

# LiDAR alapú mobil robot lokalizáció Daum–Huang-szűrő eljárással

Csuzdi Domonkos <sup>1</sup>    Törő Olivér <sup>2</sup>

<sup>1</sup>Szerző, VIK MSc. I. év

<sup>2</sup>Konzulens, KJIT tud. sgmts.

2021. november



# Tartalomjegyzék

Bevezetés

Szenzormodell

Daum–Huang-szűrő

Eredmények

Összegzés

# Motiváció

- ▶ Daum és Huang (2008): újfajta nemlineáris szűrőeljárás: Daum–Huang-szűrő (DHF), majd 2010-ben: Exact Flow Daum–Huang-szűrő (EDH)
- ▶ Egy megoldási javaslat a részecskeszűrők problémáira

# Motiváció

- ▶ Daum és Huang (2008): újfajta nemlineáris szűrőeljárás: Daum–Huang-szűrő (DHF), majd 2010-ben: Exact Flow Daum–Huang-szűrő (EDH)
- ▶ Egy megoldási javaslat a részecskeszűrők problémáira
- ▶ Mobil robotok lokalizációja: gyakran Adaptive Monte Carlo Localization (AMCL) algoritmussal (részecskeszűrő)
- ▶ Egy lokalizációs probléma esetén is jobb-e az EDH?

# Motiváció

- ▶ Daum és Huang (2008): újfajta nemlineáris szűrőeljárás: Daum–Huang-szűrő (DHF), majd 2010-ben: Exact Flow Daum–Huang-szűrő (EDH)
- ▶ Egy megoldási javaslat a részecskeszűrők problémáira
- ▶ Mobil robotok lokalizációja: gyakran Adaptive Monte Carlo Localization (AMCL) algoritmussal (részecskeszűrő)
- ▶ Egy lokalizációs probléma esetén is jobb-e az EDH?
- ▶ Nem vizsgáltak még EDH alapú mobil robot lokalizációt a szakirodalomban
- ▶ A munkám egy ilyen algoritmus kifejlesztéséről szól

# Lokalizáció I.

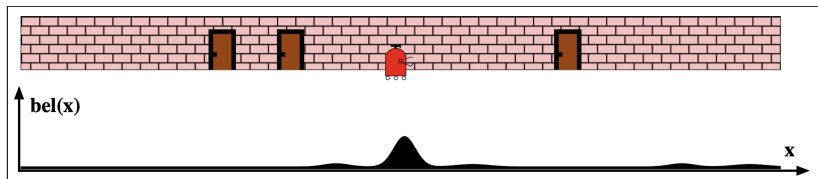
- ▶ Egy ismert térképen követni egy objektumot (ismerni a pozíciót és orientációt)

# Lokalizáció I.

- ▶ Egy ismert térképen követni egy objektumot (ismerni a pozíciót és orientációt)
- ▶ Jelen esetben csak 2D:  $\mathbf{x} = [x \ y \ \theta]^\top$   
→ állapotbecslés szenzorfüziónal: pl. LiDAR és kerék enkóder

# Lokalizáció I.

- ▶ Egy ismert térképen követni egy objektumot (ismerni a pozíciót és orientációt)
- ▶ Jelen esetben csak 2D:  $\mathbf{x} = [x \ y \ \theta]^\top$   
→ állapotbecslés szenzorfüziónal: pl. LiDAR és kerék enkóder
- ▶ Eredendően hibával terhelt mérések: sosem tudjuk pontosan a pózt → valószínűségi eloszlásként kezeljük



S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. Cambridge, Mass.: MIT Press, 2005.



## Lokalizáció II.

- Bayes-féle rekurzív becslés:

predikció:

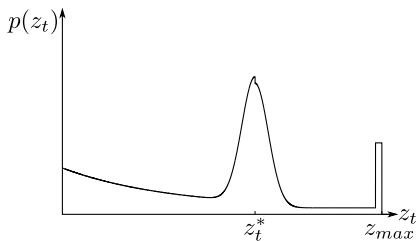
$$\overline{bel}(\mathbf{x}_t) = \int \underbrace{p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t)}_{\text{mozgásmodell}} \overline{bel}(\mathbf{x}_{t-1}) d\mathbf{x}_{t-1}, \quad (1)$$

korrekció:

$$bel(\mathbf{x}_t) = \eta \underbrace{p(\mathbf{z}_t | \mathbf{x}_t)}_{\text{szenzormodell}} \overline{bel}(\mathbf{x}_t). \quad (2)$$

# Módszerek

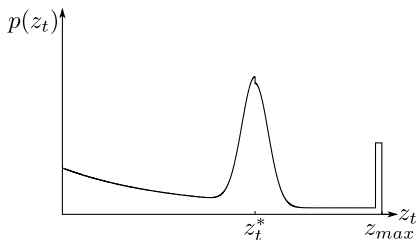
## Hagyományos modell



- ▶ Pásztázó távolságmérőkhöz
- ▶ Nem folytonos:  
→ részecsqueszűrőhöz  
(AMCL) jó, EDH-hoz nem

# Módszerek

## Hagyományos modell



- ▶ Pásztázó távolságmérőkhöz
- ▶ Nem folytonos:  
→ részecskeszűrőhöz  
(AMCL) jó, EDH-hoz nem

## Távolság-transzformáció alapú modell

- ▶ Dantanarayana és mtsai.  
2016-ban
- ▶ Eredetileg EKF-hez
- ▶ Foglaltsági hálón alapuló  
térkép + pásztázó  
távolságmérő (pl. LiDAR)
- ▶ Alapgondolat:  
távolság-transzformáció  
(képfeldolgozás)

# Távolság-transzformáció alapú mérési modell I.

- ▶ Távolság transzformáció: átlagos eltérés mérhető két pontthalmaz között

# Távolság-transzformáció alapú mérési modell I.

- ▶ Távolság transzformáció: átlagos eltérés mérhető két ponthalmaz között
- ▶ Két ponthalmaz: foglalt cellák a térképen és a LiDAR sugarak „végpontjai”

# Távolság-transzformáció alapú mérési modell I.

- ▶ Távolság transzformáció: átlagos eltérés mérhető két ponthalmaz között
- ▶ Két ponthalmaz: foglalt cellák a térképen és a LiDAR sugarak „végpontjai”
- ▶ Eltérés számítása (skalár):  $\psi(\mathbf{x}, \mathbf{z}, \mathbf{m})$

# Távolság-transzformáció alapú mérési modell I.

- ▶ Távolság transzformáció: átlagos eltérés mérhető két pontthalmaz között
- ▶ Két pontthalmaz: foglalt cellák a térképen és a LiDAR sugarak „végpontjai”
- ▶ Eltérés számítása (skalár):  $\psi(\mathbf{x}, \mathbf{z}, \mathbf{m})$
- ▶ Ideális eset (minden végpont foglalt cellán):  $\psi(\mathbf{x}, \mathbf{z}, \mathbf{m}) = 0$

# Távolság-transzformáció alapú mérési modell I.

- ▶ Távolság transzformáció: átlagos eltérés mérhető két pontthalmaz között
- ▶ Két pontthalmaz: foglalt cellák a térképen és a LiDAR sugarak „végpontjai”
- ▶ Eltérés számítása (skalár):  $\psi(\mathbf{x}, \mathbf{z}, \mathbf{m})$
- ▶ Ideális eset (minden végpont foglalt cellán):  $\psi(\mathbf{x}, \mathbf{z}, \mathbf{m}) = 0$
- ▶  $\psi$  deriváltjai már számíthatóak: megfelel az Exact Flow Daum–Huang (EDH) szűrőhöz



# Távolság-transzformáció alapú mérési modell I.

- ▶ Távolság transzformáció: átlagos eltérés mérhető két pontthalmaz között
- ▶ Két pontthalmaz: foglalt cellák a térképen és a LiDAR sugarak „végpontjai”
- ▶ Eltérés számítása (skalár):  $\psi(\mathbf{x}, \mathbf{z}, \mathbf{m})$
- ▶ Ideális eset (minden végpont foglalt cellán):  $\psi(\mathbf{x}, \mathbf{z}, \mathbf{m}) = 0$
- ▶  $\psi$  deriváltjai már számíthatóak: megfelel az Exact Flow Daum–Huang (EDH) szűrőhöz
- ▶ **Az EDH-t módosítani kellett, hogy kezeljen implicit mérési egyenletet (eredetileg  $\mathbf{z} = h(\mathbf{x})$  alak)**

# Daum–Huang-szűrő (DHF)

- ▶ Korrigálás (priorból poszteriorba)
  - ▶ részecskeszűrő: újramintavételezés (egyfajta „ugrás”)
  - ▶ DHF: „ugrás” helyett **részecskefolyam**

## Daum–Huang-szűrő (DHF)

- ▶ Korrigálás (priorból poszteriorba)
  - ▶ részecskeszűrő: újramintavételezés (egyfajta „ugrás”)
  - ▶ DHF: „ugrás” helyett **részecskefolyam**
- ▶ Folytonos átmenet (homotópia) priorból poszteriorba:

$$\log p(\mathbf{x}, \lambda) = \log \underbrace{g(\mathbf{x})}_{\text{prior}} + \lambda \log \underbrace{h(\mathbf{x})}_{\text{szenzormodell}} - \log \underbrace{K(\lambda)}_{\text{norm. konstans}} .$$

- ▶  $\lambda = 0$ : prior,  $\lambda = 1$ : poszterior

## Daum–Huang-szűrő (DHF)

- ▶ Korrigálás (priorból poszteriorba)
  - ▶ részecskeszűrő: újramintavételezés (egyfajta „ugrás”)
  - ▶ DHF: „ugrás” helyett **részecskefolyam**
- ▶ Folytonos átmenet (homotópia) priorból poszteriorba:

$$\log p(\mathbf{x}, \lambda) = \log \underbrace{g(\mathbf{x})}_{\text{prior}} + \lambda \log \underbrace{h(\mathbf{x})}_{\text{szenzormodell}} - \log \underbrace{K(\lambda)}_{\text{norm. konstans}} .$$

- ▶  $\lambda = 0$ : prior,  $\lambda = 1$ : poszterior
- ▶ Részecskék mozgását egy SDE írja le ( $\lambda$ : pszeudoidő):

$$d\mathbf{x}_i = \mathbf{f}(\mathbf{x}_i, \lambda)d\lambda + \boldsymbol{\sigma}(\mathbf{x}_i, \lambda)d\mathbf{W}_\lambda, \quad (3)$$

## Daum–Huang-szűrő (DHF)

- ▶ Korrigálás (priorból poszteriorba)
  - ▶ részecskeszűrő: újramintavételezés (egyfajta „ugrás”)
  - ▶ DHF: „ugrás” helyett **részecskefolyam**
- ▶ Folytonos átmenet (homotópia) priorból poszteriorba:

$$\log p(\mathbf{x}, \lambda) = \log \underbrace{g(\mathbf{x})}_{\text{prior}} + \lambda \log \underbrace{h(\mathbf{x})}_{\text{szenzormodell}} - \log \underbrace{K(\lambda)}_{\text{norm. konstans}} .$$

- ▶  $\lambda = 0$ : prior,  $\lambda = 1$ : poszterior
- ▶ Részecskék mozgását egy SDE írja le ( $\lambda$ : pszeudoidő):

$$d\mathbf{x}_i = \mathbf{f}(\mathbf{x}_i, \lambda)d\lambda + \boldsymbol{\sigma}(\mathbf{x}_i, \lambda)d\mathbf{W}_\lambda, \quad (3)$$

- ▶ Kérdés: mi  $\mathbf{f}$ , (mi  $\boldsymbol{\sigma}$ )?
- ▶ Különböző Daum–Huang-szűrő variációk más-más megoldást adnak.

# Exact Flow Daum–Huang-szűrő (EDH)

- ▶ Hanyagoljuk el a sztochasztikus részt, így

$$\frac{d\mathbf{x}_i}{d\lambda} = \mathbf{f}(\mathbf{x}_i, \lambda). \quad (4)$$

# Exact Flow Daum–Huang-szűrő (EDH)

- ▶ Hanyagoljuk el a sztochasztikus részt, így

$$\frac{d\mathbf{x}_i}{d\lambda} = \mathbf{f}(\mathbf{x}_i, \lambda). \quad (4)$$

- ▶ Legyen  $\mathbf{f}$  alakja

$$\mathbf{f}(\mathbf{x}_i, \lambda) = \mathbf{C}(\lambda)\mathbf{x}_i + \mathbf{c}(\lambda). \quad (5)$$

## Exact Flow Daum–Huang-szűrő (EDH)

- ▶ Hanyagoljuk el a sztochasztikus részt, így

$$\frac{d\mathbf{x}_i}{d\lambda} = \mathbf{f}(\mathbf{x}_i, \lambda). \quad (4)$$

- ▶ Legyen  $\mathbf{f}$  alakja

$$\mathbf{f}(\mathbf{x}_i, \lambda) = \mathbf{C}(\lambda)\mathbf{x}_i + \mathbf{c}(\lambda). \quad (5)$$

- ▶ Együtthatók meghatározása:

$$\mathbf{C}(\lambda) = -\frac{1}{2}\bar{\Sigma}\nabla\psi_x^\top \left( \lambda\nabla\psi_x\bar{\Sigma}\nabla\psi_x^\top + \mathbf{R} \right)^{-1} \nabla\psi_x, \quad (6)$$

$$\mathbf{c}(\lambda) = (\mathbf{I} + 2\lambda\mathbf{C}) \left[ (\mathbf{I} + \lambda\mathbf{C}) \bar{\Sigma}\nabla\psi_x^\top \mathbf{R}^{-1}\mathbf{z}' + \mathbf{C}\bar{\mathbf{x}} \right]. \quad (7)$$



## Exact Flow Daum–Huang-szűrő (EDH)

- ▶ Hanyagoljuk el a sztochasztikus részt, így

$$\frac{d\mathbf{x}_i}{d\lambda} = \mathbf{f}(\mathbf{x}_i, \lambda). \quad (4)$$

- ▶ Legyen  $\mathbf{f}$  alakja

$$\mathbf{f}(\mathbf{x}_i, \lambda) = \mathbf{C}(\lambda)\mathbf{x}_i + \mathbf{c}(\lambda). \quad (5)$$

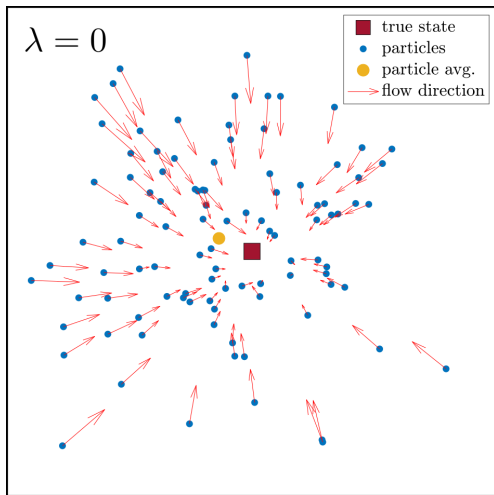
- ▶ Együtthatók meghatározása:

$$\mathbf{C}(\lambda) = -\frac{1}{2}\bar{\Sigma}\nabla\psi_x^\top \left( \lambda\nabla\psi_x\bar{\Sigma}\nabla\psi_x^\top + \mathbf{R} \right)^{-1} \nabla\psi_x, \quad (6)$$

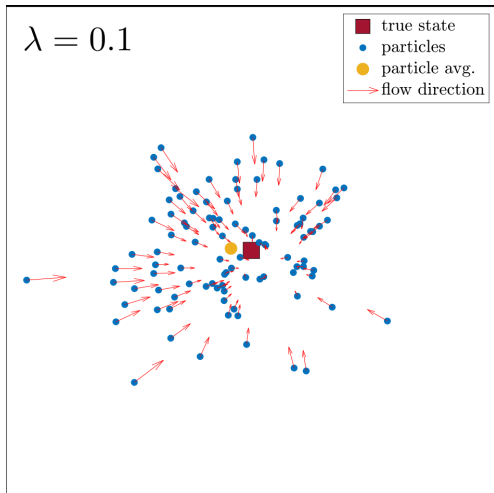
$$\mathbf{c}(\lambda) = (\mathbf{I} + 2\lambda\mathbf{C}) \left[ (\mathbf{I} + \lambda\mathbf{C}) \bar{\Sigma}\nabla\psi_x^\top \mathbf{R}^{-1}\mathbf{z}' + \mathbf{C}\bar{\mathbf{x}} \right]. \quad (7)$$

- ▶ Innentől adott a részecskék mozgása: Euler integrálása (4)-nek

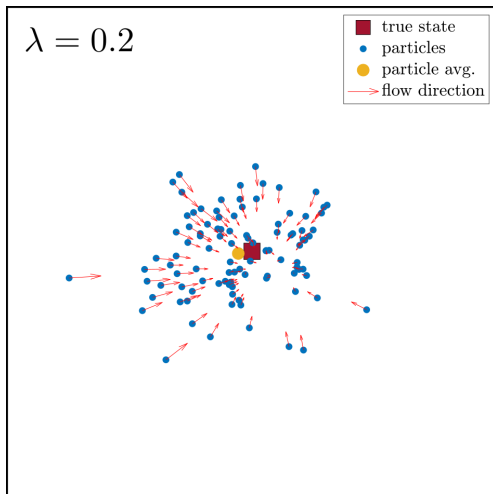
# EDH: részecskék mozgása



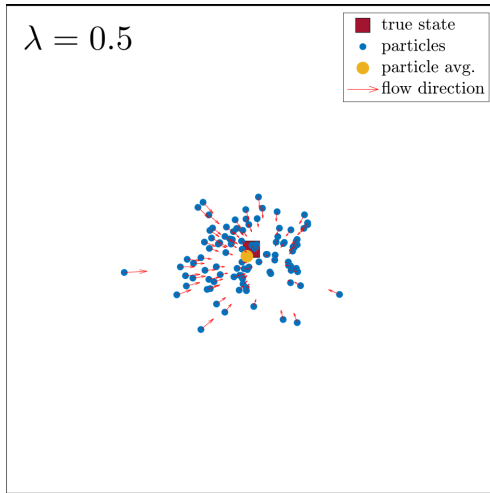
# EDH: részecskék mozgása



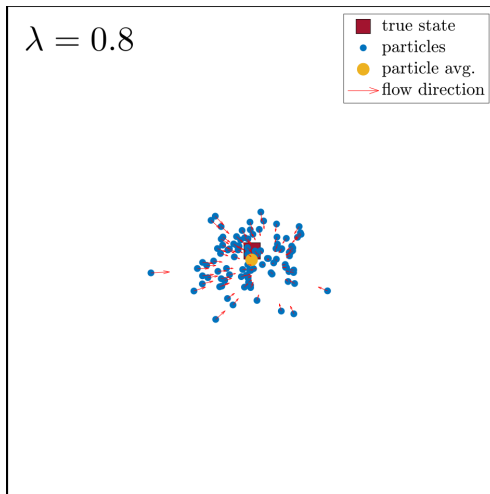
# EDH: részecskék mozgása



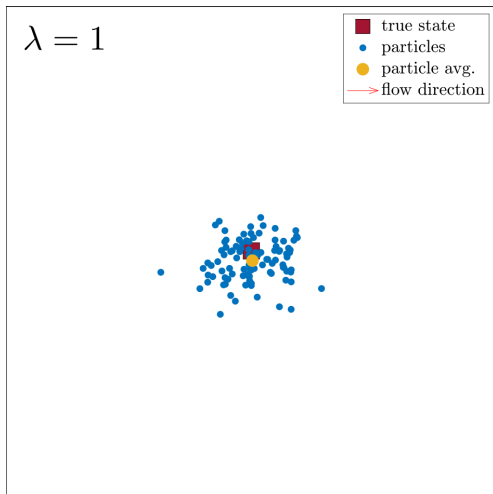
# EDH: részecskék mozgása



# EDH: részecskék mozgása

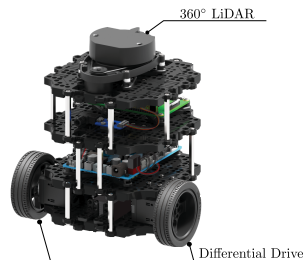


# EDH: részecskék mozgása



# Tesztelési összeállítás

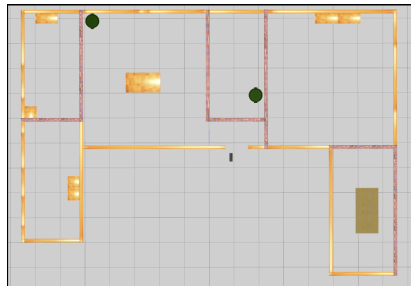
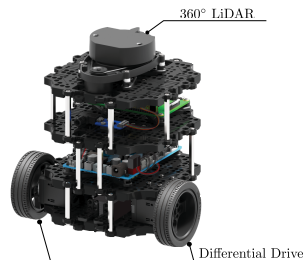
- ▶ ROBOTIS Turtlebot3 Burger
- ▶ Gazebo szimulátor + Robot Operating System (ROS)





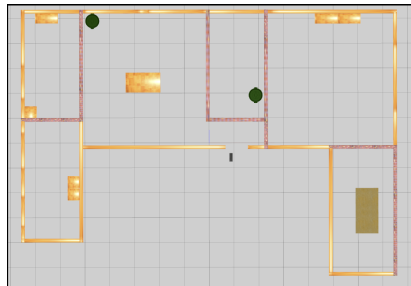
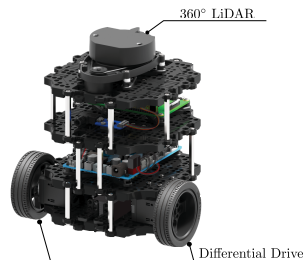
# Tesztelési összeállítás

- ▶ ROBOTIS Turtlebot3 Burger
- ▶ Gazebo szimulátor + Robot Operating System (ROS)
- ▶ Környezet: Turtlebot3 House



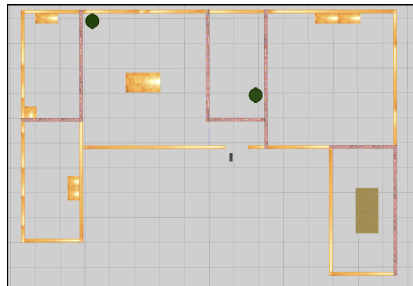
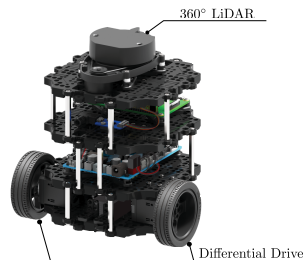
# Tesztelési összeállítás

- ▶ ROBOTIS Turtlebot3 Burger
- ▶ Gazebo szimulátor + Robot Operating System (ROS)
- ▶ Környezet: Turtlebot3 House
- ▶ Kb. 7-8 percnyi barangolás (manuálisan)



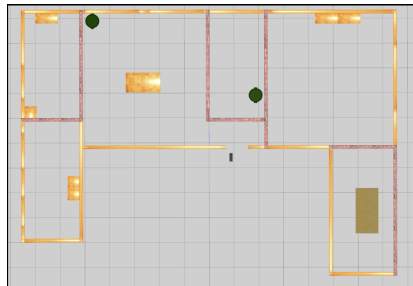
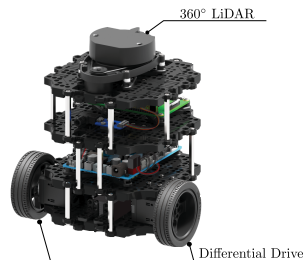
## Tesztelési összeállítás

- ▶ ROBOTIS Turtlebot3 Burger
- ▶ Gazebo szimulátor + Robot Operating System (ROS)
- ▶ Környezet: Turtlebot3 House
- ▶ Kb. 7-8 percnyi barangolás (manuálisan)
- ▶ Valós póz és becsült póz összehasonlítása

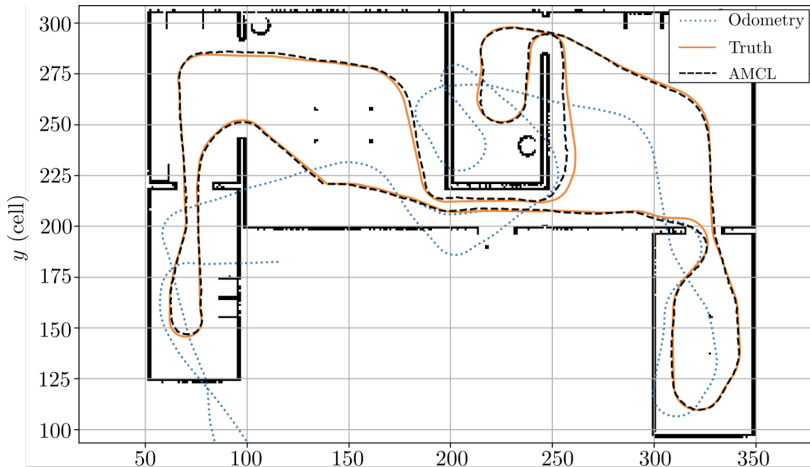


# Tesztelési összeállítás

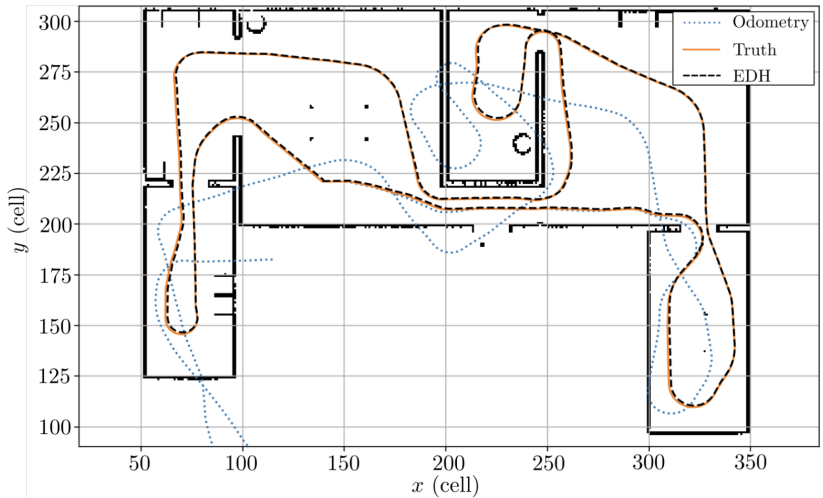
- ▶ ROBOTIS Turtlebot3 Burger
- ▶ Gazebo szimulátor + Robot Operating System (ROS)
- ▶ Környezet: Turtlebot3 House
- ▶ Kb. 7-8 percnyi barangolás (manuálisan)
- ▶ Valós póz és becsült póz összehasonlítása
- ▶ Algoritmusok: AMCL, és a készített EDH alapú



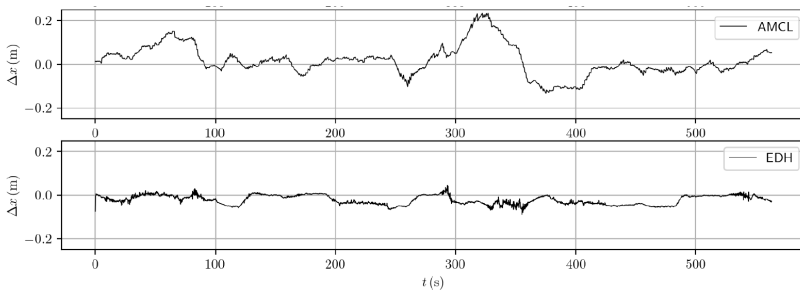
# AMCL eredmény



# EDH eredmény

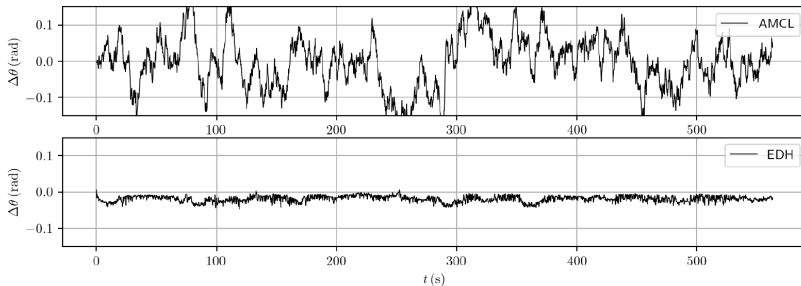


# Pozíció hiba ( $x$ irány)



Algoritmus	$\kappa$ (m)	Részecske szám	Root Mean Squared Error		Mean Absolute Error	
			pozíció (m)	orientáció (rad)	pozíció (m)	orientáció (rad)
AMCL	$10^{-4}$	adaptív	0.0816	0.0708	0.0672	0.0547
EDH	$10^{-4}$	10	0.0439	0.0205	<b>0.0381</b>	0.0188
		100	<b>0.0437</b>	0.0206	0.0383	0.0190
		500	0.0438	0.0206	0.0382	0.0190

# Orientáció hiba



Algoritmus	$\kappa$ (m)	Részecske szám	Root Mean Squared Error		Mean Absolute Error	
			pozíció (m)	orientáció (rad)	pozíció (m)	orientáció (rad)
AMCL	$10^{-4}$	adaptív	0.0816	0.0708	0.0672	0.0547
EDH	$10^{-4}$	10	0.0439	<b>0.0205</b>	0.0381	<b>0.0188</b>
		100	0.0437	0.0206	0.0383	0.0190
		500	0.0438	0.0206	0.0382	0.0190



# Összegzés

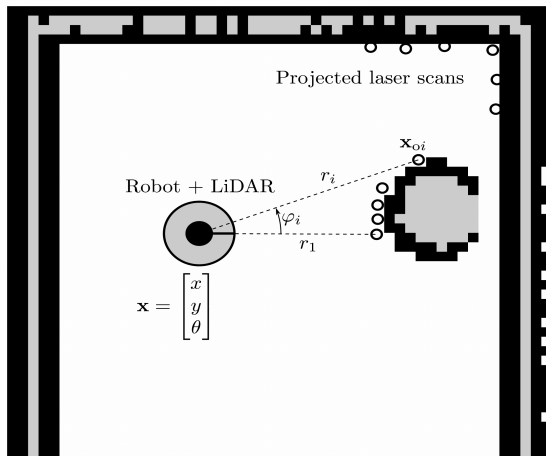
- ▶ Magabiztosan jobb eredmények, mint az AMCL esetén
- ▶ Képes valós idejű futásra: 439 s-nyi mérési adatra 40.4 s

# Összegzés

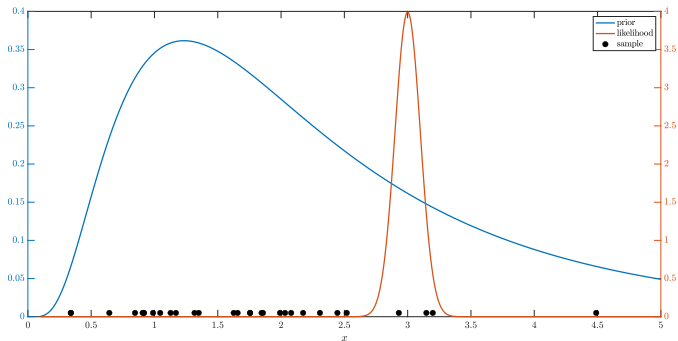
- ▶ Magabiztosan jobb eredmények, mint az AMCL esetén
- ▶ Képes valós idejű futásra: 439 s-nyi mérési adatra 40.4 s
- ▶ Jövőben:
  - ▶ Robusztusság tesztelése (pl. „elrabolt robot” probléma)
  - ▶ Egyéb DHF variációk (pl. sztochasztikus rész)
  - ▶ Futásidő csökkentése
  - ▶ Futtatás valós mérési adatokon
  - ▶ Az eredmények publikálása

# Köszönöm a figyelmet!

# LiDAR mérések projekciója



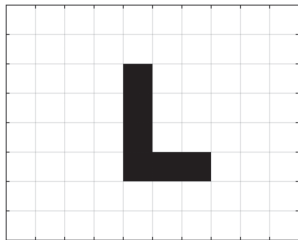
# Részecske degeneráció



# Részletes eredmények

Algorithm	$\kappa$ (m)	Particle num.	Root Mean Squared Error		Mean Absolute Error	
			position (m)	orientation (rad)	position (m)	orientation (rad)
EKF	$10^{-6}$	-	0.0559	<b>0.0212</b>	0.0492	<b>0.0168</b>
AMCL	$10^{-6}$	adaptive	0.0529	0.0233	0.0453	0.0177
EDH	$10^{-6}$	10	0.0504	0.0228	0.0449	0.0183
		100	0.0504	0.0229	0.0447	0.0183
		500	<b>0.0498</b>	0.0231	<b>0.0441</b>	0.0185
EKF	$10^{-4}$	-	0.0497	<b>0.0186</b>	0.0428	<b>0.0159</b>
AMCL	$10^{-4}$	adaptive	0.0816	0.0708	0.0672	0.0547
EDH	$10^{-4}$	10	0.0439	0.0205	<b>0.0381</b>	0.0188
		100	<b>0.0437</b>	0.0206	0.0383	0.0190
		500	0.0438	0.0206	0.0382	0.0190
EKF	$5 \cdot 10^{-3}$	-	0.0732	<b>0.0297</b>	0.0610	<b>0.0256</b>
AMCL	$5 \cdot 10^{-3}$	adaptive	0.5261	0.4940	0.3533	0.3841
EDH	$5 \cdot 10^{-3}$	10	<b>0.0587</b>	0.0401	<b>0.0494</b>	0.0372
		100	0.0616	0.0403	0.0503	0.0374
		500	0.0617	0.0402	0.0503	0.0373

# Távolság-transzformáció alapú mérési modell (kieg.)



$$DF(\mathbf{x}, \mathbf{m}) = \min_{\mathbf{v}_j \in V} \|\mathbf{x} - \mathbf{v}_j\|, \quad (8)$$

$$\mathbf{x}_{oi} = \begin{bmatrix} x_{oi} \\ y_{oi} \end{bmatrix} = \begin{bmatrix} x + r_i \cos(\varphi_i + \theta) \\ y + r_i \sin(\varphi_i + \theta) \end{bmatrix}, \quad (9)$$

$$\psi(\mathbf{x}, \mathbf{z}, \mathbf{m}) = \frac{1}{n} \sum_{i=1}^n DF(\mathbf{x}_{oi}, \mathbf{m}). \quad (10)$$

