PREDICTIVE CONTROL USING NEURAL NETWORK SYSTEMS

Dainger Adams, Malcolm Bailey, and Nathan Le

Introduction

Modern day power plants must be very versatile to meet fluctuating demand. Often times, rapidly adjusting to this changing power demand leads to accelerated equipment degradation, which in turn hurts company profit. There is a lot of ongoing research into methods of effectively meeting power demand without damaging generators. The inability to rapidly change to the needs of a region without damaging equipment results in a optimization issue between producing the right amount of power without hurting equipment excessively.
By training a neural network to aid in power demand prediction over a 24 hour period, our project can identify the most effective way to generate power for the entire day. This will lead to minimizing equipment damage and optimally supplying power to the populace. This, in turn, results in a better bottom line for any company that employs such an approach. Combining a trained neural network with an optimization algorithm will generate a smooth graph of varying setpoints of power generation. The smooth curves mean a softer demand on generators while still supplying an ample amount of power when it is needed.
This project was broken into three phases creating a predictive neural network, creating a optimizing set point generator, and controlling a system.

Samples

A years worth of power generation data from February 22, 2017 to February 22,2018 was obtained from Rocky Mountain Power. This data was broken up into 10 minute increments. The corresponding weather data for this time period was obtained through the API provided by the National Oceanic and Atmospheric Administration's (NOAA) .

Acknowledgements

ROCKY MOUNTAIN POWER

NOAA

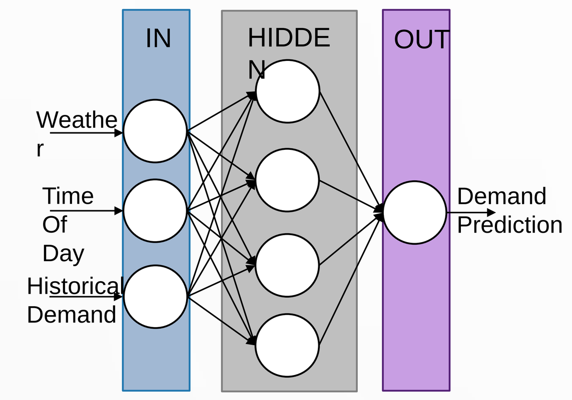
OBJECTIVES

- Train a neural network to take in historical power generation and weather data to predict future power demand
- Manipulate forecast so a set point generator can control a heat exchanger simulation optimally.

I. PREDICTION

Theory

- Three layer neural network (input layer, hidden layer, output layer).
- Trained on historical data to predict future data with similar inputs.
- Hidden layer applies function to inputs and combines the outcome to produce output.



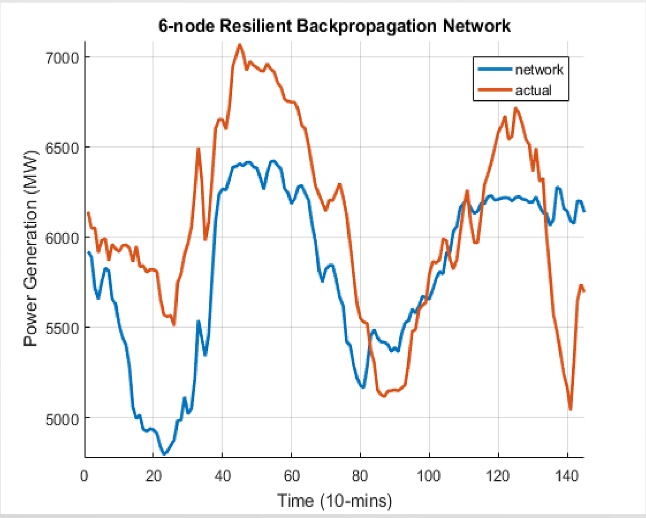
- Deciding the number of nodes is largely trial and error but should the number of nodes should not exceed number of inputs to avoid over fitting.

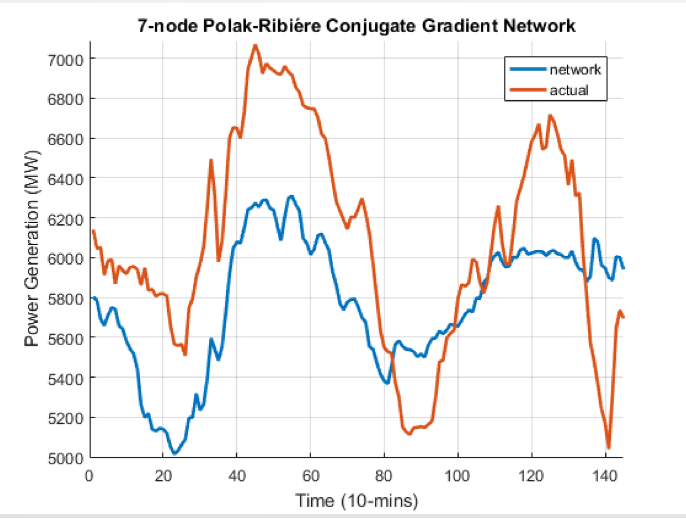
Procedure

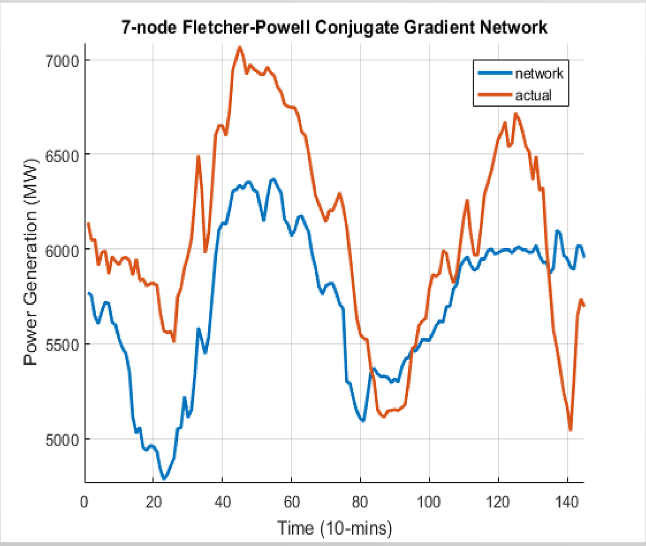
- Acquire and format power data from Rocky Mountain Power (RMP)
- Acquire and format historical weather data NOAA API
- Train the network with the known data to get a network that can be used to predict future values

Results

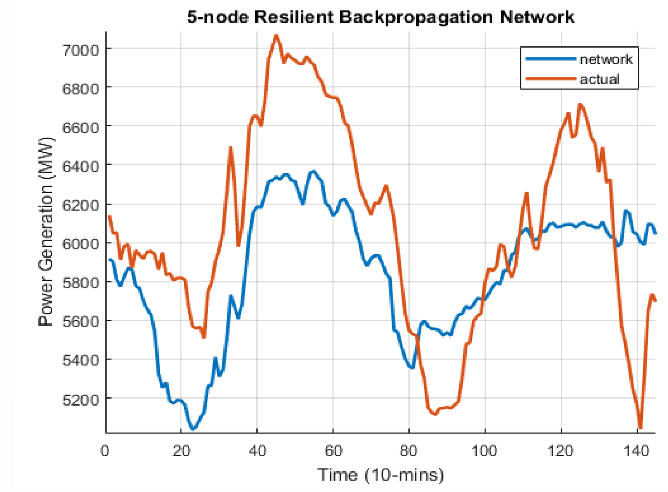
	Algorithm	nodes	Mean Squared Error (± SD)
First	Resilient Backpropagation	6	560.2 ± 59.2
Second	Polak-Ribiére Conjugate Gradient	7	598.3 ± 80.3
Third	Fletcher-Powell Conjugate Gradient	7	606.2 ± 89.6

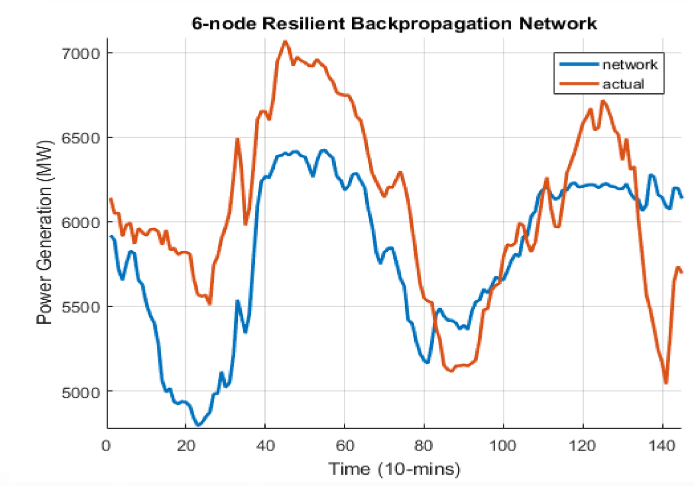


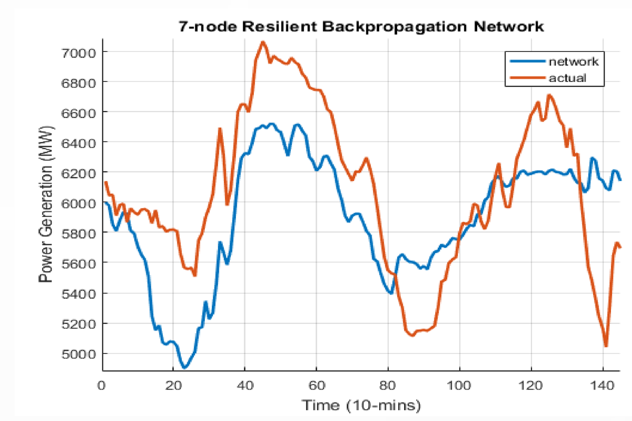




These plots show differences in the outcome for Resilient Backpropagation, Polak-Ribiére Conjugate Gradient, and Fletcher-Powell Conjugate Gradient algorithms.







These plots show differences in the number of nodes for resilient backpropagation algorithm.

	Nodes	Mean Squared Error (± SD)
First	6	560.2 ± 59.2
Second	7	588.2 ± 85.2
Third	5	623 ± 80.1

II. OPTIMIZATION

Theory

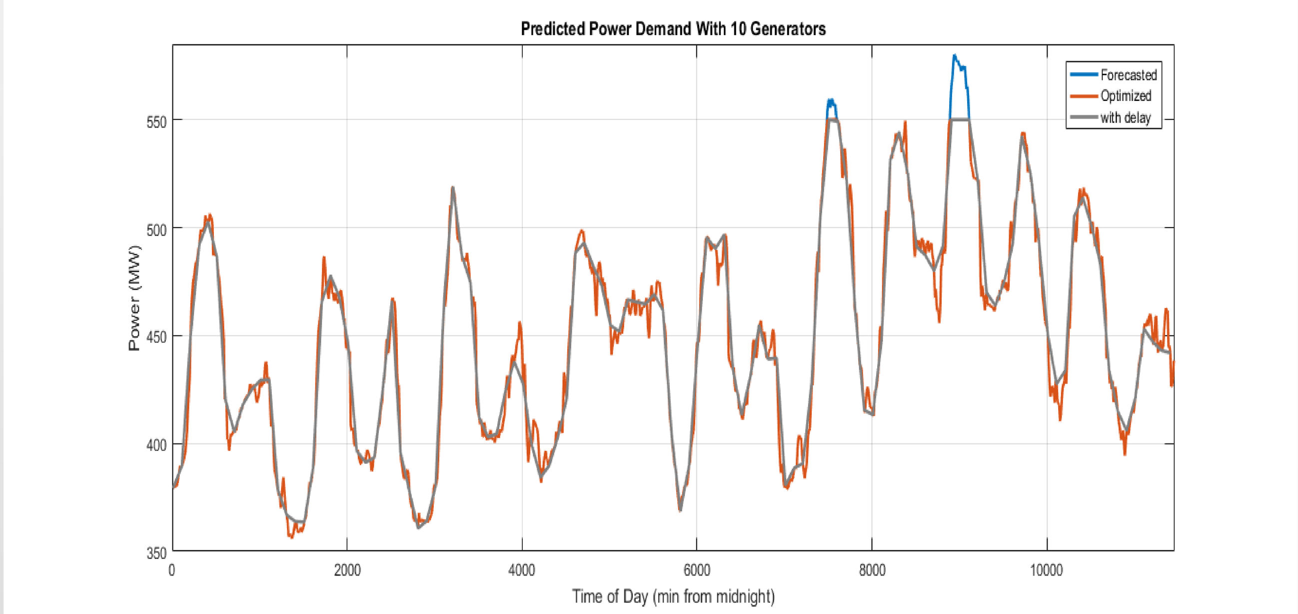
- Generator has specified ramp up and ramp down rates that can't be exceeded as well as maximum operating generation
- This script needs to convert the forecast into an operable set point.

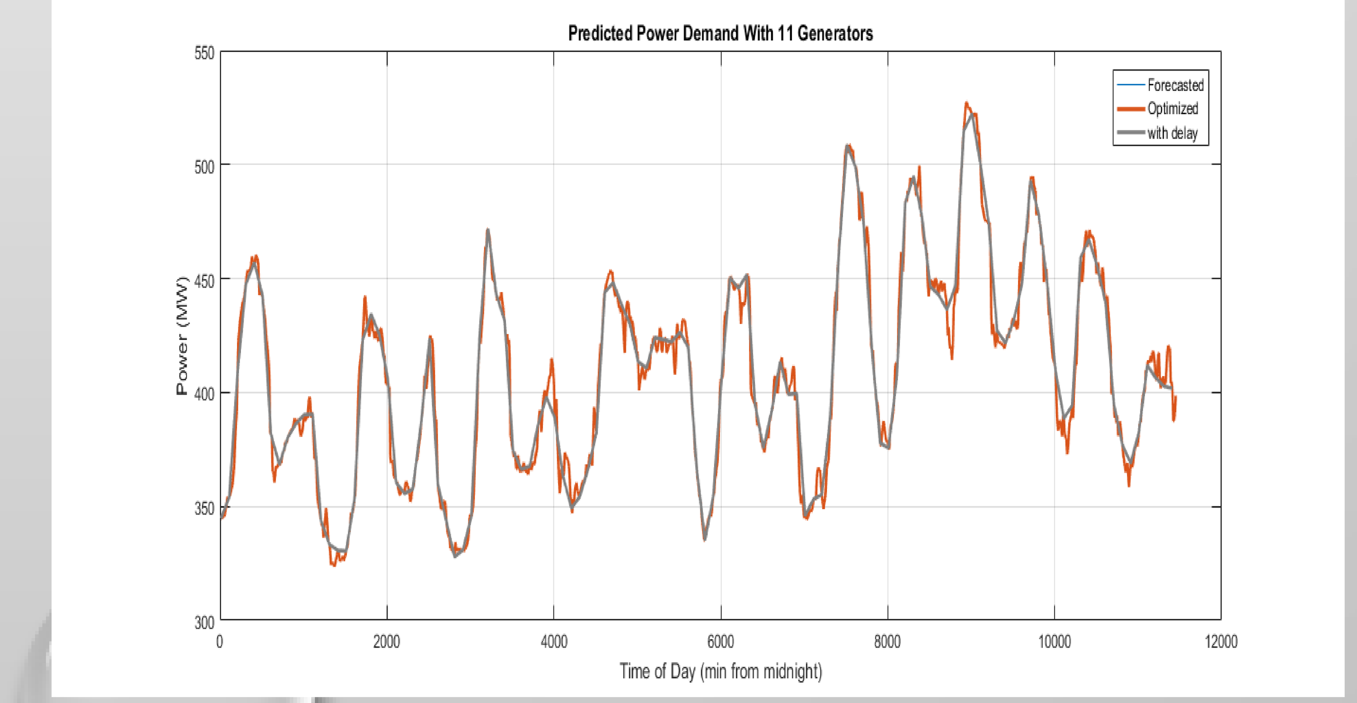
Procedure

- Create script that will keep setpoint in operable position and be able to break forecast into smaller segments as if multiple generators were being used.

Results

This shows what happens when the setpoint is achievable and when it is not. The data is broken into 100 min intervals instead of 10 min intervals to help with controllability





First Plot shows what optimizer does when it can't reach operable value, the second plot is when it can reach an operable value

III. CONTROL

Equipment

- MATLAB
  - System Identification Toolbox
  - Simulink
- Shell-and-Tube Heat Exchanger

Procedure

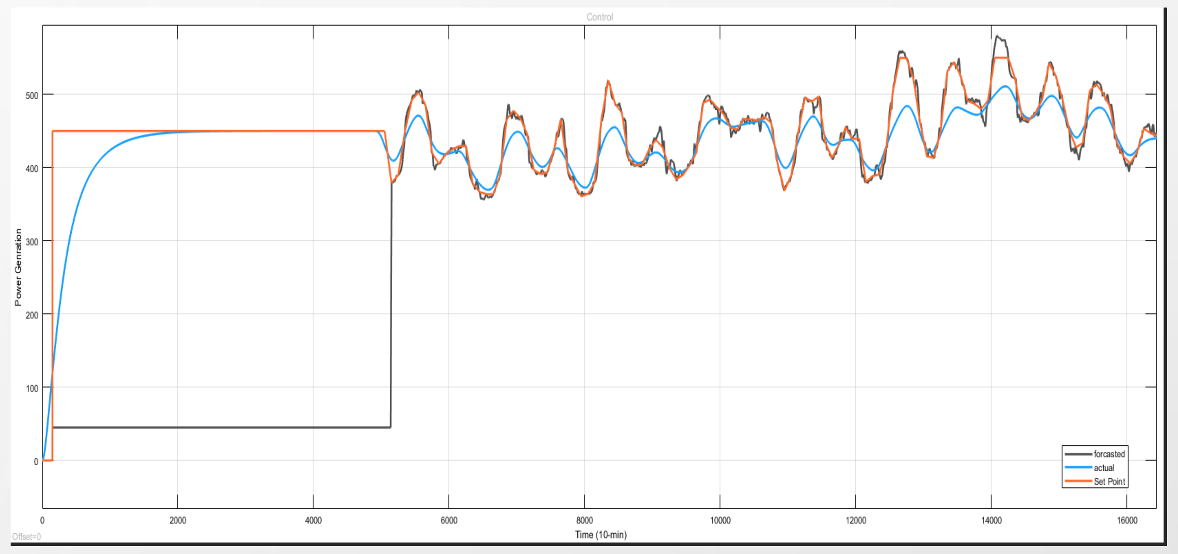
- Using MATLAB's System Identification Toolbox, create a model and use Simulink to simulate a control strategy for the heat exchanger

Results

- Obtained transfer function

$$\frac{Y(s)}{U(s)} = -\frac{0.41183}{(432.57s+1)(42.351s+1)}e^{-.5s}$$

- Obtained control response.



Plot showing the simulated controlled heat exchanger with the set point and forecasted predictions.

Discussion

In this project the overall results were that we were able to predict power demand, that demand was optimized, and a control system was created to handle that demand. The main issue comes from prediction with only a year worth of data the network is not able to closely enough predict future demand.

Future Research

In the future it would be good to see if more data, possibly around 4 to 5 years worth of data to produce a more accurate prediction as well as running the control strategy on a real life heat exchanger/generator system.

References

Arroyo, J., & Conejo, A. (2002). Optimal response of a power generator to energy, AGC, and reserve pool-based markets. IEEE Transactions on Power S ystems, 17(2). doi:10.1109/tpwrs.2002.1007910 (pg. 404-410)