Algorithm 1 Inverse Kinematics with Redundancy Handling and Joint Limits

---

Require: $\mathbf{x}_{target}$: Target position and quaternion orientation

    $\mathbf{q}_{seed}$: Initial joint configuration (seed)

    $\mathbf{q}_{min}$: Vector of lower joint limits

    $\mathbf{q}_{max}$: Vector of upper joint limits

    ϵ: Tolerance for convergence

    λ: Damping factor

    α: Secondary objective gain

1: Initialize $\mathbf{q} \leftarrow \mathbf{q}_{seed}$

2: $\lambda \leftarrow 1 \times 10^{-6}$

3: $\alpha \leftarrow 0.01$

4: **for** $i = 1$ to maxIterations **do**

5: Compute forward kinematics $\mathbf{T} \leftarrow \mathbf{T}(\mathbf{q})$

6: Compute Jacobian $\mathbf{J} \leftarrow \mathbf{J}(\mathbf{q})$

7: Compute error $\mathbf{e} \leftarrow \mathbf{x}_{target} - \mathbf{x}(\mathbf{T})$

8: **if** $\|\mathbf{e}\| < \epsilon$ **then**

9:   **return q**

10: **end if**

11: Compute damped pseudo-inverse $\mathbf{J}_{\text{pinv}} \leftarrow \mathbf{J}^T(\mathbf{J}\mathbf{J}^T + \lambda\mathbf{I})^{-1}$

12: Compute null-space projection matrix $\mathbf{N} \leftarrow \mathbf{I} - \mathbf{J}_{\text{pinv}}\mathbf{J}$

13: Compute secondary objective:

14: $\mathbf{q}_{mid} \leftarrow \frac{\mathbf{q}_{max} + \mathbf{q}_{min}}{2}$

15: $\mathbf{q}_{diff} \leftarrow \mathbf{q} - \mathbf{q}_{mid}$

16: $\mathbf{q}_{sec} \leftarrow -\alpha\mathbf{q}_{diff}$

17: Compute joint update $\Delta\mathbf{q} \leftarrow \mathbf{J}_{\text{pinv}}\mathbf{e} + \mathbf{N}\mathbf{q}_{sec}$

18: Update joint angles $\mathbf{q} \leftarrow \mathbf{q} + \Delta\mathbf{q}$

19: Enforce joint limits:

20: **for** $j = 1$ to size($\mathbf{q}$) **do**

21:   **if** $\mathbf{q}[j] < \mathbf{q}_{min}[j]$ **then**

22:    $\mathbf{q}[j] \leftarrow \mathbf{q}_{min}[j]$

23:   **end if**

24:   **if** $\mathbf{q}[j] > \mathbf{q}_{max}[j]$ **then**

25:    $\mathbf{q}[j] \leftarrow \mathbf{q}_{max}[j]$

26:   **end if**

27: **end for**

28: **end for**

29: Print "Max number of iterations reached. Solution found is proximal."

30: **return q**