

Asymptotically Stable Adaptive–Optimal Control Algorithm With Saturating Actuators and Relaxed Persistence of Excitation

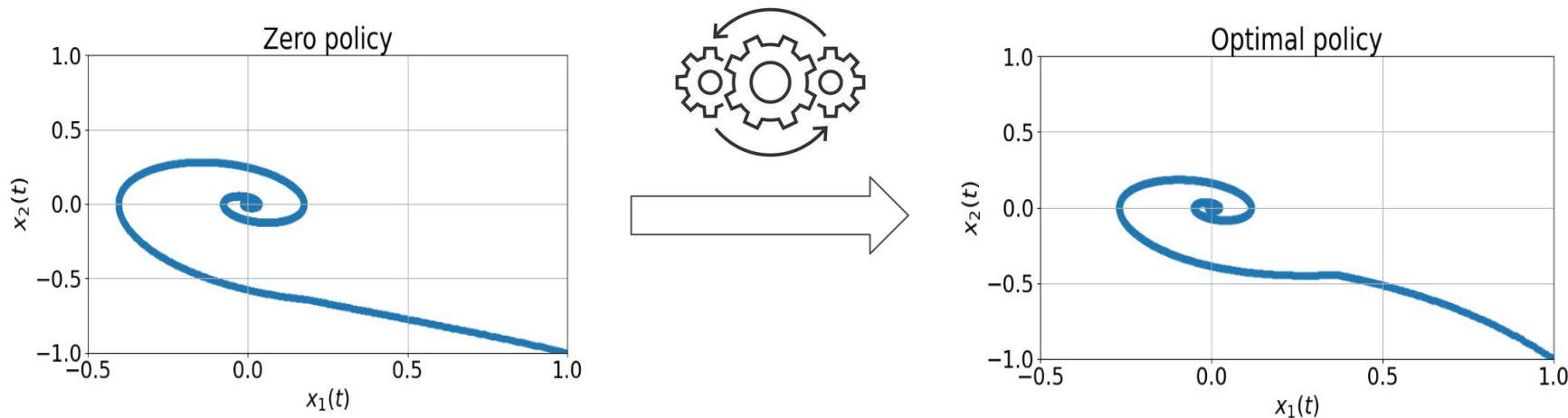
By Kyriakos G. Vamvoudakis, Member, IEEE, Marcio Fantini Miranda, and João P. Hespanha, Fellow, IEEE

Leshchev Dmitrii
Mezentsev Gleb

Skoltech 2021

Summary of the project

The contributions of this project lie in the implementation of an adaptive learning algorithm to solve an infinite-horizon optimal control problem for known deterministic nonlinear systems, while considering symmetric input constraints



Problem studied and relevant theory

Nonlinear continuous-time system

$$\begin{aligned}\dot{x}(t) &= f(x(t)) + g(x(t))u(t); \\ x(0) &:= x_0, \quad t \geq 0\end{aligned}$$

Penalty function (in matrix form)

$$\begin{aligned}R_s(u) &= 2 \int_0^u (\theta^{-1}(v))^T R dv \\ &:= 2 \int_0^u (\bar{u} \tanh^{-1}(v/\bar{u}))^T R dv > 0 \quad \forall u.\end{aligned}$$

Optimal value function

$$V^*(x(t)) = \min_{u \in U} \int_t^\infty r(x, u) d\tau \quad \forall x, \quad t \geq 0$$

Infinite horizon loss-function

$$\begin{aligned}V(x(0)) &= \int_0^\infty r(x(\tau), u(\tau)) d\tau \quad \forall x(0) \\ r(x, u) &= Q(x) + R_s(u) \quad \forall x, u\end{aligned}$$

Penalty function

$$R_s(u) = 2 \sum_{i=1}^m \int_0^{u_i} (\theta^{-1}(v_i))^T q_i dv_i \quad \forall u$$

Problem studied and relevant theory

HJB equation with optimal cost and optimal control

$$H^*(x, \bar{\mathcal{K}}^*(x), \nabla V^*(x)) := \nabla V^*(x)^T (f(x) + g(x)\bar{\mathcal{K}}^*(x)) + Q(x) + R_s(\bar{\mathcal{K}}^*(x)) = 0 \quad \forall x.$$

$$\bar{\mathcal{K}}^*(x) = W_u^{*T} \phi_u(x) + \epsilon_u(x) \quad \forall x$$

Optimal value function

Optimal control policy

$$V^*(x) = W^* \phi(x) + \epsilon(x) \quad \forall x$$

Methods: Critic

Optimal value function

Gradient-descent-like rule for
Critic NN tuning

$$V^*(x) = W^{*T} \phi(x) + \epsilon(x) \quad \forall x$$

$$\begin{aligned} \dot{\hat{W}} &= -\alpha \frac{\partial E}{\partial \hat{W}} = -\alpha \frac{\omega(t)e(t)}{(\omega(t)^T \omega(t) + 1)^2} - \alpha \sum_{i=1}^k \frac{\omega(t_i)e_{\text{buff}_i}(t_i, t)}{(\omega(t_i)^T \omega(t_i) + 1)^2} \\ &= -\alpha \frac{\omega(t)(\omega(t)^T \hat{W}(t) + R_s(u(t)) + Q(x(t)))}{(\omega(t)^T \omega(t) + 1)^2} \\ &\quad - \alpha \sum_{i=1}^k \frac{\omega(t_i)(\omega(t_i)^T \hat{W}(t) + Q(x(t_i)) + R_s(u(t_i)))}{(\omega(t_i)^T \omega(t_i) + 1)^2} \end{aligned}$$

where $\omega(t_i) := \nabla \phi(x(t_i)) (f(x(t_i)) + g(x(t_i))u(t_i))$.



Methods: Actor

Optimal control policy

Gradient-descent-like rule for
Critic NN tuning

$$\bar{\mathcal{K}}^*(x) = W_u^{*T} \phi_u(x) + \epsilon_u(x) \quad \forall x$$

$$\begin{aligned} \dot{\hat{W}}_u &= -\alpha_u \frac{\partial E_u}{\partial \hat{W}_u} = -\alpha_u \phi_u e_u \\ &= -\alpha_u \phi_u \left(\hat{W}_u^T \phi_u + \theta \left(\frac{1}{2} R^{-1} g^T(x) \nabla \phi^T \hat{W} \right) \right)^T \end{aligned}$$

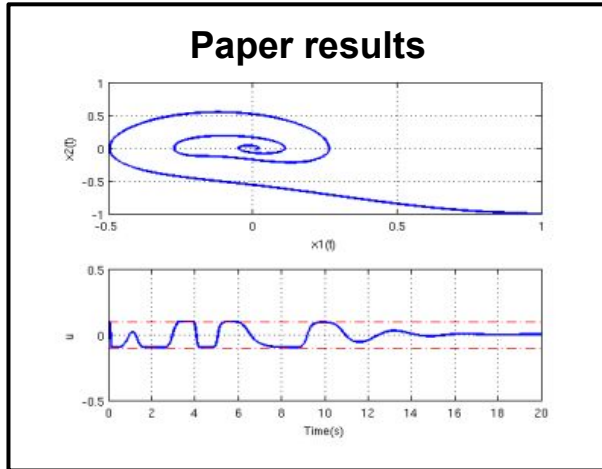
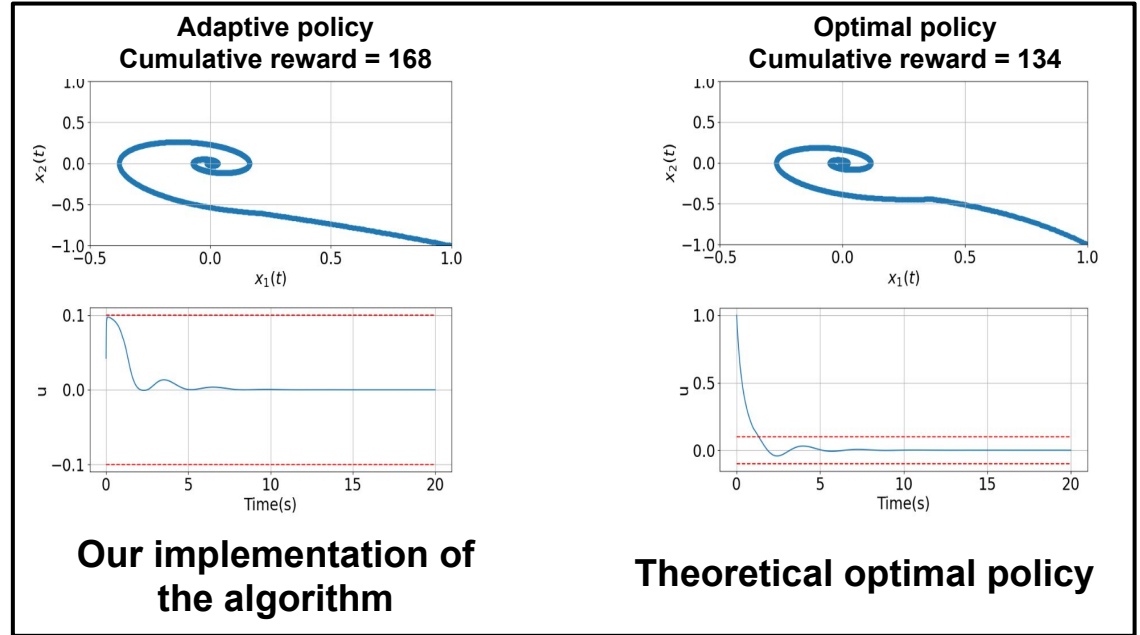
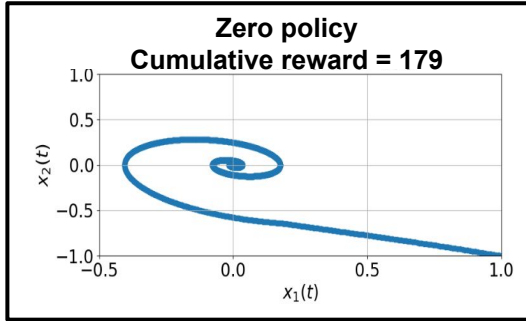


Methods. Algorithm

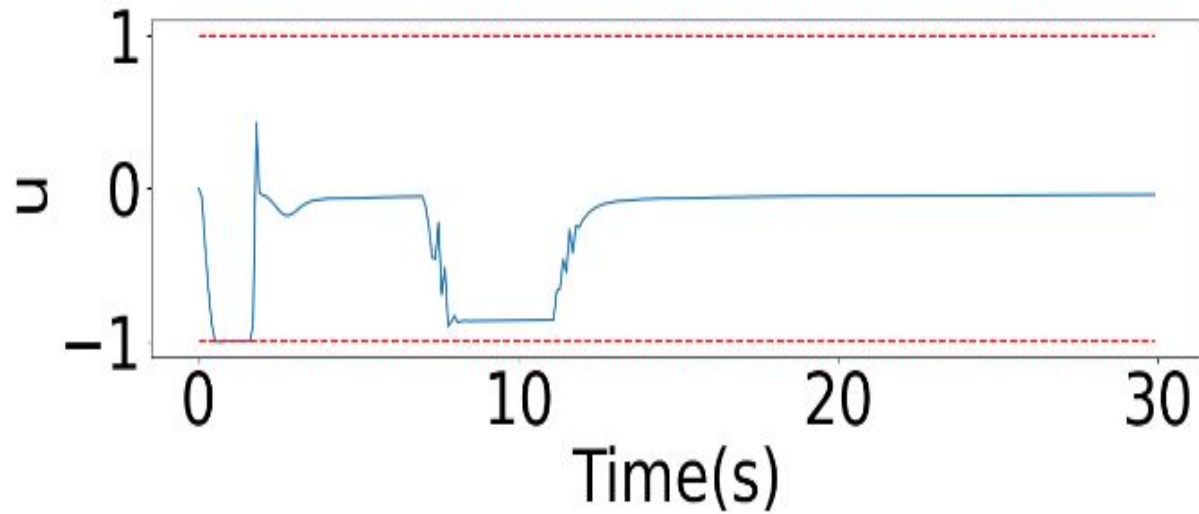
Adaptive-Optimal Control Algorithm With Relaxed PE

1. Start with initial state $x(0)$, random initial weights $\hat{W}_u(0)$, $\hat{W}(0)$ and $i = 1$
2. **procedure**
3. Propagate $t, x(t)$ using (1) and $u(t) := \hat{\mathcal{K}}(x) \triangleright \{x(t)$ comes from integrating the nonlinear system (1) using any ordinary differential equation (ode) solver (e.g. Runge Kutta) while the time t comes from the Runge Kutta integration process, i.e. $[t_i, t_{i+1}]$, $i \in \mathcal{N}$ where $t_{i+1} := t_i + h$ with $h \in \mathbb{R}^+$ the step size}
4. Propagate $\hat{W}_u(t)$, $\hat{W}(t) \triangleright \{\text{integrate } \hat{W}_u \text{ as in (31) and } \dot{\hat{W}} \text{ as in (20) using any ode solver (e.g. Runge Kutta)}\}$
5. Compute $\hat{V}(x) = \hat{W}^T \phi(x) \triangleright$ output of the Critic NN,
6. Compute $\hat{\mathcal{K}}(x) = \hat{W}_u^T \phi_u(x) \triangleright$ output of the Actor NN
7. **If** $i \neq k \triangleright \{\{\omega(t_1), \omega(t_2), \dots, \omega(t_i)\}$ has N linearly independent elements and t_k is the time instant that this happens}
8. Select an arbitrary data point to be included in the history stack (c.f. Remarks 1-2)
9. $i := i + 1$
10. **end if** \triangleright when the history stack is full
11. **end procedure**

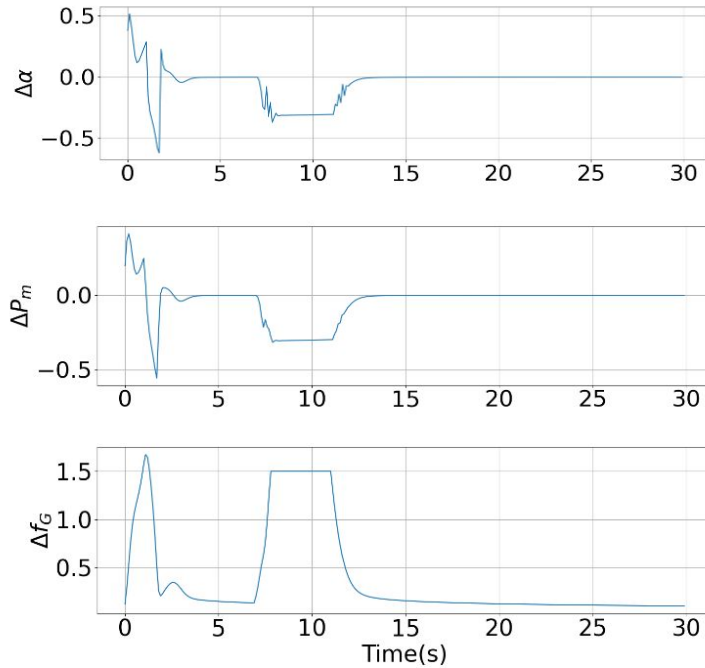
Results. Van der Pol Oscillator



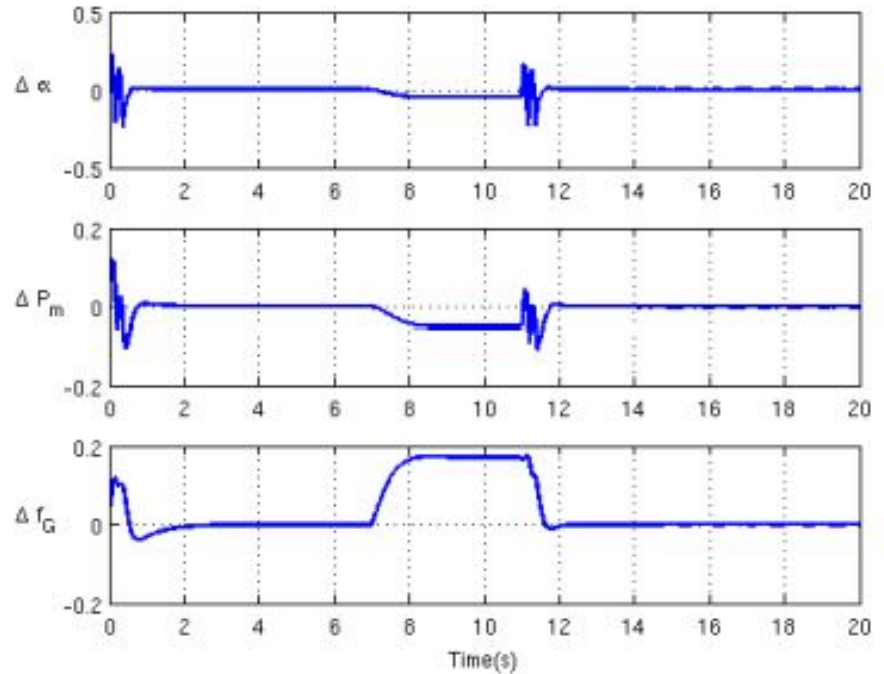
Results.Power Plant System



Results.Power Plant System



Our implementation



Paper results

Github

The screenshot shows a GitHub repository page for the user 'Glebzok' and the repository 'ASAOCAwSAaRPoE'. The page is viewed on the 'master' branch, which has 1 branch and 0 tags. The repository contains 13 commits. A recent commit by Glebzok added a README file and several Python files. The repository's description is 'Asymptotically Stable Adaptive-Optimal Control Algorithm With Saturating Actuators and Relaxed Persistence of Excitation'. The repository is primarily composed of Python code, as indicated by the 'Languages' section showing 100.0% Python.

github.com/Glebzok/ASAOCAwSAaRPoE

master 1 branch 0 tags

Go to file Add file Code

Glebzok added README 2c229df 7 hours ago 13 commits

img	added README	7 hours ago
README.md	added README	7 hours ago
actor.py	pretty version with experiments	8 hours ago
algorithm.py	pretty version with experiments	8 hours ago
critic.py	pretty version with experiments	8 hours ago
environment.py	pretty version with experiments	8 hours ago
experiments.py	pretty version with experiments	8 hours ago
requirements.txt	not initial commit	2 days ago

README.md

Asymptotically Stable Adaptive-Optimal Control Algorithm With Saturating Actuators and Relaxed Persistence of Excitation

This final "Reinforcement Learning course" project focuses on the implementation of the algorithm presented in [paper](#).

We have implemented the algorithm and tested its performance on two environments:

About

Asymptotically Stable Adaptive-Optimal Control Algorithm With Saturating Actuators and Relaxed Persistence of Excitation

Readme

Releases

No releases published

Packages

No packages published

Languages

Python 100.0%

<https://github.com/Glebzok/ASAOCAwSAaRPoE>

Conclusion

- **Implemented the algorithm**
- **Checked its efficiency on two environments**
- **Achieved improvements in convergence in both cases**
- **Satisfied all initial constraints in both cases**

Thank you for your
attention!