



# Localisation relative de robots mobiles

Rafael Accáio NOGUEIRA

Soheib FERGANI

{ranogueira, sfergani}

at laas.fr



# Qui suis-je ?

Rafael Accácia Nogueira

# Qui suis-je ?

Rafael Accácio Nogueira



Ingénieur/Master Recherche (2019/2018)

*Superviseur pour véhicule autonome (prototypage rapide)*

Superviseur Fady Shokry



# Qui suis-je ?

Rafael Accácio Nogueira



Ingénieur/Master Recherche (2019/2018)

*Superviseur pour véhicule autonome (prototypage rapide)*

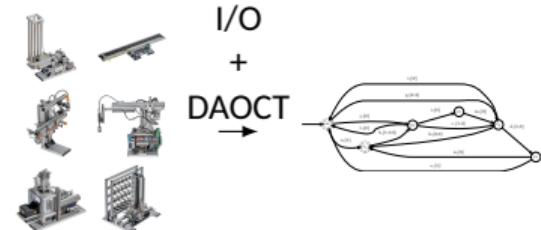
Superviseur Fady Shokry



Ingénieur (2019)

*Identification des SED pour le diagnostic de fautes*

Superviseur Marcos Vicente de Brito Moreira



# Qui suis-je ?

Rafael Accácio Nogueira



Ingénieur/Master Recherche (2019/2018)

*Superviseur pour véhicule autonome (prototypage rapide)*

Superviseur Fady Shokry

Ingénieur (2019)

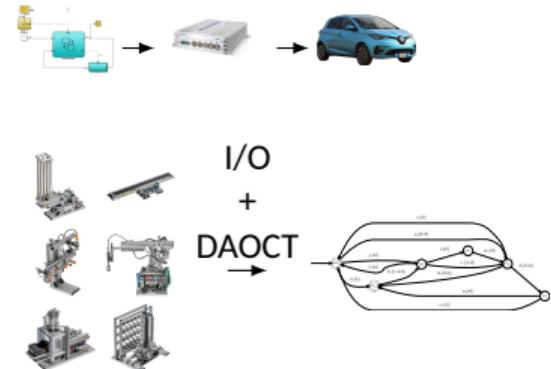
*Identification des SED pour le diagnostic de fautes*

Superviseur Marcos Vicente de Brito Moreira

Thèse (2022)

*Sécurité de dMPC sous False Data Injection*

Superviseurs Hervé Guéguen et Romain Bourdais



# Que fais-je ici ?

Post-doc mai 2023/mai 2024

*Localisation relative garantie dans le scenario anticollision  
des véhicules autonomes*

Plateforme autOCampus (GIS neOCampus)



# À propos de la plateforme aut<sup>o</sup>Campus

# À propos de la plateforme aut<sup>o</sup>Campus

«Plateforme d'expérimentation sur la mobilité du futur»

# À propos de la plateforme aut<sup>o</sup>Campus

«Plateforme d'expérimentation sur la mobilité du futur»

Le campus de Rangueil : une « smart city »

# À propos de la plateforme autocampus

«Plateforme d'expérimentation sur la mobilité du futur»

Le campus de Rangueil : une « smart city »

<https://www.irit.fr/autocampus/>

# À propos de la plateforme autOCampus

«Plateforme d'expérimentation sur la mobilité du futur»

Le campus de Rangueil : une « smart city »

<https://www.irit.fr/autocampus/>

- Simulateur SimFlex
- Navette et droïdes autonomes
- Véhicule autonome open source
- Lampadaires intelligents
- 5G privée (V2X)
- Datalake
- Centre de supervision



# À propos de la plateforme autOCampus

«Plateforme d'expérimentation sur la mobilité du futur»

Le campus de Rangueil : une « smart city »

<https://www.irit.fr/autocampus/>

- Simulateur SimFlex
- Navette et droïdes autonomes
- Véhicule autonome open source
- Lampadaires intelligents
- 5G privée (V2X)
- Datalake
- Centre de supervision

Nous serons un des premiers utilisateurs de la plateforme



# Problème : Navigation sans collision

On a plusieurs scénarios de navigation

- Carrefours
- Dépassemens
- Circulation en peloton
- ...

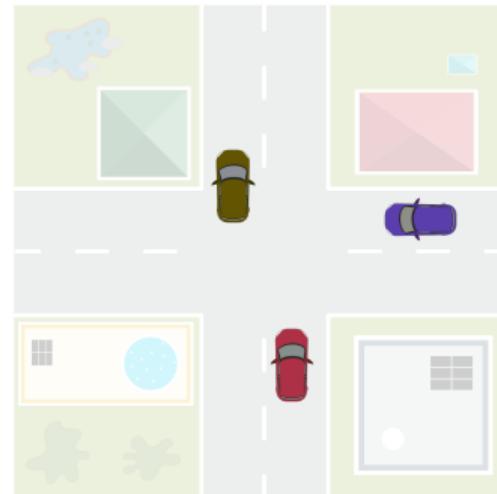


# Problème : Navigation sans collision

On a plusieurs scénarios de navigation

- Carrefours
- Dépassemens
- Circulation en peloton
- ...

Localisation absolue n'est pas nécessaire



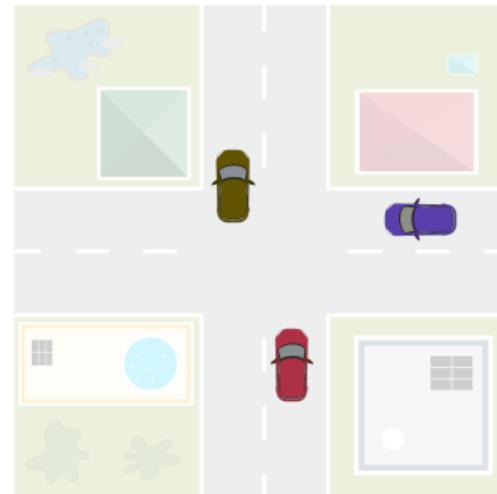
# Problème : Navigation sans collision

On a plusieurs scénarios de navigation

- Carrefours
- Dépassemens
- Circulation en peloton
- ...

Localisation absolue n'est pas nécessaire

Distances relatives



# Problème : Navigation sans collision

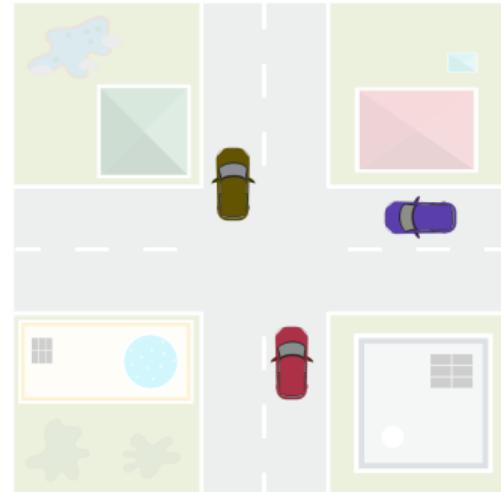
On a plusieurs scénarios de navigation

- Carrefours
- Dépassemens
- Circulation en peloton
- ...

Localisation absolue n'est pas nécessaire

Distances relatives → zones interdites

$$\mathcal{D}_i \ni d_{i,j} = p_j - p_i$$



# Problème : Navigation sans collision

On a plusieurs scénarios de navigation

- Carrefours
- Dépassemens
- Circulation en peloton
- ...

Localisation absolue n'est pas nécessaire

Distances relatives → zones interdites

$$\mathcal{D}_i \ni d_{i,j} = p_j - p_i$$



# Problème : Navigation sans collision

On a plusieurs scénarios de navigation

- Carrefours
- Dépassemens
- Circulation en peloton
- ...

Localisation absolue n'est pas nécessaire

Distances relatives → **zones interdites**

$$\mathcal{D}_i \ni d_{i,j} = p_j - p_i$$



Utiliser les  $\mathcal{D}_i$  pour des algorithmes de navigation

## Stratégie usuelle : Estimation Bayésienne (Kalman)

---

- Modèles dynamiques pour prédire état
- Capteurs embarqués pour corriger prédictions du modèle

## Stratégie usuelle : Estimation Bayésienne (Kalman)

---

- Modèles dynamiques pour prédire état
- Capteurs embarqués pour corriger prédictions du modèle

Exemples de «capteurs» embarqué dans une voiture

# Stratégie usuelle : Estimation Bayésienne (Kalman)

- Modèles dynamiques pour prédire état
- Capteurs embarqués pour corriger prédictions du modèle

Exemples de «capteurs» embarqué dans une voiture

- |          |                  |             |
|----------|------------------|-------------|
| ○ D-GNSS | ○ Ultra-Wideband | ○ Odométrie |
| ○ Sonar  | ○ Lidar          | ○ ...       |

# Stratégie usuelle : Estimation Bayésienne (Kalman)

- Modèles dynamiques pour prédire état
- Capteurs embarqués pour corriger prédictions du modèle

Exemples de «capteurs» embarqué dans une voiture

- |                       |                          |             |
|-----------------------|--------------------------|-------------|
| ○ D-GNSS (code phase) | ○ Ultra-Wideband (range) | ○ Odométrie |
| ○ Sonar               | ○ Lidar                  | ○ ...       |

$$\nabla^r \rho_{p,i,j} = \|\mathbf{p}_i - \mathbf{p}_j\| - \|\mathbf{p}_i + \mathbf{d}_{i,j} - \mathbf{p}_s\| - \|\mathbf{p}_i - \mathbf{p}_r\| + \|\mathbf{p}_i + \mathbf{d}_{i,j} - \mathbf{p}_r\| + \nabla^r \epsilon_{p,i,j}$$

$$r_{i,j} = \|\mathbf{d}_{i,j}\| + w_r$$

# Stratégie usuelle : Estimation Bayésienne (Kalman)

- Modèles dynamiques pour prédire état
- Capteurs embarqués pour corriger prédictions du modèle

Exemples de «capteurs» embarqué dans une voiture

- D-GNSS (code phase)
- Sonar
- Ultra-Wideband (range)
- Lidar
- Odométrie
- ...

$$\nabla^r \rho_{p,i,j} = \| \mathbf{p}_i - \mathbf{p}_j \| - \| \mathbf{p}_i + \mathbf{d}_{i,j} - \mathbf{p}_s \| - \| \mathbf{p}_i - \mathbf{p}_r \| + \| \mathbf{p}_i + \mathbf{d}_{i,j} - \mathbf{p}_r \| + \nabla^r \epsilon_{p,i,j}$$

$$r_{i,j} = \| \mathbf{d}_{i,j} \| + w_r$$

# Stratégie usuelle : Estimation Bayésienne (Kalman)

- Modèles dynamiques pour prédire état
- Capteurs embarqués pour corriger prédictions du modèle

Exemples de «capteurs» embarqué dans une voiture

- D-GNSS (code phase)
- Sonar
- Ultra-Wideband (range)
- Lidar
- Odométrie
- ...

$$\nabla^r \rho_{p,i,j} = \| \mathbf{p}_i - \mathbf{p}_j \| - \| \mathbf{p}_i + \mathbf{d}_{i,j} - \mathbf{p}_s \| - \| \mathbf{p}_i - \mathbf{p}_r \| + \| \mathbf{p}_i + \mathbf{d}_{i,j} - \mathbf{p}_r \| + \nabla^r \epsilon_{p,i,j}$$

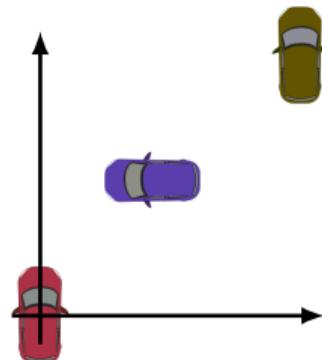
$$r_{i,j} = \| \mathbf{d}_{i,j} \| + w_r$$

$\|\cdot\| \rightarrow$  Non-linéaire  $\rightarrow$  Extensions non-linéaires du Filtre de Kalman (EKF, UKF, etc)

# Estimation Bayésienne : EKF, UKF

## Hypothèses

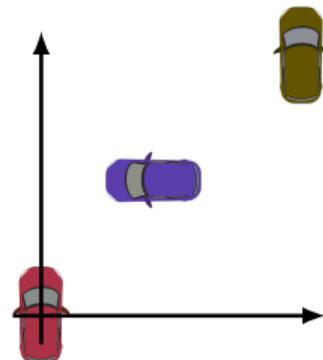
- 1 Bruit d'évolution
- 2 Bruit de mesure



# Estimation Bayésienne : EKF, UKF

## Hypothèses

- 1 Bruit d'évolution } Gaussiens
- 2 Bruit de mesure } et connus



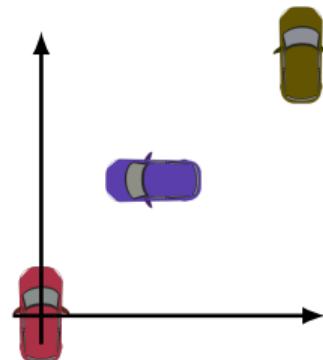
# Estimation Bayésienne : EKF, UKF

## Hypothèses

- 1 Bruit d'évolution
  - 2 Bruit de mesure
- } Gaussiens et connus

## Résultat = Erreur Gaussien

Intervalles de confiance = ellipsoïdes



# Estimation Bayésienne : EKF, UKF

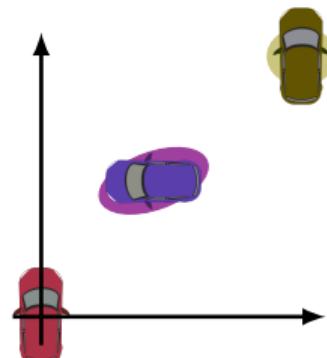
## Hypothèses

- 1 Bruit d'évolution
  - 2 Bruit de mesure
- } Gaussiens et connus

## Résultat = Erreur Gaussien

Intervalles de confiance = ellipsoïdes

$$3\sigma = 99.73\%$$



# Estimation Bayésienne : EKF, UKF

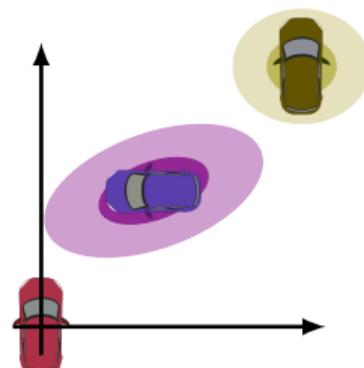
## Hypothèses

- 1 Bruit d'évolution
  - 2 Bruit de mesure
- } Gaussiens et connus

## Résultat = Erreur Gaussien

Intervalles de confiance = ellipsoïdes

- 3 $\sigma$  = 99.73%
- 4 $\sigma$  = 99.993%



# Estimation Bayésienne : EKF, UKF

## Problème dans les hypothèses

### Hypothèses

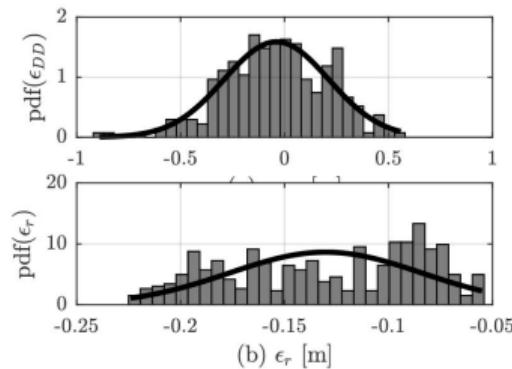
- 1 Bruit d'évolution
  - 2 Bruit de mesure
- } Gaussiens et connus

# Estimation Bayésienne : EKF, UKF

## Problème dans les hypothèses

### Hypothèses

- 1 Bruit d'évolution
  - 2 Bruit de mesure
- } Gaussiens et connus

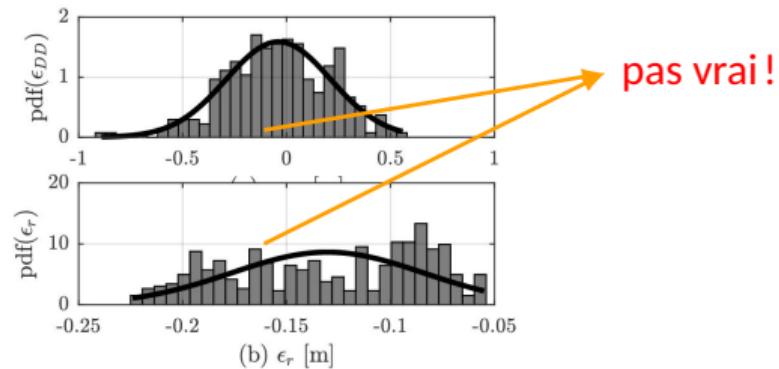


# Estimation Bayésienne : EKF, UKF

## Problème dans les hypothèses

### Hypothèses

- 1 Bruit d'évolution
  - 2 Bruit de mesure
- } Gaussiens et connus



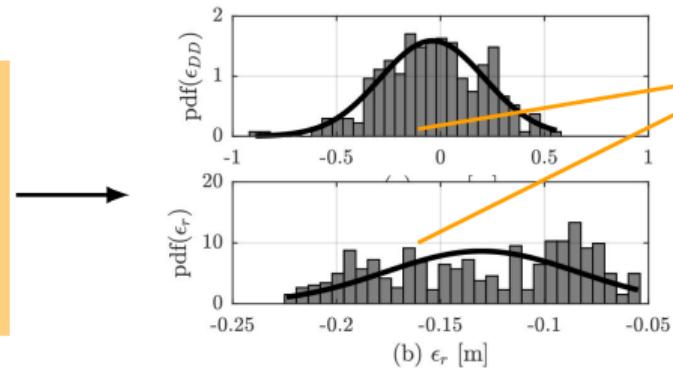
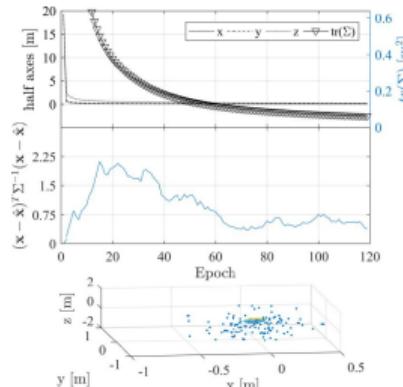
pas vrai!

# Estimation Bayésienne : EKF, UKF

## Problème dans les hypothèses

### Hypothèses

- 1 Bruit d'évolution
  - 2 Bruit de mesure
- } Gaussiens et connus

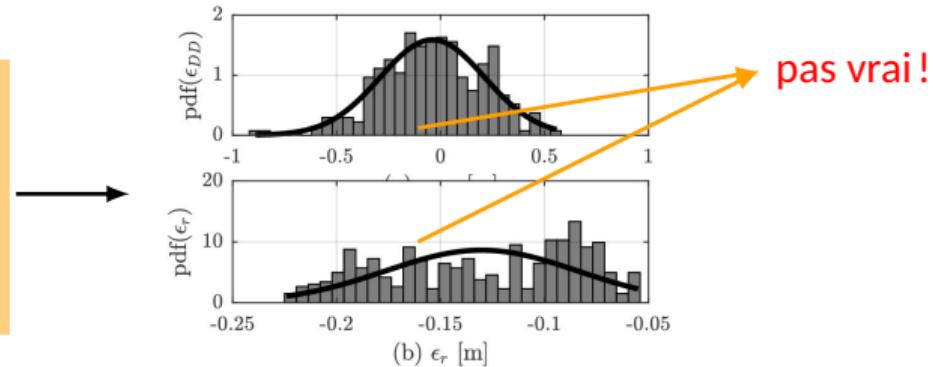
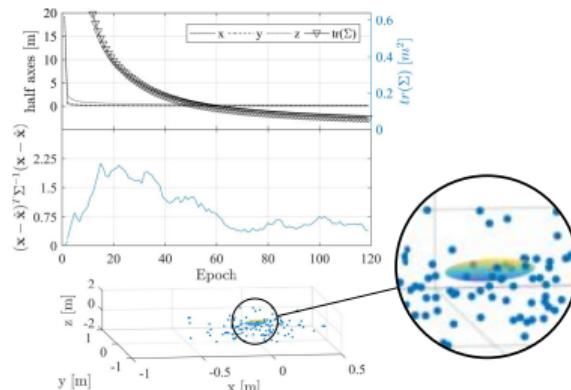


# Estimation Bayésienne : EKF, UKF

## Problème dans les hypothèses

### Hypothèses

- 1 Bruit d'évolution
  - 2 Bruit de mesure
- } Gaussiens et connus

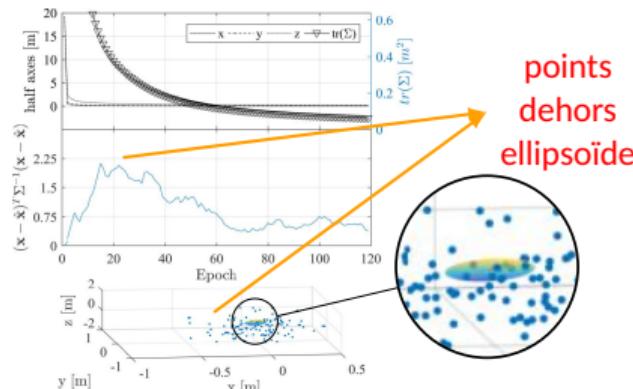


# Estimation Bayésienne : EKF, UKF

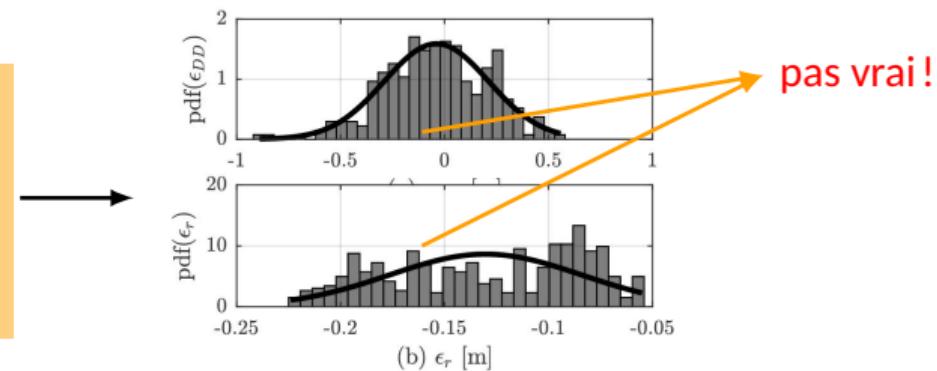
## Problème dans les hypothèses

### Hypothèses

- 1 Bruit d'évolution
  - 2 Bruit de mesure
- } Gaussiens et connus

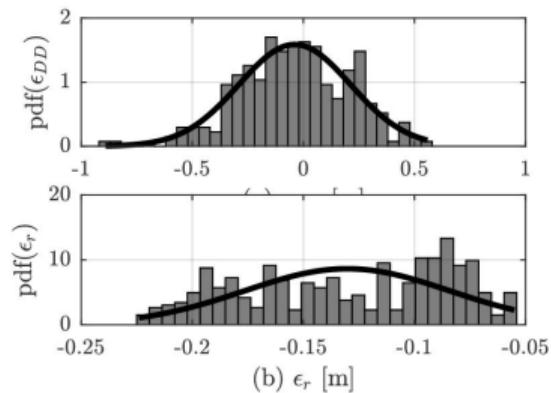


points  
dehors  
ellipsoïde



(b)  $\epsilon_r$  [m]

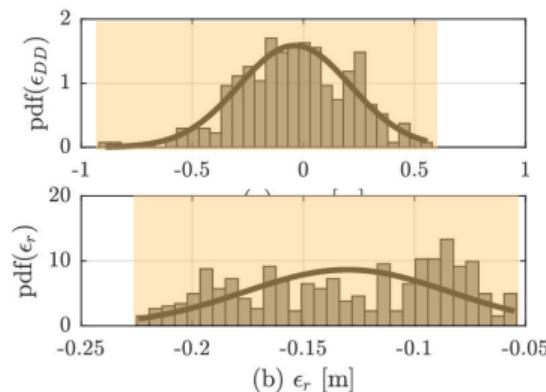
# Alternative : Méthodes ensemblistes



# Alternative : Méthodes ensemblistes

## Hypothèse principale

- On ne connaît pas la distribution des erreurs, mais ses bornes

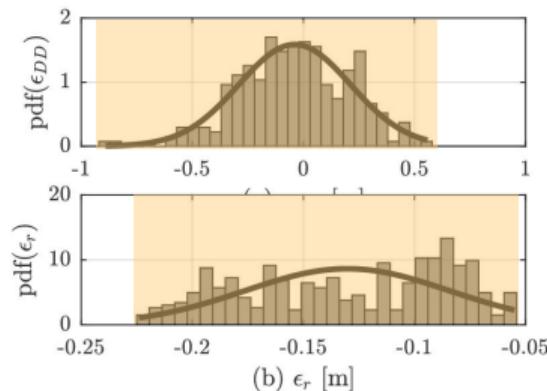


# Alternative : Méthodes ensemblistes

## Hypothèse principale

Datasheet + Erreur Modélisation

- On ne connaît pas la distribution des erreurs, mais ses bornes

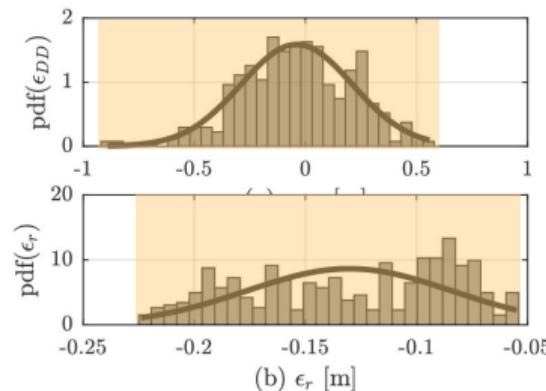


# Alternative : Méthodes ensemblistes

## Hypothèse principale

Datasheet + Erreur Modélisation

- On ne connaît pas la distribution des erreurs, mais ses bornes
- Garantie que les valeurs appartiennent à l'intervalle
- Intervalle + dynamiques génèrent un ensemble garanti

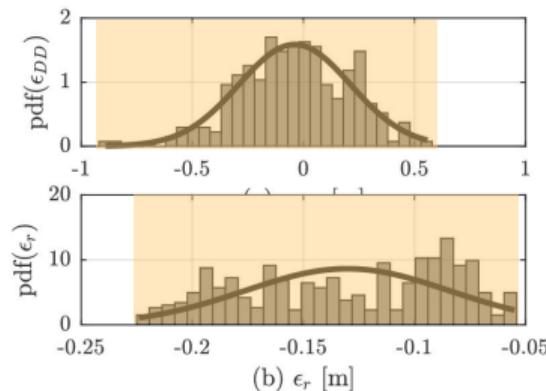


# Alternative : Méthodes ensemblistes

## Hypothèse principale

Datasheet + Erreur Modélisation

- On ne connaît pas la distribution des erreurs, mais ses bornes
- Garantie que les valeurs appartiennent à l'intervalle
- Intervalle + dynamiques génèrent un ensemble garanti (mais de forme inconnue 😐)



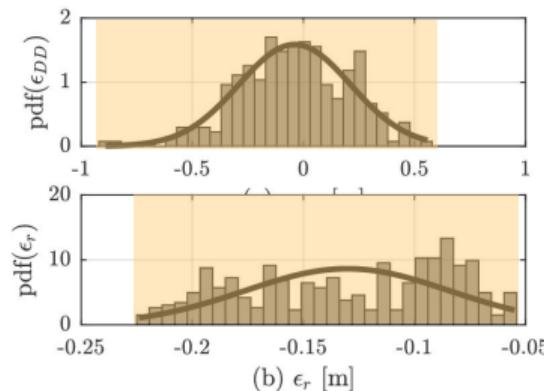
# Alternative : Méthodes ensemblistes

## Hypothèse principale

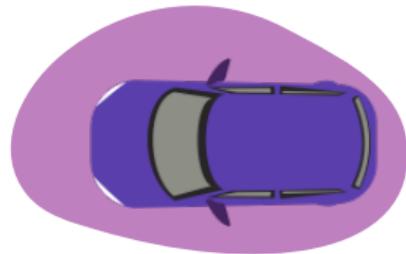
- On ne connaît pas la distribution des erreurs, mais ses bornes
- Garantie que les valeurs appartiennent à l'intervalle
- Intervalle + dynamiques génèrent un ensemble garanti (mais de forme inconnue 😐)

Datasheet + Erreur Modélisation

On peut l'approximer 😊

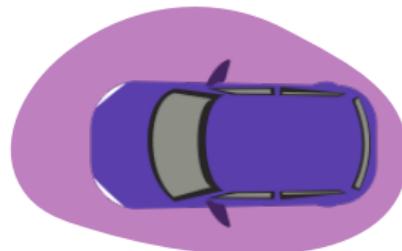


# Approximation par ensembles connus



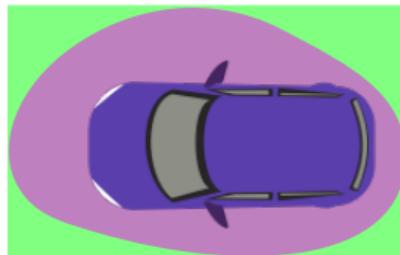
# Approximation par ensembles connus

- Compromis entre complexité de calcul et conservatisme



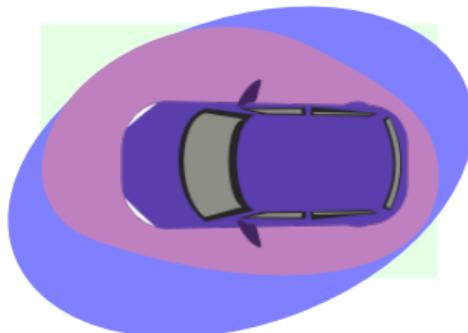
# Approximation par ensembles connus

- Compromis entre complexité de calcul et conservatisme
  - Rapide mais conservative : intervalles, ellipsoïdes, zonotopes



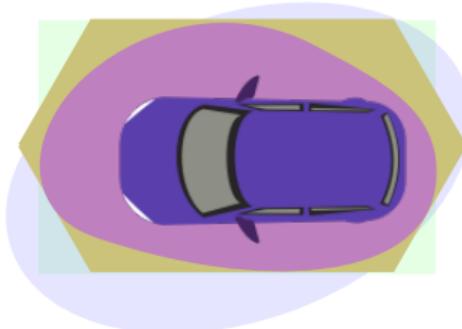
# Approximation par ensembles connus

- Compromis entre complexité de calcul et conservatisme
  - Rapide mais conservative : intervalles, [ellipsoïdes](#), zonotopes



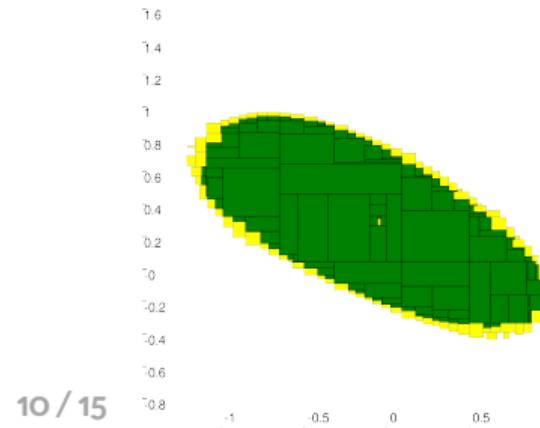
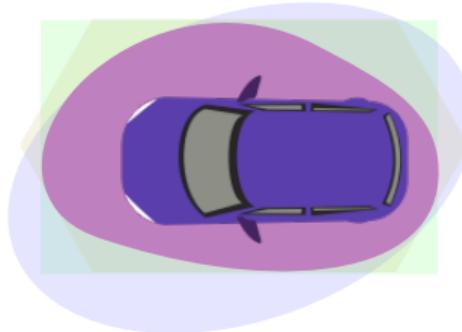
# Approximation par ensembles connus

- Compromis entre complexité de calcul et conservatisme
  - Rapide mais conservative : intervalles, ellipsoïdes, zonotopes



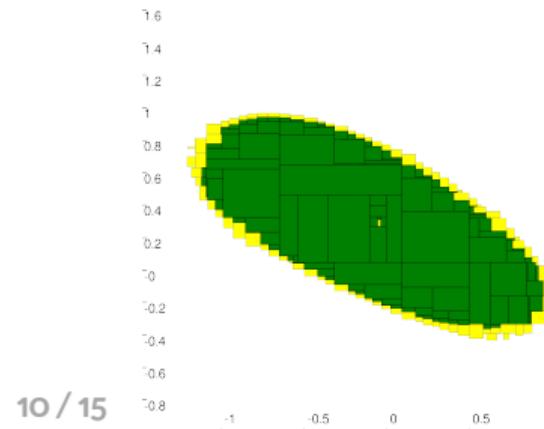
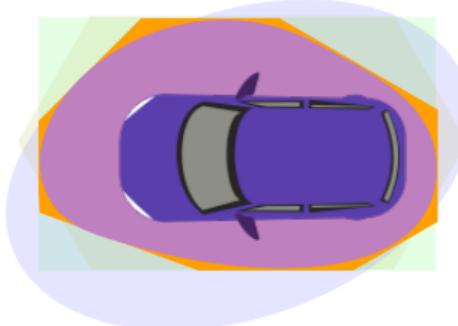
# Approximation par ensembles connus

- Compromis entre complexité de calcul et conservatisme
  - Rapide mais conservative : intervalles, ellipsoïdes, zonotopes
  - Lent mais serré (normalement hors-ligne) : **sous-pavage (SIVIA)**, polytopes



# Approximation par ensembles connus

- Compromis entre complexité de calcul et conservatisme
  - Rapide mais conservative : intervalles, ellipsoïdes, zonotopes
  - Lent mais serré (normalement hors-ligne) : sous-pavage (SIVIA), polytopes

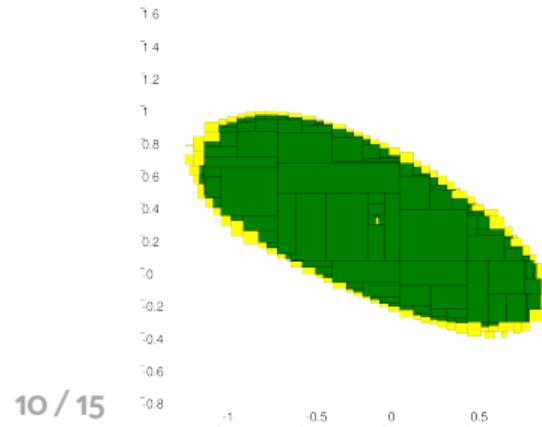
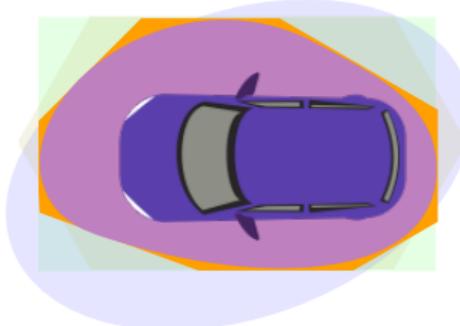


# Approximation par ensembles connus

- Compromis entre complexité de calcul et conservatisme
  - Rapide mais conservative : intervalles, ellipsoïdes, zonotopes
  - Lent mais serré (normalement hors-ligne) : sous-pavage (SIVIA), polytopes
- Zonotope Constraint<sup>a</sup> (une autre représentation de polytopes)

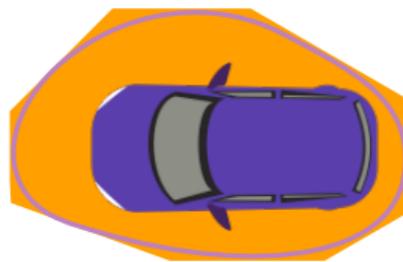
---

a. Scott et al., "Constrained Zonotopes" (2016)



# Zonotope constraint :

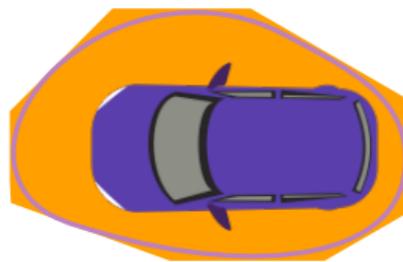
- Propriétés similaires au polytopes (Clos pour l'intersection et somme de Minkowski)



$$\mathcal{C}_{\mathcal{Z}} = \{G\xi + c \mid \|\xi\|_{\infty} \leq 1, A\xi = b\}$$

# Zonotope constraint :

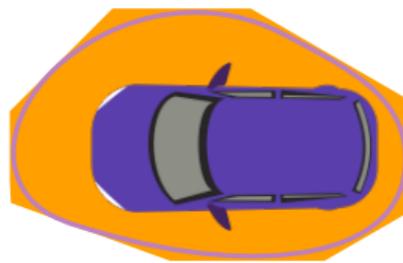
- Propriétés similaires au polytopes (Clos pour l'intersection et somme de Minkowski)
  - + Réprésentation facilite exécution des opérations



$$\mathcal{C}_{\mathcal{Z}} = \{G\xi + c \mid \|\xi\|_{\infty} \leq 1, A\xi = b\}$$

# Zonotope constraint :

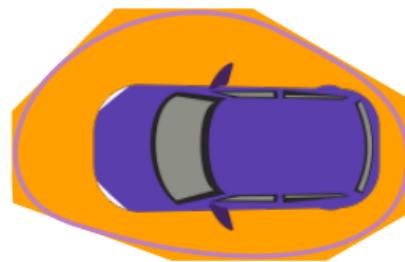
- Propriétés similaires au polytopes (Clos pour l'intersection et somme de Minkowski)
  - + Réprésentation facilite exécution des opérations
  - + Asymétrie → plus de liberté



$$\mathcal{C}_{\mathcal{Z}} = \{G\xi + c \mid \|\xi\|_{\infty} \leq 1, A\xi = b\}$$

# Zonotope constraint :

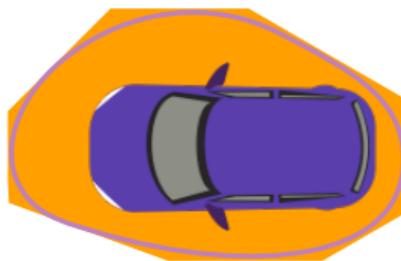
- Propriétés similaires au polytopes (Clos pour l'intersection et somme de Minkowski)
  - + Réprésentation facilite exécution des opérations
  - + Asymétrie → plus de liberté
  - + Facilement applicable pour des systèmes linéaires



$$\mathcal{C}_{\mathcal{Z}} = \{G\xi + c \mid \|\xi\|_{\infty} \leq 1, A\xi = b\}$$

# Zonotope constraint :

- Propriétés similaires au polytopes (Clos pour l'intersection et somme de Minkowski)
  - + Réprésentation facilite exécution des opérations
  - + Asymétrie → plus de liberté
  - + Facilement appliquable pour des systèmes linéaires
  - Chaque opération augmente la complexité (réduction possible au détriment du volume)

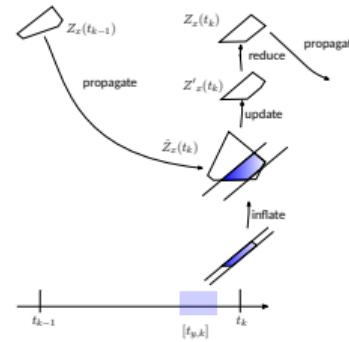


$$\mathcal{C}_{\mathcal{Z}} = \{G\xi + c \mid \|\xi\|_{\infty} \leq 1, A\xi = b\}$$

# Filtre d'estimation avec zonotopes contraints (CZESMF)

▶ plus

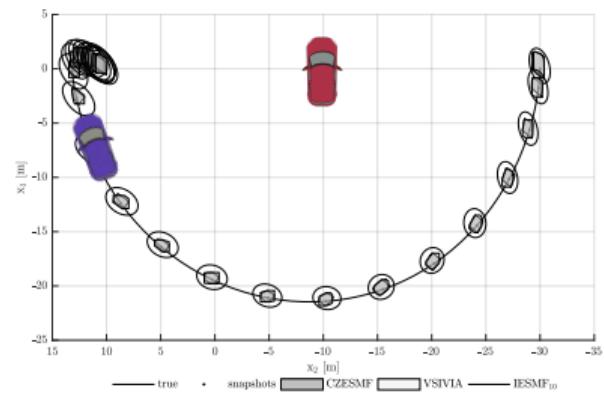
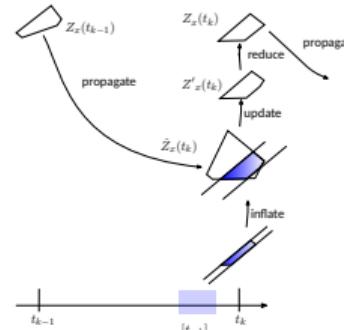
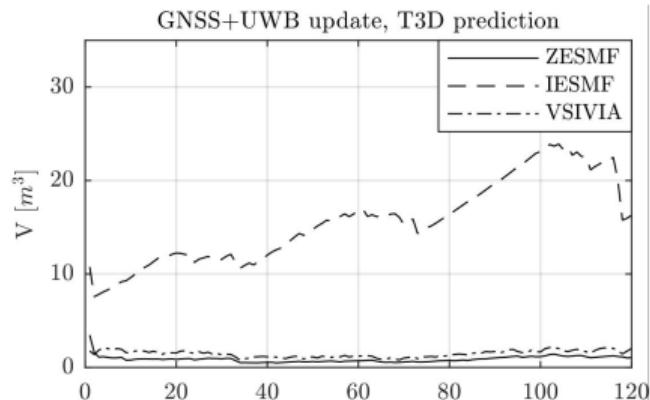
- Extension pour des systèmes non-linéaires
- Extension pour temps d'observation不确定
- Testé (en simulation) avec D-GNSS et UWB p2p
- Exécution proche du temps réel (Matlab)



# Filtre d'estimation avec zonotopes contraints (CZESMF)

▶ plus

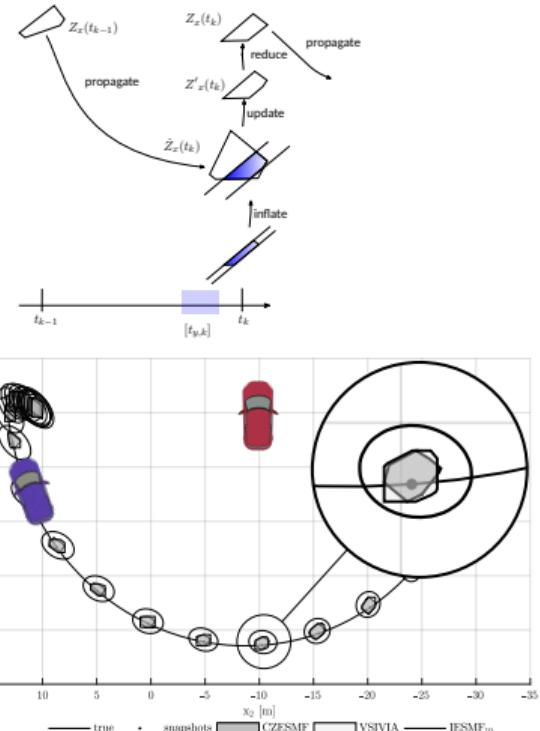
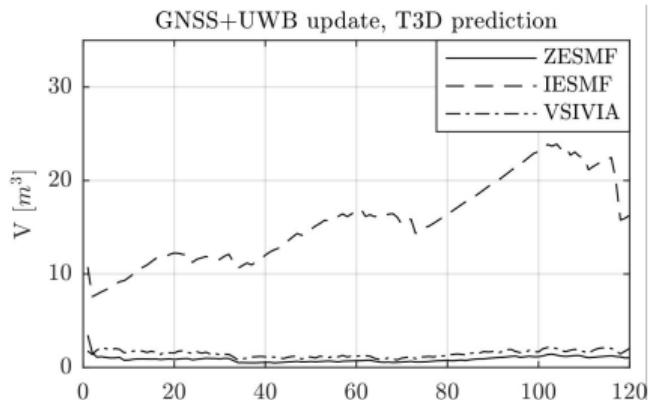
- Extension pour des systèmes non-linéaires
- Extension pour temps d'observation不确定
- Testé (en simulation) avec D-GNSS et UWB p2p
- Exécution proche du temps réel (Matlab)



# Filtre d'estimation avec zonotopes contraints (CZESMF)

▶ plus

- Extension pour des systèmes non-linéaires
- Extension pour temps d'observation不确定
- Testé (en simulation) avec D-GNSS et UWB p2p
- Exécution proche du temps réel (Matlab)



# Mission

---

- ① Validation de l'algorithme dans une cible
  - Quanser QBot 3 (Raspberry Pi 4 Model B)
    - IMU
    - Odométrie
    - Caméra
    - ...
  - Tâches
    - Configuration Robot
    - Intégration code C++ et tests
    - Adaptation (dynamique et capteurs)
    - Environnement Simulation (ROS)
    - Création démonstrateur



# Mission - Perspective

---

## ② Application : Droïdes TwinswHeel

- Données GNSS réelles du campus
- Adaptation (dynamique et capteurs)
- Tests en simulation (ROS)
- Intégration sur les droïdes



Merci ! Questions ?



Rafael Accáio NOGUEIRA  
Soheib FERGANI  
{ranogueira, sfergani}  
at laas.fr



# CZESMF - L'algorithme



Équations d'état et d'observation (Horodatage incluse)

$$\begin{aligned}\boldsymbol{x}_{t_{k+1}} &= f(\boldsymbol{x}_{t_k}, \boldsymbol{u}_{t_k}, t_{k+1}, t_k) + \boldsymbol{q}_{t_k} \\ \boldsymbol{y}_{t_{y,k+1}} &= h(\boldsymbol{x}_{t_{y,k+1}}, \boldsymbol{u}_{t_{y,k+1}}, t_{y,k+1}) + \boldsymbol{w}_{t_{y,k+1}} \\ \hat{\boldsymbol{t}}_{\boldsymbol{y}_{t_{k+1}}} &= t_{y,k+1} + w_{t_{t_{y,k+1}}}\end{aligned}$$

# CZESMF - L'algorithme

Équations d'état et d'observation (Horodatage incluse)

$$\begin{aligned}\boldsymbol{x}_{t_{k+1}} &= f(\boldsymbol{x}_{t_k}, \boldsymbol{u}_{t_k}, t_{k+1}, t_k) + \boldsymbol{q}_{t_k} \\ \boldsymbol{y}_{t_{y,k+1}} &= h(\boldsymbol{x}_{t_{y,k+1}}, \boldsymbol{u}_{t_{y,k+1}}, t_{y,k+1}) + \boldsymbol{w}_{t_{y,k+1}} \\ \hat{\boldsymbol{t}}_{\boldsymbol{y}_{t_{k+1}}} &= t_{y,k+1} + w_{t_{y,k+1}}\end{aligned}$$

Inclusion linéaire à partir de la série de Taylor + inflation à cause des délais

$$\boldsymbol{x}_{t_{k+1}} \in \hat{\mathcal{Z}}_{\boldsymbol{x}_{t_{k+1}}} = F_{t_k} \mathcal{Z}_{\boldsymbol{x}_{t_k}} + B_{t_k} \mathcal{Z}_{\boldsymbol{u}_{t_k}} + \mathcal{Z}_{\boldsymbol{q}_{t_k}} + \mathcal{Z}_{[\epsilon_{t_k}^f]} + \boldsymbol{c}_{t_k}^f$$

$$\boldsymbol{y}_{[t_{y,k+1}]} \in H_{t_{k+1}} \mathcal{Z}_{\boldsymbol{x}_{t_{k+1}}} + \mathcal{Z}_{[\epsilon']} \Leftrightarrow H_{t_{k+1}} \mathcal{Z}_{\boldsymbol{x}_{t_{k+1}}} \in \mathcal{Z}_{[\epsilon]}$$

# CZESMF - L'algorithme

Équations d'état et d'observation (Horodatage incluse)

$$\begin{aligned}\mathbf{x}_{t_{k+1}} &= f(\mathbf{x}_{t_k}, \mathbf{u}_{t_k}, t_{k+1}, t_k) + \mathbf{q}_{t_k} \\ \mathbf{y}_{t_{y,k+1}} &= h(\mathbf{x}_{t_{y,k+1}}, \mathbf{u}_{t_{y,k+1}}, t_{y,k+1}) + \mathbf{w}_{t_{y,k+1}} \\ \hat{\mathbf{y}}_{t_{y,k+1}} &= t_{y,k+1} + w_{t_{y,k+1}}\end{aligned}$$

Inclusion linéaire à partir de la série de Taylor + inflation à cause des délais

$$\mathbf{x}_{t_{k+1}} \in \hat{\mathcal{Z}}_{\mathbf{x}_{t_{k+1}}} = F_{t_k} \mathcal{Z}_{\mathbf{x}_{t_k}} + B_{t_k} \mathcal{Z}_{\mathbf{u}_{t_k}} + \mathcal{Z}_{\mathbf{q}_{t_k}} + \mathcal{Z}_{[\epsilon_{t_k}^f]} + \mathbf{c}_{t_k}^f$$

$$\mathbf{y}_{[t_{y,k+1}]} \in H_{t_{k+1}} \mathcal{Z}_{\mathbf{x}_{t_{k+1}}} + \mathcal{Z}_{[\epsilon']} \Leftrightarrow H_{t_{k+1}} \mathcal{Z}_{\mathbf{x}_{t_{k+1}}} \in \mathcal{Z}_{[\epsilon]}$$

Intersect prediction with inverse observation

$$\mathcal{Z}_{\mathbf{x}_{t_{k+1}}} = \hat{\mathcal{Z}}_{\mathbf{x}_{t_{k+1}}} \cap_{H_{t_k}} \mathcal{Z}_{[\epsilon]}$$

1 / 2

# CZESMF - L'algorithme

