Learning Stable Dynamics via Iterative Quadratic Programming

Paul Gesel and Momotaz Begum

pac48@wildcats.unh.edu, mbegum@cs.unh.edu

University of New Hampshire

Cognitive Assistive Robotics Lab

Introduction

- LSQ-IQP is a novel autonomous dynamic system (ADS) based controller for trajectory learning from demonstration (LfD)
- Two stages:
  1. Learn an energy function from demonstrations
  2. Learn an ADS that is subjected to a set of stability criteria derived from the energy function.
- We employ semi-infinite quadratic programming to enforce the energy function to have only one minimum

Background

- We can encode demonstrated trajectories into an autonomous dynamical system

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x})$$

- Robust against both spatial and temporal perturbations
- Existing methods: SEDS, CLF-DM, FSM-DS

Proposed Approach

Stability Conditions

$$\begin{aligned} V(\mathbf{0}) &= 0 \\ \forall_{\boldsymbol{x}\in\mathcal{X}\setminus\{\mathbf{0}\}} V(\boldsymbol{x}) &> 0 \\ \forall_{\boldsymbol{x}\in\mathcal{X}\setminus\{\mathbf{0}\}} \exists_{\{\boldsymbol{x}_0\in\mathcal{X}\,:\,0<\|\boldsymbol{x}-\boldsymbol{x}_0\|<\epsilon\}} V(\boldsymbol{x}_0) - V(\boldsymbol{x}) &< 0 \\ \forall_{\boldsymbol{x}\in\mathcal{X}\setminus\{\mathbf{0}\}} \|\nabla V(\boldsymbol{x})\| > 0 \implies \nabla V(\boldsymbol{x}) \cdot f(\boldsymbol{x}) &< 0 \\ \forall_{\boldsymbol{x}\in\mathcal{O}} \nabla V(\boldsymbol{x}) \cdot N(\boldsymbol{x}) &> 0 \\ \forall_{\boldsymbol{x}\in\mathcal{O}} f(\boldsymbol{x}) \cdot N(\boldsymbol{x}) &> 0 \end{aligned}$$

Energy Function Learning

$$V(\boldsymbol{w^l}, \boldsymbol{x}) = \sum_{i}^{b^l} \boldsymbol{w^l}_i \phi_i^l(\boldsymbol{x})$$

$$J^{lp}(\boldsymbol{w^l}) = \sum_{\boldsymbol{x}\in\mathcal{X}} \left(\nabla V(\boldsymbol{w^l}, \boldsymbol{x}) \cdot \frac{near(\boldsymbol{x}) - \boldsymbol{x}}{\|near(\boldsymbol{x}) - \boldsymbol{x}\|} - \beta \right)^2$$

$$J^{ld}(\boldsymbol{w^l}) = \sum_{n=1}^{N} \sum_{t=0}^{T^n} \left(\nabla V(\boldsymbol{w^l}, \boldsymbol{\xi^{t,n}}) \cdot \frac{\dot{\boldsymbol{\xi}}^{\boldsymbol{t,n}}}{\|\dot{\boldsymbol{\xi}}^{\boldsymbol{t,n}}\|} - 1 \right)^2$$

Controller Learning

$$f(\boldsymbol{w^c}, w^v, \boldsymbol{x}) = \sum_{i}^{b^c} \boldsymbol{w^c}_i \nabla \phi_i^c(\boldsymbol{x}) + w^v \nabla V(\boldsymbol{x})$$

$$J^{cp}(\boldsymbol{w^c}, w^v) = \sum_{\boldsymbol{x}\in\mathcal{X}} \|f(\boldsymbol{w^c}, w^v, \boldsymbol{x}) - g(\boldsymbol{x})\|^2$$

$$J^{cd}(\boldsymbol{x}) = \sum_{n=1}^{N} \sum_{t=0}^{T^n} \left\| f(\boldsymbol{\xi^{t,n}}) - \dot{\boldsymbol{\xi}}^{\boldsymbol{t,n}} \right\|^2$$

Constraint Generation

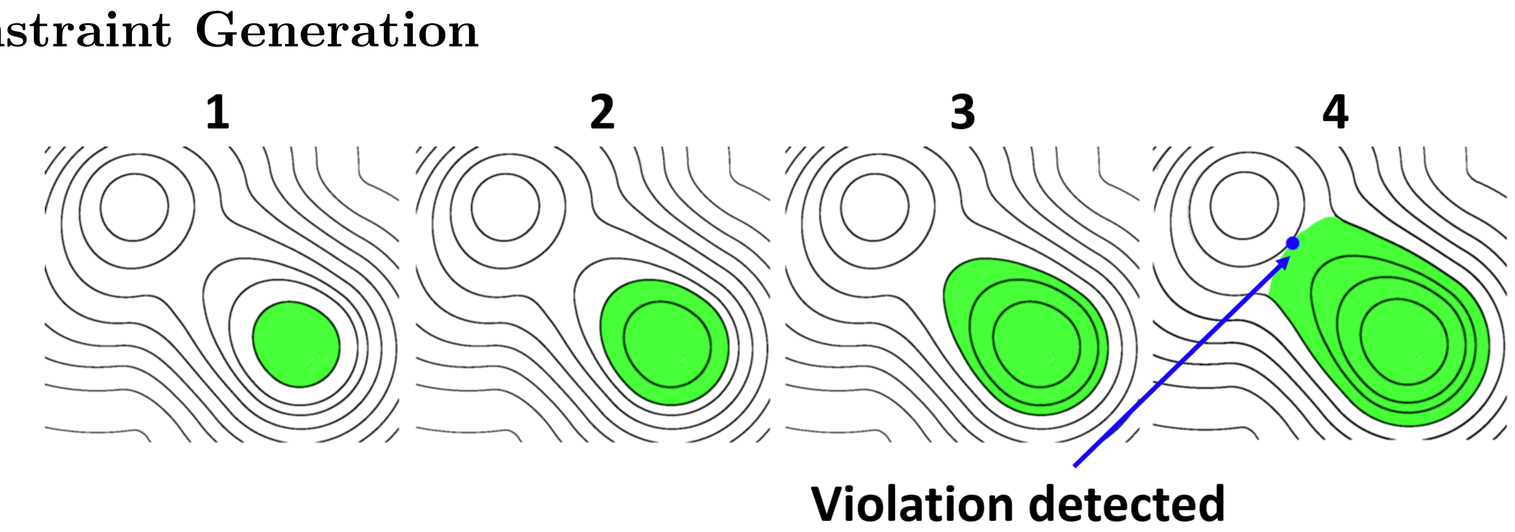

Figure 1: Four steps of constraint violation detection are shown. Steps 1-3 show an energy-first expansion from the origin. Step 4 is the first time a constraint violation is detected.

Dimension Coupling

$$\boldsymbol{y} = h(\boldsymbol{x}, \boldsymbol{g})$$

Streamline Characteristics

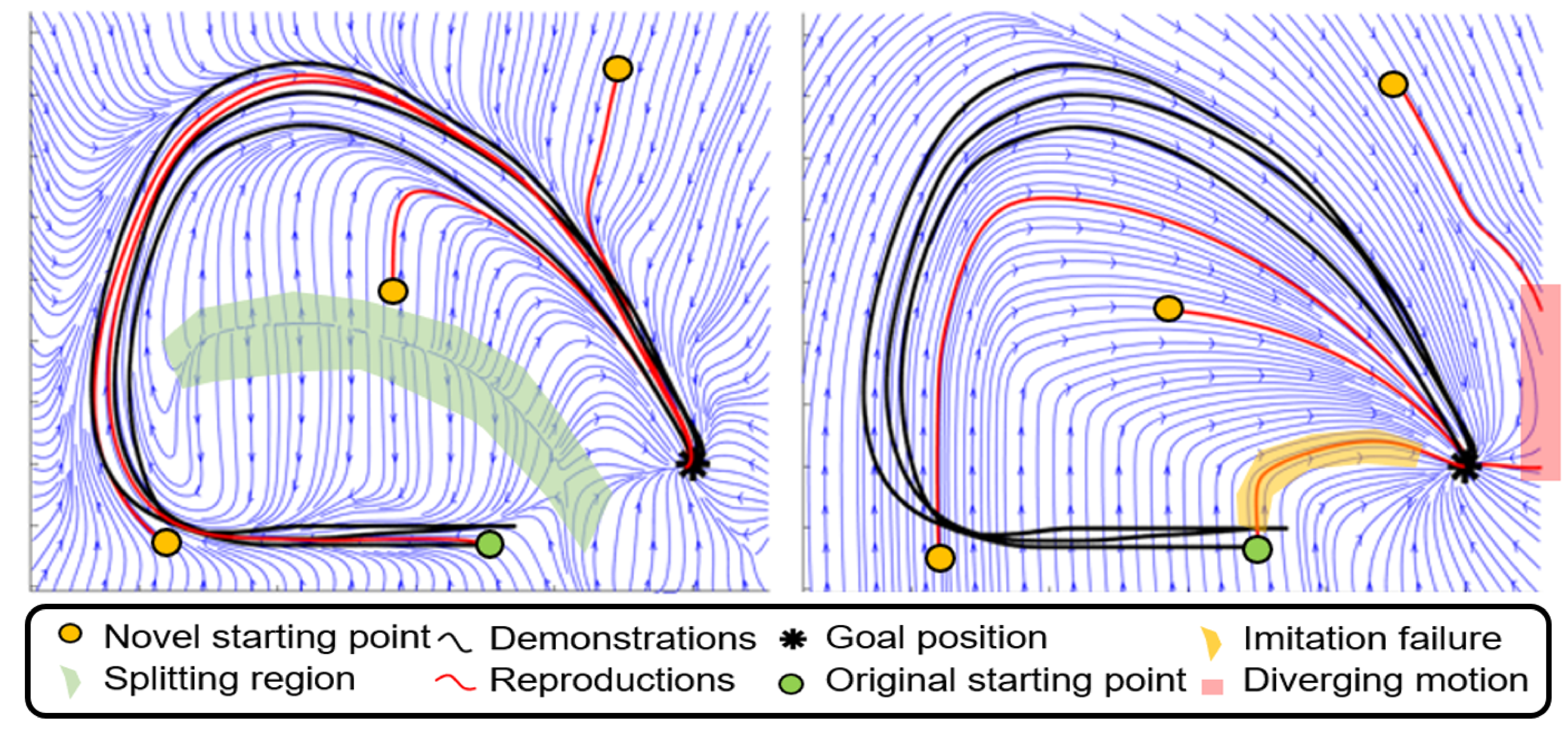

Figure 2: Comparison of streamline plots for LSD-IQP (left) and SEDS (right). LSD-IQP shows stable and accurate reproduction under spatial perturbation.

Experimental Results

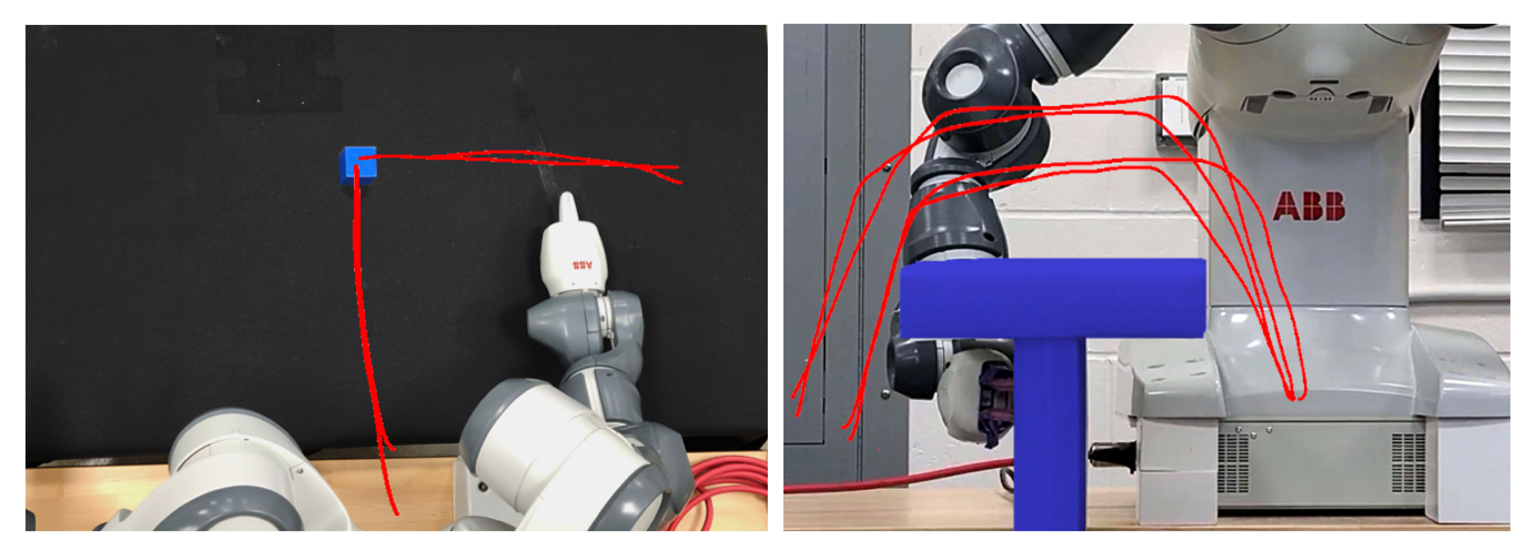

Figure 3: Demonstrations for pick and place task (left) and point to point task (right).

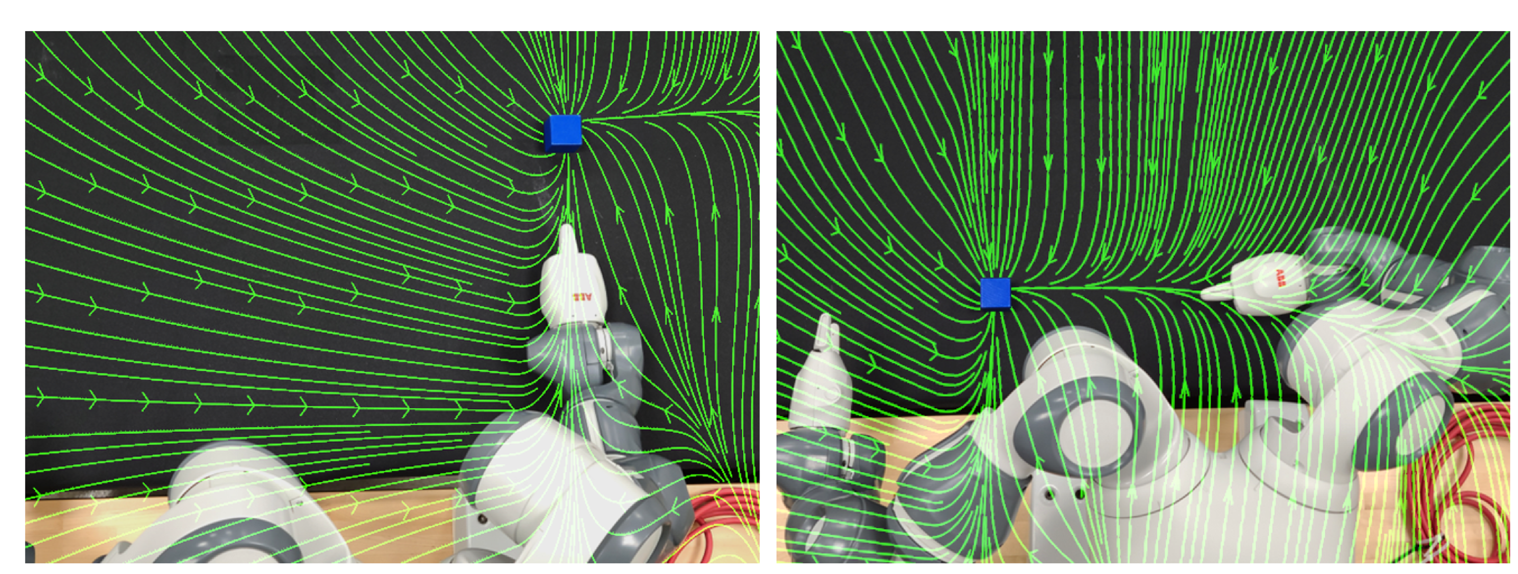

Figure 4: Streamline plots of LSD-IQP for the pick and place task for two different goal positions.

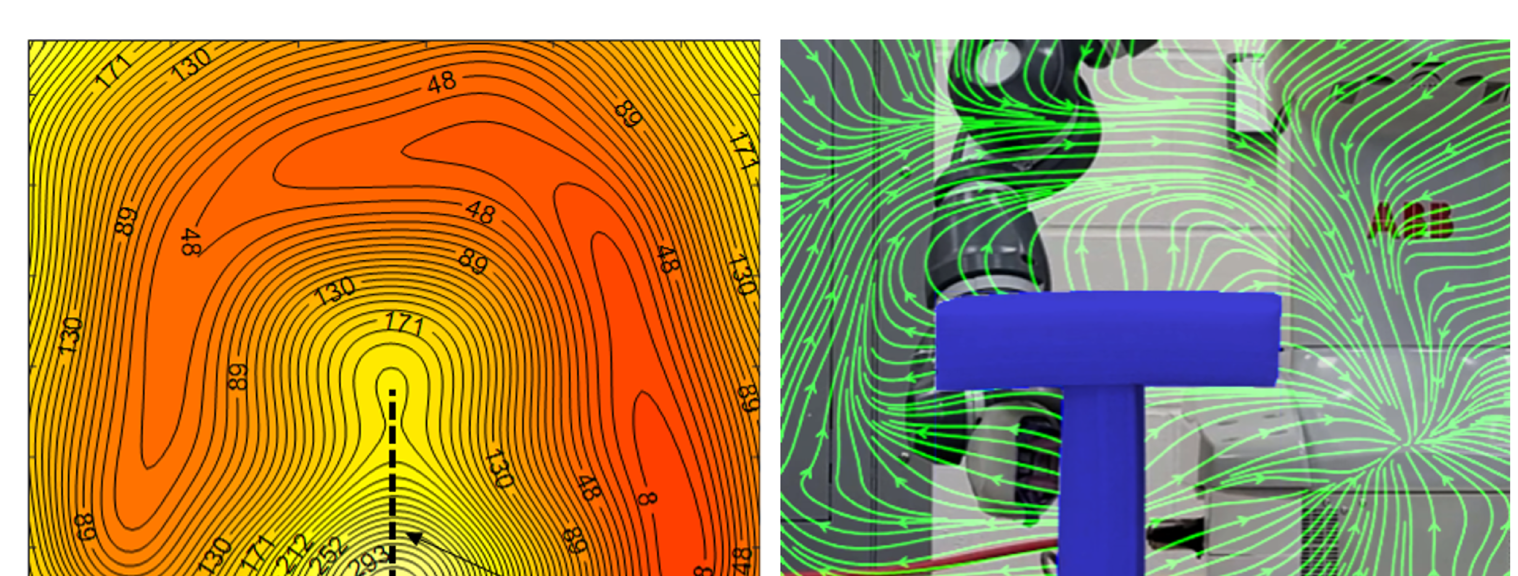

Figure 5: Energy function learned via LSD-IQP (left) and streamline plot for the learned controller.

Conclusion

- We employed iterative quadratic programming with constraint generation to learn an energy function and ADS from demonstrations.
- We validated LSD-IQP in three experiments and showed increased accuracy and capability compared to other ADS methods.

Acknowledgements

Research support
IIS 1830597