Hamiltonian Engineering via Reinforcement Learning

Will Kaufman
Dartmouth College

Hamiltonian engineering problem

The time evolution of a quantum system depends on the system's Hamiltonian, a self-adjoint operator whose eigenvalues correspond to possible energy measurements. For a system with initial density operator $\rho(0)$ and Hamiltonian $H(t)$, the state at time t is given by

$$\rho(t) = U(t)\rho(0)U^\dagger(t)$$

where the propagator $U(t)$ is defined by

$$i\hbar\frac{dU(t)}{dt} = H(t)U(t), U(0) = \mathbb{1} \tag{1}$$

The total Hamiltonian $H(t)$ can generally be broken into a time-independent system Hamiltonian and a time-dependent control Hamiltonian

$$H(t) = H_{\text{system}} + H_{\text{control}}(t) \tag{2}$$

Hamiltonian engineering seeks to design the time-dependent Hamiltonian $H_{\text{control}}(t)$ to control the system's evolution so that it appears to evolve under a target "effective" Hamiltonian (H_{target}) when measured stroboscopically. That is,

$$\rho(Nt_c) = U_{\text{target}}(Nt_c)\rho(0)U^\dagger_{\text{target}}(Nt_c) \tag{3}$$

where U_{target} is the propagator determined by H_{target}, t_c is the cycle time, and $N \in \mathbb{N}$. If $H(t)$ determines the propagator $U(t)$, then equation 3 is implied by

$$U(Nt_c) = U_{\text{target}}(Nt_c) \tag{4}$$

Average Hamiltonian theory

Average Hamiltonian theory is one approach to solving the Hamiltonian engineering problem [1] . For a system of spins in a magnetic field, the internal Hamiltonian contains a chemical shift term and a dipolar coupling term

$$H_{\text{int}} = \sum_i \delta_i I_z^{(i)} + \sum_{i,j} d_{ij}\left(3I_z^{(i)}I_z^{(j)} - \mathbf{I^{(i)}}\cdot\mathbf{I^{(j)}}\right) = H_{\text{CS}} + H_{\text{D}} \tag{5}$$

Applying a magnetic field pulse along the transverse axis performs a global rotation of the spins. The magnetic field pulses manifest themselves as a time-dependent interaction $H_{\text{rf}}(t)$. If a pulse sequence with cycle time is both cyclic and periodic [2]

$$U_{\text{rf}}(t_c) = T\exp\left(-i/\hbar\int_0^{t_c} H_{\text{rf}}(t)dt\right) = \pm\mathbb{1} \quad \text{(cyclic)} \tag{6}$$

$$H_{\text{rf}}(t) = H_{\text{rf}}(t + Nt_c) \quad \text{(periodic)} \tag{7}$$

then, using the Magnus Expansion, the propagator can be given by

$$U(t_c) = \exp\left(\frac{-i}{\hbar}t_c(\overline{H}^{(0)} + \overline{H}^{(1)} + \dots)\right) \tag{8}$$

$$\overline{H}^{(0)} = 1/t_c\int_0^{t_c} U_{\text{rf}}(t)H_{\text{int}}U_{\text{rf}}^\dagger(t)dt \tag{9}$$

To engineer a target Hamiltonian H_t, the pulse sequence is chosen so that $\overline{H}^{(0)} = H_t$.
Several pulse sequences have been developed using the average Hamiltonian theory framework [3–5]. For example, the WAHUHA 4-pulse sequence applied to a spin system with H_{int} given in eq 5 has the zeroth-order average Hamiltonian $\overline{H}^{(0)} = \frac{1}{3}\sum_i \delta_i\left(I_x^{(i)} + I_y^{(i)} + I_z^{(i)}\right)$.
Average Hamiltonian theory assumes that higher-order terms in the Magnus Expansion (eq 8) are negligible. This is often true in the regime where $t_c|H_{\text{int}}| \ll 1$, but t_c is constrained by experimental limitations in the accuracy of pulse timings and strengths.

Reinforcement learning

Reinforcement learning (RL) is being investigated as an alternative approach to Hamiltonian engineering for spin systems. "Reinforcement learning is learning what to do–how to map situations to actions–so as to maximize a numerical reward signal" [6]. RL has been applied to a variety of different problems, including playing games such as chess or Go [7] and interacting with physical environments such as balancing a pole [8].

Reinforcement learning (cont.)

The RL paradigm involves an *agent* that makes observations on the *state* of the *environment*, performs *actions* on the environment, and receives *rewards* based on its performance (see figure 1).

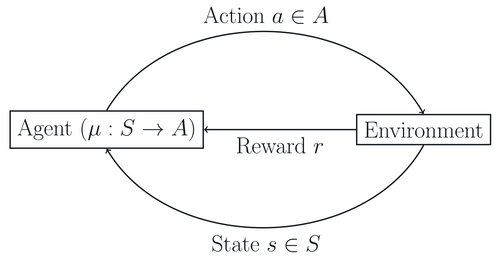

Figure 1: The general reinforcement learning paradigm.

The agent learns a policy function $\mu : S \to A$ which maps a given state to the action that will lead to the largest rewards. In many RL algorithms, an action-value function $Q_\mu(s, a)$ is also learned. The action-value function gives the expected total rewards by performing action a when in state s, then following μ.
Several algorithms have been developed for various realizations of the RL problem. Deep deterministic policy gradient (DDPG) uses neural networks to approximate the policy $\mu(s|\theta_\mu)$ and the action-value function $Q(s, a|\theta_Q)$ for continuous action spaces [8]. Both μ and Q are trained using gradient-based methods. Q is optimized by minimizing the loss function

$$L(\theta_Q) = \mathbb{E}\left[(Q(s, a|\theta_Q) - y_t)^2\right] \tag{10}$$

where

$$y_t = r(s, a) + \gamma Q(s', \mu(s')|theta_Q)$$

s' is the state after performing action a. The policy μ is optimized by maximizing the performance J (generally taken to be the action-value function Q) via gradient ascent

$$\nabla_{\theta_\mu} J = \mathbb{E}\left[\nabla_{\theta_\mu} Q(s, \mu(s|\theta_\mu))\right] \tag{11}$$

The gradients are estimated from a random subset of experiences (s, a, r, s').
In [9], a hybrid algorithm called Evolutionary Reinforcement Learning (ERL) uses both DDPG and genetic algorithms to prevent premature convergence to local optima and speed up learning. In ERL, a population of policy functions are maintained, as well as gradient-based policy and action-value functions. A diagram of the algorithm is presented in figure 2.

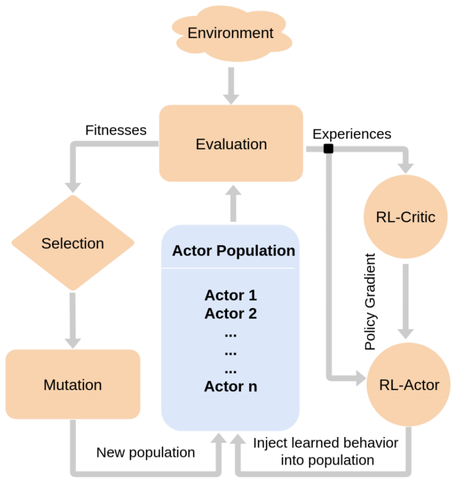

Figure 2: The general overview of the ERL algorithm, reproduced from [9].

Applying RL to Hamiltonian engineering

In the case of Hamiltonian engineering, the states correspond to the propagators of the system at different times, the actions correspond to pulses or delays, and the reward signal quantifies the degree to which the system has evolved under the target effective Hamiltonian. The fidelity of two unitary operators can be quantified by

$$\text{fidelity}(U^\dagger_{\text{target}}, U) = \left|\text{Tr}\left(\frac{U^\dagger_{\text{target}}U}{2^N}\right)\right| \tag{12}$$

and takes on values between 0 and 1, where 1 implies the two unitary operators are equal. Using this definition of fidelity, the rewards in the RL algorithm are

$$r = -\log\left(1 - \text{fidelity}(U_{\text{target}}, U_{\text{exp}})^{\tau/t}\right) \tag{13}$$

The exponent τ/t makes rewards comparable across different duration pulses sequences.
The actions are magnetic field pulses that perform global rotations, and can therefore be parametrized by the axis of rotation, the rotation angle α, and the time dt over which the rotation is performed. The axis of rotation is assumed to lie in the xy-plane, so a single pulse can be characterized by the tuple $(\phi, \alpha, dt) \in [-\pi, \pi] \times [-\alpha_{\max}, \alpha_{\max}] \times [t_{\min}, \text{inf})$. The state is represented by the sequence of all previous actions performed. The state and actions are represented by values in the interval $[-1, 1]$ according to the following map

$$\phi \in [-\pi, \pi], \phi \mapsto \phi/\pi \tag{14}$$

$$\alpha \in [-2\pi, 2\pi], \alpha \mapsto \alpha/2\pi \tag{15}$$

$$t \in [0, 5\cdot 10^{-6}], t \mapsto \frac{\log_{10}(t + 10^{-7}) + 7}{0.853785} - 1 \tag{16}$$

This encoding of action and state representations is necessary for the training and stability of the neural networks.

Results and further work

Applying ERL to Hamiltonian engineering has so far been unsuccessful. Neither the gradient-based nor population policy functions have consistently achieved rewards above 3, which corresponds to a fidelity of 0.999. For experimental applications, this is far too low a fidelity to be useful. The poor performance may be due to poor choice of hyperparameters (such as the learning rate or the number of observations from which to learn) or ineffective exploration of both state and action space.
In addition to varying hyperparameters or changing the noise processes used in ERL, reformulating the problem in terms of discrete actions (i.e. only performing $\pi/2$ rotations along the x or y axis) could be promising. This has generally been the approach within the average Hamiltonian theory framework, as well as with preliminary investigations into reinforcement learning for Hamiltonian engineering. Finally, including experimental errors or constraints, such as phase errors or finite pulse widths, would more accurately reflect the pulse sequence performances.

References

1. Haeberlen, U. & Waugh, J. S. Coherent Averaging Effects in Magnetic Resonance. *Phys. Rev.* **175,** 453–467. https://link.aps.org/doi/10.1103/PhysRev.175.453 (2 Nov. 1968).
2. Gerstein, B. & Dybowski, C. *Transient Techniques in NMR of Solids: An Introduction to Theory and Practice* 1st ed. (Academic Press, 1985).
3. Waugh, J. S., Huber, L. M. & Haeberlen, U. Approach to High-Resolution nmr in Solids. *Phys. Rev. Lett.* **20,** 180–182. https://link.aps.org/doi/10.1103/PhysRevLett.20.180 (5 Jan. 1968).
4. Choi, S., Yao, N. Y. & Lukin, M. D. Dynamical Engineering of Interactions in Qudit Ensembles. *Phys. Rev. Lett.* **119,** 183603. https://link.aps.org/doi/10.1103/PhysRevLett.119.183603 (18 Nov. 2017).
5. O'Keeffe, M. F., Horesh, L., Barry, J. F., Braje, D. A. & Chuang, I. L. Hamiltonian engineering with constrained optimization for quantum sensing and control. *New Journal of Physics* **21,** 023015. ISSN: 1367-2630. http://dx.doi.org/10.1088/1367-2630/ab00be (Feb. 2019).
6. Sutton, R. S. & Barto, A. G. *Reinforcement learning: An introduction* (MIT press, 2018).
7. Silver, D. *et al.* A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science* **362,** 1140–1144. ISSN: 0036-8075. eprint: https://science.sciencemag.org/content/362/6419/1140.full.pdf. https://science.sciencemag.org/content/362/6419/1140 (2018).
8. Lillicrap, T. P. *et al. Continuous control with deep reinforcement learning* 2015. arXiv: 1509.02971 [cs.LG].
9. Khadka, S. & Tumer, K. *Evolution-Guided Policy Gradient in Reinforcement Learning* 2018. arXiv: 1805.07917 [cs.LG].