IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

Using Support Vector Machine & Regression for Final Value Estimation of a Buck Converter

Adwit A Shah

[1]Department of Electrical Engineering, Manipal Institute of Technology, Manipal

Abstract

SPICE (Simulation Program with Integrated Circuit Emphasis) programs are ubiquitous in circuit design owing to their speed and accuracy. Problems with its speed arise when you want to simulate a complex third-party imported model that is based on Kirchhoff's laws to run a transient simulation. We propose a machine-learning based approach that predicts the output voltage of one such model from its operating characteristics. This prediction will be done in two phases; first, we identify the type of transient that the model is going through. Then by using the corresponding model, we would predict the output voltage of the model. We have chosen a simple buck converter as the model because of its low feature dimensionality compared to other models. We show that this pipelined method is very effective as it is faster than traditional SPICE matrix solvers and has comparable accuracy.

Introduction

Buck converters are an important electrical circuit that find a variety of uses in all things. A buck converter is typically used when the voltage that is available at the source is more than the voltage that is required at the output. Ever since the advent of electrical technology, the need for such a converter has always been filled in by either using a resistor or a LDO (Low-Dropout) regulator.

With the recent push in electrical efficiency and the development of low cost, high efficiency transistors; buck converters have seen a remarkable growth in their usability. Apart from the typical topology, a number of variants have popped up namely; the simple buck, the interleaved buck, the isolated buck (flyback converter) & the three-level buck. These converters make up for more than 75% by volume in voltage regulation applications in a variety of industries, namely; automobile, commercial electronics, computers etc. Figure 1 below shows a typical application of a simple buck converter.

Though the design of a buck converter involves a lot of equations and component selection, its design is straightforward. The difficult part is when engineers need to characterize its operation especially in a system level simulation. Originally for simulations, the buck had always been thought of as a circuit with a constant output voltage, but that assumption is false when we consider the real-life applications. The buck converter tends to exhibit different operating points for different load and line conditions. Figure 2 shows a typical example of a line regulation; when there is an abrupt change in the input voltage to the converter, the output of the converter suffers a minor dip before it settles down to a new operating voltage level. On the contrary, when the output voltage of the converter

Figure 1 Typical buck converter application

drops when there is an abrupt change in the output current of the converter, it is known as a load regulation as shown in Figure 3.

For the purposes of this project, we have chosen to predict only the final operating value of the buck and not the transients that accompany them.

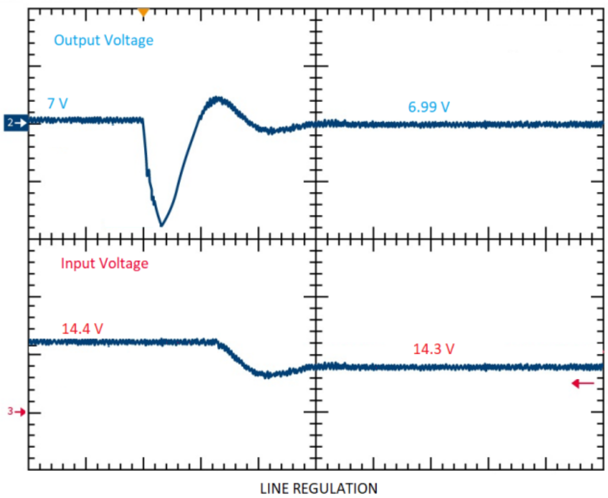

Figure 2 A typical line regulation example

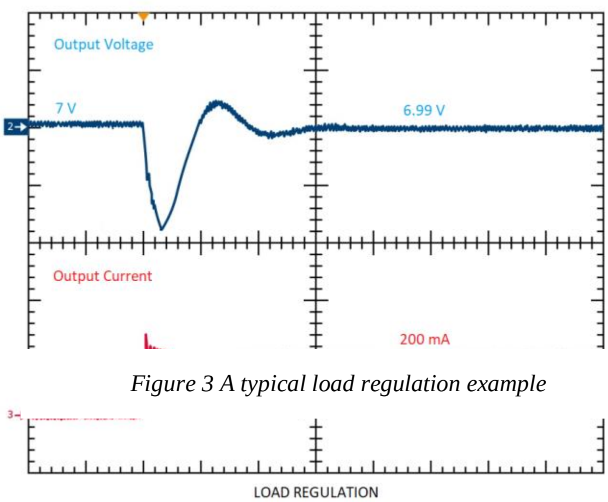

Figure 3 A typical load regulation example

Related Work

Since the advent of machine learning algorithms, electrical researchers have been using them in a variety of electrical engineering applications. There is ample work previously done on the applicability of data mining algorithms in the electrical domain. From the design of circuits with low electromagnetic emissions using SVR (1) to the diagnosis of faults in a converter circuit by using SVM (2), and another paper on Neural-Network based estimation of Power Electronic waveforms (3) show the applicability of data mining algorithms in electrical engineering. Some researchers have started using data science in the design of analog circuits; for example, using Monte-Carlo Analysis for the selection of the switching frequency. Apart from that though when it comes to modelling an IC, researchers have always preferred to use the state space method in SPICE (4).

AI Approach

The AI system architecture is depicted in Figure 4, 5. This problem can broadly be classified as a Supervised Machine Learning problem since the ability of the agent to correctly predict the change in the output of the converter will be improved as more data points are presented. The system takes an input of multiple readings of a converter's operating point. These include the output voltage, input voltage, input current and the output current. The system takes this dataset and predicts the output voltage of the buck converter. To be able to do this, the data vector is passed through a SVM classifier where it is either classified as a line regulation or a load regulation. Once the correct regulation has been judged, the system then uses the corresponding Support Vector Regression model to predict the output of the buck converter.

The problem can broadly be identified as a two-fold learning problem:

- Correctly identify the type of regulation (Load or Line) that is occurring

- Use a model of that regulation to correctly predict the change in output voltage

Classification

Classification of the vectors to classify the type of regulation is identified as a supervised learning problem. This is because as more vectors are fed into the system, the ability of the system to correctly classify the next vector improves. In order to identify the type of regulation from a data point, a classification problem; we will be using Support Vector Machine. Even though there are other methods, both supervised and unsupervised to do the same, the reasons for using support vector machine are depicted below:

1) The data that is available to us is labelled. This renders the use of cluster analysis, an unsupervised learning technique, improper for this application. Cluster analysis is typically used for automatic identification of natural grouping of things when the data that is available is unlabeled. (5,6)

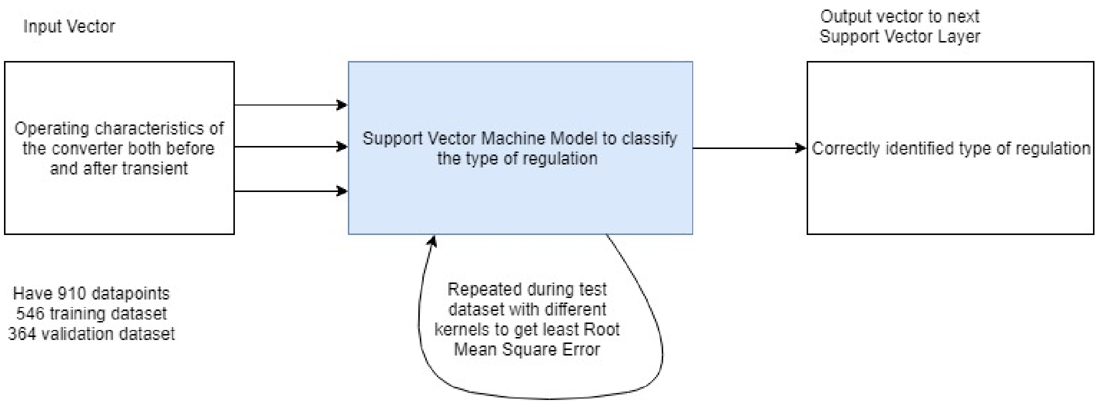

Figure 4 Support Vector Machine actigram

2) A closer look at the dataset tells us why we cannot use decision trees. Decision trees only work when the classification problem is a linear one. By using a kernel in SVM, we are able to chalk out a seemingly linear hyperplane that separates the two classes (2). This is particularly useful when there is both a line and a load regulation in the same vector. The decision tree would not be able to identify the class of the vector, categorizing it as both a load and line regulation. In turn, based on the separating hyperplane, SVM would be able to map the vector to only one of the classes.

Prediction

Predicting the change in the output voltage of the system is identified as a supervised learning regression problem. Maheshwari talks about how Artificial Neural Networks are a valid algorithm for this application (6). ANNs work great as a black box system and will continue to learn by adjusting its internal computation and communication parameters. Apart from that, they are flexible with the type of data they support and have been previously used to predict power electronics waveforms (3). The reason that we have chosen to use Support Vector Regression over ANNs is because ANNs are more prone to overfitting. Another reason for choosing SVR over Neural Networks is that ANNs are good at forecasting the transient behavior of the waveforms, but the steady state estimation is not very accurate as they tend to converge onto the local optima (7).

Implementation

The experimental setup for the collection of the converter current and voltage readings is detailed in Figure 5. The supply voltage and the load current need to be variable voltage and current sources respectively to emulate a load or line transient. Care needs to be taken so that the oscilloscope is at an edge trigger at the output current for a load transient and input voltage for a line transient. The datasets were then imported to MATLAB where it was checked for missing values and erroneous readings.

Dataset

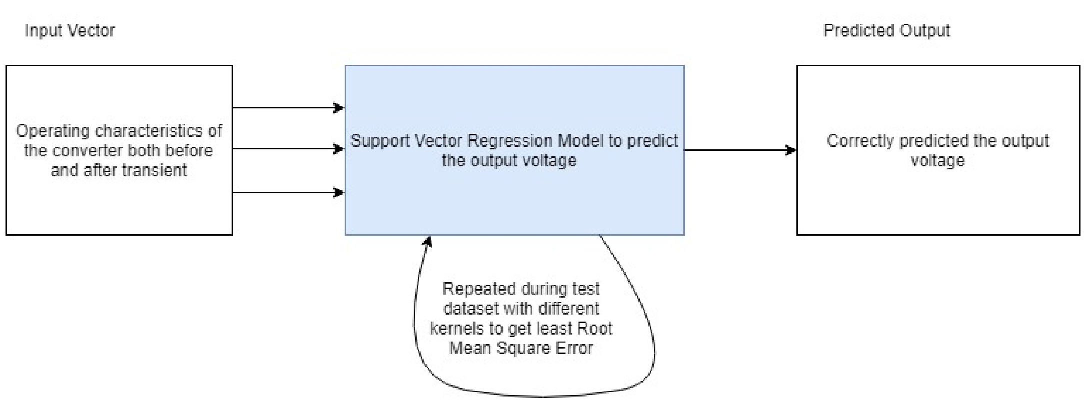

Figure 5 Support Vector Regression actigram

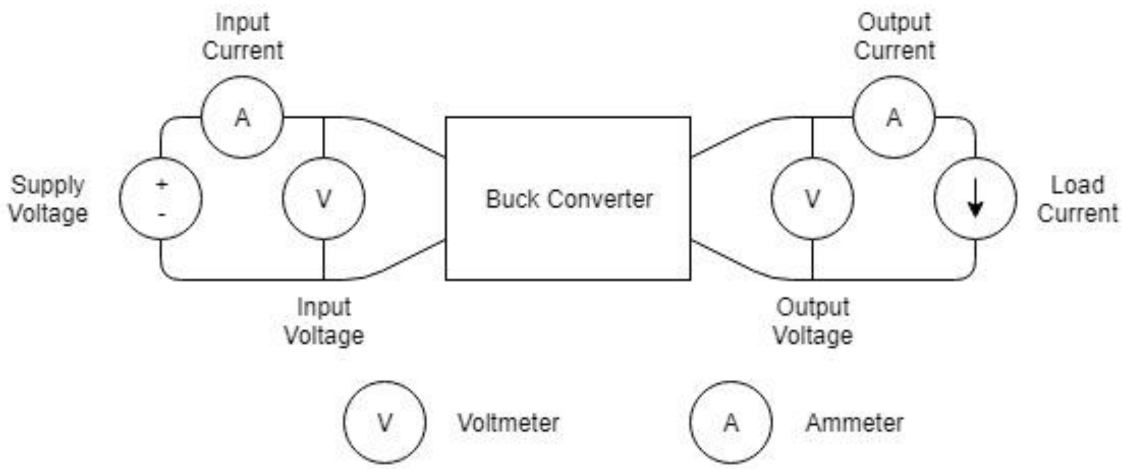

Figure 6 Experimental setup for data acquisition

The dataset consists of multiple vectors with a reading for each variable for both before and after the event. More explanation about the variables has been provided in Table 1. The dataset acquired does not have a label variable but a calculation of the change in output current and input voltage, the system can be used as an equivalent label. We restricted the type of dataset that can be used for testing to be one where either there is a load transient or a line transient. The reason we had to do this is because we want to train the SVM model on neutral data.

Table 1 Converter operating variables

Variable	Explanation
VO1	Output Voltage before event (rms)
VO2	Output Voltage after event (rms)
IO1	Output Current before event (rms)
IO2	Output Current after event (rms)
VI1	Input Voltage before event (rms)
VI2	Output Voltage after event (rms)
II1	Input Current before event (rms)
II2	Input Current after event (rms)

SVM for Classification

We used the MATLAB fitcsvm function to extract a model for classification by using all the variables from the dataset. The function also used the equivalent label vector for classifying the two classes. The function uses the labels to come up with a graphical representation of the two classes and then calculates a plane that divides the two classes within a margin of error. Because of the high dimensionality of the data (greater than 3), the graphs cannot be seen. The inputs to the function are horizontally concatenated vectors that include all the variables that were experimentally measured as well as a matrix that contains the label. The output of the function is a SVM model that can then be used to classify the validation test set. The inputs to the prediction function are the validation set without a label and the pretrained SVM model. The error for classification was calculated by using the percentage error (8) and different kernel functions were used to reduce the percentage error. Since the dimensions of the datasets are greater than three, It is not possible to visualize what the classification looks like.

SVR for prediction

We used the MATLAB fitrsvm function to extract a prediction model by using all the variables from the dataset and the target variable as different inputs. The fitrsvm uses the output voltage of the converter to curve fit a line. The coefficients of this line are dependent on the operating characteristics of the converter. The input to the function is a horizontally concatenated vector matrix that includes all the variables except for the target variable (output voltage after the event) & the target variable in another matrix. The output of the function is a SVR model that can be used to predict the output voltage for the validation data set. The inputs to the prediction function are the validation set without the output voltage and a pretrained SVR model. The error for the prediction was calculated by using the root mean square error (1). Different kernel functions were used to reduce the RMS error. Apart from there being two SVR trained models, one for line regulation and one for load regulation; we would have two different models; one subset of models would use the output of SVM from the classification model and predict the output voltage while another subset of models would use the validation set labels. This allowed us to calculate the RMS error for just the SVR for prediction as well as the combined error of the SVM for classification followed by SVR for prediction.

Figure 7 explains the program through a flowchart.

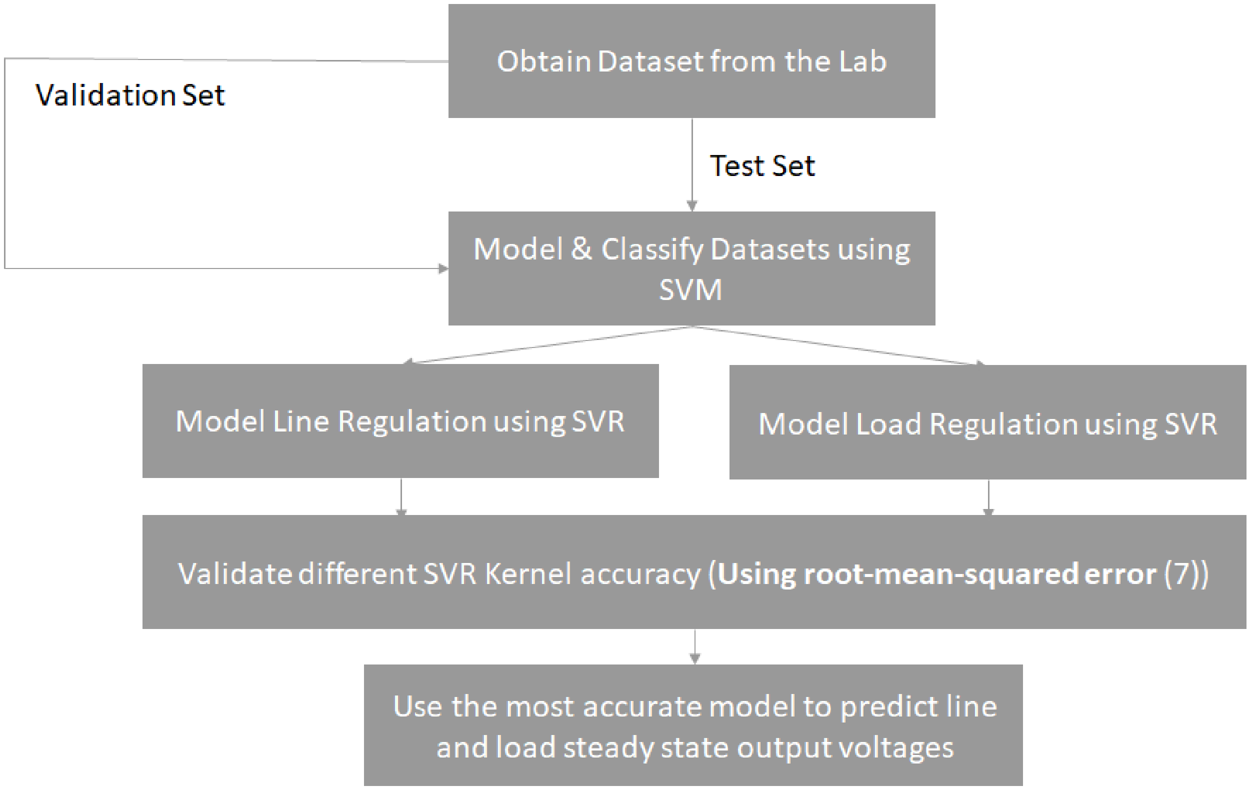

Figure 7 Program flowchart

Evaluation

In this section, we present the experimental results for our two learning tasks, namely: classification of the type of regulation & output voltage prediction. We then present some qualitative results of our approach and discuss its limitations & how to overcome them. For this project, we have chosen to measure the performance of any classifier or regression model by using the Root Mean Square error (1).

Regulation Type Classification

Recall that the primary goal of the classifier is to identify the type of regulation that takes place and then place them into one of two classes; line regulation or load regulation. To do so, we had to split the original dataset into two; training dataset (60% of the dataset) and the evaluation dataset (40% of the dataset).

Each vector was then passed through a Support Vector Machine classifier to get the correct classification of the same. Basing itself on the training dataset, we trained the SVM for classification with different kernel functions namely; linear, polynomial & Radial Basis Function (RBF). The validation dataset was then fed to the SVM classifier and the predicted classes were compared to the correct version. The kernel function with the least RMS error turned out to be the polynomial kernel function, & the SVM classifier with this kernel was chosen. Table 2 shows the results of the SVM classifier for different kernel functions. As it can be seen from the table, the polynomial function manages to reduce the error of the RBF as well as the linear function by almost 10%.

Table 2 Accuracy of the SVM classifier

SVM Accuracy	
Kernel Function	Percentage Accuracy
Linear	90.256
Polynomial	99.725
RBF	82.334

In order to check if the SVM classifier was functioning well for a line and a load transient i.e. when both the regulations take place together, the different kernels for SVM was used to classify 20 vectors of special use case (both line and load regulation) where the best kernel function again was the polynomial.

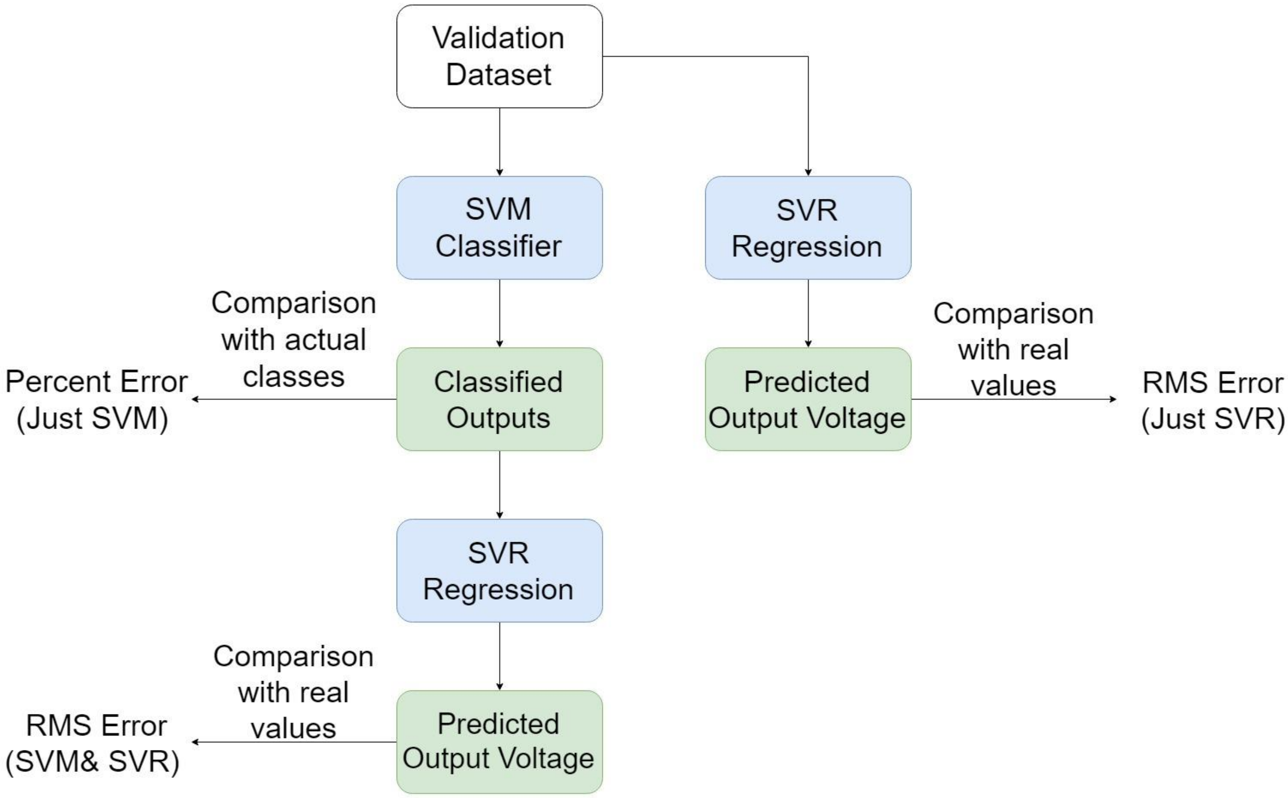

Figure 8 Error calculation flowchart

Table 3 Accuracy of the SVR

SVR Accuracy		
	RMS Error	
Kernel Function	Load Regulation Model	Line Regulation Model
Linear	0.0685	0.1012
Polynomial	No Convergence	No Convergence
RBF	0.02	0.0636

Output Voltage Prediction

Recall that the primary goal of the predicter was to forecast what the output voltage of the converter would be based on the initial operating parameters and the final operating parameters. Again, to do so, we had to use the training dataset to train the SVR curve fitting model and then use the validation dataset to come up with the best kernel function with the Root Mean Square error. We divided the complete dataset into parts; 60% was used as the training dataset and the rest 40% was used as an evaluation dataset. In order to calculate just the error that in introduced by using the SVR for prediction, we need to make sure that the validation data that we have is devoid of error from the error introduced due to the classification model. This was done by giving SVR the actual labels and not the classifier outputs.

Basing itself on the training dataset, we trained the SVR for regression model with different kernel functions, namely; linear, polynomial and Radial Basis Function. The validation dataset was then fed to the SVR for regression and then the predictions for the output voltage were compared to the correct output voltages. The kernel function with the least RMS error turned out to be the RBF kernel function, & the SVR predicter with this kernel was chosen. Table 3 shows the results of the SVM classifier for different kernel functions. As it can be seen from the table, the RBF kernel function manages to reduce the error of the polynomial as well as the linear function by almost a factor of two.

Classification & Regression Combined Evaluation

The next step is to evaluate how the system as a whole functions. In order to do that, we will have to feed the validation dataset to the SVM for classification. Following this, the classified dataset would then be fed as an input to the SVR for regression and the predicted output voltage would be compared to the actual output voltage. The Root Mean Square error would then be calculated for the same. A flowchart expressing how the different errors were calculated is presented in Figure 8. The different errors for the combined SVR & SVM system are shown in Table 4.

Table 4 Accuracy of SVR & SVM combined

SVR & SVM Combined Accuracy		
	RMS Error	
Kernel Function	Load Regulation Model	Line Regulation Model
Linear	0.0548	0.1419
Polynomial	No Convergence	No Convergence
RBF	0.0226	0.0836

Lessons Learnt

Initially, the project proposal included a different actigram. From the dataset, the system was supposed to cluster the points that were similar and hence we would not have to label the dataset. We had a good understanding of the dataset and how we would proceed with the project; the roadmap was well defined. We hit our first roadblock right in the data acquisition phase. While changing one of the parameters of the circuit, we realized that it seemed to affect another variable as well. Hence, we had to change the way the dataset was being acquired. This was done by including a label vector.

Another place where we started having problems was during the initial assessment of the project algorithms. After acquiring a small set of data, the data was prepared, and the appropriate algorithms were applied. This was done following the track textbook's (9) recommendation to divide and conquer the problem with smaller amounts of data and get closer to the heart of the solution in an iterative sequence of steps. Applying k-means clustering to the dataset, it turned out that the RMS error was the least for k=3. This was contrary to our expectations of only 2 clusters. Seeing as we did have a label vector, SVM for classification was then used. Not only did this algorithm converge but with a polynomial kernel function the algorithm was able to correctly classify the small dataset with 100% accuracy.

The next step was to apply SVR to the small dataset (50 vectors). The algorithm did not yield very good results as its predictions had an RMS error much greater than expected. After some deliberation, we realized that this could be because of the small dataset that we were modeling the trainer on. Increasing the number of vectors to include all of the training set gave us much accurate predictions reaffirming our initial assumption that this was indeed a supervised learning problem.

During the course of this project, we closely followed the 6 step Cross Industry standard process for Data-Mining (CRISP-DM). Based on those steps, we have come up with our own 4 step standard for data mining:

1) Data Understanding: A really important step to understanding whether the proposed algorithm would work for the dataset is understanding what the data is. Even though a theoretical understanding of the underlying engineering would help, the data itself could be widely bereft of the same. You not only need to understand what the data implies theoretically, but also need a practical understanding of what your feature vectors are and what your algorithm does.
2) Data Preparation: Correctly so, the textbook (9) talks about how one can spend about 60-70% of their time focusing on cleaning up the data and making sure that it can be used by the software. This step should be followed once you have understood what the data is. Apart from that, the project also required some manipulation in data so as to make the data more conducive to the algorithms.
3) Modeling: Not only do you need to be patient with the algorithms but also keep your options open to the choice of algorithm. Initially the project was needed to predict the output of the system by using ANNs, but after a thorough understanding of the pros and cons of different algorithms, SVR came out on top.
4) Evaluation: You not only want to evaluate the output of the model but also need to analyze why a certain model gives you better results. You typically spend a lot of time trying to understand what the results imply. In the project, SVR combined with the SVM yielded a lower error to when SVR was given the true classification labels. Understanding why that was happening was where we spent a lot of our time on.
5) Presentation & documentation: This is probably the most important step of all. Basing our project on some previously completed work, we understood how important clear presentation and documentation was. The report needed to be both devoid of electrical as well as AI jargon so that it would be understood by anyone from one of those domains.

Conclusion

In this paper, we tackled the problem of predicting the output voltage after a transient of a buck converter based on their initial operating characteristics. To achieve this, we developed a pipelined approach that first classifies a vector to either a load or a line regulation, then uses the corresponding regression model to predict the output voltage. The prediction and classification both used a supervised learning approach where the input was actual operating conditions of a buck converter; both initial and final. This approach when compared to the electrical black box equivalent is faster and less computationally recursive.

In future work, we will extend the domain of the problem to include prediction of the transient curves. This would involve using computer vision algorithms combined with convolution neural networks to predict the output voltage waveform. However, the final value estimation would still use SVR owing to the neural networks affinity to converge on the local optima (1). This would involve using a smoothening algorithm that would settle the discontinuity of the predicted waveform. Finally, we will need to extend our system to handle a realistic scenario. This will include ensuring that this model could be used in a SPICE (Simulation Program with Integrated Circuit Emphasis) program. For that, we would need to make sure that the user is able to call the SVR model from the SPICE software.

References:

1) Arnaud Breard, Redha Moulla, and Christian Vollaire (2016),Metamodel of Power Electronic Converters Using Learning SVR Method Coupling With Wavelet Compression

2) Hu, Q & Wang, R.-J & Zhan, Y.-J. (2008). Fault diagnosis technology based on SVM in power electronics circuit. Zhongguo Dianji Gongcheng Xuebao/Proceedings of the Chinese Society of Electrical Engineering. 28. 107-111.

3) Min-Huie Kim, Bimal K. Bose (1996), Neural-Network Based Estimation of Power Electronic Waveforms.

4) S. Ahmed, S. Mostafa (2017), State Space Modeling of a Buck Converter and Designing a Controller

5) S. Nuchprayoon, "Electricity load classification using K-means clustering algorithm," *5th Brunei International Conference on Engineering and Technology (BICET 2014)*, Bandar Seri Begawan, 2014, pp. 1-5

6) Anil Maheshwari, (2015), Data Analytics made Accessible, Chapter 4, pg 82

7) N. Suyaroj, N. Theera-Umpon and S. Auephanwiriyakul, "A comparison of NN-based and SVR-based power prediction for mobile DS/CDMA systems," *2008 International Symposium on Intelligent Signal Processing and Communications Systems*, Bangkok, 2009, pp. 1-4.

8) Ahmed Amine, Ladjici. (2009). Power System Applications of Support Vector Machine in Classification and Regression.

9) Anil Maheshwari, (2015), Data Analytics made Accessible, Chapter 4, pg 85

10) Mackworth & Poole (2017), Artificial Intelligence: Foundations of Computational Agents, Section 7.6