Adaptive Kernel Kalman Filter

Mengwei Sun, Mike E. Davies,

Ian K. Proudler, James R. Hopgood

Novelties

- ❑ Explore the potential of kernel idea within full model - based Bayesian filters
- ❑ Introduce the adaptive kernel Kalman filter (AKKF).
  - Model - based Bayesian filter
  - DSM information are used to calculate the update kernel rules <u>in an **adaptive** manner</u>
  - Prediction and posterior distributions are embedded into a kernel feature space
  - Avoid the problematic resampling in most PFs
  - Reduce the sample complexity

Outline

- ❑**Background**
- ❑Preliminaries
- ❑Adaptive Kernel Kalman Filter
- ❑Simulation Results
- ❑Conclusions

Background – Non-linear/non-Gaussian estimation

- Dynamic state-space model (DSM)
  - Transition model: $\mathbf{x}_n = f(\mathbf{x}_{n-1}, \mathbf{u}_n)$
  - Measurement model: $\mathbf{y}_n = h(\mathbf{x}_n, \mathbf{v}_n)$
- Sequential Bayesian rule
  - Prediction: $p(\mathbf{x}_n|\mathbf{y}_{1:n}) = \int p(\boldsymbol{x}_n|\mathbf{x}_{n-1})\, p(\mathbf{x}_{n-1}|\mathbf{y}_{1:n-1}) d\mathbf{x}_{n-1}$
  - Update: $p(\mathbf{x}_n|\mathbf{y}_{1:n}) = \frac{p(\mathbf{y}_n|\mathbf{x}_n) p(\mathbf{x}_n|\mathbf{y}_{1:n-1})}{p(\boldsymbol{y}_n|\mathbf{y}_{1:n-1})}$
- Two families of sequential Bayesian filters
  - Model-driven filters: DSM is given explicitly,

    e.g., Kalman Filter (KF), Unscented Kalman Filter (UKF), Particle Filter (PF)
  - Data-driven filters: DSM is unknown or partially known while the training data set is provided

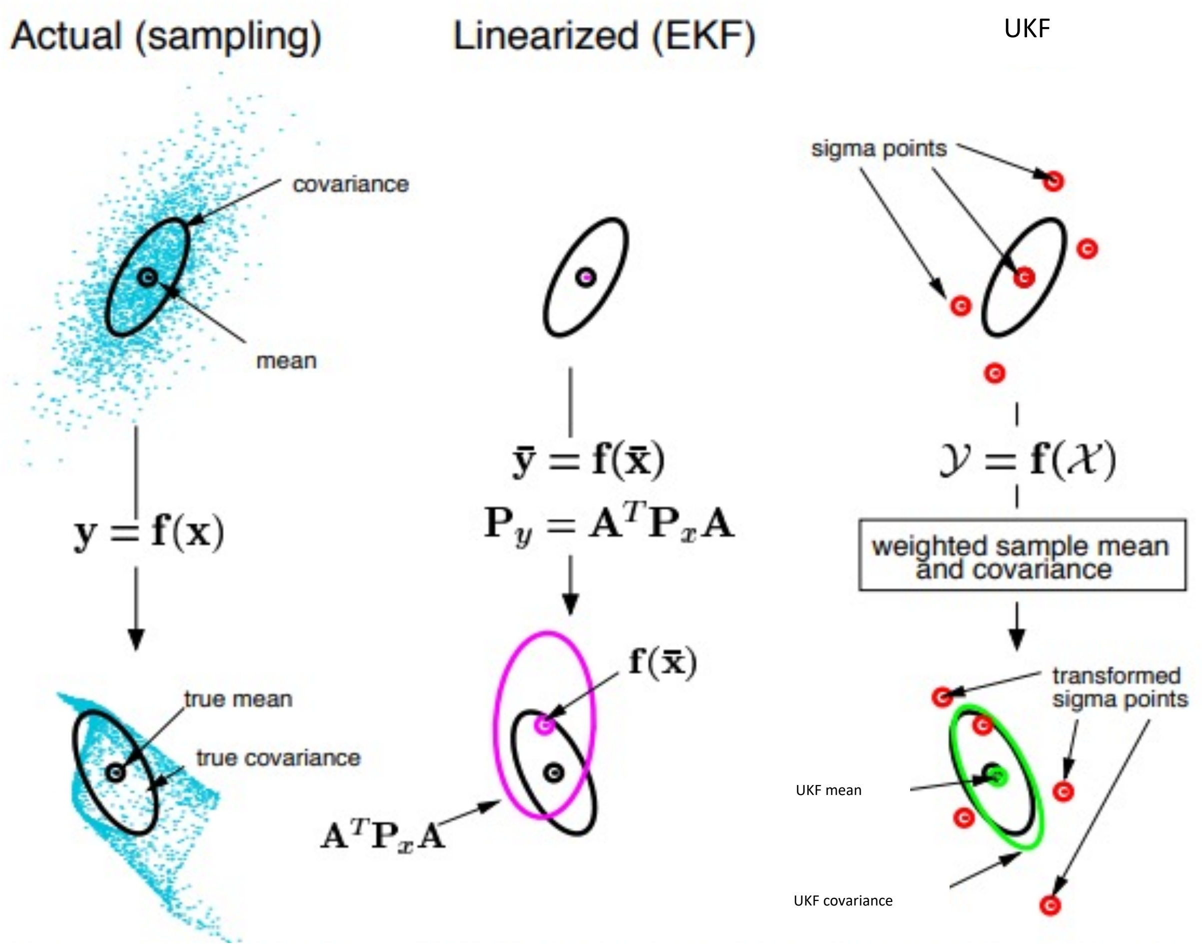
Background – Model-driven filters:

- ❑Kalman filter (KF):

  Optimal Bayesian solution for linear DSMs

- ❑Nonlinear systems

  Extended KF (EKF) & unscented KF (UKF).



Background – Model-driven filters:

- ❑Kalman filter (KF):

  Optimal Bayesian solution for linear DSMs

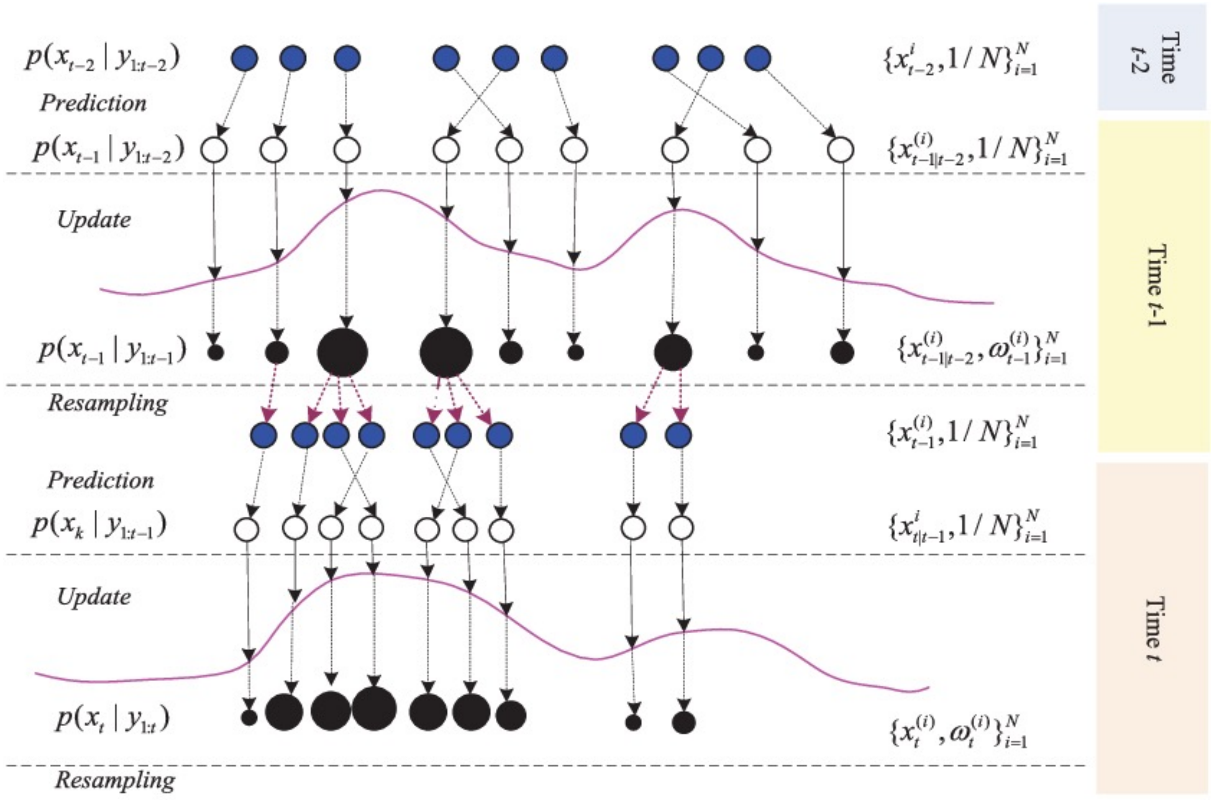
- ❑Nonlinear systems

  Extended KF (EKF) & unscented KF (UKF).

- ❑Bootstrap particle filter (PF)

  Resampling is a necessary step

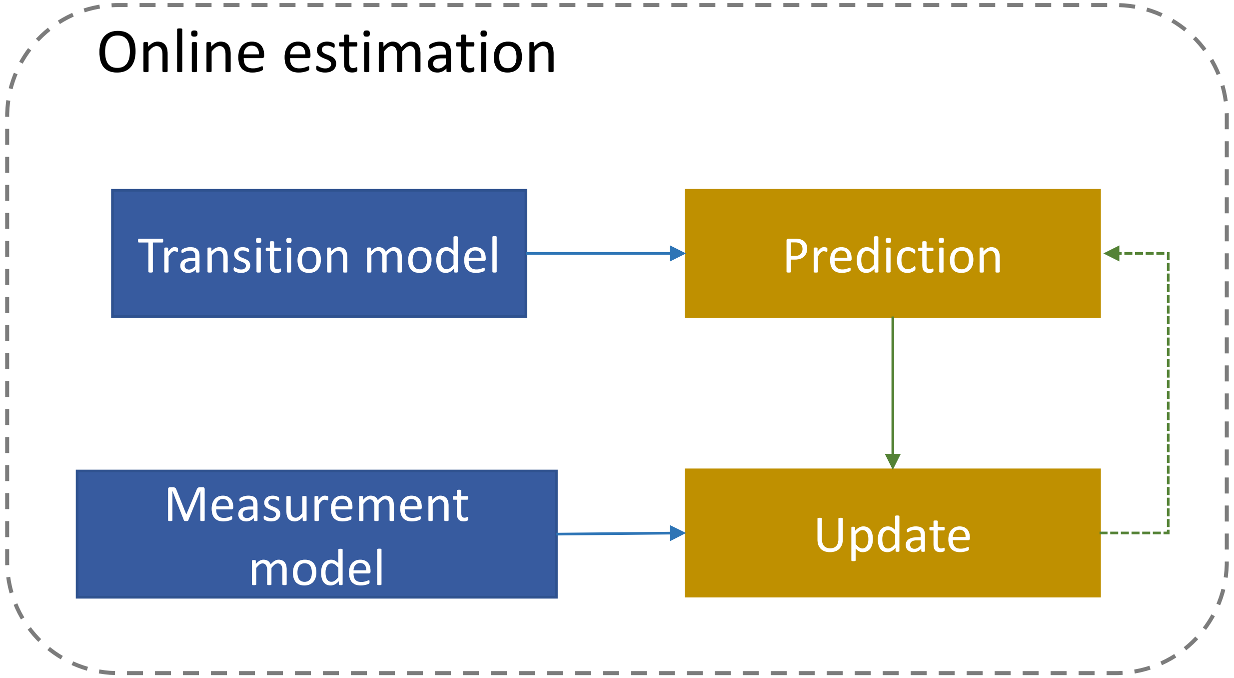
  - Increase complexity
  - Hard to parallelize



Background – Data-driven filters:

- DSMs are unknown or partially known, need to be inferred from prior training data

- **Existing methods**
  - Training data
  - Off-line training to learn the unknown transition/measurement models



- **The performance limits**
  - Difficult to incorporate theoretical DSM models
  - Problems occur if target moves outside space defined by training data

Preliminaries – Kernel mean embedding (KME)

- Reproducing kernel Hilbert space (RKHS):
  - High dimensional kernel feature space, finite/infinite space
- State point x is mapped into RKHS through a non-linear feature mapping $\phi(x)$
- The kernel embedding approach represents a probability distribution by an element in the RKHS

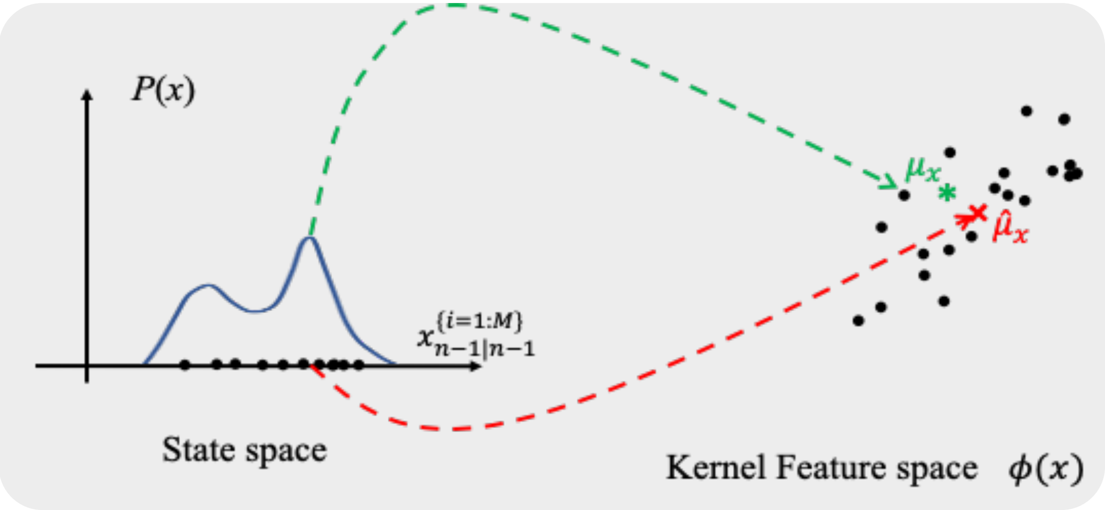
$$\mu_X \overset{\text{def}}{=} \mathbb{E}_X[\phi(X)] = \int \phi(x)dP(x)$$

- Empirical kernel estimator, given a sample set

$$\mu_X = \sum_{i=1}^{M} w_i\phi(x_i) = \Phi\mathbf{w}$$

  - If x_i are drawn from $P(x), w_i = 1/M$.

In general, kernel weights are non-uniform, positive/negative, c.f. UKF



Preliminaries – Kernel mean embedding (KME)

- The KME approach represents a conditional distribution $P(X|y)$ by an element in the RKHS

$$\mu_{X|y} := \mathbb{E}_{X|y}\left[\phi_x(X)\right] = \int_{\mathcal{X}} \phi_x(x) dP(x|y).$$

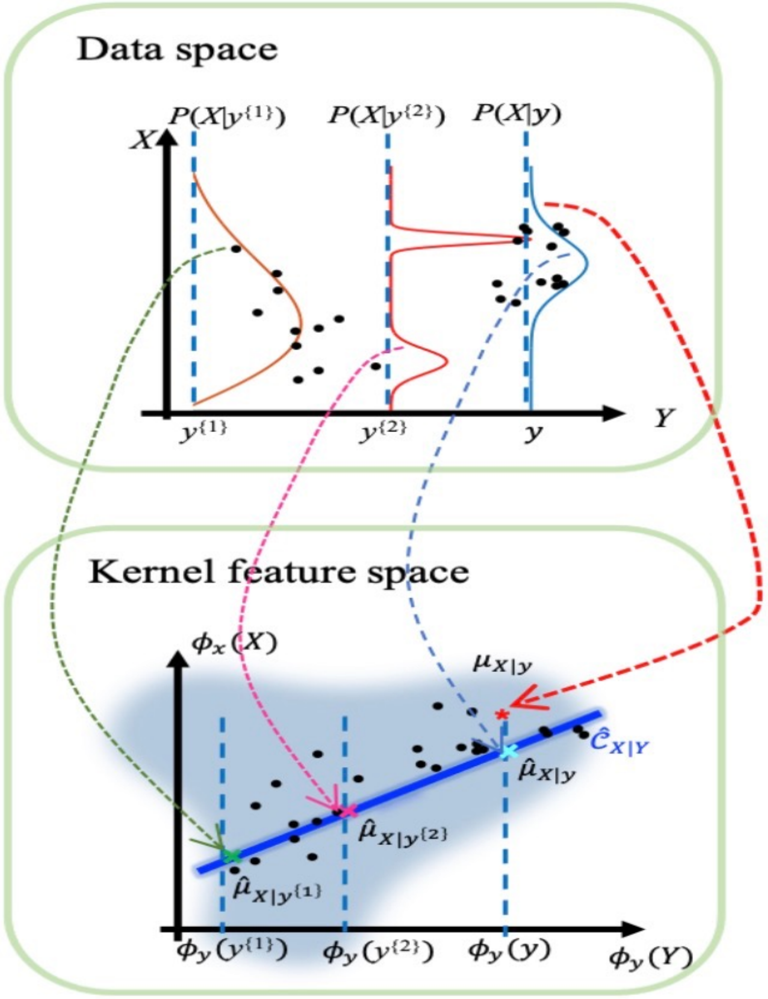
- By defining the conditional operator $\mathcal{C}_{X|Y}$ as the linear operator

$$\mu_{X|y} = \mathcal{C}_{X|Y}\phi_y(y) = \mathcal{C}_{XY}\left(\mathcal{C}_{YY} + \lambda I\right)^{-1}\phi_y(y).$$

- Empirical kernel estimator: The estimate of the $\mathcal{C}_{X|Y}$ is obtained as a linear regression in the RKHS

$$\hat{\mu}_{X|y} = \hat{\mathcal{C}}_{X|Y}\phi_y(y) = \Phi\left(G_{YY} + \lambda I\right)^{-1}\Upsilon^{\mathrm{T}}\phi_y(y) = \Phi\mathbf{w},$$
$$\mathbf{w} = \left(G_{YY} + \lambda I\right)^{-1} G_{:,y}.$$

Non-uniform weights, positive/negative, different from PFs

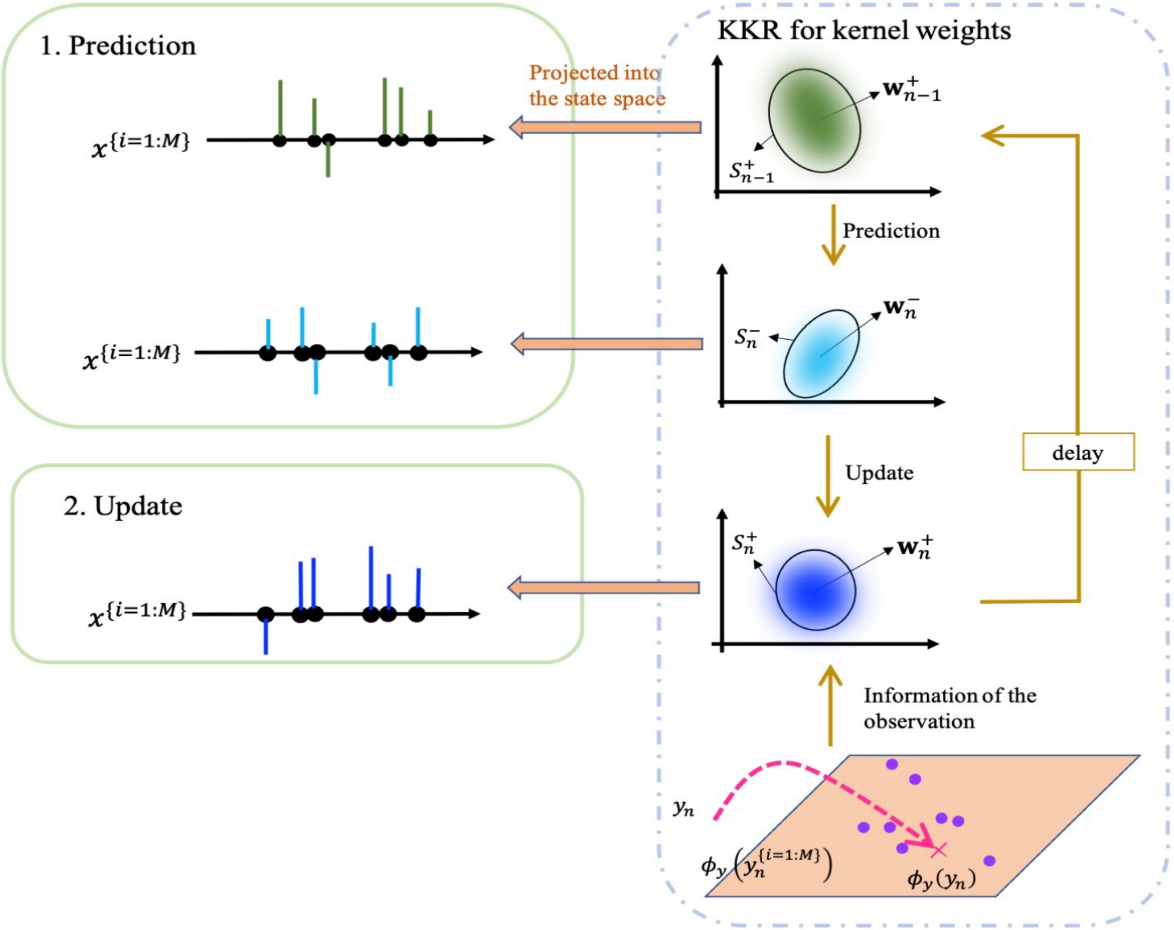


Preliminaries – Kernel Kalman filter (KKF)

- Non-linear estimation in data space

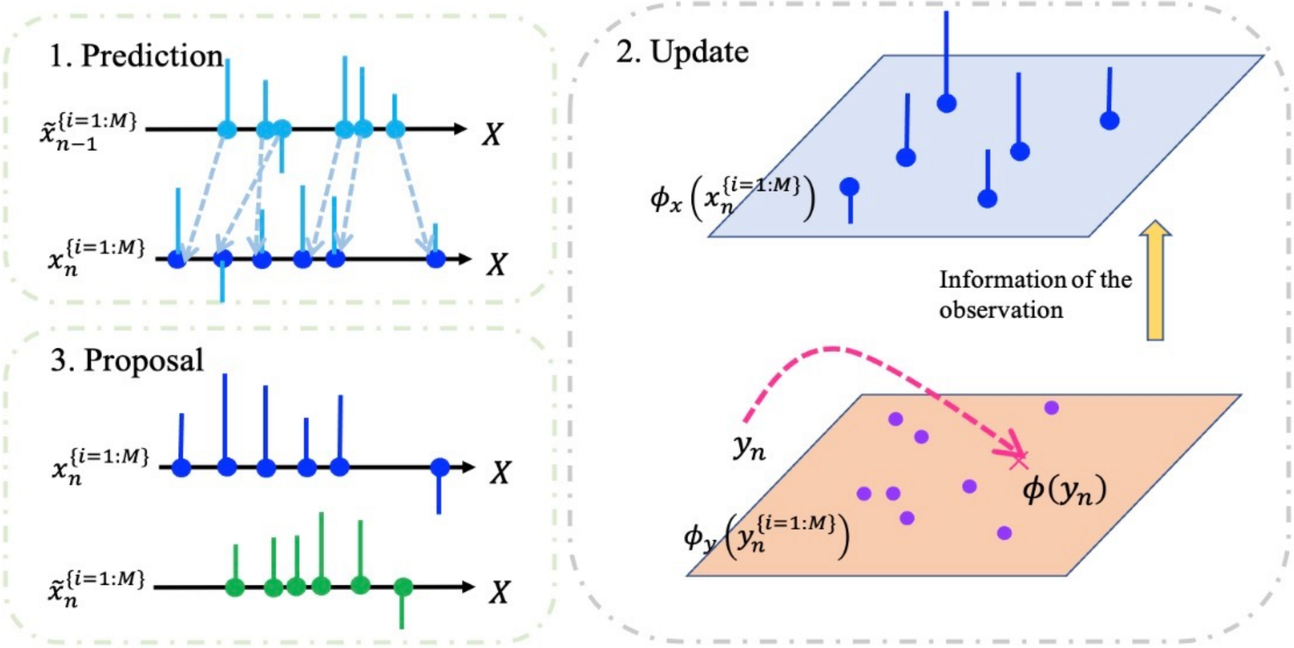
  –> Linear way in kernel feature space

- Execute conventional KF in kernel feature space
- Predict and update the kernel weight mean and covariance
- Relying on the training data set



Adaptive Kernel Kalman Filter (AKKF)

- Executed in both the data state space and kernel feature space
  - The particles are propagated and updated in the data space based on the DSM (similar to UKF & PF)
  - Kernel weight mean and covariance are predicted and updated in the kernel feature space (similar to KKF way)
- Three main steps: prediction, update, proposal



Adaptive Kernel Kalman Filter (AKKF)

❑ Embedding the Posterior Distribution at time *n*-1

$$\hat{\mu}^{+}_{x_{n-1}} = \Phi_{n-1}\mathbf{w}^{+}_{n-1},$$
$$\hat{\mathcal{C}}^{+}_{x_{n-1}x_{n-1}} = \Phi_{n-1}S^{+}_{n-1}\Phi^{\mathrm{T}}_{n-1}.$$

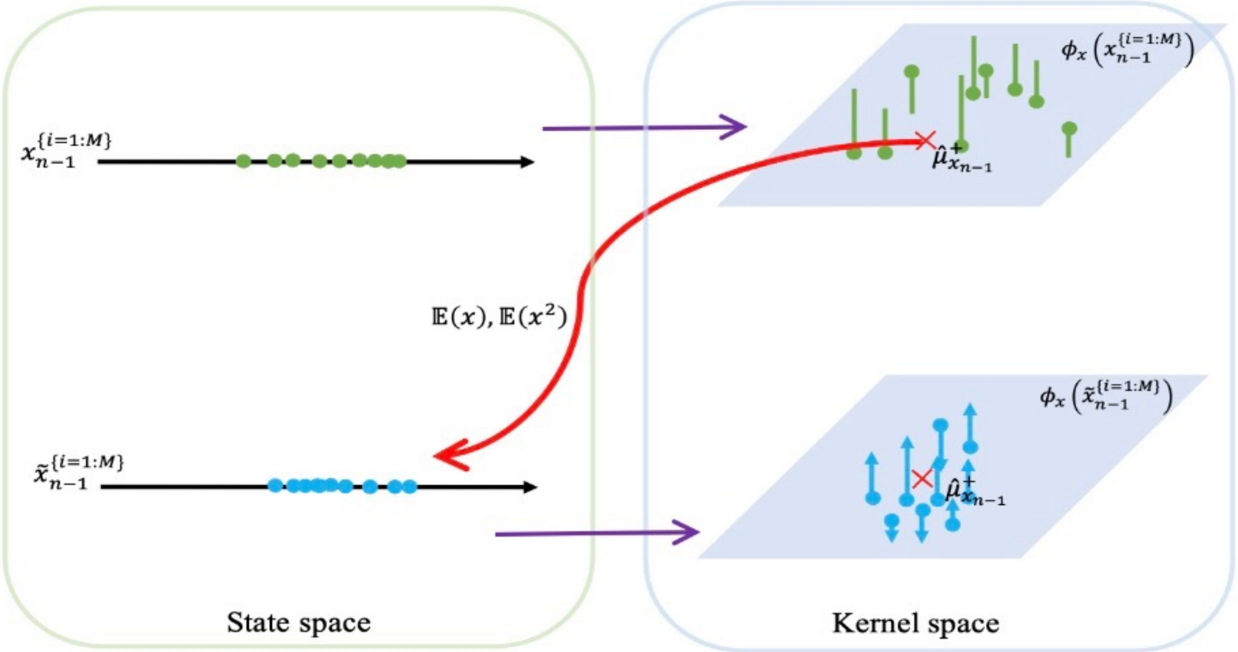
- Generated proposal particles to capture the diversity of the non-linearity (c.f. sigma points generation)

$$\tilde{x}^{\{i=1:M\}}_{n-1} \sim \mathcal{N}\left(\mathbb{E}\left(x_{n-1}\right), \mathrm{Var}\left(x_{n-1}\right)\right),$$

- For convenience, draw from Gaussian distribution

Note, due to weighting, this is **not** a Gaussian approximation

Instead, adaptive change of kernel spaces



Adaptive Kernel Kalman Filter (AKKF)

❑ Prediction from Time *n*−1 to Time *n*

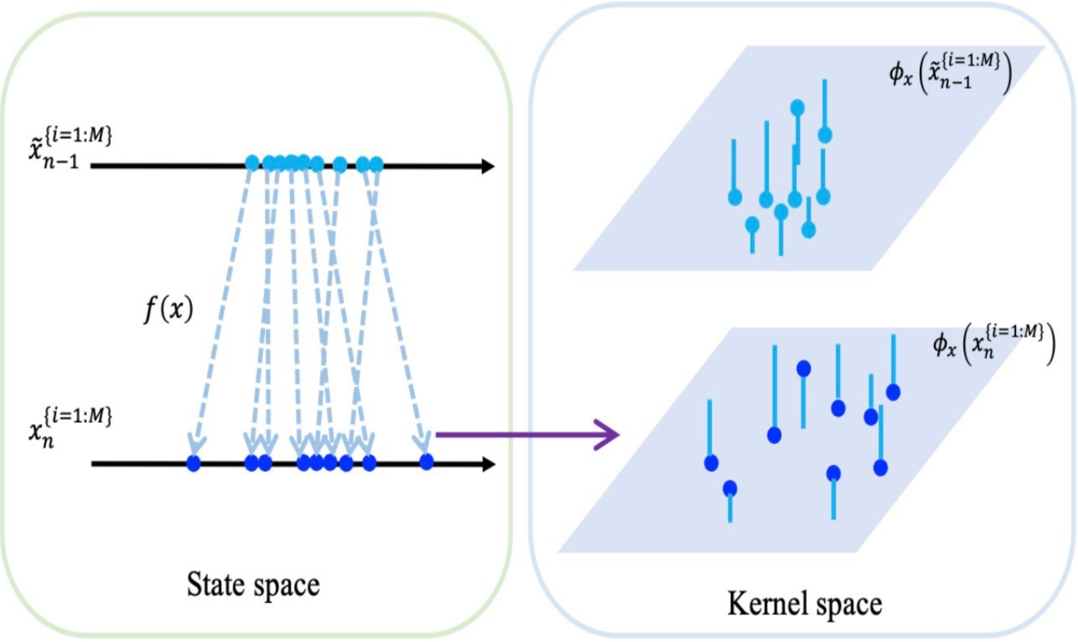
(predict step of KF)

- Predictive particles: propagate proposal particles through the transition function
- New Kernel space Φ_n
- Empirical Predictive KME by calculating conditional operator

$$p(x_n|x_{n-1}) \mapsto \hat{\mu}_{x_n}^- = \Phi_n \mathbf{w}_n^- = \hat{C}_{x_n|x_{n-1}} \hat{\mu}_{x_{n-1}}^+,$$

- Predictive kernel weight mean and covariance

$$\mathbf{w}_n^- = (K_{\tilde{x}\tilde{x}} + \lambda_{\tilde{K}} I)^{-1} K_{\tilde{x}x} \mathbf{w}_{n-1}^+ = \Gamma_{n-1} \mathbf{w}_{n-1}^+,$$
$$S_n^- = \tilde{S}_{n-1}^+ + V_n.$$



Adaptive Kernel Kalman Filter (AKKF)

❑Update at Time *n (correct step of KF)*

- Observation particles

$$y_n^{\{i\}} = g(x_n^{\{i\}}, v_n^{\{i\}}),$$

- Kernel Kalman gain calculation

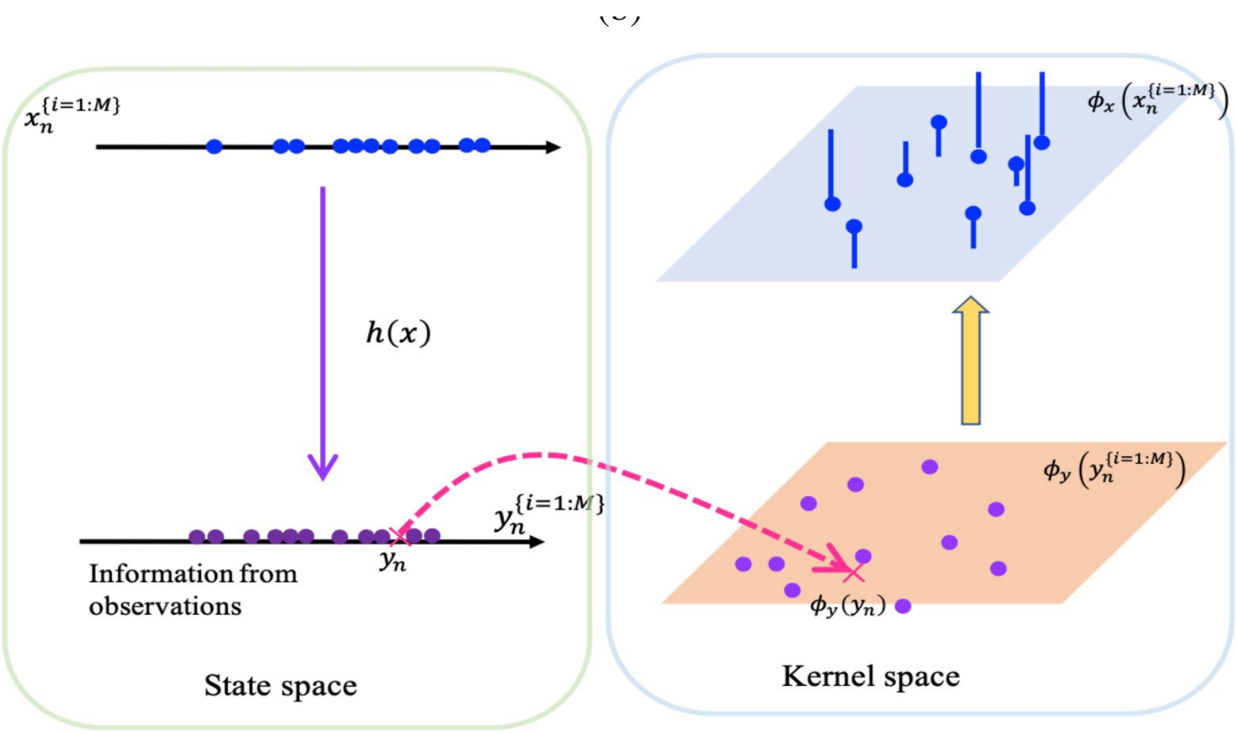
$$\hat{\mu}_{x_n}^{+} = \hat{\mu}_{x_n}^{-} + Q_n \left[\phi_y(y_n) - \hat{C}_{y_n|x_n}\hat{\mu}_{x_n}^{-}\right],$$

$$\hat{C}_{x_n x_n}^{+} = \mathrm{cov}(\phi_x(x_n) - \hat{\mu}_{x_n}^{+}).$$

- Update kernel weight mean and covariance

$$\mathbf{w}_n^{+} = \mathbf{w}_n^{-} + Q_n \left(G_{:,y_n} - G_{yy}\mathbf{w}_n^{-}\right),$$

$$S_n^{+} = S_n^{-} - Q_n G_{yy} S_n^{-}.$$

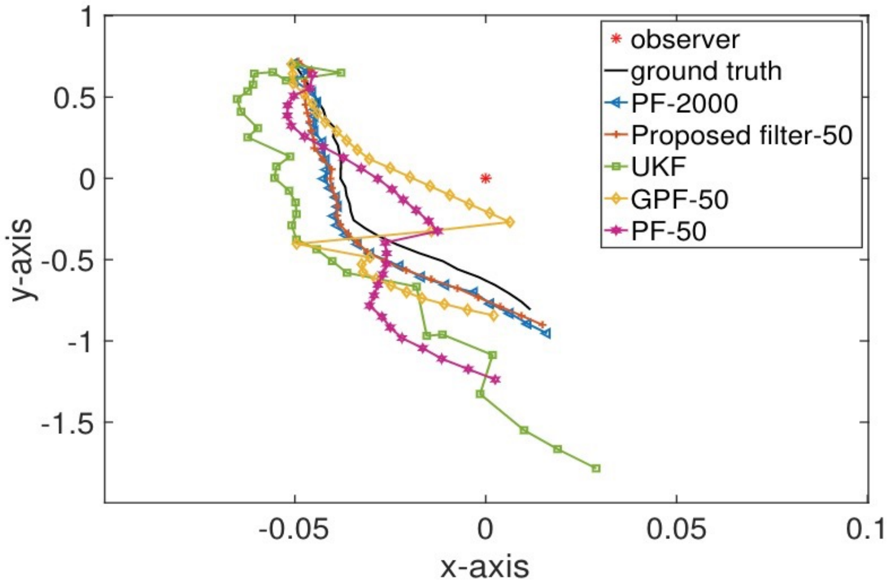


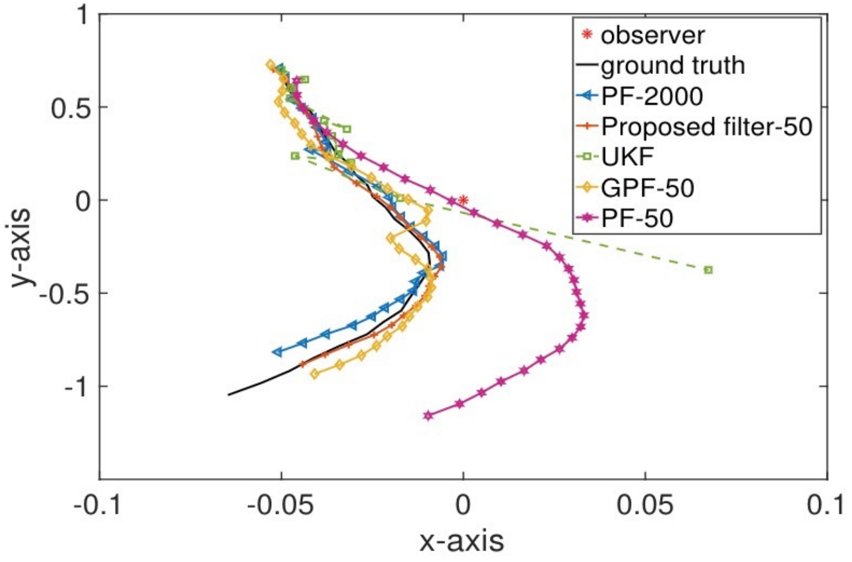
Simulation

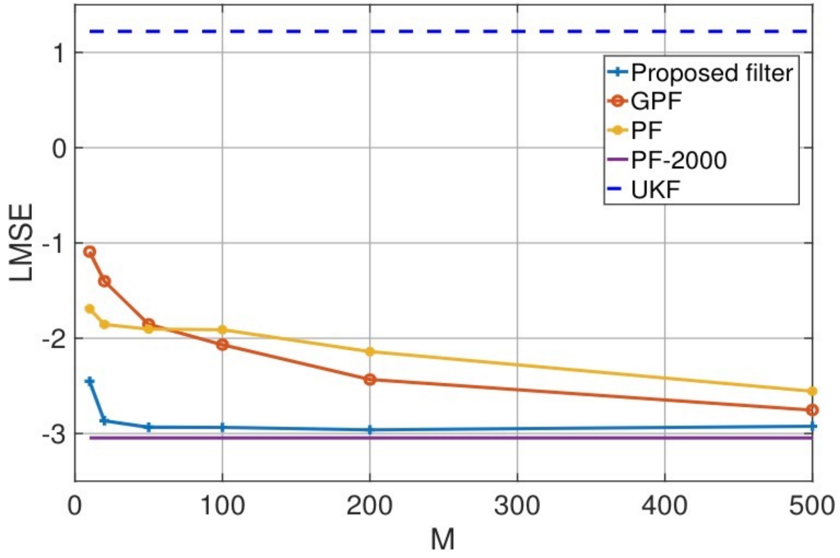
- Bearing–only tracking (BOT)

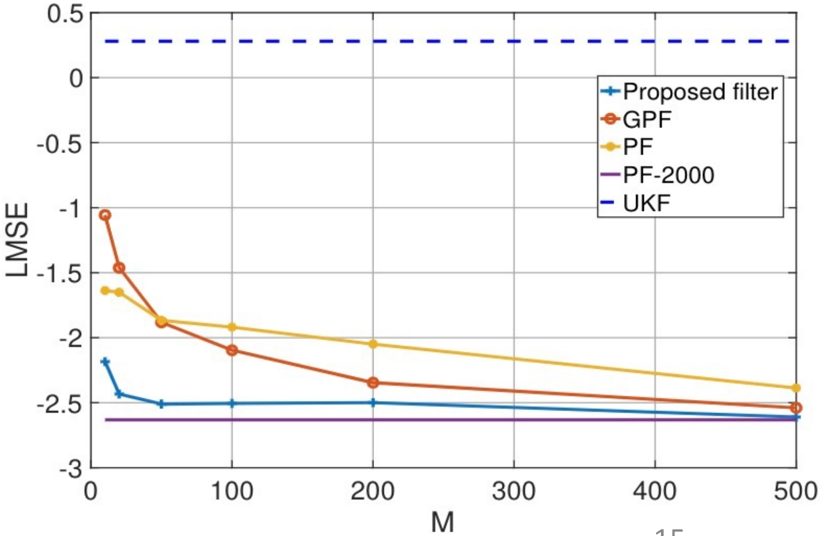
$$\mathbf{x}_n = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x}_{n-1} + \begin{bmatrix} 0.5 & 0 \\ 1 & 0 \\ 0 & 0.5 \\ 0 & 1 \end{bmatrix} \mathbf{u}_n,$$

$$y_n = \tan^{-1}\left(\frac{\eta_n}{\xi_n}\right) + v_n.$$









Conclusion

❑Summary

- Kernel mean embedding: Solve Non-linear estimation in high dimensional kernel space using linear ways
- AKKF: apply KF into kernel spaces with adaptively updated particles & kernel spaces

❑Advantages

- Nonlinear, non-Gaussian filter for Bayesian tracking
  - Incorporation of theoretical models
- Lower computation complexity
  - Remove resample
  - Smaller particle number requirement

Thank You
For Your Attention

msun@ed.ac.uk