

The Mixed-Observable Constrained Linear Quadratic Regulator Problem: The Exact Solution and Practical Algorithms

Ugo Rosolia[✉], Yuxiao Chen[✉], *Member, IEEE*, Shreyansh Daftry[✉], Masahiro Ono[✉], Yisong Yue[✉],
and Aaron D. Ames[✉], *Fellow, IEEE*

Abstract—This article studies the problem of steering a linear system subject to state and input constraints toward a goal location that may be inferred only through noisy partial observations. We assume mixed-observable settings, where the system's state is fully observable and the environment's state defining the goal location is only partially observed. In these settings, the planning problem is an infinite-dimensional optimization problem where the objective is to minimize the expected cost. We show how to reformulate the control problem as a finite-dimensional deterministic problem by optimizing over a trajectory tree. Leveraging this result, we demonstrate that when the environment is static, the observation model piecewise, and cost function convex, the original control problem can be reformulated as a mixed-integer convex program that can be solved to global optimality using a branch-and-bound algorithm. The effectiveness of the proposed approach is demonstrated on navigation tasks, where the goal location should be inferred through noisy measurements.

Index Terms—Measurement uncertainty, observability, optimal control.

I. INTRODUCTION

Model predictive control (MPC) is a mature control technology that in part owns its popularity to developments in optimization solvers [1], [2], [3], [4], [5], [6], [7], [8], [9], [10]. In MPC, at each time step, an optimal planned trajectory is computed solving a finite-dimensional optimization problem, where the cost function and constraints encode the control objectives and safety requirements, respectively. Then, the first optimal control action is applied to the system and the process is repeated at the next time step based on the new measurement. This control methodology is ubiquitous in industry, with applications ranging from autonomous driving [11], [12], [13] to large-scale power systems [14], [15], [16].

For deterministic discrete-time systems, an optimal trajectory represented by a sequence of states and control actions can be computed leveraging a predictive model of the system. On the other hand, when uncertainties are acting on the system and/or only partial state observations are available, it is not possible to plan an optimal trajectory

for the closed-loop system, as its future evolution is uncertain. In these cases, the controller should plan the evolution of the system taking into account that in the future new measurements will be available. More formally, the controller should plan the evolution of the system using a policy that is a function mapping the system's state to a control action. Unfortunately, planning over policies is computationally intractable, even for the constrained linear quadratic regulator problem when additive disturbances affect the system's dynamics [17].

Several strategies have been presented in the literature to ease the computational burden of planning over policies [18], [19], [20], [21], [22], [23], [24], [25]. When the system dynamics are affected by disturbances and the system's state can be perfectly measured, the planning problem can be simplified by computing affine disturbance feedback policies that map disturbances to control actions [18], [19], [25]. Another class of feedback policies is considered in tube MPC strategies [20], [21], [22], [23], where the control actions are computed based on a predefined feedback term and a feed-forward component that is computed online by solving an optimization problem. Similar strategies may be used in partially observable settings [26], [27], [28].

The abovementioned strategies are designed for unimodal disturbances and measurement noise. However, in several practical engineering applications, uncertainties are multimodal, and it is required to design controllers that take the structure of the uncertainty into account to reduce conservatism. For instance, in autonomous driving, a controller should plan a trajectory taking into account that surrounding vehicles and pedestrians may exhibit different behaviors that can be categorized into modes, e.g., merging or lane keeping for a car, and crossing or not crossing for a pedestrian [29], [30], [31], [32]. Planning over a trajectory tree, where each branch is associated with different uncertainty modes, is a standard strategy that has been leveraged in the literature to synthesize a controller that can handle multimodal disturbances [30], [31], [32], [33], [34], [35], when perfect state feedback is available. It is also worth mentioning that adaptive dynamic programming strategies can be used to design controllers for uncertain systems when perfect state feedback is available [36], [37], [38].

In this work, we introduce the mixed-observable constrained linear quadratic regular problem, where perfect state feedback is not available for a subset of the state space. Compared with the standard LQR problem, in our formulation, we consider state and input constraints, and most importantly, we assume that only noisy environment measurements about the goal location are available. Thus, the controller has to compute actions also to collect informative measurements. This problem arises in navigation tasks, where a robot has to find an object that could be in a finite number of candidate locations, and the exact one has to be inferred through noisy measurements. We assume that the system's state is fully observable and we model the partially observable environment state, which represents the goal location, using a hidden Markov model (HMM) [39]. The HMM is constructed based on the system and the environment states, and it allows us to characterize the observation model by describing the sensors' accuracy. We consider discrete-time systems and environments with continuous and discrete state spaces, respectively. Thus, our approach generalizes Ong

Manuscript received 3 May 2022; accepted 1 September 2022. Date of publication 29 September 2022; date of current version 28 June 2023. This work was supported in part by the Jet Propulsion Laboratory and the California Institute of Technology under a contract with the National Aeronautics and Space Administration and in part by NSF under Award #1932091. Recommended by Associate Editor A. A. Malikopoulos. (*Corresponding author: Ugo Rosolia.*)

Ugo Rosolia is with Amazon, 2540 Luxembourg City, Luxembourg (e-mail: ugo.rosolia@berkeley.edu).

Yuxiao Chen, Yisong Yue, and Aaron D. Ames are with the California Institute of Technology Pasadena, CA 91125 USA (e-mail: yuxiaoc@nvidia.com; yyue@caltech.edu; ames@caltech.edu).

Shreyansh Daftry and Masahiro Ono are with the Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109 USA (e-mail: shreyansh.daftry@jpl.nasa.gov; ono@jpl.nasa.gov).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TAC.2022.3210871>.

Digital Object Identifier 10.1109/TAC.2022.3210871

et al.'s [40] work, where they introduced the mixed-observable control problem for discrete-time systems with discrete state spaces.

Our contribution is twofold. First, we show how to reformulate the optimal control problem as a deterministic finite-dimensional optimization problem over a trajectory tree. The computational cost of solving this finite-dimensional optimal control problem increases exponentially with the horizon length; thus, we introduce an approximation that can be used to compute a feasible solution to the original problem. Then, leveraging these results, we demonstrate that through a nonlinear change of coordinates the original optimal control problem can be approximated by solving a mixed-integer convex program (MICP), when the environment is static and the observation model is piecewise. As a corollary, we show that when the observation model is constant, the value function associated with the optimal control problem is convex. Finally, we test the proposed strategy on two navigation examples.

Notations: For a vector $b \in \mathbb{R}^n$ and an integer $s \in \{1, \dots, n\}$, we denote $b[s]$ as the s th component of the vector b , b^\top indicates its transpose, $M = \text{diag}(b) \in \mathbb{R}^{n \times n}$ is a diagonal matrix with diagonal elements $M[s, s] = b[s]$, and $v = 1/b$ is defined as a vector $v \in \mathbb{R}^n$ with entries $v[s] = 1/b[s]$ for all $s \in \{1, \dots, n\}$. For a function $T : \mathbb{R}^n \rightarrow \mathbb{R}$, $T(b)$ denotes the value of the function T at b . Throughout this article, we will use capital letters to indicate functions and lower letters to indicate vectors. The set of positive integers is denoted as $\mathbb{Z}_{0+} = \{1, 2, \dots\}$, and the set of (strictly) positive reals as $(\mathbb{R}_+ = (0, \infty)) \mathbb{R}_{0+} = [0, \infty)$. Furthermore, given a set \mathcal{Z} and an integer k , we denote the k th Cartesian product as $\mathcal{Z}^k = \mathcal{Z} \times \dots \times \mathcal{Z}$ and $|\mathcal{Z}|$ as the cardinality of \mathcal{Z} . Finally, given a real number $a \in \mathbb{R}$, we define the floor function $\lfloor a \rfloor$, which outputs the largest integer $i = \lfloor a \rfloor$ such that $i \leq a$.

II. PROBLEM FORMULATION

A. System and Environment Models

We consider the following linear time-invariant system:

$$x_{k+1} = Ax_k + Bu_k \quad (1)$$

where the state $x_k \in \mathbb{R}^n$, the input $u_k \in \mathbb{R}^d$, and k indexes over discrete-time steps. Furthermore, the abovementioned system is subject to the following state and input constraints:

$$u_k \in \mathcal{U} \subseteq \mathbb{R}^d \text{ and } x_k \in \mathcal{X} \subseteq \mathbb{R}^n \forall k \geq 0. \quad (2)$$

Our goal is to control system (1) in environments represented by partially observable discrete states. The environment evolution is modeled using an HMM given by the tuple $\mathcal{H} = (\mathcal{E}, \mathcal{O}, T, Z)$, where

- 1) $\mathcal{E} = \{1, \dots, |\mathcal{E}|\}$ is a set of partially observable environment states;
- 2) $\mathcal{O} = \{1, \dots, |\mathcal{O}|\}$ is the set of observations;
- 3) the function $T : \mathcal{E} \times \mathcal{E} \times \mathbb{R}^n \rightarrow [0, 1]$ describes the probability of transitioning to a state e' given the current environment state e and system's state x , i.e., $T(e', e, x) := \mathbb{P}(e'|e, x)$;
- 4) the function $Z : \mathcal{E} \times \mathcal{O} \times \mathbb{R}^n \rightarrow [0, 1]$ describes the probability of observing o , given the environment state e and the system's state x , i.e., $Z(e, o, x) := \mathbb{P}(o|e, x)$.

As the environment state e_k is partially observable, we introduce the following belief vector:

$$b_k \in \mathcal{B} = \left\{ b \in \mathbb{R}_{0+}^{|\mathcal{E}|} : \sum_{e=1}^{|\mathcal{E}|} b[e] = 1 \right\}.$$

The belief b_k is a sufficient statistics, and each entry $b_k[e]$ represents the posterior probability that the state of the environment e_k equals $e \in \mathcal{E}$, given the observation vector $\mathbf{o}_k = [o_1, \dots, o_k]$, the system's trajectory $\mathbf{x}_k = [x_1, \dots, x_k]$, the state $x(0)$, and the belief vector $b(0)$ at time $t = 0$, i.e., $b_k[e] = \mathbb{P}(e|\mathbf{o}_k, \mathbf{x}_k, x(0), b(0))$.

Consider an example where a Mars rover has to find a science sample that may be in one of several locations, which are identified using coarse

and low-resolution surface images [41], [42]. As the exact location is unknown, the rover is required to collect measurements to identify the science sample's location. In this setting, the environment could be represented by an HMM where the set of environment states \mathcal{E} collects the possible science sample locations, e.g., $\mathcal{E} = \{1oc_1, \dots, 1oc_n\}$ and $e = 1oc_i$ if the science sample is in the i th location. In the next section, we further formalize this navigation task as a regulation problem.

B. Control Objectives

Given the environment's belief $b(t)$ and system's state $x(t)$, our goal is solve the following *finite-time optimal control problem (FTOCP)*:

$$\begin{aligned} & J(x(t), b(t)) \\ & = \min_{\pi} \mathbb{E}_{\mathbf{o}_{N-1}} \left[\sum_{k=0}^{N-1} h(x_k, u_k, e_k) + h_N(x_N, e_N) \middle| b(t) \right] \\ & \text{subject to } x_{k+1} = Ax_k + Bu_k \\ & \quad u_k = \pi_k(\mathbf{o}_k, \mathbf{x}_k, x(t), b(t)) \\ & \quad x_0 = x(t) \\ & \quad u_k \in \mathcal{U}, x_k \in \mathcal{X} \forall k \in \{0, \dots, N-1\} \end{aligned} \quad (3)$$

where the stage cost $h : \mathbb{R}^n \times \mathbb{R}^d \times \mathcal{E} \rightarrow \mathbb{R}$ and the terminal cost $h_N : \mathbb{R}^n \times \mathcal{E} \rightarrow \mathbb{R}$. Note that the objective is a function of the partially observable environment states $e_k \in \mathcal{E}$, and the expectation is over the environment observations $\mathbf{o}_{N-1} = [o_1, \dots, o_{N-1}]$, which are stochastic, as discussed in Section II-A. In the abovementioned FTOCP, the optimization is carried out over the sequence of control policies $\pi = [\pi_0, \dots, \pi_{N-1}]$, and at each time k , the policy $\pi_k : \mathcal{O}^k \times \mathcal{X}^{k+1} \times \mathcal{B} \rightarrow \mathbb{R}^d$ maps the environment observations up to time k , the system's trajectory, and the initial belief $b(t)$ to the control action u_k . Note that we focus on the solution to the abovementioned finite-time control task, and we do not analyze the stability properties of the closed-loop system.

Computing the optimal solution to the FTOCP (3) is challenging as *i*) the environment's state is partially observable, *ii*) our goal is to minimize the expected cost, and *iii*) the optimization is infinite dimensional as it is carried out over the space of feedback policies, which are functions mapping states and belief vectors to inputs. In what follows, we show that the FTOCP (3) can be reformulated as a finite-dimensional nonlinear program (NLP). Leveraging the discrete nature of the set of observations \mathcal{O} , we will show that optimizing over feedback policies is equivalent to optimizing over a tree of control actions. Furthermore, we show that when the environment is static, the cost functions $h(\cdot, \cdot, e)$ and $h_N(\cdot, e)$ are convex and quadratic, and the observation function $Z(e, o, \cdot) : \mathbb{R}^n \rightarrow [0, 1]$ is piecewise for all $e \in \mathcal{E}$ and $o \in \mathcal{O}$, then the FTOCP (3) can be recast as an MICP. Finally, we show that when the observation model is constant the FTOCP (3) can be written as a convex parametric optimization problem.

III. EXACT SOLUTION

A. Cost Reformulation

As discussed in Section II-A, the belief b_k is a sufficient statistics for an HMM [39]. Therefore, at each time k , the belief can be computed using the observation o_k , the system's state x_k , and the belief at the previous time step b_{k-1} , i.e.,

$$b_k[e] = \frac{Z(e, o_k, x_k)}{\mathbb{P}(o_k|x_k, b_{k-1})} \sum_{i \in \mathcal{E}} T(e, i, x_k) b_{k-1}[i]. \quad (4)$$

For further details about the belief update equation, refer to [40] and [42]. The abovementioned equation can be written in compact form as

follows:

$$b_k = \frac{A_e(o_k, x_k) b_{k-1}}{\mathbb{P}(o_k | x_k, b_{k-1})}$$

where $\mathbb{P}(o_k | x_k, b_{k-1})$ is a normalization constant, and the matrix $A_e(o_k, x_k) \in \mathbb{R}^{|\mathcal{E}| \times |\mathcal{E}|}$, which is a function of the observations o_k and the system's state x_k at time k , is defined as follows:

$$A_e(o_k, x_k) = \Theta(o_k, x_k) \Omega(x_k) \quad (5)$$

where

$$\Omega(x_k) = \begin{bmatrix} T(1, 1, x_k) & \dots & T(1, |\mathcal{E}|, x_k) \\ T(2, 1, x_k) & \dots & T(2, |\mathcal{E}|, x_k) \\ \vdots & & \vdots \\ T(|\mathcal{E}|, 1, x_k) & \dots & T(|\mathcal{E}|, |\mathcal{E}|, x_k) \end{bmatrix} \quad (6)$$

and

$$\Theta(o_k, x_k) = \text{diag} \left(\begin{bmatrix} Z(1, o_k, x_k) & \dots & Z(|\mathcal{E}|, o_k, x_k) \end{bmatrix} \right).$$

Leveraging the abovementioned definitions, we show that the expected cost from problem (3) can be rewritten as a summation over the set of possible observations \mathcal{O} .

Proposition 1: Consider the optimal control problem (3). The expected cost can be equivalently written as

$$\begin{aligned} \mathbb{E}_{\mathbf{o}_{N-1}} \left[\sum_{k=0}^{N-1} h(x_k, u_k, e_k) + h_N(x_N, e_N) \middle| b_0 \right] \\ = \sum_{k=0}^{N-1} \sum_{\mathbf{o}_k \in \mathcal{O}^k} \sum_{e \in \mathcal{E}} v_k^{\mathbf{o}_k}[e] h(x_k, u_k, e) \\ + \sum_{\mathbf{o}_N \in \mathcal{O}^N} \sum_{e \in \mathcal{E}} v_N^{\mathbf{o}_N}[e] h_N(x_N, e) \end{aligned} \quad (7)$$

where the unnormalized belief $v_k^{\mathbf{o}_k} = A_e(o_k, x_k) v_{k-1}^{\mathbf{o}_{k-1}}$, and the matrix $A_e(o_k, x_k) \in \mathbb{R}^{|\mathcal{E}| \times |\mathcal{E}|}$ is defined in (5).

Proof: First, we note that, as the system dynamics are deterministic, the expected stage cost at time step k can be written as

$$\begin{aligned} \mathbb{E}_{\mathbf{o}_{N-1}} [h(x_k, u_k, e_k) | \mathbf{x}_k, x_0, b_0] \\ = \sum_{\mathbf{o}_k \in \mathcal{O}^k} \mathbb{E}_{\mathbf{o}_{N-1}} [h(x_k, u_k, e_k) | \mathbf{x}_k, x_0, b_0, \mathbf{o}_k] \mathbb{P}(\mathbf{o}_k | \mathbf{x}_k, x_0, b_0) \\ = \sum_{\mathbf{o}_k \in \mathcal{O}^k} \sum_{e \in \mathcal{E}} b_k[e] h(x_k, u_k, e) \mathbb{P}(\mathbf{o}_k | \mathbf{x}_k, x_0, b_0) \\ = \sum_{\mathbf{o}_k \in \mathcal{O}^k} \sum_{e \in \mathcal{E}} v_k^{\mathbf{o}_k}[e] h(x_k, u_k, e). \end{aligned} \quad (8)$$

In the abovementioned derivation, we leveraged the independence of the observations collected at each time step, i.e., $\mathbb{P}(\mathbf{o}_k | \mathbf{x}_k, x_0, b_0) = \mathbb{P}(o_1 | x_1, x_0, b_0) \times \dots \times \mathbb{P}(o_k | \mathbf{x}_k, x_0, b_0)$, and we defined

$$v_k^{\mathbf{o}_k}[e] = Z(e, o_k, x_k) \sum_{i \in \mathcal{E}} T(e, i, x_k) v_{k-1}^{\mathbf{o}_{k-1}}[i]$$

which can be written in compact form as $v_k^{\mathbf{o}_k} = A_e(o_k, x_k) v_{k-1}^{\mathbf{o}_{k-1}}$. Finally, we note that the derivation in (8) holds also for the terminal cost function h_N . Therefore, we have that the desired result follows from (8) and the linearity of the expectation in (7). ■

B. Deterministic Reformulation

In the previous section, we showed how to leverage the beliefs associated with all possible observations to express the expectation as a summation. In this section, we show that the optimization carried

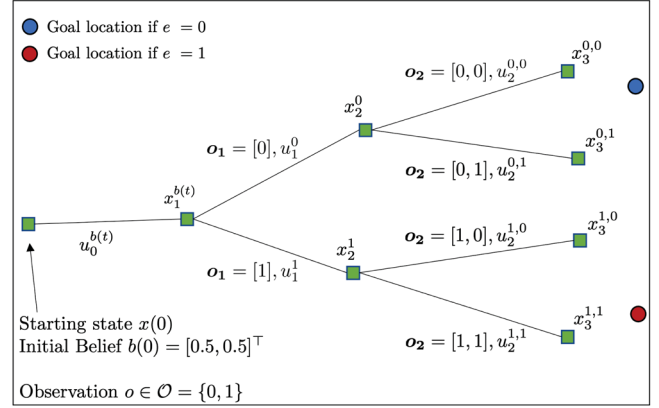


Fig. 1. Tree of trajectories for $N = 3$, where at each time k , there are $|\mathcal{O}| = 2$ possible observations. Each predicted control action $u_k^{\mathbf{o}_k}$ is associated with an observation vector $\mathbf{o}_k \in \mathcal{O}^k$. Thus, the abovementioned tree encodes a policy given by the actions that the controller would apply depending on the observations collected up to time k .

out over feedback policies can be reformulated as an optimization over a tree of trajectories, as the one shown in Fig. 1.

The control policy $\pi_k : \mathcal{O}^k \times \mathcal{X}^{k+1} \times \mathcal{B}$ from (3) maps the vector of observations $\mathbf{o}_k = [o_1, \dots, o_k] \in \mathcal{O}^k$, the system's trajectory, and the initial belief $b_0 = b(0)$ to the control action u_k , i.e., $u_k = \pi(\mathbf{o}_k, \mathbf{x}_k, x_0, b_0)$. Note that the system dynamics from problem (3) are deterministic, and therefore, given an initial condition $x(t)$ and an initial belief $b(t)$, the control action at time k is a function only of the observation vector \mathbf{o}_k . Thus, we define the control action $u_k^{\mathbf{o}_k} \in \mathbb{R}^d$ associated with the observation vector $\mathbf{o}_k \in \mathcal{O}^k$, and we reformulate problem (3) as an optimization problem over the set of control actions $\{u_k^{\mathbf{o}_k} \in \mathbb{R}^d : k \in \{0, \dots, N-1\}, \mathbf{o}_k \in \mathcal{O}^k\}$. This strategy allows us to optimize over policies as at time k , the controller plans $|\mathcal{O}|^k$ distinct actions associated with each uncertain sequence of observations $\mathbf{o}_k = [o_1, \dots, o_k] \in \mathcal{O}^k$. Basically, the controller optimizes over a tree of control actions, as shown in Fig. 1. More formally, given the environment's belief $b(t)$ and the system's state $x(t)$, we rewrite problem (3):

$$\begin{aligned} J(x(t), b(t)) = \min_{\mathbf{u}} \sum_{k=0}^{N-1} \sum_{\mathbf{o}_k \in \mathcal{O}^k} \sum_{e \in \mathcal{E}} v_k^{\mathbf{o}_k}[e] h(x_k^{\mathbf{o}_k}, u_k^{\mathbf{o}_k}, e) \\ + \sum_{\mathbf{o}_N \in \mathcal{O}^N} \sum_{e \in \mathcal{E}} v_N^{\mathbf{o}_N}[e] h_N(x_N^{\mathbf{o}_N}, e) \\ \text{subject to } x_{k+1}^{\mathbf{o}_k} = A x_k^{\mathbf{o}_{k-1}} + B u_k^{\mathbf{o}_k} \\ x_0^{\mathbf{o}_0} = x(t), v_0^{\mathbf{o}_0} = b(t) \\ v_{k+1}^{\mathbf{o}_{k+1}} = A_e(o_{k+1}, x_{k+1}^{\mathbf{o}_k}) v_k^{\mathbf{o}_k} \\ u_k^{\mathbf{o}_k} \in \mathcal{U}, x_{k+1}^{\mathbf{o}_k} \in \mathcal{X} \\ \forall \mathbf{o}_k \in \mathcal{O}^k \quad \forall k \in \{0, \dots, N-1\} \end{aligned} \quad (9)$$

where the vector of observations $\mathbf{o}_k = [o_1, \dots, o_k]$ for all $k \in \{1, \dots, N-1\}$, and at time $k=0$, we defined $\mathbf{o}_0 = \mathbf{o}_{-1} = b(t)$, and $\mathcal{O}^0 = b(t)$. In the abovementioned problem, the matrix of decision variables is defined as

$$\mathbf{u} = [u_0^{\mathbf{o}_0}, \dots, u_{N-1}^{\mathbf{o}_{N-1}}] \in \mathbb{R}^{d \times \sum_{k=0}^{N-1} |\mathcal{O}|^k}. \quad (10)$$

Note that, for each time step $k \in \{0, \dots, N-1\}$, the abovementioned matrix collects the $|\mathcal{O}|^k$ control actions associated with all observation vectors from the set \mathcal{O}^k .

Lemma 1: Assume that \mathcal{X} and \mathcal{U} are compact. Let $\mathcal{C} \subset \mathcal{X}$ be a control invariant set for system (1) subject to constraints (2), i.e., $\forall x \in \mathcal{X}$, there exists $u \in \mathcal{U}$ such that $Ax + Bu \in \mathcal{C}$. If $x(t) \in \mathcal{C}$, then problem (9) admits an optimal solution.

Proof: By definition, we have that for $x(t) \in \mathcal{C}$ there exists a sequence of N control actions that keep the system inside \mathcal{C} . Hence, problem (9) is feasible. Compactness of state and input constraint sets yields the desired result. ■

C. Practical Approach

FTOCP (9) is a finite-dimensional NLP that can be solved with off-the-self solvers. However, the computational cost of solving (9) is nonpolynomial in the horizon length, as the number of decision variables from (10) grows exponentially with the horizon length N . Indeed, at each time step k , the predicted trajectory branches as a function of the discrete observation $o_k \in \mathcal{O}$, as shown in Fig. 1. In this section, we introduce an approximation to the FTOCP (9), where the predicted trajectory branches every N_b time steps. This strategy allows us to limit the number of optimization variables and, for a prediction horizon of N steps, the computational burden is proportional to the ratio N/N_b .

Given the current state $x(t)$, the environment's belief $b(t)$, the constant $N_b \in \mathbb{Z}_{0+}$, and the prediction horizon $N = PN_b$ with $P \in \mathbb{Z}_{0+}$, we solve the following FTOCP:

$$\begin{aligned} & \hat{J}(x(t), b(t)) \\ &= \min_{\mathbf{a}} \sum_{k=0}^{N-1} \sum_{\bar{o}_{j(k)} \in \mathcal{O}^{j(k)}} \sum_{e \in \mathcal{E}} \bar{v}_k^{\bar{o}_{j(k)}}[e] h\left(s_k^{\bar{o}_{j(k)}}, a_k^{\bar{o}_{j(k)}}, e\right) \\ & \quad + \sum_{\bar{o}_{j(N)} \in \mathcal{O}^{j(N)}} \sum_{e \in \mathcal{E}} \bar{v}_k^{\bar{o}_{j(N)}}[e] h_N\left(s_N^{\bar{o}_{j(N)}}, e\right) \\ & \text{subject to } s_{k+1}^{\bar{o}_{j(k)}} = A s_k^{\bar{o}_{j(k-1)}} + B a_k^{\bar{o}_{j(k)}} \\ & \quad s_0^{-1} = x(t), \bar{v}_0^{\bar{o}_0} = b(t) \\ & \quad \bar{v}_{k+1}^{\bar{o}_{j(k+1)}} = C_e\left(\bar{o}_{j(k+1)}, s_{k+1}^{\bar{o}_{j(k)}}, k\right) \bar{v}_k^{\bar{o}_{j(k)}} \\ & \quad a_k^{\bar{o}_{j(k)}} \in \mathcal{U}, s_{k+1}^{\bar{o}_{j(k)}} \in \mathcal{X} \\ & \quad j(k) = \lfloor k/N_b \rfloor \\ & \quad \forall \bar{o}_{j(k)} \in \mathcal{O}^{j(k)} \forall k \in \{0, \dots, N-1\} \end{aligned} \quad (11)$$

where for $P = N/N_b \in \mathbb{Z}_{0+}$ and $j(k) = \lfloor k/N_b \rfloor$ the matrix of decision variables

$$\mathbf{a} = \left[a_0^{\bar{o}_0}, \dots, a_P^{\bar{o}_0}, \dots, a_k^{\bar{o}_{j(k)}}, \dots, a_{N-1}^{\bar{o}_{j(N-1)}} \right] \in \mathbb{R}^{d \times \sum_{k=0}^{P-1} N_b |\mathcal{O}|^k} \quad (12)$$

the vector of observations $\bar{o}_{j(N-1)} = \bar{o}_{P-1} = [\bar{o}_1, \dots, \bar{o}_{P-1}]$, and the matrix $C_e(\bar{o}_k, s_k^{\bar{o}_{j(k-1)}}, k)$ is defined as

$$\begin{aligned} & C_e\left(\bar{o}_{j(k)}, s_k^{\bar{o}_{j(k-1)}}, k\right) \\ &= \begin{cases} A_e\left(\bar{o}_{j(k)}, s_k^{\bar{o}_{j(k-1)}}\right) & \text{If } \lfloor k/N_b \rfloor = t/N_b \text{ and } k > 0 \\ \Omega\left(s_k^{\bar{o}_{j(k-1)}}\right) & \text{otherwise} \end{cases} \end{aligned} \quad (13)$$

where $\Omega(\cdot)$ is defined as in (6).

Compare the FTOCP (9) with the FTOCP (11). In the FTOCP (9), we optimize over the tree of trajectories shown in Fig. 1; therefore, the complexity of the problem grows exponentially with the horizon length N . On the other hand, in the FTOCP (11), we optimize over a

tree of trajectories that branches every N_b time steps, and the matrix of optimization variables (12) grows exponentially with the ratio $P = N/N_b$. Therefore, in the FTOCP (11), the user-defined constant N_b may be used to limit the computational complexity when planning over a horizon N . As a tradeoff, the optimal value function \hat{J} associated with the FTOCP (11) only approximates the value function J associated with the FTOCP (9).

IV. STATIC ENVIRONMENTS, PIECEWISE OBSERVATION MODEL, AND QUADRATIC COST: THE EXACT SOLUTION

In this section, we consider problems with static environments, piecewise observation model, and convex quadratic cost function. Under these assumptions, we show that problem (9) can be reformulated as an MICP. In what follows, we first introduce the problem setup. Then, we show how to reformulate problem (9) as an MICP.

Assumption 1 (Static environment): The environment is static, which in turns implies that the transition function T is defined as follows: $T(e, e) = 1, T(e, e') = 0, \forall e \in \mathcal{E} \text{ and } \forall e' \in \mathcal{E} \text{ such that } e \neq e'$.

Assumption 2 (Piecewise observation model): The observation model is a piecewise function of the system state x . In particular, given R disjointed polytopic regions $\{\mathcal{X}_i\}_{i=1}^R$ such that $\cup_{i=1}^R \mathcal{X}_i = \mathcal{X}$, we have that $Z(e, o, x) = M_i(e, o)$ if $x \in \mathcal{X}_i$, for a set of functions $M_i : \mathcal{E} \times \mathcal{O} \rightarrow [0, 1]$.

Assumption 3 (Convex quadratic cost function): For a fixed environment state $e \in \mathcal{E}$, the stage cost $h(\cdot, \cdot, e) : \mathbb{R}^n \times \mathbb{R}^d \rightarrow \mathbb{R}$ and the terminal cost $h_N(\cdot, e) : \mathbb{R}^n \rightarrow \mathbb{R}$ are convex and quadratic, i.e., $h(x, u, e) = \|x - x_g^{(e)}\|_Q + \|u - u_g^{(e)}\|_R, h_N(x, e) = \|x - x_g^{(e)}\|_{Q_N}$, where the weighted square norm $\|x\|_Q = x^\top Q x$ for the positive semidefinite matrix Q , and the vectors $x_g^{(e)} \in \mathbb{R}^n$ and $u_g^{(e)} \in \mathbb{R}^d$ are user defined.

Assumption 4 (Strictly positive belief): All entries of the belief vector $b(0)$ are strictly positive, i.e., $b(0) \in \mathcal{B}_+ = \{b \in \mathbb{R}_{0+}^{|\mathcal{E}|} : \sum_{i=1}^{|\mathcal{E}|} b[e] = 1\}$. Furthermore, we cannot observe the true environment state $e \in \mathcal{E}$ from any state $x \in \mathcal{X}$, i.e., $\mathbb{P}(o = e | e, x) = Z(e, o, x) : \mathcal{E} \times \mathcal{O} \times \mathbb{R}^n \rightarrow (0, 1)$.

Given the system's state $x(t)$ and the inverse belief vector $z(t) = 1/b(t) \in \mathbb{R}^{|\mathcal{E}|}$, we define the following FTOCP:

$$\begin{aligned} V(x(t), z(t)) &= \min_{\mathbf{u}, \delta} \sum_{k=0}^{N-1} \sum_{\mathbf{o}_k \in \mathcal{O}^k} \sum_{e \in \mathcal{E}} \frac{h(x_k^{\mathbf{o}_k}, u_k^{\mathbf{o}_k}, e)}{z_k^{\mathbf{o}_k}[e]} \\ & \quad + \sum_{\mathbf{o}_N \in \mathcal{O}^N} \sum_{e \in \mathcal{E}} \frac{h_N(x_N^{\mathbf{o}_N}, e)}{z_N^{\mathbf{o}_N}[e]} \\ & \text{subject to } x_{k+1}^{\mathbf{o}_k} = A x_k^{\mathbf{o}_{k-1}} + B u_k^{\mathbf{o}_k} \\ & \quad x_0^{-1} = x(t), z_0^{\mathbf{o}_0} = z(t) \\ & \quad u_k^{\mathbf{o}_k} \in \mathcal{U}, x_k^{\mathbf{o}_k} \in \mathcal{X} \\ & \quad z_{k+1}^{\mathbf{o}_{k+1}} = \sum_{i=1}^R D_i(o_{k+1}) z_k^{\mathbf{o}_k} \delta_{k,i}^{\mathbf{o}_k} \\ & \quad \delta_{k,i}^{\mathbf{o}_k} = \mathbb{1}_{\mathcal{X}_i}(x_k^{\mathbf{o}_k}) \quad \forall i \in \{1, \dots, R\} \\ & \quad \forall k \in \{0, \dots, N-1\} \end{aligned} \quad (14)$$

where the indicator function $\mathbb{1}_{\mathcal{X}_i}(x_k^{\mathbf{o}_k}) = 1$ if $x_k^{\mathbf{o}_k} \in \mathcal{X}_i$ and zero, otherwise, and the optimization variables

$$\begin{aligned} \mathbf{u} &= [u_0^{\mathbf{o}_0}, \dots, u_{N-1}^{\mathbf{o}_{N-1}}] \in \mathbb{R}^{d \times \sum_{k=0}^{N-1} |\mathcal{O}|^k}, \\ \delta &= [\delta_{0,1}^{\mathbf{o}_0}, \dots, \delta_{N-1,R}^{\mathbf{o}_{N-1}}] \in \{0, 1\}^R \sum_{k=0}^{N-1} |\mathcal{O}|^k. \end{aligned} \quad (15)$$

Note that at each time k for the vector of observations \mathbf{o}_k , we have that the integer variable $\delta_{k,i}^{\mathbf{o}_k}$ equals one if and only if the state $x_k^{\mathbf{o}_k} \in \mathcal{X}_i$.

In the abovementioned problem, for all $i \in \{1, \dots, R\}$, the entries of diagonal matrices $D_i(o) \in \mathbb{R}^{|\mathcal{E}| \times |\mathcal{E}|}$ are defined as follows:

$$D_i(o)[e, e] = 1/M_i(o, e) \quad \forall e \in \mathcal{E} \quad \forall o \in \mathcal{O}. \quad (16)$$

The following theorem shows that, under Assumptions 1–4, problem (14) is equivalent to problem (3). Furthermore, problem (14) can be recast as an MICP.

Theorem 1: Consider problems (3) and (14). Let Assumptions 1–4 hold. Then, for $z(t) = 1/b(t)$, we have that

$$J(x(t), b(t)) = V(x(t), z(t))$$

for all $x(t) \in \mathcal{X}$. Furthermore, for all $z(t) \in \mathbb{R}_+^{|\mathcal{E}|}$ and $x(t) \in \mathbb{R}^n$, problem (14) can be recast as an MICP.

Proof: First, we show that $z_k^{\text{ok}} = 1/v_k^{\text{ok}}$ for all $k \in \{0, \dots, N-1\}$. From Assumptions 1–2, we have that for $x_k \in \mathcal{X}_i$, the unnormalized belief update is

$$\begin{aligned} v_k^{\text{ok}}[e] &= Z(e, o_k, x_k) \sum_{i \in \mathcal{E}} T(e, i) v_{k-1}^{\text{ok}}[i] \\ &= Z(e, o_k, x_k) v_{k-1}^{\text{ok}}[e] \\ &= M_i(e, o_k) v_{k-1}^{\text{ok}}[e]. \end{aligned} \quad (17)$$

From the abovementioned equation and definition (16), we have that $z_k^{\text{ok}}[e] = 1/v_k^{\text{ok}}[e] \forall e \in \mathcal{E}$, which in turns implies that the optimal cost from problem (14) equals the one from problem (9); therefore,

$$J(x(t), b(t)) = V(x(t), z(t))$$

for all $x(t) \in \mathcal{X}$.

Note that the objective function in problem (14) is convex, as it is given by a convex quadratic function over a strictly positive linear function [43]. Furthermore, given the initial condition $z(t)$, we can compute an upper bound $z_k^{\text{max}}[e]$ for each e th entry of the unnormalized belief z_k^{ok} , i.e.,

$$z_k^{\text{max}}[e] = \left(\max_{o \in \mathcal{O}, i \in \{1, \dots, R\}} D_i(o) \right)^{k-1} z_0^{\text{ok}}[e] \geq z_k^{\text{ok}}[e]. \quad (18)$$

Finally, we have that the piecewise model from Assumption 2 is a mixed logical dynamical systems [44]. Thus, following the procedure presented in [44], problem (14) can be recast as an MICP using the upper bound from (18). ■

Corollary 1: Consider problem (14) and let Assumptions 1–4 hold. If the observation model is not a function of the system's state, i.e., for some $G : \mathcal{E} \times \mathcal{O} \rightarrow [0, 1]$, we have that

$$Z(e, o, x) = G(e, o) \quad \forall x \in \mathcal{X}.$$

Then, the value function $V(x(t), z(t))$ from problem (14) is convex in its arguments.

Proof: As the observation model does not depend on the system's state, we have that the belief update in problem (14) can be rewritten as follows: $z_{k+1}^{\text{ok}} = F(o_{k+1}) z_k^{\text{ok}}$, where $F(o)[e, e] = 1/G(o, e)$ for all $e \in \mathcal{E}$ and $o \in \mathcal{O}$. Therefore, problem (14) is a convex parametric program and $V(x(t), z(t))$ is a convex function [45]. ■

V. EXAMPLES

We tested the proposed strategy on two navigation problems, where a linear system has to reach a goal location that may be inferred only through partial observations. The goal location represents an object to be retrieved and whose location is only partially known. We consider

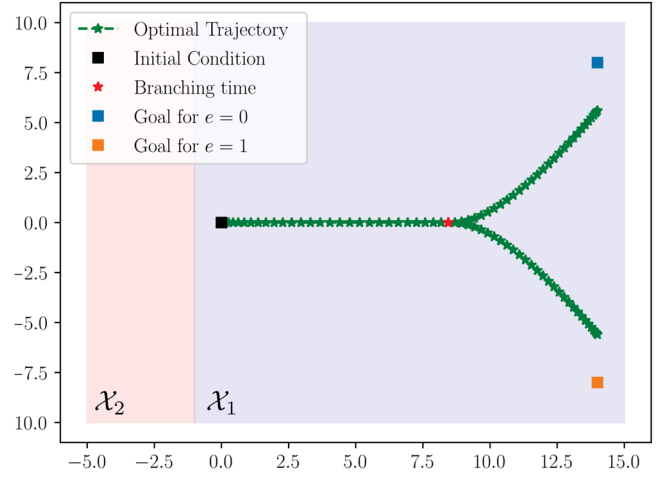


Fig. 2. Optimal trajectory computed solving the MICP (14) for $N = 60$ and assuming that an observation is collected every $N_b = 30$ time steps, as discussed in Section III-C. In this scenario, $p_1 = p_2 = 0.85$; therefore, the optimizer computes a trajectory that first steers the system toward the goals and then commits to one of the two goal locations depending on observation measured at time $t = N_b$.

the following discrete-time unstable point mass model:

$$x_{k+1} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1.1 & 0 \\ 0 & 0 & 0 & 1.1 \end{bmatrix} x_k + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} u_k \quad (19)$$

where the state vector $x_k = [X_k, Y_k, v_k^x, v_k^y]$ collects the position of the system (X_k, Y_k) and the velocity (v_k^x, v_k^y) along the X – Y plane. In the abovementioned system, the input $u_k = [a_k^x, a_k^y]$ represents the accelerations along the X and Y coordinates.

A. Mixed Observable Regulation Problem

In this example, the constraint sets are defined as follows:

$$\mathcal{U} = \{u \in \mathbb{R}^2 : \|u\|_\infty \leq 10\}$$

$$\mathcal{X} = \{[X, Y, v^x, v^y]^\top \in \mathbb{R}^4 : -5 \leq X \leq 15, \|Y\|_\infty \leq 10\}$$

and the cost matrices from Assumption 3 are

$$Q = 10^{-5} I_n, R = 10^{-3} I_d, \text{ and } Q_N = 10^2 I_n$$

where $I_n \in \mathbb{R}^{n \times n}$ represents the identity matrix.

The set of partially observable states $\mathcal{E} = \{0, 1\}$ and the associated goal locations $x_g^{(0)} = [14, 8, 0, 0]^\top$ and $x_g^{(1)} = [14, -8, 0, 0]^\top$, as shown in Figs. 2 and 3. The environment state, and consequently the goal location, is inferred through partial observations. Given the true environment state $e \in \mathcal{E}$ and the system's state $x \in \mathbb{R}^n$, the probability of measuring the observation $o = e$ is given by the following piecewise observation model:

$$Z(o = e, e, x) = \mathbb{P}(o = e | e, x) = \begin{cases} p_1 & \text{If } x \in \mathcal{X}_1 \\ p_2 = 0.85 & \text{If } x \in \mathcal{X}_2 \end{cases} \quad (20)$$

where

$$\mathcal{X}_1 = \{[X, Y, v^x, v^y] \in \mathbb{R}^4 : -1 \leq X \leq 15, \|Y\|_\infty \leq 10\}$$

$$\mathcal{X}_2 = \{[X, Y, v^x, v^y] \in \mathbb{R}^4 : -5 \leq X < -1, \|Y\|_\infty \leq 10\}.$$

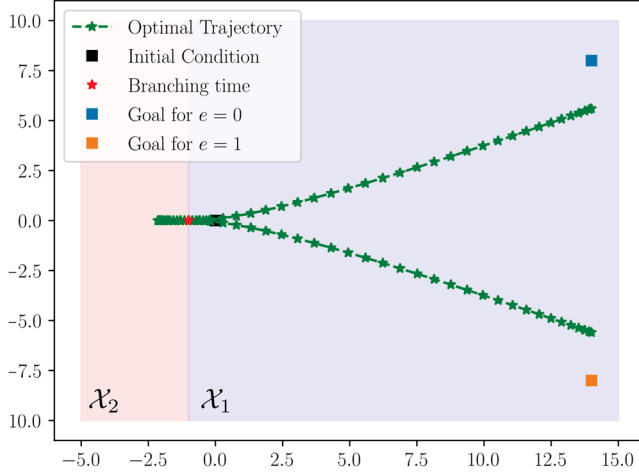


Fig. 3. Optimal trajectory computed solving the MICP (14) for $N = 60$ and assuming that an observation is collected every $N_b = 30$ time steps, as discussed in Section III-C. In this scenario, $p_1 = 0.7$; therefore, the controller steers the system backward to reach region \mathcal{X}_2 to collect a measurement that is correct with probability $p_2 = 0.85$, before committing to a goal location.

TABLE I

OPTIMAL COST $V(x(0), b(0))$ AND SOLVER TIME FOR DIFFERENT VALUES OF N_b AND CONSEQUENTLY OF $P = N/N_b$

	$N_b = 12$	$N_b = 15$	$N_b = 20$	$N_b = 30$
$V(x(0), b(0))$	1237.43	1583.31	2196.75	3265.31
Solver Time [s]	134.1	12.1	2.8	1.7
$P = N/N_b$	5	4	3	2

We implemented the finite-dimensional MICP (14) using CVXPY [46] and Gurobi [47]. In order to limit the computational burden, we leveraged the strategy discussed in Section III-C for $N = 60$, and $N_b \in \{12, 15, 20, 30\}$. All computations are run on a 2015 MacBook Pro and the code can be found at github.¹

We tested the proposed strategy for two different scenarios. In the first scenario, we set the probability p_1 of the observation model (20) equal to 0.85, and in the second one, we set $p_1 = 0.7$. In both cases, we considered as initial condition $x(0) = [0, 0, 0, 0]^T$, an initial belief $b(0) = [0.5, 0.5]^T$, and a prediction horizon $N = 60$, and we assumed that an observation is collected every $N_b = 30$ time steps. In the first scenario shown in Fig. 2, the probability $p_1 = p_2 = 0.85$ and the observations collected in regions \mathcal{X}_1 and \mathcal{X}_2 are equally informative. Thus, the optimizer steers the system forward, and after collecting an observation at time $t = N_b$ commits to a goal location. On the other hand, when $p_1 = 0.7$, the observation collected in region \mathcal{X}_1 is not as informative as the one collected in region \mathcal{X}_2 . Therefore, the optimizer plans a trajectory that moves backward and visits region \mathcal{X}_2 to collect an observation that is correct with probability $p_2 = 0.85$, before committing to steer the system toward a goal location, as shown in Fig. 3.

Table I tabulates the optimal cost and the computational time to solve the MICP for different values of N_b and for $N = 60$. As discussed in Section III-C, as $P = N/N_b$ gets larger, the optimization tree has more branches and, consequently, the problem complexity increases. In particular, the number of optimization variables $v = \sum_{k=0}^{P-1} N_b |\mathcal{O}|^k$ grows exponentially as a function of P .

¹[Online]. Available: <https://github.com/urossolia/mixed-observable-LQR>

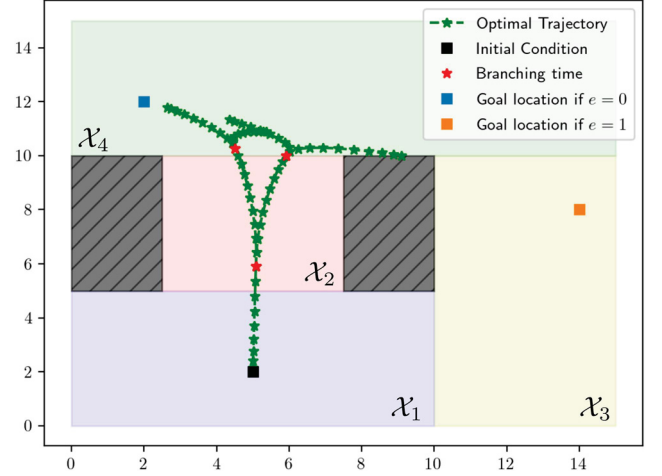


Fig. 4. Optimal trajectory computed solving the MICQ for $N = 30$ and $N_b = 10$. The objective is to steer the system to the goal location that is a function of the partially observable state $e \in \{0, 1\}$, while avoiding the two obstacles (black rectangles). In this scenario, the initial belief $b(0) = [0.8, 0.2]^T$, and the observation model is piecewise over the regions \mathcal{X}_1 , \mathcal{X}_2 , \mathcal{X}_3 , and \mathcal{X}_4 .

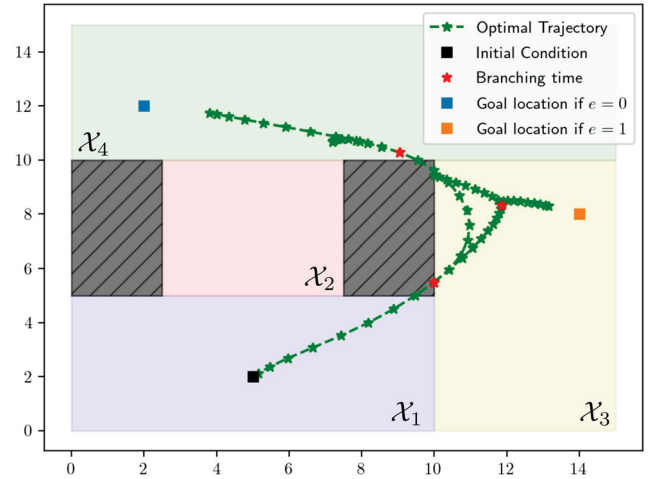


Fig. 5. Optimal trajectory computed solving the MICQ for $N = 30$ and $N_b = 10$. The objective is to steer the system to the goal location that is a function of the partially observable state $e \in \{0, 1\}$, while avoiding the two obstacles (black rectangles). In this scenario, the initial belief $b(0) = [0.5, 0.5]^T$, and the observation model is piecewise over the regions \mathcal{X}_1 , \mathcal{X}_2 , \mathcal{X}_3 , and \mathcal{X}_4 .

B. Partially Observable Navigation Problem

We test the proposed strategy on the navigation task shown in Figs. 4 and 5. In this example, there are two obstacles (black regions), and the objective is to reach a goal location that can only be inferred through partial observations. The observation model is piecewise and is defined as follows:

$$Z(o = e, e, x) = \mathbb{P}(o = e|e, x) = \begin{cases} p_1 = 0.5 & \text{If } x \in \mathcal{X}_1 \\ p_2 = 0.7 & \text{If } x \in \mathcal{X}_2 \\ p_3 = 0.85 & \text{If } x \in \mathcal{X}_3 \\ p_4 = 0.85 & \text{If } x \in \mathcal{X}_4 \end{cases} \quad (21)$$

where regions \mathcal{X}_1 , \mathcal{X}_2 , \mathcal{X}_3 , and \mathcal{X}_4 are shown in Figs. 4 and 5. Less formally, the observation function in (21) models the accuracy of the sensors that are more accurate when the system is close to the candidate

goal location and there is no occlusion caused by the obstacles. Indeed, observations collected in region \mathcal{X}_1 are not informative; on the other hand, in region \mathcal{X}_2 , the probability that an observation is correct is $p_2 = 0.7$, and the most informative observations are collected in regions \mathcal{X}_3 and \mathcal{X}_4 . Finally, we consider the unstable point mass model (19) and the cost function is defined by the matrices $Q = 10^{-4}I_n$, $R = 10^{-2}I_d$, and $Q_N = 10I_n$, where $I_n \in \mathbb{R}^{n \times n}$ represents the identity matrix.

We implemented the MICP using CVXPY [46]. Note that the feasible regions are nonconvex as there are two obstacles in the environment. For this reason, at time k , we introduced integer variables to constraint the state of the system x_k to lie in either \mathcal{X}_1 , \mathcal{X}_2 , \mathcal{X}_3 , or \mathcal{X}_4 . For implementation details, refer to the source code available at github.²

We tested the proposed strategy for two initial belief vectors. In both scenarios, we set a prediction horizon $N = 30$ and the parameter $N_b = 10$. Therefore, the computed optimal trajectory is solving the MICP branches at time $t = 10$ and time $t = 20$. Fig. 4 shows the optimal trajectory tree when the initial belief $b(0) = [0.8, 0.2]^T$. Note that, as we have a strong belief that the environment state $e = 0$, the controller plans a trajectory tree that goes through region \mathcal{X}_2 to reach the goal location associated with the state $e = 0$. On the other hand, when the initial belief $b(0) = [0.5, 0.5]^T$, the optimizer plans a trajectory that collects observations only in regions \mathcal{X}_3 and \mathcal{X}_4 , as shown in Fig. 5. This result is expected as when we do not have any prior knowledge about the goal location—in this example, $b(0) = [0.5, 0.5]^T$ —an optimal strategy should maximize the number of informative observations that are collected in regions \mathcal{X}_3 and \mathcal{X}_4 .

VI. CONCLUSION

In this work, we introduced the mixed-observable constrained linear quadratic regulator problem, where the goal of the controller is to steer the system to a goal location that may be inferred only through partial observations. We showed that when the system's state space is continuous and the environment's state is discrete, the control problem can be reformulated as a finite-dimensional optimization problem over a trajectory tree. Leveraging this result, we showed that under mild assumptions, the control problem can be recast as an MICP through a nonlinear change of coordinates.

REFERENCES

- [1] M. Morari and J. H. Lee, "Model predictive control: Past, present and future," *Comput. Chem. Eng.*, vol. 23, no. 4-5, pp. 667–682, 1999.
- [2] F. Allgower et al., "Nonlinear model predictive control: From theory to application," *J.-Chin. Inst. Of Chem. Engineers*, vol. 35, no. 3, pp. 299–316, 2004.
- [3] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, "Survey constrained model predictive control: Stability and optimality," *Automatica (J. IFAC)*, vol. 36, no. 6, pp. 789–814, 2000.
- [4] F. Borrelli, A. Bemporad, and M. Morari, *Predictive Control for Linear and Hybrid Systems*. Cambridge, U.K.: Cambridge Univ. Press, 2017.
- [5] Y. Wang and S. Boyd, "Fast model predictive control using online optimization," *IEEE Trans. Control Syst. Technol.*, vol. 18, no. 2, pp. 267–278, Mar. 2010.
- [6] R. Lopez-Negrete, F. J. D'Amato, L. T. Biegler, and A. Kumar, "Fast nonlinear model predictive control: Formulation and industrial process applications," *Comput. Chem. Eng.*, vol. 51, pp. 55–64, 2013.
- [7] J. L. Jerez, P. J. Goulart, S. Richter, G. A. Constantinides, E. C. Kerrigan, and M. Morari, "Embedded online optimization for model predictive control at megahertz rates," *IEEE Trans. Autom. Control*, vol. 59, no. 12, pp. 3238–3251, Dec. 2014.
- [8] D. Kouzoupis, G. Frison, A. Zanelli, and M. Diehl, "Recent advances in quadratic programming algorithms for nonlinear model predictive control," *Vietnam J. Math.*, vol. 46, no. 4, pp. 863–882, 2018.
- [9] A. Bemporad and V. V. Naik, "A numerically robust mixed-integer quadratic programming solver for embedded hybrid model predictive control," *IFAC-PapersOnLine*, vol. 51, no. 20, pp. 412–417, 2018.
- [10] S. Gros, M. Zanon, R. Quirynen, A. Bemporad, and M. Diehl, "From linear to nonlinear MPC: Bridging the gap via the real-time iteration," *Int. J. Control*, vol. 93, no. 1, pp. 62–80, 2020.
- [11] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat, "Predictive active steering control for autonomous vehicle systems," *IEEE Trans. Control Syst. Technol.*, vol. 15, no. 3, pp. 566–580, May 2007.
- [12] D. Hrovat, S. Di Cairano, H. E. Tseng, and I. V. Kolmanovsky, "The development of model predictive control in automotive industry: A survey," in *Proc. IEEE Int. Conf. Control Appl.*, 2012, pp. 295–302.
- [13] P. F. Lima, G. C. Pereira, J. Mårtensson, and B. Wahlberg, "Experimental validation of model predictive control stability for autonomous driving," *Control Eng. Pract.*, vol. 81, pp. 244–255, 2018.
- [14] S. Bengea, A. Kelman, F. Borrelli, R. Taylor, and S. Narayanan, "Model predictive control for mid-size commercial building HVAC: Implementation, results and energy savings," in *Proc. 2nd Int. Conf. Building Energy Environ.*, 2012, pp. 979–986.
- [15] G. Serale, M. Fiorentini, A. Capozzoli, D. Bernardini, and A. Bemporad, "Model predictive control (MPC) for enhancing building and HVAC system energy efficiency: Problem formulation, applications and opportunities," *Energies*, vol. 11, no. 3, 2018, Art. no. 631.
- [16] E. T. Maddalena, Y. Lian, and C. N. Jones, "Data-driven methods for building control—A review and promising future directions," *Control Eng. Pract.*, vol. 95, 2020, Art. no. 104211.
- [17] P. O. Scokaert and D. Q. Mayne, "Min-max feedback model predictive control for constrained linear systems," *IEEE Trans. Autom. Control*, vol. 43, no. 8, pp. 1136–1142, Aug. 1998.
- [18] P. J. Goulart, E. C. Kerrigan, and J. M. Maciejowski, "Optimization over state feedback policies for robust control with constraints," *Automatica*, vol. 42, no. 4, pp. 523–533, 2006.
- [19] Y.-S. Wang, N. Matni, and J. C. Doyle, "A system-level approach to controller synthesis," *IEEE Trans. Autom. Control*, vol. 64, no. 10, pp. 4079–4093, Oct. 2019.
- [20] L. Chisci, J. A. Rossiter, and G. Zappa, "Systems with persistent disturbances: Predictive control with restricted constraints," *Automatica*, vol. 37, no. 7, pp. 1019–1028, 2001.
- [21] D. Q. Mayne, M. M. Seron, and S. Raković, "Robust model predictive control of constrained linear systems with bounded disturbances," *Automatica*, vol. 41, no. 2, pp. 219–224, 2005.
- [22] S. Yu, C. Maier, H. Chen, and F. Allgöwer, "Tube MPC scheme based on robust control invariant set with application to Lipschitz nonlinear systems," *Syst. Control Lett.*, vol. 62, no. 2, pp. 194–200, 2013.
- [23] J. Fleming, B. Kouvaritakis, and M. Cannon, "Robust tube MPC for linear systems with multiplicative uncertainty," *IEEE Trans. Autom. Control*, vol. 60, no. 4, pp. 1087–1092, Apr. 2015.
- [24] A. Liniger, X. Zhang, P. Aeschbach, A. Georgiou, and J. Lygeros, "Racing miniature cars: Enhancing performance using stochastic MPC and disturbance feedback," in *Proc. Amer. Control Conf.*, 2017, pp. 5642–5647.
- [25] A. Ben-Tal, A. Goryashko, E. Guslitzer, and A. Nemirovski, "Adjustable robust solutions of uncertain linear programs," *Math. Program.*, vol. 99, no. 2, pp. 351–376, 2004.
- [26] D. Q. Mayne, S. V. Raković, R. Findeisen, and F. Allgöwer, "Robust output feedback model predictive control of constrained linear systems," *Automatica*, vol. 42, no. 7, pp. 1217–1222, 2006.
- [27] I. Alvarado, D. Limón, T. Alamo, and E. F. Camacho, "Output feedback robust tube based MPC for tracking of piece-wise constant references," in *Proc. IEEE 46th Conf. Decis. Control*, 2007, pp. 2175–2180.
- [28] M. Cannon, Q. Cheng, B. Kouvaritakis, and S. V. Raković, "Stochastic tube MPC with state estimation," *Automatica*, vol. 48, no. 3, pp. 536–541, 2012.
- [29] L. Svensson et al., "Safe stop trajectory planning for highly automated vehicles: An optimal control problem formulation," in *Proc. IEEE Intell. Veh. Symp.*, 2018, pp. 517–522.
- [30] I. Batkovic, U. Rosolia, M. Zanon, and P. Falcone, "A robust scenario MPC approach for uncertain multi-modal obstacles," *IEEE Contr. Syst. Lett.*, vol. 5, no. 3, pp. 947–952, Jul. 2021.
- [31] Y. Chen, U. Rosolia, W. Ubellacker, N. Csomay-Shanklin, and A. D. Ames, "Interactive multi-modal motion planning with branch model predictive control," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 5365–5372, Apr. 2022.
- [32] S. H. Nair, V. Govindarajan, T. Lin, C. Meissen, H. E. Tseng, and F. Borrelli, "Stochastic MPC with multi-modal predictions for traffic intersections," 2021, *arXiv:2109.09792*.

²[Online]. Available: <https://github.com/urosolia/mixed-observable-LQR>

- [33] P. Sopasakis, D. Hecceg, A. Bemporad, and P. Patrinos, "Risk-averse model predictive control," *Automatica*, vol. 100, pp. 281–288, 2019.
- [34] J. P. Alsterda and J. C. Gerdes, "Contingency model predictive control for linear time-varying systems," 2021, *arXiv:2102.12045*.
- [35] J. P. Alsterda, M. Brown, and J. C. Gerdes, "Contingency model predictive control for automated vehicles," in *Proc. Amer. Control Conf.*, 2019, pp. 717–722.
- [36] D. Liu, S. Xue, B. Zhao, B. Luo, and Q. Wei, "Adaptive dynamic programming for control: A survey and recent advances," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 51, no. 1, pp. 142–160, Jan. 2021.
- [37] F.-Y. Wang, H. Zhang, and D. Liu, "Adaptive dynamic programming: An introduction," *IEEE Comput. Intell. Mag.*, vol. 4, no. 2, pp. 39–47, May 2009.
- [38] J. J. Murray, C. J. Cox, G. G. Lendaris, and R. Sacks, "Adaptive dynamic programming," *IEEE Trans. Syst., Man, Cybern., Part C (Appl. Rev.)*, vol. 32, no. 2, pp. 140–153, May 2002.
- [39] V. Krishnamurthy, *Partially Observed Markov Decision Processes*. Cambridge, U.K.: Cambridge Univ. Press, 2016.
- [40] S. C. Ong, S. W. Png, D. Hsu, and W. S. Lee, "POMDPs for robotic tasks with mixed observability," *Robot., Sci. Syst.*, vol. 5, 2009, Art. no. 4.
- [41] S. Daftary et al., "MLNav: Learning to safely navigate on martian terrains," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 5461–5468, Apr. 2022.
- [42] U. Rosolia, M. Ahmadi, R. M. Murray, and A. D. Ames, "Time-optimal navigation in uncertain environments with high-level specifications," in *Proc. IEEE 60th Conf. Decis. Control*, 2021, pp. 4287–4294.
- [43] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
- [44] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407–427, 1999.
- [45] A. Bemporad and C. Filippi, "An algorithm for approximate multiparametric convex programming," *Comput. Optim. Appl.*, vol. 35, no. 1, pp. 87–108, 2006.
- [46] S. Diamond and S. Boyd, "CVXPY: A Python-embedded modeling language for convex optimization," *J. Mach. Learn. Res.*, vol. 17, no. 83, pp. 1–5, 2016.
- [47] *Gurobi Optimizer Reference Manual*, Gurobi Optimization, LLC, Houston, TX, USA, 2021. [Online]. Available: <https://www.gurobi.com>