Partial Derivative

\[ f(x, y) = 3x^2 + 4xy - \cos(y) \]
\[ \frac{\partial f}{\partial x} = 6x + 4y \quad \frac{\partial f}{\partial y} = 4x + \sin(y) \]

Differential Kinematics

How does a small change in configuration space affect the end effector pose in the workspace?

E.g.: with 2 link arm:
\[ \alpha = (\theta_1, \theta_2) \quad \Rightarrow \quad \mathbf{p} = \begin{bmatrix} x(\theta_1, \theta_2) \\ y(\theta_1, \theta_2) \end{bmatrix} \]

\[ \mathbf{J}(q) = \begin{bmatrix} \frac{\partial x}{\partial \theta_1} & \frac{\partial x}{\partial \theta_2} \\ \frac{\partial y}{\partial \theta_1} & \frac{\partial y}{\partial \theta_2} \end{bmatrix} \]

\text{Manipulator Jacobian}
\( \begin{align*} x(\theta_1, \theta_2) &= l_1 \cos \theta_1 + l_2 \cos (\theta_1 + \theta_2) \\
y(\theta_1, \theta_2) &= l_1 \sin \theta_1 + l_2 \sin (\theta_1 + \theta_2) \end{align*} \)

\[ J(q) = \begin{bmatrix} -l_1 \sin \theta_1 \quad -l_2 \sin (\theta_1 + \theta_2) \\
l_1 \cos \theta_1 \quad l_2 \cos (\theta_1 + \theta_2) \end{bmatrix} \]

**Analytic Method:**

write down FK and manually take partial derivatives

\[ \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = J(q) \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} \]

Jacobian Vel in Workspace Vel in C-space
Geometric Method
- Compute workspace positions of all joints + E.E.
- Column of J corresponds to a revolute joint is a vector perpendicular to
- Difference between E.E. pos & Joint pos
  \((\Delta x, \Delta y) \rightarrow (-\Delta y, \Delta x)\)
- Column vector for prismatic is just unit vector in axial direction

Relate Torques at Joints to Forces at E.E.
__Principle of Virtual Work__
- Change in coords does not change quality of work
  \[ \text{work} = \text{force} \times \text{distance} = \text{torque} \times \text{angular displacement} \]
- Force \times distance = torque \times angle
- Force \times Linear vel = Torque \times Angular vel
  \[ \text{T}_v \hat{v} = \text{Torque} \]
\[ \dot{q}^T F = \dot{q}^T T \]

But \( \dot{q} = J(q) \dot{q} \) (def of Jacobian)

\( (J(q) \dot{q})^T F = \dot{q}^T T \)

So \( J(q)^T F = T \)

Jacobian transpose is linear map from workspace to joint space torques

Jacobian-based IK

\[ \dot{q} = J(q) \dot{q} \] (True by definition)

\[ \Delta \dot{q} \approx J(q) \Delta q \]

Small workspace displacement

Small angular displacement

(Poles - Pactual)

So if we know desired displacement in workspace we can compute approximate desired displacement in configuration space.

\[ \Delta q = J(q)^{-1} \Delta \dot{q} \]

\[ = J(q)^{-1} (\text{Poles} - \text{Pactual}) \]
Does it invert?
- What is the size of $J$?
  - $n$ config space DOF
  - $m$ workspace DOF
- $J$ is $m \times n$
- $J^{-1}$ does not exist if $m \neq n$
- We are not interested in $n < m$

- OK if $n \geq m$

**Pseudo inverse**

- Compute least squares solution to $Ax = b$
  (Solve for $x$)

\[
Ax = b
\]

\[
ATAx = A^Tb
\]

\[
x = (ATA)^{-1}A^Tb
\]

Pseudo inverse of $A$
therefore $\Delta q = (J^T J)^{-1} J^T \Delta p$

is the pseudo inverse way of doing this

Gauss Newton Algorithm for Jacobian based FK

while (1):
  $\Delta p = \text{dques - pactual}$
  if $\|\Delta p\| \leq 3$:
    return 0
  else
    $\Delta q = \lambda (J^T J)^{-1} J^T \Delta p$
    small stepsize $\approx 0.05$
    $q = q + \Delta q$
    update pactual
  end
end

If $J$ might be singular, use damped least squares

$\Delta q = \lambda (J^T J + \lambda I)^{-1} J^T \Delta p$

regular identity matrix
Big $\lambda$ = more cautious
Feedback Control
- Let $\bar{q}$ be the observed configuration/state of the robot.
- Let $u$ be a control
  - eg. with turtlebots
    $\bar{q} = (x, y, \theta)$
    $u = (x_R, \dot{\theta})$

A feedback control scheme continuously/repeatedly adjusts $u$ in response to $\bar{q}$
aka closed-loop control

![Diagram of feedback control system]

eg. pagion seeking

$\bar{q} = (x, y, \theta)$
$x_R = c$, $\dot{\theta} = k_p c_x$
More involved example: line following
- Robot is at $\overrightarrow{a} = (x, y, \theta)$
- Want to drive along a line
  with equation
  $$\mathbf{p}(s) = \mathbf{p}_0 + s \cdot \mathbf{d}$$

Why continuity?
- Discontinuous = unstable
- Excess acceleration, jerk is bad

Pure pursuit
1. Find the point $\mathbf{p}_c$ on the line closest to the robot
2. March ahead on the line from $\mathbf{p}_c$ some distance $x$ to get to $\mathbf{p}_d$
3. Pretend $\mathbf{p}_d$ is a target and drive to it.

$$\dot{\theta} = K_P \frac{c_y}{c_x}$$
cx, cy is robot-frame coordinates of cone
or $\dot{\theta} = K_P \arctan \left( \frac{c_y}{c_x} \right)$
\[(P_c - t) \perp d\]

Let \( v = t - P_o \)

\[P_c = P_o + d \cdot v = d^T v = d d^T (t - P_o)\]

We want proj of \( v \) onto \( d = v \cdot d \cdot d^T v = v^T d\]

\[
\begin{bmatrix}
R(\theta) & y \\
0 & 1
\end{bmatrix}
\]

Pos of robot in line frame is \((x, 0)\)

Pd in line frame is \((x + a, 0)\)

Let p_{d} in robot frame be \((cx, cy)\)

\[
\begin{bmatrix}
Cx \\
cy \\
1
\end{bmatrix}
= (T^c)^{-1}\begin{bmatrix}x + a \\ 0 \end{bmatrix}
\]

Pd in line frame

\[
\text{Pd in robot frame}
\]

\[
\text{Line to robot}
\]
$P_0$ is line frame origin in world frame

$d = (d_x, d_y)$ is line direction in world frame

$$T_w = \begin{bmatrix} d_x & -d_y & P_0 \times \\
 d_y & d_x & P_0 y \\
 0 & 0 & 1 \end{bmatrix}$$

Problems

$\dot{\theta} = K_p \arctan \left( \frac{cy}{cx} \right)$

Watch out for divide by zero.