Robust Adaptive Model Predictive Control with Worst-Case Cost

Anilkumar Parsi [*] **Andrea Iannelli** [*] **Mingzhou Yin** [*]
Mohammad Khosravi [*] **Roy S. Smith** [*]

[*] *Automatic Control Laboratory, ETH Zurich, Switzerland*
(e-mail: {aparsi,iannelli,myin,khosravm,rsmith}@control.ee.ethz.ch).

Abstract: A robust adaptive model predictive control (MPC) algorithm is presented for linear, time invariant systems with unknown dynamics and subject to bounded measurement noise. The system is characterized by an impulse response model, which is assumed to lie within a bounded set called the feasible system set. Online set-membership identification is used to reduce uncertainty in the impulse response. In the MPC scheme, robust constraints are enforced to ensure constraint satisfaction for all the models in the feasible set. The performance objective is formulated as a worst-case cost with respect to the modeling uncertainties. That is, at each time step an optimization problem is solved in which the control input is optimized for the worst-case plant in the uncertainty set. The performance of the proposed algorithm is compared to an adaptive MPC algorithm from the literature using Monte-Carlo simulations.

Keywords: predictive control, adaptive MPC, impulse response, robust optimization

1. INTRODUCTION

Model predictive control (MPC) is a popular strategy used to solve multivariable control problems due to its ability to handle nonlinearities and constraints while guaranteeing feasibility and stability (Rawlings and Mayne (2009)). The idea of MPC is to compute an input sequence at each time step, such that the input is optimal with respect to given performance index and system constraints. Only the first control input is applied to the plant, and the future control inputs are recomputed at the next time step. Robust MPC algorithms have the ability to handle uncertainty in the system models, ensuring that constraints are not violated despite inaccuracies in modeling the plant (Bemporad and Morari (1999)). However, having large model uncertainties adversely affects the performance of robust controllers. Adaptive control is one technique which can be used to ameliorate the conservatism induced by robustness to modeling inaccuracies. The idea of adaptive control is to carry out controller adjustments in real time based on the collected input-output data. Since MPC is an optimization based technique, performing online adaptation can be integrated easily into the MPC control structure. Utilizing this advantage, a variety of adaptive MPC control algorithms have been developed in the last decade. The main differences in these algorithms are in the model structure (impulse response, state-space, ARMAX, etc) and the adaptation techniques (set-membership identification, recursive least squares, etc) used.

In Kim and Sugie (2008), an adaptive MPC algorithm was developed for single input multiple output, linear time invariant (LTI) systems. The system was described using a state-space model with uncertain parameters which were identified online using a recursive least squares technique. In Lorenzen et al. (2017) and Lu and Cannon (2019), adaptive MPC algorithms were proposed for multi input multi output (MIMO), linear time varying (LTV) systems, which were also described using a parametric state-space structure. The size of uncertainty in the model parameters was updated online using set-membership identification (Milanese and Vicino (1991)). Using results from robust tube MPC (Kouvaritakis and Cannon (2016)), these algorithms guarantee stability and recursive feasibility of the controllers. Instead of an estimate of the model, a worst-case cost was used to describe the performance of the MPC algorithm. This means that in addition to constraint satisfaction, the control performance is robust to the worst-case model uncertainty. However, the algorithms require the knowledge of the state-space structure and noise free measurements of the states, which can be restrictive.

Alternatively, impulse response models have been used to describe the system dynamics. In Tanaskovic et al. (2013), an adaptive MPC algorithm was presented for single input single output (SISO), LTI systems with input and output constraints and measurement noise. The algorithm uses a finite impulse response (FIR) model, and assumes that the true impulse response of the system lies inside a bounded polytope. It was shown that the algorithm can handle large uncertainties, and hence the prior bounds on the FIR coefficients need not be tight. Set-membership identification was used to refine the model set online using measurement data. In Tanaskovic et al. (2019), the algorithm was extended to MIMO, LTV systems, with basis function parameterizations. In the algorithm, the MPC objective was defined using the Chebyshev center of the model set, while the constraints were robustly satisfied

[⋆] This work is supported by the Swiss National Science Foundation under grant number: 200021_178890, and is part of the Swiss Competence Center for Energy Research SCCER FEEB&D of the Swiss Innovation Agency Innosuisse.

for all the models in the set. The control performance of the algorithm was improved in Bujarbaruah et al. (2018) by using a recursive least squares estimator of the system, and defining chance constraints on the outputs.

In this paper, we present a robust adaptive MPC algorithm which uses an FIR model description and a worst-case performance index. For notational simplicity, SISO and LTI systems are considered in this paper but the results can be extended to MIMO and time varying systems. The system is subject to input constraints, output constraints and bounded measurement noise. To define the model uncertainty, the algorithm uses a polytopic feasible system set (FSS) which is updated online using set-membership identification. The constraints are enforced for all models in the FSS, and the objective function is defined using a min-max cost. Using such a cost function optimizes the control performance over all the plants in the FSS. The proposed controller guarantees recursive feasibility, and only requires the solution of linear and quadratic programs at each time step. The performance of the robust adaptive MPC algorithm is compared against the adaptive MPC algorithm proposed in Tanaskovic et al. (2013) to track different reference trajectories. It is shown that the robust adaptive MPC algorithm improves worst-case performance for all the trajectories considered, and the mean performance for some of the trajectories.

Notation: The sets of integers and real numbers are denoted by $\mathbb{Z}$ and $\mathbb{R}$ respectively, and the set of positive integers is denoted by $\mathbb{Z}_{>0}$. For a vector b, $b^\intercal$ represents its transpose, and $[b]_i$ refers to the i^{th} element in it. The i^{th} row and j^{th} column of a matrix A are denoted by $[A]_{i*}$ and $[A]_{*j}$ respectively. The value $x(i|t)$ denotes the value of the variable x at time step i, predicted at the time step t. The absolute value of a scalar a is denoted by $|a|$. The rate of change of a signal $a(t)$ computed as $a(t) - a(t-1)$ is denoted by $\Delta a(t)$. For any real scalar-valued function J, $\max\limits_{h \in \mathbb{H}} J(h)$ refers to the maximum value of J over the set $\mathbb{H}$.

2. BACKGROUND MATERIAL

2.1 System description

We consider a SISO, discrete time, strictly proper, LTI system. The system is assumed to have unknown but stable dynamics. The true system S has an infinite impulse response (IIR) $\{h_S\}_1^\infty$ where the elements $\{h_S(1), h_S(2), \ldots\}$ are the impulse response coefficients of the system. To have a computationally tractable representation of the system for MPC, it is modeled using FIR coefficients. They are represented by the components of the vector $h_m \in \mathbb{R}^m$, where m is the length of the FIR. At any time step $t \in \mathbb{Z}_{>0}$, the system output $y(t)$ can be represented as

$$y(t) = \sum_{i=1}^{\infty} h_S(i) u(t-i), \tag{1}$$

and the model output $y_m(t)$ as

$$y_m(t) = \sum_{i=1}^{m} h_m(i) u(t-i) \doteq h_m * u(t), \tag{2}$$

where $*$ is the convolution operator and $u(t)$ is the input sequence. The measured output of the system is

$$\tilde{y}(t) = y(t) + v(t), \quad \forall t \in \mathbb{Z}_{>0},$$

where $v(t)$ is the measurement noise at the time step t. The following assumptions are made on the system and the noise.

Assumption 1. The noise $v(t)$ is bounded according to

$$|v(t)| < \epsilon, \qquad \forall t \in \mathbb{Z}_{>0}. \tag{3}$$

Assumption 2. The IIR coefficients $\{h_S\}_1^\infty$ satisfy the bounds

$$\begin{aligned} L_l \leq h_S(i) \leq L_u, \qquad & i = 1, 2, \ldots, \mu \\ L_l \rho^{i-\mu} \leq h_S(i) \leq L_u \rho^{i-\mu}, & i = \mu+1, \ldots, \infty, \end{aligned} \tag{4}$$

for parameters $L_l, L_u, \rho \in \mathbb{R}$: $L_l, L_u \geq 0, \rho \in (0,1)$ and $\mu \in \mathbb{Z}_{>0}$.

Remark 1. Assumption 1 is reasonable because the measurement noise in most systems is bounded, and the bound is specified. Assumption 2 is valid for open-loop stable systems (a common assumption in the context of system identification and adaptive control). The parameter ρ defines the rate of decay corresponding to the dominant pole in the system, while the parameters L_l, L_u and μ capture the initial dynamics. However, the assumption also restricts the sign of the IIR coefficients to be positive. This is because the algorithm presented here is suitable for systems which have uncertain parameters with a known sign. This is true for a certain class of systems, for e.g., positive systems (Farina and Rinaldi (2011)), but the presented algorithm is not restricted to this class. It can be extended to LTI systems represented using basis functions (Wahlberg and Mäkilä (1996)), with the restriction that the signs of the basis function coefficients are known.

The system is subject to the input and output constraints given by

$$\begin{aligned} |u(t)| &\leq \bar{u}, \\ |\Delta u(t)| &\leq \overline{\Delta u}, \\ |y(t)| &\leq \bar{y}, \qquad \forall t \in \mathbb{Z}_{>0}. \end{aligned} \tag{5}$$

The goal is to design a controller so that the output follows a known, desired trajectory y_{des} while satisfying the constraints (5).

2.2 Truncation error

The bounds on the FIR model coefficients h_m can be derived from Assumption 2 as

$$\begin{aligned} L_l \leq h_m(i) \leq L_u, \qquad & i = 1, 2, \ldots, \mu \\ L_l \rho^{i-\mu} \leq h_m(i) \leq L_u \rho^{i-\mu}, & i = \mu+1, \ldots, m. \end{aligned} \tag{6}$$

The bounds used in (6) are equal to the bounds on the first m IIR coefficients in (4). However, the bounds in (6) can be relaxed since the proposed controller is adaptive. That is, the exact knowledge of the true system dynamics is not necessary, and an upper bound on the decay of its impulse response is sufficient for this algorithm. Using (6), the model can account for the part of the output due to the first m impulse response coefficients of the true system. However truncating the length of the impulse response results in an error in the model's prediction, which can be bounded as

$$|y(t) - y_m(t)| = \left| \sum_{i=m+1}^{\infty} h_S(i) u(t-i) \right|$$

$$\leq \sum_{i=m+1}^{\infty} |h_S(i)u(t-i)|$$
$$\leq \bar{u} \sum_{i=m+1}^{\infty} |h_S(i)| \quad \leq \bar{u} L \rho^{m-\mu} \frac{\rho}{1-\rho} \doteq \eta_m. \qquad (7)$$

2.3 Online set-membership identification

Set-membership identification is a technique used to identify systems affected by noise with unknown statistical properties (Milanese and Vicino (1991)). Here, an initial FSS is defined as the set of all possible models that are consistent with the initial information

$$H(0) := \{h_m \in \mathbb{R}^m \,|(6)\}, \qquad (8)$$

which is a polytope in $\mathbb{R}^m$. At each time step t, a non-falsified set is used to update the set $H(t)$, which is the FSS at that time step. For systems with FIR descriptions, polytopic non-falsified sets can be constructed from measurement data. For example, using the measurement $\tilde{y}(t)$, a simple non-falsified set can be written as

$$\delta(t) := \left\{ h \in \mathbb{R}^m \left| \begin{array}{l} h * u(t) \leq \tilde{y}(t) + \eta_m + \epsilon, \\ h * u(t) \geq \tilde{y}(t) - \eta_m - \epsilon \end{array} \right. \right\} \qquad (9)$$

where η_m and ϵ are defined according to (3) and (7) respectively and $\delta(t)$ contains the set of all models that could have generated the measurement $\tilde{y}(t)$. As proposed in Chisci et al. (1998), using a block of s measurements $[\tilde{y}(t-s), \ldots, \tilde{y}(t-1)]$ to generate the non-falsified set improves the identification. Let $\phi(k) \in \mathbb{R}^m$ denote the vector of past m inputs at time step k arranged as

$$\phi(k) = [u(k), u(k-1), \ldots, u(k-m+1)]^\intercal,$$

then the convolution $h * u(k)$ can be written as $\phi(k)^\intercal h$. Similar to (9), the following non-falsified set is defined

$$\Delta_s(t) := \left\{ h \in \mathbb{R}^m \left| \begin{array}{r} \phi(k)^\intercal h \leq \tilde{y}(k) + \eta_m + \epsilon, \\ \phi(k)^\intercal h \geq \tilde{y}(k) - \eta_m - \epsilon, \\ \forall k \in [t-1, t-s] \end{array} \right. \right\} \qquad (10)$$
$$= \{h \in \mathbb{R}^m \,| A_\Delta(t) h \leq b_\Delta\}.$$

where the matrices $A_\Delta \in \mathbb{R}^{2s \times m}, b_\Delta \in \mathbb{R}^{2s}$ are used to characterize $\Delta_s(t)$, which is the set of all models that could have generated the measurements $[\tilde{y}(t-s), \ldots, \tilde{y}(t-1)]$. The FSS can be updated at each time step using $\Delta_s(t)$ according to

$$H(t) = H(t-1) \cap \Delta_s(t). \qquad (11)$$

Since the initial FSS is defined as a polytopic set and $\Delta_s(t)$ is polytopic, the set $H(t)$ remains polytopic $\forall t$ if it is updated according to (11). However, the number of hyperplanes in $H(t)$ will increase at every time step. To prevent this, $H(t)$ is defined using a finite number of polytopic constraints given in

$$H(t) := \{h | A_h h \leq b_h(t)\}, \qquad (12)$$

where $A_h \in \mathbb{R}^{p \times m}$ is a matrix chosen offline and $b_h(t) \in \mathbb{R}^p$ is updated online such that $H(t) \supseteq H(t-1) \cap \Delta_s(t)$ is satisfied. This is ensured by calculating $b_h(t)$ as a solution to the following set of p linear programs:

$$\begin{aligned} [b_h(t)]_i \; &= \max_{h \in \mathbb{R}^m} \quad [A_h]_{i*} h \\ \text{s. t.} \quad & \begin{bmatrix} A_h \\ A_\Delta \end{bmatrix} h \leq \begin{bmatrix} b_h(t{-}1) \\ b_\Delta \end{bmatrix}, \; i{=}1, \ldots, p. \end{aligned} \qquad (13)$$

3. ROBUST ADAPTIVE MODEL PREDICTIVE CONTROL

In MPC for systems with FIR models, an optimization problem is solved at each time step to calculate the control inputs for a finite prediction horizon. The control input at the end of the prediction horizon is chosen such that it can remain at a constant value after the horizon while satisfying constraints. This ensures recursive feasibility of the algorithm, whereby the optimization problem at the next time step remains feasible. The control input corresponding to the first time step is applied to the system, and the future control inputs are recalculated at the next time step. In this section, an optimization problem consisting of the worst-case cost with respect to model uncertainties is formulated, while enforcing the system constraints defined in (5). A robust adaptive MPC algorithm is then described.

Let U be the vector of predicted control inputs $[u(t|t), u(t+1|t), \ldots, u(t+N-1|t)]$, where N is the prediction horizon. Let $\{\tilde{u}(t-m+1), \tilde{u}(t-2), \ldots, \tilde{u}(t-1)\}$ be the previous $m-1$ control inputs applied to the system. The vectors $\phi(i|t) \in \mathbb{R}^m$ are constructed using past and predicted future control inputs as

$$\phi(i|t) = [q(i|t) \; \ldots \; q(i-m+1|t)]^\intercal, \qquad (14)$$
$$i = t, t+1, \ldots, t+N+m-2,$$

where $q(k|t)$ is defined as

$$q(k|t) = \begin{cases} \tilde{u}(k) & k \leq t \\ u(k|t) & t < k \leq t+N-1 \\ u(t+N-1|t) & t+N-1 < k. \end{cases} \qquad (15)$$

The vectors $\phi(i|t)$ are a combination of the past control inputs which are known, and the future control inputs which are the decision variables. These vectors will be used to represent the future outputs of the system $y(i|t)$ as $\phi(i|t)^\intercal h$, for $h \in H(t)$ and $i \in [t, t+N+m-2]$. In (15), the control inputs after the prediction horizon are set to be equal to $u(t+N-1|t)$, so that recursive feasibility can be guaranteed.

3.1 Robust objective function

The optimization problem in the MPC algorithm includes a cost minimization objective. To ensure that the performance of the controller is robust to the worst-case plant uncertainty, the cost function J is defined according to

$$J = \max_{h \in H(t)} \sum_{i=t}^{t+N-1} \left(y_{\text{des}}(i|t) - \phi(i|t)^\intercal h\right)^2, \qquad (16)$$

where y_{des} is the desired output trajectory and the vectors $\phi(i|t)$ are defined in (14) and (15). Additional terms which penalize the input u and the rate of change of input Δu can be added to the cost function, but are omitted for notational simplicity. The defined cost function ensures that the optimized control inputs minimize the maximum possible value of J over all the models in $H(t)$. This is a min-max objective, which must be reformulated in a convex manner. Using supplementary cost variables $c \in \mathbb{R}^N$, the cost function in (16) can be rewritten as

$$J = \sum_{j=1}^{N} c(j)^2, \qquad (17)$$

$$\begin{aligned} \text{s.t} \quad & \max_{h\in H(t)} \quad y_{\text{des}}(i|t) - \phi(i|t)^{\intercal} h \le c(i-t+1), \\ & \max_{h\in H(t)} \; -y_{\text{des}}(i|t) + \phi(i|t)^{\intercal} h \le c(i-t+1), \\ & \qquad\qquad\qquad\qquad\qquad\qquad \forall i \in [t, t+N-1]. \end{aligned} \tag{18}$$

In (18), each element of c is specified as an upper bound on the worst case deviation from the reference.

3.2 System constraints and optimization problem

While the constraints (18) are used to implement a min-max objective, the system's input and output constraints given in (5) must be enforced through the optimization problem. The constraints on the input at each time step can be written as

$$\begin{aligned} -\bar{u} \le u(i|t) \; &\le \bar{u}, \\ -\overline{\Delta u} \le \Delta u(i|t) \; &\le \overline{\Delta u}, \quad \forall i \in [t, t+N-1]. \end{aligned} \tag{19}$$

The output constraints must be satisfied for all the models in $H(t)$, i.e., robustly satisfied. The system outputs can be bounded according to

$$\begin{aligned} y(i|t) &\ge \min_{h\in H(t)} \phi(i|t)^{\intercal} h - \eta_m, \\ y(i|t) &\le \max_{h\in H(t)} \phi(i|t)^{\intercal} h + \eta_m, \; \forall i \in [t{+}1, t{+}N{+}m{-}2], \end{aligned}$$

where η_m is the truncation error defined in (7). Hence, the output constraints in (5) can be formulated as

$$\begin{aligned} \max_{h\in H(t)} \quad \phi(i|t)^{\intercal} h \; &\le \bar{y} - \eta_m, \\ \max_{h\in H(t)} \; -\phi(i|t)^{\intercal} h \; &\le \bar{y} - \eta_m, \;\; \forall i \in [t{+}1, t{+}N{+}m{-}2], \end{aligned} \tag{20}$$

where the outputs constraints for $i \in [t+N, t+N+m-2]$ are enforced to ensure no constraint violations occur when a constant input is applied at the end of the prediction horizon. Thus, the optimization problem in the MPC controller can be set up as

$$\begin{aligned} &\min_{U} \; J \\ &\text{subject to } (18), (19), (20). \end{aligned} \tag{21}$$

The constraints (18) and (20) can be replaced by an equivalent set of linear constraints using techniques from robust optimization (Ben-Tal et al. (2009)). Using this reformulation, (21) is simplified to a standard quadratic program.

3.3 Robust adaptive MPC algorithm

The procedure for robust adaptive MPC is described in Algorithm 1. The algorithm must be initialized with the set $H(0)$ and the past inputs applied to the system. At each time step t, the measurements $\tilde{y}(t)$ are used to update the non-falsified set $\Delta_s(t)$ and $H(t)$ according to (10) and (13) respectively. Algorithm 1 guarantees robust satisfaction of input and output constraints, as shown by the following theorem from Tanaskovic et al. (2013).

Algorithm 1 Robust adaptive MPC

```
Initialize $\Delta_s(0)$ using past measurements and inputs according to (10).
Initialize H(0) according to (8),(12).
$t \leftarrow 1$
repeat
    Obtain the measurement $\tilde{y}(t)$
    Update $\Delta_s(t)$ according to (10).
    Update $H(t)$ using (13).
    Solve optimization problem (21) to compute $U$.
    Apply the control input $u(t|t) \leftarrow U(1)$.
    $t \leftarrow t + 1$
until
```

Theorem 1. If Assumptions 1-2 hold, and the optimization problem (21) is feasible at t=0. Then, the closed loop system obtained by applying Algorithm 1 is guaranteed to satisfy the input and output constraints $\forall t > 0$.

In Algorithm 1, at each time step the FSS is truncated using measurement data. The performance of the closed loop system improves with time because the size of the uncertainty set decreases and hence the controller is adaptive. The advantage of including a worst-case cost in the objective function of the MPC controller is that it increases the robustness of the performance to model uncertainty. However, this affects its adaptivity, i.e., the amount of information it learns from a system. This is because the control inputs of robust adaptive controllers are generally smaller than that of a controller using nominal cost function. To investigate the effects of adaptivity and robustness on the overall performance on the system, a simulation study was performed.

4. SIMULATIONS

In this section, the performance of the robust adaptive MPC (RAMPC) algorithm proposed in Algorithm 1 is compared against the adaptive MPC (AMPC) algorithm given in Tanaskovic et al. (2013). The AMPC algorithm defines the objective function of the MPC controller using the Chebyshev center of the FSS as the estimate of the true system. The objective function is defined therein as a quadratic function of the predicted deviation from the reference trajectory, along with penalties on the input and the rate of change of input. To make the comparison equivalent, the penalties on input and input rate have been removed. Thus in both the AMPC and RAMPC algorithms, the only objective of the optimal control problem is reference tracking. The performance of the algorithms is compared based on the root mean squared (RMS) deviation from the reference trajectory.

4.1 Parameter Settings

The true systems which must be controlled by the AMPC and RAMPC algorithms were randomly sampled from a predefined and bounded set. The parameters defining the initial FSS are given in Table 1, where the bounds on the impulse response coefficients can be calculated from (4).

Table 1. Parameters describing bounds on impulse response, input and output

L_l	L_u	μ	ρ	ϵ	$\bar{u}$	$\overline{\Delta u}$	$\bar{y}$
0.3	1	4	0.65	0.1	2	0.8	4

A set of 200 impulse responses was obtained by random sampling of each coefficient from a uniform distribution between the bounds defined in (4). Each impulse response obtained was considered as a true system which must be controlled using the AMPC and RAMPC algorithms. The prediction horizon was set to $N = 15$ time steps, and the length of the FIR in the model was truncated to $m = 12$ parameters. This results in a truncation error $\eta_m = 0.12$, as given in (7). To define the initial FSS according to (12), a matrix A_h was constructed with $p = 156$ constraints. The initial bounds $b_h(0)$ were calculated using the parameters specified in Table 1. The value of $b_h(t)$ was updated at each time step using a block of $s = 36$ past measurements.

In both the algorithms, the control inputs form regressor vectors which are used to refine the FSS as shown in (10) and (13). The reference trajectory to be tracked influences the sequence of control inputs, and hence the refinement of the uncertainty sets. However, only a step reference was considered in Tanaskovic et al. (2013). To investigate dependence of controller performance on the reference trajectory, AMPC and RAMPC algorithms were used to track the different trajectories in Figure 1. The step trajectory has a quick change in reference which is difficult to track. A smaller peak to peak amplitude was chosen for the step so that the overall deviations from different trajectories have comparable magnitudes.

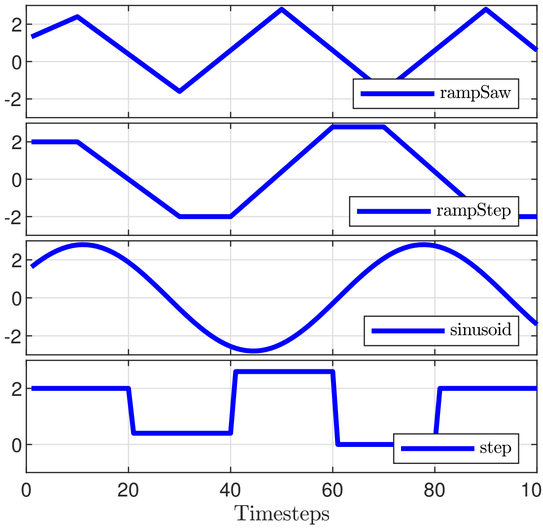

Fig. 1. Reference trajectories used to compare the performance of nominal and robust adaptive MPC.

The optimization problems in both the algorithms were solved using CVX (Grant and Boyd (2014)) and MOSEK (Andersen and Andersen (2000)).

4.2 Results and discussions

The performance of the algorithms was evaluated based on the RMS deviation from the given reference trajectory. Figure 2 shows the distribution of the RMS deviations from each of the trajectories considered. The histograms in blue and red represent the performance of the AMPC and RAMPC algorithms respectively. Since the FSS is in a high dimensional space ($m = 12$), its center might not always be a good representation of the true system. For all trajectories except the step, using a robust controller improves the control performance even when the worst-case uncertainty is not realized. It can be seen that the distribution of RMS deviation for the RAMPC algorithm has a smaller tail on the right side. This corresponds to better performance using the RAMPC algorithm when the true system is not close to the Chebyshev center of the FSS. In addition, the performance of the algorithms is dependent on the reference trajectory, as seen in the plot corresponding to the step reference. Due to the quick changes in the trajectory, the minimum value of RMS deviation measured for the step reference was higher.

The mean and maximum of the deviations from reference trajectories are given in Table 2 and the results with better performance are highlighted. The maximum of the RMS deviation corresponds to the worst-case performance, which is improved for all the trajectories while using the RAMPC algorithm. Comparing the means of the RMS deviation from reference, it can be seen that RAMPC performs better for all the trajectories except the step. This is due to the robust cost in the RAMPC algorithm, which results in conservative responses while tracking the quick changes in the step reference.

Table 2. Mean and maximum values of the root mean squared deviations from the reference.

	Mean RMS deviation		Maximum RMS deviation	
Trajectory	AMPC	RAMPC	AMPC	RAMPC
rampSaw	0.152	**0.137**	0.303	**0.228**
rampStep	0.208	**0.196**	0.352	**0.292**
sinusoid	0.182	**0.167**	0.380	**0.311**
step	**0.251**	0.269	0.354	**0.350**

In Figure 3, the trajectories with the highest RMS deviation from the step reference are plotted. It can be seen that the tracking performance improves with time due to the reduction in the model uncertainty. The conservatism of the RAMPC algorithm while tracking the quick changes is visible in the trajectories. However, the flat portion of the trajectory is tracked better with the RAMPC algorithm, and it reduces the worst-case cost of the AMPC algorithm. The computational complexity of the optimal control problem is higher in the RAMPC algorithm due to the additional robust constraints defined in (18). However, the AMPC algorithm solves an additional linear program to estimate the Chebyshev center. Table 3 compares the runtimes of the algorithms at each time step, when the number of FIR coefficients was incrementally increased.

Table 3. Runtimes for different FIR lengths

m	t_{AMPC}[in s]	t_{RAMPC}[in s]
8	2.10	1.80
10	2.37	2.32
12	2.44	2.52
14	2.92	3.10
20	7.82	8.80

The values t_{AMPC} and t_{RAMPC} correspond to the time required to calculate new control inputs at each time step for the AMPC and RAMPC algorithms respectively. The simulations were performed on an Intel i7-8550U 1.8 GHz processor, by keeping the prediction horizon constant. It can be seen that for small impulse responses, the RAMPC algorithm is faster. The runtime of the RAMPC algorithm is higher for larger systems, but is only 12% higher than the corresponding runtime for the AMPC algorithm.

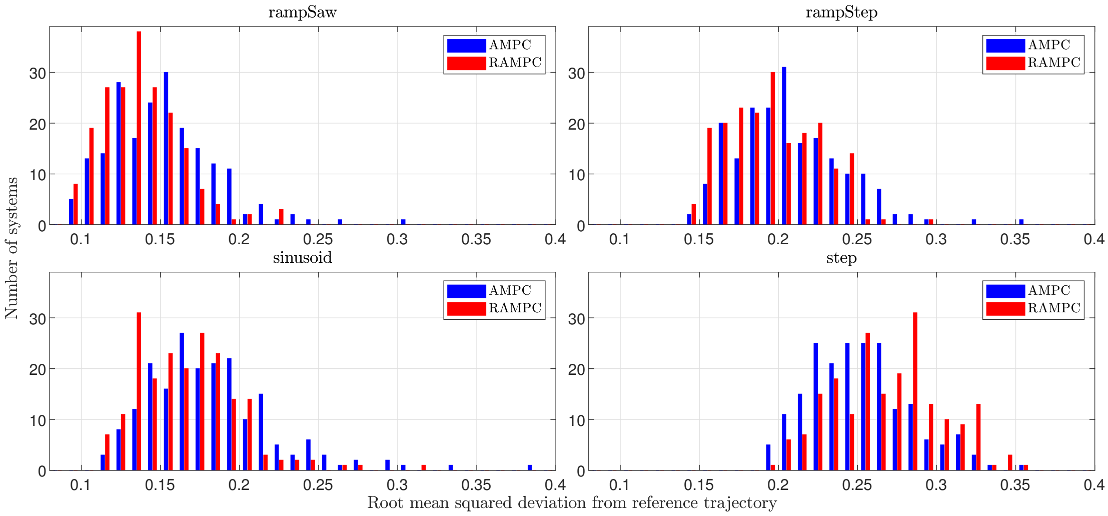

Fig. 2. Distribution of RMS deviation of output from each reference trajectory using AMPC and RAMPC algorithms.

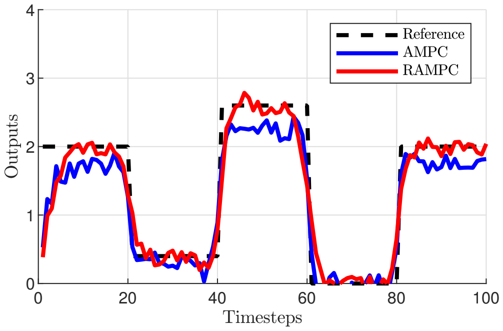

Fig. 3. Worst-case tracking performance of the AMPC and RAMPC algorithms using a step reference.

5. CONCLUSION

In this paper, a robust adaptive MPC algorithm was described which can be applied for LTI systems with input and output constraints. The algorithm uses set online membership identification to reduce the uncertainty in the model parameters, and ensures robust constraint satisfaction. The objective function of the MPC algorithm is designed so that it is robust to model uncertainty, and the resulting optimization problem was posed as a standard quadratic program. The algorithm was compared to an existing algorithm in literature using Monte Carlo simulations. The simulation study showed that using a worst-case objective function improves the performance of the adaptive MPC controller for various reference trajectories.

REFERENCES

Andersen, E.D. and Andersen, K.D. (2000). The MOSEK interior point optimizer for linear programming: an implementation of the homogeneous algorithm. In *High Performance Optimization*, 197–232. Springer.

Bemporad, A. and Morari, M. (1999). Robust model predictive control: A survey. In *Robustness in Identification and Control*, 207–226. Springer.

Ben-Tal, A., El Ghaoui, L., and Nemirovski, A. (2009). *Robust Optimization*, volume 28. Princeton University Press.

Bujarbaruah, M., Zhang, X., and Borrelli, F. (2018). Adaptive MPC with chance constraints for FIR systems. In *2018 Annual American Control Conference (ACC)*, 2312–2317.

Chisci, L., Garulli, A., Vicino, A., and Zappa, G. (1998). Block recursive parallelotopic bounding in set membership identification. *Automatica*, 34(1), 15–22.

Farina, L. and Rinaldi, S. (2011). *Positive Linear Systems: Theory and Applications*, volume 50. John Wiley & Sons.

Grant, M. and Boyd, S. (2014). CVX: Matlab software for disciplined convex programming, version 2.1. `http://cvxr.com/cvx`.

Kim, T.H. and Sugie, T. (2008). Adaptive receding horizon predictive control for constrained discrete-time linear systems with parameter uncertainties. *International Journal of Control*, 81(1), 62–73.

Kouvaritakis, B. and Cannon, M. (2016). *Model Predictive Control: Classical, Robust and Stochastic*. Springer, London.

Lorenzen, M., Allgöwer, F., and Cannon, M. (2017). Adaptive model predictive control with robust constraint satisfaction. *IFAC-PapersOnLine*, 50(1), 3313–3318.

Lu, X. and Cannon, M. (2019). Robust adaptive tube model predictive control. In *2019 American Control Conference (ACC)*, 3695–3701.

Milanese, M. and Vicino, A. (1991). Optimal estimation theory for dynamic systems with set membership uncertainty: an overview. *Automatica*, 27(6), 997–1009.

Rawlings, J.B. and Mayne, D.Q. (2009). *Model predictive control: Theory and design*. Nob Hill Pub.

Tanaskovic, M., Fagiano, L., and Gligorovski, V. (2019). Adaptive model predictive control for linear time varying MIMO systems. *Automatica*, 105, 237–245.

Tanaskovic, M., Fagiano, L., Smith, R., Goulart, P., and Morari, M. (2013). Adaptive model predictive control for constrained linear systems. In *2013 European Control Conference (ECC)*, 382–387.

Wahlberg, B. and Mäkilä, P. (1996). On approximation of stable linear dynamical systems using Laguerre and Kautz functions. *Automatica*, 32(5), 693–708.