

Methoden

Zonglin Liu* und Olaf Stursberg

Optimierungsbasierte Regelung und Steuerung Hybrid-Dynamischer Systeme

Optimal control of hybrid dynamic systems

<https://doi.org/10.1515/auto-2018-0080>

Empfangen 1. Juli 2018; angenommen 19. September 2018

Zusammenfassung: Die optimale Steuerung und Regelung hybrider Systeme ist herausfordernd, da sie sprunghafte Änderungen von Zustandsgrößen und die Kombinatorik ereignisdiskreten Verhaltens in die Optimierung von kontinuierlicher Dynamik einbringt. Dieser Beitrag gibt zunächst einen Überblick über die wichtigsten Lösungsverfahren für verschiedene Problemausprägungen. Anschließend wird für den Fall zeitdiskreter hybrider Systeme mit kontinuierlichen und diskreten Stelleingriffen eine neue Lösungsmethode vorgestellt, worin die optimale Lösung durch Verwendung von approximierten Kostenstrahlen und Nachbarschaftsbeziehungen explorierter Zustände mit relativ geringem Rechenaufwand angenähert wird.

Schlagwörter: optimale Regelung, Optimalsteuerung, hybride Systeme, Graphensuche, geschaltete Dynamiken

Abstract: The optimal control of hybrid systems is challenging as it adds resets of state variables as well as the combinatorics of discrete-event behavior to the optimization of continuous dynamics. This contribution first provides an overview of the most important methods for different variants of the hybrid optimal control problem. For a problem instance of discrete-time hybrid systems with continuous and discrete controls, the paper then proposes a new technique. In there, the optimal solution is approximated with relatively low computational effort by use of approximated cost bounds and an adjacency criterion for explored states.

Keywords: optimal control, hybrid systems, graph search, switched dynamics

***Korrespondenzautor:** Zonglin Liu, Regelungs- und Systemtheorie, FB Elektrotechnik/Informatik, Universität Kassel, D-34109 Kassel, Germany, E-Mail: z.liu@uni-kassel.de

Olaf Stursberg, Regelungs- und Systemtheorie, FB Elektrotechnik/Informatik, Universität Kassel, D-34109 Kassel, Germany, E-Mail: stursberg@uni-kassel.de

1 Einleitung

Die Klasse hybrid-dynamischer Modelle ist etabliert, um das Verhalten solcher Systeme darzustellen, in denen stetige und sprunghafte Änderungen zeitveränderlicher Größen koexistieren. Dieser Sachverhalt liegt vor, wenn bestimmte Vorgänge im betrachteten System erheblich schneller ablaufen (und damit ohne Zeitverbrauch modelliert werden können) als die das kontinuierliche Verhalten dominierenden Abläufe. Hybrid-dynamische Modelle wurden erfolgreich für Beispielsysteme aus verschiedenen Anwendungsdomänen eingesetzt, wie im Automobil, in der Robotik, der Energietechnik sowie der Fertigungs- oder Prozessindustrie. Die sprunghaften (meist schaltenden) Dynamikanteile können dabei entweder der Strecke zugeordnet sein und müssen im Entwurf der Regelung bzw. Steuerung berücksichtigt werden oder sie sind selbst Freiheitsgrad innerhalb der Entscheidungseinheit.

Dieser Beitrag widmet sich der Aufgabe des Entwurfs optimaler Regelungen bzw. Steuerungen für hybrid-dynamische Systeme. Im Vergleich zur entsprechenden Aufgabe für ausschließlich kontinuierlich modellierte Systeme werden die dem Entwurf zugrundeliegenden Optimierungsprobleme typischerweise erheblich komplizierter: (1) Die sprunghaften Effekte führen zu Unstetigkeiten in den Kostenfunktionsverläufen und erfordern somit insbesondere für gradientenbasierte Verfahren Sonderbehandlungen. (2) Die aus dem ereignisdiskreten Systemanteil resultierende Kombinatorik erhöht den Rechenaufwand meist deutlich. Je nach Ausprägung des betrachteten hybriden Modells, dem gewählten Zeitmodell und der Zielsetzung des Entwurfs (Optimalsteuerung oder optimales Regelgesetz) sind spezifische Lösungsalgorithmen erforderlich. Der vorliegende Beitrag gibt, mit Bezug zu einer allgemeinen und gängigen Definition hybrid-dynamischer Modelle (Kap. 2), einen Überblick über existierende optimierungsbasierte Verfahren zur Regelung und Steuerung (Kap. 3). Anschließend wird eine spezielle Instanz von zeitdiskreten Optimalsteuerungsproblemen herausgegriffen und ein neuer Lösungsansatz präsentiert (Kap. 4)

und auf Beispiele angewendet (Kap. 5). Abschließend werden einige Schlussfolgerungen zum Stand optimierungsbasierter Methoden für hybrid-dynamische Systeme gezogen (Kap. 6).

2 Allgemeine Problemstellung

Als Grundlage des Methodenüberblicks führen wir zunächst eine allgemeine Definition eines hybrid-dynamischen Systems an, die sich an den etablierten Fassungen *hybrider Automaten* orientiert [15, 20]. Wir beziehen uns hier zuerst auf den Fall eines kontinuierlichen Zeitmodells und abstrahieren von Ausgangsvariablen und -funktionen. Hierfür besteht ein hybrider Automat HA aus den folgenden Komponenten:

- den Zeitdomänen $T = \{t \in \mathbb{R}^{\geq 0} : t_0 \leq t \leq t_f\}$ und $T_k = \{t_0, t_1, t_2, \dots, t_f\}$ mit Zeitpunkten $t_k \in T$, auf denen die Änderungen der kontinuierlichen und diskreten Zustände definiert sind;
- dem hybriden Zustandsraum $S := Z \times X$ als kartesisches Produkt der Menge Z der diskreten Zustände $z(t_k) \in Z$ und dem kontinuierlichen Zustandsraum $X \subseteq \mathbb{R}^{n_x}$, auf dem $x(t) \in X$ definiert ist; der hybride Zustand wird mit $s = (z, x)$ bezeichnet und zur Vereinfachung der Darstellung setzen wir $Z := \{1, 2, \dots, n_z\}$ mit endlichem n_z an;
- dem hybriden Eingangsraum $\mathcal{U} := V \times U$ als kartesisches Produkt der endlichen Menge V diskreter Eingänge $v(t_k) \in V = \{0, 1, \dots, n_v\}$ und dem Raum $U \subseteq \mathbb{R}^{m_c}$ kontinuierlicher Eingänge $u(t) \in U$; der hybride Eingang wird mit $w = (v, u) \in \mathcal{U}$ bezeichnet;
- einer Menge $I = \{I_1, \dots, I_{n_z}\}$ von Zustandsinvarianten I_z als kompakte Teilmenge von X für jeden diskreten Zustand $z \in Z$;
- einer diskreten Zustandsüberföhrungsfunktion $f_d : T_k \times Z \times V \times X \rightarrow Z$, die *Transitionen* $(z, \bar{z}) \in Z \times Z$ zwischen diskreten Zuständen definiert;
- einer Menge von kompakten Übergangsmengen $G_{z, \bar{z}} \subseteq X$ für jede durch f_d definierte Transition (z, \bar{z}) ;
- einer Sprungfunktion $r_c : T_k \times X \times Z \times Z \rightarrow X$, die jeder Transition eine Vorschrift zur Aktualisierung des kontinuierlichen Zustands zuweist;
- und der kontinuierlichen Zustandsüberföhrungsfunktion $f_c : T \times X \times U \times Z \rightarrow X$, die jedem diskreten Zustand $z \in Z$ eine gewöhnliche Differentialgleichung zuordnet, die den Verlauf von $x(t)$ für eine gegebene Initialisierung und ein Intervall in T eindeutig determiniert.

Wenn das Modell HA auf den Zustand $s_0 = (z_0, x(t_0))$, $z_0 \in Z$, $x(t_0) \in I_{z_0}$ initialisiert wurde, ergibt sich für eine Sequenz von diskreten Eingängen $\phi^V = \{v_0, \dots, v_k, \dots, v_f\}$, $v_k = v(t_k) \in V$, $t_k \in T_k$, $k \in \{0, 1, 2, \dots\}$ und für eine kontinuierliche Eingangstrajektorie $\phi^U = \{u(t) \in U, t \in T\}$ eine zulässige Trajektorie der hybriden Zustände $\phi^S = \{s(t) = (z(t_k), x(t)), x(t) \in X, z(t_k) \in Z\}$ für jede Phase $t \in [t_k, t_{k+1}[$ gemäß:

1. dem kontinuierlichem Ablauf ausgehend von $x(t_k)$:

$$x(t_{k+1}^-) := x(t_k) + \int_{t_k}^{t_{k+1}} f_c(\tau, x(\tau), u(\tau), z(t_k)) d\tau,$$

wobei $x(\tau) \in I_{z_k}$ für alle $\tau \in [t_k, t_{k+1}[$ gelten muss; dabei bezeichnet $x(t_{k+1}^-)$ den linksseitigen Grenzwert $\lim_{\epsilon \rightarrow 0} x(t_{k+1} - \epsilon)$ des Zustands $x(t_{k+1})$;

2. der Transition mit Sprung: wenn $x(t_{k+1}^-) \in G_{z_k, z_{k+1}}$ und $v(t_{k+1}) \neq 0$, dann folgt der hybride Zustand $s(t_{k+1}) = (z_{k+1}, x_{k+1}^T)^T$ aus:
 - $z_{k+1} := f_d(t_{k+1}, z_k, v_{k+1}, x(t_{k+1}^-))$
 - $x_{k+1} := r_c(t_{k+1}, x(t_{k+1}^-), z_k, z_{k+1}) \in I_{z_{k+1}}$,
ansonsten gilt: $z_{k+1} := z_k$, $x_{k+1} := x(t_{k+1}^-)$.

Zur Vereinfachung der weiteren Betrachtung sei einschränkend angenommen, dass sich $\phi^S(t)$ nach diesen Regeln deterministisch ergibt.

Die Festlegung der diskreten Zeitpunkte in T_k ergibt sich also aus dem Ablauf des Modells durch das Erreichen einer Übergangsmenge $G_{z_k, z_{k+1}}$ und Selektion eines neuen diskreten Zustands über v_{k+1} . Gemäß der Kategorisierung in [6] umfasst die obige Definition sowohl *autonomes Schalten* der kontinuierlichen Dynamik (wenn die Bedingung $x(t_{k+1}^-) \in G_{z_k, z_{k+1}}$ bei beliebigem $v_{k+1} \neq 0$ die Transition bewirkt) als auch *gesteuertes Schalten*, wenn die Wahl eines bestimmten $v_{k+1} \neq 0$ (z. B. bei $G_{z_k, z_{k+1}} = I_{z_k}$) die Transition auslöst. Die Selektion von $v_{k+1} = 0$ unterbindet demgegenüber die Transition. Diese beiden Ausprägungen können mit sprunghaften Veränderungen von $x(t)$ durch r_c kombiniert werden. Gemäß [10] ist es z. B. üblich, für einen aufspringenden Ball die Geschwindigkeitsänderung als Transition mit Sprung zu modellieren, wenn dieses Ereignis deutlich weniger Zeit in Anspruch nimmt als die Flugphase des Balls.

Die optimale Regelung oder Steuerung eines hybriden Automaten legt die Definition eines Güte- oder Kostenfunctionals zu Grunde. Wir benutzen hier zunächst eine allgemeine Form $J(s(t_0), \phi^U, \phi^V)$ zur Formulierung eines Minimierungsproblems:

$$\min_{\phi^U, \phi^V} J(s(t_0), \phi^U, \phi^V), \quad (1)$$

welches unter der Randbedingung zu lösen ist, dass über die zulässigen Zustandstrajektorien von HA ausgehend von $s(t_0) = s_0$ gesucht wird. Häufig kodiert J die Minimierung des Abstandes zu einem hybriden Zielzustand $s(t_f) = s_f$, oder das Erreichen eines solchen Zustands wird als Randbedingung formuliert. Diese Fassung entspricht einem Optimalsteuerproblem, also der Bestimmung der optimalen Trajektorie von einem Anfangs- in einen Zielzustand. Eine Instanz eines solchen Problems unter Verwendung diskreter Zeit wird in Kap. 4 detaillierter behandelt. Dagegen ergibt sich eine Problemformulierung der optimalen Regelung aus dem Übergang auf die Berechnung eines Reglerfunktional für jede mögliche Initialisierung $s(t_0) \in S_0$ mit $S_0 = \bigcup_{j \in Z} (\{z_j\} \times I_{z_j})$. Der folgende Abschnitt gibt einen Überblick über vorliegende Lösungsansätze beider Problemausprägungen.

3 Literaturüberblick

In den vergangenen zwei Jahrzehnten wurden für verschiedene Varianten des zuvor beschriebenen Problems Lösungsansätze vorgeschlagen, die im Wesentlichen die aus der optimalen Steuerung und Regelung kontinuierlicher Systeme bekannten Prinzipien auf hybride Probleme erweitern. Der daraus resultierende Stand der Forschung wird im Folgenden zusammengefasst.

3.1 Dynamische Programmierung

Eine wesentliche Arbeitsrichtung mit Bezug zur optimalen Regelung hybrider Systeme bestand in der Erweiterung des Prinzips der *dynamischen Programmierung* [14, 35, 18, 27]: Das Ziel besteht darin, die so genannte *Wertefunktion* über den hybriden Zustands- und Eingangsräumen zu bestimmen. Dies erfordert die Gitterung der reellwertigen Teilräume. Für die Kombination der Gitterpunkte und der diskreten Zustände wird dann über die kontinuierlichen und (falls vorhanden) diskreten Eingänge iteriert, um die Erfüllung der Hamilton-Jacobi-Bellman-Gleichung zu erhalten. Bei Wechsel des diskreten Zustands sind die Transitionen mit Übergangsbedingungen und Sprungfunktionen geeignet zu berücksichtigen. Ansätze nach diesem Schema liegen sowohl für zeitkontinuierliche als auch zeitdiskrete Modelldefinitionen vor [5]. Die erhaltene Wertfunktion konvergiert zur global optimalen Lösung, wenn sich die Gitterweiten dem Wert Null nähern [27], so dass sich das optimale Regelgesetz beliebig genau approximieren lässt. Naturgemäß steigt der Rechen- und Speicher-

aufwand exponentiell mit der Dimension der kontinuierlichen Teilräume [26], so dass die Einsetzbarkeit auf niedrigdimensionale Systeme limitiert ist. Verschiedene Ansätze zur Reduktion dieses Problems wurden vorgeschlagen, wie etwa die Verwendung einer relaxierten Fassung der dynamischen Programmierung [25], durch die die Wertefunktion auch für größere Systeme approximiert werden kann.

3.2 Minimum-Prinzip, indirekte und direkte Methoden

Eine andere Klasse von Ansätzen erweitert das für kontinuierliche Systeme wohlbekannte *Minimum-Prinzip* von L. Pontryagin und Mitarbeitern [24] auf hybride Dynamiken, vorrangig solche mit autonomem Schalten der Dynamiken [32, 28, 22]. *Indirekte* Lösungsmethoden iterieren dabei die Zustands- und Stellgrößenverläufe so, dass die Optimalitätsbedingungen des Minimum-Prinzips schließlich erfüllt werden, siehe z. B. [4, 8]. Hierbei wird das ursprüngliche Problem als mehrfaches Randwertproblem formuliert und numerisch so gelöst, dass die Zu- und Kozustandsverläufe in jeder Phase die Bedingungen erfüllen. Für diese Vorgehensweise ist es meistens nicht leicht, eine zulässige Initialisierung zu finden, aber sie besitzen eine gute Konvergenz gegen eine lokale Lösung bei fester diskreter Zustandsfolge, falls eine solche Initialisierung vorliegt.

Die Notwendigkeit, in der hybriden Optimalsteuerung die diskrete und kontinuierliche Zustandsfolge festzulegen, macht ein hierarchisches Vorgehen naheliegend. So wurde in [12] ein zweistufiger Algorithmus vorgeschlagen, der auf der höheren Ebene die diskrete Zustandsfolge aktualisiert, indem er bei jeder Iteration einen einzelnen diskreten Zustand einfügt. Auf der unteren Ebene optimiert der Algorithmus dann die kontinuierliche Teildynamik für die von der oberen Ebene iterierte diskrete Zustandsfolge. Dagegen behandelt die Arbeit in [22, 23] das Vorgehen, dass für die Optimierung in einem partitionierten kontinuierlichen Zustandsraum eines hybriden Systems mit autonomem Schalten sich die diskrete Zustandsfolge aus der Optimierung der kontinuierlichen Teildynamik ergibt: Wenn ein Kostenabstieg einen Transitionspunkt über einen Eckpunkt der Transitionsgrenze verschiebt, so ändern sich implizit die Abfolge der diskreten Zustände und somit auch die Beschränkungen für den kontinuierlichen Zustands in aufeinanderfolgenden Phasen.

Im Gegensatz zu den indirekten Verfahren legen die *direkten Methoden* nicht die Optimalitätsbedingungen sondern die ursprüngliche Problemdefinition zu Grunde. Bei einem kontinuierlichen Zeitmodell wird zunächst

eine Zeitdiskretisierung durchgeführt und das Optimierungsproblem damit in eine endlich-dimensionale Fassung transformiert. Dieses Problem wird dann mit Methoden der direkten Kollokation [7] oder über Mehrfach-Schießverfahren gelöst [17]. Dabei werden die Werte des kontinuierlichen Zustands sowie der Eingangstrajektorie für jeden diskreten Zeitpunkt solange iteriert, bis Übergangsbedingungen erfüllt sind und keine signifikanten Kostenverbesserungen mehr erreicht werden. Die Transitionen (einschließlich der Schaltbedingungen und Zustandssprünge) werden während der Optimierung als Randbedingungen formuliert. Im Vergleich zu den indirekten Methoden sind direkte Methoden meist einfacher zu initialisieren und haben einen größeren Konvergenzbe- reich, führen aber typischerweise auf sehr umfangreiche Probleme in Bezug auf die Anzahl der Variablen und Beschränkungen.

3.3 Gemischt-ganzzahlige Programmierung und Graphensuche

Eine weitere Klasse von Ansätzen formuliert die gestellte Aufgabe in zeitdiskreter Formulierung als gemischt ganzzahliges Optimierungsproblem mit linearisierten Beschränkungen, u. a. in [3, 30]. Darin werden die logischen Bedingungen für die Zuordnungen der kontinuierlichen Dynamiken zu den diskreten Zuständen und Invarianten sowie für die Transitionen und Sprungfunktionen über binäre Variablen und algebraische Ungleichungen formuliert. Die resultierenden gemischt-ganzzahligen Optimierungsprobleme kodieren üblicherweise eine exponentiell anwachsende Kombinatorik mit steigendem Zeit- horizonz der Optimierung [9, 33]. Zur Abschwächung dieses Problems wurden verschiedene Maßnahmen vorgeschlagen. So benutzt [31] eine disjunktive Formulierung der gemischt-ganzzahligen Beschränkungen, was innerhalb von Branch-and-Bound-Algorithmen auf engere Relaxierungen der binären Variablen und somit kleinere Anteile zu durchsuchender Bäume führen kann. Ähnlich den Relaxierungen der binären Variablen in gemischt-ganzzahligen Programmen wird in der einbettenden Methode (siehe [21] hinsichtlich eines Vergleichs mit anderen Verfahren), eine Mittelung über mehrere diskrete Optionen von Gradientenfeldern vorgenommen. Eine Erweiterung auf Probleme mit Zustandsbeschränkungen erscheint hierbei nur in Sonderfällen möglich.

In einem anderen Ansatz zur optimalen Steuerung hybrider Systemen mit diskreten Eingängen (für gesteuertes Schalten) wird ein Graphensuche-Algorithmus formuliert

[29]. Ein Zweig des Graphen kodiert eine Sequenz diskreter Eingänge, und in jedem Knoten werden durch nicht-lineare Programmierung die kontinuierlichen Stellgrößen optimiert. Unter den Knoten, die sich auf nah beieinander liegende kontinuierliche Zustände beziehen, wird nur der hinsichtlich seiner Kosten lokal beste Knoten für die folgenden Berechnungen weiterverwendet.

Für hybride Systeme ohne diskrete Eingänge wird in [2] alternativ vorgeschlagen, über die multi-parametrische quadratische Programmierung [1] explizite affine Regelgesetze auf einem partitionierten kontinuierlichen Zustandsraum zu berechnen. Allerdings sind diesem Vorgehen durch die Rechenkomplexität ab mittlerer Zustandsraumdimension ebenfalls Grenzen gesetzt. Die Arbeit [16, 34] verwendet lineare temporale Logik (LTL) als formales Beschreibungsmittel, um allgemeinere Spezifikationen der optimalen Regelung umzusetzen.

Die Problematik einer effizienten Lösung von Optimalsteuerungsproblemen ist gerade bei diskreten Eingängen noch nicht befriedigend gelöst. Daher behandelt der folgende Abschnitt eine Methode, in der durch die Verwendung von approximierenden Kostenschranken der zu durchsuchende Graph möglicher Eingangssequenzen oft deutlich reduziert werden kann. Diese Methode basiert auf einem Ansatz, der in [19] für die prädiktive Regelung hybrider Systeme vorgeschlagen wurde.

4 Ein Ansatz zur Optimalsteuerung geschalteter Systeme

Diesem Abschnitt liegt eine zeitdiskrete Variante des in Kap. 2 skizzierten Problems zu Grunde. Wir beziehen uns dabei auf eine Unterklasse von HA , in der die kontinuierliche Dynamik zeitdiskret vorliegt, die Matrizen von in x_k und u_k linearen Dynamiken durch einen diskreten Eingang selektiert werden können und in der die Zustandsbeschränkungen zeitvariant sind:

$$\begin{aligned} x_{k+1} &= A_{v_k} x_k + B_{v_k} u_k \\ x_k &\in X_k, u_k \in U, v_k \in V \end{aligned} \quad (2)$$

mit $k \in \mathbb{N}^{\geq 0}$. Eine solche Modellform ist z. B. dann sinnvoll, wenn die Dynamik eines autonomen Fahrzeugs oder Roboters in zeitdiskreter linearer Form repräsentiert wird, über v_k der Bewegungsmodus (oder Gang) selektiert werden kann und sich der freie Raum für die Bewegung (also die Menge X_k) aus der Bewegung anderer Fahrzeuge ergibt. Der diskrete Eingang v_k selektiert hier also den diskreten Zustand x_k . Wir nehmen an, dass die Mengen X_k so

wie U als konvexe Mengen modelliert sind. Die X_k sind in Bezug auf die Definition von HA in Kap. 2 als zeitvariable Invarianten zu verstehen. Zudem sei es hier möglich, zu jedem Zeitpunkt k über v_k die Dynamik zu selektieren (entsprechend der Funktion f_d), so dass die Übergangsmengen $G_{z,\bar{z}}$ ebenfalls den Mengen X_k entsprechen. Die Sprungfunktion r_c ist hier zur Vereinfachung der Darstellung als Identitätsfunktion angesetzt, wobei der folgende Ansatz leicht auf allgemeinere Fälle erweiterbar ist. Die Selektion der Matrizen A_{v_k} und B_{v_k} macht die rechte Seite der Differenzgleichungen (also der Funktion f_c in der Definition von HA) nichtlinear.

Wenn Trajektorien der Eingangsgrößen $\phi_{0,N}^u = (u_0, \dots, u_{N-1})$ und $\phi_{0,N}^v = (v_0, \dots, v_{N-1})$ vorgegeben werden, kann die Trajektorie der kontinuierlichen Zustände $\phi_{0,N}^x = (x_0, x_1, \dots, x_N)$ für $k \in \{0, \dots, N-1\}$ eindeutig bestimmt werden über¹:

$$x_{k+1} = \left(\prod_{l=0}^k A_{v_l} \right) \cdot x_0 + \sum_{l=0}^k \left[\left(\prod_{t=0}^{k-l-1} A_{v_{k-t}} \right) \cdot B_{v_l} \cdot u_l \right]. \quad (3)$$

Wir betrachten hier als spezifische Instanz des Problems (1) ein Optimalsteuerungsproblem, in dem das geschaltete System (2) innerhalb eines endlichen Horizonts $k \in \{1, \dots, N\}$ von einem Initialzustand $s_0 = (z_0, x_0)$ kostenoptimal in einen Endzustand $x_f \in X_N$ überführt werden soll. Dazu wird die quadratische Kostenfunktion:

$$J(x_0, x_f, \phi_{0,N}^v, \phi_{0,N}^u) = (x_N - x_f)^T Q_N (x_N - x_f) + \sum_{k=0}^{N-1} \underbrace{(x_k - x_f)^T Q_1 (x_k - x_f) + u_k^T Q_2 u_k}_{\text{Schritt-kosten: } \mathcal{L}(x_k, u_k)} \quad (4)$$

mit Gewichtungsmatrizen $Q_1 = Q_1^T \geq 0$, $Q_2 = Q_2^T > 0$ und $Q_N = Q_N^T > 0$ (sowie einem Referenzeingang $u_f = 0^{n_u \times 1}$) verwendet. Hiermit ergibt sich als betrachtetes Optimalsteuerungsproblem:

$$\begin{aligned} \min_{\phi_{0,N}^v, \phi_{0,N}^u} J(x_0, x_f, \phi_{0,N}^v, \phi_{0,N}^u) \quad (5) \\ \text{u. B. v.: } (3), x_k \in X_k, k \in \{1, \dots, N\}, \\ u_k \in U, v_k \in V, k \in \{0, \dots, N-1\}. \end{aligned}$$

Solche Probleme können durch Lösungsverfahren für gemischt-ganzzahlige Programme (MINLP: Mixed Integer Nonlinear Programming) gelöst werden, aber die Kombinatorik in Bezug auf $\phi_{0,N}^v$ führt typischerweise zu großen Berechnungszeiten, d. h. gemäß n_v^N liegt ein mit N exponentieller Anstieg der Zahl möglicher $\phi_{0,N}^v$ vor.

¹ Merke, dass $\prod_{t \in \emptyset} A_t = 1$.

Zur Lösung des Problems schlagen wir hier einen Graphensuche-Algorithmus vor, in dem zwei Heuristiken genutzt werden, um in der Menge der möglichen $\phi_{0,N}^v$ effizient eine Trajektorie zu identifizieren, die der optimalen Lösung von (5) nahe kommt. Die erste Heuristik schließt Teile des Suchbaums über obere und untere Kostenschranken aus, wobei die Berechnung der Schranken durch einen Mechanismus für die unbeschränkte Optimierung geschalteter Systeme in [11] motiviert ist. Die zweite Heuristik, die der Idee in [29] folgt, benutzt Nachbarschaften im kontinuierlichen Zustandsraum, um die weitere Suche auf die lokal beste Stellgrößentrajektorie zu beschränken.

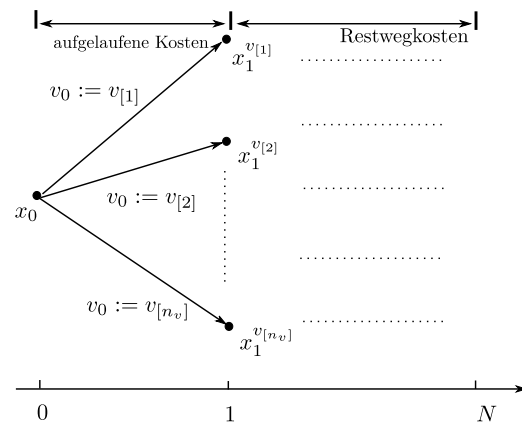


Abb. 1: Teil des Baumes \mathcal{T} mit Knotenmengen $G_0 = \{x_0\}$, $G_1 = \{x_1^{v[1]}, \dots, x_1^{v[n_v]}\}$ und Kantenmenge: $E_1 = V = \{v_{[1]}, \dots, v_{[n_v]}\}$.

Als Grundlage der Suche definieren wir einen Baum über $\mathcal{T} = \{\mathcal{G}, \mathcal{E}\}$ bestehend aus einer Menge von Knoten \mathcal{G} und einer Menge von Kanten \mathcal{E} , siehe auch Abb. 1. Jeder Knoten repräsentiert einen kontinuierlichen Zustand x_k , der durch ein Paar von Eingangssequenzen $\phi_{0,k}^u := \{u_0, \dots, u_{k-1}\}$ und $\phi_{0,k}^v := \{v_0, \dots, v_{k-1}\}$ erreichbar ist. Mit $G_k \subset \mathcal{G}$ wird für $k \in \{1, \dots, N\}$ die Teilmenge von Knoten bezeichnet, die jeweils durch ein $\phi_{0,k}^v$ erkundet werden, wobei sich das zugeordnete x_k aus dem für $\phi_{0,k}^v$ optimalen $\phi_{0,k}^u$ ergibt. Für den Wurzelknoten des Baumes gilt $G_0 = \{x_0\}$. In der Menge $\mathcal{E} = \{E_0, \dots, E_{N-1}\}$ enthält jedes Element E_k jene Kanten, die für einen diskreten Eingang $v \in V$ stehen. Eine Randbedingung hierbei ist, dass $v \in V$ dann aus E_{k-1} auszuschließen ist, wenn kein $u_{k-1} \in U$ existiert, für das mit $x_{k-1} \in G_{k-1}$ die Bedingung $x_k \in X_k$ erfüllbar ist.

Der Wahl eines Paares $\phi_{0,N}^v$ und $\phi_{0,N}^u$ sind dann bestimmte Sequenzen von Knoten und Kanten zugeordnet, die Sequenz der x_k ist gemäß (3) eindeutig festgelegt und die Kosten $J(x_0, x_f, \phi_{0,N}^v, \phi_{0,N}^u)$ bestimmt. Ein effizienter Algorithmus versucht offensichtlich, den kleinstmög-

chen Kostenwert $J^*(x_0, x_f, \phi_{0,N}^{V,*}, \phi_{0,N}^{U,*})$ nach Exploration eines möglichst kleinen Anteils von \mathcal{T} möglichst gut anzunähern.

4.1 Untere Kostenschranken

Um die Exploration des Suchbaums zu begrenzen, werden Kostenschranken benötigt, die anzeigen, welche Teile von \mathcal{T} nicht zu einer optimalen Lösung gehören können. Generell lassen sich für jedes $x_k \in \mathbb{G}_k$ mit $k \in \{1, \dots, N-1\}$ die Kosten in einen Anteil aufgelaufener Kosten (für den Weg von x_0 bis x_k) und die Restwegkosten (von x_k bis x_N) aufteilen.² Der bestmögliche Wert für den Restweg wird durch die Wertefunktion beschrieben, vergleiche z. B. [36]:

$$\mathcal{V}(x_k) := \min_{\phi_{k,N}^V, \phi_{k,N}^U} \left\{ \sum_{i=k}^{N-1} \mathcal{L}(x_i, u_i) + (x_N - x_f)^T Q_N (x_N - x_f) \right\} \quad (6)$$

u. B. v.: (3), $u_i \in U$, $v_i \in V$, $\forall i \in \{k, \dots, N-1\}$,
 $x_i \in X_i$, $\forall i \in \{k+1, \dots, N\}$.

Für $k = 0$ ist $\mathcal{V}(x_0)$ identisch zur besten Lösung des Problems (5), also gleich $J^*(x_0, x_f, \phi_{0,N}^{V,*}, \phi_{0,N}^{U,*})$. Wenn x_{k+1} durch $u_k \in U$ und $v_k \in V$ von x_k aus erreichbar ist, dann gilt auch:

$$\mathcal{V}(x_k) = \min_{v_k, u_k} \{ \mathcal{L}(x_k, u_k) + \mathcal{V}(x_{k+1}) \} \quad (7)$$

u. B. v.: $x_{k+1} = A_{v_k} x_k + B_{v_k} u_k \in X_{k+1}$
 $x_k \in X_k$, $u_k \in U$, $v_k \in V$.

Eine untere Schranke von $\mathcal{V}(x_k)$ ergibt sich offensichtlich, wenn die Beschränkungen $u_k \in U$ und $x_k \in X_k$ in (6) nicht berücksichtigt werden:

$$\mathcal{V}^{un}(x_k) := \min_{\phi_{k,N}^V, \phi_{k,N}^U} \left\{ \sum_{i=k}^{N-1} \mathcal{L}(x_i, u_i) + (x_N - x_f)^T Q_N (x_N - x_f) \right\} \quad (8)$$

u. B. v.: (3), $v_i \in V$, $\forall i \in \{k, \dots, N-1\}$,

da der vergrößerte Suchraum impliziert, dass $\mathcal{V}^{un}(x_k) \leq \mathcal{V}(x_k)$. Die Differenz zwischen $\mathcal{V}^{un}(x_k)$ und $\mathcal{V}(x_k)$ hängt von dem Ausmaß ab, mit dem die betreffenden Beschränkungen entlang des Systemablaufs aktiv sind. Die Verwendung von $\mathcal{V}^{un}(x_k)$ als untere Kostenschranke in einer Baumsuche ist dann effektiv, wenn $\mathcal{V}(x_k) - \mathcal{V}^{un}(x_k)$ klein

² Zur Vereinfachung der Notation schreiben wir x_k statt $x_k^{v_{[q]}}$, wenn die Referenz zu einem bestimmten $v_{[q]} \in V$ nicht erforderlich ist.

ist und $\mathcal{V}^{un}(x_k)$ einfach berechnet werden kann. Wie z. B. in [36] diskutiert, kann $\mathcal{V}^{un}(x_k)$ über die *Differenzen-Riccati-Gleichungen* erhalten werden:

1. für Schritt N mit $\mathcal{P}_N^{un,*} := Q_N$:

$$\mathcal{V}^{un}(x_N) := \min_{x_N} (x_N - x_f)^T \mathcal{P}_N^{un,*} (x_N - x_f); \quad (9)$$

2. für $i \in \{N, \dots, k+1\}$ und für jedes $v_{[q]} \in V$ Bestimmung einer Matrix $\mathcal{P}_{i-1}^{un, v_{[q]}}$ gemäß:

$$\mathcal{P}_{i-1}^{un, v_{[q]}} = A_{v_{[q]}}^T \mathcal{P}_i^{un,*} A_{v_{[q]}} + 2\mathcal{K}_{i-1}^T B_{v_{[q]}}^T \mathcal{P}_i^{un,*} A_{v_{[q]}} + Q_1 + \mathcal{K}_{i-1}^T (B_{v_{[q]}}^T \mathcal{P}_i^{un,*} B_{v_{[q]}} + Q_2) \mathcal{K}_{i-1} \quad (10)$$

mit:

$$\mathcal{K}_{i-1} = -(B_{v_{[q]}}^T \mathcal{P}_i^{un,*} B_{v_{[q]}} + Q_2)^{-1} B_{v_{[q]}}^T \mathcal{P}_i^{un,*} A_{v_{[q]}}. \quad (11)$$

Diese Rückwärtsberechnung liefert für den Zeitpunkt k eine mit \mathbb{P}_k^{un} bezeichnete Menge von n_v^{N-k} Matrizen. Daraus kann $\mathcal{V}^{un}(x_k)$ bestimmt werden über:

$$\mathcal{P}_k^{un,*} = \arg \min_{\mathcal{P}_k^{un, v_{[q]}} \in \mathbb{P}_k^{un}} (x_k - x_f)^T \mathcal{P}_k^{un, v_{[q]}} (x_k - x_f),$$

$$\mathcal{V}^{un}(x_k) = (x_k - x_f)^T \mathcal{P}_k^{un,*} (x_k - x_f). \quad (12)$$

Diese Berechnungsprozedur ist auch in Abb. 2 dargestellt.

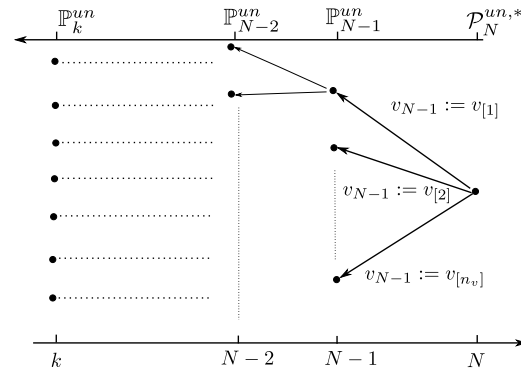


Abb. 2: Rückwärtsberechnung für $\mathcal{V}^{un}(x_k)$.

Die aus der Kombinatorik resultierende exponentiell zunehmende Rechenkomplexität ist für größere N wieder prohibitiv, so dass wir folgende Modifikation einbringen: Entsprechend (7) gilt auch für den bezüglich x_k , x_{k+1} und u_k unbeschränkten Fall:

$$\mathcal{V}^{un}(x_k) := \min_{v_k, u_k} \{ \mathcal{L}(x_k, u_k) + \mathcal{V}^{un}(x_{k+1}) \}$$

u. B. v.: $x_{k+1} = A_{v_k} x_k + B_{v_k} u_k$, $v_k \in V$, (13)

woraus unter Verwendung von (12) folgt:

$$\begin{aligned} \mathcal{V}^{un}(x_k) := & \quad (14) \\ & \min_{v_k, u_k} \{ \mathcal{L}(x_k, u_k) + (x_{k+1} - x_f)^T \mathcal{P}_{k+1}^{un,*} (x_{k+1} - x_f) \} \\ \text{u. B. v.: } x_{k+1} = & A_{v_k} x_k + B_{v_k} u_k, \quad v_k \in V. \end{aligned}$$

Falls $\mathcal{P}_{k+1}^{un,*}$ bekannt wäre, ergäbe sich für ein gegebenes x_k die optimale Zustandsfolge des unbeschränkten Problems zu $\phi_{k,N}^{x,un,*} = (x_k, x_{k+1}^{un,*}, \dots, x_{k+N}^{un,*})$ somit durch die Bestimmung von $(u_k^{un,*}, v_k^{un,*})$ im ersten Schritt und Selektion der besten Matrix $\mathcal{P}_{k+1}^{un,*}$ für den Restweg. Der zweite Term von (14) wird jetzt (zur Vermeidung der Kombinatorik) angenähert durch:

$$\min_{\substack{\mathcal{P}_{k+1, v_{[q]}}^{un} \\ \in \mathbb{P}_{k+1}^{un}}} (x_{k+1} - x_f)^T \mathcal{P}_{k+1, v_{[q]}}^{un} (x_{k+1} - x_f), \quad (15)$$

d. h. aus \mathbb{P}_{k+1}^{un} wird das Element ausgewählt, das für den Restweg optimal ist. Die Näherung besteht darin, dass (15) nur für das Vorliegen von $x_{k+1}^{un,*}$ dem zweiten Term von (14) entspricht; dem steht der Vorteil gegenüber, dass bei Auswahl der jeweils besten Matrix $\mathcal{P}_{k+i, v_{[q]}}^{un}$ aus \mathbb{P}_{k+i}^{un} bei Rückwärtslaufen von $i = N - k$ bis $i = 1$ der Aufwand linear mit $N - k$ steigt. Da die Zustandssequenz $\phi_{k+1, N}^{x,*}$ zur Auswertung von (15) noch nicht bekannt ist, wird die Matrixauswahl durch Minimierung der Spur gemäß:

$$\mathcal{P}_{k+i-1}^{un,apx} := \arg \min_{\substack{\mathcal{P}_{k+i, v_{[q]}}^{un} \\ \in \mathbb{P}_{k+i}^{un}}} \text{trace}(\mathcal{P}_{k+i}^{un, v_{[q]}}) \quad (16)$$

vorgenommen. Die iterative Ausführung von (16) für $i \in \{N - k, \dots, 1\}$ führt auf eine Matrix $\mathcal{P}_k^{un,apx}$, welche zur Näherung der untere Kostenschranke $\mathcal{P}_k^{un,*}$ genutzt werden kann:

$$\begin{aligned} \underline{\mathcal{V}}(x_k) := & (x_k - x_f)^T \mathcal{P}_k^{un,apx} (x_k - x_f) \quad (17) \\ \approx & (x_k - x_f)^T \mathcal{P}_k^{un,*} (x_k - x_f) = \mathcal{V}^{un}(x_k) \leq \mathcal{V}(x_k). \end{aligned}$$

Die Verwendung dieser Näherung erscheint gerechtfertigt, da es sich nur um die Behandlung einer Kostenschranke handelt und die Aufwandsreduktion erheblich ist: Er reduziert sich von $\sum_{i=1}^{N-k} n_v^i$ auf $(N - k) \cdot n_v$ Rechenschritte, d. h. die Berechnung von $\underline{\mathcal{V}}(x_k)$ benötigt typischerweise nur einen geringen Anteil der Gesamtrechnzeit. Die der Gl. (17) zugeordnete Sequenz diskreter Eingänge wird mit $\phi_{k,N}^{v,apx} = (v_k^{apx}, \dots, v_{N-1}^{apx})$ bezeichnet.

4.2 Obere Kostenschranken

Im Laufe der Suche nach $\phi_{0,N}^{v,*}$ und $\phi_{0,N}^{u,*}$ und damit der Exploration von \mathcal{T} werden (sofern existent) sukzessive Paare diskreter und kontinuierlicher Stellgrößentrajektorien

bestimmt, die zulässige Abläufe des geschalteten Systems in dem Sinne determinieren, dass sämtliche Beschränkungen des Problems (5) eingehalten werden. Die in jeder Iteration der Suche bis dahin beste gefundene zulässige Lösung $\hat{\phi}_{0,N}^v$ und $\hat{\phi}_{0,N}^u$ mit Kosten $\hat{J}(x_0, x_N, \hat{\phi}_{0,N}^v, \hat{\phi}_{0,N}^u) \geq \mathcal{V}(x_0)$ kann als obere Kostenschranke verwendet werden, d. h. jedes nachfolgend betrachtete Paar $\hat{\phi}_{0,\tilde{N}}^v$ und $\hat{\phi}_{0,\tilde{N}}^u$ mit Kosten $\hat{J}(x_0, x_{\tilde{N}}, \hat{\phi}_{0,\tilde{N}}^v, \hat{\phi}_{0,\tilde{N}}^u) > \hat{J}(x_0, x_N, \hat{\phi}_{0,N}^v, \hat{\phi}_{0,N}^u)$ ist suboptimal und braucht nicht weiter betrachtet zu werden. Dies gilt insbesondere auch für $\tilde{N} < N$, d. h. zeitlich kürzere Pfade mit höheren Kosten als $\hat{J}(x_0, x_N, \hat{\phi}_{0,N}^v, \hat{\phi}_{0,N}^u)$ brauchen nicht weiter ergänzt zu werden.

Wenn eine bestimmte diskrete Eingangssequenz $\phi_{k,N}^v$ mit $k \in \{0, \dots, N - 1\}$ im Laufe der Suche exploriert wird, verbleibt die Bestimmung von $\phi_{0,N}^u$, um auch die kontinuierliche Zustandssequenz $\phi_{0,N}^x$ festzulegen. Das entsprechende Optimierungsproblem lautet:

$$\begin{aligned} \bar{\mathcal{V}}(x_k) = \min_{\phi_{k,N}^u} & \left\{ \sum_{i=k}^{N-1} \mathcal{L}(x_i, u_i) + (x_N - x_f)^T Q_N (x_N - x_f) \right\} \\ \text{u. B. v.: } & (3), \quad x_i \in X_i, \quad i \in \{k + 1, \dots, N\}, \quad (18) \\ & \phi_{k,N}^{v,apx}, \quad u_i \in U, \quad i \in \{k, \dots, N - 1\}. \end{aligned}$$

Falls keine Lösung für dieses Problem existiert, wird $\bar{\mathcal{V}}(x_k) := \infty$ zugewiesen. Falls sämtliche Mengen U und X_k , $k \in \{1, \dots, N\}$ polytopisch gegeben sind (somit die Beschränkungen lineare Form besitzen), liegt die Bestimmung von $\phi_{k,N}^u$ als quadratisches Programm (QP) vor – der Rechenzeitanteil ist daher auch für die Bestimmung dieser Schranken typischerweise gering.

Zusammenfassend werden die Restwegkosten in der Suche für jeden Knoten x_k damit begrenzt durch:

$$\underline{\mathcal{V}}(x_k) \leq \mathcal{V}(x_k) \leq \bar{\mathcal{V}}(x_k). \quad (19)$$

4.3 Suchverfahren

Die zuvor berechneten Schranken der Restwegkosten werden nun in einem Baumsuche-Algorithmus verwendet, um die Anzahl der explorierten Knoten für die Bestimmung der Lösung von Problem (5) zu reduzieren. Ausgehend vom Wurzelknoten x_0 werden für jeden Knoten x_k auf Schicht k die akkumulierten Kosten, mit denen die Teilstrategien $\phi_{0,k}^u$ und $\phi_{0,k}^v$ den Zustand in x_k überführen, berechnet aus:

$$J(x_0, x_k, \phi_{0,k}^v, \phi_{0,k}^u) = \sum_{i=0}^{k-1} \mathcal{L}(x_i, u_i) \quad (20)$$

$$\text{u. B. v.: } (3), \quad u_i \in \phi_{0,k}^u, \quad v_i \in \phi_{0,k}^v, \quad i \in \{0, \dots, k - 1\}$$

Die beste Einschnittstrategie zur Überführung von x_k zu $x_{k+1}^{*,v_{[q]}}$ ergibt sich aus:

$$\min_{u_k \in U} \{ \mathcal{L}(x_k, u_k) + \mathcal{V}(x_{k+1}) \} \quad (21)$$

$$\text{u. B. v.: } x_{k+1} = A_{v_{[q]}} x_k + B_{v_{[q]}} u_k, x_{k+1} \in X_{k+1}.$$

Da $\mathcal{V}(x_{k+1})$ hier noch nicht bekannt ist, wird es in der ersten Zeile von (21) durch die Summe seiner oberen und unteren Schranke gemäß (17) und (18) ersetzt:

$$\min_{u_k \in U} \left\{ \frac{\mathcal{L} + \underline{\mathcal{V}}(x_{k+1})}{\text{untere Grenze}} + \frac{\mathcal{L} + \overline{\mathcal{V}}(x_{k+1})}{\text{obere Grenze}} \right\} \quad (22)$$

Diese Formulierung (als eine mögliche Option) minimiert gleichzeitig beide Schranken des neuen Knotens und ist relativ einfach lösbar, da es bei polytopischen Beschränkungen wieder nur ein QP-Problem darstellt. Der neue Knoten $x_{k+1}^{*,v_{[q]}}$ wird dann in die Menge G_{k+1} des Baumes eingefügt.

Für den Vergleich zweier diskreter Eingänge zum Zeitpunkt k gilt offensichtlich:

Lemma 1. *Seien $x_{k+1}^{v_{[m]}}$ und $x_{k+1}^{v_{[n]}}$ zwei verschiedene Knoten, die sich für $k+1$ als Nachfolger des Knotens x_k für zwei verschiedene diskrete Eingänge $v_k := v_{[m]}$ bzw. $v_k := v_{[n]}$ ergeben. Wenn dann gilt, dass $\underline{\mathcal{V}}(x_{k+1}^{v_{[m]}}) > \overline{\mathcal{V}}(x_{k+1}^{v_{[n]}})$, dann folgt auch: $\mathcal{V}(x_{k+1}^{v_{[m]}}) > \mathcal{V}(x_{k+1}^{v_{[n]}})$.*

Sind nun zusätzlich $u_k := u_{[m]}$ bzw. $u_k := u_{[n]}$ festgelegt, so sind $x_{k+1}^{v_{[m]}}$ und $x_{k+1}^{v_{[n]}}$ sowie die zugeordneten Kosten berechenbar. Die gesamten Kosten $J(x_0, x_f, \phi_{0,N}^v, \phi_{0,N}^u)$ für einen Weg über $x_{k+1}^{v_{[m]}}$ können formuliert werden zu:

$$\begin{aligned} J(x_0, x_f, \phi_{0,N}^v, \phi_{0,N}^u) \\ = J(x_0, x_k, \phi_{0,k}^v, \phi_{0,k}^u) + \mathcal{L}(x_k, u_{[m]}) + \mathcal{V}(x_{k+1}^{v_{[m]}}). \end{aligned} \quad (23)$$

Für den Anteil $\mathcal{V}(x_{k+1}^{v_{[m]}})$ kann man den möglichen Wertebereich durch (19) bestimmen, so dass auch der mögliche Wertebereich von $J(x_0, x_f, \phi_{0,N}^v, \phi_{0,N}^u)$ bestimmbar ist. Bei gleicher Berechnung von $J(x_0, x_f, \tilde{\phi}_{0,N}^v, \tilde{\phi}_{0,N}^u)$ für die alternative Wahl $v_k := v_{[n]}$ und $u_k := u_{[n]}$ als letzte Einträge in Strategien $\tilde{\phi}_{0,k+1}^v$ und $\tilde{\phi}_{0,k+1}^u$ gilt folgende Betrachtung: Wenn die untere Grenze von $J(x_0, x_f, \phi_{0,N}^v, \phi_{0,N}^u)$ höher als die obere Grenze von $J(x_0, x_f, \tilde{\phi}_{0,N}^v, \tilde{\phi}_{0,N}^u)$ ist, so braucht nach Lemma 1 der Knoten $x_{k+1}^{v_{[m]}}$ (sowie mögliche Nachfolger) nicht mehr weiter betrachtet werden, d. h. er entfällt aus G_{k+1} .

Als eine weitere Möglichkeit den Suchraum zu reduzieren, kann man das Konzept *lokal bester benachbarter Knoten* aus [29] auf die in G_{k+1} verbleibenden Knoten anwenden: Zwei Knoten $x_{k+1}^{v_{[m]}}$ und $x_{k+1}^{v_{[n]}}$ in G_{k+1} werden *benachbart* genannt, wenn:

$$\|x_{k+1}^{v_{[m]}} - x_{k+1}^{v_{[n]}}\|_2 \leq \gamma \quad (24)$$

für ein geeignet gewähltes $\gamma > 0$ gilt. Für jedes Paar *benachbarter* Zustände wird nur der in G_{k+1} beibehalten, der die kleineren akkumulierten Kosten besitzt.

Der folgende Algorithmus fasst die oben erläuterten Techniken zusammen, um die Menge der möglichen diskreten Eingangsequenzen zu durchsuchen:

- 1: **Gegeben:** $G_0 = \{x_0\}$, U , V ;
- 2: **for** $k = 0 : N - 1$ **do**
- 3: **for** $m = 1 : |G_k|$ **do**
- 4: **for** $q = 1 : n_v$ **do**
- 5: Lösen von (22) für $v_{[q]} \in V$ und $x_k^m \in G_k$;
- 6: Zufügen von $x_{k+1}^{*,v_{[q]}}$ zu G_{k+1} ;
- 7: Bestimmung des Wertebereichs von
- 8: $J(x_0, x_f, \phi_{0,N}^v, \phi_{0,N}^u)$ aus (23), (17)
- 9: und (18);
- 10: **end for**
- 11: **end for**
- 12: Löschen von Knoten aus G_{k+1} gemäß Lemma 1;
- 13: Löschen der Knoten aus G_{k+1} , die (24) erfüllen und
- 14: die höheren *akkumulierten Kosten* besitzen;
- 15: **end for**

Wenn für jeden beibehaltenen Knoten in den Mengen G_k der Vorgängerknoten, die besten Stellgrößenwerte, die in den Knoten führen, und die Kosten abgespeichert werden, kann ausgehend von den Einträgen in G_N abschließend das Paar $(\phi_{0,N}^{v,*}, \phi_{0,N}^{u,*})$ bestimmt werden, welches die Lösung von (5) approximiert.

5 Numerische Beispiele

Um die Leistungsfähigkeit der zuvor beschriebenen Methode zu illustrieren, werden in diesem Abschnitt verschiedene numerische Tests beschrieben. Zunächst wird die Prozedur auf eine Serie von 30 zufällig generierten geschalteten Systemen jeweils mit $n_x = 10$, $n_u = 8$, $n_v = 4$, $N = 6$ sowie Kostenfunktionen und Beschränkungen im Rahmen der Problemdefinition angewandt. Die Abbildung 3 zeigt eine Zusammenfassung der Ergebnisse für $\gamma = 5$ im Hinblick auf die Abweichung von der global optimalen Lösung (die hier für das gewählte N noch berechenbar ist über die vollständige Enumeration der diskreten Eingangsfolgen). Die durchschnittliche Abweichung beträgt 1.58 % bei einer durchschnittlichen Rechenzeit von 0.077 Sekunden. Mit $\gamma = 2$ reduziert sich die durchschnittliche Abweichung auf 1.22 %, während die durchschnittliche Rechenzeit auf 0.099 Sekunden steigt. Die Abb. 4 zeigt den Anstieg der durchschnittlichen Rechenzeit für den Fall, dass der Horizont N von 1 auf 10 steigt.

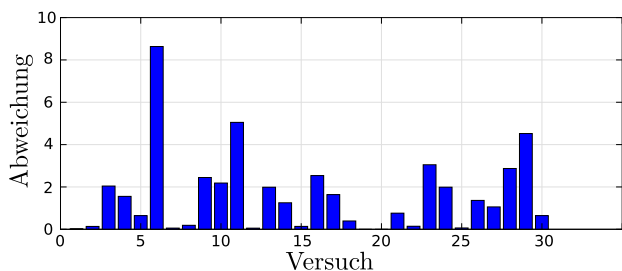


Abb. 3: Prozentuale Abweichung der mit der vorgeschlagenen Methode erzielten Kosten von den global optimalen Lösungen für 30 zufällig generierte geschaltete Systeme.

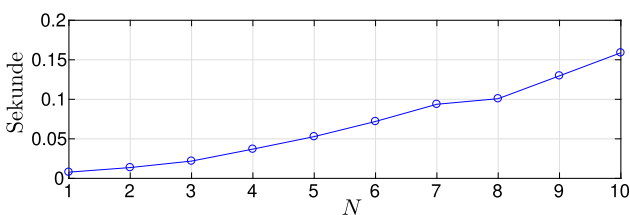


Abb. 4: Durchschnittliche Rechenzeit für 30 zufällig generierte geschaltete Systeme, wenn N von 1 auf 10 ansteigt.

Als nächstes wird ein technisch motiviertes Beispiel betrachtet, und zwar eine erweiterte Fassung des in [13] betrachteten Problems der Regelung der Geradeausfahrt einer Fahrzeug-Kolonne aus $M = 4$ Fahrzeugen (siehe Abb. 5). Entsprechend der Version in [19] ist hier für jedes Fahrzeug als diskreter Eingang die Gangwahl berücksichtigt, so dass die Modellierung als geschaltetes System mit unterschiedlichen, gangabhängigen Beschleunigungsmodi naheliegender ist. Für die Fahrzeuge sind jeweils unterschiedliche Dynamiken, Anzahlen von Gängen und Beschränkungen für mögliche Geschwindigkeiten und Beschleunigungen angesetzt. Als Regelungsziel sollen ausgehend von einer Anfangssituation definierte Zielpositionen erreicht werden. Der Abstand von den Zielpositionen ist über geeignete Kostenfunktionen abgebildet. Zusätzlich darf zur Kollisionsvermeidung jeweils ein Mindestabstand von $d = 25$ nicht unterschritten werden.

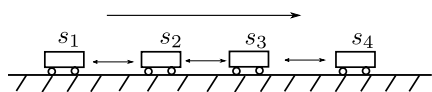


Abb. 5: Schema des Fahrzeugkolonnenbeispiels.

Für die Optimalsteuerung wird hier exemplarisch ein Zeithorizont von $N = 15$ gewählt. Für das gewählte Modell liegen pro Zeitschritt 36 Dynamiken über alle Fahrzeuge

vor, und die Modellierung der möglichen Kombinationen als gemischt-ganzzahliges Programm würde auf 540 binäre Variablen führen. Für $N = 15$ resultieren daraus etwa 2×10^{23} verschiedene diskrete Wertesequenzen für die Optimierung.

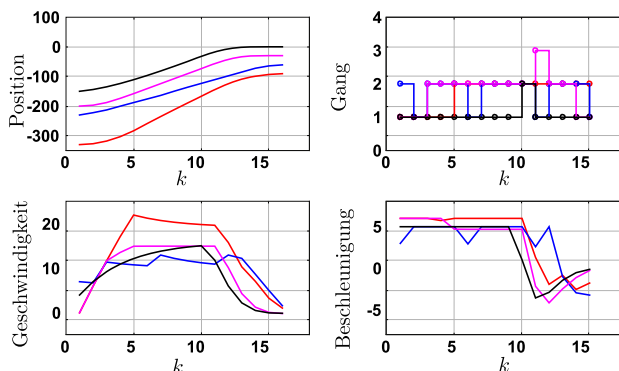


Abb. 6: Zustandsgrößenverläufe der Fahrzeuge über der Zeit k (s_1 rot, s_2 blau, s_3 magenta, s_4 schwarz).

Die in Abschnitt 4 beschriebene Methode wird hier im Sinne der prädiktiven Regelung angewandt, d. h. ein Optimalsteuerproblem gemäß (5) wird in jedem Zeitschritt mit dem Horizont $N = 15$ gelöst. Die Fahrzeuge erreichen ihre Zielpositionen nach 15 Schritten bei Einhaltung aller Beschränkungen. Die Abb. 6 zeigt die resultierenden Zustandsgrößenverläufe der Fahrzeuge. Die mittlere Rechenzeit für die Optimierung in jedem Schritt beträgt 0.96 Sekunden und liegt damit in einem Bereich, der für eine Online-Anwendung geeignet ist.

6 Fazit

In diesem Beitrag wurde ausgehend von einer allgemeinen Spezifikation der Aufgabe der optimalen Regelung / Steuerung hybrider Systeme aufgezeigt, welche Lösungsansätze in der Literatur beschrieben sind. Je nach Zeitmodell, nach Definition der Diskontinuitäten der Dynamik, nach Kostenfunktion und Kombinatorik aus diskreten Dynamiken ergeben sich unterschiedliche Herausforderungen, die in existierenden Arbeiten durch spezifische Vorschläge adressiert wurden. Ein gemeinsames Charakteristikum dieser Arbeiten ist, dass die Anwendbarkeit für die meisten Ansätze wegen der Rechenkomplexität weiterhin auf „kleine“ Probleminstanzen beschränkt ist. Um diese Problematik für die Optimalsteuerung geschalteter Systeme mit zeitdiskreter linearer Dynamik und gemischten Eingängen zu adressieren, wurde hier ein Ansatz der

Graphensuche vorgeschlagen. Die wesentliche Idee dabei ist, untere und obere Kostenschranken für die Restwegkosten der Zwischenzustände über ein einfaches (damit recheneffizientes) Schema zu approximieren, und auf dieser Basis die Menge der zu untersuchenden diskreten Stellsequenzen zu reduzieren. Die kontinuierlichen Stelltrajektorien werden dann jeweils über relativ effizient lösbare, quadratische Programme bestimmt. Hervorzuheben ist, dass die Vorgehensweise (außer dem Parameter γ im zusätzlichen Kriterium für benachbarte Knoten) keine weiteren manuell festzulegenden Parameter erfordert. Wie bei jeder Heuristik lässt sich auch hier für die Approximation der Kostenschranken nicht für den allgemeinen Fall formelmäßig angeben, um welchen Anteil der Suchbaum reduziert wird und welchen Abstand zur global optimalen Lösung existiert. Die in deutlich größerer Anzahl als hier dargestellt ausgeführten Tests weisen jedoch darauf hin, dass das Schema die Rechenzeit effizient reduziert (nicht nur gegenüber der Suche über den vollständigen Baum, sondern auch gegenüber Standardansätzen der gemischt-ganzzahligen Programmierung), während der Abstand zu global optimalen Lösung relativ gering bleibt.

Eine Modifikation des in Kap. 4 beschriebenen Ansatzes für die Optimierung auf gleitenden Horizonten im Sinne der prädiktiven Regelung ist in [19] beschrieben. Weiterer Forschungsbedarf besteht allerdings weiterhin für effiziente Algorithmen zur Lösung von Problemen zur optimalen Steuerung / Regelung von hybriden Systemen, die geschaltetes mit autonom schaltendem Verhalten verbinden.

Finanzierung: Diese Untersuchung wurde zum Teil durch die Europäische Kommission im Rahmen des Projekts UNCoVERCPS (Bewilligung Nr. 643921) gefördert.

Literatur

1. J. Acevedo and E. Pistikopoulos. An algorithm for multiparametric mixed-integer linear programming problems. *Operations Research Letters*, 24(3):139–148, 1999.
2. A. Bemporad, F. Borrelli and M. Morari. On the optimal control law for linear discrete time hybrid systems. In *Hybrid systems: computation and control*, Springer LNCS, pages 105–119, 2002.
3. A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3):407–427, 1999.
4. S. Benghea and R. DeCarlo. Optimal control of switching systems. *Automatica*, 41(1):11–27, 2005.
5. F. Borrelli, M. Baotić, A. Bemporad and M. Morari. Dynamic programming for constrained optimal control of discrete-time linear hybrid systems. *Automatica*, 41(10):1709–1721, 2005.
6. S. Branicky, S. Borkar, and S. Mitter. A unified framework for hybrid control: Model and optimal control theory. *IEEE Trans. on Automatic Control*, 43(1):31–45, 1998.
7. M. Buss, O. von Stryk, R. Bulirsch and G. Schmidt. Towards hybrid optimal control. *at-Automatisierungstechnik*, 48(9/2000):448, 2000.
8. G. Cassandras, L. Pepyne and Y. Wardi. Optimal control of a class of hybrid systems. *IEEE Trans. on Automatic Control*, 46(3):398–415, 2001.
9. A. Del Pia, S. Dey and M. Molinaro. Mixed-integer quadratic programming is in NP. *Mathematical Programming*, 162(1–2):225–240, 2017.
10. R. Goebel, R. Sanfelice and A. Teel. Hybrid dynamical systems. *IEEE Control Systems Magazine*, 29(2):28–93, 2009.
11. D. Goerges, M. Izák and S. Liu. Optimal control and scheduling of switched systems. *IEEE Trans. on Automatic Control*, 56(1):135–140, 2011.
12. H. Gonzalez, R. Vasudevan, M. Kamgarpour, S. S. Sastry, R. Bajcsy and C. J. Tomlin. A descent algorithm for the optimal control of constrained nonlinear switched dynamical systems. In *13th ACM Int. Conf. on Hybrid Systems: Computation and Control*, pages 51–60, 2010.
13. D. Groß and O. Stursberg. Distributed predictive control for a class of hybrid systems with event-based communication. *IFAC Proceedings*, 46(27):383–388, 2013.
14. S. Hedlund and A. Rantzer. Optimal control of hybrid systems. In *38th IEEE Conf. on Decision and Control*, volume 4, pages 3972–3977, 1999.
15. T. A. Henzinger. The Theory of Hybrid Automata. In *Proc. 11th IEEE Symp. on Logic in Comp. Science*, pages 278–292, 1996.
16. S. Karaman, R. Sanfelice and E. Frazzoli. Optimal control of mixed logical dynamical systems with linear temporal logic specifications. In *47th IEEE Conference on Decision and Control*, pages 2117–2122, 2008.
17. C. Kirches, H. G. Bock and S. Leyffer. Modeling mixed-integer constrained optimal control problems in AMPL. *IFAC Proceedings*, 45(2):1124–1129, 2012.
18. X. Koutsoukos. Optimal control of stochastic hybrid systems based on locally consistent markov decision processes. In *IEEE Mediterrean Conf. on Control and Automation*, pages 435–440, 2005.
19. Z. Liu and O. Stursberg. Optimizing online control of constrained systems with switched dynamics. In *IEEE European Control Conference*, pages 788–794, 2018.
20. N. Lynch, R. Segala and F. Vaandrager. Hybrid I/O Automata. *Information and Computation*, 185:105–157, 2003.
21. R. T. Meyer, M. Zefran and R. DeCarlo. A comparison of the embedding method with multiparametric programming, mixed-integer programming, gradient-descent and hybrid minimum principle-based methods. *IEEE Trans. on Control Systems Technology*, 22(5):1784–1800, 2014.
22. B. Passenberg, P. Caines, M. Sobotka, O. Stursberg and M. Buss. The minimum principle for hybrid systems with partitioned state space and unspecified discrete state sequence. In *49th IEEE Conf. on Decision and Control*, pages 6666–6673, 2010.
23. B. Passenberg, M. Sobotka, O. Stursberg, M. Buss and P. Caines. An algorithm for discrete state sequence and trajectory optimization for hybrid systems with partitioned state space.

- In *49th IEEE Conf. on Decision and Control*, pages 4223–4229, 2010.
24. L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze and E. F. Mishchenko. *The mathematical theory of optimal processes*. Interscience Pub., 1962.
 25. A. Rantzer. Relaxed dynamic programming in switching systems. *IEE Proc. Control Theory and Applications*, 153(5):567–574, 2006.
 26. M. Rungger and O. Stursberg. Optimal control for deterministic hybrid systems using dynamic programming. *IFAC Proceedings*, 42(17):316–321, 2009.
 27. M. Rungger and O. Stursberg. A numerical method for hybrid optimal control based on dynamic programming. *Nonlinear Analysis: Hybrid Systems*, 5(2):254–274, 2011.
 28. M. Shaikh and P. Caines. On the hybrid optimal control problem: theory and algorithms. *IEEE Trans. on Automatic Control*, 52(9):1587–1603, 2007.
 29. O. Stursberg. A graph search algorithm for optimal control of hybrid systems. In *43rd IEEE Conference on Decision and Control*, volume 2, pages 1412–1417, 2004.
 30. O. Stursberg and S. Engell. Optimal control of switched continuous systems using mixed-integer programming. *IFAC Proceedings*, 35(1):433–438, 2002.
 31. O. Stursberg and S. Panek. Control of switched hybrid systems based on disjunctive formulations. In *Hybrid Systems: Computation and Control*, volume 2289 of *Springer LNCS*, pages 421–435, 2002.
 32. H. Sussmann. A maximum principle for hybrid optimal control problems. In *38th IEEE Conf. on Decision and Control*, volume 1, pages 425–430, 1999.
 33. J. Till, S. Engell, S. Panek and O. Stursberg. Empirical complexity analysis of an MILP-approach for optimization of hybrid systems. *IFAC Proceedings*, 36(6):129–134, 2003.
 34. E. Wolff and R. Murray. Optimal control of nonlinear systems with temporal logic specifications. In *Robotics Research*, volume 114 of *Springer Tracts in Advanced Robotics*, pages 21–37. Springer, 2016.
 35. P. Xu, X. and Antsaklis. A dynamic programming approach for optimal control of switched systems. In *39th IEEE Conf. on Decision and Control*, volume 2, pages 1822–1827, 2000.
 36. W. Zhang and J. Hu. On optimal quadratic regulation for discrete-time switched linear systems. In *Hybrid Systems: Computation and Control*, Springer LNCS, pages 584–597, 2008.

Autoreninformationen



Zonglin Liu

Regelungs- und Systemtheorie, FB Elektrotechnik/Informatik, Universität Kassel, D-34109 Kassel, Germany
z.liu@uni-kassel.de

M. Sc. Zonglin Liu ist wissenschaftlicher Mitarbeiter im Fachgebiet für Regelungs- und Systemtheorie am Fachbereich Elektrotechnik / Informatik an der Universität Kassel. Im Rahmen seiner Forschungstätigkeiten untersucht er Verfahren zur optimalen und prädiktiven Regelung von cyberphysischen Systemen mit Unsicherheiten.



Olaf Stursberg

Regelungs- und Systemtheorie, FB Elektrotechnik/Informatik, Universität Kassel, D-34109 Kassel, Germany
stursberg@uni-kassel.de

Prof. Dr.-Ing. Olaf Stursberg ist Professor und Fachgebietsleiter für Regelungs- und Systemtheorie im Fachbereich Elektrotechnik / Informatik an der Universität Kassel. Seine Hauptarbeitsgebiete schließen Methoden zur Regelung und Analyse hybrider Systeme, zur optimalen und prädiktiven Regelung vernetzter und hierarchischer Systeme sowie Verfahren zur Regelung stochastischer und unsicherer Systeme ein.