TOP_BKN_NCLT_dataset

December 26, 2025

[1]:
```
from pathlib import Path
from scipy.io import loadmat
import sys
import os

notebook_path = os.getcwd()
print (f"Current notebook path: {notebook_path}")
project_root = os.path.dirname(notebook_path)
if project_root not in sys.path:
    sys.path.insert(0, project_root)
print (f"Added {project_root} to sys.path")
```

```
Current notebook path: /home/luky/skola/KalmanNet-main/navigation NCLT dataset
Added /home/luky/skola/KalmanNet-main to sys.path
```

[2]:
```
import torch
import matplotlib.pyplot as plt
from utils import trainer
from utils import utils
from Systems import DynamicSystem
import Filters
import torch.nn.functional as F
from torch.utils.data import TensorDataset, DataLoader
import numpy as np
from scipy.io import loadmat
from scipy.interpolate import RegularGridInterpolator
import random

torch.manual_seed(42)
np.random.seed(42)
random.seed(42)
if torch.cuda.is_available():
    torch.cuda.manual_seed_all(42)

device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
DEVICE = device  # For backward compatibility
print(f"device: {device}")
```

```
device: cuda
```

[3]:
```
import torch
import torch.nn as nn
import torch.nn.functional as F
from torch.utils.data import TensorDataset, DataLoader
import numpy as np
import matplotlib.pyplot as plt
import os
import Systems


# Parametry sekvencí
TRAIN_SEQ_LEN = 100       # Délka sekvence pro trénink (např. 100 kroků = 100␣
 ↪sekund při 1Hz)
VAL_SEQ_LEN = 250
TEST_SEQ_LEN = 1000      # Délka sekvence pro testování (delší sekvence pro␣
 ↪stabilnější vyhodnocení)
STRIDE = 20          # Posun okna (překryv) pro data augmentation
BATCH_SIZE = 128
DATA_PATH = 'data/processed'
print(f"Běží na zařízení: {device}")
```

```
Běží na zařízení: cuda
```

[4]:
```
import torch
from torch.utils.data import TensorDataset, DataLoader
import os

def prepare_sequences(dataset_list, seq_len, stride, mode='train'):
    """
    Zpracuje list trajektorií na sekvence pro trénink dle článku.

    Nový formát dle [Song et al., 2024]:
    - Vstup u (4D): [v_left, v_right, theta_imu, omega_imu]
    - Cíl x (6D):   [px, py, vx, vy, theta, omega]
    """
    X_seq_list = [] # Ground Truth (Cíl)
    Y_seq_list = [] # GPS Měření (Vstup do korekce)
    U_seq_list = [] # Control Input (IMU/Odo)

    print(f"Zpracovávám {len(dataset_list)} trajektorií pro {mode}...")

    for traj in dataset_list:
        # 1. Extrahuje data
        # GT z preprocessingu je [px, py, theta]
        gt = traj['ground_truth'].float()
        # GPS: [x, y] (obsahuje NaN!)
        gps = traj['filtered_gps'].float()

        # IMU: [ax, ay, theta, omega]
        imu = traj['imu'].float()
        theta_imu = imu[:, 2] # Orientace z IMU
        omega_imu = imu[:, 3] # Úhlová rychlost z IMU

        # ODO: [v_left, v_right]
        odo = traj['filtered_wheel'].float()

        # Fix NaN v odometrii (nahradíme nulou)
        v_left = torch.nan_to_num(odo[:, 0], nan=0.0)
        v_right = torch.nan_to_num(odo[:, 1], nan=0.0)

        # 2. Sestavení vstupu u = [v_l, v_r, theta_imu, omega_imu] (4D)
        # Toto odpovídá "State Model" definovanému v článku (sekce II.C.2)
        u = torch.stack((v_left, v_right, theta_imu, omega_imu), dim=1)

        # 3. Sestavení cíle x (6D) pro state vector [px, py, vx, vy, theta,␣
↪omega]
        # Vyplníme to, co máme z Ground Truth (px, py, theta).
        # Rychlosti (vx, vy, omega) v GT implicitně nemáme (nebo je složité je␣
↪derivovat přesně),
        # ale pro trénink Loss funkce budeme stejně porovnávat primárně pozici.
        T = gt.shape[0]
        x_target = torch.zeros(T, 6)
        x_target[:, 0] = gt[:, 0] # px
        x_target[:, 1] = gt[:, 1] # py
        x_target[:, 4] = gt[:, 2] # theta
        # Ostatní (vx, vy, omega) zůstávají 0, protože v Loss funkci budeme␣
↪maskovat nebo brát jen pozici.

        # 4. Sliding Window (Rozsekání na sekvence)
        num_samples = gt.shape[0]
        current_stride = stride if mode == 'train' else seq_len # U testu bez␣
↪překryvu

        for i in range(0, num_samples - seq_len + 1, current_stride):
            # Cíl: 6D stav
            x_seq = x_target[i : i+seq_len, :]

            # Měření: GPS [px, py]
            y_seq = gps[i : i+seq_len, :]

            # Vstup: 4D control input
            u_seq = u[i : i+seq_len, :]

            X_seq_list.append(x_seq)
            Y_seq_list.append(y_seq)
            U_seq_list.append(u_seq)

    # Stack do tenzorů
    X_out = torch.stack(X_seq_list)
    Y_out = torch.stack(Y_seq_list)
    U_out = torch.stack(U_seq_list)

    return X_out, Y_out, U_out

# === NAČTENÍ DAT ===
# Ujistíme se, že cesty a konstanty jsou definované (pokud nejsou, doplňte je␣
 ↪nahoře)
# if 'DATA_PATH' not in locals(): DATA_PATH = 'data/processed'
# if 'TRAIN_SEQ_LEN' not in locals(): TRAIN_SEQ_LEN = 100
# if 'VAL_SEQ_LEN' not in locals(): VAL_SEQ_LEN = 200
# if 'TEST_SEQ_LEN' not in locals(): TEST_SEQ_LEN = 500
# if 'STRIDE' not in locals(): STRIDE = 20
# if 'BATCH_SIZE' not in locals(): BATCH_SIZE = 256

train_data_raw = torch.load(os.path.join(DATA_PATH, 'train.pt'))
val_data_raw = torch.load(os.path.join(DATA_PATH, 'val.pt'))
test_data_raw = torch.load(os.path.join(DATA_PATH, 'test.pt'))
# Načtení celého balíku
# train_data_full = torch.load(os.path.join(DATA_PATH, 'train.pt'))

# === RYCHLÝ TEST: OŘÍZNUTÍ DAT ===
# Vezmeme jen prvních 5 trajektorií z 22.
# To radikálně zrychlí jednu epochu a umožní rychle otestovat stabilitu␣
 ↪hyperparametrů.
# train_data_raw = train_data_full[:10]

print(f"DEBUG: Pro rychlý test používám jen {len(train_data_raw)} trajektorií.")
# ... zbytek kódu beze změny

# === PŘÍPRAVA SEKVENCÍ ===
print("--- Generuji trénovací data (Paper compatible) ---")
train_X, train_Y, train_U = prepare_sequences(train_data_raw, TRAIN_SEQ_LEN,␣
 ↪STRIDE, 'train')

print("\n--- Generuji validační data ---")
val_X, val_Y, val_U = prepare_sequences(val_data_raw, VAL_SEQ_LEN, VAL_SEQ_LEN,␣
 ↪'val')
```

```
print("\n--- Generuji testovací data ---")
test_X, test_Y, test_U = prepare_sequences(test_data_raw, TEST_SEQ_LEN,␣
 ↪TEST_SEQ_LEN, 'test')

# Vytvoření DataLoaderů
train_loader = DataLoader(TensorDataset(train_X, train_Y, train_U),␣
 ↪batch_size=BATCH_SIZE, shuffle=True)
val_loader = DataLoader(TensorDataset(val_X, val_Y, val_U),␣
 ↪batch_size=BATCH_SIZE, shuffle=False)
test_loader = DataLoader(TensorDataset(test_X, test_Y, test_U),␣
 ↪batch_size=BATCH_SIZE, shuffle=False)

print(f"\n  Data připravena.")
print(f"Train batches: {len(train_loader)}")
print(f"Shapes -> X: {train_X.shape} (6D State), U: {train_U.shape} (4D Input),␣
 ↪Y: {train_Y.shape} (2D Meas)")
```

```
DEBUG: Pro rychlý test používám jen 22 trajektorií.
--- Generuji trénovací data (Paper compatible) ---
Zpracovávám 22 trajektorií pro train…

--- Generuji validační data ---
Zpracovávám 2 trajektorií pro val…

--- Generuji testovací data ---
Zpracovávám 3 trajektorií pro test…

  Data připravena.
Train batches: 43
Shapes -> X: torch.Size([5446, 100, 6]) (6D State), U: torch.Size([5446, 100,
4]) (4D Input), Y: torch.Size([5446, 100, 2]) (2D Meas)
```

[5]:
```
# === INICIALIZACE DYNAMICKÉHO MODELU (System Instance - Paper Version) ===

# 1. Parametry systému podle článku [Song et al., 2024]
# State (6D): [px, py, vx, vy, theta, omega]
# Referenční rovnice (5) v článku.
state_dim = 6
# Meas (2D):  [gps_x, gps_y]
# Referenční rovnice (6) v článku.
obs_dim = 2
# Časový krok (z preprocessingu)
dt = 1.0

# 2. Definice Matice Q (Procesní šum / Model Uncertainty)
# Nyní máme 6 stavů. Musíme definovat nejistotu pro každý z nich.
# Hodnoty jsou nastaveny heuristicky (lze ladit):
```

```
# - Pozice (idx 0,1): 0.1
# - Rychlost (idx 2,3): 0.1
# - Úhel/Omega (idx 4,5): 0.01 (IMU je v NCLT docela přesné, ale driftuje)
q_diag = torch.tensor([0.1, 0.1, 0.1, 0.1, 0.01, 0.01])
Q = torch.diag(q_diag)

# 3. Definice Matice R (Šum měření / Sensor Noise)
# GPS měří jen pozici (px, py).
# Nastavujeme 1.0 m^2. To odpovídá standardní odchylce 1m.
# Pokud je GPS v datasetu horší, KalmanNet se naučí "nedůvěřovat" vstupu y
# a spoléhat více na predikci z u (odometrie).
r_diag = torch.tensor([1.0, 1.0])
R = torch.diag(r_diag)

# 4. Počáteční podmínky (Prior)
# Ex0: Nulový vektor 6x1
Ex0 = torch.zeros(state_dim, 1)

# P0: Počáteční kovariance
# Autoři používají P k inicializaci EKF[cite: 700].
# Nastavíme rozumnou počáteční nejistotu.
P0 = torch.eye(state_dim) * 0.5

# 5. Vytvoření instance DynamicSystemNCLT
# Důležité: f=None zajistí, že se použije interní `_f_paper_dynamics` (rovnice␣
↪5),
# která očekává 4D vstup (v_l, v_r, theta, omega).
sys_model = Systems.DynamicSystemNCLT(
    state_dim=state_dim,
    obs_dim=obs_dim,
    Q=Q,
    R=R,
    Ex0=Ex0,
    P0=P0,
    dt=dt,
    f=None,  # None -> Použije se model z článku: px += vc*cos(theta_imu)...
    h=None,  # None -> Použije se GPS model: y = [px, py]
    device=DEVICE
)

print(f" System Model NCLT inicializován (Paper Version).")
print(f" - State Dim: {sys_model.state_dim} [px, py, vx, vy, theta, omega]")
print(f" - Meas Dim:  {sys_model.obs_dim} [gps_x, gps_y]")
print(f" - Input Dim: 4 [v_l, v_r, theta_imu, omega_imu]") # Implicitně v modelu
print(f" - Q Diag: {q_diag.tolist()}")
```

```
 System Model NCLT inicializován (Paper Version).
 - State Dim: 6 [px, py, vx, vy, theta, omega]
 - Meas Dim:  2 [gps_x, gps_y]
 - Input Dim: 4 [v_l, v_r, theta_imu, omega_imu]
 - Q Diag: [0.10000000149011612, 0.10000000149011612, 0.10000000149011612,
0.10000000149011612, 0.009999999776482582, 0.009999999776482582]
```

[7]:
```
import torch
import torch.optim as optim
import os
from state_NN_models import NCLT
from utils import trainer

state_knet = NCLT.StateBayesianKalmanNetNCLT(
    system_model=sys_model,
    device=DEVICE,
    hidden_size_multiplier=8,       # Větší kapacita pro složitější dynamiku
    output_layer_multiplier=4,
    num_gru_layers=1,                # 1 vrstva GRU obvykle stačí a je␣
  ↪stabilnější
    init_min_dropout=0.2,
    init_max_dropout=0.4
).to(DEVICE)
print(state_knet)

# Počet trénovatelných parametrů
params_count = sum(p.numel() for p in state_knet.parameters() if p.
  ↪requires_grad)
print(f"Model má {params_count} trénovatelných parametrů.")
# === 3. PŘÍPRAVA A SPUŠTĚNÍ TRÉNINKU ===
print("\n Spouštím tréninkovou smyčku (Vektorizovanou)...")

# 1. Zjistíme velikost datasetu (počet batchů v jedné epoše)
num_train_batches = len(train_loader)
print(f"INFO: 1 Epocha = {num_train_batches} iterací (batchů).")

# 2. Nastavení parametrů
EPOCHS = 100
WARMUP_EPOCHS = 15  # Prvních 15 epoch jen MSE, aby se model stabilizoval

# Přepočet na iterace (protože nová funkce počítá v iteracích)
total_iterations = EPOCHS * num_train_batches
total_iterations = 800
warmup_iterations = WARMUP_EPOCHS * num_train_batches
warmup_iterations=350
validation_period = 5  # Chceme validovat jednou za epochu (na konci)

# 3. Volání funkce
```

```python
trained_knet = trainer.train_BayesianKalmanNet(
    model=state_knet,
    train_loader=train_loader,
    val_loader=val_loader,
    device=DEVICE,

    # Parametry iterací
    total_train_iter=total_iterations,
    learning_rate=5e-4,        # Opatrný LR pro BKN
    clip_grad=1.0,             # Stabilizace gradientů
    J_samples=10,               # Počet MC vzorků (paralelně)

    # Logování a validace
    validation_period=validation_period,
    logging_period=10,        # Výpis loss každých 10 batchů

    # Warmup
    warmup_iterations=warmup_iterations,
    weight_decay_=1e-5
)
```

```
INFO: Aplikuji 'Start Zero' inicializaci pro Kalman Gain.
DEBUG: Výstupní vrstva vynulována (Soft Start).
StateBayesianKalmanNetNCLT(
  (dnn): DNN_BayesianKalmanNetNCLT(
    (input_layer): Sequential(
      (0): Linear(in_features=16, out_features=512, bias=True)
      (1): ReLU()
    )
    (concrete_dropout1): ConcreteDropout()
    (gru): GRU(512, 160)
    (output_layer): Sequential(
      (0): Linear(in_features=160, out_features=48, bias=True)
      (1): ReLU()
      (2): Linear(in_features=48, out_features=12, bias=True)
    )
    (concrete_dropout2): ConcreteDropout()
  )
)
Model má 340542 trénovatelných parametrů.

 Spouštím tréninkovou smyčku (Vektorizovanou)…
INFO: 1 Epocha = 43 iterací (batchů).

[DIAGNOSTIKA iter 0]
  -> Rozdíl (Pred - Target): 1.813148
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  2.47e-03

--- Validation at iteration 5 ---
  Average MSE: 20.8313, Average ANEES: 335.7215
  >>> New best VALIDATION ANEES! Saving model. <<<
--------------------------------------------------
--- Iteration [10/800] ---
    - Total Loss: 7.7659
    - NLL: 0.0000
    - Reg: 0.1679
    - p1=0.374, p2=0.399

--- Validation at iteration 10 ---
  Average MSE: 37.3800, Average ANEES: 191.9882
  >>> New best VALIDATION ANEES! Saving model. <<<
--------------------------------------------------

[DIAGNOSTIKA iter 10]
  -> Rozdíl (Pred - Target): 1.335995
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  2.74e-03

--- Validation at iteration 15 ---
  Average MSE: 21.0435, Average ANEES: 246.1053
--------------------------------------------------
--- Iteration [20/800] ---
    - Total Loss: 7.6589
    - NLL: 0.0000
    - Reg: 0.1675
    - p1=0.374, p2=0.398

--- Validation at iteration 20 ---
  Average MSE: 14.5628, Average ANEES: 227.1221
--------------------------------------------------

[DIAGNOSTIKA iter 20]
  -> Rozdíl (Pred - Target): 1.152306
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  3.25e-03

--- Validation at iteration 25 ---
  Average MSE: 8.2143, Average ANEES: 188.3475
  >>> New best VALIDATION ANEES! Saving model. <<<
--------------------------------------------------
--- Iteration [30/800] ---
    - Total Loss: 4.5368
    - NLL: 0.0000
    - Reg: 0.1671
    - p1=0.373, p2=0.397

--- Validation at iteration 30 ---
  Average MSE: 15.4657, Average ANEES: 263.3331
--------------------------------------------------

[DIAGNOSTIKA iter 30]
  -> Rozdíl (Pred - Target): 1.159997
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  2.43e-03

--- Validation at iteration 35 ---
  Average MSE: 12.8131, Average ANEES: 198.9188
--------------------------------------------------
--- Iteration [40/800] ---
    - Total Loss: 6.9037
    - NLL: 0.0000
    - Reg: 0.1666
    - p1=0.372, p2=0.396

--- Validation at iteration 40 ---
  Average MSE: 15.6414, Average ANEES: 322.0733
--------------------------------------------------

[DIAGNOSTIKA iter 40]
  -> Rozdíl (Pred - Target): 1.199098
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  2.14e-03

--- Validation at iteration 45 ---
  Average MSE: 11.3383, Average ANEES: 185.9488
  >>> New best VALIDATION ANEES! Saving model. <<<
--------------------------------------------------
--- Iteration [50/800] ---
    - Total Loss: 5.4543
    - NLL: 0.0000
    - Reg: 0.1661
    - p1=0.371, p2=0.396

--- Validation at iteration 50 ---
  Average MSE: 11.5102, Average ANEES: 218.3388
--------------------------------------------------

[DIAGNOSTIKA iter 50]
  -> Rozdíl (Pred - Target): 1.112368
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  1.02e-03
```

```
--- Validation at iteration 55 ---
  Average MSE: 10.1489, Average ANEES: 159.8814
  >>> New best VALIDATION ANEES! Saving model. <<<
--------------------------------------------------
--- Iteration [60/800] ---
    - Total Loss: 6.2605
    - NLL: 0.0000
    - Reg: 0.1656
    - p1=0.370, p2=0.395

--- Validation at iteration 60 ---
  Average MSE: 12.2046, Average ANEES: 124.0069
  >>> New best VALIDATION ANEES! Saving model. <<<
--------------------------------------------------

[DIAGNOSTIKA iter 60]
  -> Rozdíl (Pred - Target): 1.193260
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  3.54e-03

--- Validation at iteration 65 ---
  Average MSE: 8.3135, Average ANEES: 149.7909
--------------------------------------------------
--- Iteration [70/800] ---
    - Total Loss: 4.8238
    - NLL: 0.0000
    - Reg: 0.1652
    - p1=0.369, p2=0.394

--- Validation at iteration 70 ---
  Average MSE: 11.2442, Average ANEES: 122.7932
  >>> New best VALIDATION ANEES! Saving model. <<<
--------------------------------------------------

[DIAGNOSTIKA iter 70]
  -> Rozdíl (Pred - Target): 1.024722
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  3.13e-03

--- Validation at iteration 75 ---
  Average MSE: 11.9845, Average ANEES: 157.0967
--------------------------------------------------
--- Iteration [80/800] ---
    - Total Loss: 4.4905
    - NLL: 0.0000
    - Reg: 0.1649
    - p1=0.368, p2=0.393
```

```
--- Validation at iteration 80 ---
  Average MSE: 12.8258, Average ANEES: 135.1505
--------------------------------------------------

[DIAGNOSTIKA iter 80]
  -> Rozdíl (Pred - Target): 0.998969
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  4.13e-03

--- Validation at iteration 85 ---
  Average MSE: 13.7881, Average ANEES: 132.0691
--------------------------------------------------
--- Iteration [90/800] ---
    - Total Loss: 4.8382
    - NLL: 0.0000
    - Reg: 0.1648
    - p1=0.368, p2=0.393

--- Validation at iteration 90 ---
  Average MSE: 10.8060, Average ANEES: 140.2592
--------------------------------------------------

[DIAGNOSTIKA iter 90]
  -> Rozdíl (Pred - Target): 1.015304
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  1.80e-03

--- Validation at iteration 95 ---
  Average MSE: 7.9230, Average ANEES: 137.3890
--------------------------------------------------
--- Iteration [100/800] ---
    - Total Loss: 4.6629
    - NLL: 0.0000
    - Reg: 0.1646
    - p1=0.368, p2=0.393

--- Validation at iteration 100 ---
  Average MSE: 17.5678, Average ANEES: 115.2179
  >>> New best VALIDATION ANEES! Saving model. <<<
--------------------------------------------------

[DIAGNOSTIKA iter 100]
  -> Rozdíl (Pred - Target): 1.038241
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  2.79e-03

--- Validation at iteration 105 ---
  Average MSE: 10.4049, Average ANEES: 111.3852
  >>> New best VALIDATION ANEES! Saving model. <<<
--------------------------------------------------
--- Iteration [110/800] ---
    - Total Loss: 4.1243
    - NLL: 0.0000
    - Reg: 0.1645
    - p1=0.369, p2=0.392

--- Validation at iteration 110 ---
  Average MSE: 8.2017, Average ANEES: 114.7936
--------------------------------------------------

[DIAGNOSTIKA iter 110]
  -> Rozdíl (Pred - Target): 0.933707
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  3.31e-03

--- Validation at iteration 115 ---
  Average MSE: 10.4019, Average ANEES: 131.8068
--------------------------------------------------
--- Iteration [120/800] ---
    - Total Loss: 4.4941
    - NLL: 0.0000
    - Reg: 0.1643
    - p1=0.369, p2=0.391

--- Validation at iteration 120 ---
  Average MSE: 11.2274, Average ANEES: 118.3047
--------------------------------------------------

[DIAGNOSTIKA iter 120]
  -> Rozdíl (Pred - Target): 1.018751
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  3.32e-03

--- Validation at iteration 125 ---
  Average MSE: 8.4730, Average ANEES: 156.6305
--------------------------------------------------
--- Iteration [130/800] ---
    - Total Loss: 6.8227
    - NLL: 0.0000
    - Reg: 0.1641
    - p1=0.368, p2=0.390

--- Validation at iteration 130 ---
  Average MSE: 13.7900, Average ANEES: 103.3026
  >>> New best VALIDATION ANEES! Saving model. <<<
--------------------------------------------------
```

```
[DIAGNOSTIKA iter 130]
  -> Rozdíl (Pred - Target): 1.030146
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  2.43e-03

--- Validation at iteration 135 ---
  Average MSE: 16.5233, Average ANEES: 115.2235
--------------------------------------------------
--- Iteration [140/800] ---
    - Total Loss: 4.6046
    - NLL: 0.0000
    - Reg: 0.1637
    - p1=0.367, p2=0.390

--- Validation at iteration 140 ---
  Average MSE: 11.0963, Average ANEES: 147.9578
--------------------------------------------------

[DIAGNOSTIKA iter 140]
  -> Rozdíl (Pred - Target): 1.009480
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  3.11e-03

--- Validation at iteration 145 ---
  Average MSE: 7.3468, Average ANEES: 96.2440
  >>> New best VALIDATION ANEES! Saving model. <<<
--------------------------------------------------
--- Iteration [150/800] ---
    - Total Loss: 4.3097
    - NLL: 0.0000
    - Reg: 0.1634
    - p1=0.367, p2=0.389

--- Validation at iteration 150 ---
  Average MSE: 10.5350, Average ANEES: 101.3948
--------------------------------------------------

[DIAGNOSTIKA iter 150]
  -> Rozdíl (Pred - Target): 1.015473
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  2.61e-03

--- Validation at iteration 155 ---
  Average MSE: 10.0813, Average ANEES: 171.5844
--------------------------------------------------
--- Iteration [160/800] ---
    - Total Loss: 4.2172
    - NLL: 0.0000
    - Reg: 0.1631
    - p1=0.366, p2=0.388

--- Validation at iteration 160 ---
  Average MSE: 8.4781, Average ANEES: 113.9435
--------------------------------------------------

[DIAGNOSTIKA iter 160]
  -> Rozdíl (Pred - Target): 0.998008
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  3.25e-03

--- Validation at iteration 165 ---
  Average MSE: 9.7962, Average ANEES: 125.4422
--------------------------------------------------
--- Iteration [170/800] ---
    - Total Loss: 3.9152
    - NLL: 0.0000
    - Reg: 0.1630
    - p1=0.366, p2=0.388

--- Validation at iteration 170 ---
  Average MSE: 10.6910, Average ANEES: 124.6607
--------------------------------------------------

[DIAGNOSTIKA iter 170]
  -> Rozdíl (Pred - Target): 1.055560
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  3.29e-03

--- Validation at iteration 175 ---
  Average MSE: 8.3617, Average ANEES: 113.3452
--------------------------------------------------
--- Iteration [180/800] ---
    - Total Loss: 4.8322
    - NLL: 0.0000
    - Reg: 0.1628
    - p1=0.366, p2=0.387

--- Validation at iteration 180 ---
  Average MSE: 6.4444, Average ANEES: 120.8211
--------------------------------------------------

[DIAGNOSTIKA iter 180]
  -> Rozdíl (Pred - Target): 0.956754
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  2.00e-03
```

```
--- Validation at iteration 185 ---
  Average MSE: 8.7022, Average ANEES: 101.9891
--------------------------------------------------
--- Iteration [190/800] ---
    - Total Loss: 4.0546
    - NLL: 0.0000
    - Reg: 0.1626
    - p1=0.366, p2=0.387

--- Validation at iteration 190 ---
  Average MSE: 8.1826, Average ANEES: 107.2664
--------------------------------------------------

[DIAGNOSTIKA iter 190]
  -> Rozdíl (Pred - Target): 1.048492
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  2.85e-03

--- Validation at iteration 195 ---
  Average MSE: 8.3672, Average ANEES: 98.5544
--------------------------------------------------
--- Iteration [200/800] ---
    - Total Loss: 4.9139
    - NLL: 0.0000
    - Reg: 0.1624
    - p1=0.366, p2=0.386

--- Validation at iteration 200 ---
  Average MSE: 6.6355, Average ANEES: 100.7216
--------------------------------------------------

[DIAGNOSTIKA iter 200]
  -> Rozdíl (Pred - Target): 0.912643
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  2.90e-03

--- Validation at iteration 205 ---
  Average MSE: 8.0258, Average ANEES: 108.0961
--------------------------------------------------
--- Iteration [210/800] ---
    - Total Loss: 3.5279
    - NLL: 0.0000
    - Reg: 0.1622
    - p1=0.365, p2=0.386

--- Validation at iteration 210 ---
  Average MSE: 6.5842, Average ANEES: 103.3507
```

```
--------------------------------------------------

[DIAGNOSTIKA iter 210]
  -> Rozdíl (Pred - Target): 1.082328
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  2.89e-03

--- Validation at iteration 215 ---
  Average MSE: 9.6016, Average ANEES: 111.7953
--------------------------------------------------
--- Iteration [220/800] ---
    - Total Loss: 3.3752
    - NLL: 0.0000
    - Reg: 0.1619
    - p1=0.365, p2=0.385

--- Validation at iteration 220 ---
  Average MSE: 10.0815, Average ANEES: 81.9325
  >>> New best VALIDATION ANEES! Saving model. <<<
--------------------------------------------------

[DIAGNOSTIKA iter 220]
  -> Rozdíl (Pred - Target): 1.016657
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  1.84e-03

--- Validation at iteration 225 ---
  Average MSE: 11.5836, Average ANEES: 101.2669
--------------------------------------------------
--- Iteration [230/800] ---
    - Total Loss: 4.1757
    - NLL: 0.0000
    - Reg: 0.1616
    - p1=0.364, p2=0.384

--- Validation at iteration 230 ---
  Average MSE: 10.6988, Average ANEES: 116.2455
--------------------------------------------------

[DIAGNOSTIKA iter 230]
  -> Rozdíl (Pred - Target): 1.063309
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  1.24e-03

--- Validation at iteration 235 ---
  Average MSE: 11.7545, Average ANEES: 89.1696
--------------------------------------------------
--- Iteration [240/800] ---
```

```
    - Total Loss: 4.3265
    - NLL: 0.0000
    - Reg: 0.1615
    - p1=0.364, p2=0.384

--- Validation at iteration 240 ---
  Average MSE: 8.9114, Average ANEES: 107.0952
--------------------------------------------------

[DIAGNOSTIKA iter 240]
  -> Rozdíl (Pred - Target): 0.972190
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  1.42e-03

--- Validation at iteration 245 ---
  Average MSE: 6.5866, Average ANEES: 98.5746
--------------------------------------------------
--- Iteration [250/800] ---
    - Total Loss: 5.6336
    - NLL: 0.0000
    - Reg: 0.1613
    - p1=0.364, p2=0.383

--- Validation at iteration 250 ---
  Average MSE: 7.0785, Average ANEES: 108.4219
--------------------------------------------------

[DIAGNOSTIKA iter 250]
  -> Rozdíl (Pred - Target): 1.041114
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  2.99e-03

--- Validation at iteration 255 ---
  Average MSE: 8.5588, Average ANEES: 78.0972
  >>> New best VALIDATION ANEES! Saving model. <<<
--------------------------------------------------
--- Iteration [260/800] ---
    - Total Loss: 3.9046
    - NLL: 0.0000
    - Reg: 0.1612
    - p1=0.364, p2=0.383

--- Validation at iteration 260 ---
  Average MSE: 8.0225, Average ANEES: 71.2592
  >>> New best VALIDATION ANEES! Saving model. <<<
--------------------------------------------------

[DIAGNOSTIKA iter 260]
```

```
  -> Rozdíl (Pred - Target): 1.092668
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  2.81e-03

--- Validation at iteration 265 ---
  Average MSE: 10.8454, Average ANEES: 164.0609
--------------------------------------------------
--- Iteration [270/800] ---
    - Total Loss: 12.4752
    - NLL: 0.0000
    - Reg: 0.1610
    - p1=0.364, p2=0.383

--- Validation at iteration 270 ---
  Average MSE: 7.3039, Average ANEES: 100.9889
--------------------------------------------------

[DIAGNOSTIKA iter 270]
  -> Rozdíl (Pred - Target): 0.985548
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  1.41e-03

--- Validation at iteration 275 ---
  Average MSE: 13.2476, Average ANEES: 84.0938
--------------------------------------------------
--- Iteration [280/800] ---
    - Total Loss: 4.3233
    - NLL: 0.0000
    - Reg: 0.1608
    - p1=0.363, p2=0.382

--- Validation at iteration 280 ---
  Average MSE: 11.1309, Average ANEES: 84.9196
--------------------------------------------------

[DIAGNOSTIKA iter 280]
  -> Rozdíl (Pred - Target): 1.060897
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  3.62e-03

--- Validation at iteration 285 ---
  Average MSE: 8.7348, Average ANEES: 105.7923
--------------------------------------------------
--- Iteration [290/800] ---
    - Total Loss: 5.4456
    - NLL: 0.0000
    - Reg: 0.1607
    - p1=0.363, p2=0.382
```

```
--- Validation at iteration 290 ---
  Average MSE: 14.7797, Average ANEES: 78.5443
--------------------------------------------------

[DIAGNOSTIKA iter 290]
  -> Rozdíl (Pred - Target): 0.930990
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  2.22e-03

--- Validation at iteration 295 ---
  Average MSE: 7.7121, Average ANEES: 92.5174
--------------------------------------------------
--- Iteration [300/800] ---
    - Total Loss: 4.4772
    - NLL: 0.0000
    - Reg: 0.1606
    - p1=0.363, p2=0.381

--- Validation at iteration 300 ---
  Average MSE: 10.3512, Average ANEES: 87.4664
--------------------------------------------------

[DIAGNOSTIKA iter 300]
  -> Rozdíl (Pred - Target): 0.946608
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  4.22e-03

--- Validation at iteration 305 ---
  Average MSE: 9.1985, Average ANEES: 109.1824
--------------------------------------------------
--- Iteration [310/800] ---
    - Total Loss: 3.8570
    - NLL: 0.0000
    - Reg: 0.1605
    - p1=0.363, p2=0.381

--- Validation at iteration 310 ---
  Average MSE: 12.2542, Average ANEES: 107.7352
--------------------------------------------------

[DIAGNOSTIKA iter 310]
  -> Rozdíl (Pred - Target): 0.963159
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  8.40e-04

--- Validation at iteration 315 ---
  Average MSE: 8.9155, Average ANEES: 69.9328
  >>> New best VALIDATION ANEES! Saving model. <<<
--------------------------------------------------
--- Iteration [320/800] ---
    - Total Loss: 4.1374
    - NLL: 0.0000
    - Reg: 0.1603
    - p1=0.363, p2=0.381

--- Validation at iteration 320 ---
  Average MSE: 8.1951, Average ANEES: 98.9796
--------------------------------------------------

[DIAGNOSTIKA iter 320]
  -> Rozdíl (Pred - Target): 1.064174
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  1.89e-03

--- Validation at iteration 325 ---
  Average MSE: 7.7121, Average ANEES: 107.3892
--------------------------------------------------
--- Iteration [330/800] ---
    - Total Loss: 5.5166
    - NLL: 0.0000
    - Reg: 0.1602
    - p1=0.363, p2=0.380

--- Validation at iteration 330 ---
  Average MSE: 6.9812, Average ANEES: 76.3379
--------------------------------------------------

[DIAGNOSTIKA iter 330]
  -> Rozdíl (Pred - Target): 0.972974
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  2.80e-03

--- Validation at iteration 335 ---
  Average MSE: 7.5561, Average ANEES: 86.1223
--------------------------------------------------
--- Iteration [340/800] ---
    - Total Loss: 3.6423
    - NLL: 0.0000
    - Reg: 0.1600
    - p1=0.363, p2=0.380

--- Validation at iteration 340 ---
  Average MSE: 9.0256, Average ANEES: 72.1655
--------------------------------------------------
```

```
[DIAGNOSTIKA iter 340]
  -> Rozdíl (Pred - Target): 0.928191
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  2.81e-03

--- Validation at iteration 345 ---
  Average MSE: 7.5566, Average ANEES: 113.3990
--------------------------------------------------
--- Iteration [350/800] ---
    - Total Loss: 3.1682
    - NLL: 0.0000
    - Reg: 0.1599
    - p1=0.362, p2=0.380

--- Validation at iteration 350 ---
  Average MSE: 9.3154, Average ANEES: 61.4551
  >>> New best VALIDATION ANEES! Saving model. <<<
--------------------------------------------------

[DIAGNOSTIKA iter 350]
  -> Rozdíl (Pred - Target): 0.977969
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  2.92e-03

--- Validation at iteration 355 ---
  Average MSE: 16.7465, Average ANEES: 51.6080
  >>> New best VALIDATION ANEES! Saving model. <<<
--------------------------------------------------
--- Iteration [360/800] ---
    - Total Loss: 0.7729
    - NLL: 0.6135
    - Reg: 0.1594
    - p1=0.362, p2=0.380

--- Validation at iteration 360 ---
  Average MSE: 8.8914, Average ANEES: 52.3262
--------------------------------------------------

[DIAGNOSTIKA iter 360]
  -> Rozdíl (Pred - Target): 1.039996
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  3.43e-03

--- Validation at iteration 365 ---
  Average MSE: 9.4874, Average ANEES: 62.6277
--------------------------------------------------
--- Iteration [370/800] ---
    - Total Loss: 0.6034
    - NLL: 0.4451
    - Reg: 0.1583
    - p1=0.361, p2=0.380

--- Validation at iteration 370 ---
  Average MSE: 7.4180, Average ANEES: 47.6930
  >>> New best VALIDATION ANEES! Saving model. <<<
--------------------------------------------------

[DIAGNOSTIKA iter 370]
  -> Rozdíl (Pred - Target): 0.961327
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  2.54e-03

--- Validation at iteration 375 ---
  Average MSE: 9.6993, Average ANEES: 40.5449
  >>> New best VALIDATION ANEES! Saving model. <<<
--------------------------------------------------
--- Iteration [380/800] ---
    - Total Loss: 0.6051
    - NLL: 0.4481
    - Reg: 0.1570
    - p1=0.360, p2=0.379

--- Validation at iteration 380 ---
  Average MSE: 8.3917, Average ANEES: 43.6165
--------------------------------------------------

[DIAGNOSTIKA iter 380]
  -> Rozdíl (Pred - Target): 1.027276
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  2.94e-03

--- Validation at iteration 385 ---
  Average MSE: 7.5494, Average ANEES: 43.5987
--------------------------------------------------
--- Iteration [390/800] ---
    - Total Loss: 0.6351
    - NLL: 0.4797
    - Reg: 0.1554
    - p1=0.359, p2=0.377

--- Validation at iteration 390 ---
  Average MSE: 8.8469, Average ANEES: 37.9530
  >>> New best VALIDATION ANEES! Saving model. <<<
--------------------------------------------------

[DIAGNOSTIKA iter 390]
```

```
  -> Rozdíl (Pred - Target): 0.989167
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  2.87e-03

--- Validation at iteration 395 ---
  Average MSE: 9.1170, Average ANEES: 43.7458
--------------------------------------------------
--- Iteration [400/800] ---
    - Total Loss: 0.5759
    - NLL: 0.4219
    - Reg: 0.1540
    - p1=0.357, p2=0.376

--- Validation at iteration 400 ---
  Average MSE: 8.9849, Average ANEES: 47.0975
--------------------------------------------------

[DIAGNOSTIKA iter 400]
  -> Rozdíl (Pred - Target): 0.962401
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  2.03e-03

--- Validation at iteration 405 ---
  Average MSE: 7.8931, Average ANEES: 39.6240
--------------------------------------------------
--- Iteration [410/800] ---
    - Total Loss: 0.5657
    - NLL: 0.4133
    - Reg: 0.1524
    - p1=0.356, p2=0.374

--- Validation at iteration 410 ---
  Average MSE: 7.5488, Average ANEES: 42.5080
--------------------------------------------------

[DIAGNOSTIKA iter 410]
  -> Rozdíl (Pred - Target): 0.987789
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  3.18e-03

--- Validation at iteration 415 ---
  Average MSE: 9.3636, Average ANEES: 43.3083
--------------------------------------------------
--- Iteration [420/800] ---
    - Total Loss: 0.4415
    - NLL: 0.2907
    - Reg: 0.1508
    - p1=0.354, p2=0.372
```

```
--- Validation at iteration 420 ---
  Average MSE: 11.9026, Average ANEES: 45.0408
--------------------------------------------------

[DIAGNOSTIKA iter 420]
  -> Rozdíl (Pred - Target): 0.882217
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  1.80e-03

--- Validation at iteration 425 ---
  Average MSE: 11.5523, Average ANEES: 54.3717
--------------------------------------------------
--- Iteration [430/800] ---
    - Total Loss: 0.4880
    - NLL: 0.3386
    - Reg: 0.1494
    - p1=0.353, p2=0.371

--- Validation at iteration 430 ---
  Average MSE: 9.9588, Average ANEES: 46.9922
--------------------------------------------------

[DIAGNOSTIKA iter 430]
  -> Rozdíl (Pred - Target): 0.923186
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  2.43e-03

--- Validation at iteration 435 ---
  Average MSE: 10.6370, Average ANEES: 39.9162
--------------------------------------------------
--- Iteration [440/800] ---
    - Total Loss: 0.5576
    - NLL: 0.4099
    - Reg: 0.1477
    - p1=0.352, p2=0.369

--- Validation at iteration 440 ---
  Average MSE: 10.4837, Average ANEES: 42.7587
--------------------------------------------------

[DIAGNOSTIKA iter 440]
  -> Rozdíl (Pred - Target): 1.008402
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  2.00e-03

--- Validation at iteration 445 ---
  Average MSE: 9.4744, Average ANEES: 42.3872
```

```
--------------------------------------------------
--- Iteration [450/800] ---
    - Total Loss: 0.5388
    - NLL: 0.3928
    - Reg: 0.1461
    - p1=0.351, p2=0.367

--- Validation at iteration 450 ---
  Average MSE: 9.2553, Average ANEES: 43.3609
--------------------------------------------------

[DIAGNOSTIKA iter 450]
  -> Rozdíl (Pred - Target): 0.998328
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  2.63e-03

--- Validation at iteration 455 ---
  Average MSE: 8.3684, Average ANEES: 46.6202
--------------------------------------------------
--- Iteration [460/800] ---
    - Total Loss: 0.5243
    - NLL: 0.3798
    - Reg: 0.1445
    - p1=0.349, p2=0.366

--- Validation at iteration 460 ---
  Average MSE: 10.9624, Average ANEES: 46.9817
--------------------------------------------------

[DIAGNOSTIKA iter 460]
  -> Rozdíl (Pred - Target): 0.925160
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  1.84e-03

--- Validation at iteration 465 ---
  Average MSE: 10.1335, Average ANEES: 46.5213
--------------------------------------------------
--- Iteration [470/800] ---
    - Total Loss: 0.3866
    - NLL: 0.2435
    - Reg: 0.1431
    - p1=0.348, p2=0.364

--- Validation at iteration 470 ---
  Average MSE: 10.4531, Average ANEES: 41.1133
--------------------------------------------------

[DIAGNOSTIKA iter 470]
```

```
  -> Rozdíl (Pred - Target): 0.873532
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  1.69e-03

--- Validation at iteration 475 ---
  Average MSE: 10.0633, Average ANEES: 48.5902
--------------------------------------------------
--- Iteration [480/800] ---
    - Total Loss: 0.4921
    - NLL: 0.3504
    - Reg: 0.1417
    - p1=0.347, p2=0.363

--- Validation at iteration 480 ---
  Average MSE: 9.9389, Average ANEES: 38.7740
--------------------------------------------------

[DIAGNOSTIKA iter 480]
  -> Rozdíl (Pred - Target): 0.891287
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  3.14e-03

--- Validation at iteration 485 ---
  Average MSE: 8.1234, Average ANEES: 40.8472
--------------------------------------------------
--- Iteration [490/800] ---
    - Total Loss: 0.4610
    - NLL: 0.3206
    - Reg: 0.1404
    - p1=0.346, p2=0.361

--- Validation at iteration 490 ---
  Average MSE: 7.7446, Average ANEES: 37.8276
  >>> New best VALIDATION ANEES! Saving model. <<<
--------------------------------------------------

[DIAGNOSTIKA iter 490]
  -> Rozdíl (Pred - Target): 0.967566
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  1.58e-03

--- Validation at iteration 495 ---
  Average MSE: 8.7497, Average ANEES: 55.3410
--------------------------------------------------
--- Iteration [500/800] ---
    - Total Loss: 0.5088
    - NLL: 0.3697
    - Reg: 0.1391
    - p1=0.345, p2=0.360

--- Validation at iteration 500 ---
  Average MSE: 8.3515, Average ANEES: 41.4400
--------------------------------------------------

[DIAGNOSTIKA iter 500]
  -> Rozdíl (Pred - Target): 0.973981
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  1.54e-03

--- Validation at iteration 505 ---
  Average MSE: 9.3334, Average ANEES: 43.1844
--------------------------------------------------
--- Iteration [510/800] ---
    - Total Loss: 0.6025
    - NLL: 0.4646
    - Reg: 0.1379
    - p1=0.344, p2=0.359

--- Validation at iteration 510 ---
  Average MSE: 7.4476, Average ANEES: 43.4099
--------------------------------------------------

[DIAGNOSTIKA iter 510]
  -> Rozdíl (Pred - Target): 0.973107
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  1.95e-03

--- Validation at iteration 515 ---
  Average MSE: 11.0712, Average ANEES: 44.5363
--------------------------------------------------
--- Iteration [520/800] ---
    - Total Loss: 0.3815
    - NLL: 0.2448
    - Reg: 0.1367
    - p1=0.343, p2=0.358

--- Validation at iteration 520 ---
  Average MSE: 7.7661, Average ANEES: 42.0787
--------------------------------------------------

[DIAGNOSTIKA iter 520]
  -> Rozdíl (Pred - Target): 0.834144
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  3.65e-03

--- Validation at iteration 525 ---
```

```
  Average MSE: 7.3099, Average ANEES: 41.6454
--------------------------------------------------
--- Iteration [530/800] ---
    - Total Loss: 0.4219
    - NLL: 0.2864
    - Reg: 0.1355
    - p1=0.343, p2=0.357

--- Validation at iteration 530 ---
  Average MSE: 12.6842, Average ANEES: 51.3034
--------------------------------------------------

[DIAGNOSTIKA iter 530]
  -> Rozdíl (Pred - Target): 0.976890
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  2.88e-03

--- Validation at iteration 535 ---
  Average MSE: 8.6633, Average ANEES: 51.9835
--------------------------------------------------
--- Iteration [540/800] ---
    - Total Loss: 0.5178
    - NLL: 0.3833
    - Reg: 0.1345
    - p1=0.342, p2=0.356

--- Validation at iteration 540 ---
  Average MSE: 9.0310, Average ANEES: 41.8941
--------------------------------------------------

[DIAGNOSTIKA iter 540]
  -> Rozdíl (Pred - Target): 0.908892
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  2.17e-03

--- Validation at iteration 545 ---
  Average MSE: 8.9766, Average ANEES: 43.9004
--------------------------------------------------
--- Iteration [550/800] ---
    - Total Loss: 0.3974
    - NLL: 0.2641
    - Reg: 0.1333
    - p1=0.341, p2=0.355

--- Validation at iteration 550 ---
  Average MSE: 13.6225, Average ANEES: 43.8941
--------------------------------------------------
```

```
[DIAGNOSTIKA iter 550]
  -> Rozdíl (Pred - Target): 0.982014
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  1.98e-03

--- Validation at iteration 555 ---
  Average MSE: 11.3858, Average ANEES: 40.0293
--------------------------------------------------
--- Iteration [560/800] ---
    - Total Loss: 0.4958
    - NLL: 0.3636
    - Reg: 0.1321
    - p1=0.340, p2=0.354

--- Validation at iteration 560 ---
  Average MSE: 11.9617, Average ANEES: 44.5927
--------------------------------------------------

[DIAGNOSTIKA iter 560]
  -> Rozdíl (Pred - Target): 1.014207
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  2.60e-03

--- Validation at iteration 565 ---
  Average MSE: 17.9994, Average ANEES: 48.0156
--------------------------------------------------
--- Iteration [570/800] ---
    - Total Loss: 0.5269
    - NLL: 0.3958
    - Reg: 0.1311
    - p1=0.339, p2=0.352

--- Validation at iteration 570 ---
  Average MSE: 14.7626, Average ANEES: 41.5625
--------------------------------------------------

[DIAGNOSTIKA iter 570]
  -> Rozdíl (Pred - Target): 0.927627
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  1.77e-03

--- Validation at iteration 575 ---
  Average MSE: 18.4675, Average ANEES: 46.5950
--------------------------------------------------
--- Iteration [580/800] ---
    - Total Loss: 0.4327
    - NLL: 0.3027
    - Reg: 0.1301
    - p1=0.338, p2=0.351

--- Validation at iteration 580 ---
  Average MSE: 11.3487, Average ANEES: 40.4027
--------------------------------------------------

[DIAGNOSTIKA iter 580]
  -> Rozdíl (Pred - Target): 0.877814
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  2.20e-03

--- Validation at iteration 585 ---
  Average MSE: 7.9478, Average ANEES: 45.6012
--------------------------------------------------
--- Iteration [590/800] ---
    - Total Loss: 0.4316
    - NLL: 0.3026
    - Reg: 0.1290
    - p1=0.337, p2=0.350

--- Validation at iteration 590 ---
  Average MSE: 11.3101, Average ANEES: 42.0121
--------------------------------------------------

[DIAGNOSTIKA iter 590]
  -> Rozdíl (Pred - Target): 0.894985
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  2.01e-03

--- Validation at iteration 595 ---
  Average MSE: 8.4094, Average ANEES: 37.0757
  >>> New best VALIDATION ANEES! Saving model. <<<
--------------------------------------------------
--- Iteration [600/800] ---
    - Total Loss: 0.3507
    - NLL: 0.2227
    - Reg: 0.1280
    - p1=0.336, p2=0.349

--- Validation at iteration 600 ---
  Average MSE: 10.7332, Average ANEES: 47.9549
--------------------------------------------------

[DIAGNOSTIKA iter 600]
  -> Rozdíl (Pred - Target): 0.865811
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  2.04e-03
```

```
--- Validation at iteration 605 ---
  Average MSE: 7.7715, Average ANEES: 41.8769
--------------------------------------------------
--- Iteration [610/800] ---
    - Total Loss: 0.4826
    - NLL: 0.3557
    - Reg: 0.1269
    - p1=0.335, p2=0.348

--- Validation at iteration 610 ---
  Average MSE: 7.2468, Average ANEES: 40.6923
--------------------------------------------------

[DIAGNOSTIKA iter 610]
  -> Rozdíl (Pred - Target): 0.822364
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  1.97e-03

--- Validation at iteration 615 ---
  Average MSE: 8.9078, Average ANEES: 41.9769
--------------------------------------------------
--- Iteration [620/800] ---
    - Total Loss: 0.3560
    - NLL: 0.2302
    - Reg: 0.1258
    - p1=0.334, p2=0.348

--- Validation at iteration 620 ---
  Average MSE: 7.2887, Average ANEES: 32.9270
  >>> New best VALIDATION ANEES! Saving model. <<<
--------------------------------------------------

[DIAGNOSTIKA iter 620]
  -> Rozdíl (Pred - Target): 0.792655
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  1.34e-03

--- Validation at iteration 625 ---
  Average MSE: 5.3642, Average ANEES: 41.1049
--------------------------------------------------
--- Iteration [630/800] ---
    - Total Loss: 0.3940
    - NLL: 0.2692
    - Reg: 0.1248
    - p1=0.333, p2=0.346

--- Validation at iteration 630 ---
  Average MSE: 7.1715, Average ANEES: 40.1929
```

```
--------------------------------------------------

[DIAGNOSTIKA iter 630]
  -> Rozdíl (Pred - Target): 0.879569
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  1.92e-03

--- Validation at iteration 635 ---
  Average MSE: 6.2852, Average ANEES: 44.9471
--------------------------------------------------
--- Iteration [640/800] ---
    - Total Loss: 0.3573
    - NLL: 0.2336
    - Reg: 0.1237
    - p1=0.332, p2=0.345

--- Validation at iteration 640 ---
  Average MSE: 7.1677, Average ANEES: 43.6193
--------------------------------------------------

[DIAGNOSTIKA iter 640]
  -> Rozdíl (Pred - Target): 0.861807
  -> JE TO PŘESNÁ NULA? Ne
  -> Min Variance v batchi:  9.66e-04

--- Validation at iteration 645 ---
  Average MSE: 9.9093, Average ANEES: 45.7307
--------------------------------------------------
--- Iteration [650/800] ---
    - Total Loss: 0.3959
    - NLL: 0.2732
    - Reg: 0.1227
    - p1=0.331, p2=0.344

--- Validation at iteration 650 ---
```

```
---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
Cell In[7], line 40
     37 validation_period = 5  # Chceme validovat jednou za epochu (na konci)
     39 # 3. Volání funkce
---> 40 trained_knet = trainer.train_BayesianKalmanNet(
     41     model=state_knet,
     42     train_loader=train_loader,
     43     val_loader=val_loader,
     44     device=DEVICE,
     45 
     46     # Parametry iterací
     47     total_train_iter=total_iterations,
     48     learning_rate=5e-4,        # Opatrný LR pro BKN
     49     clip_grad=1.0,             # Stabilizace gradientů
     50     J_samples=10,               # Počet MC vzorků (paralelně)
     51
     52     # Logování a validace
     53     validation_period=validation_period,
     54     logging_period=10,         # Výpis loss každých 10 batchů
     55
     56     # Warmup
     57     warmup_iterations=warmup_iterations,
     58     weight_decay_=1e-5
     59 )

File ~/skola/KalmanNet-main/utils/trainer.py:1849, in␣
 ↪train_BayesianKalmanNet(model, train_loader, val_loader, device,␣
 ↪total_train_iter, learning_rate, clip_grad, J_samples, validation_period,␣
 ↪logging_period, warmup_iterations, weight_decay_)
   1847     y_t_val = y_meas_val_batch[:, t, :]
   1848     u_t_val = u_input_val_batch[:,t,:]
-> 1849     x_filtered_t, _ = model.step(y_t_val,u_t_val)
   1850     val_current_x_hats.append(x_filtered_t)
   1851 val_ensemble_trajectories.append(torch.stack(val_current_x_hats, dim=1)

File ~/skola/KalmanNet-main/state_NN_models/NCLT/StateBayesianKalmanNet.py:170,
 ↪in StateBayesianKalmanNetNCLT.step(self, y_t, u_t)
    166 # 4. NETWORK FORWARD
    167 # Kontrola hidden state před vstupem do GRU
    168 self._check_tensor(self.h_prev, "Hidden State Prev")
--> 170 K_vec, h_new, regs =␣
 ↪self.dnn(norm_state_inno, norm_innovation, norm_diff_state, norm_diff_obs, self.h_prev)
    172 # Kontrola výstupu sítě
    173 self._check_tensor(K_vec, "DNN Output (K_vec)")

File ~/.local/lib/python3.10/site-packages/torch/nn/modules/module.py:1532, in␣
 ↪Module._wrapped_call_impl(self, *args, **kwargs)
   1530     return self._compiled_call_impl(*args, **kwargs)  # type:␣
 ↪ignore[misc]
   1531 else:
-> 1532     return self._call_impl(*args, **kwargs)

File ~/.local/lib/python3.10/site-packages/torch/nn/modules/module.py:1541, in␣
 ↪Module._call_impl(self, *args, **kwargs)
   1536 # If we don't have any hooks, we want to skip the rest of the logic in
   1537 # this function, and just call forward.
   1538 if not (self._backward_hooks or self._backward_pre_hooks or self.
 ↪_forward_hooks or self._forward_pre_hooks
```

```
   1539         or _global_backward_pre_hooks or _global_backward_hooks
   1540         or _global_forward_hooks or _global_forward_pre_hooks):
-> 1541     return forward_call(*args, **kwargs)
   1543 try:
   1544     result = None

File ~/skola/KalmanNet-main/state_NN_models/NCLT/DNN_BayesianKalmanNet.py:49, i␣
 ↪DNN_BayesianKalmanNetNCLT.forward(self, state_inno, inovation, diff_state,␣
 ↪diff_obs, h_prev)
     43 # Aplikace normalizace
     44 # normalized_input = self.input_norm(nn_input)
     45
     46 # Apply dropout on normalized data
     47 activated_input, reg1 = self.concrete_dropout1(nn_input, self.
 ↪input_layer)
---> 49 out_gru, h_new = self.gru(activated_input.unsqueeze(0), h_prev)
     50 out_gru_squeezed = out_gru.squeeze(0)
     52 out_final, reg2 = self.concrete_dropout2(out_gru_squeezed, self.
 ↪output_layer)

File ~/.local/lib/python3.10/site-packages/torch/nn/modules/module.py:1532, in␣
 ↪Module._wrapped_call_impl(self, *args, **kwargs)
   1530     return self._compiled_call_impl(*args, **kwargs)  # type:␣
 ↪ignore[misc]
   1531 else:
-> 1532     return self._call_impl(*args, **kwargs)

File ~/.local/lib/python3.10/site-packages/torch/nn/modules/module.py:1541, in␣
 ↪Module._call_impl(self, *args, **kwargs)
   1536 # If we don't have any hooks, we want to skip the rest of the logic in
   1537 # this function, and just call forward.
   1538 if not (self._backward_hooks or self._backward_pre_hooks or self.
 ↪_forward_hooks or self._forward_pre_hooks
   1539         or _global_backward_pre_hooks or _global_backward_hooks
   1540         or _global_forward_hooks or _global_forward_pre_hooks):
-> 1541     return forward_call(*args, **kwargs)
   1543 try:
   1544     result = None

File ~/.local/lib/python3.10/site-packages/torch/nn/modules/rnn.py:1133, in GRU
 ↪forward(self, input, hx)
   1131 self.check_forward_args(input, hx, batch_sizes)
   1132 if batch_sizes is None:
-> 1133     result =␣
 ↪_VF.gru(input, hx, self._flat_weights, self.bias, self.num_layers,
   1134␣
 ↪                     self.dropout, self.training, self.bidirectional, self.ba ch_first)
   1135 else:
```

```
   1136     result = _VF.gru(input, batch_sizes, hx, self._flat_weights, self.
 ↪bias,
   1137                          self.num_layers, self.dropout, self.training, self
 ↪bidirectional)

KeyboardInterrupt:
```

[ ]:
```
# import gc
# import torch

# # === GPU/CUDA Cleanup Helper (Models Only) ===
# # Frees GPU memory by moving neural nets to CPU, deleting their references,
# # forcing GC, and clearing CUDA cache. Does NOT remove DataLoaders or trainer.

# def _cuda_mem_info():
#     if torch.cuda.is_available():
#         alloc = torch.cuda.memory_allocated() / 1e6
#         reserv = torch.cuda.memory_reserved() / 1e6
#         print(f"CUDA mem -> Allocated: {alloc:.1f} MB | Reserved: {reserv:.
 ↪1f} MB")
#     else:
#         print("CUDA not available.")

# print("Before cleanup:")
# _cuda_mem_info()

# # Release neural network models only
# for name in ["state_knet", "trained_model_bkn"]:
#     if name in globals():
#         obj = globals()[name]
#         try:
#             obj.to("cpu")
#         except Exception:
#             pass
#         # Drop common optimizer/scheduler refs if attached to the model
#         for attr in ["optimizer", "opt", "optim", "scheduler"]:
#             if hasattr(obj, attr):
#                 try:
#                     delattr(obj, attr)
#                 except Exception:
#                     pass
#         del globals()[name]
#         print(f"Released model '{name}' and deleted reference")

# # Force Python GC and clear CUDA cache
# gc.collect()
# if torch.cuda.is_available():
```

```
#     try:
#         torch.cuda.synchronize()
#     except Exception:
#         pass
#     torch.cuda.empty_cache()
#     try:
#         torch.cuda.ipc_collect()
#     except Exception:
#         pass

# print("After cleanup:")
# _cuda_mem_info()
```

[9]:
```
if True:
    import torch
    import os

    # Protože jsi trénink přerušil, nemáme slovník s metrikami automaticky.
    # Musíme uložit přímo model 'state_knet'.

    # 1. Definuj cestu (Metriky si do názvu musíš dopsat ručně podle toho, co␣
 ↪jsi viděl v logu naposledy,
    # nebo použij obecný název, abys o model nepřišel).
    save_path = 'best_BKN_test_results.pth'
    # Pokud si pamatuješ hodnoty z logu (např. ANEES 23.5), můžeš je tam dopsat␣
 ↪ručně:
    # save_path = 'best_kalmannet_nclt_sensor_fusion_ANEES23.57_MSE8.
 ↪17_interrupted.pth'

    # 2. Uložení
    # DŮLEŽITÉ: Voláme .state_dict() přímo na objektu state_knet (nebo␣
 ↪trained_knet, pokud jsi ho přiřadil)
    # Nepoužíváme závorky ['final_model'], protože state_knet UŽ JE ten model.
    torch.save(state_knet.state_dict(), save_path)

    print(f"\n  Záchrana modelu úspěšná! Uloženo do: {save_path}")

if False:
    import os
    trained_knet = state_knet
    # Přístup k metrikám ve slovníku je správně (přes závorky [])
    print(f"   -> Best ANEES: {trained_knet['best_val_anees']:.4f}")
    print(f"   -> Best MSE:   {trained_knet['best_val_mse']:.4f}")
    save_path =␣
 ↪f'best_kalmannet_nclt_sensor_fusion_ANEES{trained_knet["best_val_anees"]:.
 ↪4f}_MSE{trained_knet["best_val_mse"]:.4f}_nejlepsi_test_vysledky.pth '
```

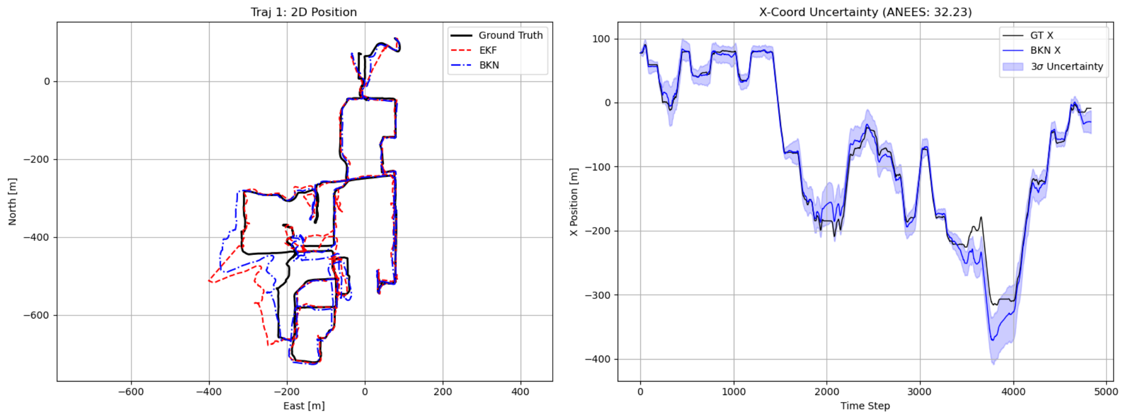
```
    # OPRAVA: Musíš vytáhnout model ze slovníku pomocí klíče 'final_model'
    # A teprve na něm zavolat .state_dict()
    torch.save(trained_knet['final_model'].state_dict(), save_path)

    print(f"\n Model úspěšně uložen do: {save_path}")
```

```
Záchrana modelu úspěšná! Uloženo do: best_BKN_test_results.pth
```

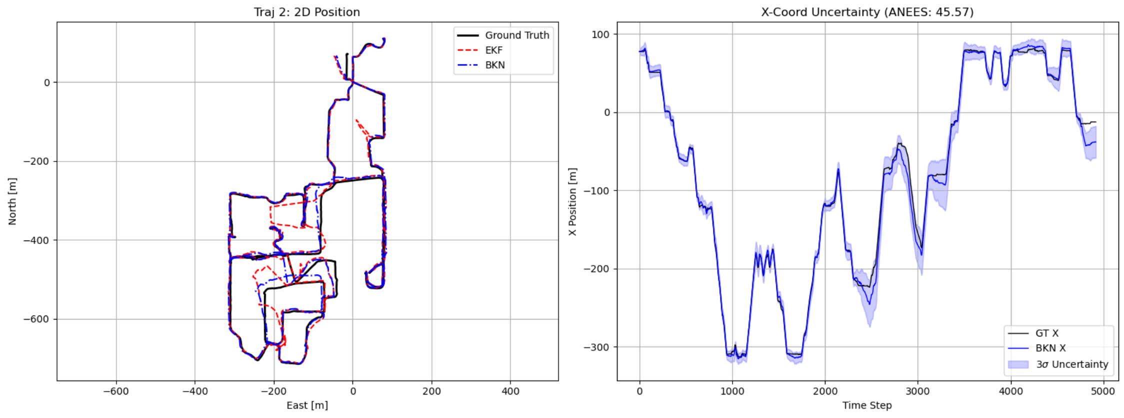
[8]:
```
import torch
import torch.nn.functional as F
import numpy as np
import matplotlib.pyplot as plt
import Filters

# ==============================================================================
# 0. KONFIGURACE A MODEL
# ==============================================================================
DT_SEC = 1.0
J_SAMPLES = 50  # 50 vzorků pro kvalitní odhad distribuce

if hasattr(sys_model, 'dt'):
    sys_model.dt = DT_SEC
    print(f"INFO: Nastaveno sys_model.dt = {DT_SEC} s")

# Načtení Bayesian modelu
try:
    trained_model_bkn = state_knet
    trained_model_bkn.eval()
    print(f"INFO: Používám Bayesian KalmanNet s J={J_SAMPLES} vzorky.")
except NameError:
    raise NameError("Chyba: 'state_knet' (BKN) neexistuje.")

from Filters import NCLT
# Inicializace klasických filtrů
ukf_filter = NCLT.UnscentedKalmanFilterNCLT(sys_model)
ekf_filter = NCLT.ExtendedKalmanFilterNCLT(sys_model)

# --- ROBUSTNÍ FUNKCE PRO ANEES ---
def calculate_anees(x_true, x_est, P_est):
    """
    Vypočítá ANEES. Očekává vstupy už se shodnou dimenzí.
    Používá pinv pro stabilitu.
    """
    T = x_true.shape[0]
    anees_list = []
    error = x_true - x_est
    for t in range(T):
        e_t = error[t].unsqueeze(1) # [n, 1]
        P_t = P_est[t]              # [n, n]

        try:
            P_inv = torch.linalg.pinv(P_t, hermitian=True)
        except RuntimeError:
            P_inv = torch.eye(P_t.shape[0], device=P_t.device)

        nees = torch.mm(torch.mm(e_t.t(), P_inv), e_t).item()
        anees_list.append(nees)

    return np.mean(anees_list)

# ==============================================================================
# 1. EVALUACE A VIZUALIZACE
# ==============================================================================
results = {
    'GPS_Sensor': [],   'GPS_Authors': [],
    'EKF_MSE': [],      'EKF_ANEES': [],
    'UKF_MSE': [],      'UKF_ANEES': [],
    'BKN_MSE': [],      'BKN_ANEES': []
}

print(f"\nSpouštím detailní evaluaci na {len(test_data_raw)} trajektoriích...")

# HLAVIČKA DETAILNÍ TABULKY
print("\n" + "-"*95)
print(f"{'TRAJEKTORIE':<12} | {'BKN MSE':<10} | {'BKN ANEES':<10} | {'EKF MSE':
 ↪<10} | {'EKF ANEES':<10} | {'GPS MSE':<10}")
print("-" * 95)

for i, traj in enumerate(test_data_raw):
    # --- PŘÍPRAVA DAT ---
    gt_raw = traj['ground_truth'].float().to(DEVICE)
    gps_filtered = traj['filtered_gps'].float().to(DEVICE)
    gps_filled = traj['gps'].float().to(DEVICE)
    imu_raw = traj['imu'].float().to(DEVICE)
    odo_raw = traj['filtered_wheel'].float().to(DEVICE)

    T_len = gt_raw.shape[0]

    # Input vector
    u_full = torch.stack((
        torch.nan_to_num(odo_raw[:, 0], nan=0.0),
        torch.nan_to_num(odo_raw[:, 1], nan=0.0),
        imu_raw[:, 2],
        imu_raw[:, 3]
    ), dim=1).to(DEVICE)

    # Init State (bereme z GT)
    # x_true má dimenzi 3 (X, Y, Theta)
    x_true = gt_raw[:, :3]

    # Init pro filtry (plná dimenze modelu)
    m = sys_model.state_dim
    x0_vec = torch.zeros(m).to(DEVICE)
    x0_vec[0] = x_true[0, 0]; x0_vec[1] = x_true[0, 1]
    if m >= 3 and x_true.shape[1] >= 3: x0_vec[4] = x_true[0, 2]

    # --- A. BĚH BAYESIAN KALMANNET ---
    trained_model_bkn.train() # Nutné pro MC Dropout

    batch_x0 = x0_vec.unsqueeze(0).repeat(J_SAMPLES, 1)
    trained_model_bkn.reset(batch_size=J_SAMPLES, initial_state=batch_x0)

    bkn_samples = []
    bkn_samples.append(batch_x0.unsqueeze(1))

    with torch.no_grad():
        for t in range(1, T_len):
            y_t = gps_filtered[t].unsqueeze(0).repeat(J_SAMPLES, 1)
            u_t = u_full[t].unsqueeze(0).repeat(J_SAMPLES, 1)
            x_est_j, _ = trained_model_bkn.step(y_t, u_t)
            bkn_samples.append(x_est_j.unsqueeze(1))

    trained_model_bkn.eval()
    bkn_ensemble = torch.cat(bkn_samples, dim=1) # [J, T, m]

    # Statistiky BKN
    x_bkn_mean = torch.mean(bkn_ensemble, dim=0)

    # Výpočet kovariance BKN
    centered = bkn_ensemble - x_bkn_mean.unsqueeze(0)
    centered_perm = centered.permute(1, 2, 0)
    P_bkn = torch.bmm(centered_perm, centered_perm.transpose(1, 2)) /␣
↪(J_SAMPLES - 1)
    P_bkn = P_bkn + torch.eye(m, device=DEVICE).unsqueeze(0) * 1e-6

    # --- B. EKF & UKF ---
    def run_filter(flt):
        try:
            res = flt.process_sequence(gps_filtered, u_seq=u_full, Ex0=x0_vec,␣
↪P0=sys_model.P0)
            return res['x_filtered'], res.get('P_filtered', torch.eye(m,␣
↪device=DEVICE).repeat(T_len, 1, 1))
        except:
            return torch.zeros(T_len, m).to(DEVICE), torch.eye(m,␣
↪device=DEVICE).repeat(T_len, 1, 1)

    x_ekf, P_ekf = run_filter(ekf_filter)
    x_ukf, P_ukf = run_filter(ukf_filter)

    # --- VYHODNOCENÍ (SLICING) ---
    eval_dim = x_true.shape[1] # 3 (X, Y, Theta)

    def evaluate_metrics(est_x, est_P):
        # MSE (poloha X, Y)
        mse = F.mse_loss(est_x[:, :2], x_true[:, :2]).item()
        # ANEES (Oříznuté na eval_dim)
        est_x_sliced = est_x[:, :eval_dim]
        est_P_sliced = est_P[:, :eval_dim, :eval_dim]
        anees = calculate_anees(x_true, est_x_sliced, est_P_sliced)
        return mse, anees

    mse_ekf, anees_ekf = evaluate_metrics(x_ekf, P_ekf)
    mse_ukf, anees_ukf = evaluate_metrics(x_ukf, P_ukf)
    mse_bkn, anees_bkn = evaluate_metrics(x_bkn_mean, P_bkn)

    valid_mask = ~torch.isnan(gps_filtered[:, 0])
    mse_gps = F.mse_loss(gps_filtered[valid_mask], x_true[valid_mask, :2]).
↪item() if valid_mask.sum() > 0 else np.nan

    # Ukládání
    results['GPS_Sensor'].append(mse_gps)
    results['EKF_MSE'].append(mse_ekf); results['EKF_ANEES'].append(anees_ekf)
    results['UKF_MSE'].append(mse_ukf); results['UKF_ANEES'].append(anees_ukf)
    results['BKN_MSE'].append(mse_bkn); results['BKN_ANEES'].append(anees_bkn)

    # --- VÝPIS ŘÁDKU TABULKY ---
    print(f"{i+1:<12} | {mse_bkn:<10.2f} | {anees_bkn:<10.2f} | {mse_ekf:<10.
↪2f} | {anees_ekf:<10.2f} | {mse_gps:<10.2f}")

    # --- VIZUALIZACE ---
    gt_np = x_true.cpu().numpy()
    bkn_np = x_bkn_mean.cpu().numpy()
    std_x_bkn = torch.sqrt(P_bkn[:, 0, 0]).cpu().numpy()
    t_axis = np.arange(T_len)
    fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(16, 6))

    # Graf 1: 2D Trajektorie
    ax1.plot(gt_np[:, 0], gt_np[:, 1], 'k-', lw=2, label='Ground Truth')
    ax1.plot(x_ekf.cpu().numpy()[:, 0], x_ekf.cpu().numpy()[:, 1], 'r--', lw=1.
  ↪5, label='EKF')
    ax1.plot(bkn_np[:, 0], bkn_np[:, 1], 'b-.', lw=1.5, label='BKN')
    ax1.set_title(f'Traj {i+1}: 2D Position')
    ax1.set_xlabel('East [m]'); ax1.set_ylabel('North [m]')
    ax1.legend(); ax1.grid(True); ax1.axis('equal')

    # Graf 2: Time Series X s Confidence Tube
    ax2.plot(t_axis, gt_np[:, 0], 'k-', lw=1, label='GT X')
    ax2.plot(t_axis, bkn_np[:, 0], 'b-', lw=1, label='BKN X')
    ax2.fill_between(t_axis,
                     bkn_np[:, 0] - 3*std_x_bkn,
                     bkn_np[:, 0] + 3*std_x_bkn,
                     color='blue', alpha=0.2, label='3$\sigma$ Uncertainty')
    ax2.set_title(f'X-Coord Uncertainty (ANEES: {anees_bkn:.2f})')
    ax2.set_xlabel('Time Step'); ax2.set_ylabel('X Position [m]')
    ax2.legend(); ax2.grid(True)

    plt.tight_layout()
    plt.show()

# ==============================================================================
# 2. FINÁLNÍ SOUHRN
# ==============================================================================
print("\n" + "="*80)
print(f"PRŮMĚRNÉ VÝSLEDKY ({len(test_data_raw)} trajektorií)")
print("="*80)
print(f"{'Metoda':<10} | {'MSE':<12} | {'RMSE':<12} | {'ANEES':<12}")
print("-" * 80)
print(f"{'BKN':<10} | {np.nanmean(results['BKN_MSE']):<12.2f} | {np.sqrt(np.
  ↪nanmean(results['BKN_MSE'])):<12.2f} | {np.nanmean(results['BKN_ANEES']):<12.
  ↪2f}")
print(f"{'EKF':<10} | {np.nanmean(results['EKF_MSE']):<12.2f} | {np.sqrt(np.
  ↪nanmean(results['EKF_MSE'])):<12.2f} | {np.nanmean(results['EKF_ANEES']):<12.
  ↪2f}")
print("="*80)
```

```
INFO: Nastaveno sys_model.dt = 1.0 s
INFO: Používám Bayesian KalmanNet s J=50 vzorky.

Spouštím detailní evaluaci na 3 trajektoriích…

--------------------------------------------------------------------------------
```

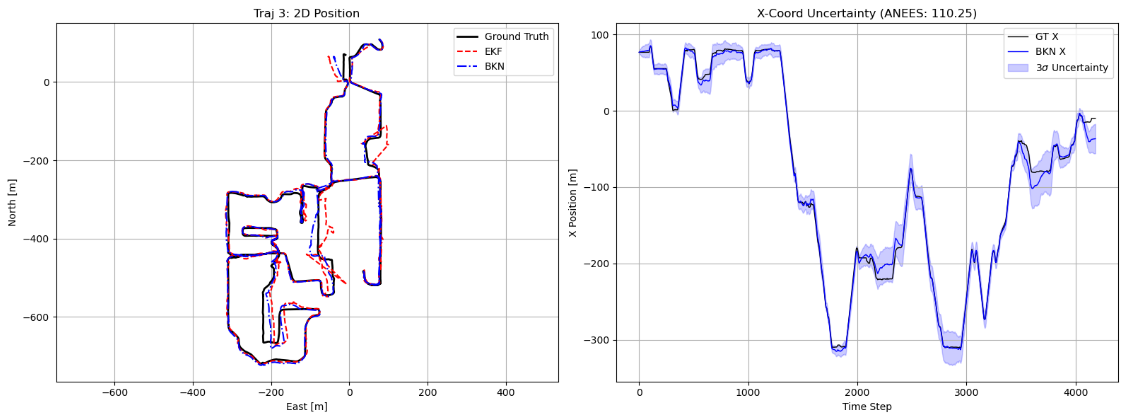
```
---------------
TRAJEKTORIE   | BKN MSE     | BKN ANEES   | EKF MSE     | EKF ANEES   | GPS MSE
--------------------------------------------------------------------------------
---------------
1             | 185.61      | 32.23       | 443.11      | 1027.02     | 164.46
```



```
2             | 44.76       | 45.57       | 221.68      | 445.10      | 51.13
```



```
3             | 34.27       | 110.25      | 128.44      | 311.25      | 39.94
```



```
================================================================================
PRŮMĚRNÉ VÝSLEDKY (3 trajektorií)
================================================================================
Metoda      | MSE            | RMSE           | ANEES
--------------------------------------------------------------------------------
BKN         | 88.21          | 9.39           | 62.68
EKF         | 264.41         | 16.26          | 594.46
================================================================================
```