Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

ARTICLE TYPE

Robust adaptive model predictive control for safe and high accuracy tracking in the presence of model errors †

Karime Pereida* | Angela P. Schoellig

[1]Institute for Aerospace Studies, University of Toronto, Ontario, Canada

Correspondence
*Corresponding author Karime Pereida, This is sample corresponding address. Email: karime.pereida@robotics.utias.utoronto.ca

Present Address
This is sample for present address text this is sample for present address text

Summary

Numerous systems require controllers that guarantee high performance since they are deployed in unknown and dynamic environments where they are subject to disturbances, unmodeled dynamics, and parametric uncertainties. Adaptive controllers enable systems to compensate for disturbances and unmodeled dynamics based on state measurements. In this paper, we propose a robust adaptive model predictive controller for safe and high accuracy trajectory tracking in the presence of model uncertainties. The proposed approach combines robust model predictive control (MPC) with an underlying discrete time ℓ_1 adaptive controller. The ℓ_1 adaptive controller forces the system to behave close to a linear reference model despite the presence of disturbances and unmodeled dynamics. The true dynamics of the underlying adaptive system may deviate from the linear reference model. In this work we prove this deviation is bounded. The robust MPC accounts for this modeling error and computes the input that minimizes the tracking error. In experiments we focus in robotic applications, in particular on a quadrotor. We show that the proposed robust adaptive MPC enables the quadrotor to achieve high accuracy trajectory tracking and successfully avoid obstacles avoidance through aggressive maneuvers that an adaptive MPC is not able to avoid.

KEYWORDS:
robust adaptive control, keyword2, keyword3, keyword4

1 | INTRODUCTION

Control systems for unstructured environments are required to achieve high overall performance in the presence of unknown disturbances, changing dynamics and parametric uncertainties. To operate in these environments controllers must adapt quickly to changing conditions and be robust to unknown disturbances. Adaptive controllers enable systems to compensate for disturbances and unmodeled dynamics based on state measurements. Additionally, the controllers are required to provide fast update rates to achieve better tracking performance. In this work we focus in robotic systems operating in unstructured environments. Robotic applications in these unstructured and changing environments include autonomous driving, assistive robotics, and unmanned aerial vehicle applications. In such robotic applications small changes in the environmental conditions may significantly deteriorate the performance and cause instability in traditional, model-based controllers (see [1] and [2]).

†This is an example for title footnote.
[0]**Abbreviations:** ?

In this work we propose a robust adaptive model predictive controller (MPC) for safe and high accuracy tracking in the presence of model errors. We propose to combine an underlying ℓ_1 adaptive controller with a *robust* MPC. The ℓ_1 adaptive controller forces a system to behave close to a specified linear reference model despite the presence of parametric uncertainties and unknown disturbances. We show that the ℓ_1 controlled system may deviate from the linear reference model, but this deviation is bounded and can be interpreted as a bound on the modeling error of the system. The robust MPC explicitly takes into account the bound on the modeling error, the linear reference model and state and input constraints to calculate an input that improves the tracking performance of the underlying ℓ_1 controlled system. Explicitly taking into account the modeling error enables the proposed robust adaptive MPC to successfully complete tasks that adaptive MPC fails to achieve. Note that in this work use $\mathcal{L}_1$ to refer to the continuous time formulation and ℓ_1 to refer to the discrete time formulation.

1.1 | Related Work

$\mathcal{L}_1$ adaptive control generates a control signal that forces a system to track a reference trajectory according to a specified linear reference model despite the presence of unknown disturbances and model uncertainties. The architecture of the $\mathcal{L}_1$ adaptive controller decouples the estimation loop from the control law. Consequently, there is guaranteed robustness in the presence of fast adaptation, without introducing persistence of excitation, gain scheduling in the controller parameters or high-gain feedback[3]. $\mathcal{L}_1$ adaptive control has been extensively developed in the continuous time[3,4,5] and has been applied to different platforms such as an underwater vehicle[6], a tailless fighter aircraft, in simulation[7], bipedal robots, in simulation[8], and a quadrotor, hexacopter, and octocopter[9]. However, $\mathcal{L}_1$ adaptive control has not been developed as extensively in the discrete time[10,11].

Model predictive control has been widely employed to control constrained systems and an extensive literature on the subject exists[12,13,14]. MPC solves a finite horizon optimal control problem at each time step to calculate a control sequence that minimizes a given objective function according to a system model and constraints. A controller is said to be robust when stability is maintained and performance specifications are met for a specified range of model variations and a class of noise signals[15]. The standard implementation of MPC using a nominal model of the system dynamics exhibits nominal robustness to small disturbances[16] and may be insufficient when larger errors are present. The term *nominal* MPC is used in this work to describe the standard implementation of MPC[12]. In order to achieve robustness to larger disturbances numerous MPC methods have been considered. The simplest is to ignore the disturbances and rely on the inherent robustness of deterministic MPC applied to the nominal system[16]. In these approaches the so-called spread of predicted trajectories resulting from disturbances is not acknowledged.

In order to acknowledge disturbances open- and closed-loop MPC methods were developed. *Open-loop* methods calculate the control action that is safe enough to cope with the effect of the worst disturbance realization[17]. These controllers can be very conservative as the disturbance effect can be mitigated by feedback in the actual operation. For this reason, *closed-loop* prediction is introduced in[18]. The work in[19] proposed a feedback MPC where the decision variable is a policy which is a sequence of control laws. Nevertheless, determining a control policy can be prohibitively difficult and simplifying approximations have been proposed[20,21,22]. A popular branch of robust MPC is based on the idea of tubes, as presented in[21] for linear systems. In[23] a robust MPC for nonlinear systems based on tubes was proposed. The proposed scheme achieves good tracking performance in simulation. However, it presents a challenge to determine the tightened input constraints and the added computational cost of solving a nonlinear optimization problem. Our proposed robust adaptive MPC approach has the potential to control a nonlinear system with a linear MPC. A survey on robust MPC can be found in[24].

One way to handle model uncertainties is to develop robust MPC methods, as described above. Another way to handle model uncertainties is through adaptive MPC. In[25] an adaptive MPC for a class of constrained linear systems, which estimates parameters online is presented. However, parameter adaptation depends on state excitation. The performance of the proposed approach is shown in simulation. The work in[26] proposed an adaptive MPC for linear systems with parametric uncertainties. At each timestep, the adaptation law updates the estimated model parameters and uses the updated model to solve an MPC problem. However, additional constraints have to be added to the MPC problem in order to guarantee an estimation error bound. These results are shown in a simulation example. It is difficult to guarantee the fulfillment of constraints in the presence of an adaptive mechanism. Further, it is difficult to guarantee feasibility and stability theoretically when an adaptation is introduced to MPC[25].

Robust adaptive control aids adaptive controllers to achieve robustness. A small number of robust adaptive MPC schemes have been proposed. The work by[27] proposes a robust adaptive MPC for building climate control. In this linear MPC implementation, the uncertainty of the model is assumed to be in a given parameter set. At each timestep *(i)* a recursive set membership identification algorithm tracks the set of all possible model parameters that are consistent with the initial assumptions and the

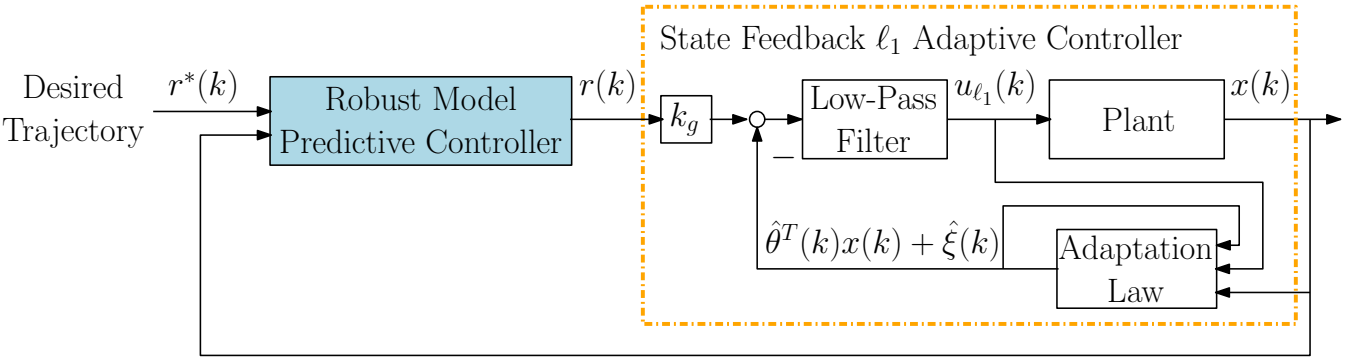


FIGURE 1 Proposed framework to achieve high performance control in the presence of modeling errors. The discrete time state feedback $\mathscr{L}_1$ adaptive controller makes the system behave in a predefined, repeatable way. The robust model predictive control improves the performance at each time step.

data collected thus far, and *(ii)* a robust MPC minimizes the cost function while guaranteeing satisfaction of constraints for all parameters in the feasible parameter set. In order to reduce computational complexity (which could grow linearly with time), further assumptions have to be made on the parameters, e.g. that the parameters lie in a larger set. The latter could incur in a conservative controller. In[28] a robust adaptive MPC for autonomous lane keeping was proposed. The linear model is assumed to accurately model the system with the exception of uncertainty introduced by discretization and noise components. The knowledge in the steering offset lies in the feasible parameter set, which is updated at every time step (the adaptive component of this framework). The feasible parameter set is used by a robust MPC to compute an input that is feasible for all the possible steering offsets in the feasible parameter set. However, the algorithm must be reset if there is any variation in the uncertainty over time is suspected. Simulation results show that the proposed approach outperforms nominal MPC.

In previous work[29] we proposed an adaptive MPC that combines an underlying $\mathscr{L}_1$ adaptive controller with a nominal MPC. In this way, adaptation is decoupled from the optimization problem. The $\mathscr{L}_1$ adaptive controller forces a system to behave close to a specified linear reference model despite the presence of parametric uncertainties and unknown disturbances. The nominal MPC uses this linear reference model to improve the trajectory tracking performance of the $\mathscr{L}_1$ controlled system. This architecture achieves high tracking performance of a possibly nonlinear system at the computational cost of a linear MPC even when disturbances are applied. Experiments on a drone showed that the adaptive MPC approach is able to achieve high accuracy trajectory tracking even when wind disturbances are applied to the vehicle. However, this approach assumes that the linear model exactly represents the underlying adaptive system, which may degrade the performance of the controller when modeling errors are present.

1.2 | Our Contributions

In order to deal with model uncertainties and further improve performance, we propose a robust and adaptive MPC and make the following contributions *(i)* introduce a modified discrete time state feedback $\mathscr{L}_1$ adaptive controller, *(ii)* provide stability and performance proofs, and *(iii)* show the performance of the proposed approach with experiments on a quadrotor. Our proposed robust adaptive MPC decouples adaptation from robustness with two main components: *(i)* an underlying discrete time state feedback $\mathscr{L}_1$ adaptive controller (orange dashed box in Fig. 1), and *(ii)* a robust MPC (blue box in Fig. 1). The $\mathscr{L}_1$ adaptive controller makes the system behave close to a linear reference model despite disturbances, which the control designer specifies. The real system state may deviate from the linear reference model state, which would cause a modeling error. In this work we show that this deviation is upper bounded and realizable. The upper bound and the linear reference model are used in a robust MPC framework to calculate the input that minimizes the tracking error of the $\mathscr{L}_1$ controlled system. Intuitively, the modeling error -characterized by the upper bound on the difference between the linear reference model and the $\mathscr{L}_1$ controlled system- can be viewed as an additive disturbance affecting the linear reference model. Recall that robust MPC frameworks such as the ones presented by[21] and[22], are proposed for linear systems with bounded additive disturbances; hence, they can be used in our framework.

The proposed robust adaptive MPC has the following improvements with respect to [30]: *(i)* the $\mathscr{l}_1$ adaptive controller is state feedback instead of output feedback, *(ii)* the $\mathscr{l}_1$ adaptive controller is formulated in the discrete time, *(iii)* introduces the bound on the modeling error, and *(iv)* explicitly accounts for the modeling error with a robust MPC. The $\mathscr{l}_1$ adaptive controller is state feedback to attain performance bounds that are realizable. The output feedback $\mathcal{L}_1$ adaptive controller used in [30] describes the system with $A(s)$, which is strictly-proper but *unknown*. This transfer function is found in all the performance bounds, which makes them difficult to calculate. Unlike output feedback, state feedback $\mathcal{L}_1$ adaptive controllers decouple the system uncertainties in the performance bounds. Hence, it is possible to calculate these bounds. The $\mathscr{l}_1$ adaptive controller was designed in the discrete time to improve robustness in the implementation. Typically, a discretized version of the continuous time $\mathcal{L}_1$ adaptive control is implemented [4,29,30,31]. However, if the sampling time is increased, the discretized version can become unstable, as shown in [32]. Additionally, MPC frameworks are generally formulated for discrete time systems. In this work we introduce the uniform bound of the difference between the state of the linear reference model and the actual system and explicitly take it into account in a robust MPC. We show in experimental results with a quadrotor that the proposed robust adaptive MPC has a marginally better performance than an adaptive MPC (no robustness in the MPC) for trajectory tracking showing that the robustness added to the MPC does not negatively affect the performance of the controller. Moreover, we show that the robust adaptive MPC is able to complete tasks that the adaptive MPC is not able to complete, because it takes into account the modeling errors with the robust MPC.

2 | PROBLEM FORMULATION

The objective of this work is to achieve safe and high accuracy trajectory tracking in the presence of modeling errors. Consider the class of discrete time systems with matched constant uncertainties (see Fig. 1) whose dynamics can be described by the following system:

$$\begin{aligned} &x(k+1) = A(k) + b_m(u(k) + \theta^T x(k) + \xi)\,,\ x(0) = x_0\,,\\ &y(k) = c_m^T x(k)\,, \end{aligned} \tag{1}$$

where $x(k) \in \mathbb{R}^n$ is the system state vector (assumed to be measured); $u(k)$ is the control signal; b_m, $c_m \in \mathbb{R}^n$ are known constant vectors; A_m is the known $n \times n$ matrix, with $(A_m\,, b_m)$ controllable; θ is a vector of *unknown* parameters, and ξ is a disturbance both belonging to the compact convex sets $\theta \in \Theta \subset \mathbb{R}^n$ and $\xi \in \Xi \subset \mathbb{R}$; and $y(k) \in \mathbb{R}$ is the regulated output.

Assumption 1. The set Θ is compact and convex. Each dimension i is bounded by $L_{\Theta_{i,\min}} \leq \theta_i \leq L_{\Theta_{i,\max}}$ and the set Θ includes all the values that satisfy this criterion.

The parameter ξ is bounded by $L_{\xi,\min} \leq \xi \leq L_{\xi,\max}$ where Ξ is the set that includes all the values that satisfy this criterion.

Generally, the unknown parameters model physical attributes of the system such as mass, moment of inertia, friction, among others. For this reason we assume these parameters lie in a compact, convex set.

The system is required to accurately track a desired trajectory $y^*(k)$ defined over a finite number of steps $N < \infty$ and is assumed to be feasible with respect to the true dynamics of the $\mathscr{l}_1$ controlled system (orange dashed box in Fig. 1). We introduce the lifted representation [33] for the desired trajectory $\mathbf{y}^* = (y^*(1), \dots, y^*(N))$, and the output of the plant $\mathbf{y} = (y(1), \dots, y(N))$. The tracking performance criterion J is defined as:

$$J \triangleq \min_{\mathbf{e}} \mathbf{e}^T \mathbf{Q} \mathbf{e}\,, \tag{2}$$

where $\mathbf{e} = \mathbf{y} - \mathbf{y}^*$ is the tracking error and $\mathbf{Q}$ is a positive definite matrix. The goal is to take into account the disturbances in the system to improve the tracking performance at each time step.

3 | DISCRETE TIME STATE FEEDBACK $\mathscr{l}_1$ ADAPTIVE CONTROL

In this Section we present a modified discrete time $\mathscr{l}_1$ adaptive controller along with proofs for stability and performance guarantees. Recall that the $\mathscr{l}_1$ adaptive controller brings two important features for the overall robust adaptive MPC. First, it forces the controlled system to behave close to a linear model despite the presence of disturbances. Second, it provides an upper bound for the modeling error that arises between the controlled system and the linear reference model. The discrete time state feedback $\mathscr{l}_1$ adaptive controller is inspired by [10]; however, in this work we *(i)* extend the system to include a disturbance ξ (see (1)), *(ii)* modify the adaptation law to avoid division by zero, and *(iii)* prove the uniform upper bound on the difference

between the $\mathscr{L}_1$ controlled system and the linear reference model. There are two main design challenges for a discrete time state feedback $\mathscr{L}_1$ adaptive controller: *(i)* the definition of an adaptation law, and *(ii)* the guarantee that the estimated parameters remain in a given set.

The first challenge is the definition of an adaptation law that allows us to prove stability and show performance guarantees. The adaptation law is fundamentally a parameter estimation scheme. Parameter estimation for deterministic discrete time systems has been widely studied[34,35]. Some of these algorithms have been used in discrete time model reference adaptive control (MRAC) frameworks[36] and discrete time $\mathscr{L}_1$ adaptive controllers[10]. The second challenge is guaranteeing that the estimated parameters remain in a given set. The latter facilitates the stability analysis and derivation of performance guarantees for the $\mathscr{L}_1$ adaptive control, as will be shown in this Section. To guarantee that the estimated parameters remain in a given set, the continuous time $\mathcal{L}_1$ adaptive controllers use the projection operator[3]. To derive similar performance guarantees in discrete time, the adaptation law used for the $\mathscr{L}_1$ adaptive controller must also include a projection operator. For this reason, the adaptation law in this work is based on the orthogonalized projection algorithm described in[35].

Considering the class of systems described in (1), the control objective is to design an adaptive state feedback control signal $u_{\mathscr{L}_1}(k)$ such that the system output $y(k)$ tracks the desired piecewise, bounded reference signal $r(k)$ with quantifiable transient and steady-state performance bounds. The output is required to track the reference according to a specified stable reference system.

3.1 | Discrete Time State Feedback $\mathscr{L}_1$ Adaptive Control Architecture

Our discrete time state feedback $\mathscr{L}_1$ adaptive controller designs a control signal such that the output $y(k)$ tracks a reference signal $r(k)$ according to a specified stable reference system $A_m(k)$. The discrete time state feedback $\mathscr{L}_1$ adaptive controller proposed in this work is shown in an orange dashed box in Fig. 1. The individual components of the $\mathscr{L}_1$ architecture are introduced below. This architecture is inspired by the architecture presented by[10].

Consider the following control structure:

$$u(k) = u_m(k) + u_{\mathscr{L}_1}(k)\,, \quad u_m(k) = -k_m^T x(k)\,, \tag{3}$$

where $k_m \in \mathbb{R}^n$ renders $A_m \triangleq A - bk_m^T$ stable, while $u_{\mathscr{L}_1}(k)$ is the adaptive component, which will be defined shortly. The static feedback gain k_m leads to the following partially closed-loop system:

$$\begin{aligned} &x(k+1) = A_m x(k) + b_m(u(k) + \theta^T x(k) + \xi)\,,\ x(0) = x_0\,,\\ &y(k) = c_m^T x(k)\,. \end{aligned} \tag{4}$$

The equations that describe the implementation of the discrete time state feedback $\mathscr{L}_1$ adaptive controller are:

Adaptation Law:

We the following projection algorithm estimator[34,35] that avoids division by zero:

$$\hat{\rho}(k+1) = \hat{\rho}(k) + \frac{\begin{bmatrix} x(k) \\ 1 \end{bmatrix} \left[b_0^T (x(k+1) - A_m x(k) - b_m u_{\mathscr{L}_1}(k)) - \hat{\rho}^T(k) \begin{bmatrix} x(k) \\ 1 \end{bmatrix} \right]}{1 + x^T(k)x(k)}\,,\quad \hat{\rho}(0) = \hat{\rho}_0 \in \Theta \times \Xi\,, \tag{5}$$

where $b_0 \triangleq \dfrac{b_m}{b_m^T b_m}$ is a constant vector. If $\hat{\rho}(k+1) \in \Theta \times \Xi$, then continue, else:

(i) orthogonally project $\hat{\rho}(k+1)$ on the boundary of $\Theta \times \Xi$ to yield $\hat{\rho}'(k+1)$:

$$\begin{aligned} \hat{\theta}_i'(k+1) &= \begin{cases} L_{\Theta_{i,\min}} & \text{if } \theta_i(k+1) \le L_{\Theta_{i,\min}} \\ L_{\Theta_{i,\max}} & \text{if } \theta_i(k+1) \ge L_{\Theta_{i,\max}} \\ \hat{\theta}_i(k+1) & \text{otherwise}\,, \end{cases} \\ \hat{\xi}'(k+1) &= \begin{cases} L_{\xi,\min} & \text{if } \hat{\xi}(k+1) \le L_{\xi,\min} \\ L_{\xi,\max} & \text{if } \hat{\xi}(k+1) \ge L_{\xi,\max} \\ \hat{\xi}(k+1) & \text{otherwise}\,. \end{cases} \end{aligned} \tag{6}$$

Since $\hat{\rho}'(k+1)$ is an orthogonal projection of $\hat{\rho}(k+1)$ onto $\Theta\times\Xi$ and since $\rho \in \Theta\times\Xi$, then as for the projection algorithm of[35]:

$$\|\hat{\rho}'(k+1) - \rho\|^2 \leq \|\hat{\rho}(k+1) - \rho\|^2 . \tag{7}$$

The additional steps in the adaptation law guarantee that the estimate $\hat{\rho}(k)$ remains in the set $\Theta\times\Xi$. The latter is needed for the Lyapunov analysis of the system. It is of particular importance to note that since the projection is orthogonal, then (7) holds. This allows us to develop the analysis only for $\tilde{\rho}(k) \triangleq \hat{\rho}(k) - \rho$, since $\tilde{\rho}'(k) \triangleq \hat{\rho}'(k) - \rho$ is smaller.

(ii) Finally, assign

$$\hat{\rho}(k+1) = \hat{\rho}'(k+1) . \tag{8}$$

Control law:

The z-transform of the adaptive control signal is:

$$u(z) = -C(z)(\hat{\eta}(z) - k_g r(z)) , \tag{9}$$

where $r(z)$ and $\hat{\eta}(z)$ are the z-transforms of command input $r(k)$ and $\hat{\eta}(k) = \hat{\rho}^T(k)\begin{bmatrix} x(k) \\ 1 \end{bmatrix}$, respectively, $k_g \triangleq (c_m^T(I_n - A_m)^{-1}b_m)^{-1}$, and $C(z)$ is a bounded-input, bounded-output (BIBO) stable, strictly-proper, discrete time transfer function with gain $C(1) = 1$, and its state-space realization assumes zero initialization.

The discrete time ℓ_1 adaptive controller is defined via (5) – (9) with $C(z)$ verifying the following ℓ_1-norm condition:

$$\lambda_\theta \triangleq \|G(z)\|_{\ell_1} L_\theta < 1 , \quad \lambda_\xi \triangleq \|G(z)\|_{\ell_1} L_\xi < \infty , \tag{10}$$

where

$$\begin{aligned} &G(z) \triangleq H(z)(1 - C(z)) , \; H(z) \triangleq (zI_n - A_m)^{-1}b_m , \\ &L_\theta \triangleq \max_{\theta\in\Theta} \|\theta\|_1 , \qquad L_\xi \triangleq \max_{\xi\in\Xi} \|\xi\|_1 . \end{aligned} \tag{11}$$

3.2 | Lyapunov Stability and Performance Bounds

In this subsection we show Lyapunov stability of the ℓ_1 controlled system and its performance bounds with respect to a closed-loop reference system. Finally, we introduce a performance bound of the ℓ_1 controlled system with respect to an ideal reference model. The ideal reference model can be used to described the ℓ_1 controlled system and the performance bound can be used as a bound on the modeling error. This subsection is divided in three parts *(i)* background theory required for subsequent proofs, *(ii)* Lyapunov stability, and *(iii)* performance bounds with respect to an ideal linear system.

3.2.1 | Background

The following results are needed to show Lyapunov stability and performance bounds. We present them next to improve readability of the main results of the paper.

We use a state predictor and its corresponding error dynamics as tools to show Lyapunov stability and performance bounds of the ℓ_1 controlled system. We consider the following state predictor:

$$\begin{aligned} &\hat{x}(k+1) = A_m\hat{x}(k) + b_m\left(u_{\ell_1}(k) + \hat{\rho}^T(k)\begin{bmatrix} x(k) \\ 1 \end{bmatrix}\right) , \; \hat{x}(0) = x_0 , \\ &\hat{y}(k) = c_m^T\hat{x}(k) , \end{aligned} \tag{12}$$

where $\hat{x}(k) \in \mathbb{R}^n$ is the state predictor and $\hat{\rho}(k) \triangleq \begin{bmatrix} \hat{\theta}(k) \\ \hat{\xi}(k) \end{bmatrix} \in \mathbb{R}^{n+1}$ is the adaptive estimate of parameters θ and ξ.

Next, we derive the error dynamics of the state predictor from (4) and (12):

$$\tilde{x}(k+1) = A_m\tilde{x}(k) + b_m(\tilde{\theta}^T(k)x(k) + \tilde{\xi}(k)) , \quad \tilde{x}(0) = 0 , \tag{13}$$

where $\tilde{x}(k) = \hat{x}(k) - x(k)$, $\tilde{\theta}(k) \triangleq \hat{\theta}(k) - \theta$, and $\tilde{\xi}(k) \triangleq \hat{\xi}(k) - \xi$. Let $\tilde{\eta}(k) \triangleq \begin{bmatrix} \tilde{\theta}(k) \\ \tilde{\xi}(k) \end{bmatrix}^T \begin{bmatrix} x(k) \\ 1 \end{bmatrix}$ and $\tilde{\eta}(z)$ be its z-transform. The error dynamics (13) can be written in the z-domain as:

$$\tilde{x}(z) = H(z)\tilde{\eta}(z) . \tag{14}$$

We need to show that the error dynamics (14) are bounded. Lemma 1 and 2 contain results needed to show boundedness of the prediction error dynamics. Lemma 1 proposes a Lyapunov function for the error dynamics of a form that is convenient to show uniform boundedness.

Lemma 1. Recall that A_m is asymptotically stable. Let P, $R \in \mathbb{R}^{n\times n}$ be positive definite matrices that satisfy:

$$P = A_m^T P A_m + R + I_n \tag{15}$$

and

$$\sigma \triangleq \sqrt{\lambda_{\max}(A_m^T P A_m)}\,, \tag{16}$$

where $\lambda_{\max}(A_m^T P A_m)$ is the maximum eigenvalue of $A_m^T P A_m$. Let $\mu > 0$ and define:

$$V_x(\tilde{x}(k)) \triangleq \ln(1 + \mu \tilde{x}^T(k) P \tilde{x}(k)) \tag{17}$$

and

$$\Delta V_x(\tilde{x}(k)) \triangleq V_x(\tilde{x}(k+1)) - V_x(\tilde{x}(k))\,. \tag{18}$$

Then

$$\Delta V_x(\tilde{x}(k)) \leq \mu \frac{-\tilde{x}^T(k) R \tilde{x}(k) + (\sigma^2+1) b_m^T P b_m [x^T(k)\tilde{\theta}(k) + \xi]^2}{1 + \mu \tilde{x}^T(k) P \tilde{x}(k)}\,,\ k \geq 0\,. \tag{19}$$

The proof of this lemma can be found in Section A.1. The result of the following Lemma is required to show uniform boundedness of the error dynamics $\tilde{x}$.

Lemma 2. There exist constants α, β, $\mu > 0$ such that

$$1 + x^T(k)x(k) \leq (1+\alpha\beta)(1 + \mu \tilde{x}^T(k) P \tilde{x}(k))\,. \tag{20}$$

The proof of Lemma 2 can be found in Section A.2. Using the results presented in Lemmas 1–2, we show that the prediction error $\tilde{x}$ is uniformly bounded.

Lemma 3. The prediction error in (13) is uniformly bounded by:

$$\|\tilde{x}\|_{\ell_\infty} \leq \sqrt{\frac{e^{w\rho_{\max}} - 1}{\mu \lambda_{\min}(P)}}\,, \quad \rho_{\max} \triangleq 4 \max_{\rho \in \Theta\times\Xi} \|\rho\|^2\,, \tag{21}$$

where P is defined in (15), $w > 0$, and $\mu > 0$ is defined in (A16).

The proof of Lemma 3 can be found in Section A.3. So far we have shown that $\tilde{x}(k)$ remains bounded, but $x(k)$ and $\hat{x}(k)$ could diverge at the same rate[3]. In the next Lemma, we show that $\hat{x}(k)$ in (12), with $u_{\ell_1}(k)$ given by (9), is uniformly bounded.

Lemma 4. The state estimate in (12) is uniformly bounded by:

$$\|\hat{x}\|_{\ell_\infty} \leq \frac{\lambda_\theta \sqrt{\frac{e^{w\rho_{\max}}-1}{\mu\lambda_{\min}(P)}} + \|G(z)\|_{\ell_1} L_\xi + \|H(z) k_g C(z)\|_{\ell_1} \|r\|_{\ell_\infty} + \|x_{in}\|_{\ell_\infty}}{1 - \lambda_\theta}\,. \tag{22}$$

The proof of this Lemma is found in Section A.4.

Finally, we define the following proper and BIBO stable transfer function

$$H_1(z) \triangleq C(z) \frac{1}{c_0^T H(z)} c_0^T\,, \tag{23}$$

where $c_0 \in \mathbb{R}^n$ and ensures that $c_0^T H(z)$ is a minimum phase transfer function with relative degree one. The full derivation is found in Section A.5.

3.2.2 | Lyapunov Stability

Lyapunov stability of the ℓ_1 controlled system is shown in two steps. First, we find a closed-loop reference model for the class of systems in (4) and show that it is stable. Then, we show that the ℓ_1 controlled system stays close to the closed-loop reference model. Consider the following nonadaptive version of the control system in (4) and (9) which defines the *closed-loop reference*

system for the class of systems in (4):

$$\begin{aligned} x_{ref}(k+1) &= A_m x_{ref}(k) + b_m(u_{ref}(k) + \theta^T x_{ref}(k) + \xi)\,, \quad x_{ref}(0) = x_0\,, \\ u_{ref}(z) &= -C(z)(\theta^T x_{ref}(z) + \xi - k_g r(z))\,, \\ y_{ref}(k) &= c_m^T x_{ref}(k)\,. \end{aligned} \tag{24}$$

In the next Lemma we first show that the closed-loop reference system (24) is stable.

Lemma 5. If $\|G(z)\|_{\ell_1} L_\theta < 1$, and $\|G(z)\|_{\ell_1} L_\xi < \infty$, then (24) is bounded-input, bounded-state (BIBS) stable with respect to $r(z)$ and x_0.

Proof. From the definition in (24), it follows that:

$$x_{ref}(z) = H(z)k_g C(z)r(z) + G(z)\theta^T x_{ref}(z) + G(z)\xi + x_{in}(z)\,, \tag{25}$$

where $x_{in}(z) \triangleq (zI_n - A_m)^{-1}x_0$. Since $H(z)$, $C(z)$ and $G(z)$ are proper BIBO stable discrete time transfer functions, it follows from (24) that for all $i \in \mathbb{N} \cup \{0\}$ the following bound holds:

$$\|x_{ref}|_i\|_{\ell_\infty} \leq \|H(z)k_g C(z)\|_{\ell_1} \|r|_i\|_{\ell_\infty} + \|G(z)\theta^T\|_{\ell_1} \|x_{ref}|_i\|_{\ell_\infty} + \|G(z)\xi\|_{\ell_1} + \|x_{in}|_i\|_{\ell_\infty}\,. \tag{26}$$

Since A_m is stable, $x_{in}(k)$ is uniformly bounded. Then, we have the following relationship

$$\|G(z)\theta^T\|_{\ell_1} = \max_{m=1,\ldots,n} \|G_m(z)\|_{\ell_1} \sum_{p=1}^{n} |\theta_p| \leq \|G(z)\|_{\ell_1} L_\theta < 1\,, \tag{27}$$

where $\|G_m(z)\|_{\ell_1}$ is the ℓ_1-norm for the impulse response for each output $g_m(k)$ where $m = 1, \ldots, n$. Recall from Assumption 1 that $\xi \in \Xi$. Then we have the following relationship

$$\|G(z)\xi\|_{\ell_1} = \max_{m=1,\ldots,n} \|G_m(z)\|_{\ell_1} |\xi| \leq \|G(z)\|_{\ell_1} L_\xi < \infty\,. \tag{28}$$

Consequently,

$$\|x_{ref}|_i\|_{\ell_\infty} \leq \frac{\|H(z)k_g C(z)\|_{\ell_1} \|r|_i\|_{\ell_\infty} + \|G(z)\|_{\ell_1} L_\xi + \|x_{in}|_i\|_{\ell_\infty}}{1 - \|G(z)\theta^T\|_{\ell_1}}\,. \tag{29}$$

Since $r(k)$ and $x_{in}(k)$ are uniformly bounded, $\|G(z)\|_{\ell_1} L_\xi$ is bounded, and (29) holds uniformly for all $i \in \mathbb{N} \cup \{0\}$, $x_{ref}(k)$ is uniformly bounded. Boundedness of $y_{ref}(k)$ follows from its definition. □

The second step to show Lyapunov stability of the ℓ_1 controlled system is to show that the system stays close to the closed-loop reference system (24). The following theorem shows that the difference between states and inputs of the ℓ_1 controlled system and the closed-loop reference system (24) is bounded.

Theorem 1. For the system in (4) and the controller defined via (5) – (6) and (9) subject to the ℓ_1-norm condition in (10), we have

$$\|x_{ref} - x\|_{\ell_\infty} \leq \gamma_1\,, \quad \|u_{ref} - u\|_{\ell_\infty} \leq \gamma_2\,, \tag{30}$$

where

$$\begin{aligned} \gamma_1 &\triangleq \frac{\|C(z)\|_{\ell_1}}{1 - \|G(z)\|_{\ell_1} L_\theta} \sqrt{\frac{e^{w\rho_{\max}} - 1}{\mu \lambda_{\min}(P)}}\,, \\ \gamma_2 &\triangleq \|H_1(z)\|_{\ell_1} \sqrt{\frac{e^{w\rho_{\max}} - 1}{\mu \lambda_{\min}(P)}} + \|C(z)\theta^T\|_{\ell_1} \gamma_1 + L_\xi \|1 - C(z)\|_{\ell_1}\,. \end{aligned} \tag{31}$$

Proof. The response of the closed-loop system in (4) with the ℓ_1 adaptive controller in (9) can be written in the z domain as:

$$\begin{aligned} x(z) &= H(z)C(z)k_g r(z) - H(z)C(z)\hat{\eta}(z) + H(z)(\theta^T x(z) + \xi) + x_{in}(z) \\ &= H(z)C(z)k_g r(z) - H(z)C(z)\tilde{\eta}(z) + G(z)(\theta^T x(z) + \xi) + x_{in}(z) \end{aligned} \tag{32}$$

The above equality holds because $\tilde{\eta}(z) = \hat{\eta}(z) - \rho^T \begin{bmatrix} x(z) \\ 1 \end{bmatrix}$ and $-\hat{\eta}(z) = -\tilde{\eta}(z) - \rho^T \begin{bmatrix} x(z) \\ 1 \end{bmatrix}$. From the definition of the closed-loop reference system in (24), it follows that

$$x_{ref}(z) = H(z)k_g C(z)r(z) + G(z)\rho^T \begin{bmatrix} x_{ref}(z) \\ 1 \end{bmatrix} + x_{in}(z)\,. \tag{33}$$

The expressions above and the predictor error dynamics in (14) allow us to write

$$\begin{aligned} x_{ref}(z) - x(z) &= G(z)\rho^T \left(\begin{bmatrix} x_{ref}(z) \\ 1 \end{bmatrix} - \begin{bmatrix} x(z) \\ 1 \end{bmatrix} \right) + H(z)C(z)\tilde{\eta}(z) \\ &= G(z)\theta^T(x_{ref}(z) - x(z)) + H(z)C(z)\tilde{\eta}(z) \\ &= G(z)\theta^T(x_{ref}(z) - x(z)) + C(z)\tilde{x}(z)\,, \end{aligned} \tag{34}$$

which implies that

$$\|(x_{ref} - x)|_i\|_{\ell_\infty} \leq \|G(z)\theta^T\|_{\ell_1} \|(x_{ref} - x)|_i\|_{\ell_\infty} + \|C(z)\|_{\ell_1} \|\tilde{x}|_i\|_{\ell_\infty}\,. \tag{35}$$

Then, the bounds in (27) and (21) lead to the following uniform upper bound:

$$\|(x_{ref} - x)|_i\|_{\ell_\infty} \leq \frac{\|C(z)\|_{\ell_1} \|\tilde{x}|_i\|_{\ell_\infty}}{1 - \|G(z)\|_{\ell_1} L_\theta} \leq \frac{\|C(z)\|_{\ell_1}}{1 - \|G(z)\|_{\ell_1} L_\theta} \sqrt{\frac{e^{\omega \rho_{\max}} - 1}{\mu \lambda_{\min}(P)}}\,. \tag{36}$$

To derive the second bound, notice that (9) and (24) lead to:

$$\begin{aligned} u_{ref}(z) - u(z) &= -C(z)\left(\rho^T \begin{bmatrix} x_{ref}(z) \\ 1 \end{bmatrix} - k_g r(z) \right) + C(z)(\hat{\eta}(z) - k_g r(z)) \\ &= -C(z)\rho^T \begin{bmatrix} x_{ref}(z) \\ 1 \end{bmatrix} + C(z)\hat{\eta}(z) \\ &= -C(z)\theta^T x_{ref}(z) - C(z)\xi + C(z)(\tilde{\eta}(z) + \theta^T x(z) + \xi) \\ &= C(z)\tilde{\eta}(z) - C(z)\theta^T(x_{ref}(z) - x(z)) - \xi(1 - C(z))\,. \end{aligned} \tag{37}$$

It follows from the error dynamics in (14) and the definition of $H_1(z)$ in (23) that

$$C(z)\tilde{\eta}(z) = C(z)\frac{c_0^T H(z)}{c_0^T H(z)}\tilde{\eta}(z) = H_1(z)H(z)\tilde{\eta}(z) = H_1(z)\tilde{x}(z)\,, \tag{38}$$

which implies that

$$u_{ref}(z) - u(z) = H_1(z)\tilde{x}(z) - C(z)\theta^T(x_{ref}(z) - x(z)) - \xi(1 - C(z))\,, \tag{39}$$

and the following bound holds

$$\|(u_{ref} - u)|_i\|_{\ell_\infty} \leq \|H_1(z)\|_{\ell_1} \|\tilde{x}(z)\|_i\|_{\ell_\infty} + \|C(z)\theta^T\|_{\ell_1} \|(x_{ref} - x)|_i\|_{\ell_\infty} + L_\xi \|1 - C(z)\|_{\ell_1}\,. \tag{40}$$

This bound holds uniformly. The bounds in (21) and (36) lead to

$$\|(u_{ref} - u)|_i\|_{\ell_\infty} \leq \|H_1(z)\|_{\ell_1} \sqrt{\frac{e^{\omega \rho_{\max}} - 1}{\mu \lambda_{\min}(P)}} + \|C(z)\theta^T\|_{\ell_1} \gamma_1 + L_\xi \|1 - C(z)\|_{\ell_1}\,. \tag{41}$$

□

In this subsection we used Lemma 5 to show that the closed-loop reference system (24) is BIBS stable. Then, in Theorem 1 we showed that the difference between the state and input of the ℓ_1 controlled system and the state and input of the closed-loop reference system is bounded. Therefore, we can conclude that the ℓ_1 controlled system is stable in the sense of Lyapunov.

3.3 | Performance Bounds

The closed-loop reference system (24) is not realizable as it depends on knowing the true value of θ and ξ; hence, it cannot be used to describe the $\mathscr{L}_1$ controlled system. In this subsection we show that the $\mathscr{L}_1$ controlled system stays close to a known linear model. Further, we show that the difference between these two systems is bounded.

The $\mathscr{L}_1$ adaptive controller is able to cancel the uncertainties of the system exactly when the uncertainties are within the bandwidth of the low-pass filter. In this *ideal* scenario, the system response is the following:

$$x_{id}(k+1) = A_m x_{id}(k) + b_m k_g r(k)\,, \tag{42}$$

where the subscript x_{id} is used to denote the ideal response. All variables of the ideal system are known; hence, it can be implemented. Next, we show that there exists a uniform upper bound for the difference between the ideal (42) and the $\mathscr{L}_1$ controlled system. This bound will be later used in the robust MPC implementation.

Theorem 2. For the system (4) and the controller defined via (5) – (6) and (9) subject to the $\mathscr{L}_1$-norm condition in (10), we have

$$\|x_{id} - x\|_{\mathscr{L}_\infty} \leq \|H(z)(1 - C(z)k_g)\|_{\mathscr{L}_1} \|r\|_{\mathscr{L}_\infty} + \|H(z)H_1(z)\|_{\mathscr{L}_1} \|\tilde{x}\|_{\mathscr{L}_\infty} + \lambda_\theta \left(\|\hat{x}\|_{\mathscr{L}_\infty} + \sqrt{\frac{e^{\omega\rho_{\max}} - 1}{\mu\lambda_{\min}(P)}} \right) + \lambda_\xi\,. \tag{43}$$

Proof. The response of the ideal system in the z domain can be written as:

$$x_{id}(z) = H(z)k_g r(z) + x_{in}(z)\,. \tag{44}$$

The response of the closed-loop system in (4) with the $\mathscr{L}_1$ adaptive controller in (9) can be written in the z domain as (32). The latter implies that:

$$\begin{aligned} x(z) - x_{id}(z) &= H(z)k_g(C(z) - 1)r(z) - H(z)C(z)\hat{\eta}(z) + H(z)(\theta^T x(z) + \xi) \\ &= H(z)k_g(C(z) - 1)r(z) - H(z)C(z)\tilde{\eta}(z) + G(z)(\theta^T x(z) + \xi) \\ &= H(z)k_g(C(z) - 1)r(z) - H(z)H_1(z)\tilde{x}(z) + G(z)(\theta^T x(z) + \xi)\,. \end{aligned} \tag{45}$$

Using the above, the following uniform bound can be derived

$$\begin{aligned} \|(x - x_{id})|_i\|_{\mathscr{L}_\infty} \leq\ & \|H(z)k_g(C(z) - 1)\|_{\mathscr{L}_1} \|r|_i\|_{\mathscr{L}_\infty} + \|H(z)H_1(z)\|_{\mathscr{L}_1} \|\tilde{x}\|_{\mathscr{L}_\infty} \\ & + \lambda_\theta \left(\|\hat{x}\|_{\mathscr{L}_\infty} + \sqrt{\frac{e^{\omega\rho_{\max}} - 1}{\mu\lambda_{\min}(P)}} \right) + \lambda_\xi\,. \end{aligned} \tag{46}$$

□

Ideally, the controller would be able to compensate for all the disturbances in the system, which would make the controlled system behave exactly as the ideal system (42). However, this is not always possible and modeling errors arise. In Theorem 2, we showed that there exists a uniform upper bound for the difference between the real and the ideal system. This bound can be interpreted as a bound on the modeling error of the adaptive system by the linear model (42), which is required for the implementation of the robust MPC, as will be shown in the next Section.

4 | MODEL PREDICTIVE CONTROL

The main disadvantages of MPC are the difficulty to incorporate plant model uncertainties explicitly [37] and to guarantee robust stability of the origin. Robust MPC implementations have two requirements: *(i)* a model of the system, and *(ii)* a bound on the disturbance in the system, which may be thought of as a modeling error. In Section 3, we presented a discrete time state feedback $\mathscr{L}_1$ adaptive controller that is able to provide a system model and an upper bound on the modeling error. Any robust MPC implementation that has a bounded additive disturbance [21,22] can be used. In this work we use the robust MPC for linear systems with additive disturbance proposed in [21]. For convenience and completeness, we briefly describe the robust MPC approach and highlight its role in the robust adaptive MPC framework. The discrete time system to be controlled by a robust MPC is described by:

$$x(k+1) = A_m x(k) + b_m u(k) + w\,, \tag{47}$$

where $x(k) \in \mathbb{R}^n$ is the state at time step k, $u \in \mathbb{R}^m$ is the input, $w \in \mathbb{R}^n$ is a bounded disturbance, and $x(k+1)$ denotes the successor state. The system is subject to state and input constraints

$$u \in \mathbb{U}, \quad x \in \mathbb{X}, \tag{48}$$

where $\mathbb{U} \subset \mathbb{R}^m$ is compact, $\mathbb{X} \subset \mathbb{R}^n$ is closed, and each set contains the origin in its interior. The disturbance w is assumed to be bounded

$$w \in W, \tag{49}$$

where W is compact and contains the origin. In our work, this disturbance is the modeling error from the ℓ_1 adaptive controller and is described by (43). Let $\mathbf{u} = [u(0), \dots, u(N_H - 1)]$ be the control sequence and $\mathbf{w} = [w(0), \dots, w(N_H - 1)]$ be the disturbance sequence. Let $\phi(k; x, \mathbf{u}, \mathbf{w})$ be the solution of (47) at time k when the initial state is x (at time 0) and the control and disturbance sequences are $\mathbf{u}$ and $\mathbf{w}$, respectively. We define the *nominal* system corresponding to (47) by

$$\bar{x}(k+1) = A_m \bar{x}(k) + b_m \bar{u}(k), \tag{50}$$

and define $\bar{\phi}(k; \bar{x}, \bar{\mathbf{u}})$ as the solution of (50) at time k when the initial state is $\bar{x}$ and the control sequence is $\bar{\mathbf{u}}$.

Let $K \in \mathbb{R}^{m \times n}$ such that $A_K \triangleq A_m + b_m K$ is stable. Further, let Z be a disturbance invariant set for the controlled uncertain system $x(k+1) = A_K x(k) + w$ satisfying

$$A_K Z \oplus W \subseteq Z, \tag{51}$$

where $\oplus$ denotes Minkowski set addition, i.e., $A \oplus B \triangleq \{a + b \mid a \in A, b \in B\}$. The set Z serves as the 'origin' for the perturbed system. In order to reduce conservativeness, the set Z should be as small as possible. Different ways to calculate and minimize Z are presented in[21]. The following Proposition is introduced as it is required for the definition of the controller. It states that the feedback policy $u(k) = \bar{u}(k) + K(x(k) - \bar{x}(k))$ keeps the states $x(k)$ of the uncertain system (47) close to the states $\bar{x}(k)$ of the nominal system (50) for all $\bar{u}(\cdot)$, $x(k) \in \bar{x}(k) \oplus Z$ if $x(0) \in \bar{x}(0) \oplus Z$.

Proposition 1. Suppose Z is disturbance invariant for $x(k+1) = A_K x(k) + w$. If $x \in \bar{x} \oplus Z$ and $u(k) = \bar{u}(k) + K(x(k) - \bar{x}(k))$, then $x(k+1) \in \bar{x}(k+1) \oplus Z$ for all $w \in W$ where $x(k+1) = A_m x(k) + b_m u(k) + w$ and $\bar{x}(k+1) = A_m \bar{x}(k) + b_m \bar{u}(k)$.

The robust MPC used in this work[21] has three main changes compared to nominal MPC *(i)* addition of the initial state as a decision variable, *(ii)* tightening of state and input constraints, and *(iii)* modification of the control law to include an ancillary controller. In order to better understand these changes, we first present the nominal MPC formulation.

4.1 | Nominal Model Predictive Control

Nominal MPC is described by a conventional optimal control problem $\mathbb{P}_{N_H}(x(k))$ with constraints that are tighter than those specified in (48), where $x(k)$ is the current state with no uncertainties. The tighter constraints guarantee that the controller for the uncertain system satisfies the original constraints; however, nominal MPC generally doesn't tighten its constraints. The optimal control problem $\mathbb{P}_{N_H}(x(k))$ is defined by

$$V^0_{N_H}(x(k)) = \min_{\mathbf{u}} \{V_{N_H}(x(k), \mathbf{u}) \mid \mathbf{u} \in \mathcal{U}_{N_H}(x(k))\}, \tag{52}$$

$$\mathbf{u}^0(x(k)) = \arg\min_{\mathbf{u}} \{V_{N_H}(x, \mathbf{u}) \mid \mathbf{u} \in \mathcal{U}_{N_H}(x(k))\}, \tag{53}$$

where the cost function $V_{N_H}(\cdot)$ is defined over the time horizon N_H by

$$V_{N_H}(x(k), \mathbf{u}) \triangleq \sum_{i=0}^{N_H - 1} l(x(i), u(i)) + V_f(x(N_H)). \tag{54}$$

Note that each $x(i)$ satisfies $x(i) = \bar{\phi}(i; x, \bar{u})$, and $\mathcal{U}_{N_H} x(k)$ is the set of control sequences satisfying the tighter control, state and terminal constraints defined by

$$\begin{aligned} &u(i) \in \bar{\mathbb{U}} \triangleq \mathbb{U} \ominus KZ, \quad i \in \mathcal{I}_{N_H - 1}, \\ &x(i) \in \bar{\mathbb{X}} \triangleq \mathbb{X} \ominus Z, \quad i \in \mathcal{I}_{N_H - 1}, \\ &x(N_H) \in X_f \subset \mathbb{X} \ominus Z, \end{aligned} \tag{55}$$

where $\mathcal{I}_{N_H - 1} \triangleq \{0, 1, \dots, N_H - 1\}$, and X_f is the terminal constraint set for $\mathbb{P}_{N_H}(x(k))$, which is assumed to have an interior. Hence,

$$\mathcal{U}_{N_H}(x(k)) = \{\mathbf{u} \mid u(i) \in \bar{\mathbb{U}},\ \bar{\phi}(i; x, \mathbf{u}) \in \bar{\mathbb{X}},\ \forall i \in \mathcal{I}_{N_H - 1},\ \bar{\phi}(N_H; x(k), \mathbf{u}) \in X_f\}, \tag{56}$$

where $x(i) = \bar{\phi}(N_H; x(k), \mathbf{u})$. The domain of the value function $V^0_{N_H}(\cdot)$ is $\bar{X}_{N_H}$ and is defined by

$$\bar{X}_{N_H} \triangleq \{x(k) \mid \mathcal{U}_{N_H}(x(k)) \neq \emptyset\}. \tag{57}$$

Further, we assume that the set W is small enough to ensure that $Z \subset$ interior($\mathbb{X}$) and $KZ \subset$ interior($\mathbb{U}$). Note that the minimal Z is proportional to W. Let

$$l(x,u) \triangleq \frac{1}{2}[x^T Q_{\text{MPC}} x + u^T R_{\text{MPC}} u], \quad V_f(x) \triangleq \frac{1}{2} x^T P_{\text{MPC}} x, \tag{58}$$

where Q_{MPC}, R_{MPC} and P_{MPC} are positive definite. The terminal cost $V_f(\cdot)$ and the terminal constraint set X_f satisfy the following usual axioms [13]:

Axiom 1. $A_K X_f \subset X_f$, $X_f \subset \mathbb{X} \ominus Z$, $KX_f \subset \mathbb{U} \ominus KZ$,

Axiom 2. $V_f(A_K x(k)) + l(x(k), Kx(k)) \leq V_f(x(k)) \; \forall x(k) \in X_f$.

Solving the optimal control problem $\mathbb{P}_{N_H}(x(k))$ yields the optimal control sequence $\mathbf{u}^0(x(k)) \triangleq \{u^0_0(x(k)), u^0_1(x(k)), \ldots, u^0_{N_H-1}(x(k))\}$ and the optimal state sequence $\mathbf{x}^0(x(k)) \triangleq \{x^0_0(x(k)), x^0_1(x(k)), \ldots, x^0_{N_H}(x(k))\}$, $x^0_0(x(k)) \triangleq x(k)$, where, for each i, $x^0_i(x(k)) \triangleq \bar{\phi}(i; x(k), \mathbf{u}^0(x(k)))$. With the following model predictive control law

$$\kappa^0_{N_H}(x(k)) \triangleq u^0_0(x(k)), \tag{59}$$

the nominal system, under model predictive control satisfies

$$x(k+1) = A_m x(k) + b_m \kappa^0_{N_H}(x(k)). \tag{60}$$

The model predictive control law ensures that the nominal system satisfies the tighter constraints for all initial states in $\bar{X}_{N_H}$. The nominal system is also stabilized as , under the assumptions above, the value function satisfies [13]

$$V^0_{N_H}(A_m x(k) + b_m \kappa^0_{N_H}(x(k))) \leq V^0_{N_H}(x(k)) - l(x(k), \kappa^0_{N_H}(x(k))). \tag{61}$$

We assume $\bar{X}_{N_H}$ is bounded, hence, the origin is exponentially stable for the controlled nominal system $x(k+1) = A_m x(k) + b_m \kappa^0_{N_H}(x(k))$ with a region of attraction $\bar{X}_{N_H}$.

Recall that nominal MPC provides nominal robustness to small disturbances in the nominal system; however, if larger disturbances are present, they should be addressed by a robust MPC.

4.2 | Robust Model Predictive Control

When there are additive disturbances, given any state $x(k) \in \bar{X}_{N_H} \setminus Z$, it may not be true that $V^0_{N_H}(A_m x(k) + b_m \kappa^0_{N_H}(x(k)) + w) \leq V^0_{N_H}(x(k))$ for all $w \in W$. As stated at the beginning of this Section (see eq. (51)), there exists a controller K for which we can find a disturbance invariant set Z. In other words, the controller K is able to keep in Z any state beginning in Z. Hence, we need to establish robust asymptotic stability of Z. To do this, the optimization problem incorporates the initial state. This is possible as the initial state x_0 in the optimal control problem is not the current state $x(k)$ of a plant, but a parameter of the control law. The new optimal control problem $\mathbb{P}^*_{N_H}(x(k))$, is a modification of the previously defined conventional optimal control problem $\mathbb{P}_{N_H}(x(k))$. The new problem $\mathbb{P}^*_{N_H}(x(k))$ is defined as follows

$$V^*_{N_H}(x(k)) = \min_{x_0, \mathbf{u}} \{V_{N_H}(x_0, \mathbf{u}) \mid \mathbf{u} \in \mathcal{U}_{N_H}(x_0), \; x \in x_0 \oplus Z\}, \tag{62}$$

$$(x^*_0(x(k)), \mathbf{u}^*(x(k))) = \arg\min_{x_0, \mathbf{u}} \{V_{N_H}(x_0, \mathbf{u}) \mid \mathbf{u} \in \mathcal{U}_{N_H}(x_0), \; x(k) \in x_0 \oplus Z\}. \tag{63}$$

The function $V_{N_H}(x(k), \mathbf{u})$ was defined in (54) and the constraint set $\mathcal{U}(x(k))$ was defined in (55). Note that the only difference between $\mathbb{P}_{N_H}(x(k))$ and $\mathbb{P}^*_{N_H}(x(k))$ is that the initial state x_0 of the model employed in the new optimal control problem $\mathbb{P}^*_{N_H}(x(k))$ is a decision variable that can be varied if it satisfies the following constraint

$$x(k) \in x_0 \oplus Z, \tag{64}$$

where $x(k)$ is the current state of the system being controlled. It can be shown that $\mathbb{P}^*_{N_H}(x(k))$ is a quadratic program that yields the optimal control sequence $\mathbf{u}^*(x(k)) \triangleq \{u^*_0(x(k)), \ldots, u^*_{N_H-1}(x(k))\}$ and the optimal state sequence $\mathbf{x}^*(x(k)) \triangleq \{x^*_0(x(k)), \ldots, x^*_{N_H}(x(k))\}$ where, for each $i > 0$, $x^*_i(x(k)) \triangleq \bar{\phi}(i; x^*_0(x(k)), \mathbf{u}^*(x(k)))$. Note that the optimal state $x^*_0(x(k))$ is not necessarily equal to the current state $x(k)$. A pair $(x_0, \mathbf{u})$ is a feasible solution of $\mathbb{P}^*_{N_H}(x(k))$ if $x(k) \in x_0 \oplus Z$ and $\mathbf{u} \in \mathbb{U}_{N_H}(x_0)$.

Motivated by Proposition 1, we propose the following implicit model predictive control law $\kappa^*_{N_H}(\cdot)$, yielded by the solution of $\mathbb{P}^*_{N_H}(x(k))$

$$\kappa^*_{N_H}(x(k)) \triangleq u^*_0(x(k)) + K(x(0) - x^*_0(x(k)))\,. \tag{65}$$

The control $\kappa^*_{N_H}(x(k))$is not necessarily equal to $u^*_0(x(k))$, as in conventional model predictive control. Let $x(k)$ be an arbitrary state in

$$X_{N_H} \triangleq \{x \mid \exists\, x_0 \text{ such that } x(k) \in x_0 \oplus Z,\, \mathbb{U}_{N_H}(x_0) \neq \}\,, \tag{66}$$

the domain of the value function $V^*_{N_H}(\cdot)$. The optimal control and state trajectories for the problem $\mathbb{P}^*_{N_H}(x(k))$, $\mathbf{u}^*(x(k))$ and $\mathbf{x}^*(x(k))$, respectively, satisfy

$$\begin{aligned} &u^*_i(x(k)) \in \mathbb{U} \ominus KZ\,, \\ &x^*_i(x(k)) \in \mathbb{X} \ominus Z\,, \forall i \in \mathcal{I}_{N_H-1} \text{ and} \\ &x^*_{N_H}(x(k)) \in X_f \subset \mathbb{X} \ominus Z\,. \end{aligned} \tag{67}$$

In[21] it was shown that for the above robust model predictive controller, the set Z is robustly exponentially stable for the controlled uncertain system $x(k+1) = A_m x(k) + b_m \kappa^*_{N_H}(x(k)) + w$ where $w \in W$, and that the region of attraction is X_{N_H}.

This robust MPC implementation provides robust stability of the invariant set Z. To achieve this the initial state is added as a decision variable in the optimization problem and the control law is modified to include an ancillary controller whose aim is to keep the state in the invariant set Z. In this way, a system with additive disturbances (47) is able to track a path that decreases the cost of the value function, despite the presence of disturbances.

5 | ROBUST ADAPTIVE MPC FRAMEWORK

This Section presents a summary of the proposed robust adaptive MPC framework in Fig. 1. We propose an $\mathscr{L}_1$ adaptive controller (orange dashed box in Fig. 1) as an underlying controller to a robust MPC (blue box in Fig. 1) to achieve safe and high accuracy tracking performance. The key idea of this framework is to leverage characteristics of the discrete time state feedback $\mathscr{L}_1$ adaptive controller to satisfy the requirements for the linear robust MPC.

In Section 3 we showed that the proposed discrete time state feedback $\mathscr{L}_1$ adaptive controller is able to make a system behave close to an ideal linear system (42). The ideal system behavior is achievable only if all the uncertainties in the system are canceled, which is not always attainable. Therefore, the behavior of the real system may deviate from the ideal system behavior. We also showed that the deviation between ideal and real behavior is bounded (43) and can be thought of as a bound on the modeling error. We can then reformulate the ideal linear system (42) as a linear system with a bounded additive disturbance defined by (43).

In Section 4.2 we described a robust MPC for linear systems with bounded additive disturbances. The robust MPC uses an ancillary controller to achieve robust stability of a disturbance invariant set. In this work we formulate a cost function that aims to minimize tracking error and control effort of the $\mathscr{L}_1$ controlled system. In summary, the underlying $\mathscr{L}_1$ adaptive controller is able to provide a linear system model with bounded additive disturbances to be used by the robust MPC; while the robust MPC provides the $\mathscr{L}_1$ adaptive controller with a reference signal that will minimize the tracking error of the $\mathscr{L}_1$ controlled system. In this way, we are enable a system to achieve safe and high accuracy tracking in the presence of modeling errors.

6 | EXPERIMENTAL RESULTS

This section presents experimental results of the proposed robust adaptive MPC applied to a quadrotor flying a given trajectory. We show that combining an underlying $\mathscr{L}_1$ adaptive controller with a robust MPC instead of a nominal MPC (as we previously proposed in[30]), enables the quadrotor to successfully complete more challenging tasks. We assess three aspects of the performance of the proposed framework: *(i)* trajectory tracking performance, *(ii)* obstacle avoidance without trajectory update, and *(iii)* obstacle avoidance with trajectory update. The performance of the proposed robust adaptive MPC is compared to the performance of an adaptive MPC framework where the same underlying discrete time state feedback $\mathscr{L}_1$ adaptive controller described in Section 3 is combined with a *nominal* MPC.

Section 6.1 describes the experimental setup and implementation details of the proposed robust adaptive MPC. Section 6.2 compares the tracking performance of adaptive MPC and robust adaptive MPC on six different trajectories. Finally, an obstacle

FIGURE 2 Parrot Bebop 2 used in the experiments.

is introduced and the ability of the controllers to steer the quadrotor to avoid a collision is tested in Section 6.3, where the trajectory is not updated to avoid the obstacle, and Section 6.4, where the trajectory is updated to avoid the obstacle.

6.1 | Experimental Setup

The vehicle used in the experiments is the Parrot Bebop 2. A central overhead motion capture camera system provides position, roll-pitch-yaw Euler angles and rotational velocity measurements, and through numerical differentiation, we obtain translational velocities of the quadrotor. We implement a discrete time state feedback $\mathscr{L}_1$ adaptive controller for position control of each axis, as described in Section 3. We assume that the x, y, and z axes are decoupled. The signal $r(k)$ in Fig. 1 is the desired position in x, y and z directions. The output of the $\mathscr{L}_1$ adaptive controller $u_{i,\mathscr{L}_1}(k)$ where $i = \mathsf{x}, \mathsf{y}, \mathsf{z}$, commanded x and y translational acceleration and commanded z velocity, respectively, is specified in the global coordinate frame. However, the interface to the real quadrotor ('Plant' in Fig. 1) requires commanded roll (ϕ_{des}), pitch (θ_{des}), vertical velocity ($\dot{z}_{des}$), and rotational velocity around the z axis (ψ_{des}) (see[38]). Therefore, we transform the signals $u_{i,\mathscr{L}_1}(k)$ through the following nonlinear transformation:

$$\begin{aligned} \phi_{des} &= -\arcsin(-u_x \sin(\psi) + u_y \cos(\psi)), \\ \theta_{des} &= \arcsin(u_x \cos(\psi) + u_y \sin(\psi)), \\ \dot{z}_{des} &= u_{\mathsf{z},\mathscr{L}_1}, \end{aligned} \tag{68}$$

where ψ is the current yaw angle. During the experiment, the desired yaw angle (ψ_{des}) is set to zero and is controlled with a proportional controller $u_\psi = k_\psi(\psi_{des} - \psi)$, with proportional gain k_ψ. We also implement a robust MPC as described in Section 4.2. The signal $x(k)$ is the state and $r^*(k)$ the desired state, which in this case is composed of position and velocity.

Nominal and robust MPC use the same weights on the value function to be minimized that is defined as follows:

$$V_{N_H}(x(k), \mathbf{u}) \triangleq \sum_{i=0}^{N_H-1} e_x^T(i) Q_{\text{MPC}} e_x(i) + u^T(i) R_{\text{MPC}} u(i) + V_f(x(N_H)), \tag{69}$$

where Q_{MPC} and R_{MPC} are positive definite matrices defined by $Q_{\text{MPC}} = qI$, $R_{\text{MPC}} = rI$, and $e_x(i)$ is the error between the state $x(i)$ and the desired state $r^*(i)$. We constrain the input to guarantee that the quadrotor remains in a given area at all times. We use IBM CPLEX optimizer to solve the above optimization problem.

To quantify the tracking error we define the average position error along the trajectory as:

$$e = \frac{\sum_{i=1}^{N} \sqrt{(r_{\mathsf{x},pos}(i) - x_{\mathsf{x},pos}(i))^2 + (r_{\mathsf{y},pos}(i) - x_{\mathsf{y},pos}(i))^2 + (r_{\mathsf{z},pos}(i) - x_{\mathsf{z},pos}(i))^2}}{N}, \tag{70}$$

where $r_{j,pos}(i)$ are the desired positions and $x_{j,pos}(i)$ are the measured positions with $j = \mathsf{x}, \mathsf{y}, \mathsf{z}$.

In Section 3 we showed that there is an upper bound on the modeling error between the real and ideal state (43). This bound is conservative and would not be suitable to use in a robust MPC framework. However, this bound can experimentally be characterized through simple step response experiments. Fig. 4 shows the response of the actual system and the ideal system of the x, y, and z axes for a step response. The deviation between the real and ideal system response is used to characterize the upper bound (43).

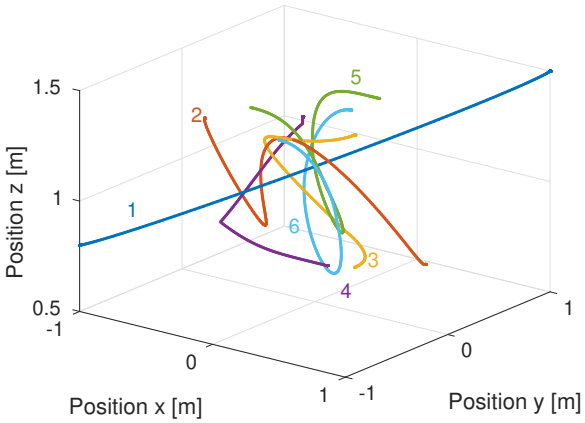


FIGURE 3 The six different trajectories that are used to test trajectory tracking of the robust adaptive MPC framework.

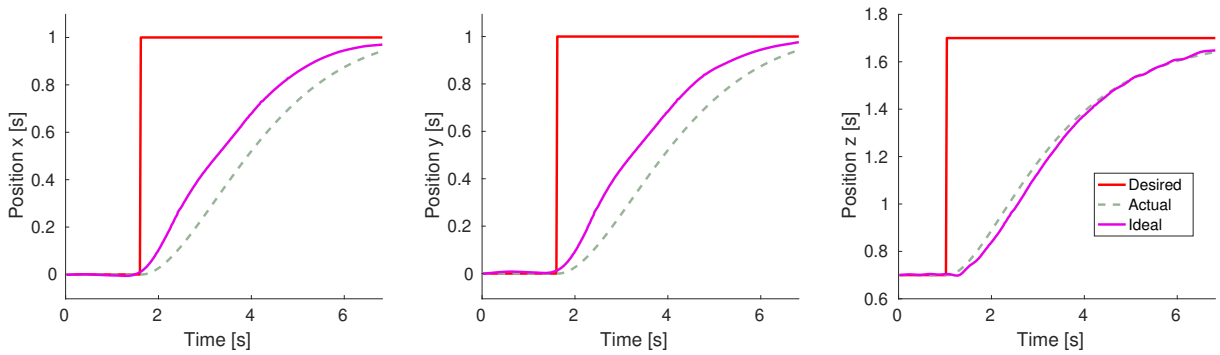


FIGURE 4 Comparison of ideal trajectory followed by the ideal system (solid magenta line) and actual trajectory followed by the quadrotor (dashed gray line).

6.2 | Tracking Performance

In our previous work[30] we showed an adaptive MPC with tracking capabilities that outperform those of non adaptive and non predictive approaches. In this subsection we want to show that the added robustness of the robust MPC does not negatively affect the tracking performance of the robust adaptive MPC. The adaptive MPC in[30] has an underlying discretized output feedback $\mathcal{L}_1$ adaptive controller. Recall that in this experiment, the adaptive and the robust adaptive MPC use the same underlying discrete time state feedback ℓ_1 adaptive controller described in Section 3. The only differences between adaptive and robust adaptive MPC are *(i)* the inclusion of the initial state as a decision variable in the robust MPC, and *(ii)* the modification of the robust MPC control law to include an ancillary controller.

We propose six different trajectories to test the tracking performance of our approach, as shown in Fig. 3. The average position tracking errors for each controller and trajectory are shown in Fig. 5. In the six trajectories both controllers have a similar performance. In five of the six trajectories, the robust adaptive MPC achieves a smaller tracking error, since it is able to incorporate the error between the ideal and real system state.

6.3 | Obstacle Avoidance without Trajectory Update

We assess the performance of the controllers when an obstacle is introduced in the environment. Fig. 6a shows the obstacle and the desired trajectory in 3D space. Fig. 6b shows a top view of the obstacle (in black), and the desired trajectory (in red). The obstacle lies on the path of the desired trajectory; hence, the controllers must compute inputs that force the system to fly around the obstacle and avoid a collision. In this experiment, the trajectory remains unchanged despite the presence of the obstacle. The

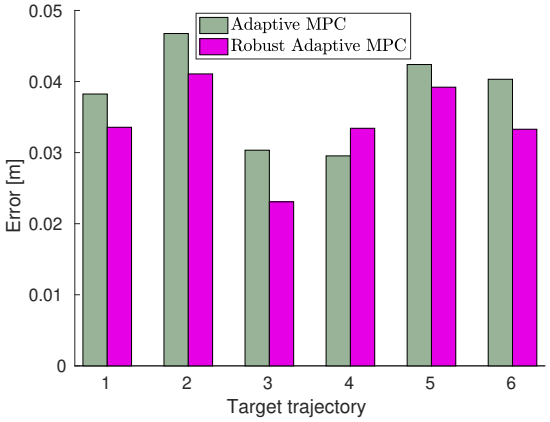


FIGURE 5 Average tracking error in quadrotor experiments of six different trajectories with adaptive MPC and the proposed robust adaptive MPC. Robust Adaptive MPC has a lower trajectory tracking error in most trajectories.

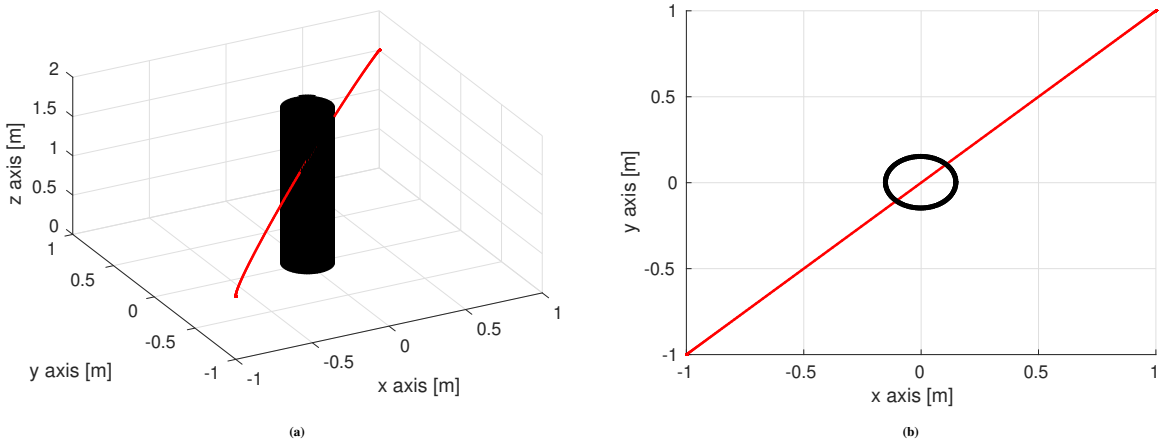


FIGURE 6 The drone is required to fly a 3D trajectory as shown in Fig. 6a. However, an obstacle is detected after the trajectory has been calculated. For this experiment the trajectory remains constant even if an obstacle lies on the trajectory. Fig. 6b is the top view and shows that the trajectory passes through the center of the obstacle. The drone must avoid the obstacle to successfully complete the task.

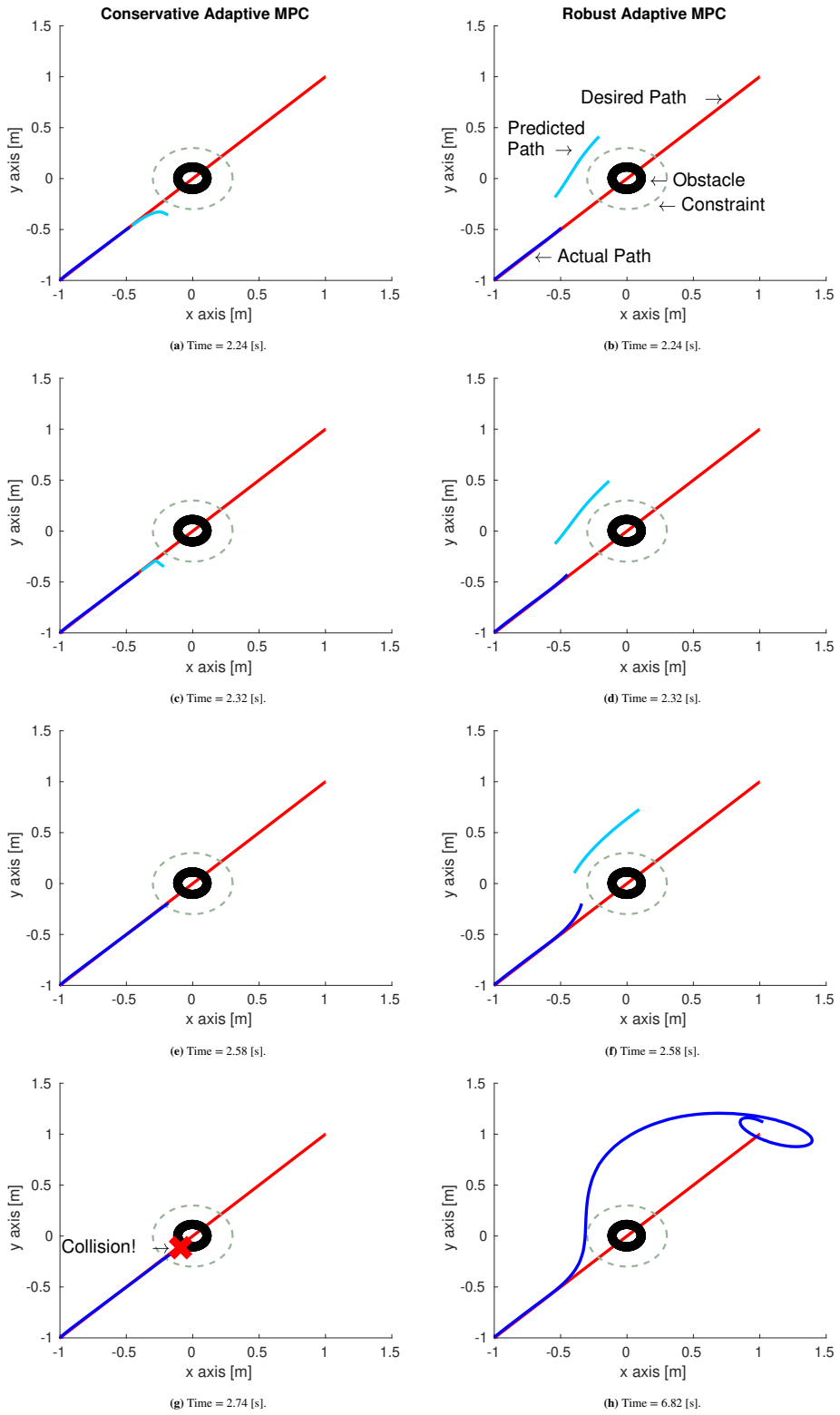
drone must avoid the obstacle during the flight in order to successfully complete the task. To avoid crashing with the obstacle, constraints on the allowable states set $\mathbb{X}$ are introduced such that

$$\mathbb{X}_{new} = \mathbb{X} \ominus \mathbb{W}, \tag{71}$$

where $\mathbb{W}$ is the obstacle set which includes the positions that lie inside the obstacle. For this experiment, the obstacle set $\mathbb{W}$ is described by

$$x^2 + y^2 \leq 0.15^2. \tag{72}$$

Recall from Section 4.2 that the robust MPC implementation we use in this work requires us to further constrain the set of allowable states. In the framework of this experiment this means that each state in the optimal state sequence must satisfy $x_i^*(x(k)) \in \mathbb{X}_{new} \ominus Z$. In this experiment we will compare the proposed robust adaptive MPC with two variations of adaptive MPC *(i)* allowable state set is unchanged, i.e., $x_i^*(x(k)) \in \mathbb{X}_{new}$, which we will refer to as *nominal* adaptive MPC and *(ii)* allowable state set is constrained, i.e., $x_i^*(x(k)) \in \mathbb{X}_{new} \ominus Z$, which we will refer to as *conservative* adaptive MPC. Typically, a nominal MPC implementation does not further constrain $\mathbb{X}_{new}$. We use two variations of adaptive MPC to show that only constraining the state set is not enough to successfully complete the proposed task.



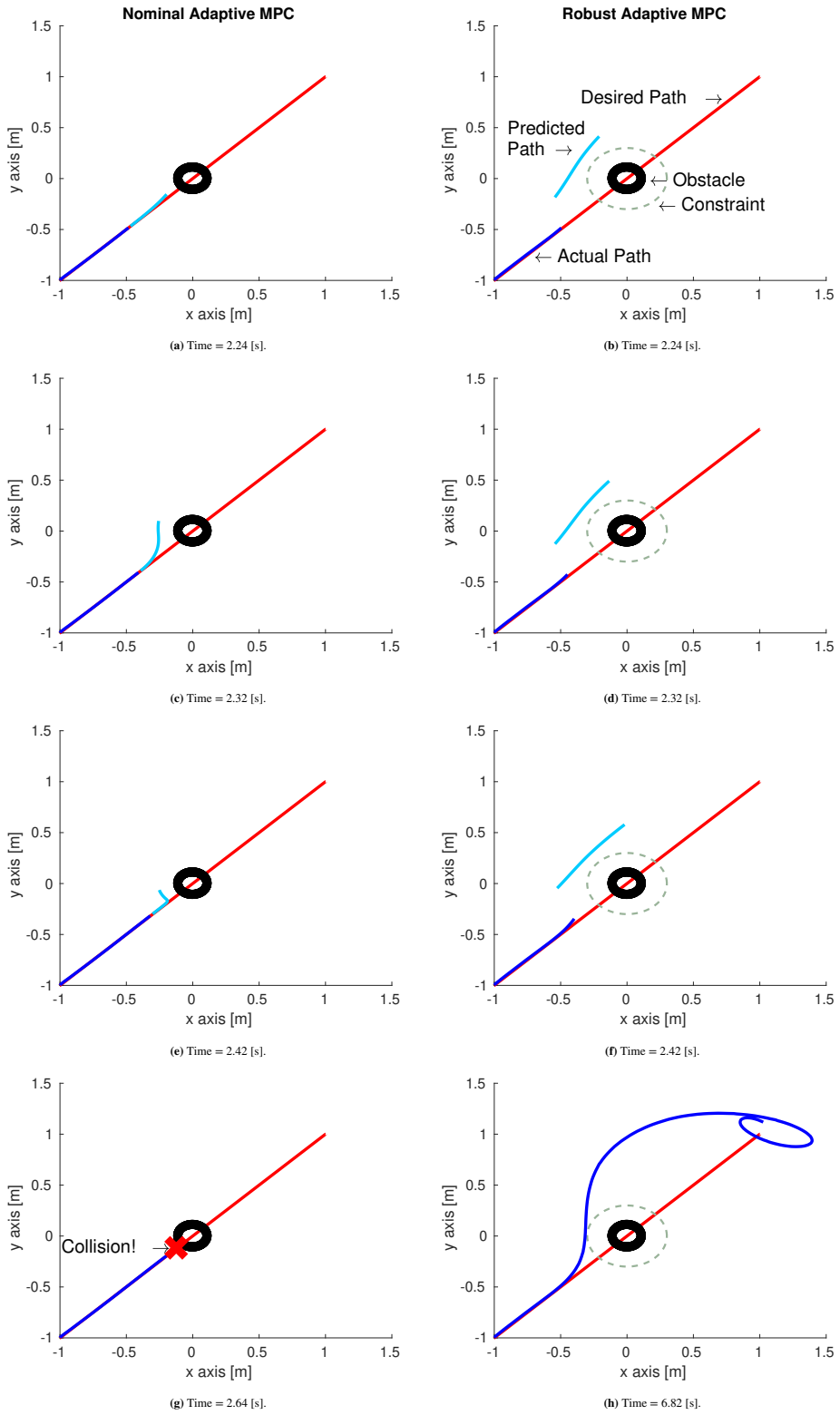
(a) Time = 2.24 [s]. **(b)** Time = 2.24 [s].

(c) Time = 2.32 [s]. **(d)** Time = 2.32 [s].

(e) Time = 2.58 [s]. **(f)** Time = 2.58 [s].

(g) Time = 2.74 [s]. **(h)** Time = 6.82 [s].

FIGURE 7 Quadrotor experiments comparing conservative adaptive MPC and robust adaptive MPC. The goal of this task is to follow the trajectory while avoiding the obstacle. Plots of the x, y plane show the evolution of the system with both controllers. Only constraining the set of allowable states $\mathbb{X}$, as in the conservative adaptive MPC, is not enough to successfully avoid the obstacle. The modeling error in the system must be included, as in the robust adaptive MPC framework, to successfully complete the task.



(a) Time = 2.24 [s]. **(b)** Time = 2.24 [s].

(c) Time = 2.32 [s]. **(d)** Time = 2.32 [s].

(e) Time = 2.42 [s]. **(f)** Time = 2.42 [s].

(g) Time = 2.64 [s]. **(h)** Time = 6.82 [s].

FIGURE 8 Quadrotor experiments comparing nominal adaptive MPC and robust adaptive MPC. The goal of this task is to follow the trajectory while avoiding the obstacle. Plots of the x, y plane show the evolution of the system with both controllers. The nominal adaptive MPC does not further constrain the set of allowable states $\mathbb{X}$, as the robust adaptive MPC does, yet it does not successfully avoid the obstacle. The modeling error in the system must be included, as in the robust adaptive MPC framework, to successfully avoid the obstacle.

In order to compare the performance of the different approaches, we plot in the x, y plane at different timesteps: *(i)* the desired trajectory (in red), *(ii)* the actual trajectory of the drone (in blue), *(iii)* the predicted trajectory at that timestep (in magenta), *(iv)* the obstacle (in black), and *(v)* the boundary of the set $\mathbb{X}_{new} \ominus Z$ (as a dashed black line), where applicable.

The performance at four different times of the changed adaptive MPC and robust MPC is shown in Fig. 7. Fig. 7a shows that at 2.24 seconds, the constrained adaptive MPC optimizes for a trajectory that is predicted to go around the constrained area. At the same time, Fig. 7b shows that the robust MPC predicts a trajectory that also goes around the constrained area. It is clear that the robust MPC changes the initial state in the optimization. Note that the predicted trajectories shown are the ones that the nominal and robust MPC calculate. They do not reflect the 'real' predicted behavior of the system as the inputs change at each time step. At 2.32 seconds, the constrained adaptive MPC still predicts a trajectory that goes around the constrained area, as shown in Fig. 7c. However, the real state of the system and the ideal state start to deviate and the adaptive MPC is unable to compute a trajectory robust enough given the increased modeling error. At 2.58 seconds the drone goes into the constrained area and the nominal MPC becomes an unfeasible problem, as seen in Fig. 7e. Finally, the drone with the changed adaptive MPC collides with the obstacle (see Fig. 7g) while the drone with the robust adaptive MPC is able to avoid the obstacle (see Fig. 7h). The behavior of the robust adaptive MPC is oscillatory at the end of the trajectory because the optimization aims to minimize the trajectory tracking error. Hence, the robust adaptive MPC generates large inputs to bring the system close to the desired trajectory faster causing oscillations.

Next, we show in Fig. 8 a comparison between the nominal adaptive MPC and robust adaptive MPC at different times in the trajectory. At 2.24 seconds the nominal adaptive MPC (see Fig. 8a) and the robust adaptive MPC (see Fig. 8b) predict trajectories that will not collide with the obstacle. A similar behavior is shown at 2.32 and 2.42 seconds for the nominal adaptive MPC (see Fig. 8c and Fig. 8e) and robust adaptive MPC (see Fig. 8d and Fig. 8f). However, the nominal adaptive MPC is not able to generate a signal robust to model inaccuracies and the drone crashes at 2.64 seconds as seen in Fig. 8g. The proposed robust adaptive MPC is able to make the drone avoid the obstacle and finish the trajectory successfully, as seen in Fig. 8h.

6.4 | Obstacle Avoidance with Trajectory Update

We also assess the performance of the controllers when an obstacle is introduced in the environment and the trajectory is modified to avoid the obstacle. Fig. 9a shows the obstacle and the desired trajectory in 3D space. Fig. 9b shows a top view into the x, y plane of the obstacle (in black), and the desired trajectory (in red). The desired trajectory surrounds the obstacle. The drone must avoid the obstacle during the flight in order to successfully complete the task. In this experiment we also constrain the allowable states set $\mathbb{X}$ to exclude the positions that lie inside the obstacle. The obstacle is described as in (72).

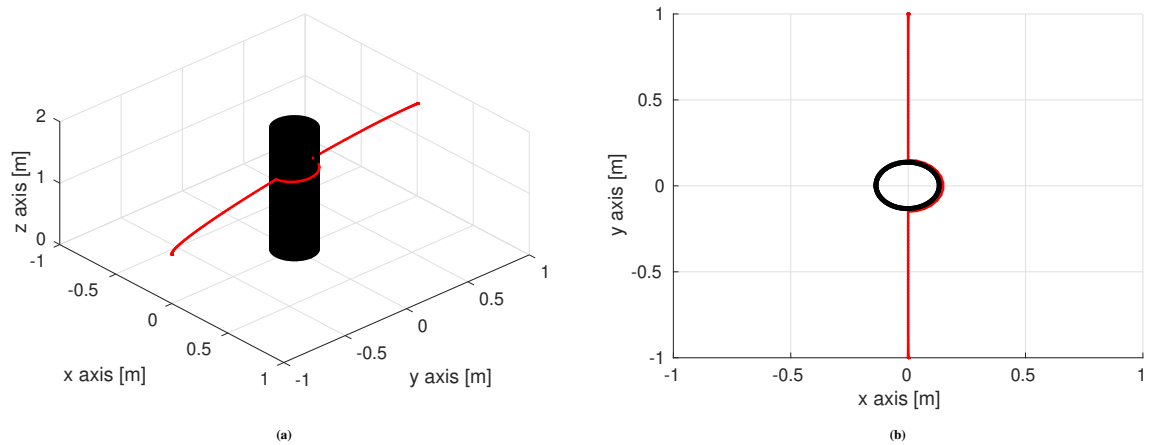
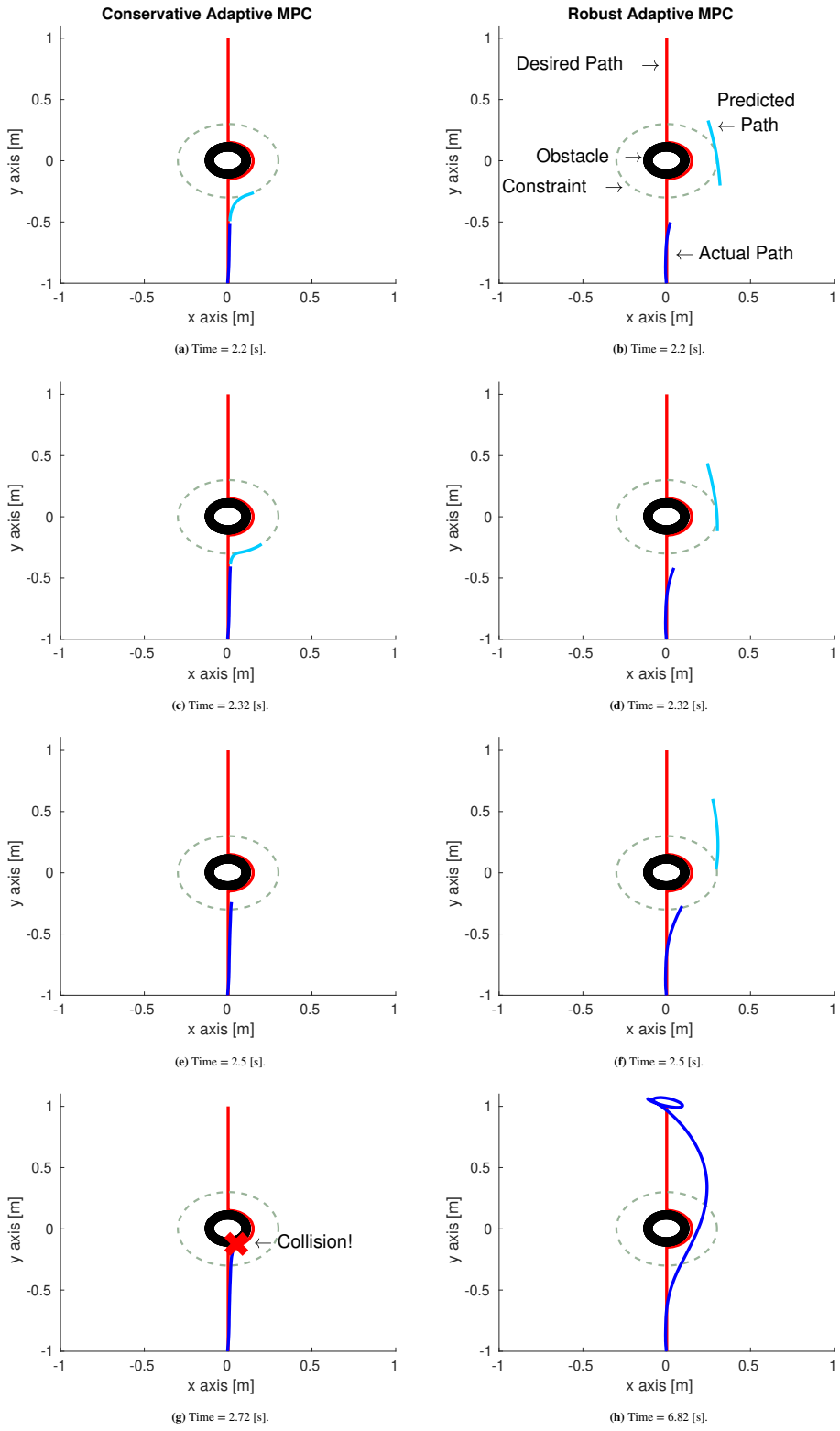


FIGURE 9 The drone is required to fly a 3D trajectory as shown in Fig. 9a. The trajectory goes around the obstacle in order to avoid colliding with it. Fig. 9b is the top view and shows that the trajectory goes around the obstacle. The drone must avoid the obstacle to successfully complete the task.



(a) Time = 2.2 [s].

(b) Time = 2.2 [s].

(c) Time = 2.32 [s].

(d) Time = 2.32 [s].

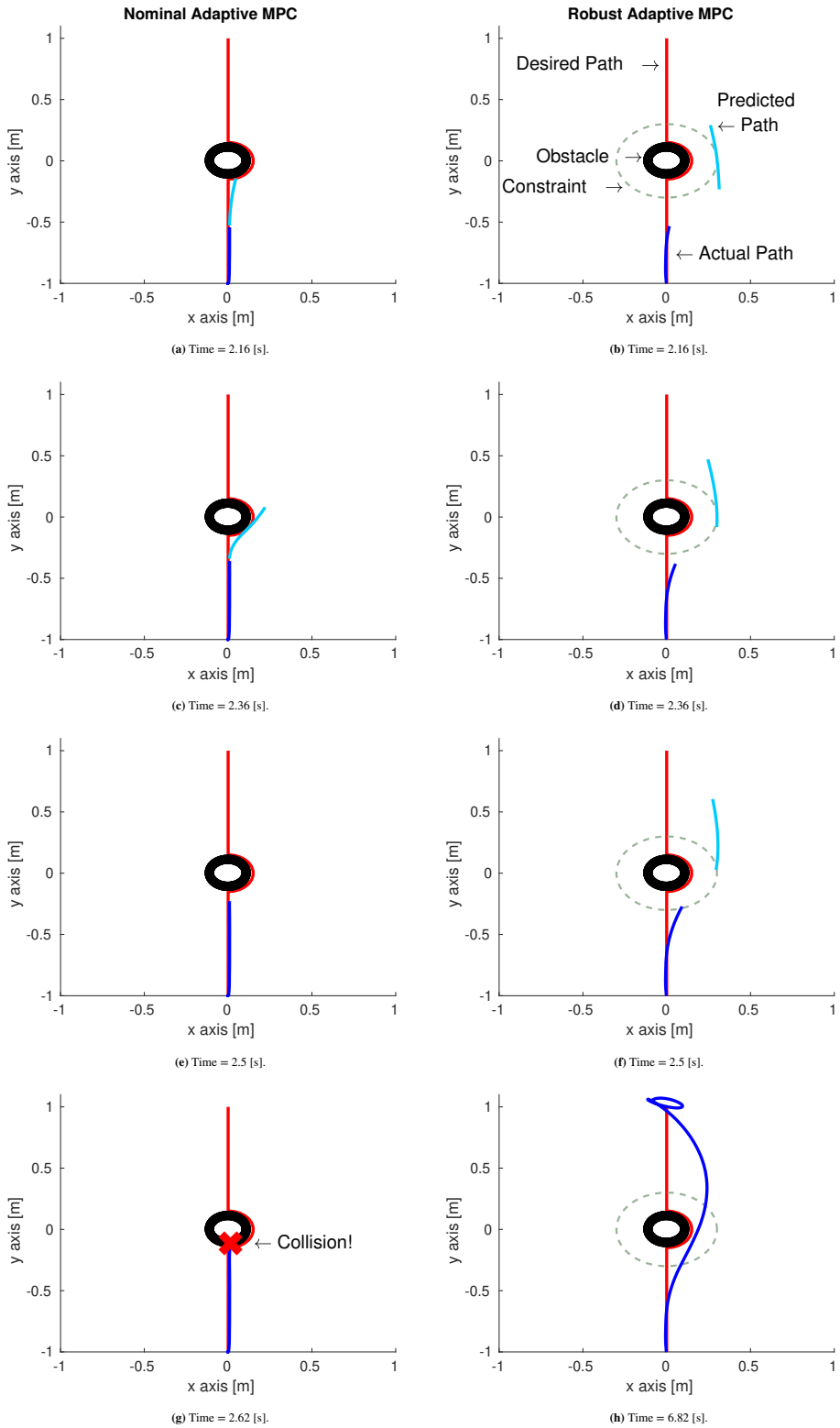
(e) Time = 2.5 [s].

(f) Time = 2.5 [s].

(g) Time = 2.72 [s].

(h) Time = 6.82 [s].

FIGURE 10 Quadrotor experiments comparing conservative adaptive MPC and robust adaptive MPC. The goal of this task is to follow the trajectory while avoiding the obstacle. In this experiment the trajectory is modified to avoid the obstacle. Plots of the x, y plane show the evolution of the system with both controllers. Only constraining the set of allowable states $\mathbb{X}$, as in the conservative adaptive MPC, is not enough to successfully avoid the obstacle. The modeling error in the system must be included, as in the robust adaptive MPC framework to successfully complete the task.



(a) Time = 2.16 [s].

(b) Time = 2.16 [s].

(c) Time = 2.36 [s].

(d) Time = 2.36 [s].

(e) Time = 2.5 [s].

(f) Time = 2.5 [s].

(g) Time = 2.62 [s].

(h) Time = 6.82 [s].

FIGURE 11 Quadrotor experiments comparing nominal adaptive MPC and robust adaptive MPC. The goal of this task is to follow the trajectory while avoiding the obstacle. In this experiment the trajectory is modified to avoid the obstacle. Plots of the x, y plane show the evolution of the system with both controllers. The nominal adaptive MPC does not further constrain the set of allowable states $\mathbb{X}$, as the robust adaptive MPC does, yet it does not successfully avoid the obstacle. Including the modeling error of the system in the robust adaptive MPC framework enables the system to successfully complete the task.

The performance at four different times of the conservative adaptive MPC and robust adaptive MPC is shown in Fig. 10. The conservative adaptive MPC and the robust adaptive MPC plan trajectories to go around the constrained area at 2.2 seconds (see Fig. 10a and Fig. 10b), and 2.32 seconds (see Fig. 10c and Fig. 10d). The conservative adaptive MPC is unable to steer the vehicle from entering the constrained area and the problem becomes infeasible at 2.5 seconds, see Fig. 10e. The vehicle with conservative adaptive MPC collides with the obstacle at 2.72 seconds, as shown in Fig. 10g. The drone with the robust adaptive MPC is able to avoid the obstacle as shown in Fig. 10h because it takes into account the modeling error.

Similarly, the nominal adaptive MPC is unable to steer the vehicle and avoid collision with the obstacle, as shown in Fig. 11. The nominal adaptive MPC plans a trajectory that avoids the obstacle as shown in Fig. 11a and Fig. 11c. Since the disturbances in the system are not included in the optimization, the input calculated by the nominal adaptive MPC is not able to make the vehicle avoid the obstacle and it finally collides at 2.62 seconds as shown in Fig. 11g. However, robust adaptive MPC is able to plan a trajectory that helps the drone to successfully complete the task.

7 | CONCLUSION

In this paper we introduced a novel robust adaptive MPC framework to achieve safe and accurate trajectory tracking in the presence of modeling errors. The robust adaptive MPC consists of an underlying discrete time state feedback ℓ_1 adaptive controller and a robust MPC. In this work we *(i)* introduce a modified discrete time state feedback ℓ_1 adaptive controller, *(ii)* provide its stability and performance proofs, *(iii)* use the performance proofs in a robust MPC framework, and *(iv)* show the performance of the proposed robust adaptive MPC with quadrotor experiments. Ideally, the discrete time state feedback ℓ_1 adaptive controller makes the system behave as a linear reference model. In reality, the behavior of the real system may deviate from the linear reference model. In this work, we show that this deviation is uniformly bounded and can be thought of as modeling error. A robust MPC is then used to compute an optimal input that minimizes the tracking error of the system while taking into account the modeling error. In experiments on a quadrotor we show that the proposed robust adaptive MPC achieves high accuracy trajectory tracking on six different trajectories. We compare the proposed robust adaptive MPC to an adaptive MPC. When an obstacle is introduced in the environment, the adaptive MPC framework is not able to make the quadrotor avoid collision while the robust adaptive MPC includes the modeling error and is able to compute an input that successfully avoids the obstacle.

ACKNOWLEDGMENTS

This is acknowledgment text.

Author contributions

This is an author contribution text.

Financial disclosure

None reported.

Conflict of interest

The authors declare no potential conflict of interests.

SUPPORTING INFORMATION

The following supporting information is available as part of the online article:
Figure S1. 500 hPa geopotential anomalies for GC2C calculated against the ERA Interim reanalysis. The period is 1989–2008.
Figure S2. The SST anomalies for GC2C calculated against the observations (OIsst).

How to cite this article: Pereida K., and A.Schoellig (2019), Robust adaptive model predictive control for safe and high accuracy tracking in the presence of model errors, *The journal.*, *2017;00:1–6*.

APPENDIX

A SUPPLEMENTAL MATERIALS

A.1 Proof of Lemma 1

The proof of Lemma 1 is presented next.

Proof. Define

$$F \triangleq \tfrac{1}{\sigma} P^{\frac{1}{2}} A_m \;\; G \triangleq \sigma P^{\frac{1}{2}} b_m \;\; J(\tilde{x}(k)) \triangleq \tilde{x}^T(k) P \tilde{x}(k)\,. \tag{A1}$$

Then,

$$\begin{aligned}
\Delta J_{\tilde{x}}(k) &\triangleq \tilde{x}^T(k+1)P\tilde{x}(k+1) - \tilde{x}^T(k)P\tilde{x}(k)\\
&= (A_m\tilde{x}(k) + b_m(\tilde{\theta}^T(k)x(k) + \tilde{\xi}(k)))^T P(A_m\tilde{x}(k) + b_m(\tilde{\theta}^T(k)x(k) + \tilde{\xi}(k))) - \tilde{x}^T(k)P\tilde{x}(k)\\
&= \tilde{x}^T(k)A_m^T P A_m\tilde{x}(k) + \tilde{x}^T(k)A_m^T P b_m(\tilde{\theta}^T(k)x(k) + \tilde{\xi}(k)) + (x^T(k)\tilde{\theta}(k) + \tilde{\xi}(k))b_m^T P A_m\tilde{x}(k)\\
&\quad +(x^T(k)\tilde{\theta}(k) + \tilde{\xi}(k))b_m^T P b_m(\tilde{\theta}^T(k)x(k) + \tilde{\xi}(k)) - \tilde{x}^T(k)P\tilde{x}(k)\\
&= \tilde{x}^T(k)(A_m^T P A_m - P)\tilde{x}(k) + \tilde{x}^T(k)A_m^T P b_m(x^T(k)\tilde{\theta}(k) + \tilde{\xi}(k))\\
&\quad +(x^T(k)\tilde{\theta}(k) + \tilde{\xi}(k))b_m^T P A_m\tilde{x}(k) + (x^T(k)\tilde{\theta}(k) + \tilde{\xi}(k))^2 b_m^T P b_m\\
&= \tilde{x}^T(k)(A_m^T P A_m - P + F^T F)\tilde{x}(k) - \tilde{x}^T(k)F^T F\tilde{x}(k) + \tilde{x}^T(k)F^T G(x^T(k)\tilde{\theta}(k) + \tilde{\xi}(k))\\
&\quad +(x^T(k)\tilde{\theta}(k) + \tilde{\xi}(k))G^T F\tilde{x}(k) + (x^T(k)\tilde{\theta}(k) + \tilde{\xi}(k))^2 b_m^T P b_m + (\tilde{x}^T(k)\tilde{\theta}(k) + \tilde{\xi}(k))^2 G^T G\\
&\quad -(\tilde{x}^T(k)\tilde{\theta}(k) + \tilde{\xi}(k))^2 G^T G\\
&= \tilde{x}^T(k)(A_m^T P A_m - P + F^T F)\tilde{x}(k) + (x^T(k)\tilde{\theta}(k) + \tilde{\xi}(k))^2(b_m^T P b_m + G^T G)\\
&\quad - \begin{bmatrix}\tilde{x}^T(k) & -(x^T(k)\tilde{\theta}(k) + \tilde{\xi}(k))\end{bmatrix}\begin{bmatrix}F^T F & F^T G\\ G^T F & G^T G\end{bmatrix}\begin{bmatrix}\tilde{x}(k)\\ -(x^T(k)\tilde{\theta}(k) + \tilde{\xi}(k))\end{bmatrix}\\
&\leq \tilde{x}^T(k)(A_m^T P A_m - P + F^T F)\tilde{x}(k) + (x^T(k)\tilde{\theta}(k) + \tilde{\xi}(k))^2(b_m^T P b_m + G^T G)\,.
\end{aligned} \tag{A2}$$

The last inequality is possible by using the Schur complement condition for positive definitiveness. Note that

$$F^T F = \frac{1}{\sigma^2} A_m^T P A_m = \frac{A_m^T P A_m}{\lambda_{\max}(A_m^T P A_m)} \leq \frac{I_n \lambda_{\max}(A_m^T P A_m)}{\lambda_{\max}(A_m^T P A_m)} = I_n\,. \tag{A3}$$

It follows from (15) that

$$A_m^T P A_m - P + F^T F \leq A_m^T P A_m - P + I_n = -R\,. \tag{A4}$$

Therefore,

$$\tilde{x}^T(k)(A_m^T P A_m - P + F^T F)\tilde{x}(k) \leq -\tilde{x}(k)R\tilde{x}(k)\,, \tag{A5}$$

which implies

$$\Delta J_{\tilde{x}}(k) \leq -\tilde{x}^T(k)R\tilde{x}(k) + (x^T(k)\tilde{\theta}(k) + \tilde{\xi}(k))^2(b_m^T P b_m + G^T G)\,, \quad G^T G = \sigma^2 b_m^T P b_m\,, \tag{A6}$$

then

$$\Delta J_{\tilde{x}}(k) \leq -\tilde{x}(k)R\tilde{x}(k) + (x^T(k)\tilde{\theta}(k) + \tilde{\xi}(k))^2((1+\sigma^2)b_m^T P b_m)\,. \tag{A7}$$

Since $\ln x \leq x - 1$ for all $x \geq 0$,

$$\begin{aligned}\Delta V_x(\tilde{x}(k)) &= \ln(1+\mu\tilde{x}^T(k+1)P\tilde{x}(k+1)) - \ln(1+\mu\tilde{x}^T(k)P\tilde{x}(k)) \\ &= \ln\left(\frac{1+\mu\tilde{x}^T(k+1)P\tilde{x}(k+1)}{1+\mu\tilde{x}^T(k)P\tilde{x}(k)}\right) = \ln\left(1+\frac{\mu\Delta J_{\tilde{x}}(k)}{1+\mu\tilde{x}^T(k)P\tilde{x}(k)}\right) \\ &\leq \frac{\mu\Delta J_{\tilde{x}}(k)}{1+\mu\tilde{x}^T(k)P\tilde{x}(k)} \leq \mu\frac{[-\tilde{x}(k)R\tilde{x}(k) + (x^T(k)\tilde{\theta}(k)+\tilde{\xi}(k))^2((\sigma^2+1)b_m^TPb_m)]}{1+\mu\tilde{x}^T(k)P\tilde{x}(k)}.\end{aligned} \tag{A8}$$

□

A.2 Proof of Lemma 2

The proof of Lemma 2 is presented next.

Proof. Using the z-transform and (9), we can write (12) as

$$\hat{x}(z) = G(z)\hat{\eta}(z) + k_g H(z)C(z)r(z) + x_{in}(z). \tag{A9}$$

For all $i \in \mathbb{N} \cup \{0\}$, the following bounds hold:

$$\begin{aligned}\|\hat{x}(k)\|_\infty &\leq \|G(z)\|_{\ell_1}\|\hat{\eta}(k)\|_\infty + \|k_g H(z)C(z)\|_{\ell_1}\|r(k)\|_\infty + \|x_{in}(k)\|_\infty, \\ \|\hat{\eta}(k)\|_\infty &\leq L_\theta\left(\|\tilde{x}(k)\|_\infty + \|\hat{x}(k)\|_\infty\right) + L_\xi, \\ \|\hat{x}(k)\|_\infty &\leq \tfrac{1}{1-\lambda_\theta}\left(\lambda_\theta\|\tilde{x}(k)\|_\infty + \|G(z)\|_{\ell_1}L_\xi + \|k_g H(z)C(z)\|_{\ell_1}\|r(k)\|_\infty + \|x_{in}(k)\|_\infty\right), \\ \|\hat{x}(k)\|_\infty &\leq \tfrac{1}{1-\lambda_\theta}\left(\lambda_\theta\|\tilde{x}(k)\|_\infty + \|G(z)\|_{\ell_1}L_\xi + \|k_g H(z)C(z)\|_{\ell_1}\|r\|_{\ell_\infty} + \|x_{in}\|_{\ell_\infty}\right), \\ \|\hat{x}(k)\|_\infty &\leq c_1\|\tilde{x}(k)\|_\infty + c_2, \\ \text{where}\quad c_1 &\triangleq \tfrac{\lambda_\theta}{1-\lambda_\theta}, \quad c_2 \triangleq \tfrac{1}{1-\lambda_\theta}\left(\|G(z)\|_{\ell_1}L_\xi + \|k_g H(z)C(z)\|_{\ell_1}\|r\|_{\ell_\infty} + \|x_{in}\|_{\ell_\infty}\right),\end{aligned} \tag{A10}$$

where $\|\cdot\|_\infty$ is the vector ∞-norm. Then,

$$\|\hat{x}(k)\|_\infty^2 \leq 2c_1^2\|\tilde{x}(k)\|_\infty^2 + 2c_2^2. \tag{A11}$$

Notice that we require $n \geq l > 0$ for the following inequality to hold,

$$\hat{x}^T(k)\hat{x}(k) \leq l\|\hat{x}(k)\|_\infty^2, \quad \|\tilde{x}(k)\|_\infty^2 \leq \tilde{x}^T(k)\tilde{x}(k). \tag{A12}$$

Hence,

$$\hat{x}^T(k)\hat{x}(k) \leq 2lc_1^2\|\tilde{x}(k)\|_\infty^2 + 2lc_2^2 \leq 2lc_1^2\tilde{x}^T(k)\tilde{x}(k) + 2lc_2^2. \tag{A13}$$

Then,

$$\begin{aligned}x^T(k)x(k) &\leq 2\hat{x}^T(k)\hat{x}(k) + 2\tilde{x}^T(k)\tilde{x}(k) \leq (4lc_1^2+2)\tilde{x}^T(k)\tilde{x}(k) + 4lc_2^2, \\ x^T(k)x(k) &\leq \alpha\beta + \alpha\tilde{x}^T(k)\tilde{x}(k), \quad \alpha \triangleq 4lc_1^2 + 2 > 0, \quad \beta \triangleq \frac{4lc_2^2}{4lc_1^2+2} > 0.\end{aligned} \tag{A14}$$

Since $1 + \alpha\beta + \alpha\tilde{x}^T(k)\tilde{x}(k) = (1+\alpha\beta)(1 + \frac{\alpha}{1+\alpha\beta}\tilde{x}^T(k)\tilde{x}(k))$, then,

$$1 + x^T(k)x(k) \leq (1+\alpha\beta)(1 + \frac{\alpha}{1+\alpha\beta}\tilde{x}^T(k)\tilde{x}(k)) \leq (1+\alpha\beta)(1+\mu\tilde{x}^T(k)P\tilde{x}(k)), \tag{A15}$$

where

$$\mu = \frac{\alpha}{(1+\alpha\beta)(\lambda_{\min}(P))} > 0, \tag{A16}$$

where $\lambda_{\min}(P)$ is the minimum eigenvalue of P, which completes the proof. □

A.3 Proof of Lemma 3

The proof of Lemma 3 is found next.

Proof. Consider the following Lyapunov function candidate:

$$V(\tilde{x}(k), \tilde{\rho}(k)) \triangleq V_x(\tilde{x}(k)) + wV_\rho(\tilde{\rho}(k)), \tag{A17}$$

where

$$V_\rho(\tilde{\rho}(k)) \triangleq \tilde{\rho}^T(k)\tilde{\rho}(k), \tag{A18}$$

and $V_x(\tilde{x}(k))$ is defined in (17). We also define

$$\Delta V(\tilde{x}(k), \tilde{\rho}(k)) = \Delta V_x(\tilde{x}(k)) + w\Delta V_\rho(\tilde{\rho}(k)), \tag{A19}$$

where

$$\Delta V_\rho(\tilde{\rho}(k)) = V_\rho(\tilde{\rho}(k+1)) - V_\rho(\tilde{\rho}(k)), \tag{A20}$$

and $\Delta V_x(\tilde{x}(k))$ is defined in (18). The orthogonal projection (6) in the adaptation law that keeps the estimate $\hat{\rho}(k)$ in the set $\Theta \times \Xi$ incurs in a lower value of $\Delta V_\rho(\tilde{\rho}'(k+1))$ than when no orthogonal projection is used $\Delta V_\rho(\tilde{\rho}(k+1))$. The latter means that only the upper bound for $\Delta V_\rho(\tilde{\rho}(k+1))$ is needed to draw conclusions for both cases. Note that from (7), we can write:

$$\tilde{\rho}'^T(k+1)\tilde{\rho}'(k+1) \leq \tilde{\rho}^T(k+1)\tilde{\rho}(k+1), \tag{A21}$$

where $\tilde{\rho}'(k+1) = \tilde{\rho}'(k+1) - \rho$. Hence, if the orthogonal projection is used,

$$\Delta V_\rho(\tilde{\rho}'(k)) = \tilde{\rho}'^T(k+1)\tilde{\rho}'(k+1) - \tilde{\rho}^T(k)\tilde{\rho}(k) \leq \tilde{\rho}^T(k+1)\tilde{\rho}(k+1) - \tilde{\rho}^T(k)\tilde{\rho}(k) = \Delta V_\rho(\tilde{\rho}(k)). \tag{A22}$$

Since $\Delta V_\rho(\tilde{\rho}'(k)) \leq \Delta V_\rho(\tilde{\rho}(k))$, we will focus the analysis on $\Delta V_\rho(\tilde{\rho}(k))$. From (4) and (5), we can write:

$$\begin{aligned}
\tilde{\rho}(k+1) &= \tilde{\rho}(k) + \frac{\begin{bmatrix} x(k) \\ 1 \end{bmatrix} \left[b_0^T \left(b_m \rho^T \begin{bmatrix} x(k) \\ 1 \end{bmatrix} \right) - \hat{\rho}^T(k) \begin{bmatrix} x(k) \\ 1 \end{bmatrix} \right]}{1 + x^T(k)x(k)} \\
&= \tilde{\rho}(k) + \frac{\begin{bmatrix} x(k) \\ 1 \end{bmatrix} \left[(\rho^T - \hat{\rho}^T(k)) \begin{bmatrix} x(k) \\ 1 \end{bmatrix} \right]}{1 + x^T(k)x(k)} = \tilde{\rho}(k) - \frac{\begin{bmatrix} x(k) \\ 1 \end{bmatrix} \tilde{\rho}^T(k) \begin{bmatrix} x(k) \\ 1 \end{bmatrix}}{1 + x^T(k)x(k)} \\
&= \left(\underbrace{I_{n+1} - \frac{\begin{bmatrix} x(k) \\ 1 \end{bmatrix} \begin{bmatrix} x(k) \\ 1 \end{bmatrix}^T}{1 + x^T(k)x(k)}}_{\Phi} \right) \tilde{\rho}(k).
\end{aligned} \tag{A23}$$

Note that $\Phi^T = \Phi > 0$. Then, $\Delta V_\rho(\tilde{\rho}(k)) = \tilde{\rho}^T(k+1)\tilde{\rho}(k+1) - \tilde{\rho}^T(k)\tilde{\rho}(k) = \tilde{\rho}^T(k)\Phi^T\Phi\tilde{\rho}(k) - \tilde{\rho}^T(k)\tilde{\rho}(k) = \tilde{\rho}^T(k)(\Phi^T\Phi - I_{n+1})\tilde{\rho}(k)$. Also note that since $0 < \Phi \leq I_{n+1}$, then $0 \leq \Phi(I_{n+1} - \Phi)$ and $\Phi\Phi \leq \Phi$. Consequently, $\Phi\Phi - I_{n+1} \leq \Phi - I_{n+1}$ and

$$\Delta V_\rho(\tilde{\rho}(k)) \leq \tilde{\rho}^T(k)(\Phi - I_{n+1})\tilde{\rho}(k) = -\tilde{\rho}^T(k)\frac{\begin{bmatrix} x(k) \\ 1 \end{bmatrix} \begin{bmatrix} x(k) \\ 1 \end{bmatrix}^T}{1 + x^T(k)x(k)}\tilde{\rho}(k) = -\frac{\left(\tilde{\rho}^T(k) \begin{bmatrix} x(k) \\ 1 \end{bmatrix} \right)^2}{1 + x^T(k)x(k)}. \tag{A24}$$

Define

$$w \triangleq \mu(1 + \alpha\beta)(\sigma^2 + 1)b_m^T P b_m. \tag{A25}$$

From (19) and (A24) and using (20) in Lemma 2, we have

$$\begin{aligned}
\Delta V(\tilde{x}(k),\tilde{\rho}(k)) &\leq \mu\frac{-\tilde{x}^T(k)R\tilde{x}(k)+\left(\tilde{\rho}^T(k)\begin{bmatrix}x(k)\\1\end{bmatrix}\right)^2(\sigma^2+1)b_m^TPb_m}{1+\mu\tilde{x}^T(k)P\tilde{x}(k)}-w\frac{\left(\tilde{\rho}^T(k)\begin{bmatrix}x(k)\\1\end{bmatrix}\right)^2}{1+x^T(k)x(k)}\\
&\leq \mu\frac{-\tilde{x}^T(k)R\tilde{x}(k)+\left(\tilde{\rho}^T(k)\begin{bmatrix}x(k)\\1\end{bmatrix}\right)^2(\sigma^2+1)b_m^TPb_m}{1+\mu\tilde{x}^T(k)P\tilde{x}(k)}-w\frac{\left(\tilde{\rho}^T(k)\begin{bmatrix}x(k)\\1\end{bmatrix}\right)^2}{(1+\alpha\beta)(1+\mu\tilde{x}^T(k)P\tilde{x}(k))}\\
&\leq \mu\frac{-\tilde{x}^T(k)R\tilde{x}(k)+\left(\tilde{\rho}^T(k)\begin{bmatrix}x(k)\\1\end{bmatrix}\right)^2(\sigma^2+1)b_m^TPb_m-\left(\tilde{\rho}^T(k)\begin{bmatrix}x(k)\\1\end{bmatrix}\right)^2(\sigma^2+1)b_m^TPb_m}{1+\mu\tilde{x}^T(k)P\tilde{x}(k)}\\
&\leq \mu\frac{-\tilde{x}^T(k)R\tilde{x}(k)}{1+\mu\tilde{x}^T(k)P\tilde{x}(k)}\leq 0\,.
\end{aligned}\tag{A26}$$

The above implies that $\tilde{x}(k)$ and $\tilde{\rho}(k)$ are uniformly bounded. Since $\tilde{x}(0)=0$, it follows that

$$\begin{aligned}
V(\tilde{x}(k),\tilde{\rho}(k)) &\geq \ln(1+\mu\tilde{x}^T(k)P\tilde{x}(k))\\
(1+\mu\tilde{x}^T(k)P\tilde{x}(k)) &\leq e^{V(\tilde{x}(k),\tilde{\rho}(k))}\\
\mu\tilde{x}^T(k)P\tilde{x}(k) &\leq e^{V(\tilde{x}(k),\tilde{\rho}(k))}-1\,.
\end{aligned}\tag{A27}$$

Finally,

$$\mu\lambda_{\min}(P)\|\tilde{x}(k)\|^2\leq\mu\tilde{x}^T(k)P\tilde{x}(k)\leq e^{V(\tilde{x}(k),\tilde{\rho}(k))}-1\leq e^{V(\tilde{x}(0),\tilde{\rho}(0))}-1\leq e^{w\tilde{\rho}^T(0)\tilde{\rho}(0)}-1\,,\tag{A28}$$

and

$$\|\tilde{x}(k)\|^2\leq\frac{e^{w\tilde{\rho}^T(0)\tilde{\rho}(0)}-1}{\mu\lambda_{\min}(P)}\,.\tag{A29}$$

Note that $\rho\in\Theta\times\Xi$ and $\tilde{\rho}^T(0)\tilde{\rho}(0)\leq\rho_{\max}$. Since $\|.\|_{\ell_\infty}\leq\|.\|$, then

$$\|\tilde{x}\|_{\ell_\infty}\leq\sqrt{\frac{e^{w\rho_{\max}}-1}{\mu\lambda_{\min}(P)}}\,,\tag{A30}$$

which holds uniformly. □

A.4 Proof of Lemma 4

The proof of Lemma 4 is found next.

Proof. Recall that the state predictor (12) can be written in the z domain as:

$$\hat{x}(z)=G(z)\hat{\eta}(z)+H(z)C(z)k_gr(z)+\hat{x}_{in}(z)\tag{A31}$$

which leads to the following upper bound:

$$\|\hat{x}|_i\|_{\ell_\infty}\leq\|G(z)\|_{\ell_1}\|\hat{\eta}|_i\|_{\ell_\infty}+\|k_gH(z)C(z)\|_{\ell_1}\|r|_i\|_{\ell_\infty}+\|\hat{x}_{in}|_i\|_{\ell_\infty}\,.\tag{A32}$$

Applying the triangular relationship for norms to the bound (21), we have:

$$|\|\hat{x}|_i\|_{\ell_\infty}-\|x|_i\|_{\ell_\infty}|\leq\sqrt{\frac{e^{w\rho_{\max}}-1}{\mu\lambda_{\min}(P)}}\,.\tag{A33}$$

The adaptation law in (5) and (6) ensures that $\hat{\rho}(k)\in\Theta\times\Xi$ and $\|\hat{\eta}|_i\|_{\ell_\infty}\leq L_\theta\|x|_i\|_{\ell_\infty}+L_\xi$. Substituting $\|x|_i\|_{\ell_\infty}$ yields

$$\|\hat{\eta}\|_{\ell_\infty}\leq L_\theta\left(\|\hat{x}|_i\|_{\ell_\infty}+\sqrt{\frac{e^{w\rho_{\max}}-1}{\mu\lambda_{\min}(P)}}\right)+L_\xi\,.\tag{A34}$$

Then, using the bounds on $\|\hat{x}|_i\|_{\ell_\infty}$ in (A32) and $\|\hat{\eta}|_i\|_{\ell_\infty}$ in (A34), and the ℓ_1 norm condition in (10), leads to:

$$\|\hat{x}|_i\|_{\ell_\infty} \leq \frac{\lambda_\theta \sqrt{\frac{e^{w\rho_{\max}}-1}{\mu\lambda_{\min}(P)}} + \|G(z)\|_{\ell_1} L_\xi + \|H(z)k_g C(z)\|_{\ell_1} \|r|_i\|_{\ell_\infty} + \|x_{in}|_i\|_{\ell_\infty}}{1-\lambda_\theta}. \tag{A35}$$

Since the bound on the right hand side is uniform, then $\hat{x}(k)$ is uniformly bounded. □

A.5 Derivation of $H_1(z)$

We first look at a special case of state-to-input stability for linear time-invariant (LTI) systems. Consider an LTI system given by

$$x(z) = (zI - A)^{-1} b u(z), \tag{A36}$$

where $x(z)$, $u(z)$ are the z-transforms of the system state $x(k)$ and input $u(k)$, $A \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$, and assume that:

$$G(z) = (zI - A)^{-1} b = \frac{N(z)}{D(z)}, \tag{A37}$$

where $D(z) = \det(zI - A)$ using Cramer's rule, and $N(z)$ is a $n \times 1$ vector with its i^{th} element being a polynomial function

$$N_i(z) = \sum_{j=1}^{n} N_{ij} z^{j-1}. \tag{A38}$$

Lemma 6. If (A, b) is controllable, then the matrix N of entries N_{ij} is full rank.

Proof. Controllability of (A, b) implies that given an initial condition $x(0) = 0$, and arbitrary k_1 and x_{k_1}, there exists $u(\kappa)$, $\kappa \in [0, k_1]$, such that $x(k_1) = x_{k_1}$. If N is not full rank, then there exists a $\psi \in \mathbb{R}^n$, $\psi \neq 0$, such that $\psi^T N(z) = 0$. Thus, for $x(0) = 0$ we have

$$\psi^T x(z) = \psi^T \frac{N(z)}{D(z)} u(z) = 0, \quad \forall u(z), \tag{A39}$$

which implies that $x(\kappa) \neq x_{k_1}$ for any κ. The latter contradicts controllability where $x(k_1) = x_{k_1}$ can be an arbitrary point in $\mathbb{R}^n$. As a result, N must be full rank. □

Corollary 1. If the pair (A, b) in (A36) is controllable, then there exists $c_0 \in \mathbb{R}^n$, such that $c_0^T \frac{N(z)}{D(z)}$ has relative degree one, i.e., $\deg(D(z)) - \deg(c_0^T N(z)) = 1$, and $c_0^T N(z)$ has all its zeros in the unit disk.

Proof. It follows from (A37) that for arbitrary vector $c_0 \in \mathbb{R}^n$:

$$c_0^T (zI - A)^{-1} b = \frac{c_0^T N [z^{n-1} \dots 1]^T}{D(z)}, \tag{A40}$$

where $N \in \mathbb{R}^{n\times n}$ is the matrix with its i^{th} row j^{th} column entry N_{ij} introduced in (A38). Since (A, b) is controllable, N is full rank, from Lemma 6. Consider an arbitrary vector $\bar{c} \in \mathbb{R}^n$ such that $\bar{c}[z^{n-1} \dots 1]^T$ is a stable $n-1$ order polynomial, and let $c_0 = (N^{-1})^T \bar{c}$. Then

$$c_0^T (zI - A)^{-1} b = \frac{\bar{c}^T [z^{n-1} \dots 1]^T}{D(z)} \tag{A41}$$

has relative degree 1 with all its zeros in the unit disk. □

Lemma 7. Let the pair (A, b) be controllable and $F(z)$ be an arbitrary strictly-proper BIBO stable transfer function. Then, there exists a proper and stable $G_1(z)$, given by

$$G_1(z) \triangleq \frac{F(z)}{c_0^T G(z)} c_0^T, \tag{A42}$$

where $c_0 \in \mathbb{R}^n$, and $c_0^T G(z)$ is a minimum phase transfer function with relative degree 1, such that

$$F(z)u(z) = G_1(z)x(z). \tag{A43}$$

Proof. From Corollary 1, it follows that there exists $c_0 \in \mathbb{R}^n$ such that $c_0^T G(z)$ has relative degree one, and $c_0^T G(z)$ has all its zeros in the unit disk. Hence,

$$F(z)u(z) = F(z)\frac{c_0^T G(z)}{c_0^T G(z)} u(z) = G_1(z)x(z), \tag{A44}$$

where the properness of $G_1(z)$ is ensured by the fact that $F(z)$ is strictly-proper, while stability follows immediately from its definition. □

Since the pair (A_m, b_m) in (1) is controllable, Lemma 7 implies that

$$H_1(z) \triangleq C(z)\frac{1}{c_0^T H(z)}c_0^T, \tag{A45}$$

is proper and BIBO stable.

References

1. Skelton R. Model error concepts in control design. *International Journal of Control* 1989; 49(5): 1725–1753.
2. Skogestad S, Postlethwaite I. *Multivariable feedback control: analysis and design*. 2. Wiley New York . 2007.
3. Hovakimyan N, Cao C. $\mathcal{L}_1$ *Adaptive Control Theory: Guaranteed Robustness with Fast Adaptation*. Philadelphia, PA: Society for Industrial and Applied Mathematics . 2010.
4. Michini B, How JP. $\mathcal{L}_1$ Adaptive Control for Indoor Autonomous Vehicles: Design Process and Flight Testing. In: ; 2009: 5754.
5. Jafarnejadsani H, Sun D, Lee H, Hovakimyan N. Optimized $\mathcal{L}_1$ Adaptive Controller for Trajectory Tracking of an Indoor Quadrotor. *Journal of Guidance, Control, and Dynamics* 2017; 40(6): 1415–1427.
6. Maalouf D, Chemori A, Creuze V. L1 adaptive depth and pitch control of an underwater vehicle with real-time experiments. *Ocean Engineering* 2015; 98: 66–77.
7. Harris J, Elliott CM, Tallant GS. Stability and Performance Robustness of an L1 Adaptive Dynamic Inversion Flight Control System. In: ; 2019: 0141.
8. Nguyen Q, Sreenath K. L1 adaptive control for bipedal robots with control Lyapunov function based quadratic programs. In: ; 2015: 862–867.
9. Mallikarjunan S, Nesbit B, Kharisov E, Xargay E, Hovakimyan N, Cao C. $\mathcal{L}_1$ Adaptive Controller for Attitude Control of Multirotors. In: ; 2012; Minneapolis, MN.
10. Jafarnejadsani H, Hovakimyan N. Optimal Filter Design for a Discrete-Time Formulation of L1-Adaptive Control. In: 2015 (pp. 0119).
11. Jafarnejadsani H, Lee H, Hovakimyan N. An $\mathcal{L}_1$ adaptive control design for output-feedback sampled-data systems. In: ; 2017: 5744–5749.
12. Morari M, Lee JH. Model predictive control: past, present and future. *Computers & Chemical Engineering* 1999; 23(4): 667–682.
13. Mayne DQ, Rawlings JB, Rao CV, Scokaert PO. Constrained model predictive control: stability and optimality. *Automatica* 2000; 36(6): 789–814.
14. Camacho EF, Alba CB. *Model predictive control*. Springer Science & Business Media . 2013.
15. Bemporad A, Morari M. Robust model predictive control: A survey. In: Springer. 1999 (pp. 207–226).
16. Marruedo DL, Alamo T, Camacho E. Input-to-state stable MPC for constrained discrete-time nonlinear systems with bounded additive uncertainties. In: . 4. ; 2002: 4619–4624.
17. Gilbert EG, Kolmanovsky I, Tan KT. Discrete-time reference governors and the nonlinear control of systems with state and control constraints. *International Journal of Robust and Nonlinear Control* 1995; 5(5): 487–504.

18. Lee JH, Yu Z. Worst-case formulations of model predictive control for systems with bounded parameters. *Automatica* 1997; 33(5): 763–781.

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

20. Scokaert PO, Mayne D. Min-max feedback model predictive control for constrained linear systems. *IEEE Transactions on Automatic control* 1998; 43(8): 1136–1142.

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

22. Limon D, Alvarado I, Alamo T, Camacho E. Robust tube-based MPC for tracking of constrained linear systems with additive disturbances. *Journal of Process Control* 2010; 20(3): 248–260.

23. Sun Z, Dai L, Liu K, Xia Y, Johansson KH. Robust MPC for tracking constrained unicycle robots with additive disturbances. *Automatica* 2018; 90: 172–184.

24. Goodwin GC, Kong H, Mirzaeva G, Seron MM. Robust model predictive control: reflections and opportunities. *Journal of Control and Decision* 2014; 1(2): 115–148.

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

26. Zhu B, Xia X. Adaptive model predictive control for unconstrained discrete-time linear systems with parametric uncertainties. *IEEE Transactions on Automatic Control* 2015; 61(10): 3171–3176.

27. Tanaskovic M, Sturzenegger D, Smith R, Morari M. Robust adaptive model predictive building climate control. *Proc. of the International Federation of Automatic Control (IFAC) World Congress* 2017; 50(1): 1871–1876.

28. Bujarbaruah M, Zhang X, Tseng HE, Borrelli F. Adaptive MPC for Autonomous Lane Keeping. *arXiv preprint arXiv:1806.04335* 2018.

29. Pereida K, Helwa MK, Schoellig AP. Data-efficient multirobot, multitask transfer learning for trajectory tracking. *IEEE Robotics and Automation Letters* 2018; 3(2): 1260–1267.

30. Pereida K, Schoellig AP. Adaptive Model Predictive Control for High-Accuracy Trajectory Tracking in Changing Conditions. In: ; 2018: 7831–7837.

31. Pereida K, Kooijman D, Duivenvoorden RR, Schoellig AP. Transfer learning for high-precision trajectory tracking through $\mathcal{L}_1$ adaptive feedback and iterative learning. *International Journal of Adaptive Control and Signal Processing* 2019; 33(2): 388–409.

32. Elnaggar M, Saad MS, Fattah HA, Elshafei AL. Discrete time L_1 adaptive control for systems with time-varying parameters and disturbances. In: ; 2016: 2115–2120.

33. Gunnarsson S, Norrlöf M. On the design of ILC algorithms using optimization. *Automatica* 2001; 37(12): 2011–2016.

34. Goodwin G, Ramadge P, Caines P. Discrete-time multivariable adaptive control. *IEEE Transactions on Automatic Control* 1980; 25(3): 449–456.

35. Goodwin GC, Sin KS. *Adaptive Filtering Prediction and Control*. Prentice Hall . 1984.

36. Akhtar S, Bernstein DS. Lyapunov-stable discrete-time model reference adaptive control. *International Journal of Adaptive Control and Signal Processing* 2005; 19(10): 745–767.

37. Cuzzola FA, Geromel JC, Morari M. An improved approach for constrained robust model predictive control. *Automatica* 2002; 38(7): 1183–1189.

38. Powers C, Mellinger D, Kumar V. Quadrotor Kinematics and Dynamics. In: Springer. 2015 (pp. 307–328).

AUTHOR BIOGRAPHY

empty.pdf

Author Name. This is sample author biography text this is sample author biography text.