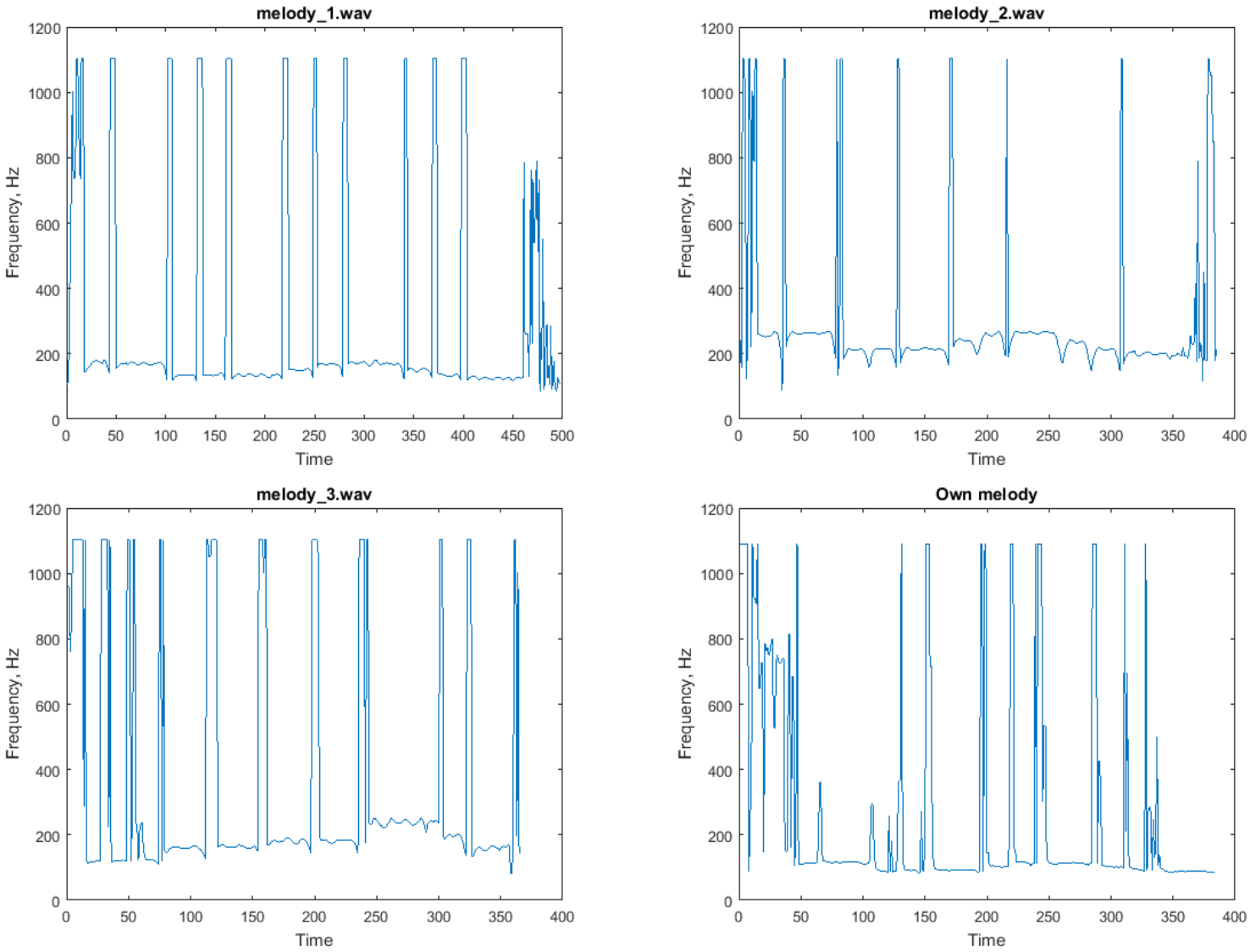

Figure 1: Frequency plots

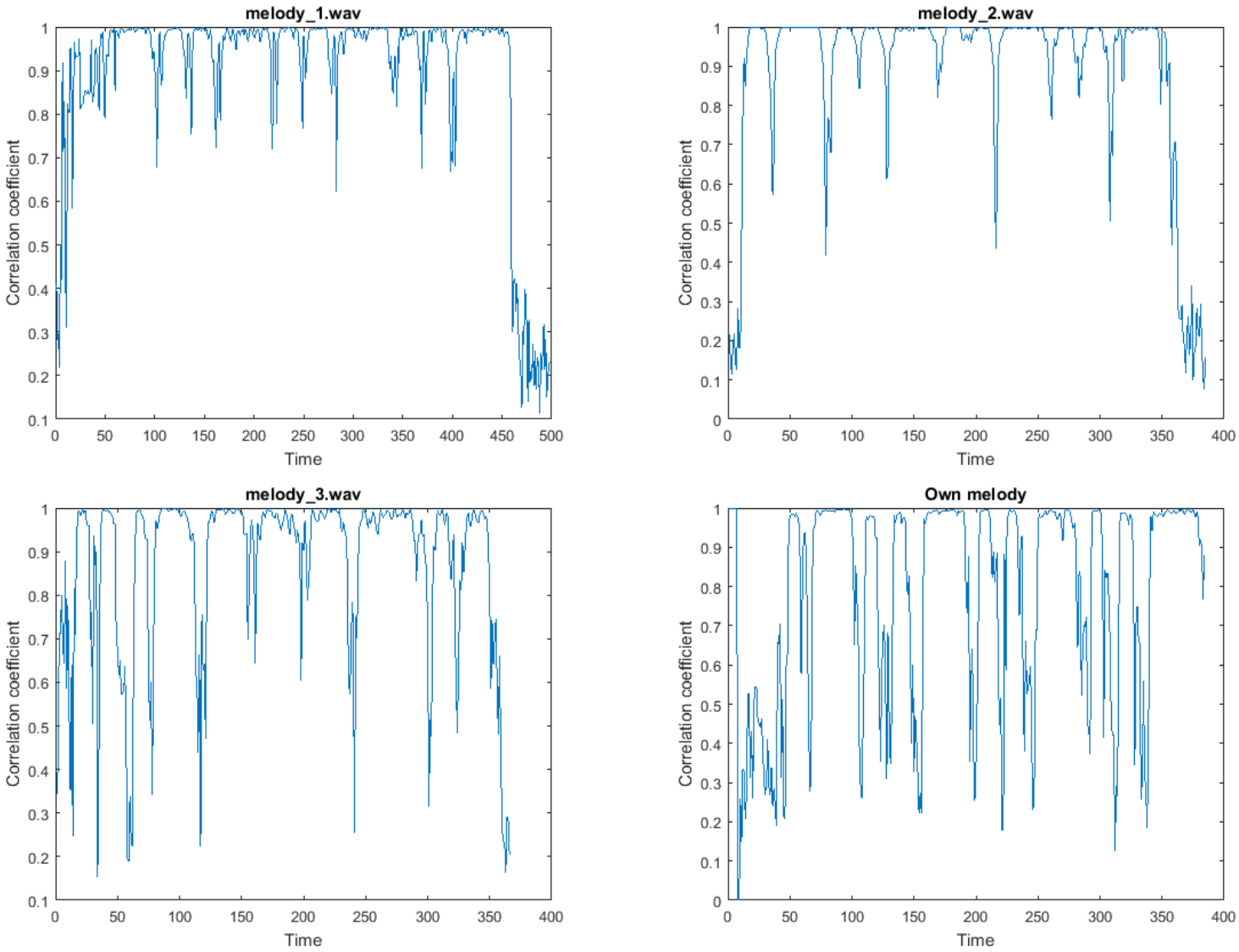

Figure 2: Correlation plots

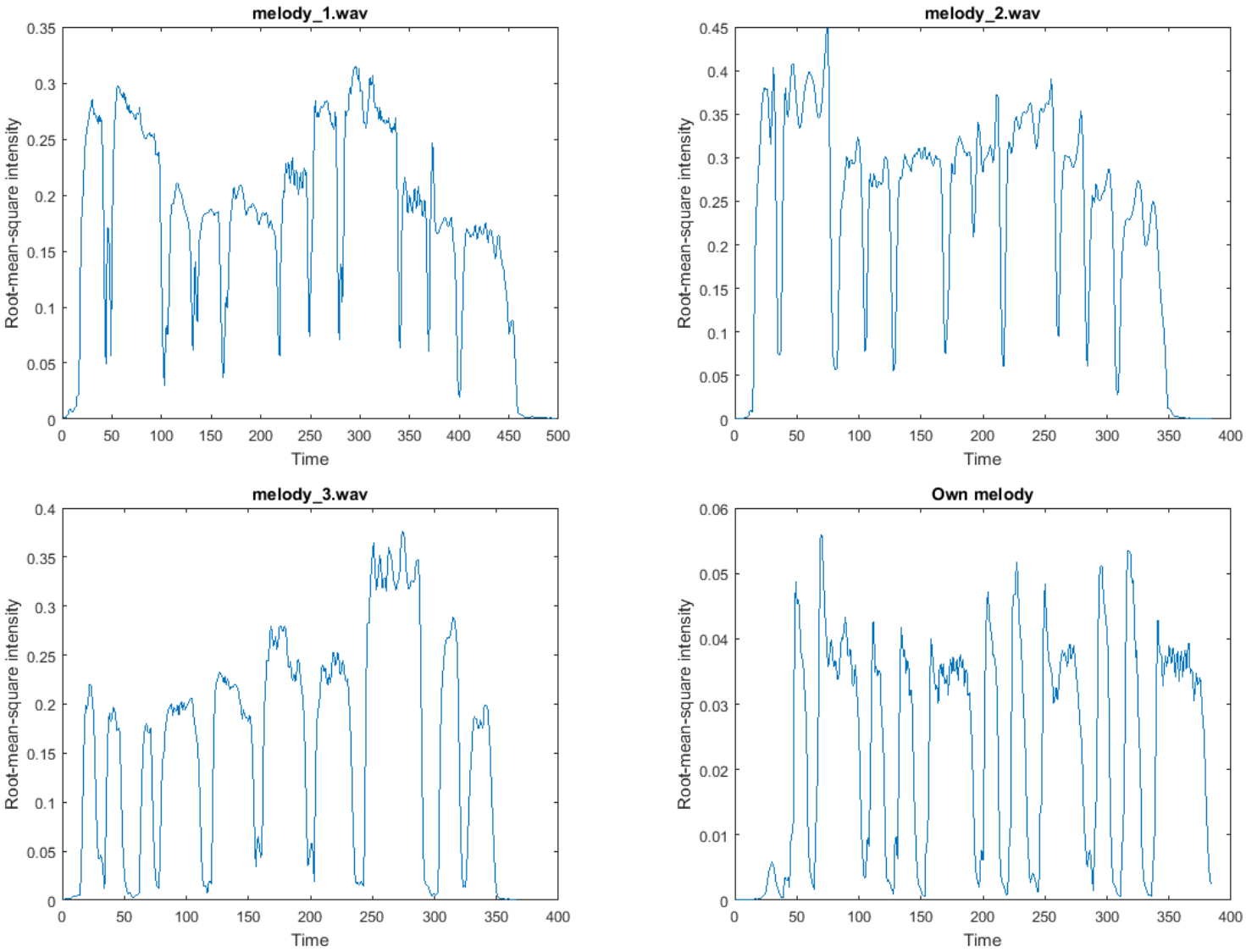

Figure 3: Intensity plots

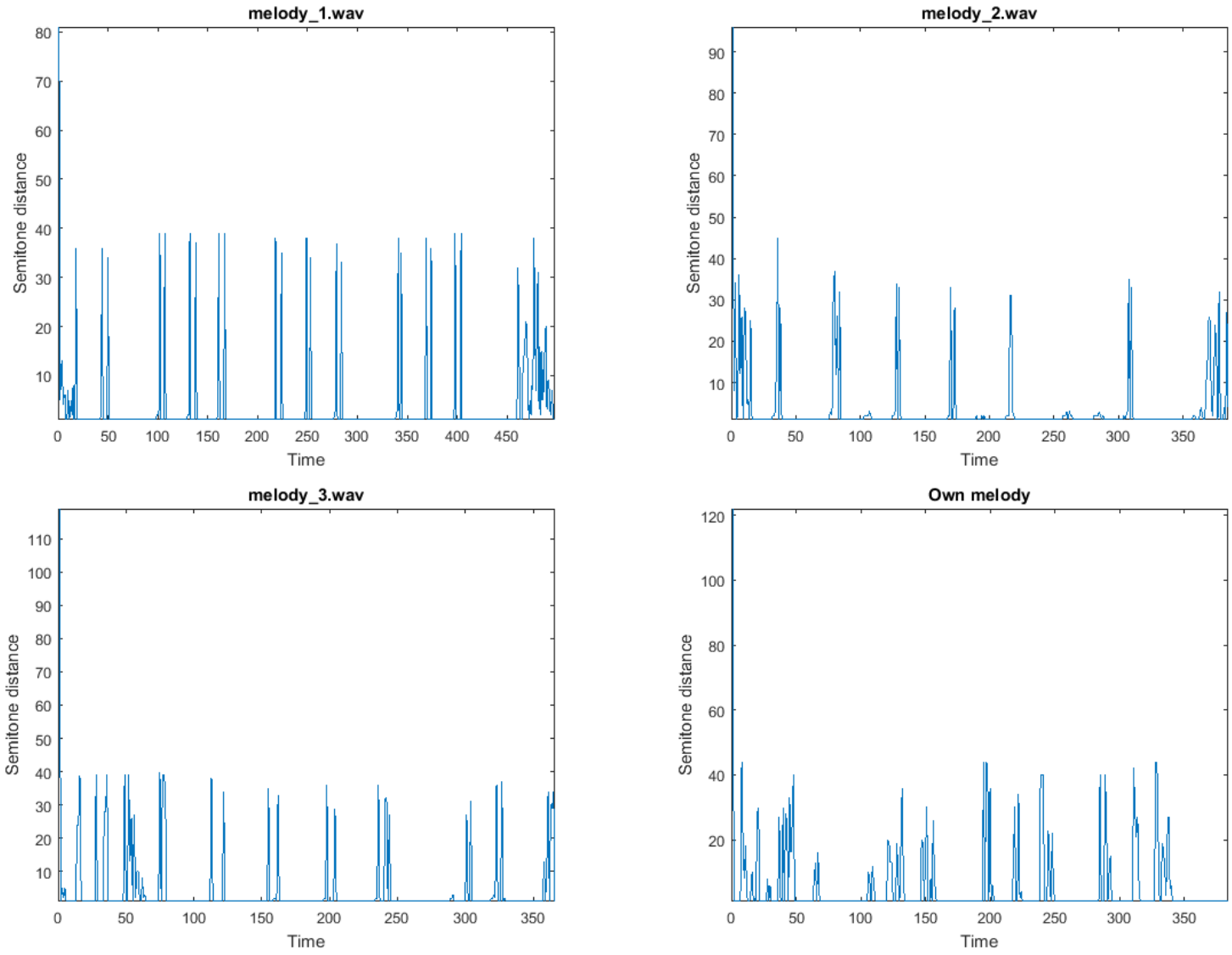

Figure 4: Semitone distance plots (our own feature)

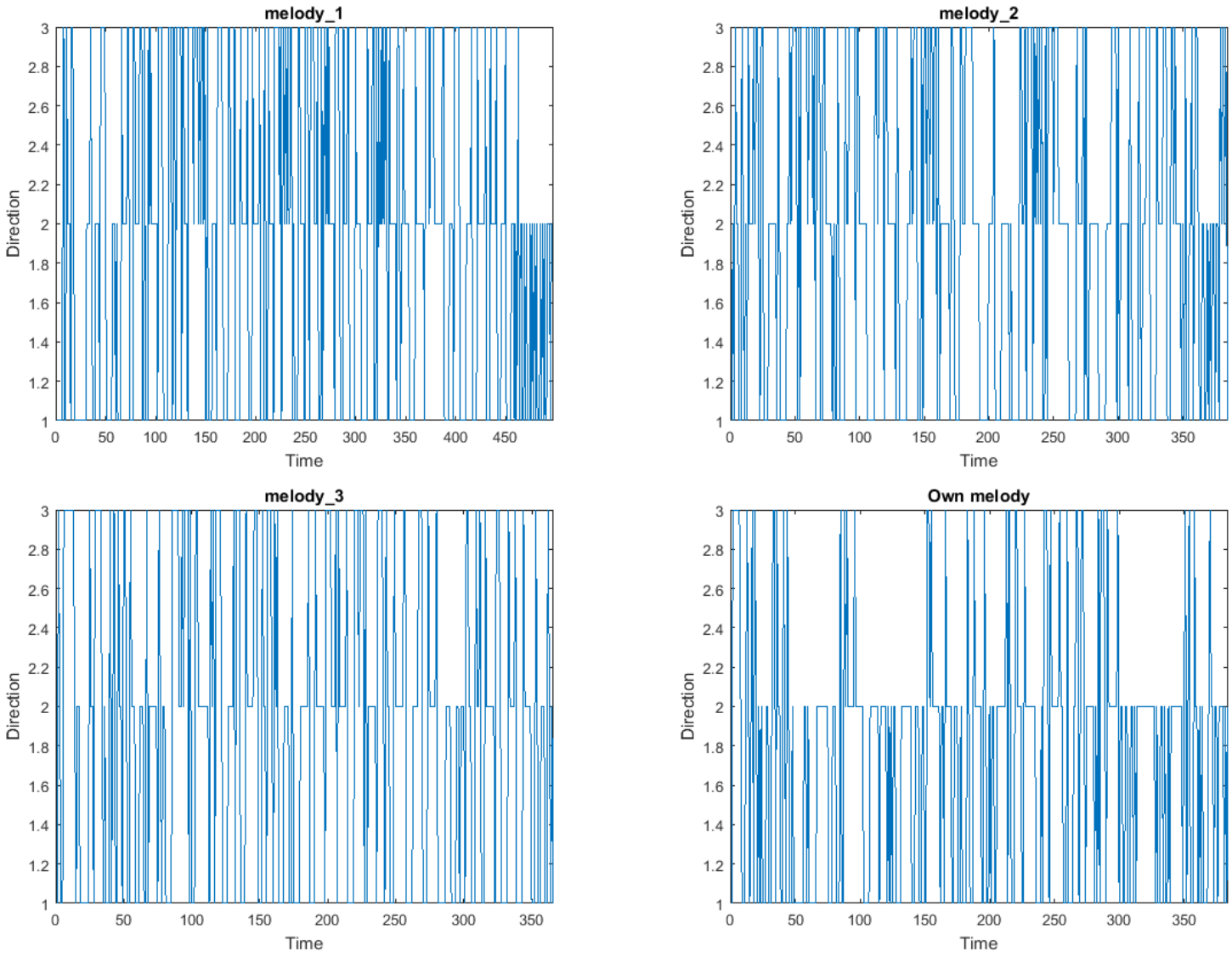

Figure 5: Plots of the direction of the semitone change (our own feature)

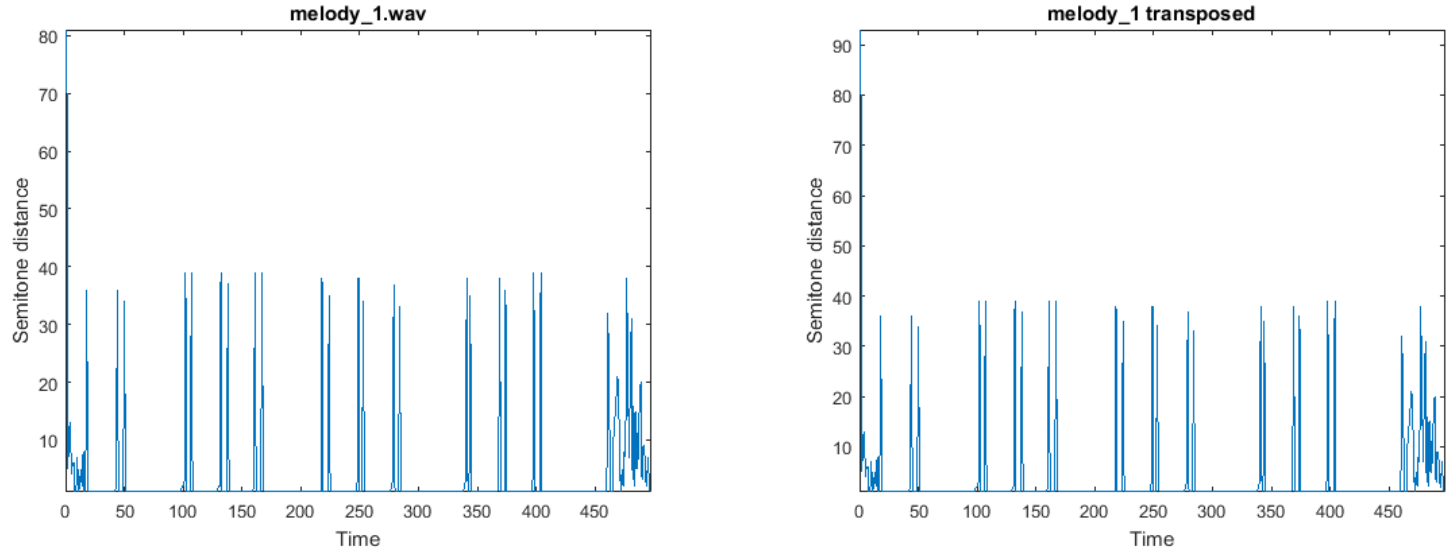

Figure 6: Plots of the semitone distance in normal pitch (left) and in transposed pitch (right)