# Point Mass Filter

## Sensor Fusion

**Fredrik Gustafsson**
fredrik.gustafsson@liu.se

Gustaf Hendeby
gustaf.hendeby@liu.se

Linköping University

# Purpose

To introduce grid based methods for filtering (and estimation)

- The Bayesian optimal filter revisited
- The key idea: Gridding the state space
- Numerical examples

# Bayes Optimal Filter: summary

General nonlinear state-space model:

$$x_{k+1} = f(x_k, u_k, v_k) \qquad x_k | x_{k-1} \sim p(x_k | x_{k-1})$$
$$y_k = h(x_k, u_k, e_k) \qquad y_k | x_k \sim p(y_k | x_k)$$

General Bayesian recursion (time and measurement updates)

$$p(x_{k+1} | y_{1:k}) = \int p(x_{k+1} | x_k) p(x_k | y_{1:k})\, dx_k,$$
$$p(x_k | y_{1:k}) = \frac{p(y_k | x_k) p(x_k | y_{1:k-1})}{p(y_k | y_{1:k-1})}.$$

- Analytic solution available in a few special cases (KF)
- Key idea: for a given trajectory $x_{1:k}$, the recursion can be computed.
- PMF: evaluate trajectories on a gridded state space

# Numerical Approximation

**Basic idea:** postulate a discrete approximation of the posterior. For the predictive density, we have

$$\hat{p}(x_k|y_{1:k-1}) = \sum_{i=1}^{N} w_{k|k-1}^{(i)} \delta(x_k - x_k^{(i)}).$$

The first moments (mean and covariance) are simple to compute from this approximation:

$$\hat{x}_{k|k-1} = \mathrm{E}(x_k) = \sum_{i=1}^{N} w_{k|k-1}^{(i)} x_k^{(i)},$$

$$P_{k|k-1} = \mathrm{Cov}(x_k) = \sum_{i=1}^{N} w_{k|k-1}^{(i)} (x_k^{(i)} - \hat{x}_{k|k-1})(x_k^{(i)} - \hat{x}_{k|k-1})^T.$$

Also, the MAP estimate can be useful:

$$\hat{x}_{k|k-1}^{\text{map}} = \arg\max_{x_k^{(i)}} \hat{p}(x_k|y_{1:k-1}).$$

# Measurement Update

The measurement update follows directly, without any extra approximations

$$\hat{p}(x_k|y_{1:k}) = \sum_{i=1}^{N} \underbrace{\frac{1}{c_k} p(y_k|x_k^{(i)}) w_{k|k-1}^{(i)}}_{w_{k|k}^{(i)}} \delta(x_k - x_k^{(i)})$$

$$c_k = \sum_{i=1}^{N} p(y_k|x_k^{(i)}) w_{k|k-1}^{(i)}.$$

The normalization constant $c_k$ corresponds to assuring that $\sum_{i=1}^{N} w_{k|k}^{(i)} = 1$.

# Time Update

Bayesian time update gives a continuous distribution

$$\hat{p}(x_{k+1}|y_{1:k}) = \sum_{i=1}^{N} w_{k|k}^{(i)} p(x_{k+1}|x_k^{(i)}).$$

To keep the approximation form, the distribution is sampled at points $x_{k+1}^{(i)}$, and the weights are updated as

$$w_{k+1|k}^{(i)} = \hat{p}(x_{k+1}^{(i)}|y_{1:k}) = \sum_{j=1}^{N} w_{k|k}^{(j)} p(x_{k+1}^{(i)}|x_k^{(j)}), \quad i = 1, 2, \ldots, N.$$

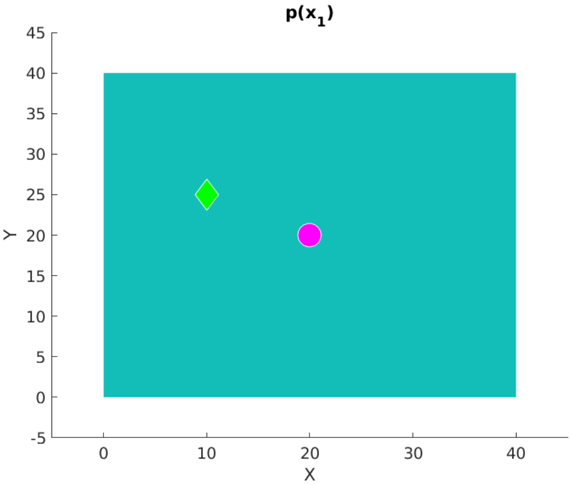
**Two principles:**

- Keep the same grid, so $x_{k+1}^{(i)} = x_k^{(i)}$, which yields the point mass filter.
- Generate new samples from the posterior distribution $x_{k+1}^{(i)} \sim \hat{p}(x_{k+1}|y_{1:k})$, which yields the marginal particle filter.

Both alternatives have quadratic complexity ($N$ weights $w_{k+1|k}^{(i)}$, each one involving a sum with $N$ terms).

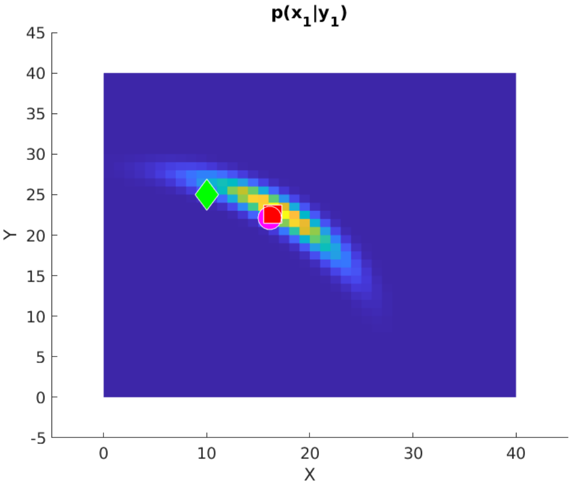
# PMF Numerical Example: 1 radar

- Range bearing measurements
- CP motion model
- $R = diag(1, .3)^2$
- $Q = diag(5, 5)$
- magenta: estimate
- green ground truth
- red measurement



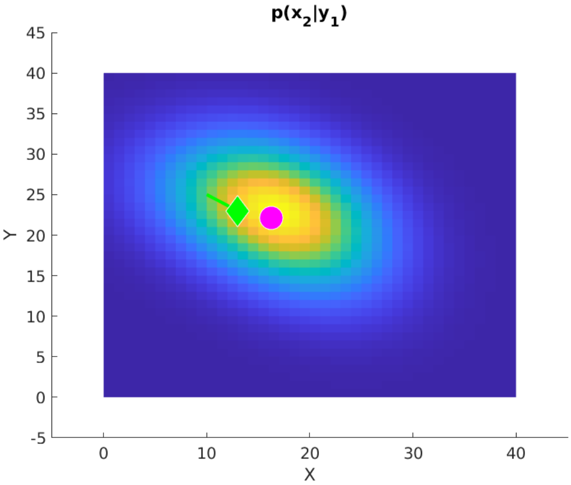
# PMF Numerical Example: 1 radar

- Range bearing measurements
- CP motion model
- $R = diag(1, .3)^2$
- $Q = diag(5, 5)$
- magenta: estimate
- green ground truth
- red measurement



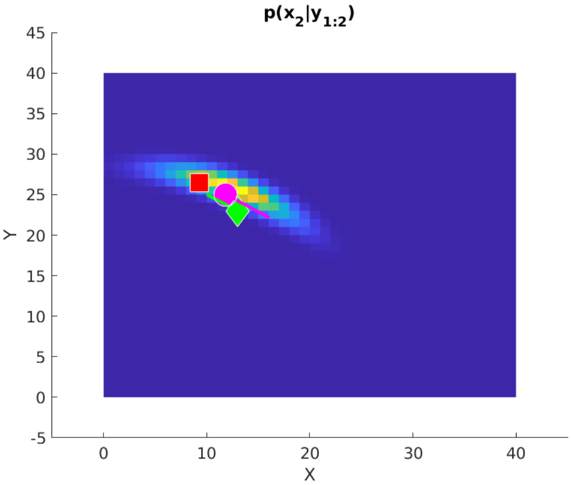
# PMF Numerical Example: 1 radar

- Range bearing measurements
- CP motion model
- $R = diag(1, .3)^2$
- $Q = diag(5, 5)$
- magenta: estimate
- green ground truth
- red measurement



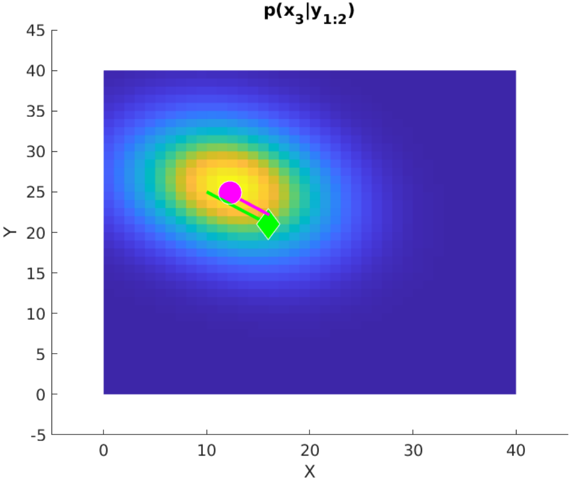
# PMF Numerical Example: 1 radar

- Range bearing measurements
- CP motion model
- $R = diag(1, .3)^2$
- $Q = diag(5, 5)$
- magenta: estimate
- green ground truth
- red measurement



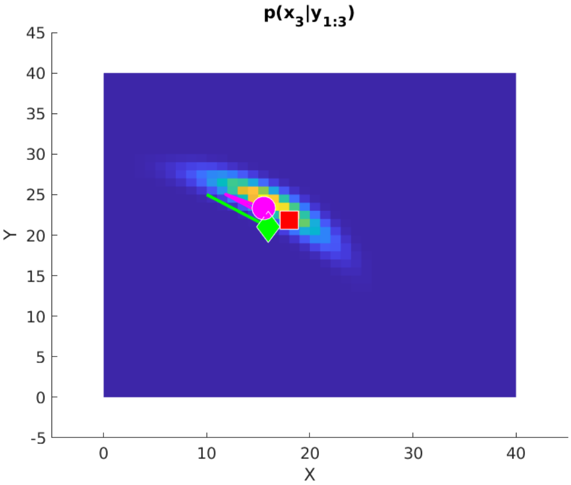
# PMF Numerical Example: 1 radar

- Range bearing measurements
- CP motion model
- $R = diag(1, .3)^2$
- $Q = diag(5, 5)$
- magenta: estimate
- green ground truth
- red measurement



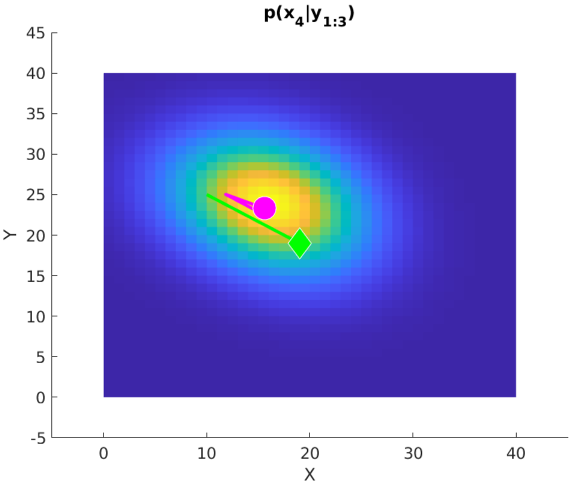
# PMF Numerical Example: 1 radar

- Range bearing measurements
- CP motion model
- $R = diag(1, .3)^2$
- $Q = diag(5, 5)$
- magenta: estimate
- green ground truth
- red measurement



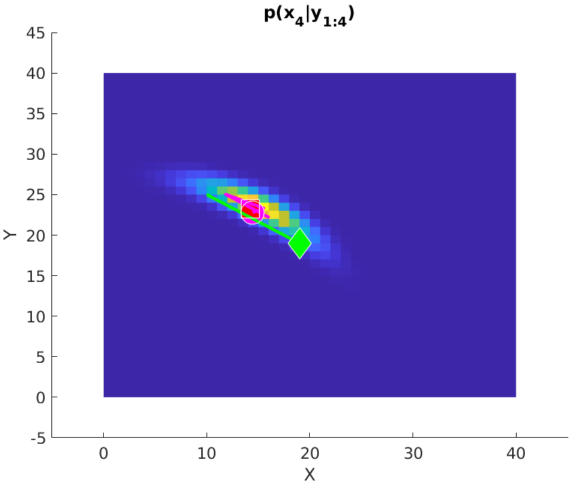
# PMF Numerical Example: 1 radar

- Range bearing measurements
- CP motion model
- $R = diag(1, .3)^2$
- $Q = diag(5, 5)$
- magenta: estimate
- green ground truth
- red measurement



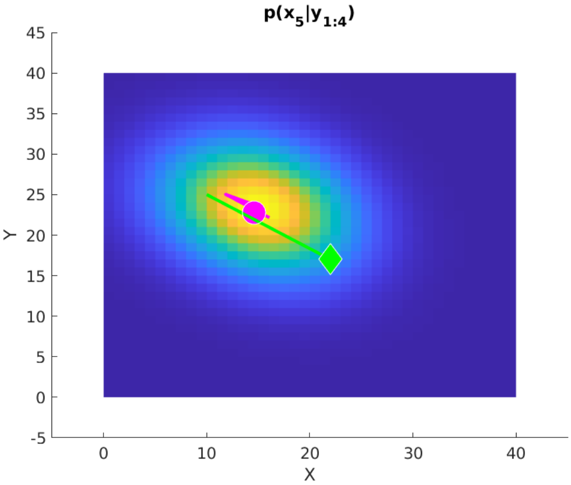
# PMF Numerical Example: 1 radar

- Range bearing measurements
- CP motion model
- $R = diag(1, .3)^2$
- $Q = diag(5, 5)$
- magenta: estimate
- green ground truth
- red measurement



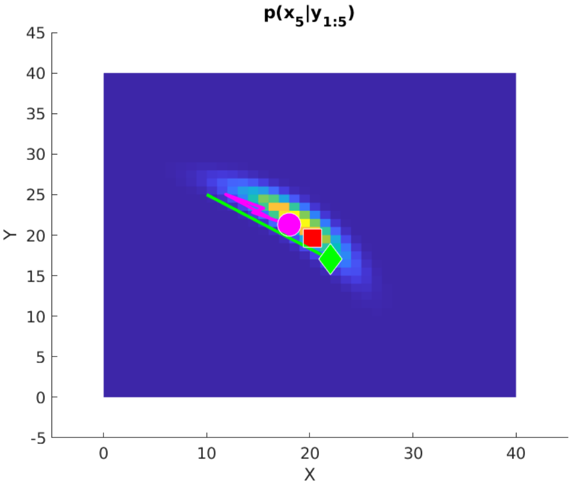
# PMF Numerical Example: 1 radar

- Range bearing measurements
- CP motion model
- $R = diag(1, .3)^2$
- $Q = diag(5, 5)$
- magenta: estimate
- green ground truth
- red measurement



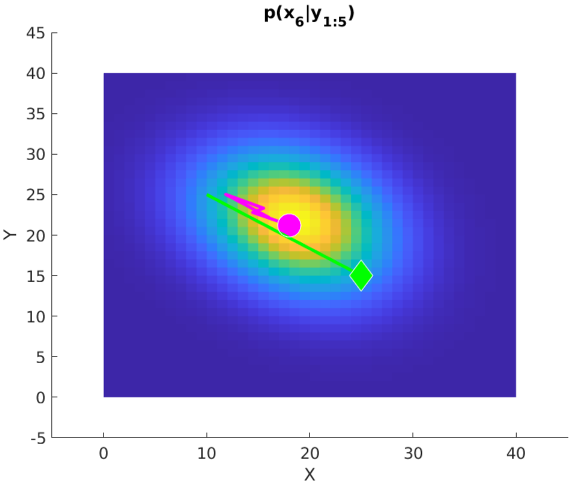
# PMF Numerical Example: 1 radar

- Range bearing measurements
- CP motion model
- $R = diag(1, .3)^2$
- $Q = diag(5, 5)$
- magenta: estimate
- green ground truth
- red measurement



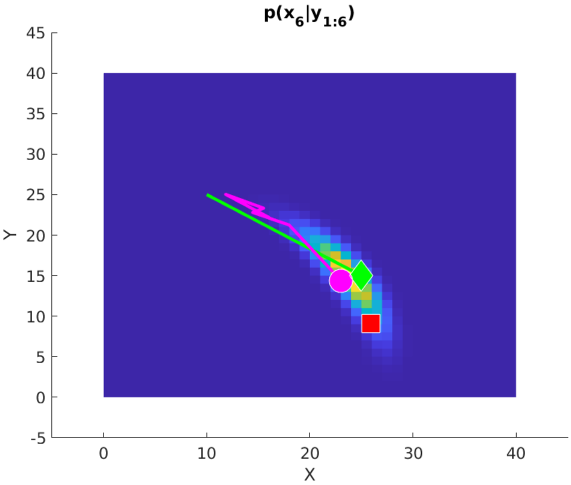
# PMF Numerical Example: 1 radar

- Range bearing measurements
- CP motion model
- $R = diag(1, .3)^2$
- $Q = diag(5, 5)$
- magenta: estimate
- green ground truth
- red measurement



# PMF Numerical Example: 1 radar

- Range bearing measurements
- CP motion model
- $R = diag(1, .3)^2$
- $Q = diag(5, 5)$
- magenta: estimate
- green ground truth
- red measurement



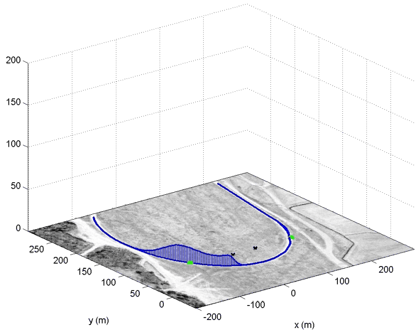
# PMF Numerical Example: 1 radar

Remarks:

- Measurement update works as a numerical NLS solver.
- For a 2D state vector, the 1600 weights are quickly updated on this 40x40 grid.
- After a while, the grid needs to be redefined to track the target. Adapting grid is one challenge with PMF.
- There is no velocity state, so time update is just a diffusion (increase uncertainty in all directions)
- With a velocity state with 40x40 more states, the number of grid points would be $40^4 = 2.56$ million. Still feasible, but complexity increases fast with state dimension.
- PF mitigates this exponential growth in complexity somewhat and includes an adaptive grid.

# PMF Application: 2 DOA sensors, 2 targets

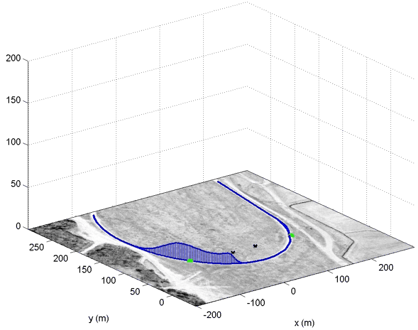
- Two microphone arrays (black x) compute two DOA's.
- Two road-bound targets (green *).
- One grid point (stem plot) every meter on the road.
- No motion model, only one state for position.
- Data from FOI-LiU collaboration




http://youtu.be/VcdTebC9uTs

# PMF Application: 2 DOA sensors, 2 targets

- Two microphone arrays (black x) compute two DOA's.
- Two road-bound targets (green *).
- One grid point (stem plot) every meter on the road.
- No motion model, only one state for position.
- Data from FOI-LiU collaboration




http://youtu.be/VcdTebC9uTs

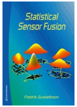
# Summary: Point-Mass Filter

## Advantages:

- Simple to implement.
- Works excellently when $n_x \leq 2$.
- Gives the complete posterior, not only $\hat{x}$ and $P$.
- Global search, no local minima.

## Problems:

- Grid inefficient in higher dimensions, since the probability to be at one grid point depends on the transition probability from all other grid points.
- The grid should be adaptive: (i) moving with object, (ii) rough initially, then finer.
- Quadratic complexity in number of grid points.




Section 9–9.2