

FAQ: Wie entsteht ein Maschinenzustandsdatenverständnis?

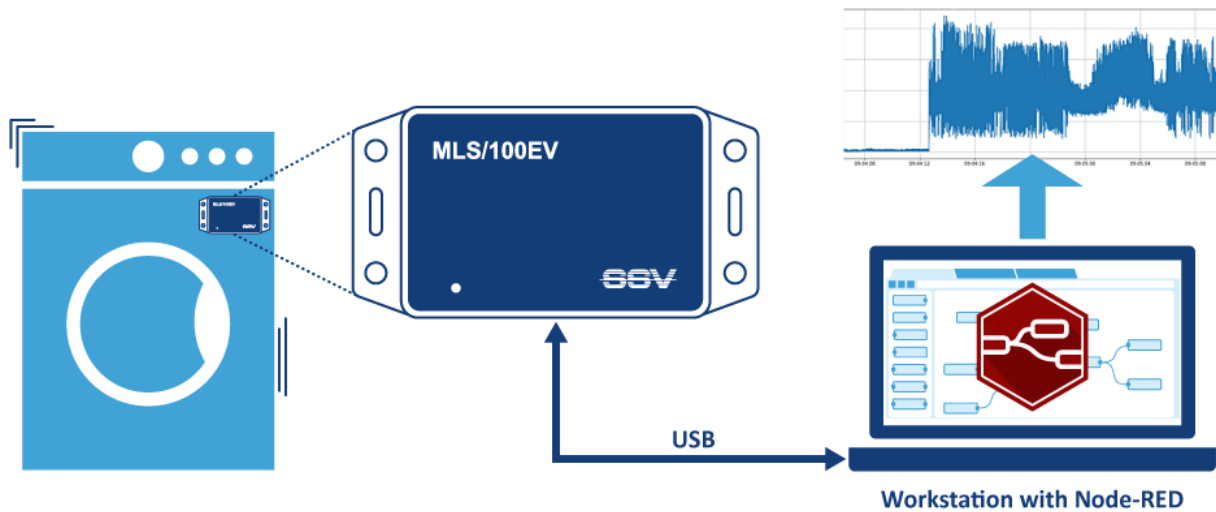


Abbildung 1: Der MLS/100EV wird z. B. an einer Maschine mit einem rotierenden Antriebselement befestigt. Die Schwingungen des Maschinenbetriebs liefern Messdaten, die per USB-Verbindung an einen PC übertragen und dort mit Hilfe einer Node-RED-Anwendung in einer CSV-Datei gespeichert werden. Die anschließende Analyse und Visualisierung der erfassten CSV-Daten lässt sich beispielsweise in einer Python-Laufzeitumgebung, wie Google Colab, durchführen.

Frage: Wofür wird ein Maschinenzustandsdatenverständnis benötigt?

Antwort: Die Erfassung und Analyse von Maschinendaten ist die Voraussetzung für verschiedene Optimierungsaufgaben im Umfeld von Maschinen und Anlagen. Hier einige Beispiele, für die qualitativ hochwertige Zustandsdaten und die damit verbundene Transparenz benötigt werden:

- **Optimierung der Produktion:** Durch die Datenanalyse können Engpässe in der Produktion identifiziert und Optimierungen vorgenommen werden.
- **Effizienzsteigerung:** Durch die Analyse von Maschinendaten lassen sich Prozesse optimieren, um unnötige Ausgaben zu vermeiden (z. B. Betriebskostenoptimierung durch die Reduzierung von Energiekosten für Strom, Gas, Wasser, Druckluft usw.).
- **Predictive Maintenance/Condition Monitoring:** Die Analyse von Maschinendaten ermöglicht es, potenzielle Fehler frühzeitig zu erkennen und durch die Optimierung der erforderlichen Wartungsaufgaben proaktiv zu beheben, bevor sie zu Ausfällen führen.
- **Verbesserte Verfügbarkeit:** Durch periodische Zustandsdatenanalysen, die mit Hilfe geeigneter (KI/ML-) Algorithmen automatisch durchgeführt werden, lassen sich Wartungstermine besser planen und präventive Wartungsmaßnahmen durchführen, um Ausfallzeiten zu reduzieren bzw. soweit möglich vollständig zu vermeiden.

Frage: Wer benötigt ein Maschinenzustandsdatenverständnis?

Antwort: Ein solches Datenverständnis sollte sowohl bei einem Maschinenbauer als auch beim Maschinenbetreiber vorhanden sein. Auch Unternehmen, die sich auf die Wartung von Maschinen und Anlagen spezialisiert haben, profitieren von einem Maschinenzustandsdatenverständnis. Die einen verbessern damit ihre Wettbewerbsfähigkeit – beispielsweise der Maschinenbauer und das Wartungsunternehmen durch Zustandsdaten-abhängige Serviceangebote für die jeweiligen Kunden. Die

anderen, also die Maschinen- und Anlagenbetreiber, können das Datenverständnis und die damit verbundene Transparenz für verschiedene Betriebskostenoptimierungen nutzen.

Frage: Wie funktioniert das MLS/100EV-Maschinenzustandsdatenverständnis-Beispiel?

Antwort: Die erforderliche Firmware ist bei der MLS/100EV-Auslieferung bereits werksseitig vorinstalliert. Wird der MLS/100EV über eine USB-Verbindung mit Spannung versorgt, sendet die Firmware ca. alle 1,5 Sekunden einen CSV-getrennten Datensatz mit 22 Elementen über die USB-Verbindung an das jeweilige Rechnersystem mit der USB-Host-Schnittstelle. Diese CSV-Daten entstehen im MLS/100EV durch eine periodische Messung der Beschleunigungsmesswerte von drei (x/y/z) IMU-Achsen (IMU = Inertial Measurement Unit, also ein Beschleunigungssensor als Halbleiterbaustein im MLS/100EV; gemessen wird im Maschinenzustandsdatenverständnis-Beispiel somit die dreidimensionale Beschleunigung einer Vibration, die beispielsweise durch ein rotierendes Antriebselement verursacht wird).

Nach jeder Messung werden eine Fouriertransformation (FFT = Fast Fourier Transformation) sowie weitere Berechnungen durchgeführt. Dabei wird u. a. auch jeweils ein Datenelement mit dem quadratischen Mittel für jede Achse (x_{rms} , y_{rms} und z_{rms} – also praktisch ein Effektivwert; Englisch: Root Mean Square, RMS) erzeugt.

Um die MLS/100EV-Daten in einer CSV-Datei zu erfassen, wird ein PC mit einer Node-RED-Installation inklusive des Node-RED-Dashboards sowie einem „serialport“-Node benötigt. In diese Node-RED-Installation wird dann der „SSV/MLS CSV Data Logger“ importiert. Dieser Node-RED-Flow erfüllt drei Aufgaben: 1. Erfassen der vom MLS/100EV per serieller (USB-) Schnittstelle empfangener Daten in einer CSV-Datei. 2. Anzeige des vollständigen MLS/100EV-CSV-Datensatzes mit allen 22 Elementen im Node-RED-Debug-Fenster. 3. Echtzeitanzeige der x_{rms} -Daten im Node-RED-Dashboard.

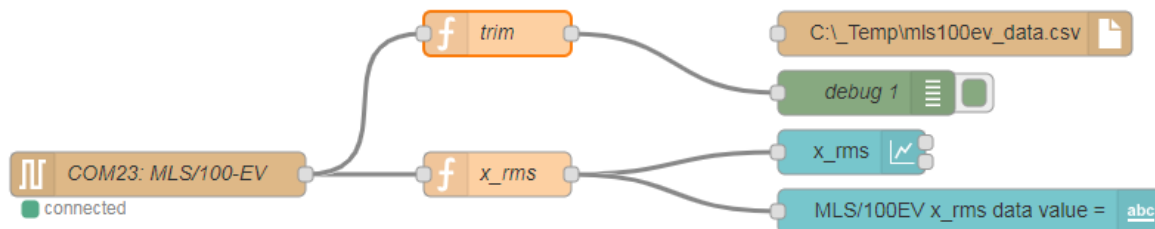


Abbildung 2: CSV-Dateierfassung ausgeschaltet. Die vom MLS/100EV empfangenen Messdaten werden im Node-RED-Debug-Fenster angezeigt. Die x_{rms} -Werte werden im Node-RED-Dashboard als Liniendiagramm dargestellt.

Start und Stopp der CSV-Dateierfassung auf dem PC lässt sich jeweils über die Verbindungslinie zwischen dem „trim“- und „CSV-Datei“-Node steuern: Zum Erfassen wird die Verbindung hergestellt und die Node-RED-Deploy-Schaltfläche betätigt. Der Stopp erfolgt durch das Löschen dieser Verbindung und einer weiteren Deploy-Schaltflächenbetätigung.

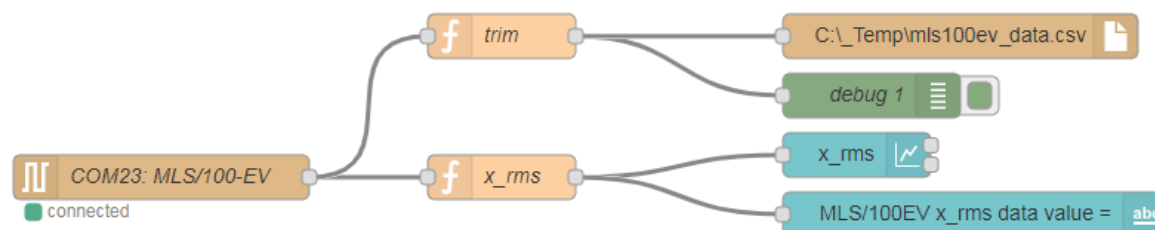


Abbildung 3: CSV-Dateierfassung eingeschaltet. Jeder vom MLS/100EV an den PC übertragene CSV-Datensatz wird im Append-Modus in einer CSV-Datei gespeichert.

Frage: Wofür ist das Liniendiagramm im Node-RED-Dashboard vorgesehen?

Antwort: Diese Darstellung soll dem Anwender dabei helfen, einen geeigneten Montagepunkt für den MLS/100EV an einem Maschinengehäuse bzw. in einer Anlage zu ermitteln. Man sieht sofort ob der jeweils gewählte Montagepunkt ein Beschleunigungsdatenbild liefert, das sich einem bestimmten Maschinenzustand zuordnen lässt bzw. ob sich über die Daten des Liniendiagramms unterschiedliche Zustände differenzieren lassen. Nur dann ist die Datenerfassung über einen längeren Zeitraum in einer CSV-Datei sinnvoll. Ansonsten sollte der MLS/100EV an einem Punkt montiert werden, der ein besseres Datenbild liefert.

MLS/100EV Machine Condition Data Understanding

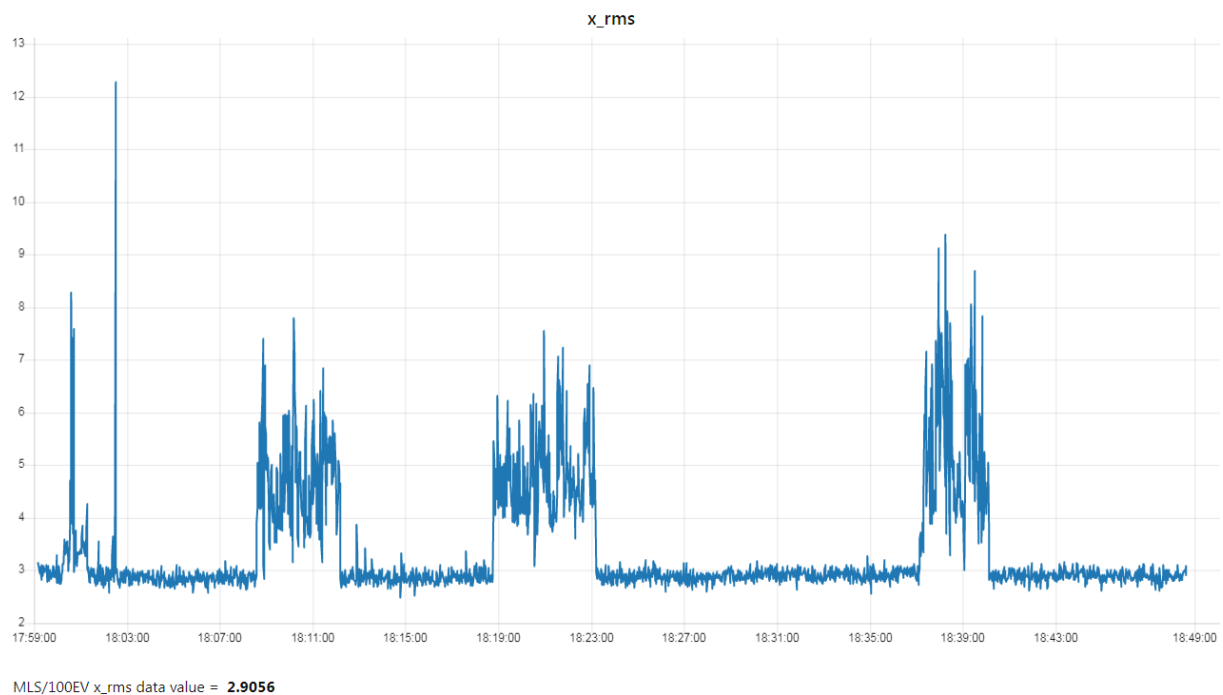


Abbildung 4: Beispiel für die dynamische Echtzeitanzeige der x_{rms} -Werte per Liniendiagramm in einem Node-RED-Dashboard für die jeweils letzte MLS/100EV-Betriebsstunde. Die Daten-Updates des Diagramms erfolgen ca. alle ein bis zwei Sekunden.

Anmerkung zur Abbildung 4: Falls Sie bei der ersten Inbetriebnahme eines MLS/100EV keinen Zugriff auf eine geeignete Maschine haben, lässt sich z. B. auch die Lautsprecherbox einer Audioanlage als Vibrationsdatenquelle nutzen: Für die Abbildung wurde der MLS/100EV mit doppelseitigem Klebeband auf ein DAB-Radiogerät geklebt. Anschließend wurde das Radio per Fernbedienung mehrfach ein- und ausgeschaltet.

Frage: Wie erfolgt die Analyse und Visualisierung einer per Node-RED erfassten CSV-Datei mit MLS/100EV-Daten unter Google Colab?

Antwort: Die einzelnen Schritte der Vorgehensweise sind im Text *FIRST STEPS WITH MLS/100EV* im MLS/100EV-Github-Repository relativ ausführlich beschrieben. Hier eine Übersicht der erforderlichen Aktivitäten:

- Das Python-Codefragment *analyze.ipynb* aus dem MLS/100EV-Github-Repository auf den eigenen PC und von dort aus in eine Google-Colab-Sitzung hochladen und als s. g. *Python Notebook* starten.

- Die auf dem PC per Node-RED erfasste CSV-Datei mit MLS/100EV-Daten in den Dateibereich der Google-Colab-Sitzung hochladen, damit der Python-Code auf die Daten zugreifen kann.
- Die einzelnen Codezellen der Reihe nach ausführen und die Ausgaben beobachten.

Google Colab enthält einen KI-basierten Assistenten, um weiteren Code zur Datenanalyse und Visualisierung der MLS/100EV-CSV-Daten zu erzeugen. Dafür werden s. g. *Prompts* eingegeben. Beispiele dazu werden wir in weiteren Versionen dieser FAQ veröffentlichen.

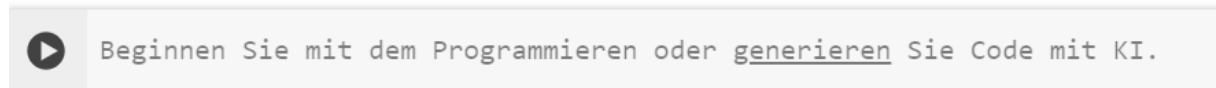


Abbildung 5: Eingabeaufforderungen des KI-Assistenten unter Google Colab. Erzeugen Sie eine zusätzliche Codezelle und klicken Sie auf den Link generieren. Geben Sie dann z. B. den Text „Erzeuge einen Scatter Plot mit den Daten zu x_rms und y_rms.“ als Prompt ein und bringen Sie den vom KI-Assistenten erzeugten Code zur Ausführung.

Frage: Wo finde ich eine Beschreibung der 22 Elementen des MLS/100EV-CSV-Datensatzes?

Antwort: Die einzelne Datenelemente sind in der folgenden Tabelle beschrieben.

Name	Kurzbeschreibung
time	Erfassungszeitpunkt in Millisekunden seit dem letzten MLS/100EV-System-Reset.
x_rms	IMU x-Achse: Quadratischer Mittelwert aller x-Achsenmesswerte.
x_freq_0	IMU x-Achse: Frequenz des höchsten Peaks in der FFT der x-Achse.
x_mag_0	IMU x-Achse: Amplitude des höchsten Peaks in der FFT der x-Achse.
x_freq_1	IMU x-Achse: Frequenz des zweithöchsten Peaks in der FFT der x-Achse.
x_mag_1	IMU x-Achse: Amplitude des zweithöchsten Peaks in der FFT der x-Achse.
x_freq_2	IMU x-Achse: Frequenz des dritthöchsten Peaks in der FFT der x-Achse.
x_mag_2	IMU x-Achse: Amplitude des dritthöchsten Peaks in der FFT der x-Achse.
y_rms	IMU y-Achse: Quadratischer Mittelwert aller y-Achsenmesswerte.
y_freq_0	IMU y-Achse: Frequenz des höchsten Peaks in der FFT der y-Achse.
y_mag_0	IMU y-Achse: Amplitude des höchsten Peaks in der FFT der y-Achse.
y_freq_1	IMU y-Achse: Frequenz des zweithöchsten Peaks in der FFT der y-Achse.
y_mag_1	IMU y-Achse: Amplitude des zweithöchsten Peaks in der FFT der y-Achse.
y_freq_2	IMU y-Achse: Frequenz des dritthöchsten Peaks in der FFT der y-Achse.
y_mag_2	IMU y-Achse: Amplitude des dritthöchsten Peaks in der FFT der y-Achse.
z_rms	IMU z-Achse: Quadratischer Mittelwert aller z-Achsenmesswerte.
z_freq_0	IMU z-Achse: Frequenz des höchsten Peaks in der FFT der z-Achse.
z_mag_0	IMU z-Achse: Amplitude des höchsten Peaks in der FFT der z-Achse.
z_freq_1	IMU z-Achse: Frequenz des zweithöchsten Peaks in der FFT der z-Achse.
z_mag_1	IMU z-Achse: Amplitude des zweithöchsten Peaks in der FFT der z-Achse.
z_freq_2	IMU z-Achse: Frequenz des dritthöchsten Peaks in der FFT der z-Achse.
z_mag_2	IMU z-Achse: Amplitude des dritthöchsten Peaks in der FFT der z-Achse.

Tabelle 1: Namen und Kurzbeschreibung der 22 Datenelemente des MLS/100EV-CSV-Datensatzes.