UCSD

UNIVERSITY OF CALIFORNIA, SAN DIEGO

Jacobs School *of* Engineering

OSIsoft Data Compression Analysis

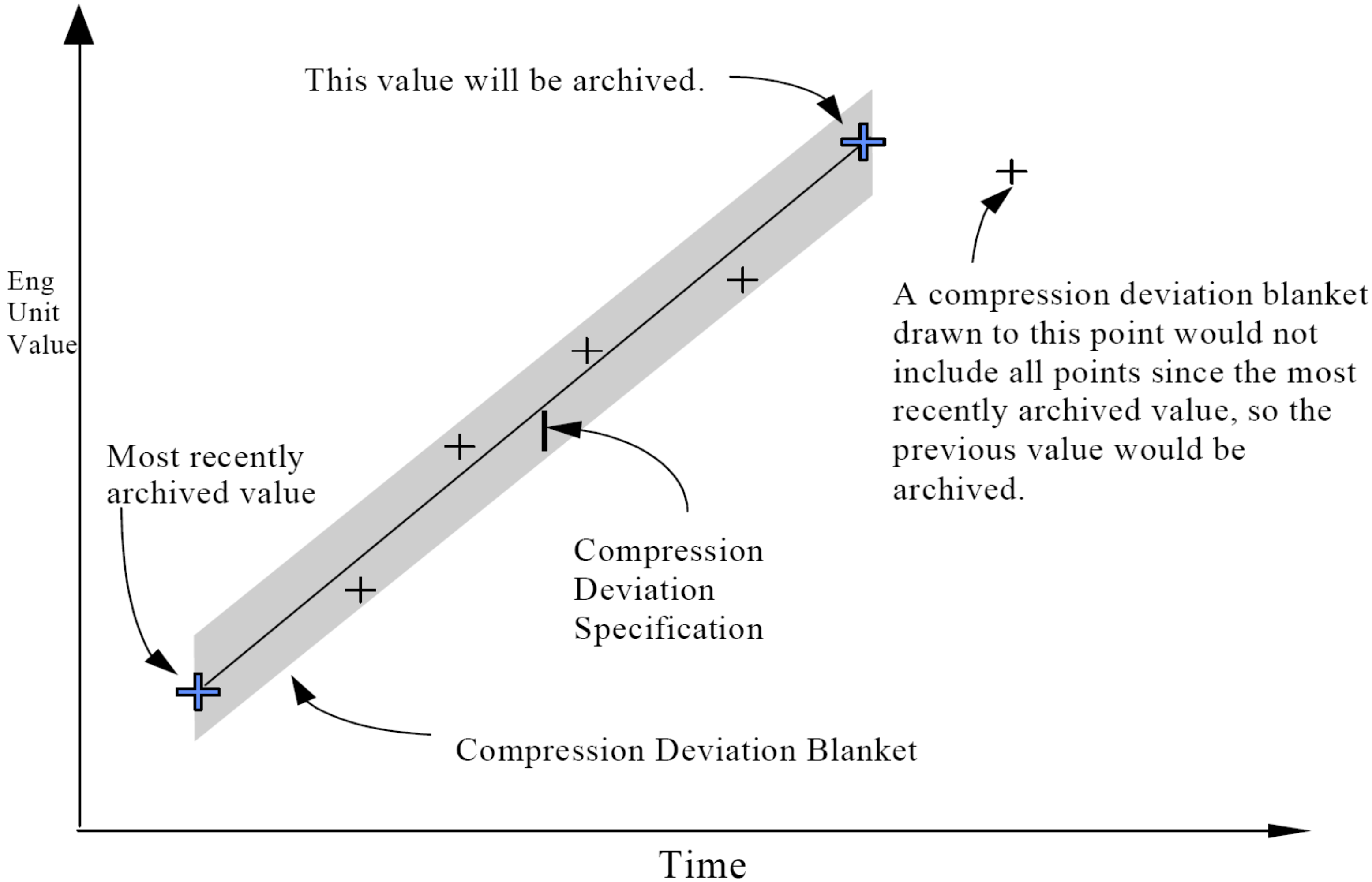
Raymond de Callafon
Charles H. Wells

University of California, San Diego (UCSD)
OSIsoft
email: callafon@ucsd.edu

NASPI Work Group Meeting, Knoxville, TN, March 11-12, 2014

UCSD Jacobs | Mechanical and Aerospace Engineering

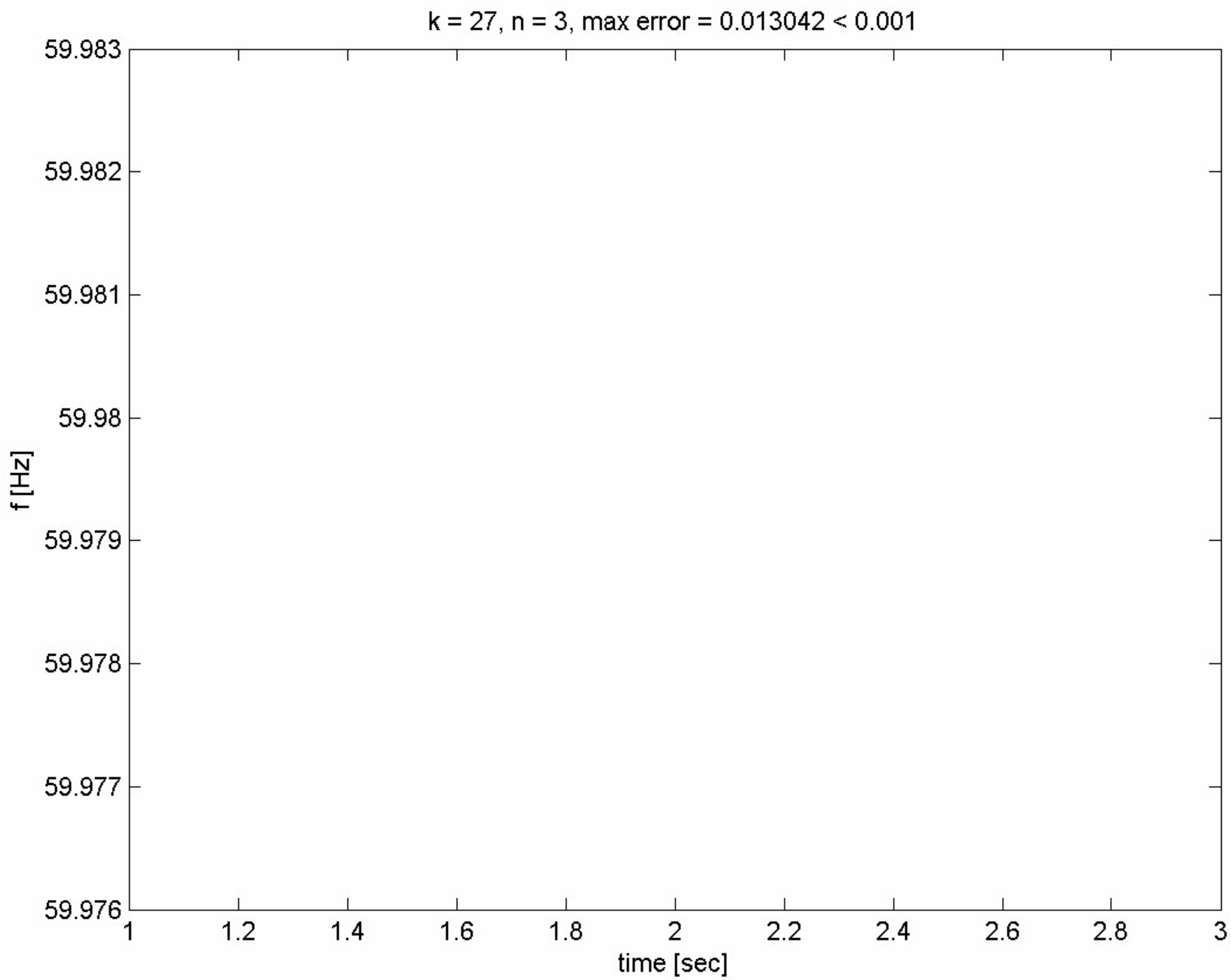
Swinging Door Compression



Swinging Door Compression

Compression in action:

- Done in real-time
- Compression level chosen here at 0.001 Hz



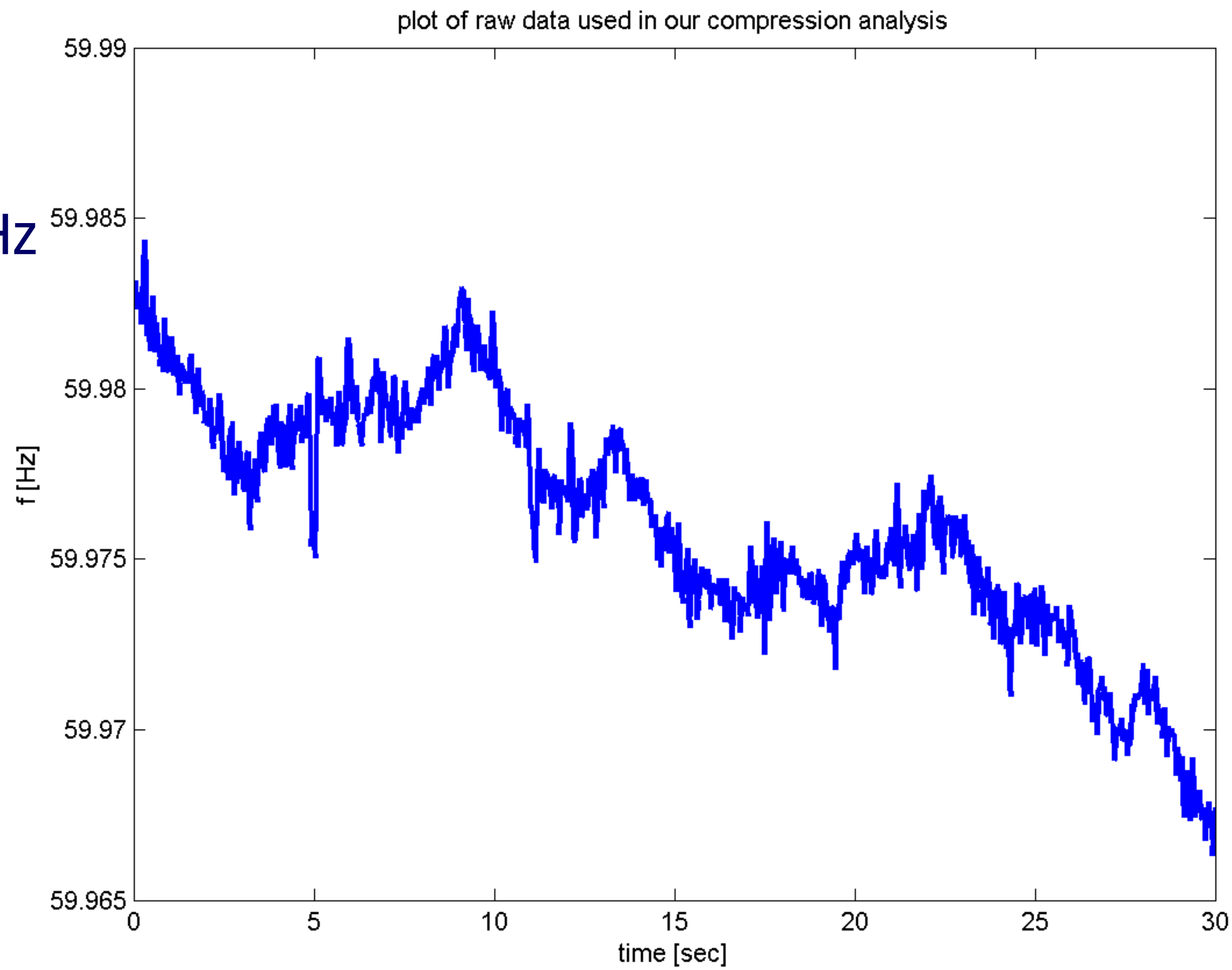
Contents of This Talk

Given "swinging door" data compression technique, consider:

- **Analysis** of data compression
- **Investigate data "loss"** for different compression settings
- **Recommended data compression setting** for synchrophasor data

Test Data as a Usecase

- Data:
  - Frequency
  - Sample at 30Hz
- Data has:
  - Trend
  - Noise
  - Small jumps



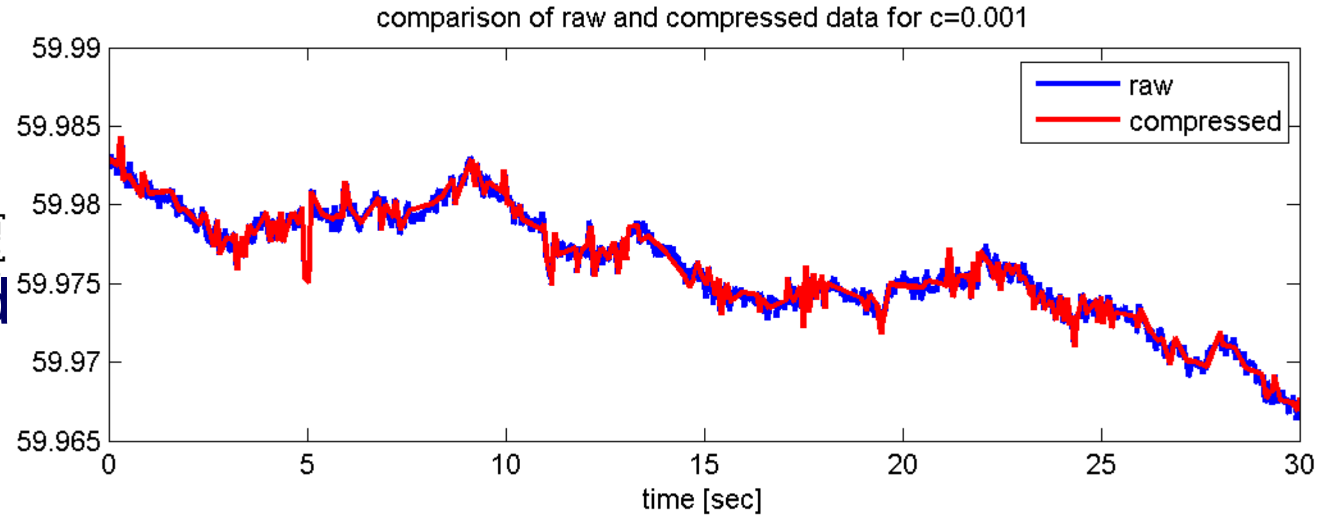
Test Data as a Usecase

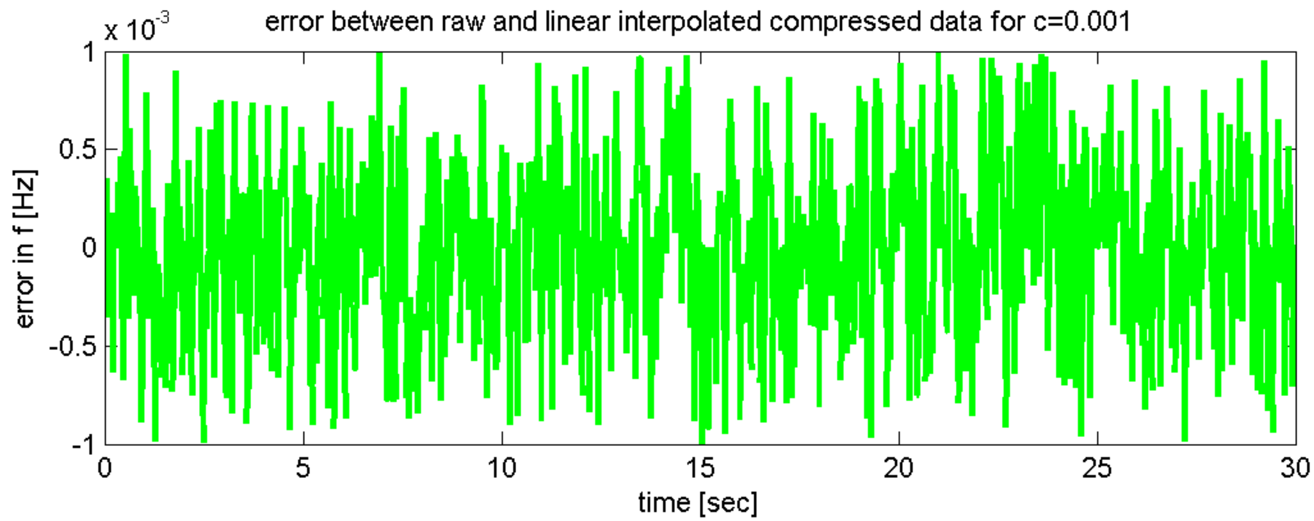
Notation:

- $e(k) = f(k) - f_c(k)$ where $f(k)$ = raw $f_c(k)$ = compressed

Observations:

- Despite trend, $|e(k)|<0.001$
- Due to small compression level, e(k) looks "white noise"

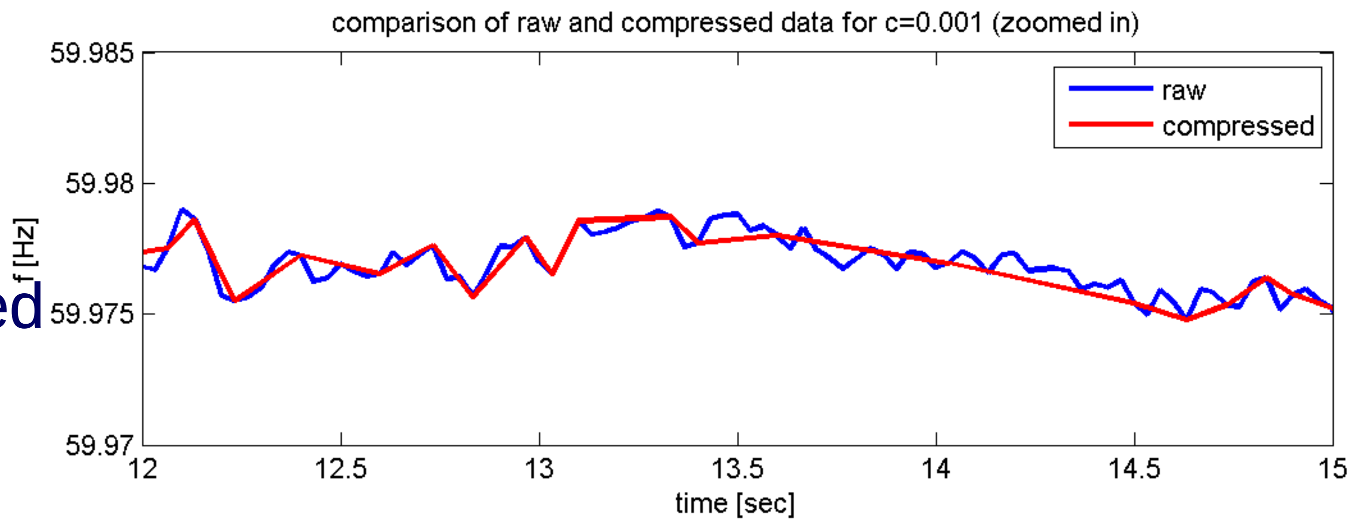




Test Data as a Usecase

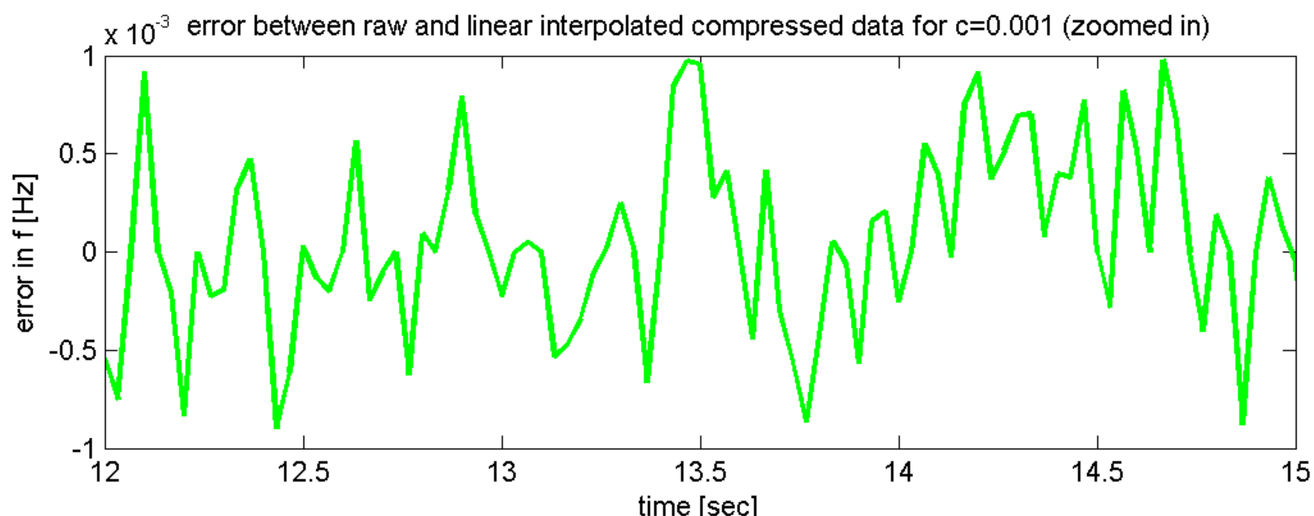
Notation:

- $e(k) = f(k) - f_c(k)$ where $f(k)$ = raw $f_c(k)$ = compressed



Observations:

- Despite trend, $|e(k)| < 0.001$
- Due to small compression level, e(k) looks “white noise”



- Effect of compression clear in time domain: 901 points -> 196 points (21.75%)

Basic Analysis (1/2)

Let $\{t(k), f(k)\}$ = raw data, $\{t_c(k), f_c(k)\}$ = compressed data, then

$$f_l(k) = L\{t_c(k), f_c(k), t(k)\} \qquad e_c(k) = f_l(k) - f(k)$$

where $L\{t_c(k), f_c(k), t(k)\}$ is linear interpolation of $\{t_c(k), f_c(k)\}$ at $t(k)$.

Let Δt be sampling time and let DFT be given by

$$F_l(\omega_n) = \sum_{k=1}^{N} f_l(k) e^{j\omega_n k \Delta t}$$

Since DFT is linear operation:

$$F_l(\omega_n) = F(\omega_n) + E_c(\omega_n)$$

where $E_c(\omega_n)$ is DFT of $e_c(k)$. Note that

$$|F_l(\omega_n)|^2 = |F(\omega_n) + E_c(\omega_n)|^2$$

Basic Analysis (2/2)

With $F_l(\omega_n) = F(\omega_n) + E_c(\omega_n)$ following observations can be made:

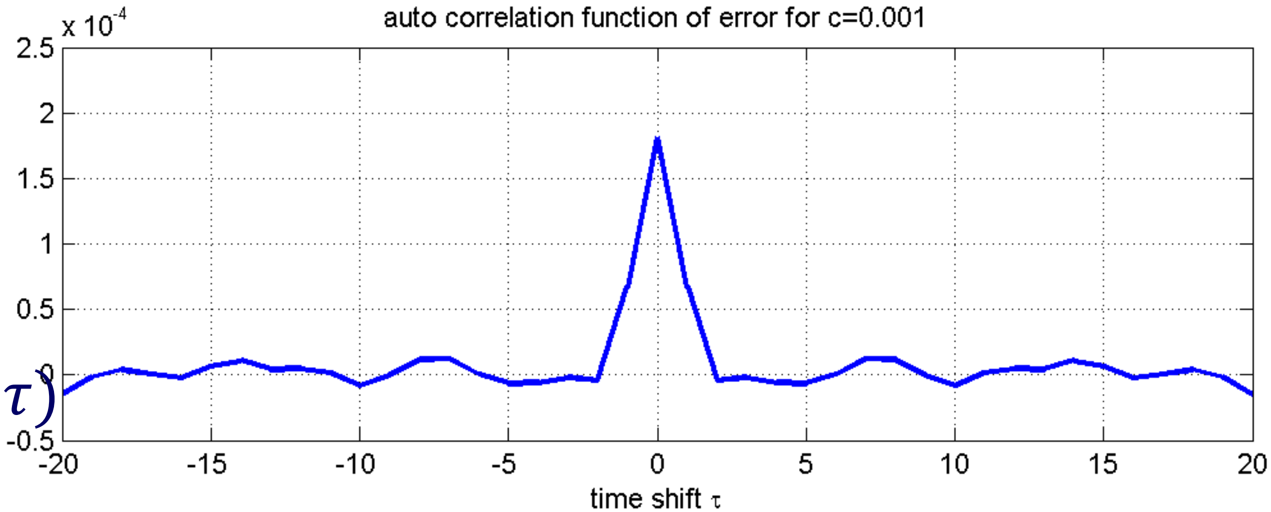
- Fourier transform (and spectra) is influenced linearly by compression error $e_c(k)$.
- Compression error $e_c(k)$ is always bounded $|e_c(k)| < c$ with $E\{e_c(k)\} = 0$ and thus $E\{{e_c}^2(k)\} < c^2/3$ if $e_c(k)$ has uniform distribution (however, not always uniform)
- Aliasing is bounded due to re-interpolation $L\{t_c(k), f_c(k), t(k)\}$ of $\{t_c(k), f_c(k)\}$ at $t(k)$ creating again a sampling frequency of $1/\Delta t$
- If $e_c(k)$ is “pure” white noise, $E_c(\omega_n)$ is a complex number with
  $|E_c(\omega_n)| = c^2/(6\pi)$ $\qquad angle\{E_c(\omega_n)\} \in [-\pi, \pi]$

Back to our Test Data as a Usecase

Check if “white”:

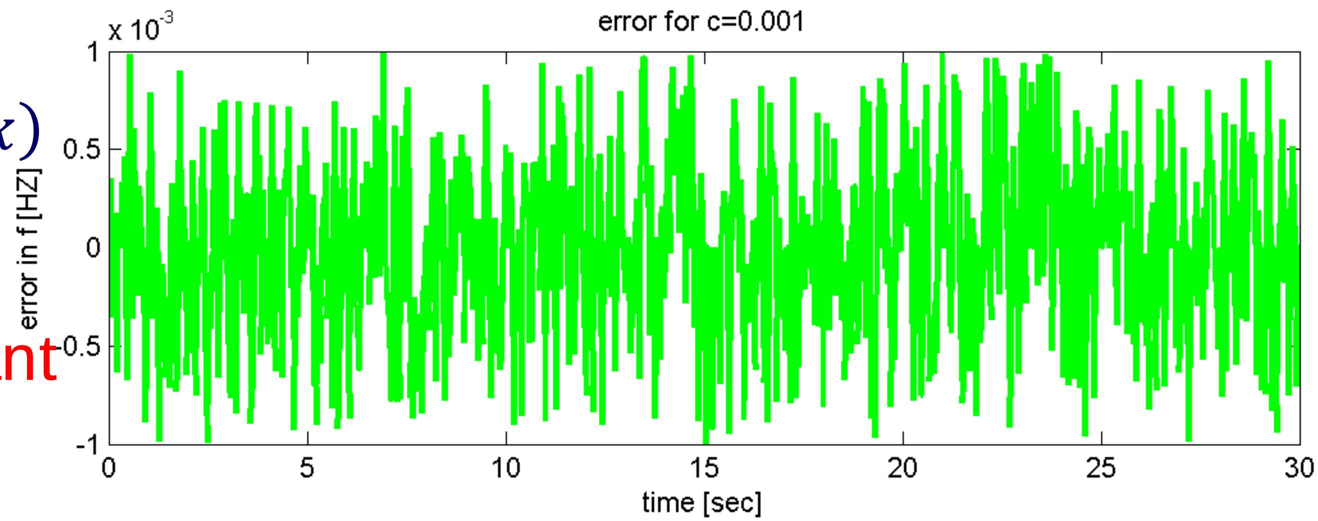
- Compute auto-correlation function:

$$R_e(\tau) = E\{e_c(k)e_c(k-\tau)\}$$



Result:

- If indeed error $e_c(k)$ is white, spectrum (periodogram) $|E_c(\omega_n)|^2 =$ constant (but noisy)



UCSD Jacobs | Mechanical and Aerospace Engineering

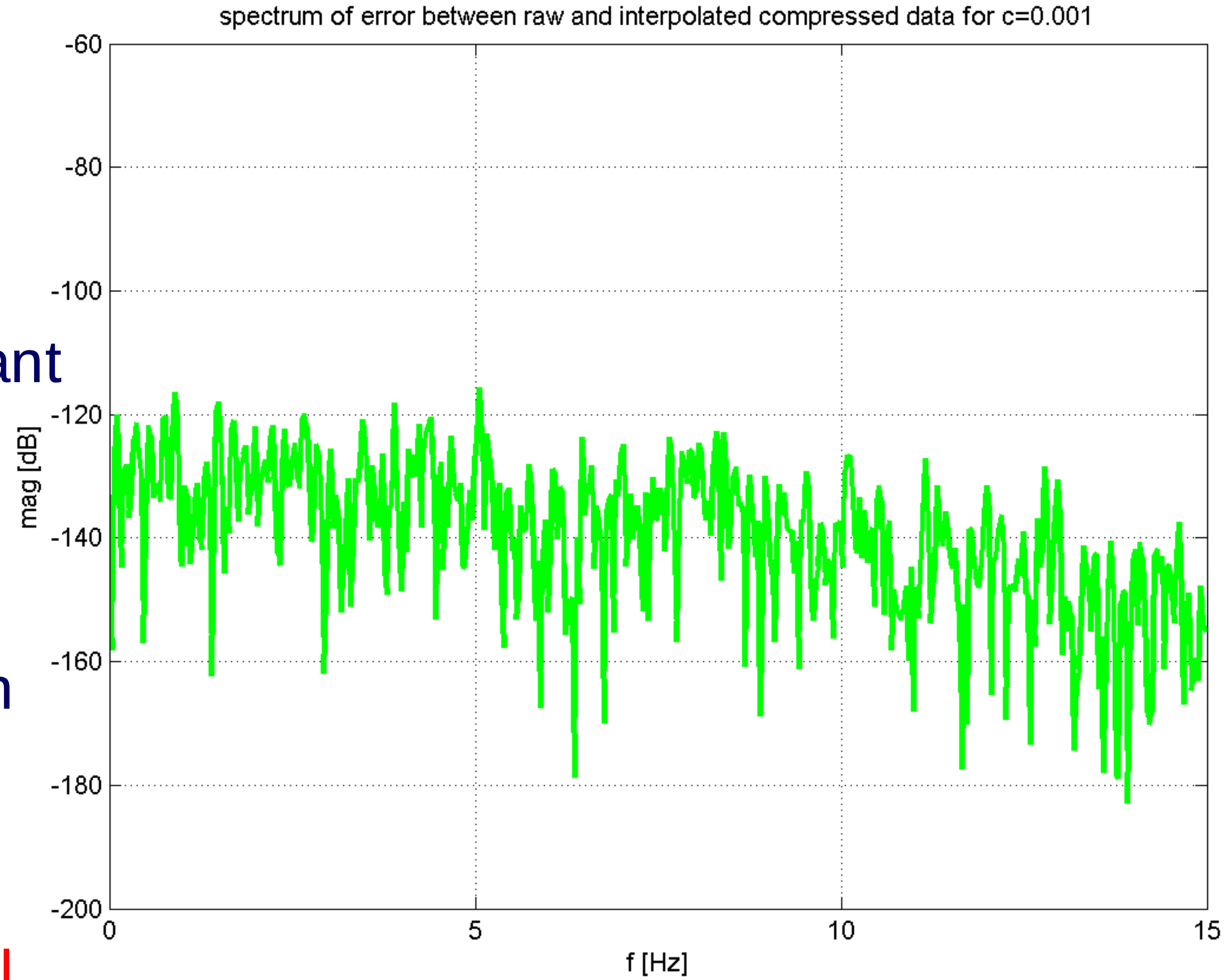
Back to our Test Data as a Usecase

Result:

- Indeed, spectrum (periodogram) $|E_c(\omega_n)|^2 =$ constant (but noisy)

Conclusion:

- Spectrum for given compression level where $e_c(k)$ is "white" noise provides base level

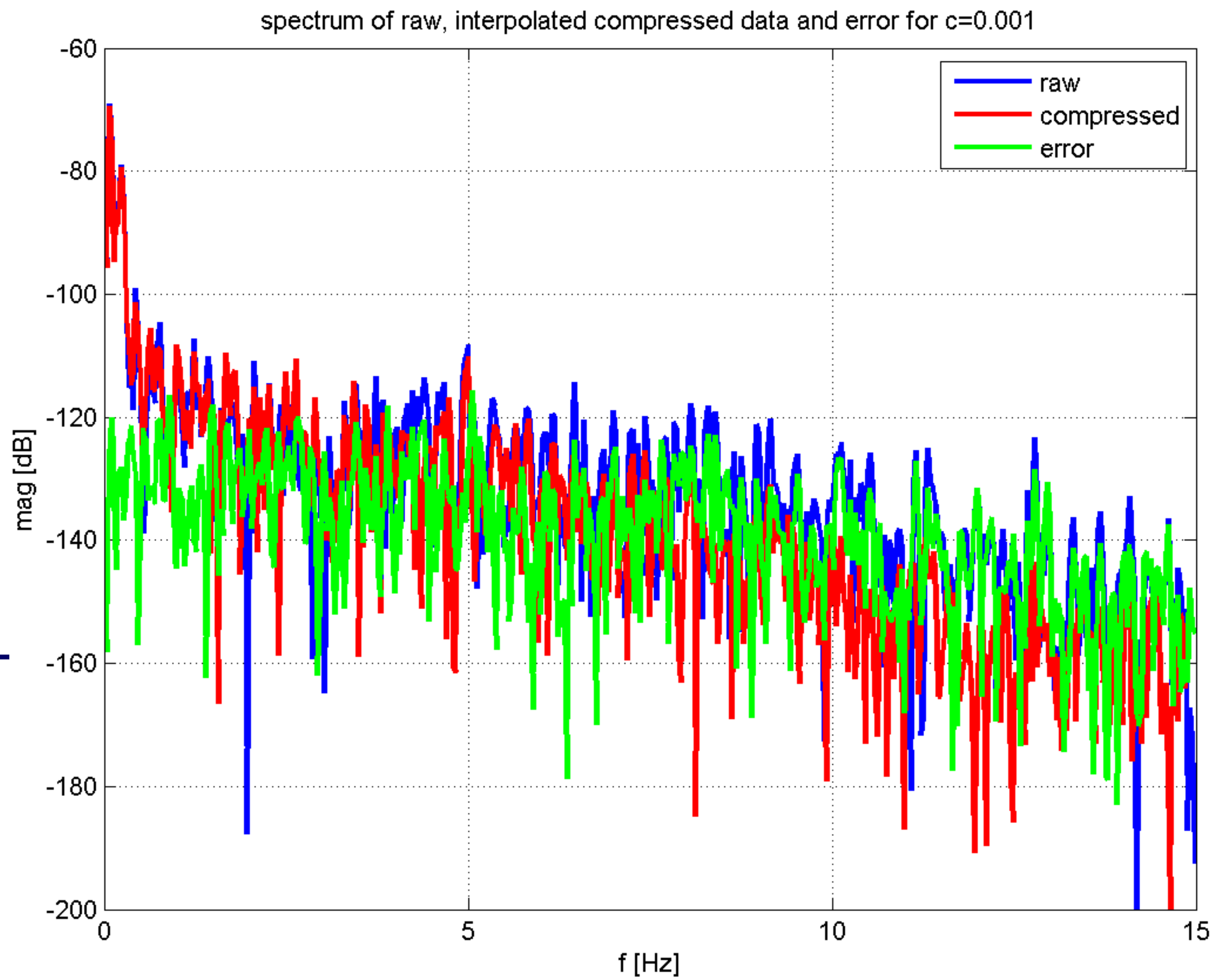


Back to our Test Data as a Usecase

Conclusion:

Compression only influences DFT significantly when

- DFT comes close to base level
- When $e_c(k)$ is NOT a “white” noise!



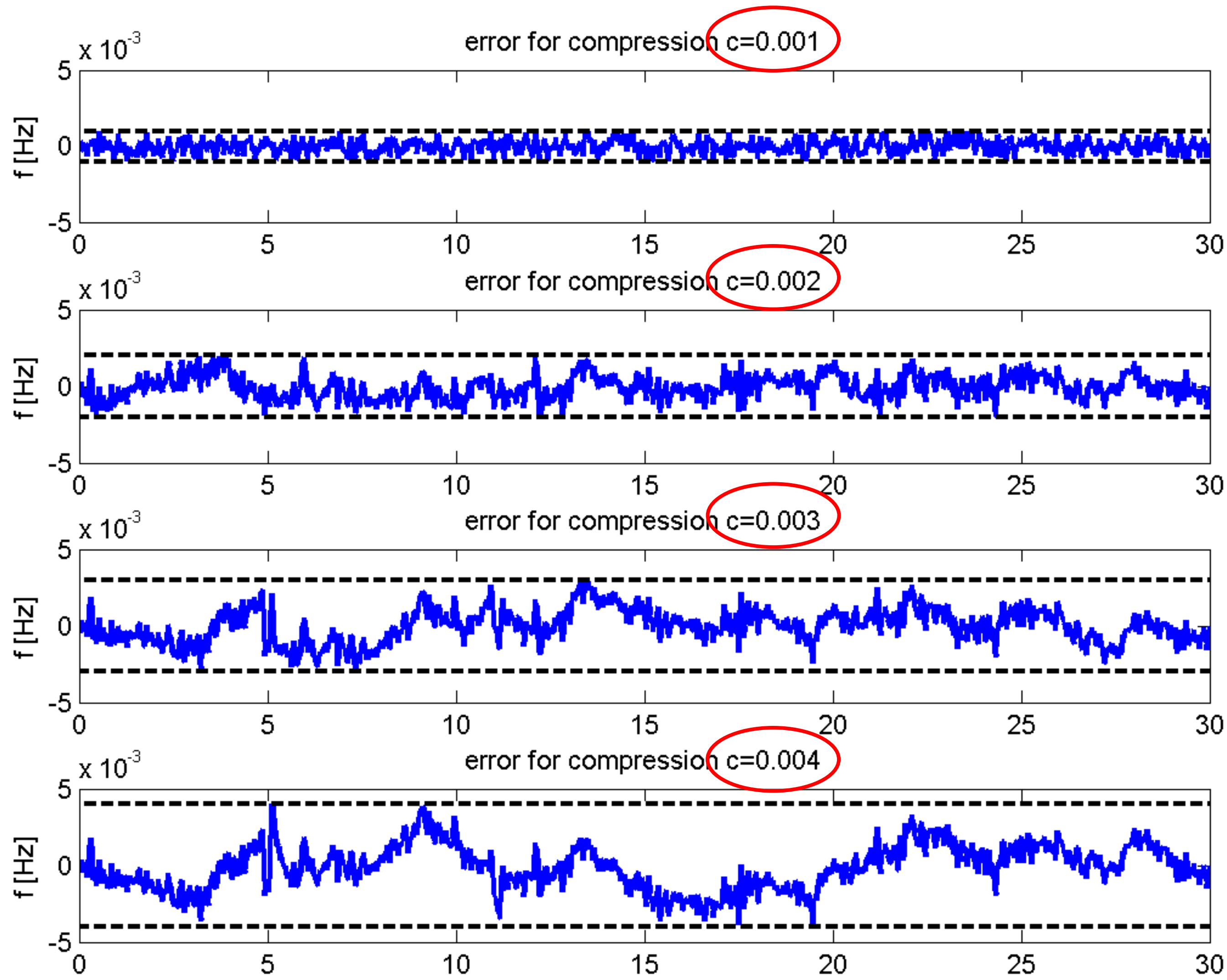
Hence:

Compression level c must be chosen such that $e_c(k)$ is a white noise!

Back to our Test Data as a Usecase

Observe:

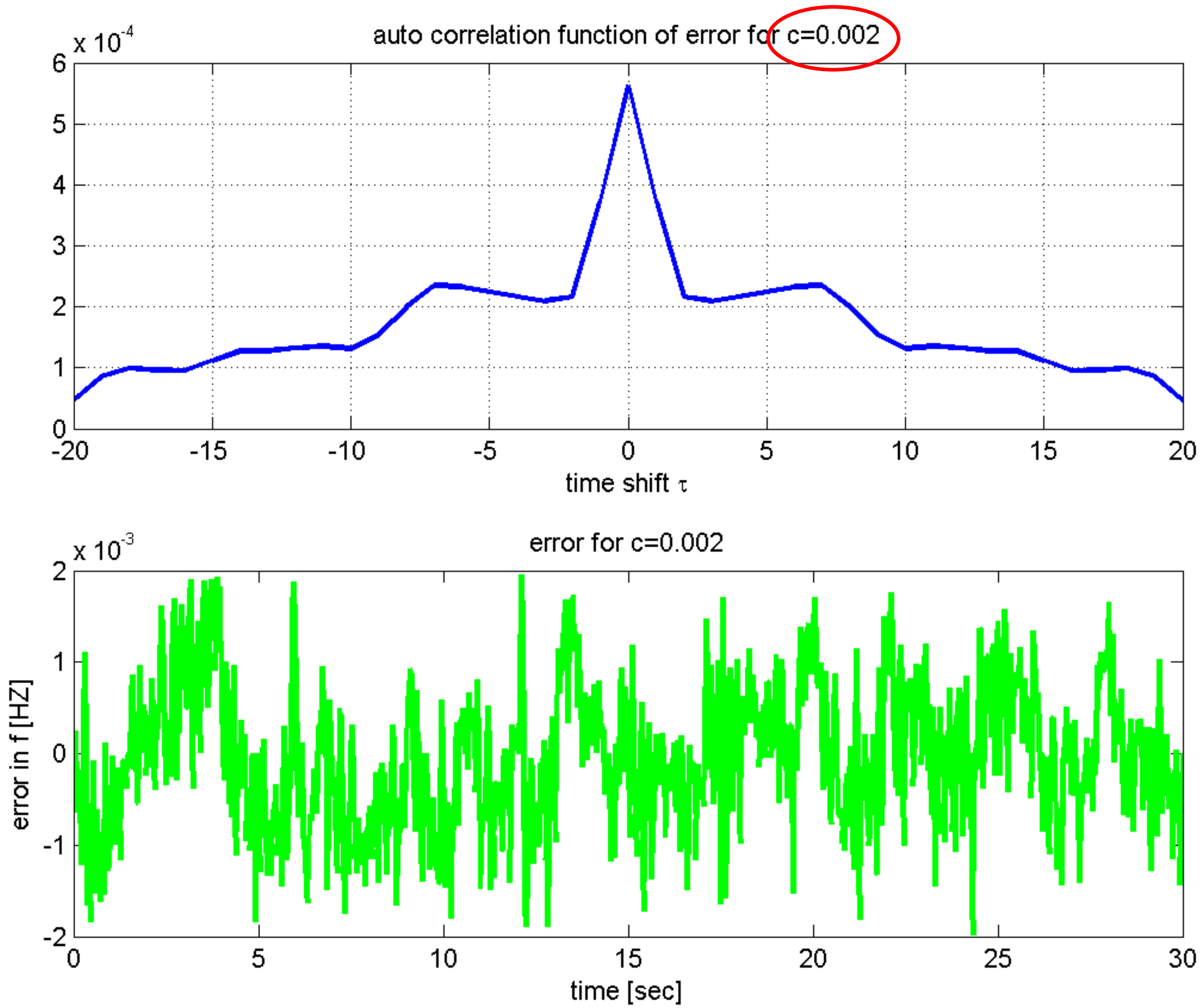
- Increasing compression level c
- At low levels of c=0.001, $e_c(k)$ is a white noise
- For higher levels of c>0.001, $e_c(k)$ is NOT a “white” noise anymore



Back to our Test Data as a Usecase

Observe:

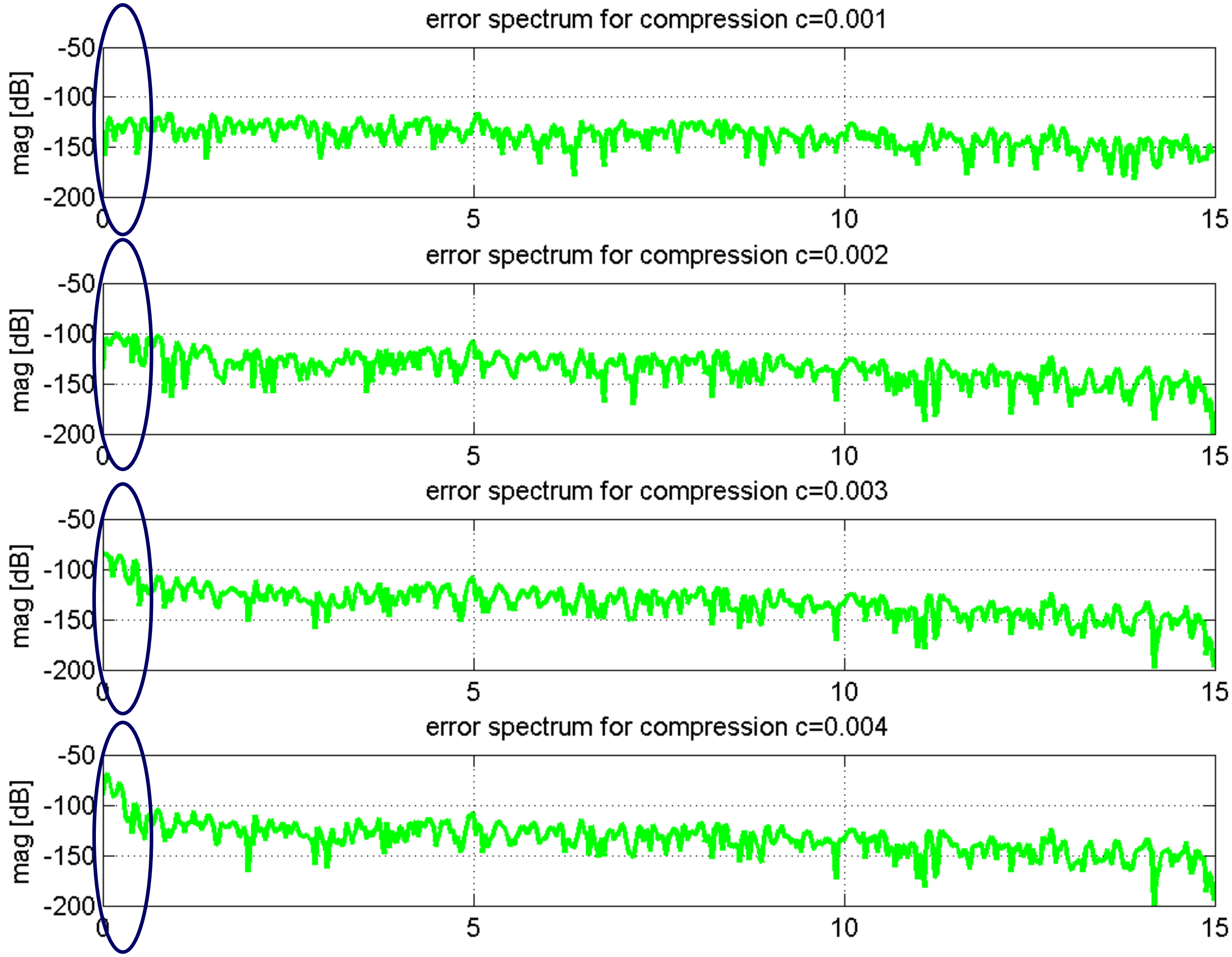
- Increasing compression level c
- $e_c(k)$ is NOT a “white” noise anymore
- Effect can also be seen in correlation function (for c=0.002)



Back to our Test Data as a Usecase

Observe:

- Increasing compression level c
- $e_c(k)$ is NOT a “white” noise anymore
- Effect can also be seen in spectra

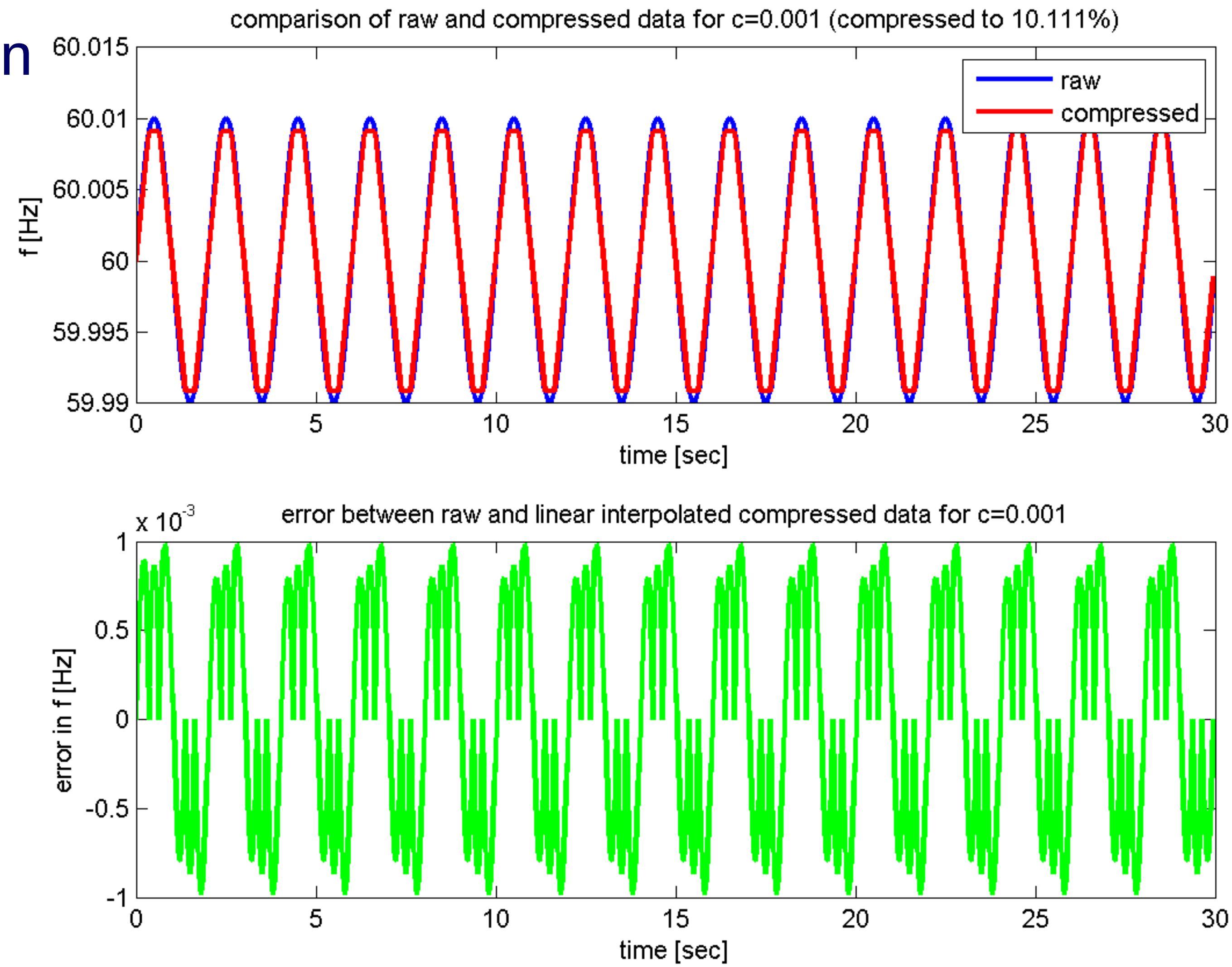


Another usecase: noise free fixed sine

Effect of compression is immediately clear when signal does not have noise:

Data:

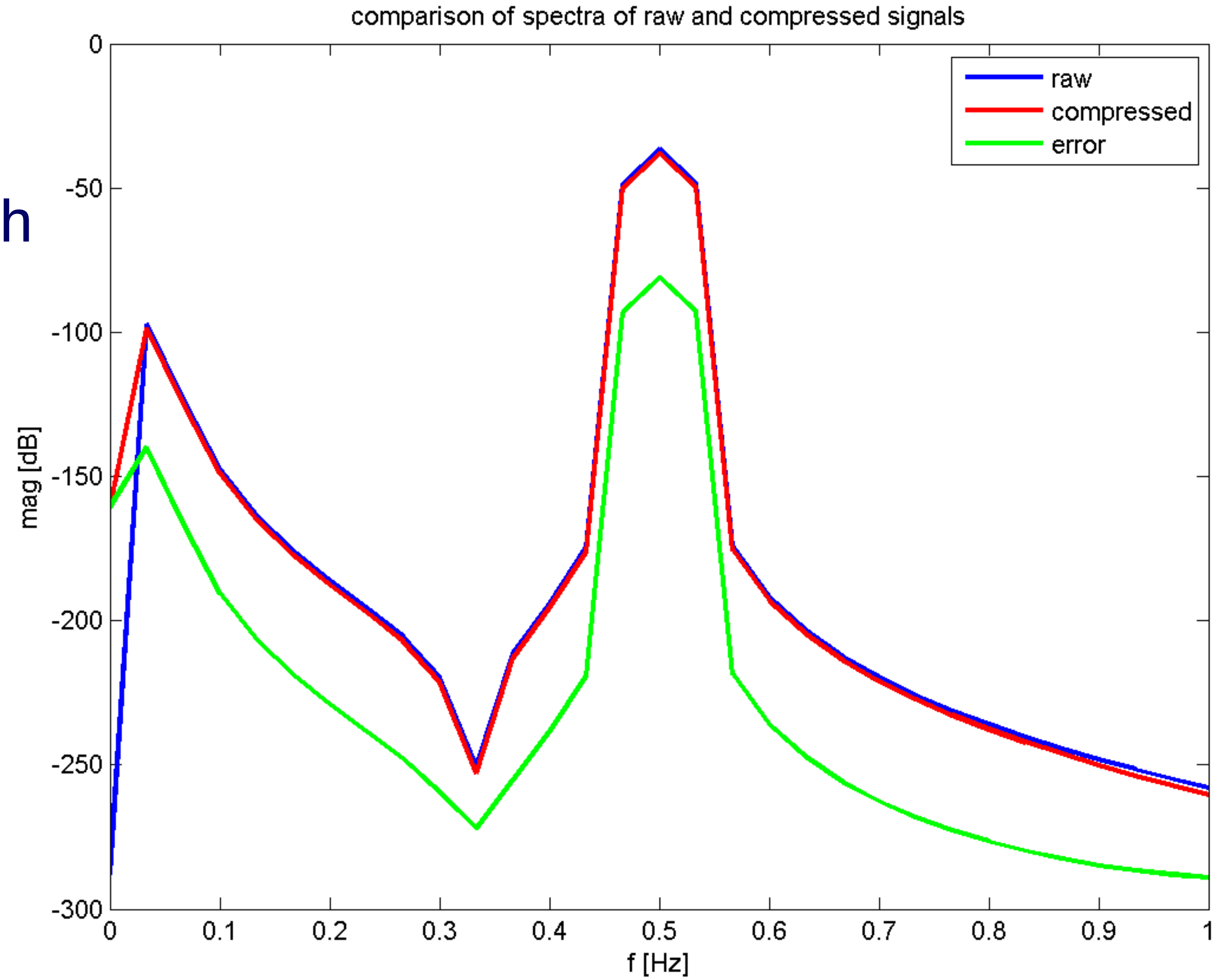
- Single sinusoid with compression level c=0.001
- 900 -> 91 points
- $e_c(k)$ is NOT a “white” noise but periodic!
- Periodic error signal may have several harmonics



Another usecase: noise free fixed sine

Data:

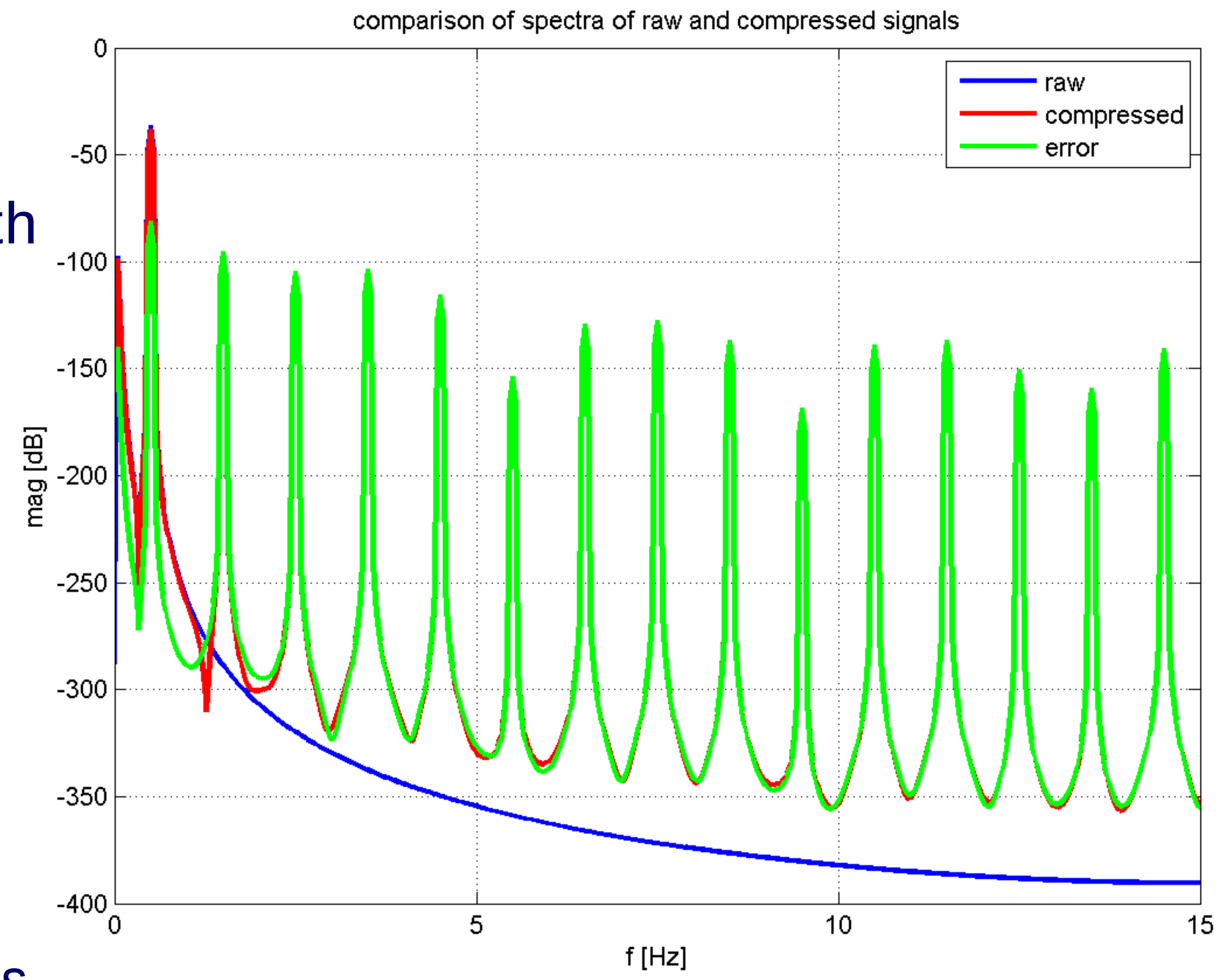
- Single sinusoid with compression level c=0.001
- 900 -> 91 points
- $e_c(k)$ is NOT a “white” noise but periodic!
- Periodic error signal may have several harmonics
- Spectrum hardly influenced at main harmonic!



Another usecase: noise free fixed sine

Data:

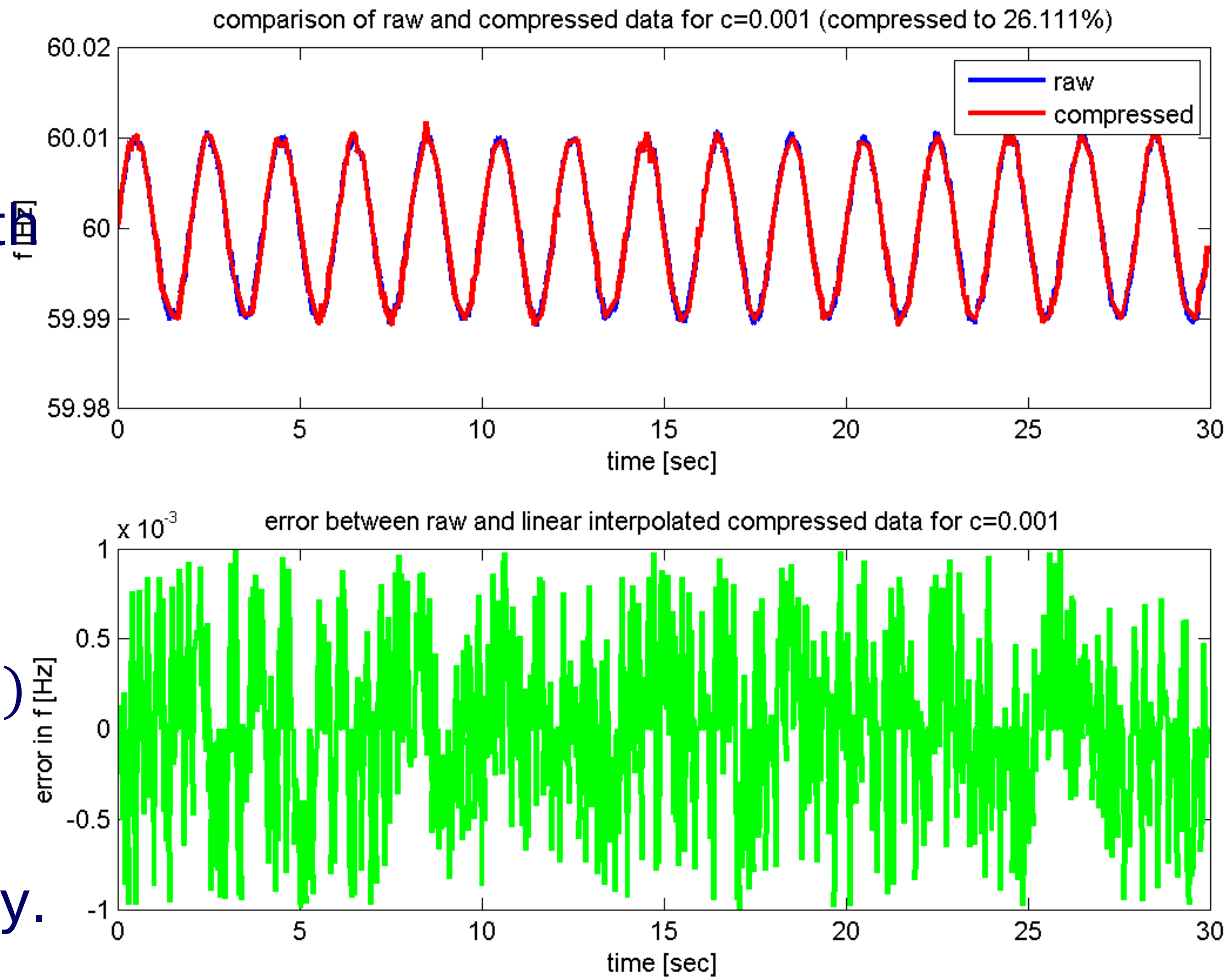
- Single sinusoid with compression level c=0.001
- 900 -> 91 points
- $e_c(k)$ is NOT a "white" noise but periodic!
- Periodic error signal may have several harmonics
- Effect of harmonics in error signal can be seen in spectra!



Another usecase: fixed sine with noise

Data:

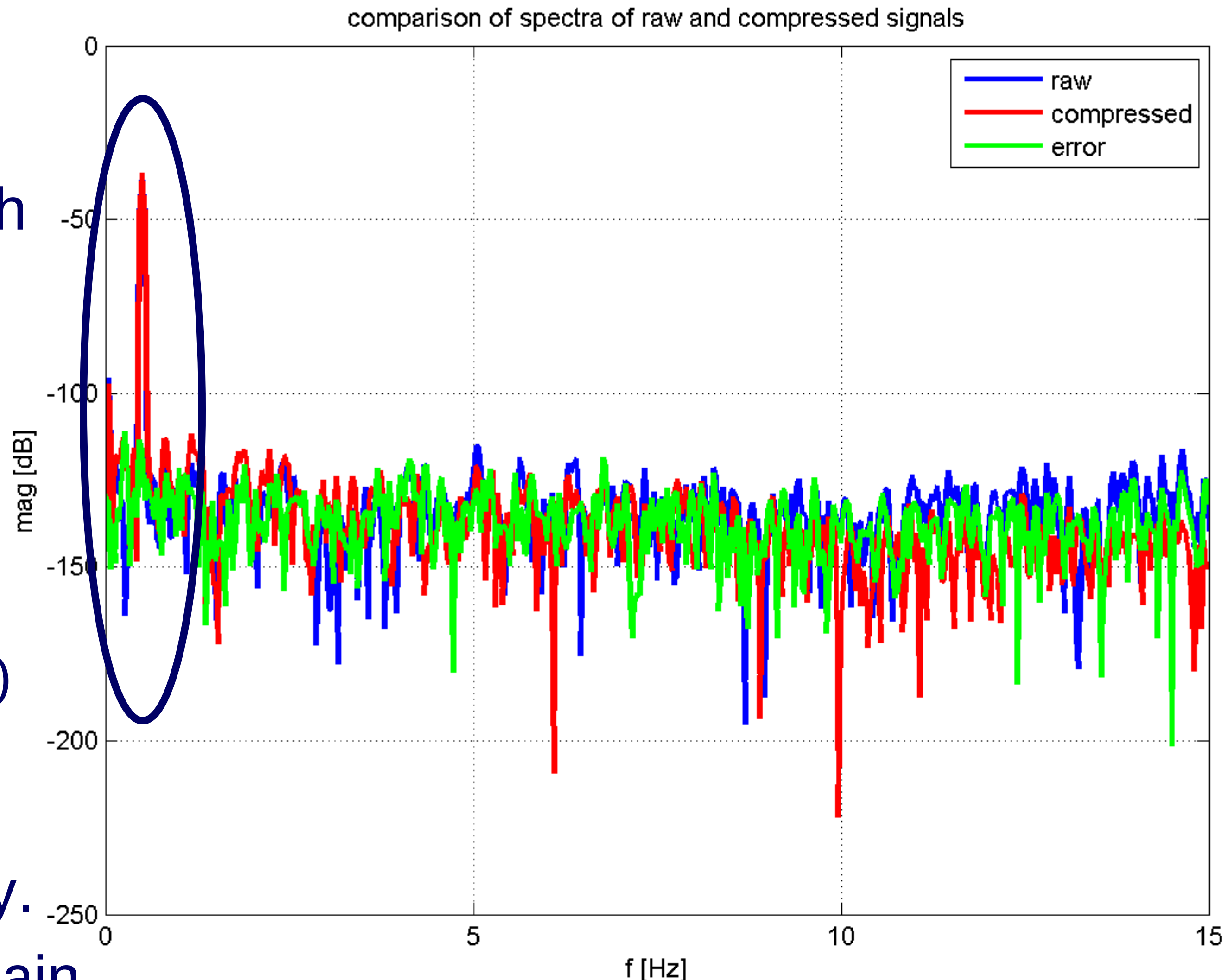
- Single sinusoid with compression level c=0.001
- Typically data has some noise
- 900 -> 235 points
- Due to noise, $e_c(k)$ can be a “white” noise, provided c is picked properly.



Another usecase: fixed sine with noise

Data:

- Single sinusoid with compression level c=0.001
- Typically data has some noise
- 900 -> 235 points
- Due to noise, $e_c(k)$ can be a "white" noise, provided c is picked properly.
- Spectrum is flat again (but noisy)



Wrap up/Conclusions

Main conclusions:

- Compression creates an error signal $|e_c(k)| < c$
- Compression level bounds aliasing as linear (re)interpolation preserves sampling frequency.

Suggestions:

- Compression level c must be chosen such that $e_c(k)$ is a white noise.
- Compression level c must be such that it resembles "noise level" of sensor data.
- For typical frequency application/oscillations c=0.001 gives about 20-25% compression.
-