Dual MPC with Reinforcement Learning

Juan E. Morinelly∗, B. Erik Ydstie∗

∗Chemical Engineering Department, Carnegie Mellon University,
Pittsburgh, PA 15213 USA
(e-mail: {jmorinelly, ydstie}@andrew.cmu.edu)

Abstract: An adaptive optimal control algorithm for systems with uncertain dynamics is formulated under a Reinforcement Learning framework. An embedded exploratory component is included explicitly in the objective function of an output feedback receding horizon Model Predictive Control problem. The optimization is formulated as a Quadratically Constrained Quadratic Program and it is solved to ε-global optimality. The iterative interaction between the action specified by the optimal solution and the approximation of cost functions balances the exploitation of current knowledge and the need for exploration. The proposed method is shown to converge to the optimal policy for a controllable discrete time linear plant with unknown output parameters.

Keywords: Adaptive control, dual control, optimal control, model predictive control, reinforcement learning, approximate dynamic programming.

1. INTRODUCTION

Approximate Dynamic Programming (ADP) and Reinforcement Learning (RL) methods have been an increasingly active research area in many engineering disciplines (Lewis and Liu (2013)). These methods provide a general structure for feedback control of complex, uncertain systems. Sutton and Barto (1998) define the theoretical development of RL under three characteristics of the problem to be solved: the closed-loop nature of the interaction between an agent and its environment, an agent that must discover which actions yield the most rewards, and a performance evaluation that considers extended periods. These aspects are closely related to the features of adaptive dual controllers as described by Wittenmark (1995).

Bertsekas and Tsitsiklis (1995) coined the term Neuro-Dynamic Programming to refer to ADP and RL methods. This is due to the fact that these methods are usually applied through the use of neural networks as approximating structures to derive control policies of nonlinear systems. Similarly, other names have been proposed as Actor-Critic RL Feedback Control by Lewis et al. (2012) or as Action Dependent Heuristic Dynamic Programming by Ferrari and Stengel (2004) to specify a feature of their proposed method. Lee and Lee (2009) propose an ADP method with exploration provided by stochastic simulations with defined suboptimal policies. The underlying theoretical basis is common to all methods and rests on the original work by Bellman (1952) approximated by a sequential learning process that combines search and long-term memory (Barto and Dietterich (2004)).

This study formulates a generalization of the dual Model Predictive Control (MPC) algorithm by Heirung et al. (2015) by introducing the effect of anticipated information under the ADP/RL framework. The algorithm is developed for the simple case of regulation of a discrete linear system with uncertain dynamics. In the RL context, the agent can be viewed as the MPC algorithm itself, which computes actions adaptively by solving an optimal control problem defined by the environment’s response in the form of output measurements.

2. MATHEMATICAL BACKGROUND

The content of this section follows the development by Bertsekas (2012) where a detailed description can be found. Consider the controllable and observable discrete-time SISO system

\[ \varphi_{t+1} = f(\varphi_t, u_t) = A\varphi_t + bu_t \]

\[ y_t = g(\varphi_t) = \theta^T \varphi_t + v_t \tag{1} \]

where the state \( \varphi_t \in \mathbb{R}^n \), the admissible control \( u_t \in \mathbb{R} \) only takes values dependent of \( \varphi_t \) (\( u_t \in U(\varphi_t) \subset \mathbb{R}, \forall \varphi_t \in \mathbb{R}^n \)), and the output \( y_t \in \mathbb{R} \) is given by \( \theta^T \varphi_t \) plus a noise sequence \( v_t \sim \mathcal{N}(0, \sigma^2) \). For the case where \( A, b, \) and \( \theta \) are deterministic, consider an agent tasked to bring the output in (1) to zero with cautious inputs. A stage cost function for a given time, \( t \), is defined in terms of the current state, \( \varphi_t \), and control action, \( u_t \):

\[ c(\varphi_t, u_t) = \varphi_t^T \theta \theta^T \varphi_t + ru_t^2 \geq 0 \tag{2} \]

where \( r > 0 \) dictates the caution level. This stage cost is a nonnegative function. Equations (1) and (2) define a Markov Decision Process (MDP) with continuous state and action spaces in which an agent perceives the environment’s response to an action in the form of a succeeding state and a stage cost. The response depends on the current state-control pair only, which is known as the Markov property. MDPs are generally defined by the agent-environment interaction under the Markov property. The state transition dynamics and the costs are not necessarily known or deterministic (Mitchell (1997)).

A policy is a function \( \pi = \mu_0, \mu_1, ... \) with \( \mu_k : \mathbb{R}^n \rightarrow U \) that belongs to the set of all admissible policies, \( \Pi \). If \( \pi = \mu, \mu, ... \) then the policy is said to be time-invariant.
or stationary. The infinite horizon total cost or cost-to-go function, \( J_\pi(\varphi) \), is defined for any state \( \varphi \in \mathbb{R}^n \), a stage cost function \( c \), and a fixed policy \( \pi \) as the limit of the expected value for the discounted accumulation of all future stage costs subject to the system dynamics.

\[
J_\pi(\varphi_t) := \lim_{N \to \infty} \mathbb{E} \left[ \sum_{k=0}^{N-1} \alpha^k c(\varphi_{t+k}, \mu_k(\varphi_{t+k})) \right]
\]

(3)

Where \( \alpha \) is a discount factor \((0 < \alpha \leq 1)\). \( J_\pi(\varphi_t) \) indicates the cost-to-going from state \( \varphi_t \) for a stationary policy \( \mu \). In the context of RL, equation (3) provides the agent a decision criterion to learn an optimal policy, \( \pi^*(\varphi) \in \Pi \) that minimizes the cost-to-go function.

\[
\pi^*(\varphi_t) := \arg \min_{\pi \in \Pi} J_\pi(\varphi_t)
\]

(4)

Controllability of \( \{A, b\} \) guarantees the existence of a stationary policy that stabilizes the system and minimizes (3). The problem we consider is the design of an algorithm that adaptively learns and implements policies forward in time for a system with unknown output parameter vector, \( \theta \). A desired property of the algorithm is that, as experience from the feedback interaction accumulates, the learning process converges to the optimal policy.

### 2.1 Bellman’s Equation

The cost-to-go function for a stationary policy \( \mu \) at the current state \( \varphi_t \) can be expressed recursively by extracting the first \( N \) contributions, the \( N \)-step cost. For a deterministic stage cost, such as (2), equation (3) can be rewritten in terms of the one-step cost as:

\[
J_\mu(\varphi_t) = c(\varphi_t, \mu(\varphi_t)) + \alpha J_\mu(\varphi_{t+1})
\]

(5)

in general, the expression for the \( N \)-step is given by

\[
J_\mu(\varphi_t) = J_{\mu,N}(\varphi_t) + \alpha J_\mu(\varphi_{t+N})
\]

(6)

where

\[
J_{\mu,N}(\varphi_t) := \sum_{k=0}^{N-1} \alpha^k c(\varphi_{t+k}, \mu(\varphi_{t+k})).
\]

These backward recursions define a consistency path followed by the state transitions under the specified stationary policy. The solution to (5) or (6) yields the value of the infinite sum for a given stationary policy, \( \mu \), in (3). Furthermore, for \( c(\varphi, u) \geq 0 \), the optimal cost-to-go satisfies:

\[
J^*(\varphi_t) = \min_{u \in U(\varphi)} \left[ c(\varphi_t, u) + \alpha J^*(\varphi_{t+1}) \right]
\]

(7)

This is Bellman’s equation and holds for any \( \varphi_t \in \mathbb{R}^n \). An optimal policy has the property that, regardless of previous actions that lead to \( \varphi_t \), the controls specified by it are optimal from then on. For the discounted case with bounded cost per stage, a function that satisfies this backward recursion is the unique solution to (7).

An agent with perfect knowledge of \( J^* \), \( f \), \( g \), and \( c \) can compute optimal control actions, forward in time, given by the argument that solves the optimization problem posed by (7). A necessary and sufficient condition for a stationary policy \( \mu \) to be optimal is

\[
J^*(\varphi) = c(\varphi, \mu(\varphi)) + \alpha J^*(f(\varphi, \mu(\varphi))), \quad \forall \varphi \in \mathbb{R}^n
\]

(8)

which states that the optimal policy is stationary when it satisfies the backward recursion given by (5). This condition holds for the discounted case with bounded cost and regardless of discounting when the cost function is nonnegative.

### 2.2 Policy Iteration with Perfect Knowledge

For system (1) with quadratic cost (2) define a set of stationary policies given by a linear state feedback gain, \( L \), such that the matrix \( A + bL \) has all of its eigenvalues in the unit circle. This stabilizing policy \( \mu(\varphi) = L\varphi \) yields quadratic cost-to-go function that can be expressed as

\[
J_{\mu}(\varphi_t) = \varphi_t^T K \varphi_t
\]

(9)

where \( K \) is a positive semidefinite, symmetric cost matrix. For any \( \mu_0(\varphi) = L_0\varphi \) belonging to the stabilizing set given above, equation (7) yields

\[
J_{\mu_0}(\varphi_t) = c(\varphi_t, L_0\varphi_t) + \alpha J_{\mu_0}(A + bL_0)\varphi_t
\]

(10)

with \( J_{\mu_0} \) defined by \( K_0 \) as in (9). Given that equation (10) must hold for all \( \varphi \in \mathbb{R}^n \)

\[
0 = \alpha(A + bL_0)^T K_0 (A + bL_0) - K_0 + \theta \theta^T + rL_0^T L_0
\]

(11)

Obtaining the solution \( K_0 \) to (11) denotes a policy evaluation step. The cost-to-go under \( \mu_0 \) for any \( \varphi \in \mathbb{R}^n \) can be evaluated with (9) and \( K_0 \). An improved policy \( \mu_1 \) such that \( J_{\mu_1} \leq J_{\mu_0} \) across the state space, can be determined by obtaining the minimizer of the one-step cost plus the cost-to-go for the current policy for the succeeding state over \( \mathbb{R}^n \)

\[
\mu_1(\varphi) = \arg \min_{u \in U(\varphi)} \left[ c(\varphi, u) + \alpha J_{\mu_1}(A\varphi + bu) \right]
\]

(12)

The solution to (12) denotes a policy improvement step, which can be followed by a new evaluation step to determine \( K_1 \). The algorithm that sequentially iterates between evaluation and improvement steps is known as Policy Iteration (PI). A sequence of cost matrices \( \{K_l\} \), where the index \( l \) denotes a PI step, is generated as a result. For a linear system with quadratic stage cost, PI converges to the optimal linear state feedback policy for a stabilizing \( L_0 \) and corresponds to Hower’s algorithm for solving the discrete-time algebraic Riccati equation (DARE) by repeated solutions of Lyapunov equations (Lewis et al. (2012)). Convergence is achieved when the cost matrix obtained during policy evaluation is unchanged (\( K_l = K_{l-1} \)). The corresponding policy satisfies (7), therefore it is optimal. Its cost matrix, \( K^* \), defines the stationary optimal feedback linear control policy:

\[
\mu^*(\varphi) = -\alpha(ab^T K^*b + r)^{-1}b^T K^*A\varphi
\]

(13)

For an agent with perfect knowledge of the transition dynamics and the stage cost, PI can be applied off-line, it is exact and does not require any learning. As such, it is not a ADP/RL method.

### 2.3 Q-learning

Q-learning is a RL method introduced by Watkins (1989) in order to overcome the limitations of an agent without the ability of making perfect predictions of the environment. The core idea is to learn a function that incorporates the system dynamics and the stage cost. The quantity being minimized in (7) serves as the definition for the Q function:

\[
Q(\varphi_t, u) := c(\varphi_t, u) + \alpha J^*(\varphi_{t+1})
\]

(14)
Q is the sum of the one step cost incurred by taking the admissible control action \( u \) from state \( \phi_t \) and following the policy \( \pi^*(\phi) \) thereafter. For system (1) with stage cost (2) the control that minimizes \( Q \) is given by the optimal linear state feedback control policy (13). \( Q(\phi_t, u) \) is given by a quadratic function in terms of the augmented state \( \chi = [\phi_t^T \ u]^T \)

\[
Q(\phi_t, u) = \chi^T \begin{bmatrix} a^T K^* A + \theta \theta^T & a^T K^* b \\ ab^T K^* A & ab^T K^* b + r \end{bmatrix} \chi
\]

where \( u \) is arbitrary and does not necessarily correspond to the input specified by the optimal policy. For system (1) with nonnegative stage cost (2), \( Q(\phi_t, u) \) is convex with respect to \( u \) for a given \( \phi_t \). Therefore, the optimal policy control action is given by the stationarity condition

\[
\frac{\partial Q(\phi_t, u)}{\partial u} = 0
\]

evaluated at \( \phi_t \in \mathbb{R}^n \). This expression yields exactly the same optimal linear feedback gain specified by (13). Q learning is equivalent to learning the Hamiltonian and applying the stationarity and convexity conditions according to Pontryagin’s minimum principle as shown by Mehta and Meyn (2009). Bradtke et al. (1994) formulate an adaptive control algorithm for continuous space using PI in terms of \( Q \) and provide convergence and stability results. This formulation requires the addition of white noise to the input in order to provide adequate exploration for its convergence.

3. ADAPTIVE AGENTS WITH ANTICIPATION

An important consideration in ADP/RL control methods is how to introduce excitation for the learning to take place effectively while maintaining control. The most common strategies to achieve this are \( \epsilon \)-greedy policy iteration and the addition of white noise to the input signal as discussed by Lewis et al. (2012). Heiring et al. (2015) develop a detailed description of the concept of anticipated information and its results applied to MPC for FIR systems. In the context of RL, the inclusion of anticipated information provides the agent with the ability to introduce exploration in optimal decisions along the control path. Below, a generalization of the FIR dual MPC control algorithm by Heiring et al. (2015), designed from a RL perspective, is provided. Main definitions from their development are included for completeness.

3.1 Anticipated Information

Consider the case where the system dynamics \( \{ A, b \} \) are known and all the uncertainty is contained in the vector of output parameters, \( \theta \). A Certainty Equivalence (CE) strategy relies on information contained in the set of previous control inputs and output measurements,

\[
\mathcal{Y}_t := \{ u_{t-1}, u_{t-2}, \ldots, y_t, y_{t-1}, \ldots \}. \tag{17}
\]

A CE output predictor for system (1) is given by the current estimate of \( \theta \) given \( \mathcal{Y}_t \)

\[
\hat{y}_{t+k} = E[\hat{y}_{t+k}|\mathcal{Y}_t], \quad \text{for } k = 0, 1, \ldots
\]

\[
= E[\theta^T |\mathcal{Y}_t] \phi_{t+k} + \hat{\theta}_t^T \phi_{t+k} = \hat{\theta}_t^T \phi_{t+k} \tag{18}
\]

where \( \hat{y}_{t+k} = y_t \) and \( \phi_{t+k} \) is deterministically given by the known current state, \( \phi_{t+k} = \phi_t \), and transition dynamics for a series of future control actions \( u_t, u_{t+1}, \ldots, u_{t+k-1} \)

\[
\phi_{t+k} = A \phi_{t+k} + B u_{t+k} \tag{19}
\]

The error covariance matrix, \( P_t \), associated with \( \hat{\theta}_t \) is defined as

\[
P_t := E[(\theta - \hat{\theta}_t)(\theta - \hat{\theta}_t)^T | \mathcal{Y}_t]. \tag{20}
\]

In order to incorporate the effect of future control inputs, the set of past and anticipated information, \( \mathcal{Y}_{k|t} \), is introduced

\[
\mathcal{Y}_{k|t} := \{ u_t, u_{t+1}, \ldots, u_{t+k}, \hat{P}_t, \hat{\theta}_t, \mathcal{Y}_t \} \tag{21}
\]

leading to the definition of the anticipated error covariance matrix:

\[
P_{t+k|t} := E[(\theta - \hat{\theta}_t)(\theta - \hat{\theta}_t)^T | \mathcal{Y}_{k|t}] = E[\theta \theta^T | \mathcal{Y}_{k|t}] - \hat{\theta}_t \hat{\theta}_t^T \tag{22}
\]

where \( P_{t+k|t} = P_t \). For a given set of anticipated control actions, \( u_N := \{ u_{t+k} \in U(\phi_{k}) \}_{k=0}^{N-1} \), the parameter uncertainty is propagated forward in time with Recursive Least Squares (RLS, Ljung (1999))

\[
R_{t+k+1} = R_{t+k} + \phi_{t+k+1}^T P_{t+k+1 | t} \phi_{t+k+1} + R_{t+k+1 | t} \tag{23}
\]

where \( R_{t+k+1} = P_{t+k+1}^\top \) and \( \phi_{t+k+1}^T \) is given by (19). An approximate stage cost function, related to (2), under output predictor parameter uncertainty is defined for \( k \geq 0 \), given the set of past and anticipated information

\[
\hat{c}_{t+k|t} := \phi_{t+k}^T [E[\theta^T | \mathcal{Y}_{k|t}] \phi_{t+k} + ru_{t+k}^2 + P_{t+k|t} \phi_{t+k}^T + ru_{t+k}^2] \tag{24}
\]

This nonlinear function can be expressed in terms of \( R_{t+k|t} \) by introducing an additional variable:

\[
\hat{z}_{t+k} := P_{t+k|t} \phi_{t+k} \tag{25}
\]

equivalently,

\[
R_{t+k} \hat{z}_{t+k} = \phi_{t+k} \tag{26}
\]

(25) is substituted into (24) to obtain

\[
\hat{c}_{t+k|t} = \hat{y}_{t+k}^T + ru_{t+k}^2 + \phi_{t+k}^T \hat{z}_{t+k} \tag{27}
\]

which can be computed exactly given \( \mathcal{Y}_{k|t} \) for any \( k > 0 \) with equations (18),(19),(21), and (26).

3.2 Dual Cost-To-Go

An approximate infinite-horizon cost function for the current state, \( \phi_t \in \mathbb{R}^n \), and a given policy, \( \pi \), can be written in terms of (27):

\[
\hat{J}_\pi(\phi_t) = \lim_{N \to \infty} \sum_{k=0}^{N-1} \alpha^k \hat{c}_{t+k|t} \tag{28}
\]

\[
= \lim_{N \to \infty} \sum_{k=0}^{N-1} \alpha^k \hat{y}_{t+k}^2 + \lim_{N \to \infty} \sum_{k=0}^{N-1} \alpha^k u_{t+k}^2 + \lim_{N \to \infty} \sum_{k=0}^{N-1} \alpha^k \hat{z}_{t+k}^2
\]

\[
= \hat{J}_\pi,\text{exploit}(\phi_t) + \hat{J}_\pi,\text{caution}(\phi_t) + \hat{J}_\pi,\text{explore}(\phi_t)
\]

This decomposition makes an explicit distinction of the desired features of a dual control policy: exploitation of
previous knowledge, caution, and exploration. It has the property that as $\hat{\theta}_t \to \theta$, $\hat{J}_\pi,\text{explore}(\phi_t) \to 0$, converging to the exact cost-to-go given by stage cost (2).

3.3 N-step Bellman’s Equation

A stationary optimal policy can be expressed by the N-step recursion given by (6)

$$J^*(\phi_t) = J_N^*(\phi_t) + \alpha^N J^*(\phi_{t+N})$$  

Corollary 1. Bellman’s equation can be expressed in terms of the N-step cost for a deterministic system with a stationary optimal policy and nonnegative stage cost.

$$J^*(\phi_t) = \min_{u_N} \left[ \sum_{k=0}^{N-1} \alpha^k c(\phi_{t+k}, u_{t+k}) + \alpha^N J^*(\phi_{t+N}) \right]$$  

Proof. Equation (29) can be rewritten using the definition for the optimal policy (4)

$$J_N^*(\phi_t) + \alpha^N J^*(\phi_{t+N}) = \min_{\pi \in \Pi} \left[ J_e,N(\phi_t) + \alpha^N J^*(\phi_{t+N}) \right]$$

$$\leq \min_{\pi \in \Pi} \left[ J_e,N(\phi_t) + \alpha^N J^*(\phi_{t+N}) \right]$$

$$= \min_{\pi \in \Pi} \left[ \sum_{k=0}^{N-1} \alpha^k c(\phi_{t+k}, u_{t+k}) + \alpha^N J^*(\phi_{t+N}) \right]$$

If the reverse inequality holds, (30) must be true. Rewrite equation (3) for a deterministic system by extracting the first N stage costs in the sum

$$J(\phi_t) = \sum_{k=0}^{N-1} c(\phi_{t+k}, u_{t+k}) + \alpha^N J(\phi_{t+N})$$

$$\geq \sum_{k=0}^{N-1} \alpha^k c(\phi_{t+k}, u_{t+k}) + \alpha^N J^*(\phi_{t+N})$$

$$\geq \min_{u_N} \left[ \sum_{k=0}^{N-1} \alpha^k c(\phi_{t+k}, u_{t+k}) + \alpha^N J^*(\phi_{t+N}) \right]$$

Since the inequality holds for any $\pi \in \Pi$, it must also hold for $J^*$

$$J^*(\phi_t) \geq \min_{u_N} \left[ \sum_{k=0}^{N-1} \alpha^k c(\phi_{t+k}, u_{t+k}) + \alpha^N J^*(\phi_{t+N}) \right]$$

Combining the two inequalities completes the proof.

3.4 Approximate Iteration

For system (1) with stage cost (2) under output parameter uncertainty, an approximate cost-to-go function is defined

$$\hat{J}^*(\phi_t) := \min_{u_N} \left[ \sum_{k=0}^{N-1} \alpha^k c(\phi_{t+k}, u_{t+k}) + \alpha^N \hat{J}^*(\phi_{t+N}|\hat{K}_t) \right]$$  

(31)

This approximation of (30) is composed of an N-step truncation of (28) and a CE cost-to-go. It assumes that for a large enough N, the exploration component in the cost-to-go for $\phi_{t+N}$ is negligible, and $\hat{J}^*(\phi_{t+N})$ is given by (9) with cost matrix $\hat{K}_t$, the solution to

$$\hat{K}_t = \hat{A}^T (\hat{K}_t - \hat{K}_t b(b^T \hat{K}_t b + \tilde{r})^{-1} b^T \hat{K}_t) \hat{A} + \tilde{\theta}_t \tilde{\theta}_t^T$$

(32)

where $\hat{A} = \sqrt{\alpha} \hat{A}$ and $\tilde{r} = r/\alpha$. The solution of (32) is only possible if the pair $\{A, b\}$ is controllable and the pair $\{A, \tilde{\theta}_t\}$ is observable.

By solving the optimization problem posed by (31) and implementing the first element of the minimizer, corresponding to $u_t$, an actor is specifying a optimal control signal that balances exploration with respect to a finite horizon window of size N, and the infinite horizon CE problem. Upon implementation of the optimal dual control action, a critic agent collects the subsequent output measurement, $y_{t+1}$, and provides improved approximations for the output parameter vector $\hat{\theta}_{t+1}$, $P_{t+1}$ and the cost-to-go function $J^*$. A new optimization problem is defined and the procedure can be repeated forward in time. This process is similar to PI in Q learning except the function being approximated includes the cost of N steps, and it has an embedded mechanism to generate excitation.

3.5 Control Algorithm

An indirect adaptive optimal control algorithm is presented based on the process described above. The distinct feature of the proposed method emerges from the inclusion of anticipated information, which is expressed in the optimal action that balances dual control considerations. In the subsequent discussion, the algorithm is referred to as RL dual MPC (RLdMPC). A detailed description is given in Fig. 1.

4. COMPUTATIONAL RESULTS

The optimization problem (33) is a Quadratically Constrained Quadratic Program (QCQP) in terms of the column vector $x$

$$x := \begin{bmatrix} [\tilde{y}_t | y_t + N]_1^T \\ [u_t | u_{t+N-1}]^T \\ [\phi_{t+1}]^T \\ \vdots \\ [\phi_{t+1}]^T \\ [z_t | z_{t+N-1}]^T \end{bmatrix}$$

(35)

$$\begin{bmatrix} \hat{R}_{t+k} \end{bmatrix}$$

where $\hat{R}_{t+k}$ is a row vector with $n^2$ elements corresponding to the concatenation of the rows of $\hat{R}_{t+k}$. Column vector $x$ contains all the variables involved in (33) and it has $Nn^2 + 2N(n+1)$ elements. The equations that define this optimization problem can be written, in terms of $x$, in the following form:

$$\min_x \ x^T M_x x + a_x^T x$$

s.t. $\begin{bmatrix} x^T M_x x + a_x^T x = b_i \\ x^T M_x x + a_x^T x \leq b_i \end{bmatrix} \in \mathcal{E} \in \mathcal{I}$

$$\begin{bmatrix} x_{min} \leq x \leq x_{max} \end{bmatrix}$$

where $\mathcal{E}$ and $\mathcal{I}$ are the sets of equality and inequality constraints respectively. $\mathcal{E}$ is composed of $n^2 + N(n+1)$ linear equations and $(N-1)n^2 + Nn$ quadratic equations. A single linear inequality and $N-1$ quadratic equations constitute $\mathcal{I}$.

4.1 Example

Consider system (1) specified by

$$A = \begin{bmatrix} 2.5 & -1 & -0.1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \theta = \begin{bmatrix} 5 \\ 2 \\ -1 \end{bmatrix}$$

269
Initialization: Specify the initial state, \( \varphi_{t_0} \), output parameter estimate, \( \hat{\theta}_{t_0} \), and its covariance matrix, \( P_{t_0} \). Set \( K_{t_0} = 0 \). Set \( t_0 \to t \)

Step 1: Check observability of \( \{ A, \hat{\theta}_t \} \). If it checks, solve (32) and update \( \hat{K}_t \) with its solution. If it doesn’t check, keep previous \( \hat{K}_t \).

Step 2: Solve
\[
\begin{align*}
\min_{u_N} & \sum_{k=0}^{N-1} \alpha^k (\tilde{y}_{t+k|t}^2 + \varphi_{t+k|t}^T z_{t+k}) \\
\text{s.t.} & \varphi_{t+k+1|t} = A \varphi_{t+k|t} + b u_{t+k} \\
& \tilde{y}_{t+k} = \begin{bmatrix} y_t \\ \hat{\varphi}_{t+k|t} \end{bmatrix}, \quad k > 0 \\
& R_{t+k} = \begin{cases} P_t^{-1}, & k = 0 \\ R_{t+k-1} + \varphi_{t+k|t}^T \varphi_{t+k|t}^T, & k > 0 \end{cases} \quad (33) \\
& \varphi_{t+k|t}^T z_{t+k} > 0 \\
& y_{\min} \leq \tilde{y}_{t+k|t} \leq y_{\max} \\
& u_{\min} \leq u_{t+k} \leq u_{\max} \\
& k \in [0, N-1]
\end{align*}
\]
and parameter estimate \( \hat{\varphi}_{t+1|t} = A \hat{\varphi}_{t|t} + b u_t \), and estimate \( \hat{\theta}_{t+1} = \hat{\theta}_t + G_{t+1} (y_{t+1} - \hat{\theta}_t^T \varphi_{t+1|t}) \)
\[
\begin{align*}
G_{t+1} &= P_t \varphi_{t+1|t} (\sigma^2 + \varphi_{t+1|t}^T P_t \varphi_{t+1|t})^{-1} \\
P_{t+1} &= (I_n - G_{t+1} \varphi_{t+1|t}) P_t
\end{align*}
\]
Step 4: Set \( t + 1 \to t \), return to step 1.

Step 3: Measure \( y_{t+1} \), update state \( \varphi_{t+1} = A \varphi_{t} + b u_t \), and estimate
\[
\begin{align*}
\hat{\theta}_{t+1} &= \hat{\theta}_t + G_{t+1} (y_{t+1} - \hat{\theta}_t^T \varphi_{t+1|t}) \\
G_{t+1} &= P_t \varphi_{t+1|t} (\sigma^2 + \varphi_{t+1|t}^T P_t \varphi_{t+1|t})^{-1} \\
P_{t+1} &= (I_n - G_{t+1} \varphi_{t+1|t}) P_t
\end{align*}
\]

Step 4: Set \( t + 1 \to t \), return to step 1.

Fig. 1. RLdMPC Algorithm

and \( v_t \sim \mathcal{N}(0, 1) \). The following output feedback regulator methods were applied:

(i) RLdMPC: \( N = 3, \hat{\theta}_{t_0} = [0 \ 0 \ 0]^T, P_{t_0} = 10^3 I_3 \)
(ii) same as (i) except \( \hat{K}_t = 0, \forall t \)
(iii) same as (i), except \( \hat{\varphi}_{t+k|t} = \tilde{y}_{t+k|t}^T + r u_{t+k}^2 \)
(iv) \( \mu^* \) computed exactly with (13)

The discount factor and the caution parameter were set at \( \alpha = 1 \) and \( r = 1 \) respectively. A common initial condition for the state, \( \varphi_{t_0} \), and noise sequence \( v_t \) were generated using the function \texttt{randn} in MATLAB®. Box constraints for the system input and output were specified with \( y_{\min} = -2000, \ u_{\min} = -100, \ y_{\max} = 2000, \) and \( u_{\max} = 100 \). Optimizations for cases (i)-(iii) were solved to global \( \epsilon \)-optimality with BARON (Tawarmalani and Sahinidis (2005)) through its MATLAB interface with a relative termination tolerance, \( \epsilon_r = 10^{-3} \). The average solution time for cases (i)-(iii) were 1.017, 0.317, and 0.274 seconds in that order. An additional case equivalent to (iii) without the cost-to-go for \( \varphi_{t+N} \) \( (\hat{K}_t = 0, \forall t) \) was also studied, however, it resulted in an infeasible optimization problem after a few steps.

The stage costs from \( t_0 \) to \( t_0 + 10 \) were added for each case using (2) and are shown in Fig. 2. By definition, optimal policy (13) gives the minimum cost, and can be considered a base cost. Any additional cost can be attributed to learning and instability. Thus, RLdMPC achieves the minimum deviation from optimality. The difference between cases (i) and (iii) is achieved by including the cost to go after \( N \) steps, while the difference between cases (i) and (ii) is obtained by including the exploratory component of (27) \( \hat{\varphi}_{t+k|t}^T z_{t+k} \) in the finite horizon portion of the objective function.

Fig. 2. Observed Total Cost

Fig. 3 displays the input and output trajectories control methods (i)-(iv). In agreement with Fig. 2, the RLdMPC trajectories are the closest to optimal, achieving fastest regulation with minimum input compared to cases (ii) and (iii). The cross marks in the bottom plot correspond to the optimal input computed with (13) for the RLdMPC state trajectory. At the second step, RLdMPC converges with the optimal policy and remains optimal thereafter.

Fig. 3. Input/Output Trajectories

The output predictor parameter identification with RLS is shown in Fig. 4 for methods (i)-(iii). The right column provides a zoomed in view once the estimates have stabi-
lized. All cases behave similarly, converging near the true value after 3 steps.

Fig. 4. Output Parameter Identification

5. CONCLUSION

RLdMPC was shown to converge to the optimal policy by the sequential interaction of an actor agent that solves an optimization problem and a critic agent that provides improved approximations to the system model and cost-to-go function. The main contribution is the inclusion of anticipated information to the decision process. This feature offers an alternative to introduce excitation automatically along the control path with an optimal balance of dual objectives. The cost of this inclusion compared to a CE strategy is expressed in an increase in problem size and the solution of a QCQP instead of a QP for the optimal control problem. The size increment could be alleviated by reducing the number of anticipation steps in the objective function. The results highlight the stabilizing effect of an approximate cost-to-go in an adaptive receding horizon MPC strategy. RLdMPC was shown for the regulation case but can be easily extended to a tracking objective.

ACKNOWLEDGEMENTS

The authors would like to thank the ACS Petroleum Research Fund for their financial support.

REFERENCES


