Lund University
Faculty of Engineering
Name: ZHANG Yifei
Email: yi4840zh-s@lu.se
Date: January 4, 2024

FRTN65 - Modeling and Learning from Data, Autumn 2023

EXAMS

Answer to Task 1

a)

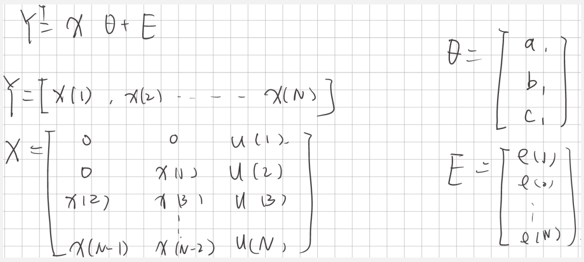

Figure 1: Task1

b) U and E should be independent. If E is white noise, then as n approaches positive infinity, it becomes bias-free.

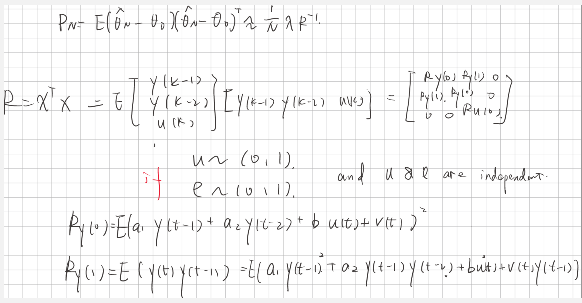

Figure 2: Task1 b

Answer to Task 2: Supervised Learning

a) I noticed that in this code, there is no regularization applied to the data and no filtering for the data.

First, I detected outliers and scaled the dataset, implementing normalization. Next, I selected relevant features through a heatmap. Finally, I used a random forest for classification.

b)

```
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier(n_estimators=50, oob_score=True, random_state=1234)
rf.fit(x_train,y_train)
yp = rf.predict(x_test)
acc = metrics.accuracy_score(y_test,yp)          # classifier accuracy
rmserr = rmse(yp,y_test.values)  # rms error
# score = rf.score(x_train,y_train)
print(" e r r  ", rmserr)
```

However, the results are not very satisfactory. The RMS error for k-NN is 2.3, and when switched to a random forest, the RMS error becomes 2.5.

Answer to Task 3: System Identification

Firstly, I observed the data and found that it was consistently zero between 1900 and 2100, which is clearly abnormal. Therefore, I removed this portion of the data. Secondly, I tested the parameters of the ARX model and selected [15 15 1]. Then, I compared it with the OE model, and the OE model's performance was slightly better than ARX. From the Bode plot and residual plot, I obtained satisfactory results

```
%%
y(1901:2100) = [];
u(1901:2100) = [];
N = length(y);


%%
N = [6 6 1];
O = oe(z,N);
compare(ztest,O,M,Inf)
% resid(ztest,O)
present(O)
```

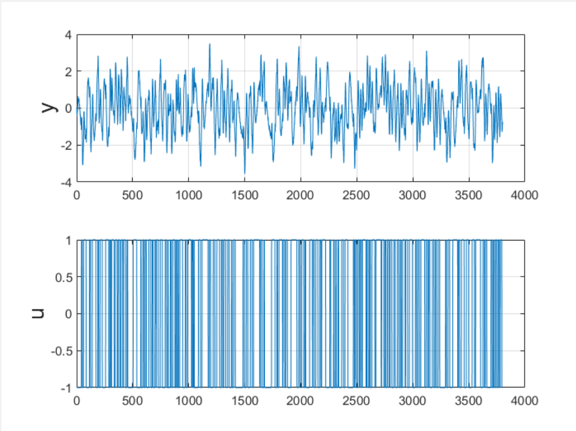

Figure 3: Task3 plot

Compare OE and ARX.

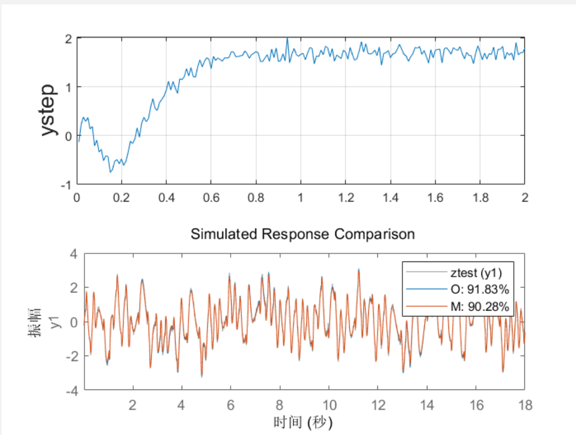

Figure 4: Task3 OE & ARX result

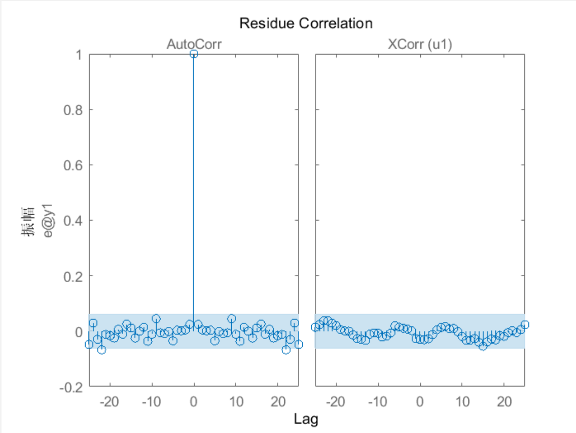

Figure 5: Task3 OE residual

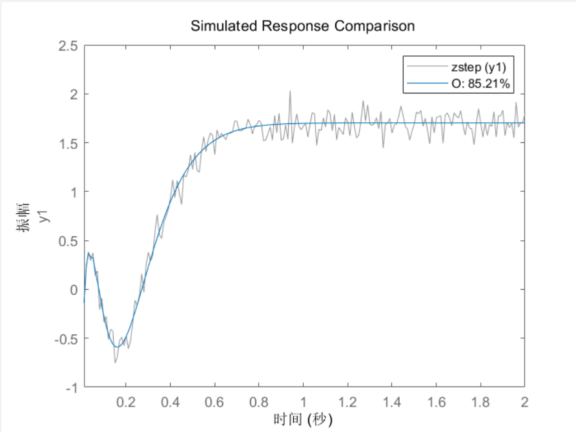

Figure 6: Task3 OE response

Answer to Task 4: Causal Inference

a) 1 b) 2 c) 1

dBecause when Y S + W, although it blocks the path S -¿ W -¿ M, there is still an unblocked backdoor path: S -¿ W ¡- D -¿ M, which affects the coefficient of S. To resolve this issue, it is necessary to choose Y S + W + D. The results I obtained after assigning a value to C in Colab also confirmed this hypothesis.

e)

```
c1 = 1
c2 = 2
c3 = 3
c4 = 4
c5 = 5

results1 = smf.ols('Y ¬ S + W + D', data=dat1).fit()
print(results1.summary())
```

It can be observed that the coefficient of S is approximately equal to C1.

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      Y   R-squared:                       0.996
Model:                            OLS   Adj. R-squared:                  0.996
Method:                 Least Squares   F-statistic:                 8.466e+05
Date:                Thu, 04 Jan 2024   Prob (F-statistic):               0.00
Time:                        14:14:47   Log-Likelihood:                -14176.
No. Observations:               10000   AIC:                         2.836e+04
Df Residuals:                    9996   BIC:                         2.839e+04
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.0022      0.010     -0.224      0.823      -0.022       0.017
S              1.0154      0.042     24.317      0.000       0.934       1.097
W              1.9965      0.010    196.940      0.000       1.977       2.016
D              3.0370      0.052     58.707      0.000       2.936       3.138
==============================================================================
Omnibus:                        9.014   Durbin-Watson:                   2.037
Prob(Omnibus):                  0.011   Jarque-Bera (JB):                9.535
Skew:                           0.044   Prob(JB):                      0.00850
Kurtosis:                       3.124   Cond. No.                         43.6
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

Figure 7: Task4 result

Answer to Task 5: Model Reduction

a) By changing the sensor position and testing the Bode plot, we can observe that when the position is less than 0.75, the accuracy of the second-order system can be maintained at 0.1 rad/s. As the sensor position gets closer to the tail, the error gradually increases in the first 30 seconds. When the position is greater than 0.75, a higher-order system is required.

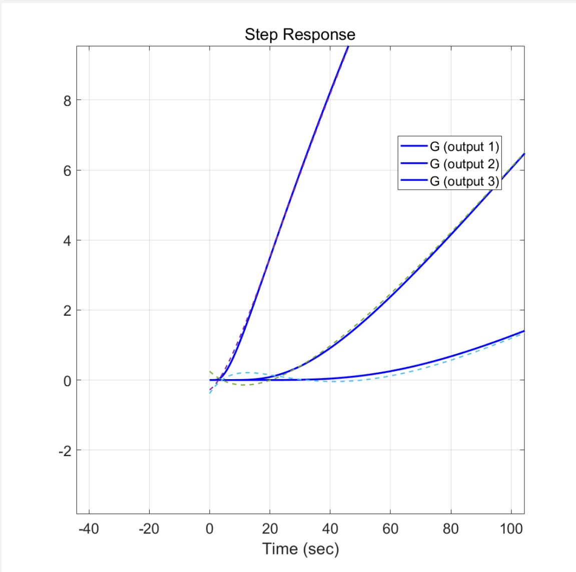

Figure 8: Task52

b) At 20% and 50% of the length of the rod, using a second-order system can maintain accuracy, but there is still some error in the first 30 seconds. For 80%, a third-order system is needed, and the error in the first 30 seconds has been significantly improved.

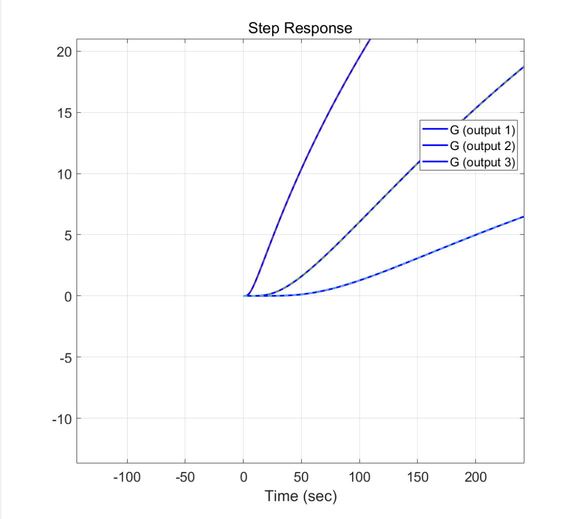

Figure 9: Task53

Answer to Task 6: System Identification Theory

$$x(t) = a_1 x(t-1) + a_2 x(t-2) + b u(t) + e(t).$$

$$E(\hat{\theta}) = \frac{1}{N} \sum_{k=1}^{N} E(y_k)$$

$$Var(x) = E(x-\bar{x})^2 \qquad \bar{x} = E(x).$$

$$Var(ax) = a^2(x-\bar{x}). \qquad Var(Ax) = A\,var(x)A^T$$

$$E(Ax) = AE(x).$$

$$R = E\left[\begin{pmatrix} y(k-1) \\ y(k-2) \\ u(k) \end{pmatrix} \begin{bmatrix} y(k-1) & y(k-2) & u(k) \end{bmatrix}\right] = \begin{bmatrix} R_y(0) & R_y(1) & 0 \\ R_y(1) & R_y(0) & 0 \\ 0 & 0 & R_u(0) \end{bmatrix}.$$

since $E(u(t)u(s)) = E(e(t)e(s)) = E(u(t).u(s)) = 0 \qquad (t \neq s)$

$E(u^2(t)) = E(e^2(t)) = 1$ and $E(e(t)u(t)) = \rho. \qquad |\rho| \leq 1$

$$R_y(0) = E(a_1 y(t-1) + a_2 y(t-2) + b\,u(t) + v(t))^2 \qquad R_u(0) = 0$$

$$R_y(1) = E(a_1 y(t-1) + a_2 y(t-2) + b\,u(t) + \rho)^2. \qquad R_u(0) = 0.$$

Figure 10: Task6