2020 59th IEEE Conference on Decision and Control (CDC)
Jeju Island, Republic of Korea, December 14-18, 2020

Attention vs. precision: latency scheduling for uncertainty resilient control systems.

Rodrigo Aldana-López, Rosario Aragüés and Carlos Sagüés

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 clear compromise between perception latency and the stability of the overall robotic system, referred here as the latency-precision trade-off. Opposed to periodic sampling, which forces a particular perception latency, we study for the first time the computation of the best sequence of latencies, obtaining different precision at different moments in time. Using this concept we give a formal problem formulation of the attention-precision trade-off. Moreover, we give a method to obtain sequences of latencies that improve the system performance which is verified by simulation experiments.**

I. Introduction

A key ingredient to achieve autonomy of mobile robots is the perception task, which consists of using sensors to obtain an estimate of the state of the robot, as well as the state of the environment. Then, such state of the system can be used to close a feedback loop in order to make the robot follow a desired behaviour [1]. The term perception latency has been used in [2] 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 by using, for example, high resolution images, large number of features and robust feature extraction methods will result in a better perception quality. However, it is clear that such strategy will take more time depending on the robot computing platform. This might be undesired, since long sampling intervals can prevent the control commands to effectively stabilize the system. Hence, there is a clear compromise between perception latency and the stability of the overall robotic system, from now on referred as the attention-precision trade-off. In existing literature [1], this trade-off is tackled by the following principle: given a maximum sampling time T_s driving a stable behaviour to the robot, use the largest perception latency possible. In other words, T_s poses a real-time constraint on the perception task of the robot. However, a different approach to overcome the attention-precision trade-off is to use different perception configurations at different moments in time maintaining or improving the precision of the control task with respect to periodic perception setting.

The problem of choosing which latency to use at each time can be traduced in some form of a non-uniform or a-periodic sampling problem. Recently, many works have adopted the notions of event-triggered and self-triggered sampling for this purpose [3], [4], [5]. In these sampling schemes, a state dependant triggering rule for sampling is obtained by means of sufficient stability conditions. However, these methods have the drawback that the sampling sequences do not take into account that shorter sampling intervals might lead to poor state estimates discussed before. Moreover, there is an inherent delay induced to the control system due to the perception latency requiring to use predictions of the current state. In this context, event-triggered and self-triggered sampling are more related to solving the problem of *minimum attention control* [5], while preserving stability rather than concerning on the particular effects that the attention-precision trade-off might generate. Despite this, such minimum attention control policies have interesting implications from an energy consumption, or communication bandwidth points of view. In [6], a rule to choose between two sampling frequencies is derived. However, this result is conservative since only high enough frequencies are used such to stabilize the system with any of them separately. Variable sampling intervals can also be studied in the context of switching-systems [7], [8]. However, as stated in [8], the problem of choosing a particular sampling sequence instead of guaranteeing stability over all possible sampling sequences is still an emerging research area.

A. Contributions

In this work, we give for the first time a formal statement of the attention-precision trade-off problem in the form of what we call the *latency-scheduling* problem, taking into account the trade-off implications between perception latency and state estimation quality. The results presented here show possible solutions to the *latency-scheduling* problem. To do this, a statistical description of a particular class of systems is given from which algorithms to solve the *latency-scheduling* problem are proposed, based in a moving horizon scheme. Moreover, simulation examples are shown to test the algorithms presented here, from which interesting behaviours derive. In particular to state how unstable sampling latencies combined with other operational latencies give stable behaviour, and how a combination of control task quality and minimum attention goals gives good results.

This work was supported by projects COMMANDIA SOE2/P1/F0638 (Interreg Sudoe Programme, ERDF), PGC2018-098719-B-I00 (MCIU/ AEI/ FEDER, UE) and DGA T45-17R (Gobierno de Aragon). The first author was funded by "Universidad de Zaragoza - Santander Universidades" program.

The authors are with Departamento de Informatica e Ingenieria de Sistemas (DIIS) and Instituto de Investigacion en Ingenieria de Aragon (I3A), Universidad de Zaragoza, Zaragoza 50018, Spain. (e-mail: rodrigo.aldana.lopez@unizar.es, raragues@unizar.es, csagues@unizar.es)

B. Notation

Let $\mathbb{R}$ and $\mathbb{N}$ represent the sets of real and natural numbers respectively and $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$. Through this work, $\mathbf{x}(t) \in \mathbb{R}^n$ represent a continuous-time signal evaluation at $t \in \mathbb{R}$. Moreover, $\mathbf{x}[k] \triangleq \mathbf{x}(\tau_k)$ 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 $\tau_k \leq \tau_{k+1}$. Furthermore, 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. Given a matrix $M \in \mathbb{R}^{n\times n}$, then $M^{1/2}$ is a matrix satisfying $M = M^{1/2}M^{1/2}$. $\mathbb{E}\{\bullet\}$ represents the expectation operator whereas $\text{cov}\{\bullet\} = \mathbb{E}\{(\bullet - \mathbb{E}\{\bullet\})(\bullet - \mathbb{E}\{\bullet\})^T\}$ represents the covariance operator. Additionally, the symbol $\mathcal{N}(\bar{\mathbf{x}}, \Sigma)$ stands for a multi-variate Gaussian distribution with mean $\bar{\mathbf{x}} \in \mathbb{R}^n$ and covariance matrix $\Sigma \in \mathbb{R}^{n\times n}$.

II. Problem statement

Consider a mobile robot described by the hybrid system

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}[k], \quad \forall t \in [\tau_k, \tau_{k+1}) \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the robot state (e.g. position and velocity for a particle robot) and $\mathbf{u} \in \mathbb{R}^m$ is the input vector (e.g. representing the actuator forces induced to the system by the robot) given sequence of instants $\tau = \{\tau_k\}_{k=0}^{\infty}$ with $\tau \in \mathbb{R}^n$ and $\tau_k \leq \tau_{k+1}$. We are interested in studying the error dynamics of the actual robot state with respect to a desired behaviour. Hence, to simplify the exposition, $\mathbf{x}(t)$ will represent such error which it is desired to maintain it as close to the origin as possible.

To present the main goal in this work, the rules of the scenario under which the robot is constrained are described in the following, which are motivated by real implementation issues, such as discrete sampling and precision in state estimation due to limited computing power:

Assumption 1: The robot described by system (1) is governed by the rules:

i) The pair (A, B) is controllable.
ii) The robot can take samples of sensors in the environment at discrete moments of time τ_k. Under this information, system (1) is observable.
iii) Different estimates $\{\hat{\mathbf{x}}^1[k], \ldots, \hat{\mathbf{x}}^D[k]\}$ of $\mathbf{x}[k]$ can be obtained under a set of different methods or algorithms indexed by $\{1, \ldots, D\}$ which calculate them using sensors information. Each method in $\{1, \ldots, D\}$ has a different corresponding latency in $\{\Delta^1, \ldots, \Delta^D\}$.
iv) The estimation method $p_k \in \{1, \ldots, D\}$, corresponding to $t = \tau_k$, has a precision depending on how much time Δ^{p_k} it takes to compute its estimate, available until $t = \tau_k + \Delta^{p_k}$. This is, $\hat{\mathbf{x}}^{p_k}[k] = \mathbf{x}[k] + \mathbf{n}^{p_k}[k]$, where $\mathbf{n}^{p_k}[k]$ is a noise induced by the precision of p_k-th method. The noises $\{\mathbf{n}^1[k], \ldots, \mathbf{n}^D[k]\}$ are assumed to be independent random variables for all $k \in \mathbb{N}_0$ and normally distributed with zero mean and covariance matrices $\{\Sigma^1, \ldots, \Sigma^D\}$ depending on characteristics of the methods $\{1, \ldots, D\}$.

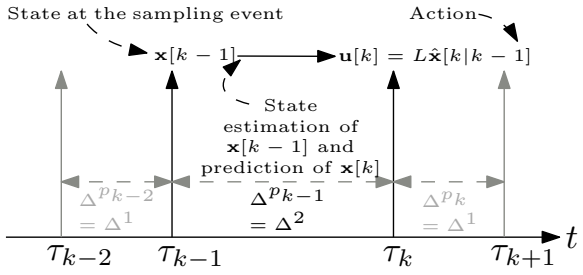

Fig. 1. Timing diagram for a simple non-uniform measure-action scheme with two different latencies $\{\Delta^1, \Delta^2\}$ as described in Example 1

v) If a sensor takes a sample at $t = \tau_k$ and a method $p_k \in \{1, \ldots, D\}$ is chosen then a new sample is taken only after the method has finished. Therefore, $\tau_{k+1} = \tau_k + \Delta^{p_k}$.

Under this setting, the main issue we try to address is to choose which estimation method to use at each time slot $[\tau_k, \tau_{k+1})$. This motivates the following definition.

Definition 1: (Latency Schedule) Under Assumption 1, p is a latency schedule if it is a sequence of estimation methods, i.e. $p = \{p_k\}_{k=0}^{\infty}$ with $p_k \in \{1, \ldots, D\}$.

Furthermore, to simplify the exposition we introduce the following notation: when using a particular latency schedule p, we refer to $\hat{\mathbf{x}}$ and $\mathbf{n}$ as the signals defined by $\hat{\mathbf{x}}[k] \triangleq \hat{\mathbf{x}}^{p_k}[k]$ and $\mathbf{n}[k] \triangleq \mathbf{n}^{p_k}[k]$, i.e., the resulting estimation signal of $\mathbf{x}$ and the resulting noise signal $\mathbf{n}$ respectively. Moreover, let $\delta \triangleq \{\Delta^{p_k}\}_{k=0}^{\infty}$ and $\Sigma \triangleq \{\Sigma^{p_k}\}_{k=0}^{\infty}$ represent the sequence of resulting latencies and covariance matrices respectively under p.

Now, consider a linear control strategy for designing the form of $\mathbf{u}[k]$. Note that at $t = \tau_k$ there is no direct available information of $\mathbf{x}[k]$, since the system takes Δ^{p_k} units of time to compute it. Hence, applying $\mathbf{u}[k] = L\mathbf{x}[k]$ explicitly is not possible. Therefore, a prediction is used to obtain an estimate of $\mathbf{x}[k]$ using the information $\hat{\mathbf{x}}[k-1]$ and the model given in (1). This prediction signal is represented as $\hat{\mathbf{x}}[k|k-1]$, i.e., the estimation of $\mathbf{x}[k]$ given the information we have until $t = \tau_{k-1}$. The following example is presented to clarify some of these aspects.

Example 1: Let us consider a latency schedule p that satisfies $p_{k-2} = 1$, $p_{k-1} = 2$ and $p_k = 1$ for some k. Moreover, there are two possible choices of estimation methods $\{1, 2\}$ with execution latencies $\{\Delta^1, \Delta^2\}$. Under this circumstance, the following sequence of events happens from $t = \tau_{k-1}$ to $t = \tau_k$ as depicted in Fig. 1:

1) At $t = \tau_{k-1}$ the unknown state of the plant is $\mathbf{x}[k-1]$ and sensors take measurements.
2) The estimation method 2 is chosen to compute an estimate of $\mathbf{x}[k-1]$
3) After Δ^2 units of time have elapsed at $t = \tau_k$, the estimate $\hat{\mathbf{x}}[k-1] = \hat{\mathbf{x}}^2[k-1] = \mathbf{x}[k-1] + \mathbf{n}^2[k-1]$ is used to compute a prediction $\hat{\mathbf{x}}[k|k-1]$ of $\mathbf{x}[k]$. The control input is calculated as $\mathbf{u}[k] = L\hat{\mathbf{x}}[k|k-1]$.
4) Just in $t = \tau_k$, $\mathbf{u}[k]$ is applied to the system.

Given this exposition, we aim to obtain an algorithm which finds the best latency schedule p such that a given performance measure is optimized. In order to focus in facet of the full scenario, let the control gain L be designed as if no sampling mechanism is present:

Assumption 2: Let A, B be as in Assumption 1, then L is chosen such that the eigenvalues of $A+BL$ have all negative real components.

Moreover, Assumption 1 motivates the assumption that no initial conditions at τ_0 are known but the initial information as given as in the following:

Assumption 3: Fix $k = 0$. Then, let p_{k-2} and p_{k-1} be two given method decisions from which their corresponding latencies $\Delta^{p_{k-2}}, \Delta^{p_{k-1}}$ define the time instants $\tau_{k-2} \triangleq \tau_{k-1} - \Delta^{p_{k-2}} \leq \tau_{k-1} \triangleq \tau_k - \Delta^{p_{k-1}} \leq \tau_k$. Moreover, assume that the values of $\mathbf{x}[k-2]$ and $\mathbf{x}[k-1]$ are random variables distributed as $\mathcal{N}(\bar{\mathbf{x}}[k-2], P[k-2])$ and $\mathcal{N}(\bar{\mathbf{x}}[k-1], P[k-1])$ respectively, for given $\bar{\mathbf{x}}[k-1], \bar{\mathbf{x}}[k-2], P[k-1]$ and $P[k-2]$. This initial information is denoted as $\mathcal{I}_0 = \{\tau_k, p_{k-1}, p_{k-2}, \bar{\mathbf{x}}[k-1], \bar{\mathbf{x}}[k-2], P[k-1], P[k-2]\}$.

In this work, the performance of a latency schedule p is measured as a combination of two costs. First, we are concerned about an expected cost for the trajectories $\mathbf{x}(t)$ of (1) in a finite-time horizon interval $[\tau_0, \tau_0 + T_f]$ given the possible realizations of the process $\{\mathbf{n}[k] : k \in \mathbb{N}_0\}$. Second, we are concerned on the cost of the attention to the system, where by attention we mean the amount to sampling-actuation events in the interval $[\tau_0, \tau_0 + T_f]$:

Definition 2: (Attention of a latency schedule) Given an interval $[\tau_\ell, \tau_\ell + T_f]$ and a latency schedule p, the attention of p is defined as $\mathcal{A}(p; \tau_\ell, T_f) = \max\{n : \sum_{k=\ell}^{n} \Delta^{p_k} \leq T_f\}$.

Henceforth, the cost of a latency schedule p under Assumptions 1-3 is defined as:

$$\begin{aligned}\mathcal{J}(p; \mathcal{I}_0, T_f) = \lambda_x \mathbb{E}\left\{ \int_{\tau_0}^{\tau_0+T_f} \mathbf{x}(t)^T Q \mathbf{x}(t) \, \mathrm{d}t \right\} \\ + \lambda_\delta \mathcal{A}(p; \tau_0, T_f).\end{aligned} \tag{2}$$

where Q is a positive definite matrix, and both objectives are pondered by the multipliers $\lambda_x, \lambda_\delta \geq 0$. Consequently, we are in position to formally state the problem:

Problem 1 (Latency Scheduling Control Problem): Given Assumptions 1-3 and the cost defined in (2), find a latency schedule p^* such that $\mathcal{J}(p^*; \mathcal{I}_0, T_f) \leq \mathcal{J}(p; \mathcal{I}_0, T_f)$ for any other latency schedule p.

In subsequent sections we present first, how to compute (2) explicitly given initial data $\mathcal{I}_0$ and second, we propose some algorithms to tackle Problem 1.

III. Cost function computation

In order to compute (2) for a particular latency schedule, note that $\mathbf{x}(t)$ behaves as a discrete-time Gaussian process despite the fact that it is a continuous time signal. Hence, $\mathbf{x}(t)$ will be distributed as $\mathcal{N}(\bar{\mathbf{x}}(t), P(t))$ for some deterministic signals $\bar{\mathbf{x}}(t)$ and $P(t)$ which can be obtained by solving a set of linear difference equations. This leads to an expression for the cost written in terms of $\bar{\mathbf{x}}(t)$ and $P(t)$ as described in the following.

A. Inter-event dynamics

In order to obtain the probability distribution of $\mathbf{x}(t)$ we show that since (1) has constant control input $\mathbf{u}[k]$ in the interval $[\tau_k, \tau_{k+1})$, then, there is an underlying discrete time dynamical system from which the trajectories of (1) can be reproduced.

Lemma 1: Let p be a latency schedule, $\mathbf{u}[k] = L\hat{\mathbf{x}}[k|k-1]$ where $\hat{\mathbf{x}}[k|k-1]$ is the model based prediction of $\mathbf{x}[k]$ given by,

$$\hat{\mathbf{x}}[k|k-1] = \tilde{A}(\Delta^{p_{k-1}})\hat{\mathbf{x}}[k-1] + \tilde{B}(\Delta^{p_{k-1}})\mathbf{u}[k-1], \tag{3}$$

and $\tilde{A}(\tau) \triangleq \exp(A\tau)$ and $\tilde{B}(\tau) \triangleq \int_0^\tau \exp(A\tau) \, \mathrm{d}\tau B$. Then, all solutions of (1) (in the sense of Caratheodory [9]), satisfy

$$\begin{aligned}\mathbf{x}(t) = \left(\tilde{A}(t-\tau_k) + \tilde{B}(t-\tau_k)L\right)\mathbf{x}[k] \\ + \tilde{B}(t-\tau_k)L\tilde{A}(\Delta^{p_{k-1}})\mathbf{n}[k-1], \quad \forall t \in [\tau_k, \tau_{k+1})\end{aligned} \tag{4}$$

Proof: Since (1) has constant control input $\mathbf{u}[k]$ in the interval $[\tau_k, \tau_{k+1})$, (1) can be solved explicitly for $\mathbf{x}(t)$ in that interval leading to

$$\begin{aligned}\mathbf{x}(t) &= \exp(A(t-\tau_k))\mathbf{x}[k] + \int_0^{t-\tau_k} \exp(A\tau) \, \mathrm{d}\tau B\mathbf{u}[k] \\ &= \tilde{A}(t-\tau_k)\mathbf{x}[k] + \tilde{B}(t-\tau_k)\mathbf{u}[k].\end{aligned} \tag{5}$$

Hence, from (5) it is clear why the prediction $\hat{\mathbf{x}}[k|k-1]$ takes the form in (3). Moreover, (5) can be used to compute $\mathbf{x}[k]$ in order to obtain the estimation error of (3), leading to

$$\begin{aligned}\hat{\mathbf{x}}[k|k-1] - \mathbf{x}[k] &= \tilde{A}(\Delta^{p_{k-1}})\hat{\mathbf{x}}[k-1] + \tilde{B}(\Delta^{p_{k-1}})\mathbf{u}[k-1] \\ &\quad - \tilde{A}(\Delta^{p_{k-1}})\mathbf{x}[k-1] - \tilde{B}(\Delta^{p_{k-1}})\mathbf{u}[k-1] \\ &= \tilde{A}(\Delta^{p_{k-1}})\mathbf{n}[k-1]\end{aligned}$$

In this way, $\mathbf{u}[k] = L\mathbf{x}[k] + L\tilde{A}(\Delta^{p_{k-1}})\mathbf{n}[k-1]$ which by substituting in (5) leads to (4). ∎

Henceforth, from (4) the underlying discrete close loop dynamics behind the continuous dynamics in (1) can be obtained by evaluating at $t = \tau_{k+1}$:

Corollary 1: Given the conditions of Lemma 1, the solutions of (1) satisfy

$$\begin{aligned}\mathbf{x}[k+1] = \left(\tilde{A}(\Delta^{p_k}) + \tilde{B}(\Delta^{p_k})L\right)\mathbf{x}[k] \\ + \tilde{B}(\Delta^{p_k})L\tilde{A}(\Delta^{p_{k-1}})\mathbf{n}[k-1], \quad \forall k \in \mathbb{N}_0.\end{aligned} \tag{6}$$

B. Probabilistic description of $\mathbf{x}(t)$

In this section we show that the solution to the linear difference equation in (6) leads to a discrete-time stochastic process in which $\mathbf{x}[k]$ is a normal random variable.

Lemma 2: Let Assumptions 1-3 and initial conditions from Assumption 3. Given a latency schedule p, then, $\mathbf{x}(t)$ is distributed as $\mathcal{N}(\bar{\mathbf{x}}(t), P(t))$ for all $t \in [\tau_k, \tau_{k+1})$ with

$$\begin{aligned}\bar{\mathbf{x}}(t) &= \Lambda(t-\tau_k)\bar{\mathbf{x}}[k], \\ P(t) &= \Lambda(t-\tau_k)P[k]\Lambda(t-\tau_k)^T \\ &\quad + \Gamma(t-\tau_k, \Delta^{p_{k-1}})\Sigma_{k-1}\Gamma(t-\tau_k, \Delta^{p_{k-1}})^T\end{aligned} \tag{7}$$

where $\Lambda(\tau) \triangleq \tilde{A}(\tau) + \tilde{B}(\tau)L$, $\Gamma(\tau,\tau') \triangleq \tilde{B}(\tau)L\tilde{A}(\tau')$ and $\mathbf{x}[k]$ and $P[k]$ are obtained from

$$\begin{aligned}\bar{\mathbf{x}}[k+1] &= \Lambda(\Delta^{p_k})\bar{\mathbf{x}}[k],\\ P[k+1] &= \Lambda(\Delta^{p_k})P[k]\Lambda(\Delta^{p_k})^T \\ &\quad + \Gamma(\Delta^{p_k},\Delta^{p_{k-1}})\Sigma_{k-1}\Gamma(\Delta^{p_k},\Delta^{p_{k-1}})^T.\end{aligned} \tag{8}$$

Proof: First, (8) follows by a direct application of Theorem 2 in the Appendix onto (6). This result implies that $\mathbf{x}(t)$ is distributed as $\mathcal{N}(\bar{\mathbf{x}}[k], P[k])$ at the instants $t = \tau_k$. Now, we are interested in what happens in the step from τ_k towards a fixed t given that we have (4), which is precisely the difference equation describing this step. Hence, Theorem 2 is applied to (4) resulting in (7). ∎

C. Cost function evaluation

In this section we take advantage of the probability distribution of $\mathbf{x}(t)$ from Lemma 2 to obtain a closed form expression of the cost $\mathcal{J}(p;\mathcal{I}_0,T_f)$ in (2).

Theorem 1: Let Assumptions 1-3 hold. Given $\bar{\mathbf{x}}(t)$ and $P(t)$ as in the statement of Lemma 2 then,

$$\begin{aligned}\mathcal{J}(p;\mathcal{I}_0,T_f) &= \lambda_x \int_{\tau_0}^{\tau_0+T_f} \bar{\mathbf{x}}(t)^T Q \bar{\mathbf{x}}(t)\mathrm{d}t \\ &\quad + \lambda_x \int_{\tau_0}^{\tau_0+T_f} \text{tr }(QP(t))\mathrm{d}t + \lambda_\delta \mathcal{A}(p;\tau_0,T_f)\end{aligned} \tag{9}$$

Proof: First, note that from Lemma 2 we know that $\mathbf{x}(t)$ is distributed as $\mathcal{N}(\bar{\mathbf{x}}(t), P(t))$. Then using Lemma 3 from the Appendix it follows that

$$\mathbb{E}\left\{\int_{\tau_0}^{\tau_0+T_f} \mathbf{x}(t)^T Q\mathbf{x}(t)\ \mathrm{d}t\right\} = \int_{\tau_0}^{\tau_0+T_f} \mathbb{E}\{\mathbf{x}(t)^T Q\mathbf{x}(t)\}\ \mathrm{d}t$$

$$= \int_{\tau_0}^{\tau_0+T_f} \left(\bar{\mathbf{x}}(t)^T Q\bar{\mathbf{x}}(t) + \text{tr }(QP(t))\right)\mathrm{d}t$$

which leads to (9) by multiplying by λ_x and adding $\lambda_\delta \mathcal{A}(p;\tau_0,T_f)$. ∎

IV. Sub-optimal solution for the latency-scheduling control problem

Theorem 1 gives us a way to evaluate explicitly the cost of a certain latency schedule p given some initial information $\mathcal{I}_0$. However, if we aim to evaluate all possible sequences, this becomes infeasible pretty quickly as we increase the horizon T_f. In fact, if $\mathcal{A}_{\max} \triangleq \max_p \mathcal{A}(p;\tau_0;T_f)$, then the amount of sequences we have to check is in the order of $D^{\mathcal{A}_{\max}}$ where D is the number of estimation methods from Assumption 1. Take as an example $D = 2$, $\Delta^1 = 10^{-3}, \Delta^2 = 10^{-2}$ and $T_f = 1$. In this case, $\mathcal{A}(1;\tau_0,1) = 1000$ and $\mathcal{A}(2;\tau_0,1) = 100$, leading to an amount to sequences in the order of 2^{1000}. Note that using a smaller value of $T_f > 0$ such that the amount of sequences to be evaluated is feasible, may lead to low quality results when extrapolated outside the window $[\tau_0,\tau_0+T_f]$ or even an unstable behaviour. To overcome this problems we propose a solution which is based in the following concept:

Definition 3: (Latency h-schedules and periodic extensions) Let p' be a latency h-schedule defined as the finite sequence $\{p'_\ell\}_{\ell=0}^{h-1}$. Moreover, let the periodic extension of p' be the latency schedule p defined as $p_\ell = p'_\ell$ for $0 \le \ell < h$ and $p_{\ell+h} = p_\ell$ for $\ell \in \mathbb{N}_0$.

Hence, we aim to obtain a sub-optimal solution to Problem 1 by looking to a small amount (feasible to evaluate) of sequences, namely in the form of h-schedules. Consider Algorithm 1 which by means of a number h looks for all h-schedules and computes the cost of their periodic extension by using Theorem 1. Then, returns the best h-schedule.

Algorithm 1: localSchedule$(h,\mathcal{I}_0)$

Result: $\{p^*,\bar{\mathbf{x}}^*,P^*\}$
$J^* = \infty$
for *all h-schedules* p' **do**
  $p \leftarrow$ the periodic extension of p'
  Compute $\{\bar{\mathbf{x}},P\}$ using Lemma 2 for p
  $J_p \leftarrow \mathcal{J}(p;\mathcal{I}_0,T_f)$ using Theorem 1
  if $J_p \le J^*$ **then**
    $p^* \leftarrow p'$
    $J^* \leftarrow J_p$
    $\{\bar{\mathbf{x}}^*,P^*\} \leftarrow \{\bar{\mathbf{x}},P\}$
  end
end

The previous sub-optimal solution for $[\tau_0,\tau_0+T_f]$ motivates a Moving Horizon Strategy (MHS) for windows $[\tau_\ell,\tau_\ell+T_f]$ as described in Algorithm 2. This strategy consists to choose online the next set of latencies when a sampling event occurs. To be precise, after taking sensor measurements, localSchedule$(h,\mathcal{I}_0)$ is called to obtain an h-calendar which is used for $h_M \le h$ sampling steps before changing decision again.

Algorithm 2: OnlineMHS$(S,h,h_M,\mathcal{I}_0)$

$\tau_s \leftarrow \tau_0$
$k \leftarrow 1$
At sampling event $t = \tau_s$ **do**
  Take measurements using sensors
  if $mod(k-1,h_M)=0$ **then**
    $\mathcal{I} \leftarrow \{\tau_s, p_{k-2}, p_{k-3}, \hat{\mathbf{x}}[k-2],$
    $\hat{\mathbf{x}}[k-3], P[k-2], P[k-3]\}$
    $\{p^*,\bar{\mathbf{x}}^*,P^*\} \leftarrow$localSchedule$(h,\mathcal{I})$
  end
  $p_{k-1} \leftarrow p^*_{\text{mod}(k-1,h_M)}$
  Use method p_{k-1} to compute $\hat{\mathbf{x}}[k-1]$
   from sensors information
  Predict $\hat{\mathbf{x}}[k|k-1]$ using (3)
  $\mathbf{u}[k] \leftarrow L\hat{\mathbf{x}}[k|k-1]$
  Apply $\mathbf{u}[k]$ to the system
  $\tau_s \leftarrow \tau_s + \Delta^{p_{k-1}}$
  $k \leftarrow k+1$
end

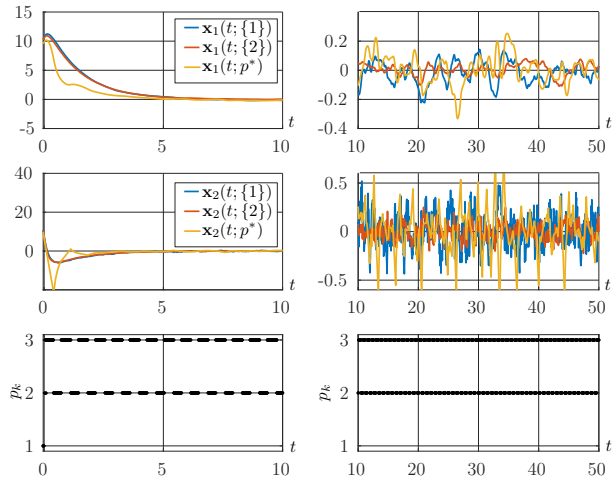

Fig. 2. Result of Example 2. $\mathbf{x}(t;\{1\}), \mathbf{x}(t;\{2\})$ and $\mathbf{x}(t;\{3,2,2,2,2\})$ stands for the realizations of (1) for the periodic extensions of $\{1\}$, $\{2\}$ and $\{3,2,2,2,2\}$ respectively. Left figures shows the behaviour for $t \in [0,10]$ and right figures shows the same trajectories for $t \in [10,50]$.

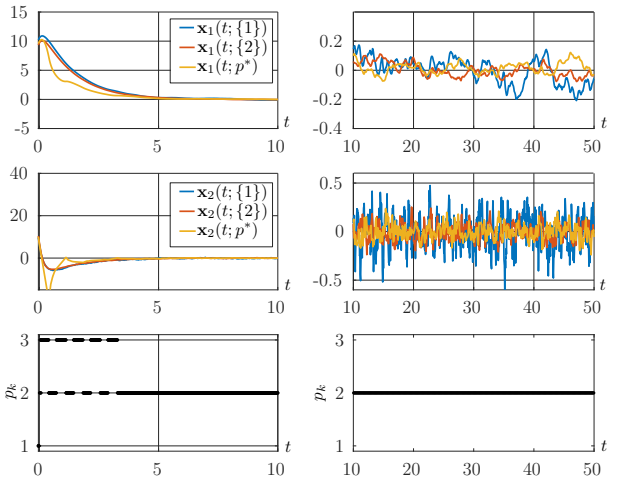

Fig. 3. Result of Example 3. $\mathbf{x}(t;\{1\}), \mathbf{x}(t;\{2\})$ and $\mathbf{x}(t;p^*)$ stands for to the realizations of (1) for the periodic extensions of $\{1\}$, $\{2\}$ and p^* respectively. Left figures shows the behaviour for $t \in [0,10]$ and right figures shows the same trajectories for $t \in [10,50]$.

V. Simulation Experiments

In this section we give some simulation examples using the system described by:

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \; B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \; L = \begin{bmatrix} -3.16, -5.13 \end{bmatrix}$$

which corresponds to a double integrator system, for which it can be verified that $A + BL$ satisfies Assumption 2. Moreover, consider that there are 3 possible estimation methods $\{1,2,3\}$ with latencies $\Delta^1 = 0.04$, $\Delta^2 = 0.08$ and $\Delta^1 = 0.4$, and covariances $\Sigma^1 = \text{diag}([0.1, 0.2])$, $\Sigma^2 = \text{diag}([0.01, 0.02])$ and $\Sigma^1 = \text{diag}([0.001, 0.002])$. Note that using estimation method $\{3\}$ results in an unstable system since Δ^3 is too big. For the cost parameters, consider $Q = \text{diag}([1, 0.1])$, $\lambda_x = 1$, $\lambda_\delta = 0.1$ and $T_f = 50$. Moreover, in the following examples $\mathbf{x}(t;p) = [\mathbf{x}_1(t;p), \mathbf{x}_2(t;p)]^T$ represent a realization of (1) for an h-schedule p.

Example 2: First, we show an example of the output of Algorithm 1. Consider h-schedules of $h = 5$. Under these parameters the output of `localSchedule(5,`$\mathcal{I}_0$`)` is the sequence $\{3,2,2,2,2\}$. Fig. 2 shows a realization of (1) for this h-schedule as well as the realizations for h-schedules $\{1\}$ and $\{2\}$. The expression of the cost function in (2) was evaluated on the interval $[0,50]$ for the resulting realization obtaining $68.26, 121.25, 116.31$ and 7.9×10^6 for $\{3,2,2,2,2\}$, $\{1\}, \{2\}$ and $\{3\}$ respectively. Moreover, $\{3,2,2,2,2\}$, $\{1\}, \{2\}$ and $\{3\}$ generated $345, 1250, 625$ and 125 sampling events in the interval $[0,50]$ respectively. Note that the transient of $\mathbf{x}(t;\{3,2,2,2,2\})$ is faster than in the periodic cases. However, the steady state of $\mathbf{x}(t;\{3,2,2,2,2\})$ is not the best, since $\mathbf{x}(t;\{2\})$ has a better response.

Now, the following example shows the behaviour of Algorithm 2 running online over a realization of system (1).

Example 3: Consider Algorithm 2 with $h = 5$ and $h_M = 3$ running online while obtaining a realization of (1). Fig. 3 shows a realization of (1) for the resulting schedule as well as the realizations for h-schedules $\{1\}$ and $\{2\}$. The expression of the cost function in (2) was evaluated on the interval $[0,50]$ for this realizations obtaining $61.53, 123.09, 119.37$ and 1.17×10^6 for $\{3,2,2,2,2\}$, $\{1\}, \{2\}$ and $\{3\}$ respectively. Moreover, p^* generated 606 sampling events in the interval $[0,50]$. Note that in this example, as well as in Example 2, the transient of $\mathbf{x}(t;p^*)$ is faster than with periodic sampling. However, the steady state is the same as of using $\{2\}$, which is the best choice.

In the following example we use `localSchedule`$(h, \mathcal{I}_0)$ for different initial conditions, giving rise to patterns for resulting schedules with respect to data in $\mathcal{I}_0$.

Example 4: Consider h-schedules with $h = 5$. Fig. 4 shows the h-schedule solution of `localSchedule`$(5, \mathcal{I}_0)$ by sweeping along $\bar{\mathbf{x}}_1[-2] = \bar{\mathbf{x}}_1[-1]$ and $\bar{\mathbf{x}}_2[-2] = \bar{\mathbf{x}}_2[-1]$ with 50 values for each in the interval $[-10, 10]$. Note that, for the sake of simplicity, not all solutions are shown but just the first two elements of them.

VI. Discussion and Future Work

From Algorithm 1 it can be deduced that increasing the value of h leads to less conservative results. To clarify this, consider 2 methods, one for which Δ^1 as constant sampling interval in (1) is stable and other in which Δ^2 lead to instability. Therefore, using $h = 1$ will result in choosing $\{1\}$ always, never giving opportunity to the unstable mode. On the other hand, for large enough h, Algorithm 2 might find useful to use $\{2\}$ in the resulting sequence, hence, obtaining a less conservative result. However, as described in Section IV, the computation cost increases quickly with respect to h. Despite this, as shown in Example 4, the results obtained from Algorithm 1 appear to be organized in well defined classes with respect to the information contained in $\mathcal{I}_0$. This motivates the usage of a classifier trained on solutions obtained from Algorithm 1. Note that, since in general the

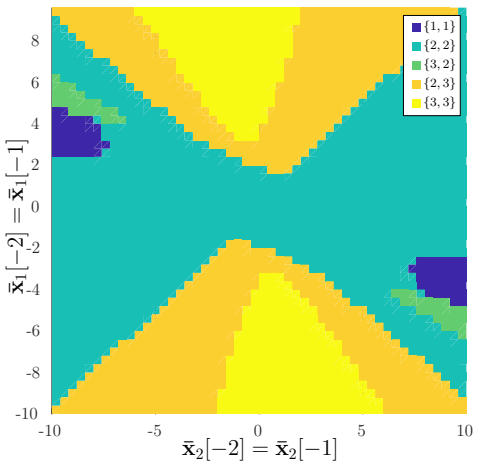

Fig. 4. Result of Example 4. The solution of `localSchedule`$(5, \mathcal{I}_0)$ grouped by the first 2 elements of it. Different results are shown by sweeping along $\bar{\mathbf{x}}_1[-2] = \bar{\mathbf{x}}_1[-1]$ and $\bar{\mathbf{x}}_2[-2] = \bar{\mathbf{x}}_2[-1]$ with 50 values for each in the interval $[-10, 10]$,

size of the data in $\mathcal{I}_0$ is low, then efficient classifiers exists in the literature realized by means of small neural networks which can be deployed in low resource systems [10], [11].

In the form that Algorithms 1-2 are presented here, the stability of the resulting realization of the system (1) now depends on the horizon T_f and the weights $[\lambda_x, \lambda_\delta]$. In fact, consider small T_f. In this case, an unstable mode might not appear harmful to the system and can be chosen by Algorithm 1 leading to instability. On the other hand, consider $[\lambda_x, \lambda_\delta] = [0, 1]$. In this case, Algorithm 1 will choose the sequence with longer sampling intervals which might be a periodic application of an unstable mode on (1). Hence, looking only for schedules for which switching between them guarantees stability is of paramount importance for the implementation of Algorithm 1 or to use the full range of the configuration parameters of the algorithms shown here. A study of such results will be presented in future work.

VII. Conclusions

In this work, a precise problem statement of the *latency-scheduling* problem as a way to tackle the attention-precision trade-off was given. Probabilistic descriptions of the model and performance measure are given which lead to algorithms that obtain sub-optimal solutions to the problem. Interesting simulation scenarios were shown to test the effectiveness of the proposal. In particular it was shown that even if some unstable latencies are used, our method can eventually take advantage of them while preserving stable behaviour, good quality performance, and achieving low attention to the system. Furthermore, the simulation results show that the solution space for the *latency-scheduling* problem are well organized over the initial information space, which motivate efficient approximate solutions based on classifiers. Moreover, some missing issues related to the stability are discussed, leading to interesting research questions regarding this problem that can be tackled in future research.

Appendix

Theorem 2: Let $\{\mathbf{n}[k] \in \mathbb{R}^m : k \in \mathbb{N}_0\}$ be a discrete-time stochastic process which satisfies that each $\mathbf{n}[k]$ is distributed as $\mathcal{N}(0, \Sigma_k)$. Let $\mathbf{z}[k] \in \mathbb{R}^n$, $\Lambda_k \in \mathbb{R}^{n\times n}$ and $\Gamma_k \in \mathbb{R}^{n\times m}$, $\forall k \in \mathbb{N}_0$. Then, the solution of the stochastic linear difference equation

$$\mathbf{z}[k+1] = \Lambda_k \mathbf{z}[k] + \Gamma_k \mathbf{n}[k]$$

satisfies that each $\mathbf{z}[k]$ is distributed as $\mathcal{N}(\bar{\mathbf{z}}[k], P[k])$ with

$$\begin{aligned} \bar{\mathbf{z}}[k+1] &= \Lambda_k \bar{\mathbf{z}}[k] \\ P[k+1] &= \Lambda_k P[k] \Lambda_k^T + \Gamma_k \Sigma_k \Gamma_k^T \end{aligned} \tag{10}$$

given $\bar{\mathbf{z}}[0] = \mathbb{E}\{\mathbf{z}[0]\}$ and $P[0] = \text{cov}\{\mathbf{z}[0]\}$.

Proof: The proof follows directly from [12, Theorem 3.1] by noticing that its proof doesn't depend on the assumption of constant covariance for the process $\{\mathbf{n}[k] : k \in \mathbb{N}_0\}$ but rather computes $\mathbb{E}\{\mathbf{z}[k+1]\}$ and $\text{cov}\{\mathbf{z}[k+1]\}$ recursivelly for each $k \in \mathbb{N}$. This facts lead to (10) by noting that $\text{cov}\{\Gamma_k \mathbf{n}[k]\} = \Gamma_k \text{cov}\{\mathbf{n}[k]\}\Gamma_k^T$. ■

Lemma 3: Let $\mathbf{z}$ be a random variable satisfying $\mathbb{E}\{\mathbf{z}\} = \bar{\mathbf{z}}$ and $\text{cov}\{\mathbf{z}\} = P$. Then, given a positive definite matrix $Q \in \mathbb{R}^{n\times n}$, $\mathbb{E}\{\mathbf{z}^T Q \mathbf{z}\} = \bar{\mathbf{z}}^T Q \bar{\mathbf{z}} + \text{tr}\,(QP)$.

Proof: Let $\mathbf{w} = Q^{1/2}\mathbf{z}$ which satisfies $\mathbb{E}\{\mathbf{w}\} = Q^{1/2}\bar{\mathbf{z}}$. Moreover, not only $\text{tr}\,(\text{cov}\{\mathbf{w}\}) = \text{tr}\,(Q^{1/2}\text{cov}\{\mathbf{z}\}Q^{1/2}) = \text{tr}\,(Q^{1/2}PQ^{1/2}) = \text{tr}\,(QP)$, by the cyclic property of the trace, but also $\text{tr}\,(\text{cov}\{\mathbf{w}\}) = \text{tr}\,(\mathbb{E}\{(\mathbf{w}-\bar{\mathbf{w}})(\mathbf{w}-\bar{\mathbf{w}})^T\}) = \text{tr}\,(\mathbb{E}\{\mathbf{w}\mathbf{w}^T\}) - \text{tr}\,(\bar{\mathbf{w}}\bar{\mathbf{w}}^T)$ where $\text{tr}\,(\bar{\mathbf{w}}\bar{\mathbf{w}}^T) = \text{tr}\,(Q^{1/2}\bar{\mathbf{z}}\bar{\mathbf{z}}^T Q^{1/2}) = \bar{\mathbf{z}}^T Q \bar{\mathbf{z}}$. Then, $\mathbb{E}\{\mathbf{z}^T Q \mathbf{z}\} = \mathbb{E}\{\mathbf{w}^T\mathbf{w}\} = \bar{\mathbf{z}}^T Q \bar{\mathbf{z}} + \text{tr}\,(QP)$ ■

References

[1] R. Siegwart and I. Nourbakhsh, *Introduction to Autonomous Mobile Robots*, ser. A Bradford book. Bradford Book, 2004.

[2] D. Falanga, S. Kim, and D. Scaramuzza, "How fast is too fast? the role of perception latency in high-speed sense and avoid," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1884–1891, April 2019.

[3] A. Anta and P. Tabuada, "To sample or not to sample: Self-triggered control for nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 55, no. 9, pp. 2030–2042, Sep. 2010.

[4] P. Tabuada, "Event-triggered real-time scheduling of stabilizing control tasks," *IEEE Transactions on Automatic Control*, vol. 52, no. 9, pp. 1680–1685, Sep. 2007.

[5] M. Miskowicz, *Event-Based Control and Signal Processing*, ser. Embedded Systems. CRC Press, 2018.

[6] J. Araújo, A. Teixeira, E. Henriksson, and K. H. Johansson, "A down-sampled controller to reduce network usage with guaranteed closed-loop performance," in *53rd IEEE Conference on Decision and Control*, Dec 2014, pp. 6849–6856.

[7] D. Liberzon, *Switching in Systems and Control*, ser. Systems & control. Birkhauser, 2003.

[8] L. Hetel, C. Fiter, H. Omran, A. Seuret, E. Fridman, J.-P. Richard, and S. I. Niculescu, "Recent developments on the stability of systems with aperiodic sampling: An overview," *Automatica*, vol. 76, pp. 309 – 335, 2017.

[9] J. Cortes, "Discontinuous dynamical systems," *IEEE Control Systems Magazine*, vol. 28, no. 3, pp. 36–73, June 2008.

[10] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, ser. Springer Series in Statistics. New York, NY, USA: Springer New York Inc., 2001.

[11] STMicroelectronics, *STM32 solutions for Artificial Neural Networks*. [Online]. Available: https://www.st.com/content/st_com/en/stm32-ann.html

[12] K. Åström, *Introduction to Stochastic Control Theory*, ser. Mathematics in science and engineering. Academic Press, 1970.