**Delft University of Technology**

**Fac. of Information Technology and Systems** **Control Systems Engineering**

Technical report bds:99-03

# Optimal control of a class of linear hybrid systems with saturation*

B. De Schutter

*If you want to cite this report, please use the following reference instead:*
B. De Schutter, "Optimal control of a class of linear hybrid systems with saturation," *SIAM Journal on Control and Optimization*, vol. 39, no. 3, pp. 835–851, 2000. doi:10.1137/S0363012999354648

Control Systems Engineering
Faculty of Information Technology and Systems
Delft University of Technology
Delft, The Netherlands
Current URL: https://www.dcsc.tudelft.nl

# Optimal Control of a Class of Linear Hybrid Systems with Saturation

Bart De Schutter*

**Abstract**

We consider a class of first order linear hybrid systems with saturation. A system that belongs to this class can operate in several modes or phases; in each phase each state variable of the system exhibits a linear growth until a specified upper or lower saturation level is reached, and after that the state variable stays at that saturation level until the end of the phase. A typical example of such a system is a traffic signal controlled intersection. We develop methods to determine optimal switching time sequences for first order linear hybrid systems with saturation that minimize criteria such as average queue length, worst case queue length, average waiting time, and so on. First we show how the Extended Linear Complementarity Problem (ELCP), which is a mathematical programming problem, can be used to describe the set of system trajectories of a first order linear hybrid systems with saturation. Optimization over the solution set of the ELCP then yields an optimal switching time sequence. Although this method yields globally optimal switching time sequences, it is not feasible in practice due to its computational complexity. Therefore, we also present some methods to compute suboptimal switching time sequences. Furthermore, we show that if there is no upper saturation then for some objective functions the globally optimal switching time sequence can be computed very efficiently. We also discuss some approximations that lead to suboptimal switching time sequences that can be computed very efficiently. Finally, we use these results to design optimal switching time sequences for traffic signal controlled intersections.

**Keywords**: Hybrid Systems, Control, Nonlinear Optimization, Extended Linear Complementarity Problem

**AMS subject classifications:** 93C10, 49N99, 90C33, 90B22

**Running title:** Control of Linear Hybrid Systems with Saturation

---

*Control Laboratory, Faculty of Information Technology and Systems, Delft University of Technology, P.O. Box 5031, 2600 GA Delft, The Netherlands, phone: +31-15-278.51.13, fax: +31-15-278.66.79, email: b.deschutter@its.tudelft.nl

# 1 Introduction

Hybrid systems arise from the interaction between continuous variable systems[1] and discrete event systems[2]. In general we could say that a hybrid system can be in one of several modes whereby in each mode the behavior of the system can be described by a system of difference or differential equations, and that the system switches from one mode to another due to the occurrence of an event. There are many frameworks to model, analyze and control hybrid systems (see e.g. [14, 11, 1, 17] and the references cited therein). An important trade-off in this context is that of modeling power versus decision power: the more accurate the model is the less we can analytically say about its properties. Furthermore, many analysis and control problems lead to computationally hard problems for even the most elementary hybrid systems [2]. Therefore, we focus on a specific class of hybrid systems that can be analyzed using a mathematical programming problem that is called the Extended Linear Complementarity Problem. More specifically, we study the design of optimal switching time sequences for a class of first order linear hybrid systems subject to saturation.

This work is an extension of the work reported in [8] in which we developed some algorithms to design optimal traffic signal switching schemes for single intersections. In [8] we only considered fixed amber durations and we could only efficiently compute suboptimal switching schemes for *approximations* of the real objective functions. Now we allow variable durations for the amber phases, and we show that — if there is no upper saturation — then for certain objective functions the optimal switching scheme can be computed very efficiently without making any approximations. Furthermore, in this paper we also consider a more general class of systems than the traffic signal controlled intersections of [8].

The work reported here is closely related to optimal traffic signal control (see e.g. [10, 13, 15, 16]). The main difference between the model presented in this paper applied to traffic signal optimization and the models used by most other researchers is that in our approach the length of the green-amber-red cycles may vary from cycle to cycle, i.e. we optimize over a fixed number of switch-overs instead of over a fixed number of time steps. This allows us to optimize not only the split but also the cycle time with continuous optimization variables (usually the optimization of split and cycle time is performed using boolean variables at each time step, each variable corresponding to the decision of switching or not the traffic signals as in UTOPIA, OPAC, SCOOT or SCATS). Our method adds an extra degree of freedom, which will in general lead to a more optimal switching scheme.

This paper is organized as follows. In Section 2 we discuss model predictive control, which is the framework in which our approach can be embedded. Next we give the definition and a brief description of the Extended Linear Complementarity Problem. In Section 3 we introduce a class of first order linear hybrid systems with saturation. We show that computing the optimal switching time instants in general leads to a non-convex optimization problem or to an optimization problem over the solution set of an extended linear complementarity problem. In Section 4 we show that if there is no upper saturation then for some objective functions the feasible set of the optimal switching problem can be replaced by a convex set without changing the optimum. In that case the optimal switching time sequence can be computed very efficiently. Furthermore, by making some approximations the problem becomes a linear programming problem. These results will be illustrated in Section 5 in which we compute

[1] Continuous variable systems are systems that can be described by a difference or differential equation.

[2] Discrete event systems are asynchronous systems where the state transitions are initiated by events; in general the time instants at which these events occur are not equidistant.

optimal traffic signal switching time sequences for traffic signal controlled intersections.

# 2 Preliminaries

## 2.1 Notation

Let $a$ and $b$ be vectors with $n$ components. The $i$th component of $a$ is denoted by $a_i$ or $(a)_i$. We use $a \geqslant b$ to indicate that $a_i \geqslant b_i$ for all $i$. The maximum operator on vectors is defined as follows: $\big(\max(a, b)\big)_i = \max(a_i, b_i)$ for all $i$. The minimum operator on vectors is defined analogously. The zero vector with $n$ components is denoted by $0_n$, or by 0 if the dimension is clear from the context. The $n$ by $n$ identity matrix is denoted by $I_n$, or by $I$ if the dimension is clear from the context. The set of the real numbers is denoted by $\mathbb{R}$.

## 2.2 Model predictive control

Model predictive control (MPC) [3, 4, 9] is a very popular controller design method in the process industry. An important advantage of MPC is that it allows the inclusion of constraints on the inputs and outputs, and that it can handle changes in the system parameters by using a moving horizon approach, in which the model and the control strategy are continuously updated. We will use the MPC framework to design optimal switching schemes for a class of hybrid systems. In general the resulting optimization problem is nonlinear and non-convex. However, if the control objective and the constraints depend monotonically on the outputs of the system, the MPC problem can be recast as problem with a convex feasible set. As a consequence, the problem can be solved very efficiently so that on-line computation is feasible.

In each step of the conventional MPC algorithm for discrete-time systems an optimal input sequence is computed that minimizes a given cost criterion over a given prediction horizon $N_\mathrm{p}$. Furthermore, for the optimization the control input $u$ is taken to be constant from a certain point on: $u(k+j) = u(k+N_\mathrm{c}-1)$ for $j = N_\mathrm{c}, N_\mathrm{c}+1, \ldots, N_\mathrm{p}-1$ where $N_\mathrm{c}$ is the control horizon and where $k$ is the first sampling index of the period under consideration. MPC uses a receding horizon principle: after computation of the optimal control sequence $u(k), u(k+1), \ldots, u(k+N_\mathrm{c}-1)$, only the first control input sample $u(k)$ will be implemented; subsequently the horizon is shifted one sample, the estimates of the state and the parameters of the system are updated using information coming from new measurements, and the optimization is restarted. Note that the continuous updating of the model and the estimates of the states also introduces a kind of feedback in the control system. In general feedback is necessary to obtain good performance and tracking in most control applications (see e.g. [12] for applications of feedback control in traffic).

The parameters $N_\mathrm{p}$ and $N_\mathrm{c}$ are the basic tuning parameters of the MPC algorithm:

- In general the prediction horizon $N_\mathrm{p}$ is selected such that the time interval $[k, k+N_\mathrm{p}-1]$ contains the crucial dynamics of the process.

- An important effect of a small control horizon $N_\mathrm{c}$ is the smoothing of the control signal (because of the emphasis on the average behavior rather than on aggressive noise reduction). The control horizon forces the control signal to a constant value. This also has a stabilizing effect since the output signal is forced to its steady-state value. Another important consequence of decreasing $N_\mathrm{c}$ is the reduction of the number of optimization variables, which results in a decrease of the computational effort.

### 2.3 The Extended Linear Complementarity Problem

The Extended Linear Complementarity Problem (ELCP) is a mathematical programming problem which is defined as follows [7]:

> Given $A \in \mathbb{R}^{p \times n}$, $B \in \mathbb{R}^{q \times n}$, $c \in \mathbb{R}^p$, $d \in \mathbb{R}^q$ and $m$ subsets $\phi_1, \phi_2, \ldots, \phi_m$ of $\{1, 2, \ldots, p\}$, find $x \in \mathbb{R}^n$ such that
>
> $$\sum_{j=1}^{m} \prod_{i \in \phi_j} (Ax - c)_i = 0 \tag{1}$$
>
> subject to $Ax \geqslant c$ and $Bx = d$, or show that no such $x$ exists.

The ELCP can be considered as a system of linear equations and inequalities ($Ax \geqslant c$, $Bx = d$), where there are $m$ groups of linear inequalities (one group for each index set $\phi_j$) such that in each group at least one inequality should hold with equality. In [7] we have developed an algorithm to compute the complete solution set of an ELCP. In general this solution set consists of the union of a subset of faces of the polyhedron $\mathcal{P}$ defined by the system $Ax \geqslant c$, $Bx = d$ (i.e. the solution set contains all the points of $\mathcal{P}$ that satisfy condition (1)). Our ELCP algorithm yields a compact representation of the solution set of an ELCP by vertices, extreme rays and a basis of the linear subspace corresponding to the largest affine subspace of the solution set. In [7] we have also shown that the general ELCP is NP-hard.

In the next section we shall show that the ELCP can be used to determine optimal switching time instants for a special class of hybrid systems.

## 3 Optimal switching time sequences for a class of linear hybrid systems with saturation

### 3.1 First order linear hybrid systems with saturation

Consider a system the evolution of which is characterized by consecutive phases. In each phase each state variable of the system exhibits a linear growth or decrease until a certain upper or lower saturation level is reached; then the state variable stays constant until the end of the phase. A system the behavior of which satisfies this description will be called a *first order linear hybrid system with saturation.*

A typical example of a first order linear hybrid system with saturation is a traffic signal controlled intersection provided that we use a continuous approximation for the queue lengths (see Section 5 and [8]). The state variables of this system correspond to the queue lengths in the different lanes. For a traffic signal controlled intersection the lower bound for the queue length is equal to 0. The upper bound could correspond to the maximal available storage space due to the distance to the preceding junction or due to the layout of the intersection. We assume that if this upper bound is reached then newly arriving cars take another route to get to their destination. Another example of a first order linear hybrid system with saturation is a system consisting of several fluid containers that are connected by tubes with valves and that have two outlets — one at the bottom (with a tube that leads to another fluid container), and one at the top (so that the fluid level in the containers can never exceed a given level), — provided that we assume that the increase or decrease of the fluid levels is linear if the system is not saturated.

Now we derive the equations that describe the evolution of the state variables in a first order linear hybrid system with saturation. In analogy with a traffic signal controlled intersection, we will use the word "queue lengths" to refer to the state variables of the system. Note however that our definition of a first order linear hybrid system with saturation is not limited to queuing systems only.

Let $M$ be the number of "queues". The length of queue $i$ at time $t$ is denoted by $q_i(t)$. Let $\alpha_{i,k}$, $b^{\mathrm{ls}}_{i,k}$ and $b^{\mathrm{us}}_{i,k}$ be respectively the queue length growth rate for queue $i$ in phase $k$, the lower saturation bound for the queue length $q_i$ in phase $k$ and the upper saturation bound for the queue length $q_i$ in phase $k$. The evolution of the system begins at time $t_0$. Let $t_1, t_2, t_3, \ldots$ be the switching time instants, i.e. the time instants at which the system switches from one phase to another. Note that in general the sequence $t_0, t_1, t_2, \ldots$ is *not* an equidistant sequence. The length of the $k$th phase is equal to $\delta_k \stackrel{\mathrm{def}}{=} t_{k+1} - t_k$. Note that $\delta_k > 0$ for all $k$. We assume that $0 \leqslant b^{\mathrm{ls}}_{i,k+1} \leqslant q_i(t_{k+1}) \leqslant b^{\mathrm{us}}_{i,k+1}$ for all $i, k$ such that the queue lengths are always nonnegative and such that there are no sudden jumps in the queue lengths due to a change in the saturation level at one of the switching time instants. For queue $i$ we have

$$\frac{dq_i(t)}{dt} = \begin{cases} \alpha_{i,k} & \text{if } b^{\mathrm{ls}}_{i,k} < q_i(t) < b^{\mathrm{us}}_{i,k} \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

for $t \in (t_k, t_{k+1})$. This implies that the evolution of the queue lengths at the switching time instants is given by

$$q_i(t_{k+1}) = \max\big(\min\left(q_i(t_k) + \alpha_{i,k}\delta_k,\, b^{\mathrm{us}}_{i,k}\right),\, b^{\mathrm{ls}}_{i,k}\big) \tag{3}$$

for $k = 0, 1, 2, \ldots$ If we define $q_{i,k} = q_i(t_k)$ and

$$q_k = \begin{bmatrix} q_{1,k} \\ q_{2,k} \\ \vdots \\ q_{M,k} \end{bmatrix}, \quad \alpha_k = \begin{bmatrix} \alpha_{1,k} \\ \alpha_{2,k} \\ \vdots \\ \alpha_{M,k} \end{bmatrix}, \quad b^{\mathrm{ls}}_k = \begin{bmatrix} b^{\mathrm{ls}}_{1,k} \\ b^{\mathrm{ls}}_{2,k} \\ \vdots \\ b^{\mathrm{ls}}_{M,k} \end{bmatrix}, \quad b^{\mathrm{us}}_k = \begin{bmatrix} b^{\mathrm{us}}_{1,k} \\ b^{\mathrm{us}}_{2,k} \\ \vdots \\ b^{\mathrm{us}}_{M,k} \end{bmatrix},$$

we obtain the vector equation

$$q_{k+1} = \max\big(\min\left(q_k + \alpha_k\delta_k,\, b^{\mathrm{us}}_k\right),\, b^{\mathrm{ls}}_k\big) \ . \tag{4}$$

If we introduce dummy vectors $z_k$, then (3) can be rewritten as

$$z_{k+1} = \min(q_k + \alpha_k\delta_k, b^{\mathrm{us}}_k) \tag{5}$$

$$q_{k+1} = \max(z_{k+1}, b^{\mathrm{ls}}_k) \ . \tag{6}$$

### 3.2 Optimal switching time sequences for linear hybrid systems with saturation

Now we consider the problem of computing an optimal (finite) sequence of switching time instants for a system described by a system of equations of the form (4) using an MPC approach.

We may assume without loss of generality that $t_0$ will be the first switching time instant in each step of the MPC algorithm. Note that this implies that switching time instant $t_1$ of the current MPC step will correspond to switching time instant $t_0$ of the next MPC step. The queue length vector $q_0 = q(t_0)$ at time $t = t_0$ can be measured[3] or estimated. Now we want to determine the optimal switching time sequence $t_0, t_1, \ldots, t_{N_\mathrm{p}}$ for a given performance criterion $J$. For the class of systems we consider it makes more sense to replace the condition that the control input is constant after the control horizon by the condition

$$\delta_k = \delta_{k-K_\mathrm{c}} \qquad \text{for } k = N_\mathrm{c}, N_\mathrm{c}+1, \ldots, N_\mathrm{p}-1, \tag{7}$$

where $K_\mathrm{c}$ is the number of switching phases in one larger cycle of the system (e.g. in traffic signal control for an intersection of two streets $K_\mathrm{c}$ could be equal to 4 corresponding to the combinations red-green, red-amber, green-red, amber-red for the traffic signals on the crossing roads (see also Section 5)). Possible performance criteria are:

- (weighted) average queue length over all queues:

$$J_1 = \sum_{i=1}^{M} w_i \, \frac{1}{t_{N_\mathrm{p}} - t_0} \int_{t_0}^{t_{N_\mathrm{p}}} q_i(t)\, dt \ , \tag{8}$$

- (weighted) average queue length over the worst queue:

$$J_2 = \max_i \left( w_i \, \frac{1}{t_{N_\mathrm{p}} - t_0} \int_{t_0}^{t_{N_\mathrm{p}}} q_i(t)\, dt \right) \ , \tag{9}$$

- (weighted) worst case queue length:

$$J_3 = \max_{i,\,t} \big( w_i \, q_i(t) \big) \ , \tag{10}$$

- (weighted) average "waiting time" over all queues[4]:

$$J_4 = \sum_{i=1}^{M} w_i \, \frac{\displaystyle\int_{t_0}^{t_{N_\mathrm{p}}} q_i(t)\, dt}{\displaystyle\sum_{k=0}^{N_\mathrm{p}-1} \alpha^{\mathrm{a}}_{i,k} \delta_k} \ , \tag{11}$$

- (weighted) average "waiting time" over the worst queue:

$$J_5 = \max_i \left( w_i \, \frac{\displaystyle\int_{t_0}^{t_{N_\mathrm{p}}} q_i(t)\, dt}{\displaystyle\sum_{k=0}^{N_\mathrm{p}-1} \alpha^{\mathrm{a}}_{i,k} \delta_k} \right) \ , \tag{12}$$

where $w_i > 0$ for all $i$ and $\alpha^{\mathrm{a}}_{i,k}$ is the arrival rate of "customers" for queue $i$ in phase $k$.

[3] Note that if we compute the switching time sequence fast enough (i.e. if the computation time is less than $\delta_0 = t_1 - t_0$) we can wait with computing the optimal sequence until after $t_0$.

[4] The average waiting time is equal to the total waiting time divided by the number of arrivals. If the initial and final queue lengths are 0, then the average waiting time for queue $i$ is given by the fraction in the expression on the right-hand side of (11). So $J_4$ is in fact an approximation of the (weighted) average waiting time.

We can impose extra conditions such as minimum or maximum queue lengths (which could be useful in order to prevent saturation at the lower or upper level for some queues), minimum and maximum durations for the switching time intervals, and so on.

This leads to the following optimization problem that should be solved in each MPC step:

$$\underset{\delta_0,\delta_1,\ldots,\delta_{N_\mathrm{c}-1}}{\text{minimize}} \; J \tag{13}$$

subject to

$$\delta_k = \delta_{k-K_\mathrm{c}} \qquad \text{for } k = N_\mathrm{c}, N_\mathrm{c}+1, \ldots, N_\mathrm{p}-1, \tag{14}$$

$$\delta_{\min,k} \leqslant \delta_k \leqslant \delta_{\max,k} \qquad \text{for } k = 0, 1, \ldots, N_\mathrm{c}-1, \tag{15}$$

$$q_{\min,k} \leqslant q_{k+1} \leqslant q_{\max,k} \qquad \text{for } k = 0, 1, \ldots, N_\mathrm{p}-1, \tag{16}$$

$$z_{k+1} = \min(q_k + \alpha_k\delta_k,\, b_k^{\mathrm{us}}) \qquad \text{for } k = 0, 1, \ldots, N_\mathrm{p}-1, \tag{17}$$

$$q_{k+1} = \max(z_{k+1},\, b_k^{\mathrm{ls}}) \qquad \text{for } k = 0, 1, \ldots, N_\mathrm{p}-1, \tag{18}$$

with $q_0 = q(t_0)$ and where $\delta_{\min,k}$ and $\delta_{\max,k}$ are respectively the minimum and the maximum values of $\delta_k$, and $(q_{\min,k})_i$ and $(q_{\max,k})_i$ are respectively the minimum and the maximum queue lengths for queue $i$ at time instant $t_{k+1}$.

**Remark 3.1** We can also use a first order linear hybrid system with saturation as an approximate model if we have a hybrid system with saturation in which the queue length growth or decrease rates are slowly time-varying since in MPC we use a moving horizon approach in which the model of the system and the estimate of the initial condition can be updated at the beginning of each control cycle. This also introduces a feedback into the control system. $\diamond$

### 3.3 The Extended Linear Complementarity Problem and optimal switching time sequences

Now we show that the system (14) – (18) can be reformulated as an ELCP.

Consider (17) for an arbitrary index $k$. This equation can be rewritten as follows:

$$\begin{gathered} z_{k+1} \leqslant q_k + \alpha_k\delta_k \\ z_{k+1} \leqslant b_k^{\mathrm{us}} \\ (z_{k+1})_i = (q_k + \alpha_k\delta_k)_i \quad \text{or} \quad (z_{k+1})_i = (b_k^{\mathrm{us}})_i \qquad \text{for } i = 1, 2, \ldots, M\,, \end{gathered}$$

or equivalently

$$q_k + \alpha_k\delta_k - z_{k+1} \geqslant 0 \tag{19}$$

$$b_k^{\mathrm{us}} - z_{k+1} \geqslant 0 \tag{20}$$

$$(q_k + \alpha_k\delta_k - z_{k+1})_i\,(b_k^{\mathrm{us}} - z_{k+1})_i = 0 \qquad \text{for } i = 1, 2, \ldots, M \tag{21}$$

Since a sum of nonnegative numbers is equal to 0 if and only if all the numbers are equal to 0, (21) is equivalent to:

$$\sum_{i=1}^{M} (q_k + \alpha_k\delta_k - z_{k+1})_i\,(b_k^{\mathrm{us}} - z_{k+1})_i = (q_k + \alpha_k\delta_k - z_{k+1})^T\,(b_k^{\mathrm{us}} - z_{k+1}) = 0 \;\;.$$

Hence, (17) can be rewritten as

$$q_k + \alpha_k \delta_k - z_{k+1} \geqslant 0 \tag{22}$$

$$b_k^{\mathrm{us}} - z_{k+1} \geqslant 0 \tag{23}$$

$$(q_k + \alpha_k \delta_k - z_{k+1})^T (b_k^{\mathrm{us}} - z_{k+1}) = 0 \ . \tag{24}$$

We can repeat this reasoning for (18) and for each index $k$. So if we define

$$x_q = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_{N_\mathrm{p}} \end{bmatrix}, \quad x_z = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_{N_\mathrm{p}} \end{bmatrix}, \quad x_\delta = \begin{bmatrix} \delta_0 \\ \delta_1 \\ \vdots \\ \delta_{N_\mathrm{c}-1} \end{bmatrix},$$

and if we replace all $\delta_k$'s with index $k \geqslant N_\mathrm{c}$ using (14), we finally get a problem of the form

$$\underset{x_\delta}{\text{minimize}} \ J \tag{25}$$

subject to

$$A x_q + B x_z + C x_\delta + d \geqslant 0 \tag{26}$$

$$E x_q + F x_z + g \geqslant 0 \tag{27}$$

$$H x_q + K x_\delta + l \geqslant 0 \tag{28}$$

$$(A x_q + B x_z + C x_\delta + d)^T (E x_q + F x_z + g) = 0 \ , \tag{29}$$

for appropriately defined matrices $A$, $B$, $C$, $E$, $F$, $H$, $K$ and vectors $d$, $g$, $l$. Equations (26), (27) and (29) correspond to (22), (23) and (24) respectively, and the system of linear inequalities (28) contains the conditions (15) and (16). It is easy to verify that the system (26) – (29) is (a special case of) an ELCP.

The time evolution of the queue lengths in a first order linear hybrid system with saturation is given by piecewise affine functions. The link between piecewise affine functions and (ordinary) linear complementarity problems has also been explored by several other authors (see e.g. [5] and the references therein).

**Remark 3.2** If we introduce additional linear equality or inequality constraints on the components of $x_\delta$ such as e.g. a maximum or total duration for the $N_\mathrm{p}$ phases ($\delta_0 + \delta_1 + \cdots + \delta_{N_\mathrm{p}} \leqslant T_{\max}$ or $\delta_0 + \delta_1 + \cdots + \delta_{N_\mathrm{p}} = T_{\mathrm{tot}}$), maximum or total durations for two or more consecutive phases (e.g. $\delta_{2k} + \delta_{2k+1} \leqslant T_{\max,k}$ or $\delta_{2k} + \delta_{2k+1} = T_{\mathrm{tot},k}$), we still obtain an ELCP. The additional linear inequality constraints lead to extra inequalities in (28), and the additional linear equality constraints lead to an extra equation of the form $P x_\delta + q = 0$, which also fits in the ELCP framework. $\diamond$

The ELCP (26) – (29) describes all feasible system trajectories for the first order linear hybrid system with saturation. In order to determine the optimal switching time sequence we could minimize the objective function $J$ over the solution set of the ELCP as follows. If we assume that $x_q$ and $x_\delta$ are bounded[5], then the solution set of the ELCP consists of a union of faces of the (finite and bounded) polytope defined by (26) – (28). Each face of the polyhedron can be represented by its vertices, and the points of the face can be written as convex combinations of these vertices. We could for each face determine for which convex combination of the vertices the objective function $J$ reaches a global minimum over the face and afterwards select the overall minimum.

[5]A sufficient condition for this is that $\delta_{\min,k}$ and $\delta_{\max,k}$ are defined and finite for all $k$.

However, the general ELCP is an NP-hard problem [7]. Furthermore, the algorithm of [7] to compute the solution set of a general ELCP requires exponential execution times. This implies that the ELCP approach sketched above is not feasible if the number of variables is large. Since the number of variables in the ELCP is equal to $2MN_{\rm p} + N_{\rm c}$, this implies that the ELCP should not be used if $M$, $N_{\rm p}$ or $N_{\rm c}$ are large.

If the ELCP is not tractable, we could either select lower values for $N_{\rm c}$ and $N_{\rm p}$ (which would result in less optimal solutions) or we could use multi-start local optimization to determine the optimal switching scheme. For given $N_{\rm p}$, $N_{\rm c}$, $K_{\rm c}$, $q_0$, $\alpha_{i,k}$'s, $b^{\rm ls}_{i,k}$'s and $b^{\rm us}_{i,k}$'s, the evolution of the system to be optimized is uniquely determined by the sequence $\delta_0, \delta_1, \ldots, \delta_{N_{\rm c}-1}$ since the remaining $\delta_k$'s, the queue lengths $q_i(t)$ and the components of $x_q$ and $x_z$ are given by (7), (2), (5) and (6) respectively. Therefore, we can consider (13) – (18) as a constrained optimization problem in $x_\delta$ where the constraints (16) – (18) are nonlinear constraints. Alternatively, these constraints can be taken into account by adding an extra penalty term to the objective function $J$ if $q_{i,k} < (q_{\min,k})_i$ or $q_{i,k} > (q_{\max,k})_i$. If we use the penalty function[6] approach, the only remaining constraints on $x_\delta$ are the simple upper and lower bound constraints (15). However, the major disadvantage of the multi-start local minimization approaches discussed above is that in general the minimization routine will only return a local minimum and that several starting points are necessary to obtain a good approximation to the global optimum. Note that the final solution $x_\delta^{\rm opt,curr}$ of the current MPC step can be used to obtain a good initial solution $x_\delta^{\rm init,next}$ for the next MPC stepping by setting $\delta_k^{\rm init,next} = \delta_{k-1}^{\rm opt,curr}$ for $k = 1, 2, \ldots, N_{\rm c}$.

Recall that in each MPC step the problem (13) – (18) has to be solved. In order to be able to do this on-line, it is important to have efficient algorithms to solve the problem. Therefore, we shall now discuss some other approaches to compute solutions very efficiently if there is no saturation at the upper level.

# 4 Optimal and suboptimal switching time sequences for systems with saturation at a lower level only

## 4.1 Optimal switching time sequences

In this section we consider systems with saturation at the lower level only. So $b^{\rm us}_{i,k}$ is equal to $\infty$ for all $i, k$, or equivalently $(q_{\max,k})_i \leqslant b^{\rm us}_{i,k}$ for all $i, k$. We also assume that $q_{\min,k} \leqslant b^{\rm ls}_k$ for all $k$, i.e. we do not impose extra lower bound conditions on the queue lengths. The optimal switching problem (13) – (18) then reduces to

$$\underset{x_\delta}{\text{minimize}}\ J \tag{30}$$

subject to

$$\begin{aligned}
\delta_k &= \delta_{k-K_{\rm c}} && \text{for } k = N_{\rm c}, N_{\rm c}+1, \ldots, N_{\rm p}-1, && (31)\\
\delta_{\min,k} \leqslant \delta_k &\leqslant \delta_{\max,k} && \text{for } k = 0, 1, \ldots, N_{\rm c}-1, && (32)\\
q_{k+1} &\leqslant q_{\max,k} && \text{for } k = 0, 1, \ldots, N_{\rm p}-1, && (33)\\
q_{k+1} &= \max(q_k + \alpha_k \delta_k,\, b^{\rm ls}_k) && \text{for } k = 0, 1, \ldots, N_{\rm p}-1. && (34)
\end{aligned}$$

[6]Note that in this case a barrier function approach is not advantageous since the optimal solution will often lie on the boundary of the feasible region.

We call this problem $\mathcal{P}$. We define the "relaxed" problem $\tilde{\mathcal{P}}$ corresponding to $\mathcal{P}$ as:

$$\underset{x_q, x_\delta}{\text{minimize}}\ J \tag{35}$$

subject to

$$\delta_k = \delta_{k-K_\mathrm{c}} \qquad \text{for } k = N_\mathrm{c}, N_\mathrm{c}+1, \ldots, N_\mathrm{p}-1, \tag{36}$$

$$\delta_{\min,k} \leqslant \delta_k \leqslant \delta_{\max,k} \qquad \text{for } k = 0, 1, \ldots, N_\mathrm{c}-1, \tag{37}$$

$$q_{k+1} \leqslant q_{\max,k} \qquad \text{for } k = 0, 1, \ldots, N_\mathrm{p}-1, \tag{38}$$

$$q_{k+1} \geqslant q_k + \alpha_k \delta_k \qquad \text{for } k = 0, 1, \ldots, N_\mathrm{p}-1, \tag{39}$$

$$q_{k+1} \geqslant b_k^{\mathrm{ls}} \qquad \text{for } k = 0, 1, \ldots, N_\mathrm{p}-1. \tag{40}$$

So compared to the original problem we have replaced (34) by relaxed equations of the form (22) – (23) without taking (24) into account. As a consequence, $x_q$ and $x_\delta$ are not directly coupled any more. The set of feasible solutions of $\tilde{\mathcal{P}}$ is a convex set, whereas the set of feasible solutions of $\mathcal{P}$ is in general not convex since (34) is a non-convex constraint. Therefore, the relaxed problem $\tilde{\mathcal{P}}$ will in general be easier to solve than the problem $\mathcal{P}$.

The objective function $J$ is a monotonically nondecreasing function of $x_q$ if for every $x_\delta$ and for every $\tilde{x}_q, \hat{x}_q$ with $\tilde{x}_q \leqslant \hat{x}_q$, we have $J(\tilde{x}_q, x_\delta) \leqslant J(\hat{x}_q, x_\delta)$. The following proposition shows that for monotonically nondecreasing objective functions any optimal solution of the relaxed problem $\tilde{\mathcal{P}}$ can be transformed into an optimal solution of the problem $\mathcal{P}$.

**Proposition 4.1** *Let the objective function $J$ be a monotonically nondecreasing function of $x_q$ and let $(x_q^*, x_\delta^*)$ be an optimal solution of $\tilde{\mathcal{P}}$. If we construct $x_q^\sharp$ such that*

$$q_1^\sharp = \max\left(q_0 + \alpha_0 \delta_0^*, b_0^{\mathrm{ls}}\right) \tag{41}$$

$$q_{k+1}^\sharp = \max\left(q_k^\sharp + \alpha_k \delta_k^*, b_k^{\mathrm{ls}}\right) \qquad \text{for } k = 1, 2, \ldots, N_\mathrm{p}-1. \tag{42}$$

*then $(x_q^\sharp, x_\delta^*)$ is an optimal solution of the problem $\mathcal{P}$.*

**Proof:** Let $(x_q^*, x_\delta^*)$ be an optimal solution of $\tilde{\mathcal{P}}$ and let $x_q^\sharp$ be defined by (41) – (42). Clearly, $(x_q^\sharp, x_\delta^*)$ is a feasible solution of $\tilde{\mathcal{P}}$. Define $q_0^* = q_0^\sharp = q_0$. Since $x_q^*$ satisfies (39) – (40), we have $\max(q_k^* + \alpha_k \delta_k^*, b_k^{\mathrm{ls}}) \leqslant q_{k+1}^*$ for all $k$. Since $q_0^* = q_0^\sharp$, this implies that $q_1^\sharp \leqslant q_1^*$ and, by induction, also that $q_k^\sharp \leqslant q_k^*$ for $k = 2, 3, \ldots, N_\mathrm{p}$. As a consequence, we have $x_q^\sharp \leqslant x_q^*$ and thus also $J(x_q^\sharp, x_\delta^*) \leqslant J(x_q^*, x_\delta^*)$ since $J$ is a monotonically nondecreasing function of $x_q$. Since $(x_q^\sharp, x_\delta^*)$ is a feasible solution of $\tilde{\mathcal{P}}$ and since $(x_q^*, x_\delta^*)$ is an optimal solution of $\tilde{\mathcal{P}}$, this implies that $(x_q^\sharp, x_\delta^*)$ is also an optimal solution of $\tilde{\mathcal{P}}$.
The set of feasible solutions of $\mathcal{P}$ is a subset of the set of feasible solutions of $\tilde{\mathcal{P}}$. Hence, the minimal value of $J$ over the set of feasible solutions of $\tilde{\mathcal{P}}$ will be less than or equal to the minimal value of $J$ over the set of feasible solutions of $\mathcal{P}$. Since $(x_q^\sharp, x_\delta^*)$ is a feasible solution of $\mathcal{P}$ and an optimal solution of $\tilde{\mathcal{P}}$, this implies that $(x_q^\sharp, x_\delta^*)$ is an optimal solution of $\mathcal{P}$. □

Recall that the objective functions $J_1$, $J_2$, $J_3$, $J_4$ and $J_5$ do not explicitly depend on $x_q$, since $x_q$ can be computed from $x_\delta$ (and eliminated from the expressions for the objective functions before considering the relaxation of $\mathcal{P}$). So we have $J_l(\tilde{x}_q, x_\delta) = J_l(\hat{x}_q, x_\delta)$ for any $\tilde{x}_q, \hat{x}_q$ and for $l \in \{1,2,3,4,5\}$. This implies that $J_1$, $J_2$, $J_3$, $J_4$ and $J_5$ are monotonically nondecreasing functions of $x_q$. So we can use Proposition 4.1 to transform the optimal switching problem for the objective functions $J_1$ up to $J_5$ into an optimization problem with a convex feasible set. The resulting (global) solution of the relaxed problem can then be transformed into an optimal switching scheme using (41) – (42). Note however that although the feasible set of the relaxed problem is convex, the objective functions $J_1$ up to $J_5$ are not convex, so that the overall problem is still non-convex (and thus in general not easily solvable). Therefore, we now introduce two subsequent approximations of the objective functions $J_1$ and $J_4$ that will lead to a linear programming problem, which can be solved very efficiently. The resulting solution can then be used as an initial starting point for the optimization of the relaxed problem.

### 4.2 A linear programming approximation

The objective function $J$ is a monotonically increasing function of $x_q$ if for every $x_\delta$ and for every $\tilde{x}_q, \hat{x}_q$ with $\tilde{x}_q \leqslant \hat{x}_q$ and $\tilde{x}_q \neq \hat{x}_q$ we have $J(\tilde{x}_q, x_\delta) < J(\hat{x}_q, x_\delta)$. The optimal solution of $\tilde{\mathcal{P}}$ will in general not be a feasible solution of $\mathcal{P}$, unless $J$ is a monotonically increasing function of $x_q$:

**Proposition 4.2** *If $J$ is a monotonically increasing function of $x_q$ then any optimal solution of the relaxed problem $\tilde{\mathcal{P}}$ is also an optimal solution of the problem $\mathcal{P}$.*

**Proof:** Let $(x_q^*, x_\delta^*)$ be an optimal solution of $\tilde{\mathcal{P}}$ and construct $(x_q^\sharp, x_\delta^*)$ as in the proof of Proposition 4.1. So $x_q^\sharp \leqslant x_q^*$ and $(x_q^\sharp, x_\delta^*)$ is also a feasible solution of $\tilde{\mathcal{P}}$.
Now we show by contradiction that $(x_q^*, x_\delta^*)$ is also a feasible solution of $\mathcal{P}$, i.e. that it satisfies (34). Suppose that $(x_q^*, x_\delta^*)$ does not satisfy (34). So $x_q^\sharp \neq x_q^*$. Since $x_q^\sharp \leqslant x_q^*$, this implies that $J(x_q^\sharp, x_\delta^*) < J(x_q^*, x_\delta^*)$, which would mean that $(x_q^*, x_\delta^*)$ is not an optimal solution of $\tilde{\mathcal{P}}$. Since this is a contradiction, our initial assumption that $(x_q^*, x_\delta^*)$ does not satisfy (34) was wrong. Hence, $(x_q^*,\, x_\delta^*)$ also is a feasible solution of the problem $\mathcal{P}$. Since the set of feasible solutions of $\mathcal{P}$ is a subset of the set of feasible solutions of $\tilde{\mathcal{P}}$, this implies that $(x_q^*,\, x_\delta^*)$ is also an optimal solution of $\mathcal{P}$. $\square$

Note that the objective functions $J_1$, $J_2$, $J_3$, $J_4$ and $J_5$ are not monotonically increasing functions of $x_q$. Now we introduce some approximations to the objective functions $J_1$ and $J_4$ that are strictly monotonically increasing functions of $x_q$ and for which Proposition 4.2 can be used[7]. This will lead to suboptimal switching time sequences that can be computed very efficiently. We will only consider the approximations for $J_1$, but for $J_4$ a similar reasoning can be made.

For a given $q_0$ and $t_0$, we define the function $\tilde{q}_i(\cdot, x_q, x_\delta)$ as the piecewise-affine function with breakpoints $(t_k, q_{i,k})$ for $k = 0, 1, \ldots, N_{\mathrm{p}}$. The approximate objective function $\tilde{J}_1$ is also defined by (8) but with $q_i$ replaced by $\tilde{q}_i$. The value of the objective functions $J_1$ and $\tilde{J}_1$ depends on the surface under the functions $q_i$ and $\tilde{q}_i$ respectively[8]. If we are computing optimal traffic switching sequences, then the surface under the function $\tilde{q}_i$ will be a reasonable approximation of the surface under the function $q_i$ and then the optimal value of $\tilde{J}_1$ will be

[7] This derivation is an extension of our work in [8] where we have considered a special subclass of first order linear hybrid systems with saturation at the lower level only. Although we did not yet use Proposition 4.1 there, we did use a proposition that is similar to Proposition 4.2.

[8] Recall that $q_i(t) \geqslant 0$ for all $i, t$ since we have assumed that $b_{i,k}^{\mathrm{ls}} \geqslant 0$ for all $i, k$.

a reasonably good approximation of the optimal value of $J_1$ (see also [6, 8]). Note that the values of $J_1$ and $\tilde{J}_1$ coincide if there is no saturation in the period under consideration. Since $\tilde{q}_i$ is a piecewise-affine with breakpoints $(t_k, q_{i,k})$ for $k = 0, 1, \ldots, N_\mathrm{p}$, we have [6]

$$\tilde{J}_1(x_q, x_\delta) \;=\; \sum_{i=1}^{M} \left( \frac{w_i}{2\,(\delta_0 + \delta_1 + \ldots + \delta_{N_\mathrm{p}-1})} \sum_{k=0}^{N_\mathrm{p}-1} \delta_k (q_{i,k} + q_{i,k+1}) \right) \qquad (43)$$

where $\delta_{N_\mathrm{c}}, \ldots, \delta_{N_\mathrm{p}-c}$ can be replaced using (7). Since $\delta_k > 0$ for all $k$, $\tilde{J}_1$ is a monotonically increasing function of $x_q$, which implies that Proposition 4.2 can be applied.

Now we discuss a further approximation of $\tilde{J}_1$ that will lead to a linear programming problem, which can be solved very efficiently. Sometimes we already have a good idea about the relative lengths of the different phases (in a traffic signal situation we know e.g. that the green phases will be much longer than the amber phases). If we assume that $\delta_k = \rho_k \bar{\delta}$ for all $k$ and for some yet unknown $\bar{\delta}$, then (43) leads to:

$$\tilde{J}_1(x_q, x_\delta) \;=\; \sum_{i=1}^{M} \frac{w_i}{2\rho_\mathrm{tot}} \left( \rho_0 q_{i,0} \;+\; \sum_{k=1}^{N_\mathrm{p}-1} (\rho_k + \rho_{k-1}) q_{i,k} \;+\; \rho_{N-1} q_{i,N_\mathrm{p}} \right) \;\stackrel{\mathrm{def}}{=}\; \hat{J}_1(x_q) \;\;.$$

with $\rho_\mathrm{tot} = \rho_0 + \rho_1 + \cdots + \rho_{N_\mathrm{p}-1}$. Note that $\hat{J}_1$ is an affine function of $x_q$. Since $w_i > 0$ for all $i$ and $\rho_k > 0$ for all $k$, $\hat{J}_1$ is a monotonically increasing function of $x_q$. Hence, by Proposition 4.2 any optimal solution of $\tilde{\mathcal{P}}$ with objective function $\hat{J}_1$ will also be an optimal solution of $\mathcal{P}$ (with objective function $\hat{J}_1$). So the optimal switching problem then reduces to a linear programming problem, which can be solved efficiently using a simplex or an interior point method.

**Remark 4.3** The values of the $\rho_k$'s are usually determined on the basis of an educated guess. Alternatively, if we have already performed an MPC step, then we can use the shifted values of the $\delta_k$'s of the previous MPC step to obtain an initial guess for the current $\rho_k$'s. Furthermore, we could also use an iterative procedure in which we first select values for the $\rho_k$'s, compute the optimal solution, use the resulting $\delta_k$'s to determine new values for the $\rho_k$'s, after which we can again compute the optimal solution, and so.
Also note that the assumption on the relative lengths ($\delta_k = \rho_k \bar{\delta}$ for all $k$) is only used to simplify the objective function; it will not be included explicitly in the linear programming problem. So the variables in this problem are still $x_q$ and $x_\delta$, but the objective only depends on $x_q$. As a consequence, the optimal $\delta_k$'s will in general not satisfy the assumption on the relative lengths (see e.g. Example 5.1). $\diamond$

# 5 Application: Optimal traffic signal control

## 5.1 Optimal traffic signal control

In order to illustrate the effectiveness of Proposition 4.1 we shall use the different approaches presented in this paper to design an optimal switching time sequence for a traffic signal controlled intersection and compare the results.

Consider an intersection of two two-way streets (see Figure 1) with lanes $L_i$ and a traffic signal $T_i$ on each corner of the intersection ($i = 1, 2, 3, 4$). The switching time sequence for the

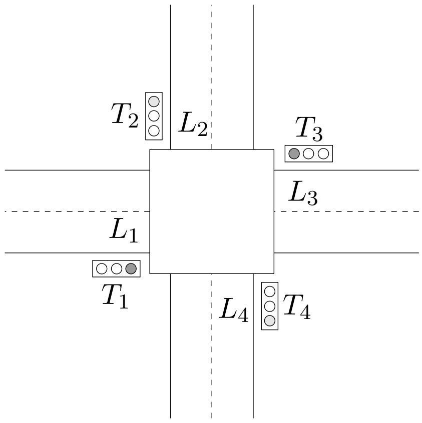

Figure 1: A traffic signal controlled intersection of two two-way streets.

intersection is given in Table 1. Since queue lengths can never become negative and since all the cars can leave a queue provided that we make the length of the green phase large enough, we have $b_k^{\rm ls} = 0$ for all $k$. We assume that there is no saturation at the upper level, either due to the fact that there is enough buffer space before the traffic signal in each lane or due to the fact that we impose additional maximal queue length conditions such that $q_{\max,k} \leqslant b_k^{\rm us}$.

In order to obtain a model that is amenable to mathematical analysis, we shall make two extra assumptions (see also [8]):

- the queue lengths are continuous variables,
- the average arrival and departure rates of the cars are constant or slowly time-varying.

These assumptions deserve a few remarks:

- Recall that the main purpose is to compute optimal traffic signal switching time sequences. Designing optimal switching time sequences is only useful if the arrival and departure rates of vehicles at the intersection are high since then the queue lengths will in general also be large and then approximating the queue lengths by continuous variables will introduce only small errors.
- If we keep in mind that one of the main purposes of the model that we shall derive, is the design of optimal traffic signal switching time sequences, then assuming that the average arrival and departure rates are constant is not a serious restriction provided that we use an MPC approach in which we can regularly update the estimates of the arrival and departure rates and of the state of the system.

Let $\alpha_i^{\rm a}$ be the average arrival rate of cars in lane $L_i$, and let $\alpha_i^{\rm d,green}$ and $\alpha_i^{\rm d,amber}$ be the departure rates of cars in lane $L_i$ when the traffic signal $T_i$ is green respectively amber. If we define

$$\alpha_{i,k} = \begin{cases} \alpha_i^{\rm a} & \text{if } T_i \text{ is red in } (t_k, t_{k+1}) \\ \alpha_i^{\rm a} - \alpha_i^{\rm d,green} & \text{if } T_i \text{ is green in } (t_k, t_{k+1}) \\ \alpha_i^{\rm a} - \alpha_i^{\rm d,amber} & \text{if } T_i \text{ is amber in } (t_k, t_{k+1}) \end{cases}$$

| Period | $T_1, T_3$ | $T_2, T_4$ |
|---|---|---|
| $t_0$–$t_1$ | red | green |
| $t_1$–$t_2$ | red | amber |
| $t_2$–$t_3$ | green | red |
| $t_3$–$t_4$ | amber | red |
| $t_4$–$t_5$ | red | green |
| $t_5$–$t_6$ | red | amber |
| $\vdots$ | $\vdots$ | $\vdots$ |

Table 1: The traffic signal switching scheme.

for all $i, k$, then the relation between the switching time instants and the queue lengths is described by a system of equations of the form

$$\frac{dq_i(t)}{dt} = \begin{cases} \alpha_{i,k} & \text{if } b^{\text{ls}}_{i,k} < q_i(t) \\ 0 & \text{otherwise.} \end{cases}$$

So the system can be considered as a first order linear hybrid system with lower saturation only. Hence, we can use the techniques presented in Sections 3.2 and 4 to compute optimal and suboptimal traffic signal switching schemes.

In the simple traffic signal set-up discussed above we did not make a distinction between cars that turn left, right or that go straight ahead. However, the approach presented in this paper can also be applied to more complex set-ups or more complex traffic signal switching schemes for single intersections such as e.g. the one depicted in Figure 2 which consists of four main phases with amber phases in between where in the first main phase cars on the north-south axis can go straight ahead or turn right, in the next main phase they can turn left, and in the next two main phases the same process is repeated for the traffic on the east-west axis.

## 5.2 Worked example

The following traffic signal control example illustrates that using Proposition 4.1 leads to efficient computation of optimal switching time sequences and that the approximations introduced in Section 4.2 lead to reasonably good suboptimal solutions. Since we are mainly interested in the computation times, we will consider only one step of the MPC algorithm.

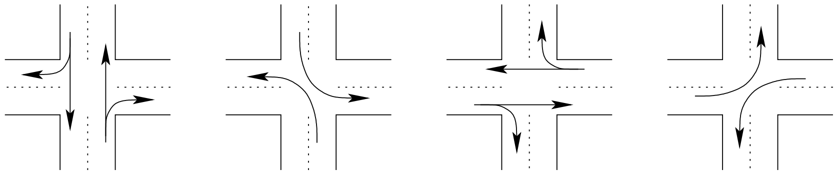

Figure 2: The four main phases of a more complex traffic signal switching scheme. The arrows indicate possible directions for the cars that receive a green signal.

All times will be expressed in seconds and all rates in vehicles per second. The numerical results will be given up to 2 decimal places.

**Example 5.1** Consider the intersection of Figure 1 with the switching scheme of Table 1 and with the following data: $N_\mathrm{p} = 14$, $N_\mathrm{c} = 8$, $\alpha_1^\mathrm{a} = 0.23$, $\alpha_2^\mathrm{a} = 0.12$, $\alpha_3^\mathrm{a} = 0.19$, $\alpha_4^\mathrm{a} = 0.11$, $\alpha_1^\mathrm{d,green} = 0.50$, $\alpha_2^\mathrm{d,green} = \alpha_4^\mathrm{d,green} = 0.35$, $\alpha_3^\mathrm{d,green} = 0.45$, $\alpha_1^\mathrm{d,amber} = \alpha_3^\mathrm{d,amber} = 0.03$, $\alpha_2^\mathrm{d,amber} = \alpha_4^\mathrm{d,amber} = 0.02$, $q_0 = [\,17\ \ 12\ \ 14\ \ 8\,]^T$ and $q_{\max,k} = [\,20\ \ 15\ \ 20\ \ 15\,]^T$ for all $k$. Since a green-amber-red cycle consists of four consecutive phases (see Table 1) we set $K_\mathrm{c} = 4$.

We want to compute a traffic signal switching sequence $t_0, t_1, \ldots, t_{N_\mathrm{c}-1}$ that minimizes $J_1$ with $w = [2\ 1\ 2\ 1]^T$. The minimum and maximum length of the green phases are respectively 9 and 90. Note that for the simple setup of this example and for the objective function $J_1$ it does not make sense to consider a varying amber duration since during the amber phases the average queue length always increases, which implies that the optimal duration of the amber phases in this case will always be equal to the given lower bound for the amber phase. Therefore, we fix the length of the amber phase by setting the minimal and the maximal length of the amber phases equal to 3.

We have computed an optimal switching interval vector $x^*_{\delta,\mathrm{elcp}}$ using the ELCP method, a suboptimal switching vector $x^*_{\delta,\mathrm{nlcon}}$ using constrained optimization with nonlinear constraints[9], and a suboptimal solution $x^*_{\delta,\mathrm{penalty}}$ using constrained optimization[10] with a quadratic penalty function for queue lengths that exceed $q_{\max,k}$. Based on Propositions 4.1 and 4.2 we have computed a solution $x^*_{\delta,\mathrm{relaxed}}$ that minimizes $J_1$ for the relaxed problem $\tilde{\mathcal{P}}$ and a solution $x^*_{\delta,\mathrm{approx}}$ that minimizes the approximate objective function $\tilde{J}_1$ for the relaxed problem $\tilde{\mathcal{P}}$. Finally, we computed a switching interval vector $x^*_{\delta,\mathrm{lp}}$ that minimizes $\hat{J}_1$ for the relaxed problem $\tilde{\mathcal{P}}$ with the affine objective function obtained by assuming that for the east-west axis the length of the green phases is 1.5 times the length of the red phases and 10 times the length of the amber phases (Note that this is just a rough guess).

We have used the sequential quadratic programming function `e04ucc` of the NAG C Library for the nonlinear optimizations. To solve the linear programming problem we have used the function `e04mfc` of the NAG C library, which uses an active set method.

In Table 2 we have listed the value of the objective functions $J_1$, $\tilde{J}_1$ and $\hat{J}_1$ for the various switching interval vectors and the CPU time needed to compute the switching interval vectors on a Pentium II 300 MHz PC running Linux and with 64 MB RAM. The CPU time values listed in the table are average values over 10 experiments[11]. All the routines used in the computations either have been implemented in C or were compiled to object code. As a consequence, all the CPU times can be considered as a measure for the number of floating point operations that were needed to compute the various (sub)optimal switching interval vectors.

Note that the optimal values of $J_1$ and $\tilde{J}_1$ differ by about 5 %, so that in this case the

[9] We give the best solution over 20 runs with random initial points. Only 14 runs resulted in a feasible solution. For these 14 runs the mean of the objective values of the local minima returned by the minimization routine was 48.50 with a standard deviation of 3.38.

[10] We give the best solution over 10 runs with random initial points. For the 9 runs that resulted in a feasible solution, the mean of the objective values was 46.45 with a standard deviation of 0.08.

[11] For $x^*_{\delta,\mathrm{nlcon}}$ we have listed the CPU time needed for 20 runs with random initial points and for $x^*_{\delta,\mathrm{penalty}}$ we have listed the CPU time needed for 10 runs with random initial points (see also footnotes 9 and 10). Note however that even for a single run the average CPU time needed for these solutions is much higher that the CPU time needed for the $x^*_{\delta,\mathrm{relaxed}}$ solutions.

| $x_\delta^*$ | $J_1(x_\delta^*)$ | $\tilde{J}_1(x_\delta^*)$ | $\hat{J}_1(x_\delta^*)$ | CPU time |
|---|---|---|---|---|
| $x_{\delta,\mathrm{elcp}}^*$ | 46.41 | 49.05 | 53.66 | 64 619.07 |
| $x_{\delta,\mathrm{nlcon}}^*$ | 46.41 | 49.05 | 53.66 | 216.71 |
| $x_{\delta,\mathrm{penalty}}^*$ | 46.41 | 49.05 | 53.66 | 29.71 |
| $x_{\delta,\mathrm{relaxed}}^*$ | 46.41 | 49.05 | 53.66 | 0.36 |
| $x_{\delta,\mathrm{approx}}^*$ | 46.41 | 49.05 | 53.66 | 3.56 |
| $x_{\delta,\mathrm{lp}}^*$ | 46.63 | 49.12 | 53.62 | 0.16 |

Table 2: The values of the objective functions $J_1$, $\tilde{J}_1$ and $\hat{J}_1$, and the CPU time needed to compute the (sub)optimal switching vectors of Example 5.1.

optimal value of $\tilde{J}_1$ is indeed a reasonably good approximation of the optimal value of $J_1$. While computing $x_{\delta,\mathrm{relaxed}}^*$ we only have $N_\mathrm{c}$ optimization variables (i.e., the $\delta_k$'s, since the $q_k$'s do not appear in the objective function and since they can be eliminated from the constraints). However, for $x_{\delta,\mathrm{approx}}^*$ we have $N_\mathrm{c} + MN_\mathrm{p}$ optimization variables (i.e., the $\delta_k$'s and the components of the $q_k$'s since in this case the $q_k$'s appear in the objective function and can thus not be eliminated). This is one of the reasons why the computation of $x_{\delta,\mathrm{relaxed}}^*$ requires less CPU time than the computation of $x_{\delta,\mathrm{approx}}^*$. Additional numerical experiments and simulations can be found in [6].

In this example the ELCP solution is only given as a reference since the CPU time needed to compute the optimal switching interval vector using the ELCP algorithm of [7] increases exponentially as $M$, $N_\mathrm{p}$ or $N_\mathrm{c}$ increase (see also [6]). This implies that the ELCP approach should never be used in practice, but one of the other approaches should be used instead.

If we look at Table 2 then we see that the $x_{\delta,\mathrm{relaxed}}^*$ solution — which is based on Proposition 4.1 — is clearly the most interesting. If we take the trade-off between optimality and efficiency into account, the $x_{\delta,\mathrm{relaxed}}^*$ solution outperforms the solutions obtained using the other approaches (see also [6]).

If we use an MPC approach then the computation time required for the $x_{\delta,\mathrm{relaxed}}^*$ solution is less than the minimum lower bound for the phase lengths, which implies that we can first measure the queue lengths at $t_0$ and start the computation at time $t_0$. In that way we can use the exact initial state $q_0$. Note that using the exact initial state $q_0$ (or a good estimate) also introduces a kind of feedback in the control loop. □

# 6 Conclusions and future research

We have considered the determination of optimal switching time sequences for a class of first order linear hybrid systems subject to saturation. First we have introduced the Extended Linear Complementarity Problem (ELCP) and indicated how it can be used to describe the set of feasible system trajectories for a first order linear hybrid system with saturation. Optimization over the solution set of the ELCP then yields the optimal switching time sequence. Since the ELCP is NP-hard, we have also discussed several other techniques to compute optimal and suboptimal switching time sequences for first order linear hybrid systems subject to saturation at the lower level only. We have shown that if the objective function is a monotonically nondecreasing function of the queue lengths, then the optimal switching problem can be transformed into an optimization problem with a convex feasible set and then the optimal switching time sequence can be computed more efficiently. By making some approximations, the optimal switching problem can even be transformed into a linear programming problem. We have illustrated these approaches by computing (sub)optimal switching time sequences for a traffic signal controlled intersection. Since the time required for the computations using the most efficient approach is less than the minimum time between two consecutive switchings, our method can be used in a model predictive control framework in which the model of the system and the optimal switching sequence are re-estimated or re-computed after each switching.

In this paper we have derived methods to optimize quantitative performance measures such as average or worst case waiting times and queue lengths for a linear hybrid system with saturation. If we are more interested in qualitative properties such as e.g. safety, we could use the techniques presented in [18].

An important topic for future research is the extension of the results obtained in this paper to networks of dependent queues, i.e. a situation where the outputs of some queues will be connected to the inputs of some other queues. If we use an MPC strategy in combination with a *decentralized control* solution, we can apply still the approach given in this paper: if we know or measure all routing rates[12] and all traveling times from one queue to another, we can use measurements from one queue to predict the arrival rates at the other queues. Other topics for further research include: development of other efficient algorithms and/or approximations to compute optimal switching time sequences for first order linear hybrid systems with saturation, investigation of the use of the ELCP to model and to control other classes of hybrid systems, and extension of the results presented in this paper to more general classes of hybrid systems.

**Acknowledgment**

We would like to thank the anonymous referees for their useful suggestions and comments, which have certainly contributed to improving the presentation of the paper.

## References

[1] P.J. Antsaklis and A. Nerode, eds., "Special issue on hybrid systems," *IEEE Transactions on Automatic Control*, vol. 43, no. 4, Apr. 1998.

[2] V.D. Blondel and J.N. Tsitsiklis, "Complexity of stability and controllability of elementary hybrid systems," *Automatica*, vol. 35, no. 3, pp. 479–489, Mar. 1999.

[3] E.F. Camacho and C. Bordons, *Model Predictive Control in the Process Industry.* Berlin, Germany: Springer-Verlag, 1995.

[4] D.W. Clarke, C. Mohtadi, and P.S. Tuffs, "Generalized predictive control – Part I. The basic algorithm," *Automatica*, vol. 23, no. 2, pp. 137–148, Mar. 1987.

[5] R.W. Cottle, J.S. Pang, and R.E. Stone, *The Linear Complementarity Problem.* Boston, Massachusetts: Academic Press, 1992.

[6] B. De Schutter, "Optimal control of a class of linear hybrid systems with saturation: Addendum," Tech. rep. bds:99-03a, Control Systems Engineering, Fac. of Information Technology and Systems, Delft University of Technology, Delft, The Netherlands, Oct. 1999.

[12]The routing rates are the numbers of cars, the amount of fluid, ... that will be routed from the output of one queue to the input of another queue.

[7] B. De Schutter and B. De Moor, "The extended linear complementarity problem," *Mathematical Programming*, vol. 71, no. 3, pp. 289–325, Dec. 1995.

[8] B. De Schutter and B. De Moor, "Optimal traffic light control for a single intersection," *European Journal of Control*, vol. 4, no. 3, pp. 260–276, 1998.

[9] C.E. García, D.M. Prett, and M. Morari, "Model predictive control: Theory and practice – A survey," *Automatica*, vol. 25, no. 3, pp. 335–348, May 1989.

[10] N.H. Gartner, J.D.C. Little, and H. Gabbay, "Simultaneous optimization of offsets, splits, and cycle time," *Transportation Research Record*, no. 596, pp. 6–15, 1976.

[11] T.A. Henzinger and S. Sastry, eds., *Hybrid Systems: Computation and Control*, vol. 1386 of *Lecture Notes in Computer Science*, Berlin, Germany: Springer-Verlag, 1998. (Proceedings of the First International Workshop on Hybrid Systems: Computation and Control (HSCC'98), Berkeley, California, Apr. 1998).

[12] P. Kachroo and K. Özbay, "Solution to the user equilibrium dynamic traffic routing problem using feedback linearization," *Transportation Research Part B*, vol. 32, no. 5, pp. 343–360, June 1998.

[13] A.D. May, *Traffic Flow Fundamentals.* Englewood Cliffs, New Jersey: Prentice-Hall, 1990.

[14] A.S. Morse, C.C. Pantelides, S. Sastry, and J.M. Schumacher, eds., "Special issue on hybrid systems," *Automatica*, vol. 35, no. 3, Mar. 1999.

[15] E.S. Park, J.H. Lim, I.H. Suh, and Z. Bien, "Hierarchical optimal control of urban traffic networks," *International Journal of Control*, vol. 40, no. 4, pp. 813–829, Oct. 1984.

[16] M.G. Singh and H. Tamura, "Modelling and hierarchical optimization for oversaturated urban road traffic networks," *International Journal of Control*, vol. 20, no. 6, pp. 913–934, 1974.

[17] F.W. Vaandrager and J.H. van Schuppen, eds., *Hybrid systems: Computation and Control*, vol. 1569 of *Lecture Notes in Computer Science*, Berlin, Germany: Springer, 1999. (Proceedings of the Second International Workshop on Hybrid Systems: Computation and Control (HSCC'99), Berg en Dal, The Netherlands, Mar. 1999).

[18] H. Wong-Toi, "The synthesis of controllers for linear hybrid automata," in *Proceedings of the 36th IEEE Conference on Decision and Control*, San Diego, California, pp. 4607–4612, Dec. 1997.

# Optimal control of a class of linear hybrid systems with saturation: Addendum

Bart De Schutter

**Abstract**

In this addendum we give some extra propositions, proofs and results of numerical experiments in connection with the design of (sub)optimal switching time sequences for a class of first order linear hybrid systems with saturation that we have introduced in the paper "Optimal control of a class of linear hybrid systems with saturation" [A-1] (*SIAM Journal on Optimization and Control*, vol. 38, no. 3, pp. 835–851, 2000).

All references in this addendum that are not preceded by a capital letter A, B or C refer to sections, equations, etc. of the paper [A-1].

# A Additional remarks for [A-1]

## A.1 Model

The model derived in [A-1] can accommodate varying amber durations. However, in many countries the amber time is fixed by regulation (e.g. to 3 s in France). If we assume that the duration of the amber phase is fixed, then we can adapt our model and reduce the number of variables (see also [8]).

## A.2 Systems with varying rate functions

In [A-1] we have already explained that the MPC approach also allows us to deal with slowly-varying rate functions by updating the model after each switching and re-computing the optimal switching sequence.

Even if the rates are assumed to be non-constant within one MPC step, we can still use our approach if we approximate the time-varying rate functions by piecewise constant functions. Although in general we do not know the exact behavior of these functions in advance the behavior can often be predicted on the basis of historical data and measurements. Also note that we do not know the lengths of the phases in advance. In order to determine the average rates for each phase, we could therefore first assume that all phases have equal length. Then we compute an optimal or suboptimal switching time sequence and use the result to get better estimates of the lengths of the phases and thus also of the average queue length growth rates in each phase, which can then be used as the input for another optimization run. If necessary we could repeat this process in an iterative way.

Also note that in practice there is always some uncertainty and variation in time of the queue length growth rates, which makes that in general computing the exact optimal switching time sequence is utopian. Moreover, in practice we are more interested in quickly obtaining a good approximation of the optimal switching time sequence than in spending a large amount of time to obtain the exact optimal switching time sequence.

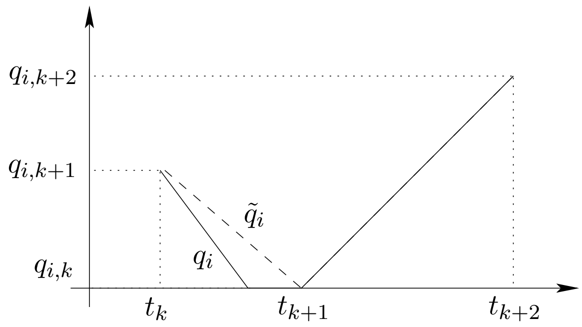


Figure A.1: The functions $q_i$ (full line) and $\tilde{q}_i$ (dashed line) for a queue with a decrease phase and a subsequent positive growth phase and without saturation at an upper level. Note that in the growth phase the functions $q_i$ and $\tilde{q}_i$ coincide.

### A.3 Approximations of the objective functions

Let $l \in \{1, 4\}$. The value of the objective functions $J_l$ and $\tilde{J}_l$ introduced in Sections 3.2 and 4.2 depends on the surface under the functions $q_i$ and $\tilde{q}_i$ respectively. For a traffic signal controlled intersection where the traffic signals alternate between green and red (with a short amber phase in between) we will usually have a queue length evolution that is similar to the one represented in Figure A.1. An optimal traffic signal switching scheme implies the absence of long periods in which no cars wait in one lane (i.e. $q_i(t) = 0$) while in the other lanes the queue lengths increase. So in that case the surface under the function $\tilde{q}_i$ will be a reasonable approximation of the surface under the function $q_i$ and then the optimal value of $\tilde{J}_l$ will be a reasonably good approximation of the optimal value of $J_l$.

Since $\tilde{q}_i$ is a piecewise-affine with breakpoints $(t_k, q_{i,k})$ for $k = 0, 1, \ldots, N_\mathrm{p}$, we have

$$\int_{t_k}^{t_{k+1}} \tilde{q}_i(t, x_q, x_\delta) dt = \frac{\delta_k}{2}(q_{i,k} + q_{i,k+1})$$

and thus

$$\tilde{J}_1(x_q, x_\delta) = \sum_{i=1}^{M} \left( \frac{w_i}{2\,(\delta_0 + \delta_1 + \ldots + \delta_{N_\mathrm{p}-1})} \sum_{k=0}^{N_\mathrm{p}-1} \delta_k (q_{i,k} + q_{i,k+1}) \right) .$$

## B Additional numerical experiments and simulations for the example of Section 5.2

The switching interval vectors of the example of Section 5.2 are given by

$$\begin{aligned}
x^*_{\delta,\mathrm{elcp}} &= [\,10.04\ \ 3.00\ \ 38.75\ \ 3.00\ \ 39.88\ \ 3.00\ \ 70.00\ \ 3.00\,]^T \\
x^*_{\delta,\mathrm{nlcon}} &= [\,10.04\ \ 3.00\ \ 30.75\ \ 3.00\ \ 39.88\ \ 3.00\ \ 70.94\ \ 3.00\,]^T \\
x^*_{\delta,\mathrm{penalty}} &= [\,10.04\ \ 3.00\ \ 30.75\ \ 3.00\ \ 39.88\ \ 3.00\ \ 70.94\ \ 3.00\,]^T \\
x^*_{\delta,\mathrm{relaxed}} &= [\,10.04\ \ 3.00\ \ 38.75\ \ 3.00\ \ 39.88\ \ 3.00\ \ 70.94\ \ 3.00\,]^T \\
x^*_{\delta,\mathrm{approx}} &= [\,10.04\ \ 3.00\ \ 38.75\ \ 3.00\ \ 39.88\ \ 3.00\ \ 70.94\ \ 3.00\,]^T
\end{aligned}$$

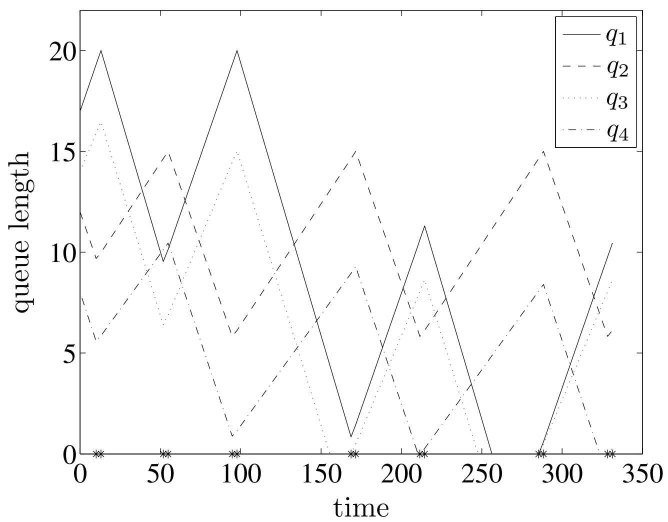


Figure A.2: The queue lengths in the various lanes as a function of time for the traffic signal switching sequence that corresponds to the switching interval vector $x^*_{\delta,\text{elcp}}$ of the example of Section 5.2. The * signs on the time axis correspond to the switching time instants.

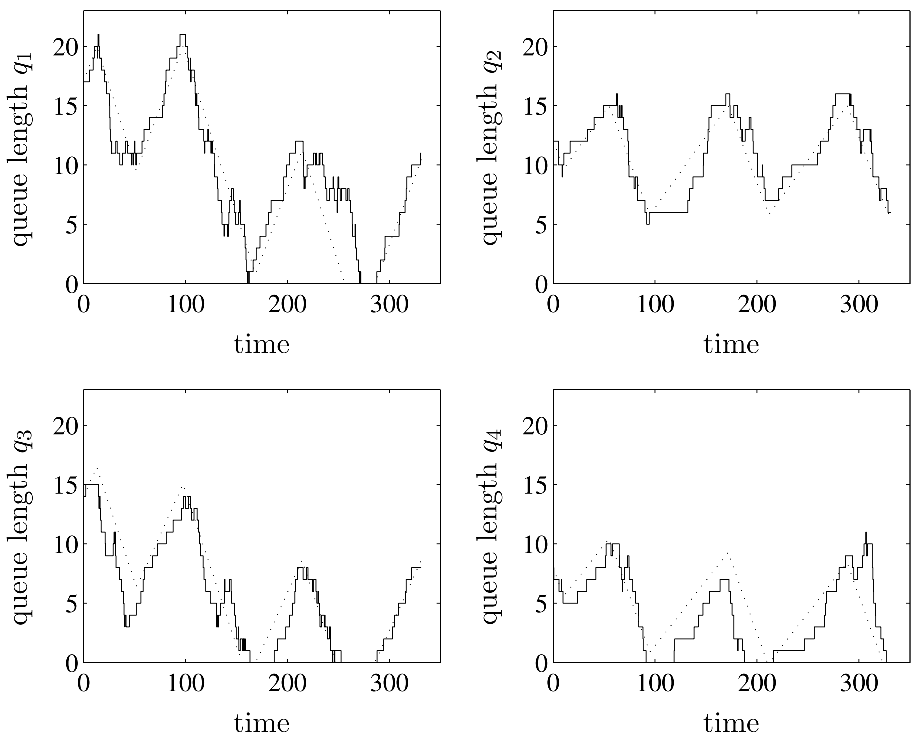


Figure A.3: The queue lengths in the various lanes as a function of time for an integer queue length simulation for the traffic signal switching sequence that corresponds to the switching interval vector $x^*_{\delta,\text{elcp}}$. The integer queue length functions are plotted in full lines and their continuous approximations in dotted lines.

| | $J_1(x^*_\delta)$ | | | CPU time | | |
|---|---|---|---|---|---|---|
| $x^*_\delta$ | $N_c = 4$ | $N_c = 6$ | $N_c = 10$ | $N_c = 4$ | $N_c = 6$ | $N_c = 10$ |
| $x^*_{\delta,\mathrm{elcp}}$ | 70.69 | 54.14 | – | 4.10 | 485.04 | – |
| $x^*_{\delta,\mathrm{nlcon}}$ | 70.69 | 54.14 | 46.41 | 115.31 | 169.59 | 282.82 |
| $x^*_{\delta,\mathrm{penalty}}$ | 70.69 | 54.14 | 46.41 | 9.59 | 20.92 | 39.28 |
| $x^*_{\delta,\mathrm{relaxed}}$ | 79.69 | 54.14 | 46.41 | 0.16 | 0.26 | 0.50 |
| $x^*_{\delta,\mathrm{approx}}$ | 70.81 | 56.24 | 53.66 | 2.73 | 2.96 | 3.73 |
| $x^*_{\delta,\mathrm{lp}}$ | 70.69 | 54.14 | 47.54 | 0.14 | 0.15 | 0.19 |

Table A.1: The values of the objective functions $J_1$ and the CPU time needed to compute the (sub)optimal switching interval vectors of the example of Section 5.2 for $N_c = 4$, 6 and 10.

$$x^*_{\delta,\mathrm{lp}} = [\, 10.04 \;\; 3.00 \;\; 38.75 \;\; 3.00 \;\; 38.81 \;\; 3.00 \;\; 68.89 \;\; 3.00 \,]^T \ .$$

The evolution of the queue lengths for the optimal switching interval vector $x^*_{\delta,\mathrm{elcp}}$ is represented in Figure A.2. In Figure A.3 we have plotted the results of an integer queue length simulation for the traffic signal switching strategy that corresponds to the optimal switching interval vector $x^*_{\delta,\mathrm{elcp}}$. The effective average queue length over all lanes for this simulation is 45.17.

In order to show how the control horizon $N_c$ influences the performance of the methods presented in [A-1] we have computed optimal and suboptimal switching time intervals for three different values of $N_c$: 4, 6 and 10. The other data and parameters have the same values as in Section 5.2. In Table A.1 we have given the optimal value of the objective function $J_1$ and the CPU time needed to compute the solution. We have not computed the ELCP solution for $N_c = 10$, since this would require too much CPU time.

We see again that the $x^*_{\delta,\mathrm{approx}}$ solution offers the best trade-off between efficiency and optimality.

While performing numerical experiments for the example of Section 5.2 and for similar examples, we noticed the following:

- The determination of the minimum value of $J_1$ over the solution set of the ELCP is a well-behaved problem in the sense that using a local minimization routine starting from different initial points almost always yields the same numerical result (within a certain tolerance). In a typical experiment in which for each face we computed the minimal value of the objective function for 20 random starting points, for almost every face 10 or more decimal places of the final objective function were the same. Therefore, we have only considered one run with an arbitrary random initial point for each face for the ELCP solution in Table 2.

- In order to obtain a good approximation to the optimal switching time vector using nonlinear constrained optimization or a penalty function approach, it is necessary to run the local minimization algorithm several times each time with a different initial starting point. In general a local minimization run for the approach that uses nonlinear constraints requires less time than a run for the penalty function approach. However, the nonlinear constraints approach requires more different starting points to obtain a good approximation of the global optimum than the penalty function approach.

- Apart from the quadratic penalty function defined by

$$F_{\text{penalty}} = 10\,000 \sum_{k=1}^{N_{\text{p}}} \sum_{i=1}^{M} \Big( \max \big(0,\, (q_k)_i - (q_{\text{max,k}})_i\big)\Big)^2 , \tag{A.1}$$

we have also used linear, exponential and mixed penalty functions in the penalty function approach. However, for the applications considered here the penalty function defined by (A.1) leads to the best performance.

- The relaxed problem (which has a convex feasible set) is much easier to solve using multi-start local optimization than the original problem (which has a non-convex feasible set). In a typical experiment for $x^*_{\delta,\text{relaxed}}$ with 20 random starting points the first 11 decimal places of the final objective function $J_1$ always had the same value. For $x^*_{\delta,\text{approx}}$ the first 9 decimal places always had the same value. This implies that in practice performing one run with an random starting point is sufficient to obtain the globally optimal solution for $x^*_{\delta,\text{relaxed}}$ and $x^*_{\delta,\text{approx}}$.

# C A generalization

## C.1 A more general class of systems

In this section we show that the results of Sections 3 and 4 can be extended to a more general class of systems that can be described by equations of the form

$$q_{k+1} = \max\big(\min\left(A_k q_k + B_k u_k,\, b_k^{\text{us}}\right),\, b_k^{\text{ls}}\big) \tag{A.2}$$

for $k = 0, 1, 2, \ldots$ where $A_k \in \mathbb{R}^{M\times M}$, $B_k \in \mathbb{R}^{M\times L}$, and $u_k \in \mathbb{R}^L$ for some integer $L$. Note that (4) is a special case of (A.2) with $A_k = I$, $B_k = \alpha_k$, $u_k = \delta_k$ and $L = 1$.

Note that the description (A.2) does *not* correspond to a switched continuous time system the behavior of which is described by

$$\frac{dq_i(t)}{dt} = \begin{cases} (A_k q_i(t) + B_k u_k)_i & \text{if } b_{i,k}^{\text{ls}} < q_i(t) < b_{i,k}^{\text{us}} \\ 0 & \text{otherwise,} \end{cases} \tag{A.3}$$

for $t \in (t_k, t_{k+1})$ and for $i = 1, 2, \ldots, M$. Indeed, for $M = 1$ (A.3) results in

$$q_{k+1} = \begin{cases} \max\Big(\min\left((q_k + A_k^{-1} B_k u_k) e^{A_k(t_{k+1}-t_k)} - A_k^{-1} B_k u_k,\, b_k^{\text{us}}\right),\, b_k^{\text{ls}}\Big) & \text{if } A_k \neq 0 \\ \max\Big(\min\left(q_k + B_k u_k (t_{k+1} - t_k),\, b_k^{\text{us}}\right),\, b_k^{\text{ls}}\Big) & \text{if } A_k = 0\,. \end{cases}$$

For $M > 1$ the relation between $q_{k+1}$, $q_k$ and $t_{k+1} - t_k$ is even more complex.

For the class of systems described by (A.2) the optimization problem that has to be solved in each major MPC step is given by:

$$\underset{u_0, u_1, \ldots, u_{N_{\text{c}}-1}}{\text{minimize}} \quad J \tag{A.4}$$

subject to

$$u_k = u_{k-K_c} \quad \text{for } k = N_c, N_c+1, \ldots, N_p - 1, \tag{A.5}$$

$$u_{\min,k} \leqslant u_k \leqslant u_{\max,k} \quad \text{for } k = 0, 1, \ldots, N_p - 1, \tag{A.6}$$

$$q_{\min,k} \leqslant q_{k+1} \leqslant q_{\max,k} \quad \text{for } k = 0, 1, \ldots, N_p - 1 \tag{A.7}$$

$$z_{k+1} = \min(A_k q_k + B_k u_k,\, b_k^{\mathrm{us}}) \quad \text{for } k = 0, 1, \ldots, N_p - 1, \tag{A.8}$$

$$q_{k+1} = \max(z_{k+1},\, b_k^{\mathrm{ls}}) \quad \text{for } k = 0, 1, \ldots, N_p - 1. \tag{A.9}$$

where $u_{\min,k}$ and $u_{\max,k}$ are respectively the minimum and the maximum values of $u_k$.

Using the same reasoning as in Section 3.3 we can show that the system (A.5) – (A.9) can be reformulated as an ELCP of the form

$$Ax_q + Bx_z + Cx_u + d \geqslant 0 \tag{A.10}$$

$$Ex_q + Fx_z + g \geqslant 0 \tag{A.11}$$

$$Hx_q + Kx_u + l \geqslant 0 \tag{A.12}$$

$$(Ax_q + Bx_z + Cx_u + d)^T (Ex_q + Fx_z + g) = 0 \ , \tag{A.13}$$

for appropriately defined matrices $A$, $B$, $C$, $E$, $F$, $H$, $K$ and vectors $d$, $g$, $l$ and with

$$x_u = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix} .$$

If we introduce additional linear equality or inequality constraints on the components of $x_u$, we still obtain an ELCP. The additional linear inequality constraints lead to extra inequalities in (A.12), and the additional linear equality constraints lead to an extra equation of the form $Px_u + q = 0$, which also fits in the ELCP framework.

Now we can determine optimal input sequences using the ELCP approach or using multi-start local optimization.

## C.2 Optimal and suboptimal input sequences for systems with saturation at a lower level only

In this section we consider systems with saturation at the lower level only. So $b_{i,k}^{\mathrm{us}}$ is equal to $\infty$ for all $i, k$, or equivalently $(q_{\max,k})_i \leqslant b_{i,k}^{\mathrm{us}}$ for all $i, k$. We also assume that $q_{\min,k} \leqslant b_k^{\mathrm{ls}}$ for all $k$, i.e. we do not impose extra lower bound conditions on the queue lengths. Furthermore, we assume that $(A_k)_{ij} \geqslant 0$ for all $i, j, k$. Note that the latter assumption always holds for the class of first order linear hybrid systems that has been introduced in Section 3.1 since for this class we have $A_k = I$ for all $k$. The problem (A.4) – (A.9) then reduces to

$$\underset{x_u}{\text{minimize}} \ J$$

subject to

$$u_k = u_{k-K_c} \quad \text{for } k = N_c, N_c+1, \ldots, N_p - 1,$$

$$u_{\min,k} \leqslant u_k \leqslant u_{\max,k} \quad \text{for } k = 0, 1, \ldots, N_{\mathrm{p}} - 1,$$
$$q_{k+1} \leqslant q_{\max,k} \quad \text{for } k = 0, 1, \ldots, N_{\mathrm{p}} - 1$$
$$q_{k+1} = \max(A_k q_k + B_k u_k,\, b_k^{\mathrm{ls}}) \quad \text{for } k = 0, 1, \ldots, N_{\mathrm{p}} - 1.$$

We call this problem $\mathcal{P}_{\mathrm{g}}$. We define the "relaxed" problem $\tilde{\mathcal{P}}_{\mathrm{g}}$ corresponding to the problem $\mathcal{P}_{\mathrm{g}}$ as:

$$\underset{x_q, x_u}{\text{minimize}}\ J$$

subject to

$$u_k = u_{k-K_{\mathrm{c}}} \quad \text{for } k = N_{\mathrm{c}}, N_{\mathrm{c}} + 1, \ldots, N_{\mathrm{p}} - 1,$$
$$u_{\min,k} \leqslant u_k \leqslant u_{\max,k} \quad \text{for } k = 0, 1, \ldots, N_{\mathrm{p}} - 1,$$
$$q_{k+1} \leqslant q_{\max,k} \quad \text{for } k = 0, 1, \ldots, N_{\mathrm{p}} - 1$$
$$q_{k+1} \geqslant A_k q_k + B_k u_k \quad \text{for } k = 0, 1, \ldots, N_{\mathrm{p}} - 1,$$
$$q_{k+1} \geqslant b_k^{\mathrm{ls}} \quad \text{for } k = 0, 1, \ldots, N_{\mathrm{p}} - 1.$$

Note that $x_q$ and $x_u$ are not directly coupled any more. The set of feasible solutions of $\tilde{\mathcal{P}}_{\mathrm{g}}$ is a convex set, whereas the set of feasible solutions of $\mathcal{P}_{\mathrm{g}}$ is in general not convex. Therefore, the relaxed problem $\tilde{\mathcal{P}}_{\mathrm{g}}$ will in general be easier to solve than the problem $\mathcal{P}_{\mathrm{g}}$.

The following proposition shows that for monotonically nondecreasing objective functions any optimal solution of the relaxed problem $\tilde{\mathcal{P}}_{\mathrm{g}}$ can be transformed into an optimal solution of the problem $\mathcal{P}_{\mathrm{g}}$.

**Proposition C.1** *Let the objective function $J$ be a monotonically nondecreasing function of $x_q$ and let $(x_q^*, x_u^*)$ be an optimal solution of $\tilde{\mathcal{P}}_{\mathrm{g}}$. If we define $x_q^\sharp$ such that*

$$q_1^\sharp = \max(A_0 q_0 + B_0 u_0^*, b_0^{\mathrm{ls}})$$
$$q_{k+1}^\sharp = \max(A_k q_k^\sharp + B_k u_k^*, b_k^{\mathrm{ls}}) \qquad \text{for } k = 1, 2, \ldots, N - 1.$$

*then $(x_q^\sharp, x_u^*)$ is an optimal solution of the problem $\mathcal{P}_{\mathrm{g}}$.*

**Proof:** This proof is analogous to the proof of Proposition 4.1. The only difference is that now we have to include the fact that $(A_k)_{ij} \geqslant 0$ for all $i, j, k$ in order to prove by induction that $q_k^\sharp \leqslant q_k^*$ for $k = 1, 2, \ldots, N_{\mathrm{p}}$. □

Since the objective functions $J_1$, $J_2$, $J_3$, $J_4$ and $J_5$ do not explicitly depend on $x_q$, we have $J_l(\tilde{x}_q, x_u) = J_l(\hat{x}_q, x_u)$ for any $\tilde{x}_q, \hat{x}_q$ and for $l \in \{1, 2, 3, 4, 5\}$. This implies that $J_1$, $J_2$, $J_3$, $J_4$ and $J_5$ are monotonically nondecreasing functions of $x_q$. So we can use Proposition C.1 to transform the optimal control problem for the objective functions $J_1$ up to $J_5$ into an optimization problem with a convex feasible set.

The optimal solution of problem $\tilde{\mathcal{P}}_{\mathrm{g}}$ will in general not be a feasible solution of $\mathcal{P}_{\mathrm{g}}$, unless $J$ is a monotonically increasing function of $x_q$:

**Proposition C.2** *If $J$ is a monotonically increasing function of $x_q$ then any optimal solution of the relaxed problem $\tilde{\mathcal{P}}_{\mathrm{g}}$ is also an optimal solution of the problem $\mathcal{P}_{\mathrm{g}}$.*

**Proof:** This proof is similar to the proof of Proposition 4.2. □

## Additional references

[A-1] B. De Schutter, "Optimal control of a class of linear hybrid systems with saturation," *SIAM Journal on Control and Optimization*, vol. 39, no. 3, pp. 835–851, 2000.