

Vom GCODE zur Cyclic-Synchronous-Position Bewegung



Author: Jürgen Rall

Eine der größten Herausforderungen bei Motion-Control ist die Interpolation von Koordinaten-Punkten. Um diese Herausforderung in den Griff zu bekommen, muss zuerst die Ausgangssituation betrachtet werden.

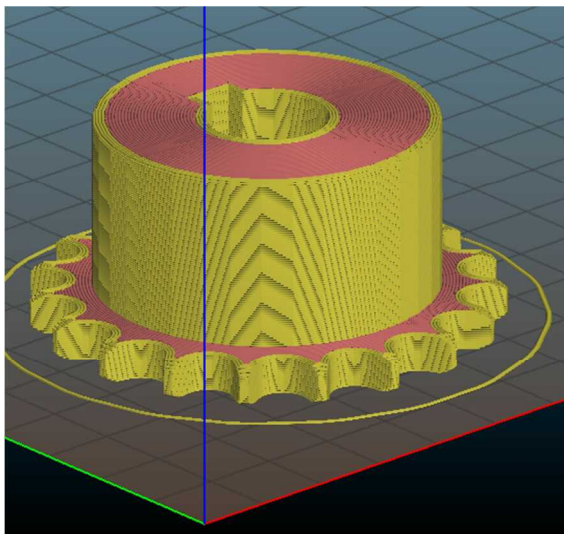
Bild: CAD-Design



In der Regel wird am Anfang über eine CAD-Software ein 3D-Modell erstellt und in einem stereolithographischen Datei-Format (z.B. STL "Standard Triangle Language") abgelegt. Die stereolithographischen Koordinaten beschreiben zwar graphisch das 3D-Modell, sind jedoch für einen Produktionsablauf ungeeignet, da sie zum einen keine geschlossenen Fahrfolgen beschreiben, zum anderen keine Füllungen der Zwischenräume berücksichtigen. Ebenfalls sind keine Informationen vorhanden, die Fahrgeschwindigkeit oder periphere Steuerbefehle beinhalten.

Um nun ein 3D-Modell produktionstechnisch zu bearbeiten, müssen die stereolithographischen Informationen in Produktionskoordinaten gewandelt werden. Dies beschreibt nun der GCODE mit einer Vektor-Abfolge in einem 3D-Raum, die nun zur produktionstechnischen Verarbeitung geeignet ist.

Bild: Slicer-Modell



Als Beispiel sei hier ein 3D-Drucker erwähnt. Dieser besitzt typischerweise jeweils eine Achse für die X/Y/Z-Bewegung, sowie eine R-Achse zum Materialvorschub. Im 3-dimensionalen Raum beschreiben 2 Koordinaten einen Vektor, der für die X/Y/Z-Achsen eine gleichmäßige Bewegung festlegt, so dass der Extruder (Werkzeugkopf des 3D-Druckers) in einer gleichmäßig linearen Bewegung von Koordinate 1 nach Koordinate 2 bewegt wird (Anmerkung: neben linearen Bahnbeschreibung kann der GCODE auch Kurvenbahnen beschreiben).

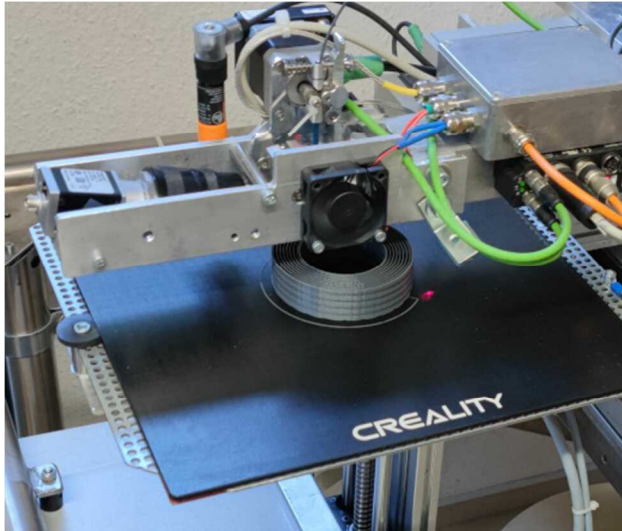
Über den räumlichen Vektor wird somit auch die Geschwindigkeit der jeweiligen Achsbewegung definiert.

Vom GCODE zur Cyclic-Synchronous-Position Bewegung



Author: Jürgen Rall

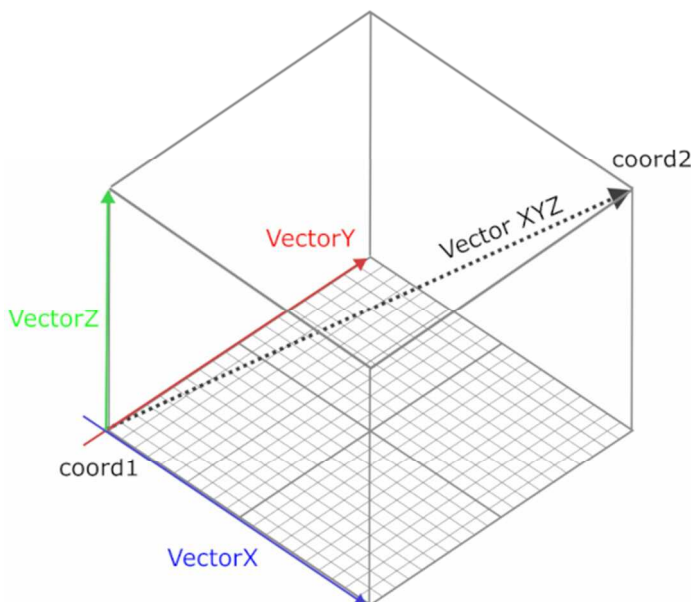
Bild: 3D-Drucker



Im Cyclic-Synchronous-Position Mode eines Antriebs wird die Achsbewegung zwischen 2 Koordinatenpunkte in Iterationsschritte unterteilt (Waypoints). Die Geschwindigkeit des Antriebs wird dabei über die Schrittweite definiert (Je größer die Schrittweite, umso höher die Geschwindigkeit). Da der Vektor zweier Raumkoordinaten die Geschwindigkeit der Gesamtbewegung (des Werkzeugkopfes) vorgibt, müssen pro Achse Teilvektoren gebildet werden, um eine koordinierte lineare Bewegung aller Achsen zur Zielkoordinate zu gewährleisten.

Die Berechnung der Teilvektoren erfolgt über trigonometrische Rechenoperationen, die entweder statisch (im Voraus), oder dynamisch (während des Prozessablaufs) durchgeführt werden.

Bild: Vektor-Iteration



Da die Anzahl der Iterationsschritte eines Vektors zweier Raum-Koordinaten auch für alle Teilvektoren gilt, kann nun die Iterationsschrittweite (Increment) zwischen den Koordinatenpunkten für jede Achse berechnet werden. Somit ergibt sich für jede Achse eine eigene Punkteliste (Waypoints), die die koordinierte Bewegung der Achse zum Zielpunkt beschreibt.

Ein 3D-Modell kann (je nach Auflösung) mehrere Millionen Waypoints pro Achse umfassen. Dies kann enorme Speichermengen benötigen.

Daher ist die dynamische Berechnung der Waypoints über vordefinierte Iterationen sinnvoller, da hierbei die Berechnung über einfachen Rechenoperationen erfolgen kann. Somit werden die Koordinatentabellen pro Achse im Voraus erstellt, die Iterationsschritte jedoch dynamisch berechnet.

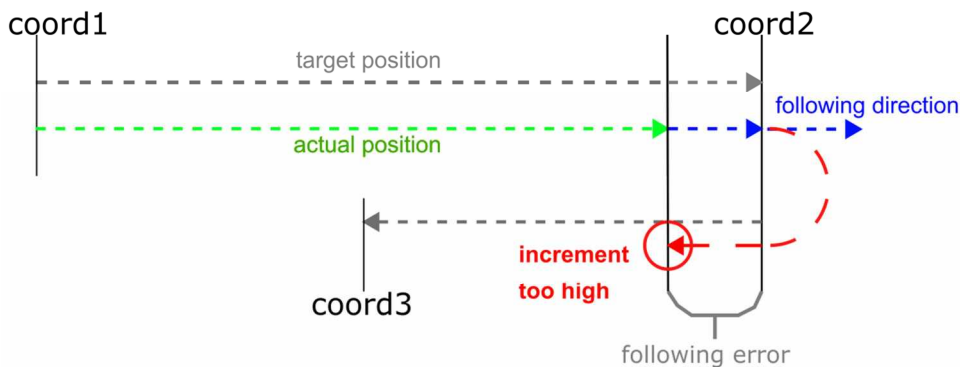
Vom GCODE zur Cyclic-Synchronous-Position Bewegung



Author: Jürgen Rall

Die Problematik besteht jedoch darin, dass ein Achsantrieb einen hoch-dynamischen Mechanismus darstellt, bei dem die Trägheitsmomente mit berücksichtigt werden müssen. Wird eine Bewegung gestartet, so bildet sich zwischen Soll-Position (Target Position) und Ist-Position (Actual Position) durch die Massenträgheit ein Abstand (Following Error), der vergleichbar einem Gummiseil, von der Antriebsregelung nachgeführt werden muss. Wird nun eine Bahn zwischen 2 Koordinatenpunkten über die Iterationspunkte (Waypoints) gefahren, so beträgt der dynamische Abstand zwischen Soll- und Ist-Position beim Erreichen der Soll-Geschwindigkeit seinen höchsten Wert, der dann im Laufe der gleichmäßigen Bewegung wieder reduziert wird.

Bild: Direction change with following error



Wird die Bewegung zwischen 2 Koordinatenpunkten in dieselbe Richtung auch zu den Folgekoordinatenpunkte weitergeführt, bleibt der dynamische Abstand zwischen Soll- und Ist-Position nahezu konstant. Bei einem Richtungswechsel hingegen kann sich der dynamische Abstand erhöhen, da die Soll-Position in die entgegengesetzte Richtung inkrementiert wird. Der dabei entstehende Following-Error kann nun die Dynamikgrenzen des Antriebs überschreiten und somit den Antrieb in einen Fehlerzustand versetzen.

Bei der Berechnung der Folge-Waypoints muss daher unbedingt ein Richtungswechsel berücksichtigt werden. Dabei müssen die Dynamikgrenzen des Antriebs (Toleranzen) bekannt sein. Bei zu geringen Toleranzvorgaben kommt es zu "abgehackten" Bewegungen durch ständigen Bewegungsstillstand, bei zu groß eingestellten Toleranzen wird die Bewegung zwar fließender, jedoch kann der Following-Error die Dynamikgrenzen überschreiten.

Die Sybera Motion-Bibliothek berücksichtigt die o.g. Effekte und bietet entsprechende Einstellmöglichkeiten für den optimalen Betrieb aller Antriebe im Cyclic-Synchronous-Position Mode.