# Kernel based estimation & tracking techniques for distributed and modular sensor networks

Mengwei Sun[1], James R. Hopgood[1], Mike E. Davies[1], Ian K. Proudler[2]

[1]University of Edinburgh

[2]University of Strathclyde

# Outlines:

- ❑Background
- ❑Preliminaries
  - Kernel Mean Embedding
  - Kernel Kalman Filter
- ❑Adaptive Kernel Kalman Filter (AKKF)
- ❑AKKF applications
  - AKKF in Single-target Single-sensor Tracking
  - AKKF Multi-Sensor Fusion
  - AKKF Multi-target Tracking(MTT)
- ❑Conclusion

# Background – Non-linear/non-Gaussian estimation

- Dynamic state-space model (DSSM)
  - 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: $p(\mathbf{x}_n|\mathbf{y}_{1:n}) = \frac{p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{y}_{\mathbf{1}:n-1})}{p(\boldsymbol{y}_n|\mathbf{y}_{\mathbf{1}:n-1})}$
- 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

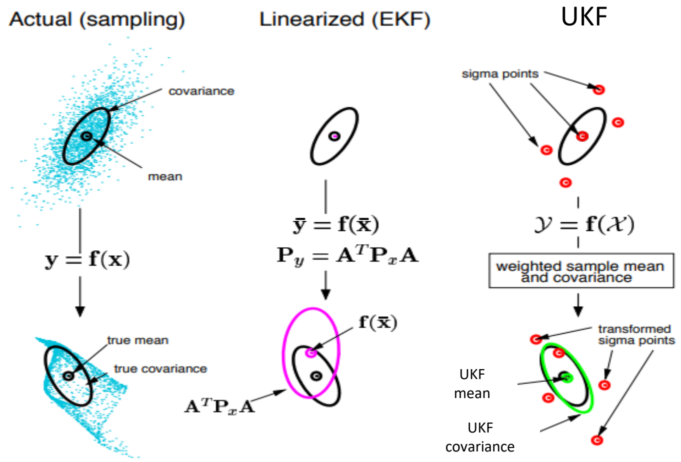
# Background – Model-driven filters:

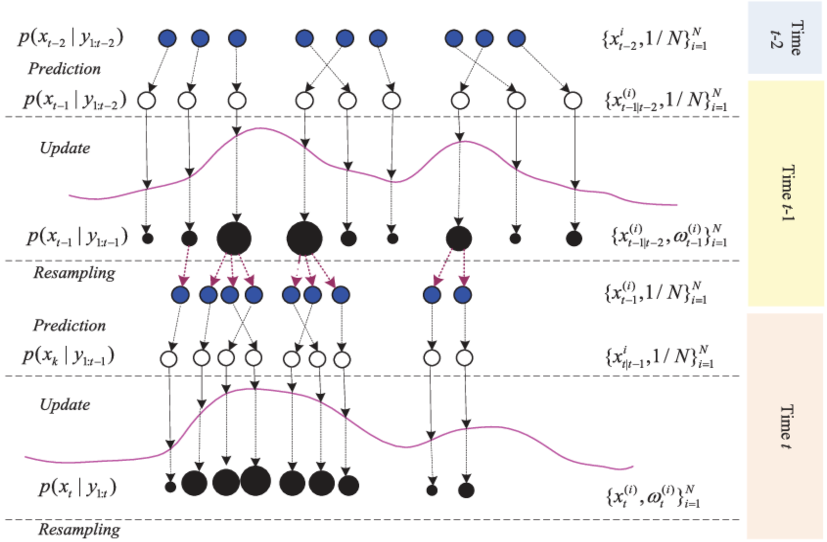
- ❑Kalman filter (KF):

  Optimal Bayesian solution for linear DSSMs
- ❑Nonlinear systems

  Extended KF (EKF) & unscented KF (UKF)
- ❑Bootstrap particle filter (PF) [1]

  **Resampling** is a necessary step, hard to parallelize



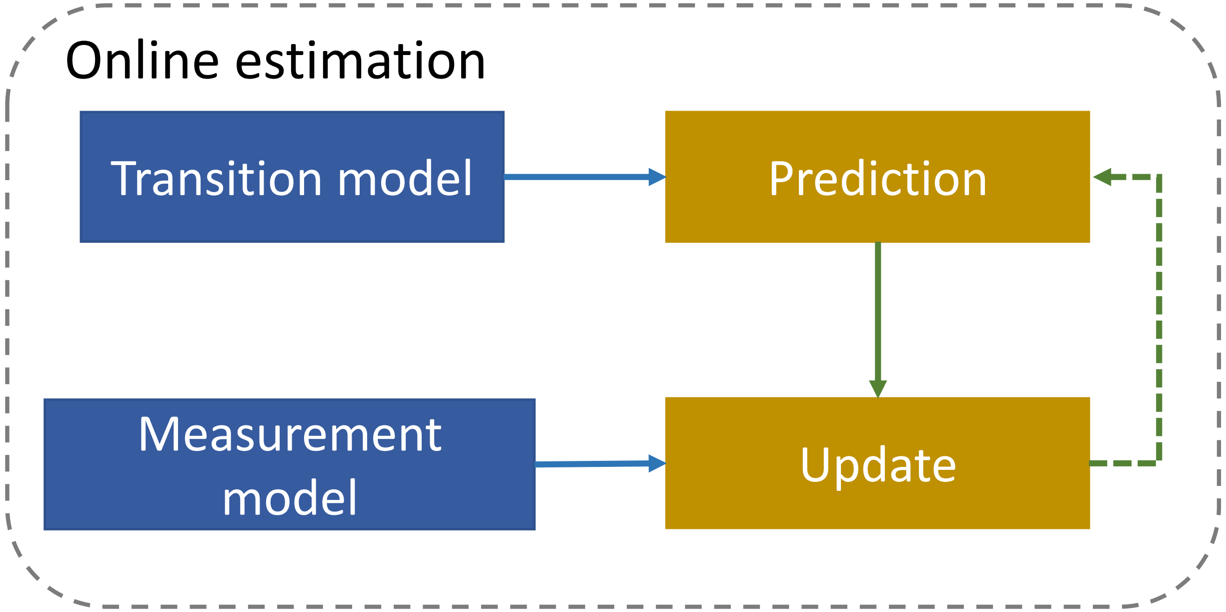

[1] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian Bayesian tracking," IEEE Trans. Signal Process, vol. 50, no. 2, pp. 174–188, 2002

# Background – Data-driven filters:

- DSSMs 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
  - On-line estimation

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




[1] M. W. Sun, M. E. Davies, I. Proudler, J.R. Hopgood, "A Gaussian Process based Method for Multiple Model Tracking," SSPD2020, published.
[2] M. W. Sun, M. E. Davies, I. Proudler, J.R. Hopgood, "Maneuvering Multi-target Tracking Based on Gaussian Process Regression," IEEE Transactions on Aerospace and Electronic, submitted

# Outlines:

- Background
- Preliminaries
    - Kernel Mean Embedding
    - Kernel Kalman Filter
- Adaptive Kernel Kalman Filter (AKKF)
- AKKF applications
    - AKKF in Single-target Single-sensor Tracking
    - AKKF Multi-Sensor Fusion
    - AKKF Multi-target Tracking(MTT)
- Conclusion

# Preliminaries – Kernel mean embedding (KME)

- State point $x$ is mapped into feature space through a non-linear feature mapping $\phi(x)$[1]

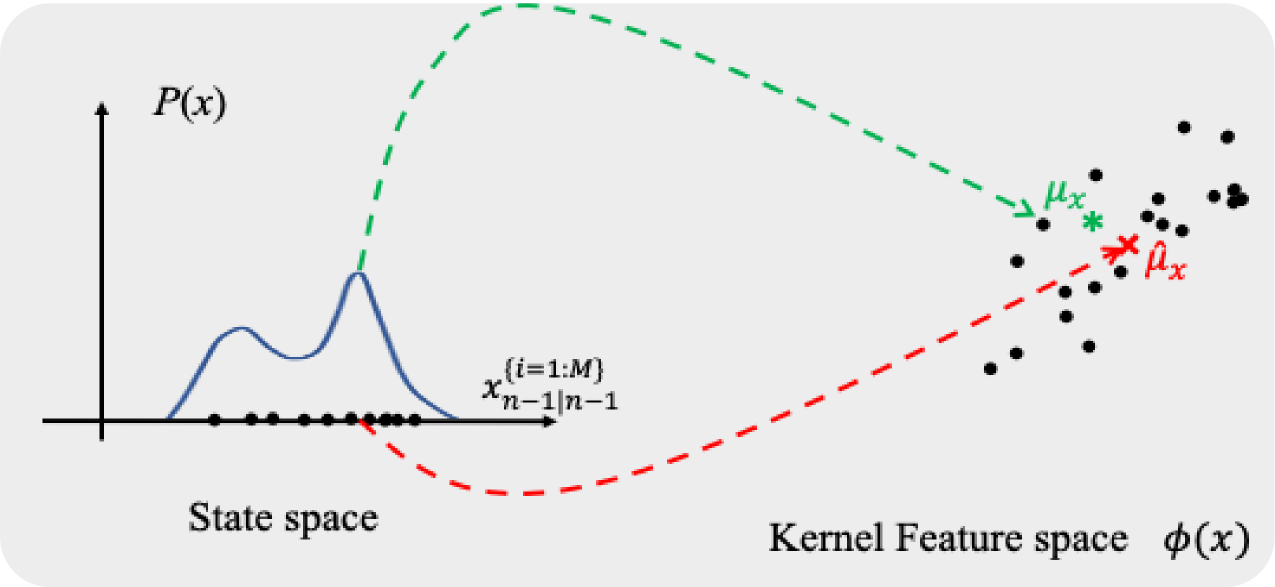
- The kernel embedding approach represents a probability distribution by an element in the feature space

$$\mu_X := \mathbb{E}_X[\phi_{\mathbf{x}}(X)] = \int_X \phi_{\mathbf{x}}(\mathbf{x})p(\mathbf{x})d\mathbf{x}.$$

- Empirical kernel estimator, given a sample set

$$\mu_X = \sum_{i=1}^{M} w_i \phi_{\mathbf{x}}(\mathbf{x}_i) = \Phi\mathbf{w}$$

  - If $\mathbf{x}_i$ are drawn from $P(\mathbf{x})$, $w_i = 1/M$.




[1] L. Song, K. Fukumizu, and A. Gretton, "Kernel embeddings of conditional distributions: A unified kernel framework for nonparametric inference in graphical models," IEEE Signal Process. Mag., vol. 30, no. 4, pp. 98–111, 2013.

# Preliminaries – Kernel mean embedding (KME)

- The KME approach represents a conditional distribution $P(X|\boldsymbol{y})$ by an element in the feature space

$$\mu_{X|\mathbf{y}} := \mathbb{E}_{X|\mathbf{y}}\left[\phi_{\mathbf{x}}(X)\right] = \int_{X} \phi_{\mathbf{x}}(\mathbf{x}) p(\mathbf{x}|\mathbf{y}) d\mathbf{x}.$$

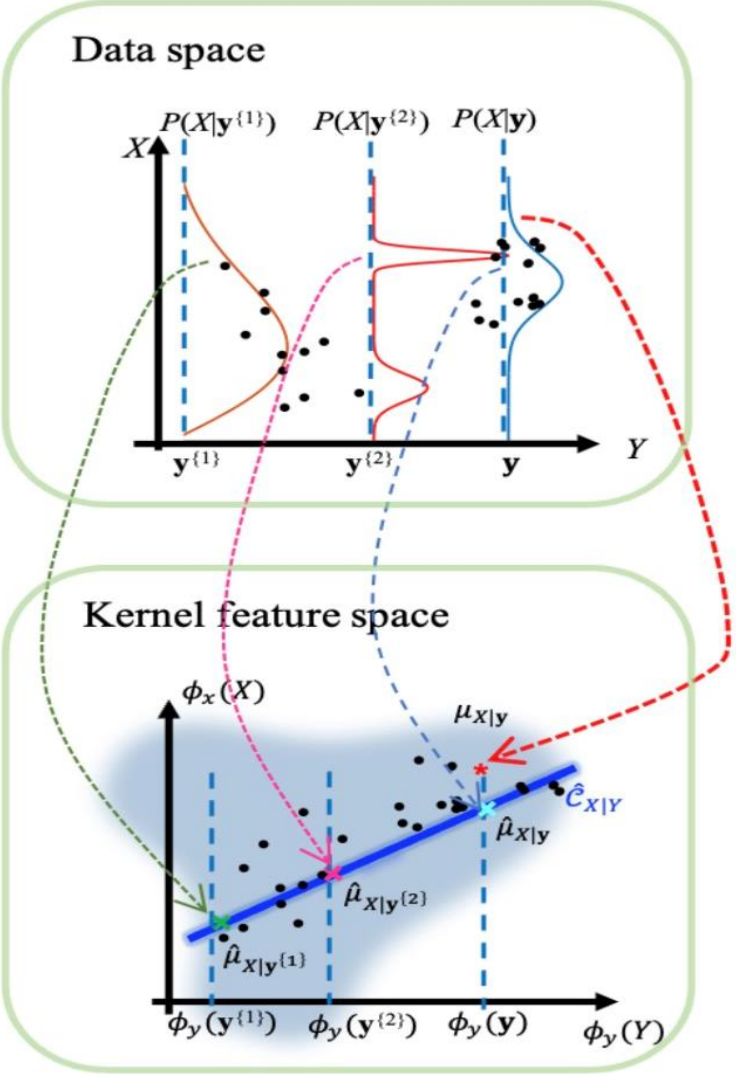
- Define a conditional operator $\mathcal{C}_{X|Y}$ as the <u>linear operator</u> in the feature space to estimate the conditional distribution

$$\mu_{X|\mathbf{y}} = \mathcal{C}_{X|Y}\phi_{\mathbf{y}}(\mathbf{y})$$

- Empirical kernel estimator: The estimate using the $\mathcal{C}_{X|Y}$ is obtained as a <u>linear regression</u> on the kernel weights based on the **training data**

$$\hat{\mu}_{X|\mathbf{y}} = \hat{\mathcal{C}}_{X|Y}\phi(\mathbf{y}) = \Phi\left(G_{\mathbf{yy}} + \kappa I\right)^{-1} \Upsilon^{\mathrm{T}}\phi_{\mathbf{y}}(\mathbf{y}) \equiv \Phi\mathbf{w}.$$

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

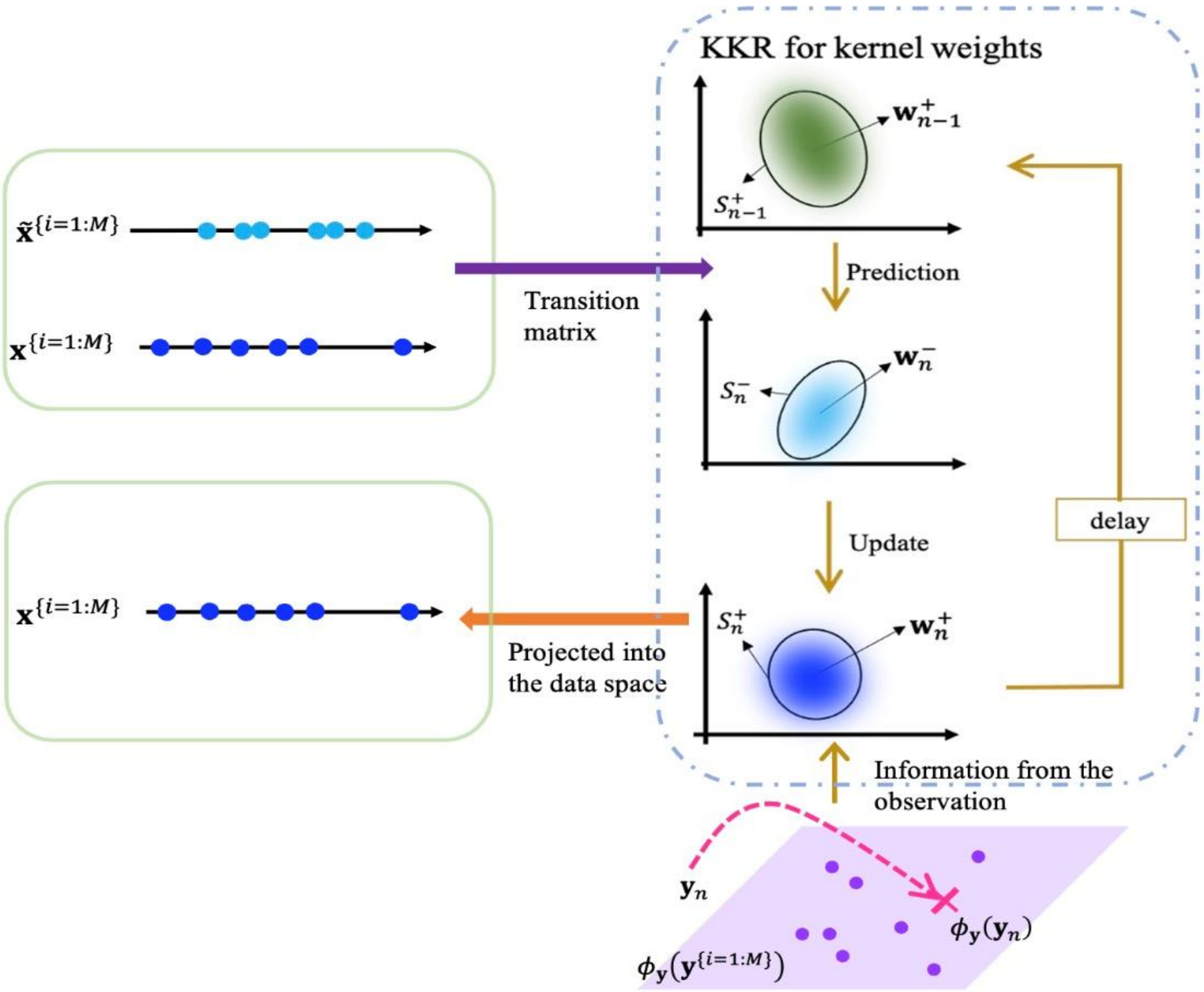


# Preliminaries – Kernel Kalman filter (KKF)

- 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**





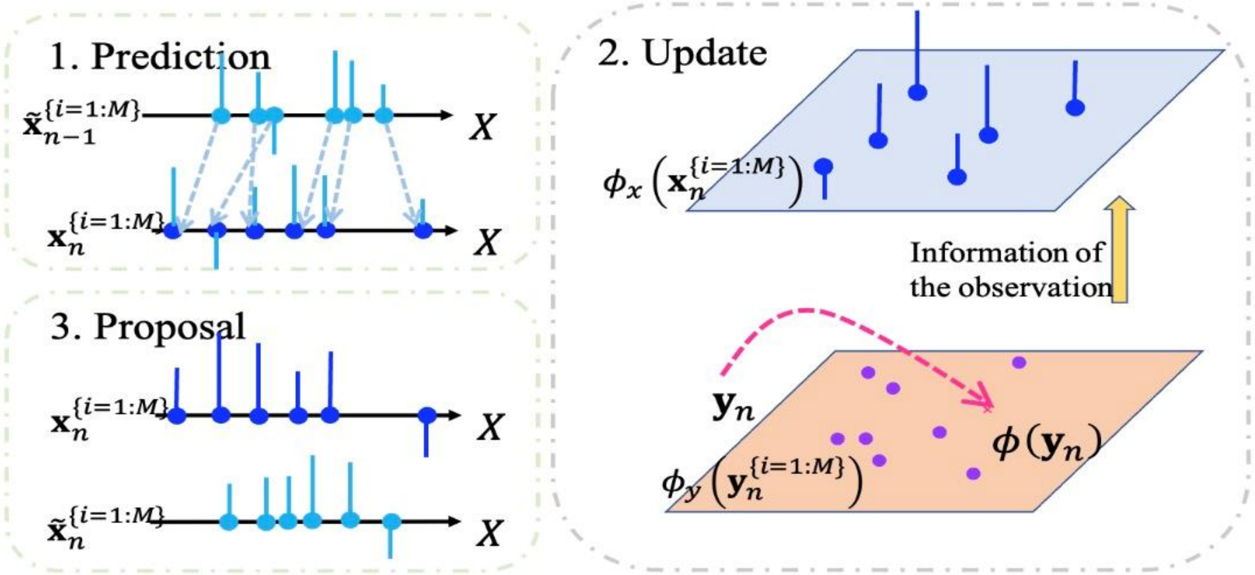

[1] G. Gebhardt, A. Kupcsik, and G. Neumann , “The kernel Kalman rule,” Mach. Learn., pp. 2113–2157, 2019.

# Outlines:

- ❑Background
- ❑Preliminaries
  - Kernel Mean Embedding
  - Kernel Kalman Filter
- ❑Adaptive Kernel Kalman Filter (AKKF)
- ❑AKKF applications
  - AKKF in Single-target Single-sensor Tracking
  - AKKF Multi-Sensor Fusion
  - AKKF Multi-target Tracking(MTT)
- ❑Conclusion

# Adaptive Kernel Kalman Filter (AKKF)

- Replace the still training data set with the **updated** particles/sigma points
- 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)
  - KME of predictive/posterior pdfs: Kernel weight mean and covariance are predicted and updated in the kernel feature space (similar to KKF way)
- Three main steps: proposal, prediction, update,




[1] M. Sun, M. E. Davies, I. Proudler, and J. R. Hopgood, “Adaptive kernel Kalman filter,” in 2021 Sensor Signal Processing for Defence Conference (SSPD), 2021, pp. 1–5
[2] M. Sun, M. E. Davies, I. Proudler, and J. R. Hopgood, “Adaptive kernel Kalman filter,” IEEE Trans. Signal Process., submitted

# Adaptive Kernel Kalman Filter (AKKF)

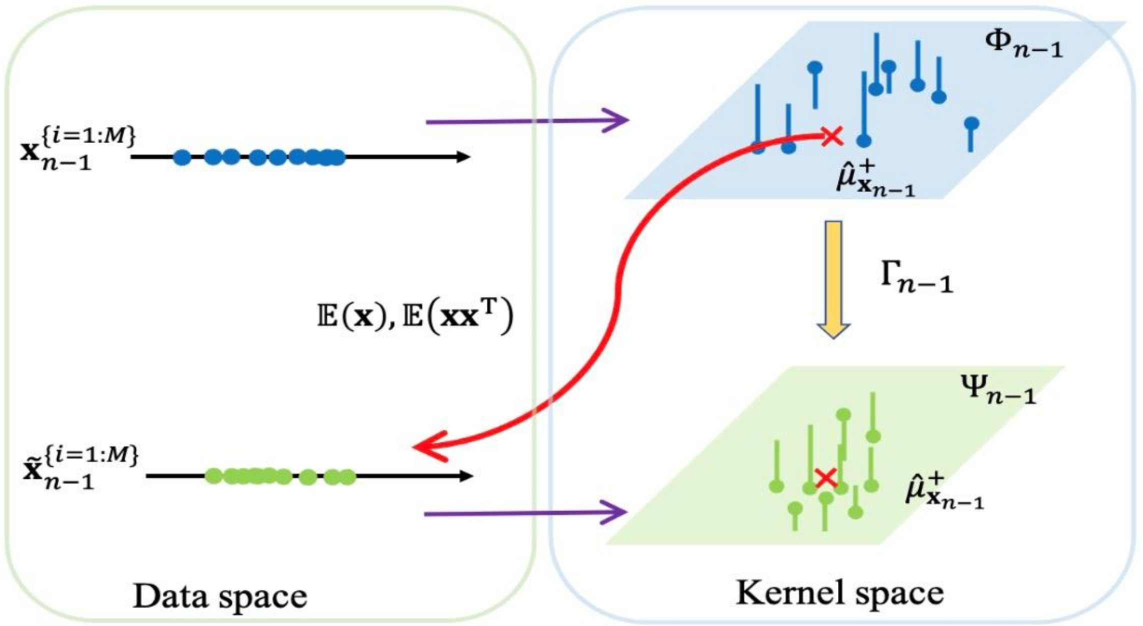
❑ Embedding the Posterior Distribution at time *n*-1

$$\hat{\mu}^{+}_{\mathbf{x}_{n-1}} = \Phi_{n-1}\mathbf{w}^{+}_{n-1},$$

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

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

- 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)*

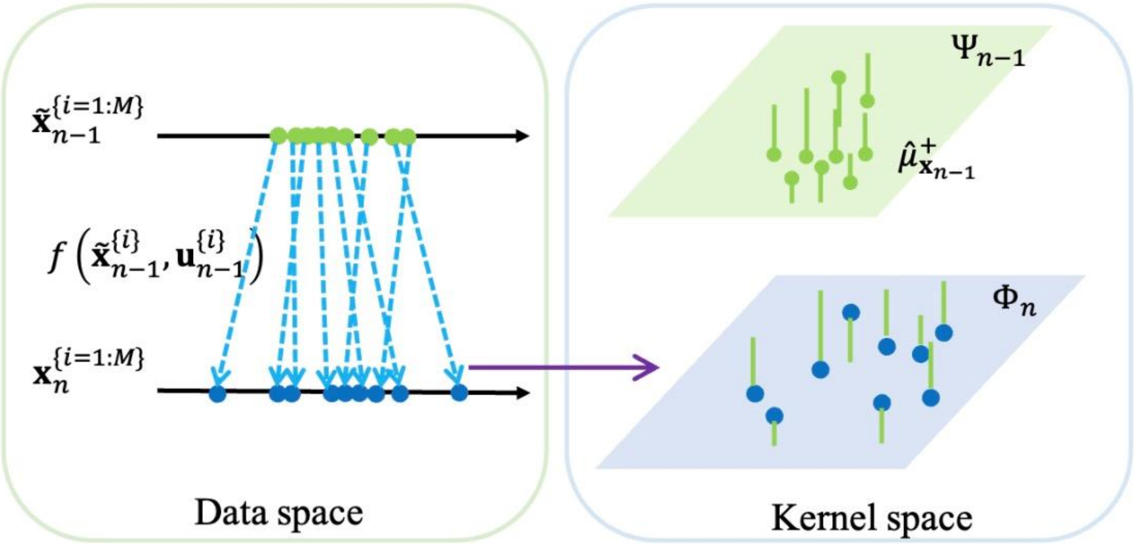
- Predictive particles: propagate proposal particles through the transition function
- New kernel space $\Phi_n$
- Empirical predictive KME by calculating conditional operator

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

- Predictive kernel weight mean and covariance

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

$$S^-_n = \tilde{S}^+_{n-1} + V_n.$$



# Adaptive Kernel Kalman Filter (AKKF)

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

- Observation particles

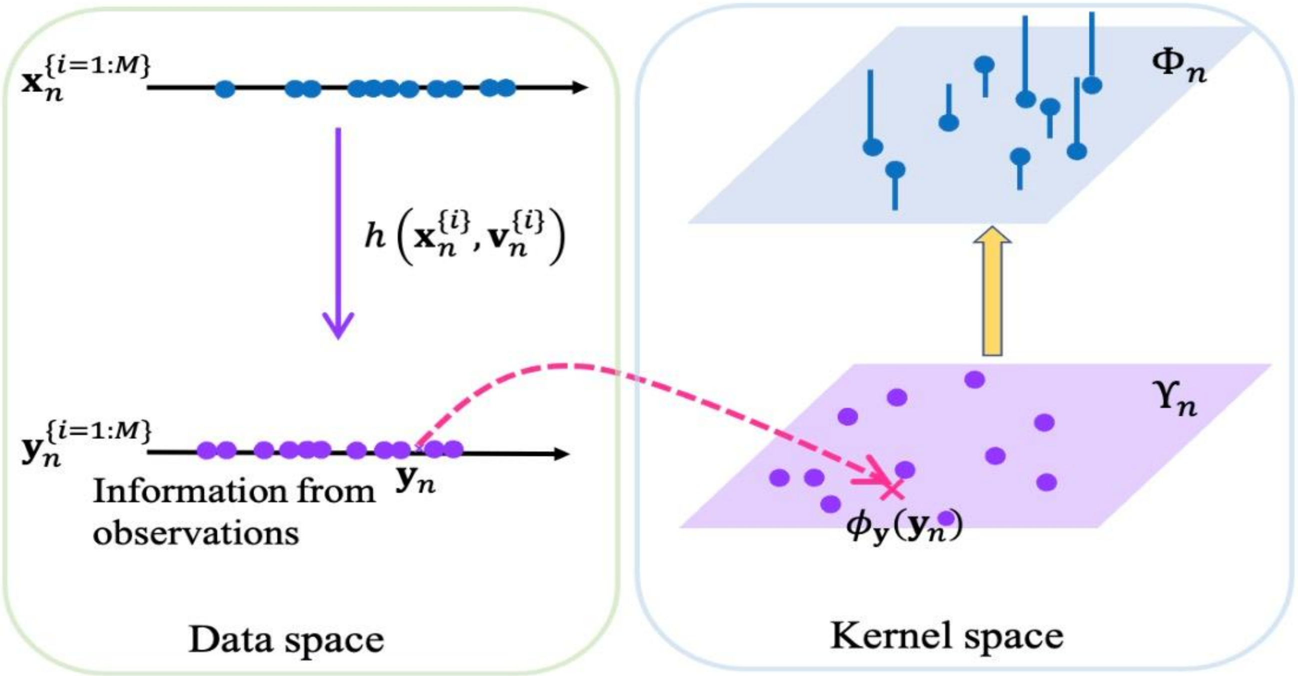
$$\mathbf{y}_n^{\{i\}} = h(\mathbf{x}_n^{\{i\}}, \mathbf{v}_n^{\{i\}}),$$

- Kernel Kalman gain calculation

$$\begin{aligned}\hat{\mu}_{\mathbf{x}_n}^{+} &= \hat{\mu}_{\mathbf{x}_n}^{-} + Q_n\left[\phi_{\mathbf{y}}(\mathbf{y}_n) - \hat{C}_{\mathbf{y}_n|\mathbf{x}_n}\hat{\mu}_{\mathbf{x}_n}^{-}\right],\\ &= \Phi_n\mathbf{w}_n^{+} = \Phi_n\mathbf{w}_n^{-} + Q_n\left[\phi(\mathbf{y}_n) - \hat{C}_{\mathbf{y}_n|\mathbf{x}_n}\hat{\mu}_{\mathbf{x}_n}^{-}\right],\end{aligned}$$

- Update kernel weight mean and covariance

$$\mathbf{w}_n^{+} = \mathbf{w}_n^{-} + Q_n\left(G_{:,\mathbf{y}_n} - G_{\mathbf{y}\mathbf{y}}\mathbf{w}_n^{-}\right).$$
$$S_n^{+} = S_n^{-} - Q_n G_{\mathbf{y}\mathbf{y}} S_n^{-}.$$

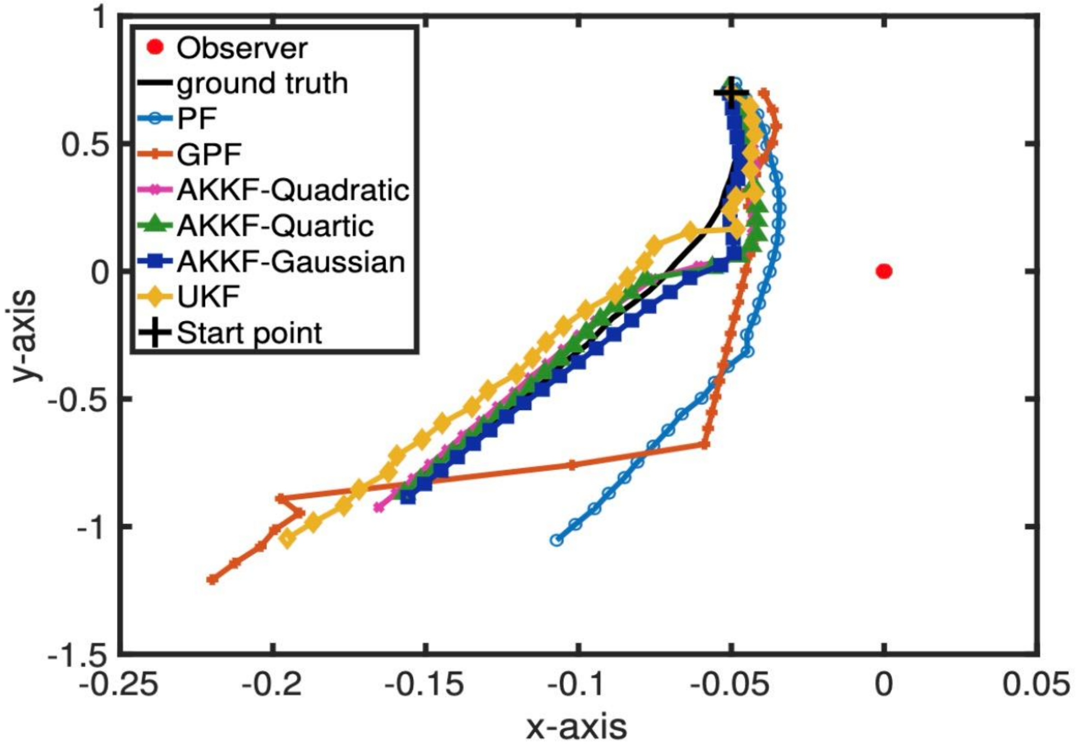


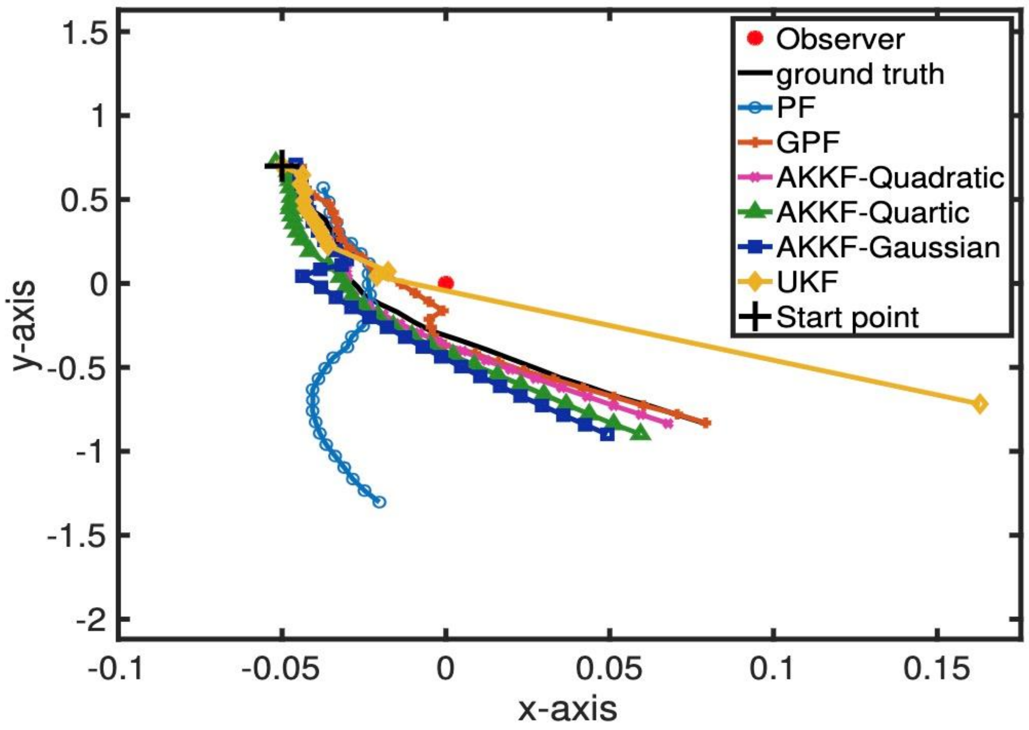
# Outlines:

- ❑Background
- ❑Preliminaries
  - Kernel Mean Embedding
  - Kernel Kalman Filter
- ❑Adaptive Kernel Kalman Filter (AKKF)
- ❑AKKF applications
  - AKKF in Single-target Single-sensor Tracking
  - AKKF Multi-Sensor Fusion
  - AKKF Multi-target Tracking(MTT)
- ❑Conclusion

# AKKF in Single-target Single-sensor Tracking

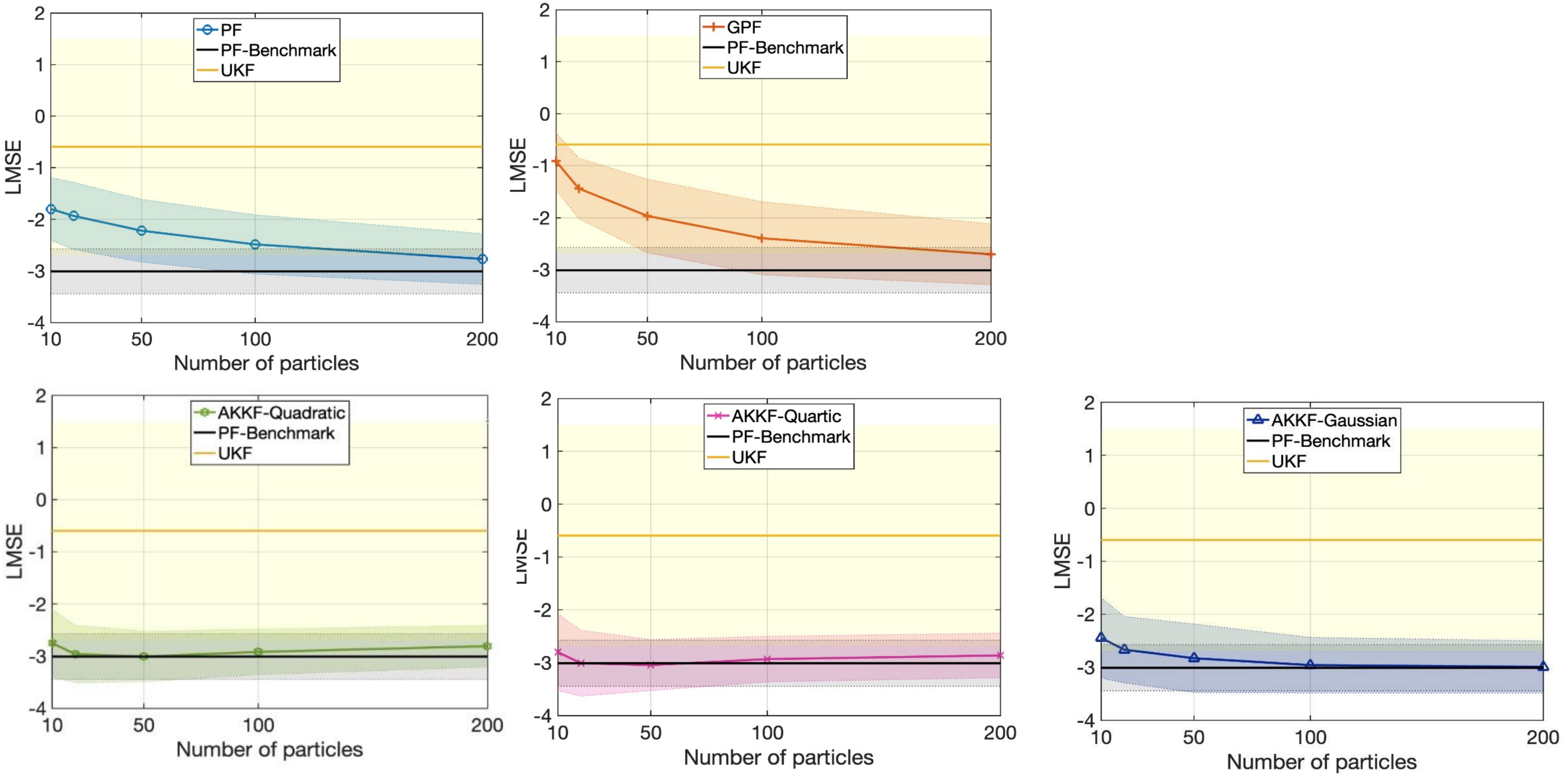
- Constant velocity (CV) motion
- Bearing-only measurement model: $y_n = \tan^{-1}(\frac{\eta_n}{\xi_n}) + v_n.$



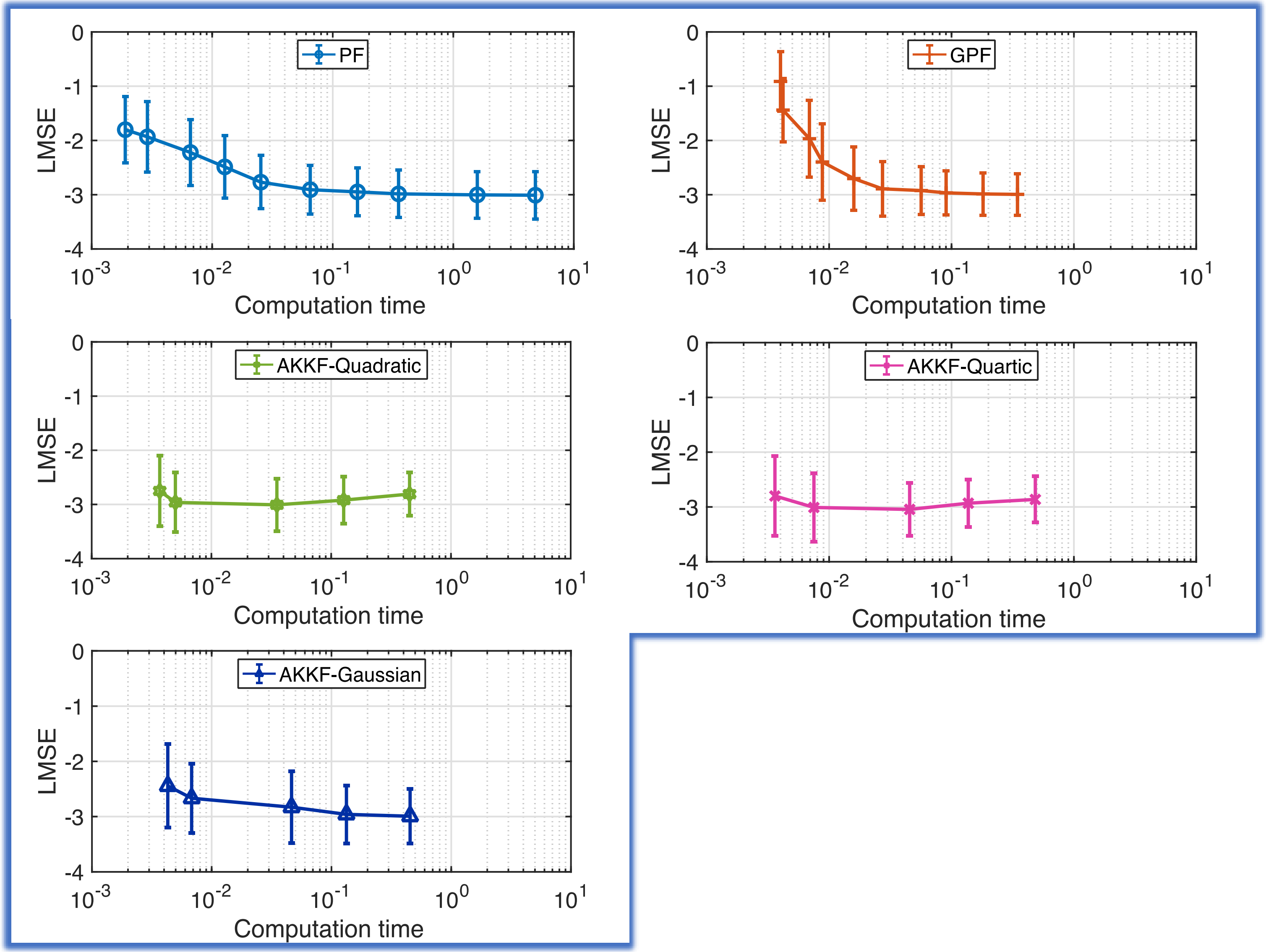



# AKKF in Single-target Single-sensor Tracking

- Average LMSE obtained for 1000 random realizations $\text{LMSE} = \log\left[\frac{1}{N}\sum_{n=1}^{N}\sqrt{(\xi_n - \hat{\xi}_n)^2 + (\eta_n - \hat{\eta}_n)^2}\right]$



# AKKF in Single-target Single-sensor Tracking




- Tracking performance (LMSE) benchmark is −3.0 (PF with 1e4 particles)

| **Filter** | Computation time (s) |
|---|---|
| PF | 0.35 |
| GPF | 0.35 |
| AKKF - quadratic | 0.035 |
| AKKF - quartic | 0.0075 |
| AKKF - Gaussian | 0.45 |

# Outlines:

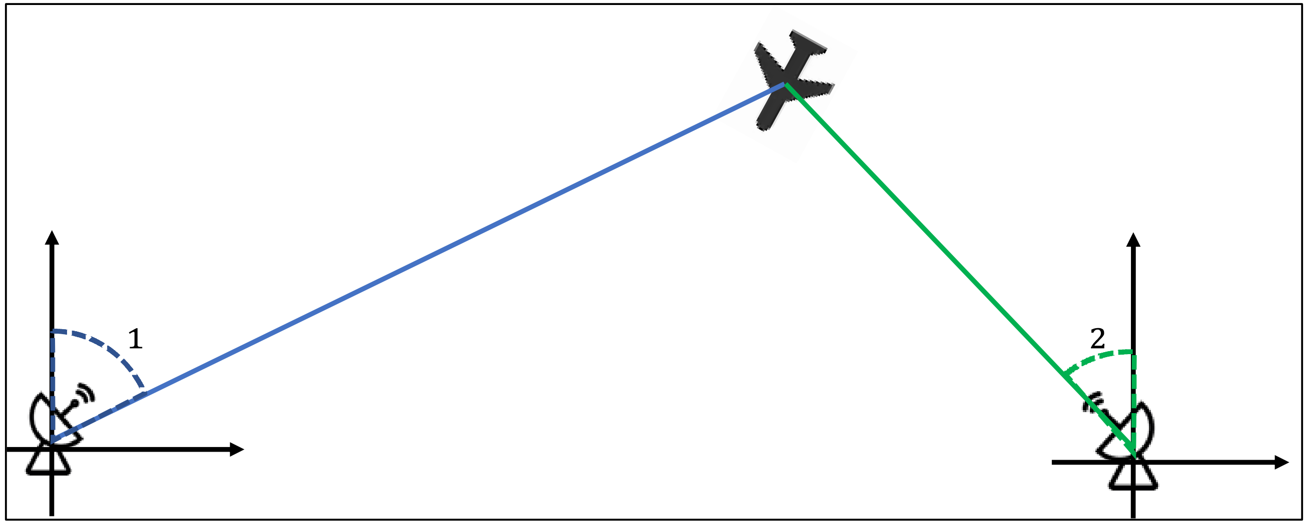
- Background
- Preliminaries
  - Kernel Mean Embedding
  - Kernel Kalman Filter
- Adaptive Kernel Kalman Filter (AKKF)
- AKKF applications
  - AKKF in Single-target Single-sensor Tracking
  - AKKF Multi-Sensor Fusion
  - AKKF Multi-target Tracking(MTT)
- Conclusion

# AKKF Multi-Sensor Fusion

❑ Dynamic state-space model (DSSM)

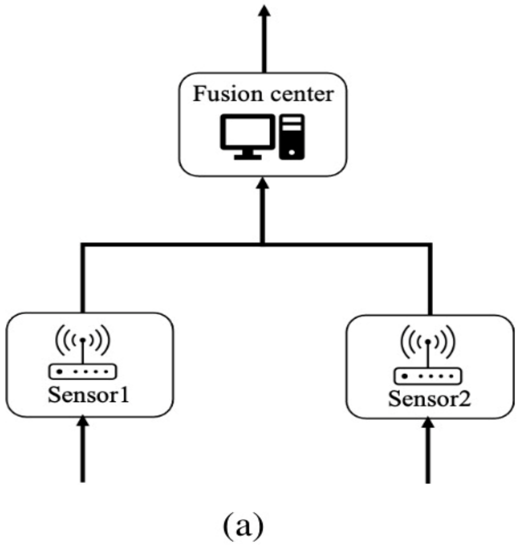
- Motion model
- Measurement model – BOT

$$y_{n,k} = \tan^{-1}\left(\frac{\eta_n - \eta_k}{\xi_n - \xi_k}\right) + v_{n,k}.$$

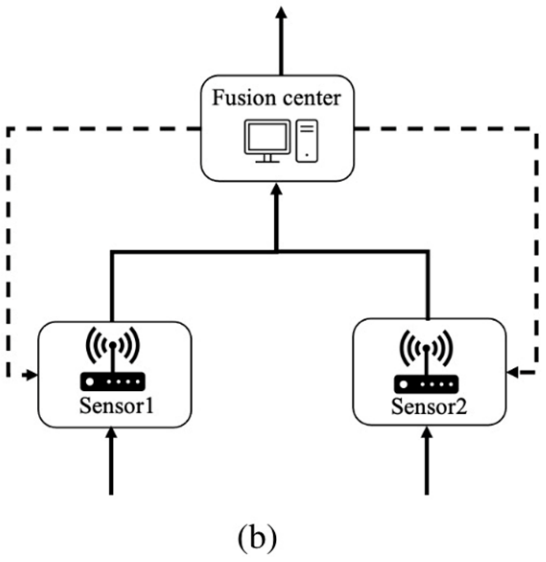


# AKKF Multi-Sensor Fusion

| Filters | Messages from sensors to FC | Messages from FC to sensors |
|---|---|---|
| Centralized fusion | Measurements, measurement models | None |
| Semi-decentralized fusion | Local posterior kernel weight vector and matrix | State particles, global prior kernel weight vector and matrix |




Centralized fusion



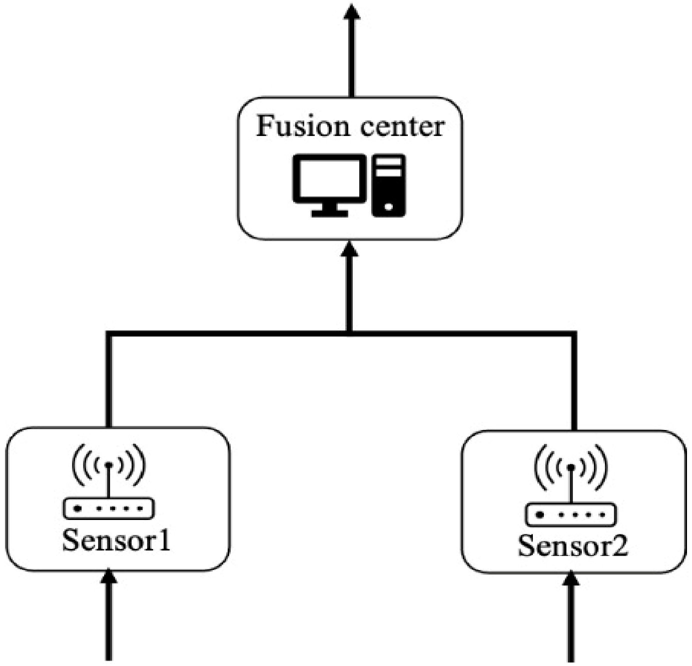

Semi-Decentralized fusion



[1] M. Sun, M. E. Davies, I. Proudler, and J. R. Hopgood, "Adaptive Kernel Kalman Filter Multi-Sensor Fusion," in 24th International Conference on Information Fusion(Fusion2021), pp. 1-8.

# AKKF Multi-Sensor Fusion – Centralized fusion

- ❑ Sensor nodes: the signals from the target are received by sensors
- ❑ The FC node:
  - Measurements from different sensors are combined as a global observation vector
  - Process AKKF in three steps: prediction, update, proposal

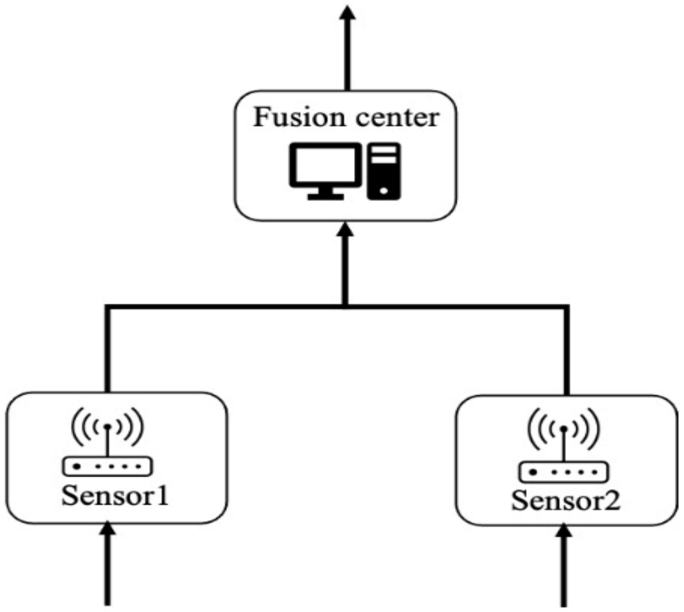



(a)

# AKKF Multi-Sensor Fusion – Centralized fusion

## ❑ Requirements

- FC: necessary processing power and calculation capacity
- Transmit power of sensor nodes
- Transmission bandwidth from sensors to the FC



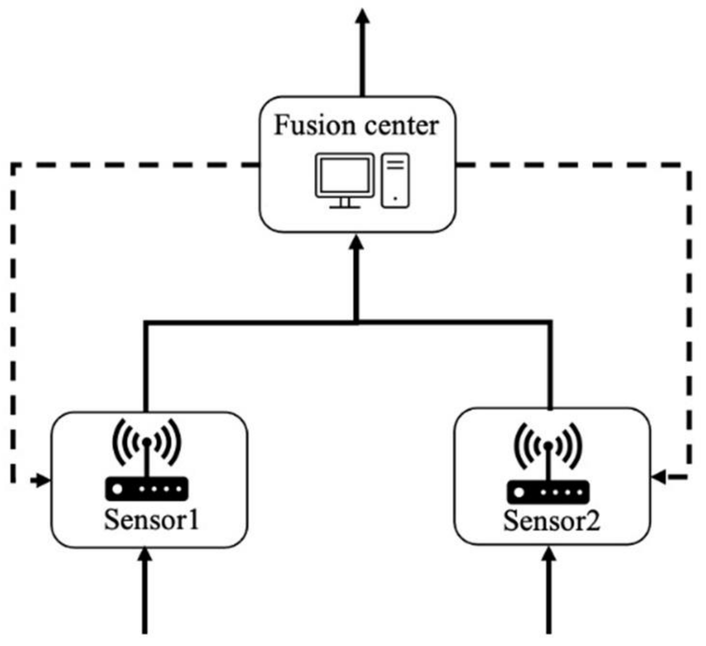

(a)

# AKKF Multi-Sensor Fusion– Semi-decentralized fusion

- FC: broadcasts the prior kernel weight vector, matrix, and the global state particles to sensors
- Sensor: The posterior weight vector and covariance matrix are calculated at each sensor
- FC: Global posterior kernel weight vector and covariance matrix: weighted Kullback–Leibler average[1]

$$(S_n^+)^{-1} = \sum_{k=1}^{K} \omega_k (S_{n,k}^+)^{-1},$$

$$(S_n^+)^{-1}\mathbf{w}_n^+ = \sum_{k=1}^{K} \omega_k (S_{n,k}^+)^{-1}\mathbf{w}_{n,k}^+.$$



(b)



[1]"Kullback–leibler average, consensus on probability densities, and dis-tributed state estimation with guaranteed stability,"Automatica, vol. 50,no. 3, pp. 707–718, 2014

# AKKF Multi-Sensor Fusion– Semi-decentralized fusion

- Modular approach to multi-sensor networks
- Advantages:
  - Reduced computation at the FC
  - Reduced transmit power at the sensors
  - Reduced forward bandwidth

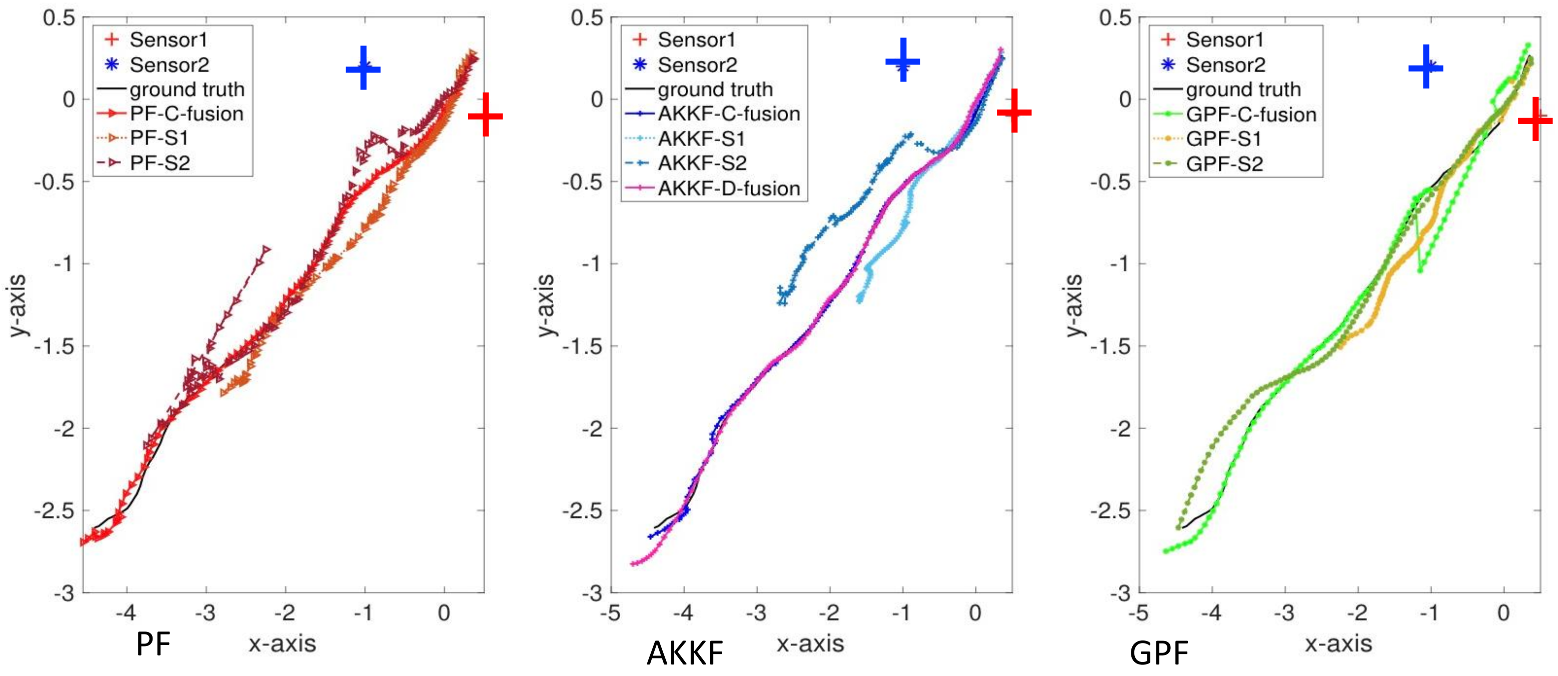
# AKKF Multi-Sensor Fusion

- DSSM

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

$$y_{n,k} = h_k(\mathbf{x}_n, v_{n,k}) = \tan^{-1}\left(\frac{\eta_n - \eta_k}{\xi_n - \xi_k}\right) + v_{n,k}.$$

- The different angular resolutions are modelled by adding different amount of noise to the exact bearing



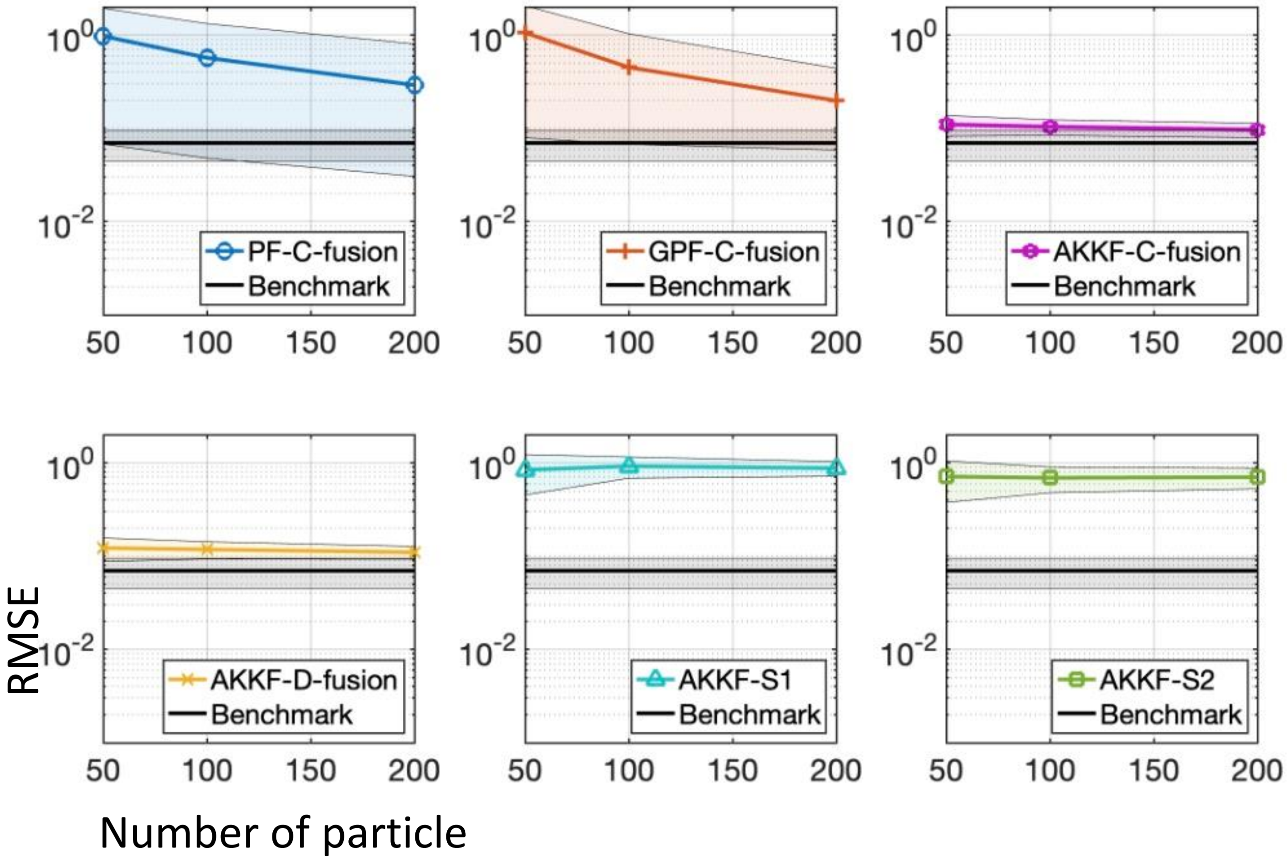

Trajectory

- Discussion of simulation results
  - Fusion helps all three filters
  - D-fusion performance is as good as C-fusion AKKF

# AKKF Multi-Sensor Fusion

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{n=1}^{N}(\xi_n - \hat{\xi}_n)^2 + (\eta_n - \hat{\eta}_n)^2}{N}}$$

- Benchmark: Centralized fusion-based PF with 2000 particles
- AKKF has improved performance/robustness with lower number of particle
- Semi-decentralized fusion–based AKKF achieves almost good performance as centralized fusion–based AKKF

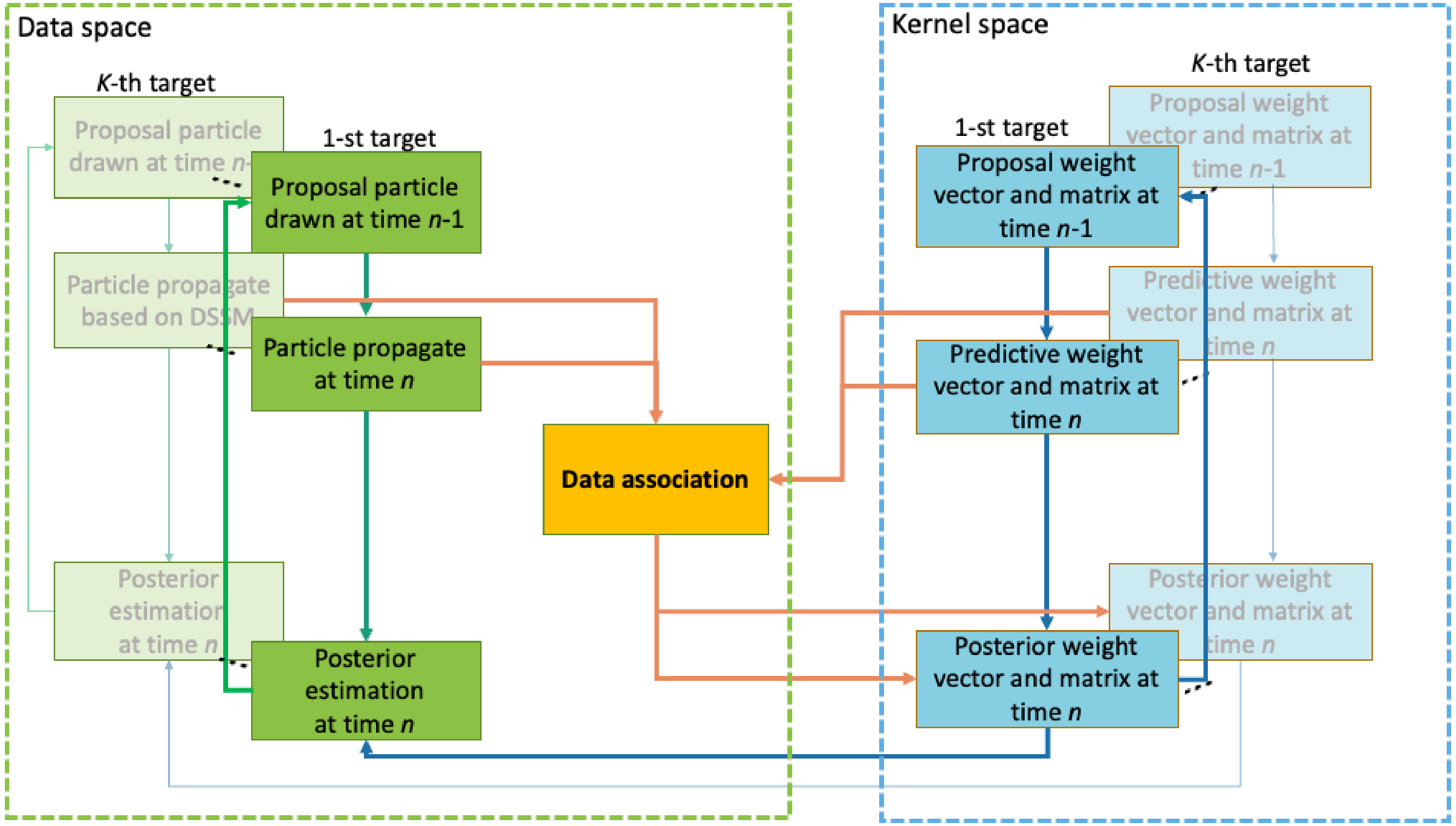


# Outlines:

- ❑ Background
- ❑ Preliminaries
  - Kernel Mean Embedding
  - Kernel Kalman Filter
- ❑ Adaptive Kernel Kalman Filter (AKKF)
- ❑ AKKF applications
  - AKKF in Single-target Single-sensor Tracking
  - AKKF Multi-Sensor Fusion
  - AKKF Multi-target Tracking(MTT)
- ❑ Conclusion

# AKKF Multi-target tracking(MTT) – Known number of targets

- Fix number of targets with clutter and missing alarm
- Data association: belief propagation outside kernel space



# AKKF Multi-target tracking(MTT) – Known number of targets

## ❑ DSSM

- Constant-velocity (CV) motion model
- Measurement model

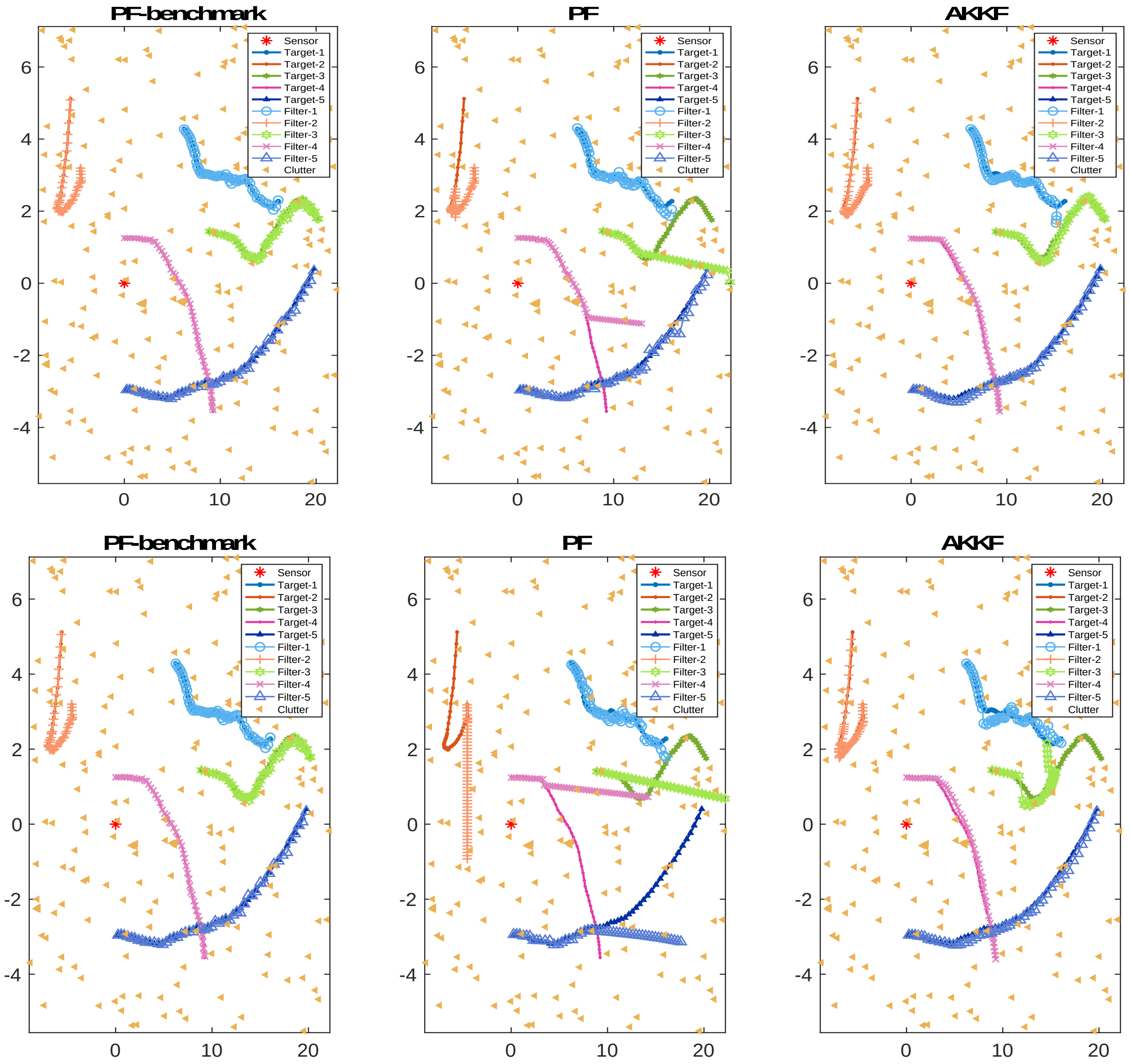
$$h(\mathbf{x}_{t,k}) = \begin{bmatrix} r_{t,k} \\ \phi_{t,k} \end{bmatrix} = \begin{bmatrix} \sqrt{\xi_{t,k}^2 + \eta_{t,k}^2} \\ \tan^{-1}(\xi_{t,k}, \eta_{t,k}) \end{bmatrix} + \mathbf{v}_{t,k}.$$

## ❑ Benchmark

- PF with 2000 particles, no missing or false alarms, prior data association

## ❑ AKKF

- Quartic kernel based




Pd = 1;
M = 50;

Pd = 0.6;
M = 50;

# Outlines:

- ❑Background
- ❑Preliminaries
  - Kernel Mean Embedding
  - Kernel Kalman Filter
- ❑Adaptive Kernel Kalman Filter (AKKF)
- ❑AKKF applications
  - AKKF in Single-target Single-sensor Tracking
  - AKKF Multi-Sensor Fusion
  - AKKF Multi-target Tracking(MTT)
- ❑Conclusion

# 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
- Extend the application of AKKF into multi-sensor multi-target tracking systems

## ❑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