

Tools

Andreas Seefried* und Andreas Pfeiffer

Trajektorien-generator zur Erzeugung beschränkter B-Splines

Method for generating implicitly constrained B-splines

<https://doi.org/10.1515/auto-2019-0067>

Empfangen 12. Juni 2019; angenommen 4. November 2019

Zusammenfassung: Der Beitrag stellt einen Algorithmus zur Generierung von mehrfach stetig differenzierbaren Trajektorien auf Basis von B-Splines vor, deren Funktionswerte und Ableitungen innerhalb von konstanten Grenzen liegen. Die Trajektorien können für Systeme genutzt werden, bei denen Grenzen in den Ableitungen des Systemeingangs vorliegen. Dies trifft beispielsweise auf Modelle zu, bei denen ein Antrieb mit Begrenzungen in der Position, Geschwindigkeit und Beschleunigung als Systemeingang genutzt wird.

Zur Generierung werden Parameter, Startwerte der Trajektorie sowie die konstanten Grenzen benötigt. Eine quadratische Optimierung liefert die Stützpunkte der B-Spline-Kurve, die dadurch definiert ist. Die implizit begrenzten B-Splines können ideal in Optimalsteuerungsproblemen und modellprädiktiven Regelungen als Steuertrajektorie eingesetzt werden. Durch die implizite Berücksichtigung von Nebenbedingungen können ebensolche aus dem eigentlichen Optimierungsproblem entfallen. Daraus resultiert eine Verringerung der Komplexität des Optimalsteuerungsproblems bzw. der modellprädiktiven Regelung was wiederum zu einer schnelleren Berechnungszeit führt.

Die implizit begrenzten B-Splines werden im Beitrag vorgestellt und exemplarisch in einer modellprädiktiven Regelung mit einem robotischen System angewandt. Sie führen zu einer deutlich schnelleren Berechnungszeit im Vergleich zur Nutzung von B-Splines mit expliziten Nebenbedingungen im Optimierungsproblem.

Schlagwörter: Trajektorien, Nebenbedingungen, gleitender Horizont, Optimalsteuerungsproblem

*Korrespondenzautor: **Andreas Seefried**, Institut für Systemdynamik und Regelungstechnik, DLR Oberpfaffenhofen, Wessling, Deutschland, E-Mail: andreas.seefried@dlr.de

Andreas Pfeiffer, Institut für Systemdynamik und Regelungstechnik, DLR Oberpfaffenhofen, Wessling, Deutschland, E-Mail: andreas.pfeiffer@dlr.de

Abstract: In this paper, a novel approach for generating continuously differentiable trajectories based on B-Splines with the implicit consideration of constraints is presented. The control points of the B-spline are evaluated by a set of parameters and a quadratic programming problem. The approach can be used for systems that have inputs with constant constraints and constant constraints in their first and second derivative.

For optimal control problems or model predictive control, the presented B-splines implicitly fulfill constraints that can be removed from the optimization problem. Thus, the complexity and computational effort of the superordinate optimization problem is reduced. The paper presents the method, examples of generated trajectories and an application of these trajectories in a model predictive control for a robot based motion simulator.

Keywords: B-splines, constraints, optimization, moving horizon, MPC

1 Einleitung

Zur Optimierung von Systemen über einen zeitlichen Horizont, wie beispielsweise Optimalsteuerungsprobleme (*optimal control problem*, OCP) oder modellprädiktive Regelung (*model predictive control*, MPC), werden zeitabhängige Steuertrajektorien $u(t)$ benötigt. Der Grundgedanke beider Verfahren ist sehr ähnlich: finde eine Trajektorie, die das gegebene System im Sinne eines gegebenen Gütekriteriums optimal steuert.

Besonders im Bereich von modellprädiktiver Regelung werden diese Steuertrajektorien häufig aus stückweise konstanten Trajektorien aufgebaut [6, 13]. Im Fall von kleinen Zeitschritten bzw. großen zeitlichen Horizonten – bei MPC sog. *Prädiktionshorizont*, bei OCP die Länge der Integration – resultiert mit dieser Methode eine große Anzahl an *Optimierungsvariablen* (OV), die in das Optimierungsproblem (OP) der MPC bzw. OCP eingehen. Diese große Anzahl an OV führt im Allgemeinen zu einer Erhöhung der Komplexität und Lösungsdauer des OP. Um die Anzahl

der OV zu reduzieren und gleichzeitig die zeitliche Länge der Trajektorie nicht zu kürzen, wird beispielsweise in [1] eine *move blocking*-Strategie verwendet, bei der einzelne OV für mehrere Zeitschritte genutzt werden. Um die Anzahl an OV weiter zu reduzieren, können auch Splines genutzt werden [11]. Die OV selbst stellen dann Stützpunkte der Spline-Funktion dar, die zu jedem Zeitschritt ausgewertet wird, s. Abb. 1. Dies führt zu einer Reduzierung der Komplexität des zu lösenden OP.

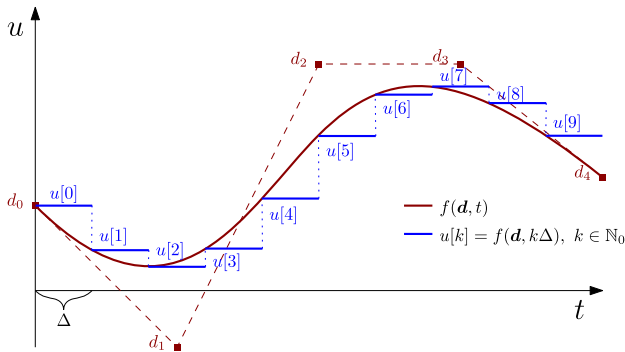


Abb. 1: Vergleich einer Spline-Funktion $f(\mathbf{d}, t)$ aus der zeitdiskrete Werte $u[k] = f(\mathbf{d}, k\Delta)$ resultieren. Durch die Nutzung von wenigen Stützpunkten \mathbf{d} der Spline-Kurve (hier 5) kann ein Signal generiert werden, das in einer zeitdiskreten Abbildung eine deutlich höhere Anzahl an Elementen (hier 10) notwendig machen würde.

Für eine bestimmte Klasse von Systemen können diese Spline-Funktionen weiter optimiert werden. Systeme bzw. Systemkomponenten, die einen doppelten Integrator darstellen, wie beispielsweise Robotergelenke oder Linearantriebe, besitzen in der Regel konstante Stellgrößenbegrenzungen in der Position, Geschwindigkeit und in den Drehmomenten. Mit dem hier vorgestellten Ansatz lassen sich Trajektorien generieren, die eine definierbare Startposition und -geschwindigkeit besitzen und neben den Nebenbedingungen für die Position und Geschwindigkeit auch konstante Grenzen für die Beschleunigung einhalten.

Sofern Systeme mit Drehmomentbegrenzung zum Einsatz kommen, müssen konstante Grenzen für die Beschleunigung gefunden werden, um die vorgestellten Trajektorien verwenden zu können. Ein Beispiel für solch ein System ist der *DLR Robotic Motion Simulator*, s. Abb. 2, der im Kapitel 5 in einer MPC Anwendung und den implizit beschränkten B-Splines angesteuert wird.

B-Spline Kurven [3] werden häufig für die Definition von Trajektorien eingesetzt. Sie kommen im Bereich für die Beschreibung und Optimierung von Konturen und Oberflächen vor [9]. Ebenso werden sie für die Optimierung von

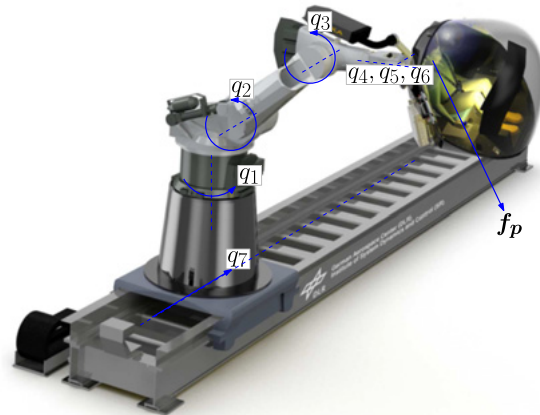


Abb. 2: Der *DLR Robotic Motion Simulator* ist ein Bewegungssimulator auf Basis eines Industrieroboters, der mit den vorgestellten Trajektorien angesteuert wird. Die Gelenkwinkel werden mit q_1 bis q_6 bezeichnet, die Linearachse mit q_7 . Der Fahrer sitzt in der Gondel des Simulators, die am Endeffektor angebracht ist, und erfährt dort die auf ihn einwirkenden Kräfte \mathbf{f}_p die sich aus der Gravitation und der Bewegung des Simulators zusammensetzen.

dynamischen Systemen wie Roboter eingesetzt [5]. Um Nebenbedingungen einzuhalten, werden aber meist die Trajektorien in ihrer Gänze ausgewertet, was ein Mehraufwand im Vergleich zur Auswertung der Stützstellen bedeutet. In [5] wird das OP effizient umgeformt, allerdings besteht die Möglichkeit, dass keine Lösung des Problems existiert.

In [4] werden Trajektorien auf Basis von B-Splines erzeugt, um sie als Hilfsmittel bspw. für die Lösung von einem OCP zu nutzen. Dafür wird ein Satz an gewünschten Kontrollpunkten mit Toleranzen vorgegeben sowie die Grenzen, die eingehalten werden sollen. Der Algorithmus findet eine Lösung, sodass die Kontrollpunkte der Trajektorie von den gewünschten Kontrollpunkten im quadratischen Mittel am wenigsten entfernt sind. Es ist bei dieser Methode allerdings möglich, dass mit den vorgegeben Kontrollpunkten, Toleranzen und Grenzen keine Lösung existiert. Auch ist es möglich, dass zwei verschiedene Optimierungsläufe zum selben Ergebnis kommen. Dies ist dann von Nachteil, wenn die Trajektorien bspw. in OCP oder MPC eingesetzt werden sollen, denn dann führt eine Veränderung der OV nicht zu einer Änderung des Ergebnisses. Aus Sicht des OP scheint es dann keinen Einfluss der OV auf die Zielfunktion zu geben.

Eine ähnliche Methode wie die hier gezeigte wird in [10] vorgestellt. B-Splines werden genutzt, um flache Systeme in einem OCP zu lösen. Im Unterschied dazu dient die Methode dieses Beitrags dazu, ganz allgemein Trajektorien zu erzeugen, die dann auf ein System angewendet werden. Beide Verfahren haben gemeinsam, dass Nebenbe-

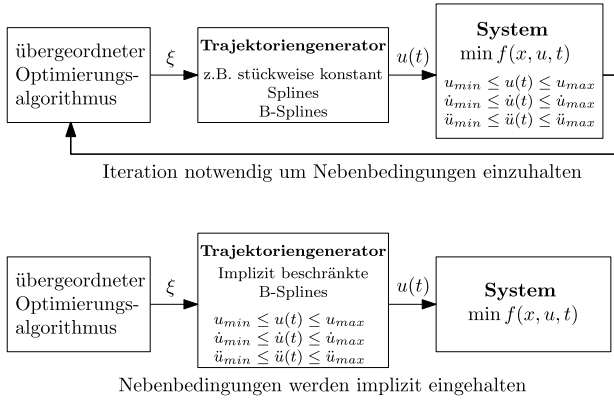


Abb. 3: Oben: Klassischerweise generiert ein Optimierungsalgorithmus einen Satz von OV ξ , aus dem ein Steuereingang $u(t)$ generiert wird. Beschränkungen im System, die allein vom Eingang abhängig sind, werden dennoch erst nach Evaluierung des Systems ausgewertet. Unten: Mit den implizit beschränkten B-Splines verlagern sich die Beschränkungen in den Prozess der Trajektoriengenerierung.

dingungen implizit eingehalten werden. Der wichtige Unterschied ist, dass in [10] das System Teil des OP ist und die Lösung desselbigen auch immer die Systemgleichungen in Betracht zieht.

Die hier vorgestellten implizit beschränkten B-Splines werden über die Lösung eines Optimierungsproblems erzeugt. Mit der *übergeordneten Optimierung* oder dem *übergeordneten Optimierungsproblem* ist die Anwendung gemeint, die mit den Trajektorien arbeitet, im Beispielkapitel 5 die MPC-Anwendung am Bewegungssimulator. Des Weiteren wird ein Parametervektor ξ vorgestellt, der die implizit beschränkten B-Splines parametrisiert. Dieser Parametervektor wird im Anwendungsbeispiel durch die OV des *übergeordneten Optimierungsproblems* gesetzt.

Die Berechnung der implizit beschränkten B-Splines kostet etwas Zeit. Für komplexe Systeme bzw. OP mit einem langen Zeithorizont ist das nicht nachteilig. Aufgrund der Verlagerung der Nebenbedingungen bzw. einem Teil davon in die Trajektorien, reduzieren sich die Iterationen aufgrund von verletzten Nebenbedingungen, s. Abb. 3. Die vorgestellte Methode ist also besonders für übergeordnete Optimierungen mit verhältnismäßig langer Berechnungszeit ein interessanter Ansatz.

1.1 Problemstellung

Für die hier vorgestellte Trajektorie ergibt sich folgende Problemstellung: Gesucht wird eine zeitabhängige Trajek-

torie $u(t)$, $t \in [t_0, t_e]$, $t_0 < t_e$ mit

$$u(t_0) = u_0 \quad (1)$$

$$\dot{u}(t_0) = \dot{u}_0 \quad (2)$$

$$u_{\min} \leq u(t) \leq u_{\max} \quad (3)$$

$$\dot{u}_{\min} \leq \dot{u}(t) \leq \dot{u}_{\max} \quad (4)$$

$$\ddot{u}_{\min} \leq \ddot{u}(t) \leq \ddot{u}_{\max}. \quad (5)$$

u_0 und \dot{u}_0 stellen die Startposition und -geschwindigkeit dar. Die konstanten Grenzen aus (3) bis (5) können frei vorgegeben werden. Die Trajektorie soll zudem zweifach stetig differenzierbar sein und über einen Parametervektor $\xi \in [0, 1]^p$ parametrisiert werden.

Es folgt eine Einführung zu B-Splines, die die Grundlage für die Trajektorien bilden. In Abschnitt 3 wird das OP zur Generierung der Trajektorie vorgestellt und mit Beispielen präsentiert. Vor einer kurzen Diskussion über die Vor- und Nachteile folgt noch die Anwendung der implizit begrenzten B-Splines am Bewegungssimulator RMS.

2 B-Splines

Die Grundlage für die zu generierenden Trajektorien stellen B-Splines [3, 12, 8] dar. Sie werden durch m Basis-Funktionen k -ter Ordnung B_j^k , $j = 0, 1, \dots, m-1$, $k > 0$, $m > 0$ gebildet.

2.1 Basisfunktion 'B-Spline'

Eine Basis-Funktion erster Ordnung $B_j^1(t)$, $j = 0, 1, \dots, n-1$ ist innerhalb eines Intervalls $[\tau_j, \tau_{j+1})$ gleich 1, sonst 0:

$$B_j^1(t) := \begin{cases} 1, & \tau_j \leq t < \tau_{j+1} \\ 0, & \text{sonst.} \end{cases} \quad (6)$$

Die Knotenpunkte τ_j bilden den *Knotenvektor* $\tau \in \mathbb{R}^{n+1}$ mit folgenden Eigenschaften:

$$\tau_0 \leq \tau_1 \leq \dots \leq \tau_{n-1} \leq \tau_n. \quad (7)$$

Der Knotenvektor besitzt an den Rändern eine Vielfachheit von k , also

$$\tau_0 = \dots = \tau_{k-1} \quad (8)$$

$$\tau_{n-k+1} = \dots = \tau_n. \quad (9)$$

Es wird für τ zusätzlich gefordert, dass

$$\tau_j < \tau_{j+k} \quad \text{für } j = 0, \dots, n-k \quad (10)$$

damit die Stetigkeit innerhalb der B-Spline nicht reduziert wird. Durch einen rekursiven Algorithmus lassen sich Basis-Funktionen der Ordnung $k > 1, j = 0, 1, \dots, n - k$ bilden,

$$B_j^k(t) = \frac{t - \tau_j}{\tau_{j+k-1} - \tau_j} B_j^{k-1}(t) + \frac{\tau_{j+k} - t}{\tau_{j+k} - \tau_{j+1}} B_{j+1}^{k-1}(t). \quad (11)$$

Zwei weitere, für diese Arbeit wichtige, Eigenschaften von B-Splines sind, s. [3, S. 109f]:

$$\sum_{j=0}^{n-k} B_j^k(t) = 1 \quad \forall t \in [\tau_0, \tau_n] \quad (12)$$

$$B_j^k(t) = 0 \quad \forall t \notin [\tau_j, \tau_{j+k}] \quad (13)$$

Zum Einen ist also die Summe von B-Splines innerhalb des Knotenvektors an jedem Punkt 1, zum anderen ist jede B-Spline außerhalb ihres sog. *Trägerintervalls* gleich 0.

2.2 B-Spline-Kurve

Aus den B-Splines lässt sich eine B-Spline-Kurve erzeugen. Mit Hilfe von m Stützpunkten d_j^0 und m B-Splines der Ordnung k kann der Verlauf für die Position

$$u(t) = \sum_{j=0}^{m-1} d_j^0 B_j^k(t) \quad (14)$$

ausgewertet werden. Die Stützpunkte d_j^0 , auch *de Boor-Punkte*, bilden den Vektor $\mathbf{d}^0 \in \mathbb{R}^m$. Aufgrund der Eigenschaft aus (12) lässt sich zeigen, dass der Wert einer B-Spline-Kurve mit $t \in [\tau_j, \tau_{j+1})$ innerhalb der konvexen Hülle der Stützpunkte d_j^0, \dots, d_{j+k}^0 liegt [3, S. 155], s. Abb. 4. Diese Aussage lässt sich auf den gesamten Verlauf anwenden,

$$\min(\mathbf{d}^0) \leq u(t) \leq \max(\mathbf{d}^0). \quad (15)$$

2.2.1 Ableitungen der B-Spline-Kurve

Die i -te zeitliche Ableitung der B-Spline-Kurve lässt sich durch einen neuen Stützpunktvektor \mathbf{d}^i in Kombination mit B-Splines reduzierter Ordnung auswerten:

$$u^{(i)}(t) = \sum_{j=0}^{m-i-1} d_j^i B_j^{k-i}(t) \quad (16)$$

mit

$$d_j^i = (k-i) \frac{d_{j+1}^{i-1} - d_j^{i-1}}{\tau_{j+k} - \tau_{j+i}} \quad j = 0, 1, \dots, m-i-1. \quad (17)$$

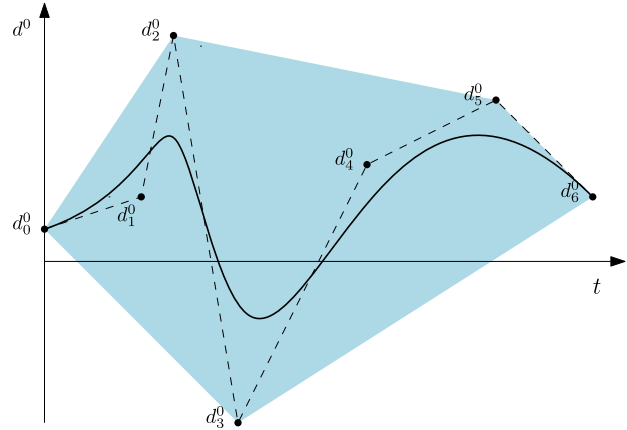


Abb. 4: Konvexe Hülle eines Polygonzuges \mathbf{d}^0 mit zugehöriger B-Spline-Kurve. Der Verlauf $u(t)$ liegt immer innerhalb der konvexen Hülle (blau dargestellt).

Die Differenz $\tau_{j+k} - \tau_{j+i}$ wird im Folgenden mit

$${}_j\Theta_i^k := \tau_{j+k} - \tau_{j+i} \quad (18)$$

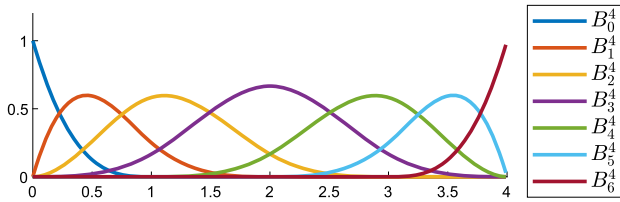
verkürzt geschrieben.

Die Ableitungen stellen also wiederum B-Spline-Kurven dar. Auch diese besitzen die Eigenschaft, dass die Werte innerhalb der konvexen Hülle der Stützpunkte liegen. Somit kann (15) allgemein ausgedrückt werden mit

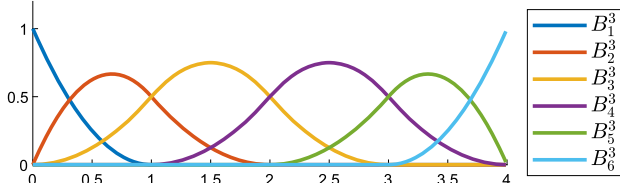
$$\min(\mathbf{d}^i) \leq u^{(i)}(t) \leq \max(\mathbf{d}^i). \quad (19)$$

Zur Veranschaulichung von B-Splines und ihren Ableitungen folgen nun die Abb. 5a bis 5d. Abbildung 5a zeigt den Verlauf von $B_j^4(t)$ mit $m = 7$ und dem Knotenvektor $\boldsymbol{\tau} = [0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4]^T$. Mit $k = 4$ werden die Basisfunktion B_0^4 bis B_6^4 gebildet. Die Abb. 5b bis 5d zeigen die B-Splines geringerer Ordnung bei gleichbleibendem Knotenvektor.

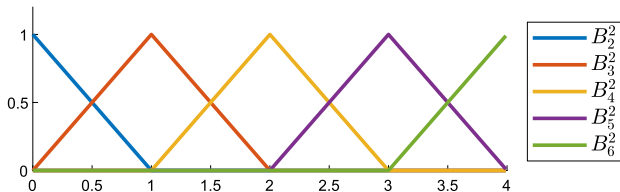
Die B-Splines haben je nach Knotenvektor ihr Maximum an einer anderen Position und das Maximum ist je nach Ordnung ≤ 1 , was in den Abbildungen gut sichtbar ist. In weiteren Abbildungen werden zur besseren Verständlichkeit den Stützpunkten d_j^i feste Positionen zugeordnet. Diese Positionen entsprechen den Abszissenwerten der Maxima der jeweiligen B-Spline und dem Stützpunktwert als Ordinatenwert. Als Beispiel dient eine B-Spline-Kurve der Ordnung $k = 4$ mit dem Knotenvektor $\boldsymbol{\tau} = [0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4]^T$ und den $m = 7$ Stützpunkten $\mathbf{d}^0 = [-10, -10, -4, 10, 10, 10, 10]^T$, s. Abb. 6. Der Verlauf $u(t)$ stellt die Summe der B-Splines dar, vgl. (14).



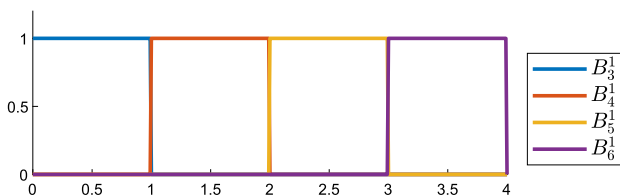
(a) B-Splines der Ordnung 4.



(b) B-Splines der Ordnung 3. Aufgrund der Vielfachheit der Randknotenpunkte sind B_0^3 und B_5^3 gleich 0 und werden nicht abgebildet.



(c) B-Splines der Ordnung 2. Die ersten und letzten 2 B-Splines sind wegen der Vielfachheit der Randknotenpunkte 0 und werden nicht abgebildet.



(d) B-Splines der Ordnung 1. Die ersten und letzten 3 B-Splines sind wegen der Vielfachheit der Randknotenpunkte 0 und werden nicht abgebildet.

Abb. 5: B-Splines verschiedener Ordnung k bei gleich bleibendem Knotenvektor τ .

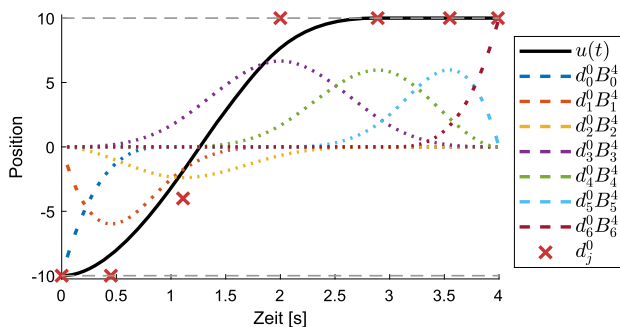


Abb. 6: B-Spline-Kurve mit Stützpunkten d^0 die nach Gleichung (14) ausgewertet ist. Die roten Marker sind durch die jeweiligen Abszissenwerte der Maxima der B-Splines und dem Stützpunktwert definiert.

3 Definition des Optimierungsproblems

Für die Parametrierung einer Trajektorie ist es grundsätzlich möglich, die einzelnen Stützpunkte d_j^0 als freie Größen vorzugeben und den Knotenvektor τ als unveränderlichen Parameter festzulegen. Bei diesem einfachen Ansatz können allerdings nur die Beschränkungen der Trajektorie selbst (siehe Bedingung (3)) direkt eingehalten werden. Die Bedingungen (4) und (5) der Ableitungen werden bei freier Wahl der Stützpunkte im Allgemeinen verletzt.

Eine weitere Idee wäre, die Stützpunkte als unveränderliche Parameter zu setzen und den Knotenvektor τ als freie Größe vorzugeben. In diesem Fall müssen zahlreiche Nebenbedingungen aufgestellt werden, um die Bedingungen (1)–(5) der resultierenden Trajektorie zu erfüllen. Es zeigt sich jedoch bei genauerer Betrachtung, dass bei bestimmten Randbedingungen von Trajektorien keine Parametrierung über den Knotenvektor möglich ist.

Aus diesem Grund werden in der vorliegenden Arbeit die Trajektorien mit einem alternativen Ansatz parametrisiert. Die ersten beiden Stützpunkte d_0^0 und d_1^0 sind durch die Startbedingungen definiert. Aufgrund der Vielfachheit der Randknoten (8), (9) und (17) ergeben sich d_0^0 und d_1^0 zu

$$d_0^0 = u_0 \tag{20}$$

$$d_1^0 = \dot{u}_0 \tag{21}$$

$$\Rightarrow d_1^0 = \frac{\dot{u}_0}{k-1} \Theta_1^k + u_0. \tag{22}$$

Die restlichen $p := m - 2$ Stützpunkte werden mit einer Linearkombination

$$d_j^0 = \xi_j d_j^{0+} + (1 - \xi_j) d_j^{0-} \quad j = 2, 3, \dots, m - 1 \tag{23}$$

dargestellt. Die Idee ist hierbei, dass es für jeden Stützpunkt d_j^0 eine obere und untere Grenze $d_j^{0+} \in \mathbb{R}$ bzw. $d_j^{0-} \in \mathbb{R}$ gibt. Innerhalb dieser Grenzen wird der Stützpunkt d_j^0 mit Hilfe des Parameters ξ_j definiert. Da $\xi_j \in [0, 1]$ ist, ist sichergestellt, dass jedes d_j^0 auch nur innerhalb der Grenzen der Hilfsgrößen d_j^{0+} und d_j^{0-} liegen kann. Abbildung 7 verdeutlicht den Zusammenhang (23). Damit die resultierenden Trajektorien die Bedingungen (1)–(5) einhalten, müssen Nebenbedingungen an die Hilfsgrößen d_j^{0+} und d_j^{0-} aufgestellt werden.

Die dann noch verbleibenden Freiheitsgrade werden dazu genutzt, das Verhalten der Trajektorie möglichst gut an ihren Einsatzzweck anzupassen. Insgesamt wird ein

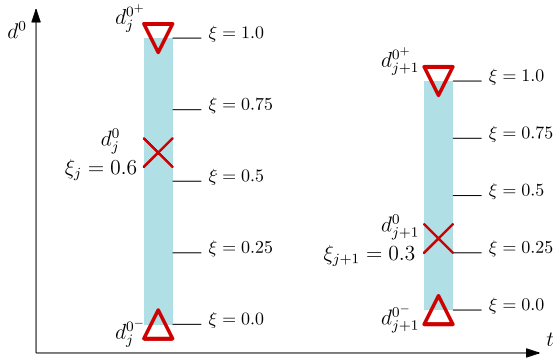


Abb. 7: Wahl der d_j^0 anhand der Hilfsgrößen d_j^{0-} und d_j^{0+} sowie der Parameter ξ_j .

quadratisches OP mit folgender Form aufgestellt:

$$\min_{\mathbf{x}} \|\mathbf{Ax} - \mathbf{b}\|_2^2 \quad (24)$$

so dass

$$\mathbf{Gx} \leq \mathbf{h}. \quad (25)$$

Der Lösungsvektor

$$\mathbf{x} = \begin{bmatrix} \mathbf{d}^{0+} \\ \mathbf{d}^{0-} \end{bmatrix}, \quad \mathbf{x} \in \mathbb{R}^{2p} \quad (26)$$

beinhaltet dabei die Hilfsgrößen \mathbf{d}^{0+} , \mathbf{d}^{0-} .

Zusammenfassend werden also die Stützpunkte \mathbf{d}^0 folgendermaßen berechnet:

$$d_j^0 = \begin{cases} u_0, & j = 0 \\ \frac{\dot{u}_0}{k-1} \Theta_1^k + u_0, & j = 1 \\ \xi_j d_j^{0+} + (1 - \xi_j) d_j^{0-}, & j = 2, 3, \dots, m-1 \end{cases} \quad (27)$$

$$\quad (28)$$

$$\quad (29)$$

Die Gütefunktion, die durch \mathbf{A} und \mathbf{b} definiert wird sowie die Nebenbedingungen, die mit \mathbf{G} und \mathbf{h} berücksichtigt werden, setzen sich wie folgt zusammen.

3.1 Gütekriterium

Die Kostenfunktion wird derart gewählt, dass die Vektordifferenzen $\mathbf{d}^{0+} - \mathbf{d}^{0-}$ möglichst groß sind. Dadurch wird der Wertebereich größer, innerhalb dessen ξ schlussendlich die Stützpunkte \mathbf{d}^0 definiert. Eine Möglichkeit stellt

¹ Zur besseren Lesbarkeit verschiebt sich der Index der Hilfsgrößen von 0 bis $p-1$ auf 2 bis $m-1$.

die zu minimierende Gütefunktion

$$J = \sum_{j=2}^{m-1} \gamma_j \|d_j^{0+} - u_{\max}\| + \gamma_j \|d_j^{0-} - u_{\min}\| \quad (30)$$

dar. Mit dem Gewichtungsvektor $\boldsymbol{\gamma} \in \mathbb{R}_+^p$ kann zusätzlich Einfluss auf das Ergebnis genommen werden. Eine Erhöhung eines γ_j führt dazu, dass die Differenz zwischen d_j^{0-} und d_j^{0+} größer wird. Besonders interessant ist das im Allgemeinen für den ersten Wert, denn mit γ_2 bzw. d_2^{0-} und d_2^{0+} wird der Stützpunkt d_2^0 definiert, der die Beschleunigung d_2^2 , also die Startbeschleunigung $\ddot{u}(t=0)$, festlegt. Je größer also die Differenz zwischen d_2^{0-} und d_2^{0+} ist, desto größer ist der Wertebereich für die Startbeschleunigung, die mit ξ_2 festgelegt wird.

Für die Darstellung in Form von (24) kann (30) mit

$$\mathbf{A} = \text{diag}(\gamma_0, \dots, \gamma_{p-1}, \gamma_0, \dots, \gamma_{p-1}), \quad (31)$$

$$\mathbf{b} = [u_{\max}, \dots, u_{\max}, u_{\min}, \dots, u_{\min}]^T \quad (32)$$

geschrieben werden.

3.2 Nebenbedingungen

Zur Erfüllung der Eigenschaften (3) bis (5) wird das konvexe Verhalten nach (19) ausgenutzt. Unter der Voraussetzung, dass

$$d_j^{0+} \leq u_{\max} \quad j = 2, 3, \dots, m-1 \quad (33)$$

$$d_j^{0-} \geq u_{\min} \quad j = 2, 3, \dots, m-1 \quad (34)$$

$$d_j^{0+} - d_j^{0-} \geq \epsilon, \quad \epsilon > 0 \quad (35)$$

und $\xi \in [0, 1]^p$ sowie den Randbedingungen (27) und (28) bleiben die Stützpunkte d_j^0 innerhalb der Grenzen

$$u_{\min} \leq d_j^0 \leq u_{\max} \quad j = 0, 1, \dots, m-1 \quad (36)$$

und damit auch der Positionsverlauf $u(t)$.

Für die weiteren Einschränkungen (4) und (5) werden die Nebenbedingungen

$$\dot{u}_{\min} \leq d_j^1 \leq \dot{u}_{\max} \quad j = 0, 1, \dots, m-2 \quad (37)$$

$$\ddot{u}_{\min} \leq d_j^2 \leq \ddot{u}_{\max} \quad j = 0, 1, \dots, m-3 \quad (38)$$

mit d_j^1 bzw. d_j^2 nach (17)

$$d_j^1 = (k-1) \frac{d_{j+1}^0 - d_j^0}{j \Theta_1^k} \quad (39)$$

$$d_j^2 = (k-2)(k-1) \frac{\frac{d_{j+2}^0 - d_{j+1}^0}{j+1 \Theta_1^k} - \frac{d_{j+1}^0 - d_j^0}{j \Theta_1^k}}{j \Theta_2^k} \quad (40)$$

definiert. Mit diesen Ungleichungsnebenbedingungen und (19) sind die Verläufe $\dot{u}(t)$ und $\ddot{u}(t)$ ebenfalls beschränkt und insgesamt die Eigenschaften (1) bis (5) erfüllt. Die Umrechnungen der Bedingungen (37) und (38) auf die Darstellung mit der Matrix \mathbf{G} und dem Vektor \mathbf{h} sind aufgrund der Größe im Anhang A aufgeführt.

3.3 Knotenvektor

Die Wahl der Knotenpunkte τ steht frei, sie werden nicht durch das OP definiert. Für die Anwendung sollten folgende Randbedingungen beachtet werden:

$$\tau_0 = \tau_1 = \dots = \tau_{k-1} = t_0 \quad (41)$$

$$\tau_m = \tau_{m+1} = \dots = \tau_{m+k-1} = t_e \quad (42)$$

$$\tau_{k-1} < \tau_k < \dots < \tau_m \quad (43)$$

$$\tau_m - \tau_0 = t_e - t_0 \geq T \quad (44)$$

Hierbei ist T die Mindestlänge der Trajektorie. Um zu gewährleisten, dass die Trajektorie die Grenzen nicht verletzen kann, muss

$$T \geq \max \left(\left| \frac{\dot{u}_{\max}}{\dot{u}_{\min}} \right|, \left| \frac{\dot{u}_{\min}}{\dot{u}_{\max}} \right| \right) \quad (45)$$

sein. T stellt die Zeit dar, die das System bei maximaler Geschwindigkeit benötigt, um mit maximaler Verzögerung zum Stillstand zu kommen. Die Grenzen des Systems werden eingehalten, wenn die Dauer der Trajektorie $t_e - t_0$ mindestens auf T gesetzt wird.

3.4 Existenz einer Lösung

Die Randbedingungen u_0 und \dot{u}_0 können nicht frei in $[u_{\min}, u_{\max}]$ bzw. $[\dot{u}_{\min}, \dot{u}_{\max}]$ gewählt werden. Ein offensichtliches, nicht lösbares Beispiel wäre $u_0 = u_{\max}, \dot{u}_0 > 0$. Existiert allerdings ein gültiges Wertepaar u_0, \dot{u}_0 , so existiert eine Trajektorie für jedes ξ :

Für u_0 und \dot{u}_0 muss gelten, dass (20), (22) und (40) gelöst werden können, sodass d_0^0, d_1^0 und d_2^0 innerhalb der Grenzen $[u_{\min}, u_{\max}]$ liegen. Die folgenden Ungleichungen entsprechen (39) und (40) für $j = 0$ und den Startwerten bzw. Grenzwerten. Sind

$$u_{\min} \leq u_0 \leq u_{\max} \quad (46)$$

$$(u_{\min} - u_0) \frac{k-1}{{}_0\Theta_1^k} \leq \dot{u}_0 \leq (u_{\max} - u_0) \frac{k-1}{{}_0\Theta_1^k} \quad (47)$$

$$\left(\frac{\dot{u}_{\min}}{k-2} {}_0\Theta_2^k + \dot{u}_0 \right) \frac{{}_1\Theta_1^k}{k-1} + \frac{\dot{u}_0}{k-1} {}_0\Theta_1^k + u_0 \leq d_2^0 \quad (48)$$

$$d_2^0 \leq \left(\frac{\dot{u}_{\max}}{k-2} {}_0\Theta_2^k + \dot{u}_0 \right) \frac{{}_1\Theta_1^k}{k-1} + \frac{\dot{u}_0}{k-1} {}_0\Theta_1^k + u_0 \quad (49)$$

$$u_{\min} \leq d_2^0 \leq u_{\max} \quad (50)$$

mit u_0 und \dot{u}_0 gültig, lässt sich für die nachfolgenden $d_j^0, j = 2, \dots, m-1$ mit einer einfachen Überlegung zeigen, dass immer eine Lösung existiert:

Es existiert für jedes d_j^0 ein nachfolgendes d_{j+1}^{0+} und d_{j+1}^{0-} , das die obere bzw. untere Schranke darstellt, innerhalb dessen die Nebenbedingungen (33), (34), (37) und (38) eingehalten werden. Mit $\xi_j \in [0, 1]$ wird ein d_{j+1}^0 definiert, das innerhalb dieses Bandes liegt. Diese Überlegung lässt sich für alle $j = 2, \dots, m-1$ fortführen. Abbildung 8 stellt diesen Gedankengang grafisch dar. Die d_j^{0+} und d_j^{0-} werden durch die Dreiecke dargestellt. Die Stützpunkte d_j^0 errechnen sich aus (27), (28) und (29).

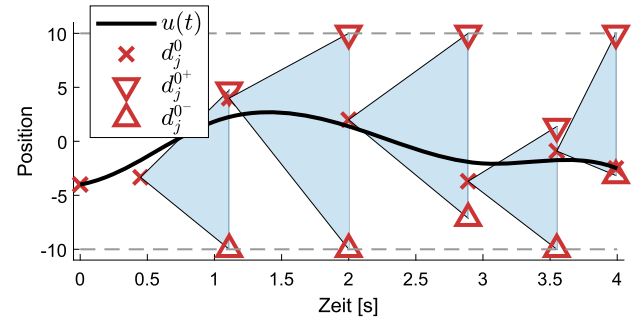


Abb. 8: Jeder Stützpunkt hat für den nachfolgenden Stützpunkt eine obere und untere Grenze, die durch die Hilfsgrößen d^{0+} und d^{0-} definiert sind. Da der nachfolgende Stützpunkt nur innerhalb dieser Grenzen gewählt werden kann, existiert auch immer eine Lösung des OP. Hier ist $\xi = [1.0, 0.6, 0.2, 0.8, 0.1]^T$.

3.5 Beispiel

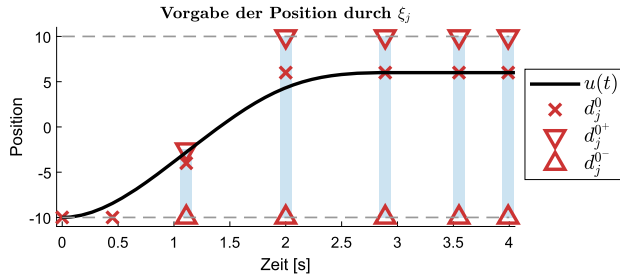
Zur Veranschaulichung wird ein Beispiel präsentiert, s. Abb. 9. Für die B-Spline-Kurve gelten die Nebenbedingungen aus Tabelle 1. Der Knotenvektor lautet

$$\tau = [0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4]^T, \quad (51)$$

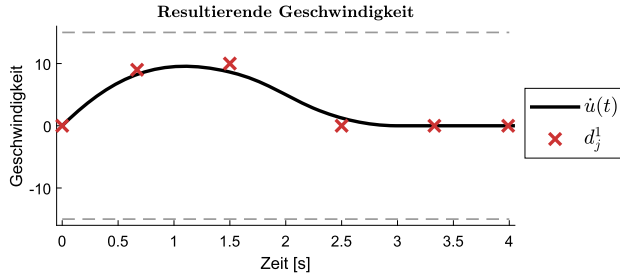
die Gewichtung

$$\mathbf{y} = \mathbf{1} = [1, 1, 1, 1, 1]^T. \quad (52)$$

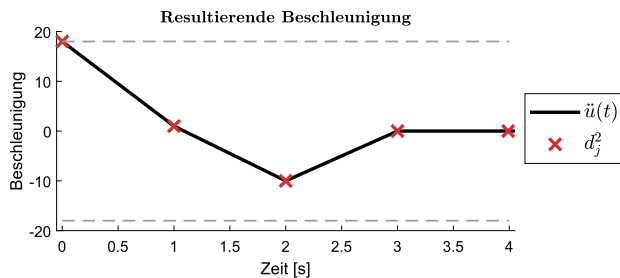
Der Parametervektor wird willkürlich zu $\xi = [0.8, \dots, 0.8]^T$ gewählt. Der Verlauf $u(t)$, der aus diesen Vorgaben und der Optimierung resultiert, ist in Abb. 9a zu sehen. d_2^{0+} wird aufgrund $d_0^0 = \dot{u}_{\max}$ begrenzt, s. Abb. 9c.



(a) Positionsverlauf $u(t)$. Die blauen Balken symbolisieren die Bereiche innerhalb derer ξ von d_j^{0-} bis d_j^{0+} die Stützpunkte d_j^0 definieren, hier ist jedes $\xi_j = 0.8$.



(b) Geschwindigkeitsverlauf $\dot{u}(t)$



(c) Beschleunigungsverlauf $\ddot{u}(t)$

Abb. 9: Verlauf der Trajektorie $u(t)$, $\dot{u}(t)$ und $\ddot{u}(t)$. ξ parametrisiert die Stützpunkte d_j^0 innerhalb des Bands von d_j^{0-} bis d_j^{0+} .

Tab. 1: Parametrierbare Werte der B-Spline-Kurve.

n	k	u_{\min}	u_{\max}	$\dot{u}_{\max} - \dot{u}_{\min}$	$\ddot{u}_{\max} - \ddot{u}_{\min}$	u_0	\dot{u}_0
7	4	-10.0	10.0	15.0	18.0	-10.0	0.0

Die weiteren oberen Hilfsgrößen d_3^{0+} bis d_6^{0+} sind maximal. Die unteren Hilfsgrößen d^0 sind minimal. Mit dem gewählten ξ strebt der Verlauf von $u(t)$ den konstanten Wert $0.8u_{\max} + 0.2u_{\min} = 6$ an. Das ist in den letzten vier d_j^0 gut erkennbar. Die Verläufe der Geschwindigkeit und Beschleunigung ergeben sich aus den optimierten Stützpunkten und den Gleichungen (37) und (38). Auch hier bleiben alle Stützpunkte und damit auch der Verlauf der Ableitungen innerhalb der definierten Grenzen.

4 Varianten der Trajektoriengenerierung

In (29) wurde definiert, dass die Stützpunkte d^0 über die Parameter ξ parametrisiert werden. Es werden nun zwei Varianten vorgestellt, bei denen ξ direkten Einfluss auf d^1 bzw. d^2 anstelle von d^0 nimmt. Es wird gezeigt, dass diese Einflussnahme durch geringe Veränderungen der Nebenbedingungen erzielt wird. Zusätzlich werden für beide Varianten jeweils eine B-Spline-Kurve mit denselben Parametern wie im ersten Beispiel erzeugt. Der Verlauf dieser Kurven unterscheidet sich stark.

4.1 Parametrierung der Geschwindigkeit

Eine Variante von (29) wird erzeugt, indem nicht direkt die Stützpunkte der Position d^0 , sondern analog zu (39) die Stützpunkte der ersten Ableitung

$$d_j^1 = \xi_{j+1} d_j^{1+} + (1 - \xi_{j+1}) d_j^{1-} \quad j = 1, \dots, m-2 \quad (53)$$

mit

$$d_j^{1+} := (k-1) \frac{d_{j+1}^{0+} - d_j^0}{j+1 \Theta_1^k} \quad j = 1, \dots, m-2 \quad (54)$$

bzw.

$$d_j^{1-} := (k-1) \frac{d_{j+1}^{0-} - d_j^0}{j+1 \Theta_1^k} \quad j = 1, \dots, m-2 \quad (55)$$

den Verlauf $\dot{u}(t)$ parametrieren. Da die d^1 die B-Spline-Kurve der Geschwindigkeit parametrieren, entsteht eine Art Durchgriff von ξ nach $\dot{u}(t)$.

Die Gütefunktion (30) wird leicht abgeändert zu

$$J = \sum_{j=1}^{m-2} \gamma_j \|d_j^{1+} - \dot{u}_{\max}\| + \gamma_j \|d_j^{1-} - \dot{u}_{\min}\|. \quad (56)$$

Es wird also dahingehend optimiert, dass die Hilfsgrößen d^{1+} jeweils das Maximum \dot{u}_{\max} und d^{1-} jeweils das Minimum \dot{u}_{\min} erreichen.

4.1.1 Nebenbedingungen

Damit die d^1 nach (53) innerhalb der Grenzen liegen, verändern sich die Nebenbedingungen (33) bis (38) leicht:

$$d_j^{0+} \leq u_{\max}, \quad d_j^{0-} \geq u_{\min} \quad j = 2, 3, \dots, m-1 \quad (57)$$

$$d_j^{1+} \leq \dot{u}_{\max}, \quad d_j^{1-} \geq \dot{u}_{\min} \quad j = 1, 2, \dots, m-2 \quad (58)$$

$$\dot{u}_{\min} \leq d_j^2 \leq \dot{u}_{\max} \quad j = 0, 1, \dots, p-1 \quad (59)$$

Wegen (53) müssen d^{1+} und d^{1-} limitiert werden, damit d^1 innerhalb der Grenzen bleibt. Die Ungleichungen (37) fallen im Gegenzug weg. Durch diese Nebenbedingungen wird der Lösungsraum für x weiter eingeschränkt.

Gleichung (53) kann mit (54), (55) und (39) umgestellt werden zu

$$d_j^0 = \xi_j d_j^{0+} + (1 - \xi_j) d_j^{0-} \quad j = 2, 3, \dots, m-1 \quad (60)$$

was Gleichung (29) entspricht. Der Zusammenhang zwischen ξ und d^0 besteht also weiterhin.

4.1.2 Beispiel

Für die folgende B-Spline-Kurve gelten dieselben Randbedingungen wie zuvor, s. Tabelle 1. Dasselbe gilt für den Knotenvektor (51) und die Gewichtung (52). Der Parametervektor beeinflusst durch die neue Gleichung (53) nun direkt die Geschwindigkeit, s. Abb. 10b. In Abb. 10a ist der Zusammenhang (60) gut erkennbar.

Eine Interpretation fällt hier schwerer als im ersten Beispiel. Das Ziel der Optimierung ist, die Abstände der Hilfsvariablen zu den Maxima zu minimieren unter Berücksichtigung der Nebenbedingungen. So sind die Hilfsgrößen d_0^{0-} , d_1^{0-} und d_2^{0-} minimal und d_2^{0+} , d_3^{0+} und d_4^{0+} maximal. d_0^{0+} wird zudem noch durch die maximale Beschleunigung d_0^2 beschränkt. Aus den verbleibenden Freiheitsgraden stellt die gezeigte Lösung das Minimum der Gütefunktion (56) dar. Es muss sich vor Augen gehalten werden, dass der eigentliche Verlauf keine besondere Rolle spielt – sondern die Frage ist, welches Gütemaß diese Trajektorie im übergeordneten OP erzeugt. Relevant ist also die Information, dass die OV, die den Parametervektor ξ festlegen, einen direkten Einfluss auf den Geschwindigkeitsverlauf besitzen.

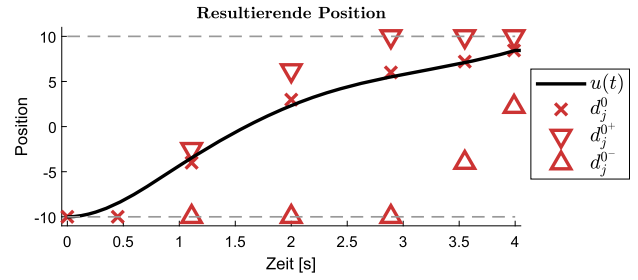
4.2 Parametrierung der Beschleunigung

Analog zu Abschnitt 4.1 wird nun ξ genutzt, um die Stützpunkte der Beschleunigung festzulegen:

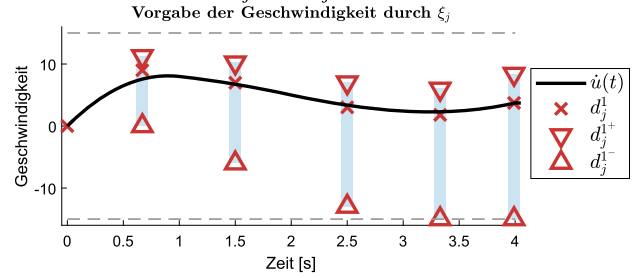
$$d_j^2 = \xi_j d_j^{2+} + (1 - \xi_j) d_j^{2-} \quad j = 0, \dots, m-3 \quad (61)$$

mit

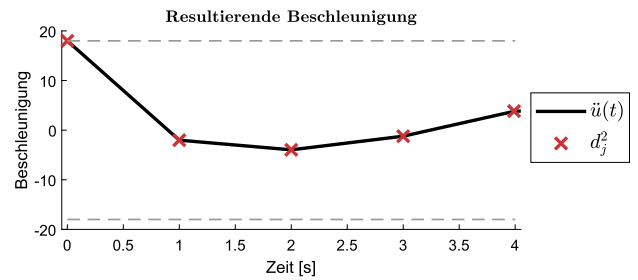
$$d_j^{2+} := (k-2)(k-1) \frac{\frac{d_{j+2}^{0+} - d_{j+1}^{0+}}{j+1 \Theta_1^k} - \frac{d_{j+1}^{0+} - d_j^{0+}}{j \Theta_1^k}}{j \Theta_2^k} \quad (62)$$



(a) Positionsverlauf $u(t)$. Die Hilfsgrößen d_j^{0-} und d_j^{0+} erreichen teilweise die Grenzen u_{\min} bzw. u_{\max} . Dies beeinflusst den wählbaren Bereich der d_j^{1-} bis d_j^{1+} .



(b) Geschwindigkeitsverlauf $\dot{u}(t)$. Die blauen Balken symbolisieren die Bereiche innerhalb derer ξ von d_j^{1-} bis d_j^{1+} die Stützstellen d_j^1 definieren, hier ist jedes $\xi_j = 0.8$.



(c) Beschleunigungsverlauf $\ddot{u}(t)$

Abb. 10: Verlauf der Trajektorie $u(t)$, $\dot{u}(t)$ und $\ddot{u}(t)$, wobei nun ξ die Stützpunkte der ersten Ableitung d^1 parametrisiert.

und

$$d_j^{2-} := (k-2)(k-1) \frac{\frac{d_{j+2}^{0-} - d_{j+1}^{0-}}{j+1 \Theta_1^k} - \frac{d_{j+1}^{0-} - d_j^{0-}}{j \Theta_1^k}}{j \Theta_2^k}. \quad (63)$$

Die Gütefunktion wird wieder leicht abgeändert:

$$J = \sum_{j=0}^{m-3} \gamma_j \|d_j^{2+} - \ddot{u}_{\max}\| + \gamma_j \|d_j^{2-} - \ddot{u}_{\min}\|. \quad (64)$$

Es wird also dahingehend optimiert, dass die Hilfsgrößen d^{2+} jeweils das Maximum \ddot{u}_{\max} und d^{2-} jeweils das Minimum \ddot{u}_{\min} erreichen.

4.2.1 Nebenbedingungen

Analog zu Abschnitt 4.1 verändern sich die Nebenbedingungen für die Stützpunkte der zweiten Ableitung. Anstelle einer oberen und unteren Beschränkung der Stützpunkte selbst müssen nun die Hilfsgrößen d^{2+} und d^{2-} beschränkt werden:

$$d_j^{0+} \leq u_{\max}, \quad d_j^{0-} \geq u_{\min} \quad j = 2, 3, \dots, m - 1 \quad (65)$$

$$d_j^{1+} \leq \dot{u}_{\max}, \quad d_j^{1-} \geq \dot{u}_{\min} \quad j = 1, 2, \dots, m - 2 \quad (66)$$

$$d_j^{2+} \leq \ddot{u}_{\max}, \quad d_j^{2-} \geq \ddot{u}_{\min} \quad j = 0, 1, \dots, m - 3 \quad (67)$$

Auch hier kann (61) mit (62), (63) und (40) umgestellt werden zu

$$d_j^0 = \xi_j d_j^{0+} + (1 - \xi_j) d_j^{0-} \quad j = 2, 3, \dots, p - 1 \quad (68)$$

was wieder (29) entspricht.

4.2.2 Beispiel

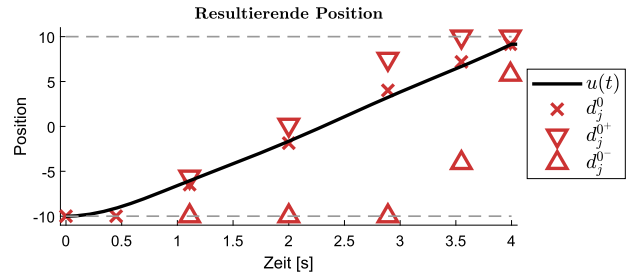
Wie zuvor bleiben alle Werte unverändert, lediglich die Nebenbedingungen schränken das System weiter ein. Der neue Verlauf der Trajektorie wird durch die Stützpunkte der Beschleunigung d^2 parametrisiert, s. Abb. 11c.

Auffallend ist, dass hier nun die verfügbaren Bereiche zwischen den Minima und Maxima deutlich kleiner sind, beispielsweise erreicht d_0^{2+} nicht mehr den Maximalwert \ddot{u}_{\max} . Das liegt am Gütekriterium (64), das die Entfernung aller Hilfsgrößen d_j^{2+} und d_j^{2-} zum jeweiligen Maximum minimiert. Die Hilfsgrößen $d_0^{0-}, d_1^{0-}, d_2^{0-}, d_3^{0+}$ und d_4^{0+} erreichen die oberen bzw. unteren Grenzen, s. 11a. Aufgrund des Zusammenhangs (68) beschränken diese im Umkehrschluss auch die Hilfsgrößen der ersten und zweiten Ableitung.

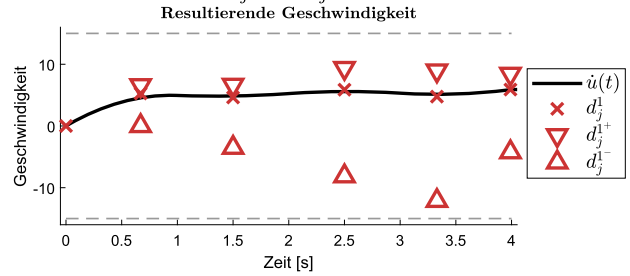
Auch hier liegt die relevante Information nicht im eigentlichen Verlauf der Trajektorie sondern in der Tatsache, dass ξ direkten Einfluss auf den Verlauf der Beschleunigung besitzt, was ggf. für das übergeordnete OP von Vorteil ist.

5 Ergebnisse am Beispiel MPC

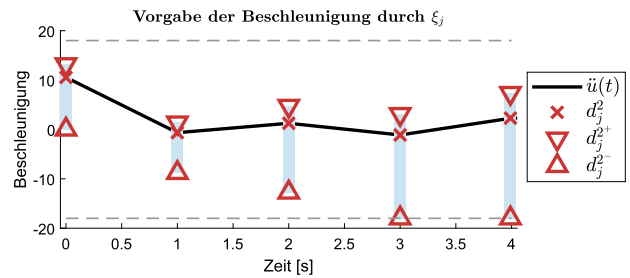
Die Methode ist das Resultat einer konkreten Problemstellung, die in diesem Beitrag als Beispiel dienen soll. Am Institut für Systemdynamik und Regelungstechnik des Deutschen Zentrums für Luft- und Raumfahrt (DLR) wird ein Bewegungssimulator auf Basis eines Industrieroboters



(a) Positionsverlauf $u(t)$. Die Hilfsgrößen d_j^{0-} und d_j^{0+} erreichen teilweise die Grenzen u_{\min} bzw. u_{\max} . Dies beeinflusst den wählbaren Bereich der d_j^{2-} bis d_j^{2+} .



(b) Geschwindigkeitsverlauf $\dot{u}(t)$



(c) Beschleunigungsverlauf $\ddot{u}(t)$. Die blauen Balken symbolisieren die Bereiche innerhalb derer ξ von d_j^{2-} bis d_j^{2+} die Stützstellen d_j^2 definieren, hier ist jedes $\xi_j = 0.8$.

Abb. 11: Verlauf der Trajektorie $u(t)$, $\dot{u}(t)$ und $\ddot{u}(t)$, wobei nun ξ die Stützpunkte der zweiten Ableitung d^2 parametrisiert.

betrieben, der *DLR Robotic Motion Simulator* (RMS) [2], s. Abb. 2. Der Roboter besitzt sechs rotatorische Achsen und eine siebte Linearachse, auf der der Roboter verfahren kann. Das System ist somit überbestimmt und kann verschiedene Gelenkwinkelkonfigurationen für eine gewünschte Position und Orientierung des Endeffektors einnehmen.

Eine Person sitzt in der Gondel am Endeffektor und kann durch Eingabegeräte bspw. eine Fahrsimulation durchführen. Die Problemstellung ist in diesem Fall, dass die auf den Fahrer wirkenden Beschleunigungen aus der Fahrsimulation möglichst exakt durch den Simulator wiedergegeben werden sollen. Vereinfacht kann eine nichtlineare Funktion angegeben werden, die das Übertragungsverhalten von den Robotergelenken q und den zeitlichen

Ableitungen $\dot{\mathbf{q}}$ und $\ddot{\mathbf{q}}$ zu den *wahrgenommenen Kräften* (englisch: perceived forces) \mathbf{f}_p beschreibt:

$$\mathbf{f}_p = f(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}). \quad (69)$$

Diese Übertragungsfunktion dient als System für einen MPC-Ansatz, mit dem Ziel, dass \mathbf{f}_p exakt einer Referenz \mathbf{f}_r folgt. Im Allgemeinen ist diese Referenz ein aktueller Wert aus dem Fahrdynamikmodell der Fahrsimulation, gekoppelt mit einem sogenannten *Wahrnehmungsmodell des Menschen* [14]. Es kann dabei vorkommen, dass eine gewünschte Bewegung nicht vom Simulator abgebildet werden kann, weil er sich in einer ungeeigneten Konfiguration befindet, z. B. beim Durchfahren kinematischer Singularitäten.

Es soll deshalb auch möglich sein, der Referenz zukünftige, wahrscheinliche Werte aufzuprägen. Solche Informationen können aus vollständig bekannten Szenarien resultieren, also Simulationsfahrten, in denen der Fahrer keinen Einfluss nehmen kann. Dies sind beispielsweise Tests von Autopiloten und können als sogenannte Offline-Optimierungen durchgeführt werden. Ebenso können Szenarien, bei denen ein bestimmtes Handeln vorausschaubar ist, mit in die Referenz einfließen. Beispiele dafür sind z. B. „Beschleunigen auf dem Beschleunigungsstreifen einer Autobahn“ oder „nahendes Stoppschild“. Mit solch prädiktiven Werten lässt sich der Simulator vorausschauend in eine Position bewegen, die eine Wiedergabe der Referenz ermöglicht.

Im Beispiel sind die Gelenkwinkel \mathbf{q} sowie deren ersten und zweiten Ableitungen begrenzt. Wie beim MPC-Verfahren üblich, werden zu jedem Zeitschritt t_s der Berechnung die aktuellen Zustände übernommen und eine neue Optimierung gestartet. Vom Ergebnis der optimalen Trajektorie wird nur der erste Wert übernommen, auf das reale System gegeben und die Optimierung für den nächsten Zeitschritt gestartet. Das Gütemaß der MPC-Anwendung ist die Minimierung des Fehlers zwischen der Referenz der wahrgenommenen Kräfte und den im Simulator wahrgenommenen Kräften.² Im Falle einer MPC wird in jedem Zeitschritt der aktuelle Startzeitpunkt t_s als Beginn der Berechnung des Gütemaßes verwendet und über einen zeitlichen Horizont, dem *Prädiktionshorizont* t_p , integriert,

$$J = \int_{t_s}^{t_s+t_p} \|\mathbf{f}_r(t) - \mathbf{f}_p(t)\|_2^2 dt. \quad (70)$$

² Am RMS werden neben den *wahrgenommenen Kräften* auch *wahrgenommene Drehraten* berücksichtigt. Zudem fließen Elemente mit in die Optimierung ein, um das System in eine Startregion zurückzubringen.

Normalerweise würde nun ein Optimierungsalgorithmus über OV einen Gelenkwinkelverlauf $\mathbf{q}(t)$ vorgeben und mit diesem das System simulieren bzw. auswerten. Durch Variation der OV verändert sich der Gelenkwinkelverlauf und entsprechend das Gütemaß. Im klassischen Ansatz würden die Ableitungen ausgewertet und als Nebenbedingungen hinzugefügt werden, die dann vom Optimierungsalgorithmus berücksichtigt werden. Mit dem hier vorgestellten Verfahren fallen diese Nebenbedingungen aus der Optimierung heraus, da sie direkt in den implizit beschränkten B-Splines berücksichtigt werden. Iterationen sind also nur noch für die Minimierung des Gütemaßes notwendig. Für Echtzeitanwendungen ist diese Verlagerung von großem Vorteil. Zur Sicherstellung der Echtzeit wird häufig eine vorgegebene Anzahl an Iterationen festgelegt. Mit den implizit beschränkten B-Splines kann eine Optimierung jederzeit abgebrochen werden ohne Gefahr zu laufen, dass Nebenbedingungen verletzt werden.

Die Implementierung der Trajektoriengenerierung erfolgte in C++. Die Lösung des OP wurde mit `dlsei` [7] durchgeführt. Für den RMS wurde ein standardisiertes Überholmanöver aus der Fahrzeugentwicklung gewählt (ISO 3888-1) und offline optimiert, das heißt, der Verlauf der Bewegung war bekannt. Als Referenz wurde ein MPC-Modell mit parametrierbaren Splines und Nebenbedingungen erstellt. Die weiteren Parameter wie Taktrate (12 ms), Prädiktionshorizont (240 ms), Lösungsalgorithmus etc. waren identisch. Durch Nutzung der vorgestellten implizit beschränkten B-Splines konnte die Berechnungszeit bei vergleichbarem Ergebnis um einen Faktor von knapp 9 reduziert werden.

Der Soll- und Ist-Verlauf der *wahrgenommenen Kraft* in laterale Richtung ist in Abb. 12 abgebildet. Sie zeigt das Aus- und Einscheren beim Überholmanöver. Die Verläufe stimmen gut überein und zeigen, dass die modellprädiktive Regelung am RMS gut funktioniert. In Abb. 13 ist der Beschleunigungsverlauf der Linearachse zu sehen, der auch

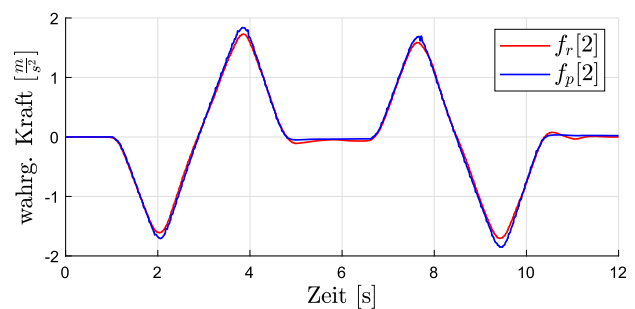


Abb. 12: Vergleich des Soll- und Ist-Verlaufs der wahrgenommenen Kraft in laterale Richtung am Beispiel des RMS.

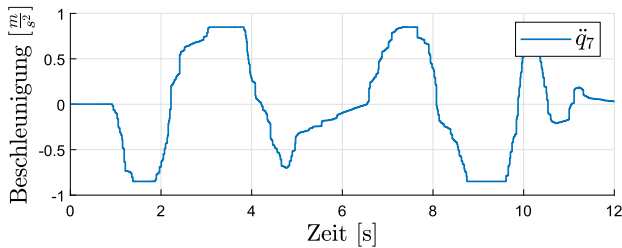


Abb. 13: Beschleunigungsverlauf der Linearachse der RMS. Die Grenzen werden häufig erreicht aber nicht überschritten.

an einigen Stellen den Grenzwert erreicht und hält. Das Ergebnis zeigt, dass die implizit beschränkten B-Splines dafür geeignet sind, in einem OP angewandt zu werden. Sie führen zudem zu einer Verbesserung der Rechenzeit, sofern das zu optimierende System derart komplex ist, dass die zusätzliche Berechnungszeit für die implizit beschränkten B-Splines deutlich kürzer ist als die Zeit für die Berechnung bzw. Simulation des Systems.

In dieser Anwendung, bei der das Gütekriterium direkt von der Beschleunigung abhängt, lieferte die Variante, bei der die Stützpunkte der zweiten Ableitung \mathbf{d}^2 durch ξ definiert werden, die mit Abstand besten Ergebnisse. Eine pauschale Aussage, in welcher Anwendung welche der drei Varianten zu bevorzugen ist, ist momentan allerdings nicht möglich.

6 Diskussion der Methode

Die Vor- und Nachteile, die sich aus der Nutzung der implizit beschränkten B-Splines ergeben, hängen maßgeblich vom Einsatzzweck ab. Für die Beurteilung wird ein Szenario wie das des Bewegungssimulators zugrunde gelegt:

Eine übergeordnete MPC wertet in jedem Zeitschritt ein Prädiktionsmodell über einen Prädiktionshorizont aus und berücksichtigt dabei eine Kostenfunktion sowie Nebenbedingungen. Dadurch, dass alle Nebenbedingungen durch die Verwendung des vorgestellten Trajektoriengenerators aus dem MPC-Problem herausgenommen werden können, ergeben sich folgende Vorteile:

- Alle Nebenbedingungen werden durch die neuen B-Spline-Kurven implizit berücksichtigt. Dies reduziert die Anzahl der Iterationen der MPC, da lediglich die übergeordnete Gütefunktion minimiert werden muss.
- Der Prädiktionshorizont der MPC ist unabhängig von der Mindestlänge der B-Spline-Kurve. Zur Absicherung, dass die Systemgrenzen eingehalten werden können, muss nun nicht mehr der Prädiktionshori-

zont lang genug gewählt werden, vgl. (45). Je kürzer der Prädiktionshorizont ist, desto schneller ist die Berechnung bzw. Simulation des Prädiktionsmodells ausgewertet.

- Zum Zeitpunkt t_0 werden u_0 und \dot{u}_0 vorgegeben, die Beschleunigung jedoch nicht. Sie kann, abhängig von u_0, \dot{u}_0 sowie ${}_0\Theta_2^k$, innerhalb von \ddot{u}_{\min} und \ddot{u}_{\max} springen wodurch die maximale Systemdynamik ausgenutzt wird.
- Die Verläufe für die Position, Geschwindigkeit und Beschleunigung liegen in analytischer Form vor. Sie können zu jedem Zeitpunkt ohne Fehler durch numerische Verfahren ausgewertet werden.
- Die Iterationen der MPC-Anwendung können begrenzt werden, ohne Gefahr zu laufen, Nebenbedingungen zu verletzen, da jede Lösung die Nebenbedingungen einhält. Dies ist eine Grundvoraussetzung für Echtzeitfähigkeit.

Nachteilig können folgende Punkte aufgeführt werden:

- Die Erzeugung der B-Spline-Kurve benötigt Rechenzeit. Auf einem Kern eines Xeon E5-1630 dauert es durchschnittlich $50.73\mu\text{s}$, um eine implizit beschränkte B-Spline-Kurve mit $m = 7$ Stützpunkten zu berechnen.
- Die B-Spline-Kurven stellen eine Nichtlinearität dar und erhöhen damit die Komplexität des Gesamtsystems.

7 Zusammenfassung und Ausblick

In dieser Arbeit wurde eine Methode vorgestellt, die eine Trajektorie auf Basis einer B-Spline-Kurve erzeugt. Die implizit begrenzten B-Splines halten konstante Grenzen für die Position, Geschwindigkeit und Beschleunigung über den gesamten Verlauf ein. Die Parametrierung der Trajektorie erfolgt über den freien Parametervektor ξ und davon abhängigen optimierten Hilfsgrößen. Die Trajektorien können insbesondere als Steuer- bzw. Eingangstrajektorien in Optimalsteuerungen oder modellprädiktiven Verfahren eingesetzt werden. Durch die implizite Berücksichtigung der Stellgrößenbeschränkungen verringert sich so die Komplexität des übergeordneten Optimierungsproblems. Auch die Echtzeitfähigkeit wird verbessert, denn ein Abbruch der Optimierung, beispielsweise aufgrund einer festgelegten Anzahl an Iterationen, liefert in jedem Fall eine Lösung, die die Nebenbedingungen nicht verletzt.

Es wurden drei verschiedene Ansätze vorgestellt, wie ξ auf die Generierung der Trajektorien wirken kann und

2. T. Bellmann. *Optimierungsbasierte Bahnplanung für interaktive robotische Bewegungssimulatoren*. PhD thesis, 2014.
3. C. De Boor. *A Practical Guide to Splines, volume 27*. Springer-Verlag New York, 1978.
4. H. Fujioaka and H. Kano. Control theoretic B-spline smoothing with constraints on derivatives. In *52nd IEEE Conference on Decision and Control*, pages 2115–2120. IEEE, 2013.
5. A. Gasparetto and V. Zanotto. Optimal trajectory planning for industrial robots. *Advances in Engineering Software*, 41(4):548–556, 2010.
6. L. Grüne and J. Pannek. *Nonlinear Model Predictive Control: Theory and Algorithms*. Communications and Control Engineering. Springer International Publishing, 2016.
7. R. Hanson and K. Haskell. Algorithm 587: Two algorithms for the linearly constrained least squares problem. *ACM Transactions on Mathematical Software (TOMS)*, 8(3):323–333, 1982.
8. K. Höllig and J. Hörner. *Approximation and Modeling with B-splines, volume 132*. SIAM, 2013.
9. T. J. Hughes, J. A. Cottrell and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer methods in applied mechanics and engineering*, 194(39-41):4135–4195, 2005.
10. W. V. Looock, G. Pipeleers and J. Swevers. B-spline parameterized optimal motion trajectories for robotic systems with guaranteed constraint satisfaction. *Mechanical Sciences*, 6(2):163–171, 2015.
11. A. Pfeiffer. Optimization library for interactive multi-criteria optimization tasks. In *9th International Modelica Conference*, 2012.
12. L. Piegl and W. Tiller. *The NURBS book*. Springer Science & Business Media, 2012.
13. J. B. Rawlings and D. Q. Mayne. *Model predictive control: Theory and design*, 2009.
14. K. Zaychik and F. Cardullo. Nonlinear motion cueing algorithm: Filtering at pilot station and development of the nonlinear optimal filters for pitch and roll. Technical Report NASA/CR-2012-217567, NF1676L-14637, NASA, May 2012.

Autoreninformationen

Andreas Seefried

Institut für Systemdynamik und Regelungstechnik, DLR
Oberpfaffenhofen, Wessling, Deutschland
andreas.seefried@dlr.de

Andreas Pfeiffer

Institut für Systemdynamik und Regelungstechnik, DLR
Oberpfaffenhofen, Wessling, Deutschland
andreas.pfeiffer@dlr.de