Sensor fusion using world models

Thomas Schön
Division of Automatic Control
Linköping University
Sweden

users.isy.liu.se/rt/schon

Joint work with (in alphabetical order): **Fredrik Gustafsson** (LiU), **Jeroen Hol** (Xsens), **Rickard Karlsson** (Nira Dynamics), **Johan Kihlberg** (Semcon), **Manon Kok** (LiU), **Henk Luinge** (Xsens), **Per-Johan Nordlund** (Saab), **Daniel Roetenberg** (Xsens), **Per Skoglar** (SenionLab), **Simon Tegelid** (Xdin), **Niklas Wahlström** (LiU).

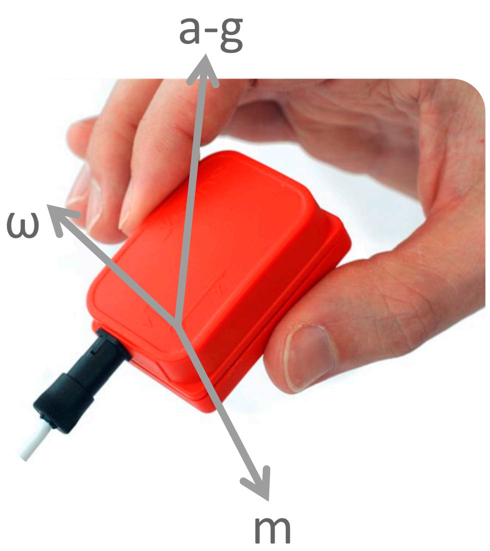
Introductory example (I/III)

Aim: Motion capture, find the motion (position, orientation, velocity and acceleration) of a person (or object) over time.

Industrial partner: Xsens Technologies.

Sensors used:

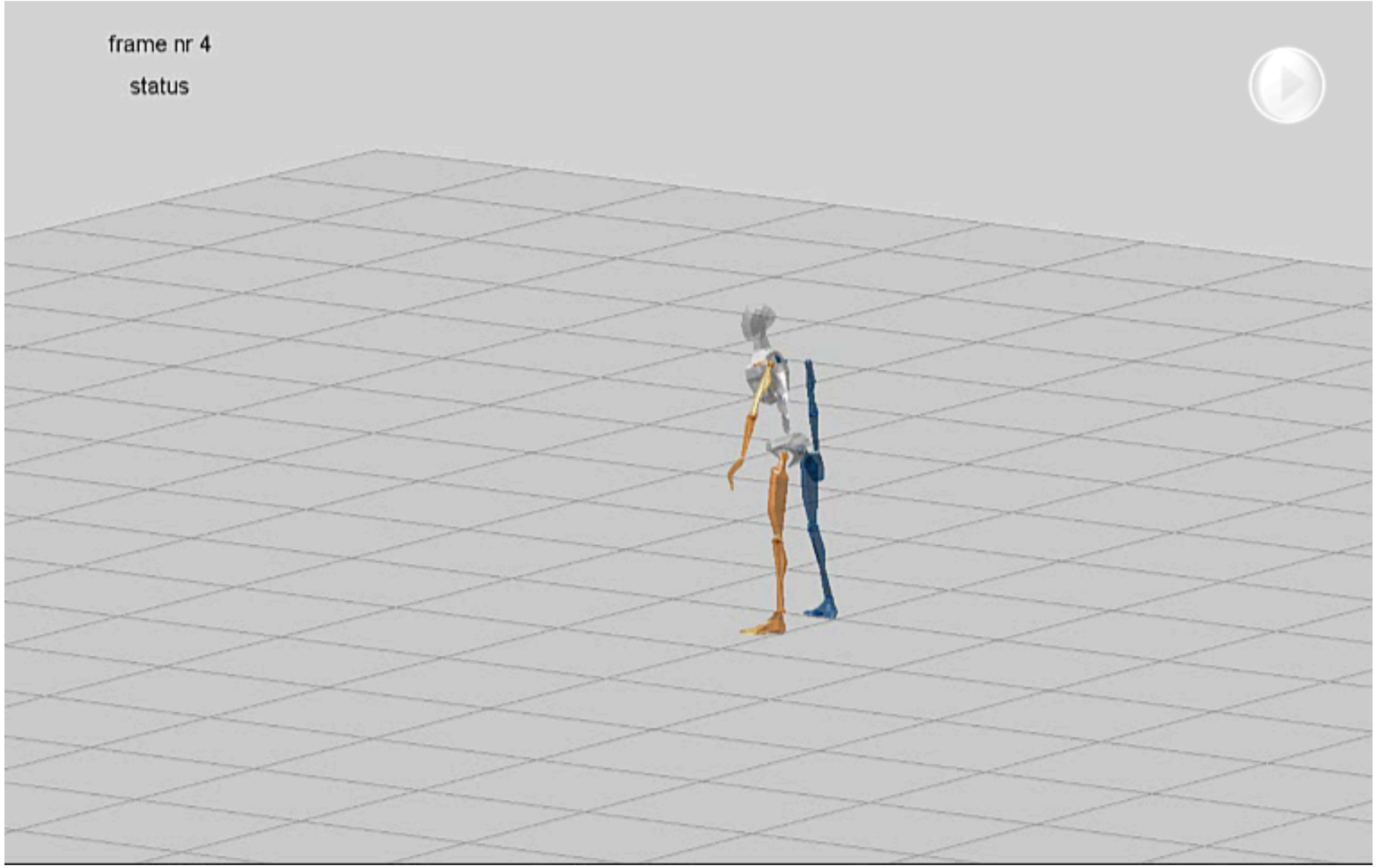
- 3D accelerometer (acceleration)
- 3D gyroscope (angular velocity)
- 3D magnetometer (magnetic field)



17 sensor units are mounted onto the body of the person.

Introductory example (II/III)

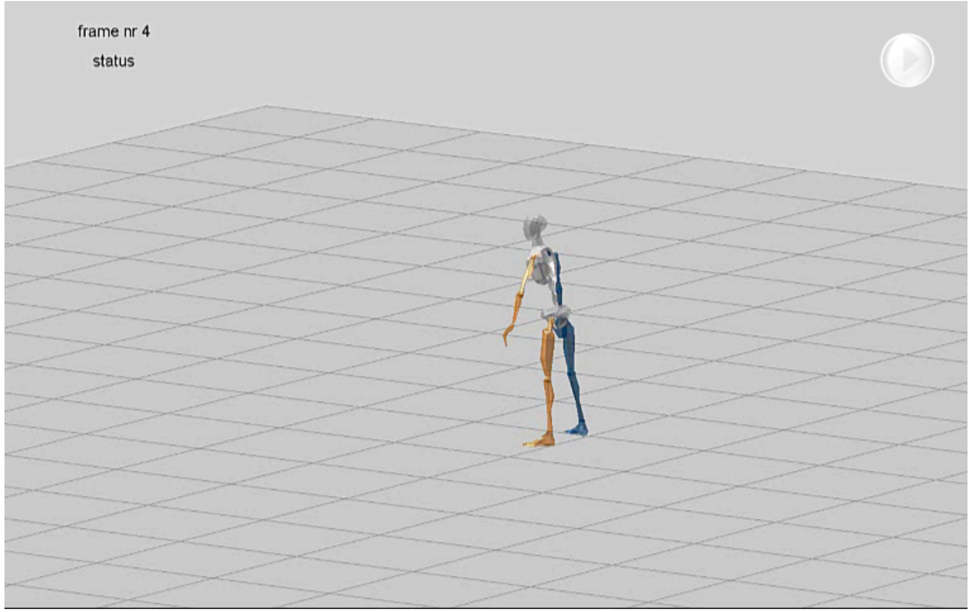
1. Only making use of the inertial information.



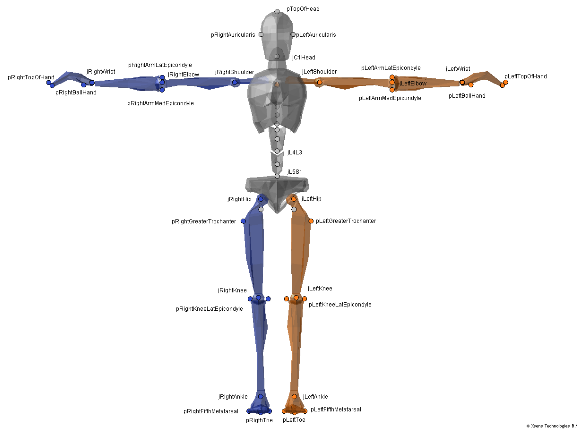
Movie courtesy of Daniel Roetenberg (Xsens)

Introductory example (III/III)

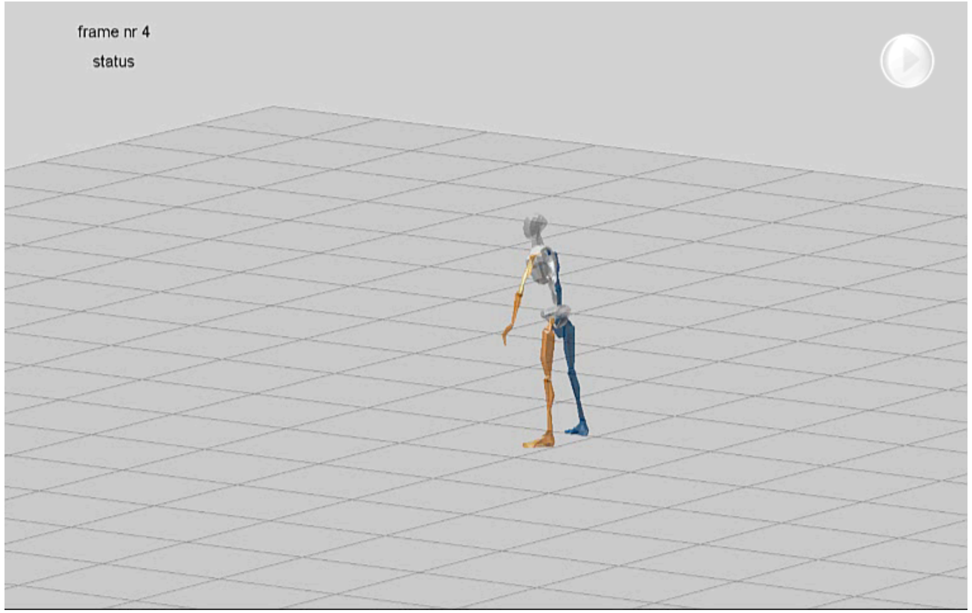
2. Inertial + biomechanical model



Movie courtesy of Daniel Roetenberg (Xsens)



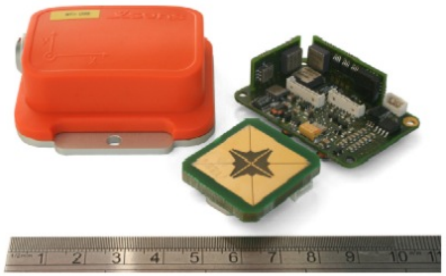
3. Inertial + biomechanical model + world model



Movie courtesy of Daniel Roetenberg (Xsens)

Adding another sensor to the introductory example

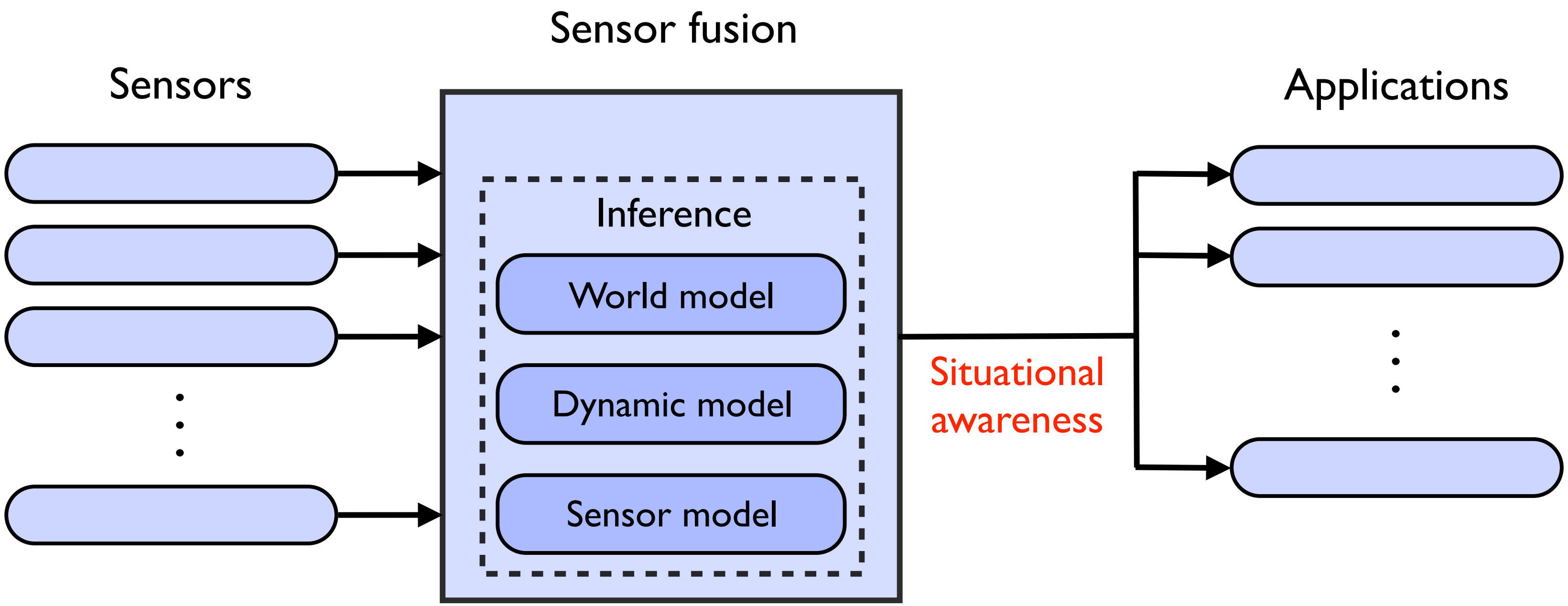
In this experiment we also make use of ultra-wideband (UWB). This allows for indoor positioning as well.



Sensor fusion - definition

Definition (sensor fusion)

Sensor fusion is the process of using information from **several different** sensors to **infer** what is happening (this typically includes finding states of dynamical systems and various static parameters).



Outlook

These introductory examples leads to several questions, e.g.,

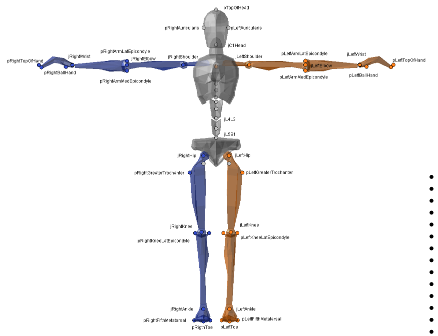
- Can we incorporate more sensors?
- Can we make use of more informative world models?
- How do we solve the inherent inference problem?
- Perhaps most importantly, can this be solved systematically?

There are quite many interesting problems that can be solved systematically, by addressing the following problem areas

1. Probabilistic models of dynamical systems
2. Sensor models
3. World models
4. Formulate and solve an inference problem
5. Surrounding infrastructure

This is what we refer to as sensor fusion!

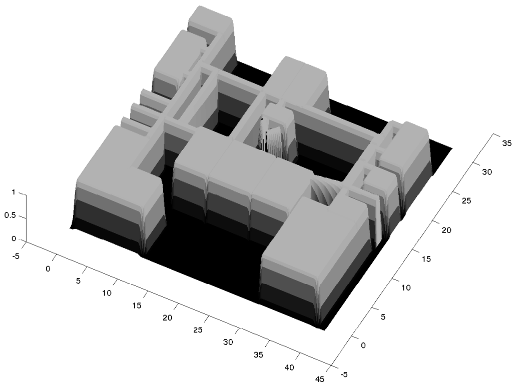
The story I am telling



$$\dot{x} = f(x, u, \theta)$$

1. We are dealing with dynamical systems

This requires a **dynamical model**.



2. The dynamical systems exist in a context.

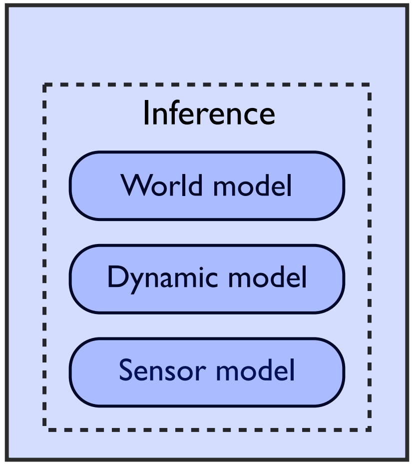
This requires a **world model**.

3. The dynamical systems must be able to perceive their own (and others') motion, as well as the surrounding world.

This requires sensors and **sensor models**.

4. We must be able to transform the measurements from the sensors into knowledge about the dynamical systems and their surrounding world.

This requires **sensor fusion**.



Outline

Sensor fusion

1. Introductory examples
2. Probabilistic models of dynamical systems
3. State inference and the particle filter
4. Rao-Blackwellized particle filter
5. Using world models in solving inference problems

Industrial application examples

1. Fighter aircraft navigation
2. Automotive localization
3. Indoor localization
4. Underwater localization

Concluding experiment

and conclusions

Probabilistic models of dynamical systems

Basic representation: Two discrete-time stochastic processes,

- $\{x_t\}_{t\geq 1}$ representing the state of the system
- $\{y_t\}_{t\geq 1}$ representing the measurements from the sensors

The probabilistic model is described using two (f and g) probability density functions (PDFs):

State → x_{t+1}; Known input → u_t

$$x_{t+1} \mid x_t \sim f_\theta(x_{t+1} \mid x_t, u_t), \qquad \text{Dynamics}$$

$$y_t \mid x_t \sim g_\theta(y_t \mid x_t, u_t). \qquad \text{Measurements}$$

Measurements → y_t; Static parameters → θ

Model = PDF

This type of model is referred to as a **state space model (SSM)** or a **hidden Markov model (HMM)**.

State inference in dynamical systems (I/III)

Aim: Compute a probabilistic representation of our knowledge of the state, based on information that is present in the measurements.

The **filtering PDF** provides a representation of the uncertainty about the state at time t, given all the measurements up to time t,

$$p(x_t \mid y_{1:t})$$

The obvious question is now, how do we compute this object?

Bayes' theorem

$$p(x_t \mid y_{1:t}) = p(x_t \mid y_t, y_{1:t-1}) = \frac{p(y_t \mid x_t, y_{1:t-1})p(x_t \mid y_{1:t-1})}{p(y_t \mid y_{1:t-1})}$$

$$= \frac{g(y_t \mid x_t)p(x_t \mid y_{1:t-1})}{p(y_t \mid y_{1:t-1})}$$

Markov property

State inference in dynamical systems (II/III)

Apparently we need an expression also for the prediction PDF

$$p(x_t \mid y_{1:t-1})$$

Let us start by noting that by marginalization we have

$$p(x_t \mid y_{1:t-1}) = \int p(x_t, x_{t-1} \mid y_{1:t-1}) \mathrm{d}x_{t-1}$$

$$\begin{aligned} p(x_t, x_{t-1} \mid y_{1:t-1}) &= p(x_t \mid x_{t-1}, y_{1:t-1}) p(x_{t-1} \mid y_{1:t-1}) \\ &= f(x_t \mid x_{t-1}) p(x_{t-1} \mid y_{1:t-1}) \end{aligned}$$

Markov property

Hence, the prediction PDF is given by

$$p(x_t \mid y_{1:t-1}) = \int f(x_t \mid x_{t-1}) p(x_{t-1} \mid y_{1:t-1}) \mathrm{d}x_{t-1}$$

State inference in dynamical systems (IIII/III)

We have now showed that for the nonlinear SSM

$$\begin{aligned} x_{t+1} \mid x_t &\sim f(x_t \mid x_{t-1}), \\ y_t \mid x_t &\sim g(y_t \mid x_t), \end{aligned}$$

the uncertain information that we have about the state is captured by the filtering PDF, which we compute sequentially using a **measurement update**

$$p(x_t \mid y_{1:t}) = \frac{\overbrace{g(y_t \mid x_t)}^{\text{measurement model}} \overbrace{p(x_t \mid y_{1:t-1})}^{\text{prediction pdf}}}{p(y_t \mid y_{1:t-1})},$$

and a **time update**

$$p(x_t \mid y_{1:t-1}) = \int \underbrace{f(x_t \mid x_{t-1})}_{\text{dynamic model}} \underbrace{p(x_{t-1} \mid y_{1:t-1})}_{\text{filtering pdf}} \mathrm{d}x_{t-1},$$

State inference - simple special case

Consider the following special case (Linear Gaussian State Space (LGSS) model)

$$x_{t+1} = Ax_t + Bu_t + v_t, \qquad v_t \sim \mathcal{N}(0, Q),$$
$$y_t = Cx_t + Du_t + e_t, \qquad e_t \sim \mathcal{N}(0, R).$$

or, equivalently,

$$x_{t+1} \mid x_t \sim f(x_{t+1} \mid x_t) = \mathcal{N}(x_{t+1} \mid Ax_t + Bu_t, Q),$$
$$y_t \mid x_t \sim g(y_t \mid x_t) = \mathcal{N}(y_t \mid Cx_t + Du_t, R).$$

It is now straightforward to show that the solution to the time update and measurement update equations is given by the Kalman filter, resulting in

$$p(x_t \mid y_{1:t}) = \mathcal{N}\left(x_t \mid \widehat{x}_{t|t}, P_{t|t}\right),$$
$$p(x_{t+1} \mid y_{1:t}) = \mathcal{N}\left(x_{t+1} \mid \widehat{x}_{t+1|t}, P_{t+1|t}\right).$$

State inference - interesting case

Obvious question: what do we do in an interesting case, for example when we have a nonlinear model including a world model in the form of a map?

- Need a general representation of the filtering PDF
- Try to solve the equations

$$p(x_t \mid y_{1:t}) = \frac{g(y_t \mid x_t)p(x_t \mid y_{1:t-1})}{p(y_t \mid y_{1:t-1})},$$

$$p(x_{t+1} \mid y_{1:t}) = \int f(x_{t+1} \mid x_t)p(x_t \mid y_{1:t})\mathrm{d}x_t,$$

as accurately as possible.

State inference - the particle filter (I/II)

The particle filter provides an approximation of the filter PDF

$$p(x_t \mid y_{1:t})$$

when the state evolves according to an SSM

$$\begin{aligned} x_{t+1} \mid x_t &\sim f(x_{t+1} \mid x_t, u_t), \\ y_t \mid x_t &\sim h(y_t \mid x_t, u_t), \\ x_1 &\sim \mu(x_1). \end{aligned}$$

The particle filter maintains an empirical distribution made up N samples (particles) and corresponding weights

$$\widehat{p}(x_t \mid y_{1:t}) = \sum_{i=1}^{N} w_t^i \delta_{x_t^i}(x_t)$$

"Think of each particle as one simulation of the system state. Only keep the good ones."

This approximation converge to the true filter PDF,

Xiao-Li Hu, Thomas B. Schön and Lennart Ljung. **A Basic Convergence Result for Particle Filtering**. *IEEE Transactions on Signal Processing*, 56(4):1337-1348, April 2008.

State inference - the particle filter (II/II)

The weights and the particles in

$$\widehat{p}(x_t \mid y_{1:t}) = \sum_{i=1}^{N} w_t^i \delta_{x_t^i}(x_t)$$

are updated as new measurements becomes available. This approximation can for example be used to compute an estimate of the mean value,

$$\widehat{x}_{t|t} = \int x_t p(x_t \mid y_{1:t}) \mathrm{d}x_t \approx \int x_t \sum_{i=1}^{N} w_t^i \delta_{x_t^i}(x_t) \mathrm{d}x_t = \sum_{i=1}^{N} w_t^i x_t^i$$

The theory underlying the particle filter has been developed over the past two decades and the theory and its applications are still being developed at a very high speed. For a timely tutorial, see

A. Doucet and A. M. Johansen. **A tutorial on particle filtering and smoothing: fifteen years later**. In *Oxford Handbook of Nonlinear Filtering*, 2011, D. Crisan and B. Rozovsky (eds.). Oxford University Press.

or my new PhD course on computational inference in dynamical systems

users.isy.liu.se/rt/schon/course_CIDS.html

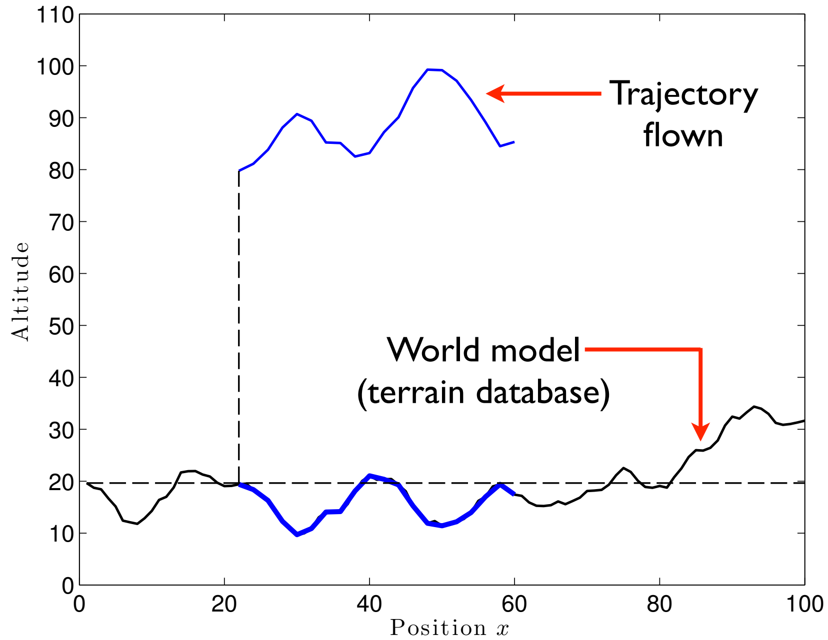
Using world models in solving state inference problems

Consider a 1D localization example.

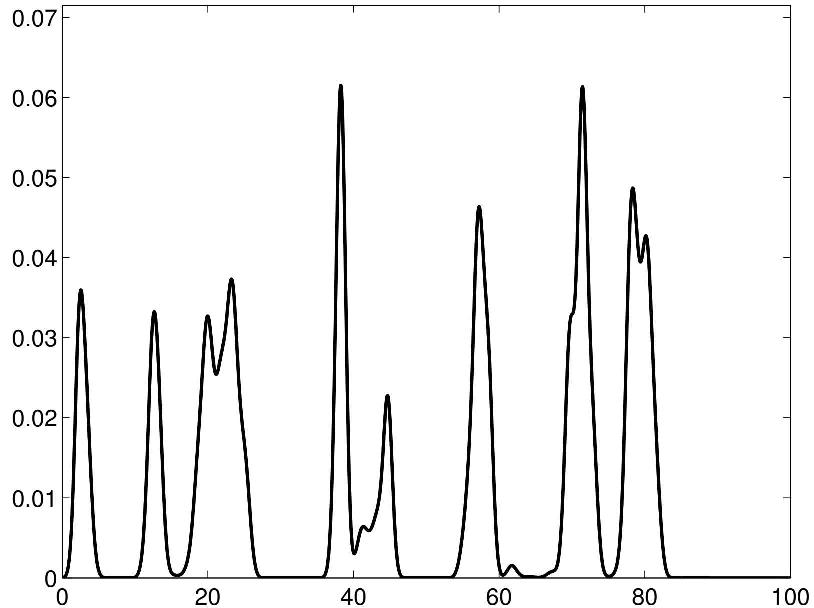
position → x_{t+1}; velocity (measured input) → u_t

$$x_{t+1} = x_t + u_t + v_t,$$
$$y_t = h(x_t) + e_t.$$

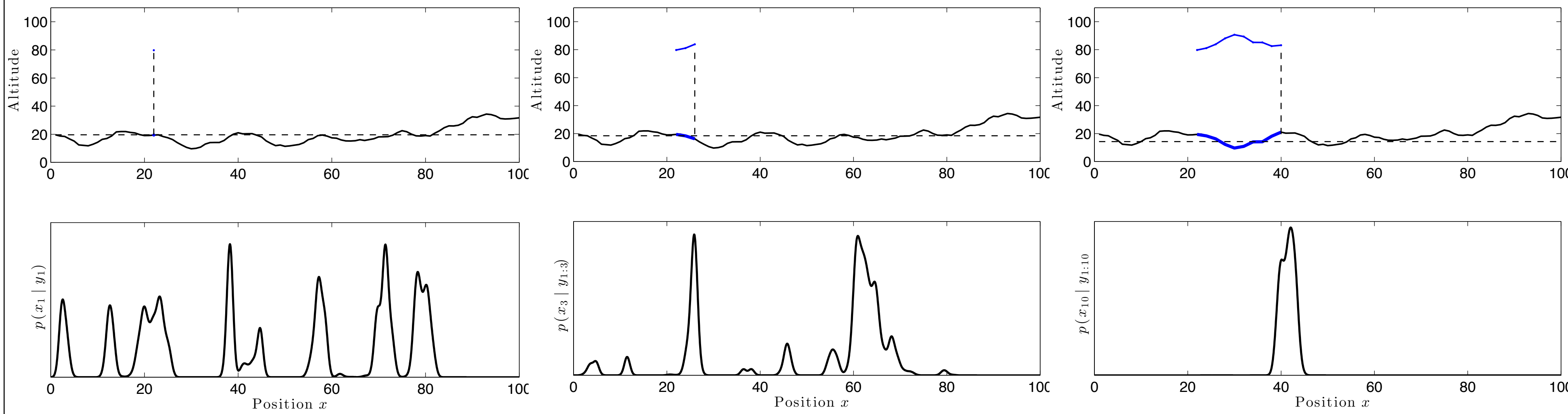
measurement (altitude) → y_t; world model (terrain database) → h



Filter PDF after 1 measurement $p(x_1 \mid y_1)$ →



Using world models in solving state inference problems



Filter PDF after 1 measurement

$p(x_1 \mid y_1)$

Filter PDF after 3 measurements

$p(x_3 \mid y_{1:3})$

Filter PDF after 10 measurements

$p(x_{10} \mid y_{1:10})$

Using world models in solving state inference problems

The simple 1D localization example is an illustration of a problem involving a multimodal filter PDF

- Straightforward to represent and work with using a PF
- Horrible to work with using e.g. an extended Kalman filter

The example also highlights the **key capabilities** of the PF:

1. To automatically handle an unknown and dynamically changing number of hypotheses.

2. Work with nonlinear/non-Gaussian models

We have implemented a similar localization solution for this aircraft (Gripen).

Industrial partner: Saab

Rao-Blackwellized particle filter (RBPF)

If there is **structure** in a problem, that should be used in constructing algorithms.

The Rao-Blackwellized particle filter (RBPF) exploits a **conditionally linear Gaussian** sub-structure. The conditionally linear Gaussian states are estimated using a Kalman filter (KF) and the nonlinear states are estimated using the PF.

The state can be divided into one "nonlinear" state and one "linear" state,

$$x_t = \begin{pmatrix} s_t \\ z_t \end{pmatrix}$$

Definition (Conditionally linear Gaussian state space (CLGSS) model):

Assume that the state of an SSM can be partitioned according to $x_t = \begin{pmatrix} s_t{}^T & z_t{}^T \end{pmatrix}^T$. The SSM is then a CLGSS model if the conditional process $\{z_t \mid s_{1:t}\}_{t \geq 1}$ is described by a linear Gaussian SSM.

Rao-Blackwellized particle filter (RBPF)

The augmented state vector consists of a "nonlinear" state and a "linear" state,

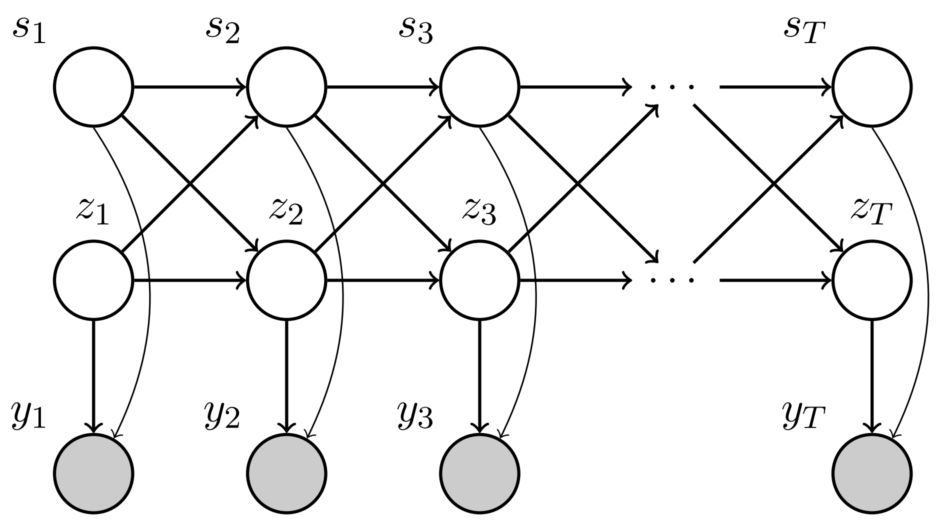
$$x_t = \begin{pmatrix} s_t \\ z_t \end{pmatrix}$$

The CLGSS model we are considering is defined:

Equations

$$\begin{aligned} s_{t+1} &= f_t^s(s_t) + A_t^s(s_t)z_t + v_t^s(s_t), \\ z_{t+1} &= f_t^z(s_t) + A_t^z(s_t)z_t + v_t^z(s_t), \\ y_t &= h_t(s_t) + C_t(s_t)z_t + e_t(s_t), \end{aligned}$$

Graphical model



Rao-Blackwellized particle filter (RBPF)

By exploiting the tractable CLGSS sub-structure, the RBPF results in more accurate estimators (lower variance) than a standard PF.

A direct result of this is that the RBPF can be used for filtering in even more challenging - e.g. high-dimensional - models.

Target this density using the particle filter

$$p(z_t, s_{1:t} \mid y_{1:t}) = p(z_t \mid s_{1:t}, y_{1:t})\, p(s_{1:t} \mid y_{1:t})$$

Compute this density using the Kalman filter (closed form expressions)

Rao-Blackwellized particle filter

The particle filter targets the nonlinear states,

$$\widehat{p}^N(s_{1:t} \mid y_{1:t}) = \sum_{i=1}^{N} w_t^i \delta\left(s_{1:t} - s_{1:t}^i\right)$$

while the conditional KFs - one for each particle - are used for the linear state,

$$p(z_t \mid s_{1:t}, y_{1:t}) = \mathcal{N}\left(z_t \mid \bar{z}_{t|t}(s_{1:t}), P_{t|t}(s_{1:t})\right)$$

The result is a weighted sum of Gaussians

$$p(z_t, s_{1:t} \mid y_{1:t}) = p(z_t \mid s_{1:t}, y_{1:t})\, p(s_{1:t} \mid y_{1:t})$$

$$\approx \sum_{i=1}^{N} w_t^i \mathcal{N}\left(z_t \mid \bar{z}_t^i, P_{t|t}^i\right) \delta\left(s_{1:t} - s_{1:t}^i\right)$$

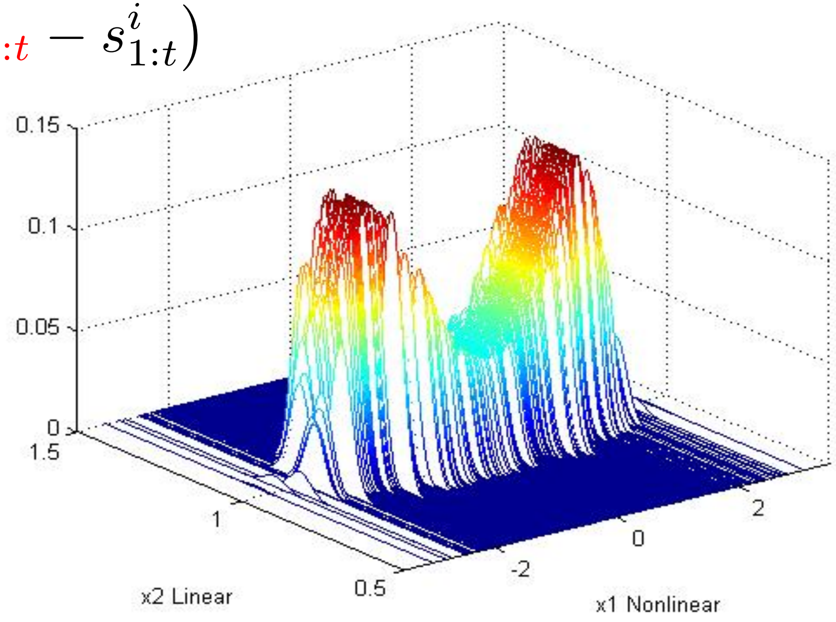
Each particle

has a KF attached to it

Rao-Blackwellized particle filter

$$p(z_t, s_{1:t} \mid y_{1:t}) \approx \sum_{i=1}^{N} w_t^i \mathcal{N}\left(z_t \mid \bar{z}_t^i, P_{t|t}^i\right) \delta\left(s_{1:t} - s_{1:t}^i\right)$$

The RBPF consists of interlinked Kalman filters and a particle filter.



Detailed derivation of the RBPF is available here (with fighter aircraft example):

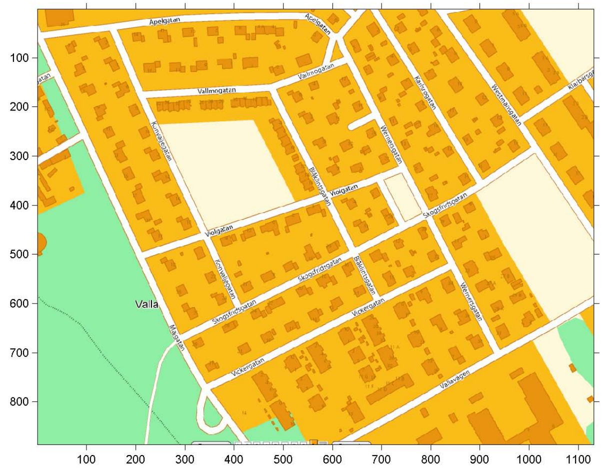
Thomas Schön, Fredrik Gustafsson, and Per-Johan Nordlund. **Marginalized Particle Filters for Mixed Linear/Nonlinear State-Space Models**. *IEEE Transactions on Signal Processing*, 53(7):2279-2289, July 2005.

Software solving a simple example using the RBPF is available here:

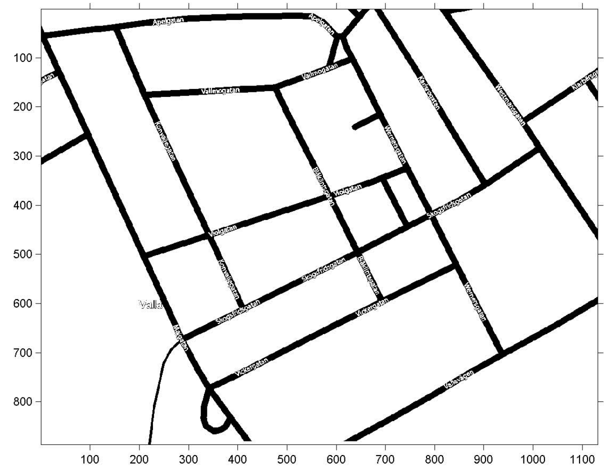
users.isy.liu.se/rt/schon/SW_RBPF.html

Using world models in solving state inference problems

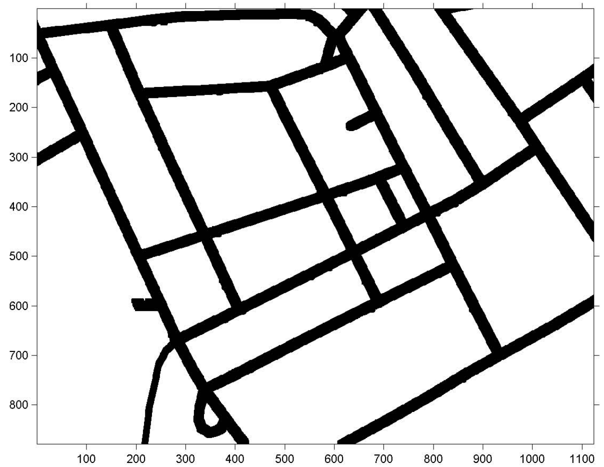
So far, just a simple 1D example, we can of course do this also in 2D, 3D and xD.



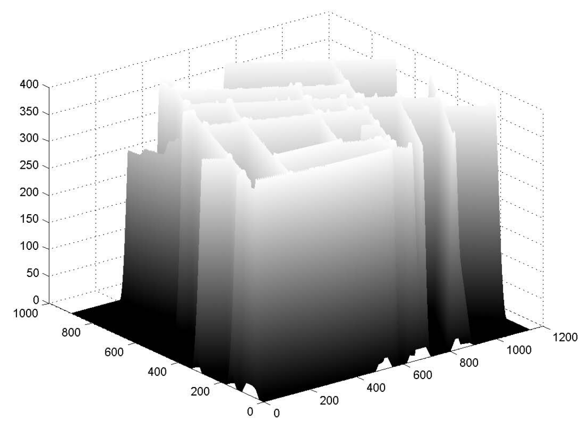
(a) Original map



(b) Binary map



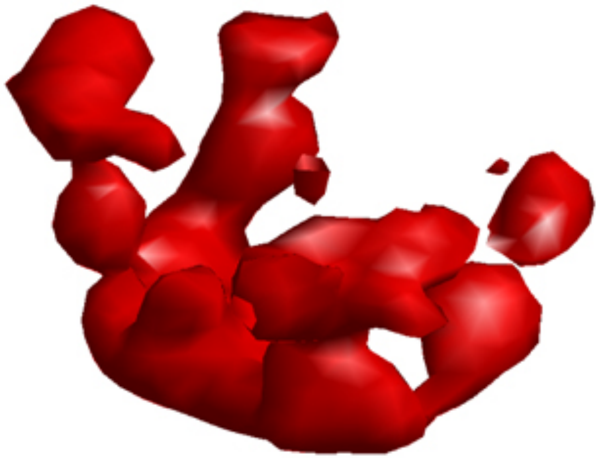
(c) Filtered binary map



(d) On-road likelihood function

xD world models

Idea: Make use of several different world models. One new world model that we are investigating is one that is induced by the magnetic field.



Estimated magnetic content in a table turned upside down.

Very much work in progress, for some initial results,

Niklas Wahlström, Manon Kok, Thomas B. Schön and Fredrik Gustafsson. **Modeling magnetic fields using Gaussian processes**. *Submitted to the 38th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Vancouver, Canada, May 2013.

Manon Kok, Niklas Wahlström, Thomas B. Schön and Fredrik Gustafsson. **MEMS-based inertial navigation based on a magnetic field map**. *Submitted to the 38th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Vancouver, Canada, May 2013

Outline

Sensor fusion

1. Introductory examples
2. Probabilistic models of dynamical systems
3. State inference and the particle filter
4. Rao-Blackwellized particle filter
5. Using world models in solving inference problems

Industrial application examples

1. Fighter aircraft navigation
2. Automotive localization
3. Indoor localization
4. Underwater localization

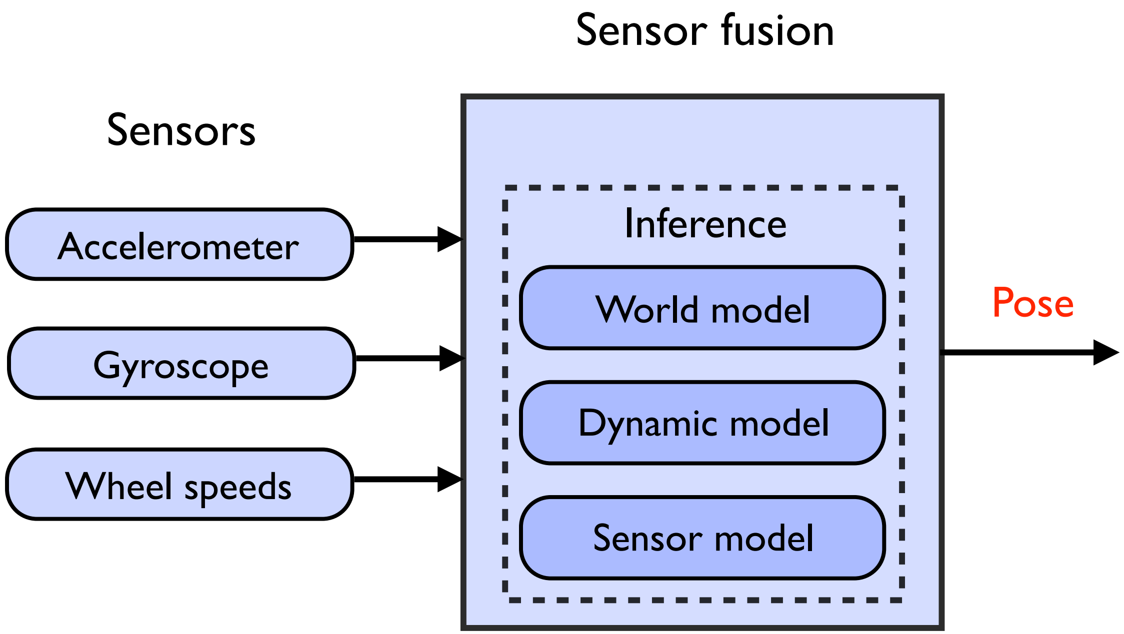
Concluding experiment

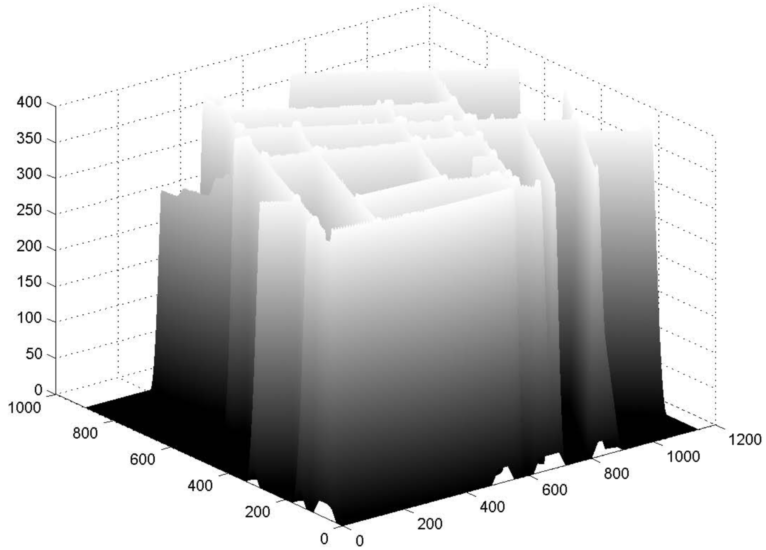
and conclusions

Example 2 - Automotive localization (I/III)

Aim: Compute the position of a car.

Industrial partner: Nira dynamics

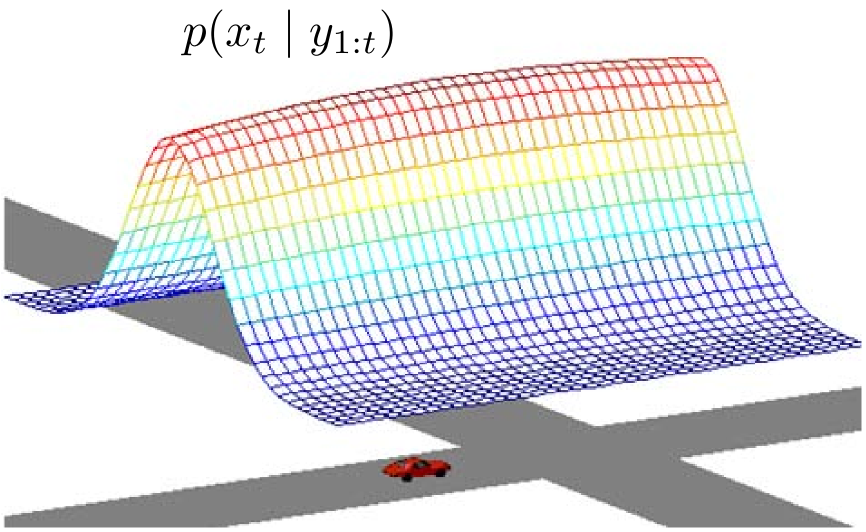




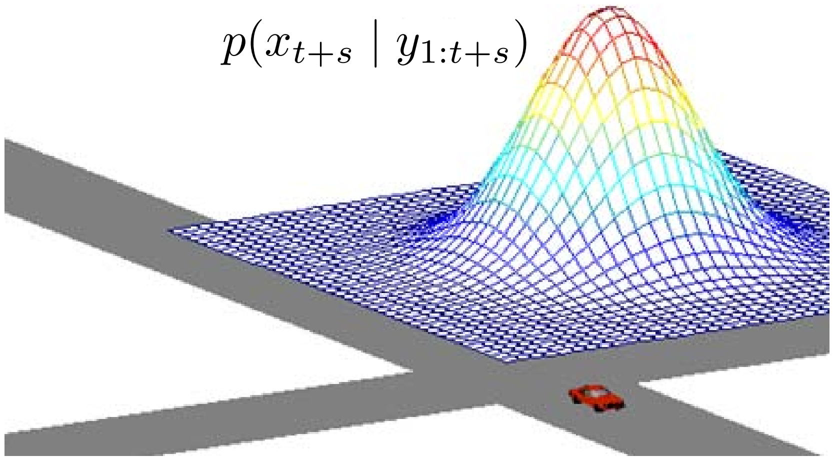
World model

Example 2 - Automotive localization (II/III)

Schematic illustration of the idea.



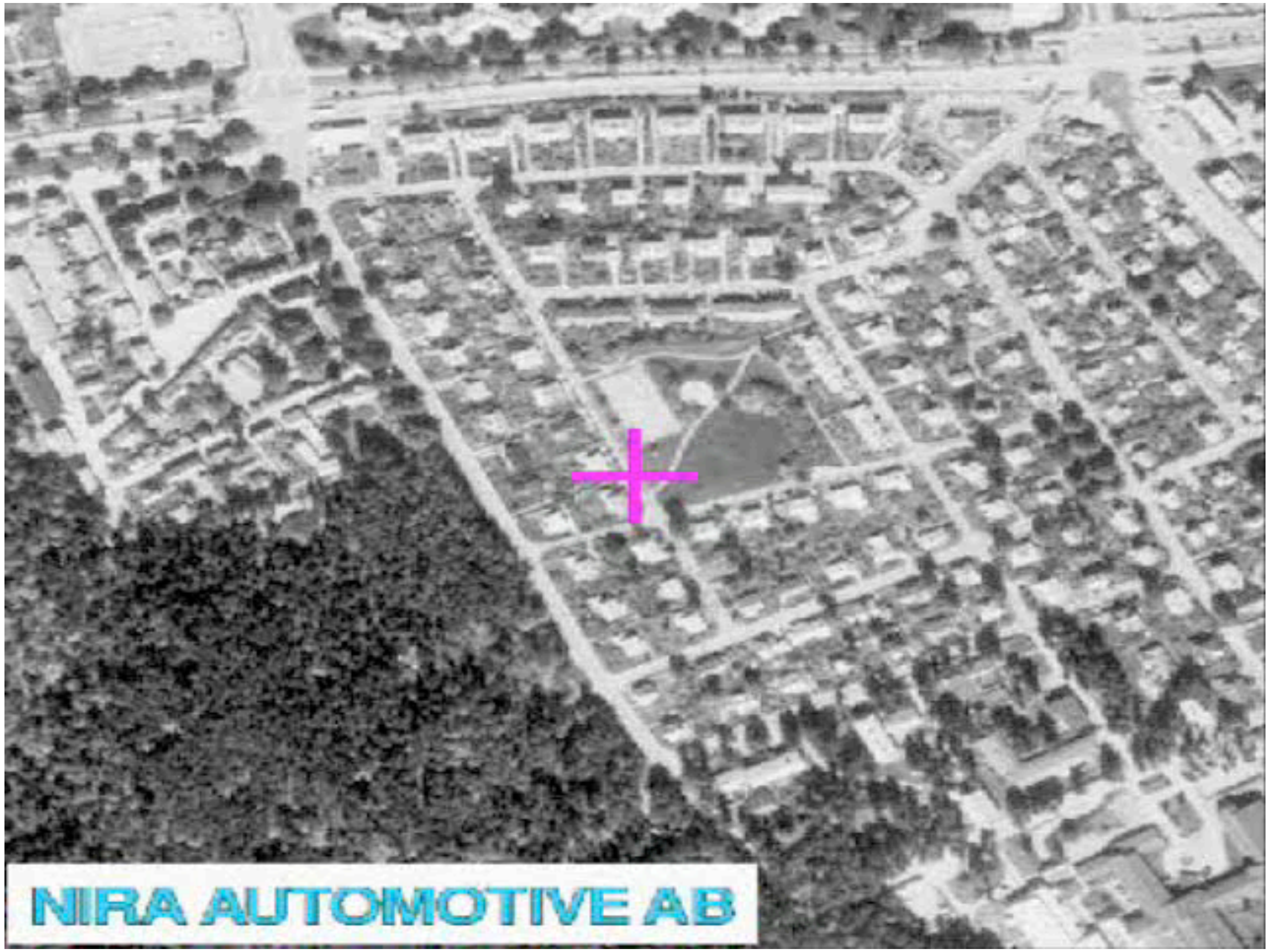
Filter PDF before the right turn



Filter PDF after the right turn

Example 2 - Automotive localization (III/III)

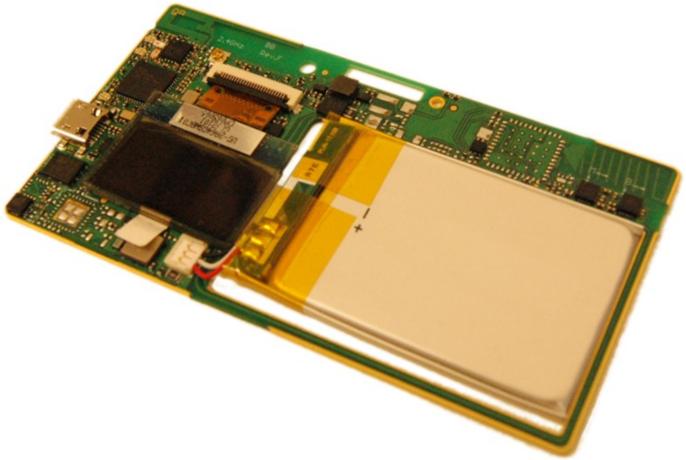
- Purple: True position
- Blue: Particles
- Light blue: estimate



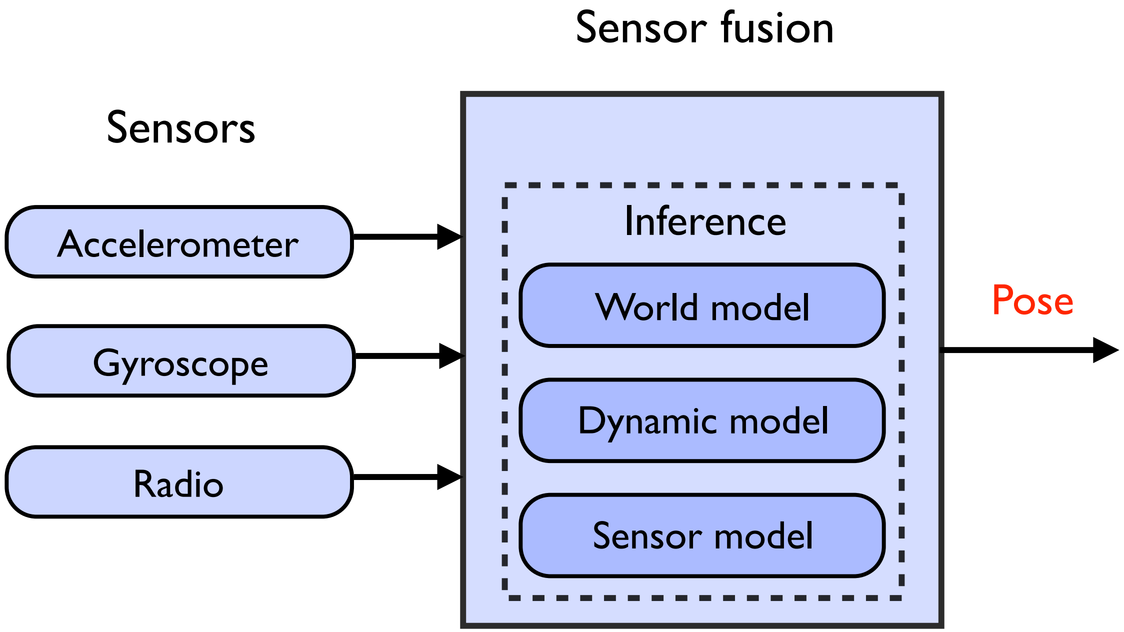
Example 3 - Indoor localization (I/III)

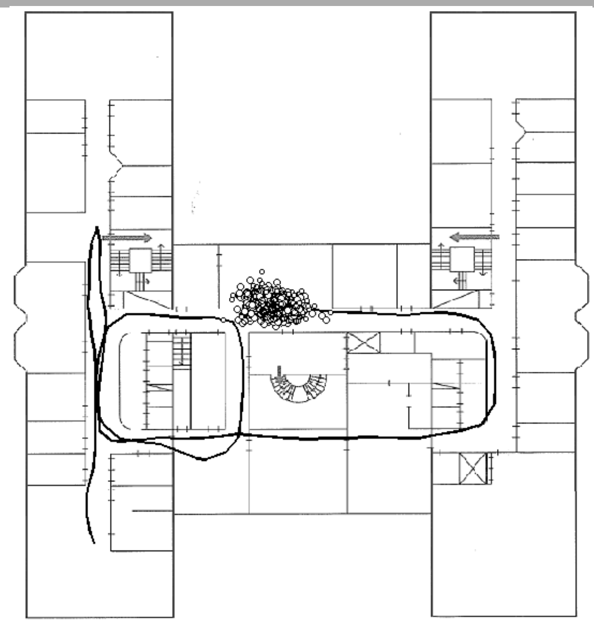
Aim: Compute the position of a person moving around indoors using sensors (inertial, magnetometer and radio) located in an ID badge and a map.

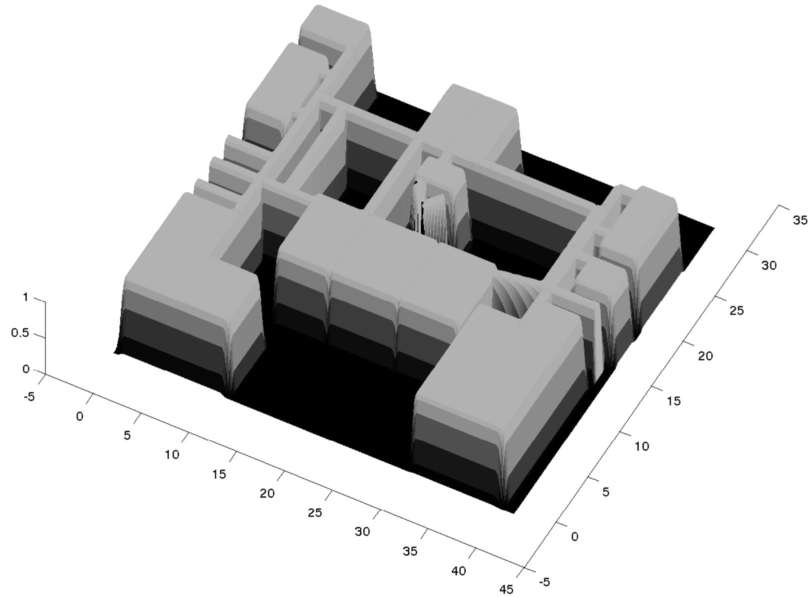
Industrial partner: Xdin



Example 3 - Indoor localization (II/III)

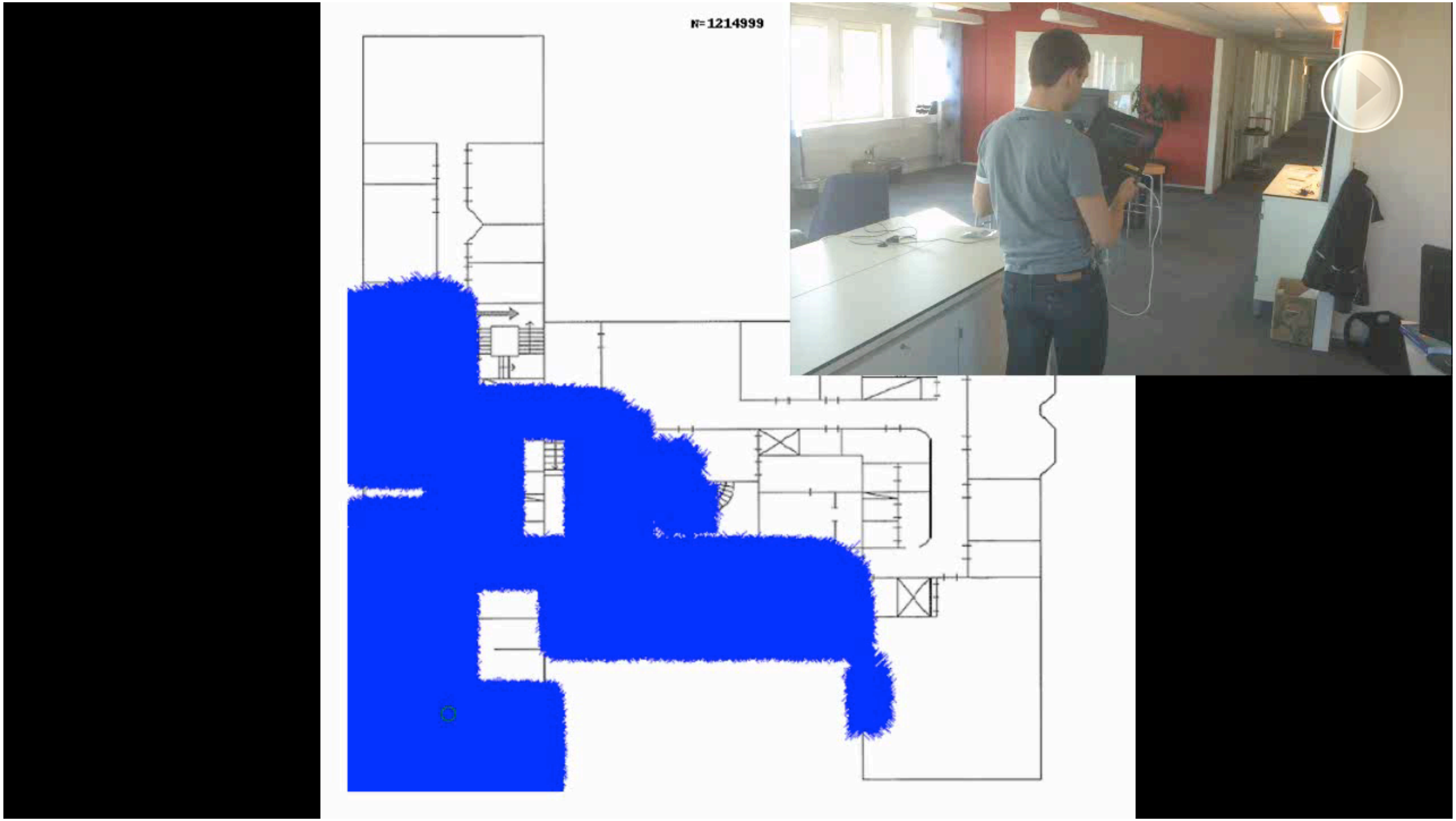






PDF of an office environment, the bright areas are rooms and corridors (i.e., walkable space).

Example 3 - Indoor localization (III/III)

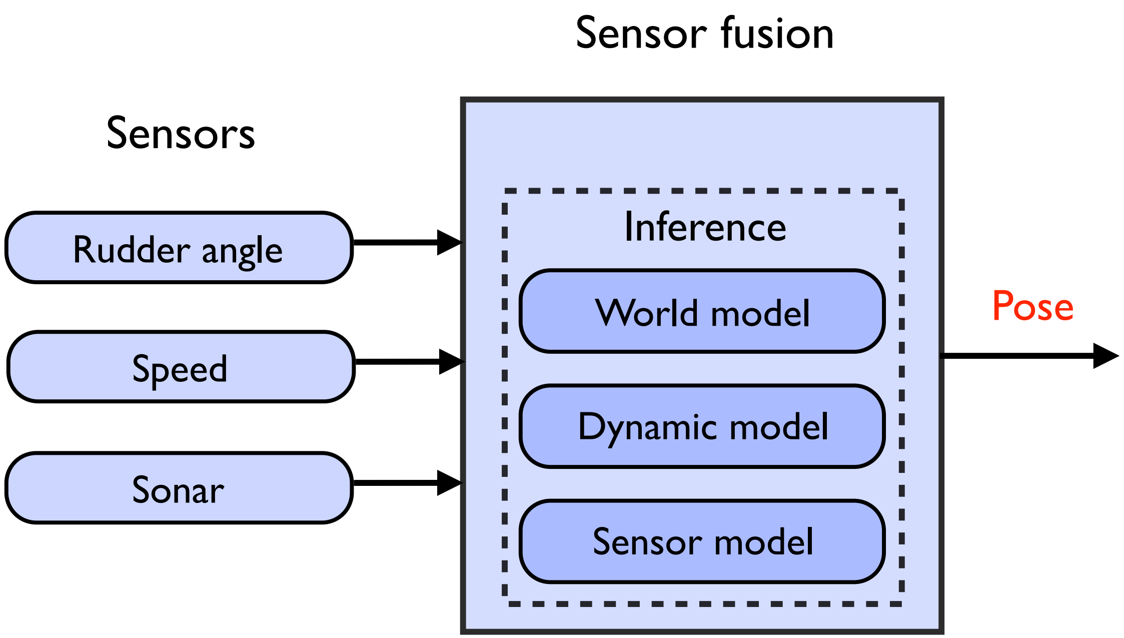


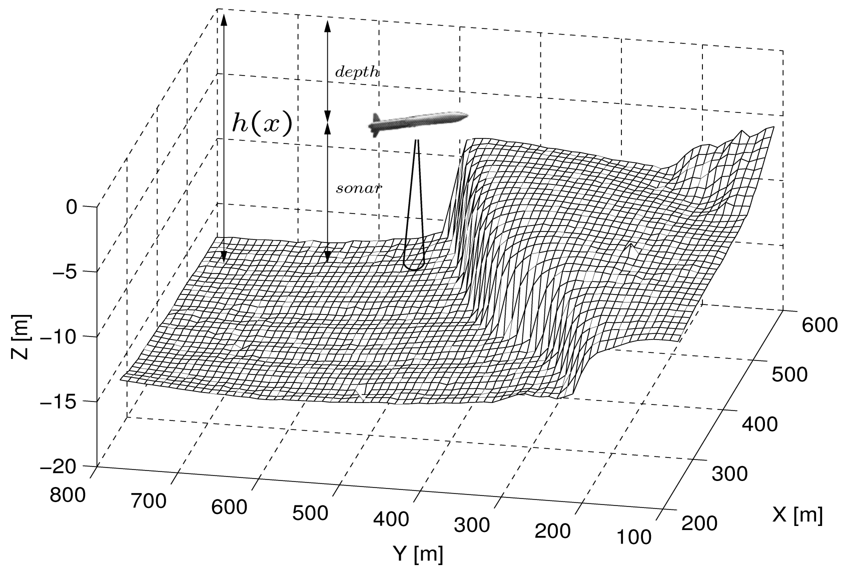
Example 4 - Underwater localization (I/II)

Aim: Find the position and orientation of an autonomous underwater vehicle.

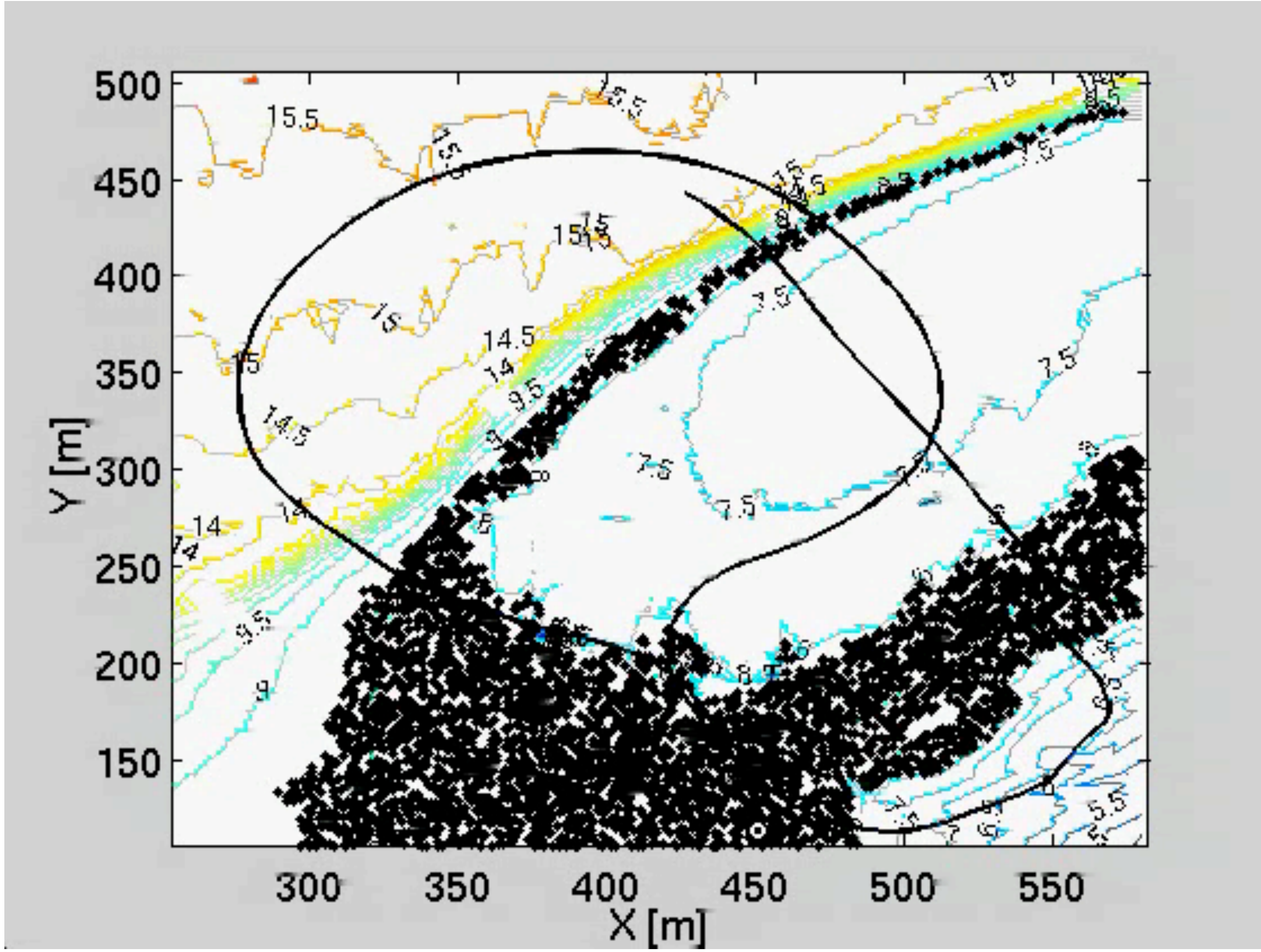
Industrial partner: Saab underwater security.

Work by my colleague Rickard Karlsson.



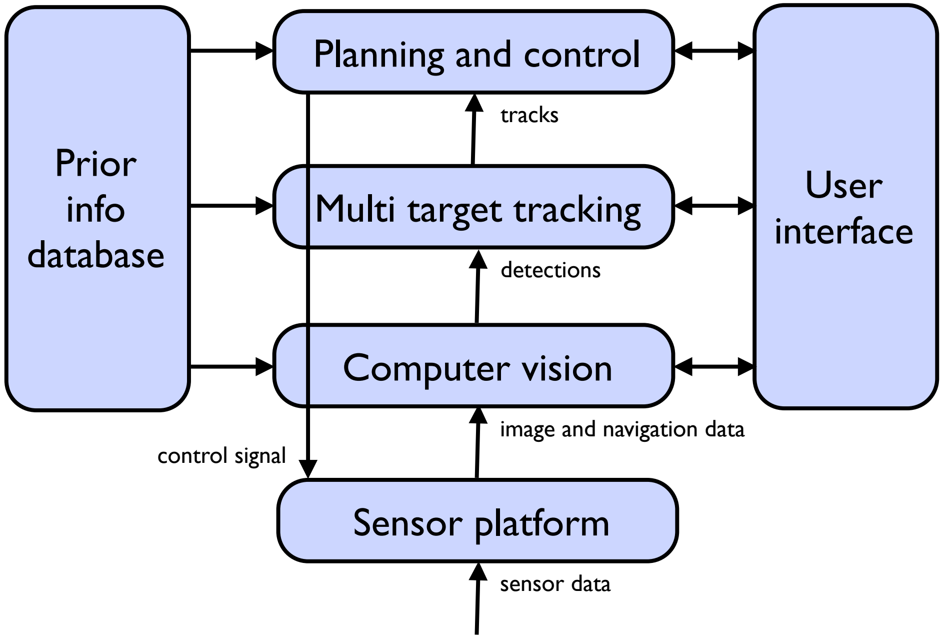


Example 4 - Underwater localization (II/II)



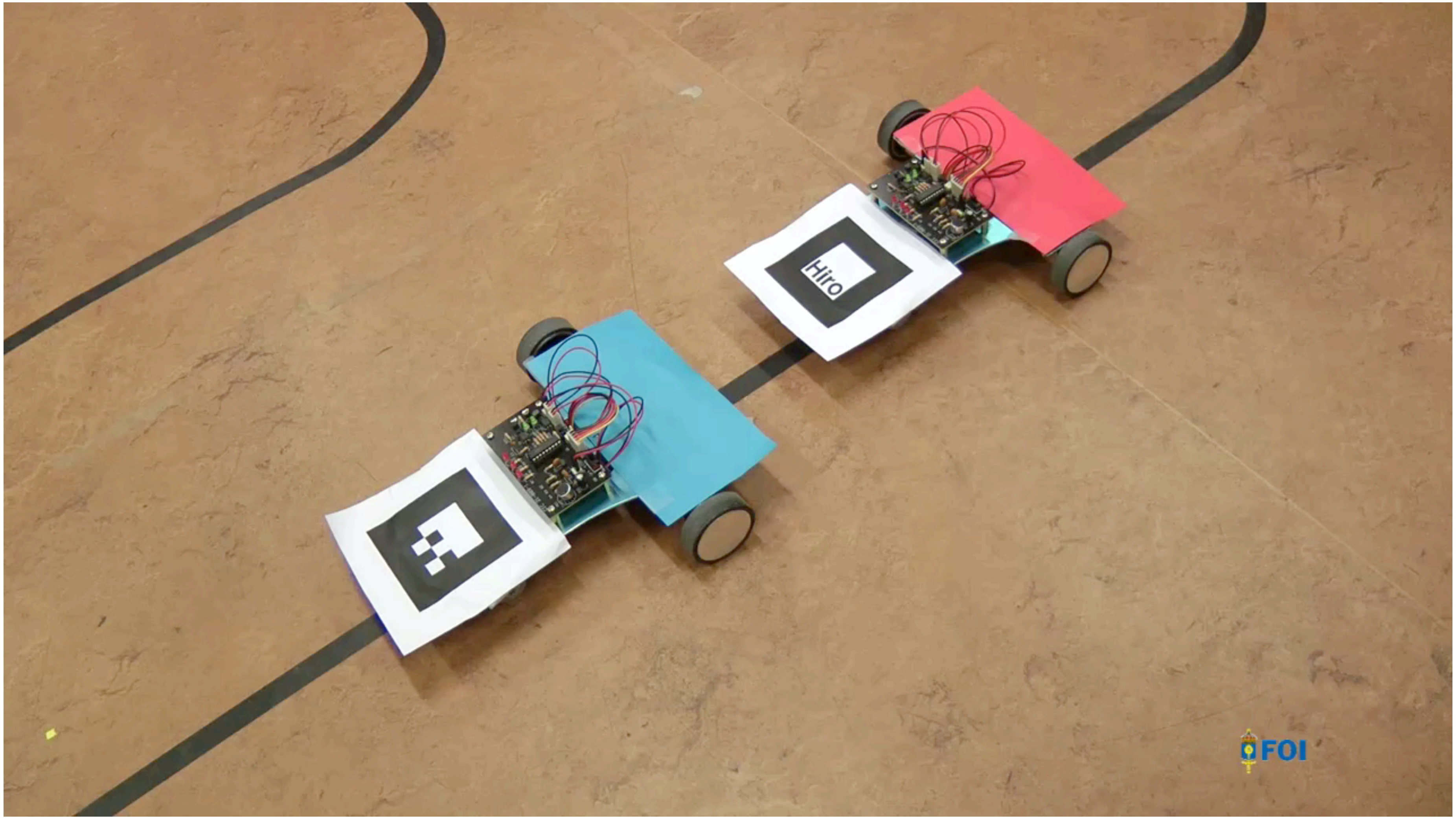
Road target search and tracking - an experiment (I/II)

Aim: Keep track of all discovered on-road targets and simultaneously search for new on-road targets by controlling the pointing direction of a camera gimbal.



Overview of the implemented solution.

Road target search and tracking - an experiment (II/II)



Movie kindly provided by Per Skoglar. For technical details see his PhD thesis,

Skoglar, P. **Tracking and planning for surveillance applications.** Linköping studies in science and technology. Dissertation No. 1432, June 2012.

The story I am telling

Quite a few different applications from different areas, all solved using the **same underlying sensor fusion strategy**

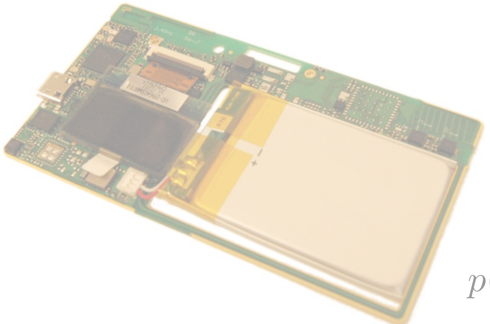
- **Model** the dynamics
- **Model** the sensors
- **Model** the world

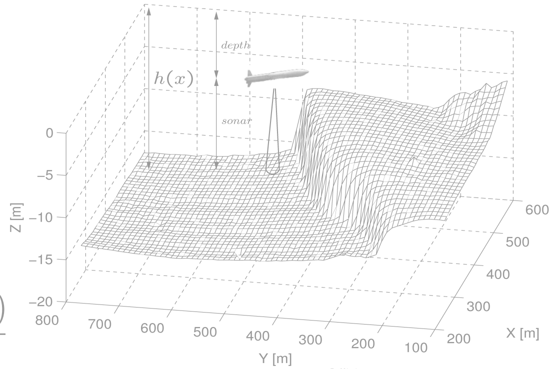
- Solve the resulting **inference** problem

and, do not underestimate the "surrounding infrastructure"!

- There is a lot of **interesting research** that remains to be done!
- The number of available sensors is currently skyrocketing
- The **industrial utility** of this technology is **growing** as we speak!

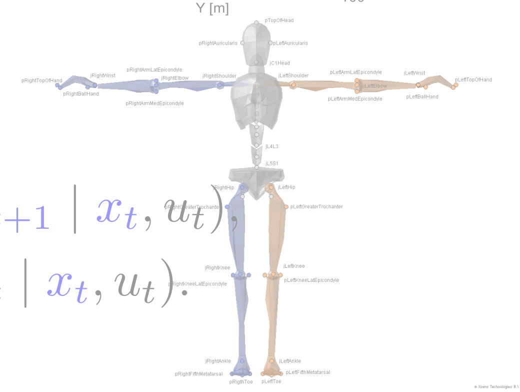
Thank you for your attention!!

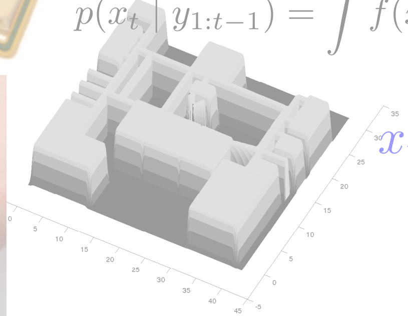




$$p(x_t \mid y_{1:t}) = \frac{h(y_t \mid x_t)p(x_t \mid y_{1:t-1})}{p(y_t \mid y_{1:t-1})}$$

$$p(x_t \mid y_{1:t-1}) = \int f(x_t \mid x_{t-1})p(x_{t-1} \mid y_1$$





$$x_{t+1} \mid x_t \sim f_\theta(x_{t+1} \mid x_t, u_t),$$
$$y_t \mid x_t \sim h_\theta(y_t \mid x_t, u_t).$$

Joint work with (in alphabetical order): **Fredrik Gustafsson** (LiU), **Jeroen Hol** (Xsens), **Rickard Karlsson** (Nira Dynamics), **Johan Kihlberg** (Semcon), **Manon Kok** (LiU), **Henk Luinge** (Xsens), **Per-Johan Nordlund** (Saab), **Daniel Roetenberg** (Xsens), **Per Skoglar** (SenionLab), **Simon Tegelid** (Xdin), **Niklas Wahlström** (LiU).