



# Automatische Adaption von Roboterprogrammen

## Intuitive Erstellung sensorgestützter Roboterprogramme

Heutzutage ist es möglich gut automatisierbare Aufgaben effizient durch Roboter ausführen zu lassen. Einfache, feste Aufgaben können alleine durch ein Abfahren fester Bewegungsfolgen ausgeführt werden, ohne Sensorik zur Umwelterkennung integrieren zu müssen.

Für komplexere Aufgaben ist es jedoch kaum möglich auf Sensorik zu verzichten, da der Roboter flexibel auf eine sich verändernde Umwelt reagieren muß.

Weiterhin können diese Sensordaten erst zur Laufzeit des Programmes ausgewertet werden. Somit sind solche Roboterprogramme in der Ausführung signifikant langsamer als die, die auf einen Sensoreinsatz verzichten können. Unter Umständen muß der Roboter sogar stoppen, um die Information zu erfassen und zu verarbeiten.

Darüber hinaus ist die Erstellung sensorgestützter Roboterprogramme komplex, zeitaufwändig und erfordert detaillierte Kenntnisse sowohl der Roboterprogrammiersprache als auch über die Art und den Informationsgehalt der Sensordaten.

Unser Ziel ist es, einem in Roboterprogrammierung unerfahrenem Anwender eine intuitive Benutzer- bzw. Programmierschnittstelle zur Verfügung zu stellen, um die Programmierung zu vereinfachen. Diese Schnittstelle soll es ermöglichen dem Roboter komplexe, sensorgesteuerte Handhabungsaufgaben durch einfache Anweisungen beizubringen, ohne detaillierte Angaben über die Arbeitsumgebung machen zu müssen.

## Flexibilität in einer sich verändernden Umwelt

Um die Entwicklung einfach und schnell zu gestalten, soll der Entwick-

ler nicht gezwungen sein, exakte und detaillierte Angaben über die Umweltveränderungen geben zu müssen. Dieser Mangel an Information muß dann durch die Sensorik des Roboters ausgeglichen werden, um dennoch eine korrekte und effiziente Ausführung zu ermöglichen.

Unbestimmtheiten bezüglich der Umwelt während oder zwischen den Ausführungen können daran unterschieden werden, ob sie durch die Aufgabe selbst bedingt sind, beispielsweise variable Positionen, oder ob sie durch Abnutzungserscheinungen der Arbeitsumgebung entstehen, wie es beispielsweise bei einem Verschleiß von Werkzeugen der Fall ist.

Weiterhin kann man die Unbestimmtheit derart klassifizieren, ob sie einmalig oder bei jeder Ausführung erneut erkannt werden muß. Eine einmalige Bestimmung wäre beispielsweise bei einer initial unbekanntem Objektposition vonnöten, die sich in den folgenden Ausführungen nicht mehr verändert. Falls diese Position jedoch bei jeder Ausführung auch nur geringfügig variiert, so muß sie in jedesmal erneut bestimmt werden. Anhand dieser zwei Kategorien kann man alle Veränderungen der Arbeitsumgebung in vier Klassen unterteilen:

Unbestimmtheit	Durch die Aufgabe bedingt	Durch Abnutzung bedingt
Einmaliges Auftreten	<i>Unbestimmtheit</i>	<i>Fehler</i>
Dauerhaftes Auftreten	<i>Variationen</i>	<i>Drift</i>

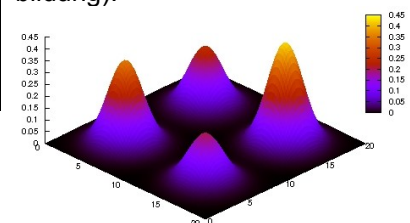
*Kalibrierungen* sind durch die Aufgabe bedingt und können zu Beginn der Ausführung bestimmt werden. Hierzu gehören beispielsweise die Lage und Orientierung von Objekten, aber auch Schwellwerte für sensorgestützte Operationen. Wenn hin-

gegen eine kontinuierliche Adaption für einen Aspekt der Aufgabe notwendig ist, so sprechen wir von *Variationen*. Hierzu gehören beispielsweise Toleranzen in Bezug auf die exakte Lage eines Objektes.

Unbestimmtheiten in der Umgebung, die nicht durch die Aufgabe selbst bedingt sind, aber die Umgebung plötzlich signifikant verändern, sind *Fehler*. Auf diese kann der Roboter nicht selbstständig reagieren, sondern muß einen Menschen benachrichtigen. Es ist jedoch möglich schleichende, minimale Veränderungen der Umgebung, den sogenannten *Drift*, durch eine kontinuierliche Adaption zu kompensieren.

## Automatische Adaption an die Arbeitsumgebung

Alle Unbestimmtheiten, seien sie gewollt - um den Fertigungsprozess zu vereinfachen - oder ungewollt - bedingt durch Verschleiß -, werden mittels automatisch ablaufender Lern- und Optimierungsverfahren kompensiert. So kann eine dauerhafte Ausführung des Roboterprogrammes ohne Ausfallzeiten gewährleistet und dennoch für einen industriellen Einsatz akzeptable Ausführungszeiten erreicht werden. Dabei wird auch eine selbstständige Rekalibrierung durchgeführt. Damit werden beispielsweise veränderbare Positionen als Wahrscheinlichkeitsverteilung abgespeichert, um die Suche bei folgenden Ausführungen beschleunigen zu können (siehe Abbildung).



## Einfache Integration in die Entwicklungsumgebung



Diese Lernverfahren müssen nicht bereits alle Kalibrierungen und er- vom Entwickler selbst konstruiert stellt erste Modelle der auftretenden werden, sondern sind bereits als Variationen. Sobald der Roboter die Komponenten in die Entwicklungs- Aufgabe korrekt durchführen kann, umgebung integriert. Der Entwickler endet die überwachte Lernphase. In wählt ein passendes Verfahren aus allen darauf folgenden Ausführungen und muß nur die zwingend notwen- überwacht der Roboter selbstständig digen Parameter angeben. An- die Umgebung um so auf einen schließlich überwacht er die ersten möglichen Drift schnell und sicher Ausführungen des Roboters. In reagieren zu können bzw. bei einem diesem Zeitraum erlernt der Roboter Fehler die Ausführung abubrechen.

### **Schnelle Entwicklung effizienter, flexibler Programme**

Mit diesem System wird eine einfache, intuitive Herangehensweise für die Erstellung sensorgestützter Roboterprogramme ermöglicht, die dennoch akzeptable Ausführungszeiten erreichen und sich selbst neu kalibrieren können.