May 2012 UILU-ENG-12-2204
DC-256

COORDINATING DISPATCH OF DISTRIBUTED ENERGY RESOURCES WITH MODEL PREDICTIVE CONTROL AND Q-LEARNING

Anupama Kowli, Ebony Mayhorn, Karanjit Kalsi, and Sean P. Meyn

Coordinated Science Laboratory
1308 West Main Street, Urbana, IL 61801
University of Illinois at Urbana-Champaign

REPORT DOCUMENTATION PAGE

Form Approved
OMB NO. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	May 2012	

4. TITLE AND SUBTITLE
Coordinating Dispatch of Distributed Energy Resources with Model Predictive Control and Q-learning

5. FUNDING NUMBERS
CPS-0931416 (NSF); DE-OE0000097 and DE-SC0003879 (DoE)

6. AUTHOR(S)
Anupama Kowli, Ebony Mayhorn, Karanjit Kalsi, and Sean P. Meyn

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, 1308 W. Main Street, Urbana, IL 61801-2307

8. PERFORMING RGANIZATION REPORT NUMBER
UILU-ENG-12-2204
DC-256

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Science Foundation, 4201 Wilson Boulevard, Arlington, VA 22230;
U.S. Department of Energy, 1000 Independence Ave. SW, Washington, DC 20585;
Pacific Northwest National Laboratory, 902 Battelle Boulevard, Richland, WA

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Distributed energy resources such as renewable generators (wind, solar), energy storage, and demand response can be used to complement fossil-fueled generators. The uncertainty and variability due to high penetration of renewable resources make power system operations and controls challenging. This work addresses the coordinated operation of these distributed resources to meet economic, reliability, and environmental objectives. Recent research proposes Model Predictive Control (MPC) to solve the problem. However, MPC may yield a poor performance if the terminal penalty function is not chosen correctly. In this work, a parameterized Q-learning algorithm is devised to approximate the optimal terminal penalty function. This approximate penalty function is then used in MPC, thus effectively combining the two techniques. It is argued that this combination approach would lead to the best solution in terms of computation, and adaptability to a changing environment. Simulation studies demonstrating the efficacy of the proposed methodology for power system dispatch problems are presented.

14. SUBJECT TERMS
approximate dynamic programming, distributed energy resources, dynamic dispatch, energy storage, model predictive control, power grid, reinforcement learning, Q-learning

15. NUMBER OF PAGES
6

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Coordinating Dispatch of Distributed Energy Resources with Model Predictive Control and Q-learning

Anupama Kowli, Ebony Mayhorn, Karanjit Kalsi and Sean P. Meyn

Abstract— **Distributed energy resources such as renewable generators (wind, solar), energy storage, and demand response can be used to complement fossil fueled generators. The uncertainty and variability due to high penetration of renewable resources makes power system operations and controls challenging. This work addresses the coordinated operation of these distributed resources to meet economic, reliability and environmental objectives. Recent research proposes Model Predictive Control (MPC) to solve this problem. However, MPC may yield a poor performance if the terminal penalty function is not chosen correctly. In this work, a parameterized Q-learning algorithm is devised to approximate the optimal terminal penalty function. This approximate penalty function is then used in MPC, thus effectively combining the two techniques. It is argued that this combination approach would lead to the best solution in terms of computation, and adaptability to a changing environment. Simulation studies demonstrating the efficacy of the proposed methodology for power system dispatch problems are presented.**

I. Introduction

Environmental concerns have spearheaded the integration of renewable energy sources into power grids all over the world. While the energy derived from the wind, sun and tides is clean with low running costs, it introduces higher variability, greater uncertainty and increased dynamics in the power grid when deployed on a large-scale. Such impacts of increased use of renewable generation can be addressed through better control designs combined with more resources for control. These resources include responsive generators, energy storage, and controllable loads (also known as demand response). The operation of these resources is subject to a range of physical constraints. For example, diesel generators operate under ramping and capacity constraints, while battery storage systems must satisfy complex inter temporal constraints associated with their state of charge. Energy usage in a building system is flexible, but it is also subject to constraints that are not fully understood today. The complexity of the power grid with its diverse set of resources, each with its own dynamical properties and constraints, makes real-time dispatch and control challenging.

Dispatch mechanisms based on model predictive control (MPC) have been proposed (see [1], [2] and references therein). However, MPC may not be effective without careful design. If the terminal penalty function is not chosen correctly, then the performance may be poor, and the closed-loop system may even be unstable. A large prediction horizon could lead to good performance, but this comes at the expense of correspondingly higher computational cost. It is well known that all of these drawbacks are resolved if the terminal penalty function is chosen as the infinite-horizon value function that solves the dynamic-programming equations (see [3] for a survey for deterministic systems, and [4] in the context of stochastic systems).In fact, even a good approximation of the infinite-horizon value function can suffice to ensure good performance with a short prediction horizon [3], [4].

The importance of the terminal penalty function can be deduced based on the close relationship between MPC and value iteration: The MPC algorithm in its standard form defines a state-feedback policy, that is precisely the same policy obtained after T steps of value iteration. It is known that if the initialization of value iteration is a Lyapunov function, then the resulting m-step value function is a Lyapunov function for the $(m+1)^{\text{th}}$ policy obtained from value iteration [4].

In this work, reinforcement learning techniques are used to approximate the infinite-horizon value function. A parameterized Q-learning algorithm is devised to construct an optimal approximation of the value function, within a parameterized class. This approach is favored because it can be applied using real-world data, thereby avoiding the need to adopt artificial assumptions regarding the system dynamics, or the underlying statistics. However, it is argued that a system model is highly valuable to formulate a basis for Q-learning [5]. The Q-learning algorithm introduced here could also be used in control synthesis for the applications of interest. However, the focus of the paper is to show how Q-learning can be coupled with MPC to provide a suitable terminal penalty function, since the Q-function provides an approximation of the infinite-horizon value function. This insight allows us to view MPC and Q-learning algorithms as complementary techniques which can be integrated seamlessly.

The effectiveness of the proposed MPC/Q-learning control architecture is tested on the dispatch problem for a representative power system. The proposed approach is compared against MPC implementations for which the terminal cost is approximated from either the original cost function or the solution to the linear quadratic regulator (LQR) problem. Simulation results indicate that MPC with terminal penalty derived from Q-function provides close-to-optimal solutions

This research is supported by the NSF grant CPS-0931416, the Department of Energy Awards DE-OE0000097 and DE-SC0003879, and the Laboratory Directed Research and Development SEED project "Cooperative Control of Distributed Energy Resources for Grid Support" at the Pacific Northwest National Laboratory.

A. Kowli is with the CSL and the ECE department at University of Illinois, Urbana-Champaign. E. Mayhorn and K. Kalsi are with Pacific Northwest National Laboratory. S. Meyn is with the ECE department at University of Florida, Gainesville.

even with a shorter prediction horizon, unlike typical MPC implementations often provide solutions much different from optimal. The proposed approach is particularly attractive for systems with more uncertainty and/or more constraints, as will be the case for power grids of the future.

The remainder of the report is organized as follows. The parameterized Q-learning algorithm and its integration into MPC framework is presented in Section II. Section III presents an application of the proposed control approach and demonstrates its effectiveness through simulation results. Conclusions and directions of future research are contained in Section IV

II. Q-LEARNING ENHANCED MODEL PREDICTIVE CONTROL

This section contains a review of MPC and Q-learning for the purposes of control, and in particular, for approximate dynamic programming. It is pointed out that the two approaches are complementary. Based on this insight, a Q-learning algorithm is proposed for control of fully observed nonlinear state space models. This algorithm is used to construct an approximation to the *optimal* terminal penalty function for MPC.

A. MPC and the Bellman equation

Consider the nonlinear state-space model described as below:

$$x(t+1) = x(t) + \bar{f}\left(x(t), u(t)\right) \tag{1}$$

where $x(t)$ is the state and $u(t)$ the input, taking values in X and U respectively. Actions may be subject to state-dependent constraints: $\mathsf{U}(x)$ is used to denote the set of control actions that satisfy the state-dependent constraints when the state is $x \in \mathsf{X}$. The "bar" is used for this deterministic discrete-time model, that typically will approximate an MDP (Markov Decision Process) model (notation and motivation borrowed from [5]).

MPC problem: The optimization criterion for the MPC algorithm in the t^{th} time step is to minimize the finite-horizon cost,

$$J(x) = \sum_{\tau=0}^{T-1} c\big(x(t+\tau), u(t+\tau)\big) + c_\bullet\big(x(t+T)\big) \tag{2}$$

where T is the prediction horizon, and x is the state measured at time step t; that is, $x(t) = x$. The one-step cost $c(\cdot,\cdot)$ and terminal penalty cost $c_\bullet(\cdot)$ are non-negative valued.

At each step t, a control sequence $\big\{u(t), u(t+1), \ldots, u(t+T-1)\big\}$ is found so that the cost (2) is minimized, subject to system dynamics and state/action constraints. The first element $u^*(t)$ of the minimizing control sequence is implemented at the current time step t, and the algorithm proceeds to next time step. This procedure defines a state-feedback control law. It is stabilizing under general conditions on the terminal cost $c_\bullet$. In fact, as we shall review next, the computational complexity of MPC can be reduced by choosing a smaller prediction horizon T if the terminal penalty $c_\bullet$ is appropriately chosen. Stability is also guaranteed under mild conditions on $c_\bullet$; see [3] for a survey.

Value function: The value function of the infinite-horizon control problem is defined as a minimum, similar to the minimum appearing in MPC:

$$V^*(x) = \min_{\boldsymbol{u}} \sum_{t=0}^{\infty} c\left(x(t), u(t)\right) \ , \ x(0) = x \in \mathsf{X}, \tag{3}$$

where $\boldsymbol{u} = \{u(0), u(1), \ldots\}$ denotes the sample path of control actions. It is assumed that this function is finite valued on X.

The corresponding dynamic programming (DP) equation is given by,

$$V^*(x) = \min_{u \in \mathsf{U}(x)} \left\{ c(x,u) + \mathcal{K}V^*\ (x,u) \right\} , \tag{4}$$

where the DP operator $\mathcal{K}$ is defined as in MDP theory. That is, for any function $h\colon \mathsf{X} \to \mathbb{R}$, $\mathcal{K}h$ denotes the function on $\mathsf{X} \times \mathsf{U}$ given by

$$\mathcal{K}h\,(x,u) = h(x + \bar{f}(x,u)) \,, \ x \in \mathsf{X}, u \in \mathsf{U}. \tag{5}$$

The DP equation can be extended to any finite time-horizon,

$$V^*(x) = \min_{\boldsymbol{u}_0^{T-1}} \Big(\sum_{t=0}^{T-1} c(x(t), u(t)) + V^*(x(T)) \Big) . \tag{6}$$

In this way the relationship with MPC is evident: If V^* is chosen as the terminal penalty function $c_\bullet$, then the MPC algorithm is infinite-horizon optimal, and this is true for *any* time-horizon $T \geq 1$. For these reasons, V^* is a perfect candidate for the terminal penalty function in the MPC algorithm. Of course, computation of the value function is very difficult in most cases.

B. Approximate Value Function from Q-Learning

Approximations of a value function can be obtained using reinforcement learning techniques such as TD- or Q-learning [6]. This approach is illustrated here, in the setting of [5], [7], [8] wherein an idealized model is used to inform the construction of a basis.

The *Q-function* used in Q-learning is a real-valued function defined on $\mathsf{X} \times \mathsf{U}$. It is closely related to the Hamiltonian in optimal control theory [7]. It is defined as the function appearing in the brackets in the DP equation (4):

$$H^*(x,u) := c(x,u) + \mathcal{K}V^*\,(x,u) \,. \tag{7}$$

On denoting

$$\underline{H}^*(x) = \min_{u \in \mathsf{U}(x)} H^*(x,u) \,, \tag{8}$$

the DP equation implies that $\underline{H}^* = V^*$. The reason for introducing the new notation is that the DP equation can be transformed to define a fixed point equation in H^*:

$$H^*(x,u) = c(x,u) + \mathcal{K}\underline{H}^*\,(x,u) \,. \tag{9}$$

In this form, it is not difficult to devise algorithms to approximate H^* and thence V^*. A new approach is described in the next paragraphs.

Approximation architecture: A natural parameterization for the approximation of the Q-function defined in (9) is of the form

$$H^\theta(x,u) = c(x,u) + \theta^T \psi(x,u) \tag{10}$$

where $\psi : \mathsf{X} \times \mathsf{U} \to \mathbb{R}^d$. Given a basis $\{\varphi_i : 1 \le i \le d\}$ intended to approximate V^*, a basis for Q-learning may be chosen as the functions on $\mathsf{X} \times \mathsf{U}$,

$$\psi_i(x,u) = \mathcal{K}\varphi_i(x,u) = \varphi_i\left(x + \bar{f}(x,u)\right),\ 1 \le i \le d. \tag{11}$$

In prior work, it is found that idealized models (e.g., fluid, diffusion, or mean-field games) can be used to obtain good bases [5], [7], [8].

Given a parameterized family of functions $\{H^\theta\}$ as defined in (10), the goal of Q-learning is to compute a parameter $\theta \in \mathbb{R}^d$ so that $H^\theta \approx H^*$. The approximation is with respect to a specific norm.

Ergodic environment for learning: In the usual Q-learning algorithm for MDPs, a randomized stationary policy is applied to allow sufficient sampling of the state-action space [9]. Similar assumptions are adopted here. It is assumed that control actions are chosen so that the joint process $(\boldsymbol{x}, \boldsymbol{u})$ is ergodic in some suitable sense, with stationary realization denoted $(\boldsymbol{X}, \boldsymbol{U})$. This can be achieved by perturbing a stabilizing state-feedback policy ϕ with an excitation signal ζ as follows,

$$u(t) = \phi\big(x(t)\big) + \zeta(t)\,. \tag{12}$$

In the approach described here, the excitation signal is obtained through a quasi-Monte-Carlo approach, following [10]:

$$\zeta(t) = \sum_{i=1}^{n} A_i \sin(\omega_i t)\,,$$

where $\{A_i\}$ are constants, $\{\omega_i\}$ are various frequencies, and $n \ge 1$ is an integer. We assume that the resulting controlled system admits a stationary realization, whose marginal distribution is denoted ϖ.

A Hilbert space setting is adopted for approximation, based on the corresponding ergodic norm: For measurable functions $F, G : \mathsf{X} \times \mathsf{U} \to \mathbb{R}$, the inner product and norm are defined as follows:

$$\begin{aligned}\langle F, G\rangle &:= \int F(x,u)G(x,u)\varpi(dx,du)\,,\\ \|F\|^2 &:= \int F^2(x,u)\varpi(dx,du)\,.\end{aligned}$$

In terms of the stationary realization $(\boldsymbol{X}, \boldsymbol{U})$,

$$\langle F, G\rangle = \mathsf{E}[F(X(t),U(t))G(X(t),U(t))]\,,$$

where the expectation is independent of time. Under general conditions on the system and input (12), the Law of Large Numbers holds and the expectation may be computed from a sample trajectory on the $\mathsf{X} \times \mathsf{U}$ space.

Error criterion: A natural criterion for choosing parameter θ is to minimize the actual error $\|H^* - H^\theta\|$, which is the viewpoint taken in TD-learning. Alternatively, error in the fixed point equation (9) may be minimized, which is known as the *Bellman error*.

The mean-square Bellman error is defined as,

$$\begin{aligned}\mathcal{E}(\theta) &:= \tfrac{1}{2}\left\|H^\theta - \big(c + \mathcal{K}\underline{H}^\theta\big)\right\|^2\\ &= \tfrac{1}{2}\,\mathsf{E}\Big[\Big(H^\theta\big(X(t),U(t)\big)\\ &\qquad - \big[c\big(X(t),U(t)\big) + \underline{H}^\theta\big(X(t+1)\big)\big]\Big)^2\Big]\,.\end{aligned} \tag{13}$$

where the function $\underline{H}^\theta \colon \mathsf{X} \to \mathbb{R}$ is defined as in (8),

$$\underline{H}^\theta(x) = \min_{u \in \mathsf{U}(x)} H^\theta(x,u)\,. \tag{14}$$

Note that if $\mathcal{E}(\theta^*) = 0$, then the fixed point equation (9) holds in a mean-square sense. Consequently, the DP equation (4) is solved a.e. $[\varpi]$.

The goal of Q-learning is to find $\theta^* \in \mathbb{R}^d$ that minimizes $\mathcal{E}$ over all $\theta \in \mathbb{R}^d$. The optimal parameter can be computed using a stochastic-approximation of steepest descent. This requires that we obtain a suitable expression for the gradient. First, recall that the function $H^\theta(x,u)$ is affine in θ with gradient $\psi(x,u)$. If $u^*_{x,\theta}$ denotes the minimizer in (14), then

$$\nabla \underline{H}^\theta(x) := \underline{\psi}^\theta(x) = \psi(x, u^*_{x,\theta})\,.$$

Using this notation, the gradient for the steepest descent algorithm is expressed,

$$\begin{aligned}\nabla\mathcal{E}(\theta) &= \langle H^\theta - \big(c + \mathcal{K}\underline{H}^\theta\big), \psi - \mathcal{K}\underline{\psi}^\theta\rangle\\ &= \langle \theta^T\psi - \mathcal{K}\underline{H}^\theta, \psi - \mathcal{K}\underline{\psi}^\theta\rangle\end{aligned} \tag{15}$$

under the assumption that the derivative and expectation can be exchanged.

This leads to the following stochastic approximation algorithm to recursively estimate θ^*,

$$\begin{aligned}\theta(t+1) = \theta(t) - \gamma_t\Big[\theta^T(t)\psi\big(x(t),u(t)\big) - \underline{H}^\theta\big(x(t+1)\big)\Big]\\ \times\Big[\psi\big(x(t),u(t)\big) - \underline{\psi}^\theta\big(x(t+1)\big)\Big]\,.\end{aligned} \tag{16}$$

where γ_t is a decreasing gain sequence, such as $1/(t+1)$ [10], [11].

C. MPC with Q-learning

Q-learning gives an approximation $H^{\theta^*}(x,u)$ to the Q-function $H(x,u)$ defined in (7) and, consequently, $\underline{H}^{\theta^*}(x)$ approximates the value function $V^*(x)$ defined in (3). Then, MPC can be solved with this modified objective function:

$$J(x) = \sum_{\tau=0}^{T-1} c\big(x(t+\tau), u(t+\tau)\big) + \underline{H}^{\theta^*}\big(x(t+T)\big)\,. \tag{17}$$

By plugging in the approximate value function from Q-learning into the MPC framework, the benefits of the two approaches can be combined. A better approximation of the Q-function can lead to stability, improved performance and smaller prediction horizon.

III. Power System Dispatch

The effectiveness of the proposed Q-learning/MPC control architecture was tested on a problem of dynamic scheduling of power system resources: The results are surveyed in this section. First, a brief overview of the classical dispatch problem is provided followed by the problem considered in this work, which is a variant of the control problem considered in [2], with load and wind data taken from [12], [13]. The section concludes with simulation results providing a comparison of several MPC architectures.

A. Overview: Economic Dispatch

The economic dispatch problem determines the optimal deployment of resources to meet predicted load demand over a specified scheduling period, while satisfying physical constraints and minimizing cost of operations. The operational costs include total fuel costs of generation, costs associated with storage, and the cost of ramping generation.

It is typically formulated as a minimization problem with the objective being the operational costs calculated across the scheduling horizon consisting of T^{sch} time steps. The problem explicitly considers capacity constraints and ramping limitations on the resources.

The dispatch problem is typically solved in "open-loop": The optimization problem is solved once over an entire horizon T^{sch} – say, one day. This approach may provide satisfactory performance when the main source of uncertainty is the load forecast error, which is typically small. Open loop control is not suitable in a highly uncertain environment. In such systems, MPC based dispatch mechanisms may work better [1], [2], but these approaches must be implemented with care, as we have surveyed in the previous section.

B. Test System Description

A small representative power system is considered, consisting of a diesel generator, a battery energy storage system (BESS), a wind power plant, and a mix of loads which constitute the total demand. Many factors are disregarded, such as system losses, and the details of dynamics and costs. The goal here is control synthesis, for which a simplified model if frequently justifiable.

It is assumed that the BESS is used to compensate for net load (total load minus wind generation) variability and that its charging/discharging is determined based on a threshold policy: The BESS is charged if the net load is less than the threshold, and discharged if it is less than the threshold. The threshold value can be viewed as power demanded by the net load and BESS; it is supplied by the diesel generator.

A balancing service term is introduced to manage real-time mismatch in supply and demand caused by the uncertainty associated with wind generation and load. It is assumed that the balancing service is procured from an expensive ancillary service resource, whose operation is independent of the other resources in the system. This resource can be thought of either as the system's interaction with the rest of the grid or as an ancillary generation source/load sink which is run only to manage the shortfalls/surpluses in system generation.

A quadratic cost structure is adopted for the diesel generator's fuel costs. The BESS operational costs are cast as proxy costs which penalize deviations of its state of charge (SOC) from a specified reference value.

C. Problem Formulation

The problem formulation is adapted from [2]. Static models are used to represent the states of the system and dynamics are introduced by set point changes in the generator's outputs and threshold values of BESS.

The output of the diesel generator, the threshold of BESS, its SOC, the output of the wind plant, the total load and the required balancing service at time t are denoted by $P_{\mathsf{G}}(t)$, $P_{\mathsf{thr}}(t)$, $\xi_{\mathsf{S}}(t)$, $P_{\mathsf{W}}(t)$, $P_{\mathsf{D}}(t)$ and $P_{\mathsf{bal}}(t)$, respectively. The power supplied by the BESS is

$$P_{\mathsf{S}}(t) = P_{\mathsf{D}}(t) - P_{\mathsf{W}}(t) - P_{\mathsf{thr}}(t)\,.$$

where $P_{\mathsf{S}}(t) > 0$ indicates discharged and $P_{\mathsf{S}}(t) < 0$ indicates charging. For the balancing service, $P_{\mathsf{bal}}(t) > 0$ indicates excess generation while $P_{\mathsf{bal}}(t) < 0$ indicates a generation deficit.

System dynamics: The generation outputs, BESS thresholds and SOC as well as the balancing service are considered to be the states of the system, and the control actions are the set-point changes in generation output and BESS threshold denoted by $\Delta P_{\mathsf{G}}(t)$ and $\Delta P_{\mathsf{thr}}(t)$. That is,

$$X(t) = [P_{\mathsf{G}}(t), P_{\mathsf{thr}}(t), \xi_{\mathsf{S}}(t), P_{\mathsf{bal}}(t)]^T$$

and

$$U(t) = [\Delta P_{\mathsf{G}}(t), \Delta P_{\mathsf{thr}}(t)]^T\,.$$

The states and actions are constrained so that

$$X^{\mathsf{min}} \le X(t) \le X^{\mathsf{max}} \text{ and } \mathsf{U}^{\mathsf{min}} \le U(t) \le U^{\mathsf{max}} \quad (18)$$

for each t. The limits X^{min} and X^{max} determined by the capacity bounds on the states, while the limits U^{min} and U^{max} depend on the ramping limitations associated with the set-point changes.

Dynamics are introduced by the changing set-points and take the form:

$$\begin{aligned}
P_{\mathsf{G}}(t+1) &= P_{\mathsf{G}}(t) + \Delta P_{\mathsf{G}}(t)\,,\\
P_{\mathsf{thr}}(t+1) &= P_{\mathsf{thr}}(t) + \Delta P_{\mathsf{thr}}(t)\,,\\
\xi_{\mathsf{S}}(t+1) &= \xi_{\mathsf{S}}(t) - \alpha_{\mathsf{S}}\big(P_{\mathsf{D}}(t) - P_{\mathsf{W}}(t) - P_{\mathsf{thr}}(t)\big) \quad (19)\\
P_{\mathsf{bal}}(t+1) &= P_{\mathsf{G}}(t+1) - P_{\mathsf{thr}}(t+1)\\
&= P_{\mathsf{G}}(t) - P_{\mathsf{thr}}(t) + \Delta P_{\mathsf{G}}(t) - \Delta P_{\mathsf{thr}}(t)
\end{aligned}$$

where $P_{\mathsf{W}}(t)$ and $P_{\mathsf{D}}(t)$ are interpreted as disturbances in the state dynamics, The parameter α_{S} represents the conversion factor,

$$\alpha_{\mathsf{S}} = \frac{\eta_{\mathsf{S}}}{E_{\mathsf{S}}^{\mathsf{max}}}\Delta t\,;$$

where η_{S} and $E_{\mathsf{S}}^{\mathsf{max}}$ represent the efficiency and energy capacity of the storage device, and Δt represents time step duration in hours.

The dynamics in (19) can be cast in a linear form

$$X(t+1) = AX(t) + BU(t) + MV(t) \quad (20)$$

where $V(t) = \big[P_{\mathsf{W}}(t), P_{\mathsf{D}}(t)\big]^{\intercal}$ is the disturbance process.

Cost Function: The dispatch problem is set-up to minimize the fuel costs of generators, operational costs of BESS, balancing service needed, and the mechanical wear and tear on generators caused by ramping. A cost function is formulated to take into account these diverse costs. The cost at time t is taken to be the weighted sum,

$$\begin{aligned} c\big(X(t),U(t)\big) &= w_1\big(\, aP_{\mathsf{G}}^2(t) + bP_{\mathsf{G}}(t) + c\,\big) \\ &\quad + w_2\big(\,\xi_{\mathsf{S}}(t) - \xi_{\mathsf{S}}^{\mathsf{ref}}\,\big)^2 + w_3 P_{\mathsf{bal}}^2(t)\,, \\ &\quad + w_4 \Delta P_{\mathsf{G}}^2(t) + w_5 \Delta P_{\mathsf{thr}}^2(t) \end{aligned} \tag{21}$$

where the weight w_i determines the relative importance of the i^{th}objective and $\sum_i w_i = 1$. The cost can reformulated in a quadratic form,

$$c\big(x,u\big) = (x - x^{\mathsf{ref}})^{\intercal} Q (x - x^{\mathsf{ref}}) + u^{\intercal} R u + \kappa\,, \tag{22}$$

where x^{ref} is a reference state and κ is a constant.

MPC set-up: A predictive model for the system dynamics is defined as follows

$$\hat{x}(t+1) = A\hat{x}(t) + B\hat{u}(t) + M\hat{v}(t)\,. \tag{23}$$

At each step t, the actual values measured from the system are used to initialize these dynamics. Predictions for the noise are made for the horizon, $\boldsymbol{v}_1^{T-1} := \{\hat{v}(t+1), \ldots, \hat{v}(t+T-1)\}$. An autoregressive integrated moving average model is used to predict the wind generation and a seasonal ARIMA model is used to forecast aggregate load.

Given noise predictions $\boldsymbol{v}_1^{T-1}$, the MPC algorithm chooses the control trajectory to minimize the predicted costs defined analogous to (2) subject to the dynamics in (23) and constraints in (18). Then, the control $U(t)$ is chosen as $\hat{u}^*(t)$ and applied to the system. In the simulation experiments, the system evolves according to dynamics in (20), so that $X(t+1)$ is defined, and the procedure is repeated to obtain $\hat{u}^*(t+1)$. More details on the MPC set-up and the forecasting techniques used are available in [2].

Q-learning set-up: The Q-learning algorithm devised in Section II-B is for a deterministic system. Although, the system under consideration has stochasticity introduced by wind generation and load demand, the algorithm of Section II-B was applied by adopting a deterministic mean-field model. In the design of the Q-learning algorithm, the dynamics are assumed to follow the recursion

$$\bar{x}(t+1) = A\bar{x}(t) + B\bar{u}(t) + M\bar{v} \tag{24}$$

where $\bar{v}$ is the mean of the disturbance process. The DP operator $\mathcal{K}$ is defined based on (24).

The basis was obtained using the structure (11), with $\{\varphi_i : 1 \le i \le d\}$ obtained using $d = 3$ as follows. The basis function φ_1 is taken to be the value function obtained from the LQR problem without state constraints (subject to the above dynamics and costs given in (22)). The other two basis functions, φ_2 and φ_3, are designed to penalize movement of the state trajectory towards the generation boundary and the SOC limits, respectively.

The Q-learning algorithm can be implemented with historical data of the wind generation and load demand. However, in the simulation results presented next, the state trajectory is computed using the mean-field model (24).

D. Simulation Results

In the simulations described here, the test system is considered to have either 5 or 3 MW diesel generation and 3.6 MWh BESS, with a tie to the grid to procure the balancing service. The wind plant data is obtained from [12] and the aggregate load data is generated using [13].

Three implementations of MPC are considered:

1) MPC with $c_\bullet(x) = (x - x^{\mathsf{ref}})^{\intercal} Q (x - x^{\mathsf{ref}})$: This is considered as the base case.
2) MPC with $c_\bullet(x) = (x - x^{\mathsf{ref}})^{\intercal} S^{\mathsf{LQR}} (x - x^{\mathsf{ref}})$: The matrix S^{LQR} is obtained from the closed form solution of the LQR problem with dynamics and costs given in (23) and (22) (and unconstrained state-action space).
3) MPC with $c_\bullet(x) = \underline{H}^{\theta^*}(x)$: The Q-learning algorithm is run to find H^{θ^*}.

The simulation studies were preformed with control steps of 10 minutes and a scheduling horizon of 24 hours, so that $T^{\mathsf{sch}} = 144$. The metric used for comparative analysis is the total normalized cost for the scheduling period, given as

$$J^*_{\mathsf{tot}} = \sum_{t=1}^{T^{\mathsf{sch}}} c\big(x^*(t), u^*(t)\big)$$

where $x^*(t)$ and $u^*(t)$ are based on the MPC state-feedback policy. The performance of each algorithm is compared in terms of J^*_{tot} for various values of the prediction horizon T.

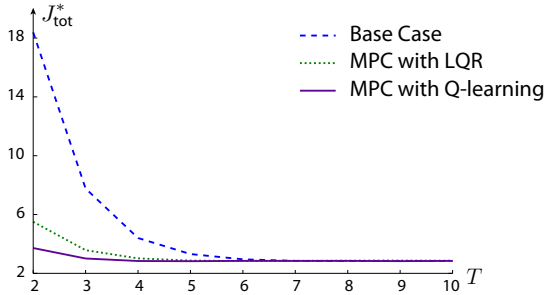

Fig. 1. Total costs as a function of the prediction horizon for $P_{\mathsf{G}}^{\mathsf{max}} = 5$ MW and low wind generation.

Fig. 1 shows the total cost for each MPC implementation, with $P_{\mathsf{G}}^{\mathsf{max}} = 5$ MW. The MPC algorithm with $c_\bullet(x) = \underline{H}^{\theta^*}$ shows much better performance when compared with the base-case, for all $T \le 5$, with a 5-fold improvement for $T = 2$. The performance of LQR-based MPC is similar to that of the MPC/Q-learning combination. A possible reason for this could be that for the given noise sample path and constraints, the state trajectory does not hit the boundaries of the state space.

A better test of the effectiveness of LQR-based MPC requires more stringent constraints and larger disturbances.

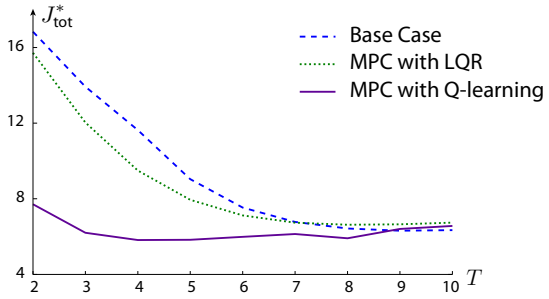

Fig. 2. Total costs as a function of the prediction horizon for $P_{\mathsf{G}}^{\max} = 3$ MW, generation ramping limit of 1 MW and high wind generation.

To compare the performance for such a noisy, constrained system, the simulations were rerun with $P_{\mathsf{G}}^{\max} = 3$ MW, a ramping constraint of $\Delta P_{\mathsf{G}}^{\max} = 1$ MW, and wind generation profile multiplied by a factor of 2. The total cost J^*_{tot} for the three MPC implementations is plotted as a function of the prediction horizon T in Fig. 2. The plot demonstrates how the MPC/Q-learning control architecture provides a better performance even shorter prediction horizons. This study demonstrates how combining the MPC and Q-learning techniques not only improves the performance of the control policy but also enhances the tolerance of the controlled system to uncertainty and constraints.

IV. Conclusion

The contributions of this paper are three-fold. First, a parameterized Q-learning algorithm is devised for the control of fully observed nonlinear state space models. Second, the Q-learning algorithm is used to construct an approximation to the *optimal* terminal penalty function for MPC: in this way, the two control approaches are coupled, and the benefits of each approach are combined. Third, an application of the proposed approach to the power system dispatch problem is presented to showcase improved performance of MPC/Q-learning control architecture.

Simulation results indicate that the proposed coupling is particularly effective for systems with large disturbances and heavily constrained state-action space. Future power grids with a deep penetration of renewable resources and limited flexibility in generation, demand response and storage, may very well be examples of such systems. We provide here an effective control mechanism for dispatching resources in such grids. The improvement in performance and greater adaptability of Q-learning based MPC may be of particular importance in large grids with many resources and many sources of uncertainty.

There are many avenues open for future research; a few are listed below:

- The selection of the right basis functions for the Q-learning algorithm can be investigated further. Fluid and diffusion models provide a starting point [5]. Basis selection for large scale, interconnected power systems can be facilitated through model reduction in more complex networked settings [8].
- The inclusion of statistical information may be used to improve the Q-learning algorithm.
- Error bounds on the Bellman error can be used to obtain performance bounds for both Q-learning and the MPC algorithm introduced here.
- Implementation in a non-time-homogeneous environment will require modifications to the proposed approach.

References

[1] L. Xie and M. Ilic, "Model predictive economic/environmental dispatch of power systems with intermittent resources," in *PES '09. IEEE Power Energy Society General Meeting*, July 2009, pp. 1 –6.

[2] E. Mayhorn, K. Kalsi, M. A. Elizondo, W. Zhang, S. Lu, N. Samaan, and K. Butler-Purry, "Optimal control of distributed energy resources using model predictive control," in *Proceedings of the 2012 IEEE PES General Meeting*, July 2012, pp. 1–8.

[3] D. Mayne, J. Rawlings, C. Rao, and P. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[4] R.-R. Chen and S. P. Meyn, "Value iteration and optimization of multiclass queueing networks," vol. 32, no. 1-3, pp. 65–97, 1999.

[5] D. Huang, W. Chen, P. Mehta, S. Meyn, and A. Surana, "Feature selection for neuro-dynamic programming," in *Reinforcement Learning and Approximate Dynamic Programming for Feedback Control*, F. Lewis, Ed. Wiley, 2011.

[6] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*, on-line edition at http://www.cs.ualberta.ca/~sutton/book/the-book.html ed. Cambridge, MA: MIT Press, 1998.

[7] P. G. Mehta and S. P. Meyn, "Q-learning and Pontryagin's minimum principle," Dec. 2009, pp. 3598–3605.

[8] S. P. Meyn, *Control Techniques for Complex Networks*. Cambridge: Cambridge University Press, 2007, pre-publication edition available online.

[9] D. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*. Cambridge, Mass: Atena Scientific, 1996.

[10] D. Shirodkar and S. Meyn, "Quasi stochastic approximation," July 2011, pp. 2429–2435.

[11] V. S. Borkar, *Stochastic Approximation: A Dynamical Systems Viewpoint*. Delhi, India and Cambridge, UK: Hindustan Book Agency and Cambridge University Press (jointly), 2008.

[12] "Wind systems integration: Data resources," http://www.nrel.gov/wind/systemsintegration/data_resources.html.

[13] "GridLAB-D residential module user's guild," http://sourceforge.net/apps/mediawiki/gridlab-d/index.php?title=Residential_module_user%27s_guide.