MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

Self-tuning Optimal Torque Control for Servomotor Drives Via Adaptive Dynamic Programming

Wang, Yebin

TR2022-125 October 12, 2022

Abstract

Data-driven methods for learning optimal control policies such as adaptive dynamic programming have garnered widespread attention. A strong contrast to full-fledged theoretical research is the scarcity of demonstrated successes in industrial applications. This paper extends an established data-driven solution for a class of adaptive optimal linear output regulation problem to achieve self-tuning torque control of servomotor drives, and thus enables online adaptation to unknown motor resistance, inductance, and permanent magnet flux. We make contributions by tackling three practical issues/challenges: 1) tailor the baseline algorithm to reduce computation burden; 2) demonstrate the necessity of perturbing reference in order to learn feedforward gain matrix; 3) generalize the algorithm to the case where F matrix in the output equation is unknown. Simulation demonstrates that the deployment of adaptive dynamic programming lands at optimal torque tracking policies.

IEEE International Conference on Systems, Man, and Cybernetics (SMC) 2022

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Self-tuning optimal torque control for servomotor drives via adaptive dynamic programming

Yebin Wang

Abstract—**Data-driven methods for learning optimal control policies such as adaptive dynamic programming have garnered widespread attention. A strong contrast to full-fledged theoretical research is the scarcity of demonstrated successes in industrial applications. This paper extends an established data-driven solution for a class of adaptive optimal linear output regulation problem to achieve self-tuning torque control of servomotor drives, and thus enables online adaptation to unknown motor resistance, inductance, and permanent magnet flux. We make contributions by tackling three practical issues/challenges: 1) tailor the baseline algorithm to reduce computation burden; 2) demonstrate the necessity of perturbing reference in order to learn feedforward gain matrix; 3) generalize the algorithm to the case where F matrix in the output equation is unknown. Simulation demonstrates that the deployment of adaptive dynamic programming lands at optimal torque tracking policies.**

I. Introduction

Permanent magnetic synchronous motors (PMSMs) have been increasingly found in a wide range of application across factory automation and electrified transportation, thanks to their high torque density and specific power density. Various applications put distinctive emphasis and requirements on PMSMs, such as, productivity for automation related applications, and energy efficiency for transportation sector. This means that control in motor drives should exhibit certain flexibility, i.e., allowing customers to reconfigure objectives and achieving online adaptation to varying load conditions. Meanwhile, the desire to build a sustainable society calls for further refining to strike the balance between energy consumption and productivity. All these pose challenges and opportunities to motor designers and control engineers.

This work investigates optimal self-tuning control design for surface-mounted PMSM (SPMSM) operating in torque control mode, i.e., the motor torque needs to track a reference while operating at constant flux and speed. For simplicity, full state feedback is assumed. Under such setup, SPMSM dynamics can be modeled as a linear time-invariant (LTI) system, with reference torque and unknown model parameters. It falls into the category of linear output regulation problem (LORP), where a plethora of theoretical contributions have been proposed. For example, seminar papers [1], [2] propose the internal model principle. It is extension to nonlinear systems can be thoroughly investigated in [3]–[5]. Assuming the exact knowledge of model, [6], [7] study optimal LORP to shape transient performance. Parametric uncertainties in model are treated by adaptive control theory [8] for LTI systems, and [9], [10] for nonlinear systems. Recent work [11], [12] study the optimal tracking of nonlinear system whereas the plant dynamics are exactly known. Little work has been focusing on adaptive optimal LORP (AOLORP) which not only deals with parametric uncertainties in the model, but also takes care of optimal performance.

Based on adaptive dynamic programming (ADP), recent work [13] proposes a data-driven solution to solve LORP optimally, even if when the plant model contains parametric uncertainties. Inspired by the advancement, this paper investigates the potential of applying [13], [14] to enhance flexibility and performance of servomotor drives running in torque control mode. From theoretical perspective, this paper is strongly tied to a series of theoretical work on ADP for LTI systems [13], [15]–[17].

Instead of making contribution to theory, this work is devoted to tackling three practical issues and challenges arising from the course of applying [13] to SPMSM. First, with resistance and inductance unknown, AOLORP for the SPMSM can be viewed as a special case of problem studied in [13]. It is meaningful to customize the original algorithm to reduce computation burden during learning process. Second, in the course of applying the customized algorithm, we observe that the customized algorithm is unable to solve the AOLORP, specifically failing to learn feedforward gain. Our analysis shows that the constructed regression matrix for the learning of the value function, feedback and feedforward control policies suffers from column rank deficiency. This discovery prompts a necessary step which learns the feedforward gain matrix by perturbing the reference signal. This leads to a two-step algorithm. Finally, in the absence of permanent magnet flux knowledge, the SPMSM admits a form where matrix F in the output equation is unknown, and thus cannot be treated by established results. We propose to first identify the permanent magnet flux by putting the SPMSM at no load condition and thus follow the two-step algorithm for learning the feedback and feedforward gain matrices. Simulation verifies that the proposed algorithm achieves convergent parameter estimation, and synthesizes the optimal output regulation control policies.

The rest of the paper is organized as follows. In section II, we formulate the self-tuning torque tracking control problem and introduce the established data-driven solution to AOLORP. Section III presents two main algorithms to synthesize optimal torque control policies, when the perma-

[1]Y. Wang is with Mitsubishi Electric Research Laboratories, Cambridge, MA 02139, USA. Email: yebinwang@ieee.org.

nent magnet flux is known, whereas Section IV investigates the case where the permanent magnet flux is unknown. Simulation results are reported in Section V. Conclusions and future work are made in Section VI. Notations used in this work are summarized in Table I.

TABLE I
NOTATIONS USED IN THE SPMSM MODEL

Notation	Description
ϕ_d	rotor flux
Ω	rotor speed
ϕ_{pm}	permanent magnet flux
i_d, i_q	current in d- and q-axis
u_d, u_q	voltage in d- and q-axis
L_s	stator inductance
p	number of pole pairs
R_s	winding resistance
γ	R_s/L_s
J	rotor inertia
T_L	load torque

II. PRELIMINARIES

The SPMSM dynamics are modeled as a system of ordinary differential equations which admit the following state space representation

$$\begin{aligned} \dot{i}_d &= -\gamma i_d - p\Omega i_q + \frac{u_d}{L_s} \\ \dot{i}_q &= -\gamma i_q - p\Omega(i_d + \frac{\phi_{pm}}{L_s}) + \frac{u_q}{L_s} \\ J\dot{\Omega} &= \frac{3p}{2}(L_s i_d + \phi_{pm}) i_q - T_L \\ y &= [i_d, i_q, \Omega]^\top, \end{aligned} \tag{1}$$

where u_d, u_q are control input. In the well-established vector control framework [18], [19], i_d is used to regulate the rotor flux ϕ_d which, with its dynamics being omitted, can be written as $\phi_d = L_s i_d + \phi_{pm}$; and i_q corresponds to the torque that the SPMSM produces, given by $T_e = 3p\phi_d i_q/2$. Note that i_d, i_q can be controlled by u_d and u_q, respectively, and thus ϕ_d and T_e can be regulated independently.

If the rotor speed is not beyond the rated value, ϕ_{pm} is the desired rotor flux for torque generation, i.e., $\phi_d^* = \phi_{pm}$, which implies $i_d^* = 0$. In order for SPMSM to operate beyond its rated speed, a non-zero i_d is applied to weaken the magnetic field established by permanent magnet so that control voltage u_q can overcome back-electromagnetic force. Regular operation is assumed in this work, i.e., $\phi_d^* = \phi_{pm}$ and $i_d^* = 0$.

In industrial applications, the SPMSM is typically put into either torque or velocity control mode. In torque control mode, the motor should produce a prescribed torque T_L^* whatever its rotor speed is; and in velocity control mode, the motor should track a prescribed speed no matter what load torque is. This work considers self-tuning in torque control mode, where Ω is constant or piecewise constant with long enough dwelling time.

A. Self-tuning torque control problem

In torque control mode, the rotor speed Ω is determined by external apparatus, and not a state variable anymore. Hence the SPMSM model is reduced to a second order ordinary differential equation representing the dynamics of i_d and i_q:

$$\begin{aligned} \dot{i}_d &= -\gamma i_d + p\Omega i_q + \frac{u_d}{L_s} \\ \dot{i}_q &= -\gamma i_q - p\Omega(i_d + \frac{\phi_{pm}}{L_s}) + \frac{u_q}{L_s} \\ y &= [i_d, i_q]^\top, \end{aligned} \tag{2}$$

where Ω is a known parameter. It is clear that the model (2) is LTI. Given state i_d, i_q, the SPMSM generates torque $T_e = \frac{3p}{2}(L_s i_d + \phi_{pm}) i_q$. With $\phi_d^* = \phi_{pm}$, the operation point (ϕ_d^*, T_L^*) corresponds to the equilibrium $x_e = \left[0, \frac{2T_L^*}{3p\phi_{pm}}\right]^\top$.

Torque control problem is to design $u(x, x_e)$ such that $x(t) \to x_e$ as $t \to \infty$. Because the reference T_L^* could change, torque control essentially requires to solve an LORP with the regulated variable being $e(t) = x(t) - x_e$.

In the case that all model parameters are known, well-recognized results in [1], [2], [5] can be directly applied to find the solution. Alternatively, one can first determine the non-vanishing control u_e at $x = x_e$, given by

$$u_e = \begin{bmatrix} p\Omega L_s i_q^* \\ L_s \gamma i_q^* + p\Omega\phi_{pm} \end{bmatrix}. \tag{3}$$

Let $u = u_e + v(e)$ with $v(e) = [v_d(e), v_q(e)]^\top$. One can obtain the tracking error dynamics

$$\begin{aligned} \dot{e}_d &= -\gamma e_d - p\Omega e_q + \frac{v_d}{L_s} \\ \dot{e}_q &= -p\Omega e_d - \gamma e_q + \frac{v_q}{L_s}. \end{aligned} \tag{4}$$

Output regulation problem is reduced to stabilization of $e = 0$. Linear control theory, such as linear quadratic regulator (LQR), can be employed to achieve optimal torque tracking.

It is unfortunate that exact values of R_s, L_s, ϕ_{pm} are unavailable in practice. The torque control has to combat against parametric uncertainties. Meanwhile, it is desirable to take the transient performance into account, considering that SPMSM is expected to operate mostly in transient. The performance and adaptability entails self-tuning optimal torque control by studying AOLORP. The self-tuning optimal torque control problem is formulated below:

Problem 1. Let SPMSM operate in torque control mode. Given the SPMSM dynamical model (2) with R_s, L_s (possibly ϕ_{pm}) being unknown parameters. Find an optimal feedback forward control u_e^* and an optimal state-feedback control policy $v^*(e)$ to ensure: 1) the zero solution of the closed-loop system (or e-dynamics) is globally exponentially stable; and 2) the following cost function is minimized during the transient toward $e = 0$

$$\mathcal{C}(u) = \int_0^\infty \left\{ e^\top Q e + (v(e))^\top R v(e) \right\} \mathrm{d}t, \tag{5}$$

where Q and R are positive definite matrices that weight tracking accuracy and control costs, respectively.

B. Adaptive optimal output regulation for LTI systems

Consider the following LTI system

$$\begin{aligned}\dot{x} &= Ax + Bu + Dw \\ \dot{w} &= Ew \\ e &= Cx + Fw,\end{aligned} \tag{6}$$

where $x \in \mathbb{R}^n$ is the system state,$u \in \mathbb{R}^m$ control input, $w \in \mathbb{R}^q$ the exogenous system state, and $e \in \mathbb{R}^r$ the tracking error. All system matrices are known, and w is the disturbance or reference. A rough formulation of LORP is: given (6), design a controller $u = -Kx + Lw$ such that the closed-loop system is globally exponentially stable and the tracking error $e(t)$ converges to zero. What is available for controller synthesis is $(x, e, w, A, B, C, D, E, F)$. Seminal paper [1] establishes that LORP is solvable if (A, B) is stabilizable, and there exists a solution to the following regulator equations

$$\begin{aligned}AX + BU + D &= XE \\ CX + F &= 0,\end{aligned} \tag{7}$$

with $L = U + KX$.

Remark 1. LORP admits a solution if

$$\text{rank} \begin{bmatrix} A - \lambda \mathbf{I}_n & B \\ C & 0 \end{bmatrix} = n + r$$

where λ is any eigenvalue of E [5, Thm. 1.9]. If LORP is solvable, then for any initial condition $x(0)$ and $w(0)$, $\lim_{t\to\infty}(u(t) - Uw(t)) = 0$ and $\lim_{t\to\infty}(x(t) - Xw(t)) = 0$.

Remark 2. The solution to optimal LORP, assuming the knowledge of system matrices A, B, C, D, E, F, can be readily attained by solving a convex optimization problem and algebraic Riccati equation sequentially.

Work [13] investigates AOLORP, where matrices A, B, D are unknown. Since the control for LOPR takes alternative parametrization: $u = -K\bar{x} + Uw$ where $\bar{x} = x - Xw$, AOLORP is about finding the optimal (X^*, U^*) satisfying

$$\min_{(X,U)} \text{Tr}(X^\top QX + U^\top RU) \text{ subject to (7)},$$

where $Q > 0, R > 0$; furthermore, with $\bar{u} = u - U^* w, \bar{x} = x - X^* w$, and its dynamics

$$\begin{aligned}\dot{\bar{x}} &= A\bar{x} + B\bar{u} \\ e &= C\bar{x},\end{aligned}$$

the optimal feedback control $\bar{u} = -K^*\bar{x}$ minimizes

$$J(\bar{u}) = \int_0^\infty \{\bar{x}^\top Q\bar{x} + \bar{u}^\top R\bar{u}\}\mathrm{d}t,$$

where $(A, \sqrt{Q})$ observable.

III. Self-tuning Torque Control: Known Flux

This section deals with the self-tuning torque control when R_s, L_s are unknown by solving AOLORP. We work on the i_d- and i_q-dynamics instead of e-dynamics. The signals to be regulated are $e = \left[i_d, i_q - i_q^*\right]^\top$, where i_q^* may or may not be known. Rewrite (4) in the form of (6) with

$$A = \begin{bmatrix} -\gamma & p\Omega \\ -p\Omega & -\gamma \end{bmatrix},\ B = \frac{1}{L_s}\mathbf{I}_2,\ C = \mathbf{I}_2,\ E = \mathbf{0},$$

and the expressions of matrices D, F depend on how exogenous signals are defined. The target of output regulation problem is to find control $u = u_e + v$ such that $\lim_{t\to+\infty} e(t) = 0$, while minimizing the cost function (5).

A. Customized algorithm

The SPMSM dynamics (2) is written in the form (6) with

$$D = \begin{bmatrix} 0 & 0 \\ \frac{1}{L_s} & 0 \end{bmatrix},\ F = \begin{bmatrix} 0 & 0 \\ 0 & \frac{-2}{3p\phi_{pm}} \end{bmatrix},\ w = \begin{bmatrix} p\Omega\phi_{pm} \\ T_L^* \end{bmatrix}. \tag{8}$$

With R_s, L_s unknown, matrices A, B, D are unknown, whereas matrices C, E, F are known.

Output regulation problem for LTI systems with unknown matrices A, B, D has been solved in [13]. With $E = 0$, self-tuning torque control for SPMSM can be treated as a special case of AOLORP considered in [13]. Specifically, the regulator equations are

$$\begin{aligned}AX + BU + D &= 0 \\ CX + F &= 0.\end{aligned} \tag{9}$$

One can immediately solve $X^* = -F$, which can be further exploited to customize the algorithm and thus lower computation burden.

What's left in (9) is to solve U^* with A, B, D being unknown. Since B is non-singular, one can immediately conclude that LORP admits a unique solution $U^* = -L_s(AX + D)$, i.e., there is no freedom in designing the steady-state control to optimize system performance. We can skip the Sylvester map and rewrite the regulation equation (9) into the following vector form

$$-(\mathbf{I}_2 \otimes B)\text{vec}(U) = \text{vec}(D + AX^*), \tag{10}$$

where B and the right hand side (RHS) will be attained as a byproduct of solving the optimal state feedback control $\bar{u} = K^*\bar{x}$ using data-driven policy iteration (PI) algorithm [20].

To facilitate the following derivation, we let $\bar{x} = x - X^* w$ and have its dynamics given by

$$\begin{aligned}\dot{\bar{x}} &= A\bar{x} + Bu + (AX^* + D)w \\ e &= C\bar{x}.\end{aligned}$$

If there exists a stabilizing state feedback control law $-K\bar{x}$ as a priori, we have

$$\dot{\bar{x}} = (A - BK)\bar{x} + B(u + K\bar{x}) + (AX^* + D)w. \tag{11}$$

Given (11), data-driven PI algorithm is employed to synthesize K^*, and determine $B, \text{vec}(AX + D)$. As usual, it begins with an initial stabilizing feedback policy $\boldsymbol{v}_0 = K_0\bar{x}$, and the value function is parameterized as a positive definite quadratic function, i.e., $V(\bar{x}) = \bar{x}^\top P_j \bar{x}$ with $P_j > 0$. At the jth policy iteration, P_j, K_{j+1} and $\text{vec}(AX + D)$ will be determined, given $\boldsymbol{v}_j = K_j\bar{x}$.

The knowledge of X^* greatly simplifies the search for K^*. Specifically, one can construct the regressor matrix based on the system $\bar{x} = x - X^* w$ instead of a sequence of $\bar{x}_i = x - X_i w$, where X_i is the basis of the solution space of $CX + F = 0$ in [13]. Hence the evaluation of a control policy $\boldsymbol{v}_j$, i.e., solving the corresponding value function, along the trajectory of the resultant closed-loop system (11) over the time interval $[t, t+\delta t]$, can be simplified as follows

$$\begin{aligned}
&\bar{x}^\top(t+\delta t) P_j \bar{x}(t+\delta t) - \bar{x}^\top(t) P_j \bar{x}(t) \\
&= \int_t^{t+\delta t} \left\{ \bar{x}^\top (A_j^\top P_j + P_j A_j)\bar{x} + 2(u + K_j \bar{x})^\top B^\top P_j \bar{x} \right. \\
&\left. + 2 w^\top (D + AX^*)^\top P_j \bar{x} \right\} \mathrm{d}\tau \\
&= -\int_t^{t+\delta t} \bar{x}^\top (Q + K_j^\top R K_j) \bar{x} \mathrm{d}\tau \\
&+ \int_t^{t+\delta t} 2(u + K_j \bar{x})^\top R K_{j+1} \bar{x} \mathrm{d}\tau \\
&+ \int_t^{t+\delta t} 2 w^\top (D + AX^*)^\top P_j \bar{x} \mathrm{d}\tau,
\end{aligned}$$

where the first term corresponds to cost related to feedback control, the second term is contributed by the non-vanishing control, and the third term related to exosignal. Vectorization of the aforementioned matrix equation gives the following linear equations from data over $[t, t+\delta t]$

$$\begin{bmatrix} \psi_P(t) & \psi_K(t) & \psi_D(t) \end{bmatrix} \Theta = \psi_b(t)$$

where

$$\begin{aligned}
\psi_P(t) &= \widehat{\mathrm{vecs}}(x(t+\delta t)) - \widehat{\mathrm{vecs}}(x(t)) \\
\psi_K(t) &= -2 \int_t^{t+\delta t} (\bar{x}^\top \otimes (u + K_j \bar{x})^\top) \mathrm{d}\tau \times (\mathbf{I}_n \otimes R) \\
\psi_D(t) &= -2 \int_t^{t+\delta t} (\bar{x}^\top \otimes w^\top) \mathrm{d}\tau \\
\Theta &= \begin{bmatrix} \mathrm{vecs}(P_j) \\ \mathrm{vec}(K_{j+1}) \\ \mathrm{vec}((D + AX^*)^\top P_j) \end{bmatrix} \\
\psi_b(t) &= -\int_t^{t+\delta t} \bar{x}^\top (Q + K_j^\top R K_j) \bar{x} \mathrm{d}\tau.
\end{aligned}$$

Here for a vector $x = [x_1, \cdots, x_n]^\top \in \mathbb{R}^n$, $\widehat{\mathrm{vecs}}(x) = [x_1^2, \cdots, x_n^2, x_1 x_2, \cdots, x_{n-1} x_n]^\top$; for a positive definite matrix $P \in \mathbb{R}^{n \times n}$, $\mathrm{vecs}(P) = [P_{11}, \cdots, P_{nn}, 2P_{12}, \cdots, 2P_{n-1,n}]^\top$; and for a matrix $T \in \mathbb{R}^{n \times m}$, $\mathrm{vec}(T) = [T_1^\top, \cdots, T_m^\top]^\top$. For self-tuning torque control, the aforementioned linear equations contain 11 variables to solve, and thus requires collecting data over at least 11 time intervals. Assume we collect N data and have

$$\begin{aligned}
\Psi \Theta &= \Psi_b \\
\Psi &= \begin{bmatrix} \psi_P(t_1) & \psi_K(t_1) & \psi_D(t_1) \\ \vdots & \vdots & \vdots \\ \psi_P(t_N) & \psi_K(t_N) & \psi_D(t_N) \end{bmatrix} \qquad (12) \\
\Psi_b &= \begin{bmatrix} \psi_b(t_1) & \cdots & \psi_b(t_N) \end{bmatrix}^\top.
\end{aligned}$$

With Θ being solved, matrix B can be determined as: $B = (R K_{j+1} P_j^{-1})^\top$. With the knowledge of $(D + AX^*)$, feedforward gain matrix U^* can be readily determined from (10). The aforementioned procedure is summarized by Algorithm 1, where $0 < \epsilon_2 \leq \epsilon_1 \ll 1$.

Algorithm 1: One-step algorithm for AOLORP

```
Initialize j = 0, δt, M;
Design an initial feedback control policy u_fb = K_0 x
  and feedforward control u_ff = U_0 w;
for j ≤ M do
    Solve (12) for Θ_j by applying
      u_j(x̄) = −K_j x̄ + U_j w + ρ(t), where ρ(t) be a
      vector of perturbation signals over [t, t + T_s];
    Update K_{j+1} for next iteration;
    if |Θ_{j−1} − Θ_j| < ε_1 then
        Update feedforward gain U_{j+1};
        if |Θ_{j−1} − Θ_j| < ε_2 then
            Break;
return Θ_j;
```

B. Two-step algorithm

To facilitate the derivation of the two-step algorithm, we define vectors of parameters $\Theta_1 = [\mathrm{vecs}(P)^\top, \mathrm{vec}(K)^\top]^\top$, and $\Theta_2 = \mathrm{vec}((D + AX)^\top P)$.

Let us take a close examination of the regression matrix Ψ. Particularly, for ψ_D, we have $\bar{x} \otimes w = [\bar{x}_1 w, \cdots, \bar{x}_n w]$. For the first q columns, we have $[\bar{x}_1 w_1, \cdots, \bar{x}_1 w_q]$. Hence, for N data points, we rearrange the first q columns below

$$\Psi_{D1} = \begin{bmatrix} \int_0^{\delta t} \bar{x}_1(\tau) w_1(\tau) \mathrm{d}\tau & \cdots & \int_0^{\delta t} \bar{x}_1(\tau) w_q(\tau) \mathrm{d}\tau \\ \vdots & \vdots & \vdots \\ \int_t^{t+\delta t} \bar{x}_1(\tau) w_1(\tau) \mathrm{d}\tau & \cdots & \int_t^{t+\delta t} \bar{x}_1(\tau) w_q(\tau) \mathrm{d}\tau \end{bmatrix}$$

Apparently, if w is constant over $[0, t+\delta t]$, Ψ_{D1} has rank 1. Hence, Θ_2 is not identifiable, and thus the feedforward gain matrix cannot be learned. This implies that one has to adopt a time-varying reference w during data collection to ensure that Ψ_D is full-column rank.

It is natural to modify Alg. 1 by perturbing reference w with an additive time-varying signal $w_d(t)$. Such a treatment however is susceptible to numerical issues, because of the co-existence of $\rho(t)$. In fact, one can see that perturbation $w_d(t), \rho(t)$ have similar impacts on Ψ_K and Ψ_D. Take a special case $R = \mathbf{I}_m$ as an example. We have $\mathbf{I}_n \otimes R$ as an identify matrix. The impacts of perturbations $\rho(t), w_d(t)$ on Ψ_K and Ψ_D can be represented by the following two terms

$$\delta \Psi_K = \int_t^{t+\delta t} \bar{x}^\top \otimes \rho^\top \mathrm{d}\tau, \ \delta \Psi_D = \int_t^{t+\delta t} \bar{x}^\top \otimes w_d^\top \mathrm{d}\tau,$$

The mutual-canceling effects of $\rho(t), w_d(t)$ would not compromise the learning K because u is time-varying, but could significantly jeopardize the rank condition of Ψ_D. As a result, we propose to learn feedback control-related parameters, e.g.,

Θ_1, and feedforward-related parameters, e.g., Θ_2, separately. Particularly, the second step tries to solve for Θ_2 from:

$$\psi_D(t)\Theta_2 = \psi_b(t) - [\psi_P(t), \psi_K(t)]\Theta_1. \quad (13)$$

The two-step algorithm is summarized in Alg. 2.

Algorithm 2: Two-step algorithm for AOLORP

Initialize $j = 0, t_0 = 0, M$;
Design an initial feedback control policy $u_{fb} = K_0 x$ and feedforward control $u_{ff} = U_0 w$;
for $j \leq M$ **do**
  Solve (12) for Θ_j by applying $u_j(\bar{x}) = -K_j\bar{x} + U_j w + \rho(t)$, where $\rho(t)$ be a vector of perturbation signals over $[t, t+T_s]$;
  Update K_{j+1} for next iteration;
  if $|\Theta_{1,j-1} - \Theta_{1,j}| < \epsilon_1$ **then**
    Solve (13) for Θ_2 by applying $u_j(\bar{x}) = -K_j\bar{x} + U_j w$ and $w(t) = w + w_d(t)$, where $w_d(t)$ be a vector of perturbation signals over $[t, t+T_s]$;
    Update Θ_j according to Θ_2;
    Update U_{j+1} according to Θ_j;
    if $|\Theta_{j-1} - \Theta_j| < \epsilon_2$ **then**
      Break;
return Θ_j;

IV. Self-tuning Torque Control: Unknown Flux

The absence of ϕ_{pm} means the previously defined F is unknown. Matrices D, F and exogenous signal w have the the following expressions

$$D = \begin{bmatrix} 0 & 0 \\ \frac{\phi_{pm}}{L_s} & 0 \end{bmatrix}, \ F = \begin{bmatrix} 0 & 0 \\ 0 & \frac{-2}{3p\phi_{pm}} \end{bmatrix}, \ w = \begin{bmatrix} p\Omega \\ T_L^* \end{bmatrix}. \quad (14)$$

The method developed in [13] is not directly applicable, because X_i cannot be constructed. We propose the following two-step algorithm to solve such a case.

Algorithm 3: Three-step solution

Initialize $j = 0, t_0 = 0, \delta, M_j, M_i, \bar{\Theta}_0^{Vg}$;
Design an initial feedback control policy $u = K_0 x$;
Operate motor with $T_L^* = 0$ and solve the first AOLORP for $K_j, P_j, (AX + D)_j, U_j$;
Solve the regulation equations (9) with D, F given by (8) for a solution $\hat{X}, \hat{U}$;
Optimal control policy is constructed as:

$$u = -K^* x + (K^* X^* + U^*)w,$$

where $K^* = K_j, X^* = \hat{X}, U^* = \hat{U}$.

At first, with $T_L^* = 0$, we have $e = 0 \Leftrightarrow x = 0$, and thus output regulation problem is reduced to the problem of stabilizing $x = 0$. The stabilization has the same problem setup as the case of known ϕ_{pm}. That is: A, B, D are unknown and C, F are known. Particularly, A, B, C are exactly the same as the known ϕ_{pm} case, while D, F are slightly different:

$$D = \left[0, \frac{\phi_{pm}}{L_s}\right]^\top, \quad F = 0. \quad (15)$$

One can follow the procedure outlined in Section III to solve Θ. Thanks to $F = 0$, the second regulation equation gives a unique zero solution $X^* = 0$. Thus we have a simpler expression of Θ

$$\Theta = \begin{bmatrix} \text{vecs}(P_j) \\ \text{vec}(K_{j+1}) \\ \text{vec}(D^\top P_j) \end{bmatrix}.$$

Once Alg. 2 converges, we obtain K_j, P_j, construct D from P_j and $\text{vec}(D^\top P_j)$, and solve U_j from (10).

At the second step, with non-zero T_L^* and the knowledge of B, D, ϕ_{pm}, one can construct a solution to output regulation problem as

$$u_j = -K_j\bar{x} + U_j w$$
$$\bar{x} = x - X_j w,$$

where $X_j = -F$.

Proposition 1. Assume that the first step gives a unique solution of $K_j, P_j, \hat{\phi}_{pm}, U_j, X_j$ with $T_L^* = 0$; and the second step gives $X = \hat{X}, U = \hat{U}$. The solution to AOLORP with unknown R_s, L_s, ϕ_{pm} can be constructed with $X^* = \hat{X}, U^* = U_j, K^* = K_j$.

Proof: We need to verify that (X^*, U^*) satisfies the regulation equations of the original AOLORP; and then show K^* is the optimal feedback control of the tracking error dynamics. We know X_j satisfies the regulation equations with D, F given by (15), and $\hat{X}$ satisfies the regulation equations with D, F given by (8). One can readily verify that

$$A(X_j + \hat{X}) + B(U_j + \hat{U}) + D = 0$$
$$C(X_j + \hat{X}) + F = 0,$$

where $X_j = 0, U_j = 0$.

Let $\bar{x} = x - X^* w, \bar{u} = u - U^* w$. We have

$$\dot{\bar{x}} = A\bar{x} + B\bar{u}, \quad e = C\bar{x}.$$

The optimal state feedback control $\bar{u} = -K^*\bar{x}$ satisfies the ARE, the solution of which is indeed solved by data-driven policy iteration in the first step as K_j. Therefore we have the overall optimal control policy

$$u = -K^* x + (K^* X^* + U^*)w.$$

This completes the proof. ■

V. Simulation

Simulation compares closed-loop systems result from three distinct control policies: optimal LQR control with exact model knowledge, Alg. 1, and Alg. 2, where ϕ_{pm} is assumed known. Simulation is based on Matlab®2020b.

TABLE II
PARAMETER VALUES

Notation	Values	Notation	Values
R_s	0.439 Ω	ω^*	10 rad/sec
L_s	0.0615 H	ϕ^*	0.5 Web
J	0.0163 Kgm^2	T_l^*	10 Nm

Motor model parameter values, references, and controller gains are provided in Table II. The system matrices A, B, C, D, E, F can be readily obtained. With the full knowledge of model parameters, the system equilibrium (x_e, u_e) is determined as $x_e = [0, 7.2464]^\top, u_e = [-8.7101, 12.3812]^\top$. Matrices Q and R in the cost function (5) are selected $Q = \text{diag}(10^4, 10^4)$ and $R = \mathbf{I}_2$. The LQR control is designed based on e-dynamics (4). We have $u_{LQR} = u_e - K^* e$ and the resultant value function is $V(e) = e^\top P^* e$. Performing LQR design yields the optimal feedback gain and the quadratic value function matrix

$$K^* = \begin{bmatrix} 31.1868 & 0 \\ 0 & 31.1868 \end{bmatrix}, \quad P^* = \begin{bmatrix} 1.8743 & 0 \\ 0 & 1.8743 \end{bmatrix}.$$

By solving LORP, we obtain the exact solution

$$X^* = \begin{bmatrix} 0 & 0 \\ 0 & 0.7246 \end{bmatrix}, \quad U^* = \begin{bmatrix} 0.0028 & -0.8660 \\ 0.9999 & 0.3181 \end{bmatrix}.$$

Based on matrices $A, B, D, K^*, P^*, X^*, U^*$, the values of parameters Θ corresponding to the optimal output regulation are given below

$$\Theta^* = [1.8743, 1.8743, 0, 31.1868, 0, 0, 31.1868, 0, 27.1642, -31.1868, -9.9210]^\top.$$

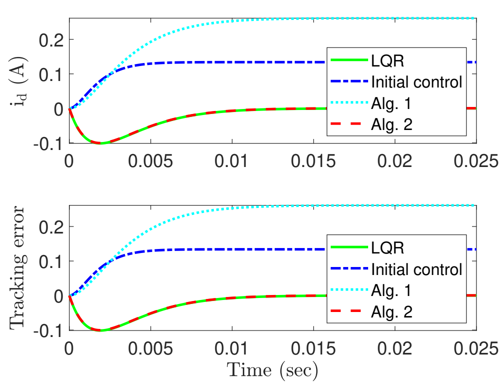

Fig. 1. d-axis current trajectories.

We apply the proposed solutions to learn Θ^* based on knowledge of C, E, F and x, w. During the implementation of Alg. 1, two perturbation signals $\rho_1(t), \rho_2(t)$, uniformly distributed over $[-1, 1]$, are injected into control inputs u_d and u_q, respectively. For Alg. 2, we additionally inject the following signal on top of the reference w:

$$w_d = \left[\sin(1000t); 0.1\cos(3000t)\right]^\top.$$

Both AOLORP algorithms start with an initial control policy

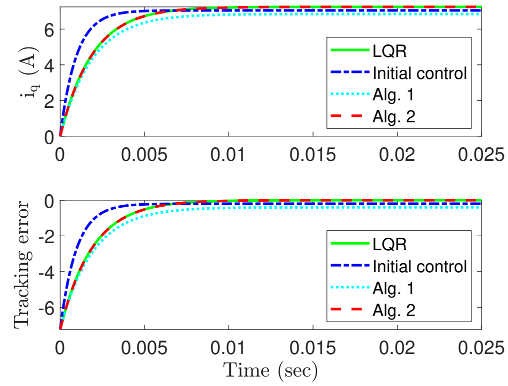

Fig. 2. q-axis current trajectories.

$u_0 = -K_0 \bar{x} + U_0 w$ where

$$K_0 = \begin{bmatrix} 20\pi & 0 \\ 0 & 20\pi \end{bmatrix}, \quad U_0 = \mathbf{0}.$$

We select an initial condition $x(0) = [0, 0]^\top$ in simulation. The sampling period is 10^{-5}sec. Each iteration of learning Θ uses an episode which restarts and runs the system for 0.005sec. The results are shown in Figs. 1-5. All figures adopt the following color code: green solid - LQR; blue dash - initial control; cyan dash - Alg. 1; and red dash - Alg. 2.

Fig. 1 depicts the trajectories of state i_d and tracking error e_d for four controls. One can tell that both initial control and Alg. 1 give the same steady state errors. This is because that Alg. 1 could not estimate $(AX + D)^\top P$ which corresponds to the last four elements in Θ, and thus fail to learn the feedforward gain matrix U^*. Similar phenomena can be observed in Fig. 2 which presents the trajectories of state i_q and tracking error e_q. It is interesting to notice two observations in Figs. 1-2: 1) Alg. 1, although giving steady state tracking errors, lands transients analogous to the LQR case; and 2) the initial control results in smaller tracking errors than Alg. 1. The first is due to the separation principle: the feedback control gain K^* and feedforward control gain U^* are independent from each other, and can be synthesized, independently, and thus Alg. 1 could synthesize K^*. The second reflects the fact that optimal feedback gain K^* is much lower than K_0 and thus leads to larger tracking errors.

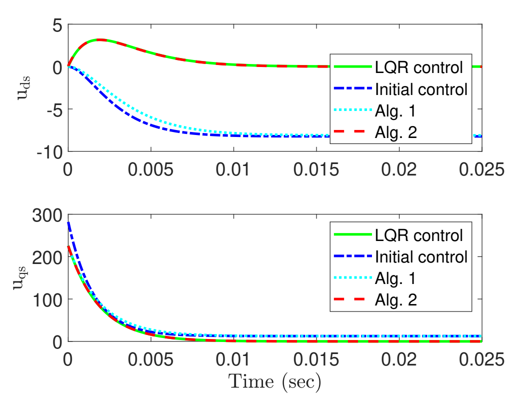

Fig. 3. Trajectories of feedback control $-K\bar{x}$.

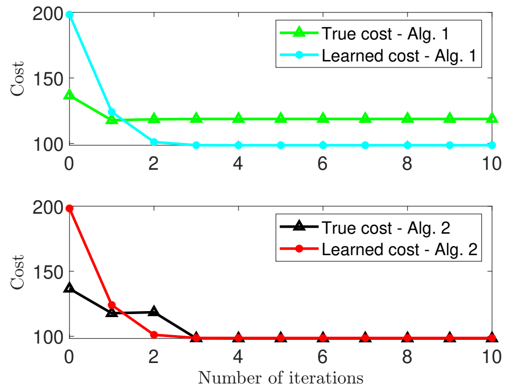

Fig. 5. True and learned cost-to-goes over iterations.

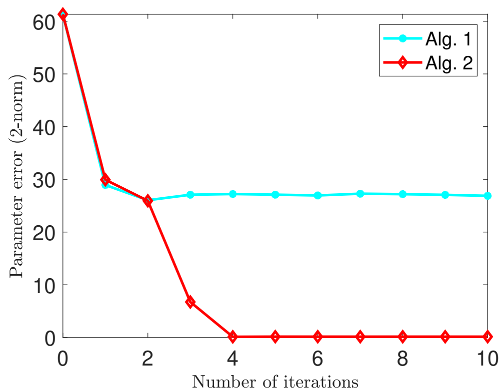

Fig. 4. 2-norm of parameter error $\Theta - \hat{\Theta}$ over iterations.

Fig. 3 plots the trajectories of feedback control portions for all four cases. For both LQR control and Alg. 2 cases, tracking error $\bar{x} = e$ converges to zero, which implies the output regulation achieved; whereas the initial control and Alg. 1 settle non-zero steady constants. Fig. 4 plots the 2-norm of parameter error vector $\Theta - \hat{\Theta}$ over iterations for both Algs. 1 and 2. Particularly, Alg. 1 stops learning after 3 iterations, in spite of the constant parameter estimation error. This is because Ψ matrix is not full-column rank. On the contrary, Alg. 2 keeps improving the parameter estimation until the parameter estimation error is close to zero. it is noteworthy that at the 3rd iteration, Alg. 2 gives less parameter error than Alg. 1 beaucase of the additional step of estimating $(AX + D)^\top P$.

Fig. 5 compares the true/learned cost-to-go from $x(0)$ corresponding to control policies that Algs. 1-2 produce. Particularly, the upper and lower plots show that the progress of cost-to-go for Algs. 1 and 2, respectively. For Alg. 1, there exists an offset between the true cost-to-go and the learned. This is because Alg. 1 could not infer the feedforward gain correctly and thus the learned value function could not predict the true value. On the other hand, Alg. 2 can learn the feedforward gain accurately, indebted to additional step of estimating $(AX + D)^\top P$, and thus the true and estimated cost-to-go values match. It is noteworthy that the learned cost-to-go for Alg. 1 matches that of Alg. 2. This confirms that Alg. 1 can learn the feedback gain and the value function, even though the feedforward gain is biased.

VI. Conclusion and Future Work

This work investigated whether ADP can facilitate self-tuning torque tracking control of SPMSM for optimal performance. The customized two-step ADP algorithm can synthesize the feedback and feedforward gain matrices separately online, in the presence of model parametric uncertainties. Our study provided a numerically promising outcome. Future work will include experiment validation, and extension to speed control mode.

References

[1] B. A. Francis and W. M. Wonham, “The internal model principle of control theory,” *Automatica*, vol. 12, no. 5, pp. 457–465, Sept. 1976.
[2] B. A. Francis, “The linear multivariable regulator problem,” *SIAM J. Control Optim.*, vol. 15, no. 3, pp. 486–505, 1977.
[3] A. Isidori and C. I. Byrnes, “Output regulation of nonlinear systems,” *IEEE Trans. Automat. Control*, vol. 35, no. 2, pp. 131–140, Feb. 1990.
[4] A. Serrani, A. Isidori, and L. Marconi, “Semi-global nonlinear output regulation with adaptive internal model,” *IEEE Trans. Automat. Control*, vol. 46, no. 8, pp. 1178–1194, 2001.
[5] J. Huang, *Nonlinear Output Regulation: Theory and Applications*. Society for Industrial and Applied Mathematics, 2004.
[6] A. J. Krener, *The Construction of Optimal Linear and Nonlinear Regulators*. Birkhauser Boston, 1992, ch. Systems, Models and Feedback: Theory and Applications, pp. 301–322.
[7] A. Saberi, A. Stoorvogel, P. Sannuti, and G. Shi, “On optimal output regulation for linear systems,” *Int. J. Control*, vol. 76, no. 4, pp. 319–333, 2003.
[8] K. S. Narendra and A. M. Annaswamy, *Stable Adaptive Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1989.
[9] M. Krstić, I. Kanellakopoulos, and P. Kokotović, *Nonlinear and Adaptive Control Design*. New York, NY: John Wiley and Sons, 1995.
[10] R. Marino and P. Tomei, *Nonlinear Control Design: Geometric, Adaptive, and Robust*. Hertfordshire, UK: Prentice-Hall, 1995.
[11] J. Lu, Q. Wei, and F.-Y. Wang, “Parallel control for optimal tracking via adaptive dynamic programming,” *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 6, pp. 1662–1674, 2020.
[12] X. Li, L. Dong, and C. Sun, “Data-based optimal tracking of autonomous nonlinear switching systems,” *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 1, pp. 227–238, 2021.
[13] W. Gao and Z.-P. Jiang, “Adaptive dynamic programming and adaptive optimal output regulation of linear systems,” *IEEE Trans. Automat. Control*, vol. 61, no. 12, pp. 4164–4169, Dec. 2016.
[14] F. L. Lewis and D. Vrabie, “Reinforcement learning and adaptive dynamic programming for feedback control,” *IEEE Circuits Syst. Mag.*, vol. 9, no. 3, pp. 40–58, Aug. 2009.
[15] D. Kleinman, “On an iterative technique for Riccati equation computations,” *IEEE Trans. Automat. Control*, vol. AC-13, no. 1, pp. 114–115, 1968.
[16] W. Gao, Y. Jiang, Z.-P. Jiang, and T. Chai, “Output feedback adaptive optimal control of interconnected systems based on robust adaptive dynamic programming,” *Automatica*, vol. 72, pp. 37–45, Oct. 2016.
[17] W. Gao, Z.-P. Jiang, F. F. Lewis, and Y. Wang, “Leader-to-formation stability of multi-agent systems: An adaptive optimal control approach,” *IEEE Trans. Automat. Control*, vol. 63, no. 10, pp. 3581–3587, Oct. 2018.
[18] J. Holtz, “Sensorless control of induction motor drives,” *Proceedings of the IEEE*, vol. 90, no. 8, pp. 1359–1394, 2002.
[19] R. Marino, P. Tomei, and C. M. Verrelli, *Induction Motor Control Design*. London, UK: Springer, 2010.
[20] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Athena Scientific Belmont, 1995.