Application of Integral Reinforcement Learning for Optimal Control of a High Speed Flux-switching Permanent Magnet Machine

Yang Yu, Xiaohe Ma, Rong Su, Tseng King Jet
Nanyang technological University
Email:yyu013@e.ntu.edu.sg

V.Viswanathan, C. J. Gajanayake
S. RamaKrishna, A. K. Gupta
Rolls-Royce Singapore Pte. Ltd

***Abstract*—A novel control method using H_∞ tracking and integral reinforcement learning is applied to a flux-switching permanent magnet (FSPM) machine in a hostile environment. The proposed controller can maintain high performance in the presence of motor parameter uncertainties and load disturbances. The conventional design procedure for an H_∞ controller is to solve the Hamilton-Jacobi-Isaacs (HJI) equation which requires full information of the system model. The novel control method, the integral reinforcement learning (IRL) makes use of neural networks to parametrically represent the control policy and the performance of system, and learns the solution of HJI equations online. Therefore, the FSPM machine can work optimally under parameter uncertainties due to different operating conditions. The simulation in Matlab/Simulink vividly illustrates the control performance for a 45kW, rotor speed 9000 rpm, 12/5 poles flux-switching permanent magnet machine.**

I. Introduction

Flux-switching permanent magnet (FSPM) machines are widely used in high speed application systems due to its prominent features such as high power density, sinusoidal back-EMF and solid rotor structure [1]-[2]. A commonly used control algorithm for FSPM machines is the vector control, in which an outer speed loop is designed based on the mechanical part of the machine and an inner current loop is designed based on the electrical dynamics. However, the conventional method cannot provide a high control performance due to inherent nonlinearity and parameter uncertainties subject to different operating conditions. In order to achieve optimal operating performance of the FSPM, the optimal control theory is normally adopted. However, the challenge of designing the optimal controller for an FSPM machine is the parameter uncertainties such as permanent magnet flux-linkage, stator inductance and resistance due to the hostile environment. Many control techniques have been proposed in the literature. For example, in [3] a robust and optimal control approach is proposed, which can deal with linear systems with uncertainties. However, it requires to know the range of the parameter variations in advance. Adaptive control is another choice to deal with parameter uncertainties, which, however, is not optimal but only minimizing a cost function of the output error. Besides, it has a low adaption for fast variations of machine parameters.

Recently, a new control approach called Integral Reinforcement Learning (IRL) has been gaining attentions, which enables the controller to learn the optimal control policy online without full knowledge of system dynamics. The application of IRL has been extensively used to solve the optimal H_2 and H_∞ regulation problems [4]-[5], and has been successfully applied to applications related to electrical engineering. The application of optimal controller design using IRL for automatic voltage regulator of power systems is firstly introduced by [6], which makes the system partially model free. In this paper, an advanced IRL technique called off-policy IRL introduced in [7]-[8] is adopted for the FSPM machine control.

This paper is organized as follows. In Section II a typical mathematical model of an FSPM machine is presented, followed by a general H_∞ control problem formulation. Application of the IRL approach to solve that control problem is described in Section III, and substantial simulation results for a linearized model and a nonlinear model are shown in section IV. Conclusions are drawn in section V.

II. Problem Formulation

A. Mathematical model of flux-switching machine

The mathematical model of a flux-switching machine in a rotating reference frame can be expressed as follows [3]-[4]:

$$\begin{bmatrix} u_d \\ u_q \end{bmatrix} = \begin{bmatrix} R_s & \\ & R_s \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} -p\omega\lambda_q \\ p\omega\lambda_d \end{bmatrix} + \begin{bmatrix} \dot{\lambda}_d \\ \dot{\lambda}_q \end{bmatrix} \tag{1}$$

$$\begin{bmatrix} \lambda_d \\ \lambda_q \end{bmatrix} = \begin{bmatrix} L_d & \\ & L_q \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} \lambda_m \\ 0 \end{bmatrix} \tag{2}$$

$$T = \frac{3}{2} p i_q [\lambda_m + (L_d - L_q) i_d] \tag{3}$$

$$J\frac{d\omega}{dt} = T - T_L \tag{4}$$

where $u_d, u_q, i_d, i_q, L_d, L_q$ are the d-axis and q-axis voltage, current and synchronous inductance respectively, $\lambda_d, \lambda_q, \lambda_m$ the d-axis, q-axis and permanent magnet flux-linkage, R_s the armature winding resistance, T, T_L, ω the electromagnetic torque, load torque and mechanical speed of the machine respectively, and p the number of rotor poles.

B. H_∞ controller design problem

We adopt a general affine nonlinear system dynamics, and formulate the electrical machine control problem an an H_∞ tracking problem, aiming to achieve a zero steady-state tracking error. To this end, a new state variable is introduced as the integral of the tracking error denoted by $q = \int(\omega_{ref} - \omega)dt$. The system dynamics is described by equations (5)-(7) below:

$$\dot{x} = f(x) + g(x)u + k(x)d \tag{5}$$

$$f(x) = \begin{bmatrix} \frac{pL_q}{L_d}\omega i_q - \frac{R}{L_d} i_d \\ \frac{-p\lambda_m}{L_q}\omega - \frac{pL_d}{L_q}\omega i_d - \frac{R}{L_q} i_q \\ \frac{3p}{2J}[\lambda_m i_q + (L_d - L_q) i_d i_q] \\ -\omega \end{bmatrix} \tag{6}$$

$$g(x) = \begin{bmatrix} \frac{1}{L_d} & 0 \\ 0 & \frac{1}{L_q} \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, k(x) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -\frac{1}{J} & 0 \\ 0 & 1 \end{bmatrix} \tag{7}$$

where $x = [\ i_d \quad i_q \quad \omega \quad q\ ]^T$, $u = [\ u_d \quad u_q\ ]^T$ $d = [\ T_L \quad \omega_{ref}\ ]^T$. The above formulation considers the reference speed and load toruqe disturbance as known disturbances of the system. In order to achieve disturbance attenuation, the cost function over an infinite horizon is defined below:

$$V(t) = \int_t^\infty (x^T Q x + u^T R u - \gamma^2 d^T d) d\tau. \tag{8}$$

Noted that γ should be chosen large enough to satisfy the bounded L_2 gain. Our **nonlinear optimal control problem** is to *find the optimal control policy* u^* *to minimize the cost* $J = V(0)$ *subject to the system dynamics (5)-(7).*

To solve the optimal control problem, we define the following Hamiltonian, where V_x is the derivative of V over $x(t)$,

$$H(u,d) = x^T Q x + u^T R u - \gamma^2 d^T d + V_x^T (f + gu + kd). \tag{9}$$

The optimal control and disturbance satisfy the stationary conditions of $\frac{\partial H}{\partial u^*} = \frac{\partial H}{\partial d^*} = 0$, which leads to

$$\begin{aligned} u^* &= -\tfrac{1}{2} R^{-1} g^T V_x^*, \\ d^* &= \tfrac{1}{2\gamma^2} k^T V_x^*. \end{aligned} \tag{10}$$

On the other hand, by using the Bellman dynamic programming concept with the value function $V(t)$ defined in (8), which can be rewritten as a functional $V(x(t),t)$, we can derive the well known Hamilton-Jacobi-Isaacs (HJI) equation, from which we shall get $H(u^*, d^*) = 0$, where u^* and d^* are obtained in (10). If we know the function g and V_x, e.g., in the case of time invariant linear dynamic system $V(x(t),t) = x(t)^T P x(t), V_x = 2Px$, where P is obtained by solving a Riccati equation, we immediately know the corresponding control law. Nevertheless, in FSPM control the function $g(t)$ cannot be accurately available and the nonlinearity of the system dynamics makes it impossible to derive the value function $V(x(t),t)$ analytically. How to derive a proper optimal control law becomes a challenge.

III. Optimal Adaptive Control Algorithm Using Off-Policy Integral Reinforcement Learning

The key idea of off-policy IRL is to avoid using function g and approximate V_x by some known base functions, when using the optimal control law specified in (10). Fig. 1 depicts a block diagram of the electrical machine control system, where the blue region contains the conventional control system, which is online adaptively tuned by the off-policy IRL algorithm shown in the pink region.

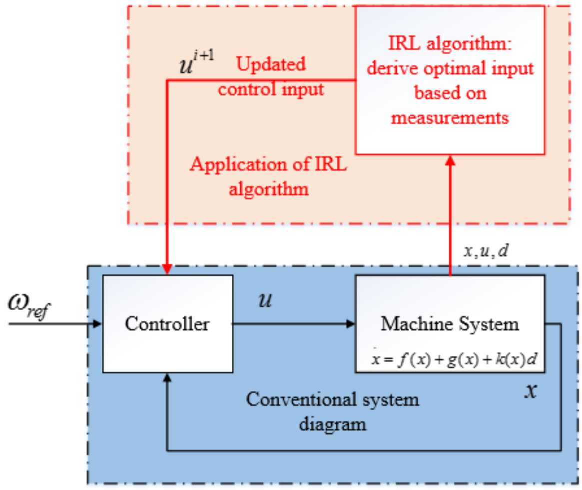

Fig. 1. Block diagram of the proposed system

In the nonlinear case, the Hamilton-Jacobi-Isaacs equation (HJI) is hard to solve directly .However we can solve the HJI equation iteratively as shown in the on-policy Reinforcement Learning (RL) algorithm [8].

On-policy RL algorithm:

1) Apply control and disturbance law u^i , d^i at i-th iteration to the system, find V^{i+1} using the Bellman equation in (11).

$$\begin{aligned} H(V_{i+1}) = x^T Q x + (V_x^{i+1})^T (f + g u^i + k d^i) + \\ {u^i}^T R u^i - \gamma^2 (d^i)^T d^i = 0 \end{aligned} \tag{11}$$

2) Update the control law and disturbance law in (12)

$$\begin{aligned} u^{i+1} &= -\tfrac{1}{2} R^{-1} g^T V_x^{i+1}, \\ d^{i+1} &= \tfrac{1}{2\gamma^2} k^T V_x^{i+1}. \end{aligned} \tag{12}$$

3) Let $i = i + 1$ and Go back to 1) until convergence.

The iteration on on-policy RL algorithm is equal to Newton's iterative sequence and convergence can be guaranteed [10]. However, it requires exact knowledge of the system dynamic which makes it not suitable for our application since the system dynamic is not accurately available. Therefore, the off-policy IRL is induced from the on-policy RL algorithm which does not require any knowledge of the system dynamics. Consider the system dynamic in equation (13):

$$\dot{x} = f(x) + g(x)(u - u^i + u^i) + k(x)(d - d^i + d^i) \tag{13}$$

where u is the pre-chosen control law, d is the actual disturbance and u^i, d^i are the control law and disturbance law to be updated. Differentiating V^{i+1} along the system dynamic in (13), yield equation (14).

$$\begin{array}{c}\dot{V}^{i+1} = (V_x^{i+1})^T(f+gu+kd) = (V_x^{i+1})^T(f+gu^i+kd^i) \\ +(V_x^{i+1})^T g(u-u^i) + (V_x^{i+1})^T k(d-d^i)\end{array} \tag{14}$$

Substituting equation (11) to (12) into equation (14) and integrating both sides from t to t+T yields equation (15).

$$\begin{array}{l}V^{i+1}(x(t+T)) - V^{i+1}(x(t)) \\ = -\int\limits_{t}^{t+T} (x^TQx + (u^i)^T Ru^i - \gamma^2 (d^i)^T d^i) d\tau - \\ \int\limits_{t}^{t+T} 2(u^{i+1})^T R(u-u^i)d\tau + \int\limits_{t}^{t+T} 2(d^{i+1})^T \gamma^2 (d-d^i)d\tau\end{array} \tag{15}$$

The off-policy Bellman equation (15) is derived based on the same value function (11) and the same updated control law in (12), therefore the off-algorithm shown below [8] shares the same convergence property as the on-policy RL algorithm.

Off-policy IRL algorithm

Step 1: Apply a pre-chosen fixed control policy u to the system and collect required system information about the state x, control input u and disturbance d at N different sampling interval during t_{j-1} to t_j ($j=[1,2,...,N]$).

Step 2: With the collected data of $x(t)$, $u(t)$ and $d(t)$ of N samples, and the optimal control policy , disturbance policy u^i, d^i at the i-th iteration, solve the following Bellman equation (16) for the cost function V^{i+1}, optimal control policy u^{i+1} and disturbance policy d^{i+1} at the $(i+1)$-th iteration.

$$\begin{array}{l}V^{i+1}(x(t_j)) - V^{i+1}(x(t_{j-1})) \\ = -\int\limits_{t_{j-1}}^{t_j} (x^TQx + (u^i)^T Ru^i - \gamma^2 (d^i)^T d^i) d\tau - \\ \int\limits_{t_{j-1}}^{t_j} 2(u^{i+1})^T R(u-u^i)d\tau + \int\limits_{t_{j-1}}^{t_j} 2(d^{i+1})^T \gamma^2 (d-d^i)d\tau\end{array} \tag{16}$$

where j is the time index and i is the iteration index

Step 3: Let $i=i+1$ and go back to step 2 until convergence

Based on the well-known high order Weierstrass approximation theorem, any smooth continuous function can be represented by a linearly independent basis function set. Therefore three neural networks (NN), i.e. the critic NN, the controller NN and the disturber NN are used to approximate the solution $V^{i+1}(x), u^{i+1}, d^{i+1}$ of the Bellman equation (16).

$$\begin{array}{r}V^{i+1}(x) = W_1^T\varphi_1(x) + \varepsilon_1 \\ u^{i+1} = W_2^T\varphi_2(x) + \varepsilon_2 \\ d^{i+1} = W_3^T\varphi_3(x) + \varepsilon_3\end{array} \tag{17}$$

Where $\varphi_1 \in R^{N_1\times 1}, \varphi_2 \in R^{N_2\times 1}$,$\varphi_3 \in R^{N_3\times 1}$ provide the base function vectors. Since V^{i+1} is scalar , $u^{i+1} \in R^{2\times 1}$ and $d^{i+1} \in R^{2\times 1}$, so the weight vectors $W_1 \in R^{N_1\times 1}, W_2 \in R^{N_2\times 2}$, $W_3 \in R^{N_3\times 2}$. N_1, N_2, N_3 are the number of neurons for each NN , $\varepsilon_1, \varepsilon_2, \varepsilon_3$ are the approximation errors using NN.

Substituting equation (17) into the original Bellman equation (16) yields equation (18). The weight vectors W_1, W_2, W_3 can be obtained by solving equation (18) as long as sufficient data are collected during the step 1 in this algorithm.

$$\begin{array}{l}W_1^T\varphi_1(x(t_j)) - W_1^T\varphi_1(x(t_{j-1})) = \\ e(t_j) - \int\limits_{t_{j-1}}^{t_j} (x^T(\tau)Qx(\tau) + u^T(\tau)Ru(\tau) - \gamma^2 d^Td)d\tau \\ -\int\limits_{t_{j-1}}^{t_j} 2\varphi_2^T(x(\tau))W_2R(u-u^i)d\tau \\ +\int\limits_{t_{j-1}}^{t_j} 2\varphi_3^T(x(\tau))W_3\gamma^2(d-d^i)d\tau\end{array} \tag{18}$$

$e(t_j)$ is the Bellman approximation error is due to $\varepsilon_1, \varepsilon_2, \varepsilon_3$.

Utilizing the actual control information u, d and optimal control policy and disturbance policy u^i, d^i from the previous iteration,the weight matrix such as W_1, W_2, W_3 can be easily obtained via least mean square method.

Define $v^1 = [\; v_1^1 \quad v_2^1 \;]^T = u - u^i$, $v^2 = [\; v_1^2 \quad v_2^2 \;]^T = d - d^i$ and assume R is diagonal matrix with each entry equal to r, therefore equation (19) is yielded.

$$\begin{array}{l}e(t_j) = W_1^T\varphi_1(x(t_j)) - W_1^T\varphi_1(x(t_{j-1})) \\ -\int\limits_{t_{j-1}}^{t_j} (-x^TQx - (u^i)^TRu^i + \gamma^2(d^i)^Td^i)d\tau \\ +2r\sum\limits_{l=1}^{2}\int\limits_{t_{j-1}}^{t_j} W_{2,l}^T\varphi_2(x(\tau))v_l^1 d\tau \\ -2\gamma^2\sum\limits_{l=1}^{2}\int\limits_{t_{j-1}}^{t_j} W_{3,l}^T\varphi_3(x(\tau))v_l^2 d\tau\end{array} \tag{19}$$

Where $W_{2,1}, W_{2,2}, W_{3,1}, W_{3,2}$ are the first and second column vector in W_2, W_3 respectively.

A standard formulation of least mean square problem is demonstrated in equation (20).

$$\begin{array}{l}y(t_j) + e(t_j) = W^Th(t_j) \\ W \in R^{(N_1+2N_2+2N_3)\times 1} \\ = [\; (W_1)^T \quad (W_{2,1})^T \quad (W_{2,2})^T \quad (W_{3,1})^T \quad (W_{3,2})^T \;]^T \\ h(t_j) = \begin{bmatrix} \varphi_1(x(t_j)) - \varphi_1(x(t_{j-1})) \\ 2r\int\limits_{t_{j-1}}^{t_j} \varphi_2(x(\tau))(v_1^1)d\tau \\ 2r\int\limits_{t_{j-1}}^{t_j} \varphi_2(x(\tau))(v_2^1)d\tau \\ -2\gamma^2\int\limits_{t_{j-1}}^{t_j} \varphi_3(x(\tau))(v_1^2)d\tau \\ -2\gamma^2\int\limits_{t_{j-1}}^{t_j} \varphi_3(x(\tau))(v_2^2)d\tau \end{bmatrix} \\ \in R^{(N_1+2N_2+2N_3)\times 1} \\ y(t_j) = \int\limits_{t_{j-1}}^{t_j} (-x^TQx - (u^i)^TRu^i + \gamma^2(d^i)^Td^i)d\tau\end{array} \tag{20}$$

Noted that there are totally $N_1 + 2N_2 + 2N_3$ independent elements in the weight vector W. Therefore in order to solve the linear equation (20) and minimize the Bellman approximation error, at least N samples should be collected, N should satisfy $N \geq N_1 + 2N_2 + 2N_3$ to make sure the LMS problem well-posed.

By collected $H = [\ h(t_1) \ \ \ldots \ \ h(t_N)\ ]$ and $Y = [\ y(t_1) \ \ \ldots \ \ y(t_N)\ ]^T$, the weight vector W based on least method square method is estimated to be

$$W = (HH^T)^{-1}HY \tag{21}$$

The flow-chart of proposed algorithm is shown in Fig.2.

Remark 1: In order to guarantee that the inverse HH^T exists, probing noise is injected into the system and the sampling time $T_j = t_j - t_{j-1}$ can be varied.

This novel algorithm works well for the linear system. This is because there is no approximation error $e(t_j)$ in equation (21) for linear systems [7]. However for nonlinear case, the approximation error always exists. Based on the formulation of the least-mean square problem in equation (21) , the choose of the basis function sets $\varphi_1, \varphi_2, \varphi_3$ has significant impacts on the Bellman approximation error $e(t_j)$. How to choose a suitable basis function also becomes a challenge.

The selection of $\varphi_1, \varphi_2, \varphi_3$ is usually a natural choice guided by engineering experience and intuition. Moreover, The numbers of neurons such as N_1, N_2, N_3 should be sufficient high to achieve good approximation and the convergence of this algorithm can be therefore guaranteed. In order to investigate the effect of the choose of basis functions on the IRL algorithm, an IRL controller with multiple basis function sets is presented in details in the nonlinear case study in section IV.

IV. Simulation Result

Based on the description of the IRL algorithm in section III, an IRL controller is implemented and tested in a linear case at first and a nonlinear case would be studied later. The parameter values used to specify a high speed FSPM machine are given in Table I.

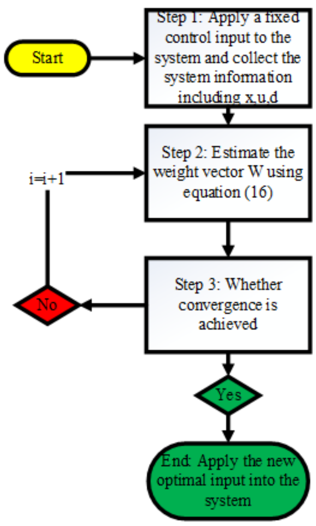

Fig. 2. Flow charts for the IRL algorithm

TABLE I
PARAMETERS OF FSPM SYSTEM

Symbol	Quantity	Value
P_{rated}	Rated power	45kW
ω	Rated mechanical angular speed	942 rad/s
N_r	Number of rotor pole	5
N_s	Number of stator pole	12
m	Number of phases	3
R_s	Stator resistance	1 mΩ
L_d	direct-axis inductance	0.8 mH
L_q	quadrature-axis inductance	0.8 mH
λ_m	permanent magnet flux-linkage	0.025Wb

A. Linear case study

In this linear case study, the machine model is linearized at nominal condition. The optimal control policy obtained from the proposed method is then compared with the solution of the game Riccati equation which are off-line solved . The case study is conducted to show that the proposed method can learn the solution of Riccati equation without knowing the system dynamics.

The linearized model as well as some user-defined variables are expressed in the equation(22).

$$\begin{aligned}
&\dot{x} = Ax + g(x)u + k(x)d \\
&A = \begin{bmatrix} -1.25 & 4712.3 & 0 & 0 \\ -4712.3 & -1.25 & -156.25 & 0 \\ 0 & 23.4375 & 0 & 0 \\ 0 & 0 & -1 & 0 \end{bmatrix} \\
&g = \begin{bmatrix} 1250 & 0 \\ 0 & 1250 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, k = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -125 & 0 \\ 0 & 1 \end{bmatrix} \\
&Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 100 \end{bmatrix}, R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \gamma = 30
\end{aligned} \tag{22}$$

The optimal control policy based on the game algebraic Riccati equation (GARE) $Q + A^TP + PA - PBR^{-1}B^TP + \frac{1}{\gamma^2}PDD^TP = 0$ are shown in equation (23).

$$u^* = -\begin{bmatrix} 1.01 & 0 & -3.15 & 13.45 \\ 0 & 1 & 0.84 & -3.59 \end{bmatrix} x \tag{23}$$

The simulation results for the optimal control policy are shown in Fig.3 The flux-switching machine accelerates to the reference speed $940 rad/s$ within 1 second. The disturbance load torque $10N * m$ is applied to the machine as a step increase at 1 second and a step decrease at 2 seconds respectively. It is obvious that the proposed H_∞ integral controller can track the reference speed ω_{ref} regardless of

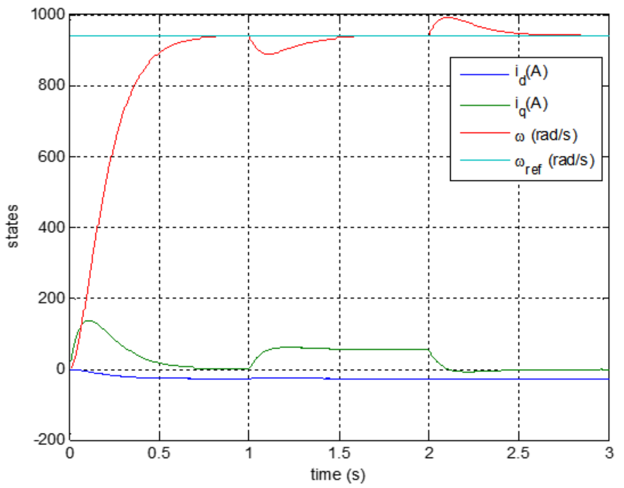

Fig. 3. Simulation results for the off-line designed controller in linear case study

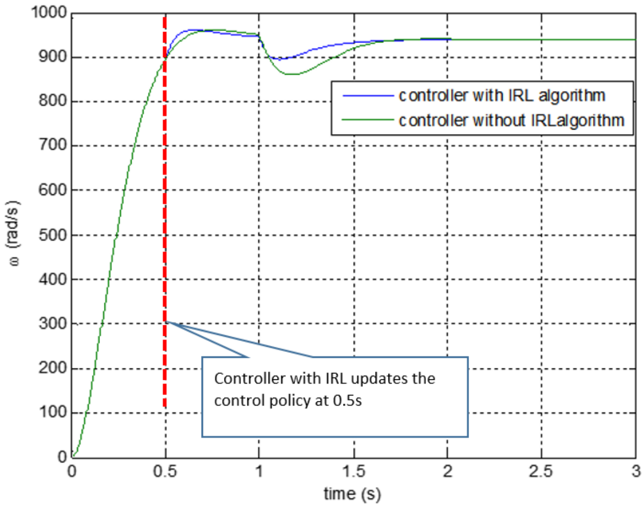

Fig. 4. Simulation results for control with IRL in linear case study

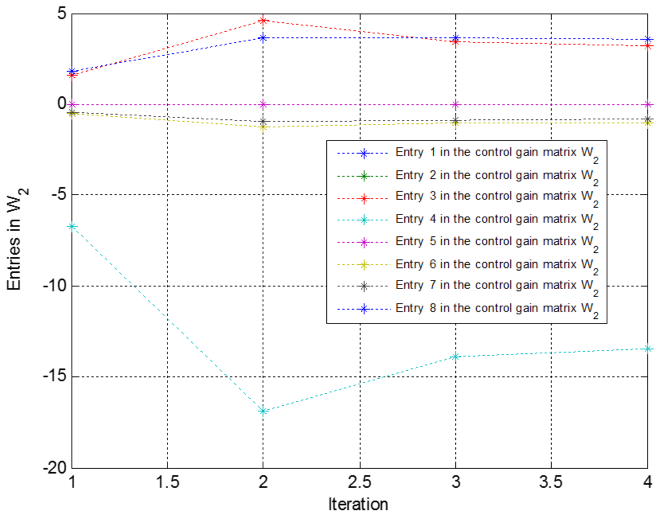

Fig. 5. Convergence of W_2 in IRL

the load variation. The simulation results might be improved by assigning different values to the Q, R matrix.

The off-line designed H_∞ controller is robust to some extend. However, the parameter variations could still affect the system performance and disturbance attenuation level. Therefore the off-policy integral reinforcement learning is then implemented with the aim to solve the GARE without any knowledge of system dynamic.

For a linear system, [7] proves that the basis function set for the cost function $V(x)$ is the quadratic polynomial associated with the states x. The basis function for the optimal control policy are the state vectors. Before implementing the algorithm, the basis function such as $\varphi_1, \varphi_2, \varphi_3$ are predetermined in equation (24).

$$\begin{aligned} \varphi_1 &= [x_i(t)x_j(t)]_{i=1:n;j=i,n} \\ &= [\ x_1^2 \quad x_1x_2 \quad x_1x_3 \quad \ldots \quad x_4^2\ ]^T \\ \varphi_2 &= [\ x_1 \quad x_2 \quad x_3 \quad x_4\ ]^T \\ \varphi_3 &= [\ x_1 \quad x_2 \quad x_3 \quad x_4\ ]^T \end{aligned} \tag{24}$$

The sampling time T is set to be 0.005 second. The sampling time can also be a user-defined value since the sampling time T generally does not affect the convergence of proposed method in the linear case.

For the simulation of control policy using IRL, the initial control policy u_0 in equation (25) is poorly designed .

$$u_0 = -\begin{bmatrix} 0.51 & 0 & -1.578 & 6.72 \\ 0 & 0.5 & 0.42 & -1.8 \end{bmatrix} x \tag{25}$$

As shown in Fig. 4, from t=0 to t=0.5 second, the controllers with and without IRL algorithm share the same initial control policy u_0. The controller with IRL algorithm collected the system information during 0 to 0.5 second and the convergence of the control gains W_2 is shown in Fig. 5. Beginning from the initial control gains u_0 in equation (25), the IRL controller finally converges to the optimal control law shown in equation (26) which is then applied to the machine at t=0.5 second. The load torque increase is applied to the system at t=1 second to investigate the system performance under load torque variations.

$$u^* = -\begin{bmatrix} 0.94 & 0 & -3.15 & 13.46 \\ 0 & 0.91 & 0.84 & -3.60 \end{bmatrix} x \tag{26}$$

Obviously the controller with IRL algorithm achieves a better control performance since the IRL algorithm solves the GARE on line.

B. Nonlinear case study

In this section, the IRL method is applied to the original nonlinear model. Due to the nonlinearity, unlike perfect representation for the value function V^{i+1} , updated control law u^{i+1} and disturbance law d^{i+1} in the linear case, there is indeed Bellman approximation error in equation (20)

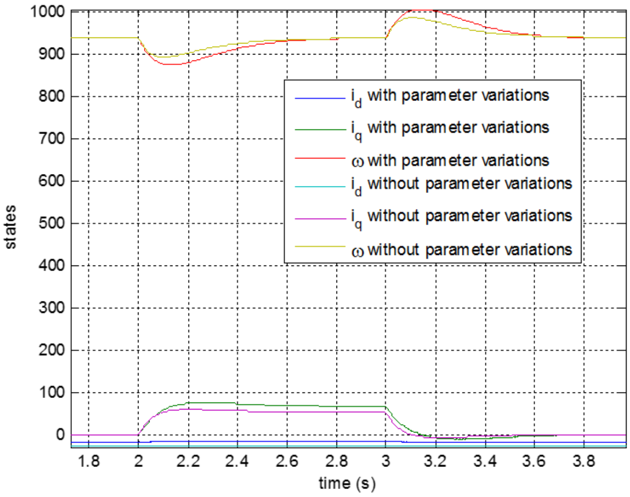

Fig. 6. System performance with and without parameter variations

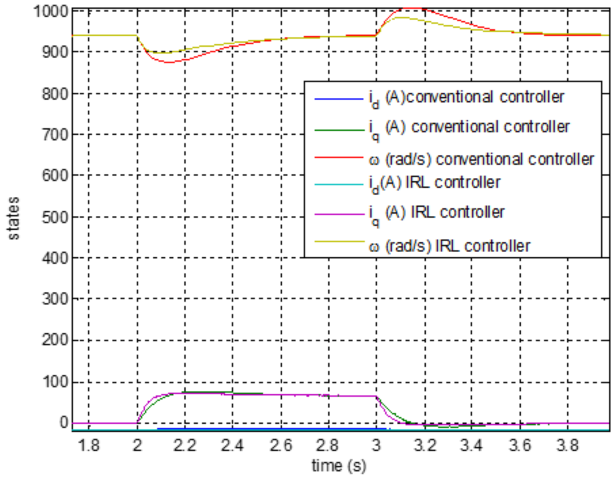

Fig. 7. System performance with conventional controller and IRL controller under parameter variations

due to neural network approximation in the nonlinear case study. Without loss of generality, the same basis function sets $\varphi_1, \varphi_2, \varphi_3$ used for the linear case study in equation (24) are adopted for the nonlinear model. Now consider the machine model when parameter variation happens and assume $L_d = L_q = 1mH, R_s = 2m\Omega, \lambda_m = 0.02Wb$. The controller u^* designed based on the nominal machine model is already given in equation (23).

The same policy u^* is applied to the machine with and without parameter variations as shown in Fig 6 . The machine reaches its reference speed before t=1.8 second. A step-change load torque 10N*M is applied to and removed from the machine at t=2 seconds and 3 seconds respectively.

The system performance under parameter variations is shown in Fig. 6. It's clear that the conventional controller can not achieve a good speed tracking performance under parameter variations.

However, IRL mechanism learns the optimal solution online based on the available measurements from the system. The optimal control policy in IRL algorithm converges to u_3^* in equation (30).

$$u_3^* = -\begin{bmatrix} 1 & 0 & -5.67 & 19.05 \\ 0 & 1 & 1.21 & -4.2 \end{bmatrix} x \qquad (27)$$

The system performance under conventional controller u^* and control policy u_3^* generated by IRL algorithm is given in Fig. 7.

Due to the parameter variations, the optimal control policy which are off-line designed in a conventional manner can no longer maintain the system optimality as defined in equation (8). The IRL controller learns the solution of HJI equation in a real time manner. Therefore the proposed controller gives a better control performance than the conventional one.

V. Conclusion

In this paper, the IRL method is applied to the nonlinear optimal control problem of electrical machine. The Full knowledge of the machine parameters with a sufficient accuracy is no longer required for this nonlinear optimal controller.

Acknowledgment

This work is conducted within the Roll-Royce@NTU Corporate Lab with the support of RR@NTU EPACS E-RT4.1 from the National Research Foundation (NRF) Singapore under the Corp-lab@Univeristy Scheme.

References

[1] Z. Zhu, Y. Pang, D. Howe, S. Iwasaki, R. Deodhar, and A. Pride, "Analysis of electromagnetic performance of flux-switching permanent-magnet machines by nonlinear adaptive lumped parameter magnetic circuit model," *IEEE Transactions on magnetics*, vol. 41, no. 11, pp. 4277–4287, 2005.

[2] J. Zhao, M. Lin, L. Jin, and W. Zhang, "Vector control study of axial field flux-switching permanent magnet machine," in *Electrical Machines and Systems (ICEMS), 2014 17th International Conference on*. IEEE, 2014, pp. 1091–1095.

[3] L. Xie, "Output feedback h control of systems with parameter uncertainty," *International Journal of control*, vol. 63, no. 4, pp. 741–750, 1996.

[4] D. Vrabie, O. Pastravanu, M. Abu-Khalaf, and F. Lewis, "Adaptive optimal control for continuous-time linear systems based on policy iteration," *Automatica*, vol. 45, no. 2, pp. 477 – 484, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0005109808004469

[5] H. Li, D. Liu, and D. Wang, "Integral reinforcement learning for linear continuous-time zero-sum games with completely unknown dynamics," *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 3, pp. 706–714, 2014.

[6] L. B. Prasad, H. O. Gupta, and B. Tyagi, "Application of policy iteration technique based adaptive optimal control design for automatic voltage regulator of power system," *International Journal of Electrical Power & Energy Systems*, vol. 63, pp. 940–949, 2014.

[7] B. Luo, H.-N. Wu, and T. Huang, "Off-policy reinforcement learning for control design," *IEEE transactions on cybernetics*, vol. 45, no. 1, pp. 65–76, 2015.

[8] H. Modares, F. L. Lewis, and Z. P. Jiang, "H tracking control of completely unknown continuous-time systems via off-policy reinforcement learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 10, pp. 2550–2562, Oct 2015.