2E4-2

Model-free Reinforcement Learning for Speed Control of Ultrasonic Motors

Abdullah Mustafa[1 ‡], Tatsuki Sasamura[1], Tokuo Sashida[2], Susumu Miyake[1] and Takeshi Morita[1]
([1]Graduate School Frontier Sciences, The Univ. of Tokyo; [2]Shinsei Corp.)

1. Introduction

Ultrasonic motors(USM) have long attracted researchers due to their superior features; however, these benefits cannot be harnessed without a proper control scheme [1]. Unfortunately, common linear controllers like PID or LQR cannot be optimally applied to USM. Developing these controllers would require linearization of the USM model which is inherently nonlinear and suffers from frequency drift, hysteresis, and jumping phenomena. Nonlinear controllers like MPC, ILQR, Fuzzy controllers, and Neural Networks would require some model of the system, expert knowledge, or supervised learning. These requirements can limit the design process of the controller. Thus, there is a need for a model-free nonlinear semi-supervised controller.

2. Reinforcement Learning (RL) Control

RL is a model-free controller that can apply an optimal nonlinear control regime by interacting with the environment. If the control objective can be formulated as an MDP (Markov Decision Process) problem, RL can learn a policy (π) that maximizes the total sum of rewards (R_t) over time [2] (**Eq. 1**).

$$V^{\pi}(s) = E_{\pi}\{\textstyle\sum_{t=0}^{\infty} \gamma^t R_t \,|s_t = s\} \qquad (1)$$

There are various approaches to learning the optimum policy, out of which Q learning can be most suitable for our speed control objective. Q-learning is a model-free RL algorithm that can directly learn the mapping from states (S_t) to actions (A_t) by interacting with the environment. The Bellman optimality equation of Q-value for a state action pair $Q(S_t, A_t)$ is a sum of the immediate reward of the current state as well as the discounted reward of all future states as in **Eq. 2**. Thus, RL can solve tasks with long-time dependency. At state (S_t), a greedy action (A_t) is chosen from a set of discrete actions by maximizing $Q(S_t, a)$ over action (a) as in **Eq. 3**. In the case of continuous action space, a continuous policy network is required but with the same objective of optimizing the Q-value.

$$Q(S_t, A_t) = R_t + \gamma\, max_a(Q(S_{t+1}, a)) \quad (2)$$

$$A_t = max_a(Q(S_t, a)) \quad (3)$$

ammustafa@s.h.k.u-tokkyo.ac.jp

3. RL USM Speed Control

To apply RL to USM speed control, a few terms need to be defined. First, the RL controller (agent) should control the frequency (action) to minimize the speed error between target and measured speed. Thus, the MDP problem can be formulated as shown in **Fig. 1**. Given the current state of the system (S_t: a continuous function of USM state and target speed), the agent decides an action (A_t: a discrete or a continuous frequency update step). Accordingly, the agent receives a reward (R_t: preferably a continuous reward function that is maximum at optimal behavior). The system state is updated (S_{t+1}) and used for deciding the next action.

For USM speed control, the system state can incorporate many variables. In its simplest form, we control the USM at constant torque (no-load), preload, driving voltage amplitude, and phase. In that case, the USM speed can be simplified as a function of driving voltage frequency (f_t) and USM temperature (T_t). However, due to the hysteresis effect, the measured speed can vary even under the same frequency and temperature. Thus, we add the current USM (V_{r_t}) speed to make the system state fully observable. Finally, target speed (V_{ref_t}) should be an additional state variable as it is critical for deciding the optimal frequency action. Since our control problem is continuous in both the state space and the action space, a neural network representation is used. Out of the many Deep RL architectures, soft actor-critic (SAC) was applied due to its state-of-the-art performance [3]. SAC maximizes both expected reward, and the state entropy, so it is stable in stochastic environments.

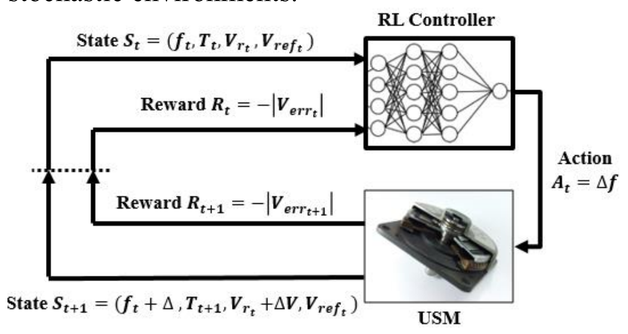
RL Controller
State $S_t = (f_t, T_t, V_{r_t}, V_{ref_t})$
Reward $R_t = -|V_{err_t}|$
Action $A_t = \Delta f$
Reward $R_{t+1} = -|V_{err_{t+1}}|$
State $S_{t+1} = (f_t + \Delta, T_{t+1}, V_{r_t} + \Delta V, V_{ref_t})$
USM

Fig. 1 MDP representation of USM speed control

4. Results

To evaluate the effectiveness of RL for USM speed control, a simplified equivalent circuit model (ECM) of USM was developed. The stator vibration amplitude (p) under given frequency (ω) was calculated by solving the angular equation in **Eq. 4**. In **Eq. 5**, speed (V_r) is directly proportional to amplitude (p) with some constant ($K = \frac{60\times\omega\times h}{(2\pi r^2)}$). Additional nonlinearity was hardcoded to simulate the pullout phenomena. If ω is lower than some threshold, V_r goes to zero. As the temperature increases, the PZT softens, and the stiffness (c) decreases. As a result, the frequency response drifts towards lower frequency as temperature increases as shown in **Fig. 2**.

$$m\ddot{p} + d\dot{p} + cp = \Theta V_0 e^{j(\omega t+\psi)} + F_N + F_T \quad (4)$$

$$V_r = Kp * (\omega > \omega_{pullout}) \quad (5)$$

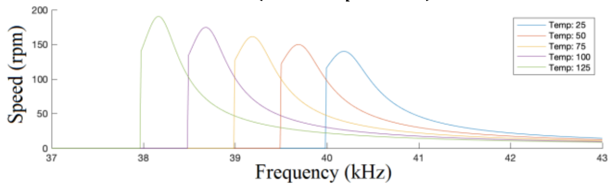

Fig. 2 Simulated USM speed response using ECM

In the MATLAB/SIMULINK environment, the USM model was developed, and combined with the SAC RL agent from the Reinforcement Learning Toolbox. The frequency action was continuous and constrained between [-2,2] kHz. The reward function was defined as the negative absolute speed error ($-|V_{err}|$). The training process was carried out for 200 episodes, each of which lasted for 30 steps (arbitrary). Each episode was initialized at a different initial frequency [35,45] kHz, and a different target speed [0,150] rpm. The learning curve is shown in **Fig. 3**. During early episodes, the episodic reward is low and fluctuating severely. Around the 50th episode, the reward started steadily increasing and nearly stabilized. During training, SAC uses stochastic actions, and thus some reward fluctuations exist. Another reason is the random initialization of frequency and target speed.

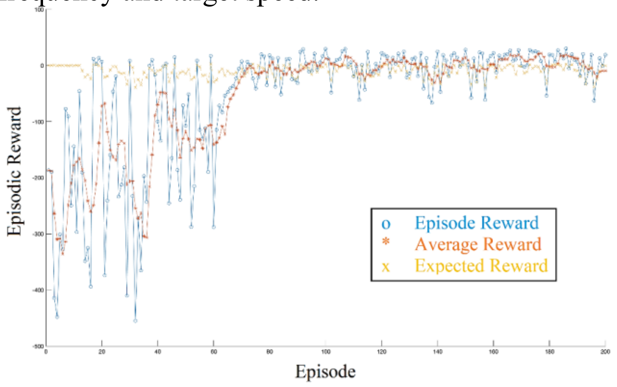

Fig. 3 Learning curve for the RL speed control

To evaluate the performance of the RL speed control in simulation, a sinusoidal target speed was tracked for an extended simulation time. The target speed was realized even with temperature rise as shown in **Fig. 4**. However, the speed tracking is imperfect, and some speed error still exists. Modifying the reward function and the network architecture can reduce this error. Experimental results of RL are shown in **Fig. 5**. Due to measurements delay and noise, the tracking has some delays as well. Such delayed response can cause the RL agent to learn wrong sequences of ($S_t \rightarrow A_t \rightarrow R_t \rightarrow S_{t+1} \rightarrow \cdots$) causing suboptimal policy. Further improvement of the experimental setup is a concern of current research.

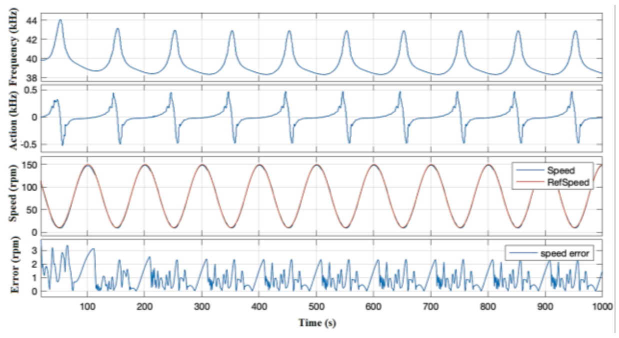

Fig. 4 Speed tracking using RL (Simulation)

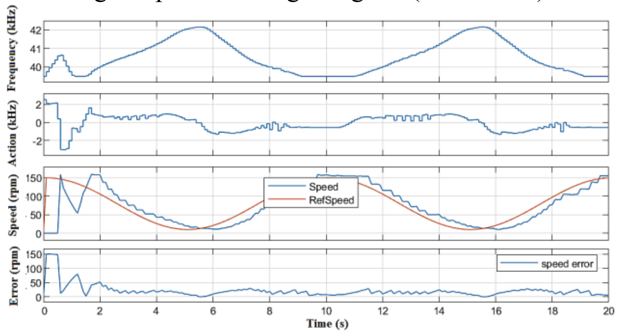

Fig. 5 Speed tracking using RL (Experiment)

5. Conclusion

In this research, RL was proposed for speed control of USM. The results showed that RL can effectively track desired speeds under changing operating conditions. Future research will focus on reward function design, network architecture optimization, hardware development, and operation under varying conditions like load torque, preload, or voltage amplitude

References

1. Y. Tomikawa and S. Ueha: Ultrasonic Motors Theory and Applications (Oxford University Press, New York, 1993)
2. R. Sutton and A. Barto: Reinforcement Learning: An Introduction (MIT Press, Cambridge, MA, 2018)
3. T. Haarnoja, A. Zhou, P. Abbeel and S. Levine: arXiv:1801.01290v2