TRAJECTORY INVERSE KINEMATICS BY NONLINEAR, NONGAUSSIAN TRACKING

Chao Qin and Miguel Á. Carreira-Perpiñán

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

1 Abstract

We study trajectory inverse kinematics: to find a feasible trajectory in angle space that produces a given trajectory in workspace. We explicitly represent the multivalued inverse mapping by the modes of a conditional density of angles given workspace coordinates, estimated by a particle filter. We find all the modes using a mean-shift algorithm and then disambiguate the angle trajectory by minimising over the set of modes a global constraint that penalises discontinuous jumps in angle space or invalid inverses. We demonstrate the method with a PUMA 560 robot arm.

2 Problem statement

- $\mathbf{x}$: position in Cartesian workspace of the end-effector
- $\boldsymbol{\theta}$: joint angle
- $\mathbf{f}: \boldsymbol{\theta} \rightarrow \mathbf{x}$ forward kinematics
- Pointwise inverse kinematics (IK): $\boldsymbol{\theta} = \mathbf{f}^{-1}(\mathbf{x})$
- Trajectory IK: Given a $\mathbf{x}$-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

3 Trajectory inverse kinematics by nonlinear, nongaussian tracking

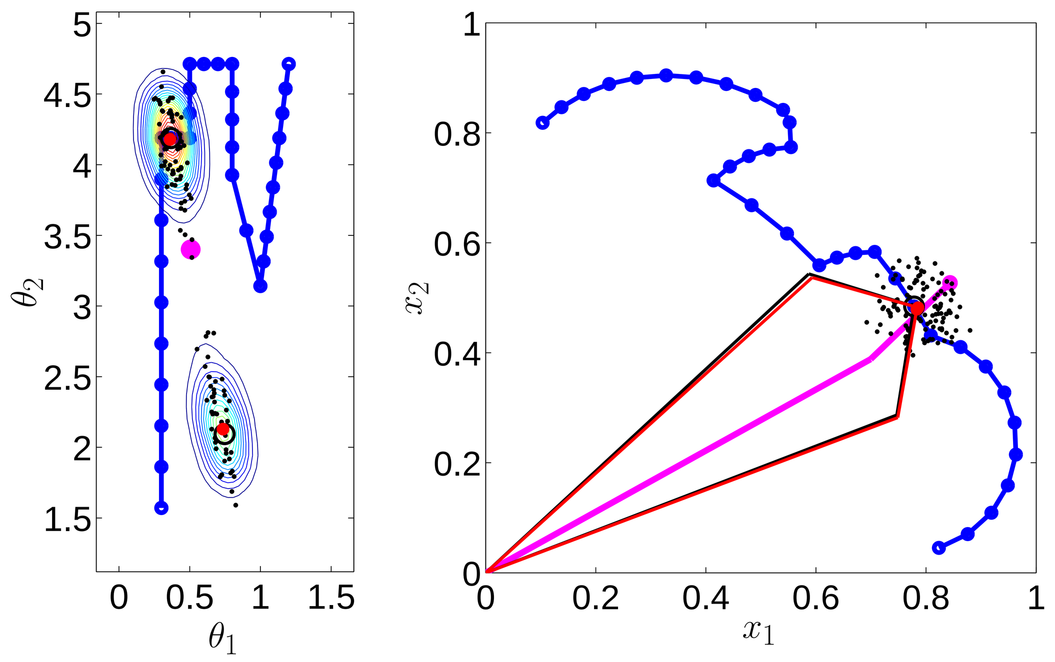
Idea of the method:

❶ At each trajectory point n, compute conditional density $p(\boldsymbol{\theta}_n|\mathbf{x}_{1:n})$ in angle space using a multimodal tracker

❷ At each trajectory point n, find the modes of $p(\boldsymbol{\theta}_n|\mathbf{x}_{1:n})$ and use them as approximated inverses $\mathbf{f}^{-1}(\mathbf{x}_n)$

❸ Over all points $n = 1, \ldots, N$, find a feasible sequence of modes that maps to $\mathbf{x}$-trajectory

Example: 2D robot arm



❶ Conditional density by tracking

- Formulate IK as a tracking problem
  - Treat $\mathbf{x}$ as observed measurements and $\boldsymbol{\theta}$ as unobserved states
  - Model the dynamics $p(\boldsymbol{\theta}_n|\boldsymbol{\theta}_{n-1})$ as a random walk with Gaussian noise ω_n, $\boldsymbol{\theta}_n = \boldsymbol{\theta}_{n-1} + \omega_n$
  - Use $\mathbf{f}$ with Gaussian noise υ_n as the measurement model, $p(\mathbf{x}_n|\boldsymbol{\theta}_n)$, $\mathbf{x}_n = \mathbf{f}(\boldsymbol{\theta}_n) + \upsilon_n$
- Run a nonlinear, nonGaussian tracker, i.e., particle filter or smoother to obtain at each $n = 1, \ldots, N$ a set $\{\boldsymbol{\theta}_n^m, w_n^m\}_{m=1}^M$ of M weighted particles to approximate posterior $p(\boldsymbol{\theta}_n|\mathbf{x}_{1:n})$ or $p(\boldsymbol{\theta}_n|\mathbf{x}_{1:N})$ at time n. We test:
  - Particle filter (PF): sequential importance resampling PF (SIR-PF)
  - Particle smoother (PS): forward-backward PS (FBS) and two-filter PS (TFS)
- Use Gaussian kernel density estimate to construct the conditional density, $p(\boldsymbol{\theta}_n|\mathbf{x}_{1:n}) = \sum_{m=1}^M w_n^m \exp(-\left\|\boldsymbol{\theta}_n - \boldsymbol{\theta}_n^m\right\|^2 / 2\sigma^2)$

❷ Mode finding

- Find all modes of conditional density $p(\boldsymbol{\theta}_n|\mathbf{x}_{1:n})$ by *Gaussian mean-shift (GMS)*, which starts iterating from every centroid of the GM and iterates $\boldsymbol{\theta}_n^{(\tau+1)} = \sum_{m=1}^M p(m|\boldsymbol{\theta}_n^{(\tau)})\boldsymbol{\theta}_n^m$
- Use all modes instead of the mean as statistical estimates to track all inverses

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
  - $\mathcal{C} = \sum_{n=1}^{N-1} \|\boldsymbol{\theta}_{n+1} - \boldsymbol{\theta}_n\|$: *continuity constraint* (integrated 1st derivative), penalise sudden angle changes
  - $\mathcal{F} = \sum_{n=1}^{N} \|\mathbf{x}_n - \mathbf{f}(\boldsymbol{\theta}_n)\|$: *forward constraint* (integrated workspace error), penalise spurious inverses

Computational complexity

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

Conditional density by tracking	Mode finding	Global optimisation
PF: $\mathcal{O}(M)$; PS: $\mathcal{O}(M^2)$	$\mathcal{O}(kM^2)$	$\mathcal{O}(N\nu^2)$

4 Experiments: PUMA 560 robot arm

Setup

- 3 DOF for $\boldsymbol{\theta}$, 3D workspace $\mathbf{x}$
- Repeat all experiments 20 times with random initialisations for each run

Methods

- SIR-PF + modes/mean
- FBS + modes/mean
- TFS + modes/mean
- Extended Kalman filter (EKF) and smoother (EKS)
- Pseudoinverse

Our method vs. local methods

- Pseudoinverse and EKF/EKS, being local, choose one of inverse branches and thus lose the rest at singularities of forward mapping. Branches chosen may have the risk of being invalid later.
- Our method, using a multimodal tracker, is able to track all local branches and only choose the global trajectory at the end.

Smoother vs. filter

- Marginal differences

Modes vs. mean

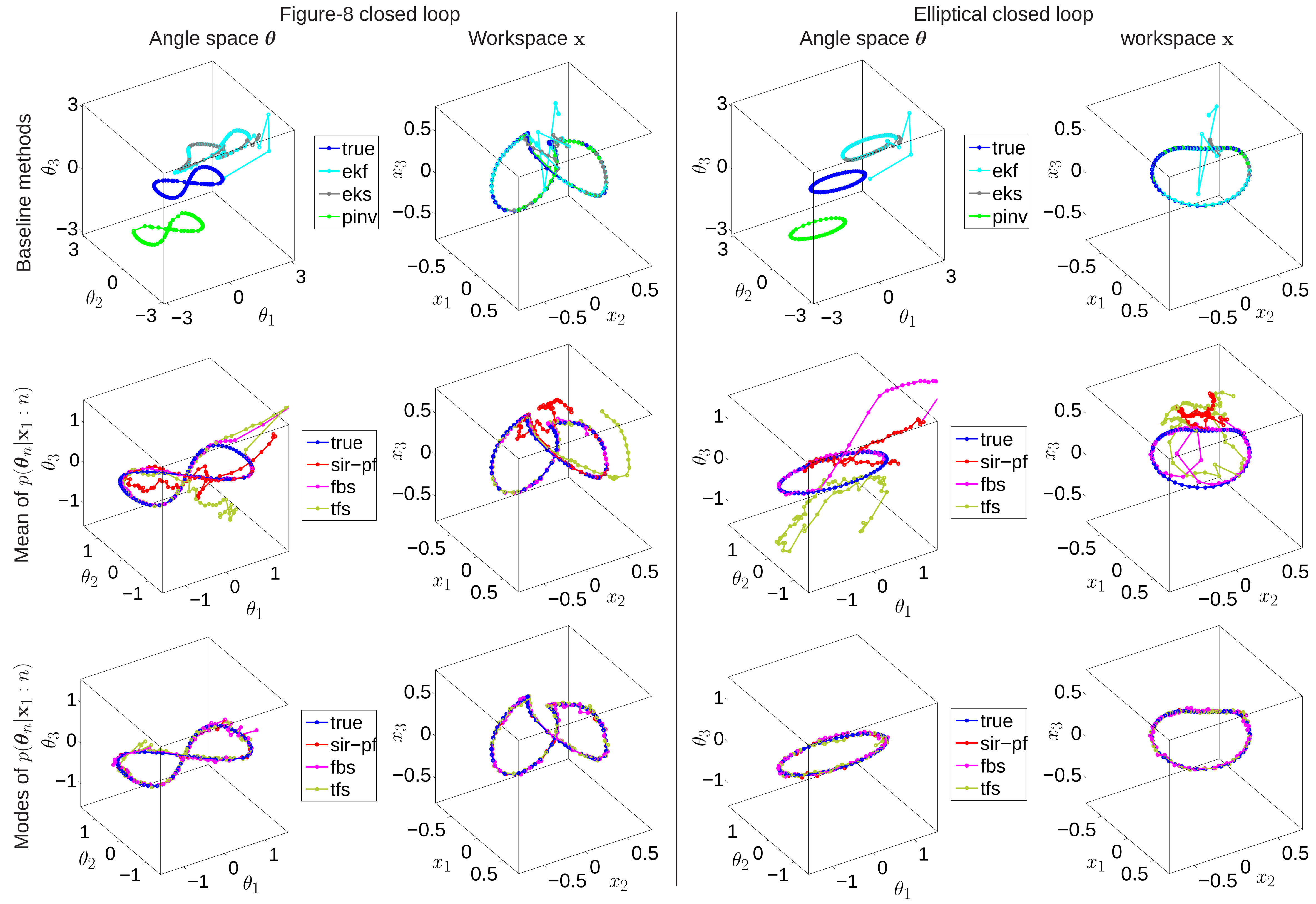
- Mean results in a wrong, "averaged" trajectory.
- Modes accurately represent the true inverses.

Online learning vs. offline learning of conditional density $p(\boldsymbol{\theta}|\mathbf{x})$

- Offline learning gives slightly lower reconstruction errors than online.
- Offline learning requires a set of training pairs $(\boldsymbol{\theta}_n, \mathbf{x}_n)$, which could be problematic as it is hard to sample a high-dim space.

Run time per trajectory point (sec), $M = 1000$ particles

pseudoinv	EKF	EKS	SIR-PF	FBS	TFS
0.03	0.015	0.017	1.4	0.8	0.8



Reconstruction error

Angle reconstruction error $\frac{1}{N}\sum_{n=1}^N \|\boldsymbol{\theta}_n - \hat{\boldsymbol{\theta}}_n\|$ (rad)

Trajectory	pseudoinv	EKF	EKS	SIR-PF mean	SIR-PF $\mathcal{C}+\lambda\mathcal{F}$	FBS $\mathcal{C}+\lambda\mathcal{F}$	TFS $\mathcal{C}+\lambda\mathcal{F}$
Elliptical	0.072	0.274	0.149	1.954 ± 0.654	0.116 ± 0.025	0.116 ± 0.025	0.100 ± 0.028
Figure–8	0.076	0.324	0.207	1.791 ± 0.520	0.141 ± 0.014	0.141 ± 0.014	0.144 ± 0.010
Open	0.042	1.230	0.706	2.543 ± 0.925	0.150 ± 0.028	0.150 ± 0.000	0.150 ± 0.028

Workspace reconstruction error $\frac{1}{N}\sum_{n=1}^N \|\mathbf{x}_n - \mathbf{f}(\hat{\boldsymbol{\theta}}_n)\|$

Trajectory	pseudoinv	EKF	EKS	SIR-PF mean	SIR-PF $\mathcal{C}+\lambda\mathcal{F}$	FBS $\mathcal{C}+\lambda\mathcal{F}$	TFS $\mathcal{C}+\lambda\mathcal{F}$
Elliptical	0.025	0.084	0.045	0.523 ± 0.182	0.033 ± 0.005	0.033 ± 0.005	0.029 ± 0.003
Figure–8	0.019	0.083	0.059	0.409 ± 0.117	0.031 ± 0.002	0.031 ± 0.003	0.033 ± 0.005
Open	0.007	0.252	0.261	0.940 ± 0.405	0.042 ± 0.006	0.041 ± 0.006	0.035 ± 0.006

5 Conclusions

We explicitly represent multivalued mappings by exploiting the power of particle filters to represent multimodal distributions and using a mode-finding algorithm for Gaussian mixtures — unlike much tracking work, which simply uses the mean. The final solution is obtained by minimising a global constraint that represents physical realisability. An advantage of the particle filter over offline learning of a conditional density model is that we do not need to collect a training set that samples the angle space (always hard in high dimensions). The method is applicable to other inverse problems over trajectories, such as articulatory inversion in speech, trajectory IK in computer graphics, and articulated pose tracking in computer vision.

Work funded by NSF CAREER award IIS-0754089