TRAJECTORY INVERSE KINEMATICS BY CONDITIONAL DENSITY MODES

Chao Qin and Miguel Á. Carreira-Perpiñán

EECS, School of Engineering, University of California, Merced, USA

1 Abstract

We present a machine learning approach for trajectory inverse kinematics: to find a feasible trajectory in angle space that produces a given trajectory in workspace. The method learns offline a conditional density model of the joint angles given the workspace coordinates. At run time, given a trajectory in the workspace, the method (1) computes the modes of the conditional density given each of the workspace points, and (2) finds the reconstructed angle trajectory by minimising over the set of modes a global, trajectory-wide constraint. We demonstrate the method with a PUMA 560 robot arm and show how it can reconstruct the true angle trajectory even when the workspace trajectory contains singularities, and when the number of inverse branches depends on the workspace location.

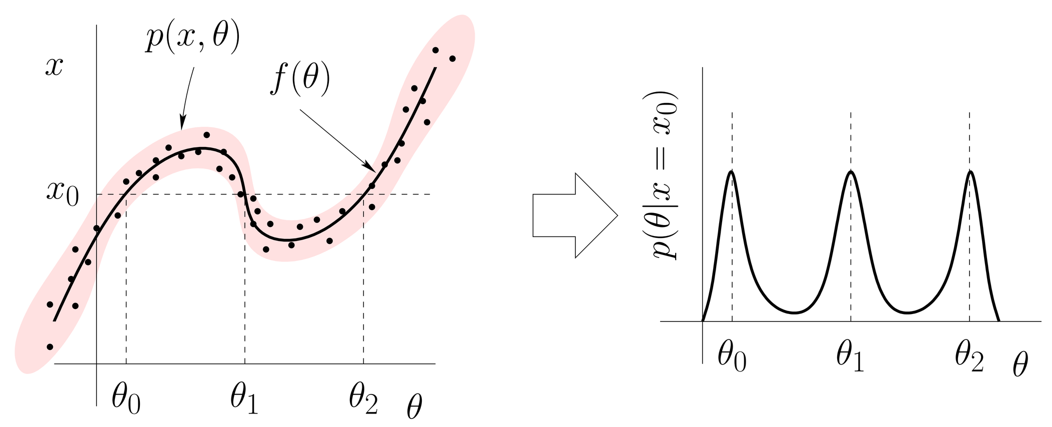
2 Problem statement

- $\mathbf{x}$: positions in Cartesian workspace of the end-effector
- $\boldsymbol{\theta}$: joint angles
- $\mathbf{f}: \boldsymbol{\theta} \rightarrow \mathbf{x}$ forward kinematics
- Pointwise inverse kinematics (IK): $\boldsymbol{\theta} = \mathbf{f}^{-1}(\mathbf{x})$
- Trajectory IK: Given an $\mathbf{x}$-trajectory, to obtain a feasible $\boldsymbol{\theta}$-trajectory that produces the $\mathbf{x}$-trajectory
- Difficulties:
  - Multivalued inverse mapping $\mathbf{f}^{-1}(\mathbf{x})$ (e.g. elbow up; elbow down)
  - $\boldsymbol{\theta}$-trajectory must be globally feasible, e.g. avoiding discontinuities or forbidden regions
- Traditional methods:
  - Analytical method
  - Pseudoinverse: $\mathbf{x} = \mathbf{f}(\boldsymbol{\theta}) \rightarrow \dot{\mathbf{x}} = \mathbf{J}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}$
  - Global method by variational approaches
  - Data driven methods

3 Trajectory inverse kinematics by conditional density modes

Idea of the method:

1. *Offline*, we estimate a density model $p(\boldsymbol{\theta}, \mathbf{x})$ for both variables, or just a conditional density $p(\boldsymbol{\theta}|\mathbf{x})$, using a training set.
2. At *run time*, for each $n = 1, \dots, N$ we obtain the conditional density $p(\boldsymbol{\theta}|\mathbf{x}_n)$ and its modes.
3. For $n = 1, \dots, N$, we obtain the $\boldsymbol{\theta}$-trajectory by minimising a constraint over the entire set of modes.



❶ Density models

- Given a training set of pairs $(\boldsymbol{\theta}_n, \mathbf{x}_n)$, estimate conditional density model $p(\boldsymbol{\theta}|\mathbf{x})$ by:
  - Learning the full density $p(\boldsymbol{\theta}, \mathbf{x})$. We test: Generative Topographic Mapping (GTM)
  - Learning directly $p(\boldsymbol{\theta}|\mathbf{x})$. We test: Mixture Density Networks (MDN)

  Both represent the density with a Gaussian mixture (GM) with M components
- Advantages:
  - Represent inverses by modes from the conditional density
  - Deal with topological changes naturally (mode merging)

❷ Mode finding

- Find all modes of conditional density $p(\boldsymbol{\theta}|\mathbf{x}_n)$ by *Gaussian mean-shift (GMS)*, which starts from every centroid of the GM and iterates $\boldsymbol{\theta}^{(\tau+1)} = \sum_{m=1}^{M} p(m|\boldsymbol{\theta}^{(\tau)})\boldsymbol{\theta}^m$

❸ Global optimisation with dynamic programming

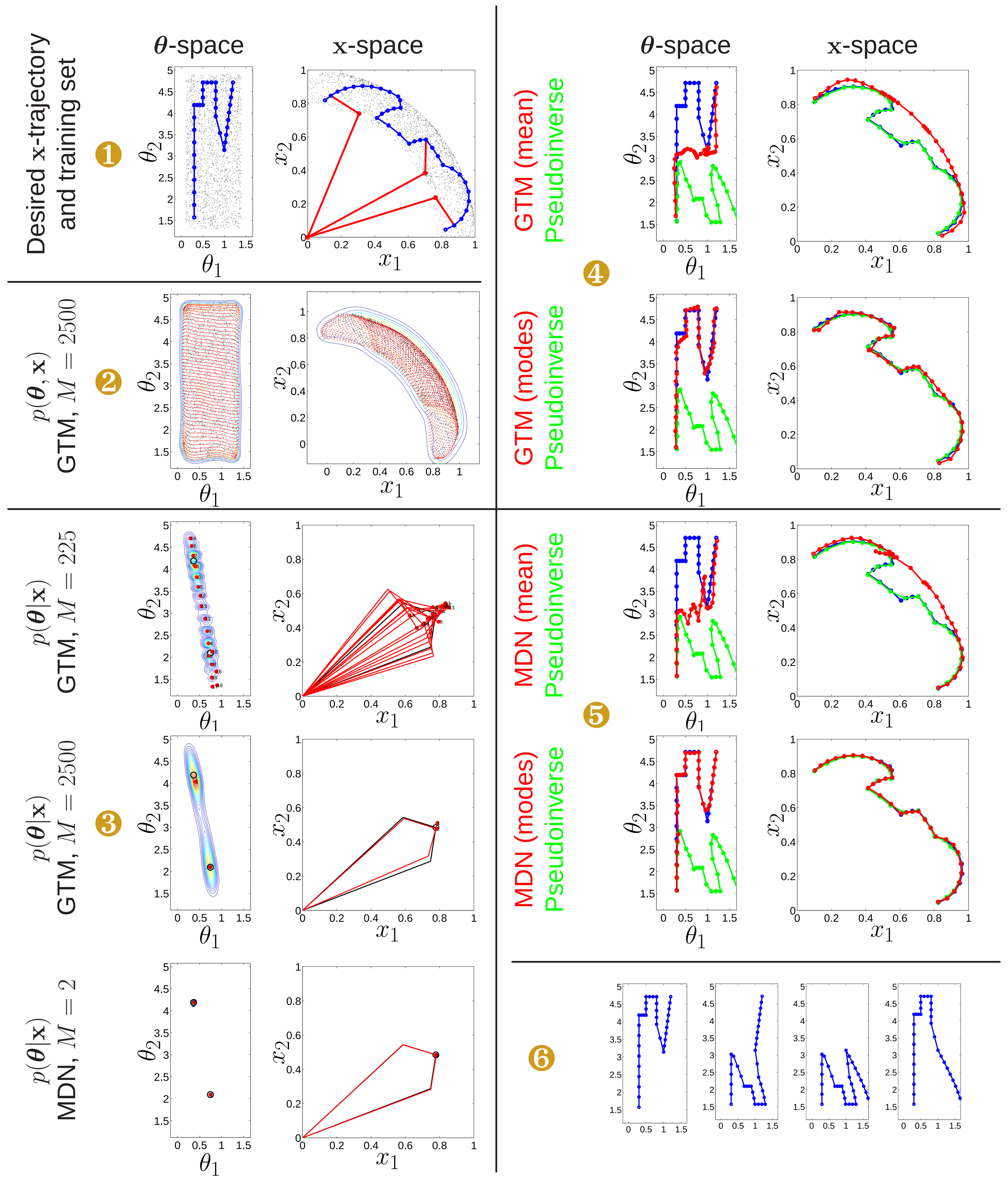
- Obtain a unique $\boldsymbol{\theta}$-trajectory by minimising $\mathcal{C} + \lambda\mathcal{F}$ over the set of modes with dynamic programming ($\lambda \geq 0$)
  - $\mathcal{C} = \sum_{n=1}^{N-1} \|\boldsymbol{\theta}_{n+1} - \boldsymbol{\theta}_n\|$: *continuity constraint* (integrated 1st derivative), penalises sudden angle changes
  - $\mathcal{F} = \sum_{n=1}^{N} \|\mathbf{x}_n - \mathbf{f}(\boldsymbol{\theta}_n)\|$: *forward constraint* (integrated workspace error), penalises spurious inverses

Computational complexity

(k: average number of GMS iterations; ν: average number of modes at each step n)

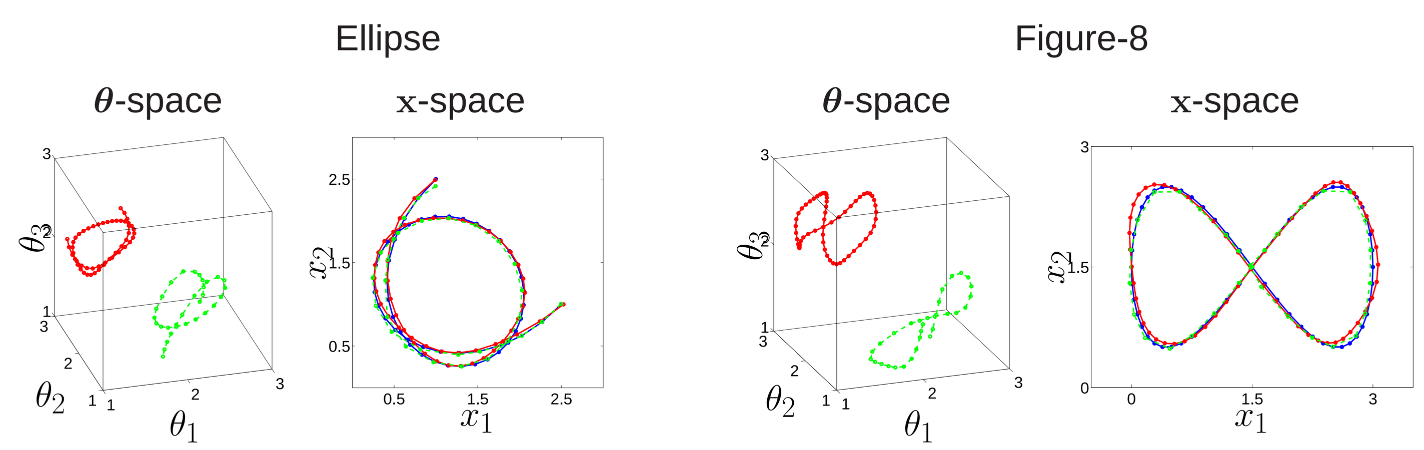
Density models	Mode finding	Global optimisation
Offline	$\mathcal{O}(kNM^2)$	$\mathcal{O}(N\nu^2)$

4 Experiments: planar 2-link arm



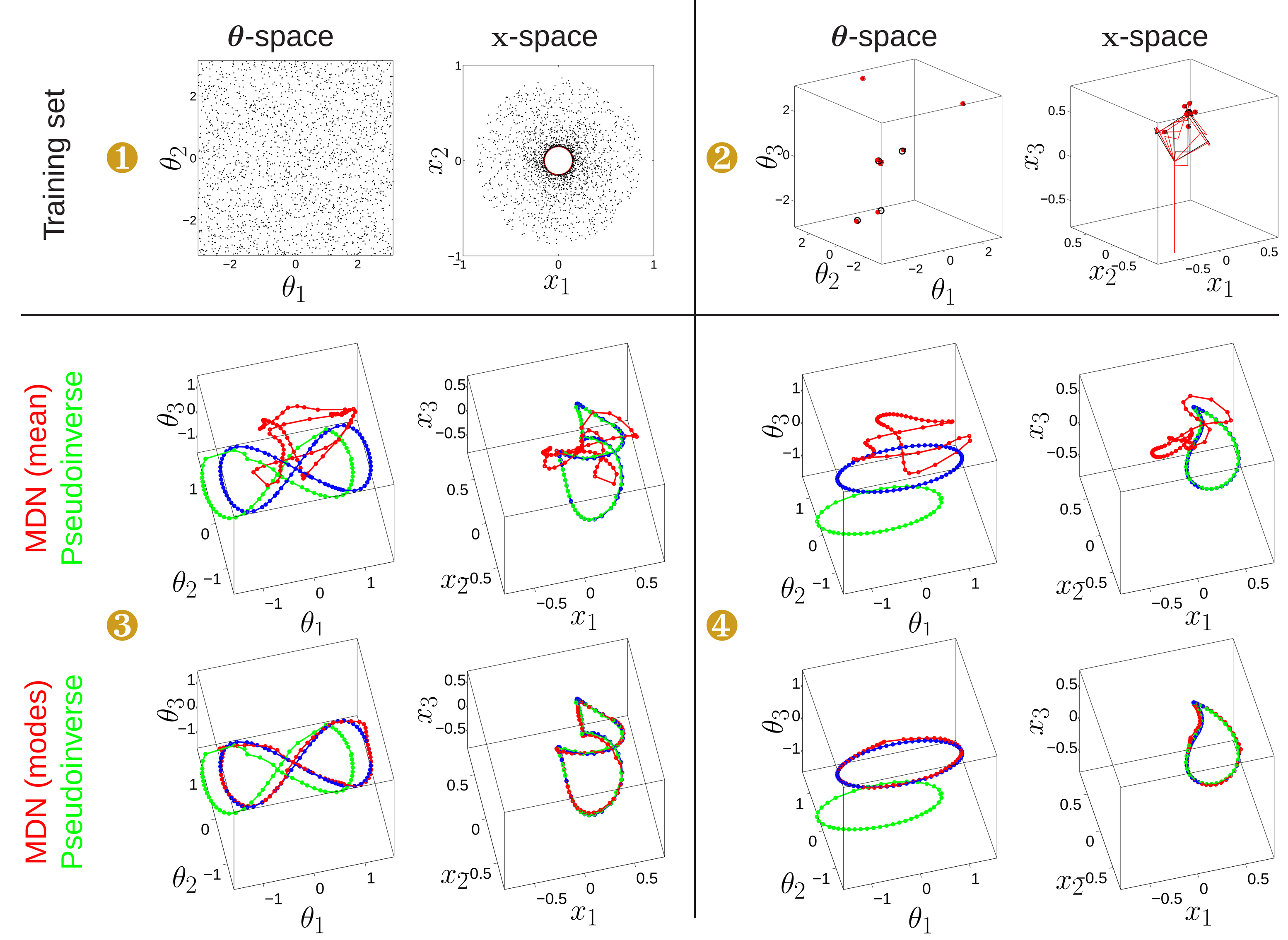
❶ We limit the $\boldsymbol{\theta}$-domain to $[0.3, 1.2] \times [1.5, 4.7]$ rad to complicate the topology and generate 2000 pairs by uniformly sampling $\boldsymbol{\theta}$-space and mapping with $\mathbf{f}$. We train: 1) full density $p(\boldsymbol{\theta}, \mathbf{x})$ by coarse ($M = 225$) and fine ($M = 2500$) GTMs (❷ shows the marginal density for the fine GTM); 2) conditional density $p(\boldsymbol{\theta}|\mathbf{x})$ by a MDN. ❸ shows the conditional density and modes for a given $\mathbf{x}$. ❹-❺ show reconstructed trajectories for the fine GTM and MDN. The conditional density mean averages two inverses, resulting in a fully-stretched arm, while the modes succeed in reconstructing true trajectories with good accuracy for all density models. ❻ shows the $\mathbf{x}$-trajectory could be produced by different $\boldsymbol{\theta}$-trajectories.

6 Experiments: redundant planar 3-link arm



MDN (36 components, 300 hidden units) using modes: The larger errors occur near singular configurations (e.g. fully-stretched arm). Pseudoinverse: is unstable and converges slowly near singularities. Both methods retrieve continuous (but different) trajectories.

5 Experiments: PUMA 560 robot arm



Results for a PUMA 560 arm with 3D angle space and 3D workspace: ❶ We generate a training set of 5000 pairs and train a MDN (12 components, 300 hidden units). ❷ shows the modes of the conditional density $p(\boldsymbol{\theta}|\mathbf{x})$ represent well the 4 true inverses (two combinations of elbow up/down) given a point in workspace. ❸-❹ show reconstructions for a figure-8 and an elliptical closed loops (original trajectories in blue). Note that symmetry of the problem results in several equivalent global solutions: our method and the pseudoinverse method choose different ones. The larger errors occur for points near a cylindrical hole (❶, right) at the centre of workspace which is not reachable by the arm, because of boundary effects of density models.

7 Discussion

- Data collection: need a training set $(\boldsymbol{\theta}_n, \mathbf{x}_n)$
- Run time
  - Bottleneck: mode-finding (may be greatly accelerated)
  - Matlab implementation: 50/10/4 ms per point (worse/average/best), while pseudoinverse takes 200/30/10 ms
- Summary
  - Obtains accurate solutions if the density model is good
  - Deals with singularities of the Jacobian and complicated angle domains naturally

8 Conclusions and future work

We introduce a machine learning method for trajectory IK that can deal with trajectories containing singularities, where the inverse mapping changes topology, and with complicated angle domains caused by mechanical constraints. Given a training set, 1) it learns a conditional density that implicitly represents the branches of the inverse mapping; the mappings are obtained by 2) finding the modes of the conditional density using a Gaussian mean-shift algorithm, and 3) the final $\boldsymbol{\theta}$-trajectory is obtained by minimising a global, trajectory-wide constraint over the set of modes. Future work will apply it to trajectory IK in animation, articulated pose tracking in computer vision, and articulatory inversion in speech.

Work funded by NSF CAREER award IIS-0754089