Accelerated ARS for PID Tuning

November 14, 2021

Abstract

Model-free deep reinforcement learning algorithms are highly successful in a range of robotic tasks and A Simple Random Search based technique ARS was shown to be highly competitive among all the other Model Free Techniques. Further, ARS achieves it using a simple Linear policy over the complex Neural Network Policies. On the other hand, the PID controllers, well known in control theory are linear and are widely used in many applications. In this work we train PID Gains using an accelerated version of ARS and further we illustrate that accelerated ARS achieves faster convergence then the Augmented Random Search on Bench Mark Simulation Tasks. This work provides a way to quickly train PID Gains for Industrial Applications using rewards appropriately with the accelerated version of ARS.

1 Introduction

1.1 Feed Back Controller-PID Controller

PID Control, is a Model Free Control Technique, iterative and a Deterministic Linear State feedback controller. The iterative process involve tuning gains, the three parameters namely the proportional gain, the integral gain and the derivative gain.

$$u_t = K_{\mathrm{p}} e(t) + K_{\mathrm{i}} \int_0^t e(\tau) d\tau + K_{\mathrm{d}} \frac{de(t)}{dt}$$

where $e(t)$ is error. Though the control is intuitive it is hard to tune gains and there are accordingly various methods for loop tuning. Learning techniques are also applied to tune the gains. ZN method is a traditional manual method, It is performed by setting the I (integral) and D (derivative) gains to zero and the "P" (proportional) gain, K_p is then increased (from zero) until it reaches the ultimate gain K_u, at which output oscillates with constant the oscillation period T_u, which are then used to set the P, I, and D gains and control is given by

$$u(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s\right) e(s) \tag{1}$$

.

1.2 Augmented Random Search

Random Search is a Derivative Free Optimisation where the gradient is estimated through finite difference Method [1] .Objective is to maximize Expected return of a policy π parameterised by θ under noise ξ

$$\max_{\theta} \mathbb{E}_{\xi} \left[r \left(\pi_{\theta}, \xi \right) \right]$$

The gradient is found from the gradient estimate obtained from gradient of smoothened version of above objective with Gaussian noise unlike from policy gradient theorem. Gradient of smoothened objective is

$$\frac{r\left(\pi_{\theta+\nu\delta},\xi_1\right)-r\left(\pi_{\theta},\xi_2\right)}{\nu}\delta$$

where δ is zero mean Gaussian. If ν is sufficiently small, the Gradient estimate would be close to the gradient of original objective. Further bias could be reduced with a two point estimate,

$$\frac{r\left(\pi_{\theta+\nu\delta},\xi_1\right)-r\left(\pi_{\theta-\nu\delta},\xi_2\right)}{\nu}\delta.$$

A Basic Random Search would involve the update of policy parameters according to

$$\theta_{j+1}=\theta_j+\frac{\alpha}{N}\sum_{k=1}^{N}\left[r\left(\pi_{j,k,+}\right)-r\left(\pi_{j,k,-}\right)\right]\delta_k \tag{2}$$

Augmented Random Search, defines an update rule,

$$\theta_{j+1}=\theta_j+\frac{\alpha}{b\sigma_R}\sum_{k=1}^{b}\left[r\left(\pi_{j,(k),+}\right)-r\left(\pi_{j,(k),-}\right)\right]\delta_{(k)} \tag{3}$$

Policy is linear,

$$pi_j(x)=(\theta_j)(x)$$

where x is the state and it proposes three Augmentations to Basic Random Search.

2 Algorithm

2.1 Accelerated ARS

Most optimisers use Adam to accelerate Stochastic Gradient Descent in practical implementations. In ARS we estimate the gradient, an acceleration technique is not used in its implementation. The [2] shows significant order of improvement in the accelerated Random Search setting. Hence we define an acceleration based Gradient Estimate to ARS for faster convergence. We adopt the acceleration technique with constant step sizes as defined in [3] The Modified ARS Algorithm

Algorithm 1: Accelerated ARS

Runaverage$_j=\sum_{i<j}\left(1-\beta\right)^i\theta_{(i-\tau)}$
$\theta_{j+1}=\theta_j+\frac{\alpha}{b\sigma_R}\sum_{k=1}^{b}\left[r\left(\pi_{j,(k),+}\right)-r\left(\pi_{j,(k),-}\right)\right]\delta_{(k)}$
$\theta_{acc_{j+1}}=\gamma\theta_{j+1}+(1-\gamma)\,Runaverage_j$

2.2 PID Tuning using Acelerated ARS

PID Control Law using ZN Method could be expressed as

$$u(s)=K_p\left(1+\frac{1}{T_i s}+T_d s\right)e(s) \tag{4}$$

This fits into Accelerated ARS framework with States as e(s),
Policy Parameters as $K_p\left(1+\frac{1}{T_i s}+T_d s\right)$, the Gains.

3 Experiments

In this section we show the faster convergence of ARS in Benchmark Mujocco Experiments. The hyper parameters for ARS are taken from the original Paper. The γ and β, hyperparameters of Accelerated are tuned accordingly that β is greater than α It could be seen that the returns observed are higher than ARS. (Blue: ARS Red: Accelerated ARS)

3.1 Accelerated ARS: Benchmark Mujocco

Figures:

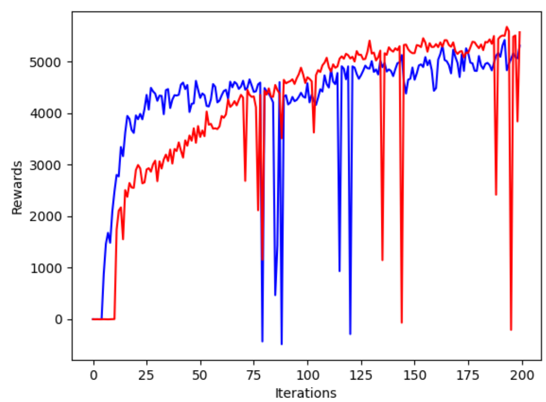

Figure 1: Reward for Half Cheetah

Table for Rewards:

3.2 Accelerated ARS for PID Tuning in Stoch, Quadruped

Stoch, a Quadruped is trained using a linear feedback policy that takes the torso orientation and the terrain slope as inputs and tracks desired joint angles, using PID control, where the desired angles are found from Inverse Kinematic solver with the policy outputs.

4 Discussions

We demonstrate a faster convergence using Acceleration with ARS. We apply the accelerated ARS to quickly tune PID Gains for Controller in Stoch.

References

[1] Horia Mania, Aurelia Guy, and Benjamin Recht. Simple random search provides a competitive approach to reinforcement learning. *arXiv preprint arXiv:1803.07055*, 2018.

[2] Yurii Nesterov and Vladimir Spokoiny. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17(2):527–566, 2017.

[3] Prateek Jain, Sham M Kakade, Rahul Kidambi, Praneeth Netrapalli, and Aaron Sidford. Accelerating stochastic gradient descent for least squares regression. In *Conference On Learning Theory*, pages 545–604. PMLR, 2018.