Lecture 6: Griffin-Lim Algorithm

Mark Hasegawa-Johnson

ECE 417: Multimedia Signal Processing, Fall 2023

1. Review: STFT and ISTFT
2. Why is inverting a spectrogram difficult?
3. Griffin-Lim Algorithm: A vector-space representation
4. Griffin-Lim Algorithm: Signal example
5. Conclusions

Outline

1. Review: STFT and ISTFT
2. Why is inverting a spectrogram difficult?
3. Griffin-Lim Algorithm: A vector-space representation
4. Griffin-Lim Algorithm: Signal example
5. Conclusions

STFT and ISTFT

Let D be the window hop length, then the STFT can be written as

- **STFT:**

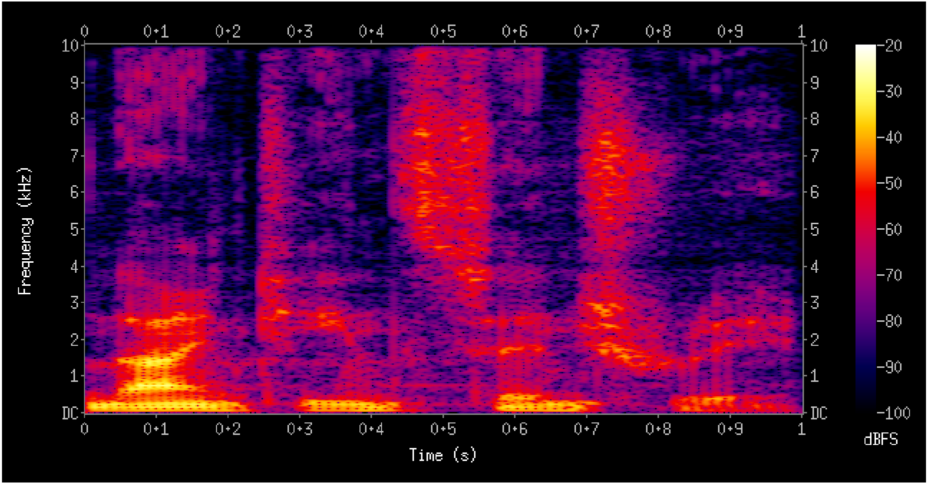
$$X_t[k] = \sum_n x[n]w[n-tD]e^{-j\omega_k(n-tD)}$$

- **ISTFT using overlap-add:**

$$x[n] = \frac{\sum_t \frac{1}{N}\sum_{k=0}^{N-1} X_t[k]e^{j\omega_k(n-tD)}}{\sum_t w[n-tD]}$$

. . . where, usually, we choose the window and the hop length so that $\sum_t w[n-tD]$ is a constant.

Spectrogram $= 20\log_{10}|\text{Short Time Fourier Transform}|$



Outline

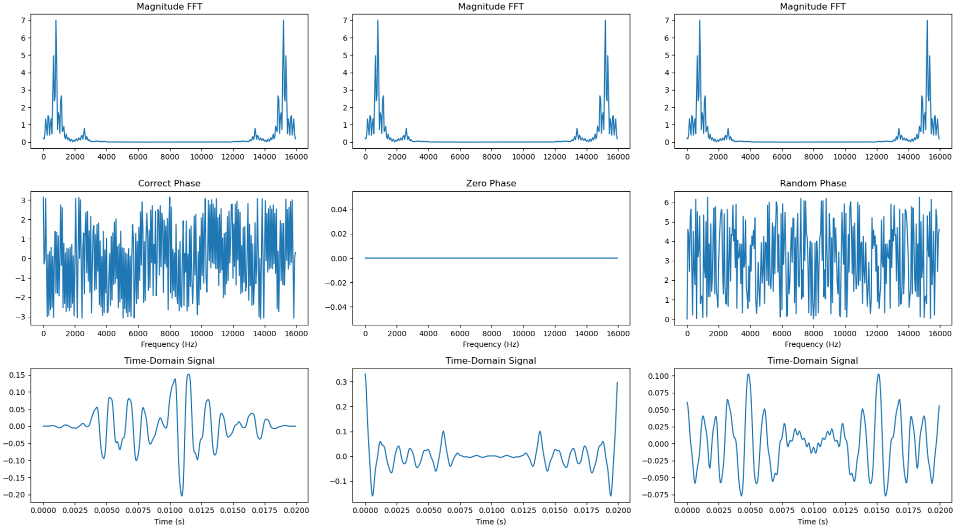
1. Review: STFT and ISTFT
2. Why is inverting a spectrogram difficult?
3. Griffin-Lim Algorithm: A vector-space representation
4. Griffin-Lim Algorithm: Signal example
5. Conclusions

Inverting a spectrogram

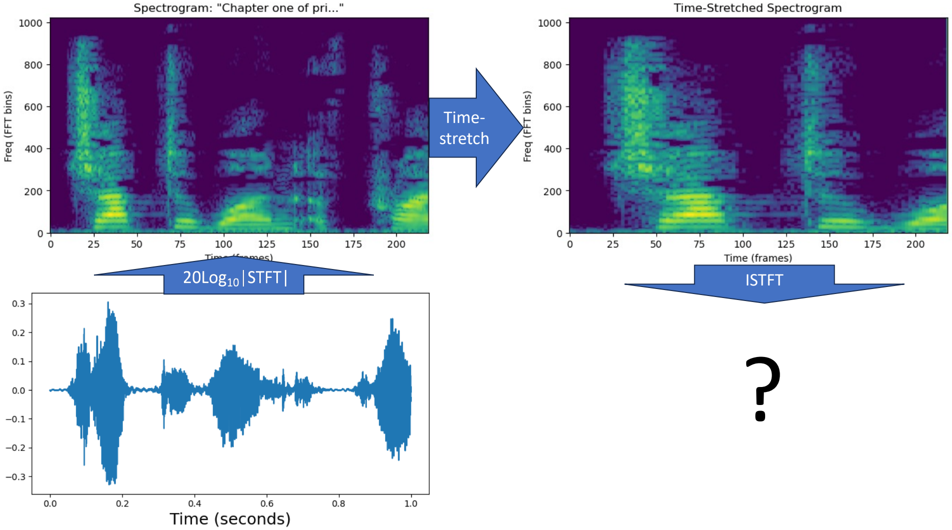
Inverting a spectrogram has two problems:

1. We don't know the phase of $X_t[k]$
2. If $X_t[k]$ was computed from a signal, then it can be inverted. But if it was computed in some other way (e.g., generated by a neural network), then it might not have a valid inverse.

Example: We don't know the phase



Example: There is no valid inverse



Outline

1. Review: STFT and ISTFT
2. Why is inverting a spectrogram difficult?
3. Griffin-Lim Algorithm: A vector-space representation
4. Griffin-Lim Algorithm: Signal example
5. Conclusions

Conjugage symmetry: A linear constraint

- In order for $x[n]$ to be real-valued, $X_t[k]$ must be conjugate-symmetric, i.e., its real part $X_{t,r}[k]$ and its imaginary part $X_{t,i}[k]$ need to satisfy:

$$X_{t,r}[k] = X_{t,r}[N-k]$$
$$X_{t,i}[k] = -X_{t,i}[N-k]$$

- Actually, this is a pretty easy constraint to satisfy. We just need to make sure that the magnitude is $M_t[k] = M_t[-k]$, and the phase is $\phi_t[k] = -\phi_t[-k]$, i.e.,

$$X_t[k] = X_t^*[-k] = M_t[k]e^{j\phi_t[k]}$$

Overlap between frames: A harder linear constraint

- Normally, we invert STFT using overlap-add.
- In order to be a valid STFT, however, you should get the same value of $x[n]$ no matter which window you use to calculate it.
- That means that, if each pair of windows overlap by $L-D$ samples (L is frame length, D is hop length), then those $L-D$ samples need to have exactly the same value, no matter which window you use to calculate them:

$$\frac{\sum_{k=0}^{N-1} X_0[k]e^{j\omega_k n}}{Nw[n]} = x[n] = \frac{\sum_{k=0}^{N-1} X_1[k]e^{j\omega_k (n-D)}}{Nw[n-D]}$$

Overlap between frames: A harder linear constraint

$X_t[k] = X_{t,r}[k] + jX_{t,i}[k]$ is a valid STFT only if it meets these $L-D$ **linear constraints** for the already-known samples of $x[n]$, the ones where $0 \le n - tD < L - D$:

$$\sum_{k=0}^{N-1} X_{t,r}[k] \left(\frac{\cos(\omega_k(n-tD))}{Nw[n-tD]} \right) - \sum_{k=0}^{N-1} X_{t,i}[k] \left(\frac{\sin(\omega_k(n-tD))}{Nw[n-tD]} \right) = x[n]$$

Known magnitude: A magnitude constraint

On the other hand, suppose we know the **magnitude** of the STFT we're trying to construct, $M_t[k]$. This is a **nonlinear constraint**:

$$M_t[k] = \sqrt{X_{t,r}^2[k] + X_{t,i}^2[k]}$$

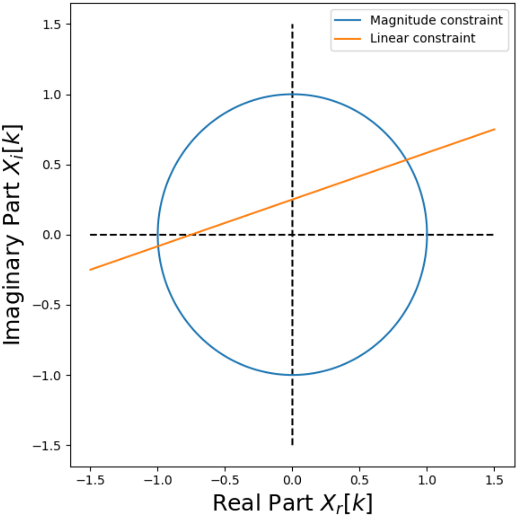
Does a particular magnitude STFT have a valid inverse?

- We want the STFT to have a desired magnitude. This is a **nonlinear constraint**:

$$M_t[k] = \sqrt{X_{t,r}^2[k] + X_{t,i}^2[k]}$$

- In order to be a valid STFT, it must be conjugate symmetric, and overlapping frames need to generate the same samples in the overlap regions. These are **linear constraints**.
- There is no guarantee that there is a valid $X_t[k]$ that satisfies $M[k] = |X_t[k]|$! But to explain the Griffin-Lim algorithm, let's consider a simple example where it is possible to meet both sets of constraints simultaneously, and let's think about how to find the valid STFT.

Combining the two constraints



The Griffin-Lim Algorithm

The Griffin-Lim algorithm tries to find a valid STFT using the following approach:

1. Initialize with random phase, $\phi_t[k] \sim U(0, 2\pi)$:

$$X_t[k] = M_t[k]e^{j\phi_t[k]}$$

2. Repeat the following two steps, over and over, until there is little change from one iteration to the next:
   - Find an $X_t[k]$ that satisfies the linear constraints:

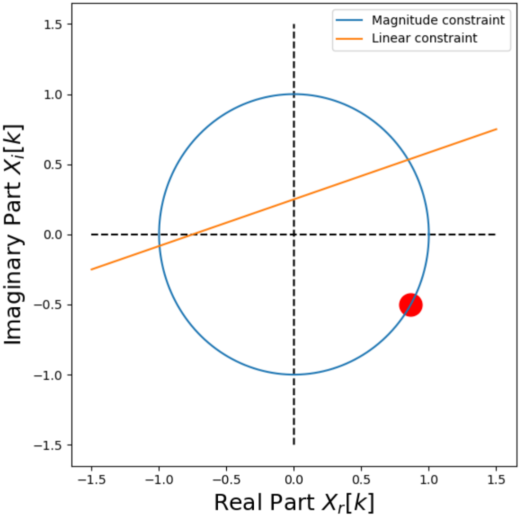
$$X_t[k] \leftarrow \text{STFT}\left\{\text{ISTFT}\left\{X_t[k]\right\}\right\}$$

   - Rescale so that it meets the magnitude constraint:

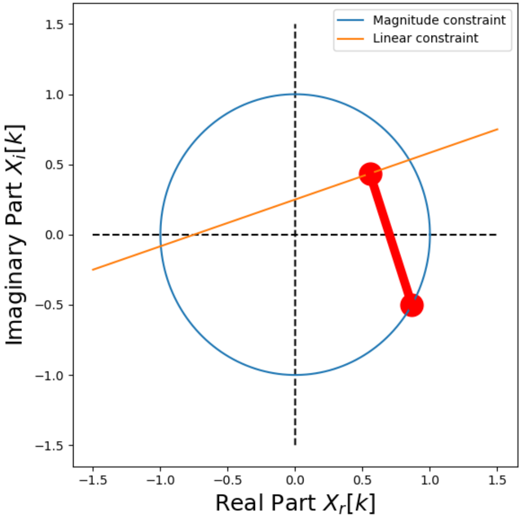
$$X_t[k] \leftarrow M_t[k]e^{j\angle X_t[k]}$$

3. Terminate: $x[n] = \text{ISTFT}\left\{X_t[k]\right\}$

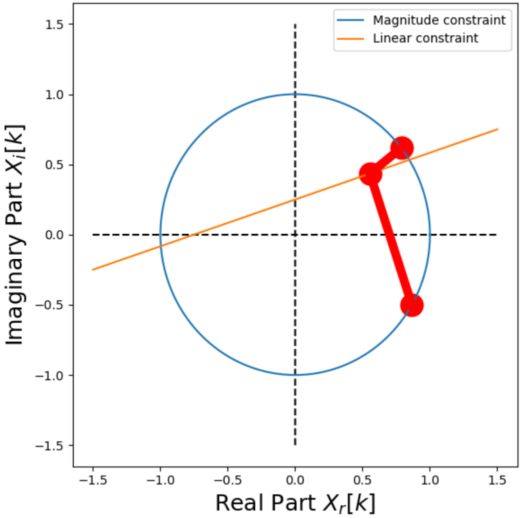
Griffin-Lim initialization: Random phase



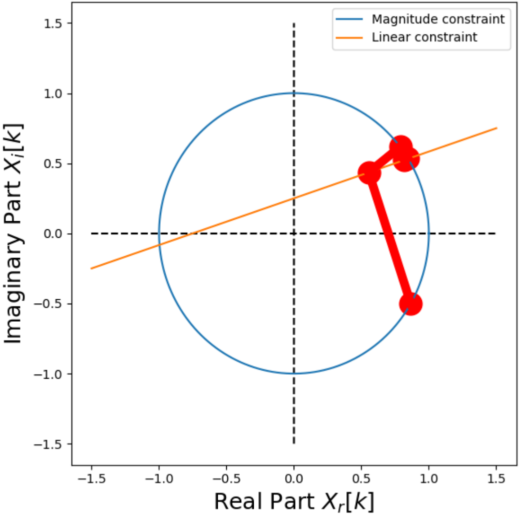
Orthogonal projection $(X_t[k] \leftarrow \mathrm{STFT}\left\{\mathrm{ISTFT}\left\{X_t[k]\right\}\right\})$



Adjusting the magnitude ($X_t[k] \leftarrow M_t[k]e^{j\angle X_t[k]}$)



Iterate until the change from one iteration to the next becomes small



The Griffin-Lim Algorithm

1. Initialize with random phase, $\phi_t[k] \sim U(0, 2\pi)$:

$$X_t[k] = M_t[k] e^{j\phi_t[k]}$$

2. Repeat the following two steps, over and over, until there is little change from one iteration to the next:
   - Find an $X_t[k]$ that satisfies the linear constraints:

$$X_t[k] \leftarrow \text{STFT}\left\{\text{ISTFT}\left\{X_t[k]\right\}\right\}$$

   - Rescale so that it meets the magnitude constraint:

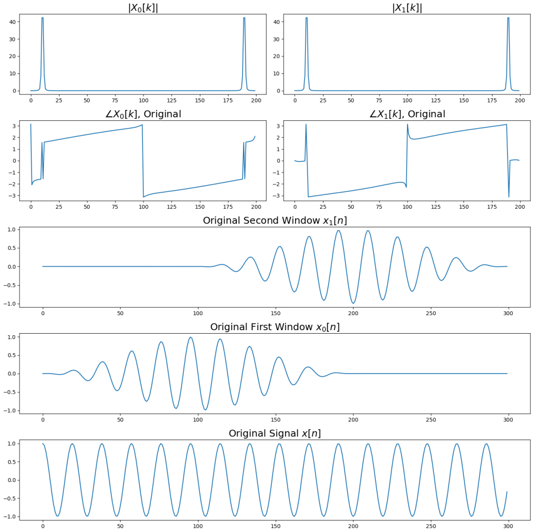
$$X_t[k] \leftarrow M_t[k] e^{j\angle X_t[k]}$$

3. Terminate: $x[n] = \text{ISTFT}\left\{X_t[k]\right\}$

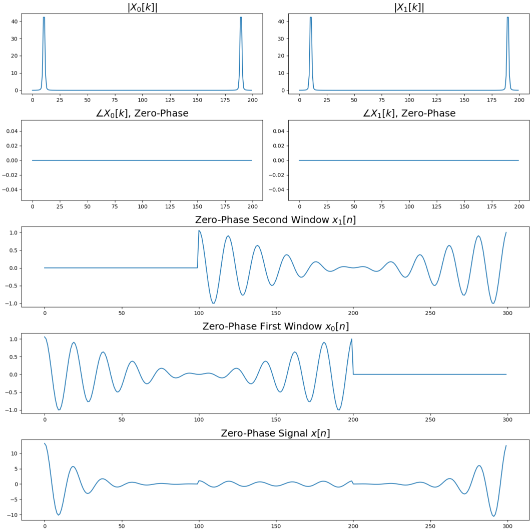
Outline

1. Review: STFT and ISTFT
2. Why is inverting a spectrogram difficult?
3. Griffin-Lim Algorithm: A vector-space representation
4. Griffin-Lim Algorithm: Signal example
5. Conclusions

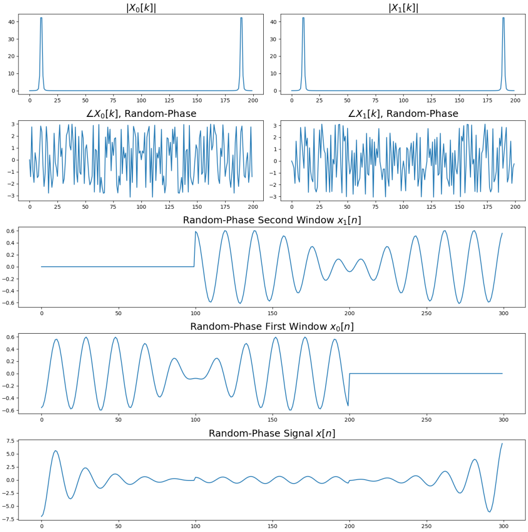
STFT of a cosine: A valid STFT



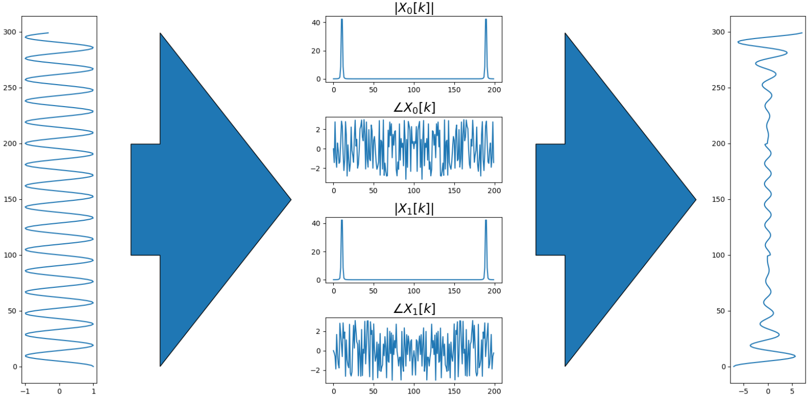
Setting the phase to zero changes the signal!



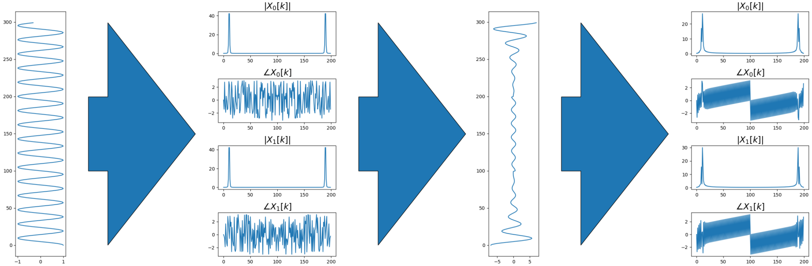
Randomizing the phase also changes the signal!



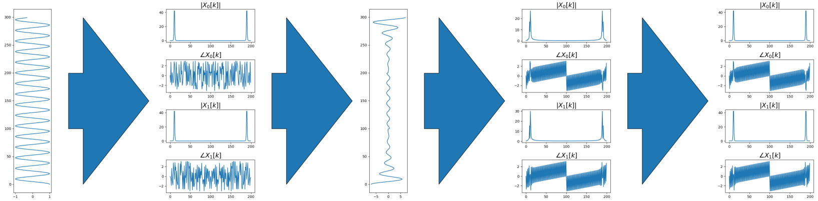
Overlap-add



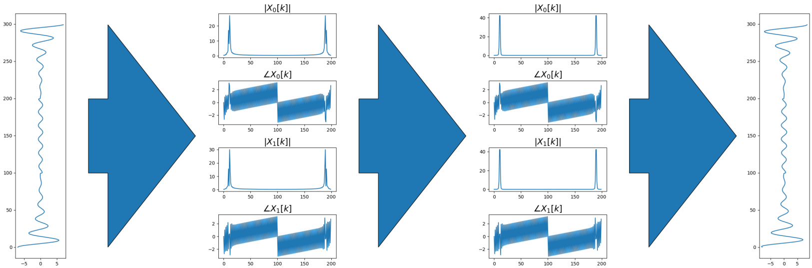
Take the STFT again



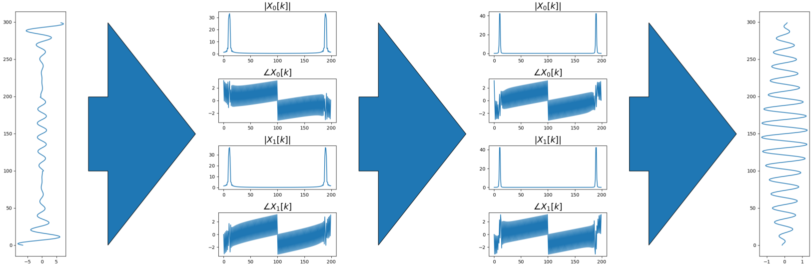
Fix the magnitude



Iterate



… and iterate again and again, until convergence



Outline

1. Review: STFT and ISTFT
2. Why is inverting a spectrogram difficult?
3. Griffin-Lim Algorithm: A vector-space representation
4. Griffin-Lim Algorithm: Signal example
5. Conclusions

Conclusions

1. Initialize with random phase, $\phi_t[k] \sim U(0, 2\pi)$:

$$X_t[k] = M_t[k]e^{j\phi_t[k]}$$

2. Repeat the following two steps, over and over, until there is little change from one iteration to the next:
   - Find an $X_t[k]$ that satisfies the linear constraints:

$$X_t[k] \leftarrow \text{STFT}\left\{\text{ISTFT}\left\{X_t[k]\right\}\right\}$$

   - Rescale so that it meets the magnitude constraint:

$$X_t[k] \leftarrow M_t[k]e^{j\angle X_t[k]}$$

3. Terminate: $x[n] = \text{ISTFT}\left\{X_t[k]\right\}$