Summer 2020 Research Report

Cheng Chen McGill University
August 2020

I. Introduction

This report includes the results from Monte Carlo simulations of certainty equivalence adaptive control algorithm, Thompson Sampling algorithm and Thompson Sampling with dynamic episodes algorithm under two different controllable parametric assumptions. Monte Carlo simulation represents a broad class of computational simulations that rely on repeated random sampling to obtain numerical results. It is widely used in many fields such as finance, project management, engineering and etc.

II. Theories

Control systems can be represented in state space form as shown below:

$$x_{t+1} = Ax_t + Bu_t + w_t \tag{1}$$

where A, B represent the system parameters, x_t is the current state, x_{t+1} is the next state, u_t is the control input and w_t is the random generated noise based on normal distribution with zero as the mean and one as the covariance. The cost at each time step is represented by:

$$c_t = x_t^T Q x_t + u_t^T R u_t \tag{2}$$

where Q and R are both positive definite. We can define θ as vec(A,B). It is well known that the optimal cost is given by [5]:

$$J(\theta) = tr(S(\theta)\Sigma) \tag{3}$$

where $S(\theta)$ is the solution to the algebraic Riccati equation and Σ is the covariance, we will use the identity matrix as the covariance in our case:

$$S(\theta) = Q + A^T S(\theta) A - A^T S(\theta) B (R + B^T S(\theta) B)^{-1} B^T S(\theta) A \tag{4}$$

The optimal control law can be obtained by:

$$u = -G(\theta)x \tag{5}$$

where $G(\theta)$ is given by:

$$G(\theta) = (R + B^T S(\theta) B)^{-1} B^T S(\theta) A \tag{6}$$

In order to measure the quality of algorithms, the cost at each time step as described in (2) will be compared to the optimal cost obtained in (3).Therefore, the regret will be defined as [1]:

$$R = \sum_{t=1}^{T} [c_t - J(\theta)] \tag{7}$$

As suggested in the literature [3], we can achieve a regret bound of

$$O(p\sqrt{T}) \tag{8}$$

,where p is a constant and T is the time horizon.

A. Certainty Equivalence

In certainty equivalence [4], we generate an estimate, $\hat{\theta}_t$, of the unknown parameters θ of the model based on the observed data and apply the optimal control law as described above. Let z_t denote $vec(x_t, u_t)$. We can write:

$$X_t = Z_t^T \theta + W_t \tag{9}$$

where X_t, Z_t, W_t are:

$$\begin{bmatrix} x_2 \\ \vdots \\ x_t \end{bmatrix}, \begin{bmatrix} z_1^T \\ \vdots \\ z_t^T \end{bmatrix}, \begin{bmatrix} w_1 \\ \vdots \\ w_t \end{bmatrix} \tag{10}$$

The linear least squares estimate can be written in recursive form as: :

$$\hat{\theta_{t+1}} = \hat{\theta}_t + \frac{\Sigma_t z_t (x_{t+1} - z_t^T \hat{\theta}_t)}{1 + z_t^T \Sigma_t z_t} \tag{11}$$

$$\Sigma_{t+1} = \Sigma_t - \frac{\Sigma_t z_t z_t^T \Sigma_t}{1 + z_t^T \Sigma_t z_t} \tag{12}$$

The following figure is the code snippet of the algorithm:

B. Thompson Sampling

Thompson Sampling maintains a posterior belief π_t over the parameter θ [4]. At each time step t, we sample a value $\tilde{\theta}_t$ from π_t and apply the control law in (5). The posterior belief π_t is a Gaussian distribution with $\hat{\theta}_t$ as the mean and Σ_t as the covariance, $\tilde{\theta}_t \sim N(\hat{\theta}_t, \Sigma_t)$, where $\hat{\theta}_t$ and Σ_t satisfy the linear least squares estimate in recursive form as described in (10) and (11). The following figure is the code snippet of the algorithm:

C. Thompson Sampling with dynamic episodes

Thompson Sampling with dynamic episodes(TSDE) is Thompson Sampling operates in episodes which are

```
for n = 1:N
    x = zeros(Float64,p,1)
    u = ones(Float64,q,1)
    w = zeros(Float64,p,1)
    θ̂ = vcat(zeros(Float64,p,p),ones(Float64,q,p))
    Σ = Symmetric(Matrix{Float64}(I,p+q,p+q))
    cost = zeros(T)
    regret = zeros(T)
    avg_cost = zeros(T)
    gain = zeros(Float64,q,p)
    for t = 1:T
        if t == 1
            cost[t] = (x'*Q*x + u'*R*u)[1]
            regret[t] = (x'*Q*x + u'*R*u)[1] - j_optimal[1]
        else
            cost[t] = cost[t-1] + (x'*Q*x + u'*R*u)[1]
            regret[t] = regret[t-1] + (x'*Q*x + u'*R*u)[1] - j_optimal[1
        end
        avg_cost[t] = cost[t] / t


        Â = reshape(θ̂[1:p,:],p,p)
        B̂ = reshape(θ̂[(p+1):(p+q),:],p,q)
        Ŝ = dare(Â,B̂,Q,R)
        gain = (R + B̂'*Ŝ*B̂)\(B̂'*Ŝ*Â)
        u = -gain * x
        w = randn(p,1)
        z = vcat(x,u)

        x = A * x + B * u + w

        normalize = 1 + (z' * Σ * z)[1]
        θ̂ = θ̂ + (Σ * z * (x - θ̂'*z)') / normalize
        Σ = Σ - Symmetric(Σ * z * z' * Σ) / normalize
    end
```

Fig. 1. code snippet for CE

```
for n = 1:N
    x = zeros(Float64,p,1)
    u = ones(Float64,q,1)
    w = zeros(Float64,p,1)
    θ̂ = vcat(zeros(Float64,p,p),ones(Float64,q,p))
    Σ = Symmetric(Matrix{Float64}(I,p+q,p+q))
    cost = zeros(T)
    regret = zeros(T)
    avg_cost = zeros(T)
    gain = zeros(Float64,q,p)
    for t in 1:T
        if t == 1
            cost[t] = (x'*Q*x + u'*R*u)[1]
            regret[t] = (x'*Q*x + u'*R*u)[1] - j_optimal[1]
        else
            cost[t] = cost[t-1] + (x'*Q*x + u'*R*u)[1]
            regret[t] = regret[t-1] + (x'*Q*x + u'*R*u)[1] - j_optimal[1]
        end
        avg_cost[t] = cost[t] / t
        θ̃ = zeros(Float64,(p+q),p)
        for i in 1:size(θ̂,2)
            θ̃[:,i] = reshape(rand(MvNormal(θ̂[:,i],Σ)),p+q,1)
        end

        Â = reshape(θ̃[1:p,:],p,p)
        B̂ = reshape(θ̃[(p+1):(p+q),:],p,q)
        Ŝ = dare(Â,B̂,Q,R)
        gain = (R + B̂'*Ŝ*B̂)\(B̂'*Ŝ*Â)
        u = -gain * x
        z = vcat(x,u)
        w = randn(p,1)
        x = A * x + B * u + w
        normalize = 1 + (z' * Σ * z)[1]
        θ̂ = θ̂ + (Σ * z * (x - θ̂'*z)') / normalize
        Σ = Σ - Symmetric(Σ * z * z' * Σ) / normalize
    end
```

Fig. 2. code snippet for TS

not fixed. The algorithm is described as below [5]:

Algorithm 1: TSDE

Input: $\Sigma_1, \hat{\theta_1}$
initialization: t $\leftarrow 1, t_j \leftarrow 0$
for *episodes j = 1,2,...* **do**
  $T_{j-1} \leftarrow t - t_j$
  $t_j \leftarrow t$
  Generate $\tilde{\theta_j} \sim N(\hat{\theta_{tj}}, \Sigma_{tj})$
  Compute $G_j = G(\tilde{\theta_j})$ from (6)
  while $t \leq t_j + T_{j-1}$ *and* $det(\Sigma_t) \geq 0.5 det(\Sigma_{tj})$ **do**
    Apply control in (5)
    Observe new state x_{t+1}
    Update $\hat{\theta_t}$ and Σ_t according to (10) and (11)
    t $\leftarrow t + 1$
  end
end

The following figure is the code snippet of the algorithm:

III. Simulation Results

During the simulation, we set N to be the number of sample paths and T to be the maximum time steps in each sample path. In our case, We set N to be 100 and T to be 10000 in all simulations to get clear plots and keep consistency. In the first simulation, we used a controllable system where:

$$A = \begin{bmatrix} 0.1 & -0.2 \\ 0.3 & 0.4 \end{bmatrix} = B \tag{13}$$

$$Q = R = I \tag{14}$$

Theoretically, we can obtain the optimal cost by (3) which is equal to 2.27. The optimal gain can be obtained by (6) which is: $\begin{bmatrix} 0.0994 & 0.0864 \\ 0.0864 & 0.1703 \end{bmatrix}$. The following figures 4-6 show the plots of average cost vs t for CE, TS and TSDE. Note that the lines with different colors represents different sample paths. As can be seen, the cost for all three algorithms are converging to the optimal cost, which is what we expected. However, the gain at the final time step is not the same as the optimal gain. Since the gain is a multidimensional vector instead of a scalar, we were not able to plot it versus time. We took average of the gain at the final time step(T) over the number of sample paths(N) for the algorithms. The average estimated gain at final time step is: $\begin{bmatrix} 0.0484 & 0.02958 \\ -0.01751 & 0.2259 \end{bmatrix}$, $\begin{bmatrix} 0.0444 & -0.0293 \\ -0.0377 & 0.2133 \end{bmatrix}$, $\begin{bmatrix} 0.0475 & -0.0283 \\ -0.0396 & 0.2103 \end{bmatrix}$ for CE, TS and TSDE respectively.

To be able to compare the performance of the algorithms, we need to plot the regret. We know that for regret:

$$R = c * T^{\alpha} \tag{15}$$

```
for n = 1:N
    x = zeros(Float64,p,1)
    u = ones(Float64,q,1)
    θ̂ = vcat(zeros(Float64,p,p),ones(Float64,q,p))
    Σ = Symmetric(Matrix{Float64}(I,p+q,p+q))
    cost = []
    regret = []
    avg_cost = []
    gain = []
    t = 1
    tⱼ = 0
    for j in 1:J
        Tⱼ₋₁ = t - tⱼ
        tⱼ = t
        θ̃ = zeros(Float64,(p+q),p)
        for i in 1:size(θ̂,2)
            θ̃[:,i] = reshape(rand(MvNormal(θ̂[:,i],Σ)),p+q,1)
        end
        Â = reshape(θ̃[1:p,:],p,p)
        B̂ = reshape(θ̃[(p+1):(p+q),:],p,q)
        Ŝ = dare(Â,B̂,Q,R)
        Gⱼ = (R + B̂'*Ŝ*B̂)\(B̂'*Ŝ*Â)
        Σⱼ = Σ
        while t <= (tⱼ + Tⱼ₋₁) && det(Σ) >= 0.5*det(Σⱼ)
            if t == 1
                push!(cost,(x'*Q*x + u'*R*u)[1])
                push!(regret,(x'*Q*x + u'*R*u)[1] - j_optimal[1])
            else
                push!(cost,cost[t-1] + (x'*Q*x + u'*R*u)[1])
                push!(regret,regret[t-1]+(x'*Q*x + u'*R*u)[1] - j_optima
            end
            push!(avg_cost,cost[t]/t)
            u = -Gⱼ * x
            push!(gain,Gⱼ)
            w = randn(p,1)
            z = vcat(x,u)
            x = A * x + B * u + w
            normalize = 1 + (z' * Σ * z)[1]
            θ̂ = θ̂ + (Σ * z * (x - θ̂'*z)') / normalize
            Σ = Σ - Symmetric(Σ * z * z' * Σ) / normalize
            t += 1
        end
```

Fig. 3. code snippet for TSDE

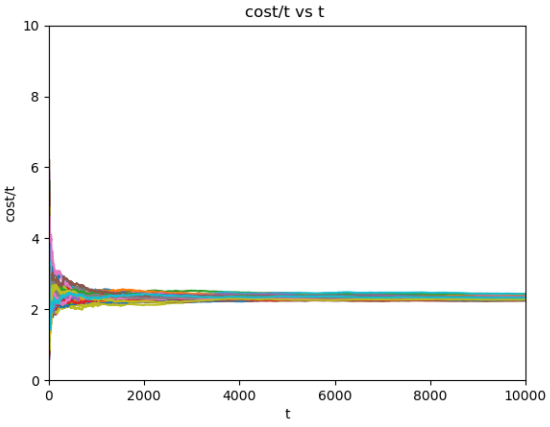

Fig. 4. average cost vs t for CE

α is expected to be around 0.5 [2]. To obtain c and α, we can plot average regret versus t and fit the data according to (14). The following figures 7-9 show the plots of average regret vs. t for CE, TS and TSDE. Note that the blue line is the actual

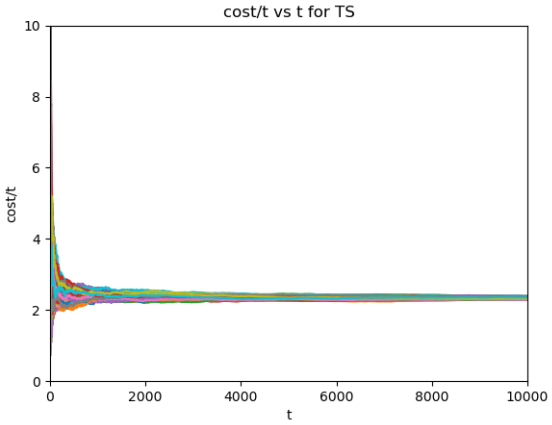

Fig. 5. average cost vs t for TS

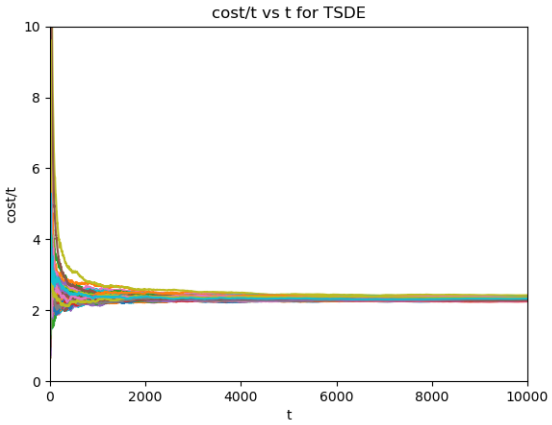

Fig. 6. average cost vs t for TSDE

data and the red line is the fitted data. The values of (c,α) are (0.2158, 0.854), (1807.5, 0.0903) and (0.2242, 0.856) for CE, TS and TSDE respectively.

We also plotted log(average regret) versus log(t), if we find the slope of the data, the value of the slope should be close to the value of α. The following figures 10-12 show the plots of log(average regret) versus log(t) for CE, TS and TSDE. The overall trend of the slopes are consistent with the values of α (TS has the smallest α), but they do not have the exact same values.

In addition, we plotted average regret versus sqrt(t). Since we expect the values of α to be 0.5, the slope of this plot should be the value of c. The following figures 13-15 show the plots of average regret versus sqrt(t) for CE, TS and TSDE. Ideally, the values of the slopes of should be the same as the values of c. However, since α is not exactly 0.5, we have inconsistent values. Figure 28 summarizes the results in the first simulation.

For the second simulation, we used a controllable system

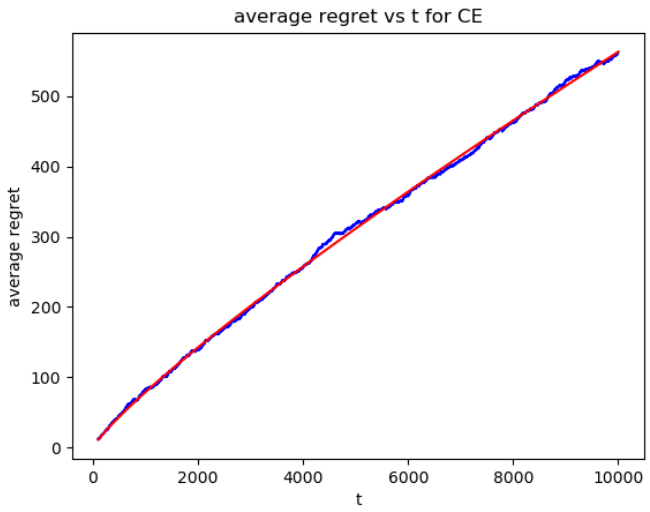

Fig. 7. average regret vs t for CE

Fig. 9. average regret vs t for TSDE

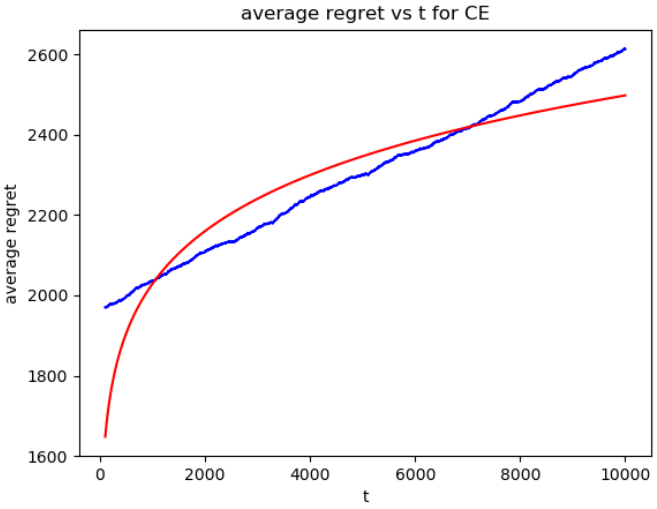

Fig. 8. average regret vs t for TS

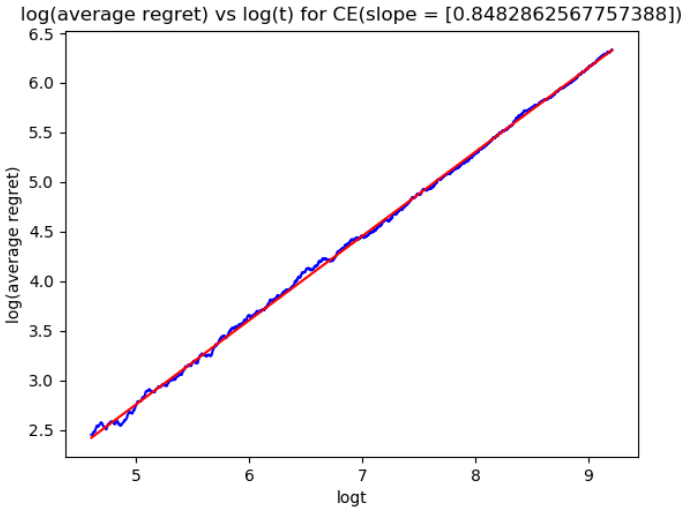

Fig. 10. log average regret vs log t CE

where:

$$A = \begin{bmatrix} 0.1 & -0.4 \\ 0.4 & 0.5 \end{bmatrix} = B \tag{16}$$

$$Q = R = I \tag{17}$$

Theoretically, we can obtain the optimal cost by (3) which is equal to 2.42. The optimal gain can be obtained by (6) which is: $\begin{bmatrix} 0.1608 & 0.1262 \\ 0.1262 & 0.2568 \end{bmatrix}$. The following figures 16-18 show the plots of average cost vs t for CE, TS and TSDE. Note that the lines with different colors represents different sample paths. As can be seen, the cost for all three algorithms are converging to the optimal cost, which is what we expected. However, the gain at the final time step is not the same as the optimal gain. Since the gain is a multidimensional vector instead of a scalar, we were not able to plot it versus time. We took average of the gain at the final time step(T) over the number of sample paths(N) for the algorithms. The average estimated gain at final time step is: $\begin{bmatrix} 0.08229 & -0.0163 \\ -0.0572 & 0.3079 \end{bmatrix}$, $\begin{bmatrix} 0.1041 & -0.0745 \\ -0.0783 & 0.3064 \end{bmatrix}$, $\begin{bmatrix} 0.1020 & -0.0727 \\ -0.0830 & 0.3046 \end{bmatrix}$ for CE, TS and TSDE respectively.

we also plotted average regret versus t to be able to compare the performance of the algorithms. The following figures 19-21 show the plots of average regret vs. t for CE, TS and TSDE. Note that the blue line is the actual data and the red line is the fitted data. The values of (c,α) are (0.1179, 1.006), (0.631, 0.854) and (0.455, 0.891) for CE, TS and TSDE respectively.

We also plotted log(average regret) versus log(t), if we find the slope of the data, the value of the slope should be close

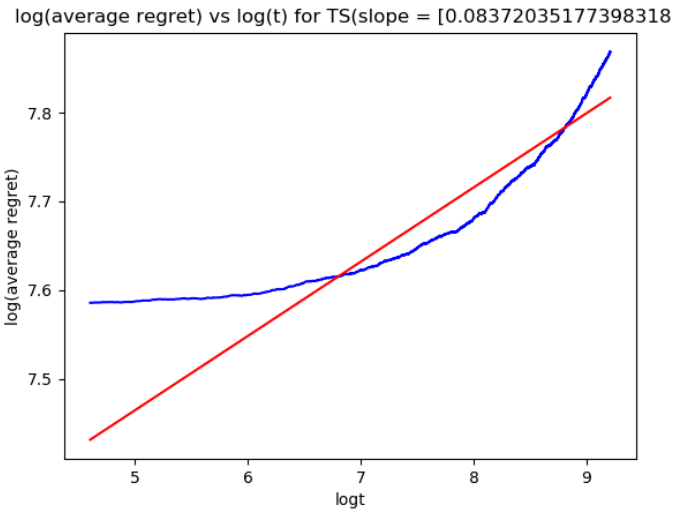

Fig. 11. log average regret vs log t TS

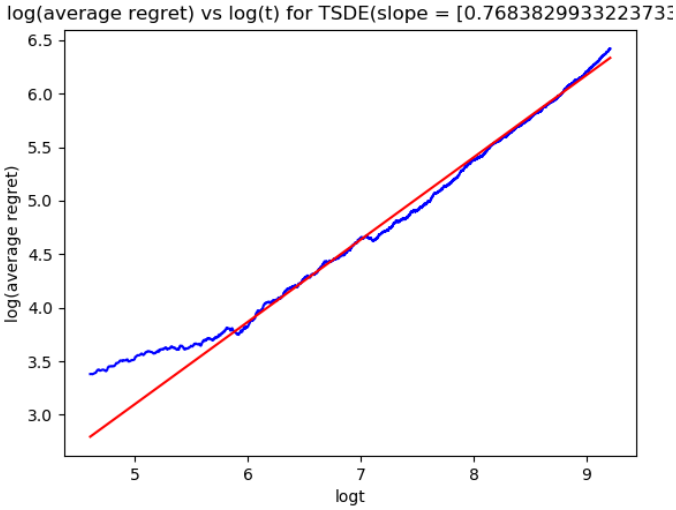

Fig. 12. log average regret vs log t TSDE

the value of α. The following figures 22-24 show the plots of log(average regret) versus log(t) for CE, TS and TSDE. The overall trend of the slopes are consistent with the values of α (TS has the smallest α), but they do not have the exact same values.

In addition, we plotted average regret versus sqrt(t). Since we expect the values of α to be 0.5, the slope of this plot should be the value of c. The following figures 25-27 show the plots of average regret versus sqrt(t) for CE, TS and TSDE. Ideally, the values of the slopes of should be the same as the values of c. However, since α is not exactly 0.5, we have inconsistent values. Figure 29 summarizes the results in the second simulation.

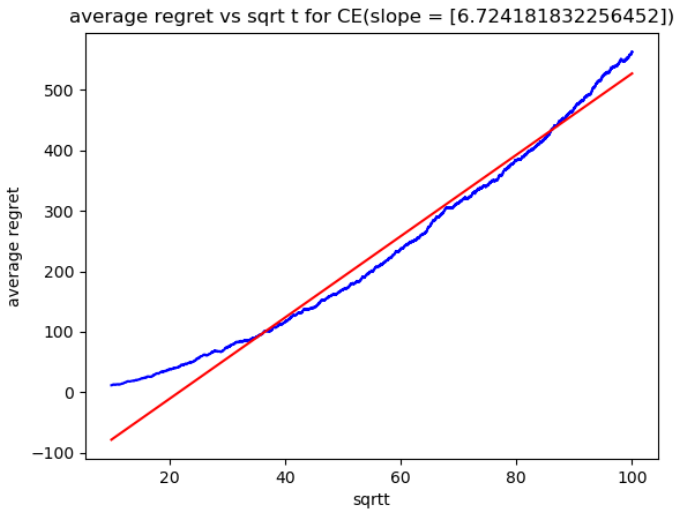

Fig. 13. average regret vs sqrt t CE

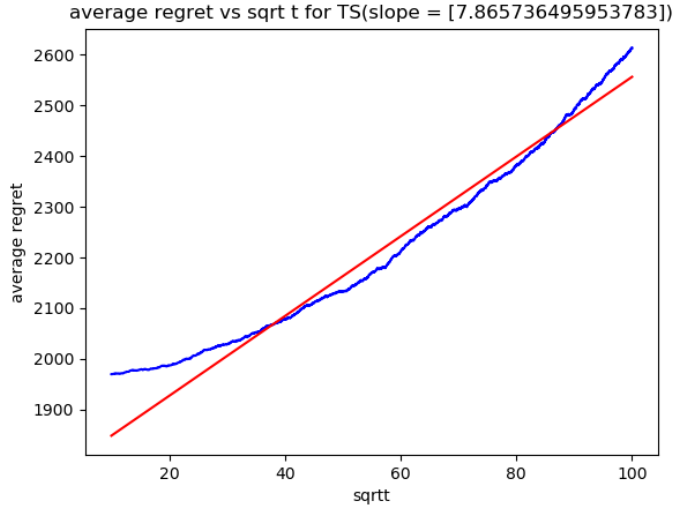

Fig. 14. average regret vs sqrt t TS

IV. Conclusion

To summarize, we found that the average cost is converging to the optimal cost as expected, which implies that there should not be any issues in the implementations of the algorithms. However, the average of estimated gain is not the same as the optimal gain, adding persistence of excitation might help solve this problem. Moreover, we found that α is much smaller than 0.5 while most literature suggested that α should be close to 0.5. To improve the accuracy of the results, we can try running the simulations for longer time and adding more sample paths.

References

[1] Yasin Abbasi-Yadkori and Csaba Szepesvári. Regret bounds for the adaptive control of linear quadratic systems. In Sham M. Kakade and

Fig. 15. average regret vs sqrt t TSDE

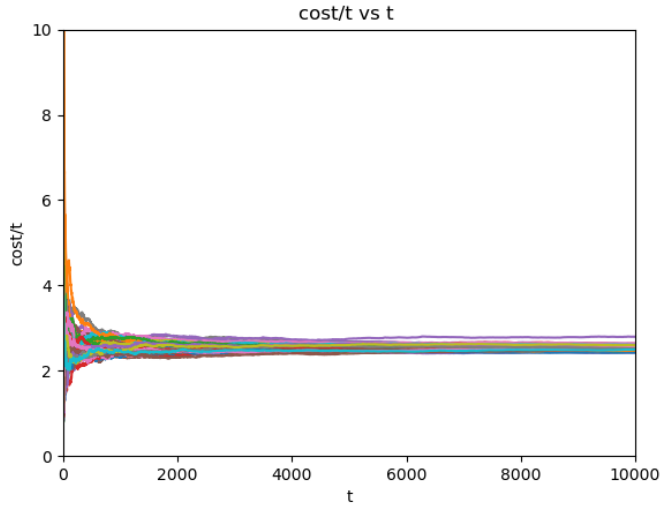

Fig. 16. average cost vs t for CE

Ulrike von Luxburg, editors, *Proceedings of the 24th Annual Conference on Learning Theory*, volume 19 of *Proceedings of Machine Learning Research*, pages 1–26, Budapest, Hungary, 09–11 Jun 2011. PMLR.

[2] Mohamad Kazem Shirani Faradonbeh, Ambuj Tewari, and George Michailidis. On optimality of adaptive linear-quadratic regulators. *ArXiv*, abs/1806.10749, 2018.

[3] Morteza Ibrahimi, Adel Javanmard, and Benjamin Van Roy. Efficient reinforcement learning for high dimensional linear quadratic systems, 2013.

[4] Aditya Mahajan. Overview of adaptive control for linear systems.

[5] Yi Ouyang, Mukul Gagrani, and Rahul Jain. Learning-based control of unknown linear systems with thompson sampling, 2017.

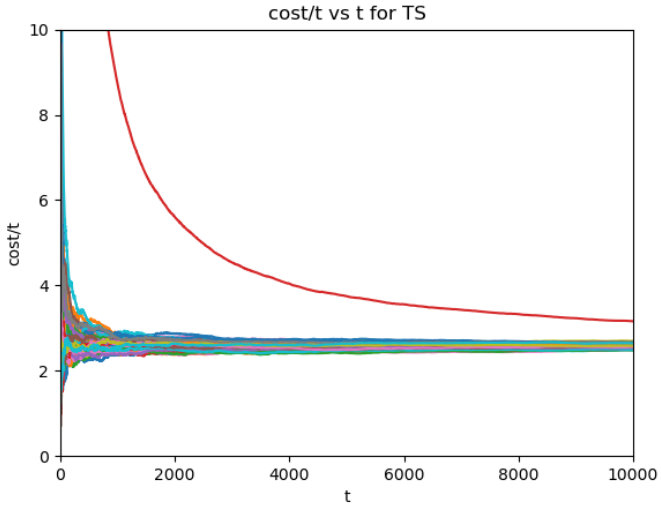

Fig. 17. average cost vs t for TS

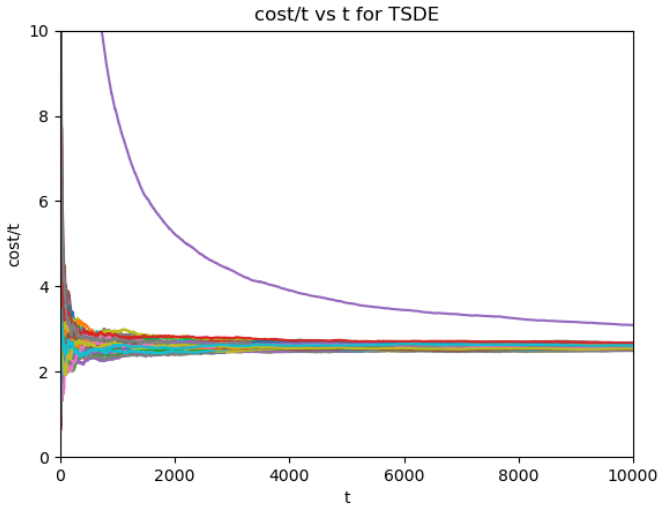

Fig. 18. average cost vs t for TSDE

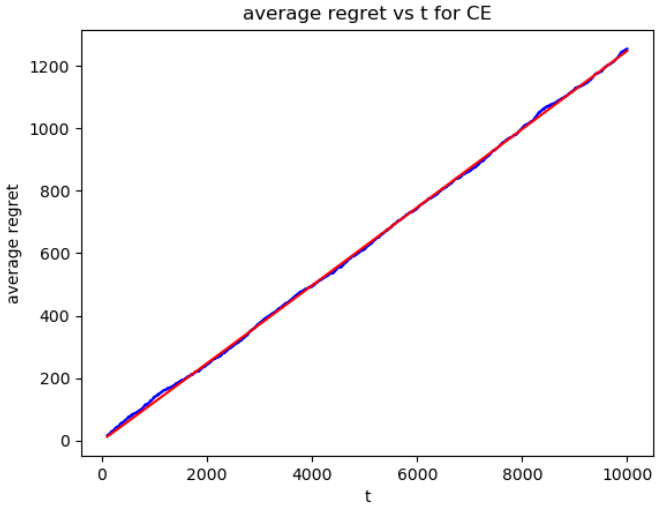

Fig. 19. average regret vs t for CE

Fig. 21. average regret vs t for TSDE

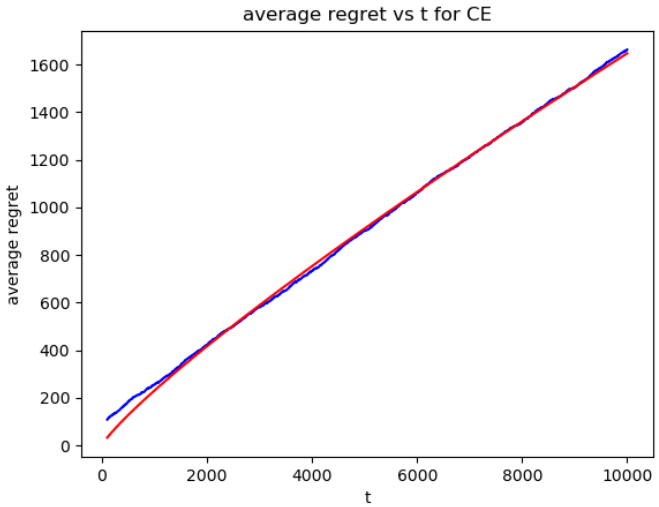

Fig. 20. average regret vs t for TS

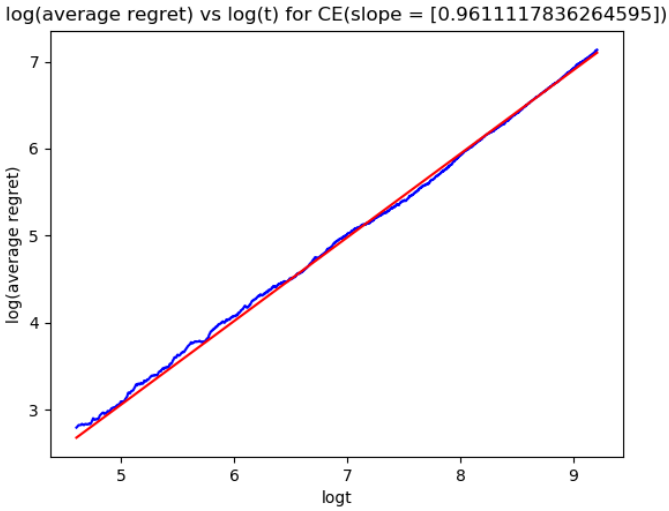

Fig. 22. log average regret vs log t CE

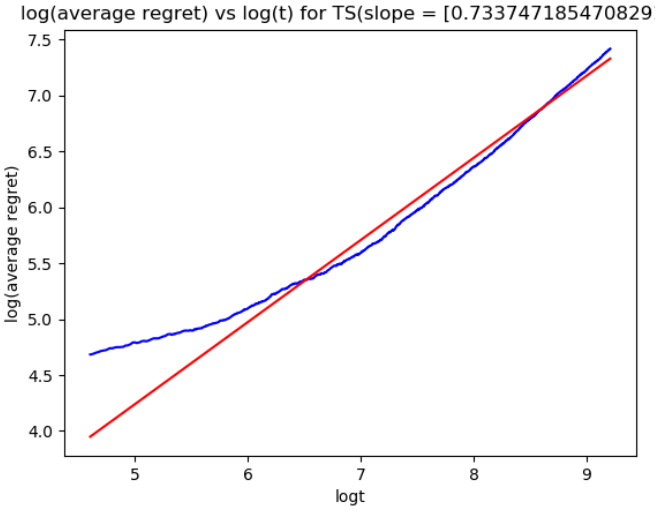

Fig. 23. log average regret vs log t TS

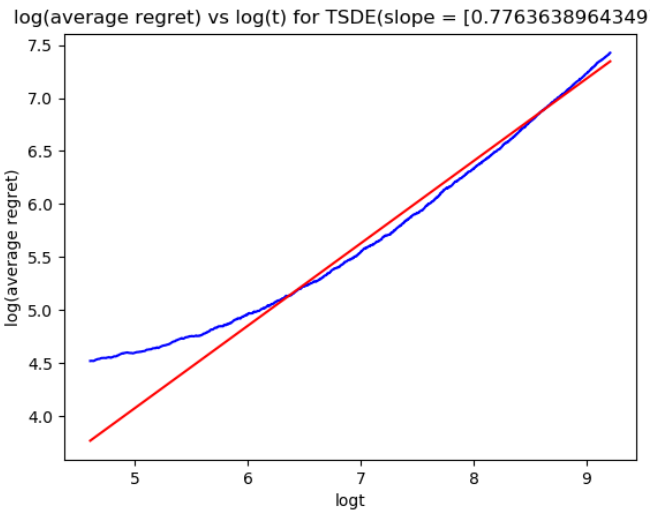

Fig. 24. log average regret vs log t TSDE

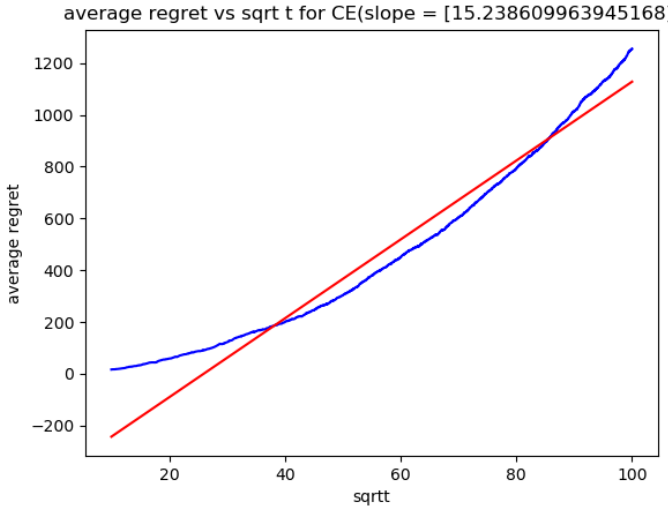

Fig. 25. average regret vs sqrt t CE

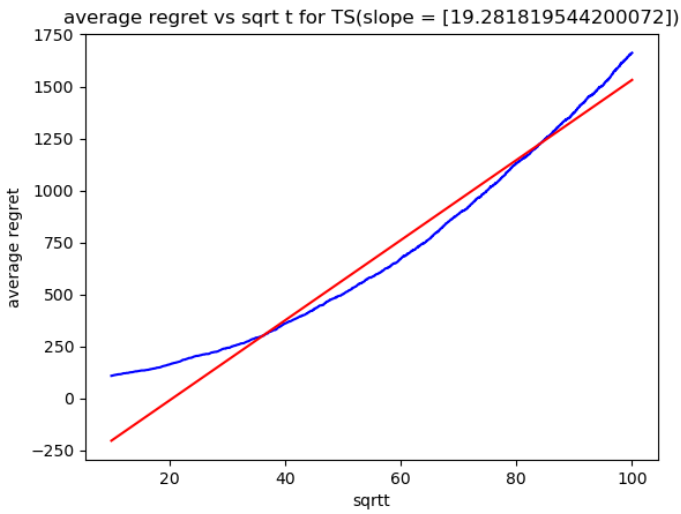

Fig. 26. average regret vs sqrt t TS

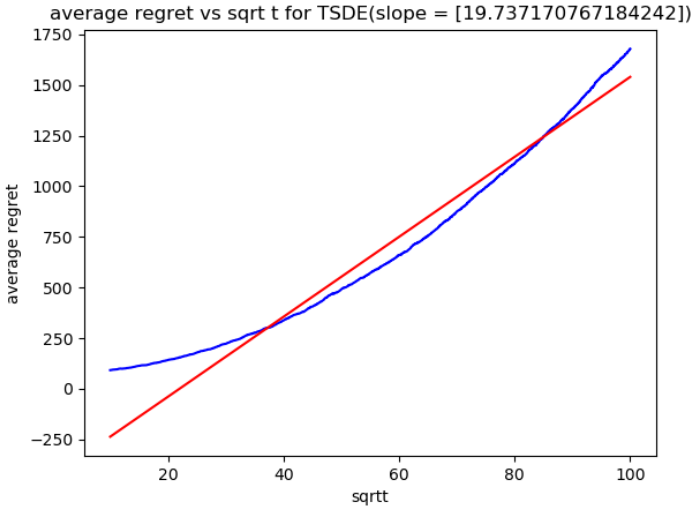

Fig. 27. average regret vs sqrt t TSDE

	CE	TS	TSDE
Avg regret vs t	C = 0.2158 Alpha = 0.854	C = 1087.5 Alpha = 0.0903	C = 0.2242 Alpha = 0.856
Log regret vs log t	Slope = 0.848	Slope = 0.0837	Slope = 0.7684
Avg regret vs sqrt t	Slope = 6.72	Slope = 7.866	Slope = 7.007

Fig. 28. summary of results in the first simulation

	CE	TS	TSDE
Avg regret vs t	C = 0.1179 Alpha = 1.006	C = 0.631 Alpha = 0.854	C = 0.455 Alpha = 0.891
Log regret vs log t	Slope = 0.961	Slope = 0.734	Slope = 0.776
Avg regret vs sqrt t	Slope = 15.24	Slope = 19.3	Slope = 19.74

Fig. 29. summary of results in the second simulation