Zad 10 - Zadanie_DMD

Temat: Dynamic Mode Decomposition (DMD)

Treść zadania

Na podstawie danych o dynamicznych stanach układu (zgodnie z wariantem zadania) w macierzach X oraz X' obliczyć na podstawie metody DMD przybliżoną macierz przekształcenia A

Wariant zadania: 14

Kod Python

In [1]:
```python
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import rcParams
import pandas as pd

rcParams.update({'font.size': 14})
plt.rcParams['figure.figsize'] = [10, 6]

# 1. WCZYTANIE DANYCH Z PLIKÓW CSV

# Wczytanie X (23 wiersze × 37 kolumn, pierwsza kolumna to indeks)
data_X = pd.read_csv('War14_X.csv', delimiter=';', decimal=',')
data_X = data_X.values

X = data_X.T  # po transpozycji: (36, 23)

# Wczytanie X' (23 wiersze × 36 kolumn)
data_Xprime = pd.read_csv('War14_Xprime.csv', delimiter=';', decimal=',')
data_Xprime = data_Xprime.values

Xprime = data_Xprime.T  # po transpozycji: (36, 23)

print("X shape:", X.shape)        # (36, 23)
print("Xprime shape:", Xprime.shape)  # (36, 23)
```

```
X shape: (36, 22)
Xprime shape: (36, 22)
```

In [2]:
```python
# 2. IMPLEMENTACJA ALGORYTMU DMD
def DMD(X, Xprime, r):
    """
    Dynamic Mode Decomposition.
    X, Xprime : macierze danych (stany w kolumnach)
    r         : rząd redukcji (r ≤ min(n, m))
    Zwraca: Phi (mody), Lambda (wartości własne), b (współczynniki),
            Atilde (zredukowana macierz dynamiki), A_approx (przybliżona macierz
    """
    # Krok 1: SVD macierzy X
```

```python
    U, Sigma, VT = np.linalg.svd(X, full_matrices=False)
    Ur = U[:, :r]
    Sigmar = np.diag(Sigma[:r])
    VTr = VT[:r, :]

    # Krok 2: Obliczenie zredukowanej macierzy Atilde
    Atilde = np.linalg.solve(Sigmar.T, (Ur.T @ Xprime @ VTr.T).T).T

    # Krok 3: Dekompozycja własna Atilde
    Lambda, W = np.linalg.eig(Atilde)  # Lambda to wektor, W to macierz wektorów
    Lambda = np.diag(Lambda)  # przekształcenie na macierz diagonalną

    # Krok 4: Odtworzenie pełnych modów DMD (Φ)
    Phi = Xprime @ np.linalg.solve(Sigmar.T, VTr).T @ W

    # Współczynniki (amplitudy) modów
    alpha1 = Sigmar @ VTr[:, 0]
    b = np.linalg.solve(W @ Lambda, alpha1)

    # Obliczenie pełnej macierzy A (przybliżenie)
    A_approx = Xprime @ np.linalg.pinv(X)

    return Phi, Lambda, b, Atilde, A_approx
```

In [3]:

```python
# 3. OBLICZENIA DLA r = 21
r = 21
Phi, Lambda, b, Atilde, A_approx = DMD(X, Xprime, r)

print("\n" + "="*60)
print("WYNIKI DMD")
print("="*60)
print(f"Zredukowana macierz Atilde ({Atilde.shape[0]}×{Atilde.shape[1]}):")
print(Atilde)
```

```
============================================================
WYNIKI DMD
============================================================
Zredukowana macierz Atilde (21×21):
[[ 1.21528665e+01  1.06701468e+02  1.32691802e+03 -2.42175166e+04
   1.23117679e+05  4.94039334e+06 -2.98211730e+07  4.88325503e+06
   3.97764233e+09  3.30964248e+10  1.51873336e+11  5.27971358e+10
   3.46897821e+12 -2.59590200e+13  1.33550887e+14 -4.14199255e+15
   5.27660916e+17  8.74433301e+17 -3.06537375e+18  1.74188153e+19
  -3.91291626e+19]
 [ 2.05831233e-07 -8.82048646e+00 -1.09550464e+02  1.99939536e+03
  -1.01645812e+04 -4.07878298e+05  2.46203256e+06 -4.03160965e+05
  -3.28393686e+08 -2.73243697e+09 -1.25386449e+10 -4.35892537e+09
  -2.86398436e+11  2.14317366e+12 -1.10259456e+13  3.41962422e+14
  -4.35636236e+16 -7.21931113e+16  2.53076899e+17 -1.43809536e+18
   3.23049910e+18]
 [ 4.45458073e-08  9.00734978e-01 -1.25725805e+00  2.24199218e+01
  -1.13986160e+02 -4.57397222e+03  2.76093840e+04 -4.52107175e+03
  -3.68262691e+06 -3.06417155e+07 -1.40609132e+08 -4.88812561e+07
  -3.21168961e+09  2.40336808e+10 -1.23645630e+11  3.83478754e+12
  -4.88525144e+14 -8.09577973e+14  2.83801985e+15 -1.61268895e+16
   3.62270148e+16]
 [ 6.60770573e-10 -2.35336347e-02 -9.34895666e-01 -1.16446466e+00
   5.99334819e+00  2.40476590e+02 -1.45156076e+03  2.37694914e+02
   1.93613757e+05  1.61098525e+06  7.39251160e+06  2.56992734e+06
   1.68854273e+08 -1.26356846e+09  6.50065710e+09 -2.01613587e+11
   2.56841626e+13  4.25634842e+13 -1.49208621e+14  8.47869664e+14
  -1.90463183e+15]
 [ 6.64084714e-11  1.79031671e-03  1.37315551e-02  1.00868564e+00
  -8.84834152e-02 -1.95314325e+00  1.17989576e+01 -1.93209797e+00
  -1.57372530e+03 -1.30943601e+04 -6.00875827e+04 -2.08888028e+04
  -1.37247605e+06  1.02704977e+07 -5.28384376e+07  1.63874929e+09
  -2.08765211e+11 -3.45963190e+11  1.21279287e+12 -6.89162794e+12
   1.54811690e+13]
 [-2.01222559e-11  3.50821127e-05 -5.05233428e-04 -5.06030028e-02
  -9.44109457e-01  2.21867669e+00 -1.33328529e+01  2.18359418e+00
   1.77857721e+03  1.47988540e+04  6.79091878e+04  2.36079006e+04
   1.55113135e+06 -1.16074093e+07  5.97164222e+07 -1.85206539e+09
   2.35940194e+11  3.90997242e+11 -1.37066221e+12  7.78871155e+12
  -1.74963541e+13]
 [ 9.47251053e-13 -8.37538136e-06 -3.83770462e-04  3.68545976e-04
  -5.86605905e-03 -8.43826610e-01 -9.84563648e-01  1.60329363e-01
   1.24679612e+02  1.03741240e+03  4.76049256e+03  1.65493417e+03
   1.08735644e+05 -8.13689389e+05  4.18617262e+06 -1.29831380e+08
   1.65396108e+10  2.74092434e+10 -9.60846016e+10  5.45995388e+11
  -1.22650949e+12]
 [ 1.18322945e-13 -6.23815840e-07  7.07826626e-05  3.75656280e-04
  -1.33138137e-03  3.75133266e-02 -9.21217735e-01  3.89963150e-02
  -1.01655327e+01 -8.45618876e+01 -3.88036894e+02 -1.34896832e+02
  -8.86325150e+03  6.63253875e+04 -3.41222983e+05  1.05828055e+07
  -1.34817550e+09 -2.23418016e+09  7.83204069e+09 -4.45051344e+10
   9.99751479e+10]
 [-2.24626432e-14  7.22942059e-10  1.62504447e-06  5.59454915e-05
   1.78942770e-04  3.86360013e-03  5.67347827e-02  9.97611226e-01
  -6.76424062e-01 -5.46571106e+00 -2.50549985e+01 -8.71038607e+00
  -5.72271658e+02  4.28241683e+03 -2.20316701e+04  6.83297693e+05
  -8.70473531e+07 -1.44253823e+08  5.05689664e+08 -2.87355330e+09
   6.45507357e+09]
 [-2.53680184e-15 -1.91492908e-09 -6.55070906e-08  3.74497405e-06
   4.10709914e-05 -2.99190058e-04  3.15863736e-03 -1.99997991e-02
```

```
 -1.19408720e+00 -1.70310452e+00 -7.76451862e+00 -2.69976266e+00
 -1.77403288e+02  1.32754251e+03 -6.82978338e+03  2.11821222e+05
 -2.69845438e+07 -4.47184603e+07  1.56763008e+08 -8.90797046e+08
  2.00106275e+09]
[-4.89781695e-16 -2.39462731e-09  5.30994859e-08 -8.66651455e-07
  1.32631230e-06 -4.42735524e-05  2.26445917e-04 -7.63035057e-03
 -1.73185989e-01 -6.72459451e-01 -1.83742879e+00 -6.11066074e-01
 -4.11829258e+01  3.08177160e+02 -1.58547388e+03  4.91724252e+04
 -6.26422344e+06 -1.03809955e+07  3.63911473e+07 -2.06790665e+08
  4.64529039e+08]
[-3.97086559e-16  1.58477053e-09 -6.30005304e-08  1.78270099e-07
  6.68875856e-06 -8.21295392e-06 -3.83104519e-05 -3.65784489e-03
  6.91993628e-02 -7.71516357e-01  6.99715195e-01  3.28789830e-01
  1.87634574e+01 -1.40431993e+02  7.22477562e+02 -2.24071646e+04
  2.85451624e+06  4.73046986e+06 -1.65829207e+07  9.42315222e+07
 -2.11679181e+08]
[-2.50546711e-17  9.31860817e-11  3.55220074e-11 -1.74594301e-08
  1.09927015e-07  7.03651514e-06  7.70687519e-05  7.86270641e-04
  5.42606716e-04  2.88303645e-03  2.55805366e-01  9.49742217e-01
 -6.27499303e-01  4.40245542e+00 -2.26429868e+01  7.02255211e+02
 -8.94624087e+04 -1.48256024e+05  5.19719599e+05 -2.95327762e+06
  6.63416416e+06]
[-3.03618041e-19 -8.31848404e-13  4.35799628e-10  2.37370263e-09
 -2.91252437e-08 -4.12058193e-07 -7.56839059e-06 -4.83730244e-05
 -2.05744118e-04  2.67730058e-03 -1.45437539e-02 -3.73838264e-02
 -1.02003733e+00  1.90715901e-01 -8.18203059e-01  2.54470126e+01
 -3.24176350e+03 -5.37221132e+03  1.88325806e+04 -1.07015090e+05
  2.40395844e+05]
[ 1.02991555e-19 -7.74535396e-13 -3.89954231e-12  8.73436878e-11
 -4.33190808e-10 -1.89505778e-08 -1.84068836e-07 -4.46829984e-06
 -4.66594968e-05 -1.62411931e-04 -7.88899012e-04  6.30516220e-05
  2.50548532e-02  1.04594715e+00 -2.38000765e-01  7.41073263e+00
 -9.43918997e+02 -1.56425111e+03  5.48356804e+03 -3.11600697e+04
  6.99971501e+04]
[-3.21784598e-21  3.10283215e-14  4.45072156e-13  1.04890757e-11
 -8.20016526e-11 -6.29124861e-10 -8.10390298e-09  6.82419776e-08
 -8.25879603e-07  6.01274330e-06  1.35254393e-05 -1.70595641e-04
 -2.36977167e-03  1.92116735e-03 -1.00276654e+00  8.44583988e-02
 -1.80402045e+01 -2.98901388e+01  1.04813179e+02 -5.95593222e+02
  1.33792433e+03]
[ 1.19355703e-22 -1.60329659e-15 -3.44594031e-14 -4.44032917e-13
 -2.20109095e-13 -2.57729674e-13  6.68124490e-11 -1.74366584e-09
  6.38514921e-08  5.93408093e-08  4.17032782e-08  1.43583251e-06
  5.44772576e-06 -7.47432095e-04  1.78598339e-02 -5.19450865e-02
  2.29177925e+00  3.78245583e+00 -1.33106169e+01  7.56571228e+01
 -1.69954580e+02]
[ 6.79920285e-22 -1.10612548e-15  1.95914404e-14 -1.91408639e-13
  5.06909353e-12  7.25095276e-12 -1.65826714e-10 -1.82243581e-08
 -7.67282818e-08 -9.38464895e-08 -4.90858582e-06  6.51110948e-06
  8.75616100e-05  1.31959399e-03  5.62714819e-02 -9.76176491e-01
 -2.61652730e+00 -4.32485658e+00  1.52939625e+01 -8.69285620e+01
  1.95275118e+02]
[ 6.81696496e-23 -1.16982249e-17 -1.45522636e-14 -1.06028285e-13
  1.20884926e-12  1.55678699e-11  1.77695033e-10  8.58492682e-11
  1.96538203e-08 -1.43632714e-07  3.58305945e-08  2.07664757e-06
  2.68138369e-05 -1.71428381e-04  7.15429818e-03  3.48212182e-02
  5.92891031e-01  1.92481844e+00 -3.24552838e+00  1.84533080e+01
 -4.14836189e+01]
[ 2.09327412e-24 -1.33878902e-17  8.06659925e-17  3.72150434e-15
 -1.00639164e-13  1.69340557e-13  5.15683430e-12  2.65553028e-10
```

```
  2.31582314e-09  3.65524968e-09  6.89274009e-08  1.35324727e-08
 -9.49095863e-07 -3.45825681e-05 -4.92683477e-04  4.83581267e-03
  3.96069561e-02  4.31673544e-02 -1.12336201e+00  6.96421119e-01
 -1.70460930e+00]
[-8.80572679e-24 -1.61704689e-16 -1.96561358e-15  1.27999684e-14
 -1.74206958e-14 -2.65952722e-12  3.14294687e-11  1.88431306e-10
 -4.93210526e-10 -2.02540236e-08 -4.34573789e-08 -5.57553869e-08
 -2.51255299e-06 -2.08949251e-05 -3.40666911e-04  1.02601504e-02
 -2.56976976e-01 -3.94276491e-01  1.59675534e+00 -8.50033262e+00
  1.90095050e+01]]
```

In [9]:

```python
print("\nWartości własne A (diagonalne):")
print(np.diag(Lambda))
```

```
Wartości własne A (diagonalne):
[ 5.06541138e+00+26.52057661j  5.06541138e+00-26.52057661j
  1.58410134e+01 +0.j          1.21528646e+01 +0.j
  5.91826040e+00+11.80929035j  5.91826040e+00-11.80929035j
  8.58077554e+00 +7.65805853j  8.58077554e+00 -7.65805853j
 -1.47382145e+01 +0.j         -1.23537164e+01 +0.j
 -9.58764550e+00 +5.4480609j  -9.58764550e+00 -5.4480609j
 -4.34456238e-01+11.06396454j -4.34456238e-01-11.06396454j
 -5.61211456e+00 +9.44976915j -5.61211456e+00 -9.44976915j
  6.47804140e+00 +0.j         -3.11598531e+00 +0.j
 -8.03546207e-02 +0.j         -1.01225200e-09 +0.j
 -9.69960800e-08 +0.j        ]
```

In [8]:

```python
print(f"\nPrzybliżona macierz A ({A_approx.shape[0]}×{A_approx.shape[1]}):")
print(A_approx)
```

```
Przybliżona macierz A (36×36):
[[ 5.41686283e-28  1.08608626e-34  3.07788698e-36 ...  4.47510028e-31
   1.50945544e-32 -4.59312054e-34]
 [-1.06158220e-27 -2.31133408e-34 -6.50010713e-36 ...  1.53556261e-30
  -4.45419222e-32  1.00155278e-33]
 [-6.08880805e-28 -1.01927510e-34 -2.94300308e-36 ... -1.98372627e-30
   3.98545437e-32 -2.44505072e-33]
 ...
 [ 2.02715794e-06  1.19346908e-12  1.64309790e-15 ...  4.61002065e-10
   1.00000000e+00  5.87771452e-13]
 [ 2.97826094e-05  1.79007608e-12  7.13844975e-14 ...  8.68344364e-10
   4.98822871e-10  1.00000000e+00]
 [ 1.41267015e+07  3.36177728e+00  9.38485915e-02 ...  2.69954983e+02
  -9.88060785e+01  1.41096173e+00]]
```