Experience-driven Predictive Control with Robust Constraint Satisfaction under Time-Varying State Uncertainty

Vishnu R. Desaraju, Alexander E. Spitzer, and Nathan Michael
The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA
{rajeswar, spitzer, nmichael}@cmu.edu

***Abstract*—We present an extension to Experience-driven Predictive Control (EPC) that leverages a Gaussian belief propagation strategy to compute an uncertainty set bounding the evolution of the system state in the presence of time-varying state uncertainty. This uncertainty set is used to tighten the constraints in the predictive control formulation via a chance constrained approach, thereby providing a probabilistic guarantee of constraint satisfaction. The parameterized form of the controllers produced by EPC coupled with online uncertainty estimates ensures this robust constraint satisfaction property persists even as the system switches controllers and experiences variations in the uncertainty model. We validate the online performance and robust constraint satisfaction of the proposed Robust EPC algorithm through a series of experimental trials with a small quadrotor platform subjected to changes in state estimate quality.**

I. Introduction

Autonomous robotic systems operating in uncertain, real-world environments must be able to track trajectories safely and reliably while obeying system limitations (e.g., actuator constraints) and operational constraints (e.g., speed limits for traversing a region of the environment or to satisfy sensor limitations). However, these systems have inherently noisy sensing and perception systems that produce state estimates with variable uncertainty. This is particularly challenging for small, agile systems, such as micro air vehicles (MAVs) that are limited in their computation and sensing capabilities by size, weight, and power restrictions. Therefore, in this work we aim to develop a computationally efficient predictive control methodology that leverages uncertainty information from the state estimator to ensure constraints on the system state and control inputs are satisfied, even in the presence of time-varying state uncertainty (illustrated in Fig. 1). We specifically focus on the problem of robust constraint satisfaction in predictive control, as the predictive formulation permits anticipating and mitigating future uncertainty while retaining an optimal control framework [21].

There are two general classes of approaches for mitigating the effects of uncertainty in predictive control. Adaptive formulations seek to estimate the uncertainty in the dynamics and update the predictive model to more accurately anticipate the system's interaction with the constraints [4, 8, 11, 27]. However, in practice, this may still lead to constraint violations due to the difference in timescales between the disturbance estimator and high-frequency noise in the state estimate.

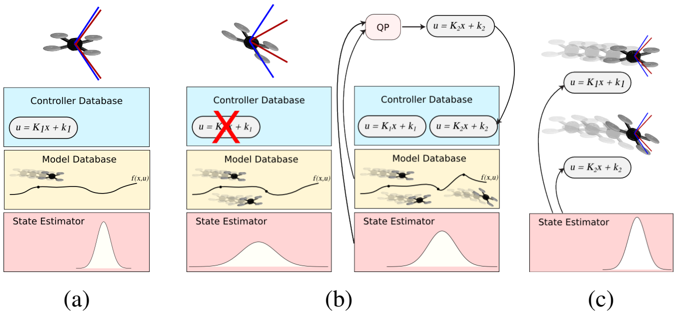


Fig. 1: Overview of the proposed approach that combines an online learned controller database with estimates of the dynamics model and state uncertainty. As uncertainty changes, the tightened constraints (red) on the MAV automatically adjust to ensure robust satisfaction of the requested constraints (blue), even as the MAV switches between controllers. In panel (a), the existing controller in the experience database is determined to be optimal and applied. Panel (b) shows the addition of a new controller to the database to accommodate higher sensor uncertainty. In panel (c), the state uncertainty parameterizes all controllers in the database as they are reused.

In contrast, robust formulations refine the constraints to explicitly account for high-frequency uncertainty. Robust MPC techniques provide constraint satisfaction guarantees in the presence of bounded, uncertain parameters [19, 20, 23, 30]. For linear dynamics, the effects of bounded uncertainty can be represented by disturbance-invariant sets [17] that can be used to tighten the set of feasible states and inputs (via the Pontryagin difference operation). In the nonlinear case, this can be generalized to min-max formulations to optimize with respect to the maximal state deviations [1]. These techniques yield more conservative controllers than the adaptive approaches, but as a result, are able to account for any variations within the bounded uncertainty set without requiring a disturbance estimator that can track rapid variations.

A subset of these Robust MPC techniques employ local feedback control laws to restrict the anticipated growth of uncertainty. This yields constraint tightening and Tube MPC approaches that enable more aggressive performance [18, 23, 30]. While many formulations assume the uncertainty set is known a priori (e.g., as a disturbance invariant set or via the min-max calculation), some approaches permit online modification of robustness bounds driven by online estimates of the uncertainty bounds [31]. An extension of this idea replaces the deterministic uncertainty set with a probabilistic representation, e.g., as a multivariate Gaussian distribution [35]. This enables the use of a Kalman filter to predict the evolution of state uncertainty instead of the recursive Pontryagin difference operations required for deterministic sets [21]. Many adaptive MPC formulations also include a robust component that is coupled to estimator uncertainty [1, 10, 11]. The resulting robust-adaptive formulations allow the adaptive component to estimate and compensate for low frequency components of the uncertainty, while variability about the current estimate is handled by the robust constraints.

In addition to constraint satisfaction under uncertainty, the other core challenge for applying predictive control strategies to systems such as MAVs stem from their onboard computational constraints. That is, we require the ability to compute predictive control commands at sufficiently high rates to ensure stability of these agile systems. Fast MPC solution strategies can be divided into four categories: leveraging fast online optimization techniques [26], optimizing approximate formulations [15], explicit enumeration of equivalent controllers [2], and semi-explicit approaches [7, 8, 28]. In this work, we consider this last class of techniques due to the reduced reliance on online optimization in a critical control loop and their scalability to available computational resources [28]. Within the class of semi-explicit approaches, the Experience-driven Predictive Control (EPC) algorithm [8] extends this efficient formulation to mitigate the effects of low-frequency disturbances to the system dynamics.

Therefore, we propose a constrained, predictive control strategy that leverages EPC for computational efficiency and adaptation to low-frequency components of the uncertainty. We extend the underlying control problem to a chance-constrained Tube MPC formulation to capture the effects of time-varying state uncertainty (e.g., due to sensors with environment-dependent performance) in the robustness bounds. The resulting Robust EPC algorithm ensures probabilistic constraint satisfaction in the presence of state uncertainty modeled by a multivariate Gaussian distribution, e.g., provided by a Kalman filter based state estimator.

II. Approach

In this section, we present an extension of the Experience-driven Predictive Control (EPC) algorithm [8] to achieve high-rate predictive control with robust constraint satisfaction. EPC constructs online a two-part experience database consisting of previously used locally optimal controllers and observed perturbations to the system's dynamics model (illustrated by the blue and yellow boxes in Fig. 1). The controllers are parameterized by the dynamics model, and thus they automatically adapt to changes in the model. We therefore propose the Robust EPC algorithm by similarly parameterizing the controllers in the database by an online updated estimate of the uncertainty in the system state. This estimate is derived from the state estimator covariance and enables the use of a belief propagation approach to construct an uncertainty tube for the evolution of the state over the prediction horizon.

A. Adaptive Stochastic Dynamics Model

We consider the general nonlinear dynamics and observation models

$$\begin{aligned}\mathbf{x}_{k+1} &= f(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{w}_k \\ \mathbf{z}_k &= h(\mathbf{x}_k) + \mathbf{v}_k\end{aligned} \tag{1}$$

where $\mathbf{x}_k \in \mathbb{R}^n$ is the system state, $\mathbf{u}_k \in \mathbb{R}^m$ is the control input, and $\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{W}_k)$ and $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{V}_k)$ denote the process and measurement uncertainty, respectively. The corresponding first order approximations about a nominal state $\mathbf{x}^*$ and nominal control $\mathbf{u}^*$ are

$$\begin{aligned}\mathbf{x}_{k+1} &\approx \mathbf{A}_k(\mathbf{x}_k - \mathbf{x}^*) + \mathbf{B}_k(\mathbf{u}_k - \mathbf{u}^*) + \tilde{\mathbf{c}} + \mathbf{w}_k \\ \mathbf{z}_k &\approx \mathbf{C}_k(\mathbf{x}_k - \mathbf{x}^*) + \mathbf{v}_k\end{aligned} \tag{2}$$

where $\tilde{\mathbf{c}}$ is the estimate of the approximation error. Updating this estimate via an online learning strategy also captures the effects of unmodeled dynamics, thus enabling adaptation to external perturbations (detailed in Sect. II-D).

To model the evolution of this uncertain system, we leverage the existence of a closed-form belief propagation law for Gaussian distributions [32] and extend (1) to a standard EKF belief state update law that yields an estimate of the state, $\mathbf{x}_k \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$,

$$\begin{aligned}\boldsymbol{\mu}_{k+1} &= f(\boldsymbol{\mu}_k, \mathbf{u}_k) + \mathbf{P}_k\mathbf{C}_k^{\mathrm{T}}\mathbf{L}_k^{-1}\left(\mathbf{z}_{k+1} - h(\boldsymbol{\mu}_k)\right) \\ \boldsymbol{\Sigma}_{k+1} &= \mathbf{P}_k - \mathbf{P}_k\mathbf{C}_k^{\mathrm{T}}\mathbf{L}_k^{-1}\mathbf{C}_k\mathbf{P}_k\end{aligned}$$

where $\mathbf{P}_k = \mathbf{A}_k\boldsymbol{\Sigma}_k\mathbf{A}_k^{\mathrm{T}} + \mathbf{W}_k$ and $\mathbf{L}_k = \mathbf{C}_k\mathbf{P}_k\mathbf{C}_k^{\mathrm{T}} + \mathbf{V}_k$. Following Platt et al. [16], we take $\mathbf{z}_{k+1} = h(\boldsymbol{\mu}_k)$ as the maximum likelihood observation to obtain a simplified belief state update law

$$\begin{aligned}\boldsymbol{\mu}_{k+1} &= f(\boldsymbol{\mu}_k, \mathbf{u}_k) \\ \boldsymbol{\Sigma}_{k+1} &= \mathbf{P}_k - \mathbf{P}_k\mathbf{C}_k^{\mathrm{T}}\mathbf{L}_k^{-1}\mathbf{C}_k\mathbf{P}_k\end{aligned} \tag{3}$$

B. Chance-constrained Tube MPC

To incorporate this uncertainty propagation model into a robust control framework, we propose a Tube MPC formulation where the control applied to the system, $\mathbf{u}_k^S$, is the combination of the MPC output, $\mathbf{u}_k$, and an ancillary stabilizing controller with gain matrix $\mathbf{S}_k$,

$$\mathbf{u}_k^S = \mathbf{u}_k + \mathbf{S}_k(\mathbf{x}_k - \boldsymbol{\mu}_k) \tag{4}$$

This gain, $\mathbf{S}_k$, is designed to stabilize the nominal system via an unconstrained MPC formulation [24] given in Sect. II-C1. The introduction of the ancillary controller restricts deviations from the predicted state mean [21] and enables the MPC formulation to account for the reduction in uncertainty due to local feedback. This results in a slight change in the belief state update law,

$$\mathbf{P}_k = (\mathbf{A}_k - \mathbf{B}_k\mathbf{S}_k)\boldsymbol{\Sigma}_k(\mathbf{A}_k - \mathbf{B}_k\mathbf{S}_k)^{\mathrm{T}} + \mathbf{W}_k$$

The Tube MPC formulation also enforces state and input constraints, $\mathbf{x}_k \in \mathcal{X}_k, \mathbf{u}_k^S \in \mathcal{U}_k$. In this work, we assume the admissible state sets, $\mathcal{X}_k$, and input sets, $\mathcal{U}_k$, are polytopic, or can be approximated by polytopes. This yields a set of half-plane constraints,

$$\begin{aligned} \mathbf{G}_\mathbf{x}(\mathbf{x}_{k+1} - \mathbf{x}^*) &\leq \mathbf{g}_\mathbf{x} \\ \mathbf{G}_\mathbf{u}(\mathbf{u}_k^S - \mathbf{u}^*) &\leq \mathbf{g}_\mathbf{u} \end{aligned} \tag{5}$$

However, due to the stochastic dynamics model, we instead employ a chance constrained formulation by requiring (5) to hold with probability $1 - \alpha$,

$$\begin{aligned} P\left(\mathbf{G}_\mathbf{x}(\mathbf{x}_{k+1} - \mathbf{x}^*) \leq \mathbf{g}_\mathbf{x}\right) &\geq 1 - \alpha \\ P\left(\mathbf{G}_\mathbf{u}(\mathbf{u}_k^S - \mathbf{u}^*) \leq \mathbf{g}_\mathbf{u}\right) &\geq 1 - \alpha \end{aligned} \tag{6}$$

Given that the belief state corresponds to a multivariate Gaussian, $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, its probability mass level sets are ellipsoids defined by a χ^2 value. The ellipsoid containing $1 - \alpha$ of the probability mass is given by $(\mathbf{x} - \boldsymbol{\mu})^\mathrm{T}\boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}) = \chi_n^2(\alpha)$. Therefore, a given chance constraint threshold, $1 - \alpha$, yields an ellipsoid defining the state uncertainty bounds.

Ensuring robust constraint satisfaction requires tightening (5) by these bounds [21], as illustrated in Fig. 2. Consequently, to retain the linear form of the constraints, we follow Domes et al. [9] to approximate the ellipsoid by its axis-aligned bounding box with side lengths given by

$$\boldsymbol{\delta}_{k+1}^\mathbf{x} = \sqrt{\chi_n^2(\alpha)\mathrm{diag}(\boldsymbol{\Sigma}_{k+1})} \tag{7}$$

where $\mathrm{diag}(\cdot)$ returns the diagonal elements of the argument as a vector.

While the MPC output, $\mathbf{u}_k$, does not introduce any control input uncertainty, the ancillary controller is a function of the uncertain future state. This yields a similar bound on the control command,

$$\boldsymbol{\delta}_k^\mathbf{u} = \sqrt{\chi_n^2(\alpha)\mathrm{diag}(\mathbf{S}_k\boldsymbol{\Sigma}_k\mathbf{S}_k^\mathrm{T})} \tag{8}$$

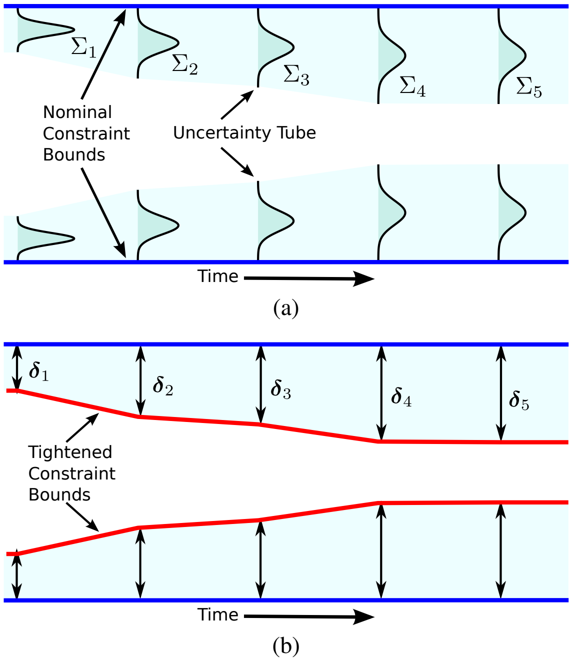
Given these bounding box dimensions, we convert the probabilistic state and input constraints (6) to tightened deterministic constraints, $\mathbf{x}_k \in \tilde{\mathcal{X}}_k, \mathbf{u}_k \in \tilde{\mathcal{U}}_k$,

$$\begin{aligned} \mathbf{G}_\mathbf{x}(\boldsymbol{\mu}_{k+1} - \mathbf{x}^*) &\leq \mathbf{g}_\mathbf{x} - \mathbf{G}_\mathbf{x}\boldsymbol{\delta}_{k+1}^\mathbf{x} = \tilde{\mathbf{g}}_\mathbf{x} \\ \mathbf{G}_\mathbf{u}(\mathbf{u}_k - \mathbf{u}^*) &\leq \mathbf{g}_\mathbf{u} - \mathbf{G}_\mathbf{u}\boldsymbol{\delta}_k^\mathbf{u} = \tilde{\mathbf{g}}_\mathbf{u} \end{aligned} \tag{9}$$

Although the bounding box is generally a conservative approximation of the ellipsoid, we observe that for any axis-aligned box constraint, tightening by the bounding box is equivalent to the exact approach of tightening by the axis-aligned suprema over the ellipsoid [5].

C. Robust EPC formulation

Although this chance-constrained Tube MPC formulation permits an optimization-based solution, in this work, we propose a novel extension to the Experience-driven Predictive Control (EPC) algorithm [8] to enable Robust MPC on computationally constrained systems. The proposed Robust EPC algorithm leverages this tube-based formulation to enforce robust constraint satisfaction while retaining the computational efficiency and model adaptation properties of EPC.



(a)

(b)

Fig. 2: Two-dimensional illustration of constraint tightening: (a) nominal state constraints (blue line) with the predicted Gaussian uncertainty tube ($\boldsymbol{\Sigma}_i$) define (b) chance-constraint bounds ($\boldsymbol{\delta}_i$) that yield tightened constraints (red).

As in EPC, we can formulate the receding-horizon control problem as a quadratic program (QP) due to the model adaptation term, $\tilde{\mathbf{c}}$, that captures the nonlinearities and other unmodeled dynamics. The QP is formulated about a nominal state, $\mathbf{x}^*$ and input, $\mathbf{u}^*$, to track a sequence of N reference states $\mathbf{r}_1, \ldots, \mathbf{r}_N$,

$$\begin{aligned} \underset{\bar{\mathbf{u}}_k}{\mathrm{argmin}} \sum_{k=0}^{N-1} &\frac{1}{2}(\bar{\mathbf{x}}_{k+1} - \bar{\mathbf{r}}_{k+1})^\mathrm{T}\mathbf{Q}(\bar{\mathbf{x}}_{k+1} - \bar{\mathbf{r}}_{k+1}) \\ &+ \frac{1}{2}(\bar{\mathbf{u}}_k - \bar{\mathbf{u}}_\mathbf{c})^\mathrm{T}\mathbf{R}(\bar{\mathbf{u}}_k - \bar{\mathbf{u}}_\mathbf{c}) \\ \text{s.t.} \quad &\bar{\mathbf{x}}_{k+1} = \mathbf{A}\bar{\mathbf{x}}_k + \mathbf{B}\bar{\mathbf{u}}_k + \tilde{\mathbf{c}} \\ &\mathbf{G}_\mathbf{x}\bar{\mathbf{x}}_{k+1} \leq \tilde{\mathbf{g}}_\mathbf{x}, \quad \mathbf{G}_\mathbf{u}\bar{\mathbf{u}}_k \leq \tilde{\mathbf{g}}_\mathbf{u} \\ &\forall\ k = 0, \ldots, N-1 \end{aligned} \tag{10}$$

where $\bar{\mathbf{x}}_k = \boldsymbol{\mu}_k - \mathbf{x}^*$, $\bar{\mathbf{r}}_k = \mathbf{r}_k - \mathbf{x}^*$, and $\bar{\mathbf{u}}_k = \mathbf{u}_k - \mathbf{u}^*$. If it is possible to derive a control input, $\bar{\mathbf{u}}_\mathbf{c}$, from the model adaptation term (e.g., if $\tilde{\mathbf{c}}$ is an acceleration disturbance, $\bar{\mathbf{u}}_\mathbf{c}$ would be the corresponding force) we subtract it in the cost function to avoid penalizing model error compensation [8].

Given that we can forward predict the mean and covariance evolution via (3), we can simplify notation by defining $\boldsymbol{x} = \left[\bar{\mathbf{x}}_1^\mathrm{T}, \ldots, \bar{\mathbf{x}}_N^\mathrm{T}\right]^\mathrm{T}$, $\boldsymbol{r} = \left[\bar{\mathbf{r}}_1^\mathrm{T}, \ldots, \bar{\mathbf{r}}_N^\mathrm{T}\right]^\mathrm{T}$, $\boldsymbol{u} = \left[\bar{\mathbf{u}}_0^\mathrm{T}, \ldots, \bar{\mathbf{u}}_{N-1}^\mathrm{T}\right]^\mathrm{T}$, $\boldsymbol{u}_\mathbf{c} = \left[\bar{\mathbf{u}}_\mathbf{c}^\mathrm{T}, \ldots, \bar{\mathbf{u}}_\mathbf{c}^\mathrm{T}\right]^\mathrm{T}$,

$$\boldsymbol{\mathcal{B}} = \begin{bmatrix} \mathbf{B} & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{AB} & \mathbf{B} & \ldots & \mathbf{0} \\ \vdots & \vdots & \ddots & \\ \mathbf{A}^{N-1}\mathbf{B} & \mathbf{A}^{N-2}\mathbf{B} & \ldots & \mathbf{B} \end{bmatrix}, \ \boldsymbol{c} = \begin{bmatrix} \tilde{\mathbf{c}} \\ (\mathbf{A} + \mathbf{I})\,\tilde{\mathbf{c}} \\ \vdots \\ \sum_{i=0}^{N-1} \mathbf{A}^i\tilde{\mathbf{c}} \end{bmatrix},$$

$\boldsymbol{\mathcal{Q}} = \operatorname{diag}(\mathbf{Q}, \dots, \mathbf{Q})$, $\boldsymbol{\mathcal{R}} = \operatorname{diag}(\mathbf{R}, \dots, \mathbf{R})$, $\boldsymbol{\mathcal{G}}_{\mathbf{x}} = \operatorname{diag}(\mathbf{G}_{\mathbf{x}}, \dots, \mathbf{G}_{\mathbf{x}})$, and $\boldsymbol{\mathcal{G}}_{\mathbf{u}} = \operatorname{diag}(\mathbf{G}_{\mathbf{u}}, \dots, \mathbf{G}_{\mathbf{u}})$, where $\operatorname{diag}(\cdot)$ here diagonally concatenates matrices. Similarly, let $\boldsymbol{g}_{\mathbf{x}} = \left[\tilde{\mathbf{g}}_{\mathbf{x}}^{\mathrm{T}}, \dots, \tilde{\mathbf{g}}_{\mathbf{x}}^{\mathrm{T}}\right]^{\mathrm{T}}$ and $\boldsymbol{g}_{\mathbf{u}} = \left[\tilde{\mathbf{g}}_{\mathbf{u}}^{\mathrm{T}}, \dots, \tilde{\mathbf{g}}_{\mathbf{u}}^{\mathrm{T}}\right]^{\mathrm{T}}$ to capture the tightened constraints (9).

Finally, we define $\boldsymbol{\mu}_0$ to be a parameter of the optimization constrained by the current state (see Sect. II-C2) rather than directly using the current state as in EPC. Therefore, the nominal state, $\mathbf{x}^* = \boldsymbol{\mu}_0$, $\bar{\mathbf{x}}_0 = \mathbf{0}$, and (10) simplifies to

$$\begin{aligned} \underset{\boldsymbol{u}}{\operatorname{argmin}} \frac{1}{2}(\boldsymbol{x} - \boldsymbol{r})^{\mathrm{T}} \boldsymbol{\mathcal{Q}} (\boldsymbol{x} - \boldsymbol{r}) &+ \frac{1}{2}(\boldsymbol{u} - \boldsymbol{u}_{\mathbf{c}})^{\mathrm{T}} \boldsymbol{\mathcal{R}} (\boldsymbol{u} - \boldsymbol{u}_{\mathbf{c}}) \\ \text{s.t.} \quad \boldsymbol{x} = \boldsymbol{\mathcal{B}} \boldsymbol{u} + \boldsymbol{c}, \quad \boldsymbol{\mathcal{G}}_{\mathbf{x}} \boldsymbol{x} &\le \boldsymbol{g}_{\mathbf{x}}, \quad \boldsymbol{\mathcal{G}}_{\mathbf{u}} \boldsymbol{u} \le \boldsymbol{g}_{\mathbf{u}} \end{aligned}$$

Incorporating the dynamics into the cost and constraints yields an equivalent QP that facilitates the state space partitioning and local controller computation steps of EPC,

$$\begin{aligned} &\underset{\boldsymbol{u}}{\operatorname{argmin}} \ \frac{1}{2} \boldsymbol{u}^{\mathrm{T}} \boldsymbol{\mathcal{H}} \boldsymbol{u} + \boldsymbol{h}^{\mathrm{T}} \boldsymbol{u} \\ &\text{s.t.} \quad \boldsymbol{\Gamma} \boldsymbol{u} \le \boldsymbol{\gamma} \end{aligned} \tag{11}$$

where $\boldsymbol{\mathcal{H}} = \boldsymbol{\mathcal{B}}^{\mathrm{T}} \boldsymbol{\mathcal{Q}} \boldsymbol{\mathcal{B}} + \boldsymbol{\mathcal{R}}$, $\boldsymbol{h} = \boldsymbol{\mathcal{B}}^{\mathrm{T}} \boldsymbol{\mathcal{Q}} (\boldsymbol{c} - \boldsymbol{r}) - \boldsymbol{\mathcal{R}} \boldsymbol{u}_{\mathbf{c}}$,

$$\boldsymbol{\Gamma} = \begin{bmatrix} \boldsymbol{\mathcal{G}}_{\mathbf{x}} \boldsymbol{\mathcal{B}} \\ \boldsymbol{\mathcal{G}}_{\mathbf{u}} \end{bmatrix}, \text{ and } \boldsymbol{\gamma} = \begin{bmatrix} \boldsymbol{g}_{\mathbf{x}} - \boldsymbol{\mathcal{G}}_{\mathbf{x}} \boldsymbol{c} \\ \boldsymbol{g}_{\mathbf{u}} \end{bmatrix}$$

As in EPC, the partitioning of the state-space for Robust EPC is determined by the Karush-Kuhn-Tucker (KKT) conditions for optimality,

$$\begin{aligned} \boldsymbol{\mathcal{H}} \boldsymbol{u} + \boldsymbol{h} + \Gamma^{\mathrm{T}} \boldsymbol{\lambda} &= \mathbf{0} \\ \boldsymbol{\Lambda} (\boldsymbol{\Gamma} \boldsymbol{u} - \boldsymbol{\gamma}) &= \mathbf{0} \end{aligned} \tag{12}$$

where $\boldsymbol{\lambda}$ is the vector of Lagrange multipliers and $\boldsymbol{\Lambda} = \operatorname{diag}(\boldsymbol{\lambda})$. Therefore, given a set of active constraints (i.e., with $\lambda > 0$), we can solve for the optimal control sequence $\boldsymbol{u}$ and corresponding $\boldsymbol{\lambda}$ by solving a linear system derived from (12),

$$\begin{bmatrix} \boldsymbol{\mathcal{H}} & \boldsymbol{\Gamma}_a^{\mathrm{T}} \\ \boldsymbol{\Gamma}_a & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{\lambda}_a \end{bmatrix} = \begin{bmatrix} -\boldsymbol{h} \\ \boldsymbol{\gamma}_a \end{bmatrix}$$

where the subscript a denotes rows corresponding to active constraints. For any linearly independent set of active constraints [3], the resulting $\boldsymbol{u}$ is affine in the predicted state mean error, $\boldsymbol{r}$,

$$\boldsymbol{u} = \boldsymbol{\mathcal{E}}_5 \boldsymbol{r} - \left(\boldsymbol{\mathcal{E}}_5 \boldsymbol{c} - \boldsymbol{\mathcal{E}}_4 \boldsymbol{\mathcal{R}} \boldsymbol{u}_{\mathbf{c}} + \boldsymbol{\mathcal{E}}_3 \begin{bmatrix} \boldsymbol{g}_{\mathbf{x}}^{+} - \boldsymbol{\mathcal{G}}_{\mathbf{x}} \boldsymbol{c} \\ -\boldsymbol{g}_{\mathbf{x}}^{-} + \boldsymbol{\mathcal{G}}_{\mathbf{x}} \boldsymbol{c} \\ \boldsymbol{g}_{\mathbf{u}}^{+} \\ -\boldsymbol{g}_{\mathbf{u}}^{-} \end{bmatrix}_a \right) \tag{13}$$

where $\boldsymbol{\mathcal{E}}_1 = \boldsymbol{\Gamma}_a \boldsymbol{\mathcal{H}}^{-1}$, $\boldsymbol{\mathcal{E}}_2 = -(\boldsymbol{\mathcal{E}}_1 \boldsymbol{\Gamma}_a^{\mathrm{T}})^{-1}$, $\boldsymbol{\mathcal{E}}_3 = \boldsymbol{\mathcal{E}}_1^{\mathrm{T}} \boldsymbol{\mathcal{E}}_2$, $\boldsymbol{\mathcal{E}}_4 = \boldsymbol{\mathcal{H}}^{-1} + \boldsymbol{\mathcal{E}}_3 \boldsymbol{\mathcal{E}}_1$, and $\boldsymbol{\mathcal{E}}_5 = \boldsymbol{\mathcal{E}}_4 \boldsymbol{\mathcal{B}}^{\mathrm{T}} \boldsymbol{\mathcal{Q}}$. Moreover, the coefficients in (13) are all functions of $\mathbf{A}, \mathbf{B}, \tilde{\mathbf{c}}, \boldsymbol{\delta}^{\mathbf{x}}$, and $\boldsymbol{\delta}^{\mathbf{u}}$. Therefore, the final control law $\boldsymbol{\kappa}(\mathbf{x}_0, \mathbf{r}_1, \dots, \mathbf{r}_N)$ is given by a parameterized feedback gain matrix $\mathbf{K}$, a parameterized feedforward vector $\mathbf{k}_{\mathrm{ff}}$, and the ancillary control gain matrix, $\mathbf{S}$,

$$\begin{aligned} \boldsymbol{\kappa}(\mathbf{x}_0, \mathbf{r}_1, \dots, \mathbf{r}_N) &= \mathbf{K}(\mathbf{A}, \mathbf{B}, \tilde{\mathbf{c}}, \boldsymbol{\delta}^{\mathbf{x}}, \boldsymbol{\delta}^{\mathbf{u}}) \boldsymbol{r} \\ &+ \mathbf{k}_{\mathrm{ff}}(\mathbf{A}, \mathbf{B}, \tilde{\mathbf{c}}, \boldsymbol{\delta}^{\mathbf{x}}, \boldsymbol{\delta}^{\mathbf{u}}) \\ &+ \left[\mathbf{S}_0(\mathbf{x}_0 - \boldsymbol{\mu}_0)^{\mathrm{T}}, \dots, \mathbf{S}_{N-1}(\mathbf{x}_{N-1} - \boldsymbol{\mu}_{N-1})^{\mathrm{T}}\right]^{\mathrm{T}} \end{aligned} \tag{14}$$

The KKT matrices that determine whether a previously computed controller is locally optimal are similarly parameterized, and the active Lagrange multipliers, $\boldsymbol{\lambda}_a$, are given by

$$\boldsymbol{\lambda}_a = -\boldsymbol{\mathcal{E}}_6 \boldsymbol{r} + \left(\boldsymbol{\mathcal{E}}_6 \boldsymbol{c} - \boldsymbol{\mathcal{E}}_3^{\mathrm{T}} \boldsymbol{\mathcal{R}} \boldsymbol{u}_{\mathbf{c}} + \boldsymbol{\mathcal{E}}_2 \begin{bmatrix} \boldsymbol{g}_{\mathbf{x}}^{+} - \boldsymbol{\mathcal{G}}_{\mathbf{x}} \boldsymbol{c} \\ -\boldsymbol{g}_{\mathbf{x}}^{-} + \boldsymbol{\mathcal{G}}_{\mathbf{x}} \boldsymbol{c} \\ \boldsymbol{g}_{\mathbf{u}}^{+} \\ -\boldsymbol{g}_{\mathbf{u}}^{-} \end{bmatrix}_a \right) \tag{15}$$

where $\boldsymbol{\mathcal{E}}_6 = \boldsymbol{\mathcal{E}}_3^{\mathrm{T}} \boldsymbol{\mathcal{B}}^{\mathrm{T}} \boldsymbol{\mathcal{Q}}$. Therefore, given a set of active constraints, the corresponding controller and KKT matrices can be reconstructed online using (13), (15), and the current $\mathbf{A}, \mathbf{B}, \tilde{\mathbf{c}}, \boldsymbol{\delta}^{\mathbf{x}}$ and $\boldsymbol{\delta}^{\mathbf{u}}$. Therefore, each controller automatically evolves with both the estimated system dynamics and state uncertainty. This also enables the construction of a controller database that recovers the functionality of (10) by switching between controllers according to the KKT conditions, thus providing the foundation for the Robust EPC algorithm detailed in Sect. II-E.

1) Ancillary Controller: In addition to the introduction of a chance-constrained formulation, the extension of EPC to Robust EPC requires two key components. The first is an ancillary controller that aims to drive the current state, $\mathbf{x}_k$, (now treated as deterministic) to the state mean sequence, $\boldsymbol{\mu}_k$, produced by (11). The corresponding unconstrained MPC formulation,

$$\underset{\mathbf{u}_k}{\operatorname{argmin}} \sum_{k=0}^{N-1} \frac{1}{2}(\mathbf{x}_{k+1} - \boldsymbol{\mu}_{k+1})^{\mathrm{T}} \mathbf{Q} (\mathbf{x}_{k+1} - \boldsymbol{\mu}_{k+1}) + \frac{1}{2} \mathbf{u}_k^{\mathrm{T}} \mathbf{R} \mathbf{u}_k$$

yields an equivalent set of feedback control gains computed analogously to (14) without constraints,

$$\operatorname{diag}(\mathbf{S}_0, \dots, \mathbf{S}_{N-1}) = (\boldsymbol{\mathcal{B}}^{\mathrm{T}} \boldsymbol{\mathcal{Q}} \boldsymbol{\mathcal{B}} + \boldsymbol{\mathcal{R}})^{-1} \boldsymbol{\mathcal{B}}^{\mathrm{T}} \boldsymbol{\mathcal{Q}} \tag{16}$$

2) Initial State Selection: The second component is the initial state mean parameter, $\boldsymbol{\mu}_0$. Due to the uncertainty in the state, $\boldsymbol{\mu}_0$ is not necessarily set to the initial state, $\mathbf{x}_0$. Instead, the tube-based formulation permits selecting $\boldsymbol{\mu}_0$ such that

$$\mathbf{x}_0 \in \boldsymbol{\mu}_0 \oplus \mathit{Box}(\boldsymbol{\delta}_0^x) \tag{17}$$

where $\mathit{Box}(\boldsymbol{\delta}_0^x)$ is the bounding box with dimensions given by $\boldsymbol{\delta}_0^x$ [22] and $\oplus$ denotes the Minkowski sum. We therefore propose a piecewise definition of $\boldsymbol{\mu}_0$,

$$\boldsymbol{\mu}_0 = \begin{cases} \mathbf{x}_0, & \mathbf{x}_0 \in \tilde{\mathcal{X}}_0 \\ \operatorname{proj}_{\tilde{\mathcal{X}}}(\mathbf{x}_0), & \mathbf{x}_0 \in \mathcal{X}_0 \backslash \tilde{\mathcal{X}}_0 \end{cases} \tag{18}$$

where the $\operatorname{proj}_{\tilde{\mathcal{X}}}(\cdot)$ operator projects the state onto the tightened constraint set, $\tilde{\mathcal{X}}$. If $\mathbf{x}_0 \in \tilde{\mathcal{X}}_0$, the initial state satisfies (17) and can be assigned to $\boldsymbol{\mu}_0$. Otherwise, we assume only the noisy state is outside $\tilde{\mathcal{X}}_0$ and use the projection operation to find the closest point in $\tilde{\mathcal{X}}_0$. Due to the chance-constrained formulation, infrequent constraint violations are possible. Therefore, if $\mathbf{x}_0 \notin \mathcal{X}_0$, an intermediate controller is applied as part of the Robust EPC algorithm detailed in Sect. II-E to recover from the constraint violation.

D. Online Model Adaptation

In addition to robust constraint satisfaction, the parameterized controllers (14) generated via Robust EPC retain the adaptation properties of EPC, thus providing a means to mitigate both high and low frequency sources of uncertainty. While EPC employs Locally Weighted Projection Regression (LWPR) to construct and update a database of local dynamics models, we consider three online model adaptation strategies to assess their effects on robust constraint satisfaction.

LWPR Model Learner: LWPR learns corrections to a nominal dynamics model via a Gaussian-weighted combination of local linear functions that are updated incrementally via partial least squares [33]. Therefore, given a state-control pair, $(\mathbf{x}, \mathbf{u})$, LWPR returns the anticipated error, $\tilde{\mathbf{c}}$ between the predicted and actual next state.

ISSGPR Model Learner: Incremental Sparse Spectrum Gaussian Process Regression (ISSGPR) [12] projects input data onto a set of trigonometric basis functions with random frequencies. Regularized linear regression in this feature space yields the predictive mean. While standard Gaussian process regression run time is cubic in the amount of data, ISSGPR achieves constant time by using an explicit feature space to avoid the expensive computation of the Gramian matrix.

Luenberger Disturbance Observer: Finally, we also consider a purely reactive adaptation strategy based on $\mathcal{L}_1$ adaptive control [34]. This approach employs a nonlinear Luenberger observer driven by the difference between the state predicted via (1) and the state reported by the state estimator.

E. Algorithm Overview

The Robust EPC algorithm leverages this formulation to achieve high-rate adaptive control while providing robust constraint satisfaction, as illustrated in Fig. 1 and detailed in Alg. 1. We incrementally construct an experience database, $\mathcal{M}$, as a mapping from experiences, $(\mathbf{x}, \mathbf{r}, \mathbf{u}, \mathbf{A}, \mathbf{B}, \tilde{\mathbf{c}}, \boldsymbol{\delta}^{\mathbf{x}}, \boldsymbol{\delta}^{\mathbf{u}})$, to controllers, $(\mathbf{K}, \mathbf{k}_{\text{ff}})$, that can be queried in future control iterations to recover the functionality of (10). In every control iteration, Robust EPC obtains the current state estimate, $\mathbf{x}_0$, reference sequence, $\mathbf{r}_1, \ldots, \mathbf{r}_N$, and dynamics model $(\mathbf{A}, \mathbf{B}, \tilde{\mathbf{c}})$ updated via adaptation. It also computes the robustness bounds, $\boldsymbol{\delta}^{\mathbf{x}}$ and $\boldsymbol{\delta}^{\mathbf{u}}$, via the current state estimate covariance and the ancillary controller gains, and sets the initial state, $\boldsymbol{\mu}_0$, according to (18). The algorithm then searches $\mathcal{M}$ and assesses the optimality of each element via the parameterized KKT conditions (line 8). If any element meets the optimality criteria, the search terminates and the corresponding parameterized controller is augmented with the ancillary controller (14) and applied (as in Fig. 1a). This implies that the current situation is only required to match the active set for the experience entry, not the entire tuple, $(\mathbf{x}, \mathbf{r}, \mathbf{u}, \mathbf{A}, \mathbf{B}, \tilde{\mathbf{c}}, \boldsymbol{\delta}^{\mathbf{x}}, \boldsymbol{\delta}^{\mathbf{u}})$. Thus we can simply store sets of active constraints in the database.

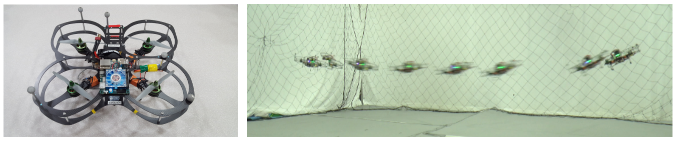
Algorithm 1 Robust Experience-driven Predictive Control

```
1:  $\mathcal{M} \leftarrow \emptyset$ or $\mathcal{M}_{\text{prior}}$
2:  while control is enabled do
3:      $\mathbf{x}_0 \leftarrow$ current system state estimate mean
4:      $\mathbf{r}_1, \ldots, \mathbf{r}_N \leftarrow$ current reference sequence
5:      $\mathbf{A}, \mathbf{B}, \tilde{\mathbf{c}} \leftarrow$ current dynamics model via adaptation
6:      Compute $\mathbf{S}$ via (16) and $\boldsymbol{\delta}^{\mathbf{x}}, \boldsymbol{\delta}^{\mathbf{u}}$ via (7),(8)
7:      Select $\boldsymbol{\mu}_0$ via (18)
8:      for each element $m_i \in \mathcal{M}$ do
9:          Compute $\boldsymbol{u}, \boldsymbol{\lambda}$ via (13),(15)
10:         if $\boldsymbol{x}, \boldsymbol{r}$ satisfy parameterized KKT criteria then
11:             importance$_i$ $\leftarrow$ current time, sort $\mathcal{M}$
12:             solution_found $\leftarrow$ true
13:             Apply control law (14) from $m_i$
14:         end if
15:     end for
16:     if solution_found is false then
17:         Apply interm. control via (11) with slack variables
18:         Update QP formulation with $(\mathbf{A}, \mathbf{B}, \tilde{\mathbf{c}}, \boldsymbol{\delta}^{\mathbf{x}}, \boldsymbol{\delta}^{\mathbf{u}})$
19:         Generate new controller via QP (11) (in parallel)
20:         if $|\mathcal{M}| =$ maximum table size then
21:             Remove element with min. importance
22:         end if
23:         Add $m_{\text{new}} = (\mathbf{K}, \mathbf{k}_{\text{ff}},$ importance$)$ to $\mathcal{M}$
24:     end if
25: end while
```

As in Fig. 1b, if no element satisfies the KKT conditions (line 16), a new element is computed via (11) and added to $\mathcal{M}$ to extend the stored experiences to include the current scenario. To avoid blocking the control loop during this computation, a short-horizon intermediate MPC with slack on state constraints (line 17) is applied in parallel. The short horizon is selected to achieve the required control rate at the expense of degraded performance, while the slack constraints ensure feasibility even in the presence of constraint violations. Robust EPC also bounds search time by limiting the size of $\mathcal{M}$. Each element is given an `importance` score based on how recently it was used, and $\mathcal{M}$ is sorted in order of decreasing `importance`. When a new element is added, the element of $\mathcal{M}$ with the minimum `importance` may be removed to maintain the size limit (line 21). As this algorithm runs, $\mathcal{M}$ will be populated with the appropriate controllers for the current situation, thereby reducing the dependence on the intermediate controller. Due to the parameterized form of the controller gains (13) and KKT matrices (15), the elements of $\mathcal{M}$ also automatically adapt to changes in the dynamics model and robustness bounds, thus maintaining robust constraint satisfaction via controller switching. Finally, we note that switching controllers within the database preserves stability as it is analogous to explicit MPC techniques [13], while transitions to and from the intermediate controller will preserve stability if they are sufficiently infrequent [6, 14].

III. Results

To assess the performance of the proposed Robust EPC algorithm, we aim to demonstrate the following results through a series of flight experiments: **R1:** stable control performance, **R2:** real-time computation of control commands, **R3:** constraint satisfaction in the presence of time-varying sensor



(a) (b)

Fig. 3: (a) The quadrotor and ODROID-XU4 used for experimental validation. (b) Snapshots from the linear trajectory.

uncertainty (i.e., robust constraint satisfaction), **R4:** improved trajectory tracking performance while satisfying constraints, and **R5:** robust constraint satisfaction during aggressive flight.

A. Experimental Setup

The experimental platform is a small, 790 g quadrotor aerial robot equipped with an ODROID-XU4 (2 GHz ARM processor with 2 GB RAM), as shown in Fig. 3a. All control algorithms are implemented in C++ via ROS [29] and run in real-time on the ODROID. We employ a cascaded setup with a translational controller providing references for an attitude controller that, in turn, provides actuator commands to the motor controllers [25].

A motion capture arena provides position and heading feedback that is combined with IMU measurements using an onboard state estimator. Due to the low variance in the motion capture feedback, we inject Gaussian noise with changing variance into the motion capture data to emulate a lower-quality sensor that exhibits changes in performance as a function of the environment (e.g., a vision-based sensor transitioning between feature-rich and feature-sparse regions). The changing uncertainty in the motion capture data is also broadcast to the state estimator and Robust EPC to inform belief state propagation via the measurement covariance term in (1).

For these experiments, we consider the problem of controlling the translational dynamics of the quadrotor [25], subject to velocity and control constraints. This yields an MPC formulation with $n = 6$ states and $m = 3$ inputs. We also consider a horizon of $N = 25$ steps at the control update rate (100 Hz) for the main Robust EPC formulation. The nominal state, $\mathbf{x}^*$, is set to the current state at each control iteration, and the nominal control, $\mathbf{u}^*$, is set to hover to avoid penalizing gravity compensation. We use $\alpha = 0.001$ for a constraint satisfaction probability of 99.9%. The intermediate controller is formulated with a horizon of $N = 10$ to yield comparable solution times to Robust EPC. The cost function weight matrices are selected such that a finite-horizon LQR using either set of weights (and the corresponding horizon) would yield the same gain matrix. The proportional and derivative gains for the $\mathcal{L}_1$ adaptive controller used as a baseline also match this LQR formulation.

B. Timescale Separation with Model Adaptation

As Robust EPC extends EPC, we retain its ability to mitigate the effects of low-frequency sources of uncertainty (e.g., due to bulk fluid flow) via online model adaptation. The quadrotor is first commanded to track the linear trajectory in Fig. 3b (five laps between two waypoints about 3.6 meters apart) subject to a 6 m/s external wind that is orthogonal to the trajectory, and Fig. 4 shows the resulting cross-track errors ($\mathcal{L}_1$ adaptive control is included for reference). As expected, all three model adaptation strategies (described in Sect. II-D with parameters tuned empirically) yield low cross track error, and LWPR and ISSGPR exhibit zero-mean tracking as they accumulate experience. However, as Fig. 5 illustrates, no choice of adaptation strategy is sufficient to mitigate the effects of state uncertainty, resulting in repeated constraint violations. This further demonstrates the need for the proposed Robust EPC formulation. Moreover, as the choice of model adaptation strategy does not fundamentally change the system's ability to mitigate high-frequency source of uncertainty, we follow EPC [8] and proceed with LWPR for the following experimental studies.

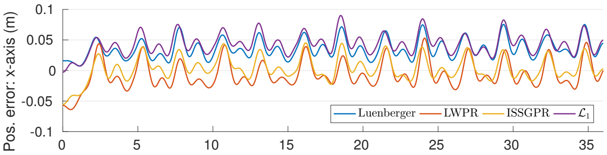


Fig. 4: Cross-track error for EPC with the three model adaptation strategies discussed and $\mathcal{L}_1$ adaptive control. LWPR and ISSGPR yield superior tracking after gaining experience. To better visualize the low-frequency components, we apply exponential smoothing with a two-second window.

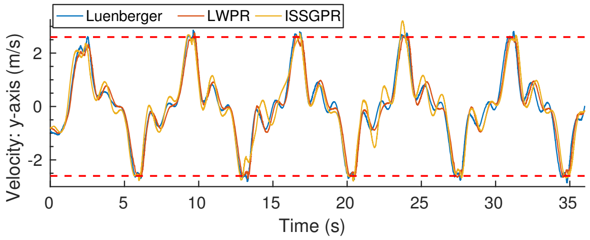


Fig. 5: With injected noise, (non-robust) EPC yields velocity constraint violations for all three model adaptation strategies.

C. Robust Constraint Satisfaction

We first evaluate Robust EPC's trajectory tracking performance along the linear trajectory in Fig. 3b. Figure 6 shows that Robust EPC stabilizes the system to track the trajectory, which achieves a maximum linear velocity of 2.7 m/s (**R1**). Table I shows the compute times for the different components of Robust EPC from one representative trial. This demonstrates that both the Query and Intermediate controller components, which constitute the primary control thread, run in real-time on the computationally constrained flight hardware (**R2**). In contrast, the variance in solving the QP may yield control iterations that violate the 100 Hz update rate, making traditional optimization-based Robust MPC approaches infeasible.

To show robust constraint satisfaction in the presence of time-varying sensor uncertainty, we inject zero-mean Gaussian noise with a standard deviation of 0.03 into the motion capture data when the y-axis position of the vehicle is between -0.5 m and 0.5 m. This makes satisfaction of the velocity constraints

TABLE I: Compute times for Robust EPC components with the number of control iterations statistics are computed across.

	Query	Interm.	QP	Add Element
Iterations	5949	18	12	12
Mean (ms)	1.089	1.303	4.427	4.891
Std. Dev. (ms)	1.463	0.886	2.720	5.393

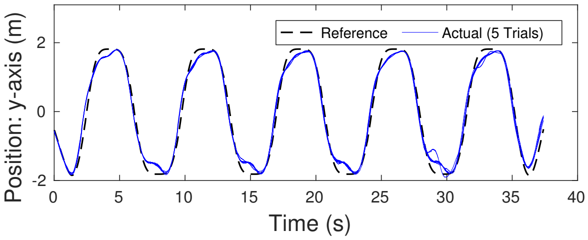


Fig. 6: Linear trajectory tracking performance across five laps

particularly difficult as the vehicle also attains its maximum speeds in this region.

In addition to Robust EPC, we consider three baseline control strategies: $\mathcal{L}_1$ adaptive control (a reactive approach), EPC, and a Robust MPC (R-MPC) formulation that solves the QP online with $N = 10$ (the reduced horizon is required to achieve comparable solution times to EPC) and slack on state constraints (to ensure problem feasibility). Figure 8 shows the resulting velocity profiles with the constraint bounds shown by the dashed lines. $\mathcal{L}_1$ adaptive control shows unconstrained control performance, which naturally violates the constraints as the reference velocity has a maximum of 2.7 m/s. The enforcement of constraints in EPC yields smaller constraint violations, but the non-robust formulation of the constraints fails to mitigate the effects of measurement uncertainty. R-MPC also exhibits substantial constraint violations. To confirm that the degraded performance of R-MPC is due to the short horizon and not the slack constraints, we also compared performance of R-MPC and Robust EPC with $N = 25$ using a high-fidelity simulator on a more powerful computer and observed comparable performance and robust constraint satisfaction (not shown). Therefore, these results illustrate that over repeat trials, only Robust EPC consistently satisfies the velocity constraints (**R3**).

Figure 7 illustrates controller generation and reuse as indicated by the amount of time each controller is applied. Note that the intermediate controller (index 1) is only used in the first few laps, while controller 2 is applied frequently as it corresponds to nominal operation, i.e., away from all constraints. This indicates that over time, all of the controllers needed to track the trajectory and satisfy constraints are enumerated and available for use in the experience database.

D. Time-Varying Uncertainty Prediction

To show that Robust EPC leverages the Gaussian nature of the state estimator output and exploits regions of low uncertainty to improve performance over more conservative approaches, we investigate its performance compared to an instantiation of Robust EPC that uses a fixed upper bound

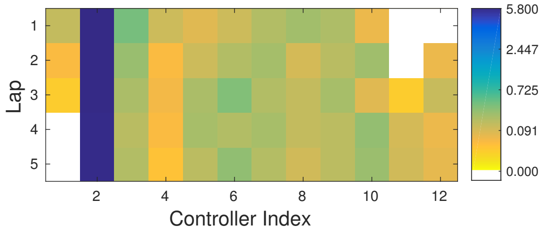
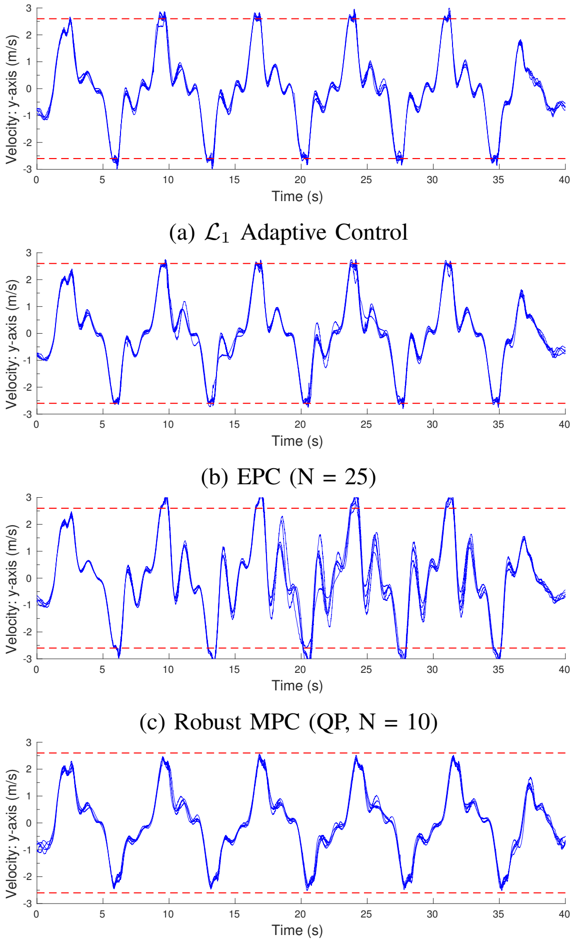


Fig. 7: Time spent using each controller per lap. Note that multiple controllers are learned and reused and that the intermediate controller (index 1) ceases to be used past lap 3.



(a) $\mathcal{L}_1$ Adaptive Control

(b) EPC (N = 25)

(c) Robust MPC (QP, N = 10)

(d) Robust EPC (N = 25)

Fig. 8: Comparison of y-velocity profiles for the line trajectory across 5 trials of each controller. Only Robust EPC satisfies the nominal velocity constraints (dashed lines).

on the uncertainty. We take the maximum bound applied by Robust EPC during a run of the trajectory as the uncertainty value for this fixed bound approach. The quadrotor is commanded to track a vertical circle trajectory while Gaussian noise with a standard deviation of 0.03 is injected when the vehicle is below one meter in height. Figure 10 shows tracking results for Robust EPC using three approaches: Gaussian belief propagation, the fixed bound approach using the true upper

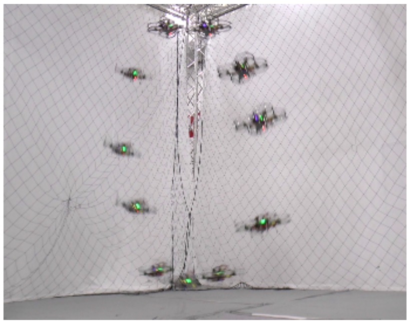


Fig. 9: Circle trajectory used to evaluate belief propagation.

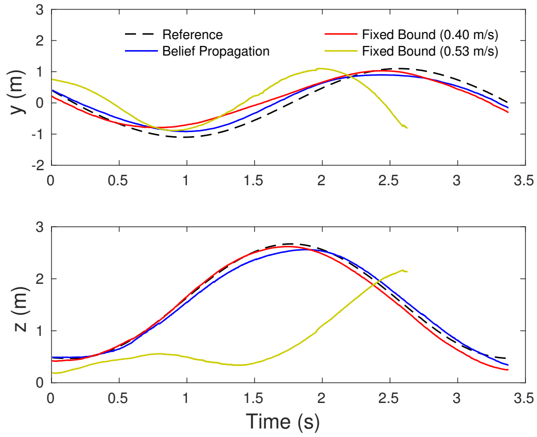


Fig. 10: Position along the y and z axes for Robust EPC and the fixed bound approach as compared to the reference trajectory. The fixed bound approach that uses the true upper bound (0.53 m/s) fails to track the trajectory. The mean and max error for Robust EPC along the y-axis are 0.22 and 0.41, respectively, while for the successful fixed bound approach (0.40 m/s), the mean and max error are 0.24 and 0.51.

bound as described above, and the fixed bound approach using the highest bound that allows for stable trajectory tracking. The fixed bound approach is unable to complete the trajectory with the true upper bound, and Robust EPC yields reduced tracking error compared to the less conservative fixed bound approach. This confirms that Robust EPC exploits the low noise region above one meter and achieves better performance than the conservative approaches (**R4**). We also consider uncertainty propagation via recursive application of the Pontryagin difference with the uncertainty set [30]. However, even with the ancillary controller, this results in an infeasible problem for the longer horizons permitted by Robust EPC. Figure 11 illustrates the tube growth with a 25-step horizon for the two approaches.

E. Implications for Aggressive Flight

To further assess the performance of Robust EPC, we test constraint satisfaction on a high speed linear trajectory with a maximum velocity of 3.6 m/s. As Fig. 12 shows, Robust EPC satisfies velocity constraints throughout the trial with the exception of a 0.03 m/s violation during the final lap. Due to the chance-constrained formulation, there is a nonzero probability of constraint violation (0.1% in our experiments). In addition, higher speeds accentuate the effects of modeling errors and may yield degraded performance if the model adaptation is not sufficiently fast. As a result, we conclude that Robust EPC reliably enforces constraints even during aggressive motion (**R5**).

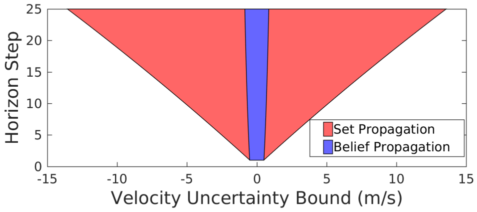


Fig. 11: Overlay of tube growth for Set Propagation and Belief Propagation based on the bounds computed by each at the start of trajectory tracking. Set Propagation growth is too fast to yield feasible constraints.

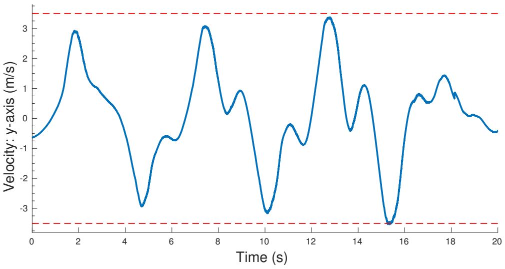


Fig. 12: Velocity of Robust EPC along a high-speed linear trajectory. There is a small constraint violation of 0.03 m/s during the last lap.

IV. Conclusions and Future Work

In this work, we present an extension to Experience-driven Predictive Control (EPC) that allows for robust constraint satisfaction in the presence of time- and state-dependent uncertainty. We have shown that the proposed approach, Robust EPC, successfully stabilizes the vehicle along a variety of trajectories (**R1**), easily meets computational requirements on a compute-constrained system (**R2**), reliably satisfies constraints in the presence of time-varying sensor uncertainty (**R3**) while improving tracking performance as compared to conservative methods (**R4**), and maintains constraint satisfaction properties during aggressive operation (**R5**). In the future, we intend to integrate this robust control strategy with a visual odometry system that introduces more realistic state uncertainty. Also, the current Robust EPC formulation assumes that the uncertainty is well-modeled by a Gaussian distribution, but incorporating distributions without closed-form propagation models is another avenue for future study. This formulation can also be extended to account for additional sources of uncertainty, including communication latency and the variance in model adaptation techniques such as LWPR and ISSGPR.

Acknowledgments

We gratefully acknowledge the support of ARL Grant W911NF-08-2-0004.

References

[1] V. Adetola and M. Guay. Robust adaptive MPC for constrained uncertain nonlinear systems. *Intl. J. of Adaptive Control & Signal Process.*, 25(2):155–167, 2011.

[2] K. Alexis, C. Papachristos, R. Siegwart, and A. Tzes. Robust Model Predictive Flight Control of Unmanned Rotorcrafts. *J. Intl. & Robot. Syst.*, 81:443–469, 2016.

[3] A. Bemporad, V. Dua M. Morari, and E. N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38:3–20, 2002.

[4] P. Bouffard, A. Aswani, and C. Tomlin. Learning-based model predictive control on a quadrotor: Onboard implementation and experimental results. In *Proc. of the IEEE Intl. Conf. on Robot. and Autom.*, St. Paul, MN, May 2012.

[5] C. Conte, M. N. Zeilinger, M. Morari, and C. N. Jones. Robust distributed model predictive control of linear systems. In *Euro. Control Conf.*, page 5, July 2013.

[6] V. Desaraju. *Safe, Efficient, and Robust Predictive Control of Constrained Nonlinear Systems*. PhD thesis, Carnegie Mellon University, 2017.

[7] V. Desaraju and N. Michael. Fast Nonlinear Model Predictive Control via Partial Enumeration. In *Proc. of the IEEE Intl. Conf. on Robot. and Autom.*, Stockholm, Sweden, May 2016.

[8] V. Desaraju and N. Michael. Leveraging Experience for Computationally Efficient Adaptive Nonlinear Model Predictive Control. In *Proc. of the IEEE Intl. Conf. on Robot. and Autom.*, Singapore, May 2017.

[9] F. Domes and A. Neumaier. Rigorous Enclosures of Ellipsoids and Directed Cholesky Factorizations. *SIAM J. Matrix Anal. Appl.*, 32:62285, 2011.

[10] M. Farrokhsiar, G. Pavlik, and H. Najjaran. An integrated robust probing motion planning and control scheme: A tube-based MPC approach. *Robot. Auton. Syst.*, 61(12):1379–1391, 2013.

[11] H. Fukushima, T.H. Kim, and T. Sugie. Adaptive model predictive control for a class of constrained linear systems based on the comparison model. *Automatica*, 43(2):301–308, 2007.

[12] A. Gijsberts and G. Metta. Real-time model learning using Incremental Sparse Spectrum Gaussian Process Regression. *Neural Networks*, 41:59–69, 2013.

[13] A. Grancharova and T.A. Johansen. *Explicit Nonlinear Model Predictive Control*, volume 429. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[14] J. P. Hespanha and A. S. Morse. Stability of switched systems with average dwell-time. In *Proc. of the IEEE Conf. on Decision and Control*, volume 3, pages 2655–2660, Phoenix, AZ, December 1999.

[15] M. Hofer, M. Muehlebach, and R. DAndrea. Application of an Approximate Model Predictive Control Scheme on an Unmanned Aerial Vehicle. In *Proc. of the IEEE Intl. Conf. on Robot. and Autom.*, Stockholm, Sweden, May 2016.

[16] R. Platt Jr., R. Tedrake, L. Kaelbling, and T. Lozano-Perez. Belief space planning assuming maximum likelihood observations. In *Proc. of Robot.: Sci. and Syst.*, 2010.

[17] I. Kolmanovsky and E. G. Gilbert. Theory and computation of disturbance invariant sets for discrete-time linear systems. *Mathematical Problems in Engineering*, 4(4):317–367, 1998.

[18] Y. Kuwata, A. Richards, and J. How. Robust Receding Horizon Control using Generalized Constraint Tightening. In *Proc. of the Amer. Control Conf.*, pages 4482–4487, New York City, NY, July 2007.

[19] W. Langson, I. Chryssochoos, S. V. Raković, and D. Q. Mayne. Robust model predictive control using tubes. *Automatica*, 40 (1):125–133, January 2004.

[20] J. Löfberg. *Minimax approaches to robust model predictive control*. PhD thesis, Linköping University, 2003.

[21] D. Q. Mayne. Model predictive control: Recent developments and future promise. *Automatica*, 50:2967–2986, 2014.

[22] D. Q. Mayne and W. Langson. Robustifying model predictive control of constrained linear systems. *Automatica*, 37:1422–1423, 2001.

[23] D. Q. Mayne, M. M. Seron, and S. V. Raković. Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica*, 41(2):219–224, February 2005.

[24] D. Q. Mayne, E. C. Kerrigan, E. J. van Wyk, and P. Falugi. Tube-based robust nonlinear model predictive control. *J. Robust and Nonlin. Control*, 21:13411353, 2011.

[25] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar. Experimental evaluation of multirobot aerial control algorithms. *IEEE Robotics & Automation Magazine*, September 2010.

[26] M. Neunert, C. de Crousaz, F. Furrer, M. Kamel, F. Farshidian, R. Siegwart, and J. Buchli. Fast Nonlinear Model Predictive Control for Unified Trajectory Optimization and Tracking. In *Proc. of the IEEE Intl. Conf. on Robot. and Autom.*, Stockholm, Sweden, May 2016.

[27] C. Ostafew, A. Schoellig, and T. Barfoot. Learning-Based Nonlinear Model Predictive Control to Improve Vision-Based Mobile Robot Path-Tracking in Challenging Outdoor Environments. In *Proc. of the IEEE Intl. Conf. on Robot. and Autom.*, pages 4029–4036. IEEE, May 2014.

[28] G. Pannocchia, J. B. Rawlings, and S. J. Wright. Fast, large-scale model predictive control by partial enumeration. *Automatica*, 43(5):852–860, 2007.

[29] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng. ROS: An open-source Robot Operating System. In *ICRA Workshop on open source software*, page 5, Kobe, Japan, 2009.

[30] A. Richards. Robust Model Predictive Control for Time-Varying Systems. In *Proc. of the IEEE Conf. on Decision and Control*, pages 3747–3752, Seville, Spain, December 2005.

[31] A. Richards and J. How. Robust Model Predictive Control with Imperfect Information. In *Proc. of the Amer. Control Conf.*, New York City, NY, July 2005.

[32] E. B. Sudderth, A. T. Ihler, M. Isard, W. T. Freeman, and A. S. Willsky. Nonparametric Belief Propagation. *Comm. of the ACM*, 53:95–103, 2010.

[33] S. Vijayakumar, A. DSouza, and S. Schaal. Incremental Online Learning in High Dimensions. *Neural Comp.*, 17(12):2602–2634, 2005.

[34] J. Wang, F. Holzapfel, E. Xargay, and N. Hovakimyan. Non-Cascaded Dynamic Inversion Design for Quadrotor Position Control with L1 Augmentation. In *Proc. of the CEAS Specialist Conf. on Guidance, Navigation & Control*, Delft, Netherlands, April 2013.

[35] J. Yan and R. R. Bitmead. Incorporating state estimation into model predictive control and its application to network traffic control. *Automatica*, 41:595–604, 2005.