

Lösung 1 (Vergleich von Robotersystemen)

Ein kleines/mittleres Unternehmen kann es sich im Normalfall nicht leisten, einen Roboter nur für eine spezielle Aufgabe verwenden zu können. Deshalb ist hier große Flexibilität in der Anwendbarkeit gefragt. Die Anforderungen können also z.B. folgendermaßen aussehen:

- Zu einem solchen Robotersystem gehört nicht nur der Roboter selbst, sondern auch die Steuereinheit, ein Programmierplatz, Förderbänder, und eine Fertigungszelle.
- Das System soll mobil sein, also an verschiedenen Orten der Produktion einsetzbar sein. Dies bedeutet, dass sowohl Roboter als auch Steuereinheit mit vertretbarem Aufwand versetzt werden können müssen.
- Das Robotersystem muss mit wenig Aufwand umprogrammierbar sein, um die verschiedenen Aufgaben bearbeiten zu können.
- Verschiedene Aufgaben wie Montage und Schweißen erfordern Flexibilität bei der Werkzeugwahl. Es muss also der Greifer gegen andere Werkzeuge austauschbar sein.

Folgende Anforderungen sollten deshalb geprüft werden:

Flexibilität: Einsatz des Systems für eine bestimmte Aufgabe oder möglichst gute Anpassbarkeit an verschiedenste Aufgaben.

Stückzahlen: Sollen nur Einzelstücke, kleine Serien oder groß Stückzahlen bearbeitet werden. Dies lässt dann auch Rückschlüsse auf die benötigte Flexibilität des Systems zu.

Einsatzort: Wird das System nur an einem festen Ort eingesetzt oder muss es mobil sein.

Programmierung: Kann die Programmierung intern erfolgen oder ist ein externer Dienstleister nötig? Wie hoch ist der Schulungsaufwand für eigene Programmierung?

Bedienung: Arbeitet das System autonom? Arbeitet es im Zusammenspiel mit einem Bediener (z.B. Einlegen von Werkstücken). Wichtig ist dabei der Schutz des Bedieners. Außerdem müssen zukünftige Benutzer auf dem System geschult werden.

Wartung: Wird das System selbst gewartet, benötigt man einen externen Dienstleister? Wie oft muss das System gewartet werden, wie lange dauert eine Wartung? Wie groß müssen Zuverlässigkeit und Verfügbarkeit sein?

Arbeitszeit: Wie lange darf ein Arbeitszyklus maximal dauern?

Modellierung: Wer modelliert das System, die Werkstücke, den Arbeitsplatz?

Erweiterbarkeit: Muss das System für spätere (bis dato unbekannte Aufgaben) anpassbar sein? Die Erweiterbarkeit betrifft auch

- andere Einsatzorte,
- andere Werkstücke,
- andere Werkzeuge.

Preis: ist ein nicht zu unterschätzender Faktor.

Lösung 2 (Bewertung von Programmierverfahren)

Die Programmierung kann auf verschiedene Arten geschehen. Entweder wird für jede Umprogrammierung ein Fachmann des Herstellers oder eines entsprechenden Dienstleisters bestellt, oder interne Mitarbeiter müssen für die Programmierung ausgebildet werden.

Die zu wählende Alternative hängt vom vorraussichtlichen Einsatz ab. Üblicherweise werden in kleinen Unternehmen externe Dienstleister bezahlt, weil interne Umschulungen sehr viel kostspieliger sind. Wird die Maschine allerdings sehr flexibel verwendet, kann sich auch die Umschulung eigener Mitarbeiter lohnen.

Man unterscheidet generell zwischen *Online*- und *Offline*-Programmierverfahren.

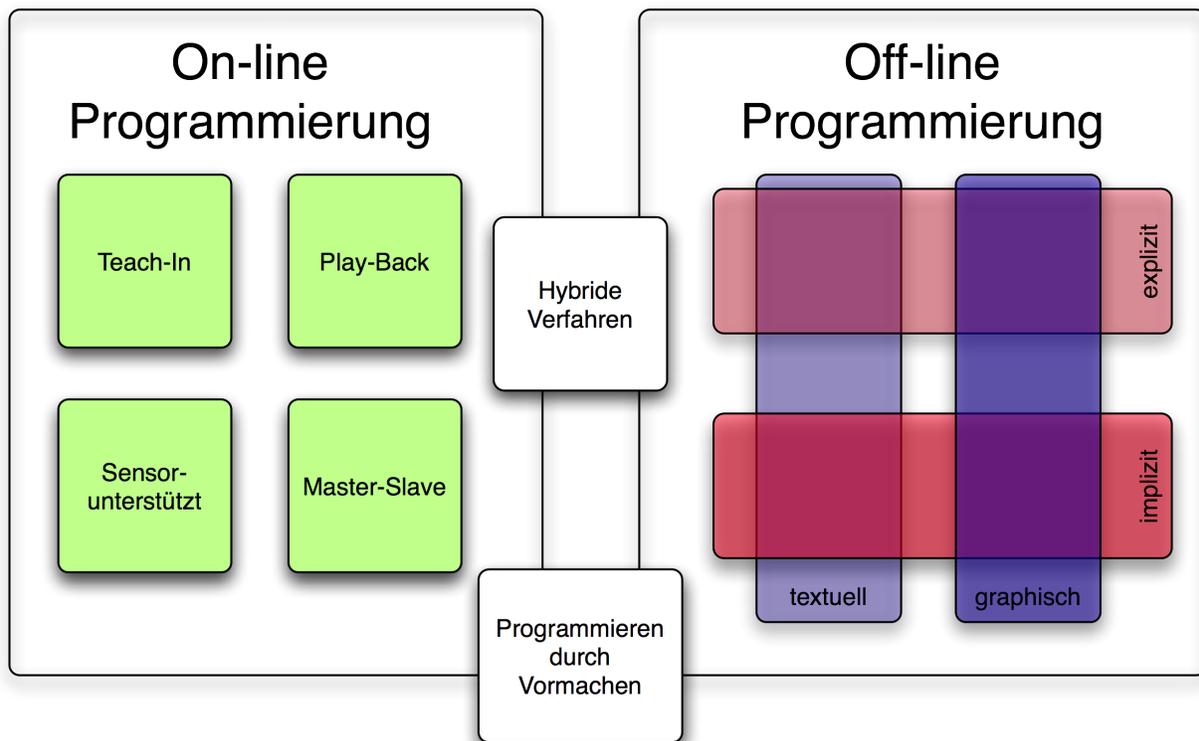


Abbildung 1: Klassifikation der Programmierarten für Robotersysteme.

Die Abbildung 1 zeigt die unterschiedlichen Arten, bekannt aus der Vorlesung.

- Online

Teach-In: Vorteile: Schnell zu lernen, einfache Anwendung, gut geeignet für simple Anwendungen. Nachteile: Unflexibel. Jede Umprogrammierung braucht wieder sehr viel Zeit. Während der Programmierung ist der Roboter belegt.

Play-Back: Vorteile: Kann von einem Fachmann der Anwendung ohne viel Vorwissen verwendet werden, schneller als Teach-In. Nachteile: benötigt zusätzlich Kraft-Momenten-Sensor sowie zusätzliche Speicher- und Steuereinheit. Analytische Trajektorien sind schwer zu programmieren.

Master-Slave: Ein zweiter (evtl. skaliertes) Roboter lohnt sich in fast keinem Fall, antiquierte Methode.

Sensorunterstützt: Vorteile: Schwer beschreibbare Bahnen können entsprechend einfach abgefahren werden. Gute Methode bei breiter Produktpalette mit hohen Ähnlichkeiten. Nachteile: Braucht zusätzliche Sensorik und Auswerteeinheiten. Höhere Fehleranfälligkeit.

- Offline Hier unterscheidet man zwischen *explizit* versus *implizit* und *textuell* versus *graphisch*. Bei den expliziten Verfahren braucht man weitreichende Kenntnisse auf Seiten des Benutzers/Programmierers. Für ein KMU ist dies üblicherweise nicht geeignet, da der Unterhalt einer spezialisierten Entwicklungsabteilung viel zu teuer ist.
- Mischverfahren
 - Hybride Verfahren
 - PdV

Lösung 3 (Umweltmodellierung)

Zuerst eine kurze Rekapitulation des Stoffes. Die folgende Abbildung 2 zeigt, aus welchen Modellierungen das Umweltmodell besteht.

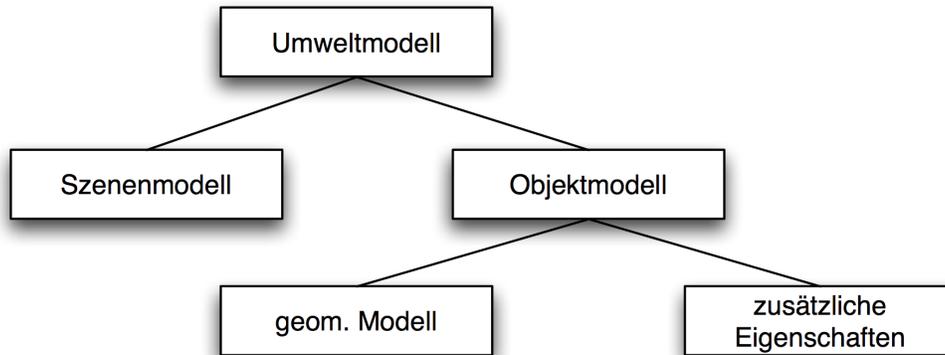


Abbildung 2: Modellierung des Umweltmodells.

Allgemein ist zu sagen, dass die Modellierung immer von der Aufgabe abhängt. So kann im Falle eines Putzroboters ein Raum als Gesamtheit modelliert werden, da zur Navigation nur die Hindernisse relevant sind, nicht aber die einzelnen Bestandteile des Raumes. Im Gegensatz dazu muss aber bei einem Serviceroboter sehr wohl zwischen den einzelnen Objekten unterschieden werden, damit der Roboter dann auch in der Lage ist, die Tasse, die sich auf dem Tisch befindet, finden zu können.

Zusammenfassend lässt sich sagen, dass man das *Szenenmodell* dazu benötigt, um die Beziehungen zwischen Objekten zu modellieren. Im Falle des Serviceroboters wird damit also modelliert, dass die Tasse *auf* dem Tisch steht. Das *Objektmodell* dient dazu, die Geometrie sowie zusätzliche Eigenschaften der Objekte zu beschreiben. Das Objektmodell benötigt man unter anderem für:

- Landmarken im SLAM (Simultaneous Localization And Mapping)
- Bahnplanung
- Kollisionsvermeidung
- Greifplanung
- Detektion von Menschen

Zur geometrischen Modellierung kann man

- Kanten
- Flächen
- 3D-Körper

verwenden. Zur Beschreibung der 3D-Körper bieten sich

- CSG (Constructive Solid Geometry)
- Zellzerlegung

an. CSG arbeitet mit

- Grundkörpern,
- Mengenoperationen und der
- Positionierung der Grundkörper.

Man muss hierbei beachten, was man mit CSG modelliert. Betrachtet man die beiden Objekte "Tasse" und "Tisch", so kann man sie als *ein* Objekt modellieren (sozusagen im Modell die Tasse auf den Tisch fixieren). Dies hat dann allerdings zur Folge, dass sie von dem oben genannten Serviceroboter nicht mehr benutzt werden kann. Wird die Tasse allerdings nicht gebraucht, so reduziert sich dadurch die Menge der Objekte und etwaige Berechnungen im Szenenmodell sind schneller.

Zur Beschreibung des Szenenmodells kann man *Entity Relationship (ER)* verwenden. Damit lassen sich Beziehungen zwischen Objekten beschreiben. Dies beinhaltet unter anderem die Lage zweier Objekte zueinander.

Die abschließende Auflistung soll einen Eindruck vermitteln, was für die anstehenden Aufgaben (Montage und Schweißen) alles zu modellieren ist:

- Werkstück
- Greifpunkte auf den Werkstücken
- Schweißpunkte, Schweißnähte
- Robotersystem
- Greifer
- Werkzeug
- Roboterzelle, Arbeitsraum
- Förderbänder
- Bewegliche Hindernisse, wie etwa Menschen

Lösung 4 (Aufgabenmodell)

Anhand der letzten Aufgabe sollen nochmals die einzelnen Teilaspekte der Übung durchgegangen werden und zusätzlich um die konkrete Programmierung des Systems erweitert werden. Der Ausgangspunkt ist in diesem Fall eine konkrete Aufgabe: das Sortieren von Schokoladentafeln.

Die Lösung gliedert sich in vier Abschnitte: eine Anforderungsanalyse des Zielsystems, die Modellierung des Systems, der Aufbau des Systems und schließlich die Erstellung des Programms.

Anforderungsanalyse

Flexibilität: keine, da feste, repetitive Aufgabe

Stückzahlen: große Stückzahlen, immer gleiches Werkstück

Einsatzort: fest, also keine Mobilität gefordert

Bedienung: keine, automatischer Betrieb

Schulung: nur für Wartung

Wartung: intern, wegen hoher Verfügbarkeit und geringer/kurzer Ausfallzeiten

Die benötigten Komponenten sind bereits in der Aufgabenstellung angegeben.

Modellierung

Objektmodellierung In der Objektmodellierung müssen mindestens das Werkstück, sowie der Arbeitsplatz, also die Roboterzelle, erfasst werden. Bei der Modellierung des Arbeitsplatzes ist zu überlegen, welche Komponenten modelliert werden müssen. Die Geometrie des "Werkstückes" (Schokoladentafeln!) muss geometrisch modelliert werden, damit

1. die Sensorik die Qualität der Tafeln messen kann
2. die Tafeln vom Roboter gegriffen werden können.

Geschickterweise ist die Geometrie der Tafeln schon bekannt; das Modell kann vom Produktentwurf übernommen werden (die Tafeln sind schließlich vorher im Prozess gegossen worden, wozu schon ein geometrisches Modell nötig ist). Festzulegen sind noch die Griffpunkte (genauer gesagt: die Ansaugpunkte).

Des Weiteren sind das Robotersystem, der Greifer und die Transportbänder zu modellieren.

Szenenmodell Im Szenenmodell müssen die Positionen der Komponenten des Systems sowie die Lage der Schokoladentafeln festgelegt werden. Im einzelnen sind zu modellieren:

- die Position von Roboter und Greifer
- die Anordnung und die Lage der Transportbänder

- die Lage der Schokoladentafeln auf den Bändern
- die Position des Sensorsystems, damit im Zusammenhang mit der Position der Transportbänder und der Lage der Tafeln auf den Bändern die Qualitätsmessung durchgeführt werden kann.
- die Modellierung von Hindernis- und Konfigurationsraum

Programmierung Noch offen.

Systemaufbau

Das System besteht im Wesentlichen aus den Komponenten:

- Transportband mit ankommenden Tafeln (gekennzeichnet mit T).
- Drei Transportbändern für die sortierten Tafeln (gekennzeichnet mit A, B, C gemäß den Güteklassen *1. Wahl*, *2. Wahl* und *Ausschuss*).
- Robotersystem
- Steuereinheit
- Sensorsystem

Abbildung 3 zeigt einen möglichen Aufbau des Systems. Die Tafeln werden von Band T auf die, gemäß ihrer Qualität, entsprechenden Bänder A-C umgesetzt.

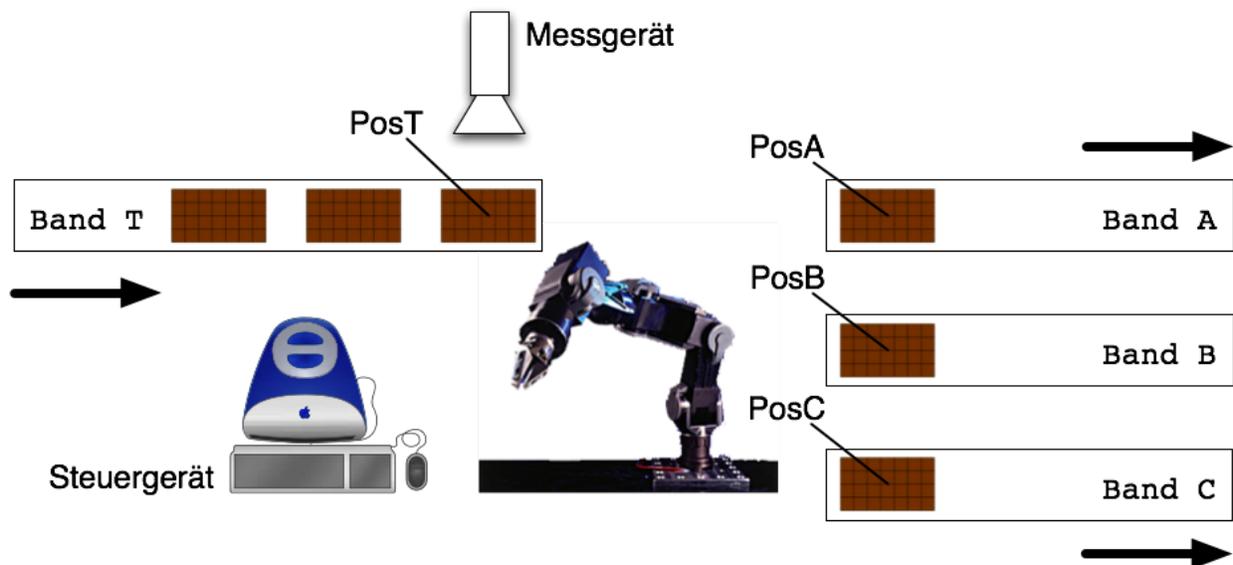


Abbildung 3: Aufbau des Robotersystems zum Sortieren von Schokoladentafeln

Es muss an dieser Stelle betont werden, dass das Schaubild nur eine mögliche Anordnung darstellt. Es sind noch viele andere Arrangements denkbar. So ist vorstellbar, dass die Tafeln mit der Qualitätsgüte *1. Wahl* gar nicht vom Robotersystem angefasst werden, sondern direkt zum Band A weitergeleitet werden, um unnötige, mögliche Beschädigungen der Tafeln zu vermeiden.

Programmierung

Das folgende Programm stellt wiederum nur eine Möglichkeit dar, wie der Ablauf sein kann.

```
while (Tafel) {
    Next();
    act_quality = Quality();
    if (act_quality == "1. Wahl"); then
        target_position = PosA;
    else if (act_quality == "2. Wahl"); then
        target_position = PosB;
    else
        target_position = PosC;
    endif
    Move(PosT);
    Grasp();
    Move(target_position);
    Release();
} end while
```

Bei diesem Programm ist zu beachten, dass die Bewegungen (Move();) nur ungenau spezifiziert worden sind. Hier sollte natürlich genauer auf die Bewegung der einzelnen Element eingegangen werden. Zumindest sollten explizit Anrück- und Abrückbewegungen programmiert werden. Zusätzlich sollten noch folgende Fragen geklärt werden:

- Wo erfolgt die Bahnplanung?
- Wie wird der Grasp()-Befehl überwacht? Was geschieht im Falle eines fehlerhaften Grasp()?
- Was passiert, wenn keine Tafel gegriffen wurde?
- Was passiert, wenn auf Band T keine Tafel vorliegt?
- Was passiert, wenn das Abtransportband voll ist?
- Wie geschieht allgemein die Fehlerbehandlung?
- Was passiert, wenn einer der Move()-Befehle fehlschlägt?

Lösung 5 (Interaktive Programmierung)

An qualitativen Merkmalen müssen der Greifvorgang für jedes Objekt, als auch die Trajektorien der Transportoperation aufgezeichnet werden. Darüber hinaus muss der Roboter Reihenfolgen des Aufräumvorgangs bzw. Ablegerelationen lernen.

Für die Aufzeichnung der Griffe sind Datenhandschuhe, für die Trajektorien Positionssensoren und für die Objektdetektion Kamerasysteme notwendig.

Die Operation, ein Buch von einem Tisch in ein Regal zu räumen, lässt sich grob segmentieren in: Greifen, Transportieren, Loslassen. Feiner kann sie in Anrücken, Zugreifen, Abrücken,

Transport in Richtung Regal, Anrücken auf Abstellposition, Abstellbewegung, Abrücken segmentiert werden.

Die Datenhandschuhe liefern die Fingerwinkel und die Positionssensoren die Geschwindigkeit der Handbewegungen. Während ein Objekt gegriffen ist, sind die Fingerwinkel sehr konstant und hoch, wenn nicht dann eher klein (Hand offen) und variabel. Die Geschwindigkeit der Hand ist während der Greifbewegung niedrig, sonst höher. Dies resultiert daraus, dass das Objekt an einem Ort gegriffen wird, während sich die Hand zur Anrückbewegung und während des Objekttransports stärker bewegt. Aus diesen - zum Teil komplementären - Informationen können die Phasen segmentiert werden.

Eine Verallgemeinerung kann durch die Makrooperator-Hierarchisierung geschehen. Transporttrajektorien können so wiederverwertet werden.

Für eine Transformation in einen anderen Arbeitsraum muss auf einer generelleren Repräsentation gearbeitet werden. Dazu gehört, dass nicht unmittelbar die kinematische Konfiguration des menschlichen Arms nachgeahmt wird, sondern nur die Trajektorie des Endeffektors, also der Hand. Jedoch kann es erforderlich sein, die Nachahmung noch unabhängiger zu gestalten, z.B. um Kollisionen zu vermeiden, die durch einen anderen Arbeitsraum auftreten können. In solchen Fällen muss der Roboter die Fähigkeit der Trajektorienplanung haben und es werden nur wichtige Ortspunkte aus der Vorführung übernommen.

Nach der Griffklassifikationshierarchie von Curkosky werden Bücher üblicherweise mit prismatischen Kraftgriffen erfasst und Stifte mit prismatischen Präzisionsgriffen gegriffen.