A modular framework for stabilizing deep reinforcement learning control

IFAC World Congress 2023

Nathan Lawrence ~ University of British Columbia ~ lawrence@math.ubc.ca
Philip Loewen, Shuyuan Wang, Michael Forbes, Bhushan Gopaluni
Maximizing reward through experience

**Reinforcement learning**

ChatGPT

Reinforcement learning

ChatGPT

Maximizing reward through experience

https://openai.com/research/learning-from-human-preferences


https://innermonologue.github.io/

RL in PSE?

A review On reinforcement learning: Introduction and applications in industrial process control
Rui Nian, Jinfeng Liu*, Biao Huang

Toward self-driving processes: A deep reinforcement learning approach to control
Steven Spielberg1 | Aditya Tulsyan1 | Nathan P. Lawrence2 | Philip D. Loewen2 | R. Bhushan Gopaluni1

Reinforcement Learning – Overview of recent progress and implications for process control
Joohyun Shin*, Thomas A. Badgwellb, Kuang-Hung Liub, Jay H. Leea,*
Can reinforcement learning help maintain control loops?
It’s complicated

In favor

• Leverage observed data to improve operations
• Minimize prior domain knowledge
• Automated maintenance on a variety of systems

Against

- Additional algorithmic complexity
- Auto-tuners exist already (but are often idle)
- Stability during and after training
- Sample efficiency

Our goal is to balance the automation and scalability of reinforcement learning with control-theoretic tools to create efficient and safe improvements
Reinforcement learning over all stable behaviour

Topics for today

1. Willems’ lemma
   Data-based characterization of dynamics

2. Youla-Kučera parameterization
   Recipe for all stabilizing controllers

3. Learning algorithms
   A module to shape system behavior

*Combining these elements gives a modular setup that decouples learning and stability*
Key ingredients
State-space model

• Define the system equations

\[ x_{t+1} = Ax_t + Bu_t \]
\[ y_t = Cx_t \]

where
\[ A \in \mathbb{R}^{n \times n}, B \in \mathbb{R}^{n \times 1}, C \in \mathbb{R}^{1 \times n} \]

• Inputs, outputs are scalars for simplicity

![State-space model diagram](image)
Willems’ fundamental lemma — a special case
(Picture form)

Impulse

What is the span of these data vectors?
Full-rank data matrix!
Willems’ fundamental lemma — general case

Data ↔ models

- Given a signal $z = \{z_t\}_{t=0}^{N-1}$, define its Hankel matrix of order $L$:

$$
H_L(z) = \begin{bmatrix}
z_0 & z_1 & \cdots & z_{N-L} \\
z_1 & z_2 & \cdots & z_{N-L+1} \\
\vdots & \vdots & \ddots & \vdots \\
z_{L-1} & z_L & \cdots & z_{N-1}
\end{bmatrix}
$$

“Persistently exciting” if full rank

- Let $\{u_t, y_t\}_{t=0}^{N-1}$ be a trajectory where $u$ is persistently exciting of order $L + n$. Then $\{\bar{u}_t, \bar{y}_t\}_{t=0}^{L-1}$ is a trajectory if and only if there exists $\alpha$ such that

$$
\begin{bmatrix}
H_L(u) \\
H_L(y)
\end{bmatrix} \alpha = \begin{bmatrix}
\bar{u} \\
\bar{y}
\end{bmatrix}
$$

Static, collected data

All possible data

A dynamic Willems’ lemma

Carrying a trajectory forward

- Start with Willems’ lemma:
  \[
  \begin{bmatrix}
    H_L(u) \\
    H_L(y)
  \end{bmatrix} \alpha_0 = \begin{bmatrix}
    \tilde{u} \\
    \tilde{y}
  \end{bmatrix}
  \]

- How to advance to the next output? Consider nested Hankel matrices:

\[
\begin{bmatrix}
  y_0 & y_1 & \cdots & y_{N-L} & y_{N-L+1} \\
  y_1 & y_2 & \cdots & y_{N-L+1} & y_{N-L+2} \\
  \vdots & \vdots & \ddots & \vdots & \vdots \\
  y_{L-1} & y_L & \cdots & y_{N-1} & y_N
\end{bmatrix}
\]

\[
H_L(y) = \begin{bmatrix}
  H_L^+(y) \\
  H_L^-(y)
\end{bmatrix}
\]

\[
H_L(y) \alpha' = H_L'(y) \alpha
\]

Multiply previous solution by shifted Hankel matrix — Then repeat!

Figure 2: A visual example of Corollary 3.5 and Theorem 3.4. We collect input–output data for 100 time steps using a standard normal probing signal. After 100 time steps, we take the last \( L \) samples as an initial value, then use recursion outlined in Eq. (5) to "continue" the rollout. This is done several times for different samples of output noise; the shaded regions are the standard deviation from the mean. The bottom figure is the evolution of the spectral radii for the noisy and noise-free matrices.

\[
\begin{bmatrix}
  \tilde{u}' \\
  \tilde{y}'
\end{bmatrix} = \begin{bmatrix}
  H_L'(u) \\
  H_L'(y)
\end{bmatrix} \alpha_0
\]

Multiply previous solution by shifted Hankel matrix — Then repeat!
So far we have characterized a system in terms of data … how do we drive its behaviour?
Youla-Kučera parameterization
All stabilizing controllers

- Hard: Given a controller $K$, is it stabilizing? What is the set of all stabilizing controllers?

- Easier: What happens when you probe $P$ with stable dynamics $Q$?
How do we turn it into a controller?
Learning stable systems
(Q in Youla-Kučera)

- \( Q \) is a global parameter, but explicitly writing it down is difficult
- We represent \( Q \) using an unconstrained set of trainable parameters
- Yields stable models suitable for RL or supervised learning
Final ingredient: learning algorithms
Reinforcement learning

Business as usual

- A “policy” $\pi$ interacts with an “environment”, generating a trajectory $s_0, a_0, r_0, s_1, a_1, r_1, \ldots$

- A “return” is accrued and averaged:
  \[ V(s) = \mathbb{E} \left[ \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right], \text{where } s = s_0 \]

- An “agent” tries to find the “best” policy
Reinforcement learning over all stable behaviour
A modular setup

1. Willems’ lemma
2. Youla-Kučera
3. Learning algorithm

Decouples learning and stability
Industrial example

- RL agent manipulates $Q$ parameter
- End-to-end stable learning with DNN based control
  - Stable during and after training without loss in performance
Conclusions

• Constant advances in deep RL push the boundaries of what is possible

• This success is often misaligned with industrial priorities
  ▶ Performance is not the only metric

• We aim to preserve flexibility of general learning algorithms & maintain key system features


See also: Lawrence, Nathan P. “Deep reinforcement learning agents for industrial control system design.” Electronic Theses and Dissertations, University of British Columbia. 2023.

doi:http://dx.doi.org/10.14288/1.0430547.
Questions?

Lawrence@math.ubc.ca