Reinforcement Learning Informed by Optimal Control

Magnus Önnheim[1,2(✉)], Pontus Andersson[1,2], Emil Gustavsson[1,2], and Mats Jirstrand[1,2]

[1] Fraunhofer-Chalmers Centre, Chalmers Science Park, 412 88 Gothenburg, Sweden
magnus.onnheim@fcc.chalmers.se
[2] Fraunhofer Center for Machine Learning, Gothenburg, Sweden

Abstract. Model-free reinforcement learning has seen tremendous advances in the last few years, however practical applications of pure reinforcement learning are still limited by sample inefficiency and the difficulty of giving robustness and stability guarantees of the proposed agents. Given access to an expert policy, one can increase sample efficiency by in addition to learning from data, and also learn from the experts actions for safer learning.

In this paper we pose the question whether expert learning can be accelerated and stabilized if given access to a family of experts which are designed according to optimal control principles, and more specifically, linear quadratic regulators. In particular we consider the nominal model of a system as part of the action space of a reinforcement learning agent. Further, using the nominal controller, we design customized reward functions for training a reinforcement learning agent, and perform ablation studies on a set of simple benchmark problems.

Keywords: Reinforcement learning · Expert learning · Linear quadratic control · Optimal control · Adaptive control · Online learning

Consider a standard problem in optimal control where one wants to find a sequence of control signals u_t such that the following optimization is solved.

$$\min_u \mathbb{E}\left[\sum_{t=0}^{T} \ell(y_t, u_t, t)\right] \tag{1a}$$

$$x_{t+1} = f(x_t, u_t) + v_t, \quad t = 0, \ldots, T-1, \tag{1b}$$

$$y_t = g(x_t, u_t) + w_t, \quad t = 0, \ldots, T, \tag{1c}$$

This work was developed in Fraunhofer Cluster of Excellence Cognitive Internet Technologies. It has also partially been funded by the Swedish Foundation for Strategic Research.

I. V. Tetko et al. (Eds.): ICANN 2019, LNCS 11731, pp. 403–407, 2019.
https://doi.org/10.1007/978-3-030-30493-5_40

where ℓ denotes a loss function, f and g denote the system and observation dynamics, and where v_t and w_t denote system and observation noise, respectively. Further, we assume that we are presented with a *nominal* version of (1), where ℓ is a quadratic form, and $f(x,u) = Ax + Bu$, $g(x,u) = Cx$, for some matrices A, B, C, and where v_t, w_t are i.i.d. samples of zero-mean Gaussian distributions with covariance matrices V and W, respectively. In the sequel we will refer to the A, B, C-matrices of (1) whenever we are talking about a nominal model.

Given the above nominal model, it is well-known from control theory that we can design an optimal nominal controller as a linear quadratic regulator (LQR), consisting of a Kalman estimator, with Kalman gains K_t, and linear feedbacks L_t (see e.g. [1]). From the optimal LQR we have a feedback law that explicitly gives the control signal through

$$\hat{x}_{t+1} = A\hat{x}_t + Bu_t + K_t\left[y_t - C\left(A\hat{x}_t + Bu_t\right)\right] \tag{2a}$$

$$u_t = L_t\hat{x}_t. \tag{2b}$$

One can alternatively consider a model-free reinforcement learning approach to solving the problem (1). Given the recent highly impressive successes of model-free reinforcement learning to highly complex domains (e.g. AlphaZero), it is perhaps surprising that such an approach can fail to perform on even simple problems [6], in particularly with regards to sample efficiency and robustness. In the authors' view, this failure is in large part due to an inherent disadvantage of model-free approaches as compared to model-based approaches in the case where good models are available.

Here we consider an indirectly model based approach to solving the problem (1). Given a fixed nominal model, we ask whether it is possible to modify the operation of the nominal controller using a reinforcement learning agent. That is, instead of using a reinforcement learning agent for directly providing actual control signals u_t as actions, we investigate various ways of letting the reinforcement learning agent's actions affect the control law in (2). This requires some care when defining the action space of the agent, and also opens up for designing various reward functions guided by the fixed nominal model, and we perform ablation studies over these design choices. We note the previous similar work done in [3,5], however, to the authors' knowledge, direct manipulation of nominal models seems to be unexplored in the literature.

1 Actions

There are many ways of modifying the operations of the nominal controller, but for brevity we here only discuss what we consider to be illustrative subsets of the full action space, left undefined here. This subset consists of

(a) Perturbations δA_t of the nominal A-matrix.
(b) Perturbations δu_t of the nominal control signal u_t.
(c) Hidden (explained later) perturbations δu_t^h of the nominal control signal u_t.

For completeness, the control law (2) using the possible actions (a)–(c) is

$$\hat{x}_{t+1} = (A + \delta A_t)\hat{x}_t + B(u_t - \delta u_t^h) + L_t \left[y_t - C \left((A + \delta A_t)\hat{x}_t + B(u_t - \delta u_t^h \right) \right], \quad (3a)$$

$$u_t = K_t \hat{x}_t + \delta u_t + \delta u_t^h, \quad (3b)$$

where the Kalman filter K_t and feedback L_t are adjusted according to the perturbations in the A-matrix. Note the difference that δu_t^h does not affect the state estimation Eq. (3a), whereas δu_t does.

2 Environment

For the observation space we will, again for brevity, only use a rolling window of measurements, that is, the observation o_t at time t that the agent receives is $[y_t, y_{t-1}, \ldots, y_{t-m}]^T$ for a window length m. To facilitate online learning, we will introduce normal shocks to the benchmark problems, simulating control towards a varying reference signal. We thus also extend the size and timing of the normal shocks to the observations. We point out however that the observation space can be extended in many different ways, e.g., by including the nominally estimated states, the nominal value function etc. to the observation.

As rewards we use the following signals:

System loss: $R_t = -\ell(y_t, u_t, t)$,
Innovation: $R_t = -\| y_t - C\left(A\hat{x}_t + Bu_t\right) \|^2$, and
Nominalized: $R_t = -\ell(y_t, u_t, t) - \delta R_t^{\text{nom}}$,

as well as a weighted aggregation of the above. *System loss* represents the naïve reward derived from (1), *Innovation* represents modifying the nominal model such that the system estimations becomes correct, *Nominalized* reward represents a reward shaping [4], intended to reduce the variance of stochastic policy gradient estimates as in Generalized Advantage Estimation [7], by factoring out a part of the raw system reward that can be considered as being the responsibility of nominal controller. That is, we may take $\delta R_t^{\text{nom}}(x_t, u_t, x_{t+1}) = \gamma V^{\text{nom}}(x_{t+1}) - V^{\text{nom}}(x_t)$, where $V^{\text{nom}}(x_t)$ denotes the (known) value function of the nominal control policy *assuming the nominal model to be exactly correct.* Concretely we implement an approximation of this by letting

$$\delta R_t^{\text{nom}} = -\ell(\hat{x}_{t+1|t,u_t}) \approx \mathbb{E}_{\pi^{\text{nom}}} \left[-\ell(x_{t+1}, u_t) | x_0, \ldots, x_t, u_0, \ldots, u_{t-1} \right]. \quad (4)$$

3 Experimental results

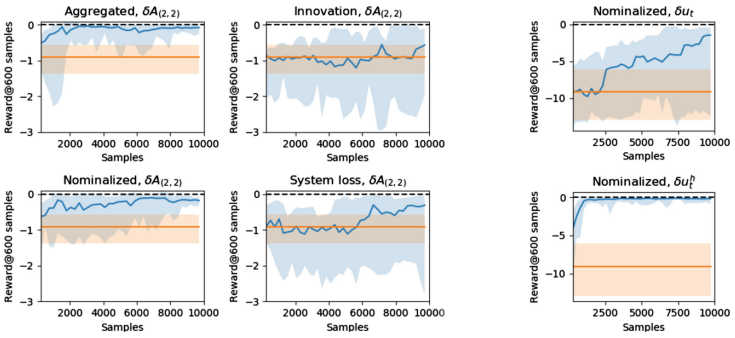
In view of [6], and the therein demonstrated failure of model-free reinforcement learning approaches to optimal control for even simple problems, we take as benchmark problems perturbations of a discrete-in-time frictionless unit mass double integrator system. The nominal model is thus

$$f^{nom}(x, y) = \begin{bmatrix} 1 & dt \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} dt^2/2 \\ dt \end{bmatrix} u, \qquad g^{nom}(x, u) = \begin{bmatrix} 1 & 0 \end{bmatrix} u. \quad (5a)$$

We train all agents with a PPO2 algorithm [8], as implemented in [2], with an increased learning rate, and use neural networks to approximate both the value functions and the policy. We train in an online fashion, i.e., we learn from a single trajectory of the system. Further, we induce large random shocks to the system at regular intervals, and all agents are trained using 10000 samples.

Misidentified linear system. The $(2,2)$-component of the A-matrix is replaced by $1-\mu \in (0,1]$, representing friction.

Piecewise linear system. $f(x,u) = f^{nom}(x,u) + \mathbb{I}_{\|x\|>1} \begin{bmatrix} 0 \\ -\text{sgn}(x)\sin\theta \end{bmatrix}$ corresponding to a mass on plane that at unit distance away from the origin slopes downward at an angle θ.



(a) Misidentified linear system (b) P.W.L. system

Fig. 1. Median reward of 12 agents compared to an optimal controller, evaluated after every 256 samples during training on a set of fixed episodes. Trained agent is in blue, nominal controller is in orange, and shaded regions indicate the 10–90th percentiles. (a) Varying reward signals. (b) Varying action spaces. (Color figure online)

Main results are presented in Fig. 1. Figure 1a shows a clear improvement in sample efficiency using reward nominalization, compared to both raw system loss and innovation rewards. A weighted aggregation appears to show an additional increase in robustness, indicated by relatively narrower error bars. Figure 1b illustrates the importance of choosing the correct action, in the top row the agents' actions enters the feedback loop of the nominal controller, and the action of the agent causes severe problems for the nominal state estimator. On the other hand, when acting invisibly, the agent successfully learns to compensate for the unmodelled nonlinearities using only roughly 1000 samples.

References

1. Glad, T., Ljung, L.: Control Theory. Taylor & Francis, London (2000)
2. Hill, A., et al.: Stable baselines (2019). https://github.com/hill-a/stable-baselines
3. Koryakovskiy, I., Kudruss, M., Vallery, H., Babuška, R., Caarls, W.: Model-plant mismatch compensation using reinforcement learning. IEEE Robot. Autom. Lett. **3**(3), 2471–2477 (2018)
4. Ng, A.Y., Harada, D., Russell, S.: Policy invariance under reward transformations. ICML **99**, 278–287 (1999)
5. Rastogi, D., Koryakovskiy, I., Kober, J.: Sample-efficient reinforcement learning via difference models. In: Machine Learning in Planning and Control of Robot Motion Workshop at ICRA (2018)
6. Recht, B.: A tour of reinforcement learning. AAnnu. Rev. Control. Robot. Auton. Syst. (2018)
7. Schulman, J., Moritz, P., Levine, S., Jordan, M., Abbeel, P.: High-dimensional continuous control using generalized advantage estimation. arXiv preprint arXiv:1506.02438 (2015)
8. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017)