Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

# On Optimal Control of Flat Hybrid Automata

**Frederik Zahn** * **Tobias Kleinert** ** **Veit Hagenmeyer** *

* *Institute for Automation and Applied Informatics, Karlsruhe Institute of Technology, Karlsruhe, Germany (e-mail: frederik.zahn, veit.hagenmeyer@kit.edu)*
** *Chair of Process Control Engineering, RWTH Aachen University, Aachen, Germany(e-mail: kleinert@plt.rwth-aachen.de).*

**Abstract:** Recently, Flat Hybrid Automata (FHA) were introduced as a new model class of hybrid systems. In order to answer the evident question of an optimal operation of the new class of FHA, a well-posed optimization problem definition for FHA is presented. This problem formulation also includes costs on discrete–state transitions and switching actions. We present a solution for a reduced problem: FHA with autonomous switching. For these, a new algorithm is introduced which solves the optimal control problem. The application of this optimization algorithm is shown with an electrical network example.

*Keywords:* Flat hybrid automata, differential flatness, hybrid and switched systems modeling, optimal control of hybrid systems

## 1. INTRODUCTION

Recently, Flat Hybrid Automata (FHA) were introduced as a new model class in Kleinert and Hagenmeyer (2019). This new subclass of hybrid systems offers controllability and explicit calculation of inputs. Further defining properties are determinism and strong connectedness (cf. Berman and Plemmons (1979)) of the discrete subsystem and differential flatness (cf. Fliess et al. (1995)) of the continuous subsystems. These properties enable straightforward trajectory planning (cf. Kleinert and Hagenmeyer (2019)) for the transition from an initial state to a final state and yield controllability and reachability. In this context, it is evident to ask the question of an optimal operation of the new class of FHA, hence a well-posed optimization problem definition for FHA and its solution are both open scientific questions.
Optimization problems have been studied for many subclasses of hybrid systems (Lunze and Lamnabhi-Lagarrigue (2009); Barton et al. (2006); Witsenhausen (1966)). In the present work we examine how an optimization problem for the FHA with its aforementioned properties can be formulated. Due to the dimorphic nature of hybrid systems such a problem consists of finding an optimal path through a graph (eg. Delling et al. (2009); Wagner and Willhalm (2003)) while simultaneously determining optimal inputs for the discrete system and the continuous subsystems. Optimization of the continuous flat subsystems involve trajectory planning (Hagenmeyer and Delaleau (2003b, 2008) and optimization of these (Sira-Ramirez and Agrawal (2004); Oldenburg and Marquardt (2002); Guay and Peters (2006)). The main contribution of the present paper is the formulation of an optimal control problem for FHA in general. This problem formulation also includes costs on discrete–state transitions and switching actions. Since this optimization problem is complex, a reduced class of FHA is considered: FHA with autonomous switching. For these, a new algorithm is introduced which yields a solution to the optimal control problem.
The present paper is organized as follows: We first present a compact definition of FHA in Section 2. For a complete introduction the reader is referred to Kleinert and Hagenmeyer (2019). In Section 3 we formulate the dynamic optimization problem for the FHA. In Section 4 autonomous switching is introduced for the FHA and an algorithm to solve the simplified optimization problem is presented. In Section 5 we apply the algorithm to an electrical network example.

## 2. FLAT HYBRID AUTOMATA

Representing a hybrid system by an automaton and continuous-time state-space models yields a hybrid automaton as described in (Goebel et al. (2009); Lunze and Lamnabhi-Lagarrigue (2009)). In the present work we consider hybrid automata with discrete and continuous subsystems with input and output. They show deterministic dynamical behavior in the sense that given the initial state and an input trajectory, the state and output trajectories exist and are unique.

### 2.1 Differential flatness

The key property of FHA is that the continuous dynamics are differentially flat. The definitions in this section apply to linear and nonlinear SISO and MIMO systems.

*Definition 1.* (Differential flatness). A system $\dot{x} = f(x, u)$ is differentially flat if there exists a bijective function

$$z = F(x, u, \dot{u}, \ddot{u}, \ldots, u^{(a)}) \quad (1)$$
$$x \in \mathbb{R}^{n_x}, u \in \mathbb{R}^{n_u}, z \in \mathbb{R}^{n_u}$$

which defines the so-called flat output (Fliess et al. (1995)). The superscript $(a)$ denotes the $a$-th time derivatives. Additionally, the functions

$$x = \Phi(z, \dot{z}, \ddot{z}, \ldots, z^{(b)}), \tag{2a}$$

$$u = \Psi(z, \dot{z}, \ddot{z}, \ldots, z^{(c)}) \tag{2b}$$

exist and can be explicitly derived. The maps $z, \Phi, \Psi$ should be of class $C^\infty$ or at least of class $C^r$ with $r$ sufficiently large.

For any flat system $\dot{x} = f(x, u)$, a given $(a+1)$-time differentiable trajectory $z^*(t), t \in [t_0, t_1]$ with consistent initial condition

$$x(t_0) = \Phi(z^*(t_0), \dot{z}^*(t_0), \ddot{z}^*(t_0), \ldots, z^{*(a)}(t_0)), \tag{3}$$

the inputs $u^*(t)$ and the states $x^*(t)$ can be explicitly calculated from Equations (2a) and (2b) without integrating differential equations (Fliess et al. (1995); Hagenmeyer and Delaleau (2003a,b)). The flat output of a system is generally not unique and therefore has to be specified explicitly for a given system. Flat systems are controllable as described in Fliess et al. (1995).

### 2.2 Mathematical Model

The discrete structure of the automaton is described as a directed graph with vertices $\mathsf{d}$ and edges $\mathsf{e} = (\mathsf{d}_i, \mathsf{d}_j)$. The vertex $\mathsf{d}_i$ is called tail, the vertex $\mathsf{d}_j$ head of the edge. To each discrete state we assign continuous dynamics with different domains and possibly different inputs. A Flat Hybrid Automaton consists of the following parts:

- Vertices $D = \{\mathsf{d}_1, \mathsf{d}_2, \ldots, \mathsf{d}_{n_\mathsf{d}}\}$ defining $n_\mathsf{d} \in \mathbb{N}$ discrete states.
- Edges $E : D \times D$ defining discrete state transitions as $\mathsf{e}_k = (\mathsf{d}_i, \mathsf{d}_j)$. The number of edges is $n_\mathsf{e} \in \mathbb{N}$.
- Continuous state spaces $\mathcal{X} = \left\{\mathcal{X}_{\mathsf{d}_1}, \mathcal{X}_{\mathsf{d}_2}, \ldots, \mathcal{X}_{\mathsf{d}_{n_\mathsf{d}}}\right\}$ with $\mathcal{X}_{\mathsf{d}_i} \subseteq \mathbb{R}^{n_{x_{\mathsf{d}_i}}}$.
- Continuous state transition functions $L = \{L_{\mathsf{e}_1}, L_{\mathsf{e}_2}, \ldots, L_{\mathsf{e}_{n_\mathsf{e}}}\}$, $L_{\mathsf{e}_k} : \mathcal{X}_{\mathsf{d}_i} \mapsto \mathcal{X}_{\mathsf{d}_j}$ which define the change of the continuous state for each transition $\mathsf{e}_k = (\mathsf{d}_i, \mathsf{d}_j)$.
- Discrete input space $V$. The discrete inputs are binary variables, i.e. $V = \{0, 1\}^{n_\mathsf{v}}$.
- Vector fields $\mathcal{F} = \left\{f_{\mathsf{d}_1}, f_{\mathsf{d}_2}, \ldots, f_{\mathsf{d}_{n_\mathsf{d}}}\right\}$ that define differential equations $\dot{x}_{\mathsf{d}_i} = f_{\mathsf{d}_i}(x_{\mathsf{d}_i}, u_{\mathsf{d}_i})$ on $\mathcal{X}_{\mathsf{d}_i} \times \mathcal{U}_{\mathsf{d}_i}$ describing the continuous dynamics in every discrete state. All $f$ have do be well posed such that a unique solution exists for some $t \in [t_0, t_1]$ and that all $f$ are Lipschitz given $x(t_0)$ and $u(t)$.
- Continuous input spaces $\mathcal{U} = \left\{\mathcal{U}_{\mathsf{d}_1}, \mathcal{U}_{\mathsf{d}_2}, \ldots, \mathcal{U}_{\mathsf{d}_{n_\mathsf{d}}}\right\}$ with $\mathcal{U}_{\mathsf{d}_i} \subseteq \mathbb{R}^{n_{u_{\mathsf{d}_i}}}$.
- Flat output maps $\mathsf{F} = \left\{F_{\mathsf{d}_1}, F_{\mathsf{d}_2}, \ldots, F_{\mathsf{d}_{n_\mathsf{d}}}\right\}$ and corresponding flat output spaces: $\mathcal{Z} = \left\{\mathcal{Z}_{\mathsf{d}_1}, \mathcal{Z}_{\mathsf{d}_2}, \ldots, \mathcal{Z}_{\mathsf{d}_{n_\mathsf{d}}}\right\}$. The dimension of the flat output space matches the dimension of the continuous input space, i.e. $\mathcal{Z}_{\mathsf{d}_i} \in \mathbb{R}^{n_{u_{\mathsf{d}_i}}}$. The maps $\Phi_\mathsf{d}$ and $\Psi_\mathsf{d}$ are defined for every $F_\mathsf{d}$.

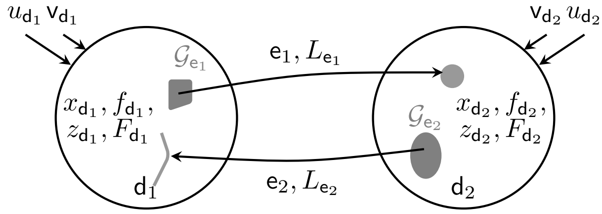


Fig. 1. Elements of the FHA

- Switching sets $\mathcal{G} = \{\mathcal{G}_{\mathsf{e}_1}, \mathcal{G}_{\mathsf{e}_2}, \ldots, \mathcal{G}_{\mathsf{e}_{n_\mathsf{e}}}\}$. These sets are uniquely defined for each edge $\mathsf{e}_k \in E$ as a set $\mathcal{G}_{\mathsf{e}_k} \subseteq \mathcal{Z} \times V$ on which state switching occurs.
- Initial condition $(\mathsf{d}^0, x^0_\mathsf{d})$ such that $x^0_\mathsf{d}$ is consistent according to Equation (3).

For the remainder of this work $\mathsf{d}$, or $\mathsf{d}(t)$ without subscript denotes any $\mathsf{d}_i \in D$, also when $\mathsf{d}$ is a subscript itself, the same applies to $\mathsf{e}$. At each discrete state $\mathsf{d}$ the continuous subsystem is described by the sets $\mathcal{X}_\mathsf{d}, \mathcal{U}_\mathsf{d}, \mathcal{Z}_\mathsf{d}$ and the maps $f_\mathsf{d}, F_\mathsf{d}$. Note that the dimensions of the spaces in the different discrete states can be different for each discrete state, e.g. $n_{x_{\mathsf{d}_i}} \neq n_{x_{\mathsf{d}_j}}$.

In Figure 1 all the parts of a FHA are depicted in a simple automaton with two discrete states. The variables $x_\mathsf{d}, u_\mathsf{d}, z_\mathsf{d}, \mathsf{v}$ are vectors from the corresponding spaces and the evolution of $x_\mathsf{d}, z_\mathsf{d}$ is defined through $f_\mathsf{d}, F_\mathsf{d}$. When the FHA reaches a state where $(\mathsf{v}, z_\mathsf{d})$ belong to a set $\mathcal{G}_\mathsf{e}$, state switching takes place immediately. State switching is defined by $\mathsf{e} = (\mathsf{d}_i, \mathsf{d}_j)$ and the change in the continuous state $x_\mathsf{d}$ is described by $L_\mathsf{e}$. We denote the states before and after state switching $\mathsf{d}^-, x^-_\mathsf{d}$ and $\mathsf{d}^+, x^+_\mathsf{d}$. In order to establish deterministic behavior of the FHA, all switching sets have to be disjoint, such that all possible pairs $(\mathsf{v}, z_\mathsf{d})$ either belong to no set from $\mathcal{G}$ or to exactly one, i.e. $\mathcal{G}\mathsf{e}_i \cap \mathcal{G}\mathsf{e}_j = \emptyset$ for $i \neq j$. The aggregation of the sets and maps that form the structure of the FHA is the data of the automaton.

### 2.3 Trajectories and solutions

The trajectory of a FHA is described on a hybrid time domain. This time domain consists of an ordered set of intervals in which the evolution of the FHA is defined by continuous motion. The boundaries of the intervals are time instants at which state switching occurs. A hybrid time domain is defined as a set $\mathcal{T} \subseteq \mathbb{R}_{\geq 0} \times \mathbb{N}$. The pairs $(t_j, j) \in \mathcal{T}$ define the union of time intervals of the form $[t_j, t_{j+1}] \times \{j\}$ such that $0 = t_0 \leq t_1 \leq \cdots \leq t_m$. At each time instant $t_1, t_2, \ldots$ state switching occurs except for initial time $t_0$ and final time $t_m$. The variable $\tau$ denotes any time instant on the hybrid time domain $\mathcal{T}$. The hybrid state of a FHA is a pair $(\mathsf{d}, x_\mathsf{d})$ where $\mathsf{d} \in D, x_\mathsf{d} \in \mathcal{X}_\mathsf{d}$.
A solution of a FHA consists of a hybrid time domain $\mathcal{T}$, an initial state $(\mathsf{d}^0, x^0_\mathsf{d})$ at $t_0$ and a pair $(\mathsf{d}(\tau), x_\mathsf{d}(\tau))$ that is generated by admissible inputs $u_\mathsf{d}(\tau), \mathsf{v}(\tau)$ and satisfies the data of the FHA (Goebel et al. (2009)). On each time interval $[t_j, t_{j+1}] \times \{j\}$ the discrete state $\mathsf{d}$ is constant and the continuous state $x_\mathsf{d}$ is a continuous function of $\tau$. The flat output to the solution is $z^*(\tau)$.
For the definition of a hybrid trajectory we need a representation of the discrete state transition and the time at which they take place. These can be written as an ordered

set $P = \{\mathsf{e}_i, \mathsf{e}_j, \dots\}$ that is called the path. Combining a path and a hybrid time domain $\mathcal{T}$ we define a path function $P_{\mathsf{e}}(\tau) : \mathcal{T} \mapsto \{E, (0,0)\}$ where $(0,0)$ denotes the case that no transition takes place. This function uniquely defines the discrete state transitions on a hybrid time domain and trivially includes the path. Remark that we only consider single transitions at a time instant (Multiple transitions at one time instant $t_n$ and Zeno behavior can be integrated but are beyond the scope of this work). The flat output $z^*(\tau)$ of a solution of a FHA with corresponding hybrid time domain $\mathcal{T}$, initial state $(\mathsf{d}^0, x^0_{\mathsf{d}})$ and path function $P_{\mathsf{e}}$ is called flat hybrid output trajectory. Flat hybrid output trajectories define all states, state transitions and inputs of a FHA on the given hybrid time domain.

### *2.4 Controllability*

FHA are controllable in the sense that from any initial state $(\mathsf{d}^0_i, x^0_{\mathsf{d}})$ every $(\mathsf{d}_i, z_{\mathsf{d}})$ can be reached through admissible inputs $u$ and $\mathsf{v}$.
For the discrete part of the system, which is represented by the automaton graph $(D, E)$, reachability is given if the graph is strongly connected (Berman and Plemmons (1979)). That means that every discrete state of the graph can be reached from any other discrete state through a sequence of discrete state transitions $\mathsf{e}$.
The continuous dynamics of a FHA can be controlled to all flat outputs $z_{\mathsf{d}} \in \mathcal{Z}_{\mathsf{d}}$, i.e. in all subsystems $\mathsf{d}_i$ for any $x_{\mathsf{d}_i}(\tau_1)$, there exist inputs $u_{\mathsf{d}_i}(\tau), \tau \in [\tau_1, \tau_2]$ that bring the system to any $z_{\mathsf{d}_i}(\tau_2)$ with $\tau_2 < \infty$ (Sontag (2013); Faulwasser et al. (2014)). This is guaranteed by differential flatness and the proper definition of $\mathcal{Z}$ and $\mathcal{U}$ (Fliess et al. (1995); Hagenmeyer and Delaleau (2003a,b)). For controllability of the complete FHA we further need all switching sets $\mathcal{G}$ in $\mathsf{d}_i$ for discrete state transitions with tail $\mathsf{d}_i$ to be reachable independently, i.e. without passing through any other switching set. Since these defined on $V \times \mathcal{Z}_{\mathsf{d}_i}$, controllability to the flat outputs is a sufficient condition as long as no constraints are imposed on the discrete inputs.

### *2.5 The Flat Hybrid Automaton*

The previous definitions define the FHA: It is a deterministic hybrid system with differentially flat continuous dynamics, a strongly connected automaton graph and it is controllable to every $z_{\mathsf{d}}$. The FHA can be partitioned into two interconnected subsystems $A$ and $C$. This yields a collection $\{A, C\}$ where $A = \{D, E, \mathcal{G}, L, V\}$ describes the automaton structure and $C = \{\mathcal{X}, \mathcal{Z}, \mathcal{F}, \mathsf{F}, \mathcal{U}\}$ contains the continuous dynamics.

## 3. DEFINITION OF THE OPTIMAL CONTROL PROBLEM OF FHA

FHA are controllable as shown in Section 2. Hence for two arbitrary hybrid states $(\mathsf{d}_0, z_0)$ and $(\mathsf{d}_{t_f}, z_{t_f})$ with end time $t_f$, it is always possible to construct input functions $u^*(\tau), \mathsf{v}^*(\tau)$ that bring a FHA from the initial state to the final state. These inputs together with the consistent initial condition (3) uniquely define all states and transitions of a FHA. For control systems a natural question is how to find input functions $u^\star(\tau), \mathsf{v}^\star(\tau)$ that achieve the state transition and that simultaneously minimize a cost functional $J(\cdot)$. In control theory this problem is called optimal control problem (Luenberger (1997)). In the following we define a problem statement for optimal control problems for the novel class of FHA. Note that thanks to the flatness of the continuous subsystems, we can directly construct output functions $z^*(\tau)$ instead of $u^*(\tau)$, cf. (2a), and use these for the optimization, cf. Oldenburg and Marquardt (2002); Sira-Ramirez and Agrawal (2004); Guay and Peters (2006).

*Problem 1.* Given a FHA $= \{\mathrm{A}, \mathrm{C}\}$ and two hybrid states $(\mathsf{d}_0, z_0)$, $(\mathsf{d}_{t_f}, z_{t_f})$, find the path $P = \{\mathsf{e}_{\varsigma 1}, \mathsf{e}_{\varsigma 2}, ..., \mathsf{e}_{\varsigma n}\}$ defined through the sequence of flat outputs $\left\{ z^\star_{\mathsf{d}_{\xi 0}}, z^\star_{\mathsf{d}_{\xi 1}}, \dots, z^\star_{\mathsf{d}_{\xi n}} \right\} \in \mathcal{Z}_{\mathsf{d}_{\xi i}}$ and the discrete inputs $\mathsf{v}^\star$ that yields

$$\min_{\{z_{\mathsf{d}_{\xi i}}(\tau)\}, \mathsf{v}(\tau)} J(\cdot) = \sum_{i=0}^{n} \alpha(z_{\mathsf{d}_{\xi i}}(\tau)) + \sum_{i=1}^{n} \beta(\mathsf{e}_{\varsigma i}) + \gamma(\mathsf{v}(\tau)) \tag{4}$$

over all possible $n$. The minimization problem is subject to

$$\alpha(z_{\mathsf{d}_{\xi i}}(\tau)) = \int_{t_i}^{t_{i+1}} L_i(\Phi_{\mathsf{d}_{\xi i}}(z_{\mathsf{d}_{\xi i}}(\zeta)), \Psi_{\mathsf{d}_{\xi i}}(z_{\mathsf{d}_{\xi i}}(\zeta))) \mathrm{d}\zeta, \tag{5a}$$

$$t_0 \leq t_i \leq t_{i+1}, \tag{5b}$$

$$(\mathsf{d}(t_0), z(t_0)) = (\mathsf{d}_0, z_0), \tag{5c}$$

$$(\mathsf{d}(t_{n+1}), z(t_{n+1})) = (\mathsf{d}_{t_f}, z_{t_f}), \tag{5d}$$

$$0 \leq \boldsymbol{c}(\Phi_{\mathsf{d}_{\xi i}}(z_{\mathsf{d}_{\xi i}}(\tau)), \Psi_{\mathsf{d}_{\xi i}}(z_{\mathsf{d}_{\xi i}}(\tau))), \;\; i \in [0, \dots, n]. \tag{5e}$$

The cost functional $J$ consists of an integrable functional $L_i : \mathbb{R}^{n_{z_{\mathsf{d}_{\xi i}}} \times b_{\mathsf{d}_{\xi i}}} \times \mathbb{R}^{n_{z_{\mathsf{d}_{\xi i}}} \times c_{\mathsf{d}_{\xi i}}} \times \mathbb{R} \to \mathbb{R}$, a function $\beta : E \to \mathbb{R}$ and a function $\gamma : \{0,1\}^{n\mathsf{v}} \to \mathbb{R}$. The constraints (5e) are vector valued functionals $\boldsymbol{c} : \mathbb{R}^{n_{z_{\mathsf{d}_{\xi i}}} \times b_{\mathsf{d}_{\xi i}}} \times \mathbb{R}^{n_{z_{\mathsf{d}_{\xi i}}} \times c_{\mathsf{d}_{\xi i}}} \times \mathbb{R} \to \mathbb{R}^{n\boldsymbol{c}}$.

At every switching time $t_i$, the decision variables $z_{\mathsf{d}_{\xi i}}(t_i)$, $\mathsf{v}(t_i)$ have to fulfill the switching rule $\mathcal{G}_{\mathsf{e}_{\varsigma i}}(\cdot)$. Furthermore the outputs $z^-_{\mathsf{d}_{\xi i}}(t_i)$ and $z^+_{\mathsf{d}_{\xi i+1}}(t_i)$ are constrained by $L$. For the sake of simplicity no explicit constraint on the derivatives of $z_{\mathsf{d}_{\xi i}}$ is included in the problem although this would be possible.
Note that a cost on each switching action is included in this framework: The discrete input of FHA $\mathsf{v}(\tau)$ is included in the cost through the term $\gamma(\mathsf{v}(\tau))$. The term $\beta(\mathsf{e}_{\varsigma i})$ can be used to put a cost on state transitions.
Note that the final time $t_{n+1} = t_f$ is not fixed in the problem definition and depends on the inputs. Fixed final times and fixed transition times can be implemented via additional constraints. Remark that the constraints (5e) have to be designed such that the controllability of the FHA is not lost, which implies that all switching sets remain reachable. In the next section we present a simplified version of the above problem.

## 4. OPTIMIZATION PROBLEM OF A REDUCED FHA AND A NEW ALGORITHM FOR ITS SOLUTION

Problem 1 consists of a path search and coupled optimal control problems with integer decision variables $\mathsf{v}$. We present an approach to solve the problem for a simplified class of FHA with discrete inputs $\mathsf{v}$ that depend on the flat outputs:

$$\mathsf{v}(z_{\mathsf{d}}(\tau)). \tag{6}$$

Thus the discrete inputs $\mathsf{v}$ do not appear in the optimal control problem anymore which eliminates one source of complexity. This is called autonomous switching because the switching sets only depend on $z$. Hence in the simplified version only the function of the flat output $z_{\mathsf{d}}(\tau)$ is a decision variable. It is still possible, however, to put a cost on the discrete inputs $\mathsf{v}$. These can be considered as binary state variables rather than exogenous system inputs in this setting. In a technical system this corresponds to automated switches that can not be operated independently.
The solution of the simplified problem consists of two elements: The optimal path $P^\star$ and the optimal flat outputs $z^*$ that yield the trajectories for every continuous subsystem that is traversed by $P^\star$. For any given path connecting $\mathsf{d}_0$ and $\mathsf{d}_{t_f}$ we can calculate optimal flat outputs $z^*_P$ that minimize $J(\cdot)$ for this particular path. In the following, we only consider non-negative cost $J(\cdot)$ with $L(\cdot)$, $\beta(\cdot)$, $\gamma(\cdot)$ all being non-negative.

*Proposition 1.* To find the optimal path $P^\star$ and the optimal flat output $z^*$ for a FHA with autonomous switching we propose Algorithm 1:

**Algorithm 1** OptPathAutoSwitchFHA

(1) Compute all possible paths $P_j$ through the FHA connecting $\mathsf{d}_0$ and $\mathsf{d}_{t_f}$ without visiting any node twice
(2) For every path $P_j$ construct the optimization problem according to Eq. (4) and (5)
(3) Solve all optimization problems to compute $\min_{\{z_{\mathsf{d}_{\xi i}}(\tau)\}} J(\cdot)$
(4) Compare results and find path $P^\star$ with minimal $J^\star(\cdot)$

*Remark 2.* We exclude cycles in the optimal path, i.e. every node can be visited only once in a path, because otherwise step 1 in Algorithm 1 becomes infeasible.

*Remark 3.* We do not provide a proof that excluding cycles actually yields the global optimal solution. This proof remains an open problem and is beyond the scope of this work. However, for sufficiently high costs $\beta(\cdot)$, $\gamma(\cdot)$ on switching actions and positive cost $J(\cdot)$ the algorithm leads to the optimal solution, because if the cost on switching is higher than the cost of the continuous dynamics between switching instants, it can not be optimal to visit nodes multiple times.

The number of paths can become very large for graphs with many states and edges. In that case the algorithm may become numerically intractable and existing heuristic algorithms to exclude paths may be needed (Delling et al. (2009)).

## 5. DC NETWORK EXAMPLE

In the following, the algorithm presented in the last section is applied to an example inspired by Gensior et al. (2006) and discussed in Kleinert and Hagenmeyer (2019). The electrical DC network in Figure 2 with two variable power sources $V_{in1}$ and $V_{in2}$ and two fluctuating loads $R_{L1}$ and $R_{L2}$ is considered. Two switches – described by the discrete inputs $\mathsf{v}_1 \in 0,1$ and $\mathsf{v}_2 \in 0,1$ – allow to configure the network with increased or decreased damping and coupling properties. The continuous inputs $V_{in1}$ and $V_{in2}$ control the voltage of load 1 ($v_{L1}$) and the current of load 2 ($i_{L2}$). The switch positions of $\mathsf{v}_1$ and $\mathsf{v}_2$ yield four discrete states of a continuous system. It is assumed that for low load the switches are set to zero, i.e.

if $v_{L1} < v_0$ then $\mathsf{v}_1 = 0$, else $\mathsf{v}_1 = 1$
if $i_{L2} < i_0$ then $\mathsf{v}_2 = 0$, else $\mathsf{v}_2 = 1$ .

Thereby, the capacitor is available to dampen step fluctuations of, e.g., $R_{L1}$ and $R_{L2}$, in case of higher network load. Thus the discrete inputs depend on the continuous states which yields autonomous switching (c.f Eq. (6)).

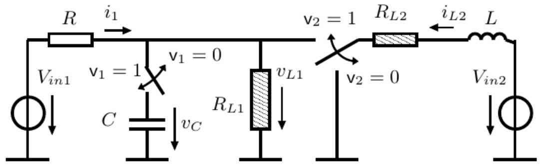


Fig. 2. Electrical DC network example

For the four discrete states, the continuous flat outputs are $z_1 = v_{L1}$ and $z_2 = i_{L2}$ and the continuous inputs are $u_1 = V_{in1}$ and $u_2 = V_{in2}$. Equation (7) describes the dynamics of the system:

$$\begin{aligned}
L\frac{\mathrm{d}i_{L2}}{\mathrm{d}t} &= V_{in2} - (R_{L2}\, i_{L2} + \mathsf{v}_2\, v_{L1}) \\
C\frac{\mathrm{d}v_C}{\mathrm{d}t} &= \mathsf{v}_1\,(i_1 - (\frac{1}{R_{L1}}\, v_{L1} - \mathsf{v}_2\, i_{L2})) \\
\mathsf{v}_1 C\frac{\mathrm{d}v_C}{\mathrm{d}t} &= i_1 - (\frac{1}{R_{L1}}\, v_{L1} - \mathsf{v}_2\, i_{L2}) \\
V_{in1} &= R i_1 + \mathsf{v}_1 v_C + (1-\mathsf{v}_1) v_{L1} \\
\mathsf{v}_1 v_C &= \mathsf{v}_1 v_{L1}
\end{aligned} \tag{7}$$

The initial condition for system (7) is $v_C(t_0) = v_{C,0}$, $i_{L2}(t_0) = i_{L2,0}$. Permuting the discrete inputs $\mathsf{v}_1$ and $\mathsf{v}_2$ in system (7) by their values $0,1$ yields the continuous system equations for the respective discrete states. The continuous subsystems are flat with respective $\Psi_{\mathsf{d}i}$:

$\mathsf{d}_1$: $\mathsf{v}_1 = 0$, $\mathsf{v}_2 = 0$
$$u_1 = \left(\frac{R}{R_{L1}} + 1\right) z_1$$
$$u_2 = L\,\dot{z}_2 + R_{L2}\, z_2$$

$\mathsf{d}_2$: $\mathsf{v}_1 = 0$, $\mathsf{v}_2 = 1$
$$u_1 = \left(\frac{R}{R_{L1}} + 1\right) z_1 - R\, z_2$$
$$u_2 = L\,\dot{z}_2 + R_{L2}\, z_2 + z_1$$

$\mathsf{d}_3$: $\mathsf{v}_1 = 1$, $\mathsf{v}_2 = 0$
$$u_1 = R\,C\,\dot{z}_1 + \left(\frac{R}{R_{L1}} + 1\right) z_1$$
$$u_2 = L\,\dot{z}_2 + R_{L2}\, z_2$$

$\mathsf{d}_4$: $\mathsf{v}_1 = 1$, $\mathsf{v}_2 = 1$
$$u_1 = R\,C\,\dot{z}_1 + \left(\frac{R}{R_{L1}} + 1\right) z_1 - R\, z_2$$
$$u_2 = L\,\dot{z}_2 + R_{L2}\, z_2 + z_1$$

In Figure 3 the automaton graph of the network is depicted with all possible transitions. All calculations use the following parameters [1]:

$R = 5$, $C = 0.8$, $L = 7$, $R_{L1} = 2$, $R_{L2} = 3$,
$v_0 = 6$, $i_0 = 2.5$ .

According to Problem 1 we have to define a cost functional

[1] $[V] = [v] = \mathrm{V}$, $[i] = \mathrm{mA}$, $[R] = \mathrm{k}\Omega$, $[C] = \mathrm{F}$, $[L] = \mathrm{kH}$.

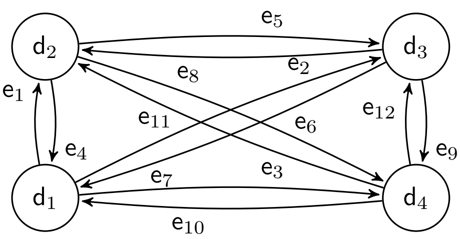


Fig. 3. Automaton graph of the DC network example

like (5a) for the optimization problem. We choose to put a cost on the continuous input $u$ with Lagrange term

$$L(\cdot) = u_1^2(\tau) + u_2^2(\tau) \tag{8}$$

and costs on the switching actions with $\beta(e_i)$. Thanks to autonomous switching, in this example costs to opening and closing of the switches $\mathsf{v}_1$ and $\mathsf{v}_2$ are equivalent to costs on the transitions $e_i$. We define

| Input | Switching | Cost | Input | Switching | Cost |
|---|---|---|---|---|---|
| $\mathsf{v}_1$ : | $0 \to 1$ | 10 | $\mathsf{v}_2$ : | $0 \to 1$ | 20 |
| | $1 \to 0$ | 100 | | $1 \to 0$ | 200 |

such that for example transition cost $\beta(e_5) = 210$. In this case we put a much higher cost on opening switches than on closing switches. Note that the transition cost $\beta(e_i)$ is purely additive. Therefore it has no effect on the continuous trajectories $u^\star$. We choose to impose a further constraint on the flat outputs at the switching times

$$z^-_{\mathsf{d}_{\xi i-1}}(t_i) = z^+_{\mathsf{d}_{\xi i}}(t_i),\ i \in [1, n] \tag{9}$$

such that no jumps in $z$ can occur. This also represents the continuous-state transition function $L$. From the dynamics of the system we deduce that this constraint can lead to jumps in the optimal input $u^*$ at switching times. If one wants to achieve continuity of the optimal input $u^*$ instead, Equation (9) can be rewritten accordingly. It is, in general, not possible to have continuity in both $z^*$ and $u^*$. We further choose

$$t_{i+1} - t_i \geq 0.5 \tag{10}$$

to avoid very fast transitions mainly in order to obtain comprehensible plots. Thus the minimum time between discrete state switching is constrained.

For the numerical solutions we choose to approximate the flat outputs $z$ as polynomial expansions like proposed by Sira-Ramirez and Agrawal (2004); Oldenburg and Marquardt (2002) such that

$$\tilde{z}_{\mathsf{d}_{\xi i}}(\tau) = \sum_{k=0}^{K} \theta_{\mathsf{d}_{\xi i},k}\phi_k(\tau). \tag{11}$$

In this representation the flat outputs in every continuous subsystem are characterized by a set of polynomials $\{\phi_k\}_{k=0,\ldots,K}$ and the coefficients $\theta_{\mathsf{d}_{\xi i},k}$. The polynomials are chosen to be fixed, hence the remaining decision variables are only the coefficients $\theta_{\mathsf{d}_{\xi i},k}$. This transforms the infinite dimensional function space of possible solutions to a fixed finite set of real numbers. Thus the shape of the flat output trajectory depends on the polynomials. These can be arbitrarily complicated, e.g. splines or higher order expansions. According to Problem 1 we are looking for a sequence of flat outputs. The number of coefficients of the whole problem therefore depends on the number of polynomials $K$ and the length of the path $n$.

We choose the most simple set of polynomials

$$\phi_0(\tau) = 1,\ \phi_1(\tau) = \tau \tag{12}$$

that yields affine functions for the flat output and

$$\tilde{z}_{\mathsf{d}}(\tau) = \theta_{\mathsf{d},1} + \theta_{\mathsf{d},2}\tau$$
$$\dot{\tilde{z}}_{\mathsf{d}}(\tau) = \theta_{\mathsf{d},2}.$$

These polynomials are chosen for the sake of simplicity. Remark, however, that it is not possible to steer the system into a steady state or to achieve higher order continuity for the optimal solutions with this approach. We first examine a simplified example before we demonstrate how to solve the complete problem.

*Example 1* In a first example we consider the case in which $\mathsf{v}_2 \equiv 0$ such that we only have the discrete states $\mathsf{d}_1, \mathsf{d}_3$ and $u_2 = z_2 \equiv 0$. Hence the right part of the network in Figure 2 is effectively deactivated. This is a simplified example to illustrate the application of Algorithm 1 of Section 4. The dynamics of the model are thereby reduced to

$\mathsf{d}_1$: $\mathsf{v}_1 = 0$
$$u_1 = \left(\frac{R}{R_{L1}} + 1\right) z_1$$
$\mathsf{d}_3$: $\mathsf{v}_1 = 1$
$$u_1 = R\,C\,\dot{z}_1 + \left(\frac{R}{R_{L1}} + 1\right) z_1$$

We set $v_{L1}(t_0) = 0.5$, $\mathsf{d}_0 = \mathsf{d}_1$ and $v_{L1}(t_f) = 18$, $\mathsf{d}_{t_f} = \mathsf{d}_3$.

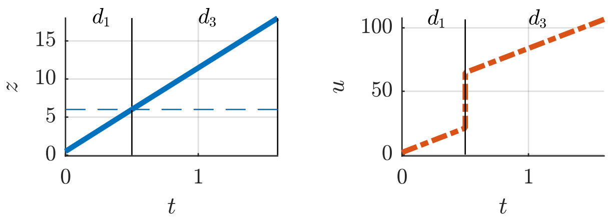


Fig. 4. Optimal flat outputs and inputs for Example 1

Applying the first step of Algorithm 1 we obtain the only possible path $P = \{\mathsf{e}_2\}$ which trivially is $P^\star$. The result of the optimization in Figure 4 shows $z^*$ and $u^*S$ with a jump in the input. In the first part in state $\mathsf{d}_1$ the trajectory is steeper than in the second part which is not surprising as $u_{1,d_1}$ does not include the derivative of $z$.

*Example 2* Considering the whole network with the dynamics described in (7) we apply Algorithm 1. We choose the initial and the final states

| | | |
|---|---|---|
| $v_{L1}(t_0) = 0.5$ | $i_{L2}(t_0) = 0.1$ | $\mathsf{d}_0 = \mathsf{d}_1$ |
| $v_{L1}(t_f) = 12$ | $i_{L2}(t_f) = 4$ | $\mathsf{d}_{t_f} = \mathsf{d}_4$. |

The possible paths and corresponding sequences without visiting any discrete state twice are

| | |
|---|---|
| $P_1 = \{\mathsf{e}_3\}$ | $S_1 = \{\mathsf{d}_1, \mathsf{d}_4\}$ |
| $P_2 = \{\mathsf{e}_1, \mathsf{e}_6\}$ | $S_2 = \{\mathsf{d}_1, \mathsf{d}_2, \mathsf{d}_4\}$ |
| $P_3 = \{\mathsf{e}_2, \mathsf{e}_9\}$ | $S_3 = \{\mathsf{d}_1, \mathsf{d}_3, \mathsf{d}_4\}$ |
| $P_4 = \{\mathsf{e}_1, \mathsf{e}_5, \mathsf{e}_9\}$ | $S_4 = \{\mathsf{d}_1, \mathsf{d}_2, \mathsf{d}_3, \mathsf{d}_4\}$ |
| $P_5 = \{\mathsf{e}_2, \mathsf{e}_8, \mathsf{e}_6\}$ | $S_5 = \{\mathsf{d}_1, \mathsf{d}_3, \mathsf{d}_2, \mathsf{d}_4\}$. |

Applying steps 2 and 3 of Algorithm 1 we construct and solve an optimization problem for every path. The resulting trajectories are depicted in Figure 5. The evaluation

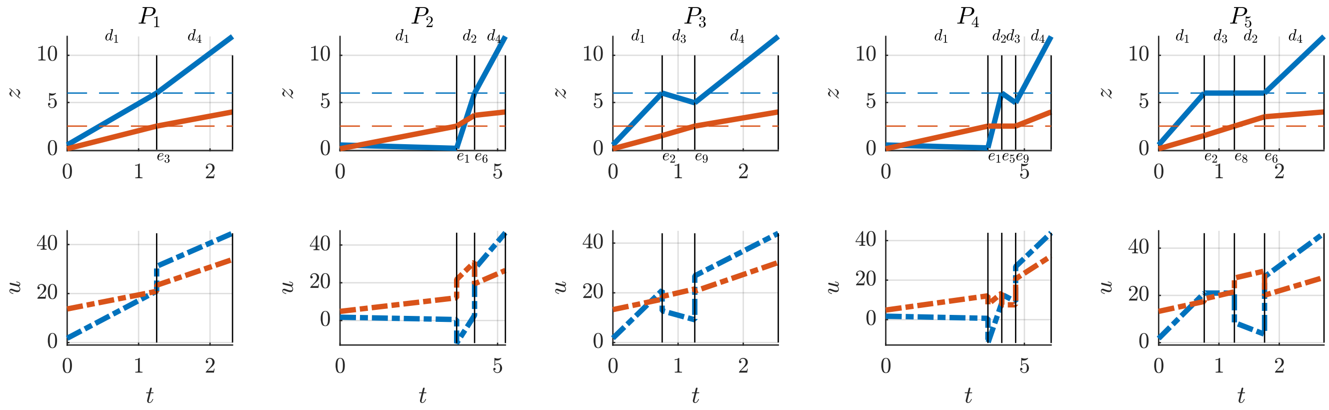

Fig. 5. Trajectories for all paths for Example 2. Blue lines $z_1$, $u_1$; red lines $z_2$, $u_2$. The vertical black lines show switching times $t^\star$, the horizontal dashed lines show $v_0$ and $i_0$.

of the value function in Table 1 shows that path $P_2$ yields the lowest cost and therefore is the optimal path $P^\star$. Consequently the corresponding trajectories in Figure 5 for path $P_2$ are $z^*$ and $u^*$. Interestingly, we see that the "shortest" path $P_1$ with only one transition yields higher cost than $P_2$. If high costs on $\mathsf{e}_1$ or $\mathsf{e}_6$ were included in the problem in order to avoid the "longer" path, path $P_1$ would become the optimal path, since for time invariant costs $\gamma$ and $\beta$ the trajectories in Figure 5 would not change.

Table 1. Value of $J^\star_{P_j}(\cdot)$ for all paths

| | |
|---|---|
| $P_1 = \{\mathsf{e}_3\}$ | 3060 |
| $\mathbf{P_2} = \{\mathsf{e}_1, \mathsf{e}_6\}$ | **2633** |
| $P_3 = \{\mathsf{e}_2, \mathsf{e}_9\}$ | 3187 |
| $P_4 = \{\mathsf{e}_1, \mathsf{e}_5, \mathsf{e}_9\}$ | 3476 |
| $P_5 = \{\mathsf{e}_2, \mathsf{e}_8, \mathsf{e}_6\}$ | 3409 |

## 6. SUMMARY AND OUTLOOK

The present work defines an optimal control problem for the new system class of Flat Hybrid Automata (FHA) with possible costs on discrete–state transitions and switching actions. In a second step, a reduced class of FHA is considered: FHA with autonomous switching. For these, a new algorithm is introduced solving the optimization problem. It is applied to an electrical network example.

For future work, it remains an open question under which conditions —on the switching costs and in general— the algorithm is able to find the solution to the problem. In the example the flat outputs are restricted to polynomials. More sophisticated base functions, e.g. splines, can be considered. Moreover, for future research it is of interest to further investigate optimization for the FHA in general – and to use the respective solution for flatness-based MPC schemes (cf. Hagenmeyer and Delaleau (2008)).

## REFERENCES

Barton, P.I., Lee, C.K., and Yunt, M. (2006). Optimization of hybrid systems. *Computers & chemical engineering*, 30(10-12), 1576–1589.

Berman, A. and Plemmons, R. (1979). *Nonnegative Matrices in the Mathematical Sciences*. Academic Press, New York.

Delling, D., Sanders, P., Schultes, D., and Wagner, D. (2009). *Engineering route planning algorithm*, 117–139. LNCS 5515, Springer.

Faulwasser, T., Hagenmeyer, V., and Findeisen, R. (2014). Constrained reachability and trajectory generation for flat systems. *Automatica*, 50(4), 1151–1159.

Fliess, M., Levine, J., Martin, P., and Rouchon, P. (1995). Flatness and defect of nonlinear systems: Introductory theory and examples. *Int. Journal of Control*, 61(6), 1327–1361.

Gensior, A., Woywode, O., Rudolph, J., and Güldner, H. (2006). On Differential Flatness, Trajectory Planning, Observers, and Stabilization for DC-DC Converters. *IEEE Transactions On Circuits And Systems–I*, 53(9), 2000–2010.

Goebel, R., Sanfelice, R.G., and Teel, A.R. (2009). Hybrid dynamical systems. *IEEE Control Systems Magazine*, 29(2), 28–93.

Guay, M. and Peters, N. (2006). Real-time dynamic optimization of nonlinear systems: A flatness-based approach. *Computers & chemical engineering*, 30(4), 709–721.

Hagenmeyer, V. and Delaleau, E. (2003a). *Exact feedforward linearisation based on differential flatness: The SISO case*, 161–170. In Nonlinear and Adaptive Control, Springer, Berlin, Heidelberg.

Hagenmeyer, V. and Delaleau, E. (2003b). Exact feedforward linearization based on differential flatness. *International Journal of Control*, 76(6), 537–556.

Hagenmeyer, V. and Delaleau, E. (2008). Continuous-time nonlinear flatness-based predictive control: an exact feedforward linearisation setting with an induction drive example. *International Journal of Control*, 81(10), 1645–1663.

Kleinert, T. and Hagenmeyer, V. (2019). Flat hybrid automata as a class of reachable systems: Introductory theory and examples. *CoRR*, abs/1906.02790. URL `http://arxiv.org/abs/1906.02790`.

Luenberger, D.G. (1997). *Optimization by vector space methods.* John Wiley & Sons.

Lunze, J. and Lamnabhi-Lagarrigue, F. (2009). *Handbook of hybrid systems control: theory, tools, applications.* Cambridge University Press.

Oldenburg, J. and Marquardt, W. (2002). Flatness and higher order differential model representations in dynamic optimization. *Computers & chemical engineering*, 26(3), 385–400.

Sira-Ramirez, H. and Agrawal, S.K. (2004). *Differentially flat systems.* Marcel Decker Inc., New York - Basel.

Sontag, E.D. (2013). *Mathematical control theory: deterministic finite dimensional systems*, volume 6. Springer Science & Business Media.

Wagner, D. and Willhalm, T. (2003). Geometric speed-up techniques for finding shortest paths in large sparse graphs. In *European Symposium on Algorithms Proceedings*, 776–787. LNCS 2832, Springer.

Witsenhausen, H. (1966). A class of hybrid-state continuous-time dynamic systems. *IEEE Transactions on Automatic Control*, 11(2), 161–167.