Konzept zur automatischen Bauteilerkennung innerhalb der FE-Software-Umgebung mittels Künstlichen Neuronalen Netzen

Tobias C. Spruegel und Sandro Wartzack
Lehrstuhl für Konstruktionstechnik;
Friedrich-Alexander-Universität Erlangen-Nürnberg

Abstract

The Finite Element Method is a very powerful tool to enhance product quality and to reduce unnecessary and costly iterations during the product development process. However numerous errors can occur during the FE simulation. A promising way to realize precise and quick FE simulations is the comparison of the FE solution with analytical machine elements equations. This is only possible if the components in an assembly can be identified automatically. A promising way is to use the presented approach in this paper for part recognition with artificial neural networks and the use of detection surfaces.

Keywords: automatic part recognition, Artificial Neural Network, ANN, FEM, detection surface

1 Einleitung

Die Finite-Elemente-Methode trägt aktuell wesentlich zur Verkürzung der gesamten Produktentwicklungszeit bei und stellt dabei in Kombination mit CAD eines der leistungsfähigsten Verfahren dar, um die Ingenieurarbeit in der Produktentwicklung zu rationalisieren und qualitativ zu optimieren [1].

Der Produktentwickler sieht sich zusehens mit der Forderung nach verkürzten Entwicklungszeiten konfrontiert, die u. a. durch die Reduzierung unnötiger Iterationen im Produktentwicklungsprozess erreicht werden können. Aus dieser Forderung einerseits und den softwareseitigen Möglichkeiten andererseits erwachsen enorme Möglichkeiten zur Optimierung der virtuellen Produktentwicklung. Ein vielversprechender Weg zur Vermeidung von Iterationen besteht in der konstruktionsbegleitenden Berechnung mittels modernen FEA-Programmen. Aktuelle Forschungsvorhaben zielen daher auf die Plausibilitätsprüfung von FE-Ergebnissen durch den Abgleich mit Modellen der Technischen Mechanik ab (z. B. analytische Gleichungen für Maschinenelemente). Für automatisierte Nachrechnungen müssen Bauteile innerhalb der FE- oder CAD-Umgebung möglichst automatisch erkannt werden. Nur so können die passenden Berechnungen im Hintergrund erfolgen und für einen Abgleich mit der FE-Lösung herangezogen werden.

2 Ableitung des Handlungsbedarfs

In der VDI Richtlinie 2230 wird insbesondere bei kritischen Schraubenverbindungen auf die Notwendigkeit von experimentellen und/oder numerischen Untersuchungen zur Verifikation der analytischen Berechnungsergebnisse verwiesen [2]. Eine numerische Simulation sollte daher mit den Ergebnissen aus einer analytischen Berechnung verglichen werden können.

Trotz der Notwendigkeit der automatischen Bauteildetektion zur weiteren Nutzung in automatisierten Berechnungen (sowohl analytisch als auch numerisch) gibt es aktuell keine Möglichkeit zur sicheren automatisierten Erkennung von Bauteilen in CAD-Baugruppen. Gleiches gilt für die Detektion von Bauteilen innerhalb der FEA-Software-Umgebung. Mit dem hier dargestellten Konzept zur automatisierten Bauteilerkennung können unterschiedliche Bauteile innerhalb der FE-Software-Umgebung erkannt werden. Es findet hierbei ein Abgleich mit zuvor eingelesenen Bauteilen mittels Künstlichen Neuronalen Netzen statt. Zunächst wird auf die prinzipiellen Architektur und die Grundlagen von Künstlichen Neuronalen Netzen eingegangen und darauf aufbauend das Konzept zur automatisierten Bauteilerkennung vorgestellt.

3 Künstliche Neuronale Netze (KNN)

Künstliche Neuronale Netze haben zum Ziel die neuronalen Strukturen von Lebewesen nachzubilden. Anstelle von den natürlich vorkommenden Axonen und Dendriten wird die Verbindung zwischen den künstlichen Neuronen mit Gewichtungen modelliert [3]. Aufgrund des vielseitigen Aufbaus können KNN in diversen Aufgabenfeldern eingesetzt werden. Nach [4] sind dies u. a. An-

wendungsfelder der Klassifikation, Rauschminderung, Bildoptimierung, Zeitreihenvorhersage, Funktionsapproximierung und Regression. In diesem Beitrag werden KNN verwendet, da sich mit ihnen große Datensätze verarbeiten lassen und eine individuelle Anpassung der Netzeigenschaften auf die vorliegenden Datensätze möglich ist.

3.1 Architektur von KNN

Künstliche Neuronale Netze sind aus einer Eingabe- und Ausgabeschicht aus künstlichen Neuronen und mindestens einer verdeckten Neuronenschicht zwischen diesen aufgebaut. Die Anzahl der Neuronen in der Eingabeschicht ist identisch mit der Anzahl an Eingabeparametern. Dies gilt auch für die Neuronen in der Ausgabeschicht und die Ausgangsgrößen des KNN. Die Anzahl der Neuronen in der Zwischenschicht und die Anzahl an verdeckten Zwischenschichten sind beliebig anpassbar und bestimmt maßgeblich die Prognosequalität des Netzes. Dieser prinzipielle Aufbau kann Bild 1 entnommen werden. Die Verbindungen zwischen den Neuronen besitzen unterschiedliche Gewichtungen und müssen während dem Training des KNN berechnet werden.

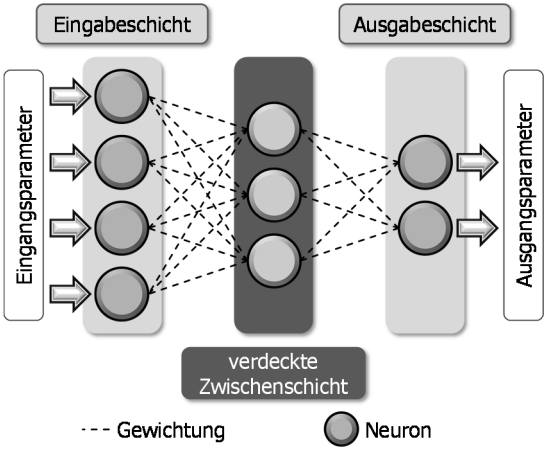

Bild 1: Struktur eines KNN mit einer verdeckten Neuronenschicht mit 3 Neuronen, 5 Eingangs- und 2 Ausgabegrößen

Die interne Berechnung der Ausgabegrößen des KNN ist abhängig von der verwendeten Transferfunktion in den jeweiligen Neuronen. Eine vielseitige und häufig genutzte Transferfunktion ist die Tangens Hyperbolicus Funktion. Die Berechnung innerhalb des KNN ist exemplarisch für ein Neuron in Bild 2 dargestellt.

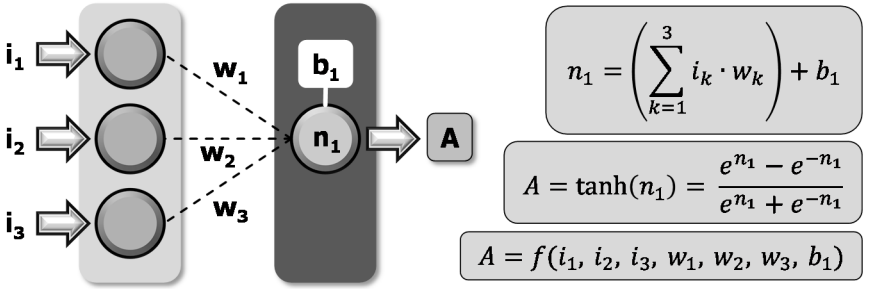

Bild 2: mathematischer Zusammenhang eines KNN mit *tanh*-Funktion

Im ersten Schritt der Berechnung des KNN werden die Eingangsgrößen des entsprechenden Neurons mit den Gewichtungen w multipliziert und dem statischen Biaswert des Neurons aufsummiert. Der Biaswert ermöglicht es den Wert der Zwischenvariable n1 statisch zu beeinflussen und es kann somit eine weitere Anpassung des Verhaltens des einzelnen Neurons erreicht werden. Im zweiten Schritt wird die so berechnete Summe n der Transferfunktion übergeben und der Ausgabewert des Neurons wird ermittelt. Dieser Vorgang wird für jedes der Neuronen separat berechnet. So können durch die Hintereinander- und Parallelschaltung von Neuronen mit unterschiedlichen Transferfunktionen sehr komplizierte und nichtlineare Zusammenhänge abgebildet werden.

3.2 Training und Anwendung von KNN

Vor der Anwendung eines KNN müssen zunächst die Gewichtungen und Bias-Werte errechnet werden. Dieser Vorgang wird als Training bezeichnet und stellt deutlich höhere Anforderungen an die Hardware als die spätere Anwendung des KNN. Typische Trainingsalgorithmen für KNN sind der Levenberg-Marquardt und der BFGS Quasi-Newton Algorithmus. Diese Algorithmen sind sehr effizient für das Training von Netzen mit wenigen Gewichtungen, sind jedoch schlechter geeignet für das Training von Netzen mit mehreren Tausend Gewichtungen [5]. Für das Training werden Eingangsgrößen und die dazu gehörigen Ausgangsgrößen an den Trainingsalgorithmus übergeben. Nach Abschluss des Trainings kann auf ein KNN zurückgegriffen werden, dass aus übergebenen Eingangswerten die zugehörigen Ausgangswerte mit einer bestimmten Prognosegenauigkeit berechnen kann. Dieser Vorgang wird als Anwenden des KNN bezeichnet. Der Vorgang des Trainings und des Anwendens eines KNN ist in Bild 3 zu erkennen.

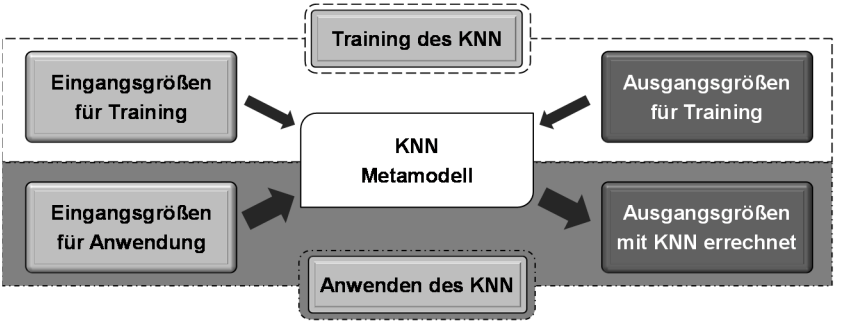

Bild 3: Training und Anwendung eines KNN

4 Vorstellung des Konzepts zur Bauteilerkennung mittels KNN

Zunächst wurde auf den prinzipiellen Aufbau und die Grundlagen von Künstlichen Neuronalen Netzen eingegangen. Im Folgenden wird das Konzept zur automatisierten Bauteilerkennung vorgestellt.

4.1 Orientierung von Bauteilen

In den zu untersuchenden virtuellen Baugruppen innerhalb der FE-Software können Bauteile, wie in realen Produkten auch, beliebig orientiert und verschoben im Raum vorliegen. Die Betrachtung der möglichen Verdrehungen um die drei Raumkoordinatenachsen erfolgt im Zuge dieses Konzeptes in vordefinierten Winkel-Schritten. Bei der Verdrehung einer Sechskantschraube nach ISO 4016 ergeben sich bei 20°-Winkel-Schritten um eine Raumachse zehn mögliche Orientierungen (dies verdeutlicht Bild 4 anhand der FE-Netz-Knoten).

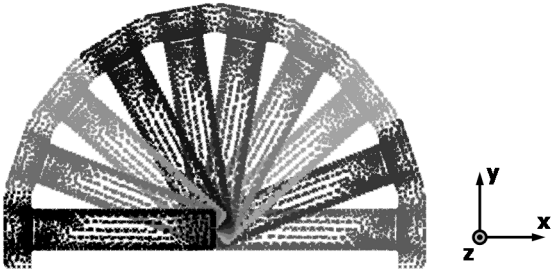

Bild 4: Mögliche Orientierungen einer Sechskantschraube bei Rotation um die z-Raumachse in 20°-Winkel-Schritten

Die Anzahl der zu untersuchenden Bauteil-Orientierungen schwankt von Bauteil zu Bauteil. Bei einer Sechskantschraube nach ISO 4016 müssen bei einer Schrittweite von 10° folgende Orientierungen untersucht werden:

- Rotation um x-Achse: 6 mögliche Orientierungen
- Rotation um y-Achse: 19 mögliche Orientierungen
- Rotation um z-Achse: 19 mögliche Orientierungen
- Zu untersuchende Orientierungen: 6 x 19 x 19 = 2.166

Um die Bauteile mit dem vorgestellten Konzept erkennen zu können, muss jede einzelne dieser Orientierungen betrachtet werden. Dies gilt nicht nur für die Bauteilart (z. B. ISO 4016, ISO 4017), sondern auch für jede Größe der Bauteile nach der einzelnen Norm.

4.2 Vernetzung der Bauteile

Die Vernetzung der Bauteile erfolgt mit einer vorgegebenen Elementgröße von 3 mm und Tetraedern als Elementart. Die Wahl einer Elementgröße von 3 mm liegt darin begründet, dass möglichst wenige Knoten pro Bauteil verwendet werden sollten. Gleichzeitig aber eine ausreichende Anzahl für die weiteren Berechnungsschritte vorhanden sein muss. Die Vernetzung wird mit ANSYS Workbench durchgeführt und erfolgt mittels striktem Netzverhalten. Hierdurch wird die Anzahl der Teilungen auf einer Kante fix vorgegeben und der Vernetzungsalgorithmus kann die Anzahl der Elemente pro Kante nicht verändern. Dies führt einerseits zu einer höheren Fehleranfälligkeit bei der Vernetzung [6], garantiert aber andererseits eine identische Vernetzung bei einer erneuten Netzgenerierung des gleichen Bauteils.

4.3 Reduktion der Knoten mittels Detektorfläche

Unterschiedlich große Bauteile weisen bei gleicher Elementgröße eine unterschiedliche Anzahl an Knoten des FE-Netzes auf. Künstliche Neuronale Netze besitzen eine fix vorgegebene Anzahl an Eingangsneuronen und müssen daher stets eine gleiche Anzahl an Eingangsparametern übergeben bekommen. Aus diesem Grund können die Netzknoten des FE-Netzes nicht direkt an das KNN übertragen werden. Um die unterschiedlich großen Anzahlen an Netzknoten auf eine stets konstante Anzahl an Eingangsgrößen zu reduzieren, werden Detektormatrizen verwendet. Hierbei werden die Bauteile durch eine Detektorfläche betrachtet und die projizierten Knoten innerhalb jedes Detektorpixels gezählt. Vor der eigentlichen Betrachtung der Bauteile werden die Punkte der FE-Netzknoten so verschoben, dass diese innerhalb der Detektorfläche liegen. Hierfür werden die Minimalwerte in jeder der drei Koordina-

tenrichtungen berechnet und anschließend von jedem Knoten des Bauteils abgezogen. Die Bauteile werden dadurch in den linken unteren Rand der Detektorfläche verschoben (siehe Bild 5).

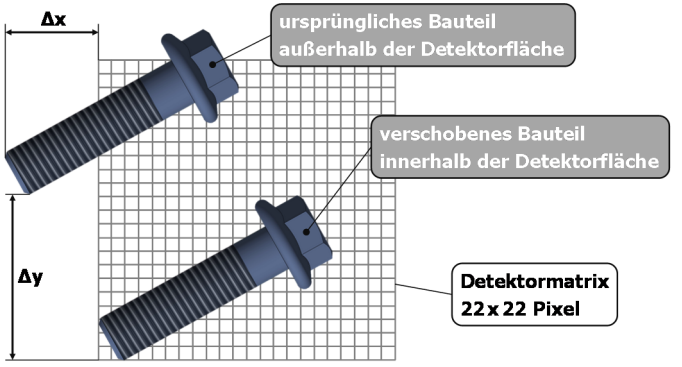

Bild 5: Verschiebung der zu erkennenden Bauteile in die Detektorfläche

Um den rechnerischen Aufwand bei der Erstellung der KNN zu verringern wird eine relativ kleine Detektorfläche mit $22 \times 22 = 484$ Pixeln verwendet. Prinzipiell kann ein Bauteil durch mehrere, zueinander verdrehte Detektormatrizen betrachtet und die Ergebnisse aller dieser Matrizen an das KNN übergeben werden. Dies hat den Vorteil, dass unterschiedliche Orientierungen der Bauteile besser erkannt werden können.

4.4 Training des KNN

Vor dem Training des KNN müssen die Detektormatrizen für jedes Bauteil und jede der zuvor erwähnten Orientierungen in einen Eingabevektor umgewandelt werden. Diese Umwandlung geschieht durch eine Transformation der Detektormatrix der Dimension 22×22 in einen Spaltenvektor der Größe 484×1. Für die Berechnung der Gewichtungen des KNN muss neben dem Eingangsvektor ebenfalls ein Ausgabevektor übergeben werden. Hierbei wird jedem einzulesenden Bauteil (z. B. Sechskantschraube ISO 4016 – M8 x 40) eine Ausgabewert zugewiesen (z. B. der Zahlenwert 10). Bei der Auswahl der Zahlenwerte ist auf einen ausreichenden Abstand der Zahlenwerte zwischen den betrachteten Bauteilen zu achten. So wird einer Sechskantschraube ISO 4016 – M8 x 50 z. B. der Zahlenwert 20 zugewiesen. Der zugewiesene Ausgabewert wird dabei jeder Orientierung eines spezifischen Bauteils zugewiesen. Somit ergibt sich für den Eingangsdatensatz der erwähnten Sechskantschraube eine Matrix der Größe 484×2.166 mit einen zugehörigen Ausgabevektor

der Größe 1 x 2.166. Dieser Ausgabevektor enthält immer den gleichen Wert da eine identische Schraube, aber in unterschiedlichen Orientierungen eine identische Schraube betrachtet wird.

Das Training des KNN stellt - in Abhängigkeit der verwendeten Trainingsalgorithmen - hohe Anforderungen an die verwendete Hardware. Dies liegt in der hohen Anzahl an Eingangsparametern (484), der Anzahl an verdeckten Neuronen in der Zwischenschicht (z. B. 100) und des Umfangs an Trainingsdatensätzen begründet. Der Umfang ergibt sich durch die zu untersuchenden Orientierungen (z. B. 2.166) pro Bauteil und die Gesamtanzahl an zu untersuchenden Normteilen in verschiedenen Größen.

Die Hardwareanforderungen bei der Generierung der KNN sind durch die Auswahl von besser geeigneten Transferfunktionen reduzierbar. Dies kann beispielsweise durch den Einsatz von Fast Elliot Sigmoid Transferfunktionen erfolgen, hierbei werden Exponentialfunktionen wie die Tangens Hyperbolicus Funktion vermieden [4]. Ebenso kann anstelle einer Berechnung mit allen Datensätzen ein schrittweises Trainieren des KNN mit mehreren kleineren Trainingsdatensätzen erfolgen.

4.5 Anwendung des KNN

Nach dem Training des KNN und der Berechnung der Gewichtungen zwischen den Neuronen kann das somit erzeugte Netz zur Bauteilerkennung herangezogen werden. Hierfür sind die Bauteile zunächst in einer zu untersuchenden Baugruppe innerhalb der FE-Umgebung mit den zuvor erwähnten Einstellungen (Elementart und -größe) zu vernetzen. Anschließend werden die X-, Y- und Z-Koordinaten der Elementknoten für jedes Bauteil ausgelesen. Nach der Betrachtung dieser Knoten durch die Detektorfläche und der Umwandlung der so erzeugten Detektormatrix in einen Eingabevektor kann dieser an das KNN übergeben werden. Das KNN berechnet aus diesem Eingabevektor den zugehörigen Ausgabewert. Je höher die Prognosequalität des KNN ist desto näher liegt der errechnete Wert des Netzes am tatsächlichen Wert. Durch die Berechnung des Ausgabewertes mit Exponentialfunktionen und unterschiedlichen Gewichtungen berechnet das KNN keine exakten Ganzzahlen und die Ausgabe muss entsprechend gerundet werden. Nach der Rückführung des Ausgabewertes in die exakte Bauteilbezeichnung können automatisierte FE-Simulationen oder analytische Vergleichsrechnungen durchgeführt werden. Die nachgelagerten Schritte sollten dabei möglichst automatisiert durchführbar sein um mit dem vorgestellten Konzept eine deutliche Effizienzsteigerung in der virtuellen Produktentwicklung zu erreichen.

5 Zusammenfassung und Ausblick

Mit FE-Simulationen können Bauteile bereits in der virtuellen Phase der Produktentwicklung hinsichtlich ihrer beanspruchungsgerechten Gestaltung überprüft werden. Hierfür ist es notwendig, schnelle und präzise FE-Simulationen durchzuführen. Zur Validierung von FE-Ergebnissen können auch analytische Maschinenelemente-Gleichungen herangezogen werden. Allerdings muss hierfür bekannt sein, welche Normteile in einer zu untersuchenden Baugruppe vorliegen. Bisher standen dazu keine anwendbaren Ansätze oder Methoden zur Verfügung. Mit dem vorgestellten Konzept zur automatisierten Bauteilerkennung mittels Künstlichen Neuronalen Netzen können unbekannte Bauteile in einer in der FE-Umgebung vorliegenden Baugruppe mit zuvor eingelesenen Bauteilen verglichen und erkannt werden. Nach dem Training des KNN ist die Erkennung von Bauteilen mittels der Detektorflächen und der Anwendung des KNN zur Klassifizierung schnell und ohne große Hardwareanforderungen möglich.

Im Zuge der weiteren Forschung ist das Konzept auf die Möglichkeit zur Erkennung von ähnlichen Bauteilen und Bauteilen mit geringen geometrischen Unterschieden hin zu überprüfen. Durch die bisherige Verwendung von Standardeinstellungen bei der Erzeugung des KNN können diverse Parameter variiert werden, um die Prognosequalität des Netzes zu erhöhen. Diese Parameter sind beispielsweise die Anzahl der Zwischenschichten des KNN, die darauf befindliche Anzahl von künstlichen Neuronen, die Transferfunktion jedes einzelnen Neurons oder der verwendete Trainingsalgorithmus. Darüber hinaus können die untersuchte Elementgröße der FE-Netze, die Anzahl der Pixel auf der Detektorfläche und die Winkelschritte der betrachteten Orientierungen adaptiert werden. Durch die Möglichkeit einer Anpassung dieser Parameter ist sichergestellt, dass mit dem KNN eine hohe Prognosequalität zur Bauteilerkennung erreicht werden kann.

Literatur

[1] Klein, B.: "FEM, Grundlagen und Anwendung der Finite-Element-Methode im Maschinen- und Fahrzeugbau", Vieweg + Teubner Verlag, Wiesbaden, 2010.

[2] VDI 2230: "Systematische Berechnung hochbeanspruchter Schraubenverbindungen Zylindrischer Einschraubverbindungen", 2003.

[3] Ertel, W.: "Grundkurs künstliche Intelligenz, Eine praxisorientierte Einführung", Vieweg + Teubner Verlag, Wiesbaden, 2009.

[4] Masters, T.: "Practical neural network recipes in C++", Academic Press, Boston, 1993.

[5] Beale, M. H., et al.: "Matlab Neural Network Toolbox, User's Guide R2013a", MathWorks, 2013.

[6] ANSYS® 15.0: "Meshing User's Guide", ANSYS, Inc.

Das hier vorgestellte Vorgehen ist Bestandteil des Teilprojekts 7 des BFS Forschungsverbunds FORPRO² (AZ 1071-13). Die Autoren danken der Bayerischen Forschungsstiftung für die finanzielle Unterstützung.