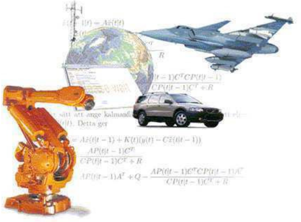
Sensor fusion and parameter inference in nonlinear dynamical systems

– Strategies and concrete examples



Thomas Schön

Division of Automatic Control
Linköping University
Sweden

Joint work with (alphabetical order): **Fredrik Gustafsson** (Linköping University), **Jeroen Hol** (Xsens technologies), **Michael I. Jordan** (UC Berkeley), **Johan Kihlberg** (Xdin), **Fredrik Lindsten** (Linköping University), **Lennart Ljung** (Linköping University), **Brett Ninness** (University of Newcastle, Australia), **Per-Johan Nordlund** (Saab), **Simon Tegelid** (Xdin) and **Adrian Wills** (MRA, Newcastle, Australia).

Outline

Parameter inference

1. The nonlinear Maximum Likelihood (ML) problem
   - Problem formulation
   - Solution using expectation maximization and a particle smoother
2. The nonlinear Bayesian problem
   - Problem formulation
   - Sketch of solution using MCMC and SMC

Sensor fusion

1. Problem formulation
2. Three industrial application examples

A state space model (SSM) consists of a Markov process $\{x_t\}_{t\geq 1}$ and a measurement process $\{y_t\}_{t\geq 1}$, related according to

$$\begin{aligned} x_{t+1} \mid x_t &\sim f_{\theta,t}(x_{t+1} \mid x_t, u_t), \\ y_t \mid x_t &\sim h_{\theta,t}(y_t \mid x_t, u_t), \\ x_1 &\sim \mu_\theta(x_1), \quad (\theta \sim p(\theta)). \end{aligned}$$

Identification problem: Find θ based on $\{u_{1:T}, y_{1:T}\}$.

ML amounts to solving,

$$\widehat{\theta}^{\text{ML}} = \arg\max_{\theta} \; \log p_\theta(y_{1:T})$$

where the log-likelihood function is given by

$$\log p_\theta(y_{1:T}) = \sum_{t=1}^{T} \log p_\theta(y_t \mid y_{1:t-1})$$

Problem formulation – ML (II/II)

There are at least two challenges with the ML formulation:

1. The one-step prediction PDF $p_\theta(y_t \mid y_{1:t-1})$ has to be computed.
2. In solving the optimization problem

$$\widehat{\theta}^{\text{ML}} = \arg\max_{\theta} \log p_\theta(y_{1:T})$$

the derivatives $\frac{\partial}{\partial \theta} p_\theta(y_t \mid y_{1:t-1})$ are useful.

The **Expectation Maximisation (EM)** algorithm together with a **Particle Smoother (PS)** provides a systematic way of dealing with both of these challenges.

The **Expectation Maximization (EM)** algorithm computes ML estimates of unknown parameters in probabilistic models involving latent variables.

Strategy: Use *structure* inherent in the probabilistic model to separate the original ML problem into *two closely linked subproblems*, each of which is hopefully in some sense more tractable than the original problem.

EM focus on the joint log-likelihood function of the observed variables $y_{1:T}$ and the latent variables $Z \triangleq \{x_1, \dots, x_T\}$,

$$\ell_\theta(x_{1:T}, y_{1:T}) = \log p_\theta(x_{1:T}, y_{1:T}).$$

Algorithm 1 Expectation Maximization (EM)

1. **Initialise:** Set $i = 1$ and choose an initial θ^1.
2. **While** not converged **do:**

   (a) **Expectation (E) step:** Compute

$$\begin{aligned}\mathcal{Q}(\theta, \theta^i) &= \mathrm{E}_{\theta^i}\left[\log p_\theta(x_{1:T}, y_{1:T}) \mid y_{1:T}\right] \\ &= \int \log p_\theta(x_{1:T}, y_{1:T}) p_{\theta^i}(x_{1:T} \mid y_{1:T}) \mathrm{d}x_{1:T}\end{aligned}$$

   (b) **Maximization (M) step:** Compute

$$\theta^{i+1} = \underset{\theta \in \Theta}{\arg\max}\ \mathcal{Q}(\theta, \theta^i)$$

   (c) $i \leftarrow i + 1$

In computing the $\mathcal{Q}$-function

$$\begin{aligned}\mathcal{Q}(\theta, \theta^i) &= \mathrm{E}_{\theta^i}\left[\log p_\theta(x_{1:T}, y_{1:T}) \mid y_{1:T}\right] \\ &= \int \log p_\theta(x_{1:T}, y_{1:T}) p_{\theta^i}(x_{1:T} \mid y_{1:T}) \mathrm{d}x_{1:T},\end{aligned}$$

we start by noting that

$$\begin{aligned}\log p_\theta(x_{1:T}, y_{1:T}) &= \log p_\theta(y_{1:T} \mid x_{1:T}) + \log p_\theta(x_{1:T}) \\ &= \log p_\theta(x_1) + \sum_{t=1}^{T-1} \log p_\theta(x_{t+1} \mid x_t) + \sum_{t=1}^{T} \log p_\theta(y_t \mid x_t)\end{aligned}$$

This results in the following expression for the $\mathcal{Q}$-function

$$\mathcal{Q}(\theta, \theta^i) = I_1 + I_2 + I_3,$$

where

$$I_1 = \int \log p_\theta(x_1) p_{\theta^i}(x_1 \mid y_{1:N}) dx_1,$$

$$I_2 = \sum_{t=1}^{T-1} \int \int \log p_\theta(x_{t+1} \mid x_t) p_{\theta^i}(x_{t+1}, x_t \mid y_{1:N}) dx_t dx_{t+1},$$

$$I_3 = \sum_{t=1}^{T} \int \log p_\theta(y_t \mid x_t) p_{\theta^i}(x_t \mid y_{1:N}) dx_t.$$

Nonlinear state smoothing problem, which we approximately solve using sequential Monte Carlo (here, **particle smoothers**).

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,

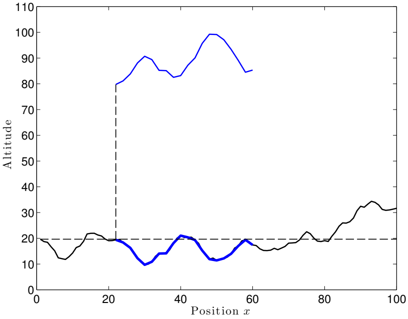
$$\begin{aligned} x_{t+1} \mid x_t &\sim f_{\theta,t}(x_{t+1} \mid x_t, u_t), \\ y_t \mid x_t &\sim h_{\theta,t}(y_t \mid x_t, u_t), \\ x_1 &\sim \mu_\theta(x_1). \end{aligned}$$

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

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

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

Consider a toy 1D localization problem.



Dynamic model:

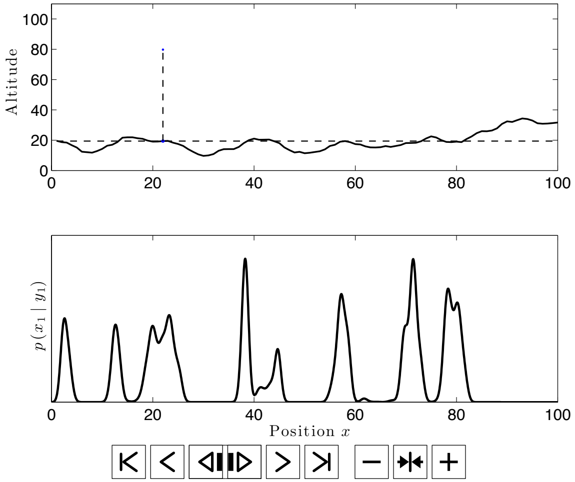
$$x_{t+1} = x_t + u_t + v_t,$$

where x_t denotes position, u_t denotes velocity (known), $v_t \sim \mathcal{N}(0, 5)$ denotes an unknown disturbance.

Measurements:

$$y_t = h(x_t) + e_t.$$

where $h(\cdot)$ denotes the world model (here the terrain height) and $e_t \sim \mathcal{N}(0, 1)$ denotes an unknown disturbance.



Highlights two **key capabilities** of the PF:

1. Automatically handles an unknown and dynamically changing number of hypotheses.
2. Work with nonlinear/non-Gaussian models.

Inserting the PS approximations into the integrals yields the approximation we are looking for,

$$\widehat{I}_1 = \int \log p_\theta(x_1) \sum_{i=1}^{N} w_{1|T}^i \delta_{x_1^i}(x_1) \mathrm{d}x_1$$

$$= \sum_{i=1}^{N} w_{1|T}^i \log p_\theta(x_1^i),$$

$$\widehat{I}_3 = \sum_{t=1}^{T} \int \log p_\theta(y_t \mid x_t) \sum_{i=1}^{N} w_{t|T}^i \delta_{x_t^i}(x_t) \mathrm{d}x_t$$

$$= \sum_{t=1}^{T} \sum_{i=1}^{N} w_{t|T}^i \log p_\theta(y_t \mid x_t^i),$$

and similarly for I_2.

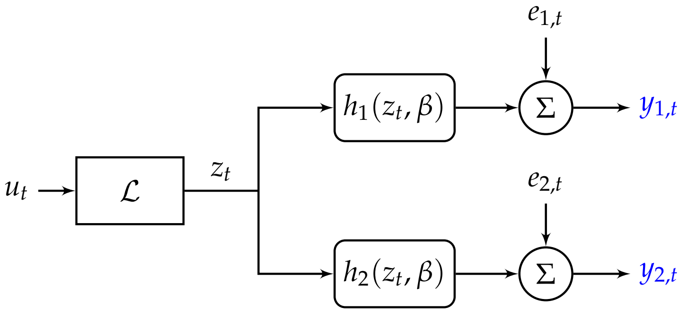
Final ML identification algorithm

Algorithm 2 EM for identifying nonlinear systems

1. **Initialise:** Set $i = 1$ and choose an initial θ^1.
2. **While** not converged **do:**
   (a) **Expectation (E) step:** Run a FFBS PS and compute
   $$\widehat{\mathcal{Q}}(\theta, \theta^i) = \widehat{I}_1(\theta, \theta^i) + \widehat{I}_2(\theta, \theta^i) + \widehat{I}_3(\theta, \theta^i)$$
   (b) **Maximization (M) step:** Compute $\theta^{i+1} = \arg\max_{\theta \in \Theta} \widehat{\mathcal{Q}}(\theta, \theta^i)$ using an off-the-shelf numerical optimization algorithm.
   (c) $i \leftarrow i + 1$

Thomas B. Schön, Adrian Wills and Brett Ninness. **System Identification of Nonlinear State-Space Models**. *Automatica*, 47(1):39-49, January 2011.

Example – blind Wiener identification (I/III)

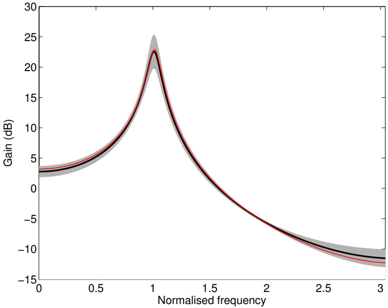


$$x_{t+1} = \begin{pmatrix} A & B \end{pmatrix} \begin{pmatrix} x_t \\ u_t \end{pmatrix}, \qquad u_t \sim \mathcal{N}(0, Q),$$
$$z_t = Cx_t, \qquad y_t = h(z_t, \beta) + e_t, \qquad e_t \sim \mathcal{N}(0, R).$$

Identification problem: Find A, B, C, β, Q, and R based on $\{y_{1,1:T}, y_{2,1:T}\}$ using EM.

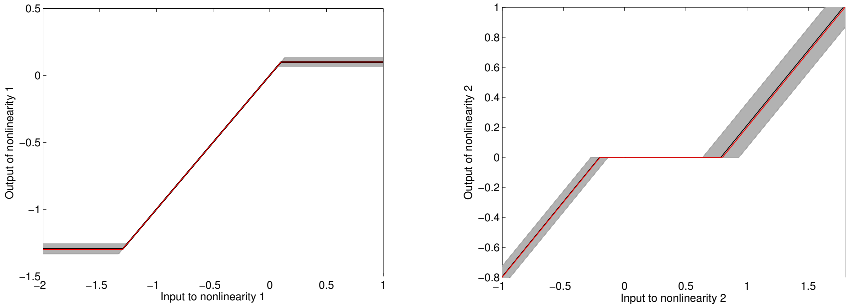
Example – blind Wiener identification (II/III)

- Second order LGSS model with complex poles.
- Employ the EM-PS with $N = 100$ particles.
- Results obtained using $T = 1000$ samples.
- The plots are based on 100 realizations of data.
- Nonlinearities (dead-zone and saturation) shown on next slide.



Bode plot of estimated mean (black), true system (red) and the result for all 100 realisations (gray).

Example – blind Wiener identification (III/III)



Estimated mean (black), true static nonlinearity (red) and the result for all 100 realisations (gray).

Adrian Wills, Thomas B. Schön, Lennart Ljung and Brett Ninness. **Identification of Hammerstein-Wiener Models**. *Automatica*, 49(1): 70-81, January 2013.

Outline

Parameter inference

1. The nonlinear Maximum Likelihood (ML) problem
   - Problem formulation
   - Solution using expectation maximization and a particle smoother
2. **The nonlinear Bayesian problem**
   - Problem formulation
   - Sketch of solution using MCMC and SMC

Sensor fusion

1. Problem formulation
2. Three industrial application examples

Bayesian model: θ is a random variable with a prior density $p(\theta)$.

The **goal** in Bayesian modeling is to compute the posterior $p(\underbrace{\theta, x_{1:T}}_{\triangleq \eta} \mid y_{1:T}) = p(\eta \mid y_{1:T})$ (or one of its marginals).

Bayesian modeling/identification amounts to:

1. Find an expression for the likelihood $p(y_{1:T} \mid \eta)$.
2. Assign priors $p(\eta)$ to all unknown stochastic variables η present in the model.
3. Determine the posterior distribution $p(\eta \mid y_{1:T})$.

The **key challenge** is that there is no closed form expression available for the posterior.

Bayesian problem formulation again

Consider a Bayesian SSM

$$\begin{aligned} x_{t+1} \mid x_t &\sim f_{\theta,t}(x_{t+1} \mid x_t, u_t), \\ y_t \mid x_t &\sim h_{\theta,t}(y_t \mid x_t, u_t), \\ x_1 &\sim \mu_\theta(x_1), \\ \theta &\sim p(\theta). \end{aligned}$$

We observe $D_T \triangleq \{u_{1:T}, y_{1:T}\}$.

Goal: Compute the posterior $p(\theta, x_{1:T} \mid D_T)$.

Markov chain Monte Carlo (MCMC) methods allows us to generate samples from an arbitrary target distribution by simulating a Markov chain.

Gibbs sampling (blocked) for SSMs amounts to iterating

- Draw $\theta[m] \sim p(\theta \mid x_{1:T}[m-1], D_T)$,
- Draw $x_{1:T}[m] \sim p(x_{1:T} \mid \theta[m], D_T)$.

The result is a Markov chain

$$\{\theta[m], x_{1:T}[m]\}_{m \geq 1}$$

with $p(\theta, x_{1:T} \mid D_T)$ as its stationary distribution!

What would a Gibbs sampler for a general nonlinear/non-Gaussian SSM look like?

- Draw $\theta[m] \sim p(\theta \mid x_{1:T}[m-1], D_T)$,
- Draw $x_{1:T}[m] \sim p(x_{1:T} \mid \theta[m], D_T)$.

Problem: $p(x_{1:T} \mid \theta, D_T)$ is not available!!

Idea: Approximate $p(x_{1:T} \mid \theta, D_T)$ using a particle smoother (PS).

(Non-trivial) solution: Careful and clever analysis of how to combine MCMC and PF/PS results in the PMCMC family of algorithms.

PMCMC methods

Facts about Particle Markov Chain Monte Carlo (PMCMC) samplers:

- Provides a systematic and provably correct combination of PF/PS and MCMC.
- Standard MCMC samplers on non-standard spaces.
- Constitutes a family of Bayesian inference methods, including
  - Particle Independent Metropolis Hastings (PIMH)
  - Particle Marginal Metropolis Hastings (PMMH)
  - **Particle Gibbs (PG)**

Christophe Andrieu, Arnaud Doucet and Roman Holenstein, **Particle Markov chain Monte Carlo methods**, *Journal of the Royal Statistical Society: Series B*, 72:269-342, 2010.

PG with backward simulation (PG-BS) sampler targeting $p(\theta, x_{1:T} \mid D_T)$.

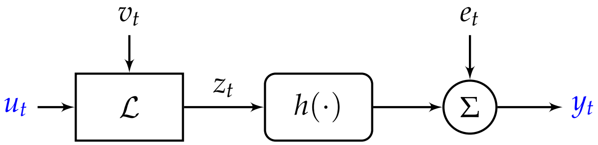
- Conditional particle filter (CPF) and backward simulation
  - Run a CPF, targeting $p(x_{1:T} \mid \theta, D_T)$;
  - Run a backward simulator to sample $x^\star_{1:T}$;
- Draw $\theta^\star \sim p(\theta \mid x^\star_{1:T}, D_T)$.

Powerful and important property of PG-BS: Provably convergent for any $N \geq 2$ particles and it works in practice!

Similarly to PG-BS, we use backward sampling to (considerably) improve the mixing of the PG kernel. Instead of using separate forward and backward sweeps as in PG-BS, however, PG-AS achieve the same effect in a single forward sweep.

Fredrik Lindsten, Michael I. Jordan and Thomas B. Schön. **Ancestor Sampling for Particle Gibbs**. *Proceedings of Neural Information Processing Systems (NIPS)*, Lake Tahoe, NV, USA, December, 2012.

Example – semiparametric Wiener model



Parametric LGSS and a nonparametric static nonlinearity:

$$x_{t+1} = \underbrace{\begin{pmatrix} A & B \end{pmatrix}}_{\Gamma} \begin{pmatrix} x_t \\ u_t \end{pmatrix} + v_t, \qquad v_t \sim \mathcal{N}(0, Q),$$

$$z_t = C x_t.$$

$$y_t = h(z_t) + e_t, \qquad e_t \sim \mathcal{N}(0, R).$$

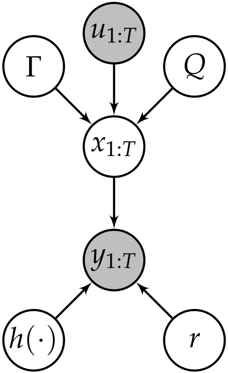
First step towards a fully data driven model (the order of the LGSS model is still assumed known).

Parameters: $\theta = \{\Gamma, Q, r, h(\cdot)\}$.

Bayesian model specified by priors:

- Conjugate priors for $\Gamma = [A\ B]$, Q and r,
  - $p(\Gamma, Q) =$ Matrix-normal inverse-Wishart
  - $p(r) =$ inverse-Wishart
- Gaussian process prior on $h(\cdot)$,

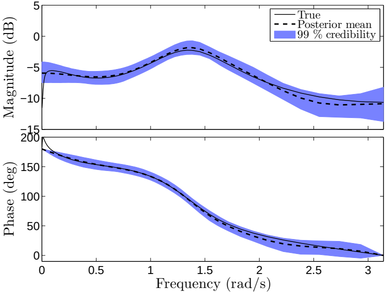
$$h(\cdot) \sim \mathcal{GP}(z, k(z, z')).$$



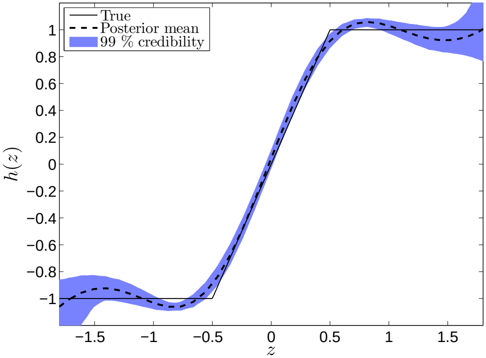
Example – known model order (I/II)

We can **quantify the uncertainty for this model** rather nicely.

- Bayesian semiparametric model with conjugate prior (MNIW).
- 6^{th} order LGSS model and a saturation.
- Using $T = 1000$ measurements.
- Employ the PG-BS sampler with $N = 15$ particles.
- Run 15000 MCMC iterations, discard 5000 as burn-in.

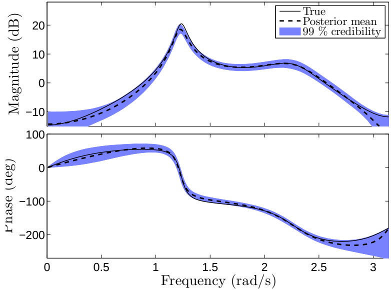


True Bode diagram of the linear system (solid black), estimated mean (dashed black) and 99% credibility interval (blue).

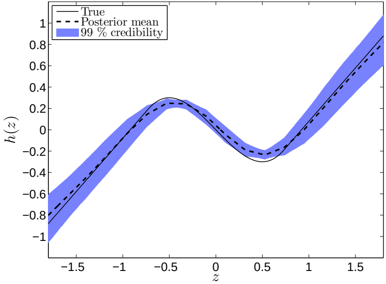


True static nonlinearity (solid black), estimated posterior mean (dashed black) and 99% credibility interval (blue).

Show movie



Bode diagram of the 4th-order linear system. Estimated mean (dashed black), true (solid black) and 99% credibility intervals (blue).



Static nonlinearity (non-monotonic), estimated mean (dashed black), true (black) and the 99% credibility intervals (blue).

Fredrik Lindsten, Thomas B. Schön and Michael I. Jordan. **Bayesian semiparametric Wiener system identification**. *Automatica*, 2013 (accepted for publication).

Parameter inference

1. The nonlinear Maximum Likelihood (ML) problem
   - Problem formulation
   - Solution using expectation maximization and a particle smoother
2. The nonlinear Bayesian problem
   - Problem formulation
   - Sketch of solution using MCMC and SMC

Sensor fusion

1. Problem formulation
2. Three industrial application examples

The sensor fusion problem

- Inertial sensors
- Camera
- Barometer

- Inertial sensors
- Radar
- Barometer
- Map

- Inertial sensors
- Cameras
- Radars
- Wheel speed sensors
- Steering wheel sensor

- Inertial sensors
- Ultra-wideband

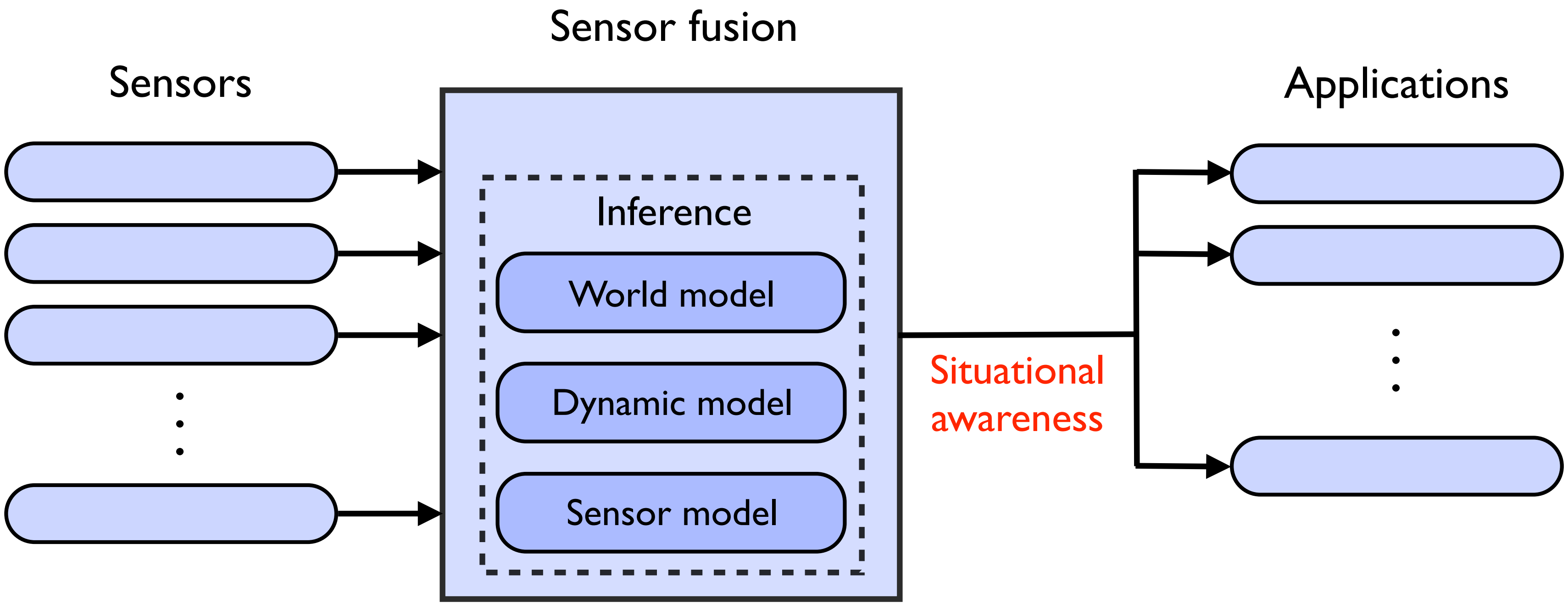
How do we combine the information from the different sensors?

Might all seem to be very different problems at first sight. However, the same strategy can be used in dealing with all of these applications.

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).



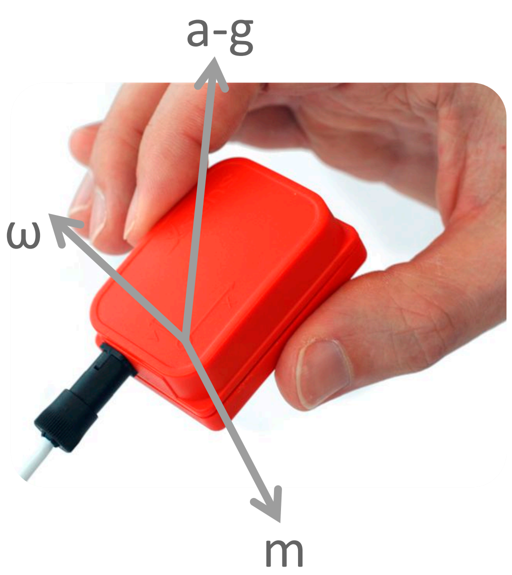
Illustrative example (I/III)

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

Industrial partner: Xsens Technologies.

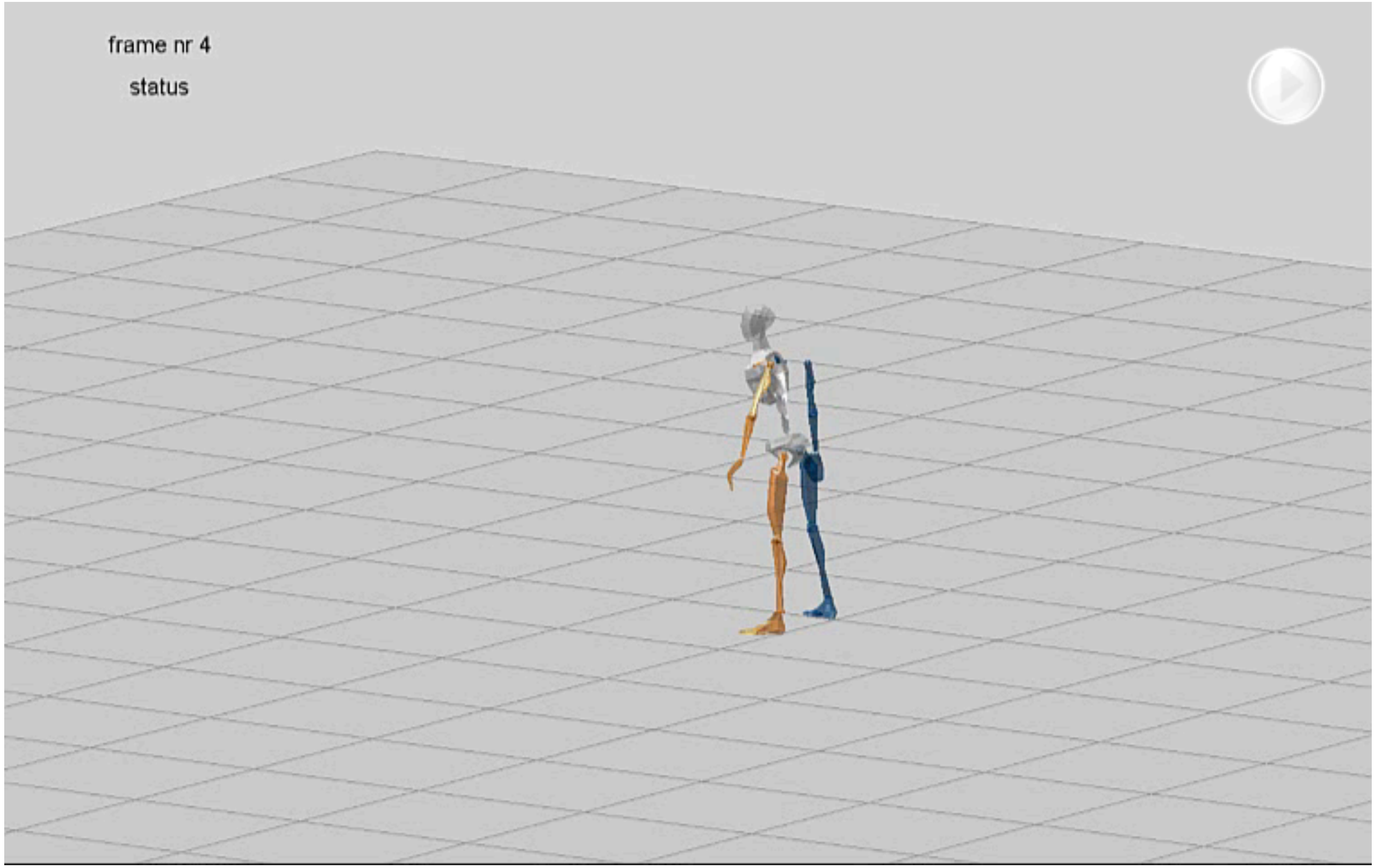
Sensors used:

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



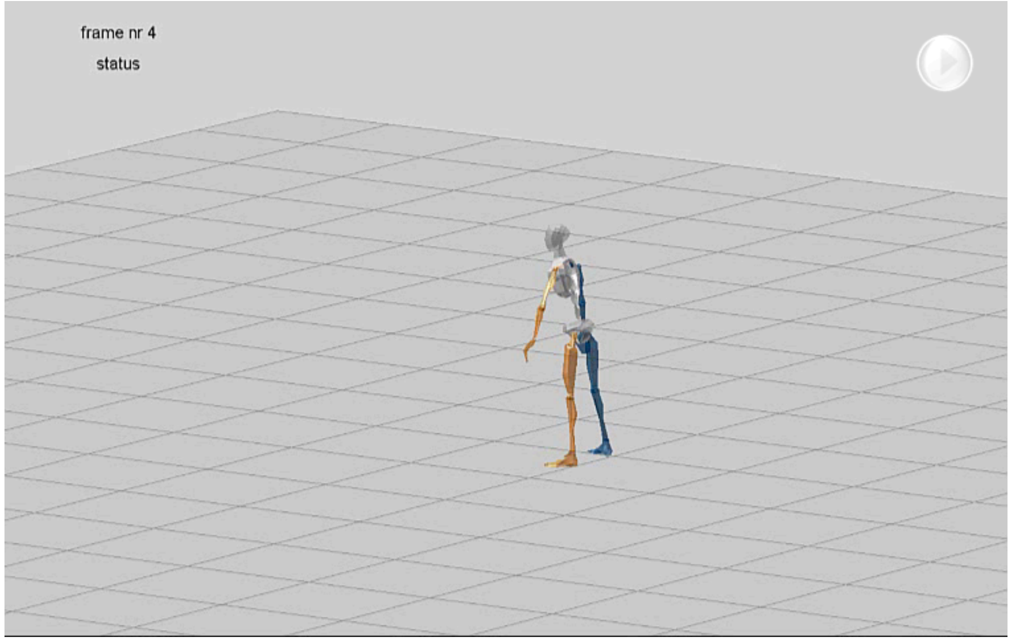
17 sensor units are mounted onto the body of the person.

1. Only making use of the inertial information.

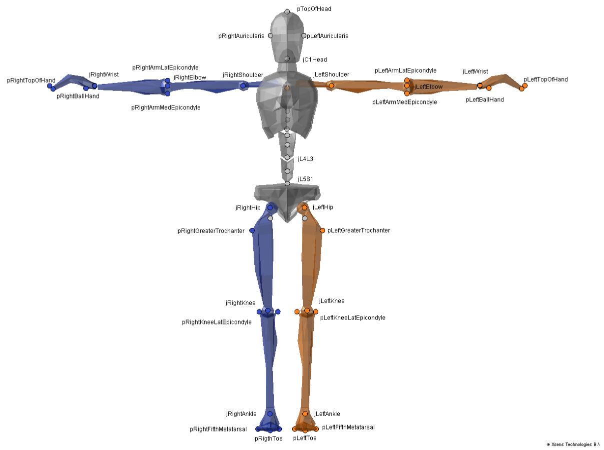


Movie courtesy of Daniel Roetenberg (Xsens)

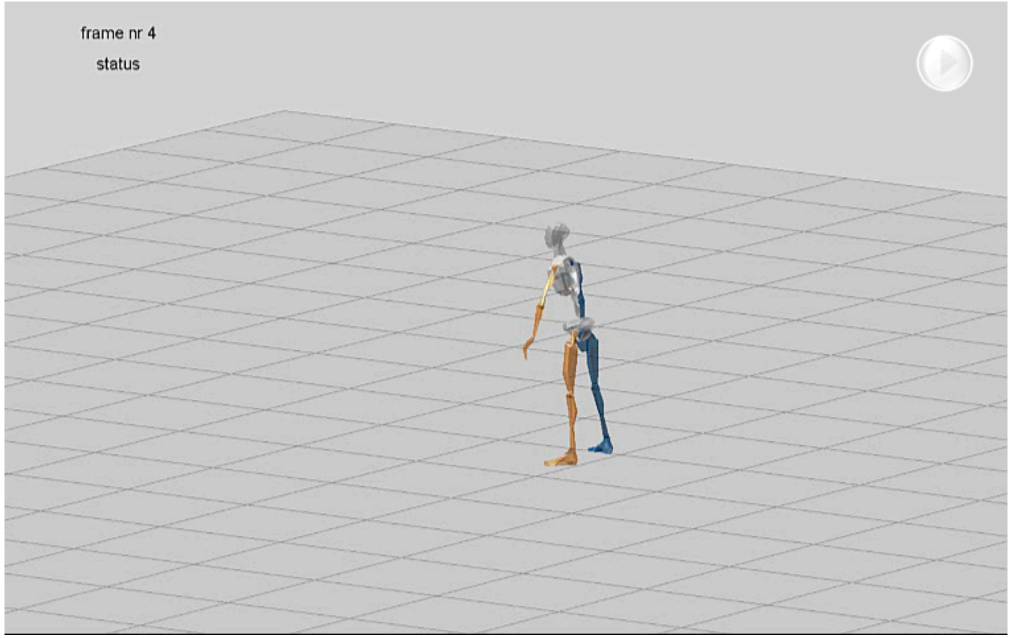
2. Inertial + biomechanical model



Movie courtesy of Daniel Roetenberg (Xsens)

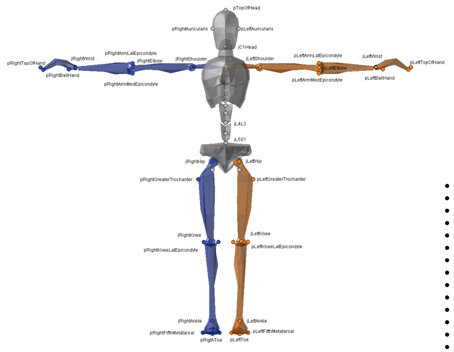


3. Inertial + biomechanical model + world model



Movie courtesy of Daniel Roetenberg (Xsens)

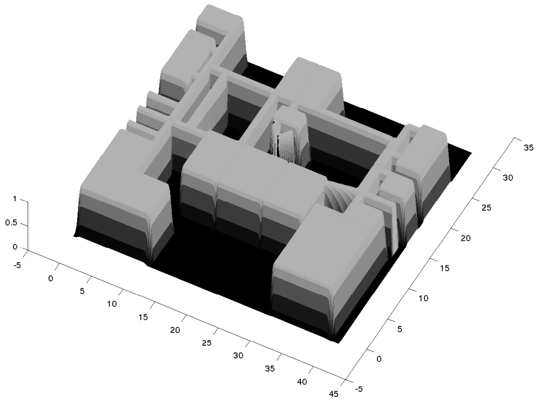
The story I am telling



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

1. We are dealing with dynamical systems

This requires a **dynamical model**.



2. The dynamical systems exist in a context.

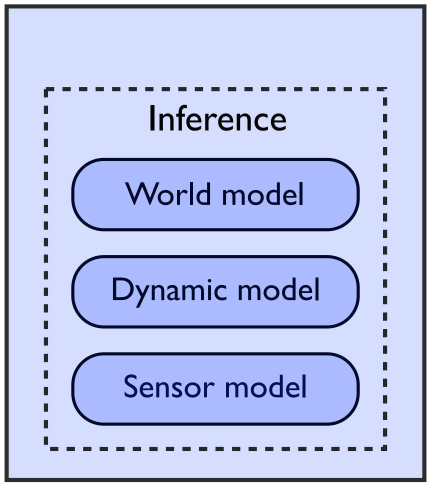
This requires a **world model**.

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

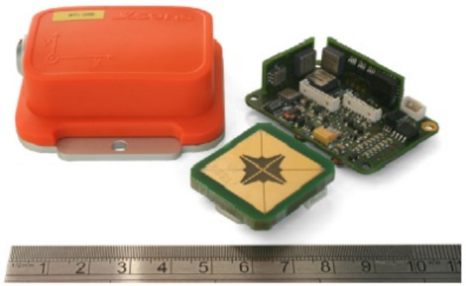
This requires sensors and **sensor models**.

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**.



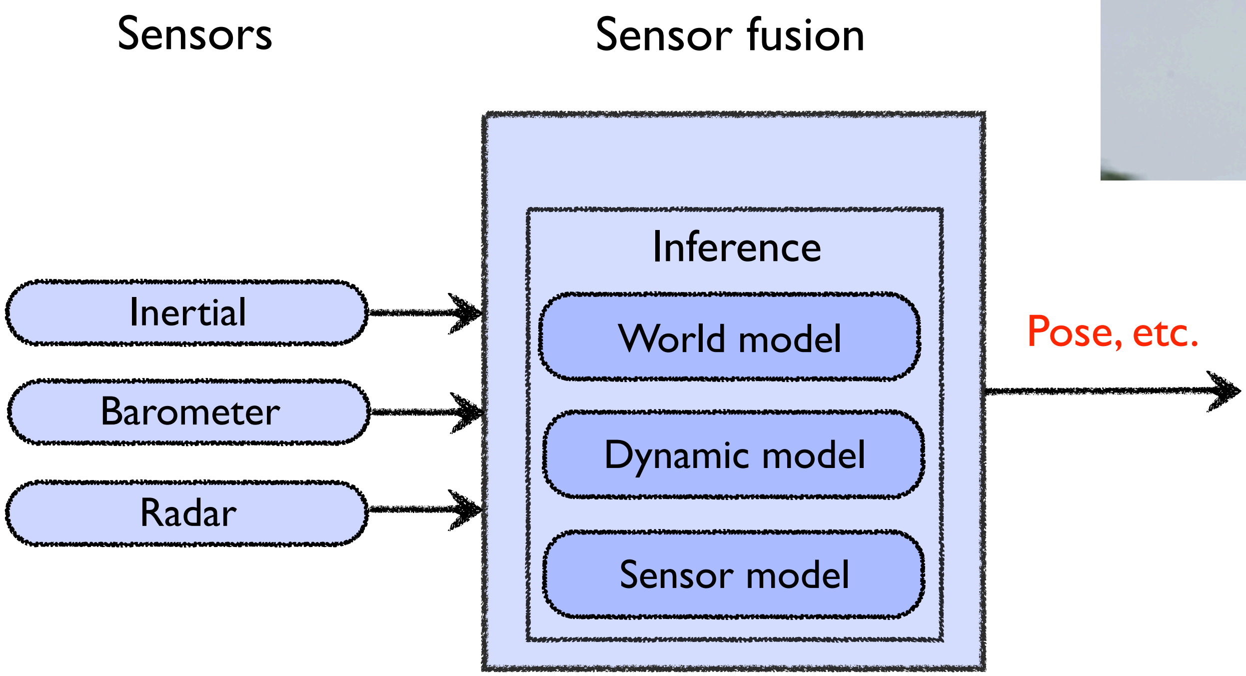
In this experiment we also make use of ultra-wideband (UWB).
This allows for indoor positioning as well.



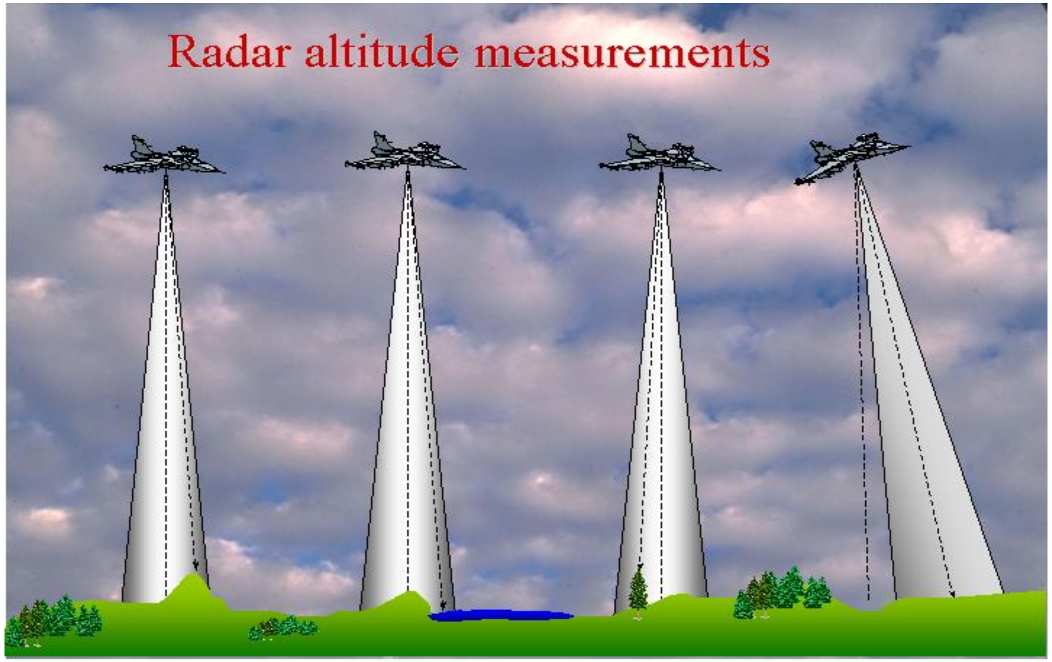
Example 2 - Fighter aircraft navigation

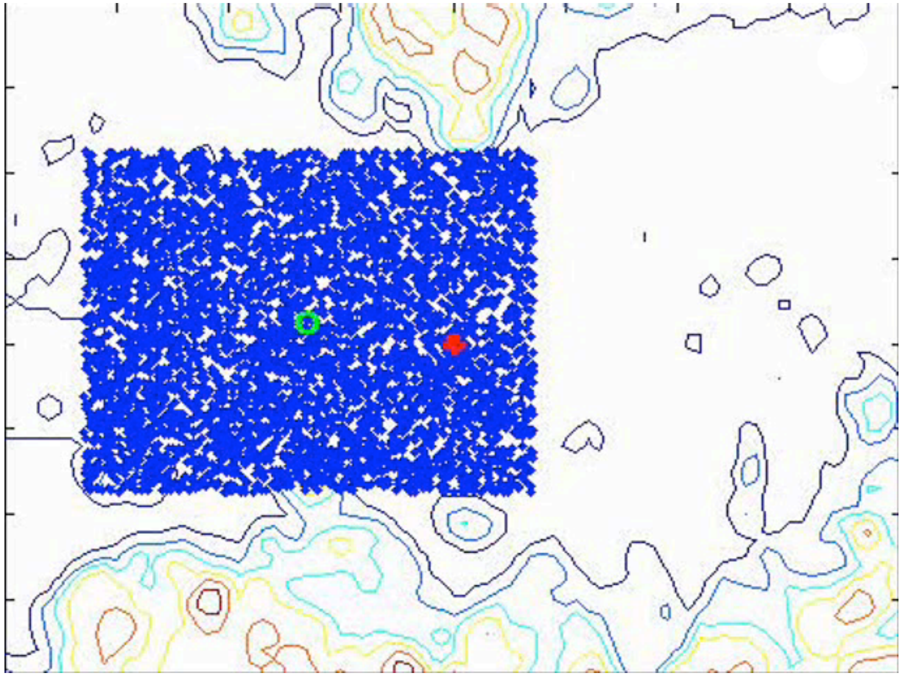
Aim: Find the position, velocity and orientation of a fighter aircraft.

Industrial partner: Saab



Example 2 - Fighter aircraft navigation





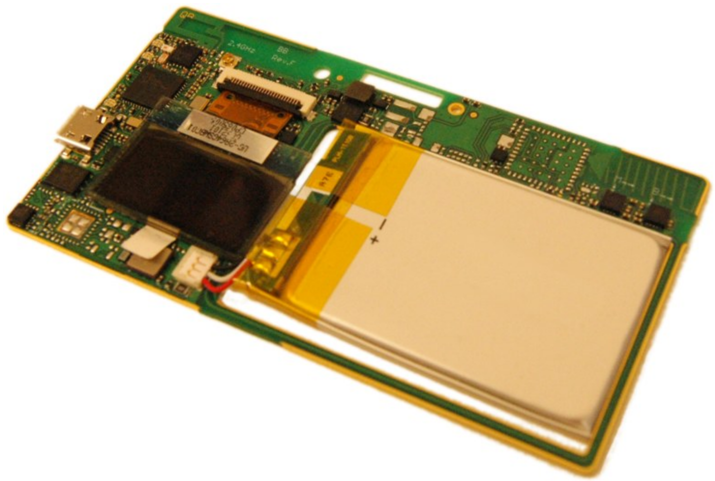
"Think of each particle as one simulation of the system state (in the movie, only the horizontal position is visualized). Only keep the good ones."

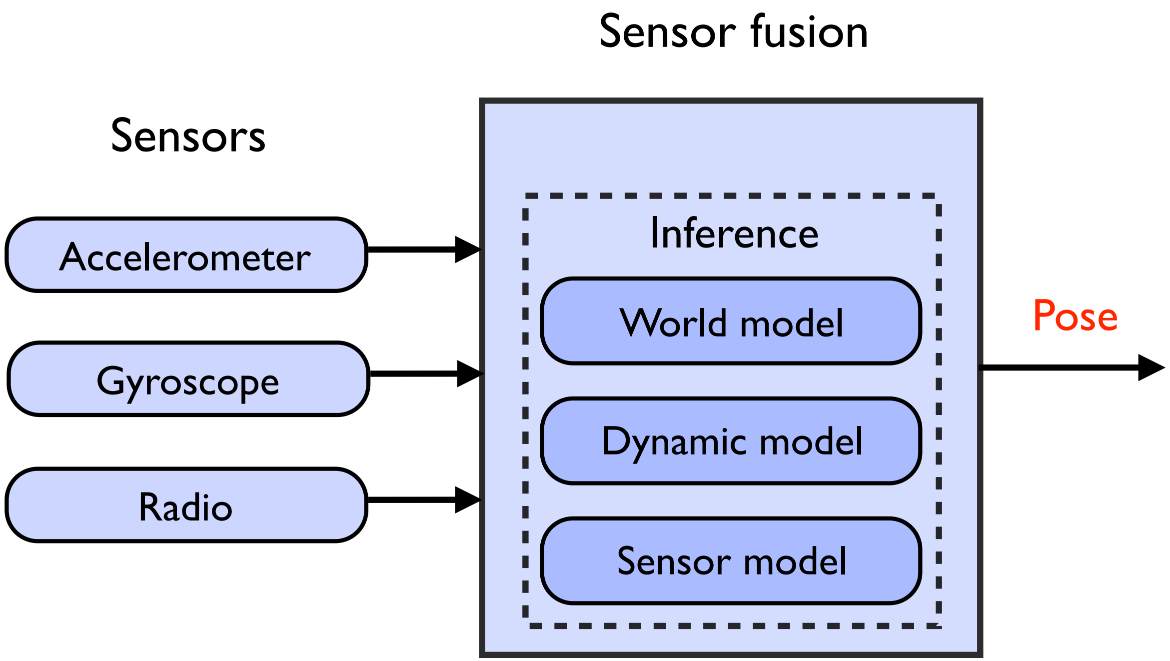
Thomas Schön, Fredrik Gustafsson, and Per-Johan Nordlund. **Marginalized Particle Filters for Mixed Linear/Nonlinear State-Space Models**. *IEEE Transactions on Signal Processing*, 53(7):2279-2289, July 2005.

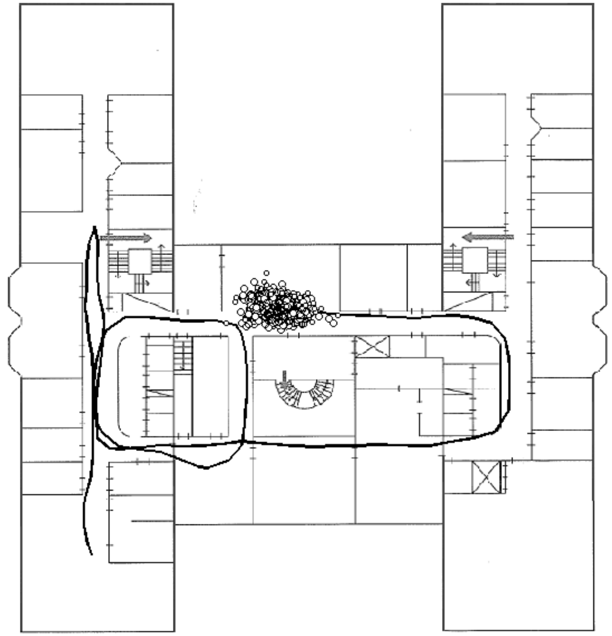
Example 3 - Indoor localization (I/III)

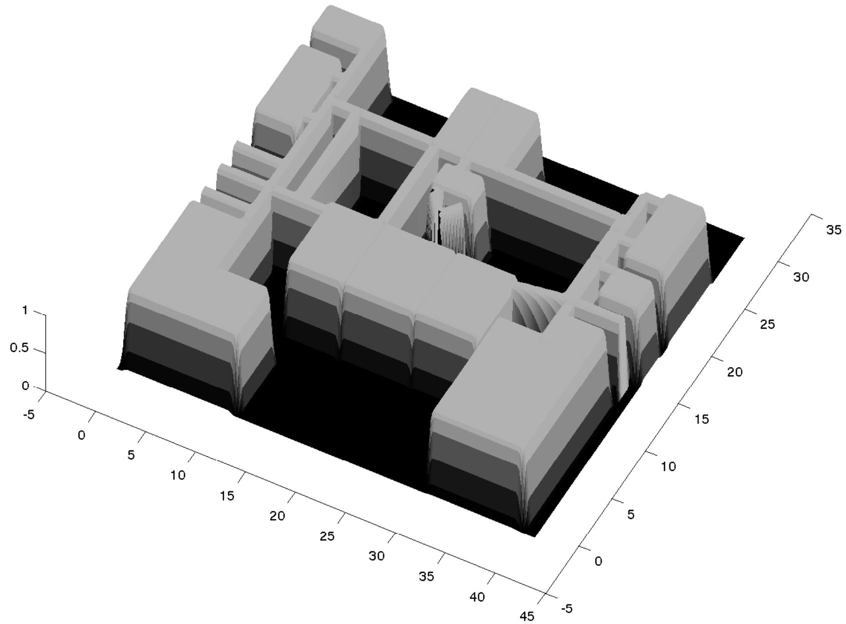
Aim: Compute the position of a person moving around indoors using sensors (inertial, magnetometer and radio) located in an ID badge and a map.

Industrial partner: Xdin

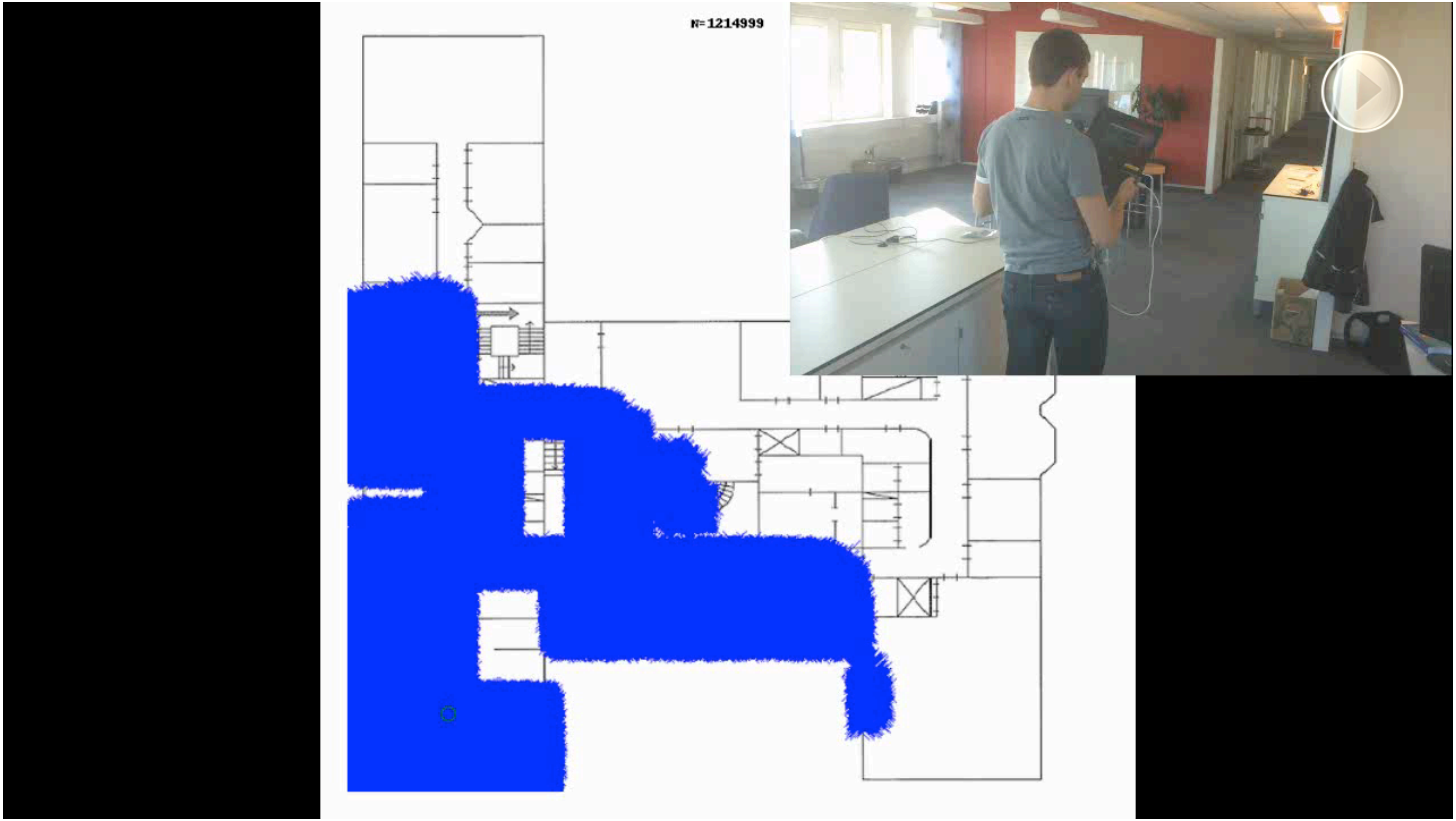








PDF of an office environment, the bright areas are rooms and corridors (i.e., walkable space).


N=1214999

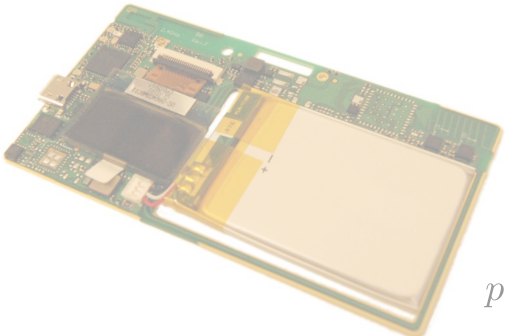
Conclusions

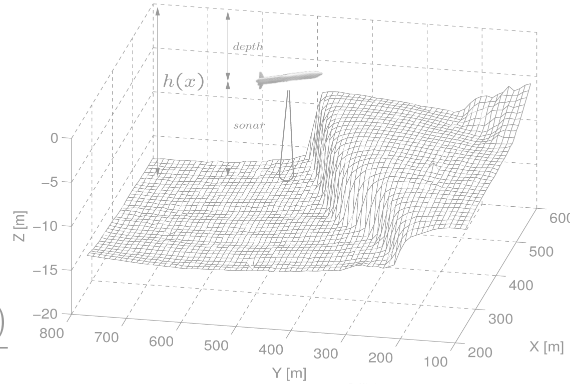
- Maximum likelihood identification
    - EM (nonlinear optimization and particle smoothing)
- Bayesian identification
    - PMCMC = combination of MCMC and PF/PS
    - We use Particle Gibbs with ancestor sampling
- Solved various Wiener identification problems for illustration
- Sensor fusion
    - Model the sensors, the dynamics and the world. Solve the resulting inference problem.
    - The industrial utility of this technology is growing as we speak!
- Much interesting research **remains to be done**!!

In this talk I introduced strategies and showed a few concrete example. Should you be interested in the details I am developing a PhD course on the topic of computational inference in dynamical systems,

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

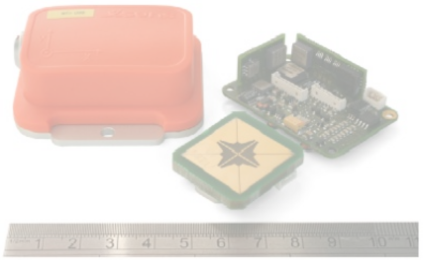
Thank you for your attention!!

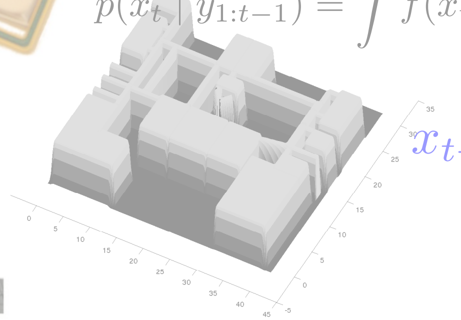


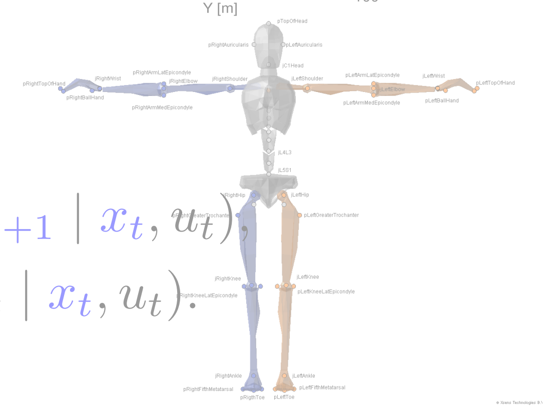


$$p(x_t \mid y_{1:t}) = \frac{h(y_t \mid x_t)p(x_t \mid y_{1:t-1})}{p(y_t \mid y_{1:t-1})}$$

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







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

Joint work with (alphabetical order): **Fredrik Gustafsson** (Linköping University), **Jeroen Hol** (Xsens technologies), **Michael I. Jordan** (UC Berkeley), **Johan Kihlberg** (Xdin), **Fredrik Lindsten** (Linköping University), **Lennart Ljung** (Linköping University), **Brett Ninness** (University of Newcastle, Australia), **Per-Johan Nordlund** (Saab), **Simon Tegelid** (Xdin) and **Adrian Wills** (MRA, Newcastle, Australia).