

Eine interaktive Umgebung für das Bewegungsdesign mit Bézierfunktionen

Andreas Geist*; Burkhard Alpers*

* Hochschule Aalen, Fakultät Maschinenbau
Andi.Geist@gmx.de , Burkhard.Alpers@htw-aalen.de

Kurzfassung

Das Bewegungsdesign befasst sich mit der Erstellung von Sollfunktionen für bestimmte Bewegungsaufgaben. Dazu unterteilt man die Aufgabe in Abschnitte, die durch Bewegungspunkte mit vorgegebenen Ableitungen begrenzt sind. Um von einem Anfangsentwurf aus zügig zu einer guten Lösung mit ruhigen Verläufen in kurzer Zykluszeit zu kommen, ohne technologisch bedingte Restriktionen zu verletzen, ist die interaktive Manipulierbarkeit wichtig. In existierenden Softwaretools beschränkt sich diese auf die Bewegungspunkte. Im vorliegenden Beitrag wird gezeigt, wie man auch innerhalb eines Abschnitts mittels Kontrollpolygonen für Bézierfunktionen interaktiv optimieren kann.

Abstract

Motion design is concerned with designing functions for certain motion tasks. For doing this, a task is divided into sections which are separated by so-called motion points for which derivatives are provided. In order to turn an initial design efficiently into a satisfactory one with smooth graphs and a short cycle time without violating certain technologically required restrictions, it is very important to be able to manipulate the design interactively. Existing tools are restricted to changing motion points but leave the interior of a section unchanged. We show how the interior can be changed using control polygons for Bézier functions.

1 Einleitung

Beim „traditionellen“ Bewegungsdesign werden zunächst Bewegungsabschnitte mit Randbedingungen (meist Geschwindigkeit und Beschleunigung) festgelegt und diese dann mit „geeigneten“ Funktionen ausgefüllt. Sind die Randbedingungen nicht genau fixiert, kann man diese in der entsprechenden Bewegungsdesignsoftware ändern und sich das Ergebnis erneut anzeigen lassen. Um das Vorgehen für den erfahrenen Getriebetechniker effizienter zu unterstützen, fordert Braune [2-4] die Möglichkeit einer interaktiven Modifikation, mit der durch Ziehen an den Randpunkten eines Bewegungsabschnittes und simultane Anzeige der Veränderung im Verlauf von Geschwindigkeit, Beschleunigung, Ruck und Produkt aus Geschwindigkeit und Beschleunigung in kurzer Zeit ein günstiger Verlauf gefunden werden kann. Dies leistet die entsprechende Software, die bei Braune (GENESYS, siehe [2-4,12]) bzw. Berger (MOCAD, siehe [7,8]) entwickelt worden ist. Allerdings beschränkt sich die Interaktivität auf die Randpunkte der Bewegungsabschnitte (inklusive Ableitungen). Will man dazwischen ebenfalls den Verlauf interaktiv verändern, so muss man „künstlich“ in weitere Bewegungsabschnitte unterteilen. Hier setzt nun der vorliegende Beitrag an. Es wurde eine interaktive Bewegungsdesignumgebung konzipiert und implementiert, bei der neben den Randpunkten auch das Innere der Bewegungsabschnitte leicht interaktiv verändert werden kann. Dazu wird für jeden Abschnitt eine Bézierfunktion mit vorgebarem Grad und zugehörigem Kontrollpolygon gebildet. Man kann dann an den inneren Punkten des Kontrollpolygons ziehen, um den Verlauf der Bewegung zu verändern, ohne dass die Randbedingungen dabei verletzt werden.

Der zweite Abschnitt dieses Artikels gibt einen kurzen Überblick über die existierenden interaktiven Umgebungen und ihre Eigenschaften. Im dritten Abschnitt werden dann die unserem Ansatz zugrunde liegenden Bézierfunktionen und ihre Ableitungen beschrieben, wobei diese auch mit einem ähnlichen Ansatz bei Ting et al. [11] verglichen werden. Der vierte Abschnitt beschreibt das Design und User Interface der neu erstellten Umgebung. Das Potential dieser Umgebung wird dann in den folgenden Abschnitten am Beispiel der interaktiven Verbesserung einzelner Abschnitte und am Beispiel der Verkürzung der Zykluszeit beim „Schoko-Einleger“ nach Braune [2,4] demonstriert. Abschnitt 7 fasst die

Ergebnisse zusammen und gibt einen Ausblick auf weitere Entwicklungen.

2 Existierende interaktive Umgebungen

Im Folgenden sollen kurz zwei existierende Umgebungen für das interaktive Bewegungsdesign vorgestellt werden: Der im Rahmen des GENESYS-Programmpakets erstellte Teil zum Bewegungsdesign (siehe [3,12]) und die Bewegungsprofilerstellung in MOCAD (siehe [7,8]). In beiden Programmen können zunächst Bewegungspunkte festgelegt werden und damit auch Bewegungsabschnitte. Bei GENESYS sind diese Punkte Festlegungen von Weg oder Winkel über der Zeit, bei MOCAD Weg oder Winkel über Winkel (zur späteren Berechnung von Kurvengetrieben). Für die Bewegungspunkte können Angaben zu den Ableitungen 1. und 2. Ordnung gemacht werden (bei MOCAD sogar zu höheren Ordnungen). Zum „Füllen“ der Bewegungsabschnitte werden nun bei GENESYS generell Polynome 5. Grades benutzt, während bei MOCAD aus einer Fülle von Bewegungsgesetzen ausgewählt werden kann. Man kann dort auch ein Polynom höheren Grades wählen und die noch nicht genutzten Freiheitsgrade zur Glättung verwenden. Ferner besteht in MOCAD die Möglichkeit festzulegen, ob eine Randbedingung fest oder variabel ist und bis zu welcher Ordnung Stetigkeit in einem Bewegungspunkt gefordert wird.

Ist in obiger Weise ein erster Entwurf erstellt worden, so wird dieser in der 0., 1. und 2. Ableitung graphisch dargestellt. Er kann dann auf folgende Weisen interaktiv durch Mausoperationen optimiert werden:

- Einfügen/Entfernen von Bewegungspunkten
- Verschiebung von Bewegungspunkten in der 0. Ableitung horizontal (Zeit in [3], Winkel in [7]) und vertikal (Weg oder Winkel)
- Verschiebung von Bewegungspunkten in der 1. bzw. 2. Ableitung vertikal (also Modifikation der Ableitungswerte)
- Verschiebung eines Bewegungsabschnitts mit konstantem Wert bzw. 1. Ableitung bzw. 2. Ableitung durch vertikale Verschiebung eines Randpunkts und ggf. Anpassung der Stammfunktionen durch Integration.

Diese interaktiven Designoperationen kann man im Bewegungsdesign in folgender Weise sinnvoll nutzen:

- Durch Einfügen neuer Punkte kann ein Bewegungsabschnitt aufgeteilt und daher in seinem Verlauf genauer gestaltet werden. Die noch nicht sinnvoll angebbaren Ableitungswerte können dann interaktiv angepasst werden.
- Durch Verschieben kann man Taktzeiten verkürzen, so dass Restriktionen z.B. bezüglich maximaler Geschwindigkeiten oder Beschleunigungen gerade noch eingehalten werden.
- Durch Anpassen von Ableitungswerten können zunächst vorhandene Knicke in der Beschleunigung (bzw. 2. Ableitung) geglättet und damit ein „ruhigerer“ Verlauf erzeugt werden.
- Häufig sind die technologischen Anforderungen bezüglich Lage und Ableitungen gar nicht ganz genau festgelegt. Dann kann man durch Verschiebungen solche Toleranzen in einfacher Weise ausnutzen und zu einem ruhigeren Verlauf mit günstigeren Kennzahlen kommen.

Die interaktiven Möglichkeiten sind dazu gedacht, dem „erfahrenen“ Getriebetechniker die Möglichkeit zu bieten, in komfortabler und zügiger Weise einen guten Bewegungsverlauf zu erstellen. Diesen könnte man dann auch noch einer Optimierung zuführen, wie dies etwa bei Callesen [5] unter Nutzung von B-Splines zu finden ist.

Die wesentliche Beschränkung dieser Ansätze besteht darin, dass zur inneren Gestaltung eines ursprünglichen Bewegungsabschnitts gegebenenfalls zahlreiche „Zwischenbewegungspunkte“ eingeschaltet und modifiziert werden müssen (siehe etwa das Beispiel in [4]), bei denen es in höheren Ableitungen wieder zu Unstetigkeiten kommen kann.

3 Definition und Eigenschaften von Bézierfunktionen

Um Zwischenpunkte zu vermeiden, aber dennoch Gestaltungsmöglichkeiten zu haben, kann man Polynome höheren Grades nutzen (vgl. auch Nolte [10]). Wie beim interaktiven 2D-Kurvendesign, bietet sich ein Ansatz nach Bézier an. Dieser wurde bereits in der Arbeit von Ting et al. [11] aus dem Jahr 1994 verfolgt, ohne dass bei den damaligen technologischen Möglichkeiten schon ein interaktives Bewegungsdesign am Bildschirm ins Auge gefasst wurde. Zudem wird in [11] mit Bézierkurven zur Gestaltung von Übertragungsfunktionen für Kurvengetriebe gearbeitet, bei denen der Antriebswinkel und die Abtriebsgrö-

ße (Weg oder Zeit) einen „Punkt“ bilden und beide Größen in Abhängigkeit von einem Laufparameter dargestellt werden. In der vorliegenden Arbeit wird ein einfacherer Ansatz mit Bézierfunktionen vorgestellt. Diese hat auch schon Callesen in seinen Betrachtungen zur Optimierung genutzt (neben den B-Splines), dort aber nicht erkannt, dass auch bei Bézierfunktionen wie bei Bézierkurven ein Kontrollpolygon existiert, mit dem man den Verlauf leicht interaktiv gestalten kann. Im Folgenden werden diese Funktionen gemäß der Darstellung in [9] vorgestellt.

Bézierfunktionen sind Linearkombinationen von so genannten Bernsteinpolynomen. Für den Polynomgrad n gibt es $(n+1)$ solcher Polynome B_i^n , die wie folgt auf dem Intervall $[0,1]$ definiert sind:

$$B_i^n(t) = \binom{n}{i} (1-t)^{n-i} \cdot t^i; \quad i = 0, \dots, n; \quad t \in [0,1]. \quad (1)$$

Man kann die Polynome aber leicht auf ein beliebiges Zeit-Intervall $[t_1, t_2]$ umschreiben:

$$B_i^n(t) = \frac{1}{(t_2-t_1)^n} \binom{n}{i} (t_2-t)^{n-i} \cdot (t-t_1)^i; \quad i = 0..n; \quad t \in [t_1, t_2]. \quad (2)$$

Die Bildung der Bézierfunktion des Grades n erfolgt als Summe der mit dem i -ten Bernsteinpolynom vom Grad n gewichteten i -ten Ordinate b_i der Kontrollpunkte:

$$s(t) = \sum_{i=0}^n b_i \cdot B_i^n(t), \quad t \in [t_1, t_2] \quad (3)$$

Die Kontrollpunkte P_i sind dabei:

$$\left(t_1 + \frac{i \cdot (t_2 - t_1)}{n}, b_i \right), \quad i = 0, \dots, n \quad (4)$$

Die Kontrollpunkte bilden das sogenannte Kontrollpolygon, welches den Kurvenverlauf grob vorgibt. Der Funktionsgraph liegt dabei immer innerhalb der konvexen Hülle des Kontrollpolygons [9]. Das einem Kontrollpunkt b_i zugehörige Bernsteinpolynom besitzt an der Stelle t_i sein Maximum. Die Werte aller anderen Bernsteinpolynome sind an dieser Stelle geringer. Der Einfluss eines Kontrollpunktes auf die Gesamtfunktion ist somit an dessen Position im Intervall am größten und damit für den Anwender im Rahmen interaktiver Positionsveränderungen des Punktes abschätzbar und nachvollziehbar. Der Funktionsgraph besitzt überdies maximal so viele Krümmungswechsel wie das Kontrollpolygon. Abb. 1 zeigt eine Bézierfunktion 5. Grades mit Kontrollpolygon auf dem Zeitintervall $[0,1]$.

Man kann eine Bézierfunktion variieren, indem man die Kontrollpunktordinaten b_i verändert. Anschaulich entspricht dies einem Ziehen an den Kontrollpunkten in vertikaler Richtung.

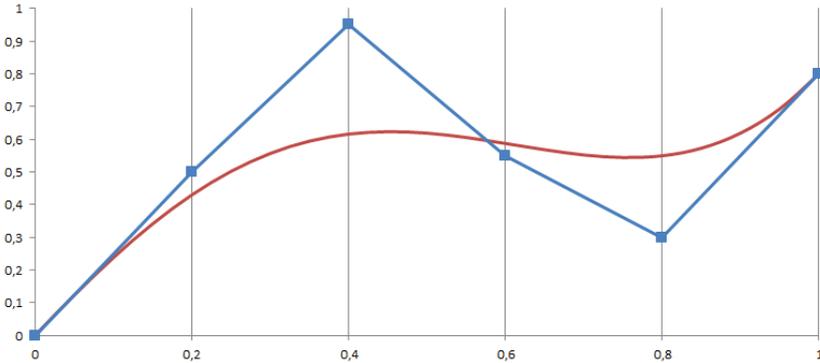


Abb. 1: Bézierfunktion 5. Grades mit Kontrollpolygon

Die Ableitungen einer Bézierfunktion lassen sich nach [9] wie folgt bestimmen:

$$\frac{d^r}{dt^r} s(t) = \frac{1}{(t_2 - t_1)^r} * \frac{n!}{(n-r)!} \sum_{i=0}^{n-r} \Delta^r b_i \cdot B_i^{n-r}(t), r \in \mathbb{N} \quad (5)$$

wobei

$$\Delta^r b_i = \sum_{j=0}^r (-1)^{r-j} \cdot \binom{r}{j} \cdot b_{i+j} \quad (6)$$

Damit ergibt sich für die Ableitungen an den Randpunkten:

$$\frac{d^r}{dt^r} s(t_1) = \frac{1}{(t_2 - t_1)^r} \cdot \frac{n!}{(n-r)!} \cdot \Delta^r b_0 \quad (7)$$

$$\frac{d^r}{dt^r} s(t_2) = \frac{1}{(t_2 - t_1)^r} \cdot \frac{n!}{(n-r)!} \cdot \Delta^r b_{n-r} \quad (8)$$

Die Formeln (7) und (8) zeigen, dass die r -te Ableitung am Rand nur von den Ordinaten b_0, \dots, b_r (linker Rand) bzw. b_{n-r}, \dots, b_n (rechter Rand) abhängt. Will man gegebene Randbedingungen also nicht verändern, so muss man die entsprechenden Kontrollpunkte sperren. Die Formeln liefern ferner Gleichungen zur Berechnung der Ordinaten bei vorgegebenen Randableitungen.

4 Interaktive Umgebung auf der Basis von Bézierfunktionen

Im erstellten Softwaretool wird dem Benutzer zunächst die Möglichkeit gegeben, erforderliche Bewegungspunkte in Form von Zeit, Weg, Geschwindigkeit und Beschleunigung vorzugeben, wie in Abb. 2 dargestellt. Später können noch neue Punkte angehängt oder bestehende Punkte gelöscht werden. Mit der Punkteingabe liegen die Bewegungsabschnitte fest. Der Benutzer kann des Weiteren spezifizieren, bis zu welcher Ableitung er Stetigkeit in den Bewegungspunkten fordert. Schließlich ist noch festzulegen, mit wie vielen Kontrollpunkten in einem Abschnitt zu interpolieren ist. Wird hier z.B. die Zahl 8 eingetragen, so wird in dem Abschnitt mit einer Bézierfunktion siebten Grades interpoliert, deren Kontrollpolygon aus 8 Punkten besteht. Dabei sind die drei linken und die drei rechten Kontrollpunkte durch die Randableitungen festgelegt, so dass nur die zwei innersten Kontrollpunkte verschoben werden können (allgemein gibt es bei n Kontrollpunkten $n-6$ frei verschiebbare Punkte).

Pkt. Nr.	Zeit t [s]	$s(t)$ [m]	$v(t)$ [m/s]	$a(t)$ [m/s ²]	Stetigkeit	Anzahl Kontrollpunkte
1	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>		
2	<input type="text" value="1"/>	<input type="text" value="2.5"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	2.Ableitung <input type="button" value="X"/>	Abschnitt 1 <input type="text" value="8"/>
3	<input type="text" value="2"/>	<input type="text" value="1.5"/>	<input type="text" value="-0.5"/>	<input type="text" value="0"/>	2.Ableitung <input type="button" value="X"/>	Abschnitt 2 <input type="text" value="8"/>
4	<input type="text" value="3"/>	<input type="text" value="1"/>	<input type="text" value="-0.5"/>	<input type="text" value="0"/>	2.Ableitung <input type="button" value="X"/>	Abschnitt 3 <input type="text" value="8"/>
5	<input type="text" value="4"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	2.Ableitung <input type="button" value="X"/>	Abschnitt 4 <input type="text" value="8"/>

Abb. 2: Eingabe von Bewegungspunkten, Randwerten, Anforderungen

Sind die Bewegungspunkte mit den zusätzlichen Angaben spezifiziert, so werden auf Knopfdruck für jeden Abschnitt die entsprechende Bézierfunktion und die Ableitungen berechnet und wie in Abb. 3 ersichtlich dargestellt. Dazu werden zunächst die durch die Randableitungen festgelegten Kontrollpunkte bestimmt. Die noch freien Kontrollpunkte im „mittleren“ Bereich werden dann durch lineare Interpolation vorläufig bestimmt. Damit steht das Kontrollpolygon für einen Abschnitt fest und die Bézierfunktion kann berechnet werden.

Sollte die Skalierung einzelner Diagramme noch ungünstig sein, so kann man diese flexibel anpassen und damit auch einen Zoom-Effekt erzielen. Dies kann auch später dazu dienen, beim interaktiven Manipulieren feiner arbeiten zu können (für weitere Details vgl. [6]).

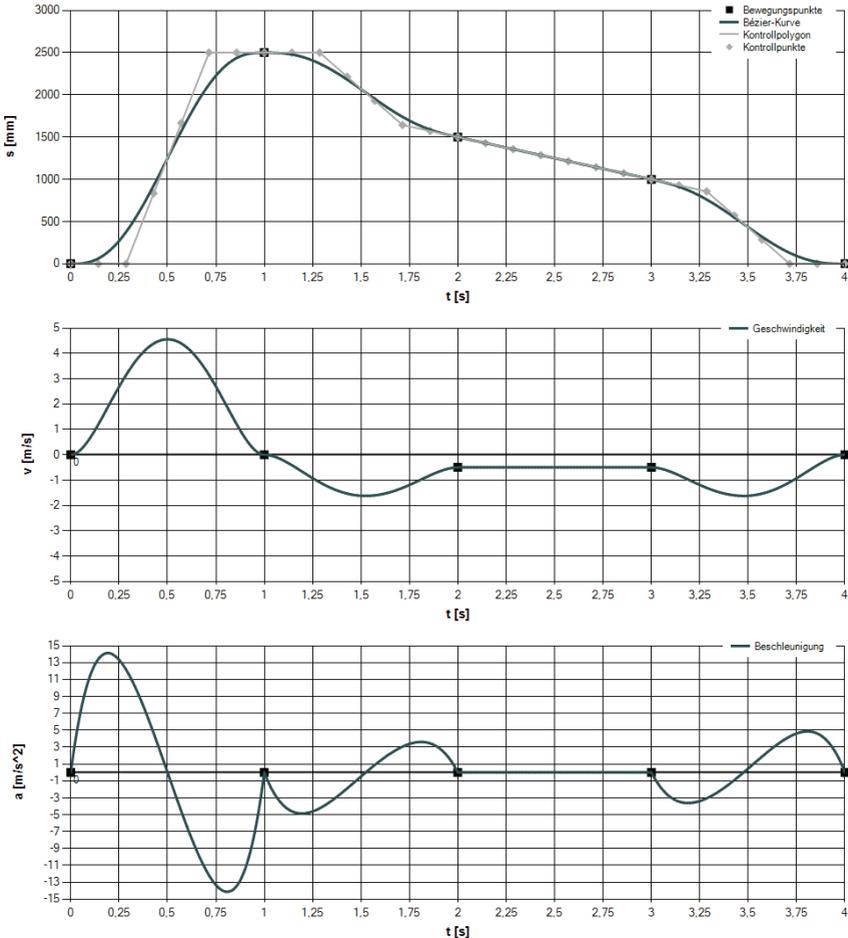


Abb. 3: Erzeugte Funktionsgraphen

Neben den in Abb. 3 dargestellten Graphen werden dem Benutzer auch noch die Ruckfunktion und das Produkt aus Geschwindigkeit und Beschleunigung angezeigt. Die Graphen bilden das zentrale Feedback-instrument des Tools. Sie können auf zwei Weisen manipuliert werden:

Die Bewegungspunkte können horizontal bewegt werden, um die Zeitintervalle der Abschnitte zu verkürzen oder zu verlängern; ferner können die freien Kontrollpunkte im $s(t)$ -Plot vertikal bewegt werden. Im Beispiel aus Abbildung 3 wären dies bei jeweils 8 Kontrollpunkten also pro Abschnitt die zwei mittleren Kontrollpunkte. Ist bei der Stetigkeitsanforderung für einen Punkt eine höhere Ableitung als die zweite angegeben, so bewirkt das vertikale Ziehen an einem Kontrollpunkt im linken angrenzenden Abschnitt eine Veränderung beim entsprechenden Kontrollpunkt im rechten angrenzenden Abschnitt und umgekehrt. Der sinnvolle Einsatz der Interaktionsmöglichkeiten soll in den folgenden zwei Abschnitten dieses Beitrags an Beispielen demonstriert werden.

Durch die lineare Interpolation bei der Bestimmung der freien Kontrollpunkte kann man auch lineare Abschnitte (mit konstanter Geschwindigkeit) erzeugen, wie man am dritten Abschnitt in Abb. 3 sieht. Will man den scharfen Knick im Beschleunigungsverlauf beim zweiten Kontrollpunkt vermeiden, so kann man auch einfach die Stetigkeit in diesem Punkt auf die 3. Ableitung setzen.

5 Beispiele zur Verbesserung eines Abschnitts

Zunächst soll eine klassische Rast-in-Rast-Aufgabe betrachtet werden, bei der man mit dem Polynom 5. Grades interpolieren kann. Will man nun das Beschleunigungsmaximum reduzieren, so empfiehlt es sich, den Beschleunigungsverlauf in Richtung „Beschleunigungtrapez“ zu verändern. Dazu wird ein Bewegungsabschnitt mit 10 Kontrollpunkten erstellt. Damit verschafft man sich 4 manipulierbare Kontrollpunkte, also 4 Freiheitsgrade. Abb. 4 zeigt zunächst die automatisch vom Tool erzeugte Bewegungsfunktion und deren Beschleunigung.

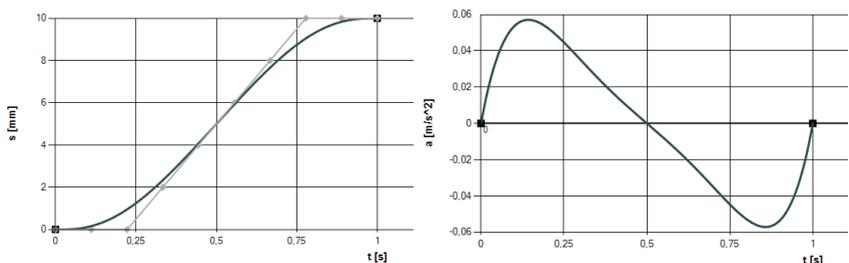


Abb. 4: Rast-in-Rast mit 10 Kontrollpunkten ($s(t)$ bzw. $a(t)$)

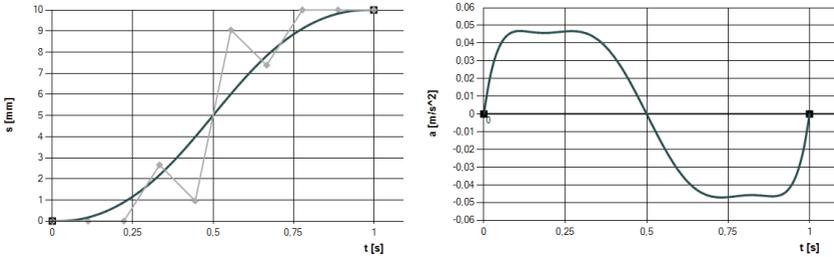


Abb. 5: Rast-in-Rast mit 10 Kontrollpunkten modifiziert

Durch weniges „Zupfen“ an den mittleren Kontrollpunkten gelangt man nun zu der in Abb. 5 dargestellten Variante, die in guter Näherung dem Beschleunigungstrapez entspricht. Eine entsprechende Bewegungsaufgabe wurde auch in der Dissertation von Callesen [5] betrachtet, der untersucht, mit welchen Ansätzen man Bewegungsfunktionen mit mathematischer Software optimieren kann. Im Ergebnis haben sich dabei B-Spline-Funktionen als am besten geeignet erwiesen. Sein Optimierungslauf für die obige Aufgabe führt zu einer Beschleunigungsfunktion, die optisch geringfügig besser als die oben gezeigte händisch erzeugte Funktion ist. Bei einem Ansatz mit 20 Kontrollpunkten konnten ähnliche Ergebnisse wie bei Callesen erzielt werden. Bei 14 freien Punkten dauert es jedoch eine Weile, bis man eine sehr gute Lösung „gezupft“ hat. Da bei Bézierfunktionen jeder Kontrollpunkt auf die gesamte Funktion wirkt, ist bei vielen Kontrollpunkten auch das Vorgehen eher experimenteller Natur, als dass man eine gezielte Strategie verfolgte.

Das zweite Beispiel zur Verbesserung eines Abschnitts betrifft das Problem des Überschwingens, für das in [1] eine rechnerische Lösung vorgestellt wurde. Die Rahmenbedingungen für das dortige Beispiel 1 lauteten: $s(0)=0$ [m], $v(0)=5,4$ [m/s], $a(0)=0$ [m/s²] und $s(1)=1$ [m], $v(1)=0$ [m/s], $a(1)=0$ [m/s²], d.h. es handelt sich um eine so genannte GR-Aufgabe (konstante Geschwindigkeit in Rast). Die automatisch erzeugte Lösung im Tool mit 10 Kontrollpunkten weist wie das Polynom 5. Grades ein Überschwingen auf, wie in Abb. 6 (links) zu erkennen ist. Durch wenige Manipulationen an den Kontrollpunkten kommt man zu der in Abb. 6 (rechts) dargestellten Lösung. Bei dieser Aufgabe wäre es natürlich sehr hilfreich, von vornherein zu wissen, mit wie vielen Kontrollpunkten man sinnvollerweise ansetzen sollte. Hierzu kann die entsprechende Berechnung aus [1] verwendet werden.

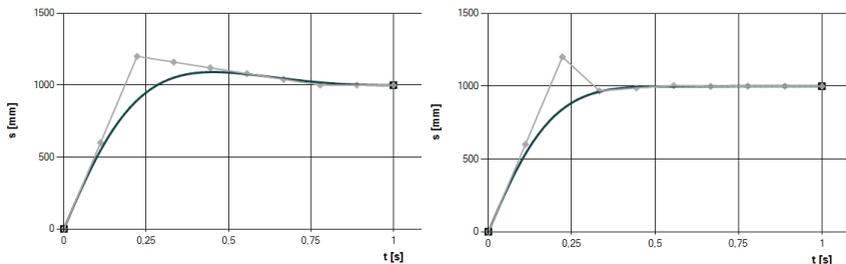


Abb. 6: GR-Aufgabe: Ausgangszustand und Modifikation

6 Beispiel mit mehreren Abschnitten: „Schoko-Einleger“

Für den Test der Programmfunktionalitäten im Rahmen einer größeren, praxisnahen Bewegungsaufgabe mit einer Reihe von Randbedingungen wird nachfolgend das von Braune eingeführte Beispiel „Schoko-Einleger“ mit dem Programm bearbeitet. Wie in Abb. 7 (Nachzeichnung von [2]) veranschaulicht, sollen in diesem Beispiel Schokoladentafeln (B) aus einem Stapelmagazin (A) in die Fächer eines kontinuierlich laufenden Transportbandes (D) befördert werden (vgl. [2,4]). Zu beschreiben ist die Bewegung des Eckpunktes (G) des Schiebers (F).

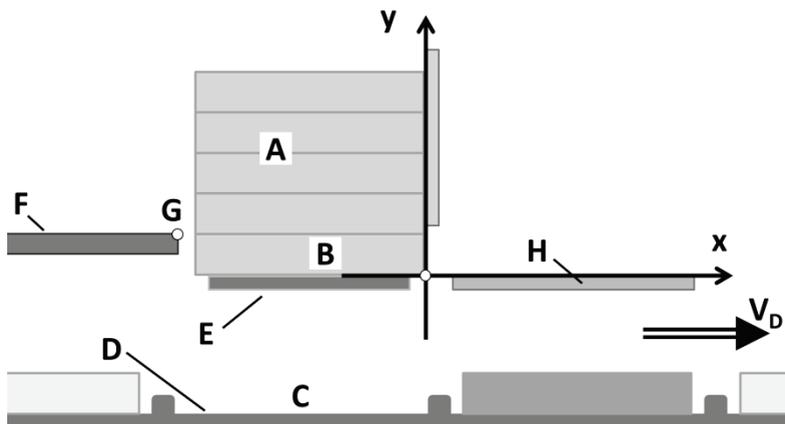


Abb. 7: Schoko-Einleger nach Braune [2, 4]

Dieser befindet sich in seiner Ausgangsposition mit einem Abstand von 5 mm hinter den 80 mm breiten und 15 mm hohen Schokoladentafeln (B) und startet aus dem Stand. Er schiebt die unterste Tafel (B) von

den Auflageschienen (E) auf die Schienen (H) und beschleunigt sie auf die Bandgeschwindigkeit v_D . Bei Erreichen der Bandgeschwindigkeit werden die Schienen (H) nach außen weggezogen und die Schokoladentafel fliegt frei in ihr Fach (C) des Transportbandes. Die Schienen (E) und (H) greifen dabei jeweils nur knapp von der Seite unter die Schokoladentafeln. Der Schieber (F) kann sich zwischen diesen hindurch bewegen.

Dabei sind folgende Restriktionen einzuhalten (diese sind teilweise direkt [2,4] entnommen, teilweise aus den dortigen Lösungen gefolgert):

- Beschleunigung in x-Richtung ist betragsmäßig $\leq 150 \text{ m/s}^2$.
- Beschleunigung in y-Richtung ist betragsmäßig $\leq 20 \text{ m/s}^2$.
- Beschleunigung in x-Richtung beim Ausschieben der Tafel ist $\leq 50 \text{ m/s}^2$.
- (Negative) Beschleunigung in y-Richtung beim Herunterfahren bis zum Absetzen des Tafelstapels ist betragsmäßig $\leq 8 \text{ m/s}^2$, um Kontaktverlust zu vermeiden.
- (Negative) Geschwindigkeit in y-Richtung beim Absetzen des Tafelstapels auf den Schienen E ist betragsmäßig $\leq 0,5 \text{ m/s}$.
- Beim Ausschieben darf der Schieber den gesamten Tafelstapel auch geringfügig anheben.
- Wenn beim Ausschieben die x-Koordinate von G gleich 0 mm geworden ist, muss die y-Koordinate kleiner gleich 15 mm sein, damit der Schieber nicht mit der Wand kollidiert. Die y-Koordinate darf auch kleiner als 15 mm werden, der Schieber soll aber bis $x=5 \text{ mm}$ weiterhin „guten“ Kontakt zur Tafel haben.
- Ist die x-Koordinate gleich 5 mm, so muss die Geschwindigkeit in x-Richtung gleich der Bandgeschwindigkeit sein, d.h. bei 1 m/s liegen. Bis dahin muss die Geschwindigkeit monoton wachsen, da die Tafel nicht abgebremst werden kann.
- Der Schieber darf in x-Richtung erst dann umkehren, wenn der Tafelstapel mit einem Sicherheitsabstand von 5 mm bereits abgelegt worden ist, d.h. wenn die y-Koordinate kleiner gleich -5 mm ist.
- Beim Wiederanheben darf die y-Koordinate erst dann den Wert -5 mm erreichen, wenn die x-Koordinate kleiner gleich -85 mm ist.

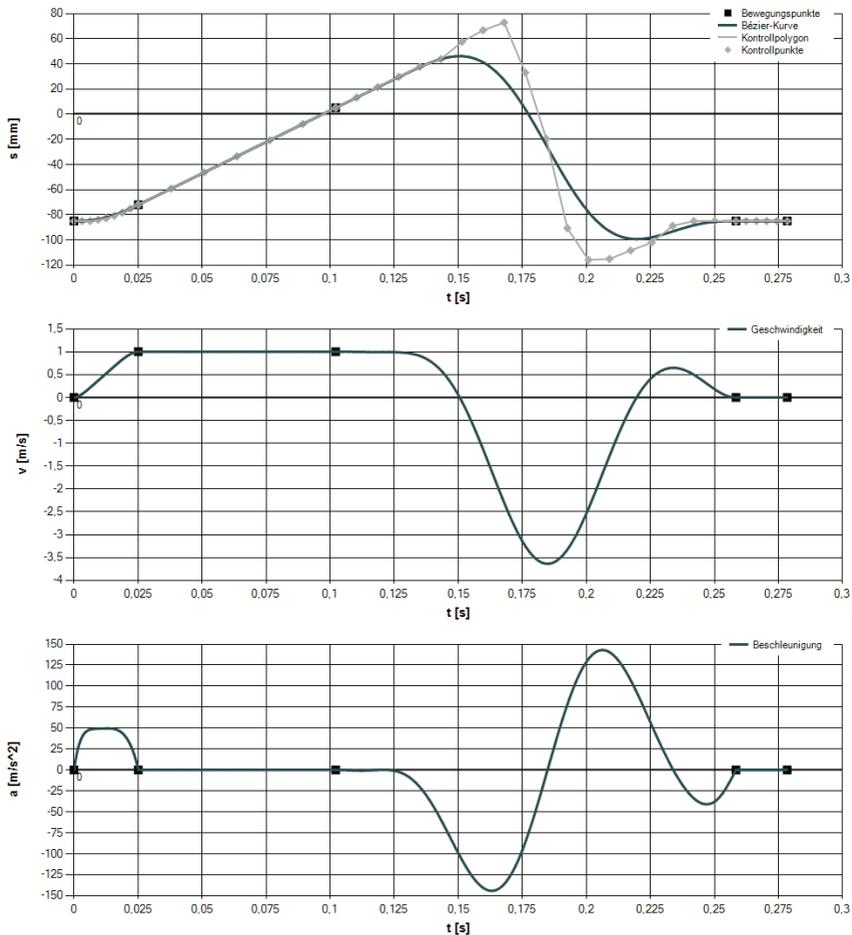


Abb. 8: Bewegungsfunktion $x(t)$ mit Ableitungen

Braune [2, 4] hat gegenüber einem ursprünglichen einfachen Entwurf, bei dem die Bewegungen in x- und y-Richtung jeweils nacheinander erfolgten und dadurch insgesamt 400ms benötigten, durch geschickte Überlagerung der Bewegungen und Nutzung von Toleranzen beim „Ausfahren“ die Zykluszeit auf 300ms reduziert. Mit dem Softwaretool auf Basis der Bézierfunktionen kann man nun zum einen die Anzahl der Bewegungspunkte und damit der Unstetigkeitsstellen im Ruck reduzieren und zum anderen innerhalb eines Bewegungsabschnitts das Beschleunigungspotential besser auszunutzen und dadurch eine

Abschnittsverkürzung erreichen. Das Vorgehen gestaltet sich dabei so, dass man in einem iterativen Prozess zunächst das Abschnittsinnere modifiziert und dann den Abschnitt verkürzt. Da dabei auch die Gestalt wieder ungünstiger werden kann, muss man ggf. wiederum das Kontrollpolygon modifizieren. Mit diesem Verfahren sind die in Abb. 8 bzw. Abb. 9 gezeigten Verläufe entstanden, die eine Zykluszeit von 279ms benötigen und damit eine weitere Verringerung um ca. 7% realisieren.

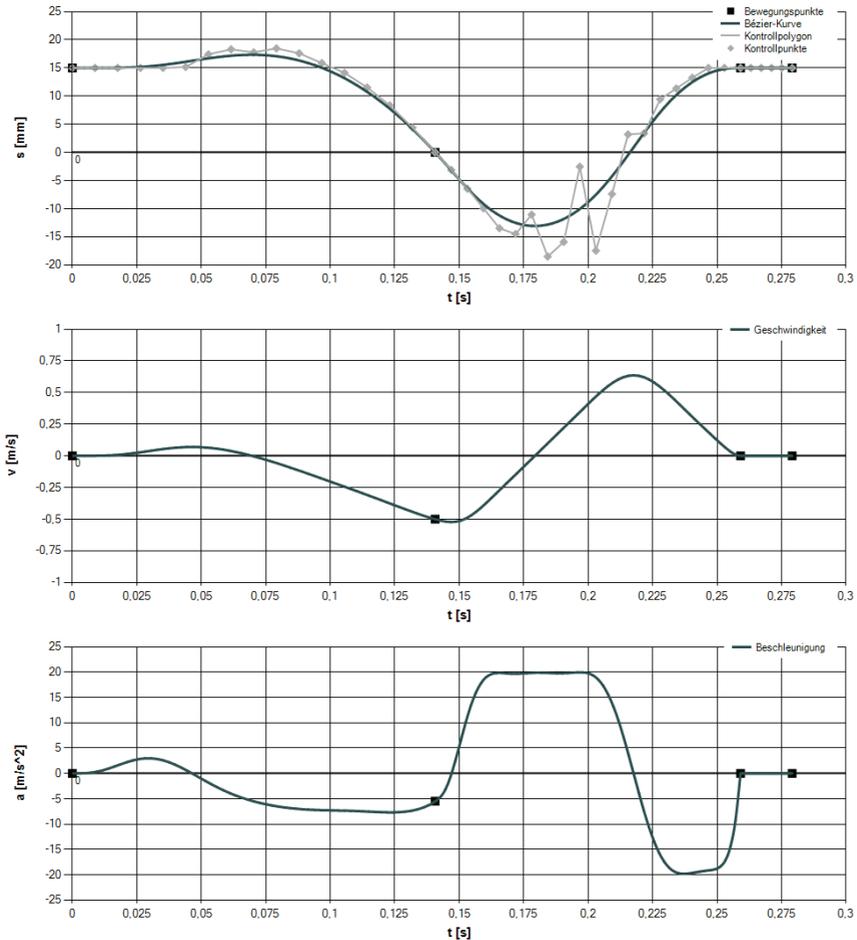


Abb. 9: Bewegungsfunktion $y(t)$ mit Ableitungen

Dabei sind in x-Richtung nur fünf und in y-Richtung nur vier Bewegungspunkte verwendet worden. Man benötigt dann allerdings für die „langen“ Abschnitte bis zu 20 Kontrollpunkte, um im Inneren so modellieren zu können, dass man die Restriktionen einhält und einen guten Verlauf mit kurzer Zykluszeit erhält. Dies erfordert zudem ein aufwändigeres Experimentieren beim Verschieben der zahlreichen Kontrollpunkte und es ist nicht einfach, dabei alle relevanten Restriktionen im Blick zu behalten. Letzteres ist einfacher, wenn man mehr Bewegungspunkte einsetzt, die an kritischen Stellen die Einhaltung der Restriktionen „markieren“ (z.B. bei einem festgelegten Absetzpunkt die Geschwindigkeit von $-0,5$ m/s in y-Richtung). Dabei erzeugt man dann aber wiederum mehr Punkte mit Unstetigkeit im Ruck, wenn man an den Übergängen nur die Stetigkeit bis zur zweiten Ableitung fordert. Es wurde hierfür ebenfalls ein Beispiel erstellt, das ungefähr dieselbe Zykluszeit benötigt, aber einen etwas unruhigeren Verlauf aufweist.

7 Zusammenfassung und Ausblick

In diesem Beitrag wird gezeigt, wie man Bézierfunktionen und ihre Kontrollpolygone nutzen kann, um ein Softwaretool für das interaktive Bewegungsdesign zu erstellen, mit dem man auch innerhalb von Bewegungsabschnitten den Verlauf beeinflussen und damit günstig gestalten kann. Neben einer geschickten Ausnutzung von Toleranzen und von möglichen Bewegungsüberlappungen, wie man sie bereits in den Beiträgen von Braune [2-4] findet, bietet dies eine weitere Möglichkeit, den Gestaltungsspielraum beim Bewegungsdesign auszunutzen, um eine möglichst kurze Zykluszeit bei „ruhigem“ Verlauf der Bewegungsfunktion und ihrer Ableitungen zu erreichen. Man kann die Einführung „künstlicher“ Zwischenpunkte vermeiden, bei denen die Stetigkeit meist nur bis zur zweiten Ableitung gewährleistet ist. Je mehr Kontrollpunkte man zulässt, desto feiner lässt sich zwar der Funktions- und Ableitungsverlauf gestalten, aber der Aufwand steigt mit den Freiheitsgraden ebenfalls enorm. Will man zudem den Beschleunigungsverlauf beeinflussen, so ist es sehr schwierig, zielgerichtet zu variieren, da man durch Verschiebung eines Punkte Nebenwirkungen auf den gesamten Bewegungsabschnitt hat. Dann resultiert die Interaktion eher in einem „Trial-and-Error“-Verfahren, das vielleicht nach einiger Zeit zu einem guten Zustand führt. Es wird daher empfohlen, sich in der Regel

auf wenige frei verschiebbare Punkte zu beschränken. An Beispielen wurde gezeigt, dass man durch interaktives Vorgehen in kurzer Zeit zu ähnlichen Ergebnissen gelangen kann wie mit mathematischer Optimierung bei Callesen [5]. Ferner konnte beim „Schoko-Einleger“-Beispiel von Braune die Zykluszeit und die Anzahl erforderlicher Bewegungsabschnitte (und damit von Bewegungspunkten mit geringeren Stetigkeitseigenschaften) verringert werden, wobei dann allerdings eine Vielzahl von Kontrollpunkten erforderlich und ein aufwändiges Experimentieren unvermeidlich ist.

Für die weitere Verbesserung der Umgebung bieten sich mehrere Richtungen an: Zum einen kann man versuchen, wie Callesen [5] mit B-Spline-Funktionen zu arbeiten, die die lokale Veränderbarkeit erlauben. Ferner wäre auch zu untersuchen, ob man nicht direkt die Beschleunigungsfunktion über Kontrollpolygone beeinflussen kann. Dabei würden jedoch auch die Randwerte von Weg- und Geschwindigkeitsfunktion verändert, so dass man hier Strategien zur Erhaltung dieser Werte entwickeln muss. Schließlich würde es dem Benutzer die Arbeit deutlich erleichtern, wenn er wie bei MOCAD [7] Templates für gewisse Aufgabenstellungen nutzen könnte. Man kann die Ergebnisse der interaktiven Arbeit zudem auch als Startlösung für Optimierungswerkzeuge verwenden, wie sie Callesen [5] implementiert hat.

Literatur

- [1] Alpers, B. „Zur Vermeidung des kinematischen Schwingens beim Bewegungsdesign“. In: *Bewegungstechnik 2014*. VDI-Berichte 2237. Düsseldorf: VDI-Verlag, 2014, S. 51-62.
- [2] Braune, R. „Bewegungsdesign – Eine Kernkompetenz des Getriebetechnikers“. In: *Kurvengetriebe, Koppelgetriebe, gesteuerte Antriebe. Problemlösungen in der Bewegungstechnik*. VDI-Getriebetagung 2000. VDI-Berichte 1567. Düsseldorf: VDI-Verlag, 2000, S. 1-23.
- [3] Braune, R. „Bewegungsdesign mit interaktiver Variation von Stützwerten – Intuitive Bewegungsoptimierung durch Ausnutzung zugelassener Gestaltungsspielräume“. In: *Bewegungstechnik 2006*. VDI-Berichte 1966. Düsseldorf 2006, S. 47-76.
- [4] Braune, R. „Leistungspotential Bewegungsdesign – Schnellere Maschinen durch bessere Bewegungsführung“. In: *Tagungsband*

- „Verarbeitungsmaschinen und Verpackungstechnik“ VVD 2009, TU Dresden 2009, S. 373-395.
- [5] Callesen, M. Der Antrieb ungleichmäßig übersetzender Getriebe durch Motion-Control-Systeme. VDI Fortschritt-Berichte 1146. Düsseldorf: VDI Verlag 2008.
- [6] Geist, A. Erstellung einer interaktiven Umgebung für das Bewegungsdesign. Bachelorarbeit Hochschule Aalen. Aalen 2015.
- [7] Heine, A., Matthes, J., Schulz, R., Berger, M. „MOCAD–Ein grafisch interaktives Werkzeug zur Berechnung und Optimierung von Kurvengetrieben und Motion Control Systemen“. In: *Bewegungstechnik 2010*. VDI-Berichte 2116. Düsseldorf: VDI-Verlag, 2010, S. 97-110.
- [8] Heine, A.; Berger, M. „Grafisch-interaktives Bewegungsdesign als effizientes Hilfsmittel zum Vergleich von Kurvengetrieben und Motion-Control-Systemen“. *Tagungsband „Verarbeitungsmaschinen und Verpackungstechnik“ VVD 2012*. TU Dresden 2012.
- [9] Hoschek, J., Lasser, D. Grundlagen der geometrischen Datenverarbeitung. 2. Auflage, Stuttgart: B.G. Teubner, 1992.
- [10] Nolte, R. „Bewegungsdesign mit Polynomsplines“. *Bewegungstechnik 2010*. VDI-Berichte 2116. Düsseldorf: VDI-Verlag, 2010, S. 77-86.
- [11] Ting, K.-L., Lee, N.L., Brandan, G.H. “Synthesis of polynomial and other curves with the Bézier technique”. *Mech. Mach. Theory* 29 (1994), pp. 887-903.
- [12] Vujevic, A. Bewegungsdesign und Geometrieverarbeitung in einem getriebetechnischen Softwaresystem. VDI-Fortschrittberichte. Reihe 20, Nr. 414. Düsseldorf: VDI-Verlag, 2008.