Multidisciplinary Biomechanics Journal

Inverse Kinematics of the Closed-Loop Shoulder: Quadratic Programming is Faster than Interior-Point Methods with Equivalent Accuracy

Pierre Puchaud[a*], Mégane Millan[b], Lucas Joseph[a]

a AUCTUS, Inria, Bordeaux, France
b WILLOW, Inria, Paris, France
* Corresponding author: pierre.puchaud@inria.fr

Received date: 16/01/2026
Accepted date: 16/03/2026
Publication date: 02/06/2026

Keywords: quadratic progamming, inverse kinematics, close-loop, constraints, optimization

1. Introduction

Shoulder girdle kinematics reconstruction is special due to the closed-loop nature of the osteoarticular mechanism. While open-loop tree structures can be solved using non-linear least squares, anatomically plausible shoulder models often require holonomic constraints. Current modeling propositions for the scapulothoracic contact vary from single-point contact on ellipsoids (Seth et al. 2019) to multi-point constraints (Naaim et al. 2017). These modeling choices may introduce a kinematic loop, specifically in between three bone-segments the clavicle, scapula, and the thorax, that must be enforced strictly during Inverse Kinematics (IK).

In the presence of holonomic constraints, the IK problem shifts from unconstrained optimization to a constrained non-linear programming (NLP) problem. A common approach is to use robust, general-purpose such as Interior-Point (IP) methods ,such as (Waechter & Biegler, 2006). However, these solvers can be computationally expensive and may be overkill for tracking tasks where the pose changes incrementally. Alternatively, Differential Inverse Kinematics (DIK) allows the problem to be formulated as a Quadratic Program (QP) by linearizing the constraints at the velocity level (Duclusaud et al. 2025). While QP is standard for real-time control in robotics, its suitability for biomechanics remains unproven, particularly regarding its ability to handle inevitable residual errors caused by Soft Tissue Artifact and model scaling mismatches compared to global IP methods. The aim of this study is to compare these two approaches (IP vs. QP) applied to a closed-loop upper limb model.

2. Methods

2.1 Model and data

We utilized experimental data from Seth et al. (2019), consisting of 18 motion capture files (shoulder flexion, abduction, and shrugging). The scaled model representing a 26-year-old healthy female subject (height: 162 cm, weight: 52 kg), had a 6-DOF floating base, a 2-DOF sternoclavicular joint, a 4-DOF thoracoscapular joint including an ellipsoid mobilizer, a 3-DOF glenohumeral joint. The model is defined by generalized coordinates $\mathbf{q}$ and parsed with the Pinocchio library (Carpentier et al., 2019). The loop closure constraint $c_{AC}(\mathbf{q})$ at the acromioclavicular (AC) joint, enforcing that the distal clavicle acromial joint location p_{AC}^{C} coincides with its scapula equivalent p_{AC}^{S} is defined as:

$$c_{AC}(\mathbf{q}) = p_{AC}^{S}(\mathbf{q}) - p_{AC}^{C}(\mathbf{q}) = 0$$

2.2 Non-linear programming (Interior-Point)

Using CasADi SX with the IPOPT solver, we formulated the standard IK problem, as finding the configuration $\mathbf{q}$ that minimizes the weighted squared distance between experimental markers $\mathbf{x}_i^{exp}$ and model markers $\mathbf{x}_i^{model}$, subject to the hard loop closure constraint:

To cite this article: Pierre Puchaud, Mégane Millan, Lucas Joseph, Inverse Kinematics of the Closed-Loop Shoulder: Quadratic Programming is Faster than Interior-Point Methods with Equivalent Accuracy, *Multidisciplinary Biomechanics Journal*, 2026, vol. 3, p. 73–75, DOI: 10.46298/mbj.17921.

$$\min_{\mathbf{q}} \frac{1}{2}\sum_{i=1}^{M} w_i \left\| \mathbf{x}_i^{exp} - \mathbf{x}_i^{model}(\mathbf{q}) \right\|^2$$

$$s.t. \; c_{AC}(\mathbf{q}) = 0$$

with weights being {100, 100, 100, 10, 10, 10, 2, 2, 2} corresponding to the thoracic landmarks (centpxt8, centijc7, ij), humeral markers (gu, centelbow, EpL), and scapular markers (ts, aa, ai), the same markers weight were used in next section. They were fine-tuned to guarantee the thorax to remain static and to guarantee the scapula visually match its markers.

2.3 Differential Inverse Kinematics

Alternatively, we formulate the problem at the velocity level using DIK. Instead of solving for absolute position **q**, we solve for a small increment **Δq** that minimize tracking error while satisfying linearized constraints. We linearize the marker error and constraints with relationship between task space velocity and joint space velocity using well-known analytical jacobians:

$$\mathbf{x}_i^{model}(\mathbf{q}) \approx \mathbf{x}_i^{model}(\mathbf{q_0}) + \mathcal{J}_{i,q}(\mathbf{q_0})\Delta\mathbf{q}$$

and,

$$c_{AC}(\mathbf{q}) \approx p_{AC}^{S}(\mathbf{q_0}) - p_{AC}^{C}(\mathbf{q_0}) + \left(\mathcal{J}_{p_{AC}^{S}}(\mathbf{q_0}) - \mathcal{J}_{p_{AC}^{C}}(\mathbf{q_0})\right)\Delta\mathbf{q}$$

This formulation results in a convex QP which can be solved extremely efficiently compared to the non-linear IP formulation, using proxQP solver. As the constraints are linearized through an initial pose, we allow up to 10 executions of the QP if the nonlinear constraint is not satisfied at 1e-6 mm.

$$\min_{\Delta\mathbf{q}} \frac{1}{2}\sum_{i=1}^{M} w_i \left\| \mathbf{x}_i^{exp} - \left(\mathbf{x}_i^{model}(\mathbf{q_0}) + \mathcal{J}_{i,q}(\mathbf{q_0})\Delta\mathbf{q}\right) \right\|^2$$

$$s.t. \, p_{AC}^{S}(\mathbf{q_0}) - p_{AC}^{C}(\mathbf{q_0}) + \left(\mathcal{J}_{p_{AC}^{S}}(\mathbf{q_0}) - \mathcal{J}_{p_{AC}^{C}}(\mathbf{q_0})\right)\Delta\mathbf{q} = 0$$

3. Results and Discussion

Both solvers were used through Python. The solvers were tested on all 18 trials (Flexion, Abduction, Shrugging). **Computation Time.** The QP solver demonstrated a speedup factor of approximately 83x in average. The average computation time per frame was 0.40 ms (± 0.1 ms) for the QP solver compared to 36.2 ms (± 4.5 ms) for the Interior-Point method. For specific tasks like Abduction (ABD01), the QP solver processed the full 10.9s trial in 0.43s, whereas IPOPT required 35.9s, see Table 1. Despite the linearization, the QP approach maintained high accuracy comparable to the non-linear solver. **Marker Residuals:** For typical trials (e.g., ABD01), the mean marker residual was 22.02 mm for both QP and IP solvers, with differences appearing only at the sub-micrometer level. This residual magnitude can be attributed to both the model minimally pre-scaled and to the marker trajectories are not gold standard data, bu recomputed from digitized locations of bony landmarks. **Constraint Violation:** The kinematic loop constraint error (AC joint gap) was respected below 1e-7 mm by IPOPT. For the QP solver, it remained effectively zero (mean 0.0003 mm, max 0.0015 mm across abduction trials). This confirms that the QP formulation effectively keeps the violation below the optical motion capture noise, with sufficient precision for use in forward dynamics. Despite all the code was written in Python (but interfaced with compiled library), the code was already performant, but we would win additional computation time using IPOPT and proxQP in a compiled language too.

Table 1. Comparison of solver performance on representative trials.

Trial	Frames	QP Time (s)	IP Time (s)	Speedup (x)
ABD01	1091	0.44	35.98	82.5
FLX01	794	0.44	31.87	72.7
SHRUG01	462	0.17	17.44	102.9

Note: Res. = Mean Marker Residual.

4. Conclusion

We compared a Quadratic Programming formulation against a standard Interior-Point method for solving inverse kinematics of a closed-loop shoulder model. Our results indicate that QP is as accurate as Interior-Point methods for this application but is significantly faster (approx. 83 times). The primary challenge with QP in biomechanics is handling the drift associated with linearization; however, our results show the constraint violation remains below sub-millimetric precision. This finding implies that complex, anatomically realistic shoulder models and other similar joints such as knee, talo-crural joint with multiple holonomic constraints can be solved in real-time. This efficiency opens new avenues for real-time applications and large-scale dataset processing without the computational penalty usually associated with constrained non-linear programming.

Acknowledgements

The author thank J. Carpentier, J. Vaillant for helping on implementing the ellipsoid joint in Pinocchio, and G. Passault, and A. Schortgen for their helpful discussion on Differential Inverse Kinematics.

Contributor Roles

PP: Conceptualization, Methodology, Software, Formal Analysis, Writing – original draft. MM : Methodology, Software. LJ : Software.

References

Carpentier, J., Saurel, G., Buondonno, G., Mirabel, J., Lamiraux, F., Stasse, O., & Mansard, N. (2019). The Pinocchio C++ library: A fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives. *2019 IEEE/SICE International Symposium on System Integration (SII)* (pp. 614–619). https://doi.org/10.1109/SII.2019.8700380

Duclusaud, M., Passault, G., Padois, V., & Ly, O. (2025). PlaCo: a QP-based robot planning and control framework. arXiv preprint arXiv:2511.06141. https://doi.org/10.48550/arXiv.2511.06141

Naaim, A., Moissenet, F., Duprey, S., Begon, M., & Cheze, L. (2017). Effect of various upper limb multibody models on soft tissue artefact correction: A case study. *Journal of biomechanics*, *62*, 102–109. https://doi.org/10.1016/j.jbiomech.2017.01.031

Seth, A., Dong, M., Matias, R., & Delp, S. (2019). Muscle contributions to upper-extremity movement and work from a musculoskeletal model of the human shoulder. *Frontiers in Neurorobotics*, *13*, 90. https://doi.org/10.3389/fnbot.2019.00090

Waechter, A., & Biegler, L. T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, *106*(1), 25–57. https://doi.org/10.1007/s10107-004-0559-y