

VOM KUNDENWUNSCH ZUR AUTOMATISIERTEN KABINENMONTAGE: EIN SYSTEM FÜR DIE KURZFRISTIGE INTEGRATION VON ANPASSUNGEN

G. Braun, M. Mayer, A. Schuster, M. Vistein, M. Malecha, L. Larsen

Deutsches Zentrum für Luft- und Raumfahrt, Zentrum für Leichtbauproduktionstechnologie,
Am Technologiezentrum 4, Augsburg, Deutschland

Abstract

Das DLR-Projekt DiCaDeMa (Digital Cabin Architectures and Design for Manufacturing) untersuchte eine Prozesskette mit digitaler Durchgängigkeit vom Kabinendesign bis zur teilautomatisierten Montage von Kabinenkomponenten. Besonderes Augenmerk wurde dabei auf die Möglichkeit gelegt, eine kurzfristige Modifikation der Konfiguration des Designs vor der Endmontage zu ermöglichen (Last Minute Customisation).

Im betrachteten Use Case wird, basierend auf einer Wunschkonfiguration der Hatracks, automatisiert die Plausibilität der Konfiguration überprüft und die Aufhängepunkte an den Frames berechnet. Die Prozessschritte werden für die Montagezelle generiert und simuliert. Kollisionsfreie Pfade werden anschließend an den Shopfloor geschickt. Für die abschließende Demonstration wurde ein Mock-Up in Originalgröße und ein AMR mit einem UR10 samt einem Endeffektor für das Markieren der Aufhängepunkte verwendet.

Dieses Paper beschreibt die Weiterentwicklung im Projekt und schließt die letztjährige Veröffentlichung ab [1].

Keywords

Cabin Design; Automated Manufacturing; Digital Process; Last Minute Customization; Process chain; Hatrack

NOMENKLATUR

ACIS	Airbus Cabin Integration Specification
AMR	autonomous mobile robot
CAD	Computer Aided Design
CPACS	Common Parametric Aircraft Configuration Schema
FUGA	Fuselage Geometry Assembler
JSON	JavaScript Object Notation
LBR	Leichtbauroboter
MES	Manufacturing Execution Systems
OPC UA	Machine-to-Machine-Kommunikationsprotokoll für die industrielle Automatisierung
REST	Representational State Transfer, Softwarearchitekturparadigma
TCP	Tool Center Point

1. EINFÜHRUNG

Airbus Cabin basiert auf einer modularen Plug-and-Play-Architektur, bei der jedes Innenraumelement – Sitze, Bordküchen, Toiletten, Gepäckfächer und Unterhaltungsgeräte – als eigenständiges Modul mit standardisierten mechanischen und elektrischen Schnittstellen hergestellt wird. Diese Modularität ermöglicht es, eine komplett

Kabine innerhalb weniger Wochen auf einer einzigen Produktionslinie zu installieren und nach der Auslieferung Umkonfigurationen vorzunehmen (z. B. zusätzliche Sitzreihen hinzufügen oder eine Bordküche austauschen). Bevor ein physisches Teil hergestellt wird, wird das Kabinenlayout als digitaler Zwilling in CATIA/ENOVIA [2] reproduziert. Das virtuelle Modell wird anhand der Flugzeugzelle auf Kollisionen, Gewichtsverteilung und Kabelführung überprüft, sodass Integrationsprobleme frühzeitig erkannt und aus der Fertigungsline entfernt werden können. So entsteht eine einzige zuverlässige Informationsquelle für Konstrukteure, Zulieferer und Fertigungsmitarbeiter. Der digitale Zwilling unterliegt der Airbus Cabin Integration Specification (ACIS), die genaue Befestigungspunkte, Schraubengrößen, Steckertypen und Datenbusprotokolle für jede Komponente definiert.

2. DEFINITION DES USE CASE

In Zusammenarbeit mit der Industrie wurde ein Szenario zur Darstellung und Validierung der durchgängigen Prozesskette erarbeitet. Ausgehend von dem Wunsch einer Fluggesellschaft die Sitzkonfiguration (Abstände, Position der Monmente) zu ändern, müssen die Aufhängepunkte, die sich über den Passagieren befindlichen Hatracks, an den Spannen angepasst werden, was von den Standardlängen der Hatracks (1-3 Spantabstände) abhängig ist. Zunächst soll die Machbarkeit anhand der Designregeln überprüft und beim positiven Ausgang die Positionen der Aufhängepunkte an den Spannen berechnet werden. Die Aufhängepunkte sollen automatisiert markiert werden,

damit ein menschlicher Werker die Bohrungen für die anschließende Montage setzen kann. Um die Punkte zu markieren wird ein AMR mit einem UR10-Roboter und einer Drehspindel zur Markierung eingesetzt. Ausgehend von den berechneten Punkten wird für die Teilnehmer des Prozesses ein Ablaufplan mit den errechneten Pfaden des AMRs und des Roboters erstellt. Dieser wird simuliert und auf Zugänglichkeit und Kollisionsfreiheit überprüft. Bei positivem Ergebnis wird der Plan an das System AMR-UR10-Endeffektor geschickt und umgesetzt. Um die Aufhängeposition exakt zu markieren, orientiert sich der Endeffektor optisch an der Umgebung und korrigiert die Ungenauigkeiten des AMR. Während das System weitere Punkte markiert, kann der menschliche Werker die markierten Punkte für die Aufnahme der Hatracks bohren. Mit diesem Szenario soll beispielhaft die Durchgängigkeit der Daten vom Design zur Montage demonstriert werden und helfen notwendige Werkzeuge, Schnittstellen und Prozessarchitektur in einem fertigungsnahen Umfeld der Luftfahrtindustrie zu definieren.

3. PROZESSABLAUF

Das Szenario wurde in mehrere Schritte unterteilt und wird im Bild 1 inklusive der etablierten Datenströme (Schnittstellen) zwischen den Prozessen dargestellt. Der Konfigurator erlaubt die Positionierung von Hatracks mit Längen von einem und drei Spantabständen sowie eines Monuments mit Länge eines Spantabstands zwischen den elf Spanten des Mock-Ups. Die Daten aus dem Konfigurator werden genutzt um mit Hilfe des am DLR in Hamburg entwickelten Tools FUGA den Kabinenentwurf anhand der CPACS-Datensatzes (Datenbank) und der innerhalb von FUGA beschriebenen Regeln zu validieren. Zusätzlich, können mit Hilfe von FUGA die Positionen (und damit die Aufhängepunkte) der Hatracks ausgeleitet werden. Das Ergebnis wird benutzt werden um ein Produkt-Prozess-Ressourcen Datenmodell für die Prozessplanung der einzelnen Prozessschritte für die Markierung der Aufhängepunkte aufzubauen. Dies wird mit Hilfe des Modellierungswerkzeugs Cameo Systems Modeler in der vorgegebenen Fertigungsumgebung (hier AMR-Roboter-Endeffektor innerhalb des Mock-Ups) durchgeführt. Die Positionen und die Modelle der Umgebung werden benutzt um Pfade für das AMR und den Roboter mit Hilfe von Klampt¹ zu generieren und in 3dExperience zu simulieren bis kollisionsfreie Pfade gefunden werden. Das Ergebnis wird zum MES des ausführenden Systems geleitet. Dieses bewegt zunächst den AMR in die Nähe des Zielspantes, bringt den Endeffektor montiert auf dem UR10 in die Zielposition mit dem Blickfeld der Kamera auf dem Spant. Die Kamera wird ausgelöst, das 3D-Kamerabild mit den CAD-Daten abgeglichen und eine Korrekturtransformation berechnet. Die Drehspindel wird auf die korrigierte Position bewegt, eingeschaltet und die Bohrmarkierung für den menschlichen Werker gesetzt.

¹<https://github.com/krishauer/Klampt>

4. WECHSELWIRKUNGEN IM PROZESSABLAUF

Für ein tiefergehendes Verständnis der fertigungsnahen Prozessschritte des Gesamtprozesses werden hier die wichtigsten Wechselwirkungen zwischen den digitalen Prozessschritten und den nachfolgenden Prozessgrößen bzw. Prozessschritten dargestellt.

Dafür wurde das Bild 1 aus [1] erweitert und das Ergebnis auf Bild 2 dargestellt. Die ausgegraute Elemente werden hier nicht betrachtet.

4.1. CAD

Basis für die Umsetzung der Prozesskette war das CAD-System CATIA. Unter Verwendung der parametrischen Modellbeschreibung wurden die verschiedenen Baugruppen und Einzelteile des Versuchsaufbaus modelliert. Dabei wurden die Vorgaben des digitalen Bereiches der Prozesskette (s. a. Bild 2, links) in das Modell eingearbeitet und weitere geometrische Größen ermittelt.

4.2. Kabinenentwurf (FUGA)

Der Kabinenentwurf wird mit einer am DLR in Hamburg entwickelte Rumpf- und Kabinenentwurfsssoftware mit dem Namen FUGA („Fuselage Geometry Assembler“) vorgenommen [3, 4].

Aus dem Kabinenentwurf werden die kundenspezifischen Anforderungen sowie physikalische Grundsätze oder empirische Informationen (z. B. aus CPACS) verarbeitet und das Ergebnis in das CAD-System übernommen. In unserem Fall sind hier Spantradius, Spantquerschnitt, Anzahl der Spante, Spantabstand und die Koordinaten der Anbohrpunkte für die Hattrackbefestigung als wichtigste geometrischen Kenngrößen zu nennen.

Diese Daten waren während unserer Versuche konstant und konnten sowohl durch einfache Textdateien als auch durch das STEP-Austauschformat schnell übertragen werden und ausgewertet werden. Die sich in der Praxis entsprechend der Konfiguration ändernden Koordinaten der Anbohrpunkte müssen nach der Datenreduzierung und Transformation einer JSON-Datei in das CAD-System eingelesen werden. Eine solche Transformation ist beispielhaft in TAB 1 dargestellt.

Diese Transformation kann schnell mittels eines Pro-

x: 0.533,	→	533,0
y: 1.246,	→	1246,0
z: 1.787	→	1787,0

TAB 1. Beispieltransformation

gramms erfolgen. Der Import dieser Punktkoordinaten in das zu verwendende CAD-System ist für die weitere Entwicklung der Fertigungskomponenten (s. a. Kapitel 5) unabdingbar.

5. FERTIGUNG

Auf Basis der durch die digitalen Prozessschritte (Bild 2, links) enthaltenen Daten werden dann die Komponenten entwickelt, mit welchen die praktische Umsetzung der An-



BILD 1. Prozessablauf für das definierte Szenario.

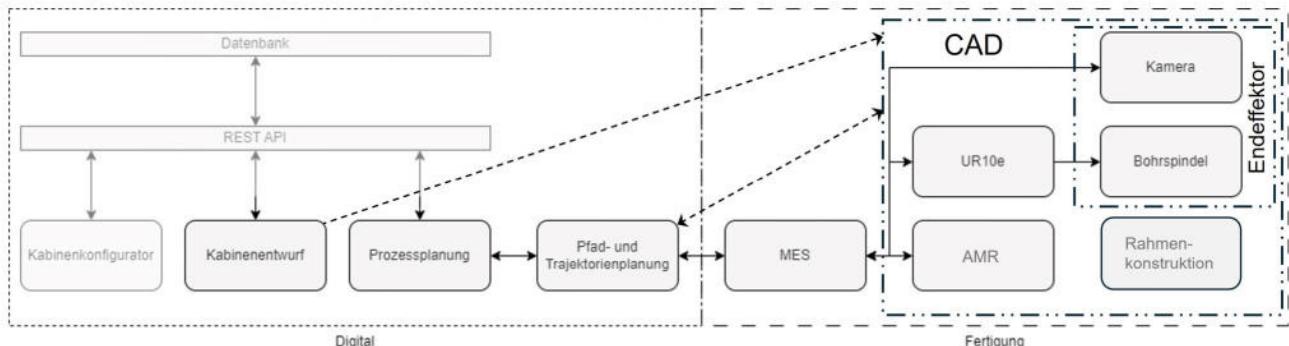


BILD 2. Darstellung der Prozesswechselwirkungen basierend auf [1]

bohrvorgänge auf Basis der Planungsdaten demonstriert werden können.

5.1. Rahmenkonstruktion

Die Entwicklung und Realisierung der Baugruppe Rahmenkonstruktion dient dazu die vom Typ her eher statischen Daten, welche den betrachteten Flugzeugtyp beschreiben, als Modell für die Prozessdemonstration bereitzustellen.

Eine Übersicht über die Konstruktion ist auf Bild 3 zu sehen.

Die wichtigsten technischen Eigenschaften der Bestandteile werden nachfolgend kurz beschrieben.

Eine Rahmenkonstruktion bestehend aus genormten Aluminiumprofilen, speziell gefertigten Aluminiumprofilen (Spantgeometrie), genormten sowie speziell für diesen Zweck konstruierten Befestigungselementen bilden die prozessrelevanten Einzelteile der Rumpfgeometrie als Shopfloor-Modell ab.

Im Rahmen des einmaligen finalen Aufbaus ist dabei darauf zu achten, dass die gesamte Rahmenkonstruktion exakt eingemessen und unverrückbar auf dem Boden befestigt ist.

Dabei geht es sowohl darum die Spante möglichst in der richtigen Position und Orientierung im Rahmen der Baugruppe Rahmenkonstruktion zu befestigen, als auch darum die Spante so zu befestigen, dass der spätere Bearbeitungsvorgang und die daraus resultierenden Bearbeitungskräfte keine nennenswerte Deformation der Spante zur Folge hat.

Eine möglichst exakte Befestigung und Ausrichtung, basierend auf dem CAD-Modell, ist hier auch von entscheidender Bedeutung für den späteren Anbohrvorgang.

Denn die Pfad- und Trajektorienplanung für den AMR und den darauf angebrachten Leichtbauroboter (s. a. Bild 3, mittig) basieren auf dem CAD-Modell.

An der Markerplatte, dargestellt auf Bild 4, findet die eigentliche Bearbeitung in Form des Anbohrvorganges statt.

Die Anzahl und Reihenfolge der Bearbeitungsvorgänge resultieren zum einen aus dem Kabinenentwurf und zum anderen aus der Pfad- und Trajektorienplanung sowie den Daten des MES (Kapitel 5.4).

Der eigentliche Bearbeitungsvorgang wird hier von einem NC-Anbohrer durchgeführt. Dabei wird hier mittels dieses Anbohrers eine kegelige Markierung für die später einzubringende Bohrung eingearbeitet.

Für diesen Fertigungsvorgang wurden in diesem Modell austauschbare Markerplatten vorgesehen, um bei Bedarf mehrere Anbohrversuche an einer bestimmten Stelle durchführen zu können, ohne das nachfolgende Anbohrversuche durch vorhergehende Versuche beeinflusst werden.

Die Markerplatten selbst werden an den Spantprofilen in definierter Position und Orientierung befestigt.

Der hier einheitliche Spantprofilquerschnitt ist auf Bild 5 dargestellt.

Das Spantprofil selbst ist das Ergebnis der Berechnungen und Entscheidungen aus dem Kabinenentwurf.

Das Spantprofil ist neben den Anbohrpunkten eine der wichtigsten Größen für die Pfad- und Projektplanung sowie für die konstruktive Auslegung des Endeffektors.

5.2. Endeffektor

Der Endeffektor ist Bestandteil der mobilen Baugruppe bestehend aus AMR, Leichtbauroboter Universal Robots UR10e und Endeffektor (s. a. Bild 3, mittig).

Die Hauptfunktion des Endeffektors ist es, die Anbohrungen entsprechend der vorgegebenen Bohrpunktkoordinaten in die Markerplatten einzuarbeiten.

Zur Sicherstellung einer möglichst genauen Position und Orientierung der Anbohrung wird die Ausrichtung des Endeffektors mittels einer Kamera vorgenommen. Eine detailliertere Beschreibung der Funktionsweise des Systems findet man in Kapitel 5.6.

Die wichtigsten Komponenten des Endeffektors werden auf nachfolgendem Bild 6 dargestellt.

Der gesamte Endeffektor wird mit einem Leichtbauroboter [6] gehandhabt. Leichtbauroboter (LBR) und Endeffek-

tor werden mit Hilfe eines Autonomen Mobilen Roboters (AMR) in der Halle bewegt. Durch die begrenzten Energiespeichermöglichkeiten des AMR und die räumlichen Limitationen wurde ein Leichtbauroboter ausgewählt. Diese Roboterauswahl wiederum hatte eine Gewichtsbeschränkung des Endeffektors und eine geringe Robotersteifigkeit zur Folge. Die Tragfähigkeitslimitierung konnte durch eine Leichtbauweise des Endeffektors ausreichend ausgeglichen werden.

Die mechanische Basis des Endeffektors bildet ein Blechbiegeteil, welches die erforderlichen Komponenten aufnimmt und deren Position und Orientierung zueinander festlegt und dass der Kraftübertragung dient. Weiterhin wird hiermit eine lösbare aber definierte Verbindung zum Leichtbauroboter hergestellt, wodurch eine reproduzierbare, zielgerichtete Handhabung des Endeffektors durch den LBR möglich wird.

Bei der Kamera handelt es sich um eine 3D-Kamera IDS Ensenso N36-606-16-BL [7]. Mit dieser Kamera ist es möglich Daten aufzunehmen, die ein dreidimensionales Bild beschreiben. Diese werden dann zur Verarbeitung weitergeleitet. Durch den geringen Energiebedarf der 3D-Kamera kann sowohl der Datentransfer als auch die Stromversorgung über das Netzwerkkabel erfolgen (POE).

Wie schon weiter oben angemerkt war es das Ziel eine hinreichend genaue Markierung auf dem Werkstück aufzubringen. In diesem Fall soll die Position für ein Bohrloch markiert werden.

Nach dem Test verschiedener Methoden wurde ein NC-Anbohrer mit 6mm Außendurchmesser als Werkzeug ausgewählt [8]. Ein solcher Werkzeugtyp wird auch in Werkzeugmaschinen zur Bohrlochvorbereitung eingesetzt [9]. Dieser Werkzeugtyp ist bestens geeignet für das Zentrieren und Ansenken von Bohrungen. Er zeichnet sich durch eine stabile Geometrie sowie durch eine hohe Standzeit aus. Weiterhin geht das Ansenken weit über ein reines graphisches Markieren der Bohrung hinaus und erleichtert den nachfolgenden Bohrvorgang.

Als Antrieb für den Bohrvorgang wird hier ein luftgekühlter, bürstenloser Frä斯-Spindelmotor [10] eingesetzt. Dieser

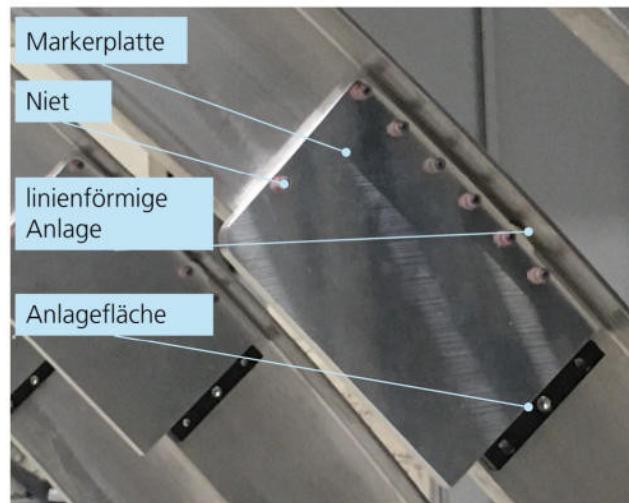


BILD 4. Markerplatte

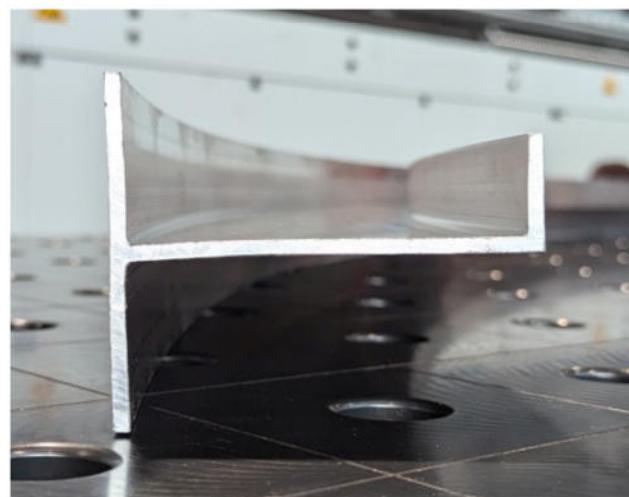


BILD 5. Spantprofil

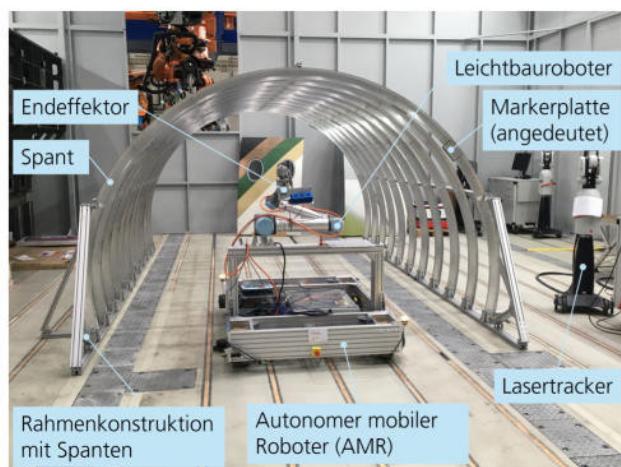


BILD 3. Übersicht über die Rahmenkonstruktion und das AMR [5] samt Aufbauten

luftgekühlte Gleichstrommotor wird durch Gleichstrom bei einer Spannung von 48V vom AMR versorgt.

Als Werkzeugaufnahme für den NC-Anbohrer kommt eine ER11 Spannzange zum Einsatz. Der Spindelmotor, die Spannzange und die Klemme (Motorklemmschelle) des Bürstenmotors wiegen circa 2kg. Durch den geringen Bauraumbedarf, das geringe Gewicht der genannten Teile sowie die Tatsache, dass durch den AMR die Stromversorgung sichergestellt werden kann, wurde diese Lösung ausgewählt und realisiert.

5.3. Autonomous Mobile Robot (AMR)

Der in [1] vorgestellte mobile Roboter (AMR) wurde für die Demonstration ausgiebig getestet und weiter verbessert. Mit Hilfe der vier Mecanum Räder kann die mobile Plattform sich frei in jede Raum-Richtung bewegen. Zur Lokalisierung werden zwei Sick nanoScan Laserscanner eingesetzt. Die Software „LiDAR-LOC“ von Sick berechnet aus den Scan-Bildern beider Scanner und einer zuvor aufgezeichneten Karte die aktuelle Position und Drehung des AMR. Hierbei ist zu beachten, dass die Genauigkeit der Positionierung nur bei etwa ± 5 mm liegt. Außerdem sind ausreichend markante Merkmale in der Umgebung

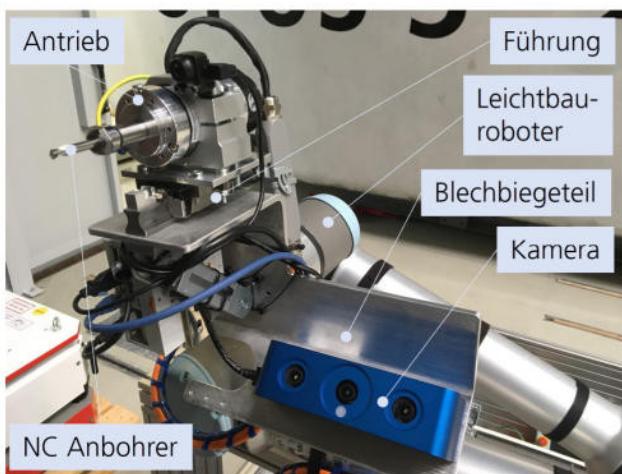


BILD 6. UR10e Leichtbauroboter und zentrale Bestandteile des Endeffektors

erforderlich. Die Spanten des Mock-Ups alleine sind z.B. nicht ausreichend (da dies im 2D-Scannerbild lediglich einzelne Punkte mit großer Symmetrie sind).

Zur Kommunikation mit der übergeordneten Steuerung (siehe auch Abschnitt 5.4) steht eine OPC UA Schnittstelle im WiFi-System des AMR zur Verfügung. Über diese Schnittstelle werden folgende Informationen ausgetauscht:

- Richtung zu MES
 - Aktuelle gemessene Position und Orientierung des AMR auf der Karte (mit Unsicherheit behaftet)
 - Betriebsbereitschaft (u.a. Automatik-Betrieb aktiviert, Fahrerfreigabe vorhanden)
 - Ergebnis des letzten Fahrbefehls (erfolgreich, abgebrochen)
 - Fahrbefehl aktiv
- Richtung von MES
 - Ziel-Position und -Orientierung
 - Maximale Geschwindigkeit
 - Bewegung starten

Um eine Positionierung des AMR zu erreichen, muss das MES-System die gewünschte Zielposition und -orientierung übermitteln und eine steigende Flanke in das Feld zum Starten der Bewegung schreiben. Die interne Steuerung des AMR fährt dann auf direktem Weg zur Zielposition. Die Trajektorie wird hierbei unter Einbeziehung der aktuellen Messdaten des Lokalisierungssystems geregelt. Die Planung der eigentlichen Bahn muss vom MES-System durchgeführt werden, das AMR führt immer Punkt-zu-Punkt Bewegungen aus.

5.4. Manufacturing execution system (MES)

Um die geplanten Prozessschritte aus Abschnitt 3 auf dem Shopfloor abzuarbeiten wird ein Ausführungssystem (MES) benötigt. Das MES erhält die Prozessschrittdefinitionen als JSON-Datei, netzwerkfähige Erweiterungen für mehr Interaktivität werden derzeit untersucht. Unabhängig vom Kommunikationsweg muss sich das MES eine Datenstruktur aus den Prozessschritten aufbauen und die jeweiligen Parameter im Speicher ablegen. Hierfür wurde eine Baumstruktur aus Varianten gewählt,

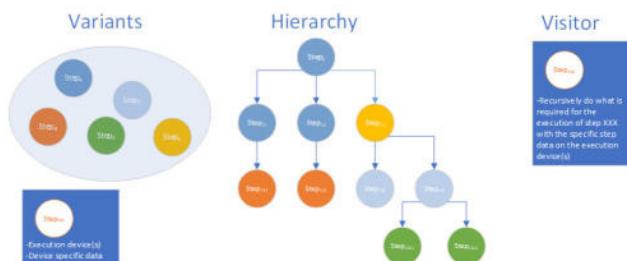


BILD 7. Gesamtprozess aus Prozessschritten generiert aus der JSON-Beschreibung, dargestellt durch eine Baumstruktur aus Varianten. Der Visitor rechts dient zur Abarbeitung der Einzelschritte.

durch die verschiedenste Abläufe flexibel realisierbar sind. Bild 7 illustriert das Vorgehen.

Varianten der Prozessschritte sind hierbei Bewegungsbausteine für AMR und UR, der Messschritt des 3D-Kamerasystems sowie der Bohrschritt. Diese werden aus dem JSON hierarchisch im Arbeitsspeicher aufgebaut und können im Prozess mittels eines sog. Visitor-Patterns abgearbeitet werden (vgl. [11]). Die Verwendung des Visitor-Patterns sorgt dabei für eine strikte Trennung und die typsichere Ausführung der Prozessschritte, was für eine spätere Erweiterbarkeit förderlich sein kann, jedoch einen gewissen Mehraufwand für die benötigte Kommunikation zwischen den Prozessschritten erfordert. Beispielsweise hängen die Bewegungsschritte für den UR davon ab, wo der AMR steht, und diese Information muss nun extra in einem Datenaustauschobjekt vom AMR-Schritt für den UR-Schritt bereitgestellt werden, da die beiden Prozessschritte absolut nichts voneinander wissen. Ähnliches gilt für die Fehlerbehandlung, die in dem Projekt nur rudimentär angegangen werden konnte (Abbruch der Folgeschritte und manuelles Freifahren der Anlage).

Zur Ausführung der Prozessschritte muss der Visitor die Schritte in elementare Befehle für die einzelnen Aggregate herunterbrechen und diese an die Aggregate verteilen. Zum Beispiel muss für den Bohrschritt eine Vorposition angefahren, der Bohrer aktiviert, die Bohrposition züglich der Bohrtiefe angefahren und der Bohrer wieder zurückgezogen werden. Um dies zu tun hat der Prozessschritt ein Ausführungsgerät, das mit der tatsächlichen Hardware kommuniziert, ohne dass der Visitor wissen muss wie dies genau umgesetzt ist. Eine Übersicht über das gesamte MES ist in Bild 8 gegeben.

Auf jedem Aggregat läuft ein Server (unterste Ebene). Im konkreten Fall sind es zwei verschiedene Servertypen, die in der zweiten Ebene von unten je ein Client-Gegenstück besitzen, das eben genannte Ausführungsgerät. Dies bildet eine Abstraktionsschicht für die darunterliegenden Hardware, mit der die Ausführungsschicht des MES in der nächsthöheren Ebene kommunizieren kann, ohne sich mit Details abgeben zu müssen. In der obersten Schicht befindet sich noch eine grafische Benutzeroberfläche (GUI), über die der aktuelle Job geladen werden kann sowie Prozessschritte ausgewählt und gestartet werden können sowie ein Server, der diese Funktionen für eine eventuelle spätere Bedienung über einen Leitstand verfügbar macht.

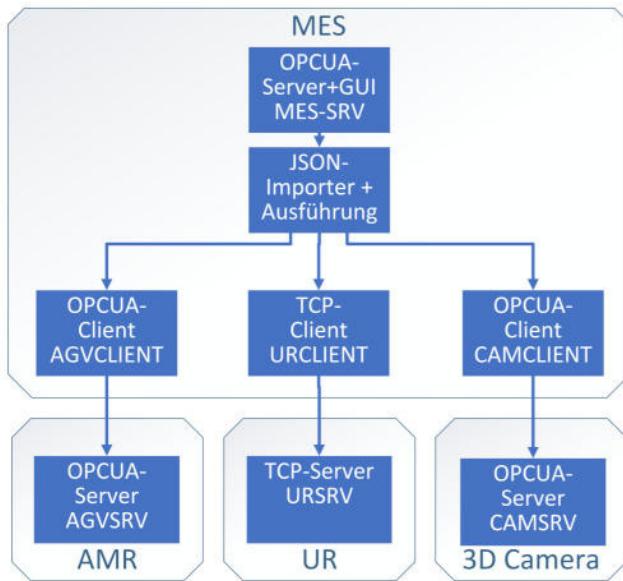


BILD 8. Struktur des MES mit Komponenten-Clients (AMR, UR und 3D-Kamera) und Server-Gegenstücken im MES sowie der Ausführungsschicht mit übergeordnetem Server

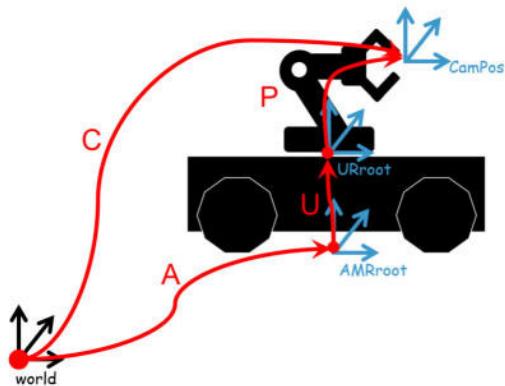


BILD 9. Verwendete Koordinatensysteme und deren Umrechnung ineinander

5.5. Koordinatentransformationen

Dem MES fällt auch die Aufgabe zu, sämtliche Koordinatensysteme der Teilnehmer ineinander zu überführen. Hintergrund ist, dass die Ziel-Koordinaten aus dem CAD-Bestand kommen, AMR und UR aber je eigene Koordinatensysteme haben, wobei beim UR zusätzlich auch noch das verwendete Tool zu berücksichtigen ist.

Bild 9 gibt einen Überblick der verwendeten Koordinatensysteme für den Fall, dass das Kamera-Tool ausgewählt ist und die Kamera auf ihre CAD-Messposition $C = \text{CamPos}_{\text{World}}$ gefahren werden soll.

Zunächst muss der AMR in eine Position verbracht werden, in der dies auch möglich ist. Diese Position hat das MES aus der JSON-Beschreibung extrahiert und sie kann direkt als Sollposition verwendet werden. Da der AMR wie im Abschnitt 5.3 beschrieben nur eine Positioniergenauigkeit von $\approx \pm 5 \text{ mm}$ hat, muss der UR dies kompensieren. Der Bezugspunkt des AMR ist das Lot vom Mittelpunkt der Draufsicht des Fahrzeugs auf den Hallenboden (AMRroot), die aktuelle Position des AMR

mit Orientierung sei A. Der Bezugspunkt des UR (URroot) liegt im Roboterfuß und ist bezüglich der AMRroot um U versetzt. Die Position des UR in der URroot sei P. Der UR soll nun so positioniert werden, dass sein Kamera-Tool auf die gewünschte Kameraposition zeigt. Dann gilt gemäß Bild 9:

$$A \cdot U \cdot P = C$$

bzw. nach multiplizieren der Inversen von A und U von links:

$$P = U^{-1} \cdot A^{-1} \cdot C$$

Der Einfachheit halber wurde hier das World-Koordinatensystem des AMR auf das CAD-World gelegt, ansonsten hätten wir noch eine Transformation in der Kette, was je nach Anwendungsfall sinnvoll sein kann oder auch nicht. Da die Position des AMR mit dem Messfehler der LIDAR-Sensoren behaftet ist benötigt man noch die Korrekturwerte δ aus Abschnitt 5.6; diese liegen, nachdem wir die Kamera im entsprechenden Tool-System eingemessen haben, für den Kamera-TCP vor und werden ebenfalls vom MES verrechnet und ins Bohr-Koordinatensystem transformiert, da das MES die hierfür benötigten TCPs ($T_{\text{cam}}, T_{\text{drill}}$) kennt. Es gilt:

$$\delta_{\text{drill}} = T_{\text{drill}}^{-1} \cdot T_{\text{cam}} \cdot \delta_{\text{cam}} \cdot T_{\text{cam}}^{-1} \cdot T_{\text{drill}}$$

Abschließend wird die World-Bohrsollposition D ins UR-Fußkoordinatensystem umgerechnet, um δ korrigiert und mit dem Bohr-Tool angefahren mittels:

$$P_{\text{drill}} = U^{-1} \cdot A^{-1} \cdot D \cdot \delta_{\text{drill}}$$

5.6. Lokale Orientierung

Das System zur lokalen Orientierung besteht aus der 3D-Kamera, die eine Punktwolke generiert. Eine Hand-Eye-Calibration stellt sicher, dass die Punktwolke in einem vom CAD vorgegebenen Koordinatensystem vorliegt. Es hat sich herausgestellt, dass die Ausgabe der Punktwolke im Flanschkoordinatensystem nicht hilfreich für die Berechnung der Transformationskette ist, da dem MES nur das Bohr- und Kamerakoordinatensystem bekannt ist. Dieses Kamerakoordinatensystem ist nicht mit dem default-Koordinatensystem der 3D-Kamera zu verwechseln, das sich innerhalb des 3D-Kameragehäuses befindet. Das Kamerakoordinatensystem aus dem CAD liegt außerhalb im vorgegebenen Arbeitsabstand der Kamera. Der Ursprung des Kamerakoordinatensystems liegt damit während der Messung auf der Markerplatte. Deshalb wird nun das Kamerakoordinatensystem aus dem CAD als TCP auf der Robotersteuerung hinterlegt.

Anschließend werden verschiedene Positionen angefahren bei denen die Kamera aus unterschiedlichen Richtungen auf ein Schachbrett-Muster blickt. Anhand der Positionen und Kameraaufnahmen wird abschließend die Hand-Eye-Calibration berechnet. Jede aufgenommene Punktwolke liegt nun im Kamerakoordinatensystem aus dem CAD vor. Somit ist das MES in der Lage die Transformationen zu verketten.

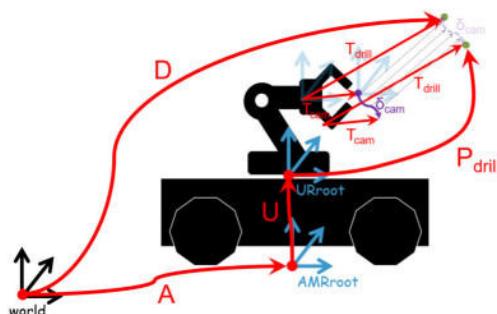


BILD 10. Beaufschlagung der ungenau angefahrenen Bohrposition mit dem Korrekturwert der 3D-Kamera

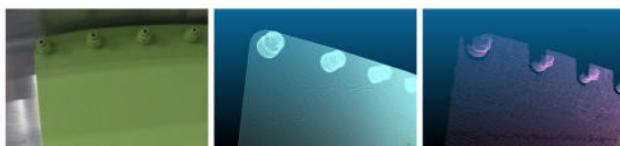


BILD 11. Links: reale Platte mit Nieten, Mitte: CAD-Punktwolke, Rechts: 3D-Kamera-Punktwolke

Die 3D-Kamera berechnet anhand einer aktuellen Aufnahme die Korrektur zur CAD-Vorgabe. Damit wird die aktuelle Bohrposition korrigiert, die aufgrund von fehlerbehafteter Positionierung des AMR und des Roboters nicht ohne Überprüfung und Berichtigung angefahren werden sollte. Die 3D-Kamera stellt dem MES eine Korrekturtransformation δ_{cam} (siehe Bild 10) via OPCUA zur Verfügung.

Anhand der Bohrposition im Weltkoordinatensystem und dem Wissen über die Position des AMR sowie der Montageposition des Roboters lässt sich die Bohrposition in Bezug zum Roboterkoordinatensystem berechnen. Die Bohrposition im Roboterkoordinatensystem wird um den Korrekturwert verschoben, wodurch der Roboter an die richtige Position bewegt werden kann. Da sich die Korrekturtransformation auf das Kamerakoordinatensystem bezieht, ist eine Reihe von Koordinatentransformationen nötig um die Bohrposition korrekt mit dem Korrekturwert zu beaufschlagen (siehe Bild 10).

Die Korrekturtransformation wird vorab mittels mehrstufiger Pipeline zwischen CAD-Punktwolke und aufgenommener Punktwolke berechnet. Die Pipeline ist in C++ implementiert und verwendet die Point Cloud Library (PCL). Im Vergleich zu [1] konnte die Berechnungszeit von ca. 8 Sekunden für die Korrekturtransformation auf ca. 1 Sekunde reduziert werden. Ausschlaggebend ist eine geeignete Wahl der Blattgröße bei der Voxelisierung der Punktwolke um die Normalenberechnung möglichst kurz zu halten. Zusätzlich wurde ein rechenintensiver Schritt in der Pipeline weggelassen und durch einen anderen effizienteren Schritt am Ende der Pipeline ersetzt.

Genauer gesagt wird die Position der einzelnen Fastener erst nur grob abgeschätzt, und nicht mehr genau berechnet. Dadurch lässt sich ein erster Fit zwischen aufgenommener Punktwolke und CAD-Punktwolke erreichen, der die Grundlage für den abschließenden Schritt mittels Iterative-Closest-Point bildet. Für Iterative-Closest-Point wird eine CAD-Punktwolke benötigt, die mittels CAD-Programm aus der Oberfläche des Bauteils erstellt und in

das Kamerakoordinatensystem exportiert wird. Bild 11 zeigt die generierte CAD-Punktwolke in der Mitte.

Erste Genauigkeitsuntersuchungen haben gezeigt, dass der Roboter hinreichend genau ist, da die lokale Neuorientierung nur in einem kleinen Bereich um die aktuelle Position stattfindet. Beim AMR gilt es zu berücksichtigen, dass dieses während der Messung und des Markievorgangs die Position nicht ändern darf. Wenn das AMR an der Messposition angekommen ist, wird die Positionsregelung fixiert, d.h. die Motoren werden so geregelt, dass sie nicht mehr drehen dürfen und die Laserscanner werden ignoriert. Damit werden minimale Bewegungen des AMR verhindert, die sich aus der Streuung der Laserscanner ergeben und die die Koordinatensysteme sonst bei den statischen Transformationsberechnungen verschieben würden.

Ein weiterer, nicht zu verhindernder Effekt, liegt in der Streuung der Punktwolken. Trotz identischer Kameraposition können die Punkte der Punktwolken variieren und damit u.a. die Iterative-Closest-Point Ausgabe ändern. Im Folgeprojekt werden umfassende Genauigkeitsuntersuchungen durchgeführt um eine valide Aussage zur Bohrgenauigkeit treffen zu können und die größten Einflussfaktoren zu identifizieren.

6. DEMONSTRATION

Zum Projektabschluss wurde eine vollautomatische Markierung an mehreren Bohrpositionen demonstriert. Für jeweils vier unterschiedliche Bohrpositionen je Spant wurden hierzu entsprechende Positionsdaten und Trajektorien berechnet und in einer JSON-Datei bereitgestellt. Mit Hilfe des MES wurde diese Datei verarbeitet und auf den einzelnen Systemen ausgeführt.

Zur Erhöhung der Reichweite ist der UR10e Roboter außerhalb auf der mobilen Plattform montiert. Dies führt allerdings dazu, dass das AMR je nach Seite um 180° rotiert werden muss. Auf Grund der Platzverhältnisse ist diese Drehung nur genau in der Mitte der Kabine möglich. Daher muss das MES darauf achten, beim Wechsel der Seiten das AMR zunächst mittig zu platzieren, und anschließend eine Drehung um die eigene Achse zu veranlassen, bevor die nächsten Bohrpositionen angefahren werden.

Die Navigation des AMR innerhalb des Mock-Ups funktioniert mit Hilfe einer vorher sorgfältig eingemessenen Karte gut, so dass keine Kollisionen vorkamen. Allerdings ist es notwendig, außer den Spanten weitere Merkmale der Umgebung (wie z.B. die Wand der Halle) in der Karte zu erfassen, da diese alleine zu klein und gleichförmig für eine Lokalisierung sind. Mit Hilfe der Laser-Scanner kann das AMR auf eine Genauigkeit von ungefähr ± 5 mm positioniert werden. Diese Genauigkeit ist ausreichend, um mit dem UR10e eine sichere Position zur Bildaufnahme zu erreichen.

Mittels MES wurden sämtliche Teilprozesse zu einem Gesamtprozess orchestriert und ermöglichen so die reibungslose Demonstration am Mock-Up. 3D-Kamera und UR10e haben für sich betrachtet ausreichende Genauigkeit $< \pm 0.1$ mm in ersten Vorversuchen gezeigt.

Viele Einflussfaktoren z. B. Verschleiß, Elastizitäten, Reibung entlang der Ungenauigkeitskette können eine Rolle spielen. Einige davon wurden bereits eliminiert, wie die Motorregelungsabschaltung während der Messung und Anbohrung. Insgesamt ist die Genauigkeit der Anbohrpositionen vielversprechend, aber noch im Folgeprojekt zu validieren und zu quantifizieren.

7. ZUSAMMENFASSUNG

Den Auftakt bildete die Definition eines gängigen Use Cases: Konfiguration der Hatrack-Positionen und exaktes Setzen der Bohrungen für die Aufhängungen im Shopfloor.

Um eine reibungslose automatisierte Fertigung zu verwirklichen, wurde der Prozess in der digitalen Welt aufbereitet. Die DLR-Eigenentwicklung FUGA verarbeitet und validiert die kundenspezifischen Anforderungen des Kabinenentwurfs aus dem Konfigurator und übernimmt diese ins CAD. Die anschließende Prozess- und Pfadplanung übernehmen Softwaretools wie Cameo, Klampt und 3dExperience.

Zur Übermittlung relevanter Daten an das MES wurde eine JSON-Prozessbeschreibungsdatei als Schnittstelle verwendet. Der Demonstratoraufbau bildet die relevanten Teile einer Flugzeugschale (Spante und Nietstellen) anhand der Ausgabe von FUGA nach. Zur Bohrlohrmarkierung wurde ein Endeffektor, bestückt mit 3D-Kamera und Anbohrer, realisiert.

In der realen Umsetzung übernimmt das MES mittels JSON-Prozessbeschreibungsdatei das Abfahren der Pfade aus der Pfadplanung, die Robotersteuerung und die 3D-Kamera-Triggerung sowie die Koordinatentransformationsberechnungen um die einzelnen Komponenten - AMR, UR und Kamera - korrekt positionieren zu können. Dazu wurde eine Software designed und implementiert und ermöglicht mittels Baumstruktur aus Varianten eine flexible Realisierung unterschiedlichster Prozessschritt-kombinationen. Der Prozessschritt zur lokalen Orientierung wird mittels 3D-Kamera ausgeführt.

Anhand realer 3D-Daten werden Merkmale am Spant wie Nietpositionen mit dem CAD-Modell abgeglichen. Daraus wird innerhalb einer Sekunde eine Korrekturtransformation berechnet, die dem MES zur Verfügung gestellt wird und eine präzise Bohrlochmarkierung ermöglicht.

Bei der Umsetzung der oben beschriebenen Prozesskette (s. a. Bild 1 und Bild 2) zeigte sich, dass nach einem anfänglichen Datentransfer zwischen Kabinenentwurf und CAD-System CATIA eine parallele Entwicklung von Konstruktion und Entwicklung und den dem Kabinenentwurf nachfolgenden Prozessschritte erfolgen konnte. Der weitere Austausch spielte sich überwiegend als kurzfristiger Datenaustausch zwischen Pfad- und Trajektorienplanung (s. a. Bild 2) und dem CAD-System ab. Die mittels CAD entwickelten und somit teilweise neu entstandenen zusätzlichen Baugruppen wie AMR und Endeffektor sowie der LBR als Kaufteil konnten nach gemeinsamer Definition einiger Festlegungen zum Datenaustausch schnell und unkompliziert übermittelt werden. Auch Zusatzinformationen für die Pfad- und Trajektorienplanung wie z. B. die Koordinatensysteme (s. a. Kapitel 5.5) konnten in Verbin-

dung mit der CAD-Geometrie schnell gemeinsam definiert und übertragen werden. Grundsätzlich kann gesagt werden, dass der Aufwand für Datenaustausch zwischen CAD und Pfad- und Trajektorienplanung mit fortschreitender Umsetzung der Baugruppen im Shopfloor abnimmt. Die in der Prozesskette definierten Schnittstellen und Spezialisierung der Aufgabenbereiche wirkten sich schon bei der Erstellung des Testsystem positiv auf den Zeitbedarf der Entwicklungsarbeiten aus. Bei unveränderten Gegebenheiten bezüglich der Rumpfgeometrie können die Modifikationen der Position der Anbohrpunkte dann entsprechend kurzfristig vorgenommen werden.

8. VORSCHAU

Die Demonstration zeigte die Vorteile und das große Potential der durchgängigen Prozesskette, speziell im Fall der Last-Minute-Customisation. Die entwickelten Lösungen und die Datenarchitektur können für weitere Szenarios eingesetzt werden. Dies wird in dem Nachfolgeprojekt DiCAdemoRe, das im Januar 2026 gestartet wird, weiter untersucht. Das Szenario wird zusammen mit der Industrie anhand der aktuellen Bedarfe festgelegt und wird den Prozess um flexible Fertigung samt der Logistikkette auf Fabriklevel hochskalieren.

9. DANKSAGUNG

Die Autoren bedanken sich bedanken sich bei allen Mitgliedern des Projektteams sowie bei allen internen und externen Unterstützern. Wir bedanken uns weiterhin bei dem Deutschen Bundestag für die Bereitstellung eines Budgets, welches es dem DLR ermöglichte das Projekt DiCaDeMa zu finanzieren.

Literatur

- [1] M. Malecha, V. Srinivasan, Y. Ghanjaoui, P. Satwan, M. Schönheits, M. Mayer, and A. Schuster. Vom Kabinenplaner zur Fertigung - das DLR-Projekt DiCADCeMa. Deutsche Gesellschaft für Luft- und Raumfahrt - Lilienthal-Oberth e.V., Dec. 2024. [DOI:10.25967/630416](https://doi.org/10.25967/630416).
- [2] ENOVIA. aufgerufen am 05.09.2025. <https://4-plm.de/enovia/>.
- [3] Jan-Niclas Walther, Christian Hesse, Jörn Biedermann, and Björn Nagel. Extensible aircraft fuselage model generation for a multidisciplinary, multi-fidelity context. In *33rd Congress of the International Council of the Aeronautical Sciences (ICAS)*, 2022.
- [4] Jan-Niclas Walther. *Knowledge-based engineering to provide aircraft fuselage design details for multidisciplinary and multifidelity analysis model generation*. Phd thesis, Technische Universität Berlin, Berlin, 2024. [DOI:10.14279/depositonce-20898](https://doi.org/10.14279/depositonce-20898).
- [5] Stefan Hesse and Viktorio Malisa, editors. *Taschenbuch Robotik - Montage - Handhabung*. Fachbuchverlag Leipzig im Carl Hanser Verlag, München, 2., neu bearbeitete auflage edition, 2016. ISBN:978-3-446-44549-9 978-3-446-44365-5. OCLC: 932644906.
- [6] UR10e Technische Daten. aufgerufen am 29.08.2025. <https://ml-robotik.de/storage/2024/03/Universal-Robots-UR10e-Datenblatt.pdf>.
- [7] N36-606-16-BL. abgerufen am 29.08.2025. <https://de.ids-imaging.com/ensensoprodukt.html?modelName=N36-606-16-BL>.
- [8] NC-Anbohrer ø6mm HSS-Co 120Grad PROMAT. abgerufen am 29.08.2025. <https://www.b2b-discount.de/nc-anbohrer-durchmesser-6mm-hss-co-120grad-promat-4000860086-4015448477303>.
- [9] Heinz Jung, Werner Schröder, and Jörg-Peter Pahl. *Fachpraxis Metall: mit Arbeitsplanung und CNC-Technik*. Cornelsen, Berlin, 1996. ISBN:978-3-590-82122-4. OCLC: 180533309.
- [10] VEVOR Luftgekühlter Spindelmotor. aufgerufen am 29.08.2025. https://www.vevor.de/spindelmotor-c_1_0130/cnc-0-4kw-brushless-spindle-motor-fraesspindel-schaltnetzteil-pwm-controller-p_010543973069.
- [11] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns - Elements of Reusable Object-Oriented Software*. Addison Wesley, 1995. ISBN: 0-201-63361-2.