

RMG/SSDK1: Softsensorentwicklungs-kit mit RMG/941 und MLS/160

Digitalisierungsaufgaben mit Maschinen und Anlagen benötigen hochwertige Zustandsdaten. Beim gegenwärtigen Stand der Technik lassen sich diese Daten nur sehr selten direkt aus den jeweiligen Steuerungen auslesen. Stattdessen müssen sie über eine geeignete IoT-Retrofit-Sensorik indirekt gewonnen werden. Eine typische Beispielanwendung wäre die Überwachung einer Produktionsmaschine für eine Condition-Monitoring-Lösung, um mit Hilfe eines Softsensors ungewöhnliche Betriebszustände zu erkennen. An Hand der jeweiligen Zustände lassen sich zum Beispiel frühzeitige Warnungen erzeugen, um auf den drohenden Ausfall hinzuweisen. Alternativ könnte ein Softsensors sogar die Restnutzungsdauer eines Bauelements bis zur nächsten fälligen Wartung berechnen.



Messtechnisch kann man von der Annahme ausgehen, dass praktisch jede Maschine eine individuelle Schwingungscharakteristik aufweist, die sich mit geeigneten Sensorelementen erfassen lässt. Insofern könnte man die Schwingungen einer Maschine im dreidimensionalen Raum (X-, Y- und Z-Achse) mit einem entsprechenden Sensorelement messen und über einen Machine-Learning-Algorithmus (beispielsweise ein neuronales Netzwerk oder multivariantes Verfahren des statistischen Lernens) aus den Sensorikmessdaten (Rohdaten) die erforderliche Softsensorausgangsinformation bilden.

Was ist ein Softsensor?

Laut Wikipedia ist ein Softsensor kein real existierender Sensor, sondern eine Modell-basierte Abhängigkeitssimulation mit stellvertretenden Messgrößen und einer (nicht direkt messbaren) Ausgangsinformation. Ein solcher Softsensor basiert somit auf der Abhängigkeit von korrelierenden „Eingangs-

messwerten“ zu einer Zielgröße am Ausgang des virtuellen Sensors. Als Eingangsdaten kommen sowohl die Ausgangsdaten realer Sensoren als auch Prozessvariable aus Steuerungen oder anderen geeigneten Datenquellen in Frage.

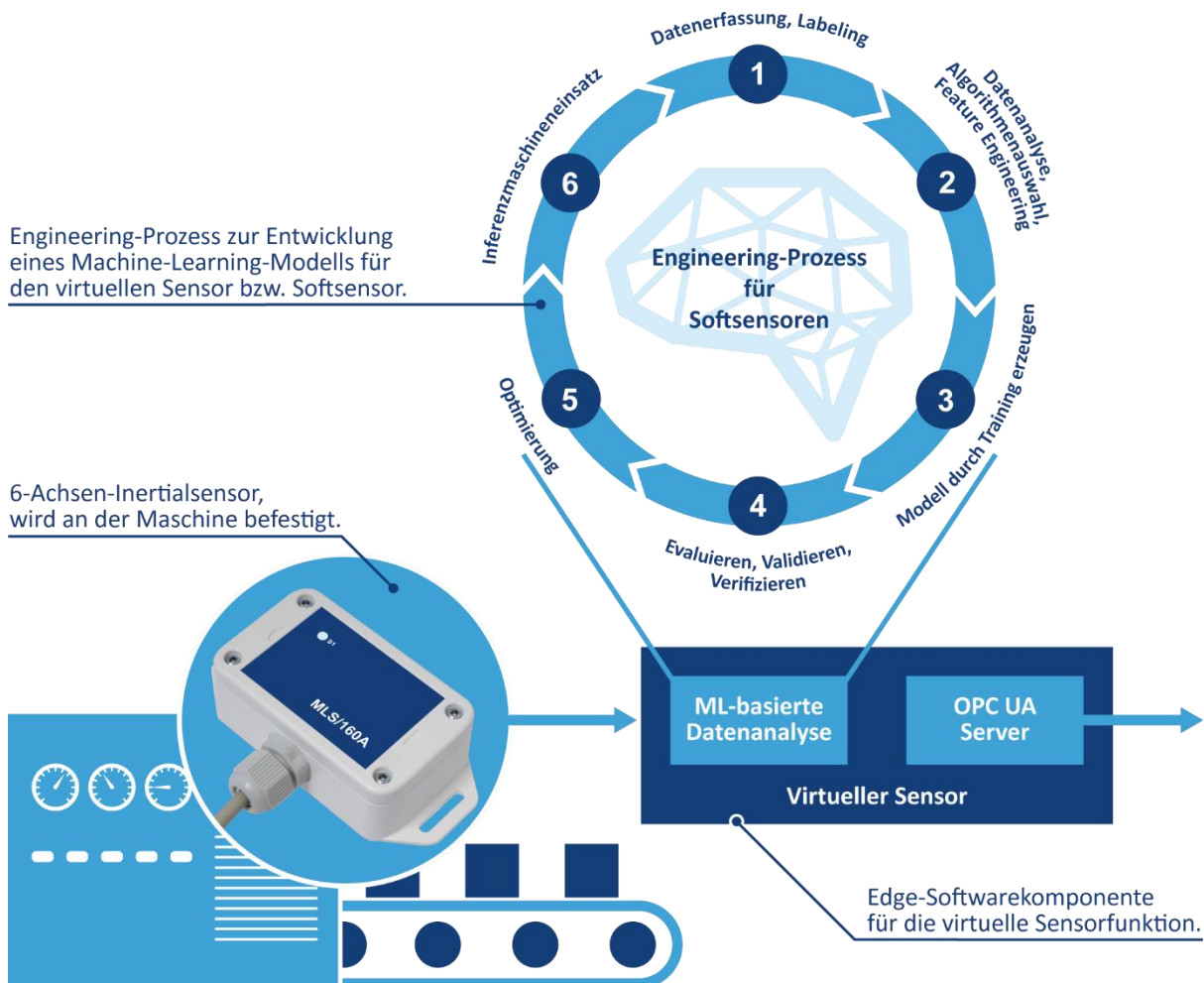


Abbildung 1: Durch das Nachrüsten einer passenden Sensorlösung lässt sich eine Produktionsmaschine mit einer Condition-Monitoring-Lösung überwachen. Als Sensorik eignen sich virtuelle Sensoren (Softsensoren), die mit Hilfe von neuronalen Netzen oder multivariater Verfahren des statistischen Lernens die Ausgangsdaten eines Sensorelements in die gewünschte Zielgröße umrechnen und anderen Anwendungen z. B. als OPC UA-Datenpunkt zur Verfügung stellen.

Ein Softsensor, den die meisten von uns häufig nutzen, ist z. B. der GPS-Sensor eines Navigationssystems oder Smartphones. Die Position unseres aktuellen Standorts lässt sich auch nicht direkt messen. Sie wird stattdessen per Trilateration – also die Entfernungs- bzw. Abstandsmessung zu drei Punkten – aus den empfangenen Signalen mehrerer Satelliten gebildet. Da GPS-Sensoren heute als Halbleiterchips und somit auf den ersten Blick als real existierende Sensoren zur Verfügung stehen, bezieht sich der Begriff „Softsensor“ in erster Linie auf die Methode, wie aus den Eingangsdaten die jeweilige Ausgangsinformation gewonnen wird. Gebräuchlich sind sowohl analytische als auch empirische Verfahren. Im ersten Fall wird der Sensorausgangswert durch ein programmiertes Ver-

fahren berechnet, wie beispielsweise bei einem GPS-Sensor. Der empirische Ansatz basiert auf Historiendaten, also „Erfahrungswerten“, die zur Modellbildung dienen. An Hand des Modells mit dem „Erfahrungswissen“ wird dann jeweils aus den Echtzeit-Eingangsdaten per Klassifizierung oder Regression die Ausgangsinformation gebildet.

Varianten des maschinellen Lernens

Machine-Learning-Algorithmen gehören zum Themenbereich der künstlichen Intelligenz (KI). Da die wesentlichen KI-Grundlagen und somit des maschinellen Lernens aus den 50er Jahren stammen, gibt es mittlerweile eine schwer überschaubare Methoden- und Algorithmenvielfalt. Für lernfähige Softsensoren eignen sich besonders zwei Verfahren:

Supervised Machine Learning: Überwachtes maschinelles Lernen. Die meisten der gegenwärtig in der Praxis genutzten Machine-Learning-Algorithmen, wie zum Beispiels CNNs (Convolutional Neural Networks) gehören zu dieser Kategorie. Bei diesem empirischen Verfahren ist der Zusammenhang zwischen den Eingangs- und Ausgangsgrößen anhand von Historiendaten im Vorfeld bekannt. Insofern haben wir es mit einem Mapping der Eingangsdaten auf den Ausgang zu tun. Der jeweils zum Einsatz kommende Algorithmus muss zunächst trainiert werden. Dabei entsteht ein Modell. Für die Trainingsphase werden gelabelte (Historien-) Daten benötigt. Bei einer sinnvollen Abstimmung zwischen den Daten und dem jeweils gewählten Algorithmus sowie einer ausreichenden Trainingsdatenmenge von guter Qualität lassen sich anschließend mit Hilfe bisher unbekannter Datenwerte relativ genaue diskrete Klassifizierungs- oder kontinuierliche Regressionswerte vorher-sagen. Typische Anwendungsbeispiele für überwachtes Lernen sind die Objekterkennung in Bilddaten (Mustererkennung) und die Vorhersage des Energiebedarfs einer Maschine. Die in der Trainingsphase des Supervised Machine Learnings entstehenden Modelle sind statisch und müssen bei Bedarf durch ein erneutes Training an veränderte Bedingungen angepasst werden.

Unsupervised Machine Learning: Unüberwachtes maschinelles Lernen. Dieses Verfahren wird bei Bedarf auf Daten mit unbekanntem Zusammenhängen angewendet, um in den Eingangsdaten mit Rechnerunterstützung nach Mustern (Clustern) und den Grenzen zwischen den gefundenen Clustern zu suchen. Insofern spricht man bei dieser Kategorie des maschinellen Lernens auch häufig von Clusteranalysen, also die Zuordnung der vorliegenden Datenpunkte zu bestimmten Gruppen (den Clustern). Die in der Mathematik zur Verfügung stehenden Methoden nutzen unterschiedliche Ansätze, wie z. B. das prototypische Bilden von Cluster-Zentren in einem n-dimensionalen kontinuierlichen Raum oder Dichte-basierte Regionen als Zentrum eines möglichen Clusters. Darüber hinaus gibt es auch noch hierarchische Clusterverfahren. Die gefundenen Ähnlichkeitsstrukturen und die per

Clusteranalyse festgelegten Gruppen können in der Praxis als Vorlage zum Labeln der bisher unbekannt Daten für ein anschließendes Supervised Machine Learning dienen.

In drei Schritten zum lernfähigen Sensor

Im Gegensatz zu konventionellen Sensoren, die in der Regel betriebsbereit ausgeliefert werden, muss ein (virtueller) Softsensor zunächst für die jeweilige Aufgabenstellung in Betrieb genommen werden. Für einen Maschinensensor mit MEMS-Inertialsensorelementen sind dafür beispielsweise lediglich drei Einzelschritte erforderlich.

In einem ersten Schritt wird die Sensor-Hardware fix an der Maschine befestigt, um permanent die Vibrationen im dreidimensionalen Raum zu messen. Dabei ist zu beachten, dass die Position des Sensorelements während des späteren Praxiseinsatzes nicht mehr verändert werden darf, weil die für ein Modell erzeugten Messdaten zu einer bestimmten X-, Y- und Z-Achsausrichtung gehören. Jede direkte oder indirekte Veränderung der Sensorposition erfordert daher jeweils ein neues Modell.

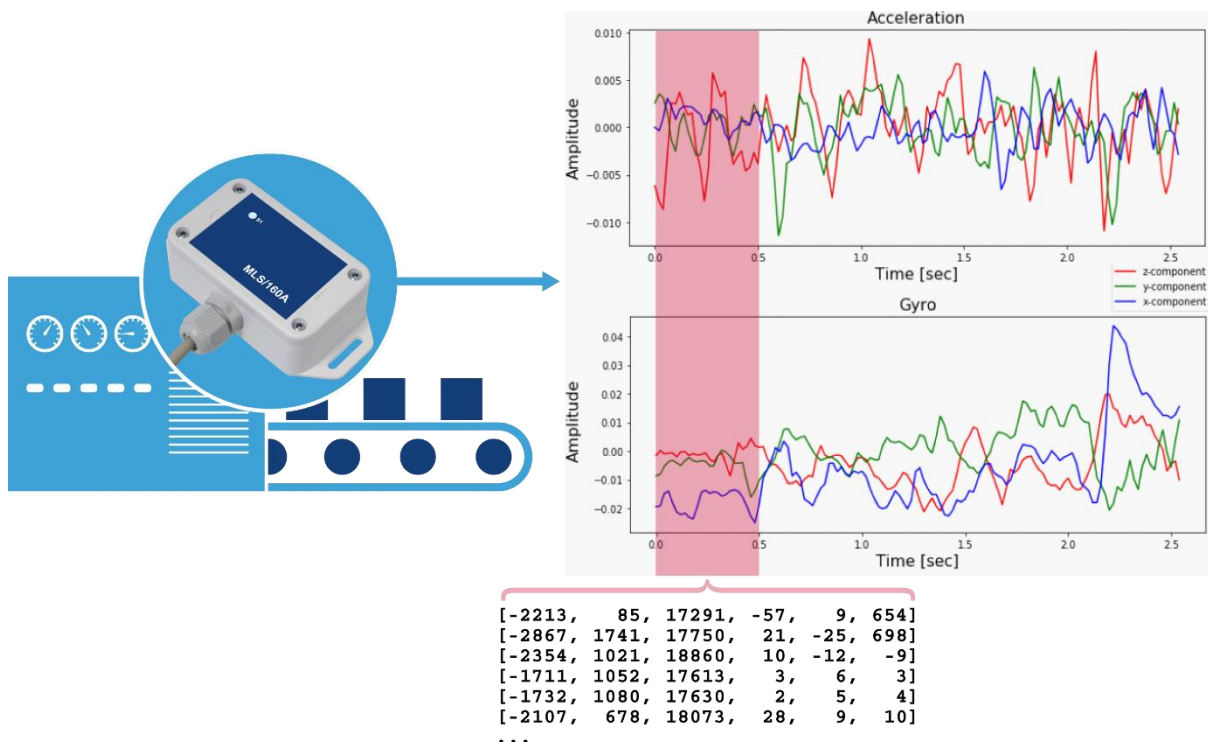
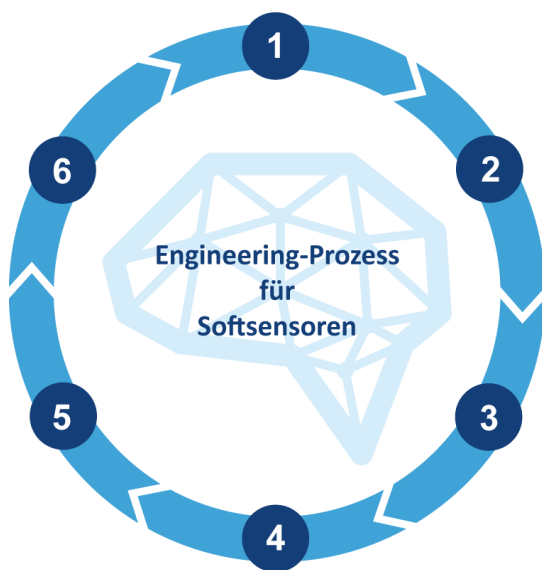


Abbildung 2: Die 3D-Messdaten des Inertialsensors zu Winkelgeschwindigkeit und Beschleunigung werden mit einer bestimmten Frequenz abgetastet und digitalisiert. Jede einzelne abgetastete Messung erzeugt jeweils eine Liste mit sechs Messwerten. Drei Messwerte repräsentieren die Beschleunigung in X/Y/Z-Richtung, die weiteren drei Zahlen die Winkelgeschwindigkeit im dreidimensionalen Raum. Für das maschinelle Lernen bildet jede Liste einen Merkmalsvektor mit sechs Merkmalen (Features). Die Gesamtmenge aller durch Messungen erzeugten Merkmalsvektoren (also eine Merkmalsvektorliste mit Messwertlisten) entspricht einem 2D-Array (zweidimensionales Array mit sechs Spalten).

Nach abgeschlossener Sensormontage wird dann als zweiter Schritt ein 6-stufiger Engineering-Prozess durchlaufen, um ein Machine-Learning-Modell mit einem zur Aufgabe passenden Algorithmus zu erstellen und eine logische Ausgangsschnittstelle für den Softsensor zu definieren (siehe Tabelle 1). Im letzten Schritt dieses Prozesses werden Modell und Algorithmus in eine sogenannte Inferenzmaschine integriert, die in einer Endlosschleife (State Machine mit zwei Zuständen) aus den Echtzeit-Vibrationsdaten eine Ausgangsinformation zum aktuellen Maschinenzustand berechnet und an andere Funktionen weiterleitet. Bei ausreichender Modellqualität lassen sich damit beispielsweise Stillstand, Leerlauf, Normalbetrieb mit unterschiedlichen Belastungen, Überlastsituationen (Anomalien) usw. mit sehr hoher Genauigkeit klassifizieren. Innerhalb des gesamten Prozesses sind zwei Iterationsschleifen vorgesehen, um zum einen die Fehlerquote mit verschiedenen Machine-Learning-Algorithmen zu untersuchen und zum anderen die sogenannten Hyperparameter für den gewählten Algorithmus anzupassen. Des Weiteren kann natürlich auch der gesamte Engineering-Prozess beliebig oft durchlaufen werden, bis der virtuelle Sensor ein zufriedenstellendes Ergebnis liefert.



1: Datenerfassung, Labeling

Sensordaten erfassen, Merkmalsvektoren erzeugen, Vektoren per Labeling einer Zielgröße zuordnen (Mapping von x auf y , x sind alle Messdaten im Merkmalsvektor, y ist die gewünschte Zielgröße).

2: Datenanalyse, Algorithmenauswahl, Feature Engineering

Manuelle Datenanalyse, geeigneten Algorithmus auswählen. Falls erforderlich: Feature Engineering für Merkmalsvektoren.

3: Modell durch Training erzeugen

Erfasste Merkmalsvektoren in Trainings- und Testdaten aufteilen. Modell mit Hilfe der Trainingsdaten erzeugen.

4: Evaluieren, Validieren, Verifizieren

Zulässige Fehler festlegen. Dann Vorhersagequalität des Modells mit Hilfe der Testdaten prüfen und bewerten.

5: Optimierung

Hyperparameter optimieren, ggf. anderen Algorithmus auswählen.

6: Inferenzmaschineneinsatz

Modell in Inferenzmaschine übertragen, Ausgangsstufe zur Informationsweitergabe konfigurieren, beides zusammen einsetzen.

Abbildung 3: Der Engineering-Prozess für den Praxiseinsatz eines (virtuellen) Softsensors besteht aus sechs Teilaufgaben. Nach dem Erfassen von Trainings- und Testdaten und einer explorativen Datenanalyse wird ein Machine-Learning-Modell erzeugt und getestet. Innerhalb des Prozesses sind zwei Iterationsschleifen vorgesehen, um zum einen die Fehlerquote mit verschiedenen Machine-Learning-Algorithmen zu untersuchen und zum anderen die sogenannten Hyperparameter für den gewählten Algorithmus anzupassen.

Im finalen dritten Schritt auf dem Weg zum lernfähigen Sensor wird der Ausgang des virtuellen Sensors mit Hilfe der physikalischen und logischen Schnittstelle, beispielsweise Ethernet und OPC UA, mit der übergeordneten Anwendung (z. B. die KI-Anwendung zur Gesamteffizienzoptimierung) verbunden und die Gesamtlösung getestet.

Übersichtstabelle zum Softsensor-Engineering-Prozess

Seq.-Nr.	Funktion	Beschreibung
1.	Datenerfassung, Labeling	Sensordaten erfassen, Merkmalsvektoren erzeugen, Vektoren per Labeling einer bestimmten Zielgröße zuordnen (Mapping von x auf y, x sind hier alle Messdaten in einem Merkmalsvektor, y ist die jeweils gewünschte Zielgröße).
2.	Datenanalyse, Algorithmenauswahl, Feature Engineering	Manuelle Datenanalyse, passenden Algorithmus auswählen, falls erforderlich: ein Feature Engineering für die Merkmalsvektoren durchführen.
3.	Modell durch Training erzeugen	Die zuvor erfassten Merkmalsvektoren in Trainings- und Testdaten aufteilen. Ein Machine-Learning-Modell mit Hilfe der Trainingsdaten erzeugen.
4.	Evaluieren, Validieren, Verifizieren	Zuvor sollte der zulässige Fehler feststehen. Dann die Vorhersagequalität des Modells mit Hilfe der Testdaten prüfen und bewerten. Je nach Ergebnis wird die weitere Bearbeitung bei 2., 3. oder 6. fortgesetzt.
5.	Optimieren	Die Hyperparameter für den gewählten Algorithmus optimieren, ggf. einen anderen Algorithmus auswählen.
6.	Inferenzmaschineneinsatz	Das Machine-Learning-Modell in eine Inferenzmaschine übertragen, Ausgangsstufe zur Informationsweitergabe konfigurieren, beides zusammen einsetzen.

Tabelle 1: Bei der Inbetriebnahme des Softsensors wird ein 6-stufiger Engineering-Prozess iterativ durchlaufen, um u.a. aus zuvor erfassten Trainings- und Testdaten ein Modell für einen Supervised-Machine-Learning-Algorithmus zu erzeugen. Dieses Modell wird dann ausführlich getestet, um die erforderliche Qualität der Sensorausgangsinformationen zu gewährleisten.