Autonomous Mobile Robots

CHEAT SHEET BY JANIS HUTZ
ETHZ, FS2026

1 Introduction

1.1 Probability

D 1.1 *(Sum rule)* $P(X) = \sum P(X, Y) = \sum P(X \cap Y)$

D 1.2 *(Prod)* $P(X, Y) = P(X|Y)P(Y) = P(Y|X)P(X)$

T 1.3 *(Bayes)* $P(Y_i|X) = \frac{P(X|Y_i)P(Y_i)}{\sum_{j=1}^{n} P(X|Y_j)P(Y_j)}$

D 1.4 *(Cont. Var)* Sums become integrals
e.g. $\sum_X P(X) = 1$ becomes $\int p(x)\ \mathrm{d} = 1$

D 1.5 *(Indep.)* x, y indep. iff $p(x, y) = p(x)p(y)$

D 1.6 *(Cond. Indep.)* iff $p(x, y|z) = p(x|z)p(y|z)$

D 1.7 $E[\boldsymbol{x}] = \int_{-\infty}^{\infty} \boldsymbol{x}p(\boldsymbol{x})\ \mathrm{d}\boldsymbol{x}$, also for $\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{x})$

D 1.8 $\mathrm{Cov}[x] = E[\boldsymbol{x}\boldsymbol{x}^\top] - E[\boldsymbol{x}]E[\boldsymbol{x}]^\top = \boldsymbol{\Sigma}$

D 1.9 *(Gauss. Dist.)* $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ ($\boldsymbol{\mu}$ mean, $\boldsymbol{\Sigma}$ cov.),
PDF: $p(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^k|\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)$

1.2 Measurement models

$\boldsymbol{z} = \boldsymbol{b}_C + s\boldsymbol{M}_S\boldsymbol{\omega} + \boldsymbol{b} + \boldsymbol{n} + \boldsymbol{o}$: $\boldsymbol{b}_C$ const bias, $\boldsymbol{b}$ time bias, $\boldsymbol{M}$ missal., $\boldsymbol{n} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R})$ noise, $_S\omega$ corr. meas., $\boldsymbol{o}$ other infl.

1.3 Trigonometry

2 Locomotion & Kinematics

2.1 Positioning

D 2.1 *(Position Vector)* $_W\boldsymbol{t}_B = {}_W\boldsymbol{t}_{WB}$, Original Frame, End point, Target Frame, $\sin = s, \cos = c$

D 2.2 *(State vector)* x_R: x, v of rob in W, pos of sensors

D 2.3 *(Rot. Mat.)* $\boldsymbol{R}_z = \begin{bmatrix} c(\psi) & -s(\psi) & 0 \\ s(\psi) & c(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$

$\boldsymbol{R}_y(\theta) = \begin{bmatrix} c(\psi) & 0 & s(\psi) \\ 0 & 1 & 0 \\ -s(\psi) & 0 & c(\psi) \end{bmatrix}$ $\boldsymbol{R}_x(\varphi) \begin{bmatrix} 1 & 0 & 0 \\ 0 & c(\psi) & -s(\psi) \\ 0 & s(\psi) & c(\psi) \end{bmatrix}$

R 2.4 Application: $_W\boldsymbol{a} = \boldsymbol{R}_{WB}\,{}_B\boldsymbol{a}$

L 2.5 $\boldsymbol{R}_{BW} = \boldsymbol{R}_{WB}^{-1} = \boldsymbol{R}_{WB}^\top$, $\det(\boldsymbol{R}_{WB}) = 1$ (orth.)

R 2.6 Cols of $\boldsymbol{R}_{WB}$ are basis vec. of Frame $\underset{\rightarrow}{\mathcal{F}}_B$ in $\underset{\rightarrow}{\mathcal{F}}_W$

D 2.7 *(Euler Angles)* Yaw (z), Pitch (y), Roll (x), mult. rotation matrices, e.g. $\boldsymbol{R}_{EB} = \boldsymbol{R}_z(\psi) \cdot \boldsymbol{R}_y(\theta) \cdot \boldsymbol{R}_x(\varphi)$, bound..
$[\boldsymbol{n}]^\times = \boldsymbol{n}\boldsymbol{x}^\top$ (matrix from vec + arg $\boldsymbol{x}$)

D 2.8 *(Rot. Vec)* $\boldsymbol{\alpha} = \alpha\boldsymbol{n}$ ($\boldsymbol{n}$ normal)
$\boldsymbol{R}(\alpha, \boldsymbol{n}) = \boldsymbol{I}_3 + \sin(\alpha)[\boldsymbol{n}]^\times + (1 - \cos(\alpha))([\boldsymbol{n}]^\times)^2$

D 2.9 *(Quaternions)* $q = q_w + q_x i + q_y j + q_z k$ with $i^2 = j^2 = k^2 = -1$, ($ij = -ji = k$, same for jk and ki)

D 2.10 *(Transf. M)* $\boldsymbol{T}_{AB} = \begin{bmatrix} \boldsymbol{R}_{AB} & {}_A\boldsymbol{t}_B \\ \boldsymbol{0}_{1\times 3} & 1 \end{bmatrix}$

$\boldsymbol{T}_{BA} = \boldsymbol{T}_{AB}^{-1} = \begin{bmatrix} \boldsymbol{R}_{AB}^\top & -\boldsymbol{R}_{AB}^\top\,{}_A\boldsymbol{t}_B \\ \boldsymbol{0}_{1\times 3} & 1 \end{bmatrix}$ $\boldsymbol{T}_{AC} = \boldsymbol{T}_{AB}\boldsymbol{T}_{BC}$

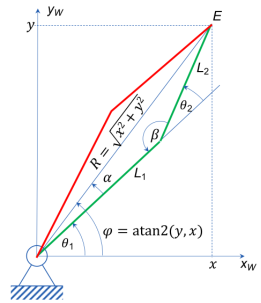
2.2 Forward Kinematics (FK)

$T_{WB_n}(\theta) = \boldsymbol{T}_{WB_0}\boldsymbol{T}_{B_0B_1}(\theta_1) \cdots \boldsymbol{T}_{B_{n-1}B_n}(\theta_n)$.

For 2R system: $_W\boldsymbol{t}_{WE} = \begin{bmatrix} L_1\cos(\theta_1) + L_2\cos(\theta_1 + \theta_2) \\ L_1\sin(\theta_1) + L_2\sin(\theta_1 + \theta_2) \end{bmatrix}$

With workspace (pos) W for $\theta_1, \theta_2 \in [-\pi, \pi]$

2.3 Inverse Kinematics (IK)

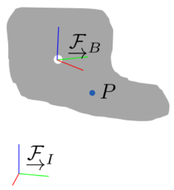


Option: Solve Forward Kinematics for angles.
Better: Law of cosine with polar coordinates. Compute angle using cosine rule,
$\theta_1 = \phi \pm \alpha$, $\theta_2 = \pm(\pi - \beta)$
(Positive for Elbow Down, Negative for Elbow Up)
Extension to 6R: 1. Waist: spherical coords (2 sol.)
2. 2 sols from 2R for shoulder + elbow
3. Solve for wrist joints (no influence on pos)

2.4 Temporal Models

For ***Cont-time n.-lin. system of ODE*** $\dot{\boldsymbol{x}} = \boldsymbol{f}_C(\boldsymbol{x}(t), \boldsymbol{u}(t))$, with measurements $\boldsymbol{z}(t) = \boldsymbol{h}(\boldsymbol{x}(t)) + \boldsymbol{v}(t)$.
Need linearised (around $\boldsymbol{f}_C(\overline{\boldsymbol{x}}, \overline{\boldsymbol{y}}) = 0$, at ***equilibrium***):
$\delta\dot{\boldsymbol{x}}(t) = \boldsymbol{f}_C(\overline{\boldsymbol{x}}, \overline{\boldsymbol{u}}) + \boldsymbol{F}_C\delta\boldsymbol{x}(t) + \boldsymbol{G}_C\delta\boldsymbol{u}(t) + \boldsymbol{L}_C\boldsymbol{w}(t)$
$\delta\boldsymbol{z}(t) = \boldsymbol{H}\delta\boldsymbol{x}(t) + \boldsymbol{v}(t)$. Herein, $\boldsymbol{H}$ is measurements, $\boldsymbol{F}_C$ system, $\boldsymbol{G}$ input gain, $\boldsymbol{w}$ process noise, $\boldsymbol{v}$ measurement noise, both zero-mean ***Gaussian White Noise Process***.
For ***n-lin. cont-time system***: $\dot{\boldsymbol{x}}(t) = \boldsymbol{f}_C(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{w}(t))$
$\boldsymbol{z}(t) = \boldsymbol{h}(\boldsymbol{x}(t)) = \boldsymbol{v})(t)$, linearization is the same
To ***discretize***, integrate from t_{k-1} to t_k:
$\boldsymbol{x}_k = \boldsymbol{f}(\boldsymbol{x}_{k-1}, \boldsymbol{u}_k, \boldsymbol{w}_k)$ $\boldsymbol{z}_k = \boldsymbol{h}(\boldsymbol{x}_k) + \boldsymbol{v}_k$, ***linearised***:
$\delta\boldsymbol{x}_k = \boldsymbol{f}(\overline{\boldsymbol{x}}, \overline{\boldsymbol{u}}) + \boldsymbol{F}\delta\boldsymbol{x}_{k-1} + \boldsymbol{G}_k\delta\boldsymbol{u}_k + \boldsymbol{L}_k\boldsymbol{w}_k$; $\delta\boldsymbol{z}_k = \boldsymbol{H}_k\delta\boldsymbol{x}_k$
Trapezoidal num. int $\Delta\boldsymbol{x}_1 = \Delta t\boldsymbol{f}_C(\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1}, t_{k-1})$
$\Delta\boldsymbol{x}_2 = \Delta t\boldsymbol{f}_C(\boldsymbol{x}_{k-1} + \Delta\boldsymbol{x}_1, \boldsymbol{u}_k, t_k)$, then:
$\boldsymbol{x}_k = \boldsymbol{x}_{k-1} + 0.5 \cdot (\Delta\boldsymbol{x}_1 + \Delta\boldsymbol{x}_2)$

2.5 Rigid body & IMU kinematics



Velocity $_I\boldsymbol{v}_{IB} = \frac{\mathrm{d}}{\mathrm{d}t}({}_I\boldsymbol{t}_B)$
Rot. Velocity $_I\boldsymbol{\omega}_{IB} = \frac{\mathrm{d}}{\mathrm{d}t}(\alpha)\ {}_I\boldsymbol{t}$
Velocity point P $_B\boldsymbol{v}_{IP} = {}_B\boldsymbol{v}_{IB} + {}_B\boldsymbol{\omega}_{IB} \times {}_B\boldsymbol{t}_P$
Rotation Matrices

- For left pertubing $\dot{\boldsymbol{R}}_{IB} = [{}_I\omega_{IB}]^\times \boldsymbol{R}_{IB}$
- For right pertubing $\dot{\boldsymbol{R}}_{IB} = \boldsymbol{R}_{IB}[{}_I\omega_{IB}]^\times$
- Constant angular velocity ($\exp[\Delta\alpha]^\times = \delta\boldsymbol{R}(\Delta\alpha)$) $\boldsymbol{R}_{IB}(t + \Delta t) = \exp[\Delta\alpha]^\times \boldsymbol{R}_{IB}(t)$

Quaternions

- For left pertubing $\dot{\boldsymbol{q}}_{IB} = \frac{1}{2}\begin{bmatrix} {}_I\boldsymbol{\omega}_{IB} \\ 0 \end{bmatrix} \otimes \boldsymbol{q}_{IB}$
- For right pertubing $\dot{\boldsymbol{q}}_{IB} = \frac{1}{2}\boldsymbol{q}_{IB} \otimes \begin{bmatrix} {}_B\boldsymbol{\omega}_{IB} \\ 0 \end{bmatrix}$

IMU (Outputs $_S\tilde{\boldsymbol{a}}$ (accel.), $_S\tilde{\boldsymbol{\omega}}$ (rot. accel.))
$_W\dot{\boldsymbol{t}}_S = {}_W\boldsymbol{v}$, $\dot{\boldsymbol{q}}_{WS} = \frac{1}{2}\boldsymbol{q}_{WS} \otimes \begin{bmatrix} {}_S\tilde{\boldsymbol{\omega}} + \boldsymbol{w}_g - \boldsymbol{b}_g \\ 0 \end{bmatrix}$
$_W\dot{\boldsymbol{v}} = \boldsymbol{R}_{WS}\ ({}_S\tilde{\boldsymbol{a}} + \boldsymbol{w}_a - \boldsymbol{b}_a) + {}_W\boldsymbol{g}$ where gray parts only IRL (in theor. models, leave out), with $\dot{\boldsymbol{b}}_g = \boldsymbol{w}_{b_g}$ and $\dot{\boldsymbol{b}}_a = \boldsymbol{w}_{b_a}$
IMU Sensor Model: $\tilde{z} = \boldsymbol{b}_C + s\boldsymbol{M}\boldsymbol{z} + \boldsymbol{b} + \boldsymbol{n} + \boldsymbol{o}$ where bias $\boldsymbol{b}$ and scale s often modled as time-varying state $\dot{b}(t) = \sigma_C n(t)$. $\boldsymbol{b}_C$ const. calib. and $\boldsymbol{n}$ the model.

2.6 Rigid Body Dynamics

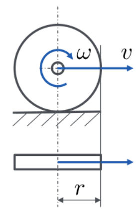
Definition 2.11 *(Newton II)* For fin. body w/ mass m and inertia mat. I, with force $\boldsymbol{F}$ and torque $\boldsymbol{T}$ on ***Centre of Mass*** (CoM), expressed in body frame:

$$_B\boldsymbol{F} = \sum {}_B\boldsymbol{F}_i = m({}_B\dot{\boldsymbol{v}}_{CoM}) + m_B\boldsymbol{\omega} \times {}_B\boldsymbol{v}_{CoM}$$

$$_B\boldsymbol{T} = \sum {}_B\boldsymbol{T}_i = \boldsymbol{I}({}_B\dot{\boldsymbol{\omega}}) + {}_B\boldsymbol{\omega} \times \boldsymbol{I}_B\boldsymbol{\omega}$$

$_B\boldsymbol{v}_{CoM}$ vel. of CoM, $_B\omega$ rot. speed; both w.r.t. world frame

2.7 Wheeled robot Kinematics



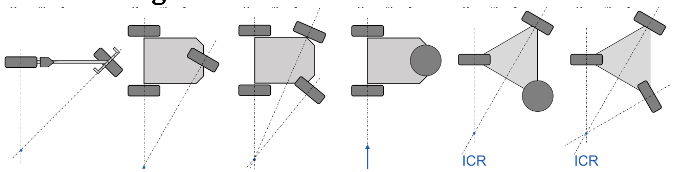
Non-holonomic systems **not integrable**, no inst. move in every direct.
Wheel constraints $v_i = \omega_i r_i$

- *Driving straight* all $\boldsymbol{v}$ equal
- *Turning* Wheel axis must intersect the ***Instant Centre of Rotation*** (ICR), speeds: $v_i \div R_i = \Omega$ (R_i dist. wheel-ICR, Ω, vehicle body rotation rate)

Maneuverability

- Deg. of Mobility: $\delta_m = 3 - \#$constrained directions
- Deg. of Steerability: $\delta_s = \#$steerable wheels
- Deg. of Maneuverability: $\delta_M = \delta_m + \delta_s$

Wheel Configurations



Bicyle	Tricycle	Ackermann	Diff. Drive	Two-Steer	Three-Steer
$\delta_m = 1$	$\delta_m = 1$	$\delta_m = 0$	$\delta_m = 2$	$\delta_m = 1$	$\delta_m = 0$
$\delta_s = 1$	$\delta_s = 1$	$\delta_s = 2$	$\delta_s = 0$	$\delta_s = 2$	$\delta_s = 3$
$\delta_M = 2$	$\delta_M = 2$	$\delta_M = 2$	$\delta_M = 2$	$\delta_M = 3$	$\delta_M = 3$

Differential Drive Kinematics

State vec $\boldsymbol{x} = [x_1, x_2, \theta]^\top$, ***Inputs*** $\boldsymbol{u} = [\omega_l, \omega_r]^\top$, r_r radius of right wheel, w width of robot

Gen. eq. of Motion $\dot{x}_1 = v\cos(\theta)$, $\dot{x}_2 = v\sin(\theta)$, $\dot{\theta} = \Omega$, with $v = 0.5 \cdot (\omega_l r_l + \omega_r + r_r)$, $\Omega = \frac{\omega_r r_r - \omega_l r_l}{w}$

Straight: $v = \omega_l r_l = \omega_r r_r$, $\Omega = 0$, $D = v\Delta t$.

$$\boldsymbol{b}_s = \begin{bmatrix} D\cos(\theta) \\ D\sin(\theta) \\ 0 \end{bmatrix} \quad \boldsymbol{b}_t = \begin{bmatrix} R(\sin(\Delta\theta + \theta) - \sin(\theta)) \\ -R(\cos(\Delta\theta + \theta) - \cos(\theta)) \\ \Delta\theta \end{bmatrix}$$

Turning: $\Omega = (\omega_l r_l)/R_l = (\omega_r r_r)/R_r$, $R = v/\Omega$, $\Delta\theta = \Omega\Delta t$

Discretized: $\boldsymbol{x}_k = \boldsymbol{x}_{k-1} b_i$ with $i \in \{s, t\}$. ($\int \ldots$ dΔt)