RL and MPC
Safety, Stability, and some more recent results

Sébastien Gros, Mario Zanon

Dept. of Cybernetic, NTNU
Faculty of Information Tech.

Freiburg

Outline

Outline

Safety for dynamic systems

- Let us assume that we can mathematically describe "safety" as a set of safe states:

$$\mathbf{h}\left(\mathbf{s}_k\right) \leq 0$$

Safety for dynamic systems

- Let us assume that we can mathematically describe "safety" as a set of safe states:

$$\mathbf{h}\left(\mathbf{s}_k\right) \leq 0$$

- Safe actions: $\mathbb{S}\left(\mathbf{s}_k\right)$ is the set of action $\mathbf{a}_k$ for state $\mathbf{s}_k$ such that $\mathbf{h}\left(\mathbf{s}_{k+i}\right) \leq 0$ can be enforced at all time in the future ($i > 0$)

Policy $\boldsymbol{\pi}\left(\mathbf{s}\right)$ is safe if

$$\boldsymbol{\pi}\left(\mathbf{s}\right) \in \mathbb{S}\left(\mathbf{s}\right)$$

for all states $\mathbf{s}$ that can be visited under policy $\boldsymbol{\pi}$

Safety for dynamic systems

- Let us assume that we can mathematically describe "safety" as a set of safe states:

$$\mathbf{h}(\mathbf{s}_k) \leq 0$$

- Safe actions: $\mathbb{S}(\mathbf{s}_k)$ is the set of action $\mathbf{a}_k$ for state $\mathbf{s}_k$ such that $\mathbf{h}(\mathbf{s}_{k+i}) \leq 0$ can be enforced at all time in the future ($i > 0$)

> Policy $\boldsymbol{\pi}(\mathbf{s})$ is safe if
>
> $$\boldsymbol{\pi}(\mathbf{s}) \in \mathbb{S}(\mathbf{s})$$
>
> for all states $\mathbf{s}$ that can be visited under policy $\boldsymbol{\pi}$

Remarks:

- $\mathbb{S}(\mathbf{s})$ is intrinsically *predictive* (looking into the future)

Safety for dynamic systems

- Let us assume that we can mathematically describe "safety" as a set of safe states:

$$\mathbf{h}\left(\mathbf{s}_k\right) \leq 0$$

- Safe actions: $\mathbb{S}\left(\mathbf{s}_k\right)$ is the set of action $\mathbf{a}_k$ for state $\mathbf{s}_k$ such that $\mathbf{h}\left(\mathbf{s}_{k+i}\right) \leq 0$ can be enforced at all time in the future $(i > 0)$

> Policy $\boldsymbol{\pi}\left(\mathbf{s}\right)$ is safe if
>
> $$\boldsymbol{\pi}\left(\mathbf{s}\right) \in \mathbb{S}\left(\mathbf{s}\right)$$
>
> for all states $\mathbf{s}$ that can be visited under policy $\boldsymbol{\pi}$

Remarks:

- $\mathbb{S}\left(\mathbf{s}\right)$ is intrinsically *predictive* (looking into the future)
- Computing $\mathbb{S}\left(\mathbf{s}\right)$ is "as hard as" Dynamic Programming

Safety for dynamic systems

- Let us assume that we can mathematically describe "safety" as a set of safe states:

$$\mathbf{h}\left(\mathbf{s}_k\right) \leq 0$$

- Safe actions: $\mathbb{S}\left(\mathbf{s}_k\right)$ is the set of action $\mathbf{a}_k$ for state $\mathbf{s}_k$ such that $\mathbf{h}\left(\mathbf{s}_{k+i}\right) \leq 0$ can be enforced at all time in the future ($i > 0$)

Policy $\boldsymbol{\pi}\left(\mathbf{s}\right)$ is safe if

$$\boldsymbol{\pi}\left(\mathbf{s}\right) \in \mathbb{S}\left(\mathbf{s}\right)$$

for all states $\mathbf{s}$ that can be visited under policy $\boldsymbol{\pi}$

Remarks:

- $\mathbb{S}\left(\mathbf{s}\right)$ is intrinsically *predictive* (looking into the future)
- Computing $\mathbb{S}\left(\mathbf{s}\right)$ is "as hard as" Dynamic Programming
- Data-based $\mathbb{S}\left(\mathbf{s}\right)$ (e.g. via MC sampling) requires data $\rightarrow \infty$ if safety must be ensured with probability $\rightarrow 1$

Safety for dynamic systems

- Let us assume that we can mathematically describe "safety" as a set of safe states:

$$\mathbf{h}\left(\mathbf{s}_k\right) \leq 0$$

- Safe actions: $\mathbb{S}\left(\mathbf{s}_k\right)$ is the set of action $\mathbf{a}_k$ for state $\mathbf{s}_k$ such that $\mathbf{h}\left(\mathbf{s}_{k+i}\right) \leq 0$ can be enforced at all time in the future ($i > 0$)

Policy $\boldsymbol{\pi}\left(\mathbf{s}\right)$ is safe if

$$\boldsymbol{\pi}\left(\mathbf{s}\right) \in \mathbb{S}\left(\mathbf{s}\right)$$

for all states $\mathbf{s}$ that can be visited under policy $\boldsymbol{\pi}$

Remarks:

- $\mathbb{S}\left(\mathbf{s}\right)$ is intrinsically *predictive* (looking into the future)
- Computing $\mathbb{S}\left(\mathbf{s}\right)$ is "as hard as" Dynamic Programming
- Data-based $\mathbb{S}\left(\mathbf{s}\right)$ (e.g. via MC sampling) requires data $\rightarrow \infty$ if safety must be ensured with probability $\rightarrow 1$
- Achieving $\boldsymbol{\pi}\left(\mathbf{s}\right) \in \mathbb{S}\left(\mathbf{s}\right)$ using generic function approximations (e.g. DNN) and sampling can be challenging

Let's take one step back: NLP-based Reinforcement Learning

Approximate $Q^\star$ using a parametric NLP

$$\begin{aligned} Q_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{a}) = \min_{\mathbf{w}} \quad & \Phi_{\boldsymbol{\theta}}(\mathbf{w},\mathbf{s},\mathbf{a}) \\ \text{s.t.} \quad & \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{w},\mathbf{s},\mathbf{a}) = 0 \\ & \mathbf{h}_{\boldsymbol{\theta}}(\mathbf{w},\mathbf{s},\mathbf{a}) \leq 0 \end{aligned}$$

NLP can be an MPC scheme but not necessarily

where

- current state & action $\mathbf{s}$, $\mathbf{a}$
- parameters $\boldsymbol{\theta}$ (to be adjusted by RL)
- "auxiliary variables" $\mathbf{w}$

Let's take one step back: NLP-based Reinforcement Learning

Approximate $Q^\star$ using a parametric NLP

$$\begin{aligned} Q_{\boldsymbol{\theta}}(\mathrm{s}, \mathbf{a}) = \min_{\mathbf{w}} \quad & \Phi_{\boldsymbol{\theta}}(\mathbf{w}, \mathrm{s}, \mathbf{a}) \\ \text{s.t.} \quad & \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{w}, \mathrm{s}, \mathbf{a}) = 0 \\ & \mathbf{h}_{\boldsymbol{\theta}}(\mathbf{w}, \mathrm{s}, \mathbf{a}) \leq 0 \end{aligned}$$

NLP can be an MPC scheme but not necessarily

where

- current state & action s, $\mathbf{a}$
- parameters $\boldsymbol{\theta}$ (to be adjusted by RL)
- "auxiliary variables" $\mathbf{w}$

Then

$$\begin{aligned} V_{\boldsymbol{\theta}}(\mathrm{s}) = \min_{\mathbf{w}, \mathbf{a}} \quad & \Phi_{\boldsymbol{\theta}}(\mathbf{w}, \mathrm{s}, \mathbf{a}) \\ \text{s.t.} \quad & \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{w}, \mathrm{s}, \mathbf{a}) = 0 \\ & \mathbf{h}_{\boldsymbol{\theta}}(\mathbf{w}, \mathrm{s}, \mathbf{a}) \leq 0 \end{aligned}$$

Let's take one step back: NLP-based Reinforcement Learning

Approximate $Q^\star$ using a parametric NLP

$$Q_\theta(\mathbf{s},\mathbf{a}) = \min_{\mathbf{w}} \quad \Phi_\theta(\mathbf{w},\mathbf{s},\mathbf{a}) \quad \text{s.t.} \quad \mathbf{g}_\theta(\mathbf{w},\mathbf{s},\mathbf{a}) = 0, \quad \mathbf{h}_\theta(\mathbf{w},\mathbf{s},\mathbf{a}) \leq 0$$

NLP can be an MPC scheme but not necessarily

where

- current state & action $\mathbf{s}$, $\mathbf{a}$
- parameters $\boldsymbol{\theta}$ (to be adjusted by RL)
- "auxiliary variables" $\mathbf{w}$

Then

$$V_\theta(\mathbf{s}) = \min_{\mathbf{w},\mathbf{a}} \quad \Phi_\theta(\mathbf{w},\mathbf{s},\mathbf{a}) \quad \text{s.t.} \quad \mathbf{g}_\theta(\mathbf{w},\mathbf{s},\mathbf{a}) = 0, \quad \mathbf{h}_\theta(\mathbf{w},\mathbf{s},\mathbf{a}) \leq 0$$

$$\mathbf{w}^\star(\mathbf{s}), \mathbf{a}^\star(\mathbf{s}) = \arg\min_{\mathbf{w},\mathbf{a}} \quad \Phi_\theta(\mathbf{w},\mathbf{s},\mathbf{a}) \quad \text{s.t.} \quad \mathbf{g}_\theta(\mathbf{w},\mathbf{s},\mathbf{a}) = 0, \quad \mathbf{h}_\theta(\mathbf{w},\mathbf{s},\mathbf{a}) \leq 0$$

and $\boldsymbol{\pi}_\theta(\mathbf{s}) = \mathbf{a}^\star(\mathbf{s})$

Let's take one step back: NLP-based Reinforcement Learning

Approximate $Q^{\star}$ using a parametric NLP

$$\begin{aligned} Q_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{a}) = \min_{\mathbf{w}} \quad & \Phi_{\boldsymbol{\theta}}(\mathbf{w},\mathbf{s},\mathbf{a}) \\ \text{s.t.} \quad & \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{w},\mathbf{s},\mathbf{a}) = 0 \\ & \mathbf{h}_{\boldsymbol{\theta}}(\mathbf{w},\mathbf{s},\mathbf{a}) \leq 0 \end{aligned}$$

where

- current state & action $\mathbf{s}$, $\mathbf{a}$
- parameters $\boldsymbol{\theta}$ (to be adjusted by RL)
- "auxiliary variables" $\mathbf{w}$

NLP can be an MPC scheme but not necessarily

Remarks:

- NLP can represent any function, hence this form is generic
- Can think of this as a "generalization" of RL-MPC
- Constrains can "naturally" block unsafe actions

Then

$$\begin{aligned} V_{\boldsymbol{\theta}}(\mathbf{s}) = \min_{\mathbf{w},\mathbf{a}} \quad & \Phi_{\boldsymbol{\theta}}(\mathbf{w},\mathbf{s},\mathbf{a}) \\ \text{s.t.} \quad & \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{w},\mathbf{s},\mathbf{a}) = 0 \\ & \mathbf{h}_{\boldsymbol{\theta}}(\mathbf{w},\mathbf{s},\mathbf{a}) \leq 0 \end{aligned}$$

$$\begin{aligned} \mathbf{w}^{\star}(\mathbf{s}),\mathbf{a}^{\star}(\mathbf{s}) = \arg\min_{\mathbf{w},\mathbf{a}} \quad & \Phi_{\boldsymbol{\theta}}(\mathbf{w},\mathbf{s},\mathbf{a}) \\ \text{s.t.} \quad & \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{w},\mathbf{s},\mathbf{a}) = 0 \\ & \mathbf{h}_{\boldsymbol{\theta}}(\mathbf{w},\mathbf{s},\mathbf{a}) \leq 0 \end{aligned}$$

and $\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s}) = \mathbf{a}^{\star}(\mathbf{s})$

Safety filters - Safe RL via projections

- RL can discover policy parameters $\boldsymbol{\theta}$ such that policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s})$ has good closed-loop performances, ignoring safety (e.g. $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ stems from a DNN). "Learning" safety implicitly is difficult.

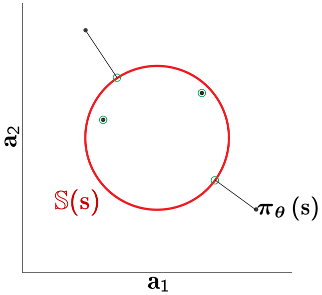
Safety filters - Safe RL via projections

- RL can discover policy parameters $\boldsymbol{\theta}$ such that policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s})$ has good closed-loop performances, ignoring safety (e.g. $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ stems from a DNN). "Learning" safety implicitly is difficult.
- If safe set $\mathbb{S}(\mathrm{s})$ is somehow known, then we can
  - follow learned policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s})$ when

$$\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s}) \in \mathbb{S}(\mathrm{s})$$

  - take "closest" action $\mathbf{a} \in \mathbb{S}(\mathrm{s})$ when

$$\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s}) \notin \mathbb{S}(\mathrm{s})$$



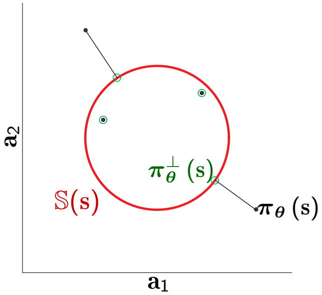
Safety filters - Safe RL via projections

- RL can discover policy parameters $\boldsymbol{\theta}$ such that policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s})$ has good closed-loop performances, ignoring safety (e.g. $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ stems from a DNN). "Learning" safety implicitly is difficult.
- If safe set $\mathbb{S}(\mathrm{s})$ is somehow known, then we can
  - follow learned policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s})$ when

$$\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s}) \in \mathbb{S}(\mathrm{s})$$

  - take "closest" action $\mathbf{a} \in \mathbb{S}(\mathrm{s})$ when

$$\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s}) \notin \mathbb{S}(\mathrm{s})$$



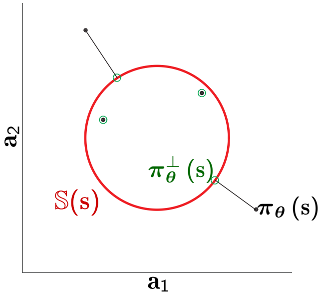
More formally, safe policy e.g. reads as...

$$\begin{aligned} \boldsymbol{\pi}_{\boldsymbol{\theta}}^{\perp}(\mathrm{s}) = \arg\min_{\mathbf{a}} \quad & \|\mathbf{a} - \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s})\|^2 \\ \text{s.t.} \quad & \mathbf{a} \in \mathbb{S}(\mathrm{s}) \end{aligned}$$

...though other norms or penalties than $\|.\|^2$ could be used

Safety filters - Safe RL via projections

- RL can discover policy parameters $\boldsymbol{\theta}$ such that policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s})$ has good closed-loop performances, ignoring safety (e.g. $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ stems from a DNN). "Learning" safety implicitly is difficult.
- If safe set $\mathbb{S}(\mathrm{s})$ is somehow known, then we can



- follow learned policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s})$ when

$$\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s}) \in \mathbb{S}(\mathrm{s})$$

- take "closest" action $\mathbf{a} \in \mathbb{S}(\mathrm{s})$ when

$$\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s}) \notin \mathbb{S}(\mathrm{s})$$

More formally, safe policy e.g. reads as...

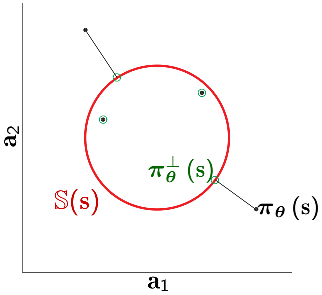
$$\begin{aligned} \boldsymbol{\pi}^{\perp}_{\boldsymbol{\theta}}(\mathrm{s}) = \arg\min_{\mathbf{a}} \quad & \|\mathbf{a} - \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s})\|^2 \\ \text{s.t.} \quad & \mathbf{a} \in \mathbb{S}(\mathrm{s}) \end{aligned}$$

...though other norms or penalties than $\|.\|^2$ could be used

Is that a good idea?

Safety filters - Safe RL via projections

- RL can discover policy parameters $\boldsymbol{\theta}$ such that policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s})$ has good closed-loop performances, ignoring safety (e.g. $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ stems from a DNN). "Learning" safety implicitly is difficult.
- If safe set $\mathbb{S}(\mathrm{s})$ is somehow known, then we can
  - follow learned policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s})$ when
  $$\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s}) \in \mathbb{S}(\mathrm{s})$$
  - take "closest" action $\mathbf{a} \in \mathbb{S}(\mathrm{s})$ when
  $$\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s}) \notin \mathbb{S}(\mathrm{s})$$



More formally, safe policy e.g. reads as...

$$\begin{aligned} \boldsymbol{\pi}_{\boldsymbol{\theta}}^{\perp}(\mathrm{s}) = \arg\min_{\mathbf{a}} \quad & \left\|\mathbf{a} - \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s})\right\|^2 \\ \mathrm{s.t.} \quad & \mathbf{a} \in \mathbb{S}(\mathrm{s}) \end{aligned}$$

...though other norms or penalties than $\|.\|^2$ could be used

Is that a good idea? It depends...

Safety filters - How to obtain optimality?

Q learning: $Q_\theta \approx Q^\star$ learned via classic RL, ignoring safety.

Safety filters - How to obtain optimality?

Q learning: $Q_\theta \approx Q^\star$ learned via classic RL, ignoring safety. Then

$$\boldsymbol{\pi}_{\boldsymbol{\theta}}^{\perp}(\mathrm{s}) = \arg\min_{\mathbf{a}} \quad \|\mathbf{a} - \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s})\|^2 \quad \text{s.t.} \quad \mathbf{a} \in \mathbb{S}(\mathrm{s})$$

where

$$\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s}) = \arg\min \; Q_{\boldsymbol{\theta}}(\mathrm{s}, \mathbf{a})$$

Safety filters - How to obtain optimality?

Q learning: $Q_\theta \approx Q^\star$ learned via classic RL, ignoring safety. Then

$$\boldsymbol{\pi}_{\boldsymbol{\theta}}^{\perp}(\mathrm{s}) = \arg\min_{\mathbf{a}} \quad \|\mathbf{a} - \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s})\|^2 \qquad \text{s.t.} \quad \mathbf{a} \in \mathbb{S}(\mathrm{s})$$

where

$$\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s}) = \arg\min \; Q_{\boldsymbol{\theta}}(\mathrm{s}, \mathbf{a})$$

yields suboptimal policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}^{\perp}$

Safety filters - How to obtain optimality?

Q learning: $Q_{\theta} \approx Q^{\star}$ learned via classic RL, ignoring safety. Then

$$\begin{aligned} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s}) = \arg\min_{\mathbf{a}} \quad & Q_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{a}) \\ \text{s.t.} \quad & \mathbf{a} \in \mathbb{S}(\mathbf{s}) \end{aligned}$$

instead of a least-squares approach. Provably optimal (safe) policy.

Safety filters - How to obtain optimality?

Q learning: $Q_\theta \approx Q^\star$ learned via classic RL, ignoring safety. Then

$$\begin{aligned} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s}) = \arg\min_{\mathbf{a}} \quad & Q_{\boldsymbol{\theta}}(\mathrm{s}, \mathbf{a}) \\ \text{s.t.} \quad & \mathbf{a} \in \mathbb{S}(\mathrm{s}) \end{aligned}$$

instead of a least-squares approach. Provably optimal (safe) policy.

Deterministic Policy gradient (actor-critic): the “regular expression”

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\pi}_{\boldsymbol{\theta}}) = \mathbb{E}\left[\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}} \nabla_{\mathrm{a}} A_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}\right]$$

yields incorrect gradients

Safety filters - How to obtain optimality?

Q learning: $Q_{\theta} \approx Q^{\star}$ learned via classic RL, ignoring safety. Then

$$\begin{aligned} \boldsymbol{\pi}_{\theta}(\mathrm{s}) = \arg\min_{\mathbf{a}} \quad & Q_{\theta}(\mathrm{s}, \mathbf{a}) \\ \text{s.t.} \quad & \mathbf{a} \in \mathbb{S}(\mathrm{s}) \end{aligned}$$

instead of a least-squares approach. Provably optimal (safe) policy.

Deterministic Policy gradient (actor-critic): make sure to evaluate the gradient using

$$\nabla_{\theta} J\left(\boldsymbol{\pi}_{\theta}^{\perp}\right) = \mathbb{E}\left[\nabla_{\theta}\boldsymbol{\pi}_{\theta}^{\perp}\nabla_{\mathrm{a}} A_{\boldsymbol{\pi}_{\theta}^{\perp}}\right]$$

where

$$\begin{aligned} \boldsymbol{\pi}_{\theta}^{\perp}(\mathrm{s}) = \arg\min_{\mathbf{a}} \quad & \|\mathbf{a} - \boldsymbol{\pi}_{\theta}(\mathrm{s})\|^2 \\ \text{s.t.} \quad & \mathbf{a} \in \mathbb{S}(\mathrm{s}) \end{aligned}$$

i.e. **account for projection (differentiate NLP). Provably correct gradients.**

Safety filters - How to obtain optimality?

Q learning: $Q_\theta \approx Q^\star$ learned via classic RL, ignoring safety. Then

$$\begin{aligned} \boldsymbol{\pi}_\theta(\mathbf{s}) = \arg\min_{\mathbf{a}} \quad & Q_\theta(\mathbf{s},\mathbf{a}) \\ \text{s.t.} \quad & \mathbf{a} \in \mathbb{S}(\mathbf{s}) \end{aligned}$$

instead of a least-squares approach. Provably optimal (safe) policy.

Deterministic Policy gradient (actor-critic): make sure to evaluate the gradient using

$$\nabla_\theta J\left(\boldsymbol{\pi}_\theta^\perp\right) = \mathbb{E}\left[\nabla_\theta \boldsymbol{\pi}_\theta^\perp \nabla_{\mathbf{a}} A_{\boldsymbol{\pi}_\theta^\perp}\right] \quad \text{where} \quad \begin{aligned} \boldsymbol{\pi}_\theta^\perp(\mathbf{s}) = \arg\min_{\mathbf{a}} \quad & \|\mathbf{a} - \boldsymbol{\pi}_\theta(\mathbf{s})\|^2 \\ \text{s.t.} \quad & \mathbf{a} \in \mathbb{S}(\mathbf{s}) \end{aligned}$$

i.e. **account for projection (differentiate NLP). Provably correct gradients.**

Stochastic policy gradient: where π_θ is a probability density over the actions

$$\nabla_\theta J\left(\pi_\theta^\perp\right) = \mathbb{E}\left[\log \nabla_\theta \pi_\theta \nabla_{\mathbf{a}} A_{\pi_\theta^\perp}\right]$$

i.e. **do not account for projection (cannot). Provably correct gradients.**

Safety filters - How to obtain optimality?

Q learning: $Q_\theta \approx Q^\star$ learned via classic RL, ignoring safety. Then

$$\begin{aligned} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s}) = \arg\min_{\mathbf{a}} \quad & Q_{\boldsymbol{\theta}}(\mathrm{s}, \mathbf{a}) \\ \text{s.t.} \quad & \mathbf{a} \in \mathbb{S}(\mathrm{s}) \end{aligned}$$

instead of a least-squares approach. Provably optimal (safe) policy.

Deterministic Policy gradient (actor-critic): make sure to evaluate the gradient using

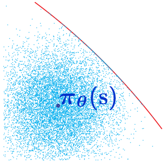
$$\nabla_{\boldsymbol{\theta}} J\left(\boldsymbol{\pi}_{\boldsymbol{\theta}}^{\perp}\right) = \mathbb{E}\left[\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}^{\perp} \nabla_{\mathrm{a}} A_{\boldsymbol{\pi}_{\boldsymbol{\theta}}^{\perp}}\right] \quad \text{where} \quad \begin{aligned} \boldsymbol{\pi}_{\boldsymbol{\theta}}^{\perp}(\mathrm{s}) = \arg\min_{\mathbf{a}} \quad & \left\|\mathbf{a} - \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s})\right\|^2 \\ \text{s.t.} \quad & \mathbf{a} \in \mathbb{S}(\mathrm{s}) \end{aligned}$$

i.e. **account for projection (differentiate NLP). Provably correct gradients.**

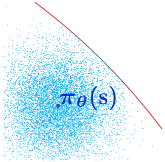
Safe Reinforcement Learning via projection on a safe set: how to achieve optimality? S. Gros, M. Zanon, A. Bemporad, IFAC 2020

Safe exploration

Learning requires exploration. E.g. apply $\mathbf{a} = \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s}) + \mathbf{d}$ to the real system where $\mathbf{d}$ is a "disturbance"



Safe exploration



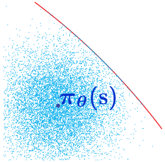
Learning requires exploration. E.g. apply $\mathbf{a} = \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s}) + \mathbf{d}$ to the real system where $\mathbf{d}$ is a "disturbance"

Explore while keeping $\mathbf{a} \in \mathbb{S}(\mathbf{s})$?

- Clearly an arbitrary "policy disturbance" $\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s}) + \mathbf{d}$ is a poor idea...
- NLP-based policy: "disturb" the cost function instead! (different options)

Safe exploration

Learning requires exploration. E.g. apply $\mathbf{a} = \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s}) + \mathbf{d}$ to the real system where $\mathbf{d}$ is a "disturbance"



Explore while keeping $\mathbf{a} \in \mathbb{S}(\mathbf{s})$?

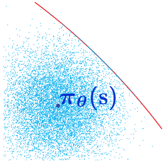
- Clearly an arbitrary "policy disturbance" $\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s}) + \mathbf{d}$ is a poor idea...
- NLP-based policy: "disturb" the cost function instead! (different options)

Safe policy given by $\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s}) = \mathbf{a}_0^\star(\mathbf{s})$ with

$$\begin{aligned} \min_{\mathbf{w},\mathbf{a}} \quad & \Phi_{\boldsymbol{\theta}}(\mathbf{w}, \mathbf{s}, \mathbf{a}) \\ \text{s.t.} \quad & \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{w}, \mathbf{s}, \mathbf{a}) = 0 \\ & \mathbf{h}_{\boldsymbol{\theta}}(\mathbf{w}, \mathbf{s}, \mathbf{a}) \leq 0 \end{aligned}$$

Safe exploration

Learning requires exploration. E.g. apply $\mathbf{a} = \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s}) + \mathbf{d}$ to the real system where $\mathbf{d}$ is a "disturbance"



Explore while keeping $\mathbf{a} \in \mathbb{S}(\mathbf{s})$?

- Clearly an arbitrary "policy disturbance" $\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s}) + \mathbf{d}$ is a poor idea...
- NLP-based policy: "disturb" the cost function instead! (different options)

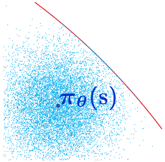
Safe policy with exploration: $\boldsymbol{\pi}^{\mathrm{e}}_{\boldsymbol{\theta}}$ given by

$$\begin{aligned} \min_{\mathbf{w},\mathbf{a}} \quad & \Phi_{\boldsymbol{\theta}}(\mathbf{w}, \mathbf{s}, \mathbf{a}) - \mathbf{d}^\top \mathbf{a} \\ \text{s.t.} \quad & \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{w}, \mathbf{s}, \mathbf{a}) = 0 \\ & \mathbf{h}_{\boldsymbol{\theta}}(\mathbf{w}, \mathbf{s}, \mathbf{a}) \leq 0 \end{aligned}$$

satisfies the constraints by construction

Safe exploration

Learning requires exploration. E.g. apply $\mathbf{a} = \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s}) + \mathbf{d}$ to the real system where $\mathbf{d}$ is a "disturbance"



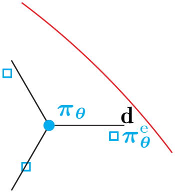
Explore while keeping $\mathbf{a} \in \mathbb{S}(\mathbf{s})$?

- Clearly an arbitrary "policy disturbance" $\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s}) + \mathbf{d}$ is a poor idea...
- NLP-based policy: "disturb" the cost function instead! (different options)

Safe policy with exploration: $\boldsymbol{\pi}^{\mathrm{e}}_{\boldsymbol{\theta}}$ given by

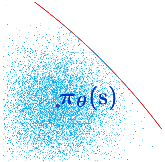
$$\begin{aligned} \min_{\mathbf{w},\mathbf{a}} \quad & \Phi_{\boldsymbol{\theta}}(\mathbf{w},\mathbf{s},\mathbf{a}) - \mathbf{d}^\top \mathbf{a} \\ \text{s.t.} \quad & \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{w},\mathbf{s},\mathbf{a}) = 0 \\ & \mathbf{h}_{\boldsymbol{\theta}}(\mathbf{w},\mathbf{s},\mathbf{a}) \leq 0 \end{aligned}$$

satisfies the constraints by construction



Safe exploration

Learning requires exploration. E.g. apply $\mathbf{a} = \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s}) + \mathbf{d}$ to the real system where $\mathbf{d}$ is a "disturbance"



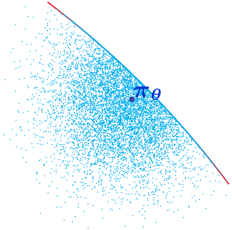
Explore while keeping $\mathbf{a} \in \mathbb{S}(\mathbf{s})$?

- Clearly an arbitrary "policy disturbance" $\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s}) + \mathbf{d}$ is a poor idea...
- NLP-based policy: "disturb" the cost function instead! (different options)

Safe policy with exploration: $\boldsymbol{\pi}^{\mathrm{e}}_{\boldsymbol{\theta}}$ given by

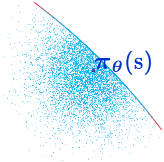
$$\begin{aligned} \min_{\mathbf{w},\mathbf{a}} \quad & \Phi_{\boldsymbol{\theta}}(\mathbf{w}, \mathbf{s}, \mathbf{a}) - \mathbf{d}^\top \mathbf{a} \\ \text{s.t.} \quad & \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{w}, \mathbf{s}, \mathbf{a}) = 0 \\ & \mathbf{h}_{\boldsymbol{\theta}}(\mathbf{w}, \mathbf{s}, \mathbf{a}) \leq 0 \end{aligned}$$

satisfies the constraints by construction



Safe exploration

Learning requires exploration. E.g. apply $\mathbf{a} = \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s}) + \mathbf{d}$ to the real system where $\mathbf{d}$ is a "disturbance"



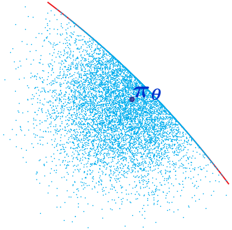
Explore while keeping $\mathbf{a} \in \mathbb{S}(\mathbf{s})$?

- Clearly an arbitrary "policy disturbance" $\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s}) + \mathbf{d}$ is a poor idea...
- NLP-based policy: "disturb" the cost function instead! (different options)

Safe policy with exploration: $\boldsymbol{\pi}^{\mathrm{e}}_{\boldsymbol{\theta}} = \mathbf{a}_0^\star$:

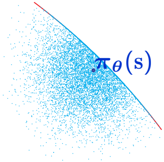
$$\begin{aligned} \min_{\mathbf{s},\mathbf{a}} \quad & T(\mathbf{s}_N) - \mathbf{d}^\top \mathbf{a}_0 + \sum_{k=0}^{N-1} L(\mathbf{s}_k, \mathbf{a}_k) \\ \text{s.t.} \quad & \mathbf{s}_{k+1} = \mathbf{f}(\mathbf{s}_k, \mathbf{a}_k) \\ & \mathbf{h}(\mathbf{s}_k, \mathbf{a}_k) \leq 0, \quad \mathbf{s}_N \in \mathbb{T} \end{aligned}$$

satisfies the constraints by construction



Safe exploration

Learning requires exploration. E.g. apply $\mathbf{a} = \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s}) + \mathbf{d}$ to the real system where $\mathbf{d}$ is a "disturbance"



Explore while keeping $\mathbf{a} \in \mathbb{S}(\mathbf{s})$?

- Clearly an arbitrary "policy disturbance" $\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s}) + \mathbf{d}$ is a poor idea...
- NLP-based policy: "disturb" the cost function instead! (different options)

Safe policy with exploration: $\boldsymbol{\pi}^{\mathrm{e}}_{\boldsymbol{\theta}} = \mathbf{a}_0^\star$:

$$\begin{aligned} \min_{\mathbf{s},\mathbf{a}} \quad & T(\mathbf{s}_N) - \mathbf{d}^\top \mathbf{a}_0 + \sum_{k=0}^{N-1} L(\mathbf{s}_k, \mathbf{a}_k) \\ \mathrm{s.t.} \quad & \mathbf{s}_{k+1} = \mathbf{f}(\mathbf{s}_k, \mathbf{a}_k) \\ & \mathbf{h}(\mathbf{s}_k, \mathbf{a}_k) \leq 0, \quad \mathbf{s}_N \in \mathbb{T} \end{aligned}$$

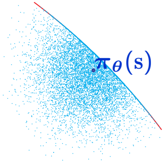
satisfies the constraints by construction

Remarks:

- Exploration $\mathbf{e} = \boldsymbol{\pi}^{\mathrm{e}}_{\boldsymbol{\theta}} - \boldsymbol{\pi}_{\boldsymbol{\theta}}$ is not centred-isotopric
- Can create some technical issues with actor-critic methods (linear compatible $A_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}$), biased policy gradient estimation
- Bias seems not necessarily large in practice

Safe exploration

Learning requires exploration. E.g. apply $\mathbf{a} = \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s}) + \mathbf{d}$ to the real system where $\mathbf{d}$ is a "disturbance"



Explore while keeping $\mathbf{a} \in \mathbb{S}(\mathbf{s})$?

- Clearly an arbitrary "policy disturbance" $\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s}) + \mathbf{d}$ is a poor idea...
- NLP-based policy: "disturb" the cost function instead! (different options)

Safe policy with exploration: $\boldsymbol{\pi}^{\mathrm{e}}_{\boldsymbol{\theta}} = \mathbf{a}_0^\star$:

$$\begin{aligned} \min_{\mathbf{s},\mathbf{a}} \quad & T(\mathbf{s}_N) - \mathbf{d}^\top \mathbf{a}_0 + \sum_{k=0}^{N-1} L(\mathbf{s}_k, \mathbf{a}_k) \\ \text{s.t.} \quad & \mathbf{s}_{k+1} = \mathbf{f}(\mathbf{s}_k, \mathbf{a}_k) \\ & \mathbf{h}(\mathbf{s}_k, \mathbf{a}_k) \leq 0, \quad \mathbf{s}_N \in \mathbb{T} \end{aligned}$$

satisfies the constraints by construction

Remarks:

- Exploration $\mathbf{e} = \boldsymbol{\pi}^{\mathrm{e}}_{\boldsymbol{\theta}} - \boldsymbol{\pi}_{\boldsymbol{\theta}}$ is not centred-isotopric
- Can create some technical issues with actor-critic methods (linear compatible $A_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}$), biased policy gradient estimation
- Bias seems not necessarily large in practice

Bias Correction in Reinforcement Learning via the Deterministic Policy Gradient Method for MPC-Based Policies, S. Gros, M. Zanon, ACC 2021

Bias Correction in Deterministic Policy Gradient Using Robust MPC, A. Kordabad, S. Gros ECC 2021

Outline

Robust MPC - Uncertainty model

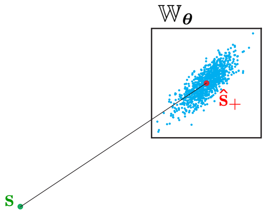
True system: $\mathbf{s}_+ \sim \mathbb{P}\left[\,\cdot\,|\mathbf{s},\mathbf{a}\,\right]$

Deterministic model: $\hat{\mathbf{s}}_+ = \mathbf{f}_{\boldsymbol{\theta}}\left(\mathbf{s},\mathbf{a}\right)$

Robust MPC - Uncertainty model

True system: $\mathbf{s}_+ \sim \mathbb{P}[\,\cdot\,|\mathbf{s},\mathbf{a}\,]$

Deterministic model: $\hat{\mathbf{s}}_+ = \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{a})$



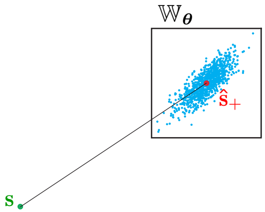
Dispersion: $\mathbf{f}(\mathbf{s},\mathbf{a}) + \mathbb{W}_{\boldsymbol{\theta}}$ contains the support of $\mathbb{P}[\,\cdot\,|\mathbf{s},\mathbf{a}\,]$, i.e.

$$\mathbf{s}_+ \in \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{a}) + \mathbb{W}_{\boldsymbol{\theta}} \tag{1}$$

with probability 1

Robust MPC - Uncertainty model

$$\text{True system:} \quad \mathbf{s}_+ \sim \mathbb{P}\left[\,\cdot\,|\mathbf{s},\mathbf{a}\,\right]$$
$$\text{Deterministic model:} \quad \hat{\mathbf{s}}_+ = \mathbf{f}_{\boldsymbol{\theta}}\left(\mathbf{s},\mathbf{a}\right)$$



Dispersion: $\mathbf{f}\left(\mathbf{s},\mathbf{a}\right) + \mathbb{W}_{\boldsymbol{\theta}}$ contains the support of $\mathbb{P}\left[\,\cdot\,|\mathbf{s},\mathbf{a}\,\right]$, i.e.

$$\mathbf{s}_+ \in \mathbf{f}_{\boldsymbol{\theta}}\left(\mathbf{s},\mathbf{a}\right) + \mathbb{W}_{\boldsymbol{\theta}} \tag{1}$$

with probability 1

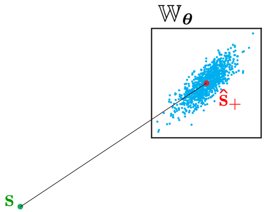
Remarks:

- Identifying $\mathbb{W}_{\boldsymbol{\theta}}$ is a set-membership identification problem, well studied
- Obviously $\mathbb{W}_{\boldsymbol{\theta}}$ is not unique
- Ensuring probability 1 is not possible $\rightarrow$ probabilistic guarantees
- Model parameters $\boldsymbol{\theta}$ must be such that (1) holds on every known data point

Robust MPC - Uncertainty model

True system: $\mathbf{s}_+ \sim \mathbb{P}\left[\,\cdot\,|\mathbf{s},\mathbf{a}\,\right]$

Deterministic model: $\hat{\mathbf{s}}_+ = \mathbf{f}_{\boldsymbol{\theta}}\left(\mathbf{s},\mathbf{a}\right)$



Dispersion: $\mathbf{f}\left(\mathbf{s},\mathbf{a}\right) + \mathbb{W}_{\boldsymbol{\theta}}$ contains the support of $\mathbb{P}\left[\,\cdot\,|\mathbf{s},\mathbf{a}\,\right]$, i.e.

$$\mathbf{s}_+ \in \mathbf{f}_{\boldsymbol{\theta}}\left(\mathbf{s},\mathbf{a}\right) + \mathbb{W}_{\boldsymbol{\theta}} \tag{1}$$

with probability 1

Condition

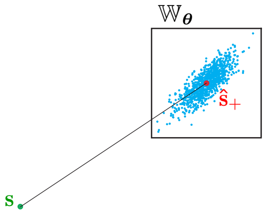
$$\mathbf{s}_+ - \mathbf{f}_{\boldsymbol{\theta}}\left(\mathbf{s},\mathbf{a}\right) \in \mathbb{W}_{\boldsymbol{\theta}}$$

for all observed triplets $(\mathbf{s},\mathbf{a},\mathbf{s}_+)$
$\rightarrow$ constraints on $\boldsymbol{\theta}$

Remarks:

- Identifying $\mathbb{W}_{\boldsymbol{\theta}}$ is a set-membership identification problem, well studied
- Obviously $\mathbb{W}_{\boldsymbol{\theta}}$ is not unique
- Ensuring probability 1 is not possible $\rightarrow$ probabilistic guarantees
- Model parameters $\boldsymbol{\theta}$ must be such that (1) holds on every known data point

Robust MPC - Uncertainty model



$$\text{True system:} \quad \mathbf{s}_+ \sim \mathbb{P}\left[\,\cdot\,|\mathbf{s},\mathbf{a}\,\right]$$
$$\text{Deterministic model:} \quad \hat{\mathbf{s}}_+ = \mathbf{f}_{\boldsymbol{\theta}}\left(\mathbf{s},\mathbf{a}\right)$$

Dispersion: $\mathbf{f}\left(\mathbf{s},\mathbf{a}\right) + \mathbb{W}_{\boldsymbol{\theta}}$ contains the support of $\mathbb{P}\left[\,\cdot\,|\mathbf{s},\mathbf{a}\,\right]$, i.e.

$$\mathbf{s}_+ \in \mathbf{f}_{\boldsymbol{\theta}}\left(\mathbf{s},\mathbf{a}\right) + \mathbb{W}_{\boldsymbol{\theta}} \tag{1}$$

with probability 1

Remarks:

- Identifying $\mathbb{W}_{\boldsymbol{\theta}}$ is a set-membership identification problem, well studied
- Obviously $\mathbb{W}_{\boldsymbol{\theta}}$ is not unique
- Ensuring probability 1 is not possible $\rightarrow$ probabilistic guarantees
- Model parameters $\boldsymbol{\theta}$ must be such that (1) holds on every known data point

Condition

$$\mathbf{s}_+ - \mathbf{f}_{\boldsymbol{\theta}}\left(\mathbf{s},\mathbf{a}\right) \in \mathbb{W}_{\boldsymbol{\theta}}$$

for all observed triplets $(\mathbf{s},\mathbf{a},\mathbf{s}_+)$
$\rightarrow$ constraints on $\boldsymbol{\theta}$

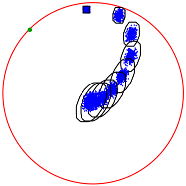
Containing the model-system mismatch becomes constraints in the parameters $\boldsymbol{\theta}$. Constraints can be readily formulated in terms of data.

Safe policies via robust (N)MPC

Robust (N)MPC delivers policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{x}_0) = \mathbf{u}_0^\star$ from

$$\mathbf{u}^\star = \arg\min_{\mathbf{u}} \max_{\mathbf{w}\in\mathbb{W}_{\boldsymbol{\theta}}{}^N} T_{\boldsymbol{\theta}}(\mathbf{x}_N) + \sum_{k=0}^{N-1} L_{\boldsymbol{\theta}}(\mathbf{x}_k, \mathbf{u}_k)$$

$$\text{s.t.} \quad \mathbf{u}_{0,\dots,N} \in \mathbb{U}$$



- $\mathbf{x}_{0,\dots,N}$ is the propagation of the state dispersion
- max cost treats a worst-case scenario, required for stability
- $\mathbf{w} = \{\mathbf{w}_0, \dots, \mathbf{w}_N\}$ is the disturbance with $\mathbf{w}_k \in \mathbb{W}_{\boldsymbol{\theta}}$

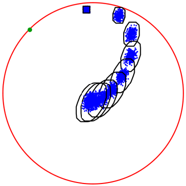
Safe policies via robust (N)MPC

Robust (N)MPC delivers policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{x}_0) = \mathbf{u}_0^\star$ from

$$\mathbf{u}^\star = \arg\min_{\mathbf{u}} \max_{\mathbf{w}\in\mathbb{W}_{\boldsymbol{\theta}}{}^N} T_{\boldsymbol{\theta}}(\mathbf{x}_N) + \sum_{k=0}^{N-1} L_{\boldsymbol{\theta}}(\mathbf{x}_k, \mathbf{u}_k)$$

$$\text{s.t.} \quad \mathbf{u}_{0,\ldots,N} \in \mathbb{U}$$

$$\mathbf{x}_{1,\ldots,N-1}(\mathbf{u}, \mathbf{x}_0, \boldsymbol{\theta}, \mathbf{w}) \in \mathbb{X}, \quad \forall\, \mathbf{w} \in \mathbb{W}_{\boldsymbol{\theta}}{}^{N-1}$$

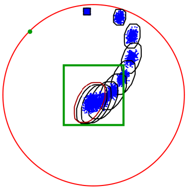


- $\mathbf{x}_{0,\ldots,N}$ is the propagation of the state dispersion
- max cost treats a worst-case scenario, required for stability
- $\mathbf{w} = \{\mathbf{w}_0, \ldots, \mathbf{w}_N\}$ is the disturbance with $\mathbf{w}_k \in \mathbb{W}_{\boldsymbol{\theta}}$
- $\mathbf{x}_{1,\ldots,N-1}(\mathbf{u}, \mathbf{x}_0, \boldsymbol{\theta}, \mathbf{w})$ are the trajectories subject to $\mathbf{w}$
- $\mathbb{X}$ is the "safe" set where the state should be at all time

Safe policies via robust (N)MPC

Robust (N)MPC delivers policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{x}_0) = \mathbf{u}_0^\star$ from

$$\begin{aligned}
\mathbf{u}^\star = \arg\min_{\mathbf{u}} \max_{\mathbf{w}\in\mathbb{W}_{\boldsymbol{\theta}}{}^N} \quad & T_{\boldsymbol{\theta}}\left(\mathbf{x}_N\right) + \sum_{k=0}^{N-1} L_{\boldsymbol{\theta}}\left(\mathbf{x}_k, \mathbf{u}_k\right) \\
\text{s.t.} \quad & \mathbf{u}_{0,\dots,N} \in \mathbb{U} \\
& \mathbf{x}_{1,\dots,N-1}\left(\mathbf{u}, \mathbf{x}_0, \boldsymbol{\theta}, \mathbf{w}\right) \in \mathbb{X}, \quad \forall\, \mathbf{w} \in \mathbb{W}_{\boldsymbol{\theta}}{}^{N-1} \\
& \mathbf{x}_N\left(\mathbf{u}, \mathbf{x}_0, \boldsymbol{\theta}, \mathbf{w}\right) \in \mathbb{T}_{\boldsymbol{\theta}}, \quad \forall\, \mathbf{w} \in \mathbb{W}_{\boldsymbol{\theta}}{}^{N-1}
\end{aligned}$$



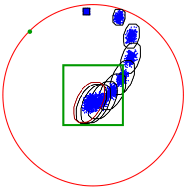
- $\mathbf{x}_{0,\dots,N}$ is the propagation of the state dispersion
- max cost treats a worst-case scenario, required for stability
- $\mathbf{w} = \{\mathbf{w}_0, \dots, \mathbf{w}_N\}$ is the disturbance with $\mathbf{w}_k \in \mathbb{W}_{\boldsymbol{\theta}}$
- $\mathbf{x}_{1,\dots,N-1}\left(\mathbf{u}, \mathbf{x}_0, \boldsymbol{\theta}, \mathbf{w}\right)$ are the trajectories subject to $\mathbf{w}$
- $\mathbb{X}$ is the "safe" set where the state should be at all time
- Terminal set $\mathbb{T}_{\boldsymbol{\theta}}$ (required for recursive feasibility & stability)

Safe policies via robust (N)MPC

Robust (N)MPC delivers policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{x}_0) = \mathbf{u}_0^\star$ from

$$\mathbf{u}^\star = \arg\min_{\mathbf{u}} \max_{\mathbf{w}\in\mathbb{W}_{\boldsymbol{\theta}}^N} T_{\boldsymbol{\theta}}\left(\mathbf{x}_N\right) + \sum_{k=0}^{N-1} L_{\boldsymbol{\theta}}\left(\mathbf{x}_k, \mathbf{u}_k\right)$$

$$\begin{aligned} \text{s.t.} \quad & \mathbf{u}_{0,\dots,N} \in \mathbb{U} \\ & \mathbf{x}_{1,\dots,N-1}\left(\mathbf{u}, \mathbf{x}_0, \boldsymbol{\theta}, \mathbf{w}\right) \in \mathbb{X}, \quad \forall\, \mathbf{w} \in {\mathbb{W}_{\boldsymbol{\theta}}}^{N-1} \\ & \mathbf{x}_N\left(\mathbf{u}, \mathbf{x}_0, \boldsymbol{\theta}, \mathbf{w}\right) \in \mathbb{T}_{\boldsymbol{\theta}}, \quad \forall\, \mathbf{w} \in {\mathbb{W}_{\boldsymbol{\theta}}}^{N-1} \end{aligned}$$

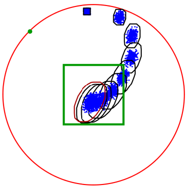


- $\mathbf{x}_{0,\dots,N}$ is the propagation of the state dispersion
- max cost treats a worst-case scenario, required for stability
- $\mathbf{w} = \{\mathbf{w}_0, \dots, \mathbf{w}_N\}$ is the disturbance with $\mathbf{w}_k \in \mathbb{W}_{\boldsymbol{\theta}}$
- $\mathbf{x}_{1,\dots,N-1}\left(\mathbf{u}, \mathbf{x}_0, \boldsymbol{\theta}, \mathbf{w}\right)$ are the trajectories subject to $\mathbf{w}$
- $\mathbb{X}$ is the "safe" set where the state should be at all time
- Terminal set $\mathbb{T}_{\boldsymbol{\theta}}$ (required for recursive feasibility & stability)
- If $\boldsymbol{\theta}$ is such that $\mathbb{W}_{\boldsymbol{\theta}}$ encloses state dispersion, MPC is safe

Safe policies via robust (N)MPC

Robust (N)MPC delivers policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{x}_0) = \mathbf{u}_0^\star$ from

$$\begin{aligned} \mathbf{u}^\star = \arg\min_{\mathbf{u}} \; & \max_{\mathbf{w}\in\mathbb{W}_{\boldsymbol{\theta}}{}^N} \; T_{\boldsymbol{\theta}}(\mathbf{x}_N) + \sum_{k=0}^{N-1} L_{\boldsymbol{\theta}}(\mathbf{x}_k, \mathbf{u}_k) \\ \text{s.t.} \quad & \mathbf{u}_{0,\dots,N} \in \mathbb{U} \\ & \mathbf{x}_{1,\dots,N-1}(\mathbf{u}, \mathbf{x}_0, \boldsymbol{\theta}, \mathbf{w}) \in \mathbb{X}, \quad \forall\, \mathbf{w} \in \mathbb{W}_{\boldsymbol{\theta}}{}^{N-1} \\ & \mathbf{x}_N(\mathbf{u}, \mathbf{x}_0, \boldsymbol{\theta}, \mathbf{w}) \in \mathbb{T}_{\boldsymbol{\theta}}, \quad \forall\, \mathbf{w} \in \mathbb{W}_{\boldsymbol{\theta}}{}^{N-1} \end{aligned}$$



- $\mathbf{x}_{0,\dots,N}$ is the propagation of the state dispersion
- max cost treats a worst-case scenario, required for stability
- $\mathbf{w} = \{\mathbf{w}_0, \dots, \mathbf{w}_N\}$ is the disturbance with $\mathbf{w}_k \in \mathbb{W}_{\boldsymbol{\theta}}$
- $\mathbf{x}_{1,\dots,N-1}(\mathbf{u}, \mathbf{x}_0, \boldsymbol{\theta}, \mathbf{w})$ are the trajectories subject to $\mathbf{w}$
- $\mathbb{X}$ is the "safe" set where the state should be at all time
- Terminal set $\mathbb{T}_{\boldsymbol{\theta}}$ (required for recursive feasibility & stability)
- If $\boldsymbol{\theta}$ is such that $\mathbb{W}_{\boldsymbol{\theta}}$ encloses state dispersion, MPC is safe

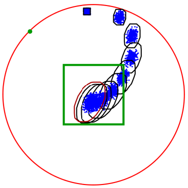
Closed-loop stability under some conditions on $\boldsymbol{\theta}$ (not trivial), need $\gamma = 1$ (for now)

Safe policies via robust (N)MPC

Robust (N)MPC delivers policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{x}_0) = \mathbf{u}_0^\star$ from

$$\mathbf{u}^\star = \arg\min_{\mathbf{u}} \max_{\mathbf{w}\in\mathbb{W}_{\boldsymbol{\theta}}{}^N} \quad T_{\boldsymbol{\theta}}(\mathbf{x}_N) + \sum_{k=0}^{N-1} L_{\boldsymbol{\theta}}(\mathbf{x}_k, \mathbf{u}_k)$$

$$\begin{aligned} \text{s.t.} \quad & \mathbf{u}_{0,\dots,N} \in \mathbb{U} \\ & \mathbf{x}_{1,\dots,N-1}(\mathbf{u}, \mathbf{x}_0, \boldsymbol{\theta}, \mathbf{w}) \in \mathbb{X}, \quad \forall\, \mathbf{w} \in \mathbb{W}_{\boldsymbol{\theta}}{}^{N-1} \\ & \mathbf{x}_N(\mathbf{u}, \mathbf{x}_0, \boldsymbol{\theta}, \mathbf{w}) \in \mathbb{T}_{\boldsymbol{\theta}}, \quad \forall\, \mathbf{w} \in \mathbb{W}_{\boldsymbol{\theta}}{}^{N-1} \end{aligned}$$



$$\nabla_{\boldsymbol{\theta}} J = \mathbb{E}\left[\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}} \nabla_{\mathbf{u}} A_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}\right]$$

- $\mathbf{x}_{0,\dots,N}$ is the propagation of the state dispersion
- max cost treats a worst-case scenario, required for stability
- $\mathbf{w} = \{\mathbf{w}_0, \dots, \mathbf{w}_N\}$ is the disturbance with $\mathbf{w}_k \in \mathbb{W}_{\boldsymbol{\theta}}$
- $\mathbf{x}_{1,\dots,N-1}(\mathbf{u}, \mathbf{x}_0, \boldsymbol{\theta}, \mathbf{w})$ are the trajectories subject to $\mathbf{w}$
- $\mathbb{X}$ is the "safe" set where the state should be at all time
- Terminal set $\mathbb{T}_{\boldsymbol{\theta}}$ (required for recursive feasibility & stability)
- If $\boldsymbol{\theta}$ is such that $\mathbb{W}_{\boldsymbol{\theta}}$ encloses state dispersion, MPC is safe

Closed-loop stability under some conditions on $\boldsymbol{\theta}$ (not trivial), need $\gamma = 1$ (for now)

Robust MPC - Safety-constrained learning

Robust NMPC parameters $\boldsymbol{\theta}$

Policy gradient

$$\nabla_{\boldsymbol{\theta}} J = \mathbb{E}\left[\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}} \nabla_{\mathrm{u}} A_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}\right]$$

adjusts $\boldsymbol{\theta}$ for performance

Condition

$$\mathbf{s}_{+} - \mathbf{f}\left(\mathbf{s}, \mathbf{a}, \boldsymbol{\theta}\right) \in \mathbb{W}_{\boldsymbol{\theta}}$$

enforces safety through $\boldsymbol{\theta}$

Robust MPC - Safety-constrained learning

Robust NMPC parameters $\boldsymbol{\theta}$

Policy gradient

$$\nabla_{\boldsymbol{\theta}} J = \mathbb{E}\left[\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}} \nabla_{\mathrm{u}} A_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}\right]$$

adjusts $\boldsymbol{\theta}$ for performance

Condition

$$\mathbf{s}_{+} - \mathbf{f}\left(\mathbf{s}, \mathbf{a}, \boldsymbol{\theta}\right) \in \mathbb{W}_{\boldsymbol{\theta}}$$

enforces safety through $\boldsymbol{\theta}$

- No clear connection to SYSID
- Sometimes does opposite of SYSID

Robust MPC - Safety-constrained learning

Robust NMPC parameters $\boldsymbol{\theta}$

Policy gradient

$$\nabla_{\boldsymbol{\theta}} J = \mathbb{E}\left[\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}} \nabla_{\mathrm{u}} A_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}\right]$$

adjusts $\boldsymbol{\theta}$ for performance

- No clear connection to SYSID
- Sometimes does opposite of SYSID

Condition

$$\mathbf{s}_{+} - \mathbf{f}\left(\mathbf{s}, \mathbf{a}, \boldsymbol{\theta}\right) \in \mathbb{W}_{\boldsymbol{\theta}}$$

enforces safety through $\boldsymbol{\theta}$

- Can be interpreted as a form of SYSID (see set-membership)

Robust MPC - Safety-constrained learning

Robust NMPC parameters $\boldsymbol{\theta}$

Policy gradient

$$\nabla_{\boldsymbol{\theta}} J = \mathbb{E}\left[\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}} \nabla_{\mathbf{u}} A_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}\right]$$

adjusts $\boldsymbol{\theta}$ for performance

- No clear connection to SYSID
- Sometimes does opposite of SYSID

Condition

$$\mathbf{s}_{+} - \mathbf{f}\left(\mathbf{s}, \mathbf{a}, \boldsymbol{\theta}\right) \in \mathbb{W}_{\boldsymbol{\theta}}$$

enforces safety through $\boldsymbol{\theta}$

- Can be interpreted as a form of SYSID (see set-membership)

Safe RL?

Classic RL steps: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} J$

Robust MPC - Safety-constrained learning

Robust NMPC parameters $\boldsymbol{\theta}$

Policy gradient

$$\nabla_{\boldsymbol{\theta}} J = \mathbb{E}\left[\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}} \nabla_{\mathbf{u}} A_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}\right]$$

adjusts $\boldsymbol{\theta}$ for performance

- No clear connection to SYSID
- Sometimes does opposite of SYSID

Condition

$$\mathbf{s}_{+} - \mathbf{f}\left(\mathbf{s}, \mathbf{a}, \boldsymbol{\theta}\right) \in \mathbb{W}_{\boldsymbol{\theta}}$$

enforces safety through $\boldsymbol{\theta}$

- Can be interpreted as a form of SYSID (see set-membership)

Safe RL?

Classic RL steps: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} J$

Also reads as:

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \Delta\boldsymbol{\theta}$$

$$\Delta\boldsymbol{\theta} = \arg\min_{\Delta\boldsymbol{\theta}} \frac{1}{2\alpha} \left\|\Delta\boldsymbol{\theta}\right\|^2 + \nabla_{\boldsymbol{\theta}} J^\top \Delta\boldsymbol{\theta}$$

Robust MPC - Safety-constrained learning

Robust NMPC parameters $\boldsymbol{\theta}$

Policy gradient

$$\nabla_{\boldsymbol{\theta}} J = \mathbb{E}\left[\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}} \nabla_{\mathbf{u}} A_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}\right]$$

adjusts $\boldsymbol{\theta}$ for performance

- No clear connection to SYSID
- Sometimes does opposite of SYSID

Condition

$$\mathbf{s}_{+} - \mathbf{f}\left(\mathbf{s}, \mathbf{a}, \boldsymbol{\theta}\right) \in \mathbb{W}_{\boldsymbol{\theta}}$$

enforces safety through $\boldsymbol{\theta}$

- Can be interpreted as a form of SYSID (see set-membership)

Safe RL?

Classic RL steps: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} J$

Also reads as:

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \Delta\boldsymbol{\theta}$$

$$\Delta\boldsymbol{\theta} = \arg\min_{\Delta\boldsymbol{\theta}} \frac{1}{2\alpha} \left\|\Delta\boldsymbol{\theta}\right\|^2 + \nabla_{\boldsymbol{\theta}} J^\top \Delta\boldsymbol{\theta}$$

Safe RL steps $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \Delta\boldsymbol{\theta}$:

$$\begin{aligned} \Delta\boldsymbol{\theta} = \arg\min_{\Delta\boldsymbol{\theta}} \ & \frac{1}{2\alpha} \left\|\Delta\boldsymbol{\theta}\right\|^2 + \nabla_{\boldsymbol{\theta}} J^\top \Delta\boldsymbol{\theta} \\ \text{s.t.} \ & \mathbf{s}_{+} - \mathbf{f}\left(\mathbf{s}, \mathbf{a}, \boldsymbol{\theta} + \Delta\boldsymbol{\theta}\right) \in \mathbb{W}_{\boldsymbol{\theta}+\Delta\boldsymbol{\theta}} \\ & \forall \left(\mathbf{s}, \mathbf{a}, \mathbf{s}_{+}\right) \ \text{in data set} \end{aligned}$$

Robust MPC - Safety-constrained learning

Robust NMPC parameters $\boldsymbol{\theta}$

Policy gradient

$$\nabla_{\boldsymbol{\theta}} J = \mathbb{E}\left[\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}} \nabla_{\mathbf{u}} A_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}\right]$$

adjusts $\boldsymbol{\theta}$ for performance

- No clear connection to SYSID
- Sometimes does opposite of SYSID

Condition

$$\mathbf{s}_{+} - \mathbf{f}\left(\mathbf{s}, \mathbf{a}, \boldsymbol{\theta}\right) \in \mathbb{W}_{\boldsymbol{\theta}}$$

enforces safety through $\boldsymbol{\theta}$

- Can be interpreted as a form of SYSID (see set-membership)

Safe RL?

Classic RL steps: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} J$

Also reads as:

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \Delta\boldsymbol{\theta}$$

$$\Delta\boldsymbol{\theta} = \arg\min_{\Delta\boldsymbol{\theta}} \frac{1}{2\alpha} \left\|\Delta\boldsymbol{\theta}\right\|^2 + \nabla_{\boldsymbol{\theta}} J^\top \Delta\boldsymbol{\theta}$$

Safe RL steps $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \Delta\boldsymbol{\theta}$:

$$\begin{aligned} \Delta\boldsymbol{\theta} = \arg\min_{\Delta\boldsymbol{\theta}} \ & \frac{1}{2\alpha} \left\|\Delta\boldsymbol{\theta}\right\|^2 + \nabla_{\boldsymbol{\theta}} J^\top \Delta\boldsymbol{\theta} \\ \text{s.t.} \ & \mathbf{s}_{+} - \mathbf{f}\left(\mathbf{s}, \mathbf{a}, \boldsymbol{\theta} + \Delta\boldsymbol{\theta}\right) \in \mathbb{W}_{\boldsymbol{\theta}+\Delta\boldsymbol{\theta}} \\ & \forall \left(\mathbf{s}, \mathbf{a}, \mathbf{s}_{+}\right) \ \text{in data set} \end{aligned}$$

Safe RL steps seek performance under safety constraints

Robust MPC - Safety-constrained learning

Robust NMPC parameters $\boldsymbol{\theta}$

Policy gradient

$$\nabla_{\boldsymbol{\theta}} J = \mathbb{E}\left[\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}} \nabla_{\mathbf{u}} A_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}\right]$$

adjusts $\boldsymbol{\theta}$ for performance

- No clear connection to SYSID
- Sometimes does opposite of SYSID

Condition

$$\mathbf{s}_+ - \mathbf{f}\left(\mathbf{s}, \mathbf{a}, \boldsymbol{\theta}\right) \in \mathbb{W}_{\boldsymbol{\theta}}$$

enforces safety through $\boldsymbol{\theta}$

- Can be interpreted as a form of SYSID (see set-membership)

Safe RL?

Classic RL steps: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} J$

Also reads as:

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \Delta\boldsymbol{\theta}$$

$$\Delta\boldsymbol{\theta} = \arg\min_{\Delta\boldsymbol{\theta}} \frac{1}{2\alpha} \|\Delta\boldsymbol{\theta}\|^2 + \nabla_{\boldsymbol{\theta}} J^\top \Delta\boldsymbol{\theta}$$

Safe RL steps $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \Delta\boldsymbol{\theta}$:

$$\Delta\boldsymbol{\theta} = \arg\min_{\Delta\boldsymbol{\theta}} \frac{1}{2\alpha} \|\Delta\boldsymbol{\theta}\|^2 + \nabla_{\boldsymbol{\theta}} J^\top \Delta\boldsymbol{\theta}$$

$$\text{s.t.} \quad \mathbf{s}_+ - \mathbf{f}\left(\mathbf{s}, \mathbf{a}, \boldsymbol{\theta} + \Delta\boldsymbol{\theta}\right) \in \mathbb{W}_{\boldsymbol{\theta}+\Delta\boldsymbol{\theta}}$$

$$\forall \left(\mathbf{s}, \mathbf{a}, \mathbf{s}_+\right) \text{ in data set}$$

Safe RL steps seek performance under safety constraints

Safe Reinforcement Learning Using Robust MPC, Transaction on Automatic Control, 2020
Safe Reinforcement Learning with Stability & Safety Guarantees Using Robust MPC, S.Gros, M. Zanon, Automatica 2021

Outline

Stability of MPC

Policy π_{MPC} from

$$\begin{aligned} \min_{\mathbf{s},\mathbf{a}} \quad & T(\mathbf{s}_N) + \sum_{k=0}^{N-1} L(\mathbf{s}_k, \mathbf{a}_k) \\ \text{s.t.} \quad & \mathbf{s}_{k+1} = \mathbf{f}(\mathbf{s}_k, \mathbf{a}_k) \\ & \mathbf{h}(\mathbf{s}_k, \mathbf{a}_k) \leq 0, \quad \mathbf{s}_N \in \mathbb{T} \end{aligned}$$

Equivalent MPC

$$\begin{aligned} \min_{\mathbf{s},\mathbf{a}} \quad & -\lambda(\mathbf{s}_0) + \tilde{T}(\mathbf{s}_N) + \sum_{k=0}^{N-1} \tilde{L}(\mathbf{s}_k, \mathbf{a}_k) \\ \text{s.t.} \quad & \mathbf{s}_{k+1} = \mathbf{f}(\mathbf{s}_k, \mathbf{a}_k) \\ & \mathbf{h}(\mathbf{s}_k, \mathbf{a}_k) \leq 0, \quad \mathbf{s}_N \in \mathbb{T} \end{aligned}$$

where $\tilde{L}(\mathbf{s}, \mathbf{a}) \geq \kappa(\|\mathbf{s} - \mathbf{s}_{\mathrm{s}}\|), \quad \forall \mathbf{s}, \mathbf{a}$

Stability of MPC

Policy π_{MPC} from

$$\begin{aligned}\min_{\mathbf{s},\mathbf{a}} \quad & T(\mathbf{s}_N) + \sum_{k=0}^{N-1} L(\mathbf{s}_k, \mathbf{a}_k) \\ \mathrm{s.t.} \quad & \mathbf{s}_{k+1} = \mathbf{f}(\mathbf{s}_k, \mathbf{a}_k) \\ & \mathbf{h}(\mathbf{s}_k, \mathbf{a}_k) \leq 0, \quad \mathbf{s}_N \in \mathbb{T}\end{aligned}$$

Equivalent MPC

$$\begin{aligned}\min_{\mathbf{s},\mathbf{a}} \quad & -\lambda(\mathbf{s}_0) + \tilde{T}(\mathbf{s}_N) + \sum_{k=0}^{N-1} \tilde{L}(\mathbf{s}_k, \mathbf{a}_k) \\ \mathrm{s.t.} \quad & \mathbf{s}_{k+1} = \mathbf{f}(\mathbf{s}_k, \mathbf{a}_k) \\ & \mathbf{h}(\mathbf{s}_k, \mathbf{a}_k) \leq 0, \quad \mathbf{s}_N \in \mathbb{T}\end{aligned}$$

where $\tilde{L}(\mathbf{s}, \mathbf{a}) \geq \kappa(\|\mathbf{s} - \mathbf{s}_\mathrm{s}\|), \quad \forall \mathbf{s}, \mathbf{a}$

If for some K_∞ function κ ("bowl-shaped"):

$$L(\mathbf{s}, \mathbf{a}) \geq \kappa(\|\mathbf{s} - \mathbf{s}_\mathrm{s}\|), \quad \forall \mathbf{s}, \mathbf{a}$$

holds, then MPC scheme is stabilizing

Stability of MPC

Policy π_{MPC} from

$$\min_{\mathbf{s},\mathbf{a}} \quad T\left(\mathbf{s}_N\right) + \sum_{k=0}^{N-1} L\left(\mathbf{s}_k, \mathbf{a}_k\right)$$
$$\text{s.t.} \quad \mathbf{s}_{k+1} = \mathbf{f}\left(\mathbf{s}_k, \mathbf{a}_k\right)$$
$$\mathbf{h}\left(\mathbf{s}_k, \mathbf{a}_k\right) \leq 0, \quad \mathbf{s}_N \in \mathbb{T}$$

Equivalent MPC

$$\min_{\mathbf{s},\mathbf{a}} \quad -\lambda\left(\mathbf{s}_0\right) + \tilde{T}\left(\mathbf{s}_N\right) + \sum_{k=0}^{N-1} \tilde{L}\left(\mathbf{s}_k, \mathbf{a}_k\right)$$
$$\text{s.t.} \quad \mathbf{s}_{k+1} = \mathbf{f}\left(\mathbf{s}_k, \mathbf{a}_k\right)$$
$$\mathbf{h}\left(\mathbf{s}_k, \mathbf{a}_k\right) \leq 0, \quad \mathbf{s}_N \in \mathbb{T}$$

where $\tilde{L}\left(\mathbf{s}, \mathbf{a}\right) \geq \kappa\left(\|\mathbf{s} - \mathbf{s}_{\mathrm{s}}\|\right), \quad \forall\, \mathbf{s}, \mathbf{a}$

For generic L (economic), if there is λ such that

$$\tilde{L}\left(\mathbf{s}, \mathbf{a}\right) = L\left(\mathbf{s}, \mathbf{a}\right) + \lambda\left(\mathbf{s}\right) - \lambda\left(\mathbf{f}\left(\mathbf{s}, \mathbf{a}\right)\right) \geq \kappa\left(\|\mathbf{s} - \mathbf{s}_{\mathrm{s}}\|\right), \quad \forall\, \mathbf{s}, \mathbf{a}$$

then MPC scheme is stabilizing

Remarks:

- No discount $\gamma = 1$
- Exact model, deterministic

Stability of MPC

Policy π_{MPC} from

$$\min_{\mathbf{s},\mathbf{a}} \quad T(\mathbf{s}_N) + \sum_{k=0}^{N-1} L(\mathbf{s}_k, \mathbf{a}_k)$$
$$\text{s.t.} \quad \mathbf{s}_{k+1} = \mathbf{f}(\mathbf{s}_k, \mathbf{a}_k)$$
$$\mathbf{h}(\mathbf{s}_k, \mathbf{a}_k) \leq 0, \quad \mathbf{s}_N \in \mathbb{T}$$

Equivalent MPC

$$\min_{\mathbf{s},\mathbf{a}} \quad -\lambda(\mathbf{s}_0) + \tilde{T}(\mathbf{s}_N) + \sum_{k=0}^{N-1} \tilde{L}(\mathbf{s}_k, \mathbf{a}_k)$$
$$\text{s.t.} \quad \mathbf{s}_{k+1} = \mathbf{f}(\mathbf{s}_k, \mathbf{a}_k)$$
$$\mathbf{h}(\mathbf{s}_k, \mathbf{a}_k) \leq 0, \quad \mathbf{s}_N \in \mathbb{T}$$

where $\tilde{L}(\mathbf{s}, \mathbf{a}) \geq \kappa(\|\mathbf{s} - \mathbf{s}_\text{s}\|), \quad \forall \mathbf{s}, \mathbf{a}$

For generic L (economic), if there is λ such that

$$\tilde{L}(\mathbf{s}, \mathbf{a}) = L(\mathbf{s}, \mathbf{a}) + \lambda(\mathbf{s}) - \lambda(\mathbf{f}(\mathbf{s}, \mathbf{a})) \geq \kappa(\|\mathbf{s} - \mathbf{s}_\text{s}\|), \quad \forall \mathbf{s}, \mathbf{a}$$

then MPC scheme is stabilizing

Remarks:

- No discount $\gamma = 1$
- Exact model, deterministic

Theory does not apply to MDPs
Can we extend to $\gamma < 1$ and stochastic dynamics?

Stability of MPC

Policy π_{MPC} from

$$\min_{\mathbf{s},\mathbf{a}} \quad \gamma^N T(\mathbf{s}_N) + \sum_{k=0}^{N-1} \gamma^k L(\mathbf{s}_k, \mathbf{a}_k)$$

$$\text{s.t.} \quad \mathbf{s}_{k+1} = \mathbf{f}(\mathbf{s}_k, \mathbf{a}_k)$$

$$\mathbf{h}(\mathbf{s}_k, \mathbf{a}_k) \leq 0, \quad \mathbf{s}_N \in \mathbb{T}$$

MDP:

$$\min_{\boldsymbol{\pi}} \quad \mathbb{E}_{\boldsymbol{\pi}}\left[\sum_{k=0}^{\infty} \gamma^k L(\mathbf{s}_k, \mathbf{a}_k)\right]$$

where $\mathbf{a}_k = \boldsymbol{\pi}(\mathbf{s}_k)$ and system dynamics

$$\mathbf{s}_{k+1} \sim \mathbb{P}\left[\cdot \,|\, \mathbf{s}_k, \mathbf{a}_k\right]$$

Stability of MPC

Policy π_{MPC} from

$$\begin{aligned} \min_{\mathbf{s},\mathbf{a}} \quad & \gamma^N T(\mathbf{s}_N) + \sum_{k=0}^{N-1} \gamma^k L(\mathbf{s}_k, \mathbf{a}_k) \\ \text{s.t.} \quad & \mathbf{s}_{k+1} = \mathbf{f}(\mathbf{s}_k, \mathbf{a}_k) \\ & \mathbf{h}(\mathbf{s}_k, \mathbf{a}_k) \leq 0, \quad \mathbf{s}_N \in \mathbb{T} \end{aligned}$$

MDP:

$$\min_{\boldsymbol{\pi}} \quad \mathbb{E}_{\boldsymbol{\pi}}\left[\sum_{k=0}^{\infty} \gamma^k L(\mathbf{s}_k, \mathbf{a}_k)\right]$$

where $\mathbf{a}_k = \boldsymbol{\pi}(\mathbf{s}_k)$ and system dynamics

$$\mathbf{s}_{k+1} \sim \mathbb{P}[\,\cdot \,|\, \mathbf{s}_k, \mathbf{a}_k\,]$$

Discounted Strict Dissipativity:

$$L(\mathbf{s},\mathbf{a}) + \lambda(\mathbf{s}) - \gamma\lambda(\mathbf{f}(\mathbf{s},\mathbf{a})) \geq \kappa(\|\mathbf{s} - \mathbf{s}_{\text{s}}\|)$$

Stability of MPC

Policy π_{MPC} from

$$\begin{aligned} \min_{\mathbf{s},\mathbf{a}} \quad & \gamma^N T(\mathbf{s}_N) + \sum_{k=0}^{N-1} \gamma^k L(\mathbf{s}_k, \mathbf{a}_k) \\ \text{s.t.} \quad & \mathbf{s}_{k+1} = \mathbf{f}(\mathbf{s}_k, \mathbf{a}_k) \\ & \mathbf{h}(\mathbf{s}_k, \mathbf{a}_k) \leq 0, \quad \mathbf{s}_N \in \mathbb{T} \end{aligned}$$

MDP:

$$\min_{\boldsymbol{\pi}} \quad \mathbb{E}_{\boldsymbol{\pi}} \left[\sum_{k=0}^{\infty} \gamma^k L(\mathbf{s}_k, \mathbf{a}_k) \right]$$

where $\mathbf{a}_k = \boldsymbol{\pi}(\mathbf{s}_k)$ and system dynamics

$$\mathbf{s}_{k+1} \sim \mathbb{P}[\,\cdot\,|\,\mathbf{s}_k, \mathbf{a}_k\,]$$

Strong Discounted Strict Dissipativity:

$$L(\mathbf{s},\mathbf{a}) + \lambda(\mathbf{s}) - \gamma\lambda(\mathbf{f}(\mathbf{s},\mathbf{a})) \geq \kappa(\|\mathbf{s} - \mathbf{s}_{\text{s}}\|)$$
$$L(\mathbf{s},\mathbf{a}) + \lambda(\mathbf{s}) - \lambda(\mathbf{f}(\mathbf{s},\mathbf{a})) + (\gamma - 1)V_\star^\gamma(\mathbf{f}(\mathbf{s},\mathbf{a})) \geq \kappa(\|\mathbf{s} - \mathbf{s}_{\text{s}}\|)$$

where $V_\star^\gamma$ is the discounted value function of the problem.

Stability of MPC

Policy $\boldsymbol{\pi}_{\mathrm{MPC}}$ from

$$\min_{\mathbf{s},\mathbf{a}} \quad \gamma^N T(\mathbf{s}_N) + \sum_{k=0}^{N-1} \gamma^k L(\mathbf{s}_k, \mathbf{a}_k)$$

$$\text{s.t.} \quad \mathbf{s}_{k+1} = \mathbf{f}(\mathbf{s}_k, \mathbf{a}_k)$$

$$\mathbf{h}(\mathbf{s}_k, \mathbf{a}_k) \leq 0, \quad \mathbf{s}_N \in \mathbb{T}$$

MDP:

$$\min_{\boldsymbol{\pi}} \quad \mathbb{E}_{\boldsymbol{\pi}}\left[\sum_{k=0}^{\infty} \gamma^k L(\mathbf{s}_k, \mathbf{a}_k)\right]$$

where $\mathbf{a}_k = \boldsymbol{\pi}(\mathbf{s}_k)$ and system dynamics

$$\mathbf{s}_{k+1} \sim \mathbb{P}[\,\cdot\,|\,\mathbf{s}_k, \mathbf{a}_k\,]$$

- Classic dissipativity does not readily extend to stochastic systems. E.g.

$$\mathbb{E}\left[L(\mathbf{s},\mathbf{a}) + \lambda(\mathbf{s}) - \lambda(\mathbf{f}(\mathbf{s},\mathbf{a})) \geq \kappa(\|\mathbf{s} - \mathbf{s}_{\mathrm{s}}\|)\right]$$

  does not work...

- Lyapunov arguments do not readily apply to stochastic systems. Why?
  - The classic notion of "steady-state" fails because of the stochasticity
  - Decreasing Lyapunov function does not exist. E.g. for any V convex:

$$\mathbf{s}_+ \sim \mathcal{N}(\mathbf{s}, \Sigma), \qquad \mathbb{E}\left[V(\mathbf{s}_+)\,|\,\mathbf{s}\right] \geq V(\mathbf{s})$$

  - What to do? Work on the state density rather than the state itself!

Stability of MPC

Policy π_{MPC} from

$$\begin{aligned} \min_{\mathbf{s},\mathbf{a}} \quad & \gamma^N T(\mathbf{s}_N) + \sum_{k=0}^{N-1} \gamma^k L(\mathbf{s}_k, \mathbf{a}_k) \\ \text{s.t.} \quad & \mathbf{s}_{k+1} = \mathbf{f}(\mathbf{s}_k, \mathbf{a}_k) \\ & \mathbf{h}(\mathbf{s}_k, \mathbf{a}_k) \leq 0, \quad \mathbf{s}_N \in \mathbb{T} \end{aligned}$$

MDP:

$$\min_{\boldsymbol{\pi}} \quad \mathbb{E}_{\boldsymbol{\pi}} \left[\sum_{k=0}^{\infty} \gamma^k L(\mathbf{s}_k, \mathbf{a}_k) \right]$$

where $\mathbf{a}_k = \boldsymbol{\pi}(\mathbf{s}_k)$ and system dynamics

$$\mathbf{s}_{k+1} \sim \mathbb{P}[\,\cdot\,|\,\mathbf{s}_k, \mathbf{a}_k\,]$$

Functional dissipativity: if there is a functional λ such that:

$$\mathcal{L}[\rho, \boldsymbol{\pi}] - \lambda[\rho_+] + \lambda[\rho] \geq \kappa(D(\rho\,||\,\rho^{\mathrm{s}})), \qquad \mathbf{s} \sim \rho, \; \mathbf{s}_+ \sim \rho_+$$

then the state distribution ρ converges to ρ^{s}

where

- $\mathcal{L}$ is the problem cost functional, e.g. $\mathcal{L} = \mathbb{E}[L(\mathbf{s}, \mathbf{a})]$
- $D(\,\cdot\,||\,\cdot\,)$ is a dissimilarity measure, e.g. Kullback-Liebler Divergence

Stability of MPC

Policy $\boldsymbol{\pi}_{\mathrm{MPC}}$ from

$$\begin{aligned} \min_{\mathbf{s},\mathbf{a}} \quad & \gamma^N T(\mathbf{s}_N) + \sum_{k=0}^{N-1} \gamma^k L(\mathbf{s}_k, \mathbf{a}_k) \\ \mathrm{s.t.} \quad & \mathbf{s}_{k+1} = \mathbf{f}(\mathbf{s}_k, \mathbf{a}_k) \\ & \mathbf{h}(\mathbf{s}_k, \mathbf{a}_k) \leq 0, \quad \mathbf{s}_N \in \mathbb{T} \end{aligned}$$

MDP:

$$\min_{\boldsymbol{\pi}} \quad \mathbb{E}_{\boldsymbol{\pi}}\left[\sum_{k=0}^{\infty} \gamma^k L(\mathbf{s}_k, \mathbf{a}_k)\right]$$

where $\mathbf{a}_k = \boldsymbol{\pi}(\mathbf{s}_k)$ and system dynamics

$$\mathbf{s}_{k+1} \sim \mathbb{P}[\,\cdot\,|\,\mathbf{s}_k, \mathbf{a}_k\,]$$

Functional dissipativity: if there is a functional λ such that:

$$\mathcal{L}[\rho, \boldsymbol{\pi}] - \lambda[\rho_+] + \lambda[\rho] \geq \kappa(D(\rho\,||\,\rho^{\mathrm{s}})), \qquad \mathbf{s} \sim \rho, \;\; \mathbf{s}_+ \sim \rho_+$$

then the state distribution ρ converges to ρ^{s}

where

- $\mathcal{L}$ is the problem cost functional, e.g. $\mathcal{L} = \mathbb{E}[L(\mathbf{s}, \mathbf{a})]$
- $D(\,\cdot\,||\,\cdot\,)$ is a dissimilarity measure, e.g. Kullback-Liebler Divergence

Stability-constrained Learning-based MPC

Goal: given arbitrary stage cost $L(\mathrm{s}, \mathrm{a})$, build a **stable policy** $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ minimizing:

$$J(\boldsymbol{\pi}_{\boldsymbol{\theta}}) = \sum_{k=0}^{\infty} L(\mathbf{s}_k, \mathbf{a}_k)$$

Stability-constrained Learning-based MPC

Goal: given arbitrary stage cost $L(\mathrm{s}, \mathrm{a})$, build a **stable policy** $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ minimizing:

$$J(\boldsymbol{\pi}_{\boldsymbol{\theta}}) = \sum_{k=0}^{\infty} L(\mathbf{s}_k, \mathbf{a}_k)$$

Parametrized policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ from MPC

$$\begin{aligned} \min_{\mathbf{s},\mathbf{a}} \quad & -\lambda_{\boldsymbol{\theta}}(\mathbf{s}_0) + T_{\boldsymbol{\theta}}(\mathbf{s}_N) + \sum_{k=0}^{N-1} L_{\boldsymbol{\theta}}(\mathbf{s}_k, \mathbf{a}_k) \\ \text{s.t.} \quad & \mathbf{s}_{k+1} = \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{s}_k, \mathbf{a}_k) \\ & \mathbf{s}_N \in \mathbb{T} \end{aligned}$$

Stability-constrained Learning-based MPC

Goal: given arbitrary stage cost $L(\mathrm{s},\mathrm{a})$, build a **stable policy** $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ minimizing:

$$J(\boldsymbol{\pi}_{\boldsymbol{\theta}}) = \sum_{k=0}^{\infty} L(\mathbf{s}_k, \mathbf{a}_k)$$

Parametrized policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ from MPC

$$\begin{aligned} \min_{\mathbf{s},\mathbf{a}} \quad & -\lambda_{\boldsymbol{\theta}}(\mathbf{s}_0) + T_{\boldsymbol{\theta}}(\mathbf{s}_N) + \sum_{k=0}^{N-1} L_{\boldsymbol{\theta}}(\mathbf{s}_k, \mathbf{a}_k) \\ \text{s.t.} \quad & \mathbf{s}_{k+1} = \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{s}_k, \mathbf{a}_k) \\ & \mathbf{s}_N \in \mathbb{T} \end{aligned}$$

- Perform undiscounted RL
- Learning based on L
- Impose constraint:

$$L_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{a}) \geq \kappa(\|\mathbf{s} - \mathbf{s}_{\mathrm{s}}\|), \quad \forall\, \mathbf{s}, \mathbf{a}$$

  throughout the learning
- $L_{\boldsymbol{\theta}}$ different than L (stability)
- Term $-\lambda_{\boldsymbol{\theta}}(\mathbf{s}_0)$ is required for MPC to yield the correct Q, V functions

Stability-constrained Learning-based MPC

Goal: given arbitrary stage cost $L(\mathrm{s},\mathrm{a})$, build a **stable policy** $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ minimizing:

$$J(\boldsymbol{\pi}_{\boldsymbol{\theta}}) = \sum_{k=0}^{\infty} L(\mathbf{s}_k, \mathbf{a}_k)$$

- Perform undiscounted RL
- Learning based on L
- Impose constraint:

$$L_{\boldsymbol{\theta}}(\mathrm{s},\mathrm{a}) \geq \kappa(\|\mathrm{s}-\mathrm{s}_{\mathrm{s}}\|), \quad \forall \mathbf{s}, \mathbf{a}$$

  throughout the learning
- $L_{\boldsymbol{\theta}}$ different than L (stability)
- Term $-\lambda_{\boldsymbol{\theta}}(\mathrm{s}_0)$ is required for MPC to yield the correct Q, V functions

Parametrized policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ from MPC

$$\begin{aligned}
\min_{\mathbf{s},\mathbf{a}} \quad & -\lambda_{\boldsymbol{\theta}}(\mathrm{s}_0) + T_{\boldsymbol{\theta}}(\mathbf{s}_N) + \sum_{k=0}^{N-1} L_{\boldsymbol{\theta}}(\mathbf{s}_k, \mathbf{a}_k) \\
\text{s.t.} \quad & \mathbf{s}_{k+1} = \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{s}_k, \mathbf{a}_k) \\
& \mathbf{s}_N \in \mathbb{T}
\end{aligned}$$

Theorem: for a "rich" parametrization:

- $\boldsymbol{\pi}_{\boldsymbol{\theta}} \rightarrow \boldsymbol{\pi}_\star$ if $\boldsymbol{\pi}_\star$ is stabilizing†
- $\boldsymbol{\pi}_{\boldsymbol{\theta}} \rightarrow$ best stabilizing† policy otherwise

Stability-constrained Learning-based MPC

Goal: given arbitrary stage cost $L(\mathrm{s}, \mathrm{a})$, build a **stable policy** $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ minimizing:

$$J(\boldsymbol{\pi}_{\boldsymbol{\theta}}) = \sum_{k=0}^{\infty} L(\mathbf{s}_k, \mathbf{a}_k)$$

- Perform undiscounted RL
- Learning based on L
- Impose constraint:

$$L_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{a}) \geq \kappa(\|\mathbf{s} - \mathbf{s}_{\mathrm{s}}\|), \quad \forall\, \mathbf{s}, \mathbf{a}$$

  throughout the learning
- $L_{\boldsymbol{\theta}}$ different than L (stability)
- Term $-\lambda_{\boldsymbol{\theta}}(\mathbf{s}_0)$ is required for MPC to yield the correct Q, V functions

Parametrized policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ from MPC

$$\begin{aligned} \min_{\mathbf{s},\mathbf{a}} \quad & -\lambda_{\boldsymbol{\theta}}(\mathbf{s}_0) + T_{\boldsymbol{\theta}}(\mathbf{s}_N) + \sum_{k=0}^{N-1} L_{\boldsymbol{\theta}}(\mathbf{s}_k, \mathbf{a}_k) \\ \text{s.t.} \quad & \mathbf{s}_{k+1} = \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{s}_k, \mathbf{a}_k) \\ & \mathbf{s}_N \in \mathbb{T} \end{aligned}$$

Theorem: for a "rich" parametrization:

- $\boldsymbol{\pi}_{\boldsymbol{\theta}} \rightarrow \boldsymbol{\pi}_{\star}$ if $\boldsymbol{\pi}_{\star}$ is stabilizing†
- $\boldsymbol{\pi}_{\boldsymbol{\theta}} \rightarrow$ best stabilizing† policy otherwise

†We are talking about nominal stability here...

Stability-constrained Learning-based MPC

Goal: given arbitrary stage cost $L(\mathrm{s},\mathrm{a})$, build a **stable policy** $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ minimizing:

$$J(\boldsymbol{\pi}_{\boldsymbol{\theta}}) = \sum_{k=0}^{\infty} L(\mathbf{s}_k, \mathbf{a}_k)$$

Constraint

$$L_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{a}) \geq \kappa(\|\mathbf{s}-\mathbf{s}_{\mathrm{s}}\|), \quad \forall\, \mathbf{s}$$

is semi-infinite programming... **not trivial**

Some solutions:

- Sum-of-Squares (SOS) prog.
- Convex representation of $L_{\boldsymbol{\theta}}$
- Something else?

Parametrized policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ from MPC

$$\begin{aligned} \min_{\mathbf{s},\mathbf{a}} \quad & -\lambda_{\boldsymbol{\theta}}(\mathbf{s}_0) + T_{\boldsymbol{\theta}}(\mathbf{s}_N) + \sum_{k=0}^{N-1} L_{\boldsymbol{\theta}}(\mathbf{s}_k, \mathbf{a}_k) \\ \text{s.t.} \quad & \mathbf{s}_{k+1} = \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{s}_k, \mathbf{a}_k) \\ & \mathbf{s}_N \in \mathbb{T} \end{aligned}$$

Theorem: for a "rich" parametrization:

- $\boldsymbol{\pi}_{\boldsymbol{\theta}} \rightarrow \boldsymbol{\pi}_\star$ if $\boldsymbol{\pi}_\star$ is stabilizing†
- $\boldsymbol{\pi}_{\boldsymbol{\theta}} \rightarrow$ best stabilizing† policy otherwise

†We are talking about nominal stability here...

Change of philosophy from "classic" dissipativity theory. Stable design rather than stability analysis.

Stability-constrained Learning-based MPC

Goal: given arbitrary stage cost $L(\mathrm{s},\mathrm{a})$, build a **stable policy** $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ minimizing:

$$J(\boldsymbol{\pi}_{\boldsymbol{\theta}}) = \sum_{k=0}^{\infty} L(\mathrm{s}_k, \mathbf{a}_k)$$

Extension to stable policy for MDPs?

- Build argument from robust MPC? Weak results...
- Hopefully the new dissipativity theory will help us!

Parametrized policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ from MPC

$$\begin{aligned} \min_{\mathbf{s},\mathbf{a}} \quad & -\lambda_{\boldsymbol{\theta}}(\mathrm{s}_0) + T_{\boldsymbol{\theta}}(\mathrm{s}_N) + \sum_{k=0}^{N-1} L_{\boldsymbol{\theta}}(\mathrm{s}_k, \mathbf{a}_k) \\ \text{s.t.} \quad & \mathrm{s}_{k+1} = \mathbf{f}_{\boldsymbol{\theta}}(\mathrm{s}_k, \mathbf{a}_k) \\ & \mathrm{s}_N \in \mathbb{T} \end{aligned}$$

Theorem: for a "rich" parametrization:

- $\boldsymbol{\pi}_{\boldsymbol{\theta}} \rightarrow \boldsymbol{\pi}_\star$ if $\boldsymbol{\pi}_\star$ is stabilizing†
- $\boldsymbol{\pi}_{\boldsymbol{\theta}} \rightarrow$ best stabilizing† policy otherwise

†We are talking about nominal stability here...

Change of philosophy from "classic" dissipativity theory. Stable design rather than stability analysis.

Stability-constrained Learning-based MPC

Goal: given arbitrary stage cost $L(\mathbf{s}, \mathbf{a})$, build a **stable policy** $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ minimizing:

$$J(\boldsymbol{\pi}_{\boldsymbol{\theta}}) = \sum_{k=0}^{\infty} L(\mathbf{s}_k, \mathbf{a}_k)$$

Parametrized policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ from MPC

$$\begin{aligned} \min_{\mathbf{s},\mathbf{a}} \quad & -\lambda_{\boldsymbol{\theta}}(\mathbf{s}_0) + T_{\boldsymbol{\theta}}(\mathbf{s}_N) + \sum_{k=0}^{N-1} L_{\boldsymbol{\theta}}(\mathbf{s}_k, \mathbf{a}_k) \\ \text{s.t.} \quad & \mathbf{s}_{k+1} = \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{s}_k, \mathbf{a}_k) \\ & \mathbf{s}_N \in \mathbb{T} \end{aligned}$$

Extension to stable policy for MDPs?

- Build argument from robust MPC? Weak results...
- Hopefully the new dissipativity theory will help us!

Theorem: for a "rich" parametrization:

- $\boldsymbol{\pi}_{\boldsymbol{\theta}} \rightarrow \boldsymbol{\pi}_\star$ if $\boldsymbol{\pi}_\star$ is stabilizing†
- $\boldsymbol{\pi}_{\boldsymbol{\theta}} \rightarrow$ best stabilizing† policy otherwise

†We are talking about nominal stability here...

Change of philosophy from "classic" dissipativity theory. Stable design rather than stability analysis.

Stability-Constrained Markov Decision Processes Using MPC, Automatica 2021

Outline

1. Safe RL via MPC
2. Safe RL via Robust MPC
3. Stability-constrained Learning with MPC
4. Some more results (in brief)

RL & SYSID with MPC

RL & SYSID are doing two different things (closed-loop performance vs. model fitting). Can they cohabit though?

RL & SYSID with MPC

RL & SYSID are doing two different things (closed-loop performance vs. model fitting). Can they cohabit though?

- In the safe RL context, SYSID handles model uncertainties (set membership) and RL handles performance

RL & SYSID with MPC

RL & SYSID are doing two different things (closed-loop performance vs. model fitting). Can they cohabit though?

- In the safe RL context, SYSID handles model uncertainties (set membership) and RL handles performance
- A more direct combination is meaningful:
  - ✓ Can create an algorithm that tunes the MPC model for fitting **and** the MPC scheme for closed-loop performance at the same time. There can be a "conflict" though.
  - ✓ RL supersedes SYSID, can be implemented via null-space approaches in Q-learning
  - ✓ Extension to policy gradient understood, some technical difficulties, to be published...

RL & SYSID with MPC

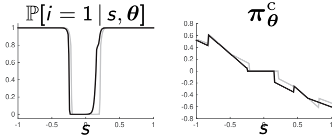
RL & SYSID are doing two different things (closed-loop performance vs. model fitting). Can they cohabit though?

- In the safe RL context, SYSID handles model uncertainties (set membership) and RL handles performance
- A more direct combination is meaningful:
  - ✓ Can create an algorithm that tunes the MPC model for fitting **and** the MPC scheme for closed-loop performance at the same time. There can be a "conflict" though.
  - ✓ RL supersedes SYSID, can be implemented via null-space approaches in Q-learning
  - ✓ Extension to policy gradient understood, some technical difficulties, to be published...

Combining system identification with reinforcement learning-based MPC, A. B. Martinsen, A. M. Lekkas, S. Gros, IFAC 2020

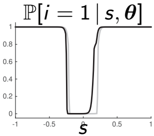
RL & Mixed integer problem in MPC

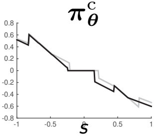
Mixed-integer problems are common.
Mixed-integer MPC schemes are expensive but realistic . Can we combine them to RL as well?



RL & Mixed integer problem in MPC

Mixed-integer problems are common.
Mixed-integer MPC schemes are expensive but realistic . Can we combine them to RL as well?

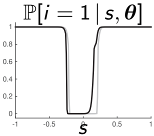


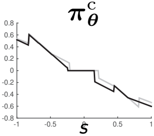


- With Q-learning, fairly trivial... incorrect if no exploration, though

RL & Mixed integer problem in MPC

Mixed-integer problems are common.
Mixed-integer MPC schemes are expensive but realistic . Can we combine them to RL as well?



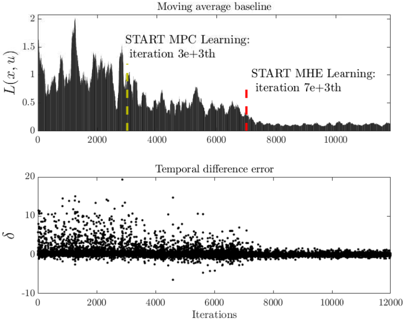


- With Q-learning, fairly trivial... incorrect if no exploration, though
- For policy gradient, devil is in the details
  - ✓ Integer inputs best treated via stochastic policy approach, continuous ones via deterministic policy
  - ✓ Propose a hybrid policy gradient method combining deterministic and stochastic policies, with corresponding compatible linear A_{π_θ} approximations
  - ✓ Works well on mixed-integer MPC examples

Reinforcement Learning for mixed-integer problems based on MPC, S. Gros, M. Zanon, IFAC 2020

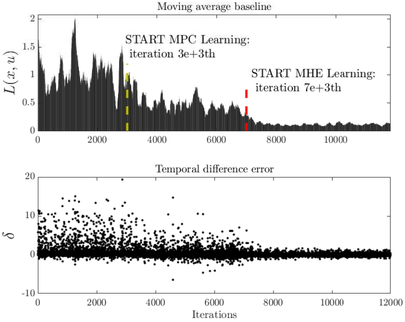
RL & MHE-MPC

The full state of the system is often not available, or not even modelled, use observer (e.g. MHE). Can we still do RL and how?



RL & MHE-MPC

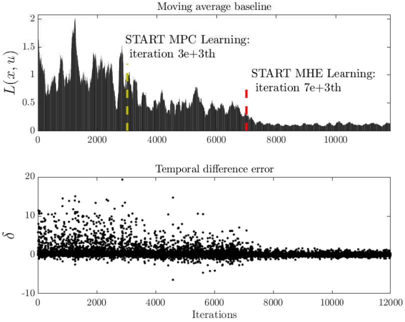
The full state of the system is often not available, or not even modelled, use observer (e.g. MHE). Can we still do RL and how?



- Problem becomes POMDP when MPC model does not include all states

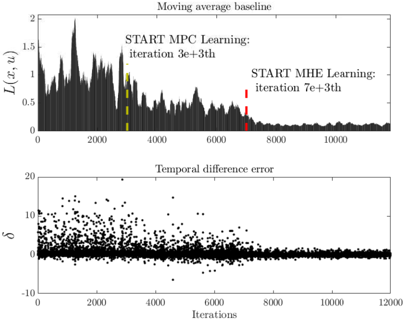
RL & MHE-MPC

The full state of the system is often not available, or not even modelled, use observer (e.g. MHE). Can we still do RL and how?



- Problem becomes POMDP when MPC model does not include all states
- MHE is an intrinsic component of the policy, must be treated in RL as well
  - ✓ Propose an RL scheme that tunes MHE and MPC jointly for closed loop performance in the context of Q learning
  - ✓ Algorithmic is simple, performances on simple example are very promising
  - ✓ The MHE tuning has a strong impact on performance (on our examples), better than model fitting
  - ✓ Extension to policy gradient understood, to be published
  - ✓ Works also if MPC model omits some of the real states

RL & MHE-MPC



The full state of the system is often not available, or not even modelled, use observer (e.g. MHE). Can we still do RL and how?

- Problem becomes POMDP when MPC model does not include all states
- MHE is an intrinsic component of the policy, must be treated in RL as well
  - ✓ Propose an RL scheme that tunes MHE and MPC jointly for closed loop performance in the context of Q learning
  - ✓ Algorithmic is simple, performances on simple example are very promising
  - ✓ The MHE tuning has a strong impact on performance (on our examples), better than model fitting
  - ✓ Extension to policy gradient understood, to be published
  - ✓ Works also if MPC model omits some of the real states

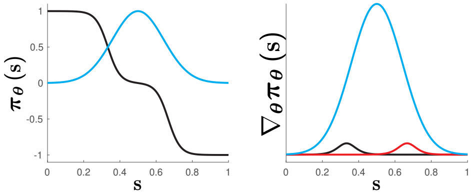
Reinforcement Learning based on MPC/MHE for Unmodeled and Partially Observable Dynamics, H.N. Esfahani, S. Gros, ACC 2021

RL & MPC for “strongly economic” problems

Some policies are dominated by “switches”, difficult to treat in RL because $\nabla_{\theta}\pi_{\theta} = 0$ on most of the state space. Hence

$$\nabla_{\theta} J(\pi_{\theta}) = \mathbb{E}\left[\nabla_{\theta}\pi_{\theta}\nabla_{\mathrm{u}}A_{\pi_{\theta}}\right]$$

is based on the contribution from a very small number of samples. Parameter updates become “infrequent and jumpy”.

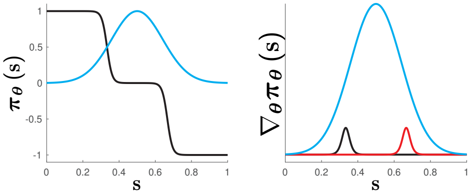


RL & MPC for “strongly economic” problems

Some policies are dominated by “switches”, difficult to treat in RL because $\nabla_{\theta}\pi_{\theta} = 0$ on most of the state space. Hence

$$\nabla_{\theta} J(\pi_{\theta}) = \mathbb{E}\left[\nabla_{\theta}\pi_{\theta}\nabla_{\mathrm{u}}A_{\pi_{\theta}}\right]$$

is based on the contribution from a very small number of samples. Parameter updates become “infrequent and jumpy”.

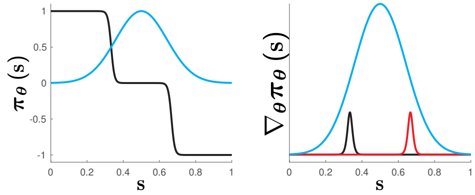


RL & MPC for "strongly economic" problems

Some policies are dominated by "switches", difficult to treat in RL because $\nabla_{\theta}\pi_{\theta} = 0$ on most of the state space. Hence

$$\nabla_{\theta} J(\pi_{\theta}) = \mathbb{E}\left[\nabla_{\theta}\pi_{\theta}\nabla_{\mathrm{u}}A_{\pi_{\theta}}\right]$$

is based on the contribution from a very small number of samples. Parameter updates become "infrequent and jumpy".

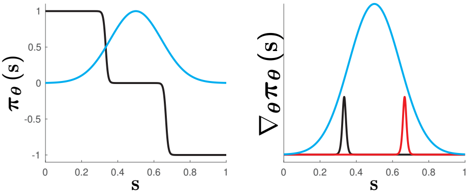


RL & MPC for "strongly economic" problems

Some policies are dominated by "switches", difficult to treat in RL because $\nabla_\theta \pi_\theta = 0$ on most of the state space. Hence

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}\left[\nabla_\theta \pi_\theta \nabla_{\mathrm{u}} A_{\pi_\theta}\right]$$

is based on the contribution from a very small number of samples. Parameter updates become "infrequent and jumpy".

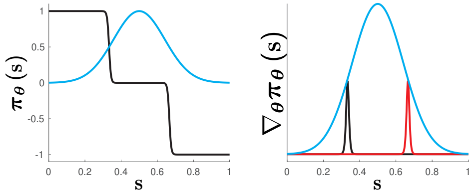


RL & MPC for "strongly economic" problems

Some policies are dominated by "switches", difficult to treat in RL because $\nabla_\theta \pi_\theta = 0$ on most of the state space. Hence

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}\left[\nabla_\theta \pi_\theta \nabla_{\mathrm{u}} A_{\pi_\theta}\right]$$

is based on the contribution from a very small number of samples. Parameter updates become "infrequent and jumpy".

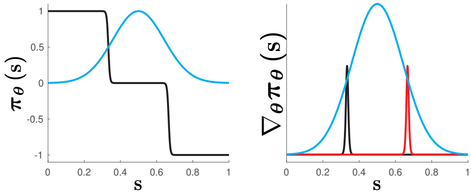


RL & MPC for "strongly economic" problems

Some policies are dominated by "switches", difficult to treat in RL because $\nabla_\theta \pi_\theta = 0$ on most of the state space. Hence

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}\left[\nabla_\theta \pi_\theta \nabla_{\mathrm{u}} A_{\pi_\theta}\right]$$

is based on the contribution from a very small number of samples. Parameter updates become "infrequent and jumpy".

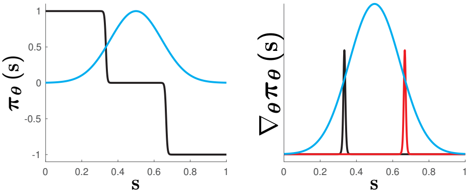


RL & MPC for “strongly economic” problems

Some policies are dominated by “switches”, difficult to treat in RL because $\nabla_\theta \pi_\theta = 0$ on most of the state space. Hence

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}\left[\nabla_\theta \pi_\theta \nabla_{\mathrm{u}} A_{\pi_\theta}\right]$$

is based on the contribution from a very small number of samples. Parameter updates become “infrequent and jumpy”.

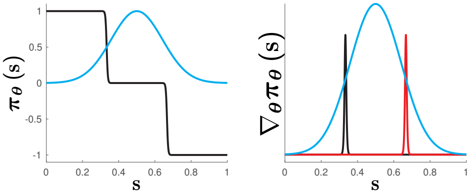


RL & MPC for “strongly economic” problems

Some policies are dominated by “switches”, difficult to treat in RL because $\nabla_\theta \pi_\theta = 0$ on most of the state space. Hence

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}\left[\nabla_\theta \pi_\theta \nabla_{\mathrm{u}} A_{\pi_\theta}\right]$$

is based on the contribution from a very small number of samples. Parameter updates become “infrequent and jumpy”.

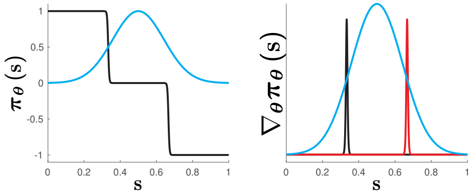


RL & MPC for "strongly economic" problems

Some policies are dominated by "switches", difficult to treat in RL because $\nabla_\theta \pi_\theta = 0$ on most of the state space. Hence

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}\left[\nabla_\theta \pi_\theta \nabla_{\mathrm{u}} A_{\pi_\theta}\right]$$

is based on the contribution from a very small number of samples. Parameter updates become "infrequent and jumpy".

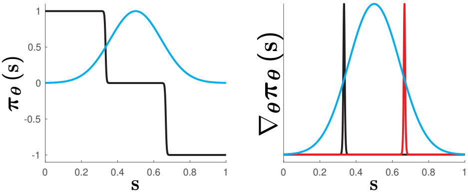


RL & MPC for "strongly economic" problems

Some policies are dominated by "switches", difficult to treat in RL because $\nabla_\theta \pi_\theta = 0$ on most of the state space. Hence

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}\left[\nabla_\theta \pi_\theta \nabla_{\mathrm{u}} A_{\pi_\theta}\right]$$

is based on the contribution from a very small number of samples. Parameter updates become "infrequent and jumpy".



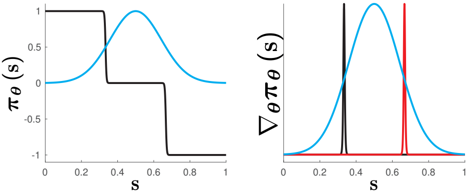
RL & MPC for “strongly economic” problems

Some policies are dominated by “switches”, difficult to treat in RL because $\nabla_\theta \pi_\theta = 0$ on most of the state space. Hence

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}\left[\nabla_\theta \pi_\theta \nabla_{\mathrm{u}} A_{\pi_\theta}\right]$$

is based on the contribution from a very small number of samples. Parameter updates become “infrequent and jumpy”.

- ✓ Proposed policy relaxation techniques based on Interior-Point formulations, such that $\nabla_\theta \pi_\theta \neq 0$ almost everywhere
- ✓ Converge the policy to the true one over the learning



MPC-based Reinforcement Learning for Economic Problems with Application to Battery Storage, A. Kordabad, W. Cay, S. Gros, ECC 2021

Tuning of the MPC "meta"-parameters

MPC "meta"-parameters:

- Horizon length N
- When to recompute control sequence (event-based MPC)

MPC:

$$\min_{\mathbf{s},\mathbf{a}} \quad T(\mathbf{s}_N) + \sum_{k=0}^{N-1} L(\mathbf{s}_k, \mathbf{a}_k)$$

$$\text{s.t.} \quad \mathbf{s}_{k+1} = \mathbf{f}(\mathbf{s}_k, \mathbf{a}_k)$$

$$\mathbf{h}(\mathbf{s}_k, \mathbf{a}_k) \leq 0$$

yields $\boldsymbol{\pi}_{\mathrm{MPC}}(\mathbf{s}_0) = \mathbf{a}_0^\star$

Event-triggered:

- apply input profile $\mathbf{a}^\star_{0,\dots,n}$ until re-computation is triggered
- often used to reduce computational demand, energy, etc.

Tuning of the MPC "meta"-parameters

MPC "meta"-parameters:

- Horizon length N
- When to recompute control sequence (event-based MPC)

MPC:

$$\min_{\mathbf{s},\mathbf{a}} \quad T(\mathbf{s}_N) + \sum_{k=0}^{N-1} L(\mathbf{s}_k, \mathbf{a}_k)$$

$$\text{s.t.} \quad \mathbf{s}_{k+1} = \mathbf{f}(\mathbf{s}_k, \mathbf{a}_k)$$

$$\mathbf{h}(\mathbf{s}_k, \mathbf{a}_k) \leq 0$$

yields $\boldsymbol{\pi}_{\mathrm{MPC}}(\mathbf{s}_0) = \mathbf{a}_0^\star$

Event-triggered:

- apply input profile $\mathbf{a}^\star_{0,\ldots,n}$ until re-computation is triggered
- often used to reduce computational demand, energy, etc.

Fairly simple idea, requires some care to be treated correctly:

- ✓ Define augmented state to preserve Markov property (essential for RL methods)
- ✓ Stochastic policy gradient methods required, must define the densities very carefully

Tuning of the MPC "meta"-parameters

MPC "meta"-parameters:

- Horizon length N
- When to recompute control sequence (event-based MPC)

MPC:

$$\min_{\mathbf{s},\mathbf{a}} \quad T(\mathbf{s}_N) + \sum_{k=0}^{N-1} L(\mathbf{s}_k, \mathbf{a}_k)$$

$$\text{s.t.} \quad \mathbf{s}_{k+1} = \mathbf{f}(\mathbf{s}_k, \mathbf{a}_k)$$

$$\mathbf{h}(\mathbf{s}_k, \mathbf{a}_k) \leq 0$$

yields $\boldsymbol{\pi}_{\mathrm{MPC}}(\mathbf{s}_0) = \mathbf{a}_0^\star$

Event-triggered:

- apply input profile $\mathbf{a}^\star_{0,\ldots,n}$ until re-computation is triggered
- often used to reduce computational demand, energy, etc.

Fairly simple idea, requires some care to be treated correctly:

✓ Define augmented state to preserve Markov property (essential for RL methods)

✓ Stochastic policy gradient methods required, must define the densities very carefully

Optimization of the Model Predictive Control Update Interval Using Reinforcement Learning, E. BÃ¸hn, S. Gros, S. Moe, T.A. Johansen, MICNON, 2021

RL to evaluate the storage function

Policy π_{MPC} from

$$\min_{\mathbf{s},\mathbf{a}} \quad T(\mathbf{s}_N) + \sum_{k=0}^{N-1} L(\mathbf{s}_k, \mathbf{a}_k)$$

$$\text{s.t.} \quad \mathbf{s}_{k+1} = \mathbf{f}(\mathbf{s}_k, \mathbf{a}_k)$$

$$\mathbf{h}(\mathbf{s}_k, \mathbf{a}_k) \leq 0, \quad \mathbf{s}_N \in \mathbb{T}$$

If for some λ function:

$$L(\mathbf{s}, \mathbf{a}) + \lambda(\mathbf{s}) - \lambda(\mathbf{f}(\mathbf{s}, \mathbf{a})) \geq \kappa(\|\mathbf{s} - \mathbf{s}_\mathrm{s}\|), \quad \forall\, \mathbf{s}, \mathbf{a}$$

holds, then MPC scheme is stabilizing

How to evaluate λ?

- Approximate $\mathbf{f}$ as a polynomial, then Sum-of-Squares technique can be used
- We propose: parametrize λ and evaluate it via Q-learning

To finish

Some bibliography

Optimization of the MPC "meta-parameters" (horizon, sampling, event-triggered)

1. Optimization of the Model Predictive Control Update Interval Using Reinforcement Learning, LDCC, 2021
2. Reinforcement Learning of the Prediction Horizon in Model Predictive Control, NMPC 2021

Safe RL via Robust MPC

3. Safe Reinforcement Learning Using Robust MPC, TAC, 2020
4. Approximate Robust NMPC using Reinforcement Learning, ECC2021
5. Reinforcement Learning based on Scenario-tree MPC for ASVs, S. Gros, ACC 2021
6. Safe Reinforcement Learning via projection on a safe set: how to achieve optimality? IFAC 2020

Stable Learning using MPC

7. Stability-Constrained Markov Decision Processes Using MPC, Automatica, 2021
8. Safe Reinforcement Learning with Stability & Safety Guarantees Using Robust MPC, S.Gros, M. Zanon, TAC, 2021
9. A Dissipativity Theory for Undiscounted Markov Decision Processes, Automatica, 2021
10. A New Dissipativity Condition for Asymptotic Stability of Discounted Economic MPC, Automatica, 2021
11. Verification of Dissipativity and Evaluation of Storage Function in Economic NMPC using Q-Learning, NMPC 2021

Policy gradient methods for MPC

12. Bias Correction in RL via the Deterministic Policy Gradient Method for MPC-Based Policies, ECC 2021
13. Reinforcement Learning based on MPC and the Stochastic Policy Gradient Method, ACC 2021
14. Bias Correction in Deterministic Policy Gradient Using Robust MPC, ACC 2021

RL for mixed-integer MPC

15. Reinforcement Learning for mixed-integer problems with MPC-based function approximation, IFAC 2020

RL-MPC and SYSID

16. Combining system identification with reinforcement learning-based MPC, IFAC 2020

RL-MPC and State Estimation

17. Reinforcement Learning based on MPC/MHE for Unmodeled and Partially Observable Dynamics, ACC 2021

Wild cards

18. MPC-based Reinforcement Learning for Economic Problems with Application to Battery Storage, ECC 2021
19. Reinforcement Learning Based on Real-Time Iteration NMPC, ECC 2021