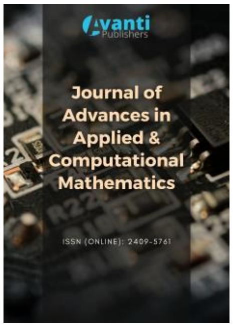
Publishers

Published by Avanti Publishers

Journal of Advances in Applied & Computational Mathematics

ISSN (online): 2409-5761



Avanti Publishers

Journal of Advances in Applied & Computational Mathematics

ISSN (ONLINE): 2409-5761

An Adversarial Approach to Adaptive Model Predictive Control#

Pawel Wachel[1,*] and Cristian R. Rojas[2]

[1] *Department of Control Systems and Mechatronics, Wroclaw University of Science and Technology, Wroclaw, Poland.*
[2] *School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, 100 44 Stockholm, Sweden.*

ARTICLE INFO

Article Type: Research Article

Keywords:
Linear system
Adaptive control
Multi-armed bandits
Model predictive control
State-space representation

Timeline:
Received: July 18, 2022
Accepted: August 10, 2022
Published: September 19, 2022

Citation: Wachel P, Rojas CR. An Adversarial Approach to Adaptive Model Predictive Control. J Adv App Comput Math. 2022; 9: 135-146.

DOI: https://doi.org/10.15377/2409-5761.2022.09.10

ABSTRACT

This paper presents a novel approach to introducing adaptation in Model Predictive Control (MPC). Assuming limited a priori knowledge about the process, we consider a finite set of possible models (a dictionary), and use the theory of adversarial multi-armed bandits to develop an adaptive version of MPC called adversarial adaptive MPC (AAMPC). Under weak assumptions on the dictionary components, we then establish theoretical bounds on the performance of AAMPC and show its empirical behaviour via simulation examples.

#This work has been supported by the Swedish Research Council under contract number 2016-06079 (NewLEADS) and by the Digital Futures project EXTREMUM.

*Corresponding Author
Emails: pawel.wachel@pwr.edu.pl
Tel: (+48) 71 320 3357

1. Introduction

Model Predictive Control (MPC) is one of the most popular control strategies used in the process industry, thanks to its ability to handle constraints explicitly [1-3].

Due to its structure, MPC requires an explicit model of the process to be controlled, and obtaining such a model is often a very costly operation (both in terms of time and money). Indeed, it has been estimated that modelling can account for up to 75% of the total control commissioning cost [4]. Among the reasons for such a high cost lie the complexity of the processes being controlled and the fact that, in general, the process has to be estimated in a closed loop (for example, under a poorly tuned MPC controller), which has many caveats [5].

Several approaches have been proposed to reduce the pitfalls of closed-loop identification and make it possible to estimate the process from operating data. For example, the data collected during the operation of a standard MPC controller may suffer from a lack of persistence of excitation [5], so if one is collecting data from a de-tuned MPC controller to re-commission it, several authors have suggested modifying the MPC strategy in order to enforce sufficient excitation [6-10]. Also, considerable effort has been put into experiment design for the (re-)tuning of MPC controllers [11-14], and into the design of MPC strategies with simultaneous model identification [15-17]. This has also led to the development of adaptive variants of MPC [18], where the control input aims to balance *exploration* and *exploitation*. In the last ten years, there have been many publications in adaptive MPC; please refer to [19,20] and the references therein for recent overviews of the literature.

Researchers have developed data-driven MPC formulations as an alternative approach to reducing the cost of designing MPC controllers. Two such methodologies are “subspace model predictive control” [21,22], which combines tools from subspace system identification [23,24] and MPC, avoiding the intermediate step of building an explicit model of the process, and DeePC [25], where MPC is formulated directly in terms of a single batch of input-output data, by using *Willems’ Fundamental Lemma* [26].

In this paper, we propose a new adaptive version of model predictive control, based on recent developments within computational learning theory, and in particular adversarial multi-armed bandits (MABs) [27,28]. MABs, in their most basic formulation, can be seen as a simple form of adaptive control, where an agent interacts with an unknown environment in order to minimize a given cost, and to do so, it has to balance the exploration of the environment and its exploitation; in control jargon, the input thus plays a *dual* role [29]. In the adversarial setup, the environment can behave arbitrarily (as long as the cost remains bounded), and the goal is to design an agent whose performance is not much worse than that of the best in a finite pool of agents.

Adversarial MABs and the related framework of prediction with expert advice have been successfully applied to tasks such as iterative estimation of the H_∞-norm of linear systems [30], and to the aggregative modeling of nonlinear systems [31, 32]. However, applications of Adversarial MABs within adaptive control have been scarce, and to the best of our knowledge, [33] is one of the earliest papers where these tools are used to the simple noiseless LQR problem. Other related techniques based on online convex programming have been discussed, *e.g.* , in [34,35].

Based on the above, our main contributions are:

- A novel adversarial adaptive MPC algorithm (AAMPC) is derived by applying results from adversarial MABs (in particular, the EXP3 algorithm) to a standard MPC formulation. The algorithm does not make strong assumptions about the process but only considers a finite set, or dictionary, of process models; the models in the dictionary can be almost arbitrary. To the best of our knowledge, this is one of the first papers that apply adversarial MABs to a standard control setup such as MPC.
- Theoretical bounds on the performance of AAMPC are established under mild requirements.
- The empirical behaviour of the algorithm is verified through numerical simulations.

Note that the setup considered in the paper is fairly simple, focusing on linear state-feedback MPC, even though the approach is applicable under more general assumptions and controller types. However, our goal is to introduce the use of adversarial bandit tools in control, particularly the EXP3 algorithm, and for that purpose, the linear state-feedback MPC case is an ideal scenario because it is a well understood and widely used in industry.

The paper is structured as follows. In Section 2, the problem is formulated. In Section 3, the notion of deterministic multi-armed bandits is introduced. Section 4 presents the main contribution of the paper, namely, an adaptive MPC strategy (AAMPC) based on the EXP3 algorithm for deterministic multi-armed bandits, as well as some theoretical results to support the performance of this new method. Section 5 presents some simulation results that illustrate the ability of AAMPC to achieve performance similar to the MPC controller based on the best model in the dictionary. Finally, Section 6 concludes the paper.

2. Problem Formulation

Consider a scalar discrete-time linear time-invariant process described by the equation

$$x_{k+1} = Ax_k + Bu_k + e_k, \qquad k \in \mathbb{N}_0, \tag{1}$$

where $x_k \in \mathbb{R}^n$ is the state, assumed to be directly measurable; $u_k \in \mathbb{R}$ is the input, (A, B) is a pair of stabilizable but unknown matrices of compatible dimensions and $e_k \in \mathbb{R}^n$ is a white noise sequence of zero mean and unknown covariance matrix. If the state vector, x_k, were not directly measurable, the control mechanism proposed in the paper can be combined with a suitably tuned Kalman filter for each model; Due to the construction of our method, the algorithm can be applied without major changes to the basic setup.

In MPC, assuming that A and B are known, the input at time k, u_k, is determined as the solution $\tilde{u}_k$ of the optimization problem

$$\mathrm{MPC}(A, B, N_p, \lambda, U, x_k): \tag{2}$$

$$\begin{aligned} \min_{\tilde{u}_k, \dots, \tilde{u}_{k+N_p-1}; \tilde{x}_{k+1}, \dots, \tilde{x}_{k+N_p}} \quad & \sum_{i=k}^{k+N_p-1} (\tilde{x}_{i+1}^T \tilde{x}_{i+1} + \lambda \tilde{u}_i^2) \\ \text{s.t.} \quad & \tilde{x}_{i+1} = A\tilde{x}_i + B\tilde{u}_i, \quad i = k, \dots, k+N_p \\ & |\tilde{u}_i| \leqslant U, \quad i = k, \dots, k+N_p-1 \\ & \tilde{x}_k = x_k. \end{aligned}$$

Here, the tilded variables represenłt predicted versions of the true (un-tilded) signals, $U > 0$ is an upper bound on the input signal, $\lambda \geqslant 0$ is a known tuning parameter, N_p is the prediction horizon, and x_k is the true state at time k.

2.1. Dictinary of Models

Since, in practice, matrices A and B are not known exactly, it is necessary to estimate them. One way to do this is adaptive, *i.e.* , while the process is being controlled. One can construct different candidate models for A and B, *e.g.* , by applying different estimation techniques leading to different competitive estimators of those matrices, by using a single estimation method based on different tuning parameter values, or simply by considering a set of fixed matrices that would play the role of models.

In this paper, we assume that a set of $\mathfrak{D} < \infty$ estimators or fixed (time-independent) models is available for each pair of matrices A and B (we call this set a *dictionary*). For simplicity, we assume the estimators to be fixed models and denote them as A^d and B^d, respectively, where $d \in \{1, \dots, \mathfrak{D}\}$ is the estimator index.

Clearly, based on the limited *a priori* knowledge about the true process (1), one cannot directly apply MPC as in (2), and also having access to a (possibly large) dictionary of models $\{(A^d, B^d): d = 1,2, \dots, \mathcal{D}\}$ does not provide explicit information on which model should be preferred. Therefore, our goal is to construct an adaptive control algorithm that simultaneously explores the dictionary components and exploits the collected data up to the current time for the selection of the model used in (2).

3. Deterministic Multi-Armed Bandits

Multi-armed bandits [27, 28] constitute a general framework for reinforcement learning [36], where an agent has to chose between different possible decisions (or *arms*) in each iteration in order to minimize some cumulative loss and where the outcome of each possible decision is not completely known. The agent, therefore, is subject to an *exploration/exploitation* tradeoff: it should play each arm enough times in order to determine how the losses depend on each arm, but it should also aim to play the best arm (*i.e.* , the one giving the smallest loss) most of the time.

There are three basic types of multi-armed bandits, depending on how the loss depends on the arm being taken: stochastic bandits, deterministic (or adversarial) bandits, and Markovian bandits. In stochastic bandits, one assumes that the losses are random variables (independent between iterations) whose distribution depends on the arm chosen. In deterministic bandits, the losses are assumed to be arbitrary but bounded, and they may depend on the current and past arms chosen by the agent as well as on the strategy it uses to make those choices (but not on any randomization mechanism used by the agent). Finally, in Markovian bandits, the loss associated with each arm is a Markov process that evolves when that arm is played (in that case its new state is revealed to the agent).

In this paper, we will focus on the deterministic bandit framework since it allows for great flexibility in the behaviour of the losses (as long as they remain bounded). This framework is closely related to the so-called "prediction with expert advice" setup [37], where an agent should predict the next value of a quantity which may vary arbitrarily inside a bounded set, and to this end the agent may consult a set of *experts*, whose predictions are available to the agent, as well as all their past prediction successes (or failures); the goal of the agent is then to make a prediction almost as accurate as one derived by the best expert. To achieve this goal, a standard strategy is the *weight majority algorithm* [37,38], which consists in weighting the prediction of each expert according to their past successes, and then considering a weighted vote of their current predictions, or the prediction of an expert picked randomly according to their weights (interpreted as probabilities).

In contrast to prediction with expert advice, for deterministic bandits, the losses of the arms/experts which were not picked are unknown to the agent, while in the former framework, all losses are available to it. A variant of the weighted majority algorithm has been developed to address this difference, where unbiased estimates replace the unknown losses. The resulting algorithm, known as EXP3 [28,39], will be used in the next section to incorporate adaptation into an MPC controller.

4. Adversarial Adaptive MPC

We will now describe the construction of the proposed AAMPC algorithm with a particular focus on the model selection routine. In general, the dictionary elements will yield MPC controllers of varying performance when applied to the true system. Hence, we begin with the assessment of the models selected by AAMPC. Let u_k^d denote a control value calculated as the solution of $MPC(A^d, B^d, N_p, \lambda, U, x_k)$ at time step k, and let $T \in \mathbb{N}$ be a user-defined constant determining the model switching moments. If $k = 0,1,2, \dots$ denotes the time index, an *active* MPC model is allowed to be replaced by another model only if $k \in \{T, 2T, \dots\}$. Since the selected model (defined by a $d \in \{1,2, \dots, D\}$) is fixed within a single time interval of length T, we can define the time-windowed quality function

$$l_n(d;T) := \frac{1}{T} \sum_{k=(n-1)T}^{nT-1} [x_{k+1}^T x_{k+1} + \lambda (u_k^d)^2], \tag{3}$$

where λ is the same tuning parameter used in (2). To ensure good statistical properties of the proposed algorithm, we will assume that l_n is upper bounded by a known constant $C_l < \infty$, as detailed in Theorem 1.

Remark 1. *The above requirement has a natural interpretation, namely, that one should avoid working with poor-quality models leading to unacceptably high values of l_n. Note that for any process (1) with initial state $\|x_0\| \leqslant C_x \leqslant \infty$, bounded noise e_k, and input u_k^d being a solution of $MPC(A^d, B^d, N_p, \lambda, U, x_k)$, a constant $C_l < \infty$ always exists but may remain unknown.*

A natural averaged counterpart of $l_n(d;T)$ is

$$\begin{aligned} L_N([d];T) &:= \frac{1}{N}\sum_{n=1}^{N} l_n(d_n;T) \\ &= \frac{1}{NT}\sum_{k=0}^{NT-1} [x_{k+1}^T x_{k+1} + \lambda(u_k^{d_n})^2], \end{aligned} \tag{4}$$

where $[d]$ stands for the sequence of models used by our method, $n \in \{1,2,\dots,N\}$ is the index of the consecutive model switchings, and N is the total number of model switchings (to be defined by the user). Notice that k represents the actual time instant and is related to n via $n = \lceil k/T \rceil$, and $d_n = d_{\lceil k/T \rceil}$, where $\lceil x \rceil$ is a ceiling function (*i.e.* , the smallest integer not less than x).

The proposed adversarial adaptive MPC method (AAMPC) is described in Algorithm 1. There, $\delta_n \in \{1,2,\dots,\mathfrak{D}\}$ denotes the model selected by AAMPC at the switching step n. In the proposed approach, the decision δ_n is *sampled* from a discrete probability distribution P_n, initially uniform (for $n = 1$) but adjusted after consecutive model selections. In particular, P_n is determined by the past selections and losses (3), *i.e.*, $P_n(d) = P\{\delta_n = d | d_1, l_1, \dots, d_{n-1}, l_{n-1}\}$, for all $d \in \{1,2,\dots,\mathfrak{D}\}$. To simplify the notation, in Algorithm 1 and the sequel, we omit the index n in δ_n when it is clear from the context.

Remark 2. *In Algorithm 1, we use a double indexing scheme according to which k denotes the usual time index (cf. true process (1)) whereas $n = 1,2,\dots,N$ refers to the number of consecutive model switchings, as previously defined.*

Remark 3. *In general, the algorithm focuses on the overall control performance rather than on the direct selection of the most accurate model of the true process. This is achieved by a sequential stochastic sampling of dictionary components. Exploration-exploitation nature of the method prefers models with relatively low loss $l_n(d;T)$ but also explores other, less-accurate, or yet unverified models of the true process* (1).

Some theoretical properties of the proposed control technique are discussed in the next section.

4.1. Theoretical properties of AAMPC

Given a dictionary $\{(A^d, B^d): d = 1,2,\dots,\mathfrak{D}\}$, one can ask about the model d^* leading to the best control performance in a given sense. Based on the definition of cumulative loss in (4), such a model should be a minimizer of L_N, *i.e.* , $d^* \in argmin_d L_N(d;T)$; note, however that, due the presence of disturbances, even if the true process belongs to the dictionary, *i.e.* , $(A,B) \in \{(A^d, B^d): d = 1,2,\dots,\mathfrak{D}\}$, it may not be the case that $(A,B) = (A^{d^*}, B^{d^*})$. In our setup, d^* is unknown, but it can be used as a reference in the formal analysis of AAMPC. In this direction, Theorem 1 below compares the performance of the AAMPC control policy to that of an MPC controller based on model d^*.

Theorem 1. *For the true process in (1) and an arbitrary dictionary of models $\{(A^d, B^d): d = 1,2,\dots,\mathfrak{D}\}$, the performance of AAMPC (Algorithm 1) with model-switching horizon $T < N$ satisfies*

Algorithm 1 Adversarial Adaptive MPC

```
input: $\{(A^d, B^d): d = 1, \dots, \mathcal{D}\}, x_0, \lambda, N_p, N, U, T, C_l$
1:  set: $k = 0, \eta = \sqrt{ln(\mathcal{D})/(N\mathcal{D})}, S_0(d) = 0$ for all $d$
2:  for $n = 1,2, \dots, N$ do
3:      calculate distribution $P_n$:
            $P_n(d) := \frac{exp(\eta S_{n-1}(d))}{\sum_{j=1}^{\mathcal{D}} exp(\eta S_{n-1}(j))}, \quad 1 \leqslant d \leqslant \mathcal{D}$        (5)
4:      sample model $\delta \sim P_n$
5:      for $k = (n-1)T, \dots, (nT-1)$
6:          calculate $u_k^\delta$ as the solution $\tilde{u}_k$ of
                $MPC(A^\delta, B^\delta, N_p, \lambda, U, x_k)$        (6)
7:          apply $u_k^\delta$ in (1) and observe $x_{k+1}$
8:      end for
9:      calculate
            $l_n(\delta; T) = \frac{1}{T} \sum_{k=(n-1)T}^{nT-1} [x_{k+1}^T x_{k+1} + \lambda (u_k^\delta)^2]$
10:     update $S_n(d)$ for all $d$:
            $S_n(d) := S_{n-1}(d) + 1 - \frac{\mathbb{I}\{d = \delta\} l_n(\delta; T)}{C_l P_n(d)}$
11: end for
```

$$E\{L_N(\delta; T)\} - \min_d L_N(d; T) \leqslant 2C_l \sqrt{\mathcal{D}\, ln\, \mathcal{D}/N}, \tag{7}$$

if $l_n(d; T) \leqslant C_l$ for every $n = 1,2, \dots, N$ and $d = 1,2, \dots, \mathcal{D}$, where $C_l < \infty$ is a known constant. The expectation in (7) is taken with respect to the distribution of the AAMPC model selections.

See Section 7 for the proof of Theorem 1.

According to the inequality in Theorem 1, the control policy of AAMPC yields a performance that is close in the mean sense to that of the MPC controller based on the best model d^*. Indeed, for a fixed number of dictionary components, the average *regret* (*i.e.* , the difference between the averaged performance of δ and d^*) decreases with rate $O(N^{-1/2})$. A more detailed analysis of the algorithm's construction indicates, however, that the performance of AAMPC cannot be improved by tuning N 'on-line' since N needs to be *a priori* known for the proper initialization of the constant η (*cf.* line 1 of Algorithm 1). To some extent, this difficulty can be overcome by applying the so-called 'doubling-trick' – a technique often used in MAB approaches, but we will not pursue it here (for details, see *e.g.* , [28,37]).

Continuing the discussion on Theorem 1, it should be emphasized that the upper-bound in (7) holds if the time-windowed loss l_n is bounded by some known constant C_l. In practice, assuming poor *a priori* knowledge about the true process, one can select a large value for C_l, and check if $l_n \leqslant C_l$ during the whole duration of control. If the condition is met for all $n = 1,2, \dots, N$, then (7) holds. However, since a large C_l weakens the upper bound in the theorem, its careful selection is of crucial importance.

Finally, regarding the influence of the model selection horizon T on the control performance, it can be seen that it affects the variance of the time-windowed loss l_n with respect to the noise distribution, which is of order $Var\{l_n(\delta; T)\} = O(T^{-1})$. However, since the optimal selection of T depends on $\mathcal{D}$ and naturally influences the entire control duration (NT time steps), further theoretical studies of its tuning should be performed.

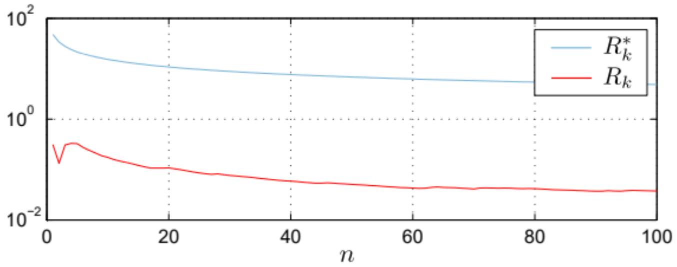


Figure 1: Empirical regret of AAMPC policy ($R_k := \tilde{L}_k(\delta;T) - \tilde{L}_k(d;T)$; red) vs. theoretical upper-bound (7) (denoted as R_k^*; blue).

5. Numerical Simulations

In this section, we present simulation results performed in MATLAB with the YALMIP toolbox [40], which illustrate the application of the AAMPC algorithm to stabilize an unstable process with the following matrices:

$$A = \begin{bmatrix} 1.24 & -0.15 & -0.51 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}. \tag{8}$$

This process has a real pole at $p_1 = -0.5$ and two complex poles of magnitude $|p_2| = |p_3| = 1.01$ and argument $\sphericalangle p_2, p_3 \approx \pm\pi/6$, and initial state $x_0 = [1,1,1]^T$. The process noise e_k is an *i.i.d.* uniformly distributed sequence $U[-0.05,0.05]$, and the state x_k is observed with an *i.i.d.* additive noise $U[-0.05,0.05]$. The dictionary of models (with $\mathcal{D} = 10$ elements) is composed of matrices as in (8) but having all entries perturbed with Gaussian white noise $\mathcal{N}(\mu, \sigma_{\mathcal{D}}^2)$, with $\mu = 0.1, \sigma_{\mathcal{D}}^2 = 0.2$. Thus, the dictionary resembles a set of perturbed estimates of A and B and does not contain an exact model of the true system. For the MPC step in (6) we assume $N_p = 20$, $\lambda = 1$ and $U = 1$ whereas the switching horizon T is equal to 5. The total number of model switchings N equals 100, and $C_l = 5$

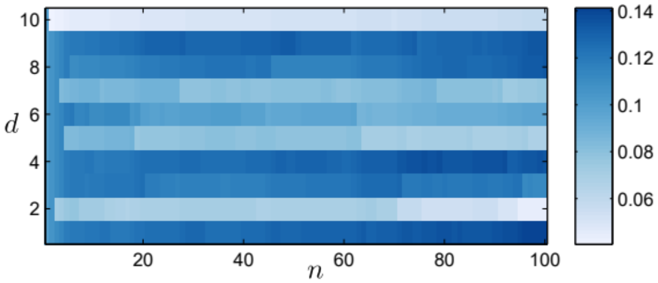


Figure 2: Evolution of the sampling distribution P_n, as a function of the index of model switchings $n = 1,2,\ldots,N$.

In Figure **2** an example of the evolution of the distribution P_n is presented. Clearly, the lack of knowledge at the beginning of the control process (for $n = 1$) is represented by the uniform distribution P_1. However, along with the consecutive switchings (*cf.* Fig. **3**), AAMPC tends to exploit models with relatively low losses l_n, and thus concentrates the measure P_n on the corresponding dictionary elements.

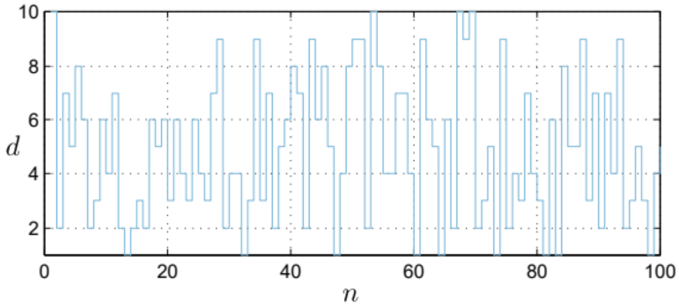


Figure 3: Sequence of models δ selected by AAMPC algorithm vs. index of model switchings $n = 1,2,\ldots,N$.

Figure **1** presents the evolution of the regret of the AAMPC policy, defined as $R_k := \tilde{L}_k(\delta;T) - \tilde{L}_k(d;T)$, and compares it with the upper bound (7) given by Theorem 1 (which is a bound on the *expectation* of R_k). As the figure confirms, AAMPC yields a decreasing regret R_k.

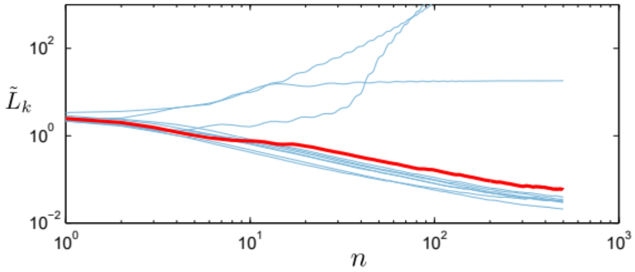


Figure 4: Averaged losses $\tilde{L}_k(d;T)$ for models $(d = 1,2,\dots,D)$ fixed in a whole duration of experiment (blue), and $\tilde{L}_k(\delta;T)$ – the control performance of AAMPC (red).

For the on-line assessment of the control performance, the following counterpart of (4) is used:

$$\tilde{L}_k([d];T) := k^{-1} \sum_{i=0}^{k-1} [x_{i+1}^T x_{i+1} + \lambda (u_i^d)^2]. \quad (9)$$

In particular, in Fig. **4**, the loss $\tilde{L}_k$ is used to show the AAMPC control performance with respect to policies based on fixed models within the entire duration of the experiment. As it can be observed, the performance of the AAMPC policy follows closely that of the controllers based on the best dictionary components and is robust against the influence of the three single-model strategies with increasing (or at least constant) loss $\tilde{L}_k$.

6. Conclusions

In this paper, we have proposed a new adaptive model predictive control algorithm based on the adversarial multi-armed bandit decision framework. In contrast to standard MPC techniques, the novel AAMPC policy does not require an accurate *a priori* knowledge about the control system but is based on a finite set of estimates of the process. At the heart of the AAMPC approach lies a probabilistic self-adapting mechanism that explores dictionary components and exploits the best-performing models. In general, since the considered self-adaptation imposes very mild requirements on the models/experts, the proposed scheme could also be applied to other control methods such as PIDs, fixed-structure parametric controllers, or even to nonlinear ones. As shown in Theorem 1, the applied exploration-exploitation technique leads to a control performance comparable (in the sense of inequality (7)) to that attainable by the best model in the dictionary, with a diminishing regret of order $O(N^{-1/2})$. The empirical performance of AAMPC has also been verified via simulation studies.

For future work, we consider a further theoretical analysis of the algorithm, particularly its closed-loop stability and extended numerical examinations.

7. Proof of Theorem 1

The proof is based on [28, Th.~11.1]. To simplify the notation, we will omit the dependence on T.

Define $\rho_n(d) := 1 - (\mathbb{I}\{d = \delta\} l_n(\delta))/(C_l P_n(d))$, and note that according to Lemma 1 (see Section 8), $\rho_n(d)$ is an unbiased[1] estimate of the normalized reward $r_n(d) := 1 - l_n(d)/C_l$, *i.e.*, $E\{\rho_n(d)\} = r_n(d)$. Observe also that $\rho_n(d)$ is bounded from above, *i.e.*, $\rho_n(d) \leqslant 1$.

[1] All the expectations are taken with respect to the distribution of model selections (not with respect to the noise distribution).

Step 1) For the proof, it is enough to show that $E\{\sum_{n=1}^{N} l_n(\delta)\} - \sum_{n=1}^{N} l_n(d)$ is upper bounded, for all d, by $2C_l\sqrt{N\text{Ɗ}\, ln(\text{Ɗ})}$. Equivalently (since $l_n(d) = C_l(1 - r_n(d))$), we aim to show that

$$\sum_{n=1}^{N} r_n(d) - E\{\sum_{n=1}^{N} r_n(\delta)\} \leqslant 2\sqrt{N\text{Ɗ}\, ln(\text{Ɗ})}. \tag{10}$$

From Lemmas 2 and 3 (see Section 8), we note that

$$\begin{aligned} \sum_{n=1}^{N} r_n(d) &= E\{S_N(d)\} \\ E\{\sum_{n=1}^{N} r_n(\delta)\} &= E\{\sum_{n=1}^{N}\sum_{d=1}^{\text{Ɗ}} P_n(d)\rho_n(d)\} \quad , \end{aligned} \tag{11}$$

where $S_N(d) := \sum_{n=1}^{N}[1 - (\mathbb{I}\{d = \delta\}l_n(\delta))/(C_l P_n(d))]$. Therefore, the left hand side of (10) is equal to

$$R_N = E\{S_N(d) - k_N\}, \tag{12}$$

where $k_N := \sum_{n=1}^{N} \sum_{d=1}^{D} P_n(d)\rho_n(d)$.

Step 2) To bound (12), let $w_N := \sum_{d=1}^{\text{Ɗ}} exp(\eta S_N(d))$. Note that $w_0 = \sum_{d=1}^{\text{Ɗ}} exp(0) = \text{Ɗ}$. Furthermore,

$$\begin{aligned} exp(\eta S_N(d)) \quad &\leqslant \sum_{d=1}^{\text{Ɗ}} exp(\eta S_N(d)) \\ &= w_N = w_0 \frac{w_1}{w_0} \cdots \frac{w_N}{w_{N-1}}, \end{aligned} \tag{13}$$

and hence

$$exp(\eta S_N(d)) \leqslant \text{Ɗ} \prod_{n=1}^{N} \frac{w_n}{w_{n-1}}. \tag{14}$$

Focusing on w_n/w_{n-1} observe that

$$w_n/w_{n-1} = \sum_{d=1}^{\text{Ɗ}} \frac{exp(\eta S_{n-1}(d))}{w_{n-1}} exp(\eta\rho_n(d)).$$

Due to (5), we obtain

$$w_n/w_{n-1} = \sum_{d=1}^{\text{Ɗ}} P_n(d)\, exp(\eta\rho_n(d)). \tag{15}$$

Next, since $\eta \leqslant 1$, $\rho_n(d) \leqslant 1$, and $exp(x) \leqslant 1 + x + x^2$ for all $x \leqslant 1$, we have that $exp(\eta\rho_n(d)) \leqslant 1 + \eta\rho_n(d) + \eta^2\rho_n^2(d)$. Hence, for (15), we have

$\frac{w_n}{w_{n-1}} \leqslant 1 + \eta \sum_{d=1}^{\text{Ɗ}} P_n(d)\rho_n(d) + \eta^2 \sum_{d=1}^{\text{Ɗ}} P_n(d)\rho_n^2(d)$ .As $1 + x \leqslant exp(x)$ for all $x \in \mathbb{R}$,

$$\frac{w_n}{w_{n-1}} \leqslant exp\left(\eta \sum_{d=1}^{\text{Ɗ}} P_n(d)\rho_n(d) + \eta^2 \sum_{d=1}^{\text{Ɗ}} P_n(d)\rho_n^2(d)\right).$$

An application of the above result in (14) gives

$$exp(\,\eta S_N(d)) \leqslant \mathcal{D}\, exp\left(\eta \sum_{n=1}^{N} \sum_{d=1}^{\mathcal{D}} P_n(d)\rho_n(d) + \eta^2 \sum_{n=1}^{N} \sum_{d=1}^{\mathcal{D}} P_n(d)\rho_n^2(d)\right). \tag{16}$$

Equivalently,

$exp(\,\eta S_N(d)) \leqslant \mathcal{D}\, exp\big(\eta k_N + \eta^2\, \sum_{k=1}^{N} \sum_{d=1}^{\mathcal{D}}\, P_n(d)\rho_n^2(d)\big)$, and after taking the logarithm of both sides, we obtain

$$S_N(d) - k_N \leqslant \frac{ln\,\mathcal{D}}{\eta} + \eta \sum_{n=1}^{N} \sum_{d=1}^{\mathcal{D}} P_n(d)\rho_n^2(d), \tag{17}$$

where the expectation of the left hand side is equal to (12).

Step 3) Focusing on the expectation of the last (double-summation) element in (17) we see that

$$E\{\sum_{n=1}^{N} \sum_{d=1}^{\mathcal{D}} P_n(d)\rho_n^2(d)\} = \sum_{n=1}^{N} E\{1 - 2\bar{l}_n(\delta) + E_n\{\sum_{d=1}^{\mathcal{D}} \frac{\mathbb{I}\{d=\delta\}\bar{l}_n^2(\delta)}{P_n(d)}\}\}, \tag{18}$$

where $\bar{l}_n(\delta) := l_n(\delta)/C_l$ (*i.e.*, $\bar{l}_n(\delta) \in [0,1]$) and E_n is the expectation conditioned by prior decisions and rewards, *i.e.* , $E_n\{\cdot\} = E_n\{\cdot\,|\delta_1, r_1, \ldots, \delta_{n-1}, r_{n-1}\}$. Hence, (18) is equal to

$$\sum_{n=1}^{N} E\{1 - 2\bar{l}_n(\delta) + \sum_{d=1}^{\mathcal{D}} \bar{l}_n^2(d)\}$$
$$= \sum_{n=1}^{N} E\{[1 - \bar{l}_n(\delta)]^2 + \sum_{d=1;d\neq\delta}^{\mathcal{D}} \bar{l}_n^2(d)\}.$$

Note, next, that the right-hand side in the formula above is upper-bounded by $NE\{[1 - \bar{l}_n(\delta)]^2 + \sum_{d=1;d\neq\delta}^{\mathcal{D}} \bar{l}_n^2(d)\} \leqslant NE\{1 + \sum_{d=1;d\neq\delta}^{\mathcal{D}} 1\} \leqslant N\mathcal{D}$. Finally, we see from the above, (17) and (12) that

$$R_N \leqslant \eta^{-1}\, ln\,\mathcal{D} + \eta N\mathcal{D}. \tag{19}$$

Step 4) The value of η for which the upper bound in (19) is minimized is equal to $\sqrt{ln\,\mathcal{D}\,/(N\mathcal{D})}$, which leads to the final result $R_N \leqslant 2\sqrt{N\mathcal{D}\, ln\,\mathcal{D}}$.

8. Additional Lemmas

Lemma 1 $\rho_n(d)$ is an unbiased estimate of the normalized reward $r_n(d)$, i.e. , $E\{\rho_n(d)\} = r_n(d)$ for all d.

Proof. We have that

$$E\{\rho_n(d)\} = 1 - (E\{\mathbb{I}\{d=\delta\}l_n(\delta)\})/(C_l P_n(d))$$

and $E\{\mathbb{I}\{d=\delta\}l_n(\delta)\} = P_n(d)l_n(d)$. Therefore,

$$E\{\rho_n(d)\} = 1 - \frac{P_n(d)l_n(d)}{C_l P_n(d)} = 1 - \frac{l_n(d)}{C_l}.$$

Lemma 2 It holds that $\sum_{n=1}^{N} r_n(d) = E\{S_N(d)\}$.

Proof. The result follows from Lemma 1, since $E\{S_N(d)\} = \sum_{n=1}^{N} E\{\rho_n(d)\} = \sum_{n=1}^{N} r_n(d)$.

Lemma 3 It holds that

$$E\{\sum_{n=1}^{N} r_n(\delta)\} = E\{\sum_{n=1}^{N} \sum_{d=1}^{\mathcal{D}} P_n(d)\rho_n(d)\}.$$

Proof. Observe that $E_n\{r_n(d)\} = \sum_{d=1}^{\mathcal{D}} P_n(d) r_n(d)$, where E_n is the expectation conditioned by prior decisions and rewards. Based on Lemma 1, $E\{\rho_n(d)\} = r_n(d)$. Therefore, $E_n\{r_n(\delta)\} = \sum_{d=1}^{\mathcal{D}} P_n(d) E\{\rho_n(d)\}$ and $E\{\sum_{n=1}^{N} r_n(\delta)\} = E\{\sum_{n=1}^{N} E_n\{r_n(\delta)\}\}$, which is equal to $E\{\sum_{n=1}^{N} \sum_{d=1}^{\mathcal{D}} P_n(d)\rho_n(d)\}$.

References

[1] Aggelogiannaki E, Sarimveis H. Multiobjective constrained MPC with simultaneous closed-loop identification. Int. J. Adapt. Control and Sig. Proc., 20(4):145-173, 2006. https://doi.org/10.1002/acs.892

[2] Arora S, Hazan E, Kale S. The multiplicative weights update method: a meta-algorithm and applications. Th. of Computing, 8:121-164, 2012.

[3] Åström KJ,. Wittenmark B. Adaptive Control, 2nd Edition. Addison-Wesley, 1995.

[4] Auer P, Cesa-Bianchi N, Freund Y, Schapire RE. The nonstochastic multi-armed bandit problem. SIAM J. Comput., 32(1):48-77, 2002. https://doi.org/10.1137/S0097539701398375

[5] Bubeck S, Cesa-Bianchi N. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. Foundations and Trends in Machine Learning, 5(1):1-122, 2012. https://doi.org/10.1561/2200000024

[6] Cesa-Bianchi N, Lugosi G. Prediction, Learning, and Games. Cambridge Univ. Press, 2006. https://doi.org/10.1017/CBO9780511546921

[7] Coulson J, Lygeros J, Dörfler F. Data-enabled predictive control: In the shallows of the DeePC. In Proc. 18th Eur. Control Conf., pages 307-312, 2019. https://doi.org/10.23919/ECC.2019.8795639

[8] Ebadat A, Annergren M, Larsson CA, Rojas CR, Wahlberg B, Hjalmarsson H, Sjöberg J. Application set approximation in optimal input design for model predictive control. In Proc. 13th Eur. Control Conf., pages 744-749, Strasbourg, France, 2014. https://doi.org/10.1109/ECC.2014.6862496

[9] Ebadat A, Valenzuela PE, Rojas CR, Wahlberg B. Model predictive control oriented experiment design for system identification: A graph theoretical approach. J. Proc. Control, 52:75-84, 2017. https://doi.org/10.1016/j.jprocont.2017.02.001

[10] Goodwin GC, Seron MM, De Don A. Constrained Control and Estimation: An Optimisation Approach. Springer, 2005. https://doi.org/10.1007/b138145

[11] Grüne L, Pannek J. Nonlinear Model Predictive Control, 2nd Edition. Springer, 2017. https://doi.org/10.1007/978-3-319-46024-6

[12] Hale ET, Qin SJ. Subspace model predictive control and a case study. In Proc. Amer. Control Conf., pages 4758-4763, 2002. https://doi.org/10.1109/ACC.2002.1025411

[13] Heirung TAN, Foss B, Ydstie BE. MPC-based dual control with online experiment design. J. Proc. Control, 32:64-76, 2015. https://doi.org/10.1016/j.jprocont.2015.04.012

[14] Iannelli A, Khosravi M, Smith RS. Structured exploration in the finite horizon linear quadratic dual control problem. arXiv preprint arXiv:1910.14492, 2019. https://doi.org/10.1016/j.ifacol.2020.12.1263

[15] Katayama T. Subspace Methods for System Identification. Springer, 2005. https://doi.org/10.1007/1-84628-158-X

[16] Kumar PR. An adaptive controller inspired by recent results on learning from experts. In K.J. Å ström, G.C. Goodwin, and P.R. Kumar, editors, Adapt. Control, Filtering, and Sig. Proc., pages 199-204. Springer, 1995. https://doi.org/10.1007/978-1-4419-8568-2_8

[17] Larsson CA, Annergren M, Hjalmarsson H, Rojas CR, Bombois X, Mesbah A, Modén PE. Model predictive control with integrated experiment design for output error systems. In Proc. Eur. Control Conf., pages 3790-3795, Zurich, Switzerland, 2013. https://doi.org/10.23919/ECC.2013.6669533

[18] Larsson CA, Ebadat A, Rojas CR, Bombois X, Hjalmarsson H. An application-oriented approach to dual control with excitation for closed-loop identification. Eur. J. of Control, 29:1-16, 2016. https://doi.org/10.1016/j.ejcon.2016.03.001

[19] Lattimore T, Szepesvári C. Bandit algorithms. Cambridge Univ. Pr., 2020. https://doi.org/10.1017/9781108571401

[20] Ljung L. System Identification: Theory for the User, 2nd Edition. Prentice Hall, 1999.

[21] Löfberg J. YALMIP: A toolbox for modeling and optimization in MATLAB. In Proc. CACSD Conf., pages 284-289, Taipei, Taiwan, 2004.

[22] Lorenzen M, Cannon M, Allgöwer F. Robust mpc with recursive model update. Automatica, 103:461-471, 2019. https://doi.org/10.1016/j.automatica.2019.02.023

[23] Marafioti G, Bitmead RR, Hovd M. Persistently exciting model predictive control. Int. J. Adapt. Cont. Sig. Proc., 28(6):536-552, 2014. https://doi.org/10.1002/acs.2414

[24] Marafioti G. Enhanced model predictive control: dual control approach and state estimation issues. Ph.d. thesis, Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), Norway, 2010.

[25] Mardi N. Data-driven subspace-based model predictive control. Ph.d. thesis, RMIT University, Australia, 2010.

[26] Mesbah A. Stochastic model predictive control with active uncertainty learning: A survey on dual control. Ann. Rev. Control, 45:107-117, 2018. https://doi.org/10.1016/j.arcontrol.2017.11.001

[27] Van Overschee P, De Moor BL. Subspace identification for linear systems: Theory-Implementation-Applications. Springer, 2012.

[28] Parsi A, Iannelli A, Smith RS. An explicit dual control approach for constrained reference tracking of uncertain linear systems. IEEE Transactions on Automatic Control, 2022. https://doi.org/10.1109/TAC.2022.3176800

[29] Patwardhan RS, Gopaluni RB. A moving horizon approach to input design for closed loop identification. J. Proc. Control, 24(3):188-202, 2014. https://doi.org/10.1016/j.jprocont.2013.10.018

[30] Raginsky M, Rakhlin A, Yuksel S. Online convex programming and regularization in adaptive control. In Proc. 49th IEEE Conf. Decision and Control, pages 1957-1962, Atlanta, USA, 2010. https://doi.org/10.1109/CDC.2010.5717262

[31] Rallo G, Formentin S, Rojas CR, Oomen T, Savaresi SM. Data-driven -norm estimation via expert advice. In Proc. 56th IEEE Conf. Decision and Control, pages 1560-1565, Melbourne, Australia, 2017.

[32] Rawlings JB, Mayne DQ, Diehl MM. Model Predictive Control: Theory, Comp., and Design, 2nd Ed. Nob Hill Pub., 2017.

[33] Shouche MS, Genceli H, Nikolaou M. Effect of on-line optimization techniques on model predictive control and identification (mpci). Computers & Chemical Engineering, 26(9):1241-1252, 2002. https://doi.org/10.1016/S0098-1354(02)00091-1

[34] Shouche M, Genceli H, Vuthandam P, Nikolaou M. Simultaneous constrained model predictive control and identification of DARX processes. Automatica, 34(12):1521-1530, 1998. https://doi.org/10.1016/S0005-1098(98)80005-8

[35] Sutton RS, Barto AG. Reinforcement Learning, An Introduction, 2nd Ed. MIT Press, 2018.

[36] Wachel P, Sliwinski, P. Aggregative modeling of nonlinear systems. IEEE Signal Processing Letters, 22(9):1482-1486, 2015. https://doi.org/10.1109/LSP.2015.2405613

[37] Wachel P. Wiener system modelling by exponentially weighted aggregation. International Journal of Control, 90(11):2480-2489, 2017. https://doi.org/10.1080/00207179.2016.1254818

[38] Willems JC, Rapisarda P, Markovsky I, De Moor BLM. A note on persistency of excitation. Syst. & Contr. Lett., 54(4):325-329, 2005. https://doi.org/10.1016/j.sysconle.2004.09.003

[39] Zhu YC. System identification for process control: recent experiment and outlook. In IFAC Symposium on System Identification, pages 89-103, Newcastle, Australia, 2006. Plenary presentation, available at http://taijicontrol.com/SYSID06YZhu.pdf.

[40] Zinkevich M. Online convex programming and generalized infinitesimal gradient ascent. In Proc. Int. Conf. Mach. Learn. (ICML), pages 928-936, 2003.