Pengolahan Sinyal Digital

ANALISIS PERBANDINGAN TIME LAG:

FILTER IIR BUTTERWORTH ZERO-PHASE, MOVING AVERAGE, DAN EXPONENTIAL MOVING AVERAGE PADA DATA SINTETIS

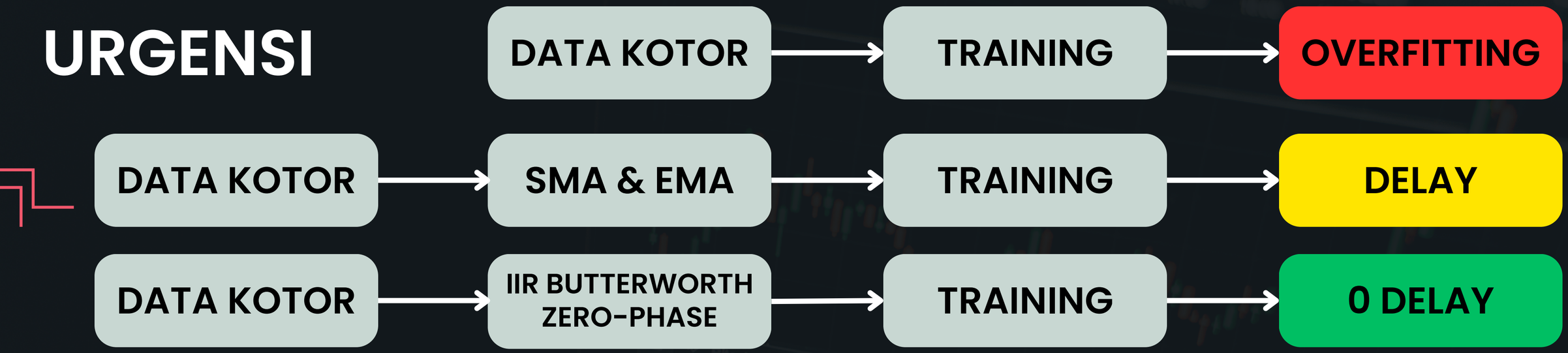
Presented by

Rafael Hartono [24031554166]
Meisya Natasafira [24031554174]

PENDAHULUAN

PENDAHULUAN

URGENSI



MANFAAT

- Menciptakan Ground Truth
- Menghilangkan Noise

PENDAHULUAN

electronics

MDPI

Article

Evaluation of Different Filtering Methods Devoted to Magnetometer Data Denoising

Tiago Pereira [1,*], Victor Santos [1], Tiago Gameiro [1], Carlos Viegas [2] and Nuno Ferreira [1,3]

[1] Polytechnic Institute of Coimbra, Coimbra Institute of Engineering, Rua Pedro Nunes-Quinta da Nora, 3030-199 Coimbra, Portugal; vsantos@isec.pt (V.S.); a2019124292@isec.pt (T.G.); nunomig@isec.pt (N.F.)
[2] ADAI (Associação para o Desenvolvimento da Aerodinâmica Industrial), Department of Mechanical Engineering, University Coimbra, Rua Luís Reis Santos, Pólo II, 3030-788 Coimbra, Portugal; carlos.viegas@uc.pt
[3] GECAD—Knowledge Research Group on Intelligent Engineering and Computing for Advanced Innovation and Development of the Engineering Institute of Porto (ISEP), Polytechnic Institute of Porto (IPP), 4200-465 Porto, Portugal
* Correspondence: a21280799@isec.pt

Abstract: In this article, we describe a performance comparison conducted between several digital filters intended to mitigate the intrinsic noise observed in magnetometers. The considered filters were used to smooth the control signals derived from the magnetometers, which were present in an autonomous forestry machine. Three moving average FIR filters, based on rectangular Bartlett and Hanning windows, and an exponential moving average IIR filter were selected and analyzed. The trade-off between the noise reduction factor and the latency of the proposed filters was also investigated, taking into account the crucial importance of latency on real-time applications and control algorithms. Thus, a maximum latency value was used in the filter design procedure instead of the usual filter order. The experimental results and simulations show that the linear decay moving average (LDMA) and the raised cosine moving average (RCMA) filters outperformed the simple moving average (SMA) and the exponential moving average (EMA) in terms of noise reduction, for a fixed latency value, allowing a more accurate heading angle calculation and position control mechanism for autonomous and unmanned ground vehicles (UGVs).

Keywords: signal denoise; magnetometer noise; moving average filters; forestry machine

Citation: Pereira, T.; Santos, V.; Gameiro, T.; Viegas, C.; Ferreira, N. Evaluation of Different Filtering Methods Devoted to Magnetometer Data Denoising. *Electronics* **2024**, *13*, 2006. https://doi.org/10.3390/

1. Introduction

- **Validasi Filter dengan Sinusoidal**
- **Evaluasi Delay dan SNR [1]**

A Dual Network Solution (DNS) for Lag-Free Time Series Forecasting

Subhrajit Samanta
ERI@N-IGS
NTU
Singapore
sama0021@e.ntu.edu.sg

Mahardhika Pratama
SCSE
NTU
Singapore
MPRATAMA@ntu.edu.sg

Suresh Sundaram
Aerospace Engineering
IISC
Bengaluru, India
sureshsundaram@iisc.ac.in

Narasimalu Srikanth
ERI@N
NTU
Singapore
nsrikanth@ntu.edu.sg

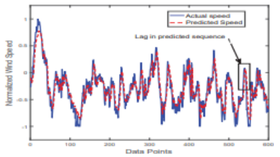
***Abstract*—When it comes to time series forecasting, lag in the predicted sequence can be a predominant issue. Unfortunately, this is often overlooked in most of the time series literature as this does not contribute to a high prediction error (i.e. MSE). However, it leads to a rather poor forecast in terms of movement prediction in time series. In this article, we tackle this basic problem with a novel trend driven mechanism. Trend, defined as the inherent pattern of the data, is extracted here and utilized next to perform a lag-free forecasting. We propose a generic and light Dual Network Solution (DNS), where the first network predicts the trend and the second network utilizes that predicted trend along with its historical information to capture the dynamical behavior of the time series efficiently. DNS exhibits a substantially improved ($\approx 10\%$ better) performance compared to more complex and resource-intensive state-of-the-art algorithms in large scale regression problems. Apart from the traditional Mean Squared Error (MSE), we also propose a new Movement Prediction Metric or MPM (for detection of lag in time series) as a new complementary performance metric to evaluate the efficacy of DNS better.**

***Index Terms*—Lag in time series prediction, Dual network solution, Movement prediction, Lag free time series prediction**

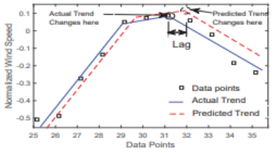
I. Introduction

Time series modelling and forecasting are one of the most popular areas of research in the machine learning and data science community for its wide applicability in various domains across different sectors such as energy demand prediction, weather forecasting, financial prediction, etc. This ever-growing field of study has seen significant improvement over the past decades. Starting from the statistical methods of regression analysis to recent advents in machine learning algorithms have propelled this research to a new high. However, across most of this literature, a basic problem of lag difference (in the predicted sequence) persists. Next, we discuss the problem

prediction task is performed here with a Multi-Layer Perceptron network or MLP (hidden node with sigmoid activations in the single hidden layer) with only past values of the target sequence as inputs and the actual vs prediction plot is provided in figure 1. From time instance 25 to 31 in figure 1b the trend is upward which leads the MLP to predict an upward trend at point 31 (marked in the figure 1b) in spite of an actual downward trend. This results in a predicted sequence with a distinct lag as seen in the figure 1a and 1b.



(a) Actual vs prediction plot

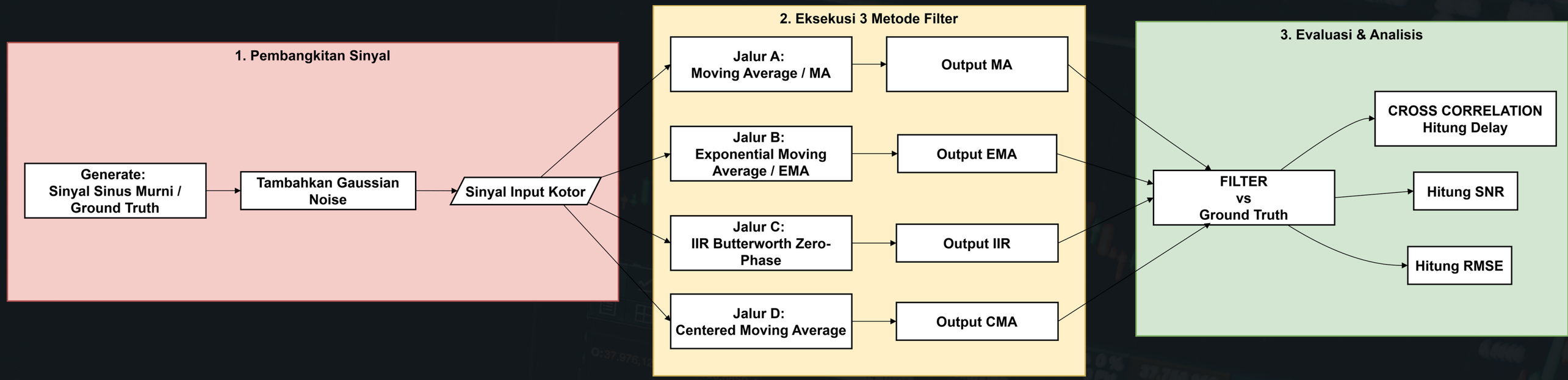


(b) Zoomed in to show Lag

Fig. 1: Demonstration of Lag in Time Series Forecasting

- **Urgensi 0 Delay [3],[5]**

METODE



FILTFILT [2]

CROSS EVALUATION [1]

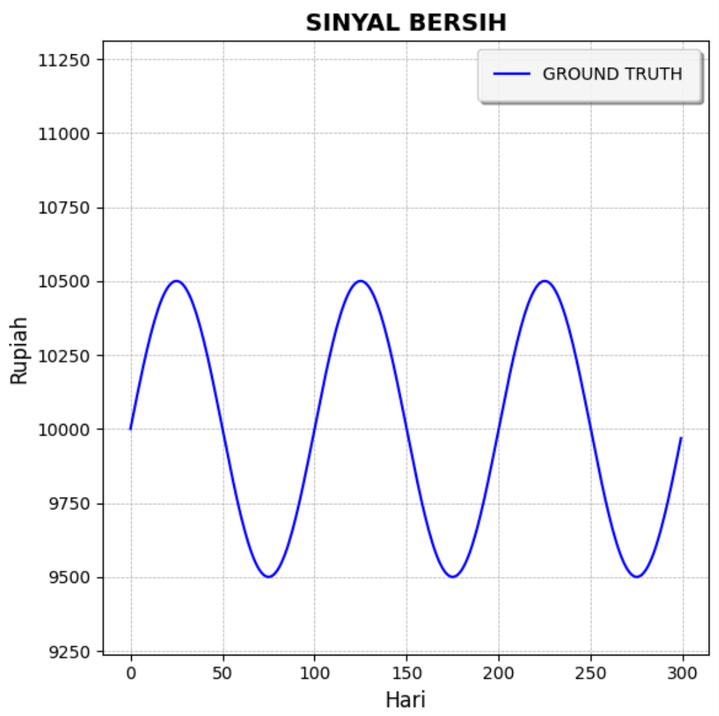
PERHITUNGAN DELAY [1]

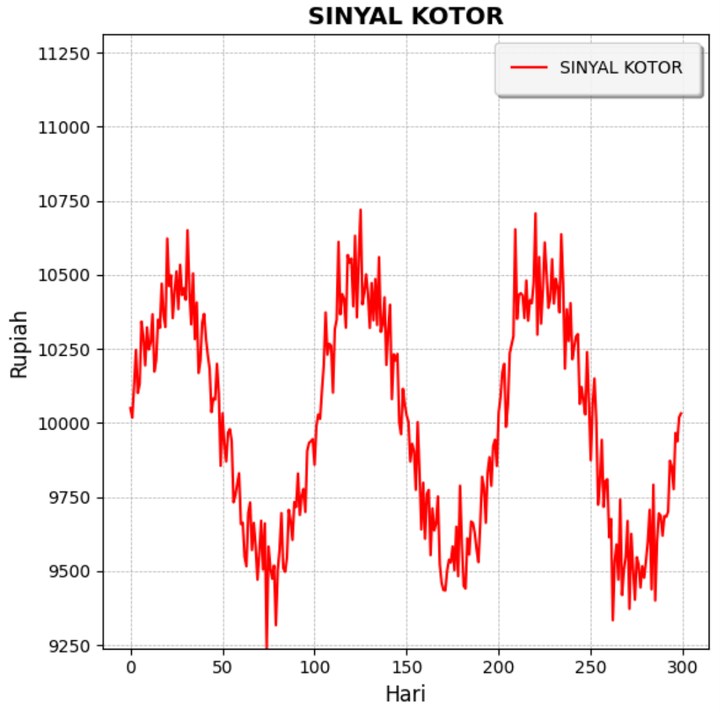
RMSE [6]

METODE

MEMBUAT DATA SINUS BUATAN

```
fs = 1.0
durasi = 300
t = np.linspace(0, durasi, int(fs * durasi), endpoint=False)

harga_base = 10000

freq_trend = 0.01
sinyal_bersih = 500 * np.sin(2 * np.pi * freq_trend * t)

np.random.seed(42)
noise = np.random.normal(0, 100, t.shape)


harga_pasar = harga_base + sinyal_bersih + noise
total_data = len(harga_pasar)


filter_hari = 20

ground_truth = harga_base + sinyal_bersih
```





METODE

SMA

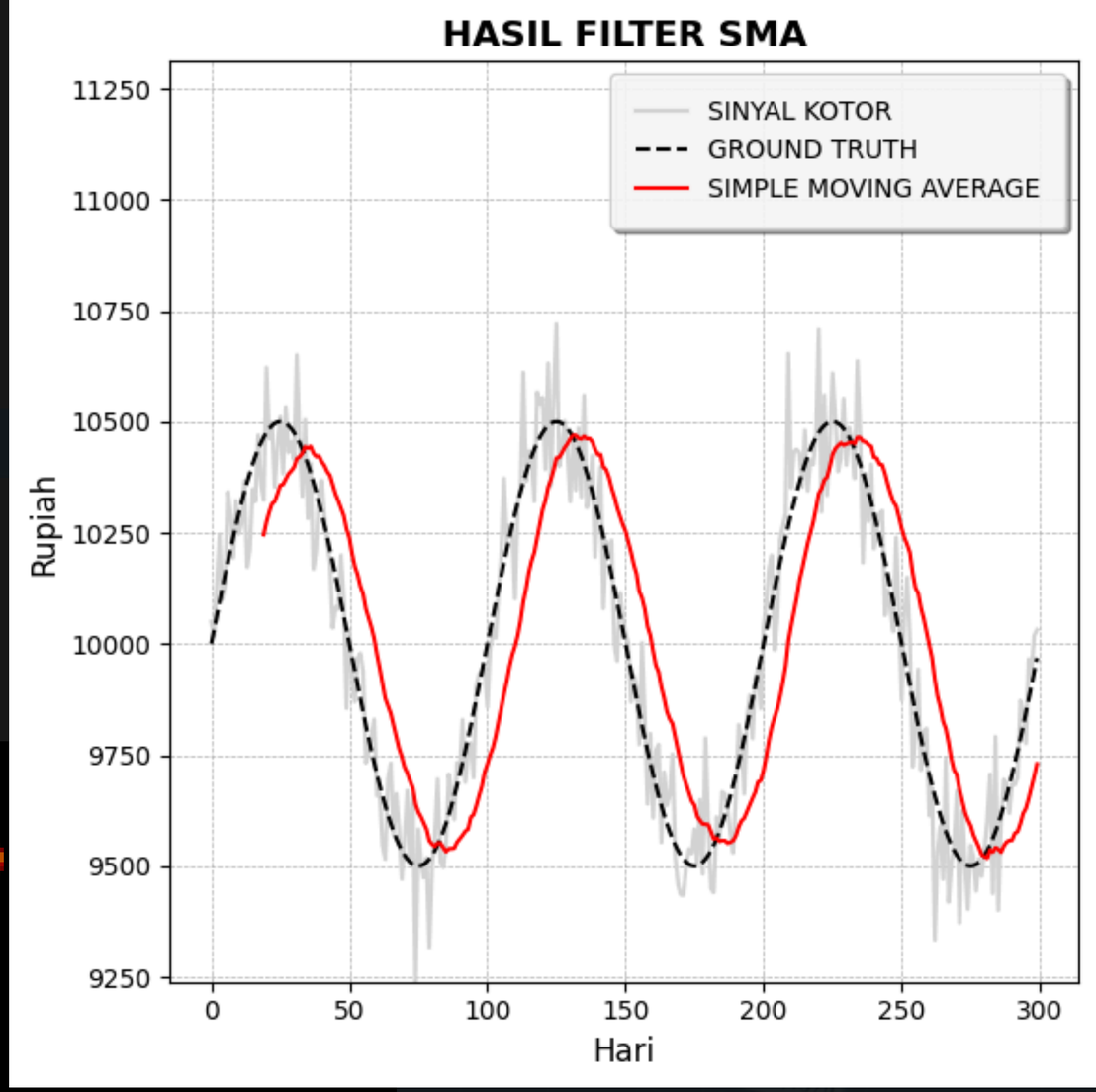
$$y[i] = \frac{1}{N} \sum_{j=0}^{N-1} x[i-j]$$

[4]

$$y[9] = \frac{x[9] + x[8] + x[7] + x[6] + x[5]}{5}$$

[4]

```
print('======FILTER SMA======')
hasil_sma = np.full_like(harga_pasar, np.nan)
window_sma = filter_hari
print('Window SMA:', window_sma)


for i in range(window_sma - 1, total_data):
    data_window = harga_pasar[i - window_sma + 1 : i + 1]
    hasil_sma[i] = np.mean(data_window)
```



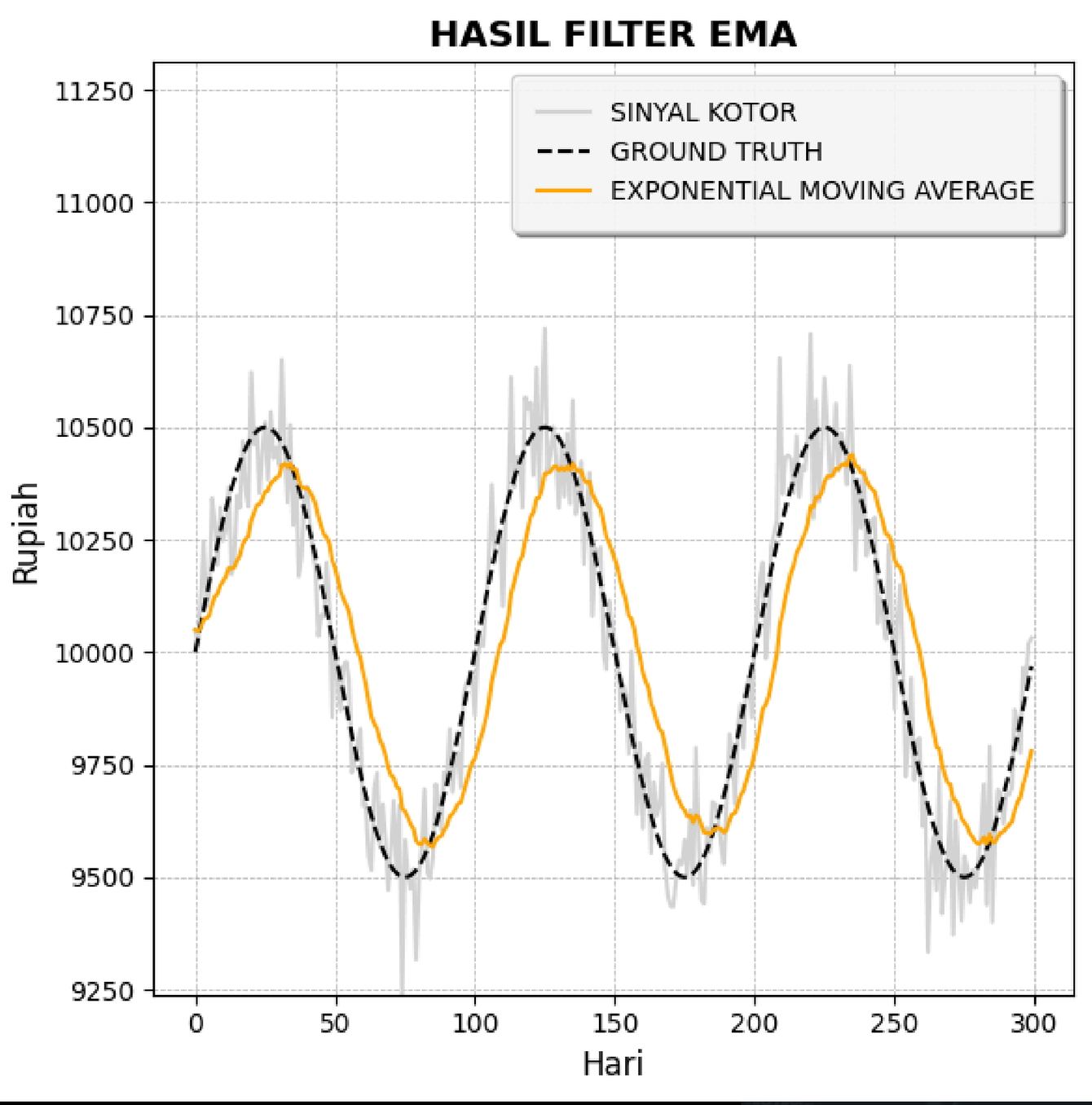
METODE

EMA

$$y[i] = \alpha \cdot x[i] + (1-\alpha) \cdot y[i-1]$$
$$y[i] = \alpha \cdot x[i] + (1-\alpha) \cdot (\alpha \cdot x[i-1] + (1-\alpha) \cdot y[i-2])$$
$$y[i] = \alpha \cdot x[i] + (1-\alpha) \cdot (\alpha \cdot x[i-1] + (1-\alpha) \cdot (\alpha \cdot x[i-2] + (1-\alpha) \cdot y[i-3]))$$
$$\ldots$$
$$y[i] = \alpha \sum_{k=0}^{n} (1-\alpha)^k x[n-k]$$

[4]

```
print('======FILTER EMA======')
hasil_ema = np.zeros_like(harga_pasar)
window_ema = filter_hari
alpha = 2 / (window_ema + 1)

print('Window EMA:', window_ema)
print('Alpha EMA:', alpha)


hasil_ema[0] = harga_pasar[0]

for i in range(1, total_data):
    hasil_ema[i] = (alpha * harga_pasar[i]) + ((1 - alpha) * hasil_ema[i-1])
```

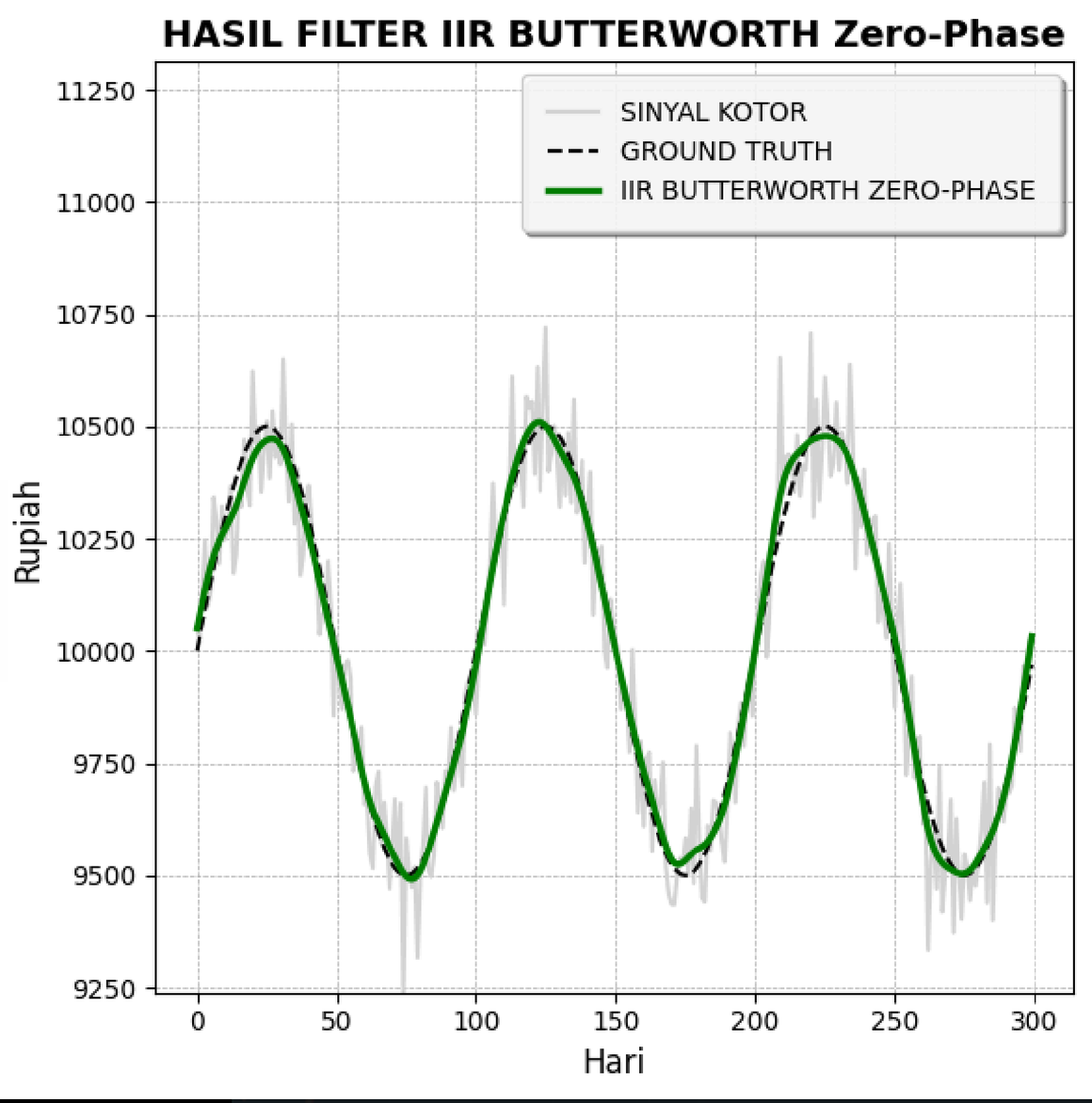


METODE

IIR BUTTERWORTH ZERO-PHASE

$$|H(q)|^2 = \frac{B(q)}{A(q)}\frac{B(q^{-1})}{A(q^{-1})} = \frac{C(q)}{A(q)} + \frac{D(q^{-1})}{A(q^{-1})}, \quad [7]$$

```
105 print('======FILTER IIR BUTTERWORTH Zero-Phase======')
106 print('Window IIR:', filter_hari)
107 cutoff_Hz = 1.0 / filter_hari
108 print('Cutoff Frequency:', cutoff_Hz)
109 filter_order = 2
110 print('Filter Order:', filter_order)
111
112
113 b, a = butter(N=filter_order, Wn=cutoff_Hz, btype='lowpass', fs=fs)
114
115
116 hasil_iir_zp = filtfilt(b, a, harga_pasar, padlen=20)
117
```

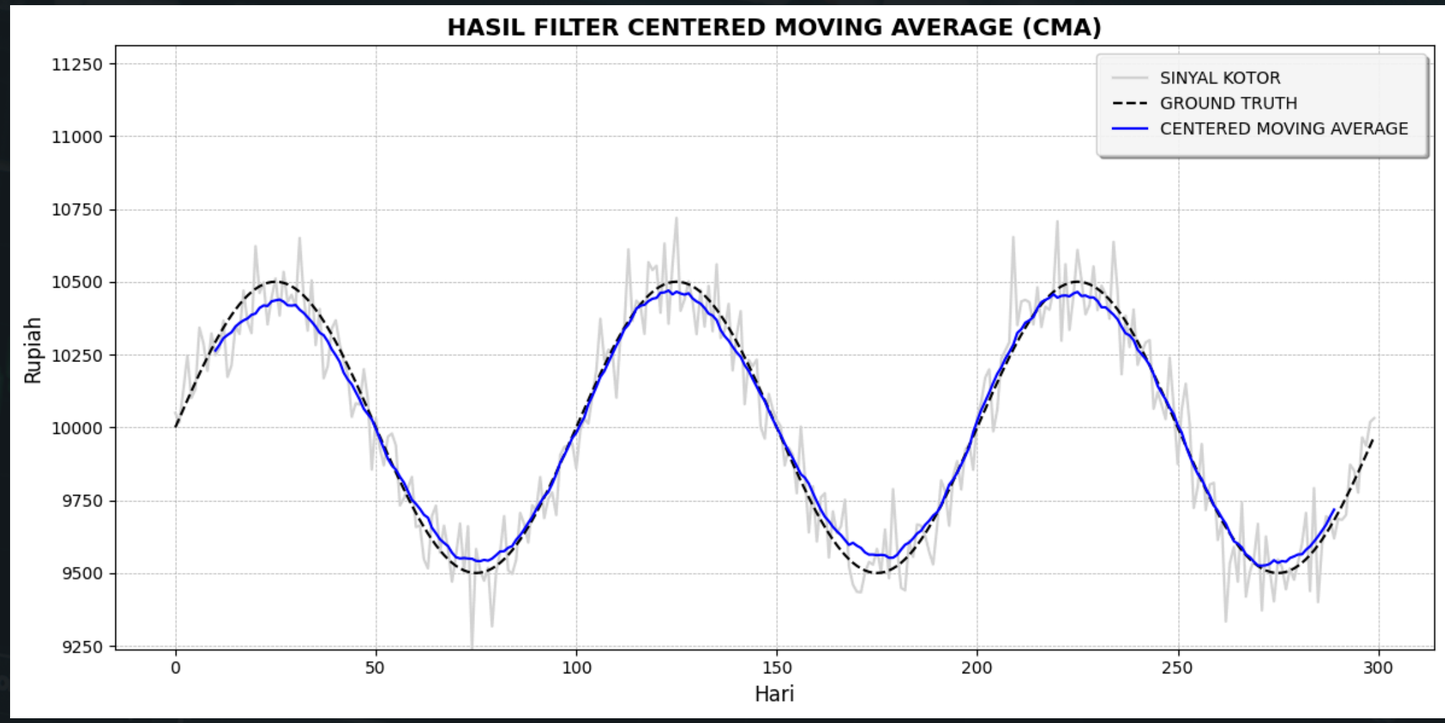


METODE

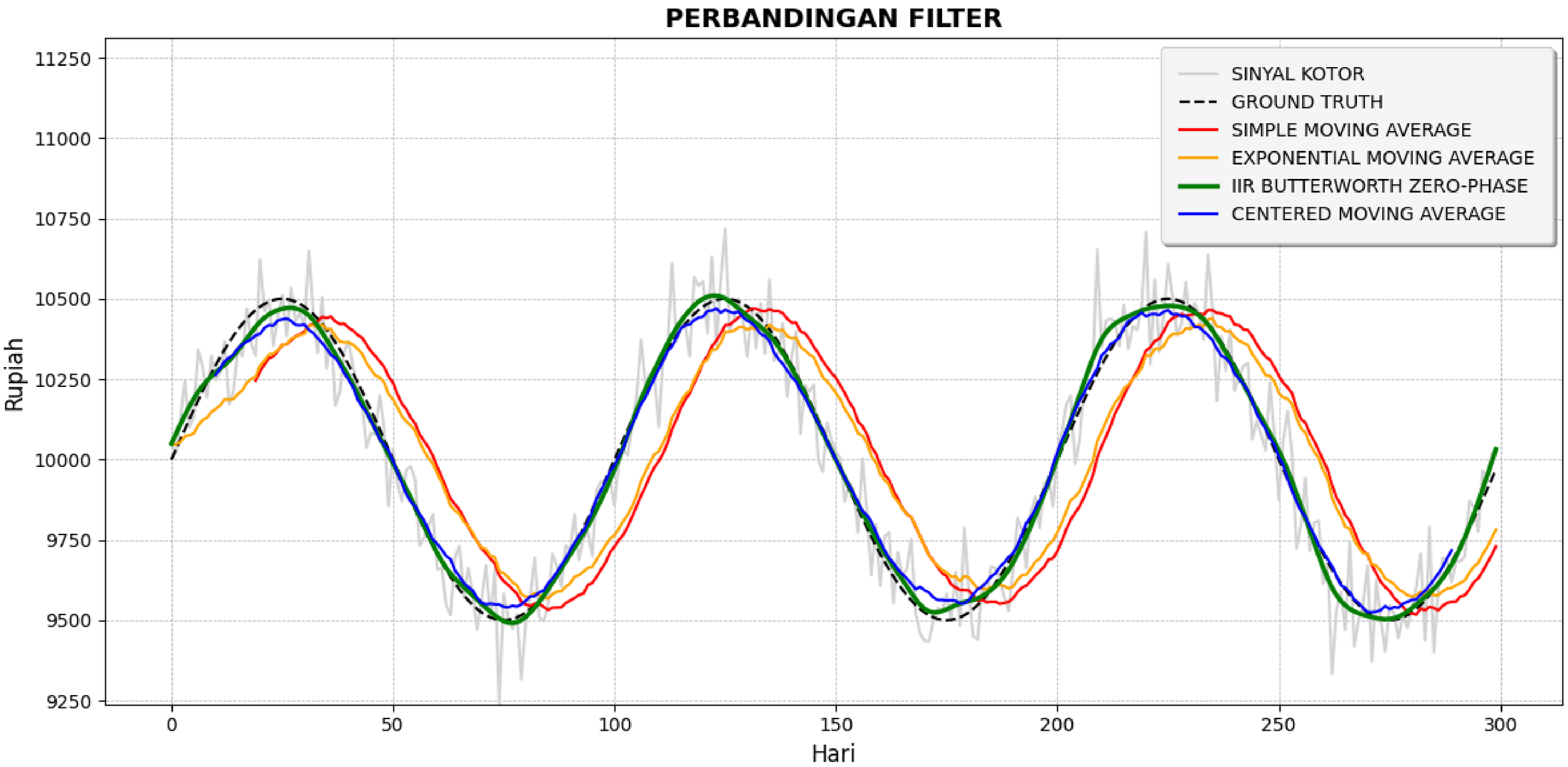
CENTERED MOVING AVERAGE

$$y[i] = \frac{1}{N} \sum_{j=-(N-1)/2}^{+(N-1)/2} x[i-j]$$

```
print('======FILTER CENTERED MOVING AVERAGE (CMA)======')
hasil_cma = np.full_like(harga_pasar, np.nan)
window_cma = filter_hari


if window_cma % 2 == 0:
    window_cma += 1

half_window = window_cma // 2

print('Window CMA:', window_cma)
print('Sisi Kiri & Kanan:', half_window)


for i in range(half_window, total_data - half_window):


    data_window = harga_pasar[i - half_window : i + half_window + 1]
    hasil_cma[i] = np.mean(data_window)

```



HASIL


PERBANDINGAN FILTER
SINYAL KOTOR
GROUND TRUTH
SIMPLE MOVING AVERAGE
EXPONENTIAL MOVING AVERAGE
IIR BUTTERWORTH ZERO-PHASE
CENTERED MOVING AVERAGE
Rupiah
Hari
11250
11000
10750
10500
10250
10000
9750
9500
9250
0
50
100
150
200
250
300

HASIL

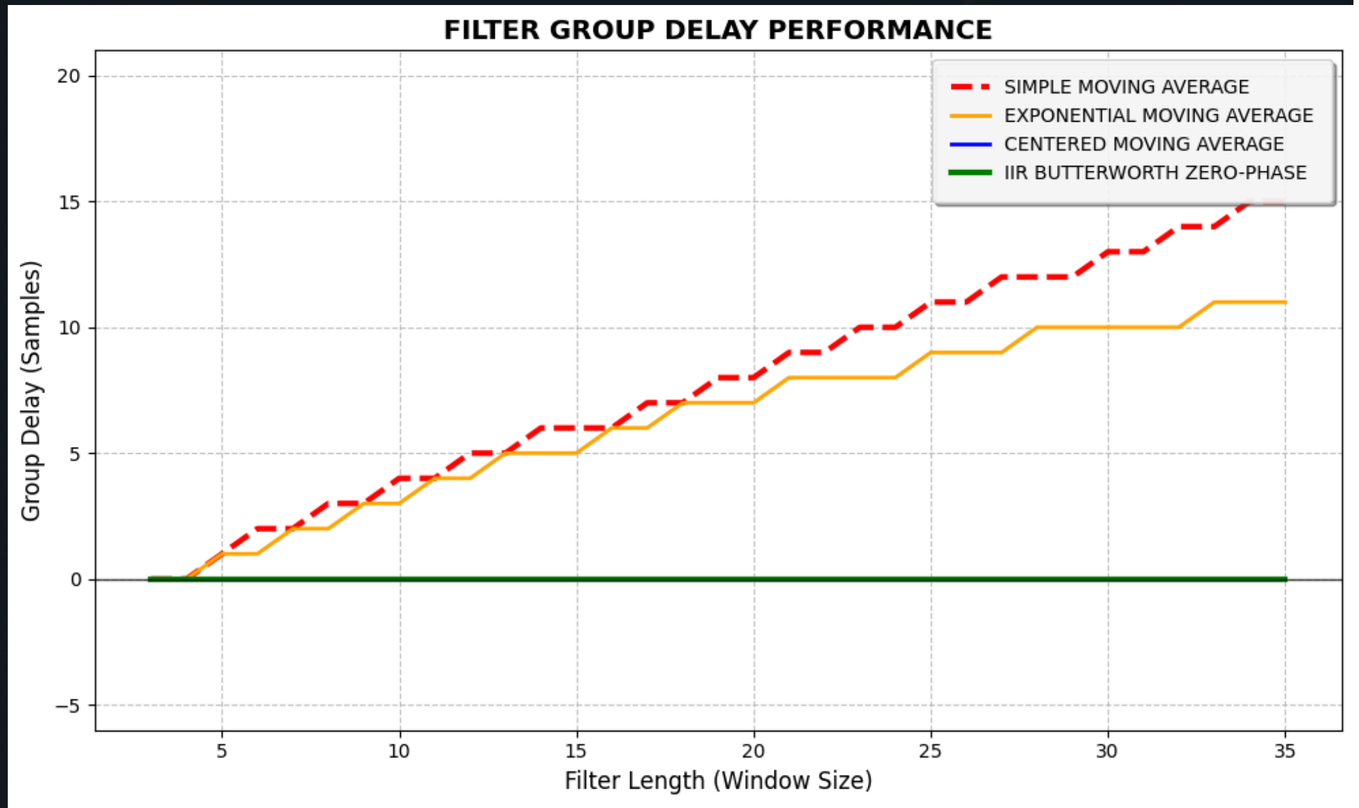
EVALUASI

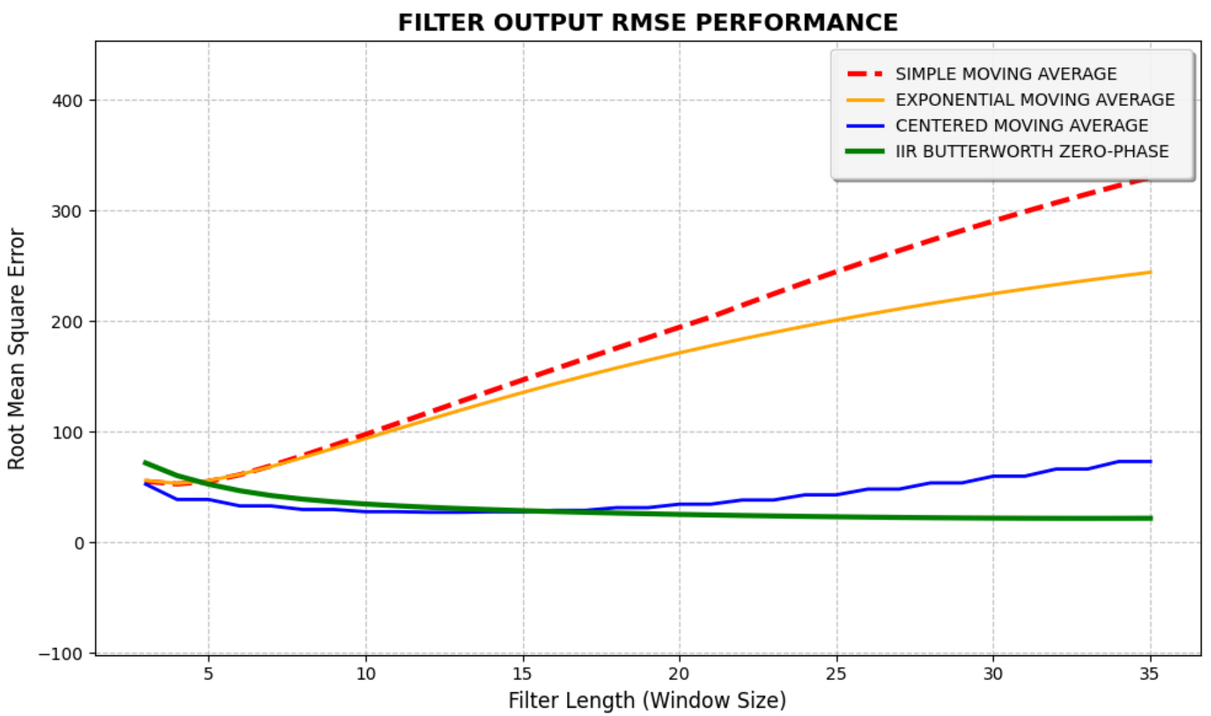
```
===   ====== HASIL EVALUASI GROUND TRUTH ======
      Window : 20

      ====== HASIL EVALUASI RMSE (Single Test) ======
      RMSE SMA              : 194.2365
      RMSE EMA              : 171.0803
      RMSE Zero-Phase       : 25.0285
      RMSE CMA              : 34.2637

      ====== HASIL EVALUASI SNR (Single Test) ======
      SNR SMA               : 5.2761 dB
      SNR EMA               : 6.3787 dB
      SNR Zero-Phase        : 23.0740 dB
      SNR CMA               : 20.3460 dB

      ====== HASIL EVALUASI LAG (Single Test) ======
      LAG SMA               : 8 sampel
      LAG EMA               : 7 sampel
      LAG Zero-Phase        : 0 sampel
      LAG CMA               : 0 sampel
```

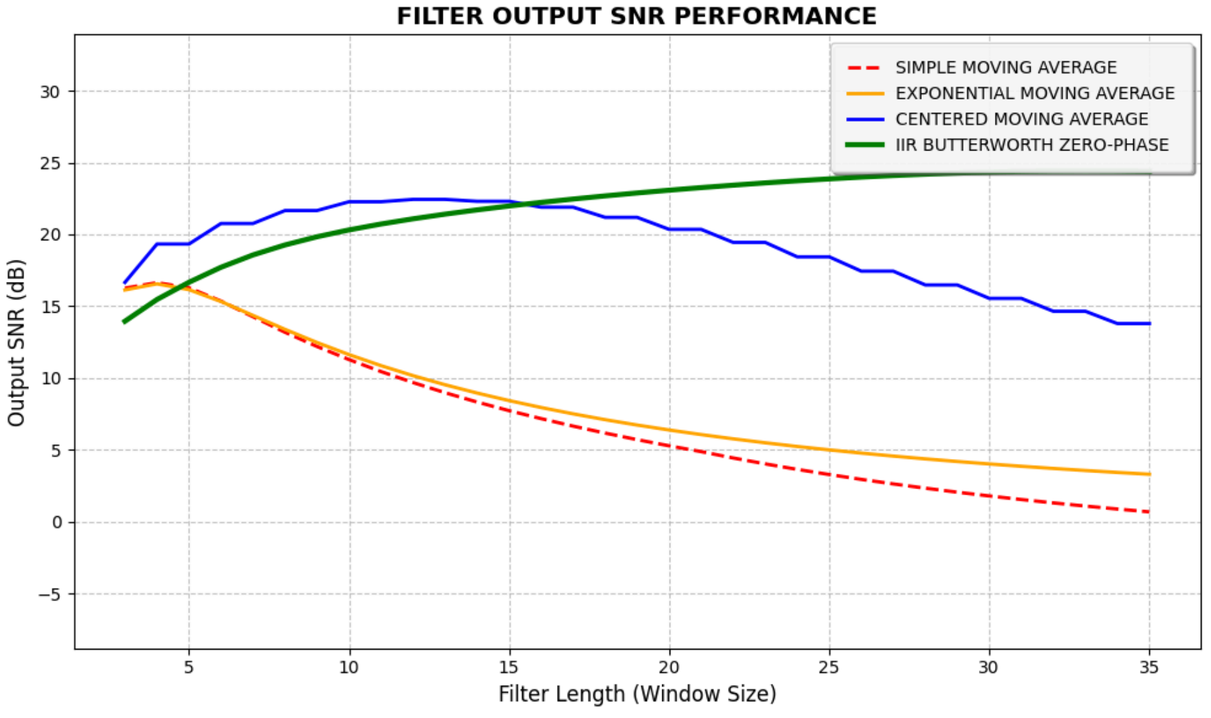
```python
print('====== HASIL EVALUASI GROUND TRUTH ======')
print('Window :', filter_hari)
ground_truth = harga_base + sinyal_bersih

start_valid = filter_hari


valid_target = ground_truth[start_valid:]
valid_sma = hasil_sma[start_valid:]
valid_ema = hasil_ema[start_valid:]
valid_iir = hasil_iir_zp[start_valid:]
valid_cma = hasil_cma[start_valid:]

signal_power = np.var(valid_target)

print('\n====== HASIL EVALUASI RMSE (Single Test) ======')

mse_ma = np.mean((valid_sma - valid_target)**2)
mse_ema = np.mean((valid_ema - valid_target)**2)
mse_iir = np.mean((valid_iir - valid_target)**2)


mse_cma = np.nanmean((valid_cma - valid_target)**2)


print('\n====== HASIL EVALUASI SNR (Single Test) ======')

snr_sma = 10 * np.log10(signal_power / mse_ma)
snr_ema = 10 * np.log10(signal_power / mse_ema)
snr_iir = 10 * np.log10(signal_power / mse_iir)
snr_cma = 10 * np.log10(signal_power / mse_cma)


print('\n====== HASIL EVALUASI LAG (Single Test) ======')


norm_target = valid_target - np.mean(valid_target)
norm_sma = valid_sma - np.mean(valid_sma)
norm_ema = valid_ema - np.mean(valid_ema)
norm_iir = valid_iir - np.mean(valid_iir)


mask_cma = ~np.isnan(valid_cma)
cma_clean = valid_cma[mask_cma]
target_clean_for_cma = valid_target[mask_cma]


norm_cma = cma_clean - np.mean(cma_clean)
norm_target_cma = target_clean_for_cma - np.mean(target_clean_for_cma)

def hitung_lag_revised(filter_signal, target_signal):
    corr = correlate(filter_signal, target_signal, mode='full')
    lags = np.arange(-len(filter_signal) + 1, len(filter_signal))
    lag_idx = lags[np.argmax(corr)]
    return lag_idx

lag_sma = hitung_lag_revised(norm_sma, norm_target)
lag_ema = hitung_lag_revised(norm_ema, norm_target)
lag_iir = hitung_lag_revised(norm_iir, norm_target)


lag_cma = hitung_lag_revised(norm_cma, norm_target_cma)
```

HASIL

EVALUASI







KESIMPULAN

1. FILTER ZERO-PHASE TERBUKTI 0 DELAY
2. FILTER ZERO-PHASE SKOR ERROR KECIL
3. FILTER ZERO-PHASE COCOK DIGUNAKAN UNTUK GROUND TRUTH
4. PERLU ADANYA PENGORBANAN ANTARA DELAY DAN DATA REALTIME

REFERENSI

UTAMA:

[1] T. Pereira, V. Santos, T. Gameiro, C. Viegas, and N. Ferreira, “Evaluation of Different Filtering Methods Devoted to Magnetometer Data Denoising,” Electronics 2024, Vol. 13, Page 2006, vol. 13, no. 11, p. 2006, May 2024, doi: 10.3390/ELECTRONICS13112006.

SUPPORT:

[2] “11.2. Using IIR filters — Digital Signals Theory.” Accessed: Dec. 13, 2025. [Online]. Available: https://brianmcfee.net/dstbook-site/content/ch11-iir/UsingIIR.html

[3] 2020 International Joint Conference on Neural Networks (IJCNN) : 2020 conference proceedings. IEEE, 2020.

[4] “Exponential Moving Average (EMA) Filters | mbedded.ninja.” Accessed: Nov. 26, 2025. [Online]. Available: https://blog.mbedded.ninja/programming/signal-processing/digital-filters/exponential-moving-average-ema-filter/

[5] I. O. Ajao, A. C. Adeyeye, and U. K. Ugochukwu, “Investigating the Robustness of Filters for Integrated Processes in Business Cycles,” Scholars Journal of Physics, Mathematics and Statistics, vol. 11, no. 05, pp. 105–112, Jun. 2023, doi: 10.36347/SJPMS.2023.V10I05.001.

[6] T. O. Hodson, “Root-mean-square error (RMSE) or mean absolute error (MAE): when to use them or not,” Geoscientific Model Development, vol. 15, no. 14, pp. 5481–5487, Jul. 2022, doi: 10.5194/GMD-15-5481-2022.

[7] F. Gustafsson, “Determining the initial states in forward-backward filtering,” IEEE Transactions on Signal Processing, vol. 44, no. 4, pp. 988–992, 1996, doi: 10.1109/78.492552.

THANK YOU FOR YOUR ATTENTION