2 Numerical Inverse Kinematics

- Iterative numerical methods can be applied if the IK equations do not admit analytic solutions.
- Even in cases where an analytic solution does exist, numerical methods are often used to improve the accuracy of these solutions.
- There exist a variety of iterative methods for finding the roots of a nonlinear equation, and our aim is to develop ways in which to transform the IK equations so that they become amenable to existing numerical methods.
- An approach fundamental to nonlinear root-finding will be Newton-Raphson method.
- We seek the closest approximate solution; or, conversely, an infinity of IK solutions exists (i.e., if the robot is kinematically redundant) and we seek a solution that is optimal with respect to some criterion.

2.1 Newton-Raphson Method

- To solve the equation $g(\theta) = 0$ numerically for a given differentiable function $g : \Re^n \to \Re^n$, assume $\theta^0 \in \Re^n$ is an initial guess for the solution.

- Write the Taylor expansion of $g(\theta)$ at $\theta = \theta^0$ and truncate it at first order:

$$g(\theta) = g(\theta^0) + \frac{\partial g}{\partial \theta^T}(\theta^0)(\theta - \theta^0) + h.o.t \qquad \text{where} \qquad \frac{\partial g}{\partial \theta^T}(\theta^0) = \frac{\partial \begin{bmatrix} g_1 \\ \vdots \\ g_n \end{bmatrix}}{\partial \begin{bmatrix} \theta_1 & \dots & \theta_n \end{bmatrix}} = \begin{bmatrix} \frac{\partial g_1}{\partial \theta_1}(\theta) & \cdots & \frac{\partial g_1}{\partial \theta_n}(\theta) \\ \vdots & & \vdots \\ \frac{\partial g_n}{\partial \theta_1}(\theta) & \cdots & \frac{\partial g_n}{\partial \theta_n}(\theta) \end{bmatrix}$$

- Keeping only the terms up to first order, set $g(\theta) = 0$ and solve for θ to obtain

$$\theta = \theta^0 - \left(\frac{\partial g}{\partial \theta^T}(\theta^0) \right)^{-1} g(\theta^0)$$

- Using this value of θ as the new guess for the solution and repeating the above, we get the following iteration:

$$\theta^{k+1} = \theta^k - \left(\frac{\partial g}{\partial \theta^T}(\theta^k) \right)^{-1} g(\theta^k)$$

- The above iteration is repeated until some stopping criterion is satisfied.

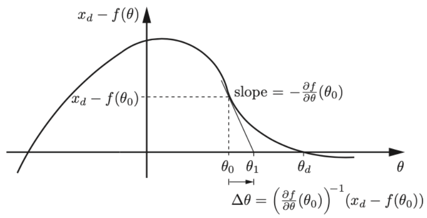

Figure 6.7: The first step of the Newton–Raphson method for nonlinear root-finding for a scalar x and θ. In the first step, the slope $-\partial f/\partial\theta$ is evaluated at the point $(\theta^0, x_d - f(\theta^0))$. In the second step, the slope is evaluated at the point $(\theta^1, x_d - f(\theta^1))$ and eventually the process converges to θ_d. Note that an initial guess to the left of the plateau of $x_d - f(\theta)$ would be likely to result in convergence to the other root of $x_d - f(\theta)$, and an initial guess at or near the plateau would result in a large initial $|\Delta\theta|$ and the iterative process might not converge at all.

2.2 Numerical Inverse Kinematics Algorithm

- For the Newton-Raphson method, let us define $g(\theta_d) = x_d - f(\theta_d)$ to find joint coordinates $\theta_d \in \Re^n$ from the desired end-effector coordinate $x_d \in \Re^m$

$$g(\theta_d) = x_d - f(\theta_d) = 0$$

- Given an initial guess θ^0 which is close to a solution θ_d, the kinematics can be expressed as the Taylor expansion

$$x_d = f(\theta_d) = f(\theta_0) + \left.\frac{\partial f}{\partial \theta^T}\right|_{\theta=\theta^0} (\theta_d - \theta^0) + h.o.t$$

- Let us define the Jacobian $J(\theta_0) = \left.\frac{\partial f}{\partial \theta^T}\right|_{\theta=\theta^0}$, then we have the approximate and iterative solution

$$\theta_d = \theta^0 + J^+(\theta_0)(x_d - f(\theta^0)) \quad \rightarrow \quad \theta^{k+1} = \theta^k + J^+(\theta_k)(x_d - f(\theta^k))$$

where $\theta^k \rightarrow \theta_d$ satisfying $x_d = f(\theta_d)$, as $k \rightarrow \infty$.

Pseudoinverse

Moore-Penrose pseudoinverse J^+ : consider the equation $z = Jy$ with $y \in \Re^n$ and $z \in \Re^m$

- J is square and full rank, J^{-1} is obtained using LU decomposition
- J is fat ($n > m$) and full rank, $J^+ = J^T(JJ^T)^{-1}$ (right inverse) minimizes the two-norm $\|y\|^2$:

$$\min \frac{1}{2} y^T y \qquad \text{subject to} \quad z = Jy$$

  The optimization brings two-norm minimum solution

$$H = \frac{1}{2} y^T y + \lambda^T (z - Jy) \qquad \frac{\partial H}{\partial y} = y - J^T \lambda = 0$$

$$z = Jy = JJ^T \lambda \qquad \lambda = (JJ^T)^{-1} z \qquad y = J^T \lambda = J^T (JJ^T)^{-1} z = J^+ z$$

  If $n > m$ then the solution is the smallest joint variable change (in the two-norm sense) that exactly satisfies Equation $z = Jy$

- J is thin (tall) ($n < m$) and full rank, $J^+ = (J^T J)^{-1} J^T$ (left inverse) minimizes the error two-norm $\|z - Jy\|^2$

$$H = \frac{1}{2} (z - Jy)^T (z - Jy) \qquad \frac{\partial H}{\partial y} = -J^T z + J^T J y = 0 \qquad y = (JJ^T)^{-1} J^T z = J^+ z$$

  If $n < m$ then the solution may not exactly satisfy Equation $z = Jy$, but it satisfies this condition as closely as possible in a least-squares sense.

Numerical IK using Newton-Raphson Method

1. Initialization: Given $x_d \in \Re^m$ and an initial guess $\theta^0 \in \Re^n$, set $i = 0$
2. Set $e = x_d - f(\theta^i)$, while $\|e\| > \epsilon$ for some small ϵ
   - Set $\theta^{i+1} = \theta^i + J^+(\theta_i)e$
   - Increment i

- To modify this algorithm to work with a desired end-effector configuration represented as $T_{sd} \in SE(3)$ instead of a coordinate vector x_d, we can replace the coordinate Jacobian J with the end-effector body Jacobian $J_b \in \Re^{6\times n}$.
- Note that the vector $e = x_d - f(\theta^i)$, representing the direction from the current guess (evaluated through the forward kinematics) to the desired end-effector configuration, cannot simply be replaced by $T_{sd} - T_{sb}(\theta^i)$; the pseudoinverse of J_b should act on a body twist $\mathcal{V}_b \in \Re^6$.
- To find the right analogy, we should think of $e = x_d - f(\theta^i)$ as a velocity vector which, if followed for unit time, would cause a motion from $f(\theta^i)$ to x_d.
- Similarly, we should look for a body twist $\mathcal{V}_b$ which, if followed for unit time, would cause a motion from $T_{sb}(\theta^i)$ to the desired configuration T_{sd}.
- To find this $\mathcal{V}_b$, we first calculate the desired configuration in the body frame,

$$T_{bd}(\theta^i) = T_{sb}^{-1}(\theta^i)T_{sd} = T_{bs}(\theta^i)T_{sd}$$

- Then $\mathcal{V}_b$ is determined using the matrix logarithm,

$$[\mathcal{V}_b] = \log T_{bd}(\theta^i).$$

This leads to the following IK algorithm, which is analogous to the above coordinate-vector algorithm:

1. Initialization: Given $T_{sd} \in SE(3)$ and an initial guess $\theta^0 \in \Re^n$, set $i = 0$
2. Set $[\mathcal{V}_b] = \log(T_{sb}^{-1}(\theta^i)T_{sd})$, while $\|\omega_b\| > \epsilon_\omega$ or $\|v_b\| > \epsilon_v$ for some small ϵ_ω, ϵ_v:
   - Set $\theta^{i+1} = \theta^i + J_b^+(\theta_i)\mathcal{V}_b$
   - Increment i

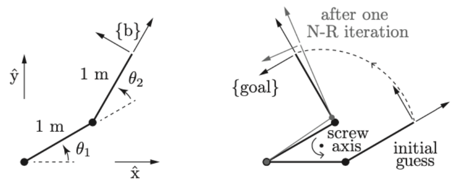

Figure 6.8: (Left) A 2R robot. (Right) The goal is to find the joint angles yielding the end-effector frame {goal} corresponding to $\theta_1 = 30°$ and $\theta_2 = 90°$. The initial guess is $(0°, 30°)$. After one Newton–Raphson iteration, the calculated joint angles are $(34.23°, 79.18°)$. The screw axis that takes the initial frame to the goal frame (by means of the curved dashed line) is also indicated.

Example 6.1. *(Planar 2R robot). Now we apply the body Jacobian Newton-Raphson IK algorithm to the 2R robot. Each link is* $1m$ *in length, and we would like to find the joint angles that place the tip of the robot at* $(x_d, y_d) = (0.366m, 1.366m)$*, which corresponds to* $\theta_d = (30°, 90°)$ *and*

$$T_{sd} = \begin{bmatrix} -0.5 & -0.866 & 0 & 0.366 \\ 0.866 & -0.5 & 0 & 1.366 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- The forward kinematics, expressed in the end-effector frame, is given by

$$M = \begin{bmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \mathcal{B}_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 2 \\ 0 \end{bmatrix} \qquad \mathcal{B}_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

- Our initial guess at the solution is $\theta^0 = (0°,\ 30°)$, and we specify an error tolerance of $\epsilon_\omega = 0.001 rad\ (or\ 0.057°)$ and $\epsilon_v = 10^{-4} m\ (100\ microns)$.

- The progress of the Newton-Raphson method is illustrated in the table below

i	$(\theta_1,\ \theta_2)$	(x, y)	$\mathcal{V}_b = (\omega_{zb}, v_{xb}, v_{yb})$	$\lVert\omega_b\rVert$	$\lVert v_b\rVert$
0	$(0.00, 30.00°)$	$(1.866, 0.500)$	$(1.571, 0.498, 1.858)$	1.571	1.924
1	$(34.23°, 79.18°)$	$(0.429, 1.480)$	$(0.115, -0.074, 0.108)$	0.115	0.131
2	$(29.98°, 90.22°)$	$(0.363, 1.364)$	$(-0.004, 0.000, -0.004)$	0.004	0.004
3	$(30.00°, 90.00°)$	$(0.366, 1.366)$	$(0.000, 0.000, 0.000)$	0.000	0.000

- The iterative procedure converges to within the tolerances after three iterations.

- The constant body velocity $\mathcal{V}_b$ that takes the initial guess to $\{$goal$\}$ in one second is a rotation about the screw axis indicated in the figure.

3 Inverse Velocity Kinematics

- One solution for controlling a robot so that it follows a desired end-effector trajectory $T_{sd}(t)$ is to calculate the IK $\theta_d(k\Delta t)$ at each discrete timestep k, then control the joint velocities $\dot{\theta}$ as follows

$$\dot{\theta} = \frac{\theta_d(k\Delta t) - \theta((k-1)\Delta t)}{\Delta t}$$

  This amounts to a feedback controller since the desired new joint angles $\theta_d(k\Delta t)$ are being compared with the most recently measured actual joint angles $\theta((k-1)\Delta t)$ in order to calculate the required joint velocities.

- Another option that avoids the computation of IK is to calculate the required joint velocities $\dot{\theta}$ directly from the relationship $\dot{\theta} = J^{+}\mathcal{V}_d$, The desired twist $\mathcal{V}_d(t)$ can be chosen to be $T_{sd}^{-1}(t)\dot{T}_{sd}(t)$ (the body twist of the desired trajectory at time t) or $\dot{T}_{sd}(t)T_{sd}^{-1}(t)$ (the spatial twist), depending on whether the body Jacobian or space Jacobian is used; however small velocity errors are likely to accumulate over time, resulting in increasing position error. Thus, a position feedback controller should choose $\mathcal{V}_d(t)$ so as to keep the end-effector following $T_{sd}(t)$ with little position error.

Pseudoinverse

The use of the pseudoinverse $J^+(\theta)$ returns joint velocities $\dot{\theta}$ minimizing the two-norm $\|\dot{\theta}\|$

$$\min \ \frac{1}{2}\dot{\theta}^T\dot{\theta} \qquad \textbf{subject to} \ \ \mathcal{V}_d = J\dot{\theta}$$

$$\begin{aligned}
H &= \frac{1}{2}\dot{\theta}^T\dot{\theta} + \lambda^T(\mathcal{V}_d - J\dot{\theta}) \\
\frac{\partial H}{\partial \dot{\theta}} &= \dot{\theta} - J^T\lambda = 0 \\
\mathcal{V}_d &= J\dot{\theta} = JJ^T\lambda \\
\lambda &= (JJ^T)^{-1}\mathcal{V}_d \\
\dot{\theta} &= J^T\lambda = J^T(JJ^T)^{-1}\mathcal{V}_d = J^+\mathcal{V}_d
\end{aligned}$$

Inertia-weighted Pseudoinverse

Let us find the joint velocities $\dot{\theta}$ minimizing the kinetic energy $\frac{1}{2}\dot{\theta}^T M(\theta)\dot{\theta}$

$$\min \ \frac{1}{2}\dot{\theta}^T M(\theta)\dot{\theta} \qquad \textbf{subject to} \ \ \mathcal{V}_d = J\dot{\theta}$$

$$\begin{aligned}
H &= \frac{1}{2}\dot{\theta}^T M(\theta)\dot{\theta} + \lambda^T(\mathcal{V}_d - J\dot{\theta}) \\
\frac{\partial H}{\partial \dot{\theta}} &= M(\theta)\dot{\theta} - J^T\lambda = 0 \\
\mathcal{V}_d &= J\dot{\theta} = JM^{-1}J^T\lambda \\
\lambda &= (JM^{-1}J^T)^{-1}\mathcal{V}_d \\
\dot{\theta} &= M^{-1}J^T\lambda = M^{-1}J^T(JM^{-1}J^T)^{-1}\mathcal{V}_d = J_M^+\mathcal{V}_d
\end{aligned}$$

where $J_M^+ = M^{-1}J^T(JM^{-1}J^T)^{-1}$

Weighted Pseudoinverse

Let us find the joint velocities $\dot{\theta}$ minimizing the kinetic energy plus the rate of change of the potential energy

$$\frac{1}{2}\dot{\theta}^T M(\theta)\dot{\theta} + \nabla h(\theta)^T \dot{\theta}$$

where $h(\theta)$ could be the gravitational potential energy, or an artificial potential function whose value increases as the robot approaches an obstacle. The rate of change of $h(\theta)$ is

$$\frac{d}{dt}h(\theta) = \frac{dh(\theta)}{d\theta^T}\frac{d\theta}{dt} = \nabla h(\theta)^T \dot{\theta}$$

$$\min \ \frac{1}{2}\dot{\theta}^T M(\theta)\dot{\theta} + \nabla h(\theta)^T \dot{\theta} \qquad \textbf{subject to} \ \ \mathcal{V}_d = J\dot{\theta}$$

$$\begin{aligned}
H &= \frac{1}{2}\dot{\theta}^T M(\theta)\dot{\theta} + \nabla h(\theta)^T \dot{\theta} + \lambda^T(\mathcal{V}_d - J\dot{\theta}) \\
\frac{\partial H}{\partial \dot{\theta}} &= M(\theta)\dot{\theta} + \nabla h - J^T\lambda = 0 \\
\mathcal{V}_d &= J\dot{\theta} = JM^{-1}(J^T\lambda - \nabla h) = JM^{-1}J^T\lambda - JM^{-1}\nabla h \\
\lambda &= (JM^{-1}J^T)^{-1}(\mathcal{V}_d + JM^{-1}\nabla h) \\
\dot{\theta} &= M^{-1}(J^T\lambda - \nabla h) = M^{-1}J^T(JM^{-1}J^T)^{-1}\mathcal{V}_d + M^{-1}J^T(JM^{-1}J^T)^{-1}JM^{-1}\nabla h - M^{-1}\nabla h \\
&= J_M^+\mathcal{V}_d + (I - J_M^+ J)M^{-1}(-\nabla h)
\end{aligned}$$

Interpretation of J_M^+

With $J_M^+ = M^{-1}J^T(JM^{-1}J^T)^{-1}$, the kinematic resolution of

$$\lambda = (JM^{-1}J^T)^{-1}(\mathcal{V}_d + JM^{-1}\nabla h)$$
$$\dot{\theta} = J_M^+\mathcal{V}_d + (I - J_M^+J)M^{-1}(-\nabla h)$$

- The Lagrange multiplier λ (see Appendix D) can be interpreted as a wrench in task space, from $\tau = J^T\mathcal{F}$
- Moreover, in the expression $\lambda = (JM^{-1}J^T)^{-1}(\mathcal{V}_d + JM^{-1}\nabla h)$,
  - the first term, $(JM^{-1}J^T)^{-1}\mathcal{V}_d$, can be interpreted as a dynamic force generating the end-effector velocity $\mathcal{V}_d$
  - the second term, $(JM^{-1}J^T)^{-1}JM^{-1}\nabla h$, can be interpreted as the static wrench counteracting gravity.

4 Homework : Chapter 6

- Please solve and submit Exercise 6.3, 6.4, 6.5, 6.6, 6.8, 6.10, 6.11, 7.15 , till May 10th (upload it as a pdf form or email me)