



This is a repository copy of *Impulsively-controlled systems and reverse dwell time: A linear programming approach*.

White Rose Research Online URL for this paper:  
<http://eprints.whiterose.ac.uk/89653/>

Version: Accepted Version

---

**Article:**

Bauso, D. (2015) Impulsively-controlled systems and reverse dwell time: A linear programming approach. *Nonlinear Analysis: Hybrid Systems*, 16. 40 - 51. ISSN 1751-570X

<https://doi.org/10.1016/j.nahs.2014.11.004>

---

**Reuse**

Unless indicated otherwise, fulltext items are protected by copyright with all rights reserved. The copyright exception in section 29 of the Copyright, Designs and Patents Act 1988 allows the making of a single copy solely for the purpose of non-commercial research or private study within the limits of fair dealing. The publisher or other rights-holder may allow further reproduction and re-use of this version - refer to the White Rose Research Online record for this item. Where records identify the publisher as the copyright holder, users can verify any specific terms of use on the publisher's website.

**Takedown**

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing [eprints@whiterose.ac.uk](mailto:eprints@whiterose.ac.uk) including the URL of the record and the reason for the withdrawal request.



[eprints@whiterose.ac.uk](mailto:eprints@whiterose.ac.uk)  
<https://eprints.whiterose.ac.uk/>

# Impulsively-controlled systems and reverse dwell time: a linear programming approach<sup>1</sup>

Dario Bauso

*Dipartimento di Ingegneria Chimica, Gestionale, Informatica, Meccanica, Università di Palermo, Italy, and “Research Fellow” at Dipartimento di Matematica, Università di Trento, email: dario.bauso@unipa.it.*

---

## Abstract

We present a receding horizon algorithm that converges to the exact solution in polynomial time for a class of optimal impulse control problems with uniformly distributed impulse instants and governed by so-called reverse dwell time conditions. The cost has two separate terms, one depending on time and the second monotonically decreasing on the state norm. The obtained results have both theoretical and practical relevance. From a theoretical perspective we prove certain geometrical properties of the discrete set of feasible solutions. From a practical standpoint, such properties reduce the computational burden and speed up the search for the optimum thus making the algorithm suitable for on-line implementation in real-time problems. Our approach consists in approximating the optimal impulse control problem via a binary linear programming problem with a totally unimodular constraint matrix. Hence, solving the binary linear programming problem is equivalent to solving its linear relaxation. Then, given the feasible solution from the linear relaxation, we find the optimal solution via receding horizon and local search. Numerical illustrations of a queueing system are performed.

*Keywords:* Impulse Control; Hybrid Systems; Reverse Dwell Time.

---

<sup>1</sup>Conference version [3] in Proc. of the 48th CDC, Shanghai, China, 2009. Research supported by PRIN 20103S5RN3 “Robust decision making in markets and organization”. Part of this work has been conducted during a sabbatical period that the author spent as academic visitor at the Department of Engineering Science, University of Oxford, UK.

## 1. Introduction

Cyber-physical systems are often characterized by shared communication channels. Thus, a main issue is the one of minimizing the “attention” that a control task requires, or which is the same, maximizing the *dwell* time (also called *interexecution* time), i.e., time between two consecutive executions [10]. We reframe the above challenge in the context of optimal impulse control problems (see, e.g., [5, 7, 8, 18, 22, 23] and references therein) with uniformly distributed impulse instants.

The decision problem consists in finding the optimal schedule of the impulses to drive and keep the system in a safe operating interval, while minimizing a function related to the cost of the impulses. The decision variables are thus binary (whether to act with an impulse at a given time instant or not) and this is an element in common with mixed integer predictive control [1], boolean-controlled systems [2], and alphabet control [12]. The problem applies to practical examples such as queueing systems, batch deliveries in inventory systems (see, e.g., [7] Example 3.5), bank account flows in economics [17], impulsively controlled mechanical systems (bouncing balls) (see, e.g., [7] Example 3.2).

We cast the problem within the framework of Input to State Stabilizability (ISS) of impulsively controlled systems with reverse dwell time [15]. Actually, we start from the observation that the problem of driving and keeping the system in a safe operating interval can be reviewed as an ISS problem over an infinite horizon. In this context, it is assumed that the continuous time system has the tendency to become unstable, while the applied impulses have a stabilizing effect upon the system. Under the aforesaid assumption, previous results in [15] establish that the system will be ISS as long as the exerted impulses are separated by a time interval bounded by above. This last condition is known as reverse dwell time condition. We then impose a cost structure on the applied impulses thus turning the basic stabilizability problem into a decision optimization one. The areas of the impulses at the time instants under consideration are predetermined, i.e., one does not choose the area of the impulse but simply whether or not to apply an impulse at a certain point in time. When introducing the optimization problem, we also reduce the horizon window into a finite one as this has the meaning of minimizing the rate for the accumulation of the considered cost. Actually, though ISS has a meaning over an infinite operational horizon, the cost minimization is well-posed only over a finite horizon. This is clear as over an

infinite horizon we would end up with infinite impulses and costs.

The problem at hand differs from other optimal impulse control problems [14] in two main aspects. First, the considered cost has a discrete-time structure which has its roots in the discrete-time nature of the controlled impulses. Second, due to the nature of the reverse dwell time constraints, we can turn the problem into an integer nonlinear program [20].

The present paper improves the results in [3] in at least three directions: i) a simplified reformulation of the problem based on previous results on reverse dwell time impulse control [15], ii) a completely new rearrangement of the receding horizon framework and iii) additional numerical illustrations of a queueing system [11, 19].

We highlight three main contributions that apply to problems with reverse dwell time conditions. First, we provide cost bounds by approximating the optimal impulse problem by a more tractable linear program. Second, we analyze constraints to reduce the computational burden associated to the exploration of the solution domain. Third, we tighten the above bounds via a receding horizon algorithm embedding the above constraints.

The obtained results have both theoretical and practical relevance. From a theoretical perspective we prove certain geometrical properties of the feasible solutions. Actually, the set of feasible solutions is an integral polyhedron. Such properties can be used in practice to reduce the computational burden and speed up the search for the optimum. Indeed, the approximate solution (given by the linear program) provides constraints that reduce the computational burden for the optimal search. Visualizing the domain exploration on a decision tree, such constraints allow to prune nodes at a higher level. Practical relevance derives then from the possibility of implementing the algorithm on-line in real-time problems.

The decision problem is formulated as an integer nonlinear program (*INP*). This requires the exact knowledge of the interval between the different candidate impulse times over the considered horizon [13, 16]. To solve it, we go through an integer linear reformulation (*ILP*) [20], which can eventually be turned into an equivalent linear program (*LP*) [9]. All the three formulations are also casted within a receding horizon framework to apply to real-time control schemes. The cost structure and the local search are new contributions with respect to [2]. To be more specific, in the current paper, costs are, in general, nonlinear and state dependent whereas in [2] costs were independent of state (see eq. (9) in [2]). The result is that here we deal with a nonlinear binary program while in [2] we are faced with a linear binary

one. The nonlinear binary program is derived under the assumption that the cost has two separate terms: one depending on time and the other monotonically decreasing on the state norm. For the aforesaid reasons, finding the optimal schedule of the impulses is a real challenge for this work.

This paper is organized as follows. In Section 2, we introduce the problem. In Section 3, we provide motivations. In Section 4, we establish the main results and collect all the proofs in an appendix. In Section 5, we present a simulation example. Finally, in Section 6, we draw some conclusions and discuss future works.

**Notation.** We use the notation  $x'$  to mean the transpose of any given vector  $x$ . We write  $\|x\|$  to denote its Euclidean norm, and  $\|x(t)\|_\infty := \max_i |x_i(t)|$  its infinity norm. For two vectors  $x$  and  $y$ , we use  $x < y$  ( $x \leq y$ ) to denote  $x_i < y_i$  ( $x_i \leq y_i$ ) for all coordinate indices  $i$ . We use  $t$  to denote the continuous-time index and  $t_k$  (or simply  $k$ ) the discrete time index. The symbol  $t^+$  indicates the time instant right after an impulse at time  $t$ . Further,  $x(t^+)$  is given by the right-limit operator as  $x(t^+) = \lim_{\tau \rightarrow t^+} x(\tau)$  for a piecewise continuous function  $x(\cdot)$ . We denote by  $\mathbb{Z}_+$  and  $\mathbb{R}_+$  the set of nonnegative integers and reals respectively. The symbol  $\{0, 1\}^n$  denotes the finite set of  $n$ -dimensional binary vectors. Similarly, for given positive scalars  $m$  and  $N$ , we indicate by  $\{0, 1\}^{m \times N}$  the  $m \times N$ -dimensional binary matrix. We usually indicate with  $N$  the horizon length. We use  $\mathbf{u} := (u(0), \dots, u(N-1))'$  to indicate the vector of impulses over the horizon  $[0, N]$ . Given  $\mathbf{u}$ , and for generic integers  $\alpha, \beta \in [0, N-1]$  with  $\alpha < \beta$ , we denote by  $u_{[\alpha, \beta]} := (u(\alpha), \dots, u(\beta))'$  the vector of impulses over the interval  $[\alpha, \beta]$  extracted from  $\mathbf{u}$ . We also write  $u_{[\alpha, \beta]}(k)$  to indicate  $u(k)$ , i.e., the  $k$ th coordinate of  $u_{[\alpha, \beta]}$  for generic  $k$  between  $\alpha$  and  $\beta$ .

## 2. Impulsively-controlled systems and reverse dwell time

Consider an infinite set of strictly increasing candidate impulse times  $\{t_0, t_1, t_2, \dots\}$  defined over the infinite horizon  $[0, \infty)$  and a control law  $u : \{t_0, t_1, t_2, \dots\} \rightarrow \{0, 1\}$  which allows us to activate impulses at any discrete time  $t_k$  from 0 to  $\infty$ . Specifically, the control law  $u(\cdot)$  returns one in correspondence to an impulse and zero otherwise (see, e.g., boolean-controlled systems in [2]). With this premise, the system takes on the form in equation (1), where the function  $f : \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}^n$  is the continuous dynamics of state  $x(t)$ , the function  $h(x(t), d(t))$  is the impulse dynamics, and the variable

$d(t)$  is a disturbance:

$$\begin{aligned} \dot{x}(t) &= f(x(t), d(t)) && \text{if } t \neq t_k \text{ OR } u(t_k) = 0 \\ x(t^+) &= h(x(t), d(t)) && \text{otherwise.} \end{aligned} \quad k = 0, 1, 2, \dots \quad (1)$$

In the above dynamics the areas of the impulses at the time instants under consideration are predetermined. Further, let  $f$  be locally Lipschitz, and all signals be right-continuous.

Let  $\mathcal{N}(\tau, t_i)$  be the number of impulses in the open interval  $[t_i, t_i + \tau)$  for any  $0 \leq t_i < t_i + \tau$ .

**Definition 1 (reverse dwell time condition).** *A reverse dwell time condition in the most simple form requires that there must exist some positive  $T$  for which (2) holds for every time  $t_i$ :*

$$\mathcal{N}(T, t_i) \geq 1. \quad (2)$$

The above condition comes into play in ISS problems over an infinite horizon. Here a controller has to stabilize asymptotically (the control horizon is infinite) an “unstable” dynamics through “infinitely often” and not “too rare” controlled impulses [15].

The next step towards the problem formulation is to introduce the function to minimize. To this purpose, let  $V : \mathbb{R}^n \rightarrow \mathbb{R}$ , be a locally Lipschitz, positive definite, radially unbounded function. Let the cost of an impulse at time  $t_k$  and state  $x(t_k)$  be  $c(x(t_k), t_k) = K(t_k) + \Psi(V(x(t_k)))$ , where the function  $\Psi : \mathbb{R}_+ \mapsto \mathbb{Z}_+$  denotes a nonnegative integer cost depending on  $V(x(t_k))$  and the function  $K : \mathbb{R}_+ \mapsto \mathbb{Z}_+$  is a nonnegative integer cost depending on time  $t_k$ . Though the overall cost of any given control law  $u(\cdot)$  over the infinite horizon  $[0, \infty)$  is infinite, it is still meaningful to minimize the rate for the accumulation of the cost over a finite horizon. In this spirit, let  $\{t_0, t_1, \dots, t_{N-1}\}$  be the finite set of candidate impulse times over the finite horizon  $[t_0, t_N]$  whose length is fixed a priori. Then, the total cost over the horizon takes on the nonlinear form (we omit dependence on initial state  $x(0)$ )

$$J(u(\cdot)) = \sum_{k=0,1,\dots,N-1} c(x(t_k), t_k)u(t_k). \quad (3)$$

At this point, we need to introduce some monotonicity properties for the cost. Monotonicity plays a central role in the local search procedure discussed later on (see, e.g., the proof of Theorem 2).

**Assumption 1 (monotonicity of cost).** *Function  $\Psi(\cdot)$  is decreasing in  $V(x(t))$ .*

The above assumption simply says that, let  $V$  be a norm function, part of the cost is monotonically decreasing in the state norm. This is a key factor in reducing computations in the local search procedure. In the queueing system presented later on, we will see that monotonicity has to do with the economies of scale [6]. Similarly, in bank account flows, impulses are payments and monotonicity can be interpreted as interests deriving from late payments. We finally need to remark that when introducing the prediction step for  $\Psi(V(x(t_k)))$  in the receding horizon algorithm, we will ignore the disturbance or will replace it with its mean value.

Finally, on the basis of the above definition, we are in the position of stating the problem of interest.

**Problem 1.** *Find an impulse control law  $u(\cdot)$  that minimizes the cost (3) and satisfies  $m$  reverse dwell time conditions (2) on intervals  $[t_i, t_i + T)$ ,  $i = 1, \dots, m$ , and  $0 \leq t_i < t_i + T \leq t_N$ .*

Note that the optimization procedure requires the exact knowledge of the interval between the different candidate impulse times over the considered horizon [13, 16].

Two main points distinguish Problem 1 from other optimal impulse control problems. First, the cost (3) has a discrete-time structure which has its roots on the discrete-time nature of the controlled impulses [14]. Second, the nature of the constraints (2) allows us to turn the problem into an integer nonlinear program as shown in the following.

### 3. Motivations

#### 3.1. ISS stability to justify reverse dwell time conditions

Reverse dwell time conditions find their roots in Input-to-state stability. We recall that in Theorem 1 and Corollary 1 (b) in [15] it is proved that under a certain assumption on the continuous and impulse dynamics (1) (see Assumption (4a)-(4b) in [15]), a sufficient condition for system (1) to be ISS over an infinite horizon is that impulses satisfy a so-called *reverse dwell time condition*. In the following, we recall the required assumption.

**Assumption 2 (ISS-Lyapunov function).** *There exists a function  $V : \mathbb{R}^n \rightarrow \mathbb{R}$ , locally Lipschitz, positive definite, radially unbounded, and which*

satisfies the following condition for given  $p < 0$  and  $q > 0$  and for some function  $\chi \in \mathcal{K}_\infty$ <sup>2</sup>:

$$\nabla V(x(t))f(x(t), d(t)) \leq -pV(x(t)) + \chi(|d(t)|), \quad \forall x(t), d(t), \quad (4)$$

$$V(h(x(t), d(t))) \leq e^{-q}V(x(t)) + \chi(|d(t)|) \quad \forall x(t), d(t). \quad (5)$$

The above function  $V(\cdot)$  is called *candidate exponential ISS-Lyapunov function for (1)*. Roughly speaking, condition (4) with  $p < 0$  tells us that the continuous flow  $f(\cdot, \cdot)$  tends to destroy ISS (see the definition in [15]) by pushing the state far from the origin. To counter this effect and preserve ISS, impulses tend to drive the state towards the origin as stated in condition (5) for  $q > 0$ . Note that, despite the binary nature of variable  $u(t_k)$ , the sign of the impulse depends on the sign of the current state through the function  $h(\cdot, d(t))$ .

A possible approach to compute ISS-Lyapunov function is based on the discretization of the system in which case the problem turns into a feasibility problem for the integer nonlinear program which we will set-up in the next section.

Under the aforementioned assumption, Corollary 1 in [15], which we adapt and copy below, establishes that the system will be ISS as long as the exerted impulses are separated by a time interval  $T$  no greater than  $q/|p|$ .

**Lemma 1.** *System (1) is ISS if  $\mathcal{N}(T, t_i) \geq 1$ , where  $T \leq q/|p|$ .*

Note that the preliminary result reported above is a special case of Corollary 1 in [15].

### 3.2. Economies of scales to justify cost monotonicity

Consider a queue describing a constant demand accumulating in a production/distribution inventory system or at a service provider. The system manager chooses in which instants to process the accumulated demand thus to empty the queue. Then, the queue length is the system state. Each time the queue is served an impulse acts on the system reducing the state to zero. The queue is described by the following impulsive dynamics:

$$\begin{aligned} \dot{x}(t) &= d(t) & \text{if } t \neq t_k \text{ OR } u(t_k) = 0 & & k = 0, 1, 2, \dots \\ x(t^+) &= 0 & \text{otherwise.} & & \end{aligned} \quad (6)$$

---

<sup>2</sup>A function  $\theta : [0, \infty) \rightarrow [0, \infty)$  is of class  $\mathcal{K}$ , and we write  $\theta \in \mathcal{K}$  when  $\theta$  is continuous, strictly increasing, and  $\theta(0) = 0$ . If  $\theta$  is also unbounded, then we say that it is of class  $\mathcal{K}_\infty$  and write  $\theta \in \mathcal{K}_\infty$ .



In this context, the term  $\Psi(V(x(t_k)))$  is representative of the economies of scales. Actually, suppose that the system manager pays a processing cost  $cost(x(t_k))$  increasing and concave on  $x(t_k)$  anytime he/she processes the demand. Then, if  $x(t_k) > 0$  is the demand processed at time  $t_k$ , we can interpret cost  $\Psi(\cdot)$  as *the cost per unit* of demand processed, i.e.,  $\Psi(V(x(t_k))) := \frac{cost(x(t_k))}{x(t_k)}$  which is decreasing due to concavity of  $cost(\cdot)$ . As regards the cost term  $K(t_k)$ , it can be used to model inertia, namely the tendency of the system manager to postpone processes, and this is captured by taking  $K(t_k)$  decreasing. Conversely, increasing  $K(t_k)$  would mean a prompt demand processing on the part of the system manager. Constant  $K(t_k)$  can be used whenever the manager's aim is to simply minimize the number of actions (impulses) on the queue system over a finite horizon.

#### 4. Main results

We highlight three main results pertaining to the case of impulsive systems with uniformly distributed impulse instants. First, we analyze constraints to reduce the computational burden associated with the exploration of the solution domain. Second, we tighten the above bounds via a receding horizon algorithm embedding the above constraints. Third we show that the receding horizon algorithm is optimal for the case  $T = N$ . As a preliminary result, we provide cost bounds by replacing the original problem with a more tractable linear program.

##### 4.1. Preliminary result: cost bounds

In this section, we provide an upper bound for the cost. To do this, we first show how the optimal impulse control problem described in Problem 1 can be turned into an integer nonlinear programming one (*INP*). We then introduce an associated integer linear program (*ILP*) and use the optimal solution of this latter problem to compute an upper bound for the cost.

The (*INP*) is based on a discretized model for which we need to use an exact integration over the interval between two candidate impulse times. In order to simplify the notation of the (*INP*) to be defined in this section, let us assume that the impulses for the problem described in the previous section may occur only at integer times, and therefore we have  $t_k = k$ , and we can simply write  $x(k)$ ,  $u(k)$  instead of  $x(t_k)$  and  $u(t_k)$ . Let us define  $\mathbf{f}(x(k), d(k)) := \int_{\tau=k}^{k+1} f(x(\tau), d(\tau))d\tau$ . In the case where the latter integration is subject to uncertainty, see e.g. [21], one can deal with a robust

approximation which would appear in the form of a min-max program [4]. Observe that the reverse dwell time conditions (2) can be described by linear constraints of the form:

$$\underbrace{\begin{bmatrix} \overbrace{1 \dots 1}^T & 0 \dots 0 & \dots & \overbrace{0 \dots 0}^T \\ 0 & 1 \dots \dots \dots 1 & \dots & 0 \dots 0 \\ \vdots & & \ddots & \vdots \\ 0 \dots 0 & 0 \dots 0 & \dots & 1 \dots 1 \end{bmatrix}}_A \begin{bmatrix} u(0) \\ \vdots \\ u(N-1) \end{bmatrix} \geq \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}. \quad (7)$$

In the above constraints, we have denoted by  $N$  the number of candidate impulses, which follows from assuming  $t_k = k$ . From the latter assumption, though  $T$  in the original problem is not an integer but a real, we can simply take  $T$  as integer. The above linear constraints are useful to introduce the set of feasible solutions for our nonlinear integer program. To this aim, we denote by  $\mathbf{u} = [u(0), \dots, u(N-1)]'$  the column vector of candidate impulses. Then all feasible solutions are in the discrete set:

$$\mathcal{F} = \{\mathbf{u} \in \{0, 1\}^N : \text{linear constraints (7)}\}.$$

To complete the formulation of the nonlinear integer program, let us define the vector of exact costs  $\mathbf{c} = [c(x(0), 0), \dots, c(x(N-1), N)]$ , and let us recall that  $\mathbf{f}(x(k), d(k)) := \int_{\tau=k}^{k+1} f(x(\tau), d(\tau)) d\tau$ .

For the case of systems with impulse times of the form  $t_k = k$ , Problem 1 is transformed in the following mathematical problem:

$$\begin{aligned} (INP) \quad & \min_{\mathbf{u} \in \mathcal{F}} \mathbf{c}\mathbf{u} & (8) \\ & x(k+1) = x(k) + \mathbf{f}(x(k), d(k)) \\ & + [h(x(k), d(k)) - x(k)]u(k), \quad k = 0, \dots, N-1. \end{aligned}$$

Here, the state evolution  $x(k)$ ,  $k = 0, 1, \dots, N$  is captured by the above sampled dynamics obtained integrating (1). The term  $[h(x(k), d(k)) - x(k)]u(k)$  is such that when we compute the integration over the interval between two candidate impulse times, say  $k$  and  $k+1$ , the initial value of the state is either  $h(x(k), d(k))$  or  $x(k)$  depending on whether there is an impulse at time  $k$ , namely  $u(k) = 1$ , or not.

Looking at the above problem, we would like to reiterate that a binary control variable simply captures the possibility of activating a discrete event

to stabilize an unstable dynamics. It is an impulse in our example, but it could also be a mode switch as in [2].

Note that a mathematical problem (*INP*) of type (8) can be obtained in the more general case where the candidate impulse times are not uniformly distributed under certain conditions. First, we need to know exactly the interval between the different candidate impulse times over the considered horizon. Second, we need to be able to perform an exact integration over the interval between two candidate impulse times.

To put more emphasis on certain geometric aspects of the feasible set  $\mathcal{F}$ , we next get rid of any nonlinearities in the costs and focus on a linear rearrangement of the (*INP*) centered around the same set  $\mathcal{F}$ . Specifically, let  $\tilde{\mathbf{c}} = [K(0), \dots, K(N-1)]$  be the vector of the costs that depend only on time, and refer to the following integer linear program:

$$(ILP) \quad \min_{\mathbf{u} \in \mathcal{F}} \tilde{\mathbf{c}}\mathbf{u}. \quad (9)$$

The above two problems (8) and (9) share the same feasible set  $\mathcal{F}$ . We are now in the position of introducing the receding horizon approach used to solve the above problems.

To convexify the set of feasible solution, we start by the observation that  $\mathcal{F}$  is a discrete set in the sense that it contains only integer points. However we can replace the integrality constraints  $\mathbf{u} \in \{0, 1\}^N$  by the relaxed and more tractable constraints  $0 \leq \mathbf{u} \leq 1$  and consider the resulting polytope

$$\mathcal{P} = \{\mathbf{u} \in \mathbb{R}^N : \text{linear constraints (7), } 0 \leq \mathbf{u} \leq 1\}.$$

Relaxing the integer constraints has the advantage of turning (*ILP*) into a more tractable linear program:

$$(LP) \quad \min_{\mathbf{u} \in \mathcal{P}} \tilde{\mathbf{c}}\mathbf{u}. \quad (10)$$

Denote  $\mathbf{u}^{LP}$  an optimal solution of (*LP*) and write  $J_{LP}$  to mean the optimal cost returned by (*INP*) in correspondence to the solution  $\mathbf{u}^{LP}$ . Similarly, let  $J_{ILP}$  be the optimal cost returned by (*INP*) in correspondence to an optimal solution  $\mathbf{u}^{ILP}$  of (*ILP*). We are in the position to state the main result of this paper. The novelty is that this result is based on geometric considerations borrowed from combinatorial optimization (see [20], part III.1 on integral polyhedra, p. 540).

**Theorem 1.** *Solution  $\mathbf{u}^{LP}$  is also optimal for (ILP) and therefore there exists a  $\mathbf{u}^{ILP} = \mathbf{u}^{LP}$  for which it holds:*

$$J_{LP} = J_{ILP} \geq J_{INP}. \quad (11)$$

*Proof.* Given in the appendix.  $\square$

The above result hinges upon the fact that  $\mathcal{P}$  is an integral polyhedron. As a consequence we have that (LP), which is the linear relaxation of the (ILP), has an integral optimal solution. We recall from the combinatorial optimization that, roughly speaking, a *linear relaxation* is a problem where all integer constraints have been dropped. Now, if  $\mathcal{P}$  is an integral polyhedron, all its vertices are integer. This implies that, when we solve the LP problem we obtain an optimal solution which corresponds to a vertex of  $\mathcal{P}$  and therefore it is integer, even if the LP problem does not impose a-priori for the solution to be integer (see, e.g., total unimodularity in [20]). The advantages of adopting an integral polyhedron is common to another work of the author [2], but for a different problem.

The significance of the above result is three-fold: i) we can formulate a linear program whose solution  $\mathbf{u}^{LP}$  satisfies the reverse dwell time conditions and as such implicitly stabilizes the dynamics in the sense of ISS, ii) we can solve a linear program to bound from above the optimal cost  $J_{INP}$ , iii) we show that tightness of the bound depends solely on the nonlinear dependence of the cost  $c(x(k), k)$  on  $x(k)$  (but not on the integral nature of the decision variables).

#### 4.2. Tightening the bounds via receding horizon and local search

This section introduces a method to improve the bounds provided in the preceding section. The method is based on the receding horizon algorithm illustrated in Table 1. In the following, we first introduce the algorithm and then discuss its properties.

##### 4.2.1. The algorithm

The algorithm receives  $\mathbf{u}^{LP}$  as input, and gives as output the impulse sequence  $\mathbf{u}^{rh}$  whose cost is  $J_{rh} := \mathbf{c}\mathbf{u}^{rh}$ . We recall that  $\mathbf{u}^{LP}$  is an optimal solution of (LP) in (10). The method consists in solving problem (INP) online over a receding horizon  $[\alpha(j), \beta(j)]$  with  $0 \leq \alpha(j) < \beta(j) < N$ . Let us denote by  $(INP_{[\alpha(j), \beta(j)]})$  the problem solved at the generic iteration  $j$ . In addition, let us denote by  $u_{[\alpha(j), \beta(j)]}^*$  the optimal solution of  $(INP_{[\alpha(j), \beta(j)]})$ .

---

### Receding horizon algorithm

---

**Input:** system parameters (see eq. (1)-(3)) and  $\mathbf{u}^{LP}$ .

**Output:** impulse sequence  $\mathbf{u}^{rh}$

- 1 : **Initialize** Assign  $j := 0$ ;  $\alpha(j) = 0$  and  $\beta(j) = T - 1$ ,
  - 2 : **If**  $\beta(j) < N - 1$
  - 3 :   (Prediction horizon) solve ( $INP_{[\alpha(j),\beta(j)]}$ ) and find  $u_{[\alpha(j),\beta(j)]}^*$  as in (12)
  - 4 :   (Control horizon) implement  $u(\alpha(j)) \dots u(\beta(j))$
  - 5 :   set  $j = j + 1$ ,  $\alpha(j) = \alpha(j) + 1$ , and  $\beta(j) = \beta(j) + 1$
  - 6 : **else** implement  $u(\alpha(j)) = \dots = u(N - 1) = 0$
  - 7 : **end if**
- 

Table 1: Receding horizon algorithm.

We will show that  $u_{[\alpha(j),\beta(j)]}^*$  has only one non-null component at time  $\gamma \in [\alpha(j), \beta(j)]$ , namely  $u_{[\alpha(j),\beta(j)]}^* = (u^*(\alpha(j)) \dots u^*(\beta(j)))'$  where

$$\begin{aligned} u^*(\alpha(j)) = \dots = u^*(\gamma(j) - 1) = 0, \quad u^*(\gamma) = 1, \\ u^*(\gamma + 1) = \dots = u^*(\beta(j)) = 0. \end{aligned} \tag{12}$$

The problem solved at iteration  $j$  can be defined inductively. To do this, let us introduce the column vector of decisions over the horizon  $[\alpha(j), \beta(j)]$

$$u_{[\alpha(j),\beta(j)]} = (u(\alpha(j)), \dots, u(\beta(j)))',$$

and the feasible set

$$\mathcal{F}_{[\alpha(j),\beta(j)]} = \{u_{[\alpha(j),\beta(j)]} \in \{0, 1\}^T : \sum_{k=\alpha(j)}^{\beta(j)} u(k) \geq 1\}.$$

Now, let the receding horizon solution returned at iteration  $j - 1$  be

$$\mathbf{u}^{j-1} := (u_{[0,\alpha(j)-1]}^{*'}, u_{[\alpha(j),N-1]}^{LP'})'.$$

We recall that  $u_{[\alpha(j),N-1]}^{LP}$  involves the components of  $\mathbf{u}^{LP}$  in the interval  $[\alpha(j), N - 1]$ . Thus, the solution  $\mathbf{u}^{j-1}$  differs from  $\mathbf{u}^{LP}$  only in the first

$\alpha(j)$  components. For  $j = 0$ , we initialize  $\alpha(j) = 0$ ,  $u_{[0,\alpha(j)-1]}^* = nil$  and  $\mathbf{u}^{j-1} := u_{[0,N-1]}^{LP}$ . Given the inductive reasoning, the solution  $u_{[0,\alpha(j)-1]}^*$  at  $j - 1$  is unambiguously determined once we clarify how to obtain  $\mathbf{u}^j = (u_{[0,\alpha(j+1)-1]}^*, u_{[\alpha(j+1),N-1]}^{LP'})'$ , and in particular  $u_{[0,\alpha(j+1)-1]}^*$  at iteration  $j$ .

The problem solved at the generic iteration  $j$  is given by:

$$\begin{aligned}
& (INP_{[\alpha(j),\beta(j)]}) \quad \min_{u_{[\alpha(j),\beta(j)]} \in \mathcal{F}_{[\alpha(j),\beta(j)]}} \quad \mathbf{c}\mathbf{u} & (13) \\
& u_{[\beta(j)+1,N-1]} = u_{[\beta(j)+1,N-1]}^{LP} \\
& u_{[0,\alpha(j)-1]} = u_{[0,\alpha(j)-1]}^* \\
& x(k+1) = x(k) + \mathbf{f}(x(k), d(k)) \\
& \quad + [h(x(k), d(k)) - x(k)]u(k), \quad k = 0, \dots, N-1.
\end{aligned}$$

Essentially, the vectors  $u_{[0,\alpha(j)-1]}$  and  $u_{[\beta(j)+1,N-1]}$  are fixed and the optimization involves only the components in the interval  $[\alpha(j), \beta(j)]$ , namely  $u_{[\alpha(j),\beta(j)]}$ .

The interval extremes  $\alpha(j)$  and  $\beta(j)$  are set as explained next. We set  $\alpha(j)$  equal to the first instant after an impulse has been activated, i.e.,  $u_{[0,\alpha(j)-1]}^*(\alpha(j) - 1) = 1$ . We set  $\beta(j)$  so that

**C1** the interval  $[\alpha(j), \beta(j)]$  contains at most  $T$  candidate impulses, and

**C2** the vector  $u_{[\alpha(j),\beta(j)]}^{LP}$  has only one non-null component, i.e.,

$$\begin{aligned}
u^{LP}(\alpha(j)) = \dots = u^{LP}(\delta(j) - 1) = 0, \quad u^{LP}(\delta) = 1, \\
u^{LP}(\delta + 1) = \dots = u^{LP}(\beta(j)) = 0.
\end{aligned}$$

In the above, we denote by  $\delta$  the index (time instant) of the non-null component of  $u_{[\alpha(j),\beta(j)]}^{LP}$ .

In other words, we set  $\beta(j) := \min\{\alpha(j) + T - 1, r\}$ , where  $r$  is the instant of a second impulse that should appear in  $u_{[\alpha(j),\alpha(j)+T-1]}^{LP}$ , formally,  $r := \{\min_{\hat{r} \in [\alpha(j), N-1]} \sum_{k=\alpha(j)}^{\hat{r}} u_{[\alpha(j),N-1]}^{LP}(k) = 2\}$ .

To compute the solution  $\mathbf{u}^j = (u_{[0,\alpha(j+1)-1]}^*, u_{[\alpha(j+1),N-1]}^{LP'})'$ , we need to compute the vector  $u_{[0,\alpha(j+1)-1]}^*$ . This is obtained by juxtaposition as clarified next. Let us consider two consecutive intervals  $[0, \alpha(j) - 1]$  and  $[\alpha(j), \alpha(j + 1) - 1]$ . Let the optimal solutions in the two intervals be

$$\begin{aligned}
u_{[0,\alpha(j)-1]}^* &= (u^*(0), \dots, u^*(\alpha(j) - 1))', \\
u_{[\alpha(j),\alpha(j+1)-1]}^* &= (u^*(\alpha(j)), \dots, u^*(\alpha(j + 1) - 1))',
\end{aligned}$$

and juxtapose both solutions in order to define a new vector

$$u_{[0,\alpha(j+1)-1]}^* = (u_{[0,\alpha(j)-1]}^{*'}, u_{[\alpha(j),\alpha(j+1)-1]}^{*'})'. \quad (14)$$

The solution returned by the algorithm at iteration  $j$  is then

$$\mathbf{u}^j = (u_{[0,\alpha(j+1)-1]}^{*'}, u_{[\alpha(j+1),N-1]}^{LP'})'. \quad (15)$$

Given  $u_{[\alpha(j),\beta(j)]}^*$ , we set  $\alpha(j+1) = \gamma + 1$  and reiterate the procedure.

The method can be visualized using the decision tree displayed in Fig. 1.

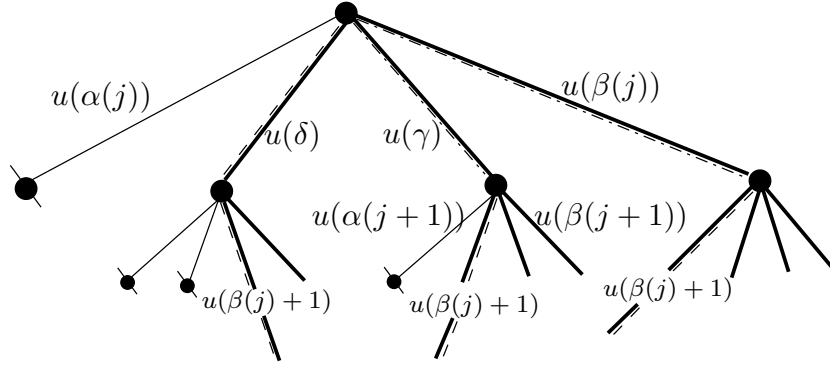


Figure 1: Decision tree illustrating the exploration of the solution domain.

There, edges correspond to variables and nodes correspond to time instants. The edges departing from any node identify the variables at future times. In particular, the edges at level 1 represent the decision variables  $u(\alpha(j)), \dots, u(\beta(j))$  involved in  $(INP_{[\alpha(j),\beta(j)]})$ . The dashed edge at level 1 corresponds to the non-null component  $u^{LP}(\delta) = 1$  introduced in condition **C2**. The dashed edges at level 2 are all associated with the component of vector  $u_{[\beta(j)+1,N-1]}$  which is fixed to one. We recall that in the definition of  $(INP_{[\alpha(j),\beta(j)]})$  the vector  $u_{[\beta(j)+1,N-1]}$  is fixed. The algorithm builds upon the idea that  $u_{[\alpha(j),\beta(j)]}^*$  has only one non-null component at time  $\gamma$ , as expressed by (12). In addition to this, we have  $\gamma \geq \delta$ , where  $\delta$  is the non-null component of  $\mathbf{u}^{LP}$  in  $[\alpha(j), \beta(j)]$ , as discussed in condition **C2**. This is illustrated in Fig. 1, where the edge for  $u(\gamma)$  is on the right of the edge for  $u(\delta)$ . This fact is formally stated in the next section.

#### 4.2.2. Properties of the algorithm

Let us start by noting that problem  $(INP_{[\alpha(j),\beta(j)]})$  guarantees the reverse dwell time condition on the interval  $[\alpha(j),\beta(j)]$  from the definition itself of  $\mathcal{F}_{[\alpha(j),\beta(j)]}$ .

With regards to the setting of  $\beta(j)$ , condition **C1** is fundamental to guarantee feasibility of the solution returned by the algorithm as shown in Lemma 2, and furthermore it allows us to specialize the reverse dwell time conditions to  $[\alpha(j),\beta(j)]$  and so we write them as  $u_{[\alpha(j),\beta(j)]} \in \mathcal{F}_{[\alpha(j),\beta(j)]}$ . Condition **C2** is critical to arrive at a solution not worse than  $u^{LP}$  as established in Lemma 3.

The next theorem establishes that  $u_{[\alpha(j),\beta(j)]}^*$  has only one non-null component at time  $\gamma \geq \delta$ , where  $\delta$  identifies the non-null component  $u^{LP}(\delta) = 1$  as in condition **C2**.

**Theorem 2.** *Solution  $u_{[\alpha(j),\beta(j)]}^*$  has the structure given in (12), namely, for a generic integer  $\gamma \in [\alpha(j),\beta(j)]$  with  $\gamma \geq \delta$ :*

$$\begin{aligned} u^*(\alpha(j)) &= \dots = u^*(\gamma(j) - 1) = 0, & u^*(\gamma) &= 1, \\ u^*(\gamma + 1) &= \dots = u^*(\beta(j)) = 0. \end{aligned}$$

*Proof.* Given in the appendix. □

The significance of the above result is mainly in that we can ignore earlier impulses (see the thin edges and pruned nodes in the decision tree of Fig. 1) thus reducing the computational complexity of the receding horizon algorithm. In this sense, an exhaustive search would imply to explore all thick edges. The provided algorithm reduces furtherly the computational burden as it fixes future decisions and this results in comparing  $u^{LP}$  with only two other solutions, emphasized in dash-dot in Fig. 1.

The next result establishes that the solution returned by the algorithm at iteration  $j$ , which is given in (15), satisfies the constraints of  $(INP)$ .

**Lemma 2.** *The solution  $\mathbf{u}^j$  is feasible for the  $(INP)$ .*

*Proof.* Given in the appendix. □

Furthermore, it can be shown that the algorithm at each iteration returns a solution which is not worse than the one returned at the previous iteration. In other words, we have that  $\mathbf{u}^j$  improves  $\mathbf{u}^{j-1}$ . To see this, let us write  $\Delta J(j) := \mathbf{c}(\mathbf{u}^j - \mathbf{u}^{j-1})$  to denote the cost improvement between two consecutive iterations.



**Lemma 3.** For all  $j$ , it holds  $\Delta J(j) \leq 0$ .

*Proof.* Given in the appendix. □

The receding horizon algorithm returns a feasible solution in linear time as remarked next.

**Theorem 3.** The receding horizon solution  $\mathbf{u}^{rh}$  is feasible for (INP) and provides a tighter upper bound:

$$J_{LP} \geq J_{rh} \geq J_{INP}.$$

Also, the receding horizon algorithm finds  $\mathbf{u}^{rh}$  in the worst-case in  $O(N)$ . Furthermore, if  $T = N$  then the solution  $\mathbf{u}^{rh}$  is also optimal and the bound is tight, i.e.,  $J_{rh} = J_{INP}$ .

*Proof.* Given in the appendix. □

The above result is a novelty as it uses geometric considerations borrowed from combinatorial optimization in a receding horizon context (see [20], part III.1 on integral polyhedra, p. 540).

## 5. Numerical Illustrations

This section provides a detailed numerical analysis of a simple network controlled system extracted from the queueing literature. The example adds another level of complexity in the optimization problem, namely, that of trying to stabilize simultaneously ten different systems instead of one. This reflects in the use of an additional scheduling component that concerns the selection of the system component (in this case a queue) that will receive the next impulse. Despite this new element, we can still reframe the example within the framework of ISS and reverse dwell time impulse control.

Assume we have queues describing demands accumulating at different geographic sites in a production/distribution inventory system. The system manager can serve only one queue (or site) at the time and up to a limited number of demands established by certain capacity constraints. Then, the queue lengths are the system state. Each time a queue is served an impulse acts on the corresponding system reducing the state of a fixed quantity. Each queue is described by the following impulsive dynamics. When no impulse occurs, for all  $i = 1, \dots, n$ ,

$$\dot{x}_i(t) = a_i(t)x_i(t) + \rho \text{rand}(-1, 1) \quad t \neq t_k \quad k = 0, 1, 2, \dots$$

$\lambda$	$\rho$	$x_0$	$a$	$\epsilon$	$N$	$\tilde{c}$
50	150	rand([5,...,10])	rand([0.8,...,1])	$5(\frac{n3^2}{2^2})$	1700	rand([1,...,N])

Table 2: Parameters of the experiment.

with rate  $a_i(t) > 0$  and where  $rand(-1, 1)$  is a random disturbance uniformly distributed in the interval between  $-1$  and  $1$ ,  $\rho$  is the maximal disturbance. Furthermore, we assume that the impulse acts on the longest (in magnitude) queue. Then, denoting by  $i_k = \arg \max_i |x_i(t_k)|$  such a queue, the impulsive dynamics at impulse times is

$$x_i(t^+) = \begin{cases} x_i(t) - \text{sign}(x_i(t)) & \text{if } |x_i(t)| > 1 \\ 0 & \text{if } |x_i(t)| \leq 1 \end{cases} \quad \text{if } i = i_k \quad t = t_k, k = 0, 1, 2, \dots$$

$$x_i(t^+) = x_i(t) \quad \text{if } i \neq i_k$$

Note that the queues tend to diverge if no impulses are activated. So, roughly speaking, impulses have a stabilizing effect while the dynamics is unstable.

Serving a queue has a cost which depends on time, and also decreases with the queue length. In general, such a decreasing term models economies of scale [6]. This is a term used to describe situations where the cost of producing an additional unit of an outcome, (i.e, a good or service) decreases with the total volume of the outcome.

In mathematical terms, this means that activating an impulse has a cost  $c(x(t), t) = K(t) + \Psi(V(x(t)))$  where the state dependent term is a linear decreasing function on the infinity norm of  $x(t)$ , i.e.,  $V(x(t)) = \|x(t)\|_\infty$ . In formulas, this corresponds to

$$\Psi(V(x(t))) = \lambda \cdot \max_t K(t) \cdot \frac{UB(\|x\|_\infty) - \|x(t)\|_\infty}{UB(\|x\|_\infty)},$$

where  $\lambda \geq 0$  and it is specified a-priori. We have denoted by  $UB(\|x\|_\infty)$  an *upper bound of the infinity norm of the state* and we take for it the value 10. Parameter  $\lambda$  weights the influence of the state dependent term  $\Psi(V(x(t)))$  with respect to the time-varying term  $K(t)$ . Actually, for  $\|x(t)\|_\infty = 0$ , the term  $\Psi(V(x(t)))$  is  $\lambda$  times greater than the maximal time varying cost,  $\max_t K(t)$ .

This study is carried out by setting the parameters as listed in Table 2. In particular, we take  $\lambda = 50$  and set the maximal disturbance  $\rho = 150$ .

Queues start from random values between 5 and 10, and the rate coefficients are randomly distributed from 0.8 to 1, i.e.,  $a_i = \text{rand}([0.8, \dots, 1])$  for all  $i = 1, \dots, n$ . The horizon length is  $N = 1700$ , and the values of  $K(t)$ , or, which is the same, of the approximate costs  $\tilde{c}$  are extracted randomly from the interval 1 to  $N$ .

Also, we take as time unit the value  $-\log((\epsilon - 1)/\epsilon)$  where  $\epsilon = 10(\frac{n3^2}{2^2})$ . This value is an estimate of the rate  $q$  used in (5) and it derives from the fact that between two consecutive impulses we can guarantee the condition  $\|x(t^+)\|_2 \leq \frac{\epsilon-1}{\epsilon}\|x(t)\|_2$  at least on the average because of the random disturbance  $\text{rand}(-1, 1)$  (the same condition is always guaranteed in absence of a random disturbance). Beyond this, as the rates  $a_i$  are upper bounded by 1, we can estimate the rate  $p$  used in (4) equal to 1, so that the value  $-\log((\epsilon - 1)/\epsilon)$  can also be taken as dwell time  $T$  for our simulations. The sample interval is  $\frac{1}{10}$  of the dwell time  $T = -\log((\epsilon - 1)/\epsilon)$ , i.e.,  $\Delta t = -\log((\epsilon - 1)/\epsilon)0.1$ .

We simulate the receding horizon procedure starting at  $x(0)$ . In the experiment, we compare the solutions of (9) and (8). The first solution is obtained by solving the linear problem

$$\min_{\mathbf{u} \in \mathcal{P}} \tilde{\mathbf{c}}\mathbf{u}, \quad \mathcal{P} = \{\mathbf{u} \in \mathbb{R}^N : \text{linear constraints (7), } 0 \leq \mathbf{u} \leq 1\},$$

and returns a suboptimal cost. The second solution is the optimal solution obtained from the receding horizon algorithm. In particular, we illustrate the ISS properties of both solutions for a system with  $n = 10$  independent queues.

In Fig. 2, left, starting from  $x_0 = [8 \ 10 \ 8 \ 10 \ 10 \ 7 \ 5 \ 6 \ 10 \ 10]'$ , the state is driven to the interval -1 and 1 at about  $t = 1200$  and kept within such interval for the rest of the time. The two state trajectories are obtained by implementing the suboptimal control (top) and the optimal control (bottom). In Fig. 2, right, a zoom in the first 100 samples evidences the impulses and shows the difference between the two controls. The optimal control postpones the first impulse at about  $t = 10$  (see, bottom) while with the suboptimal control the first impulse is at  $t = 0$  and thus, because of the effect of this first impulse, one of the five systems with initial state 10 actually evolves from 9.

## 6. Conclusions and future directions

In summary, this article provides a detailed analysis of a class of impulse optimal control problems characterized by uniformly distributed impulse times, reverse dwell time conditions and a special structure of the cost.

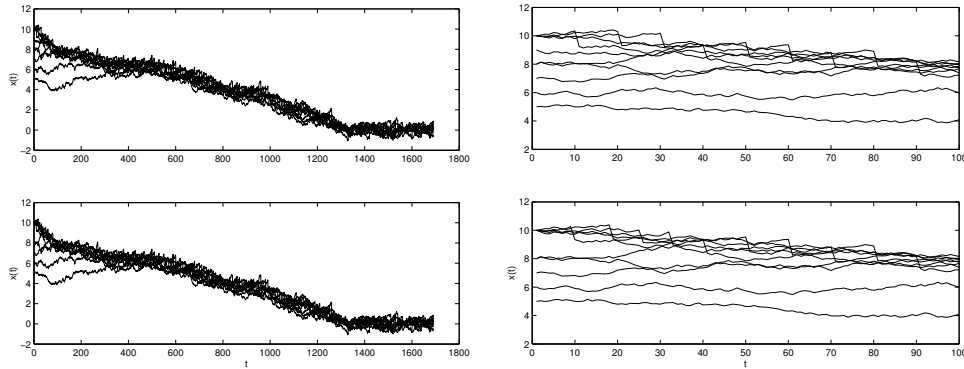


Figure 2: Second experiment: time plot of  $x(t)$  with approximate and exact controls (left). Zoom on the first 10 samples (right).

From a theoretical perspective the obtained results have proved certain geometrical properties of the discrete set of feasible solutions. From a practical standpoint, such properties have reduced the computational burden thus making the algorithm suitable for the on-line implementation.

Future research will elaborate more on the connections between predictive control and combinatorial optimization techniques in order to circumvent the challenges deriving from the mixed integer nature of the controls. We do this motivated by the observation that mixed integer predictive control is attracting the attention from an increasing number of researchers in the optimization and control community. More specifically, we first think of extending the use of cutting planes algorithms to all those problems that do not benefit from total unimodularity. In all these cases, we can no longer solve the linear relaxation and obtain binary solutions. So, cutting planes are introduced iteratively with the aim of eliminating fractional solutions. Second, we would like to investigate analogies between regeneration intervals used in combinatorial optimization and reverse dwell time conditions developed in hybrid/impulse control. Our current belief is that both notions have some common physical meaning behind them which we would like to inspect more in detail in the future.

## Appendix

### *Proof of Theorem 1*

We start by observing that the constraint matrix  $A \in \{0, 1\}^{m \times N}$  used in the inequalities (7) turns out to be an *interval matrix*, i.e., it has 0-1 entries and each row is of the form

$$(0, \dots, 0 \quad \underbrace{1, \dots, \dots, 1}_{\text{consecutive 1's}} \quad 0, \dots, 0).$$

Now, it is well known from the literature [20] that each interval matrix is totally unimodular where we remind here that a matrix is *totally unimodular* if the determinant of any square sub-matrix is equal to  $-1$ ,  $0$  or  $1$ . This means that the polyhedron  $\mathcal{P}$  is an integral polyhedron. As a consequence we have that the linear relaxation of the (*ILP*) has an integral optimal solution as established in the next lemma.

**Lemma 4 ([2]).** *Solving (*ILP*) is equivalent to solving the linear programming problem:*

$$(LP) \quad \min_{\mathbf{u} \in \mathcal{P}} \quad \tilde{\mathbf{c}}\mathbf{u}.$$

Invoking the above lemma, we then have that  $\mathbf{u}^{LP}$  is also optimal for (*ILP*). This means that (*ILP*) has at least one optimal solution  $\mathbf{u}^{ILP} = \mathbf{u}^{LP}$ . Then, we can substitute the latter solution in (*INP*) thus obtaining  $J_{LP} = J_{ILP}$  and the first equality of (11) is proved.

To prove  $J_{ILP} \geq J_{INP}$  observe that  $\mathbf{u}^{LP}$  belongs to  $\mathcal{F}$  and as such it is feasible for (*INP*). In other words, we have  $\mathbf{u}^{LP} \in \mathcal{F}$  which implies  $J_{INP} = \min_{\mathbf{u} \in \mathcal{F}} \mathbf{c}\mathbf{u} \leq \mathbf{c}\mathbf{u}^{LP}$  and this concludes our proof.

### *Proof of Theorem 2*

The underlying idea is to make use of  $u_{[\alpha(j), \beta(j)]}^{LP}$  to find  $u_{[\alpha(j), \beta(j)]}^*$ . Let us introduce the subset of solutions with multiple impulses

$$\mathcal{I} = \{u_{[\alpha(j), \beta(j)]} : \sum_{k=\alpha(j)}^{\beta(j)} u(k) \geq 2\}.$$

Also, let us recall that  $u^{LP}(\delta)$  is the non-null component of  $u_{[\alpha(j),\beta(j)]}^{LP}$  and construct the subset  $\mathcal{D} \subset \{0, 1\}^T$  of solutions presenting one or more impulses at any time earlier than  $\delta$ , namely:

$$\mathcal{D} = \{u_{[\alpha(j),\beta(j)]} : \sum_{k=\alpha(j)}^{\delta-1} u(k) \geq 1, u(\delta) = \dots = u(\beta(j)) = 0\}.$$

Now, to prove (12) we need to show that  $u_{[\alpha(j),\beta(j)]}^*$  is not in  $\mathcal{I}$  nor in  $\mathcal{D}$  as established next. Formally, solution  $u_{[\alpha(j),\beta(j)]}^* \notin \mathcal{I} \cup \mathcal{D}$ .

To see that  $u_{[\alpha(j),\beta(j)]}^*$  has a single non-null component,  $u_{[\alpha(j),\beta(j)]}^* \notin \mathcal{I}$  observe that any other solution with additional non-null components would provide a higher cost.

To prove  $u_{[\alpha(j),\beta(j)]}^* \notin \mathcal{D}$ , it suffices to show that  $u_{[\alpha(j),\beta(j)]}^*$  dominates any other solution in  $\mathcal{D}$ . To see this, we need to refer to Assumptions 2 and 1. Actually, it holds  $c(x(k), k) = K(k) + \Psi(V(x(k))) > \tilde{c}(k) = K(k)$  for all  $k$ . Now, if the index  $\delta$  gives the minimum to the cost with  $\tilde{c}(k)$ , then it gives also the minimum to the cost with  $c(x(k), k)$ , because the sequence  $\Psi(V(x(k)))$  is decreasing on  $V(x(k))$ .

*Proof of Lemma 2*

We need to prove that  $\mathbf{u}^j \in \mathcal{F}$ . The latter is true if

- i)  $u_{[0,\alpha(j+1)-1]}^{*}$  satisfies conditions  $\sum_{k=\alpha(\hat{j})}^{\beta(\hat{j})} u(k) \geq 1$  for all integer  $\hat{j}$  such that  $0 \leq \hat{j} \leq \alpha(j) - 1$ , and
- ii)  $u_{[\alpha(j+1),N-1]}^{LP}$  satisfies  $\sum_{k=\alpha(\hat{j})}^{\beta(\hat{j})} u(k) \geq 1$  for all integer  $\hat{j}$  such that  $\alpha(j+1) \leq \hat{j} \leq N - 1$ .

Now, condition ii) is satisfied as  $u^{LP}$  is feasible for  $(INP)$ . To prove condition i) we first observe that solution  $u_{[0,\alpha(j)-1]}^*$  satisfies  $\sum_{k=\alpha(\hat{j})}^{\beta(\hat{j})} u(k) \geq 1$  for all integer  $\hat{j}$  such that  $0 \leq \hat{j} \leq \alpha(j) - 1$ . We also note that  $u_{[0,\alpha(j+1)-1]}^*$  satisfies  $\sum_{k=\alpha(\hat{j})}^{\beta(\hat{j})} u(k) \geq 1$  for all integer  $\hat{j}$  such that  $0 \leq \hat{j} \leq \alpha(j+1) - 1$  and this concludes our proof.

*Proof of Lemma 3*

Note that  $u_{[\alpha(j),\beta(j)]}^{LP}$  is feasible for  $(INP_{[\alpha(j),\beta(j)]})$  as  $u_{[\alpha(j),\beta(j)]}^{LP} \in \mathcal{F}_{[\alpha(j),\beta(j)]}$ . Invoking the optimality of  $u_{[\alpha(j),\beta(j)]}^*(k)$  the we can infer the following inequality which proves our thesis:

$$\begin{aligned} \Delta J(j) &= \left( \sum_{k=\alpha(j)}^{\beta(j)} c(x(k), k) u_{[\alpha(j),\beta(j)]}^*(k) + \sum_{k=\beta(j)+1}^{N-1} c(x(k), k) u_{[\alpha(j),\beta(j)]}^{LP}(k) \right) \\ &\quad - \left( \sum_{k=\alpha(j)}^{\beta(j)} c(x(k), k) u_{[\alpha(j),\beta(j)]}^{LP}(k) + \sum_{k=\beta(j)+1}^{N-1} c(x(k), k) u_{[\alpha(j),\beta(j)]}^{LP}(k) \right) \leq 0. \end{aligned}$$

*Proof of Theorem 3*

Let us show that  $J_{LP} \geq J_{rh}$ . Recalling that  $\mathbf{u}^{rh} = \mathbf{u}^{j^*}$  where  $j^*$  is the last iteration then we have  $J_{rh} - J_{LP} = \sum_{j=0}^{j^*} \Delta(j) \leq 0$  the latter inequality being a consequence of Lemma 3. Then we have the thesis.

To prove that  $J_{rh} \geq J_{INP}$ , note that  $\mathbf{u}^{rh}$  is feasible for  $(INP)$  from Lemma 2. In other words, this means that  $\mathbf{u}^{rh} \in \mathcal{F}$  which implies  $J_{INP} = \min_{\mathbf{u} \in \mathcal{F}} \mathbf{c}\mathbf{u} \leq \mathbf{c}\mathbf{u}^{rh}$  and this concludes our proof.

To see that the receding horizon algorithm finds  $\mathbf{u}^{rh}$  in the worst-case in  $O(N)$  observe that the worst-case is when we have  $T$  equal to  $N$ . In the latter case it turns  $\alpha(j) = 0$ ,  $\beta(j) = N - 1$ . If  $\delta = 1$  then the algorithm performs  $N$  comparisons between the solutions of  $\mathcal{S}$ .

We next prove that if  $T = N$  then  $\mathbf{u}^{rh}$  is optimal and  $J_{rh} = J_{INP}$ . Actually, by definition we have  $\mathbf{u}^{rh} := u_{[0,N-1]}^*$ . Also observe that in this case  $\beta(0) = N - 1$  and therefore  $u_{[0,N-1]}^*$  is optimal for  $(INP_{[0,N-1]})$ . Invoking the equivalence between  $(INP_{[0,N-1]})$  and  $(INP)$  we can conclude that  $\mathbf{u}^{rh}$  is optimal for  $(INP)$ . As a consequence we have  $J_{rh} = J_{INP}$  and this concludes the proof.

## References

- [1] D. Axehill, L. Vandenberghe, and A. Hansson, “Convex relaxations for mixed integer predictive control”, *Automatica*, vol. 46, 2010, pp. 1540–1545.
- [2] D. Bauso, “Boolean-controlled systems via receding horizon and linear programming” *Mathematics of Control, Signals, and Systems (MCSSS)*, vol. 21, no. 1, 2009, pp. 69–91.

- [3] D. Bauso, “Optimal Impulse Control Problems and Linear Programming”, *Proc. of 48th IEEE Conference on Decision and Control* Shanghai, China, Dec. 2009, pp. 5299–5304.
- [4] D. Bauso, Q. Zhu, and T. Başar. “Mixed Integer Optimal Compensation: Decompositions and Mean-Field Approximations”, *Proc. of the American Control Conference*, 2012, pp. 2663-2668.
- [5] F. Borrelli, M. Baotić, A. Bemporad, and M. Morari, “Dynamic programming for constrained optimal control of discrete-time linear hybrid systems”, *Automatica*, vol. 41, no. 10, 2005, pp. 1709–1721.
- [6] D. D. Botvich and N. G. Duffield, “Large deviations, economies of scale, and the shape of the loss curve in large multiplexers”, *Queueing Systems*, vol. 20, 1995, pp. 293–320.
- [7] M. S. Branicky, V. S. Borkar, and S. K. Mitter, “A Unified Framework for Hybrid Control: Model and Optimal Control Theory”, *IEEE Trans. on Automatic Control*, vol. 43, no. 1, 1998, pp. 31–45.
- [8] C. G. Cassandras, D. L. Pepyne, and Y. Wardi, “Optimal control of a class of hybrid systems”, *IEEE Trans. Automatic Control*, vol. 46, no. 3, 2001, pp. 398–415.
- [9] O. L. V. Costa, “Impulse control of piecewise-deterministic processes via linear programming”, *IEEE Trans. on Automatic Control*, vol. 36, no. 3, 1991, pp. 371–375.
- [10] M.C.F. Donkers, P. Tabuada, and W.P.M.H. Heemels, “Minimum Attention Control for Linear Systems: A Linear Programming Approach”, *Discrete Event Dynamic Systems Theory and Applications*, vol. 24, no. 2, June 2014, pp. 199–218.
- [11] M. Gazarik and Y. Wardi, “Optimal release times in a single server: An optimal control perspective”, *IEEE Trans. on Automatic Control*, vol. 43, no. 7, 1998, pp. 998–1002.
- [12] G.C. Goodwin and D.E. Quevedo, “Finite alphabet control and estimation”, *International Journal of Control, Automation, and Systems*, vol. 1, no. 4, 2003, pp. 412–430.



- [13] Y. Guo, Y. Wang, L. Xie, and J. Zheng, “Stability analysis and design of reset systems: Theory and an application”, *Automatica*, vol. 45, no. 2, 2009, pp. 492–497.
- [14] W. M. Haddad, V. Chellaboina, and S. G. Nersesov, *Impulsive and hybrid dynamical systems*, Princeton Series in Applied Mathematics, Princeton University Press, 2006.
- [15] J. Hespanha, D. Liberzon, and A. Teel, “Lyapunov Characterizations of Input-to-State Stability for Impulsive Systems”, *Automatica*, vol. 44, no. 11, 2008, pp. 2735–2744.
- [16] L. Hetel, J. Daafouz, S. Tarbouriech, and C. Prieur, “Stabilization of linear impulsive systems through a nearly-periodic reset”, *Nonlinear Analysis: Hybrid Systems*, vol. 7, no. 1, 2013, pp. 4-15.
- [17] M. Jeanblanc-Piqué, “Impulse control method and exchange rate”, *Math. Finance*, vol. 3, 1993, pp. 161–177.
- [18] J. W. Lee and G. E. Dullerud, “Joint synthesis of switching and feedback for linear systems in discrete time”, *Proc. of the 14th international conference on Hybrid systems: computation and control*, 2011, pp. 201–210 ACM.
- [19] F. Martinelli, C. Shu, and J. Perkins, “On the optimality of myopic production controls for single-server, continuous flow manufacturing systems”, *IEEE Trans. on Automatic Control*, vol. 46, no. 8, 2001, pp. 1269–1273.
- [20] G. L. Nemhauser and L. A. Wolsey, *Integer and Combinatorial Optimization*, John Wiley & Sons Ltd, New York, 1988.
- [21] S. Monaco and D. Normand-Cyrot, “Advanced Tools for Nonlinear Sampled-Data Systems”, *European Journal of Control*, 13-23, Hermès Sciences, Paris, 2007, pp. 221-241.
- [22] X. Xu and P.J. Antsaklis, “Optimal control of switched systems based on parameterization of the switching instants”, *IEEE Trans. Automatic Control* vol. 49, no. 1, 2004, pp. 475–482.

- [23] W. Zhang, A. Abate, J. Hu, and M. P. Vitus, “Exponential stabilization of discrete-time switched linear systems”, *Automatica*, vol. 45, no. 11, 2009, pp. 2526–2536.