Extended target tracking using Gaussian processes

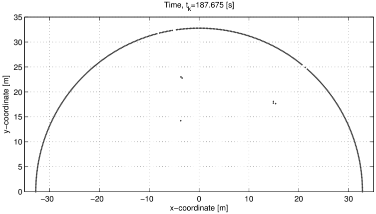
Niklas Wahlström, Emre Özkan

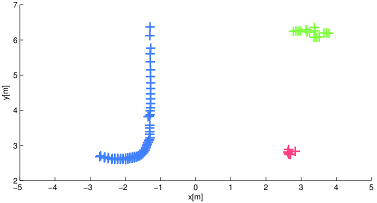
Division of Automatic Control
Linköping University
Linköping, Sweden

May 28, 2015

Extended target tracking

Many sensors generates more than one measurement per target





Extended target (definition)

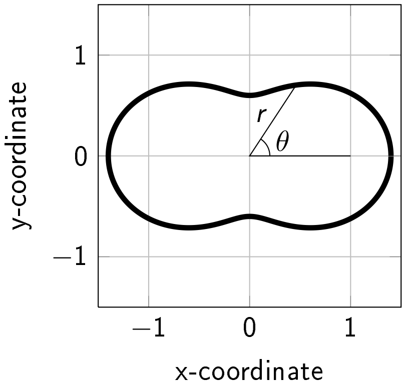
Targets that potentially give rise to multiple measurements at each time step

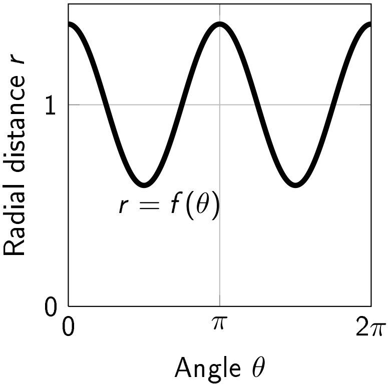
Goal: We want to estimate target position, target orientation and target extent *jointly*.

Related work

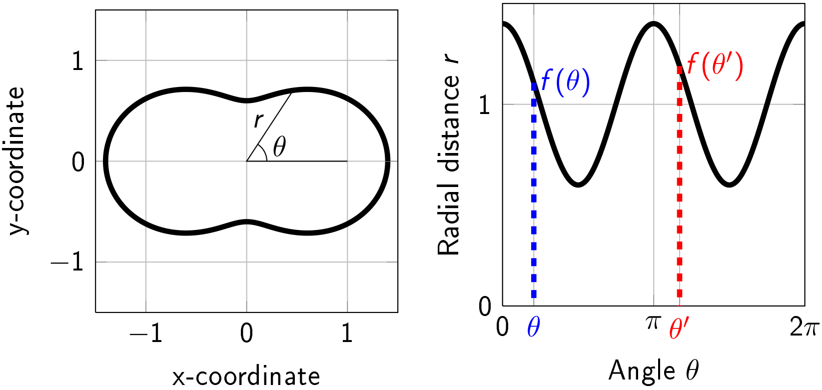
- Elliptical targets using inverse Wishart prior
  - First paper (Koch, Trans. Aerospace 2008).
  - Multiple ellipses per target (Lan and Li, Fusion 2012).
  - Encoding orientation (Granström and Orguner, Trans. Aeorspace 2014)
- Parametrized objects, rectangles, ellipses etc. (Granström, Fusion 2011)
- Random hyper-surface model (Baum and Hanebeck, Fusion 2011)
- Computer vision: B-splines (Blake et al., 1993, 1995, 1998)

Modeling using polar coordinates





Modeling using polar coordinates

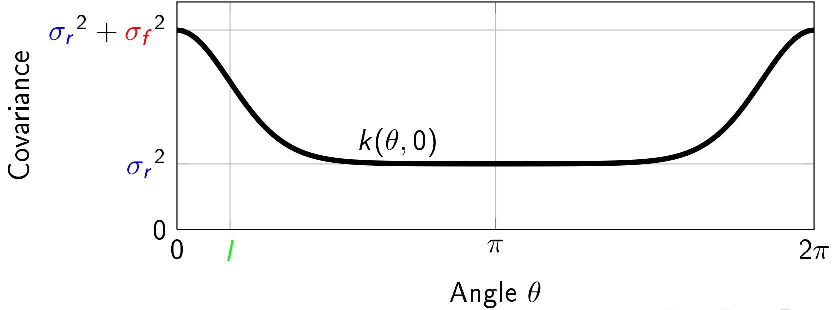


We model $f(\theta)$ using a Gaussian process.

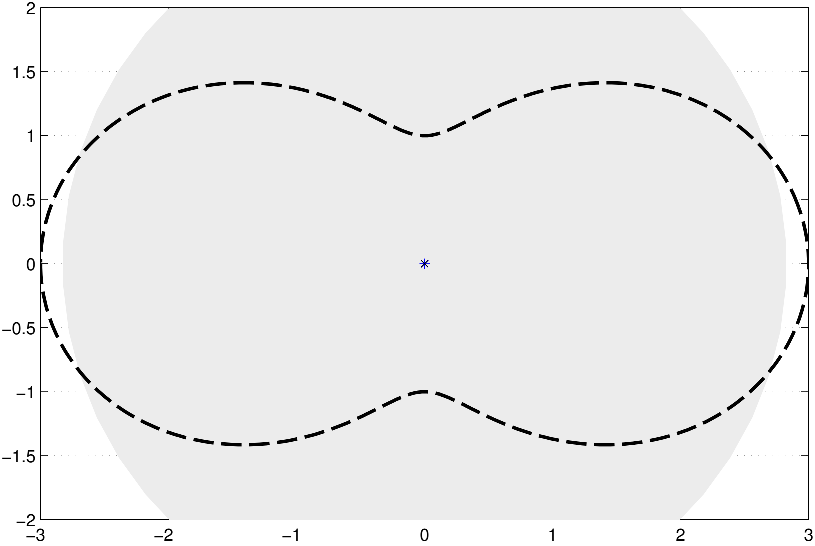
$$f(\theta) \sim \mathcal{GP}(0, k(\theta, \theta')), \qquad \mathbb{E}[f(\theta)f(\theta')] = k(\theta, \theta')$$

- We use a periodic covariance function to model the periodicity
- We use a additional constant covariance to model a constant (but unknown) mean, corresponding to the mean radius of the target

$$k(\theta, \theta') = {\sigma_f}^2 e^{-\frac{2\sin^2\left(\frac{|\theta-\theta'|}{2}\right)}{l^2}} + {\sigma_r}^2$$



Example

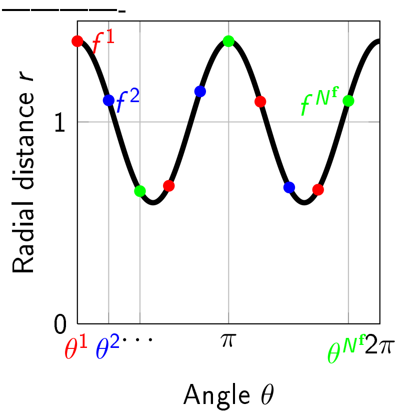


Idea: Consider function values $f^1, f^2, \dots, f^{N^{\mathrm{f}}}$ to be the state components

$$\mathbf{x}^{\mathrm{f}} = \begin{bmatrix} f^1 \\ f^2 \\ \vdots \\ f^{N^{\mathrm{f}}} \end{bmatrix}$$

This can be cast into a state space model

$$\mathbf{x}^{\mathrm{f}}_{k+1} = \mathbf{x}^{\mathrm{f}}_k$$
$$\mathbf{y}_k = H\mathbf{x}^{\mathrm{f}}_k + \mathbf{e}_k$$
$$\mathbf{x}^{\mathrm{f}}_0 \sim \mathcal{N}(0, P^{\mathrm{f}}_0)$$



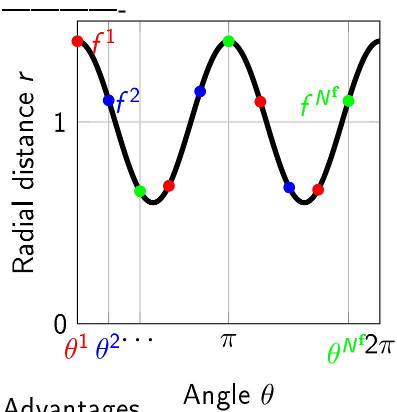
Recursive GP regression

Idea: Consider function values $f^1, f^2, \dots, f^{N^{\mathrm{f}}}$ to be the state components

$$\mathbf{x}^{\mathrm{f}} = \begin{bmatrix} f^1 \\ f^2 \\ \vdots \\ f^{N^{\mathrm{f}}} \end{bmatrix}$$

This can be cast into a state space model

$$\mathbf{x}^{\mathrm{f}}_{k+1} = \mathbf{x}^{\mathrm{f}}_k$$
$$\mathbf{y}_k = H\mathbf{x}^{\mathrm{f}}_k + \mathbf{e}_k$$
$$\mathbf{x}^{\mathrm{f}}_0 \sim \mathcal{N}(0, P^{\mathrm{f}}_0)$$



Advantages

- Recursive update with KF

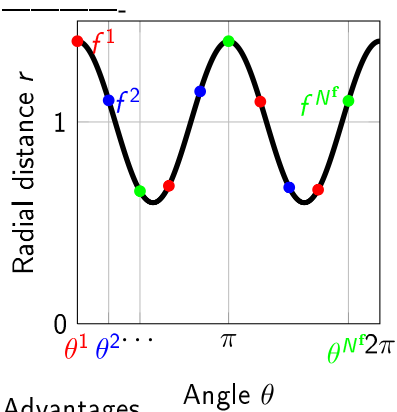
Recursive GP regression

Idea: Consider function values $f^1, f^2, \ldots, f^{N^\mathrm{f}}$ to be the state components

$$\mathbf{x}^\mathrm{f} = \begin{bmatrix} f^1 \\ f^2 \\ \vdots \\ f^{N^\mathrm{f}} \end{bmatrix}$$

This can be cast into a state space model

$$\mathbf{x}^\mathrm{f}_{k+1} = F\mathbf{x}^\mathrm{f}_k + \mathbf{w}_k$$
$$\mathbf{y}_k = H\mathbf{x}^\mathrm{f}_k + \mathbf{e}_k$$
$$\mathbf{x}^\mathrm{f}_0 \sim \mathcal{N}(0, P^\mathrm{f}_0)$$



Advantages

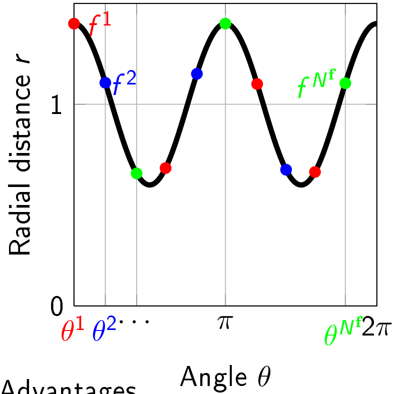
- Recursive update with KF
- Add process noise

Idea: Consider function values $f^1, f^2, \dots, f^{N^{\mathrm{f}}}$ to be the state components

$$\mathbf{x}^{\mathrm{f}} = \begin{bmatrix} f^1 \\ f^2 \\ \vdots \\ f^{N^{\mathrm{f}}} \end{bmatrix}$$

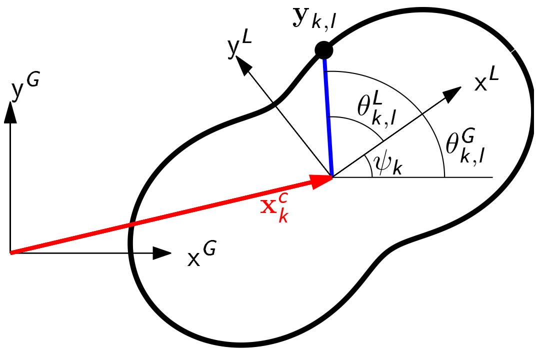
This can be cast into a state space model

$$\mathbf{x}^{\mathrm{f}}_{k+1} = F\mathbf{x}^{\mathrm{f}}_k + \mathbf{w}_k$$
$$\mathbf{y}_k = H\mathbf{x}^{\mathrm{f}}_k + \mathbf{e}_k$$
$$\mathbf{x}^{\mathrm{f}}_0 \sim \mathcal{N}(0, P^{\mathrm{f}}_0)$$



Advantages

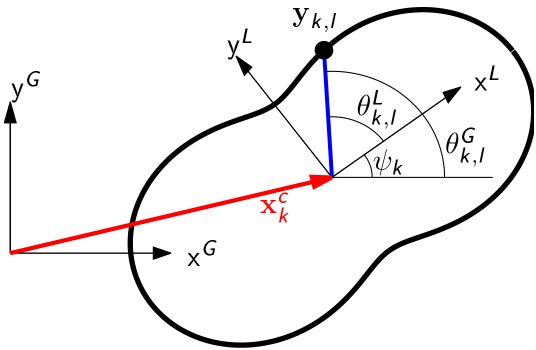
- Recursive update with KF
- Add process noise
- $\mathbf{x}^{\mathrm{f}}_k$ can be augmented with target position $\mathbf{x}^c_k$ and



Measurement is the sum of target position and offset due to target extent

$$\mathbf{y}_{k,l} = \mathbf{x}^c_k + \begin{bmatrix} \cos(\theta^G_{k,l}) \\ \sin(\theta^G_{k,l}) \end{bmatrix} f(\theta^L_{k,l}) + \mathbf{e}_{k,l}, \qquad \theta^G_{k,l} = \theta^G_{k,l}(\mathbf{x}^c_k)$$

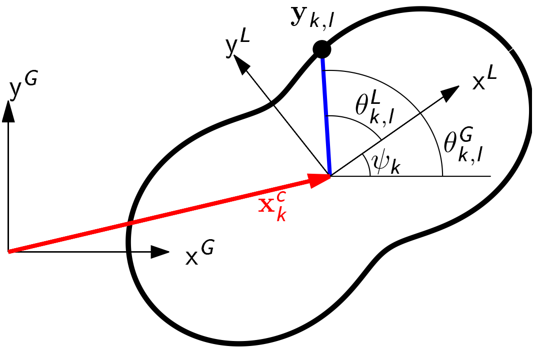
$$\theta^L_{k,l} = \theta^L_{k,l}(\mathbf{x}^c_k, \psi_k)$$



Measurement is the sum of target position and offset due to target extent

$$\mathbf{y}_{k,l} = \mathbf{x}^c_k + \begin{bmatrix} \cos(\theta^G_{k,l}) \\ \sin(\theta^G_{k,l}) \end{bmatrix} H(\theta^L_{k,l})\mathbf{x}^{\mathbf{f}}_k + \mathbf{e}_{k,l}, \qquad \theta^G_{k,l} = \theta^G_{k,l}(\mathbf{x}^c_k)$$

$$\theta^L_{k,l} = \theta^L_{k,l}(\mathbf{x}^c_k, \psi_k)$$



This can be summarized into a non-linear sensor model

$$\mathbf{y}_{k,l} = \mathbf{x}^c_k + H(\mathbf{x}^c_k, \psi_k)\mathbf{x}^{\mathbf{f}}_k + \mathbf{e}_{k,l}$$

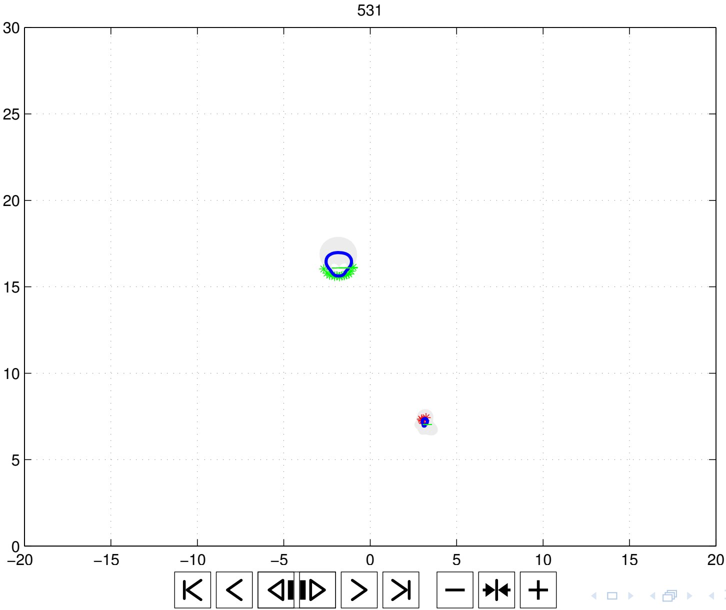
Real data experiment

- Laser range data
- Multi-target scenario (cars, bicycles, humans)
- Almost no clutter

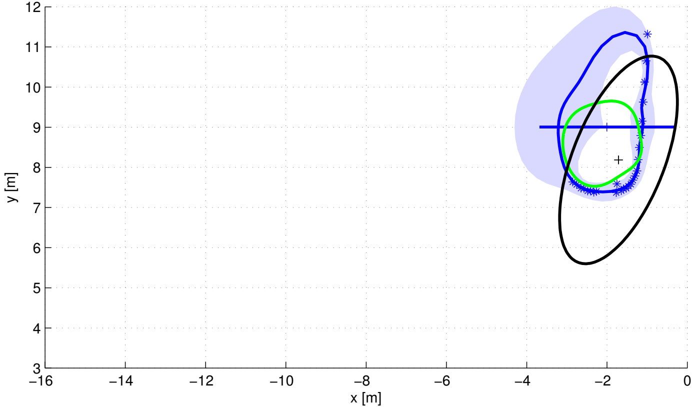
We used a simple logic-based multi-target tracker:

- Gating based likelihood
- Associate a measurement with the most likely target
- Cluster all ungated measurements and form new targets

Real data experiment -result

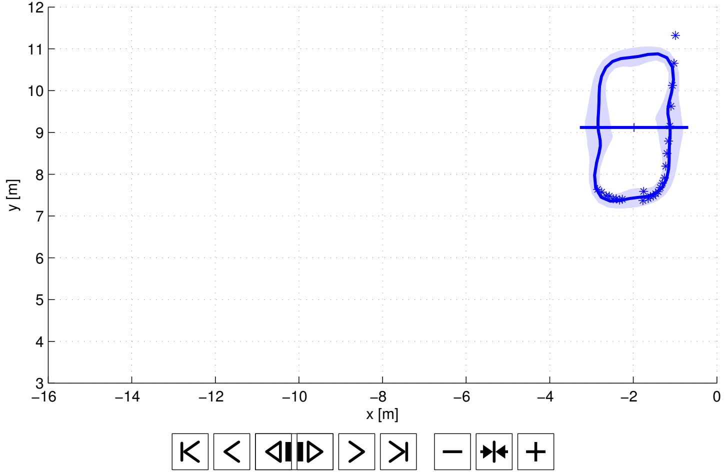


Green: RHM (Baum and Hanebeck). Black: Elliptical target (Koch...), Blue: proposed model



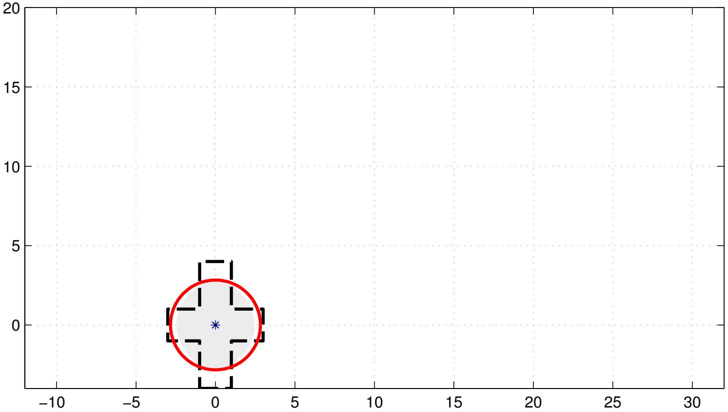
Extension - Symmetry assumptions

If we assume that $f(\theta)$ has a period of π instead of 2π, we can encode symmetry assumptions.



If the measurements originate from the target interior, we can add a random scalar to compensate for that

$$\mathbf{y}_{k,l} = \mathbf{x}_k^c + s_{k,l} H(\mathbf{x}_k^c, \psi_k)\mathbf{x}_k^{\mathbf{f}} + \mathbf{e}_{k,l}, \quad s_{k,l} \in [0, 1]$$



Conclusions and Future work

Conclusions

- Model the target extent with a Gaussian process
- Estimate target extent and kinematic state jointly
- Fully recursive update provided

Future work

- Use Rao-Blackwellized PF - target extent state can be marginalized
- Exploit target symmetry properties even further
- A more sophisticated multi-target tracker

LINKÖPINGS UNIVERSITET

LINKÖPINGS UNIVERSITET