Glottal Inverse Filtering

Samuli Siltanen
University of Helsinki, Finland
Franco-German Summer School
Inverse Problems and Partial Differential Equations
October 11, 2013, Bremen, Germany
The goal of Glottal Inverse Filtering is to recover the glottal excitation and the vocal tract filter.

**Direct problem:** If we know the glottal excitation signal and the shape of the vocal tract, what does the microphone record?

**Inverse problem:** Given a speech signal recorded by a microphone, find the glottal excitation and the vocal tract.
Glottal flow as function of time

- Closed
- Opening
- Open
- Closing

Air flow over time
Air flow

Air pressure

Signal recorded by microphone

Time (milliseconds)
The effect of the vocal tract corresponds to settings of a frequency equalizer. Here /a/:
The effect of the vocal tract corresponds to settings of a frequency equalizer. Here /e/:
An improved GIF algorithm has important applications

1. **Computational speech synthesis:**
   Clearer information announcements, more efficient automatic telephone-based services, and devices that help handicapped people express emotions.

2. **Noise-robust automatic speech recognition:**
   Efficient and reliable man-machine interfaces.
We use the Klatt model for the glottal excitation.

The Klatt model has only one parameter called $K$, with values between zero and one.
We use the Klatt model for the glottal excitation.

The Klatt model has only one parameter called K, with values between zero and one.
We use the Klatt model for the glottal excitation

The Klatt model has only one parameter called K, with values between zero and one.
We model the effect of the vocal tract by a linear filter, much like a frequency equalizer.
Technically, the frequency response is described by an all-pole filter.
Looking at two first formants leads to four parameters for the vocal tract.

Angle and length of the pole corresponding to the first formant

Angle and length of the pole corresponding to the first formant
In this model, glottal inverse filtering means finding these five numbers:
But where does Monte Carlo come in the picture?
Inverse filtering is done via random sampling
The Markov chain Monte Carlo method produces a long sequence of excitations and vocal tracts.

\[
\begin{bmatrix}
K_1 \\
\theta_1 \\
r_1 \\
\theta_1 \\
r_1
\end{bmatrix},
\begin{bmatrix}
K_2 \\
\theta_2 \\
r_2 \\
\theta_2 \\
r_2
\end{bmatrix}, \ldots ,
\begin{bmatrix}
K_N \\
\theta_N \\
r_N \\
\theta_N \\
r_N
\end{bmatrix},
\begin{bmatrix}
K_{N+1} \\
\theta_{N+1} \\
r_{N+1} \\
\theta_{N+1} \\
r_{N+1}
\end{bmatrix},
\begin{bmatrix}
K_{N+2} \\
\theta_{N+2} \\
r_{N+2} \\
\theta_{N+2} \\
r_{N+2}
\end{bmatrix}, \ldots ,
\begin{bmatrix}
K_M \\
\theta_M \\
r_M \\
\theta_M \\
r_M
\end{bmatrix}
\]

![The choosing of the members in the above chain is done in a controllably random way. As a result, the chain explores all combinations of glottal excitation and vocal tract filter that (1) Produce closely the measured signal, and (2) Satisfy our a priori information (for example, if we know the vowel, we have a rough idea where the main formants are located).](image)
The Markov chain Monte Carlo method can be parallelized in two simple ways.

We can simulate the same chain on many processors, each starting with a different seed for the random number generator.

We use Microsoft Windows Azure cloud computing service to get many parallel processors.

Also, we use a middleware provider (Techila Ltd.) that makes it easy to parallelize our Matlab codes automatically.
This is an illustration of all the cross-correlations of parameters in the whole MCMC chain.

M = 40000 and N = 10000, so we average over 30000 members in the chain.
As a result, we can improve on the classical IAIF glottal inverse filtering method.

Alku, Auvinen, Raitio, S & Story (to appear in *Computer Speech and Language*)