MMAE 500

Getting a Grip: Tire Modeling with the Magic Formula and SINDy

Jay Steinberg

4/27/2023

Abstract:

Laboratory tire testing data is examined and fit with the popular Pacejka Magic Formula model. The data is then explored using sparse identification of nonlinear dynamics (SINDy) in an attempt to glean a dynamical model directly from the data. Several SINDy configurations were used, and many models were produced, but none were viable in describing the behavior of the tested tire.

Introduction:

Tires are ubiquitous in the modern world. Nearly every ground vehicle, and most aircraft, uses tires of some sort. Tires operate in a very nonlinear manner, in some ways acting as a continuum solid, and in other ways acting as a fluid. There have been a multitude of studies on tires, and while great advances have been made in understanding, modeling, and predicting tire behavior, it is still considered a bit of a "black art". This project seeks to understand the standard modeling process of a tire, and then apply modern data-driven methods to laboratory tire test data to match or improve upon the current state of the art in tire modeling.

The motivation for this exploration is the recent availability of a new round of tire testing data from the Formula Society of Automotive Engineers (SAE) Tire Test Consortium (TTC). Formula SAE is a standardized program of university teams building single-seat formula race cars for competition. Several involved universities and SAE partnered as the TTC to test and quantify tire performance as an aid to the engineering and development of race cars. The TTC data is available to member schools. The Illinois Tech FSAE team, IIT Motorsports, recently became reacquainted with the TTC as we are developing a new car.

The TTC is supported by the Calspan Tire Research Facility (Calspan). Calspan has developed a highly standardized and controlled tire testing program and apparatus. All available test data is produced on Calspan equipment. Calspan also provides a detailed testing schedule supplement with individual test program details and parameters. The most recent 9th round of TTC testing data was used for this study.

Calspan produces separate run files for drive/brake testing and pure cornering testing. The data from provided in DAT file format for general use, and the TTC creates MAT files for easy import into MATLAB. This analysis was performed in Python so the DAT form was used. The test data is accompanied by guidance documents including a “run guide” outlining which tires and wheels were used for each run, a “contents” guide outlining the data channels, test plans, and data formats, as well as summary tables that include specific comments and loading schema. A sample of the data used for this analysis as well as the accompanying documentation is included with this paper. The data channel and abbreviation guide is shown in Figure 1.

Channel	Units	Description
AMBTMP	degC or degF	Ambient room temperature
ET	sec	Elapsed time for the test
FX	N or lb	Longitudinal Force
FY	N or lb	Lateral Force
FZ	N or lb	Normal Load
IA	deg	Inclination Angle
MX	N-m or lb-ft	Overturning Moment
MZ	N-m or lb-ft	Aligning Torque
N	rpm	Wheel rotational speed
NFX	unitless	Normalized longitudinal force (FX/FZ)
NFY	unitless	Normalized lateral force (FY/FZ)
P	kPa or psi	Tire pressure
RE	cm or in	Effective Radius
RL	cm or in	Loaded Radius
RST	degC or degF	Road surface temperature
SA	deg	Slip Angle
SL	unitless	Slip Ratio based on RE (such that SL=0 gives FX=0). This is “traditional” or “textbook” slip ratio.
SR	unitless	Slip Ratio based on RL (used for Calspan machine control, SR=0 does not give FX=0).
TSTC	degC or degF	Tire Surface Temperature--Center
TSTI	degC or degF	Tire Surface Temperature--Inboard
TSTO	degC or degF	Tire Surface Temperature--Outboard
V	kph or mph	Road Speed

Fig. 1 Calspan data channels. Data was recorded at 100hz.

To understand and make use of tire testing data, we use a common reference frame and coordinate system defined by the Society of Automotive Engineers. The standardized coordinates and directions are outlined in the SAE J670e standard. For our purposes, the reference frame is a right-handed coordinate system with positive x in the tire's direction of steering, y to the right, and z into the ground as shown in Figure 2. Moments are described as about each primary direction vector, with positive sign conventions also as shown. Finally, key angles referenced are slip angle α (the angle between the direction of steer *x* and the actual direction of travel, and inclination angle *γ*, the angle between the normal vector of the ground surface and the tire's tilt.

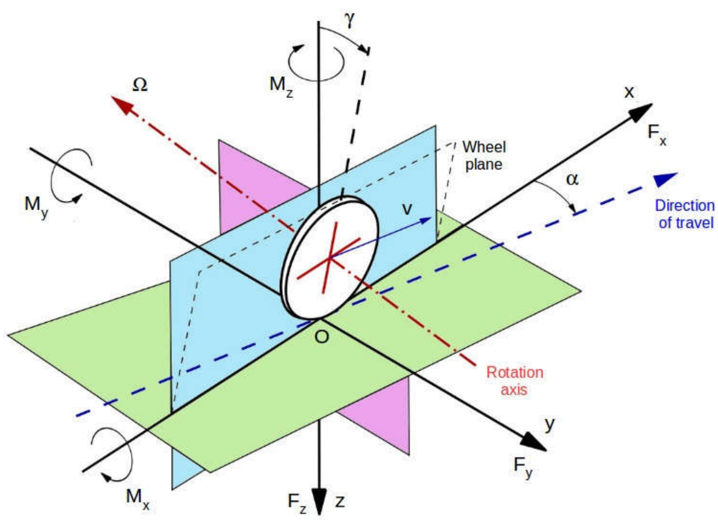

Fig. 2 Standard coordinate reference frame for tire analyses (SAE J670e).

After understanding how to describe the tire's coordinate frame, we can begin with modeling. The foremost authority on the subject of tire modeling and vehicle dynamics is Hans Pacejka. He developed several models to describe tire behavior, most notably the *Magic Formula* (MF). The MF is a semi-empirical model that correlates well with tire test data. It has been in use in industry for decades, and to this day is the standard tire model. The MF has been revised several times, and the various versions of it include a wide variety of parameters; however, for the scope of this project, the simplest version, called the four-term model, is used (Pacejka, 2006). The basic MF for side force is in Equation 1:

$$FY = D\sin\left[C\arctan\{B\alpha - E(B\alpha - \arctan(B\alpha))\}\right] \tag{1}$$

Where FY is lateral/cornering force and *α* is slip angle as before, parameter B is a stiffness factor, D is a peak factor/scalar, and C and E are generally fit with regression (Pacejka, 2006). This Magic Formula format is regularly used for basic tire models in student designs, video game physics, and general tire analyses, and is typically shown as several fit curves superimposed for different vertical loads.

An interesting and recent development in data-driven modeling is the algorithm for sparse identification of nonlinear dynamics (SINDy). This method obtains a dynamical system model from data. SINDy analyzes time series data and the derivatives thereof, and by assuming that physical systems only have a few dominant terms, it uses a regression that promotes sparsity among a library of nonlinear “candidate functions”. This method has been used successfully to solve the Lorenz system and many other dynamical systems, including those exhibiting significant noise. This paper will use SINDy to look for alternative models that fit the TTC data.

Methods:

The plan for this project was to analyze and interpret the available TTC cornering test data, fit a Pacejka Magic Formula model for reference, and use SINDy to recreate a similar or improved model. Several run data files were analyzed and compared to the Calspan Run Guide to find a suitable dataset. Many of the runs included drive/brake data, as well as other specific tests and tire run-in. The files for Runs 6, 9, 12, and 15 were suitable. Run 6 was used for the analyses shown here.

Data was imported into a Pandas dataframe for manipulation. Pandas dataframes are versatile structures used to represent two-dimensional, typically time series data. They can be easily polled, plotted, or subdivided. Unneeded notes early in the data file were skipped upon import. Initially, all data channels were plotted individually to provide a general look at the structure and signals. This plot is shown in Appendix A and is also included as a separate attached file at higher resolution for reference. Several of the channels showed very little variation over the duration of the test, which helped determine. Certain other channels, namely Pressure (P) and Inclination Angle (IA) showed very clear step regimes. These were used to create subsets of constant values for given parameters, which was helpful for curve fitting other channels. These regions can be seen in the grey-highlighted areas in Figure 3, and the specific constant IA regions are highlighted in Figure 4.

Once the various subset regions were chosen based on these stepped parameters, a closer look showed similar stepping in the Normal Force (FZ) parameter. This was used to further subdivide the data for fitting Pacejka MF models at different vertical loads. These regions are shown in Figure 5. Clear enough data was now available to define the four-term MF model and use the SciPy curve fit tool to determine fits. After some issues of extremely high variance, it was found that the original publication of the Magic Formula recommends a constant $C = 1.3$ for lateral force curves (Blundell, 2015). Setting $C = 1.3$ helped establish a more typical looking Pacejka model fit, though one loading scheme did not fit similarly. The fit models are shown in Figure 6. While the variances for the fit were still high, these curves were enough of a representation of a typical MF model to continue.

First attempts at using SINDy to find dynamical systems in the data were exploratory. The PySINDy package for Python was used to help streamline the process of building several different models to test. First, a PySINDy model was fit to the entire Run 6 data stream. The package worked well but was found to have far too many terms to be a realistic and sparse model. Some improvement was found using the time series step index rather than the test’s elapsed time (ET) variable, but this still appeared to be far too complex. Using adjustments to the PySINDy optimizer and trimming functions yielded similarly poor results.

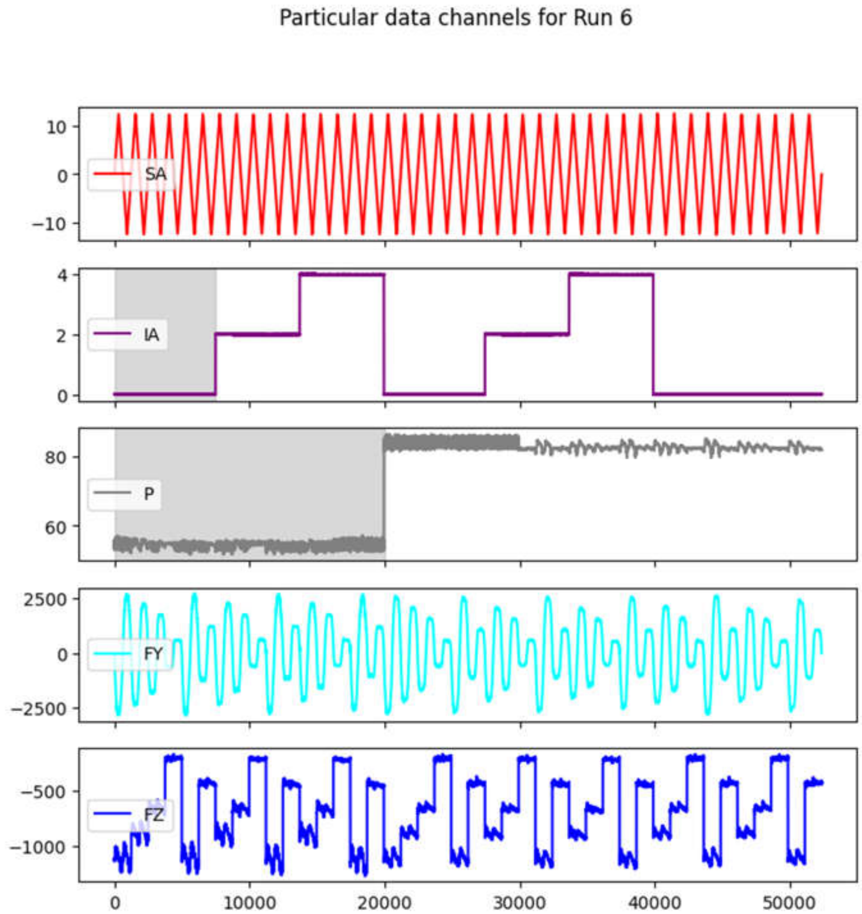

Fig. 3 Data channels of particular interest, showing highlighted regions of constant *IA* and *P*.

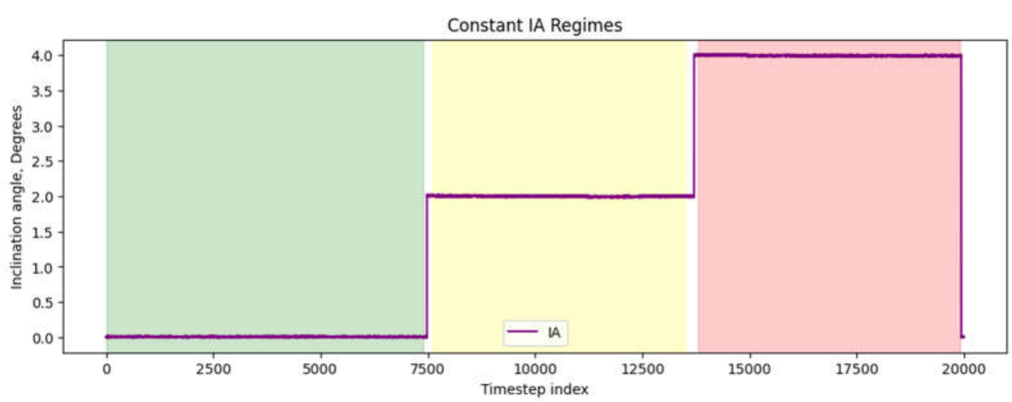

Fig. 4 Constant regions in Inclination Angle at zero, two, and four degrees camber.

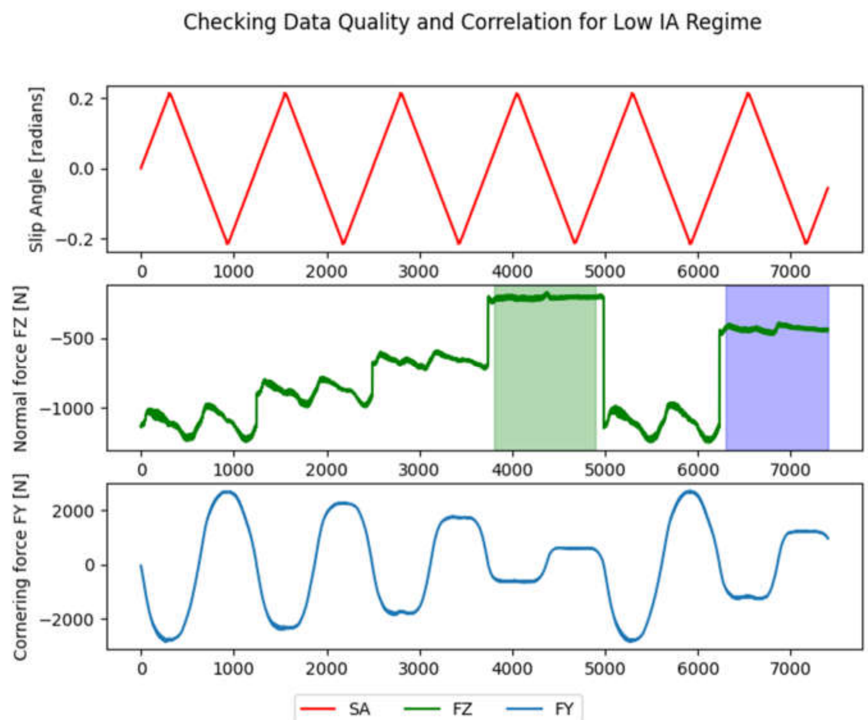

Fig. 5 Regions of constant normal force (*FZ*) at a particular inclination angle.

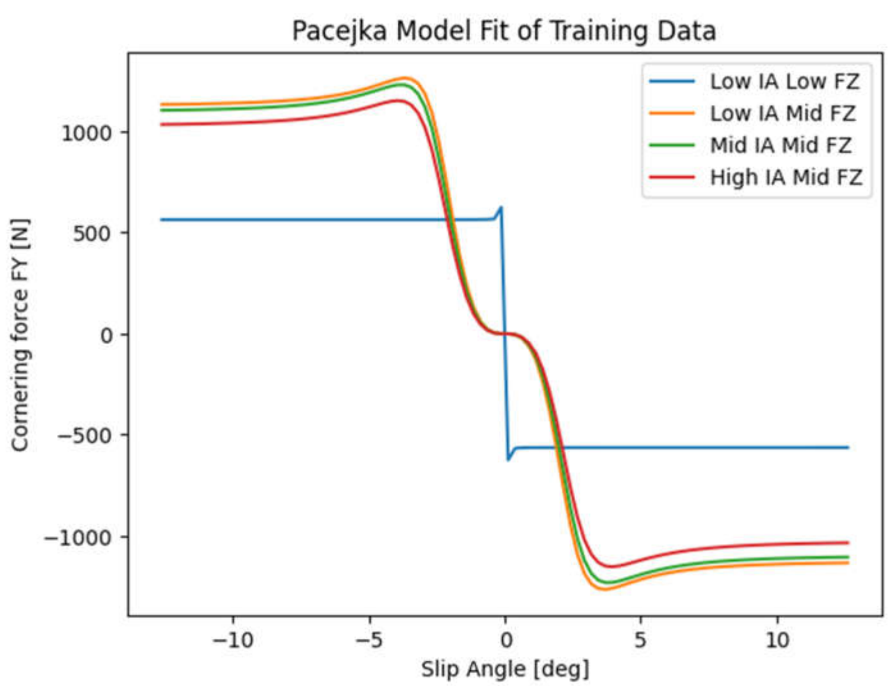

Fig. 6 Four-term Magic Formula curve fits for the four observed loading scenarios.

Another option that PySINDy offers is creating custom library functions. This feature was used to build a library incorporating several nonlinear functions, such as typical trigonometric functions and those used in the four-term Magic Formula. The custom library was saved for use with several formats of the data set. When applied to the whole data set, the results were even more complicated and not sparse.

To make the process more straightforward, trials were run using PySINDy with smaller subsets of the data. A simple set consisting only of the cornering force FY and slip angle SA data streams was used, and a potentially viable model was found of the form

$$(FY') = 0.699\, SA - 2.602\, SA^2 \tag{2}$$

This simple model was plotted in Figure 7 using the SciPy ODEInt integration tool, against a control sweep of slip angle from -12° to +12°. Clearly it did not exhibit the symmetric behavior expected of steering a tire left and right, which is depicted correctly in the Pacejka Magic Formula.

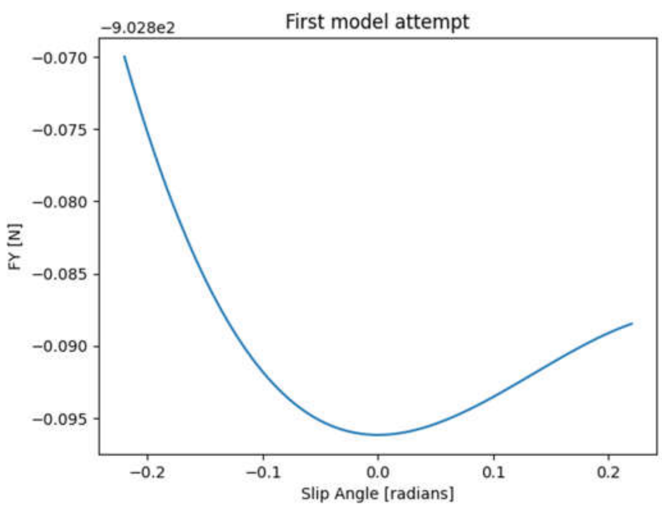

Fig. 7 Initial model found by SINDy on the simplified dataset.

Since the simple subset seemed viable, another model was fit using the previously defined custom library functions. This yielded a model of the form

$$\begin{aligned}(FY)' = {} & 0.555\sin(FY) - 129250.796\sin(SA) + 1531.069\arctan(FY) + 67439.175\arctan(SA) \\ & + -0.708\sin(FY + SA) - 1530.163\arctan(FY + SA) + 61837.177\,(SA) \\ & + -2.575\,(SA^2)\end{aligned} \tag{3}$$

This model is shown in Figure 8, once again using ODEInt, and again it does not conform to the expected behavior. Several other attempts to use the most basic subset of data (just FY and SA) yielded no useful model from SINDy.

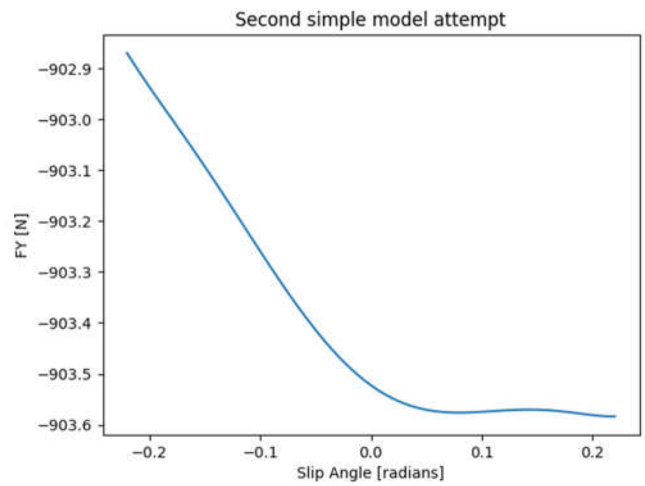

Fig. 8 Model formed using custom library functions.

Another subset of the data with more parameters was then used to see if the SINDy algorithm might work better. In this subset, data streams for slip angle *SA*, inclination angle *IA*, tire pressure *P*, cornering force *FY*, normal force *FZ*, and aligning moment *MZ* were included. Oddly, a similar model to the one shown in Figure 7 was produced, without reference to any of the other parameters. Opening up to the custom library functions produced a very dense model that was not viable.

One last subset of data was created, this time replacing the cornering force *FY* and normal force *FZ* with the normalized cornering force *NFY*. *NFY* is given by dividing *FY* by *FZ*. The plot of these data streams shown in Figure 9 depicts *NFY* varying directly with slip angle, without significant change over the variation of other parameters. This seemed promising as a better candidate for a model fit, and using the custom library functions a model was found of the form

$$(NFY)' = 0.156 \sin(NFY) + -0.157 \arctan(SA) + -0.147 \sin(NFY + SA) \tag{4}$$

This model is plotted and shown in Figure 10. It varies in an opposite fashion from what is expected given the Magic Formula model and therefore is not viable.

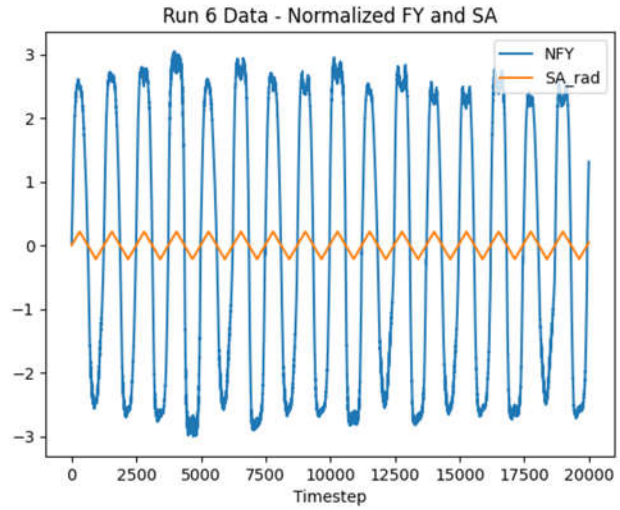

Fig. 9 Normalized cornering force *NFY* seems to vary more directly with *SA*.

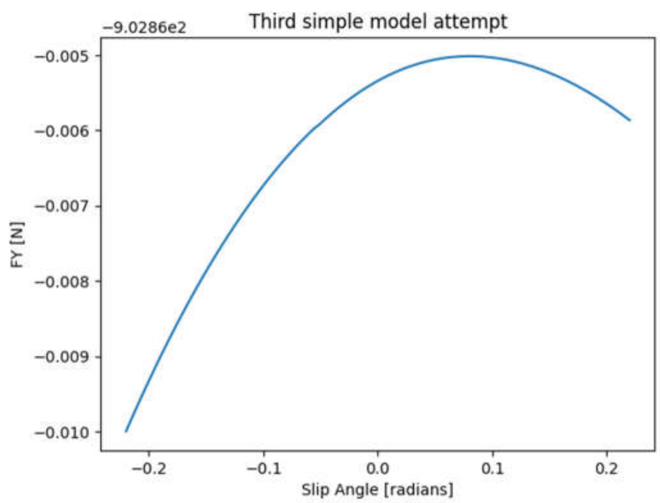

Fig. 10 The dataset using *NFY* did not produce a viable model.

As SINDy had not yielded any useful models to compare to the Pacejka model, an even simpler approach was taken. It's clear that *NFY* and *SA* vary almost directly, a simple linear model was fit to correlate them. This produced a model that did align to the test data but with significant error. An alternative sine fit was created that more closely represented the data, and both rudimentary fits are shown in Figure 11.

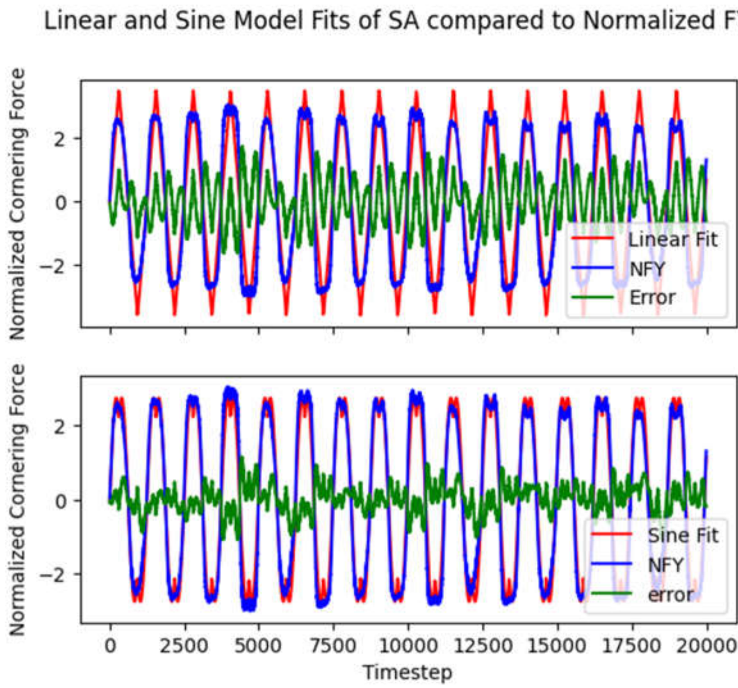

Fig. 11 Basic linear and sine model fits for *NFY*.

Results:

None of the candidate models produced using PySINDy were viable to describe the behavior of the tire tested in Run 6. SINDy failed to produce a model that was comparable to the Pacejka Magic Formula fit of the test data, despite exhaustive adjustments to the choice of data subset or the PySINDy optimizer and custom library functions. Candidate models shown in Figures 7, 8, and 10, are not comparable to the Pacejka model fit in Figure 6. Simplified linear and sine fits of the test data were a better representation of the real behavior in lab testing.

Discussion and Conclusions:

Significant manipulation of the data subsets and parameters was necessary to get particularly useful candidate models from this SINDy algorithm. Even so, none of the models proved to be useful at all. The datasets available from Calspan/TTC are very clean and low noise, so disturbances are not suspected as being a cause of the poor modeling. It is possible a more customized SINDy algorithm, maybe using a different package or one designed specifically for this purpose might find better correlations and therefore candidate models, but many of the explored options should make it flexible enough for our needs.

It is possible that using an even larger dataset, or several different types of run files, might work better to help SINDy find its correlations, but even the data in the Run 6 file has more than 50,000 data points for each channel. It would seem that the work of Hans Pacejka and the nearly worldwide use of his Magic Formula is not without merit. The semi-empirical Magic Formula is a much more effective model for tire behavior than any learned here using sparse identification of nonlinear dynamics.

References:

Blundell, M., Harty D. The Multibody Systems Approach to Vehicle Dynamics. Elsevier, 2015.

Kasprzak, E., Gentz, D. The Formula SAE Tire Test Consortium – Tire Testing and Data Handling. Society of Automotive Engineers, 2006.

Milliken, D. Formula SAE Tire Test Consortium. Milliken Research Associates, Inc. - FSAE Tire Test Consortium. Retrieved April 27, 2023, from https://www.millikenresearch.com/fsaettc.html

H. Pacejka. Tyre and Vehicle Dynamics. Elsevier, 2006.

SAE International Ground Vehicle Standard "Vehicle Dynamics Terminology," SAE Standard J670e, Rev. Jan. 2018.

Appendix A: All data channels for Test Run 6.