NPNL Power Oscillation Contest

Ben Bragg

What is a Power Grid Oscillation?

- Some oscillation is natural and fine.

- Bad oscillations:
  - Low-frequency
  - Continuous
  - Growing

- Can be caused by:
  - Load changes
  - Equipment going offline
  - Problematic equipment

system oscillation of HSG and Wind Park

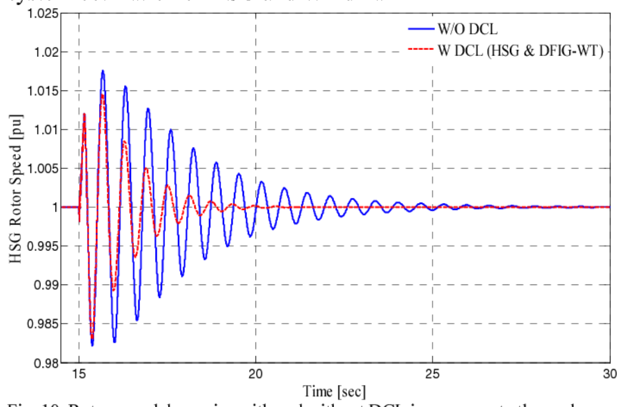
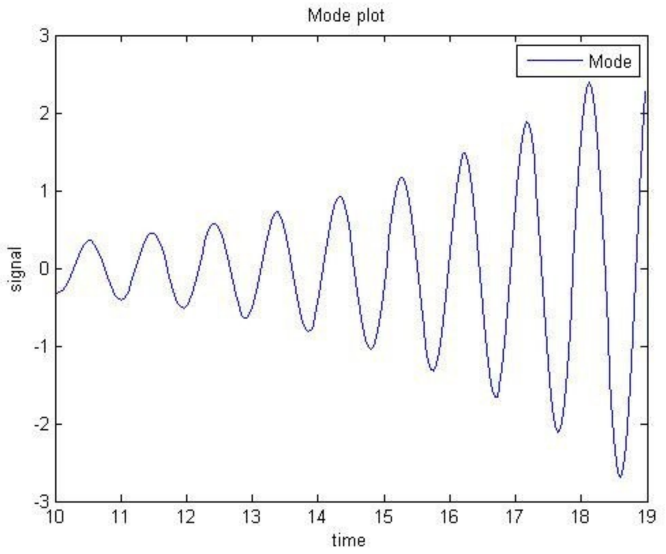


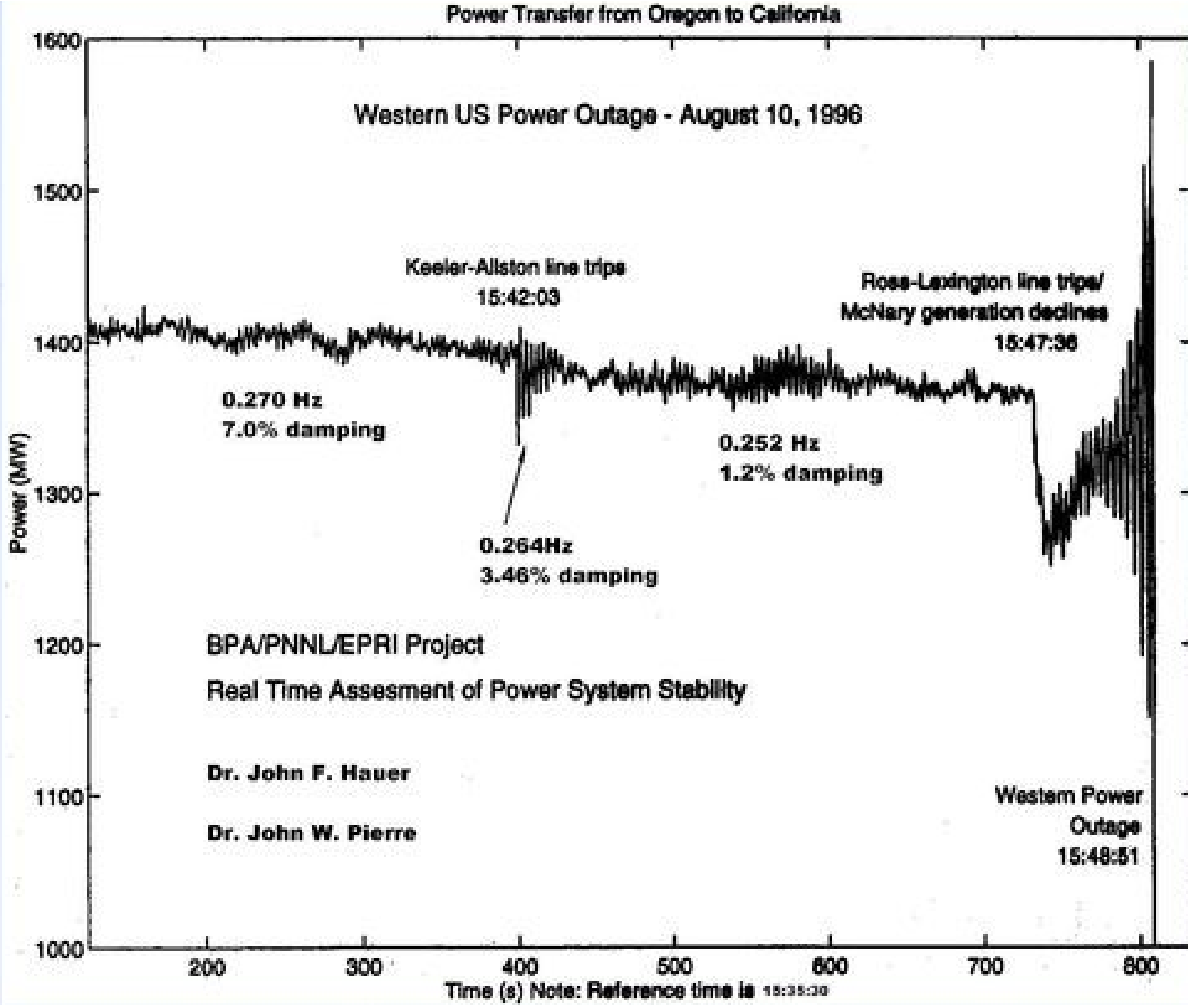
Fig. 10. Rotor speed dynamics with and without DCL in response to three-phase-



Why is it Important to Detect Them?

- Can damage equipment and cause cascading blackouts
- (See graphic to right: Western US Power Outage on August 10, 1996)

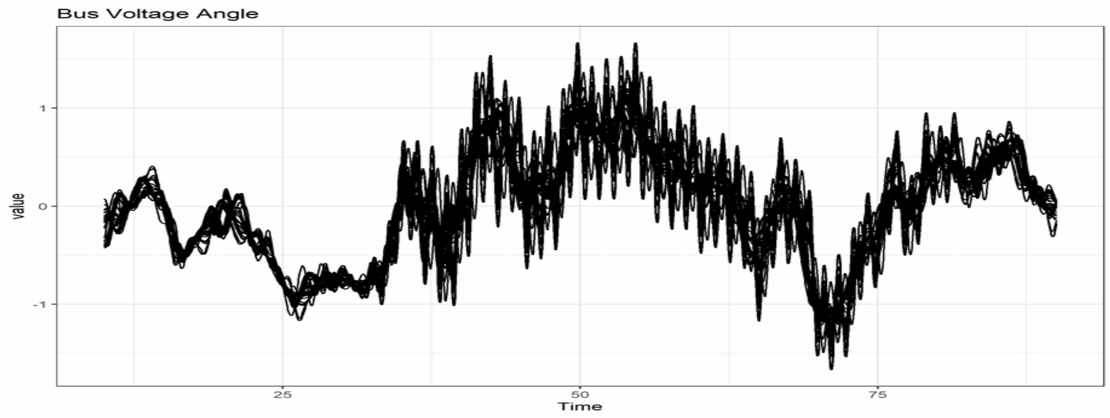
- Early detection can identify where and what the problem is, allowing controllers to by-pass the issues and keep the power flowing.

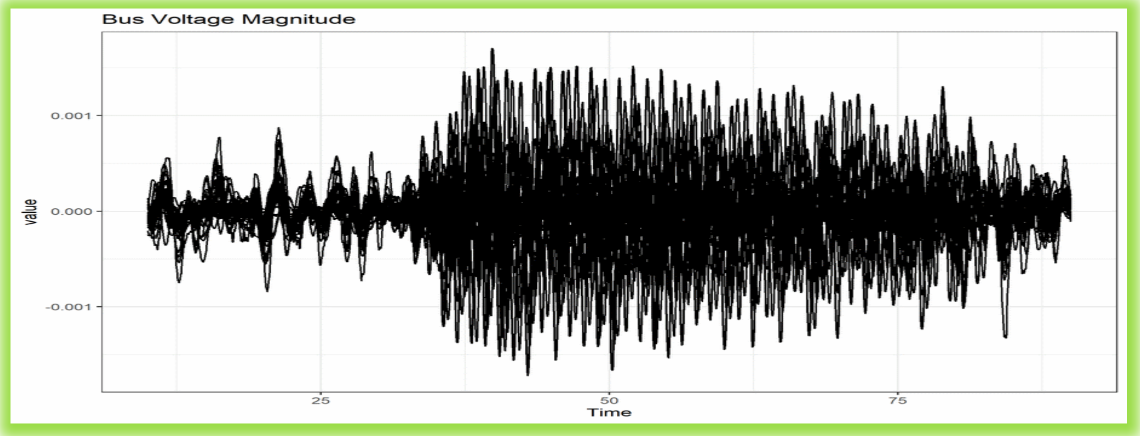


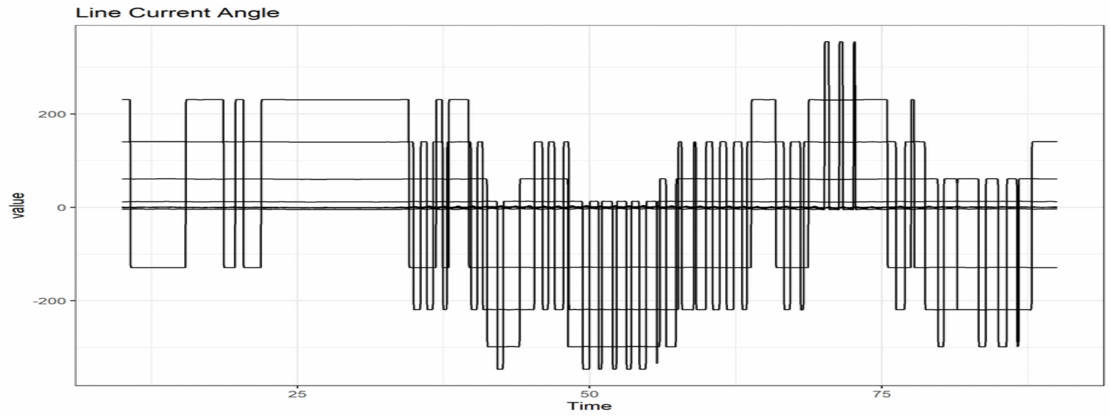
The Contest

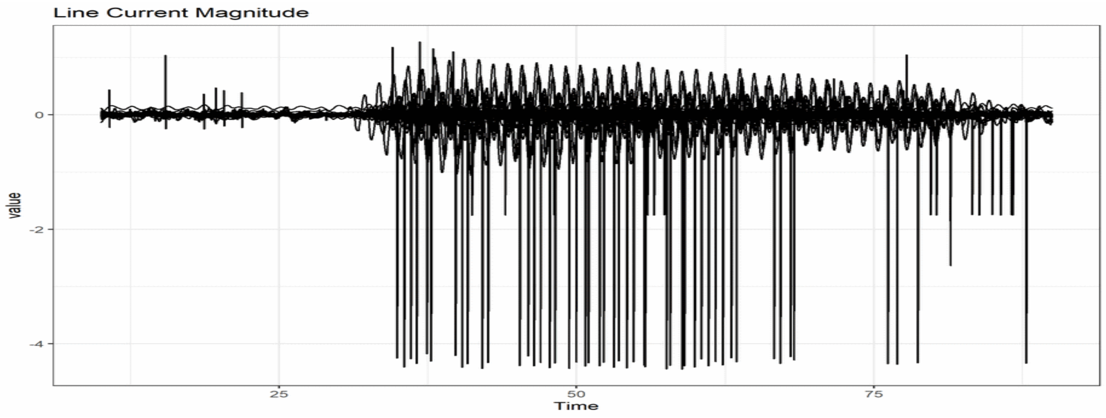
- Hosted by PNNL
- Goal: Locate source of oscillation
- 4 sets, 13 cases
  - Bus Voltage Angle
  - **Bus Voltage Magnitude**
  - Line Current Angle
  - Line Current Magnitude

Bus Voltage Magnitude Was Best









My Plan

1) **Classification through Regression**

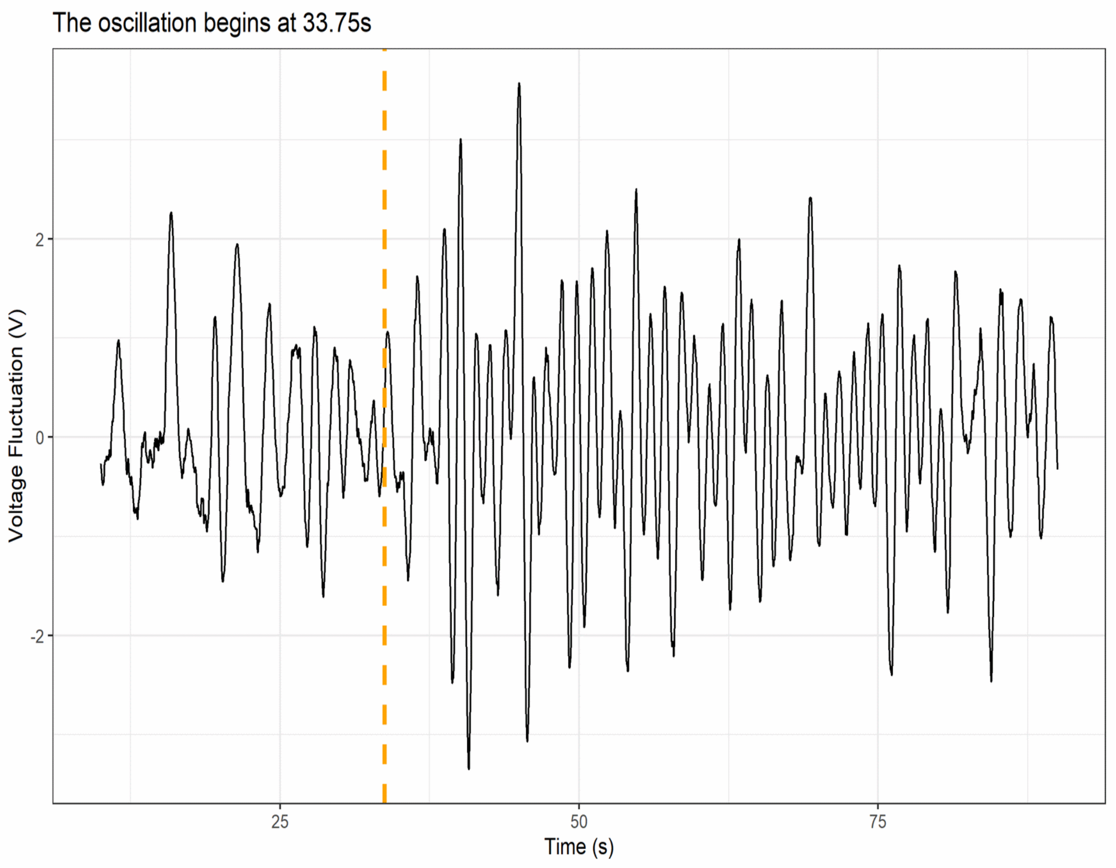
2) Mark Case 1 columns with timestamps
    1) (Visual Inspection)

3) Build 1D-CNN model

4) Train model on Case 1 data

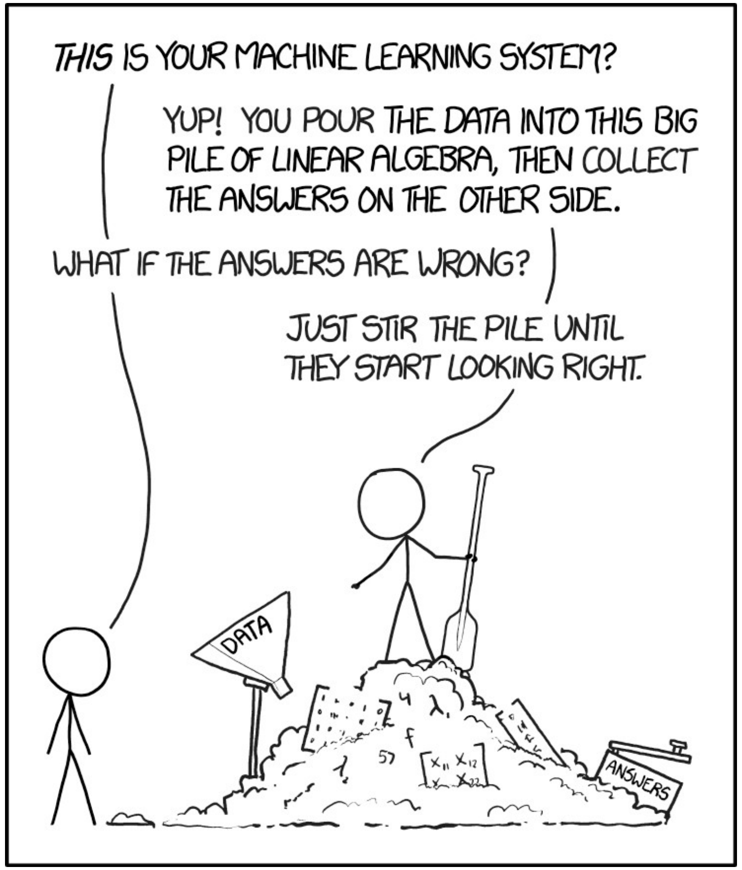
5) Apply model to cases 2 – 13

6) Mark lowest timestamp in each case as oscillation origin



Convolutional Neural Networks (CNNs)

- Dense Neural Networks: Recognize specific full sequences
- CNNs: Recognize patterns in space/time
  - 2DCNNs: Images
  - **1DCNNs: Time-series**



https://xkcd.com/1838/

My Tools

- IDE: RStudio
- Languages: R 4.1, Python 3.7
- Libraries:
  - Pacman (manage libraries)
  - Tidyverse (data wrangling and visualization)
  - TensorFlow/Keras/KerasTuneR (neural networks)
  - Reticulate (Python integration)

The Data

- Data provided in ‘table’ text format
- Shape
  - Time in s, 2701 1/30s increments
  - 58 buses monitored
- Cleaning
  - Remove 1st 10s of data (startup artifacts)
  - Impute median over NA values
  - Normalize data

```r
case_01_bus_vol_mag <- read.table("2021_IEEE_NASPI_OSL_Contest_Data_Set/Case1/BusVolMag.txt",
                                  header = TRUE, quote = "'") %>%
  as_tibble() %>%
  filter(Time >= 10)

## messing with apply() to create averages and impute NAs

list_na <- colnames(case_01_bus_vol_mag)[ apply(case_01_bus_vol_mag, 2, anyNA) ]
median_missing <- apply(case_01_bus_vol_mag[, colnames(case_01_bus_vol_mag) %in% list_na],
                        2,
                        median,
                        na.rm = TRUE)
## replace missing values with means
c1bvm_imputed <- case_01_bus_vol_mag %>%
  mutate(X7031.COLOEAST....20.0 = if_else(is.na(X7031.COLOEAST....20.0), median_missing[1], X7031.COLOEAST....20.0),
         X6533.EMERY.......20.0 = if_else(is.na(X6533.EMERY.......20.0), median_missing[2], X6533.EMERY.......20.0),
         X1002.FOURCORN....345. = if_else(is.na(X1002.FOURCORN....345.), median_missing[3], X1002.FOURCORN....345.),
         X6202.GARRISON....500. = if_else(is.na(X6202.GARRISON....500.), median_missing[4], X6202.GARRISON....500.),
         X6404.GONDER......345. = if_else(is.na(X6404.GONDER......345.), median_missing[5], X6404.GONDER......345.),
         X2100.IMPERIAL....230. = if_else(is.na(X2100.IMPERIAL....230.), median_missing[6], X2100.IMPERIAL....230.),
         X6101.MIDPOINT....500. = if_else(is.na(X6101.MIDPOINT....500.), median_missing[7], X6101.MIDPOINT....500.),
         X4201.NORTH.......500. = if_else(is.na(X4201.NORTH.......500.), median_missing[8], X4201.NORTH.......500.),
         X3906.ROUND.MT....500. = if_else(is.na(X3906.ROUND.MT....500.), median_missing[9], X3906.ROUND.MT....500.),
         X6403.VALMY.......345. = if_else(is.na(X6403.VALMY.......345.), median_missing[10], X6403.VALMY.......345.))
```

The Network

1) Split data and labels into *train* and *test* groups
2) Reshape data and labels for 1DCNN
3) Create, tune, and fit model to data and labels
4) Evaluate model on test data
5) Use model to make predictions from other data

- `1DCNN Input Layer`
- `MaxPooling1D Layer`
- `1DCNN Layer`
- `MaxPooling1D Layer`
- `1DCNN Layer`
- `MaxPooling1D Layer`
- `Flatten Layer`
- `Dense Layer`
- `Dense Output Layer`

```
Loss: 6.98      MAE: 1.99
```

Results

Case	Bus
c01	X1431.PALOVRD2....20.0
c02	X1403.PARKER......230.
c03	X2202.MIGUEL......230.
c04	X6301.BRIDGER.....345.
c05	X4101.COULEE......500.
c06	X6404.GONDER......345.
c07	X2202.MIGUEL......230.
c08	X6301.BRIDGER.....345.
c09	X1403.PARKER......230.
c10	X1403.PARKER......230.
c11	X4009.BIG.EDDY....230.
c12	X6333.BRIDGER.....20.0
c13	X2619.SYLMARLA....230.