A Sampling-and-Discarding Approach to Stochastic Model Predictive Control for Renewable Energy Systems ⋆

Balázs Cs. Csáji * **Krisztián B. Kis** * **András Kovács** *

* *Institute for Computer Science and Control (SZTAKI), Budapest, Kende utca 13-17, Hungary, H-1111*
{balazs.csaji, krisztian.kis, andras.kovacs}@sztaki.hu

Abstract: The paper applies the scenario approach to stochastic model predictive control for renewable energy systems. First, the controllable and the (quasi-periodic) uncontrollable parts are decomposed. The latter is modeled by a Box-Jenkins system with appropriately chosen inputs. For the controllable part, a linear state space model is used with an affine state-feedback controller. Several numerical experiments are presented on a public lighting microgrid, e.g., about forecasting the energy balance, the effects of various controller parametrizations, reoptimization frequencies, and discarding unfavorable scenarios. The results indicate that even a low order, time-independent controller with a slow reoptimization frequency can be efficient.

Keywords: stochastic model predictive control, chance-constrained optimization, partially controllable systems, randomized methods, scenario approach, renewable energy systems

1. INTRODUCTION

Model predictive control (MPC) or *receding horizon control* (RHC) is a widespread control design technology that builds on advancements in mathematical optimization and provides solutions which can handle flexible constraints imposed on the states and the inputs (Maciejowski, 2002).

The parameters of the models are often *uncertain*, since they are typically based on empirical data, which should be taken into account during optimization. *Robust* MPC optimizes w.r.t. the worst-case situation, while *stochastic* MPC incorporates a probabilistic description of the uncertainties and uses chance-constraints, namely, it allows constraint violations with some given probability. This, however, often leads to hard problems (Mesbah, 2016).

The *scenario approach* provides a promising compromise (Calafiore and Campi, 2006; Garatti and Campi, 2013). It assumes that we can generate i.i.d. (independent and identically distributed) samples of the uncertainties. This approach leads to efficient MPC solutions with *distribution-free* stochastic guarantees (Calafiore and Fagiano, 2012; Schildbach et al., 2014). Scenario-based MPC has many applications from river control (Nasir et al., 2018) to dispatching in power systems (Modarresi et al., 2019).

The paper investigates the applicability of scenario-based stochastic model predictive control (SMPC) to a real *renewable energy system*, a public lighting microgrid, that has a bi-directional connection to the power grid, and contains photovoltaic (PV) panels, LED luminaries, which regulate their lighting levels, and a battery. The control objective is to minimize the total energy cost with constraints on limited-time island mode operations, in case of power outages (Kovács et al., 2016). Such systems are of course only *partially controllable* as the energy production and consumption are determined by external factors.

We start by *decomposing* the states into fully controllable and uncontrollable, but observable, parts. We apply a Box-Jenkins model with special inputs for the uncontrollable part, and compare its performance with NARX models.

The process noise is estimated by its empirical distribution function (cf. bootstrap) taking the time-varying nature of the problem into account, as the system is quasi-periodic. With the identified model and the noise distribution at hand, samples of potential future trajectories ("scenarios") can be generated a priori, before optimization.

Linear state space models are used for the controllable part with an affine feedback controller. Various parametrizations are considered for the controller, as the number of decision variables heavily influences the scenario approach, especially if some of the scenarios are discarded.

Finally, several numerical experiments are presented on the aforementioned public lighting microgrid. Potential models for the uncontrollable part are evaluated and the efficiency of scenario-based SMPC is investigated. Simulation studies on various parametrizations, optimization horizons and reoptimization frequencies are shown.

2. DECOMPOSABLE MARKOV MODEL

We interact with an uncertain dynamical system that we model with the following (possibly nonlinear) process,

$$x_t \,=\, f(\,x_{t-1}, u_t, \varepsilon_t\,), \tag{1}$$

where $x_t \in \mathbb{R}^{d_x}$ is the state, $u_t \in \mathbb{R}^{d_u}$ is the control input, and $\varepsilon_t \in \mathbb{R}_t^{d_\varepsilon}$ is the driving noise term, at time t.

⋆ This research has been supported by the grants ED_18-2-2018-0006 (NKFIH) and GINOP_2.3.2-15-2016-00002. A. Kovács acknowledges the support of the János Bolyai Research Scholarship.

In some situations $\{x_t\}$ is only *partially controllable* and it can be *decomposed* into two parts (possibly after a state transformation), $\{x'_t\}$ and $\{x''_t\}$, where $\{x''_t\}$ is unaffected by the chosen inputs (uncontrollable). Recall that in the classical case of linear time-invariant (LTI) systems, such states exist if the controllability matrix is not full rank, and the aforementioned partition of states can be achieved by a suitable change of basis (cf. Kálmán decomposition). For the nonlinear case, see for example (Cheng et al., 2010).

Having such a decomposition allows us to write

$$x'_t \;=\; f'(\,x_{t-1}, u_t, \varepsilon_t\,), \tag{2}$$

$$x''_t \;=\; f''(\,x''_{t-1}, \varepsilon_t\,), \tag{3}$$

where functions f' and f'' are the controlled and uncontrolled parts of the dynamics, respectively. Note that x'_t is allowed to depend on x''_{t-1}, but not the other way around.

We consider time-varying, state-feedback controllers, i.e.,

$$u_t \;=\; \pi_t(x_{t-1}), \tag{4}$$

where function π_t is called the (Markov) *control policy* at time t. A sequence of control policies, $\{\pi_t\}$, is denoted by π and the set of all such policy sequences is denoted by Π.

Henceforth, we will make a notational distinction between *generic* state and input variables, $\{x_t\}$ and $\{u_t\}$, and the *realized* ones, $\{x^*_t\}$ and $\{u^*_t\}$. Similarly, π_t will denote a generic policy, while the "true", realized policy is π^*_t.

3. MODEL PREDICTIVE CONTROL

In this section we briefly overview the relevant approaches and results from the theory of model predictive control.

At first, for simplicity, assume that there is no external driving process, therefore the system is *deterministic*. Let us denote the *current time* by t_0, then *ideally* we aim at finding a policy π that optimizes the *cost-to-go* or *value* function J^π which is $\mathbb{R}$-valued and often takes an additive form summing the immediate (or stage) cost of each visited state-input pair. Formally, the cost-to-go of policy π is

$$J^\pi_\infty(x^*_{t_0}) \;\doteq\; \sum_{k=0}^{\infty} \ell_k(x^*_{t_0+k}, u^*_{t_0+k+1}), \tag{5}$$

where $\{\ell_k\}$ are (given) $\mathbb{R}$-valued immediate-cost functions. Dependence on k allows, e.g., *discounting* future costs. An issue is that at time t_0 we only know $\{x^*_t\}$ up to $x^*_{t_0}$, but do not know exactly how the process will evolve using π.

MPC uses the model to estimate the progress of the system and replaces J^π with a finite horizon version, J^π_n,

$$J^\pi_\infty(x^*_{t_0}) \;\approx\; J^\pi_n(x_0) \;\doteq\; \sum_{k=0}^{n-1} \ell_k(x_k, u_{k+1}) + \ell^\top_n(x_n), \tag{6}$$

where $x_0 = x^*_{t_0}$ and $x_1, \dots, x_n$ are generated by the model and $\ell^\top_n$ is an optional term for the terminal cost (e.g., an estimate of the remaining costs). Then, we optimize J^π_n (optionally w.r.t. some constraints on the states and inputs) on a *receding horizon*. Namely, after optimization, we only execute the first input, $\pi_1(x_0)$ becomes $u^*_{t_0}$, and we discard the rest. After obtaining $x^*_{t_0+1}$ at t_0+1, we repeat the process: we solve another optimization problem, now starting from $x^*_{t_0+1}$, execute its first input, and so on.

Formally, the optimization scheme of MPC is as follows:

$$\begin{array}{ll} \underset{\pi\in\Pi}{\text{minimize}} & J^\pi_n(x_0) \\ \text{subject to} & x_0 = x^*_{t_0} \\ & u_k = \pi_k(x_{k-1}) \\ & x_k = f(x_{k-1}, u_k) \\ & u \in \mathcal{U},\; x \in \mathcal{X} \\ & k = 1, \dots, n \end{array} \tag{7}$$

where $x \doteq (x_0, x_1, \dots, x_n)^{\mathrm{T}}$, $u \doteq (u_0, u_1, \dots, u_n)^{\mathrm{T}}$, $\pi \doteq (\pi_1, \dots, \pi_n)$ are sequences of states, inputs and policies, respectively, and $\mathcal{X}, \mathcal{U}$ are given (constant) constraint sets for (the allowed sequences of Markov) states and inputs.

This problem is an abstract description of the optimization involved in an MPC step, but in general it is *intractable*. Furthermore, unless some assumptions on the policies are made, the problem can even be *infinite dimensional*.

This latter problem can be addressed, e.g., if the control policies are parametrized by a *finite dimensional* vector.

In the deterministic case, it is often enough to optimize over input *sequences* resulting in (open-loop) *planning* problems. Even in this case the receding horizon approach ensures a *closed-loop* overall behavior, as the starting state is updated in each MPC step (Maciejowski, 2002). In order to make the optimization problem above tractable, J, f, $\mathcal{X}$ and $\mathcal{U}$ are typically chosen to be *convex*.

If the system dynamics can be decoupled into a controllable, f', and uncontrollable, f'', part, then by definition all uncontrolled states $\{x''_k\}$ can be calculated in advance. Hence, they can be treated as *constants* from the viewpoint of optimization, and therefore *any* (even nonconvex) f'' can be used without making the problem intractable.

Now, assume that the system is *uncertain*, i.e., there are $\{\varepsilon_k\}$ variables in the model. If we assume that they are *stochastic*, then $J^\pi_n(x_0)$ becomes a random variable and therefore a type of *stochastic dominance* concept should be selected to make the optimization meaningful. A standard choice is to optimize the *expected value* of the objective function with (joint) *chance constraints*, which leads to the following stochastic programming problem:

$$\begin{array}{ll} \underset{\pi\in\Pi}{\text{minimize}} & \mathbb{E}\,[\,J^\pi_n(x_0)\,] \\ \text{subject to} & x_0 = x^*_{t_0} \\ & u_k = \pi_k(x_{k-1}) \\ & x_k = f(x_{k-1}, u_k, \varepsilon_k) \\ & \mathbb{P}_\varepsilon\,(\,u \in \mathcal{U},\; x \in \mathcal{X}\,) \geq 1-\delta \\ & k = 1, \dots, n \end{array} \tag{8}$$

where $\varepsilon \doteq (\varepsilon_1, \dots, \varepsilon_n)$ is a random matrix, and δ is the (given) allowed probability of *constraint violation*.

There are a lot of variants of this problem, e.g., there could be *individual* chance constraints (if $\mathcal{X}$ builds up from the intersection of several convex sets) or *hard* constraints (which should always be satisfied, e.g., for all inputs), see (Mesbah, 2016). These problems are typically hard to solve, unless some sampling-based approximation is used. Then, the expectation in the objective is often replaced by its sample average, or the maximum over the samples corresponding to *value-at-risk* (Schildbach et al., 2014).

An arch-typical choice for f is the case of *linear* dynamics

$$x_t \;=\; f(x_{t-1}, u_t, \varepsilon_t) \;=\; A\, x_{t-1} + B\, u_t + \varepsilon_t, \qquad (9)$$

where A and B are (constant) matrices of appropriate size.

For uncertain systems it is advantageous to optimize over *feedback* controllers (Mayne et al., 2000), instead of control sequences. A typical choice is to use a *linear* controller, i.e., $u_t = K_t x_{t-1}$, which however, may lead to *nonconvex* optimization problems (Goulart et al., 2006).

This nonconvexity issue of the resulting MPC optimization can be avoided by observing that (Nasir et al., 2018)

$$\begin{aligned} u_t \;=\; K_t x_{t-1} \;&=\; K_t \prod_{i=1}^{t-1} (A + BK_i)\, x_0 + \\ &\sum_{j=2}^{t-1} K_t \prod_{i=j}^{t-1} (A + BK_i) \varepsilon_{j-1} + K_t \varepsilon_{t-1}, \end{aligned} \qquad (10)$$

with $\varepsilon_0 = 0$, thus, the policy can also be parametrized as

$$u_t \;=\; \pi_t(x_{t-1}) \;=\; \varphi_t \;+\; \sum_{i=1}^{t-1} \Phi_{t,i} \varepsilon_i, \qquad (11)$$

to ensure that we have a *convex* problem (Goulart et al., 2006). Note that, of course, $\varepsilon_t = x_t - A\, x_{t-1} - B\, u_t$, thus, available. This solution is called the *full* affine controller.

As we will see, the number of decision variables plays an important role in the scenario approach, thus, we consider two subclasses of affine controllers, which require less parameters to describe. The first one is defined as

$$u_t \;=\; \pi_t(x_{t-1}) \;=\; \varphi_t \;+ \sum_{\substack{i\,=\,1\vee \\ (t-1-p)}}^{t-1} \Phi_{t,i} \varepsilon_i, \qquad (12)$$

where $a \vee b = \max(a, b)$, to which we will refer as a (time-varying) affine controller with *past order* p.

Finally, if $\varphi_t = \varphi$ and $\Phi_{t,i} = \Phi_i$ for all t, i, then we talk about a *time-independent* controller with past order p.

An alternative, which even works for *nonstochastic* uncertainties, is to look for a *worst-case* solution which is *robust* w.r.t. all possible uncertainties. Henceforth, we assume that our policies are affine and they are parametrized by $\theta \in \mathbb{R}^m$, i.e., θ encodes vectors $\{\varphi_t\}$ and matrices $\{\Phi_{i,t}\}$.

Given a scalar h, we will use the notation $(h;\theta) \doteq (\,h, \theta^{\mathrm{T}})^{\mathrm{T}}$. Let us introduce uncertainty-dependent constraint sets:

$$\begin{aligned} \mathcal{Z}(\varepsilon) \;\doteq\; \big\{\, &(h;\theta) \in \mathbb{R}^{m+1} : J_n^{\pi,\varepsilon}(x_0^{\varepsilon}) \,\leq\, h,\; x_0^{\varepsilon} = x_{t_0}^{*}, \\ &u_k^{\varepsilon} = \pi_k(\, x_{k-1}^{\varepsilon} \mid \theta\,),\; x_k^{\varepsilon} = f(\, x_{k-1}^{\varepsilon}, u_k^{\varepsilon}, \varepsilon_k\,), \\ &x^{\varepsilon} \in \mathcal{X},\; u^{\varepsilon} \in \mathcal{U},\; k = 1, \dots, n \,\big\}, \end{aligned} \qquad (13)$$

where $\varepsilon \in \mathbb{R}^n$ is a given uncertainty sequence, and $\pi_k(u) \doteq \pi_k(u \mid \theta_k)$ is a notation to emphasize that $\{\pi_k\}$ are parametrized by θ. We also assume that $J_n^{\pi,\varepsilon}(x_{t_0}^{*})$ is a convex function in θ, thus $\mathcal{Z}(\varepsilon)$ is convex for all ε.

Then, the optimization step of *robust* MPC becomes:

$$\begin{aligned} &\underset{h \in \mathbb{R},\, \theta \in \mathbb{R}^m}{\text{minimize}} \quad h \\ &\text{subject to} \quad (h;\theta) \in \mathcal{Z}(\varepsilon),\; \text{for all}\; \varepsilon \in E, \end{aligned} \qquad (14)$$

where E is the (not necessarily convex) set of all possible uncertainties. Obviously, h is the maximum of $J_n^{\pi,\varepsilon}(x_{t_0}^{*})$, where the maximum is taken over E, hence, this program minimizes the objective w.r.t. the worst-case situation.

Note that typical issues with such worst-case approaches are that (i) they often lead to overly *conservative* solutions; moreover, (ii) the optimization problem above is, in general, *semi-infinite*, as E can be an uncountable set.

4. SCENARIO-BASED STOCHASTIC MPC

If the problem is *stochastic* (i.e., there is a probability distribution over the uncertainties), and we have access to a *generative model* (e.g., we can simulate i.i.d. realizations of variable ε), then the *scenario approach* is a promising technique to modulate the robustness of the solution, see (Calafiore and Campi, 2006; Garatti and Campi, 2013).

Assume we generate N i.i.d. "scenarios", $\varepsilon^{(1)}, \dots, \varepsilon^{(N)}$; then, we can define the following (random) scenario problem as an approximation of the original worst-case one:

$$\begin{aligned} &\underset{h \in \mathbb{R},\, \theta \in \mathbb{R}^m}{\text{minimize}} \quad h \\ &\text{subject to} \quad (h;\theta) \in \mathcal{Z}(\varepsilon^{(i)}),\; \text{for}\;\; i = 1, \dots, N, \end{aligned} \qquad (15)$$

which (sampled) problem is now a standard (finite) convex programming problem, assuming $\{\mathcal{Z}(\varepsilon^{(i)})\}$ are convex. Let us denote the optimal solution by $z_N^{\star} \doteq (h^{\star};\theta^{\star})$. A natural question is: what can we say about the rest of the uncertainties, for example, how much of them are violated?

The *violation probability* of a fixed $z \in \mathbb{R}^{m+1}$ is defined by

$$V(z) \;\doteq\; \mathbb{P}\,\{\, \varepsilon \in E : z \notin \mathcal{Z}(\varepsilon)\,\}\,. \qquad (16)$$

In other words, $V(z) \in [0, 1]$ quantifies the probability that a randomly selected noise realization sequence, ε, results in a constraint, $\mathcal{Z}(\varepsilon)$, that is violated by $z = (h;\theta)$.

Remark: $z_N^{\star}$ is a *random variable*, since the scenario problem (15) depends on the randomly sampled $\varepsilon^{(1)}, \dots, \varepsilon^{(N)}$, hence, its violation probability, $V(z_N^{\star})$, is also *random*.

Nevertheless, it can be proved (Garatti and Campi, 2013) that the violation probability of $z_N^{\star} \doteq (h^{\star};\theta^{\star})$ satisfies

$$\mathbb{P}\,\{\, V(z_N^{\star}) > \delta \,\} \;\leq\; \beta(\delta, d, N), \qquad (17)$$

for all $\delta \in (0, 1)$, assuming (15) is feasible and has a unique solution for all $\{\varepsilon^{(i)}\}$, where $\beta(\delta, d, N)$ is the tail of the *beta distribution* with parameters $(d, N - d + 1)$, that is

$$\beta(\delta, d, N) \;\doteq\; \sum_{i=0}^{d-1} \binom{N}{i} \delta^i (1 - \delta)^{N-i}, \qquad (18)$$

where $d = \dim(z_N^{\star})$, i.e., the number of decision variables.

Surprisingly, the bound on the violation probability $V(z_N^{\star})$ is independent of the probability distribution by which the scenarios, $\{\varepsilon^{(i)}\}$, are generated. Further, as the number of scenarios grows, $\beta(\delta, d, N)$ tends to zero exponentially fast.

Given a maximum allowed constraint violation probability, δ, and a confidence probability, $1 - \beta$, we can compute the sample size that is sufficient by finding the smallest integer, N, such that $\beta(\delta, d, N) \,\leq\, \beta$, for example, by *binary* (logarithmic) search.[1] Then, we can claim with

[1] Suitable upper and lower bounds for the initialization of the logarithmic search can be found in (Garatti and Campi, 2013).

confidence at least $1-\beta$ that the constraints are satisfied at least with probability $1-\delta$ (Garatti and Campi, 2013).

An explicit expression can also be provided for the needed number of scenarios (Alamo et al., 2015) given δ, β, namely

$$N \;\geq\; {}^1\!/\!_\delta \left(d-1+\log({}^1\!/\!_\beta)+\sqrt{2(d-1)\log({}^1\!/\!_\beta)}\right), \quad (19)$$

which guarantees that $\mathbb{P}\{\, V(z_N^\star) \leq \delta \,\} \;\geq\; 1-\beta$.

Note that this formula only logarithmically depends on β, thus, even if β is very small, it does not increase N much.

The effectiveness of the obtained solution can often be increased if some scenarios are discarded. Of course, removing constraints increases the constraint violation probability. Nonetheless, it can be shown (Campi and Garatti, 2011) that if we remove k scenarios and β satisfies

$$\binom{k+d-1}{k} \sum_{i=0}^{k+d-1} \binom{N}{i} \delta^i (1-\delta)^{N-i} \;\leq\; \beta, \quad (20)$$

then we can guarantee $V(z_{N-k}^\star) \;\leq\; \delta$ with confidence $1-\beta$.

This result is independent of the algorithm for selecting the scenarios to be removed, but k should be chosen a priori. Trying all possible choices of k scenarios would be computationally very expensive, but one can choose, e.g., the constraints that have the highest Lagrange multipliers.

Remark: our scenario problem, (15), corresponds to the *value-at-risk* formulation of the problem (Schildbach et al., 2014), as we want to minimize (under chance constraints) the *maximum* cost-to-go value over the sampled uncertainties (noise trajectories), instead of their *sample average*.

An important property of this approach is that since the optimization step of SMPC is performed periodically and in each step the constraint violation probability is below δ, then the expected time-average of *closed-loop* constraint violations also remains upper bounded by δ, under some mild technical conditions (Schildbach et al., 2014).

5. RENEWABLE ENERGY SYSTEMS

The proposed scenario-based SMPC approach has been applied to control the energy management of the E+grid experimental lighting microgrid (Kovács et al., 2016). The system comprises 191 intelligent LED luminaries that adjust their lighting levels according to the actual traffic conditions, energy generation by roof-mounted PV panels with a total active surface area of 152.5 m^2 and peak power of 21 kWp, as well as battery storage with a capacity of 18.5 kWh. Batteries and bidirectional grid connection make the system an active player in the electricity market: it can charge the batteries with electricity produced during the day or purchased from the grid in valley periods, and then use this energy to operate the luminaries during the night or feed it into the grid in peak periods.

Nevertheless, the operation of the system must be robust against potential power outages. This requirement is captured by chance constraints stating that at any point in time, the batteries must store sufficient energy to cover the net consumption of the following three hours. With this constraint, the objective is to minimize the cost of electricity consumed subject to a given time-of-use electricity tariff (or equivalently, to maximize profit when the system is a net producer and tariff is suitable).

Table 1. Comparing Time-Series Models

Model Type		Performance by Model Order					
		1	2	3	4	5	6
MLP	RMSE	0.534	0.531	0.520	0.540	0.494	0.507
	STD	0.127	0.132	0.167	0.150	0.128	0.160
SVR	RMSE	0.563	0.539	0.519	0.527	0.513	0.527
	STD	0.092	0.135	0.151	0.162	0.135	0.148
BJ	RMSE	0.480	0.481	0.482	0.471	0.461	0.458
	STD	0.126	0.133	0.133	0.154	0.155	0.154

5.1 Uncontrollable Part

Decomposing the uncontrollable elements is straightforward in the case of the particular microgrid we consider, as neither energy production nor energy consumption can be directly controlled by the system; the controller can decide only the amount of energy purchased or sold.

In order to generate *forecasts* for the scenario approach, we model the *energy balance*, which is the difference of the energy production and the energy consumption.

Two Nonlinear AutoRegressive eXogenous (NARX) models, a Multilayer Perceptron (MLP) based and a Support Vector Regression (SVR) based with Gaussian kernels, and a (linear) Box-Jenkins (BJ) model were compared regarding their capacity of predicting the energy balance.

Let ε_t denote the energy balance at time t, where in our case the time step is *one hour*. The system is quasi-periodic with a 24-hour period. Based on historical data, we can compute the *average energy balance* for each hour of the day, denoted by $\{v_t\}$. Note that it is (fully) periodic.

The NARX models of the energy balance take the form

$$\varepsilon_t \;=\; g(\varepsilon_{t-1}, \ldots, \varepsilon_{t-p}, v_t) \;+\; n_t, \quad (21)$$

where n_t is the process noise at time t, and p is called the *order* of the model. Note that g is time-independent, but since it depends on v_t and the noise can be time-varying (see later), the approach is appropriate. The difference between the two models is that in one of them g is realized by an MLP, while we apply SVR for the other one.

In our case, the Box-Jenkins (BJ) model takes the form

$$\varepsilon_t \;=\; F^{-1}(q)B(q)\, v_{t-k_v} + D^{-1}(q)C(q)\, n_t, \quad (22)$$

where B, C, D and F are finite polynomials in q^{-1}, the backward *shift* operator; namely, $q^{-1}\varepsilon_t = \varepsilon_{t-1}$.

In our experiments, we set $k_v = 0$, the degree of C was 0, the degree of B was 1, and the degrees of F and D were set to p, which we henceforth refer to as simply the "order".

Table 1 demonstrates the effectiveness of the three models evaluated by *ten-fold cross validation* on a dataset of 500 points. The average and standard deviation (STD) of the root mean square errors (RMSE) are displayed for various models and model orders on the validation (test) datasets.

For the MLP, one hidden layer was used with 8 neurons having the logsig activation function (and a linear one for the output neuron). For the SVR, ν-SVR was used with Gaussian kernels. It was realized by the LibSVM library [2] with parameters $\epsilon = 0.01$, $\nu = 1$, $c = 1$, and $\gamma = 0.1$.

[2] `https://www.csie.ntu.edu.tw/~cjlin/libsvm/`

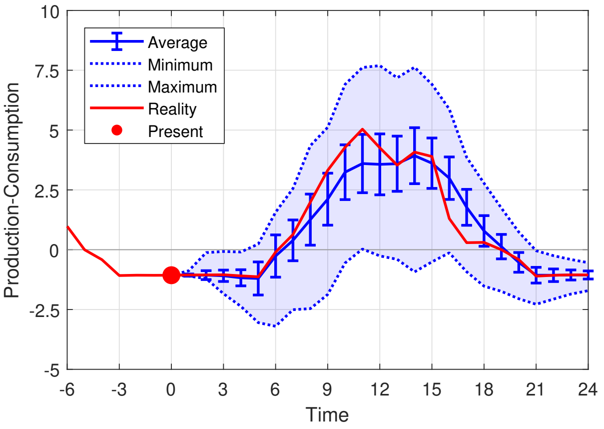

Fig. 1. Generating possible future trajectories for the energy balance with a Box-Jenkins model (noise generation: bootstrap). The average trajectory, with the standard deviation of the forecasts, as well as the area which contains all trajectories are presented.

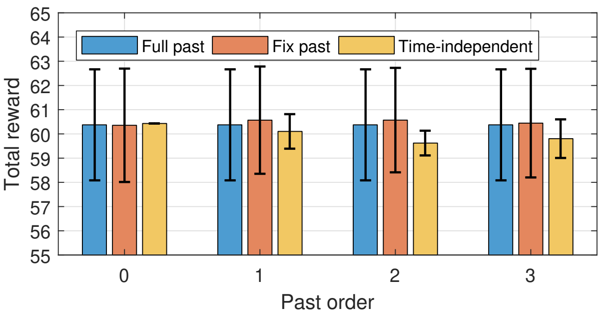

Fig. 2. Comparing controllers with various parametizations, depending on how many past observations are taken into account.

The results indicate that BJ models were the best for this particular problem, though the various models behaved similarly. We used BJ with $p = 1$ as our base model.

Having a model, we can compute (estimate) realizations of the noise terms and build an *empirical distribution function* (EDF) for *each hour of the day*. Then, we can simulate possible future trajectories using the model and by sampling noises from the EDF, often called *bootstrap*. Figure 1 illustrates generating trajectories by this way.

5.2 Controllable Part

In the E+grid system, the main aim of the controller is to trade with the electricity in a cost-effective way, while guaranteeing that the system can still operate for $r = 3$ hours, even in case of a potential power outage.

The cost-to-go function of a policy π is defined as

$$J_n^{\pi}(x_0) \;=\; \sum_{t=1}^{n} \alpha^{t-1}\big(c_t^+ u_t^+ - c_t^- u_t^-\big), \tag{23}$$

where $\alpha \in (0,1)$ is a discount factor (in the experiments, $\alpha = 0.95$), c_t^+ and c_t^- are the costs of buying and selling electricity (in the experiments: $c_t^+ = 1.0$ and $c_t^- = 0.95$), u_t^+ and u_t^- are the amount of bought and sold electricity. They are the positive and the negative components of u_t.

We consider three types of controllers discussed in Section 3: the full affine controller, (11), the controllers with fixed past orders, (12), and the time-independent controllers.

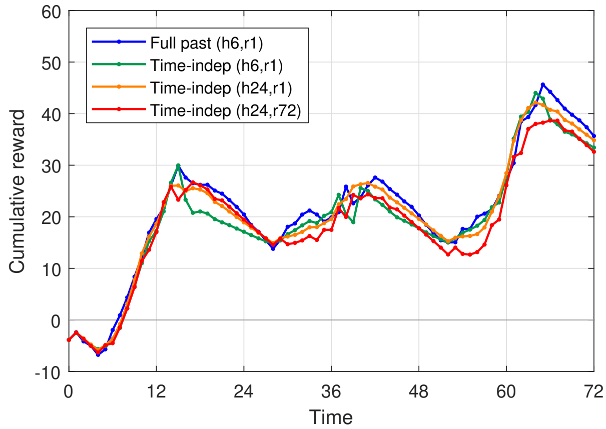

Fig. 3. Cumulative rewards (average of ten runs) of four controllers: a full affine one and three time-independent variants (with past order 2). The notation (hx, ry) encodes that the optimization horizon was x and the reoptimization frequency was y.

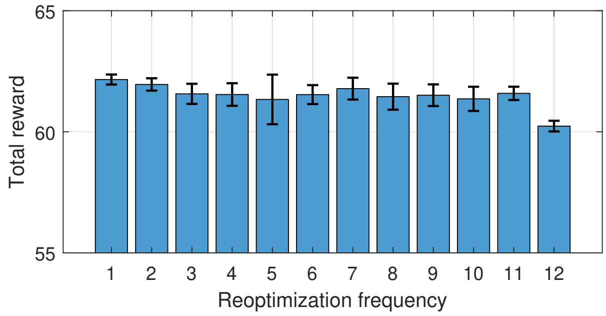

Fig. 4. Comparing various reoptimization frequencies using a (time-independent) affine controller with (fixed) past order 1.

Note that, in our case, these controllers are (potentially time-varying) affine functions of the energy balance, $\{\varepsilon_t\}$.

The uncertainty-dependent constraint sets, with one of the aforementioned controllers (parametrized by θ), are

$$\begin{aligned}
\mathcal{Z}(\varepsilon) \;=\; \Big\{\, (h;\theta) \in \mathbb{R}^{m+1} : \, & J_n^{\pi,\varepsilon}(x_0) \,\le\, h,\; x_0^{\varepsilon} = x_{t_0}^{*}, \\
& u_t^{\varepsilon} = \pi_t(x_{t-1}^{\varepsilon} \mid \theta),\; x_t^{\varepsilon} = x_{t-1}^{\varepsilon} + u_t^{\varepsilon} + \varepsilon_t, \\
& u_t^{\varepsilon} = u_t^+ - u_t^-,\; u_t^+ \ge 0,\; u_t^- \ge 0, \\
& B \ge x_t^{\varepsilon} \ge -\varepsilon_t - \cdots - \varepsilon_{t+r}, \\
& R \ge -u_t^{\varepsilon} - \varepsilon_t \ge -R,\; t = 1,\dots,n \,\Big\},
\end{aligned} \tag{24}$$

where x_t is the state of charge of the battery, ε_t is the energy balance (recall: the difference of energy production and consumption), R is the maximum charge rate of the battery, and B is the maximum capacity of the battery.

Then, we can simulate N i.i.d. “scenarios”, namely, energy balance trajectories (as described in Section 5.1), and solve the resulting scenario problem, (15), which is in this case a standard (finite) linear programming (LP) problem.

Several numerical experiments were initiated to test the SMPC approach. In all experiments, the bound on the constraint violation probability was $\delta = 0.1$, and the confidence probability was $1 - \beta = 0.999$. Ten-fold cross validations were made, hence, the averages as well as the standard deviations of the results are displayed. Each of the ten generated datasets contained 128 real energy balance data, with the corresponding generated trajectories.

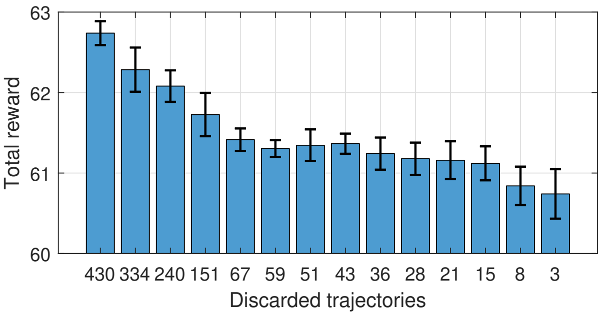

Fig. 5. Total rewards of a time-indep. controller with past order 1, depending on the number of discarded trajectories, out of 1000.

Table 2. Number of Discarded Trajectories vs Constraint Guarantees ($N = 1000$, $d = 3$, $\beta = 0.001$)

430	151	67	43	28	21	15	8	3
50%	80%	90%	93%	95%	96%	97%	98%	99%

As we saw in Section 4, the number of decision variables heavily influences the necessary number of trajectories to guarantee a given chance constraint. Figure 2 presents experiments about controllers with restricted number of parameters. In that experiment, the horizon was $n = 6$ and $N = 500$ trajectories were generated, to help comparing the solutions. It can be observed that the simplest time-independent controller (even with very low past order) achieved similar results to the full affine controller.

One of the advantages of using a feedback controller is that it can be applied for a longer period of time, hence, we may not need to reoptimize the controller in each MPC step. Figure 4 demonstrates, via the example of a (time-independent) controller with past order 1 and horizon $n = 12$, that the efficiency only slightly decreases over time if we keep using the controller for more than one step.

The cumulative rewards (the profit for selling electricity) of various controllers over a 72 hour period are demonstrated by Figure 3. That experiment also studied the effects of various horizon lengths and reoptimization frequencies. The results show that although the full affine controller with reoptimization frequency 1 (which had 54 parameters for a horizon of $n = 6$, and needed 783 trajectories to guarantee the constraints with 90 %) was the best, the much simpler time-independent variants (with past order $p = 2$, having only 4 parameters, which just needed 126 trajectories for a 90 % guarantee) achieved comparable results, surprisingly, also the one which was not reoptimized at all, during the whole experiment.

Finally, Figure 5 presents an experiment in which $N = 1000$ scenarios were generated, but several of them were discarded. The performance over the 128 steps of a time-independent controller with past order 1 are shown. To select which scenarios to discard, we simulated the controller (obtained in the previous optimization step) on all the 1000 trajectories and removed the ones which achieved the lowest rewards. Table 2 overviews the guarantees that the constraints are not violated after discarding various number of scenarios, see formula (20). Note that $\beta = 0.001$ was fixed, and there were only 3 decision variables. Figure 5 demonstrates that we can trade-off the constraint violation probability for achieving higher total rewards.

6. CONCLUSIONS

The paper studied a sampling-and-discarding approach to SMPC for renewable energy systems. After the system was decomposed into controllable and uncontrollable parts, the value-at-risk formulation of the scenario approach for SMPC was described and its theoretical guarantees were surveyed. The ideas were then applied to a public lighting microgrid. Several experiments were presented (e.g., about generating trajectories by bootstrap, the effects of controller parametrizations, reoptimization frequencies and discarding unfavorable scenarios) demonstrating the viability of the approach, even for low order, time-independent controllers that are rarely reoptimized.

REFERENCES

Alamo, T., Tempo, R., Luque, A., and Ramirez, D.R. (2015). Randomized methods for design of uncertain systems: Sample complexity and sequential algorithms. *Automatica*, 52, 160–172.

Calafiore, G.C. and Campi, M.C. (2006). The scenario approach to robust control design. *IEEE Transactions on Automatic Control*, 51(5), 742–753.

Calafiore, G.C. and Fagiano, L. (2012). Robust model predictive control via scenario optimization. *IEEE Transactions on Automatic Control*, 58(1), 219–224.

Campi, M.C. and Garatti, S. (2011). A sampling-and-discarding approach to chance-constrained optimization: feasibility and optimality. *Journal of Optimization Theory and Applications*, 148(2), 257–280.

Cheng, D., Hu, X., and Shen, T. (2010). *Analysis and design of nonlinear control systems*. Springer.

Garatti, S. and Campi, M.C. (2013). Modulating robustness in control design: Principles and algorithms. *IEEE Control Systems*, 33(2), 36–51.

Goulart, P., Kerrigan, E., and Maciejowski, J.M. (2006). Optimization over state feedback policies for robust control with constraints. *Automatica*, 42(4), 523–533.

Kovács, A., Bátai, R., Csáji, B.Cs., Dudás, P., Háy, B., Pedone, G., Révész, T., and Váncza, J. (2016). Intelligent control for energy-positive street lighting. *Energy*, 114, 40–51.

Maciejowski, J.M. (2002). *Predictive Control with Constraints*. Pearson Education.

Mayne, D.Q., Rawlings, J.B., Rao, C.V., and Scokaert, P.O. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, 36(6), 789–814.

Mesbah, A. (2016). Stochastic model predictive control: An overview and perspectives for future research. *IEEE Control Systems*, 36(6), 30–44.

Modarresi, M.S., Xie, L., Campi, M.C., Garatti, S., Carè, A., Thatte, A., and Kumar, P. (2019). Scenario-based economic dispatch with tunable risk levels in high-renewable power systems. *IEEE Transactions on Power Systems*, 34.

Nasir, H.A., Carè, A., and Weyer, E. (2018). A scenario-based stochastic MPC approach for problems with normal and rare operations with an application to rivers. *IEEE Transactions on Control Systems Technology*, 27(4), 1397–1410.

Schildbach, G., Fagiano, L., Frei, C., and Morari, M. (2014). The scenario approach for stochastic model predictive control with bounds on closed-loop constraint violations. *Automatica*, 50(12), 3009–3018.