Journal Title
XX(X):1–18
©The Author(s) 2017
Reprints and permission:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/ToBeAssigned
www.sagepub.com/
SAGE

Leveraging Experience for Robust, Adaptive NMPC on Computationally Constrained Systems with Time-Varying State Uncertainty

Vishnu R. Desaraju, Alexander E. Spitzer, Cormac O'Meadhra, Lauren Lieu, and Nathan Michael

Abstract

This paper presents a robust, adaptive nonlinear model predictive control (NMPC) technique that leverages past experiences to achieve tractability on computationally constrained systems. We propose a robust extension of the Experience-driven Predictive Control (EPC) algorithm via a Gaussian belief propagation strategy that computes 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 trials with a simulated ground robot and three experimental platforms: 1) a small quadrotor aerial robot executing aggressive maneuvers in wind with degraded state estimates, 2) a skid-steer ground robot equipped with a laser-based localization system, and 3) a hexarotor aerial robot equipped with a vision-based localization system.

Keywords

Robust Nonlinear Model Predictive Control, Experience, Model Adaptation, Belief Propagation

1 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 (Brunner and Peynot 2010) and can lead to control actions that compromise the safety and reliability of the system (Richards and How 2005; Bouffard et al. 2012). Safe and reliable operation 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. Even larger platforms with computationally expensive sensing and localization systems may have limited resources for control (Brockers et al. 2014). 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 (Mayne 2014).

There are two general classes of approaches for mitigating the effects of uncertainty in predictive control. The first class consists of adaptive formulations that seek to estimate and mitigate the uncertainty in the dynamics model (Fukushima et al. 2007; Aswani et al. 2013; Ho et al. 2014; Tanaskovic et al. 2014). These approaches leverage online estimates of the mean perturbation to the system dynamics, e.g. from a Kalman filter (Bouffard et al. 2012) or nonparametric estimator (Ostafew et al. 2014), that improves the accuracy with which the predictive model can anticipate the system's evolution and interaction with the constraints. As a result, online adaptation can yield improved trajectory tracking accuracy and constraint satisfaction, even in the presence of severe, nonlinear perturbations (Desaraju and Michael 2017). However, in practice, these techniques may still lead to constraint violations due to the difference in timescales between the disturbance estimator and high-frequency noise in the state estimate (Desaraju 2017).

In contrast, the second class of techniques consists of robust formulations that refine the constraints to explicitly account for high-frequency uncertainty. Robust MPC techniques provide constraint satisfaction guarantees in

Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA

Corresponding author:
Vishnu R. Desaraju, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA
Email: vdesaraj@alumni.cmu.edu

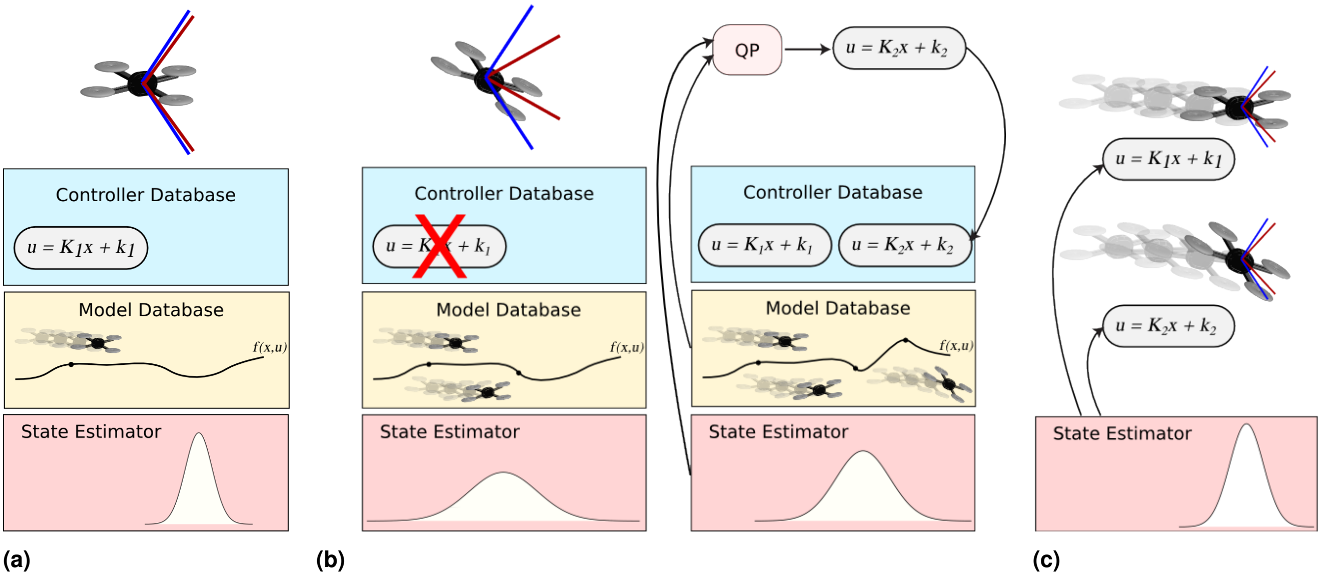


Figure 1. Overview of the proposed approach that combines an online learned controller database with estimates of the dynamics model and state uncertainty. As uncertainty changes, the tightened constraints (red) on the MAV automatically adjust to ensure robust satisfaction of the requested constraints (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.

the presence of bounded, uncertain parameters (Kothare et al. 1996; Mayne and Langson 2001; Löfberg 2003; Langson et al. 2004; Richards 2005). For linear dynamics, the effects of bounded uncertainty can be represented by disturbance-invariant sets (Kolmanovsky and Gilbert 1998) that can be used to tighten the set of feasible states and inputs (e.g., via the Pontryagin difference operation) (Mayne et al. 2005). In the nonlinear case, this can be generalized to min-max formulations to optimize with respect to the maximal state deviations (Adetola and Guay 2011). 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 (Kuwata et al. 2007; Richards 2005; Mayne et al. 2005). 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 (Richards and How 2005). An extension of this idea replaces the deterministic uncertainty set with a probabilistic representation, e.g., as a multivariate Gaussian distribution (Yan and Bitmead 2005). This representation 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 (Mayne 2014), thus mirroring the belief propagation techniques applied in belief-space planning (Platt et al. 2010). Some of these belief-space approaches employ a chance constrained formulation to transform the probabilistic formulation into a deterministic one based on a tunable parameter that controls allowable risk (Toit and Burdick 2010; Bry and Roy 2011).

Many adaptive MPC formulations also include a robust component that is coupled to estimator uncertainty (Fukushima et al. 2007; Adetola and Guay 2011; Farrokhsiar et al. 2013). 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 (Ostafew et al. 2016).

However, these robust-adaptive techniques may not be tractable on computationally constrained platforms. Therefore, in addition to constraint satisfaction under uncertainty, we require the ability to compute predictive control commands at sufficiently high rates to ensure stability of these resource constrained and often highly agile systems. Fast MPC solution strategies can be divided into four categories: leveraging fast online optimization techniques (Houska et al. 2011; Neunert et al. 2016), optimizing approximate formulations (Hofer et al. 2016), explicit enumeration of equivalent controllers (Alexis et al. 2016; Domahidi et al. 2011), and semi-explicit approaches (Pannocchia et al. 2007; Zeilinger et al. 2014; Desaraju and Michael 2016, 2017). 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 (Pannocchia et al. 2007). Within the class of semi-explicit approaches, the Experience-driven Predictive Control (EPC) algorithm (Desaraju and Michael 2017) extends this efficient formulation to mitigate the effects of low-frequency disturbances to the system dynamics.

Therefore, in this paper, 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 encapsulates the two core contributions of this work:

1) Computationally efficient solutions to MPC via a semi-explicit solution strategy for nonlinear systems with uncertain state estimates and dynamics models.

2) Probabilistic constraint satisfaction in the presence of time-varying state uncertainty modeled by a multivariate Gaussian distribution, e.g., provided by a Kalman filter based state estimator.

Section 2 details the Robust EPC algorithm, including the chance-constrained Tube MPC formulation, modifications to (non-robust) EPC, and model adaptation strategies. In Sect. 3, we present a set of simulation and experimental studies that demonstrate the following key results:

- Stable control performance
- Real-time computation of control commands
- Experience reuse to reduce online computation
- Constraint satisfaction in the presence of time-varying sensor uncertainty
- Improved trajectory tracking performance while satisfying constraints
- Robust-adaptive constraint satisfaction in a challenging environment

This manuscript refines an earlier conference presentation of the Robust EPC algorithm (Desaraju et al. 2017) including a more detailed discussion of the model adaptation techniques in Sect. 2.4. We also present new simulation results with a ground robot (Sect. 3.1), and several new experimental studies with three different platforms: a skid-steer ground robot equipped with a laser-based localization system (Sect. 3.3), a hexarotor aerial robot equipped with a vision-based localization system (Sect. 3.4), and additional results with a small quadrotor aerial robot executing aggressive maneuvers in a strong, spatially varying wind field (Sect. 3.2.4).

2 Approach

In this section, we present an extension of the Experience-driven Predictive Control (EPC) algorithm (Desaraju and Michael 2017) 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.

Notation: For consistency throughout the presentation, we reuse letters with different formatting to denote related variables (e.g., $\boldsymbol{x}$ is a concatenated vector of $\mathbf{x}_i$). The formatting indicates different variable types and is summarized below:

Variable Type	Examples
Scalar:	x, r, u, N
Vector:	$\mathbf{x}, \mathbf{r}, \mathbf{u}, \mathbf{c}, \mathbf{g}_\mathbf{x}$
Concatenated Vector:	$\boldsymbol{x}, \boldsymbol{r}, \boldsymbol{u}, \boldsymbol{c}, \boldsymbol{g}_\mathbf{x}$
Matrix:	$\mathbf{A}, \mathbf{B}, \mathbf{R}, \mathbf{G}_\mathbf{x}, \mathbf{M}$
Concatenated Matrix:	$\boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}}, \boldsymbol{\mathcal{R}}, \boldsymbol{\mathcal{G}}_\mathbf{x}, \boldsymbol{\mathcal{M}}$
Set:	$\mathcal{X}_k, \mathcal{U}_k$
Function:	$J(\cdot), f(\cdot), \kappa(\cdot)$

2.1 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 sum of the constant term in the Taylor series approximation, $f(\mathbf{x}^*, \mathbf{u}^*)$, and the predicted model error, $\hat{\boldsymbol{\rho}}$. Updating $\hat{\boldsymbol{\rho}}$ via an online model learning strategy captures both the linearization error and the effects of time-varying, unmodeled dynamics, thus enabling adaptation to external perturbations (detailed in Sect. 2.4).

To model the evolution of this uncertain system, we leverage the existence of a closed-form belief propagation law for Gaussian distributions (Sudderth et al. 2010) 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. (2010), we take $\mathbf{z}_{k+1} = h(\boldsymbol{\mu}_k)$ as the maximum likelihood observation to obtain a simplified belief state update law

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

2.2 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 (Mayne et al. 2011) given in Sect. 2.3.1. The introduction of the ancillary controller restricts deviations from the predicted state mean (Mayne 2014) 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,

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

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

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

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 (Mayne 2014), as illustrated in Fig. 2. Consequently, to retain the linear form of the constraints, we follow Domes and Neumaier (2011) 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, $\mathbf{x}_k \in \tilde{\mathcal{X}}_k, \mathbf{u}_k \in \tilde{\mathcal{U}}_k$,

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

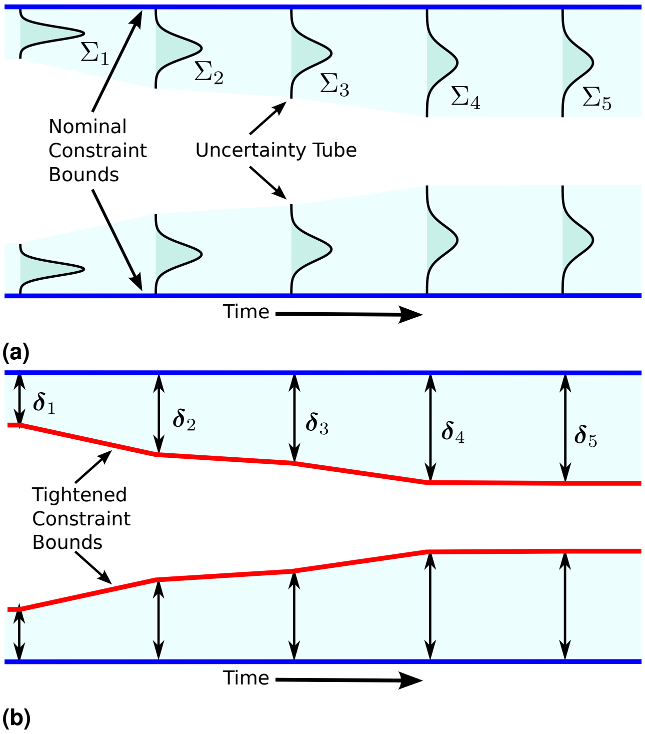


Figure 2. Two-dimensional illustration of constraint tightening: (a) nominal state constraints (blue line) with the predicted Gaussian uncertainty tube ($\boldsymbol{\Sigma}_i$) define (b) chance-constraint bounds ($\boldsymbol{\delta}_i$) that yield tightened constraints (red).

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 (Conte et al. 2013).

2.3 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 (Desaraju and Michael 2017) 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.

The underlying NMPC problem can be formulated as a nonlinear program (NLP) that computes the control sequence, $\mathbf{u}_0, \ldots, \mathbf{u}_{N-1}$, given the current state, $\mathbf{x}_0$, and N reference states, $\mathbf{r}_1, \ldots, \mathbf{r}_N$, (e.g., from a desired trajectory). We also enforce the dynamics model (1) and constraints given by

$$g(\mathbf{x}_{k+1}, \mathbf{u}_k) \le 0$$

The resulting NLP is given by

$$\begin{aligned} \underset{\mathbf{u}_0,\ldots,\mathbf{u}_{N-1}}{\operatorname{argmin}} \quad & \sum_{k=0}^{N-1} J(\mathbf{x}_{k+1}, \mathbf{r}_{k+1}, \mathbf{u}_k) \\ \text{s.t.} \quad & \dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) \\ & g(\mathbf{x}_{k+1}, \mathbf{u}_k) \le 0 \;\forall k = 0, \ldots, N-1 \end{aligned} \tag{10}$$

where the cost function, $J(\cdot)$, is selected to penalize tracking error and extraneous control effort, and the differential equation constraint is enforced via numerical integration.

As in EPC, we can leverage (2) to reformulate (10) as a quadratic program (QP) where the nonlinear dynamics and constraints are re-linearized in every control iteration about a nominal state, $\mathbf{x}^*$ and input, $\mathbf{u}^*$. The model adaptation term, $\tilde{\mathbf{c}}$, captures the linearization error and other unmodeled dynamics. The resulting QP is given by

$$\begin{aligned}\underset{\bar{\mathbf{u}}_k}{\operatorname{argmin}} \sum_{k=0}^{N-1} &\frac{1}{2}(\bar{\mathbf{x}}_{k+1}-\bar{\mathbf{r}}_{k+1})^{\mathrm{T}}\mathbf{Q}(\bar{\mathbf{x}}_{k+1}-\bar{\mathbf{r}}_{k+1}) \\ &+\frac{1}{2}(\bar{\mathbf{u}}_k-\bar{\mathbf{u}}_{\mathbf{c}})^{\mathrm{T}}\mathbf{R}(\bar{\mathbf{u}}_k-\bar{\mathbf{u}}_{\mathbf{c}}) \\ \text{s.t.}\quad &\bar{\mathbf{x}}_{k+1}=\mathbf{A}\bar{\mathbf{x}}_k+\mathbf{B}\bar{\mathbf{u}}_k+\tilde{\mathbf{c}} \\ &\mathbf{G}_{\mathbf{x}}\bar{\mathbf{x}}_{k+1}\le\tilde{\mathbf{g}}_{\mathbf{x}},\quad \mathbf{G}_{\mathbf{u}}\bar{\mathbf{u}}_k\le\tilde{\mathbf{g}}_{\mathbf{u}} \\ &\forall\, k=0,\dots,N-1\end{aligned}\tag{11}$$

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 (Desaraju and Michael 2017).

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

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

$\boldsymbol{\mathcal{Q}}=\operatorname{diag}(\mathbf{Q},\dots,\mathbf{Q})$, $\boldsymbol{\mathcal{R}}=\operatorname{diag}(\mathbf{R},\dots,\mathbf{R})$, $\boldsymbol{\mathcal{G}}_{\mathbf{x}}=\operatorname{diag}(\mathbf{G}_{\mathbf{x}},\dots,\mathbf{G}_{\mathbf{x}})$, 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. 2.3.2) 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 (11) 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,

$$\begin{aligned}\underset{\boldsymbol{u}}{\operatorname{argmin}}\ &\frac{1}{2}\boldsymbol{u}^{\mathrm{T}}\boldsymbol{\mathcal{H}}\boldsymbol{u}+\boldsymbol{h}^{\mathrm{T}}\boldsymbol{u}\\ \text{s.t.}\ &\boldsymbol{\Gamma}\boldsymbol{u}\le\boldsymbol{\gamma}\end{aligned}\tag{12}$$

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,

$$\begin{aligned}\boldsymbol{\mathcal{H}}\boldsymbol{u}+\boldsymbol{h}+\boldsymbol{\Gamma}^{\mathrm{T}}\boldsymbol{\lambda}&=\mathbf{0}\\ \boldsymbol{\Lambda}(\boldsymbol{\Gamma}\boldsymbol{u}-\boldsymbol{\gamma})&=\mathbf{0}\end{aligned}\tag{13}$$

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 (13),

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

where the subscript a denotes rows corresponding to active constraints. For any linearly independent set of active constraints (Bemporad et al. 2002), the resulting $\boldsymbol{u}$ is affine in the predicted state mean error, $\boldsymbol{r}$,

$$\boldsymbol{u}=\boldsymbol{\mathcal{E}}_5\boldsymbol{r}-\left(\boldsymbol{\mathcal{E}}_5\boldsymbol{c}-\boldsymbol{\mathcal{E}}_4\boldsymbol{\mathcal{R}}\boldsymbol{u}_{\mathbf{c}}+\boldsymbol{\mathcal{E}}_3\begin{bmatrix}\boldsymbol{g}_{\mathbf{x}}^{+}-\boldsymbol{\mathcal{G}}_{\mathbf{x}}\boldsymbol{c}\\ -\boldsymbol{g}_{\mathbf{x}}^{-}+\boldsymbol{\mathcal{G}}_{\mathbf{x}}\boldsymbol{c}\\ \boldsymbol{g}_{\mathbf{u}}^{+}\\ -\boldsymbol{g}_{\mathbf{u}}^{-}\end{bmatrix}_a\right)\tag{14}$$

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 (14) 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, $\mathbf{S}$,

$$\begin{aligned}\boldsymbol{\kappa}(\mathbf{x}_0,\mathbf{r}_1,\dots,\mathbf{r}_N)&=\mathbf{K}(\mathbf{A},\mathbf{B},\tilde{\mathbf{c}},\boldsymbol{\delta}^{\mathbf{x}},\boldsymbol{\delta}^{\mathbf{u}})\boldsymbol{r}\\ &+\mathbf{k}_{\mathrm{ff}}(\mathbf{A},\mathbf{B},\tilde{\mathbf{c}},\boldsymbol{\delta}^{\mathbf{x}},\boldsymbol{\delta}^{\mathbf{u}})\\ &+\left[\mathbf{S}_0(\mathbf{x}_0-\boldsymbol{\mu}_0)^{\mathrm{T}},\dots,\mathbf{S}_{N-1}(\mathbf{x}_{N-1}-\boldsymbol{\mu}_{N-1})^{\mathrm{T}}\right]^{\mathrm{T}}\end{aligned}\tag{15}$$

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

$$\boldsymbol{\lambda}_a=-\boldsymbol{\mathcal{E}}_6\boldsymbol{r}+\left(\boldsymbol{\mathcal{E}}_6\boldsymbol{c}-\boldsymbol{\mathcal{E}}_3^{\mathrm{T}}\boldsymbol{\mathcal{R}}\boldsymbol{u}_{\mathbf{c}}+\boldsymbol{\mathcal{E}}_2\begin{bmatrix}\boldsymbol{g}_{\mathbf{x}}^{+}-\boldsymbol{\mathcal{G}}_{\mathbf{x}}\boldsymbol{c}\\ -\boldsymbol{g}_{\mathbf{x}}^{-}+\boldsymbol{\mathcal{G}}_{\mathbf{x}}\boldsymbol{c}\\ \boldsymbol{g}_{\mathbf{u}}^{+}\\ -\boldsymbol{g}_{\mathbf{u}}^{-}\end{bmatrix}_a\right)\tag{16}$$

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 (14), (16), 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 (11) by switching between controllers according to the KKT conditions, thus providing the foundation for the Robust EPC algorithm detailed in Sect. 2.5.

2.3.1 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 (12). 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 (15) without constraints,

$$\operatorname{diag}(\mathbf{S}_0, \dots, \mathbf{S}_{N-1}) = (\boldsymbol{\mathcal{B}}^{\mathrm{T}}\boldsymbol{\mathcal{Q}}\boldsymbol{\mathcal{B}} + \boldsymbol{\mathcal{R}})^{-1}\boldsymbol{\mathcal{B}}^{\mathrm{T}}\boldsymbol{\mathcal{Q}} \tag{17}$$

2.3.2 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

$$\mathbf{x}_0 \in \boldsymbol{\mu}_0 \oplus \mathit{Box}(\boldsymbol{\delta}_0^x) \tag{18}$$

where $\mathit{Box}(\boldsymbol{\delta}_0^x)$ is the bounding box with dimensions given by $\boldsymbol{\delta}_0^x$ (Mayne and Langson 2001) and $\oplus$ denotes the Minkowski sum. We therefore propose a piecewise definition of $\boldsymbol{\mu}_0$,

$$\boldsymbol{\mu}_0 = \begin{cases} \mathbf{x}_0, & \mathbf{x}_0 \in \tilde{\mathcal{X}}_0 \\ \operatorname{proj}_{\tilde{\mathcal{X}}}(\mathbf{x}_0), & \mathbf{x}_0 \in \mathcal{X}_0 \backslash \tilde{\mathcal{X}}_0 \end{cases} \tag{19}$$

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 (18) 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. 2.5 to recover from the constraint violation.

2.4 Online Model Adaptation

In addition to robust constraint satisfaction, the parameterized controllers (15) 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 (Vijayakumar et al. 2005) 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.

2.4.1 LWPR Model Learner: We apply Locally Weighted Projection Regression (LWPR) to learn corrections to a nominal dynamics model via a Gaussian-weighted combination of local linear functions that are updated incrementally via partial least squares (Vijayakumar et al. 2005). Let $\boldsymbol{\zeta}_{k-1} = (\mathbf{x}_{k-1}, \mathbf{u}_{k-1})$ be the input to the model learner and $\boldsymbol{\rho}_k$ be the output such that $\mathbf{x}_k = \tilde{\mathbf{x}}_k + \boldsymbol{\rho}_k$, where $\tilde{\mathbf{x}}_k$ is the state predicted by the deterministic, linearized dynamics model and $\mathbf{x}_k$ is the observed state. LWPR learns an element-wise prediction model, each of which consists of N_i local models, $(\alpha_j, \boldsymbol{\beta}_j)$, with weights w_j defined by a Gaussian kernel (mean $\mathbf{m}_j$, covariance $\mathbf{D}_j$),

$$\hat{\rho}_i(\boldsymbol{\zeta}) = \frac{1}{W}\sum_{j=1}^{N_i} w_j(\boldsymbol{\zeta})(\alpha_j + \boldsymbol{\beta}_j^{\mathrm{T}}(\boldsymbol{\zeta} - \mathbf{m}_j))$$
$$w_j(\boldsymbol{\zeta}) = \exp\left(-\frac{1}{2}(\boldsymbol{\zeta} - \mathbf{m}_j)^{\mathrm{T}}\mathbf{D}_j^{-1}(\boldsymbol{\zeta} - \mathbf{m}_j)\right)$$
$$W = \sum_{j=1}^{N_i} w_j(\boldsymbol{\zeta})$$

Therefore, given the previous state-control pair, $\boldsymbol{\zeta}_{k-1}$, LWPR returns a prediction, $\hat{\boldsymbol{\rho}}_k$, of the error between the predicted and actual current state.

2.4.2 ISSGPR Model Learner: We can also learn the same input-output model via Incremental Sparse Spectrum Gaussian Process Regression (ISSGPR) (Gijsberts and Metta 2013). ISSGPR 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 has cubic run time in the number of data points, ISSGPR achieves constant time by using an explicit sinusoidal feature space to avoid the expensive computation of the Gramian matrix. As ISSGPR also regresses to a scalar output, we fit the dynamics model element-wise.

To generate predictions, ISSGPR approximates the Asymmetric Squared Exponential kernel function (commonly used in Gaussian process regression),

$$k(\boldsymbol{\zeta}_i, \boldsymbol{\zeta}_j) = \sigma^2 e^{-\frac{1}{2}(\boldsymbol{\zeta}_i - \boldsymbol{\zeta}_j)^{\mathrm{T}}\mathbf{M}(\boldsymbol{\zeta}_i - \boldsymbol{\zeta}_j)}$$

by the inner product, $\boldsymbol{\phi}^{\mathrm{T}}\boldsymbol{\phi}$, where

$$\boldsymbol{\phi}(\boldsymbol{\zeta}) = \frac{1}{\sqrt{D}}\left[\cos(\boldsymbol{\omega}_1^{\mathrm{T}}\boldsymbol{\zeta}), \sin(\boldsymbol{\omega}_1^{\mathrm{T}}\boldsymbol{\zeta}), \dots, \cos(\boldsymbol{\omega}_D^{\mathrm{T}}\boldsymbol{\zeta}), \sin(\boldsymbol{\omega}_D^{\mathrm{T}}\boldsymbol{\zeta})\right]^{\mathrm{T}}$$

The $\boldsymbol{\omega}_d$ terms denote D random frequencies drawn from the multivariate Gaussian, $\mathcal{N}(\mathbf{0}, \mathbf{M})$. With this approximation, ISSGPR follows standard Gaussian process regression to generate predictions (Rasmussen and Williams 2006). Additionally, this formulation enables incremental updates on the prediction model, with the predicted mean given by

$$\hat{\boldsymbol{\rho}}_k = \hat{\boldsymbol{\rho}}_{k-1} + \boldsymbol{\phi}^{\mathrm{T}}\boldsymbol{\rho}_k$$

2.4.3 Luenberger Disturbance Observer: Finally, we also consider a purely reactive adaptation strategy based on $\mathcal{L}_1$ adaptive control (Wang et al. 2013). 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.

Additionally, for all three approaches, we follow the insight from $\mathcal{L}_1$ adaptive control and apply a low-pass filter to the output of the learner/observer before it is provided to the controller. The bandwidth of this filter is tuned to the system's response time to avoid destabilizing the system with rapid model perturbations.

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 (17) and $\boldsymbol{\delta}^{\mathbf{x}}$, $\boldsymbol{\delta}^{\mathbf{u}}$ via (7),(8)
 7:     Select $\boldsymbol{\mu}_0$ via (19)
 8:     for each element $m_i \in \mathcal{M}$ do
 9:         Compute $\boldsymbol{u}, \boldsymbol{\lambda}$ via (14),(16)
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 (15) from $m_i$
14:         end if
15:     end for
16:     if solution_found is false then
17:         Apply interm. control via (12) 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 (12) (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
```

2.5 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 (11). 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 (19). 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 (15) 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.

As in Fig. 1b, if no element satisfies the KKT conditions (line 16), a new element is computed via (12) 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 (14) and KKT matrices (16), 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 (Grancharova and Johansen 2012), while transitions to and from the intermediate controller will preserve stability if they are sufficiently infrequent (Hespanha and Morse 1999; Desaraju 2017).

3 Results

To assess the performance of the proposed Robust EPC algorithm, we consider a set of simulation studies with a skid-steer ground robot and a series of hardware experiments with three robot platforms: 1) a quadrotor micro air vehicle, 2) a skid steer ground robot with laser-based localization, and 3) a hexarotor aerial vehicle with visual odometry. Through these four cases, we aim to demonstrate the following results: **R1**: stable control performance, **R2**: real-time computation of control commands, **R3**: experience reuse, **R4**: constraint satisfaction in the presence of time-varying sensor uncertainty (i.e., robust constraint satisfaction), **R5**: improved trajectory tracking performance while satisfying constraints, and **R6**: robust constraint satisfaction during aggressive motion in challenging environments.

3.1 Simulated Ground Robot

We first consider a simulated ground robot with skid-steer dynamics equipped with a planar laser-scanner (270° field of view, 1081 beams, 30 m maximum range). The simulator implements a high-fidelity model of a skid-steer robot, including accelerations and actuator dynamics, and provides state feedback via a Simultaneous Localization and Mapping (SLAM) architecture that employs an unscented Kalman filter (UKF) to fuse estimates from ICP-based laser odometry and histogram filter-based localization (Nelson 2015). The UKF also provides covariance estimates that capture the uncertainty in the state estimate due to imperfect ICP solutions, thereby enabling use of the proposed Robust EPC algorithm.

To formulate Robust EPC, we employ a dynamics model that captures the translational dynamics in the $x - y$ plane, heading, θ, and the angular velocities of the left and right wheels, w_l and w_r, respectively. As the robot is modeled after a skid-steer platform with low-level velocity control, the control inputs available to Robust EPC are the body-frame velocity and angular velocity commands, v^{des} and

ω^{des}, respectively. The resulting nonlinear dynamics model is given by

$$f(\mathbf{x},\mathbf{u}) = \begin{cases} \dot{x} &= v\cos(\theta) \\ \ddot{x} &= \dot{v}\cos(\theta) - v\dot{\theta}\sin(\theta) \\ \dot{y} &= v\sin(\theta) \\ \ddot{y} &= \dot{v}\sin(\theta) + v\dot{\theta}\cos(\theta) \\ \dot{\theta} &= \omega \\ \ddot{\theta} &= \dot{\omega} \\ \dot{w}_l &= \frac{1}{R}\dot{v} - \frac{1}{2}\frac{T}{R}\dot{\omega} \\ \dot{w}_r &= \frac{1}{R}\dot{v} + \frac{1}{2}\frac{T}{R}\dot{\omega} \end{cases} \tag{20}$$

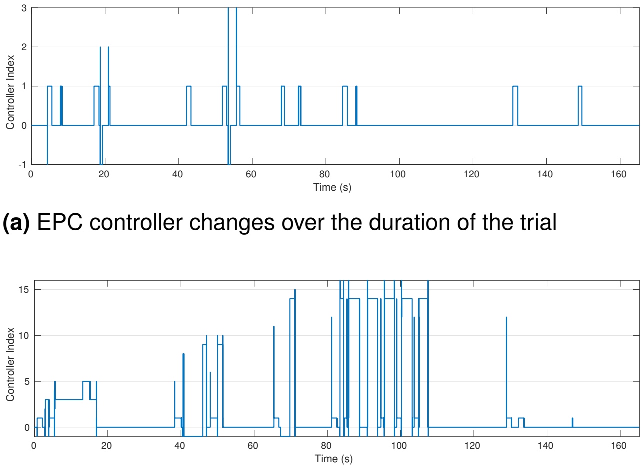
where $v = \frac{1}{2}R(w_l + w_r)$, $\omega = \frac{R}{T}(w_r - w_l)$, $\dot{v} = -K_f(v - v^{\text{des}})$, $\dot{\omega} = -K_\tau(\omega - \omega^{\text{des}})$, R is the wheel radius, T is the vehicle track (distance between left and right wheels), and K_f and K_τ are the low-level velocity control gains. We apply a simple observation model on the wheel speeds,

$$h(\mathbf{x}_k) = \begin{bmatrix} w_l \\ w_r \end{bmatrix}$$

The simulated ground robot is commanded to track a set of trajectories through the environment shown in Fig. 3 (e.g., for exploration or mapping applications). The localization system provides state estimates but also introduces uncertainty in these estimates due to imperfect registration of laser returns, thus replicating a common source of state estimate degradation in physical robotic systems. The dynamics model in (20) yields an MPC formulation with $n = 8$ states and $m = 2$ inputs. We apply a horizon of $N = 10$ steps at the controller update rate (200 Hz) and enforce constraints on the two control inputs (linear and angular velocity commands) as well as the translational and rotational rates $(\dot{x}, \dot{y}, \dot{\theta})$. The chance constraint parameter α is set to 0.001 to yield a constraint satisfaction probability of 99.9%.

To evaluate robust constraint satisfaction performance in the presence of imperfect state information, we compare Robust EPC with nominal (i.e., non-robust) EPC when commanded to track the same trajectory. Although the simulations are not run on a compute-constrained system (2.9 GHz Intel mobile processor), the relative query times demonstrate that the chance constrained extension does not significantly increase the compute times over regular EPC. For this trial, EPC yields a mean database query time of 0.1305 ms with a standard deviation of 0.0927 ms, while Robust EPC yields a mean of 0.1767 ms with a standard deviation of 0.1548 ms (**R2**). Additionally, both approaches learn and reuse controllers to enforce constraints, as expected. However, as Fig. 4 illustrates, Robust EPC computes and reuses more entries in its controller database, i.e., 17 entries, as opposed to four for EPC (**R3**). This increase is consistent with the constraint tightening formulation, as Robust EPC is expected to encounter the tightened constraints more frequently than EPC encounters the nominal constraints.

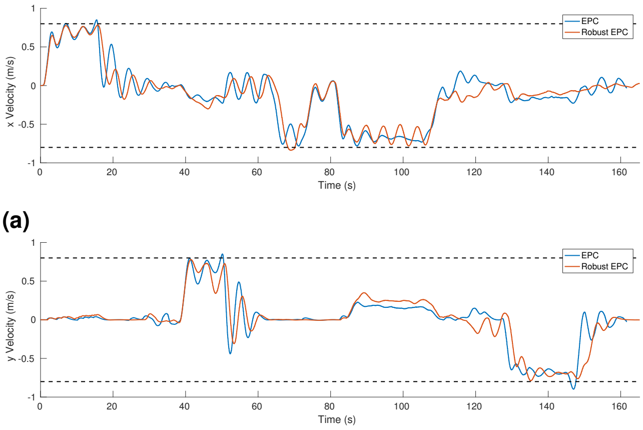
The effects of this increased database size and application of the corresponding controllers is evident in the resulting velocity profiles. Although both EPC and Robust EPC yield stable trajectory tracking (**R1**), as Fig. 5 shows, the nominal EPC formulation yields multiple velocity constraint violations along both axes. However, Robust EPC only yields one small violation of the x-axis velocity constraint, thus demonstrating robust constraint satisfaction in the presence of imperfect state information derived from the simulated laser-based localization system (**R4**).



(a) EPC controller changes over the duration of the trial

(b) Robust EPC controller changes over the duration of the trial

Figure 4. (a) EPC on the simulated ground robot computes and reuses four controllers (indexed 0-3) to enforce the nominal state and input constraints, while (b) Robust EPC applies 17 controllers to ensure robust constraint satisfaction (an index of -1 denotes application of the intermediate controller).



(a)

(b)

Figure 5. Velocity profiles for the simulated ground robot tracking the commanded trajectory using EPC and Robust EPC. The robust formulation yields more reliable constraint satisfaction (velocity constraints shown by dashed lines).

3.2 Experimental Platform 1: Small Quadrotor

The first experimental platform we consider 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. 6. All control algorithms are implemented in C++ via ROS (Quigley et al. 2009) and run in real-time on the ODROID.

The nonlinear dynamics of the quadrotor are modeled as

$$f(\mathbf{x},\mathbf{u}) = \begin{cases} \dot{\mathbf{p}} &= \mathbf{v} \\ \dot{\mathbf{v}} &= \frac{1}{m}F\mathbf{R}_{\boldsymbol{\xi}}\mathbf{e}_3 - g\mathbf{e}_3 \\ \dot{\boldsymbol{\xi}} &= \mathbf{S}_{\boldsymbol{\xi}}\boldsymbol{\omega} \\ \dot{\boldsymbol{\omega}} &= \mathbf{J}^{-1}\left(\boldsymbol{\tau} - \boldsymbol{\omega}\times\mathbf{J}\boldsymbol{\omega}\right) \end{cases} \tag{21}$$

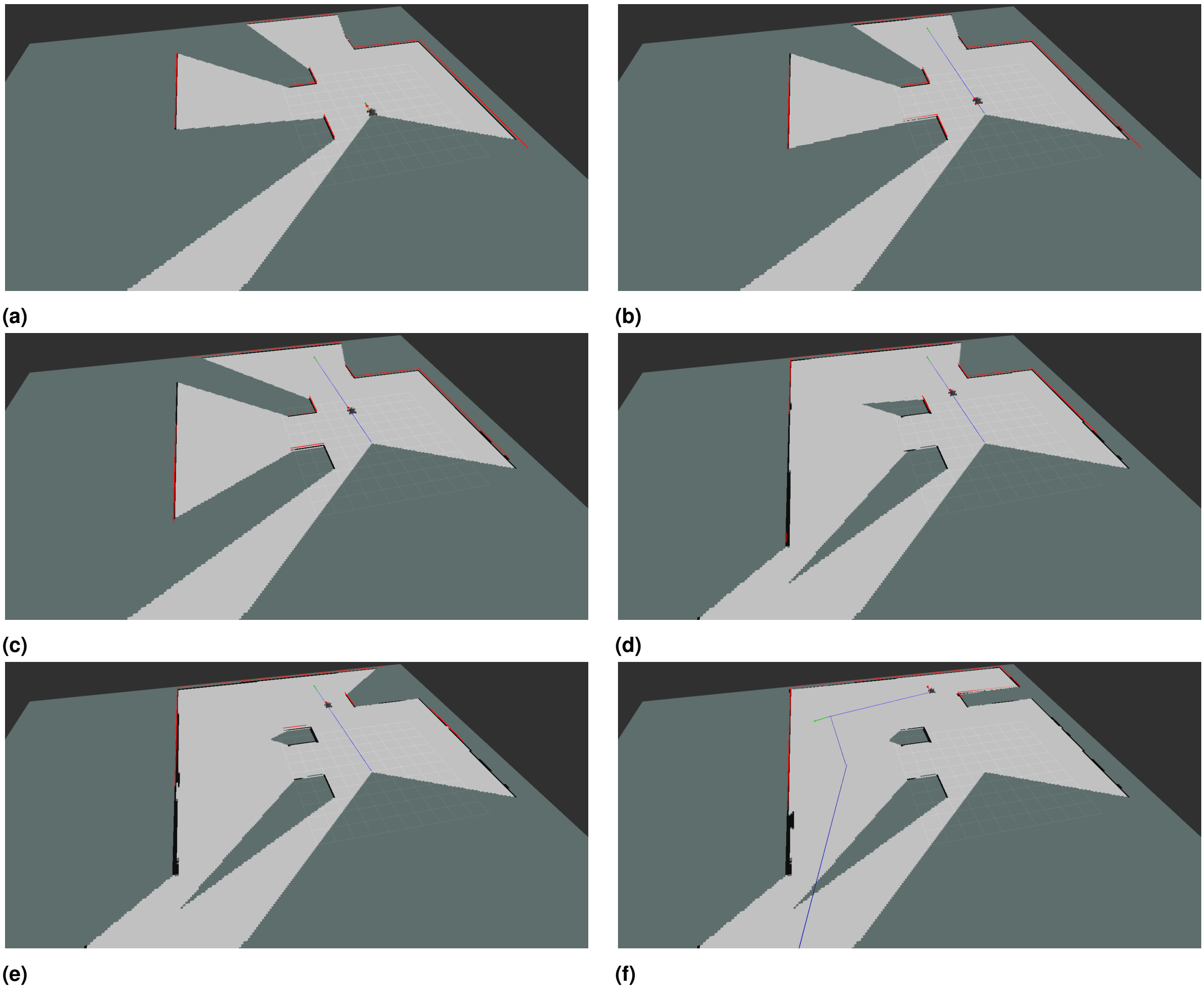

(a)
(b)
(c)
(d)
(e)
(f)

Figure 3. A series of snapshots showing a segment of the ground robot simulation trial. The blue lines denote the trajectory being tracked by the ground robot as it traverses the unknown environment. The successive frames illustrate the simulated laser scanner (red dots denote simulated laser returns) building a map of the environment that drives the localization subsystem.

where the constants g, m, and $\mathbf{J}$ denote gravity, vehicle mass, and inertia, respectively. The vector $\mathbf{e}_3$ is the third column of the 3×3 identity matrix, $\mathbf{R}_{\boldsymbol{\xi}}$ denotes the rotation matrix formed from the ZYX Euler angles $\boldsymbol{\xi}$ that takes vectors from body frame to world frame, and $\mathbf{S}_{\boldsymbol{\xi}}$ is the inverse of the Jacobian that relates ZYX Euler angle rates to angular velocities (Michael et al. 2010). We therefore employ a cascaded control setup with a translational controller providing references for an attitude controller that, in turn, provides actuator commands to the motor controllers.

A motion capture arena provides position and heading feedback that is combined with IMU measurements using an onboard UKF that employs the observation model,

$$h(\mathbf{x}_k) = \begin{bmatrix} \mathbf{p} \\ \psi \end{bmatrix}$$

where ψ denotes the yaw component of $\boldsymbol{\xi}$. 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).

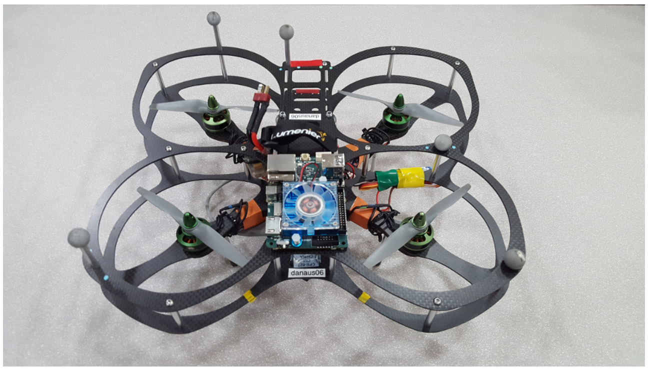


Figure 6. The quadrotor and ODROID-XU4 used for experimental validation of the Robust EPC algorithm.

For these experiments, we consider the problem of controlling the translational dynamics of the quadrotor

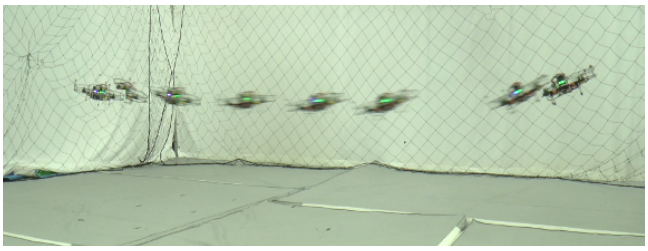


Figure 7. Snapshots of the quadrotor executing the linear trajectory.

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.

3.2.1 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. 7 (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. 8 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. 2.4 with parameters tuned empirically) yield low cross track error, and LWPR and ISSGPR exhibit zero-mean tracking as they accumulate experience. However, as Fig. 9 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 (Desaraju and Michael 2017) and proceed with LWPR for the following experimental studies.

3.2.2 Robust Constraint Satisfaction: We first evaluate Robust EPC's trajectory tracking performance along the linear trajectory in Fig. 7. Figure 10 shows that Robust EPC stabilizes the system to track the trajectory, which achieves a maximum linear velocity of 2.7 m/s (**R1**). Table 1 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.

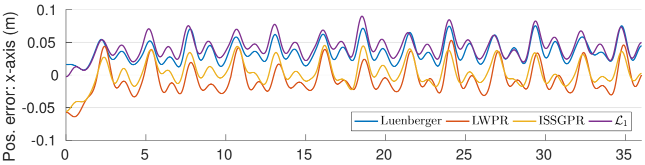


Figure 8. Cross-track error for quadrotor applying EPC with the three model adaptation strategies discussed and $\mathcal{L}_1$ adaptive control. LWPR and ISSGPR yield superior tracking after gaining experience. To better visualize the low-frequency components, we apply exponential smoothing with a two-second window.

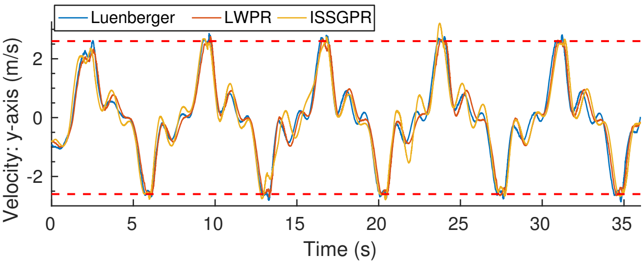


Figure 9. With injected noise, (non-robust) EPC yields velocity constraint violations for all three model adaptation strategies.

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 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 12 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

Table 1. Compute times for Robust EPC components running onboard the quadrotor. The first row indicates the number of control iterations across which the statistics are computed.

	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

[*] The attitude dynamics are stabilized via an instance of EPC with loose constraints. As the trajectories executed do not approach these constraint boundaries, we focus the analysis on the translational dynamics where constraints are activated.

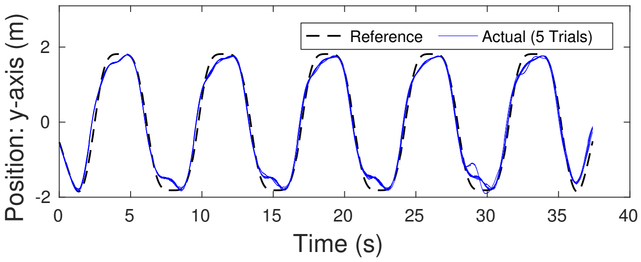


Figure 10. Linear trajectory tracking performance across five laps

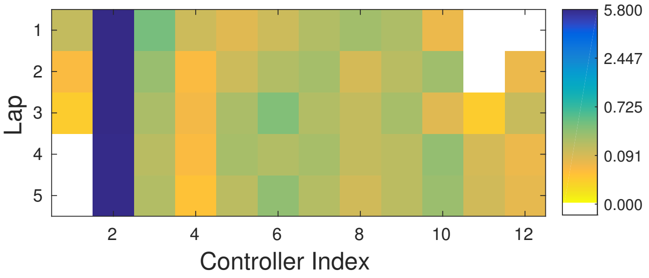
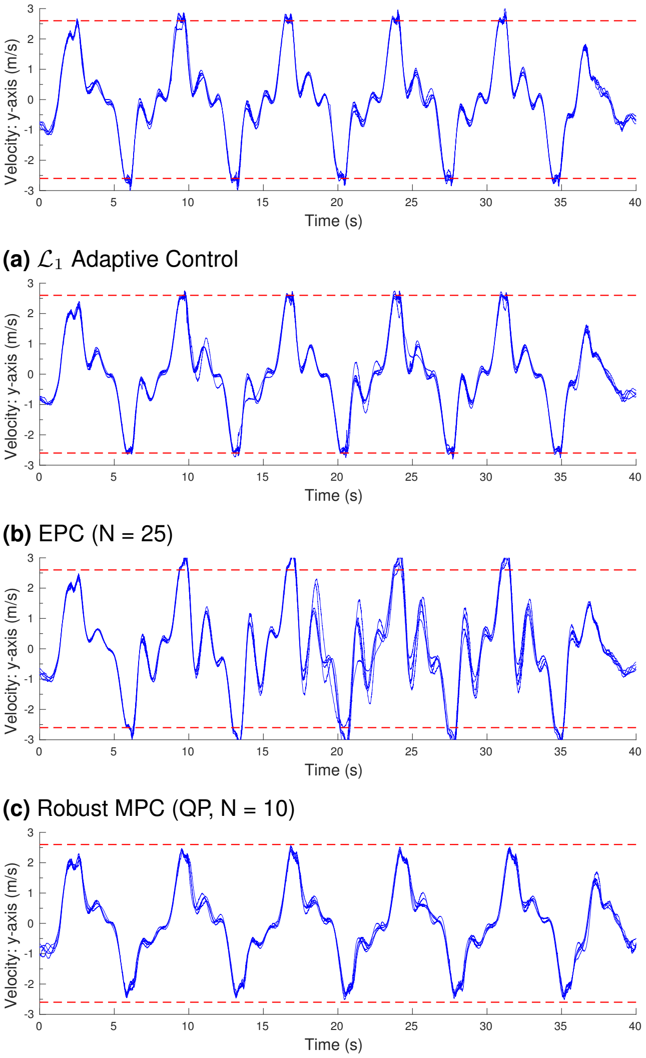


Figure 11. Time spent using each controller per lap. Note that multiple controllers are learned and reused and that the intermediate controller (index 1) ceases to be used past lap 3.

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 (**R4**).

Figure 11 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 (corresponding to operation away from constraints) is applied frequently (**R3**). 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.

3.2.3 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 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 14 shows tracking results for Robust EPC using three approaches: Gaussian belief propagation, the fixed bound approach using the true upper 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



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

(b) EPC (N = 25)

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

(d) Robust EPC (N = 25)

Figure 12. Comparison of y-velocity profiles for the quadrotor tracking the linear trajectory across 5 trials of each controller. Only Robust EPC satisfies the nominal velocity constraints (dashed lines).

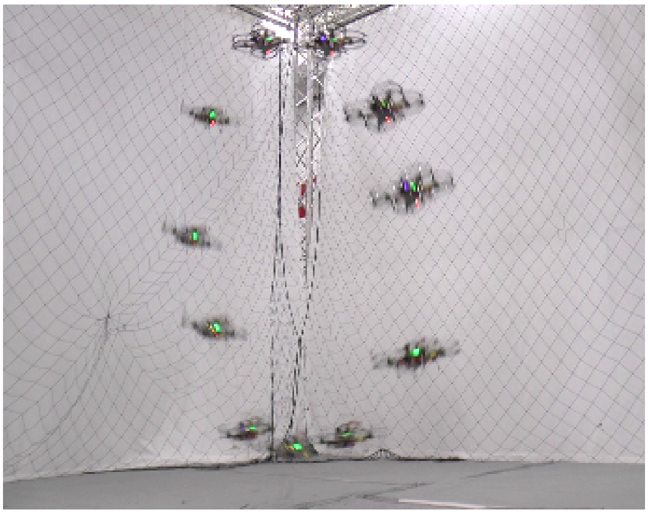


Figure 13. Quadrotor executing the vertical circle trajectory used to evaluate belief propagation.

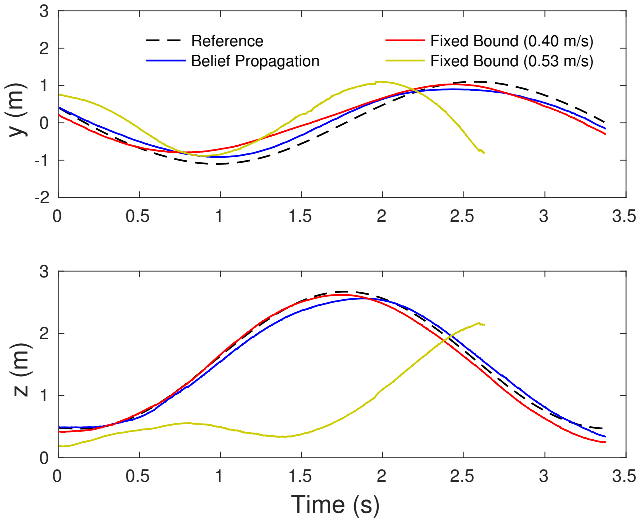


Figure 14. Quadrotor position along the y- and z-axes for Robust EPC and the fixed bound approach as compared to the reference trajectory. The fixed bound approach that uses the true upper bound (0.53 m/s) fails to track the trajectory. The mean and max error for Robust EPC along the y-axis are 0.22 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.

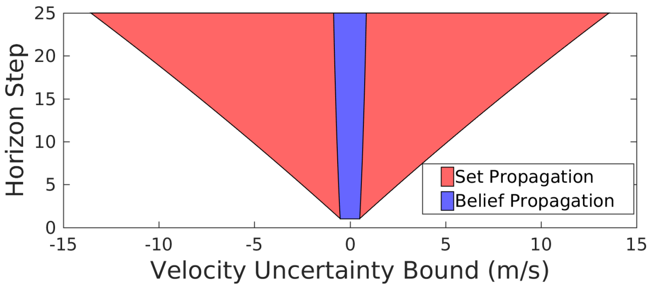


Figure 15. Overlay of tube growth for Set Propagation and Belief Propagation based on the bounds computed by each at the start of trajectory tracking. Set Propagation growth is too fast to yield feasible constraints.

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 (**R5**). We also consider uncertainty propagation via recursive application of the Pontryagin difference with the uncertainty set (Richards 2005). However, even with the ancillary controller, this results in an infeasible problem for the longer horizons permitted by Robust EPC. Figure 15 illustrates the tube growth with a 25-step horizon for the two approaches.

3.2.4 Aggressive Flight: To further assess the performance of Robust EPC, we consider two aggressive flight scenarios. The first scenario aims to test constraint satisfaction on a high speed back and forth trajectory with a maximum velocity of 3.6 m/s. As Fig. 16 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 (**R6**).

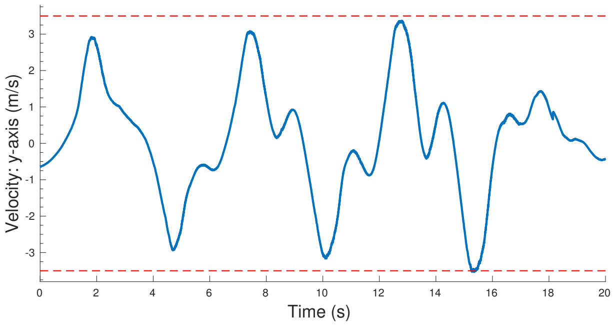


Figure 16. The quadrotor's velocity profile applying Robust EPC along a high-speed linear trajectory. There is a small constraint violation of 0.03 m/s during the last lap.

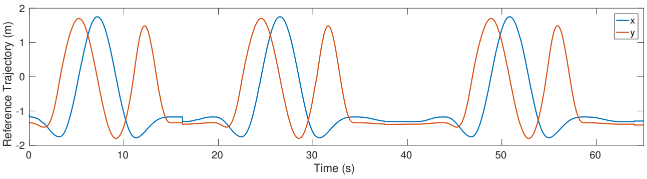


Figure 18. The x and y components of the horizontal circle trajectory showing the three laps executed by the quadrotor.

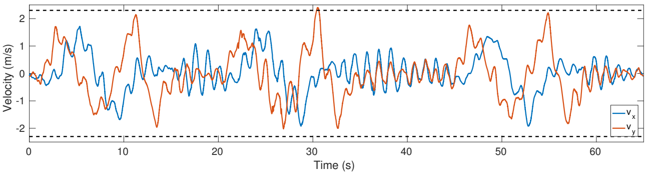


Figure 19. Velocity profile for the quadrotor applying Robust EPC along the circle trajectory in the high-wind scenario. The velocity obeys the constraint bound aside from one minor constraint violation of 0.09 m/s.

In the second aggressive flight scenario, the quadrotor is commanded to fly three laps around a circle in the x-y plane that traverses a turbulent wind field generated by eight, high-power fans, as illustrated in Figs. 17 and 18. The average wind velocity directly in front of each fan is approximately 6 m/s (Yao et al. 2016), and the placement of the fans around the flight volume results in significant spatial variation in the disturbance forces acting on the vehicle. For this scenario, we employ ISSGPR as the online model learner due to empirical evidence that it adapts to dynamic scenarios faster than LWPR (Desaraju 2017).

The reference trajectory commands a maximum velocity of 2.0 m/s, but due to the wind field, the vehicle may often overshoot the command. However, Robust EPC enforces a velocity limit of 2.3 m/s, and as Fig. 19 shows, the resulting velocity profile satisfies this constraint with just one minor violation. As a result, we conclude that Robust EPC adequately handles constraints even during aggressive flight with strong external perturbations to the dynamics model (**R6**).

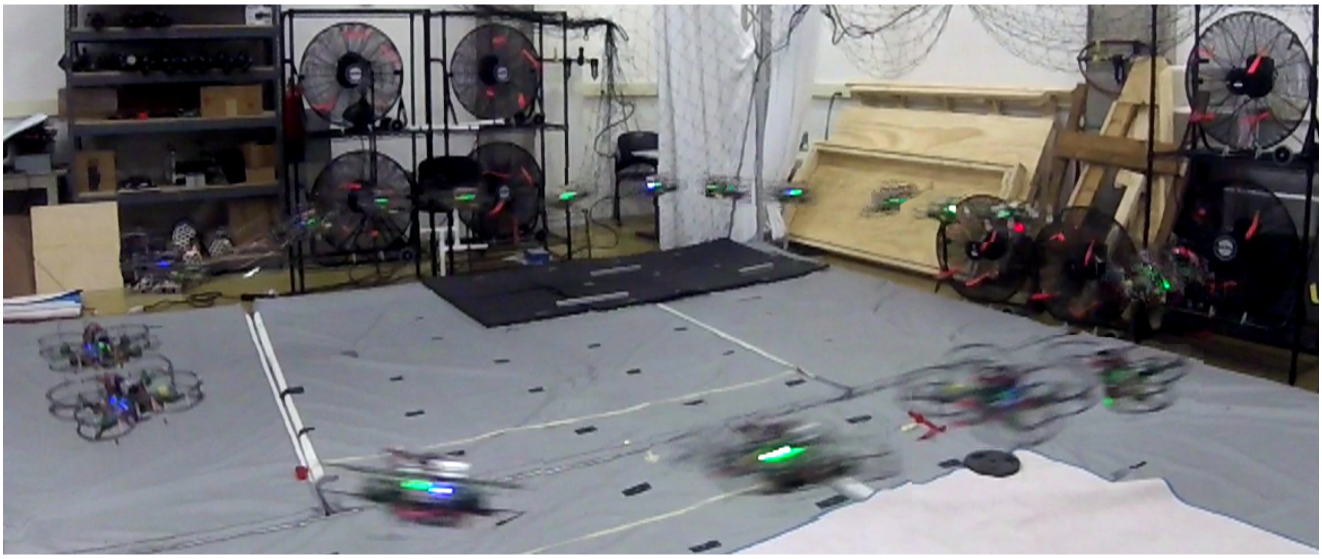


Figure 17. Snapshots of quadrotor executing the horizontal circle trajectory in a high-speed, turbulent wind field generated via a set of eight high-power fans

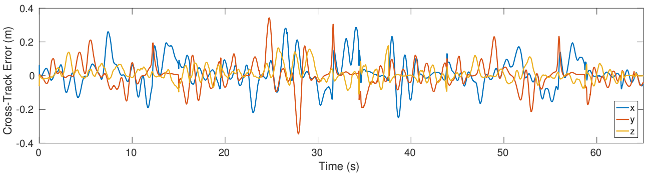


Figure 20. Cross-track error while the quadrotor executes the circle trajectory in the high-wind scenario is nearly zero-mean and shows some improvement over time.

Due to the velocity constraints being activated in each lap, it is difficult to assess flight performance via the tracking error (e.g., a high velocity may be needed to overcome wind-induced lag). We therefore look at cross-track error as a measure of the deviation from the trajectory, as shown in Fig. 20. As Table 2 shows, the cross-track error about all three axes improves significantly by the third lap as the controller database and model learner accumulate sufficient experience. This also matches empirical observations during the flight test that the vehicle exhibits improved stability and smoothness over successive laps (**R5**).

3.3 Experimental Platform 2: Ground Robot with Laser-based SLAM

The second experimental platform we consider is the skid-steer ground robot shown in Fig. 21. It is equipped with a Hokuyo UTM-30LX LIDAR (40 Hz scan rate, 270° field of view with 0.25° angular resolution, 30 m range with 30 mm accuracy) and a 3DM-GX3-35 IMU. The Robust EPC algorithm and the SLAM architecture described in Sect. 3.1 are implemented onboard the robot via a Gigabyte Brix CPU (2.4 GHz Intel i7 processor with 16 GB RAM).

We employ a slight variation on the simulated dynamics model (20) with a state space representation consisting of position, heading, and individual wheel velocities. The nonlinear dynamics model is given below

$$f(\mathbf{x},\mathbf{u}) = \begin{cases} \dot{x} &= v\cos(\theta) \\ \dot{y} &= v\sin(\theta) \\ \dot{\theta} &= \omega \\ \dot{w}_l &= \frac{1}{R}\dot{v} - \frac{1}{2}\frac{T}{R}\dot{\omega} \\ \dot{w}_r &= \frac{1}{R}\dot{v} + \frac{1}{2}\frac{T}{R}\dot{\omega} \end{cases} \tag{22}$$

where $v = \frac{1}{2}R(\omega_l + \omega_r)$ and $\omega = \frac{R}{T}(\omega_r - \omega_l)$ are the longitudinal and angular velocities, R is the wheel radius, and T is the track length of the vehicle. As the robot has low-level velocity controllers, the control inputs available to Robust EPC are the desired linear and angular velocities, v_d and ω_d, respectively, which enter (22) via linear and angular accelerations,

$$\dot{v} = \frac{F}{m} \quad \text{where} \quad F = -K_f(v - v_d)$$
$$\dot{\omega} = \frac{\tau}{J} \quad \text{where} \quad \tau = -K_\tau(\omega - \omega_d)$$

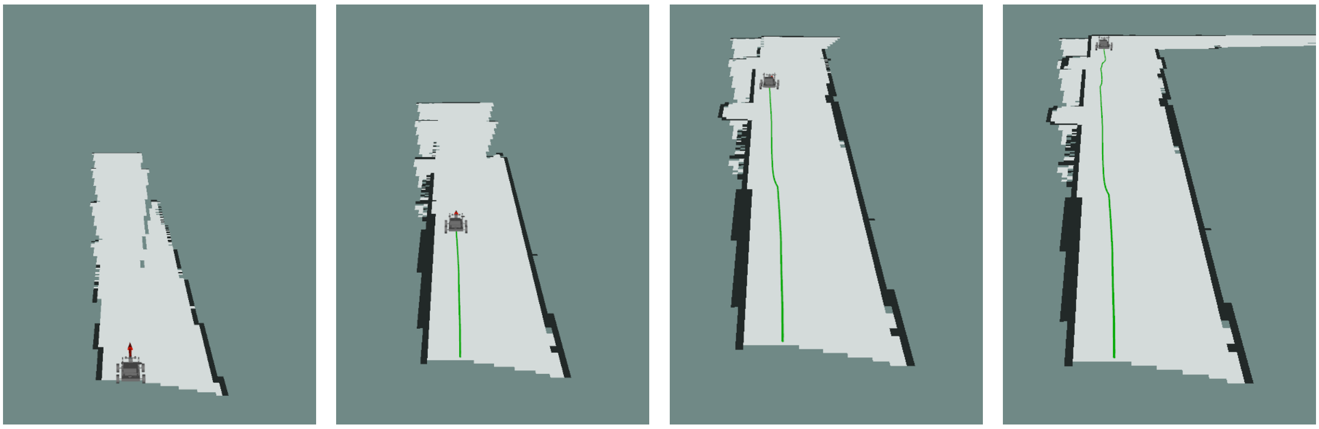
with K_f and K_τ denoting the low-level controller gains, m is the robot mass, and J is the robot inertia.

This model yields a Robust EPC formulation with $n = 5$ states, $m = 2$ inputs, and $N = 10$ steps at the controller update rate of 200 Hz. We set the chance constraint parameter to α to 0.001 for a constraint satisfaction probability of 99.9%. The ground robot is commanded to track a trajectory through a long hallway, as shown in Fig. 22 and Fig. 23, with constraints on longitudinal velocity and control inputs.

To verify constraint satisfaction, we first observe that Robust EPC computes and switches between 13 different controllers while executing this trajectory (**R3**). Figure 24 shows the velocity profiles for both EPC and Robust EPC executing this trajectory (**R1**). As expected, Robust EPC yields more reliable constraint satisfaction than EPC (**R4**). We do observe a brief constraint violation by Robust EPC though, similar to Sect. 3.1. While there is a small probability of constraint violation due to the chance-constrained formulation, this violation may also be indicative

Table 2. Cross-track error statistics for the quadrotor executing the high-wind, horizontal circle trajectory

	x-axis		y-axis		z-axis	
	Mean (m)	Std. Dev. (m)	Mean (m)	Std. Dev. (m)	Mean (m)	Std. Dev. (m)
Lap 1	0.0287	0.0893	0.0075	0.0694	0.0161	0.0364
Lap 2	0.0266	0.1060	0.0061	0.1057	0.0242	0.0528
Lap 3	0.0110	0.0785	0.0012	0.0740	0.0022	0.0378



(a) (b) (c) (d)

Figure 22. Snapshots of the ground robot platform tracking a trajectory through a 17.5 m long hallway. The green line indicates the robot trajectory and the black borders are the environment map constructed via the laser-based SLAM architecture.

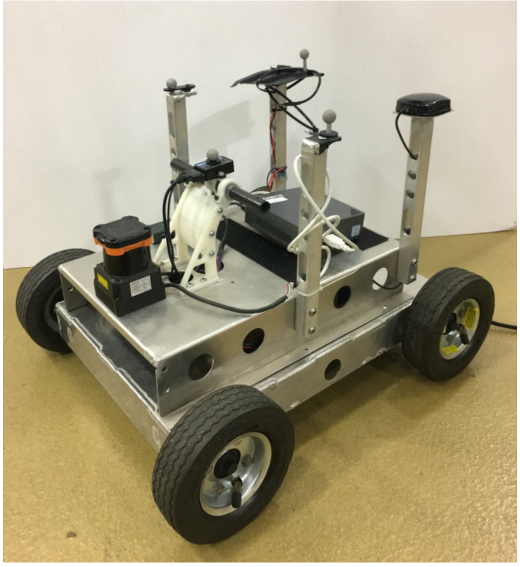


Figure 21. The laser-equipped ground robot platform used for experimental validation.

of insufficiently fast convergence in the LWPR model learner to capture fast, unmodeled dynamics (e.g., actuator effects).

As the ground robot has substantially increased computational resources compared to the quadrotor in Sect. 3.2, Robust EPC yields a mean database query time of 0.090 ms with a standard deviation of 0.0182 ms (**R2**). However, the increased resources stem from the need to run other processes for autonomy. Therefore, we also note that Robust EPC consumes less than 10% of the total CPU time and less than 0.1% of the total memory, thereby enabling more computationally expensive processes, such as SLAM (33% CPU time) or trajectory planning, to operate unimpeded.

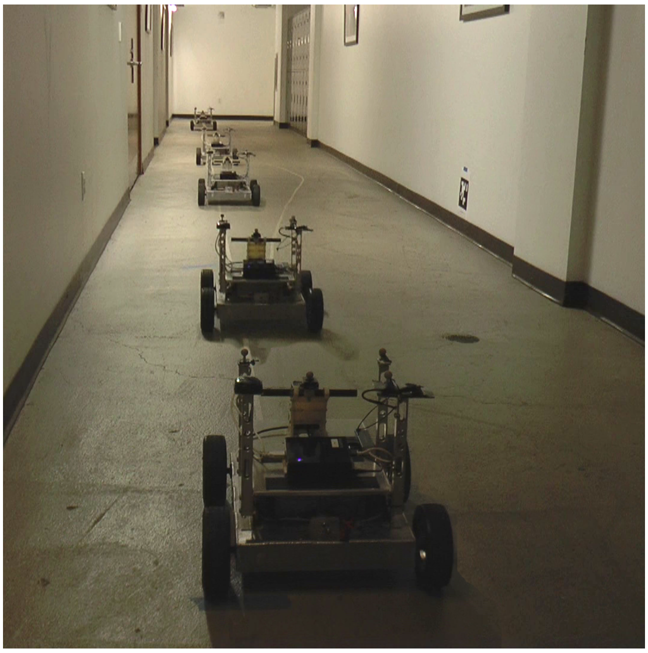


Figure 23. Overlaid video frames showing the ground robot tracking the trajectory through the long hallway via SLAM-based localization.

3.4 Experimental Platform 3: Hexarotor with Visual Odometry

The third experimental platform is the 3.46 kg hexarotor aerial robot shown in Fig. 25. The hexarotor obtains state feedback via a visual odometry system that consists of a sparse Lucas-Kanade tracker (Baker and Matthews 2004) applied to a discrete grid of points on a downward facing mvBlueFox camera's image stream running at 60 Hz. Ego motion of the vehicle is unrotated using an onboard VN100

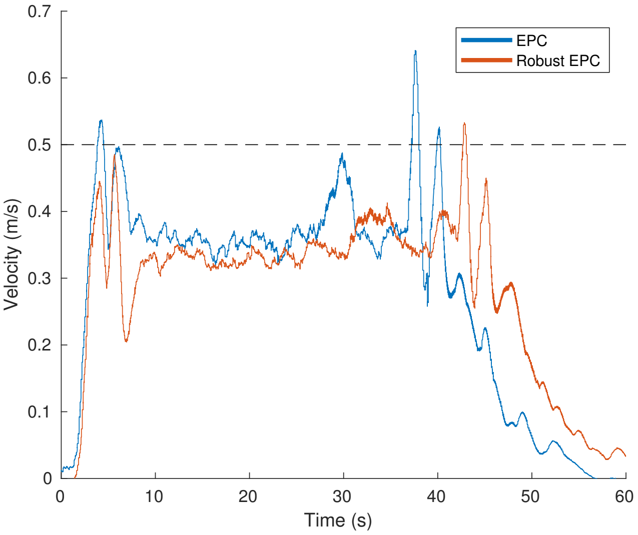


Figure 24. Longitudinal velocity profiles for the ground robot running EPC and Robust EPC. The dashed line indicates the enforced constraint bound.

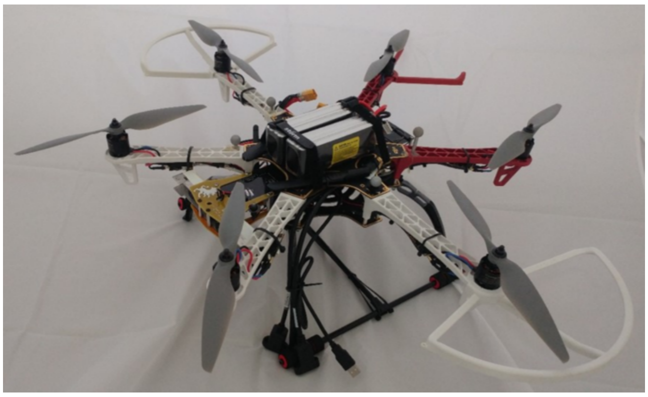


Figure 25. The hexarotor platform with a self-contained visual odometry system.

IMU, and a TeraRanger One laser altimeter provides scale. An unscented Kalman filter fuses the Lucas-Kanade tracker, altimeter readings, and VN100 IMU outputs to generate the state estimate. A window of 10 readings is used to compute running estimates of the standard deviation for the Lucas-Kanade tracker's estimated velocities, which are used to calculate the diagonal entries of the measurement noise covariance matrix. Robust EPC and the visual odometry system are implemented on the hexarotor's NVIDIA Jetson TX2 CPU (2 GHz ARM Cortex-A57, 2 GHz NVIDIA Denver2, 8 GB RAM).

We consider the same dynamics model and Robust EPC formulation used for the quadrotor in Sect. 3.2 with appropriate changes to the vehicle parameters and cost function. As the availability of the estimated measurement noise covariance yields more accurate belief propagation, we apply a reduced α of 0.01 to yield a constraint satisfaction probability of 99%.

The hexarotor tracks a series of smooth, linear trajectories at varying altitudes, as shown in Fig. 26 and Fig. 27 (**R1**). While tracking these trajectories, Robust EPC computes and switches between 33 different controllers (**R3**), and the mean database query time is 0.808 ms with standard deviation of 1.57 ms (**R2**).

The visual odometry system provides both the state estimate covariance and an estimate of the time-varying sensor measurement covariance, thus enabling Robust EPC to adapt the constraint tightening bounds over time. However, the fidelity of the visual odometry system degrades with increasing vehicle velocity (e.g., due to motion blur and reduced correspondences across image frames) and with increasing altitude (e.g., due to increased image coarseness). Figure 28 illustrates this behavior. The variation in the constraint bound, $\boldsymbol{\delta}^{\mathbf{x}}$, shown in Figs. 28a and 28c, automatically reflects the increase in uncertainty when the vehicle velocity increases (Figs. 28b and 28d) or altitude increases (Fig. 28e).

Figure 29 shows the results of a non-robust EPC executing the trajectory, resulting in repeated violations of the velocity

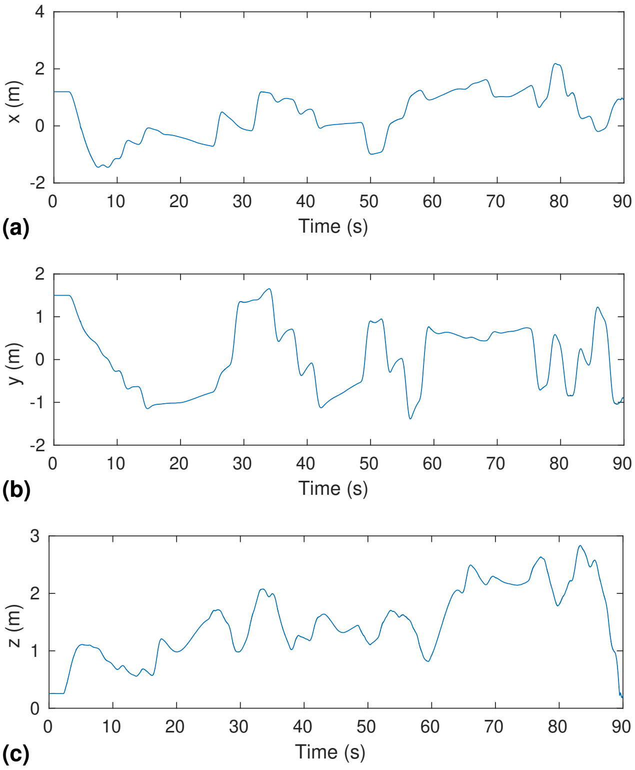


Figure 26. The set of random trajectories tracked by the hexarotor.

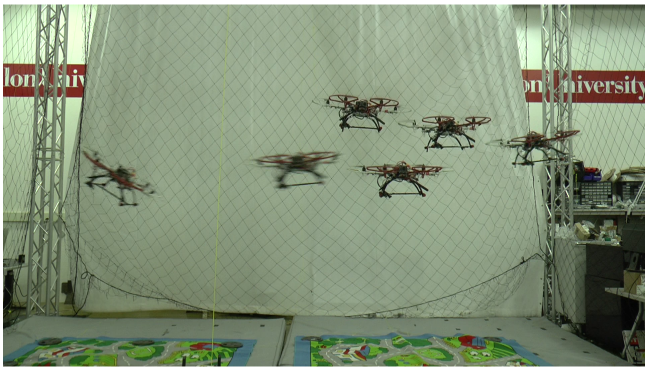
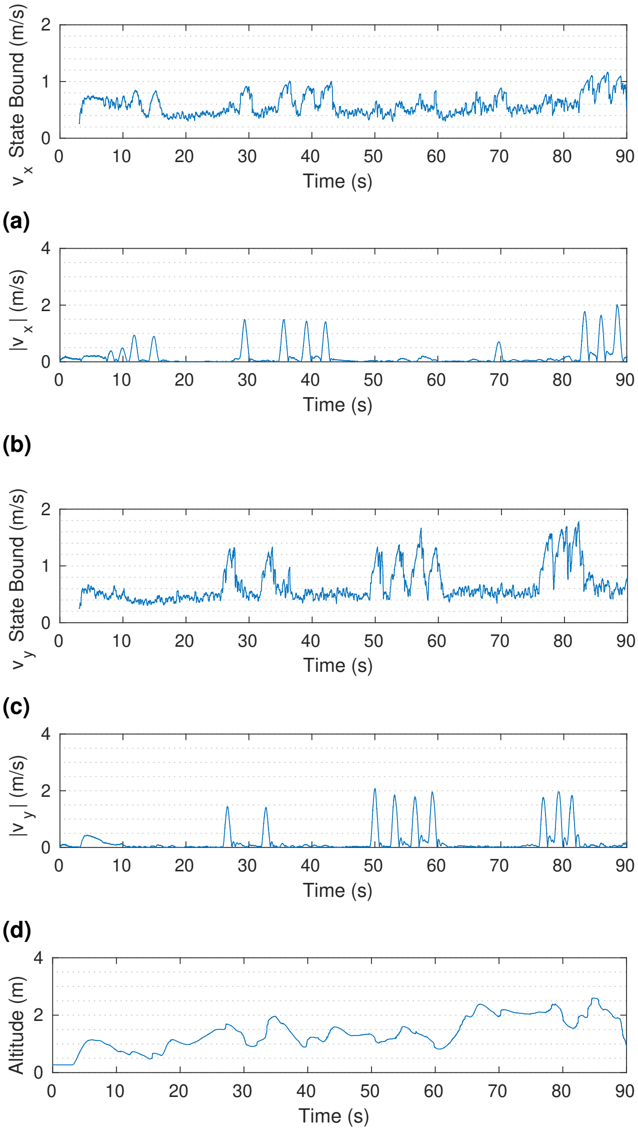


Figure 27. Overlay of snapshots showing the hexarotor executing a short segment of the random set of trajectories.



(a)

(b)

(c)

(d)

(e)

Figure 28. Constraint bounds on hexarotor x-velocity (a) and y-velocity (c) vary with the vehicle's current speed in each direction (b,d) and altitude (e), reflecting the changing uncertainty reported by visual odometry.

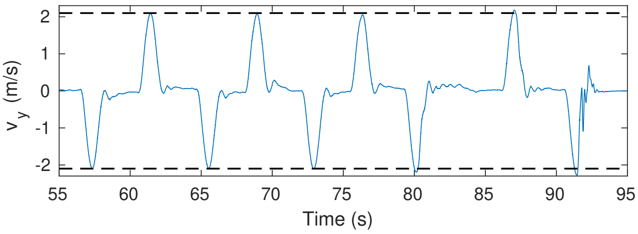


Figure 29. Velocity profile for the hexarotor applying (non-robust) EPC. The dashed lines indicate the constraint bounds.

constraints. In contrast, Robust EPC leverages this time-varying constraint bound and is more conservative at higher speeds or altitudes, thus yielding robust constraint satisfaction, as illustrated in Fig. 30 (**R4**).

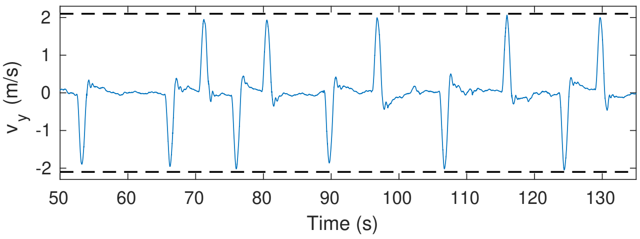


Figure 30. Velocity profile for the hexarotor applying Robust EPC. The dashed lines indicate the constraint bounds.

4 Conclusions and Future Work

In this work, we present an extension to Experience-driven Predictive Control (EPC) that yields robust constraint satisfaction in the presence of time- and state-dependent uncertainty. We have shown through simulation and experimental studies with three platforms, 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**), leverages past experiences via controller reuse (**R3**), reliably satisfies constraints in the presence of time-varying sensor uncertainty (**R4**) while improving tracking performance as compared to conservative methods (**R5**), and maintains constraint satisfaction properties during aggressive operation (**R6**). Moreover, the ground robot and hexarotor experiments demonstrate that Robust EPC yields robust constraint satisfaction with realistic state estimation systems.

The current Robust EPC formulation assumes that the uncertainty is well-modeled by a Gaussian distribution, but a potential future direction is to incorporate distributions without closed-form propagation models. 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.

Acknowledgements

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

References

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

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

Aswani A, Gonzalez H, Sastry SS and Tomlin C (2013) Provably Safe and Robust Learning-Based Model Predictive Control. *Automatica* 49(5): 1216–1226.

Baker S and Matthews I (2004) Lucas-Kanade 20 years on: A unifying framework. *Intl. J. of Computer Vision* 56(3): 221–255.

Bemporad A, M Morari VD and Pistikopoulos EN (2002) The explicit linear quadratic regulator for constrained systems. *Automatica* 38: 3–20.

Bouffard P, Aswani A and Tomlin C (2012) 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.

Brockers R, Humenberger M, Weiss S and Matthies L (2014) Towards Autonomous Navigation of Miniature UAV. In: *IEEE Conf. on Comp. Vision and Pattern Recog. Workshops*. Columbus, OH.

Brunner C and Peynot T (2010) Visual metrics for the evaluation of sensor data quality in outdoor perception. In: *Proc. of the Workshop on Perf. Metrics for Intell. Sys.* Baltimore, MD, pp. 1–8.

Bry A and Roy N (2011) Rapidly-exploring Random Belief Trees for motion planning under uncertainty. In: *Proc. of the IEEE Intl. Conf. on Robot. and Autom.* Shanghai, China.

Conte C, Zeilinger MN, Morari M and Jones CN (2013) Robust distributed model predictive control of linear systems. In: *Euro. Control Conf.* p. 5.

Desaraju VR (2017) *Safe, Efficient, and Robust Predictive Control of Constrained Nonlinear Systems*. PhD Thesis, Carnegie Mellon University.

Desaraju VR and Michael N (2016) Fast Nonlinear Model Predictive Control via Partial Enumeration. In: *Proc. of the IEEE Intl. Conf. on Robot. and Autom.* Stockholm, Sweden.

Desaraju VR and Michael N (2017) Leveraging Experience for Computationally Efficient Adaptive Nonlinear Model Predictive Control. In: *Proc. of the IEEE Intl. Conf. on Robot. and Autom.* Singapore.

Desaraju VR, Spitzer AE and Michael N (2017) Experience-driven Predictive Control with Robust Constraint Satisfaction under Time-Varying State Uncertainty. In: *Proc. of Robot.: Sci. and Syst.* Cambridge, MA.

Domahidi A, Zeilinger MN, Morari M and Jones CN (2011) Learning a Feasible and Stabilizing Explicit Model Predictive Control Law by Robust Optimization. In: *Proc. of the IEEE Conf. on Decision and Control*. IEEE, pp. 513–519.

Domes F and Neumaier A (2011) Rigorous Enclosures of Ellipsoids and Directed Cholesky Factorizations. *SIAM J. Matrix Anal. Appl.* 32: 62285.

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

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

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

Grancharova A and Johansen T (2012) *Explicit Nonlinear Model Predictive Control*, volume 429. Berlin, Heidelberg: Springer Berlin Heidelberg.

Hespanha JP and Morse AS (1999) Stability of switched systems with average dwell-time. In: *Proc. of the IEEE Conf. on Decision and Control*, volume 3. Phoenix, AZ, pp. 2655–2660.

Ho YK, Yeoh HK and Mjalli FS (2014) Generalized Predictive Control Algorithm with Real-Time Simultaneous Modeling and Tuning. *Industrial & Eng. Chem. Research* 53(22): 9411–9426.

Hofer M, Muehlebach M and DAndrea R (2016) 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.

Houska B, Ferreau HJ and Diehl M (2011) An auto-generated real-time iteration algorithm for nonlinear MPC in the microsecond range. *Automatica* 47(10): 2279–2285.

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

Kothare MV, Balakrishnan V and Morari M (1996) Robust constrained model predictive control using linear matrix inequalities. *Automatica* 32(10): 1361–1379.

Kuwata Y, Richards A and How J (2007) Robust Receding Horizon Control using Generalized Constraint Tightening. In: *Proc. of the Amer. Control Conf.* New York City, NY, pp. 4482–4487.

Langson W, Chryssochoos I, Raković SV and Mayne DQ (2004) Robust model predictive control using tubes. *Automatica* 40(1): 125–133.

Löfberg J (2003) *Minimax approaches to robust model predictive control*. PhD Thesis, Linköping University.

Mayne DQ (2014) Model predictive control: Recent developments and future promise. *Automatica* 50: 2967–2986.

Mayne DQ, Kerrigan EC, van Wyk EJ and Falugi P (2011) Tube-based robust nonlinear model predictive control. *J. Robust and Nonlin. Control* 21: 13411353.

Mayne DQ and Langson W (2001) Robustifying model predictive control of constrained linear systems. *Automatica* 37: 1422–1423.

Mayne DQ, Seron MM and Raković SV (2005) Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica* 41(2): 219–224.

Michael N, Mellinger D, Lindsey Q and Kumar V (2010) Experimental evaluation of multirobot aerial control algorithms. *IEEE Robot. Autom. Mag.* .

Nelson EA (2015) *Environment Model Adaptation for Autonomous Exploration*. Master's Thesis, Carnegie Mellon University.

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

Ostafew C, Schoellig A and Barfoot T (2014) 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.* IEEE, pp. 4029–4036.

Ostafew CJ, Schoellig AP and Barfoot TD (2016) Robust Constrained Learning-based NMPC enabling reliable mobile robot path tracking. *Intl. J. of Robotics Research* 35(13): 1547–1563.

Pannocchia G, Rawlings JB and Wright SJ (2007) Fast, large-scale model predictive control by partial enumeration. *Automatica* 43(5): 852–860.

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

Quigley M, Conley K, Gerkey B, Faust J, Foote T, Leibs J, Wheeler R and Ng AY (2009) ROS: An open-source Robot Operating System. In: *ICRA Workshop on open source software*. Kobe, Japan, p. 5.

Rasmussen CE and Williams CKI (2006) *Gaussian Processes for Machine Learning*. MIT Press.

Richards A (2005) Robust Model Predictive Control for Time-Varying Systems. In: *Proc. of the IEEE Conf. on Decision and Control*. Seville, Spain, pp. 3747–3752.

Richards A and How J (2005) Robust Model Predictive Control with Imperfect Information. In: *Proc. of the Amer. Control Conf.* New York City, NY.

Sudderth EB, Ihler AT, Isard M, Freeman WT and Willsky AS (2010) Nonparametric Belief Propagation. *Comm. of the ACM* 53: 95–103.

Tanaskovic M, Fagiano L, Smith R and Morari M (2014) Adaptive receding horizon control for constrained mimo systems. *Automatica* 50(12): 3019–3029.

Toit NED and Burdick JW (2010) Robotic motion planning in dynamic, cluttered, uncertain environments. In: *Proc. of the IEEE Intl. Conf. on Robot. and Autom.* Anchorage, AK.

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

Wang J, Holzapfel F, Xargay E and Hovakimyan N (2013) 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.

Yan J and Bitmead RR (2005) Incorporating state estimation into model predictive control and its application to network traffic control. *Automatica* 41: 595–604.

Yao J, Desaraju VR and Michael N (2016) Experience-Based Models of Surface Proximal Aerial Robot Flight Performance in Wind. In: *Proc. of the Intl. Sym. on Exp. Robot.* Tokyo, Japan, pp. 1–10.

Zeilinger MN, Morari M and Jones CN (2014) Soft constrained model predictive control with robust stability guarantees. *IEEE Trans. Autom. Control* 59(5): 1190–1202.