



Latency vs precision: Stability preserving perception scheduling[☆]

Rodrigo Aldana-López^{*}, Rosario Aragüés, Carlos Sagüés

Departamento de Informática e Ingeniería de Sistemas (DIIS) and Instituto de Investigación en Ingeniería de Aragón (I3A), Universidad de Zaragoza, Zaragoza 50018, Spain



ARTICLE INFO

Article history:

Received 21 July 2022
Received in revised form 20 April 2023
Accepted 3 May 2023
Available online xxxx

Keywords:

Perception-latency
Scheduling
Robust control
Uncertain systems

ABSTRACT

In robotic systems, perception latency is a term that refers to the computing time measured from the data acquisition to the moment in which perception output is ready to be used to compute control commands. There is a compromise between perception latency, precision for the overall robotic system, and computational resource usage referred to here as the latency-precision trade-off. In this work, we analyze a robot model given by a linear system, a zero-order hold controller, and measurements taken by several perception mode possibilities with different noise levels. We show that the analysis of this system is reduced to studying an equivalent switching system. Our goal is to schedule perception modes such that stability is attained while optimizing a cost function that models the latency-precision trade-off. Our solution framework comprises three main tools: the construction of perception scheduling policy candidates, admissibility verification for policy candidates, and optimal strategies based on admissible policies.

© 2023 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introduction

One of the key ingredients to achieving autonomy of mobile robots is the perception task, which consists of using sensors to obtain an estimate of the state of the robot and the environment. Then, the state of the system can be used to close a feedback loop in order to make the robot follow the desired behavior (Siegwart & Nourbakhsh, 2004). The term perception latency has been used in Falanga, Kim, and Scaramuzza (2019) to refer to the perception computing time, measured from the data acquisition to the moment in which perception output is ready to be used to compute control commands. In general, using high-resolution images, a large number of features, and robust feature extraction methods will result in a better perception quality at the expense of larger computing times. This trade-off has been verified experimentally for visual odometry, localization, and mapping in Pant et al. (2021) and Strasdat, Montiel, and Davison (2012). In addition, it has been shown that deep learning models used for perception share this type of trade-off as well (Forster, Zhang, Gassner, Werlberger, & Scaramuzza, 2017). However, increasing the perception latency might be undesired since long sampling intervals can prevent the control commands from stabilizing the

system (Pant et al., 2021). In contrast, a smaller latency may be harmful since low-quality measurements lead to faster sampling, which can negatively impact the performance from energetic or CPU load points of view.

Hence, there is a practical compromise between perception latency and the performance of the overall robotic system, from now on referred to as the latency-precision trade-off. The most basic approach in this setting is the following: given a maximum sampling time T_s driving a stable behavior to the robot, use the largest perception latency possible (Pant et al., 2021). In other words, T_s poses a real-time constraint on the perception task of the robot. This approach has been used repeatedly in the robotics literature, e.g. for multi-rotor control using onboard vision sensors (Campos-Macías, Aldana-López, de la Guardia, Parra-Vilchis, & Gómez-Gutiérrez, 2021; Fraundorfer et al., 2012; Oleynikova, Taylor, Siegwart, & Nieto, 2018). As studied here, a different approach to overcoming the latency-precision trade-off involves using different perception modes at different moments.

In this work, we analyze a robot model given by a linear system, a zero-order hold controller, and measurements that can be taken using several perception mode possibilities with different noise levels. The perception latencies chosen through time dictate the sampling instants for perception and control. We show that the analysis of this system is reduced to studying an equivalent switching system. Moreover, we construct a cost function that includes a control precision term as well as a perception latency penalty which can be used to incorporate the energetic and CPU load points of view. Our goal is to schedule perception modes such that asymptotic stability for the expected value of the state

[☆] The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Giordano Sciarciotti under the direction of Editor Sophie Tarbouriech.

^{*} Corresponding author.

E-mail addresses: 794214@unizar.es (R. Aldana-López), raragues@unizar.es (R. Aragüés), csagues@unizar.es (C. Sagüés).

of the system is attained while optimizing for the proposed cost function. Our solution framework is composed of three main tools. First, we propose a class of perception scheduling policy candidates for optimizing the cost of interest. Second, we devise a procedure for checking if policy candidates are admissible from a stability point of view. Third, we propose and study the performance of optimal strategies based on previously constructed scheduling policies.

1.1. Related work

Scheduling between different sensor configurations was studied in van der Weijst, van Loon, Heertjes, and Heemels (2018) for linear control systems. However, the trade-off between measurement quality and latency was not considered. This trade-off has been studied for state estimation in Ballotta, Schenato, and Carbone (2020). In addition, Gatsis, Hassani, and Pappas (2021) studied another related type of latency trade-off arising from the quality of a communication channel in networked systems. Nonetheless, these approaches are mainly focused on the resulting estimation quality rather than the overall performance of a closed-loop system using such an estimation framework. In Pant et al. (2021), a control-estimator co-design is proposed in a periodic setting, where the relationship between latency and estimator quality is modeled and taken into account in a Model Predictive Control (MPC) strategy. Thus, more emphasis is given to the feasibility of an optimization program rather than on ensuring some form of stability outside the sliding window used in the MPC, which is particularly important when switching between different perception modes.

The problem of perception latency scheduling is quite close to the notion of scheduling sampling instants for which some works have used event-triggered and self-triggered sampling (Anta & Tabuada, 2010). In these sampling schemes, a state-dependent triggering rule for sampling is obtained by employing sufficient stability conditions. In particular, Ristevski, Yucelen, and Muse (2021) study variable sampling intervals to model information exchange instants in networked control systems and schedule them accordingly using an event-triggered approach. However, these methods do not use a latency model in which shorter sampling intervals lead to poor state estimates as in the latency-precision trade-off.

Another approach is to model the effect of variable sampling intervals as a time-varying delay (Hetel et al., 2017). However, these works mostly study robustness against arbitrary sampling sequences. On the other hand, as mentioned before, variable sampling problems can be studied using switching systems theory. Switching systems stability results may be divided into two categories. The first one studies stability under arbitrary switching signals. Examples of this type of analysis can be found in Liberzon (2003) and Shorten, Wirth, Mason, Wulff, and King (2007) for continuous-time and in Ahmadi, Jungers, Parrilo, and Roostehani (2014) and Ding, Peres, Ranade, and Zhai (2016) for discrete-time. The switching signal can be considered a state-dependent input in the second category. Examples of the study of stability under state-dependant switching signals can be found in Blanchini and Savorgnan (2008) and Griggs, King, Shorten, Mason, and Wulff (2010) for continuous-time systems and in Antunes and Heemels (2017), Blanchini and Savorgnan (2008), Fiacchini and Jungers (2014), Fiacchini, Jungers, and Girard (2018) and Geromel and Colaneri (2006) for discrete-time systems. However, as we show later, the connection between stability and optimality may not be evident, particularly with the cost function considered in this work. To this regard, the works (Antunes & Heemels, 2017; Wu & He, 2020) deal with the optimality of state-dependent switching signals for some particular forms of cost functions that do not extend to the cost function modeling the latency-precision trade-off as we propose.

1.2. Notation

Through this work, $\mathbf{x}(t) \in \mathbb{R}^n$ represents a continuous-time signal evaluation at $t \in \mathbb{R}$. Moreover, $\mathbf{x}[k] := \mathbf{x}(\tau_k)$ stands for the discrete-time signal obtained by evaluating $\mathbf{x}(t)$ under a particular sequence of time instants $\{\tau_k\}_{k=0}^{\infty}$ with $\tau_k \in \mathbb{R}$ and $\tau_k < \tau_{k+1}$. Furthermore, $\text{tr}(\bullet)$ stands for the trace of a matrix. Let $\mathbf{z} \in \mathbb{R}^n$, then $\text{diag}(\mathbf{z}) \in \mathbb{R}^{n \times n}$ represents the diagonal matrix with the components of \mathbf{z} as the diagonal elements. The symbol $\mathcal{N}(\bar{\mathbf{x}}, P)$ stands for a multi-variate Gaussian distribution with mean $\bar{\mathbf{x}} \in \mathbb{R}^n$ and covariance matrix $P \in \mathbb{R}^{n \times n}$. For any finite set S we denote its cardinality with $|S|$, and for any finite sequence $a = \{a_i\}_{i=1}^m$, we denote with $\text{len}(a) = m$ its length. Let ∂S and $\text{int}(S) = S \setminus \partial S$ be the boundary and the interior operators for any set $S \subset \mathbb{R}^n$. In addition, given a finite set Γ , we usually denote with $\Gamma' \subseteq \Gamma$ an arbitrary element of the power set $\mathcal{P}(\Gamma)$.

2. Problem statement

Consider a mobile robot described by the model

$$d\mathbf{x}(t) = (A\mathbf{x}(t) + B\mathbf{u}[k]) dt + d\mathbf{w}(t) \quad (1)$$

for intervals of the form $t \in [\tau_k, \tau_{k+1})$ where $\mathbf{x}(t) \in \mathbb{R}^n$ is the robot state (e.g. position and velocity for a particle robot), $\mathbf{u}[k] \in \mathbb{R}^{n_u}$ is a zero-order hold input (e.g. representing the actuator forces induced to the system by the robot) and a sequence of instants $\tau = \{\tau_k\}_{k=0}^{\infty}$ with $\tau_k \in \mathbb{R}$, $\tau_k < \tau_{k+1}$ and $\tau_0 \equiv 0$. In addition, $\mathbf{w}(t)$ is an n -dimensional stochastic process capturing disturbances and non-modeled dynamics.

Assumption 1. The initial condition of (1) satisfies $\mathbf{x}(0) \sim \mathcal{N}(\bar{\mathbf{x}}_0, P_0)$. Moreover, $\mathbf{w}(t)$ is a Wiener process with covariance function $\mathbb{E}\{\mathbf{w}(s)\mathbf{w}(r)^T\} = W_0 \min(s, r)$ and W_0 positive semi-definite (Åström, 1970, Page 63).

We are interested in studying the error dynamics of the actual robot state with respect to a desired behavior. Hence, to simplify the exposition, $\mathbf{x}(t)$ will represent such error which is desired to be maintained as close as possible to the origin. However, motivated by the perception latency problem, we consider that at $t = \tau_k$ a state estimate cannot be obtained directly. Instead, indirect information is obtained (e.g. an image captured by a camera) that has to be processed through a perception procedure. To do so, there are D available perception modes where the quality of their estimates depends on the perception latency $\Delta^1, \dots, \Delta^D$ for each perception mode.

Assumption 2. Let $C \in \mathbb{R}^{n_z \times n}$ be the measurement matrix for the perception setting with the pair (A, C) observable and $p_k \in \{1, \dots, D\}$ be the perception mode chosen at $t = \tau_k$. Then, the output of the perception method is $\mathbf{z}[k] = C\mathbf{x}[k] + \mathbf{n}[k]$, available at $t = \tau_k + \Delta^{p_k}$, with noise $\mathbf{n}[k] \sim \mathcal{N}(0, \Sigma^{p_k})$. Here, $\Sigma^1, \dots, \Sigma^D$ capture the different noise levels of the accuracy of methods $1, \dots, D$.

The setting in which different perception modes are available, achieving different noise levels and perception latency, captures the actual practical behavior of usual perception algorithms when different configurations are available (Ballotta et al., 2020; Falanga et al., 2019; Pant et al., 2021; Strasdat et al., 2012). The values of the latency Δ^{p_k} and noise level Σ^{p_k} for a concrete perception method p_k can be obtained offline through a statistical analysis of its performance. In practice, it is expected that a longer latency Δ^{p_k} results in a more concentrated distribution $\mathcal{N}(0, \Sigma^{p_k})$. For instance, in Ballotta et al. (2020), Σ^{p_k} is modeled to be inversely proportional to the latency, which is also supported by the experimental data in Pant et al. (2021) and Strasdat et al.

(2012). However, to allow our methods to be tailored to any latency-precision model, we do not assume a particular relation between the latency Δ^{p_k} and its corresponding noise level Σ^{p_k} but that these might be different between perception methods.

Due to [Assumption 2](#) the only available information at $t = \tau_k$ is $\{\mathbf{z}[0], \dots, \mathbf{z}[k-1]\}$. Hence, a control is designed as $\mathbf{u}[k] = L^{p_k} \hat{\mathbf{x}}[k|k-1]$ where $\hat{\mathbf{x}}[k|k-1] = \mathbb{E}\{\mathbf{x}[k] \mid \mathbf{z}[0], \dots, \mathbf{z}[k-1]\}$ is the conditional mean of $\mathbf{x}[k]$ using old measurements as given in [Appendix A](#). Moreover, the gains L^1, \dots, L^D are assumed to be given, which may have been designed for each latency by separate, provided that the pair (A, B) is controllable. Furthermore, note that at $t = \tau_k + \Delta^{p_k}$ the robot is ready to take a new measurement, thus we set $\tau_{k+1} = \tau_k + \Delta^{p_k}$. Finally, any (perhaps infinite) sequence of perception modes p is referred to as a *perception schedule*. We refer to a rule that generates a perception schedule p based on state information as a *scheduling policy*.

In this context, the goal of this work is the following:

Problem 3 (Perception Scheduling Problem). Consider an interval of interest $[0, T_f]$ and $\text{att}(p; [0, T_f])$, referred to as the *attention* of p , the amount of sampling instants in the interval $[0, T_f]$ induced by a perception schedule p . Moreover, let $0 < r^{p_k}$ be the associated penalty for each perception mode. Thus, given [Assumptions 1](#) and [2](#), and given $\lambda_x, \lambda_r, T_f > 0$ and Q, Q_f positive semi-definite, the problem is to find a perception schedule $p = \{p_k\}_{k=0}^\infty$ such that

$$\begin{aligned} \mathcal{J}(p; \bar{\mathbf{x}}_0, P_0) &= \frac{\lambda_r}{T_f} \sum_{k=0}^{\alpha-1} r^{p_k} \\ &+ \lambda_x \mathbb{E} \left\{ \frac{1}{T_f} \int_0^{T_f} \mathbf{x}(t)^T Q \mathbf{x}(t) dt + \mathbf{x}(T_f)^T Q_f \mathbf{x}(T_f) \right\} \end{aligned} \quad (2)$$

with $\alpha = \text{att}(p; [0, T_f])$, is minimized for system (1). Moreover, if there exists a stabilizing perception schedule for system (1) under arbitrary T_f , then p must induce $\lim_{t \rightarrow \infty} \mathbb{E}\{\mathbf{x}(t)\} = 0$.

As usual in minimum variance control problems ([Åström, 1970](#), Page 172) the persistent introduction of uncertainty originated from disturbances in the model, and measurement noise prevents a linear controller from making $\mathbf{x}(t)$ converge to the origin in the mean-squared error sense. Hence, to concentrate our efforts on building a schedule p , it is more practical to ensure convergence of $\mathbb{E}\{\mathbf{x}(t)\}$ towards the origin, and deal with the effect of the second order moment of $\mathbf{x}(t)$ during the finite interval of interest $[0, T_f]$ by minimizing (2).

The cost in (2) is meant to model the latency-precision trade-off we described before in the following way. The second term in (2) is composed of an expected quadratic penalty for cost $\mathbf{x}(t)$ of (1), which penalizes deviations from the origin. On the other hand, the first term in (2) is the accumulation of the penalties r^{p_0}, r^{p_1}, \dots over the interval $[0, T_f]$. If $r^{p_0} = r^{p_1} = \dots = 1$, this term is proportional to the attention of p . Keeping the number of sampling events small is desirable both from an energetic point of view and to minimize the use of I/O buses when using the sensors. On the other hand, consider that the latency Δ^{p_k} is not used exclusively for perception, but also that the method frees the computing unit for an interval of length $(1 - f^{p_k})\Delta^{p_k}$ with $f^{p_k} \in (0, 1)$. Thus, the *perception CPU load* in the interval $[0, T_f]$ is $(1/T_f) \sum_{k=0}^{\alpha-1} f^{p_k} \Delta^{p_k}$ and can be taken into account in (2) by using $r^{p_k} = f^{p_k} \Delta^{p_k}$.

Consider the last requirement in [Problem 3](#). Note that the cost in (2) only penalizes the schedule during the interval $[0, T_f]$, which can be used to penalize a transient response. In practice, it may be beneficial to pose a similar problem once $[0, T_f]$ has elapsed in a moving horizon fashion. However, this strategy will

not ensure asymptotic stability for $\mathbb{E}\{\mathbf{x}(t)\}$ on the long run for arbitrary values of $\lambda_x, \lambda_r, T_f, Q, Q_f$ since the individual terms in (2) are often conflicting. As an example, take $\lambda_x = 0$ and $\lambda_r > 0$ in which the system's error is not penalized and thus, minimizing (2) will not ensure stability regardless of T_f or how often the optimization problem is solved. Hence, a connection between optimality and stability is not evident in general problem setting.

Our strategy is based on the following observation. Using [Proposition 24](#) in [Appendix A](#), it is obtained that $\bar{\mathbf{x}}[k] := \mathbb{E}\{\mathbf{x}[k]\}$ is given by

$$\bar{\mathbf{x}}[k+1] = \Lambda(\Delta^{p_k})\bar{\mathbf{x}}[k], \quad \bar{\mathbf{x}}[0] = \bar{\mathbf{x}}_0 \quad (3)$$

where $\Lambda(\Delta^{p_k}) := \exp(A\Delta^{p_k}) + \int_0^{\Delta^{p_k}} \exp(A\tau) d\tau B L^{p_k}$. As a result, (3) is a switched system which switches between matrices $\{\Lambda(\Delta^1), \dots, \Lambda(\Delta^D)\}$ according to the perception schedule p as the switching signal. Hence, the asymptotic stability character of [Problem 3](#) is tied to the stability of the switching system in (3). In this context, the outline of our solution has 3 main components:

- First, in [Section 3](#) we analyze some results in the literature regarding stabilizing switching signals for discrete-time systems. We provide an extension of the results found in the literature to enable the construction of multiple scheduling policy candidates for optimizing (2). These scheduling policies are all stabilizing for (3) when an admissibility condition is attained.
- Second, for completeness, in [Section 4](#) we provide a new algorithm to check if a scheduling policy candidate is admissible.
- Third, in [Section 5](#) we provide a new sub-optimal algorithm for [Problem 3](#) based on the multiple scheduling policies previously constructed. We analyze its theoretical performance and discuss some heuristics and approximations.

3. Stability preserving perception schedules

In this section, we develop scheduling policies which ensure (3) is stable. In [Fiacchini and Jungers \(2014\)](#) an interesting framework is proposed which provides sufficient and necessary conditions for asymptotic stability of discrete-time switching linear systems as (3), and generates state-dependant switching laws based on such conditions. In the following, we analyze some related ideas and propose a framework suitable for [Problem 3](#). This is required since not only a single switching rule is needed, but several switching rule possibilities have to be available as the search space for the optimization of (2). First, let us introduce the following concept:

Definition 4. A set of schedules is any set of the form $\Gamma = \{\gamma^1, \dots, \gamma^{|\Gamma|}\}$ with $|\Gamma| < \infty$ where $\gamma^i \in \{1, \dots, D\}^{\text{len}(\gamma^i)}$ are individual schedules with $\text{len}(\gamma^i) < \infty, \forall i \in \{1, \dots, |\Gamma|\}$.

We aim to construct stabilizing perception schedules by piecing together individual finite-length schedules found in a set of schedules. Up to this point, sets of schedules are arbitrary and can be constructed randomly. However, stabilizing switching signals cannot arise for any set of schedules. In the following, we study which sets of schedules induce stability. Consider the following objects: the hyper-ellipse

$$\mathbb{S}_0 := \{\mathbf{x} \in \mathbb{R}^n : \mathbf{x}^T M_0 \mathbf{x} \leq 1\}$$

for some arbitrary positive definite matrix M_0 and

$$\mathbb{S}_\gamma := \{\mathbf{x} \in \mathbb{R}^n : \mathbf{x} = \Lambda(\Delta^{\gamma_0})^{-1} \dots \Lambda(\Delta^{\gamma_{\ell-1}})^{-1} \mathbf{y}, \mathbf{y} \in \mathbb{S}_0\}$$

where $\gamma := \{\gamma_k\}_{k=0}^{\ell-1}$ is a schedule of finite length $\text{len}(\gamma) = \ell$. Thus, by construction, for any $\bar{\mathbf{x}}_0 \in \mathbb{S}_\gamma$, then $\bar{\mathbf{x}}[\text{len}(\gamma)] \in \mathbb{S}_0$ for such schedule γ . This is verified as:

$$\begin{aligned} \bar{\mathbf{x}}[\text{len}(\gamma)] &= \Lambda^\gamma \bar{\mathbf{x}}_0 \in \Lambda^\gamma \mathbb{S}_\gamma \\ &= \{\mathbf{z} \in \mathbb{R}^n : \mathbf{z} = \Lambda^\gamma \mathbf{x}, \mathbf{x} = (\Lambda^\gamma)^{-1} \mathbf{y}, \mathbf{y}^T M_0 \mathbf{y} \leq 1\} \\ &= \{\mathbf{z} \in \mathbb{R}^n : \mathbf{z} = \Lambda^\gamma (\Lambda^\gamma)^{-1} \mathbf{y}, \mathbf{y}^T M_0 \mathbf{y} \leq 1\} \\ &= \{\mathbf{z} \in \mathbb{R}^n : \mathbf{z}^T M_0 \mathbf{z} \leq 1\} = \mathbb{S}_0 \end{aligned}$$

with $\Lambda^\gamma := \Lambda(\Delta^{\gamma_{\ell-1}}) \cdots \Lambda(\Delta^{\gamma_0})$ as the multiplication chain in (3) for the schedule γ . Moreover, note that \mathbb{S}_γ has the shape of an hyper-ellipse defined by the positive definite matrix $M_\gamma = (\Lambda^\gamma)^T M_0 (\Lambda^\gamma)$ as $\mathbb{S}_\gamma \equiv \{\mathbf{x} \in \mathbb{R}^n : \mathbf{x}^T M_\gamma \mathbf{x} \leq 1\}$.

The key idea is the following. Consider Γ to be a set of schedules γ . Thus, if the interior of the set

$$\mathbb{S}_\Gamma^* := \bigcup_{\gamma \in \Gamma} \mathbb{S}_\gamma \quad (4)$$

contains \mathbb{S}_0 , it means that for each initial condition, there exists a schedule $\gamma \in \Gamma$ which contracts \mathbb{S}_Γ^* into itself after $\text{len}(\gamma)$ steps. This is, if $\bar{\mathbf{x}}_0 \in \mathbb{S}_\Gamma^*$, thus $\bar{\mathbf{x}}[\text{len}(\gamma)] \in \mathbb{S}_0 \subset \text{int}(\mathbb{S}_\Gamma^*)$ for some $\gamma \in \Gamma$.

Definition 5 (Admissible Set Of Schedules). A finite set of schedules Γ is admissible if $\mathbb{S}_0 \subset \text{int}(\mathbb{S}_\Gamma^*)$.

As we show in a subsequent result, if Γ is admissible, a contracting schedule can be found $\forall \bar{\mathbf{x}} \in \mathbb{R}^n$ as:

$$\gamma^*(\bar{\mathbf{x}}; \Gamma) := \arg \min_{\gamma \in \Gamma} \{r \in \mathbb{R} : r = \bar{\mathbf{x}}^T M_\gamma \bar{\mathbf{x}}\} \quad (5)$$

Using this switching rule for various admissible sets of schedules $\Gamma^1, \dots, \Gamma^m$, the following strategy can be used online to compute a stabilizing perception method at each $t = \tau_k$ given $\bar{\mathbf{x}}[k]$.

Definition 6. [Stability Preserving Scheduling] The SP² (Stability Preserving Scheduling Policy) strategy is defined by Algorithm 1.

Algorithm 1 SP² strategy

Input: $\bar{\mathbf{x}}_0$.

Output: Perception schedule p .

```

1:  $\gamma \leftarrow \emptyset$ .
2:  $i \leftarrow 0$ 
3: for each instant  $\tau_k, k = 0, 1, \dots$  do
4:   if ( $i < \text{len}(\gamma)$ ) then
5:      $p_k \leftarrow \gamma_i$  #For chosen schedule  $\gamma$ , traverse its elements
        $\gamma_0, \dots, \gamma_{\text{len}(\gamma)}$ .
6:      $i \leftarrow i + 1$ 
7:   else
8:      $\Gamma \leftarrow$  any set of schedules chosen from the admissible
       options  $\Gamma^1, \dots, \Gamma^m$ .
9:      $\gamma \leftarrow \gamma^*(\bar{\mathbf{x}}[k]; \Gamma)$  from (5). #Once previous schedule has
       been consumed, choose a new schedule
10:     $i \leftarrow 0$ 
11:   end if
12: end for

```

Remark 7. A similar rule to (5) was studied in Fiacchini and Jungers (2014) for a particular fixed set of schedules Γ constructed through a combinatorial approach. This is, all possible combinations of elements in $\{1, \dots, D\}$ up to the first length at which admissibility is attained (Fiacchini & Jungers, 2014, Algorithm 1). As a result, a single scheduling policy is considered regardless of the choice of (2), and with no additional degrees of freedom to improve performance. Unlike the previous approach,

Algorithm 2 Construction of a set of schedules

Input: ℓ, D .

Output: Γ .

```

1:  $\Gamma = \emptyset$ 
2: repeat
3:   if ( $\{1, \dots, D\}^\ell \setminus \Gamma = \emptyset$ ) then
4:     # If all schedules have been evaluated
5:     Increase  $\ell$ 
6:   end if
7:   # Generate a random sequence over  $\{1, \dots, D\}$  with
     random length  $\ell' \leq \ell$ .
8:    $\ell' \leftarrow \text{randomSample}(\{1, \dots, \ell\})$ 
9:    $\Gamma \leftarrow \Gamma \cup \text{randomSample}(\{1, \dots, D\}^{\ell'} \setminus \Gamma)$ 
10: until  $\mathbb{S}_0 \subset \text{int}(\mathbb{S}_\Gamma^*)$  # Using Algorithm 3

```

we introduce an extra degree of freedom by constructing multiple sets of schedules $\Gamma^1, \dots, \Gamma^m$. Then, as explained in detail in Section 5, a switching law of the form (5) is used for each set of schedules as scheduling policy candidates to optimize for (2). This makes our approach more beneficial for the perception scheduling problem since intuitively, given an initial condition $\bar{\mathbf{x}}[0]$, the approach with a single set of schedules will result in a fixed schedule p , regardless of the cost (2) whereas our method can adapt p to the cost.

In this sense, an alternative for the combinatorial approach is given in Algorithm 2 in which random schedules are appended to Γ . These schedules are generated with length up to a given value ℓ , unless all schedules are contained in Γ , in which case ℓ is increased. This procedure terminates once Γ is admissible and, similarly to the combinatorial approach, this must happen if there exists a stabilizing schedule for (3). When compared to the combinatorial approach, the advantage of using Algorithm 2 is two-fold. First, depending on the initial value of the length ℓ , longer sequences can be tested earlier. This is complicated using the combinatorial approach due to the combinatorial explosion involved in such a strategy. Note that even if the combinatorial approach is used starting from an initial length $\ell > 1$ to test longer sequences earlier, this strategy will neglect short sequences, which may also be useful for rapid decision-making if available. Hence, Algorithm 2 covers short and long sequences from the beginning. Moreover, Algorithm 2 allows us to generate multiple admissible sets of schedules, namely $\Gamma^1, \dots, \Gamma^m$ useful for our optimization strategy.

However, since in line 8 of Algorithm 1 we allow to change the set of schedules Γ to any admissible one between the m options $\{\Gamma^1, \dots, \Gamma^m\}$, the scheduling policy in (5) may be different each time in line 9 of Algorithm 1 is executed. The following results show that even in this case, Algorithm 1 manages to stabilize (3).

Theorem 8. Assume that there exists at least one admissible set of schedules and that the SP² strategy is used in (3). Thus $\lim_{k \rightarrow \infty} \bar{\mathbf{x}}[k] = 0$.

Proof. All proofs are provided in Appendix C.

Checking if a set of schedules is admissible is not a trivial task. Fiacchini and Jungers (2014, Remark 6) propose to check if there is at least one $\gamma \in \Gamma$ such that $\mathbb{S}_0 \subset \text{int}(\mathbb{S}_\gamma)$ as a sufficient condition for admissibility. However, this is a very conservative condition that will not be attained in general, even when Γ is admissible. Motivated by this, in the following section, we provide a novel analysis to check if a set of schedules is admissible.

4. Non-conservative admissibility checking

The basis of a non-conservative admissibility checking is formulated as a nonlinear program as follows.

Theorem 9. *Let*

$$R := \min_{\bar{\mathbf{x}} \in \partial \mathbb{S}_r^*} \bar{\mathbf{x}}^T M_0 \bar{\mathbf{x}}. \quad (6)$$

Then, $\mathbb{S}_0 \subset \mathbb{S}_r^*$ if and only if $R > 1$.

Hence, under the light of [Theorem 9](#), studying if Γ is admissible is equivalent to solving the nonlinear program in (6) and checking if $R > 1$. Note that the attempt to solve numerically for local minima in (6) may fail since the condition $R > 1$ is only useful when the global minimum is considered. To the best of our knowledge, there is no previous work which solves for the global minimum of a non-convex program of the form (6). Thus, the following efforts are dedicated to this task.

Note that most of the theory presented up to now remains the same even when \mathbb{S}_0 is not a hyper-ellipse. An arbitrary C^* set as in [Definition 25](#) in [Appendix B](#) can be used instead of a hyper-ellipse. For example, polyhedral sets are also considered in [Fiacchini and Jungers \(2014\)](#). However, note that the set $\partial \mathbb{S}_r^*$ may be non-convex, implying that there might be many local minima for (6), which complicates the analysis. Despite this, we show that for the case where \mathbb{S}_0 is a hyper-ellipse, we can compute the global minimum of (6). The strategy is the following: we split the nonlinear program in (6) into subprograms of the following form:

$$\min_{\bar{\mathbf{x}} \in \mathbb{R}^n} \bar{\mathbf{x}}^T M_0 \bar{\mathbf{x}} \quad (7)$$

$$\text{s.t. : } h_\gamma(\bar{\mathbf{x}}) := \bar{\mathbf{x}}^T M_\gamma \bar{\mathbf{x}} - 1 = 0, \quad \forall \gamma \in \Gamma' \in \mathcal{P}(\Gamma)$$

where Γ' is any subset of Γ , i.e. Γ' is in the power set $\mathcal{P}(\Gamma)$. Intuitively, the sub-program in (7) is useful since due to the shape of $\partial \mathbb{S}_r^*$, the global optimum of (6) must lie either in the boundary of a single hyper ellipse \mathbb{S}_γ for some $\gamma \in \Gamma$ which has the form $h_\gamma(\bar{\mathbf{x}}) = 0$, or in the intersection of the boundaries of multiple hyper-ellipses. As an example, consider $\Gamma = \{p, q\}$ with arbitrary schedules p, q . Hence, the global optimum of (6) must be the global optimum of (7) with either $\Gamma' = \{p\}$, $\Gamma' = \{q\}$, which correspond to checking points in $\partial \mathbb{S}_p, \partial \mathbb{S}_q$ by separate or $\Gamma' = \{p, q\}$ which corresponds to checking points in $\partial \mathbb{S}_p \cap \partial \mathbb{S}_q$ where the two constraints $h_p(\bar{\mathbf{x}}) = 0, h_q(\bar{\mathbf{x}}) = 0$ are active. Therefore, the solution for (6) comes from (7) for some $\Gamma' \in \mathcal{P}(\Gamma) = \{\emptyset, \{p\}, \{q\}, \{p, q\}\}$. This idea is formalized in the following:

Corollary 10. *Let $\bar{\mathbf{X}}_{\Gamma'}$ be the set of all critical points (local minima candidates) $\bar{\mathbf{x}}^*$ of (7) such that $\bar{\mathbf{x}}^* \notin \mathbb{S}_\gamma$ for any $\gamma \in \Gamma \setminus \Gamma'$ and any $\Gamma' \in \mathcal{P}(\Gamma)$. Thus, the global minimum R of (6) can be obtained as*

$$R = \min \left\{ r = \bar{\mathbf{x}}^T M_0 \bar{\mathbf{x}} : \bar{\mathbf{x}} \in \bigcup_{\Gamma' \in \mathcal{P}(\Gamma)} \bar{\mathbf{X}}_{\Gamma'} \right\} \quad (8)$$

[Corollary 10](#) implies that the solution of (6) can be obtained by solving programs of the form (7), where critical points of (7) are rejected if such points are contained in other hyper-ellipses not involved in (7) (written as $\bar{\mathbf{x}}^* \notin \mathbb{S}_\gamma, \forall \gamma \in \Gamma \setminus \Gamma'$ in [Corollary 10](#)). Moreover, all programs of the form (7) involve from one hyper-ellipse at the time, to all combinations of $\mathbb{S}_\gamma, \forall \gamma \in \Gamma$.

Remark 11. Note that if $|\Gamma'| = n$ in (7), the set of all points $\bar{\mathbf{x}}$ with $\bar{\mathbf{x}}^T M_\gamma \bar{\mathbf{x}} = 1, \forall \gamma \in \Gamma'$ contains only isolated points for almost all $\{M_\gamma\}_{\gamma \in \Gamma'}$. Moreover, the set of equations $\bar{\mathbf{x}}^T M_\gamma \bar{\mathbf{x}} = 1$ are n polynomial equations with n variables which have n^n complex solutions due to the Bezout's Theorem ([Kubler, Renner,](#)

[& Schmedders, 2014](#), Theorem 4.14). Homotopy solvers such as the one in [Verschelde \(1999\)](#) can track all such real solutions for polynomial systems or obtained explicitly if $n \leq 2$. Hence, all reals solutions can be checked directly to build the set $\bar{\mathbf{X}}_{\Gamma'}$. In contrast, the case with $|\Gamma'| > n$ can be ignored since those solutions are either contained in the $|\Gamma'| = n$ case or are nonexistent.

In general, an optimization program of the form (7) can be solved by the Lagrange multipliers method. However, depending on the problem constraints, in this case, given in the form $h_\gamma(\bar{\mathbf{x}}) = 0$, not all local minima can be found through this method. Such local minima are often called non-regular ([Bertsekas, 1999](#), Page 279). Due to the fact that we look for the global optimum of (7), we are forced to analyze both regular and non-regular points. Formally, critical points in $\bar{\mathbf{X}}_{\Gamma'}$ for (7) in which $|\Gamma'| < n$ will be classified into regular or non-regular points for their analysis as follows:

Definition 12 ([Bertsekas, 1999](#), Page 279). Acritical point $\bar{\mathbf{x}}$ for (7) is said to be regular if the vectors $\nabla h_\gamma(\bar{\mathbf{x}}) = 2M_\gamma \bar{\mathbf{x}}, \gamma \in \Gamma'$ are linearly independent.

In the next section, we use the Lagrange multiplier method to find the list of critical regular points for (7). Then, we use a tailored analysis for the non-regular ones.

4.1. Regular points analysis

We start the study of (7) with the characterization of regular points by means of the Lagrange multiplier theorem in [Bertsekas \(1999, Proposition 3.1.1\)](#).

Theorem 13. *Let $\lambda := \{\lambda_\gamma\}_{\gamma \in \Gamma'}$ be a solution to the system of equations*

$$\text{tr}(G(\lambda)^\dagger (M_\gamma - M_\nu)) = 0, \quad \gamma, \nu \in \Gamma' \quad (9a)$$

$$\det G(\lambda) = 0 \quad (9b)$$

where

$$G(\lambda) := M_0 + \sum_{\gamma \in \Gamma'} \lambda_\gamma M_\gamma \quad (10)$$

and $G(\lambda)^\dagger$ is the adjugate matrix of $G(\lambda)$ ([Horn & Johnson, 2012, Page 22](#)). Thus, all regular critical points $\bar{\mathbf{x}}^*$ of program (7) are in the kernel of $G(\lambda)$ for some λ . This is, there exists a solution λ of (9) such that $G(\lambda)\bar{\mathbf{x}}^* = 0$ and have $(\bar{\mathbf{x}}^*)^T M_0(\bar{\mathbf{x}}^*) = -\sum_{\gamma \in \Gamma'} \lambda_\gamma$.

Remark 14. Note that the number of equations in (9) are evidently more than $|\Gamma'|$. However, most of them are linearly dependent. They can be arranged as $|\Gamma'|$ independent equations with $|\Gamma'|$ variables by using $|\Gamma'| - 1$ equations of the form (9a) in addition to (9b). These $|\Gamma'|$ equations are independent for most choices of $\{M_\gamma\}_{\gamma \in \Gamma'}$. Moreover, note that all equations in (9) are polynomials in λ and therefore, the number of solutions can be counted by means of Bezout's theorem and obtained similarly as pointed out in [Remark 11](#) using the algorithm in [Verschelde \(1999\)](#).

Moreover, the nullity (dimension of kernel) of the singular matrix $G(\lambda)$ may be of dimension up to n . However, among the possible singular matrices, ones with nullity greater than one lie in a set of measure zero (with respect to the possible choices of $\{M_\gamma\}_{\gamma \in \Gamma'}$). This fact is evidenced in the following result.

Corollary 15. *Let λ satisfy (9), assume that $G(\lambda)$ has nullity greater than 1, and let $G_{ij}(\lambda)$ be the sub-matrix obtained by deleting the i th row and j th column of $G(\lambda)$. Then, λ complies with*

$$\det G_{ij}(\lambda) = 0, \quad \forall (i, j) \in \{1, \dots, n\}^2 \quad (11)$$

As a result of [Corollary 15](#), the case in which $G(\lambda)$ has nullity greater than one is not common in a general setting since $\lambda \in \mathbb{R}^{|\Gamma' |}$ would have to comply $|\Gamma'| - 1$ equations of the type [\(9a\)](#) in addition to n^2 equations of the type [\(11\)](#). Henceforth, we ignore such cases in our analysis. Now, for $G(\lambda)$ with nullity 1, let \mathbf{g} be any vector in the kernel of $G(\lambda)$. Thus, as a result of [Theorem 13](#) the critical regular points for such λ are parallel to \mathbf{g} , and can be uniquely computed by scaling \mathbf{g} in order to comply $(\bar{\mathbf{x}}^*)^T M_0 \bar{\mathbf{x}}^* = -\sum_{\gamma \in \Gamma'} \lambda_\gamma$ as:

$$\bar{\mathbf{x}}^* = \pm \mathbf{g} \sqrt{\frac{-\sum_{\gamma \in \Gamma'} \lambda_\gamma}{\mathbf{g}^T M_0 \mathbf{g}}} \quad (12)$$

and therefore, the condition $\bar{\mathbf{x}}^* \in \mathcal{S}_\gamma$ for $\gamma \in \Gamma \setminus \Gamma'$ can be checked by computing if $(\bar{\mathbf{x}}^*)^T M_\gamma (\bar{\mathbf{x}}^*) > 1$ in order to build the set $\bar{\mathbf{X}}_{\Gamma'}$ as required in [Corollary 10](#).

4.2. Non-regular points analysis

Non-regular points for program [\(7\)](#) are points $\bar{\mathbf{x}}$ which comply $\bar{\mathbf{x}}^T M_\gamma \bar{\mathbf{x}} = 1, \forall \gamma \in \Gamma'$ and the vectors $\{M_\gamma \bar{\mathbf{x}}\}_{\gamma \in \Gamma'}$ linearly dependent. In general, linearly dependent vectors are rare among all possible vectors, and hence is not surprising that non-regular points will not exist in arbitrary settings. In the following result, we characterize when they exist by obtaining what additional concrete conditions must be satisfied.

Proposition 16. *Let $W(\bar{\mathbf{x}})$ an $n \times |\Gamma'|$ matrix whose columns are the vectors $M_\gamma \bar{\mathbf{x}}, \forall \gamma \in \Gamma'$ and let $W_\alpha(\bar{\mathbf{x}})$ be the $|\Gamma'| \times |\Gamma'|$ sub-matrix of $W(\bar{\mathbf{x}})$ obtained by deleting $n - |\Gamma'|$ rows indexed in $\alpha \in \mathcal{A} \subset \{1, \dots, n\}^{n-|\Gamma'|}$ with \mathcal{A} containing all elements in $\{1, \dots, n\}^{n-|\Gamma'|}$ with non repeating entries. Therefore, any non-regular point for program [\(7\)](#) must comply with*

$$\bar{\mathbf{x}}^T M_\gamma \bar{\mathbf{x}} = 1, \quad \forall \gamma \in \Gamma' \quad (13a)$$

$$\det W_\alpha(\bar{\mathbf{x}}) = 0, \quad \forall \alpha \in \mathcal{A} \quad (13b)$$

Note that the amount of equations of the type [\(13b\)](#) are equivalent to the number of sub-matrices of size $|\Gamma'| \times |\Gamma'|$ inside $W(\bar{\mathbf{x}})$ which is exactly $n! / (|\Gamma'|!(n - |\Gamma'|)!)$ ([Horn & Johnson, 2012, Page 21](#)). Note that there are at least $n - |\Gamma'| + 1$ independent equations of this type since that correspond to the number of $|\Gamma'| \times |\Gamma'|$ sub-matrices of $W(\bar{\mathbf{x}})$ with consecutive rows. Hence, the number of equations in [\(13\)](#) are $|\Gamma'|$ of the type [\(13a\)](#) and at least $n - |\Gamma'| + 1$ of type [\(13b\)](#). Equivalently, a non-regular point $\bar{\mathbf{x}} \in \mathbb{R}^n$ would have to comply with at least $n + 1$ polynomial equations. One could solve n equations picked from [\(13\)](#) and check if the results comply with the remaining equations. However, due to this over-determined feature of equations in [\(13\)](#) non-regular points will not exist in general.

4.3. Admissibility checking algorithm

Using the results in the previous sections, we summarize the methodology for admissibility checking in the following. Consider [Algorithm 3](#) which takes as an input the dimension of the state space n , the matrix M_0 , the set of matrices $M := \{M_\gamma\}_{\gamma \in \Gamma}$ and the set of schedules Γ . This function builds a set \mathcal{R} of all global minima candidates of [\(6\)](#) and checks $\min \mathcal{R} > 1$ as this condition is equivalent to admissibility due to [Theorem 9](#). In line 2, all subsets of schedules in Γ are checked, ranging from no schedules at all, one schedule at the time, pairs of schedules, and so on. In line 3 we check if the constraint space is of dimension n , since in such case the constraint space is comprised of just isolated points, and we obtain critical points as in [Remark 11](#) by calling `IsolatedSolutions`. Otherwise, we obtain regular values by

calling `RegularSolutions`. Cases in which the constraint space is of dimension more than n are ignored. [Algorithm 4](#) obtains critical objective values when the constraint space contains only isolated points. This algorithm looks for real solutions of the system of n equations and n variables $\{\bar{\mathbf{x}}^T M_\gamma \bar{\mathbf{x}} = 1, \forall \gamma \in \Gamma\}$ in line 1. Then, for all solutions we check one by one if they are not contained in other hyper-ellipses for $\gamma \in \Gamma \setminus \Gamma'$. If any of those conditions are complied we reject such points in line 3 in order to build the set $\bar{\mathbf{X}}_{\Gamma'}$ required in [Corollary 10](#). [Algorithm 5](#) looks for regular points of [\(6\)](#). We build a set of critical objective values in \mathcal{R} and then return $r \leftarrow \min \mathcal{R}$. First, we solve the system of equations in [\(9\)](#). Then, for all real solutions λ of such polynomial equations we compute $G(\lambda)$ and the critical point as in [\(12\)](#). These points are rejected if they are contained in other hyper-ellipses $\Gamma \setminus \Gamma'$ in line 7. Then, the critical objective is given by $-\sum_{\gamma \in \Gamma'} \lambda_\gamma$ as obtained in [Theorem 13](#).

Remark 17. Note that in [Algorithm 3](#) we explicitly assumed that $G(\lambda)$ has nullity 1 and that there are no non-regular points. This assumption is reasonable since the system of equations made by [\(9a\)](#) in addition to [\(11\)](#) and the system of equations in [\(13\)](#) are over-determined for almost any $\{M_\gamma\}_{\gamma \in \Gamma}$.

Remark 18. Note that due to line 2 in [Algorithm 3](#), the complexity of the admissibility checking procedure grows as $|\mathcal{P}(\Gamma)| = 2^{|\Gamma|}$. Still, this procedure can be stopped as soon as some value contained \mathcal{R} is less than 1. Moreover, recall that checking for admissibility is not required online for the SP^2 strategy in [Algorithm 1](#). Instead, this procedure is used to construct admissible sets of schedules in [Algorithm 2](#) offline.

Algorithm 3 Admissibility

Input: $n, M_0, \{M_\gamma\}_{\gamma \in \Gamma}, \Gamma$.

Output: $(\mathcal{S}_0 \subset \text{int}(\mathcal{S}_\Gamma^*))?$

```

1: Set  $\mathcal{R} \leftarrow \emptyset$ .
2: for  $\Gamma' \in \mathcal{P}(\Gamma)$  do
3:   if  $|\Gamma'| = n$  then
4:      $\mathcal{R} \leftarrow \mathcal{R} \cup \{$ 
       IsolatedSolutions $(n, M_0, \{M_\gamma\}_{\gamma \in \Gamma}, \Gamma, \Gamma')\}$ 
5:   else
6:     if  $|\Gamma'| < n$  then
7:        $\mathcal{R} \leftarrow \mathcal{R} \cup \{$ 
         RegularSolutions $(n, M_0, \{M_\gamma\}_{\gamma \in \Gamma}, \Gamma, \Gamma')\}$ 
8:     end if
9:   end if
10: end for
11: return  $\min \mathcal{R} > 1$ 

```

One important input for [Algorithm 3](#) is the matrix M_0 which acts as an hyper-parameter for the overall system. Note that admissibility of the set of schedules Γ only depends on some arbitrary choice of M_0 as long as it is positive definite. Hence, setting $M_0 = I$ can be considered a systematic default option. On the other hand, the choice of M_0 may affect the time until admissibility is concluded in [Algorithm 3](#). However, optimizing M_0 to reduce such time is considered out of the scope of this work but could be explored in future research.

5. Towards optimal scheduling

Up to this point, the analysis has been devoted to the stability aspect of [Problem 3](#). In this section, we focus on the optimality aspect of the problem based on the stability results obtained so far. First, we study the inherent complexity of [Problem 3](#). Due to

Algorithm 4 IsolatedSolutions**Input:** $n, M_0, \{M_\gamma\}_{\gamma \in \Gamma}, \Gamma, \Gamma'$.**Output:** r .

- 1: $\bar{\mathbf{X}}_{\Gamma'} \leftarrow$ real solutions of $\{\bar{\mathbf{x}}^T M_\gamma \bar{\mathbf{x}} = 1, \forall \gamma \in \Gamma'\}$ as in Remark 11.
- 2: **for** $\bar{\mathbf{x}} \in \bar{\mathbf{X}}_{\Gamma'}$ **do**
- 3: **if** exists $\gamma \in \Gamma \setminus \Gamma'$ such that $\bar{\mathbf{x}}^T M_\gamma \bar{\mathbf{x}} < 1$ **then**
- 4: $\bar{\mathbf{X}}_{\Gamma'} \leftarrow \bar{\mathbf{X}}_{\Gamma'} \setminus \{\bar{\mathbf{x}}\}$.
- 5: **end if**
- 6: **end for**
- 7: **return** $r \leftarrow \min\{\bar{\mathbf{x}}^T M_0 \bar{\mathbf{x}} : \bar{\mathbf{x}} \in \bar{\mathbf{X}}_{\Gamma'}\}$

Algorithm 5 RegularSolutions**Input:** $n, M_0, \{M_\gamma\}_{\gamma \in \Gamma}, \Gamma, \Gamma'$.**Output:** r .

- 1: Set $Y \leftarrow$ real solutions of (9) as in Remark 14
- 2: $\mathcal{R} \leftarrow \emptyset$
- 3: **for** $\lambda \in Y$ **do**
- 4: $G(\lambda) \leftarrow M_0 + \sum_{\gamma \in \Gamma'} \lambda_\gamma M_\gamma$.
- 5: $\mathbf{g} \leftarrow$ any element of $\ker(G(\lambda))$.
- 6: $\bar{\mathbf{x}} \leftarrow \mathbf{g} \sqrt{\frac{-\sum_{\gamma \in \Gamma'} \lambda_\gamma}{\mathbf{g}^T M_0 \mathbf{g}}}$.
- 7: **if** $\bar{\mathbf{x}}^T M_\gamma \bar{\mathbf{x}} > 1, \forall \gamma \in \Gamma \setminus \Gamma'$ **then**
- 8: $\mathcal{R} \leftarrow \mathcal{R} \cup \{-\sum_{\gamma \in \Gamma'} \lambda_\gamma\}$
- 9: **end if**
- 10: **end for**
- 11: **return** $r \leftarrow \min \mathcal{R}$.

the possible conflicting objectives in (2), the most general setting of Problem 3 is reduced to a combinatorial search between the possible scheduling decisions. This makes it hard to expect that exact and efficient solutions for Problem 3 exist except for very particular cases. This issue is formalized in the following result.

Proposition 19. *Problem 3 is NP-hard.*

Henceforth, our approach from this point will be to obtain approximate solutions to Problem 3 and study their performance with respect to the optimal solution. Consider $m > 0$ sets of schedules $\Gamma^1, \dots, \Gamma^m$ constructed using Algorithm 2, leading to m scheduling policies. Theorem 8 ensures that it is possible to choose between any of these scheduling policies without compromising asymptotic stability. Hence, instead of deciding between different perception configurations, a sub-optimal schedule for Problem 3 will be constructed by an appropriate decision sequence between the m scheduling policies induced by $\Gamma^1, \dots, \Gamma^m$.

Using the dynamic programming framework (Bertsekas, 2000), Algorithm 6 aims to obtain an optimal schedule for Problem 3 when the schedule is constrained to be constructed using the scheduling policies induced by $\Gamma^1, \dots, \Gamma^m$. Note that this algorithm proceeds to evaluate all scheduling policies for the current state $\bar{\mathbf{x}}_0$ as in line 3. Then, proceeds to compute $\bar{\mathbf{x}}(t), P(t)$ in line 7 for $t \in [\tau, \min(\tau^+, T_f)]$ which is the interval in which the schedule piece γ is active before $t = T_f$, as well as the cost for such interval in line 8. Note that $\bar{\mathbf{x}}(t), P(t)$ and the cost can be computed explicitly as a result of Proposition 24 in Appendix A. Then, $\tau^+ < T_f$ it means that γ is not sufficiently long to fill the entire window $[0, T_f]$. Hence, the remaining cost to go J^+ and schedule p^+ optimal for $t \in [\tau^+, T_f]$ is obtained by a recursive call in line 10. Otherwise, the terminal cost is added in line 13. Once all m options have been evaluated, dynprog returns the one with

the best cost as well as the optimal schedule as in line 16, where \oplus means concatenation of schedules. The optimal properties of this algorithm are detailed in the following.

Algorithm 6 dynprog**Input:** $T_f, \tau, \bar{\mathbf{x}}_0, P_0$.**Output:** p^*, J^*, Γ^* .

- 1: $J^* \leftarrow \infty$.
- 2: $\Gamma^* \leftarrow \emptyset$
- 3: **for** Γ in $\{\Gamma^1, \dots, \Gamma^m\}$ **do**
- 4: $\gamma \leftarrow \gamma^*(\bar{\mathbf{x}}_0; \Gamma)$ from (5).
- 5: $\tau^+ \leftarrow \tau + \sum_{k=0}^{\text{len}(\gamma)-1} \Delta^{\gamma k}$
- 6: $\alpha \leftarrow \text{att}(\gamma, [\tau, \min(\tau^+, T_f)])$
- 7: Compute $\bar{\mathbf{x}}(t), P(t)$ for schedule γ using $\bar{\mathbf{x}}(\tau) = \bar{\mathbf{x}}_0, P(\tau) = P_0$ on (A.5) for $t \in [\tau, \min(\tau^+, T_f)]$
- 8: $J \leftarrow \frac{\lambda_{\mathbf{x}}}{T_f} \int_{\tau}^{\min(\tau^+, T_f)} \bar{\mathbf{x}}(t)^T Q \bar{\mathbf{x}}(t) + \text{tr}(QP(t)) dt + \frac{\lambda_r}{T_f} \sum_{k=0}^{\alpha-1} r^{\gamma k}$
- 9: **if** $(\tau^+ < T_f)$ **then**
- 10: $\{p^+, J^+, \Gamma^+\} \leftarrow$
 dynprog($T_f, \tau^+, \bar{\mathbf{x}}(\tau^+), P(\tau^+)$)
- 11: $J \leftarrow J + J^+$
- 12: **else**
- 13: $J \leftarrow J + \lambda_{\mathbf{x}} (\bar{\mathbf{x}}(T_f)^T Q_f \bar{\mathbf{x}}(T_f) + \text{tr}(QP(T_f)))$
- 14: $p^+ \leftarrow \emptyset$
- 15: **end if**
- 16: **if** $J \leq J^*$ **then**
- 17: $J^* \leftarrow J$
- 18: $p^* \leftarrow \gamma \oplus p^+$
- 19: $\Gamma^* \leftarrow \Gamma$
- 20: **end if**
- 21: **end for**

Proposition 20. *Calling dynprog($T_f, 0, \bar{\mathbf{x}}_0, P_0$) obtains the optimal schedule and cost p^*, J^* for Problem 3 for the case when the schedule is constrained to be constructed using the scheduling policies for $\{\Gamma^1, \dots, \Gamma^m\}$. Moreover, the worst case complexity of Algorithm 6 is $O(m^{\lfloor T_f/c \rfloor})$ where $c = \min\{\sum_{k=0}^{\text{len}(\gamma)-1} \Delta^{\gamma k} : \gamma \in \Gamma^1 \cup \dots \cup \Gamma^m\}$.*

Remark 21. Due to NP-hardness of Problem 3, it is not surprising that the dynamic programming approach for this problem leads to an algorithm that does not have a polynomial complexity in T_f . However, these types of algorithms, in which unrolling the recursive calls lead to an exponentially growing tree as in Algorithm 6, are widely studied. Hence, some performance improvements in terms of complexity and run-time reduction can be made to Algorithm 6 such as applying a branch-and-bound technique (Bertsekas, 2000, Chapter 2.3.3). In addition, heuristics and approximate solutions such as limited look-ahead policies, roll-out algorithms, among others (Bertsekas, 2000, Chapter 6) can be applied in practice. For instance, a balanced SP² strategy can be applied where the set of schedules Γ in line 8 of Algorithm 1 is selected by calling $\{-, -, \Gamma\} = \text{dynprog}(T, 0, \hat{\mathbf{x}}[k|k-1], \hat{P}[k])$ as a moving horizon strategy with a look-ahead window of size T .

Nevertheless, all heuristics and approximate approaches mentioned before will have a relative performance loss with respect to the optimal schedule for Problem 3 only when the solution is constrained to be constructed using the scheduling policies induced by $\Gamma^1, \dots, \Gamma^m$, and not necessarily the real optimum of Problem 3. Despite this, in the following result, we show that the solution of Algorithm 6 can approximate the optimal performance for sufficiently large number of sets of schedules m , increasing the computational power applied.

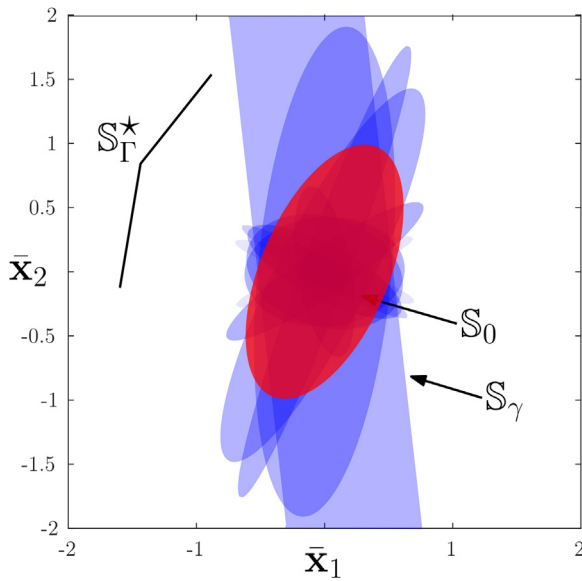


Fig. 1. Set \mathbb{S}_Γ^* for the set of schedules Γ described in Section 6.1 as well as \mathbb{S}_0 to show admissibility.

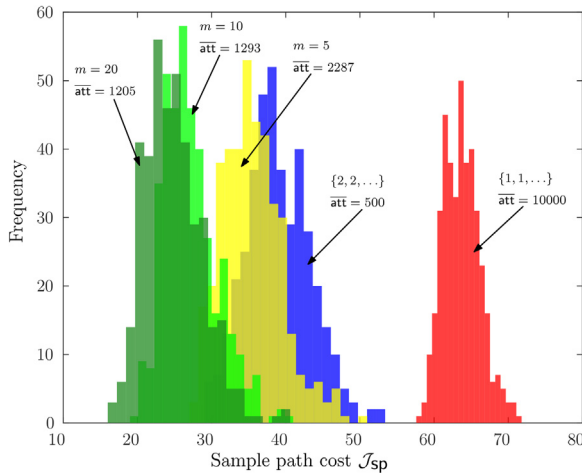


Fig. 2. Resulting histograms for the sample path cost \mathcal{J}_{sp} for example of Section 6.1 using the balanced SP^2 strategy with $m = 5, m = 10, m = 20$ and static schedules $\{1, 1, \dots\}, \{2, 2, \dots\}$.

Theorem 22. Let p^* be the optimal cost for the general setting of Problem 3 and assume that (3) is stabilizable. Then, for any compact sets $B_{\bar{x}} \subset \mathbb{R}^n, B_p \subset \mathbb{R}^{n \times n}$ and any $\varepsilon > 0$ there exists $\Gamma^1, \dots, \Gamma^m$ with $m < \infty$ such that

$$|\mathcal{J}(p^*; \bar{\mathbf{x}}_0, P_0) - \mathcal{J}(p; \bar{\mathbf{x}}_0, P_0)| < \varepsilon$$

with p from by Algorithm 6, for any $\bar{\mathbf{x}}_0 \in B_{\bar{x}}, P_0 \in B_p$.

6. Simulation examples

6.1. Double integrator

As a first example consider system (1) with

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, C = [1, 0], W_0 = 1$$

as well as $D = 2$ with perception latencies $\Delta^1 = 0.01, \Delta^2 = 0.1$ with their corresponding covariances $\Sigma^1 = 0.5, \Sigma^2 = 0.01$. In

addition, consider

$$L^1 = L^2 = [-1.5 \ -3], M_0 = \begin{bmatrix} 3.53 & -1.10 \\ -1.10 & 1.36 \end{bmatrix}$$

where M_0 was chosen as an arbitrarily positive definite matrix. To depict admissibility, Algorithm 2 was used with $\ell = 20$ to generate a set of schedules Γ and Algorithm 3 was used to check admissibility. Note that in Algorithm 3 the only meaningful subsets of schedules Γ' in the power set $\mathcal{P}(\Gamma)$ are Γ' containing either a single schedule, or pairs of schedules as a result of the discussion in Remark 11. Moreover, note that in the case of Γ' containing a single schedule γ , (10) results in $G(\lambda) = M_0 + \lambda M_\gamma$ and therefore possible solutions of Lagrange multipliers $\lambda \in \mathbb{R}$ are eigenvalues of $M_0^{-1}M_\gamma$. The resulting nullity of $G(\lambda)$ for any Γ' in this experiment was 1 and hence regular critical points were computed as in (12). In this case, there are no non-regular points, since for $|\Gamma'| = 1$ there is a single vector $\nabla h_\gamma(\bar{\mathbf{x}})$. For the case of $|\Gamma'| = n = 2$ critical points were obtained by solving a polynomial system of equations given by the intersection of the two resulting ellipses as in Remark 11. Hence, in this setting the admissibility value was obtained to be $R = 1.142$ which indicates that Γ is admissible. Fig. 1 shows \mathbb{S}_Γ^* and \mathbb{S}_0 for this example where it shown that $\mathbb{S}_0 \subset \text{int}(\mathbb{S}_\Gamma^*)$. Moreover, note that none of the individual sets $\mathbb{S}_\gamma, \gamma \in \Gamma$ manage to cover \mathbb{S}_0 by separate.

Now, we show the performance of an heuristics implementation of the strategy presented in Section 5. To do so, we simulated system (1) using explicit Euler–Maruyama method with time step $h = 10^{-5}$ over the interval $[0, T_f = 100]$ for each experiment. A balanced SP^2 strategy as described in Remark 21 was used with limited look-ahead window of size $T = 2$. The performance is evaluated using the cost (2) for each sample path with $Q = Q_f = \text{diag}([2, 1]), \lambda_x = 1, \lambda_r = 0.05$. Moreover, $r^1 = r^2 = 1$ such that the number of sampling events is penalized in the cost. Furthermore, 400 sample paths were generated with $\bar{\mathbf{x}}_0 = [1, 1], P_0 = I$ for $m = 5, m = 10, m = 20$ respectively. For comparison, the same amount of sample paths were obtained for schedules $\{1, 1, \dots\}, \{2, 2, \dots\}$ respectively to study the performance of the classical approach of maintaining the same perception configuration during the whole experiment. The resulting histograms for the sample path cost \mathcal{J}_{sp} for the whole interval $[0, T_f]$ are shown in Fig. 2 for each case. It can be observed that the average cost of our approach is reduced when compared to the static approaches. Moreover, the results suggest that increasing the number of sets of schedules m improves the cost \mathcal{J}_{sp} and average attention $\bar{\text{att}}$.

6.2. Particle mobile robot

Now consider a system with state $\mathbf{x} = [\mathbf{x}_1, \dots, \mathbf{x}_6]^T$, and dynamics given $\dot{\mathbf{x}}_1 = \mathbf{x}_2, \dot{\mathbf{x}}_3 = \mathbf{x}_4, \dot{\mathbf{x}}_5 = \mathbf{x}_6$ as well as $\dot{\mathbf{x}}_2 = \mathbf{u}_1/\mu + \mathbf{w}_1, \dot{\mathbf{x}}_4 = \mathbf{u}_2/\mu + \mathbf{w}_2, \dot{\mathbf{x}}_6 = \mathbf{u}_3/\mu + \mathbf{w}_3$ with $\mu = 0.1$. This system corresponds to the model of a particle mobile robot with mass μ . This kind of system has been useful as a model for aerial vehicles such as multi-rotors in Campos-Macías et al. (2021), Faessler, Franchi, and Scaramuzza (2018) and Mellinger and Kumar (2011) were $[\mathbf{x}_1, \mathbf{x}_3, \mathbf{x}_5]^T$ is the position of the multi-rotor and $[\mathbf{x}_2, \mathbf{x}_4, \mathbf{x}_6]^T$ is the velocity. In fact, a common strategy for the multi-rotor control is adopt a hierarchical control approach, where control signals $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3$ are designed exclusively for position as a high-level controller. In a second stage, the resulting position controls are used to construct a low level controller for the attitude of the multi-rotor. Moreover, measurements for position are taken as $C\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_3, \mathbf{x}_5]^T$ using visual perception (Campos-Macías et al., 2021). In this example we adopt the noise model from Ballotta et al. (2020, Equation (9)) such that

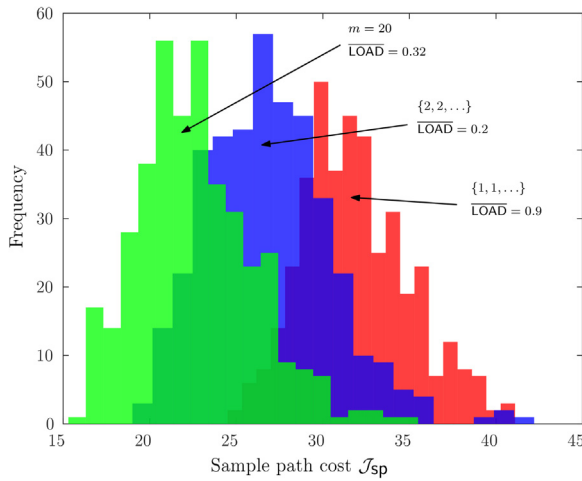


Fig. 3. Resulting histograms for the sample path cost \mathcal{J}_{sp} for the example of Section 6.1 using the balanced SP² strategy with $m = 20$ and static schedules $\{1, 1, \dots\}$, $\{2, 2, \dots\}$.

given a perception latency Δ the resulting measurement covariance is $\Sigma(\Delta) = (b/\Delta)I$ with $b = 0.2$. This model is reasonable for demonstrative purposes in this work. However, in a more practical setting, $\Sigma(\Delta)$ must be characterized according to the relation between latency and precision of the actual perception algorithms and sensors used, following a similar procedure to the one described in Pant et al. (2021). In this case, even when the measurement covariance is reduced as Δ is increased, the disturbance covariance $W_d(\Delta)$, given in (A.4) from Appendix A, increases.

Now, we test our scheduling approach assuming only two different latencies $\Delta^1 = 1/30$, $\Delta^2 = 4/30$ corresponding to the frame-rate of a typical camera or 4 times the frame-rate. Let $\Sigma^1 = \Sigma(\Delta^1)$, $\Sigma^2 = \Sigma(\Delta^2)$ with the covariance model $\Sigma(\Delta)$ described before and $W_0 = 0.5I$. Moreover, consider penalties $r^1 = 0.9\Delta^1$, $r^2 = 0.2\Delta^2$ which correspond to CPU loads of 90% and 20% respectively for each method. In addition, the parameters $Q = Q_f = \text{diag}([2, 1, 2, 1, 2, 1])$, $\lambda_x = 1$, $\lambda_r = 0.1$ were chosen. The SP² strategy was tested similarly as in the example of Section 6.1 with time step $h = 10^{-5}$, $T_f = 100$ and limited look-ahead window length $T = 10$. Fig. 3 shows the resulting histograms for sample paths with $\bar{\mathbf{x}}_0 = [1, 1, 1, 1, 1, 1]$, $P_0 = I$ for our approach with $m = 20$ and static schedules $\{1, 1, \dots\}$, $\{2, 2, \dots\}$ with 400 sample paths in each case. It is observed that our approach outperforms the static approaches in the sample cost \mathcal{J}_{sp} . Moreover, the average CPU load $\overline{\text{LOAD}}$ is reduced with respect to the worst case value of 90%.

7. Conclusion

This work studied a perception scheduling approach to tackle the latency-precision trade-off, where measurements can be taken using several perception mode possibilities with different noise levels. It was shown that the stability of the closed-loop model is tied to the stability of a switching system. Moreover, the latency-precision trade-off was modeled by means of a cost function taking into account control precision and a penalty for each perception mode to incorporate energetic and CPU load points of view. Three main tools were proposed in this context to solve the problem: scheduling policy candidate construction using switching systems theory, an admissibility checking algorithm for policy candidates, and sub-optimal strategies based upon the scheduling policy candidates. Different from general switching

systems theory, we take advantage of the particular structure of the problem to provide new insights on the construction of multiple scheduling policies for optimization purposes and on its theoretical complexity. Finally, some illustrative examples were provided, including one application for a mobile robot. These results motivate future research on this problem with other noise models for the perception method or imperfect timing in the perception latency.

Acknowledgments

This work was supported via projects PID2021-124137OB-I00 and TED2021-130224B-I00 funded by MCIN/AEI/10.13039/501100011033, by ERDF A way of making Europe and by the European Union NextGenerationEU/PRTR, by the Gobierno de Aragón under Project DGA T45_23R, by the Universidad de Zaragoza and Banco Santander, by the Consejo Nacional de Ciencia y Tecnología (CONACYT-Mexico) with grant number 739841.

Appendix A. Distribution, estimation and cost of $\mathbf{x}(t)$

In this section, we obtain expressions for $\hat{\mathbf{x}}[k|k-1] := \mathbb{E}\{\mathbf{x}[k]|\mathbf{z}[0], \dots, \mathbf{z}[k-1]\}$, the distribution of $\mathbf{x}(t)$ as the closed-loop solution of (1) and an expression for (2).

Proposition 23. Consider Assumptions 1 and 2. Then, given measurements $\mathbf{z}[0], \dots, \mathbf{z}[k-1]$ and a perception schedule p , the value of $\hat{\mathbf{x}}[k|k-1] := \mathbb{E}\{\mathbf{x}[k] | \mathbf{z}[0], \dots, \mathbf{z}[k-1]\}$ is computed according to:

$$\begin{aligned} H[k] &= A_d(\Delta^{p_k})\hat{P}[k]C^T \left(C\hat{P}[k]C^T + \Sigma^{p_k} \right)^{-1} \\ \hat{\mathbf{x}}[k+1|k] &= A_d(\Delta^{p_k})\hat{\mathbf{x}}[k|k-1] + B_d(\Delta^{p_k})\mathbf{u}[k] \\ &\quad + H[k](\mathbf{z}[k] - C\hat{\mathbf{x}}[k|k-1]) \\ \hat{P}[k+1] &= (A_d(\Delta^{p_k}) - H[k]C)\hat{P}[k]A_d(\Delta^{p_k})^T \\ &\quad + W_d(\Delta^{p_k}) \end{aligned} \tag{A.1}$$

with $\hat{\mathbf{x}}[0|-1] = \bar{\mathbf{x}}_0$, $\hat{P}[0] = P_0$, $A_d(\tau) := \exp(A\tau)$, $B_d(\tau) := \int_0^\tau A_d(s)dsB$, $W_d(\tau) := \int_0^\tau A_d(s)W_0A_d(s)^T ds$. Moreover, $\text{cov}\{\hat{\mathbf{x}}[k|k-1] - \mathbf{x}[k]\} = \hat{P}[k]$.

Proof. First, note that the explicit solution to (1) in the interval $t - \tau_k \in [0, \Delta^{p_k}]$ is given by

$$\mathbf{x}(t) = A_d(t - \tau_k)\mathbf{x}[k] + B_d(t - \tau_k)\mathbf{u}[k] + \mathbf{w}_d(t) \tag{A.2}$$

with $\mathbf{w}_d(t) \sim \mathcal{N}(0, W_d(t - \tau_k))$ (Soderstrom, 2002, Section 4.5.2). Note that evaluating (A.2) at $t = \tau_{k+1}$ leads to the discrete-time system which transitions from $\mathbf{x}[k]$ to $\mathbf{x}[k+1]$. Thus, the result follows by applying the estimator in Åström (1970, Theorem 4.1, Page 228).

Proposition 24. Let p a perception schedule and $\Lambda(t - \tau_k) := A_d(t - \tau_k) + B_d(t - \tau_k)L^{p_k}$, $t - \tau_k \in [0, \Delta^{p_k}]$. Moreover, consider Assumptions 1 and 2. Then, the solution $\mathbf{x}(t)$ of the SDE in (1) under the controller $\mathbf{u}[k] = L^{p_k}\hat{\mathbf{x}}[k|k-1]$ given samples $\{\mathbf{z}[0], \dots, \mathbf{z}[k-1]\}$ is

$$\mathbf{x}(t) = \Lambda(t - \tau_k)\mathbf{x}[k] + \mathbf{w}_\Lambda(t) \tag{A.3}$$

for $t - \tau_k \in [0, \Delta^{p_k}]$ and $\mathbf{w}_\Lambda(t) \sim \mathcal{N}(0, W_\Lambda(t - \tau_k))$ with:

$$\begin{aligned} W_\Lambda(t - \tau_k) &= B_d(t - \tau_k)L^{p_k}\hat{P}[k]L^{p_k}B_d(t - \tau_k)^T \\ &\quad + W_d(t - \tau_k) \end{aligned} \tag{A.4}$$

Moreover, $\mathbf{x}(t) \sim \mathcal{N}(\bar{\mathbf{x}}(t), P(t))$ for $t - \tau_k \in [0, \Delta^{p_k}]$ with

$$\begin{aligned} \bar{\mathbf{x}}(t) &= \Lambda(t - \tau_k)\mathbf{x}[k], \quad \bar{\mathbf{x}}[0] = \bar{\mathbf{x}}_0 \\ P(t) &= \Lambda(t - \tau_k)P[k]\Lambda(t - \tau_k)^T + W_\Lambda(t - \tau_k) \end{aligned} \tag{A.5}$$

with $P[0] = P_0$. In addition, with $\alpha := \text{att}(p; [0, T_f])$, cost (2) is

$$\begin{aligned} \mathcal{J}(p; \bar{\mathbf{x}}_0, P_0) &= \frac{\lambda_{\mathbf{x}}}{T_f} \int_0^{T_f} \bar{\mathbf{x}}(t)^T Q \bar{\mathbf{x}}(t) + \text{tr}(QP(t)) dt \\ &+ \lambda_{\mathbf{x}} (\bar{\mathbf{x}}(T_f)^T Q_f \bar{\mathbf{x}}(T_f) + \text{tr}(Q_f P(T_f))) + \frac{\lambda_r}{T_f} \sum_{k=0}^{\alpha-1} r^{pk} \end{aligned} \quad (\text{A.6})$$

Proof. Proposition 23 implies that $\hat{\mathbf{x}}[k|k-1] = \mathbf{x}[k] + \tilde{\mathbf{x}}[k]$ where $\tilde{\mathbf{x}}[k] \sim \mathcal{N}(0, \hat{P}[k])$ given samples $\{\mathbf{z}[0], \dots, \mathbf{z}[k-1]\}$. Hence, using $\mathbf{u}[k] = L^{pk}\mathbf{x}[k] + L^{pk}\tilde{\mathbf{x}}[k]$ in (A.2) leads directly to (A.3) with $\mathbf{w}_A = B_d(t - \tau_k)L^{pk}\tilde{\mathbf{x}}[k] + \mathbf{w}_d(t)$. Moreover, according to Soderstrom (2002, Section 4.5.2) the process $\{\mathbf{w}_d[0], \dots, \mathbf{w}_d[k], \mathbf{w}_d(t)\}$ is a white noise process and thus, its individual random variables are uncorrelated. Therefore, $\tilde{\mathbf{x}}[k]$ and $\mathbf{w}_d(t)$ are uncorrelated. Henceforth, computing the covariance of $\mathbf{w}_A(t)$ leads to (A.4). Furthermore, direct computation of the expectation and covariance over (A.3) leads to (A.5). Finally, (A.6) is computed as in Aldana-López, Aragüés, and Sagüés (2020, Theorem 1).

Appendix B. C* Sets and gauge functions

Definition 25 (Fiacchini & Jungers, 2014, Definition 1). A C* set $\mathbb{S} \subset \mathbb{R}^n$ is a compact, star-convex set with the origin as a center i.e., for any $\mathbf{x} \in \mathbb{S}$ the whole line $\{\lambda \mathbf{x} \in \mathbb{R}^n : \lambda \in [0, 1]\}$ is contained in \mathbb{S} and $0 \in \text{int}(\mathbb{S})$. Moreover, the gauge function of a C* set \mathbb{S} is defined as

$$\Psi(\mathbf{x}; \mathbb{S}) := \inf\{\alpha \in \mathbb{R} : \alpha \geq 0, \mathbf{x} \in \alpha\mathbb{S}\} \quad (\text{B.1})$$

This is, $\Psi(\mathbf{x}; \mathbb{S})$ is the smallest scale α such that \mathbf{x} is still contained in $\alpha\mathbb{S}$. In the following, we enumerate some properties regarding C* sets and their gauge functions:

Lemma 26. Let \mathbb{S} be a C* set and $\Psi(\bullet; \mathbb{S}) : \mathbb{R}^n \rightarrow [0, +\infty)$ be its gauge function. Then, the following statements are true:

- (1) Let $0 < \alpha < 1$ and $\mathbf{x} \in \mathbb{S}$, then $\alpha\mathbf{x} \in \mathbb{S}$.
- (2) Let $0 < \alpha < \beta$, then $\alpha\mathbb{S} \subset \beta\mathbb{S}$.
- (3) Let $\alpha := \Psi(\mathbf{x}; \mathbb{S})$, then $\mathbf{x} \in \partial(\alpha\mathbb{S})$.
- (4) $\Psi(\bullet; \mathbb{S})$ is homogeneous of degree one, i.e. $\Psi(\beta\mathbf{x}; \mathbb{S}) = \beta\Psi(\mathbf{x}; \mathbb{S})$, $\forall \beta \geq 0$ and $\forall \mathbf{x} \in \mathbb{R}^n$.
- (5) Let \mathbb{S}' be a C* set with $\mathbb{S}' \subset \mathbb{S}$. Thus $\Psi(\mathbf{x}; \mathbb{S}') > \Psi(\mathbf{x}; \mathbb{S})$, $\forall \mathbf{x} \in \mathbb{R}^n$.
- (6) Let $\mathbb{S}_0 = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{x}^T M_0 \mathbf{x} \leq 1\}$ for some positive definite matrix M_0 . Then, $\Psi(\mathbf{x}; \mathbb{S}_0) = \sqrt{\mathbf{x}^T M_0 \mathbf{x}}$ for any $\mathbf{x} \in \mathbb{R}^n$.

Proof. These properties can be verified equivalently as the properties of gauge functions of standard convex sets (Rockafellar, 1970, Page 28).

Appendix C. Proofs

C.1. Proof of Theorem 8

In order to show Theorem 8 we first provide 2 auxiliary technical lemmas:

Lemma 27. Let Γ be an admissible set of schedules and let $\gamma^* := \gamma^*(\bar{\mathbf{x}}_0; \Gamma)$ be a schedule obtained from (5) for the initial condition $\bar{\mathbf{x}}_0 \in \mathbb{S}_\gamma^*$. Thus, if the SP² strategy is used in (3) then $\bar{\mathbf{x}}[\text{len}(\gamma^*)] \in \mathbb{S}_0 \subseteq \text{int}(\mathbb{S}_\gamma^*)$.

Proof. Since $\bar{\mathbf{x}}_0 \in \mathbb{S}_\gamma^*$ then there exists a non empty set $\Gamma' \subseteq \Gamma$ such that $\bar{\mathbf{x}}_0 \in \mathbb{S}_\gamma \forall \gamma \in \Gamma'$. Moreover, $\bar{\mathbf{x}}_0^T M_\gamma \bar{\mathbf{x}}_0 \leq 1$ for any $\gamma \in \Gamma'$ and $\bar{\mathbf{x}}_0^T M_\gamma \bar{\mathbf{x}}_0 > 1$ for $\gamma \in \Gamma \setminus \Gamma'$. Hence, $\gamma^* = \gamma^*(\bar{\mathbf{x}}_0; \Gamma) \in \Gamma'$ and $\bar{\mathbf{x}}_0 \in \mathbb{S}_{\gamma^*}$. Therefore, by construction of the set \mathbb{S}_{γ^*} , system (3) after $\text{len}(\gamma^*)$ steps will result in $\bar{\mathbf{x}}[\text{len}(\gamma^*)] \in \mathbb{S}_0 \subset \mathbb{S}_\gamma^*$.

Lemma 28. Let Γ be an admissible set of schedules, $\Psi(\bullet; \mathbb{S}_\gamma^*)$ be its corresponding gauge function and let $\gamma^* := \gamma^*(\bar{\mathbf{x}}_0; \Gamma)$ be a schedule obtained from (5) for the initial condition $\bar{\mathbf{x}}_0 \neq 0$. Thus, if the SP² strategy is used in (3) then

- (1) $\Psi(\bar{\mathbf{x}}[\text{len}(\gamma^*)]; \mathbb{S}_\gamma^*) < \Psi(\bar{\mathbf{x}}_0; \mathbb{S}_\gamma^*)$
- (2) $\Psi(\bar{\mathbf{x}}[\text{len}(\gamma^*)]; \mathbb{S}_0) < \Psi(\bar{\mathbf{x}}_0; \mathbb{S}_0)$

Proof. Note that by Lemma 26-(3) in Appendix B it is obtained that $\bar{\mathbf{x}}_0 \in \partial(\alpha_0 \mathbb{S}_\gamma^*)$ with $\alpha_0 := \Psi(\bar{\mathbf{x}}_0; \mathbb{S}_\gamma^*)$. Similarly, one can obtain that $\bar{\mathbf{x}}[\text{len}(\gamma^*)] \in \partial(\alpha \mathbb{S}_\gamma^*)$ with $\alpha := \Psi(\bar{\mathbf{x}}[\text{len}(\gamma^*)]; \mathbb{S}_\gamma^*)$. Moreover, since $\bar{\mathbf{x}}_0/\alpha_0 \in \partial\mathbb{S}_\gamma^*$, then Lemma 27 implies that $\bar{\mathbf{x}}[\text{len}(\gamma^*)]/\alpha \in \mathbb{S}_0 \subseteq \text{int}(\mathbb{S}_\gamma^*)$. Hence, $\partial(\alpha \mathbb{S}_\gamma^*) \subset \text{int}(\alpha_0 \mathbb{S}_\gamma^*)$ strictly, unless $\alpha_0 = 0$ which is not possible since $\bar{\mathbf{x}}_0 \neq 0$. However, for that to be true, we require $\alpha < \alpha_0$ which is exactly item (1). For item (2), since Γ is admissible then $\mathbb{S}_0 \subset \mathbb{S}_\gamma^*$ and therefore $1 = \Psi(\bar{\mathbf{x}}_0/\alpha_0; \mathbb{S}_\gamma^*) < \Psi(\bar{\mathbf{x}}_0/\alpha_0; \mathbb{S}_0)$ by Lemma 26-(5). Equivalently $\alpha_0 < \Psi(\bar{\mathbf{x}}_0; \mathbb{S}_0)$. Recall that $\bar{\mathbf{x}}[\text{len}(\gamma^*)]/\alpha \in \mathbb{S}_0$ and therefore $\Psi(\bar{\mathbf{x}}[\text{len}(\gamma^*)]/\alpha; \mathbb{S}_0) \leq 1$ or equivalently $\Psi(\bar{\mathbf{x}}[\text{len}(\gamma^*)]; \mathbb{S}_0) \leq \alpha_0 < \Psi(\bar{\mathbf{x}}_0; \mathbb{S}_0)$.

Now, we are in position to prove Theorem 8. Note that as a consequence of the previous lemma, both $\Psi(\bullet; \mathbb{S}_\gamma^*)$ and $\Psi(\bullet; \mathbb{S}_0)$ are Lyapunov function candidates for (3). Note that $\Psi(\bullet; \mathbb{S}_\gamma^*)$ can be used for a similar setting. However, we use $\Psi(\bullet; \mathbb{S}_0)$ instead to take advantage of the fact that this Lyapunov function is independent of the set of schedules used. Henceforth, switching between admissible sets of schedules as in line 8 of Algorithm 1 is possible. Let $\{\ell_\kappa\}_{\kappa=0}^\infty$ be the sequence of switching time instants obtained by using the strategy SP² on (3). Moreover, define $\mathbf{y}[\kappa] := \bar{\mathbf{x}}[\ell_\kappa]$ as the state at instants where the schedule changes. Consider the Lyapunov function candidate $V(\mathbf{y}[\kappa]) := \Psi(\mathbf{y}[\kappa]; \mathbb{S}_0)$. Lemma 28 implies $V(\mathbf{y}[\kappa+1]) - V(\mathbf{y}[\kappa]) < 0$ which in turn implies $\lim_{\kappa \rightarrow \infty} \mathbf{y}[\kappa] = \lim_{\kappa \rightarrow \infty} \bar{\mathbf{x}}[\ell_\kappa] = 0$. For any $\bar{\mathbf{x}}[k]$ with $k \notin \{\ell_\kappa\}_{\kappa=0}^\infty$, i.e. values of $\bar{\mathbf{x}}$ not at schedule change instants, it can be obtained from (3) that $\bar{\mathbf{x}}[k] = \Lambda(\Delta^{p_{k-1}})\Lambda(\Delta^{p_{k-2}})\dots\Lambda(\Delta^{p_{\ell'}})\bar{\mathbf{x}}[\ell']$ with $\ell' = \max\{\ell \in \{\ell_\kappa\}_{\kappa=0}^\infty : \ell < k\}$. Hence $\|\bar{\mathbf{x}}[k]\| \leq \rho_k \|\bar{\mathbf{x}}[\ell']\|$ with ρ_k the spectral radius of $\Lambda(\Delta^{p_{k-1}})\Lambda(\Delta^{p_{k-2}})\dots\Lambda(\Delta^{p_{\ell'}}$, which since $\lim_{\ell' \rightarrow \infty} \bar{\mathbf{x}}[\ell'] = \lim_{\kappa \rightarrow \infty} \mathbf{y}[\kappa] = 0$ implies that $\lim_{k \rightarrow \infty} \bar{\mathbf{x}}[k] = 0$ too.

C.2. Proof of Theorem 9

First, note that due to Lemma 26-(6) in Appendix B, $R > 1$ if and only if $\Psi(\bar{\mathbf{x}}; \mathbb{S}_0) > 1, \forall \bar{\mathbf{x}} \in \partial\mathbb{S}_\gamma^*$, where $\Psi(\bullet; *)$ is the gauge function given in Definition 25 from Appendix B. Now, we show that if $R > 1$ then $\mathbb{S}_0 \subset \mathbb{S}_\gamma^*$. We proceed by contradiction: let us assume that there exists $\bar{\mathbf{x}} \in \mathbb{S}_0$ such that $\Psi(\bar{\mathbf{x}}; \mathbb{S}_\gamma^*) > 1$. Let $\beta := \Psi(\bar{\mathbf{x}}; \mathbb{S}_\gamma^*)$ and thus $\bar{\mathbf{x}}/\beta \in \partial\mathbb{S}_\gamma^*$ due to Lemma 26-(3). Henceforth, since $R > 1$, then $\Psi(\bar{\mathbf{x}}/\beta; \mathbb{S}_0) > 1$ or equivalently $\Psi(\bar{\mathbf{x}}; \mathbb{S}_0) > \beta > 1$ due to Lemma 26-(4) and due to the assumption $\beta = \Psi(\bar{\mathbf{x}}; \mathbb{S}_\gamma^*) > 1$. However, for $\bar{\mathbf{x}}$ to belong in \mathbb{S}_0 we require $\Psi(\bar{\mathbf{x}}; \mathbb{S}_0) < 1$ which leads to a contradiction. Therefore, $\beta = \Psi(\bar{\mathbf{x}}; \mathbb{S}_\gamma^*) \leq 1$ for all $\bar{\mathbf{x}} \in \mathbb{S}_0$. Hence, $\bar{\mathbf{x}} \in \partial(\beta\mathbb{S}_\gamma^*) \subseteq \beta\mathbb{S}_\gamma^* \subseteq \mathbb{S}_\gamma^*$ due to Lemma 26-(3) and Lemma 26-(1) for any $\bar{\mathbf{x}} \in \mathbb{S}_0$ and thus $\mathbb{S}_0 \subseteq \mathbb{S}_\gamma^*$. Finally, note that since all points $\bar{\mathbf{x}} \in \partial\mathbb{S}_\gamma^*$ have $\bar{\mathbf{x}}^T M_0 \bar{\mathbf{x}} > 1$ (by the assumption $R > 1$), then they do not belong to \mathbb{S}_0 . Thus, $\mathbb{S}_0 \cap \partial\mathbb{S}_\gamma^* = \emptyset$ and then $\mathbb{S}_0 \subset \text{int}(\mathbb{S}_\gamma^*)$. Now, we show that if $\mathbb{S}_0 \subset \text{int}(\mathbb{S}_\gamma^*)$, then $R > 1$. From here, note that $\mathbb{S}_0 \subseteq \text{int}(\mathbb{S}_\gamma^*)$. We proceed by contradiction: assume that there exists $\bar{\mathbf{x}} \in \partial\mathbb{S}_\gamma^*$ such that $\bar{\mathbf{x}}^T M_0 \bar{\mathbf{x}} \leq 1$. Hence, $\bar{\mathbf{x}} \in \mathbb{S}_0 \subseteq \text{int}(\mathbb{S}_\gamma^*)$ which is a contradiction. Therefore, $R > 1$ for all $\bar{\mathbf{x}} \in \partial\mathbb{S}_\gamma^*$ which concludes the proof.

C.3. Proof of Corollary 10

Let $\bar{\mathbf{x}}^*$ be the critical point of (6) which leads to the global minimum R and note that $\bar{\mathbf{x}}^* \in \partial S_{\Gamma}^*$. Hence, $\bar{\mathbf{x}}^* \in \partial S_{\gamma}$ for at least one $\gamma \in \Gamma$ and at most all $\gamma \in \Gamma$. Let Γ' be the set of all those $\gamma \in \Gamma$ and note that $\Gamma' \in \mathcal{P}(\Gamma)$. Therefore, $(\bar{\mathbf{x}}^*)^T M_{\gamma}(\bar{\mathbf{x}}^*) = 1$ for all $\gamma \in \Gamma'$ and is a local minimum of (7) for such Γ' . Since $\bar{\mathbf{x}}^* \in \partial S_{\Gamma}^*$ and $\bar{\mathbf{x}}^* \in \partial S_{\gamma}$ only for $\gamma \in \Gamma'$, thus $\bar{\mathbf{x}}^* \notin S_{\gamma}$ for $\gamma \in \Gamma \setminus \Gamma'$, and thus $\bar{\mathbf{x}}^* \in \bar{\mathbf{X}}_{\Gamma'}$. Now, for $\bar{\mathbf{x}}^*$ to be the global minimum, critical points $\bar{\mathbf{x}}$ comply $(\bar{\mathbf{x}}^*)^T M_0(\bar{\mathbf{x}}^*) \leq \bar{\mathbf{x}}^T M_0 \bar{\mathbf{x}}$ in (8).

C.4. Proof of Theorem 13

In order to show Theorem 13 we proceed by duality analysis, which is summarized in the following two technical lemmas.

Lemma 29. Any $\bar{\mathbf{x}}^*$ which is a regular critical point of program (7) comply with

$$\begin{aligned} G(\lambda)\bar{\mathbf{x}}^* &= 0 \\ (\bar{\mathbf{x}}^*)^T M_{\gamma}(\bar{\mathbf{x}}^*) &= 1 \quad \forall \gamma \in \Gamma' \end{aligned} \tag{C.1}$$

for some unique $\lambda = (\lambda_{\gamma})_{\gamma \in \Gamma'}$.

Proof. Let $\mathcal{L}(\bar{\mathbf{x}}, \lambda) = \bar{\mathbf{x}}^T M_0 \bar{\mathbf{x}} + \sum_{\gamma \in \Gamma'} \lambda_{\gamma} (\bar{\mathbf{x}}^T M_{\gamma} \bar{\mathbf{x}} - 1)$ be the Lagrangian for (7) with Lagrange multipliers λ . Moreover, $\mathcal{L}(\bar{\mathbf{x}}, \lambda)$ can be written as

$$\mathcal{L}(\bar{\mathbf{x}}, \lambda) = \bar{\mathbf{x}}^T G(\lambda) \bar{\mathbf{x}} - \sum_{\gamma \in \Gamma'} \lambda_{\gamma} \tag{C.2}$$

According to Bertsekas (1999, Proposition 3.1.1), all regular critical points $\bar{\mathbf{x}}^*$ must satisfy $(\bar{\mathbf{x}}^*)^T M_{\gamma} \bar{\mathbf{x}}^* = 1, \forall \gamma \in \Gamma'$ and $\nabla_{\bar{\mathbf{x}}} \mathcal{L}(\bar{\mathbf{x}}^*, \lambda) = 0$ or equivalently $G(\lambda)\bar{\mathbf{x}}^* = 0$ for some unique Lagrange multipliers λ .

Lemma 30. Let λ^* be any critical point of the dual problem of (7) and $g(\lambda)$ be its dual objective (in the sense of Boyd and Vandenberghe (2004, Chapter 5)). Moreover, let $(\bar{\mathbf{x}}^*)^T M_0(\bar{\mathbf{x}}^*) - g(\lambda^*)$ be the duality gap between the primal in (7) and its dual. Then, any regular critical point $\bar{\mathbf{x}}^*$ of the primal has zero duality gap. Consequently, the duality gap is zero if and only if λ^* is also a critical point of

$$\begin{aligned} \max_{\lambda} \quad & \left(- \sum_{\gamma \in \Gamma'} \lambda_{\gamma} \right) \\ \text{s.t. } \det G(\lambda) &= 0 \end{aligned} \tag{C.3}$$

Proof. First, we compute the dual function (see Boyd and Vandenberghe (2004, Page 216)) of the objective in (7) as $g(\lambda) = \inf_{\bar{\mathbf{x}} \in \mathbb{R}^n} \mathcal{L}(\bar{\mathbf{x}}, \lambda) = (\inf_{\bar{\mathbf{x}} \in \mathbb{R}^n} \bar{\mathbf{x}}^T G(\lambda) \bar{\mathbf{x}}) - \sum_{\gamma \in \Gamma'} \lambda_{\gamma}$ where $\mathcal{L}(\bar{\mathbf{x}}, \lambda)$ is the Lagrangian given in (C.2). Therefore, we conclude that

$$g(\lambda) = \begin{cases} - \sum_{\gamma \in \Gamma'} \lambda_{\gamma} & \text{if } G(\lambda) \geq 0 \text{ or } \det G(\lambda) = 0 \\ -\infty & \text{otherwise} \end{cases}$$

since the infimum of the form $\bar{\mathbf{x}}^T G(\lambda) \bar{\mathbf{x}}$ is either zero (for $G(\lambda) \geq 0$ or $\det G(\lambda) = 0$ with $\bar{\mathbf{x}}$ in the kernel of $G(\lambda)$) or $-\infty$ (see Boyd and Vandenberghe (2004, Page 220)). Hence, the dual program results in

$$\begin{aligned} \max_{\lambda} \quad & \left(- \sum_{\gamma \in \Gamma'} \lambda_{\gamma} \right) \\ \text{s.t. } G(\lambda) &\geq 0 \end{aligned} \tag{C.4}$$

making the constraints explicit (see Boyd and Vandenberghe (2004, Page 224)), ignoring the case when $g(\lambda) = -\infty$. Note that

due to Lemma 29 it is true that $G(\lambda^*)\bar{\mathbf{x}}^* = 0$ for some λ^* . Hence, we can multiply

$$\begin{aligned} (\bar{\mathbf{x}}^*)^T G(\lambda^*) \bar{\mathbf{x}}^* &= (\bar{\mathbf{x}}^*)^T M_0(\bar{\mathbf{x}}^*) + \sum_{\gamma \in \Gamma'} \lambda_{\gamma}^* (\bar{\mathbf{x}}^*)^T M_{\gamma}(\bar{\mathbf{x}}^*) \\ &= (\bar{\mathbf{x}}^*)^T M_0(\bar{\mathbf{x}}^*) + \sum_{\gamma \in \Gamma'} \lambda_{\gamma}^* = (\bar{\mathbf{x}}^*)^T M_0(\bar{\mathbf{x}}^*) - g(\lambda^*) = 0 \end{aligned} \tag{C.5}$$

which results in zero duality gap, only possible if λ^* is a critical point of the dual problem by the definition of $g(\lambda)$. Note that zero duality gap plus the condition that $(\bar{\mathbf{x}}^*)^T M_{\gamma}(\bar{\mathbf{x}}^*) = 1$ implies $\bar{\mathbf{x}}^* \neq 0$. Then, $G(\lambda^*)\bar{\mathbf{x}}^* = 0$ if and only if $\det G(\lambda^*) = 0$ and $\bar{\mathbf{x}}^*$ in the kernel of $G(\lambda^*)$.

Using these results, we proceed to show Theorem 13. Build a Lagrangian for (C.3) as $\mathcal{L}'(\lambda, \mu) = - \sum_{\gamma \in \Gamma'} \lambda_{\gamma} + \mu \det G(\lambda)$ with Lagrange multiplier $\mu \in \mathbb{R}$. Thus, λ^* comply with $\nabla_{\lambda} \mathcal{L}'(\lambda^*, \mu) = 0$ (Bertsekas, 1999, Proposition 3.1.1) and $\partial \mathcal{L}' / \partial \lambda_{\gamma} = -1 + \mu \text{tr}(G(\lambda^*)^{\dagger} M_{\gamma}) = 0$ using Jacobi's formula for the derivative of the determinant (Horn & Johnson, 2012, Page 29). Thus, $\text{tr}(G(\lambda^*)^{\dagger} M_{\gamma}) = 1/\mu$ or equivalently

$$\text{tr}(G(\lambda^*)^{\dagger} M_{\gamma}) = \text{tr}(G(\lambda^*)^{\dagger} M_{\nu}), \quad \forall \gamma, \nu \in \Gamma' \tag{C.6}$$

This result, in addition to the condition $\det G(\lambda^*) = 0$ result in (9). From Lemma 30 we can also conclude that any critical point complies with $(\bar{\mathbf{x}}^*)^T M_0(\bar{\mathbf{x}}^*) = - \sum_{\gamma \in \Gamma'} \lambda_{\gamma}^*$ since this condition is equivalent to the zero duality gap property of (C.3). Finally, due to Lemma 29, we know that $G(\lambda^*)\bar{\mathbf{x}}^* = 0$ and this is true only if $\bar{\mathbf{x}}^*$ is in the kernel of $G(\lambda^*)$.

C.5. Proof of Corollary 15

Recall that the rank of $G(\lambda)$ is equivalent to the size of the largest invertible sub-matrix $G(\lambda)$ (Horn & Johnson, 2012, Page 12). We proceed by contradiction: assume that λ does not comply with at least one Eq. (11) for some pair (i, j) . This would mean that $G_{ij}(\lambda)$ is invertible resulting in the rank of $G(\lambda)$ to be $n - 1$. However, the nullity of $G(\lambda)$ was greater than 1. Then, by the rank-nullity theorem (Horn & Johnson, 2012, Page 6) the rank should have been less than $n - 1$ leading to a contradiction.

C.6. Proof of Proposition 16

Let $V = \{\mathbf{v}_1, \dots, \mathbf{v}_r\}$ be arbitrarily $r = n - |\Gamma'|$ vectors $\mathbf{v}_1, \dots, \mathbf{v}_r \in \mathbb{R}^n$. Build a matrix $W_V(\bar{\mathbf{x}}) = [W(\bar{\mathbf{x}}), \mathbf{v}_1, \dots, \mathbf{v}_r]$. The vectors $\{M_{\gamma} \bar{\mathbf{x}}\}_{\gamma \in \Gamma'}$ are linearly dependent if and only if the matrix $W_V(\bar{\mathbf{x}})$ is singular for any choice of $\mathbf{v}_1, \dots, \mathbf{v}_r$. Using the Laplace expansion of the determinant of $W_V(\bar{\mathbf{x}})$ (Horn & Johnson, 2012, Page 8) we obtain $\det W_V(\bar{\mathbf{x}}) = \sum_{\alpha \in \mathcal{A}} q_{\alpha}(V) \det W_{\alpha}(\bar{\mathbf{x}})$ where $q_{\alpha}(V)$ are polynomials in the components of the vectors in V . In order to comply $\det W_V(\bar{\mathbf{x}}) = 0$ for any set of vectors V , then $\det W_{\alpha}(\bar{\mathbf{x}}) = 0$ for all $\alpha \in \mathcal{A}$.

C.7. Proof of Proposition 19

The proof follows by performing a Karp reduction (Korte & Vygen, 2012, Definition 15.15) of the 3-SAT problem, known to be NP-complete (Korte & Vygen, 2012, Theorem 15.22), to an instance of Problem 3. First, consider an instance of Problem 3 with $A = 0, B = I$ and $\Delta^1 = \dots = \Delta^D = \Delta, T_f = \alpha \Delta, \alpha \in \mathbb{N}$ with $(I + L^1 \Delta), \dots, (I + L^D \Delta)$ being 0-1 left stochastic matrices. Moreover, let $P_0 = 0, W_0 = 0, \Sigma^1 = \dots = \Sigma^D = 0$ as well as $Q = 0, \lambda_r = 0, \lambda_x = 1$ and $Q_f = \text{diag}(c)$ for some 0-1 vector c . Finally, consider the initial condition $\bar{\mathbf{x}}_0$ to be a 0-1 vector as well. This results in $\Lambda(\Delta^{pk}) = I + L^{pk} \Delta$ being 0-1 matrices and as a consequence $\bar{\mathbf{x}}[\alpha]$ is a 0-1 vector. Thus,

$\mathcal{J}(p) = \bar{\mathbf{x}}[\alpha]^T \text{diag}(c) \bar{\mathbf{x}}[\alpha] = c^T \bar{\mathbf{x}}[\alpha]$. Hence, solving this instance of [Problem 3](#) requires to find a schedule $p_0, \dots, p_{\alpha-1}$ such that the linear function $c^T \bar{\mathbf{x}}[\alpha]$ is minimized, or equivalently $-c^T \bar{\mathbf{x}}[\alpha]$ is maximized. Thus, the Problem ([Wu & He, 2020](#), Problem (P)) for stochastic matrices, initial condition and vector c all with $0 - 1$ components can be reduced to [Problem 3](#). Finally, the proof of [Wu and He \(2020, Theorem 3.1\)](#) implies that the 3-SAT problem can be reduced to this instance of [Wu and He \(2020, Problem \(P\)\)](#).

C.8. Proof of [Proposition 20](#)

The proof of correctness follows by a direct application of the dynamic programming algorithm ([Bertsekas, 2000](#), Page 23). For the complexity, note that the largest amount of recursive calls in [Algorithm 6](#) is obtained when the schedule covering the smallest amount of time c over $[0, T_f]$ is chosen repeatedly. In this case, the depth of recursive calls to `dynprog` is $\lfloor T_f/c \rfloor$. Hence, due to line 2 in [Algorithm 6](#), the total number of recursive calls is at most $m^{\lfloor T_f/c \rfloor}$.

C.9. Proof of [Theorem 22](#)

For the proof, for any matrix P let $\|P\|$ denote its matrix norm ([Horn & Johnson, 2012](#), Section 5.6) induced by the Euclidean norm. To show [Theorem 22](#) we provide the following.

Lemma 31. *Let p be a fixed perception schedule over $[0, T_f]$. Moreover, let any compact sets $B_{\bar{\mathbf{x}}} \in \mathbb{R}^n$, $B_p \in \mathbb{R}^{n \times n}$ and any $\varepsilon > 0$. Then, there exists $\delta_{\bar{\mathbf{x}}}, \delta_p > 0$ such that for any $\bar{\mathbf{x}}_0, \bar{\mathbf{x}}'_0 \in B_{\bar{\mathbf{x}}}$ and any $P, P' \in B_p$ with $\|\bar{\mathbf{x}}_0 - \bar{\mathbf{x}}'_0\| < \delta_{\bar{\mathbf{x}}}$ and $\|P - P'\| < \delta_p$ implies $|\mathcal{J}(p; \bar{\mathbf{x}}_0, P_0) - \mathcal{J}(p; \bar{\mathbf{x}}'_0, P'_0)| < \varepsilon$.*

Proof. First, let $r_{\bar{\mathbf{x}}} = \sup\{\|\bar{\mathbf{x}}\| : \bar{\mathbf{x}} \in B_{\bar{\mathbf{x}}}\}$. Then, we leverage continuity of solutions of [\(A.5\)](#) with respect to initial conditions. Concretely, using the bound in [Khalil \(2002, Page 356 - Equation \(9.28\)\)](#) applied to [\(A.5\)](#) for the compact interval $[0, T_f]$ it follows that $\|\bar{\mathbf{x}}(t) - \bar{\mathbf{x}}'(t)\| \leq \|\bar{\mathbf{x}}_0 - \bar{\mathbf{x}}'_0\| \exp(c_{\bar{\mathbf{x}}} T_f) \leq \delta_{\bar{\mathbf{x}}} \exp(c_{\bar{\mathbf{x}}} T_f), \forall t \in [0, T_f]$ for some constant $c_{\bar{\mathbf{x}}} \in \mathbb{R}$. A similar reasoning leads to obtain $\|P(t) - P'(t)\| \leq \delta_p \exp(c_p T_f), \forall t \in [0, T_f]$ for some constant $c_p \in \mathbb{R}$. Moreover, $|\bar{\mathbf{x}}(t)^T Q \bar{\mathbf{x}}(t) - \bar{\mathbf{x}}'(t)^T Q \bar{\mathbf{x}}'(t)| = |(\bar{\mathbf{x}}(t) + \bar{\mathbf{x}}'(t))^T Q(\bar{\mathbf{x}}(t) - \bar{\mathbf{x}}'(t))| \leq 2r_{\bar{\mathbf{x}}} \|Q\| \delta_{\bar{\mathbf{x}}} \exp(c_{\bar{\mathbf{x}}} T_f)$ using the matrix norm properties in [Horn and Johnson \(2012, Section 5.6\)](#). Similarly, $|\text{tr}(QP(t)) - \text{tr}(QP'(t))| = |\text{tr}(Q(P(t) - P'(t)))| \leq n \|Q\| \delta_p \exp(c_p T_f)$. Hence, using these results on $e := |\mathcal{J}(p; \bar{\mathbf{x}}_0, P_0) - \mathcal{J}(p; \bar{\mathbf{x}}'_0, P'_0)|$ which using [\(A.6\)](#) leads to conclude that for appropriate (sufficiently small) $\delta_{\bar{\mathbf{x}}}, \delta_p > 0$: $e \leq 2\lambda_{\bar{\mathbf{x}}} r_{\bar{\mathbf{x}}} \delta_{\bar{\mathbf{x}}} (\|Q\| + \|Q_f\|) \exp(c_{\bar{\mathbf{x}}} T_f) + n \lambda_{\bar{\mathbf{x}}} \delta_p (\|Q\| + \|Q_f\|) \exp(c_p T_f) \leq \varepsilon$

We proceed to show [Theorem 22](#). First, given $\varepsilon > 0$ and $B_{\bar{\mathbf{x}}}, B_p$ choose $\delta_{\bar{\mathbf{x}}}, \delta_p$ as in [Lemma 31](#) and note that these values are independent on initial conditions as long as they lie in $B_{\bar{\mathbf{x}}}, B_p$. Hence, there is $N > 0$ such that we can partition the compact set $B_{\bar{\mathbf{x}}}$ into regions $B_{\bar{\mathbf{x}}}^1, \dots, B_{\bar{\mathbf{x}}}^N$ which cover all $B_{\bar{\mathbf{x}}}$ and $\|\bar{\mathbf{x}} - \bar{\mathbf{x}}^i\| \leq \delta_{\bar{\mathbf{x}}}, \forall \bar{\mathbf{x}}, \bar{\mathbf{x}}^i \in B_{\bar{\mathbf{x}}}^i, \forall i \in \{1, \dots, N\}$. A similar partition of N sections can be performed for B_p as well. Now, choose any $\bar{\mathbf{x}}_0^i \in B_{\bar{\mathbf{x}}}^i, P_0^i \in B_p^i$ for partition i and denote with p^{*i} the optimal schedule in this case for the general setting of [Problem 3](#). As a consequence of [Lemma 31](#), the cost of p^{*i} for any $\bar{\mathbf{x}}_0 \in B_{\bar{\mathbf{x}}}^i, P_0 \in B_p^i$ differs from the optimal one in that partition by at most ε . Now, let Γ be any admissible set of schedules. Moreover, since p^{*i} is a solution of [Problem 3](#), then it must be a stabilizing perception schedule for $\bar{\mathbf{x}}[0] = \bar{\mathbf{x}}_0^i$. Thus, there exists $\ell^i < \infty$ (sufficiently large) with $T_f \leq \sum_{k=0}^{\ell^i-1} \Delta p_k^{*i}$ such that $\|\bar{\mathbf{x}}[\ell^i]\|^2 < \|M_0\|^{-1} \min_{\gamma \in \Gamma} (\bar{\mathbf{x}}_0^i)^T M_{\gamma} \bar{\mathbf{x}}_0^i$ for the truncated schedule $\gamma^i = \{p_k^{*i}\}_{k=0}^{\ell^i-1}$. This last relation

implies $(\bar{\mathbf{x}}_0^i)^T M_{\gamma^i} \bar{\mathbf{x}}_0^i = (\Lambda^{\gamma^i} \bar{\mathbf{x}}_0^i)^T M_0 (\Lambda^{\gamma^i} \bar{\mathbf{x}}_0^i) = \bar{\mathbf{x}}[\ell^i]^T M_0 \bar{\mathbf{x}}[\ell^i] \leq \|M_0\| \|\bar{\mathbf{x}}[\ell^i]\|^2 < \min_{\gamma \in \Gamma} (\bar{\mathbf{x}}_0^i)^T M_{\gamma} \bar{\mathbf{x}}_0^i$. Hence, $\Gamma^i = \Gamma \cup \{\gamma^i\}$ is admissible and complies $\gamma^{*i}(\bar{\mathbf{x}}_0^i; \Gamma^i) = \gamma^i$ by [\(5\)](#) and similarly for any $\bar{\mathbf{x}}_0 \in B_{\bar{\mathbf{x}}}^i$ and $\delta_{\bar{\mathbf{x}}}$ sufficiently small. Therefore, the result follows by choosing $m = N$ and all $\Gamma^1, \dots, \Gamma^m$ constructed in this way, one for each partition. The reason is that for any $\bar{\mathbf{x}}_0 \in B_{\bar{\mathbf{x}}}, P_0 \in B_p$, a schedule with cost differing from the optimal one by at most ε is always evaluated in [Algorithm 6](#).

References

- Ahmadi, A. A., Jungers, R., Parrilo, P., & Roostbehani, M. (2014). Joint spectral radius and path-complete graph Lyapunov functions. *SIAM Journal on Control and Optimization*, 52, 687–717.
- Aldana-López, Rodrigo, Aragués, Rosario, & Sagüés, Carlos (2020). Attention vs. precision: latency scheduling for uncertainty resilient control systems. In *2020 59th IEEE conference on decision and control* (pp. 5697–5702).
- Anta, A., & Tabuada, P. (2010). To sample or not to sample: Self-triggered control for nonlinear systems. *IEEE Transactions on Automatic Control*, 55(9), 2030–2042.
- Antunes, D., & Heemels, W. P. M. (2017). Linear quadratic regulation of switched systems using informed policies. *IEEE Transactions on Automatic Control*, 62(6), 2675–2688.
- Åström, K. J. (1970). *Mathematics in science and engineering, Introduction to stochastic control theory*. Academic Press.
- Ballotta, L., Schenato, L., & Carlone, L. (2020). From sensor to processing networks: Optimal estimation with computation and communication latency. In *IFAC world congress*.
- Bertsekas, D. P. (1999). *Nonlinear programming*. Athena Scientific.
- Bertsekas, D. P. (2000). *Dynamic programming and optimal control* (2nd ed.). Athena Scientific.
- Blanchini, F., & Savorgnan, C. (2008). Stabilizability of switched linear systems does not imply the existence of convex Lyapunov functions. *Automatica*, 44(4), 1166–1170.
- Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge University Press.
- Campos-Macías, L., Aldana-López, R., de la Guardia, R., Parra-Vilchis, J. I., & Gómez-Gutiérrez, D. (2021). Autonomous navigation of MAVs in unknown cluttered environments. *Journal of Field Robotics*, 38(2), 307–326.
- Ding, J., Peres, Y., Ranade, G., & Zhai, A. (2016). When multiplicative noise stymies control. *ArXiv*, abs/1612.03239.
- Faessler, M., Franchi, A., & Scaramuzza, D. (2018). Differential flatness of quadrotor dynamics subject to rotor drag for accurate tracking of high-speed trajectories. *IEEE Robotics and Automation Letters*, 3(2), 620–626.
- Falanga, D., Kim, S., & Scaramuzza, D. (2019). How fast is too fast? The role of perception latency in high-speed sense and avoid. *IEEE Robotics and Automation Letters*, 4(2), 1884–1891.
- Fiacchini, M., & Jungers, M. (2014). Necessary and sufficient condition for stabilizability of discrete-time linear switched systems: A set-theory approach. *Automatica*, 50, 75–83.
- Fiacchini, M., Jungers, R., & Girard, A. (2018). Stabilization and control Lyapunov functions for language constrained discrete-time switched linear systems. *Automatica*, 93, 64–74.
- Forster, C., Zhang, Z., Gassner, M., Werlberger, M., & Scaramuzza, D. (2017). SVO: Semidirect visual odometry for monocular and multicamera systems. *IEEE Transactions on Robotics*, 33(2), 249–265.
- Fraundorfer, F., Heng, L., Honegger, D., Lee, G., Meier, L., Tanskanen, P., et al. (2012). Vision-based autonomous mapping and exploration using a quadrotor MAV. In *IEEE/RSJ IROS* (pp. 4557–4564).
- Gatsis, K., Hassani, H., & Pappas, G. J. (2021). Latency-reliability tradeoffs for state estimation. *IEEE Transactions on Automatic Control*, 66(3), 1009–1023.
- Geromel, J. C., & Colaneri, P. (2006). Stability and stabilization of discrete time switched systems. *International Journal of Control*, 79(7), 719–728.
- Griggs, W., King, C., Shorten, R., Mason, O., & Wulff, K. (2010). Quadratic Lyapunov functions for systems with state-dependent switching. *Linear Algebra and its Applications*, 433, 52–63.
- Hotel, L., Fiter, C., Omran, H., Seuret, A., Fridman, E., Richard, J. P., et al. (2017). Recent developments on the stability of systems with aperiodic sampling: An overview. *Automatica*, 76, 309–335.
- Horn, R. A., & Johnson, C. R. (2012). *Matrix analysis* (2nd ed.). USA: Cambridge University Press, ISBN: 0521548233.
- Khalil, H. K. (2002). *Nonlinear systems* (3rd ed.). Upper Saddle River, NJ: Prentice-Hall.
- Korte, B., & Vygen, J. (2012). *Combinatorial optimization: Theory and algorithms* (5th ed.). Springer Publishing Company, Incorporated.

- Kubler, F., Renner, P., & Schmedders, K. (2014). Chapter 11 - computing all solutions to polynomial equations in economics. In Karl Schmedders, & Kenneth L. Judd (Eds.), *Handbook of computational economics, Vol. 3* (pp. 599–652). Elsevier.
- Liberzon, D. (2003). *Systems & control, Switching in systems and control*. Birkhauser.
- Mellinger, D., & Kumar, V. (2011). Minimum snap trajectory generation and control for quadrotors. In *2011 IEEE ICRA* (pp. 2520–2525). <http://dx.doi.org/10.1109/ICRA.2011.5980409>.
- Oleynikova, H., Taylor, Z., Siegwart, R., & Nieto, J. (2018). Safe local exploration for replanning in cluttered unknown environments for micro-aerial vehicles. *IEEE Robotics and Automation Letters*.
- Pant, Y. V., Abbas, H., Mohta, K., Quaye, R. A., Nghiem, T. X., Devietti, J., et al. (2021). Anytime computation and control for autonomous systems. *IEEE Transactions on Control Systems Technology*, 29(2), 768–779.
- Ristevski, S., Yucelen, T., & Muse, J. A. (2021). An event-triggered distributed control architecture for scheduling information exchange in networked multiagent systems. *IEEE Transactions on Control Systems Technology*, 1–12.
- Rockafellar, R. Tyrrell (1970). *Princeton Mathematical Series, Convex analysis*. Princeton, N. J.: Princeton University Press.
- Shorten, R., Wirth, F., Mason, O., Wulff, K., & King, C. (2007). Stability criteria for switched and hybrid systems. *SIAM Review*, 49, 545–592.
- Siegwart, R., & Nourbakhsh, I. R. (2004). *A Bradford book, Introduction to autonomous mobile robots*. Bradford Book.
- Soderstrom, T. (2002). *Discrete-time stochastic systems: Estimation and control* (2nd ed.). Berlin, Heidelberg: Springer-Verlag.
- Strasdat, H. M., Montiel, J. M. M., & Davison, A. J. (2012). Visual SLAM: Why filter? *Image and Vision Computing*, 30, 65–77.
- Verschelde, J. (1999). Algorithm 795: Phcpack: A general-purpose solver for polynomial systems by homotopy continuation. *ACM Transactions on Mathematical Software*, 25(2), 251–276.
- van der Weijst, R., van Loon, B., Heertjes, M., & Heemels, M. (2018). Scheduled controller design for systems with varying sensor configurations: A frequency-domain approach. *IEEE Transactions on Control Systems Technology*, 26(2), 523–534.
- Wu, Z., & He, Q. (2020). Optimal switching sequence for switched linear systems. *SIAM Journal on Control and Optimization*, 58, 1183–1206.



Rodrigo Aldana-Lopez received the B.Sc. degree in electronics engineering from the Instituto Tecnológico y de Estudios Superiores de Occidente, Tlaquepaque, Mexico, in 2016. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Systems Engineering, University of Zaragoza, Zaragoza, Spain. His research interests include real-time robotics, machine learning, and communication protocols and control for distributed systems.



Rosario Aragiús, received the M.S. and Ph.D. degrees in system engineering and computer science from the University of Zaragoza, Zaragoza, Spain, in 2008 and 2012, respectively. She was a Postdoctoral Researcher with the Institut Pascal, CNRS, UMR 6602, Clermont-Ferrand, France, from April 2012 to October 2013. She is currently an Associate Professor with the Department of Computer Science and Systems Engineering, University of Zaragoza. Her research interests include multirobot perception and control using distributed and decentralized strategies.



Carlos Sagüés received the M.S. degree in computer science and systems engineering and the Ph.D. degree in industrial engineering from the University of Zaragoza, Zaragoza, Spain, in 1989 and 1992, respectively. In 1994 he joined, as an Associate Professor with the Departamento de Informática e Ingeniería de Sistemas, University of Zaragoza, Zaragoza, Spain, where he became a Full Professor in 2009, and was also the Head Teacher. He was engaged in research on force and infrared sensors for robots. His current research interests include control systems and industry applications, computer vision, visual control, and multivehicle cooperative control.