Chapter 4: Forward Kinematics — 7.10.22

The end-effector position and orientation is uniquely determined from the joint positions (for an open chain).

Forward Kinematics: find position and orientation of end-effector frame given the joint positions. Product of exponentials (PoE) formula describes forward kinematics of open chains.

Product of Exponentials Formula

Assign a stationary frame $\{s\}$ to the base of the robot and a frame $\{b\}$ to the end-effector.
M describes $\{b\}$ when the robot is at zero position.

4.1.1 Screw Axes in the Base Frame — 7.10.22

Product of Exponential (PoE): Regard each joint as applying a screw motion to all the links.

- choose fixed frame $\{s\}$
- choose end-effector frame $\{b\}$
- place Robot in "zero-position": all joint values are zero
- Let $M \in SE(3)$ end-effector relative to $\{s\}$ in "zero-position"

Then joint n with joint value θ_n:
M is displaced:

$$T = e^{[S_n]\theta_n} M$$

where $T \in SE(3)$: new configuration of end-effector
and $S_n = (\omega_n, v_n)$: screw-axis of joint n in $\{s\}$

For S and θ if joint is Revolute (zero pitch):
- $\omega_n \in \mathbb{R}^3$ is unit vector $\|\omega_n\| = 1$ (direction positive of joint axis)
- $v_n = -\omega_n \times q_n$, with q_n: arbitrary point on joint axis (in $\{s\}$)
- θ_n: joint angle

For S and θ if joint is prismatic:

- $\omega_n = 0$
- $v_n \in \mathbb{R}^3$ is a unit vector $|v_n| = 1$ (direction: positive translation)
- θ_n: prismatic extension / retraction

Applying PoE to all joints:

PoE Space Form

$$\boxed{T(\theta) = e^{[S_1]\theta_1} \dots e^{[S_{n-1}]\theta_{n-1}} \cdot e^{[S_n]\theta_n} M}$$

4.1.3 Screw Axes in the End-Effector Frame

The screw axes of the joints can be represented in the End-Effector frame $\{b\}$ as:

$$[B_i] = [Ad_{M^{-1}}] \cdot S_i \qquad \text{for } i = 0, \dots, n$$

PoE with joint axes repesented in End-Effector Frame: (body form)

$$T(\theta) = M e^{[B_1]\theta_1} \dots e^{[B_{n-1}]\theta_{n-1}} \cdot e^{[B_n]\theta_n}$$

28.10.22

Chapter 5: Velocity Kinematics and Statics

Calculating end-effector twist from joint positions and velocities.

Represent end-effector configuration by minimal set of coordinates $x \in \mathbb{R}^m$ (twist $V \in \mathbb{R}^6$ is covered later)

Velocity : $\dot{x} = \frac{dx}{dt} \in \mathbb{R}^m$

⇒ forward Kinematics $x(t) = f(\theta(t))$ where $\theta \in \mathbb{R}^n$: set of joint variables

time derivative at time t:
(chain rule)

$$\dot{x} = \frac{\partial f(\theta)}{\partial \theta} \cdot \frac{d\theta(t)}{dt}$$

$$= \frac{\partial f(\theta)}{\partial \theta} \cdot \dot{\theta} = \underbrace{J(\theta)}_{\text{Jacobian}} \dot{\theta}$$

The Jacobian $J(\theta) = \frac{\partial f(\theta)}{\partial \theta} \in \mathbb{R}^{m \times n}$ represents the linear sensitivity of the end-effector velocity $\dot{x}$ to the joint velocity $\dot{\theta}$ and is a function of the variables θ.

Manipulability Ellipsoid:

Mapping a unit circle of joint velocities ("iso-effort") where the total actuator effort is sum of squares of joint velocities.
This circle maps trough the Jacobian to an ellipsoid in the space of tip velocities (v_{tip}). This ellipsoid is called manipulability ellipsoid.
Using the manipulability ellipsoid one can quantify how close a given posture is to a singularity.

9.11.22

Static Analysis and Force:

The Jacobian also plays a central role in static analysis.
↳ What are the joint torques required to resist an external force?

Power measured at the robot tip must be equal to the power generated at the joints (conservation of power)

$$f_{tip}^T v_{tip} = \tau^T \dot{\theta}$$

f_{tip} : tip force vector
τ : joint torque vector

with $v_{tip} = J(\theta)\dot{\theta}$

↳ $f_{tip}^T J(\theta) \cdot \dot{\theta} = \tau^T \dot{\theta}$

$\Rightarrow \quad \tau = J^T(\theta) f_{tip}$

if configuration θ is not a singularity: $J(\theta)$ and $J^T(\theta)$ are invertible

$$f_{tip} = ((J(\theta))^T)^{-1} \tau = J^{-T}(\theta)\tau$$

Force Ellipsoid:

Mapping a unit circle "~~...~~ iso-effort" from τ to f via the Jacobian transpose inverse $J^{-T}(\theta)$.
The force ellipsoid illustrates how easily the robot can generate forces in different directions

Jacobian

n = Number of Joints

$$J(\theta) = \frac{\partial f(\theta)}{\partial \theta} \in \mathbb{R}^{m \times n}$$

- Column i of the Jacobian corresponds to the end-effector velocity when joint i moves at unity speed and all other joints are stationary.
- If the Jacobian is not full rank (one row becomes a scalar multiple of another row) for a given configuration the robot is at a singularity.

Jacobian Matrix

$f : \mathbb{R}^n \to \mathbb{R}^m$

$\hookrightarrow f(x) \in \mathbb{R}^m \quad , x \in \mathbb{R}^n$

$\hookrightarrow$ first order partial derivatives of f need to exist

$$J = \begin{bmatrix} \frac{\partial f}{\partial x_1} & \cdots & \frac{\partial f}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \nabla^T f_1 \\ \vdots \\ \nabla^T f_m \end{bmatrix}$$

$$= \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

J: Jacobian

$\nabla^T f_i$: transpose (row vector) of gradient of the i component

5.1 Manipulator Jacobian   18.11.22

Tip velocity vector v_{tip} is linearly related to joint velocity vector $\dot{\theta}$:

$$v_{tip} = J(\theta) \cdot \dot{\theta}$$

Given configuration θ of robot:

$$J_i(\theta) \in \mathbb{R}^6 \quad :(\text{column } i \text{ of } J(\theta))$$

is the twist V when $\dot{\theta}_i = 1$ and all other joint velocities are 0.

The twist V is determined the same way as the joint screw axes are determined: using a point q_i on joint axis i for revolute joints. The difference ~~the~~ is that the screw axes of the ~~[illegible]~~ Jacobian depend on the joints (θ) where the screw axes for forward kinematics are always for $\theta = 0$.

Two ~~[illegible]~~ standard types of the Jacobian:

- space Jacobian: $V_s = J_s(\theta)\dot{\theta}$ — each column $J_{si}(\theta)$ corresponds to a screw axis in $\{s\}$
- body Jacobian: $V_b = J_b(\theta)\dot{\theta}$ — each column $J_{bi}(\theta)$ corresponds to a screw axis in $\{b\}$

5.1.1. Space Jacobian

End-effector velocity : twist V_s

$$V_s = J_s(\theta)\dot{\theta}$$
$$= [J_{s1}(\theta)\ J_{s2}(\theta) \dots J_{sn}(\theta)]\dot{\theta}$$

V_s: twist (end-effector velocity)
J_s: space Jacobian
$J_s(\theta) \in \mathbb{R}^{6\times n}$
n: number of joints

Column $J_{si}(\theta)$ is the spacial twist when joint i velocity is 1 and velocity of all other joints is 0.

Space Jacobian $J_s(\theta)$ is defined by

$$\boxed{\begin{array}{ll} & V_s = J_s(\theta)\dot{\theta} \\ \text{where} & J_s(\theta) = [J_{s1}(\theta)\ J_{s2}(\theta) \dots J_{sn}(\theta)] \in \mathbb{R}^{6\times n} \\ \text{with} & J_{s1} = S_1 \quad \text{(Screw axis } S_1 \text{ when robot is at zero config.)} \\ \text{and} & J_{si}(\theta) = [Ad_{e^{[S_1]\theta_1} \dots e^{[S_{i-1}]\theta_{i-1}}}] S_i \text{ for } i = 2, \dots, n \end{array}}$$

- The first column $\overset{= J_{s1}}{S_1}$ of space Jacobian J_s:
  - ↳ just screw axis S_1 wen robot at zero conf (does not depend on joint positions)
- All other coluns J_{si} (for $i>1$):
  - ↳ Screw axis S_i premuliplied by transformation to express screw axis in $\{s\}$

<u>No differentiation</u> is necessary to calculate the Jacobian.

Space Jacobian is independent of choice of $\{b\}$ frame (end-effector frame)

5.1.2 Body Jacobian

26.11.22

Body Jacobian $J_b(\theta)$ is defined by

$$V_b = J_b(\theta)\dot{\theta}$$

where $J_b(\theta) = [J_{b1}(\theta) \dots J_{bn-1}(\theta)\, J_{bn}] \in \mathbb{R}^{6\times n}$

with $J_{bn} = B_n$

and $J_{bi}(\theta) = [Ad_{e^{-[B_n]\theta_n} \dots e^{-[B_{i+1}]\theta_{i+1}}}] B_i$ for $i = 1, \dots, n-1$

- The last column $\underset{= B_n}{J_{bn}}$ of the body Jacobian J_b:
  ↳ just screw axis B_n when robot at zero conf (does not depend on joint positions)

- All other colums J_{bi} (for $i < n$):
  ↳ Screw axis B_i premultiplied by transformation to express screw axis in $\{b\}$

Conversion between Space and Body Jacobian

$$\boxed{\begin{aligned} J_b(\theta) &= [Ad_{T_{bs}}]\, J_s(\theta) \\ J_s(\theta) &= [Ad_{T_{sb}}]\, J_b(\theta) \end{aligned}}$$

where $T_{sb} = T(\theta)$

J_b and J_s have always the same <u>rank</u>.

Geometric vs. Analytic Jacobian

The geometric (basic) Jacobian described in the book is calculated using twists.
There is also an analytic Jacobian that is calculated by differatiation of the forward kinematics w.r.t. the joint variables.

5.2. Statics of open Chains

3.12.22

Conservation of power:

power at joints = power to move robot
+ power at end-effector

If robot is not moving (static equilibrium):

↳ power at joints = power at end-effector
(no power used to move the robot)

$$\tau^T \dot{\theta} = \mathcal{F}_b^T \mathcal{V}_b$$

τ : column vector of joint torques

using $\mathcal{V}_b = J_b(\theta)\dot{\theta}$

$$\boxed{\begin{aligned} \tau &= J_b^T(\theta)\,\mathcal{F}_b \\ \text{and} \quad \tau &= J_s^T(\theta)\,\mathcal{F}_s \end{aligned}}$$

5.3 Singularity Analysis 29.12.22

The Jacobian helps to identify postures at which the robot's end-effector loses the ability to move in one or more directions: this is called a <u>singularity</u> (kinematic singularity)

Mathematically:

singular posture: Jacobian $J(\theta)$ is <u>not</u> of maximal rank.

Singularities are the same for the space- and the body-Jacobian.

$$\operatorname{rank} J_s(\theta) = \operatorname{rank} J_b(\theta)$$

Singularities are also independent of the reference frame.

5.4. Manipulability

10.3.23

Conditions:
- $m \leq n$
- J is invertible

$A = JJ^T \in \mathbb{R}^{m \times m}$

v_i : eigenvectors of A

λ_i : eigenvalues of A

J is either J_b or J_s

V : Volume of manipulability ellipsoid

V is proportional to $\sqrt{\lambda_1 \lambda_2 \ldots \lambda_m} = \sqrt{\det(A)} = \sqrt{\det(JJ^T)}$

If J is full rank (i.e rank m) A and A^{-1} are
- square
- symmetric
- positive definite

J can be expressed as $J(\theta) = \begin{bmatrix} J_\omega(\theta) \\ J_v(\theta) \end{bmatrix}$ — J_ω top 3 rows, J_v bottom 3 rows of J

↳ 3 dimensional manipulability ellipsoids
- J_ω for angular velocities → $A = J_\omega J_\omega^T$
- J_v for linear velocities → $A = J_v J_v^T$

→ principal semi-axis : aligned w/ its eigenvectors of A : v_i
- length : square root of eigenvalues : $\sqrt{\lambda_i}$

Measures for Manipulability

- ratio of longest to shortest semi-axis

$$\mu_1(A) = \frac{\sqrt{\lambda_{max}(A)}}{\sqrt{\lambda_{min}(A)}} = \sqrt{\frac{\lambda_{max}(A)}{\lambda_{min}(A)}} \geq 1$$

- When $\mu_1(A)$ is low (≈ 1) manipulability is high
- when $\mu_1(A)$ is high ($\to \infty$) manipulability is low

- Condition number

$$\mu_2(A) = \frac{\lambda_{max}(A)}{\lambda_{min}(A)} \geq 1$$

- small → manipulability is high
- big → manipulability is low

- proportional to volume of manipulability ellipsoid (relative)

$$\mu_3(A) = \sqrt{\lambda_1 \lambda_2 \ldots} = \sqrt{\det(A)}$$

- larger is better

Modern Robotics | 12.2.23

Velocity kinematics

$J(\theta)$: Jacobian ← linear — function of joint variables (θ)
sensitivity of endeffector velocity v_{tip} to joint velocity $\dot{\theta}$

usually twist $\in \mathbb{R}^6$

Given θ : configuration of Robot

$$J(\theta) = \left(J_1(\theta) \;\; J_2(\theta) \; \dots \; J_i(\theta) \dots J_n(\theta) \right)$$

J_i : twist V when $\dot{\theta}_i = 1$
all other $\dot{\theta}_j = 0$ for $j \neq i$

Twist J_i is determined as screw axes for forward kinematics using a point q_i on axis i for revolute joints.
Difference: screw axes of J_i depend on join variables θ. (For forward kinematic: $\theta = 0$)

Defining Joints for Forward Kinematics or Velocity Kinematics:

for joint n

Revolute joint (zero pitch):

- $\omega_n \in \mathbb{R}^3$ unit vector in positive direction of joint axis n
- $v_n = -\omega_n \times q_n$ with q_n any point on joint axis n
- θ_n: joint angle

Prismatic joint:

- $\omega_n = 0$
- $v_n \in \mathbb{R}^3$: unit vector in direction of positive translation
- θ_n: prismatic extension/retraction.

Screw axis S:

$$S_n = (\omega_n, v_n)$$

6. Inverse Kinematics

Given a homogeneous transform $X \in SE(3)$ find θ that satisfy $T(\theta) = X$.

Helpful formulas/laws:

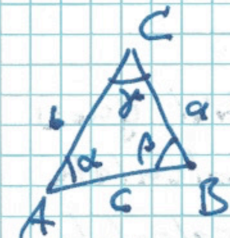
implementations often use this argument order

- $\text{arctan2}(y, x) : \mathbb{E} \to (-\pi, \pi]$

$$(x, y) = \begin{cases} \arctan\left(\frac{y}{x}\right) & \text{for } x > 0 \quad \text{Quadrant I and IV} \\ \arctan\left(\frac{y}{x}\right) + \pi & \text{for } x < 0, y > 0 \quad \text{Quadrant II} \\ \pm\pi & \text{for } x < 0, y = 0 \\ \arctan\left(\frac{y}{x}\right) - \pi & \text{for } x < 0, y < 0 \quad \text{Quadrant III} \\ +\frac{\pi}{2} & \text{for } x = 0, y > 0 \\ -\frac{\pi}{2} & \text{for } x = 0, y < 0 \end{cases}$$

- law of cosines (Kosinussatz)

$$c^2 = a^2 + b^2 - 2ab\cos(C) \quad \text{“}\gamma\text{”}$$



See the Book for some Analytic Inverse Kinematic examples.

6.2 Numerical Inverse Kinematics

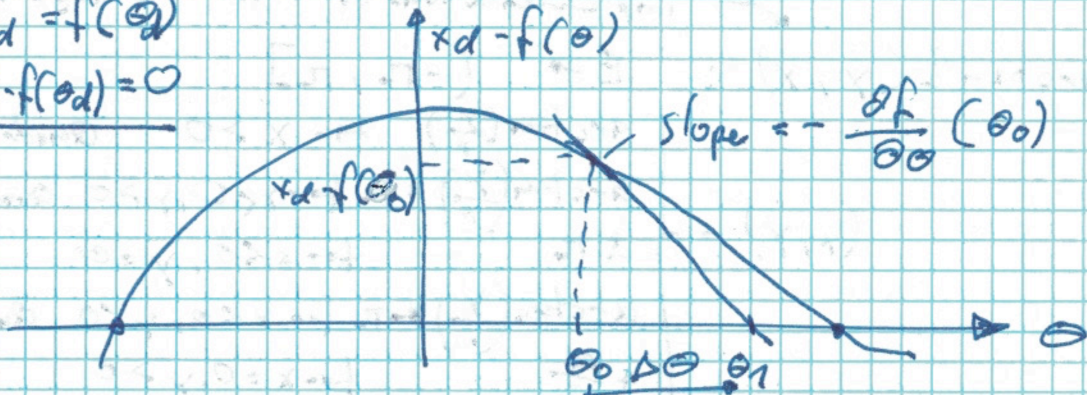
Forward Kinematics : $x = f(\theta)$ — f maps n joint coordinates to m end-effector coordinates
(coordinate-based)

$f : \mathbb{R}^n \to \mathbb{R}^m$

x_d : desired end-effector coordinates

↳ $x_d = f(\theta_d)$

↳ $x_d - f(\theta_d) = 0$



- Newton-Raphson: the roots correspond to joint values
(for scalar) θ that solve the inverse kinematics

1. initial guess θ_0
   ↳ calculate $x_d - f(\theta_0)$
   ↳ calculate slope $= \frac{\partial f}{\partial \theta}(\theta_0)$
2. where slope crosses θ axis: new guess θ_1

$$\Delta\theta = \left(\frac{\partial f}{\partial \theta}(\theta_0)\right)^{-1}(x_d - f(\theta_0))$$

   θ_1 is close to a solution

3. repeat $\theta_2 \dots$

- Newton-Raphson for vector of joints and endpoint coordinates

Taylor expansion: $x_d = f(\theta_d) = f(\theta_i) + \underbrace{\frac{\partial f}{\partial \theta}\Big|_{\theta_i}}_{J(\theta_i)} \underbrace{(\theta_d - \theta_i)}_{\Delta\theta} + \dots$

higher order terms are ignored

↳ $x_d - f(\theta_i) = J(\theta_i)\Delta\theta$

$\Rightarrow J^{-1}(\theta_i)(x_d - f(\theta_i)) = \Delta\theta$ — θ_i: current guess

if J is not invertible → use pseudo inverse

Pseudo inverse

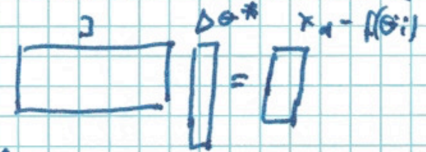
17.3.23

$$x_d - f(\theta_i) = J(\theta_i)\,\Delta\theta$$

$$\hookrightarrow \quad J^{+}(\theta_i)(x_d - f(\theta_i)) = \Delta\theta^{*} \qquad J^{+}: \text{pseudo inverse}$$

2 cases for Pseudo inverse:

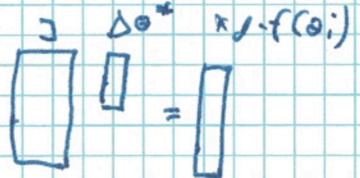
- Underdetermined: $n < m$ (short-fat J)



- ∞ many solutions (in general)

$\Delta\theta^{*}$ has smallest 2-norm among all solutions

- Overdetermined: $n > m$ (tall-skinny J)



- 0 solutions (in general)

$\Delta\theta^{*}$ comes closest to a solution in the 2-norm sense

Pseudo-Inverse can be calculated by Singular Value Decomposition
$\hookrightarrow$ Moore-Penrose

Numerical Inverse Kinematics with Twists

(a) Initialization: Given T_{sd} (desired end-effector config.) and initial guess $\theta^{0} \in \mathbb{R}^{n}$, set $i = 0$

(b) Set $[V_b] = \log(T_{sb}^{-1}(\theta^{i})\,T_{sd})$.

check for linear and angular components of body twist

While $\|\omega_b\| > \varepsilon_\omega$ or $\|v_b\| > \varepsilon_v$ for small $\varepsilon_\omega, \varepsilon_v$:

- set $\theta^{i+1} = \theta^{i} + J_b^{+}(\theta^{i})\,V_b$
- increment i

- There are different possibilities to define the error of the current solution and use it as an condition for the loop

Initial guess:

- Robot controller: previous joint vector
- (approx.) analytical inverse kinematics
- ...

Inverse Velocity Kinematics

Sometimes we only need the joint velocities to achiev a desired end-effector twist.

$$\dot{\theta} = J^{+}(\theta) \mathcal{V}_d$$