

Multi-Frames und Audio Coding

oder

**Lineare inverse Probleme
mit nicht-quadratischen Multi-Frame Constraints**

... **Wer Wie Was ??? Wieso Weshalb Warum ??? wer nicht spielt ...**

- Wie stellen wir Signale dar ??
- Wie könnten wir es noch machen ??

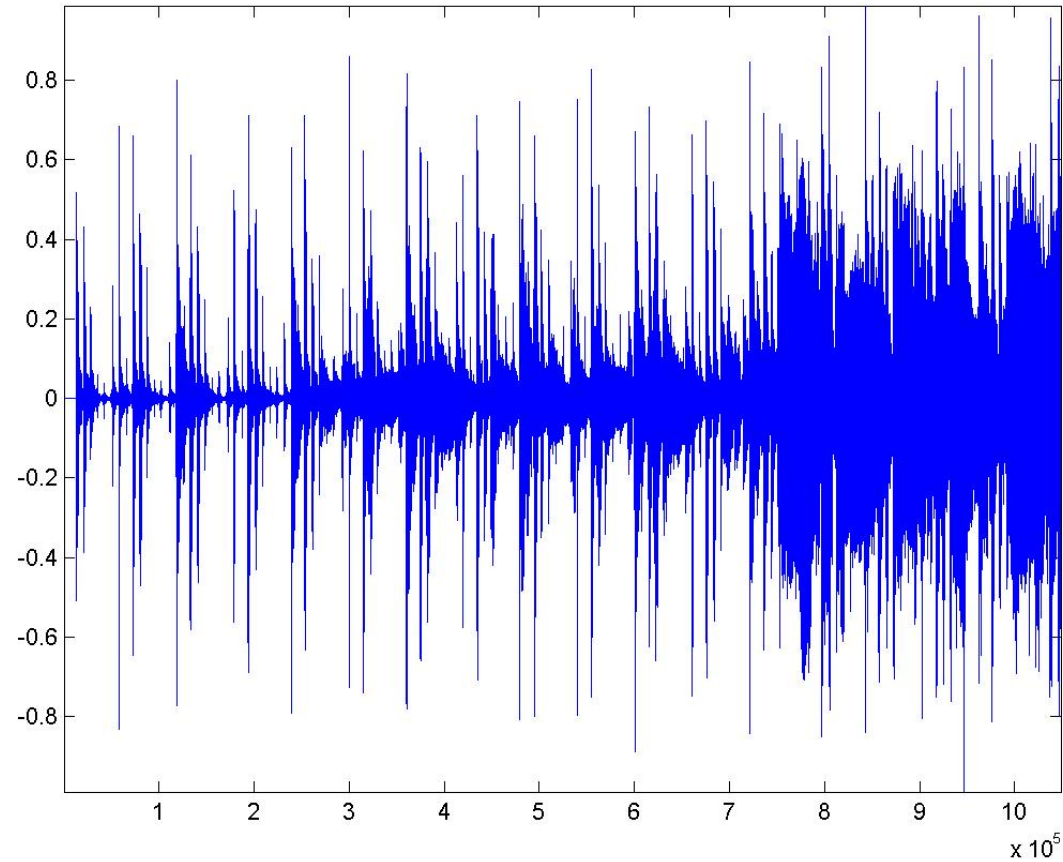
... Warum führt dies auf ein inverses Problem ??

... Welche Rolle spielen Multi-Frames ...

... und warum kann das was mit Audio Coding zu tun haben ??

Ein Beispiel ...

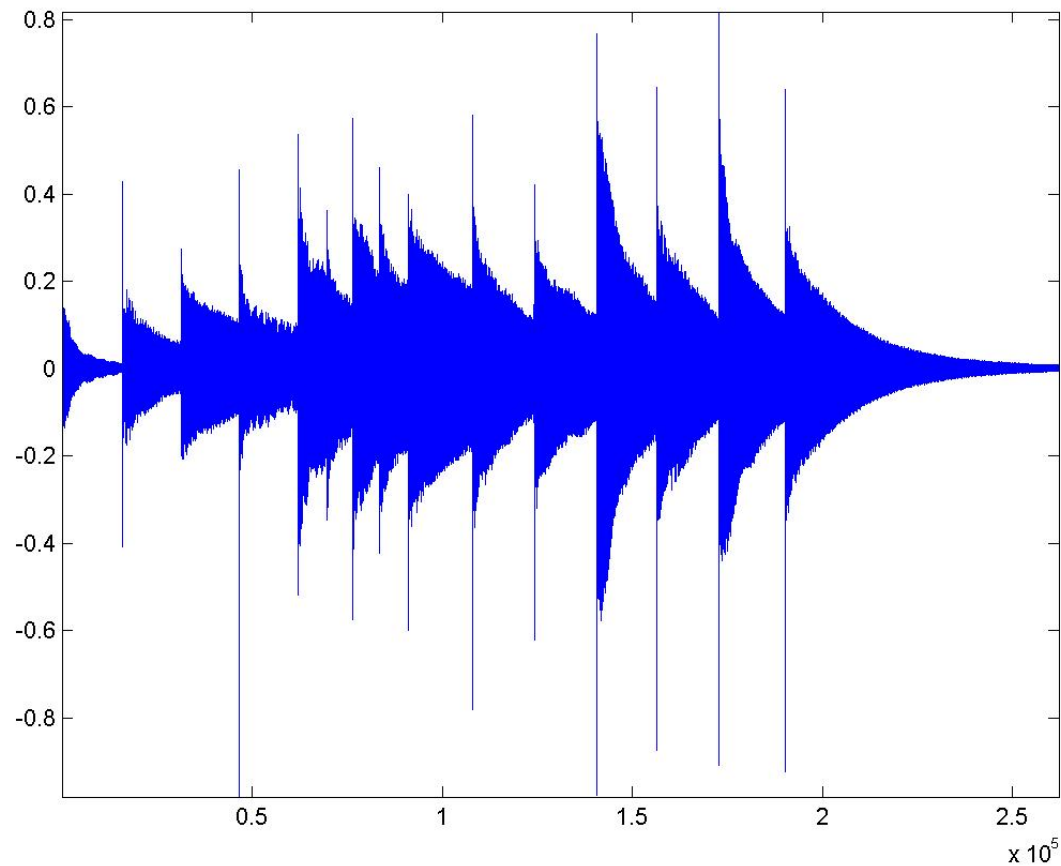
mamavatu.wav: **Original (1048576)**



(play)

Noch ein Beispiel ...

Glock.wav: **Original (262144)**



(play)

Darstellung von Signalen

- $\{\phi_\lambda\}$ Basis:

$$v = \sum_{\lambda} \langle v, \phi_\lambda \rangle \phi_\lambda$$

—→ eindeutige Darstellung, relativ unflexibel

- $\{\phi_\lambda\}$ linear abhängige Menge von Vektoren:

$$v = \sum_{\lambda} g_\lambda \phi_\lambda$$

—→ es gibt viele Folgen $\{g_\lambda\}$ die in Frage kommen, viele Freiheiten

Wie kommen wir auf $\{g_\lambda\}$?

- Frame Operator: $F : v \mapsto \{\langle v, \phi_\lambda \rangle\}$ mit $F^* : \{c_\lambda\} \mapsto \sum_\lambda c_\lambda \phi_\lambda$
- g ist somit Lösung des inversen Problems: $F^* g = v$

→ Standard Frame Theorie:

$$- F^* F v = \sum_\lambda \langle v, \phi_\lambda \rangle \phi_\lambda$$

$$- v = \sum_\lambda \langle v, \phi_\lambda \rangle (F^* F)^{-1} \phi_\lambda = \sum_\lambda \langle v, (F^* F)^{-1} \phi_\lambda \rangle \phi_\lambda$$

$$- g^\dagger = (F^*)^\dagger v = F (F^* F)^{-1} v = \{\langle v, (F^* F)^{-1} \phi_\lambda \rangle\}$$

Wie kommen wir auf $\{g_\lambda\}$?

- **Idee:** approximiere v bzw. $(F^*F)^{-1}\phi_\lambda$ mit Frame Algorithmus

$$v_{m+1} = v_m + \gamma F^*F(v - v_m)$$

- Schreibe diesen Algorithmus mit Hilfe von $(FF^*)_{\eta,\lambda} = \langle \phi_\eta, \phi_\lambda \rangle$ und der Annahme $v_m = \sum_\lambda (g_m)_\lambda \phi_\lambda$ um:

$$g_{m+1} = g_m + \gamma F(v - F^*g_m)$$

Grenzwert löst $F^*g = v$ bzw. minimiert $\|v - F^*g\|^2$

Manchmal nur indirekte Kenntnisse !!!

- Manchmal nicht v , sondern nur $Av = z$
- Manchmal nicht z , sondern nur $f = z + e = Av + e$
- Um nun eine Schätzung für v aus Daten f zu erhalten, minimiere $\|f - Av\|^2$
- **Somit ergibt sich neues inverses Problem:** $AF^*g = f$:

$$\|f - AF^*g\|^2$$

Wir wollen v mit möglichst wenig Koeffizienten g_λ approximieren!

- Variationsfunktional:

$$\Phi(g) = \|f - AF^*g\|^2 + \alpha \|g\|_{\ell_p}^p \quad (p < 2)$$

- **Problem:**

Manchmal besteht v aus sehr verschiedenen Komponenten, so dass ein spezielles Frame $\{\phi_\lambda\}$ **nicht** ausreicht, um v in einer möglichst dünnen Darstellung gut zu approximieren!

Wir wollen v mit möglichst wenig Koeffizienten g_λ approximieren!

- **Multi-Frame:** wähle deshalb eine **Familie** von Frames $\{\phi_\lambda^i\}$ mit $i = 1, \dots, n$
- **Neue Idee:** suche nicht eine sondern n Folgen g^1, \dots, g^n , so dass

$$v = \sum_{i=1}^n F_i^* g^i \quad \text{bzw.} \quad f = \sum_{i=1}^n A F_i^* g^i$$

- Variationsfunktional:

$$\Phi(g^1, \dots, g^n) = \|f - \sum_{i=1}^n A F_i^* g^i\|^2 + \alpha_1 \|g^1\|_{\ell_{p_1}}^{p_1} + \dots + \alpha_n \|g^n\|_{\ell_{p_n}}^{p_n}$$

$$\Phi(g) = \|f - K_A g\|^2 + \alpha \cdot \|g\|, \quad g = (g^1, \dots) \in (\ell_2)^n \quad \alpha = (\alpha_1, \dots)$$

Wir wollen v mit möglichst wenig Koeffizienten g_λ approximieren!

- benutzter Formalismus:

- $F_i : \mathcal{H} \rightarrow \ell_2$

- $K_A : \ell_2 \times \dots \times \ell_2 = (\ell_2)^n \rightarrow \mathcal{H}', (g^1, \dots, g^n) \mapsto \sum_i A F_i^* v^i$

- $\langle g, h \rangle_{(\ell_2)^n} = \langle g^1, h^1 \rangle_{\ell_2} + \dots + \langle g^n, h^n \rangle_{\ell_2}, \|g\|_{(\ell_2)^n}^2 = \|g^1\|_{\ell_2}^2 + \dots + \|g^n\|_{\ell_2}^2$

- $K_A^* : \mathcal{H}' \rightarrow (\ell_2)^n, g \mapsto K_A^* g = (F_1 A^* g, \dots, F_n A^* g)$

- Falls $\|A\| < \tilde{C}$, dann $\|K_A\| < \tilde{C} \sqrt{B_1 + \dots + B_n}$

Wir wollen v mit möglichst wenig Koeffizienten g_λ approximieren!

- Strategie: Löse $\Phi(g)$ iterativ
- Resultat: **Multi-dim. shrinked Landweber Verfahren**

$$((g^1)_m, \dots, (g^n)_m) = g_m = \mathbf{S}_{\frac{\alpha}{2C^2}}^p \left(\frac{1}{C^2} \{K_A^* f + C^2 g_{m-1} - K_A^* K_A g_{m-1}\} \right)$$

- ! Beweis siehe unsere Bildverarbeitungsvorlesung, Konvergenz: Topologie in $(\ell_2)^n$
- ! Optimalität, d.h. Auswahl von α geregelt so lange $p_1 = \dots = p_n$

Zwei *live* Experimente:

Matlab

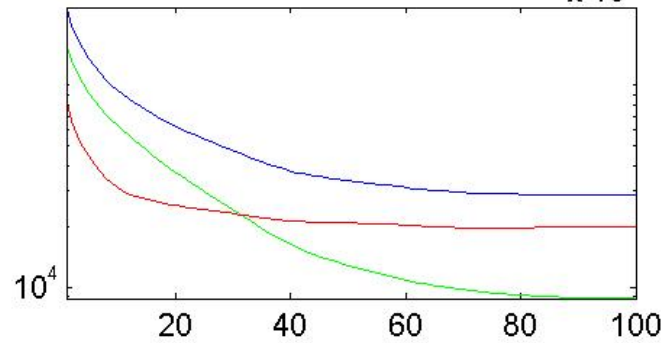
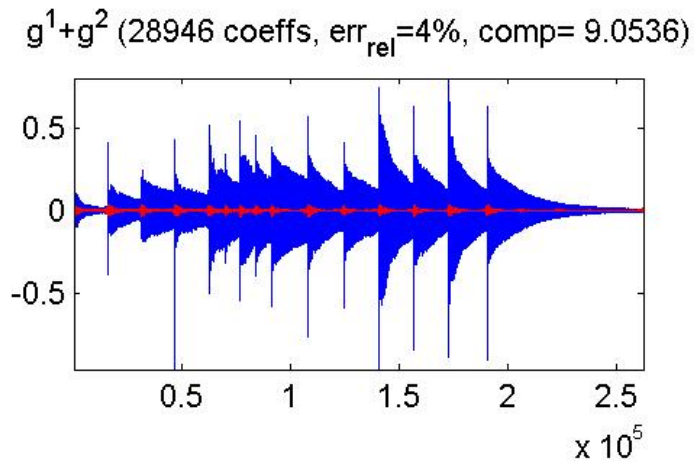
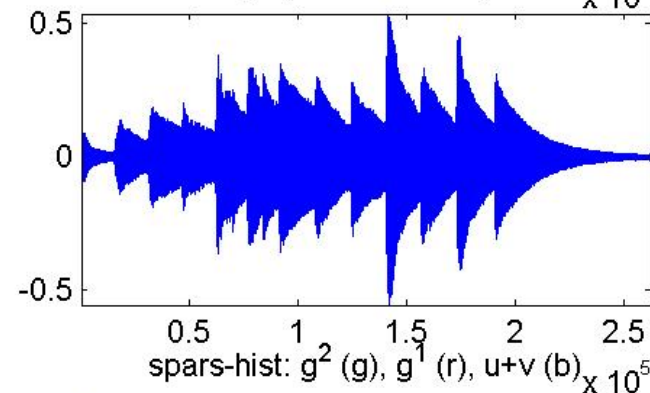
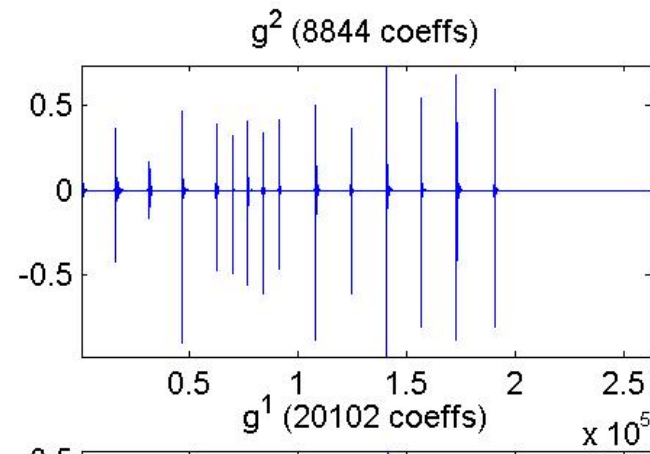
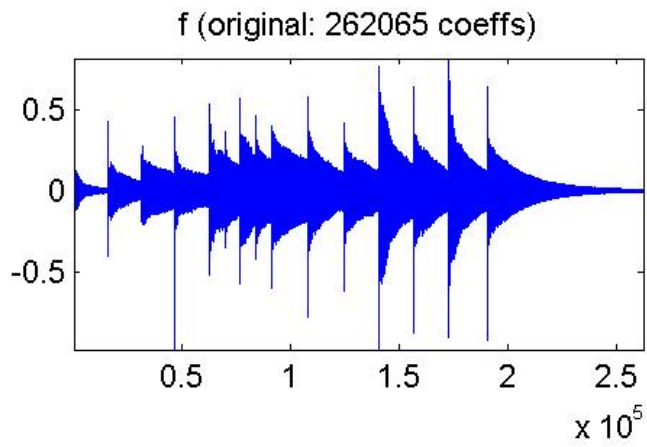
Experiment: Audio Coding ...

- $\{\phi_\lambda\}$ (Gabor) und $\{\psi_\lambda\}$ (wavelet)
- $F_1 : v \mapsto \{\langle v, \phi_\lambda \rangle\}$ und $F_2 : v \mapsto \{\langle v, \psi_\lambda \rangle\}$
- $A = Id$

$$\Rightarrow \Phi(g^1, g^2) = \|f - (F_1^* g^1 + F_2^* g^2)\|^2 + \alpha_1 \|g^1\|_{\ell_1} + \alpha_2 \|g^2\|_{\ell_1}$$

- **Shrunked Landweber Verfahren:**

$$\begin{bmatrix} (g^1)_{m+1} \\ (g^2)_{m+1} \end{bmatrix} = \begin{bmatrix} S_{\alpha_1/2C^2} (C^{-2}\{F_1 f + C^2(g^1)_m - F_1 F_1^*(g^1)_m - F_1 F_2^*(g^2)_m\}) \\ S_{\alpha_2/2C^2} (C^{-2}\{F_2 f + C^2(g^1)_m - F_2 F_1^*(g^1)_m - F_2 F_2^*(g^2)_m\}) \end{bmatrix}$$

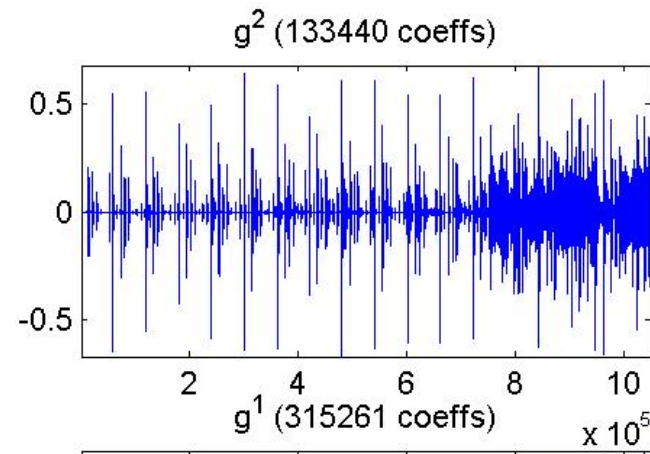
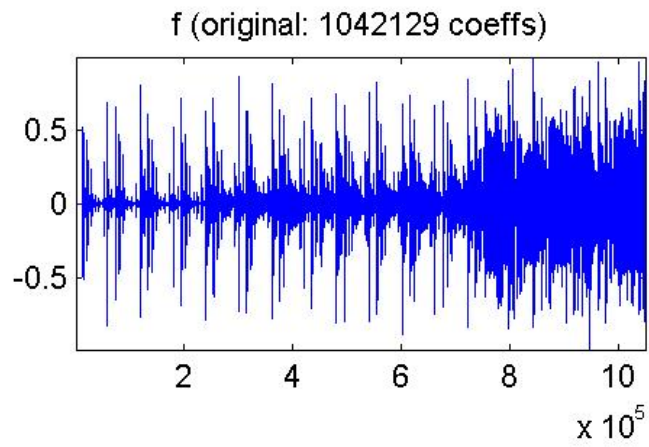


f_{orig}

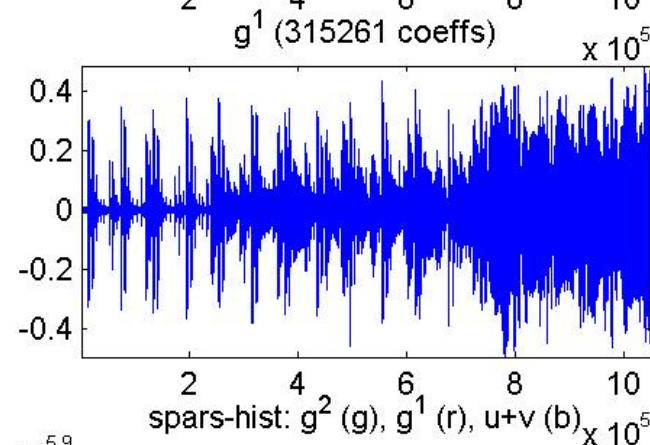
$g^1 + g^2$

g^1

g^2



f_{orig}

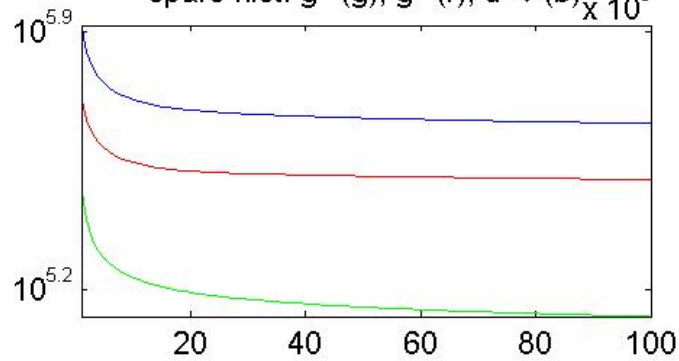
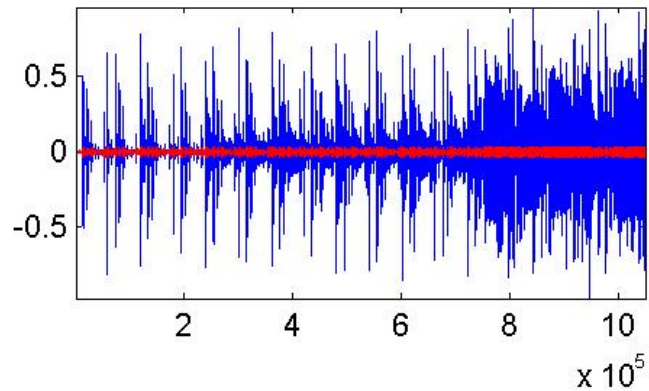


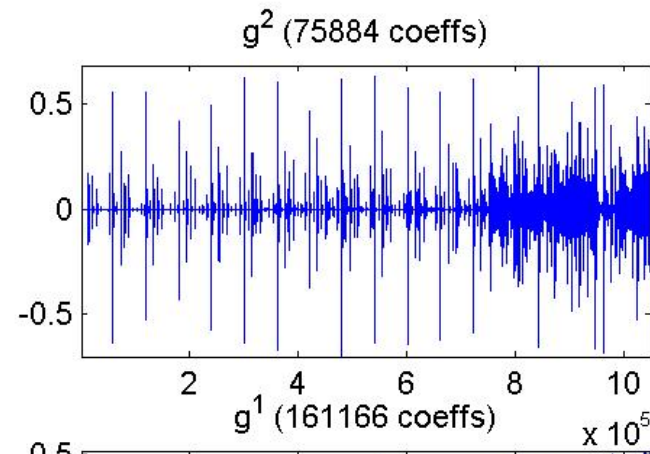
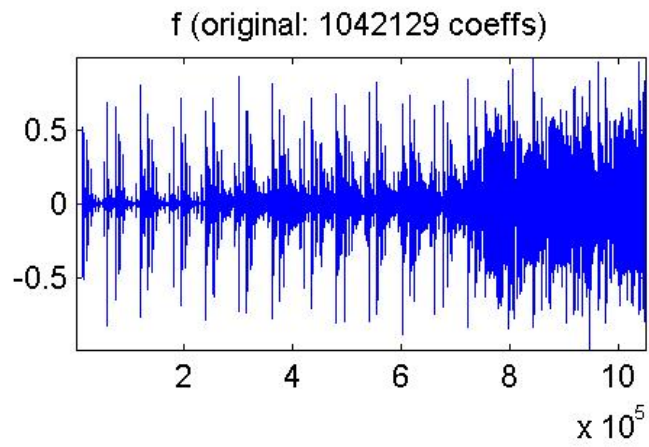
$g^1 + g^2$

g^1

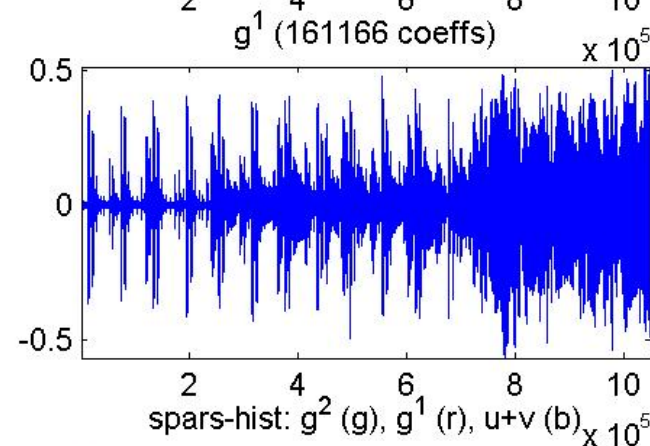
g^2

g^1+g^2 (448701 coeffs, $err_{rel}=5\%$, $comp=2.3225$)





f_{orig}



$g^1 + g^2$

g^1

g^2

g^1+g^2 (237050 coeffs, $err_{rel}=8\%$, $comp=4.3962$)

