Automatica 136 (2022) 110061

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Transmission scheduling for multi-process multi-sensor remote estimation via approximate dynamic programming[☆]

Ali Forootani [a,*], Raffaele Iervolino [b], Massimo Tipaldi [c], Subhrakanti Dey [a]

[a] Hamilton Institute, Maynooth University, Maynooth, Co. Kildare W23F2K8, Ireland
[b] Department of Electrical Engineering and Information Technology, University of Naples, Napoli 80125, Italy
[c] Department of Engineering, University of Sannio, Benevento 82100, Italy

ARTICLE INFO

Article history:
Received 15 April 2021
Received in revised form 31 August 2021
Accepted 1 November 2021
Available online 3 December 2021

Keywords:
Markov Decision Process
Approximate dynamic programming
Least Squares Temporal Difference
Wireless sensor networks
Kalman filter
Sensor scheduling

ABSTRACT

In this paper, we consider a remote estimation problem where multiple dynamical systems are observed by smart sensors, which transmit their local estimates to a remote estimator over channels prone to packet losses. Unlike previous works, we allow multiple sensors to transmit simultaneously even though they can cause interference, thanks to the multi-packet reception capability at the remote estimator. In this setting, the remote estimator can decode multiple sensor transmissions (successful packet arrivals) as long as their signal-to-interference-and-noise ratios (SINR) are above a certain threshold. In this setting, we address the problem of optimal sensor transmission scheduling by minimizing a finite horizon discounted expected estimation error covariance cost across all systems at the remote estimator, subject to an average transmission cost. While this problem can be posed as a stochastic control problem, the optimal solution requires solving a Bellman equation for a dynamic programming (DP) problem, the complexity of which scales exponentially with the number of systems being measured and their state dimensions. In this paper, we resort to a novel Least Squares Temporal Difference (LSTD) Approximate Dynamic Programming (ADP) based approach to approximating the value function. More specifically, an off-policy based LSTD approach, named in short Enhanced-Exploration Greedy LSTD (EG-LSTD), is proposed. We discuss the convergence analysis of the EG-LSTD algorithm and its implementation. A Python based program is developed to implement and analyse the different aspects of the proposed method. Simulation examples are presented to support the results of the proposed approach both for the exact DP and ADP cases.

1. Introduction

Wireless Sensor Networks (WSN) are adopted in numerous applications in remote estimation and control due to their advantage in reduced wiring, modularity, accuracy in measurements with multiple sensors, and better agility (Pezzutto, Schenato, & Dey, 2020). It may not be desirable to have all the sensors transmitting their measurements at all time instants due to the constraints on communication bandwidth and the sensor battery life. Multiple simultaneous sensor transmissions can also cause interference with each other and sensor transmissions can be lost due to collisions. Hence, the problem of defining a proper sensor scheduling policy arises, where the objective is the activation of different subsets of sensors at different time slots in order to obtain an optimal trade-off between estimation accuracy and energy use. Unlike the traditional channel multiplexing techniques such as scheduling a single sensor per resource block (e.g. per time-slot (TDMA) or per frequency slot (FDMA)), the modern wireless communication systems are equipped with decoding signals in the presence of interference using a more complex receiver that employs multi-user detection (mobile broadband communication such 3G systems), and multipacket reception (Wireless LAN) (Pezzutto et al., 2020). These receivers, by allowing multiple transmissions at the transmitters in the same resource block, make better use of the available resources, as opposed to TDMA or FDMA. While this is widely researched in the context of multi-terminal network information theory involving multiple access/broadcast/interference channels, it has not been investigated in the context of wireless control systems.

1.1. Literature review on sensor scheduling

As state estimation is crucial to feedback control systems, efficient processing of sensory data under limited resources is

[☆] The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Luca Schenato under the direction of Editor Christos G. Cassandras.

[*] Corresponding author.

E-mail addresses: Ali.forootani@mu.ie (A. Forootani), rafierv@na.it (R. Iervolino), mtipaldi@unisannio.it (M. Tipaldi), Subhra.Dey@mu.ie (S. Dey).

https://doi.org/10.1016/j.automatica.2021.110061
0005-1098/© 2021 Elsevier Ltd. All rights reserved.

important (Wu, Jia, Johansson, & Shi, 2013). The quality and the accuracy of the estimation can be improved by trading off communication bandwidth, energy budget, and transmissions power (Nourian, Leong, & Dey, 2014; Shi, Cheng, & Chen, 2011a, 2011b).

As communication is expensive in wireless sensor networks, efficient utilization of online information to reduce communication rate is another research interest, see Leong, Dey, and Quevedo (2017), Ren, Wu, Johansson, Shi, and Shi (2018) where event-triggered transmission strategies can be found. The bandwidth of wireless communication channels can be limited, and only a few sensors are allowed to be activated in each time slot since the sensors can interfere with each other. In Gupta, Chung, Hassibi, and Murray (2006), the authors analysed the performance of a stochastic sensor selection algorithm for two problems, that is to say, sensor scheduling under channel bandwidth limitation and sensor coverage in a given location. The reader can also refer to Han, Wu, Zhang, and Shi (2017), Ren, Wu, Dey, and Shi (2018) for similar studies.

Sensor scheduling problems have also been investigated in infinite time horizon frameworks, e.g., in Zhao, Zhang, Hu, Abate, and Tomlin (2014) it has been proven that, under some mild conditions, both the optimal infinite-horizon average-per-stage cost and the corresponding optimal sensor scheduling policy are independent of the covariance matrix of the initial state. More complex sensor scheduling problem scenarios can be found in literature. For instance, in Liu, Quevedo, Johansson, Vucetic, and Li (2021), remote state estimation of multiple systems over multiple Markov fading channels has been considered together with the need of guaranteeing the existence of a stabilizing sensor scheduling policy. In Leong, Ramaswamy, Quevedo, Karl, and Shi (2020), a deep Q-learning approach has been presented to deal with a multi-process multi-channel sensor scheduling problem. However, Q learning based approaches have to take into account some issues, e.g., instability and convergence problems and sampling mechanism complexity (Bertsekas, 2012).

In almost all the mentioned works, it is assumed that either a single sensor can be scheduled at each time slot, or there is no interference caused with simultaneous transmissions. There are two notable exceptions where interference is taken into account: in Gatsis, Ribeiro, and Pappas (2018), the authors proposed a channel-adaptive optimal random access scheme for remote control of multiple systems, and in Li, Chen, and Wong (2019), the authors studied the optimal power allocation for remote estimation. In Pezzutto et al. (2020), the authors have recently considered two types of interference over a wireless transmission channel with two sensors deployed for the remote state estimation of a linear time invariant dynamic system. In this work, the sensitivity of the different parameters on a single stage Dynamic Programming (DP) optimization problem has been investigated.

1.2. Scope and contribution of the paper

This work addresses the sensor scheduling problem for remote state estimation of multiple linear time-invariant Gauss–Markov processes. Each process state is measured by a smart sensor, which is able to compute the local state estimate of the process and transmit it to a remote estimator. The packet reception model accounts both for the interference due to incoming signals transmitted by other sensors and the external noise. We consider a multi-packet reception scheme based on the capture property of the wireless receiver (Zanella & Zorzi, 2012), where any sensor state estimate with a Signal-to-Interference-and-Noise Ratio (SINR) above a certain threshold can be successfully decoded by the remote estimator. In this scheme, each sensor can observe the interference due to the other sensor transmissions and the external noise.[1]

Unlike previous works, we consider here computationally efficient solutions to sensor scheduling problems for linear time invariant dynamic systems (Single Input-Single Output/ Multi Input-Multi Output), see Leong, Dey, and Quevedo (2017), Pezzutto et al. (2020). We can notice that finding an analytical solution for the sensor scheduling problems is impractical unless for special cases with conservative assumptions and small size systems (see Leong, Dey, & Quevedo, 2017; Pezzutto et al., 2020). We formulate the scheduling problem of Multiple Sensors-Multiple Processes (MSMP) over a noisy wireless transmission channel as an MDP with an error covariance discounted cost function, computed over a finite time horizon. Our MDP framework differs from the ones shown in previous works such as Leong, Dey, and Quevedo (2017), Wu, Ren, Dey, and Shi (2018) in three main aspects: the modelling assumption and the related problem formulation, the handling of the MDP scalability issues, and the proposed solution. Our main objective is to compute an appropriate sensor scheduling policy to minimize an expected cost function which depends on the state estimation error covariance of each sensor. It is natural to apply Dynamic Programming (DP) techniques (Bertsekas, 2012; Forootani, Iervolino, & Tipaldi, 2019; Forootani, Tipaldi, Ghaniee Zarch, Liuzza, & Glielmo, 2020b) to solve MSMP transmission scheduling problems modelled as MDP. However, the scalability issues of practical size MSMP problems call for the usage of ADP based techniques (Bertsekas, 2012). Recent applications of the ADP can be found in different fields, such as multi-agent robotic systems (Deng, Chen, & Belta, 2017), optimal stopping problems (Forootani, Tipaldi, Iervolino, & Dey, 2022), and resource allocation problems (Forootani, Iervolino, Tipaldi, & Neilson, 2020; Forootani, Liuzza, Tipaldi, & Glielmo, 2019).

In this paper, we approximate the optimal cost function by a compact parametric representation, which is also referred to as approximation architecture (Bertsekas, 2011; De Farias & Van Roy, 2003; Forootani et al., 2022; Geist & Pietquin, 2013). A new algorithm, named in short Enhanced-exploration Greedy Least Squares Temporal Difference (EG-LSTD), is proposed to compute such approximation. The EG-LSTD algorithm is a special type of the classical LSTD method (Tsitsiklis & Van Roy, 1997). It has been derived from the Multi-trajectories Greedy LSTD (MG-LSTD) algorithm, presented by the authors in Forootani, Tipaldi, Ghaniee Zarch, Liuzza, and Glielmo (2020a). The EG-LSTD algorithm can be regarded as an off-policy LSTD approach since the initial state of each trajectory is selected by applying a probability distribution different from the frequencies of the MDP at hand. In general, off-policy LSTD methods may not be convergent (see Tsitsiklis & Van Roy, 1997, Th. 3). Hence, inspired by Bertsekas and Yu (2009), we derive a condition for the selection of the initial states of each trajectory to guarantee the convergence of the EG-LSTD algorithm.

The main contributions of this article can be summarized as follows: (i) modelling MSMP transmission scheduling problems over a wireless packet dropping channel with SINR as an MDP; (ii) considering a stochastic discounted DP framework to solve the resulting MDP with the error covariance matrices of the state estimation filters (Kalman filters) as the cost per stage;

[1] In information theoretic context, for additive Gaussian white noise channels, the bit error probability is assumed negligible if the signal-to-noise ratio (SNR) exceeds a certain probability, typically of the order of 10^{-4} to 10^{-6} with suitable modulation and coding. Similar to the SNR case, it can be shown that, with a suitable choice of modulation and error control coding, the packet loss probability can be made negligibly small when the SINR exceeds the required threshold (Perez-Neira & Campalans, 2010).

(iii) proposing a novel LSTD based algorithm for cost function approximation of the MSMP problem over an infinite time horizon to make use of it for the finite time horizon decision making; (iv) developing a Python based program to implement the different phases of the MSMP transmission scheduling problem modelling, resolution, and result analysis. This paper is organized as follows. Section 2 provides an overview on MDP and projected based ADP techniques applicable to this work. The modelling approach for the MSMP problem is given in Section 3. Its stochastic DP formulation is presented in Section 4. The EG-LSTD algorithm along with its convergence properties is discussed in Section 5. Numerical simulations and their results are provided in Section 6. Section 7 concludes the paper. Finally, all proofs of the main Lemmas are provided in Appendix A.

2. Preliminaries on MDPs and ADP

This section briefly provides an overview of some MDP and ADP concepts used in the paper.

2.1. Preliminaries on MDPs

The basic structure of an MDP is defined as follows (Forootani et al., 2020):

- $\mathcal{S}$: a finite set of states with the cardinality Ω. We denote by $S(k) = S \in \mathcal{S}$ and $S(k+1) = S' \in \mathcal{S}$ two generic elements of this set at the consecutive time slots k and $k+1$.
- $\mathcal{U}$: a finite set of actions. We denote by $u(k) = u \in \mathcal{U}$ a generic element of this set at the time slot k, which is assumed to be an N-dimensional vector with the associated elements 0 or 1. The cardinality of $\mathcal{U}$ is 2^N.
- A state transition probability function $\mathcal{P}_{SS'}(u) := \left[\mathcal{P}(S'|S,u)\right]$, $\mathcal{P} : \mathcal{S} \times \mathcal{U} \times \mathcal{S} \to [0,1]$, which is defined as the probability that an action u, performed in the state S at time slot k, leads to state S' at time slot $k+1$.
- $\mathcal{L} : \mathcal{S} \times \mathcal{U} \to [0, +\infty)$ an instantaneous cost function. It is denoted as $\mathcal{L}(S,u)$ for any generic $S \in \mathcal{S}$ and $u \in \mathcal{U}$.

We define a decision $u(S,k)$ as the mapping between the whole state space $\mathcal{S}$ and the set of actions $\mathcal{U}$, at a given time slot k. For the sake of simplicity, in the paper we remove the explicit dependency on the state, i.e., $u(S,k) := u(k)$. By denoting with $\pi = \{u(0), \dots, u(\mathcal{N}-1)\}$ the sequence of decisions as the policy over the finite time horizon $\mathcal{N}$, the associated expected cost function starting from the initial state $S(0)$ over $\mathcal{N}$ is written as follows

$$J_\pi\big(S(0)\big) = E\left[\sum_{k=0}^{\mathcal{N}-1} \alpha^k \mathcal{L}\big(S(k), u(k)\big) + J_{\mathcal{N}}\big(S(\mathcal{N})\big)\right], \tag{1}$$

where $E\{\cdot\}$ is the expectation operator calculated over the visited states when applying the policy π, $0 < \alpha < 1$ is the discount factor, $J_\pi\big(S(0)\big)$ is the expected cost function of the policy π, and $J_{\mathcal{N}}(\cdot)$ is a given terminal cost function evaluated at final stage. The optimal finite time horizon discounted cost function can be expressed as

$$J^*\big(S(0)\big) = \min_{\pi} \; E\left[\sum_{k=0}^{\mathcal{N}-1} \alpha^k \mathcal{L}\big(S(k), u(k)\big) + J_{\mathcal{N}}\big(S(\mathcal{N})\big)\right]. \tag{2}$$

Note that the expressions (1),(2) can be easily extended to the infinite time horizon case by computing their limit value for $\mathcal{N} \to \infty$ and setting $J_{\mathcal{N}}$ to zero (Bertsekas, 2012). With a slight abuse of notation, we can define (1) and (2) for a generic state $S \in \mathcal{S}$ as $J_\pi(S)$ and $J^*(S)$, respectively. When we refer to the whole state space, we can define the overall cost function $J : \mathcal{S} \to \mathbb{R}^\Omega$ as a vector whose components are $J_\pi(S)$.

Definition 2.1. Let us denote the MDP state transition probability matrix for the optimal stationary policy $\pi^* = \{u^*, u^*, \dots\}$ with $\mathcal{P}^* \in \mathbb{R}^{\Omega\times\Omega}$ with elements $\mathcal{P}^*_{SS'} = \mathcal{P}_{SS'}(u^*)$, any given cost function vector with $J \in \mathbb{R}^\Omega$ (with components $J(S)$), and the optimal instant cost vector with $\mathcal{L}^* \in \mathbb{R}^\Omega$ (with components $\mathcal{L}(S,u^*)$). We define the mapping $\mathcal{F}^* : \mathbb{R}^\Omega \to \mathbb{R}^\Omega$ (Bellman operator)

$$(\mathcal{F}^*J)(S) = \min_{u\in\mathcal{U}}\Big[\mathcal{L}(S,u) + \alpha \sum_{S'\in\mathcal{S}} \mathcal{P}_{SS'}(u) J(S')\Big]. \tag{3}$$

By using a compact matrix form, we can express the Bellman operator as: $\mathcal{F}^*J = \mathcal{L}^* + \alpha\mathcal{P}^*J$. As stated in Bertsekas (2012), the mapping (3) provides a convenient shorthand notation in expressions that would be too complicated to write otherwise. It can be regarded as the optimal cost function for the one-stage problem with stage cost $\mathcal{L}$ and terminal cost αJ.

When we refer to any stationary policy $\pi = \{u, u, \dots, \}$ (either optimal or not) with corresponding stochastic matrix $\mathcal{P} \in \mathbb{R}^{\Omega\times\Omega}$ (with elements $\mathcal{P}_{SS'} = \mathcal{P}_{SS'}(u)$) and any given cost function J, the related Bellman operator becomes: $\mathcal{F}_\pi J = \mathcal{L} + \alpha\mathcal{P}J$, where, with a slight abuse of notation, we have removed the explicit dependency of $\mathcal{P}$ and $\mathcal{L}$ from the specific stationary policy. As shown later, the off-policy with respect to the optimal stationary policy π^* is denoted with $\bar{\pi} = \{\bar{u}, \bar{u}, \dots, \}$. By defining its state transition probability matrix as $\bar{\mathcal{P}} \in \mathbb{R}^{\Omega\times\Omega}$ and the instant cost vector as $\bar{\mathcal{L}}$ (with components $\bar{\mathcal{L}}(S,\bar{u})$), the related Bellman operator becomes: $\bar{\mathcal{F}}J = \bar{\mathcal{L}} + \alpha\bar{\mathcal{P}}J$.

2.2. Preliminaries on ADP

In general, solving the Bellman equation has an exponential complexity with respect to the state and action space, and is computationally prohibitive. The goal is to approximate the cost function $J : \mathcal{S} \to \mathbb{R}^\Omega$ of an MDP with a parametric architecture of the form $\tilde{J}(S,r)$, $\tilde{J} : \mathcal{S} \times \mathbb{R}^m \to \mathbb{R}^\Omega$, where $r \in \mathbb{R}^m$ is a parameter vector that has to be computed. The choice of the architecture is very significant for the success of the approximation approach. One possibility is to use the linear form

$$\tilde{J}(S,r) = \sum_{i=1}^{m} r_i \phi_i(S), \tag{4}$$

where r_i is the ith component of parameter vector $r \in \mathbb{R}^m$, and $\phi_i(S)$ are some known scalars that depend on the state S. For each state S, the approximate value $\tilde{J}(S,r)$ is the inner product of $\phi(S)$ and r where $\phi(S) = \big[\phi_1(S), \dots, \phi_m(S)\big]^T$. We refer to $\phi(S)$ as the feature vector of S, and components $\phi_i(S)$ as features. Thus the cost function is approximated by a vector in the feature subspace $\Delta = \{\Phi r | r \in \mathbb{R}^m\}$, where $\Phi \in \mathbb{R}^{\Omega\times m}$ is called feature matrix with each row $\phi(S)^T$. Note that in general we have $m \ll \Omega$. The m columns of Φ are viewed as basis functions, and Φr as a linear combination of basis functions. The vector r can be computed via the Monte Carlo simulations approaches, e.g., the LSTD method (Tsitsiklis & Van Roy, 1997). Now we recall some definitions, assumptions, and results useful for the understanding of the paper.

Assumption 1. For each admissible stationary policy π^*, $\bar{\pi}$, and π the underlying Markov chain is irreducible and regular. The related stochastic matrices $\mathcal{P}^*$, $\bar{\mathcal{P}}$, and $\mathcal{P}$ have unique steady state probability vectors $\xi^* \in \mathbb{R}^\Omega_+$ with components $\xi^*_S > 0$, $\bar{\xi} \in \mathbb{R}^\Omega_+$ with components $\bar{\xi}_S > 0$, and ξ with components $\xi_S > 0$ respectively.

Assumption 2. The matrix Φ has rank m.

Assumption 1 is equivalent to assuming that the Markov chain is irreducible, i.e., has a single recurrent class and no transient states. As explained in Bertsekas (2012), thanks to Assumption 1, the contraction property of the associated Bellman operator in the feature subspace holds, which implies the existence of the fixed point of the projected Bellman equation. On the other hand, Assumption 2 is equivalent to the basis functions (the columns of Φ) being linearly independent, and is analytically convenient because it implies that each vector J in the feature subspace Δ is represented in the form Φr with a unique parameter vector r. As a result, one can sample according to such steady probability distribution in order to compute via Monte Carlo simulations the parameter vector r of linear cost function approximation (4). These aspects are exploited in the proposed EG-LSTD algorithm.

We use the weighted Euclidean norm of any cost function vector $J \in \mathbb{R}^{\Omega}$ with respect to the vector of positive weights ξ

$$\parallel J \parallel_{\xi} = \sqrt{(J^T \Xi J)}, \tag{5}$$

where $\Xi \in \mathbb{R}^{\Omega \times \Omega}$ is the diagonal matrix with the steady state probabilities ξ_S, $S \in \mathcal{S}$ along the diagonal. Let Π denote the projection operator of any $J \in \mathbb{R}^{\Omega}$ onto Δ with respect to this norm. In other words, ΠJ implies computing the unique vector in Δ that minimizes the following

$$\hat{r} = \arg\min_{r \in \mathbb{R}^m} \|J - \Phi r\|_{\xi}^2. \tag{6}$$

It can also be written as $\Pi J = \Phi \hat{r}$, with

$$\hat{r} = (\Phi^T \Xi \Phi)^{-1} \Phi^T \Xi J, \tag{7}$$

where $\Pi = \Phi(\Phi^T \Xi \Phi)^{-1} \Phi^T \Xi$. Thanks to Assumptions 1 and 2, the inverse $(\Phi^T \Xi \Phi)^{-1}$ exists. By using the projection operator, we can introduce the projected Bellman operator $\Pi \mathcal{F}_{\pi} J = \Pi(\mathcal{L} + \alpha \mathcal{P} J)$. As mentioned before, thanks to Assumption 1, the corresponding projected Bellman equation $\Phi r = \Pi \mathcal{F}_{\pi}(\Phi r)$ has a unique fixed point (Bertsekas, 2011). The mapping $\Pi \mathcal{F}_{\pi}$ is contraction of modulus α with respect to the weighted Euclidean norm (5). We denote with $\tilde{J}_{\pi} = \Phi r_{\pi}$ the fixed point of $\Pi \mathcal{F}_{\pi}$, i.e. $\Phi r_{\pi} = \Pi \mathcal{F}_{\pi}(\Phi r_{\pi})$.

Given the system model, the LSTD algorithm adopts Monte Carlo simulations to compute the approximate cost function vector $\tilde{J}_{\pi}$ of a given policy in the lower dimensional feature space. The LSTD is performed as the policy evaluation step of the Policy Iteration algorithm (Bertsekas, 2011). For further discussion regarding the LSTD method and its convergence analysis, the reader can refer to the work reported in Tsitsiklis and Van Roy (1997).

3. Model description

In this paper, we consider N dynamical systems whose states have to be estimated by a remote estimator, for the case of N transmitting sensors through a shared wireless channel. The central node is equipped with a receiver capable of multi-packet reception, thus allowing more than one sensor to transmit simultaneously. To avoid confusion, we denote by transmission period the time interval during which each scheduled sensor transmits its encoded information to the remote estimator. For simplicity, we assume that the transmission periods are contained within the corresponding sampling periods and are synchronized across the sensors.

Consider the discrete time processes with the dynamics

$$x_i(k+1) = A_i x_i(k) + \omega_i(k), \tag{8}$$

where $x_i(k) \in \mathbb{R}^{n_i}$, with $i = 1, \dots N$ and $n_i \in \mathbb{N}$, denote the set of states associated to process i, and $\omega_i(k)$s are i.i.d. Gaussian process noises with zero mean and covariances $Q_i \in \mathbb{R}^{n_i \times n_i}$. There are N sensors, one for each dynamical system,[2] with the ith sensor having measurements

$$y_i(k) = C_i x_i(k) + v_i(k), \quad i = 1, \dots, N, \tag{9}$$

where $y_i(k) \in \mathbb{R}^{m_i}$, with $m_i \in \mathbb{N}$, and the measurement noise $v_i(k)$ are i.i.d. Gaussian noises with zero mean and covariance R_i; $\omega_i(k)$ and $v_i(k)$ are assumed to be mutually independent, and are also independent of the initial state $x_i(0)$. We assume that the sensors are smart and can run a local Kalman filter. Define $\mathcal{Y}_i(k) = \{y_i(0), y_i(1), \dots, y_i(k)\}$. Then, the local state estimates and error covariances are given by

$$\hat{x}_i^s(k|k-1) = E\Big[x_i(k)|\mathcal{Y}_i(k-1)\Big], \ \hat{x}_i^s(k|k) = E\Big[x_i(k)|\mathcal{Y}_i(k)\Big],$$

$$P_i^s(k|k-1) = E\Big[\ \big(x_i(k) - \hat{x}_i^s(k|k-1)\big)\ \big(\ x_i(k) - \hat{x}_i^s(k|k-1)\ \big)^T |\mathcal{Y}_i(k-1)\ \Big],$$

$$P_i^s(k|k) = E\Big[\big(x_i(k) - \hat{x}_i^s(k|k)\big)\big(x_i(k) - \hat{x}_i^s(k|k)\big)^T |\mathcal{Y}_i(k)\Big],$$

and can be computed using the standard Kalman filtering equations at sensors $i = 1, \dots, N$ for each process i (note that the superscript s denotes the sensor side measurement). Moreover, we assume that each pair (A_i, C_i) is detectable and the pair $(A_i, Q_i^{\frac{1}{2}})$ is stabilizable. Let $\bar{P}_i^s$ be the steady state value of $P_i^s(k|k-1)$ and $\bar{P}_i$ be the steady state value of $P_i^s(k|k)$, as $k \to \infty$, both of which exist due to the detectability assumptions.

The Kalman filtering equations at the sensor i are given by

$$\begin{aligned}
\hat{x}_i^s(k|k-1) =& A_i \hat{x}_i^s(k-1|k-1) \\
P_i^s(k|k-1) =& A_i P_i^s(k-1|k-1) A_i^T + Q_i \\
\mathcal{K}_i^s(k) =& P_i^s(k|k-1) C_i^T \big(C_i P_i^s(k|k-1) C_i^T + R_i\big)^{-1} \\
\hat{x}_i^s(k|k) =& \hat{x}_i^s(k|k-1) + \mathcal{K}_i^s(k)\big(y_i(k) - C_i \hat{x}_i^s(k|k-1)\big) \\
P_i^s(k|k) =& \big(I_{n_i} - \mathcal{K}_i^s(k) C_i\big) P_i^s(k|k-1),
\end{aligned} \tag{10}$$

where $\mathcal{K}_i^s(k)$ is the optimal filter gain, and $I_{n_i} \in \mathbb{R}^{n_i \times n_i}$ is the identity matrix.

Let $u_i(k) \in \{0, 1\}, i = 1, \dots, N$ be decision variables such that $u_i(k) = 1$ if and only if $\hat{x}_i^s(k|k)$ is to be transmitted to the remote estimator at time k. Note that transmitting state estimates when there are packet drops generally provides better estimation performance than transmitting measurements (Schenato, 2008). We consider the situation where $u_i(k)$ are computed at the remote estimator at time $k-1$ and communicated to the sensors without error via feedback links before transmission at the next time instant k.[3] Since our interest lies in decision making at the remote estimator, we assume that the decisions $u_i(k)$ do not depend on the current value of $x_i(k)$ (or functions of $x_i(k)$, such as measurements and local state estimates). Specifically, in this paper we assume that $u_i(k)$ depends only on the error covariances at the remote estimator. During the kth transmission period, a packet containing the estimated state $\hat{x}_i^s(k|k)$ is communicated according to the decision variable $u_i(k)$ to a remote estimator: if $u_i(k) = 1$, then $\hat{x}_i^s(k|k)$ is transmitted, while it is not transmitted if $u_i(k) = 0$. When scheduled, a transmission may not be successfully completed due to the interference of other transmissions and channel and receiver noise. We represent

[2] This setup can be easily extended to consider the scenario where each process is observed by a set of sensors that result in a detectable system, as considered in Pezzutto et al. (2020) for a single process, as long as the sets of sensors observing different processes are mutually exclusive.

[3] Feedback from base stations or wireless access points to the associated user equipment is common in wireless systems. These feedback information carry crucial commands regarding power control, scheduling/routing decisions and many other parameters (Pezzutto, Schenato, & Dey, 2021a).

this process through the variable $\eta_i(k)$, which is equal to 1 if the transmission of $\hat{x}_i(k|k)$ is successfully completed, 0 otherwise. The information set available at the fusion centre at the time instant k is $\mathcal{I}(k) = \bigcup_{i=1}^{N} \mathcal{I}_i(k)$, with

$$\mathcal{I}_i(k) = \Big\{ u_i(0)\eta_i(0)\hat{x}_i^s(0|0), u_i(1)\eta_i(1)\hat{x}_i^s(1|1), \dots, u_i(k-1)\eta_i(k-1)\hat{x}_i^s(k-1|k-1) \Big\}, \tag{11}$$

where, with a slight abuse of notation, if $u_i(t)\eta_i(t) = 0$ then, $u_i(t)\eta_i(t)\hat{x}_i^s(t|t) = \varnothing$, i.e. $\hat{x}_i^s(t|t)$ is missing, $t \in \{0, 1, \dots, k-1\}$. We denote the state estimates and error covariances at the remote estimator for each process by

$$\begin{aligned} \hat{x}_i(k|k) &= E\big[x_i(k)|\mathcal{I}_i(k)\big] \\ \hat{x}_i(k+1|k) &= E\big[x_i(k+1)|\mathcal{I}_i(k)\big] \\ P_i(k|k) &= E\Big[\big(x(k) - \hat{x}(k|k)\big)\big(x_k - \hat{x}(k|k)\big)^T|\mathcal{I}(k)\Big] \\ P_i(k+1|k) &= E\Big[\big(x(k+1) - \hat{x}(k+1|k)\big)\big(x(k+1) \\ &\quad - \hat{x}(k+1|k)\big)^T|\mathcal{I}(k)\Big], \end{aligned}$$

Based on the above settings on u_i and η_i, the remote estimator updates its estimation of the states as follows

$$\hat{x}_i(k|k) = \begin{cases} \hat{x}_i^s(k|k), & u_i(k)\eta_i(k) = 1 \\ A_i\hat{x}_i(k-1|k-1), & u_i(k)\eta_i(k) = 0, \end{cases}$$

$$P_i(k|k) = \begin{cases} \bar{P}_i, & u_i(k)\eta_i(k) = 1 \\ A_iP_i(k-1|k-1)A_i^T + Q_i, & u_i(k)\eta_i(k) = 0. \end{cases} \tag{12}$$

For our subsequent analysis, we introduce the matrix function

$$f_i(X) = A_iXA_i^T + Q_i. \tag{13}$$

If we define the countably infinite set

$$\mathcal{S}_i = \big\{\bar{P}_i, f_i(\bar{P}_i), f_i^2(\bar{P}_i), \dots\big\}, \quad i = 1, \dots, N, \tag{14}$$

where $f^l(\cdot)$ denotes the $l-$fold composition of $f(\cdot)$. Then it is clear from (12) that $\mathcal{S}_i$ consists of all possible values of $P_i(k|k)$ at the remote estimator. With a slight abuse of notation, hereinafter, we use $P_i(k)$ in place of $P_i(k|k)$. The affine mapping of symmetric matrices $f_i^l(\cdot)$ are defined as

$$f_i^0(X) = X, \tag{15}$$

$$f_i^1(X) = A_iXA_i^T + Q_i, \tag{16}$$

$$f_i^l(X) = \underbrace{f_i \circ \cdots \circ f_i}_{l}(X) = A_i^lX(A_i^T)^l + \sum_{t=0}^{l-1} A_i^tQ_i(A_i^T)^t. \tag{17}$$

In computations, since the state space is (countably) infinite, we will use a truncated version of $\mathcal{S}_i$ in (14) to[4]

$$\mathcal{S}_i = \big\{\bar{P}_i, f_i(\bar{P}_i), f_i^2(\bar{P}_i), \dots, f_i^{q-1}(\bar{P}_i)\big\}, \quad i = 1, \dots, N, \tag{18}$$

which will cover all possible error covariances with up to $q-1$ successive packet drops or non-transmissions. Note that it can be shown that the error due to this truncation decays exponentially with increasing N. The schematic diagram of the sensor scheduling problem addressed in this paper is shown in Fig. 1.

Remark 1. While transmitting estimates require more computational effort at the sensors, at the steady state, each sensor only needs to compute the current estimate using the steady state Kalman gain and the current measurement innovations, which is computationally not that expensive, thus justifying this choice over transmitting measurements. Moreover, if the sensors are severely computation-constrained, and are restricted to transmit measurements only, the resulting error covariance matrices for the ith process at the remote estimator will belong to the general set of positive semi-definite matrices, instead of the ordered countably infinite set $\mathcal{S}_i$ (Gupta, Hassibi, & Murray, 2007; Xu & Hespanha, 2005).

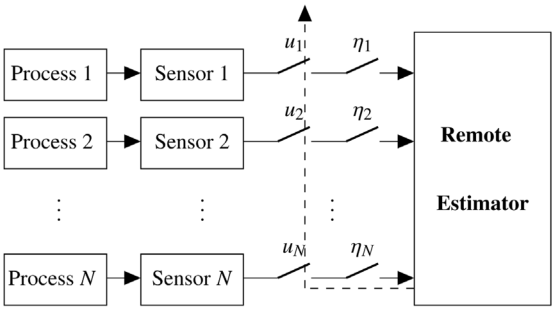

Fig. 1. Schematic diagram of the sensor scheduling problem.

Remark 2. It is worth noting it is the remote estimator which needs to know whether the packets have been received or lost, whereas the sensors simply perform their local estimates based on their local measurements only.

3.1. Channel model

Denote by $\mathbf{P}_i^{\text{tx}}$ the transmitted power of the ith sensor, while g_i denotes the slow fading component of the channel power gain from the ith sensor to the remote estimator (usually due to distance based attenuation and shadow fading, and $h_i(k)$ is the fast fading component (due to multipath effects and mobility) of the same channel during the kth transmission period. We assume that $\mathbf{P}_i^{\text{tx}}$ and g_i are constant, while $h_i(k)$ is modelled as a temporally independent identically distributed exponential random variable (Rayleigh fading) with unity mean, i.e. $h_i(k) \sim Exp(1)$, with $h_i(k), h_j(t)$ being mutually statistically independent for $\forall i \neq j$ (Pezzutto et al., 2020). It follows that the received power at the remote estimator from the ith sensor $\mathbf{P}_i^{\text{rc}}(k)$ during the kth transmission interval is

$$\mathbf{P}_i^{\text{rc}}(k) = \begin{cases} \mathbf{P}_i^{\text{tx}}g_ih_i(k), & \text{if } u_i(k) = 1 \\ 0, & \text{if } u_i(k) = 0. \end{cases} \tag{19}$$

Given $u_i(k) = 1$, the received power is an exponential random variable with mean $\lambda_i = (g_i\mathbf{P}_i^{\text{tx}})^{-1}$, i.e. $\mathbf{P}_i^{\text{rc}}(k) \sim Exp(\lambda_i)$. Due to the intrinsic nature of the wireless medium, background channel and/or receiver noise is also present. We model it as an Additive White Gaussian Noise (AWGN) whose average power at the remote estimator is δ^2 (Pezzutto et al., 2020).

Without Successive Interference Cancellation (SIC) at the remote estimator (which imposes an optimized decoding order by decoding the strongest signal and removing its contribution from the received signal iteratively Pezzutto et al., 2020), the Signal-to-Interference-and-Noise Ratio (SINR) corresponding to the packet containing $\hat{x}_i^s(k|k)$ is

$$SINR_i(k) = \frac{u_i(k)\mathbf{P}_i^{\text{tx}}g_ih_i(k)}{\sum_{j\neq i} u_j(k)\mathbf{P}_j^{\text{tx}}g_jh_j(k) + \delta^2}. \tag{20}$$

Note that SIC can further improve the performance at the remote estimator, but for simplicity, we do not consider it in this work.

A packet from the ith sensor at the kth time slot can be decoded without error if $SINR_i(k) > \gamma$, where $\gamma > 0$ is a threshold that is chosen based on the modulation and coding schemes used.

[4] With a slight abuse of notation, we say $\mathcal{S}_i$ the truncated version of (14).

We need to have $\gamma \in (0,1)$, which is necessary to enable multi-packet reception, and can be achieved via clever modulation and coding schemes at the transmitters. It follows that the packet arrival process from the ith sensor can be expressed as

$$\eta_i(k) = \begin{cases} 1, & \text{if } \ SINR_i(k) > \gamma \\ 0, & \text{otherwise.} \end{cases} \tag{21}$$

Since the channel gains are independent across time slots, $\eta_i(k)$ is also an i.i.d. Bernoulli process. However $\eta_i(k)$, $\eta_j(k)$ for $j \neq i$ may be dependent on each other due to interference within a given time slot. Arrival probabilities $\mathcal{P}(\eta_i(k)=1)$ can be computed for fixed transmission powers and scheduling policies, using the joint distribution of the fast fading gains, as shown in Papandriopoulos, Evans, and Dey (2005), and are not repeated here.

4. Dynamic programming formulation of sensor scheduling problems

In Section 3, we have provided a model description for MSMP transmission scheduling problem over a noisy wireless communication channel. Each sensor i is equipped with a local Kalman filter characterized by the corresponding steady state error covariance matrix $\bar{P}_i$. In this section, we formulate formally the MSMP sensor scheduling problem as an MDP with the associated DP framework. If we define the finite set

$$\mathcal{S}_i = \big\{\bar{P}_i, f_i(\bar{P}_i), f_i^2(\bar{P}_i), \dots, f_i^{q-1}(\bar{P}_i)\big\}, \quad i = 1, \dots, N, \tag{22}$$

where the local state at the i-the sensor is $S_i(k) = P_i(k)$, and clearly $S_i(k) \in \mathcal{S}_i$. Similarly, the scheduling/control action at sensor i is given by $u_i \in \mathcal{U}_i = \{1,0\}$, where $u_i = 1$ represents "*transmit*" and $u_i = 0$ denotes the action of "*do not transmit*". The decision at time k for the state $S_i(k) = P_i(k)$ is denoted as $u_i(k) \in \mathcal{U}_i(S_i(k))$. Based on these definitions, one can then define a combined time-homogeneous MDP at the remote estimator as a tuple $\mathcal{Q} = \langle \mathcal{S}, \mathcal{U}, \mathcal{T}, \mathcal{L}\rangle$.

4.1. MDP formulation of multiple sensors multiple processes scheduling problems

The various components of the combined MDP are defined as follows.

- $\mathcal{S}$ is the entire state space, that is $\mathcal{S} = \bigotimes_{i=1}^N \mathcal{S}_i$, where $\bigotimes$ denotes the Cartesian product. The cardinality of $\mathcal{S}$ is denoted by Ω, i.e. $|\mathcal{S}| = \Omega$. Moreover, we denote with $S(k) = S$ and $S(k+1) = S'$ two generic states at consecutive time slots.
- $\mathcal{U} = \bigotimes_{i=1}^N \mathcal{U}_i$ is the overall action set. We denote with $\mathcal{U}(S)$ the set of actions at state S. We denote by $u\big(S(k)\big) : \mathcal{S} \to \mathcal{U}$ as the mapping between the whole state space $\mathcal{S}$ and the set of actions $\mathcal{U}$, at a given time slot k.
- $\mathcal{T}$ is the state transition mapping, represented by the state transition probability matrix with the elements $\mathcal{P}_{SS'}\big(u\big) := \mathcal{P}[S'|S,u]$, i.e. $\mathcal{P} : \mathcal{S} \times \mathcal{U} \times \mathcal{S} \to [0,1]$, which is formally defined as the probability that an action u in state S at the time slot k will led to state S' at time slot $k+1$.
- $\mathcal{L} : \mathcal{S} \times \mathcal{U} \to [0, +\infty)$ is the instantaneous cost. In our case $\mathcal{L}(S,u) = \sum_{i=1}^N \big(Tr(S_i) + \mu\ u_i\big)$.

where recall that $S_i(k) = P_i(k)$. We denote by $\pi = \{u_i(k), i = 1, \dots, N, k = 0, \dots, \mathcal{N}-1\}$ the policy, that is to say, the sequence of control functions applied over the finite time horizon $\mathcal{N} - 1$. DP techniques aim at computing the optimal policy $\pi^*\big(S(0)\big) \in \arg\min_\pi J_\pi\big(S(0)\big)$. Under the assumption of having a relative small number of states, we can apply the exact DP algorithm, which exploits Bellman's principle of optimality. In particular, the *optimal cost function* can be recursively expressed as

$$\begin{aligned} J_k^*\big(S(k)\big) &= \min_{u\in\mathcal{U}} E\Big\{\sum_{i=1}^N Tr\big(S_i(k)\big) + \mu \sum_{i=1}^N u_i(k) + \alpha J_{k+1}^*\big(S(k+1)\big)\Big\} \\ &= \min_{u\in\mathcal{U}}\Big\{\sum_{i=1}^N Tr\big(S_i(k)\big) + \mu \sum_{i=1}^N u_i(k) + \alpha \sum_{S'\in\mathcal{S}} \mathcal{P}_{SS'}(u) J_{k+1}^*(S')\Big\}. \end{aligned} \tag{23}$$

Solving the above cost function for different values of μ means minimizing the sum of the trace of the expected error covariances across the sensors for different average number of total transmissions (across the sensors). Thanks to Bellman's principle of optimality, the generated cost function $J_k^*(\cdot)$ at each time slot k is equal to the optimal cost function for the tail sub-problem from time k to time $\mathcal{N} - 1$. It is worth noting that the control chosen at the time slot k affects the state transitions from $S(k)$ to $S(k+1)$, and thus the expected cost function $J_{k+1}^*(\cdot)$. The cost generated at the last step is equal to the optimal cost function $J^*\big(S(0)\big)$ starting from $S(0)$. Note that the terminal cost function has to be defined to compute $J^*\big(S(0)\big)$. To do that, one can choose, e.g., the instantaneous cost calculated at the terminal stage or the cost function computed over an infinite time horizon (Bertsekas, 2011, 2012; Forootani, Liuzza, et al., 2019). Moreover, μ is the weighting parameter that can be interpreted as the cost of transmission, e.g. transmission power. This can be easily extended to the case by considering different weighting parameters μ_i, for the ith sensor. More details can be found in Pezzutto, Schenato, and Dey (2021b), however for single sensor and using traditional dynamic programming techniques.

Remark 3. The results of this paper can be transferable to the case where there is a feedback control loop to each of the processes, provided the control signals are received without error at the actuators for each process, that is there is no link error in the feedback link for the control messages. As proved in Schenato, Sinopoli, Franceschetti, Poolla, and Sastry (2007) the estimator and controller can be designed separately and the resulted controller can be considered as a linear function of the state estimate given by the modified time-varying Kalman filter with packet losses. As a result, an optimal remote Linear Quadratic (LQ) control of our multi-process multi-sensor system can be implemented by using the optimal transmission scheduling policies derived by minimizing a finite/infinite horizon weighted sum of the error covariance matrices summed over all the processes, along with linear optimal LQ control signals for individual processes, that can be fed back without error to the corresponding actuators (see also Leong, Quevedo, Tanaka, Dey, & Ahlen, 2017).

4.2. Approximation architectures

We use linear feature-based parametric architectures to approximate cost functions. In particular, in case of stationary systems, the cost function of a policy $J_\pi(S)$ and the optimal cost function $J^*(S)$ can be approximated respectively as $\tilde{J}_\pi(S,r) = \phi(S)^T r_\pi$ and $\tilde{J}^*(S,r) = \phi(S)^T r^*$. More specifically, $\phi(S)^T = [\phi_1(S), \phi_2(S), \dots, \phi_m(S)]$ is a vector of feature (or basis) functions evaluated for a given S, while r is a vector of m parameters to be tuned by training the selected architecture. Training the architecture means computing a suitable $\hat{r}^*$ (or $\hat{r}_\pi$) by some simulation based mechanisms to further approximate $J^*(S)$ (or $J_\pi(S)$).

In this paper, for each state $S = (S_1, S_2, \ldots, S_N)$, we define the following feature vector

$$\phi(S)^T = [1, \log_2\big(Tr(S_1)\big), \ldots, \log_2\big(Tr(S_N)\big)]. \tag{24}$$

The number of features m is equal to $N + 1$ and the features are defined as the logarithm of the trace of the error covariance matrix S_i. The reason of such feature vector definition basically lies in the nature of the problem since the range of values $Tr(P_i(k))$ can vary widely for independent variables, i.e., in our case we deal with quantities with different order of magnitude. In order to make them comparable, we opted for logarithmic basis functions to scale down the range of independent variables and in addition to limit the growth rate of the feature functions. This paper does not address the construction of the feature functions, which is indeed an important research area, see Bertsekas and Yu (2009), Busoniu, Ernst, De Schutter, and Babuska (2011). A limited number of well-crafted feature functions can capture the dominant non-linearities of cost functions for complex systems, and thus their linear combination can work well as an approximation architecture, see Bertsekas (2012), Busoniu et al. (2011).

5. The enhanced exploration Greedy LSTD algorithm and its convergence properties

This section describes the EG-LSTD algorithm, which stems from the MG-LSTD algorithm presented in Forootani et al. (2020a), where its convergence properties were discussed qualitatively and supported by experimental verification. The EG-LSTD algorithm is employed for computing the cost function approximation of the MSMP problem over an infinite time horizon, which can be used for the finite time horizon decision making process, see Section 6. Unlike the basic LSTD algorithm (Bertsekas, 2012), the EG-LSTD directly focuses on the approximate optimal cost function by calculating the vector $\hat{r}$ with the policy improvement within multi-trajectory Monte Carlo simulations. For more details, the reader can refer to Bertsekas (2012), Forootani et al. (2020a). As shown in this section, the EG-LSTD algorithm is an off-policy method since: (i) it chooses the initial state of each trajectory by an off-policy mechanism; (ii) it embeds the policy improvement step within the LSTD iterations. The first aspect enhances the EG-LSTD exploration capabilities, while the second one its exploitation means. On the other hand, the conventional LSTD algorithm is an on-policy method, and its basic drawback comes from the fact that it learns the state value function of the fixed policy learnt via single trajectory Monte Carlo simulations (Bertsekas, 2011; Forootani et al., 2020a, 2022), thus it cannot be used for control problems.

It is worth highlighting that the main differences of the EG-LSTD and the MG-LSTD are as follows: (i) MG-LSTD is a multi-trajectory Monte Carlo simulation method where the length of each trajectory is considered long enough so that the Markov chain forgets it initial state. However, in EG-LSTD, the length of each trajectory is 1, (ii) the EG-LSTD is an off policy Monte Carlo simulation method, whereas the MG-LSTD is categorized as an on-policy method since the total length of trajectories are significantly greater than number of initial states. Unlike the MG-LSTD where the selection of the initial state of each trajectory does not affect the convergence property, in the EG-LSTD, we have to provide sufficient conditions to guarantee convergence. Therefore in this paper we provide a condition for choosing the initial state of each trajectory in such a way that convergence of the proposed method can be guaranteed.

Algorithm 1 shows the pseudo code of the EG-LSTD. It computes the approximate parameter vector $\hat{r}^*$ by embedding the policy improvement step within the LSTD iterations and by using Monte Carlo simulations. The following initial conditions are set: $C_{-1} = 0$, $d_{-1} = 0$, and $r_0 = \bar{r}$, with $\bar{r}$ an initial guess. Moreover, Σ is selected as a symmetric positive definite matrix and σ is a positive scalar.

Algorithm 1 EG-LSTD pseudo code

```
While $l \leq \mathcal{H}$, $l \in \{0, 1, \ldots, \mathcal{H}\}$
  1. For $\forall u \in \mathcal{U}\big(S(l)\big)$
      (a) Generate a candidate new state $S'_u(l)$ from $S(l)$ by Monte Carlo simulation and by applying to the system model the admissible control action $u$ and transition probability $\mathcal{P}_{SS'}(u)$
      (b) Compute the corresponding features vector $\phi\big(S'_u(l)\big)$
      (c) Compute the cost $\mathcal{L}\big(S(l), u\big)$
      (d) Calculate the candidate matrix $C_l(u)$
          $C_l(u) = (1 - \frac{1}{l+1})C_{l-1} + (\frac{1}{l+1})\phi\big(S(l)\big)\Big(\phi\big(S(l)\big) - \alpha\phi\big(S'_u(l)\big)\Big)^T$
      (e) Calculate the vector $d_j$
          $d_l = (1 - \frac{1}{l+1})d_{l-1} + \frac{1}{l+1}\phi\big(S(l)\big)\mathcal{P}\big(S(l)\big)$
      (f) Having $C_l(u)$ and $d_l(u)$, compute the candidate parameters vector update $\hat{r}_l(u)$ as follows
          $\hat{r}_{l+1}(u) = \big(C_l(u)^T\Sigma^{-1}C_l(u) + \sigma I\big)^{-1}\big(C_l(u)^T\Sigma^{-1}d_l + \sigma\hat{r}_l\big)$
  2. Choose the pair $\big(\hat{r}_{l+1}, S'_u(l)\big)$ and the corresponding control action $\hat{u}^*$ by
      $\hat{u}^*(S(l)) : \arg\min_{u \in \mathcal{U}(S(l))} \Big(\mathcal{L}(S(l), u) + \phi\big(S'_u(l)\big)^T\hat{r}_{l+1}(u)\Big),$
  3. Set $l \leftarrow l + 1$, $C_{l-1} \leftarrow C_l(u)$, and $d_{l-1} \leftarrow d_l$
  4. Generate the new initial state $S(l)$ from $S(l-1)$ based on $\bar{\mathcal{P}}$ in (25) (and hence $\mathcal{P}^*$).
```

5.1. Convergence analysis of EG-LSTD

Inspired by the work reported in Bertsekas and Yu (2009), we discuss hereafter the convergence properties of the EG-LSTD. Such algorithm chooses the initial state of each trajectory with the probability distribution $\bar{\xi}$ associated to an irreducible state transition probability matrix

$$\bar{\mathcal{P}} = (I - \mathcal{B})\mathcal{P}^* + \mathcal{B}\mathcal{E}, \tag{25}$$

where $\mathcal{B}$ is a diagonal matrix with diagonal components $\beta_S \in (0, 1)$ and $\mathcal{E}$ is another state transition probability matrix. In this framework, at state S, the next state is generated with probability $1 - \beta_S$ according to state transition probabilities $\mathcal{P}^*_{SS'}$, and with probability β_S according to the state transition probabilities $\mathcal{E}_{SS'}$. Here pairs (S, S') with $\mathcal{E}_{SS'} > 0$ need not correspond to physically plausible state transitions. Now consider to find the fixed point of the following projected equation

$$\Phi r = \bar{\Pi}\mathcal{F}^*(\Phi r), \tag{26}$$

where $\bar{\Pi}$ is projection with respect to the norm $\|\cdot\|_{\bar{\xi}}$ corresponding to the steady state distribution $\bar{\xi}$ of $\bar{\mathcal{P}}$. More specifically, we desire to solve the following least squares minimization problem: $r^* = \arg\min_{r \in \mathbb{R}^m} \|\Phi r - \mathcal{F}^*(\Phi r)\|_{\bar{\xi}}$. The following lemma provides a condition on the selection of distribution $\bar{\xi}$ and diagonal elements of the matrix $\mathcal{B}$, i.e. β_i, to ensure the contraction property with respect to $\|\cdot\|_{\bar{\xi}}$.

Lemma 5.1. *Assume that $\bar{\mathcal{P}}$ is irreducible and $\bar{\xi}$ is its unique invariant distribution (see Assumption 1). Then $\mathcal{F}^*$ and $\bar{\Pi}\mathcal{F}^*$ are contraction with respect to $\|\cdot\|_{\bar{\xi}}$, provided $\bar{\alpha} < 1$, where $\bar{\alpha} = \alpha/\sqrt{1-\beta}$. The associated modulus of contraction is at most equal to $\bar{\alpha}$, with $\beta = \max_{S\in\mathcal{S}} \ \beta_S$.*

The next lemma gives an estimate of the error in estimating J^* with the fixed point of $\bar{\Pi}\mathcal{F}^*$.

Lemma 5.2. *Consider the fixed point Φr^* of $\bar{\Pi}\mathcal{F}^*$, then we have the error bound: $\|J^* - \Phi r^*\|_{\bar{\xi}} \le 1/\sqrt{1-\bar{\alpha}^2}\ \|J^* - \bar{\Pi}J^*\|_{\bar{\xi}}$.*

The main concept regarding convergence result is that if the initial state of each trajectory is chosen slightly differently from the frequencies natural to the underlying MDP, then we can guarantee $\bar{\alpha} < 1$, hence convergence can be preserved. In the following, we discuss the construction of simulation based approximations to the projected $\Phi r = \bar{\Pi}\mathcal{F}^*(\Phi r)$. In this regard let us consider the iterative estimation of the parameter vector r given by $\Phi r_{l+1} = \bar{\Pi}\mathcal{F}^*(\Phi r_l)$. By expressing the projection as a least squares minimization, we see that r_{l+1} is given by

$$r_{l+1} = \arg\min_{r\in\mathbb{R}^m} \|\Phi r - \mathcal{F}^*(\Phi r_l)\|^2_{\bar{\xi}}, \tag{27}$$

or equivalently

$$r_{l+1} = \arg\min_{r\in\mathbb{R}^m} \sum_{S\in\mathcal{S}} \bar{\xi}_S \left(\phi(S)^T r - \left\{\mathcal{L}(S,u^*) + \alpha \sum_{S'\in\mathcal{S}} \mathcal{P}^*_{SS'}\phi(S')^T r_l\right\}\right)^2$$

where $\sum_{S'\in\mathcal{S}} \mathcal{P}^*_{SS'} = 1$, $\mathcal{P}^*$ is a row stochastic matrix, $\phi(S)^T$ and $\phi(S')^T$ are features corresponding to states S and S', respectively. By setting the gradient of the minimized expression above to 0 we have

$$r_{l+1} = \left(\sum_{S\in\mathcal{S}} \bar{\xi}_S \phi(S)\phi(S)^T\right)^{-1} \left(\sum_{S\in\mathcal{S}} \bar{\xi}_S \phi(S)\left\{\mathcal{L}(S,u^*) + \alpha \sum_{S'\in\mathcal{S}} \mathcal{P}^*_{SS'}\phi(S')^T r_l\right\}\right).$$

Since we do not have the probability distribution $\bar{\xi}_S$ and $\mathcal{P}^*_{SS'}$ we try to calculate them by simulation. The basic simulation methodology consists of generating a sequence of states $\left\{S(0), S(1), \dots\right\}$ according to the distribution $\bar{\xi}$, and a sequence of state transitions $\left\{\big(S(0), S'(0)\big), \big(S(1), S'(1)\big), \dots\right\}$ with probabilities $\mathcal{P}^*_{SS'}$. We add that this scenario is different with generating single trajectory according to the distribution ξ^*. In this sense we try to estimate the above recursive equation by using the following relation

$$r_{l+1} = \left(\sum_{t=0}^{l} \phi\big(S(t)\big)\phi\big(S(t)\big)^T\right)^{-1} \left(\sum_{t=0}^{l} \phi\big(S(t)\big)\Big(\mathcal{L}\big(S(t),u^*\big) + \alpha\phi\big(S'(t)\big)^T r_t\Big)\right). \tag{28}$$

The probabilistic mechanism in (28) is subject to the following conditions:

1. The sequence $\left\{S(0), S(1), \dots\right\}$ is generated based on the distribution $\bar{\xi}$ associated to $\bar{\mathcal{P}}$, which defines projection norm $\|\cdot\|_{\bar{\xi}}$, in the sense that with probability 1,

$$\lim_{l\to\infty} \sum_{t=0}^{l} \delta\big(S(t)=S\big)/l+1 = \bar{\xi}_S, \qquad \forall S\in\mathcal{S}, \tag{29}$$

where $\delta(\cdot)$ denotes the indicator function.

2. The sequence $\left\{\big(S(0), S'(0)\big), \big(S(1), S'(1)\big), \dots\right\}$ is generated according to stochastic matrix $\mathcal{P}^*$ with state transition probabilities $\mathcal{P}^*_{SS'}$, that is

$$\lim_{l\to\infty} \frac{\sum_{t=0}^{l}\delta\big(S(t)=S, S'(t)=S'\big)}{\sum_{t=0}^{l}\delta\big(S(t)=S\big)} = \mathcal{P}^*_{SS'}, \ \ \forall S, S'\in\mathcal{S}. \tag{30}$$

3. The policy improvement is embedded into the LSTD steps: given a generic initial state $S(l)$ along the current Monte Carlo trajectory, we generate a set of possible next state $S'_u(l)$, one for each admissible action $u \in \mathcal{U}\big(S(l)\big)$. This set is created from the state transition probabilities $\mathcal{P}_{SS'}(u)$. The corresponding parameter vector update $\hat{r}_{l+1}(u)$ is calculated at each candidate next state (note that the same definitions used in the recursive LSTD are applied). Then, we choose the control action by minimizing the sampled approximate cost function shown in Algorithm 1 step (2).

The iteration (28) can be written equivalently as

$$\hat{r}_{l+1} = \left(\sum_{S\in\mathcal{S}} \hat{\bar{\xi}}_{l,S}\ \phi(S)\phi(S)^T\right)^{-1} \left(\sum_{S\in\mathcal{S}} \hat{\bar{\xi}}_{l,S}\phi(S)\Big(\mathcal{L}(S,u^*) + \alpha\sum_{S'\in\mathcal{S}} \hat{\mathcal{P}}^*_{l,SS'}\phi(S')^T\hat{r}_l\Big)\right), \tag{31}$$

where

$$\hat{\bar{\xi}}_{l,S} = \frac{\sum_{t=0}^{l}\delta\big(S(t)=S\big)}{l+1}, \qquad \forall S\in\mathcal{S}, \tag{32}$$

$$\hat{\mathcal{P}}^*_{l,SS'} = \frac{\sum_{t=0}^{l}\delta\big(S(t)=S, S'(t)=S'\big)}{\sum_{t=0}^{l}\delta\big(S(t)=S\big)}, \quad \forall S, S'\in\mathcal{S}. \tag{33}$$

Thanks to Assumption 1, we have $\hat{\bar{\xi}}_{l,S} \to \bar{\xi}_S$, $\hat{\mathcal{P}}^*_{l,SS'} \to \mathcal{P}^*_{SS'}$. Let us consider again the fixed point mapping $\Phi r^* = \bar{\Pi}\mathcal{F}^*(\Phi r^*)$, we can write $\Phi^T\bar{\Xi}\left(\Phi r^* - \alpha\mathcal{P}^*\Phi r^* - \mathcal{L}^*\right) = 0$ where $\bar{\Xi}$ is the matrix having diagonal elements equal to steady-state distribution of $\bar{\xi}$. This condition can be written in matrix form as $Cr^* = d$, where

$$C = \Phi^T\bar{\Xi}(I - \alpha\mathcal{P}^*)\Phi, \qquad d = \Phi^T\bar{\Xi}\mathcal{L}^*, \tag{34}$$

We approximate the matrix C and vector d by:[5]

$$C_l = 1/(l+1)\sum_{t=0}^{l}\phi\big(S(t)\big)\Big(\phi\big(S(t)\big) - \alpha\phi\big(S'(t)\big)\Big)^T,$$

and

$$d_l = 1/(l+1)\sum_{t=0}^{l}\phi\big(S(t)\big)\mathcal{L}\big(S(t),u^*\big),$$

the corresponding approximation then is $r^*_l = C_l^{-1}d_l$, with

$$C_l = \sum_{S\in\mathcal{S}} \frac{\sum_{t=0}^{l}\delta\big(S(t)=S\big)}{l+1}\phi(S)\ \Bigg(\ \phi(S) - \alpha\sum_{S'\in\mathcal{S}}\frac{\sum_{t=0}^{l}\delta\big(S(t)=S, S'(t)=S'\big)}{\sum_{t=0}^{l}\delta\big(S(t)=S\big)}\phi(S')\ \Bigg)^T$$

and finally: $C_l = \sum_{S\in\mathcal{S}}\hat{\bar{\xi}}_{l,S}\phi(S)\Big(\phi(S) - \alpha\sum_{S'\in\mathcal{S}}\hat{\mathcal{P}}^*_{l,SS'}\phi(S')\Big)$. Similarly we can write: $d_l = \sum_{S\in\mathcal{S}}\hat{\bar{\xi}}_{l,S}\phi(S)\mathcal{L}(S,u^*)$. Since the empirical frequencies $\hat{\bar{\xi}}_{l,S}$ and $\hat{\mathcal{P}}^*_{l,SS'}$ asymptotically converge to the

[5] We can compute matrix C recursively as noted in Algorithm 1 where the dependency to the control input is shown.

probabilities $\bar{\xi}_S$ and $\mathcal{P}^*_{SS'}$ respectively, we have with probability 1 (Bertsekas, 2011; Nedic & Bertsekas, 2003)

$$C_l \longrightarrow \sum_{S\in\mathcal{S}} \bar{\xi}_S \phi(S)\Big(\phi(S) - \alpha \sum_{S'\in\mathcal{S}} \mathcal{P}^*_{SS'}\phi(S')\Big)^T$$
$$= \Phi^T \bar{\Xi}(I - \alpha\mathcal{P}^*)\Phi = C,$$

and we have

$$d_l \longrightarrow \sum_{S\in\mathcal{S}} \bar{\xi}_S \phi(S) \sum_{S'\in\mathcal{S}} \mathcal{P}_{SS'}\mathcal{L}(S,u^*) = \Phi^T \bar{\Xi}\mathcal{L}^* = d.$$

In the following we introduce some useful lemmas applicable for the convergence results of the proposed algorithm.

Lemma 5.3. *For any stochastic matrix $\mathcal{P}^* \in \mathbb{R}^{\Omega\times\Omega}$, the matrix $\bar{\Xi}(I - \alpha\mathcal{P}^*)$ is positive definite.*[6]

Lemma 5.4. *Matrix C is positive definite.*

Consider again

$$r_{l+1} = \arg\min_{r\in\mathbb{R}^m} \|\Phi r - \mathcal{F}^*(\Phi r_l)\|^2_{\bar{\xi}}. \tag{35}$$

By setting to 0 the gradient with respect to r of the above quadratic expression, we obtain the orthogonality condition

$$\Phi^T \bar{\Xi}\big(\Phi r_{l+1} - (\mathcal{L}^* + \alpha\mathcal{P}^*\Phi r_l)\big) = 0,$$

which yields

$$r_{l+1} = r_l - G^{-1}(Cr_l - d), \ \ G = \Phi^T\bar{\Xi}\Phi. \tag{36}$$

Lemma 5.5. *$G^{-1} = (\Phi^T\bar{\Xi}\Phi)^{-1}$ exists and it is symmetric positive definite.*

Theorem 5.6. *If matrix C is positive definite (but not symmetric in general) then the following holds: (i) Eigenvalues of C have positive real parts, (ii) det($G^{-1}C$) $<$ 1, (iii) $G^{-1}C$ is positive definite, (iv) $I - G^{-1}C$ has the eigenvalues strictly within unit circle, hence the iteration (36) is convergent.*

Note that a similar proof can be found in Bertsekas (2011). However, in our case the approach is different since we considered the projection with respect to a weighted Euclidean norm different from the natural frequencies of the MDP. The recursive equation (36) in the more general form can be written as follows

$$r_l = r_{l-1} - G_l^{-1}(C_l r_{l-1} - d_l), \tag{37}$$

where $r_l \rightarrow r^*$ (it is convergent), provided that $C_l \rightarrow C$, $d_l \rightarrow d$, and $G_l \rightarrow G$ with probability 1 and the matrix $I - G^{-1}C$ is contraction (see Bertsekas, 2011 for a similar iterative equation). In Theorem 5.6, we have proved matrix C is invertible and positive definite. However, this property may not hold for C_l until a sufficient number of samples in the Monte Carlo simulation are acquired for its calculation. To resolve this issue, a regularization term is introduced. More specifically, in each iteration along the Monte Carlo trajectory, we compute $\hat{r}_l$ by solving the following least squares problem

$$\min_r \Big\{\big(d_l - C_l r\big)^T \Sigma^{-1}\big(d_l - C_l r\big) + \sigma \parallel r - \hat{r}_l \parallel^2\Big\}.$$

By setting the objective function gradient to 0, we have

$$\hat{r}_{l+1} = \big(C_l^T\Sigma^{-1}C_l^T + \sigma I\big)^{-1}\big(C_l^T\Sigma^{-1}d_l + \sigma\hat{r}_l\big), \tag{38}$$

where the quadratic term $\sigma\|r - \hat{r}_l\|^2$ is known as a regularization term (here, $\|\cdot\|$ denotes the L_2-norm), and has the effect of biasing the estimate $\hat{r}_{l+1}$ towards the previous parameter vector estimation $\hat{r}_l$. We consider the heuristic guess $\bar{r}$ for the parameter vector $\hat{r}_0$. It is based on some intuition about the problem at hand. Moreover, the matrix Σ and the coefficient σ are respectively positive definite and positive (Hoffman, Lazaric, Ghavamzadeh, & Munos, 2012). To see more discussion on the selection of matrix Σ, we refer the reader to Bertsekas (2011), Wang, Polydorides, and Bertsekas (2009), and for an empirical study, to Forootani et al. (2020a). The convergence of the iteration (38) follows from $C_l \rightarrow C$, $d_l \rightarrow d$ and the following result, the proof of which can be found in Appendix A.

Lemma 5.7. *The recursive iteration $\hat{r}_{l+1} = \big(C^T\Sigma^{-1}C^T + \sigma I\big)^{-1}\big(C^T\Sigma^{-1}d + \sigma\hat{r}_l\big)$ is convergent.*

6. Numerical results

In this section, some numerical examples are presented to show the effectiveness of the proposed approach for MSMP transmission scheduling problems, both for the exact DP and ADP cases. A Python based program is developed to implement and demonstrate the different performance related aspects of the proposed method. In the reported examples, we limit ourselves to the case of $N = 3$ and $N = 5$ processes. It can be shown that the state space explosion can occur even with relatively small values of N and q. Indeed, increasing such parameters causes an exponential growth in the size of the state space. In the following examples, we assume (for simplicity) that all the slow fading channel gains and the sensor transmission powers are identical for all the processes (see Pezzutto et al., 2020 for a similar assumption).

Let us consider the following MIMO dynamic systems

$$A_1 = \begin{bmatrix} 1.16 & -1 & 0.2 & 0.1 \\ 0.1 & 1.8 & 0.2 & 0.5 \\ 1.5 & 0.2 & 0.1 & 0.6 \\ 0.1 & 0.7 & -0.3 & 1 \end{bmatrix}, \ C_1 = \begin{bmatrix} 1 & 2 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix},$$

$$Q_1 = \begin{bmatrix} 0.4 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0.3 \end{bmatrix}, \ R_1 = \begin{bmatrix} 0.2 & 0 \\ 0 & 0.3 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} -1.8 & -0.2 & -0.1 & 0 \\ -0.65 & -0.45 & 0 & -0.3 \\ -0.8 & -0.8 & -0.3 & -0.4 \\ 0.3 & 0.2 & -0.3 & 0.4 \end{bmatrix}, \ C_2 = \begin{bmatrix} 2 & 0 & 0 & 1 \\ 0 & 0 & 2 & 0 \end{bmatrix},$$

$$Q_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \ R_2 = \begin{bmatrix} 2 & 0 \\ 0 & 0.15 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} 0.16 & -0.17 & 2.8 & 1.2 \\ 0.8 & 0.5 & 0.2 & 1.6 \\ 1.9 & 0.25 & -0.15 & 0.5 \\ 0.6 & 0.4 & -0.3 & 0.7 \end{bmatrix}, \ C_3 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0.5 \end{bmatrix},$$

$$Q_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \ R_3 = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.2 \end{bmatrix}$$

$$A_4 = \begin{bmatrix} -0.16 & 0.12 & 0.2 & 0.1 \\ 0.1 & 0.3 & -0.2 & -0.6 \\ 0.3 & 0.2 & 0.1 & 0.5 \\ -1 & 0.5 & 0.9 & 1.2 \end{bmatrix}, \ C_4 = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix},$$

[6] Positive definiteness of a matrix refers to its symmetric part.

$$Q_4 = \begin{bmatrix} 0.4 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0.2 \end{bmatrix}, \; R_4 = \begin{bmatrix} 1 & 0 \\ 0 & 0.3 \end{bmatrix},$$

$$A_5 = \begin{bmatrix} -0.1 & 0.8 & 0.2 & 1.3 \\ 0.2 & 0.3 & -0.2 & 0.7 \\ 0.4 & 0.1 & 0.05 & 0 \\ 0.6 & 0.5 & -0.3 & 0.9 \end{bmatrix}, \; C_5 = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0.2 \end{bmatrix},$$

$$Q_5 = \begin{bmatrix} 0.6 & 0 & 0 & 0 \\ 0 & 0.8 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0.4 \end{bmatrix}, \; R_5 = \begin{bmatrix} 2 & 0 \\ 0 & 0.1 \end{bmatrix}.$$

It is straightforward to calculate $\bar{P}_i$ for each dynamic system by using (10). In the first two examples, due to the computational load of the DP algorithm, we use a relatively low value for the discount factor $\alpha = 0.7$.

Example 1. Consider the systems 1, 2; $\mathcal{N} = 30$; $q = 10$.

In this example, the maximum number of packet drops is $q = 10$, hence the cardinality of the state space is $\Omega = q^2 = 100$ (see (18)). We compare the exact DP algorithm results with two other non-optimal policies, i.e., the myopic policy and the maximum covariance policy. The myopic policy and the maximum covariance policy are defined as follows for the sensor scheduling problem with N dynamic systems.

Myopic policy: At the kth time slot, we generate random $(h_1, h_2, \dots, h_N)$ as independent exponentially distributed random variables. We choose the sensor with the maximum h_i as the candidate that is likely to be scheduled. The received SNR for this sensor is then compared with γ. If the SNR exceeds γ, the sensor is scheduled, its packet will be received and the total error covariance at the receiver will be $\bar{P}_i + \sum_{j \neq i} f_j\big(P_j(k-1)\big)$, where the ith sensor had the maximum channel gain. The cost of transmission will be μ since only one sensor is transmitting. Therefore the total cost for that time slot is

$$\bar{P}_i + \sum_{j \neq i} f_j\big(P_j(k-1)\big) + \mu. \tag{39}$$

If the sensor is not scheduled (even though it had the best channel gain), the total cost is

$$\sum_{i=1}^{N} f\big(P_i(k-1)\big). \tag{40}$$

We then decide whether the sensor with the best channel should be scheduled or not, depending on whether (39) is less than (40).

Maximum Covariance Policy: It is defined similarly to myopic policy, however at the kth time slot we schedule the sensor corresponding to the process that has maximum error covariance at the remote estimator. Then by using (39) and (40) we evaluate whether or not it is beneficial to schedule a sensor.

We perform 100 experiments each having the horizon length of $\mathcal{N} = 30$ corresponding to different values of μ. In Fig. 2, the average discounted total error-covariance vs. the average discounted total transmissions is shown. As expected, the curve is monotonically decreasing as more transmissions are allowed on average (i.e., μ is reduced). Fig. 3 shows the comparison among the exact DP, the myopic policy and the maximum covariance policy. To make the curves comparable in a single figure, we have reported the Logarithmic scale with base 10 of the average discounted total error covariance along the vertical axis. The error covariance corresponding to the DP is significantly less than its counterparts in the myopic and the maximum covariance policy (for the same μ). It can be noticed that myopic policy has the worst performance with respect to the others.

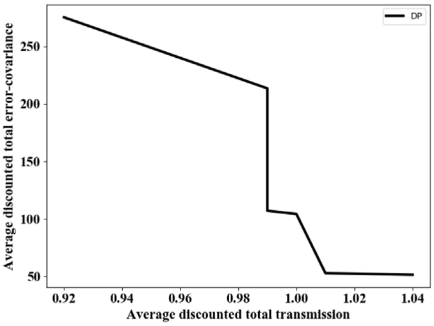

Fig. 2. Exact DP result, average discounted total error-covariance vs. discounted total number of transmissions in Example 1 corresponding to different values of μ (decreasing from left to right).

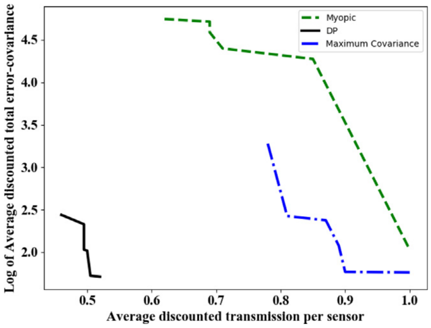

Fig. 3. Comparing the exact DP with myopic policy, and maximum covariance policy, average error-covariance vs. average discounted total transmissions per sensor in Example 1 corresponding to different values of μ (decreasing from left to right).

Example 2. Consider the systems 1,2,3; $\mathcal{N} = 30$; $q = 8$.

In this example, we compare the result of the exact DP algorithm with our EG-LSTD method and the maximum covariance policy. The maximum number of packet drops is $q = 8$, hence the cardinality of the state space is $\Omega = q^3 = 512$ (see (18)). For the EG-LSTD, we consider an arbitrary initial value $\bar{r}$ for the parameter vector, $\sigma = 0.2$, and $\Sigma = 0.2I$ (i.e., a diagonal matrix with 0.2 on its diagonal). We set $\mathcal{H} = 200$. Moreover, $\beta_i = \frac{1}{\Omega}$ for all i and $\alpha = 0.7$ to satisfy the condition $\beta_i \leq (1 - \alpha^2)$ required by Lemma 5.1 to guarantee the convergence. The cost function approximation $\tilde{J}^* = \Phi \hat{r}^*$ of the MSMP problem for different values of μ over an infinite time horizon is computed via the proposed EG-LSTD algorithm, and then used over the finite time horizon $\mathcal{N}$. More specifically, we can calculate the (stationary) EG-LSTD approximate optimal policy $\tilde{u}^*(\cdot)$ by replacing $\tilde{J}^*$ as the terminal cost function in the Bellman optimality operator

$$\tilde{u}^*(S) = \arg \min_{u(k) \in \mathcal{U}} \left[\mathcal{L}(S, u) + \alpha \sum_{S' \in \mathcal{S}} \mathcal{P}_{SS'}\big(u\big) \phi(S')^T \hat{r}^* \right]. \tag{41}$$

As for the exact DP, the expression (23) is recursively employed in order to compute the optimal policy $\pi^* = \{u^*(0), \dots, u^*(\mathcal{N}-1)\}$. A total number of 100 experiments over the finite time horizon $\mathcal{N}$ are performed by applying both the exact DP and the EG-LSTD policies. As we increase the value of the coefficient μ, we give more importance to the transmission (see (23)), therefore the expected cost increases accordingly. Fig. 4 shows the comparison of the average discounted total error-covariance vs. the average discounted total transmission for the exact DP and the EG-LSTD. As expected, all the curves are monotonically decreasing with respect to the discounted total number of transmissions. The

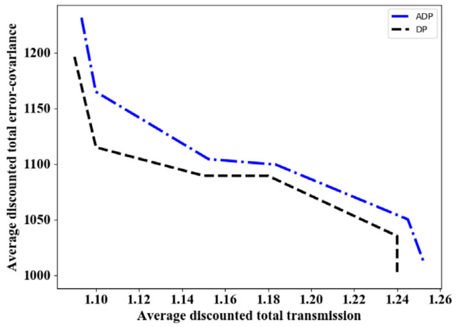

Fig. 4. Comparing the exact DP with the EG-LSTD, the average discounted error-covariance vs. average discounted total transmission in Example 2.

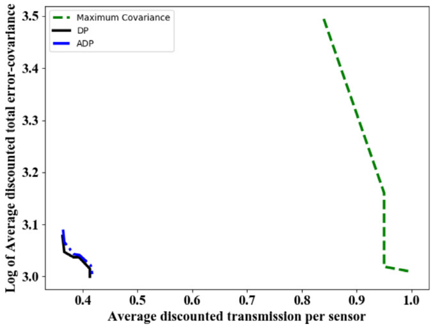

Fig. 5. Comparing the exact DP with the EG-LSTD and the maximum covariance, average discounted total error covariance vs. average discounted transmission per sensor Example 2.

DP policy always exhibits a lower average discounted total error covariance compared to the EG-LSTD policy. In particular for different values of μ, the average discounted total error covariance is plotted vs. the average discounted transmissions per sensor. For a fixed value of μ, the DP has the lowest average discounted total error-covariance and average discounted transmissions per sensor. In Fig. 5, the comparison of the exact DP, the EG-LSTD, and the maximum covariance is shown. Considering that the logarithmic values are reported on the vertical axis, the maximum covariance policy performs significantly worse with respect to the others.

Example 3. Consider the systems 1,2,3,4,5; $\mathcal{N} = 30$; $q = 10$.

The cardinality of the state space is $\Omega = q^5 = 10^5$, hence applying exact DP is not practical. We choose $\beta_i = \frac{1}{\Omega}$ for all i and $\alpha = 0.9$ to satisfy the condition $\beta_i \leq (1 - \alpha^2)$ required by Lemma 5.1 to guarantee the convergence. In this example, we consider different values for the parameter μ and we use the EG-LSTD to compute the corresponding parameter vector $\hat{r}^*$. An arbitrary initial value $\bar{r}$, $\sigma = 0.1$, and $\Sigma = 0.2I$ are set.

For $\mathcal{H} = 800$, they are

1. $\hat{r}^{*^T} = [235.7,\ 3.3,\ 3.6,\ 2.48,\ 1.9,\ 2.6]$ for $\mu = 0$
2. $\hat{r}^{*^T} = [987.2,\ 2.94,\ 3.3,\ 2.5,\ 1.9,\ 3.9]$ for $\mu = 2$
3. $\hat{r}^{*^T} = [1729.9,\ 2.8,\ 3.2,\ 2.5,\ 2.3,\ 3]$ for $\mu = 4$
4. $\hat{r}^{*^T} = [2482,\ 3,\ 3,\ 2.7,\ 2.1,\ 2.6]$ for $\mu = 6$
5. $\hat{r}^{*^T} = [3231,\ 3.5,\ 3.2,\ 2.5,\ 1.8,\ 3.4]$ for $\mu = 8$.

Norm-2 of the computed parameter vectors during the EG-LSTD algorithm iterations is shown in Fig. 6 for the selected values of μ. We make use of (41) for the computation of the EG-LSTD approximate optimal policy, which is then employed over the finite time horizon $\mathcal{N}$. Fig. 7 compares our proposed ADP method with the greedy maximum covariance policy. The average discounted total error-covariance vs. the average discounted transmission per sensor for different values of μ and $\hat{r}^*$ for both methods are shown. As we increase the parameter μ, the expected error covariance increases, while average discounted transmissions per sensor decreases. It is evident that EG-LSTD significantly outperforms the maximum covariance policy.

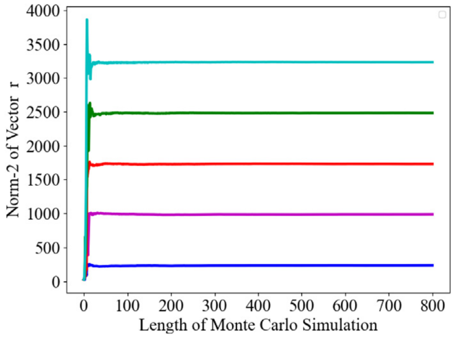

Fig. 6. Evolution of the Norm-2 of the computed parameter vectors within Monte Carlo simulation for Example 3: $\mu = 0$ (blue), $\mu = 2$ (violet), $\mu = 4$ (red), $\mu = 6$ (green), $\mu = 8$ (cyan). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

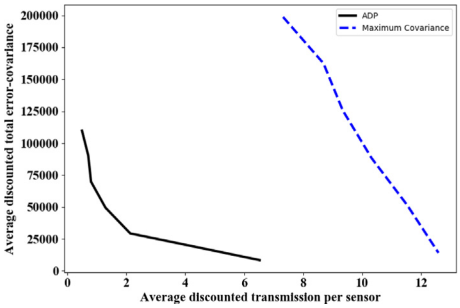

Fig. 7. EG-LSTD result and maximum covariance, average discounted total error covariance vs. average discounted total transmissions in Example 3.

7. Conclusion

A novel Least Squares Temporal Difference (LSTD) Approximate Dynamic Programming (ADP) based algorithm has been applied to the value function approximation in a Multi-Sensor Multi-Process (MSMP) transmission scheduling problem. This approach, named Enhanced-Exploration Greedy LSTD (EG-LSTD), adopts Monte Carlo simulations to generate state samples by using probability distributions different from the frequencies natural to the underlying Markov Decision Process (MDP). The convergence properties of the EG-LSTD algorithm have also been analysed and proved. Numerical simulations have been performed to verify the convergence of the EG-LSTD algorithm as well as its applicability to MSMP transmission scheduling problems of practical size, illustrating the efficacy of our method compared with the exact DP (for small number of states) and two different greedy suboptimal policies (for larger number of states). As for the future work we consider the scenarios where the control signals are present and can suffer from packet losses in the feedback link.

Appendix A. Proof of Lemma 5.1

$\forall J \in \mathbb{R}^n$ with $J \neq 0$, we have

$$\begin{aligned}
\|\alpha\mathcal{P}^*J\|_{\bar{\xi}}^2 &= \sum_{S\in\mathcal{S}} \bar{\xi}_S \Bigl(\sum_{S'\in\mathcal{S}} \alpha\mathcal{P}^*_{SS'}J(S')\Bigr)^2 \\
&= \alpha^2 \sum_{S\in\mathcal{S}} \bar{\xi}_S \Bigl(\sum_{S'\in\mathcal{S}} \mathcal{P}^*_{SS'}J(S')\Bigr)^2 \\
&\leq \alpha^2 \sum_{S\in\mathcal{S}} \bar{\xi}_S \sum_{S'\in\mathcal{S}} \mathcal{P}^*_{SS'}J^2(S') \leq \alpha^2 \sum_{S\in\mathcal{S}} \bar{\xi}_S \sum_{S'\in\mathcal{S}} \frac{\bar{\mathcal{P}}_{SS'}}{1-\beta_S}J^2(S') \\
&\leq \frac{\alpha^2}{1-\beta} \sum_{S'\in\mathcal{S}} \sum_{S\in\mathcal{S}} \bar{\xi}_S\bar{\mathcal{P}}_{SS'}J^2(S') = \bar{\alpha}^2 \sum_{S'\in\mathcal{S}} \bar{\xi}_{S'}J^2(S') = \bar{\alpha}^2\|J\|_{\bar{\xi}}^2,
\end{aligned}$$

where the first inequality follows from the convexity of the quadratic function, the second inequality follows from the fact $(1-\beta_S)\mathcal{P}^*_{SS'} \leq \bar{\mathcal{P}}_{SS'}$, and the next two last equalities follow from the property $\sum_{S\in\mathcal{S}} \bar{\xi}_S\bar{\mathcal{P}}_{SS'} = \bar{\xi}_{S'}$, of the invariant distribution and the definition of weighted Euclidean norm. Thus, $\alpha\mathcal{P}^*$ is a contraction with respect to $\|\cdot\|_{\bar{\xi}}$ with modulus at most $\bar{\alpha}$. The sufficient range of values for the exploration probabilities in order for $\bar{\Pi}\mathcal{F}^*$ to be a contraction is $\beta_S < 1-\alpha^2, \quad S \in \mathcal{S}$.

Appendix B. Proof of Lemma 5.2

The proof can be easily derived from

$$\begin{aligned}
\|J^* - \Phi r^*\|_{\bar{\xi}}^2 &= \|J^* - \bar{\Pi}J^*\|_{\bar{\xi}}^2 + \|\bar{\Pi}J^* - \Phi r^*\|_{\bar{\xi}}^2 \\
&= \|J^* - \bar{\Pi}J^*\|_{\bar{\xi}}^2 + \|\bar{\Pi}\mathcal{F}^*J^* - \bar{\Pi}\mathcal{F}^*(\Phi r^*)\|_{\bar{\xi}}^2 \\
&\leq \|J^* - \bar{\Pi}J^*\|_{\bar{\xi}}^2 + \bar{\alpha}^2\|J^* - \Phi r^*\|_{\bar{\xi}}^2
\end{aligned}$$

where the first equality uses the orthogonality of the projection, the second equality holds because J^* is the fixed point of $\mathcal{F}^*$ and Φr^* is the fixed point of $\bar{\Pi}\mathcal{F}^*$, and the inequality uses the contraction property of $\bar{\Pi}\mathcal{F}^*$.

Appendix C. Proof of Lemma 5.3

For any vector $J \in \mathbb{R}^{\Omega}, J \neq 0$, it can be written

$$\begin{aligned}
J^T\bar{\Xi}(I-\alpha\mathcal{P}^*)J &= \|J\|_{\bar{\xi}}^2 - \alpha J^T\bar{\Xi}\mathcal{P}^*J \geq \|J\|_{\bar{\xi}}^2 - \alpha\|J\|_{\bar{\xi}}\|\mathcal{P}^*J\|_{\bar{\xi}} \\
&\geq \|J\|_{\bar{\xi}}^2 - \bar{\alpha}\|J\|_{\bar{\xi}}\|J\|_{\bar{\xi}} = (1-\bar{\alpha})\|J\|_{\bar{\xi}}^2 > 0,
\end{aligned}$$

where for the second inequality we used Lemma 5.1, which implies that $\bar{\Xi}(I-\alpha\mathcal{P}^*)$ is positive definite.

Appendix D. Proof of Lemma 5.4

For all $r \in \mathbb{R}^m$, $r \neq 0$, we have

$$\|\alpha\bar{\Pi}\mathcal{P}^*\Phi r\|_{\bar{\xi}} \leq \|\alpha\mathcal{P}^*\Phi r\|_{\bar{\xi}} \leq \|\bar{\alpha}\Phi r\|_{\bar{\xi}}. \tag{D.1}$$

From properties of the orthogonal projection $\bar{\Pi}$ we have

$$\|\mathcal{P}^*\Phi r\|_{\bar{\xi}}^2 = \|\bar{\Pi}\mathcal{P}^*\Phi r\|_{\bar{\xi}}^2 + \|(I-\bar{\Pi})\mathcal{P}^*\Phi r\|_{\bar{\xi}}^2. \tag{D.2}$$

Moreover, from properties of projections, all vectors of the form Φr are orthogonal to all vectors of the form $J - \bar{\Pi}J$, i.e.

$$r^T\Phi^T\bar{\Xi}(I-\bar{\Pi})J = 0, \quad \forall r \in \mathbb{R}^m, \quad \forall J \in \mathbb{R}^{\Omega}. \tag{D.3}$$

Hence we have

$$\begin{aligned}
r^TCr &= r^T\Phi^T\bar{\Xi}(I-\alpha\mathcal{P}^*)\Phi r \\
&= r^T\Phi^T\bar{\Xi}\bigl(I-\alpha\bar{\Pi}\mathcal{P}^* + \alpha(\bar{\Pi}-I)\mathcal{P}^*\bigr)\Phi r \\
&= r^T\Phi^T\bar{\Xi}(I-\alpha\bar{\Pi}\mathcal{P}^*)\Phi r = \|\Phi r\|_{\bar{\xi}}^2 - \alpha r^T\Phi^T\bar{\Xi}\bar{\Pi}\mathcal{P}^*\Phi r \\
&\geq \|\Phi r\|_{\bar{\xi}}^2 - \alpha\|\Phi r\|_{\bar{\xi}}\|\bar{\Pi}\mathcal{P}^*\Phi r\|_{\bar{\xi}} \geq \|\Phi r\|_{\bar{\xi}}^2 - \bar{\alpha}\|\Phi r\|_{\bar{\xi}}\|\bar{\Pi}\mathcal{P}^*\Phi r\|_{\bar{\xi}} \\
&\geq \|\Phi r\|_{\bar{\xi}}^2 - \bar{\alpha}\|\Phi r\|_{\bar{\xi}}^2 \geq (1-\bar{\alpha})\|\Phi r\|_{\bar{\xi}}^2 > 0,
\end{aligned}$$

where the third equality follows from (D.3), the first inequality follows from the Cauchy Schwartz inequality, and the second inequality follows from (D.1).

Appendix E. Proof of Lemma 5.5

Obviously $\Phi^T\bar{\Xi}\Phi$ is symmetric and it is also positive definite since, for all $r \in \mathbb{R}^m$, $r \neq 0$, we have $r^T\Phi^T\bar{\Xi}\Phi r = \|\Phi r\|_{\bar{\xi}}^2 > 0$. As a result, $(\Phi^T\bar{\Xi}\Phi)^{-1}$ exists and it is symmetric positive definite, too.

Appendix F. Proof of Theorem 5.6

By following the same arguments in Lemma 5.3, we can simply prove that $\bar{\Xi}^{\frac{1}{2}}(I-\alpha\mathcal{P}^*)\bar{\Xi}^{\frac{1}{2}}$ is a positive definite matrix, which has the same eigenvalues of $\bar{\Xi}(I-\alpha\mathcal{P}^*)$. Similarly, $\bar{\Xi}^{\frac{1}{2}}(I-\alpha\mathcal{P}^{*^T})\bar{\Xi}^{\frac{1}{2}}$ is positive definite, and it has the same eigenvalues of $\bar{\Xi}(I-\alpha\mathcal{P}^*)$. In this regard we can say

$$\bar{\Xi}^{\frac{1}{2}}\Bigl(2I - \alpha(\mathcal{P}^* + \mathcal{P}^{*^T})\Bigr)\bar{\Xi}^{\frac{1}{2}} > 0, \tag{F.1}$$

and the eigenvalues of (F.1) are the sum of the eigenvalues of $\bar{\Xi}(I-\alpha\mathcal{P}^*)$ and of $\bar{\Xi}(I-\alpha\mathcal{P}^{*^T})$, they are real and positive. The conclusion is that the eigenvalues of $\Phi^T\bar{\Xi}(I-\alpha\mathcal{P}^*)\Phi$ have positive real parts. From the other side the eigenvalues of $\Phi^T\bar{\Xi}\Phi$ are real, and positive (see Lemma 5.5). From Assumptions 1 and 2, it is

$$\begin{aligned}
\det(G) &= \det(\Phi^T\bar{\Xi}\Phi) = \det(\Phi^T\Phi)\det(\bar{\Xi}) \\
\det(C) &= \det\bigl(\Phi^T\bar{\Xi}(I-\alpha\mathcal{P}^*)\Phi\bigr) = \det(\Phi^T\Phi)\det\bigl(\bar{\Xi}(I-\alpha\mathcal{P}^*)\bigr) \\
\det\bigl(G^{-1}C\bigr) &= \det(G^{-1})\det(C) = \frac{\det(C)}{\det(G)} \\
&= \frac{\det(\Phi^T\Phi)\det\bigl(\bar{\Xi}(I-\alpha\mathcal{P})\bigr)}{\det(\Phi^T\Phi)\det(\bar{\Xi})} = \frac{\det(\bar{\Xi})\det(I-\alpha\mathcal{P}^*)}{\det(\bar{\Xi})} \\
&= \det(I-\alpha\mathcal{P}^*) < 1,
\end{aligned}$$

where the eigenvalues of $\alpha\mathcal{P}^*$ are strictly within the unit circle. To show the last claim we construct $I - G^{-1}C$

$$\begin{aligned}
I - G^{-1}C &= I - \bigl(\Phi^T\bar{\Xi}\Phi\bigr)^{-1}\bigl(\Phi^T\bar{\Xi}\Phi - \alpha\Phi^T\bar{\Xi}\mathcal{P}^*\Phi\bigr) \\
&= I - \bigl(\Phi^T\bar{\Xi}\Phi\bigr)^{-1}\bigl(\Phi^T\bar{\Xi}\Phi\bigr) + \bigl(\Phi^T\bar{\Xi}\Phi\bigr)^{-1}\bigl(\alpha\Phi^T\bar{\Xi}\mathcal{P}^*\Phi\bigr) \\
&= I - I + \bigl(\Phi^T\bar{\Xi}\Phi\bigr)^{-1}\bigl(\alpha\Phi^T\bar{\Xi}\mathcal{P}^*\Phi\bigr) \\
&= \bigl(\Phi^T\bar{\Xi}\Phi\bigr)^{-1}\bigl(\alpha\Phi^T\bar{\Xi}\mathcal{P}^*\Phi\bigr),
\end{aligned}$$

since $C > 0$, then we have $\alpha\Phi^T\bar{\Xi}\mathcal{P}^*\Phi < \Phi^T\bar{\Xi}\Phi$, hence it is

$$I - G^{-1}C = \bigl(\Phi^T\bar{\Xi}\Phi\bigr)^{-1}\bigl(\alpha\Phi^T\bar{\Xi}\mathcal{P}^*\Phi\bigr) < \bigl(\Phi^T\bar{\Xi}\Phi\bigr)^{-1}\bigl(\Phi^T\bar{\Xi}\Phi\bigr) = I, \tag{F.2}$$

therefore the spectral radius $\rho(I - G^{-1}C) < 1$ which implies that eigenvalues of $I - G^{-1}C$ are strictly within unit circle. From (F.2) we have $I - G^{-1}C < I \rightarrow G^{-1}C > 0$, i.e., $G^{-1}C$ is positive definite.

Appendix G. Proof of Lemma 5.7

Once again consider Eq. (36), one choice is to assume $G^{-1} = (C^T\Sigma^{-1}C + \sigma I)^{-1}C^T\Sigma^{-1}$ then we have

$$\hat{r}_{l+1} = \hat{r}_l - (C^T\Sigma^{-1}C + \sigma I)^{-1}(C^T\Sigma^{-1}C\hat{r}_l - C^T\Sigma^{-1}d), \tag{G.1}$$

by expanding and adding/subtracting $\sigma\hat{r}_l$ in the second parenthesis we have

$$\begin{aligned}\hat{r}_{l+1} &= \hat{r}_l - (C^T\Sigma^{-1}C + \sigma r_l)^{-1}(C^T\Sigma^{-1}C\hat{r}_l + \sigma\hat{r}_l - \sigma\hat{r}_l - C^T\Sigma^{-1}d)\\ &= \Big(I - (C^T\Sigma^{-1}C + \sigma I)^{-1}(C^T\Sigma^{-1}C + \sigma I)\Big)\hat{r}_l\\ &+ (C^T\Sigma^{-1}C + \sigma I)^{-1}(C^T\Sigma^{-1}d + \sigma\hat{r}_l)\\ &= (C^T\Sigma^{-1}C + \sigma I)^{-1}(C^T\Sigma^{-1}d + \sigma\hat{r}_l),\end{aligned} \tag{G.2}$$

which is the same as (38) except that matrices are not dependent on iteration steps. To prove that recursive iteration (G.2) is convergent it is enough to show that the eigenvalues of $I - G^{-1}C = I - (C^T\Sigma^{-1}C + \sigma I)^{-1}C^T\Sigma^{-1}C$ are strictly within unit circle (see (G.1)). To see this let $\lambda_1, \dots, \lambda_m$ be the eigenvalues of $C^T\Sigma^{-1}C$ and let $U\Lambda U^T$ be its singular value decomposition, where $\Lambda = \text{diag}\{\lambda_1, \dots, \lambda_m\}$ and U is a unitary matrix ($UU^T = I$). It is $(C^T\Sigma^{-1}C + \sigma I) = U(\Lambda + \sigma I)U^T$, so $G^{-1}C = \Big(U(\Lambda + \sigma I)U^T\Big)^{-1}U\Lambda U^T = U(\Lambda + \sigma I)^{-1}\Lambda U^T$. The eigenvalues of $G^{-1}C$ are then $\frac{\lambda_i}{(\lambda_i + \sigma)}$, $i = 1, \dots, m$, and lie in the interval $(0, 1)$ which implies that the eigenvalues of $I - G^{-1}C$ also lie in the unit circle and the proof is complete.

References

Bertsekas, D. P. (2011). Temporal difference methods for general projected equations. *IEEE Transactions on Automatic Control*, *56*(9), 2128–2139.

Bertsekas, D. P. (2012). *Dynamic programming and optimal control, vol. II*,. MA, Belmont, Massachusetts, USA: Athena Scientific.

Bertsekas, D. P., & Yu, H. (2009). Projected equation methods for approximate solution of large linear systems. *Journal of Computational and Applied Mathematics*, *227*, 27–50.

Busoniu, L., Ernst, D., De Schutter, B., & Babuska, R. (2011). Cross-entropy optimization of control policies with adaptive basis functions. *IEEE Transaction on Systems, Man, and Cybernetics - Part B: Cybernetics*, *41*(1), 196–209.

De Farias, D. P., & Van Roy, B. (2003). The linear programming approach to approximate dynamic programming. *Operations Research*, *51*(6), 850–865.

Deng, K., Chen, Y., & Belta, C. (2017). An approximate dynamic programming approach to multi-agent persistent monitoring in stochastic environments with temporal logic constraints. *IEEE Transactions on Automatic Control*, *62*(9), 4549–4563.

Forootani, A., Iervolino, R., & Tipaldi, M. (2019). Applying unweighted least-squares based techniques to stochastic dynamic programming: theory and application. *IET Control Theory & Applications*, *13*(15), 2387–2398.

Forootani, A., Iervolino, R., Tipaldi, M., & Neilson, J. (2020). Approximate dynamic programming for stochastic resource allocation problems. *IEEE/CAA Journal of Automatica Sinica*, *7*(4), 975–990.

Forootani, A., Liuzza, D., Tipaldi, M., & Glielmo, L. (2019). Allocating resources via price management systems: a dynamic programming-based approach. *International Journal of Control*.

Forootani, A., Tipaldi, M., Ghaniee Zarch, M., Liuzza, D., & Glielmo, L. (2020a). A least-squares temporal difference based method for solving resource allocation problems. *IFAC Journal of Systems and Control*, *13*, 1–15.

Forootani, A., Tipaldi, M., Ghaniee Zarch, M., Liuzza, D., & Glielmo, L. (2020b). Modelling and solving resource allocation problems via a dynamic programming approach. *International Journal of Control*, *94*(6), 1544–1555.

Forootani, A., Tipaldi, M., Iervolino, R., & Dey, S. (2022). Enhanced exploration least-squares methods for optimal stopping problems. *IEEE Control System Letter*, *6*, 271–276.

Gatsis, K., Ribeiro, A., & Pappas, G. J. (2018). Random access design for wireless control systems. *Automatica*, *91*, 1–9.

Geist, M., & Pietquin, O. (2013). Algorithmic survey of parametric value function approximation. *IEEE Transactions on Neural Networks and Learning Systems*, *24*(6), 845–867.

Gupta, V., Chung, T. H., Hassibi, B., & Murray, R. M. (2006). On a stochastic sensor selection algorithm with applications in sensor scheduling and sensor coverage. *Automatica*, *42*(2), 251–260.

Gupta, V., Hassibi, B., & Murray, R. M. (2007). Optimal LQG control across packet-dropping links. *Systems & Control Letters*, *56*(6), 439–446.

Han, D., Wu, J., Zhang, H., & Shi, L. (2017). Optimal sensor scheduling for multiple linear dynamical systems. *Automatica*, *75*, 260–270.

Hoffman, M. W., Lazaric, A., Ghavamzadeh, M., & Munos, R. Regularized least squares temporal difference learning with nested l_2 and l_1 penalization. In *European Workshop on Reinforcement Learning* (pp. 102-114).

Leong, A. S., Dey, S., & Quevedo, D. E. (2017). Sensor scheduling in variance based event triggered estimation with packet drops. *IEEE Transactions on Automatic Control*, *62*(4), 1880–1895.

Leong, A. S., Quevedo, D. E., Tanaka, T., Dey, S., & Ahlen, A. (2017). Event-based transmission scheduling and LQG control over a packet dropping link. *IFAC-PapersOnLine*, *50*(1), 8945–8950.

Leong, A. S., Ramaswamy, A., Quevedo, D. E., Karl, H., & Shi, L. (2020). Deep reinforcement learning for wireless sensor scheduling in cyber–physical systems. *Automatica*, *113*, Article 108759.

Li, Y., Chen, C. S., & Wong, W. S. (2019). Power control for multi-sensor remote state estimation over interference channel. *Systems & Control Letters*, *126*.

Liu, W., Quevedo, D. E., Johansson, K. H., Vucetic, B., & Li, Y. (2021). Remote state estimation of multiple systems over multiple Markov fading channels. arXiv preprint arXiv:2104.04181.

Nedic, A., & Bertsekas, D. P. (2003). Least squares policy evaluation algorithms with linear function approximation. *Discrete Event Dynamic Systems*, *13*, 79–110.

Nourian, M., Leong, A. S., & Dey, S. (2014). Optimal energy allocation for Kalman filtering over packet dropping links with imperfect acknowledgments and energy harvesting constraints. *IEEE Transactions on Automatic Control*, *59*(8), 2128–2143.

Papandriopoulos, J., Evans, J. S., & Dey, S. (2005). Optimal power control for Rayleigh-faded multiuser systems with outage constraints. *IEEE Transactions on Wireless Communication*, *4*(6), 2705–2715.

Perez-Neira, A. I., & Campalans, M. R. (2010). *Cross-layer resource allocation in wireless communications: Techniques and models from PHY and MAC layer interaction*. Academic Press.

Pezzutto, M., Schenato, L., & Dey, S. (2020). Transmission scheduling for remote estimation with multi-packet reception under multi-sensor interference. In *21st proceedings of IFAC world congress*.

Pezzutto, M., Schenato, L., & Dey, S. (2021a). Transmission power allocation for remote estimation with multi-packet reception capabilities. arXiv preprint arXiv:2101.12493.

Pezzutto, M., Schenato, L., & Dey, S. (2021b). Transmission power allocation for remote estimation with multi-packet reception capabilities. arXiv preprint arXiv:2101.12493.

Ren, X., Wu, J., Dey, S., & Shi, L. (2018). Attack allocation on remote state estimation in multi-systems: Structural results and asymptotic solution. *Automatica*, *87*.

Ren, X., Wu, J., Johansson, K. H., Shi, G., & Shi, L. (2018). Infinite horizon optimal transmission power control for remote state estimation over fading channels. *IEEE Transactions on Automatic Control*, *63*(1), 85–100.

Schenato, L. (2008). Optimal estimation in networked control systems subject to random delay and packet drop. *IEEE Transactions on Automatic Control*, *53*(5), 1311–1317.

Schenato, L., Sinopoli, B., Franceschetti, M., Poolla, K., & Sastry, S. S. (2007). Foundations of control and estimation over lossy networks. *Proceedings of the IEEE*, *95*(1), 163–187.

Shi, L., Cheng, P., & Chen, J. (2011a). Optimal periodic sensor scheduling with limited resources. *IEEE Transactions on Automatic Control*, *56*(9), 2190–2195.

Shi, L., Cheng, P., & Chen, J. (2011b). Sensor data scheduling for optimal state estimation with communication energy constraint. *Automatica*, *47*(8), 1693–1698.

Tsitsiklis, J. N., & Van Roy, B. (1997). An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control*, *42*(5).

Wang, M., Polydorides, N., & Bertsekas, D. P. (2009). *Approximate simulation-based solution of large-scale least squares problems*: *Lab. Information and Decision Systems Report LIDS-P-2819*, MIT.

Wu, J., Jia, Q. S., Johansson, K. H., & Shi, L. (2013). Event-based sensor data scheduling: Trade-off between communication rate and estimation quality. *IEEE Transactions on Automatic Control*, *26*(4).

Wu, S., Ren, X., Dey, S., & Shi, L. (2018). Optimal scheduling of multiple sensors over shared channels with packet transmission constraint. *Automatica*, *96*, 22–31.

Xu, Y., & Hespanha, J. P. (2005). Estimation under uncontrolled and controlled communications in networked control systems. In *Proceedings of the 44th IEEE conference on decision and control* (pp. 842-847).

Zanella, A., & Zorzi, M. (2012). Theoretical analysis of the capture probability in wireless systems with multiple packet reception capabilities. *IEEE Transactions on Communications*, *60*(4), 1058–1071.

Zhao, L., Zhang, W., Hu, J., Abate, A., & Tomlin, C. J. (2014). On the optimal solutions of the infinite-horizon linear sensor scheduling problem. *IEEE Transactions on Automatic Control*, *59*(10), 825–2830.

Ali Forootani received the M.Sc. degree in electrical engineering and automatic control system from Power and Water University of Technology, Iran, in 2011, and the Ph.D. degree in automatic control system and information technology from Department of Engineering, University of Sannio, Italy, in 2019. From 2011 to 2015 he worked both on research and industry at Niroo Research Institute (Tehran, Iran) and at the Ministry of Power and Energy (Water and Sewage khuzestan Engineering Company). From 2019 to 2020 he served as Postdoctoral researcher at the Measurement and Instrumentation Laboratory University of Sannio, as well as University of Salerno on the topics of drone image signal processing and AI based network disease analysis. He is currently with the Hamilton Institute, Maynooth University, Ireland. His current research interests include Markov decision processes, approximate dynamic programming, reinforcement learning in optimal control, and learning in network control systems. He is a Member of IEEE Control System Society and his papers were considered as the selected publications on International Journal of Control and IEEE Transaction of Automatica Sinica.

Raffaele Iervolino received the laurea degree cum laude in aerospace engineering from the University of Naples, Italy, in 1996, where he also obtained the Ph.D. degree in electronic and computer science engineering in 2002. Since 2003 he is an Assistant Professor of automatic control at the University of Naples. From 2005 he is also an Adjoint Professor of automatic control with the Department of Electrical Engineering and Information Technology at the same University. He is Senior Member of IEEE Control System Society. His research interests include piecewise affine systems, opinion dynamics and consensus in social networks, and human telemetry systems.

Massimo Tipaldi received the master degree in computer science engineering and the Ph.D. degree in information technology from the University of Sannio, Italy, in 1998 and 2017, respectively. He possesses more than 20 years of industrial experience in the managerial and technical coordination of ESA/ASI/CNES space project (satellite systems, experimental equipment for the International Space Station, and ground segments). He holds 1 patent and has co-authored more than 40 papers published in proceedings of international conferences or international archival journals. His research interests include space systems engineering, critical SW systems, reinforcement learning, approximate dynamic programming, stochastic systems, and advanced system control techniques.

Subhrakanti Dey received the Bachelor in Technology and Master in Technology degrees from the Department of Electronics and Electrical Communication Engineering, Indian Institute of Technology, Kharagpur, in 1991 and 1993, respectively, and the Ph.D. degree from the Department of Systems Engineering, Research School of Information Sciences and Engineering, Australian National University, Canberra, in 1996. He is currently a Professor with the Dept of Electronic Engineering, National University of Ireland, Maynooth, Ireland. Prior to this, he was a Professor with the Dept. of Engineering Sciences in Uppsala University, Sweden (2013–2017), Professor with the Department of Electrical and Electronic Engineering, University of Melbourne, Parkville, Australia, from 2000 until early 2013, and a Professor of Telecommunications at University of South Australia during 2017–2018. From September 1995 to September 1997, and September 1998 to February 2000, he was a Postdoctoral Research Fellow with the Department of Systems Engineering, Australian National University. From September 1997 to September 1998, he was a Postdoctoral Research Associate with the Institute for Systems Research, University of Maryland, College Park. His current research interests include wireless communications and networks, signal processing for sensor networks, networked control systems, and distributed machine learning. Professor Dey currently serves as a Senior Editor on the Editorial Board *IEEE Transactions on Control of Network Systems*, and as an Associate Editor/Editor for Automatica, *IEEE Control Systems Letters*, and *IEEE Transactions on Wireless Communications*. He was also an Associate Editor for *IEEE* and *Transactions on Signal Processing*, (2007–2010, 2014–2018), *IEEE Transactions on Automatic Control* (2004–2007), and *Elsevier Systems and Control Letters* (2003–2013).