

(12)

Patentschrift

(21) Anmeldenummer: A 50360/2022
(22) Anmeldetag: 23.05.2022
(45) Veröffentlicht am: 15.11.2024

(51) Int. Cl.: **G05B 13/02** (2006.01)
G05B 19/404 (2006.01)

(56) Entgegenhaltungen:
DE 102017011544 A1
DE 102015115838 B4
KRUGER, G.H. et al. "Intelligent machine agent architecture for adaptive control optimization of manufacturing process" Advanced Engineer Informatics 25 (2011) 03. September 2011, Seiten 783 - 796; ISSN: 1474-0346; doi:10.1016/j.aei.2011.08.003

(73) Patentinhaber:
Fill Gesellschaft m.b.H.
4942 Gurten (AT)

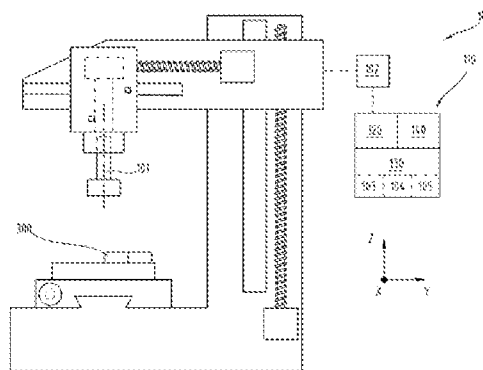
(72) Erfinder:
Stanzl Verena Dipl.-Ing.
3500 Krems (AT)
Gillhofer Michael MSc
4040 Linz (AT)
Murauer Stefan
4910 Ried im Innkreis (AT)

(74) Vertreter:
Anwälte Burger und Partner Rechtsanwalt
GmbH
4580 Windischgarsten (AT)

(54) Optimieren einer numerischen Steuerung einer Werkzeugmaschine

(57) Die Erfindung betrifft ein Verfahren zum Optimieren einer numerischen Steuerung einer Werkzeugmaschine mit einem Werkzeug zum Bearbeiten eines Werkstückes. Es werden numerische Werkzeuginformationen eine Bearbeitung betreffend erhalten. Dann werden durch ein trainiertes neuronales Netzwerk basierend auf den erhaltenen Werkzeuginformationen Zeitinformationen ermittelt. Aus den ermittelten Zeitpunkten und den Werkzeuginformationen wird ein Satz von Bahninformationen für die Bearbeitung generiert, und ein Minimalabstand des Werkzeugs zum Werkstück vor der Phase Annäherung aus mehreren Sätzen von Bahninformationen mehrerer vorheriger Bearbeitungen für die nächste Bearbeitung bestimmt. Es wird ein Verfahren zum Optimieren einer numerischen Steuerung einer Werkzeugmaschine sowie eine Vorrichtung zur Ausführung des Verfahrens, eine Werkzeugmaschine zur Bearbeitung von Werkstücken durch Ausführung des Verfahrens, ein Computerprogrammprodukt, ein computerlesbares Medium, ein Verfahren zum Bearbeiten von Werkstücken bereitzustellen sowie ein Verfahren zum Trainieren eines neuronalen Netzwerks bereitgestellt.

Fig.1



Beschreibung

[0001] Die Erfindung betrifft das Optimieren einer numerischen Steuerung einer Werkzeugmaschine mit einem Werkzeug zum Bearbeiten eines Werkstückes.

[0002] Aus der EP 3 720 651 A1 sind Werkzeugmaschinen bekannt, die beispielsweise eine spanende Verarbeitung durch ein Werkzeug auf einem Werkstück durchführen können. Darüber hinaus ist seit vielen Jahrzehnten die Verwendung einer sogenannten numerischen Steuerung (NC, engl. numerical control) bekannt. Mit einer numerischen Steuerung werden Maschinen gesteuert, indem die Maschine Steuerbefehle liest, die als Code auf einem Datenträger gespeichert sind. Zu Beginn wurden als Datenträger Lochstreifen oder Lochkarten verwendet. Die Steuerbefehle werden in Arbeits- bzw. Bewegungsabläufe umgesetzt. Durch das Zurverfügungstellen von verschiedenen Codelisten auf verschiedenen Datenträgern oder Dateien, kann eine numerisch gesteuerte Maschine an verschiedene Bearbeitungsverläufe, z.B. ein anderes Werkstück, angepasst werden. Etwa seit 1980 werden zur Steuerung Computer eingesetzt, was CNC (engl. computerized numerical control) genannt wird. Die Abläufe werden aktuell in normierten Datenstrukturen hinterlegt, beispielsweise gemäß DIN 66025/ISO 6983 im sogenannten G-Code. Dadurch können Werkzeugmaschinen durch den Einsatz von Steuerungstechnik Werkstücke mit hoher Präzision auch für komplexe Formen automatisch herstellen.

[0003] DE 102017011544 A1 beschreibt eine Steuerung und eine maschinelle Lernvorrichtung, die ein neuronales Netz mit einem Lernmodus und einem Vorhersagemodus umfasst. Das System verstärkt die Genauigkeit der Steuerung einer Werkzeugmaschine.

[0004] DE 102015115838 B4 beschreibt die Verwendung eines neuronalen Netzes in einer Bearbeitungsmaschine wobei die Bearbeitungsmaschine mindestens einen entlang zumindest einer Linearachse verschieb- und positionierbaren Werkzeugaufnahmekopf, mindestens einen um zumindest eine Drehachse rotier- und positionierbaren und/oder entlang zumindest einer Linearachse verschieb- und positionierbaren Werkstückaufspanntisch sowie für jede Linear- und/oder Drehachse eine Lageerkennungs- und Positionssteuerung aufweist.

[0005] KRUGER, G.H. et al. beschreiben in "Intelligent machine agent architecture for adaptive control optimization of manufacturing process" eine Architektur zur Optimierung der adaptiven Regelung bei einem Bearbeitungsprozess einer Werkzeugmaschine (z.B. Bohrprozess). Die Steuerungsarchitektur inkorporiert dabei intelligente Algorithmen bezüglich Lernens, Flexibilität und Entscheidungsfindung in heterogenen Prozessen mittels der Verwendung neuronaler Netze.

[0006] Im Standard-Prozess der Erstellung eines (C)NC Programmes zur Bauteilfertigung werden verschiedene Abfolgen von sogenannten G-Codes implementiert. Dabei wird zwischen Eilvorschüben zur Positionierung von Achsen und Fräsvorschüben zur aktiven Bearbeitung unterschieden. Diese unterscheiden sich vor allem in der Geschwindigkeit. Um das zu verwendende Werkzeug sicher betreiben zu können, wird eine Anfahrposition gewählt, die mit höherer Geschwindigkeit angefahren wird, bevor auf die aktive Bearbeitungsgeschwindigkeit umgeschaltet wird. Die Anfahrposition zeichnet sich durch einen Sicherheitsabstand vom Werkstück aus. Die Wahl der Größe dieses Sicherheitsabstandes hängt einerseits von geometrischen Bedingungen (z.B. Schwankungen in der Bauteilgeometrie, Werkzeuggeometrie, Aufspanntoleranzen des Werkstückes) sowie auch teilweise von der Erfahrung der Person, die den G-Code programmiert und deren fehlendem Wissen über die Maschinenfunktionalität und -geometrie ab.

[0007] Nachteilig im Stand der Technik ist also, dass aktuelle Verfahren Effizienzmängel aufweisen. Konkret geht durch die Wahl eines zu großen Sicherheitsabstandes Potential bei der Taktzeit und damit dem Durchsatz verloren, während ein zu kleiner Abstand zu einer Beschädigung des Werkzeuges führen kann.

[0008] Aufgabe der vorliegenden Erfindung ist es somit, die Nachteile des Standes der Technik zu überwinden. Es wird ein Verfahren zum Optimieren einer numerischen Steuerung einer Werkzeugmaschine sowie eine Vorrichtung zur Ausführung des Verfahrens, eine Werkzeugmaschine zur Bearbeitung von Werkstücken durch Ausführung des Verfahrens, ein Computerprogramm-

produkt, ein computerlesbares Medium, ein Verfahren zum Bearbeiten von Werkstücken bereit-zustellen sowie ein Verfahren zum Trainieren eines neuronalen Netzwerks bereitgestellt.

[0009] Diese Aufgabe wird durch die Vorrichtungen bzw. Verfahren gemäß den unabhängigen Ansprüchen gelöst. In den abhängigen Ansprüchen finden sich vorteilhafte Weiterbildungen.

[0010] Im Zusammenhang der hier beschriebenen Werkzeugmaschinen, kann eine Bearbeitung beispielsweise folgende Phasen umfassen: Anfahrt, Annäherung, Werkzeugeintritt, Arbeitsphase, Werkzeugaustritt und Abfahrt. Die Einteilung kann jedoch auch anders vorgenommen werden. Die Begriffe dienen lediglich beispielhaft der Bezugnahme auf jeweilige Teile einer Bearbeitung.

[0011] Vorteilhaft ist dabei, dass der Eintrittszeitpunkt (und Austrittszeitpunkt) des Werkzeuges im Material ermittelt wird und so der tatsächliche Leerlauf ermittelt werden kann. Durch Aggregieren dieses Wertes über mehrere Bearbeitungen (d.h. mehrere Werkstücke) hinweg wird die Verteilung berechnet, die sich durch die natürlichen Geometrieschwankungen ergeben. Durch Ermittlung des Minimalwertes wird das Optimum ermittelt zur Reduktion der Taktzeit bei sicherem Betrieb des Werkzeuges.

[0012] Das Verfahren kann bereits vor der Serienproduktion sowie im laufenden Prozess zur Optimierung eingesetzt werden. Der wirtschaftliche Nutzen ergibt sich daraus, dass die Taktzeit der Bauteilfertigung reduziert und die Durchsatzquote erhöht wird.

[0013] Das Projekt, das zu dieser Anmeldung geführt hat, wurde von der Europäischen Union im Rahmen des Horizon 2020 Forschungs- und Innovationsprogramm durch die Bewilligungsvereinbarung Nr. 857191 gefördert.

[0014] Gemäß einer ersten Ausführungsform umfasst ein erfindungsgemäßes Verfahren zum Optimieren einer numerischen Steuerung einer Werkzeugmaschine mit einem Werkzeug zum Bearbeiten eines Werkstückes, ein Erhalten von numerischen Werkzeuginformationen eine Bearbeitung betreffend, wobei die Werkzeuginformationen für jeden Zeitpunkt der Bearbeitung ein Spindeldrehmoment des Werkzeugs und Achspositionen des Werkzeugs umfassen. Das Verfahren umfasst weiterhin ein Ermitteln, durch ein trainiertes neuronales Netzwerk, von Zeitinformationen basierend auf den erhaltenen Werkzeuginformationen. Dabei umfassen, die Zeitinformationen einen Zeitpunkt des Werkzeugeintrittes in das Werkstück und einen Zeitpunkt des Werkzeugaustrittes aus dem Werkstück. Anschließend wird ein Satz von Bahninformationen für die Bearbeitung aus den ermittelten Zeitpunkten und den Werkzeuginformationen generiert. Ein solcher Satz von Bahninformationen umfasst einen Fahrweg des Werkzeugs, der die Phasen Annäherung, Werkzeugeintritt, Arbeitsphase, und Werkzeugaustritt umfasst, und einen Abstand des Werkzeugs zum Werkstück vor der Phase Annäherung. Aus mehreren Sätzen von Bahninformationen mehrerer vorheriger Bearbeitungen wird dann für die nächste Bearbeitung ein Minimalabstandes des Werkzeugs zum Werkstück vor der Phase Annäherung bestimmt.

[0015] Durch das erfindungsgemäße Verfahren kann ein optimaler Minimalabstand ermittelt werden, wodurch die Taktzeit und damit der Durchsatz verbessert wird. Darüber hinaus wird dennoch die Beschädigung des Werkzeuges vermieden.

[0016] Das Verfahren kann weiter verbessert werden, wenn die Werkzeuginformationen weiterhin die Spindeldrehzahl des Werkzeugs umfasst.

[0017] Durch die Berücksichtigung der Spindeldrehzahl kann das Ermitteln der Zeitinformationen durch das neuronale Netzwerk verbessert werden.

[0018] Die numerischen Werkzeuginformationen können hierbei von der numerischen Steuerung empfangen werden. Alternativ oder zusätzlich können numerische Werkzeuginformationen aus einem Speicher, einer Datenbank oder einem Datenträger ausgelesen und berücksichtigt werden.

[0019] Es hat sich gezeigt, dass die Verwendung von ggf. zusätzlichen, sogenannten historischen, Daten früherer Bearbeitungen die Genauigkeit des neuronalen Netzwerks in der Inferenz verbessert.

[0020] Die Ermittlung der Zeitinformationen durch das neuronale Netzwerk kann zusätzlich optimiert werden, wenn vor der Ermittlung die Werkzeuginformationen durch Entfernen von Informationen, die für die Bestimmung des Minimalabstandes irrelevant sind, bereinigt werden.

[0021] Es hat sich gezeigt, dass in einigen Datensätzen Daten enthalten sind, die für die Bestimmung des Minimalabstandes irrelevant sind. Durch die Bereinigung der Daten vor der Inferenz, kann die Ermittlung der Zeitinformationen beschleunigt werden.

[0022] Weiterhin kann es zweckmäßig sein, nicht nur den Minimalabstand vor der Annäherung zu bestimmen, sondern auch vor oder während der Anfahrt, während oder nach der Annäherung, und/oder während oder nach der Abfahrt. Auch diese können aus mehreren Sätzen von Bahninformationen mehrerer vorheriger Bearbeitungen für die nächste Bearbeitung bestimmt werden.

[0023] Hierbei kann durch die Sicherstellung eines Minimalabstandes in den entsprechenden Phasen ebenfalls eine Beschädigung des Werkzeugs und des Werkstücks durch eine unerwünschte Berührung vermieden werden.

[0024] Es kann sich als vorteilhaft erweisen, die aktuelle Bearbeitung zum Nachtrainieren des trainierten neuronalen Netzwerks zu verwenden. Hierbei können die ermittelten, generierten und bestimmten Informationen, oder einer Auswahl davon zum Nachtrainieren eingesetzt werden.

[0025] Das Nachtrainieren durch die aktuelle Bearbeitung hat einen geringen Einfluss auf die Genauigkeit des neuronalen Netzwerks. Durch das fortgesetzte Nachtrainieren wird das neuronale Netzwerk jedoch über die Zeit stetig besser. Die aktuelle Bearbeitung kann auch zuerst gespeichert werden, und nach dem Ansammeln einer gewissen Anzahl von Daten von Bearbeitungen, z.B. am Ende eines Tages oder einer Woche, o.Ä., kann das neuronale Netzwerk mit den gespeicherten Daten nachtrainiert werden. Dies kann beispielsweise über Nacht oder am Wochenende unüberwacht durchgeführt werden. Ein überwachtes Nachtrainieren ist aber ebenfalls möglich.

[0026] Für das Bestimmen von Minimalabständen können weiterhin eines oder mehrere aus Geometrien von Werkstücken vor der Bearbeitung, Geometrien von Werkstücken nach der Bearbeitung, Geometrie des Werkzeugs, Aufspanntoleranzen der Werkstücke und Werkzeugverschleiß berücksichtigt werden.

[0027] Es hat sich gezeigt, dass in einigen Fällen die Berücksichtigung dieser Daten ebenfalls zu einer Verbesserung führt, und dadurch ungewollte Beschädigungen vermieden werden können, indem ungewollte Kollisionen vermieden werden, und die Effizienz gesteigert werden kann, da die Bewegungsbahnen, oder Verfahrbahnen, optimaler geplant werden können.

[0028] Gemäß einer weiteren Ausführungsform der Erfindung wird eine Vorrichtung, umfassend mindestens eine Recheneinheit, mindestens eine Speichereinrichtung und ein trainiertes neuronales Netz bereitgestellt, wobei die Vorrichtung dazu eingerichtet ist, die Schritte eines erfindungsgemäßen Verfahrens auszuführen. Diese Vorrichtung kann Teil einer Werkzeugmaschine sein, oder auch eigenständig.

[0029] Gemäß einer weiteren Ausführungsform der Erfindung wird eine Werkzeugmaschine mit Mitteln, die geeignet sind, die Schritte eines erfindungsgemäßen Verfahrens auszuführen und Werkstücke zu bearbeiten, wobei die bestimmten Minimalabstände bei der Bearbeitung berücksichtigt werden.

[0030] Gemäß einer weiteren Ausführungsform der Erfindung wird ein Computerprogrammprodukt, umfassend Befehle, die bei der Ausführung des Programms durch einen Computer diesen veranlassen, ein erfindungsgemäßes Verfahren auszuführen. Das Computerprogrammprodukt kann auf einem computerlesbaren Medium gespeichert sein.

[0031] Gemäß einer weiteren Ausführungsform der Erfindung wird ein Verfahren zum Bearbeiten eines Werkstücks durch die erfindungsgemäße Werkzeugmaschine bereitgestellt.

[0032] Gemäß einer weiteren Ausführungsform der Erfindung wird ein Verfahren zum Trainieren eines neuronalen Netzwerkes bereitgestellt. Dabei werden eine Vielzahl von Trainingsdatensät-

zen erhalten, wobei jeder Trainingsdatensatz aus numerischen Werkzeuginformationen besteht, die eine Bearbeitung betreffen, und die in Form einer Zeitreihe für jeden Zeitpunkt der Bearbeitung ein Spindeldrehmoment eines Werkzeugs einer Werkzeugmaschine umfassen. Die Trainingsdatensätze werden dann segmentiert, wodurch die entsprechende Zeitreihe jedes Trainingsdatensatzes durch Markieren der Zeitpunkte für einen Werkzeugeintritt und für einen Werkzeugaustritt in mindestens drei Bereiche geteilt wird. Nach einem Padding jedes Trainingsdatensatzes, wodurch am Anfang und am Ende jeder Zeitreihe Fülldaten angefügt werden, wird das neuronale Netzwerk mit den Trainingsdatensätzen trainiert.

[0033] Die Ausführungsformen zeigen mögliche Ausführungsvarianten, wobei die Erfindung nicht auf die speziell dargestellten Ausführungsvarianten derselben eingeschränkt ist, sondern vielmehr auch Kombinationen der einzelnen Ausführungsvarianten untereinander möglich sind.

[0034] Zum besseren Verständnis der Erfindung wird diese anhand der nachfolgenden Figuren näher erläutert.

[0035] Es zeigen jeweils in stark vereinfachter, schematischer Darstellung:

[0036] Fig. 1 zeigt eine Werkzeugmaschine mit einem Werkzeug zum Bearbeiten eines Werkstückes gemäß einer Ausführungsform;

[0037] Fig. 2 zeigt schematische Flussdiagramme gemäß Ausführungsformen;

[0038] Fig. 3 zeigt eine beispielhafte Struktur des neuronalen Netzes gemäß einer Ausführungsform, und

[0039] Fig. 4 zeigt schematisch Ergebnisse des ausführungsformgemäßen neuronalen Netzes.

[0040] Einführend sei festgehalten, dass in den unterschiedlich beschriebenen Ausführungsformen gleiche Teile mit gleichen Bezugszeichen bzw. gleichen Bauteilbezeichnungen versehen werden, wobei die in der gesamten Beschreibung enthaltenen Offenbarungen sinngemäß auf gleiche Teile mit gleichen Bezugszeichen bzw. gleichen Bauteilbezeichnungen übertragen werden können. Auch sind die in der Beschreibung gewählten Lageangaben, wie z.B. oben, unten, seitlich usw. auf die unmittelbar beschriebene sowie dargestellte Figur bezogen und sind diese Lageangaben bei einer Lageänderung sinngemäß auf die neue Lage zu übertragen.

[0041] Fig. 1 zeigt eine Werkzeugmaschine 100 mit einem Werkzeug 101. Dieses Werkzeug kann ein Fräskopf, ein Bohrer, eine Bürste oder ein anderes spanendes Werkzeug, das materialabtragend in ein Werkstück 300 eingreift. Das Werkzeug wird von einer numerischen Steuerung 102 gesteuert und von einer Spindel rotierend angetrieben. Die Werkzeugmaschine kann eine Rechenvorrichtung 110 umfassen. Die Rechenvorrichtung kann alternativ auch eine eigenständige Vorrichtung sein.

[0042] Zuerst wird die grundlegende Arbeitsweise der vorgestellten Werkzeugmaschine 100 erläutert. Eine Bearbeitung eines Werkstückes 300 durch ein Werkzeug 101 hat mehrere Abschnitte oder (Bearbeitungs-)Phasen.

[0043] Zuerst wird das Werkzeug 101 in die Nähe des Werkstückes 300 verfahren. Dies ist die Anfahrt oder der Eilgang. Die Anfahrt kann mit einer hohen Geschwindigkeit (z.B. 500 mm/s) durchgeführt werden. Bei der Anfahrt kann das Werkzeug 101 noch in einem Ruhezustand betrieben werden. Ein Bohrer, eine Säge oder eine Fräse würde sich dann noch nicht drehen, könnte also noch kein Material bearbeiten bzw. abtragen. Das Werkzeug 101 kann aber auch bereits im Betriebszustand der Bearbeitung betrieben werden. Die Anfahrt endet, wenn ein vorher festgelegter Minimalabstand zu der Position, an der das Werkstück 300 erwartet wird, erreicht ist. Der tatsächliche Abstand zu diesem Zeitpunkt kann sich für jedes Werkstück 300 unterscheiden, da die Werkstücke 300 möglicher Weise nicht exakt einander entsprechen und es möglicher Weise Produktions- oder Aufspanntoleranzen gibt. Ein Werkstück 300 könnte auch verunreinigt sein, oder es könnte nicht exakt positioniert sein. Bei der Anfahrt werden die vorhandenen Geometrien berücksichtigt, d.h. das Werkzeug 101 wird derart zum Werkstück 300 bewegt, dass die gewünschte Position zum Werkstück 300 erreicht wird. Die gewünschte Position kann durch eine,

zwei oder drei Achsen im Raum festgelegt werden. Soll durch eine relativ große Säge ein Werkstück komplett durchgeschnitten werden, reicht eventuell eine Achse. Für eine Bohrung an einer Stelle eines Werkstückes reichen eventuell zwei Achsen, und für eine Fräsung in einem bereits ausgeformten Körper sind unter Umständen 3 Achsen notwendig. Die Anfahrt berücksichtigt weiterhin, dass eine Kollision während der Anfahrt nicht möglich ist, und der vorher festgelegte Mindestabstand eingehalten wird.

[0044] Der Begriff Abstand bzw. Minimalabstand bezieht sich generell immer auch den geringsten Abstand, den das Werkzeug 101 an einer beliebigen Stelle zum Werkstück 300 aufweist.

[0045] Ist der festgelegte Minimalabstand erreicht, und das Werkzeug 101 noch nicht in Betrieb, so kann es nun angeschaltet werden. Das Werkzeug 101 wird dann langsam, d.h. in der geplanten Bearbeitungsgeschwindigkeit (z.B. 36 mm/s), an das Werkstück 300 angenähert. Dies ist die Annäherung. Dies wird gemacht, damit beim tatsächlichen Kontakt des Werkzeugs 101 mit dem Werkstück 300, also dem Werkzeugeintritt, die Bearbeitung bereits beginnt. Darüber hinaus hat die langsame Geschwindigkeit den Vorteil, dass die Bearbeitung in der gewünschten Qualität erfolgen kann. Eine zu schnelle Geschwindigkeit könnte möglicherweise zu einer zu hohen Kraft beim Beginn des Materialabtrag und damit zu einer unregelmäßigen Bewegung des Werkzeugs 101 führen, sogenanntes Rattern, wodurch unerwünschte Abtragungen, sogenannte „Rattermale“, die Qualität beeinträchtigen können. Darüber hinaus könnte das Werkzeug 101 beim Werkzeugeintritt durch zu hohe Kraft Schaden nehmen.

[0046] Nach dem Werkzeugeintritt beginnt die Arbeitsphase. Bei einem Bohrer wäre dies die Bohrung und bei einer Fräse der Fräsvorschub oder Materialabtrag. Die Arbeitsphase endet mit dem Werkzeugaustritt.

[0047] Anschließend erfolgt die Abfahrt, die wieder schneller durchgeführt werden kann, wodurch Zeit im Gesamtprozess gespart werden kann. Die Abfahrt endet, sobald das Werkzeug 101 an der gewünschten Endposition angekommen ist. Die Endposition kann eine Ruheposition sein, oder auch eine Position außerhalb des Werkstückes 300, so dass beispielsweise das Werkstück 300 entfernt und das nächste Werkstück 300 bereitgestellt werden kann. Möglich ist auch, dass sich eine bewegliche Fördereinrichtung, auf der die Werkstücke 300 angeordnet sind, weiterbewegen soll, dann wird das Werkzeug 101 in eine Position gebracht, bei der das möglich ist.

[0048] Bei der Abfahrt werden ebenfalls wieder die vorhandenen Geometrien berücksichtigt, d.h. das Werkzeug 101 wird derart vom Werkstück 300 fortbewegt, dass eine Kollision nicht möglich ist.

[0049] In einer Ausführungsform wird die Abfahrt direkt als Anfahrt für die nächste Bearbeitung ausgeführt. Dies ist beispielsweise der Fall, wenn in ein Werkstück 300 mehrere Löcher gebohrt werden sollen. Das Bearbeiten eines Werkstückes 300 kann also mehrere Bearbeitungen umfassen.

[0050] Zurückkehrend zur Fig. 1, wird nun die Rechenvorrichtung 110 genauer beschrieben. Die im Folgenden beschriebenen Elemente der Rechenvorrichtung 110 können dementsprechend auch alle oder vereinzelt Teil der Werkzeugmaschine 100 sein. Die Rechenvorrichtung 110, oder die Werkzeugmaschine 100, kann dementsprechend eine Recheneinheit 120 umfassen. Diese wird auch Prozessor oder CPU (engl. central processing unit) genannt. Die Recheneinheit 120 kann auch aus mehreren Prozessoren bestehen, wobei ein oder mehrere davon auch unterstützende Prozessoren sein können, wie z.B. GPUs (engl. graphics processing unit). Es können auch Prozessoren anderer Rechner verwendet werden, d.h. die Verarbeitung wird ausgelagert.

[0051] Die (Rechen-)Vorrichtung 110 kann weiterhin eine Speichereinrichtung 130 umfassen, auf der Eingabe-, Ausgabe-, Zwischenergebnis- und/oder Programmdateien abgelegt sein können. Auch Steuerdaten der numerischen Steuerung können auf der Speichereinrichtung abgelegt sein. Die Speichereinrichtung 130 kann sich auf einen Speicher 103, eine Datenbank 104 /und oder einen Datenträger 105 beziehen, wobei letzterer in einer entsprechenden Schnittstelle, wie etwa Laufwerk, oder kabelgebundene oder kabellose Schnittstelle, an die Rechenvorrichtung 110 bzw. die Werkzeugmaschine 100 angebunden ist.

[0052] Die Vorrichtung 110 oder die Werkzeugmaschine 100 umfasst weiterhin ein trainiertes neuronales Netzwerk 140, kurz NN. Das NN 140 kann in verschiedenen - bekannten - Ausprägungen gestaltet sein. Zur Erläuterung der vorliegenden Erfindung wird exemplarisch eines näher beschrieben, jedoch wären auch gängige Alternativen denkbar, wie etwa ein Autoencoder oder ein Convolutional Neural Network (CNN).

[0053] Das beschriebene Beispiel-NN wird gemäß der U-Net-Architektur gebildet, wie in Fig. 3 gezeigt. Die Merkmale und die Funktionsweise der U-Net-Architektur sind im Allgemeinen bekannt, daher wird nicht in allen Details darauf eingegangen. Wird die U-Net Architektur verwendet, kann das NN durch die Convolution-Filter die Features für die Berechnung selbst extrahieren und diese erlernen. Durch diesen Aufbau wird eine (Bild-)Segmentierung verwirklicht. Gemäß der vorliegenden Erfindung wird eine Zeitreihe segmentiert, die für das NN als einzeliges Bild interpretiert wird. Nach einer Faltung wird eine Batch-Normalisierung durchgeführt mit dem Zweck, die Schichten durch Neuzentrierung und Neuskalierung schneller und stabiler zu machen. Abschließend werden die Neuronen mit einer Aktivierungsfunktion aktiviert, beispielsweise mit einer rectified linear activation function (ReLU). Nach jeweils zwei Mal Falten und Normalisieren werden mittels Max-Pool Layers die markanten Features für die nächsten Schichten weiterverwendet, so dass die Anzahl der Parameter nicht zu groß wird.

[0054] Der Input in das trainierte NN 140 ist dann dementsprechend der Größe $T \times 1$, wobei T die Länge der Zeitreihe ist, also die Anzahl an Messdaten zu jeweiligen Zeitpunkten (Timestamps). Diese Länge ist variabel und muss auf Grund der gewählten Architektur auch nicht bekannt sein, denn das U-Net (wie auch das CNN) verwendet nur Faltungen (engl. Convolutions), transponierte Convolutions, Max-Pooling, Normalisierung und Aktivierungsfunktion, die alle an keine fixe Eingangsgröße gebunden sind. Das ermöglicht, dass die Größe des Inputs variabel ist und nicht fix definiert sein muss. Somit können Bearbeitungen unterschiedlicher Länge verarbeitet werden.

[0055] Für die Berechnung der Faltung muss die Größe des Kernels, sowie die Parameter Dilation, Padding und Stride (Schrittweite) festgelegt werden. Für das U-Net im Versuch wurde die Größe des Kernels auf 5, die Dilation auf 7 und das Padding auf 14 sowie der Stride auf 1 gesetzt. Mit der Wahl dieser Parameter ist die Ausgangsgröße der Zeitreihe nach den Berechnungen in dieser Ebene gleich wie die Eingangsgröße der Zeitreihe. In den tieferen Ebenen bleibt die Eingangsgröße der so genannten Feature-Mappen gleich. Die Normalisierung verändert diese Länge ebenfalls nicht. Der Pooling-Layer verringert jedoch die Eingangsgröße, je nach gewählten Parametern. Bei der Wahl von Kernelgröße 2, Padding 0, Dilation 1 sowie Stride 2 verändert sich die Eingangsgröße N mit jedem Pooling Layer nach der Berechnung $\left\lfloor \frac{N-2}{2} + 1 \right\rfloor$. Daher ergibt sich eine Komprimierung in den tieferen Lagen, die mit der Decodierung wieder aufgelöst wird und die Zeitreihe am Ende wieder ihre Länge T aufweist.

[0056] Auf der Decodierungsseite sind die Parameter für die Faltung gleich gewählt, für die transponierte Faltung wird die Kernelgröße auf 5 gesetzt, der Stride auf 2, die Dilation auf 1, das Padding auf 1 sowie ein finales Output Padding ebenfalls auf 1. Dadurch ergibt sich nach der transponierten Faltung eine Ausgangsgröße von $2 \times N + 2$. Dieser Ausgangs-layer wird mit den Merkmalen der zugehörigen Enkodierung als Input zusammengeführt und auf gleiche Größe geführt (das sogenannten copy and crop Verfahren) und führen letzten Endes wieder zur Ausgangsgröße T . Die Parameter werden hier nicht genauer erläutert, da sie einer Fachperson bekannt sind und in gängiger Literatur nachzulesen sind. Die Wahl der Parameter wurde in Versuchen, um die beste Genauigkeit für die vorliegenden Daten zu finden, ermittelt und kann entsprechend angepasst werden.

[0057] Im Folgenden wird bezugnehmend auf Fig. 3, die Funktionsweise in der Inferenz anhand einer Beispielkonstellation erläutert. Die Eingangsdaten, d.h. die Zeitreihe mit den Drehmomentwerten werden als Bild der Größe 100×1 interpretiert.

[0058] Die Netzwerkarchitektur besteht aus einem kontrahierenden Pfad (linke Seite) und einem expansiven Pfad (rechte Seite). Der Kontraktionspfad folgt der typischen Architektur eines Fal-

tungsnetzwerks. Es besteht aus der wiederholten Anwendung von zwei 5x1-Faltungen (Faltungen ohne Padding, in Fig. 3 durch einfache Pfeile -> nach rechts dargestellt), jeweils gefolgt von einer ReLU und einer 2x1-Max-Pooling-Operation (in Fig. 3 durch Pfeile nach unten dargestellt) mit Stride 2 zum Downsampling. Bei jedem Downsampling-Schritt verdoppeln sich die Anzahl der Feature-Kanäle. Jeder Schritt im expansiven Pfad besteht aus einem Upsampling der Feature Map, gefolgt von einer 5x1-Faltung (Aufwärtsfaltung, in Fig. 3 durch Pfeile nach oben dargestellt), die die Anzahl der Merkmalskanäle halbiert, einer Verkettung mit der entsprechend zugeschnittenen Merkmalskarte aus dem Kontraktionspfad und zwei 5x1 Faltungen (in Fig. 3 wieder durch einfache Pfeile -> nach rechts dargestellt), denen jeweils eine ReLU folgt.

[0059] Das Zuschneiden und Übertragen (engl. crop and copy, in Fig. 3 durch dreifache Pfeile => nach rechts dargestellt) ist aufgrund des Verlusts von Randpixeln in jeder Faltung erforderlich. Auf der letzten Schicht wird eine 1x1-Faltung (in Fig. 3 durch einen doppelten Pfeil => nach rechts dargestellt) verwendet, um jeden Merkmalsvektor mit 16 Komponenten auf die gewünschte Anzahl von Klassen abzubilden. Insgesamt hat das Netzwerk 18 Faltungsschichten. Um eine nahtlose Kachelung der Ausgabe-Segmentierungskarte zu ermöglichen (siehe Abbildung 2), ist es wichtig, die Größe der Eingabekacheln so zu wählen, dass alle 5x1-Max-Pooling-Operationen auf eine Ebene mit einer gleichmäßigen x- und y-Größe angewendet werden.

[0060] Die Eingangsdaten müssen jedoch aufgrund der Komprimierung in der gewählten Konstellation eine Mindestgröße von 32 Timestamps vor der ersten Schicht aufweisen. Hinsichtlich der Abstände zwischen den Zeitpunkten (Timestamps) gibt es keine zwingenden Umstände, jedoch hat sich herausgestellt, dass ein Abstand von 0,006 Sekunden zwischen den Messpunkten zu guten Ergebnissen führt. Daher wäre in diesem Fall mindestens eine Zeitreihe der Länge 0,186 Sekunden erforderlich, was 32 Messpunkten mit 31 Abständen zu je 0,006 Sekunden entspricht.

[0061] Fig. 4 zeigt beispielhafte Ergebnisse des NN 140. Oben zu sehen sind 3 verschiedene Erfassungen von Drehmomentwerten gegen die Zeit, und in den unteren Abbildungen kann die durch das NN 140 erstellte Auswertung gesehen werden.

[0062] Dabei sind die Bereiche, in denen das Werkzeug als „im Werkstück“ ermittelt wurde, mit einem gestrichelten Rahmen markiert eingefärbt. Das nach dem Durchlaufen des NN 140 als Ergebnis anzusehende Merkmal gibt in binärer Weise, d.h. für jeden Drehmomentmesswert wird die Klassifikation zu 0 oder 1 vorgenommen, wieder ob das Werkzeug außerhalb des Werkstücks ist (Wert 0) oder ob das Werkzeug innerhalb des Werkstücks ist (Wert 1). Somit können die Übergänge 0 zu 1 und 1 zu 0 als Werkzeugeintritt bzw. -austritt bestimmt werden, und sind durch die Grenze des gestrichelten Bereichs, d.h. der gestrichelten Linie, in Fig. 4 bestimmt.

Wie oben erwähnt kann alternativ auch eine Autoencoder-Architektur verwendet werden, die die Größe der eingegebenen Informationen reduziert. Danach wird hier eine Decodierung durchgeführt, und anschließend die lineare Merkmalsdarstellung erlernt, wodurch der Raster allmählich größer wird. Am Ende dieser Architektur entspricht die Ausgabegröße der Eingabegröße. Ein bekannter Vorteil der Autoencoder-Architektur ist, dass die Anfangsgröße erhalten werden kann. Diesen Vorteil hat die U-Net-Architektur ebenfalls. Allerdings wird hier, beim Autoencoder, der Input linear komprimiert, sodass nicht alle Merkmale übertragen werden können.

[0063] Demgegenüber ist die U-Net-Architektur besser geeignet, da durch die U-Form die Entfaltung auf der Seite des Decoders erfolgt, wodurch das bei einer Auto-Encoder-Architektur auftretende Flaschenhalsproblem und damit der Verlust von Merkmalen vermieden wird. Konkret wird dem neuronalen Netzwerk bei jeder Entfaltung die Merkmale der zugehörigen Enkodierung als Input (auch „copy and crop“ genannt, engl. für Kopieren und Zuschneiden). Diese sind in Fig. 3 durch dreifache Pfeile $\Rightarrow$ nach rechts dargestellt. Dadurch kann das NN Merkmale, die es bei der Enkodierung (auch Faltung genannt) lernt, auch für die Dekodierung (auch Entfaltung genannt) verwenden, was es vom CNN unterscheidet. Daher ist die U-Net-Architektur gegenüber dem Autoencoder sowie dem CNN im Vorteil, da so das Lernen signifikant verbessert wird. Insbesondere von Vorteil ist der Einsatz der U-Net-Architektur, da dieses bekanntermaßen mit relativ wenig Trainingsdaten trainiert werden kann. Daher können mit der U-Net-Architektur mit weniger Trainingsdaten bessere Ergebnisse erzielt werden als mit anderen neuronalen Netzwerkarchitek-

turen, beispielsweise auch als mit einem Autoencoder, die oft mehrere Tausend Trainingsdaten benötigen.

[0064] Das in den Ausführungsformen eingesetzte NN 140 muss zuvor trainiert werden. Dies geschieht durch ein Verfahren 500 zum Trainieren des NN 140. Zum Training werden zuerst die Daten bereits durchgeführter Bearbeitungen in Schritt 510 erhalten. Das Erhalten 510 kann verwirklicht werden, indem die entsprechenden Trainingsdatensätze erzeugt werden, beispielsweise durch wiederholtes Durchführen der zu trainierenden Bearbeitung, oder auch indem Daten früherer Bearbeitungen (d.h. Wiederholungen der Bearbeitung) von einem Datenspeicher geladen werden, auf dem diese zuvor gespeichert wurden. In anderen Worten, falls die jeweilige Bearbeitung bereits mehrfach durchgeführt wurde, so stehen diese Daten unter Umständen bereits in gespeicherter Form zur Verfügung. Sollten entsprechende Trainingsdaten nicht vorliegen, so müssen diese - durch wiederholtes Durchführen der entsprechenden Bearbeitung - erst generiert werden.

[0065] Konkret wird das Training als überwachtes Training durchgeführt, bei dem sowohl die Eingangsdaten als auch die erwarteten Ausgangsdaten vorliegen. Für die jeweilige Bearbeitung, also beispielsweise für die Bohrung/das Fräsen eines Loches in einem Werkstück, müssen eine Vielzahl von Daten von genau dieser Bohrung/Fräsung vorliegen. Jede dieser Bearbeitungen (also Bohrung bzw. Fräsung) umfasst jeweils die Phasen Annäherung, Werkzeugeintritt, Arbeitsphase und Werkzeugaustritt. Es können weiterhin Teile von An- und Abfahrt enthalten sein. Für das Training werden diese aber nicht verwendet, und können, falls sie vorhanden sind, mit Steuerungsinformation ausgefiltert werden.

[0066] Es hat sich als ausreichend erwiesen, einige Hundert oder mehr Bearbeitungen durchzuführen, bzw. deren Daten zu verwenden. Falls mehr Datensätze verwendet werden, müssen mehr Datensätze verarbeitet werden, das Ergebnis erlaubt aber ein besseres Training und dementsprechend eine höhere Genauigkeit in der Inferenz. Diese werden zuerst durch fachkundiges Personal aufbereitet, wozu beispielsweise eine Annotierungssoftware verwendet werden kann, um die entsprechenden Zeitreihen zu annotieren bzw. zu segmentieren.

[0067] Für das Segmentieren, als Schritt 520, werden dabei beispielsweise die Bereiche als 0 (außerhalb des Materials, d.h. Werkstück 300) und 1 (innerhalb des Materials, d.h. während der Materialabtragung im Werkstück 300) zu jedem Zeitpunkt abgespeichert. Die Einteilung in diese Klassen, 0 und 1, wird hierbei anhand des Drehmoments durchgeführt, wobei die Drehmomentwerte jeweils ein signifikant unterschiedliches Muster aufweisen, je nachdem ob sich das Werkzeug im Leerlauf befindet (außerhalb des Werkstückes 300, d.h. vor und nach der Arbeitsphase bzw. Bearbeitung), oder ob ein Eintritt bzw. Austritt in bzw. aus dem Werkstück 300 stattfinden. Die Einteilung kann auch weitere Unterbereiche enthalten, beispielsweise die oben genannten Phasen einer Bearbeitung. Darüber hinaus ist die Bezeichnung der Bereiche mit 0 und 1 lediglich als Beispiel zu verstehen.

[0068] Anschließend wird in Schritt 530 jeder Trainingsdatensatz an dessen Rändern um unwichtige Daten, beispielsweise Null- oder Mittelwerte, erweitert, da die Randbereiche durch die Faltung verloren gehen, bzw. ignoriert werden. Dies nennt man Padding. Bei den Zeitreihen werden am Anfang und am Ende jeder Zeitreihe solche Paddingdaten ergänzt. In Versuchen wurde festgestellt, dass für die vorhandenen Daten 20 derartige Paddingdaten jeweils am Anfang und Ende gute Ergebnisse liefern.

[0069] Das NN 140 kann dann beispielsweise mit einem Train-Test Split von 80% und 20% im Schritt 540 trainiert werden. Die prozentuale Verteilung ist hierbei lediglich als Beispiel anzusehen. Es sind auch andere Werte möglich, in der Regel werden jedoch für die im Folgenden näher beschriebene Testphase des Trainings verhältnismäßig weniger Daten verwendet als für das Training. Mit der oben genannten 80% zu 20% Aufteilung werden 80% der vorhandenen Daten für das Training (das sogenannte Trainingsset) verwendet, die verbleibenden 20% der Daten werden nach jedem Lernvorgang zum Testen (so genanntes Testset) verwendet. Nach jedem Trainingsdurchlauf wird der Fehler zwischen Ausgabe des Netzes und der erwarteten Ausgabe, der sogenannte Loss, sowohl auf dem Trainingsset wie auch dem Testset berechnet. Dieser Vor-

gang dient dazu, das Training zu überwachen. Der Fehler für das Trainingsset wird im Laufe des Trainings kleiner, weil das Netz so konstruiert ist, die Ausgabe basierend auf den enthaltenen Mustern der Trainingsdaten zu optimieren. Lernt das Netz die tatsächlich relevanten Merkmale, so verringert sich auch der Fehler auf dem Testset. Sobald der Fehler des Testsets steigt, der Fehler des Trainingssets weiterhin sinkt, so spricht man von Überanpassung (englisch „overfitting“) und das Training kann beendet werden. In Versuchen wurde gezeigt, dass ein Training mit einer Batch-Size von 100 Samples pro Trainingsvorgang gut funktioniert. Üblicher Weise sind Batch-Größen zwischen 50 und 200 Samples. In einem beispielhaften Versuch konnte beispielsweise nach 43 Epochen keine signifikante markante Verbesserung mehr im Loss festgestellt werden und das Training wurde dann beendet. Der Loss wird beispielsweise mit Hilfe einer Metrik, z.B. Binary Cross Entropy, Dice Koeffizient oder Mean Square Error berechnet. Für die Optimierung wird anschließend ein Optimierungsalgorithmus eingesetzt. Beispiele für solche Optimierungsalgorithmen sind beispielsweise der Adam Algorithmus oder der Stochastic Gradient Optimizer. Weiterhin kann eine initiale Learning-Rate von 0,001 festgelegt werden, diese kann beispielsweise nach jeweils 10 Updates mit dem Faktor 0,1 multipliziert werden. Die Parameter Batch-Size, Epochen, Learning-Rate und Skalierungsfaktor sind sogenannte Hyperparameter und können sich je nach konkretem Datenset unterscheiden, die angegebenen Parameter haben in Versuchen zum besten Ergebnis, gemessen an der Genauigkeit, geführt. Um die bestmögliche Parameterkombination zu finden, werden Versuche mit unterschiedlichen Kombinationen durchgeführt und die Kombination gewählt, die zum besten Ergebnis führt.

[0070] Das trainierte NN 140 kann dann zum Optimieren einer Werkzeugmaschine 100 mit einem Werkzeug 101 eingesetzt werden, wie unter Bezugnahme auf Fig. 2 im Folgenden dargelegt.

[0071] Hierbei werden im Verfahren 200 zuerst im Schritt 210 numerische Werkzeuginformationen, die eine Bearbeitung betreffen erhalten. Eine Bearbeitung ist in diesem Zusammenhang, wie oben bereits erwähnt beispielsweise als eine Bohrung bzw. Fräsung zu verstehen, die jeweils die Phasen Anfahrt, Annäherung, Werkzeugeintritt, Arbeitsphase, Werkzeugaustritt und Abfahrt umfasst. Es müssen jedoch nicht für alle Phasen Informationen erhalten werden. Insbesondere Informationen für die Phasen An- und Abfahrt sind nicht notwendig.

[0072] Die Werkzeuginformationen umfassen hierbei für jeden Zeitpunkt der Bearbeitung mindestens das Drehmoment, beispielsweise ein Spindeldrehmoment, des Antriebes des Werkzeugs 101 sowie die Achspositionen des Werkzeugs 101, die eineindeutig die jeweilige Position des Werkzeuges dreidimensional zum jeweiligen Zeitpunkt bestimmen. Es können weitere Informationen das Werkzeug und die Bearbeitung umfasst sein. Beispielsweise, der Werkstoff des Werkstücks, eine Seriennummer des Werkzeugs oder anderer in die Bearbeitung involvierter Elemente oder eine Drehzahl, z.B. die Spindeldrehzahl, oder eine andere, die den Antrieb des Werkzeugs 101 betreffende Kennzahl oder Messwerte.

[0073] Das wie oben beschrieben trainierte NN 140 ermittelt dann in einem Schritt 220 aus den erhaltenen Werkzeuginformationen der einen Bearbeitung Zeitinformationen. Diese Zeitinformationen umfassen den Zeitpunkt des Werkzeugeintrittes in das Werkstück und den Zeitpunkt des Werkzeugaustrittes aus dem Werkstück.

[0074] Hierbei wird, wie das NN 140 gelernt hat, die Zeitreihe in Verbindung gebracht mit den Drehmomenten, und die entsprechenden Zeitpunkte klassifiziert, so dass durch das NN 140 der Eintritt und der Austritt erkannt werden können.

[0075] Sind der Zeitpunkt des Werkzeugeintritts und der Zeitpunkt des Werkzeugaustritts bekannt, so ist automatisch auch der Zeitraum, den die Arbeitsphase einnimmt, bekannt, da dieser durch Werkzeugeintritt bzw. -austritt begrenzt ist.

[0076] Anschließend wird in einem Schritt 230 aus den ermittelten Zeitpunkten und den Werkzeuginformationen ein Satz von Bahninformationen erzeugt. Diese Bahninformationen geben im Wesentlichen die Bahn wieder, die das Werkzeug 101 bei der einen Bearbeitung zurückgelegt hat. Die Bahninformationen umfassen den Fahrweg des Werkzeugs und den Abstand des Werkzeugs zum Werkstück vor der Phase der Annäherung. Der Fahrweg umfasst hierbei die

Phasen Annäherung, Werkzeugeintritt, Arbeitsphase und Werkzeugaustritt. An- und Abfahrt können ebenfalls umfasst sein. Aus den Bahninformationen und den Werkzeuginformationen können überdies weitere Informationen abgeleitet bzw. berechnet werden. Beispielsweise kann zu jedem Zeitpunkt die Position des Werkzeugs aus den Achspositionen ermittelt werden. Da die Bearbeitung auch bereits abgeschlossen ist, kann - im Nachhinein - auch der Abstand des Werkzeugs 101 zum Eintrittspunkt ins Werkstück 300 exakt ermittelt werden.

[0077] Der Abstand des Werkzeugs 101 zum Werkstück 300 hängt jedoch im Allgemeinen auch von Fertigungstoleranzen des Werkstücks ab. So kann eine Metallplatte durchaus eine Toleranz von mehreren Millimetern aufweisen, so dass der Abstand im Voraus eben nicht exakt bestimmt werden kann, wodurch das langsame Annähern überhaupt erst notwendig ist. Die exakten Maße an der Eintrittsstelle sind nach der Bearbeitung zwar bekannt, durch den Eintrittszeitpunkt, jedoch nicht die Toleranzen an anderen Stellen.

[0078] Daher kann der Gesamtabstand des Werkzeugs vom Werkstück andere Stellen als den Eintrittspunkt nur innerhalb der Toleranzen, die für die Bearbeitung relevant sind, bestimmt werden.

[0079] In einem Schritt 240 wird dann der Minimalabstandes des Werkzeugs 101 zum Werkstück 300 vor der Phase Annäherung aus mehreren Sätzen von Bahninformationen mehrerer vorheriger Bearbeitungen für die nächste Bearbeitung bestimmt. Hierzu werden Daten vorheriger Bearbeitungen, d.h. der für frühere Bearbeitungen generierte Abstand des Werkzeugs 101 zum Werkstück 300 vor der Phase der Annäherung, ausgewertet, und falls der für die aktuelle Bearbeitung generierte Abstand geringer ist als die der vorherigen, dann wird der Mindestabstand auf den Abstand der aktuellen Bearbeitung vergrößert.

[0080] Dies bedeutet, dass falls der Minimalabstand, der für die aktuelle Bearbeitung nicht ausgereicht hat, weil das Werkzeug 101 früher auf das Werkstück 300 getroffen ist als erwartet, dann wird der Minimalabstand für die nächste Bearbeitung vergrößert.

[0081] Durch regelmäßige Kontrollen können Ausreißer, die beispielsweise dadurch entstehen können, dass ein Werkstück deformiert, fehlerhaft oder verunreinigt war, aus den Datensätzen entfernt werden, so dass diese unberücksichtigt bleiben.

[0082] Wie bereits zuvor erwähnt, können die Daten für die Durchführung des Verfahrens auch aus gespeicherten Daten stammen. Es können also die Daten von der numerischen Steuerung 102 empfangen 211 werden, beispielsweise direkt nach einer Bearbeitung. Es können aber stattdessen oder auch zusätzlich Daten aus einem Speicher 103, aus einer Datenbank 104 oder von einem Datenträger 105 geladen bzw. gelesen 212 werden.

In einer Ausführungsform können die Werkzeuginformationen durch eine Vorverarbeitung bereinigt 215 werden. Dabei können beispielsweise vor dem Ermitteln der Zeitinformationen Informationen entfernt werden, die für die Bestimmung des Minimalabstandes irrelevant sind. Dies kann beispielsweise durch eine Filterung durch z.B. einen Signalfilter oder Frequenzfilter durchgeführt werden. Alternativ können auch Zeitpunkte, Zeitbereiche oder Bearbeitungsphasen weggelassen werden. Hierbei ist es beispielsweise vorteilhaft, nur die Daten jener Phasen zu betrachten, in denen das Werkstück auch bearbeitet wird. Dazu kann zwischen den Phasen der Positionierung der Achsen (Anfahrt, Annäherung, Abfahrt) und des aktiven Verschiebens der Achsen für die Arbeitsphase unterschieden werden. Es können auch Daten ausgeblendet werden in denen ein anderer, nicht Bearbeitungs-bezogener Prozess abläuft. Möglich wäre auch beispielsweise, eventuell vorkommende Phasen einer Repositionierung wegzufiltern.

[0083] Dadurch würden nur die Daten an das neuronale Netzwerk weitergeleitet, die für die Ermittlung, bzw. Generierung und/oder Bestimmung relevanter Daten notwendig sind.

[0084] Darüber hinaus kann die Datenbasis für die Ermittlung durch das neuronale Netzwerk durch für neuronale Netzwerke übliche Vorverarbeitungsschritte vorbereitet werden. Darunter fallen beispielsweise Skalierung, Padding, etc.

[0085] Diese vorverarbeitenden Schritte können beispielsweise durch die Datenverteilung über einen Nachrichtenbus (engl. message broker oder message bus) durchgeführt und mit Skripten

für die Datenvorverarbeitung sowie Berechnung durch das neuronale Netz durchgeführt werden. Dies hat den Vorteil, dass die Ergebnisse annähernd in Echtzeit zur Verfügung stehen. Alternativ können diese durch eine Recheneinheit auch anders bereitgestellt werden. Auch das Auslesen von Daten aus einer Datei und iteratives Abarbeiten davon ist möglich. Auch ein Auslesen der Daten aus einer Datei oder einem historischen Speicher (z.B. einer Datenbank, die Zeitreihen früherer Bearbeitungen und/oder Simulationen enthält) ist möglich. Diese können dann auch wieder in einzelne Bearbeitungen aufgetrennt werden (mit Hilfe der Steuerinformationen) und weiterverarbeitet werden. Es können auch einzelne Bearbeitungen als Datei abgespeichert sein, wodurch keine Aufteilung mehr notwendig ist, und sogleich mit der Bearbeitung - gegebenenfalls nach entsprechenden Vorverarbeitungen für das Modell, wie oben beschrieben - begonnen werden kann.

[0086] Da, wie oben beschrieben, nach der Bearbeitung alle notwendigen Daten vorliegen, können im Rahmen des Verfahrens auch in einem weiteren Schritt 241 zusätzlich Minimalabstände des Werkzeugs 101 zum Werkstück 300 vor oder während der Anfahrt, während oder nach der Annäherung, oder während oder nach der Abfahrt bestimmt werden. Dies funktioniert, wie oben im Rahmen des Schritt 240 beschrieben, ebenfalls aus mehreren Sätzen von Bahninformationen mehrerer vorheriger Bearbeitungen, und legt gegebenenfalls jeweils den entsprechenden Minimalabstand für die nächste Bearbeitung fest.

[0087] Es kann beispielsweise festgestellt werden, dass das Werkzeug 101 bei der Anfahrt relativ nahe an ein aufragendes Teil des Werkstücks herankam. Um eine mögliche Kollision bei zukünftigen Bearbeitungen noch unwahrscheinlicher zu machen, kann der entsprechende Minimalabstand vergrößert werden, und bei der Bahnplanung der nächsten Bearbeitung berücksichtigt werden.

[0088] Bei der Bestimmung von Minimalabständen gemäß den Schritten 240 oder 241 können auch weitere Informationen berücksichtigt werden. Beispielsweise kann die Geometrie der Werkstücke vor und/oder nach der Bearbeitung berücksichtigt werden. Ein Beispiel hier könnte sein, dass ein Loch in einem halbkugelförmigen Werkstück platziert werden muss. Da die Geometrie der Halbkugelform bekannt ist, kann errechnet werden, welchen Abstand das Werkzeug zum Werkstück auf seiner Bahn hat bzw. haben muss. Die Geometrien vor und nach der Bearbeitung können sich unterscheiden, da evtl. Material abgetragen wird, so dass das Werkstück 300 nach der Bearbeitung anders gestaltet ist. Auch die Geometrie des Werkzeugs 101 kann berücksichtigt werden, genauso wie die Aufspanntoleranzen der Werkstücke 300 und der Werkzeugverschleiß.

[0089] Das NN 140 kann gegebenenfalls in Schritt 250 nachtrainiert werden. Wie oben dargelegt, kann die aktuelle Bearbeitung, oder mehreren Bearbeitungen, bei der/denen das trainierte NN 140 zum Einsatz kam, zum Nachtrainieren des trainierten neuronalen Netzwerks 140 verwendet werden. Hierbei können die ermittelten, generierten und bestimmten Informationen, oder einer Auswahl davon zum Nachtrainieren eingesetzt werden. Dazu werden diese vor dem Nachtrainieren auf ihre Korrektheit geprüft und ggf. korrigiert. Ein fehlerhaftes Labeling (also die Bezeichnung eines Bereiches mit einer falschen Bezeichnung (Label)) könnte nämlich zu einer Verschlechterung des Modells, d.h. des NN 140 führen.

[0090] Generell wird ein Nachtrainieren 250 dann in Betracht gezogen, wenn sich die Genauigkeit des Modells verschlechtert. Das kann beispielsweise passieren, wenn sich die Daten durch vorher unbekannte Einflüsse verändern, z.B. Verschleiß der Spindel oder des Werkzeugs, oder bei Veränderungen der Mechanik in anderen Maschinenkomponenten. Diese Einflüsse sind möglicherweise in den Daten, die zuvor für das Training verwendet wurden, nicht enthalten. Durch das Nachtrainieren kann das Datenset vergrößert werden und die Genauigkeit des Modells auch wieder verbessert werden.

[0091] Das Nachtrainieren 250, kann hierbei als Wiederaufnahme des Trainingsverfahrens 500 oder auch nach dem praktischen Einsatz als Zusatzschritt der Inferenz 200 gesehen werden.

[0092] In einer Ausführungsform wird eine Vorrichtung 110 bereitgestellt, die mindestens eine Recheneinheit 120, mindestens eine Speichereinrichtung 130 und ein trainiertes neuronales Netz

140 umfasst, wobei die Vorrichtung 110 dazu eingerichtet ist, die Schritte des oben dargelegten Verfahrens 200 auszuführen. So eine Vorrichtung 110 ist also den Bedingungen der Bearbeitung entsprechend programmiert, und umfasst das erfindungsgemäß trainierte NN 140. Die Vorrichtung 110 kann die Ausgestaltung eines Computers haben, der ohne Werkzeugmaschine 100 die Daten der früheren und aktuellen Bearbeitung aus seiner Speichereinrichtung 130 liest und die entsprechenden Abstände bestimmt. Die Speichereinrichtung 130 kann sich auf einen Speicher 103, eine Datenbank 104 /und oder einen Datenträger 105 beziehen, wobei letzterer in einer entsprechenden Schnittstelle, wie etwa ein Laufwerk, oder eine kabelgebundene oder kabellose Schnittstelle, an die Rechenvorrichtung 110 angebunden ist.

[0093] Alternativ ist die Vorrichtung 110 Teil der Werkzeugmaschine 100, wobei dann die zusätzliche Möglichkeit besteht, dass die Vorrichtung die Daten der aktuellen Bearbeitung direkt von der numerischen Steuerung 102 erhält.

[0094] Die Werkzeugmaschine 100 kann auch selbst entsprechende Elemente umfassen, die sie in die Lage versetzt, das Verfahren 200 wie oben dargelegt auszuführen. Auch in diesem Fall besteht die zusätzliche Möglichkeit, dass die Daten der aktuellen Bearbeitung direkt von der numerischen Steuerung 102 erhalten werden. Die Werkzeugmaschine 100 ist dann dazu eingerichtet, Werkstücke 300 zu bearbeiten, wobei die bestimmten Minimalabstände bei der Bearbeitung berücksichtigt werden.

[0095] Eine weitere Ausführungsform ist ein Computerprogrammprodukt, das Befehle umfasst, die bei der Ausführung des Programms durch einen Computer diesen veranlassen, eines der oben dargelegten Verfahren 200 oder 500 auszuführen.

[0096] Eine weitere Ausführungsform ist ein computerlesbares Medium, auf dem das Computerprogrammprodukt gespeichert ist.

[0097] Eine weitere Ausführungsform ist ein Verfahren 400 zum Bearbeiten eines Werkstücks 300 durch die zuvor beschriebene Werkzeugmaschine 100.

[0098] Die Ausführungsbeispiele zeigen mögliche Ausführungsvarianten, wobei an dieser Stelle bemerkt sei, dass die Erfindung nicht auf die speziell dargestellten Ausführungsvarianten derselben eingeschränkt ist, sondern vielmehr auch diverse Kombinationen der einzelnen Ausführungsvarianten untereinander möglich sind und diese Variationsmöglichkeit aufgrund der Lehre zum technischen Handeln durch gegenständliche Erfindung im Können der auf diesem technischen Gebiet tätigen Fachperson liegt.

[0099] Der Schutzbereich ist durch die Ansprüche bestimmt. Die Beschreibung und die Zeichnungen sind jedoch zur Auslegung der Ansprüche heranzuziehen. Einzelmerkmale oder Merkmalskombinationen aus den gezeigten und beschriebenen unterschiedlichen Ausführungsbeispielen können für sich eigenständige erfinderische Lösungen darstellen. Die den eigenständigen erfinderischen Lösungen zugrundeliegende Aufgabe kann der Beschreibung entnommen werden.

[00100] Sämtliche Angaben zu Wertebereichen in gegenständlicher Beschreibung sind so zu verstehen, dass diese beliebige und alle Teilbereiche daraus mitumfassen, z.B. ist die Angabe 1 bis 10 so zu verstehen, dass sämtliche Teilbereiche, ausgehend von der unteren Grenze 1 und der oberen Grenze 10 mit umfasst sind, d.h. sämtliche Teilbereiche beginnen mit einer unteren Grenze von 1 oder größer und enden bei einer oberen Grenze von 10 oder weniger, z.B. 1 bis 1,7, oder 3,2 bis 8,1, oder 5,5 bis 10.

[00101] Der Ordnung halber sei abschließend darauf hingewiesen, dass zum besseren Verständnis des Aufbaus Elemente teilweise unmaßstäblich und/oder vergrößert und/oder verkleinert dargestellt wurden.

BEZUGSZEICHENLISTE

- 100 Werkzeugmaschine
- 101 Werkzeug
- 102 numerische Steuerung
- 103 Speicher
- 104 Datenbank
- 105 Datenträger
- 110 (Rechen-)Vorrichtung
- 120 Recheneinheit
- 130 Speichereinrichtung
- 140 neuronales Netzwerk
- 200 Verfahren zum Optimieren
- 210 Erhalten von numerischen Werkzeuginformationen
- 211 Empfangen der Werkzeuginformationen
- 212 Auslesen der Werkzeuginformationen
- 215 Bereinigen der Werkzeuginformationen
- 220 Ermitteln von Zeitinformationen
- 230 Generieren eines Satzes von Bahninformationen
- 240 Bestimmen eines Minimalabstandes
- 241 Bestimmen von Minimalabständen
- 250 Nachtrainieren des neuronalen Netzwerks
- 300 Werkstück/e
- 400 Verfahren zum Bearbeiten eines Werkstücks
- 500 Verfahren zum Trainieren eines neuronalen Netzwerks
- 510 Erhalten von Trainingsdatensätzen
- 520 Segmentieren von Trainingsdatensätzen
- 530 Padding von Trainingsdatensätzen
- 540 Trainieren des neuronalen Netzwerks
- 550 Nachtrainieren des neuronalen Netzwerks

Patentansprüche

1. Computerimplementiertes Verfahren (200) zum Optimieren einer numerischen Steuerung (102) einer Werkzeugmaschine (100) mit einem Werkzeug (101) zum Bearbeiten eines Werkstückes (300), umfassend:
 - Erhalten (210) von numerischen Werkzeuginformationen eine Bearbeitung betreffend, wobei die Werkzeuginformationen für jeden Zeitpunkt der Bearbeitung ein Spindeldrehmoment des Werkzeugs (101) und Achspositionen des Werkzeugs (101) umfassen;
 - Ermitteln (220) durch ein trainiertes neuronales Netzwerk (140) von Zeitinformationen, wobei die Zeitinformationen umfassen: einen Zeitpunkt des Werkzeugeintrittes in das Werkstück und einen Zeitpunkt des Werkzeugaustrittes aus dem Werkstück, basierend auf den erhaltenen Werkzeuginformationen;
 - Generieren (230) eines Satzes von Bahninformationen für die Bearbeitung aus den ermittelten Zeitpunkten und den Werkzeuginformationen, wobei der Satz von Bahninformationen umfasst:
 - einen Fahrweg des Werkzeugs, der die Phasen Annäherung, Werkzeugeintritt, Arbeitsphase, und Werkzeugaustritt umfasst, und
 - einen Abstand des Werkzeugs zum Werkstück vor der Phase Annäherung; und
 - Bestimmen (240) eines Minimalabstandes des Werkzeugs (101) zum Werkstück (300) vor der Phase Annäherung aus mehreren Sätzen von Bahninformationen mehrerer vorheriger Bearbeitungen für die nächste Bearbeitung.
2. Verfahren (200) gemäß einem der vorherigen Ansprüche, wobei die Werkzeuginformationen weiterhin die Spindeldrehzahl des Werkzeugs (101) umfasst.
3. Verfahren (200) gemäß einem der vorherigen Ansprüche, wobei das Erhalten (210) der numerischen Werkzeuginformationen umfasst
 - ein Empfangen (211) der numerischen Werkzeuginformationen von der numerischen Steuerung (102), und/oder
 - ein Auslesen (212) der numerischen Werkzeuginformationen aus einem Speicher (103), einer Datenbank (104) oder einem Datenträger (105).
4. Verfahren (200) gemäß einem der vorherigen Ansprüche, wobei das Verfahren vor dem Schritt des Ermitteln (220) weiterhin ein Bereinigen (215) der Werkzeuginformationen durch Entfernen von Informationen, die für die Bestimmung des Minimalabstandes irrelevant sind, umfasst.
5. Verfahren (200) gemäß einem der vorherigen Ansprüche, wobei das Verfahren weiterhin ein Bestimmen (241) von einem oder mehreren Minimalabständen des Werkzeugs (101) zum Werkstück (300) vor oder während einer Anfahrt, während oder nach der Annäherung, oder während oder nach einer Abfahrt aus mehreren Sätzen von Bahninformationen mehrerer vorheriger Bearbeitungen für die nächste Bearbeitung umfasst.
6. Verfahren (200) gemäß einem der vorherigen Ansprüche, wobei beim Bestimmen (240, 241) von Minimalabständen weiterhin eines oder mehrere berücksichtigt werden aus Geometrien von Werkstücken vor der Bearbeitung, Geometrien von Werkstücken nach der Bearbeitung, Geometrie des Werkzeugs, Aufspanntoleranzen der Werkstücke und Werkzeugverschleiß.
7. Verfahren (500) zum Trainieren eines neuronalen Netzwerkes (140), umfassend die folgenden Schritte:
 - Erhalten (510) einer Vielzahl von Trainingsdatensätzen, wobei jeder Trainingsdatensatz aus numerischen Werkzeuginformationen besteht, die eine Bearbeitung betreffen, und die in Form einer Zeitreihe für jeden Zeitpunkt der Bearbeitung ein Spindeldrehmoment eines Werkzeugs einer Werkzeugmaschine umfassen;
 - Segmentieren (520) jedes Trainingsdatensatzes, wodurch die entsprechende Zeitreihe durch Markieren der Zeitpunkte für einen Werkzeugeintritt und für einen Werkzeugaustritt in mindestens drei Bereiche geteilt wird;

- Padding (530) jedes Trainingsdatensatzes, wodurch am Anfang und am Ende jeder Zeitreihe Fülldaten angefügt werden; und
Trainieren (540) des neuronalen Netzwerks (140) mit den Trainingsdatensätzen.
8. Verfahren (200, 500) gemäß einem der vorherigen Ansprüche, wobei das Verfahren weiterhin ein Nachtrainieren (250, 550) des trainierten neuronalen Netzwerks (140) unter Verwendung von ermittelten, generierten und bestimmten Informationen, oder einer Auswahl davon umfasst.
 9. Vorrichtung (110), umfassend mindestens eine Recheneinheit (120), mindestens eine Speichereinrichtung (130) und ein neuronales Netz (140), wobei die Vorrichtung (110) dazu eingerichtet ist, die Schritte eines Verfahrens (200) gemäß einem der Ansprüche 1 bis 8 auszuführen.
 10. Werkzeugmaschine (100) mit Mitteln, die geeignet sind, die Schritte eines Verfahrens (200) gemäß einem der Ansprüche 1 bis 6 auszuführen, und Werkstücke (300) zu bearbeiten, wobei die bestimmten Minimalabstände bei der Bearbeitung berücksichtigt werden.
 11. Computerprogrammprodukt, umfassend Befehle, die bei der Ausführung des Programms durch einen Computer diesen veranlassen, das Verfahren (200) nach einem der Ansprüche 1 bis 8 auszuführen.
 12. Computerlesbares Medium, auf dem das Computerprogrammprodukt nach Anspruch 11 gespeichert ist.
 13. Verfahren (400) zum Bearbeiten eines Werkstücks (300) durch die Werkzeugmaschine (100) gemäß Anspruch 10.

Hierzu 4 Blatt Zeichnungen

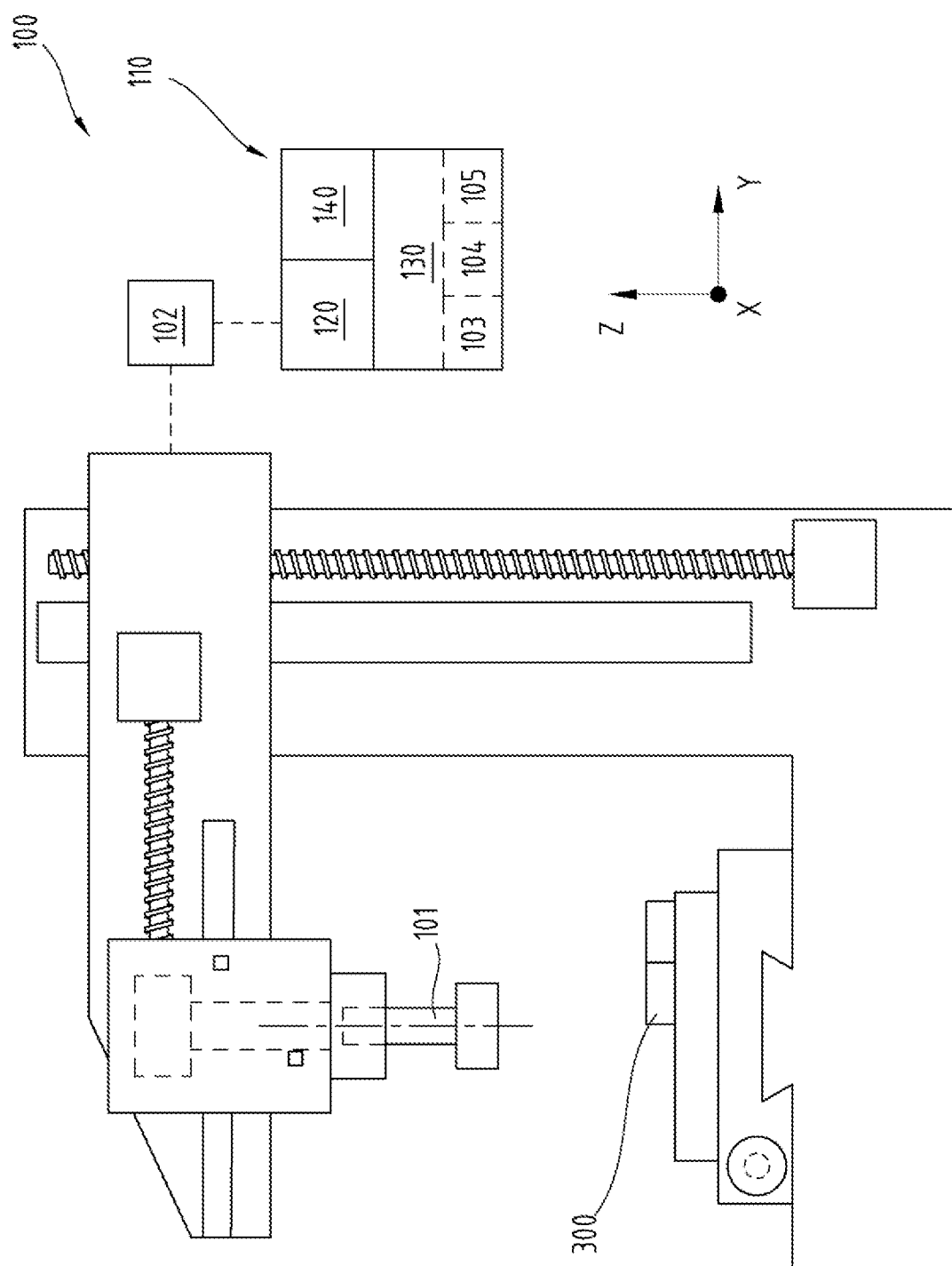


Fig. 1

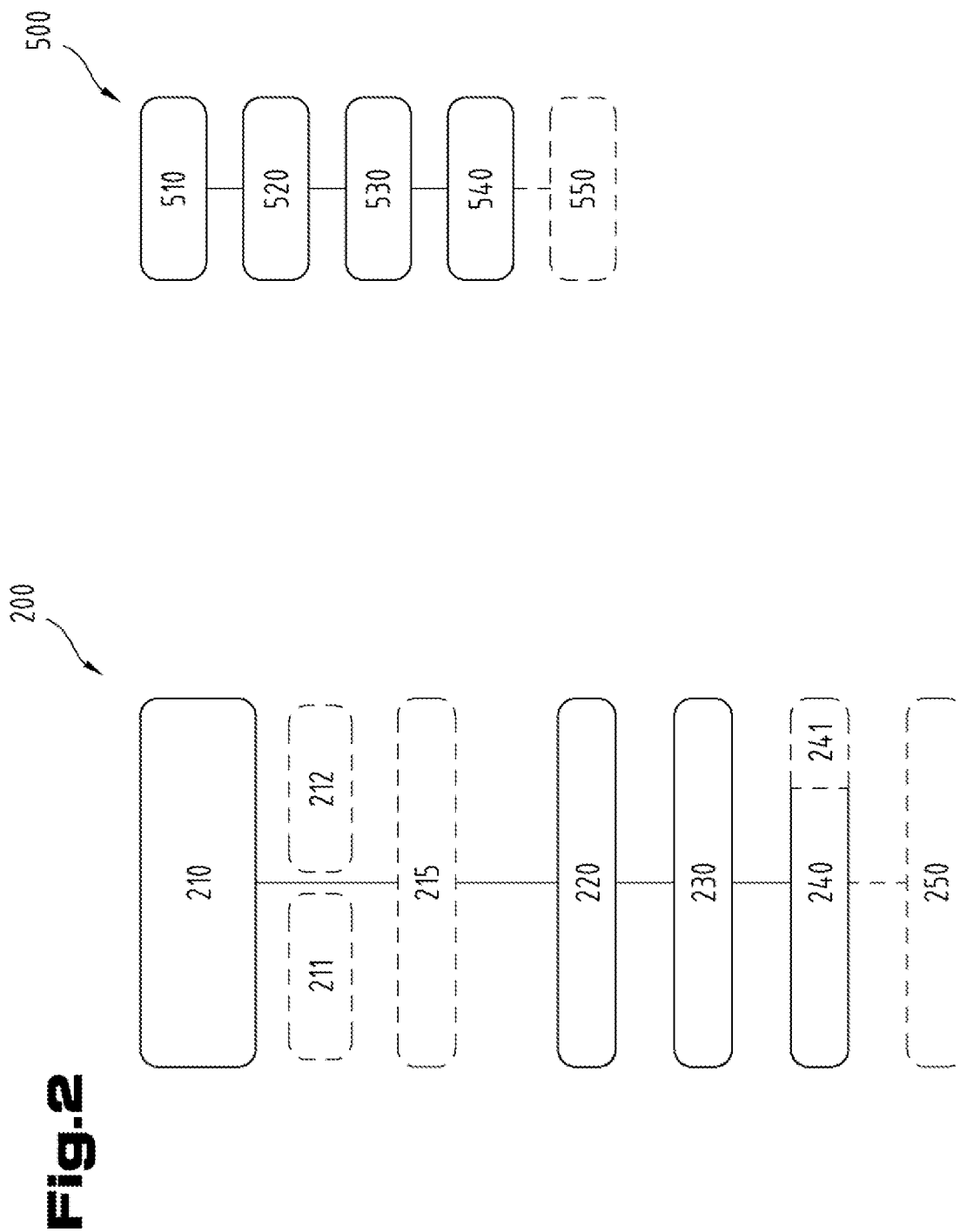


Fig. 3

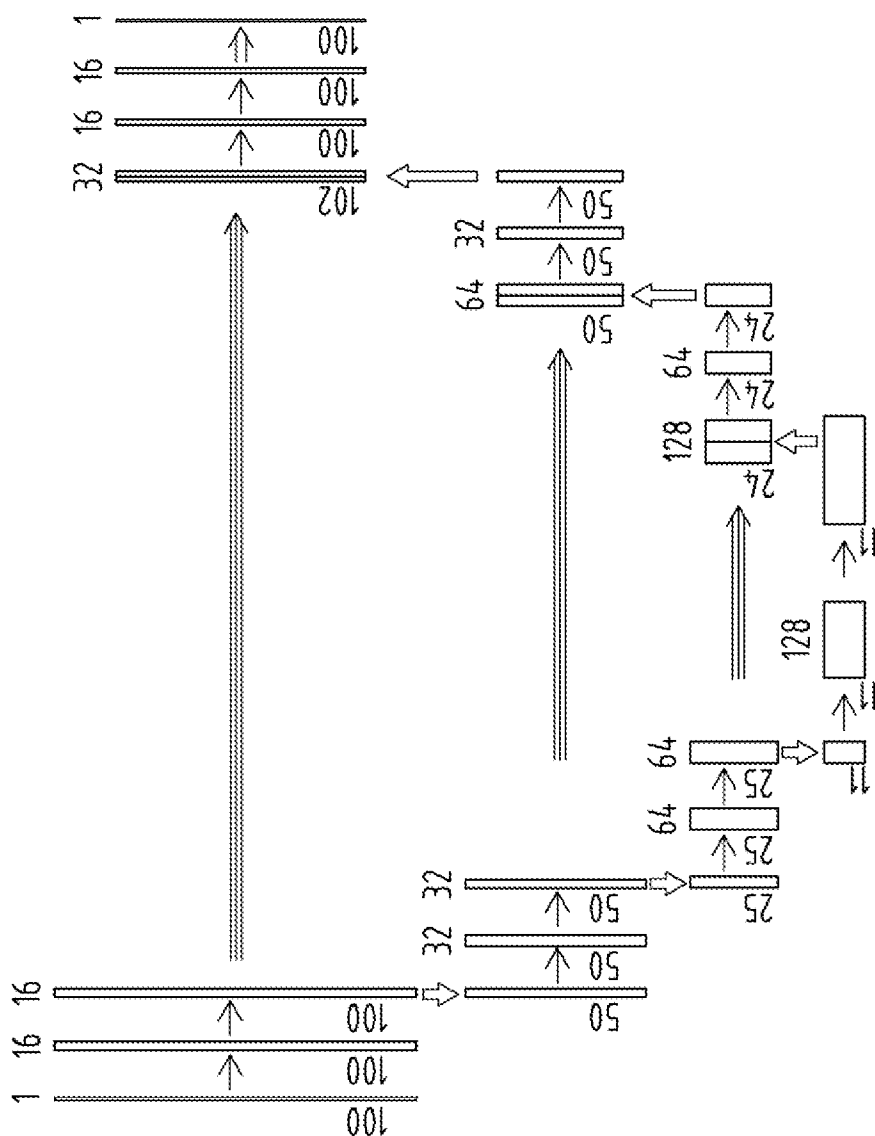


Fig.4

