FRTN65 - Exam solutions 2022 April

1 Dimensional analysis

a) The product is dimensionless if the following equations (one for each unit) hold:

$$\begin{cases} a+b+d+f & = 0 \ \text{[kg]} \\ 3b+2d+2f & = 0 \ \text{[m]} \\ -2b-d-2f & = 0 \ \text{[s]} \\ -2b+c & = 0 \ \text{[Coulomb]} \end{cases}$$

We can rewrite the system of equations as

$$\begin{bmatrix} 1 & 1 & 0 & 1 & 1 \\ 0 & 3 & 0 & 2 & 2 \\ 0 & -2 & 0 & -1 & -2 \\ 0 & -2 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ f \end{bmatrix} = 0,$$

which we can solve by hand or using MATLAB's `null` function, giving the solutions

$$\begin{bmatrix} a \\ b \\ c \\ d \\ f \end{bmatrix} = t \begin{bmatrix} 1 \\ 2 \\ 4 \\ -2 \\ -1 \end{bmatrix}$$

where $t \in \mathbb{Z}$. This correspond to a dimensionless variable(s) of the form Π^t where

$$\Pi = \frac{mk^2e^4}{\hbar E}$$

b) Selecting the solution with $f = -1$ corresponds to $t = 1$ in the expression above. This together with $D = 1/2$ gives the expression for the energy as (easily found online also)

$$E = \frac{mk^2e^4}{2\hbar} \approx 2 \cdot 10^{-18} J$$

This value corresponds well with what you find online (though often expressed as 13.6 eV). Using the numerical values in the problem, it is off by around 10%, but inserting more exact values for $m, k, e, \hbar$ gives higher accuracy and a perfect match.

2 Supervised Learning

a) Looking at the scatter plots under Data visualization we see that the two features "body mass" and "flipper length" overlap significantly for two of the classes (Adelie and Chinstrap).

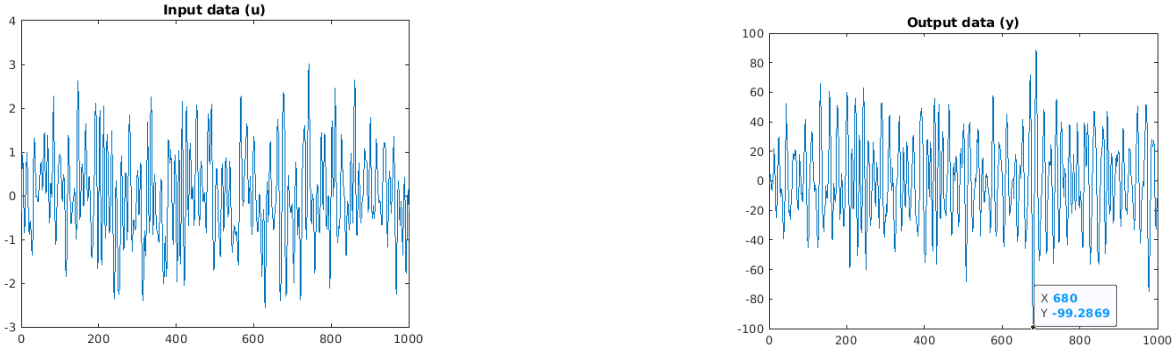
b) The addition of the scaler causes the improvement (removing it yields the same as previously). The two features examined typically take values of different magnitudes, which indicates that scaling might help.

c) Choosing two features that are clearly separated in the scatter plots (for example "bill length" and "flipper length") yields a much better results (95.6% accuracy for the mentioned features).

d) The `dropna` command is necessary since some of the penguins do not have data available for the "gender" feature. The `get_dummies` command transforms categorical variables ("gender", "island") into one-hot encoded form. The two `make_pipeline` commands specify two different models (and scaling). The `RepeatedKFold` command performs cross-validation of the data split into the specified number of partitions and performs the specified number of iterations. The LDA performs slightly better, probably due to the built-in assumption that the data originates from a Gaussian distribution, which seems approximately correct in this case, as can be seen from the scatterplot.

3 System Identification Hands-on

Inspecting the data we find no obvious outliers. The most deviating datapoint was within four standard deviations. This has a likelihood of around 0.1% and occurs once out of 1000 datapoints in the output, which is not unrealistic. Looking at the spectrum of input and output data we see that the input is exciting up to around 0.25 times the Nyquist frequency. The data should be split into appropriate training and testing sets (for example 50/50 or 70/30).

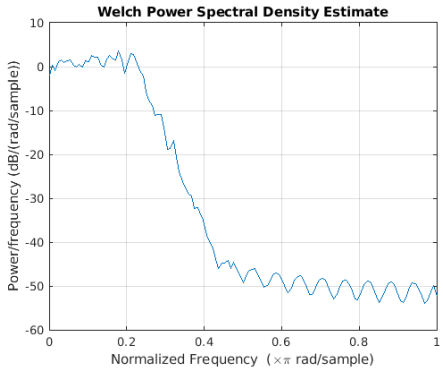


(a) (b)

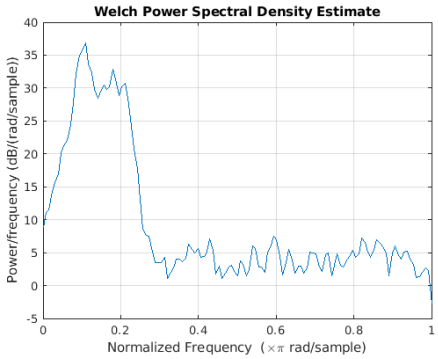
Figure 1: Input and output data in problem 3.

Testing a range of different ARX models we find the optimal, which gives an 81% fit of the testing data. Inspection of the residuals show significant protrusion beyond the expected interval (Figure 3a). Inspecting the bode plot shows suspiciously high gain for high frequencies (Figure 3). An uncertain fit in this frequency region is expected since we only have excitation up to about 25% of the Nyquist frequency. Since we are not happy with the ARX results we continue looking at OE, ARMAX and BJ models.

Fitting an OE model to the data, after tuning of the degree parameters, achieves an accuracy of 89.24% for the choice of degrees [441] (`oe[441]` is the true generating model, but similar models

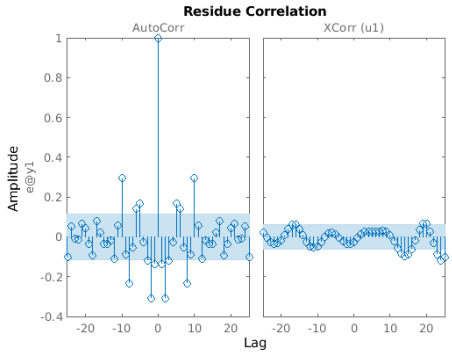


(a)

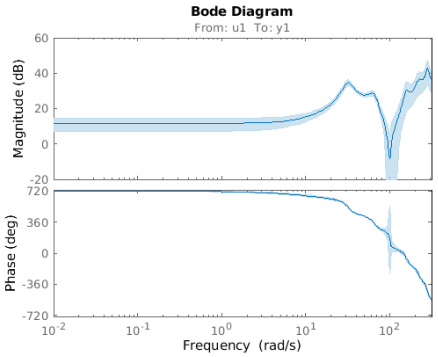


(b)

Figure 2: Welch power spectra in problem 3.



(a)

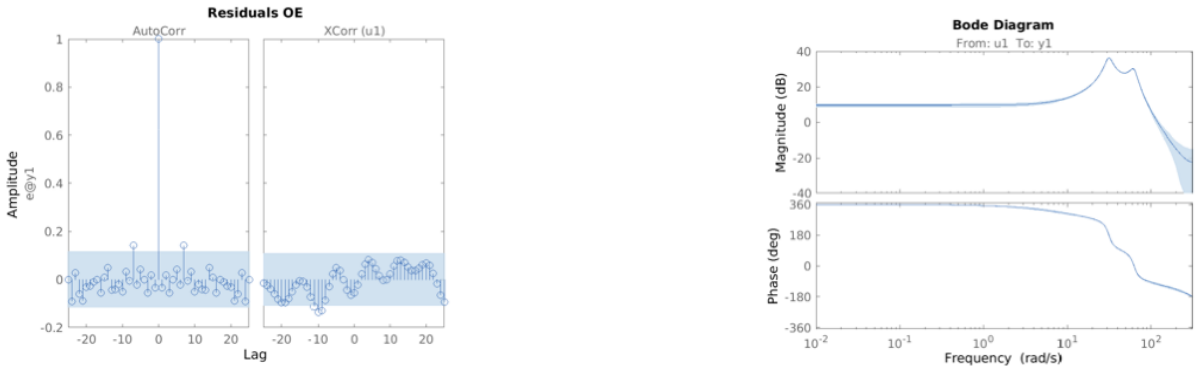


(b)

Figure 3: Residue and bode plot for ARX model.

such as `oe[262]` and `oe[341]` also give similarly good results). The residuals are now considerably smaller (Figure 4a). The bode plot (Figure 4b) also looks improved, with smaller gain for high frequencies. The result of Matlab's `present()` command (Figure 5) on the fitted model shows that no parameters seem insignificant. (We can notice though that the F-polynomial coefficients are considerably more accurately estimated than the B-polynomial.)

Attempting to find more complex models (ARMAX or Box-Jenkins) that fit the data yields a very marginal increase in performance at the cost of many more parameters, likely only leading to overfitting.



(a) (b)

Figure 4: Residue and bode plot for OE model.

```
model_oe =
Discrete-time OE model: y(t) = [B(z)/F(z)]u(t) + e(t)
  B(z) = -0.1456 (+/- 0.05167) z^-1 - 0.5556 (+/- 0.1489) z^-2 + 0.6877 (+/- 0.1595) z^-3 + 0.115 (+/- 0.06293) z^-4

  F(z) = 1 - 3.368 (+/- 0.002054) z^-1 + 4.63 (+/- 0.005317) z^-2 - 3.056 (+/- 0.005077) z^-3 + 0.8279 (+/- 0.001786) z^-4

Sample time: 0.01 seconds

Parameterization:
   Polynomial orders:   nb=4   nf=4   nk=1
   Number of free coefficients: 8
   Use "polydata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Termination condition: Near (local) minimum, (norm(g) < tol)..
Number of iterations: 9, Number of function evaluations: 30

Estimated using OE on time domain data "ztrain".
Fit to estimation data: 89.24%
FPE: 8.025, MSE: 7.772
More information in model's "Report" property.
```

Figure 5: Output from Matlab's `present()` command on the fitted `oe[441]` model.

4 Causal Inference and DAGs

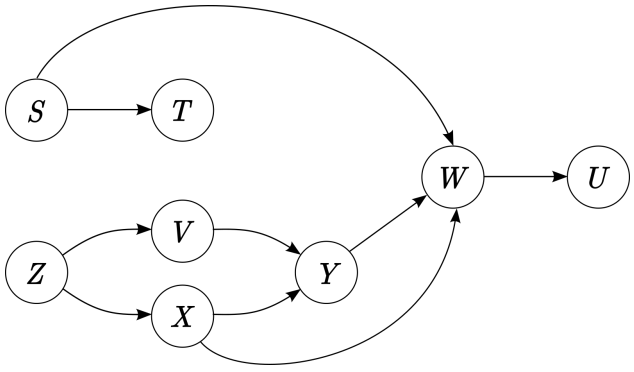


Figure 6: Directed Acyclic Graph (DAG) describing the causality relations in problem 4.

We can use the DAG to check the statements in the problem, using the concept of d-separation, see lecture on causal inference.

Since the model is linear and Gaussian, the code can also be used to check these statements about independence and conditionally independence. For instance, if X is conditionally independent of Y given S then a numerical linear regression of the form

$$X = \alpha_1 Y + \alpha_2 S + \text{ noise}$$

should give a value α_1 close to 0 and statistically significant non-zero coefficient would indicate the opposite. The code can therefore be used to verify your conclusions from the graph, so you can make sure you did not make a mistake.

a) The graph can be seen in Figure 6.

b) False. T and W are not independent due to the confounder S.

c) True. Conditioning on X and V blocks all paths between Y and Z. Therefore they are independent given X and V.

d) False. U is the descendant of W, which forms a fork together with Y and S. Conditioning on U therefore opend the path across W, making S and Y dependent.

e) False. T and U are both influenced by S (U via W).

f) True. X and S are connected only via the fork on W, which is closed since we do not condition on W or its descendants.

g) True. Conditioning on W blocks the path between X and U.

h) True. Conditioning on the confounder Z renders V and X independent.

i) False. If we take the scenario in h) but add the conditioning on U we open the path across the fork on Y, since U is a descendant of Y.

As an example of how to use the code, in subproblem b, the following output from the linear regression $T = \alpha_1 W + \text{ noise}$ indicates a statistically significant non-zero coefficient α_1, indicating that T and W are NOT independent.

```
results3 = smf.ols('T ~ W - 1', data=dat1).fit()
print(results3.summary())
```

```
                                 OLS Regression Results
=======================================================================================
Dep. Variable:                      T   R-squared (uncentered):                   0.004
Model:                            OLS   Adj. R-squared (uncentered):              0.003
Method:                 Least Squares   F-statistic:                              35.28
Date:                Sat, 23 Apr 2022   Prob (F-statistic):                    2.94e-09
Time:                        08:59:30   Log-Likelihood:                         -17540.
No. Observations:               10000   AIC:                                  3.508e+04
Df Residuals:                    9999   BIC:                                  3.509e+04
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
W              0.0125      0.002      5.940      0.000       0.008       0.017
```

Figure 7: Simulation output for subproblem b, shows that the coefficient 0.0125 ± 0.002 is nonzero with high statistical significance since $P > |t| = 0.000$ (outside 5.94 standard deviations)

5 Estimation theory

a) The probability of the obtained outcome is given by

$$p(y,\theta) = (\theta^2)^{n_1}(2\theta(1-\theta))^{n_2}((1-\theta)^2)^{n_3} = 2^{n_2}\theta^{2n_1+n_2}(1-\theta)^{n_2+2n_3}$$

The log-likelihood function is

$$l(y,\theta) = \log(p(y,\theta)) = (2n_1+n_2)\log(\theta) + (n_2+2n_3)\log(1-\theta) + \text{const.}$$

We calculate the first and second derivatives of $l(y,\theta)$ with respect to the parameter θ

$$\frac{\partial l}{\partial \theta} = \frac{2n_1+n_2}{\theta} - \frac{n_2+2n_3}{1-\theta}$$

$$\frac{\partial^2 l}{\partial \theta^2} = -\frac{2n_1+n_2}{\theta^2} - \frac{n_2+2n_3}{(1-\theta)^2}$$

In order to find the Fisher information matrix $\mathcal{I}(\theta)$ we must take the expected value of the second derivative calculated above, using $E(n_1) = \theta^2$, etc.

$$\mathbb{E}\left[\frac{\partial^2 l}{\partial \theta^2}\right] = -\frac{2N\theta^2 + 2N\theta(1-\theta)}{\theta^2} - \frac{2N\theta(1-\theta) + 2N(1-\theta)^2}{(1-\theta)^2} = -\frac{2N}{\theta} - \frac{2N}{1-\theta} = -\frac{2N}{\theta(1-\theta)}$$

The Cramer Rao Lower Bound (CRLB) is then given by

$$(\mathcal{I}(\theta))^{-1} = (-\mathbb{E}\left[\frac{\partial^2 l}{\partial \theta^2}\right])^{-1} = \frac{\theta(1-\theta)}{2N}$$

This seems realistic, since we expect the variance to decrease as we get more data (larger N).

Remark With some experience you can recognize the result as the error variance of the estimation of the parameter θ for a Bernoulli distribution, using $2N$ samples. Which you then quickly also recognize is the situation we have in this problem.

b) Inspired by the optimality result on slide 18 of lecture 13-14 we rewrite the function $\frac{\partial l}{\partial \theta}$ as

$$\begin{aligned}\frac{\partial l}{\partial \theta} &= \frac{2n_1+n_2}{\theta} - \frac{n_2+2n_3}{1-\theta} = \dots \\ &= \frac{2N}{\theta(1-\theta)}\left[\frac{2n_1+n_2}{2N} - \theta\right],\end{aligned}$$

where we have used $n_3 = N - n_1 - n_2$. Since this is of the form

$$\frac{\partial l(y,\theta)}{\partial \theta} = \mathcal{I}(\theta)\left[\hat{\theta}(y) - \theta\right]$$

if we use $\mathcal{I}(\theta) = \frac{2N}{\theta(1-\theta)}$ and $\hat{\theta}(y) = \frac{2n_1+n_2}{2N}$, this shows that this bias-free estimate achieves the CRLB and is optimal.

Remark In general it is not enough to derive the expression for the maximum-likelihood estimator and claim that this achieves the CRLB bound. The general result on optimality of ML-estimators is an asymptotic result for the case when the number of data N approaches infinity.

However the estimate $\hat{\theta}(y) = \frac{2n_1+n_2}{2N}$ derived above is easily seen to be the ML-estimate in this situation. The easiest way to see this, is to realize that we have the standard situation for $2N$ Bernoulli distributed variables with parameter θ.