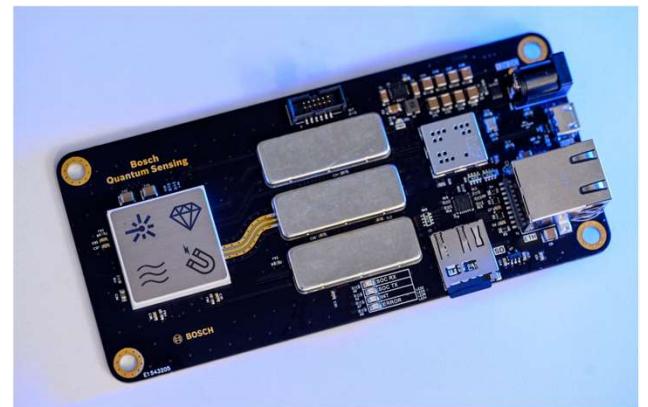
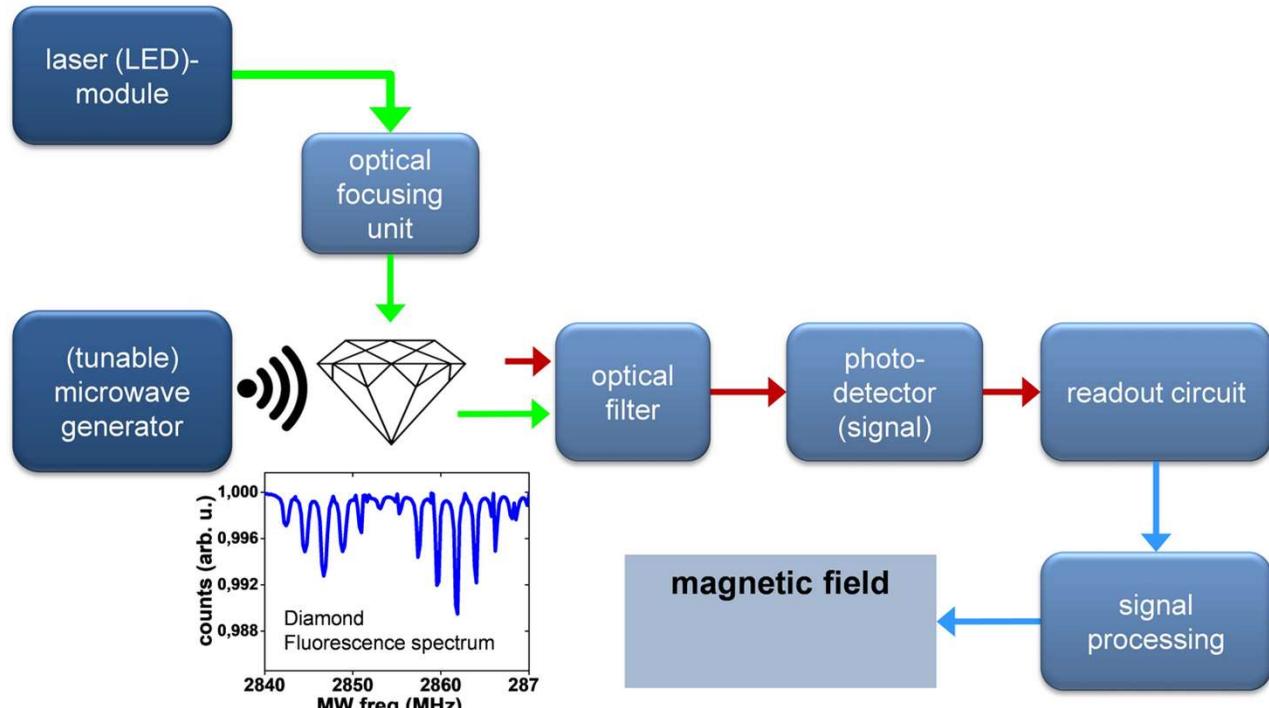


Quantensensor Signal Analyse Pipeline

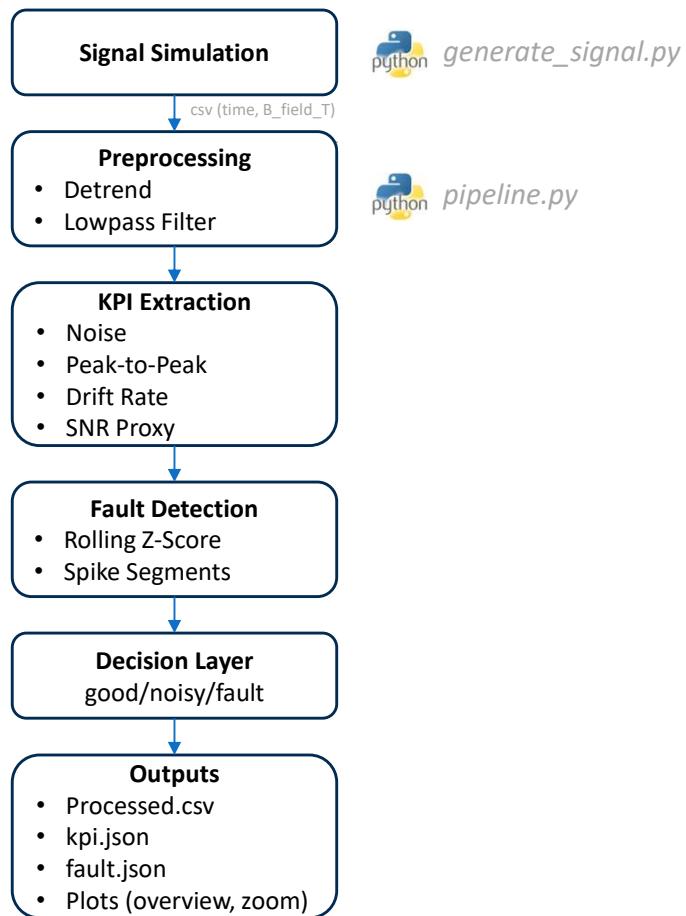
SIMULATIONSGESTÜTZTE ANALYSE VON MAGNETOMETER SIGNALEN
INKLUSIVE RAUSCH-, DRIFT- UND SPIKE-DETEKTION

Funktionsprinzip des Quantensensors



NV-Zentrum basierter Quantensensor (ODMR-Prinzip)

Architekturdiagramm



End-to-end Software Pipeline zur Auswertung von Quantensensor Magnetfeldsignalen inklusive Preprocessing, KPI-Analyse und automatischer Fault Detection.

Simulation des Verhaltens des Magnetfeldsensors (Quantensensor)

Der Spike simuliert einen kurzzeitigen magnetischen Transienten* (z.B EMI, Stromsprung oder Sensoroffset) wie er in realen Fahrzeug- und Quantensensoranwendungen auftritt. Er dient als künstlicher Fault zur Anomalie Erkennung.

Feature-/ KPI Extraktion & Szenarien Labeling

- Signalbasierte KPI-Extraktion (σ , p2p, drift, spikes)
- Labelbare Szenarien (good/ noisy/ fault)
- Grundlage für regelbasierte Qualitätsklassifikation (Thresholds und Heuristiken)

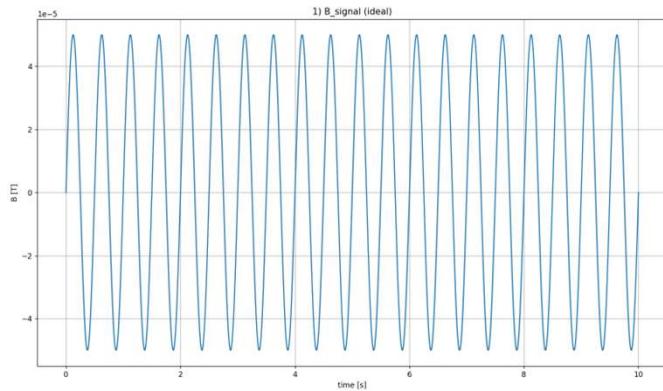
Regelbasierte Anomalieerkennung und Qualitätsklassifikation:

- Noise Bewertung $\sigma \neq \text{noisy}$ (Photon Shot Noise, NV-Fluoreszenzstatistik, Elektronik Noise)
- Drift Bewertung $\text{drift_rate} \neq 0$ (Temperaturänderung, Alterung, Versorgungsspannung driftet, Magnetische Umgebung)
- Transienten Erkennung: *spike_count*, *spike_segments* (plötzlicher Sprung, EMI burst: Relais schaltet, Stromsprung, ADC Glitches)

* Beispiel: ABS Magnet Pickup Störung, Radgeschwindigkeit Störimpuls, IMU transient, NV-Sensor magnetic glitch

Datengenerierung (Signal-Simulation)

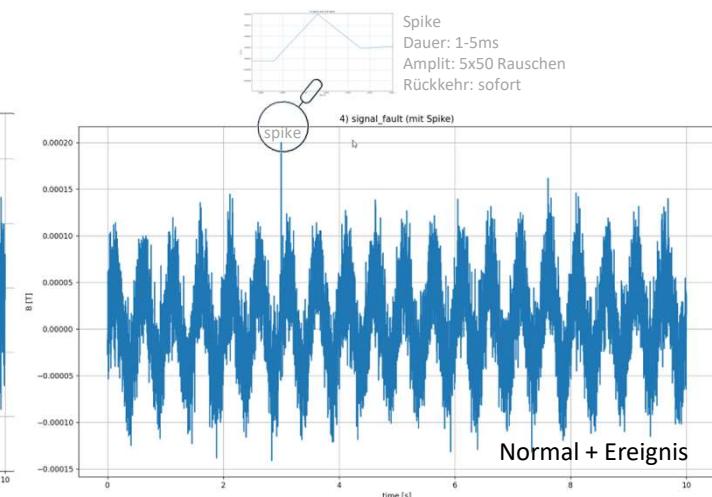
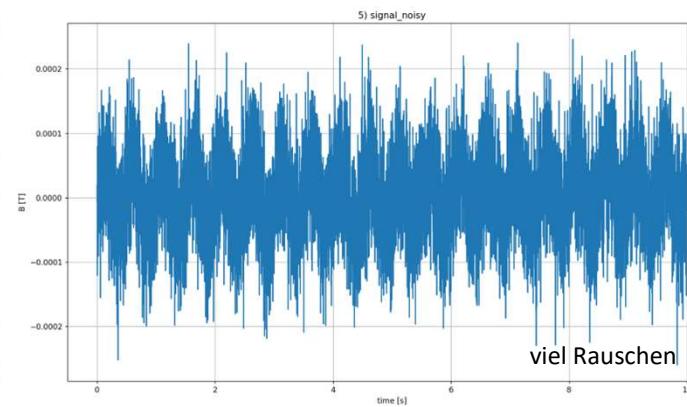
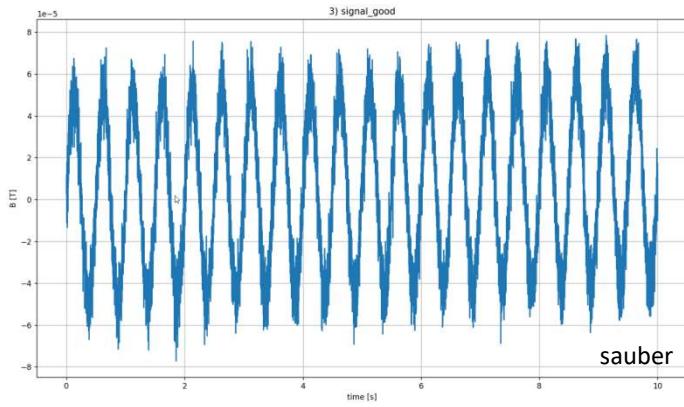
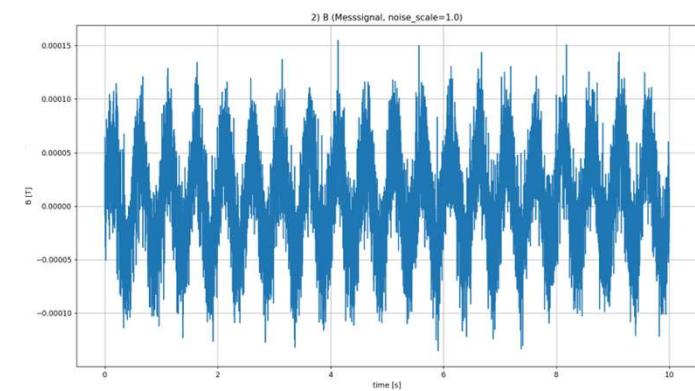
 generate_signal.py



integration_time = 1e-2
 integration_time ↑ → Noise ↓ → SNR ↑

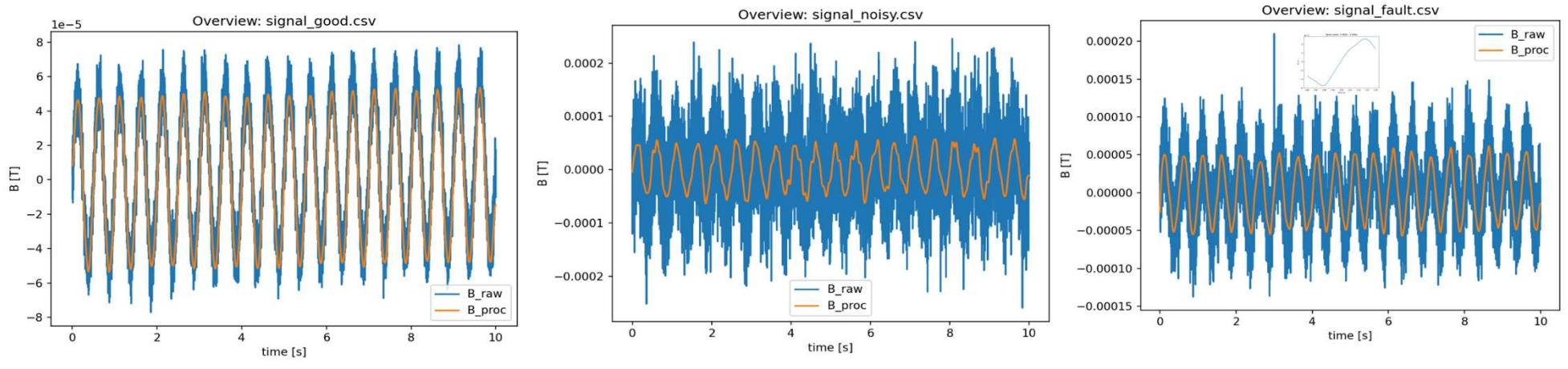
Simulation berücksichtigt:

- Quantenschussrauschen (integration_time)
- Drift (Temperatur/ Versorgung)
- Kurzzeit-Transienten (EMI, Stromsprung)



Realistische Basis für Inbetriebnahme-Test

Preprocessing & KPI Analyse



Linear Detrend (drift entfernen) und Lowpass Filter (Noise Suppression)

- **σ (Noise level): Standardabweichung des Signals**
- **Peak-to-Peak** – Maximale Feldvariation
- **Drift Rate** – Feldänderung pro Sekunde
- **SNR-Proxy**: $p2p/(2\sigma)$

Scenario	sigma [T]	Drift [T/s]	snr_proxy
good	3.539832	-1.697449e-08	1.516275
noisy	3.564075	5.301185e-09	1.665621
fault	3.495511	1.469641e-08	1.811625

Quantitative Bewertung von Signalqualität und Sensorstabilität

Faults detection

Spike Detection Pipeline:

1. Ableitung bilden ΔB ; $db[i] = b[i] - B[i-1]$
2. Rolling Z-Score; man vergleicht jede Änderung mit dem lokalen Rauschpegel
3. Threshold
4. Segmentierung

Ergebnisse:

- **spike_count** – Anzahl der Spikes
- **spike_segments** – Zeitfenster der Anomalie

Scenario	spike_count	spike_segments
good	0	-
noisy	0	-
fault	1	3.0003

Summary

Entwicklung einer End-to-End Software Pipeline zur:

- ✓ Simulation von Quantensensor-Signalen
- ✓ Signalaufbereitung (Preprocessing)
- ✓ Physikalisch motivierter KPI-Extraktion
- ✓ Automatischer Spike-basierter Fehlerdetektion

Fazit

- ✓ Automatisierte Sensor-Inbetriebnahme
- ✓ Robuste Fehlererkennung
- ✓ Übertragbar auf reale Quantensensoren