UNIVERSITÉ DE STRASBOURG

# MLMS

Inria
inventors for the digital world

# Parameters estimation using Kalman filtering for predictive simulation

Sergei Nikolaev

# Data assimilation

- **DA is a method to combine observations with a model to improve the model accuracy [M. Asch 2016]**

- **We need DA to predict the state of the system and its future in a best possible way [M. Asch 2016]**

- **For prediction we rely on models, but when they are not periodically corrected they have a little value (inverse problem) [M. Asch 2016]**

# Well- and ill-posed problems

- A problem is well-posed if:
  - Solution exists
  - Solution is unique
  - Solution depends continuously on the data
- Inverse problems are usually ill-posed !!!

# Methods

- **Statistical (Kalman filtering family)**
- **Variational (3Dvar, 4Dvar)**
- **Hybrid and others …**

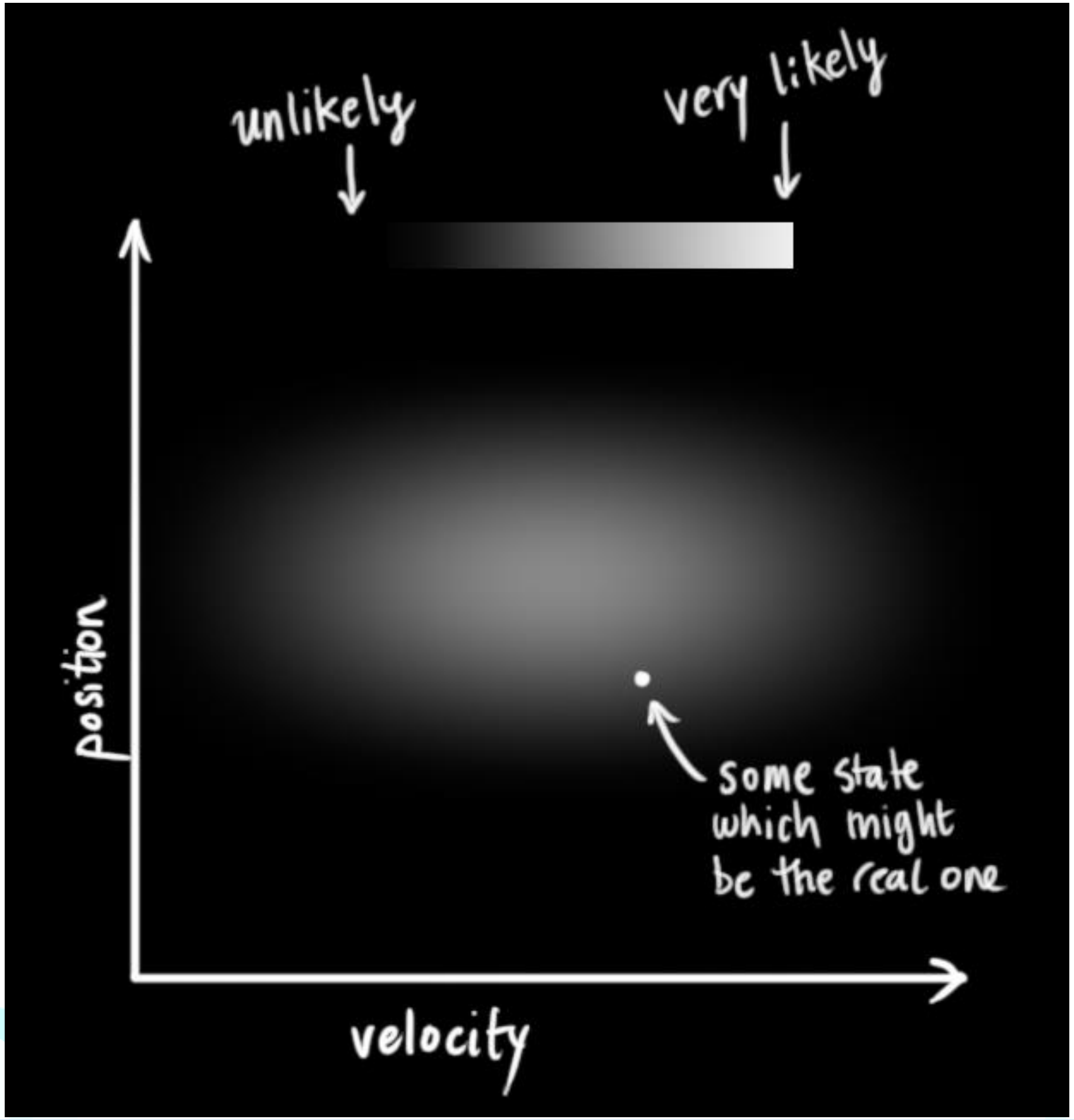
# Kalman filtering example

**Reference:**

**https://www.bzarg.com/p/how-a-kalman-filter-works-in-pictures/**

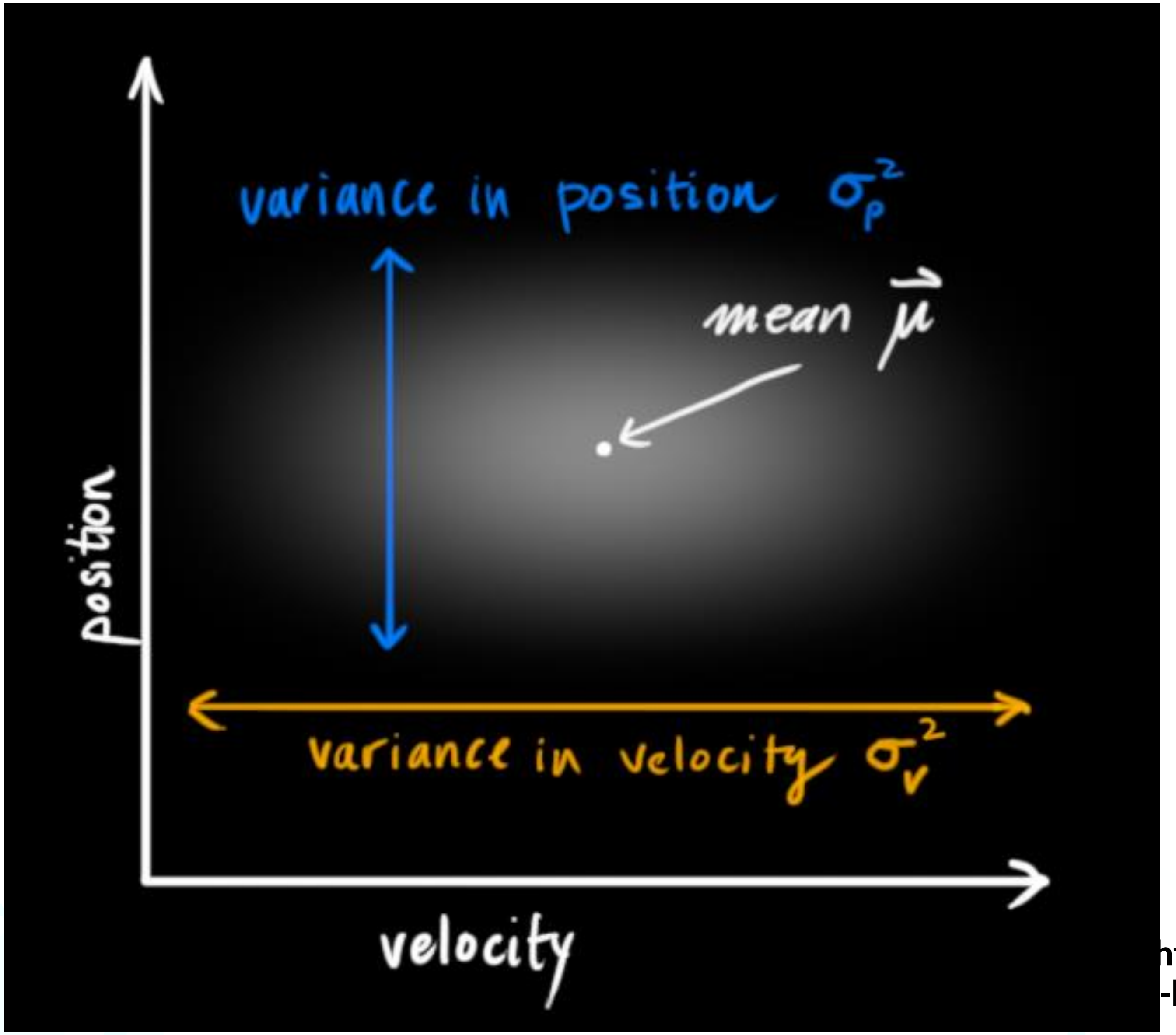
- **Robot, estimate positions and velocity**
- **Photo and speed tracking elements to get information from real world**

# Kalman filtering example



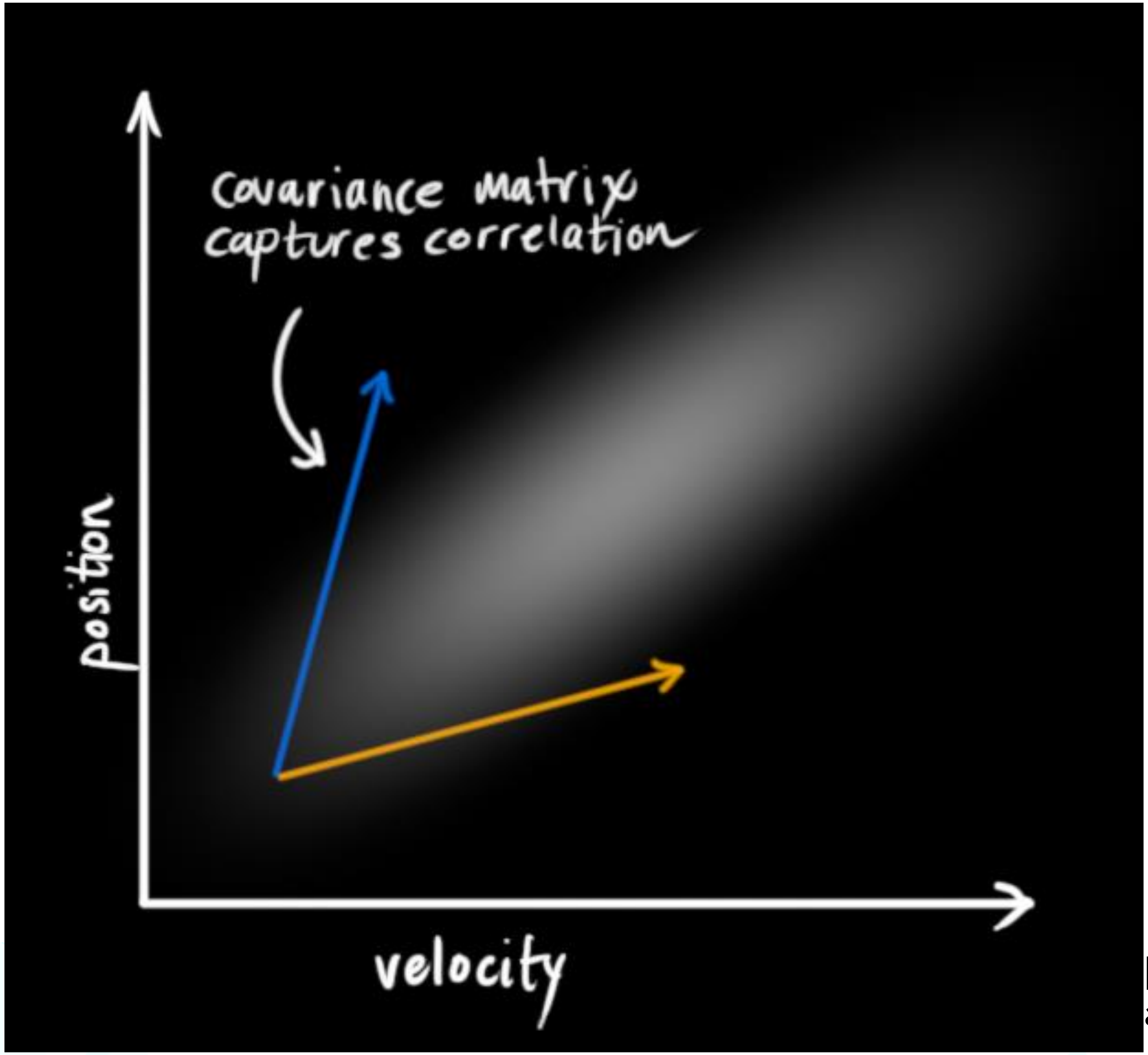

[https://www.bzarg.com/p/how-a-kalman-filter-works-in-pictures/]

# Kalman filtering example



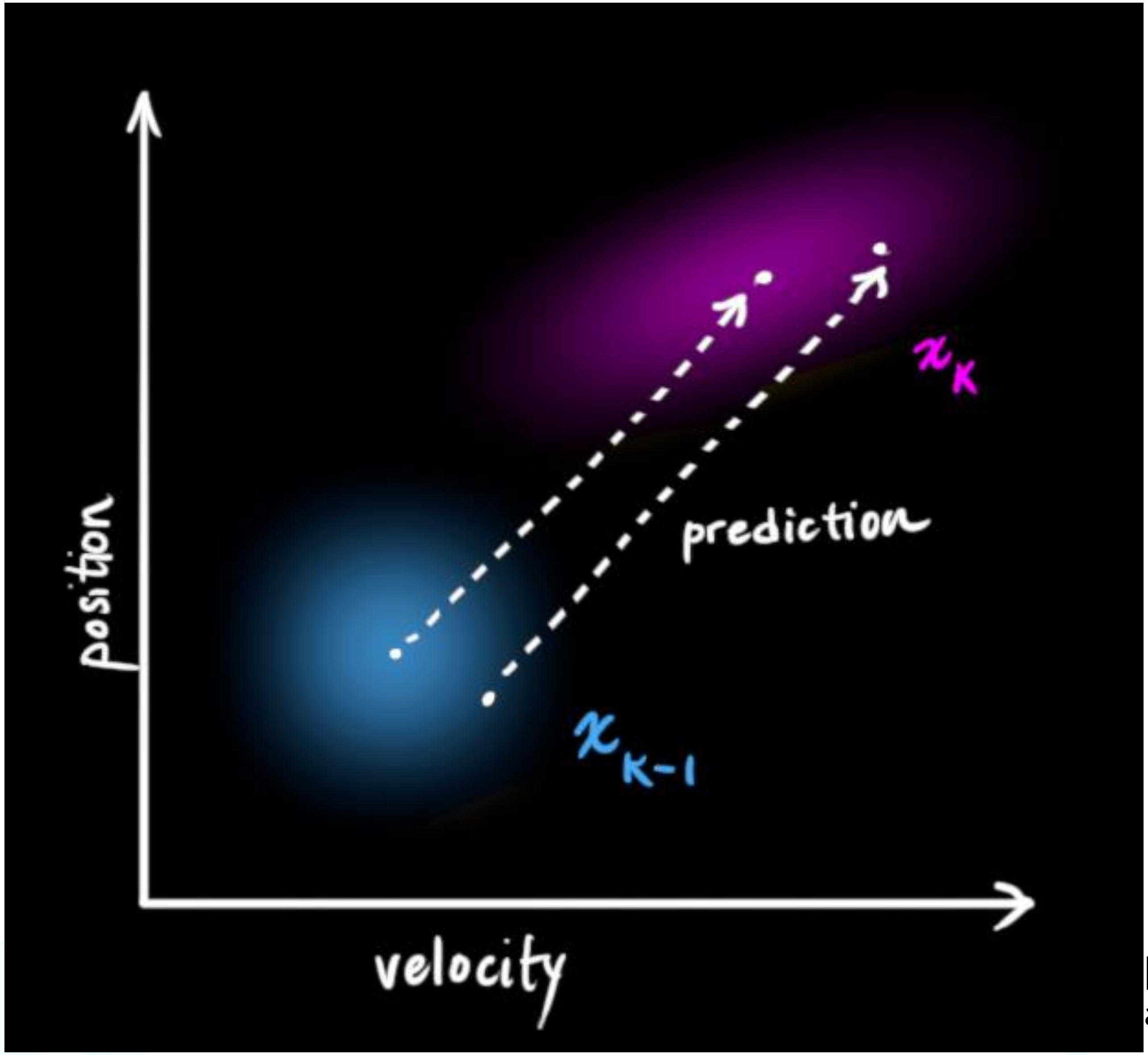

[https://www.bzarg.com/p/how-a-kalman-filter-works-in-pictures/]

# Kalman filtering example



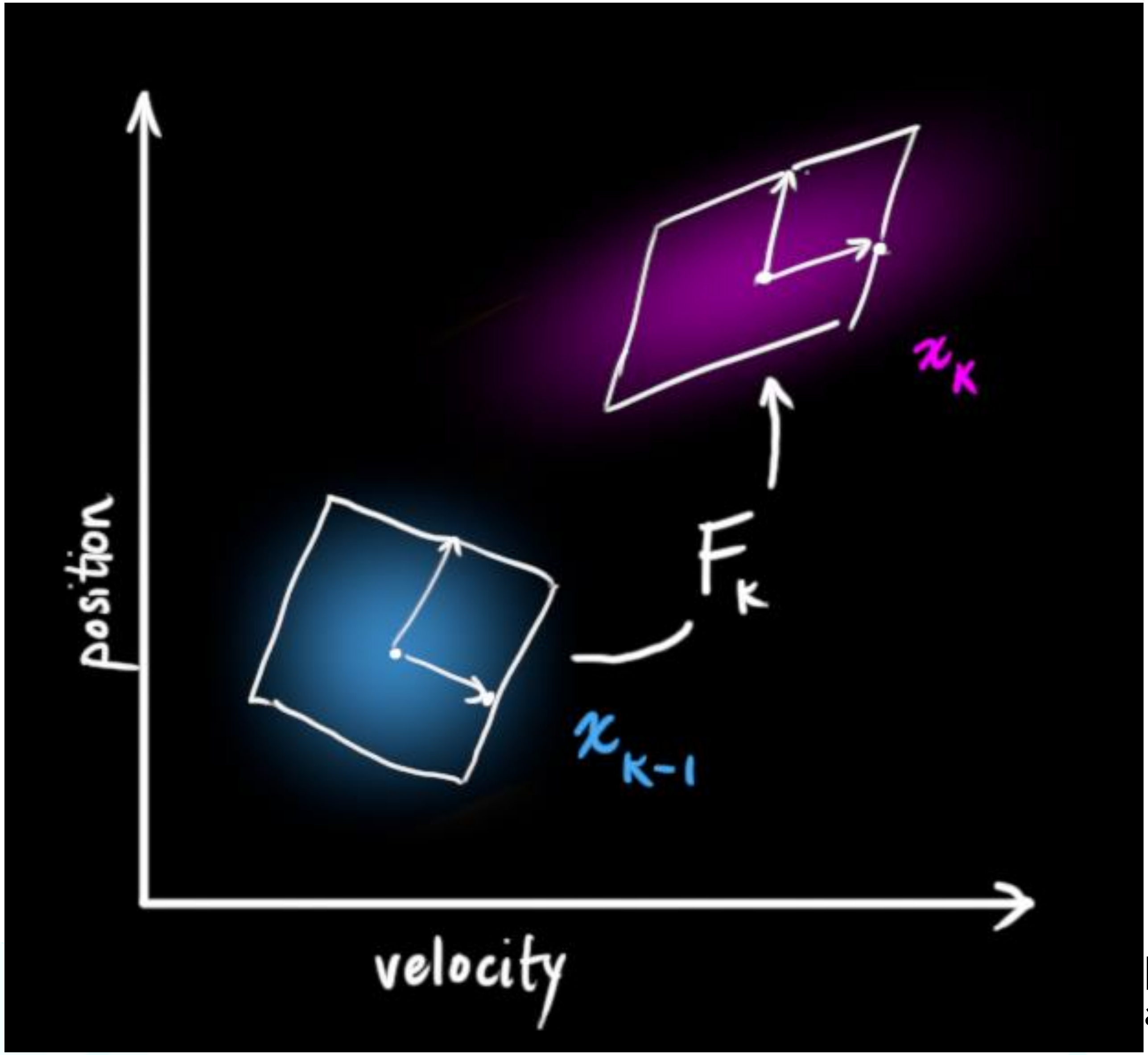

[https://www.bzarg.com/p/how-a-kalman-filter-works-in-pictures/]

# Kalman filtering example



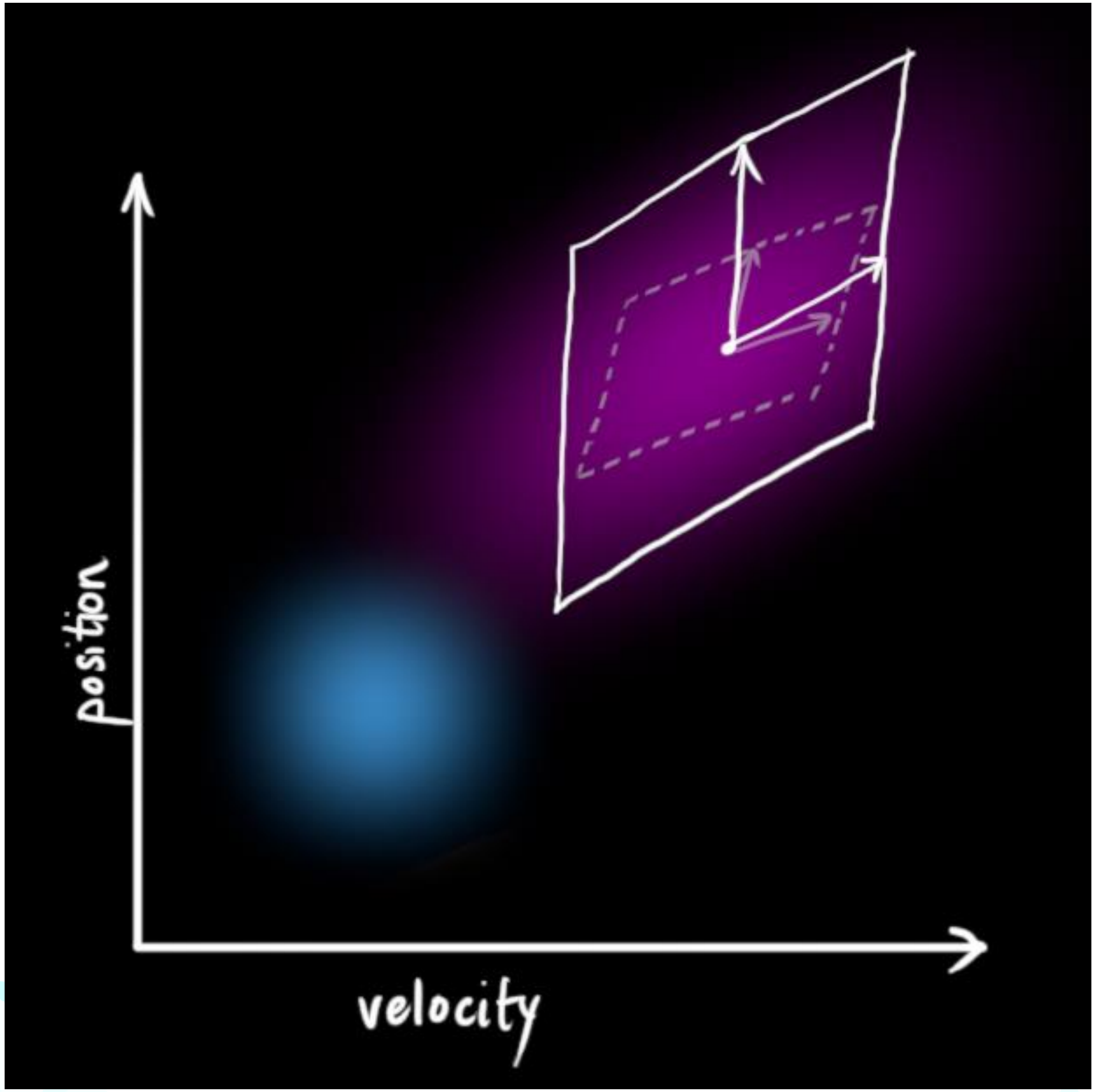

[https://www.bzarg.com/p/how-a-kalman-filter-works-in-pictures/]

# Kalman filtering example



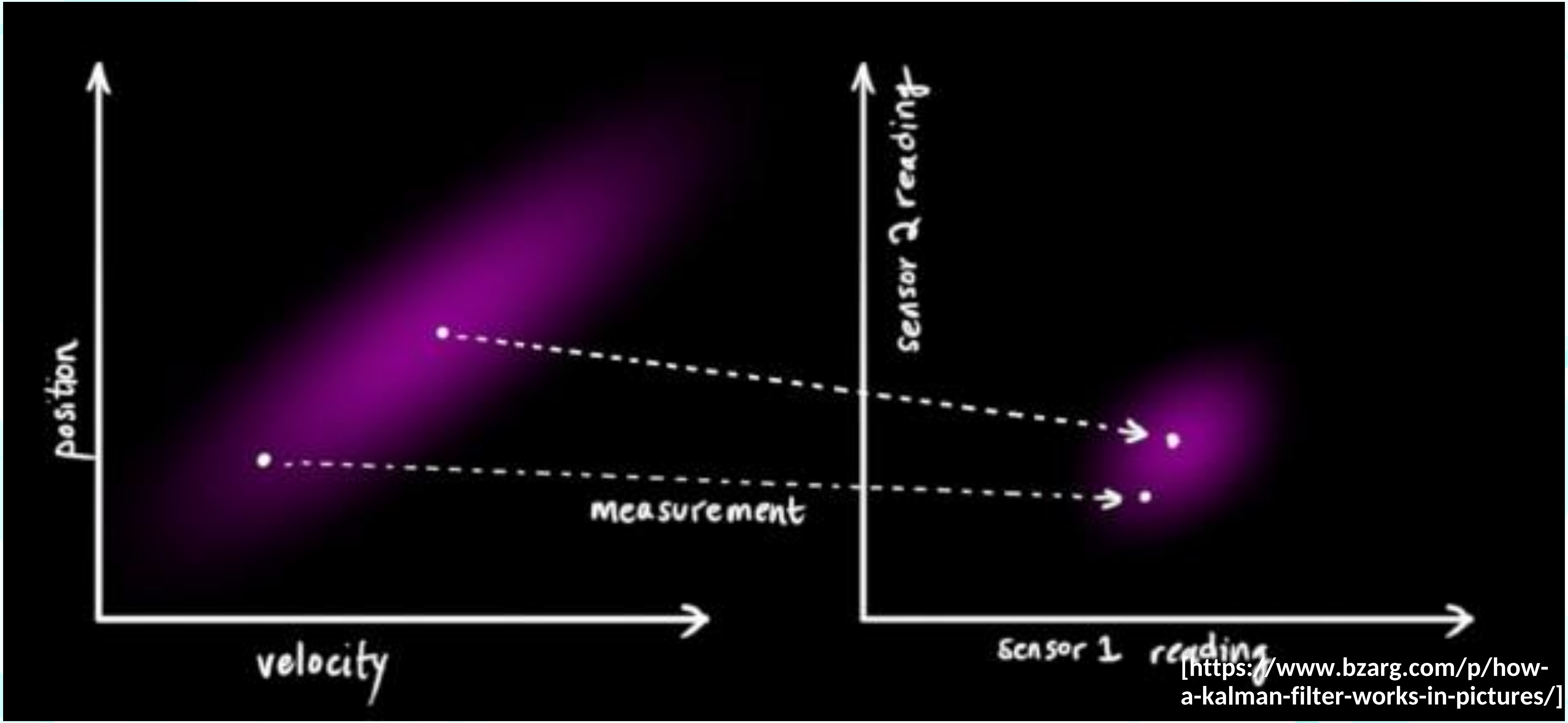

[https://www.bzarg.com/p/how-a-kalman-filter-works-in-pictures/]

# Kalman filtering example



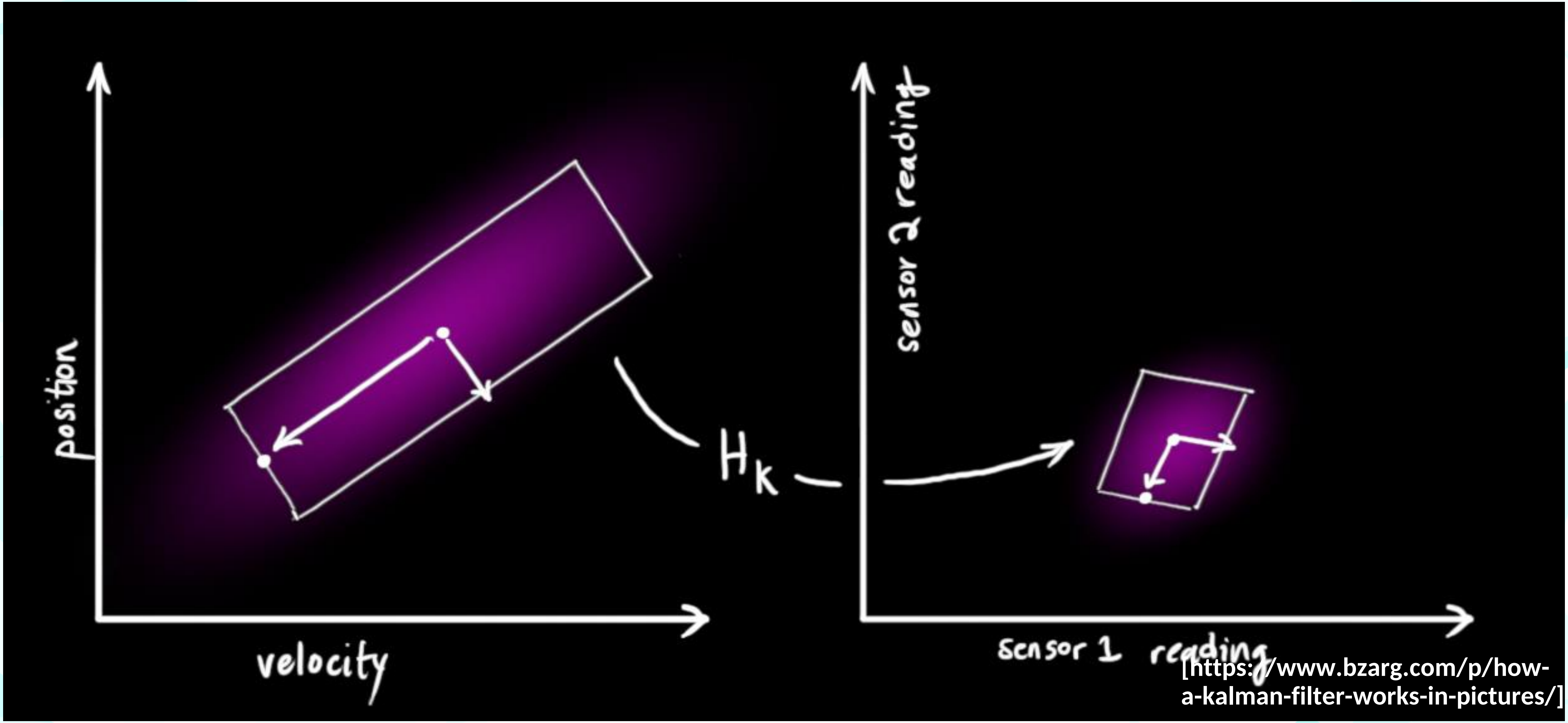

[https://www.bzarg.com/p/how-a-kalman-filter-works-in-pictures/]

# Kalman filtering example



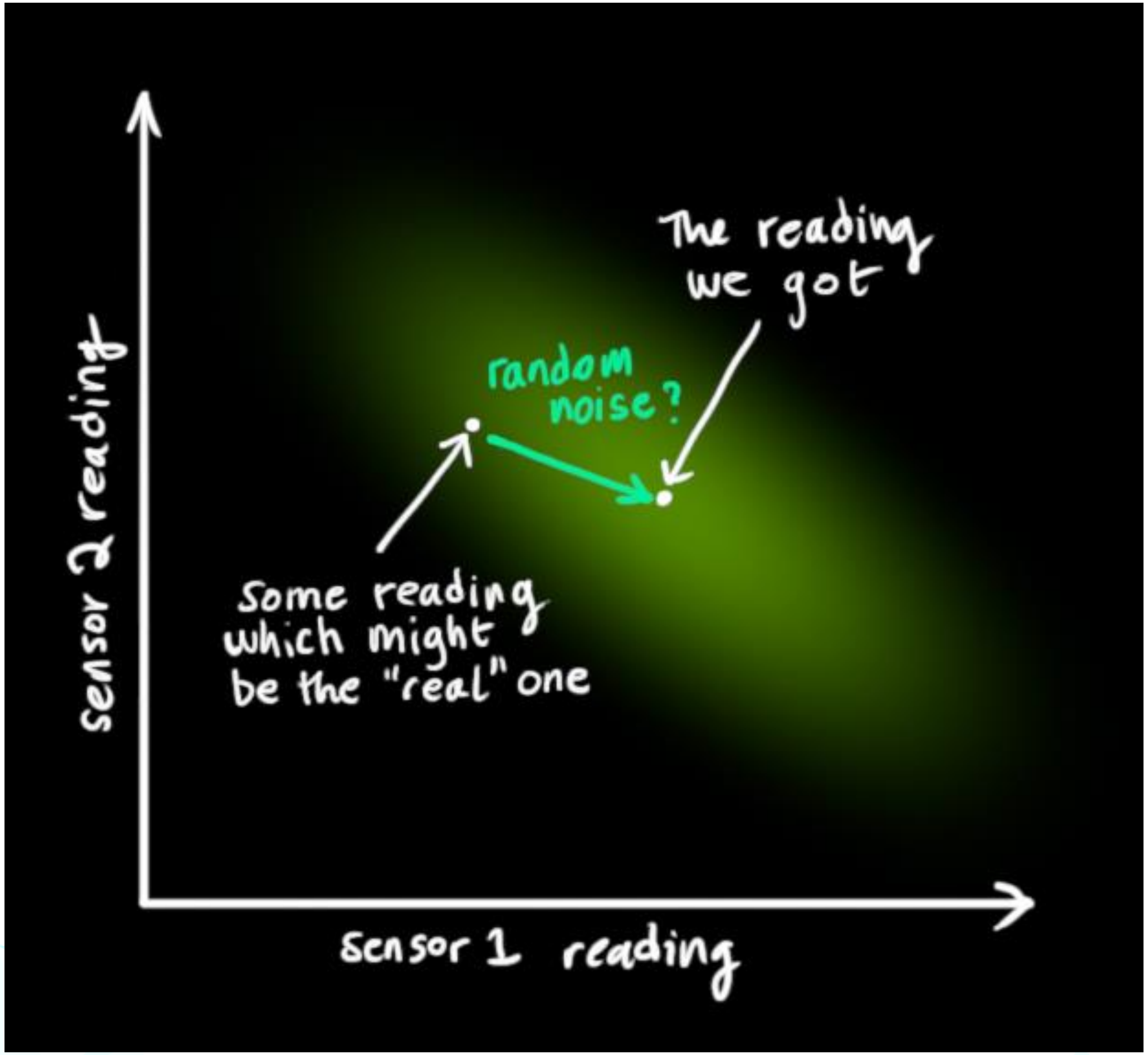

[https://www.bzarg.com/p/how-a-kalman-filter-works-in-pictures/]

# Kalman filtering example



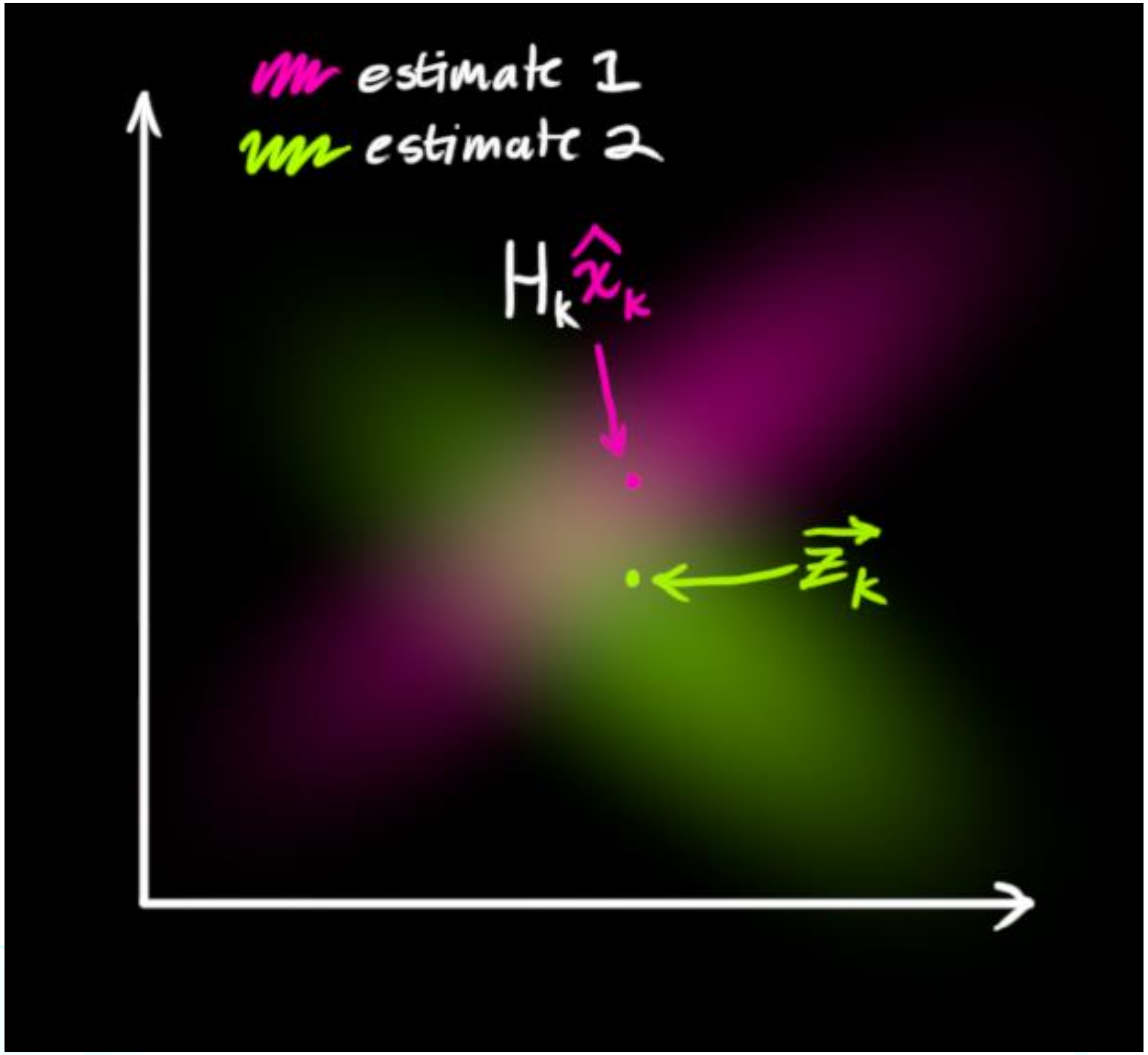

[https://www.bzarg.com/p/how-a-kalman-filter-works-in-pictures/]

# Kalman filtering example



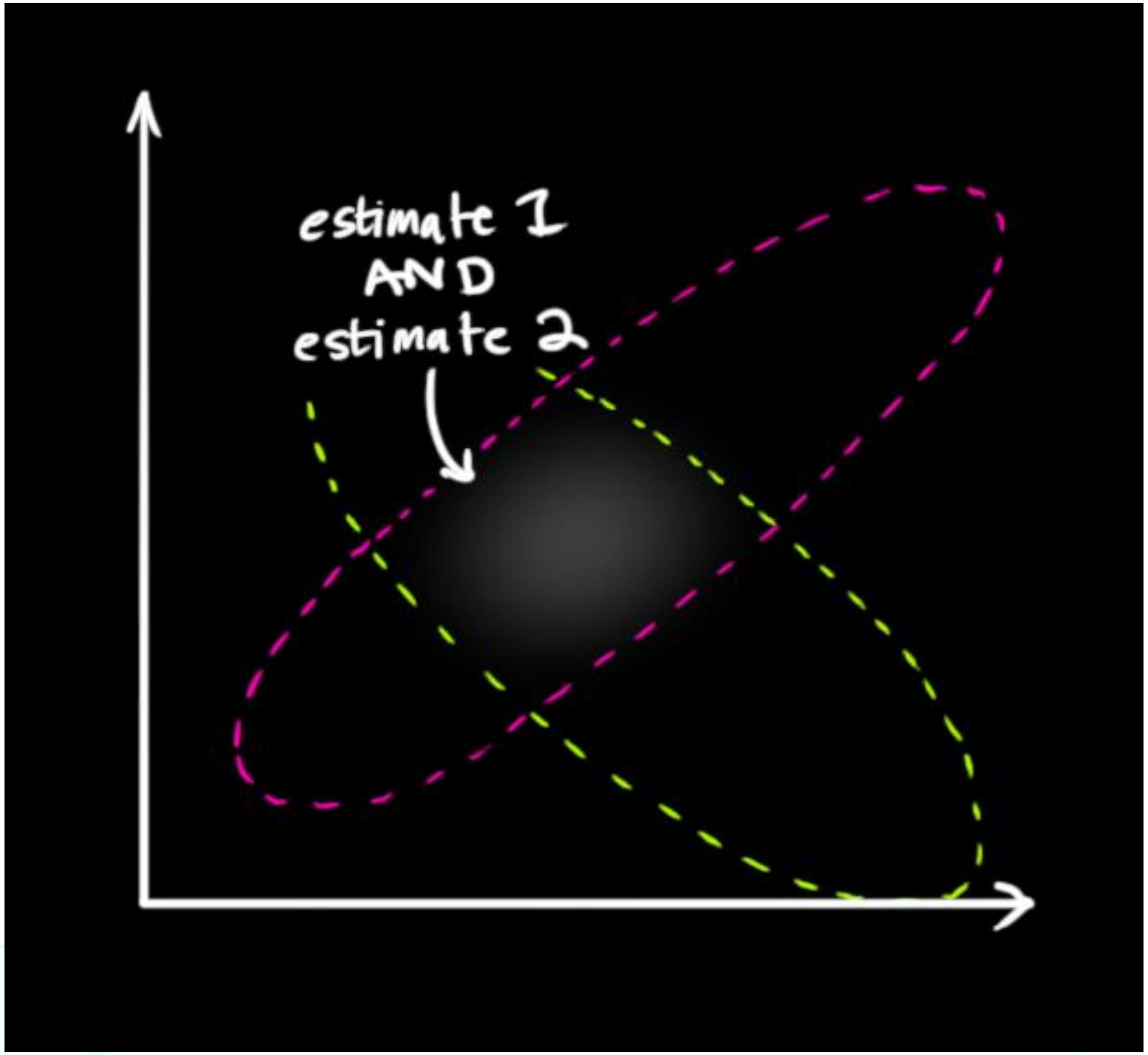

**[https://www.bzarg.com/p/how-a-kalman-filter-works-in-pictures/]**

# Kalman filtering example



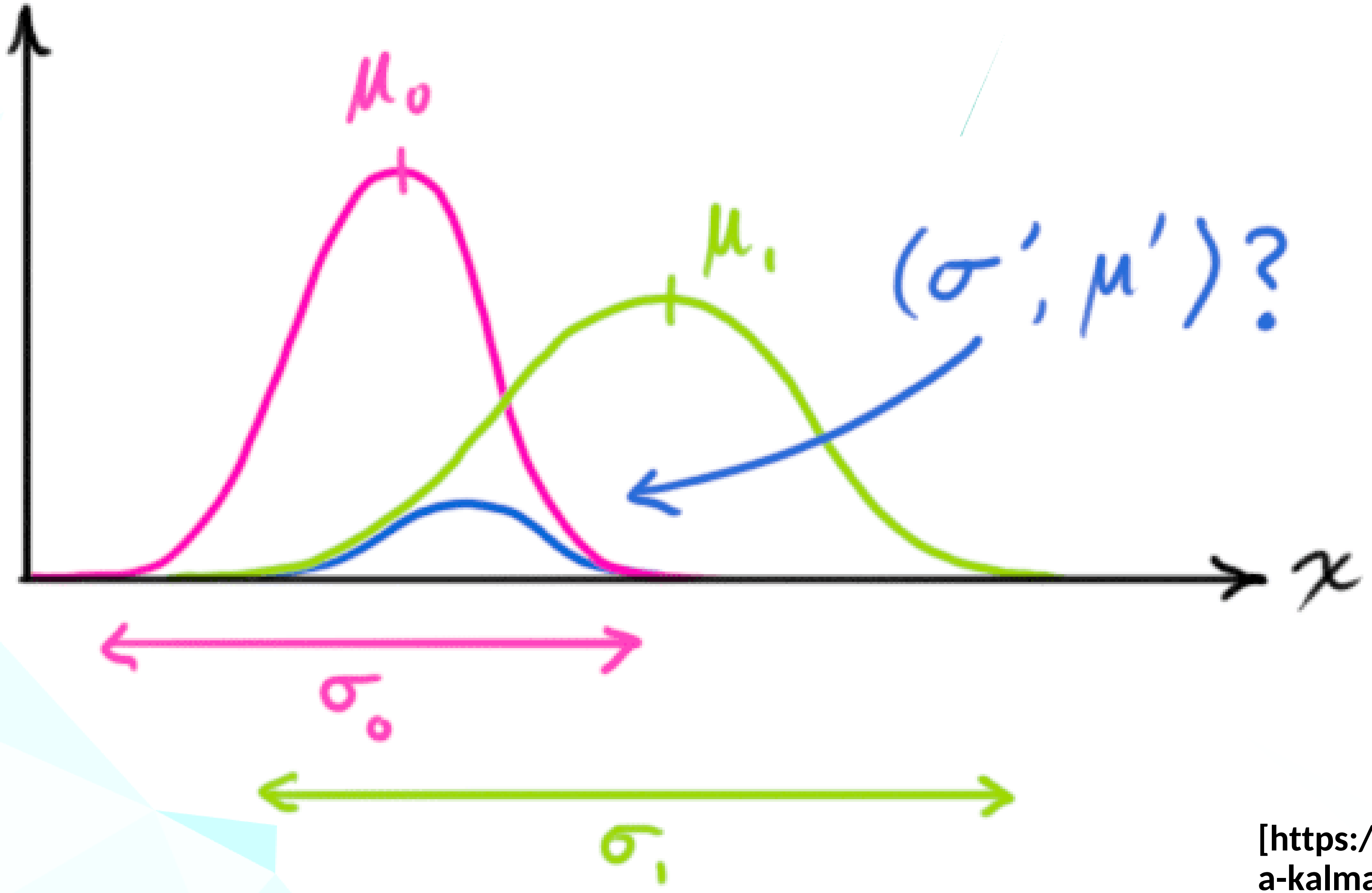

[https://www.bzarg.com/p/how-a-kalman-filter-works-in-pictures/]

# Kalman filtering example




[https://www.bzarg.com/p/how-a-kalman-filter-works-in-pictures/]

# Kalman filtering example




[https://www.bzarg.com/p/how-a-kalman-filter-works-in-pictures/]

# Formalization

## Linear

Suppose we have iterative process

$$x_k = A * x_{k-1} + B * u_k + w \qquad p(w) \sim N(0, Q)$$

$$z_k = H * x_k + v \qquad p(v) \sim N(0, R)$$

where

- $x$ - unknown state of size $n$
- $u$ - input sequence
- $z$ - observations data of size $m$
- $w$ - process noise
- $v$ - observation noise

## Nonlinear

The iterative process is

$$x_k = f(x_{k-1}, u_k, w) \qquad p(w) \sim N(0, Q)$$

$$z_k = h(x_k, v) \qquad p(v) \sim N(0, R)$$

where

- $x$ - unknown state size $n$
- $u$ - input sequence
- $z$ - observations data size $m$
- $w$ - process noise
- $v$ - observation noise

# Kalman filtering process

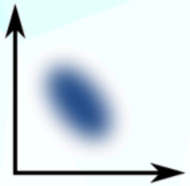
## linear-quadratic estimation (linear system, quadratic cost)

- **estimates unknown variables based on noisy observations which are sequentially acquired**

**Prediction step**

**based on physical model** → $P_{k|k-1}$, $\hat{x}_{k|k-1}$

**Prior knowledge of state**



→ $P_{k-1|k-1}$, $\hat{x}_{k-1|k-1}$

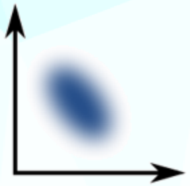
# Kalman filtering process

## linear-quadratic estimation (linear system, quadratic cost)

**Minimizing cost function:**

$$\min_x J = \frac{1}{2}(z_k - Hx)^T R^{-1}(z_k - Hx) + \frac{1}{2}\left(x - \hat{x}_{k|k-1}\right)^T P_{k|k-1}^{-1}\left(x - \hat{x}_{k|k-1}\right)$$

**Prior knowledge of state**



$P_{k-1|k-1}$

$\hat{x}_{k-1|k-1}$

**Prediction step**

**based on physical model**

$P_{k|k-1}$

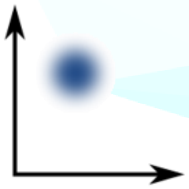
$\hat{x}_{k|k-1}$

**Update step**

**Compare prediction to measurements**

**Measurements**

$z_k$

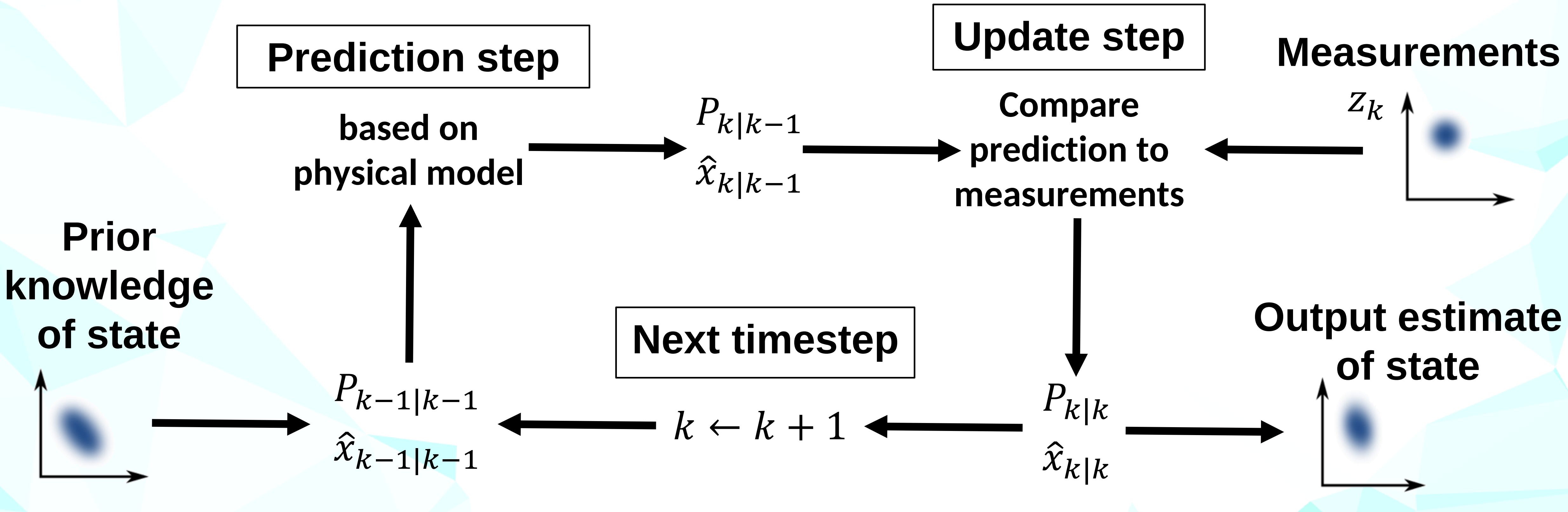


$P_{k|k}$

$\hat{x}_{k|k}$

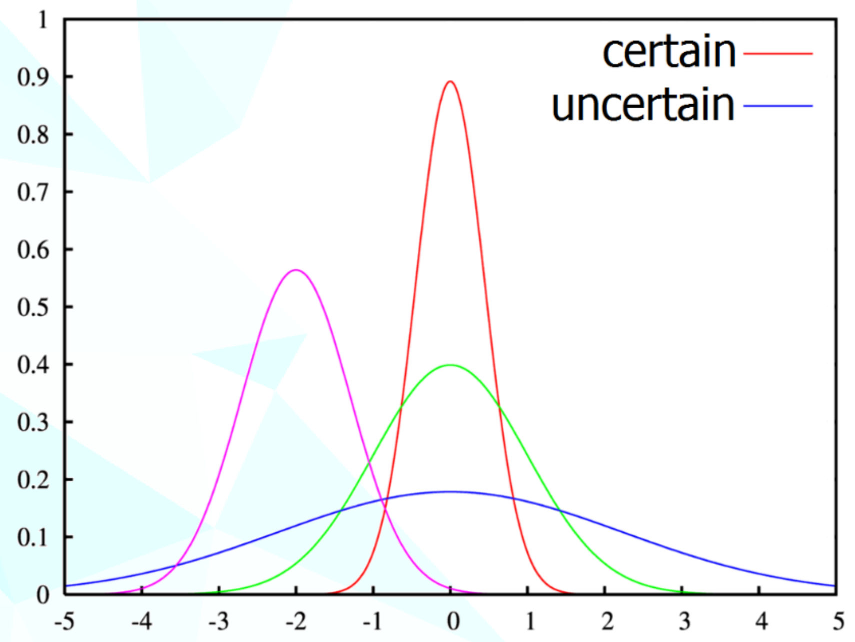
# Kalman filtering process

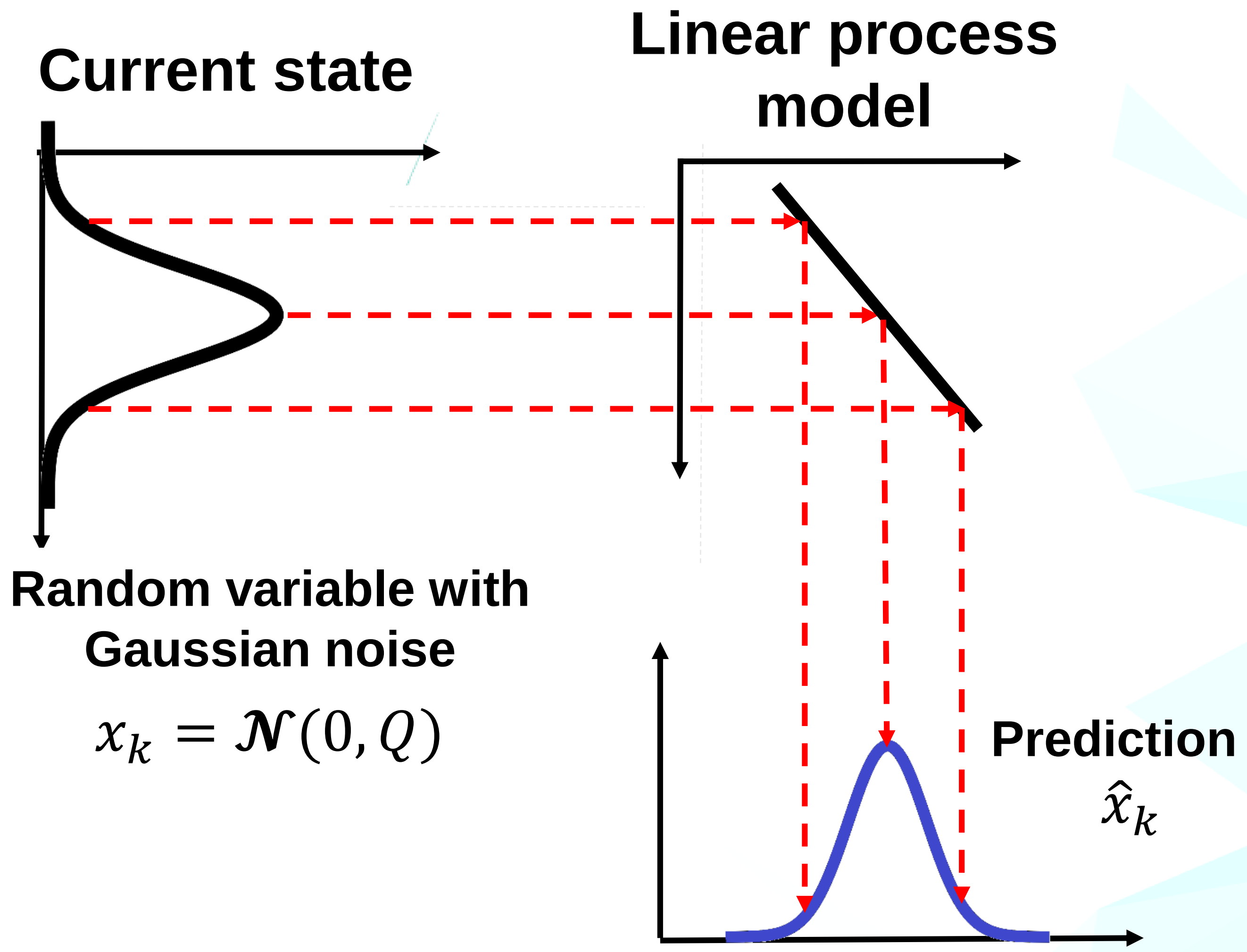
## linear-quadratic estimation (linear system, quadratic cost)



# Distribution Transformation

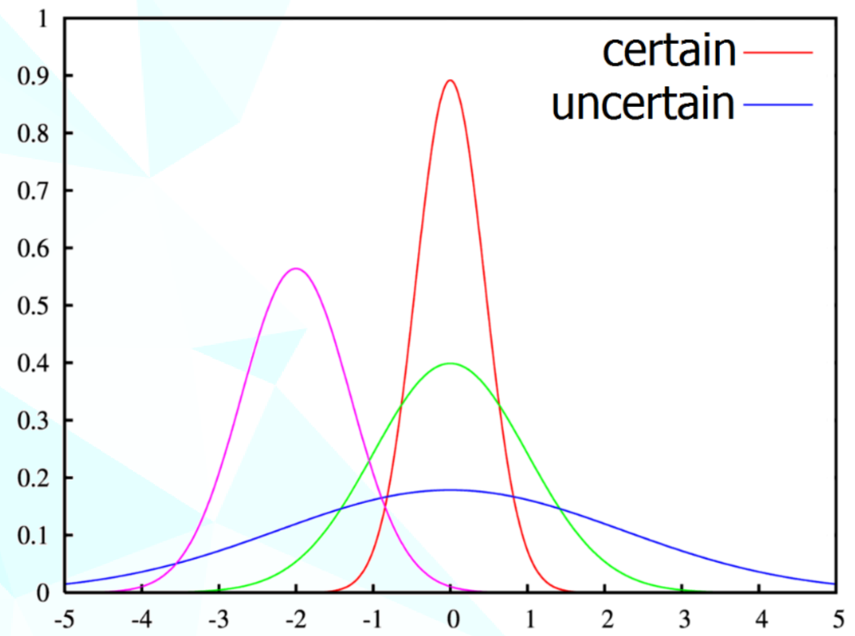
- every unknown is random variable with Gaussian distribution

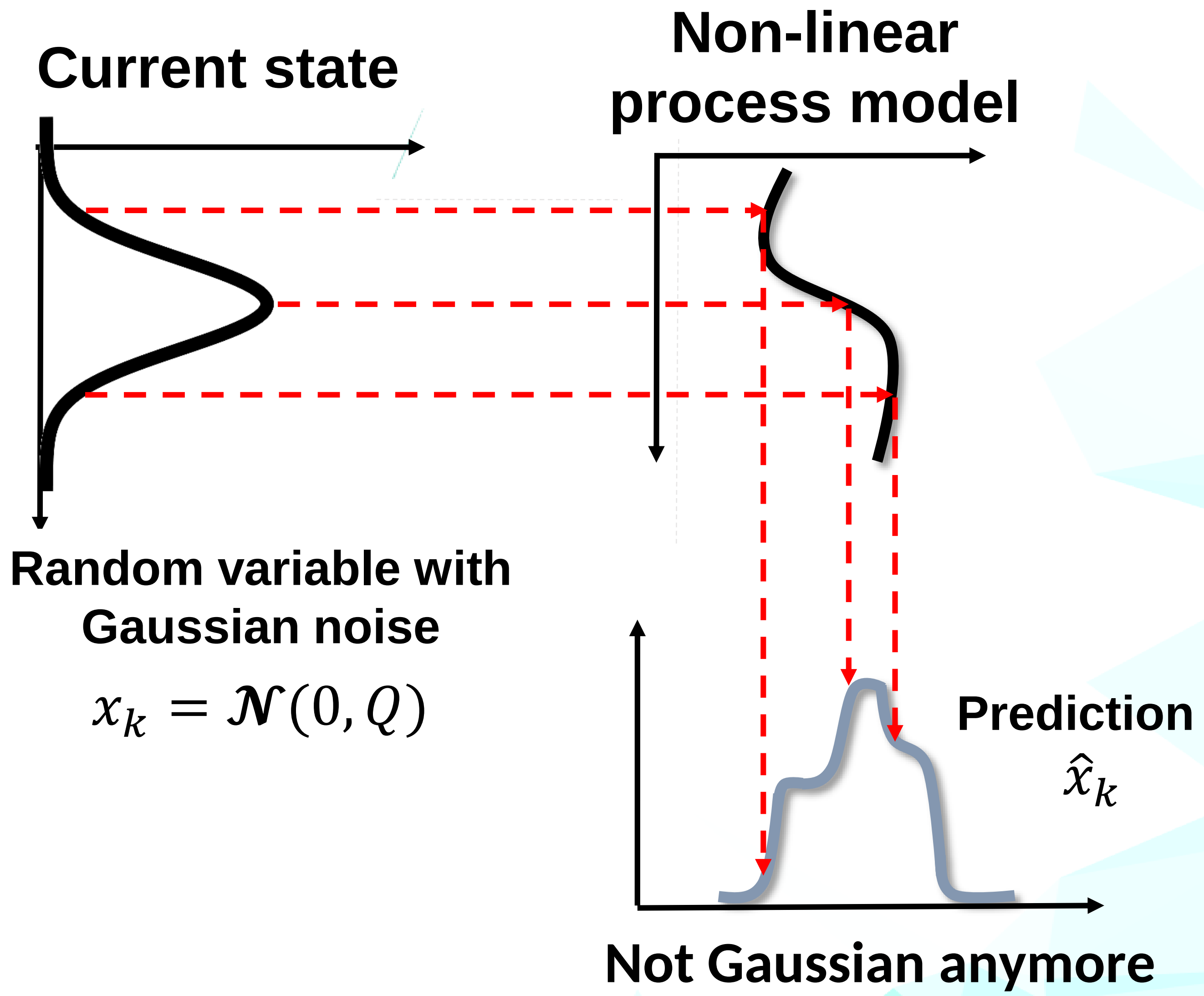


# Distribution Transformation

- every unknown is random variable with Gaussian distribution



# Nonlinear Kalman filters

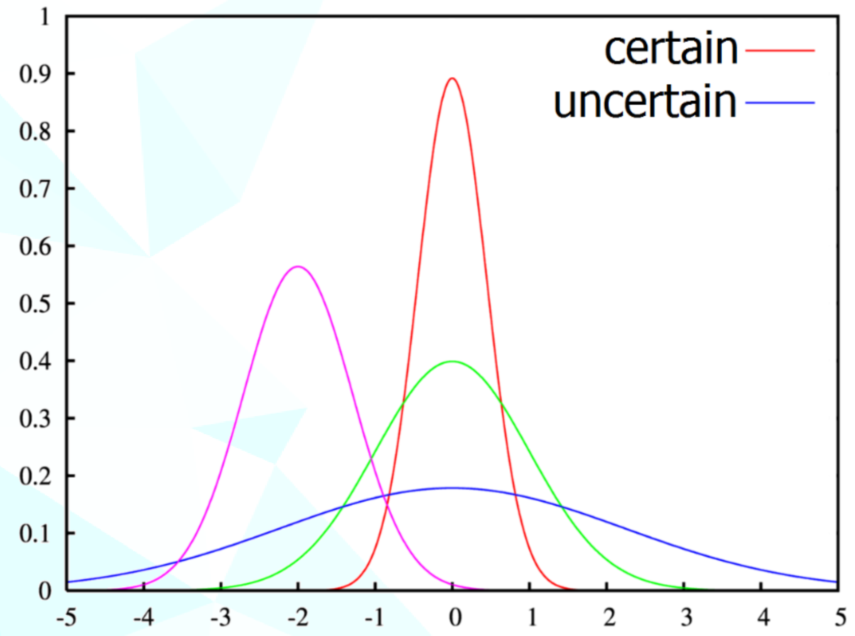
- **Transformation (simulation step) system linearization (EKF)**
  - **Main issue – system complexity**
    1. **Multidimensional derivative (positions, {velocities}, material parameters, spring stiffness, {contact parameters})**
    2. **Iterative solvers processing**
    3. **Approximation accuracy (second order derivative)**
- **Probabilistic distribution discretization (UKF, EnKF)**

  **It is easier to approximate a known Gaussian distribution than an arbitrary nonlinear function/transformation**

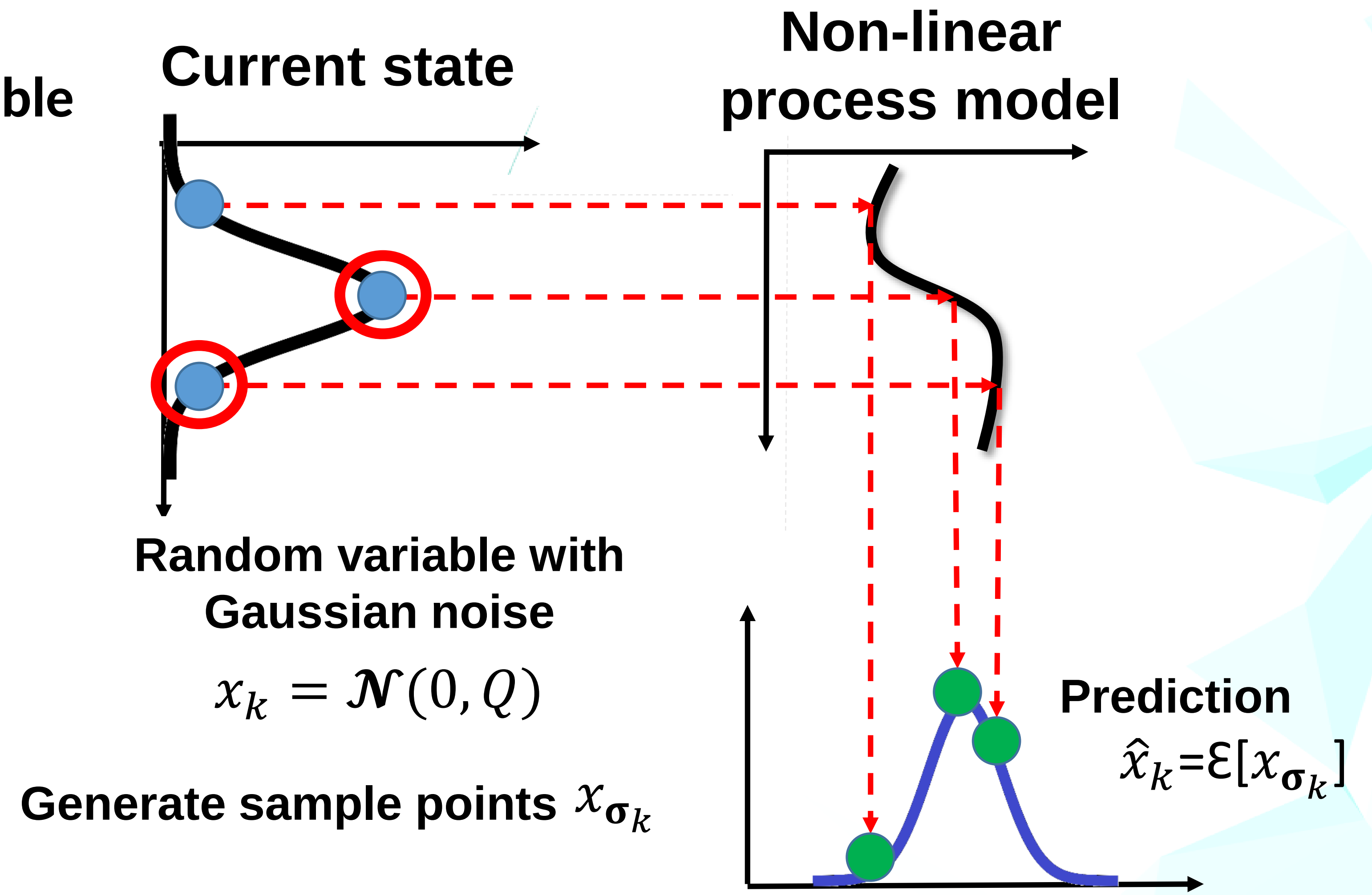
  - **Main issue – computation time**

# Transformation of Uncertainty

- every unknown is random variable with Gaussian distribution




**Prediction step takes 85-95 % of the whole time**



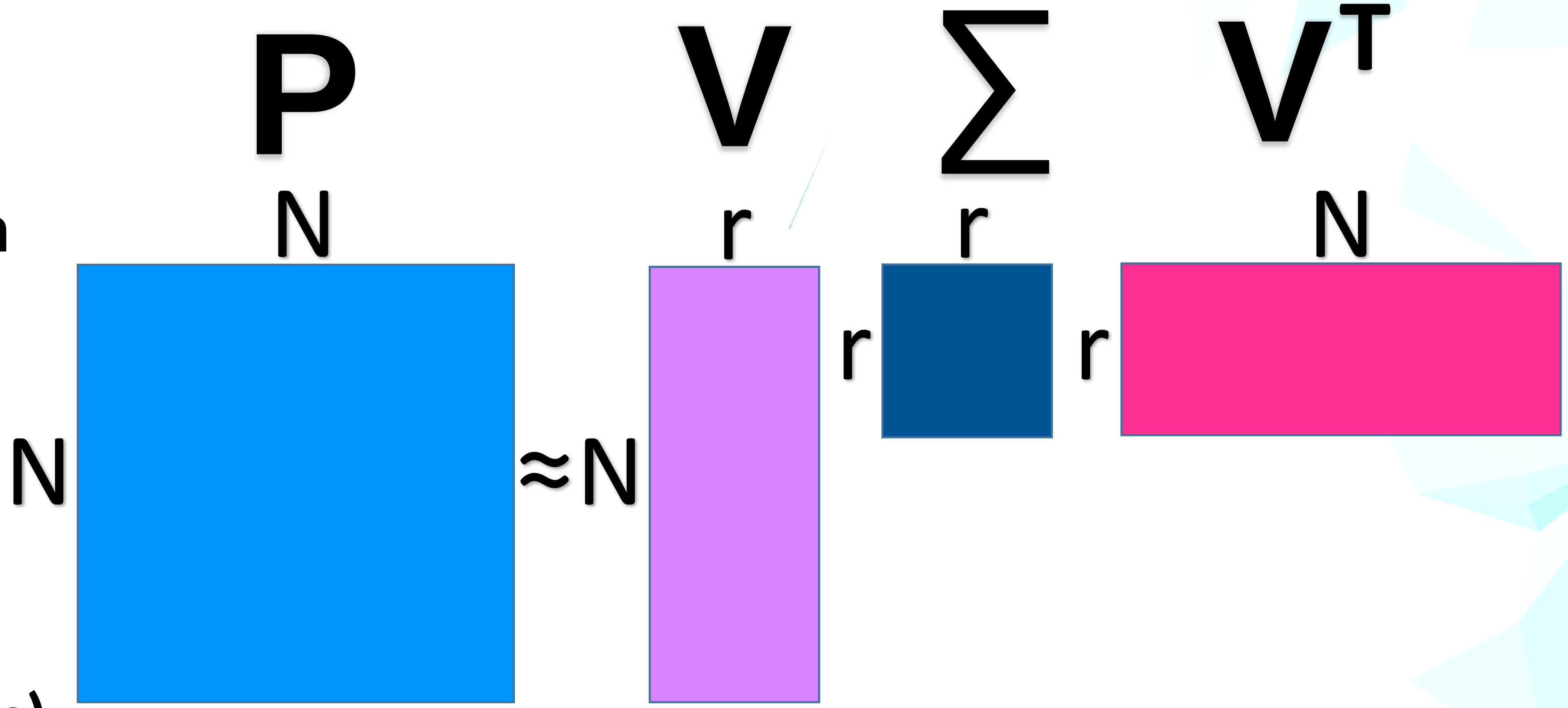

**at least *N+1* sample points for *N*-dimensional space**

# Order reduction

1. **System model reduction**
   - + simple model (coarse mesh),
   - – POD, wavelets, …
2. **Covariance matrix reduction**
   - + simple model (coarse mesh), SVD, EOF, Monte-Carlo estimation
   - – factorization, multiscale analysis,
3. **Filtering loop improvement**
   - + EnTKF
   - – wavelet rank reduction, Singular Evolutive Extended Kalman Filter – replace analysis matrix by a lower rank approximation

# Covariance matrix reduction

- Singular value decomposition (SVD)

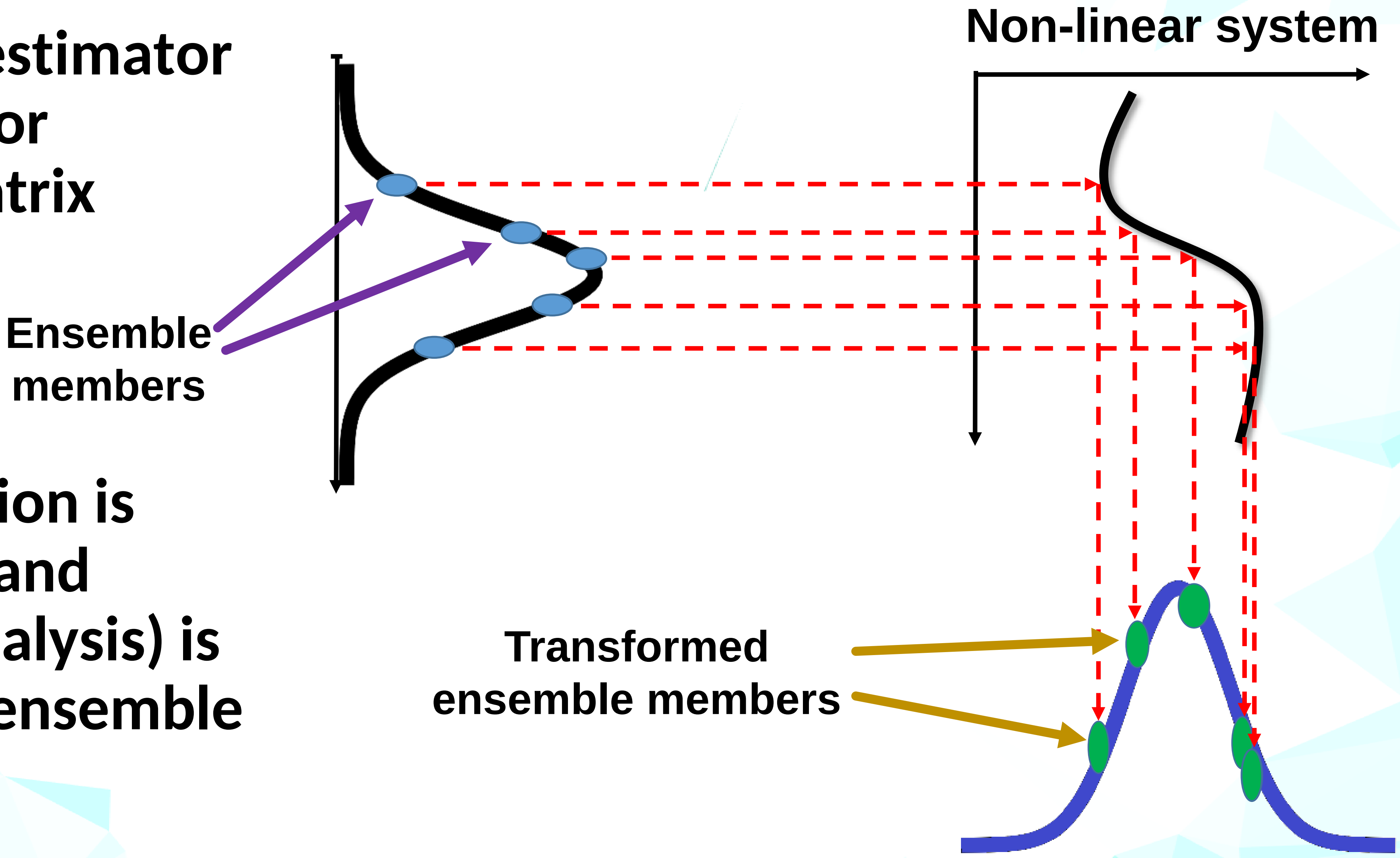



- Empirical orthogonal functions (EOFs)
  - The basic functions are chosen to be different from each other and to account for as much variance as possible

# Ensemble Transform Kalman filter (Idea)

- Monte-Karlo estimator (integration) for covariance matrix

- The cost function is reformulated and Correction (analysis) is performed in ensemble subspace

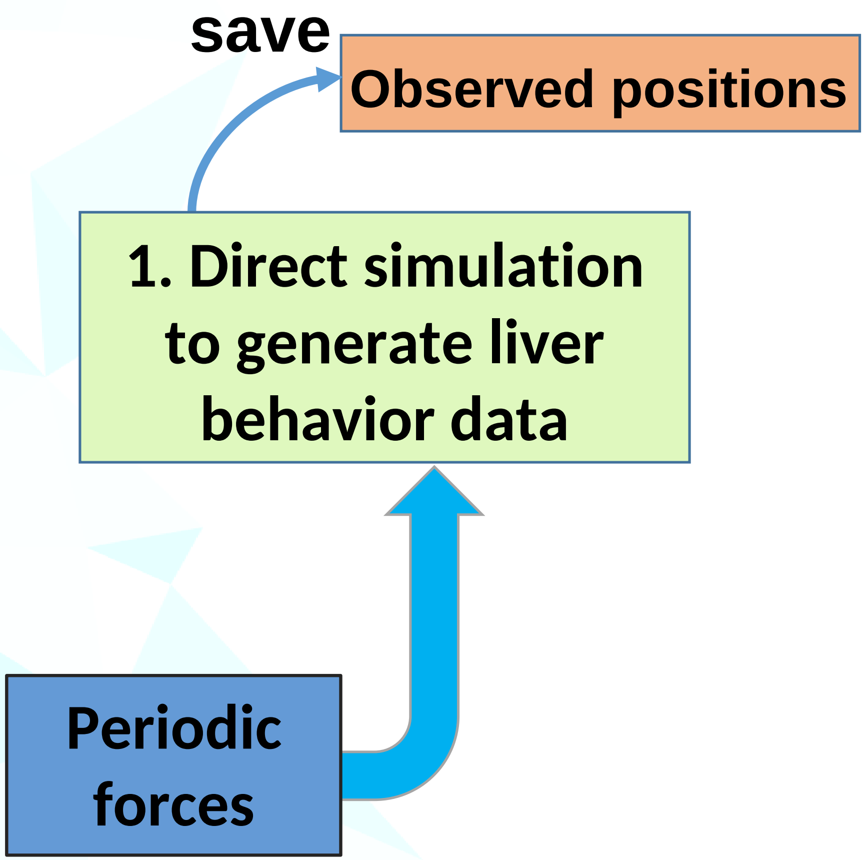


# Ensemble Transform Kalman filter (Tips and tricks)

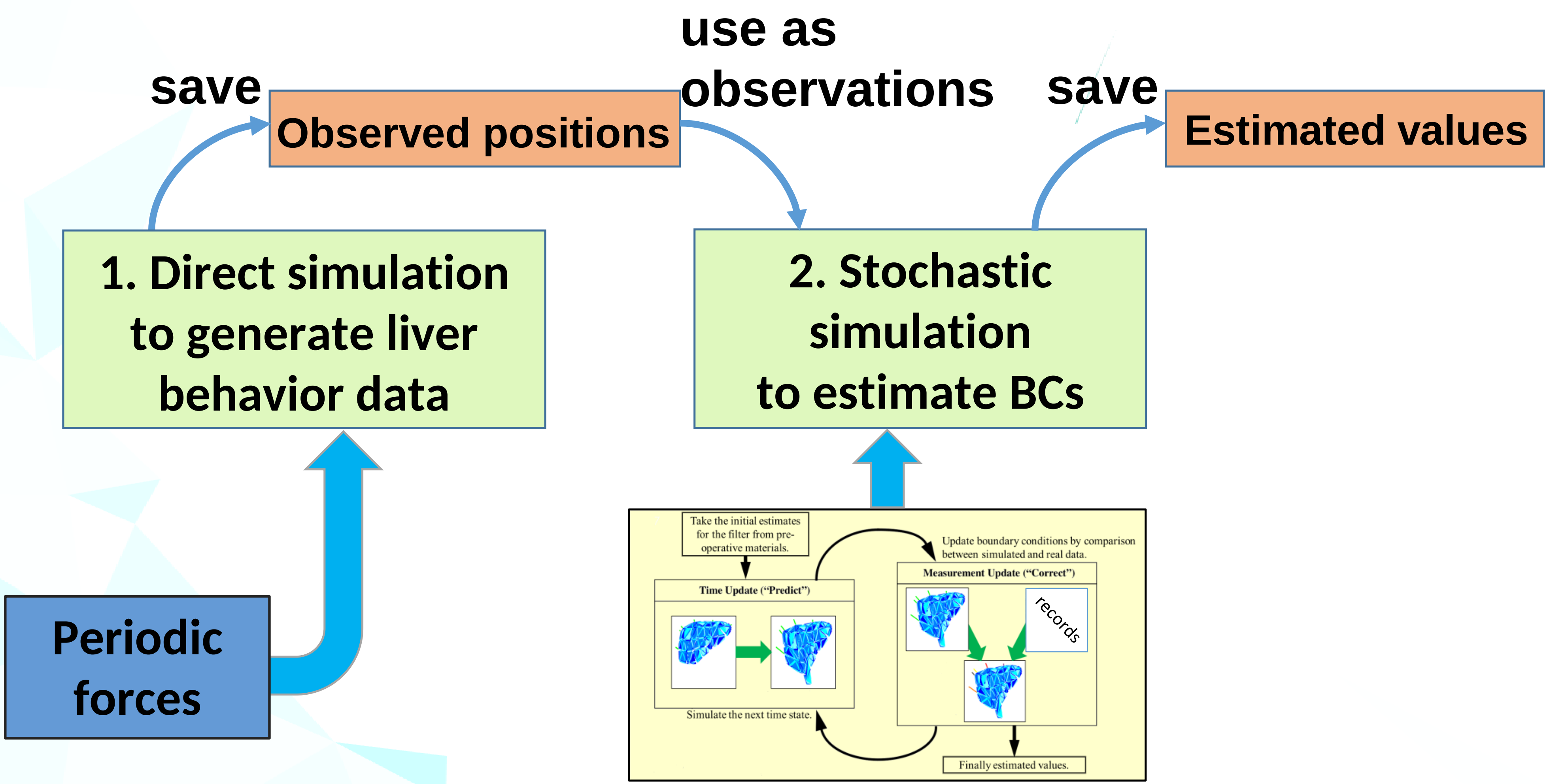
- **Localization – to approximate covariance more precisely**
  - **Take observations close to estimated unknowns**
  - **All observations are local ??**

- **Inflation – to prevent filter divergence**
  - **Equivalent to process noise addition to increase uncertainty**

# Experimental setup



# Experimental setup



# Experimental setup

original values

save

Observed positions

use as observations

save

Estimated values

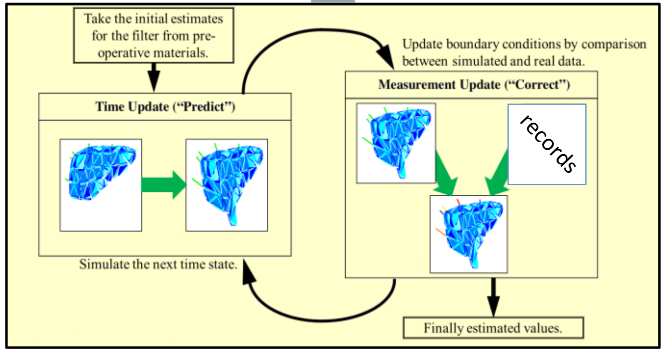
use for simulation

1. Direct simulation to generate liver behavior data

2. Stochastic simulation to estimate BCs

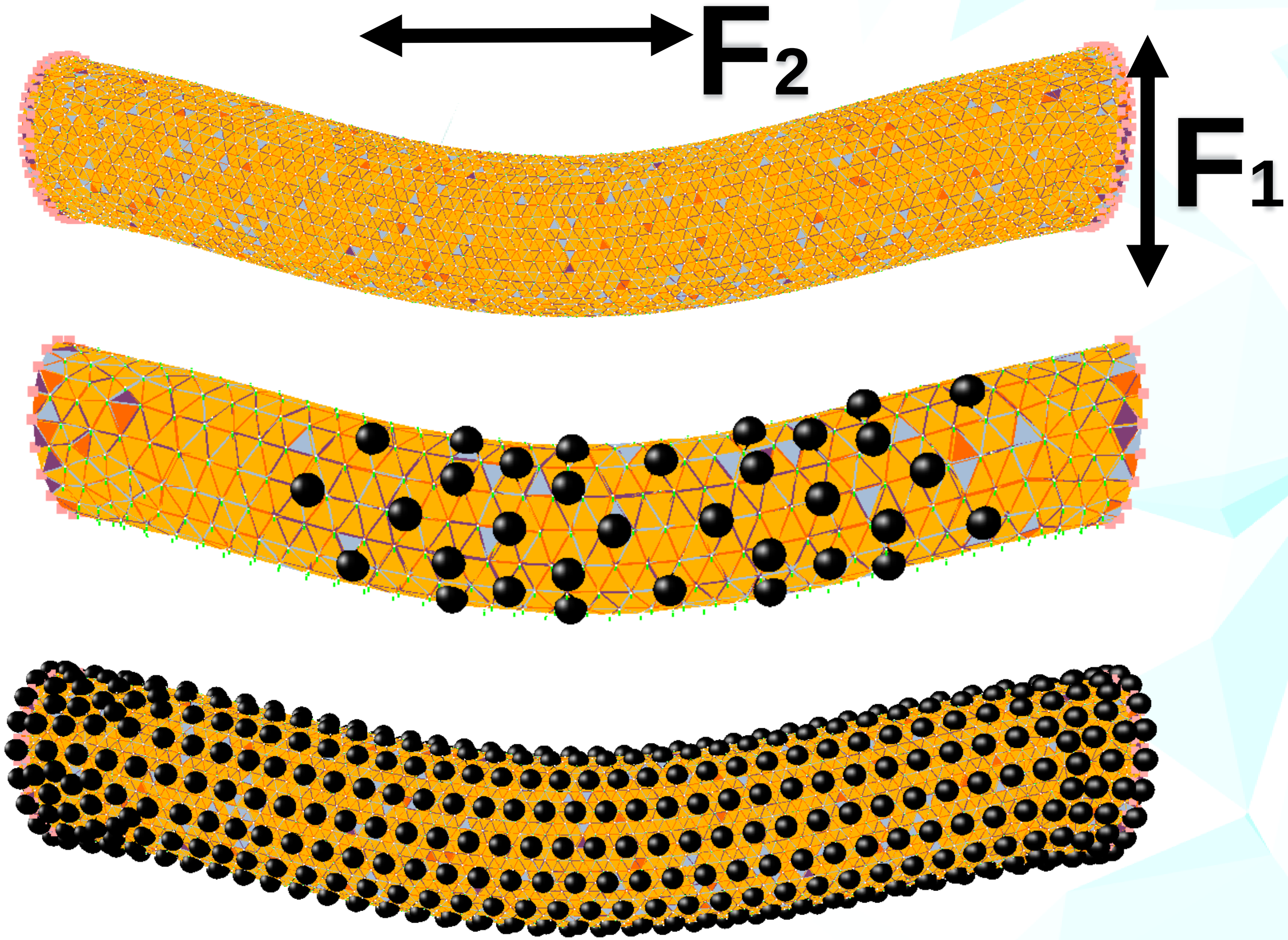
3. Direct simulation to estimate the difference

Periodic forces

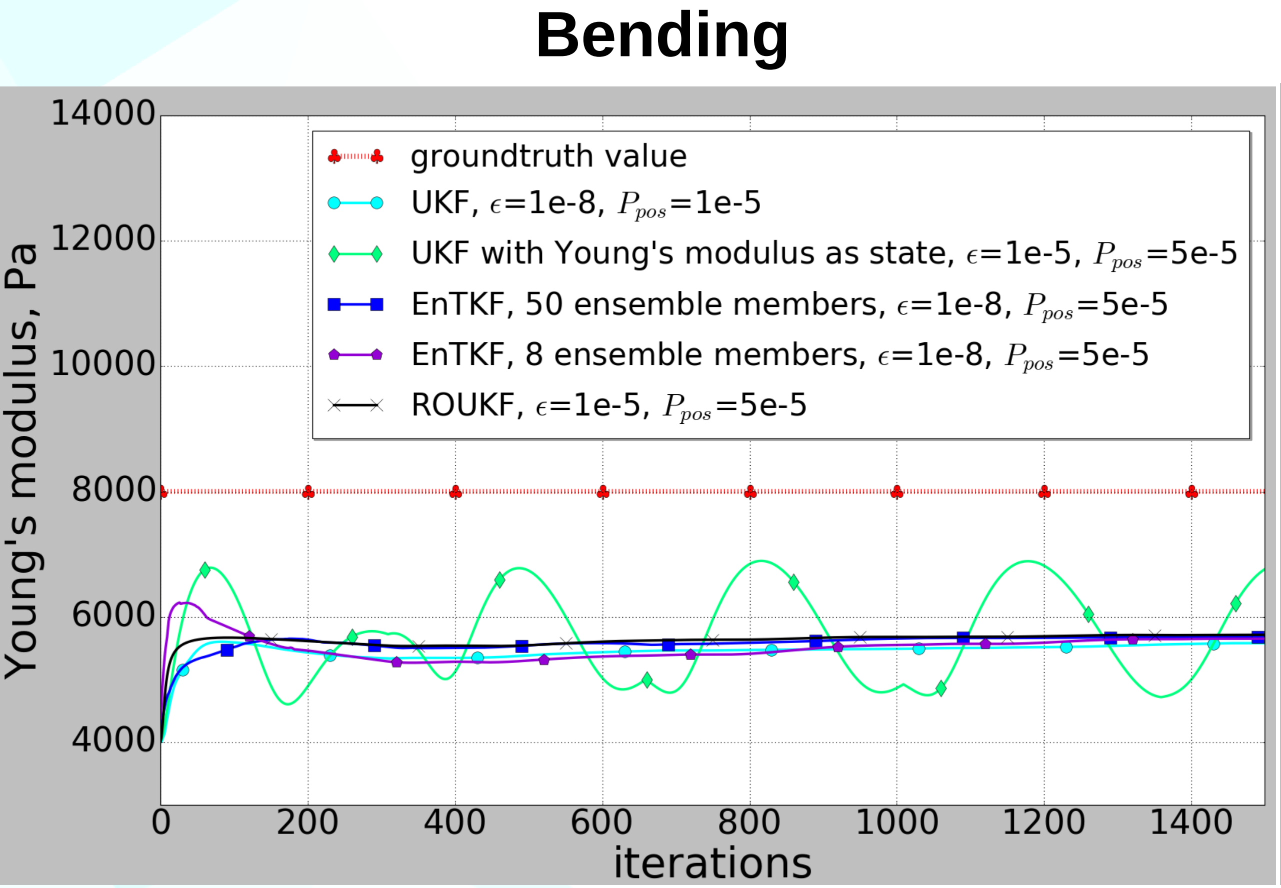


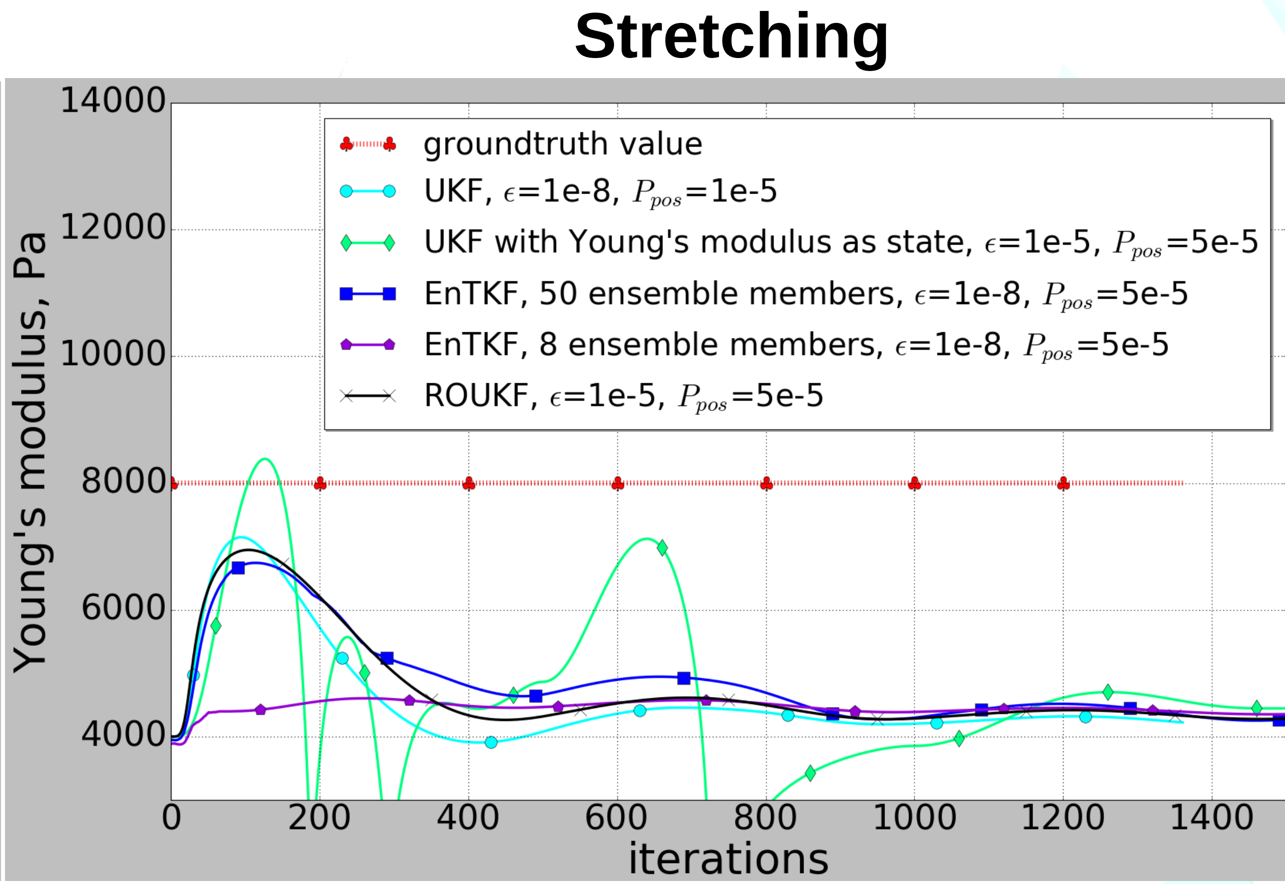
# Cylinder Experiment Description

- Groundtruth model: 3cm diam. by 18 cm approx. 52000 el., NeoHookean material 8000 Pa, Poisson 0.49
- Periodic forces to simulate impact. First scenario bending, second scenario stretching
- Observed data – 33 surface markers
- DA model: 3cm diam. by 18 cm approx. 3000 el., StVK material
- Init modulus 4000 Pa, std 350
- Compare points of coarse mesh



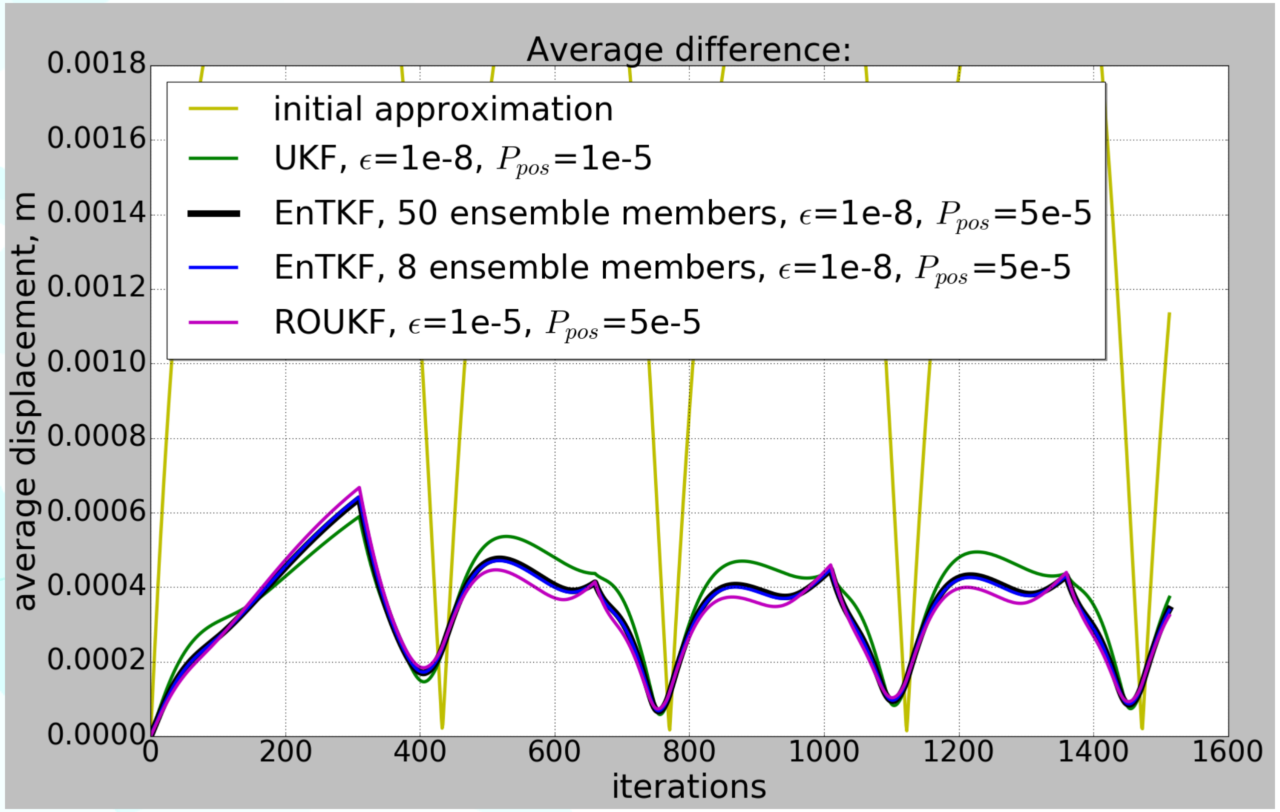
# Cylinder Experiment Results



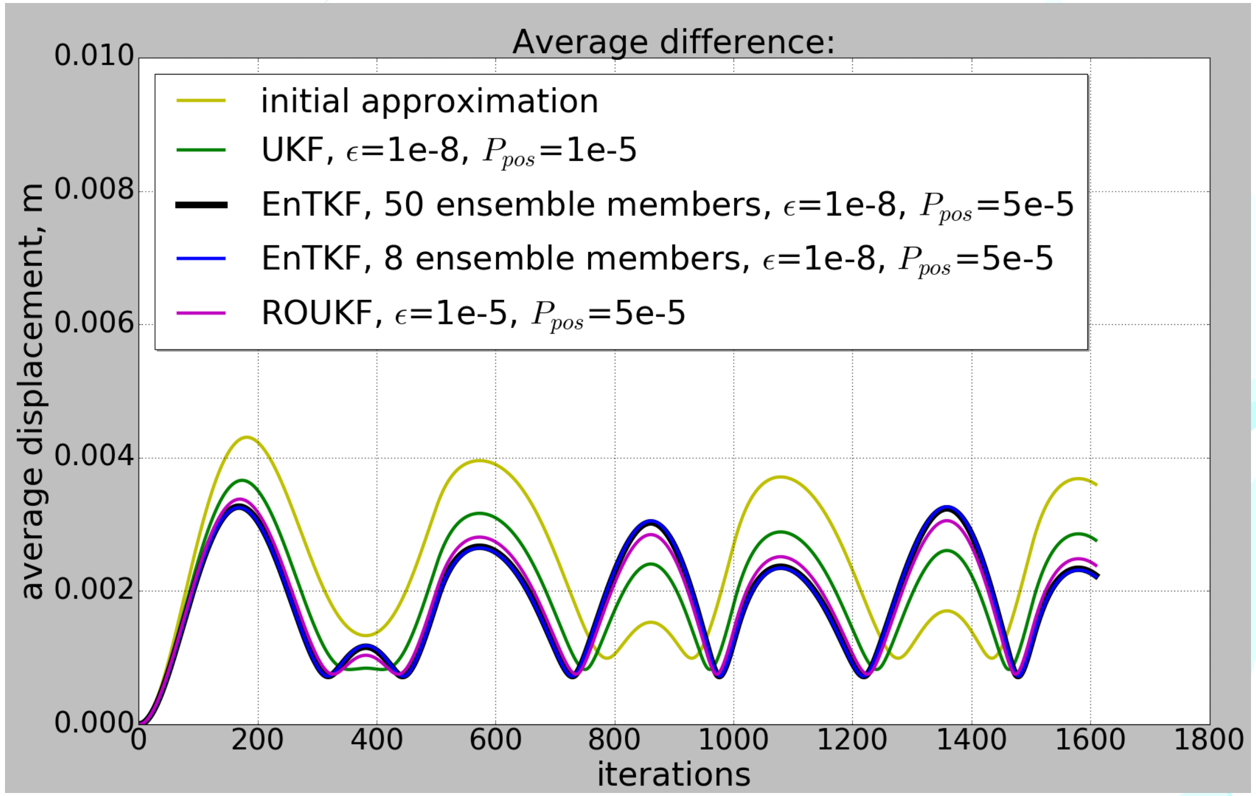



# Cylinder Experiment Validation Results (1)

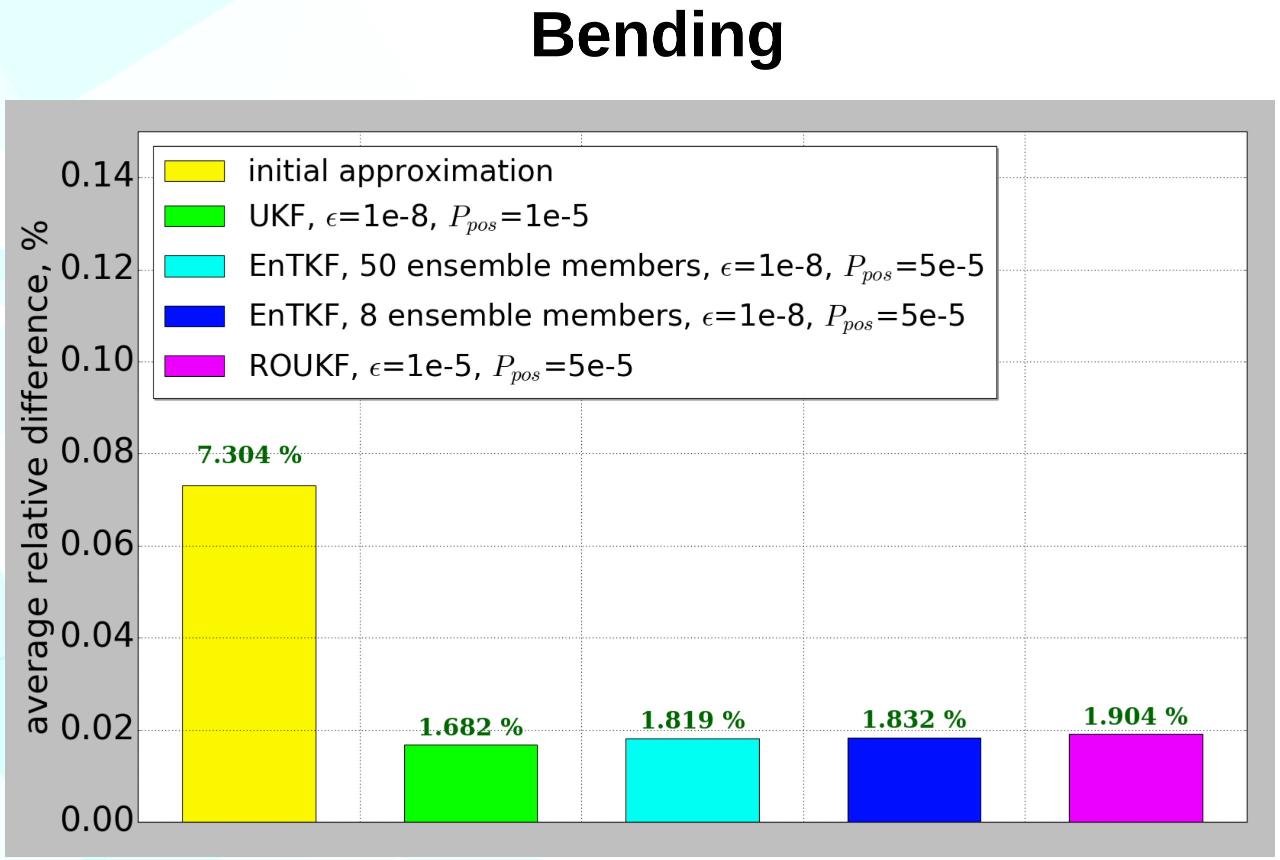
## Bending

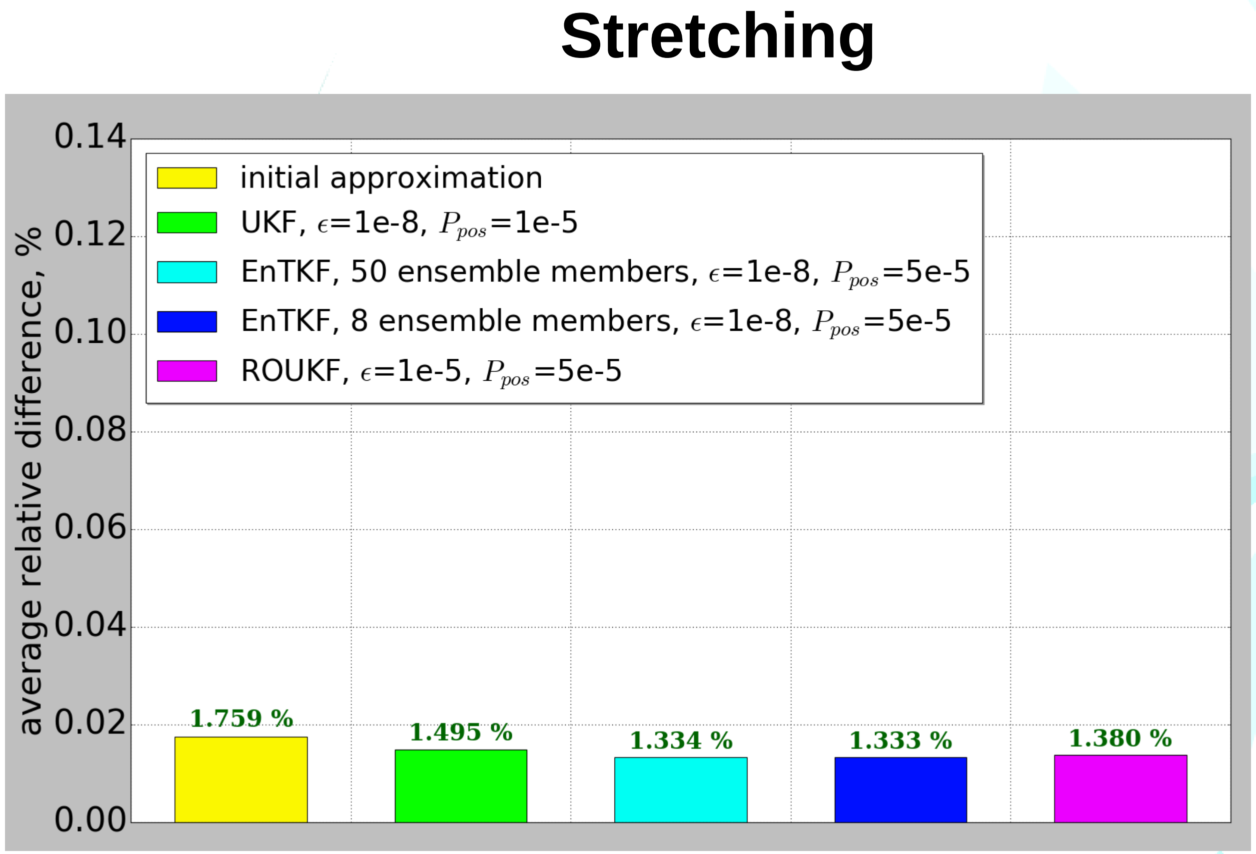



## Stretching



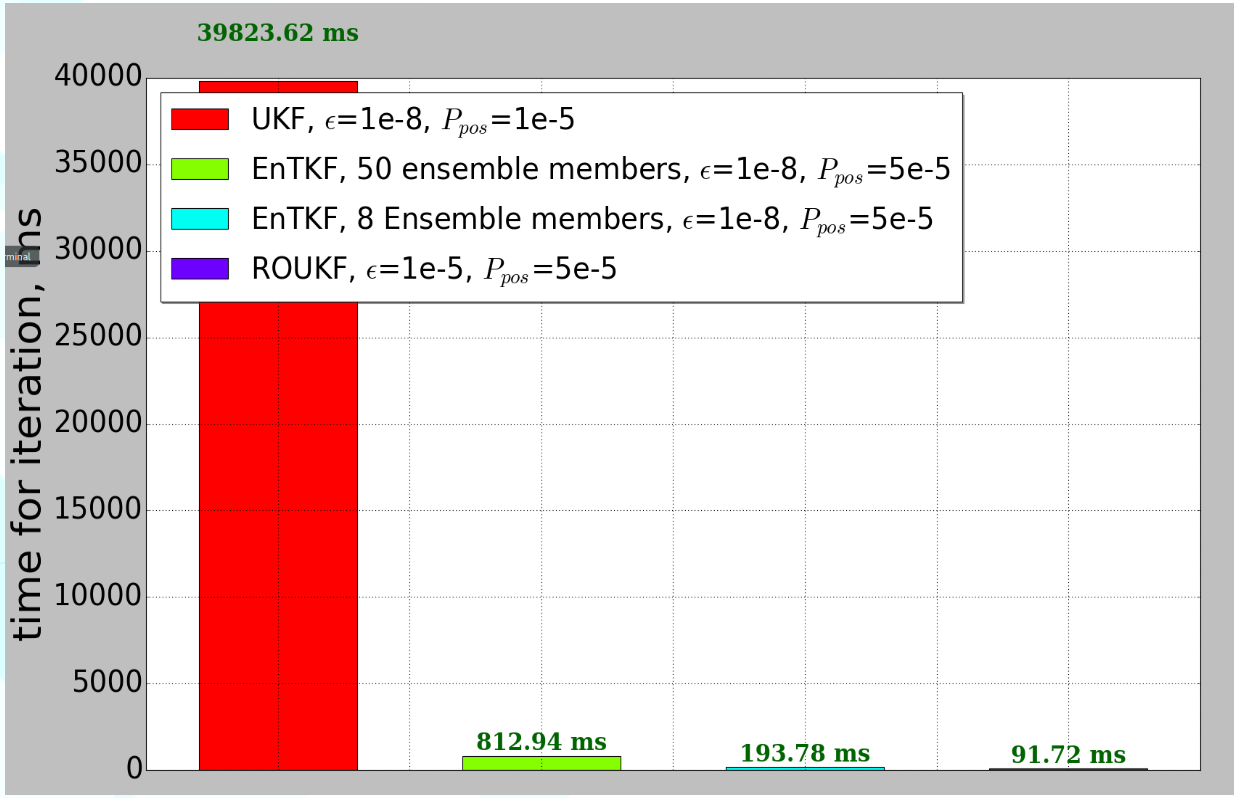
# Cylinder Experiment Validation Results (2)



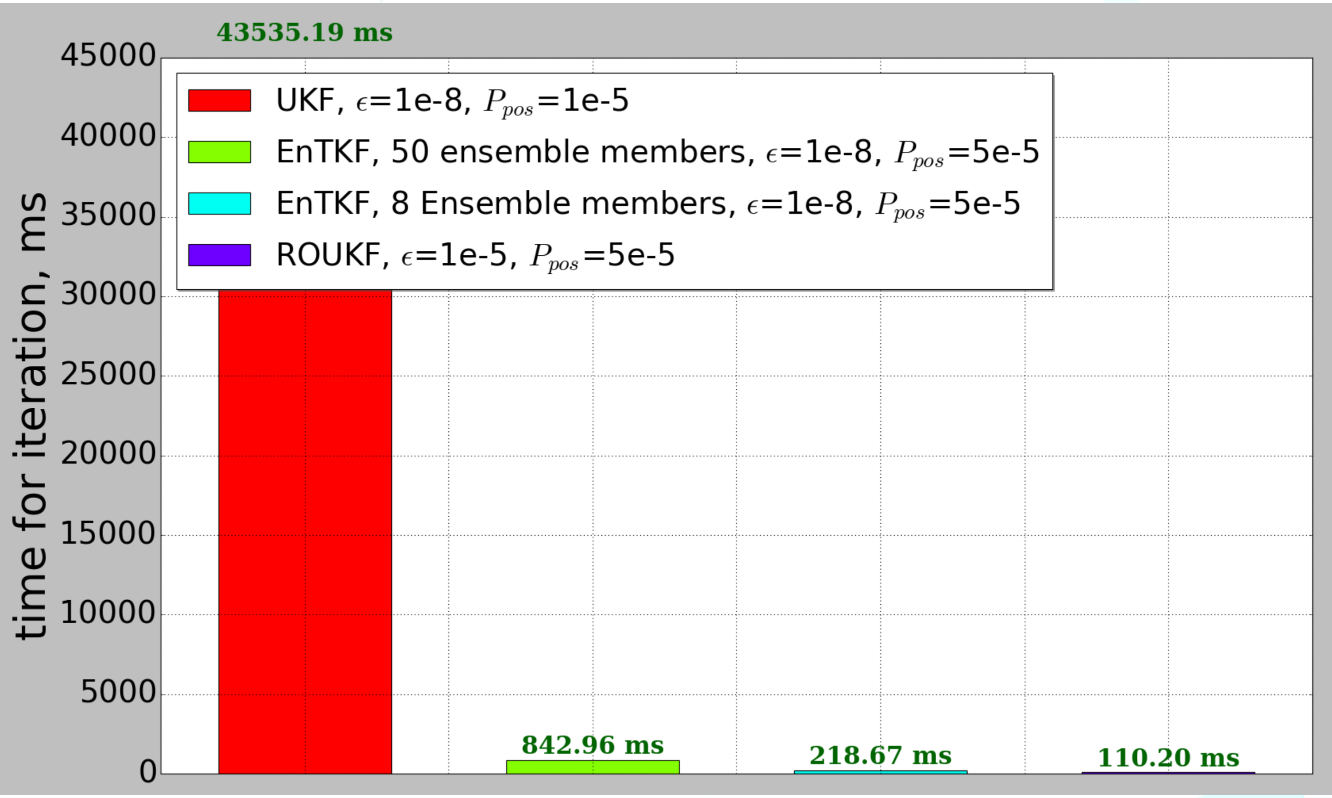



# Cylinder Experiment Performance Results

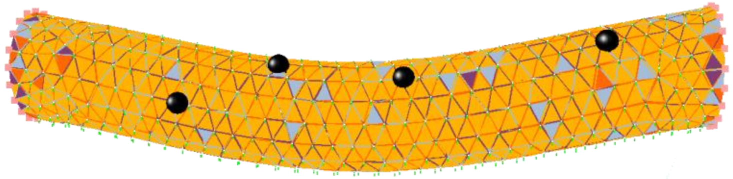
**Bending**




**Stretching**

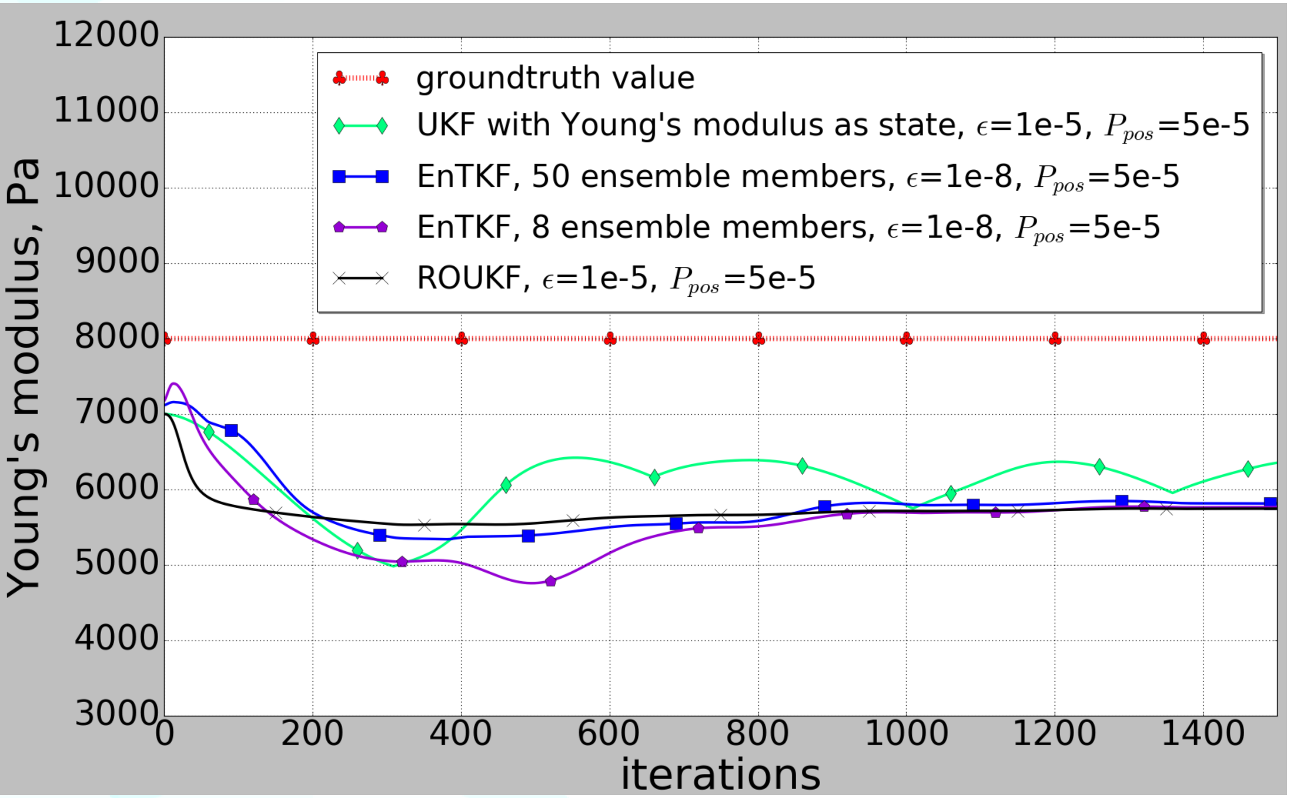


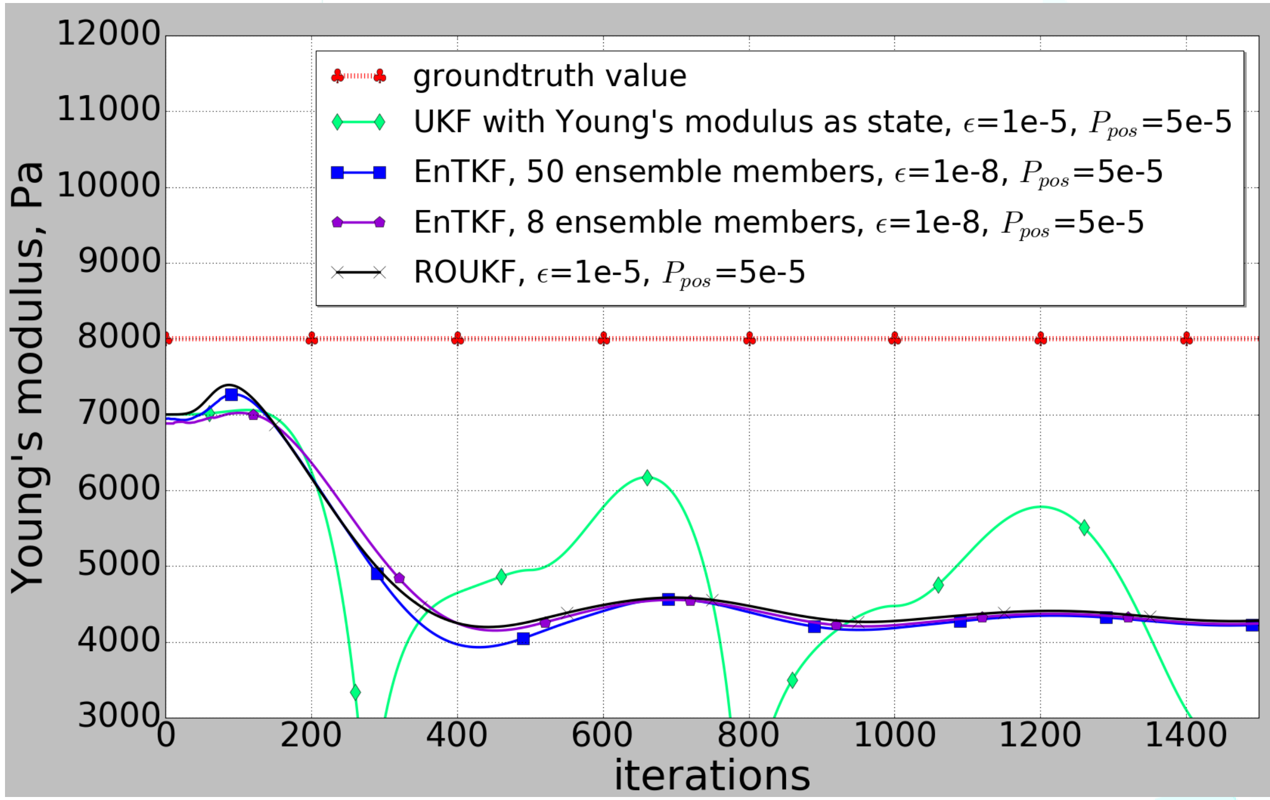
# Cylinder Experiment Results



**Bending**

**Stretching**

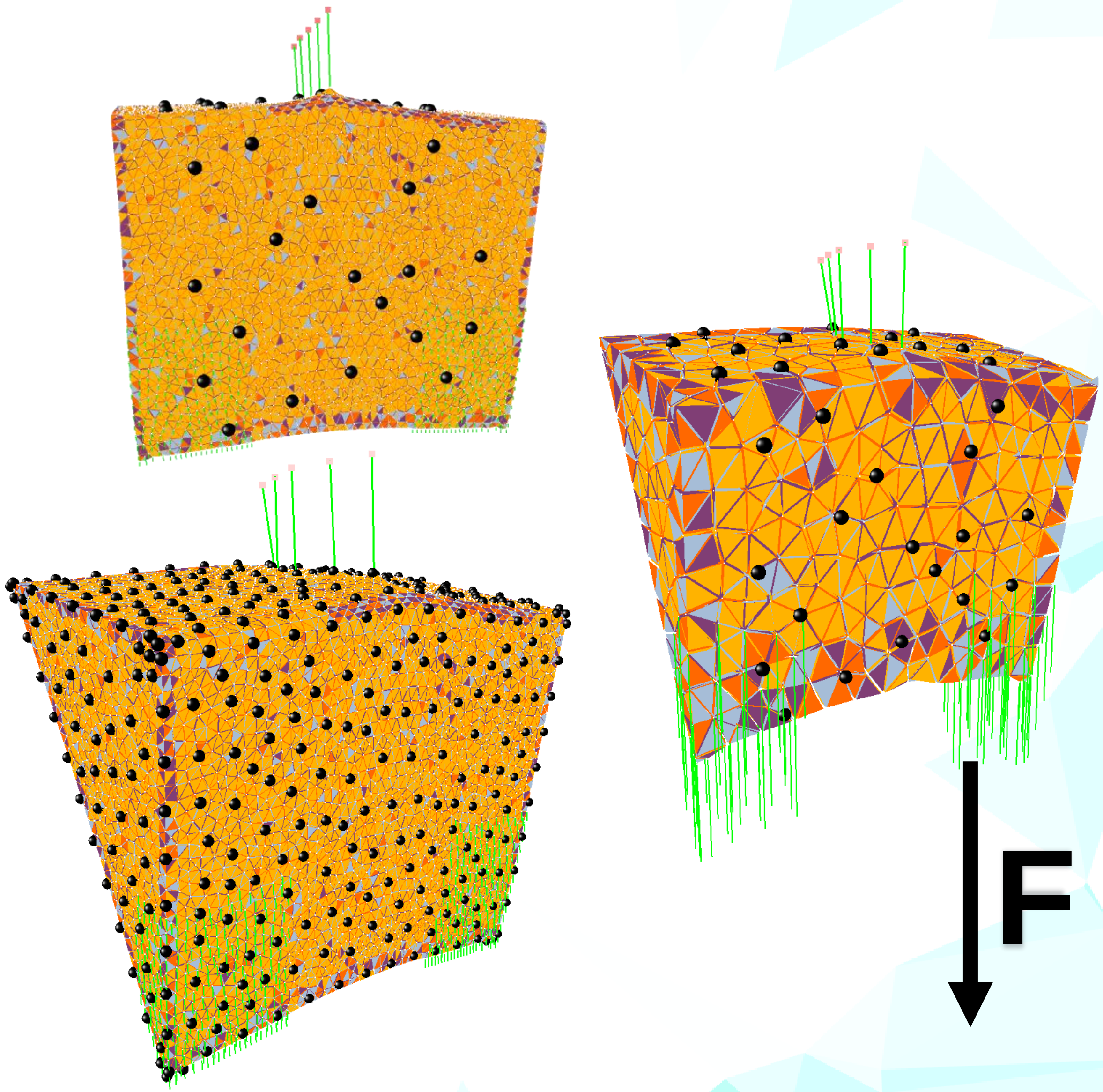


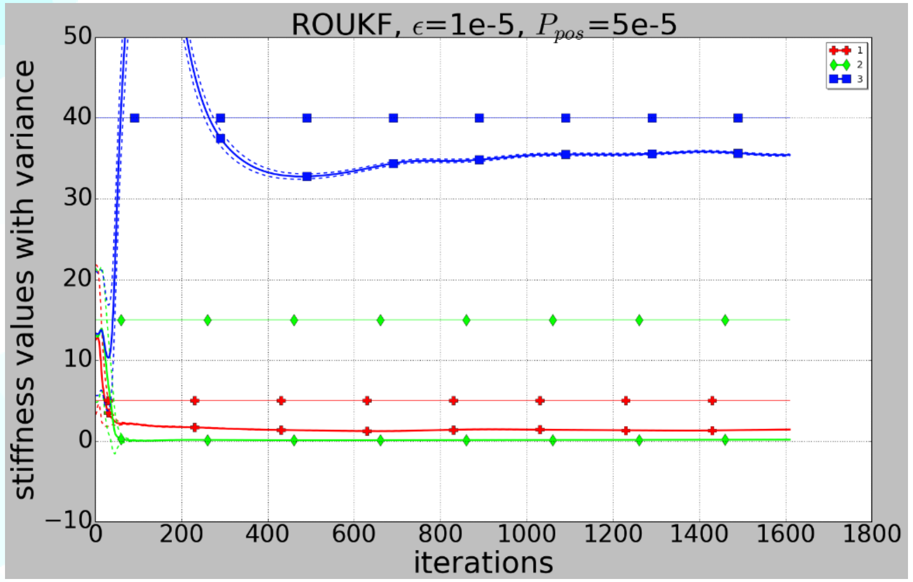

- **Observed data – 4 surface markers**
- **Another Init modulus 7000 Pa**

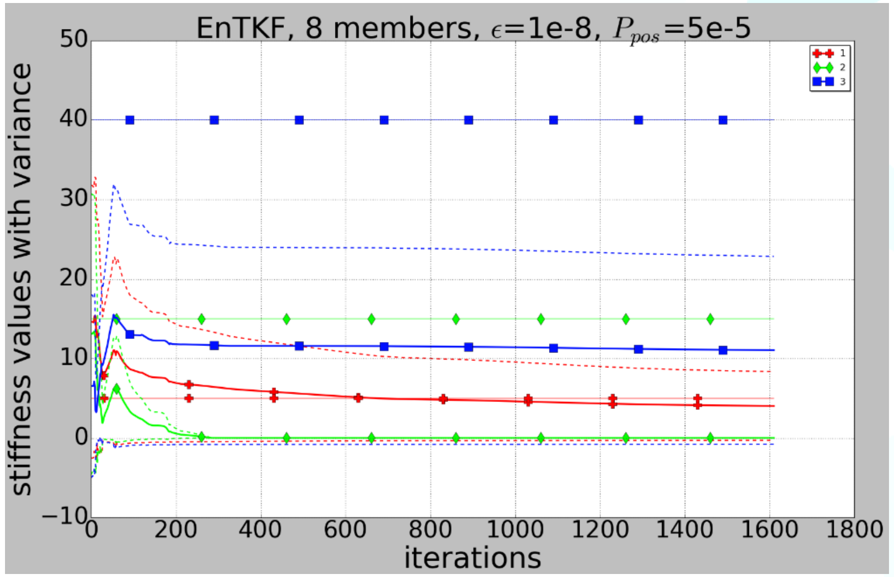
# Boundary conditions Experiment Description

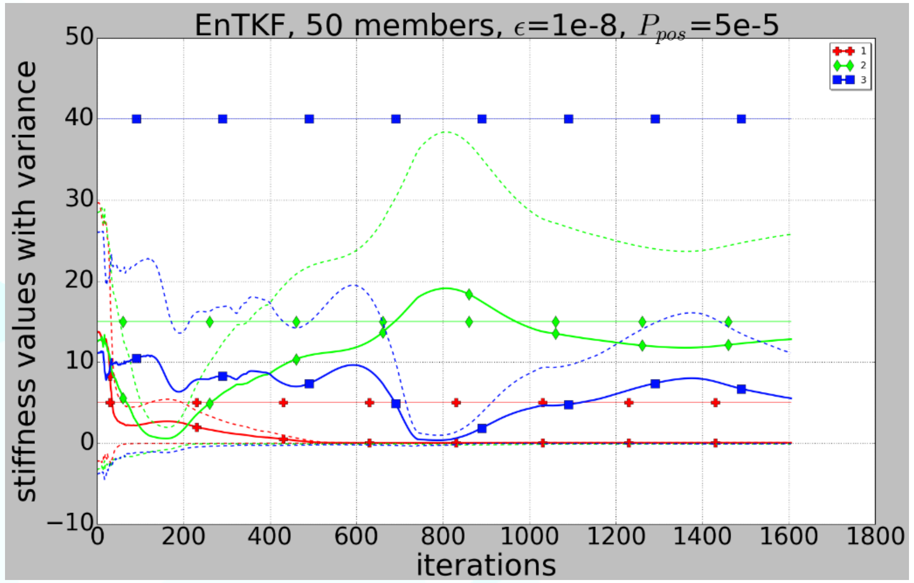
- **Groundtruth model: triangular prism 10cm by 14 cm by 18 cm, approx. 55000 el., StVK material 8000 Pa, Poisson 0.49**
- **Periodic forces to simulate impact.**
- **Several springs with shared parameters to simulate BCs**
- **Observed data – observe anterior and superior surfaces in first scenario and only superior in the second one**
- **DA model 10cm by 14 cm by 18 cm, approx. 2200 el., StVK material**
- **Compare points of coarse mesh**



# BCs Experiment Results

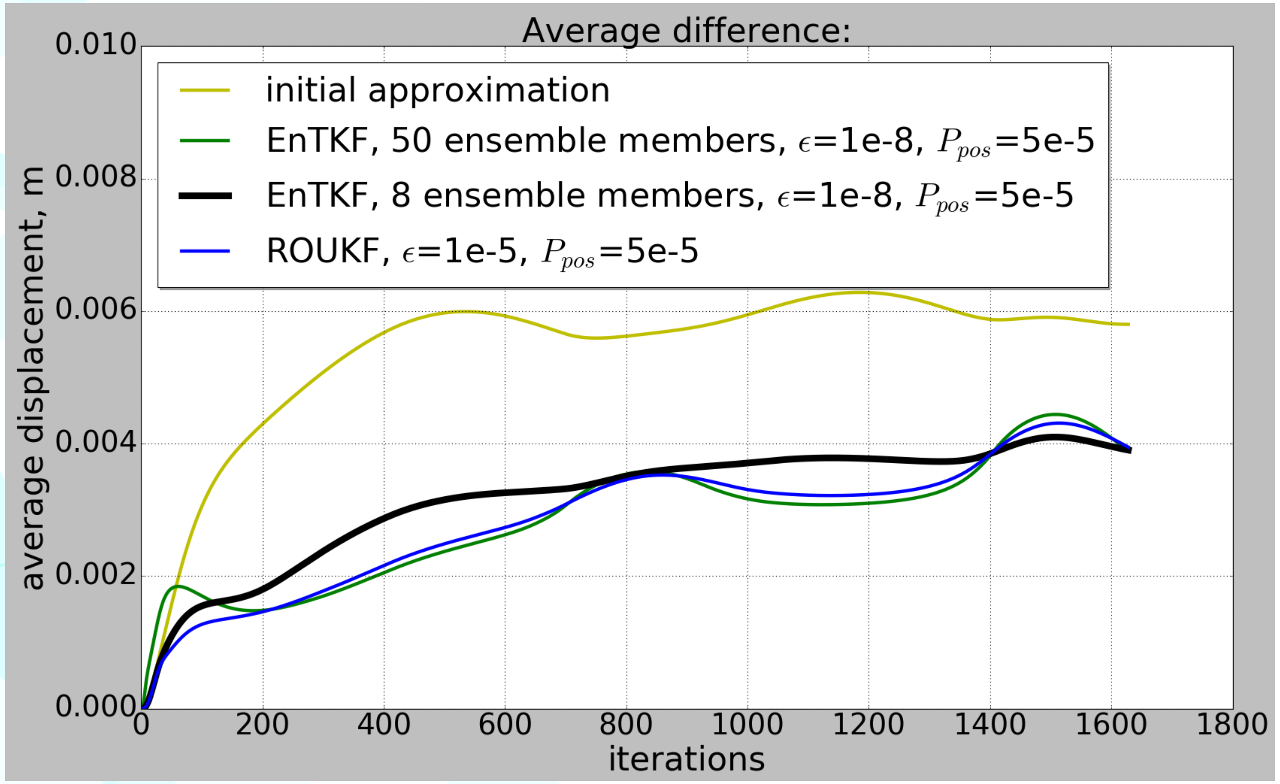


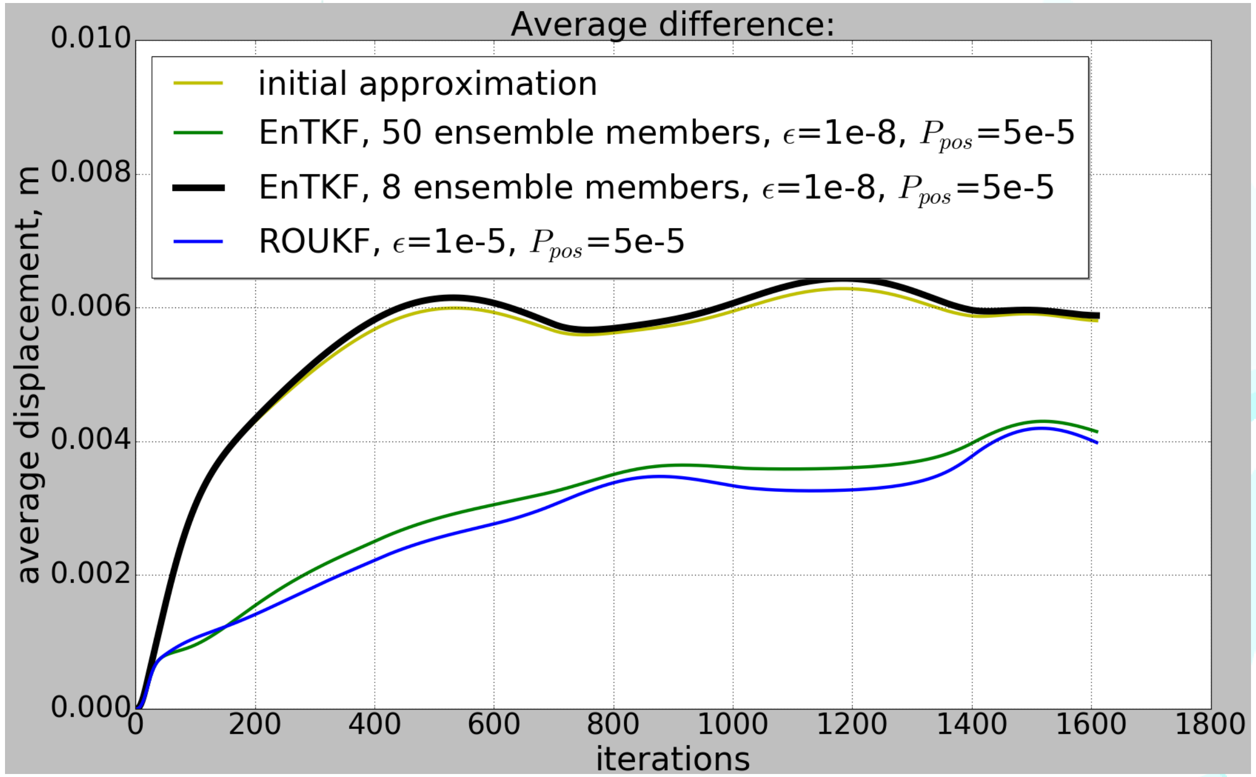



# BCs Experiment Validation Results (1)

## Anterior and superior

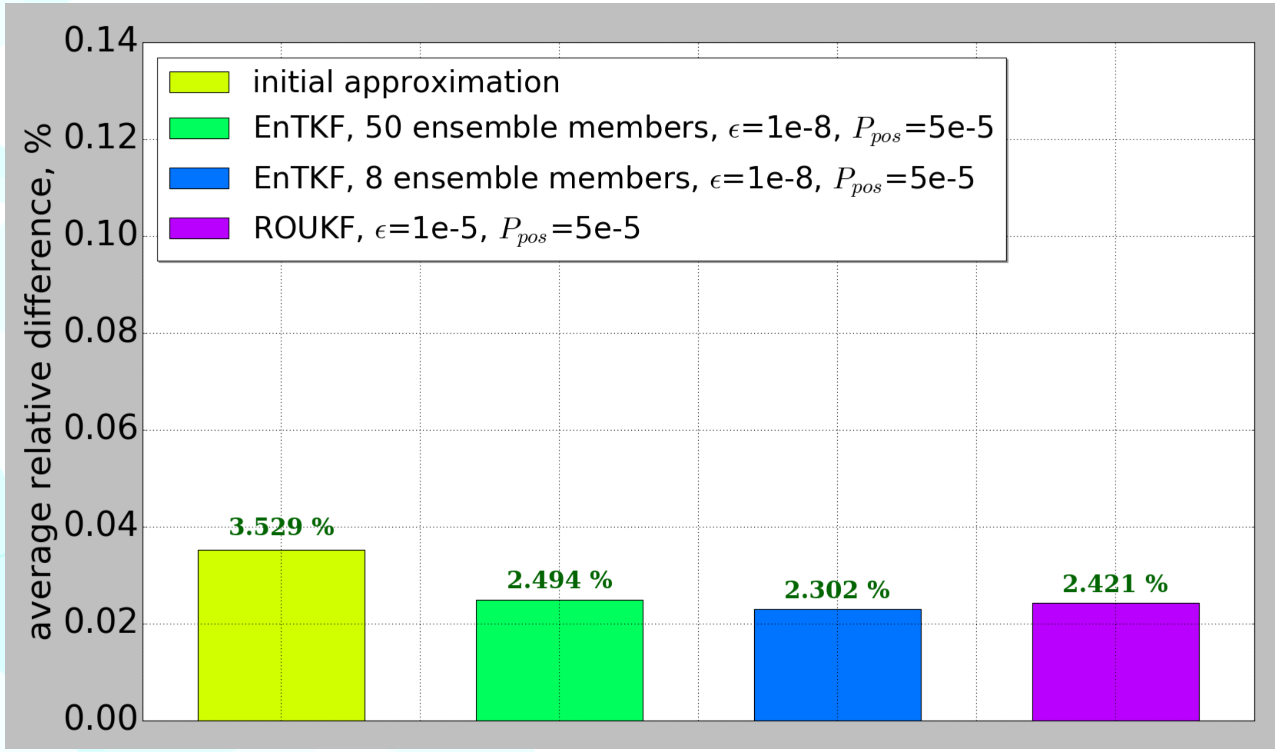



## Superior

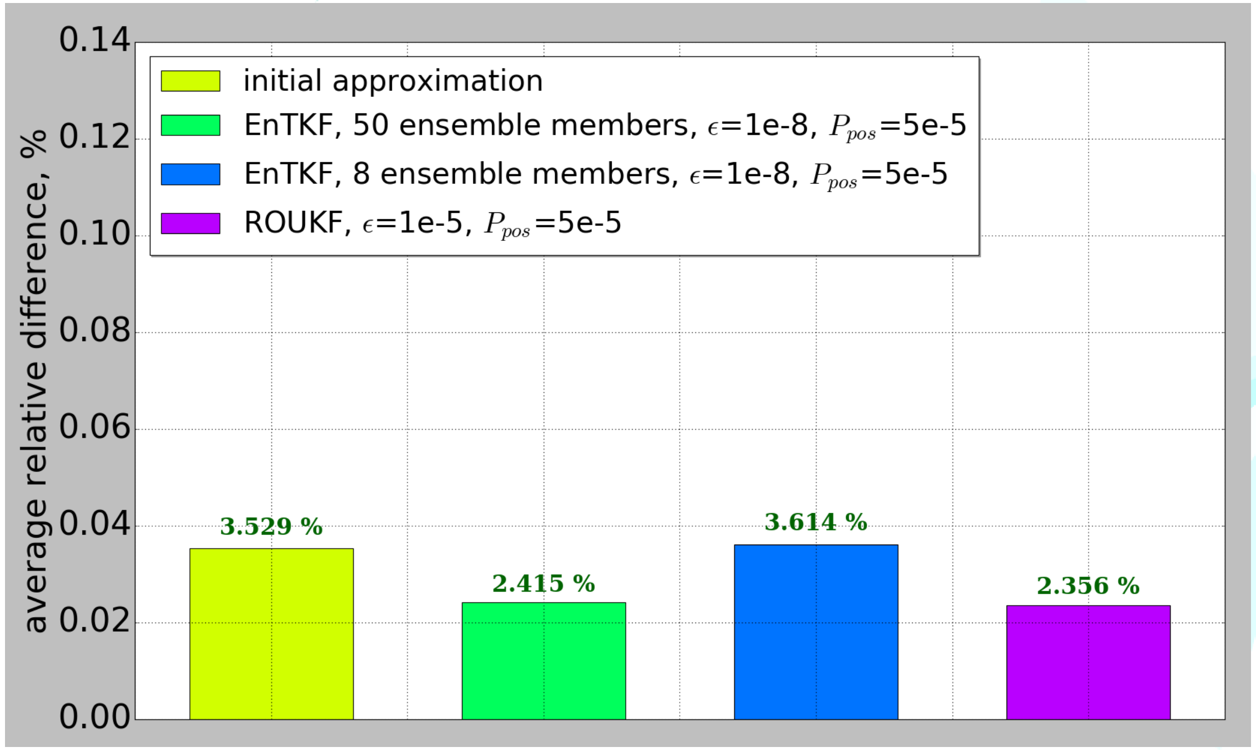


# BCs Experiment Validation Results (2)

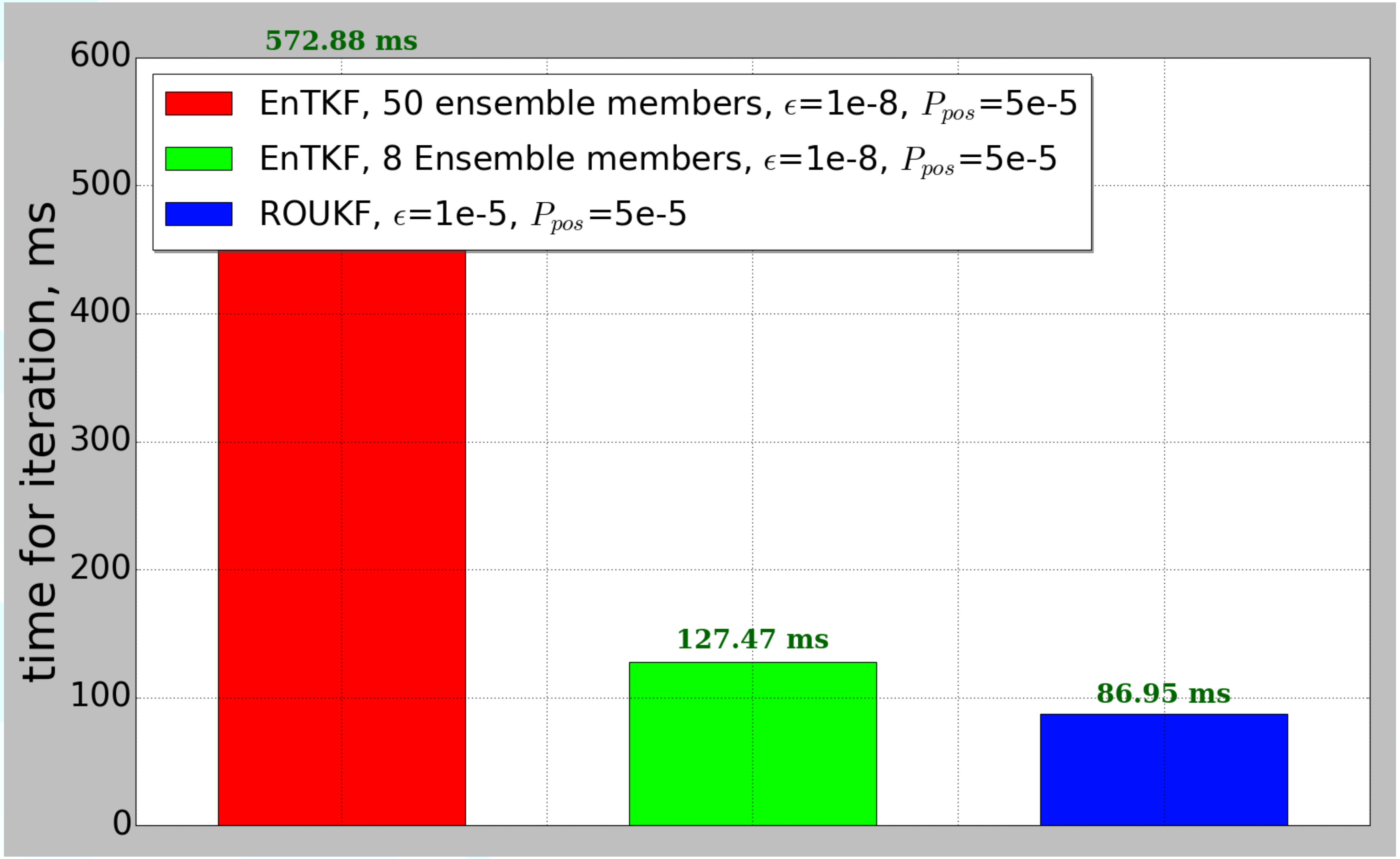
## Anterior and superior




## Superior

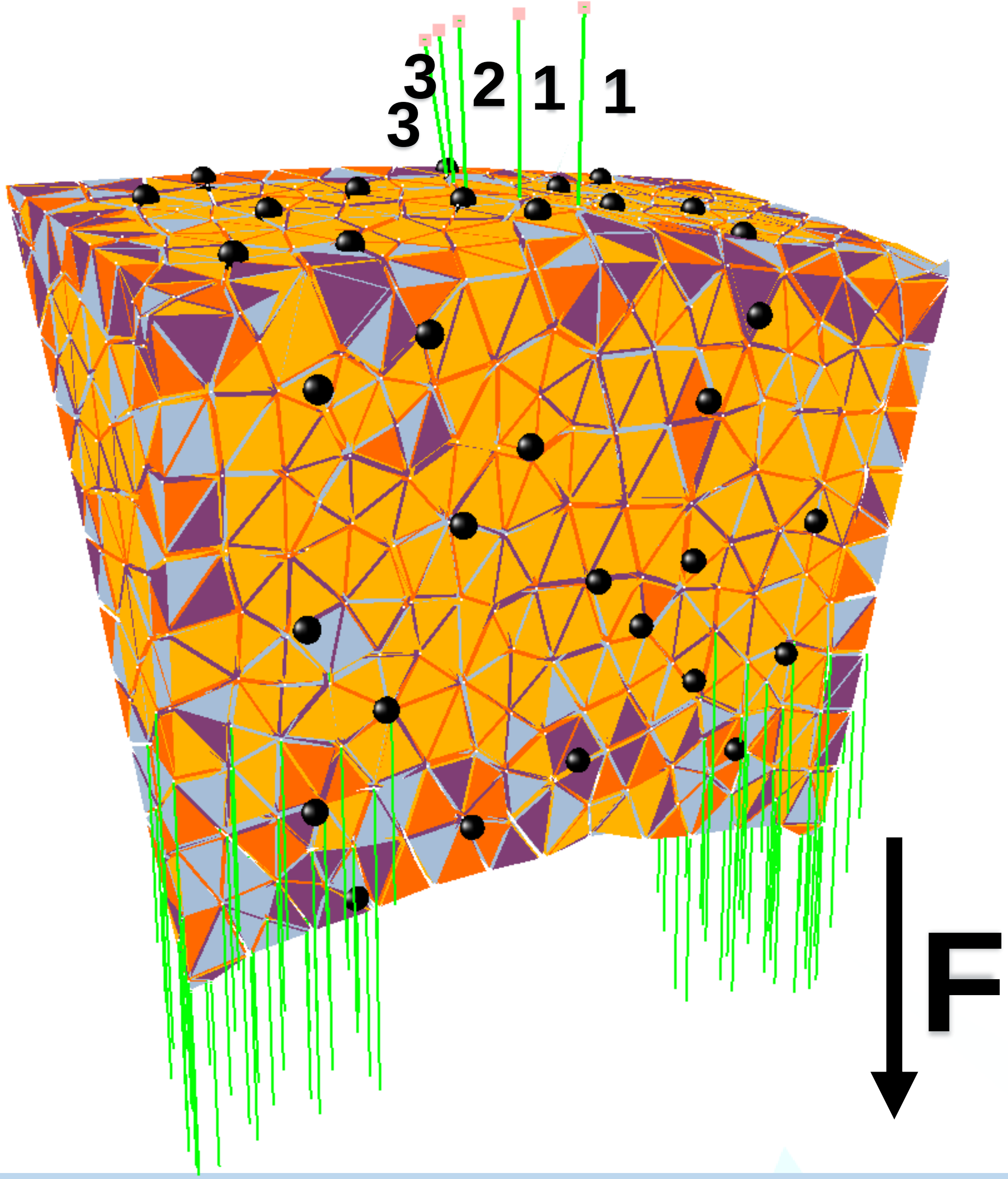


# BCs Experiment Performance Results



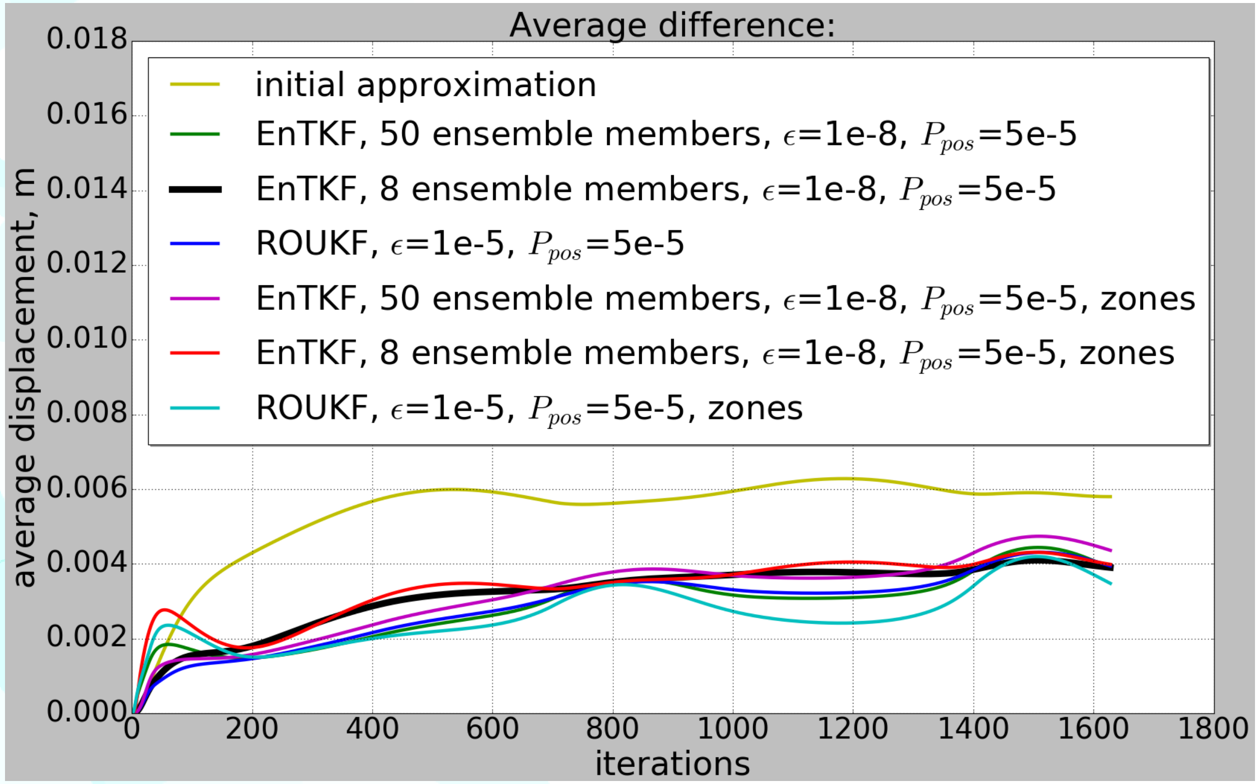
# BCs Second Experiment Description

- **Compared with previous one the idea is to split springs on regions**

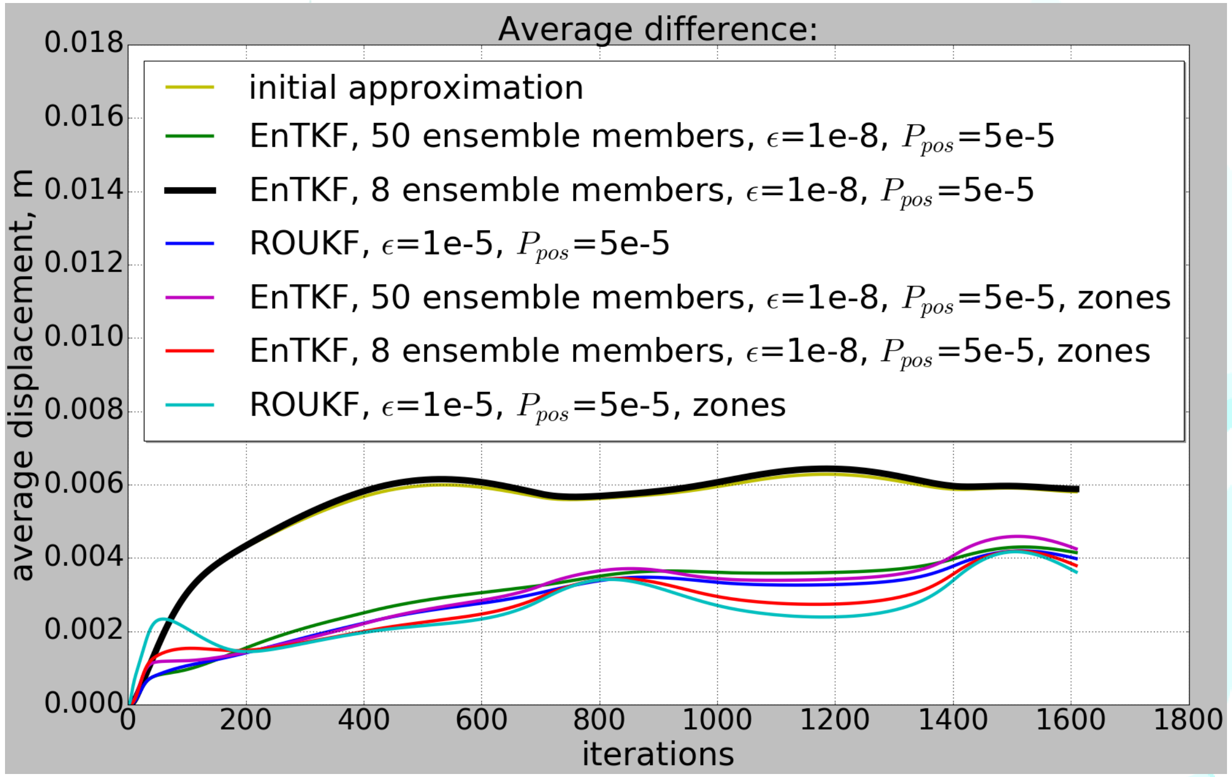


# BCs Second Experiment Validation Results (1)

## Anterior and superior

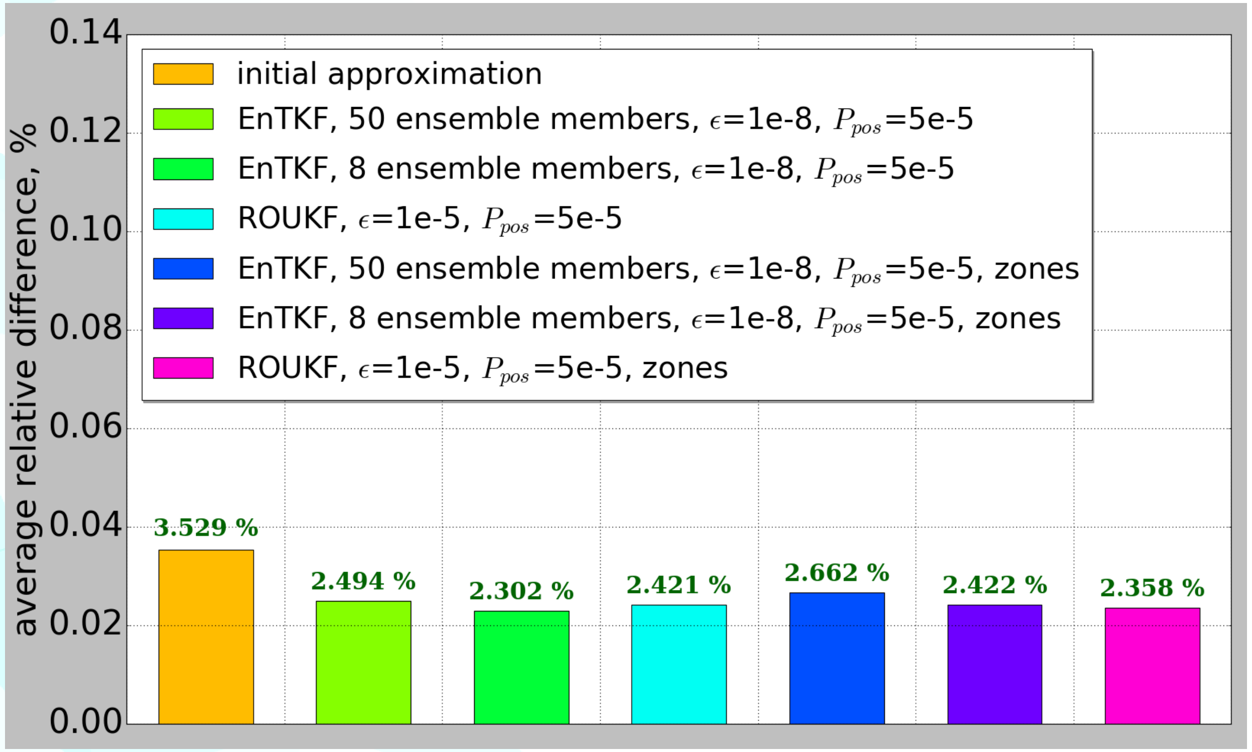



## Superior

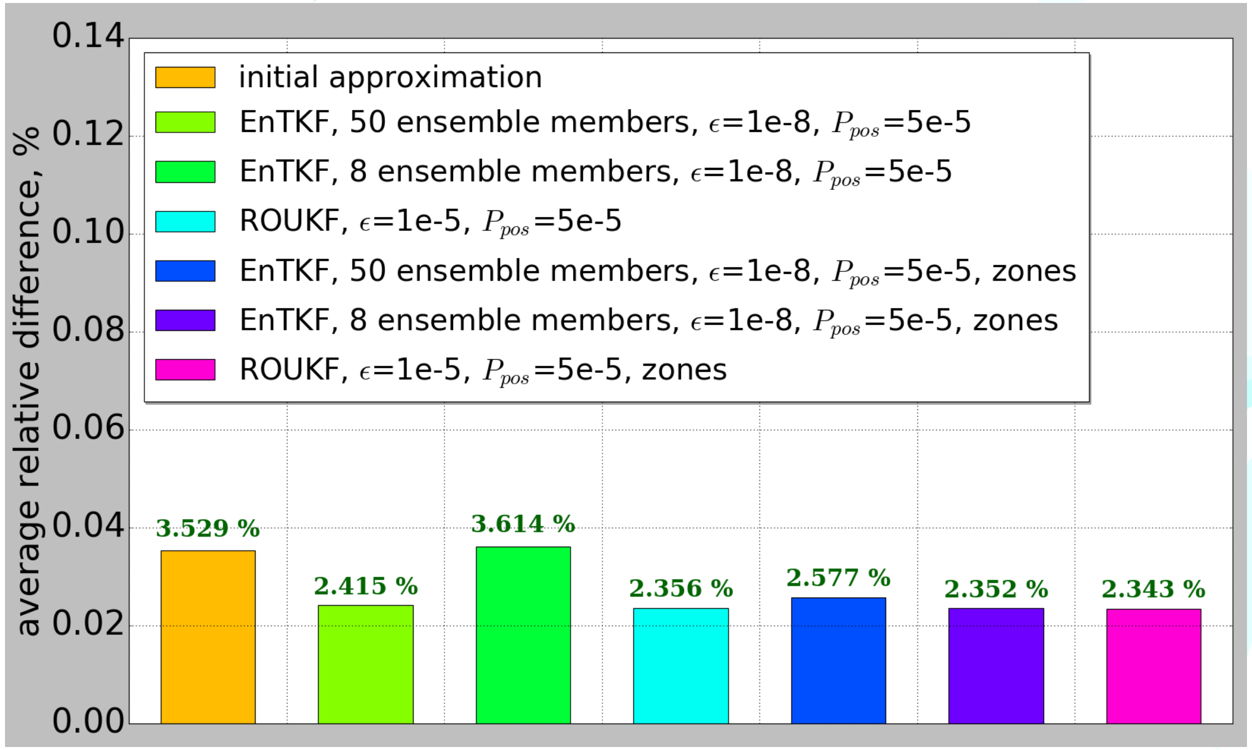


# BCs Second Experiment Validation Results (2)

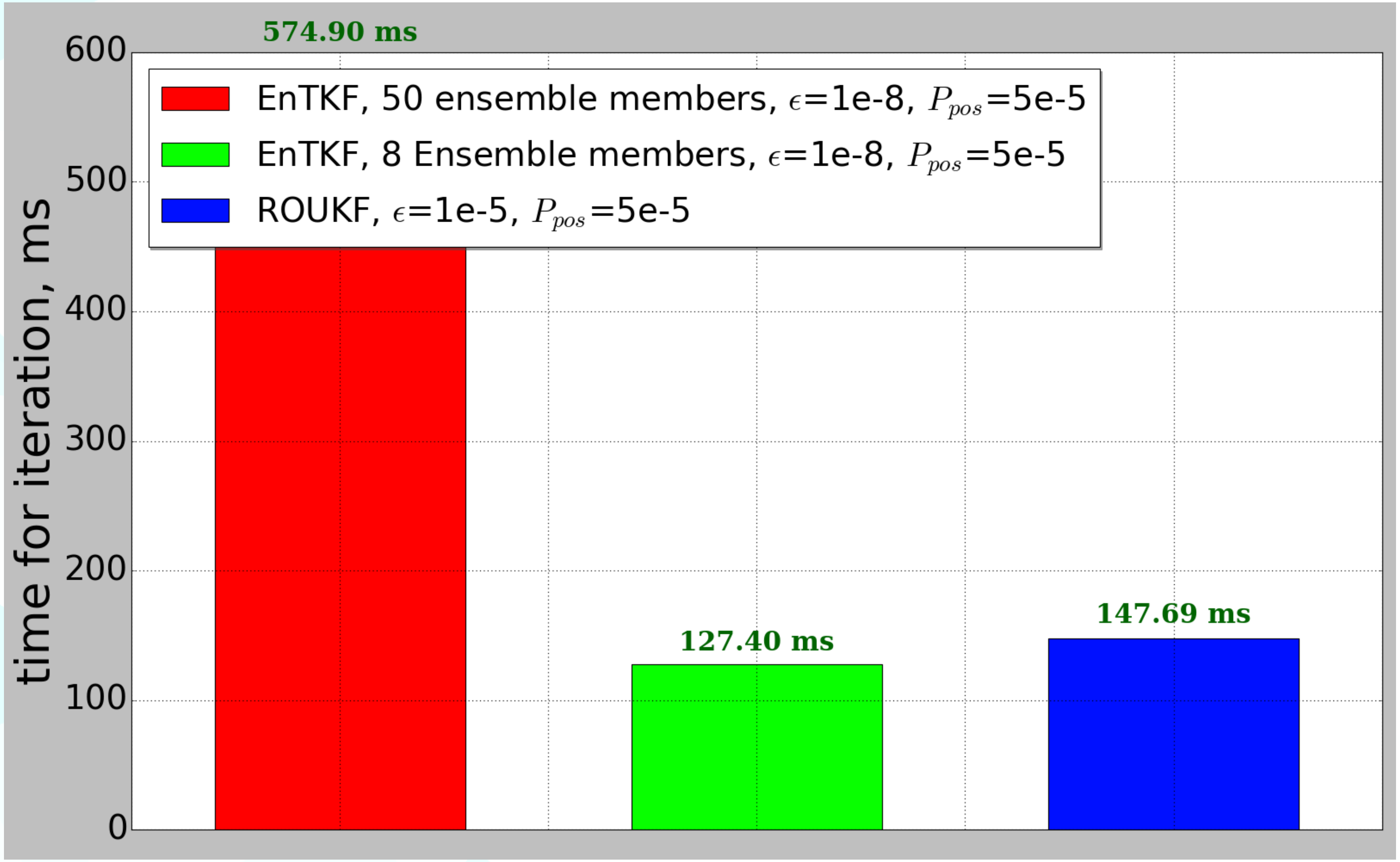
## Anterior and superior




## Superior



# BCs Second Experiment Performance Results



# Performance improvement

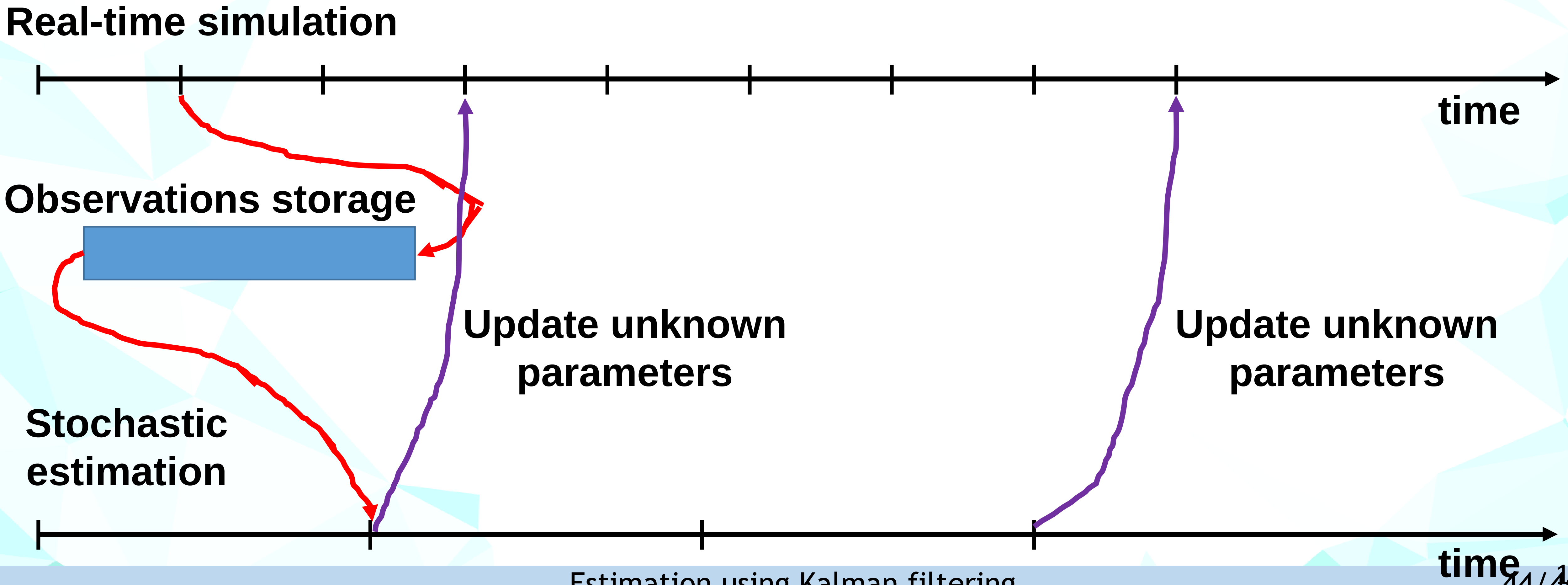
**Data assimilation process is still not real-time but ...**

**Improving computational performance:**

- **Parallel computation of sigma points/ensemble members**
- **Preconditioning**
- **GPU implementation of data assimilation**
- **Delayed estimation**

# Delayed estimation

**Update estimated parameters after prediction-correction (analysis) iteration**



# Future steps

- **Additional experiments before moving to real data**
- **More complex analysis of the corrected model**
- **Preconditioning**

# References

1. Mark Asch, Marc Bocquet, Maelle Nodet – Data Assimilation. Methods Algorithms and Applications, 2016

2. Dan Simon – Optimal state Estimation, Kalman, H∞, and Nonlinear Approaches, 2006