IEEE Control Systems Letters paper presented at
2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Model Predictive Control in Partially Observable Multi-Modal Discrete Environments

Ugo Rosolia, Dario C. Guastella, Giovanni Muscato, Francesco Borrelli

***Abstract*— Autonomous systems operate in environments that can be observed only through noisy measurements. Thus, controllers should compute actions based on their beliefs about the surroundings. In these settings, we design a Model Predictive Controller (MPC) based on a *continuous-state* Linear Time-Invariant (LTI) system model operating in a *discrete-state* environment described by a Hidden Markov Model (HMM). Environment constraints are modeled as chance constraints and environment observations can be asynchronous with system state measurements and controller updates. We show how to approximate the solution of the MPC problem defined over the space of feedback policies by optimizing over a trajectory tree, where each branch is associated with an environment measurement. The proposed approach guarantees chance constraint satisfaction and recursive feasibility. Finally, we test the proposed strategy on navigation examples in partially observable environments, where the proposed MPC guarantees chance constraint satisfaction.**

I. Introduction

Autonomous systems operating in uncertain environments make decisions based on noisy measurements. When uncertainties are uni-modal, the decision-making process is usually divided into two steps. First, noisy measurements are leveraged to estimate the system state. Then, the controller is designed assuming perfect state feedback [1]. This separation strategy is optimal for stabilizing linear unconstrained systems affected by additive Gaussian disturbances and noises [2]. For systems subject to state and input constraints, the estimation and control problems can be separated, but it is necessary to compute error bounds associated with the state estimate. These bounds should then be leveraged in a robust control design to guarantee constraint satisfaction [3].

When uncertainties are multi-modal, the optimal control policy minimizing the expected cost can be computed by modeling the problem using a Partially Observable Markov Decision Process (POMDP), which is a decision-making formalism to jointly model estimation and control problems. Unfortunately, solving POMDPs is computationally intractable, even for systems with discrete state and action spaces [4]. For this reason, several strategies have been proposed in the literature to approximate the solution to POMDPs [5]–[9].

We focus on Mixed-Observable Markov Decision Processes (MOMDPs) where perfect state feedback is not available only for a subset of the state space [10]. In particular, we focus on control problems where perfect state feedback is available for the system state, whereas only partial noisy discrete measurements are available to estimate the environment mode. These settings are common in several practical applications such as autonomous driving and robot navigation, where it is often possible to compute a reasonably accurate estimate for the vehicle state, but it is hard to estimate the state of the surroundings which is multi-modal. For instance, in autonomous driving the environment state could encode the intentions of other vehicles or pedestrians, e.g., the intent of drivers to perform lane change or lane-keeping maneuvers [11]–[13].

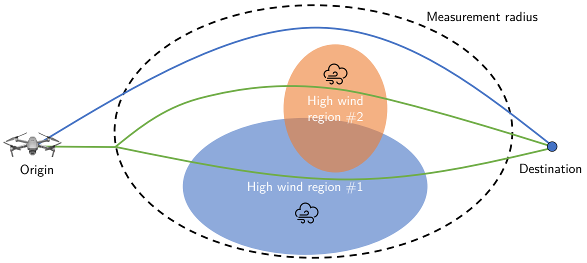


Fig. 1: Navigation task where a drone has to plan a route without knowing the exact location of the windy region, which will be inferred during navigation via noisy measurements.

Several strategies have been proposed for the control design of autonomous systems operating in partially observable environments [11]–[19]. These approaches leverage a Model Predictive Controller (MPC) which solves an optimization problem over a trajectory tree. Each branch of the tree is associated with either a sensor measurement, a disturbance realization, or an environment mode; thus such a trajectory tree encodes a policy.[1] The resulting MPC policy computes actions to influence the environment or to gather sensor measurements that can be used for inference [13]–[16].

In this work, we model the environment evolution and the sensor accuracy using a Hidden Markov Model (HMM). Then, we design an MPC policy that optimizes a trajectory tree constructed based on the environment's HMM and the current belief. Our contribution is twofold. First, we show how to construct a trajectory tree that guarantees chance constraint satisfaction. Compared to previous works, we update the constraint enforced at each branch of the tree based on the environment belief and the imposed chance constraints. In particular, we design Algorithm 1 to compute a set of constraints that guarantee chance constraint satisfaction. As

The work of Dario C. Guastella was supported by the Project PON R&I REACT-EU. The work by Ugo Rosolia is done outside Amazon. Ugo Rosolia is with Amazon.com, Inc. Dario C. Guastella and Giovanni Muscato are with the University of Catania, Italy (e-mail: dario.guastella@unict.it). Francesco Borrelli is with the University of California at Berkeley, USA.

[1] In [20], it was shown that for linear systems subject to state and input constraints optimizing over a trajectory tree is equivalent to optimizing over the space of feedback policies, when the objective is to minimize the worst-case cost and perfect state feedback is available.

shown in the results section, the proposed strategy guarantees chance constraint satisfaction, while standard scenario MPC approaches fail. Second, we show that our MPC design guarantees recursive feasibility. To guarantee recursive feasibility in the case of asynchronous observations and chance constraints, we design an MPC problem where the optimization is defined for a trajectory tree, where each branch is associated with an observation sequence and a different set of constraints that are time-varying. Finally, we test our strategy on a navigation example, where the environment state is unknown to the controller. We show that our MPC guarantees chance constraint satisfaction and recursive feasibility, even when only noisy environment measurements are available.

Notation: We denote the ith element of a vector $v \in \mathbb{R}^n$ as $v[i]$. For a function $Z : \mathbb{R}^n \to \mathbb{R}$ and a vector $v \in \mathbb{R}^n$, we indicate $Z(v)$ as the value of the function Z at the point v. Furthermore, for a vector v, we define the function $\texttt{Sort}(v)$ sorting the elements of v in descending order and the function $\texttt{ArgSort}(v)$ returning the indices of the vector v that would sort the vector, i.e., $v[\texttt{ArgSort}(v)] = \texttt{Sort}(v)$. Given a set $\mathcal{S} \subset \mathbb{R}^n$, we define its complement as $\mathcal{S}^c = \mathbb{R}^n \setminus \mathcal{S}$ and its cardinality as $|\mathcal{S}|$. The set of positive integers is denoted as $\mathbb{Z}_{0+} = \{0, 1, 2, \ldots\}$, and the set of positive reals as $\mathbb{R}_{0+} = [0, \infty)$. Finally, we use the symbol $\emptyset$ to denote the empty set.

II. Problem Formulation

A. System and Environment Models

We consider the following linear time-invariant system:

$$x_{t+1} = Ax_t + Bu_t, \tag{1}$$

where $x_t \in \mathbb{R}^n$ and $u_t \in \mathbb{R}^m$ denote the state and the control input at time t. The system operates in an environment represented by partially observable discrete states. We model the environment evolution as a hidden Markov model (HMM) given by the tuple $\mathcal{H} = (\mathcal{E}, \mathcal{O}, T, Z)$, where:

- $\mathcal{E} = \{1, \ldots, |\mathcal{E}|\}$ is a set of partially observable environment states;
- $\mathcal{O} = \{1, \ldots, |\mathcal{O}|\}$ is the set of observations.
- The function $T : \mathcal{E} \times \mathcal{E} \to [0, 1]$ describes the probability of transitioning to a state e' given the current environment state e, i.e., $T(e', e) = \mathbb{P}(e'|e)$.
- The function $Z : \mathcal{E} \times \mathcal{O} \times \mathbb{Z}_{0+} \to [0, 1]$ describes the probability of observing o at time step t, given the environment state e, i.e., $Z(e, o, t) = \mathbb{P}(o|e, t)$.

As the environment state e_t is partially observable, it is common practice to introduce the following *belief vector* [21]:

$$b_t \in \mathcal{B} = \{b \in \mathbb{R}^{|\mathcal{E}|}_{0+} : \textstyle\sum_{e=1}^{|\mathcal{E}|} b[e] = 1\}, \tag{2}$$

where each element $b_t[e]$ represents the posterior probability that the state of the environment e_t equals $e \in \mathcal{E}$, given the observation vector $\mathbf{o_t} \in \mathcal{O}^k$ collecting k observations stored up to time t, the system trajectory $\mathbf{x_t} \in \mathbb{R}^{n \times (t+1)}$, and the belief vector b_0 at time $t = 0$, i.e., $b_t[e] = \mathbb{P}(e|\mathbf{o_t}, \mathbf{x_t}, b_0)$. We recall that the belief vector is a *sufficient statistic* and it can be recursively updated by using the Bayes rule [21].

System (1) is subject to the input and state constraints:

$$u_t \in \mathcal{U} \text{ and } \mathbb{P}\big(x_t \in \mathcal{X}(e_t)|b_t\big) \geq 1 - \epsilon, \; \forall t \in \{0, 1, \ldots\}. \tag{3}$$

Notice that at each time t the constraint set $\mathcal{X}(e_t)$ is a function of the partially observable environment state e_t that is not known at execution. For this reason, the above chance constraint is conditioned on the environment belief b_t.

B. Control Objectives

Our goal is to design a control policy π mapping the state $x_t \in \mathbb{R}^n$ and the environment belief $b_t \in \mathcal{B}$ to a control action $u_t \in \mathbb{R}^m$, i.e.,

$$\pi^{\text{MPC}} : \mathbb{R}^n \times \mathcal{B} \to \mathbb{R}^m. \tag{4}$$

The above policy (4) in closed-loop with system (1) should guarantee that input and state constraints (3) are satisfied. Throughout the paper we make the following assumptions.

Assumption 1. The input and state constraint set $\mathcal{U}$ and $\mathcal{X}(e)$ are compact sets containing the origin for all $e \in \mathcal{E}$.

Assumption 2. During the control task, we collect K environment observations. Furthermore, we know the time steps $t_1, \ldots, t_K$ at which these K observations are collected. Thus, we introduce the following set collecting these time instances:

$$\mathcal{T}_{\text{obs}} = \{t_1, \ldots, t_K\}. \tag{5}$$

Our problem is motivated by the navigation task shown in Figure 1, where a drone has to fly from an origin to a destination while avoiding high windy areas. In this example, the system state x_t represents the position of the drone, while the environment state e_t represents the location of the windy area. Such a location is unknown, but we know that it may be either in region #1 (blue ellipse) or region #2 (red ellipse). In this example, a robust plan (blue trajectory) would simply avoid the possible windy areas. On the other hand, a policy based on observations about the wind location would first fly the drone toward the windy regions and then adjust its trajectory based on measurements (green tree of trajectories).

III. Control Design

A. Belief update

In this section, we present the belief update equation. The belief vector from (2) is a sufficient statistic for an HMM and it can be recursively computed based on observations [21]. As discussed in Assumption 2, the time instances at which observations are collected are known and stored in the set $\mathcal{T}_{\textbf{obs}}$. Given such a set of time instances, we write the belief update as follows [21]:

$$b_t = f_b(b_{t-1}, o_t, t) = \begin{cases} \Theta(o_t, t)\Omega b_{t-1}/\eta_t & \text{If } t \in \mathcal{T}_{\text{obs}} \\ \Omega b_{t-1} & \text{otherwise} \end{cases} \tag{6}$$

where $\eta_t = \mathbb{P}(o_t|b_{t-1})$,

$$\Omega = \begin{bmatrix} T(1,1) & \ldots & T(1,|\mathcal{E}|) \\ T(2,1) & \ldots & T(2,|\mathcal{E}|) \\ \vdots & & \vdots \\ T(|\mathcal{E}|,1) & \ldots & T(|\mathcal{E}|,|\mathcal{E}|) \end{bmatrix} \tag{7}$$

and

$$\Theta(o_t,t) = \text{diag}\Big(\begin{bmatrix} Z(1,o_t,t) & \dots & Z(|\mathcal{E}|,o_t,t) \end{bmatrix} \Big). \quad (8)$$

B. The MPC optimization problem

Given the system state x_t and the environment belief b_t at time t, we introduce the following MPC optimization problem:

$$\begin{aligned} J_\infty(x_t,b_t) = \min_{\boldsymbol{\pi}_t} \quad & \mathbb{E}\left[\sum_{k=t}^{t+N-1} h(x_{k|t},u_{k|t}) + V(x_{t+N|t}) \Big| b_t \right] \\ \text{subject to} \quad & x_{k+1|t} = Ax_{k|t} + Bu_{k|t}, \\ & b_{k+1|t} = f_b(b_{k|t}, o_{k+1|t}, k+1), \\ & x_{t|t} = x_t, b_{t|t} = b_t, \\ & u_{k|t} = \pi_{k|t}(x_{k|t}, b_{k|t}), \\ & u_{k|t} \in \mathcal{U}, \\ & \mathbb{P}\big(x_{k|t} \in \mathcal{X}(e_{k|t}) | b_{k|t}\big) \geq 1-\epsilon, \\ & x_{t+N|t} \in \mathcal{X}_F, \forall k \in \{t,\dots,t+N\}. \end{aligned} \quad (9)$$

where $\boldsymbol{\pi}_t = \{\pi_{t|t},\dots,\pi_{t+N-1|t}\}$. In the above problem, $h: \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ and $V: \mathbb{R}^n \rightarrow \mathbb{R}$ represent the stage cost and the terminal cost, respectively. Furthermore, the terminal constraint $\mathcal{X}_F$ satisfies the following assumption:

Assumption 3. The terminal constraint set $\mathcal{X}_F \subset \mathcal{X}(e)$ for all $e \in \mathcal{E}$ is a control invariant set, i.e., for all $x \in \mathcal{X}_F$ there exists a $u \in \mathcal{U}$ such that $Ax + Bu \in \mathcal{X}_F$.

In problem (9), the variable $x_{k|t}$ indicates the predicted state at time k for a prediction computed at time t. The same notation is used for the control action $u_{k|t}$, the belief vector $b_{k|t}$, the observation $o_{k|t}$, and the environment state $e_{k|t}$. Note that if $k \notin \mathcal{T}_{\texttt{obs}}$, we have that $o_{k|t} = \emptyset$. Thus at the predicted time k, the policy $\pi_{k|t}$ maps the predicted state $x_{k|t}$ and belief $b_{k|t}$ to the control action $u_{k|t}$.

Solving problem (9) is challenging for two reasons: (i) the optimization is defined over the space of feedback policies $\{\pi_{t|t},\dots,\pi_{t+N-1|t}\}$ that are continuous functions with uncountable degrees of freedom which render the optimization problem infinite-dimensional, and (ii) the system predicted states are subject to chance constraints. To overcome these challenges, in the next section we first rewrite the above problem as an optimization over a tree of trajectories. Then, we leverage this reformulation to approximate the chance constraints. The proposed reformulation builds upon our previous work [16] where we did not consider constraint sets that change as a function of the environment state.

C. Finite-dimensional reformulation

In this section, we reformulate the chance-constrained optimization problem (9) as a finite-dimensional problem. First, we introduce the observation vector $\mathbf{o_{t:t+N}}$ collecting the observations from time t to time $t+N$, i.e.,

$$\mathbf{o_{t:t+N}} = [o_{t_k},\dots,o_{t_j}], \quad (10)$$

where t_k and t_j are the time steps at which the kth and jth observations are collected. Without loss of generality, we assume that $t \leq t_k < \dots < t_j \leq t+N$. Let $M_{t:t+N}$ be the number of observations collected from time t to $t+N$, we have that there are $|\mathcal{O}|^{M_{t:t+N}}$ possible sequences of observations that we denote as:

$$\mathbf{o^i_{t:t+N}} = \{o^i_{t_k},\dots,o^i_{t_j}\} \text{ for all } i \in \{1,\dots,|\mathcal{O}|^{M_{t:t+N}}\}. \quad (11)$$

Leveraging the $S_{t:k} = |\mathcal{O}|^{M_{t:k}}$ sequence of observations (10), we define the finite-dimensional optimization problem:

$$J_f(x_t,b_t) =$$

$$\begin{aligned} \min_{\mathbf{u}_t} \sum_{k=t}^{t+N-1} \sum_{i=1}^{S_{t:k}} v^i_{k|t} h(x^i_{k|t},u^i_{k|t}) + \sum_{i=1}^{S_{t:t+N}} v^i_{t+N|t} V(x^i_{t+N|t}) \end{aligned}$$

$$\begin{aligned} \text{subject to } & x^i_{k+1|t} = Ax^i_{k|t} + Bu^i_{k|t}, && \text{(12a)} \\ & b^i_{k+1|t} = f_b(b^i_{k|t}, o^i_{k+1|t}, k+1), && \text{(12b)} \\ & v^i_{k+1|t} = f_v(v^i_{k|t}, o^i_{k+1|t}, k+1), && \text{(12c)} \\ & x^i_{t|t} = x_t, b^i_{t|t} = b_t, v^i_{t|t} = b_t, && \text{(12d)} \\ & u^i_{k|t} = u^j_{k|t}, \text{ if } \mathbf{o^i_{t:k}} = \mathbf{o^j_{t:k}}, && \text{(12e)} \\ & u^i_{k|t} \in \mathcal{U},\ x^i_{t+N|t} \in \mathcal{X}_F, && \text{(12f)} \\ & \mathbb{P}\big(x^i_{k|t} \in \mathcal{X}(e^i_{k|t}) | b^i_{k|t}\big) \geq 1-\epsilon, && \text{(12g)} \\ & \forall k \in \{t,\dots,t+N\}, \\ & \forall i \in \{1,\dots,S_{t:t+N}\}, \forall j \in \{1,\dots,S_{t:t+N}\}. \end{aligned}$$

where $\mathbf{u}_t = \{u^1_{t|t},\dots,u^{S_{t:t+N}}_{t+N-1|t}\}$ and $u^i_{k|t}$ is the predicted control action at time k for a prediction computed at time t and observation sequence i. Constraint (12e) enforces causality, i.e., if observation sequences $\mathbf{o^i_{t:t+N}}$ and $\mathbf{o^j_{t:t+N}}$ are indistinguishable up to time $\bar{k}$, then the control actions $u^i_{k|t}$ must be equal to $u^j_{k|t}$ for all $k \in \{t,\dots,\bar{k}\}$. As we collect $M_{t:k}$ observations from time t to time k, the causality constraint (12e) guarantees that at the predicted time k we have at most $|\mathcal{O}|^{M_{t:k}}$ different control actions, i.e., $u^i_{k|t} \neq u^j_{k|t}$ if and only if $\mathbf{o^i_{t:k}} \neq \mathbf{o^j_{t:k}}$. In problem (12), $v^i_{k|t}$ is the unnormalized belief vector and constraint (12c) represents the unnormalized belief vector update equation:

$$v_t = f_v(v_{t-1},o_t,t) = \begin{cases} \Theta(o_t,t)\Omega v_{t-1} & \text{If } t \in \mathcal{T}_{\text{obs}} \\ \Omega v_{t-1} & \text{otherwise} \end{cases}$$

where Θ and Ω are defined as in (7)–(8). The unnormalized belief vector is initialized using the belief b_t and it allows us to rewrite the expectation as a summation [16, Proposition 1].

Proposition 1. *For all $x \in \mathbb{R}^n$ and $b \in \mathcal{B}$, we have that $J_\infty(x,b) = J_f(x,b)$. Furthermore, let $\{\pi^*_{t|t},\dots,\pi^*_{t+N-1|t}\}$ and $\{u^{1,*}_{t|t},\dots,u^{S_{t:t+N},*}_{t+N-1|t}\}$ be the optimizer of problems* (9) *and* (12) *respectively, we have that $\pi^*_{t|t}(x_t,b_t) = u^{1,*}_{t|t}$.*

Proof: As the predicted belief $b_{k|t}$ is defined by the belief b_t and the $M_{t:k}$ observations collected from time t to time k, we have that the policy $\pi_{k|t}$ is evaluated at most $|\mathcal{O}|^{M_{t:k}}$ times. Thus, optimizing over the set of policies from (9) is equivalent to optimizing over the set of control actions from (12), each associated with an observation sequence $\mathbf{o^i_{t:k}}$ for $i \in \{1,\dots,|\mathcal{O}|^{M_{t:t+N}}\}$. Thus, as from [16, Proposition 1]

Algorithm 1: Chance Constraint Approximation

```
1  inputs: $b_t$, $t$, $\epsilon$;
2  Compute $b^i_{k|t}$ for all $k \in \{t, \dots, t+N-1\}$ and for all
     $i \in \{1, \dots, |\mathcal{O}|^{M_{t:t+N}}\}$;
3  for $k \in \{t, \dots, t+N-1\}$ do
4      for $i \in \{1, \dots, |\mathcal{O}|^{M_{t:t+N}}\}$ do
5          $\mathcal{C}^i_{k|t} = \emptyset$;
6          $b_{\texttt{sort}} = \texttt{Sort}(b^i_{k|t})$;
7          $e_{\texttt{sort}} = \texttt{ArgSort}(b^i_{k|t})$;
8          $p_{\texttt{env}} = 0$, $j = 0$;
9          while $p_{env} \leq 1 - \epsilon$ do
10             $p_{\texttt{env}} = p_{\texttt{env}} + b_{\texttt{sort}}[j]$;
11             $\mathcal{C}^i_{k|t}.\texttt{append}(e_{\texttt{sort}}[j])$;
12             $j = j + 1$;
13         end
14     end
15 end
16 Return: $\mathcal{C}^i_{k|t}$
```

we have that the expected cost from (9) is equivalent to the cost function in (12), we conclude that $J_\infty(x, b) = J_f(x, b)$ and $\pi^*_{t|t}(x_t, b_t) = u^{1,*}_{t|t}$ for all $x \in \mathbb{R}^n$ and $b \in \mathcal{B}$. ■

The key assumption leveraged by the proposed reformulation is that the HMM is defined for a set of discrete states. This allows us to reformulate the expectation as a summation and the policy as a finite set of control actions. Thus, no assumption on the cost function is required.

D. Chance constraint reformulation

We present the chance constraint approximation strategy. For each predicted time k and observation sequence i, we use Algorithm 1 to compute the set of environment states $\mathcal{C}^i_{k|t}$ such that $\mathbb{P}(e^i_{k|t} \in \mathcal{C}^i_{k|t} | b^i_{k|t}) \geq 1 - \epsilon$. Then, we leverage such a set to reformulate the chance constraint from problem (12).

In Algorithm 1, we first compute the predicted belief $b^i_{k|t}$. Then, we sort the belief vector (line 6) and compute the vector $e_{\texttt{sort}}$ collecting the environment states sorted in descending order by their belief (lines 7), i.e.,

$$\mathbb{P}(e_{\texttt{sort}}[j] = e^i_{k|t}) = b_{\texttt{sort}}[j], \forall j \in \mathcal{E}.$$

See Section I for further details on notation. In line 8, we initialize the scalar $p_{\texttt{env}}$ to keep track of the probability that $e^i_{k|t} \in \mathcal{C}^i_{k|t}$, i.e., $p_{\texttt{env}} = \mathbb{P}(e^i_{k|t} \in \mathcal{C}^i_{k|t})$. Finally, we append $e_{\texttt{sort}}[j]$ to the set $\mathcal{C}^i_{k|t}$, until the probability that $e^i_{k|t}$ belongs to $\mathcal{C}^i_{k|t}$ is greater than $1 - \epsilon$.

Given the sets $\mathcal{C}^i_{k|t}$ computed with Algorithm 1, we introduce the following finite time optimal control problem:

$$\begin{aligned} &\min_{\mathbf{u}_t} \sum_{k=t}^{t+N-1} \sum_{i=1}^{S_{t:k}} v^i_{k|t} h(x^i_{k|t}, u^i_{k|t}) + \sum_{i=1}^{S_{t:t+N}} v^i_{t+N|t} V(x^i_{t+N|t}), \\ &\text{subject to (12a) – (12f),} \\ &\qquad x^i_{k|t} \in \mathcal{X}(e), \forall e \in \mathcal{C}^i_{k|t}, \\ &\qquad \forall k \in \{t, \dots, t+N-1\}, \forall i \in \{1, \dots, S_{t:t+N}\}, \end{aligned} \tag{13}$$

Given the optimal solution from the above problem $\{u^{1,*}_{t|t}, \dots, u^{S_{t:t+N},*}_{t+N-1|t}\}$, we define the MPC policy as:

$$\pi^{\text{MPC}}(x_t) = u^{*,1}_{t|t}. \tag{14}$$

Next, we show that the above policy is recursively feasible and it guarantees that state and input constraints are satisfied.

IV. Properties

First, we show that the policy (14) returns a feasible control action at all times, if (13) is feasible at time $t = 0$.

Theorem 1. *Let Assumptions 1–3 hold. If problem* (13) *is feasible at time $t = 0$, then problem* (13) *is feasible at all time steps $t \in \{1, 2, \dots\}$. Furthermore, we have that $\pi^{MPC}(x_t) \in \mathcal{U}$ for all $t \in \{0, 1, \dots\}$.*

Proof: Assume that the belief and control sequences

$$\begin{aligned} &\{b^{1,*}_{t|t}, \dots, b^{|\mathcal{O}|^{M_{t:t+N}},*}_{t+N|t}\}, \\ &\{u^{1,*}_{t|t}, \dots, u^{|\mathcal{O}|^{M_{t:t+N}},*}_{t+N-1|t}\}, \end{aligned} \tag{15}$$

are the optimal solution from problem (13) at time t. For $j \in \{1, \dots, |\mathcal{O}|^{M_{t+1:k}}\}$, we define the observation vector $\bar{\mathbf{o}}^{\mathbf{j}}_{\mathbf{t:k}}$ collecting the jth sequence of observations from time $t+1$ to k and the observation o_t measured at time t if $t \in \mathcal{T}_{\texttt{obs}}$, i.e.,

$$\bar{\mathbf{o}}^{\mathbf{j}}_{\mathbf{t:k}} = \begin{cases} [o_t, \mathbf{o}^{\mathbf{j}}_{\mathbf{t+1:k}}] & \text{If } t \in \mathcal{T}_{\texttt{obs}} \\ \mathbf{o}^{\mathbf{j}}_{\mathbf{t+1:k}} & \text{Otherwise.} \end{cases} \tag{16}$$

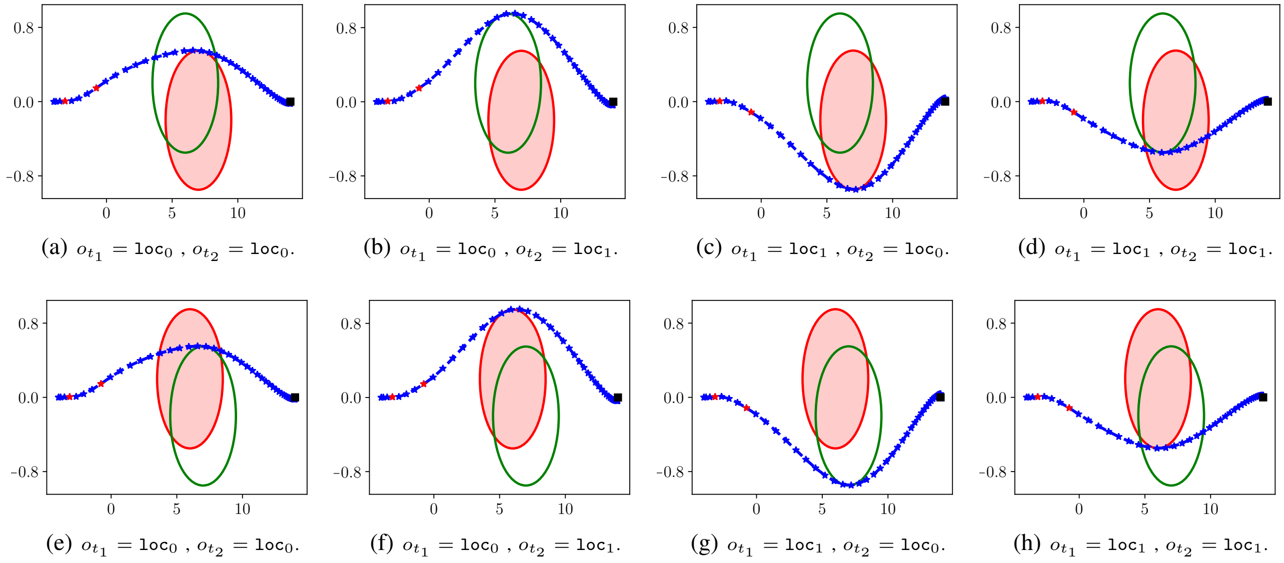
Let $\mathcal{T} = \{t+1, \dots, t+N-1\}$ and $t_f = t+N-1$, we leverage (15) and (16) to define the following candidate solution:

$$\begin{aligned} \bar{b}^j_{k|t+1} &= \begin{cases} b^{i,*}_{k|t} & \text{If } \mathbf{o}^{\mathbf{i}}_{\mathbf{t:k}} = \bar{\mathbf{o}}^{\mathbf{j}}_{\mathbf{t:k}} \text{ and } k \in \mathcal{T}, \\ f_b(b^{i,*}_{t_f|t}, o^{i,*}_{t_f|t}, t_f) & \text{If } \mathbf{o}^{\mathbf{i}}_{\mathbf{t:t+N-1}} = \bar{\mathbf{o}}^{\mathbf{j}}_{\mathbf{t:t+N-1}} \end{cases} \\ \bar{u}^j_{k|t+1} &= \begin{cases} u^{i,*}_{k|t} & \text{If } \mathbf{o}^{\mathbf{i}}_{\mathbf{t:k}} = \bar{\mathbf{o}}^{\mathbf{j}}_{\mathbf{t:k}} \text{ and } k \in \mathcal{T}, \\ \bar{u}^i & \text{If } \mathbf{o}^{\mathbf{i}}_{\mathbf{t:t+N-1}} = \bar{\mathbf{o}}^{\mathbf{j}}_{\mathbf{t:t+N-1}} \end{cases} \end{aligned} \tag{17}$$

where $\bar{u}^i$ satisfies $Ax^{i,*}_{t+N|t} + B\bar{u}^i \in \mathcal{X}_F$. Note that the existence of such a control action is guaranteed from Assumption 3. Furthermore, from definition (16) we have that $\bar{b}^i_{t+1|t+1} = b_{t+1}$ for all $i \in \{1, \dots, |\mathcal{O}|^{M_{t+1:k}}\}$. From this fact we have that $\bar{b}^{i,*}_{k|t+1}$ is feasible for all $k \in \{t, \dots, t+N\}$ and $i \in \{1, \dots, S_{t+1:t+1+N}\}$. Furthermore, by equation (17) we have $\bar{b}^j_{k|t+1} = b^{i,*}_{k|t}$ for all $k \in \mathcal{T}$, which in turn implies that $\mathcal{C}^j_{k|t+1} = \mathcal{C}^i_{k|t}$. Thus, the tree of trajectories associated with the predicted candidate input from (17) satisfies state and input constraints. Finally, from Assumption 3, we have that $\mathcal{X}_F \subset \mathcal{X}(e)$ for all $e \in \mathcal{E}$, which implies that for all $e \in \mathcal{E}$, $x \in \mathcal{X}_F$ and $b \in \mathcal{B}$, we have that $\mathbb{P}(x \in \mathcal{X}(e) | b) = 1$. This fact, together with the feasibility of the optimal solution (15), implies that (17) is a feasible solution for problem (13) at time $t+1$. Thus, $\pi^{\text{MPC}}(x_t) = u^{*,1}_{t|t} \in \mathcal{U}$ at all times. ■

In Proposition 1, we showed that the infinite-dimensional problem (9) is equivalent to the finite-dimensional chance constraint problem (12), which is still challenging to solve. Next, we show that the optimal solution from problem (13) is feasible for problem (12), i.e., the chance constraint problem (12) can be approximated by solving (13).

Proposition 2. *An optimal solution to problem* (13) *is a feasible solution for problem* (12).



(a) $o_{t_1} = \texttt{loc}_0$, $o_{t_2} = \texttt{loc}_0$.
(b) $o_{t_1} = \texttt{loc}_0$, $o_{t_2} = \texttt{loc}_1$.
(c) $o_{t_1} = \texttt{loc}_1$, $o_{t_2} = \texttt{loc}_0$.
(d) $o_{t_1} = \texttt{loc}_1$, $o_{t_2} = \texttt{loc}_1$.
(e) $o_{t_1} = \texttt{loc}_0$, $o_{t_2} = \texttt{loc}_0$.
(f) $o_{t_1} = \texttt{loc}_0$, $o_{t_2} = \texttt{loc}_1$.
(g) $o_{t_1} = \texttt{loc}_1$, $o_{t_2} = \texttt{loc}_0$.
(h) $o_{t_1} = \texttt{loc}_1$, $o_{t_2} = \texttt{loc}_1$.

Fig. 2: The above figures show the closed-loop trajectories (in blue) for different wind region locations (red ellipse) and noisy observations o_{t_1} and o_{t_2} that are collected at $t_1 = 4$ and $t_2 = 8$. The red dots represent the location of the drone when the observations are collected, and the figures' sub-captions detail which observations are collected in each scenario.

Proof: Let (15) be an optimal solution to problem (12). By definition, we have that (15) satisfies constraints (12a)–(12f) of problem (12). Furthermore, by construction we have that $x^{i,*}_{k|t} \in \mathcal{X}(e), \forall e \in \mathcal{C}^i_{k|t}$ implies that $\mathbb{P}(x^{i,*}_{k|t} \in \mathcal{X}(e)|b^{i,*}_{k|t}) \geq 1 - \epsilon$, which leads to the desired result. ■

Note that the result from Proposition 2 and the recursive feasibility from Theorem 1 imply that the chance constraint from (3) is satisfied at all times.

V. Navigation Example

We consider the following LTI system:

$$x_{t+1} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} x_t + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} u_t, \qquad (18)$$

where at each time t the state x_t collects the system position (X_t, Y_t) and velocity (v^x_t, v^y_t). The control action $u_t = [a^x_t, a^y_t]$, where a^x_t and a^y_t represent the acceleration subject to saturation constraints, i.e., $u_t \in \mathcal{U} = \{u \in \mathbb{R}^m : ||u||_\infty \leq 20\}$ for all time $t \in \{0, 1, \ldots\}$.

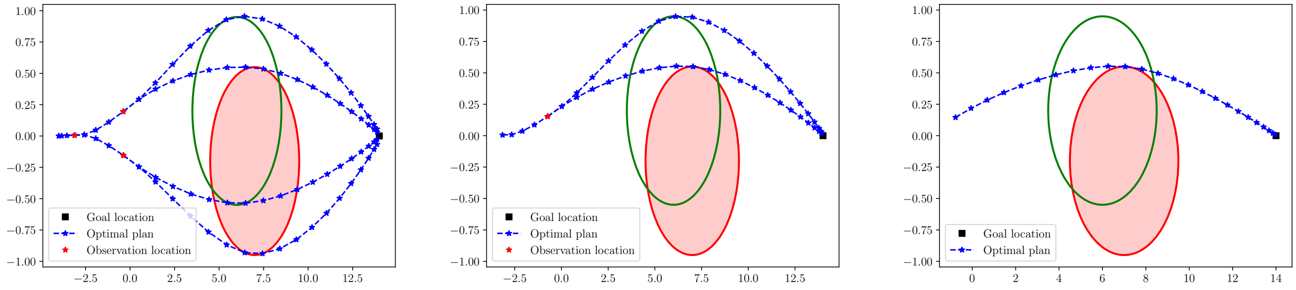
For the initial condition $x_0 = [-4, 0, 0, 0]$ and $b_0 = [0.5, 0.5]$, we tested the proposed strategy on a navigation task where a drone has to reach a goal state $x_{\texttt{goal}} = [14, 0, 0, 0]$, while avoiding with high probability a windy region $\mathcal{X}_{\texttt{wind}}$. The MPC problem with horizon $N = 22$ is solved with CasADi [22] and the cost function $h(x, u) = 0.1||x - x_{\texttt{goal}}||^2_2 + ||u||^2_2$ and $V(x) = 10^3||x - x_{\texttt{goal}}||^2_2$.[2] The exact location of the wind region is partially known and it has to be inferred by partial noisy observations. In particular, we know that the center of the windy region may be either at $\texttt{loc}_0 = [7, -0.2]$ or $\texttt{loc}_1 = [6, 0.2]$. We design a controller that avoids the windy region $\mathcal{X}_{\texttt{wind}}$ with high probability by enforcing the following chance constraint:

$$\mathbb{P}(x_t \notin \mathcal{X}_{\texttt{wind}}|b_t) = \mathbb{P}(x_t \in \mathcal{X}^c_{\texttt{wind}}|b_t) \geq 0.8, \qquad (19)$$

for all $t \in \{0, 1, \ldots\}$. In the above chance constraint, each element of the two-dimensional belief vector b_t represents the center of the windy region being in locations $\texttt{loc}_0$ or $\texttt{loc}_1$. The belief is computed based on the noisy observations collected at time $t_1 = 4$ and $t_2 = 8$. At time t_1 the sensor returns an observation that is exact with probability 0.6, and at t_2 the probability of the observations being correct is 0.75. Notice that as time passes the accuracy of the sensor increases as it would be in a real scenario, since we get closer to the area of interest. Furthermore, we assume to know the region where the measurements can be taken.

We performed the control task 1000 times by randomly sampling the wind location and the noisy observations collected by the controller. Out of these 1000 trials, the controller flew the drone over the windy region only 106 times. Thus, we verify that the chance constraint is empirically satisfied for the closed-loop system. We emphasize that the controller does not know the exact location of the windy area and the control action is computed based on noisy observations and the known sensor accuracy. Figure 2 shows the closed-loop trajectories for all possible wind locations and noisy observations. Notice that in the scenarios from Figures 2b, 2c, 2f, and 2g, the controller receives contradicting observations about the wind location, thus it decides to avoid both regions where the wind may be located. Indeed when the observations collected at time t_1 and t_2 are not in agreement, the controller does not have a strong belief about the wind location and to satisfy the chance constraint (19) it is forced to avoid both regions. On the other hand, when the two observations are in agreement the controller decides to fly over one of the possible windy areas, as shown in

[2] Code available online at: https://github.com/urosolia/Mixed_observable_MPC. All experiments are run on a 2015 Macbook Pro with a 2.5GHz i7 and 16 GB of RAM.



(a) Predicted strategy at time $t = 1$

(b) Predicted strategy at time $t = 5$

(c) Predicted strategy at time $t = 9$

Fig. 3: Trajectory tree planned by the MPC at different time steps for an experiment where $o_{t_1} = \texttt{loc}_0$ and $o_{t_2} = \texttt{loc}_0$.

Figures 2a, 2d, 2e, and 2h. It is important to underline that, as the sensor accuracy is 0.6 at time t_1 and 0.75 at time t_2, there is a low probability that both measurements are incorrect and that the controller flies over the wind region.

Figure 3 shows the planned tree of trajectories at time $t \in \{1, 5, 9\}$ for an experiment where $o_{t_1} = \texttt{loc}_0$ and $o_{t_2} = \texttt{loc}_0$. Note that observations about the wind location are collected at time $t \in \mathcal{T}_{\text{obs}} = \{t_1 = 4, t_2 = 8\}$. Thus, for time $t < t_1$ the controller plans a trajectory tree that branches twice, as the controller will behave differently as a function of the collected observation (Fig. 3a). For $t_1 < t < t_2$, the controller plans a trajectory that branches once, as only one observation will be collected in the future (Fig. 3b). Finally, for $t > t_2$ the controller plans a single trajectory (Fig. 3c). This example shows that the tree of trajectories encodes a policy where each branch represents how the closed-loop system would evolve depending on the collected observations. Most importantly, we notice that each branch satisfies different constraints, i.e., the planned trajectory avoids either the wind location #1 (red ellipse), the wind location #2 (green ellipse), or both regions. These constraints are computed via Algorithm 1 and they allow us to guarantee chance constraints satisfaction.

We compare the proposed approach with a scenario MPC, where the optimization problem is carried out over a trajectory tree and in each branch only the constraint associated with one environment mode is considered, as in [11], [13]. Table I shows the percentage of constraint violations over 1000 random simulations. Notice that only the proposed approach empirically satisfies the chance constraint (19).

TABLE I: Comparison with a scenario MPC approach.

	Proposed Approach	Scenario MPC
% of constraint violation	10.6%	25.5%

VI. Conclusions

We presented an MPC design for autonomous systems operating in partially observable discrete environments. First, we reformulated the MPC problem as a finite-dimensional optimization problem over a trajectory tree. Then, we presented an algorithm to compute the constraints enforced at each tree branch. We demonstrated that our approach guarantees recursive feasibility and chance constraint satisfaction.

References

[1] K. J. Åström, *Introduction to stochastic control theory*. Courier Corporation, 2012.

[2] D. P. Joseph *et al.*, "On linear control theory," *Transactions of the American Institute of Electrical Engineers, Part II: Applications and Industry*, vol. 80, no. 4, pp. 193–196, 1961.

[3] D. Q. Mayne *et al.*, "Robust output feedback model predictive control of constrained linear systems," *Automatica*, vol. 42, no. 7, pp. 1217–1222, 2006.

[4] E. J. Sondik, "The optimal control of partially observable Markov processes over the infinite horizon: Discounted costs," *Operations Research*, vol. 26, no. 2, pp. 282–304, 1978.

[5] P. Poupart *et al.*, "Approximate linear programming for constrained partially observable markov decision processes," in *Twenty-ninth AAAI Conference on Artificial Intelligence*, 2015.

[6] D. Kim *et al.*, "Point-based value iteration for constrained POMDPs," in *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[7] J. Pineau *et al.*, "Point-based value iteration: An anytime algorithm for POMDPs," in *IJCAI*, vol. 3, 2003, pp. 1025–1032.

[8] M. Bouton *et al.*, "Point-based methods for model checking in partially observable markov decision processes." in *AAAI*, 2020, pp. 10 061–10 068.

[9] G. Shani *et al.*, "A survey of point-based POMDP solvers," *Autonomous Agents and Multi-Agent Systems*, vol. 27, no. 1, pp. 1–51, 2013.

[10] S. C. Ong *et al.*, "Planning under uncertainty for robotic tasks with mixed observability," *The International Journal of Robotics Research*, vol. 29, no. 8, pp. 1053–1068, 2010.

[11] S. H. Nair *et al.*, "Stochastic MPC with multi-modal predictions for traffic intersections," *arXiv preprint arXiv:2109.09792*, 2021.

[12] I. Batkovic *et al.*, "A robust scenario MPC approach for uncertain multi-modal obstacles," *IEEE Control Systems Letters*, vol. 5, no. 3, pp. 947–952, 2020.

[13] Y. Chen *et al.*, "Interactive multi-modal motion planning with branch model predictive control," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5365–5372, 2022.

[14] H. Hu *et al.*, "Active uncertainty reduction for human-robot interaction: An implicit dual control approach," *arXiv preprint arXiv:2202.07720*, 2022.

[15] E. Arcari *et al.*, "Dual stochastic MPC for systems with parametric and structural uncertainty," in *Learning for Dynamics and Control*. PMLR, 2020, pp. 894–903.

[16] U. Rosolia *et al.*, "The mixed-observable constrained linear quadratic regulator problem: the exact solution and practical algorithms," *IEEE Transactions on Automatic Control*, 2022.

[17] J. P. Alsterda *et al.*, "Contingency model predictive control for linear time-varying systems," *arXiv preprint arXiv:2102.12045*, 2021.

[18] J. P. Alsterda *et al.*, "Contingency model predictive control for automated vehicles," in *2019 American Control Conference (ACC)*. IEEE, 2019, pp. 717–722.

[19] L. Svensson *et al.*, "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.

[20] P. O. Scokaert *et al.*, "Min-max feedback model predictive control for constrained linear systems," *IEEE Transactions on Automatic Control*, vol. 43, no. 8, pp. 1136–1142, 1998.

[21] M. L. Puterman, "Markov decision processes," *Handbooks in operations research and management science*, vol. 2, pp. 331–434, 1990.

[22] J. A. E. Andersson *et al.*, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.