Mode-constrained Model-based Reinforcement Learning via Gaussian Processes

Aidan Scannell[1] Carl Henrik Ek[2] Arthur Richards[3]

[1]Aalto University, [2]University of Cambridge, [3]University of Bristol

TL;DR

- We present ModeRL, a model-based RL algorithm constrained to a single dynamic mode.
- This is a difficult problem because the mode constraint is a hidden variable associated with the environment's dynamics.
- Our probabilistic dynamic model infers the mode constraint alongside the underlying dynamic modes.
- ModeRL leverages the model's well-calibrated uncertainty to:
  - Enforce the mode constraint up to a given probability during training,
  - Escape local optima induced by the constraint, see Figure 1.
- We validate ModeRL in a simulated quadcopter navigation task.

Problem Statement

- **Dynamics**: In a given state $\mathbf{s}_t \in \mathcal{S} \subseteq \mathbb{R}^{D_x}$, one of K dynamic modes $f = \{f_k : \mathcal{S}_k \times \mathcal{A} \to \mathcal{S}\}_{k=1}^{K}$ (and associated noise models $\boldsymbol{\epsilon}_k$) governs the system, as indicated by $\alpha : \mathcal{S} \to \{1, \dots, K\}$:

$$\mathbf{s}_{t+1} = f_k(\mathbf{s}_t, \mathbf{a}_t) + \boldsymbol{\epsilon}_{k,t}, \quad \text{if } \alpha(\mathbf{s}_t) = k. \tag{1}$$

- **Goal**: Find policy π that maximises sum of rewards in expectation over transition noise $J(\pi, f) = \mathbb{E}_{\boldsymbol{\epsilon}_{0:T}}\left[\sum_{t=0}^{T} r(\mathbf{s}_t, \mathbf{a}_t) \mid \mathbf{s}_0\right]$, whilst remaining in the desired dynamic mode's k^* state domain $\mathcal{S}_{k^*} = \{\mathbf{s} \in \mathcal{S} \mid \alpha(\mathbf{s}) = k^*\}$:

$$\pi^* = \arg\max_{\pi \in \Pi} J(\pi, f) \quad \text{s.t.} \quad \underbrace{\alpha(\mathbf{s}_t) = k^* \quad \forall t \in \{0, \dots, T\}}_{\text{mode constraint}}, \tag{2}$$

- PROBLEM: can't satisfy constraint during training!
  - So relax to being mode-constrained with high probability...

ModeRL

Model Learning

- **Goal** Jointly infer mode constraint alongside dynamic modes:
  - Data set of state-action inputs and state diff outputs $\mathcal{D} = \{\hat{\mathbf{s}}_t, \Delta\mathbf{s}_{t+1}\}_{t=1}^{N} = (\hat{\mathbf{S}}, \Delta\mathbf{S})$,
  - Formulate prior such that ModeRL can potentially never violate the constraint,
  - Disentangle sources of uncertainty in the mode constraint.
- **Dynamic modes** GP prior over each dynamic mode,
  - Each mode should be assigned a subset of the state-action inputs $\hat{\mathbf{S}}_k \subseteq \hat{\mathbf{S}}$,

$$f_k(\hat{\mathbf{S}}_k) \mid \boldsymbol{\alpha} \sim \mathcal{N}\left(\mu_k(\hat{\mathbf{S}}_k), k_k(\hat{\mathbf{S}}_k, \hat{\mathbf{S}}_k)\right), \qquad \hat{\mathbf{S}}_k = \{\hat{\mathbf{s}}_t \in \hat{\mathbf{S}} \mid \alpha(\mathbf{s}_t) = k\}$$

  - But we sidestep the assignment of observations to modes by augmenting each mode with its own inducing points $\boldsymbol{\zeta}_k \in \mathcal{S} \times \mathcal{A}$,

$$f_k(\boldsymbol{\zeta}_k) \sim \mathcal{N}\left(\mu_k(\boldsymbol{\zeta}_k), k_k(\boldsymbol{\zeta}_k, \boldsymbol{\zeta}_k)\right) \qquad q(f_k(\boldsymbol{\zeta}_k)) = \mathcal{N}\left(f_k(\boldsymbol{\zeta}_k) \mid \mathbf{m}_k, \mathbf{L}_k\mathbf{L}_k^T\right)$$

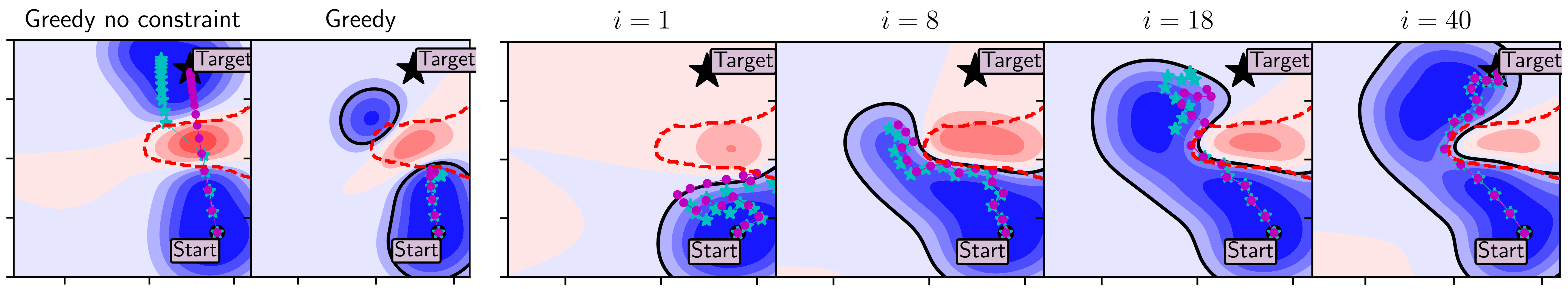

Figure 1. **Mode-constrained quadcopter navigation**. The goal is to navigate to the black star without entering the turbulent dynamic mode (·····). Left plot shows that without our δ-mode-constraint the greedy strategy fails to remain in the desired mode. Second left plot shows that without our exploration term it gets stuck in a local optimum. Right four plots show iterations of ModeRL, which successfully navigates to the target with constraint satisfaction during training. The δ-mode-constraint (▬) expands at each episode i.

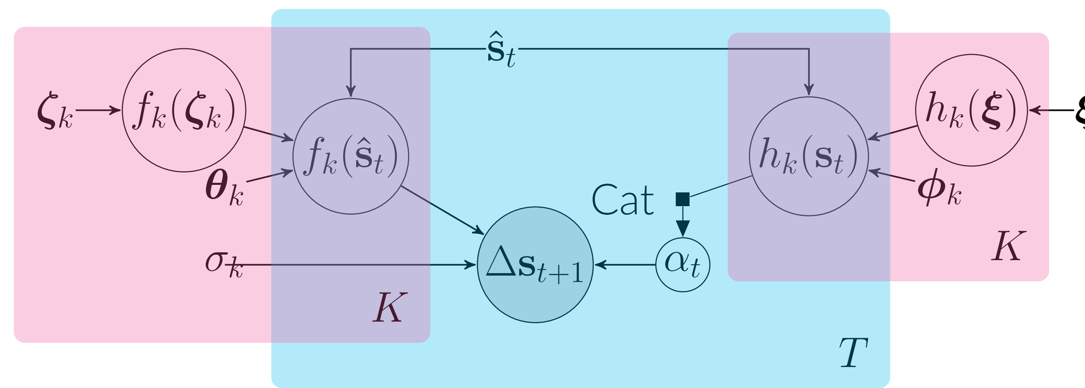

Figure 2. Dynamic model's augmented joint probability space.

- **Mode constraint** Model as GP classifier,
  - i.e. classification likelihood parameterised by K functions $\mathbf{h} = \{h_k : \mathcal{S} \to \mathbb{R}\}_{k=1}^{K}$ with GP priors:

$$\alpha_t \mid \mathbf{s}_t, \mathbf{h}(\mathbf{s}_t) \sim \text{softmax}_k(\mathbf{h}(\mathbf{s}_t)) \qquad h_k(\mathbf{S}) \sim \mathcal{N}\left(\hat{\mu}_k(\mathbf{S}), \hat{k}_k(\mathbf{S}, \mathbf{S})\right) \tag{3}$$

  - Augment with inducing points $\mathcal{N}(h_k(\boldsymbol{\xi}) \mid \hat{\mathbf{m}}_k, \hat{\mathbf{L}}_k\hat{\mathbf{L}}_k^T)$
- **Variational inference** Optimise variational params $\{\mathbf{m}_k, \hat{\mathbf{m}}_k, \mathbf{L}_k, \hat{\mathbf{L}}_k\}_{k=1}^{K}$, inducing inputs $\{\boldsymbol{\zeta}_k\}_{k=1}^{K}$, $\boldsymbol{\xi}$ and GP hyperparams/noise using ELBO.

Planning

Open-loop trajectory optimisation:

$$\arg\max_{\mathbf{a}_0} \max_{\mathbf{a}_1, \dots, \mathbf{a}_{T-1}} \underbrace{\mathbb{E}_{p(f_{k^*} \mid \mathcal{D}_{0:i})}\left[J(\pi, f_{k^*})\right]}_{\text{greedy exploitation}} + \beta \underbrace{\mathcal{H}\left[h_{k^*}(\mathbf{s}_{0:T})\right]}_{\text{exploration}} \tag{4a}$$

$$\text{s.t.} \quad \underbrace{\Pr\left(\alpha_t = k^* \mid \mathbf{s}_0, \mathbf{a}_{0:t}, \mathcal{D}_{0:i}\right) \geq 1 - \delta}_{\delta\text{-mode constraint}} \quad \forall t \in \{0, \dots, T\}, \tag{4b}$$

- **Greedy exploitation** Expected objective under dynamic's posterior,
  - **Multi-step predictions** Assume always in desired dynamic mode,
    - So we can approximate multi-step predictions using moment-matching,
    - For quadratic reward funcs $\mathbb{E}_{p(f_{k^*} \mid \mathcal{D}_{0:i})}\left[J(\pi, f_{k^*})\right]$ has closed form.
- **Exploration** Entropy of mode constraint's GP posterior,
  - Needed to escape local optima induced by constraint.
- **δ-mode constraint** Enforces constraint up to a given probability.
  - For two dynamic modes the δ-mode constraint has closed form.

Experiments

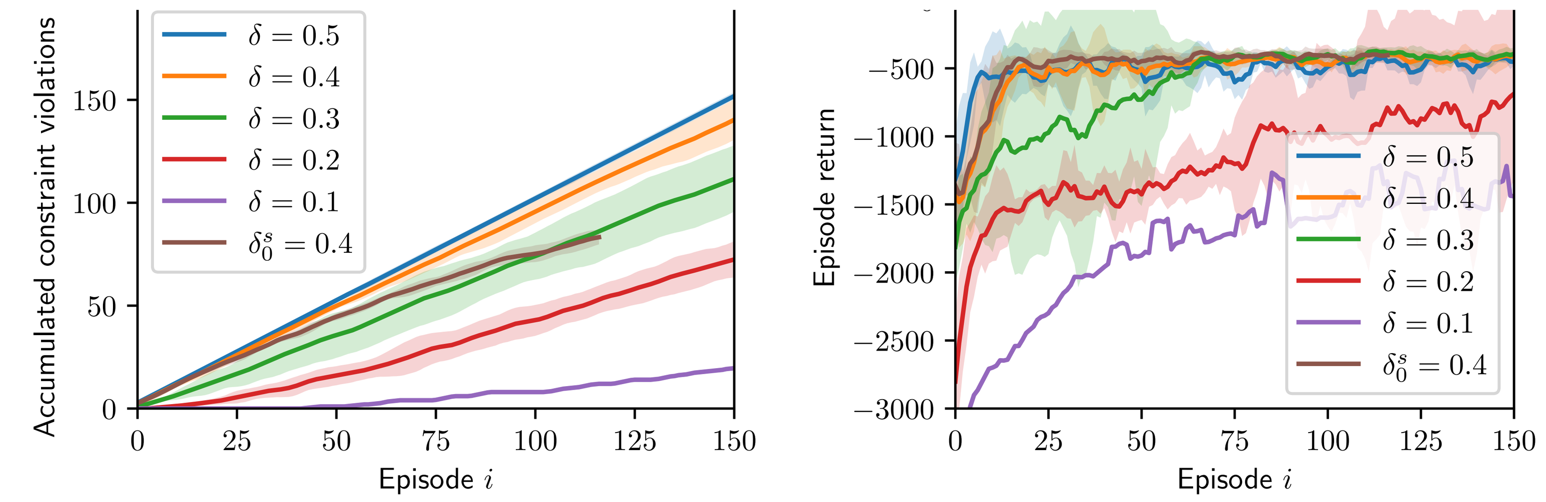

Figure 3. **Constraint level ablation** Left shows that tighter constraints (lower δ) results in fewer constraint violations. However, training curves (right) show that if constraint is too tight (i.e. $\delta \leq 0.2$) then ModeRL gets stuck in local optima. $\delta_0^s = 0.4$ (▬) converged in fewer episodes by using an exponentially decaying schedule from $\delta = 0.4$ at episode $i = 0$.

- **Greedy exploitation (baseline) fails** - Fig. 1 left
  - Without our mode constraint the greedy strategy violates the mode constraint.
  - Without our exploration term the greedy strategy gets stuck in a local optimum.
- **ModeRL works!** - Fig. 1 right
- **ModeRL has constraint satisfaction during training** - Fig. 3
  - Tightening constraint (lower δ) leads to less constraint violations during training,
  - But if too tight (i.e. $\delta \leq 0.2$) it can make the problem infeasible,
  - **How to set δ?** A schedule works well in practice, see $\delta_0^s = 0.4$ (▬) in Fig. 3.

Outlook

- ModeRL must violate the mode constraint in order to learn it,
  - Can we use external sensors to infer constraint without violating it?
- We use an open-loop policy,
  - Can we improve speed so that we can get a closed-loop policy via MPC?
- Code available @ `https://github.com/aidanscannell/moderl`

aidan.scannell@aalto.fi