Task_AR_MA

July 2, 2025

# 1 Task_AR_MA

## 1.1 Task description

Each task consists of generating a synthetic signal, constructing at least two different models, estimating their parameters, and comparing model quality.

Variant 6: - Simulate an AR(3) signal. Fit and compare AR(3) using Yule-Walker and Burg. - Compare frequency-domain PSDs.

## 1.2 Python code

[2]:
```
! pip install spectrum
```

```
Collecting spectrum
  Downloading spectrum-0.9.0.tar.gz (231 kB)
  Preparing metadata (setup.py): started
  Preparing metadata (setup.py): finished with status 'done'
Collecting easydev (from spectrum)
  Downloading easydev-0.13.3-py3-none-any.whl.metadata (4.0 kB)
Requirement already satisfied: numpy in e:\users\dawid\anaconda3\lib\site-
packages (from spectrum) (1.26.4)
Requirement already satisfied: scipy in e:\users\dawid\anaconda3\lib\site-
packages (from spectrum) (1.13.1)
Requirement already satisfied: matplotlib in e:\users\dawid\anaconda3\lib\site-
packages (from spectrum) (3.9.2)
Requirement already satisfied: colorama<0.5.0,>=0.4.6 in
e:\users\dawid\anaconda3\lib\site-packages (from easydev->spectrum) (0.4.6)
Collecting colorlog<7.0.0,>=6.8.2 (from easydev->spectrum)
  Downloading colorlog-6.9.0-py3-none-any.whl.metadata (10 kB)
Collecting line-profiler<5.0.0,>=4.1.2 (from easydev->spectrum)
  Downloading line_profiler-4.2.0-cp312-cp312-win_amd64.whl.metadata (35 kB)
Collecting pexpect<5.0.0,>=4.9.0 (from easydev->spectrum)
  Downloading pexpect-4.9.0-py2.py3-none-any.whl.metadata (2.5 kB)
Collecting platformdirs<5.0.0,>=4.2.0 (from easydev->spectrum)
  Downloading platformdirs-4.3.8-py3-none-any.whl.metadata (12 kB)
Requirement already satisfied: contourpy>=1.0.1 in
e:\users\dawid\anaconda3\lib\site-packages (from matplotlib->spectrum) (1.2.0)
Requirement already satisfied: cycler>=0.10 in
```

```
e:\users\dawid\anaconda3\lib\site-packages (from matplotlib->spectrum) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in
e:\users\dawid\anaconda3\lib\site-packages (from matplotlib->spectrum) (4.51.0)
Requirement already satisfied: kiwisolver>=1.3.1 in
e:\users\dawid\anaconda3\lib\site-packages (from matplotlib->spectrum) (1.4.4)
Requirement already satisfied: packaging>=20.0 in
e:\users\dawid\anaconda3\lib\site-packages (from matplotlib->spectrum) (24.1)
Requirement already satisfied: pillow>=8 in e:\users\dawid\anaconda3\lib\site-
packages (from matplotlib->spectrum) (10.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in
e:\users\dawid\anaconda3\lib\site-packages (from matplotlib->spectrum) (3.1.2)
Requirement already satisfied: python-dateutil>=2.7 in
e:\users\dawid\anaconda3\lib\site-packages (from matplotlib->spectrum)
(2.9.0.post0)
Requirement already satisfied: ptyprocess>=0.5 in
e:\users\dawid\anaconda3\lib\site-packages (from
pexpect<5.0.0,>=4.9.0->easydev->spectrum) (0.7.0)
Requirement already satisfied: six>=1.5 in e:\users\dawid\anaconda3\lib\site-
packages (from python-dateutil>=2.7->matplotlib->spectrum) (1.16.0)
Downloading easydev-0.13.3-py3-none-any.whl (57 kB)
Downloading colorlog-6.9.0-py3-none-any.whl (11 kB)
Downloading line_profiler-4.2.0-cp312-cp312-win_amd64.whl (128 kB)
Downloading pexpect-4.9.0-py2.py3-none-any.whl (63 kB)
Downloading platformdirs-4.3.8-py3-none-any.whl (18 kB)
Building wheels for collected packages: spectrum
  Building wheel for spectrum (setup.py): started
  Building wheel for spectrum (setup.py): finished with status 'done'
  Created wheel for spectrum: filename=spectrum-0.9.0-cp312-cp312-win_amd64.whl
size=228366
sha256=286267087610e3c22788639a483b6f4d41388843e39d442676a6728d01cf71bf
  Stored in directory: c:\users\dawid\appdata\local\pip\cache\wheels\19\a0\e0\e0
4656d89dd723adbe6ea41ab5fe702f5d4ccf95653eb54b04
Successfully built spectrum
Installing collected packages: platformdirs, pexpect, line-profiler, colorlog,
easydev, spectrum
  Attempting uninstall: platformdirs
    Found existing installation: platformdirs 3.10.0
    Uninstalling platformdirs-3.10.0:
      Successfully uninstalled platformdirs-3.10.0
  Attempting uninstall: pexpect
    Found existing installation: pexpect 4.8.0
    Uninstalling pexpect-4.8.0:
      Successfully uninstalled pexpect-4.8.0
Successfully installed colorlog-6.9.0 easydev-0.13.3 line-profiler-4.2.0
pexpect-4.9.0 platformdirs-4.3.8 spectrum-0.9.0
```

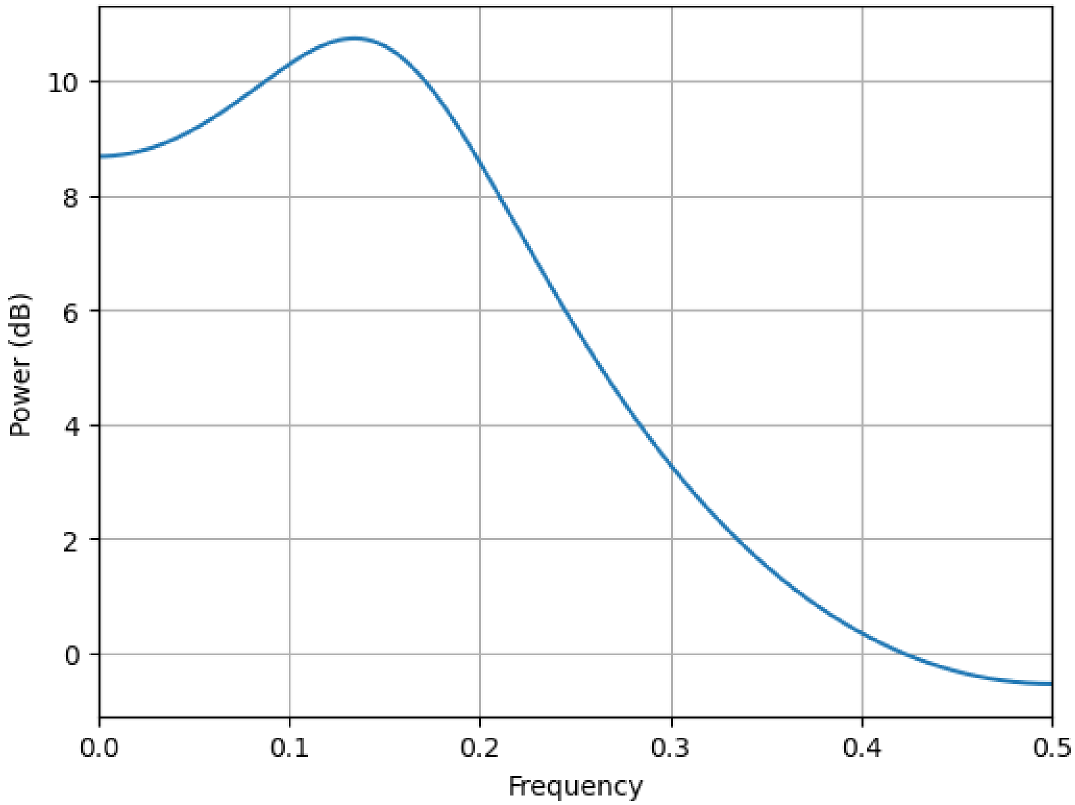
[17]:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import lfilter, freqz
from statsmodels.regression.linear_model import yule_walker
from spectrum import pburg

# 1. Simulate AR(3) Signal
np.random.seed(42)  # For reproducibility
N = 1000            # Number of samples
w = np.random.normal(0, 1, N)  # White Gaussian noise

# True AR(3) coefficients (a1, a2, a3)
a_true = [-0.75, 0.5, -0.25]

ar_signal = lfilter([1], a_true, w)
```

[18]:

```
# 2. Estimate Parameters

# Yule-Walker Estimation
rho_yw, _ = yule_walker(ar_signal, order=3)
a_yw = -rho_yw  # Convert to AR coefficients

# Burg Estimation
burg = pburg(ar_signal, order=3, NFFT=1024)
burg.plot()

a_burg = burg.ar.real

# 3. Compare Parameters
print("True Coefficients:", a_true)
print(f"Yule-Walker Estimates: {a_yw}")
print(f"Burg Estimates: {a_burg}")
```

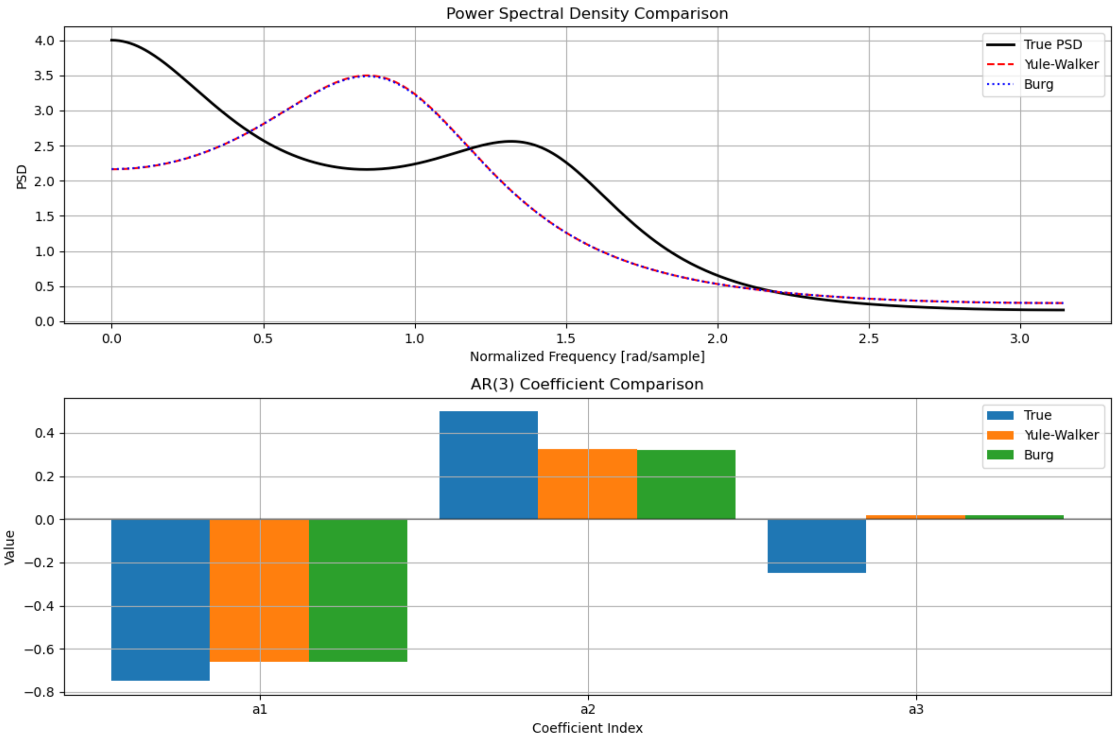
```
True Coefficients: [-0.75, 0.5, -0.25]
Yule-Walker Estimates: [-0.66034807  0.323189    0.01716961]
Burg Estimates: [-0.65969626  0.32231419  0.01687455]
```



[19]:

```
# 4. Compute PSDs

# Frequency vector
freqs = np.linspace(0, np.pi, 512)

# True PSD
_, h_true = freqz(1, np.r_[1, a_true], worN=freqs)
psd_true = np.abs(h_true)**2

# Yule-Walker PSD
_, h_yw = freqz(1, np.r_[1, a_yw], worN=freqs)
psd_yw = np.abs(h_yw)**2

# Burg PSD
_, h_burg = freqz(1, np.r_[1, a_burg], worN=freqs)
psd_burg = np.abs(h_burg)**2

# 5. Plot Results
plt.figure(figsize=(12, 8))

# PSD Comparison
```

```
plt.subplot(2, 1, 1)
plt.plot(freqs, psd_true, 'k-', linewidth=2, label='True PSD')
plt.plot(freqs, psd_yw, 'r--', label='Yule-Walker')
plt.plot(freqs, psd_burg, 'b:', label='Burg')
plt.title('Power Spectral Density Comparison')
plt.xlabel('Normalized Frequency [rad/sample]')
plt.ylabel('PSD')
plt.legend()
plt.grid(True)

# Coefficient Comparison
ax = plt.subplot(2, 1, 2)
index = np.arange(3)
bar_width = 0.3
plt.bar(index, a_true, bar_width, label='True')
plt.bar(index + bar_width, a_yw, bar_width, label='Yule-Walker')
plt.bar(index + 2*bar_width, a_burg, bar_width, label='Burg')
plt.axhline(0, color='k', linestyle='-', alpha=0.3)
plt.title('AR(3) Coefficient Comparison')
plt.xlabel('Coefficient Index')
plt.ylabel('Value')
plt.xticks(index + bar_width, ['a1', 'a2', 'a3'])
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```


Power Spectral Density Comparison
True PSD
Yule-Walker
Burg
PSD
Normalized Frequency [rad/sample]
AR(3) Coefficient Comparison
True
Yule-Walker
Burg
Value
a1
a2
a3
Coefficient Index