Nonlinear Hybrid Dynamical Systems: Modeling, Optimal Control, and Applications

Martin Buss[1], Markus Glocker[2], Michael Hardt[2], Oskar von Stryk[2], Roland Bulirsch[3], and Günther Schmidt[4]

[1] Control Systems Group, Technische Universität Berlin, Berlin, Germany
[2] Simulation and Systems Optimization Group, Technische Universität Darmstadt, Darmstadt, Germany
[3] Zentrum Mathematik, Technische Universität München, München, Germany
[4] Institute of Automatic Control Engineering, Technische Universität München, München, Germany

Abstract. Nonlinear hybrid dynamical systems are the main focus of this paper. A modeling framework is proposed, feedback control strategies and numerical solution methods for optimal control problems in this setting are introduced, and their implementation with various illustrative applications are presented. Hybrid dynamical systems are characterized by discrete event and continuous dynamics which have an interconnected structure and can thus represent an extremely wide range of systems of practical interest. Consequently, many modeling and control methods have surfaced for these problems. This work is particularly focused on systems for which the degree of discrete/continuous interconnection is comparatively strong and the continuous portion of the dynamics may be highly nonlinear and of high dimension. The hybrid optimal control problem is defined and two solution techniques for obtaining suboptimal solutions are presented (both based on numerical direct collocation for continuous dynamic optimization): one fixes interior point constraints on a grid, another uses branch-and-bound. These are applied to a robotic multi-arm transport task, an underactuated robot arm, and a benchmark motorized traveling salesman problem.

1 Introduction

The recent interest in nonlinear hybrid dynamical systems has forced the merger of two very different modeling and control methodologies, namely those for discrete and for continuous systems. The investigation of hybrid systems attempts to effectively unite these two formalisms in order to model, investigate, and design these systems with analytical and numerical tools. The attempt to provide a unified hybrid modeling scheme well-suited to the study of hybrid dynamical systems has inspired many researchers [4,14,16,23, 30,35,36,40], including the hybrid modeling approach presented here which is based on previous work in [18]. The characteristic behavior of hybrid systems is discussed and illustrated using this modeling scheme. In particular, the multiple potential dynamical events that may occur due to the strong interconnection of discrete and continuous elements are highlighted.

Theoretical work on controllability properties of nonlinear hybrid dynamical systems is still in its early stages and to date only several problems of low state and control dimension can be thoroughly understood [43]. Nevertheless, there has been a strong interest in numerical methods for determining controllers for these systems, inspired from the success of such approaches in conventional nonlinear optimal control problems. Nonlinear optimal control plays a key role in modern mechatronics and robotics, in particular in the area of path, trajectory, and action planning. To mention some of the many applications: walking pattern and trajectory planning [26], mobile robot path planning [29], optimal payload (weight) lifting, and acrobatics [2,34], etc. Numerical algorithms designed for hybrid optimal control problems (HOCPs) with variable structure, nonlinear differential equations have recently been published [15, 19, 28, 41]. These efforts were applied to low-dimensional illustrative problems, yet the results presented here demonstrate that numerical methods do exist which are promising for dealing with realistic, higher-dimensional system models.

The key to numerically solving HOCPs seems to be the combination of efficient numerical solvers – such as direct collocation – for optimal control problems together with (heuristical) approaches to reduce the combinatorial complexity of the discrete event aspect in HOCPs [19, 47, 48, 44]. This paper presents numerical solution techniques for HOCPs with applications in mechatronics and robotics. An example problem of three robotic arms cooperatively transporting an object from an initial to a goal position is solved suboptimally by fixing interior point times and state constraints to fixed values on a grid. The trajectory planning problem of an underactuated robot with an unactuated joint equipped with a holding brake in the passive joint is solved by branch-and-bound to obtain optimal hybrid trajectories, in particular, the optimal number of switches for the holding brake. Finally the solution for the benchmark motorized traveling salesman problem is presented which is a problem that is easily scalable to higher dimensions.

The solution approaches presented here rely on the efficient numerical tool DIRCOL, which implements a direct collocation method to approximately solve nonlinear optimal control problems by advanced nonlinear programming methods [45], see also [6, 26, 46]. The organization of the paper is as follows: Sect. 2 proposes the *Hybrid State Model* HSM as a general hybrid modeling framework. Hybrid feedback control architectures are introduced in Sect. 3. In Sect. 4 a broad class of HOCPs is defined. In Sect. 4.2 numerical solution strategies to obtain suboptimal solutions on interior point constraints on grids and a branch-and-bound strategy are proposed. The solution of three illustrative hybrid problems in robotics are presented in Sect. 5 followed by a discussion of more realistic, higher dimensional problems currently being investigated.

2 Modeling of Hybrid Dynamical Systems

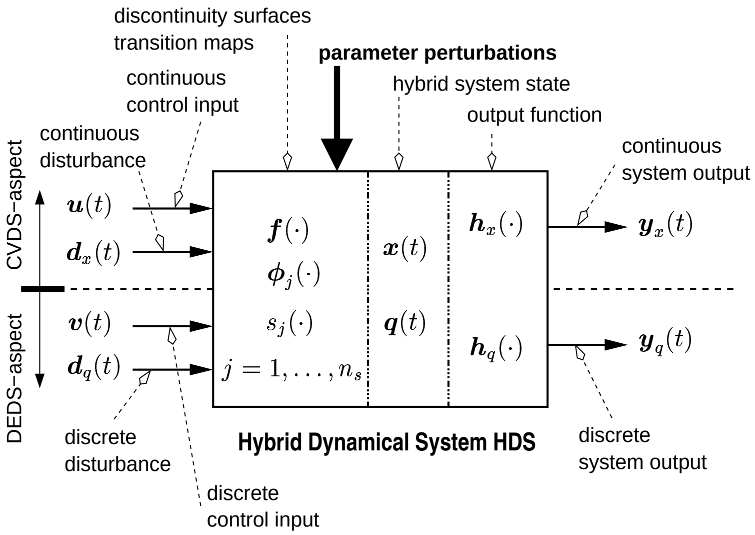

Fig. 1. Hybrid dynamical system (HDS) with continuous variable (CVDS) and discrete-event (DEDS) aspects composed of input, output, and state vectors, discontinuity surfaces and jump maps

A conventional continuous dynamical system is described by the velocity vector field $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, t)$, which depends on the continuous state $\boldsymbol{x}$, the continuous control input $\boldsymbol{u}$, and time t; the continuous output $\boldsymbol{y}_x$ is generated by the output function $\boldsymbol{h}_x(\boldsymbol{x}, \boldsymbol{u}, t)$. The dynamics of a lumped parameter continuous time systems are thus defined by a set of ordinary differential (algebraic) equations. Systems with purely discrete state dynamics are often modeled by a finite state automaton or a Petri-Net. Interconnections of these very different system descriptions are denoted as hybrid dynamical systems and a variety of modeling paradigms have been proposed for which we refer to [14,23,30,35,39]. The hybrid modeling approach presented here is rooted in the theory of continuous dynamical systems and includes discrete system elements such as discontinuous nonlinearities and switching actions as extensions to these systems. This leads to a general hybrid system model for the class of systems denoted as hybrid dynamical systems (HDS).

A HDS consists of, in addition to continuous dynamical system aspects, a *discrete (symbolic) state* $\boldsymbol{q} \in \mathbb{N}^l$, a *discrete (symbolic) control input* $\boldsymbol{v} \in \mathbb{N}^k$, a *discrete (symbolic) system output* $\boldsymbol{y}_q$, *discrete event* generating functions s_j, and *discrete dynamics* $\boldsymbol{\phi}_j$, see Fig. 1. The continuous dynamical behavior is the result of the velocity vector field $\boldsymbol{f}(\cdot)$. Discrete events are caused by the discontinuity indicator functions s_j and hybrid successor states are specified by transition (jump) maps $\boldsymbol{\phi}_j$, $j = 1, \ldots, n_s$. Hence, the hybrid dynamics are specified by the three components $\boldsymbol{f}(\cdot)$, $s_j(\cdot)$, $\boldsymbol{\phi}_j(\cdot)$, see left part of Fig. 1. Inputs to the hybrid dynamical system are the continuous control input $\boldsymbol{u}(t)$, the discrete control input $\boldsymbol{v}(t)$, the continuous disturbance $\boldsymbol{d}_x(t)$, and the discrete disturbance signals $\boldsymbol{d}_q(t)$. The hybrid output $\boldsymbol{y}(t) = [\boldsymbol{y}_x(t)^T \boldsymbol{y}_q(t)^T]^T$ is produced by the output functions $\boldsymbol{h}(\cdot) = (\boldsymbol{h}_x(\cdot),\ \boldsymbol{h}_q(\cdot))$.

2.1 The Hybrid State Model

In this section the *hybrid state model* (HSM) is proposed for the modeling of a fairly general class of nonlinear hybrid dynamical systems. The model is related to the Branicky-Borkar-Mitter BBM model, see [7,8,9,10,11,12,13,14]. The main difference lies in the use of discontinuity surfaces defined by switching functions instead of jump sets used in the BBM model. A benefit of the HSM model is that switching functions have close ties to variable structure control; another advantage is that simulation and implementation of the HSM is straightforward.

Definition 1 (HSM). A hybrid dynamical system (HDS) is defined by its hybrid state model (HSM) as follows:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{q}, t) \qquad \text{if} \quad s_j(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{q}, \boldsymbol{v}, t) \neq 0, \quad j = 1, \ldots, n_s \tag{1}$$

$$\begin{bmatrix} \boldsymbol{x}(t^+) \\ \boldsymbol{q}(t^+) \end{bmatrix} = \boldsymbol{\phi}_j(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{q}, \boldsymbol{v}, t^-) \quad \text{if} \quad s_j(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{q}, \boldsymbol{v}, t) = 0, \quad j \in \{1, \ldots, n_s\} \tag{2}$$

$$\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{q}, \boldsymbol{v}, t)\ , \tag{3}$$

where (1), (2) describe the continuous and discrete dynamic behavior, respectively; the notation $\boldsymbol{x}(t^+)$ denotes the successor state (limit from the right) of $\boldsymbol{x}$ at time t. The hybrid output $\boldsymbol{y}$ is generated by (3). The continuous state vector $\boldsymbol{x}(t) \in \boldsymbol{\mathcal{X}} \subseteq \mathbb{R}^n$ and the discrete state vector $\boldsymbol{q}(t) \in \boldsymbol{\mathcal{Q}} \subseteq \mathbb{N}^l$ together form the hybrid state vector

$$\boldsymbol{\zeta}(t) = \begin{bmatrix} \boldsymbol{x}(t) \\ \boldsymbol{q}(t) \end{bmatrix} \in \boldsymbol{\mathcal{X}} \times \boldsymbol{\mathcal{Q}} \subseteq \mathbb{R}^n \times \mathbb{N}^l\ .$$

The continuous control input $\boldsymbol{u}(t) \in \boldsymbol{\mathcal{U}} \subseteq \mathbb{R}^m$ belongs to the set $\boldsymbol{\mathcal{U}}$ of permissible controls. The discrete (symbolic) control input vector is $\boldsymbol{v}(t) \in \boldsymbol{\mathcal{V}} \subseteq \mathbb{N}^k$. The hybrid output vector

$$\boldsymbol{y}(t) = \begin{bmatrix} \boldsymbol{y}_x \\ \boldsymbol{y}_q \end{bmatrix} \in \boldsymbol{\mathcal{Y}} \subseteq \mathbb{R}^p \times \mathbb{N}^r$$

combines a p-dimensional continuous output $\boldsymbol{y}_x$ and a r-dimensional discrete (symbolic) output $\boldsymbol{y}_q$; $\boldsymbol{y}$ is generated by the hybrid output function

$$\boldsymbol{h} : \boldsymbol{\mathcal{X}} \times \boldsymbol{\mathcal{U}} \times \boldsymbol{\mathcal{Q}} \times \boldsymbol{\mathcal{V}} \times \mathbb{R} \to \mathbb{R}^p \times \mathbb{N}^r\ . \tag{4}$$

The continuous behavior of the HDS is given by the vector field

$$\boldsymbol{f} : \boldsymbol{\mathcal{X}} \times \boldsymbol{\mathcal{U}} \times \boldsymbol{\mathcal{Q}} \times \mathbb{R} \to \mathbb{R}^n \tag{5}$$

Discontinuous behavior of the HDS is caused by events occurring when the hybrid state intersects discontinuity surfaces

$$s_j : \boldsymbol{\mathcal{X}} \times \boldsymbol{\mathcal{U}} \times \boldsymbol{\mathcal{Q}} \times \boldsymbol{\mathcal{V}} \times \mathbb{R} \to \mathbb{R} \ , \tag{6}$$

for $j = 1, \ldots, n_s$. Note, that the discontinuity surfaces may depend on the continuous and/or the discrete control input $\boldsymbol{u}(t)$, $\boldsymbol{v}(t)$. The hybrid successor state

$$\boldsymbol{\zeta}(t_1^+) = \begin{bmatrix} \boldsymbol{x}(t_1^+) \\ \boldsymbol{q}(t_1^+) \end{bmatrix} \ , \tag{7}$$

after discrete events is given by the transition (jump) maps

$$\boldsymbol{\phi}_j : \boldsymbol{\mathcal{X}} \times \boldsymbol{\mathcal{U}} \times \boldsymbol{\mathcal{Q}} \times \boldsymbol{\mathcal{V}} \times \mathbb{R} \to \boldsymbol{\mathcal{X}} \times \boldsymbol{\mathcal{Q}} \ \ , \tag{8}$$

see also (2). As long as all discontinuity surface functions $s_j(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{q}, \boldsymbol{v}, t) \neq 0$, for $j = 1, \ldots, n_s$, the system trajectory evolves continuously according to (1).

Remark 1. A sliding-mode condition [37] also fits into the model from Definition 1 when it is permitted that infinitely many discrete transitions occur in a finite time period. Results describing such cases may be found in [37,21,22].

Remark 2. It has been shown that the BBM model incorporates alternative modeling formalisms such as the Tavernini TAV model [40], the Back-Guckenheimer-Myers BGM model [4], the Nerode-Kohn NK model [36] and the Brockett BRO model [16]. This applies here as well to the proposed HSM defined in Definition 1, which also includes further modeling paradigms such as [35], see [18] for a detailed discussion.

2.2 Characterization of Hybrid Dynamic Behavior

The dynamic behavior of a HDS is strongly influenced by discontinuities in its system trajectories. Discontinuities include *state resets* (SR) resulting in *state jumps*, *vector field switches* (VFS) resulting in a switch of the velocity vector field, and their combination (SRVFS). These may be triggered by a *time event* (TE) occurring at a certain time or by a *state event* (SE) if the system state reaches a certain value. Further events include *control events* (CE) caused by the introduction of a hybrid control action into the discrete control input or *disturbance events* (DE) caused by discrete disturbance inputs. These events may be interdependent as, for example, a SE may either be induced externally (*controlled*) as a result of a CE or DE or induced internally (*autonomous*) [43]. Other dynamic effects of HDS include chaotic behavior, see e.g. [17, 24], or sliding mode, see e.g. [37, 42]. Further discussion of hybrid dynamic characteristics may be found in [18].

In Fig. 2 an example of a typical path for a hybrid trajectory is plotted. The HDS starts with the discrete state $\boldsymbol{q} = \boldsymbol{q}_1$ and continuous state $\boldsymbol{x}(0) \in \boldsymbol{\mathcal{X}}_1 \subseteq \mathbb{R}^n$ and evolves within the portion of state space open from the left on the left-hand side of Fig. 2. As soon as the discontinuity surface $s_1 = 0$ is reached, the hybrid state is reinitialized with a state reset (SR) after which the system trajectory continues in the discrete state $\boldsymbol{q} = \boldsymbol{q}_2$ corresponding to the continuous portion of state space $\boldsymbol{\mathcal{X}}_2 \subseteq \mathbb{R}^n$. The trajectory then enters into a CE region when the discontinuity surface s_2 is crossed. The CE in this case must first be triggered by a discrete control input $\boldsymbol{v} = \boldsymbol{v}_2$ which occurs upon reaching approximately the center of the CE region. The resulting SR causes the system to make the transition into the discrete state $\boldsymbol{q} = \boldsymbol{q}_3$ and its respective portion of state space $\boldsymbol{\mathcal{X}}_3 \subseteq \mathbb{R}^n$. There a TE occurs in combination with a SR whereby the discrete state does not change after the TE. The portions of state space $\boldsymbol{\mathcal{X}}_3$, $\boldsymbol{\mathcal{X}}_4 \subseteq \mathbb{R}^n$ corresponding to the discrete states $\boldsymbol{q}_3$, $\boldsymbol{q}_4$ are separated by a discontinuity surface s_3 from one another. The system trajectory reaches this discontinuity surface and enters with its fulfillment of the necessary sliding-mode conditions into a sliding state along the discontinuity surface $s_3 = 0$. Finally the existence conditions for the sliding-mode are no longer fulfilled resulting in the system evolution in the discrete state $\boldsymbol{q} = \boldsymbol{q}_4$ in the state space region $\boldsymbol{\mathcal{X}}_4$ until the SE $s_4 = 0$.

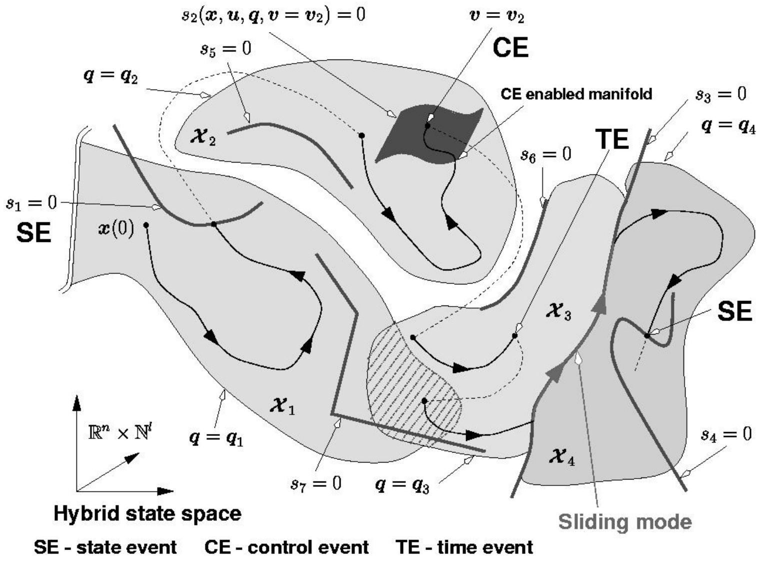
$s_2(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{q}, \boldsymbol{v} = \boldsymbol{v}_2) = 0$
$\boldsymbol{v} = \boldsymbol{v}_2$
CE
$s_5 = 0$
$\boldsymbol{q} = \boldsymbol{q}_2$
CE enabled manifold
$s_3 = 0$
$\boldsymbol{q} = \boldsymbol{q}_4$
$\boldsymbol{\mathcal{X}}_2$
TE
$s_6 = 0$
$s_1 = 0$
SE
$\boldsymbol{x}(0)$
$\boldsymbol{\mathcal{X}}_3$
SE
$\mathbb{R}^n \times \mathbb{N}^l$
$\boldsymbol{q} = \boldsymbol{q}_1$
$\boldsymbol{\mathcal{X}}_1$
$s_4 = 0$
$\boldsymbol{\mathcal{X}}_4$
$s_7 = 0$
$\boldsymbol{q} = \boldsymbol{q}_3$
Hybrid state space
Sliding mode
SE - state event
CE - control event
TE - time event

Fig. 2. An example of the evolution of a typical hybrid system trajectory in a hybrid state space

In Fig. 2 further examples are displayed of discontinuity surfaces s_5, s_6, s_7 that are irrelevant for the example trajectory. Furthermore it is shown how state space regions corresponding to certain discrete states, e.g. $\boldsymbol{\mathcal{X}}_3$ and $\boldsymbol{\mathcal{X}}_4$, can overlap. The allowable region $\boldsymbol{\mathcal{X}}_1$ corresponding to the discrete state $\boldsymbol{q}_1$ continues unbounded into infinity in Fig. 2. The portrayal of the hybrid state space in Fig. 2 is planar, usually it will be of much higher dimension.

3 Hybrid Feedback Control

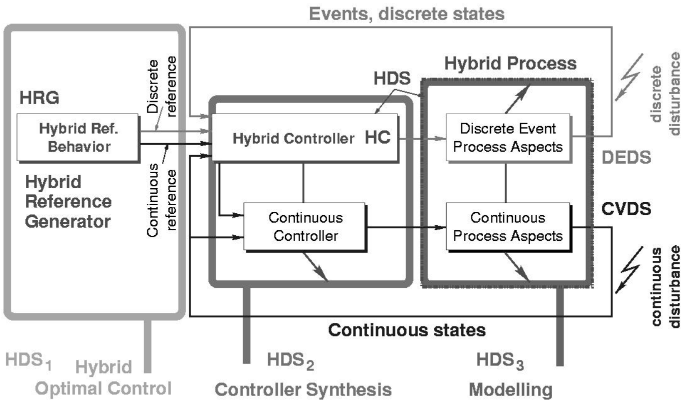

Fig. 3. General hybrid control architecture

In this investigation of hybrid dynamical systems, a general hybrid control architecture is proposed consisting of three main parts, see Fig. 3: (i) the hybrid process model, cf. Sect. 2; (ii) the hybrid controller (HC) controlling this process to be discussed in this section; and (iii) the hybrid reference trajectory generator (HRG). The synthesis of reference trajectories implemented in the HRG as solutions to hybrid optimal control problems will be discussed in Sect. 4.

Hybrid Control and Error Compensation Taking the HSM of Definition 1 as the basis for modeling a HDS and keeping in mind the control architecture described above, it is possible to generalize classical control concepts such as output-following control to the hybrid case. The resulting hybrid output control (HOC) block diagram with hybrid control signals is depicted in Fig. 4. The hybrid output controller compares in Fig. 4 hybrid reference values with actual output values and produces hybrid control signals such that the output tracks the reference value with small error. Calculating the error between discrete reference value and the actual discrete output is an important question which has received little attention. A discussion can be found in [18]. An obvious way, for example, to define the discrete comparison operator $\uplus$ would be to perform the arithmetic difference of two discrete values resulting in an integer-valued discrete error.

In principle, the goal of a hybrid controller is to eventually make the hybrid control error small. In case of a discrete error, this may not be easy as the hybrid process may be in contact with a moving system other than that assumed by the hybrid controller. One solution to hybrid error compensation is shown in Fig. 4, where a discrete error activates a continuous prefilter to modify the continuous reference $\boldsymbol{y}_x^s \rightarrow \boldsymbol{y}_x^f$ in such a way that both the discrete

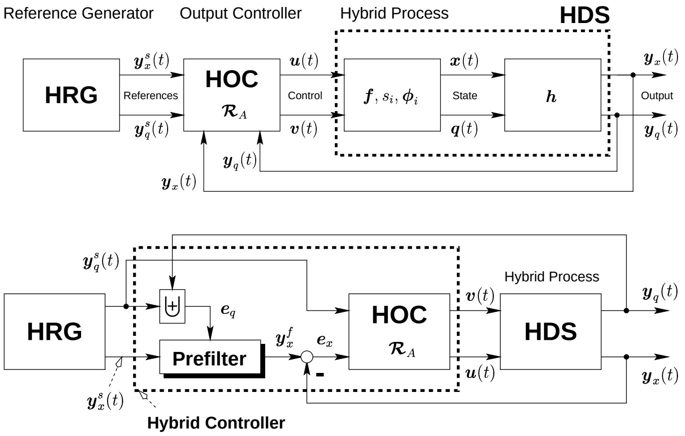

Fig. 4. Hybrid output control (top) and hybrid error compensation by means of a continuous prefilter (bottom)

as well as the continuous control error eventually vanish. Similar concepts are a discrete prefilter, more complicated discrete dynamics in the compensation controller, or a combined reference generator adaptation scheme, see [18] for details.

4 Hybrid Optimal Control

The discrete-continuous process model of a hybrid optimal control problem (HOCP) consists of a set of ordinary differential or differential-algebraic equations of variable structure and variable constraint equations. The system structure varies among a (finite) discrete set of system descriptions each of which is associated with a specific discrete state of the considered hybrid system. The challenging aspect of this model is that the value of the discrete variable can determine the sequence, type and number of phase dynamics. Thus, the dynamics in a phase and even the dimension or number of constraints may be completely different for different values of the discrete variable.

4.1 Hybrid Optimal Control Problem

The HOCP is to find optimal hybrid (i.e., continuous $\boldsymbol{u}$ and discrete $\boldsymbol{v}$) control trajectories such that an integral cost index, typically an integral of a function of the hybrid system state and control input, is minimized subject to the system dynamics, initial, terminal, and further equality or inequality constraints.

Definition 2. The HOCP is defined as the minimization of the real valued, hybrid cost index J

$$\min_{\boldsymbol{u},\,\boldsymbol{v}} J(\boldsymbol{u},\boldsymbol{v}) = \Theta + \int_{t_0}^{t_f} \psi(\boldsymbol{x},\boldsymbol{u},\boldsymbol{q},\boldsymbol{v},t)\ \mathrm{d}t\ , \tag{9}$$

subject to

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u},\boldsymbol{q},\boldsymbol{v},t) \qquad \text{if } s_j(\boldsymbol{x},\boldsymbol{u},\boldsymbol{q},\boldsymbol{v},t) \neq 0 \quad j = 1,\ldots,n_s \tag{10}$$

$$\begin{bmatrix} \boldsymbol{x}(t_i^+) \\ \boldsymbol{q}(t_i^+) \end{bmatrix} = \boldsymbol{\phi}_j(\boldsymbol{x},\boldsymbol{u},\boldsymbol{q},\boldsymbol{v},t_i^-) \ \ \text{if } s_j(\boldsymbol{x},\boldsymbol{u},\boldsymbol{q},\boldsymbol{v},t_i^-) = 0 \quad j \in \{1,\ldots,n_s\} \tag{11}$$

$$\boldsymbol{u}(t) \in \boldsymbol{\mathcal{U}} \subset \mathbb{R}^{n_u},\ \boldsymbol{v}(t) \in \boldsymbol{\mathcal{V}} \subset \mathbb{Z}^{n_v},$$

$$\boldsymbol{x}(t) \in \boldsymbol{\mathcal{X}} \subset \mathbb{R}^{n_x},\ \boldsymbol{q}(t) \in \boldsymbol{\mathcal{Q}} \subset \mathbb{Z}^{n_q}, \quad \forall t \in [t_0,t_f] \tag{12}$$

$$0 \leq \boldsymbol{g}(\boldsymbol{x},\boldsymbol{u},\boldsymbol{q},\boldsymbol{v},t), t \in [t_0,t_f] \text{ inequality constraints,} \tag{13}$$

$$\boldsymbol{x}(t_0) = \boldsymbol{x}_0,\ \boldsymbol{q}(t_0) = \boldsymbol{q}_0 \text{ initial conditions,} \tag{14}$$

$$\boldsymbol{x}(t_f) = \boldsymbol{x}_f,\ \boldsymbol{q}(t_f) = \boldsymbol{q}_f \ \text{ terminal conditions,} \tag{15}$$

where the initial and final times, written as t_0, t_f, are free or fixed, s_j are the n_s switching functions and ϕ_j denotes the explicit phase transition conditions (jump maps) occurring at the zeros of one of the switching functions. The Mayer type part Θ of the performance index is a general function of the phase transition times (events) t_i, $i = 0,\ldots,N$ and of the continuous $\boldsymbol{x}(t_i^-)$, $\boldsymbol{x}(t_i^+)$ and discrete states $\boldsymbol{q}(t_i^-)$, $\boldsymbol{q}(t_i^+)$ just before and just after the $N-1$ interior transition events and at the beginning and final times respectively written as

$$\Theta := \Theta[\ \boldsymbol{x}(t_0^+),\ldots,\boldsymbol{x}(t_N^-); \\ \boldsymbol{q}(t_0^+),\ldots,\boldsymbol{q}(t_N^-); t_0,\ldots,t_N]\ \in \mathbb{R}\,.$$

Here, $t_f = t_N$ is assumed while the number of phases N may be given or free. The integrand ψ is a real-valued function of the continuous/discrete state and control variables and of time.

The minimization of (9) is subject to the initial and terminal conditions (14), (15), admissible values for the continuous/discrete control variables (12), and inequality constraints (13). Obviously, valid hybrid optimal trajectories must obey the differential equations (10) and the discrete-based phase transition equations (11). The optimization parameters to be determined are the continuous $\boldsymbol{u}(t)$ and discrete control input trajectories $\boldsymbol{v}(t)$ and all, some, or none of the phase transition times.

The solutions to the HOCPs described in Definition 2 are deterministic *open-loop* trajectories. Like in conventional optimal control this problem class can be generalized to a stochastic setting or to treat issues like optimal closed-loop feedback control. The numerical solution of closed-loop hybrid feedback control problems, however, is at even a much earlier stage and the primarily finite-element based solution strategies that have been presented for their solution [15, 28, 41] cannot readily handle nonlinear systems of more than three dimensions due to the well-known curse of dimensionality [26].

A framework for modeling and (optimally) controlling mixed logical dynamical systems described by linear dynamic equations subject to linear inequalities involving real and integer variables has been proposed by [5]. The on-line optimization problems resulting from a predictive control scheme are solved numerically by application of a mixed-integer quadratic programming branch-and-bound method. However, the approach is not applicable to our class of HOCPs with *nonlinear* dynamics equations subject to *nonlinear* constraints.

4.2 Numerical Solution Strategies

A set of several different numerical strategies is presented here for the approximation of the solution to the HOCP. The basis for the suboptimal solution strategies is the highly efficient direct collocation method implemented in the software package Dircol [45] to approximately solve optimal control problems using solutions to (sparse) nonlinear programs. Dircol was primarily designed for the solution of optimal control problems related to piecewise continuous, nonlinear dynamical systems though it handles well important discrete system components such as unknown interior time events (TE) when state resets (SR) or vector field switches (VFS) may occur. Other discrete state aspects it cannot handle directly such as the number of interior SR or VFS events. These aspects must be specified in advance. For this reason, the proposed solution strategy is to use Dircol in the inner optimization iteration and other strategies to solve for the combinatorial aspect of the discrete-event in an outer level optimization. The key to cope with the possibly overwhelming combinatorial complexity of HOCPs is to reduce the number of candidates to be evaluated in the outer iteration.

After providing some insights into the method Dircol, two alternatives HOCP solution strategies will be shown: (i) suboptimal solution with interior event time and state constraints fixed on a grid combined with graph search, and (ii) transformation to a mixed-binary-optimal control problem and its subsequent solution using a branch-and-bound algorithm.

Sparse Direct Collocation

The numerical method of sparse direct collocation implemented in Dircol can efficiently solve multi-phase optimal control problems with a fixed discrete state trajectory. The state $\boldsymbol{x}$ is approximated by cubic Hermite polynomials $\tilde{\boldsymbol{x}}(t) = \sum_j \alpha_j \hat{\boldsymbol{x}}_j(t)$ and the control vector $\boldsymbol{u}$ by piecewise linear functions $\tilde{\boldsymbol{u}}(t) = \sum_k \alpha_k \hat{\boldsymbol{x}}_k(t)$ on a discretization grid $t_i^c = t_1^{(i)} < t_2^{(i)} < \ldots < t_{n_t^{(i)}}^{(i)} = t_{i+1}^c$ in each phase. The state differential equations (10) are pointwise fulfilled at the grid points and grid midpoints, resulting in a set of nonlinear NLP equality constraints $\boldsymbol{a(y)} = 0$. The control or state inequality constraints are to be satisfied at the grid points resulting in a set of nonlinear NLP inequality constraints $\boldsymbol{b(y)} \geq 0$. The vector $\boldsymbol{y}$ contains the n_y parameters $\boldsymbol{y} = (\alpha_1, \alpha_2, \ldots, \beta_1, \beta_2, \ldots, p, t_1, \ldots, t_{N-1}, t_f)^T$ where $p_i \in [0,1]$, $i = 1, \ldots, n_p$ denotes the set of relaxed binary variables. With ϕ as the parameterized cost index (18), the nonlinearly constrained optimization problem may be written as the nonlinear program (NLP)

$$\min_{\boldsymbol{y}} \phi(\boldsymbol{y}) \quad \text{subject to} \quad \boldsymbol{a(y)} = 0,\ \boldsymbol{b(y)} \geq 0\,. \tag{16}$$

The transcription of the optimal control problem to an NLP is made by DIRCOL [45], the NLP is solved efficiently with the advanced SQP-based sparse nonlinear program solver SNOPT [25], and subsequently DIRCOL processes the solution to provide state and control trajectories, error estimates and output that may be used to verify the optimality of the solution.

Important features of the method are:

- As the grid becomes finer, the discretized solution converges to a solution of the Euler-Lagrange differential equations (EL-DEQs) according to the Maximum Principle.
- Reliable estimates of the adjoint variable trajectories $\tilde{\lambda}$ along the discretization grid may be derived from the Lagrange multipliers of the NLP. They enable a verification of the optimality conditions of the discretized solution without solving explicitly the EL-DEQs.
- Local optimality error estimates can be derived which enable efficient strategies for successively refining a first solution on a coarse grid.
- The NLP Jacobians ($\boldsymbol{\nabla a(y), \nabla b(y)}$) are sparse and structured, permitting the use of sparse solvers.
- Computation is fast because ODE simulation and control optimization are performed simultaneously (unlike shooting methods).
- In extension of (10), the method is also applicable to systems described by differential-algebraic equations of differential index 1. In this case, the algebraic state variables are discretized analogously to the control variables by piecewise linear functions.

Suboptimal Solution Technique

Suboptimal solutions may be obtained by fixing interior point times and states to fixed values on a (fine) grid. Between all these grid points standard optimal control problems with fixed boundary conditions are solved. Finally, the suboptimal solution to the HOCP is obtained by a graph search with each grid point forming nodes and the optimal cost weighing the vertices of this graph. This solution strategy is applied to solve the cooperative multi-arm transport problem in Sect. 5.1, see also [19, 18, 20]. Disadvantages of this approach are the possibly high number of multi-point boundary value problems to be solved and the inherent suboptimality of the obtained solution. On the other hand, an appealing advantage is that by problem understanding one often has good insight as to how the grids need to be specified, and that useful solutions usually can be obtained easily.

Branch-and-Bound

The solution method for mixed-binary optimal control problems (MBOCP) using a combination of sparse direct collocation and branch-and-bound was first presented in [44] and further investigated in [19, 47, 48]. Given certain assumptions, the HOCP may be transformed into a MBOCP with a simple transformation of its discrete variables. For this we assume:

(A1) The number $N-1 \geq 0$ of event times t_i and, thus, the number N of phases are finite and known (this assumption may be circumvented with yet another "outer" iteration to vary N).

(A2) The discrete state variable $\boldsymbol{q}$ and the discrete control variable $\boldsymbol{v}$ are constant in each phase and may only change at an event t_i.

Each discrete variable $q_k(t)$ (or $v_l(t)$), $0 \leq t \leq t_f$, is described by an integer variable $\boldsymbol{z}_k \in \mathbb{Z}^{n_c+1}$ with $q_k(t) = z_{k,i}$ in the i-th phase. A scalar, integer variable z_1 with given lower and upper bounds $z_1 \in [z_{1,\min}, z_{1,\max}] \subset \mathbb{Z}$ can be transformed into a binary variable $\boldsymbol{\omega} \in \{0,1\}^{n_\omega}$ of dimension n_{z_1} by

$$z_1 = z_{1,\min} + \omega_1 + 2^1\omega_2 + \ldots + 2^{n_\omega - 1}\omega_{n_\omega}, \tag{17}$$

with $n_\omega = 1 + \mathrm{INT}\,\{\log\,(z_{1,\max} - z_{1,\min})/\log 2\}$. In this manner, a binary control vector $\boldsymbol{\omega}$ may be used to represent both the unknown discrete state $\boldsymbol{q}$ in each phase and the discrete control variable $\boldsymbol{v}$ which controls the order and types of phase transitions.

The MBOCP is to minimize the real-valued, hybrid performance index

$$J[\boldsymbol{u},\boldsymbol{\omega}] = \Theta + \sum_{i=1}^{N} \int_{t_{i-1}}^{t_i} \psi(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\omega}, t)\,\mathrm{d}\,t \tag{18}$$

subject to (10)-(15) with the discrete variables $\boldsymbol{q}$ and $\boldsymbol{v}$ substituted by the binary control vector $\boldsymbol{\omega} \in \{0,1\}^{n_\omega}$ in both Θ and ψ. The solutions of the MBOCP are the optimal (open loop) trajectories of $\boldsymbol{x}^*(t)$, $\boldsymbol{u}^*(t)$, $0 \leq t \leq t_f$, the optimal phase transition times $t_i^{c\,*}$, the possibly free final time t_f^* , and the optimal binary control vector $\boldsymbol{\omega}^*$.

Remark 3. The nature of the binary control vector $\boldsymbol{\omega}$ appearing in the MBOCP is twofold. On the one hand it represents the discrete control variable $\boldsymbol{v}$ that controls the order and types of phase transitions, on the other hand it also represents the discrete state $\boldsymbol{q}$ in each phase.

To avoid solving all $\{0,1\}^{n_w}$ MBOCPs, a branch-and-bound strategy in combination with a binary search tree is employed: The subproblems solved by Dircol provide approximate upper and lower bounds to the MBOCP performance index. If the lower bound at a node is greater than the global upper bound, that branch is discarded. The comparison of subproblem solutions is additionally aided by the use of the optimality error estimate (confidence interval) computed by Dircol [45]. A subproblem is constructed by either fixing a component of the binary control vector $\boldsymbol{\omega_i}$ to 0 or 1 or relaxing it $0 \leq \boldsymbol{\omega}_i \leq 1$, $i \in \{1, 2, \ldots, n_\omega\}$. The MBOCP is thus reduced to a "continuous" multi-phase optimal control problem.

Remark 4. The B&B procedure on the binary control vector requires existence of solutions to relaxed MBOCPs, or more precisely, the existence of continuous relaxations to the MBOCP. For some MBOCPs, numerical solutions may not exist for their relaxations. When they exist, the relaxed binary variables may not necessarily have any physical meaning with respect to the underlying application. This however does not present any numerical difficulties. The solution of subproblems in the B&B is analagous to the application of the interior-point solution method to linear programming problems. The iterative procedure normally first delivers a well-defined solution at termination of the algorithm. Usually additional modeling effort will be required in defining suitable "meta"-MBOCPs allowing useful relaxations analogously to the definition of superstructures for mixed-integer nonlinear programming problems [1].

Remark 5. As it must be expected that some modeling effort for the MBOCP is required before applying numerical methods, it has been suggested to derive suitably simplified and problem specific "screening models" [3]. A screening model can be solved to simultaneously guarantee global optimality and to yield a rigorous lower bound on the solution of the MBOCP, thus avoiding the need for dealing with relaxed MBOCPs. An application for a simple batch process development has successfully been investigated in [3]. Although in principle the idea seems to be applicable to a wide class of problems, there is no constructive way to obtain a screening model for a concrete MBOCP.

Remark 6. The challenge in solving relaxed MBOCPs during the binary tree search cannot be underestimated. There is no numerical method available that solves optimal control problems with nonlinear dynamics defined in multiple phases and subject to nonlinear constraints and with phase transitions at unknown times *guaranteeing* the global optimum or that even guarantees a locally optimal solution in general at all. However, not only the global optimum is of interest. For many types of MBOCPs, even a "good" solution obtained by the proposed approach that significantly improves the initial guess will be highly appreciated.

The branch-and-bound procedure is outlined as follows:

1. Find a global upper bound. Make an initial guess for $\boldsymbol{\omega}$ and solve the resulting control problem with $\boldsymbol{\omega}$ fixed;

2. At the root node, relax all binary variables ($0 \leq \boldsymbol{\omega}_i \leq 1, i \in \{1, 2, \dots, n_\omega\}$) and solve to obtain a lower bound to the solution;
3. Select the branching variable $\boldsymbol{\omega}_i$ and solve both subproblems with that component set to 0 and 1 thereby creating two offspring to the current node;
4. Select the next node where to continue the branching process by either: Breadth First Search (node with minimal performance out of those with the least amount of fixed components), Depth First Search (node with minimal performance out of those with the maximum amount of fixed components), Minimum Bound Strategy (node with minimal performance);
5. If the lower bound in a node is greater than the current best upper bound of the whole search tree, then all subsequent branches from this node are trimmed.

Depending on the problem, this approach may get caught in local minima which can be avoided by perturbations for the relaxed problems. It is also hard to guarantee that trimmed branches do not contain the true global minimum. A positive note is that useful suboptimal solutions are readily computable.

5 Applications

5.1 Multi-Arm Transportation Task

Fig. 5 shows a cooperative multi-arm transport task. The square object is initially on the right and is to be transported to the elevated goal position on the left. This is to be accomplished by picking up the object with transport arm 1, handing it over to arm 2, then to arm 3, and finally placing it in the goal position. Each transport arm j has two rotational joints $\theta_{j,i}$ driven by control input torques $u_{j,i}$, $j = 1, 2, 3$, $i = 1, 2$. The effector of each transport arm can be opened/closed to grasp/release the object by a discrete control input v_j. The transportation task should be performed such that the cost index of quadratic power consumption is minimized

$$\min_{u_{j,i}(t), v_j(t)} J = \int_0^{t_f} \sum_{j=1}^{3} \sum_{i=1}^{2} (u_{j,i}\ \dot{\theta}_{j,i})^2 \ \mathrm{d}t \ .$$

To solve this HOCP we need to determine the optimal hybrid control trajectories $u^*_{j,i}(t)$, $v^*_j(t)$, the positions, velocities and times of object handover. The physical parameters of the multi-arm system are assumed as: mass $m_1 = m_2 = 5$, length $l_1 = l_2 = 1$ of link 1, 2, respectively, object mass $m_o = 10$, ground distance from arm mount point $x_g = 1.5$. The distance between two arms is $d = 1.5$, the grid points for possible handovers of arm 1 are at $y_{1,ho} = -0.75$, $x_{1,ho} = 1.5/x_{1,ho} = 1$ (ground/air), and likewise for the other arms.

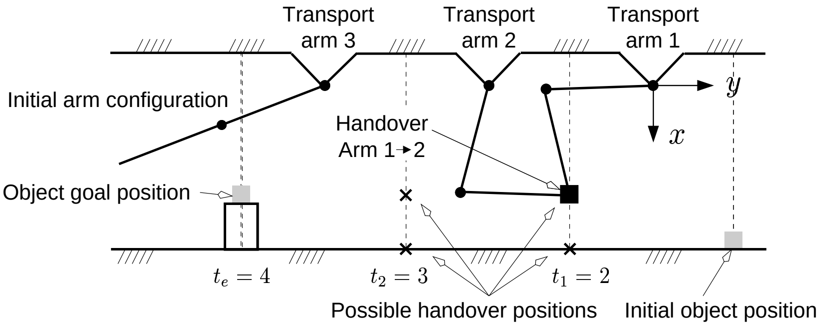

Fig. 5. Cooperative multi-arm transport task

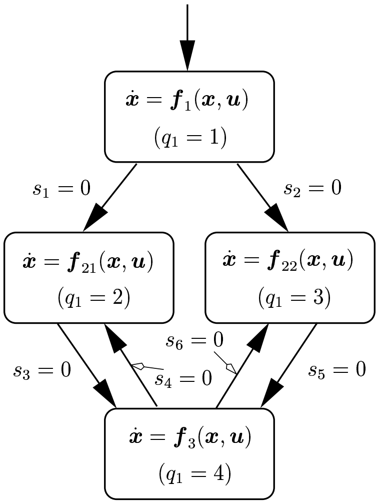
$\dot{\boldsymbol{x}} = \boldsymbol{f}_1(\boldsymbol{x}, \boldsymbol{u})$
$(q_1 = 1)$
$s_1 = 0$
$s_2 = 0$
$\dot{\boldsymbol{x}} = \boldsymbol{f}_{21}(\boldsymbol{x}, \boldsymbol{u})$
$(q_1 = 2)$
$\dot{\boldsymbol{x}} = \boldsymbol{f}_{22}(\boldsymbol{x}, \boldsymbol{u})$
$(q_1 = 3)$
$s_6 = 0$
$s_3 = 0$
$s_4 = 0$
$s_5 = 0$
$\dot{\boldsymbol{x}} = \boldsymbol{f}_3(\boldsymbol{x}, \boldsymbol{u})$
$(q_1 = 4)$

Fig. 6. Hybrid model for a single arm

For each arm $i = 1, 2, 3$ the hybrid model has 4 discrete states $q_i = 1, 2, 3, 4$ as follows: $q_i = 1$: arm has no contact with environment, effector open; $q_i = 2$: arm holds object in configuration 1 (elbow right) object has contact to ground; $q_i = 3$: arm holds object in configuration 2 (elbow left) object has contact to ground; $q_i = 4$: arm holds object in the air, no contact with environment. The variable structure q_i dependent motion differential equation for arm i then are:

$$\dot{\boldsymbol{x}}_i = \boldsymbol{f}(\boldsymbol{x}_i, \boldsymbol{u}_i, q_i) = \begin{cases} \boldsymbol{f}_1(\boldsymbol{x}_i, \boldsymbol{u}_i) & \text{if } q_i = 1 \\ \boldsymbol{f}_{21}(\boldsymbol{x}_i, \boldsymbol{u}_i) & \text{if } q_i = 2 \\ \boldsymbol{f}_{22}(\boldsymbol{x}_i, \boldsymbol{u}_i) & \text{if } q_i = 3 \\ \boldsymbol{f}_3(\boldsymbol{x}_i, \boldsymbol{u}_i) & \text{if } q_i = 4 \end{cases} \tag{19}$$

Note that if $q_i = 2, 3$ the arm is also subject to a kinematic equality constraint as ground contact needs to be maintained. Environment forces must also be

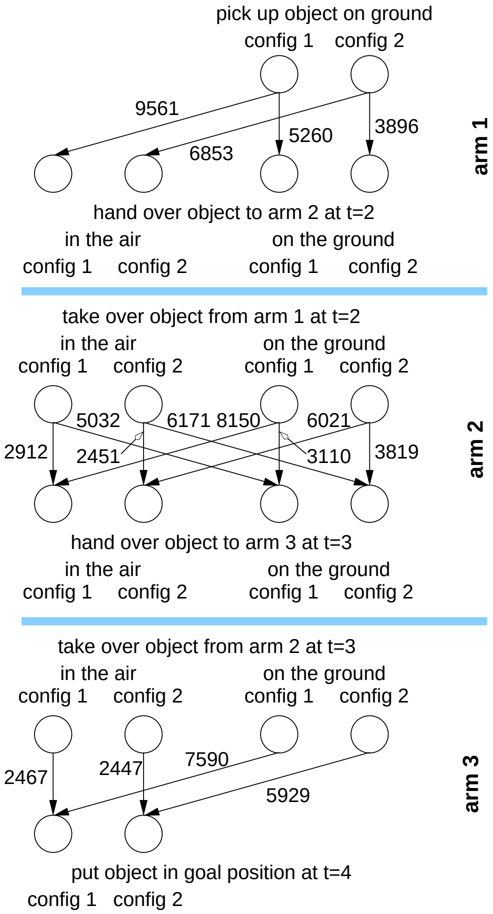

Fig. 7. Feasible handover TPBVPs for each arm

taken into account during such phases. The complete hybrid model of a single arm is shown in Fig. 6.

Applying the suboptimal solution strategy outlined in Sect. 4.2, the coupling of the optimal control problems is first eliminated for each of the transport arms by fixing the possible times and states of handover to constant values on a grid, see Fig. 5. The object handover time from arm 1 to 2 is fixed to $t_1 = 2$ and only two possible handover positions (on the ground and in the air and at zero velocity) are considered. Some of the handover possibilities can be excluded because of internal arm collision problems, e.g. handover in the air between arms 1, 2 with configuration 2, 1, respectively.

All remaining feasible handover TPBVPs (Two Point Boundary Value Problems) and the cost of the optimal solutions obtained by DIRCOL are shown in Fig. 7. The three subgraphs are then combined into the complete

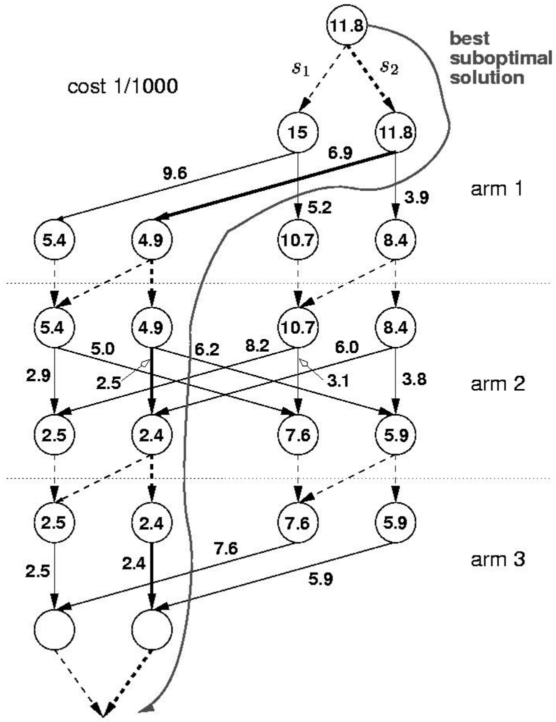

Fig. 8. Graph connecting all feasible discrete sequence candidates

graph in Fig. 8, in which the best suboptimal solution is obtained by minimum path search; also marked in Fig. 8.

The best suboptimal solution to the transport task is to pick up the object by arm 1 and hand it over to arms 2/3 in the air at the fixed positions and times as shown in Fig. 5. Fig. 9 shows some snapshots of the suboptimal coordinated transportation task. [1].

5.2 Underactuated Two Degree-of-Freedom Robot Arm

The trajectory planning example application is considered for a 2-link SCARA robotic arm with two rotational degrees-of-freedom, yet only one actuated (R2D1). In the first joint a torque u_1 may be applied while the second joint may be influenced only by a holding brake controlled by $v_1(t) \in \{0, 1\}$, see Fig. 10 and [33, 32]. The brake can only be set when the second joint has reached a zero relative velocity. A discrete control action can switch back

[1] An animated movie of the suboptimal solution to the multi-arm transportation task is available at `http://www.rs.tu-berlin.de/videos`

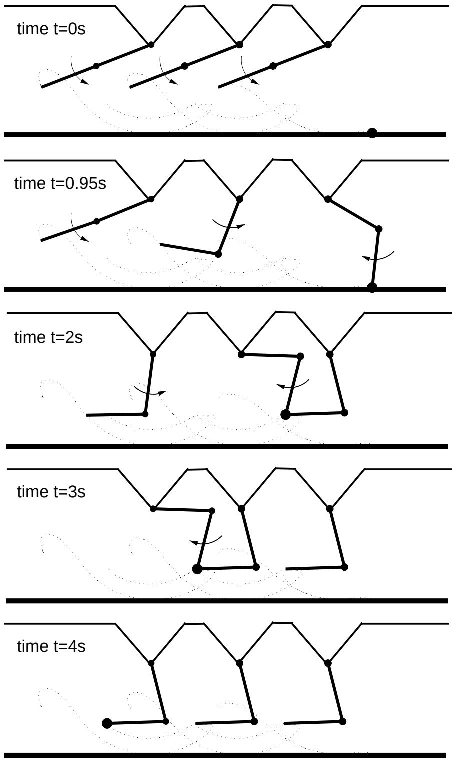

Fig. 9. Snapshot sequence of suboptimal transport solution

and forth between the passive and locked modes for the second joint while a continuous control force is applied to the first joint actuator. We are interested in finding not only the optimal continuous state and control trajectories, but also the optimal discrete strategy composed of the optimal *number* and *times* of the switches necessary to move the R2D1 from a given initial state to a goal state.

The following $\mathcal{H}_2$ performance index is considered

$$J[u_1, v_1] = \int_0^{t_f} (\boldsymbol{x}(t) - \boldsymbol{x}_f)^T \boldsymbol{W} (\boldsymbol{x}(t) - \boldsymbol{x}_f) + \alpha (u_1(t) - u_{1,f})^2 \, \mathrm{d}t \tag{20}$$

where $\boldsymbol{W} \in \mathbb{R}^{4\times 4}$, $\boldsymbol{W} \geq 0$, and $\alpha > 0$. Here, we use $\boldsymbol{W} = \boldsymbol{I}$ and $\alpha = 1$. Furthermore, $\boldsymbol{x}_f \in \mathbb{R}^4$ denotes a desired final state, and $u_{1,f}$ is the control value for which the system is at equilibrium at $\boldsymbol{x}_f$. The final time is constrained, e. g., by $t_f \leq 10\,\mathrm{s}$. The HOCP is to minimize J subject to the robot dynamics

$$\begin{aligned} \ddot{\boldsymbol{\theta}} &= \begin{pmatrix} u_1 \\ 0 \end{pmatrix} - v_1(t)\, \boldsymbol{F}_1(\boldsymbol{\theta}(t), \dot{\boldsymbol{\theta}}(t)) - (1 - v_1(t))\, \boldsymbol{F}_2(\boldsymbol{\theta}(t), \dot{\boldsymbol{\theta}}(t)) \\ \boldsymbol{F}_i(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) &= \boldsymbol{M}_i^{-1}(\boldsymbol{\theta}) \big(\boldsymbol{C}_i(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \boldsymbol{g}_i(\boldsymbol{\theta}) + \boldsymbol{r}_i(\dot{\boldsymbol{\theta}}) \big), \quad i = 1, 2 \end{aligned} \tag{21}$$

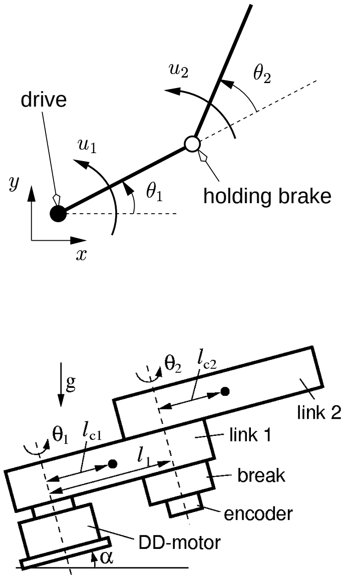

Fig. 10. Kinematic structure of R2D1 [32, 33]

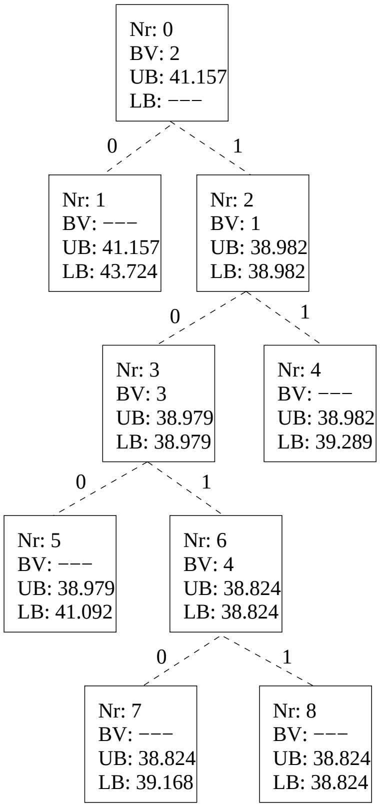

Fig. 11. Branch-and-bound search using *minimum bound* strategy. *Nr* – node number from search order, *BV* – branching variable, *UB* – global upper bound, *LB* – lower bound for branch

$$\begin{array}{llll}
\boldsymbol{x}(t) & = (\theta_1(t), \dot{\theta}_1(t), \theta_2(t), \dot{\theta}_2(t)) & \boldsymbol{u}(t) \in \mathcal{U} & = \mathbb{R} \\
\boldsymbol{x}(0) & = \boldsymbol{x}_0 = (1.2,\ 0,\ 0.8,\ 0)^T & \boldsymbol{x}(t) \in \mathcal{X} & = SO(1) \times SO(1) \times \mathbb{R}^2 \\
\boldsymbol{x}(t_f) & = \boldsymbol{x}_f = (\pi/2,\ 0,\ -\pi/2,\ 0)^T & \boldsymbol{v}(t) \in \mathcal{V} & = \{0, 1\} \\
v_1(t_f) = 1 & \quad \text{(brake on)} & \boldsymbol{q}(t) \in \mathcal{Q} & = \emptyset
\end{array} \tag{22}$$

where $\boldsymbol{M}_i$ are the mass-inertia matrices for each dynamical configuration, $\boldsymbol{C}_i$ are the vectors of Coriolis and centrifugal forces, $\boldsymbol{g}_i$ are the vectors of gravitational forces, and $\boldsymbol{r}_i$ are the friction forces. The physical parameters in standard units are: $l_1 = 0.300$, $l_{c1} = 0.206$, $l_{c2} = 0.092$, $I_1 = 0.430$, $I_2 = 0.127$, $m_1 = 10.2$, $m_2 = 5.75$.

The optimal control problem for R2D1 is formulated as a MBOCP, and the numerical approach discussed in Sect. 4.2 is applied. The time $t_f \leq 10$ is initially divided into a fixed number $m = 8$ of phases, though the intermediate times corresponding to the phase transitions may vary freely. Included in the problem formulation are a set of constant, unknown binary parameters $p_i \in \{0, 1\}$, $i = \{1, \dots, n_p\}$ which are related to the unknown binary variables ω_i. They determine the total number of switches and indicate at which of the predefined phase transitions a switch occurs. The first component p_1 indicates in which discrete state the system starts, $\{p_1 = 0$, brake off; $p_1 = 1$, brake on$\}$. The remaining components of $\boldsymbol{p}$ are a binary representation of the total number of switches taking place during the time interval. For example, if five switches occur beginning with the brake off, then $p = [p_1\ p_2\ p_3\ p_4] = [0\ 1\ 0\ 1]$ and the switches are assigned to the predefined phase transitions using the scheme: $\quad p_k = 1 \ \Rightarrow\ 2^{(n_p-k)}$ switches with one every 2^{k-1} phase transitions beginning with number $2^{(k-2)th} + 1$.

Fig. 12 depicts the phase transitions over which the binary parameter p_k exerts an influence.

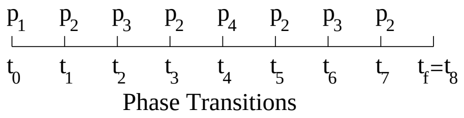

Fig. 12. Phase transitions influenced by binary parameters p_k

The branch-and-bound search strategy was used together with a minimum-bound node selection strategy. Fig. 11 displays the complete binary search path for the problem. An initial solution with $\boldsymbol{p}$ fixed at $[0\ 1\ 0\ 0]$ (4 switches) is first calculated to obtain an upper bound of $J^* = 41.157$. Lower bounds were first calculated for the children of the root node, and the second binary variable is arbitrarily first selected as the branching variable. The final optimal solution has a discrete solution of $\boldsymbol{p}^* = [0\ 1\ 1\ 1]$ corresponding to 7 switches starting with the brake off and an objective value of $J^* = 38.824$. As is normally the case in a branch-and-bound search, the search procedure ends if an integer solution obtained from a relaxed problem is the new best lower bound. In this case, our optimal solution was obtained already at node 2, after the third optimization run. The search though was continued here to verify the solution and ensure that it did not correspond to a local minimum.

In order to avoid convergence to a local minimum, at intermediate steps all relaxed binary parameters in the optimization are initialized to 0.5 to perturb the system away from its starting values and therewith avoid local minima. The final solution[2] as displayed in Fig. 13 has an optimality error of $\tilde{w} = 0.567$ [45]. The incremental difference in the objective decreases rapidly with an increasing number of switches such that the solution with 5 or 6 switches lie within the error margin for the optimal solution with 7 switches. The

[2] An animated movie of the final solution for the R2D1 robot control is available at `http://www.sim.informatik.tu-darmstadt.de/videos`

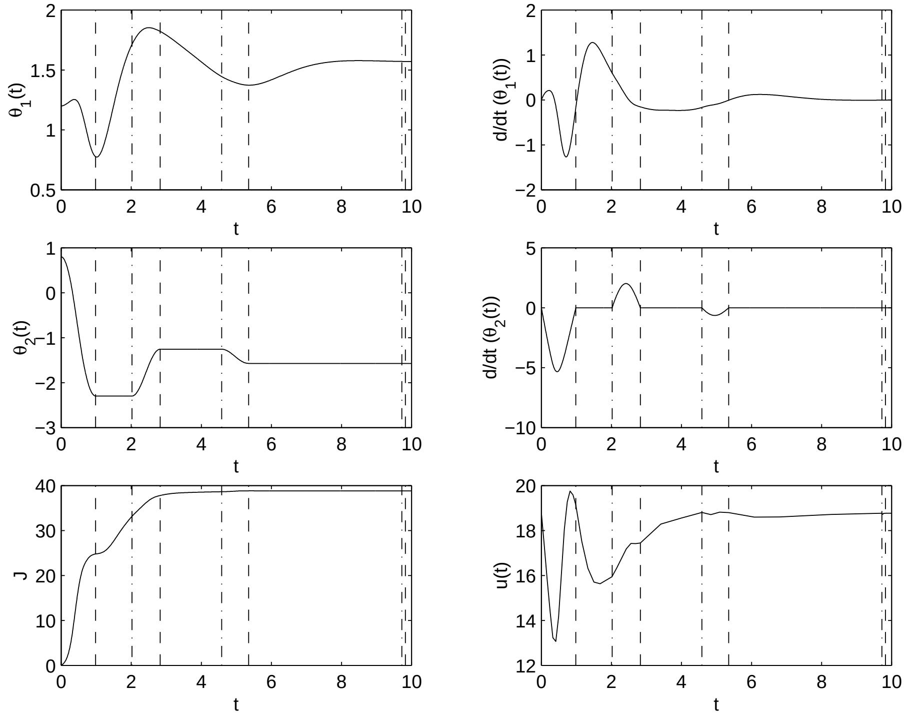

Fig. 13. Final optimal hybrid switching solution with 7 switches

optimality tolerance [25,45] set at 10^{-4} may then be reduced to obtain more accurate solutions in order to correctly distinguish between them. It is also possible at this point to lengthen the search by reinitializing the binary search with more predefined phase transitions thereby allowing for more switches to take place. The average computational time by Dircol for each optimal control problem (the solution at a given node) was 19.6 seconds on a Pentium III 500 MHz computer, the average grid size $\sum_{i=1}^{N} n_t^{(i)}$ was 56.3, and the average NLP dimension was $n_y = 278$, $n_a = 230$.

5.3 The Motorized Traveling Salesman

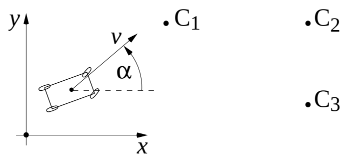

Fig. 14. Motorized traveling salesman problem (MTSP)

We consider the hybrid dynamical extension of one of the most popular combinatorial optimization problems: *A motorized salesman is on his way to visit n_c cities at most one time. He is not allowed to stop in the cities, instead he should drive through them on a smooth curve. He starts at the origin and returns there after his journey. How should he steer and accelerate and in which order should he pass through the cities to minimize the overall traveling time?*

In the standard setting as a combinatorial optimization problem, the interconnections between two cities are independent of each other. In the problem setting here, the salesman has to travel on a smooth curve and the performance in between two cities depends on the overall selection of the continuous (steering wheel, gas and brake pedal) and discrete (order of cities) controls. This benchmark hybrid optimal control problem serves to demonstrate the strong interaction of continuous and discrete dynamics that may occur for even low dimensional systems.

The motorized traveling salesman (MTSP) can be described by a simplified kinematical model describing a point mass moving in a (x, y)-plane

$$\begin{aligned} \dot{x}(t) &= v_x(t), & x(0) &= 0 = x(t_f), \\ \dot{y}(t) &= v_y(t), & y(0) &= 0 = y(t_f), \\ \dot{v}_x(t) &= a_x(t), & v_x(0) &= 0 = v_x(t_f), \\ \dot{v}_y(t) &= a_y(t), & v_y(0) &= 0 = v_y(t_f), \\ a_x^2 + a_y^2 &\le 7 \quad . \end{aligned} \tag{23}$$

Hereby v_x and v_y denote the velocity and a_x, a_y the acceleration or braking of the car in x respectively in y direction, i.e., the continuous state and control variables. The MTSP is formulated as an MBOCP according to Section 4.2 by $\boldsymbol{u} = (a_x, a_y)$, $\boldsymbol{x} = (x, y, v_x, v_y)$ and

$$\min_{\boldsymbol{u},\, \boldsymbol{\omega}} J[\boldsymbol{u}, \boldsymbol{\omega}] := t_f + 0.002 \int_0^{t_f} (u_1^2 + u_2^2) \,\mathrm{dt} \tag{24}$$

$$\boldsymbol{r}_{(i)}(\boldsymbol{x}(t_i^-), \boldsymbol{x}(t_i^+), \boldsymbol{\omega}, t_i) := \begin{pmatrix} x(t_i^-) \\ y(t_i^-) \end{pmatrix} - \sum_{k=1}^{N-1} \omega_{i,k} \begin{pmatrix} x_k \\ y_k \end{pmatrix} \tag{25}$$

$$\boldsymbol{x}(t_i^+) = \boldsymbol{x}(t_i^-) \tag{26}$$

$$\sum_{i=1}^{N-1} \omega_{i,k} = 1, \quad \sum_{k=1}^{N-1} \omega_{i,k} = 1, \quad 0 \le \omega_{i,k} \le 1 \tag{27}$$

At the end of each phase the salesman must visit one of the $(N-1)$ cities $(x_k, y_k)^T$. This is ensured by (25). The linear constraints make sure, that each city is visited exactly once. Thus the final matrix $\Omega = (\omega_{i,k})_{i,k\in\{1,\ldots,N-1\}} \in \mathrm{I\!R}^{(N-1)\times(N-1)}$ has in each column and each row exactly one entry equal to 1. The other values are equal to 0. If $\omega_{i,k} = 1$, the k-th city is visited at the end of the i-th phase. Each tour is a permutation of the $(N-1)$ cities. Thus

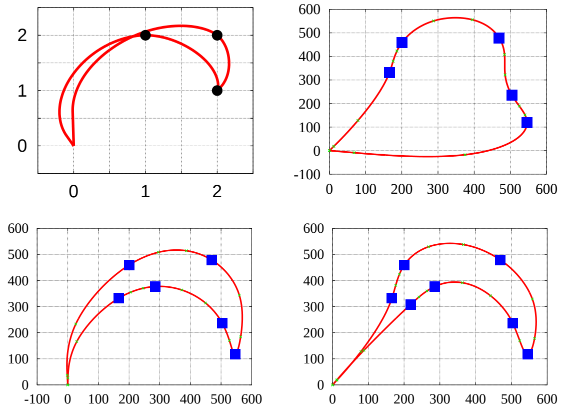

Fig. 15. Solutions for the MTSP for 3, 5, 6, and 7 cities

each feasible matrix Ω can be obtained by a permutation of the columns of the identity matrix.

If the salesman has to visit $(N-1)$ cities, then there are $(N-1)!$ possible tours, including the symmetric ones. Figure 15 shows solutions to three possible scenarios. In the present formulation $(N-1)^2$ binary values are used resulting in a branch & bound tree with a depth of $(N-1)^2$ and a breadth of $2^{(N-1)^2}$ nodes. The tree has $(2^{(N-1)^2+1}-1)$ nodes; most of them are infeasible though with respect to the linear constraints (27).

If a tree search is performed beginning at the root of the tree without the knowledge of an upper bound for the problem, at least $(N-1)^2$ nodes have to be analyzed to obtain an initial upper bound. In our numerical experiments, however, even more steps are usually needed to reach the leaves. Thus, the search for a optimum should begin at the leaves of the search tree until an initial upper bound is provided. The branch & bound algorithm starts afterwards to prove whether this bound is optimal (in convex cases) or to find a better one.

For each of the tours the continuous controls and switching times were optimized using the direct collocation method of Sect. 4.2 with respect to the terminal time t_f for a given discrete variable, i. e., order of cities, i. e., sequence of phases. To start the iterative direct collocation method, initial guesses for the switching points consisting of $t_{i,\text{estimate}} = i$, $i = 1, \ldots, N$, are used. A linear interpolation of the coordinates of the cities is applied as an initial guess for x and y, whereas v and a were initially set to zero. Computational times for obtaining a final solution can vary between a few minutes (for 5 cities) and several hours (for 7 cities) on a Pent. III, 900 MHz PC.

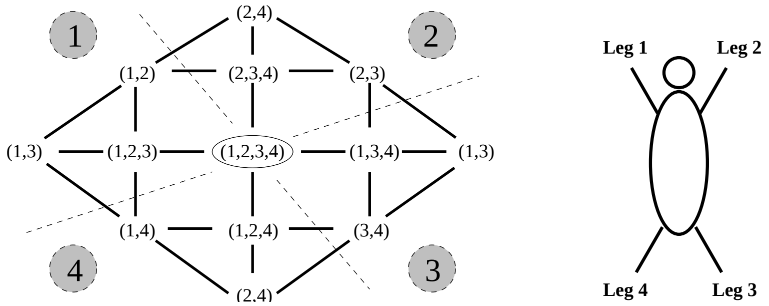

Fig. 16. Hybrid automaton for the quadruped. The nodes represent the different discrete states; the numbers in parentheses refer to the numbers of the support legs. Edges indicate discrete transitions (a leg has either broken ground contact or just entered a contact condition)

6 Other Problems

The robotic applications presented in this work serve primarily as illustrative examples to demonstrate the complexity existing in the optimal control of strongly interconnected discrete–continuous systems. A more realistic and challenging problem however that is currently being investigated using these approaches is the gait generation problem for four-legged robots. Quadrupeds are ideal for many applications due to their increased dexterity in comparison to legged robots with more legs and its increased stability compared to a biped. An unsolved problem, however, remains the determination of the optimal gait for moving at a given velocity where the order of leg movement and ground contact conditions at each moment in time are discrete characteristics of the problem. Preliminary work on this problem may be found in [27]. Fig. 16 displays the hybrid automaton for quadruped legged locomotion. Each node represents a different discrete state, where a different combination of legs are supporting the quadruped. A periodic gait is characterized, apart from the periodicity of its continuous states, by the discrete condition that each leg has exactly one period of ground contact and another period without contact during the gait. As a result, periodic gaits are represented by periodic paths which must visit all four quadrants in the hybrid automaton (Fig. 16) and then return to its starting point; thus, this problem is closely related to the MTSP.

The underlying HOCP for step sequence planning in humanoid walking is also an open challenge; see [31] for preliminary results combining step sequences from pre-calculated suboptimal step primitives. Another important robotic problem within this context is manipulation using multi-fingered dextrous robotic hands [38].

7 Conclusions

A methodology for the modeling and control of hybrid nonlinear dynamical systems is presented. The dynamical model, feedback solutions, and the numerical methods presented for the solution of hybrid optimal control problems are all geared towards the analysis of hybrid problems where the degree of discrete–continuous interconnection is strong, and the continuous dynamics may be highly nonlinear and of high dimension. In particular, the hybrid optimal control problem (HOCP) is defined and two approaches are described for its solution. The first approach decouples HOCPs by fixing interior point time and state constraints to a grid of possible values. Then, solutions to the decoupled TPBVPs are obtained, their optimal cost assigned to a graph with nodes representing the grid points and vertices the optimal cost. In this graph the best suboptimal solution is found by minimum path search. Alternatively, a branch-and-bound strategy is proposed based on the decomposition of HOCPs into MBOCPs. Binary variables are successively relaxed to obtain upper and lower bounds on the solutions. The search in the resulting solution tree is performed by branch-and-bound. The solutions to three hybrid control problems in robotics illustrate the effectiveness and scalability of the numerical methods presented here.

8 Acronyms

B&B	Branch and Bound
CE	Control Event
DE	Disturbance Event
EL-DEQ	Euler-Lagrange Differential Equation
HDS	Hybrid Dynamical System
HOCP	Hybrid Optimal Control Problem
HSM	Hybrid State Model
MBOCP	Mixed-Binary Optimal Control Problem
MTSP	Motorized Traveling Salesman Problem
NLP	Nonlinear Program
SQP	Sequential Quadratic Programming
SR	State Reset
SRVFS	State Reset and Vector Field Switch
TPBVP	Two Point Boundary Value Problem
TE	Time Event
VFS	Vector Field Switch

References

1. C.S. Adjiman, C.A. Schweiger, and C.A. Floudas. Mixed-integer nonlinear optimization in process synthesis. In *Handbook of Combinatorial Optimization*

(Du D.-Z., Pardalos P.M., eds.), volume 1, pages 1–76. Kluwer Acadademic Publisher, 1998.

2. J. Albro and J. Bobrow. Optimal motion primitives for a 5 dof experimental hopper. In *Proceedings of the IEEE International Conference on Robotics and Automation (Seoul, Korea)*, pages 3630–3635, 2001.
3. R.J. Allgor and P.I. Barton. Mixed integer dynamic optimization. *Computational Chemical Engineering*, 21:451–456, 1997.
4. A. Back, J. Guckenheimer, and M. Myers. A dynamical simulation facility for hybrid systems. In *Lecture Notes in Computer Science: Hybrid Systems (R. Grossmann, A. Nerode, A. Ravn, and H. Rischel, eds.)*, volume 736, pages 255–267. Springer Verlag, 1993.
5. A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3):407–427, 1999.
6. J. Betts. Survey of numerical methods for trajectory optimization. *AIAA Journal of Guidance, Control, and Dynamics*, 21(2):193–207, 1998.
7. M. Branicky. Topology of hybrid systems. In *Proceedings of the 32nd IEEE Conference on Decision and Control (San Antonio, TX)*, pages 2309–2314, 1993.
8. M. Branicky. Analyzing continuous switching systems: Theory and examples. In *Proceedings of the American Control Conference (Baltimore, MD)*, pages 3110–3114, 1994.
9. M. Branicky. Stability of switched and hybrid systems. In *Proceedings of the 33rd IEEE Conference on Decision and Control (Lake Buena Vista, FL)*, pages 3498–3503, 1994.
10. M. Branicky. A unified framework for hybrid control. In *Proceedings of the 33rd IEEE Conference on Decision and Control (Lake Buena Vista, FL)*, pages 4228–4234, 1994.
11. M. Branicky. *Studies in Hybrid Systems: Modeling, Analysis and Control.* PhD thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, 1995.
12. M. Branicky. General hybrid dynamical systems: Modeling, analysis, and control. In *Lecture Notes in Computer Science: Hybrid Systems III (R. Alur, T. Henzinger, and E. Sontag, eds.)*, volume 1066, pages 186–200. Springer Verlag, 1996.
13. M. Branicky. Multiple Lyapunov functions and other analysis tools for switched and hybrid systems. *IEEE Transactions on Automatic Control*, 43:475–482, 1998.
14. M. Branicky, V. Borkar, and S. Mitter. A unified framework for hybrid control: Model and optimal control theory. *IEEE Transactions on Automatic Control*, 43:31–45, 1998.
15. M. Branicky, R. Hebbar, and G. Zhang. A fast marching algorithm for hybrid systems. In *Proceedings of the 38th IEEE Conference on Decision and Control (Phoenix, AZ)*, pages 4897–4902, 1999.
16. R. Brockett. Hybrid models for motion control systems. In *Essays on Control: Perspectives in the Theory and its Applications (H. Trentelmann and J. Willems, eds.)*, pages 29–53. Boston: Birkhäuser, 1993.
17. M. Bühler and D. Koditschek. From stable to chaotic juggling: Theory, simulation, and experiments. In *Robot Control – Dynamics, Motion Planning, and Analysis (M. Spong, F. Lewis, and C. Abdallah, eds.)*, pages 525–530. New York: IEEE Press, 1993.

18. M. Buss. Control methods for hybrid dynamical systems – models, control loops, optimal control, computation tools, and mechatronic applications – (in german), 2000. Habilitation Dissertation, Institute of Automatic Control Engineering, Technische Universität München.
19. M. Buss, O. von Stryk, R. Bulirsch, and G. Schmidt. Towards hybrid optimal control. *at–Automatisierungstechnik*, 48:448–459, 2000.
20. J. Denk. Online optimal control strategies for mechatronic systems under multiple contact configurations. Technical report, Institute of Automatic Control Engineering, Technische Universität München, 1999. Internal Report.
21. M. Doğruel and Ü. Özgüner. Modeling and stability issues in hybrid systems. In *Lecture Notes in Computer Science: Hybrid Systems II (P. Antsaklis, W. Kohn, A. Nerode, and S. Sastry, eds.)*, volume 999, pages 148–165. Springer Verlag, 1995.
22. M. Doğruel, Ü. Özgüner, and S. Drakunov. Sliding-mode control in discrete-state and hybrid systems. *IEEE Transactions on Automatic Control*, 41:414–419, 1996.
23. S. Engell. Modellierung und analyse hybrider dynamischer systeme. *at–Automatisierungstechnik*, 45(4):152–162, 1997.
24. S. Engell, I. Hoffmann, and L. Sapronowa. Chaos in einfachen kontinuierlich-diskreten dynamischen systemen. *at–Automatisierungstechnik*, 45(9):399–406, 1997.
25. P. Gill, W. Murray, and M. Saunders. *User's guide for SNOPT 5.3: a fortran package for large-scale nonlinear programming.* Department of Mathematics, Univ. of California San Diego, 1997.
26. M. Hardt, J. Helton, and K. Kreutz-Delgado. Numerical solution of nonlinear $\mathcal{H}_2$ and $\mathcal{H}_\infty$ control problems with application to jet engine compressors. *IEEE Transactions on Control Systems Technology*, 8(1):98–111, 2000.
27. M. Hardt and O. von Stryk. Towards optimal hybrid control solutions for gait patterns of a quadruped. In *CLAWAR 2000 – 3rd International Conference on Climbing and Walking Robots, Madrid, 2–4 October, Professional Engineering Publishing, UK*, pages 385–392, 2000.
28. S. Hedlund and A. Rantzer. Optimal control of hybrid systems. In *Proceedings of the 38th IEEE Conference on Decision and Control (Phoenix, AZ)*, pages 3972–3977, 1999.
29. K. Kondak and G. Hommel. Computation of time optimal movements for autonomous parking of non-holonomic mobile platforms. In *Proceedings of the IEEE International Conference on Robotics and Automation (Seoul, Korea)*, pages 2698–2703, 2001.
30. G. Labinaz, M. Bayoumi, and K. Rudie. Modeling and control of hybrid systems: A survey. In *Preprints of the 13th World Congress (J. Gertler, J. Cruz, and M. Peshkin, eds.)*, volume C, pages 293–304. San Francisco: International Federation of Automatic Control – IFAC, 1996.
31. O. Lorch, J. Denk, J.F. Seara, M. Buss, F. Freyberger, and G. Schmidt. Vigwam — an emulation environment for a vision guided virtual walking machine. In *Proceedings of the First IEEE-RAS International Conference on Humanoid Robots HUMANOIDS 2000 (Cambridge, MA, USA)*, 2000.
32. J. Mareczek, M. Buss, and G. Schmidt. Robust Global Stabilization of the Underactuated 2-DOF Manipulator R2D1. In *Proceedings of the IEEE International Conference on Robotics and Automation (Leuven, Belgium)*, pages 2640–2645, 1998.

33. J. Mareczek, M. Buss, and G. Schmidt. Robust Control of a Non-Holonomic Underactuated SCARA Robot. In *Lecture Notes in Control and Information Sciences: Progress in System and Robot Analysis and Control Design (S. Tzafestas and G. Schmidt, eds.)*, volume 243, pages 381–396. Springer Verlag, 1999.
34. B. Martin and J. Bobrow. Minimum effort motions for open chain manipulators with task-dependent end-effector constraints. In *Proceedings of the IEEE International Conference on Robotics and Automation (Albuquerque, New Mexiko)*, pages 2044–2049, 1997.
35. G. Nenninger, M. Schnabel, and V. Krebs. Modellierung, Simulation und Analyse hybrider dynamischer Systeme mit Netz-Zustands-Modellen. *at–Automatisierungstechnik*, 47:118–126, 1999.
36. A. Nerode and W. Kohn. Models for hybrid systems: Automata, topologies, controllability, observability. In *Lecture Notes in Computer Science: Hybrid Systems (R. Grossmann, A. Nerode, A. Ravn, and H. Rischel, eds.)*, volume 736, pages 317–356. Springer Verlag, 1993.
37. T. Schlegl, M. Buss, and G. Schmidt. Development of numerical integration methods for hybrid (discrete-continuous) dynamical systems. In *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics AIM'97 (Tokyo, Japan, Paper No. 154)*, 1997.
38. T. Schlegl, M. Buss, and G. Schmidt. Hybrid control of multi-fingered dextrous hands. *This volume*, 2002.
39. T. Schlegl, M. Schnabel, M. Buss, and V. Krebs. State reconstruction and error compensation in discrete-continuous control systems. *at–Automatisierungstechnik*, 48:438–447, 2000.
40. L. Tavernini. Differential automata and their discrete simulators. *Nonlinear Analysis, Theory, Methods, and Applications*, 11:665–683, 1987.
41. C. Tomlin. Towards efficient computation of solutions to hybrid systems. In *Proceedings of the 38th IEEE Conference on Decision and Control (Phoenix, AZ)*, pages 3532–3537, 1999.
42. V. Utkin. *Sliding Modes in Control Optimization*. Springer Verlag, 1992.
43. A. van der Schaft and H. Schumacher. An introduction to hybrid dynamical systems. In *Lecture Notes in Control and Information Sciences*, volume 251. Springer Verlag, 2000.
44. O. von Stryk. Numerical hybrid optimal control and related topics, 2000. Habilitation Dissertation, Technische Universität München.
45. O. von Stryk. User's guide for DIRCOL version 2.1: A direct collocation method for the numerical solution of optimal control problems. Technical report, Simulation and Systems Optimization Group, Technische Universität Darmstadt, 2001. WWW: www.sim.informatik.tu-darmstadt.de/sw/.
46. O. von Stryk and R. Bulirsch. Direct and indirect methods for trajectory optimization. *Annals of Operations Research*, 36:357–373, 1992.
47. O. von Stryk and M. Glocker. Decomposition of mixed-integer optimal control problems using branch and bound and sparse direct collocation. In *ADPM – 4th Int'l Conf. on Automation of Mixed Processes: Hybrid Dynamic Systems*, pages 99–104, 2000.
48. O. von Stryk and M. Glocker. Numerical mixed-integer optimal control and motorized traveling salesmen problems. *APII – JESA (Journal européen des systèmes automatisés – European Journal of Control)*, 35(4):519–533, 2001.