Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 53-2 (2020) 11024–11031

From Sensor to Processing Networks: Optimal Estimation with Computation and Communication Latency ⋆

Luca Ballotta * **Luca Schenato** * **Luca Carlone** **

* *Department of Information Engineering, University of Padova, Padova, 35131, Italy (e-mail: {ballotta, schenato}@dei.unipd.it)*

** *Laboratory for Information & Decision Systems, Massachusetts Institute of Technology, Boston, 02139, USA (e-mail: lcarlone@mit.edu)*

Abstract: This paper investigates the use of a networked system (*e.g.,* swarm of robots, smart grid, sensor network) to monitor a time-varying phenomenon of interest in the presence of communication and computation latency. Recent advances in edge computing have enabled processing to be spread across the network, hence we investigate the fundamental communication-computation trade-off, arising when a sensor has to decide whether to transmit raw data (incurring communication delay) or preprocess them (incurring computational delay) in order to compute an accurate estimate of the state of the phenomenon of interest. We propose two key contributions. First, we formalize the notion of *processing network*. Contrarily to *sensor and communication networks*, where the designer is concerned with the design of a suitable communication policy, in a *processing network* one can also control when and where the computation occurs in the network. The second contribution is to provide analytical results on the optimal preprocessing delay (*i.e.,* the optimal time spent on computations at each sensor) for the case with a single sensor and multiple homogeneous sensors. Numerical results substantiate our claims that accounting for computation latencies (both at sensor and estimator side) and communication delays can largely impact the estimation accuracy.

Keywords: Networked systems, communication latency, processing latency, processing network, resource allocation, sensor fusion, edge computing, smart sensors.

1. INTRODUCTION

Networked systems are becoming an ubiquitous technology across many application domains, including city-wide air-pollution monitoring (Maag et al., 2018), smart power grids (Pasqualetti et al., 2011), swarms of mobile robots for target tracking (Li Fan et al., 2009), interconnected autonomous vehicles and self-driving cars (Shalev-Shwartz et al., 2017). Progress on communication systems, such as the development of 5G, carries the promise of further expanding the reach of these systems by enabling more effective and larger-scale deployments. At the same time, recent advances on embedded computing, from embedded GPU-CPU systems to specialized hardware (Suleiman et al., 2018), are now providing unprecedented opportunities for *edge-computing*, where sensor data are processed locally at the sensor to minimize the communication burden.

The availability of powerful embedded computers creates a nontrivial *communication-computation trade-off*: is it best to transmit raw sensor data and incur larger communication and data fusion delays at a central station, or to perform more preprocessing at the sensors and transmit more refined (less noisy and more compressed) estimates? Fig. 1 provides

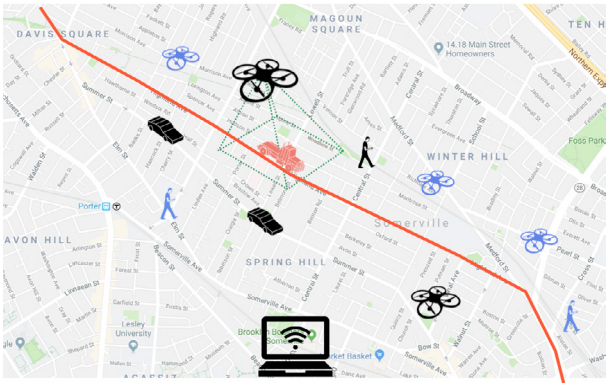

Fig. 1. Example of *processing network*: smart sensors (in blue and black) collect, process, and communicate data to track the state of a vehicle (in red) in the presence of communication and computation latency.

an example of this trade-off: the figure depicts a network of *smart sensors* (in black and blue) observing and tracking the state of a moving vehicle (in red) and transmitting data to a central fusion station (the computer at the bottom of the figure), which is in charge of monitoring the state of the red truck. The smart sensors may have heterogeneous computational resources: for instance, the large drone (in black) might have a powerful onboard GPU-CPU system, while other smart sensors (in blue, e.g., smaller drones,

⋆ This work was partially funded by the ONR RAIDER program (N00014-18-1-2828), the CARIPARO Foundation Visiting Programme "HiPeR", and the Italian Ministry of Education PRIN n. 2017NS9FEY and under the initiative "Departments of Excellence" (Law 232/2016).

10.1016/j.ifacol.2020.12.223

mobile phones) might have limited computation. Therefore, some sensors might prefer sending raw data and incur larger delays for transmission, while some other sensors might prefer preprocessing the data at the edge. These choices will impact the quality of the red vehicle estimate: larger computation and communication delays will lead to less accurate estimates, hindering the tracking task.

In this paper we investigate the communication-computation trade-off that arises in a networked system responsible for estimating the state of a time-varying phenomenon of interest in the presence of computational and communication delays. Related work in the IoT community focuses on optimizing data transmission by means of smart communication policies, with respect to estimation performance (Wu et al., 2018) or the so-called *Age of Information* (AoI) due to delays and unreliability (Zhou and Saad, 2019; Yates and Kaul, 2019). Kosta et al. (2017) and Bisdikian et al. (2013) introduce the *Value of Information of Update* (VoIU), which addresses the impact new samples have on the current state estimate. Also, the former formalize the *Cost of Update Delays* (CoUD), a non-linear function of AoI expanding such concept, which concurs in the VoIU of samples. Contrary to this line of work, we focus on monitoring a dynamical system and advocate a unified task-driven framework, where computation and communication are jointly modeled in an optimal estimation framework. In hindsight, we propose a paradigm shift from *sensor and communication networks*, in which one has to decide the best communication policy, to *processing networks*, where one also controls when and where the computation occurs. Moreover, we analyze the relation between computation/communication delays and system dynamics, while previous work mostly focuses on the channel properties.

Related work in control, cyber-physical systems, and robotics focuses on either the co-design of estimation and control in the presence of communication constraints (Borkar and Mitter, 1997; Shafieepoorfard and Raginsky, 2013), or on the design of the system's sensing and actuation (Carlone and Karaman, 2018; Summers et al., 2016). Tzoumas et al. (2018) establish a more direct connection between sensing and estimation performance, by proposing co-design approaches for sensing, estimation, and control. While these works focus on communication constraints, we attempt to explicitly model *computation delays* and understand how they impact the performance of the estimation task. In robotics, Chinchali et al. (2019) adopt a learning approach for computation offloading in cloud-robotics applications. Tsiatsis et al. (2005) seek a policy to tackle edge-computing delays within a static framework. Taami et al. (2019) characterize the performance of resource-constrained devices with cloud fog offloading (with case study on Fast Fourier Transform computation), while Imagane et al. (2016) investigate multimedia data processing with pipeline and parallel architectures. Contrary to these works, we consider the system dynamics, we explicitly model communication and computational delays, and we are concerned with the analytical derivation of optimal computation policies for estimation.

We propose the following contributions. First, we formalize the notion of processing network and provide a model which is amenable for analysis (Section 2). The networked system is modeled as a set of smart sensors in charge of estimating the state of a dynamical system in the presence of communication and computation latency. We assume that edge devices run so-called *anytime algorithms*, *i.e.,* the quality of their estimates improves with longer runtime. The key idea is to capture the impact of the preprocessing at each sensor using a processing-dependent measurement noise, such that more processing leads to more refined measurements. Second, we derive fundamental limits for such model: we prove that in two instantiations of the model there is an optimal choice for the amount of preprocessing done at each sensor which can be computed analytically. In particular, Sections 3–4 consider the continuous-time case with a single sensor and provide closed-form expressions for the optimal computational delay, while Section 5 generalizes the setup to multiple homogeneous sensors. A discussion of potential extensions to heterogeneous sensors and discrete-time systems is briefly presented in Section 6, while we refer the interested reader to the preprint Ballotta et al. (2019) for a more comprehensive discussion. Conclusions are drawn in Section 7.

2. ESTIMATION IN PROCESSING NETWORKS: PROBLEM FORMULATION

A *processing network* is a set of interconnected *smart sensors* that collect sensor data and leverage onboard computation to locally preprocess the data before communicating it to a central fusion center. The goal of the network is to obtain an accurate estimate of the state of a time-varying phenomenon observed by the sensors, in the face of communication and computation latencies.

2.1 Anatomy of a Processing Network

Dynamical system: We consider a processing network monitoring a time-varying phenomenon described by the following linear time-invariant (LTI) stochastic model:

$$dx_t = a\,x_t dt + dw_t \tag{1}$$

where $x_t \in \mathbb{R}$ is the to-be-estimated state of the system at time t, $a \in \mathbb{R}$ is a constant describing the system dynamics, and $w_t \in \mathbb{R}$ represents process noise. We focus on the scalar system (1) which can be analyzed analytically and postpone the discussion on the multi-variate case to Section 6.2.

Smart sensors: The processing network includes N smart sensors, $\mathcal{N} = \{1, ..., N\}$. After acquiring data, each sensor may refine raw measurements via some local preprocessing. For instance, in the robotics application of Fig. 1, each robot is a smart sensor that may process raw data (e.g., images) to obtain local measurements of the state (e.g., the tracked vehicle location in Fig. 1). Depending on the time and computational resources, the robot may use more sophisticated algorithms (or a larger number of visual features (Hartley and Zisserman, 2004)) to obtain more accurate measurements. More generally, the use of *anytime algorithms* (Zilberstein, 1996) at each sensor entails a trade-off, where the more time is spent on preprocessing, the more accurate is the measurement by the sensor. We capture the dependence of the preprocessing time on the refined measurements through the following model:

$$z_t(\tau_p) = Cx_t + v_t(\tau_p),\;\; z_t(\tau_p) = \left[z_t^{(1)}(\tau_{p,1})\;...\;z_t^{(N)}(\tau_{p,N})\right]^T \tag{2}$$

where $z_t^{(i)}$ is the measurement collected at time t by the i-th sensor, $\tau_{p,i}$ is the *preprocessing delay* associated with

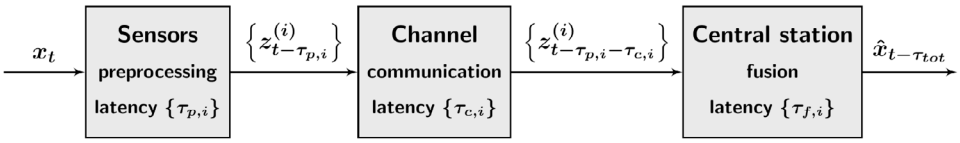

Fig. 2. Block diagram of the processing network with latency contributions by preprocessing, communication, and fusion.

the i-th sensor, and v_t is white noise; $\tau_p \doteq \{\tau_{p,i}\}_{i\in\mathcal{N}}$, and z_t contain the delays and measurements from all sensors. In order to capture the anytime nature of the sensor preprocessing, we model the intensity of the white noise v_t as a decreasing function of τ_p, see Sections 3 and 3.2.

Communication: The sensors send preprocessed data to the central station for data fusion. To simplify the mathematical analysis, we assume what follows.

Assumption 1. (Reliable channel). Packet loss and channel erasure probabilities are equal to zero.

Assumption 2. (Unconstrained channel capacity). All sensors can transmit "in parallel".

These assumptions are quite strong in practice, but they are needed for a tractable analytical approach. Future work may include more realistic communication models. In Ballotta et al. (2019), channel capacity is addressed.

Given limited bandwidth, also data transmission induces a *communication delay* $\tau_{c,i}$ for each sensor i. We consider two models for $\tau_{c,i}$ as a function of $\tau_{p,i}$:

- *constant* $\tau_{c,i}$*:* the transmitted packet length/number is fixed and does not depend on the preprocessing; in this case the communication delay is a constant, irrespective of the preprocessing delay $\tau_{p,i}$.
- *decreasing* $\tau_{c,i}$*:* in this case, sensor preprocessing *compresses* the measurements, such that a longer preprocessing leads to less packets to transmit.

These models are used in Sections 4.1 and 4.2, respectively.

Fusion center: The central station is in charge of fusing all sensor data to compute a state estimate. We assume that $\mathcal{Z}_t(\tau_p) = \{z^{(i)}_{s_i}(\tau_{p,i}), s_i \in [t_0, t-\tau_{p,i}-\tau_{c,i}]\}_{i\in\mathcal{N}}$ is the dataset available at time t (starting from an initial time t_0). Fusion adds further latency, namely the *fusion delay* $\tau_{f,tot}$, which is the sum of the delays $\tau_{f,i}$ required to process the data stream from each sensor i. In particular, as above, we assume that either $\tau_{f,i}$ is constant, or it decreases with the preprocessing delay $\tau_{p,i}$ (intuitively, the more preprocessing is done at the sensor, the less effort is needed for fusion). Fig. 2 gives an insight on the processing network with the different latency contributions - by sensor preprocessing, communication, and central station fusion.

2.2 Optimal Estimation in Processing Networks

While the sensor data might be received and fused with some (computation and communication) delay, we are interested in obtaining an accurate state estimate at the current time t; this entails fusing sensor information $\mathcal{Z}_t(\tau_p)$ (partially outdated, due to the computation and communication delays) with the open-loop system prediction in (1). This raises a nontrivial communication-computation trade-off: is it best to transmit raw sensor data and incur larger communication and fusion delays, or to perform more preprocessing at the edge and transmit more refined (less noisy and more compressed) estimates? For instance, consider again Fig. 1 where robots compute local estimations from images. Each extracted feature both enhances sensor-side accuracy and possibly reduces communication and fusion delays. However, feature extraction comes with preprocessing (edge computation) delay. A trade-off emerges: on one hand, many features cause a long prediction; on the other hand, few provide a poor estimation. An *optimal estimation* policy would decide the preprocessing at each sensor in a way to maximize the final estimation accuracy.

Problem formulation: In general, one may wish to optimize the estimation performance at all times, *i.e.*, as for Mean Squared Error (MSE) estimators, find $\arg\min_{\tau_p\in\mathbb{R}_+^N} \operatorname{var}(x_t - \hat{x}_t(\tau_p))$, where $\hat{x}_t(\tau_p) \doteq g(\mathcal{Z}_t(\tau_p))$ is a state estimator. However, such problem comes with the nuisance of time variance. Instead, we resort to its time-invariant steady-state counterpart by exploiting communication reliability (Assumption 1).

Problem 1. Given the system (1) with sensor set $\mathcal{N}$ and measurement model (2), find the optimal preprocessing delays $\tau_p = \{\tau_{p,i}\}_{i\in\mathcal{N}}$ that minimize the steady-state estimation error variance:

$$\underset{\tau_{p,i}\,\in\,\mathbb{R}_+,\, i\,\in\,\mathcal{N}}{\arg\min} \quad p_{\infty|\infty-\tau_{tot}}(\tau_p) \tag{3}$$

where the total delay is

$$\tau_{tot} \doteq \underbrace{\min_{i\in\mathcal{N}}(\tau_{p,i}+\tau_{c,i})}_{\doteq\,\tau_s} + \underbrace{\sum_{i\in\mathcal{N}}\tau_{f,i}}_{\doteq\tau_{f,tot}} \tag{4}$$

and $p_{\infty|\infty-\tau_{tot}}(\tau_p) \doteq \lim_{t\to+\infty} \operatorname{var}(x_t - \hat{x}_t(\tau_p))$ is the steady-state estimation error variance. τ_{tot} accounts for the fact that, due to delays, the steady-state estimate relies on partially outdated measurements: τ_s is the time it takes to collect data from all sensors (including the freshest available), while $\tau_{f,tot}$ is the time it takes to fuse them.

We start by analyzing the single-sensor case to gain some intuition on Problem 1. In the following, to keep notation more readable, we drop the subscripts from the preprocessing delay $\tau_{p,i}$ and refer to it as τ.

3. SINGLE SENSOR: PREPROCESSING DELAY τ

The goal of this section is twofold: (i) to provide a closed-form expression for $p_{\infty|\infty-\tau_{tot}}(\tau)$ for the case in which we have a single sensor ($N = 1$) and we neglect communication and fusion delays ($\tau_{tot} = \tau$), and (ii) to compute the optimal preprocessing delay that solves Problem 1. Towards this goal, we use a Kalman filter for state estimation, which is an MSE estimator for linear Gaussian systems. Moreover, we assume $\operatorname{var}(v_t) \doteq \sigma_v^2(\tau)$ is inversely proportional to preprocessing delay. This choice is motivated by the observation that the variance of least squares estimation is inversely proportional to the number of independent data

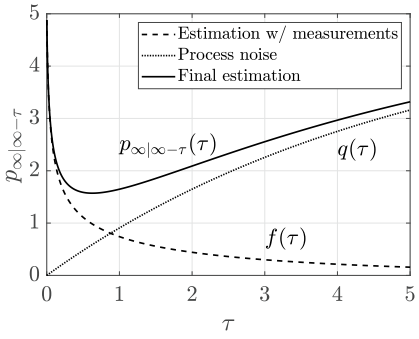

Fig. 3. Visual representation of $p_{\infty|\infty-\tau}(\tau)$, with contributions due to estimation $f(\tau)$ and process noise $q(\tau)$.

point (e.g., features extracted in an image). Other models are discussed in Section 3.2.

In the single-sensor setup, we can compute the steady-state error variance $p_{\infty|\infty-\tau}(\tau)$ in closed form and derive an analytical solution for Problem 1.

Theorem 2. (Optimal preprocessing, single sensor). Consider the LTI stochastic scalar system

$$\begin{cases} dx_t = ax_t dt + dw_t \\ z_t(\tau) = x_t + v_t(\tau) \end{cases} \tag{5}$$

with state matrix $a \in \mathbb{R}$, process noise $w_t \sim (0, \sigma_w^2)$ with $\sigma_w^2 > 0$, measurement noise $v_t(\tau) \sim (0, \sigma_v^2(\tau))$ with

$$\sigma_v^2(\tau) = \frac{b}{\tau} \qquad b > 0 \tag{6}$$

and initial condition $x_{t_0} \sim (\mu_0, p_0)$, $p_0 \geq 0$. Assume $\hat{x}_t(\tau)$ is the Kalman filter estimate at time t given measurements collected till time $t - \tau$. Then, the steady-state error variance $p_{\infty|\infty-\tau}(\tau)$ has the following expression:

$$p_{\infty|\infty-\tau}(\tau) = \underbrace{\frac{b\mathrm{e}^{2a\tau}}{\tau}\left(a + \sqrt{a^2 + \sigma_w^2 \frac{1}{b}\tau}\right)}_{f(\tau)} + \underbrace{\frac{\sigma_w^2}{2a}\left(\mathrm{e}^{2a\tau} - 1\right)}_{q(\tau)}$$

with limits

$$\lim_{\tau\to 0^+} p_{\infty|\infty-\tau}(\tau) = \lim_{\tau\to+\infty} p_{\infty|\infty-\tau}(\tau) = \begin{cases} +\infty, & a \geq 0 \\ \dfrac{\sigma_w^2}{2|a|}, & a < 0 \end{cases} \tag{7}$$

Moreover, $p_{\infty|\infty-\tau}(\tau)$ has a unique point of global minimum $\tau_{opt} > 0$ that satisfies:

$$\frac{\sigma_w^2}{b}\tau_{opt}^3 = -a^2\tau_{opt}^2 + \frac{1}{4} \tag{8}$$

Proof. See Appendix A.

Fig. 3 illustrates the cost function of Theorem 2, together with the contributions due to projecting in open-loop the measurement-based estimation and the process noise ($f(\tau)$ and $q(\tau)$, respectively).

3.1 Parameter dependence of optimal delay

Eq. (8) provides a characterization for the optimal preprocessing delay τ_{opt}. Here we discuss how τ_{opt} behaves as a function of each system's parameter. Notice that b and σ_w^2 do not affect τ_{opt} independently, as they appear in the same coefficient: therefore, we can focus on their ratio $s := \sigma_w^2/b$. Also, this suggests that what really matters is the relative intensity between the process noise and the uncertainty reduction due to preprocessing.

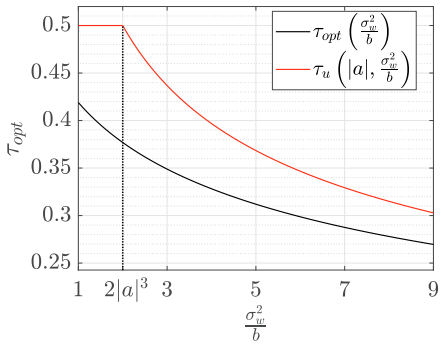

Fig. 4. Optimal delay τ_{opt} as a function of $s = \sigma_w^2/b$ with $a = 1$, and upper bound τ_u as per (9).

Proposition 3. τ_{opt} is strictly decreasing with s and a^2.

Proof. See Appendix B.

On one hand, Proposition 3 states that it is more convenient to choose small preprocessing delays for "unpredictable systems", characterized by fast dynamics or large process noise. On the other hand, if the sensor noise is large, it is convenient to perform further preprocessing, which explains why τ_{opt} grows with b.

The proof of Proposition 3 also yields the following upper bound, which may turn useful with uncertain models.

Corollary 4. (Upper bound for τ_{opt})

$$\tau_{opt} \leq \tau_u\left(|a|, \frac{\sigma_w^2}{b}\right) := \begin{cases} \dfrac{1}{2|a|}, & |a| > \sqrt[3]{\dfrac{\sigma_w^2}{2b}} \\ \sqrt[3]{\dfrac{b}{4\sigma_w^2}}, & \text{otherwise} \end{cases} \tag{9}$$

Fig. 4 shows how τ_{opt} varies with s, together with the upper bound in (9). The dependence on a^2 is qualitatively similar and omitted for space reasons.

3.2 Alternative preprocessing models

Here we consider two different models for the relation between the measurement variance and the preprocessing delay, which can be used in place of (6). The models involve a coefficient $\gamma > 0$ that can be understood as a convergence rate of an anytime algorithm. The following case generalizes model (6) accounting for non-ideality of preprocessing algorithms (as dependent samples).

Corollary 5. (Non-ideal preprocessing). Given system (5) and hypotheses as per Theorem 2 with

$$\sigma_v^2(\tau) = \frac{b}{\tau^\gamma} \qquad \gamma > 0$$

the steady-state error variance $p_{\infty|\infty-\tau}(\tau)$ has a unique global minimum $\tau_{opt} > 0$.

Proof. It can be seen that limits (7) hold and $p_{\infty|\infty-\tau}(\tau)$ is strictly quasi-convex on $\mathbb{R}_+$ (*e.g.,* via graphical analysis).

The second model comes into play with anytime algorithms with exponential convergence, as in Rudolph (2013).

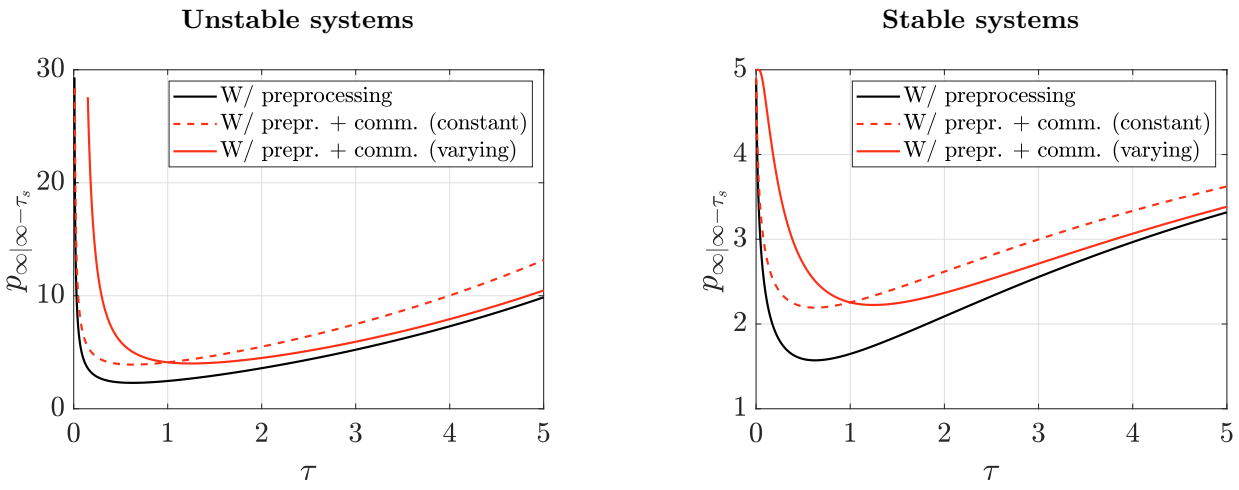

Fig. 5. Steady-state error variance $p_{\infty|\infty-\tau_s}(\tau)$ with $\sigma_w^2 = b = 1$, $a = 0.1$ (left) and $a = -0.1$ (right). Black line: no communication delay ($\tau_s = \tau$). Dashed red line: constant communication delay ($\tau_c = 1$). Solid red line: τ-varying communication delay ($c = 1$).

Corollary 6. (Anytime algorithms). Given system (5) and hypotheses as per Theorem 2 with

$$\sigma_v^2(\tau) = b\mathrm{e}^{-\gamma\tau} \qquad \gamma > 0$$

the steady-state error variance $p_{\infty|\infty-\tau}(\tau)$ has a unique global minimum $\tau_{opt} > 0$ if and only if

$$\gamma > 2\sqrt{\frac{\sigma_w^2}{b} + a^2} \tag{10}$$

Proof. In this case, τ_{opt} can be computed analytically. Condition (10) is required to make τ_{opt} positive.

Remark 7. (Phase transition). The algorithms whose convergence rate is too slow with respect to the system dynamics are discarded by condition (10): if the latter does not hold, $\tau_{opt} = 0$, *i.e.,* transmitting raw measurements is the optimal choice at sensor side.

4. SINGLE SENSOR: PREPROCESSING AND COMMUNICATION DELAYS τ, τ_C

In this section we add the communication delay, according to the two models mentioned in Section 2. The prediction step therefore stretches to the sensor delay τ_s (cf. (4)).

4.1 Constant communication delay

In this case, the communication delay τ_c is constant (*i.e.,* the preprocessing does not imply data compression): in particular, the transmitted packet number/length is independent on the time spent for preprocessing, which only affects the measurement noise variance. This situation may occur whenever the sensors send quantities whose dimension only depends on the system/algorithms, such as local state estimates. Being the communication delay constant, it does not impact the optimization with respect to the preprocessing: the steady-state variance $p_{\infty|\infty-\tau_s}(\tau)$ is simply multiplied by the coefficient $\mathrm{e}^{2a\tau_c}$ due to the longer open-loop prediction induced by τ_c. Therefore, the optimal delay is again τ_{opt} as per Theorem 2. The dependencies studied in Sec. 3.1 still hold.

4.2 Computation-dependent communication delay

We now turn to the case where the preprocessing also performs data compression, leading to a τ-varying communication delay which is modelled as:

$$\tau_c(\tau) = \frac{c}{\tau} \quad c > 0 \tag{11}$$

with known c. We have the following result.

Theorem 8. (Optimal preprocessing and communication). Given system (5) with measurement noise variance $\sigma_v^2(\tau)$ as per (6), and communication delay $\tau_c(\tau)$ as per (11), the steady-state error variance has expression

$$p_{\infty|\infty-\tau_s}(\tau) = \frac{b\mathrm{e}^{2a\tau_s}}{\tau}\left(a + \sqrt{a^2 + \frac{\sigma_w^2}{b}\tau}\right) + \frac{\sigma_w^2}{2a}\left(\mathrm{e}^{2a\tau_s} - 1\right)$$

with $\tau_s = \tau + {}^{c}\!/_{\tau}$. Moreover, $p_{\infty|\infty-\tau_s}(\tau)$ admits limits as per (7), and has a unique point of global minimum $\tau_{opt} > 0$.

Proof. See Appendix C.

Fig. 5 compares the steady-state error variance with no communication delay (but with preprocessing delay, in black), and with communication delay (in red, dashed for constant and solid for τ-varying delays) for an unstable and an asymptotically stable systems. Notice that the steepness of the black curve decreasing portion suggests that it is preferable to round τ_{opt} in excess, if needed, as a lower approximation likely worsens performance. The first communication-delay model (constant τ_c) shifts upward and slightly sharpens the curve, while the second smooths it. In this case, monotonicity of τ_{opt} as in Section 3.1 cannot be guaranteed. Also, notice that the red curves cross: the model with constant/no compression is outperformed by the τ-varying one if the preprocessing is longer than a minimum threshold.

Remark 9. While model (11) is mainly used for mathematical convenience, in a real setup the compression function should be learned or estimated from data.

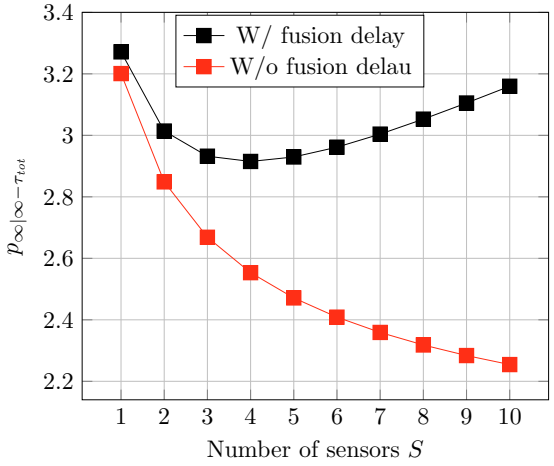

Fig. 6. Variance $p_{\infty|\infty-\tau_{tot}}(S)$ with $a = -1$, $\sigma_w^2 = 10$, $b = \tau = 0.1$, $\tau_c = 0.1$, $\tau_f = 0.02$ (black) and $\tau_f = 0$ (red, no fusion delay).

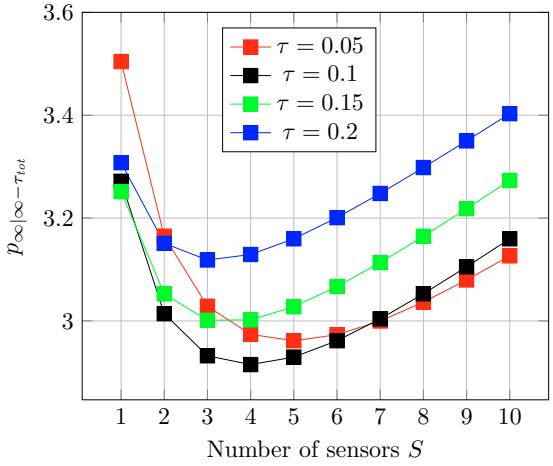

Fig. 7. Variance $p_{\infty|\infty-\tau_{tot}}(S)$ with $a = -1$, $\sigma_w^2 = 10$, $b = \tau_c = 0.1$, $\tau_f = 0.02$, $\tau \in \{0.05, 0.1, 0.15, 0.2\}$.

5. MULTIPLE SENSORS: PREPROCESSING, COMMUNICATION, AND FUSION DELAYS

We now consider the multi-sensor case and complete our framework by adding the fusion delay. We model the latter similar to the communication delay: $\tau_{f,i}$ is either assumed to be constant, or we assume $\tau_{f,i}(\tau_i) = f_i/\tau_i$, where $f_i > 0$ is a known constant. We consider for simplicity N identical independent sensors, each with the same delays τ (preprocessing), $\tau_c(\tau)$ (communication) and $\tau_f(\tau)$ (fusion), the latter two being constant or varying. The overall delay (4) becomes

$$\tau_{tot} = \underbrace{\tau + \tau_c(\tau)}_{\tau_s} + \underbrace{\tau_f(\tau) N}_{\tau_{f,tot}} \tag{12}$$

and the network measurements in (2) are then modeled as

$$z_t(\tau) = [1 \ \dots \ 1]^T x_t + v_t(\tau) \qquad R(\tau) = \frac{b}{\tau} I_N \tag{13}$$

From the least squares framework, it is well known that such system yields an overall variance reduction for $z_t(\tau)$ which is linear with the number of samples. Alternatively, the homogeneous N-sensor network can be seen (from the standpoint of the estimation performance) as a single sensor with processing noise variance $\sigma_v^2(\tau)$ reduced by a factor N with respect to each sensor in (13), and total delay (12). Then, the optimal computational delay for such virtual single sensor also maximizes the performance for the network (13).

Remark 10. The advantage of multi-sensor networks is reducing the measurement-noise variance for each sensor, yielding more accurate state estimation.

Remark 11. A common wisdom in estimation theory is that adding more sensors always yields better estimates (possibly with performance saturation). We show that, if fusion time has to be considered, the optimal solution is adding sensors up to a certain amount, since the cost of processing more overtakes the sample variance reduction.

If $p_{\infty|\infty-\tau_{tot}}$ is seen as a discrete function of the number of sensors S ($S \leq N$) with fixed τ, one can also ask if there is an *optimal sensor quantity* S_{opt}, corresponding to $\arg\min p_{\infty|\infty-\tau_{tot}}(S)$ [1]. Fig. 6 shows the performance behaviour according to the number of sensors, with fixed τ. Notice that neglecting the fusion delay may yield important performance drops (about 12% with S_{opt} sensors and 32% with all N). Then, Problem 1 can be extended to decide on the optimal number of sensors to be used for estimation.

Problem 12. (Homogeneous network). Given system (5) with N identical sensors and measurement model (13), find the optimal sensor amount S and preprocessing delay τ that minimize the steady-state error variance:

$$\begin{array}{c} \arg\min \\ S \in \{1, \dots, N\} \\ \tau \in \mathbb{R}_+ \end{array} \quad p_{\infty|\infty-\tau_{tot}}(S, \tau)$$

Proving uniqueness of the solution in this case in nontrivial, due to both the discrete domain of the cost function (S must be natural) and the difficulty of proving quasi-convexity (or a suitably equivalent characterization). However, simulations results suggest that Problem 12 admits a unique solution: Fig. 7 shows $p_{\infty|\infty-\tau_{tot}}(S)$ corresponding to different values of τ with constant delay. The τ-varying model is similar and omitted for space reasons.

6. EXTENSIONS TO HETEROGENEOUS, MULTI-VARIATE, AND DISCRETE-TIME SYSTEMS

This section presents extensions to heterogeneous networks and discrete-time systems, as well as future work directions. The interested reader can find more details in Ballotta et al. (2019), where some realistic scenarios are analyzed.

6.1 Sensor selection in heterogeneous networks

In general, sensors in the processing network might have different resources, resulting in different coefficients b in (6). If this is the case, computing the optimal preprocessing becomes a multivariate problem. Moreover, if sensors are heterogeneous, one also faces the choice of whether to use the data from all sensors or disregard data from some of them. Therefore, a potential generalization of Problems 1 and Problem 12 is as follows.

[1] Due to its discrete domain, $p_{\infty|\infty-\tau_{tot}}(S)$ may have two points of global minimum with $p_{\infty|\infty-\tau_{tot}}(S^*) = p_{\infty|\infty-\tau_{tot}}(S^* + 1)$.

Problem 13. (Heterogeneous network). Given system (5) with sensor set $\mathcal{N}$ and measurement model (2), find the optimal sensor subset $\mathcal{S}$ and preprocessing delays $\tau = \{\tau_i\}_{i\in\mathcal{S}}$ that minimize the steady-state error variance:

$$\begin{aligned} \underset{\mathcal{S}\subseteq\mathcal{N}}{\arg\min} \quad & p_{\infty|\infty-\tau_{tot}}(\mathcal{S},\tau) \\ \tau_i \in \mathbb{R}_+, i\in\mathcal{S} & \end{aligned}$$

Note that the combinatorial nature of the problem makes it difficult to compute exact solutions. In the extended paper Ballotta et al. (2019) we investigate this formulation with simulations, and propose approximate algorithms.

6.2 Discrete-time and multi-dimensional systems

While the continuous-time framework was instrumental to obtain insights and analytical solutions, in practical problems it is interesting to consider a discrete-time formulation due to the digital nature of involved systems and algorithms. In Ballotta et al. (2019), we extend the setup considered in this paper to more general scenarios, accounting for discrete-time and multi-dimensional states. Numerical simulations confirm that the trends observed in the scalar case also arise in such more general case: the communication-computation trade-off can be optimized by suitably selecting sensors and preprocessing delays.

6.3 Dealing with channel constraints

An interesting avenue for future research is to consider more realistic communication model, including finite bandwidth, channel capacity, unreliability, or packet loss. For instance, to model limited channel capacity, an upper bound may be imposed on the total communication delay:

$$\sum_{i\in\mathcal{S}} \tau_{c,i}(\tau_i) \leq \tau_{u,c}$$

In this way, each sensor is forced not to keep the channel busy for too long, letting all sensors transmit their data.

7. CONCLUSIONS

In this paper, we investigate optimal estimation in a processing network in the presence of communication and computational delays. We model sensor-side preprocessing as a stochastic measurement model, whose noise intensity decreases with the computational delay. Similarly, communication and fusion delays are modeled as a constant or decreasing function of computation delay, simulating data compression. For the continuous-time, scalar, single-sensor scenario, we prove that the resulting trade-off between preprocessing and computation can be optimized analytically. We further extend these results to the case of a network of homogeneous sensors, where one has also to account for the fusion delay incurred at the central station which is in charge of fusing all the sensor measurements. We conclude the paper by discussing several ongoing efforts to extend this work to the case of a multi-variate, heterogeneous processing networks, monitoring a discrete-time system.

REFERENCES

Ballotta, L., Schenato, L., and Carlone, L. (2019). Computation-Communication Trade-offs and Sensor Selection in Real-time Estimation for Processing Networks. *arXiv e-prints*, arXiv:1911.05859.

Bisdikian, C., Kaplan, L.M., and Srivastava, M.B. (2013). On the quality and value of information in sensor networks. *ACM Trans. Sen. Netw.*, 9(4), 48:1–48:26. doi: 10.1145/2489253.2489265. URL `http://doi.acm.org/10.1145/2489253.2489265`.

Borkar, V. and Mitter, S. (1997). LQG control with communication constraints. *Comm., Comp., Control, and Signal Processing*, 365–373.

Carlone, L. and Karaman, S. (2018). Attention and anticipation in fast visual-inertial navigation. *IEEE Trans. Robotics*. Arxiv preprint: 1610.03344.

Chinchali, S., Sharma, A., Harrison, J., Elhafsi, A., Kang, D., Pergament, E., Cidon, E., Katti, S., and Pavone, M. (2019). Network offloading policies for cloud robotics: a learning-based approach. *arXiv e-prints*, arXiv:1902.05703.

Hartley, R.I. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition.

Imagane, K., Kanai, K., Katto, J., and Tsuda, T. (2016). Evaluation and analysis of system latency of edge computing for multimedia data processing. In *2016 IEEE 5th Global Conference on Consumer Electronics*, 1–2. doi:10.1109/GCCE.2016.7800393.

Kosta, A., Pappas, N., Ephremides, A., and Angelakis, V. (2017). Age and value of information: Non-linear age case. In *2017 IEEE Intl. Symp. on Inf. Theory (ISIT)*, 326–330. doi:10.1109/ISIT.2017.8006543.

Li Fan, Dasgupta, P., and Ke Cheng (2009). Swarming-based mobile target following using limited-capability mobile mini-robots. In *2009 IEEE Swarm Intelligence Symposium*, 168–175. doi:10.1109/SIS.2009.4937860.

Maag, B., Zhou, Z., and Thiele, L. (2018). A survey on sensor calibration in air pollution monitoring deployments. *IEEE Internet of Things Journal*, 5(6), 4857–4870. doi: 10.1109/JIOT.2018.2853660.

Pasqualetti, F., Dörfler, F., and Bullo, F. (2011). Cyber-physical attacks in power networks: Models, fundamental limitations and monitor design. In *IEEE Conference on Decision and Control and European Control Conference*, 2195–2201.

Rudolph, G. (2013). Convergence rates of evolutionary algorithms for quadratic convex functions with rank-deficient hessian. In M. Tomassini, A. Antonioni, F. Daolio, and P. Buesser (eds.), *Adaptive and Natural Computing Algorithms*, 151–160. Springer Berlin Heidelberg, Berlin, Heidelberg.

Shafieepoorfard, E. and Raginsky, M. (2013). Rational inattention in scalar LQG control. In *IEEE Conf. on Decision and Control (CDC)*, 5733–5739.

Shalev-Shwartz, S., Shammah, S., and Shashua, A. (2017). On a formal model of safe and scalable self-driving cars. *ArXiv*, abs/1708.06374.

Suleiman, A., Zhang, Z., Carlone, L., Karaman, S., and Sze, V. (2018). Navion: A 2mW fully integrated real-time visual-inertial odometry accelerator for autonomous navigation of nano drones. *IEEE Journal of Solid-State Circuits*.

Summers, T., Cortesi, F., and Lygeros, J. (2016). On submodularity and controllability in complex dynamical networks. *IEEE Transactions on Control of Network Systems*, 3(1), 91–101.

Taami, T., Krug, S., and O'Nils, M. (2019). Experimental characterization of latency in distributed iot systems with cloud fog offloading. In *2019 15th IEEE Intl. Workshop on Factory Comm. Systems (WFCS)*, 1–4. doi:10.1109/WFCS.2019.8757960.

Tsiatsis, V., Kumar, R., and Srivastava, M.B. (2005). Computation hierarchy for in-network processing. *Mobile Networks and Applications*, 10(4), 505–518. doi:10.1007/s11036-005-1563-z. URL `https://doi.org/10.1007/s11036-005-1563-z`.

Tzoumas, V., Carlone, L., Pappas, G., and Jadbabaie, A. (2018). Sensing-constrained LQG control. In *American Control Conference*, 197–202. Milwaukee, WI. Arxiv preprint: 1709.08826.

Wu, S., Ding, K., Cheng, P., and Shi, L. (2018). Optimal Scheduling of Multiple Sensors over Lossy and Bandwidth Limited Channels. *arXiv e-prints*, arXiv:1804.05618.

Yates, R.D. and Kaul, S.K. (2019). The age of information: Real-time status updating by multiple sources. *IEEE Transactions on Information Theory*, 65(3), 1807–1827. doi:10.1109/TIT.2018.2871079.

Zhou, B. and Saad, W. (2019). Joint status sampling and updating for minimizing age of information in the internet of things. *IEEE Transactions on Communications*, 1–1. doi:10.1109/TCOMM.2019.2931538.

Zilberstein, S. (1996). Using anytime algorithms in intelligent systems. *AI Magazine*, 17(3).

Appendix A. PROOF OF THEOREM 2

By considering model (6), the steady-state Kalman error variance associated with $\hat{x}_{t-\tau}(\tau)$ (outdated estimate) is

$$p_\infty(\tau) = \frac{b}{\tau}\left(a + \sqrt{a^2 + \frac{\sigma_w^2}{b}\tau}\right) \tag{A.1}$$

The model-based open-loop predictor error has dynamics

$$d\tilde{x}_s(\tau) = a\tilde{x}_s(\tau)ds + dw_s, \quad t-\tau \le s \le t \tag{A.2}$$

Then, the error at time t is given by solving (A.2) as a Cauchy problem with initial condition $\tilde{x}_{t-\tau}(\tau)$:

$$\tilde{x}_t(\tau) = \mathrm{e}^{a\tau}\tilde{x}_{t-\tau}(\tau) + \bar{w}(\tau)$$

where $\bar{w}(\tau)$ is the stochastic integral of w_s in $[t-\tau,\, t]$. The steady-state prediction error variance is then

$$\begin{aligned} p_{\infty|\infty-\tau}(\tau) &\overset{(i)}{=} \mathrm{var}(\mathrm{e}^{a\tau}\tilde{x}_{t-\tau}) + \mathrm{var}(\bar{w}(\tau)) = \\ &= \frac{b\mathrm{e}^{2a\tau}}{\tau}\left(a + \sqrt{a^2 + \frac{\sigma_w^2}{b}\tau}\right) + \frac{\sigma_w^2}{2a}\left(\mathrm{e}^{2a\tau} - 1\right) \end{aligned}$$

where (i) is motivated by uncorrelated terms. Indeed, $\tilde{x}_{t-\tau} \in \mathrm{span}\{x_{t_0}, w_s, v_s : t_0 \le s \le t-\tau\}$, while $\bar{w}(\tau) \in \mathrm{span}\{w_s,\ t-\tau \le s \le t\}$, with w_t white noise and $w_t \perp x_{t_0}, v_s\ \forall t \ge t_0, \forall s$ by hypothesis. The only sample providing nonzero correlation is $w_{t-\tau}$, but having zero Lebesgue measure its contribution to $\bar{w}_t$ is null.

We proceed now in studying critical points of $p_{\infty|\infty-\tau}(\tau)$, since being limits (7) equal at both domain extrema and being $p_{\infty|\infty-\tau}(\tau) \in \mathcal{C}^0(\bar{\mathbb{R}}_+)$ at least one must exist. By setting $p'_{\infty|\infty-\tau}(\tau) = 0$ and rejecting $\tau = 0$, we get

$$\frac{\sigma_w^2}{b}\tau^3 + a^2\tau^2 - \frac{1}{4} = 0 \tag{A.3}$$

E. (A.3) always admits a real positive solution, in virtue of Bolzano's theorem by considering $F(\tau) := \sigma_w^2/b\,\tau^3 + a^2\tau^2 - 1/4$ and $I := [0, 1/2|a|]$. Strict quasi-convexity of $p_{\infty|\infty-\tau}(\tau)$ on $\mathbb{R}_+$ can be checked via convexity of its sublevel sets with graphical analysis. Such property guarantees that the critical point is the unique point of global minimum.

Appendix B. PROOF OF PROPOSITION 3

For this proof we are going to exploit the implicit function theorem, whose statement is recalled for convenience.

Theorem 14. (Dini's theorem). Let F be a continuously differentiable function on some open $D \subset \mathbb{R}^2$. Assume that there exists a point $(\bar{x}, \bar{y}) \in D$ such that:

- $F(\bar{x}, \bar{y}) = 0$;
- $\dfrac{\partial F}{\partial y}(\bar{x}, \bar{y}) \neq 0$.

Then, there exist two positive constant a, b and a function $f : I_{\bar{x}} := (\bar{x}-a, \bar{x}+a) \mapsto J_{\bar{y}} := (\bar{y}-b, \bar{y}+b)$ such that

$$F(x,y) = 0 \iff y = f(x) \quad \forall x \in I_{\bar{x}},\ \forall y \in J_{\bar{y}}$$

Moreover, $f \in \mathcal{C}^1(I_{\bar{x}})$ and

$$f'(x) = -\frac{F_x(x, f(x))}{F_y(x, f(x))} \quad \forall x \in I_{\bar{x}} \tag{B.1}$$

We can see the left-hand term in eq. (8) as a one-parameter function of two positive-valued variables, namely

$$F : \mathbb{R}_+ \times \mathbb{R}_+ \to \mathbb{R},\ (\pi, \tau) \mapsto F(\pi, \tau) = s\tau^3 + a^2\tau^2 - \frac{1}{4}$$

with $\pi = \{s, a^2\}$. Let us check if Dini's theorem hypotheses are satisfied: given a solution $(\bar{\pi}, \bar{\tau}_{opt})$ of eq. (8) it holds

- $F(\bar{\pi}, \bar{\tau}_{opt}) = 0$, by construction;
- $F_\tau(\bar{\pi}, \bar{\tau}_{opt}) = 3s\bar{\tau}_{opt}^2 + 2a^2\bar{\tau}_{opt} > 0$, since all variables are positive.

Then Theorem 14 applies and there exists a function $\tau(\pi)$ such that $F(\pi, \tau_{opt}) = 0 \iff \tau_{opt} = \tau(\pi)$, with π in some open neighbourhood of $\bar{\pi}$. In fact, since we did not pose constraints on $(\bar{\pi}, \bar{\tau}_{opt})$, such a function is defined for all $\pi \in \mathbb{R}_+$. The two cases for π are studied independently.

$\boldsymbol{\pi = s}$ By (B.1), the first derivative of $\tau(\pi) = \tau(s)$ is

$$\tau'(s) = -\frac{F_s(s, \tau(s))}{F_\tau(s, \tau(s))} = -\frac{\tau(s)^2}{3s\tau(s) + 2a^2} \tag{B.2}$$

We conclude that $\tau'(s) < 0\ \forall s \in \mathbb{R}_+$, namely, τ_{opt} is strictly decreasing with s.

$\boldsymbol{\pi = a^2}$ The first derivative of $\tau(\pi) = \tau(a^2)$ is

$$\tau'(a^2) = -\frac{F_{a^2}(a^2, \tau(a^2))}{F_\tau(a^2, \tau(a^2))} = -\frac{\tau(a^2)}{3s\tau(a^2) + 2a^2} \tag{B.3}$$

We conclude that $\tau'(a^2) < 0\ \forall a \in \mathbb{R}$, namely, τ_{opt} is strictly decreasing with a^2 (regardless of sign(a)).

Appendix C. PROOF OF THEOREM 8

By applying open-loop prediction to $p_\infty(\tau)$ (A.1) so as to cover the delay $\tau_s = \tau + \tau_c$, the steady-state prediction error variance takes the expression in Theorem 8. Strict quasi-convexity holds also in this case (again, this can be shown with graphical analysis). In virtue of this fact, continuity and limits (7), we conclude that the point of minimum exists unique and is different from zero.