# Unscented Kalman Filter

S-88.4221 Postgraduate Seminar on Signal Processing

Pekka Jänis

# Outline

- Unscented transformation (UT)
- Unscented Kalman filter (UKF)
- Example / Matlab-demo
- Homework

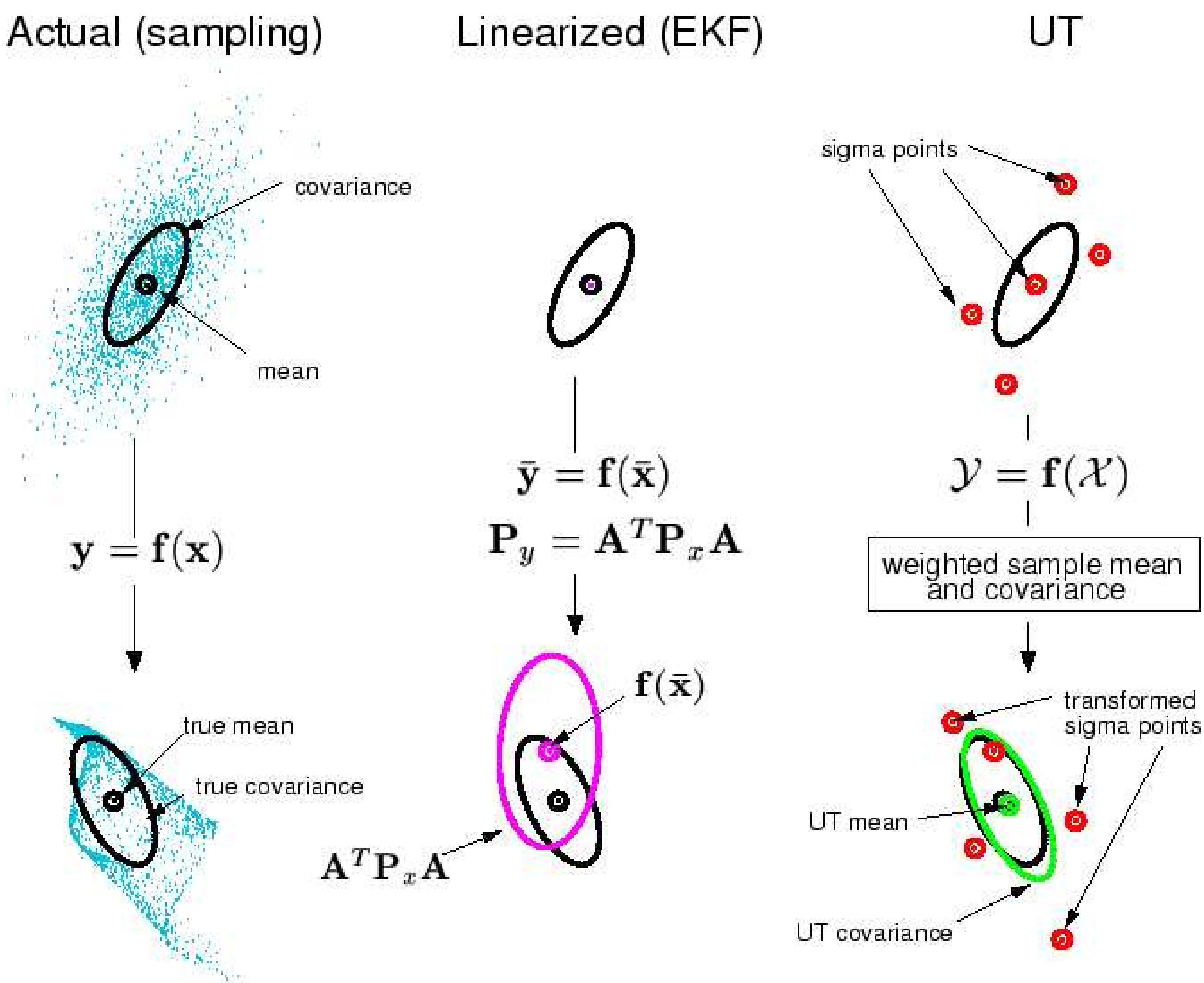
# Why yet another KF

- EKF is difficult to tune, the Jacobian can be hard to derive, and it can only handle limited amount of nonlinearity
- PF can handle arbitrary distributions and non-linearities but is computationally very complex
- UKF gives a nice tradeoff between PF and EKF

# Unscented transformation

- Problem: Given an $n$-dimensional r.v. $x$ with mean and covariance $(\bar{x}, P_x)$, find $(\bar{y}, P_y)$ where $y = h(x)$ is a transformed r.v. and $h$ general (non-linear) function.
- Solutions:
  - Solve analytically $\rightsquigarrow$ gray hair.
  - Use Taylor series $\rightsquigarrow$ EKF.
  - Use UT $\rightsquigarrow$ UKF.
  - (Monte-Carlo integration $\rightsquigarrow$ particle filter.)

# Unscented transformation



# Unscented transformation

- $n$-dimensional r.v. $x$ is approximated with $2n+1$ sigma points $x^{(i)}$ and weights $W^{(i)}$.
- The sigma points are chosen such that the weighted "sample mean and covariance" of the sigma points match $(\bar{x}, P_x)$.
- Transformed sigma points yield then approximate $(\bar{y}, P_y)$ of r.v. $y = h(x)$.

# Unscented transformation

$$x^{(0)} = \bar{x}, \qquad W^{(0)} = \frac{\kappa}{n+\kappa}$$

$$x^{(i)} = \bar{x} + \left(\sqrt{(n+\kappa)P_x}\right)_i, \qquad W^{(i)} = \frac{1}{2(n+\kappa)}$$

$$x^{(i+n)} = \bar{x} - \left(\sqrt{(n+\kappa)P_x}\right)_i, \qquad W^{(i+n)} = W^{(i)}$$

$$\bar{y} = \sum_{i=0}^{2n} W^{(i)} f(x^{(i)}), \quad P_y = \sum_{i=0}^{2n} W^{(i)} (f(x^{(i)}) - \bar{y})(f(x^{(i)}) - \bar{y})^T \ldots$$

where $(\sqrt{A})_i$ denotes the $i$th column of a matrix square root of $A$ and $\kappa$ is a user-defined constant.

# Unscented transformation

The mean and covariance of r.v. $y = h(x)$ can be written using Taylor series expansion of $h$ as ($x = \bar{x} + \tilde{x}$)

$$\bar{y} = h(\bar{x}) + \frac{1}{2!}E[D_{\tilde{x}}^2 h] + \frac{1}{4!}E[D_{\tilde{x}}^4 h] + \dots$$

$$\begin{aligned} P_y = HP_xH^T &+ E[\frac{1}{3!}D_{\tilde{x}}h(D_{\tilde{x}}^3 h)^T + \frac{1}{2!2!}D_{\tilde{x}}^2 h(D_{\tilde{x}}^2 h)^T + \frac{1}{3!}D_{\tilde{x}}^3 h(D_{\tilde{x}}h)^T] \\ &+ E[\frac{1}{2!}D_{\tilde{x}}^2 h]E[\frac{1}{2!}D_{\tilde{x}}^2 h]^T + \dots \end{aligned}$$

where $D_{\tilde{x}}^k h$ is shorthand notation:

$$D_{\tilde{x}}^k h = \left( \sum_{i=1}^{n} \tilde{x}_i \frac{\partial}{\partial x_i} \right)^k h(x)\Big|_{\tilde{x}}.$$

# Unscented transformation

- UT gives correct mean up to third order and covariance up to the second order for any $h$. Linearized mean is correct only up to first order. Covariance in UT and linearization have the same order of accuracy. The magnitude of the error is, however, smaller in UT.
- UT with $\kappa \neq 0$ is termed generalized UT. With Gaussian $x$ choosing $\kappa = 3 - n$ gives smaller fourth order terms of the error in $(\bar{y}, P_y)$.
- There are also other UT's available, e.g. the spherical UT which gives the same *order of* accuracy but with less sigma points ($n + 1$).

# Unscented transformation

- UT actually resembles MC, but here the points are chosen in a deterministic way.
- High order information is captured using a very small amount of points.
- In UT, the distribution of $x$ is approximated by the sigma points and the function $h$ is kept intact. In EKF the distribution is approximated with mean and covariance, and $h$ is approximated.
- The choice of a specific $\sqrt{P}$ does not influence the properties of UT. Cholesky factor is a good pick.

# The System

We have $n$-state discrete-time non-linear system

$$
\begin{aligned}
x_{k+1} &= f(x_k, u_k, t_k, w_k) \\
y_k &= h(x_k, t_k, v_k) \\
w_k &\sim (0, Q_k) \\
v_k &\sim (0, R_k),
\end{aligned}
$$

where we have included the noise terms in $f$ and $h$ since they may not be additive. If they are not additive they must be augmented to the state variable.

# Unscented Kalman Filter

(0.) Initialize mean and covariance as usual
1. Do the time update from $(x_{k-1}^+, P_{k-1}^+)$ to $(x_k^-, P_k^-)$ using UT and $f$
2. (Optional) Calculate new sigma points about $(x_k^-, P_k^-)$ (or stick with the ones from $(x_{k-1}^+, P_{k-1}^+)$)
3. Transform the sigma points of step 2 with $h$ to get predicted measurements sigma points $\hat{y}^{(i)}$.
4. Calculate $\hat{y}_k$, $P_y$, and $P_{xy}$ as "sample statistics" of the sigma points. (note that this is not a random sample!)
5. Now, simply,

$$K_k = P_{xy} P_y^{-1}$$

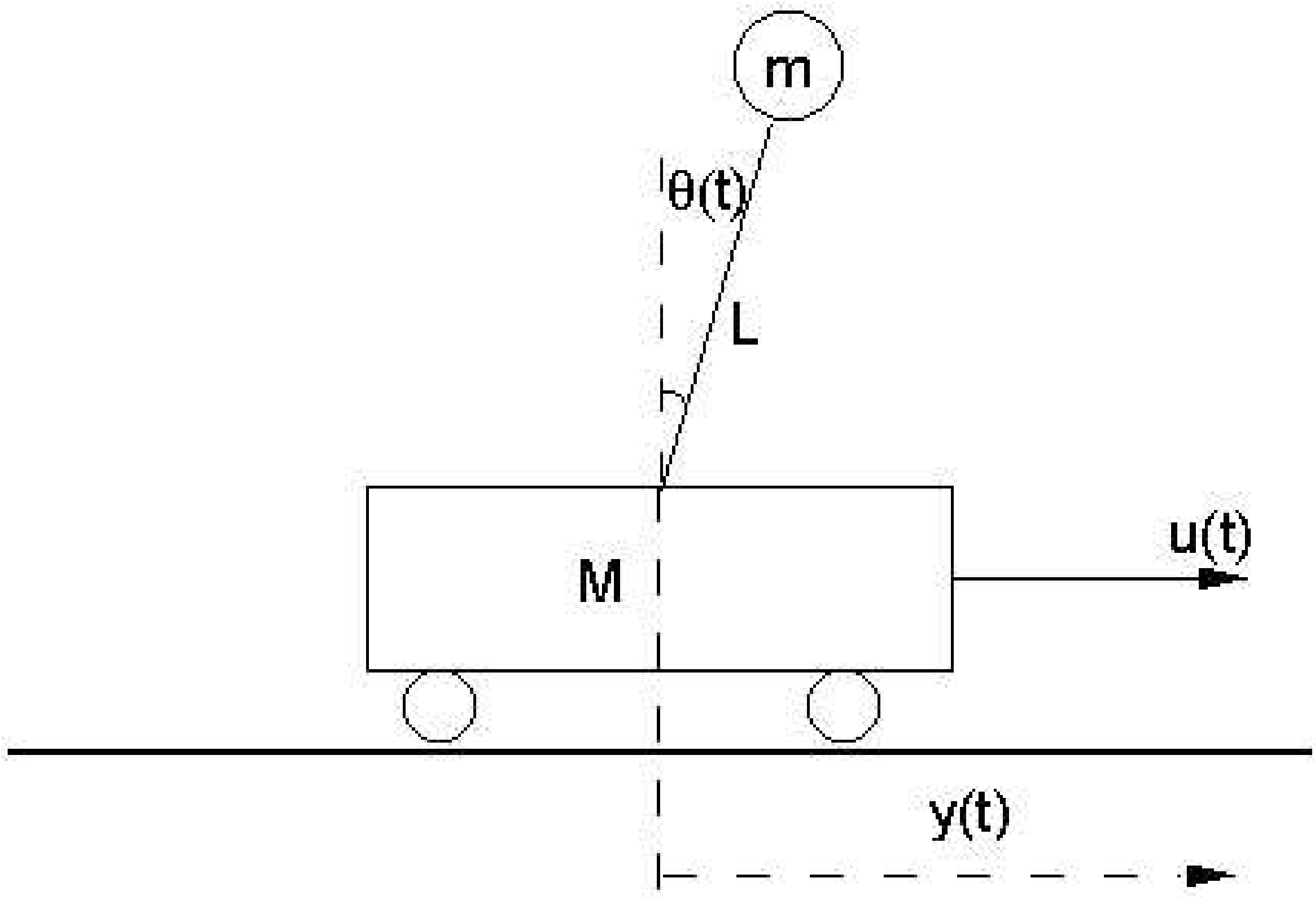
$$x_k^+ = x_k^- + K_k(y_k - \hat{y}_k)$$

$$P_k^+ = P_k^- - K_k P_y K_k^T$$

# To UKF or not to UKF?

- Choose UKF when
  - $f$ and/or $h$ non-linear.
  - difficulties in EKF implementation / poor EKF performance
- Go instead for
  - KF if model is linear
  - PF if model is “strange” ($f$, $h$ too non-linear, distributions are e.g. bimodal...)
  - EKF if it works fine and is computationally cheaper.

# Example / Inverted Pendulum



# Why unscented?

scented

*adjective*

1. having the sense of smell; "keen-scented hounds" [ant: scentless]
2. filled or impregnated with perfume; "perfumed boudoir"; "perfumed stationery"; "scented soap" [syn: perfumed]
3. having a natural fragrance; "odoriferous spices"; "the odorous air of the orchard"; "the perfumed air of June"; "scented flowers" [syn: odoriferous]
4. (used in combination) having the odor of; "clean-scented laundry"; "a manure-scented barnyard"



The following slides are adapted from Simon Julier's UKF slides:

`http://soma.crl.mcmaster.ca/ASLWeb/Resources/data/LakeLouise2003/Julier_slides_trans.pdf`

# Why unscented?

- Some history:
  - The algorithm later to be known as the unscented transformation was derived in the summer of 1994 at the Robotics Research Group (RRG), Oxford UK.
  - In a 1995 ACC paper it was called the New Filter Due to the somewhat ambiguous (and boring) nature of the name, it was decided that a new and better name was needed.
- Naming process:
  - Approximately 20 different names were identified. Any involving names of authors or institutions were automatically excluded (other options included the Sigma Point Filter and the FAB Filter).
  - A democratic vote was taken by members of RRG to choose the name Unscented Kalman Filter.

# But why unscented?

- Further research showed that the concept of deterministically choosing points to match statistics could be generalised out of a Kalman filter context.
- Therefore, the terminology was redefined. The UT was defined to be the transformation process to make sure that it can stand on its own.
- Any claims that the name was chosen to imply that the EKF stinks or to irritate Hugh (Durrant-Whyte) are pure speculation. Honest.
  - [Hugh Durrant-Whyte is(was?) a robotics professor. Apparenty *he* stinks. This seems to be a control theorists' inside joke]

# the difficult and time-consuming homework. . .

. . . just kidding, Matlab=lotsa fun! And, moreover, ppl should have a lot of spare time during holidays. Think of it as me saving you from boredom.

Get the EKF/UKF toolbox from LCE@TKK `http://www.lce.hut.fi/research/mm/ekfukf/`.

Do a modified version of problem 14.15 in the book. The catch: Dan's model solution is extremely messy and there are some discrepancies between it and problem statement.

# the difficult and time-consuming homework. . .

The IP model and quantities are as in book. The text is modified as follows:

- The angle $\theta$ is measured every 10 ms with a std. of 0.1 rad. `H=[0 0 1 0];`
- Discrete-time process noise covariance is $Q = \text{diag}(0, 0.0004, 0, 0.04)dt$.
- Initial state is `x(:,1)=[0 0 .4 0]';` and UKF does not know it (you may init `xp(:,1)=[0 0 0 0]';`, `Pp(:,:,1)=eye(4)`)
- Feedback control is given as

$$u = 100\hat{\theta} + 20\hat{\dot{\theta}},$$

and it is perfectly known to the UKF (see, the state is not known but the control is, which is somewhat unrealistic assumption).

# the difficult and time-consuming homework. . .

Since I appreciate being alive and all, here's some hints:

- Do not look at Dan's code pp. 285→ in the the solution manual. It will probably just confuse you.
- If you disregarded previous hint: You'll be better off by defining only an 4×N array `x` for the state instead of Dan's 10+ variables on the order of `theta`, `thetadot`, `thetadotdot_array`, `theta_old`, and so on. At least I had a near-death experience with those.
- Get my supplementary functions from `http://www.tkk.fi/~pjanis/`. Or write your own if you don't trust me. (Might even be wise. . . )
- Take a look at the toolbox manual
- If all else fails, email `pekka.janis@tkk.fi` for advice.