technische universität dortmund

Robotics Theory WS 10/11

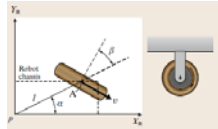
Robot Motion Models

Frank Hoffmann

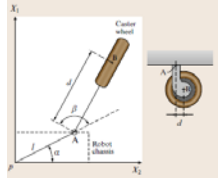
Univ.-Prof. Dr.-Ing. Prof. h.c. Torsten Bertram
Lehrstuhl für Regelungssystemtechnik

Types of Wheels

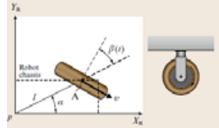
- Passive wheel



- Passive or active off-centered wheel (caster wheel)



- Active orientable wheel



Wheel Encoders

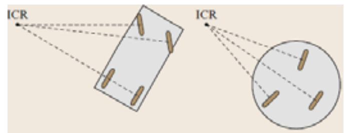
These modules require +5V and GND to power them, and provide a 0 to 5V output. They provide +5V output when they "see" white, and a 0V output when they "see" black.

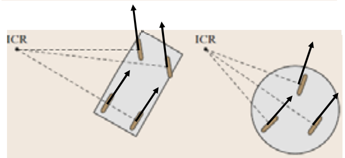
These disks are manufactured out of high quality laminated color plastic to offer a very crisp black to white transition. This enables a wheel encoder sensor to easily see the transitions.

Source: http://www.active-robots.com/

Instantaneous Center of Rotation (ICR)

- At each instant robot motion is regarded as instantaneous rotation around the center of rotation (ICR)
- Geometric construction of the ICR
  - Intersection of normals to the velocity vectors





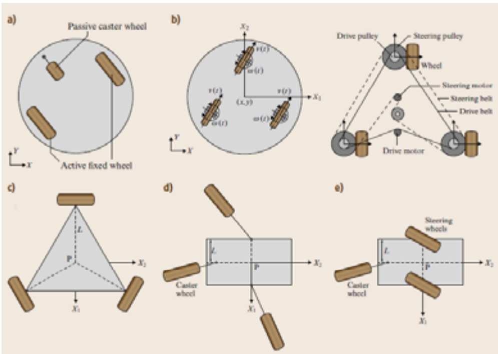
Degree of Mobility and Steerability

- The degree of mobility satisfies $1 \leq \delta m \leq 3$.
- The degree of steerability satisfies $0 \leq \delta s \leq 2$. *T*
- Number of steering wheels that can be oriented independently in order to steer the robot
- The following is satisfied: $2 \leq \delta m + \delta s \leq 3$.

Types of Mobile Robots

- Type (3,0) Robots
  - no fixed and no steering wheels
  - only Swedish or caster wheels
  - *omnimobile,* full mobility in the plane
  - able to move in any direction without any reorientation
- Type (2,0) Robots
  - no steering wheels, but either one or several fixed wheels with a common axle.
  - mobility is restricted to two dimensions
  - differential drive robot

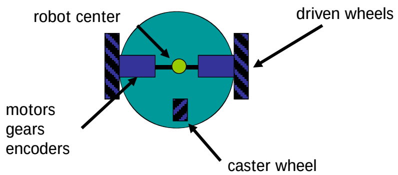
Types of Robots



Types of Mobile Robots

- Type (2,1) Robots
  - no fixed wheels and at least one steering wheel
  - if there is more than one steering wheel, their orientations must be coordinated
- Type (1,1) Robots
  - one or several fixed wheels on a single common axle
  - also one or several steering wheels
  - their centers are not located on the common axle of the fixed wheels, and orientations are coordinated.
- Type (1,2) Robots
  - no fixed wheels
  - at least two steering wheels
  - if there are more than two steering wheels, then their orientation arecoordinated

Differential Drive



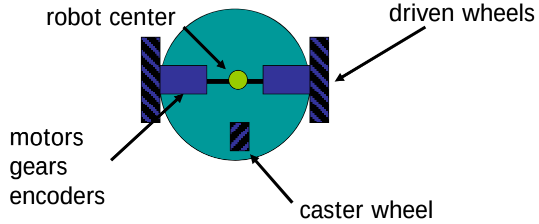
Pros

- simple mechanical structure
- simple kinematic model and low fabrication cost.
- zero turning radius
- systematic errors are easy to calibrate.

Cons

- difficulty of moving irregular surfaces
- On uneven surfaces its orientation might change abruptly if one of the active wheels loses contact with the ground
- only bidirectional movement is available

Differential Drive



$c_m = \pi D_n / nC_e$

c_m = conversion factor that translates encoder pulses into linear wheel displacement

D_n = nominal wheel diameter (in mm)

C_e = encoder resolution (in pulses per revolution)

n = gear ratio of the reduction gear between the motor and the drive wheel.

Differential Drive

Left and right wheel encoders show a pulse increment of N_L and N_R

Incremental travel distance for the left and right wheel, $\Delta U_{L,i}$ and $\Delta U_{R,i}$

$$\Delta U_{L/R,i} = c_m N_{L/R,i}$$

Incremental linear displacement of the robot's center point C, denoted ΔU_i

$$\Delta U_i = (\Delta U_R + \Delta U_L)/2$$

Robot's incremental change of orientation

$$\Delta \theta_i = (\Delta U_R - \Delta U_L)/b$$

where b is the wheelbase of the vehicle.

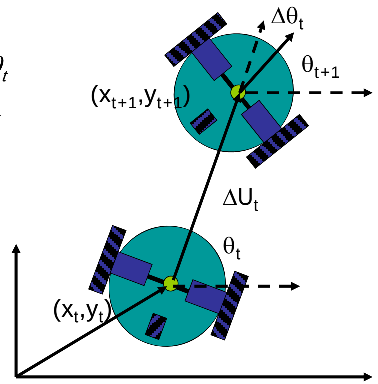
Motion model

- Discrete motion

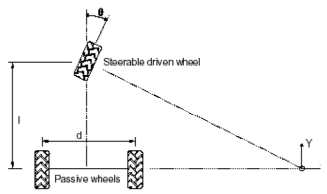
$$x_{t+1} = x_t + \Delta U_t \cos\theta_t$$
$$y_{t+1} = y_t + \Delta U_t \sin\theta_t$$
$$\theta_{t+1} = \theta_t + \Delta\theta_t$$

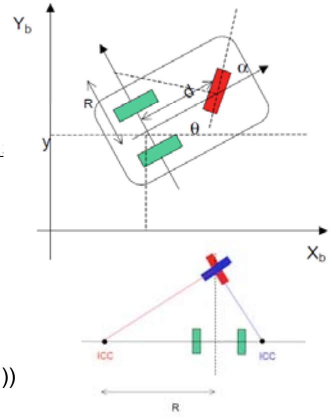
- Continuous motion

$$\dot{x} = v(t)\cos\theta(t)$$
$$\dot{y} = v(t)\sin\theta(t)$$
$$\dot{\theta} = \omega(t)$$



Tricyle Steering





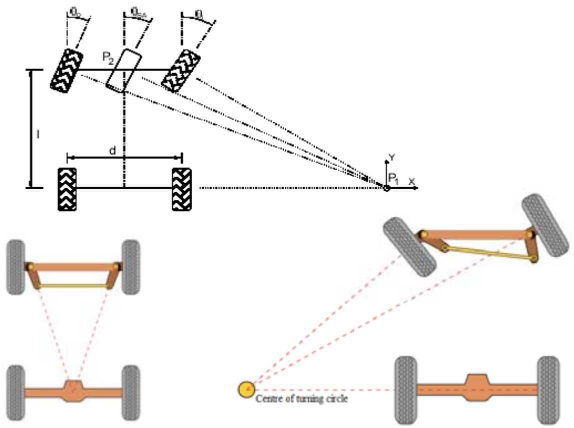
controls ω_s and α_s of steering wheel

$$v_s(t) = \omega_s(t) r$$

$$R(t) = d \tan\left(\frac{\pi}{2} - \alpha(t)\right)$$

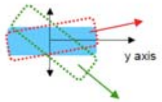
$$\omega(t) = \frac{\omega_s(t) r}{\sqrt{d^2 + R(t)^2}} = \frac{v_s(t)}{d} \sin(\alpha(t))$$

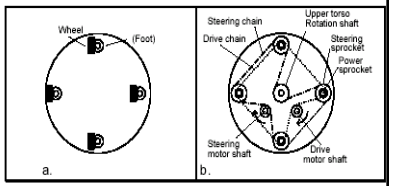
Ackermann Steering

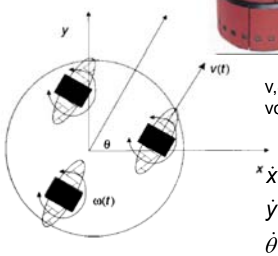


technische universität dortmund

Synchrononous Drive





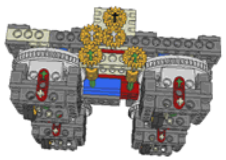


v,ω sind unabhängig voneinander steuerbar

$$\dot{x} = v(t)\cos\theta(t)$$

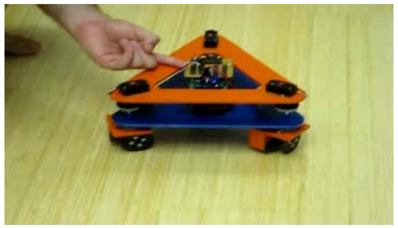
$$\dot{y} = v(t)\sin\theta(t)$$

$$\dot{\theta} = \omega(t)$$



technische universität dortmund

Synchronous Drive



technische universität dortmund

Omnidirektionale Lenkung

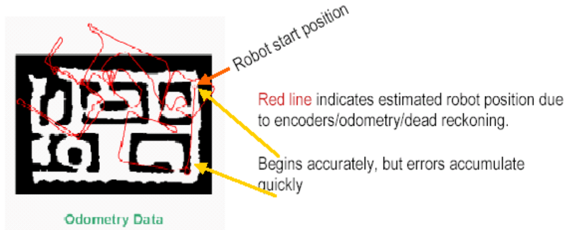
Omnidirectional Drive

Dead Reckoning

- measure turning distance of motors (in terms of numbers of rotations)
- convert to distance of left and right wheel
- convert to robot center translation and rotation
- parameters
  - gearing
  - wheel diameter
  - wheel base

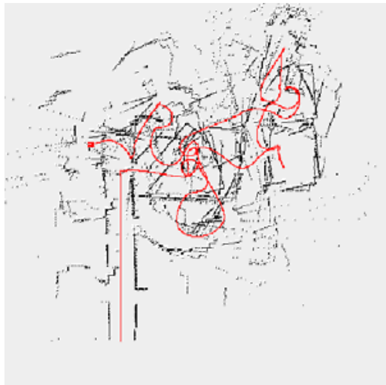
Dead Reckoning Errors

- Challenges/issues:
  - Motion of wheels not corresponding to robot motion, e.g., due to wheel spinning
  - Wheels don't move but robot does, e.g., due to robot sliding
- Error accumulates quickly, especially due do turning:



Dead Reckoning Errors

- Plot of overlaid laser scans overlaid based strictly on odometry:



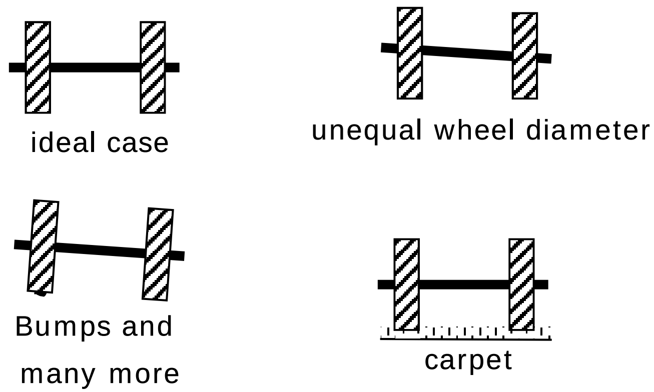
Odometry

- widely used mobile robot positioning.
- good short-term accuracy
- Inexpensive
- high sampling rates.
- integration of incremental motion information over time (dead reckoning).
- Errors accumulate over time
- Orientation errors induce unbounded position errors

Odometry

- Fuse with absolute position measurements to provide better and more reliable position estimation.
- Pose tracking between absolute position updates with landmarks.
- Given a required positioning accuracy, increased accuracy in odometry allows for less frequent absolute position updates.
- Many mapping and landmark matching algorithms assume that the robot can maintain its position well enough to allow the robot to look for landmarks in a limited area and to match features in that limited area to achieve short processing time and to improve matching correctness.

Causes for Odometry Errrors



Systematic Odometry Errors

- unequal wheel diameters.
- average of actual wheel diameters differs from nominal wheel diameter.
- actual wheelbase differs from nominal wheelbase.
- misalignment of wheels.
- finite encoder resolution.
- finite encoder sampling rate.

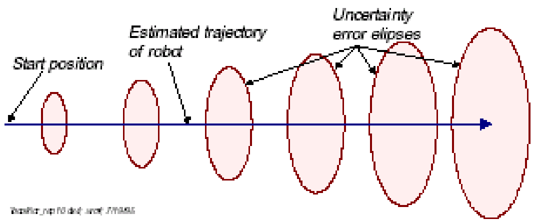
Non- Systematic Odometry Errors

- travel over uneven floors.
- travel over unexpected objects on the floor.
- wheel-slippage due to:
  - slippery floors.
  - overacceleration.
  - fast turning (skidding).
  - external forces (interaction with external bodies).
  - internal forces (castor wheels).
  - non-point wheel contact with the floor.

technische universität dortmund

Odometry Errors

- Error ellipse indicates uncertainty in position estimate
- Error ellipses grow with travel distance
- An absolute position measurement reduces the growing uncertainty and thereby shrinks the error ellipse (Kalman filter approach).



technische universität dortmund

Odometry Errors

- The scaling error due to incorrect average wheel diameter

  $E_s = D_{actual}/D_{nominal}$
- The error due to unequal wheel diameters, defined as

  $E_d = D_R/D_L$,

  where D_R and D_L are the actual wheel diameters of the right and left wheel, respectively.
- The error due to uncertainty about the effective wheel-base, defined as

  $E_b = b_{actual}/b_{nominal}$,

  where b is the wheelbase of the vehicle.

technische universität dortmund

Odometry Errors

- The scaling error Es is easily measured with an ordinary tape measure.
- Let the robot travel along a straight line for a nominal distance based on dead-reckoning $L_{nominal}$ and compare that value with the actual distance L_{actual} between start and end point.
  $E_s = L_{actual} / L_{nominal}$
- Compensation for the linear scaling error is achieved by replacing the nominal wheel diameter with the scaled wheel diameter
  $D_{actual} = E_S \, D_{nominal}$
- Typical accuracy for the scaling error will be E_s<0.03%.

Unidirectional Square-Path Experiment

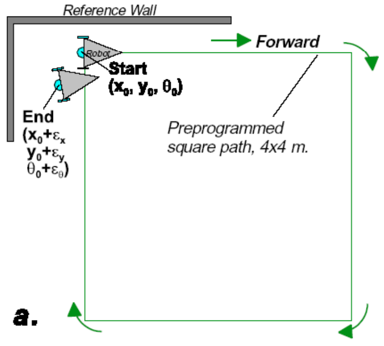
$$\varepsilon_x = x_{abs} - x_{calc}$$
$$\varepsilon_y = y_{abs} - y_{calc}$$
$$\varepsilon_\theta = \theta_{abs} - \theta_{calc}$$

where

$\varepsilon_x, \varepsilon_y, \varepsilon_\theta$ = position and orientation errors due to odometry

$x_{abs}, y_{abs}, \theta_{abs}$ = absolute position and orientation of the robot

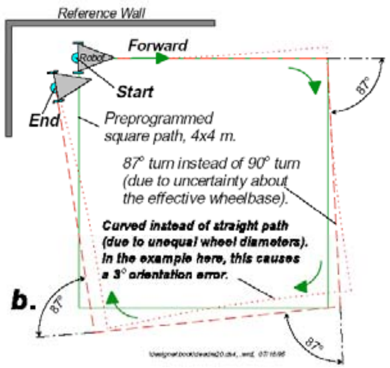
$x_{calc}, y_{calc}, \theta_{calc}$ = position and orientation of the robot as computed from odometry



Unidirectional Square-Path Experiment

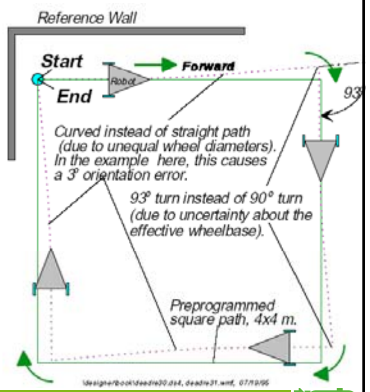
The path is comprised of

- four straight-line segments
- four pure rotations about the robot's center-point



Unidirectional Square-Path Experiment

- In the unidirectional square path experiment the errors due to unequal wheel diameters and incorrect wheelbase might conceal each other.
- The unidirectional square path experiment is not suitable to reliably detect systematic odometry errors.

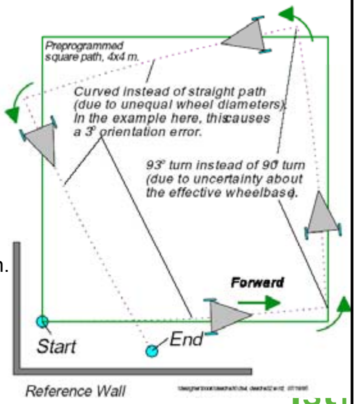


The Bidirectional Square-Path Experiment (UMBmark)

UMBmark requires that the square path experiment be performed in both clockwise and counterclockwise direction.

The concealed dual error becomes clearly visible when the square path is performed in the opposite direction.

The two dominant systematic errors, which may compensate for each other when run in only one direction, add up to each other and increase the overall error when run in the opposite direction.



Bidirectional Square-Path Experiment (UMBmark)

$$x_{c.g.,cw/ccw} = \frac{1}{n}\sum_{i=1}^{n} x_{i,cw/ccw}$$

$$y_{c.g.,cw/ccw} = \frac{1}{n}\sum_{i=1}^{n} y_{i,cw/ccw}$$

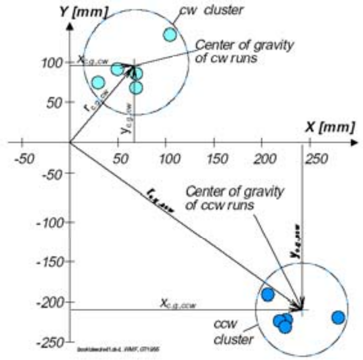
$$r_{c.g.,cw} = \sqrt{(x_{c.g.,cw})^2 + (y_{c.g.,cw})^2}$$

and

$$r_{c.g.,ccw} = \sqrt{(x_{c.g.,ccw})^2 + (y_{c.g.,ccw})^2}$$

Finally, the larger value among $r_{c.g.,cw}$ and $r_{c.g.,ccw}$ is defined as the "measure of odometric accuracy for systematic errors":

$$E_{max,syst} = \max(r_{c.g.,cw}\,;\, r_{c.g.,ccw}).$$

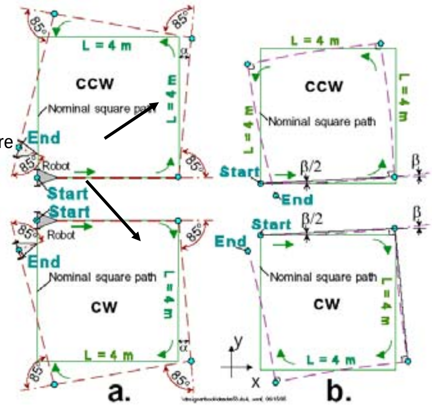


Systematic Calibration

- A Type A error is defined as an orientation error that ***reduces*** (or ***increases***) the total amount of rotation of the robot during the square-path experiment in ***both cw*** and ***ccw direction***.
- By contrast, Type B is defined as an orientation error that ***reduces*** (***or increases***) the total amount of rotation of the robot during the square-path experiment in **one direction**, but ***increases*** (***or reduces***) the amount of rotation when going in the other direction.

Systematic Calibration

Type A errors are caused only by the wheelbase error E_b.



Type B errors are caused only by unequal wheel diameters (E_d).

Systematic Calibration

For wheelbase errors the robot ends up turning ***less*** than the desired amount in ***both the cw and the ccw***

$$\left|\theta_{total,cw}\right| < \left|\theta_{\text{nominal}}\right| \text{ and } \left|\theta_{total,ccw}\right| < \left|\theta_{\text{nominal}}\right|$$

Unequal wheel diameters express themselves in a curved path that **adds** to the overall orientation at the end of the run in **ccw** direction, but it **reduces** the overall rotation in the **cw** direction, i. e.,

$$\left|\theta_{total,ccw}\right| > \left|\theta_{\text{nominal}}\right| \text{ but } \left|\theta_{total,cw}\right| < \left|\theta_{\text{nominal}}\right|$$

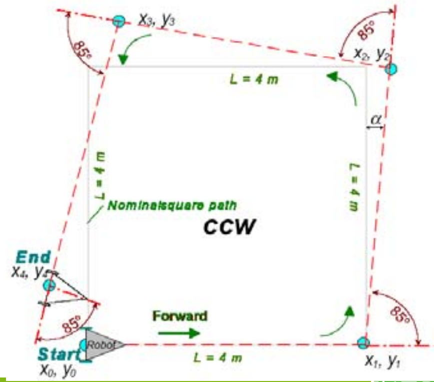
Type A Errors in ccw Direction

$$x_1 = x_0 + L$$
$$y_1 = y_0$$

$$x_2 = x_1 + L\sin\alpha \approx L + L\alpha$$
$$y_2 = y_1 + L\cos\alpha \approx L$$

$$x_3 = x_2 - L\cos 2\alpha \approx L\alpha$$
$$y_3 = y_2 + L\sin 2\alpha \approx L + 2L\alpha$$

$$x_4 = x_3 - L\sin 3\alpha \approx -2L\alpha$$
$$y_4 = y_3 - L\cos 3\alpha \approx 2L\alpha$$



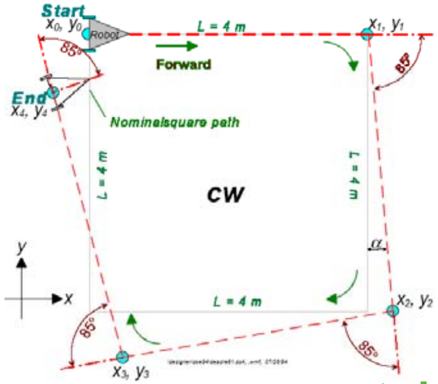
Type A Errors in cw Direction

$$x_1 = x_0 + L$$
$$y_1 = y_0$$

$$x_2 = x_1 + L\sin\alpha \approx L + L\alpha$$
$$y_2 = y_1 + L\cos\alpha \approx -L$$

$$x_3 = x_2 - L\cos 2\alpha \approx L\alpha$$
$$y_3 = y_2 - L\sin 2\alpha \approx -L - 2L\alpha$$

$$x_4 = x_3 - L\sin 3\alpha \approx -2L\alpha$$
$$y_4 = y_3 + L\cos 3\alpha \approx -2L\alpha$$



Systematic Calibration

- Type A errors are caused mostly by E_b.
- Type A errors cause too much or too little turning at the corners of the square path.
- The (unknown) amount of erroneous rotation in each nominal 90-degree turn it denoted as α and measured in [deg].

$$\alpha = \frac{y_{c.g.,cw} - y_{c.g.,ccw}}{-4L}\frac{180°}{\pi}$$

and

$$\alpha = \frac{x_{c.g.,cw} + x_{c.g.,ccw}}{-4L}\frac{180°}{\pi}$$

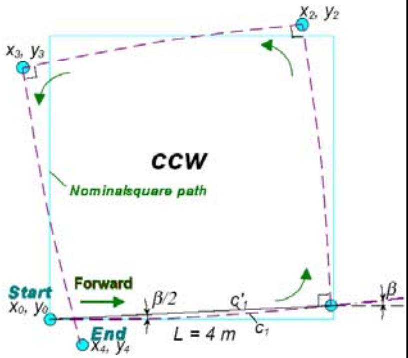
Type B Errors in ccw Direction

$$x_1 = x_0 + L\cos(\beta/2) \approx L$$
$$y_1 = y_0 + L\sin(\beta/2) \approx L\beta/2$$

$$x_2 = x_1 + L\sin(3\beta/2) \approx L - 3L\beta/2$$
$$y_2 = y_1 + L\cos(3\beta/2) \approx L\beta/2 + L$$

$$x_3 = x_2 - L\cos(5\beta/2) \approx -3L\beta/2$$
$$y_3 = y_2 - L\sin(5\beta/2) \approx -2L\beta + L$$

$$x_4 = x_3 + L\sin(7\beta/2) \approx 2L\beta$$
$$y_4 = y_3 - L\cos(7\beta/2) \approx -2L\beta$$



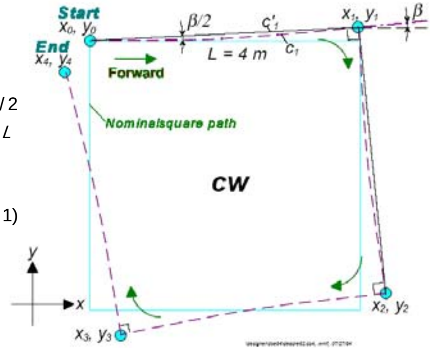
Type B Errors in cw Direction

$$x_1 = x_0 + L\cos(\beta/2) \approx L$$
$$y_1 = y_0 + L\sin(\beta/2) \approx L\beta/2$$

$$x_2 = x_1 + L\sin(3\beta/2) \approx L + 3L\beta/2$$
$$y_2 = y_1 - L\cos(3\beta/2) \approx L\beta/2 - L$$

$$x_3 = x_2 - L\cos(5\beta/2) \approx 3L\beta/2$$
$$y_3 = y_2 - L\sin(5\beta/2) \approx -L(2\beta + 1)$$

$$x_4 = x_3 - L\sin(7\beta/2) \approx -2L\beta$$
$$y_4 = y_3 + L\cos(7\beta/2) \approx -2L\beta$$



Superimposing Type A and B Errors

Superimposing Type A and Type B errors for the cw experiment in x-direction yields

$$\mathbf{x}_{cw} : -2L\alpha - 2L\beta = -2L(\alpha + \beta) = x_{c.g.,cw} \qquad (1)$$

$$\mathbf{x}_{ccw} : -2L\alpha + 2L\beta = -2L(\alpha - \beta) = x_{c.g.,ccw} \qquad (2)$$

Subtracting (2) from (1) yields

$$-4L\beta = x_{c.g.c.w.} - x_{c.g.,ccw}$$

or

$$\beta = \frac{x_{c.g.,cw} - x_{c.g.,ccw}}{-4L}\frac{(180°)}{\pi}$$

for β in degrees.

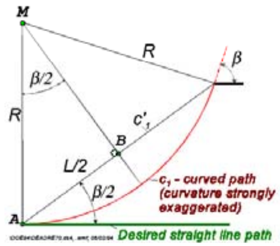
Comparing terms in y-direction yields a similar result

$$\beta = \frac{y_{c.g.,cw} + y_{c.g.,ccw}}{-4L}\frac{(180°)}{\pi}$$

Relation between Error β and Curvature R

Using simple geometric relations, the radius of curvature R of the curved path can be found from the triangle ABM

$$R = \frac{L/2}{\sin(\beta/2)}$$

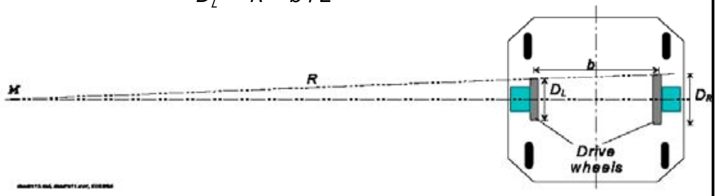


Geometric relations for finding the radius of curvature

Error due to Unequal Wheel Diameter

Once the radius R is computed, it is easy to determine the ratio between the two wheel diameters that caused the robot to travel on a curved, instead of a straight path.

$$E_d = \frac{D_R}{D_L} = \frac{R + b/2}{R - b/2} \qquad (3)$$



Unequal wheel diameters cause the robot to travel on a curved path of radius R

Systematic Calibration

Using simple geometric relations, the radius of curvature R of the curved path of the figure b. can be found as

$$R = \frac{L/2}{\sin \beta /2}.$$

Once the radius R is computed, it is easy to determine the ratio between the two wheel diameters that caused the robot to travel on a curved, instead of a straight path

$$E_d = \frac{D_R}{D_L} = \frac{R + b/2}{R - b/2}.$$

Similarly one can compute the wheelbase error E_b. Since the wheelbase b is directly proportional to the actual amount of rotation, one can use the proportion:

Error due to Incorrect Wheelbase

$$-4L\alpha = x_{c.g.,cw} + x_{c.g.,ccw}$$

$$\alpha = \frac{x_{c.g.,cw} + x_{c.g.,ccw}}{-4L} \frac{(180°)}{\pi}$$

$$\alpha = \frac{y_{c.g.,cw} - y_{c.g.,ccw}}{-4L} \frac{(180°)}{\pi}$$

$$\frac{b_{actual}}{90°} = \frac{b_{nominal}}{90° - \alpha}$$

$$b_{actual} = \frac{90°}{90° - \alpha} b_{nominal}$$

$$E_b = \frac{90°}{90° - \alpha}$$

Systematic Calibration

$$\frac{b_{actual}}{90°} = \frac{b_{nominal}}{90° - \alpha}$$

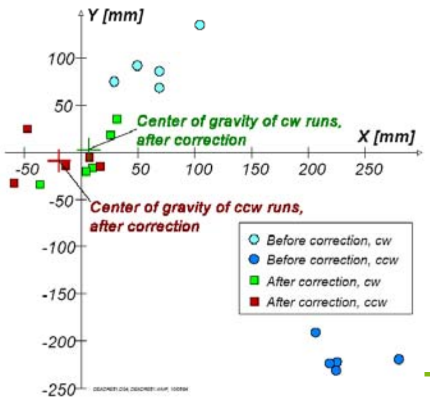
so that

$$b_{actual} = \frac{90°}{90° - \alpha} b_{nominal}$$

where, per definition of $E_b = b_{actual} / b_{nominal}$

$$E_b = \frac{90°}{90° - \alpha}.$$

Odometry Errors Before and After Calibration

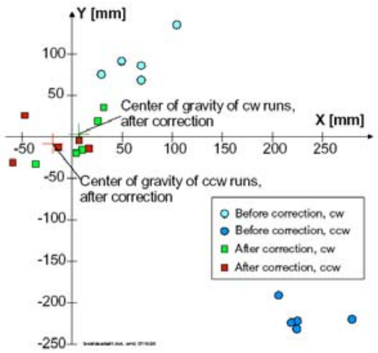


Errors before and after Calibration

Position errors after completion of the bidirectional square-path experiment (4 x 4).

Before calibration: b = 340.00 mm, D_R/D_L = 1.00000.

After Calibration: b = 336.17, D_R/D_L = 1.00084.



Linear Stochastic Process

Discrete linear process described by a stochastic differential equation

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1}$$

Stochastic measurement

$$z_k = Hx_k + v_k$$

Process noise and measurement noise with zero mean and covariances Q and R

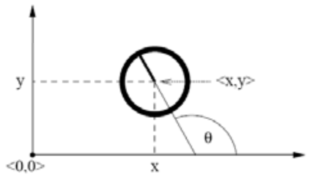
$$p(w) : N(0, Q),$$
$$p(v) : N(0, R).$$

Probabilistic Motion Model

- Implemention of Bayes filter requires a state transition model *p(x | x', u)*.
- The expression *p(x | x', u)* defines the posterior probability, that the control input *u* moves the robot from *x'* to *x*.
- *p(x | x', u)* can be derived from the kinematic equations.

Coordinate Systems

- The general configuration of a robot is described by six degrees of freedom.
- Three-dimensional Cartesian coordinates and three Euler angles.
- Simplifying assumption: the robot moves in a horizontal plane.
- The state space is reduced to three dimensions (x,y,θ).



Motion Models

- In practice, one often finds two types of motion models:
  - **Odometry-based**
  - **Velocity-based** (**dead reckoning**)
- Odometry-based models are used when systems are equipped with wheel encoders.
- Velocity-based models have to be applied when no wheel encoders are given.
- They calculate the new pose based on the velocities and the time elapsed.

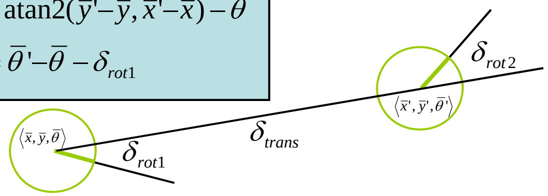
Motion Model

- Robot moves from $\langle \bar{x}, \bar{y}, \bar{\theta} \rangle$ to $\langle \bar{x}', \bar{y}', \bar{\theta}' \rangle$.
- Odometry information $u = \langle \delta_{rot1}, \delta_{rot2}, \delta_{trans} \rangle$.

$$\delta_{trans} = \sqrt{(\bar{x}' - \bar{x})^2 + (\bar{y}' - \bar{y})^2}$$

$$\delta_{rot1} = \mathrm{atan2}(\bar{y}' - \bar{y}, \bar{x}' - \bar{x}) - \bar{\theta}$$

$$\delta_{rot2} = \bar{\theta}' - \bar{\theta} - \delta_{rot1}$$



atan2 Function

- Extends the inverse tangent and correctly copes with the signs of x and y.

$$\mathrm{atan2}(y,x) = \begin{cases} \mathrm{atan}(y/x) & \text{if } x > 0 \\ \mathrm{sign}(y)\,(\pi - \mathrm{atan}(|y/x|)) & \text{if } x < 0 \\ 0 & \text{if } x = y = 0 \\ \mathrm{sign}(y)\,\pi/2 & \text{if } x = 0, y \neq 0 \end{cases}$$

Noise Model for Odometry

- The measured motion is given by the true motion corrupted with noise.

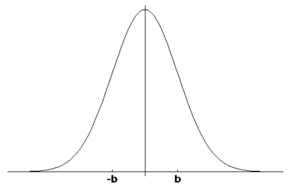
$$\hat{\delta}_{rot1} = \delta_{rot1} + \varepsilon_{\alpha_1|\delta_{rot1}|+\alpha_2|\delta_{trans}|}$$

$$\hat{\delta}_{trans} = \delta_{trans} + \varepsilon_{\alpha_3|\delta_{trans}|+\alpha_4|\delta_{rot1}+\delta_{rot2}|}$$

$$\hat{\delta}_{rot2} = \delta_{rot2} + \varepsilon_{\alpha_1|\delta_{rot2}|+\alpha_2|\delta_{trans}|}$$

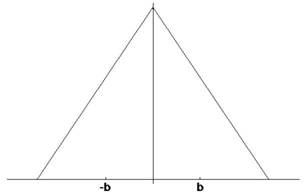
Typical Distributions for Probabilistic Motion Models

Normal Distribution



$$\varepsilon_{\sigma^2}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\frac{x^2}{\sigma^2}}$$

Triangular Distribution



$$\varepsilon_{\sigma^2}(x) = \begin{cases} 0 \text{ if } |x| > \sqrt{6\sigma^2} \\ \frac{\sqrt{6\sigma^2} - |x|}{6\sigma^2} \end{cases}$$

technische universität dortmund

Calculation of Probabilities

- Normal distribution
  1. Algorithm **prob_normal_distribution**(a,b):
  2. return $\frac{1}{\sqrt{2\pi\, b^2}} \exp\left\{-\frac{1}{2}\frac{a^2}{b^2}\right\}$
- Triangular distribution
  1. Algorithm **prob_triangular_distribution**(a,b):
  2. return $\max\left\{0, \frac{1}{\sqrt{6}\, b} - \frac{|a|}{6\, b^2}\right\}$

technische universität dortmund

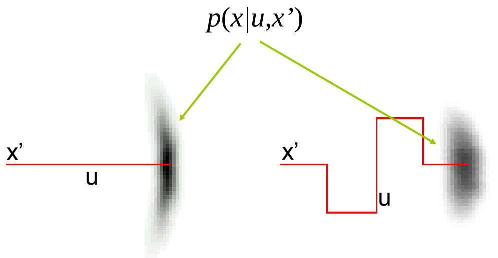
Calculating the Posterior given x, x', and u

1. Algorithm **motion_model_odometry(x,x',u)**
2. $\delta_{trans} = \sqrt{(\bar{x}'-\bar{x})^2 + (\bar{y}'-\bar{y})^2}$
3. $\delta_{rot1} = \text{atan2}(\bar{y}'-\bar{y}, \bar{x}'-\bar{x}) - \bar{\theta}$ — Odometry values (u)
4. $\delta_{rot2} = \bar{\theta}' - \bar{\theta} - \delta_{rot1}$
5. $\hat{\delta}_{trans} = \sqrt{(x'-x)^2 + (y'-y)^2}$
6. $\hat{\delta}_{rot1} = \text{atan2}(y'-y, x'-x) - \bar{\theta}$ — Values of interest (x,x')
7. $\hat{\delta}_{rot2} = \theta' - \theta - \hat{\delta}_{rot1}$
8. $p_1 = \text{prob}(\delta_{rot1} - \hat{\delta}_{rot1}, \alpha_1 |\hat{\delta}_{rot1}| + \alpha_2 \hat{\delta}_{trans})$
9. $p_2 = \text{prob}(\delta_{trans} - \hat{\delta}_{trans}, \alpha_3 \hat{\delta}_{trans} + \alpha_4(|\hat{\delta}_{rot1}| + |\hat{\delta}_{rot2}|))$
10. $p_3 = \text{prob}(\delta_{rot2} - \hat{\delta}_{rot2}, \alpha_1 |\hat{\delta}_{rot2}| + \alpha_2 \hat{\delta}_{trans})$
11. return $p_1 \cdot p_2 \cdot p_3$

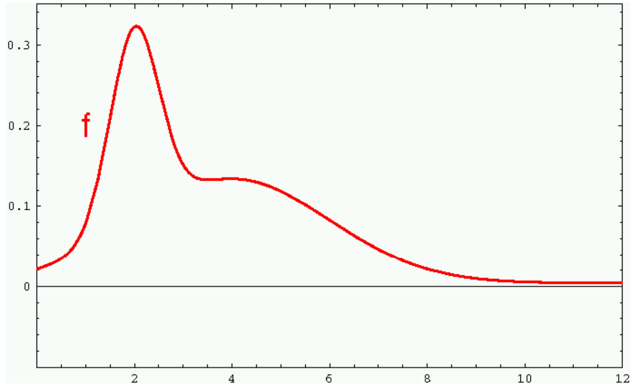
technische universität dortmund

Repeated Application

- Repeated application of the motion model for short movements.
- Typical banana-shaped distributions obtained for 2d-projection of 3d posterior.

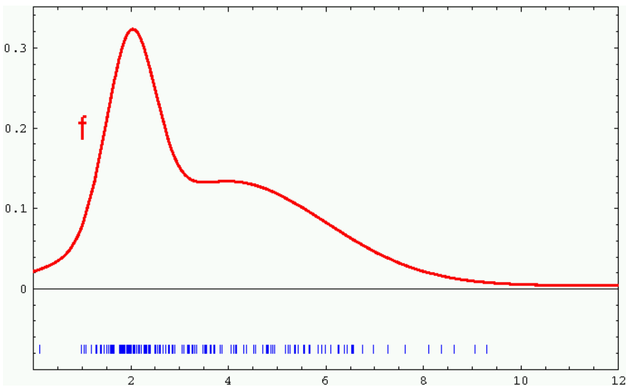


Sample Based Probability Representation



technische universität dortmund

Sample Based Probability Representation



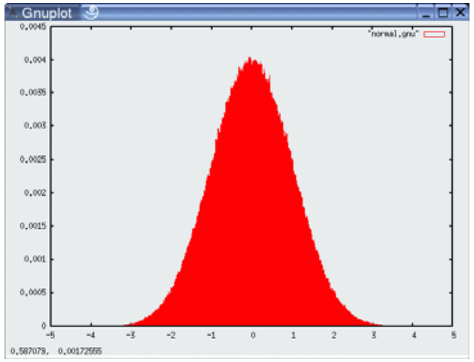
technische universität dortmund

How to generate samples from a normal or triangular distribution?

- Sample from a normal distribution
  1. Algorithm **sample_normal_distribution**(b):
  2. return $\frac{1}{2}\sum_{i=1}^{12} rand(-b,b)$
- Sample from a triangular distribution
  1. Algorithm **sample_triangular_distribution**(b):
  2. return $\frac{\sqrt{6}}{2}\,[\mathrm{rand}(-b,b)+\mathrm{rand}(-b,b)]$

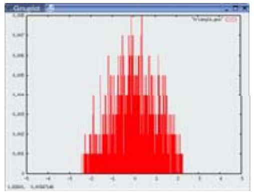
technische universität dortmund

Normally Distributed Samples

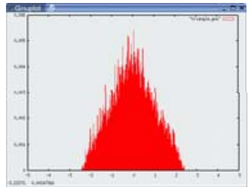


10^6 samples

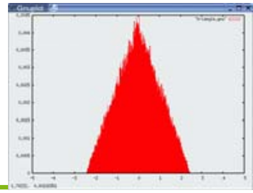
Triangular Distribution



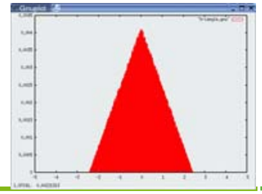
10^3 samples



10^4 samples



10^5 samples



10^6 samples

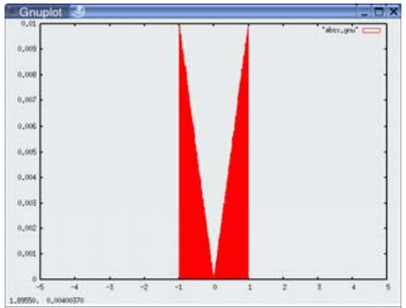
Rejection Based Sampling

- Sample from arbitrary distributions

1. Algorithm **sample_distribution**(f,b):
2. repeat
3. $\quad x = \mathrm{rand}(-b, b)$
4. $\quad y = \mathrm{rand}(0, \max\{f(x) \mid x \in (-b, b)\})$
5. until $(y \leq f(x))$
6. return x

Rejection Based Sampling Example

- Sampling from $f(x) = \begin{cases} \mathrm{abs}(x) & x \in [-1; 1] \\ 0 & \text{otherwise} \end{cases}$



Sampling Odometry Model

1. Algorithm **sample_motion_model**(u, x):

   $u=\langle\delta_{rot1},\delta_{rot2},\delta_{trans}\rangle, x=\langle x,y,\theta\rangle$

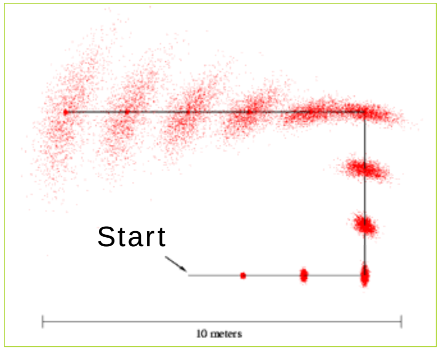
1. $\hat{\delta}_{rot1}=\delta_{rot1}+\text{sample}(\alpha_1|\delta_{rot1}|+\alpha_2\,\delta_{trans})$
2. $\hat{\delta}_{trans}=\delta_{trans}+\text{sample}(\alpha_3\,\delta_{trans}+\alpha_4\,(|\delta_{rot1}|+|\delta_{rot2}|))$
3. $\hat{\delta}_{rot2}=\delta_{rot2}+\text{sample}(\alpha_1|\delta_{rot2}|+\alpha_2\,\delta_{trans})$

4. $x'=x+\hat{\delta}_{trans}\cos(\theta+\hat{\delta}_{rot1})$
5. $y'=y+\hat{\delta}_{trans}\sin(\theta+\hat{\delta}_{rot1})$
6. $\theta'=\theta+\hat{\delta}_{rot1}+\hat{\delta}_{rot2}$

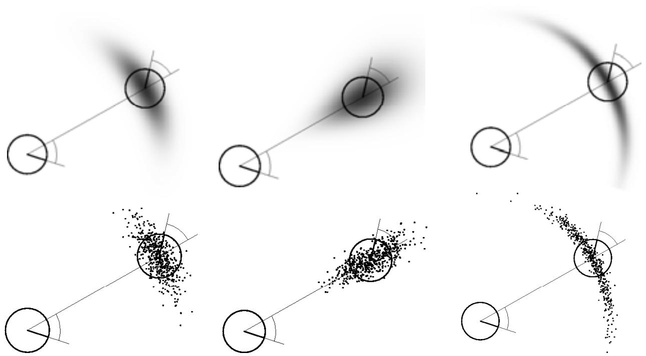
7. Return $\langle x',y',\theta'\rangle$

sample_normal_distribution

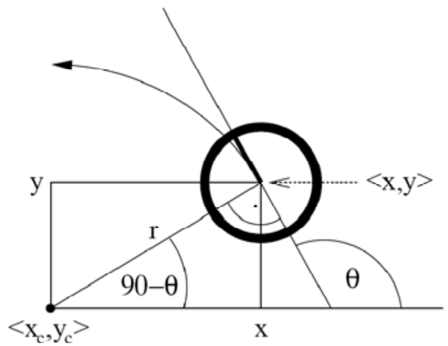
Sampling Motion Model



Examples Motion Odometry Based Model



Velocity Based Motion Model



Velocity Based Motion Model

Circle center:

$$\begin{pmatrix} x^* \\ y^* \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} -\lambda \sin\theta \\ \lambda \cos\theta \end{pmatrix} = \begin{pmatrix} \frac{x+x'}{2} + \mu(y-y') \\ \frac{y+y'}{2} + \mu(x'-x) \end{pmatrix}$$

with

$$\mu = \frac{1}{2} \frac{(x-x')\cos\theta + (y-y')\sin\theta}{(y-y')\cos\theta - (x-x')\sin\theta}$$

Posterior Probability for Velocity Model

Algorithm motion_model_velocity(x_t, u_t, x_{t-1})**:**

$\mu = \frac{1}{2} \frac{(x-x')\cos\theta + (y-y')\sin\theta}{(y-y')\cos\theta - (x-x')\sin\theta}$

$x^* = \frac{x+x'}{2} + \mu(y-y')$

$y^* = \frac{y+y'}{2} + \mu(x'-x)$

$r^* = \sqrt{(x-x^*)^2 + (y-y^*)^2}$

$\Delta\theta = \operatorname{atan2}(y'-y^*, x'-x^*) - \operatorname{atan2}(y-y^*, x-x^*)$

$\hat{v} = \frac{\Delta\theta}{\Delta t} r^*$

$\hat{\omega} = \frac{\Delta\theta}{\Delta t}$

$\hat{\gamma} = \frac{\theta'-\theta}{\Delta t} - \hat{\omega}$

return $\mathbf{prob}(v-\hat{v}, \alpha_1|v| + \alpha_2|\omega|) \cdot \mathbf{prob}(\omega-\hat{\omega}, \alpha_3|v| + \alpha_4|\omega|)$
$\cdot\, \mathbf{prob}(\hat{\gamma}, \alpha_5|v| + \alpha_6|\omega|)$

Sampling Velocity Based Model

Algorithm sample_motion_model_velocity(u_t, x_{t-1})**:**

$\hat{v} = v + \mathbf{sample}(\alpha_1|v| + \alpha_2|\omega|)$

$\hat{\omega} = \omega + \mathbf{sample}(\alpha_3|v| + \alpha_4|\omega|)$

$\hat{\gamma} = \mathbf{sample}(\alpha_5|v| + \alpha_6|\omega|)$

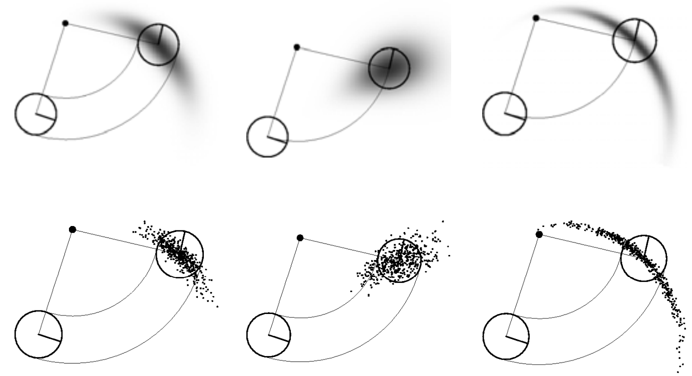
$x' = x - \frac{\hat{v}}{\hat{\omega}} \sin\theta + \frac{\hat{v}}{\hat{\omega}} \sin(\theta + \hat{\omega}\Delta t)$

$y' = y + \frac{\hat{v}}{\hat{\omega}} \cos\theta - \frac{\hat{v}}{\hat{\omega}} \cos(\theta + \hat{\omega}\Delta t)$

$\theta' = \theta + \hat{\omega}\Delta t + \hat{\gamma}\Delta t$

return $x_t = (x', y', \theta')^T$

Distributions Velocity based Motion Model



Map Consistent Motion Model

$$p(x \mid u, x') \neq p(x \mid u, x', m)$$

Approximation: $p(x \mid u, x', m) = \eta \; p(x \mid m) \; p(x \mid u, x')$

Maximum Likelihood Estimate

- Distribution $\quad p(x) = \frac{1}{(2\pi\sigma)^{1/2}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$
- Samples $\quad \{x_1, \ldots, x_n\}$
- Likelihood $\quad p(\{x_1, \ldots, x_n\} \mid \mu, \sigma) = \prod_i \frac{1}{(2\pi\sigma)^{1/2}} e^{\frac{-(x_i-\mu)^2}{2\sigma^2}}$
- Maximum Likelihood $\quad \arg\max_{\mu,\sigma} p(\{x_1, \ldots, x_n\} \mid \mu, \sigma) = \prod_i \frac{1}{(2\pi\sigma)^{1/2}} e^{\frac{-(x_i-\mu)^2}{2\sigma^2}}$

technische universität dortmund

Maximum Likelihood Estimate

- Maximum Likelihood

$$\arg\max_{\mu,\sigma} p(\{x_1, \ldots, x_n\} \mid \mu, \sigma) = \prod_i \frac{1}{(2\pi\sigma)^{1/2}} e^{\frac{-(x_i-\mu)^2}{2\sigma^2}}$$

- Maximum Log Likelihood

$$\arg\max_{\mu,\sigma} \ln p(\{x_1, \ldots, x_n\} \mid \mu, \sigma) = 1/2 \ln 2\pi\sigma + \sum \frac{-(x_i-\mu)^2}{2\sigma^2}$$

$$\frac{\partial \ln p}{\partial \mu} = \sum_i \frac{-(x_i-\mu)}{\sigma^2} = 0 \qquad\qquad \mu = \frac{1}{n}\sum_i x_i$$

$$\frac{\partial \ln p}{\partial \sigma^2} = -\frac{1}{2\sigma^2} + \sum_i \frac{(x_i-\mu)^2}{2\sigma^{2^2}} = 0 \quad \longrightarrow \quad \sigma^2 = \frac{1}{n}\sum_i (x_i-\mu)^2$$

technische universität dortmund

Maximum Likelihood Estimate

- Distribution $\quad p(x \mid d) = \frac{1}{(2\pi d\sigma)^{1/2}} e^{\frac{-(x-d\mu)^2}{d^2\sigma^2}}$
- Samples $\quad \{(x_1, d_1), \ldots, (x_n, d_n)\}$
- Likelihood $\quad p(\{x_1, \ldots, x_n\} \mid \mu, \sigma, \{d_1, \ldots, d_n\}) = \prod_i \frac{1}{(2\pi d_i\sigma)^{1/2}} e^{\frac{-(x_i-d_i\mu)^2}{d_i^2\sigma^2}}$
- Maximum Likelihood

$$\arg\max_{\mu,\sigma} p(\{x_1, \ldots, x_n\} \mid \mu, \sigma, \{d_1, \ldots, d_n\}) = \prod_i \frac{1}{(2\pi d_i\sigma)^{1/2}} e^{\frac{-(x_i-d_i\mu)^2}{d_i^2\sigma^2}}$$

maximize log likelihood $\longrightarrow$

$$\mu = \frac{1}{n}\sum_i \frac{x_i}{d_i}$$

$$\sigma^2 = \frac{1}{n}\sum_i \frac{(x_i-d_i\mu)^2}{d_i^2}$$

technische universität dortmund

Maximum Likelihood Estimate

- Distribution $$p(x\,|\,d,t) = \frac{1}{\left(2\pi\left(d^2\sigma_d{}^2 + t^2\sigma_t{}^2\right)^{1/2}\right)^{1/2}} e^{\frac{-(x-d\mu_d - t\mu_t)^2}{d^2\sigma_d{}^2 + t^2\sigma_t{}^2}}$$
- Samples $$\{(x_1,d_1,t_1),\ldots,(x_n,d_n,t_n)\}$$
- Likelihood $$p(\{x_1,\ldots,x_n\}\,|\,\mu_d,\sigma_d,\mu_t,\sigma_t,\{d_1,\ldots,d_n\},\{t_1,\ldots,t_n\})$$ $$= \prod_i \frac{1}{\left(2\pi\left(d_i{}^2\sigma_d{}^2 + t_i{}^2\sigma_t{}^2\right)^{1/2}\right)^{1/2}} e^{\frac{-(x-d_i\mu_d - t_i\mu_t)^2}{d_i{}^2\sigma_d{}^2 + t_i{}^2\sigma_t{}^2}}$$
- Maximum Likelihood $$\underset{\mu_d,\sigma_d,\mu_t,\sigma_t}{\arg\max}\; p(\{x_1,\ldots,x_n\}\,|\,\mu_d,\sigma_d,\mu_t,\sigma_t,\{d_1,\ldots,d_n\},\{t_1,\ldots,t_n\})$$ $$= \underset{\mu_d,\sigma_d,\mu_t,\sigma_t}{\arg\max} \prod_i \frac{1}{N} e^{\frac{-(x-d_i\mu_d - t_i\mu_t)}{d_i{}^2\sigma_d{}^2 + t_i{}^2\sigma_t{}^2}}$$

Linear Least Squares

System of p linear equations in q unknowns:

$$\begin{cases} u_{11}x_1 + u_{12}x_2 + \cdots + u_{1q}x_q = y_1 \\ u_{21}x_1 + u_{22}x_2 + \cdots + u_{2q}x_q = y_2 \\ \cdots \\ u_{p1}x_1 + u_{p2}x_2 + \cdots + u_{pq}x_q = y_p \end{cases} \Leftrightarrow \mathcal{U}\mathbf{x} = \mathbf{y}$$

with

$$\mathcal{U} = \begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1q} \\ u_{21} & u_{22} & \cdots & u_{2q} \\ \cdots & \cdots & \cdots & \cdots \\ u_{p1} & u_{p2} & \cdots & u_{pq} \end{pmatrix}, \quad \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \cdots \\ x_q \end{pmatrix} \quad und \quad \mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \cdots \\ y_p \end{pmatrix}.$$

Linear Least Squares

- For $p < q$ the solutions span a $(q - p)$-dimensional subspace
- For $p = q$ there is in $\mathbb{R}^q$ general a unique solution.
- For p > q there is in general no solution.

For the over-constrained case $p > q$ find a vector $\mathbf{x}$, which minimizes the quadratic error.

$$E \overset{def}{=} \sum_{i=1}^{p} (u_{i1}x_1 + \cdots + u_{iq}x_q - y_1)^2 = |\mathcal{U}\mathbf{x} - \mathbf{y}|^2.$$

Vector representation: $E = \mathbf{e}\cdot\mathbf{e},\ \mathbf{e} \overset{def}{=} \mathcal{U}\mathbf{x} - \mathbf{y}$.

Linear Least Squares

At a minimum of E its partial derivative of the error with respect to the parameters x_i vanishes:

$$\frac{\partial E}{\partial x_i} = 2\frac{\partial \mathbf{e}}{\partial x_i}\cdot\mathbf{e} = 0 \quad \text{für} \quad i = 1,\ldots,q.$$

Denoting column vectors by $\mathbf{c}_j = (u_{1j},\ldots,u_{mj})^T$, results in

$$\frac{\partial \mathbf{e}}{\partial x_i} = \frac{\partial}{\partial x_i}\left[\begin{pmatrix} \mathbf{c}_1 & \cdots & \mathbf{c}_q \end{pmatrix}\begin{pmatrix} x_1 \\ \cdots \\ x_q \end{pmatrix} - \mathbf{y}\right] = \frac{\partial}{\partial x_1}(x_1\mathbf{c}_1 + \cdots + x_q\mathbf{c}_q) = \mathbf{c}_i.$$

Linear Least Squares

From $\partial E / \partial x_i = 0$ one obtains $\mathbf{c}_i^T(\mathcal{U}\mathbf{x} - \mathbf{y}) = 0.$

The normalisation equation für the minimum quadratic error becomes

$$\mathbf{0} = \begin{pmatrix} \mathbf{c}_1^T \\ \cdots \\ \mathbf{c}_q^T \end{pmatrix} (\mathcal{U}\mathbf{x} - \mathbf{y}) = \mathcal{U}^T(\mathcal{U}\mathbf{x} - \mathbf{y}) \Leftrightarrow \mathcal{U}^T\mathcal{U}\mathbf{x} = \mathcal{U}^T\mathbf{y}$$

If $\mathcal{U}$ has maximal rang q, the matrix $\mathcal{U}^T\mathcal{U}$ is invertable. There is a unique solution $\mathbf{x} = \mathcal{U}^\dagger\mathbf{y}$ with the pseudo-inverse matrix

$$\mathcal{U}^\dagger \overset{def}{=} [(\mathcal{U}^T\mathcal{U})^{-1}\mathcal{U}^T].$$

Note:

- no need to compute pseudo inverse explicitly
- efficient solution by singular value decomposition (SVD).

Maximum Likelihood Estimate

$\boldsymbol{\mu} = [\mu_d, \mu_t],\ \boldsymbol{\sigma}_c^{\ 2} = [\sigma_d^{\ 2}, \sigma_t^{\ 2}]$ parameters:

$\mathbf{O} = [d_i, t_i],\ \mathbf{O}^2 = [d_i^{\ 2}, t_i^{\ 2}]$ odometric motion:

$\mathbf{C} = [x_1, ..., x_n],\ \mathbf{C}^{\sigma^2} = (\mathbf{O}\boldsymbol{\mu}_\mathbf{c} - \mathbf{C})^2,$ observed motions:

Maximum likelihood:

$$\underset{\mu_d, \sigma_d, \mu_t, \sigma_t}{\arg\max}\ p(\{x_1, \ldots, x_n\} \mid \mu_d, \sigma_d, \mu_t, \sigma_t, \{d_1, \ldots, d_n\}, \{t_1, \ldots, t_n\})$$

$$= \underset{\substack{\mu_d, \sigma_d, \mu_t, \sigma_t \\ i}}{\arg\max\ \Pi} \frac{1}{\left(2\pi\left(d_i^{\ 2}\sigma_d^{\ 2} + t_i^{\ 2}\sigma_t^{\ 2}\right)^{1/2}\right)^{1/2}} e^{\frac{-(x - d_i\mu_d - t_i\mu_t)^2}{d_i^{\ 2}\sigma_d^{\ 2} + t_i^{\ 2}\sigma_t^{\ 2}}}$$

Least squares problem:

mean: $\mathbf{O}\boldsymbol{\mu}_c = \mathbf{C} \quad \boldsymbol{\mu}_c = \mathbf{O}^\dagger\mathbf{C}$

variance: $\mathbf{O}^2\sigma_c^{\ 2} = \mathbf{C}^{\sigma 2} \quad \sigma_c^{\ 2} = \mathbf{O}^{2^\dagger}\mathbf{C}^{\sigma 2}$

Summary

- We discussed motion models for odometry-based and velocity-based systems
- We discussed ways to calculate the posterior probability $p(x|\ x', u)$.
- We also described how to sample from $p(x|\ x', u)$.
- Typically the calculations are done in fixed time intervals Δt.
- In practice, the parameters of the models have to be learned.
- We also discussed an extended motion model that takes the map into account.

technische universität dortmund

Vielen Dank für Ihre Aufmerksamkeit!

Univ.-Prof. Dr.-Ing. Prof. h.c. Torsten Bertram
Lehrstuhl für Regelungssystemtechnik