

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

Ugo Rosolia, Yuxiao Chen, Shreyansh Daftry, Masahiro Ono, Yisong Yue, and Aaron D. Ames

Abstract—This paper studies the problem of steering a linear time-invariant system subject to state and input constraints towards a goal location that may be inferred only through 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 (MICP) 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 system has to reach a goal location identified from partial observations.

Index Terms—Optimal control, observability, measurement uncertainty.

I. INTRODUCTION

Advances in optimization theory and the availability of open-source optimization solvers has revolutionized the field of control theory over the past three decades. Modern control algorithms leverage this technology to compute control actions based on sensors measurements [1]–[7]. This strategy allows the controller to constantly adapt and re-plan when disturbances and unexpected events occur, i.e., feedback is naturally incorporated into the system.

Model Predictive Control (MPC) is a mature control technology that was enabled by recent developments in optimization solvers [8]–[13]. In MPC, at each time step an optimal planned trajectory and the associated sequence of control actions are 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 entire 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 [14]–[16] to large scale power systems [17]–[19].

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 the future evolution of the system may be affected by external random phenomena and/or only a state estimate may be available. 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 future trajectory of the system using a policy that is a function mapping the system’s state to a control action. Unfortunately planning over policies is non-polynomial time, even for the constrained linear quadratic regulator problem when disturbances affect the system’s dynamics [20].

Several strategies have been presented in the literature to ease the computational burden of planning over policies [21]–[27]. When the system dynamics are subjected to disturbance 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. This strategy was originally presented in the operational research community [28] and it was analysed in an MPC framework in [21], where the authors showed that there is a one-to-one mapping between affine disturbance feedback and affine state feedback policies. The idea of considering policies that map disturbances to actions was leveraged also in the system-level synthesis approach presented in [26]. Another class of feedback policies is considered in tube MPC strategies [22]–[25], 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 when only partial observations about the system’s state are available [29]–[31]. In this case, the system’s state is not known exactly and estimation errors should be considered when computing control actions.

The above mentioned strategies are designed for uni-modal disturbances and measurements noise. When the uncertainties are multi-modal, the controller should plan for different modes of operations to reduce conservatism. For instance, in autonomous driving a controller should plan a trajectory for nominal road conditions, but it should make sure that a safe backup maneuver is available when unexpected events

Ugo Rosolia, Yuxiao Chen, Yisong Yue, and Aaron D. Ames are at the California Institute of Technology, Pasadena, USA. E-mails: {urosolia, chenyx, yue, ames}@caltech.edu. Shreyansh Daftry and Masahiro Ono are with the Jet Propulsion Laboratory, California Institute of Technology, Pasadena, USA. E-mails: {daftry, masahiro.ono}@jpl.nasa.gov. This research was carried out at the Jet Propulsion Laboratory and the California Institute of Technology under a contract with the National Aeronautics and Space Administration. The authors like to acknowledge also the support of the NSF award #1932091.

occur [32]–[34]. 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 controller that can handle multi-model disturbances [33]–[36], when perfect state feedback is available.

In this work, we introduce the mixed-observable constrained linear quadratic regular problem. 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) [37]. The HMM is constructed based on the system and the environment states and it allows us to characterize the observation model that describes the sensors accuracy. In our problem formulation, the system and environment state spaces are assumed continuous and discrete, respectively. For this reason, our approach generalizes the strategy presented in [38], where the authors considered mixed-observable systems with discrete state spaces.

As we consider systems with continuous state and action spaces, the optimal control problem is an infinite-dimensional optimization problem over the space of feedback policies, where the objective is to minimize the expected cost. 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 which can be used to compute a feasible solution to the original problem. 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 examples where the objective is to steer the system to a goal location that can be inferred only through imperfect observations.

Notation: 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 the paper, 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 $\mathcal{Z}^k = \mathcal{Z} \times \dots \times \mathcal{Z}$. 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

This section describes the problem formulation. First, we introduce the system’s dynamics and the discrete model describing the partial observable environment. Afterwards, we introduce the control problem under study.

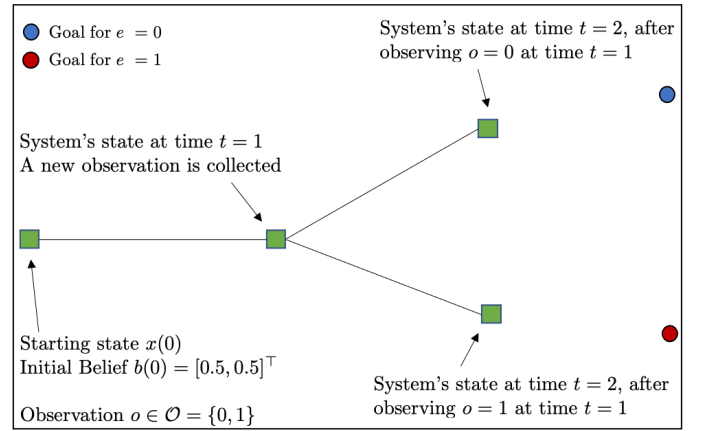


Fig. 1. In the above example, the system has to reach a partially observable goal location that is a function of the partially observable environment state e . Notice the closed-loop trajectory is a function of the observation collected at time step $t = 1$.

A. System and Environment Models

We consider linear time-invariant dynamical systems of the following form:

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

where the state is $x_k \in \mathbb{R}^n$, the input is $u_k \in \mathbb{R}^d$, and k indexes over discrete time steps. Furthermore, the above 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. In particular, the environment evolution is modeled using a hidden Markov model (HMM) given by the tuple $\mathcal{H} = (\mathcal{E}, \mathcal{O}, T, Z)$, where:

- $\mathcal{E} = \{1, \dots, |\mathcal{E}|\}$ is a set of partially observable environment states;
- $\mathcal{O} = \{1, \dots, |\mathcal{O}|\}$ is the set of observations for the partially observable state $e \in \mathcal{E}$;
- 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.,

$$T(e', e, x) := \mathbb{P}(e' | e, x).$$

- The function $Z : \mathcal{E} \times \mathcal{O} \times \mathbb{R}^n \rightarrow [0, 1]$ describes the probability of observing o , given the current 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: $b_k \in \mathcal{B} = \{b \in \mathbb{R}_{0+}^{|\mathcal{E}|} : \sum_{i=1}^{|\mathcal{E}|} b[i] = 1\}$. The belief b_k is a sufficient statistics and each entry $b_k[e]$ represents the posterior probability that the state of the system 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.,

$$b_k[e] = \mathbb{P}(e | \mathbf{o}_k, \mathbf{x}_k, x(0), b(0)).$$

B. Control Objectives

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

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

where the stage cost $h : \mathbb{R}^n \times \mathbb{R}^d \times \mathcal{E} \rightarrow \mathbb{R}$ and the terminal cost $h_N : \mathbb{R}^n \times \mathcal{E} \rightarrow \mathbb{R}$ are functions of the partially observable environment state $e \in \mathcal{E}$, 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 above 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 . An example of a control task that can be modeled using the FTOCP (3) is shown in Figure 1, where the system has to reach a goal location that may be inferred only through partial environment observations.

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 carried out over the space of feedback policies, which are functions mapping states to inputs. Thus, the optimization problem is infinite-dimensional as a function cannot be represented by finite number of parameters. In what follows, we show that the FTOCP (3) can be reformulated as a finite-dimensional non-linear program (NLP). Leveraging the discrete nature of the set of observations \mathcal{O} , we will show that *i*) optimizing over feedback policies is equivalent to optimizing over a tree of open-loop actions and *ii*) the expectation can be rewritten as a summation using the belief vector. Furthermore, we show that when the environment is static, the cost functions $h(\cdot, \cdot, e)$ and $h_N(\cdot, e)$ are 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 a Mixed Integer Convex Program (MICP).

III. THE EXACT SOLUTION

This section shows how to reformulate the FTOCP (3) as a finite-dimensional optimization problem.

A. Cost Reformulation

As discussed in Section II-A, the belief b_k is a sufficient statistics for an HMM [37]. 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.,

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

For more details about the belief update equation please refer to the Appendix VII-A. The above equation can be written in compact form:

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

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:

$$A_e(o_k, x_k) = \Theta(o_k, x_k) \Omega(x_k), \tag{5}$$

where

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

and

$$\Theta(o_k, x_k) = \text{diag} \left([Z(1, o_k, x_k) \quad \dots \quad Z(|\mathcal{E}|, o_k, x_k)] \right).$$

Notice that, given the initial environment belief $b_0 = b(0)$, the expected stage cost at time $k = 0$ is simply

$$\mathbb{E}_{\mathbf{o}_{N-1}} [h(x_0, u_0, e_0) | b_0] = \sum_{e \in \mathcal{E}} b_0[e] h(x_0, u_0, e).$$

Next, we show that the belief update equation (4) can be used to rewrite the expectation of the stage cost as a summation. The evolution of the system from problem (3) is deterministic, therefore the expected cost at time $k = 1$ can be written as

$$\begin{aligned}
& \mathbb{E}_{\mathbf{o}_{N-1}} [h(x_1, u_1, e_1) | b_0, x_1] \\
&= \sum_{o_1 \in \mathcal{O}} \mathbb{E}_{\mathbf{o}_{N-1}} [h(x_1, u_1, e_1) | b_0, x_1, o_1] \mathbb{P}(o_1 | x_1, b_0) \\
&= \sum_{o_1 \in \mathcal{O}} \sum_{e \in \mathcal{E}} b_1[e] h(x_1, u_1, e) \mathbb{P}(o_1 | x_1, b_0) \\
&= \sum_{o_1 \in \mathcal{O}} \sum_{e \in \mathcal{E}} Z(e, o_1, x_1) \sum_{i \in \mathcal{E}} T(e, i, x_1) b_0[i] h(x_1, u_1, e) \\
&= \sum_{o_1 \in \mathcal{O}} \sum_{e \in \mathcal{E}} v_1^{o_1}[e] h(x_1, u_1, e)
\end{aligned} \tag{6}$$

where the unnormalized belief is

$$v_1^{o_1}[e] = Z(e, o_1, x_1) \sum_{i \in \mathcal{E}} T(e, i, x_1) b_0[i].$$

In equation (6), we first rewrote the expectation of the stage cost at time $k = 1$ leveraging the belief b_1 , which is a function of the observation o_1 , the system's state x_1 , and the belief b_0 . Then, we substituted the belief update equation (4). It is important to underline that the expected cost in (6) is not a function of the normalization constant $\mathbb{P}(o_1|x_1, b_0)$. This fact will allow us to show that problem (3) can be reformulated as a finite-dimensional MICP that can be solved with a branch-and-bound algorithm, when the environment is static, the cost function is convex, and the observation model is a piecewise.

The following proposition generalizes the result from (6). In particular, we show that the expected cost from problem (3)

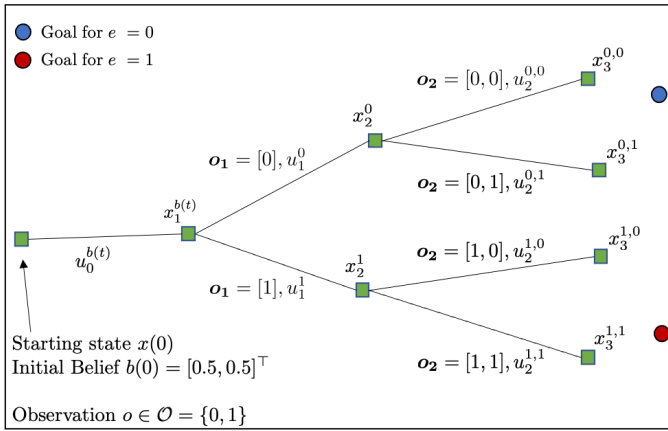


Fig. 2. Tree of trajectories for $N = 3$, where at each time k there are $|\mathcal{O}| = 2$ possible observations. Each node represents the system's predicted state $x_k^{\mathbf{o}_k}$ and each edge the control action $u_k^{\mathbf{o}_k}$, which is applied when the observation vector \mathbf{o}_k is measured.

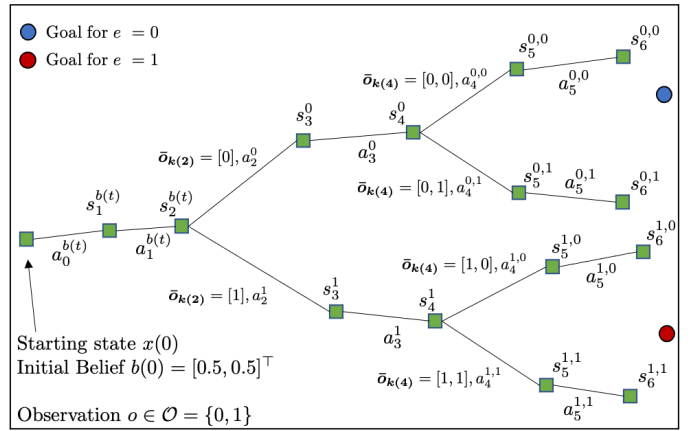


Fig. 3. Tree of trajectories for $N = 6$, where every $N_b = 2$ time steps $|\mathcal{O}|$ possible observations may be collected. Each node represents the predicted state $s_k^{\mathbf{o}_k^{(k)}}$ of the system and each edge the control action $u_k^{\mathbf{o}_k^{(k)}}$, which is applied when the observation vector $\bar{\mathbf{o}}_k^{(k)}$ is measured.

can be rewritten as a summation over the discrete set of possible observations \mathcal{O}^k .

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

Proof: First we notice 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 above 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

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

which can be written in compact form as $v_k^{\mathbf{o}_k} = A_e(o_k, x_k) v_{k-1}^{\mathbf{o}_k}$. Finally, we notice 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 equation (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 out over feedback policies can be reformulated as an optimization over a tree of open-loop trajectories.

The control policy $\pi_k : \mathcal{O}^k \times \mathcal{X}^{k+1} \times \mathcal{B}$ 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.,

$$u_k = \pi(\mathbf{o}_k, \mathbf{x}_k, x_0, b_0).$$

Notice 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 observations vector \mathbf{o}_k . Thus, we define the control action $u_k^{\mathbf{o}_k} \in \mathbb{R}^n$ associated with the observations vector $\mathbf{o}_k \in \mathcal{O}^k$, and we reformulate problem (3) as an optimization problem carried out over the tree of control actions $u_k^{\mathbf{o}_k}$ for all $\mathbf{o}_k \in \mathcal{O}^k$ and $k \in \{0, \dots, T\}$. Figure 2 shows a trajectory tree associated with a tree of control actions, where there are $|\mathcal{O}| = 2$ possible observations. More formally, given the environment belief $b(t)$ and the system's state $x(t)$, we rewrite problem (3) as the following FTOCP:

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

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 above problem the matrix of decision variables is

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

where at time k the control action $u_k^{\mathbf{o}_k} \in \mathbb{R}^d$ is associated with the observation vector $\mathbf{o}_k \in \mathcal{O}^k$.

C. Practical Approach

The FTOCP (9) is a finite-dimensional NLP that can be solved with off-the-self solvers. However, the computational cost of solving (9) is non-polynomial 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 Figure 2. In this section, we introduce an approximation to the FTOCP (9), where the predicted trajectory branches every N_b time steps, as shown in Figure 3. 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 . In what follows, we first introduce an approximation to the FTOCP (9). Then, we show that such approximation yields an upper-bound to the cost of the FTOCP (9).

Given the current state $x(t)$, the environment 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:

$$\begin{aligned} & \hat{J}(x(t), b(t)) \\ &= \min_{\mathbf{a}} \sum_{k=0}^{N-1} \sum_{\bar{\mathbf{o}}_{j(k)} \in \mathcal{O}^{j(k)}} \sum_{e \in \mathcal{E}} \bar{v}_k^{\bar{\mathbf{o}}_{j(k)}}[e] h(s_k^{\bar{\mathbf{o}}_{j(k)}}, a_k^{\bar{\mathbf{o}}_{j(k)}}, e) \\ & \quad + \sum_{\bar{\mathbf{o}}_{j(N)} \in \mathcal{O}^{j(N)}} \sum_{e \in \mathcal{E}} \bar{v}_k^{\bar{\mathbf{o}}_{j(N)}}[e] h_N(s_N^{\bar{\mathbf{o}}_{j(N)}}, e) \\ & \text{s.t. } s_{k+1}^{\bar{\mathbf{o}}_{j(k)}} = A s_k^{\bar{\mathbf{o}}_{j(k-1)}} + B a_k^{\bar{\mathbf{o}}_{j(k)}}, \\ & \quad x_0^{\bar{\mathbf{o}}_{j(0)}} = x(t), \bar{v}_0^{\bar{\mathbf{o}}_0} = b(t), \\ & \quad \bar{v}_{k+1}^{\bar{\mathbf{o}}_{j(k+1)}} = C_e(\bar{o}_{j(k+1)}, s_{k+1}^{\bar{\mathbf{o}}_{j(k)}}, k) \bar{v}_k^{\bar{\mathbf{o}}_{j(k)}}, \\ & \quad a_k^{\bar{\mathbf{o}}_{j(k)}} \in \mathcal{U}, s_{k+1}^{\bar{\mathbf{o}}_{j(k)}} \in \mathcal{X}, \\ & \quad j(k) = \lfloor k/N_b \rfloor, \\ & \quad \forall \bar{\mathbf{o}}_{j(k)} \in \mathcal{O}^{j(k)}, \forall k \in \{0, \dots, N-1\}, \end{aligned} \quad (11)$$

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

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

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

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

Compare the FTOCP (9) with the FTOCP (11). In the FTOCP (9) we optimize over the tree of open-loop sequences shown in Figure 2, and therefore the complexity of the problem grows exponentially with the horizon length N . Indeed, the matrix of optimization variables (10) collects $\sum_{k=0}^{N-1} |\mathcal{O}|^k = O(|\mathcal{O}|^{N-1})$ control actions. On the other hand, in the FTOCP (11) we optimize over the tree of open-loop sequences shown in Figure 3, and the matrix of optimization variables (12) grows exponentially with the ratio $P = N/N_b$. In particular, we have $N_b \sum_{k=0}^{P-1} |\mathcal{O}|^k = O(|\mathcal{O}|^{P-1})$ decision variables which are collected in the matrix (12). Therefore, in the FTOCP (11) the user-defined constant N_b may be used to limit the computational complexity when planning over a horizon $N = PN_b$. As a trade-off, the optimal value function \hat{J} associated with the FTOCP (11) only approximates the value function J associated with the FTOCP (9). The following theorem shows that the value function \hat{J} is an upper-bound to the value function J .

First, we introduce the set of observation vectors $\mathcal{M}(\bar{\mathbf{o}}_{j(k)}, k)$, which we will leverage to construct a feasible solution to the FTOCP (9) using the optimal solution to the FTOCP (11).

Definition 1. Consider the vectors of observations

$$\mathbf{o}_k = [o_1, \dots, o_k] \text{ and } \bar{\mathbf{o}}_{j(k)} = [\bar{o}_1, \dots, \bar{o}_{j(k)}],$$

where $j(k) = \lfloor k/N_b \rfloor$. Given the above vectors, we define the following set:

$$\begin{aligned} \mathcal{M}(\bar{\mathbf{o}}_{j(k)}, k) &= \{\mathbf{o}_k \in \mathcal{O}^k : \forall t \in \{1, \dots, k\}, \\ & \quad \text{If } \lfloor t/N_b \rfloor = t/N_b, \text{ then} \\ & \quad o_t = \bar{o}_{j(t)}, j(t) = \lfloor t/N_b \rfloor\}. \end{aligned}$$

The above set $\mathcal{M}(\bar{\mathbf{o}}_{j(k)}, k) \subset \mathcal{O}^k$ collects all observation vectors $\mathbf{o}_k \in \mathcal{O}^k$ such that for $t \in \{N_b, 2N_b, \dots, (P-1)N_b\}$ the observation o_t from the tree shown in Figure 2 equals the observation $\bar{o}_{j(t)}$ from the tree depicted in Figure 3. Less formally, we have that subset of observations collected every N_b time steps $\{o_{N_b}, o_{2N_b}, \dots, o_{(P-1)N_b}\}$ in Figure 2 are equal to the observation measured every N_b time steps $\{\bar{o}_1, \bar{o}_2, \dots, \bar{o}_{P-1}\}$ from Figure 3, as discussed in the following example.

Example 1. Let $N = 4$, $N_b = 2$, $P = N/N_b = 2$, and $\mathcal{O} \in \{0, 1\}$. Then, the vectors of observations $\mathbf{o}_{N-1} = [o_1, o_2, o_3] \in \mathcal{O}^3$, $\mathbf{o}_{j(N-1)} = \mathbf{o}_{P-1} = \bar{\mathbf{o}}_1 \in \mathcal{O}$ and the sets

$$\begin{aligned} \mathcal{M}(0, N-1) &= \{[0, 0, 0], [1, 0, 0], [0, 0, 1], [1, 0, 1]\} \\ \mathcal{M}(1, N-1) &= \{[0, 1, 0], [1, 1, 0], [0, 1, 1], [1, 1, 1]\}. \end{aligned}$$

Theorem 1. Let $\mathcal{C} \subseteq \mathbb{R}^n \times \mathcal{B}$ be the set of initial conditions $(x(t), b(t))$ such that problem (9) is feasible. Then, we have that for all $(x, b) \in \mathcal{C}$ the value function (11) is an upper-bound to the value function (9), i.e.,

$$J(x, b) \leq \hat{J}(x, b),$$

for all $(x, b) \in \mathcal{C}$.

Proof: We show that an optimal solution to problem (11) can be used to construct a feasible solution to problem (9).

Let

$$\mathbf{a}^* = [a_0^{\bar{o}_0,*}, \dots, a_{N_b-1}^{\bar{o}_{N_b-1,*}}, \dots, a_k^{\bar{o}_{j(k),*}}, \dots, a_{N-1}^{\bar{o}_{j(N-1),*}}] \quad (14)$$

be the optimal sequence of inputs to problem (11), we define the following candidate solution:

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

where for all $k \in \{0, \dots, N-1\}$ the control action

$$u_k^{\mathbf{o}_k,c} = a_k^{\bar{o}_{j(k),*}}, \forall \mathbf{o}_k \in \mathcal{M}(\bar{o}_{j(k)}, k), \forall \bar{o}_{j(k)} \in \mathcal{O}^{j(k)}. \quad (16)$$

Basically, the above sequence of candidate inputs are defined only by the observations collected at time $t \in \{N_b, 2N_b, \dots, (P-1)N_b\}$. As a result, when the above sequence of candidate inputs is applied to system (1) several branches of the trajectory tree from Figure 2 overlap and the tree collapses to the one shown in Figure 3. Therefore, the definition of the candidate inputs (16) implies that for all $k \in \{0, \dots, N\}$ we have that

$$s_k^{\bar{o}_{j(k),*}} = x_k^{\mathbf{o}_k,c}, \forall \mathbf{o}_k \in \mathcal{M}(\bar{o}_{j(k)}, k), \forall \bar{o}_{j(k)} \in \mathcal{O}^{j(k)}, \quad (17)$$

where $s_k^{\mathbf{o}_k,*} \in \mathbb{R}^n$ is the state associated with the optimal input sequence (14) and $x_k^{\mathbf{o}_k,c} \in \mathbb{R}^n$ is the state associated with the candidate solution (15). Now notice that the system dynamics, the input constraints, and the state constraints in problems (9) and (11) are independent of the belief vector $b(t)$. Therefore, the feasibility of the optimal solution (14) for problem (11) implies the feasibility of the candidate inputs (15) and associated states (17) for problem (9).

Finally, we show that the cost associated with the feasible solution (15) to problem (9) equals the optimal cost from (11). From definitions (5) and (13), we have that

$$C_e(o, s_k^{\bar{o}_{j(k),*}}, k) = A_e(o, x_k^{\mathbf{o}_k,c}), \quad (18)$$

for all $o \in \mathcal{O}$ and for all $k \in \{N_b, 2N_b, \dots, (P-1)N_b\}$, and

$$C_e(\bar{o}, s_k^{\bar{o}_{j(k),*}}, k) = \Omega(s_k^{\bar{o}_{j(k),*}}) = \sum_{o \in \mathcal{O}} A_e(o, x_k^{\mathbf{o}_k,c}) \quad (19)$$

for all $\bar{o} \in \mathcal{O}$ and for all $k \in \{1, \dots, N\}$ such that $[i/N_b] \neq i/N_b$. Next, we define the following unnormalized belief vectors:

$$\bar{v}_k^{\bar{o}_{j(k),*}} \in \mathcal{B} \subset \mathbb{R}^{|\mathcal{E}|} \text{ and } v_k^{\mathbf{o}_k,c} \in \mathcal{B} \subset \mathbb{R}^{|\mathcal{E}|},$$

which are associated with the optimal sequence of inputs (14) and the feasible sequence of inputs (15), respectively. Next, we show by induction that

$$\bar{v}_k^{\bar{o}_{j(k),*}} = \sum_{\mathbf{o}_k \in \mathcal{M}(\bar{o}_{j(k)}, k)} \bar{v}_k^{\mathbf{o}_k,c}. \quad (20)$$

Assume that the above equation (20) holds, then from equation (18) we have that for $k+1 \in \{N_b, 2N_b, \dots, (P-1)N_b\}$, for all $\mathbf{o}_k \in \mathcal{M}(\bar{o}_{j(k)}, k)$ and for all $\bar{o}_{j(k)} \in \mathcal{O}^{j(k)}$

$$\begin{aligned} \bar{v}_{k+1}^{\bar{o}_{j(k+1),*}} &= C_e(o_{k+1}, s_{k+1}^{\bar{o}_{j(k+1),*}}, k+1) \bar{v}_k^{\bar{o}_{j(k),*}} \\ &= A_e(o_{k+1}, s_{k+1}^{\bar{o}_{j(k+1),*}}) \bar{v}_k^{\bar{o}_{j(k),*}} \\ &= A_e(o_{k+1}, x_{k+1}^{\mathbf{o}_k,c}) \sum_{\mathbf{o}_k \in \mathcal{M}(\bar{o}_{j(k)}, k)} \bar{v}_k^{\mathbf{o}_k,c} \\ &= \sum_{\mathbf{o}_{k+1} \in \mathcal{M}(\bar{o}_{j(k+1)}, k+1)} v_{k+1}^{\mathbf{o}_{k+1},c}. \end{aligned} \quad (21)$$

Furthermore when equation (20) holds, from equation (19) we have that for all $\mathbf{o}_k \in \mathcal{M}(\bar{o}_{j(k)}, k)$ and for all $\bar{o}_{j(k)} \in \mathcal{O}^{j(k)}$

$$\begin{aligned} \bar{v}_{k+1}^{\bar{o}_{j(k+1),*}} &= C_e(o_{k+1}, s_{k+1}^{\bar{o}_{j(k+1),*}}, k+1) \bar{v}_k^{\bar{o}_{j(k),*}} \\ &= \sum_{o_{k+1} \in \mathcal{O}} A_e(o_{k+1}, s_{k+1}^{\bar{o}_{j(k+1),*}}) \bar{v}_k^{\bar{o}_{j(k),*}} \\ &= \sum_{o_{k+1} \in \mathcal{O}} A_e(o_{k+1}, x_{k+1}^{\mathbf{o}_k,c}) \sum_{\mathbf{o}_k \in \mathcal{M}(\bar{o}_{j(k)}, k)} \bar{v}_k^{\mathbf{o}_k,c} \\ &= \sum_{\mathbf{o}_{k+1} \in \mathcal{M}(\bar{o}_{j(k+1)}, k+1)} v_{k+1}^{\mathbf{o}_{k+1},c}, \end{aligned} \quad (22)$$

for $k+1 \in \{1, \dots, N\}$ such that $[(i+1)/N_b] \neq (i+1)/N_b$. Now notice that by definition we have that $\bar{v}_0^{\bar{o}_0} = v_0^{\mathbf{o}_0}$, thus from equations (21)–(22) we conclude by induction that equations (20) holds for all $k \in \{1, \dots, N-1\}$. Finally, we have that the equation (20), together with equations (16) and (17), implies that

$$\begin{aligned} &\sum_{\bar{o}_{j(k)} \in \mathcal{O}^{j(k)}} \sum_{e \in \mathcal{E}} \bar{v}_k^{\bar{o}_{j(k),*}} [e] h(s_k^{\bar{o}_{j(k),*}}, a_k^{\bar{o}_{j(k),*}}, e) = \\ &= \sum_{\bar{o}_{j(k)} \in \mathcal{O}^{j(k)}} \sum_{e \in \mathcal{E}} \sum_{\mathbf{o}_k \in \mathcal{M}(\bar{o}_{j(k)}, k)} v_k^{\mathbf{o}_k,c} [e] h(x_k^{\mathbf{o}_k,c}, u_k^{\mathbf{o}_k,c}, e) \\ &= \sum_{\mathbf{o}_k \in \mathcal{O}^k} \sum_{e \in \mathcal{E}} v_k^{\mathbf{o}_k,c} [e] h(x_k^{\mathbf{o}_k,c}, u_k^{\mathbf{o}_k,c}, e) \end{aligned} \quad (23)$$

and, therefore, the cost associated with the feasible solution (15) equals the cost associated with the optimal solution (14), thus we conclude that

$$J(x, b) \leq \hat{J}(x, b),$$

for all $(x, b) \in \mathcal{C}$. ■

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 cost function. Under these assumptions, we show that problem (9) can be reformulated as a Mixed-Integer Convex Program (MICP) that can be solved to global optimality using the branch-and-bound algorithm.

In what follows, we first introduce the problem set-up. Then, we show how to reformulated problem (9) as a MICP.

Assumption 1 (Static Environment). The environment is static, which in turns implies that the transition function T is defined as follows:

$$T(e, e) = 1, T(e', e) = 0, \forall e \in \mathcal{E}, \forall e' \in \mathcal{E}, e \neq e'.$$

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:

$$Z(e, o, x) = M_i(e, o) \text{ if } x \in \mathcal{X}_i,$$

for a set of functions $M_i : \mathcal{E} \times \mathcal{O} \rightarrow [0, 1]$.

Assumption 3 (Convex 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 quadratic, i.e.,

$$\begin{aligned} h(x, u, e) &= \|x - x_g^{(e)}\|_Q + \|u - u_g^{(e)}\|_R, \\ h_N(x, e) &= \|x - x_g^{(e)}\|_{Q_N} \end{aligned}$$

where the weighted square norm $\|x\|_Q = x^\top Q x$, and the vectors $x_g^{(e)} \in \mathbb{R}^n$ and $u_g^{(e)} \in \mathbb{R}^d$ are defined by the user.

Assumption 4 (Strictly Positive Belief). All entries of the belief vector $b(t)$ are strictly positive, i.e., $b(t) \in \mathcal{B}_+ = \{b \in \mathbb{R}_{0+}^{|\mathcal{E}|} : \sum_{i=1}^{|\mathcal{E}|} b[e] = 1\}$. 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:

$$\begin{aligned} V(x(t), z(t)) &= \min_{\mathbf{u}, \delta} \sum_{k=0}^{N-1} \sum_{\mathbf{o}_k \in \mathcal{O}^k} \sum_{e \in \mathcal{E}} \frac{h(x_k^{\mathbf{o}_k}, u_k^{\mathbf{o}_k}, e)}{z_k^{\mathbf{o}_k}[e]} \\ &\quad + \sum_{\mathbf{o}_N \in \mathcal{O}^N} \sum_{e \in \mathcal{E}} \frac{h_N(x_N^{\mathbf{o}_N}, e)}{z_N^{\mathbf{o}_N}[e]} \\ \text{s.t.} \quad &x_{k+1}^{\mathbf{o}_k} = Ax_k^{\mathbf{o}_k} + Bu_k^{\mathbf{o}_k}, \\ &x_0^{\mathbf{o}_0} = x(t), z_0^{\mathbf{o}_0} = z(t), \\ &u_k^{\mathbf{o}_k} \in \mathcal{U}, x_k^{\mathbf{o}_k} \in \mathcal{X}, \\ &z_{k+1}^{\mathbf{o}_k} = \sum_{i=1}^R D_i(o_{k+1}) z_k^{\mathbf{o}_k} \delta_{k,i}^{\mathbf{o}_k}, \\ &\delta_{k,i}^{\mathbf{o}_k} = \mathbb{1}_{\mathcal{X}_i}(x_k^{\mathbf{o}_k}), \forall i \in \{1, \dots, R\}, \\ &\forall k \in \{0, \dots, N-1\}, \end{aligned} \quad (24)$$

where the indicator function

$$\mathbb{1}_{\mathcal{X}_i}(x_k^{\mathbf{o}_k}) = \begin{cases} 1 & \text{If } x_k^{\mathbf{o}_k} \in \mathcal{X}_i \\ 0 & \text{Otherwise} \end{cases},$$

and optimization variables

$$\begin{aligned} \delta &= [\delta_{0,1}^{\mathbf{o}_0}, \dots, \delta_{N-1,R}^{\mathbf{o}_N}] \in \{0, 1\}^{R \sum_{k=0}^{N-1} |\mathcal{O}|^k}, \\ \mathbf{u} &= [u_0^{\mathbf{o}_0}, \dots, u_{N-1}^{\mathbf{o}_N}] \in \mathbb{R}^{d \times \sum_{k=0}^{N-1} |\mathcal{O}|^k}. \end{aligned} \quad (25)$$

Notice 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 above 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:

$$D_i(o)[e, e] = 1/M_i(o, e), \forall e \in \mathcal{E}, \forall o \in \mathcal{O}. \quad (26)$$

The following theorem shows that, under Assumption 1–4, problem (24) is equivalent to problem (3). Furthermore, problem (24) can be recast as a Mixed-Integer Convex Program (MICP), which can be solved using a branch-and-bound algorithm that computes upper and lower bounds solving convex sub-problems.

Theorem 2. Consider problem (3) and problem (24). Let Assumption 1–4 hold and $b(t) \in \mathcal{B}_+ = \{b \in \mathbb{R}_{0+}^{|\mathcal{E}|} : \sum_{i=1}^{|\mathcal{E}|} b[e] = 1\}$. 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 (24) can be recast as a Mixed-Integer Convex Program (MICP).

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

$$\begin{aligned} v_k^{\mathbf{o}_k}[e] &= Z(e, o_k, x_k) \sum_{i \in \mathcal{E}} T(e, i) v_{k-1}^{\mathbf{o}_k}[i] \\ &= Z(e, o_k, x_k) v_{k-1}^{\mathbf{o}_k}[e] \\ &= M_i(e, o_k) v_{k-1}^{\mathbf{o}_k}[e]. \end{aligned} \quad (27)$$

From the above equation and definition (26), we have that

$$z_k^{\mathbf{o}_k}[e] = 1/v_k^{\mathbf{o}_k}[e], \forall e \in \mathcal{E},$$

which in turns implies that the optimal cost from problem (24) equals the one from problem (9) and therefore

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

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

Notice that the objective function in problem (24) is convex, as it is given by a convex quadratic function over a strictly positive linear function [39]. Furthermore, given the initial condition $z(t)$ we can compute an upper-bound $M_k[e] \in \mathbb{R}_+^{|\mathcal{E}|}$ for each e th entry of the unnormalized belief $z_k^{\mathbf{o}_k}$, i.e.,

$$z_k^{\max}[e] = \left(\max_{o \in \mathcal{O}, i \in \{1, \dots, R\}} D_i(o) \right)^{k-1} z_0^{\mathbf{o}_0}[e] \geq z_k^{\mathbf{o}_k}[e]. \quad (28)$$

Finally, we have the piecewise model from Assumption 2 is a mixed logical dynamical (MLD) systems [40], thus following the procedure presented in [40] problem (24) can be recast as a MICP using the upper-bound from (28). ■

Corollary 1. Consider problem (24) and let Assumptions 1–4 hold. If the belief $b(t) \in \mathcal{B}_+$ and the observation model is not a function of the system's state, i.e.,

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

Then, the value function $V(x(t), z(t))$ from problem (24) is convex in its arguments.

Proof: As the observation model is not dependent on the system's state, we have that problem (24) can be rewritten as

$$\begin{aligned} V(x(t), z(t)) &= \min_{\mathbf{u}} \sum_{k=0}^{N-1} \sum_{\mathbf{o}_k \in \mathcal{O}^k} \sum_{e \in \mathcal{E}} \frac{h(x_k^{\mathbf{o}_k}, u_k^{\mathbf{o}_k}, e)}{z_k^{\mathbf{o}_k}[e]} \\ &\quad + \sum_{\mathbf{o}_N \in \mathcal{O}^N} \sum_{e \in \mathcal{E}} \frac{h_N(x_N^{\mathbf{o}_N}, e)}{z_N^{\mathbf{o}_N}[e]} \\ \text{s.t.} \quad &x_{k+1}^{\mathbf{o}_k} = Ax_k^{\mathbf{o}_k} + Bu_k^{\mathbf{o}_k}, \\ &x_0^{\mathbf{o}_0} = x(t), z_0^{\mathbf{o}_0} = z(t), \\ &u_k^{\mathbf{o}_k} \in \mathcal{U}, x_k^{\mathbf{o}_k} \in \mathcal{X}, \\ &z_{k+1}^{\mathbf{o}_k} = F(o_{k+1}) z_k^{\mathbf{o}_k}, \\ &\forall k \in \{0, \dots, N-1\}, \end{aligned}$$

where $F(o)[e, e] = 1/G(o, e)$ for all $e \in \mathcal{E}$ and $o \in \mathcal{O}$. The above problem is a convex parametric program and therefore $V(x(t), z(t))$ is a convex function [41]. ■

Remark 1. In Corollary 1, we have shown that the value function from problem (24) is convex. On the other hand, we have that the value function from problem (9) is not necessary convex. This fact suggests that it is easier to approximate the value function from problem (24) rather than one from (9). In particular, we have that the nonlinear change of coordinates that we used to define the inverse belief $z(t)$ allows us to convexify the problem, when the environment is static, the observation model constant, and the constraints and cost convex.

Remark 2. The complexity of solving the MICP (24) grows exponentially with the horizon N , and the computational burden may be reduced using the strategy presented in Section III-C, where we assumed that an observation is collected every N_b time steps. In this case, the computational complexity scales with the ratio N/N_b , as we will show in the next section.

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.

A. Mixed Observable Regulation Problem

We consider the following discrete time point mass model:

$$x_{k+1} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} u_k, \quad (29)$$

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 above system the input $u_k = [a_k^x, a_k^y]$ represents the accelerations along the X and Y coordinates. The state and input constraints 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 function matrices from Assumption 3 are

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

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

In this example, the environment is static and the set of partially observable states $\mathcal{E} = \{0, 1\}$. Each environment state is associated with a goal location as discussed in Assumption 3. In particular, we have

$$x_g^{(0)} = \begin{bmatrix} 14 \\ 8 \\ 0 \\ 0 \end{bmatrix} \text{ and } x_g^{(1)} = \begin{bmatrix} 14 \\ -8 \\ 0 \\ 0 \end{bmatrix}.$$

The environment state, and consequently the goal location, is inferred though partial observations. Given the true environment state $e \in \mathcal{E}$ and the state of the system $x \in \mathbb{R}^n$, the

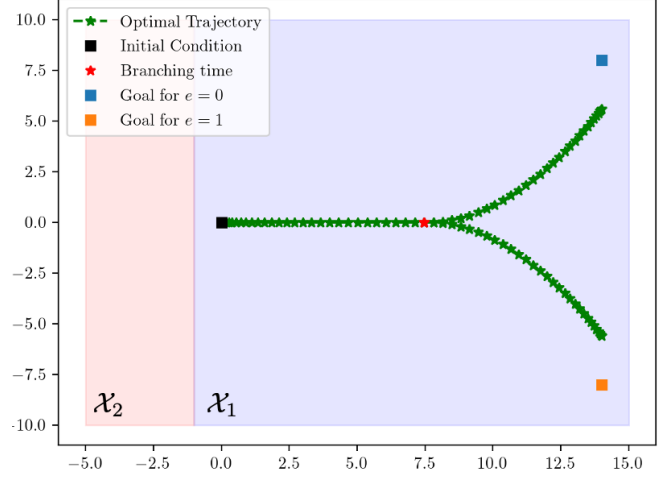


Fig. 4. Optimal trajectory computed solving the MICP (24) for $N = 60$ and assuming that an observation is collected every $N_b = 30$ time steps, as discussed in Section III-C. In this scenario $p_1 = p_2 = 0.85$, therefore the optimizer computes a trajectory that first steers the system towards the goals and then commits to one of the two goal locations depending on observation measured at time $t = N_b$.

probability of measuring the observation $o = e$ is given by the following piecewise observation model:

$$Z(o = e, e, x) = \mathbb{P}(o = e | e, x) = \begin{cases} p_1 & \text{If } x \in \mathcal{X}_1 \\ p_2 = 0.85 & \text{If } x \in \mathcal{X}_2 \end{cases}, \quad (30)$$

where

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

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

We implemented the finite-dimensional MICP (24) using CVXPY [42] and Gurobi [43]. In order to limit the computational burden, we leveraged the strategy discussed in Remark 2 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 <https://github.com/urossolia/mixed-observable-LQR>.

We tested the proposed strategy for two different scenarios. In the first scenario, we set the probability p_1 of the observation model (30) equal to 0.85, and in the second one we set $p_1 = 0.7$. In both cases, we considered an initial condition $x(0) = [0, 0, 0, 0]^\top$, a prediction horizon $N = 60$ and we assumed that an observation is collected every $N_b = 30$ time steps. Notice that in the first scenario 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 receiving an observation at time $t = N_b$ commits to a goal location, as shown in Figure 4. 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 trajectories that moves backward and visits region \mathcal{X}_2 to collect an observations that is correct with probability $p_2 = 0.85$, before committing to steer the system towards a goal location, as shown in Figure 5.

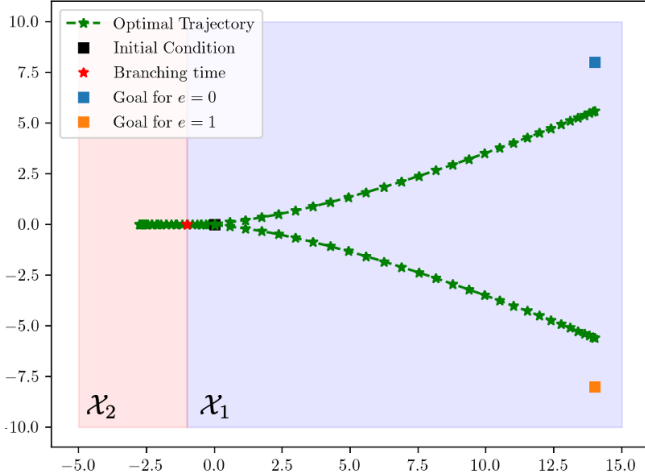


Fig. 5. Optimal trajectory computed solving the MICP for $N = 60$ and assuming that an observation is collected every $N_b = 30$ time steps, as discussed in Section III-C. In this scenario $p_1 = 0.5$, therefore the controller steers the system backwards to reach regions \mathcal{X}_2 where the observation is correct with probability 0.85, before committing to a goal location.

Table I shows 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 . Furthermore, in the MICP the piecewise observation model is implemented using $l = \sum_{k=0}^{P-1} |\mathcal{O}|^k$ integer variables which render the MICP computationally expensive. For more details please refer to the Appendix VII-B.

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

Finally, we compare the optimal trajectories for $N = 60$, $N_b \in \{30, 12\}$, and $p_1 = 0.7$. Figures 5–6 show the optimal trajectories for $N_b = 30$ and $N_b = 12$, respectively. Notice that both trajectories are computed for a horizon of $N = 60$ time steps and the constant $N_b \in \{12, 30\}$ determines how often an observation is collected, and therefore the number of tree branches. When $N_b = 30$, we have that the optimal trajectory branches only at time $t = N_b = 30$ and only one observation is collected. On the other hand, the trajectory from Figure 6 is computed by setting $N_b = 12$, and therefore the trajectory branches at each time $t \in \{12, 24, 36, 48\}$, and four observations are collected. It is interesting to notice that the optimizer plans a trajectory that steers the system to region \mathcal{X}_2 to collect three observations, which are correct with probability

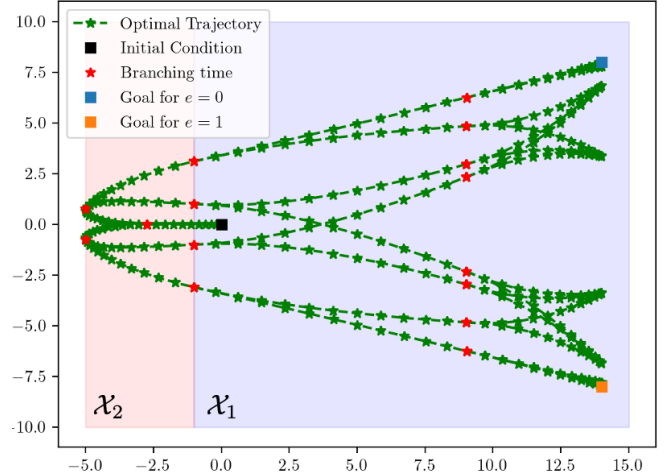


Fig. 6. Optimal trajectory computed solving the MICP for $N = 60$ and assuming that an observation is collected every $N_b = 12$ time steps, as discussed in Section III-C. In this scenario $p_1 = 0.5$ and $P = 5$, therefore compared to Figure 5 the optimal tree of trajectories is composed by more branches.

$p_2 = 0.85$, before driving the system towards one of the two goals. Furthermore, the optimal cost associated with the trajectory from Figure 6 is lower than the one associated with the trajectory from Figure 5, as shown in Table I. This result is expected as the approach presented in Section III-C better approximates the original problem when more observations are collected along the planning horizon. Indeed, as we discussed in Theorem 1 the value function associated problem (11), where an observation is collected every N_b time steps, is an upper bound to the value function of the original problem (3).

B. Partially Observable Navigation Problem

We tested the proposed strategy on the navigation task shown in Figures 7–8. In this example there are two obstacle (black regions) and the objective is to regulate the system to a goal location that can only be inferred through partial observations. The observation model is piecewise and it is defined as follows:

$$Z(o = e, e, x) = \mathbb{P}(o = e | e, x) = \begin{cases} p_1 = 0.5 & \text{If } x \in \mathcal{X}_1 \\ p_2 = 0.7 & \text{If } x \in \mathcal{X}_2 \\ p_3 = 0.85 & \text{If } x \in \mathcal{X}_3 \\ p_4 = 0.85 & \text{If } x \in \mathcal{X}_4 \end{cases}, \quad (31)$$

where regions \mathcal{X}_1 , \mathcal{X}_2 , \mathcal{X}_3 , and \mathcal{X}_4 are depicted in Figures 7–8. Less formally, 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, the system evolution follows the discrete time dynamics (29) and the cost function is defined by the following matrices:

$$Q = 10^{-4}I_n, R = 10^{-2}I_d, \text{ and } Q_N = 10I_n,$$

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

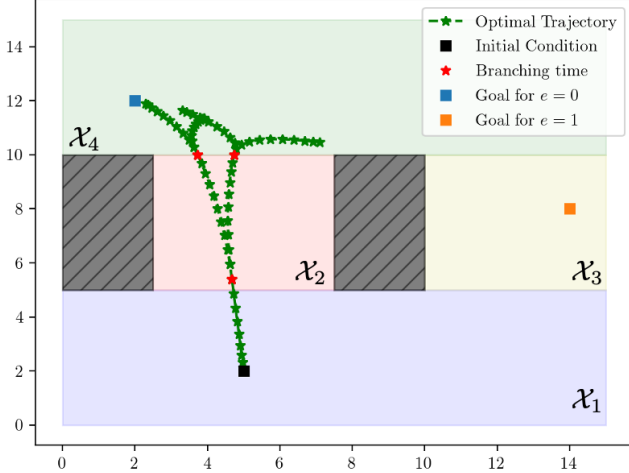


Fig. 7. Optimal trajectory computed solving the MICQ for $N = 30$ and $N_b = 10$. The objective is to steer the system to the goal location that is a function of the partially observable state $e \in \{0, 1\}$, while avoiding the two obstacles (black rectangles). In this scenario, the initial belief $b_0 = [0.8, 0.2]^\top$ and the observation model is piecewise over the regions \mathcal{X}_1 , \mathcal{X}_2 , \mathcal{X}_2 , and \mathcal{X}_4 .

We implemented the MICP using CVXPY [42] and Gurobi [43]. Notice that the feasible regions is non-convex 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 in \mathcal{X}_1 , \mathcal{X}_2 , \mathcal{X}_3 , or \mathcal{X}_4 . For implementation details please refer the Appendix VII-B and/or the source code available at <https://github.com/urossolia/mixed-observable-LQR>.

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 optimal trajectory computed solving the MICP branches at time $t = 12$ and time $t = 24$. Figure 7 shows the optimal trajectory tree when the initial belief $b_0 = [0.9, 0.1]^\top$. Although the observation model (31) is less accurate in region \mathcal{X}_2 compared to regions \mathcal{X}_3 and \mathcal{X}_3 , the optimizer plans a tree of trajectories to collect the first observation is region \mathcal{X}_2 . This plan results in a direct path steering the system towards the top left corner where the goal is located with high probability, as suggested by the initial belief $b_0 = [0.9, 0.1]^\top$. On the other hand, when the initial belief $b_0 = [0.5, 0.5]^\top$, the optimizer plans a trajectory that collects observations only in regions \mathcal{X}_3 and \mathcal{X}_4 , as shown in Figure 8. This result is expected as when we do not have any prior on the location of the goal, an optimal strategy should maximize the number of informative observations that are collected in regions \mathcal{X}_3 and \mathcal{X}_4 .

VI. CONCLUSIONS

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 state discrete, the control problem can be reformulated as

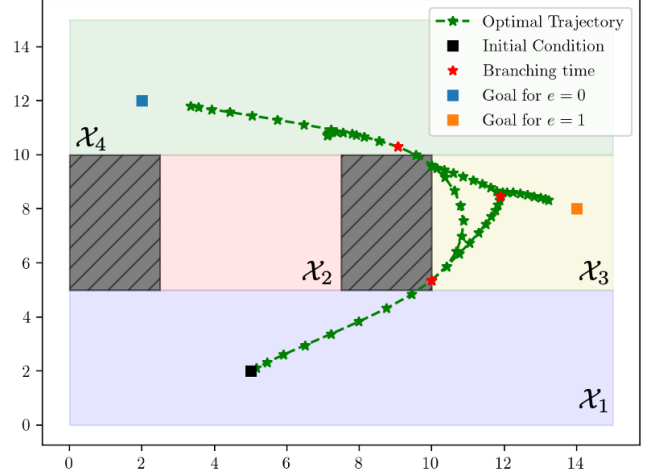


Fig. 8. Optimal trajectory computed solving the MICQ for $N = 30$ and $N_b = 10$. The objective is to steer the system to the goal location that is a function of the partially observable state $e \in \{0, 1\}$, while avoiding the two obstacles (black rectangles). In this scenario, the initial belief $b_0 = [0.5, 0.5]^\top$ and the observation model is piecewise over the regions \mathcal{X}_1 , \mathcal{X}_2 , \mathcal{X}_2 , and \mathcal{X}_4 .

a finite-dimensional deterministic optimization problem over a trajectory tree. Leveraging this result, we shown that under mild assumptions the control problem can be recast as a MICP though a nonlinear change of coordinates. This result suggests that in the new coordinate space the control problem is easier to solve. In future work, we wish to investigate the effectiveness of approximate dynamics programming and reinforcement learning strategies that would benefit from the properties of the value function in the new coordinate space.

VII. APPENDIX

A. Belief Derivation

The belief update equation (4) is derived using the total law of probability and Bayes's rule. In particular, we have that

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

For more details about the above derivation please refer to [37, Chapter 3].

B. MICP Reformulation

In this section, we show how to rewrite problem (24) as a MICP using the strategy from [40]. We assume that the disjoint polytopic regions \mathcal{X}_i from Assumption 2 are boxes¹, i.e.,

$$\mathcal{X}_i = \{x \in \mathbb{R}^n : I_{2n}x \leq f_i\},$$

where the matrix

$$I_{2n} = [\text{diag}([1, \dots, 1]), -\text{diag}([1, \dots, 1])]^\top \in \mathbb{R}^{n \times n}$$

and the vector $f_i \in \mathbb{R}^{2n}$. We notice that, as the regions \mathcal{X}_i are disjoint for all $i \in \{1, \dots, R\}$, the constraints

$$\delta_{k,i}^{\text{ok}} = \mathbb{1}_{\mathcal{X}_i}(x_k^{\text{ok}}), \forall i \in \{1, \dots, R\}, \quad (32)$$

can be rewritten as

$$I_{2n}x_k^{\text{ok}} \leq \sum_{i=1}^R f_i \delta_{k,i}^{\text{ok}} \text{ and } \sum_{i=1}^R \delta_{k,i}^{\text{ok}} = 1.$$

Furthermore, we have that the following constraint

$$z_{k+1}^{\text{ok}+1} = \sum_{i=1}^R D_i(o_{k+1}) z_k^{\text{ok}} \delta_{k,i}^{\text{ok}}$$

can be rewritten as

$$\begin{aligned} z_{k+1}^{\text{ok}+1} &= \sum_{i=1}^R D_i(o_{k+1}) y_{k,i}^{\text{ok}+1}, \\ y_{k,i}^{\text{ok}+1} &\leq z_k^{\text{max}} \delta_{k,i}^{\text{ok}}, \\ y_{k,i}^{\text{ok}+1} &\geq z_k^{\text{min}} \delta_{k,i}^{\text{ok}}, \\ y_{k,i}^{\text{ok}+1} &\leq z_k^{\text{ok}} - m(1 - \delta_{k,i}^{\text{ok}}) \\ y_{k,i}^{\text{ok}+1} &\geq z_k^{\text{ok}} - M(1 - \delta_{k,i}^{\text{ok}}) \end{aligned} \quad (33)$$

where the auxiliary variables $y_{k,i}^{\text{ok}+1} \in \mathbb{R}^n$ for all $k \in \{0, \dots, N-1\}$ and $i \in \{1, \dots, R\}$. In the above expression, the constants z_k^{max} from (28) and $z_k^{\text{min}} = 0$ are upper and lower bounds to the inverse belief vector z_k^{ok} , respectively. For more details about the above derivation please refer to [40]. Finally, leveraging the above reformulations we have that problem (24) can be rewritten as

$$\begin{aligned} V(x(t), z(t)) &= \min_{\mathbf{u}, \delta} \sum_{k=0}^{N-1} \sum_{\mathbf{o}_k \in \mathcal{O}^k} \sum_{e \in \mathcal{E}} \frac{h(x_k^{\text{ok}}, u_k^{\text{ok}}, e)}{z_k^{\text{ok}}[e]} \\ &\quad + \sum_{\mathbf{o}_N \in \mathcal{O}^N} \sum_{e \in \mathcal{E}} \frac{h_N(x_N^{\text{on}}, e)}{z_N^{\text{on}}[e]} \\ \text{s.t. } &x_{k+1}^{\text{ok}} = Ax_k^{\text{ok}+1} + Bu_k^{\text{ok}}, \\ &x_0^{\text{ok}+1} = x(t), z_0^{\text{ok}+1} = z(t), \\ &u_k^{\text{ok}} \in \mathcal{U}, x_k^{\text{ok}} \in \mathcal{X}, \\ &(32), (33), \\ &\forall k \in \{0, \dots, N-1\}, \end{aligned}$$

The above problem is a MICP than can be solved with a branch-and-bound algorithm. In our examples, we used CVXPY [42] and Gurobi [43] to solve the MICP. The code can be found online at <https://github.com/urossolia/mixed-observable-LQR>.

¹This strategy can be used also when the regions \mathcal{X}_i are polytope. We presented this spacial case to streamline the presentation. For more details on the general case please refer to [40].

REFERENCES

- [1] C. E. Garcia, D. M. Prett, and M. Morari, "Model predictive control: Theory and practice—a survey," *Automatica*, vol. 25, no. 3, pp. 335–348, 1989.
- [2] M. Morari and J. H. Lee, "Model predictive control: past, present and future," *Computers & Chemical Engineering*, vol. 23, no. 4-5, pp. 667–682, 1999.
- [3] F. Allgower, R. Findeisen, Z. K. Nagy *et al.*, "Nonlinear model predictive control: From theory to application," *Journal-Chinese Institute Of Chemical Engineers*, vol. 35, no. 3, pp. 299–316, 2004.
- [4] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, "Survey constrained model predictive control: Stability and optimality," *Automatica (Journal of IFAC)*, vol. 36, no. 6, pp. 789–814, 2000.
- [5] F. Borrelli, A. Bemporad, and M. Morari, *Predictive control for linear and hybrid systems*. Cambridge University Press, 2017.
- [6] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2016.
- [7] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *2019 18th European Control Conference (ECC)*. IEEE, 2019, pp. 3420–3431.
- [8] Y. Wang and S. Boyd, "Fast model predictive control using online optimization," *IEEE Transactions on control systems technology*, vol. 18, no. 2, pp. 267–278, 2009.
- [9] R. Lopez-Negrete, F. J. D'Amato, L. T. Biegler, and A. Kumar, "Fast nonlinear model predictive control: Formulation and industrial process applications," *Computers & Chemical Engineering*, vol. 51, pp. 55–64, 2013.
- [10] 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 Transactions on Automatic Control*, vol. 59, no. 12, pp. 3238–3251, 2014.
- [11] D. Kouzoupis, G. Frison, A. Zanelli, and M. Diehl, "Recent advances in quadratic programming algorithms for nonlinear model predictive control," *Vietnam Journal of Mathematics*, vol. 46, no. 4, pp. 863–882, 2018.
- [12] 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.
- [13] S. Gros, M. Zanon, R. Quirynen, A. Bemporad, and M. Diehl, "From linear to nonlinear mpc: bridging the gap via the real-time iteration," *International Journal of Control*, vol. 93, no. 1, pp. 62–80, 2020.
- [14] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat, "Predictive active steering control for autonomous vehicle systems," *IEEE Transactions on control systems technology*, vol. 15, no. 3, pp. 566–580, 2007.
- [15] D. Hrovat, S. Di Cairano, H. E. Tseng, and I. V. Kolmanovsky, "The development of model predictive control in automotive industry: A survey," in *2012 IEEE International Conference on Control Applications*. IEEE, 2012, pp. 295–302.
- [16] P. F. Lima, G. C. Pereira, J. Mårtensson, and B. Wahlberg, "Experimental validation of model predictive control stability for autonomous driving," *Control Engineering Practice*, vol. 81, pp. 244–255, 2018.
- [17] 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 *Second international conference on building energy and environment*, 2012, pp. 979–986.
- [18] 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, p. 631, 2018.
- [19] E. T. Maddalena, Y. Lian, and C. N. Jones, "Data-driven methods for building control—a review and promising future directions," *Control Engineering Practice*, vol. 95, p. 104211, 2020.
- [20] P. O. Scokaert and D. Q. Mayne, "Min-max feedback model predictive control for constrained linear systems," *IEEE Transactions on Automatic control*, vol. 43, no. 8, pp. 1136–1142, 1998.
- [21] 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.
- [22] 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.
- [23] 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.

-
- [24] S. Yu, C. Maier, H. Chen, and F. Allgöwer, "Tube mpc scheme based on robust control invariant set with application to lipschitz nonlinear systems," *Systems & Control Letters*, vol. 62, no. 2, pp. 194–200, 2013.
- [25] J. Fleming, B. Kouvaritakis, and M. Cannon, "Robust tube mpc for linear systems with multiplicative uncertainty," *IEEE Transactions on Automatic Control*, vol. 60, no. 4, pp. 1087–1092, 2014.
- [26] Y.-S. Wang, N. Matni, and J. C. Doyle, "A system-level approach to controller synthesis," *IEEE Transactions on Automatic Control*, vol. 64, no. 10, pp. 4079–4093, 2019.
- [27] A. Liniger, X. Zhang, P. Aeschbach, A. Georghiou, and J. Lygeros, "Racing miniature cars: Enhancing performance using stochastic mpc and disturbance feedback," in *2017 American Control Conference (ACC)*. IEEE, 2017, pp. 5642–5647.
- [28] A. Ben-Tal, A. Goryashko, E. Guslitzer, and A. Nemirovski, "Adjustable robust solutions of uncertain linear programs," *Mathematical programming*, vol. 99, no. 2, pp. 351–376, 2004.
- [29] 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.
- [30] I. Alvarado, D. Limón, T. Alamo, and E. F. Camacho, "Output feedback robust tube based mpc for tracking of piece-wise constant references," in *2007 46th IEEE Conference on Decision and Control*. IEEE, 2007, pp. 2175–2180.
- [31] 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.
- [32] L. Svensson, L. Masson, N. Mohan, E. Ward, A. P. Brenden, L. Feng, and M. Törngren, "Safe stop trajectory planning for highly automated vehicles: An optimal control problem formulation," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 517–522.
- [33] J. P. Alsterda and J. C. Gerdes, "Contingency model predictive control for linear time-varying systems," *arXiv preprint arXiv:2102.12045*, 2021.
- [34] J. P. Alsterda, M. Brown, and J. C. Gerdes, "Contingency model predictive control for automated vehicles," in *2019 American Control Conference (ACC)*. IEEE, 2019, pp. 717–722.
- [35] P. Sopasakis, D. Hecceg, A. Bemporad, and P. Patrinos, "Risk-averse model predictive control," *Automatica*, vol. 100, pp. 281–288, 2019.
- [36] I. Batkovic, U. Rosolia, M. Zanon, and P. Falcone, "A robust scenario mpc approach for uncertain multi-modal obstacles," *IEEE Control Systems Letters*, vol. 5, no. 3, pp. 947–952, 2020.
- [37] V. Krishnamurthy, *Partially observed Markov decision processes*. Cambridge university press, 2016.
- [38] S. C. Ong, S. W. Png, D. Hsu, and W. S. Lee, "Pomdps for robotic tasks with mixed observability," in *Robotics: Science and Systems*, vol. 5, 2009, p. 4.
- [39] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.
- [40] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407–427, 1999.
- [41] A. Bemporad and C. Filippi, "An algorithm for approximate multiparametric convex programming," *Computational optimization and applications*, vol. 35, no. 1, pp. 87–108, 2006.
- [42] S. Diamond and S. Boyd, "CVXPY: A Python-embedded modeling language for convex optimization," *Journal of Machine Learning Research*, vol. 17, no. 83, pp. 1–5, 2016.
- [43] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2021. [Online]. Available: <https://www.gurobi.com>