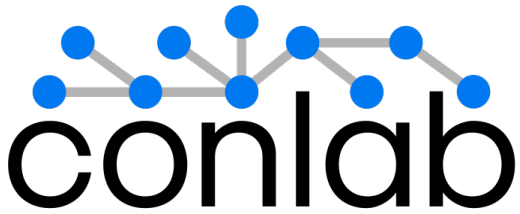
Policy Iteration for Multiplicative Noise Output Feedback Control

Benjamin Gravell, Matilde Gargiani,
John Lygeros, and Tyler Summers





61st IEEE Conference on Decision and Control
Cancún, Mexico - December 6-9, 2022

So you thought linear systems were easy?

Consider two complicating factors

Random dynamics

$$x_{t+1} = A_t x_t + B_t u_t + w_t$$

A_t, B_t, C_t, w_t, v_t are iid random

Random partial observability

$$y_t = C_t x_t + v_t$$

Limited view of internal state

Multiplicative Noise Output Feedback Optimal Control (MOFOC)

$$\underset{\pi}{\text{minimize}} \quad \lim_{T\to\infty} \mathbb{E}\frac{1}{T}\sum_{t=0}^{T-1} \begin{bmatrix} x_t \\ u_t \end{bmatrix}^\intercal \begin{bmatrix} Q_{xx} & Q_{xu} \\ Q_{ux} & Q_{uu} \end{bmatrix} \begin{bmatrix} x_t \\ u_t \end{bmatrix}$$

$$\text{subject to} \quad \begin{bmatrix} x_{t+1} \\ y_t \end{bmatrix} = \begin{bmatrix} A_t & B_t \\ C_t & 0 \end{bmatrix} \begin{bmatrix} x_t \\ u_t \end{bmatrix} + \begin{bmatrix} w_t \\ v_t \end{bmatrix}$$

Complicating factors **interact**:
Optimal linear policy is a **coupled** controller and state estimator

~~**separation principle**~~

A short and sweet summary

Prior work:

- Value iteration for MOFOC [1] - *slow*
- Policy iteration for LQG [2, 3, 4] - *no multiplicative noise*

This work:
Policy iteration for MOFOC

Who cares?

1. Solve MOFOC problems faster
2. Open the door to novel approximate policy iteration algorithms

Closed-loop dynamics under linear policies

Linear Dynamic Controller (F, K, L)

$$\hat{x}_{t+1} = F\hat{x}_t + Ly_t \qquad \text{(state estimator)}$$
$$u_t = K\hat{x}_t \qquad \text{(feedback control)}$$

Using (F, K, L), the closed-loop augmented state dynamics are

$$\begin{bmatrix} x_{t+1} \\ \hat{x}_{t+1} \end{bmatrix} = \underbrace{\begin{bmatrix} A_t & B_t K \\ LC_t & F \end{bmatrix}}_{\Phi_t'} \begin{bmatrix} x_t \\ \hat{x}_t \end{bmatrix} + \begin{bmatrix} I_n & 0 \\ 0 & L \end{bmatrix} \begin{bmatrix} w_t \\ v_t \end{bmatrix}$$

Why restrict search to linear policies?

Optimal policy in this class can be computed from Riccati equations.

Stochastic stability concepts

Definition 1 (Mean-square stability [1])

The second moment of the augmented state converges to a finite positive semidefinite matrix, i.e.

$$\lim_{t\to\infty} \mathbb{E}\left[\begin{bmatrix} x_t \\ \hat{x}_t \end{bmatrix}\begin{bmatrix} x_t \\ \hat{x}_t \end{bmatrix}^{\mathsf{T}}\right] = S' \succeq 0$$

Definition 2 (Mean-square compensatability [1])

There exists a controller which renders the system ms-stable.

Assumption: The system is mean-square compensatable.

Building block: policy evaluation

If (F, K, L) is ms-stabilizing, the **generalized discrete-time Lyapunov equations (GDLYAP)** have well-defined solutions:

$$P' = \mathbb{E}\left[\Phi_t'^{\mathsf{T}} P' \Phi_t'\right] + Q' \qquad \text{(steady-state value function)}$$

$$S' = \mathbb{E}\left[\Phi_t' S' \Phi_t'^{\mathsf{T}}\right] + W' \qquad \text{(steady-state second moment)}$$

How to solve GDLYAP?

- Vec-kron trick to convert to a system of linear equations
- Specialized Krylov subspace techniques [5]

Later we need an array of four matrices

$$X = (P, \hat{P}, S, \hat{S})$$

which picks and adds certain blocks of (P', S')

Building block: policy improvement

Gain operators

$$\mathcal{K}(X) := -\mathcal{G}_{uu}^{-1}(X)\mathcal{G}_{ux}(X),$$
$$\mathcal{L}(X) := \mathcal{H}_{xy}(X)\mathcal{H}_{yy}^{-1}(X).$$

generate an improved policy based on the value array X

$\mathcal{G}(\cdot)$ and $\mathcal{H}(\cdot)$ are affine operators

- Represent the "Q-function" that characterizes the cost of taking a control action + state estimate update while being in a given augmented state, then following (F, K, L) forever after
- Too big to fit on this slide

Optimal policy design equations

Solve the coupled nonlinear matrix Riccati equations [1]

$$\mathcal{R}(X) := \begin{pmatrix} -P + \mathcal{G}_{xx}(X) - \mathcal{G}_{xu}(X)\mathcal{G}_{uu}^{-1}(X)\mathcal{G}_{ux}(X) \\ -\hat{P} + \mathcal{E}(X) + \mathcal{G}_{xu}(X)\mathcal{G}_{uu}^{-1}(X)\mathcal{G}_{ux}(X) \\ -S + \mathcal{H}_{xx}(X) - \mathcal{H}_{xy}(X)\mathcal{H}_{yy}^{-1}(X)\mathcal{H}_{yx}(X) \\ -\hat{S} + \mathcal{F}(X) + \mathcal{H}_{xy}(X)\mathcal{H}_{yy}^{-1}(X)\mathcal{H}_{yx}(X) \end{pmatrix} = 0$$

where

$$\mathcal{E}(X) := (A - \mathcal{L}(X)C)^\intercal \hat{P}(A - \mathcal{L}(X)C)$$
$$\mathcal{F}(X) := (A + B\mathcal{K}(X))\hat{S}(A + B\mathcal{K}(X))^\intercal$$

Then the optimal policy (F^*, K^*, L^*) is

$$K^* = \mathcal{K}(X)$$
$$L^* = \mathcal{L}(X)$$
$$F^* = A + BK^* - L^*C$$

A tale of two algorithms

Value iteration [1] - state-of-the-art since 1992, but *slow*

$$X^{k+1} = X^k + \mathcal{R}(X^k)$$

Motivating question

Can we solve the optimal policy design equations more quickly?

Key ideas:

Policy iteration is an instance of the provably rapidly convergent Newton's method in the fully observed setting [6, 7]

Policy evaluation and policy improvement steps can be generalized to the partially observed setting

Stands to reason that generalizing policy iteration to the partially observed setting would yield rapid convergence

Policy iteration algorithm for MOFOC

Input

ms-stable policy (F^0, K^0, L^0)

threshold ϵ

Policy Evaluation

$X^k \leftarrow \text{GDLYAP}(F^k, K^k, L^k)$

while $\|X^{k+1} - X^k\| > \epsilon$

Policy Improvement

$$K^{k+1} = \mathcal{K}(X^k)$$
$$L^{k+1} = \mathcal{L}(X^k)$$
$$F^{k+1} = A + BK^{k+1} - L^{k+1}C$$

Output

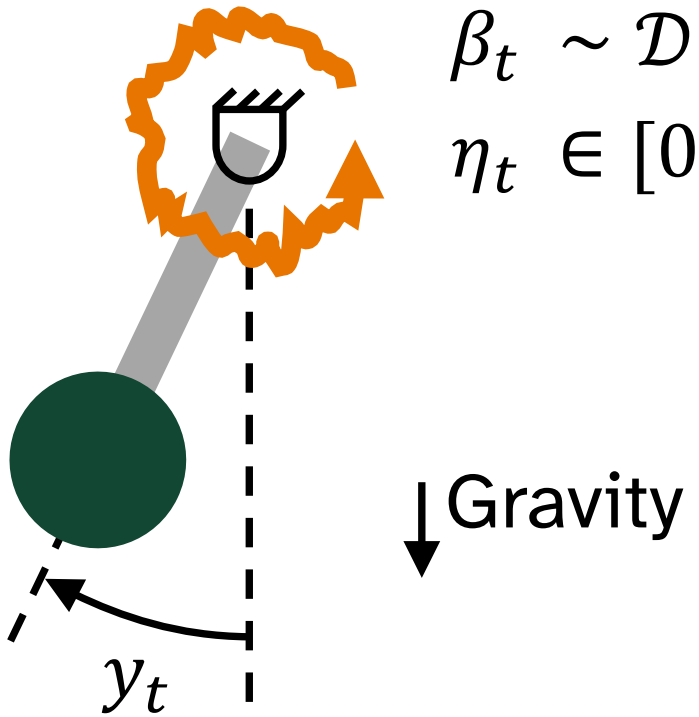
ϵ-optimal policy (F^k, K^k, L^k)

Experiment: pendulum system, setup

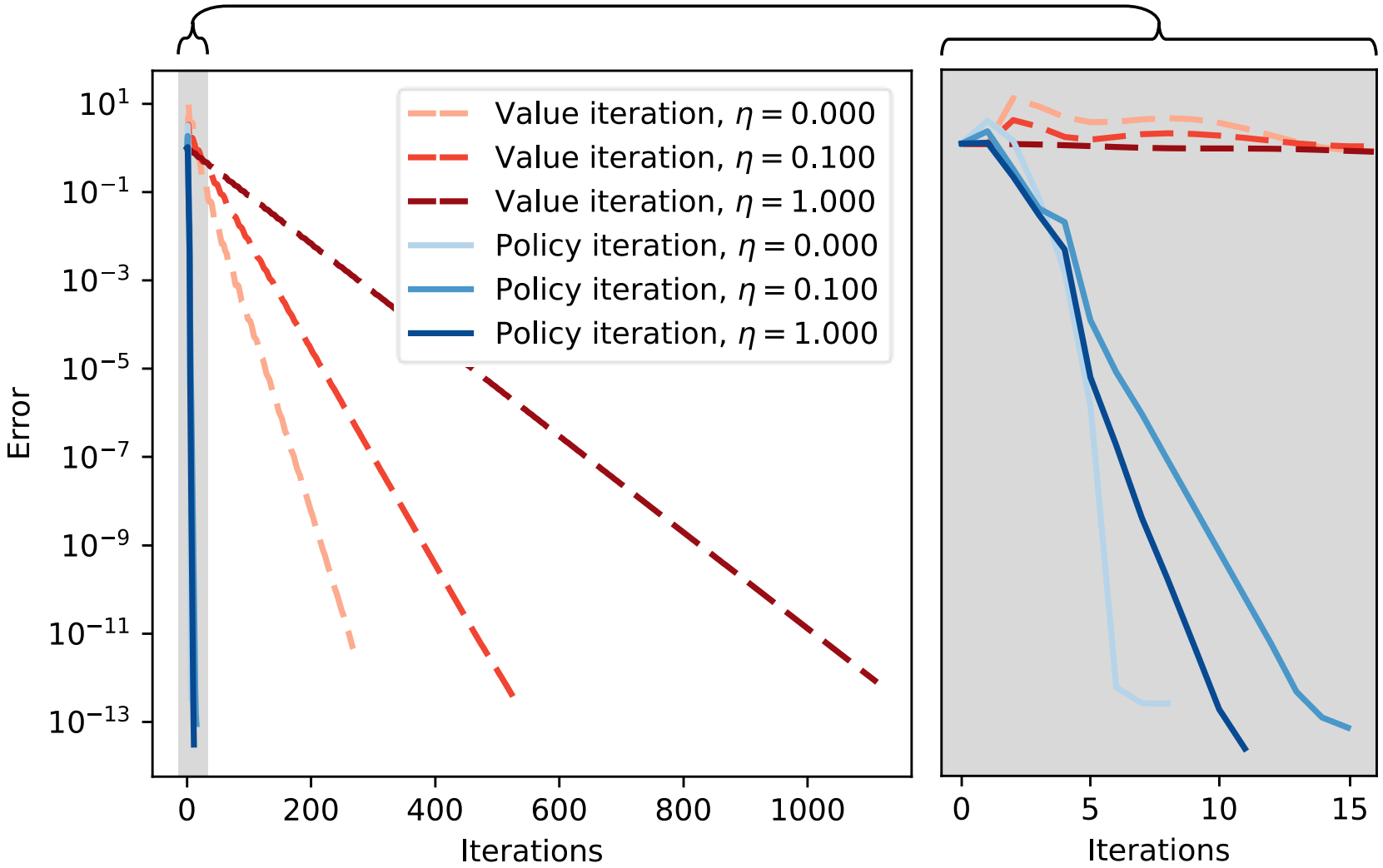
$$\text{Torque} = (0.1 + \beta_t)u_t$$

$$\beta_t \sim \mathcal{D}(0, \eta^2)$$

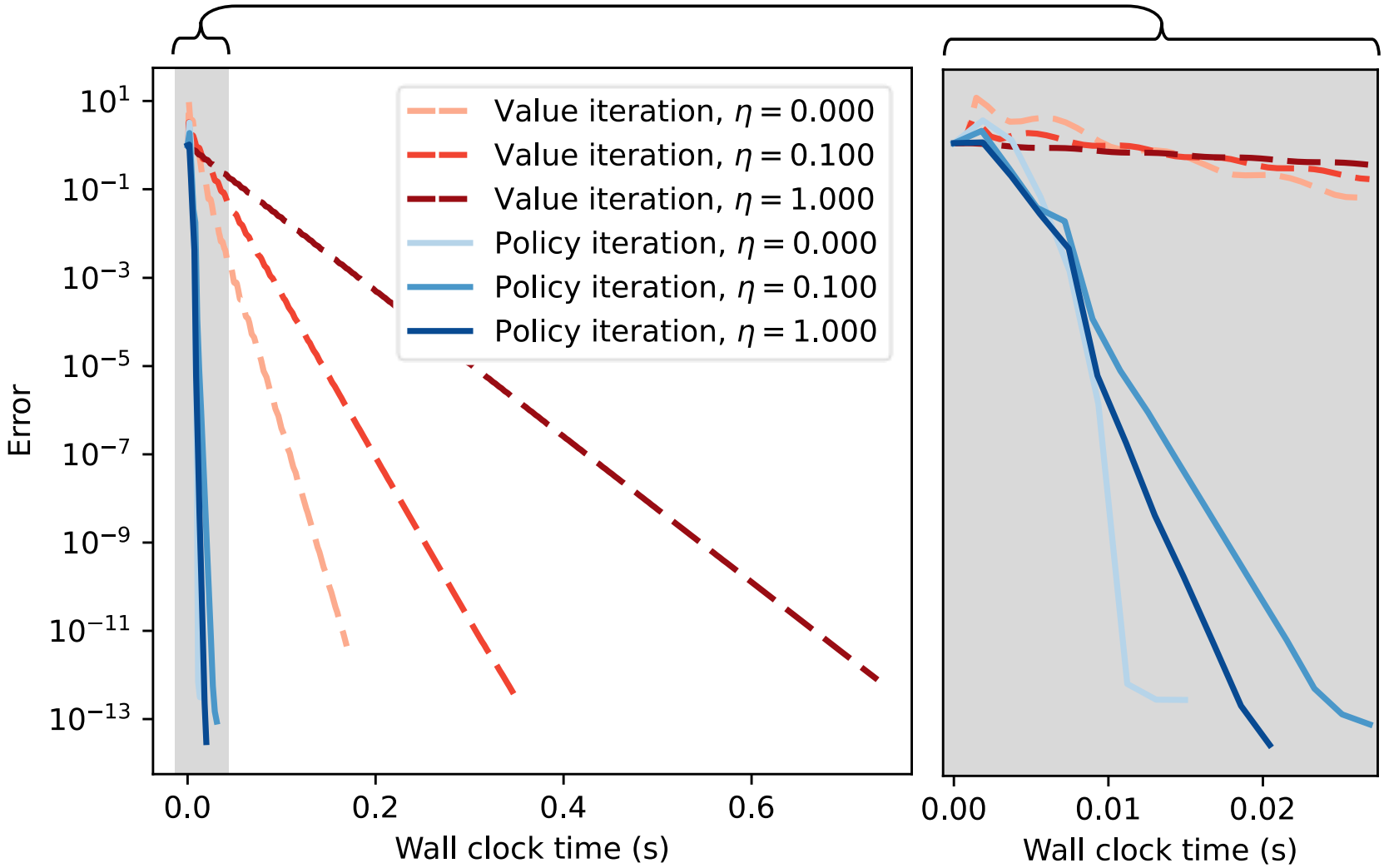
$$\eta_t \in [0, 1]$$



Experiment: pendulum system, iterations



Experiment: pendulum system, clock time



Experiment: random systems, setup

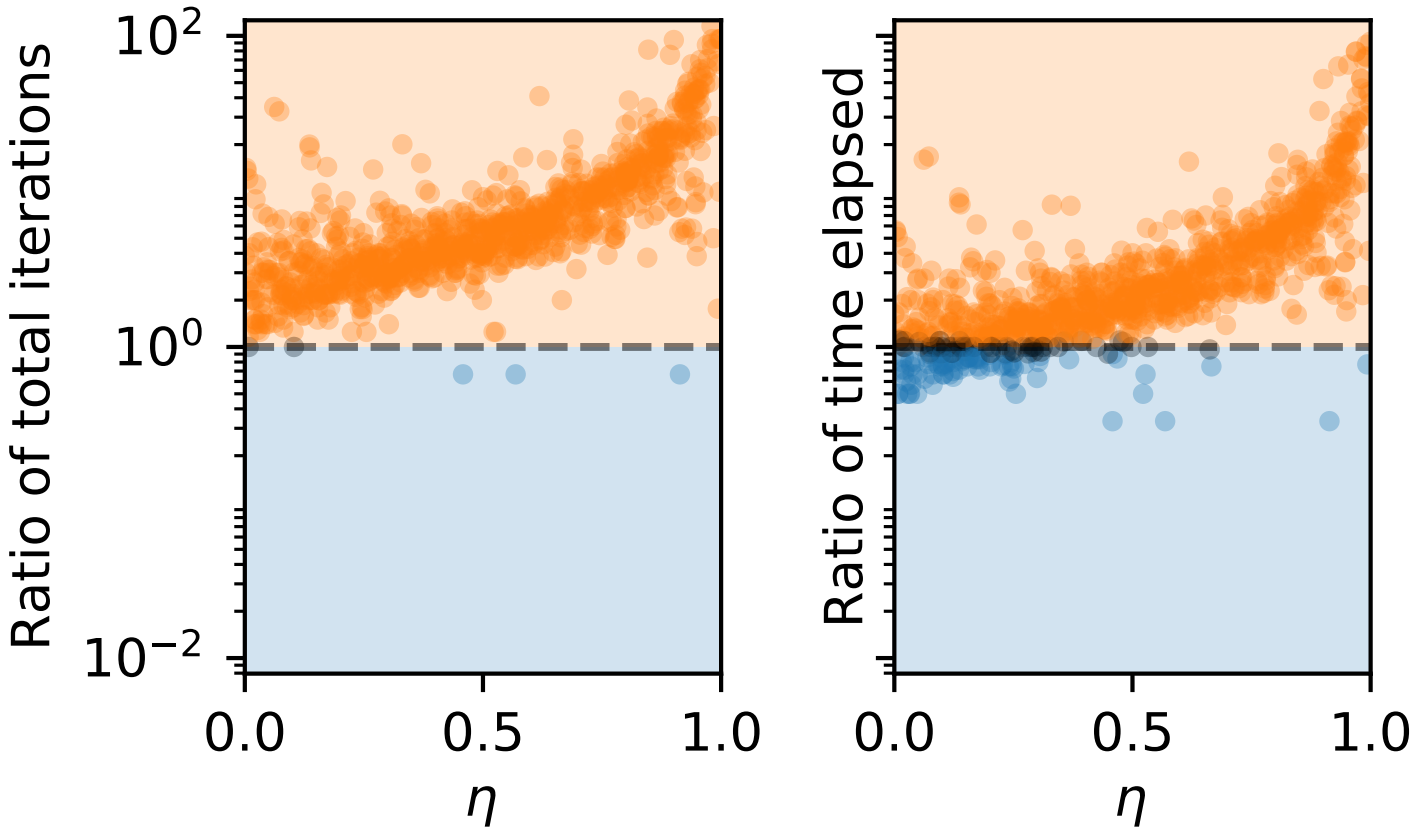
2 states, 1 input, 1 output

Problem data randomly generated from standard normal distributions

System matrices and noise levels scaled to get open-loop ms-stability
$\rightarrow$ Initial policy $(A, 0, 0)$ is valid

Noise level η drawn randomly between 0 and 1 (critically ms-stable)

Value iteration vs policy iteration



Parting thoughts

Summary

We proposed a policy iteration algorithm for multiplicative noise optimal output feedback control design that, empirically, converges far faster than value iteration.

Future work

- Formalize connection to Newton method & leverage this view to prove a quadratic rate of convergence like in [8, 9, 10]
- Investigate novel approximate policy iteration algorithms that use input-output data to approximately execute policy evaluation and policy improvement steps to solve POMDPs.

Python code available at
https://github.com/TSummersLab/policy-iteration-mlqc

Thanks!

Contact information:

Ben Gravell	bjgravell@gmail.com
Matilde Gargiani	gmatilde@ethz.ch
John Lygeros	jlygeros@ethz.ch
Tyler Summers (PI)	tyler.summers@utdallas.edu











This material is based on work supported by:

- The United States Air Force Office of Scientific Research under award FA2386-19-1-4073
- The National Science Foundation under award ECCS-2047040
- The European Research Council under OCAL project 787845

Bibliography I

[1] W. L. De Koning.
Compensatability and optimal compensation of systems with white parameters.
IEEE Transactions on Automatic Control, 37(5):579–588, May 1992.

[2] F. L. Lewis and Kyriakos G. Vamvoudakis.
Reinforcement learning for partially observable dynamic processes: Adaptive dynamic programming using measured output data.
IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 41(1):14–25, 2011.

[3] Syed Ali Asad Rizvi and Zongli Lin.
Output feedback reinforcement learning control for the continuous-time linear quadratic regulator problem.
In *2018 Annual American Control Conference (ACC)*, pages 3417–3422, 2018.

Bibliography II

[4] Farnaz Adib Yaghmaie, Fredrik Gustafsson, and Lennart Ljung.
Linear quadratic control using model-free reinforcement learning.
IEEE Transactions on Automatic Control, pages 1–1, 2022.

[5] Tobias Damm.
Direct methods and ADI-preconditioned Krylov subspace methods for generalized Lyapunov equations.
Numerical Linear Algebra with Applications, 15(9):853–871, 2008.

[6] Martin L Puterman and Shelby L Brumelle.
On the convergence of policy iteration in stationary dynamic programming.
Mathematics of Operations Research, 4(1):60–69, 1979.

[7] Dimitri Bertsekas.
Lessons from alphazero for optimal, model predictive, and adaptive control.
arXiv preprint arXiv:2108.10315, 2021.

Bibliography III

[8] David Kleinman.
On an iterative technique for Riccati equation computations.
IEEE Transactions on Automatic Control, 13(1):114–115, 1968.

[9] G. Hewer.
An iterative technique for the computation of the steady state gains for the discrete optimal regulator.
IEEE Transactions on Automatic Control, 16(4):382–384, 1971.

[10] T. Damm and D. Hinrichsen.
Newton's method for a rational matrix equation occurring in stochastic control.
Linear Algebra and its Applications, 332-334:81–109, 2001.