Optimal Control for Hybrid Systems With Partitioned State Space

Benjamin Passenberg, Peter E. Caines, Marion Leibold, Olaf Stursberg, and Martin Buss

Abstract—**For hybrid systems where the continuous state space is partitioned by switching manifolds, the discrete state changes autonomously if the continuous state hits a manifold. Recently, we introduced a version of the minimum principle for the optimal control of such systems, where necessary optimality conditions were provided for the case that a trajectory passes through the intersection of switching manifolds [1]. Further, we introduced an algorithm based on these conditions for computing optimal controls by varying not only the continuous state on switching manifolds based on gradient information, but also the sequence of discrete states [2]. It was shown that, the combinatorial complexity of former computational schemes based on the minimum principle can be avoided, since not all possible discrete state sequences need to be analyzed separately. In this note, theoretical aspects of the algorithm and practical considerations for a successful implementation are discussed in detail. Further, a comparison to existing algorithms for the solution of hybrid optimal control problems with autonomous switching is presented. The efficiency of the proposed algorithm is demonstrated by a novel numerical example.**

Index Terms—**Hybrid minimum principle, hybrid systems, intersecting switching manifolds, optimal control.**

I. Introduction

Hybrid systems with autonomous switching, as considered in the following, are formed by continuous dynamics that switches autonomously whenever the continuous state trajectory reaches a switching manifold [3]. The determination of optimal controls of such systems is a challenge since not only the optimality principle for classical continuous optimal control needs to be considered but also the combinatorics arising from the set of possible discrete state sequences due to the lack of smoothness of the hybrid state trajectory. A variety of methods for the optimal control of different forms of hybrid systems were proposed [4]–[7], [20].

An important aspect of this challenge is the design of efficient algorithms for solving hybrid optimal control problems (HOCPs). A major difficulty is to find the optimal discrete state sequence simultaneously with the optimal continuous solution. This is due to the combinatorial complexity in the number of possible switches in a discrete state sequence, since a considerable number of algorithms search for an optimal trajectory in every possible discrete state sequence separately [6], [7], [20]. For hybrid systems with controlled switching, efficient algorithms were proposed in [6], [8]–[11]. Controlled switching means the discrete state can be changed by a discrete control whenever desired. On the contrary, only few approaches for reducing the complexity exist for HOCPs with autonomous switching [12]–[15]. Especially algorithms based on the hybrid minimum principle (HMP) [7], [20], which gives necessary conditions for an optimal solution, cannot handle the case of intersecting switching manifolds. This limits the algorithmic search for a locally optimal solution to a differentiable segment of a switching manifold. Therefore, a search over all candidates for a globally optimal solution is combinatorially complex since all possible sequences have to be analyzed separately in the worst case [6], [7], [20].

For hybrid systems with autonomous switching and a partitioned state space, we introduced a version of the HMP that provides optimality conditions for intersections and corners of switching manifolds in addition to the known optimality conditions [1]. Based on this extended version of the HMP, we further proposed an efficient algorithm, which varies the discrete state sequence based on gradient information simultaneously with the continuous optimization [2]. This approach avoids the combinatorial complexity of the algorithms in [6], [7], [20] as well as the problem of not ensuring convergence.

The contribution of this note is the following: After a review of the HOCP (Section II), the algorithm (Section III), and its proof of convergence (Section IV), theoretical and practical aspects like implementation issues, convergence limitations, initialization, and possible extensions are discussed in detail in Section V. In Section VI, a novel numerical example illustrates the efficiency of the algorithm. Section VII compares the algorithm qualitatively with other algorithms for the optimal control of hybrid systems with autonomous switching.

II. Hybrid System

Definition 1: A 6-tuple $\mathbb{H} := \{\mathcal{Q}, \mathcal{X}, \Gamma, U, \mathcal{F}, \mathcal{M}\}$ defines a hybrid system with partitioned state space and autonomous switching on switching manifolds using the following assumptions.

a) $\mathcal{Q} = \{1, 2, \dots, N_q\}$ is the set of N_q discrete states q.

b) $\mathcal{X} = \bigcup_{1 \leq q \leq N_q} \mathcal{X}_q = \mathbb{R}^{n_x}$ is the union of disjoint state regions $\mathcal{X}_q = \partial\mathcal{X}_q \cup \hat{\mathcal{X}}_q \subset \mathbb{R}^{n_x}$ assigned to a discrete state q and the continuous state is $x \in \mathcal{X}$. Boundaries of neighboring state regions have a common set $\partial\mathcal{X}_i \cap \partial\mathcal{X}_k \neq \emptyset$ for $i, k \in \mathcal{Q}, i \neq k$.

c) $U = \{U_q\}_{q \in \mathcal{Q}}$ is the collection of compact sets $U_q \subset \mathbb{R}^{n_u}$ of admissible continuous control values u. $\mathcal{U} = \{\mathcal{U}_q\}_{q \in \mathcal{Q}}$ is the set of all measurable and bounded control trajectories $u : [t_0, t_e] \to U$.

d) $\mathcal{F} = \{f_q\}_{q \in \mathcal{Q}}$ is the collection of continuously differentiable and Lipschitz continuous vector fields $f_q \in C^1(\mathbb{R}^{n_x} \times U_q, \mathbb{R}^{n_x})$ with $q \in \mathcal{Q}$.

e) $\mathcal{M} = \{M_{i,k}\}_{i,k \in \mathcal{Q}, i \neq k}$ is the collection of time-independent switching manifolds $M_{i,k}$ defining the boundary $\partial\mathcal{X}_i \cap \partial\mathcal{X}_k$ between neighboring partitions. An autonomous transition from the discrete state i to k occurs at time t_j for $x(t_j)$ on the manifold $M_{i,k}$. A switching manifold $M_{i,k} = \bigcup_\alpha M_{i,k}^\alpha, \alpha \in \mathbb{N}$, contains submanifolds of $\mathbb{R}^{n_x}$ with co-dimension 1, which are smooth, i.e. C^∞, possibly have a boundary $\partial M_{i,k}^\alpha$, and are expressed by $M_{i,k}^\alpha := \{x | m_{i,k}^\alpha(x) = 0\}$.

f) $\Gamma : \mathcal{Q} \times \mathcal{X} \to \mathcal{Q}$ is the discrete transition map.

□

Definition 2: An execution of a hybrid system is given by $\sigma = (\tau, \varrho, \chi, \upsilon)$, where $\tau = (t_0, \dots, t_{N_\sigma+1} = t_e)$ is a strictly increasing sequence of initial, switching, and final times with N_σ switchings, and $\varrho = (q_0, \dots, q_{N_\sigma})$ denotes a sequence of discrete states. The term $\chi = (\chi_{q_0}, \dots, \chi_{q_{N_\sigma}})$ is a sequence of absolutely (left-)continuous state trajectories $\chi_{q_j} : [t_j, t_{j+1}) \to \mathcal{X}_{q_j}$, which fulfill $x_{q_0}(t_0) =$

Manuscript received October 28, 2011; revised June 21, 2012; accepted January 23, 2013. Date of publication February 07, 2013; date of current version July 19, 2013. Recommended by Associate Editor L. Zaccarian.

B. Passenberg, M. Leibold, and M. Buss are with the Institute of Automatic Control Engineering, Department of Electrical Engineering and Information Technology, Technische Universität München, Munich, 80290, Germany (e-mail: passenberg@tum.de; marion.sobotka@tum.de; mb@tum.de).

P. E. Caines is with the Department of Electrical and Computer Engineering and the Center for Intelligent Machines, McGill University, Montréal, QC, Canada (e-mail: peterc@cim.mcgill.ca).

O. Stursberg is with the Institute of Control and System Theory, Department of Electrical Engineering and Computer Science, University of Kassel, Kassel, 34121, Germany (e-mail: stursberg@uni-kassel.de).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAC.2013.2246112

x_0 and $x_{q_{j-1}}(t_j^-) = x_{q_j}(t_j)$ with $t_j^- := \lim_{t\to t_j, t<t_j} t$ and evolve according to

$$\dot{x}_{q_j} = f_{q_j}\left(x_{q_j}(t), u_{q_j}(t)\right) \tag{1}$$

for a.e. $t \in [t_j, t_{j+1})$ and $j \in \{0, 1, \ldots, N_\sigma\}$. The control $\upsilon = (\upsilon_{q_0}, \ldots, \upsilon_{q_{N_\sigma}})$ is a sequence of control trajectories $\upsilon_{q_j} : [t_j, t_{j+1}) \to U_{q_j}$. ☐

Definition 3: An optimal solution $\sigma^* = (\tau^*, \varrho^*, \chi^*, \upsilon^*)$ locally minimizes the cost functional

$$J = g\left(x_{q_e}(t_e)\right) + \sum_{j=0}^{N_{\sigma^*}} \int_{t_j}^{t_{j+1}} \phi_{q_j}\left(x_{q_j}(t), u_{q_j}(t)\right) dt \tag{2}$$

with $x(t_0) = x_0$ and q_0 specified, $x_{q_e}(t_e)$, q_e, and $N_\sigma < \infty$ free, and $g \in C^1(\mathbb{R}^{n_x}, \mathbb{R})$ and $\phi_q \in C^1(\mathbb{R}^{n_x} \times \mathbb{R}^{n_u}, \mathbb{R}^{n_x})$ for $q \in \mathcal{Q}$. ☐

Additionally, it is assumed that hybrid system executions exist and are unique as well as that an optimal solution exists, see [1], [2] for details. For this class of hybrid systems, the authors introduced a novel version of the HMP, which provides necessary optimality conditions to be satisfied by an optimal execution σ^* [1].

III. Algorithm

This section introduces an algorithm that finds a locally optimal trajectory χ^* based on the optimality conditions of [1, Theorem 1], which is able to deal with a partitioned state space.

The HOCP from Def. 3, written in brief as $\min_{\upsilon\in\mathcal{U}} J(\upsilon)$, is decomposed, such that the algorithm can be set up with two layers. From the decomposition, an HOCP $\min_z J(z)$ is obtained, which is solved for the optimal switching point vector $z^* = (z_1^{*T}, \ldots, z_{N_{\sigma^*}}^{*T})^T$ on the upper layer of the algorithm, where a switching point z_j is the state-time pair $(x^T(t_j)\ t_j)^T$. The optimal switching points z^* satisfy the adjoint transversality and Hamiltonian value conditions provided in [1, Theorem 1]. Between two consecutive switching points z_j and z_{j+1}, only state and control trajectories are allowed, that satisfy the state and adjoint differential equations, the terminal adjoint condition, and the Hamiltonian minimization condition with respect to the continuous control from [1, Theorem 1]. These trajectories are determined in the lower layer.

In every iteration l, Algorithm 1 produces a feasible sequence of discrete states $\varrho^l = (q_0^l, \ldots, q_{N(l)}^l)$ and switching points $z_j^l = (x_j^{lT}\ t_j^l)^T$ for $j = 1, \ldots, N(l)$. Here, feasibility means that a control $\upsilon_{q_j^l} \in \mathcal{U}_{q_j^l}$ for all $j \in \{0, 1, \ldots, N(l)\}$ exists, which transfers system (1) from switching point z_j^l to z_{j+1}^l without leaving the state space $\mathcal{X}_{q_j^l}$.

Algorithm 1:

Step 0 Initialization:[1] Choose a feasible sequence of discrete states $\varrho^0 = (q_0^0, \ldots, q_{N(0)}^0)$ and switching points z^0. Set $l = 0$ and $j_z = 1$.

Step 1: (Lower Layer) Optimal Control of Subproblems: Solve

$$\min_{\upsilon_{q_j^l}\in\mathcal{U}_{q_j^l}} J_{q_j^l} = \min_{\upsilon_{q_j^l}\in\mathcal{U}_{q_j^l}} \int_{t_j^l}^{t_{j+1}^l} \phi_{q_j^l}\left(x_{q_j^l}(t), u_{q_j^l}(t)\right) dt \left(+g\left(x(t_e)\right)\right) \tag{3}$$

for every $j \in \{0, 1, \ldots, N(l)\}$. For the last discrete state $q_{N(l)}$, the terminal cost $g(x(t_e))$ is added. The optimization uses z_j^l as initial value and z_{j+1}^l as bounding condition, and it takes the dynamics (1) and feasible control trajectories $\upsilon_{q_j^l} \in \mathcal{U}_{q_j^l}$ into account. Indirect shooting on first-order optimality conditions is used [6], [16] because it provides highly accurate solutions including state, adjoint, and control trajectories. The accuracy is important for determining the gradients in Step 2.

Step 2a: (Upper Layer) Gradients of Costs on Switching Manifolds: Determine for all z_j^l with $j = \{1, \ldots, N(l)\}$ the gradients of the costs $\nabla_{z_j^l}^{\mathrm{m}} J(z^l) = (\nabla_{x_j^l}^{\mathrm{m}} J(z^l)\ \nabla_{t_j^l}^{\mathrm{m}} J(z^l))$ projected onto the switching manifolds $m_{q_{j-1}^l, q_j^l} = 0$ using the adjoint transversality and Hamiltonian value condition from [1, Theorem 1]

$$\begin{aligned}\nabla_{x_j^l}^{\mathrm{m}} J^T(z^l) = &\sum_{i\in\mathcal{Q}} \nabla_x m_{q_{j-1}^l, i}^T \left(x\left(t_j^l\right)\right) \pi_{j,(i)} + \lambda\left(t_j^l\right) \\ &+ \sum_{i,k\in\mathcal{Q}} \nabla_x m_{i,k}^T \left(x\left(t_j^l\right)\right) \pi_{j,(ik)} - \lambda\left(t_j^{l-}\right)\end{aligned} \tag{4}$$

$$\nabla_{t_j^l}^{\mathrm{m}} J(z^l) = H_{q_{j-1}^l}\left(t_j^{l-}\right) - H_{q_j^l}\left(t_j^l\right) \tag{5}$$

where $M_{q_{j-1}^l, i}$ and $M_{i,k}$ are hit at the j-th switch. If $\nabla_{z_j^l}^{\mathrm{m}} J(z^l) = 0$ $\forall z_j^l$, then stop.

Step 2b: (Upper Layer) Update of Switching Points: Set $j := j_z$, $d_{z_j^l} = -\nabla_{z_j^l}^{\mathrm{m}} J^T(z^l)$, $\alpha = c\beta^p$ with $c \in \mathbb{R}^+$, $\beta \in (0, 1)$, $p = 0$, and $z_j^{l+1} = z_j^l + \alpha d_{z_j^l}$. If the updated switching state x_j^{l+1} is beyond the boundaries of $M_{q_{j-1}^l, q_j^l}$, then a change in the discrete state sequence ϱ^l is required and the following procedure is applied:[2] (i) Search for the discrete state $q \in \mathcal{Q}$, which is a direct neighbor of q_{j-1}^l and satisfies $x_j^{l+1} \in \mathcal{X}_q$. If no such q is found, then repeat Step 2b with $p := p + 1$. (ii) If the found q equals the discrete state q_{j+1}^l of sequence ϱ^l, then set the new sequence ϱ^{l+1} to ϱ^l except for removing the discrete state q_j^l. Otherwise, set the new sequence ϱ^{l+1} to ϱ^l except for including the additional discrete state $q_j^{l+1} := q$ after q_{j-1}^{l+1}. (iii) Project the switching state x_j^{l+1} to the corresponding neighboring switching manifolds, such that $\|z_j^{l+1} - z_j^l\| = \|\alpha d_{z_j^l}\|$ holds. Next, execute Step 1 again with an updated switching point vector z^{l+1} to obtain $J(z^{l+1})$. Define $\Delta J(z^l) := J(z^{l+1}) - J(z^l)$. The update z_j^{l+1} is accepted if the Armijo-like criterion [17]

$$\Delta J(z^l) \leq \begin{cases} -\tau_0 \alpha d_{z_j^l}^T d_{z_j^l} & \text{if } x_j^{l+1} \in M_{q_{j-1}^l, q_j^l} \\ -\tilde{\tau}_0 \alpha d_{\min}^T d_{\min} & \text{if } x_j^{l+1} \notin M_{q_{j-1}^l, q_j^l} \end{cases} \tag{6}$$

is satisfied with $\tau_0 \in (0, 1)$, $\tilde{\tau}_0 = \beta\tau_0$, and

$$d_{\min} = \underset{d\in\left\{d_{z_j^l}, d_{z_j^{l+1}}, d_{z_{j+1}^{l+1}}\right\}}{\arg\min} \left\{\left\|d_{z_j^l}\right\|, \left\|d_{z_j^{l+1}}\right\|, \left\|d_{z_{j+1}^{l+1}}\right\|\right\}. \tag{7}$$

If z_j^{l+1} is rejected, then set $p := p + 1$ and repeat Step 2b. If z_j^{l+1} is accepted, set $l := l + 1$, $p := 0$, and $j_z := j_z + 1$. If $j_z > N(l)$ (the algorithm has just updated the last switching point $z_{N(l)}^l$), then set $j_z := 1$. Goto Step 2a.

IV. Convergence

In the following, the convergence of the algorithm to a locally optimal solution is proven. A more detailed version is provided in [2]. An algorithm is convergent, if it has bounded cost and sufficient descent, see Prop. 1 and Def. 7 for sufficient descent in [2].

[1]Feasible initializations can be found by intuition, by randomized initializations, or by an advanced initialization concept as mentioned in Section V.

[2]For a concise notation, we here restrict the consideration to the case, where the old sequence ϱ^l and the new one ϱ^{l+1} only differ by one discrete state, the switching manifolds are affine and the discrete state spaces $\mathcal{X}_q$ are convex.

The cost J is bounded in Algorithm 1 [2]. If updates z^{l+1} of the switching points z^l leave the sequence of discrete states $\varrho^{l+1} = \varrho^l$ unchanged, sufficient descent in Algorithm 1 can be shown by contradiction, see [2, Prop. 2]. The sufficient descent also holds for updates leading to a change in the discrete state sequence. Therefore, define the switching point $z_{s,j}^l = z_j^l + \beta^{p_s} d_{z_j^l}$ with $p_s \in \mathbb{R}_0^+$ on the intersection of two neighboring switching manifolds, that means $x_{s,j}^l \in \partial M_{q_{j-1}^l, q_j^l} \cap \partial M_{q_{j-1}^{l+1}, q_j^{l+1}}$. Further, a switching point z^l, that remains unchanged except for a change of $\beta^p d_{z_j^l}$ in the j-th subvector, is denoted by $z^l(\beta^p d_{z_j^l})$.

Proposition 1: For iteration k, let $x_i^* \notin M_{q_{j-1}^k, q_j^k}$ $\forall i \in \{1, \ldots, N_{\sigma^*}\}$, $\|z_i^* - z\| > \delta$ $\forall x \in M_{q_{j-1}^k, q_j^k}$, $\delta > 0$, and $x_j^k \in M_{q_{j-1}^k, q_j^k}$. Then there exists an iteration $l \in \mathbb{N}$, $l \geq k$, and a $p \in \mathbb{N}_0$ such that

$$J(z^{l+1}) - J(z^l) \leq -\tilde{\tau}_0 \beta^p d_{\min}^T d_{\min} \tag{8}$$

with $d_{\min}$ from (7), $J(\cdot)$ is bounded, and the updated sequence ϱ^{l+1} is different from ϱ^l. □

The idea of the proof is the following. An update of a switching point leading to a new discrete state sequence is separated into two update steps: The first update step leads to a switching point on the old switching manifold and the second update step to a point on the new switching manifolds. By formulating the difference between the costs of the new and the old discrete state sequence in terms of Armijo steps in both parts of the update step and by combining the results, it can be shown that (8) can always be fulfilled.

Proof: Assume first, that $x_j^k \in \hat{M}_{q_{j-1}^k, q_j^k}$, where $\hat{M}_{q_{j-1}^k, q_j^k}$ is the interior of $M_{q_{j-1}^k, q_j^k}$. By [2, Prop. 2] and $x_j^l \in \hat{M}_{q_{j-1}^k, q_j^k}$ being the result of several update steps starting from x_j^k with $l \geq k$, $\|z_j^l - z_{s,j}^l\|$ can be made arbitrarily small with increasing l. In particular for some l, there exists κ^l, s.t. $\|z_j^l - z_i^*\| > \kappa^l > (1/\beta^2)\|z_j^l - z_{s,j}^l\|$. Now, let $l, p \in \mathbb{N}$, such that

$$x_j^l + \beta^{p+2} d_{x_j^l} \in M_{q_{j-1}^l, q_j^l} \quad x_j^l + \beta^{p+1} d_{x_j^l} \notin M_{q_{j-1}^l, q_j^l} \tag{9}$$

$$J\left(z^l\left(\beta^{p+y} d_{z_j^l}\right)\right) - J(z^l) \leq -\tau_0 \beta^{p+y} d_{z_j^l}^T d_{z_j^l} \tag{10}$$

for every $y \in \{0, 1, 2\}$ and $d_{z_j^l} = -\nabla_{z_j^l}^{\mathrm{m}} J^T(z^l)$, see Fig. 1. Equation (9) implies

$$\beta^{p+2} \left\| d_{z_j^l} \right\| \leq \left\| z_j^l - z_{s,j}^l \right\| \leq \beta^{p+1} \left\| d_{z_j^l} \right\| \tag{11}$$

which leads together with $\kappa^l > (1/\beta^2)\|z_j^l - z_{s,j}^l\|$ to $\beta^p \|d_{z_j^l}\| \leq (1/\beta^2)\|z_j^l - z_{s,j}^l\| < \kappa^l$. The projection $P(z_j^l, \beta^p \|d_{z_j^l}\|)$ delivers the new switching points $z_v \in \{z_j^{l+1}, z_{j+1}^{l+1}\}$, which are used if the Armijo-like test (8) is successful. As the projection maintains the step length

$$\left\| z_v - z_j^l \right\| = \beta^p \left\| d_{z_j^l} \right\| < \kappa^l \tag{12}$$

is true $\forall z_v$. The following inequality holds $\forall z_v$ with $\|z_v - z_j^l\| = \|z_v - z_{s,j}^l + z_{s,j}^l - z_j^l\| \leq \|z_v - z_{s,j}^l\| + \|z_{s,j}^l - z_j^l\|$:

$$\left\| z_v - z_{s,j}^l \right\| \geq \left\| z_v - z_j^l \right\| - \left\| z_{s,j}^l - z_j^l \right\| \overset{(12),(11)}{\geq} \beta^p (1-\beta) \left\| d_{z_j^l} \right\|. \tag{13}$$

Define z_s^l to be the vector of switching points that only differs from z^l in replacing z_j^l by $z_{s,j}^l$. By [2, Prop. 2] it can be concluded that there exists a finite $l \in \mathbb{N}$ leading to a sufficiently small $\beta^p \|d_{z_j^l}\|$, such that $\beta^{\tilde{p}_s} \|d_s\|$ exists to satisfy

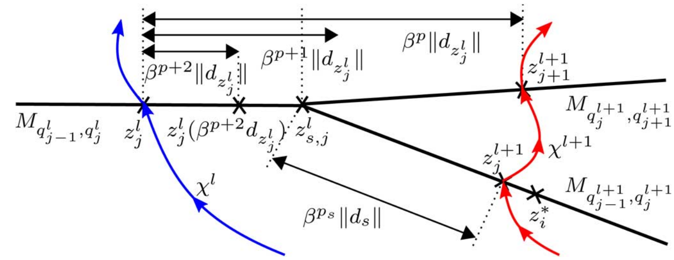

Fig. 1. Switching points used in the proof.

$$J(z^{l+1}) - J\left(z_s^l\right) \leq -\tau_0 \beta^{\tilde{p}_s} d_s^T d_s \tag{14}$$

where d_s and $\beta^{\tilde{p}_s}$ such that $\|d_s\| = \min(\|d_v\|)$, $v \in \{j, j+1\}$, $d_v = -\nabla_{z_v}^{\mathrm{m}} J^T(z_s^l)$, and

$$\left\| z_v - z_{s,j}^l \right\| = \beta^{\tilde{p}_s} \|d_s\|. \tag{15}$$

By the continuity of J and $\|z_v - z_j^l\| < \|z_j^l - z_i^*\|$, it follows for sufficiently small $\beta^p \|d_{z_j^l}\|$ that $d_s^T d_v > 0$. Now, it remains to show that (8) holds

$$\begin{aligned}
\Delta J(z^l) &= J(z^{l+1}) - J\left(z_s^l\right) + J\left(z_s^l\right) - J(z^l) \\
&\leq J(z^{l+1}) - J\left(z_s^l\right) + J\left(z^l\left(\beta^{p+2} d_{z_j^l}\right)\right) - J(z^l) \\
&\overset{(14),(10)}{\leq} -\tau_0 \beta^{p_s} d_s^T d_s - \tau_0 \beta^{p+2} d_{z_j^l}^T d_{z_j^l} \\
&\overset{(15),(13)}{\leq} -\tau_0 \beta^p \left\| d_{z_j^l} \right\| \left[(1-\beta)\|d_s\| + \beta^2 \left\| d_{z_j^l} \right\| \right] \\
&\leq -\tau_0 \beta^p \left\| d_{z_j^l} \right\| \left[\beta(1-\beta)\|d_s\| + \beta^2 \left\| d_{z_j^l} \right\| \right] \\
&\leq -\tau_0 \beta \beta^p \left\| d_{z_j^l} \right\| \|d_{\min}\| \leq -\tilde{\tau}_0 \beta^p d_{\min}^T d_{\min}.
\end{aligned}$$

Likewise, the same result can be obtained, if $x_j^l \in \partial M_{q_{j-1}^l, q_j^l} \cap \partial M_{q_{j-1}^{l+1}, q_j^{l+1}}$ with $q_j^l \neq q_j^{l+1}$ or q_j^l vanishes, i.e. $q_j^{l+1} = q_{j+1}^l$. ■

Combining the results above, the sufficient descent property follows, which proves the convergence of Algorithm 1.

Proposition 2: For every $l \in \mathbb{N}$ and $\kappa^l > 0$ there exists $\eta > 0$, such that, if $\|z_j^l - z_i^*\| > \kappa^l$ for $j \in \{1, \ldots, N(l)\}$ and corresponding $i \in \{1, \ldots, N_{\sigma^*}\}$, then

$$J(z^{l+1}) - J(z^l) \leq -\eta. \tag{16}$$

□

Proof: If $\|z_j^l - z_i^*\| > \kappa^l$ for $j \in \{1, \ldots, N(l)\}$ and corresponding $i \in \{1, \ldots, N_{\sigma^*}\}$, then there exists $\epsilon > 0$ such that $\|\nabla_{z_j^l}^{\mathrm{m}} J(z^l)\| = \|d_{z_j^l}\| > \epsilon$. Prop. 2 in [2] and Prop. 1 imply that there always exists a finite $p \in \mathbb{N}_0$, such that a new switching point z^{l+1} can be determined with $\|z_v - z_j^l\| = \beta^p \|\nabla_{z_j^l}^{\mathrm{m}} J(z^l)\|$ $\forall z_v \in \{z_j^{l+1}, \ldots, z_{j+\varpi}^{l+1}\}$. Here, z_v is any updated switching point and x_v may also be on $M_{q_{j-1}^l, q_j^l}$. If the update is across an intersection of switching manifolds and the algorithm does not reach the optimal switching point z_i^* exactly with this update, then there also exist $\epsilon_v > 0$, such that $\|\nabla_{z_v}^{\mathrm{m}} J(z^{l+1})\| = \|d_v\| > \epsilon_v$ holds $\forall z_v$. Let $\epsilon_{\min}$

be $\min\{\epsilon, \epsilon_v\}$ $\forall \epsilon_v$ and recall $\|d_{\min}\| = \min\{\|d_{z_j^l}\|, \|d_v\|\}$. In each iteration l, the sufficient descent condition

$$J(z^{l+1}) - J(z^l) \leq -\tilde{\tau}_0 \beta^p \epsilon_{\min}^2 = -\eta \tag{17}$$

holds, since either the descent property from Prop. 2 in [2] or (8) is satisfied with $\tilde{\tau}_0 < \tau_0$ and $\|d_{\min}\| > \epsilon_{\min}$. ■

V. Discussion of the Algorithm

For hybrid systems with partitioned state space, Algorithm 1 varies the discrete state sequence while searching for optimal hybrid trajectories. The algorithm uses an extended version of the HMP presented in [1] that accounts for corners and intersections of switching manifolds. Consequently, gradients of the cost can be calculated at those corners or intersections and the gradient information is exploited in the algorithm. Therefore, the algorithm avoids the problem of former algorithms based on the HMP that the computational complexity increases combinatorially with the number of switchings. In the current formulation, the algorithm finds locally optimal controls and discrete state sequences.

The gradient of the costs, which is applied for shifting a switching state, is provided by the adjoint transversality condition for autonomous switching from Theorem 1 in [1]. The adjoint transversality condition can be evaluated directly using the adjoints obtained by the solution of the underlying, purely continuous optimal control problems specified in Step 1 of Algorithm 1. The associated Lagrange multipliers π in (4) are calculated numerically by a linear regression, which is scalar if the switching point is not on an intersection of switching manifolds. The optimality condition contains several normals of switching manifolds if the continuous state trajectory meets an intersection of switching manifolds. To obtain unique results, only a subset of linearly independent normals is used in the algorithm from the entire set of possibly linearly dependent normals that appear in the adjoint transversality condition. The linear regression ensures that the error in the adjoint transversality condition is minimized and consequently that the gradients are projected onto the switching manifolds. With the adjoint transversality condition, the gradients of switching points can be calculated inexpensively and with high accuracy.

On the lower layer of Algorithm 1, two-point BVPs are solved with indirect multiple shooting as follows: The state and adjoint dynamics are integrated until the state trajectory hits a switching manifold and not until a desired switching time is reached. This implies that all switching times t_j^l for $j \in \{1, \ldots, N(l)\}$ are determined indirectly by the choice of x_j^l and not anymore directly by Armijo update steps of t_j^l. It is crucial for the success of the algorithm that the target switching state x_{j+1}^l is reachable from the initial switching state x_j^l. Leaving the switching time t_{j+1}^l unspecified during the forward integration stabilizes the numerical optimization since the chances to reach x_{j+1}^l are increased. In our experience, the Armijo update steps for the switching times t_j^l are very sensitive to small changes in the switching times, in contrast to the updates of the switching states x_j^l.

The stopping condition for the integration implies that each subproblem (3) is one with free terminal time t_{j+1}^l for $j \in \{0, \ldots, N(l) - 1\}$. Further, an optimal solution of the entire HOCP requires that the Hamiltonians just before and just after the autonomous switching are equal

$$\begin{aligned} &H_{q_j^l}\left(x\left(t_{j+1}^{l-}\right), \lambda\left(t_{j+1}^{l-}\right), u_{q_j^l}\left(t_{j+1}^{l-}\right)\right) \\ &\qquad = H_{q_{j+1}^l}\left(x\left(t_{j+1}^{l}\right), \lambda\left(t_{j+1}^{l}\right), u_{q_{j+1}^l}\left(t_{j+1}^{l}\right)\right) \end{aligned}$$

if the continuous state trajectory avoids intersections of switching manifolds. The optimal solutions of the subproblems have to consider the Hamiltonian continuity condition in this case. To be able to solve the optimal control subproblems separately, the following observation is important: The optimal Hamiltonian is constant for almost every time, if the optimal trajectory does not pass through intersections of switching manifolds. This can be verified by analyzing its non-differentiable points and its derivative

$$\begin{aligned} \frac{d}{dt} H_{q_j^*} = \nabla_t H_{q_j^*} + \nabla_x H_{q_j^*} \dot{x}^*(t) + \\ \nabla_\lambda H_{q_j^*} \dot{\lambda}^*(t) + \nabla_u H_{q_j^*} \dot{u}_{q_j^*}^*(t) = 0. \end{aligned}$$

The first term $\nabla_t H_{q_j^*}$ is zero since the Hamiltonian does not explicitly depend on time. The two terms $\nabla_x H_{q_j^*} \dot{x}^*(t)$ and $\nabla_\lambda H_{q_j^*} \dot{\lambda}^*(t)$ cancel each other, compare the state and adjoint differential equations and the definition of the Hamiltonian in [1]. In the last term, either the components of $\nabla_u H_{q_j^*}$ are zero when the corresponding components of the optimal control $u_{q_j^*}^*(t)$ are within their bounds in $U_{q_j^*}$ or the components of $\dot{u}_{q_j^*}^*(t)$ are zero when the corresponding components of $u_{q_j^*}^*(t)$ are on the boundary $\partial U_{q_j^*}$. At jumps of the optimal control and at autonomous switchings, the Hamiltonian is continuous. (Compare the definition of the Hamiltonian, assumptions on the hybrid system, and the Hamiltonian value and minimization condition from [1, Theorem 1].)

If the final time t_e of the entire HOCP is unspecified, there are no constraints on the terminal times t_{j+1}^l of the different subproblems. To find optimal terminal times t_{j+1}^l, the constant Hamiltonian values have to be zero at the terminal times: $H_{q_j^l}(x(t_{j+1}^{l-}), \lambda(t_{j+1}^{l-}), u_{q_j^l}(t_{j+1}^{l-})) = 0$ for $j \in \{1, \ldots, N(l)+1\}$. This condition is added to the indirect multiple shooting algorithm.

If the final time t_e is specified, then the following problem arises: On the one hand, the optimal control subproblems have to be solved one after another starting with the first subproblem from t_0. Otherwise, it remains unknown how much time can be spent in the last subproblem. On the other hand, the optimal control subproblems need to be solved in backward order to determine the unknown Hamiltonians $H_{q_{j+1}^l}(x(t_{j+1}^l), \lambda(t_{j+1}^l), u_{q_{j+1}^l}(t_{j+1}^l))$ just after an autonomous switching. The conflict is solved by using the Hamiltonians $H_{q_{j+1}^{l-1}}(x(t_{j+1}^{l-1}), \lambda(t_{j+1}^{l-1}), u_{q_{j+1}^{l-1}}(t_{j+1}^{l-1}))$ just after an autonomous switching from the previous iteration $l-1$ as target values for the Hamiltonians $H_{q_j^l}(x(t_{j+1}^{l-}), \lambda(t_{j+1}^{l-}), u_{q_j^l}(t_{j+1}^{l-}))$ in the current iteration l.

Indirect multiple shooting methods [16] provide solutions with almost arbitrary accuracy. Other optimization approaches like indirect collocation, direct multiple shooting, direct collocation, and approximate dynamic programming do not reach a comparable accuracy for the same number of iterations. However, indirect multiple shooting has a small domain of convergence compared to dynamic programming and direct optimization approaches. The reason is a high sensitivity of the optimization since the state or adjoint differential equations need to be integrated in their instable direction. The initialization may be difficult since the adjoints are not physically intuitive and the domain of convergence is small. For the numerical example in Section VI, an initialization of the adjoints with zeros is sufficient for convergence. For larger, nonlinear examples, finding a feasible initialization may become a challenge. In this case, an initialization concept is applied, where the optimal control subproblems are solved approximately by a direct method first, which is easier to initialize. In a second step, the solution is then used to initialize the indirect multiple shooting method. We proposed this initialization concept in [18].

On the upper layer, the gradient descent based optimization of the switching points and the discrete state sequence works reliable. If the current solution is not in the neighborhood of a locally optimal solution,

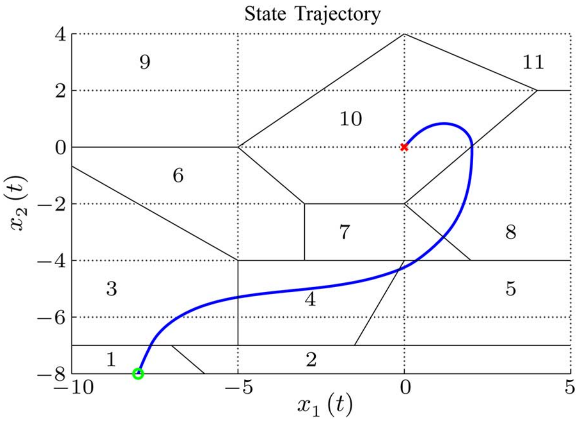

Fig. 2. Optimal discrete state sequence and state trajectory: The initial and terminal conditions are $x(t_0) = (-8, -8, \pi/2)^T$, $q_0 = 1$, $t_0 = 0$, and $x_1(t_e) = x_2(t_e) = 0$ with $x_3(t_e)$ unspecified and $t_e = 2$.

the upper layer converges fast towards a solution. Here, the step size is usually constrained by the convergence properties of the lower layer. In the neighborhood of an optimal solution, the convergence rate of the gradient based optimization is low. In this phase of the optimization, it seems to be advantageous to switch to a Newton based update scheme for the switching points in future.

For slightly nonlinear optimal control subproblems like the numerical example in Section VI, the indirect multiple shooting method on the lower layer is reliable. For more complex subproblems with, e.g., strongly nonlinear switching manifolds, the convergence of the indirect multiple shooting is difficult to achieve and sometimes only possible for very small update steps of switching points.

VI. Numerical Example

In the considered example, the partitioned state space contains 11 discrete states. A discrete state $q \in \{1, \ldots, 11\}$ is separated from each of its direct neighbors k by a switching manifold $m_{q,k} = m_1 x_1 + m_2 x_2 + m_0$. In the example, the task is to control a unicycle

$$\dot{x}_1 = v_q \cos x_3, \quad \dot{x}_2 = v_q \sin x_3, \quad \dot{x}_3 = u_q \tag{18}$$

with coordinates x_1, x_2 in the plane, and x_3 as the orientation, $u_q \in U_q$ is the steering input, and v_q is the velocity. Each partition $\mathcal{X}_q$ simulates a different surface, which causes the unicycle to drive with a fixed velocity v_q in that partition. The cost function is

$$\phi_q(x(t), u(t)) = \frac{1}{2}\left(x^T(t) S_q x(t) + R_q u_q^2(t)\right) + T_q x(t). \tag{19}$$

The algorithm is initialized with the discrete state sequence $\varrho_0 = (1,3,4,6,7,10)$, which is the straight connection of q_0 and a likely q_e. The sequence ϱ_0 is varied over $\varrho = (1,3,4,7,10)$ and $\varrho = (1,3,4,7,8,10)$ to the optimal sequence $\varrho^* = (1,3,4,5,7,8,10)$; see Fig. 2. The optimal costs are $J = 16.67$ and the computations took 2.7 h. The computations were performed on a 1.6-GHz processor using Matlab 2009a. The optimal solution and the computations are compared to results of an implementation based on indirect multiple shooting [6]. That algorithm analyzes every of the 3976 possible discrete state sequences in the partitioned state space separately with at most six switchings beginning from q_0. The optimum obtained by the brute-force search is the same, but the solution took 348.7 h.

VII. Comparison and Conclusions

Further existing algorithms for the indirect solution of HOCPs with autonomous switching are a multiple phase multiple shooting method [6] and an indirect collocation method [7], [20]. For optimizing the discrete state sequence, these approaches have to analyze every possible discrete state sequence, which has a combinatorial computational complexity with respect to the number of autonomous switches. In [13], this complexity can be reduced by introducing a branch-and-bound method based on a relaxation of autonomous switching to controlled switching. Compared to the approaches [6], [7], [20], the complexity of Algorithm 1 is considerably lower as no combinatorial factor appears. In contrast to Algorithm 1, the mentioned approaches rely on a version of the HMP, which is not able to deal with corners or intersections of switching manifolds. This means that the convergence to a locally optimal solution cannot be ensured if an optimal switching point is on or in the direct neighborhood of an intersection or corner of a switching manifold due to incorrect gradient information in this case.

Dynamic programming (DP) is a further class of algorithms that can solve HOCPs [3], [15]. In general, the state and control space is discretized in DP and the value function, which provides the minimal costs from any point in the state space to the goal, is numerically approximated via function approximations. We recall that the key property of DP is that it delivers globally optimal solutions and that it only has a linear increase in the computational complexity with respect to the number of discrete states. A major drawback of DP is the curse of dimensionality, which means that the solution complexity grows exponentially with the dimension of the continuous state and control spaces. Due to the discretization, the accuracy of solutions from DP does not reach the accuracy that Algorithm 1 based on indirect multiple shooting can provide. Further, the convergence of DP for the case of intersecting switching manifolds has not been shown yet.

Mixed-integer programming (MIP) belongs to direct methods and is often used for the solution of HOCPs with autonomous switching [4], [14]. In the approach, the HOCP is discretized in time and integer (or boolean) variables are introduced to account for the discrete dynamics. The resulting optimization problem is solved, e.g., by sequential quadratic programming (SQP), usually in combination with branch-and-bound methods to find the optimal discrete state sequence. An advantage of MIP is that the domain of convergence is in general larger compared to indirect methods while finding locally optimal solutions. This simplifies the initialization together with the observation that physically intuitive state and control trajectories have to be guessed initially instead of state and adjoint variables. When introducing boundaries for switching manifolds, MIP is able to handle intersections of switching manifolds as Algorithm 1. A disadvantage is that the accuracy of solutions is in general lower than for Algorithm 1 for the same number of iterations due to the discretization of the HOCP. The major disadvantage of MIP in comparison to Algorithm 1 is that the computational complexity to find an optimal discrete state sequence is combinatorial in the number of discrete states. Further, the combinatorial complexity grows with the number of time steps in [4] instead of the number of switches in [6], [7], [20]. An approach for reducing the complexity of branch-and-bound methods is given in [12], where numerical techniques for solving continuous optimization problems are combined with symbolic techniques for solving constraint satisfaction problems of logic variables.

Another alternative (also a direct method) is based on an SQP algorithm for non-hybrid optimal control problems [19]. The algorithm allows to find a locally optimal solution including a locally optimal discrete state sequence with no increase in the computational complexity compared to purely continuous optimal control problems. That approach has the key drawback in the presented form that the convergence cannot be proven since the discontinuity of vector fields is not rigorously considered. This can cause jumps in the costs and jumps in the errors in the constraints during the optimization. Consequently, the SQP solver may converge to such a jump though the corresponding solution is actually not a locally optimal solution of the HOCP with autonomous switching.

In summary, the method considered in this technical note appears as a suitable compromise between complexity, convergence, and accuracy of the solution.

REFERENCES

[1] B. Passenberg, P. Caines, M. Sobotka, O. Stursberg, and M. Buss, "The minimum principle for hybrid systems with partitioned state space and unspecified discrete state sequence," in *Proc. 49th IEEE Conf. Decision and Control*, 2010, pp. 6666–6673.

[2] B. Passenberg, M. Sobotka, O. Stursberg, M. Buss, and P. Caines, "An algorithm for discrete state sequence and trajectory optimization for hybrid systems with partitioned state space," in *Proc. 49th IEEE Conf. Decision and Control*, 2010, pp. 4223–4229.

[3] M. Branicky, V. Borkar, and S. Mitter, "A unified framework for hybrid control: Model and optimal control theory," *IEEE Trans. Autom. Control*, vol. 43, no. 1, pp. 31–45, Jan. 1998.

[4] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, pp. 407–427, 1999.

[5] S. Hedlund and A. Rantzer, "Optimal control of hybrid systems," in *Proc. 38th Conf. Decision and Control*, 1999, pp. 3972–3977.

[6] P. Riedinger, J. Daafouz, and C. Iung, "About solving hybrid optimal control problems," in *Proc. 17th IMACS World Congr.*, 2005.

[7] M. Shaikh and P. Caines, "On the hybrid optimal control problem: Theory and algorithms," *IEEE Trans. Autom. Control*, vol. 52, no. 9, pp. 1587–1603, Sep. 2007.

[8] P. Caines and M. Shaikh, "Optimality zone algorithms for hybrid systems: Efficient algorithms for optimal location and control computation," in *Hybrid Systems: Computation and Control*, ser. LNCS 3927. Berlin, Germany: Springer, 2006, pp. 123–137.

[9] S. Bengea and R. DeCarlo, "Optimal control of switching systems," *Automatica*, vol. 41, no. 1, pp. 11–27, 2005.

[10] M. Egerstedt, Y. Wardi, and H. Axelsson, "Transition-time optimization for switched-mode dynamical systems," *IEEE Trans. Autom. Control*, vol. 51, no. 1, pp. 110–115, Jan. 2006.

[11] S. Sager, "Numerical Methods for Mixed-Integer Optimal Control Problems," Ph.D. dissertation, Universität Heidelberg, Heidelberg, Germany, 2006.

[12] A. Bemporad and N. Giorgetti, "Logic-based solution methods for optimal control of hybrid systems," *IEEE Trans. Autom. Control*, vol. 51, no. 6, pp. 963–976, Jun. 2006.

[13] B. Passenberg and O. Stursberg, "Graph search for optimizing the discrete location sequence in hybrid optimal control," in *Proc. 3rd IFAC Conf. Analysis and Design of Hybrid Systems*, 2009, pp. 304–309.

[14] O. Stursberg and S. Panek, "Control of switched hybrid systems based on disjunctive formulations," in *Hybrid Systems: Computation and Control*, ser. LNCS 2289. New York: Springer, 2002, pp. 421–435.

[15] M. Rungger and O. Stursberg, "Optimal control for deterministic hybrid systems using dynamic programming," in *Proc. 3rd IFAC Conf. on Analysis and Design of Hybrid Systems*, 2009, pp. 316–321.

[16] H. Bock, "Numerical solution of nonlinear multipoint boundary value problems with applications to optimal control," *ZAMM*, vol. 58, pp. 407–409, 1978.

[17] L. Armijo, "Minimum of functions having lipschitz-continuous first partial derivatives," *Pacific J. of Mathemat.*, vol. 16, pp. 1–3, 1966.

[18] B. Passenberg, M. Kroeninger, G. Schnattinger, M. Leibold, O. Stursberg, and M. Buss, "Initialization concepts for optimal control of hybrid systems," in *Proc. 18th IFAC World Congr.*, 2011, pp. 10 274–10 280.

[19] S. Wei, K. Uthaichana, M. Zefran, R. DeCarlo, and S. Bengea, "Applications of numerical optimal control to nonlinear hybrid systems," *Nonlinear Anal.: Hybrid Syst.*, vol. 1, no. 2, pp. 264–279, 2007.

[20] M. Shaikh and P. Caines, "On the hybrid optimal control problem: Theory and algorithms," *Corrigendum*, vol. 54, no. 6, p. 1440, 2009.

Computational Methods for Distributed Control of Heterogeneous Cyclic Interconnection Structures

Justin K. Rice and Jan-Willem van Wingerden

***Abstract*—We address the problem of fast system analysis and controller synthesis for heterogeneous subsystems interconnected in a loop. Such distributed systems have state-space realizations with a special matrix structure, which we show to be a generalization of the Sequentially Semi-Separable (SSS) structure employed in recent research on systems interconnected in a Cartesian array. By extending the $\mathcal{O}(N)$ structure-preserving SSS arithmetic to the matrices induced by this circular type of interconnection, we introduce a new cyclic SSS matrix structure and arithmetic which leads to fast and efficient procedures for linear computational complexity optimal distributed controller synthesis for arbitrarily heterogeneous subsystems connected in a loop. In the homogeneous case, where all subsystems are identical, the computational complexity reduces to $\mathcal{O}(1)$ and an interesting relationship with the infinite case is demonstrated. CSSS matrices can be non-Toeplitz (as compared to circulant matrices), a great expansion of domain, but when restricted to be Toeplitz (and thus circulant), the arithmetic reduces to $\mathcal{O}(1)$, a huge cost-savings. The procedures are demonstrated on two computational examples, using a freely available MATLAB toolbox implementation of these algorithms.**

***Index Terms*—Computational methods, cyclic interconnection structures, large scale systems, network analysis and control.**

I. INTRODUCTION

Many areas of control, e.g., distributed control, periodic systems, etc. use circulant matrices in their state space realizations (see, e.g. [1]–[6]). This is advantageous because circulant matrices are closed under addition, multiplication, and inversion, and all of these arithmetic operations can be conducted in $\mathcal{O}(N\log(N))$ or less. The disadvantages are of course that circulant matrices are by definition Toeplitz—so systems must be invariant in space or invariant in time to be modeled by circulant matrices, a major limitation. Much other work has been devoted to more general "Cyclic Interconnection Structures" with heterogeneous subsystems [7]–[9], and "Cyclic Negative Feedback Systems" [10]–[12], but the computational results have mostly been limited to special cases of low-order or limited inter-subsystem communication.

Recently, special structure exploiting methods have arisen for efficient analysis and controller computation for some distributed systems, i.e., the LMI [13], $\mathcal{H}$-matrix [14], Fourier transform [15], distributed negotiation [16], and SSS-matrix techniques of [17]. In [17] and associated papers it is shown that heterogeneous distributed systems interconnected on a line, as in [13], induce a special matrix structure in the lifted system, called 'Sequentially Semi-Separable' (SSS). This structure can then be exploited in $\mathcal{O}(N)$ computational complexity algorithms for nonconservative stability and performance

Manuscript received July 09, 2012; revised December 11, 2012; accepted February 08, 2013. Date of publication February 18, 2013; date of current version July 19, 2013. This work was supported by the NWO Veni Grant 11930 "Reconfigurable Floating Wind Farms". Recommended by Associate Editor G. E. Dullerud.

The authors are with the Delft Center for Systems and Control, Delft University, 2628CD Delft, The Netherlands (e-mail: JustinKRice@gmail.com; J.W.vanWingerden@tudelft.nl).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAC.2013.2247699