TRAJECTORY INVERSE KINEMATICS BY NONLINEAR, NONGAUSSIAN TRACKING

Chao Qin *Miguel Á. Carreira-Perpiñán*

EECS, School of Engineering, University of California, Merced. P.O. Box 2039, Merced, CA, USA
Email: {cqin,mcarreira-perpinan}@ucmerced.edu

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.

Index Terms— inverse kinematics, particle filters, mode finding, constraint minimisation.

1. INTRODUCTION

We consider the problem of trajectory inverse kinematics (IK) [1] of a (say) robot arm, where given a sequence of positions $\mathbf{x}_1, \dots, \mathbf{x}_N$ in (Cartesian) workspace of the end-effector, we want to obtain a feasible sequence of joint angles $\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_N$ that produce the $\mathbf{x}$-sequence (we do not consider dynamics in this paper). Given the joint angles, the end-effector position is given by the forward kinematics mapping, $\mathbf{x} = \mathbf{f}(\boldsymbol{\theta})$, which is usually (but not necessarily) known. However, the inverse $\mathbf{f}^{-1}(\mathbf{x})$ can take multiple values (e.g. see elbow up/down in fig. 1), or for redundant manipulators (where $\dim \boldsymbol{\theta} > \dim \mathbf{x}$), an infinite number of them; this makes it difficult to represent and compute $\mathbf{f}^{-1}$. At the same time, we want the recovered sequence of joint angles to trace a continuous, realisable trajectory. Importantly, our goal is not only to solve IK at each trajectory point, but also to obtain an angle trajectory that is globally feasible (e.g. avoiding discontinuities or forbidden regions). Problems related to IK arise in other areas: in computer graphics, where one wants to achieve realistic animation of articulated characters (e.g. [2]); in the articulatory inversion problem of speech, where an acoustic waveform ("position $\mathbf{x}$") may be produced by different vocal tract shapes ("angles $\boldsymbol{\theta}$") [3, 4], and we want to recover a physically feasible sequence of vocal tract shapes that produce a given acoustic utterance ("$\mathbf{x}$–trajectory").

The forward kinematics mapping $\mathbf{f}$ can usually be obtained in closed form for a kinematic chain as a product of homogeneous transformation matrices, one per link (however, we remark that this is not always the case, as in articulatory inversion). There exist many approaches to IK; we briefly review some of them here (see [5, 6] for review). In *analytic* approaches, one tries to obtain the IK mapping in closed form (e.g. [7]); this is only possible for certain types of manipulators, and even then it can be complicated. *Local* methods [8, 6] are based on linearising the forward mapping to obtain $\dot{\mathbf{x}} = \mathbf{J}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}$, where $\mathbf{J}$ is the Jacobian of $\mathbf{f}$ (Resolved Motion Rate Control [8]). This equation can then be integrated numerically in order to obtain the global trajectory for $\boldsymbol{\theta}$. For redundant manipulators, where $\mathbf{J}$ has more columns than rows, a unique value of $\dot{\boldsymbol{\theta}}$ may be obtained by optimising a suitable objective (such as energy) over the nullspace of the Jacobian; in particular, one can obtain the pseudoinverse method by minimising $\|\dot{\boldsymbol{\theta}}\|^2$, yielding $\dot{\boldsymbol{\theta}} = \mathbf{J}^+(\boldsymbol{\theta})\dot{\mathbf{x}}$, at a computational cost $\mathcal{O}(m^2 n)$ where $m = \dim \mathbf{x} < n = \dim \boldsymbol{\theta}$. However, the idea breaks down at singularities $\boldsymbol{\theta}^*$, where $\mathbf{J}(\boldsymbol{\theta}^*)$ becomes singular; this is caused by the existence of multiple inverse branches intersecting at $\boldsymbol{\theta}^*$. Also, the cost is high since many pseudoinverses of nonsparse Jacobians must be computed, and the numerical error accumulates over time. Other local methods [6] use an augmented set of variables $(\dot{\mathbf{x}}, \boldsymbol{\theta})$ rather than just $\dot{\mathbf{x}}$. Another local method (well-known in articulatory inversion) is *analysis-by-synthesis,* which directly finds an inverse value $\boldsymbol{\theta}$ of $\mathbf{f}$ by iteratively minimising the squared error $E(\boldsymbol{\theta}) = \|\mathbf{x} - \mathbf{f}(\boldsymbol{\theta})\|^2$ with a numerical optimisation method, e.g. gradient descent, where $\nabla E = 2\,\mathbf{J}(\boldsymbol{\theta})^T(\mathbf{f}(\boldsymbol{\theta}) - \mathbf{x})$. Unfortunately, which inverse value is found depends on the initial value for $\boldsymbol{\theta}$, and the iteration may also get stuck at non-inverse values where $\mathbf{J}(\boldsymbol{\theta})^T(\mathbf{f}(\boldsymbol{\theta}) - \mathbf{x}) = \mathbf{0}$ but $\mathbf{f}(\boldsymbol{\theta}) \neq \mathbf{x}$. However, the method is useful if the initial $\boldsymbol{\theta}$ is sufficiently close to the inverse sought. *Global* methods [9, 10] propose a variational formulation where the trajectory of $\boldsymbol{\theta}$ minimises a functional $\int_{t_0}^{t_1} G(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, t)\,dt$ (such as energy and manipulability) subject to the forward kinematic constraint $\mathbf{x}(t) = \mathbf{f}(\boldsymbol{\theta}(t))$ and appropriate boundary conditions. The trajectory is obtained by numerical integration of the corresponding Euler-Lagrange equation. However, the method still suffers from singularities [10] and needs the user to provide boundary conditions that are often unknown. Thus, an important problem of many of these methods are the singularities of the Jacobian. These correspond to the intersection of multiple inverse branches (violating the inverse function theorem), and while locally any of these branches is valid a priori, globally perhaps only one is valid.

A machine learning approach to trajectory IK was proposed in [13]. This directly models the multivalued inverse mapping $\boldsymbol{\theta} = \mathbf{f}^{-1}(\mathbf{x})$ by learning offline a multimodal conditional density model $p(\boldsymbol{\theta}|\mathbf{x})$; its modes represent the inverse solutions for $\boldsymbol{\theta}$. Given a trajectory in $\mathbf{x}$–space, one finds all the $\boldsymbol{\theta}$–modes at each trajectory point and then disambiguates the entire $\boldsymbol{\theta}$–trajectory by minimising a continuity constraint with dynamic programming. Here we propose a tracking approach, where we construct a conditional distribution of $\boldsymbol{\theta}$ (= unobserved states) given past and current $\mathbf{x}$ (= measurements) using a particle filter; then, we apply mode finding and constraint minimisation to recover the angle trajectory. Earlier work on the related problem of articulatory inversion [11, 12] has used (extended) Kalman filters (EKFs) to estimate the sequence of vocal tracts $\{\boldsymbol{\theta}_n\}$ that produce acoustics $\{\mathbf{x}_n\}$. However, EKFs cannot represent multimodal distributions and so are unable to track multiple solution branches at once; this is crucial in IK since some of these branches may turn out to be invalid in later points of the trajectory. We describe our method next and demonstrate it in experiments in section 3.

2. TRAJECTORY INVERSE KINEMATICS BY TRACKING

Consider a given $\mathbf{x}$–trajectory $\mathbf{x}_1, \dots, \mathbf{x}_N$ in workspace. Our overall algorithm works as follows. (1) We run a particle filter (or smoother) to obtain at each $n = 1, \dots, N$ a conditional distribution $p(\boldsymbol{\theta}_n|\mathbf{x}_{1:n})$ (or $p(\boldsymbol{\theta}_n|\mathbf{x}_{1:N})$). (2) We run a mode-finding algorithm on each distribution to find all its modes, which represent the multiple inverse solutions at each $\mathbf{x}_n$. (3) We obtain a unique $\boldsymbol{\theta}$–trajectory by minimising a constraint $\mathcal{C} + \lambda\mathcal{F}$ over the entire set of modes with dynamic programming. Steps 2–3 are the same as in [13]. Let us describe each step in detail.

Conditional density by tracking Our eventual objective is to obtain the multiple inverses of each $\mathbf{x}_n$ (i.e., values $\boldsymbol{\theta}$ s. t. $\mathbf{f}(\boldsymbol{\theta}) = \mathbf{x}_n$) from the modes of the conditional distribution of angles $\boldsymbol{\theta}$ given coordinates $\mathbf{x}$, $p(\boldsymbol{\theta}_n|\mathbf{x}_{1:n})$. We propose to construct the latter with a nonlinear, nongaussian tracker, where we consider the coordinates $\mathbf{x}$ as observed measurements and the angles $\boldsymbol{\theta}$ as unobserved states. Under the tracking framework [14, 15], the dynamic state-space model is given by $\boldsymbol{\theta}_n = \boldsymbol{\theta}_{n-1} + \omega_{n-1}$, $\mathbf{x}_n = \mathbf{f}(\boldsymbol{\theta}_n) + \upsilon_n$. We model the dynamics $p(\boldsymbol{\theta}_n|\boldsymbol{\theta}_{n-1})$ as a random walk with Gaussian noise ω_n, and the measurement model $p(\mathbf{x}_n|\boldsymbol{\theta}_n)$ is readily given by the forward mapping $\mathbf{f}$ with Gaussian noise υ_n. If the posterior distribution $p(\boldsymbol{\theta}_n|\mathbf{x}_{1:n})$ can be assumed unimodal, *extended Kalman filters (EKFs)* [14] can succeed. But, crucially, the posterior $p(\boldsymbol{\theta}_n|\mathbf{x}_{1:n})$ for IK is multimodal due to the many-to-one forward mapping $\mathbf{f}$. We then consider *particle filters (PFs)*, which can approximate multimodal distributions by a set of weighted samples (the particles), and have demonstrated superior performance over EKFs in many nonlinear, nongaussian problems. Various versions of PFs have been developed, including *sequential importance resampling PF (SIR-PF)*, sigma-point PF, unscented PF and others [15]. They mostly differ in the choice of the proposal distribution, whose support should cover the support of the true posterior. Here, we focus on the SIR-PF, which uses the transition prior as the proposal distribution, but other PFs would work as well as long as they can approximate multimodal posteriors. Furthermore, one can refine the posterior distribution by using all the measurements: $p(\boldsymbol{\theta}_n|\mathbf{x}_{1:N})$. This leads to *extended Kalman smoothers (EKSs)* and *particle smoothers (PSs)*. We consider two versions of PSs: *forward-backward smoother (FBS)*, which maintains the original particle locations from the PF but reweights them; and *two-filter smoother (TFS)* [16, 17], which combines a forward and a backward PF. The computational cost for M particles is $\mathcal{O}(M)$ for PFs and $\mathcal{O}(M^2)$ for PSs, which may be reduced to $\mathcal{O}(M \log M)$ by approximate, fast algorithms [18].

PFs (or PSs) only provide a set $\{\boldsymbol{\theta}_n^m, w_n^m\}_{m=1}^M$ of weighted particles to approximate the posterior $p(\boldsymbol{\theta}_n|\mathbf{x}_{1:n})$ at time n. Here, we use a Gaussian kernel density estimate to construct the conditional density, $p(\boldsymbol{\theta}_n|\mathbf{x}_{1:n}) = \sum_{m=1}^M w_n^m \exp(-\left\|\boldsymbol{\theta}_n - \boldsymbol{\theta}_n^m\right\|^2/2\sigma^2)$, where σ is the kernel width. Unlike most tracking work, we use all the modes instead of just the mean as the statistical estimate from the conditional density, as we must track multiple inverse branches.

Mode finding The conditional density $p(\boldsymbol{\theta}|\mathbf{x}_{1:n})$ has the form of a Gaussian mixture (GM). Efficient algorithms for finding all the modes of a GM exist [19] that iterate a hill-climbing algorithm from every centroid of the GM. In particular, *Gaussian mean-shift* iterates

$$\boldsymbol{\theta}_n^{(\tau+1)} = \textstyle\sum_{m=1}^M p(m|\boldsymbol{\theta}_n^{(\tau)})\boldsymbol{\theta}_n^m$$

where the posterior probability $p(m|\boldsymbol{\theta}_n^{(\tau)})$ is the normalised version of $w_n^m \exp\left(-\left\|(\boldsymbol{\theta}_n^{(\tau)} - \boldsymbol{\theta}_n^m)/2\sigma\right\|^2\right)$. Gaussian mean-shift does not require inverting matrices and takes $\mathcal{O}(kM^2)$ where M is the number of particles and k the average number of iterations per particle. The computational time can be drastically reduced, for example discarding low-weight particles, or eliminating redundant particles (having essentially identical locations and weights) which often occur during resampling; see [20] for other accelerations.

Global optimisation with dynamic programming Assume we have collected for each step n in the trajectory all the modes (candidate inverses). In principle, each of these modes represents a correct solution for step n (following a certain solution branch), but a given branch that is valid for part of the trajectory may be invalid for another part (e.g. because certain joint angles' values are forbidden due to mechanical constraints). In order to determine the solution, we minimise a global, trajectory-wide constraint over the set of modes. In this paper, we consider a constraint of the form $\mathcal{C} + \lambda\mathcal{F}$ (for $\lambda \geq 0$), where $\mathcal{C} = \sum_{n=1}^{N-1} \|\boldsymbol{\theta}_{n+1} - \boldsymbol{\theta}_n\|$ represents a *continuity constraint* (integrated 1st derivative). This penalises discontinuous jumps in $\boldsymbol{\theta}$-space and encourages short trajectories. $\mathcal{F} = \sum_{n=1}^{N} \|\mathbf{x}_n - \mathbf{f}(\boldsymbol{\theta}_n)\|$ represents a *forward constraint* (integrated workspace error), and penalises invalid inverses, i.e., modes $\boldsymbol{\theta}_n$ that do not map near the desired $\mathbf{x}_n$. This helps to eliminate spurious modes produced by ripple in the density model. *Global* minimisation of the constraint can be obtained by dynamic programming in $\mathcal{O}(N\nu^2)$ where ν is the average number of modes per step (usually very small), thus in linear time on the trajectory length N. Computationally, this is generally negligible compared to the mode-finding step.

3. EXPERIMENTS WITH A PUMA 560 ROBOT ARM

We illustrate the methods' performance with known ground truth. All experiments were repeated 20 times with random initialisations for each run in order to calculate variance estimates of performance.

We consider a PUMA 560 robot arm (a widespread industrial manipulator) with 3 dof for position $\boldsymbol{\theta} = (\theta_1,\ \theta_2,\ \theta_3)$, 3 dof for orientation (which we ignore), and a 3D workspace $\mathbf{x} \in \mathbb{R}^3$. The (point) IK can be solved analytically for this robot [7] and yields 4 solution branches (two combinations of elbow up/down); we use the implementation of the Matlab Robotics Toolbox [21]. We limit the angle domain to $[-\pi, \pi] \times [-\pi, \pi] \times [-\pi, \pi]$ in order to complicate the topology of the inverse mapping. We run the traditional pseudoinverse method and our approach with the following trackers: EKF, EKS, SIR-PF, FBS, TFS. The process and measurement noises are isotropic with variances $\sigma_\omega^2 = 0.125$ and $\sigma_\upsilon^2 = 0.005$, respectively. For SIR-PF/FBS/TFS we used $M = 1\,000$ particles and a kernel density estimate width $\sigma = 0.04$. Fig. 1 illustrates (for a 2D robot arm, which is easier to visualise) how the modes of the conditional posterior on $\boldsymbol{\theta}$ represent the inverses of $\mathbf{f}$. Figs. 2–3 show reconstructions for a figure–8 and elliptical trajectories. Because of the symmetry of the forward mapping, there are several $\boldsymbol{\theta}$–trajectories that produce the given $\mathbf{x}$–trajectory. The pseudoinverse method, being local, sticks to one of these trajectories from the beginning (depending on its initialisation) and can never recover the others later on. The same thing happens when using trackers such as EKF/EKS which assume a unimodal distribution; they can only recover one such trajectory, depending on initialisation. Likewise, at singularities of the forward mapping (fully stretched arm), where multiple inverse branches merge or diverge, these methods (pseudoinverse, EKF/EKS) choose one of the branches and the rest are irreversibly lost. The problem with this local choice is that it may

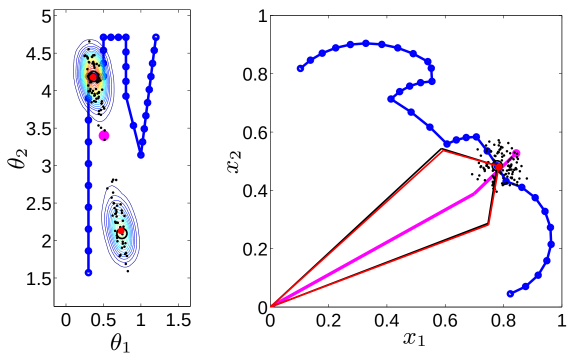

Fig. 1. Illustration for a 2D robot arm of using conditional modes to represent multiple inverses. *Right*: a point $\mathbf{x} \in [0,1]^2$ in workspace can be reached by two joint angle configurations, elbow up/down. *Left*: the two configurations are captured as the two modes of a conditional density $p(\boldsymbol{\theta}|\mathbf{x})$ obtained from a PF (contours). Its mean (magenta dot) is not a valid inverse, mapping to a fully stretched arm. The plots show a trajectory in $\mathbf{x}$ and $\boldsymbol{\theta}$ space (blue) and the particles (black dots) using a SIR-PF.

turn out to be wrong later on in the trajectory, e.g. only an elbow-up configuration may be valid in certain workspace regions because of mechanical limits on some angles. In our method, if using a multimodal tracker (SIR-PF, FBS, TFS) we are able to track all branches (thus all solutions) at every time; it is only at the end that a globally valid trajectory (avoiding discontinuities) is obtained by minimising the constraint. Fig. 2 also shows how if we simply use the mean of the PF distribution (instead of all its modes), a wrong, "average" trajectory is obtained. Table 1 gives the errors in $\boldsymbol{\theta}$ and $\mathbf{x}$ wrt ground truth for several trajectories. Generally, we find little difference among the 3 multimodal trackers tested (SIR-PF, FBS, TFS), which all succeed in recovering the ground truth with good accuracy. We do occasionally find (results not shown) that, depending on the initialisation, the SIR-PF may fail to track all the modes and thus miss possible inverse branches (this could be corrected using more particles). The smoothers (FBS, TFS) are more robust in that the backward filter helps to recover such modes. We also obtained very similar results using a smoothness constraint (second-order derivative) instead of continuity; and using a continuity constraint only (i.e., $\lambda = 0$), which indicates that the modes from the conditional density are accurate representatives of the true inverses.

Table 2 lists average running times in seconds (per trajectory point) with different trackers in our Matlab implementation. Most of the runtime for SIR-PF is spent on the forward mapping `fkine` (inefficiently implemented in [21]). The PSs do not rely on the forward mapping and thus appear to run faster.

Finally, we compared deriving the conditional density from a particle filter with learning offline a conditional density $p(\boldsymbol{\theta}|\mathbf{x})$ (e.g. with a Gaussian mixture) given a training set of pairs $(\boldsymbol{\theta}_n, \mathbf{x}_n)$, as done in [13]. The latter gave slightly lower reconstruction errors ($\boldsymbol{\theta}/\mathbf{x}$ spaces: elliptical, $0.071/0.029$; figure–8, $0.081/0.027$; open, $0.173/0.055$). However, offline collection of such training data may be a problem, as it is difficult to sample a high-dimensional space.

4. CONCLUSIONS

We have considered the problem of trajectory inverse kinematics, which is hard because of the multiple solutions of (pointwise) IK and the existence of singularities, and because of the need to recover angle trajectories that are feasible (e.g. avoiding discontinuities). 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 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.

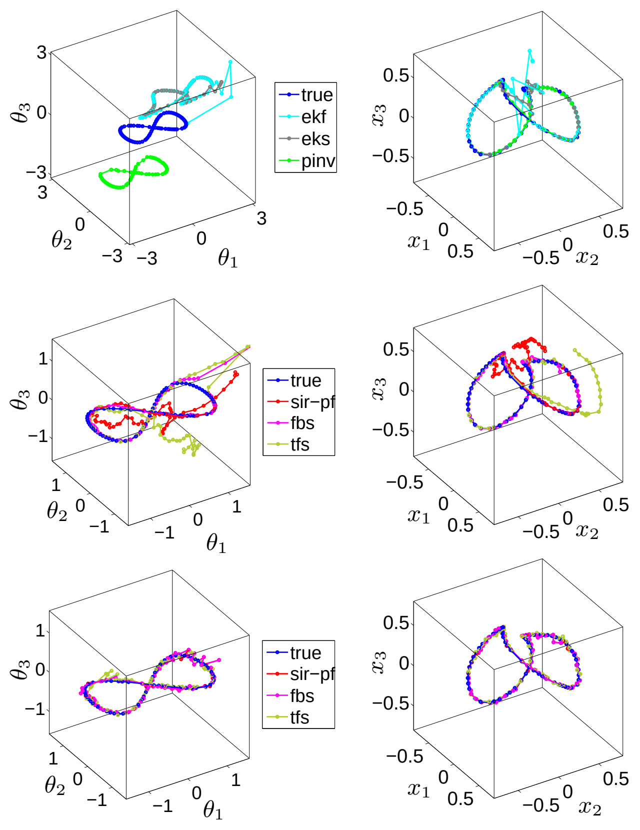

Fig. 2. Reconstruction of a figure-8 trajectory for the PUMA 560 robot arm. *Left*: $\boldsymbol{\theta}$-space, *right*: $\mathbf{x}$-space. *Top*: pseudoinverse, EKF, EKS. *Middle*: conditional means from SIR-PF, FBS, TFS. *Bottom*: conditional modes from SIR-PF, FBS, TFS and constraint $\mathcal{C} + \lambda\mathcal{F}$.

Table 2. Run time per trajectory point (sec.), $M = 1\,000$ particles.

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

5. ACKNOWLEDGEMENTS

MACP thanks Sethu Vijayakumar and Marcelo Kallman for valuable discussions. Work funded by NSF CAREER award IIS–0754089.

Table 1. Reconstruction errors for PUMA 560 robot arm ($\mathbf{x}_n$: given $\mathbf{x}$–trajectory, $\hat{\boldsymbol{\theta}}_n$: reconstructed $\boldsymbol{\theta}$–trajectory, $\boldsymbol{\theta}_n$: true $\boldsymbol{\theta}$–trajectory).

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	FBS mean	TFS 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	0.925 ± 0.655	1.304 ± 0.898	0.116 ± 0.025	0.116 ± 0.025	0.100 ± 0.028
Figure–8	0.076	0.324	0.207	1.791 ± 0.520	1.239 ± 0.969	1.505 ± 0.807	0.141 ± 0.014	0.141 ± 0.014	0.144 ± 0.010
Open	0.042	1.230	0.706	2.543 ± 0.925	1.157 ± 0.996	0.826 ± 0.160	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	FBS mean	TFS 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	0.052 ± 0.037	0.320 ± 0.236	0.033 ± 0.005	0.033 ± 0.005	0.029 ± 0.003
Figure–8	0.019	0.083	0.059	0.409 ± 0.117	0.039 ± 0.022	0.252 ± 0.111	0.031 ± 0.002	0.031 ± 0.003	0.033 ± 0.005
Open	0.007	0.252	0.261	0.940 ± 0.405	0.119 ± 0.095	0.079 ± 0.037	0.042 ± 0.006	0.041 ± 0.006	0.035 ± 0.006

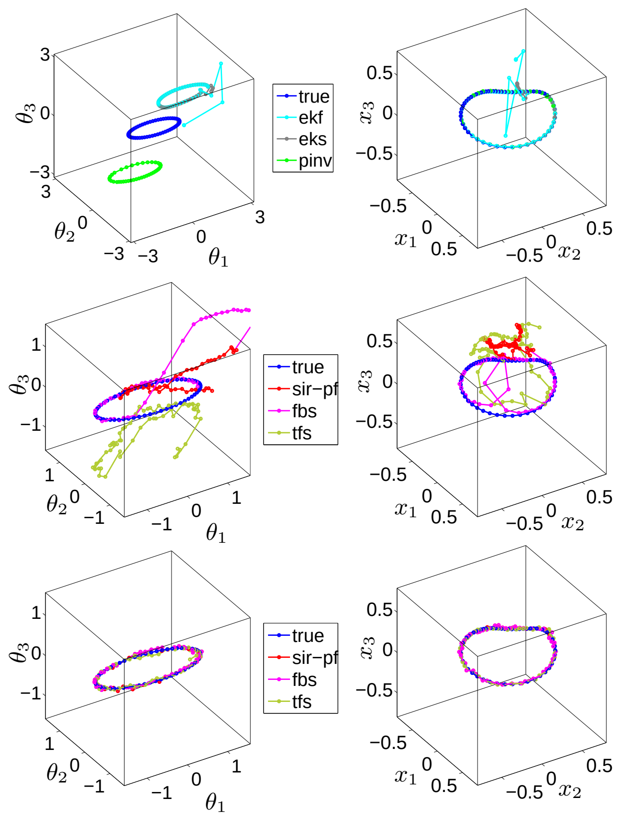

Fig. 3. As fig. 2 but for an elliptical trajectory.

6. REFERENCES

[1] J. Craig, *Introduction to Robotics. Mechanics and Control*, Addison-Wesley, Reading, MA, 1989.

[2] K. Grochow, S. L. Martin, A. Hertzmann, and Z. Popović, "Style-based inverse kinematics," *ACM Trans. Graphics*, 2004.

[3] J. Schroeter and M. M. Sondhi, "Techniques for estimating vocal-tract shapes from the speech signal," *IEEE Trans. Speech and Audio Process.*, 1994.

[4] C. Qin and M. Á. Carreira-Perpiñán, "An empirical investigation of the nonuniqueness in acoustic-to-articulatory mapping," in *Proc. Interspeech*, 2007.

[5] C. A. Klein and C. H. Huang, "Review of pseudoinverse control for use with kinematically redundant manipulators," *IEEE Trans. Systems, Man, and Cybernetics*, 1983.

[6] A. D'Souza, S. Vijayakumar, and S. Schaal, "Learning inverse kinematics," in *IROS*, vol. 1, pages 298-303, 2001.

[7] R. P. Paul and H. Zhang, "Computationally efficient kinematics for manipulators with spherical wrists based on the homogeneous transformation representation," *Int. J. of Robotics Research*, 1986.

[8] D. E. Whitney, "Resolved motion rate control of manipulators and human prostheses," *IEEE Trans. Man-Machine Systems*, vol. 10, pages 47-53, 1969.

[9] Y. Nakamura and H. Hanafusa, "Optimal redundancy control of robot manipulators," *Int. J. of Robotics Research*, 1987.

[10] D. P. Martin, J. Baillieul, and J. M. Hollerbach, "Resolution of kinematic redundancy using optimization techniques," *IEEE Trans. Robotics & Automation*, 1989.

[11] S. King and A. Wrench, "Dynamical system modelling of articulator movement," in *Proc. Int. Congress Phonetic Sciences (ICPhS'99)*, 1999.

[12] S. Dusan and L. Deng, "Recovering vocal tract shapes from MFCC parameters," in *ICSLP*, 1998.

[13] M. Á. Carreira-Perpiñán, "Reconstruction of sequential data with probabilistic models and continuity constraints," in *NIPS*, vol. 12, pages 414-420, 2000.

[14] B. Anderson and J. Moore, *Optimal Filtering*, 1979.

[15] A. Doucet, N. de Freitas, and N. Gordon, *An introduction to sequential Monte Carlo Methods*, Springer-Verlag, 2001.

[16] G. Kitagawa, "Monte Carlo filter and smoother for non-Gaussian nonlinear state space models," *J. of Computational and Graphical Statistics*, 1996.

[17] M. Isard and A. Blake, "A smoothing filter for CONDENSATION," in *ECCV*, vol. 1, pages 767-781, 1998.

[18] M. Klaas, M. Briers, N. de Freitas, A. Doucet, S. Maskell, and D. Lang, "Fast particle smoothing: If I had a million particles," in *ICML*, pages 481-488, 2006.

[19] M. Á. Carreira-Perpiñán, "Mode-finding for mixtures of Gaussian distributions," *IEEE Trans. PAMI*, 22(11), 2000.

[20] M. Á. Carreira-Perpiñán, "Acceleration strategies for Gaussian mean-shift image segmentation," in *CVPR*, 2006.

[21] P. I. Corke, "A robotics toolbox for MATLAB," *IEEE Robotics & Automation Magazine*, 1996.