

Bewegungsbefehle

Bewegungsbefehle eines Roboters enthalten folgende Angaben:

- eine Roboterstellung
- detaillierte Angaben zu der Bewegung, mit der die betreffende Roboterstellung angefahren werden soll.

Spezifikation der Bewegung

- Geschwindigkeit / Ausführungszeit
- Überschleifzone
- Verwendetes Werkzeug
- Werkstück-Koordinatensystem auf das sich die Koordinaten beziehen
- Bahnkorrektur

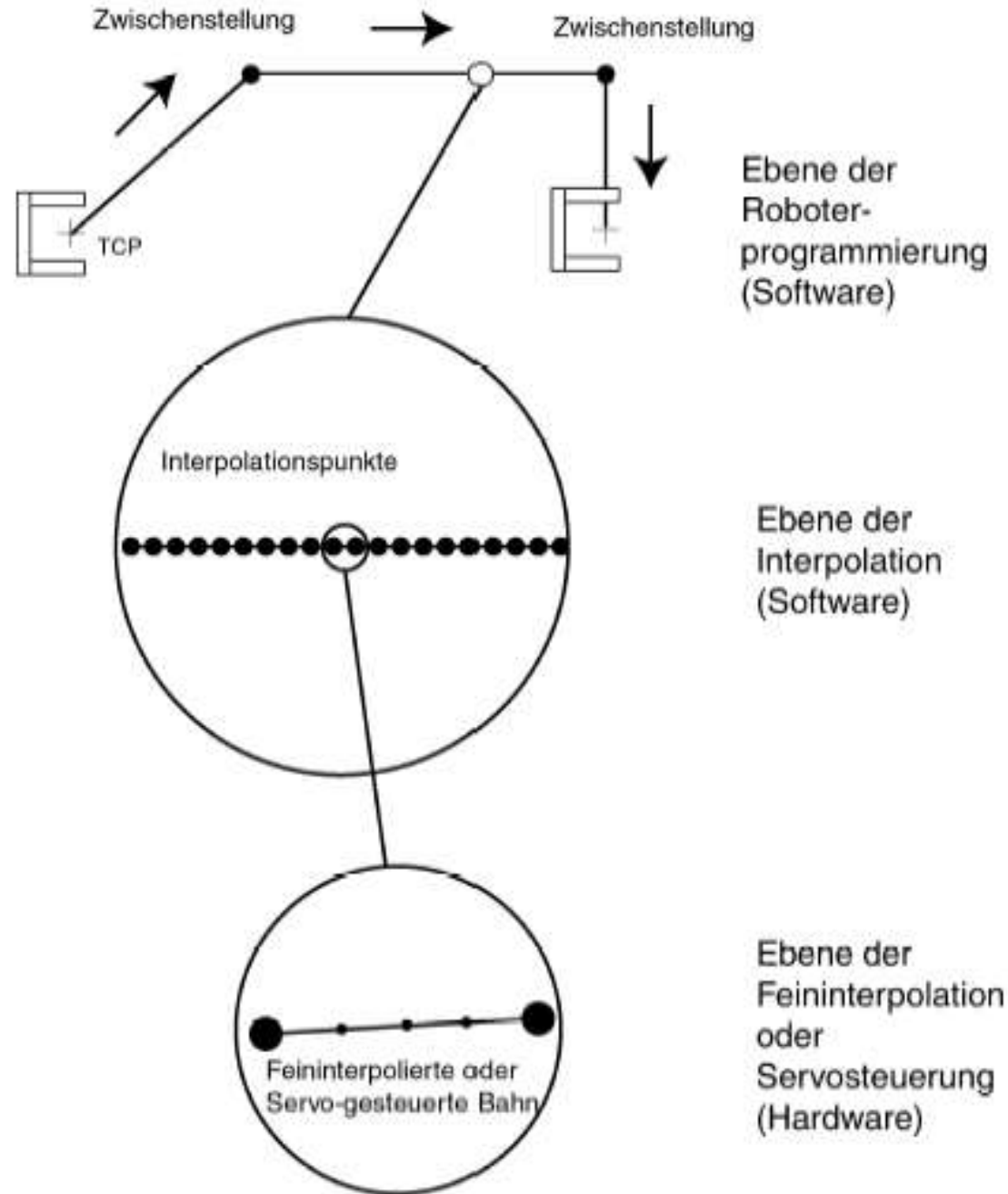
Interpolation

Um eine bessere Kontrolle über die Bahn zu erhalten, werden von der Steuerung Interpolationspunkte (Zwischenpunkte) eingefügt.

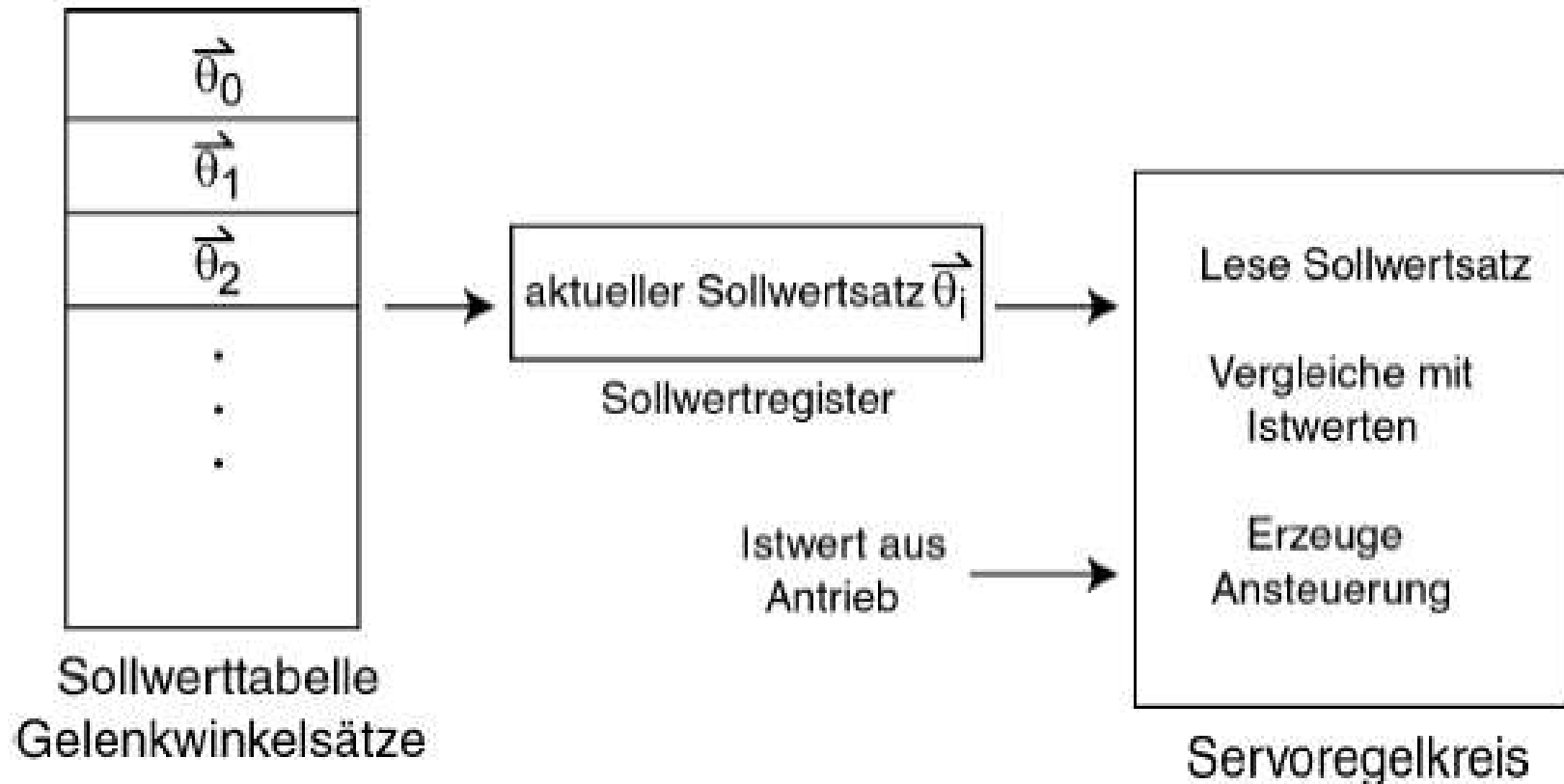
Ein Achsrechner berechnet die Stellung der Gelenke in den Interpolationspunkten mit der Rückwärtstransformation.

Vom Achsrechner werden die Sollwerte an eine parametrisierbare Regelung übergeben, die die Bewegung der Gelenke letztlich kontrolliert.

Programmierung, Interpolation und Regelung



Übergabe der Sollwerte an die Servoregler



Bahnparameter und Geschwindigkeitsprofil

Bahnparameter:

Winkel oder Weg, entsprechend Gelenktyp und Steuerungsart.

Aufgaben der Steuerung:

- Aus dem Stillstand heraus auf konstante Geschwindigkeit bringen oder umgekehrt.
- Von einer konstanten Geschwindigkeit auf eine andere konstante Geschwindigkeit bringen

Warum Geschwindigkeitsprofile?

Gleichmäßige Verteilung der Interpolationspunkte:
Hohe Beschleunigungen am Anfang und Ende

Folge: Verschleiß, Beschädigungen Werkstücke und
Werkzeuge

Notwendig: Begrenzung der Beschleunigung, „weiche“
Bewegungen

Rampenprofil

Die Geschwindigkeit wird linear, d.h. mit konstanter Beschleunigung, erhöht oder erniedrigt, bis der gewünschte Wert erreicht ist.

Nachteil: Die Beschleunigung und damit die Kraft ändert sich sprunghaft.

Sinuidenprofil

Die Beschleunigung wird mit einem \sin^2 -Profil weich bis zum Maximalwert gesteigert und fällt dann weich wieder auf Null ab.

Dabei ändert sich die Kraft stetig und weich, die Bewegung wird ruckfrei und flüssig.

$$\ddot{s}(t) = \sin^2 \left(\pi \frac{t}{t_B} \right) \quad \text{für} \quad 0 \leq t \leq t_B$$

Bahnsteuerungspolynome

Man kann für den Bahnparameter ein Steuerungspolynom 4. Ordnung verwenden.

Folge: Kräfte werden durch Polynom 2. Ordnung beschrieben, was ebenfalls einen weichen Übergang zwischen zwei beliebigen Geschwindigkeitswerten bewirkt.

Punkt-zu-Punkt-Steuerung (1)

PTP = point to point control.

Nur für den Zielpunkt werden mit der Rückwärtstransformation Gelenkwinkel bestimmt.

Mit Interpolationspunkten auf Gelenkwinelebene kann ein Geschwindigkeitsprofil eingehalten werden.

Wird auf Interpolationspunkte ganz verzichtet, entscheiden nur die Servokreise über die Bewegung.

Punkt-zu-Punkt-Steuerung (2)

Die PTP-Bewegung ist für die Steuerung einfach und problemlos.

Der TCP erreicht exakt den Zielpunkt, aber die Bahn ist unkontrolliert und nicht vorhersehbar. Gefahr!

Die auftretenden Geschwindigkeiten und Beschleunigungen sind nicht vorhersehbar!

Punkt-zu-Punkt-Steuerung (3)

PTP-Steuerung ist *ungeeignet* für:

- Bewegungen mit Werkstückbearbeitung,
- Bewegungen in der Nähe von Hindernissen.

PTP-Steuerung ist *geeignet* für:

- Reine Positionierbewegungen bei genügend freiem Raum um den Arm,
- um Singularitäten zu umgehen.

Punkt-zu-Punkt-Steuerung (4)

Varianten:

Asynchrone PTP-Steuerung:

Die Gelenke werden so schnell wie möglich in die Endstellung gefahren. Folge: Die Achsen beenden nacheinander ihre Bewegung.

Synchrone PTP-Steuerung:

Die Verfahrzeit der übrigen Achsen werden an die langsamste Achse angepasst, alle Achsen beenden ihre Bewegung gleichzeitig.

Ermittlung der Interpolationspunkte für synchrone PTP-Steuerung

- Berechnung der Bewegungsdauer für jedes Gelenk aus Verfahrestrecke, max. Geschwindigkeit und Profil
- Die Achse mit größter Bewegungsdauer wird Leitachse
- Die übrigen Achsen werden verlangsamt, bis ihre Bewegungsdauer mit der Leitachse übereinstimmt
- Vollsynchrone PTP-Steuerung:
Beschleunigungsphasen werden an Leitachse angepasst
- Berechnung der Interpolationspunkte.

Kartesische Bahnsteuerung (Continuous Path, CP) (1)

Es werden in kurzen Abständen Interpolations-punkte entlang der gewünschten Bahn erzeugt.

Für jeden Interpolationspunkt wird die Rückwärtstransformation durchgeführt.

Zwischen den Interpolationspunkten findet Feininterpolation oder Servosteuerung statt.

Der TCP bewegt sich (fast) exakt auf der gewünschten Bahn.

Kartesische Bahnsteuerung (Continuous Path, CP) (2)

Hauptvarianten:

Lineare CP-Bahn:

Der TCP bewegt sich auf einer Geraden.

Zirkulare CP-Bahn:

Der TCP bewegt sich auf einem Kreisausschnitt.

Kartesische Bahnsteuerung (Continuous Path, CP) (3)

Die CP-Steuerung wird in der Regel mit einem „weichen“ *Geschwindigkeitsprofil* kombiniert, das sich dann auf die tatsächliche Geschwindigkeit des TCP auf der Bahn bezieht.

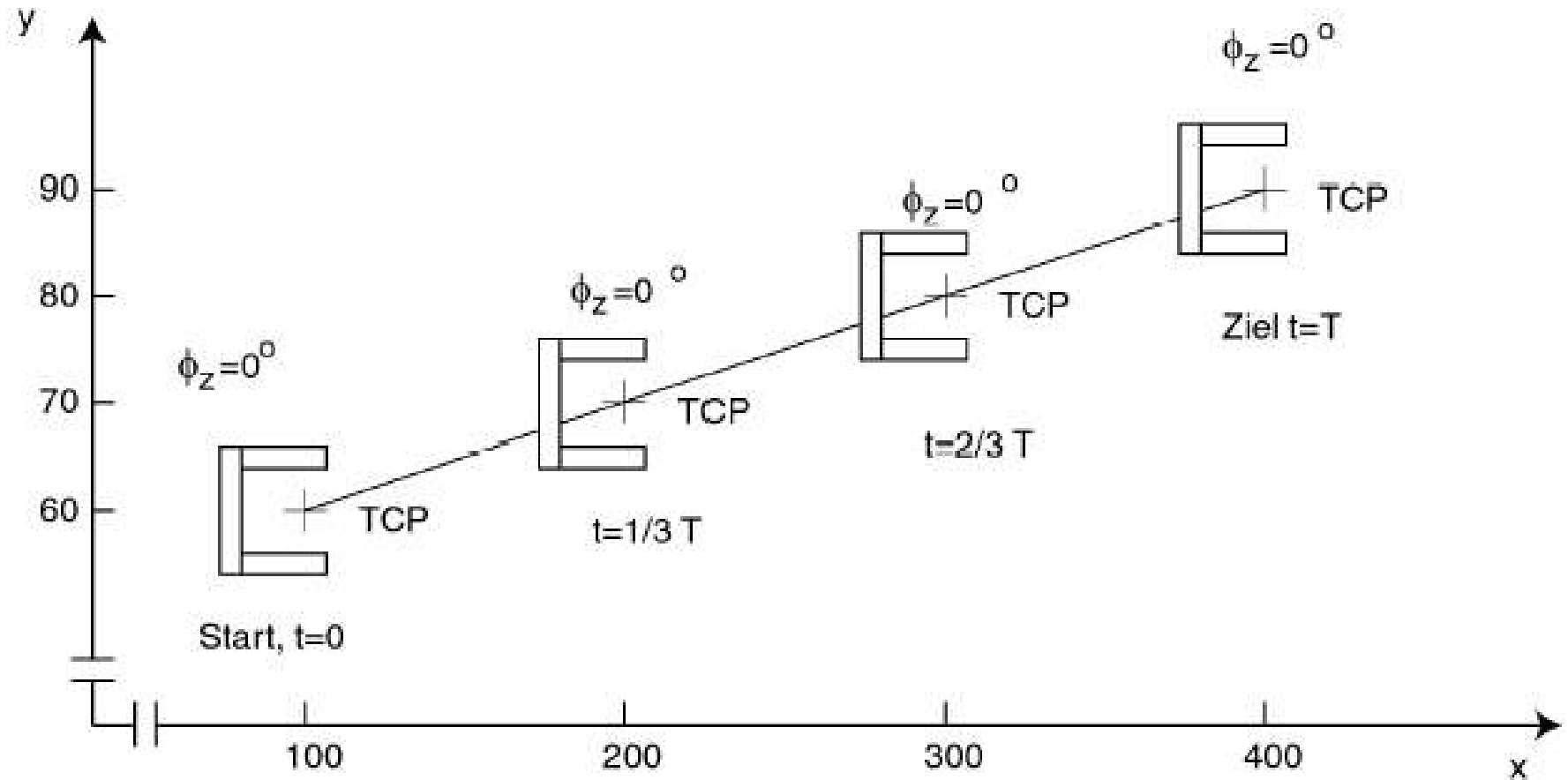
Auf der CP-gesteuerten Bahn kann auch eine *Umorientierung* (Rotation) stattfinden, die synchron zur Fortbewegung (Translation) stattfindet.

Eine CP-Bewegung erfordert mehr Gelenkbewegungen und dauert deshalb manchmal länger als eine PTP-Bewegung.

Vorgehensweise beim Entwurf einer CP-Bahn

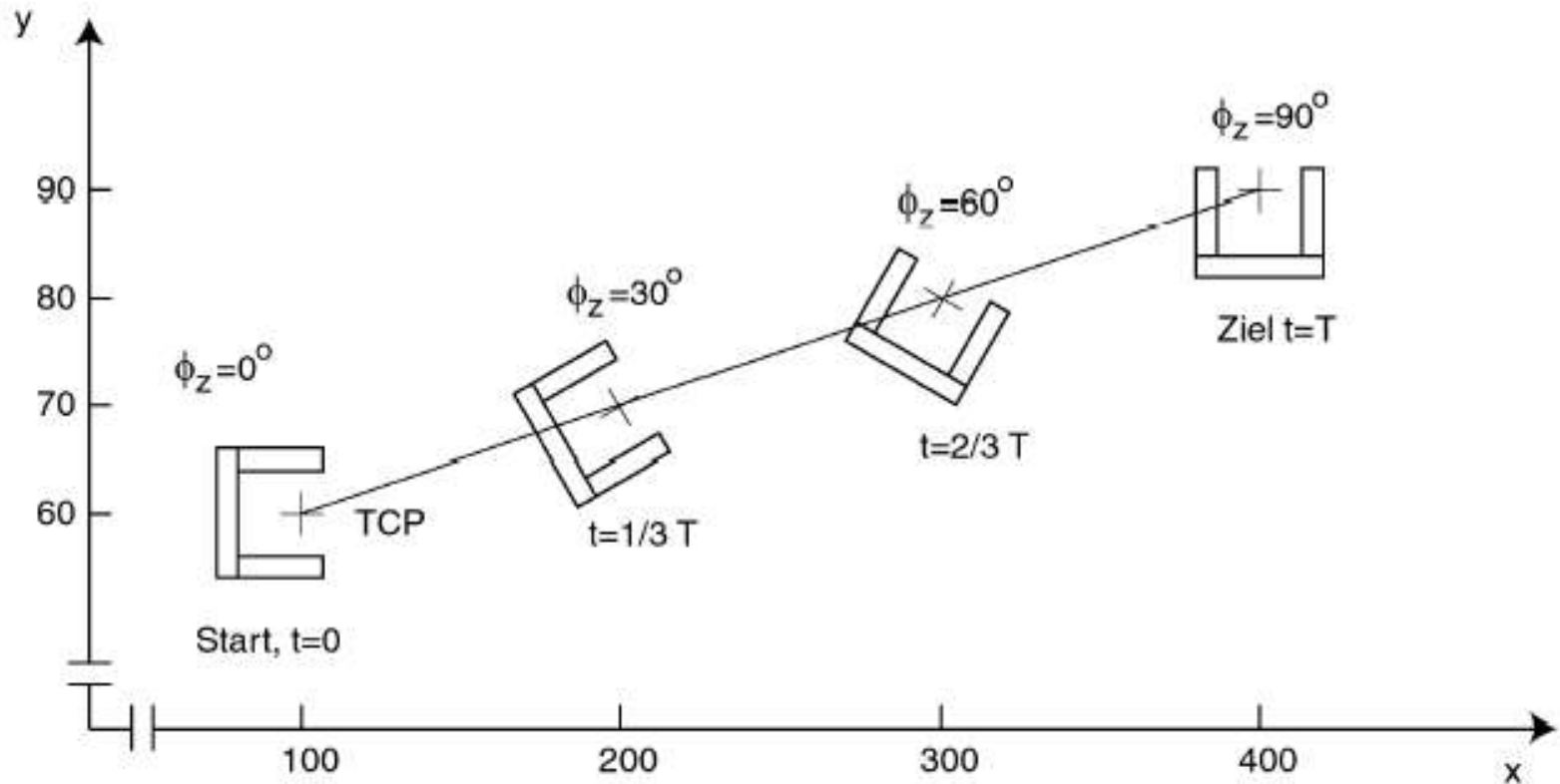
1. Berechnung der Translationsstrecke und der Orientierungsänderung,
2. Berechnung der Bewegungsdauer als das Maximum aus den beiden Bewegungsdauern,
3. Festlegung der Geschwindigkeits- und Beschleunigungswerte,
4. Festlegung der Interpolationspunkte mit Pos. u. Orient.,
5. Ermittlung der Gelenkkoordinaten in den Interpolationspunkten durch Rückwärtstransformation,
6. Eintragung der Gelenkkoordinaten in die Sollwerttabelle, von dort Übergabe an die Steuerungshardware,
7. Evtl. Feininterpolation durch die Steuerungshardware, Übergabe an Servokreise, von dort Motoransteuerung.

Beispiel für eine lineare CP-Bahn ohne Umorientierung



Lineare kartesische Interpolation (lineare CP-Bahn) ohne Orientierungsänderung

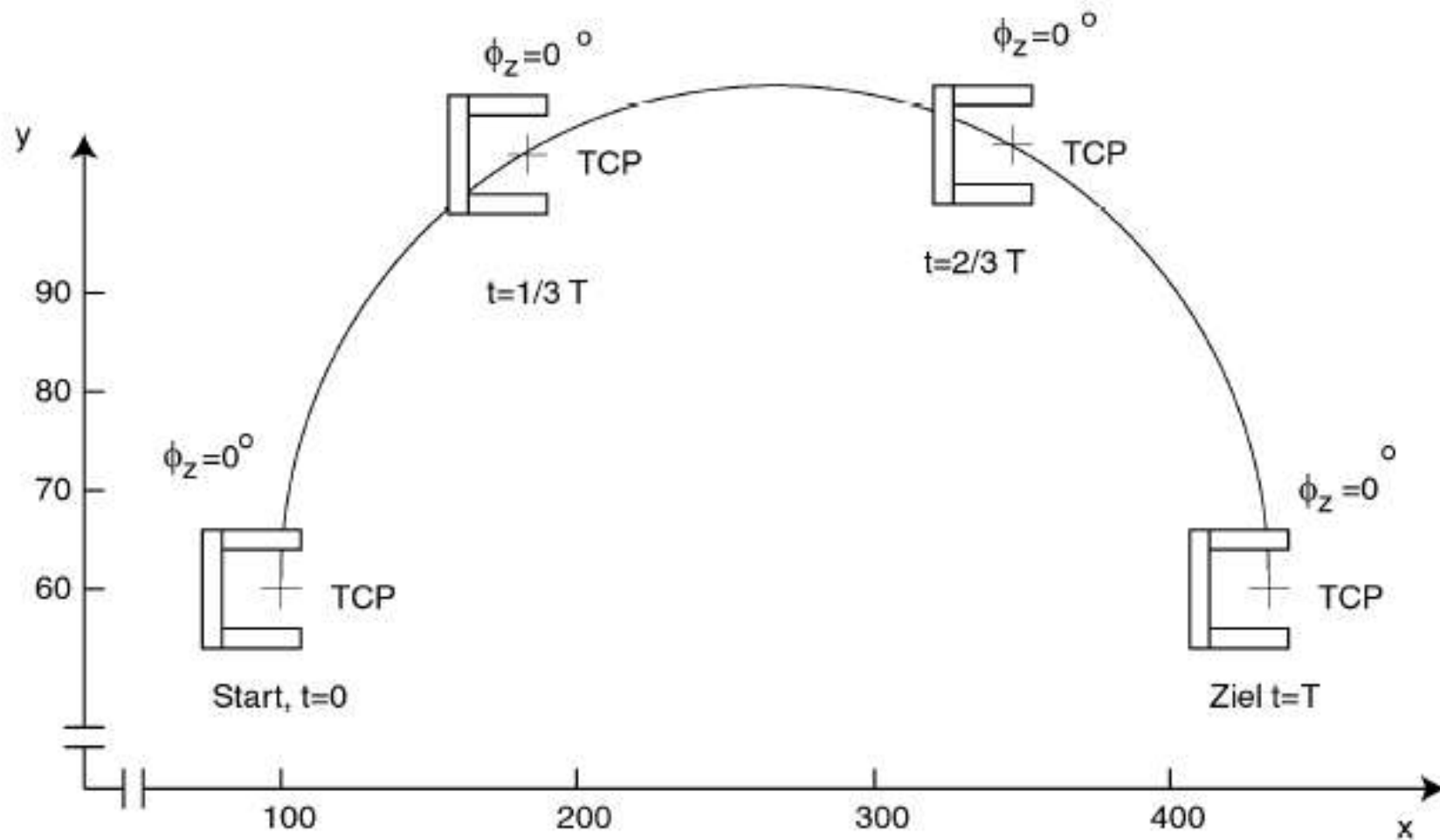
Beispiel für eine lineare CP-Bahn mit Umorientieng



Lineare kartesische Interpolation (lineare CP-Bahn) mit Orientierungsänderung

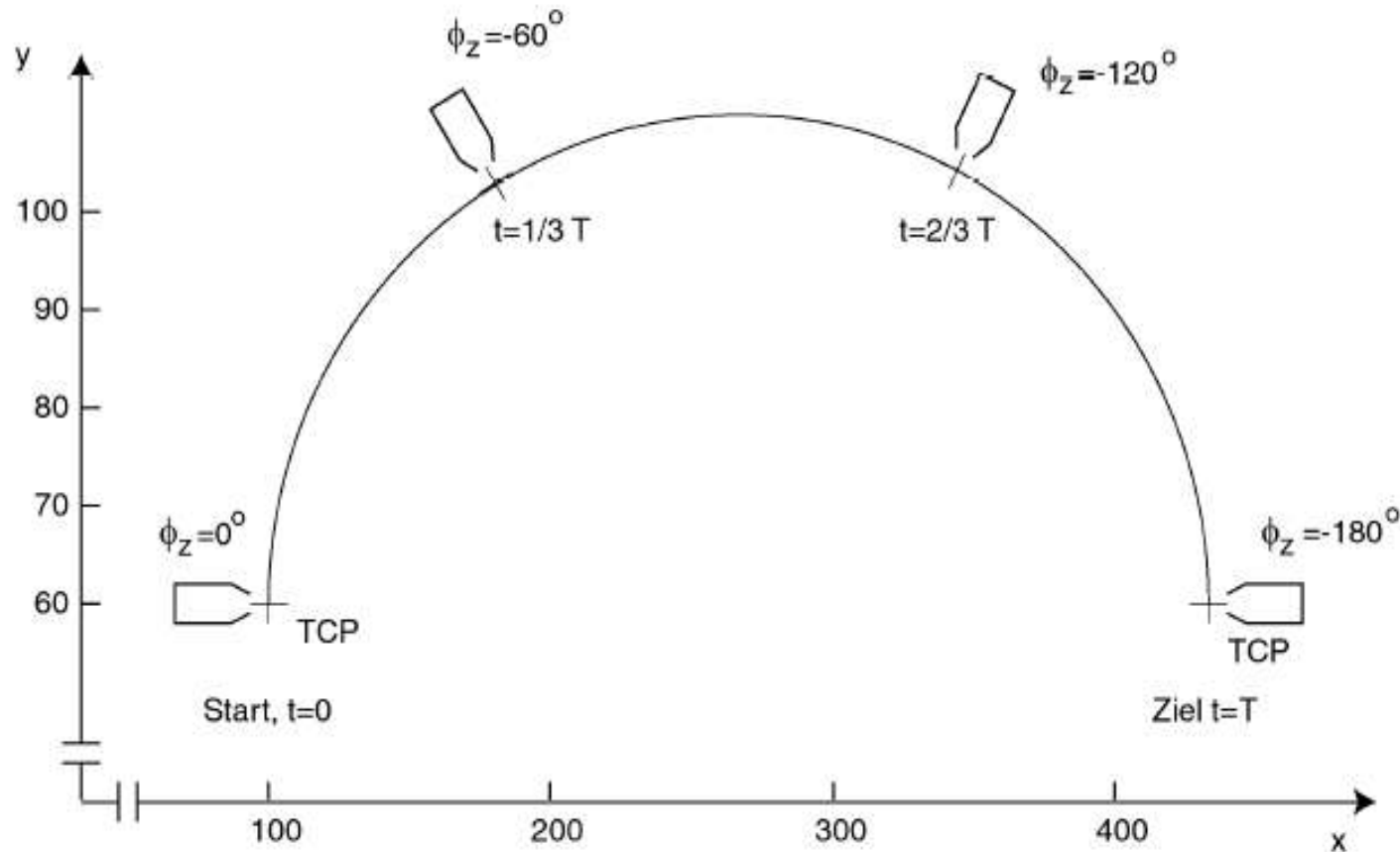
Abbildung 2.5: Lineare kartesische Interpolation (lineare CP-Bahn) ohne (oben) und mit (unten) Orientierungsänderung. Jeder Zwischenpunkt ist mit Position und Orientierung festgelegt. Im unteren Beispiel ist die Startstellung $x = 100, y = 60, \phi_z = 0^\circ$ und die Zielstellung $x = 400, y = 90, \phi_z = 90^\circ$. Es sind nur vier Interpolationspunkte gezeichnet, um eine gute lineare Bahn zu erhalten wird die Steuerung mehr als vier Punkte berechnen und durchfahren.

Beispiel für eine zirkuläre CP-Bahn ohne Umorientierung



Zirkulare kartesische Interpolation (zirkuläre CP-Bahn) ohne Orientierungsänderung

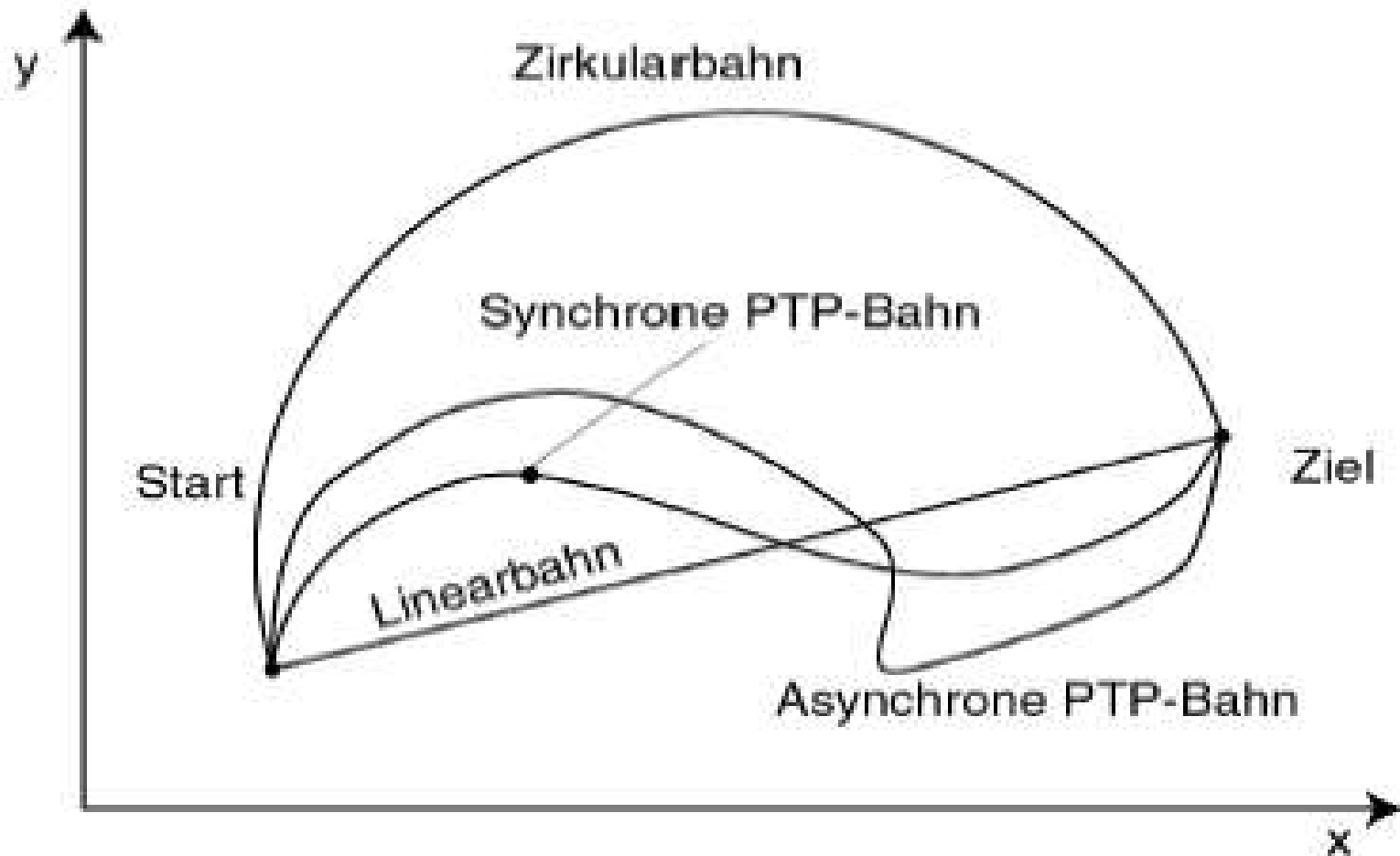
Beispiel für eine zirkulare CP-Bahn mit Umorientierung



Zirkulare kartesische Interpolation (zirkulare CP-Bahn) mit Orientierungsänderung

Abbildung 2.6: Zirkulare Interpolation ohne (oben) und mit Orientierungsänderung (unten). Es sind nur zwei Interpolationspunkte eingezeichnet, für eine gute kartesische Bahnsteuerung braucht man viel mehr Punkte.

Bahnbeispiele für verschiedene Steuerungsarten



Praktische Anwendung der CP-Bahnen

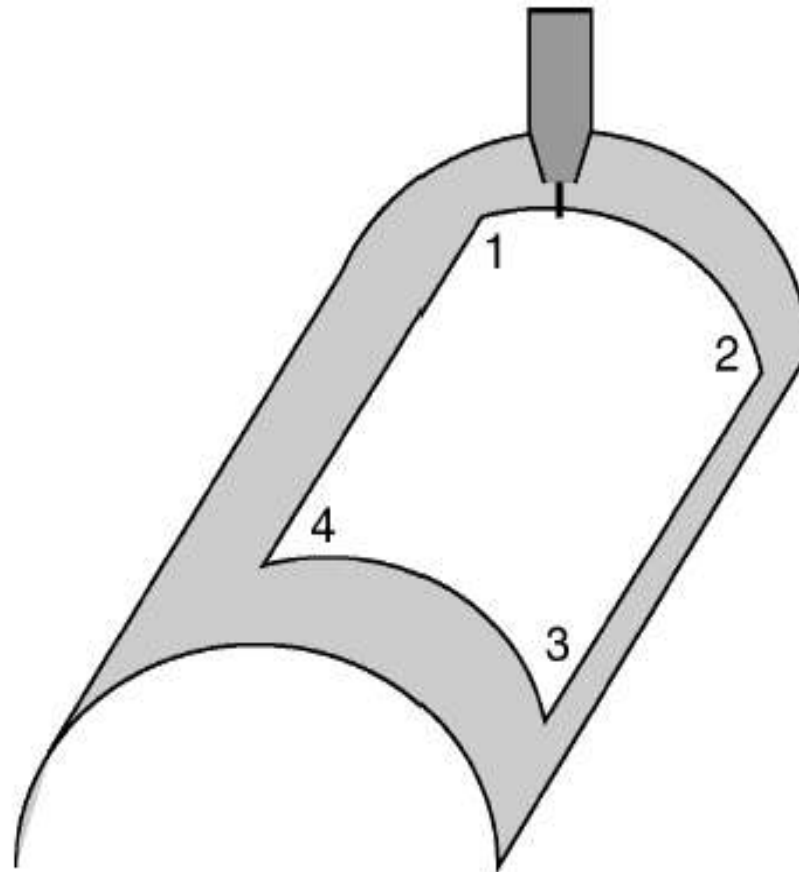


Abbildung 2.7: Aus einem Halbzylinder wird ein Ausschnitt herausgesägt. Dabei wird der TCP auf den Bahnsegmenten 2-3 und 4-1 auf linearer CP-Bahn bewegt und auf den Abschnitten 1-2 und 3-4 auf zirkularer CP-Bahn.

Durchfahren von Zwischenstellungen ohne Anhalten („Überschleifen“)

Für die Zwischenstellungen einer Bahn gibt es zwei Möglichkeiten:

1. *Am Zwischenpunkt ist ein Halt mit Stillstand des Roboters erforderlich.*

Mögliche Gründe: Warten auf ein Signal, prozessbedingtes Warten, Werkzeugwechsel.

2. *Am Zwischenpunkt ist kein Halt erforderlich.*

Möglich Gründe: Der Zwischenpunkt wurde nur eingefügt, um ein Hindernis zu umgehen, der Prozess erlaubt kleine Bahnabweichungen.

Durchfahren von Zwischenstellungen ohne Anhalten, Nutzen

Das Durchfahren eines Zwischenpunktes ohne Anhalten vermeidet unnötiges Abbremsen und Beschleunigen, d.h.

- spart Zeit
- lässt weniger Kräfte wirken
- spart Energie

Aber: Würde man einen Zwischenpunkt (mit Richtungsänderung) exakt anfahren und ohne Stopp fortsetzen, wäre ein scharfer Ruck die Folge!

Fly-By-Punkte, Überschleifzone, Überschleifkugel

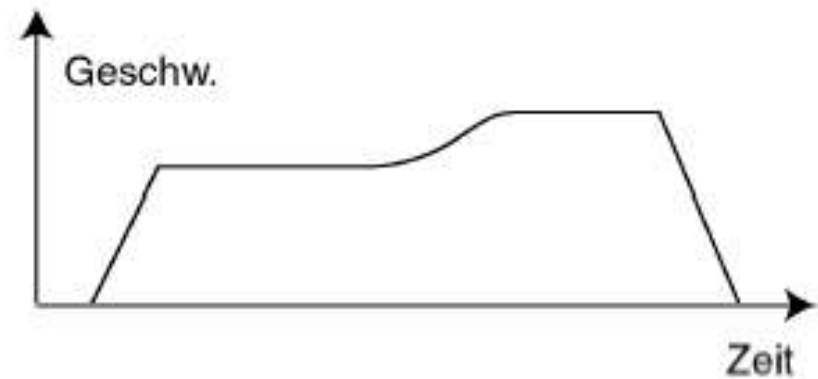
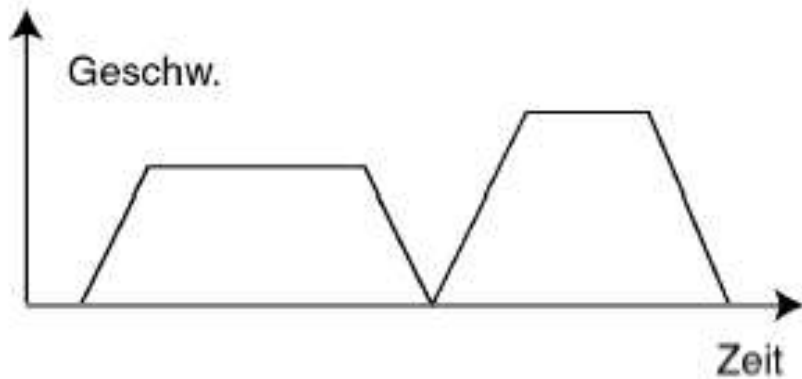
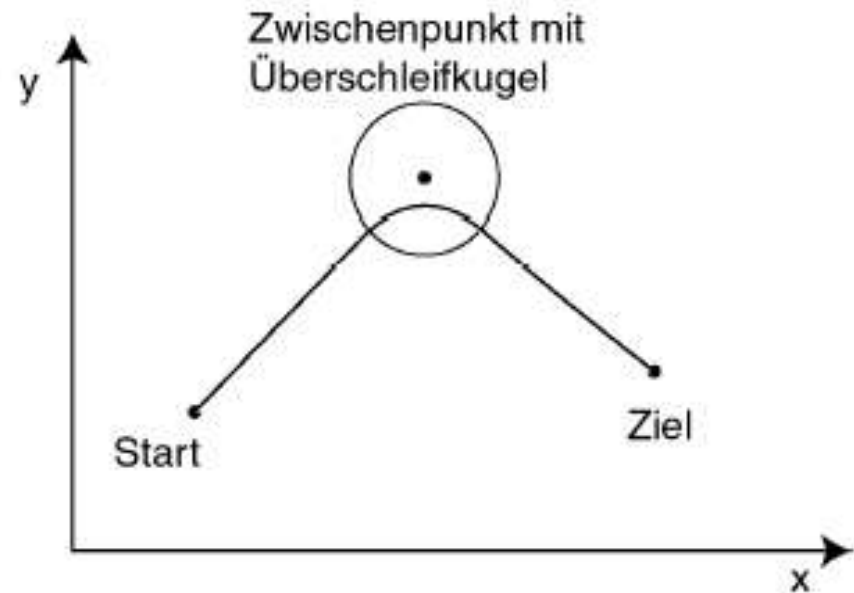
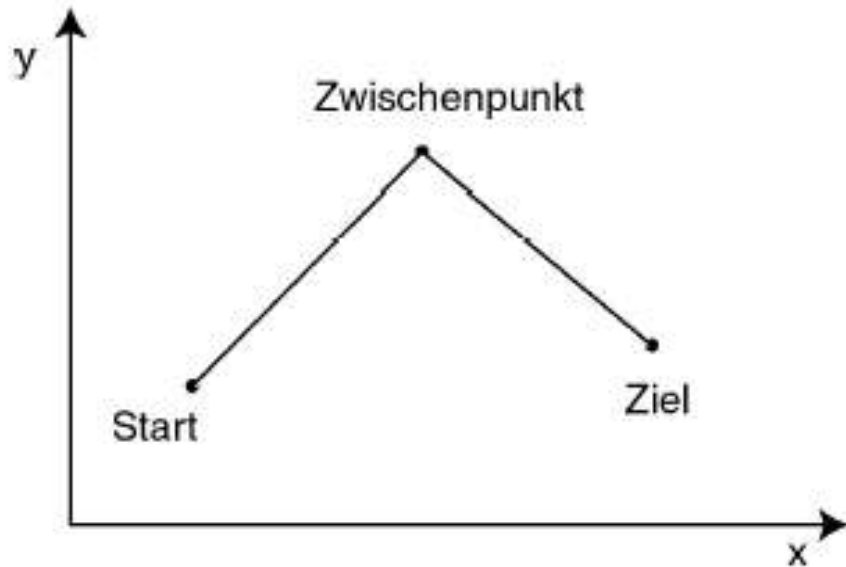
Überschleifen: Abweichung von exakter Bahn und allmähliches Einschwenken auf die neue Bahn mit flüssiger und weicher Bewegung.

Dazu ist Raumzone nötig: *Überschleifzone* oder *Überschleifkugel*.

Nur die Überschleifzone wird erreicht, nicht der exakte Bahnpunkt! Daher: *Fly-By-Punkt*.

Die Geschwindigkeit wird in der Überschleifzone nicht Null.

Beispiel zum Überschleifen



Bewegung links ohne und rechts mit Überschleifen.

Geschwindigkeits- und Positions- überschleifen

Das *Geschwindigkeitsüberschleifen* beginnt, wenn die Geschwindigkeit einen festgelegten Minimalwert unterschreitet.

Nachteil: Abhängig vom Geschwindigkeitsprofil.

Das *Positionsüberschleifen* beginnt, wenn der TCP in die Überschleifkugel eintritt. Außerhalb der Überschleifkugel wird die Bahn exakt eingehalten.

Vorteil: Gut kontrollierbar für Programmierer.

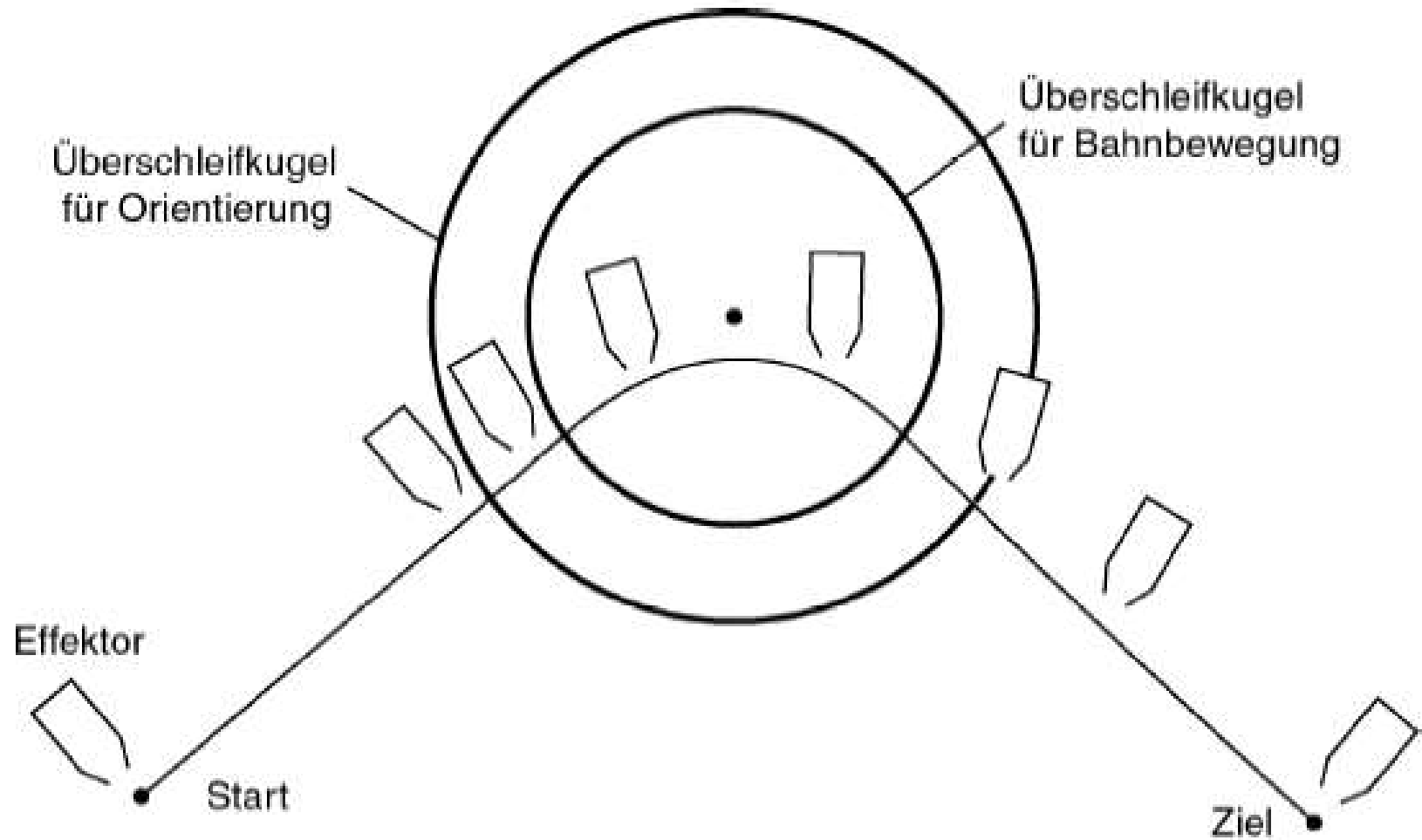
Weitere Details zum Überschleifen

Das Überschleifen kann auf Gelenkwinelebene oder auf CP-Ebene erfolgen

Wenn der Zielpunkt nach dem Fly-By-Punkt eine andere Orientierung aufweist erlauben viele Steuerungen eine separate Zone für die Umorientierung.

Ein Fly-By-Punkt kann für den Wechsel der Steuerungsart genutzt werden.

Beispiel für zwei verschiedene Überschleifkugeln



Syntaxbeispiel für linearen Bewegungsbefehle in RAPID

MoveJ p1,vmax,z30,Zange1

MoveJ = Move Joint = synchrone PTP-Bewegung,

Zielstellung p1,

Geschwindigkeit vmax (vordefiniert),

Überschleifradius z30=30mm, Ziel wird nicht erreicht,
TCP kommt nicht zum Stillstand

Werkzeug „Zange1“, Werkzeug-KS von Zange1
erreicht Zielstellung

Syntaxbeispiel lineare CP-Bewegung

MoveL p2,v100,fine,Duese2

MoveL = Move linear = lineare CP-Steuerung,

Zielstellung p2,

Geschwindigkeit v100,

fine = kein Überschleifen, TCP kommt in Zielstellung zum Stillstand

Werkzeug Duese2.

Syntaxbeispiel zirkulare CP-Bewegung

MoveC Kreispunkt,EndPunkt,v500,z10,Dorn

MoveC = Move circular = zirkulare CP-Steuerung,

Zielstellung EndPunkt,

Zwischenpunkt Kreispunkt legt Kreisbogen fest,

Geschwindigkeit v500=500mm/s,

Überschleifzone z10=10mm, TCP kommt nicht zum Stillstand und erreicht EndPunkt nicht,

TCP von Werkzeug Dorn wird benutzt.