Inverse Filtering for Hidden Markov Models With Applications to Counter-Adversarial Autonomous Systems

Robert Mattila, *Student Member, IEEE*, Cristian R. Rojas, *Member, IEEE*, Vikram Krishnamurthy, *Fellow, IEEE*, and Bo Wahlberg, *Fellow, IEEE*

***Abstract*—Bayesian filtering deals with computing the posterior distribution of the state of a stochastic dynamic system given noisy observations. In this paper, motivated by applications in counter-adversarial autonomous systems, we consider the following inverse filtering problem: Given a sequence of posterior distributions from a Bayesian filter, what can be inferred about the transition kernel of the state, the observation likelihoods of the sensor and the measured observations? For finite-state Markov chains observed in noise (hidden Markov models), we show that a least-squares fit for estimating the parameters and observations amounts to a combinatorial optimization problem with non-convex objective. Instead, by exploiting the algebraic structure of the corresponding Bayesian filter, we propose an algorithm based on convex optimization for reconstructing the transition kernel, the observation likelihoods and the observations. We discuss and derive conditions for identifiability. As an application of our results, we demonstrate the design of a counter-adversarial autonomous system: By observing the actions of an autonomous enemy, we estimate the accuracy of its sensors and the observations it has received. The proposed algorithms are illustrated via several numerical examples.**

***Index Terms*—Bayesian filtering, inverse filtering, hidden Markov models, counter-adversarial autonomous systems, unique identifiability, group LASSO, nullspace clustering, adversarial signal processing.**

I. Introduction

In a partially observed stochastic dynamic system, the state is hidden in the sense that it can only be observed in noise via a sensor. Formally, with p denoting a generic probability density (or mass) function, such a system is represented by the conditional densities:

$$x_k \sim P_{x_{k-1},x} = p(x|x_{k-1}), \quad x_0 \sim \pi_0, \tag{1}$$

$$y_k \sim B_{x_k,y} = p(y|x_k), \tag{2}$$

where by $\sim$ we mean "distributed according to" and $k = 0, 1, \ldots$ denotes discrete time. In (1), the state x_k evolves according to a Markovian transition kernel P on state-space $\mathcal{X}$, and π_0 is its initial distribution. In (2), an observation y_k (in observation-space $\mathcal{Y}$) of the state is measured at each time instant according to observation likelihoods B. An important example of (1)–(2), where (1) is a finite-state Markov chain, is the so called *hidden Markov model* (HMM) [1], [2].

In the Bayesian (stochastic) filtering problem [3], one seeks to compute the conditional expectation of the state given noisy observations by evaluating a recursive expression for the posterior distribution of the state:

$$\pi_k(x) = p(x_k = x|y_1, \ldots, y_k), \qquad x \in \mathcal{X}. \tag{3}$$

The recursion for the posterior is given by the Bayesian filter

$$\pi_k = T(\pi_{k-1}, y_k; P, B), \tag{4}$$

where the operator T for the Bayesian filtering update (4) is

$$\{T(\pi, y; P, B)\}(x) = \frac{B_{x,y} \int_{\mathcal{X}} P_{\zeta,x} \pi(\zeta) d\zeta}{\int_{\mathcal{X}} B_{x',y} \int_{\mathcal{X}} P_{\zeta',x'} \pi(\zeta') d\zeta' dx'}, \quad x \in \mathcal{X}, \tag{5}$$

– see, e.g., [1], [2] for derivations and details. Two well known finite dimensional cases of (5) are the Kalman filter, where the dynamical system (1)–(2) is a linear Gaussian state-space model, and the HMM filter, where the system (1) is a finite-state Markov chain.

In this paper, we formulate and provide solutions to the following inverse filtering problem:

> *Given a sequence of posteriors $\pi_1, \ldots, \pi_N$ from the filter (4), reconstruct (estimate) the filter's parameters, namely, the system's transition kernel P, the sensor's observation likelihoods B and the measured observations $y_1, \ldots, y_N$.*

A. Motivation

In this section, we outline various motivating applications where the inverse filtering problem is of relevance.

Manuscript received January 31, 2020; revised July 6, 2020; accepted August 9, 2020. Date of publication August 24, 2020; date of current version September 14, 2020. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. V. Raghavan. This work was supported in part by the Swedish Research Council (NewLEADS, 2016-06079, 2018-03438), in part by the Wallenberg AI, Autonomous Systems and Software Program (WASP), in part by the U.S. Air Force Office of Scientific Research (FA9550-18-1-0007), in part by the U.S. Army Research Office (W911NF-19-1-0365), and in part by the National Science Foundation (1714180). *(Corresponding author: Robert Mattila.)*

Robert Mattila, Cristian R. Rojas, and Bo Wahlberg are with the Division of Decision and Control Systems, School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, 100 44 Stockholm, Sweden (e-mail: rmattila@kth.se; crro@kth.se; bo@kth.se).

Vikram Krishnamurthy is with the School of Electrical and Computer Engineering, Cornell University, Ithaca, NY 14853 USA (e-mail: vikramk@cornell.edu).

Digital Object Identifier 10.1109/TSP.2020.3019177

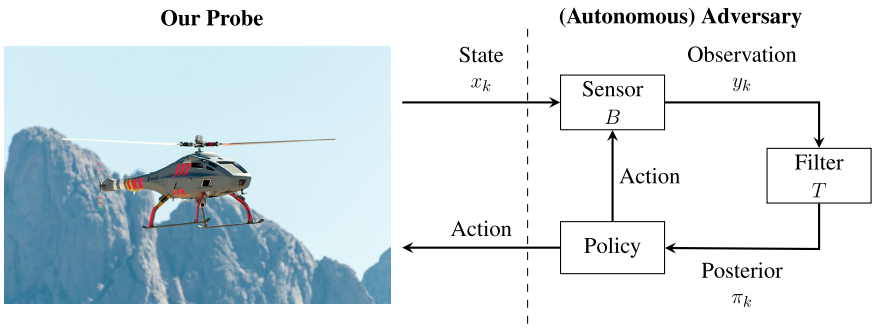

Fig. 1. We employ a drone to probe a sophisticated autonomous adversary that: i) measures our state and filters the data, ii) ranks different actions via a policy, and iii) automatically executes its choice. We aim to devise a counter-adversarial autonomous system that can remotely estimate the accuracy of the adversary's sensor (see Problem 3 in Section IV). This forms a basis for predicting, and taking measures against, its future actions. (*Photo: UMS SKELDAR V-200 / UMS AERO Group and Saab.*)

1) Fault Detection: Methods for model-based fault-detection study the discrepancy between the observed output y_k of a system and that predicted by a model. The simplest tests flag that a fault has occurred if the instantaneous value of the residual is above some predetermined threshold, but more sophisticated algorithms consider also the statistics and trends in the residuals [4].

It can, however, be difficult (or even impossible) to access raw sensor data in integrated smart sensors since they are often tightly encapsulated. Without access to the observations y_k, the residuals cannot be computed. For example, in robotics applications, the pose (position and orientation) is an important quantity that has to be estimated. The pose can indirectly be measured from several different sensor systems such as odometry, computer vision, sonar and laser. The output from these so-called pose providers are often limited to state estimates π_k [5], [6] via a filter (5) – the raw observations y_k can be difficult to extract.

Hence, a relevant question for fault-detection of such sensor systems is: Can the residuals be reconstructed from state estimates π_k? Solutions to the inverse filtering problem posed above – the ability to reconstruct the observations y_k measured by a filtering system from only its output π_k – offers an affirmative answer to this question, and therefore lays a foundation for the application of model-based fault-detection algorithms.

2) Counter-Adversarial Autonomous Systems: Our main motivating application is the design of counter-adversarial autonomous systems [7]–[10]: Given measurements of the actions of a sophisticated autonomous adversary, how can we estimate information private to the adversary (e.g., the accuracy of its sensors) and use such insights to devise counter-measures (e.g., predict and guard against future actions)?

As a concrete example, consider an adversarial scenario where we control a drone or an *unmanned aerial vehicle* (UAV).[1] The adversary employs a radar system to measure our kinematic state x_k. Its noisy measurements y_k are processed by a filtering system T via (5) to produce a posterior distribution π_k. Based on its posterior π_k and a policy, an action is taken which we can measure or infer – for example, with a cognitive radar, the adversary's response can be to have its resource manager adjust, e.g., waveform, beam orientation and aperture [11]. Based on such actions, we aim to estimate the adversary's sensor B. See Fig. 1 for a schematic illustration.

The inverse filtering problem arises as a subproblem: Given the adversary's posteriors π_k, reconstruct (estimate) its sensor B. (The complementary subproblem of estimating the adversary's posteriors from its actions is treated in [8], [9], [12], and is discussed in Section IV-C.)

3) Transfer Learning: Many machine learning algorithms assume that training and test data are drawn from the same distribution. If conditions change, the statistical model needs to be re-estimated from scratch from newly collected training data. In many real-world applications, this can be expensive, in terms of data collection (e.g., clinical trials) and computation. Transfer learning addresses this problem by allowing knowledge gained while solving one problem to be reused and applied to a different, but related problem – see, e.g., [13]–[15].

In the context of reinforcement learning [16], in many practical cases, there exists an agent interacting with an environment that usually has acceptable, but suboptimal performance, and it is requested to replace such an agent by an automatic system with comparable or superior performance [14], [15]. For example, this is the ambition of replacing human drivers by self-driving vehicles.

The inverse filtering problem appears as a means to extract the internal model (P, B) of a filtering-based agent [10, Sec. 1.1.2]. The extracted model can then serve as a basis (i.e., as a transfer of knowledge) for designing an automatic substitute system. In comparison to constructing such an automatic system from scratch, by extracting ("reverse engineering") the model of the existing agent, one can effectively warm-start the model calibration phase, yielding accelerated and safe learning[14], [15], [17].

4) Cyber-Physical Security: Consider a malicious actor performing a stealth attack on a cyber-physical control system. The attacker (for reasons of, e.g., sabotage, financial gain or terrorism) injects a malicious signal while aiming to avoid detection. For example, this could amount to counterfeiting the sequence of posteriors, or modifying the system and sensor parameters while presenting the expected ones to the operator – the Stuxnet cyberweapon is a motivating real-world example [18].

Revealing and alleviating the consequences of such attacks have received increasing attention during the last decade[19]–[21]. Solutions to the inverse filtering problem could function as anomaly detectors that allow the operator to infer inconsistencies

[1]More generally, we control a probe signal x_k that can be of electromagnetic, cyber or physical nature.

between the estimates π_k produced by a filtering-based system and its parameters (and, hence, detect the attack).

B. Main Results and Outline

To construct a tractable analysis, we consider the case where (1)–(2) constitute a *hidden Markov model* (HMM) on a finite observation-alphabet. The main results of this paper are:

- We analyze the uniqueness of the updates of the Bayesian filter (5) for HMMs (Theorems 1 and 2), and derive an alternative characterization (Theorem 3) that highlights important structural properties.
- We introduce the nullspace clustering problem (Problem 2 in Section III) and propose an algorithm based on the group LASSO [22], [23] to solve it. In Theorem 4, we detail a procedure to uniquely factorize unnormalized nullspaces into HMM parameters.
- By leveraging the previous two points we demonstrate how the transition kernel as well as the observation likelihoods of an HMM can be reconstructed from a sequence of posteriors (Algorithm 1); then, the corresponding sequence of observations can be reconstructed (Remark 5).
- We apply our results to the remote sensor calibration problem (Problem 3 in Section IV) for counter-adversarial autonomous systems. Even in a mismatched setting where the adversary employs uncertain estimates $\hat{P}$ and $\hat{B}$ in its filter, we can estimate the sensor of the adversary and, surprisingly, we can do so regardless of the quality of its estimates $\hat{P}$ and $\hat{B}$.
- Finally, the performance of our proposed inverse filtering algorithms is demonstrated in numerical experiments.

The paper is structured as follows. Section II formulates the problems we consider, discusses identifiability, and shows that a direct approach is computationally infeasible for large data sizes. Our proposed inverse filtering algorithms are given in Section III. In Section IV, we consider the design of counter-adversarial autonomous systems and show how an adversary's sensor can be estimated from its actions. In Section V, the proposed algorithms are illustrated via numerical experiments. Detailed proofs and algebraic manipulations are available in the appendices.

C. Related Work

Kalman's inverse optimal control paper [24] from 1964, aiming to determine for what cost criteria a given control policy is optimal, is an early example of an inverse problem in signal processing and automatic control. More recently, an interest for similar problems has been sparked in the machine learning community with the success of topics such as inverse reinforcement learning, imitation learning and apprenticeship learning [25]–[29]. In essence, these works aim to determine the cost function or policy employed by an expert agent. In contrast, in the inverse filtering problem treated in this paper, we target the system and sensor parameters together with the sample path of observations given posterior distributions from an HMM filter.

Variations of inverse filtering problems can be found in the microeconomics literature (social learning and revealed preferences; [30], [31]) and the fault detection literature (e.g., [4], [32]–[34]), where the stochastic filter is a standard tool.

In relation to the discussion in Section I-A1 on fault detection, in [5], [6], an extended observer was used to reconstruct the residuals from state estimates in a linear state-space model. The work [35] derives a similar solution for discrete systems, but where instead of state estimates, it is assumed that only part of the system's output can be observed. In contrast, i) we consider HMMs and work with the posterior distributions from an HMM filter, and ii) we exploit the structure of the HMM filter directly to reconstruct the sample path of observations.

To the best of the authors' knowledge, the specific inverse filtering problem we consider – reconstructing system and sensor parameters directly from posteriors – was first introduced in [36] for HMMs, and later discussed for linear Gaussian state-space models in [37]. However, in these papers, strong simplifying assumptions were made. It was assumed that i) the transition kernel P of the system was known, and that ii) the system and the filter were matched in the sense that the update $T(\pi_{k-1}, y_k; P, B)$ was used in (4) and not the more realistic, mismatched, $T(\pi_{k-1}, y_k; \hat{P}, \hat{B})$, where $\hat{P}$ and $\hat{B}$ denote estimates. This paper disposes both of these assumptions.

The latter is of crucial importance when applying inverse filtering algorithms in counter-adversarial scenarios. Recall from Section I-A2 that in such, an adversary is trying to estimate our state (via Bayesian filtering). Previous works, [8], [9], [36], assume that both we and the enemy know the transition kernel P. In reality, if we generate the signal x_k, then the enemy estimates P (e.g., maximum likelihood estimate) and employs a mismatched filtering system; hence, we have to *estimate the enemy's estimate* of P. The first part of this paper constructs algorithms for doing this based on observing (intercepting) posterior distributions. In the second part of the paper, we consider the general setting where only actions based on the enemy's posteriors are observed.

Lastly, the previous works [8], [9], [36], [37] do not address general identifiability issues in inverse filtering. The current paper gives necessary and sufficient conditions for identifiability of P and B given a sequence of posteriors.

II. Preliminaries and Problem Formulation

In this section, we first detail our notation and provide necessary background material on hidden Markov models and their corresponding Bayesian filter. We then formally state the problem we consider and discuss the uniqueness of its solution. Finally, we outline a "direct" approach to the problem and point to potential computational concerns.

A. Notation

All vectors are column vectors unless transposed. The vector of all ones is denoted $\mathbb{1}$ and the ith Cartesian basis vector e_i. The element at row i and column j of a matrix is $[\cdot]_{ij}$, and the element at position i of a vector is $[\cdot]_i$. The Kronecker product is denoted with $\otimes$. The vector operator $\operatorname{diag}(\cdot) : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ gives the matrix where the vector has been put on the diagonal, and all other elements are zero. The operator $\operatorname{vec}(\cdot) : \mathbb{R}^{m\times n} \to \mathbb{R}^{mn}$ converts a matrix into a column vector by stacking the columns of the matrix on top of another. The indicator function $\mathrm{I}\{\cdot\}$ takes the value 1 if the expression $\cdot$ is fulfilled and 0 otherwise. The unit simplex is denoted as Δ. The nullspace of a matrix is $\ker \cdot$, and $\cdot^\dagger$ denotes pseudo-inverse.

B. Hidden Markov Models

We refer to a partially observed stochastic dynamical model (1)–(2) whose state-space $\mathcal{X} = \{1, \ldots, X\}$ is discrete as a *hidden Markov model* (HMM). We limit ourselves to HMMs with observation processes on a finite alphabet $\mathcal{Y} = \{1, \ldots, Y\}$.

For such HMMs, the state x_k evolves according to the $X \times X$ transition probability matrix P with elements

$$[P]_{ij} = \Pr[x_{k+1} = j | x_k = i], \quad i, j \in \mathcal{X}. \tag{6}$$

The corresponding observation y_k is measured according to the $X \times Y$ observation probability matrix B with elements

$$[B]_{ij} = \Pr[y_k = j | x_k = i], \quad i \in \mathcal{X}, j \in \mathcal{Y}. \tag{7}$$

We denote column y of the observation matrix as $b_y \in \mathbb{R}^X$ – therefore

$$B = \begin{bmatrix} b_1 & \ldots & b_Y \end{bmatrix}. \tag{8}$$

Note that both P and B are row-stochastic matrices; their elements are non-negative and the elements in each row sum to one.

Under this model structure, it can be shown – see [1] or [2] for complete treatments – that the Bayesian filter (5) for updating the posterior takes the form

$$\pi_k = T(\pi_{k-1}, y_k; P, B) = \frac{\operatorname{diag}(b_{y_k}) P^T \pi_{k-1}}{\mathbb{1}^T \operatorname{diag}(b_{y_k}) P^T \pi_{k-1}}, \tag{9}$$

initialized by π_0,[2] which we refer to as the *HMM filter*. Here, the posterior $\pi_k \in \mathbb{R}^X$ has elements

$$[\pi_k]_i = \Pr[x_k = i | y_1, \ldots, y_k], \quad i \in \mathcal{X}. \tag{10}$$

Note that $\pi_k \in \{\pi \in \mathbb{R}^X : \pi \geq 0,\ \mathbb{1}^T \pi = 1\} \stackrel{\text{def.}}{=} \Delta \subset \mathbb{R}^X$. That is, the posterior π_k lies on the $(X-1)$-dimensional unit simplex.

C. Inverse Filtering for HMMs

Although the problems we consider in this paper can be generalized to partially observed models (1)–(2) on general state and observation spaces, to obtain tractable algorithms and analytical expressions, we limit ourselves to only discrete HMMs as introduced in the previous section:

Problem 1 (Inverse Filtering for HMMs): Given a sequence of posteriors $\pi_0, \pi_1, \ldots, \pi_N \in \mathbb{R}^X$ from an HMM filter (9) with known state and observation dimensions X and Y, reconstruct the following quantities: i) the transition matrix P; ii) the observation matrix B; iii) the observations $y_1, \ldots, y_N$.

To ensure that Problem 1 is well-posed, and to simplify our analysis, we make the following two assumptions:

Assumption 1 (Ergodicity): The transition matrix P and the observation matrix B are elementwise (strictly) positive.

Assumption 2 (Identifiability): The transition matrix P and the observation matrix B are full column rank.

Assumption 1 serves as a proxy for ergodicity of the HMM and the HMM filter – it is a common assumption in statistical inference for HMMs [2], [38]. Assumption 2 is related to identifiability and assures that no state or observation distribution is a convex combination of that of another.

Remark 1: Neither of these two assumptions are strict; we violate Assumption 1 in the numerical experiments in Section V, and Assumption 2 could be relaxed according to [36, Sec. 2.4]. However, they simplify our analysis and the presentation.

D. Identifiability in Inverse Filtering

In this section, we first give a general characterization of identifiability in inverse filtering for HMMs: essentially, that different HMMs yield different HMM filters (as measured by how they update an arbitrary distribution) and *vice versa*. We then provide a specific sufficient condition to verify identifiability from a finite set of posterior distributions, which is of importance in relation to Problem 1.

Under Assumptions 1 and 2, we have the following identifiability result:

Theorem 1: Suppose that two HMMs (P, B) and $(\tilde{P}, \tilde{B})$ both satisfy Assumptions 1 and 2. Then the HMM filter is uniquely identifiable in terms of the transition and observation matrices. That is, with $\mathcal{Y} = \{1, \ldots, Y\}$,

$$T(\pi, y; P, B) = T(\pi, y; \tilde{P}, \tilde{B}), \quad \forall y \in \mathcal{Y},\ \forall \pi \in \Delta, \tag{11}$$

if and only if $P = \tilde{P}$ and $B = \tilde{B}$.

Put differently, Theorem 1 says that the Bayesian maps $T(\cdot, \cdot\,; P, B)$ and $T(\cdot, \cdot\,; \tilde{P}, \tilde{B})$ are equivalent on $\Delta \times \mathcal{Y}$ if and only if $P = \tilde{P}$ and $B = \tilde{B}$.

Theorem 1 guarantees that the HMM filter update (9) is unique in the sense that two HMMs with different transition and/or observation matrices cannot generate exactly the same filtering updates for all arbitrary distributions $\pi \in \Delta$. In turn, this leads us to expect that Problem 1 is well-posed: if we knew how any distribution $\pi \in \Delta$ would be updated by the filter we seek to identify, we should be able to reconstruct both P and B uniquely.

In practice, however, we are only given a finite sequence $\{\pi_k\}_{k=0}^N$ of posterior distributions (see Problem 1) and so Theorem 1 cannot be used (since it requires a continuum of $\pi \in \Delta$). Thus Theorem 1 does not directly lead to a practical algorithm. Note that even if we let $N \to \infty$, the realized sequence of posteriors might still not visit the whole simplex.

The following theorem is a generalization of Theorem 1 and is the main identifiability result of this paper.

Theorem 2: Suppose that two HMMs (P, B) and $(\tilde{P}, \tilde{B})$ both satisfy Assumptions 1 and 2. For each $y \in \mathcal{Y} = \{1, \ldots, Y\}$, let $\Delta_y = \{\pi_1^y, \ldots, \pi_X^y, \pi_{X+1}^y\} \subset \Delta$ be a set of $X+1$ posteriors such that $\pi_1^y, \ldots, \pi_X^y \in \mathbb{R}^X$ are linearly independent. Suppose the last posterior $\pi_{X+1}^y \in \mathbb{R}^X$ can be written as

$$\pi_{X+1}^y = [\beta]_1 \pi_1^y + \cdots + [\beta]_X \pi_X^y, \tag{12}$$

with $\beta \in \mathbb{R}^X$ and $[\beta]_i \neq 0$ for each i. Then,

$$T(\pi, y; P, B) = T(\pi, y; \tilde{P}, \tilde{B}), \quad \forall y \in \mathcal{Y},\ \forall \pi \in \Delta_y, \tag{13}$$

if and only if $P = \tilde{P}$ and $B = \tilde{B}$.

Theorem 2 relaxes Theorem 1, since relation (11) implies relation (13), in the sense that, for each observation $y = 1, \ldots, Y$, we only need $X+1$ distributions satisfying the conditions for

[2]For notational simplicity, we assume that the initial prior for the filter is the same as the initial distribution of the HMM.

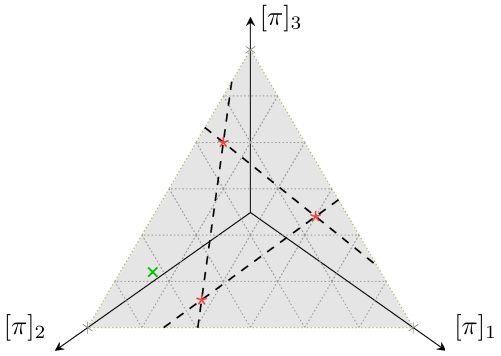

Fig. 2. Illustration of the set Δ_y in Theorem 2 (for $X = 3$). The red stars correspond to $\pi_1^y, \ldots, \pi_X^y$ and the green cross to π_{X+1}^y. For the conditions of the theorem to be fulfilled, the point π_{X+1}^y cannot lie on the dashed black lines (which correspond to $[\beta]_i = 0$ in equation (12), for some i).

the set Δ_y. Applied in the context of Problem 1, Theorem 2 guarantees that we can uniquely identify the HMM parameters once there exist subsequences $\Delta_y \subset \{\pi_k\}_{k:y_k=y} \subset \{\pi_k\}_{k=0}^N$, for $y = 1, \ldots, Y$, that satisfy the conditions given in the theorem. In particular, condition (12) says that when the last posterior vector in the set Δ_y is expressed in the basis corresponding to the first X vectors, it has non-zero components.

To make Theorem 2 more concrete, we provide an illustration in Fig. 2 for the case $X = 3$. The linear independence conditions of Theorem 2 mean that the posteriors given in Problem 1 corresponding to a measurement of each different observation cannot all lie on the lines connecting some three posteriors.

E. Direct Approach to the Inverse Filtering Problem

At a first glance, any formulation of Problem 1 as an optimization problem appears computationally intractable: there are combinatorial elements (due to the unknown sequence of observations) and non-convexity from the products between columns b_y of the observation matrix and the transition matrix P in the HMM filter (9).

In order to reconstruct parameters that are consistent with the data (i.e., that satisfy the filter equation (9) and fulfill the non-negativity and sum-to-one constraints imposed by probabilities), a direct approach is to solve the following constrained optimization problem:

$$\begin{aligned} \min_{\{y_k\}_{k=1}^N, \{b_y\}_{y=1}^Y, P} \quad & \sum_{k=1}^{N} \left\| \pi_k - \frac{\operatorname{diag}(b_{y_k}) P^T \pi_{k-1}}{\mathbb{1}^T \operatorname{diag}(b_{y_k}) P^T \pi_{k-1}} \right\| \\ \text{s.t.} \quad & y_k \in \{1, \ldots, Y\}, \qquad \text{for } k = 1, \ldots, N, \\ & b_y \geq 0, \qquad \text{for } y = 1, \ldots, Y, \\ & [b_1 \ldots b_Y]\mathbb{1} = \mathbb{1}, \\ & P\mathbb{1} = \mathbb{1}, \quad P \geq 0, \end{aligned} \tag{14}$$

where the choice of norm is arbitrary since the cost is zero for any feasible set of parameters.

The problem (14) is combinatorial (in $\{y_k\}_{k=1}^N$) and non-convex (in $\{b_y\}_{y=1}^Y$ and P). In the next section, we will propose an indirect approach that results in a computationally more attractive solution to Problem 1.

III. Inverse Filtering by Exploiting the Structure of the HMM Filter

In this section, we first derive an alternative characterization of the HMM filter (9). The properties of this characterization allow us to formulate an alternative solution to Problem 1. This solution still requires solving a combinatorial problem (a nullspace clustering problem, see Problem 2 below) that is, essentially, equivalent to (14). However, by leveraging insights from its geometrical interpretation, we derive a convex relaxation based on structured sparsity regularization (the fused group LASSO [22], [23]). The final outcome of this section is Algorithm 1 which yields a practical and computationally attractive solution to Problem 1.

A. Alternative Characterization of the HMM Filter

Our first result is a variation of the key result derived in [36]. First note that the HMM filter (9) can be rewritten as

$$(9) \iff \mathbb{1}^T \operatorname{diag}(b_{y_k}) P^T \pi_{k-1} \pi_k = \operatorname{diag}(b_{y_k}) P^T \pi_{k-1}, \tag{15}$$

by simply multiplying by the denominator (which is allowed under Assumption 1). By restructuring[3] (15), we obtain an alternative characterization of the HMM filter (9):

Theorem 3: Under Assumptions 1 and 2, the HMM filter-update (9) can be equivalently written as

$$(\pi_{k-1}^T \otimes [\pi_k \mathbb{1}^T - I]) \operatorname{vec}(\operatorname{diag}(b_{y_k}) P^T) = 0, \tag{16}$$

for $k = 1, \ldots, N$.

To see why the reformulation (16) is useful, recall that in Problem 1, we aim to estimate the transition matrix P, the observation matrix B and the observations y_k given posteriors π_k. Hence, the coefficient matrix $(\pi_{k-1}^T \otimes [\pi_k \mathbb{1}^T - I])$ on the left-hand side of (16) is known to us, and all that we aim to estimate is contained in its nullspace.

B. Reconstructing P and B from Nullspaces

It is apparent from (16) that everything we seek to estimate (i.e., the transition matrix P, the observation matrix B and the observations) is accommodated in a vector that lies in the nullspace of a known coefficient matrix. Even so, it is not obvious that the sought quantities can be reconstructed from this. In particular, since a nullspace is only determined up to scalings of its basis vectors, by leveraging (16) we can at most hope to reconstruct the *directions* of vectors $\operatorname{vec}(\operatorname{diag}(b_y) P^T)$:

$$\alpha_y \operatorname{vec}(\operatorname{diag}(b_y) P^T) \in \mathbb{R}^{X^2}, \quad y = 1, \ldots, Y, \tag{17}$$

where $\alpha_y \in \mathbb{R}_{>0}$ correspond to scale factors.

Can a set of vectors (17) be factorized into P and B, and do the undetermined scale factors α_y (which are due to the nullspace basis only being determined up to scaling) pose a problem? Our next theorem shows that it can be done and that they do not. First, note that by reshaping (17), we equivalently have access to matrices $\alpha_y \operatorname{diag}(b_y) P^T \in \mathbb{R}^{X \times X}$ for $y = 1, \ldots, Y$, which we denote by V_y in the following theorem.

[3] Detailed algebraic manipulations can be found in the appendices.

Theorem 4: Given matrices $V_y \overset{\text{def.}}{=} \alpha_y \operatorname{diag}(b_y) P^T$ for $y = 1, \ldots, Y$, where P and $B = \begin{bmatrix} b_1 & \ldots & b_Y \end{bmatrix}$ are HMM parameters satisfying Assumptions 1 and 2, and α_y are strictly positive scalars.[4] Let $L \overset{\text{def.}}{=} \sum_{y=1}^{Y} V_y^T$. Then, the transition matrix P can be reconstructed as

$$P = L \operatorname{diag}(L^{-1}\mathbb{1}). \tag{18}$$

Subsequently, let $\bar{B} \overset{\text{def.}}{=} \begin{bmatrix} V_1 P^{-T}\mathbb{1} & \ldots & V_Y P^{-T}\mathbb{1} \end{bmatrix}$. Then, the observation matrix B can be reconstructed as

$$B = \bar{B} \operatorname{diag}(\bar{B}^{\dagger}\mathbb{1}). \tag{19}$$

The proof of Theorem 4 amounts to algebraically verifying that the relations (18) and (19) hold by employing properties of row-stochastic matrices. The last factor in each equation can be interpreted as a sum-to-one normalization.

C. How to Compute the Nullspaces?

Theorem 4 gives us a procedure for reconstructing the transition and observation matrices from vectors (17) – i.e., vectors parallel to $\operatorname{vec}(\operatorname{diag}(b_y)P^T)$ for $y = 1, \ldots, Y$. The goal in the remaining part of this section is to compute such vectors from the known coefficient matrices in (16).

If the nullspace of the coefficient matrix of (16) were one-dimensional, we could proceed as in [36]: Since there are only a finite number of values, namely Y, that the y_k's can take, there are only a finite number of directions in which the nullspaces can point; along the vectors $\operatorname{vec}(\operatorname{diag}(b_y)P^T)$ for $y = 1, \ldots, Y$. Hence, once a coefficient matrix corresponding to each observation y has been obtained (i.e., once $\{1, \ldots, Y\} \subset \{y_k\}_{k=1}^{N}$), these directions can be reconstructed.

Unfortunately, the nullspace of the coefficient matrix of (16) is *not* one-dimensional:

Lemma 1: Under Assumptions 1 and 2, we have that

$$\operatorname{rank}(\pi_{k-1}^T \otimes [\pi_k \mathbb{1}^T - I]) = X - 1. \tag{20}$$

Since $\operatorname{vec}(\operatorname{diag}(b_y)P^T) \in \mathbb{R}^{X^2}$, the nullspace is, in fact, $X^2 - (X-1)$ dimensional. Below, we demonstrate how, by intersecting multiple nullspaces, we can obtain a one-dimensional subspace (a vector) that is parallel to the vector $\operatorname{vec}(\operatorname{diag}(b_y)P^T)$ that we seek.

Remark 2: The above is not surprising in the light of that every update (9) of the posterior corresponds to X equations.[5] In [36], only the X parameters of $\operatorname{diag}(b_y)$ had to be reconstructed at each time instant since P was assumed known. Now, instead, we aim to reconstruct the X^2 parameters of $\operatorname{diag}(b_y)P^T$, which cannot be done with just one update (i.e., with X equations). Hence, we will need to employ the equations from several filtering updates.

D. Special Case: Known Sequence of Observations

To make the workings of our proposed method more transparent, suppose for the moment that we have access to the sequence of observations $y_1, \ldots, y_N$ that were processed by the filter (9). By Theorem 3, we know that the vector $\operatorname{vec}(\operatorname{diag}(b_{y_k})P^T)$ lies in the nullspace of the coefficient matrix $(\pi_{k-1}^T \otimes [\pi_k \mathbb{1}^T - I])$ for all $k = 1, \ldots, N$.

If we consider only the time instants when a certain observation, say y, was processed, then the vector $\operatorname{vec}(\operatorname{diag}(b_y)P^T)$ lies in the nullspace of all the corresponding coefficient matrices:

$$\operatorname{vec}(\operatorname{diag}(b_y)P^T) \in \bigcap_{k:\, y_k = y} \ker\,(\pi_{k-1}^T \otimes [\pi_k \mathbb{1}^T - I]), \tag{21}$$

for $y = 1, \ldots, Y$. Now, if the intersection on the right-hand side of (21) is one-dimensional, this gives us a way to reconstruct the direction of $\operatorname{vec}(\operatorname{diag}(b_y)P^T)$. In this case,

$$\operatorname{span}(\operatorname{vec}(\operatorname{diag}(b_y)P^T)) = \bigcap_{k:\, y_k = y} \ker\,(\pi_{k-1}^T \otimes [\pi_k \mathbb{1}^T - I]), \tag{22}$$

and we simply compute the one-dimensional intersection. Recall that the next step would then be to factorize these directions into the products P and B via Theorem 4.

The identifiability result of Theorem 2 tells us that this happens when there exist subsequences $\Delta_y \subset \{\pi_k\}_{k:y_k=y}$ that satisfy the criteria in Theorem 2.

Remark 3: In practice, by Remark 2, roughly X updates for each y should be enough for this to occur. An upper bound on how many samples are expected to be required can be obtained via similar reasoning as in [36, Lemma 3].

E. Inverse Filtering via Nullspace Clustering

Problem 1 is complicated by the fact that in the inverse filtering problem, we do *not* have access to the sequence of observations – we only observe a sequence of posteriors. Thus, we do not know at what time instants a certain observation y was processed and, hence, which nullspaces to intersect, as in (22), to obtain a vector parallel to $\operatorname{vec}(\operatorname{diag}(b_y)P^T)$.

An abstract version of this problem can be posed as follows:

Problem 2 (Nullspace Clustering): Given is a set of matrices $\{A_k\}_{k=1}^{N}$ that can be divided into Y subsets (clustered) such that the intersection of the nullspaces of the matrices in each subset is one-dimensional. That is, there are numbers $\{y_k\}_{k=1}^{N}$ with $y_k \in \{1, \ldots, Y\}$ such that,

$$\bigcap_{k:\, y_k = y} \ker A_k = \operatorname{span}(v_y), \tag{23}$$

for some vector v_y with $y = 1, \ldots, Y$. Find the vectors $\{v_y\}_{y=1}^{Y}$ that span the intersections.

We provide a graphical illustration of the nullspace clustering problem in Fig. 3. Note that, in our instantiation of the problem, $A_k = (\pi_{k-1}^T \otimes [\pi_k \mathbb{1}^T - I])$ is the coefficient matrix of the HMM filter (16), and each vector $v_y = \operatorname{vec}(\operatorname{diag}(b_y)P^T)$ is what we aim to reconstruct (up to a positive scale factor).

The problem was simplified in the previous section, since by knowing the observations $y_1, \ldots, y_N$, we know which vector each subspace is generated about (i.e., the subset assignments) and can simply intersect the subspaces in each subset to obtain the vectors $v_1, \ldots, v_Y$. By not having direct access to the sequence of observations, the problem becomes combinatorial: which nullspaces should be intersected? Albeit a solution can

[4] Note that the scalars α_y are unknown (if they were known, one would first reconstruct P by summing $\alpha_y^{-1} V_y$ over y, which then readily yields B).

[5] Actually, only $X - 1$ equations since the sum-to-one property of the posterior makes one equation superfluous.

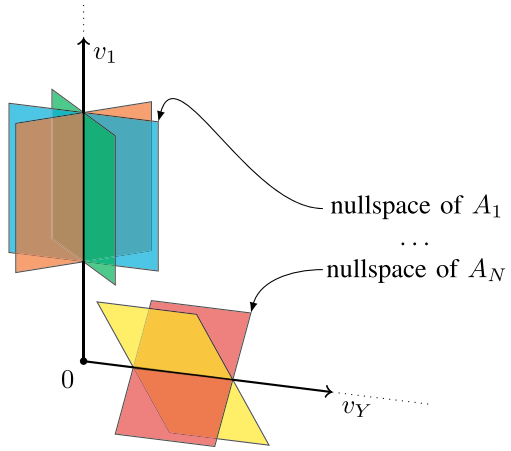

Fig. 3. We are given a set of unlabeled subspaces parametrized by the nullspaces of matrices $A_1, \ldots, A_N$. These subspaces have the property that each subspace contains one of the Y vectors $v_1, \ldots, v_Y$. In the nullspace clustering problem (Problem 2), the aim is to find the (directions of) vectors $v_1, \ldots, v_Y$.

be obtained via mixed-integer optimization in much the same fashion as in (14), since Problem 2 is merely a reformulation of the original problem, such an approach can be highly computationally demanding.

We propose instead the following two-step procedure that consists of, first, a convex relaxation, and second, a refinement step using local heuristics. We emphasize that if the two steps succeed, then there is nothing approximate about the solution we obtain; the directions of the vectors v_y are obtained exactly.

Step 1. Convex Relaxation: Compute a solution to the convex optimization problem:

$$\begin{aligned} \min_{\{w_k\}_{k=1}^N} \quad & \sum_{i=1}^{N} \sum_{j>i}^{N} \|w_i - w_j\|_\infty \\ \text{s.t.} \quad & A_k w_k = 0, \qquad \text{for } k = 1, \ldots, N, \\ & w_k \geq \mathbb{1}, \qquad \text{for } k = 1, \ldots, N, \end{aligned} \tag{24}$$

where, in our instantiation of the problem, $A_k \in \mathbb{R}^{X \times X^2}$ and $w_k \in \mathbb{R}^{X^2}$. Problem (24) aims to find N vectors w_k that each lie in the nullspace of the corresponding matrix A_k (first constraint). The objective function can be interpreted as a fused group LASSO [22], [23] that promotes the vectors to coincide – that is, the set $\{w_k\}_{k=1}^N$ is *sparse* in the number of unique vectors.

Problem (24) is a relaxation because the hard constraint of there only being exactly Y different vectors has been dropped. The second constraint $w_k \geq \mathbb{1}$ (which could be replaced by, e.g., $\mathbb{1}^T w_k \geq 1$) has been included in order to avoid the trivial solution $w_k = 0$ for all k . The $\|\cdot\|_\infty$-norm is used for convenience since problem (24) can then be recast as a linear program, which can be solved using a range of efficient algorithms [39], [40].

Step 2. Refinement via Spherical Clustering: The solution of (24) does not completely solve our problem for two reasons: i) it is not guaranteed to return precisely Y unique basis vectors, and ii) it does not tell us to which subset the nullspace of each A_k should be assigned (i.e., we still do not know which nullspaces to intersect).

In order to address these two points, we perform a local refinement using spherical k-means clustering [41] on the set of vectors $\{w_k\}_{k=1}^N$ resulting from (24). This provides us with a set of Y centroid vectors, as well as a cluster assignment of each vector w_k. We employ the spherical version of k-means since we seek nullspace basis vectors – the appropriate distance measure is angular spread, and not the Euclidean norm employed in standard k-means clustering.

Now, the centroid vectors should provide good approximations of the vectors we seek (since $\{w_k\}_{k=1}^N$ are expected to be spread around the intersections of the nullspaces by the sparsity promoting objective in (24)). However, they do not necessarily lie in any nullspace since their computation is unconstrained. To obtain an exact solution to the problem, we go through the w_k's assigned to each cluster in order of distance to the cluster's centroid and intersect the nullspaces of the corresponding A_k's until we obtain a one-dimensional intersection.

It should be underlined that when an intersection that is one-dimensional has been obtained, the (direction of) the vector v_y has been computed exactly.[6] Recall that in our instantiation of the problem, each vector $v_y = \mathrm{vec}(\mathrm{diag}(b_y)P^T)$, so that once these have been reconstructed, they can be decomposed into the transition matrix P and the observation matrix B according to Theorem 4.

Remark 4: Theorem 4 assumes that we are given the matrices V_y sorted according to the actual labeling of the HMM's observations. If we use the method described above, the vectors are only obtained up to permutations of the observation labels (this corresponds to the label assigned to each cluster in the spherical k-means algorithm). Hence, in practice, we will obtain B up to permutations of its columns.

F. Summary of Proposed Algorithm

For convenience, the complete procedure for solving Problem 1 is summarized in Algorithm 1. It should be pointed out that the algorithm will fail to determine a solution to Problem 1 if it can not intersect down to a one-dimensional subspace for some y. Then, the direction of the vector $\mathrm{vec}(\mathrm{diag}(b_y)P^T)$ cannot be determined uniquely, and the full set of these vectors is required in Theorem 4. This is because this observation has not been measured enough times, or that the convex relaxation has failed.

Remark 5: Once P and B have been reconstructed via Algorithm 1, to obtain the sequence of observations, simply check which observation $y \in \mathcal{Y}$ maps π_{k-1} to π_k via the HMM filter (9) for $k = 1, \ldots, N$. This can be done in linear time (in N).

IV. Applications of Inverse Filtering to Counter-Adversarial Autonomous Systems

During the last decade, the importance of defense against adversarial autonomous treats has been highlighted on numerous occasions – e.g., [7], [18], [42]. In this section, we illustrate how the results in the previous section can be generalized when i) the

[6] Intersecting the blue and green nullspaces in Fig. 3 reconstructs the vector v_1 exactly (assuming $v_y \in \mathbb{R}^3$). By stopping once a one-dimensional intersection has been obtained, we do not intersect nullspaces of vectors classified to the wrong clusters by the k-means algorithm (hence, a misclassified orange nullspace would not cause a problem).

Algorithm 1: Inverse Filtering (Solution to Problem 1).

Require: Sequence of posteriors $\{\pi_k\}_{k=0}^N$, dimensions X, Y

Compute the coefficient matrices $A_k = (\pi_{k-1}^T \otimes [\pi_k \mathbb{1}^T - I])$ in (16) for $k = 1, \ldots, N$
Compute a solution $\{w_k\}_{k=1}^N$ to the convex problem (24)
Run spherical k-means clustering for Y clusters on the vectors $\{w_k\}_{k=1}^N$
for each of the Y clusters **do**
  k-set $= \{\}$
  for each w_k in order of increasing distance to its cluster's centroid **do**
    Compute intersection of current, and past, corresponding A_k's nullspaces:
      i) Add k to k-set,
      ii) Compute $\bigcap_{k \in k\text{-set}} \ker A_k$
    if intersection is one-dimensional **then**
      Save as v_y and proceed to the next cluster
    end if
  end for
end for
Factorize $\{v_y\}_{y=1}^Y$ into P and B using equations (18) and (19), respectively
To obtain the corresponding sequence of observations, see Remark 5

posteriors from the Bayesian filter are observed in noise, and ii) the filtering system is mismatched. The problem is motivated by remotely calibrating (i.e., estimating) the sensors of an autonomous adversary by observing its actions (c.f., Section I-A2).

A. Counter-Adversarial Autonomous Systems

Consider an adversary that employs an autonomous filtering and control system that estimates our state and takes actions based on a control policy. The goal in the design of a *counter-adversarial autonomous* (CAA) system is to infer information private to the adversary, and to predict and guard against its future actions [7]–[9].

Formally, it can be interpreted as a two-player game in the form of a *partially observed Markov decision process* (POMDP; [1]), where information is partitioned between two players: *us* and the *adversary*. The model (1)–(2) is then generalized to:

$$\text{us:} \quad x_k \sim P_{x_{k-1},x} = p(x|x_{k-1}),\ x_0 \sim \pi_0 \tag{25}$$

$$\text{adversary:} \quad y_k \sim B_{x_k,y} = p(y|x_k), \tag{26}$$

$$\text{adversary:} \quad \pi_k = T(\pi_{k-1}, y_k; \hat{P}, \hat{B}), \tag{27}$$

$$\text{adversary \& us:} \quad u_k \sim G_{\pi_k,u} = p(u|\pi_k), \tag{28}$$

which should be interpreted as follows.

The state $x_k \in \mathcal{X}$, with initial condition π_0, is *our* state that we use to probe the adversary. The observation $y_k \in \mathcal{Y}$ is measured by the *adversary*, who subsequently computes its posterior (in this setting, we refer to it also as a *belief*) π_k of our state using the Bayesian filter T from (5).[7] Note that the adversary does *not* have perfect knowledge of our transition kernel P nor of its sensor B; it uses estimates $\hat{P}$ and $\hat{B}$ in (27). Finally, the adversary takes an action $u_k \in \mathcal{U}$, where $\mathcal{U}$ is an action set, according to a control policy G based on its belief.

A schematic overview is drawn in Fig. 4, where the dashed boxes demarcate information between the players (public means both we and the adversary have access).

B. Remote Calibration of an Adversary's Sensors

Various questions can be asked that are of importance in the design of a CAA system. The specific problem we consider is that of remotely calibrating the adversary's sensor:

Problem 3 (Remote Sensor Calibration in CAA Systems): Consider the CAA system (25)–(28). Given knowledge of our realized state sequence $x_0, x_1, \ldots, x_N$, our transition kernel P and the actions taken by the adversary $u_1, \ldots, u_N$, estimate the observation likelihoods B of the adversary's sensor.

For a concrete instantiation of the model (25)–(28) and Problem 3, see the example described in Section I-A2.

Note that our final targets in Problem 3 are the actual sensor likelihoods B and *not* the adversary's own estimate $\hat{B}$. Previous work [8], [9], [36] that considered the above problem, or variations thereof, assumed that the adversary's filter was perfectly matched (i.e., $\hat{P} = P$ and $\hat{B} = B$). This results in two significant simplifications. First, the assumption that $\hat{P} = P$ implies that the adversary knows our transition kernel. In the present setup, we will have to estimate the adversary's estimate $\hat{P}$. Second, if $\hat{B} = B$ then it is enough to estimate the model $\hat{B}$ employed by the adversary in its filter (since it is exact). In the present setup, we will have to reconstruct the observations measured by the adversary so as to form our own estimate of the sensor B.

In order to leverage the results obtained in Section III, we consider only discrete CAA systems – that is, where the state space $\mathcal{X} = \{1, \ldots, X\}$ and observation space $\mathcal{Y} = \{1, \ldots, Y\}$ are discrete. To simplify, we assume that the dimensions X and Y are known to both us and the adversary.

C. Reconstructing Beliefs from Actions

The feasibility of Problem 3 clearly depends on the adversary's policy. For example, if the policy is independent of its belief π_k, we can hardly hope to estimate anything regarding its sensor. A natural assumption is that the adversary is rational and that its policy G is based on optimizing its expected cost[43]–[45]:

$$\min_{u_k \in \mathcal{U}} \quad \mathbb{E}_{x_k} \{c(x_k, u_k) \,|\, y_1, \ldots, y_k\}$$
$$\text{s.t.} \quad u_k \in \mathcal{C}, \tag{29}$$

where $c(x, u)$ is a cost function that depends on our state and an action $u \in \mathcal{C} \subset \mathcal{U}$, with $\mathcal{C}$ and $\mathcal{U}$ being constraint and action sets, respectively.

Recall that the results in Section III reconstruct filter parameters from posteriors (see Algorithm 1). In order to employ these

[7] Again, for notational simplicity, we assume that the initial prior for the filter is the same as the initial distribution of the state.

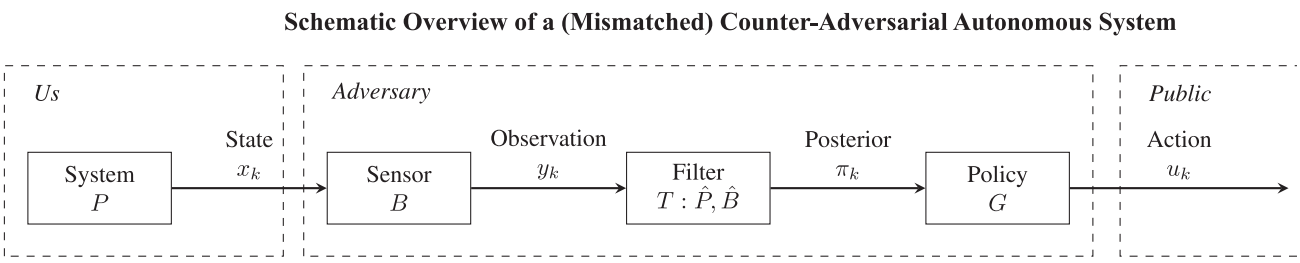

Fig. 4. Schematic illustration of the setup in (25)–(28). An autonomous adversary measures our state x_k as y_k via a sensor B. A mismatched (note that estimates $\hat{P}$ and $\hat{B}$ are employed) Bayesian filter is used to compute its posterior π_k of our state. Based on this posterior and a policy G, a public action u_k is taken. In the remote sensor calibration problem (Problem 3), the aim is to estimate the adversary's sensor B (which, in general, is different from $\hat{B}$).

results for Problem 3, we first need to obtain the adversary's posterior distributions from its actions. This is discussed to a longer extent in [8], [9] in a Bayesian framework and in [12] in an analytic setting. We will use and briefly recap the main results of the latter below.

Note, however, that even with a structural form such as (29) in place, the set of potential policies is still infinite. Without any prior assumptions on the adversary's preferences and constraints, it is impossible to conclusively infer specifics regarding its posteriors. In [12], it is assumed that the action set is continuous $\mathcal{U} = \mathbb{R}^U$, for some integer U, and that:

Assumption 3: We know the adversary's cost function $c(x, u)$ and its constraint set $\mathcal{C}$. Moreover, $c(x, u)$ is convex and differentiable in u.

Under this assumption, the full set of posteriors that the adversary could have had at any time instant was characterized in [12] using techniques from inverse optimization [46]. Some regularity conditions are needed to guarantee that a unique posterior can be reconstructed – in general, several posteriors could result in the same action, which would complicate our upcoming treatment of Problem 3. One set of such conditions is the following:

Assumption 4: The adversary's decision is unconstrained in Euclidean space ($\mathcal{C} = \mathcal{U} = \mathbb{R}^U$) and the matrix

$$F(u) = \begin{bmatrix} \nabla_u c(1,u) & \dots & \nabla_u c(X,u) \\ 1 & \dots & 1 \end{bmatrix} \tag{30}$$

has full column rank when evaluated at the observed actions $u_1, \dots, u_N$.

Intuitively, the cost functions $c(1, u), \dots, c(X, u)$ assigned to each state have to be sufficiently different for us to be able to distinguish in which state the adversary believes we are, from its actions. The matrix $F(u)$ in (30) measures how characteristic each cost function is, with respect to the others, by the linear independence of their gradients. In essence, Assumption 4 assures that no information is truncated via active constraints, and that the cost functions employed in problem (29) are sufficiently disperse.

Remark 6: The "worst case" would be that the adversary associates the same cost function to two different states $c(i, u) = c(j, u)$, for some $i \neq j \in \mathcal{X}$. In that case, it would be impossible for us to infer how likely the adversary thought each state was since both i and j would result in the same behavior.

The key result is then the following theorem:

Theorem 5: Under Assumptions 3 and 4, the posteriors of an adversary selecting actions according to (29) can be uniquely reconstructed by us from its actions as

$$\pi_k = F(u_k)^\dagger \begin{bmatrix} 0 & \dots & 0 & 1 \end{bmatrix}^T, \tag{31}$$

for $k = 1, \dots, N$, where the matrix $F(u)$ is defined in (30) and the last vector consists of X zeros and a single one.

The theorem follows directly from [12, Theorem 1] and the linear independence of columns of the matrix $F(u)$ in (30), with details available in the appendices.

D. Solution to the Remote Sensor Calibration Problem

In this section, we provide a solution to Problem 3 that leverages the above result and the inverse filtering algorithm from Section III.

Step 1. Reconstruct Posteriors: Using Theorem 5, reconstruct the adversary's sequence of posteriors $\pi_1, \dots, \pi_N$ from the observed actions $u_1, \dots, u_N$ and the structural form (29) of its policy.

Step 2. Reconstruct $\hat{P}$ and $\hat{B}$: Apply Algorithm 1 from Section III on the sequence of posteriors. Note that the posteriors were computed by the adversary using a mismatched filter (27) – i.e., as $T(\pi, y; \hat{P}, \hat{B})$ –, so that Algorithm 1 reconstructs the adversary's estimates $\hat{P}$ and $\hat{B}$ of our transition matrix P and its sensor B.

Step 3. Reconstruct the Observations: As mentioned in Remark 5, once the parameters of the filter are known the corresponding sequence of observations $y_1, \dots, y_N$ can be reconstructed (by checking which y maps π_{k-1} to π_k).

Step 4. Calibrate the Adversary's Sensor: We now have access to the observations $y_1, \dots, y_N$ that were realized by the HMM (P, B) and, by the setup of the CAA system (25)–(28), the corresponding state sequence $x_1, \dots, x_N$. With this information, we can compute *our* maximum likelihood estimate $\check{B}$ of the adversary's sensor B via

$$[\check{B}]_{ij} = \frac{\sum_{k=1}^N \mathrm{I}\{x_k = i, y_k = j\}}{\sum_{k=1}^N \mathrm{I}\{x_k = i\}}, \tag{32}$$

which corresponds to the M-step in the *expectation-maximization* (EM) algorithm for HMMs – see, e.g., [47, Section 6.2.3]. This completes the solution to Problem 3.

Discussion: It is worth making a few remarks at this point. First of all, it should be underlined that $\hat{B} \neq \check{B}$ – that is, our

estimate $\check{B}$ is not necessarily equal to that of the adversary $\hat{B}$. In fact, our estimate depends on the number N of observed actions and is, as such, *improving* over time by the consistency of the maximum likelihood estimate. If the adversary does not recalibrate its estimate $\hat{B}$ online, then for large enough N, our estimate will eventually *be more accurate* than the adversary's own estimate.

Moreover, the steps in Section IV-D are independent of the accuracy of the adversary's estimates $\hat{P}$ and $\hat{B}$ (as long as they fulfill Assumptions 1 and 2) since we are exploiting the algebraic structure of its filter. This means that, even if the adversary employs a bad estimate of our transition matrix P, as long as it is taking actions, we can improve our estimate of its sensor B.

Finally, as a potential extension, if the setup is modified so that we do not have access to the transition matrix P or the realized state sequence in (25), then the step (32) would be replaced by the full EM algorithm. This setup would correspond to a third-party observer aiming to infer the probe's transition matrix and the adversary's sensor. Again, by the asymptotic properties of the maximum likelihood estimate, the accuracy of the estimates formed by the observer (that depend on N) will eventually surpass those of the adversary.

V. Numerical Experiments

In this section, we illustrate the proposed inverse filtering algorithms in numerical examples. All simulations were run in MATLAB R2018a on a 1.9 GHz CPU. To solve problem (24) we used CVX, a package for specifying and solving convex programs[48].

A. Reconstructing P and B via Inverse Filtering

Recall that Problem 1 aims to reconstruct HMM parameters given a sequence of posterior distributions. Algorithm 1 is deterministic, but there is randomness in terms of the data (the realization of the HMM) which can cause the algorithm to fail to reconstruct the HMM's parameters. This can happen for three different reasons. First, if a certain observation has been measured too few times then there are fundamentally too few equations available to reconstruct the parameters – see Remark 2 (in Section III-C). Second, if too few independent equations have been generated, we do not have identifiability and cannot intersect to a one-dimensional subspace in (22) – see Theorem 2. Third, we rely on a convex relaxation to solve the original combinatorial problem. This is a heuristic and it is not guaranteed to converge to a solution of the original problem. Hence, in these simulations, we estimate the probability of the algorithm succeeding (with respect to the realization of the HMM data).

In order to demonstrate that the assumptions we have made in the paper are not strict, we first consider the following HMM:

$$P = \begin{bmatrix} 0 & 1/2 & 0 & 0 & 1/2 \\ 1/2 & 0 & 1/2 & 0 & 0 \\ 0 & 1/2 & 0 & 1/2 & 0 \\ 0 & 0 & 1/2 & 0 & 1/2 \\ 1/2 & 0 & 0 & 1/2 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 2/5 & 2/5 & 1/5 \\ 2/5 & 2/5 & 1/5 \\ 2/5 & 1/5 & 2/5 \\ 1/5 & 2/5 & 2/5 \\ 1/5 & 2/5 & 2/5 \end{bmatrix}. \tag{33}$$

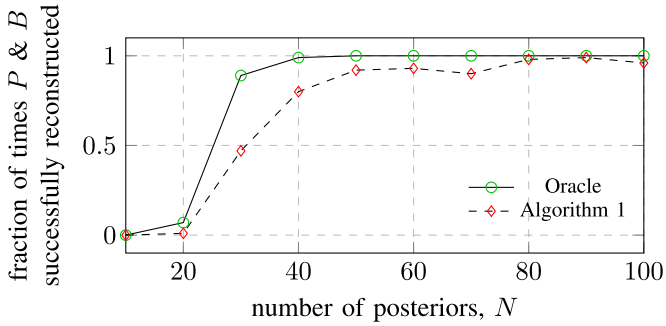

Fig. 5. In Problem 1, we obtain a sequence $\{\pi_k\}_{k=0}^{N}$ of posteriors from an HMM filter (9). Out of 100 realizations, for each value of N, we compute the fraction of times that Algorithm 1 successfully reconstructs the transition matrix P and the observation matrix B of the HMM (red diamonds). We also plot the success rate of an oracle method (green circles) that has access to the observations. With around $N = 50$ posteriors, the proposed algorithm succeeds in more than 90% of the cases.

Note that its transition matrix corresponds to a random walk, which violates Assumption 1.

We consider a reconstruction successful if the error in norm is smaller than 10^{-3} for both P and B. We generated 100 independent realizations for a range of values of N (the number of posteriors). The fractions of times that P and B were successfully reconstructed are plotted in Fig. 5 with red diamonds for Algorithm 1, and with green circles for an oracle method that has access to the corresponding sequence of observations (Section III-D). The oracle method provides an upper bound on the success rate (if it fails, it is not possible to uniquely reconstruct the HMM parameters, as discussed above). The gap between the curves is due to the convex relaxation.

A few things should be noted from Fig. 5. First, with only around 50 posteriors from the HMM filter, the fraction of times the algorithm succeeds in solving Problem 1 is high ($\approx 93\%$). Second, the gap between the two curves is small – hence, the convex relaxation is successful in achieving a solution to the original combinatorial problem. Finally, it should be mentioned that with 50 posteriors, the run-time is approximately thirty seconds.

B. Evolution of the Posterior Distribution

Next, to illustrate the conditions of Theorem 2, we plot the set $\{\pi_k\}_{k=0}^{50}$ on the simplex. To be able to visualize the data, we consider an HMM with dimension $X = 3$:

$$P = \begin{bmatrix} 0.8 & 0.1 & 0.1 \\ 0.05 & 0.9 & 0.05 \\ 0.2 & 0.1 & 0.7 \end{bmatrix}, \quad B = \begin{bmatrix} 0.7 & 0.1 & 0.2 \\ 0.1 & 0.8 & 0.1 \\ 0.05 & 0.05 & 0.9 \end{bmatrix}. \tag{34}$$

In Fig. 6, each posterior has been marked according to which observation was subsequently measured. We can clearly find four posteriors, for each observation, that are sufficiently disperse (see the illustration in Fig. 2), and hence fulfill the conditions for Theorem 2 that guarantee a unique solution.

The results of simulations with the same setup as before are shown in Fig. 7. Similar conclusions can be drawn, namely, with around 50 posteriors, the algorithm has a high rate of success ($\approx 95\%$).

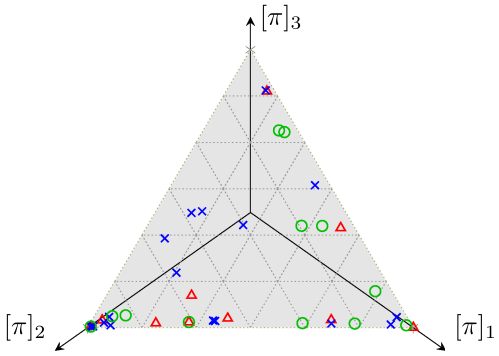

Fig. 6. A realization of the set $\{\pi_k\}_{k=0}^{50}$, corresponding to the HMM (34), illustrated on the simplex. The points are labeled according to what observation was measured: $y = 1$ (red triangle), $y = 2$ (blue cross), $y = 3$ (green circle). To fulfill the conditions of Theorem 2, the points corresponding to each observation cannot all lie on the lines connecting some three points (c.f., Fig. 2).

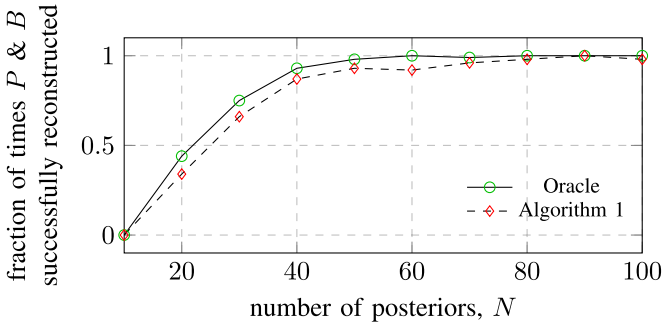

Fig. 7. Same setup as in Fig. 5, but with the HMM (34). With access to around 50 posteriors, Algorithm 1 succeeds in reconstructing P and B in about 95% of the simulations.

VI. Conclusion

This paper has investigated and extended results in inverse filtering problems for stochastic dynamic systems, as well as provided early steps towards an understanding of adversarial signal processing for counter-adversarial autonomous systems.

A. Summary

We presented four main results. First, in Section II, we formulated the inverse filtering problem for finite-alphabet HMMs: Given a sequence of posteriors $\pi_0, \ldots, \pi_N$ from an HMM filter (9), estimate the filter's parameters, comprising the transition matrix P, the observation matrix B and the measured observations $y_1, \ldots, y_N$. In relation to this, we discussed identifiability in inverse filtering by first providing a general characterization of identifiability in Theorem 1; if two HMMs induce the same filtering updates everywhere, they have to coincide. Subsequently, Theorem 2 provided instead a specific sufficient condition to verify identifiability; sufficient in the sense that it is applicable only for special sequences of posteriors and outputs.

Second, Section III accommodates solutions to the inverse problem. Motivated by computational concerns with a direct approach (14), we derived an alternative characterization of the HMM filter in Theorem 3. More specifically, Theorem 3 says that the transition and observation matrices can be obtained from the nullspace of a coefficient matrix that consists only of observed quantities. To account for the fact that the nullspace (at a single time-step) can be multi-dimensional, we abstracted it to a nullspace clustering problem: the unique solution vector (per observation value, up to scaling) is an intersection of the nullspace basis vectors of all the coefficient matrices for a cluster. Performing nullspace clustering amounts to a combinatorial feasibility problem, but its geometric interpretation allowed us to formulate a convex relaxation (24) based on the group LASSO and refined using local heuristics (Algorithm 1). After having determined the transition and observation matrices, the sequence of observations is trivially obtained from the filtering equation (Remark 5).

Third, our main motivation for studying the inverse filtering problem was its potential implications in the design of counter-adversarial autonomous systems. In Section IV, we formulated a general mathematical abstraction (25)–(28) of the setup in such a system (e.g., Section I-A2 and Fig. 1). By reconstructing the posteriors of an adversary from its actions, or directly intercepting them, we demonstrated how our inverse filtering algorithms can be used to estimate the sensor of a sophisticated autonomous adversary in Section IV-D.

Finally, Section V gave numerical examples that validated our proposed inverse filtering algorithms and demonstrated that the assumptions made are not critical to their success.

B. Extensions

There are several interesting extensions that can be made in future work. In the paper, the scope was limited to HMMs on finite observation-alphabets. Could computationally tractable algorithms for inverse filtering in general HMMs be derived? It is worth noting that the HMM filter (9) has the same structure for continuous-valued observation processes [1, Sec. 3.5]. Inverse filtering for general partially observed stochastic dynamic models (1)–(2) could potentially be approached by formulating a Bayesian filter [8] to invert suboptimal filters (e.g., particle filters), or exploiting structure in the Kalman filter [37].

Considering the case that $Y = |\mathcal{Y}|$ is unknown has a number of interesting implications. If one applies inverse filtering for an assumed $\tilde{Y}$ smaller than the true Y, does this correspond to a model-order reduction? If so, what type of model-order reduction? In practice, this could be done by modifying Step 3 of Algorithm 1 to compute $\tilde{Y}$ clusters (instead of Y). In continuation, what if one applies inverse filtering to posterior distributions π_k generated by a completely different filter architecture (e.g., a suboptimal particle filter, or deep learning)? Could one approximate such a filter with an HMM filter via inverse filtering (find the closest HMM that best approximates the alternative structure)?

With respect to the theoretical foundation of inverse filtering, it would be interesting to establish probabilistic guarantees for satisfying Theorem 2 using, e.g., recent results on reachability of positive systems [49] – the posterior π_k can via (9) be interpreted as a positive dynamical system driven by the signal y_k.

Lastly, there is a wide range of other important problems in relation to the design of counter-adversarial autonomous systems: How can we design our probing sequence x_k (via P) so as to obtain maximally informative measurements, or

maximally confuse the adversary? How can we predict the future behavior of the adversary and, in extension, devise appropriate counter-measure against it?

APPENDIX A
PROOF OF THEOREM 1

To prove Theorem 1, we use the following auxiliary lemma:

Lemma 2: Let $A \in \mathbb{R}^{X\times X}$ and $M \in \mathbb{R}^{X\times X}$ be two non-singular matrices. If

$$Ax = \kappa(x)Mx, \qquad \forall x \in \Delta, \tag{35}$$

where $\kappa(x)$ is a non-zero scalar, and $\Delta = \{x \in \mathbb{R}^X : x \geq 0, \mathbb{1}^T x = 1\}$ is the unit simplex, then

$$A = \kappa M, \tag{36}$$

where κ is a non-zero constant scalar.

Proof: Consider the ith Cartesian basis vector $e_i \in \Delta$:

$$Ae_i = \kappa(e_i)Me_i. \tag{37}$$

Concatenate (37) for $i = 1, \ldots, X$, to get

$$A\begin{bmatrix} e_1 & \ldots & e_X \end{bmatrix} = M\begin{bmatrix} \kappa(e_1)e_1 & \ldots & \kappa(e_X)e_X \end{bmatrix} \Rightarrow$$

$$A = M\begin{bmatrix} \kappa(e_1) & & 0 \\ & \ddots & \\ 0 & & \kappa(e_X) \end{bmatrix}. \tag{38}$$

Next, consider any vector on the simplex with non-zero components:

$$x = [x]_1 e_1 + \cdots + [x]_X e_X \qquad \in \Delta, \tag{39}$$

such that $[x]_i \neq 0$ for $i = 1, \ldots, X$. Introducing (38) in (35) for this x yields

$$M\begin{bmatrix} \kappa(e_1) & & 0 \\ & \ddots & \\ 0 & & \kappa(e_X) \end{bmatrix}([x]_1 e_1 + \cdots + [x]_X e_X)$$

$$= \kappa(x)M([x]_1 e_1 + \cdots + [x]_X e_X) \Rightarrow$$

$$\kappa(e_1)[x]_1 e_1 + \cdots + \kappa(e_X)[x]_X e_X$$

$$= \kappa(x)[x]_1 e_1 + \cdots + \kappa(x)[x]_X e_X, \tag{40}$$

where in the implication we have multiplied by M^{-1} from the left and simplified the expression. Since the e_i's are linearly independent, consider any component of (40):

$$\kappa(e_i)[x]_i = \kappa(x)[x]_i \Rightarrow \kappa(e_i) = \kappa(x), \tag{41}$$

for $i = 1, \ldots, X$, since $[x]_i \neq 0$. In other words,

$$\kappa(e_1) = \cdots = \kappa(e_X) = \kappa(x) \overset{\text{def.}}{=} \kappa \tag{42}$$

is constant. Introducing this in (38) yields

$$A = M\begin{bmatrix} \kappa & & 0 \\ & \ddots & \\ 0 & & \kappa \end{bmatrix} = \kappa M. \tag{43}$$

To employ Lemma 2, we reformulate (11) as follows:

$$T(\pi, y; P, B) = T(\pi, y; \tilde{P}, \tilde{B}) \Rightarrow$$

$$\frac{\text{diag}(b_y)P^T\pi}{\mathbb{1}^T\text{diag}(b_y)P^T\pi} = \frac{\text{diag}(\tilde{b}_y)\tilde{P}^T\pi}{\mathbb{1}^T\text{diag}(\tilde{b}_y)\tilde{P}^T\pi} \Rightarrow$$

$$\text{diag}(b_y)P^T\pi = \frac{\mathbb{1}^T\text{diag}(b_y)P^T\pi}{\mathbb{1}^T\text{diag}(\tilde{b}_y)\tilde{P}^T\pi}\text{diag}(\tilde{b}_y)\tilde{P}^T\pi, \tag{44}$$

which holds for all $\pi \in \Delta$ and $y = 1, \ldots, Y$.

Next, we consider (44) for a fixed y, and note that the matrices $\text{diag}(b_y)P^T$ and $\text{diag}(\tilde{b}_y)\tilde{P}^T$ are non-singular (by Assumptions 1 and 2). Lemma 2 then yields that

$$\text{diag}(b_y)P^T = \alpha(y)\text{diag}(\tilde{b}_y)\tilde{P}^T \Rightarrow$$

$$P\text{diag}(b_y) = \tilde{P}\alpha(y)\text{diag}(\tilde{b}_y), \tag{45}$$

where $\alpha(y) \in \mathbb{R}$ is a scalar and which holds for $y = 1, \ldots, Y$.

If we sum equations (45) over y, and use the fact that B is a stochastic matrix,

$$\sum_{y=1}^{Y} P\ \text{diag}(b_y) = \sum_{y=1}^{Y} \tilde{P}\alpha(y)\text{diag}(\tilde{b}_y), \qquad \Rightarrow$$

$$P = \tilde{P}\sum_{y=1}^{Y}\alpha(y)\text{diag}(\tilde{b}_y), \tag{46}$$

and then right-multiply by $\mathbb{1}$, and use that P is a stochastic matrix, we obtain

$$P\mathbb{1} = \tilde{P}\sum_{y=1}^{Y}\alpha(y)\text{diag}(\tilde{b}_y)\mathbb{1} \Rightarrow \mathbb{1} = \tilde{P}\sum_{y=1}^{Y}\alpha(y)\tilde{b}_y. \tag{47}$$

Pre-multiplying by $\tilde{P}^{-1}$ yields

$$\tilde{P}^{-1}\mathbb{1} = \tilde{P}^{-1}\tilde{P}\sum_{y=1}^{Y}\alpha(y)\tilde{b}_y \Rightarrow \mathbb{1} = \sum_{y=1}^{Y}\alpha(y)\tilde{b}_y, \tag{48}$$

where we have used the fact that the row-sums of the inverse of a (row) stochastic matrix are all equal to one.[8] By applying the diag()-operation to (48), we see that

$$I = \sum_{y=1}^{Y}\alpha(y)\text{diag}(\tilde{b}_y), \tag{49}$$

which when introduced in (46) yields that

$$\tilde{P} = P. \tag{50}$$

Next, note that (48) can be rewritten as $\mathbb{1} = \tilde{B}[\alpha(1)\ \ldots\ \alpha(Y)]^T$. We know that $\tilde{B}\mathbb{1} = \mathbb{1}$, since it is a stochastic matrix, and that $\tilde{B}$ has full column rank by assumption. Hence, $\mathbb{1} = [\alpha(1)\ \ldots\ \alpha(Y)]^T$. This yields

$$b_y = \tilde{b}_y, \tag{51}$$

for $y = 1, \ldots, Y$, from (45) by first pre-multiplying by P^{-1}.

[8] To see this, consider an invertible stochastic matrix A: $A\mathbb{1} = \mathbb{1} \Rightarrow A^{-1}A\mathbb{1} = A^{-1}\mathbb{1} \Rightarrow A^{-1}\mathbb{1} = \mathbb{1}$.

The other direction is trivial; if $P = \tilde{P}$ and $B = \tilde{B}$, then $T(\pi, y; P, B) = T(\pi, y; \tilde{P}, \tilde{B})$ for all $\pi \in \Delta$ and each $y = 1, \ldots, Y$ by (9):

$$\frac{\text{diag}(b_y)P^T\pi}{\mathbb{1}^T\text{diag}(b_y)P^T\pi} = \frac{\text{diag}(\tilde{b}_y)\tilde{P}^T\pi}{\mathbb{1}^T\text{diag}(\tilde{b}_y)\tilde{P}^T\pi}. \tag{52}$$

Appendix B
Proof of Theorem 2

We begin by giving a generalization of Lemma 2:

Lemma 3: Let $A \in \mathbb{R}^{X\times X}$ and $M \in \mathbb{R}^{X\times X}$ be two non-singular matrices. Let $\mathcal{Z} = \{z_1, \ldots, z_X, z_{X+1}\}$ be a set of vectors where $z_1, \ldots, z_X \in \mathbb{R}^X$ are linearly independent and the last vector $z_{X+1} \in \mathbb{R}^X$ when expressed in this basis has non-zero components. That is, the vector z_{X+1} can be expressed as

$$z_{X+1} = [\beta]_1 z_1 + \cdots + [\beta]_X z_X, \tag{53}$$

with $\beta \in \mathbb{R}^X$ and $[\beta]_i \neq 0$ for $i = 1, \ldots, X$. If

$$Ax = \kappa(x)Mx, \qquad \forall x \in \mathcal{Z}, \tag{54}$$

where $\kappa(x)$ is a non-zero scalar, then

$$A = \kappa M, \tag{55}$$

where κ is a non-zero constant scalar.

Proof: For the ith vector in $\mathcal{Z}$, we have by (54) that

$$Az_i = \kappa(z_i)Mz_i, \tag{56}$$

which can be concatenated to

$$\begin{aligned} A\begin{bmatrix} z_1 & \ldots & z_X \end{bmatrix} &= M\begin{bmatrix} \kappa(z_1)z_1 & \ldots & \kappa(z_X)z_X \end{bmatrix} \Rightarrow \\ AZ &= M\begin{bmatrix} z_1 & \ldots & z_X \end{bmatrix} \\ &\quad \times \begin{bmatrix} \kappa(z_1) & & 0 \\ & \ddots & \\ 0 & & \kappa(z_X) \end{bmatrix} \\ &= MZ\begin{bmatrix} \kappa(z_1) & & 0 \\ & \ddots & \\ 0 & & \kappa(z_X) \end{bmatrix}, \end{aligned} \tag{57}$$

where we have denoted $Z = \begin{bmatrix} z_1 & \ldots & z_X \end{bmatrix}$.

We can rewrite (53) with this definition of Z as

$$z_{X+1} = Z\beta, \tag{58}$$

which together with (57) yields

$$\begin{aligned} Az_{X+1} &= AZ\beta \\ &= MZ\begin{bmatrix} \kappa(z_1) & & 0 \\ & \ddots & \\ 0 & & \kappa(z_X) \end{bmatrix}\beta. \end{aligned} \tag{59}$$

Next, by employing (54) for z_{X+1} and using (58), we obtain

$$\begin{aligned} Az_{X+1} &= \kappa(z_{X+1})Mz_{X+1} \\ &= \kappa(z_{X+1})MZ\beta. \end{aligned} \tag{60}$$

Equating (59) and (60) yields

$$\begin{aligned} MZ\begin{bmatrix} \kappa(z_1) & & 0 \\ & \ddots & \\ 0 & & \kappa(z_X) \end{bmatrix}\beta = \kappa(z_{X+1})MZ\beta \Rightarrow \\ \begin{bmatrix} \kappa(z_1) & & 0 \\ & \ddots & \\ 0 & & \kappa(z_X) \end{bmatrix}\beta = \kappa(z_{X+1})\beta, \end{aligned} \tag{61}$$

by multiplying by the inverse of MZ from the left. The ith component of (61) is

$$\begin{aligned} \kappa(z_i)[\beta]_i = \kappa(z_{X+1})[\beta]_i \Rightarrow \\ \kappa(z_i) = \kappa(z_{X+1}), \end{aligned} \tag{62}$$

since $[\beta]_i \neq 0$. Hence,

$$\kappa(z_1) = \cdots = \kappa(z_X) = \kappa(z_{X+1}) \overset{\text{def.}}{=} \kappa, \tag{63}$$

which when introduced in (57) yields

$$AZ = MZ\begin{bmatrix} \kappa & & 0 \\ & \ddots & \\ 0 & & \kappa \end{bmatrix} = \kappa MZ, \tag{64}$$

or, finally, $A = \kappa M$ by multiplying with the inverse of Z from the right. ■

Remark 7: Note that Lemma 2 follows from Lemma 3 by considering the vectors $e_1, \ldots, e_X \in \Delta$ and any vector in the interior of the simplex.

As in the proof of Theorem 1, to employ Lemma 3, we first reformulate (13) as follows:

$$\begin{aligned} &T(\pi, y; P, B) = T(\pi, y; \tilde{P}, \tilde{B}) \Rightarrow \\ &\frac{\text{diag}(b_y)P^T\pi}{\mathbb{1}^T\text{diag}(b_y)P^T\pi} = \frac{\text{diag}(\tilde{b}_y)\tilde{P}^T\pi}{\mathbb{1}^T\text{diag}(\tilde{b}_y)\tilde{P}^T\pi} \Rightarrow \\ &\text{diag}(b_y)P^T\pi = \frac{\mathbb{1}^T\text{diag}(b_y)P^T\pi}{\mathbb{1}^T\text{diag}(\tilde{b}_y)\tilde{P}^T\pi}\text{diag}(\tilde{b}_y)\tilde{P}^T\pi, \end{aligned} \tag{65}$$

which holds for $y = 1, \ldots, Y$ and $\pi \in \Delta_y$.

For a fixed y, the conditions of Lemma 3 are fulfilled; identify Δ_y with the set $\mathcal{Z}$ and note that the matrices $\text{diag}(\tilde{b}_y)\tilde{P}^T$ and $\text{diag}(\tilde{b}_y)\tilde{P}^T$ are non-singular. Hence,

$$\text{diag}(b_y)P^T = \alpha(y)\text{diag}(\tilde{b}_y)\tilde{P}^T, \tag{66}$$

or, by taking the transpose,

$$P\ \text{diag}(b_y) = \tilde{P}\alpha(y)\text{diag}(\tilde{b}_y), \tag{67}$$

for $y = 1, \ldots, Y$. The setup is now exactly the same as after (45) in the proof of Theorem 1 – the rest of the proof is identical.

APPENDIX C
PROOF OF THEOREM 3

Multiply expression (9) for the HMM filter by its denominator to obtain (15), and then reshuffle the terms:

$$\begin{aligned}
&\mathbb{1}^T \text{diag}(b_{y_k}) P^T \pi_{k-1} \pi_k = \text{diag}(b_{y_k}) P^T \pi_{k-1} \iff \\
&\pi_k \mathbb{1}^T \text{diag}(b_{y_k}) P^T \pi_{k-1} = \text{diag}(b_{y_k}) P^T \pi_{k-1} \iff \\
&\left(\pi_k \mathbb{1}^T - I\right) \text{diag}(b_{y_k}) P^T \pi_{k-1} = 0. \qquad (68)
\end{aligned}$$

By vectorizing and employing a well-known result relating the vectorization operator to Kronecker and matrix products[9][50], with an appropriate grouping of the terms, we obtain that

$$\begin{aligned}
&\text{vec}\left\{\left[\pi_k \mathbb{1}^T - I\right]\left(\text{diag}(b_{y_k}) P^T\right) \pi_{k-1}\right\} = 0 \iff \\
&\left(\pi_{k-1}^T \otimes [\pi_k \mathbb{1}^T - I]\right) \text{vec}(\text{diag}(b_{y_k}) P^T) = 0. \qquad (69)
\end{aligned}$$

APPENDIX D
PROOF OF THEOREM 4

By definition, we have that

$$\begin{aligned}
L &\overset{\text{def.}}{=} \sum_{y=1}^{Y} V_y^T \\
&= \sum_{y=1}^{Y} (\alpha_y \text{diag}(b_y) P^T)^T \\
&= P \sum_{y=1}^{Y} \alpha_y \text{diag}(b_y). \qquad (70)
\end{aligned}$$

First, note that L is invertible since P is invertible (by Assumption 2) and the result of the summation is a diagonal matrix with strictly positive entries (by Assumption 1). Next, we evaluate $L\ \text{diag}(L^{-1}\mathbb{1})$ by introducing (70):

$$\begin{aligned}
&L\ \text{diag}(L^{-1}\mathbb{1}) \\
&= P \left(\sum_{y=1}^{Y} \alpha_y\ \text{diag}(b_y)\right) \text{diag}\left\{\left(P \sum_{y=1}^{Y} \alpha_y\ \text{diag}(b_y)\right)^{-1} \mathbb{1}\right\} \\
&= P \left(\sum_{y=1}^{Y} \alpha_y\ \text{diag}(b_y)\right) \\
&\qquad\qquad \text{diag}\left\{\left(\sum_{y=1}^{Y} \alpha_y\ \text{diag}(b_y)\right)^{-1} P^{-1}\mathbb{1}\right\} \\
&= P \left(\sum_{y=1}^{Y} \alpha_y\ \text{diag}(b_y)\right)\ \text{diag}\left\{\left(\sum_{y=1}^{Y} \alpha_y\ \text{diag}(b_y)\right)^{-1} \mathbb{1}\right\} \\
&= P \left(\sum_{y=1}^{Y} \alpha_y\ \text{diag}(b_y)\right) \left(\sum_{y=1}^{Y} \alpha_y\ \text{diag}(b_y)\right)^{-1} = P, \qquad (71)
\end{aligned}$$

where in the third equality we used the fact that the inverse of a row-stochastic matrix has elements on each row that sum to one,[10] and in the fourth that the result of the summation is a diagonal matrix and that it has a diagonal inverse that is obtained by inverting each element.[11] This allows us to reconstruct the transition matrix P.

To reconstruct the observation matrix, we proceed as follows. First note that by multiplying V_y by $P^{-T}\mathbb{1}$ from the right, we obtain

$$V_y P^{-T}\mathbb{1} = \alpha_y \text{diag}(b_y) P^T (P^{-T}\mathbb{1}) = \alpha_y b_y, \qquad (72)$$

which is column y of the observation matrix scaled by a factor α_y. By horizontally stacking such vectors, we build the matrix

$$\begin{aligned}
\bar{B} &\overset{\text{def.}}{=} \begin{bmatrix} V_1 P^{-T}\mathbb{1} & \dots & V_Y P^{-T}\mathbb{1} \end{bmatrix} \\
&= \begin{bmatrix} \alpha_1 b_1 & \cdots & \alpha_Y b_Y \end{bmatrix} \\
&= \begin{bmatrix} b_1 & \cdots & b_Y \end{bmatrix} \text{diag}([\alpha_1\ \dots\ \alpha_Y]^T) \\
&= B\ \text{diag}([\alpha_1\ \dots\ \alpha_Y]^T), \qquad (73)
\end{aligned}$$

which is the observation matrix B with scaled columns.

From (73), it is clear that each column of $\bar{B}$ is colinear with each corresponding column of B. Hence, we seek a diagonal matrix that properly normalizes $\bar{B}$:

$$B = \bar{B}\ \text{diag}(d), \qquad (74)$$

where $d \in \mathbb{R}^Y$ is the vector of how much each column should be scaled. By multiplying (74) from the right by $\mathbb{1}$ and employing the sum-to-one property of B, we obtain that the following should hold

$$B\mathbb{1} = \bar{B}\ \text{diag}(d)\mathbb{1} \Rightarrow \mathbb{1} = \bar{B}d. \qquad (75)$$

A solution to this equation exists by (73) and that the α_y's are non-zero – each element of d is simply the inverse of each α_y. Now, since B is full column rank and the α_y's are non-zero, relation (73) implies that $\bar{B}$ is also full column rank. Hence, the unique vector of normalization factors d is

$$d = \bar{B}^{\dagger}\mathbb{1}. \qquad (76)$$

APPENDIX E
PROOF OF LEMMA 1

We use the following result for Kronecker products[50]:

$$\text{rank}(A \otimes B) = \text{rank}(A)\,\text{rank}(B), \qquad (77)$$

which implies that

$$\text{rank}(\pi_{k-1}^T \otimes [\pi_k \mathbb{1}^T - I]) = 1 \times \text{rank}(\pi_k \mathbb{1}^T - I). \qquad (78)$$

The last factor $\text{rank}(\pi_k \mathbb{1}^T - I)$ is equal to $X - 1$, since it is a rank-1 perturbation to the identity matrix.

[9]For matrices A, B and C of compatible dimensions: $\text{vec}(ABC) = (C^T \otimes A)\,\text{vec}(B)$.

[10]Assume A is invertible and row-stochastic: $A\mathbb{1} = \mathbb{1} \Rightarrow A^{-1}A\mathbb{1} = A^{-1}\mathbb{1} \Rightarrow \mathbb{1} = A^{-1}\mathbb{1}$.

[11]If D is a diagonal matrix, then $\text{diag}(D\mathbb{1}) = D$.

APPENDIX F
PROOF OF REMARK 5

To see that a unique observation can be reconstructed at each time instant k, suppose that $\pi_k = T(\pi_{k-1}, y_k; P, B) = T(\pi_{k-1}, \tilde{y}_k; P, B)$. We have, for a general posterior vector $\pi \in \Delta$ and two observations $y, \tilde{y} \in \mathcal{Y}$, that

$$\begin{aligned} T(\pi, y; P, B) = T(\pi, \tilde{y}; P, B) \Rightarrow \\ \frac{\text{diag}(b_y)P^T\pi}{\mathbb{1}^T\text{diag}(b_y)P^T\pi} = \frac{\text{diag}(b_{\tilde{y}})P^T\pi}{\mathbb{1}^T\text{diag}(b_{\tilde{y}})P^T\pi} \Rightarrow \\ \text{diag}(b_y)P^T\pi = \alpha\text{diag}(b_{\tilde{y}})P^T\pi, \end{aligned} \tag{79}$$

where $\alpha \in \mathbb{R}_{>0}$ is a positive scalar. In continuation, equation (79) implies

$$\begin{aligned} \left(\text{diag}(b_y) - \alpha\text{diag}(b_{\tilde{y}})\right) P^T\pi = 0 \Rightarrow \\ \text{diag}(b_y - \alpha b_{\tilde{y}})P^T\pi = 0 \Rightarrow \\ \left[b_y - \alpha b_{\tilde{y}}\right]_i \left[P^T\pi\right]_i = 0, \end{aligned} \tag{80}$$

for $i = 1, \ldots, X$. Under Assumption 1, we have that $\left[P^T\pi\right]_i > 0$, so that we must have $[b_y - \alpha b_{\tilde{y}}]_i = 0$, for $i = 1, \ldots, X$. Equivalently,

$$b_y = \alpha b_{\tilde{y}}. \tag{81}$$

This yields, under Assumption 2, that $\alpha = 1$ and $y = \tilde{y}$.

APPENDIX G
PROOF OF THEOREM 5

In essence, [12, Theorem 1] amounts to formulating the Karush-Kuhn-Tucker conditions for (29) and considering the posterior as an unknown variable. It follows directly from this result that the adversary could have held a belief π_k when making the decision u_k if and only if

$$\pi_k \in \left\{ \pi \in \Delta : \sum_{i=1}^{X} [\pi]_i \nabla_u c(i, u_k) = 0 \right\}. \tag{82}$$

The set in (82) can be rewritten on matrix-vector form as

$$\left\{ \pi \in \mathbb{R}^X_{\geq 0} : \begin{bmatrix} \nabla_u c(1, u_k) & \ldots & \nabla_u c(X, u_k) \\ 1 & \ldots & 1 \end{bmatrix} \pi = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \right\}, \tag{83}$$

which is non-empty by the fact that the adversary made a decision. Since, by Assumption 4, the matrix

$$F(u_k) = \begin{bmatrix} \nabla_u c(1, u_k) & \ldots & \nabla_u c(X, u_k) \\ 1 & \ldots & 1 \end{bmatrix} \tag{84}$$

has full column rank, the set (82) – or, equivalently (83) – is singleton and the sole posterior in it is

$$\pi_k = F(u_k)^\dagger \begin{bmatrix} 0 & \ldots & 0 & 1 \end{bmatrix}^T. \tag{85}$$

REFERENCES

[1] V. Krishnamurthy, *Partially Observed Markov Decision Processes*. Cambridge, MA, USA: Cambridge University Press, 2016.

[2] O. Cappé, E. Moulines, and T. Rydén, *Inference in Hidden Markov Models*. Berlin, Germany: Springer, 2005.

[3] B. Anderson and J. Moore, *Optimal Filtering*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1979.

[4] I. Hwang, S. Kim, Y. Kim, and C. E. Seah, "A survey of fault detection, isolation, and reconfiguration methods," *IEEE Trans. Control Syst. Technol.*, vol. 18, no. 3, pp. 636–653, May 2009.

[5] P. Sundvall, P. Jensfelt, and B. Wahlberg, "Fault detection using redundant navigation modules," in *Proc. 6th IFAC Symp. Fault Detection, Supervision Safety Tech. Processes*, 2006, vol. 1, pp. 522–527.

[6] B. Wahlberg and A. C. Bittencourt, "Observers data only fault detection," in *Proc. 7th IFAC Symp. Fault Detection, Supervision Safety Tech. Processes*, 2009, vol. 42, pp. 959–964.

[7] A. Kuptel, "Counter unmanned autonomous systems (CUAxS): Priorities. Policy. Future Capabilities," *Multinational Capability Develop. Campaign*, pp. 1–34, 2017. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2963835

[8] V. Krishnamurthy and M. Rangaswamy, "How to calibrate your adversary's capabilities? Inverse filtering for counter-autonomous systems," *IEEE Trans. Signal Process.*, vol. 67, no. 24, pp. 6511–6525, Dec. 2019.

[9] R. Mattila, I. Lourenço, V. Krishnamurthy, C. R. Rojas, and B. Wahlberg, "What did your adversary believe? Optimal filtering and smoothing in counter-adversarial autonomous systems," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 5495–5499.

[10] R. Mattila, "Hidden Markov models: Identification, inverse filtering and applications," Ph.D. dissertation, Division of Decision and Control Systems, KTH Royal Institute of Technology, 2020. [Online]. Available: https://kth.diva-portal.org/smash/get/diva2:1428900/FULLTEXT01.pdf

[11] S. Haykin, "Cognitive radar: A way of the future," *IEEE Signal Process. Mag.*, vol. 23, no. 1, pp. 30–40, Jan. 2006.

[12] R. Mattila, I. Lourenço, C. R. Rojas, V. Krishnamurthy, and B. Wahlberg, "Estimating private beliefs of Bayesian agents based on observed decisions," *IEEE Control Syst. Lett.*, vol. 3, no. 3, pp. 523–528, Jul. 2019.

[13] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[14] M. E. Taylor and P. Stone, "Transfer learning for reinforcement learning domains: A survey," *J. Mach. Learn. Res.*, vol. 10, pp. 1633–1685, 2009.

[15] A. Lazaric, "Transfer in reinforcement learning: A framework and a survey," in *Reinforcement Learning: State-of-the-Art*, M. Wiering and M. van Otterlo, Eds. Springer, 2012, pp. 143–173.

[16] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.

[17] C. G. Atkeson and J. C. Santamaria, "A comparison of direct and model-based reinforcement learning," in *Proc. Int. Conf. Robot. Autom.*, 1997, vol. 4, pp. 3557–3564.

[18] J. P. Farwell and R. Rohozinski, "Stuxnet and the future of cyber war," *Survival*, vol. 53, no. 1, pp. 23–40, 2011.

[19] F. Pasqualetti, F. Dörfler, and F. Bullo, "Attack detection and identification in cyber-physical systems," *IEEE Trans. Autom. Control*, vol. 58, no. 11, pp. 2715–2729, 2013.

[20] A. Teixeira, I. Shames, H. Sandberg, and K. H. Johansson, "A secure control framework for resource-limited adversaries," *Automatica*, vol. 51, pp. 135–148, 2015.

[21] Y. Mo, S. Weerakkody, and B. Sinopoli, "Physical authentication of control systems: Designing watermarked control inputs to detect counterfeit sensor outputs," *IEEE Control Syst. Mag.*, vol. 35, no. 1, pp. 93–109, Feb. 2015.

[22] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *J. Roy. Statistical Soc. Series B (Statistical Methodology)*, vol. 68, no. 1, pp. 49–67, 2006.

[23] T. Hastie, R. Tibshirani, and M. Wainwright, *Statistical Learning With Sparsity: The Lasso and Generalizations*. Boca Raton, FL, USA: CRC Press, 2015.

[24] R. E. Kalman, "When is a linear control system optimal," *J. Basic Eng.*, vol. 86, no. 1, pp. 51–60, 1964.

[25] D. Hadfield-Menell, S. J. Russell, P. Abbeel, and A. Dragan, "Cooperative inverse reinforcement learning," in *Proc. Advances Neural Inf. Process. Syst.*, 2016, pp. 3909–3917.

[26] J. Choi and K.-E. Kim, "Nonparametric Bayesian inverse reinforcement learning for multiple reward functions," in *Proc. Advances Neural Inf. Process. Syst.*, 2012, pp. 305–313.

[27] E. Klein, M. Geist, B. Piot, and O. Pietquin, "Inverse reinforcement learning through structured classification," in *Proc. Advances Neural Inf. Process. Syst.*, 2012, pp. 1007–1015.

[28] S. Levine, Z. Popovic, and V. Koltun, "Nonlinear inverse reinforcement learning with Gaussian processes," in *Proc. Advances Neural Inf. Process. Syst.*, 2011, pp. 19–27.

[29] A. Ng, "Algorithms for inverse reinforcement learning," in *Proc. 17th Int. Conf. Mach. Learn*, 2000, pp. 663–670.
[30] C. Chamley, *Rational Herds: Economic Models of Social Learning*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[31] A. Caplin and M. Dean, "Revealed preference, rational inattention, and costly information acquisition," *Amer. Econ. Rev.*, vol. 105, no. 7, pp. 2183–2203, 2015.
[32] J. Gertler, *Fault Detection and Diagnosis in Engineering Systems*. New York, NY, USA: Marcel Dekker, 1998.
[33] J. Chen and R. J. Patton, *Robust Model-Based Fault Diagnosis for Dynamic Systems*. Berlin, Germany: Springer, 1999.
[34] F. Gustafsson, *Adaptive Filtering and Change Detection*. Hoboken, NJ, USA: Wiley, 2000.
[35] J. Lunze and J. Schrder, "Sensor and actuator fault diagnosis of systems with discrete inputs and outputs," *IEEE Trans. Syst., Man, Cybern., Part B*, vol. 34, no. 2, pp. 1096–1107, Apr. 2004.
[36] R. Mattila, C. R. Rojas, V. Krishnamurthy, and B. Wahlberg, "Inverse filtering for hidden Markov models," in *Proc. Advances Neural Inf. Process. Syst.*, 2017, pp. 4207–4216.
[37] R. Mattila, C. R. Rojas, V. Krishnamurthy, and B. Wahlberg, "Inverse filtering for linear Gaussian state-space models," in *Proc. 57th IEEE Conf. Decision Control*, 2018, pp. 5556–5561.
[38] L. E. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state Markov chains," *Ann. Math. Statist.*, vol. 37, no. 6, pp. 1554–1563, 1966.
[39] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[40] D. G. Luenberger and Y. Ye, *Linear and Nonlinear Programming*, 3rd ed. Berlin, Germany: Springer, 2008.
[41] C. Buchta, M. Kober, I. Feinerer, and K. Hornik, "Spherical k-means clustering," *J. Statistical Softw.*, vol. 50, no. 10, pp. 1–22, 2012.
[42] M. Barni and F. Pérez-González, "Coping with the enemy: Advances in adversary-aware signal processing," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2013, pp. 8682–8686.
[43] M. J. Machina, "Choice under uncertainty: Problems solved and unsolved," *J. Econ. Perspectives*, vol. 1, pp. 121–154, 1987.
[44] A. Mas-Colell, M. D. Whinston, and J. R. Green, *Microeconomic Theory*, vol. 1. London, U.K.: Oxford University Press, 1995.
[45] D. G. Luenberger, *Microeconomic Theory*. New York, NY, USA: McGraw-Hill, 1995.
[46] G. Iyengar and W. Kang, "Inverse conic programming with applications," *Oper. Res. Lett.*, vol. 33, pp. 319–330, 2005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S016763770400063X
[47] M. Vidyasagar, *Hidden Markov Processes: Theory and Applications to Biology*. Princeton, NJ, USA: Princeton Univ. Press, 2014.
[48] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," [Online]. Available: https://cvxr.com/cvx, 2014.
[49] Y. Zeinaly, J. H. van Schuppen, and B. D. Schutter, "Linear positive systems may have a reachable subset from the origin that is either polyhedral or nonpolyhedral," *SIAM J. Matrix Anal. Appl.*, vol. 41, no. 1, pp. 279–307, 2020.
[50] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1991.

Robert Mattila (Student Member, IEEE) graduated from the Engineering Physics programme of KTH Royal Institute of Technology, Stockholm, Sweden, in 2015 with an M.Sc. in systems, control and robotics. The same year, he was awarded the KTH Electrical Engineering Scholarship of Excellence. In 2020, he received his Ph.D. degree from KTH by submitting the thesis *Hidden Markov Models: Identification, Inverse Filtering and Applications*. He has been a Visiting Researcher at the California Institute of Technology (Caltech), USA, the University of British Colombia (UBC), Canada, and Cornell University, USA. His primary research interests are within inference and control of stochastic dynamical systems.

Cristian R. Rojas (Member, IEEE) was born in 1980. He received the M.S. degree in electronics engineering from the Universidad Técnica Federico Santa María, Valparaíso, Chile, in 2004, and the Ph.D. degree in electrical engineering at the University of Newcastle, NSW, Australia, in 2008. Since October 2008, he has been with the KTH Royal Institute of Technology, Stockholm, Sweden, where he is currently Associate Professor of the Division of Decision and Control Systems, School of Electrical Engineering and Computer Science. His research interests lie in system identification, signal processing, and machine learning. Dr. Rojas is a member of IEEE and the IEEE Technical Committee on System Identification and Adaptive Processing since 2013, and of the IFAC Technical Committee TC1.1. on Modelling, Identification, and Signal Processing since 2013. He is Associate Editor for the IFAC journal *Automatica* and for the IEEE CONTROL SYSTEMS LETTERS (LCSS).

Vikram Krishnamurthy (Fellow, IEEE) received the Ph.D. degree from the Australian National University, Canberra, Australia, in 1992. He is currently a Professor at the Department of Electrical and Computer Engineering, Cornell University. From 2002–2016, he was a Professor and Canada Research Chair at the University of British Columbia, Canada. His research interests include statistical signal processing, computational game theory, and stochastic control in social networks. He served as Distinguished Lecturer for the IEEE Signal Processing Society and Editor-in-Chief of the IEEE JOURNAL ON SELECTED TOPICS IN SIGNAL PROCESSING. In 2013, he was awarded an Honorary Doctorate from KTH Royal Institute of Technology, Sweden. He is an author of the book *Partially Observed Markov Decision Processes*, published by Cambridge University Press in 2016.

Bo Wahlberg (Fellow, IEEE) received the M.Sc. degree in electrical engineering and the Ph.D. degree from Linköping University, Sweden, in 1983 and 1987, respectively. He is the Professor of Chair in Automatic Control at KTH Royal Institute of Technology since 1991. He is Head of the Division of Decision and Control Systems at KTH. He is a Co-Founder of Centre of Autonomous Systems (CAS) at KTH, the Linnaeus Center ACCESS on networked systems at KTH, and the KTH director and in the program management of the Wallenberg AI, Autonomous Systems and Software Program (WASP). He was elected to IEEE Fellow in 2007, "for contributions to system identification using orthonormal basis functions" and IFAC Fellow in 2019. His main research interest is in estimation and optimization in system identification, control and signal processing with applications in process industry and transportation.