Supplementary Material - Robust and Cheap Safety Measure for Exoskeletal Learning Control with Estimated Uniform PAC (EUPAC)

1 Numerical Estimation of Uniform PAC

To make UPAC usable as an online heuristic, we derive a numerical estimate. This encompasses 1) linking the UPAC probability to a set of seen regrets and their density, and 2) checking if seen regrets fulfill the UPAC condition at the current timestep.

For 1) we condition the overall probability on ordered sets of W regrets Ω_W, that is

$$\Omega = \bigcup_k \Omega_{W,k}\,. \tag{1}$$

Inside the UPAC criterion, we bound the polynomial F from below with a $\bar{F}$ and count $N_\tau(\Omega)$. The regret probability can be modelled with any joint density estimation approach. We estimate the univariate regret density as a floating average density of bin-counted regrets. Over several timesteps, we use a mulitnomial distribution of the univariate regret density as the joint density estimate. To tame the multinomial calculations we represent any number W' of seen regrets Δ by their binned regrets $\Delta_D(\Delta) = (\Delta_u - \Delta_l)/2 \cdot \mathbb{1}_{\Delta_u > \Delta > \Delta_l}$ for two consecutive Δ_u, Δ_l of the ordered boundary set $D = \{0\} \cup \left\{\Delta \,|\, N_\Delta(\Omega_W) = \bar{F}(1/\Delta)\right\} \cup \{\Delta_{\max}\}$. Note that $|D| = W + 2$. For 2) we use quantifier elimination on the complementary set of the conditioned UPAC criterion. The actual core calculation of Estimated UPAC (EUPAC) then amounts to be a set of simply checkable inequalities (EUPAC by Interval Checking) for the binned regrets.

In detail, we bound F from below with a $\bar{F}$. Since S, A, H and δ are constant and positive for one and the same learning problem, set

$$\bar{F}(1/\tau) = c_1 + c_2/\tau + O(1/\tau^2) \tag{2}$$

as a lower bound for F. This tightens the upper bound for $N_\tau(\Omega)$ rendering the criterion to be more conservative than UPAC and cheapens calculation cost if higher order terms are held negligible via some $p = (1/\tau_0^2)/(c_1 + c_2/\tau_0) \to 0$. It further allows for use in continuous problems wherein S and A are not typically defined. To find the constants c_1 and c_2, hold

$$\bar{F}(1/\tau_i) = W_0 \gamma_i \tag{3}$$

at arbitrary target bound proportions γ_0 and γ_1 of a reference amount of regrets W_0.

For example, only the proportion of γ_0 of W_0 regrets should be higher than τ_0. Thus

$$c_1 = W_0 \frac{\tau_0\gamma_0 - \tau_1\gamma_1}{\tau_0 - \tau_1}\,, \tag{4}$$

$$c_2 = W_0 \frac{\tau_0\tau_1(\gamma_1 - \gamma_0)}{\tau_0 - \tau_1} \quad \text{and} \tag{5}$$

$$p = 1/\left(W_0\gamma_0\tau_0^2\right)\,. \tag{6}$$

We decide that if p is higher than some nominal value p_0, the desired target bounds can not be guaranteed by the chosen parametrization. Typically, $\tau_0 < \tau_1$ and $\gamma_0 > \gamma_1$, such that $c_2 > 0$. Note that if c_1 is negative, EUPAC will be 0 for all possible sets of regrets because then there are regret bounds τ for which $\bar{F}$ is negative while the minimum of $N_\tau(\Omega)$ cannot be negative by definition. In practice, we therefore cut the upper bound by $\max(\bar{F}, 0)$. We assume all of the former to be true for all further considerations. Conditioned on ordered sets of binned regrets $\Omega_{W,k} = \{\Delta_1, \Delta_2, \ldots, \Delta_i, \ldots, \Delta_W\}_k$, EUPAC is

$$\mathrm{EUPAC}(\Omega) = \sum_k \mathbb{P}(\Omega_{W,k})\mathbb{P}\left(\forall\tau > 0 :\, N_\tau(\Omega_{W,k}) \leq \max\left(\bar{F}(1/\tau),\, 0\right)\right)\,. \tag{7}$$

The number of τ-regrets in $\Omega_{W,k}$ is at most W, gets reduced by 1 at any $\tau = \Delta_i$ and cannot be lower than 0. Therefore

$$N_\tau(\Omega_{W,k}) = \begin{cases} W & ,\ 0 \leq \tau < \Delta_1 \\ W - i & ,\ \exists i : \Delta_i < \tau < \Delta_{i+1} \\ 0 & ,\ \tau > \Delta_W\,. \end{cases} \tag{8}$$

To calculate $\mathbb{P}(\forall\tau : \cdot)$, we look at its complement (Fig. 1). The existence of one such

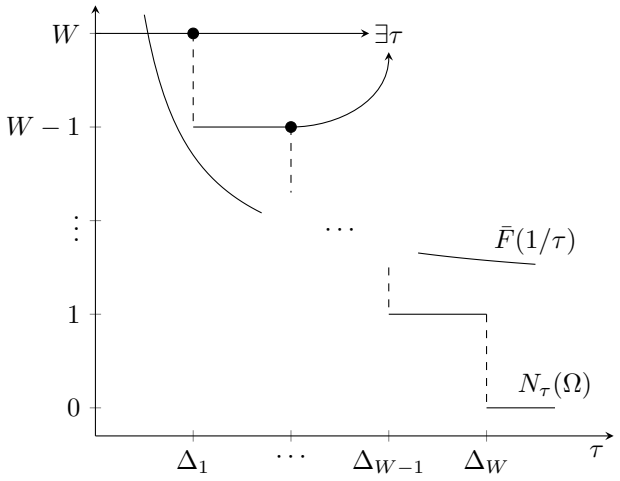

Figure 1: Example evaluation of the EUPAC criterion for $N_\tau(\Omega)$ by (8) and $\bar{F}(1/\tau)$ with $\Omega = \Omega_{W,k}$

$\exists\tau : \neg E$	$C_1^F : \bar{F} = c_1 + c_2/\tau > 0$	$C_2^F : \bar{F} = c_1 + c_2/\tau < 0$
$C_1^\tau : 0 < \tau < \Delta_1$	$C_{11}^{\text{EUPAC}} : W > c_1 + c_2/\tau$	$C_{12}^{\text{EUPAC}} : W > 0$
$C_i^\tau : \Delta_i < \tau < \Delta_{i+1}$	$C_{i1}^{\text{EUPAC}} : W - i > c_1 + c_2/\tau$	$C_{i2}^{\text{EUPAC}} : W - i > 0$
$C_W^\tau : \tau > \Delta_W$	$C_{W1}^{\text{EUPAC}} : 0 > c_1 + c_2/\tau$	$C_{W2}^{\text{EUPAC}} : 0 > 0$

Table 1: Conditions to be checked

τ under all former conditions results in a set of easily checkable interval conditions for EUPAC parameters and Δ_i (see Table 1). That is

$$\exists\tau : \neg E = \bigvee_{i,j}^{W,2} C_i^\tau \wedge C_j^{\bar{F}} \wedge C_{ij}^{\text{EUPAC}} \,. \tag{9}$$

If for a given parametrization and Δ_i those yield true, then $\mathbb{P}(\forall\tau : \cdot) = 0$ such that the corresponding regret probability does not contribute to EUPAC. That is, regrets that are not safe by the UPAC criterion will not change the safety value. Vice-versa, regrets that are safe by the UPAC criterion will change the safety value by their probability of occurence. This alludes to an intuitive understanding of the EUPAC heuristic.

2 Detailed Algorithm

In Algorithm 1 you can find the detailed algorithm of EUPAC by Interval Checking. EUPAC settings are according to Table 2.

Algorithm 1 EUPAC by Interval Checking

Require: k, D, α, $\mathbb{P}_\Delta^0$, W, c_1, c_2, N_{MN}
Ensure: $p < p_0$
  Observe new regret window $\Omega_{W',k}$
  count $\leftarrow 0 \in \mathbb{R}^{W+2}$
  for all $\Delta_i \in \Omega_{W',k}$ **do**
    $\Delta_{D,i} \leftarrow \Delta_D(\Delta_i)$
    $\text{count}(\Delta_{D,i}) \leftarrow \text{count}(\Delta_{D,i}) + 1$
  end for
  $\mathbb{P}'_\Delta \leftarrow \frac{\text{count}(\Delta)}{\sum_\Delta \text{count}(\Delta)}$
  $\mathbb{P}_\Delta^k \leftarrow \alpha\mathbb{P}_\Delta^{k-1} + (1-\alpha)\mathbb{P}'_\Delta$
  Calculate EUPAC by Interval Checking (7) across all multinomial cases
  EUPAC $\leftarrow 0$
  for all $n \in \mathbb{R}^{W+2} : n_0 + n_1 + n_2 + \ldots + n_W + n_{W+1} = N_{\text{MN}}$ **do**
    EUPAC $\leftarrow$ EUPAC $+ \text{MN}(N_{\text{MN}}, n, \mathbb{P}_\Delta^k)(1 - \text{IC}(\Delta(n)))$
  end for

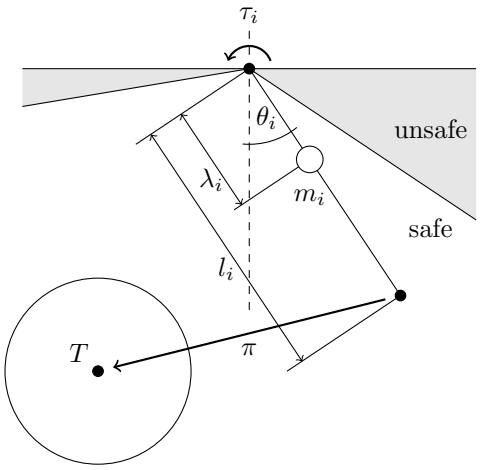

Figure 2: Benchmark system

3 The Benchmark System

The proposed pendulum systems (see Fig. 2) are typical models for sagittal leg movement in exoskeletons as we wanted to allude to those and the similar Reacher environment of reinforcement learning benchmark systems [1]. System parameters were therefore chosen to represent human-like properties for the masses, lengths and center-of-gravitations of shank, thigh and foot respectively with an arbitrary weight of 80 kg and total body height of 1.8 m [2, 3]. Parameters for lower degreee of freedom systems were calculated from the system with 3 degrees of freedom by subsequently looking at the combined center of mass dynamics. For the impedance, stiffness is not introduced, whereas velocity damping values are set to be 10^{2-i} for each joint i respectively. Joint angles clip outside of ± 7rad, joint angle velocities outside of $\pm 5\,\mathrm{rad}/s$ and joint control torques outside of $\pm 100\,Nm$.

The benchmark ran on a Intel i7-10870H with 2.2 GHz and a GeForce RTX 3060.

Parameter	Regret by Reward	Regret by Observation
Lower Regret τ_0	50	0.01
Lower Proportion of Reference Window γ_0	0.8	0.8
Upper Regret τ_1	1000	0.08
Upper Proportion of Reference Window γ_1	0.0	0.0
Reference Window W_0	100	100
Binned Regret Window W	30	30
Regret Density Floating α	0.5	0.5
Multinomial Sample Window N_{MN}	3	3

Table 2: Settings of all relevant parameters in EUPAC by Interval Checking

Implementations use Python with Numpy, Scipy Optimization, Tensorflow [4], and Stable Baselines [5]. Reacher environments are made compatible with OpenAI Gym [1]. Both the Reacher environment and EUPAC by Interval Checking are available at /github.com/flxweiske/eupac.

4 Learning Results

For learning the environments, it was necessary to see that the environment 1) can be learned and 2) gets progressively harder to learn. Looking at the learning curves differentiated by different aspects under the image folder, it is clear that all agents across all environments and algorithms succeed to minimize regret across varied learning results. It is of note that learning without CBF guidance has a higher variance around the average learning of the resulting regrets by reward than those with CBF guidance. Also, the worst regrets without CBF guidance are around 3000 with a lot more outlier learning curves compared to those with CBF guidance with worst regrets at around 2000 and less outlier learning curves. We infer that learning without CBF guidance is harder with respect to result in a safe agent behaviour since unsafe behaviour is pre-emptively allowed for. Comparing the reinforcement algorithms shows that DDPG has the worst overall regret by reward and the highest variance around the highest average learning curve whereas TRPO is the best for all of it. SAC is inbetween both. This consolidates previous results for RL algorithm benchmarks [6, 7, 8]. With higher DOF the proposed pendulum systems show worse results for worst overall regrets and variance around the average learning curve. In all, these results lay a solid foundation for the evidence of following EUPAC results.

References

[1] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016.

[2] David A Winter. *Biomechanics and Motor Control of Human Movement*. John Wiley & Sons, 2009.

[3] Stanley Plagenhoef, F Gaynor Evans, and Thomas Abdelnour. Anatomical data for analyzing human motion. *Research quarterly for exercise and sport*, 54(2):169–178, 1983.

[4] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig

Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[5] Ashley Hill, Antonin Raffin, Maximilian Ernestus, Adam Gleave, Anssi Kanervisto, Rene Traore, Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, and Yuhuai Wu. Stable baselines, 2018.

[6] Yan Duan, Xi Chen, Rein Houthooft, John Schulman, and Pieter Abbeel. Benchmarking deep reinforcement learning for continuous control. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1329–1338, New York, New York, USA, 20–22 Jun 2016. PMLR.

[7] Riashat Islam, Peter Henderson, Maziar Gomrokchi, and Doina Precup. Reproducibility of benchmarked deep reinforcement learning tasks for continuous control, 2017.

[8] Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), April 2018.