Reinforcement Learning for Safety-Critical Applications

Enrique Mallada

ISC Seminar, JHU Applied Physics Laboratory
Jan 11, 2024
Networks, Dynamics, and Learning Laboratory

Electrical and Computer Engineering

Johns Hopkins University
**NetD-Lab Members**

**PhD Students**

- Agustin Castellano PhD, 2nd year
- Roy Siegelman PhD, AMS, 3rd year
- Yue Shen PhD, 4th year
- Rajni Bansal PhD, 2023, Postdoc UCSD
- Tianqi Zheng PhD, 2023, Postdoc UPenn
- Hancheng Min PhD, 2018
- Jay Guthrie PhD 2022, Apple
- Mohammad Hajiesmaili Postdoc 2017-2018, Asst. Prof. UMASS CS
- Mengnan Zhao, MSE 2018, PhD, Rice University
- Zachary Nelson, MSE 2018, Lockheed Martin
- Elijah Pivo, BS ’18, PhD, MIT IDSS
- Jesse Rines, BS ’18, US Navy, Nuclear Energy
- Aurik Sarker, BS’18, JHU APL

**Postdocs and Research Scientists**

- Dhajanajay (DJ) Annand RSc 2022-Present, Asst. Prof. Pekin University
- Pengcheng You PhD, 2019-2021, Postdoc 2019-2021
- M. Dewrim Kaba PhD, eBay, 2018-2020
- Mohammad Hajiesmaili PhD, 2017-2018, Asst. Prof. UMASS CS

**MSE and Undergraduates**

- Yue Shen PhD, 4th year
- Mohammad Hajiesmaili PhD, 2023, Postdoc UPenn
- Agustin Castellano PhD, 2nd year
- Roy Siegelman PhD, AMS, 3rd year
- Yue Shen PhD, 4th year
- Rajni Bansal PhD, 2023, Postdoc UCSD
- Tianqi Zheng PhD, 2023, Postdoc UPenn
- Hancheng Min PhD, 2018
- Jay Guthrie PhD 2022, Apple
- Mohammad Hajiesmaili Postdoc 2017-2018, Asst. Prof. UMASS CS
- Mengnan Zhao, MSE 2018, PhD, Rice University
- Zachary Nelson, MSE 2018, Lockheed Martin
- Elijah Pivo, BS ’18, PhD, MIT IDSS
- Jesse Rines, BS ’18, US Navy, Nuclear Energy
- Aurik Sarker, BS’18, JHU APL

* co-advised
NetD-Lab: Research Areas

- Control theory
- Optimization
- Dynamical systems
- Graph theory
- Economics
- Machine learning
- Statistics
NetD-Lab: Research Methodology

Core Research – Networked Dynamical Systems
- control theory
- optimization
- dynamical systems
- economics
- statistics

Constraints + Limitations

Insights + Solutions

Applications – Engineering Systems
- information networks
- power networks
- safety-critical systems
Recent Projects

**Learning, Dynamics, and Control**

- Learning safety via recurrence
- Reinforcement Learning with almost sure constraints
- Online C-RL via regularized saddle flow dynamics
- The role of overparameterization in learning dynamics of deep networks

**Power System Operations and Control**

- Frequency Shaping Control
- Spectral Clustering and model reduction in network dynamical systems
- A Market Mechanism for energy storage
- Market power in two-stage Market
- Market Dynamics: Stability, Efficiency, and Incentive Alignment
Recent Projects

Learning, Dynamics, and Control

• Learning safety via recurrence
• Reinforcement Learning with almost sure constraints
• Online C-RL via regularized saddle flow dynamics
• The role of overparameterization in learning dynamics of deep networks

Power System Operations and Control

• Frequency Shaping Control
• Spectral Clustering and model reduction in network dynamical systems
• A Market Mechanism for energy storage
• Market power in two-stage Market
• Market Dynamics: Stability, Efficiency, and Incentive Alignment
A World of Success Stories

2017 Google DeepMind’s DQN

2017 AlphaZero – Chess, Shogi, Go

2019 AlphaStar – Starcraft II

OpenAI – Rubik’s Cube

Boston Dynamics

Waymo
Can we adapt reinforcement learning algorithms to address physical systems challenges?
Challenges of RL for Physical Systems

- Physical systems must meet **multiple objectives**
  - Need to **trade off between** the different goals
  - **Constrained RL** allows to explore the Pareto Front \([1,2]\)

\[
\max_{\pi} (1 - \gamma) \mathbb{E}_{\pi, S_0 \sim q} \left[ \sum_{t=0}^{+\infty} \gamma^t R_t^{(0)} \right] \\
\text{s.t. } (1 - \gamma) \mathbb{E}_{\pi, S_0 \sim q} \left[ \sum_{t=0}^{+\infty} \gamma^t R_t^{(i)} \right] \geq h_i, \forall i \in [n]
\]

- **Failures** have a qualitatively different impact
  - Expectation constraints cannot meet safety requirements
  - **Hard (almost sure)** constraints can guarantee safety \([3,4]\)

\[
\max_{\pi} \mathbb{E}_{\pi, S_0 \sim q} \left[ \sum_{t=0}^{+\infty} \gamma^t R_{t+1} \right] \\
\text{s.t. } \mathbb{P}_{\pi, S_0 \sim q} \left[ S_t \notin G \right] = 1, \forall t \geq 0
\]

[3] Castellano, Min, Bazerque, M, Reinforcement Learning with Almost Sure Constraints, L4DC 2022
Saddle Flow Dynamics: Observable Certificates and Separable Regularization
Pengcheng You, Enrique Mallada

Constrained Reinforcement Learning via Dissipative Saddle Flow Dynamics
Tianqi Zheng, Pengcheng You, Enrique Mallada
Outline

• Intro to Constrained RL

• Dissipative Saddle Flows for Bilinear Saddles

• Solving Constrained RL via D-SGDA
Constrained Reinforcement Learning

**Goal:** Given initial state $S_0 \sim q$, find policy $\pi^* \in \Pi_\theta$ that solves:

$$\max_{\pi \in \Pi_\theta} V_q^{(0)}(\pi) \quad \text{s.t.} \quad V_q^{(i)}(\pi) \geq h_i, \quad \forall i \in [n]$$

where $V_q^{(i)}(\pi) := (1 - \gamma)\mathbb{E}_{\pi, S_0 \sim q} \left[ \sum_{t=0}^{\infty} \gamma^t R_t^{(i)} \right]$.

**General Approach:** Lagrange relaxation

$$\max_{\pi \in \Pi_\theta} \min_{\mu \geq 0} L(\pi, \mu) := V_q^{(0)}(\pi) + \sum_{i=1}^{n} \mu_i (V_q^{(i)}(\pi) - h_i)$$

Non-convex yet has zero duality gap! [1],[2]


Constrained Reinforcement Learning

**Goal:** Given initial state $S_0 \sim q$, find policy $\pi^* \in \Pi_\theta$ that solves:

$$\max_{\pi \in \Pi_\theta} V_q^{(0)}(\pi) \quad \text{s.t.} \quad V_q^{(i)}(\pi) \geq h_i, \quad \forall i \in [n]$$

where $V_q^{(i)}(\pi) := (1 - \gamma) \mathbb{E}_{S_0 \sim q} \left[ \sum_{t=0}^{\infty} \gamma^t R_t^{(i)} \right]$.

**General Approach:** Lagrange relaxation

$$\min_{\mu \geq 0} \max_{\pi \in \Pi_\theta} L(\pi, \mu) := V_q^{(0)}(\pi) + \sum_{i=1}^{n} \mu_i (V_q^{(i)}(\pi) - h_i)$$

Non-convex yet has zero duality gap! [1],[2]

Prior Work: Algorithms for Constrained RL [1]-[8]

Use primal and/or dual methods of the form:
\[
\begin{align*}
\pi_{k+1} &= \begin{cases} 
\pi_k + \eta \nabla_{\pi} \tilde{L}(\pi_k, \mu_k; \zeta_k) \\
\arg \max_{\pi} \tilde{L}(\pi, \mu_k; \zeta_k)
\end{cases} \\
\mu_{k+1} &= \begin{cases} 
\mu_k - \eta \nabla_{\mu} \tilde{L}(\pi_k, \mu_k; \zeta_k) \\
\arg \min_{\mu \geq 0} \tilde{L}(\pi_k, \mu; \zeta_k)
\end{cases}
\end{align*}
\]

where \( \tilde{L}(\pi, \mu; \zeta) := L(\pi, \mu; \zeta) + \Omega(\pi, \mu; \zeta) \) is a regularized Lagrangian

- **Parametrization of \( \Pi_\theta \):** Soft-max [1,4], occupancy measures [2,3], greedy.
- **Horizon:** Infinite \( \gamma \)-discounting [1-4], finite \( H \) [5-7], or average [8]

- **Regret:**

\[
\mathbb{E} \left[ \sum_{k=0}^{T-1} V_q^{(0)}(\pi^*) - V_q^{(0)}(\pi_k) \right] = O(T^{1/2}) \quad \mathbb{E} \left[ \sum_{k=1}^{T-1} c_i - V_q^{(i)}(\pi_k) \right] = O(T^p), \quad p \in [0, 3/4]
\]

- **Policy:** Iterates \( \pi_k \) lack convergence guarantees: Instead \( \hat{\pi}_T = \sum_{t=0}^{T-1} \alpha_k \pi_k \rightarrow \pi^* \) [2,3]
Prior Work: Algorithms for Constrained RL [1]-[8]

Use primal and/or dual methods of the form:

\[
\pi_{k+1} = \begin{cases} 
\pi_k + \eta \nabla_\pi \tilde{L}(\pi_k, \mu_k; \zeta_k) \\
\arg \max_\pi \tilde{L}(\pi, \mu_k; \zeta_k)
\end{cases} \quad \mu_{k+1} = \begin{cases} 
\mu_k - \eta \nabla_\mu \tilde{L}(\pi_k, \mu_k; \zeta_k) \\
\arg \min_{\mu \geq 0} \tilde{L}(\pi_k, \mu; \zeta_k)
\end{cases}
\]

where \( \tilde{L}(\pi, \mu; \zeta) := L(\pi, \mu; \zeta) + \Omega(\pi, \mu; \zeta) \) is a regularized Lagrangian

- Parametrization of \( \Pi_\theta \): Soft-max [1,4], occupancy measures [2,3], greedy.
- Horizon: Infinite \( \gamma \)-discounting [1-4], finite \( H \) [5-7], or average [8]

- Regret:

\[
\mathbb{E} \left[ \sum_{k=0}^{T-1} V_q^{(0)}(\pi^*) - V_q^{(0)}(\pi_k) \right] = \mathcal{O}(T^{1/2}) \quad \mathbb{E} \left[ \sum_{k=1}^{T-1} c_i - V_q^{(i)}(\pi_k) \right] = \mathcal{O}(T^p), \ p \in [0, 3/4)
\]

- Policy: Iterates \( \pi_k \) lack convergence guarantees: Instead \( \hat{\pi}_T = \sum_{t=0}^{T-1} \alpha_t \pi_k \rightarrow \pi^* \) [2,3]

Question: Can we achieve convergence of the policy iterates \( \pi_k \rightarrow \pi^* \text{ a.s.} \), or is learning from rewards a fundamental limitation?
Towards convergent $\pi_k$ iterates – Good news

**Good news:** Non-convexity of $L(\pi, \mu)$ is not so bad...

- There exists a convex parametrization $\Pi_\theta$ that makes it convex-concave

$$\begin{align*}
\max_{\pi} \ (1 - \gamma) \mathbb{E}_{\pi, S_0 \sim q} \left[ \sum_{t=0}^{+\infty} \gamma^t R^{(0)}_{t+1} \right] \\
\text{s.t.} \ (1 - \gamma) \mathbb{E}_{\pi, S_0 \sim q} \left[ \sum_{t=0}^{+\infty} \gamma^t R^{(i)}_{t+1} \right] \geq h_i, \ \forall i \in [n]
\end{align*}$$

- **LP Formulation:**\(^{[1]}\)

$$\begin{align*}
\max_{\lambda \geq 0} \sum_a \lambda_a^T r_a^{(0)} \\
\text{s.t.} \ \sum_a \lambda_a^T r_a^{(i)} \geq h_i, \ \forall i \in [n] \quad (\mu_i) \\
\sum_a (I - \gamma P_a^T) \lambda_a = (1 - \gamma) q \quad (v)
\end{align*}$$

- where $\lambda_{s,a} = (1 - \gamma) \sum_{t=0}^{+\infty} \gamma^t \mathbb{P}_{\pi, S_0 \sim q}(S_t = s, A_t = a)$ is the occupancy measure

Towards convergent $\pi_k$ iterates – Bad news

Bad news: Non-stricness of $L(\lambda, \mu, \nu)$

• **LP Formulation:**

  - Outline

    $$\max_{\lambda \geq 0} \sum_a \lambda_a r_a^{(0)}$$

    s.t. $\sum_a \lambda_a r_a^{(i)} \geq h_i, \forall i \in [n]$ (dual vars)

    $\sum_a (I - \gamma P_a^T) \lambda_a = (1 - \gamma)q$ (v)

• where $\lambda_{s,a} = (1 - \gamma) \sum_{t=0}^{+\infty} \nu^t \mathbb{P}_{\pi} S_0 \sim q(S_t = s, A_t = a)$ is the occupancy measure

• **Bilinear Lagrangian:**

  - Lacks strict convexity/concavity necessary for convergence of primal-dual algorithms

    $$\min_{\mu \geq 0, v} \max_{\lambda \geq 0} L(\lambda, \mu, v) = \lambda^T M \begin{bmatrix} \mu \\ v \end{bmatrix}$$
Outline

• Intro to Constrained RL

• Dissipative GDA Flows for Convex-concave $L$

• Solving Constrained RL via D-SGDA
We start by looking at a Naïve GDA Flow on a scalar bilinear Lagrangian

- Min-max Problem:
  \[
  \min_x \max_y L(x, y) := xy \quad x, y \in \mathbb{R}
  \]
- Saddle-point at \((x^*, y^*) = (0,0)\)
- Naïve Gradient Descent-Ascent (GDA) Flow
  \[
  \begin{bmatrix}
  \dot{x} \\
  \dot{y}
  \end{bmatrix} =
  \begin{bmatrix}
  \nabla_x L(x, y) \\
  +\nabla_y L(x, y)
  \end{bmatrix} =
  \begin{bmatrix}
  0 & -1 \\
  1 & 0
  \end{bmatrix}
  \begin{bmatrix}
  x \\
  y
  \end{bmatrix}
  \]
- Energy Dissipation:
  \[
  V(x, y) = \frac{1}{2} x^2 + \frac{1}{2} y^2, \quad \dot{V}(x, y) = x(-y) + yx \equiv 0
  \]

**Remark:** Behavior generalizes for general non-strict convex-concave Lagrangians \[1\].

## Naïve GDA Flow Scalar Case

<table>
<thead>
<tr>
<th></th>
<th>Naïve GDA Flow</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lagrangian</td>
<td>$L(x, y) = xy$</td>
</tr>
<tr>
<td>Dynamics</td>
<td></td>
</tr>
<tr>
<td>Energy Function</td>
<td></td>
</tr>
<tr>
<td>Energy Dissipation</td>
<td></td>
</tr>
<tr>
<td>Asympt. Behavior</td>
<td></td>
</tr>
</tbody>
</table>
Dissipative GDA Flow Algorithm

- Given general convex-concave $L(x, y)$, we consider

$$
\hat{L}(x, \hat{x}, y, \hat{y}) = L(x, y) + \frac{\rho}{2}\|x - \hat{x}\|^2 - \frac{\rho}{2}\|y - \hat{y}\|^2
$$

- **Remarks:**
  - If $(x^*, y^*)$ is a saddle point of $L$, then $(x^*, x^*, y^*, y^*)$ is a saddle point of $\hat{L}$.
  - $\hat{L}$ is neither strictly convex, nor strictly concave (don’t worry)

- **Dissipative GDA Flow:**
  - Just apply Naïve GDA on $\hat{L}(x, \hat{x}, y, \hat{y})$!

\[
\begin{align*}
\dot{x} &= -\nabla_x L(x, y) - \rho(x - \hat{x}) \\
\dot{\hat{x}} &= -\rho(\hat{x} - x) \\
\dot{y} &= +\nabla_y L(x, y) - \rho(y - \hat{y}) \\
\dot{\hat{y}} &= -\rho(\hat{y} - y)
\end{align*}
\]
Dissipative GDA Flow Algorithm

• Dissipative GDA Flow:
  
  - Just apply Naïve GDA on $\hat{L}(x, \hat{x}, y, \hat{y}) = L(x, y) + \frac{\rho}{2}\|x - \hat{x}\|^2 - \frac{\rho}{2}\|y - \hat{y}\|^2$ !

  $\dot{x} = -\nabla_x L(x, y) - \rho(x - \hat{x})$

  $\dot{\hat{x}} = -\rho(\hat{x} - x)$

  $\dot{y} = +\nabla_y L(x, y) - \rho(y - \hat{y})$

  $\dot{\hat{y}} = -\rho(\hat{y} - y)$

• Scalar case:
  
  - $\hat{L}(x, \hat{x}, y, \hat{y}) = xy + \frac{\rho}{2}(x - \hat{x})^2 + \frac{\rho}{2}(y - \hat{y})^2$

  $\begin{bmatrix}
  \dot{x} \\
  \dot{\hat{x}} \\
  \dot{y} \\
  \dot{\hat{y}} 
  \end{bmatrix} =
  \begin{bmatrix}
  -\rho & \rho & -1 & 0 \\
  \rho & -\rho & 0 & 0 \\
  1 & 0 & -\rho & \rho \\
  0 & 0 & \rho & -\rho 
  \end{bmatrix}
  \begin{bmatrix}
  x \\
  \hat{x} \\
  y \\
  \hat{y} 
  \end{bmatrix}$

![Solution of Bilinear Lagrangian](image)
# Dissipative GDA Flow Scalar Case

<table>
<thead>
<tr>
<th></th>
<th>Naïve GDA Flow</th>
<th>Dissipative GDA Flow</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Lagrangian</strong></td>
<td>$L(x, y) = xy$</td>
<td></td>
</tr>
<tr>
<td><strong>Dynamics</strong></td>
<td>$\begin{bmatrix} \dot{x} \ \dot{y} \end{bmatrix} = \begin{bmatrix} -\nabla_x L(x, y) \ +\nabla_y L(x, y) \end{bmatrix}$</td>
<td></td>
</tr>
<tr>
<td><strong>Energy Function</strong></td>
<td>$V(x, y) = \frac{1}{2}(x^2 + y^2)$</td>
<td></td>
</tr>
<tr>
<td><strong>Energy Dissipation</strong></td>
<td>$\dot{V} \equiv 0$</td>
<td></td>
</tr>
<tr>
<td><strong>Asympt. Behavior</strong></td>
<td>$V(t) \equiv c$</td>
<td></td>
</tr>
</tbody>
</table>
General Analysis of Dissipative GDA Flows

Theorem [You, M ACC 21]
Consider the minimax problem

\[
\min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} L(x, y)
\]

where \( L(x, y) \) is convex-concave, and the sets \( \mathcal{X} \) and \( \mathcal{Y} \) are convex polyhedral. Then, for any initial feasible point \((x_0, \hat{x}_0, y_0, \hat{y}_0)\) the Dissipative GDA Flow

\[
\dot{x} = \Pi_{\mathcal{X},x} \left[ -\nabla_x L(x, y) - \rho (x - \hat{x}) \right] \\
\dot{\hat{x}} = -\rho (\hat{x} - x) \\
\dot{y} = \Pi_{\mathcal{Y},y} \left[ +\nabla_y L(x, y) - \rho (y - \hat{y}) \right] \\
\dot{\hat{y}} = -\rho (\hat{y} - y)
\]

converges to some saddle point.

• Remarks:
  • Convergence is guaranteed point-wise, to some saddle point
  • Proof uses LaSalle on the same dissipation property \( \dot{V} \leq -\rho \|\hat{x}\|^2 - \rho \|\hat{y}\|^2 \)
  • For unconstrained bilinear problems \textit{convergence is exponential}

Outline

• Intro to Constrained RL

• Dissipative GDA Flows for Convex-concave $L$

• Solving Constrained RL via D-SGDA
Dissipative GDA for Constrained MDPs

**LP Formulation of C-RL**

\[
\begin{align*}
\max_{\lambda \geq 0} & \quad \sum_a \lambda_a^T r_a^{(0)} \\
\text{s.t.} & \quad \sum_a \lambda_a^T r_a^{(i)} \geq h_i, \quad \forall i \in [n] \\
& \quad \sum_a (I - \gamma P_a^T) \lambda_a = (1 - \gamma) q
\end{align*}
\]

\[\min_{\mu \geq 0, v} \max_{\lambda \geq 0} L(\lambda, \mu, v) = \lambda^T M \begin{bmatrix} \mu \\ v \end{bmatrix}\]

**D-GDA Flow**

\[
\begin{align*}
\dot{v} &= \sum_a (I - \gamma P_a^T) \lambda_a - (1 - \gamma) q - \rho (v - \hat{v}) \\
\dot{\mu}_i &= \Pi_{\mathbb{R}^+} \left[ \mu; h_i - \sum_a \lambda_a^T r_a^{(i)} - \rho (\mu_i - \hat{\mu}_i) \right] \\
\dot{\lambda}_a &= \Pi_{\Delta} \left[ \lambda_a; r_a^{(0)} - (I - \gamma P_a) v + \sum_{i \in [n]} \mu_i r_a^{(i)} - \rho (\lambda_a - \hat{\lambda}_a) \right]
\end{align*}
\]

\[
\dot{\hat{v}} = -\rho (\hat{v} - v) \\
\dot{\hat{\mu}}_i = -\rho (\hat{\mu}_i - \mu_i) \\
\dot{\hat{\lambda}}_a = -\rho (\hat{\lambda}_a - \lambda_a)
\]

unknowns
Dissipative Stochastic GDA for Constrained RL

• Oracle: At each time $t$ sample $S_0 \sim q$, $(S_t, A_t) \sim \xi$, $S_{t+1} \sim \mathbb{P}(\cdot | S_t, A_t)$:

• DS-GDA Update:

$$
v^{t+1} = v^t + \alpha^t \left[ \mathbb{1}_{\{\xi(S_t, A_t) > 0\}} \frac{\lambda_{S_t, A_t}^t}{\xi(S_t, A_t)} \left( e_s - \gamma e_{s+1} - (1-\gamma) e_s - \rho (v^t - \hat{v}^t) \right) \right], \quad \hat{v}^{t+1} = v^t - \alpha^t \rho (\hat{v}^t - v^t)
$$

$$
\mu_{i}^{t+1} = \mu_{i}^t + \alpha^t \left( h_i - \mathbb{1}_{\{\xi(S_t, A_t) > 0\}} \frac{\lambda_{S_t, A_t}^{t+1}}{\xi(S_t, A_t)} - \rho (\mu_i^t - \hat{\mu}_i^t) \right), \quad \hat{\mu}_{i}^{t+1} = \mu_{i}^t - \alpha^t \rho (\hat{\mu}_i^t - \mu_i^t)
$$

$$
\lambda_{a}^{t+1} = \lambda_{a}^t + \alpha^t \left( \mathbb{1}_{\{\xi(S_t, A_t) > 0 \& A_t=a\}} \sum_{i=1}^{n} \mu_i^{t} R_{t+1}^{(i)} + \gamma v_{S_{t+1}}^{t} - v_{S_t}^{t} \right) \frac{\lambda_{S_t, A_t}^{t}}{\xi(S_t, A_t)} e_s - \rho (\lambda_{a}^t - \hat{\lambda}_{a}^t) \right] +, \quad \hat{\lambda}_{a}^{t+1} = \lambda_{a}^t - \alpha^t \rho (\hat{\lambda}_a^t - \lambda_a^t)
$$

**Theorem [Zheng, You, M ‘22]**

Under mild assumptions, as $t \to \infty$ the sequence $(\lambda^t, \mu^t, v^t)$ generated by S-GDA converges to the optimal solution to the C-RL LP Problem.

In particular, the iterates $\pi_t(a|s) = \frac{\lambda_{S_t,a}^t}{\sum_{a'} \lambda_{S_t,a'}^t} \to \pi^*$ a.s.
Challenges of RL for Physical Systems

- Physical systems must meet **multiple objectives**
  - Need to **trade off between** the different goals
  - Constrained RL allows to explore the Pareto Front $[1,2]$

$$\max_\pi (1 - \gamma) \mathbb{E}_{\pi,S_0 \sim q} \left[ \sum_{t=0}^{+\infty} \gamma^t R^{(0)}_{t+1} \right]$$

s.t. $$\max_\pi \mathbb{E}_{\pi,S_0 \sim q} \left[ \sum_{t=0}^{+\infty} \gamma^t R^{(i)}_{t+1} \right] \geq h_i, \ \forall i \in [n]$$

- **Failures** have a qualitatively different impact
  - Expectation constraints cannot meet safety requirements
  - **Hard (almost sure) constraints** can guarantee safety $[3,4]$

$$\max_\pi \mathbb{E}_{\pi,S_0 \sim q} \left[ \sum_{t=0}^{+\infty} \gamma^t R_{t+1} \right]$$

s.t. $$\mathbb{P}_{\pi,S_0 \sim q} \left[ S_t \notin \mathcal{G} \right] = 1, \ \forall t \geq 0$$

---

[3] Castellano, Min, Bazerque, M, Reinforcement Learning with Almost Sure Constraints, L4DC 2022
Reinforcement Learning for Safety-Critical Systems

\[
\max_{\pi} \mathbb{E}_{\pi,S_0 \sim q} \left[ \sum_{t=0}^{+\infty} \gamma^t R_{t+1} \right]
\]

s.t. \( \mathbb{P}_{\pi,S_0 \sim q} \left[ S_t \notin \mathcal{G} \right] = 1, \forall t \geq 0 \)

Challenges of SC-RL:

- Avoiding unsafe regions requires anticipation
  - A car at 100 mph at 10 feet from a wall still hasn’t hit the wall!
Reinforcement Learning for Safety-Critical Systems

\[
\max_{\pi} \mathbb{E}_{\pi,S_0 \sim q} \left[ \sum_{t=0}^{+\infty} \gamma^t R_{t+1} \right]
\]

s.t. \[
\mathbb{P}_{\pi,S_0 \sim q} [ \text{safe trajectory} ] = 1
\]

Challenges of SC-RL:
- Avoiding unsafe regions requires anticipation.
- A car at 100 mph at 10 feet from a wall still hasn’t hit the wall!
Reinforcement Learning for Safety-Critical Systems

\[
\max_{\pi} \mathbb{E}_{\pi,S_0 \sim q} \left[ \sum_{t=0}^{+\infty} \gamma^t R_{t+1} \right] \\
\text{s.t. } \mathbb{P}_{\pi,S_0 \sim q} \left[ S_t \notin \mathcal{G} \right] = 1, \ \forall t \geq 0
\]

Challenges of SC-RL:

- Avoiding unsafe regions requires anticipation
  - A car at 100 mph at 10 feet from a wall still hasn’t hit the wall!
  - Model-based \(\Rightarrow\) Reachability Theory
- Model-free:
  - Constraints not given a priori: Need to learn from experience!
  - Constraint violations are inevitable \(\Rightarrow\) Maybe not all constraints can be learned online
Example: Transient Stability in Power Systems

\[ \begin{align*}
\dot{\delta} &= \omega \\
\dot{\omega} &= \frac{1}{M} \left( u - D\omega - P_e \sin \delta \right) \\
u &\in [u_{\text{min}}, u_{\text{max}}]
\end{align*} \]

- **Q:** Which states can reach a neighborhood of the stable equilibrium?
Example: Air Collision Avoidance

\[
\begin{align*}
\dot{x}_1 &= -v + v \cos x_3 + ax_2 \\
\dot{x}_2 &= v \sin x_3 - ax_2 \\
\dot{x}_3 &= b - a \\
b, a &\in [-1, 1]
\end{align*}
\]

• **Q:** From which states can the evader avoid collision?
Related Work

Reachability Theory[1-2]
- **Model-based**: Via Hamilton Jacobi Issacs Equations (cont. time), or iterative set updates (discrete time).
- **Constraints**: Provides hard/almost sure guarantees
- **Output**: Finds the *maximum control invariant set (M-CIS) outside* $\mathcal{G}$

Control Barrier Functions (CBF)[3-4]
- **Model-based**: Requires knowledge of dynamics and *finding such CBF!*
- **Constraints**: Provides hard/almost sure guarantees
- **Output**: Possibly conservative CIS

Safety Critics (SC)[5-7]
- **Model-free**: Q-Learning-like algorithms, computes function such that $Q_{\text{safe}}(s,a) \geq \eta_{\text{thresh}} \Rightarrow "\text{safety}"$
- **Constraints**: Provides soft/approximate guarantees, depending on discounting factor $\gamma \in (0,1)$
- **Output**: Converges to maximum CIS as $\gamma \to 1$

Related Work

Reachability Theory\textsuperscript{[1-2]}
- \textbf{Model-based:} Via Hamilton Jacobi Issacs Equations (cont. time), or iterative set updates (discrete time).
- \textbf{Constraints:} Provides hard/almost sure guarantees
- \textbf{Output:} Finds the maximum control invariant set (M-CIS) outside $\mathcal{G}$

Control Barrier Functions (CBF)\textsuperscript{[3-4]}
- \textbf{Model-based:} Requires knowledge of dynamics and finding such CBF!
- \textbf{Constraints:} Provides hard/almost sure guarantees
- \textbf{Output:} Possibly conservative CIS

Safety Critics (SC)\textsuperscript{[5-7]}
- \textbf{Model-free:} Q-Learning-like algorithms, computes function such that $Q_{safe}(s,a) \geq \eta_{thresh} \Rightarrow \text{“safety”}$
- \textbf{Constraints:} Provides soft/approximate guarantees, depending on discounting factor $\gamma \in (0,1)$
- \textbf{Output:} Converges to maximum CIS as $\gamma \to 1$

<table>
<thead>
<tr>
<th>Method</th>
<th>Model-free</th>
<th>Constraint Type</th>
<th>Set Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reachability Theory\textsuperscript{[1-2]}</td>
<td>No</td>
<td>Hard</td>
<td>Maximal</td>
</tr>
<tr>
<td>Control Barrier Functions\textsuperscript{[3-4]}</td>
<td>No</td>
<td>Hard</td>
<td>Subset</td>
</tr>
<tr>
<td>Safety Critics\textsuperscript{[5-7]}</td>
<td>Yes</td>
<td>Soft/Approx.</td>
<td>Maximal</td>
</tr>
</tbody>
</table>
Our Work

Reachability Theory[1-2]
- **Model-based**: Via Hamilton Jacobi Issacs Equations (cont. time), or iterative set updates (discrete time).
- **Constraints**: Provides hard/almost sure guarantees
- **Output**: Finds the maximum control invariant set (M-CIS) outside \( \mathcal{G} \)

Control Barrier Functions (CBF)[3-4]
- **Model-based**: Requires knowledge of dynamics and finding such CBF!
- **Constraints**: Provides hard/almost sure guarantees
- **Output**: Possibly conservative CIS

Safety Critics (SC)[5-7]
- **Model-free**: Q-Learning-like algorithms, computes function such that \( Q_{safe}(s,a) \geq \eta_{thresh} \Rightarrow "safety" \)
- **Constraints**: Provides soft/approximate guarantees, depending on discounting factor \( \gamma \in (0,1) \)
- **Output**: Converges to maximum CIS as \( \gamma \to 1 \)

<table>
<thead>
<tr>
<th>Method</th>
<th>Model-free</th>
<th>Constraint Type</th>
<th>Set Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reachability Theory[1-2]</td>
<td>No</td>
<td>Hard</td>
<td>Maximal</td>
</tr>
<tr>
<td>Control Barrier Functions[3-4]</td>
<td>No</td>
<td>Hard</td>
<td>Subset</td>
</tr>
<tr>
<td>Safety Critics[5-7]</td>
<td>Yes</td>
<td>Soft/Approx.</td>
<td>Maximal</td>
</tr>
<tr>
<td>Ours</td>
<td>Yes</td>
<td>Hard</td>
<td>Maximal and Subsets</td>
</tr>
</tbody>
</table>
Reinforcement Learning for Safety-Critical Systems

\[
\max_{\pi} \mathbb{E}_{\pi,S_0 \sim q} \left[ \sum_{t=0}^{\infty} \gamma^t R_{t+1} \right] \\
\text{s.t. } \mathbb{P}_{\pi,S_0 \sim q} \left[ S_t \notin G \right] = 1, \ \forall t \geq 0
\]

Methodology:

- Enhance RL with **logical** feedback naturally arising from constraint violations
  \[ S_t \in G \iff D_t = 1 \]
- Decouple **feasibility** from optimality: **Separation Principle**
- Develop algorithms for learning fixed points of **non-contractive operators**
Outline

• Separation Principle for Joint Safety & Optimality

• Learning Safety with Limited Failures

• One-sided Bellman Equations for Continuous States
Recap: RL with Almost Sure Constraints

\[
\max_{\pi} \mathbb{E}_{\pi,S_0 \sim q} \left[ \sum_{t=0}^{+\infty} \gamma^t R_{t+1} \right] \\
\text{s.t. } \mathbb{P}_{\pi,S_0 \sim q} \left[ S_t \notin \mathcal{G} \right] = 1, \ \forall t \geq 0 \iff D_{t+1} = 0 \text{ almost surely } \forall t
\]

- Damage indicator \( D_t \in \{0,1\} \) turns on \((D_t = 1)\) when constraints are violated
Formulation via hard barrier indicator

Safe RL problem:

\[ V^*(s) := \max_{\pi} \mathbb{E}_\pi \left[ \sum_{t=0}^{\infty} \gamma^t R_{t+1} \mid S_0 = s \right] \]

s.t.: \( D_{t+1} = 0 \) almost surely \( \forall t \)

Equivalent unconstrained formulation:

\[ \sim \max_{\pi} \mathbb{E}_\pi \left[ \sum_{t=0}^{\infty} \gamma^t R_{t+1} + \log[1 - D_{t+1}] \mid S_0 = s \right] \]

0 if \( D_{t+1} = 0 \)

\( -\infty \) if \( D_{t+1} = 1 \)

Questions/Comments:

• Is this just a standard RL problem with \( \tilde{R}_{t+1} = R_{t+1} + \log(1 - D_{t+1}) \) ?
• Standard MDP assumptions for Value Iteration, Bellman’s Eq., Optimality Principle, etc., do not hold!
• Not to mention convergence of stochastic approximations.

Key idea: Separate the problem of safety from optimality
Hard Barrier Action-Value Functions

Consider the Q-function for a given policy $\pi$, 

$$Q^\pi(s, a) = \mathbb{E}_\pi \left[ \sum_{t=0}^{\infty} \left( \gamma^t R_{t+1} + \log(1 - D_{t+1}) \right) \mid S_0 = s, A_0 = a \right]$$

and define the hard-barrier function 

$$B^\pi(s, a) = \mathbb{E}_\pi \left[ \sum_{t=0}^{\infty} \log(1 - D_{t+1}) \mid S_0 = s, A_0 = a \right]$$

Notes on $B^\pi(s, a)$:

• $B^\pi(s, a) \in \{0, -\infty\}$

• Summarizes safety information
  • $B^\pi(s, a) = 0$ iff $\pi$ is safe after choosing $A_t = a$ when $S_t = s$

• It is independent of the reward process
Separation Principle

**Theorem** (Separation principle)
Assume rewards $R_{t+1}$ are bounded almost surely for all $t$. Then for every policy $\pi$:

$$Q^\pi(s, a) = Q^\pi(s, a) + B^\pi(s, a)$$

In particular, for optimal $\pi^*$:

$$Q^*(s, a) = Q^*(s, a) + B^*(s, a)$$

**Approach:** Learn feasibility (encoded in $B^*$) independently from optimality.
Optimal Hard Barrier Action-Value Function

**Theorem (Safety Bellman Equation for $B^*$)**

Let $B^*(s,a) := \max_{\pi} B^\pi(s,a)$, then the following holds:

$$B^*(s,a) = \mathbb{E} \left[ -\log(1 - D_{t+1}) + \max_{a'} B^*(S_{t+1}, a') \mid S_0 = s, A_0 = a \right]$$

Understanding $B^*(s,a)$:

- $B^*(s,a) \in \{0, -\infty\}$ summarizes safety information of the entire MDP
  - $B^*(s,a) = 0$ if $\exists$ safe $\pi$ after choosing $A_t = a$ when $S_t = s$ (Control Invariant)
  - $B^*(s,a) = -\infty$ if no safe policy exists after choosing $A_t = a$ when $S_t = s$ (Unsafe)

Discrete States

- $V^*(s) = \max_a B^*(s,a) = 0$

Continuous States

- $R(G)$
- $\mathcal{G}$
- Controlled safe trajectory

- $V^*(s) = \max_a B^*(s,a) = -\infty$
Properties of Safety Bellman Equation

Understanding the Solutions to the Safety Bellman Equation (SBE):

\[
\tilde{B}(s, a) = \mathbb{E} \left[ - \log(1 - D_{t+1}) + \max_a \tilde{B}(S_{t+1}, a) \mid S_0 = s, A_0 = a \right]
\]

• SBE can have multiple solutions, including \( \tilde{B}(s, a) = -\infty \), for all pairs \((s, a)\)

• If the function \( \tilde{B} \) is a solution to the SBE, then:
  • The set \( \mathcal{C} := \{ s : \max_a \tilde{B}(s, a) = 0 \} \) is a control invariant safe set
  • \( \mathcal{C} \) is maximal: If \( S_0 \notin \mathcal{C} \), then \( S_t \) never reaches \( \mathcal{C} \) for all policies \( \pi \)
Outline

• Separation Principle for Joint Safety & Optimality
• Learning Safety with Limited Failures
• One-sided Bellman Equations for Continuous States
Learning the barrier in finite MDPs...

Pros:
- Wraps around learning algorithms (Q-learning, SARSA)
- Use the B to trim the exploration set and avoid repeating unsafe actions

...with a generative model:
- Sample a transition \((s, a, s', d)\) according to the MDP. Update barrier function.

**Algorithm 3:** barrier_update

| B-function (initialized as all-zeros); |
| Input: \((s, a, s', d)\) |
| Output: Barrier-function \(B(s, a)\) |
| \(B(s, a) \leftarrow B(s, a) + \log(1 - d) + \max_{a'} B(s', a')\) |

**Algorithm 5:** Barrier Learner Algorithm

- **Data:** Constrained Markov Decision Process \(\mathcal{M}\)
- **Result:** Optimal action-value function \(B^*\)
- Initialize \(B^{(0)}(s, a) = 0, \forall (s, a) \in S \times A\)

\[ \text{for } t = 0, 1, \ldots \text{ do} \]

| Draw \((s_t, a_t) \sim \text{Unif}\{\{(s, a) : B^{(t)}(s, a) \neq -\infty\}\} \) |
| Sample transition \((s_t, a_t, s'_t, d_t)\) according to |
| \(P(S_{t+1} = s'_t, D_t = d_t | S_t = s_t, A_t = a_t)\) |
| \(B^{(t+1)} \leftarrow \text{barrier_update}(B^{(t)}, s_t, a_t, s'_t, d_t)\) |
| end |

Initially, all \((s, a)\)-pairs are “safe”

Draw \((s, a)\)-pair uniformly among those considered to be “safe” at time \(t\)

Update barrier function
Convergence in Expected Finite Time

Theorem (Safety Guarantee): Let \( T = \min_{t} \{B^{(t)} = B^*\} \), then

\[
\mathbb{E}T \leq (L + 1) \frac{|S||A|}{\mu} \left( \sum_{k=1}^{|S||A|} \frac{1}{k} \right)
\]

• After \( T = \min_{t} \{B^{(t)} = B^*\} \), all “unsafe” \((s, a)\)-pairs are detected

• \( \mu \): Lower bound on the non-zero transition probability
  \[
  \mu = \min\{p(s', d|s, a): p(s', d|s, a) \neq 0\}
  \]

• \( L \): Lag of the MDP

\[
L = \max_{(s, a)} \left\{ \begin{array}{l}
\text{Minimum number of transitions} \\
\text{needed to observe damage,} \\
\text{starting from unsafe } (s, a) \\
\end{array} \right. \\
B^*(s, a) = -\infty
\]
Lag of the MDP: $L$

$$L = \max_{(s,a)} \{ \text{Minimum number of transitions needed to observe damage, starting from unsafe } (s,a) \}$$

$$B^*(s,a) = -\infty$$

$L = 3$

Diagram showing a Markov Decision Process with $D_t = 1$, $B^*(s,a) = -\infty$, and $B^*(s,a) = 0$. The maximum value of $L$ is 3.
Sample Complexity of Safety

Theorem (Sample Complexity): With at least $1 - \delta$ probability, the algorithm learns optimal barrier function $B^*$ after

$$(L + 1) \frac{|S||A|}{\mu} \left( \sum_{k=1}^{\frac{|S||A|}{\varepsilon}} \frac{1}{k} \right) \log \frac{1}{\delta}$$

iterations

- Concentration of sum of exponential random variables
- **Much more sample-efficient** than “learning an $\varepsilon$-optimal policy with $1 - \delta$ probability” (Li et al. 2020)

$$N = \frac{|S||A|}{(1 - \gamma)^4 \varepsilon^2} \log^2 \left( \frac{|S||A|}{(1 - \gamma)\varepsilon \delta} \right)$$
Theorem (Sample Complexity): With at least $1 - \delta$ probability, the algorithm learns optimal barrier function $B^*$ after

$$(L + 1) \frac{|S||A|}{\mu} \left( \sum_{k=1}^{\frac{|S||A|}{1}} \frac{1}{k} \right) \log \frac{1}{\delta}$$

iterations

• Concentration of sum of exponential random variables

• If the Barrier Function is learnt first, then learning an $\epsilon$-optimal policy takes

$$N' = \frac{|S_{safe}||A_{safe}|}{(1 - \gamma)^4 \varepsilon^2} \log^2 \left( \frac{|S_{safe}||A_{safe}|}{(1 - \gamma)\varepsilon\delta} \right)$$

samples (Trimming the MDP by learning the barrier)
Numerical Experiments

Goal: Reach the end of the aisle \((R_{t+1} = 10)\)

Touching the wall gives \(D_{t+1} = 1\), resets the episode.

Results

Why does Assured Q-learning perform much better?

If \(D_{t+1} = 1 \implies B_{\pi}(s, a) = -\infty \implies \text{Never take action } a \text{ at } s \text{ again!}

Takeaways:
- Adding constraints to the problem can accelerate learning
- Barrier function avoids actions that lead to further wall bumps
Outline

• Separation Principle for Joint Safety & Optimality

• Learning Safety with Limited Failures

• One-sided Bellman Equations for Continuous States
Recall: Properties of Safety Bellman Equation

Understanding the Solutions to the Safety Bellman Equation (SBE):

\[
\tilde{B}(s, a) = \mathbb{E} \left[ -\log(1 - D_{t+1}) + \max_a \tilde{B}(S_{t+1}, a) \mid S_0 = s, A_0 = a \right]
\]

Understanding the Solutions to the Safety Bellman Equation (SBE):
- SBE can have multiple solutions, including \( \tilde{B}(s, a) = -\infty \), for all pairs \((s, a)\)
- If the function \( \tilde{B} \) is a solution to the SBE, then:
  - The set \( \mathcal{C} := \left\{ s : \max_a \tilde{B}(s, a) = 0 \right\} \) is a control invariant safe set
  - \( \mathcal{C} \) is maximal: If \( S_0 \notin \mathcal{C} \), then \( S_t \) never reaches \( \mathcal{C} \) for all policies \( \pi \)

**Problem:** Maximal solutions can be very close to unsafe region \( \mathcal{R}(G) \)
One-Sided Safety Bellman Equation

**Theorem** (One-Sided Safety Bellman Equation)

Let $\tilde{B}(s, a)$ be a solution of the following set of inequalities:

$$\tilde{B}(s, a) \leq \mathbb{E} \left[ -\log(1 - D_{t+1}) + \max_{a'} \tilde{B}(S_{t+1}, a') | S_0 = s, A_0 = a \right]$$

The set $\mathcal{C} := \{ s : \max_{a} \tilde{B}(s, a) = 0 \}$ is a control invariant safe set, not necessarily maximal.

---

Solution

Not a Solution

New Solutions
Learning CIS Using Deep Neural Nets

Algorithm Summary

• Uses axiomatic data \((s, a, d, s') \in D_{safe}\) known to be safe

• \(\text{Initialize } \hat{b}^\theta(s, a) = 0, \text{ where } \hat{b}(s, a) = 1 - e^{B(s,a)}\) (all presumed safe)

• At each iteration, take N episodes starting from \(D_{safe}\)
  
  • Behavioral policy: uniform safe policy
    
    \[
    \pi^\theta(a|s) = \begin{cases} 
    0 & \text{if } \hat{b}^\theta(s, a) = 1 \\
    1/\sum_{a' \in A} \mathbb{1}\{\hat{b}^\theta(s, a') = 0\} & \text{if } \hat{b}^\theta(s, a) = 0
    \end{cases}
    \]

• Train NN using SGD until fully fitting the data

• Start a new iteration, and repeat
Numerical Illustration

Control Engineer Favorite’s: Inverted Pendulum

SBE = Fisac’s ‘19 Safety Critic
Summary and future work

• Methodologies to Adapt Reinforcement Learning to Safety-Critical Systems

• C-RL via Dissipative Saddle Flows
  • Investigate methods to learn saddle-points in deterministic and stochastic settings
  • Proposed a general methodology to ensure convergence to saddle points of general convex-concave functions
  • Application to Constrained RL problems

  • Takeaways:
    • Dissipative GDA guarantees convergence on a wide family of minimax problems
    • When combined with stochastic approximations (D-SGDA) renders convergent policy iterates $\pi_k \rightarrow \pi^*$ a.s.

• RL with Almost Sure Constraints
  • Treat constraints separately or in parallel (Barrier Learner)
  • Finite State-Spaces: Can characterize all feasible policies ($D_i \equiv 0$) with finite mistakes
  • Continuous State-Spaces: Requires learning using Bellman equations with non-unique solutions

  • Takeaways:
    • Learning feasible policies is simpler than learning the optimal ones
    • Adding constraints makes optimal policies, easier to find
    • One-sided Safe Bellman can be used to find CISs that are not maximal
Thanks!

Related Publications:
[1] Castellano, Min, Bazerque, M, Reinforcement Learning with Almost Sure Constraints, L4DC, 2022

Enrique Mallada
mallada@jhu.edu
http://mallada.ece.jhu.edu