

# Extensions of Learning-Based Model Predictive Control for Real-Time Application to a Quadrotor Helicopter

Anil Aswani, Patrick Bouffard, Claire Tomlin

**Abstract**—A new technique called learning-based model predictive control (LBMPC) rigorously combines statistics and learning with control engineering, while providing levels of guarantees about safety, robustness, and convergence. This paper describes modifications of LBMPC that enable its real-time implementation on an ultra-low-voltage processor that is onboard a quadrotor helicopter testbed, and it also discusses the numerical algorithms used to implement the control scheme on the quadrotor. Experimental results are provided that demonstrate the improvement to dynamic response that the learning in LBMPC provides, as well as the robustness of LBMPC to mis-learning.

## I. INTRODUCTION

Linear control is popular in applications because of its simplicity and robustness, but this can come at the expense of performance. Increasing societal and economic pressures require techniques that more easily handle the tradeoff between robustness and performance, and learning-based model predictive control (LBMPC) [1] is one such new method: It handles system constraints, optimizes performance with respect to a cost function, uses statistical identification tools to learn model uncertainties, and provably converges.

In one application, LBMPC led to a 30-70% reduction in electrical energy usage on BRITE (an air-conditioning testbed) over the standard thermostat control [2]; LBMPC had more consistent temperature regulation and energy reduction properties than linear MPC. Encouraged by the results on BRITE, we have implemented LBMPC on a quadrotor helicopter. This paper presents a version of LBMPC that is modified to enable its application on the quadrotor.

The LBMPC method has its origins in robust [3], [4], adaptive [5], [6], [7], learning-based [8], [9], [10], and model predictive control (MPC) [11], [12], [13], [14], [15], [16]. It has similarities to the adaptive MPC in [17]. Adaptive control modifies controller parameters to match a reference closed-loop response, and learning-based control improves performance by using expert-guidance or iterated experiments. Unfortunately, model learning alone cannot be used to guarantee safety [18], [19], and so these methods often rely on expert-guided trajectories for safety.

The main insight of LBMPC is that performance and safety in MPC can be decoupled. Performance is specified by

making the cost function be dependent on learned dynamics, and safety is ensured by letting the constraints be dependent on an approximate model and its uncertainty. The LBMPC scheme is provably safe as long as it can be shown that a nominal controller provides safety for the approximate model with its uncertainty. This can be done for linear systems [20], [21], piecewise-affine systems [22], [23], [24], [25], and (nonlinear) continuous-time systems [26], [27], [28], [29].

The differences between LBMPC in [1] and the modifications here include: extension to linear affine systems, restructuring constraints to reduce conservativeness, decoupling of the feedback gains used in the cost function and used to compute the terminal set, a method for computing an outer approximation of an important reachability set, an analysis to ensure that the learning is designed so as to not do more than fundamental theoretical limitations preclude, and an extension of dual extended Kalman filters to handle the type of learning used for the quadrotor application.

We introduce the quadrotor and then discuss approximations of an important reachable set. The modified LBMPC and the learning used on the quadrotor are summarized. Next, the optimization software used to compute the control is described. Lastly, we provide results from two experiments of LBMPC on the quadrotor. A companion paper [30] contains details on system identification, equipment setup, and additional experiments. The video [http://www.youtube.com/watch?v=dL\\_ZFSvLXlU](http://www.youtube.com/watch?v=dL_ZFSvLXlU) shows experiments.

## II. QUADROTOR HELICOPTER TESTBED

We experimented on an AscTec Pelican quadrotor with a 1.6Ghz single-core Intel Atom N270 processor, 1GB of RAM, and 8GB of storage memory. It runs a software stack developed for quadrotor experimentation [31]. The “low-level” controller provided by AscTec accepts attitude (i.e., roll, pitch, yaw rate) and thrust commands. Our laboratory environment is equipped with a Vicon MX motion capture system, which tracks the 3D position of small retro-reflective markers using an array of cameras with near-infrared illumination strobes. The system can measure the rigid body pose of an object equipped with several markers, and this provides measurements of the system state. A laptop computer integrates these components through a local area network, with the quadrotor using WiFi. The laptop allows both manual and autonomous modes of flight.

## III. PRELIMINARIES

This section introduces the components of our technique, and it defines the discrete time model used for control.

This material is based upon work supported by NSF #CNS-0931843, ONR MURI (N00014-09-1-1051), and an NSERC fellowship (P. Bouffard). The conclusions are those of the authors and should not be interpreted as representing the official policies of the NSF, ONR, and NSERC.

A. Aswani, P. Bouffard, and C. Tomlin are with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, USA {aaswani,bouffard,tomlin}@eecs.berkeley.edu

### A. Notation

The  $j$ -th element of a vector  $v$  is denoted as  $v_j$ , and we use  $A'$  to denote the matrix transpose of  $A$ . Marks above a variable distinguish the state, output, and input of different models of the same system. The true system has state  $x$ , the estimated state is  $\hat{x}$ , the linear model with disturbance has state  $\bar{x}$ , and the model with oracle [1] has state  $\tilde{x}$ . Similar marks are used for the corresponding inputs and outputs.

### B. Polytopes

Let  $\mathcal{V}$  be a convex, compact polytope in  $\mathbb{R}^n$ . This set can be represented as a set of linear inequalities [15]. If  $L$  is the corresponding number of inequalities, then this set is

$$\mathcal{V} = \{x : F_v x \leq h_v\}, \quad (1)$$

where  $F_v \in \mathbb{R}^{L \times n}$  and  $h_v \in \mathbb{R}^L$ . There is another representation of  $\mathcal{V}$  as the convex hull of a set of points, but this formulation is not amenable to numerical optimization.

The Minkowski sum [32] of two sets  $\mathcal{U}, \mathcal{V}$  is defined as  $\mathcal{U} \oplus \mathcal{V} = \{u + v : u \in \mathcal{U}; v \in \mathcal{V}\}$ , and their Pontryagin set difference [32], [33] is defined as  $\mathcal{U} \ominus \mathcal{V} = \{u : u \oplus \mathcal{V} \subseteq \mathcal{U}\}$ . Their properties are non-obvious: For instance,  $(\mathcal{U} \oplus \mathcal{V}) \ominus \mathcal{V} \neq \mathcal{U}$  in general. For the set  $\mathcal{U}$ , the linear transformation of the set by matrix  $T$  is given by  $T\mathcal{U} = \{Tu : u \in \mathcal{U}\}$ .

The Pontryagin set difference and linear transformation can be quickly computed [33], [15]; however, computing the Minkowski sum is NP-hard [34], except in special cases [35]. Thus, we mainly use it for theoretical arguments.

### C. Model

A discrete-time model of the quadrotor is used, where time is indexed by  $n \in \mathbb{Z}_+$ . The state  $x \in \mathbb{R}^{10}$  includes six states for position and velocity and four states for angular position and velocity; we do control in a frame of reference in which yaw is fixed. The inputs  $u \in \mathbb{R}^3$  specify thrust and two angular positions, and the measured output  $y \in \mathbb{R}^5$  is the three position states and two angular position states.

Using this notation, the dynamics of the quadrotor can be given by a model of the form

$$\begin{aligned} x[n+1] &= Ax[n] + Bu[n] + k + h(x[n], u[n]) \\ y[n] &= Cx[n] + \epsilon[n]. \end{aligned} \quad (2)$$

The matrices  $A \in \mathbb{R}^{10 \times 10}$ ,  $B \in \mathbb{R}^{10 \times 3}$  and vector  $k \in \mathbb{R}^{10}$  define the nominal model, and the readout matrix  $C \in \mathbb{R}^{5 \times 10}$  reflects the fact that only five out of the ten states are measured. Also,  $k$  is an affine term that incorporates the effect of gravity; the term  $h(x[n], u[n]) : \mathbb{R}^{10} \times \mathbb{R}^3 \rightarrow \mathbb{R}^{10}$  is Lipschitz continuous with respect to  $(x, u)$ , and it represents unmodeled dynamics. The  $\epsilon[n] \in \mathbb{R}^5$  term represents measurement error, and it is assumed to be independent and identically distributed at each time step. We assume that  $\epsilon[n] \in \mathcal{E}$ , where  $\mathcal{E}$  is a convex, compact polytope.

The system is safe if it satisfies state  $x[n] \in \mathcal{X}$  and input  $u[n] \in \mathcal{U}$  constraints, where  $\mathcal{X}, \mathcal{U}$  are convex, compact polytopes. Similarly, we assume that the function  $h(x, u)$  satisfies the constraints  $h(x, u) \in \mathcal{W}$  for all  $(x, u) \in (\mathcal{X}, \mathcal{U})$ , where  $\mathcal{W}$  is a convex, compact polytope. The intuition of

these assumptions is that the model represents a linear system with bounded uncertainty.

### D. The Oracle

The dynamics of the model that uses the oracle are given by

$$\tilde{x}[n+1] = A\tilde{x}[n] + B\tilde{u}[n] + k + \mathcal{O}_n(\tilde{x}[n], \tilde{u}[n]), \quad (3)$$

where  $\mathcal{O}_n$  is a time-varying oracle which returns an estimate of the unmodeled dynamics and  $A, B, k$  are taken from (2).

An important case is when  $\mathcal{O}_n$  is a disturbance  $d[n] \in \mathcal{D}$  for a bounded, convex polytope  $\mathcal{D}$ . We model the unknown dynamics and estimation error as a bounded disturbance and use a linear model to construct sets such that the control generated by the MPC with oracle is guaranteed to be safe. In this case, the state  $\bar{x}$ , output  $\bar{y}$ , and input  $\bar{u}$  of the linear model have dynamics

$$\bar{x}[n+1] = A\bar{x}[n] + B\bar{u}[n] + k + d[n]. \quad (4)$$

The form of the oracle  $\mathcal{O}_n$  that we use for model learning in the quadrotor application is the linear affine function

$$\mathcal{O}_n(x, u) = Fx + Hu + z, \quad (5)$$

where  $F \in \mathbb{R}^{10 \times 10}$ ,  $H \in \mathbb{R}^{10 \times 3}$ , and  $z \in \mathbb{R}^{10}$  are sparse. For example, the  $A_{2,1}$  entry is set to zero since the velocity does not depend on position, and the corresponding  $F_{2,1}$  entry is also set to zero because this entry should not be modified by the oracle. On the other hand, the  $B_{10,3}$  entry describes the aerodynamic lift provided per unit of thrust input, and the associated  $H_{10,3}$  entry corrects for changes in aerodynamic lift due to proximity to the ground (i.e., the ‘‘ground effect’’).

## IV. LEARNING-BASED MODEL PREDICTIVE CONTROL

This section presents the LBMPC method as used on the quadrotor, and theoretical results about the method can be found in [1]. Among these results is the fact that LBMPC can maintain three types of robustness. First, the value function of the optimization problem in LBMPC is continuous. Second, LBMPC is input-to-state stable (ISS) under certain conditions. Third, system constraints can be satisfied subject to all possible disturbances.

In some cases, maintaining a particular kind of robustness could possibly lead to overly conservative design. As designed, our implementation always maintains robustness due to continuity of the value function. Heuristics for robustness due to ISS and constraint satisfaction under all disturbances are designed based on the original LBMPC theory.

### A. Feasible Set Point Tracking

Before we present the modified LBMPC, we need to introduce some concepts related to feasible set point tracking for a constrained linear affine system subject to disturbances. The steady-state output  $y_s$  corresponds to some  $x_s, u_s$ , and it can be characterized [13], [16] by the set of solutions to

$$\begin{pmatrix} A - \mathbb{I} & B & 0 \\ C & 0 & -\mathbb{I} \end{pmatrix} \begin{pmatrix} x_s \\ u_s \\ y_s \end{pmatrix} = \begin{pmatrix} -k \\ 0 \end{pmatrix}. \quad (6)$$

These solutions form a set of affine subspaces that can be parametrized—similarly to [13], [16], [1]—as  $x_s = \Lambda\theta + x_0$ ,  $u_s = \Psi\theta + u_0$ ,  $y_s = \Pi\theta + y_0$ , where  $\theta \in \mathbb{R}^3$ ;  $\Lambda, \Psi, \Pi$ ; and  $x_0, u_0, y_0$  are (possibly zero) vectors.

The maximal output admissible disturbance invariant set  $\Omega \subseteq \mathcal{X} \times \mathbb{R}^3$  was defined in [33]. It is a set of points such that any trajectory of the system with initial condition chosen from this set remains within the set for any sequence of bounded disturbance, while satisfying constraints on the state and input. Consider any point  $(x, \theta) \in \Omega$ , where the  $\theta$  component parametrizes points that can feasibly be tracked using a linear controller. If  $\bar{K}$  is a nominal feedback gain such that  $(A + B\bar{K})$  is Schur stable, then the set  $\Omega$  satisfies

- disturbance invariance:

$$\begin{aligned} & (Ax + B\bar{K}(x - (\Lambda\theta + x_0)) + B(\Psi\theta + u_0) + k, \theta) \\ & \oplus (\mathcal{D}, 0) \subseteq \Omega; \quad (7) \end{aligned}$$

- constraint satisfaction:

$$\begin{aligned} \Omega \subseteq \{ & (x, \theta) : x \in \mathcal{X}; \Lambda\theta + x_0 \in \mathcal{X}; \\ & \bar{K}(x - (\Lambda\theta + x_0)) + (\Psi\theta + u_0) \in \mathcal{U}; \Psi\theta + u_0 \in \mathcal{U} \}. \end{aligned} \quad (8)$$

### B. Invariant Set Computations for the Quadrotor

Robust constraint satisfaction is the property that following the control law provided by the LBMPC will never lead to a situation in which a constraint is violated at some point in the future. It holds for both the original and modified LBMPC, provided that the set  $\Omega$  can be computed; however, we are unable to compute this set for the quadrotor. Because of these difficulties, we use an approximation of  $\Omega$  that displays good performance and robustness on the quadrotor.

Methods for computing  $\Omega$  [20], [21] start with an initial approximation of the set, and they refine the approximation by successively adding linear constraints to the polytope. The  $\Omega$  for the quadrotor may be comprised of many constraints, and this may explain why we could not compute  $\Omega$ : We may not have allowed the algorithms enough time to terminate. However, using a complex  $\Omega$  would increase the computation time for the LBMPC so as to render it not implementable in real time. The second possible explanation for why we cannot compute  $\Omega$  is that it does not exist for the parameters of our quadrotor, but this would be reflective of the overly-conservative nature of this type of robustness rather than a statement about the controllability of the system.

To overcome these difficulties, we use an outer approximation of the set  $\Omega$ . The idea is to start by rewriting the disturbance invariance condition (7) as

$$(Ax + B\bar{K}(x - (\Lambda\theta + x_0)) + B(\Psi\theta + u_0) + k, \theta) \subseteq \Omega \ominus (\mathcal{D}, 0) \quad (9)$$

and then relax it to

$$(Ax + B\bar{K}(x - (\Lambda\theta + x_0)) + B(\Psi\theta + u_0) + k, \theta) \subseteq \mathcal{X} \ominus (\mathcal{D}, 0). \quad (10)$$

The set of points that satisfy (9) are a subset of the points that satisfy (10), because  $\Omega \subseteq \mathcal{X}$  by construction.

Because the constraints  $\mathcal{X}, \mathcal{U}$  and disturbance bounds  $\mathcal{D}$  are compact, convex polytopes, we can exactly represent the outer approximation of  $\Omega$  as the points that satisfy a set of linear inequalities. Let  $F_P x \leq h_P$ ,  $F_x x \leq h_x$ , and  $F_u x \leq h_u$  be the inequality representations of the polytopes  $\mathcal{X} \ominus \mathcal{D}$ ,  $\mathcal{X}$ , and  $\mathcal{U}$ , respectively. The approximation  $\omega$  is given by the set of points  $(x, \theta)$  that satisfy the linear inequalities

$$\begin{bmatrix} F_P(A + B\bar{K}) & F_P(B(\Psi - \bar{K}\Lambda)) \\ F_x & 0 \\ 0 & F_x\Lambda \\ F_u\bar{K} & F_u(\Psi - \bar{K}\Lambda) \\ 0 & F_u\Psi \end{bmatrix} \begin{bmatrix} x \\ \theta \end{bmatrix} \leq \begin{bmatrix} K_1 \\ h_x \\ h_x - F_x\bar{x}_0 \\ K_2 \\ h_u - F_u\bar{u}_0 \end{bmatrix}, \quad (11)$$

where  $K_1 = h_P - F_P(k + B(\bar{K}\bar{x}_0 - \bar{u}_0))$  and  $K_2 = h_u - F_u(\bar{K}\bar{x}_0 - \bar{u}_0)$ .

### C. Modified LBMPC for the Quadrotor

LBMPC is based on a linear MPC scheme for tracking [13] that can be robustified using tube-MPC [11], [12], [16], and LBMPC reduces conservativeness by making the initial condition be fixed rather than be an optimization parameter (cf., [12], [16]). We further modify LBMPC to reduce conservativeness by assuming that noise enters only into the first time step of the model *prediction* (noise still enters into all time steps of the true system); the advantage of this is greatly enlarged regions of feasibility for the optimization problem that defines the LBMPC. This formulation of LBMPC has the same robust constraint satisfaction properties of the original form of LBMPC (the results in [1] trivially extend to this case).

Suppose the control objective is to track  $x_s$ . The estimated steady-state control that keeps the system at this point is given by  $u_s$  such that  $(A + F)x_s + (B + H)u_s + (k + z) = x_s$ , and this has an explicit solution in terms of the Moore-Penrose pseudoinverse. The modified LBMPC is given by

$$\begin{aligned} & \min_{c[\cdot], \theta} \|\tilde{x}[m + N] - \bar{x}_s\|_P^2 + \\ & \sum_{j=0}^{N-1} \|\tilde{x}[m + j] - \bar{x}_s\|_Q^2 + \|\check{u}[m + j] - \bar{u}_s\|_R^2 \quad (12) \\ \text{s.t. } & \tilde{x}[m] = \hat{x}[m], \quad \bar{x}[m] = \hat{x}[m] \quad (13) \\ & \tilde{x}[m + i] = (A + F)\tilde{x}[m + i - 1] \\ & \quad + (B + H)\check{u}[m + i - 1] + k + z \quad (14) \\ & \bar{x}[m + i] = A\bar{x}[m + i - 1] + B\check{u}[m + i - 1] + k \\ & \check{u}[m + i - 1] = K\bar{x}[m + i - 1] + c[m + i - 1] \\ & \bar{x}[m + i] \in \mathcal{X}, \quad \check{u}[m + i - 1] \in \mathcal{U} \\ & \bar{x}[m + 1] \in \mathcal{X} \ominus \mathcal{D} \\ & (\bar{x}[m + 1], \theta) \in \omega \quad (15) \end{aligned}$$

for all  $i \in I = \{1, \dots, N\}$  in the constraints;  $K$  is a nominal feedback gain;  $\omega$  is the outer approximation of  $\Omega$ ; and the oracle is the function  $Fx + Hu + z$ . The nominal feedback gain  $K$  is different from the gain  $\bar{K}$ , and it is chosen so that the discrete-time algebraic Riccati equation (DARE)

$$(A + BK)'P(A + BK) - P = -(Q + K'RK) \quad (16)$$

is satisfied. The constraints in this scheme are applied to the linear model (4) and *not* to the oracle model (3). However, the same control  $\tilde{u}[\cdot]$  is used in both models.

Because the functions in the optimization problem are continuous and the constraints are linear, this implies continuity of the value function [1]; this is in fact one type of robustness [36]. ISS can be shown if the oracle is bounded and this scheme can be proven to be convergent for the nominal model [1], but we have been unable to prove this. The reason for this is that existing proof techniques apply to the case where  $\bar{K} \equiv K$ , but that is not true here. However, our empirical observation of an implementation on the quadrotor is that this scheme is effectively ISS; the quadrotor tracks a steady point within a root-mean-square error of about 2 cm in each positional direction.

The reason for having  $\bar{K} \neq K$  is that it leads to better empirical performance on the quadrotor. The intuition for why this is the case is as follows. The smaller the gain  $\bar{K}$  is, then the larger the set  $\Omega$  will be. The advantage of a larger  $\Omega$  is a larger feasible region of the LBMPC, and so performance and robustness will be better. On the other hand, having a large gain  $K$  leads to fast convergence and tracking for the quadrotor, but it would lead to a very small set  $\Omega$ . Thus, we use different values for these two gains.

## V. FILTERING

Not all states are directly measured, and so the LBMPC technique needs to be able to estimate these states in order to be able to do control; however, the situation is more complicated because there are unknown coefficients  $\beta$  in the oracle  $\mathcal{O}_n(x, u) = Fx + Hu + z$ . Estimating the states and coefficients simultaneously must be done with care because of two reasons. The first is that there are fundamental limitations: In general, it is not possible to simultaneously estimate all parameters of an oracle and all states [1]. The second reason is that the more parameters that need to be estimated, the higher the estimation error will generally be.

Among the conditions required to ensure that the LBMPC technique is ISS is that the oracle should be bounded. Even though we cannot ensure all of the conditions for ISS (specifically, we cannot show that the value function is a Lyapunov function), it is good design practice to meet as many conditions for ISS as possible. We enforce *a priori* bounds on the parameters of the oracle

$$|\beta_i| \leq M_i, \forall i \quad (17)$$

and these bounds can be derived, for instance, from the bounds on  $h(x, u)$  which represents the uncertainty on the nominal model. Placing these bounds on the parameters and oracle help to ensure good empirical performance.

### A. Checking Observability

Suppose that  $z$  were not sparse and that every entry is a parameter  $z = [\beta_8 \ \dots \ \beta_{17}]$ . Results in [1], for instance, show that it is impossible in this system to simultaneously identify the parameters of this  $z$  while computing state estimates  $\hat{x}$  based on noisy measurements  $y$ . The central

issue is that of observability: If we augment the state to  $(x, \beta)$ , then being able to identify all the parameters  $\beta$  and estimate the states  $x$  is equivalent to the observability of the system

$$\begin{bmatrix} x[n+1] \\ \beta[n+1] \end{bmatrix} = \begin{bmatrix} (A+F)x[n] + (B+H)u[n] + (k+z) \\ \beta[n] \end{bmatrix}. \quad (18)$$

These equations are jointly nonlinear in  $(x, \beta)$ , even though they are linear in  $x$  for fixed  $\beta$  and *vice versa*. In our application to the quadrotor, we used standard techniques [37] to verify the joint observability of the unmeasured states and the parameters to be identified by the oracle.

### B. Dual Extended Kalman Filter

A natural approach for joint state and parameter estimation is to use an extended Kalman filter (EKF), and we use a modified EKF [38] that has improved convergence conditions when the system is linear in the state for fixed parameters and *vice versa*. The reason for its improved convergence is the use of a Luenberger observer to estimate the state and an EKF to estimate the parameters [38], instead of an EKF for both the state and parameters.

In the quadrotor, the parameters of the oracle correspond to a linearization of the unmodeled dynamics about operating points that vary as the quadrotor moves through the state space; consequently, the parameters will change. The EKF of [38] requires the addition of a noise term in order to handle this.

Let  $P_2[0] \in \mathbb{R}^{10 \times 12}$  be the initial cross-covariance matrix between the initial state estimate  $\hat{x}[0]$  and the initial parameter estimate  $\beta[0]$ , and  $P_3[0] \in \mathbb{R}^{12 \times 12}$  be the initial covariance matrix of the parameter estimates. Define  $\Xi \in \mathbb{R}^{5 \times 5}$  to be the covariance of the noise in the measurements  $\epsilon[n]$ . In a change to [38], we model the parameters as  $\beta[n+1] = \beta[n] + \mu[n]$ , where  $\mu[n]$  is noise and  $\Upsilon \in \mathbb{R}^{12 \times 12}$  is the covariance of the noise.

Let  $\hat{K}$  be a feedback matrix such that  $A + F - \hat{K}C$  is exponentially stable for all  $\beta$  that satisfy the appropriate bounds (17). This can be verified using the Bauer-Fike theorem [39], for example. The dual EKF is defined by the following equations

$$\begin{aligned} \hat{x}[n+1] &= (A+F)\hat{x}[n] + (B+H)u[n] \\ &\quad + (k+z) + \hat{K}\zeta[n] \end{aligned} \quad (19)$$

$$\zeta[n] = y[n] - C\hat{x}[n] \quad (20)$$

$$L[n] = P_2[n]'C'\Xi^{-1} \quad (21)$$

$$P_2[n+1] = (A+F)P_2[n] + M[n]P_3[n] - \hat{K}\Xi L[n]' \quad (22)$$

$$P_3[n+1] = P_3[n] - L[n]\Xi L[n]' - \delta P_3[n]P_3[n]' + \Upsilon \quad (23)$$

$$M[n] = \frac{\partial}{\partial \beta}(F\hat{x}[n] + Hu[n] + z), \quad (24)$$

where  $\delta > 0$  is a tuning parameter. Furthermore, the parameter update is given by

$$\beta[n+1] = \text{bound}(\beta[n] + L[n]\zeta[n]), \quad (25)$$

where the function  $\text{bound}(\cdot)$  is a function that clips a vector so that it satisfies the constraints in (17) on  $\beta$ .

## VI. OPTIMIZATION FORMULATION

The LBMPC technique in Sect. IV-C is a quadratic program (QP), and this can be numerically solved using a variety of algorithms. Our current implementation uses the LSSOL solver [40], which is a dense active set solver. We decided to use this solver because it is what we are currently most familiar with, and future extensions involve using solvers that are designed for MPC [41], [42].

Because LSSOL is a dense solver, reducing the number of variables in the optimization problem will improve computation speed. This can be done by making substitutions for the variables  $\bar{x}$ ,  $\tilde{u}$ , and  $\tilde{x}$ . For the quadrotor, when using a horizon of  $N = 15$ , this leads to a reduction from 363 variables to 33 variables. For example, the variables  $\bar{x}$  can be removed by noting that

$$\bar{x}[m+i] = (A+B\bar{K})^i \hat{x}[m] + \sum_{j=0}^{i-1} (A+B\bar{K})^{i-j} (Bc[m+i-1]+k), \quad (26)$$

where  $\hat{x}[m]$  is the state estimate and is constant. Similar reductions can be used to remove  $\tilde{x}$  and  $\tilde{u}$ .

LSSOL provides another opportunity for improving solution speed, and this is based on the optimization formulation. The LSSOL solver can also handle the problem when it is formulated as a constrained least squares problem

$$\min_{\mu} \|W_1 \mu - v_1\|_2^2 \quad (27)$$

$$\text{s.t. } b_L \leq U\mu \leq b_U. \quad (28)$$

Numerical tests show that this formulation requires two-thirds of the computation time on the onboard computer of the quadrotor as compared to solving the standard formulation of a QP

$$\min_{\mu} \mu' W_2 \mu + v_2' \mu \quad (29)$$

$$\text{s.t. } b_L \leq U\mu \leq b_U. \quad (30)$$

## VII. EXPERIMENTS

We describe two sets of experiments that demonstrate the features that LBMPC provides, and we run the MPC at 40 Hz sampling rate with horizon  $N = 15$ . The first experiment is two step inputs applied in relatively quick succession. With the quadrotor at  $x_1 = -1$  m, the desired position is set to  $x_1 = 1$  m for 3.5 s and then set back to  $x_1 = -1$  m. A comparison of the performance of LBMPC with linear MPC is shown in Fig. 1: The reference command is the dotted blue line, the LBMPC response is the dashed red line, and the linear MPC response is the solid green line. The LBMPC leads to less overshoot on the quadrotor than the linear MPC, though the rise times are roughly the same.

A second experiment demonstrated the robustness of LBMPC on the quadrotor. The situation in which the learning performs poorly and identifies incorrect parameters was tested by making the EKF unstable: This was achieved by significantly increasing the noise process covariance of the parameters  $\Upsilon$  from the nominally chosen values. The results of the experiment are shown in Fig. 2. The quadrotor was

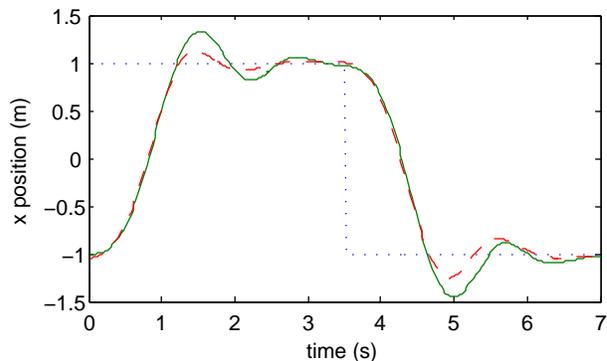


Fig. 1. With the quadrotor at  $x_1 = -1$  m, the desired position is set to  $x_1 = 1$  m for 3.5 s and then set back to  $x_1 = -1$  m. A comparison of the performance of LBMPC with linear MPC is shown in Fig. 1: The reference command is the dotted blue line, the LBMPC response is the dashed red line, and the linear MPC response is the solid green line.

commanded to maintain a height of 0.85 m, but because of the mis-learning the quadrotor makes a drastic fall in altitude; however, the LBMPC demonstrates robustness by adjusting the control action, and consequently the quadrotor does not hit the ground, which is at 0 m.

## VIII. CONCLUSION

We have presented a modification of LBMPC that is tailored to deal with the challenges associated with achieving a real-time implementation on a quadrotor helicopter. One future direction motivated by this work is an exploration of numerical optimization algorithms that are optimized for MPC [41], [42].

## REFERENCES

- [1] A. Aswani, H. Gonzales, S. Sastry, and C. Tomlin, "Provably safe and robust learning-based model predictive control," arXiv:1107.2487v1. [Online]. Available: <http://arxiv.org/abs/1107.2487v1>
- [2] A. Aswani, N. Master, J. Taneja, D. Culler, and C. Tomlin, "Reducing transient and steady state electricity consumption in HVAC using learning-based model-predictive control," *Proceedings of the IEEE*, 2011.
- [3] K. Zhou and J. Doyle, *Essentials of robust control*. Prentice Hall, 1998.
- [4] S. Skogestad and I. Postlethwaite, *Multivariable feedback control: analysis and design*. John Wiley, 2005.
- [5] S. S. Sastry and M. Bodson, *Adaptive Control: Stability, Convergence, and Robustness*. Prentice-Hall, 1989.
- [6] S. S. Sastry and A. Isidori, "Adaptive control of linearizable systems," *IEEE Transactions on Automatic Control*, vol. 34, no. 11, pp. 1123–1131, 1989.
- [7] K. Åström and B. Wittenmark, *Adaptive control*. Addison-Wesley, 1995.
- [8] C. Anderson, P. Young, M. Buehner, J. Knight, K. Bush, and D. Hittle, "Robust reinforcement learning control using integral quadratic constraints for recurrent neural networks," *IEEE Transactions on Neural Networks*, vol. 18, no. 4, pp. 993–1002, 2007.
- [9] R. Tedrake, "LQR-trees: Feedback motion planning on sparse randomized trees," in *Robotics: Science and Systems*, 2009, pp. 17–24.
- [10] P. Abbeel, A. Coates, and A. Ng, "Autonomous helicopter aerobatics through apprenticeship learning," *International Journal of Robotics Research*, vol. 29, no. 13, pp. 1608–1639, 2010.
- [11] L. Chisci, J. Rossiter, and G. Zappa, "Systems with persistent disturbances: predictive control with restricted constraints," *Automatica*, vol. 37, pp. 1019–1028, 2001.

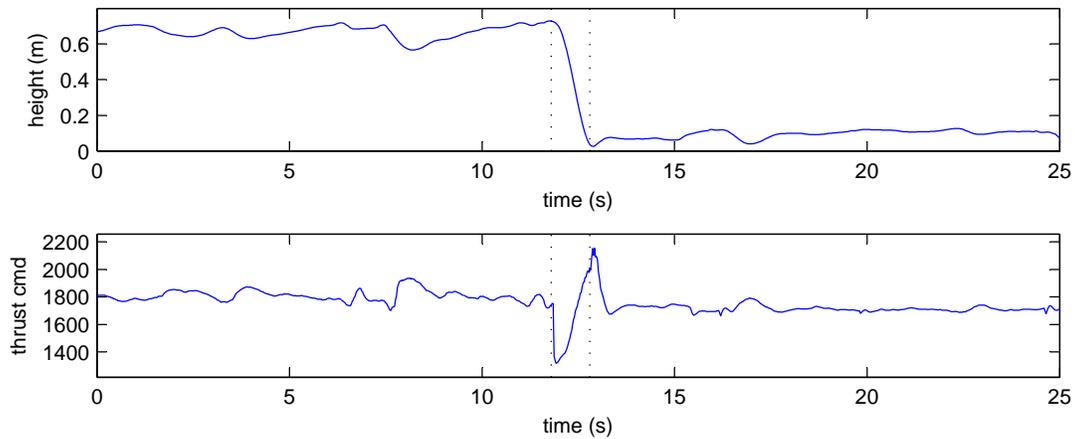


Fig. 2. The quadrotor was commanded to maintain a height of 0.85 m, but because of mis-learning the quadrotor makes a drastic fall in altitude. Fortunately, the robustness of the LBMPC compensates and ensures that the quadrotor does not hit the ground at 0 m. The instances at which the quadrotor begins and ends its rapid descent are marked by black, vertical dashed lines. The height of the quadrotor is shown on the top plot, and the thrust command generated by the LBMPC is shown on the bottom.

- [12] W. Langson, I. Chrysochoos, S. Raković, and D. Mayne, “Robust model predictive control using tubes,” *Automatica*, vol. 40, no. 1, pp. 125–133, 2004.
- [13] D. Limon, I. Alvarado, T. Alamo, and E. Camacho, “MPC for tracking piecewise constant references for constrained linear systems,” *Automatica*, vol. 44, no. 9, pp. 2382–2387, 2008.
- [14] A. Ferramosca, D. Limon, I. Alvarado, T. Alamo, and E. Camacho, “MPC for tracking with optimal closed-loop performance,” *Automatica*, vol. 45, no. 8, pp. 1975–1978, 2009.
- [15] F. Borelli, A. Bemporad, and M. Morari, *Constrained Optimal Control and Predictive Control for linear and hybrid systems*, 2009, in preparation.
- [16] D. Limon, I. Alvarado, T. Alamo, and E. Camacho, “Robust tube-based MPC for tracking of constrained linear systems with additive disturbances,” *Journal of Process Control*, vol. 20, no. 3, pp. 248–260, 2010.
- [17] 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*, vol. 43, no. 2, pp. 301–308, 2007.
- [18] A. Aswani, P. Bickel, and C. Tomlin, “Statistics for sparse, high-dimensional, and nonparametric system identification,” in *International Conference on Robotics and Automation*, 2009.
- [19] —, “Regression on manifolds: Estimation of the exterior derivative,” *Annals of Statistics*, vol. 39, no. 1, pp. 48–81, 2011.
- [20] E. Gilbert and K. Tan, “Linear systems with state and control constraints: the theory and application of maximal output admissible sets,” *IEEE Transactions on Automatic Control*, vol. 36, no. 9, pp. 1008–1020, 1991.
- [21] S. Rakovic and M. Baric, “Parameterized robust control invariant sets for linear systems: Theoretical advances and computational remarks,” *Automatic Control, IEEE Transactions on*, vol. 55, no. 7, pp. 1599–1614, 2010.
- [22] E. Asarin, O. Bournez, T. Dang, and O. Maler, “Approximate reachability analysis of piecewise-linear dynamical systems,” in *Hybrid Systems: Computation and Control 2000*, 2000, pp. 20–31.
- [23] R. Ghosh and C. Tomlin, “Symbolic reachable set computation of piecewise affine hybrid automata and its application to biological modelling: Delta-Notch protein signalling,” *Systems Biology*, vol. 1, no. 1, pp. 170–183, Jun. 2005.
- [24] A. Aswani and C. Tomlin, “Reachability algorithm for biological piecewise-affine hybrid systems,” in *Hybrid Systems: Computation and Control 2007*, 2007, pp. 633–636.
- [25] S. Raković, E. Kerrigan, D. Mayne, and J. Lygeros, “Reachability analysis of discrete-time systems with disturbances,” *IEEE Transactions on Automatic Control*, vol. 51, no. 4, pp. 546–561, 2006.
- [26] A. Chutinan and B. H. Krogh, “Verification of polyhedral-invariant hybrid automata using polygonal flow pipe approximations,” in *HSCC*, 1999, pp. 76–90.
- [27] E. Asarin, T. Dang, and A. Girard, “Reachability analysis of nonlinear systems using conservative approximation,” in *Hybrid Systems: Computation and Control 2003*, 2003, pp. 20–35.
- [28] O. Stursberg and B. Krogh, “Efficient representation and computation of reachable sets for hybrid systems,” in *Hybrid Systems: Computation and Control 2003*, 2003, pp. 482–497.
- [29] I. Mitchell, A. Bayen, and C. Tomlin, “A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games,” *IEEE Transactions on Automatic Control*, vol. 50, no. 7, pp. 947–957, 2005.
- [30] P. Bouffard, A. Aswani, and C. Tomlin, “Learning-based model predictive control on a quadrotor: Onboard implementation and experimental results,” 2011, submitted.
- [31] P. Bouffard. (2011) starmac-ros-pkg ros repository. [Online]. Available: <http://www.ros.org/wiki/starmac-ros-pkg>
- [32] R. Schneider, *Convex bodies: the Brunn-Minkowski theory*. Cambridge University Press, 1993.
- [33] I. Kolmanovsky and E. Gilbert, “Theory and computation of disturbance invariant sets for discrete-time linear systems,” *Mathematical Problems in Engineering*, vol. 4, pp. 317–367, 1998.
- [34] H. R. Tiwary, “On the hardness of computing intersection, union and minkowski sum of polytopes,” *Discrete and Computational Geometry*, vol. 40, pp. 469–479, September 2008.
- [35] A. Girard and C. Guernic, “Zonotope/hyperplane intersection for hybrid systems reachability analysis,” in *Proceedings of the 11th international workshop on Hybrid Systems: Computation and Control*, ser. HSCC ’08. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 215–228.
- [36] G. Grimm, M. J. Messina, S. E. Tuna, and A. R. Teel, “Examples when nonlinear model predictive control is nonrobust,” *Automatica*, vol. 40, no. 10, pp. 1729–1738, 2004.
- [37] F. Albertini and D. D’Alessandro, “Observability and forward-backward observability of discrete-time nonlinear systems,” *Mathematics of Control, Signals, and Systems*, vol. 15, pp. 275–290, 2002.
- [38] L. Ljung, “Asymptotic behavior of the extended kalman filter as a parameter estimator for linear systems,” *Automatic Control, IEEE Transactions on*, vol. 24, no. 1, pp. 36–50, feb 1979.
- [39] F. Bauer and C. Fike, “Norms and exclusion theorems,” *Numerische Mathematik*, vol. 2, pp. 137–141, 1960.
- [40] P. Gill, S. Hammarling, W. Murray, M. Saunders, and M. Wright, *LSSOL 1.0 User’s Guide*, 1986.
- [41] H. Ferreau, H. Bock, and M. Diehl, “An online active set strategy to overcome the limitations of explicit mpc,” *International Journal of Robust and Nonlinear Control*, vol. 18, no. 8, pp. 816–830, 2008.
- [42] Y. Wang and S. Boyd, “Fast model predictive control using online optimization,” *IEEE Transactions on Control Systems Technology*, vol. 18, no. 2, pp. 267–278, 2010.