A comparison of three-dimensional rotation formalisms for least-squares and Bayesian inverse kinematics
Appendix A

Ben Serrien, Klevis Aliaj, Todd Pataky

Mathematical and statistical details of the inverse kinematic methods that are omitted in the main paper are described here. This appendix is not meant to be a review or full derivation of these methods, we refer to the reference list for that purpose. It is meant to present the details of the different methods in a single notational framework and to serve as an explanation of the code used in the simulations. In Appendix B, a `python` notebook explains the use of the package.

1 3D rigid body kinematics and the $SO(3)$ group

In the main paper we describe the accuracy of inverse kinematics methods for rigid bodies with 3 rotational degrees of freedom. That is, we assume that a set of n markers attached to a rigid body are displaced in 3D Euclidean space through a rotation of the body around an axis passing through the origin of a fixed reference frame. The coordinates of the markers are described by 3-vectors $\boldsymbol{r}_i$ $(i = 1, ..., n)$. In the simulations we always used $n = 4$, which is a number used in many biomechanical models. The forward kinematics equation describing this rotation between an initial (0) and final (1) position is the following:

$$\boldsymbol{r}_{i1} = \boldsymbol{R}(\boldsymbol{\theta}) \cdot \boldsymbol{r}_{i0} \tag{1}$$

with $\boldsymbol{R}(\boldsymbol{\theta}) \in \mathbb{R}^{3\times3}$ a rotation matrix. The objective of inverse kinematics (IK) is to find $\boldsymbol{\theta}$ based on the 3D coordinates. In reality, not the true but noisy coordinates are observed (measured): $\tilde{\boldsymbol{r}}_i = \boldsymbol{r}_i + \boldsymbol{\epsilon}_i$ which complicates the task. In the main paper we assume that the measurement noise is zero-mean uncorrelated Gaussian isotropic noise:

$$\boldsymbol{\epsilon}_i \overset{i.i.d.}{\sim} \mathcal{N}\left(\boldsymbol{0}, \sigma^2 \boldsymbol{I}_3\right) \tag{2}$$

or equivalently because we assume uncorrelated and isotropic noise:

$$\epsilon_{ij} \overset{i.i.d.}{\sim} \mathcal{N}\left(0, \sigma^2\right) \quad (j = x, y, z) \tag{3}$$

The rotation matrix is only one of the representations of a 3D rotation. In general, rotations in 3D Euclidean space form an algebraic group, called the Special Orthogonal group $SO(3)$ which can be represented in various ways. All rotations are denoted by $\boldsymbol{\theta} \in SO(3)$. The rotation matrix can be constructed based on the specific representation of $\boldsymbol{\theta}$ and consists of 9 numbers while only 3 degrees of freedom are present in this setting. More compact representations are the rotation vector, Euler/Cardan angles and unit quaternions (explained below). In the package, we used the `Rotation` class from the `scipy.spatial.transform` module which incorporates these representations, conversions between them and other useful functions.

1.1 Rotation vector

The rotation vector representation is also known as the axis-angle or helical axis representation. The rotation vector is a pseudo-vector whose unit vector $(\boldsymbol{u})$ describes the orientation of the axis around which the rigid body is rotating and whose magnitude (θ) is the angle over which the rotation occurs. This representation consists thus of 4 numbers plus a unit vector constraint: $\boldsymbol{\theta} = (\boldsymbol{u}, \theta) = (u_x, u_y, u_z, \theta)$. The rotation matrix takes the following form:

$$\boldsymbol{R}(\boldsymbol{u}, \theta) = \begin{pmatrix} \cos\theta + u_x^2(1-\cos\theta) & u_x u_y(1-\cos\theta) - u_z\sin\theta & u_x u_z(1-\cos\theta) + u_y\sin\theta \\ u_y u_x(1-\cos\theta) + u_z\sin\theta & \cos\theta + u_y^2(1-\cos\theta) & u_y u_z(1-\cos\theta) - u_x\sin\theta \\ u_z u_x(1-\cos\theta) - u_y\sin\theta & u_z u_y(1-\cos\theta) + u_x\sin\theta & \cos\theta + u_z^2(1-\cos\theta) \end{pmatrix} \tag{4}$$

1.2 Euler and Cardan angles

Euler and Cardan angles describe 3D rotations as a sequence of three rotations around a specific set of axes (in contrast to the rotation vector which describes it as a rotation over an angle around 1 axis). This representation thus consists of three numbers: $\boldsymbol{\theta} = (\alpha, \beta, \gamma)$. The International Society of Biomechanics has a set of guidelines (Wu et al., 2002, 2005) that describe the recommended sequence for intrinsic Euler and Cardan angles for various joint coordinate systems. For most joints, the ZXY sequence is recommended, but the YXY sequence is used for the shoulder. The rotation matrices for these sequences are given below with α, β, γ the angles corresponding to the first, second and third rotations respectively:

$$\boldsymbol{R}_{ZXY}(\alpha,\beta,\gamma) = \begin{pmatrix} \cos\alpha\cos\gamma - \sin\alpha\sin\beta\sin\gamma & -\cos\beta\sin\alpha & \cos\alpha\sin\gamma + \cos\gamma\sin\alpha\sin\beta \\ \cos\gamma\sin\alpha + \cos\alpha\sin\beta\sin\gamma & \cos\alpha\cos\beta & \sin\alpha\sin\gamma - \cos\alpha\cos\gamma\sin\beta \\ -\cos\beta\sin\gamma & \sin\beta & \cos\beta\cos\gamma \end{pmatrix} \tag{5}$$

$$\boldsymbol{R}_{YXY}(\alpha,\beta,\gamma) = \begin{pmatrix} \cos\alpha\cos\gamma - \cos\beta\sin\alpha\sin\gamma & \sin\alpha\sin\beta & \cos\alpha\sin\gamma + \cos\beta\cos\gamma\sin\alpha \\ \sin\beta\sin\gamma & \cos\beta & -\cos\gamma\sin\beta \\ -\cos\gamma\sin\alpha - \cos\alpha\cos\beta\sin\gamma & \cos\alpha\sin\beta & \cos\alpha\cos\beta\cos\gamma - \sin\alpha\sin\gamma \end{pmatrix} \tag{6}$$

1.3 Unit quaternions

Quaternions (q) are 4-dimensional numbers, consisting of a complex vector part ($\boldsymbol{q}$) and a real scalar part (q_w): $q = \boldsymbol{q} + q_w = q_x\boldsymbol{i} + q_y\boldsymbol{j} + q_z\boldsymbol{k} + q_w$. See for example Horn (1987) for a reference on quaternion algebra in the context of 3D rotations. Unit quaternions are quaternions whose norm equals 1. The relation between unit quaternions and the rotation vector representation is the following: $q = \boldsymbol{u}\sin\frac{\theta}{2} + \cos\frac{\theta}{2}$; it is seen that for each $\boldsymbol{u}$ and θ, this gives a unit quaternion. Each rotation can thus be represented by a single parameter consisting of 4 numbers: $\boldsymbol{\theta} = q$. Conventional scalars and 3-vectors can be represented as quaternions with a zero-vector part and a zero-scalar part respectively. A rotation of a 3-vector to its new position can be written equivalently as premultiplying it with a rotation matrix or using the quaternion product as follows[1]:

$$\boldsymbol{r}_{i1} = \boldsymbol{R}(q)\cdot\boldsymbol{r}_{i0} = q\otimes\boldsymbol{r}_{i0}\otimes q^* \tag{7}$$

The rotation matrix in terms of the components of the unit quaternion takes the following form:

$$\boldsymbol{R}(q) = \begin{pmatrix} q_w^2 + q_x^2 - q_y^2 - q_z^2 & 2(q_xq_y - q_wq_z) & 2(q_xq_z + q_wq_y) \\ 2(q_xq_y + q_wq_z) & q_w^2 - q_x^2 + q_y^2 - q_z^2 & 2(q_yq_z - q_wq_x) \\ 2(q_xq_z - q_wq_y) & 2(q_yq_z + q_wq_x) & q_w^2 - q_x^2 - q_y^2 + q_z^2 \end{pmatrix} \tag{8}$$

Unit quaternions are known to be numerically more stable and not to suffer from singularities or gimbal lock like the Euler/Cardan angles (Zatsiorsky, 1998; Bottema and Roth, 1979).

2 Least-Squares Inverse Kinematics (LS-IK)

In LS-IK, finding $\boldsymbol{\theta}$ is done by minimizing the sum of squared deviations between the observed noisy markers and the predicted marker coordinates:

$$\hat{\boldsymbol{\theta}}_{\mathrm{LS}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \sum_{i=1}^{n} \left(\tilde{\boldsymbol{r}}_{i1} - \boldsymbol{R}(\boldsymbol{\theta})\cdot\tilde{\boldsymbol{r}}_{i0}\right)^2 \tag{9}$$

2.1 Singular Value Decomposition (SVD)

Söderkwist and Wedin (1993) described a LS-IK solution using SVD. The method works as follows. Let the mean-centered marker coordinates be defined as $\boldsymbol{A} = [\tilde{\boldsymbol{r}}_{0,1} - \bar{\boldsymbol{r}}_0, ..., \tilde{\boldsymbol{r}}_{0,n} - \bar{\boldsymbol{r}}_0]$ and $\boldsymbol{B} = [\tilde{\boldsymbol{r}}_{1,1} - \bar{\boldsymbol{r}}_1, ..., \tilde{\boldsymbol{r}}_{1,n} - \bar{\boldsymbol{r}}_1]$, both $\in \mathbb{R}^{3\times n}$ ($\bar{\boldsymbol{r}}_{0/1} = \frac{1}{n}\sum_{i=1}^{n}\tilde{\boldsymbol{r}}_{0/1,i}$) and define $\boldsymbol{C} = \boldsymbol{B}\cdot\boldsymbol{A}^T$. The 3×3 matrix $\boldsymbol{C}$ can be decomposed using SVD:

$$\boldsymbol{C} = \boldsymbol{U}\cdot\boldsymbol{D}\cdot\boldsymbol{V}^T \tag{10}$$

The SVD estimate for the rotation matrix can now be written as:

$$\hat{\boldsymbol{R}} = \boldsymbol{U}\cdot\mathrm{diag}(1, 1, \det(\boldsymbol{U}\cdot\boldsymbol{V}))\cdot\boldsymbol{V} \tag{11}$$

[1] The conjugate of $q = \boldsymbol{q} + q_w$ is given by $q^* = -\boldsymbol{q} + q_w$

2.2 Quaternion based LS-IK (lsQUAT)

A similar algorithm to SVD was proposed by Horn (1987) using quaternion algebra. The matrix $\boldsymbol{C}$ is defined in the same way as for SVD; this matrix contains all the information necessary for finding the LS solution to the rotation (Horn, 1987). Now define the 4×4 matrix $\boldsymbol{N}$:

$$\boldsymbol{N} = \begin{pmatrix} \text{trace}(\boldsymbol{C}) & \Delta^T \\ \Delta & \boldsymbol{C} + \boldsymbol{C}^T - \text{trace}(\boldsymbol{C})\boldsymbol{I}_3 \end{pmatrix} \tag{12}$$

where $\Delta = (C_{2,3} - C_{3,2}, C_{3,1} - C_{1,3}, C_{1,2} - C_{2,1})^T$. Horn (1987) shows that minimizing the sum of squared errors is done by the unit quaternion $\hat{q}$ which is the eigenvector of $\boldsymbol{N}$ corresponding to its most positive eigenvalue (λ_m); $\boldsymbol{N}\hat{q} = \lambda_m \hat{q}$.

2.3 Non-Linear Least-Squares (NLLS) IK

Because of the similarity to Variational BIK (infra), the NLLS method will be used with the unit quaternion representation. In NLLS, the forward model $\boldsymbol{f}$ is approximated by a local linearization using a first-order Taylor series.

$$\begin{aligned} \boldsymbol{r}_{i1} = \boldsymbol{f}_i(q) &= q \otimes \boldsymbol{r}_{i0} \otimes q^* \\ &\approx \boldsymbol{f}_i(m) + \boldsymbol{J}_i(q - m) \end{aligned} \tag{13}$$

The Jacobian of the forward model is given by the following (Solà, 2017)[2]:

$$\boldsymbol{J}_i = \frac{\partial \boldsymbol{f}_i}{\partial q} = 2[\boldsymbol{q}^T \boldsymbol{r}_{i0} \boldsymbol{I}_3 + \boldsymbol{q}\boldsymbol{r}_{i0}^T + \boldsymbol{r}_{i0}\boldsymbol{q}^T - q_w[\boldsymbol{r}_{i0}]_\times \mid q_w \boldsymbol{r}_{i0} + \boldsymbol{q} \times \boldsymbol{r}_{i0}] \in \mathbb{R}^{3\times 4} \tag{14}$$

The full Jacobian of all markers is thus a $3n$×4 real matrix. Minimizing the sum of squared errors under this approximation leads to an iterative update equation of m. Using the Gauss-Newton algorithm, this is given by:

$$m^{(t+1)} = m^{(t)} + \left(\boldsymbol{J}^{(t)T}\boldsymbol{J}^{(t)}\right)^{-1} \boldsymbol{J}^{(t)T}\boldsymbol{k}^{(t)} \tag{15}$$

with $\boldsymbol{k}^{(t)} = \tilde{\boldsymbol{r}}_1 - \boldsymbol{f}(m^{(t)})$ (a vector of length $3n$); the residual between the observed and predicted coordinates at iteration t (Chappell et al., 2009). At each iteration, m is properly normalized to a unit quaternion and the Jacobian is evaluated using the noisy coordinates. For starting the iteration at a good guess, we used the SVD estimate ($m^{(0)} = \hat{q}_{\text{SVD}}$). Convergence of the NLLS algorithm can be defined using various criteria. In the main paper we defined convergence if the change between two succesive iterations was smaller than 0.01°.

3 Bayesian Inverse Kinematics (BIK)

The major difference between LS-IK and BIK is that LS-IK is a purely mathematical optimization procedure, while BIK is a stochastic inference method (statistical optimization). Distributional assumptions are assigned to both the rotation ("parameter of interest"), the measurement noise ("nuisance parameter") and to the final noisy marker coordinates ("data")[3]. Initial noisy marker coordinates are treated as constants. In BIK, finding $\boldsymbol{\theta}$ is done by maximizing the posterior probability of $\boldsymbol{\theta}$ given the noisy coordinates and the prior information:

$$\hat{\boldsymbol{\theta}}_{\text{BIK}} = \underset{\boldsymbol{\theta}}{\text{argmax}}\ p(\boldsymbol{\theta}, \tau | \tilde{\boldsymbol{r}}_1) \tag{16}$$

The posterior distribution is obtained using Bayes' rule:

$$p(\boldsymbol{\theta}, \tau | \tilde{\boldsymbol{r}}_1) \propto p(\tilde{\boldsymbol{r}}_1 | \boldsymbol{\theta}, \tau) \cdot p(\boldsymbol{\theta}) \cdot p(\tau) \tag{17}$$

with $p(\boldsymbol{\theta})$ and $p(\tau)$ the prior distributions on the rotation and precision and $p(\tilde{\boldsymbol{r}}_1 | \boldsymbol{\theta}, \tau)$ the likelihood of the data. Bayes' rule is written without the proportionality constant because this integral is often not tractable. Numerical sampling methods like Markov Chain Monte Carlo (MCMC) can be used to draw samples from the full posterior. Given enough samples, the posterior mean can be approximated by the mean of these samples and can be used as point estimate $\hat{\boldsymbol{\theta}}_{\text{BIK}}$. The various representations of $\boldsymbol{\theta}$ lead to different BIK models and are explained below.

[2] $[.]_\times$ is the skew-symmetric operator giving the 3×3 matrix of the vector product

[3] In the software package that we used (PyMC2), the normal distribution is not characterized by the variance σ^2 but by the precision: $\tau = \frac{1}{\sigma^2}$ (inverse of the variance).

3.1 Common parts of all MCMC BIK models

Each BIK model is comprised of a forward model, a likelihood for the data and a (set of) prior(s). The forward model predicts the values of the final coordinates given the noisy initial positions and the current value $\boldsymbol{\theta}$ that was sampled in the chain. For the likelihood a normal distribution was used, centered at the predicted value from the forward model and with a precision τ. For the prior on the measurement precision, we used a Gamma prior with two small constants, reflecting weak prior information on the true values. The constants were chosen in preliminary simulations that yielded good results (using a completely uninformative Gamma prior could be done by setting both constants to 0). The Gamma prior has a peak at zero, acting therefore as a shrinkage prior, directing the estimation of τ to zero (low precision). This resulted in better estimates because with only 4 markers (12 data points), the precision is hard to estimate accurately. The specific form of the prior on the rotation $p(\boldsymbol{\theta})$ is specified below for each of the different $SO(3)$ parameterizations. In each case, we used vague uninformative priors, indicating no *a-priori* knowledge of the rotation.

This model structure is summarized in the directed acyclic graph of Figure 1.

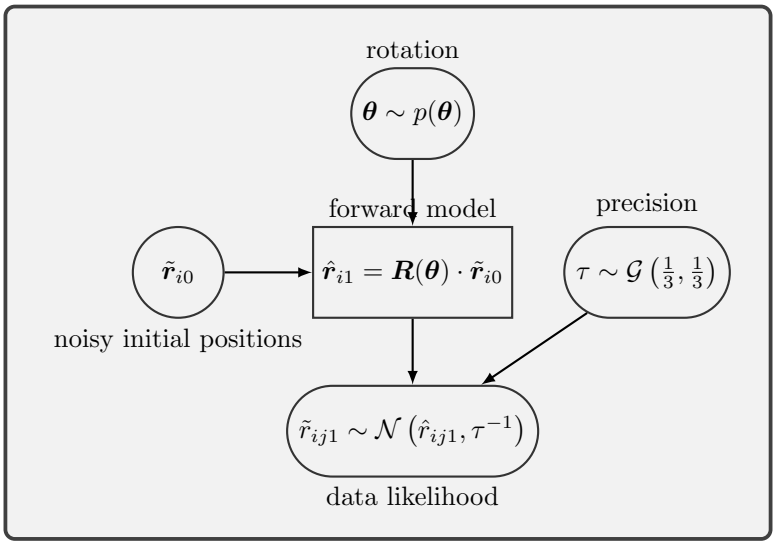

Figure 1: *Directed acyclic graph of the generic BIK model. Stochastic variables are denoted by rounded rectangles (noisy final coordinates, rotation, precision), constants as circles (noisy initial positions) and deterministic random variables as rectangles (forward kinematics model).*

To sample from the posterior distribution, Markov Chain Monte Carlo (MCMC) was used in the form of an adaptive Metropolis-Hastings algorithm. The parameters of this algorithm are given below; in preliminary simulation experiments we found these values to be stable and to provide good levels of accuracy.

- 150.000 iterations
- 50.000 burn-in
- thinning rate of 5
- updating the estimated covariance matrix every 10.000 iterations
- starting point for $\boldsymbol{\theta}$: SVD estimate
- starting point for τ: true value (in laboratory settings, a good initial guess for τ can be obtained via dynamic trials with an object with known properties like a calibration wand)

3.2 MCMC model using the rotation vector

For the rotation vector formalism (axis-angle), we specified separate priors for the unit vector of the axis (multivariate standard normal) and the angle (uniform). A normalization was done before passing it to the forward model.

$$\begin{aligned} p(\boldsymbol{u'}) &= \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_3) \qquad \boldsymbol{u} = \boldsymbol{u'}/|\boldsymbol{u'}| \\ p(\theta) &= \mathcal{U}(0, \pi) \end{aligned} \tag{18}$$

In preliminary simulations, we tested also another model where we used uniform priors on the azimuthal and polar angles of the unit vector and transformed these to cartesian coordinates of. This model yielded very similar results and was therefore not used in the final simulation experiment.

3.3 MCMC model using Euler and Cardan angles

Uniform priors were assigned to the Euler and Cardan angles according to their specific ranges (α, β, γ correspond to the angles of the first, second and third rotations respectively):

$$\begin{aligned} p(\alpha) &= \mathcal{U}(\pm\pi) \\ p(\beta) &= \mathcal{U}(0, \pi) \quad \text{Euler} \\ p(\gamma) &= \mathcal{U}(\pm\pi) \end{aligned} \tag{19}$$

$$\begin{aligned} p(\alpha) &= \mathcal{U}(\pm\pi) \\ p(\beta) &= \mathcal{U}\left(\pm\frac{\pi}{2}\right) \quad \text{Cardan} \\ p(\gamma) &= \mathcal{U}(\pm\pi) \end{aligned} \tag{20}$$

3.4 MCMC model using unit quaternions

A multivariate standard normal prior was given to the unnormalized quaternion. A normalization was done before passing it to the forward model.

$$p(q') = \mathcal{N}(\mathbf{0}, \boldsymbol{I}_4) \qquad q = q'/|q'| \tag{21}$$

This normalization could be avoided by using a Von Mises-Fisher prior on the S^3 hypershpere . In preliminary simulation experiments we tested this option (using a concentration of zero, i.e. uniform) and found similar accuracy levels to the model using the normal prior. While the Von Mises-Fisher prior avoids the need for normalization, computation time was slightly higher because the sampling takes more time than with the normal prior.

3.5 MCMC model using the rotation matrix

In theory, the rotation matrix representation could also be used in MCMC-based BIK. The matrix-Von-Mises-Fisher distribution could be used as a prior distribution on $\boldsymbol{R}$ directly, but because this distribution is not proper, the posterior distribution is not proper either. In initial tests, the MCMC did not converge to a good solution using this representation (code in the package should not be used).

4 Variational Bayesian Inverse Kinematics (VBIK)

Whereas full Bayesian inference uses numerical sampling methods as MCMC to describe the posterior, Variational Bayesian inference uses an approximation to the true posterior that can be obtained using iterative optimization which is computationally much faster. Generally, this approximation comes at a cost of less accuracy (especially in the tails of the posterior). In the main paper we showed that this lower accuracy is not an issue for inverse kinematics (point estimation). The approximation works only for models where the likelihood comes from the exponential family of distributions and for conjugate priors. In BIK we always worked with a Normal likelihood, which is from the exponential family. A conjugate prior is a prior which, in combination with the likelihood gives a posterior that is of the same family. For a Normal likelihood with unknown mean and precision, the conjugate prior is a Normal and Gamma prior on the mean and precision respectively. This is akin to the BIK model using the unit quaternion representation. Other $SO(3)$ representations are impossible to model using conjugate Normal priors as the parameters of these representations are restricted to specific ranges and cannot be normalized (Euler/Cardan angles, angle of the rotation vector).

Chappell et al. (2009) described the following method for non-linear forward models using a Variational Bayes algorithm that can be directly applied to IK. The true posterior $p(q, \tau|\tilde{\boldsymbol{r}}_1)$ is approximated by a factorized posterior distribution $p'(q, \tau|\tilde{\boldsymbol{r}}_1) = p'(q|\tilde{\boldsymbol{r}}_1) \cdot p'(\tau|\tilde{\boldsymbol{r}}_1)$ that can be obtained iteratively. The same linearization using a first-order Taylor series of the forward model as with NLLS is done to make the expectations tractable. The following priors are assigned to q and τ:

$$\begin{aligned} p'(q) &= \mathcal{N}\left(m_0, \boldsymbol{\Lambda}_0^{-1}\right) \qquad (m_0 = \hat{q}_{\text{SVD}}, \boldsymbol{\Lambda}_0 = 1 \times 10^{-10}\boldsymbol{I}_4) \\ p'(\tau) &= \mathcal{G}(s_0, c_0) \qquad (s_0 = c_0 = 1/3) \end{aligned} \tag{22}$$

leading to the following approximate posteriors:

$$\begin{aligned} p'(q|\tilde{\boldsymbol{r}}_1) &= \mathcal{N}\left(m, \boldsymbol{\Lambda}^{-1}\right) \\ p'(\tau|\tilde{\boldsymbol{r}}_1) &= \mathcal{G}(s, c) \end{aligned} \tag{23}$$

of which the parameters $m, \mathbf{\Lambda}, s, c$ can be derived via the following Expectation-Maximization (EM) scheme (Chappell et al., 2009):

$$\begin{aligned} c &= \frac{3n}{2} + c_0 \\ \mathbf{\Lambda} &= sc\left(\boldsymbol{J}^T\boldsymbol{J}\right) + \mathbf{\Lambda}_0 \\ m^{(t+1)} &= \mathbf{\Lambda}^{-1}\left(sc\boldsymbol{J}^T\left(\boldsymbol{k} + \boldsymbol{J}m^{(t)}\right) + \mathbf{\Lambda}_0 m_0\right) \\ s &= \left(\frac{1}{s_0} + \frac{1}{2}\boldsymbol{k}^T\boldsymbol{k} + \frac{1}{2}\text{trace}\left(\mathbf{\Lambda}^{-1}\boldsymbol{J}^T\boldsymbol{J}\right)\right)^{-1} \end{aligned} \tag{24}$$

(iteration indices t are only shown for m as the other parameters are not dependent on their previous values). The EM-algorithm was initiated using the parameter values of the prior: $m^{(0)} = \hat{q}_{\text{SVD}}, \mathbf{\Lambda}^{(0)} = \mathbf{\Lambda}_0, s^{(0)} = s_0$. The convergence of the variational posterior to the true posterior is monitored via the free energy F (= lower bound of the model evidence):[4]

$$\begin{aligned} F = &-\frac{sc}{s_0} + \left(\frac{3n}{2} + c_0 - 1\right)\left[\log(s) + \psi(c)\right] \\ &- \frac{1}{2}\left[(m - m_0)^T\mathbf{\Lambda}_0(m - m_0) + \text{trace}\left(\mathbf{\Lambda}^{-1}\mathbf{\Lambda}_0\right)\right] \\ &- \frac{1}{2}\left[\boldsymbol{k}^T\boldsymbol{k} + \text{trace}\left(\mathbf{\Lambda}^{-1}\boldsymbol{J}^T\boldsymbol{J}\right)\right] - s\log(c) - \log\left(\Gamma(c)\right) \\ &- c + \left(\frac{3n}{2} + c - 1\right)\left[\log(s) + \psi(c)\right] + \frac{1}{2}\log\left(\det(\mathbf{\Lambda})\right) + \text{constant} \end{aligned} \tag{25}$$

Maximizing the free energy is equivalent to minimizing the Kullback-Leibler distance between the true and approximated posteriors (Chappell et al., 2009). In the simulations, we considered the algorithm to be converged when the free energy changed less than 0.001% between successive iterations.

Chappell et al. (2009) show that this Variational solution gives the exact same point estimates as NLLS in the absence of prior information. In the simulations, we used a weakly informative prior on the rotation (normal prior centered at the SVD estimate and using a very small but non-zero precision) and a shrinkage prior on the precision, yielding slightly better results than NLLS at the lowest noise levels.

5 Overview

An overview of the different IK methods can be found in Table 1. Note that within the general classification of the methods, other algorithms are available than the ones used here.

Table 1: *Overview of IK methods' classification and algorihtms.*

abbreviation	reference	general classification	algorithm
SVD	Soderkvist and Wedin (1993)	linear least sqauares	singular value decomposition
lsQUAT	Horn (1987)	linear least squares	eigenvalue decomposition
NLLS	Chapell et al. (2009)	non-linear least squares	Gauss-Newton iteration
VBIK	Chapell et al. (2009)	approximate Bayesian	Variational inference (Expectation Maximization)
BIK	Pataky et al. (2019)	full Bayesian	adaptive Metropolis-Hastings MCMC

References

See main paper

[4] In the free energy equation $\Gamma(.)$ and $\psi(.)$ are the gamma and di-gamma functions respectively