Temperature prediction for permanent magnet synchronous motors using linear and multiple linear regression

Shahrukh Sattar Chowdhury  
*Dept. of Engineering (Electrical and Computer Engineering)*  
*Lakehead University*ON, Canada  
schowdh9@lakeheadu.ca

*Abstract*—

Keywords— PMSM

# Introduction

Recent years have seen an increase in the popularity of permanent magnet synchronous motor electric motors. The PMSM is chosen because automotive motor applications demand high power and torque densities. High thermal stress is placed on motors in the quest of high efficiency and full utilization of the motor's capabilities. This can lead to the failure of a few components, such as the melting of stator wiring or the loss of flux in the permanent magnet [1]. High temperatures reduce vehicle performance and thermal robustness, according to numerous experimental studies. However, standard procedures make it nearly hard to measure these temperatures. Sensor-based temperature monitoring can only be used for the measurement of the exterior motor chassis since the rotor's intricate interior structure makes it impossible to gain access to. Direct observations are therefore replaced by indirect ones, and the interior temperature is derived from traditional thermal models [2]. However, this necessitates extensive topic expertise and cannot simply be applied to all situations. Regression, Multiple Regression and K-nearest neighbors are used in this study to investigate and estimate dynamic motor temperatures using basic, quick machine learning techniques. Since these algorithms are tiny and efficient, they're suitable for use in small, low-power systems. Better motor temperature forecasts are made in this study, which can then be used in real-life scenarios like issuing warnings when the temperature of the motor is excessively high and optimizing the performance of the motor is applied to. Sensor data from the PMSM motor is used as input to the algorithm [3]. On a test bench, it measures the temperatures of various components and outputs that information. The target temperatures are estimated using extensive feature engineering in a causal approach. Due to low power embedded traction drive hardware, temperature estimates in production will be implemented in an automotive setting where lean computation and lightweight implementation are crucial. It has been shown that adopting approaches like kernel features and minimal regularization parameters can assist reduce the bias in linear regression, resulting in higher overall performance. Minimizing model sizes and processing time is critical for maintaining real-time functionality.

.

# Literature Review

In earlier work of predicting the motor temperature was done using heat models. These models were based on the heat transmission theory. These models did not exploit the use of testing data to make thermal predictions. A Study showed overheating causes significant motor damage and is a major source of concern for the machine's design. Due to a lack of precise temperature calculations, device utilization is lower and material costs are higher. Traditional thermal models based on thermodynamic theory outperform ML techniques based solely on acquired data [4]. On another paper, for traction drive applications, PMSMs are commonly employed in the manufacturing and electric car industries. Controlling the effects of overheating by monitoring the temperatures of its vital components. Machine learning techniques are increasingly being used in other industries, such as healthcare, with promising outcomes. Permanent magnet synchronous motors were the subject of the research [5]. Jun Lee and Jung-Ik Ha conducted a pilot study keeping the temperature of a permanent magnet synchronous machine (PMSM) in a safe range is essential for maintaining machine performance. Feedforward neural network (FNN) for the estimation of a PMSM is proposed. The proposed model estimates the temperatures of multiple parts of interest with a heat point of view [6]. The last review was done based on a German OEM’s prototype model. Paderborn University's LEA department gathered test bench measurements. The primary goal of recording the data set is to be able to simulate the stator and rotor temperatures of a PMSM in real time. Direct measurement of rotor temperatures using thermal sensors is not possible due to the intricate construction of an electric traction drive, and even in the case of stator temperatures, sensor outage or even deterioration cannot be administered properly without redundant modelling. Furthermore, as the need of functional safety grows, correct thermal modelling becomes increasingly crucial. The main goal of this project is to create a model with proper feature engineering that casually forecasts four target temperatures [7].

# Methodology

In this project, different statistical models have been estimated and statistical tests have also been conducted. A linear regression model was used to find out the mean squared error. After visualizing the models, a performance matrix was used in regression to predict a number. After linear regression, a multiple linear regression was used where all the independent values where in X while the dependent was assigned to Y. Grid search has been used with the help of some libraries such as Decision Tree Regressor, Decision Tree Classifier, KNeighbourRegressor. Besides, these polynomial, KNN and Randomforest regression was used to get an accurate temperature prediction.

For the data, we used 12 columns from the possible 14 columns of the dataset. Features- As of now, the data collected will be preprocessed, and the dataset will have ten attributes. The features of the dataset, as well as the specifics of the features, are listed in the table below.

However, because many data will be irrelevant to this topic, the data must be cleansed to remove the noisy data. We'll remove any instances where any of the feature information is missing, and we'll focus on supervised techniques with our target characteristics.

Table 1: Dataset with all the column details

|  |  |
| --- | --- |
| u\_q | Voltage q-component measurement in dq-coordinates (in V) |
| coolant | Coolant temperature |
| Stator\_winding | Stator winding temperature measured with thermocouples |
| u\_d | Voltage d-component measurement in dq-coordinates |
|  |  |
| Motor\_speed | Motor speed (in rpm) |
| I\_d | Current d-component measurement in dq-coordinates |
| I\_q | Current q-component measurement in dq-coordinates |
| pm | Permanent magnet temperature in Celsius measured with thermocouples and transmitted wirelessly |
| Stator\_yoke | Stator yoke temperature in Celsius measured with thermocouples |
| Ambient | Ambient temperature (in °C) |
| Torque | Motor Torque (in Nm) |

The columns in the data are not null. They have a data type of float. After transposing we could get the count, mean, standard deviation and the value of them in each quartile. For the data formation, in the first model there was no testing and training but for the rest of the data 80 percentage was for the training and the rest was for the testing.

For the data visualization, the seaborn library was used. It is a python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

A picture containing text, screenshot

Description automatically generated

Figure 1. A Joint plot representation between stator winding and permanent magnet temperature

Figure 1 shows the two histograms are shown in the margins of the graph in the above scatter plot. There appears to be a correlation between the columns "stator winding" and "pm" in the scatterplot, as values of one grow, the other increases as well. Because the data points are dispersed throughout the graph, it appears as though the association is compelling in strength.

Graphical user interface, chart, treemap chart

Description automatically generated

Figure 2. Heatmap demonstration of the correlated features

Figure 2 depicts the correlated features between all the attributes. The graph shows a stronger correlation on the negative when it is green. On the contrary, a strong correlation can be seen on the positive while it is more on the aquamarine side. For example, the class label (permanent magnet temperature) has a dynamic correlation with the features stator winding, stator tooth, stator yoke, ambient and current d-component measurement in dq-coordinates.

A picture containing diagram

Description automatically generated

Fig 3. Model Equation

The above graph in Fig 3. shows a linear regression model. The model presents and equation where pm equals to -0.002 + stator\_winding \* -0.725. In the graph, there is a linear relationship between stator winding and permanent magnet temperature. After setting the dependent and independent values, an ordinary least square was done. Ordinary least square done to check inconsistent and different values. The method OLS evaluates the association between one or more independent factors and a dependent variable by minimizing the sum of the squares in the difference between the observed and predicted values of the dependent variable set as a straight line. OLS regression will be explained in this post in the context of a bivariate model, which is a model in which just one independent variable (X) predicts a dependent variable (Y). The logic of OLS regression, on the other hand, is simply extended to a multivariate model with two or more independent variables.

After doing it, a constant value of the dependent variable stood still at 23.5141. A mean squared error was done but it was too much and not good. Then fitting the models, predicted and real values of first five were shown. We can see from the graph that the strong linear relation between the permanent magnet temperature and stator winding temperature leads to a better overall temperature prediction.

For hyperparamter tuning, Grid Search was performed. Knn was used to arrange the neighbours and KNeighborsRegressor was used. A cross validation was done in the grid search. All possible combinations were calculated by using fivefold. The training examples are class-labeled vectors in a multidimensional feature space. The training phase of the method consists solely of storing the training samples' feature vectors and class labels. k is a user-defined constant in the classification phase, and an unlabeled vector (a query or test point) is classified by assigning the label that appears most frequently among the k training examples closest to that query point. Euclidean distance is a widely used distance metric for continuous variables. Another metric, such as the overlap metric, can be used for discrete variables, such as text classification (or Hamming distance). For example, in the context of gene expression microarray data, k-NN has been used as a metric in conjunction with correlation coefficients such as Pearson and Spearman. When the distance metric is learned with specific algorithms like Large Margin Nearest Neighbor or Neighborhood components analysis, the classification accuracy of k-NN can often be greatly improved.

Lastly, A random forest regression was used. Regression problems can be solved using the Random Forest algorithm. Regression trees are utilized instead of classification trees to form the forest in this case. Because of this, they are an ensemble of regression trees and are utilized for nonlinear multiple regression. Continuous output variables are distributed throughout each leaf, and their mean value is used to determine the final output.

# Results And Discussion

After making a heatmap with co-related features and setting permanent magnet temperature as a dependent variable, we used ordinary least square on that. At last, we got a summary of the ordinary least square. The result of the OLS Regression is given in the table below:

Table 2: OLS Regression Result for the Linear Model

|  |  |
| --- | --- |
| Model | OLS |
| No of Observations | 1330816 |
| Data Frame Residuals | 1330814 |
| Data Frame Model | 1 |
| Covariance Type | Non Robust |
| R-squared | 0.633 |
| Adj R-squared | 0.633 |
| F-statistic | 2.300e+06 |
| Log-likelihood | -5.1391e+06 |
| Aic (Akaike Information Criterion) | 1.028e+07 |
| Bayesian Information Criterion | 1.028e+07 |
| Skew | -0.445 |
| Kurtosis | 4.947 |
| Jarque-Bera | 254026.525 |
| Cond. No. | 182. |

The constant has a co-efficient of 23.5141, standard error of 0.025, t (co-efficient divided by its standard error) is 935.428 while with the stator winding, it has a co-efficient of 0.5275, standard error of 0 and a t statistic of 1516.504. At first, the mean squared error (mse) was calculated to be 304370245.0472444. The error was significantly high and so rsquared error, which is the standardized version of MSE. The real and predicted values of the 5 first instances are presented in a Table 3.

Table 3: Real and Predicted Values of RMSE

|  |  |  |
| --- | --- | --- |
| Real Values | Predicted Values | Error |
| 24.554 | 33.581 | -9.027 |
| 24.538 | 33.584 | -9.046 |
| 24.544 | 33.582 | -9.038 |
| 24.554 | 33.579 | -9.025 |
| 24.565 | 33.579 | -9.013 |

For the multiple linear regression, the results of the OLS Regression are given below:

Table 4: OLS Regression for the Multiple Linear Model

|  |  |
| --- | --- |
| Model | OLS |
| No of Observations | 1064652 |
| Data Frame Residuals | 1064641 |
| Data Frame Model | 11 |
| Covariance Type | Non Robust |
| R-squared | 0.986 |
| Adj R-squared | 0.986 |
| F-statistic | 6.698e+06 |
| Log-likelihood | -3.6329e+06 |
| Aic (Akaike Information Criterion) | 7.266e+07 |
| Bayesian Information Criterion | 7.266e+07 |
| Skew | 0.512 |
| Kurtosis | 5.345 |
| Jarque-Bera | 290410.683 |
| Cond. No. | 4.75e+03 |

In the second algorithm, we also applied multiple linear regression using OLS where all the independent values are in x while all the dependent values are in y. The root mean squared error (RMSE) was calculated to be 7.3399 for the training set and 7.35244 for the test set. Then, the model was cross-validated using ten-fold cross validation and a performance metric of r squared error. The average error value for the training model was 0.85503 and the test model was 0. 8545.The same procedure was repeated to calculate the RMSE of the model where the value for the training set was 7.234 and test set was 7.248.

Chart, line chart

Description automatically generated

Fig 4. Predicted and Actual values of the instances

The actual and predicted values of our class label PM was plotted in Figure 4. The color of the training data was red while blue represented the test values. And from the distribution plot, we can see that both the train and tested values were closely predicted. It is possible to see the data's variety by looking at a Distplot, or distribution plot. Distributions of continuous data variables are shown in a Seaborn Distplot. The distplot is depicted using the Seaborn module and the Matplotlib module, respectively.

For the KNN algorithm, a grid search was performed varying the values of K from 1 to 5, validated using 5-fold cross validation. The optimal K value was found to be 4. Using the best parameter, ie, k = 4, the RMSE value was 3.7098. Apart from that, the RMSE values of the training set was also calculated for k = 1 and k = 2. The results of the error for k = 1 was 0 and 4.5704 without and with cross validation respectively. Similarly, for k = 2 where the values were 2.063 and 4.1742.

Another algorithm was used for the dataset which was a second-degree polynomial regression model. Residual error was calculated for the model which is the difference between the mean of the actual values and the predicted values.

Chart, scatter chart

Description automatically generated

Fig 5. Scatter Plot

From the Fig 5. we can see a graph of the residual error. A linear regression object was created and stored. After that the models were trained using the train data sets. Coefficients and variance score was found using the regression coefficient and regression score method. The above graph has a scatter plot. Two numeric variables are represented by dots in a scatter plot. An individual data point's value is indicated by the position of each dot on the horizontal and vertical axes. A scatter plot is a visual representation of the relationship between two or more data points. Here the green one is the test data where the blue one is test. The scatter plot above shows the train and test data for residual error. From the plot, we can see a generally tight positive correlation between them.

Table 5: Comparison of RMSE of the proposed methods.

|  |  |
| --- | --- |
| **Algorithm** | **RMSE (Root mean squared error)** |
| Ordinary Least Square (Linear Regression) | 11.504 |
| Ordinary Least Square (Multiple Linear Regression | 7.248 |
| KNN | 3.7098 |
| Polynomial | 0.8977 |
| Random Forest (still working) |  |

Table 5 represents the RMSE Scores of all the algorithms that were applied. The Root mean squared error for the OLS in linear was very high while the polynomial had the lowest one. Still working on the Random forest one.

##### References

1. Kirgsn, “Electric Motor Temperature”, https://www.kaggle.com/wkirgsn/electric-motor-temperature, Jun. 18, 2019 [Feb. 2,2020].
2. IT. Huber, W. Peters, and J. B¨ocker. Monitoring Critical Temperatures in Permanent Magnet synchronous Motors Using LowOrder Thermal Models. In International Power Electronics Conference, 2014.
3. D. Huger and D. Gerling, “An Advanced Lifetime Prediction Method for Permanent Magnet Synchronous Machines,” in Proc. of Electrical Machines (ICEM), 2014.
4. Kirchgässner et al., Data-Driven Permanent Magnet Temperature Estimation in Synchronous Motors with Supervised Machine Learning 2021
5. Kenneth Anufuro, Temperature Estimation in Permanent Magnet Synchronous Motor (PMSM) Components using Machine Learning 2020
6. J. Lee and J. Ha, "Temperature Estimation of PMSM Using a Difference-Estimating Feedforward Neural Network," in IEEE Access, vol. 8, pp. 130855-130865, 2020, doi: 10.1109/ACCESS.2020.3009503.
7. Soprun, O., Bublyk, M., Matseliukh, Y., Andrunyk, V., Chyrun, L., Dyyak, I.I., Yakovlev, A., Emmerich, M., Osolinskyi, O., & Sachenko, A. (2020). Forecasting Temperatures of a Synchronous Motor with Permanent Magnets Using Machine Learning. MoMLeT+DS.