DIGITAL IDLE SPEED CONTROL OF AUTOMOTIVE ENGINES USING HYBRID MODELS [1]

Elena De Santis, Maria Domenica Di Benedetto, Giovanni Girasole

Center of Excellence DEWS
Department of Electrical Engineering
University of L'Aquila
Monteluco di Roio, 67040 L'Aquila, Italy.
{desantis, dibenede, girasole}@ing.univaq.it

Abstract: Idle speed control for an automotive engine is formulated as the problem of computing a maximal safe set for a hybrid system modeling an SI engine in idle mode. Since this problem is computationally intractable, we exploit the relations between safe sets for a continuous-time switching system S and an appropriate discrete-time switching system associated with S to reduce drastically its complexity. An algorithm in the discrete-time domain is proposed for the determination of the maximal safe set. The methodology is general enough to be easily extended to different hybrid control problems. In particular, we solved the problem of the computation of the maximal safe set in the case of un-synchronized switching and sampling times for idle control, an open problem for quite some time. Simulation results show the efficiency of the proposed approach.

Keywords: Engine Control, Hybrid systems, Safe Set Computation

1. INTRODUCTION

Hybrid models for automotive engine control have been proposed in the recent past, e.g., De Santis *et al.* [2004]. These models are more accurate than the standard average models that do not capture the transients in the dynamics of the engine. The general hybrid model for the engine control problem can be simplified considerably considering the particular region of operation we are interested in. A hybrid model for an engine specialized for the idle mode is presented in detail first. The model is used to find a controller that maintains engine speed in a given range; this constraint on engine speed is re-formulated in terms of *safety specification*, allowing the use of algorithms, based on discretization, for finding the safe set and the use of techniques for controller selection starting from the safe set computation. In particular, we make use of the results in De Santis *et al.* [2004] where we showed that the maximal controlled invariant set for a continuous-time switching system can be arbitrarily closely approximated by an appropriate controlled invariant set for the discretized switching system. The piecewise constant control, obtained from the control law that makes the set invariant for the discrete-time switching system by holding the value at sampling times over the sampling period, makes the set safe for the continuous-time switching system.

[1] This work has been partially supported by European Commission under Project HYBRIDGE IST–2001–32460 and IST NoE HyCON contract n. 511368.

With respect to previous work in this area, our procedure does not assume that sampling times and switching times are synchronized. This is an important technical aspect in general, and in particular for our application, where there is no reason why sampling and switching caused by the piston reaching a dead center [2] should be synchronized. Further, it allows the extension of the results obtained in [Balluchi *et al.*, 2002] where the error due to the lack of synchronization was not taken into account.

The paper is organized as follows. In Section 2, we describe the hybrid model of the engine in idle mode. In Section 3, we discretize the hybrid model introducing minimum and maximum dwell time and the errors due to unsynchronized sampling and switching times. In Section 4, we solve the problem of finding the maximal safe set for the sampled model. Finally, in Section 5, the controller derived using the sampled-time system is applied to the original continuous-time system. Simulation results are presented to demonstrate the efficiency of the method.

2. IDLE MODE ENGINE MODEL

In deriving a model for the engine control problem of interest, we exploit the peculiarities of the idle region of operation to avoid including complexity in the model that is not needed to solve our control problem. The engine is said to be in idle mode if the accelerator pedal is released and no gear is inserted. In this operation region, the car is not moving but the engine should stay "alive". The interesting aspect of this problem is that the revolutions of the engine should stay nearly constant no matter which load is applied to it. It is in general difficult, if not impossible, forecasting when loads such as air conditioning are applied. The control objective is to *maintain crankshaft engine speed (n) limited in a range, given in terms of nominal speed (n_0) and maximum absolute tolerance (Δ_n):*

$$n \in [n_0 - \Delta_n, n_0 + \Delta_n]$$

2.1 Continuous dynamics in idle-mode

In the idle speed control problem two dynamics are of interest: the intake manifold pressure and the crankshaft dynamics. The manifold pressure p is regulated by the throttle opening angle α:

$$\dot{p}(t) = a_p p(t) + b_p \alpha(t) \qquad (1)$$

[2] A dead center is the highest (TDC: Top Dead Center) or lowest (BDC: Bottom Dead Center) position reached by the piston in the cylinder.

The control variable $\alpha(t)$ is limited to a given interval, $\alpha(t) \in [0 \quad \alpha_{max}]$, in order to avoid manifold pressure to raise too much and to limit control range for safety reasons.

Crankshaft variables of interest are the angular position θ_C, expressed in degrees [°], and the revolution speed n, expressed in RPM (Revolutions Per Minute); crankshaft angle θ_C evolves according to the following:

$$\dot{\theta}_C(t) = K_C n(t) \qquad (2)$$

where K_C is the factor that transforms RPM in $[°]/[s]$. Crankshaft speed evolves with the dynamics:

$$\dot{n}(t) = a_n n(t) + b_n(T(t) - T_l(t)) \qquad (3)$$

where T is the engine generated torque, $T_l \in [0, T_{lmax}]$ is the disturbance torque modeling the effect of loads to the engine coming from subsystems that take energy from the engine, $a_n = \frac{-30B}{\pi J_{eq}}$ and $b_n = \frac{30}{\pi J_{eq}}$, with B viscous friction coefficient and J_{eq} driveline momentum of inertia.

2.2 Torque generation

In a four-stroke gasoline engine, torque is generated by a a piston when it reaches the highest position in the cylinder and the air-fuel mix entrapped is ignited. In the model, torque is assumed constant during the entire expansion stroke. The torque generation mechanism and the stroke evolution [3] are represented by the FSM in Figure 1; each transition occurs when the piston reaches one of the dead centers.

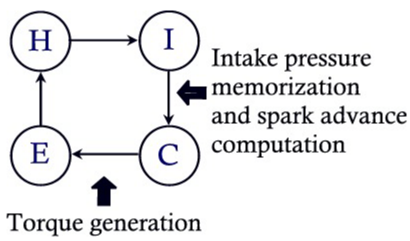

Fig. 1. Single cylinder evolution

Engine torque is expressed either with complex polynomials or look-up tables that cover almost each engine speed and manifold pressure range. In our application, since engine speed is limited in a range, and consequently we have limited torque range, we can simplify substantially the model:

$$T(t_{C-E}) = c_1 p(t_{I-C}) + c_2 \theta_s(t_{I-C}) + c_3 \qquad (4)$$

where $p(t_{I-C})$ and $\theta_s(t_{I-C})$ are respectively intake manifold pressure and spark advance at the end of intake stroke, corresponding to a bottom dead center. We consider the spark advance angle θ_s as the deviation from optimal spark advance,

[3] We are dealing with 4-strokes SI engines.

given as a function of the engine working point. The spark advance angle is bounded to avoid knock (too much advance) and misfire (too little advance). For example, $\theta_s = 0$ means that spark coils are programmed to provide the spark at the angular position corresponding to the optimal spark advance.

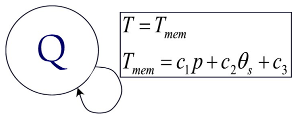

Fig. 2. Torque generation model

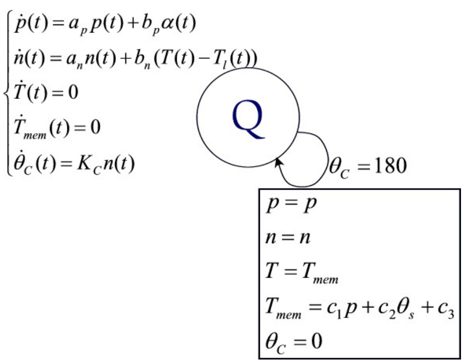

Fig. 3. Hybrid engine model

In a 4-cylinders 4-strokes engine only one cylinder can be in any one stroke, so only one cylinder is producing torque. Hence, torque is generated every 180° of crankshaft angular position and the FSM of Figure 1 reduces to the one shown in Figure 2, where the transition is taken when a piston reaches the TDC_{C-E}, (that will be referred to in the sequel simply as TDC).

2.3 Hybrid model

The models presented in Subsections 2.1 and 2.2 are heterogeneous; pressure and angular speed follow continuous time dynamics, while torque generation is *event-driven*, because of torque value reset every TDC. These models merge in a single hybrid model, shown in Figure 3. The transition occurs when crankshaft angle reaches 180°, and the reset is performed; it is important to execute reset assignments in the sequence, in order to consider compression and spark programming delay.

3. SAMPLED DATA MODEL

All control strategies are implemented in an Electronic Control Unit (ECU), which is a discrete time system being based on one or more microprocessors. Hence, the first step in controller design is sampling the original continuous time dynamics. Sampling is intrinsecally periodic and is dictated by the clock of the ECU. The discrete time dynamics of the hybrid model is due to switching caused by a piston reaching a TDC. Since TDC depends on the speed of the engine, it is not periodic and there is no synchronization possible between switching and sampling.

3.1 Dwell-time and synchronization error

We solve the non-synchronization problem introducing the *minimum and maximum dwell time.* After one transition, the system remains in the state at least for the minimum dwell time and at most for the maximum dwell time. In a sampled time framework, dwell time corresponds to the minimum and maximum number of samples during which the system stays in a given discrete state. Given the constraints on engine speed in idle mode, the distance in time between TDCs is limited to an interval. In fact, given the engine speed specification $n \in [n_{min}, n_{max}]$, it is possible to find the range for a TDC period: $TDC_{min} = \frac{180}{K_C n_{max}}$ and $TDC_{max} = \frac{180}{K_C n_{min}}$; the minimum number of samples taken while the system stays in a state after a transition is given by $N_1 = \rfloor(TDC_{min}/t_{sampling})$, while the maximum number of samples is given by $N_2 = \rceil(TDC_{max}/t_{sampling})$.

Non-synchronization effects are described by two physical phenomena that are taken into account as errors in the expressions of the reset.

Manifold pressure reading error. If the reset occurs exactly at a sampling instant, the pressure read is the manifold pressure at the end of intake stroke used in (4). This is not true in general, so the reading error must be estimated. By integrating the continuous pressure dynamics (1) and considering that the control value is constant during intersampling periods, we obtain: $p(t) = e^{a_p t}\hat{p} + \frac{b_p \alpha}{a}(e^{a_p t} - 1)$. The error is given by $\Delta p = p(t) - \hat{p} = (e^{a_p t} - 1)(\hat{p} + \frac{b_p \alpha}{a_p})$, and since $a_p < 0$ and $b_p > 0$, the maximum absolute value is given by

$$\Delta p_{max} = |(e^{a_p t_{sampling}} - 1)|(max|(p + \frac{b_p \alpha}{a_p})|)$$

where max computation is performed over the set given by $p \in [p_{min}, p_{max}]$ and $\alpha \in [\alpha_{min}, \alpha_{max}]$. Let $I_p = [-\Delta p_{max}, \Delta p_{max}]$ be the admissible value set for Δp.

Engine torque reset. There is no reason to believe that the reset occurs exactly at a sampling instant. If reset does not occur at a sampling instant, then torque is not detected for a sample period, i.e. the torque value changes but the controller cannot detect this change for an entire sample period. Considering the torque difference ΔT we can compute the engine speed deviation, treating ΔT as a disturbance in the discrete time dynamics, $n^+ = a_{nd} n + b_{nd}(T - T_l - \Delta T)$, where $\Delta T \in [-\Delta T_{max}, \Delta T_{max}]$. The engine speed disturbance Δn takes values in $I_n = [-b_{nd}\Delta T_{max}, b_{nd}\Delta T_{max}]$.

The two errors described above are taken into account in the expressions of the reset as:

$$n = n + \Delta n;$$
$$T_{mem} = c_1(p + \Delta p) + c_2\theta_s + c_3.$$

Δp and Δn are monotonically decreasing with sampling period.

3.2 Discrete time model

The dynamics of the discrete time system in matrix form are the following:

$$x(k+1) = A_d x(k) + B_d u(k) + F_d T_l(t)$$

where $x(t) = [p(t) \quad n(t) \quad T(t) \quad T_{mem}(t)]'$, $u(t) = \alpha(t)$ and the dynamical matrices are

$$A_d = \begin{bmatrix} a_{pd} & 0 & 0 & 0 \\ 0 & a_{nd} & b_{nd} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} B_d = \begin{bmatrix} b_{pd} \\ 0 \\ 0 \\ 0 \end{bmatrix} F_d = \begin{bmatrix} 0 \\ -b_{nd} \\ 0 \\ 0 \end{bmatrix}$$

Consider now the reset function, in matrix form, occurring at instant t_{TDC}:

$$x(t_{TDC}^+) = A_{reset}\, x\,(t_{TDC}^-) + B_{reset} u_r(t_{TDC}^-) + \\ + F_{reset} d_r(t_{TDC}^-) + G_{reset}$$

where

$$u_r(t_{TDC}^-) = \theta_s(t_{TDC}^-) \in [\theta_{smin}, \theta_{smax}]$$
$$d_r(t_{TDC}^-) = [\Delta p(t_{TDC}^-) \quad \Delta n(t_{TDC}^-)]' \in I_p \times I_n$$

and the matrices are

$$A_{reset} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ c_1 & 0 & 0 & 0 \end{bmatrix} B_{reset} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ c_2 \end{bmatrix}$$
$$G_{reset} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ c_3 \end{bmatrix} F_{reset} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ c_1 & 0 \end{bmatrix}$$

The model is shown in Figure 4.

4. SAFE SET COMPUTATION

Before presenting the algorithm that, by means of external approximation, finds the maximal safe set, we show that ta non-empty safe set exists

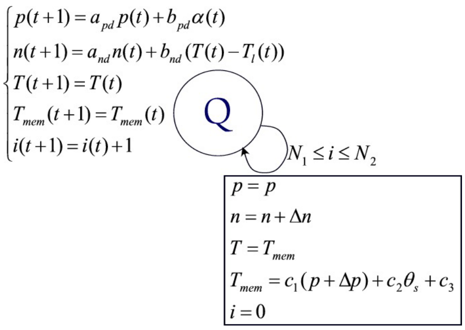

Fig. 4. Discrete time model with dwell time and non-synchronization error

if the bounds on the disturbances are sufficiently small. Consider the dynamics of the discrete time system in matrix form as in Subsection 3.2, obtained by neglecting the disturbance torque $T_l(t)$:

$$x(t+1) = A_d x(t) + B_d u(t)$$

and the reset function occurring at instant t_{TDC}, obtained by neglecting the non-synchronization error $[\Delta p \quad \Delta n]'$:

$$x(t_{TDC}^+) = A_{reset} x(t_{TDC}^-) + B_{reset} u_r(t_{TDC}^-) + G_{reset}$$

The evolution of the system is governed by $N-$consecutive discrete steps, with $N \in [N_1, N_2]$, followed by the reset function. A controlled equilibrium point exists for both discrete dynamics and reset function, given by:

$$\begin{cases} T_0 = \dfrac{1 - a_{nd}}{b_{nd}} n_0 \\ T_0 = T_{mem0} \\ T_{mem0} = c_1 p_0 + c_2\theta_{s0} + c_3 \\ p_0 = \dfrac{-b_{pd}}{a_{pd} - 1}\alpha_0 \end{cases} \quad (5)$$

where $x_0 = [p_0 \quad n_0 \quad T_0 \quad T_{mem0}]'$ is the equilibrium point, α_0 and θ_{s0} the control values that guarantee $x(k) \stackrel{\triangle}{=} x_k = x_0, \quad \forall k \geq 0$.

Let us now introduce the torque disturbance, T_l, and the non-synchronization errors, Δp and Δn, as described in Subsection 3.1. Only the engine speed dynamics is affected by those disturbances, hence it is enough to analyze the evolution of $n(t)$. Consider a neighborhood $\mathcal{O}$ of the equilibrium point n_0 and suppose that, after a reset, n belongs to $\mathcal{O}$. Since the engine speed dynamics is asymptotically stable ($|a_{nd}| < 1$), before the next switching n belongs to a contraction of $\mathcal{O}$ and after the reset n belongs to $\mathcal{O}$, if T_l and the sampling time $t_{sampling}$ are sufficiently small. We can determine precise bounds on T_l and $t_{sampling}$ as follows. Engine speed after N samples and immediately after the reset, is:

$$n_N = a_{nd}^N n_0 + \sum_{i=0}^{N-1} a_{nd}^i b_{nd}(T_0 + c_1\Delta p) - \qquad (6)$$
$$- \sum_{i=0}^{N-1} a_{nd}^i b_{nd} T_l(k-i-1) + (1 + a_{nd}^N)\Delta n$$

Iterating equation (6), engine speed after the k^{th} reset is:

$$n_{kN} = a_{nd}^{kN} n_0 + \sum_{i=0}^{kN-1} a_{nd}^i b_{nd}(T_0 + c_1\Delta p) - \qquad (7)$$
$$- \sum_{i=0}^{kN-1} a_{nd}^i b_{nd} T_l(k-i-1) + \sum_{i=0}^{k-1} (a_{nd}^N)^i \Delta n;$$

Since $0 < a_{nd} < 1$, $b_{nd} > 0$, the torque disturbance T_l and the non-synchronization errors Δp and Δn are bounded, it is possible to find the asymptotic value of the engine speed subject to disturbances:

$$n_\infty = \lim_{k\to\infty} n_{kN} = \frac{1}{1-a_{nd}} b_{nd}(T_0 + c_1\Delta p)$$
$$- \frac{1}{1-a_{nd}} b_{nd} T_l + \frac{1}{1-a_{nd}^N}\Delta n$$

that, considering equation (5), becomes

$$n_\infty = n_0 + \frac{1}{1-a_{nd}} b_{nd} c_1 \Delta p - \frac{1}{1-a_{nd}} b_{nd} T_l$$
$$+ \frac{1}{1-a_{nd}^N}\Delta n. \qquad (8)$$

By considering the extremal values T_l, Δp and Δn in (8), we can find the extremal values for n_∞.

4.1 Algorithm

The algorithm for safe-set computation shown in Figure 5 is based on the algorithm developed in [Berardi *et al.*, 2000]. The following operators are used in the computation:

- $R^{-1}(X,Y) = \{x \in Y | \exists u \in U : \forall v \in V, \quad R(x,u,v) \in X\}$, with $R(\cdot,\cdot,\cdot)$ the reset function that transforms the state x according to a feasible control action $u \in U$ and an admissible disturbance action $v \in V$; in the considered model we have $u = \theta_s$ and $v = d_r = [\Delta p \quad \Delta n]'$;
- $Reach(X,Y) = \{x \in Y | \exists u \in U : \forall v \in V, \quad Ax + Bu + Fv \in X\}$, with A, B, F dynamical matrices of the discrete time system, $u \in U$ a feasible control and $v \in V$ an admissible disturbance; in the considered model we have $u = \alpha$ and $d = T_l$.

Algorithm outputs a sequence of sets $\{\Omega_i\}_{i=0...N}$, with the following properties:

$INIT : \Omega_0 = \Lambda;$
$MEM : \Omega_{0old} = \Omega_0;$
  $\Omega_{N_2} = R^{-1}(\Omega_0) \cap \Lambda;$
  if $\Omega_{N_2} = \oslash$ **goto** $STOP\_NOK;$
  $j = 1;$
  while $j \le (N_2 - N_1)$
    $\Omega_{N_2-j} = Reach(\Omega_{N_2-j+1}, \Omega_{N_2});$
    if $\Omega_{N_2-j} = \oslash$ **goto** $STOP\_NOK;$
    $j = j + 1;$
  while $j \le N_2$
    $\Omega_{N_2-j} = Reach(\Omega_{N_2-j+1}, \Lambda);$
    if $\Omega_{N_2-j} = \oslash$ **goto** $STOP\_NOK;$
    $j = j + 1;$
  if $\Omega_0 = \Omega_{0old}$ **goto** $STOP\_OK;$
    else goto $MEM;$
$STOP\_NOK :$ *Safe Set doesn't exist!*
$STOP\_OK :$ *Safe Set exists*

Fig. 5. Safe set computation algorithm

- Ω_0 is the safe set after the reset, i.e. spark advance controller selects θ_s in order to control the state in Ω_0;
- Ω_i, for $i \in [1, N]$ is the safe set at the i^{th} sample after a TDC, i.e. throttle angle controller selects $\alpha(i-1)$ in order to control the state $x(i)$ in Ω_i.

The evolution of the controlled system can be summarized by the following sequence:

$$\Omega_0 \xrightarrow{\alpha_0} \Omega_1 \xrightarrow{\alpha_1} \dots \xrightarrow{\alpha_{N-1}} \Omega_N \xrightarrow{\theta_s} \Omega_0 \xrightarrow{\alpha_0} \dots$$

where the label upon the arrow represents the particular control used to move in the target set.

The set Λ is a polytope, because of the initial constraint given for the state variables. In particular, constraints on n are part of the specifications of the problem, while the constraints on the other variables are essentially of physical nature: for example, the torque is limited by engine power. The dynamics and the reset function are respectively linear and affine, so that each set Ω_i is a polytope too. This property simplifies the controller design; in fact, given the affine dynamics,

$$x(t+1) = Ax(t) + Bu(t) + Fv(t) + G$$

the controller u that guides the state in the set $\Omega_i = \{x \in \Lambda : W_i x \le M_i\}$ belongs to the set $U_{safe} = \{u \in U : (W_i B)u \le M_i - (W_i A)x - W_i C - max(W_i F v)\}$, where U is the set of feasible controller values.

5. DIGITAL CONTROL OF THE CONTINUOUS TIME SYSTEM

By extending the results of De Santis *et al.* [2004] to the current framework, the sequence of sets $\{\Omega_i\}$ computed by the algorithm of Figure 5 can be proven to be safe for the continuous time switching system with respect to the constraint

$$n \in [n_0 - \mu\Delta_n, \quad n_0 + \mu\Delta_n]$$

where μ is a computable factor. μ is actually a decreasing function of the sampling period and approaches 1 as the sampling period goes to 0. The control law that makes the sequence safe is a piecewise constant function, computed on the basis of the control law that makes the sequence safe for the discretized system.

The simulation results are shown in the following figures and demonstrate how the controller found for the sampled-time system can be successfully applied to the original continuous time system.

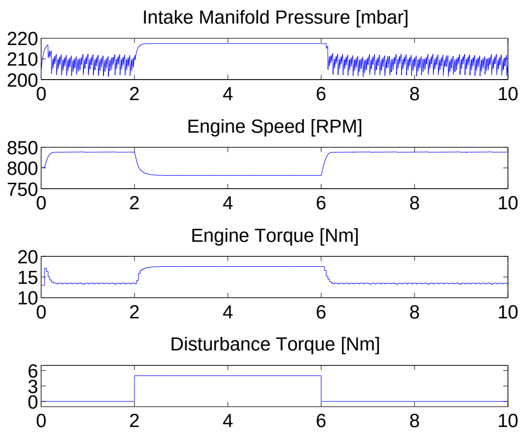

Fig. 6. Controlled System Simulation - Continuous Time Model

6. CONCLUDING REMARKS

Idle control in automotive design is a challenging problem that has been the subject of extensive investigation. In this paper, we used hybrid systems to model and to solve the problem with accuracy adequate for real applications. The idle control problem consists of maintaining the speed of the engine within a given range in presence of torque disturbances that model the energy demand on the engine posed by subsystems, such as air conditioning, that are activated at unpredictable times. We took advantage of the characteristics of the problem to simplify a general hybrid model of the engine that is used to derive the control law. Our procedure does not assume that sampling times and switching times are synchronized, a common assumption made by other authors who tackled the idle control problem. This is an important technical aspect for problems where discrete dynamics are due to both time and event driven phenomena.

REFERENCES

A. Balluchi, L. Benvenuti, L. Berardi, E. De Santis, M.D. Di Benedetto, G. Girasole, G. Pola. "Engine Idle Speed Control via Maximal Safe-Set Computation in the Crank-Angle Domain", 2002 IEEE International Symposium on Industrial Electronics (IEEE-ISIE 2002), L'Aquila, Italy, 8-11 July 2002.

A. Balluchi, M. D. Di Benedetto, A. Ferrari, G. Gaviani, G. Girasole, C. Grossi, W. Nesci, M. Pennese, A. L. Sangiovanni-Vincentelli. "Design of a motorcycle engine control unit using an integrated control-implementation approach", in Proceedings of 1st IFAC Symposium on Advances in Automotive Control (IFAC-AAC'04), University of Salerno, Italy, April 2004.

E. De Santis, M.D. Di Benedetto, L. Berardi, "Computation of Maximal Safe Sets for Switching Systems", IEEE Transaction on Automatic Control, Volume 49, Issue 2, pp. 184-195, Feb. 2004.

E. De Santis, M.D. Di Benedetto, G. Pola. "Digital control of continuous-time switching systems with safety constraints", 43rd IEEE Conference on Decision and Control 2004 (CDC), Atlantis, Paradise Island, Bahamas, Dec. 2004.

L. Palopoli, C. Pinello, A.L. Sangiovanni-Vincentelli, L. El-Ghaoui, A. Bicchi. "Synthesis of robust control systems under resource constraints", in Hybrid Systems: Computation and Control, ed. M. Greenstreet and C. Tomlin, pp. 337-350, Springer-Verlag 2002.