TÉCNICO LISBOA

Data Analitics for Smart Grids

MEEC

Lab 1 - Phase Identification

Authors:

Rodrigo Contreiras (90183) rodrigo.contreiras@tecnico.ulisboa.pt
Gonçalo Aniceto (96218) goncalo.aniceto@tecnico.ulisboa.pt

2022/2023 – 2nd Semester, P3

Contents

1 Problem description

The increased deployment of distributed energy generation, as well as the integration of new, large electric loads such as electric vehicles and heat pumps, puts the correct and reliable operation of low voltage distribution systems under strain. Active management solutions, which require distribution system models that include the phase connectivity of all consumers in the network, are proposed in the literature to address potential problems. Even though grid maintenance phase sequence initiatives exist, information on phase connectivity of single-phase connected customers is often erroneous or missing. Therefore it becomes increasingly important to correctly identify phase connectivity, a problem consisting of classifying each client with one out of three possible labels a, b, or c.

2 Objectives

The aim of this assignment is to address the phase identification problem following a Multivariate Regression approach. Active power consumption data will be available for each phase at the substation and for each client at the smart meters. This data is to be correlated, correctly assigning clients to phases a, b, or c. An algorithm will be proposed and tested on various scenarios where losses are taken into account, similar client consumption occurs, and three-phase clients are present or black-outs are present in smart-meter data.

3 Algorithm

Admitting a single bus representation, the per-phase energy readings, y_p, can be approximated by a summation function of the corresponding customer i measurements, x_i, if the correspondence is established by assigning to i a binary coefficient β_i^p:

$$y_p(k) \approx \sum_{i=1}^{N} \beta_i^p x_i(k)$$

This summation equality can be decomposed into three per-phase inner product equalities, $y_p \approx \langle \beta_i^p, x_i \rangle$ and be written in matrix form as a multivariate regression equation:

$$Y = BX + \epsilon$$

Now the labelling problem can be decomposed as two smaller sub-problems. First, ordinary least squares shall be used to estimate the of conditional expectation, $E(Y|X = x)$, searching for correlations between Y and X. With this linear regression technique positive and negative errors won't cancel out and bigger errors will be much more penalized than smaller ones. The solution is given by:

$$B = (X^T X)^{-1} X^T Y$$

Our ability to find the correlations will depend on the noise level, σ^2, and on the measurement's conditioning, $cond(X)$. Notice that the high covariances of B will lead to high error probability:

$$Var(B) = (X^T X)^{-1} \sigma^2$$

To label the costumers, the binary coefficient corresponding to the largest component of the estimates vector is assigned to 1, whereas the smaller ones are assigned to 0. The complete algorithm is as follows:

Algorithm 1 Phase Identification

Require: $M \geq N$
  $min_B\ (Y - BX)^2$
  for $i \leq N \wedge p \in \{a, b, c\}$ **do**
    if $arg\ max_{a,b,c} B(i,:) = p$ **then**
      $\beta_i^p = 1$
    else
      $\beta_i^p = 0$

4 Phase Identification Problems

4.1 Initial Problem

In this section a simple case will be analysed where four clients are connected to a metered MV/LV transformer. The phase identification task at hand is easy enough one can do it by observing. Nevertheless, solving using the proposed algorithm will bring valuable insight.

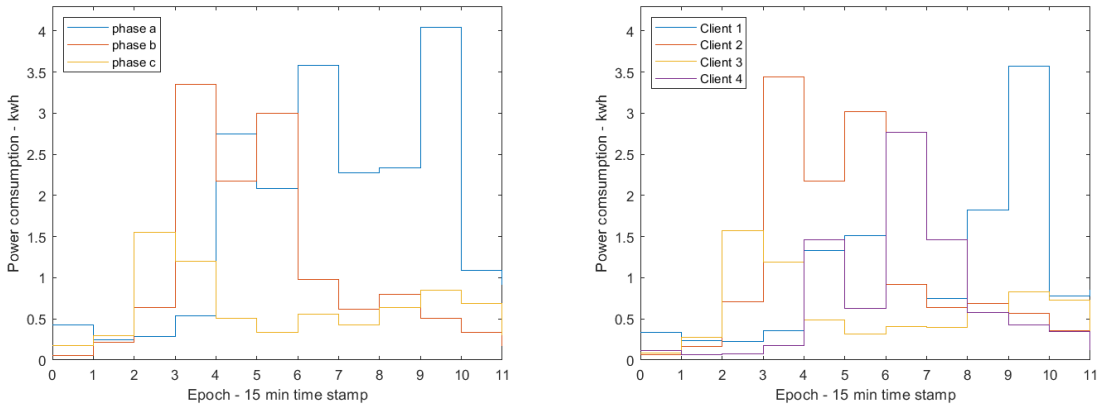

Figure 1: Chronological representation of phase A, B, C measurement totals and the corresponding four customer readings, for which phase labels (a, b, c) need to be assigned.

Considering the smart-meter readings, X, as the same as the ones provided [1], the per-phase readings, Y can be obtained by randomly assigning clients to phases and adding error, $\epsilon \sim \mathcal{N}(0, \sigma^2)$, ie $Y = BX + \epsilon$. The solution considering a moderate error, $\sigma = 0.25$, can be correctly achieved mapping regression results:

$$B = (X^T X)^{-1} X^T Y = \begin{bmatrix} 1.0238 & 0.0073 & -0.0392 \\ 0.0087 & 0.9893 & 0.0315 \\ -0.0159 & 0.0349 & 0.9998 \\ 0.9909 & 0.0073 & 0.0074 \end{bmatrix} \longrightarrow \beta = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

By increasing the standard deviation, σ, the covariances of the variance–covariance matrix of the least squares parameter estimates will also increase:

$$Var(B) = (X^T X)^{-1} \sigma^2$$

The correlations between Y and X will be harder to identify and it is expected for the number of committed errors to increase with σ. This can be seen in the graphs where the moving-average number of errors and error probability are plotted against the variance, with steps of $\Delta\sigma^2 = 0.04$.

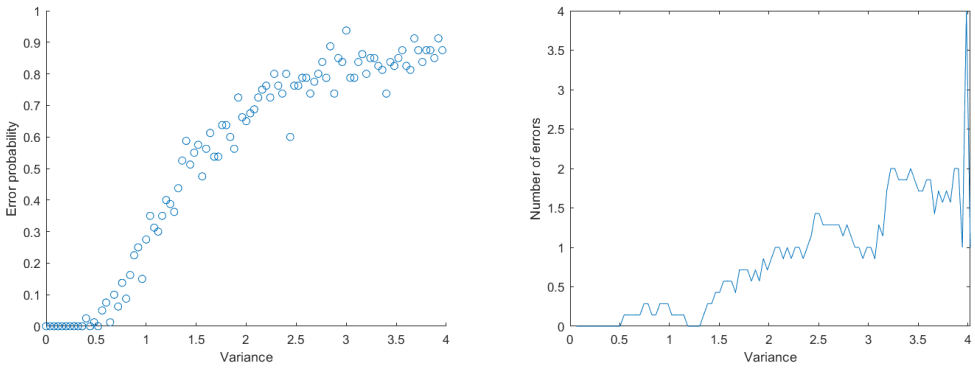

Figure 2: Error probability and average number of errors in function of the noise level, σ^2.

If the number of clients, N, is increased to twelve, it will become computationally harder as the complexity grows with 3^N. Visually, it is no longer solvable by inspection and, if the sample size, M, doesn't increase, the covariances of B will increase through $(X^T X)^{-1}$, making the problem harder to solve. Note that, without the presence of noise ie, $\sigma = 0$, unequivocal correlations are found as long as $N > M$ since the problem is determined and $(X^T X)$ isn't singular.

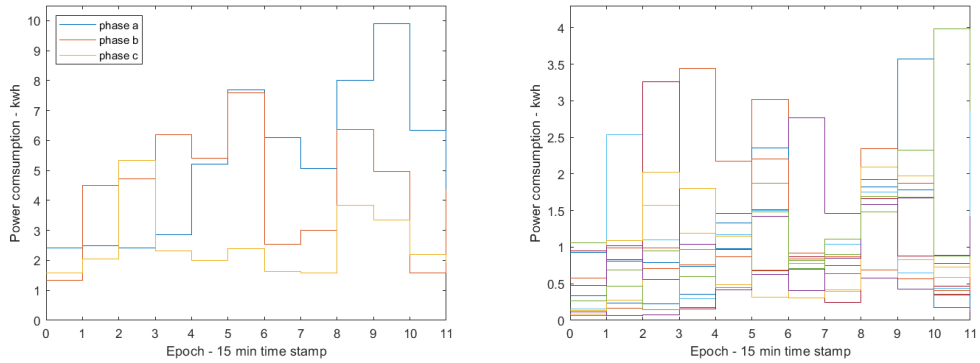

Figure 3: Chronological representation of phase A, B, C measurement totals and the corresponding twelve customer readings, for which phase labels (a, b, c) need to be assigned.

The solution considering a moderate error, $\sigma = 0.25$ and a relatively high number of customers, $N = M = 12$ can still be correctly achieved mapping regression results:

$$B = (X^T X)^{-1} X^T Y = \begin{bmatrix} 1.0449 & -0.1777 & -0.2873 \\ -0.0456 & 1.2018 & 0.3001 \\ 0.0629 & -0.6074 & -0.3328 \\ 1.0629 & -0.2220 & -0.1554 \\ 0.9619 & -0.0021 & 0.0615 \\ -0.1065 & 1.1986 & 0.1964 \\ -0.0569 & 0.5814 & 1.8011 \\ -0.0475 & 1.0196 & -0.0430 \\ 1.1718 & -0.8188 & -1.1906 \\ -0.0063 & 0.8930 & 0.1777 \\ 0.4215 & -0.6081 & 0.3248 \\ 0.5972 & 1.3740 & 1.7737 \end{bmatrix} \longrightarrow \beta = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

By increasing the standard deviation, σ, the correlations between Y and X are expected to be harder to identify. This can be seen in the graphs where the moving-average number of errors and error probability are plotted against the variance, with steps of $\Delta\sigma^2 = 0.04$. Note that the probability of error grows much faster with noise, σ^2 and for the measurements proposed, X, the error probability takes upon unreasonable values for $\sigma^2 > 0.08$ corresponding to a standard deviation of $\sigma = 0.2828$.

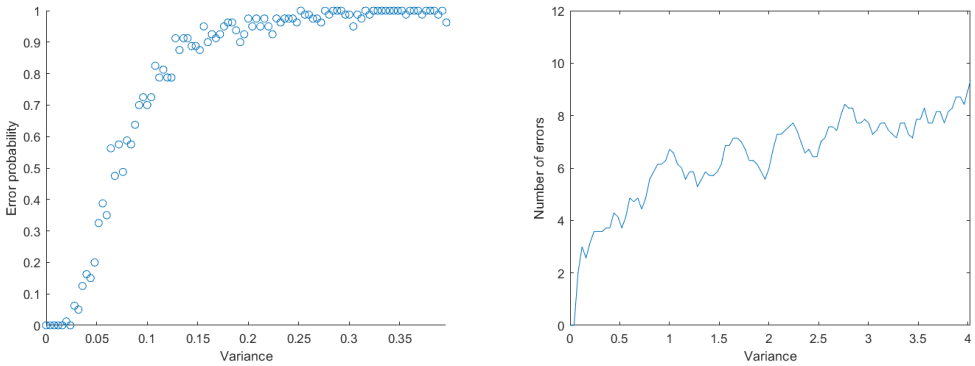

Figure 4: Error probability and average number of errors in function of the noise level, σ^2.

When the error probabilities of both cases, $N = 4$ and $N = 12$, are set side by side it becomes clear that the error grows much faster in the second case. One can therefore conclude that the problem difficulty grows with dimentionality and noise.

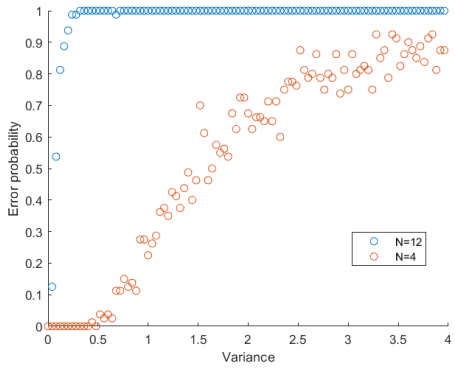

Figure 5: Error probability in function of the noise level, σ^2.

4.2 Losses

Electrical systems are inevitably restricted by losses. According to recent assessments [2], [3], energy losses in distribution networks (low voltage and medium voltage) in European nations range from 2% to 13.5% of total energy input. In this section those losses will be modelled and the effect on proposed phase identification algorithm will be assessed.

Losses affect readings at the substation resulting in a non-null mean value, ie $Y = BX + \epsilon$, with $\epsilon \sim \mathcal{N}(\mu, \sigma^2)$. Since the cable losses depend non-linearly on demanded power, the mean value of per-phase error in a give time epoch, $\mu_p(k)$, shall be a function of per-phase consumption. Considering the given data, X, the typical power loss can be approximated to obtained by a linear function:

$$\mu_p(k) = 0.1354 y_p(k) - 0.01223$$

Note that, by considering that errors have non-null mean values, the Gauss-Markov theorem no longer guarantees that OLS produces better estimates than all other linear model estimation methods. As a result, coefficient estimates B will be biased. Nevertheless, let's consider the problem of labelling $N = 6$ customer with $M = 12$ readings:

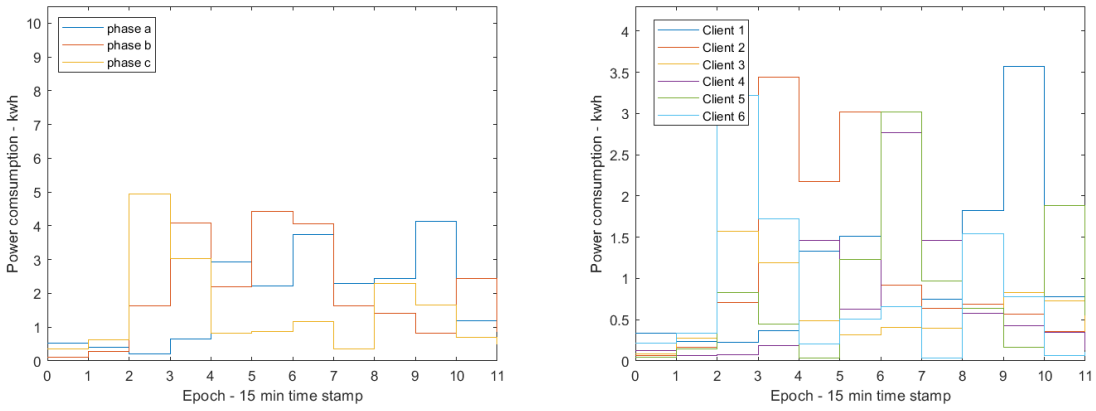

Figure 6: Chronological representation of phase A, B, C measurement totals and the corresponding six customer readings, for which phase labels (a, b, c) need to be assigned.

By increasing the standard deviation, σ, the correlations between Y and X become harder to estimate and the error probability will increase. Even though 13,5% is a very high number from a losses point of view, this value isn't high enough to influence the phase identification method proposed. In fact, no significant performance loss was recorded and the error probability grew similarly, independent of the error mean value, μ.

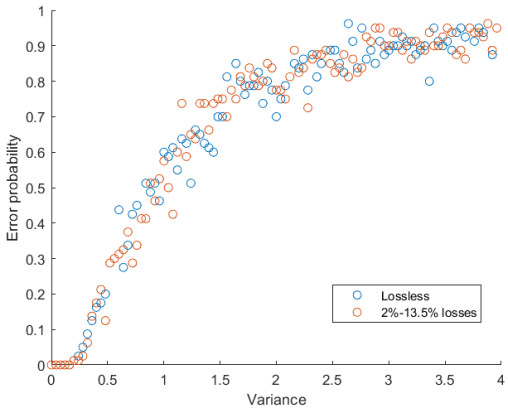

Figure 7: Error probability in function of the noise level, σ^2.

4.3 Similar Client Consumption

At first glance, we can analyse this problem by focusing solely on the algebraic side. Taking in consideration the mathematical formula of the B calculation, we see that it involves the inverse of the product matrix $X^T X$. Well, in order for it to be invertible it needs to have full rank.

If a matrix's columns and rows are linearly independent, the matrix is said to have full rank. This implies that none of the independent variables in a regression analysis are linear combinations of the other independent variables.

Also, if the number of independent variables in the regression model is greater than the number of observations, then $X^T X$ will not have full rank and the inverse of the product matrix will not exist. In this case, alternative methods such as ridge regression or principal component regression may be used to estimate the regression coefficients.

To use this method, we used again the previous 4 costumers. We started by making two of them identical and created a code that would increase the difference between them, in order for us to understand how the closeness between two different consumers influences the correct prediction of the phases. The results are shown in the following figure:

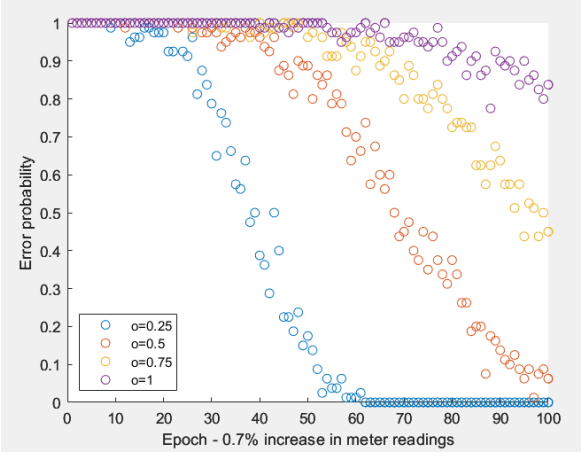

Figure 8: Error probability in function of the distance between two costumers and for 4 different noise level, σ^2.

As we can see, it is clearly visible that the error probability is 100% when the costumers have equal or similar consumption and the bigger the difference between them, the lower the error probability. Something that it is also visible, is that the noise level as an effect on the error probability as well. With the lowest noise level (represented by the blue dots on the figure), the error probability fades away much faster than the rest, with the worst being the one with the highest noise.

4.4 Three-phase Clients

Several households are installing EV chargers and heat pumps as part of decarbonization projects. This causes an increase in power consumption, requiring a three-phase connection to the grid. It should be noted that this does not imply that the loads will be balanced. As the meter readings are for total power usage, phase identification becomes more challenging. In this section three-phase clients will be modelled and the effect on proposed phase identification algorithm will be assessed.

Per-phase power usage data isn't recorded by the smart meter, but can be approximated to a third of the total consumption, $x_i^p \approx \frac{x_i^{total}}{3}$. Note that this approximation will lead X being ill-conditioned:

Mono-Phase	Cond(X)
N=6 M=12	11.4716
N=12 M=12	230.656
Three-Phase	Cond(X)
N=6 M=12	1.1127e+32
N=12 M=12	4.5569e+17

If the problem is ill-conditioned, a slight change in the inputs (smart meter readings) results in a large change in the outcome or dependent variable. This indicates that finding the proper solution/answer to the problem becomes difficult.

After running multiple scenarios, we realized that in this way the three-phase consumers would create added noise to the substation measurement matrix. This happens because if the power consumption isn't balanced, among the three phases, by subtracting a third of the power from the loads, we are

going to have the difference between them added to the following calculations. So, the more unbalanced the power distribution is and the higher is the power from the three phase consumers, the more errors will appear. This translates to the same case off when we have higher noise level. In the worst cases scenarios, this will lead to very high covariances on the variance-covariance matrix from the least square parameters estimate matrix.

$$cov(B)_{big-three-phase} = \begin{bmatrix} 14.4358 & -5.6387 & -8.4481 \\ -5.6387 & 2.2425 & 3.2595 \\ -8.4481 & 3.2595 & 4.9850 \end{bmatrix}$$

$$cov(B)_{small-three-phase} = \begin{bmatrix} 1.1073 & -0.2856 & -0.8390 \\ -0.2856 & 0.1734 & 0.1102 \\ -0.8390 & 0.1102 & 0.7501 \end{bmatrix}$$

$$cov(B)_{mono-phase} = \begin{bmatrix} 0.3506 & -0.1830 & -0.1608 \\ -0.1830 & 0.2468 & -0.0718 \\ -0.1608 & -0.0718 & 0.2360 \end{bmatrix}$$

As it is visible from the matrices above, when we use a noise level that leads to zero errors when identifying the phases when we have only mono-phase costumers(last matrix), the covariance represented by the diagonal values are very low. When we add the three phase consumers, we can see that if the power distribution isn't balanced, the covariance can still be small enough to lead to minimal errors and to a correct phase identification, as shown by the second matrix. However, the more power consumption three phase consumers have, the more difficult it is for a proper identification. The first matrix which as big covariances, leads to a 100% error rate.

4.5 Missing smart-meter data

Smart meters have powerful measurement and computing hardware, software, calibration, and communication capabilities. Although they are meant to perform functions and store and communicate data in accordance with established standards in order to be interoperable within a smart grid architecture, they are not fail-proof, and some reading is routinely lost. In this section missing data will be modelled and the effect on proposed phase identification algorithm will be assessed.

When the readings are missing the assigned value will be $x_{missing} = 0$. This will difficult the problem as the inaccurate data makes correlations between X and Y harder to estimate. Data preprocessing can be done, removing the erroneous reading or coming up with a estimate for this value. Assuming no losses, this value can be obtained by doing an energy balance for the time period, k. The modified algorithm will be :

Let's consider the problem of labelling $N = 6$ customer with $M = 12$ readings where 10% of the total readings are missing:

By increasing the noise level the correlations between Y and X become harder to estimate and the error probability will increase. It is worth noting that, in the presence of missing readings, the likelihood of mistakenly labeling a customer increases at a comparable rate, but the error probability ceases to be null at a lower value of variance, σ^2. Preprocessing the raw data resulted in a considerable performance gain, with a customer being 31,25% less likely to be mislabeled when $\sigma^2 = 0,6$.

Algorithm 2 Modified Phase Identification

Require: $M \geq N$
  if $x_i(k) = 0$ **then**
    $x_i(k) = \sum y_p(k) - \sum_{j \neq i} x_j(k)$
  end if
  $min_B\ (Y - BX)^2$
  for $i \leq N \wedge p \in \{a, b, c\}$ **do**
    if $arg\ max_{a,b,c} B(i,:) = p$ **then**
      $\beta_i^p = 1$
    else
      $\beta_i^p = 0$
    end if

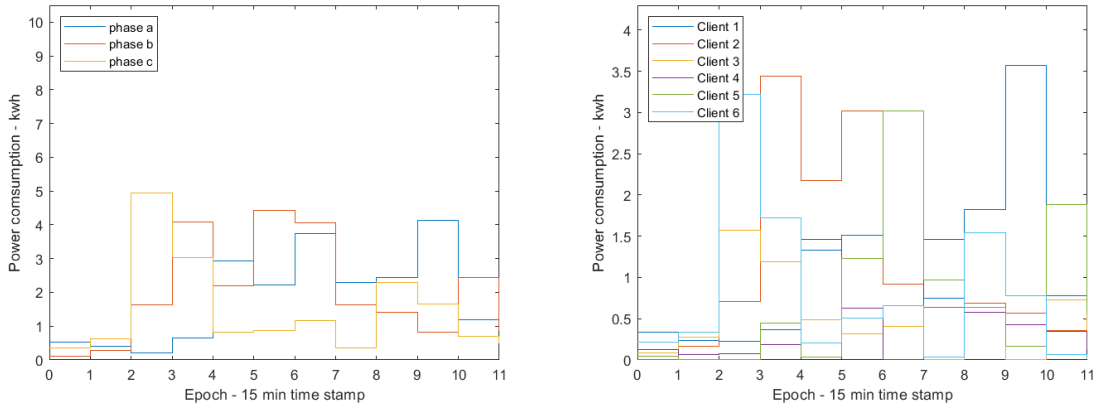

Figure 9: Chronological representation of phase A, B, C measurement totals and the corresponding six customer (incomplete) readings, for which phase labels (a, b, c) need to be assigned.

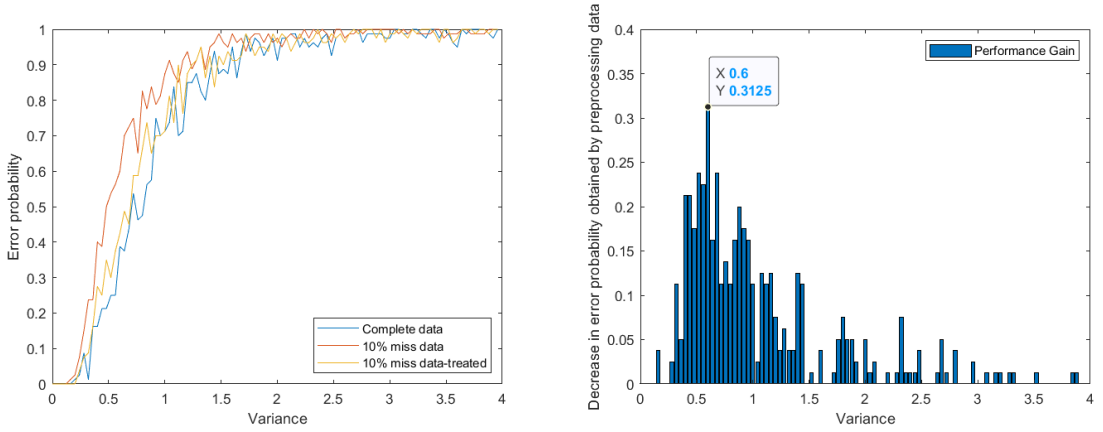

Figure 10: Error probability and performance gain in function of the noise level, σ^2.

Note that, when two or more readings are missing on the same time frame, the result of the power balance shall be distributed through the consumers. When high amounts of data points are available, one can choose discard the missing ones.

5 Conclusion

The proposed phase identification algorithm was able to assign metered active power consumption to the correct phase in high noise and high client (in comparison to data point) situations. It was accessed that the algorithm was resilient to the expectable values of losses. Although there was an underperfomance when there were missing data-points, an alternative procedure was proposed and this problem was corrected.

When we have costumers with similar power consumptions, it becomes more challenging to correctly identify the phases. There needs to be a difference between the consumers, which needs to be bigger when exposed to higher meter readings noise. With real data readings, it is easier to separate the consumers, because they are taken during long periods of time, which will in most cases lead to more noticeable differences.

Lastly, three phase consumers seem to add extra difficulty when it comes to this kind of problems. Although it is still possible to get to correct answers, the more power consumption and the more unbalanced is the power distribution across the phases, the more errors the predictions are going to have. But something that could be assumed, is that across multiple three phase consumers, their added power consumption might be close to being balanced, which in that case wouldn't affect the predictions that much and the error probability would be mostly affected by the metering measurments noise level.

References

1)**Lecture Notes: Chap 2.1 Phase Id**

2)**CEER Report on Power Losses (Ref C17-EQS-80-03), Council of European Energy Regulators; 2017.**

3)**Identifying energy efficiency improvements and saving potential in energy networks, including analysis of the value of demand response, DG Energy, European Commission; 2015.**

Appendix

.1 Phase Identification

```
%Lab1 DASG
clear all;
format compact;

%Comsumption Data (Given)
 X = [0.332 0.064 0.084 0.12
 0.236 0.164 0.276 0.064
 0.224 0.708 1.572 0.072
 0.36  3.44  1.188 0.18
 1.332 2.176 0.484 1.464
 1.516 3.02  0.316 0.624
 0.92  0.916 0.404 2.772
 0.752 0.64  0.396 1.464
 1.828 0.684 0.576 0.576
 3.568 0.564 0.828 0.428
 0.78  0.356 0.728 0.348
 0.856 0.22  0.308 0.12];

 %Atribute comsumers to phase: abca
 beta_orig = [1 0 0
              0 1 0
              0 0 1
              1 0 0];


 %Consumers aggregation by phase and noise inclusion
 Y = zeros(3,1);

 for k = 1:12
    Y(k,1) = X(k,1) + X(k,4) + normrnd(0,0.25^2);
    Y(k,2) = X(k,2) + normrnd(0,0.25^2);
    Y(k,3) = X(k,3) + normrnd(0,0.25^2);
 end

 %Multivariate Regression
 B = (X.'*X)^-1*X.'*Y;

 %Mapping
 beta = zeros(4,3);

 for k = 1:4
    if B(k,1)>B(k,2)
        if B(k,1)>B(k,3)
            beta(k,1) = 1;
        else
```

```
            beta(k,1) = 1;
        end
    elseif B(k,2)>B(k,3)
            beta(k,2) = 1;
        else
            beta(k,3) = 1;
    end
 end


%Plot
figure()
ep=0:11;
stairs(ep,Y)
xlabel("Epoch - 15 min time stamp")
ylabel("Power comsumption - kwh")
axis([0 11 0 4.3])
legend({'phase a','phase b','phase c'},'Location','northwest')
```

.2 Error Probability

```
    %Comsumption Data (Given)
 X =    [0.332 0.064 0.084 0.123 0.042 0.221
         0.236 0.164 0.276 0.064 0.142 0.333
         0.224 0.708 1.572 0.072 0.829 3.221
         0.365 3.44  1.188 0.188 0.452 1.721
         1.332 2.176 0.484 1.464 0.034 0.203
         1.516 3.023 0.316 0.624 1.235 0.508
         0.92  0.916 0.404 2.772 3.018 0.654
         0.752 0.64  0.396 1.464 0.965 0.032
         1.828 0.684 0.576 0.576 0.638 1.543
         3.568 0.564 0.828 0.428 0.165 0.777
         0.78  0.356 0.728 0.348 1.879 0.067
         0.856 0.222 0.308 0.121 0.543 0.112];




 %Atribute comsumers to phase: abca
  beta_orig = [1 0 0
               0 1 0
               0 0 1
               1 0 0
               0 1 0
               0 0 1];

 %Consumers aggregation by phase and noise inclusion
 Y = zeros(12,1);
 var = 0.25^2;
```

```
for kkk = 1:100

        var = 0 +(kkk-1)*0.04;
        err_aux = 0;

        for kk = 1:80

             for k = 1:12
                Y(k,1) = X(k,1) + X(k,4);
                Y(k,2) = X(k,2) + X(k,5);
                Y(k,3) = X(k,3) + X(k,6);
             end

             for k = 1:12
                mu(k,1) = 0.1354*Y(k,1) - 0.01223;
                mu(k,2) = 0.1354*Y(k,2) - 0.01223;
                mu(k,3) = 0.1354*Y(k,3) - 0.01223;
             end

             for k = 1:12
                Y(k,1) = Y(k,1) + normrnd(mu(k,1),var);
                Y(k,2) = Y(k,2) + normrnd(mu(k,2),var);
                Y(k,3) = Y(k,3) + normrnd(mu(k,3),var);
             end


             %Multivariate Regression
             B = (X.'*X)^-1*X.'*Y;

             %Mapping
             beta = zeros(6,3);

             for k = 1:6
                if B(k,1)>B(k,2)
                    if B(k,1)>B(k,3)
                        beta(k,1) = 1;
                    else
                        beta(k,3) = 1;
                    end
                elseif B(k,2)>B(k,3)
                        beta(k,2) = 1;
                    else
                        beta(k,3) = 1;
                end
             end

            %Error count
```

```
            err = 0;

            for k = 1:6
                if beta(k,1) ~= beta_orig(k,1) || beta(k,2) ~= beta_orig(k,2) || beta(k,3) ~=
                    err = err + 1;
                end
            end

            if(err>0)
            err_aux = err_aux+1;
            end
        end

        err_record(kkk) = err_aux/80;
        var_record(kkk) = var;
end


Y = zeros(12,1);

for kkk = 1:100

        var = 0 +(kkk-1)*0.04;
        err_aux = 0;

        for kk = 1:80

             for k = 1:12
                Y(k,1) = X(k,1) + X(k,4) + normrnd(0,var);
                Y(k,2) = X(k,2) + X(k,5) + normrnd(0,var);
                Y(k,3) = X(k,3) + X(k,6) + normrnd(0,var);
             end


             %Multivariate Regression
             B = (X.'*X)^-1*X.'*Y;

             %Mapping
             beta = zeros(6,3);

             for k = 1:6
                if B(k,1)>B(k,2)
                    if B(k,1)>B(k,3)
                        beta(k,1) = 1;
                    else
                        beta(k,3) = 1;
                    end
                elseif B(k,2)>B(k,3)
                        beta(k,2) = 1;
```

```
                    else
                        beta(k,3) = 1;
                end
             end

            %Error count
            err = 0;

            for k = 1:6
                if beta(k,1) ~= beta_orig(k,1) || beta(k,2) ~= beta_orig(k,2) || beta(k,3) ~=
                    err = err + 1;
                end
            end

            if(err>0)
            err_aux = err_aux+1;
            end
        end

        err_record1(kkk) = err_aux/80;

 end

X2 =    [0.332 0.000 0.084 0.123 0.042 0.221
         0.236 0.164 0.276 0.064 0.000 0.333
         0.224 0.708 1.572 0.072 0.000 3.221
         0.365 3.44  1.188 0.188 0.452 1.721
         1.332 2.176 0.484 1.464 0.034 0.203
         1.516 3.023 0.316 0.624 1.235 0.508
         0.92  0.916 0.404 0.000 3.018 0.654
         0.752 0.64  0.000 1.464 0.965 0.032
         1.828 0.684 0.576 0.576 0.638 1.543
         3.568 0.564 0.000 0.428 0.165 0.777
         0.78  0.356 0.728 0.348 1.879 0.067
         0.856 0.222 0.308 0.121 0.543 0.000];




for kkk = 1:100

        var = 0 +(kkk-1)*0.04;
        err_aux = 0;

         for kk = 1:80
              for k = 1:12
                Y(k,1) = X(k,1) + X(k,4) + normrnd(0,var);
                Y(k,2) = X(k,2) + X(k,5) + normrnd(0,var);
```

```
            Y(k,3) = X(k,3) + X(k,6) + normrnd(0,var);
        end


        %Multivariate Regression
        B = (X2.'*X2)^-1*X2.'*Y;

        %Mapping
        beta = zeros(6,3);

        for k = 1:6
            if B(k,1)>B(k,2)
                if B(k,1)>B(k,3)
                    beta(k,1) = 1;
                else
                    beta(k,3) = 1;
                end
            elseif B(k,2)>B(k,3)
                    beta(k,2) = 1;
                else
                    beta(k,3) = 1;
            end
        end

       %Error count
       err = 0;

       for k = 1:6
           if beta(k,1) ~= beta_orig(k,1) || beta(k,2) ~= beta_orig(k,2) || beta(k,3) ~=
               err = err + 1;
           end
       end

       if(err>0)
       err_aux = err_aux+1;
       end
    end

    err_record2(kkk) = err_aux/80;

end


for kkk = 1:100

    var = 0 +(kkk-1)*0.04;
    err_aux = 0;

     for kk = 1:80
```

```
 for k = 1:12
   Y(k,1) = X(k,1) + X(k,4) + normrnd(0,var);
   Y(k,2) = X(k,2) + X(k,5) + normrnd(0,var);
   Y(k,3) = X(k,3) + X(k,6) + normrnd(0,var);
end

%data treatment-Energy Balance
   X3 = X2;

   X3(1,2) = Y(1,1) + Y(1,2) + Y(1,3) - X2(1,1) - X2(1,3) - X2(1,4) - X2(1,5) - X
   X3(2,5) = Y(2,1) + Y(2,2) + Y(2,3) - X2(2,1) - X2(2,2) - X2(2,3) - X2(2,4) - X
   X3(3,5) = Y(3,1) + Y(3,2) + Y(3,3) - X2(3,1) - X2(3,2) - X2(3,3) - X2(3,4) - X
   X3(7,4) = Y(7,1) + Y(7,2) + Y(7,3) - X2(7,1) - X2(7,2) - X2(7,3) - X2(7,5) - X
   X3(8,3) = Y(8,1) + Y(8,2) + Y(8,3) - X2(8,1) - X2(8,2) - X2(8,4) - X2(8,5) - X
   X3(10,3) = Y(10,1) + Y(10,2) + Y(10,3) - X2(10,1) - X2(10,2) - X2(10,4) - X2(1
   X3(12,6) = Y(12,1) + Y(12,2) + Y(12,3) - X2(12,1) - X2(12,2) - X2(12,3) - X2(1

%Multivariate Regression
B = (X3.'*X3)^-1*X3.'*Y;

%Mapping
beta = zeros(6,3);

for k = 1:6
   if B(k,1)>B(k,2)
       if B(k,1)>B(k,3)
           beta(k,1) = 1;
       else
           beta(k,3) = 1;
       end
   elseif B(k,2)>B(k,3)
           beta(k,2) = 1;
       else
           beta(k,3) = 1;
   end
end

%Error count
err = 0;

for k = 1:6
    if beta(k,1) ~= beta_orig(k,1) || beta(k,2) ~= beta_orig(k,2) || beta(k,3) ~=
        err = err + 1;
    end
end

if(err>0)
err_aux = err_aux+1;
```

```
            end
        end

        err_record3(kkk) = err_aux/80;

 end


for k =1:100
compare(1,k) = err_record1(k)
compare(2,k) = err_record3(k)
performacegain(k)= (err_record2(k) - err_record3(k));
end


%Plot probability info
figure()
bar(var_record,performacegain)
xlabel("Variance")
ylabel("Decrease in error probability obtained by preprocessing data")
axis([0 4 0 0.4])
legend({'Performance Gain',},'Location','northeast')
```