TSRT14: Sensor Fusion
Lecture 5

— Sensor and motion models

Gustaf Hendeby

gustaf.hendeby@liu.se

LINKÖPING UNIVERSITY

Lecture 5: sensor and motion models

Whiteboard:

- Principles and some examples

Slides:

- Sampling formulas
- Noise models
- Standard motion models
  - Position as integrated velocity, acceleration, ..., in nD.
  - Orientation as integrated angular speed in 2D and 3D.
- Odometry

Lecture 4: summary

- Detection problems as hypothesis tests:
$$H_0: \mathbf{y} = \mathbf{e},$$
$$H_1: \mathbf{y} = \bar{x} + \mathbf{e} = \mathbf{h}(x) + \mathbf{e}.$$
- Neyman-Pearson's lemma: $T(y) = p_{\mathbf{e}}\big(y - \mathbf{h}(x^0)\big)/p_{\mathbf{e}}(y)$ maximizes P_D for given P_{FA} (best ROC curve).
- In general case
$$\bar{T}(y) = 2 \log p_{\mathbf{e}}\big(y - \mathbf{h}(\hat{x}^{\mathrm{ML}})\big) - 2 \log p_{\mathbf{e}}(y) \sim \chi^2_{n_x}\big(x^{0,T}\,\mathcal{I}(x^0)x^0\big).$$
- Bayes optimal filter
$$p(x_k|y_{1:k}) \propto p_{e_k}\big(y_k - h(x_k)\big)p(x_k|y_{1:k-1})$$
$$p(x_{k+1}|y_{1:k}) = \int p_{v_k}\big(x_{k+1} - f(x_k)\big)p(x_k|y_{1:k})\,dx_k.$$
- Intuitive work flow of nonlinear filter:
  - MU: estimation from $y_k = h(x_k) + e_k$ and fusion with $\hat{x}_{k|k-1}$.
  - TU: nonlinear transformation $z = f(x_k)$ and diffusion from $x_{k-1} = z_k + v_k$.

Modeling and Motion Models

Chapters 12–14 Overview

- Chapter 12: Principles and methods
  - Principles for deriving discrete time models from continuous time ones
  - Discretized-linearization
  - Linearized-discretization
  - Calibration
- Chapter 13: Motion models
  - Kinematics
  - Rotations
  - Vehicle models
  - Examples
- Chapter 14: Sensor models
  - Techniques
  - Examples

Modeling

First problem:
Physics give continuous time model, filters require (linear or nonlinear) discrete time model:

Classification	Nonlinear	Linear
Continuous time	$\dot{x} = a(x,u) + v$ $y = c(x,u) + e$	$\dot{x} = Ax + Bu + v$ $y = Cx + Du + e$
Discrete time	$x_{k+1} = f(x,u) + \bar{v}$ $y = h(x,u) + e$	$x_{k+1} = Fx + Gu + \bar{v}$ $y = Hx + Ju + e$

Sampling Formulas (1/2)

Linear time-invariant (LTI) state-space model:

Continuous time	Discrete time
$\dot{x} = Ax + Bu$	$x_{k+1} = Fx_k + Gu_k$
$y = Cx + Du$	$y_k = Hx_k + Ju_k$

u is either input or process noise (then J denotes cross-correlated noise!).

- **Zero-order hold (ZOH) sampling** assuming the input is piecewise constant:

$$x(t+T) = e^{AT}x(t) + \int_0^T e^{A\tau}Bu(t+T-\tau)\,d\tau$$
$$= \underbrace{e^{AT}}_{F}x(t) + \underbrace{\int_0^T e^{A\tau}\,d\tau\,B}_{G}\,u(t).$$

- **First order hold (FOH) sampling** assuming the input is piecewise linear, is another option.

Sampling Formulas (2/2)

- **Bilinear transformation (BIL)** assumes band-limited input

$$\frac{2}{T}\frac{\Delta - 1}{\Delta + 1}x(t) \approx \tfrac{d}{dt}x(t) = Ax + Bu,$$

where Δ is the delay operator, $\Delta x(t) = x(t+T)$, which yields

$$M = (I_{n_x} - T/2A)^{-1}$$
$$F = M(I_{n_x} + T/2A)$$
$$G = T/2MB$$
$$H = CM$$
$$J = D + HG.$$

Sampling of Nonlinear Models

There are two options:

- Discretized linearization (general):
  1. Linearize:

$$A = \nabla_x a(x,u) \qquad B = \nabla_u a(x,u) \qquad C = \nabla_x c(x,u) \qquad D = \nabla_u c(x,u)$$

  2. Discretize (sample): $F = e^{AT}$, $G = \int_0^T e^{A\tau}\, d\tau\, B$, $H = C$, and $J = D$
- Linearized discretization (best, if possible!):
  1. Discretize (sample nonlinear):

$$x(t+T) = f\big(x(t), u(t)\big) = x(t) + \int_t^{t+T} a\big(x(\tau), u(\tau)\big)\, d\tau$$

  2. Linearize: $F = \nabla_x f(x_k, u_k)$ and $G = \nabla_u f(x_k, u_k)$

Sampling of State Noise

Different solutions exist, they are all approximations except in the linear case:

- v_t is white noise such that its total influence during one sample interval is TQ (alternative (12.14d) in the book):

$$\bar{Q}_d = TQ$$

- v_t is a discrete white noise sequence with variance TQ. That is, we assume that the noise enters immediately after a sample time, so $x(t+T) = f\big(x(t) + v(t)\big)$ (alternative (12.14e) in the book):

$$\bar{Q}_e = TGQG^T$$

Recommendation

In practice simple solutions works well, but *remember to scale with T*!

Motion Models

Continuous time (physical) and discrete time counterparts

$$\dot{x}(t) = a(x(t), u(t), w(t); \theta)$$
$$x(t+T) = f(x(t), u(t), w(t); \theta, T).$$

- **Kinematic models**: Do not attempt to model forces, but are 'Black-box' multi-purpose models.
  - *Translation kinematics* describes position, often based on $F = ma$.
  - *Rotational kinematics* describes orientation.
  - *Rigid body kinematics* combines translational and rotational kinematics.
  - *Constrained kinematics.* Coordinated turns (circular path motion).
- **Application specific force models**
- **Gray-box models** Parameters θ must be calibrated (estimated, identified) from data.

Translational Motion with n Integrators

Translational kinematics models in nD, where $p(t)$ denotes:

- Position: $X(t)$, $\big(X(t), Y(t)\big)^T$, or $\big(X(t), Y(t), Z(t)\big)^T$
- Rotation: $\psi(t)$ or $\big(\phi(t), \theta(t), \psi(t)\big)^T$

The signal $w(t)$ is process noise for a pure kinematic model and a motion input signal in position, velocity, and acceleration, respectively, for the case of using sensed motion as an input rather than as a measurement.

State, $\mathbf{x}$	**Continuous time, $\dot{\mathbf{x}}$**	**Discrete time, $\mathbf{x(t+T)}$**
p	w	$x + Tw$
$\begin{pmatrix} p \\ v \end{pmatrix}$	$\begin{pmatrix} 0_n & I_n \\ 0_n & 0_n \end{pmatrix} x + \begin{pmatrix} 0_n \\ I_n \end{pmatrix} w$	$\begin{pmatrix} I_n & TI_n \\ 0_n & I_n \end{pmatrix} x + \begin{pmatrix} \frac{T^2}{2} I_n \\ TI_n \end{pmatrix} w$
$\begin{pmatrix} p \\ v \\ a \end{pmatrix}$	$\begin{pmatrix} 0_n & I_n & 0_n \\ 0_n & 0_n & I_n \\ 0_n & 0_n & 0_n \end{pmatrix} x + \begin{pmatrix} 0_n \\ 0_n \\ I_n \end{pmatrix} w$	$\begin{pmatrix} I_n & TI_n & \frac{T^2}{2} I_n \\ 0_n & I_n & TI_n \\ 0_n & 0_n & I_n \end{pmatrix} x + \begin{pmatrix} \frac{T^3}{6} I_n \\ \frac{T^2}{2} I_n \\ TI_n \end{pmatrix} w$

Different Sampled Models of Double Integrator

Models

$$\dot{x} = Ax + Bu \qquad x_{k+1} = Fx_k + Gu_k$$
$$y = Cx + Du \qquad y_k = Hx_k + Ju_k$$

State: $x = \begin{pmatrix} p(t) \\ v(t) \end{pmatrix}$

Continuous time	$A = \begin{pmatrix} 0_n & I_n \\ 0_n & 0_n \end{pmatrix}$	$B = \begin{pmatrix} 0_n \\ I_n \end{pmatrix}$	$C = (I_n,\ 0_n)$	$D = 0_n$
ZOH	$F = \begin{pmatrix} I_n & TI_n \\ 0_n & I_n \end{pmatrix}$	$G = \begin{pmatrix} \frac{T^2}{2} I_n \\ TI_n \end{pmatrix}$	$H = (I_n,\ 0_n)$	$J = 0_n$
FOH	$F = \begin{pmatrix} I_n & TI_n \\ 0_n & I_n \end{pmatrix}$	$G = \begin{pmatrix} T^2 I_n \\ TI_n \end{pmatrix}$	$H = (I_n,\ 0_n)$	$J = \frac{T^2}{6} I_n$
BIL	$F = \begin{pmatrix} I_n & TI_n \\ 0_n & I_n \end{pmatrix}$	$G = \begin{pmatrix} \frac{T^2}{4} I_n \\ \text{[illegible]} \end{pmatrix}$	$H = \left(I_n,\ \frac{T}{2} I_n\right)$	$J = \frac{T^2}{2} I_n$

Navigation Models

- Navigation models have access to inertial information.
- 2D orientation (course, or yaw rate) much easier than 3D orientation.

Rotational Kinematics in 2D

The course, or yaw, in 2D can be modeled as integrated white noise

$$\dot{\psi}(t) = w(t),$$

or any higher order of integration. Compare to the tables for translational kinematics with $p(t) = \psi$ and $n = 1$.

Rotational Kinematics in 3D

Much more complicated in 3D than 2D! Could be a course in itself.
Coordinate notation for rotations of a body in local coordinate system (x, y, z) relative to an earth fixed coordinate system:

Motion components	**Rotation Euler angle**	**Angular speed**
Longitudinal forward motion x	Roll ϕ	ω^x
Lateral motion y	Pitch θ	ω^y
Vertical motion z	Yaw ψ	ω^z

Euler Orientation in 3D

An earth fixed vector $\mathbf{g}$ (for instance the gravitational force) is in the body system oriented as $Q\mathbf{g}$, where

$$\begin{aligned} Q &= Q^x_\phi Q^y_\theta Q^z_\psi \\ &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{pmatrix} \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix} \begin{pmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\cos\theta \\ \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\cos\theta \end{pmatrix}. \end{aligned}$$

Note:

The result depends on the order of rotations $Q^x_\phi Q^y_\theta Q^z_\psi$. Here, the xyz rule is used, but there are other options.

Euler Rotation in 3D

When the body rotate with ω, the Euler angles change according to

$$\begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} = \begin{pmatrix} \dot\phi \\ 0 \\ 0 \end{pmatrix} + Q^x_\phi \begin{pmatrix} 0 \\ \dot\theta \\ 0 \end{pmatrix} + Q^x_\phi Q^y_\theta \begin{pmatrix} 0 \\ 0 \\ \dot\psi \end{pmatrix}.$$

The dynamic equation for Euler angles can be derived from this as

$$\begin{pmatrix} \dot\phi \\ \dot\psi \\ \dot\theta \end{pmatrix} = \begin{pmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)\sec(\theta) & \cos(\phi)\sec(\theta) \end{pmatrix} \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix}.$$

Singularities (gimbal lock) when $\theta = \pm\frac{\pi}{2}$, can cause numeric divergence!

Unit Quaternions

- Vector representation: $q = (q^0, q^1, q^2, q^3)^T$.
- Norm constraint of unit quaternion: $\|q\| = q^T q = 1$.
- The quaternion can be interpreted as as an axis angle:

$$q = \begin{pmatrix} \cos(\frac{1}{2}\alpha) \\ \sin(\frac{1}{2}\alpha)\hat v \end{pmatrix},$$

  where q represents a rotation with α around the axis defined by $\hat v$, $\|\hat v\| = 1$.

Pros and Cons

+ No singularity.
+ No 2π ambiguity.
– More complex and non-intuitive algebra.
– The norm must be maintained; this can be handled. by projection or as a virtual measurement with small noise.

Quaternion Orientation in 3D

The orientation of the vector $\mathbf{g}$ in body system is $Q\mathbf{g}$, where

$$\begin{aligned} Q &= \begin{pmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2q_1q_2 - 2q_0q_3 & 2q_0q_2 + 2q_1q_3 \\ 2q_0q_3 + 2q_1q_2 & q_0^2 - q_1^2 + q_2^2 - q_3^2 & -2q_0q_1 + 2q_2q_3 \\ -2q_0q_2 + 2q_1q_3 & 2q_2q_3 + 2q_0q_1 & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{pmatrix} \\ &= \begin{pmatrix} 2q_0^2 + 2q_1^2 - 1 & 2q_1q_2 - 2q_0q_3 & 2q_1q_3 + 2q_0q_2 \\ 2q_1q_2 + 2q_0q_3 & 2q_0^2 + 2q_2^2 - 1 & 2q_2q_3 - 2q_0q_1 \\ 2q_1q_3 - 2q_0q_2 & 2q_2q_3 + 2q_0q_1 & 2q_0^2 + 2q_3^2 - 1 \end{pmatrix}. \end{aligned}$$

Quaternion Rotation in 3D

Rotation with ω gives a dynamic equation for q which can be written in two equivalent forms:

$$\dot{q} = \frac{1}{2}S(\omega)q = \frac{1}{2}\bar{S}(q)\omega,$$

where

$$S(\omega) = \begin{pmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{pmatrix}, \qquad \bar{S}(q) = \begin{pmatrix} -q_1 & -q_2 & -q_3 \\ q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{pmatrix}.$$

Sampled Form of Quaternion Dynamics

The ZOH sampling formula

$$q(t+T) = e^{\frac{1}{2}S\big(\omega(t)\big)T}q(t)$$

actually has a closed form solution

$$\begin{aligned} q(t+T) &= \Big(\cos(\tfrac{T}{2}\|\omega(t)\|)I_4 + \tfrac{T}{2}\overbrace{\frac{\sin(\frac{T}{2}\|\omega(t)\|)}{\frac{T}{2}\|\omega(t)\|}}^{\operatorname{sinc}(\frac{T}{2}\|\omega(t)\|)} S\big(\omega(t)\big)\Big)q(t) \\ &\approx \Big(I_4 + \tfrac{T}{2}S\big(\omega(t)\big)\Big)q(t). \end{aligned}$$

The approximation coincides with Euler forward sampling approximation, and has to be used in more complex models where, *e.g.*, ω is part of the state vector.

Double Integrated Quaternion

$$\begin{pmatrix} \dot{q}(t) \\ \dot{\omega}(t) \end{pmatrix} = \begin{pmatrix} \frac{1}{2}S(\omega(t))q(t) \\ w(t) \end{pmatrix}.$$

There is no known closed form discretized model. However, the approximate form can be discretized using the chain rule to

$$\begin{pmatrix} q(t+T) \\ \omega(t+T) \end{pmatrix} \approx \underbrace{\begin{pmatrix} I_4 \frac{T}{2}S\big(\omega(t)\big) & \frac{T}{2}\bar{S}\big(q(t)\big) \\ 0_{3\times 4} & I_3 \end{pmatrix}}_{F(t)} \begin{pmatrix} q(t) \\ \omega(t) \end{pmatrix} + \underbrace{\begin{pmatrix} \frac{T^3}{4}S\big(\omega(t)\big)I_4 \\ TI_3 \end{pmatrix}}_{G(t)} v(t).$$

Rigid Body Kinematics

A “multi-purpose” model for all kind of navigation problems in 3D (22 states)

$$\begin{pmatrix} \dot{p} \\ \dot{v} \\ \dot{a} \\ \dot{q} \\ \dot{\omega} \\ \dot{b}^{\mathsf{acc}} \\ \dot{b}^{\mathsf{gyro}} \end{pmatrix} = \begin{pmatrix} 0 & I & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{2}S(\omega) & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} p \\ v \\ a \\ q \\ \omega \\ b^{\mathsf{acc}} \\ b^{\mathsf{gyro}} \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} v^a \\ v^\omega \\ v^{\mathsf{acc}} \\ v^{\mathsf{gyro}} \end{pmatrix}.$$

Bias states for the accelerometer and gyroscope have been added as well.

Sensor Model for Kinematic Model

Inertial sensors (gyroscope, accelerometer, magnetometer) are used as sensors.

$$
\begin{aligned}
y_t^{\text{acc}} &= R(q_t)(a_t - \mathbf{g}) + b_t^{\text{acc}} + e_t^{\text{acc}}, & e_t^{\text{acc}} &\sim \mathcal{N}(0, R_t^{\text{acc}}),\\
y_t^{\text{mag}} &= R(q_t)\mathbf{m} + b_t^{\text{mag}} + e_t^{\text{mag}}, & e_t^{\text{mag}} &\sim \mathcal{N}(0, R_t^{\text{mag}}),\\
y_t^{\text{gyro}} &= \omega_t + b_t^{\text{gyro}} + e_t^{\text{gyro}}, & e_t^{\text{gyro}} &\sim \mathcal{N}(0, R_t^{\text{gyro}}).
\end{aligned}
$$

Bias observable, but special calibration routines are recommended:

Stand-still detection: When $\|y_t^{\text{acc}}\| \approx \mathbf{g}$ and/or $\|y_t^{\text{gyro}}\| \approx 0$, the gyro and acc bias is readily read off. Can decrease drift in dead-reckoning from cubic to linear.

Ellipse fitting: When "waving the sensor" over short time intervals, the gyro can be integrated to give accurate orientation, and acc and magnetometer measurements can be transformed to a sphere from an ellipse.

Tracking Models

- Navigation models have access to inertial information, tracking models have not.
- Orientation mainly the direction of the velocity vector.
- Course (yaw rate) critical parameter.
- Less differences between the 2D and 3D cases.

Coordinated Turns in 2D World Coordinates

Cartesian velocity	**Polar velocity**
$\dot X = v^X$ $\dot Y = v^Y$ $\dot v^X = -\omega v^Y$ $\dot v^Y = \omega v^X$ $\dot\omega = 0$	$\dot X = v\cos(h)$ $\dot Y = v\sin(h)$ $\dot v = 0$ $\dot h = \omega$ $\dot\omega = 0$
$A = \begin{pmatrix} 0 & 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & -\omega & -v^Y\\ 0 & 0 & \omega & 0 & v^X\\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$	$A = \begin{pmatrix} 0 & 0 & \cos(h) & -v\sin(h) & 0\\ 0 & 0 & \sin(h) & v\cos(h) & 0\\ 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 1\\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
$X_{t+T} = X + \frac{v^X}{\omega}\sin(\omega T) - \frac{v^Y}{\omega}(1-\cos(\omega T))$ $Y_{t+T} = Y + \frac{v^X}{\omega}(1-\cos(\omega T)) + \frac{v^Y}{\omega}\sin(\omega T)$ $v^X_{t+T} = v^X\cos(\omega T) - v^Y\sin(\omega T)$ $v^Y_{t+T} = v^X\sin(\omega T) + v^Y\cos(\omega T)$ $\omega_{t+T} = \omega$	$X_{t+T} = X + \frac{2v}{\omega}\sin(\frac{\omega T}{2})\cos(h + \frac{\omega T}{2})$ $Y_{t+T} = Y - \frac{2v}{\omega}\sin(\frac{\omega T}{2})\sin(h + \frac{\omega T}{2})$ $v_{t+T} = v$ $h_{t+T} = h + \omega T$ $\omega_{t+T} = \omega$

Automotive Example: Coordinated Turns in 2D Body Coordinates

Basic motion equations

$$
\begin{aligned}
\dot\psi &= \frac{v_x}{R} = v_x R^{-1},\\
a_y &= \frac{v_x^2}{R} = v_x^2 R^{-1} = v_x\dot\psi,\\
a_x &= \dot v_x - v_y\frac{v_x}{R} = \dot v_x - v_y v_x R^{-1} = \dot v_x - v_y\dot\psi.
\end{aligned}
$$

can be combined to a model suitable for the sensor configuration at hand. For instance,

$$
x = \begin{pmatrix}\psi\\ R^{-1}\end{pmatrix},\quad u = v_x,\quad y = R^{-1}
$$

$$
\dot x = f(x,u) + w = \begin{pmatrix} v_x R^{-1}\\ 0\end{pmatrix} + w
$$

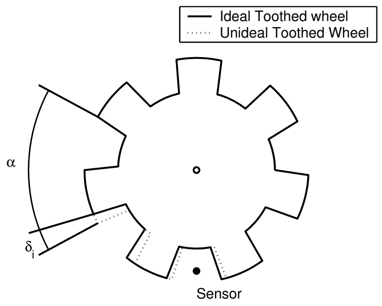
is useful when speed is measured, and a vision system delivers a local estimate of (inverse) curve radius.

Automotive Example: wheel speed sensor

Each tooth passing the sensor (electromagnetic or Hall) gives a pulse. The number n of clock cycles between a number m of teeth are registered within each sample interval.

$$\omega(t_k) = \frac{2\pi}{N_{\mathsf{cog}}(t_k - t_{k-1})} = \frac{2\pi}{N_{\mathsf{cog}} T_c} \frac{m}{n}.$$

Problems:
Angle and time quantization. Synchronization. Angle offsets δ in sensor teeth.



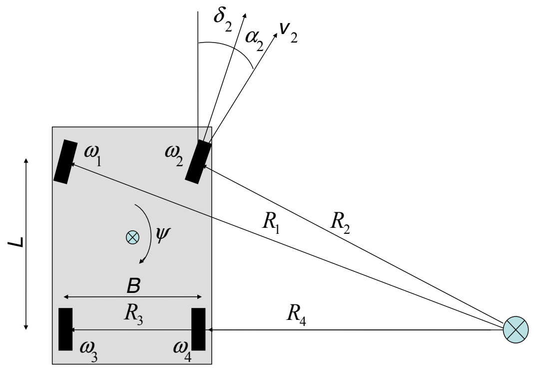
Automotive Example: virtual sensors

Longitudinal velocity, yaw rate and slip on left and right driven wheel (front wheel driven assumed) can be computed from wheel angular speeds **if** the radii are known:

$$\begin{aligned}
v_x &= \frac{v_3 + v_4}{2} = \frac{\omega_3 r_3 + \omega_4 r_4}{2}, \\
\dot{\Psi} &= \frac{\omega_3 r_3 - \omega_4 r_4}{B}, \\
s_1 &= \frac{\omega_1 r_1}{v_1} - 1, \quad s_2 = \frac{\omega_2 r_2}{v_2} - 1, \\
v_1 &= v_x \sqrt{(1 + \tfrac{1}{2} R^{-1} B)^2 + (R^{-1} L)^2}, \\
v_2 &= v_x \sqrt{(1 - \tfrac{1}{2} R^{-1} B)^2 + (R^{-1} L)^2}.
\end{aligned}$$

Automotive Example: geometry

The formulas are based on geometry, the relation $\dot{\psi} = v_x R^{-1}$ and notation below.



Automotive Example: odometry

Odometry is based on the virtual sensors

$$\begin{aligned}
v_x^m &= \frac{\omega_3 r_3 + \omega_4 r_4}{2} \\
\dot{\psi}^m &= v_x^m \frac{2}{B} \frac{\frac{\omega_3}{\omega_4}\frac{r_3}{r_4} - 1}{\frac{\omega_3}{\omega_4}\frac{r_3}{r_4} + 1}.
\end{aligned}$$

and the model

$$\begin{aligned}
\psi_t &= \psi_0 + \int_0^t \dot{\psi}_\tau \, d\tau, \\
X_t &= X_0 + \int_0^t v_\tau^x \cos(\psi_\tau) \, d\tau, \\
Y_t &= Y_0 + \int_0^t v_\tau^x \sin(\psi_\tau) \, d\tau.
\end{aligned}$$

to dead-reckon the wheel speeds to a relative position in the global frame.
The position $(X_t(r_3, r_4), Y_t(r_3, r_4))$ depends on the values of wheel radii r_3 and r_4.
Further sources of error come from wheel slip in longitudinal and lateral direction. More sensors needed for navigation.

Summary

Summary Lecture 5

- Standard models in global coordinates:
  - Translation $p_t^{(m)} = w_t^p$.
  - 2D orientation for heading $h_t^{(m)} = w_t^h$.
  - Coordinated turn model

$$\dot{X} = v^X \qquad \dot{Y} = v^Y$$
$$\dot{v}^X = -\omega v^Y \qquad \dot{v}^Y = \omega v^X$$
$$\dot{\omega} = 0.$$

- Standard models in local coordinates (x, y, ψ):
  - Odometry and dead reckoning for (x, y, ψ)

$$X_t = X_0 + \int_0^t v_\tau^x \cos(\psi_\tau)\, d\tau \qquad Y_t = Y_0 + \int_0^t v_\tau^x \sin(\psi_\tau)\, d\tau$$
$$\psi_t = \psi_0 + \int_0^t \dot{\psi}_\tau\, d\tau.$$

  - Force models for $(\dot{\psi}, a_y, a_x, \dots)$.
  - 3D orientation $\dot{q} = \frac{1}{2} S(\omega) q = \frac{1}{2} \bar{S}(q) \omega$.