

# SeqIKPy: a Python package for inverse kinematics in insects

Pembe Gizem Özdil<sup>1,2</sup>, Chuanfang Ning<sup>2</sup>, Auke Ijspeert<sup>2</sup>, and Pavan Ramdya<sup>1</sup>

<sup>1</sup> Neuroengineering Laboratory, Brain Mind Institute, EPFL, Lausanne, Switzerland <sup>2</sup> Biorobotics Laboratory, Institute of Bioengineering, EPFL, Lausanne, Switzerland

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

## Software

- [Review](#)
- [Repository](#)
- [Archive](#)

Editor: [Olexandr Kononov](#)

## Reviewers:

- [@aceglia](#)
- [@davidpagnon](#)

Submitted: 22 January 2025

Published: unpublished

## License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

## Summary

SeqIKPy is a Python package for inverse kinematics (IK) calculation in animal bodies with complex joint configurations. The name stands for Sequential Inverse Kinematics in Python, as our method computes joint angles sequentially by performing IK for each joint along a kinematic chain.

Our framework contains:

- Pose alignment: map tracked key point locations in 3D onto an animal body template.
- Inverse kinematics: calculate joint angles sequentially from 3D poses.
- Visualization: plot and animate the results in 3D.

SeqIKPy is aimed at researchers studying detailed joint motion in animals with complex, multiple degrees-of-freedom body appendages. We provide examples for the fruit fly, *Drosophila melanogaster*. However, each module can easily be extended to be used with another model organism; the only requirements are the 3D kinematics of the target animal and its corresponding kinematic chain. Our package requires minimal Python knowledge and we provide extensive tutorials at <https://nely-epfl.github.io/sequential-inverse-kinematics>.

## Statement of need

Over the past decade, deep-learning based computer vision algorithms have transformed the analysis of behaviors in laboratory animals (Pereira et al., 2020), including for the widely-used model organism, *Drosophila melanogaster*. Recently, researchers have developed deep learning-based 3D pose estimation tools (Günel et al., 2019; Karashchuk et al., 2021) and detailed biomechanical models (Lobato-Rios et al., 2022; Vaxenburg et al., 2024; Wang-Chen et al., 2024), creating a growing need for tools to obtain more detailed descriptions of how body parts move in joint space. These computed joint angles can be replayed in physics-based simulations to estimate unmeasured physical quantities like joint torques (Lobato-Rios et al., 2022).

Inverse Kinematics (IK) spans multiple domains including robotics, biomechanics, and character animation (Aristidou et al., 2018). In robotics, IK typically computes joint angles to achieve a desired end-effector position while respecting joint constraints. By contrast, in biomechanics, IK algorithms calculate joint angles to track all marker positions rather than only a single end-effector. This process is also known as multi-body kinematics optimization and is a well-established area in human biomechanics research (Begon et al., 2018; Delp et al., 2007; Pagnon et al., 2022; Werling et al., 2023).

However, multi-body kinematics optimization is still an emerging field in insect research. Existing methods fall on two extremes along a spectrum of complexity. Simple approaches

have been employed to compute joint angles using the dot product of two consecutive body segment vectors (Karashchuk et al., 2021; Lobato-Rios et al., 2022). This often results in deviations from reference poses due to the lack of iterative corrections. On the other hand, more advanced gradient-based optimization methods have been developed to estimate joint angles in a biomechanical model of the fly (Vaxenburg et al., 2024), using simulation-based Jacobians in physics-engines like MuJoCo (Todorov et al., 2012). Although these methods provide higher accuracy, they require complex setups that are not easily accessible to all researchers. Therefore, there is no stand-alone inverse kinematics software satisfying the trade-off between accuracy and complexity.

To address this gap, we have developed SeqIKPy, a fast and lightweight Python package for multi-body kinematics optimization in insects. Our package has two main stages: marker registration (aligning joint markers to a template in 3D) and inverse kinematics. The second stage draws upon the open-source IKPy (Manceron, 2016) library in a sequential manner. Through 3D visualizations, we demonstrate that our package can reliably reconstruct body kinematics for various fruit fly behaviors. Furthermore, recent studies have shown that the joint angles computed using SeqIKPy can accurately replicate animal behavior, both ground walking (Wang-Chen et al., 2024) and grooming (Özdil et al., 2024), in physics-based simulations. Although our examples focus on the fly, our package's modular design allows for customization and extension to other animals with a similar body morphology.

## Overview

SeqIKPy assumes that the 3D pose estimation has the following orientation (Figure 1, left):

- x-axis: anterioposterior axis
- y-axis: mediolateral axis
- z-axis: dorsoventral axis

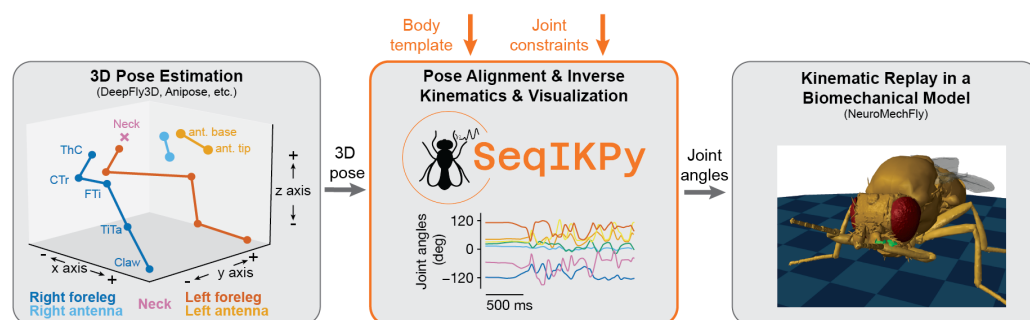
After setting this orientation, users can use the AlignPose class to map body keypoints to a template body model (Figure 1, middle). Despite being optional for inverse kinematics, this step has two benefits:

- It aligns measured kinematics to a standardized body template, facilitating replay of behaviors in body models (Figure 1, right).
- It reduces noise and variation in kinematics by standardizing body lengths.

We provide a default body template based on a CT scan of the fly (Lobato-Rios et al., 2022). Users can also define custom templates manually or by importing SDF files. Utility functions are included to convert data into the required formats.

Next, the KinematicChainSeq class defines a pre-configured kinematic chain for fly legs. Users need a dictionary containing segment lengths and joint bounds (Figure 1, middle). Segment lengths can be derived from 3D kinematics, while joint bounds are optional. Using the defined kinematic chain, the LegInvKin class calculates joint angles sequentially for each leg. This process supports parallelization to perform IK on multiple legs simultaneously. Additionally, our package contains features for animation and visualization of data in 3D. For more technical details on the implementation, please refer to the methodology section at <https://nely-epfl.github.io/sequential-inverse-kinematics>.

On a MacBook Pro with a 2.3 GHz Quad-Core Intel Core i7, the pipeline computes inverse kinematics for all six legs across 100 frames in 36 s when parallelized.



**Figure 1:** Overview of the SeqIKPy pipeline. **(Left)** Pose estimation tools (e.g., DeepFly3D and Anipose) estimates 3D positions of body key points. **(Middle)** SeqIKPy aligns these points to a body template, performs sequential inverse kinematics using joint constraints to calculate joint angles. **(Right)** The computed joint angles can be used to replay measured motions in biomechanical models, such as NeuroMechFly.

## Acknowledgements

We acknowledge the contributors and maintainers of the open-source software tools that SeqIKPy builds upon, including Python, NumPy, SciPy, Matplotlib, and IKPy (Manceron, 2016), among others. PGÖ acknowledges support from a Swiss Government Excellence Scholarship for Doctoral Studies and a Google PhD Fellowship. PR acknowledges support from an SNSF Project Grant (175667) and an SNSF Eccellenza Grant (181239).

## References

- Aristidou, A., Lasenby, J., Chrysanthou, Y., & Shamir, A. (2018). Inverse kinematics techniques in computer graphics: A survey. *Computer Graphics Forum*, 37, 35–58. <https://doi.org/10.1111/cgf.13310>
- Begon, M., Andersen, M. S., & Dumas, R. (2018). Multibody kinematics optimization for the estimation of upper and lower limb human joint kinematics: A systematized methodological review. *Journal of Biomechanical Engineering*, 140(3), 030801. <https://doi.org/10.1115/1.4038741>
- Delp, S. L., Anderson, F. C., Arnold, A. S., Loan, P., Habib, A., John, C. T., Guendelman, E., & Thelen, D. G. (2007). OpenSim: Open-source software to create and analyze dynamic simulations of movement. *IEEE Transactions on Biomedical Engineering*, 54(11), 1940–1950. <https://doi.org/10.1109/TBME.2007.901024>
- Günel, S., Rhodin, H., Morales, D., Campagnolo, J., Ramdya, P., & Fua, P. (2019). DeepFly3D, a deep learning-based approach for 3D limb and appendage tracking in tethered, adult *Drosophila*. *eLife*, 8, e48571. <https://doi.org/10.7554/eLife.48571>
- Karashchuk, P., Rupp, K. L., Dickinson, E. S., Walling-Bell, S., Sanders, E., Azim, E., Brunton, B. W., & Tuthill, J. C. (2021). Anipose: A toolkit for robust markerless 3D pose estimation. *Cell Reports*, 36(13), 109730. <https://doi.org/10.1016/j.celrep.2021.109730>
- Lobato-Rios, V., Ramalingasetty, S. T., Özdil, P. G., Arreguit, J., Ijspeert, A. J., & Ramdya, P. (2022). NeuroMechFly, a neuromechanical model of adult *Drosophila melanogaster*. *Nature Methods*, 19(5), 620–627. <https://doi.org/10.1038/s41592-022-01466-7>
- Manceron, P. (2016). IKPy. In *GitHub repository*. GitHub. <https://doi.org/10.5281/zenodo.6551105>
- Özdil, P. G., Arreguit, J., Scherrer, C., Ijspeert, A., & Ramdya, P. (2024). Centralized brain

- 113 networks underlie body part coordination during grooming. *bioRxiv*. <https://doi.org/10.1101/2024.12.17.628844>  
114
- 115 Pagnon, D., Domalain, M., & Reveret, L. (2022). Pose2Sim: An open-source python package  
116 for multiview markerless kinematics. *Journal of Open Source Software*, 7(77), 4362.  
117 <https://doi.org/10.21105/joss.04362>
- 118 Pereira, T. D., Shaevitz, J. W., & Murthy, M. (2020). Quantifying behavior to under-  
119 stand the brain. *Nature Neuroscience*, 23(12), 1537–1549. <https://doi.org/10.1038/s41593-020-00734-z>  
120
- 121 Todorov, E., Erez, T., & Tassa, Y. (2012). Mujoco: A physics engine for model-based control.  
122 *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 5026–5033.  
123 <https://doi.org/10.1109/IROS.2012.6386109>
- 124 Vaxenburg, R., Siwanowicz, I., Merel, J., Robie, A. A., Morrow, C., Novati, G., Stefanidi, Z.,  
125 Card, G. M., Reiser, M. B., Botvinick, M. M., Branson, K. M., Tassa, Y., & Turaga, S. C.  
126 (2024). *Whole-body simulation of realistic fruit fly locomotion with deep reinforcement*  
127 *learning*. <https://doi.org/10.1101/2024.03.11.584515>
- 128 Wang-Chen, S., Stimpfling, V. A., Lam, T. K. C., Özdil, P. G., Genoud, L., Hurtak, F., &  
129 Ramdya, P. (2024). NeuroMechFly v2: Simulating embodied sensorimotor control in adult  
130 drosophila. *Nature Methods*, 1–10. <https://doi.org/10.1038/s41592-024-02497-y>
- 131 Werling, K., Bianco, N. A., Raitor, M., Stingel, J., Hicks, J. L., Collins, S. H., Delp, S. L.,  
132 & Liu, C. K. (2023). AddBiomechanics: Automating model scaling, inverse kinematics,  
133 and inverse dynamics from human motion data through sequential optimization. *Plos One*,  
134 18(11), e0295152. <https://doi.org/10.1371/journal.pone.0295152>