**Überprüfung der FFT Implementierung in Hadoop.TS.NG**

Die Klasse TimeSeries und TimeSeriesFFT nutzen das Paket Commons-Math für die Berechnung der Fourier-Transformation mittes FFT[[1]](#footnote-1).

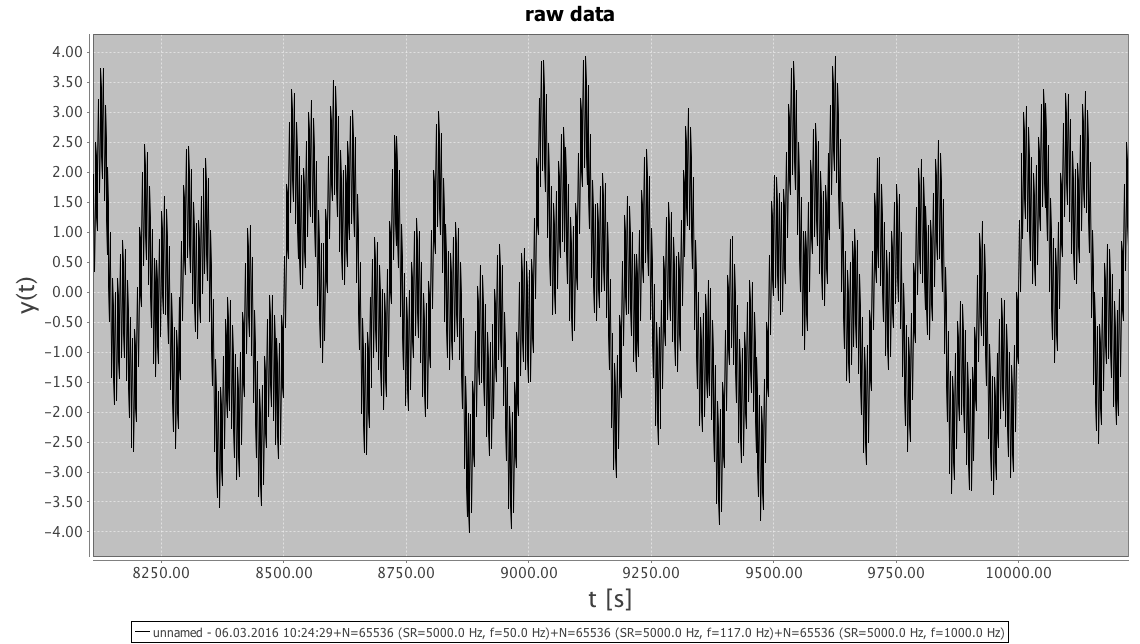
Dieser Funktionstest überprüft, ob das Resultat der Rücktransformation gleich der übergebenen Rohdaten ist. Als Testdaten kommen Sinusfunktionen zum Einsatz. Es wird kein Rauschen hinzugefügt.

Die Länge der Reihe soll einer 2-er Potenz entstprechen. Beispiel: ***l=***2^N mit N= 16 führt zu 65536 Werten. Kürzere Reihen sind am Ende mit 0 aufzufüllen bis die Länge einer 2-er Potenz entspricht.

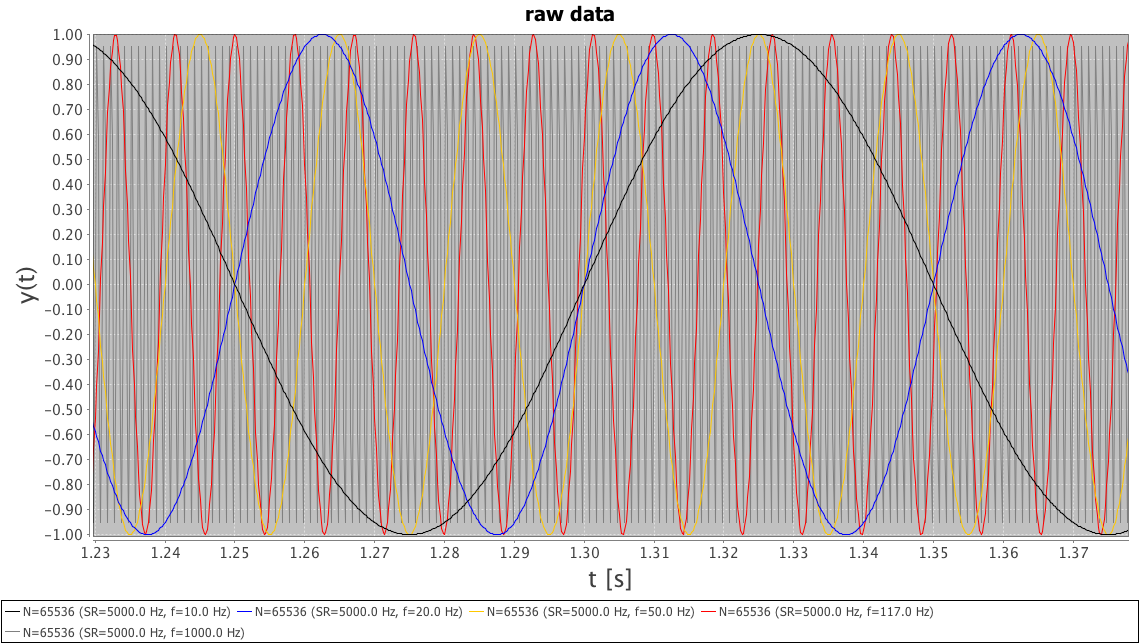
Für die Erstellung von gaussverteilten Zufallszahlen genügt die Angabe der Länge der gewünschten Reihe. Der Generator für Sinusfunktionen benötigt die Frequenz des gewünschten Signals, die Abtastfrequenz ***sr*** (sampling rate) und die Länge des Signals *t* in Sekunden. Wir berechnen t aus: **t = l / sr**. Damit wird eine Reihe der gwünschten Länge generiert.

**Raw Data**

Überlagerte Sinus-Funktionen der Frequenzen f ∈ { 10, 20, 50, 117, 1000 } mit Amplitude 1 werden als Testdatensatz benutzt.



**Fig. 1 :** Eingangssignal aus 5 überlagerten Sinus-Funktionen.

****

**Fig. 2 :** Komponenten des Eingangssignals.

Kontrolle der FFT:

> L = 65536 (Länge der Zeitreihe)

> L\_exp = 65536 (Expandiert)

> N = 65536.0 (Anzahl der Koeffizienten)

> SR = 5000.0 werte/s; // sampling rate

> t = 13.1072 s

c[0]= ( 7.786412504971452, i\*0.0) ) **[Konstanter-Anteil]**

c[1]= ( 7**.78687848859702, i \* - 0.26540412962585846**)

c[N/2 - 1]= ( -0.21887391521529365, i\*1.0477217657678395E-5)

c[N/2]= ( -0.21887391520623245, i\*0.0) **[Nyquist-Anteil]**

c[N/2 + 1]= ( -0.21887391521529365, i\*-1.0477217657678395E-5)

c[N-1]= ( 7.788276765727779, i\*0.5309009709961984)

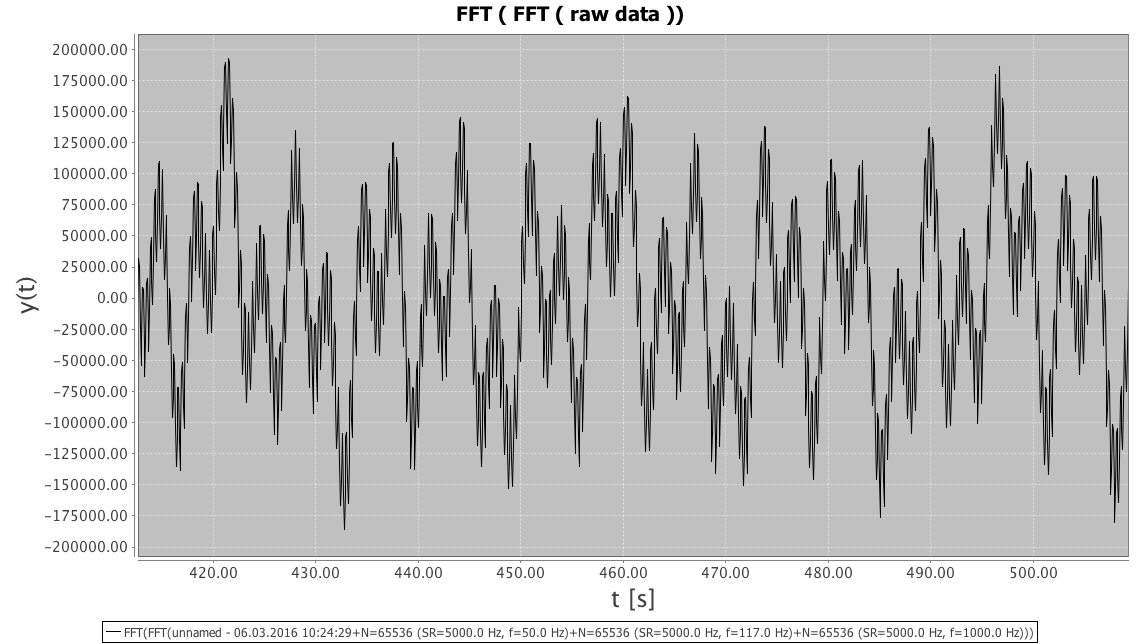
c[N]= ( **7.78687848859702, i \* 0.26540412962585846**)

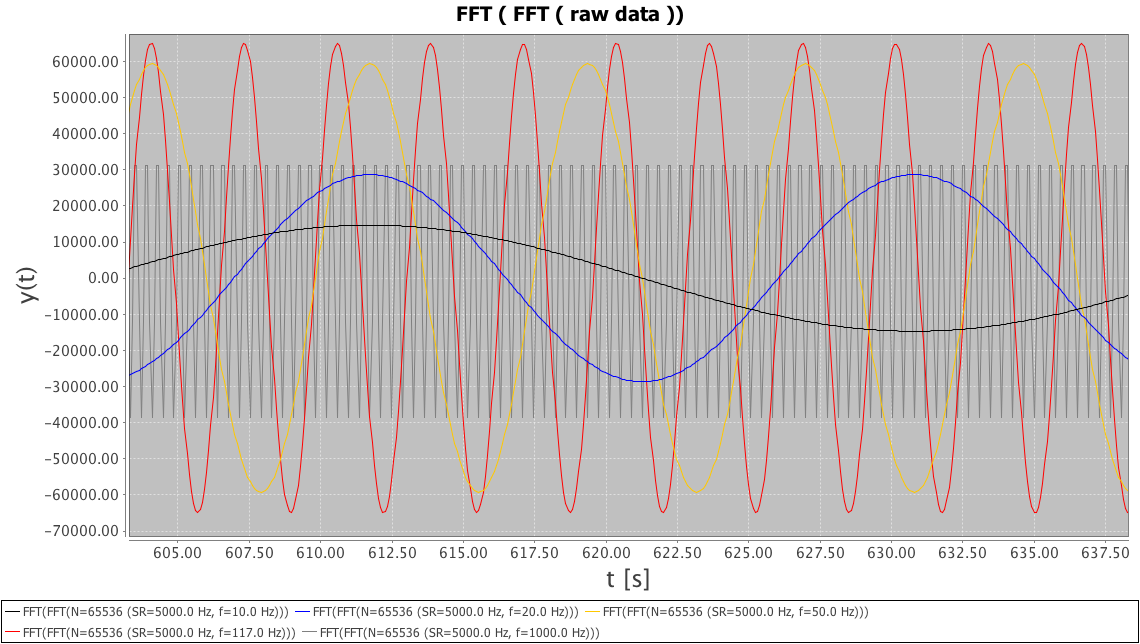
N=65536.0;

f\_max=2500.0;

f\_min=df=0.0762939453125

**Nach Rücktransformation:** FFT ( FFT ( raw data ))





**Das Ergebnis der Rücktransformation stimmt nicht mit dem Eingangssignal überein. Eine Betrachtung der Amplituden der Komponenten zeigt, dass diese nicht korrekt berechnet werden.**

**Frage:** Gibt es hier eine Universal- Normierung? Eine individuelle Normierung der Komponenten wäre hier möglich, da ich die Amplituden des Eingangssignals kenne. Da dies in einem gemessenen Signal nicht der Fall ist, kann ich mir im Moment keine geeignete Korrektur vorstellen.

**Problem:** Amplituden werden nicht korrekt wiedergegeben.

> L = 65536 (Länge der Zeitreihe)

> L\_exp = 65536 (Expandiert)

> N = 65536.0 (Anzahl der Koeffizienten)

> SR = 5000.0 werte/s; // sampling rate

> t = 13.1072 s

> c[0]= ( -4.800313035957515E-9, i\*0.0)

> c[1]= ( 14363.699821064862, i\*-2.8634000059923892E-8)

> c[N/2 - 1]= ( -14696.648112561008, i\*2.8446945243842947E-8)

> c[N/2]= ( -14697.808589574392, i\*0.0) [Nyquist-Anteil]

> c[N/2 + 1]= ( -14696.648112561008, i\*-2.8446945243842947E-8)

> c[N-1]= ( 14401.72200548701, i\*2.6024938204471937E-8)

> c[N]= ( 14363.699821064862, i\*2.8634000059923892E-8)

N=65536.0;

f\_max=2500.0;

f\_min=df=0.0762939453125

Nächster Test:

Erstelle eine Zufallsreihe und transformiere diese direkt ohne dabzu die TimeSeries Klasse zu verwenden.

**RESULTAT:**

Das FFT-Modul von Appache Commons Math3 bietet 2 Methoden:

* transform
* inverstransform

**Unter Verwendung der korrekten Normierung und der richtigen Transformation werden korrekte Amplituden berechnet.**

1. https://commons.apache.org/proper/commons-math/apidocs/org/apache/commons/math3/transform/FastFourierTransformer.html [↑](#footnote-ref-1)