Regression Tree Flow Field Forecasting

Kyle A. Caudle * Patrick Fleming † Michael Frey ‡

Key Words: times series, forecasting, multivariate data streams, regression trees

1. Introduction

Flow field (FF) forecasting is a statistical learning methodology used for forecasting that was developed by Michael Frey and Kyle Caudle [7, 8] . It is based on the premise that past associations between history and change are predictive of changes associated with current histories/future changes. FF forecasting as a method of forecasting a univarite time series was shown to be competitive with the more traditional forecasting methods of Box Jenkins ARIMA [2], exponential smoothing [4, 5] and neural networks [9].

FF forecasting has three basic steps.

1. Extract data histories (levels and subsequent changes)
2. Interpolate between observed levels in histories
3. Use the interpolator to step-by-step predict the process forward to the desired forecast horizon

For univariate time series, the interpolator in step 2 was Gaussian Process Regression (GPR) [10]. For the bivariate case we used a Nearest neighbors/pattern matching approach [6] and for the multivariate case (i.e. large dimension) we use regression trees. Regression tree forecasting will be the primary focus of this paper.

Flow field forecasting begins by organizing the historical time series data. For ease of illustration, assume we have a bivariate time series with n observations.

$$\{(X_1,Y_1),(X_2,Y_2),...,(X_n,Y_n)\}$$

We choose a history depth $H = 4$. this results in the following overlapping history segments.

$$\mathbf{s_0} = \{(X_{n-3},Y_{n-3}),(X_{n-2},Y_{n-2}),(X_{n-1},Y_{n-1}),(X_n,Y_n)\}$$
$$\mathbf{s_1} = \{(X_{n-4},Y_{n-4}),(X_{n-3},Y_{n-3}),(X_{n-2},Y_{n-2}),(X_{n-1},Y_{n-1})\}$$
$$\vdots$$
$$\mathbf{s_{n-4}} = (X_1,Y_1),(X_2,Y_2),(X_3,Y_3),(X_4,Y_4)\}$$

*South Dakota School of Mines and Technology, Rapid City, SD 57701, USA
†South Dakota School of Mines and Technology, Rapid City, SD 57701, USA
‡Bucknell University, Lewisburg, PA 17837

From our time series, we arbitrarily choose a set of meaninful predictors from each history segment. These predictors may consist of current and post observations of x and y, lagged values or even functions of current or lagged values. For example,

$$\mathbf{h_0} = \{X_n, Y_n, X_n - X_{n-1}, Y_{n-1} - Y_{n-2}, |Y_{n-3}|\}$$
$$\mathbf{h_1} = \{X_{n-1}, Y_{n-1}, X_{n-1} - X_{n-2}, Y_{n-2} - Y_{n-3}, |Y_{n-4}|\}$$
$$\vdots$$
$$\mathbf{h_{n-4}} = \{X_4, Y_4, X_4 - X_3, Y_3 - Y_2, |Y_1|\}$$

where $\mathbf{h_0}$ is the current history, $\mathbf{h_1}$ is the first previous history etc. For the Closest History version of flow field forecasting (CHFF) as outlined in [6], we would like to know which of the histories is most like $\mathbf{h_0}$. We determine the change in x and y progressing forward from the closest history as a prediction of the next future change in the time series. For large dimensions this approach fails. It often fails because this approach will exceed the computational resources, but it also fails because in high dimensions everything is "far" apart and the concept of "closeness" is ill-defined [1].

2. Regression Tree Flow Field Forecasting (RTFF)

Regression trees as a methodology was developed by Brieman et al. [3] in 1984. A regression tree takes a data set and separates it so as to minimize the total sum of squares. The classification and regression tree (CART) algorithm developed by Brieman et al. [3] uses a greedy approach in that it checks all possible split points for each variable and choose the split point and variable that results in the smallest sum of squares. For regression trees, one needs a response variable in order to determine the split points. Our choice for the response variable are the lag-1 differences. Therefore, prior to building the tree the algorithm first determines the pairwise time ordered differences between the variables.

For example, suppose we have a time series that consists of 3 variables x, y, and z. We start by building the following history matrix:

x_0	x_1	vx	y_0	y_1	vy	z_0	z_1	vz
x_n	x_{n-1}	$(x_n - x_{n-1})$	y_n	y_{n-1}	$(y_n - y_{n-1})$	z_n	z_{n-1}	$(z_n - z_{n-1})$
x_{n-1}	x_{n-2}	$(x_{n-1} - x_{n-2})$	y_{n-1}	y_{n-2}	$(y_{n-1} - y_{n-2})$	z_{n-1}	z_{n-2}	$(z_{n-1} - z_{n-2})$
x_{n-2}	x_{n-3}	$(x_{n-2} - x_{n-3})$	y_{n-2}	y_{n-3}	$(y_{n-2} - y_{n-3})$	z_{n-2}	z_{n-3}	$(z_{n-2} - z_{n-3})$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

Next we generate 3 regression trees, one for x, one for y and one for z. For each tree, the CART algorithm finds the point p and splitting variable j that partitions the plane into two halves so as to minimize the sum of squares of the two halves.

$$H_1(j,p) = \{X | X_j \leq p\} \text{ and } H_2(j,p) = \{X | X_j > p\} \tag{1}$$

$$TSS = \Sigma_{x_i \in H_1(j,p)} (vx_i - v_1)^2 + \Sigma_{x_i \in H_2(j,p)} (vx_i - v_2)^2 \tag{2}$$

where the estimates of v_1 and v_2 can be found by,

$$\hat{v}_1 = \text{ave}(vx_i | x_i \in H_1(j,p)) \text{ and } \hat{v}_2 = \text{ave}(vx_i | x_i \in H_2(j,p)) \tag{3}$$

As an example, suppose the CART algorithm generated the regression trees shown in figure 1. Further suppose that the current history $\mathbf{h}_0$ has $x = 3, y = 4$ and $z = 1$. Furthermore, suppose the current history $\mathbf{h}_0$ has $x = 3, y = 4$ and $z = 1$. Since $x = 3$ which is less than 5, we go left at the first decision point for the vx tree. The next decision point leads us to the right since $x > -2$. Our estimate of vx is therefore equal to 3. Similarly, our estimate vy would be 2 and our estimate of vz would be 5.

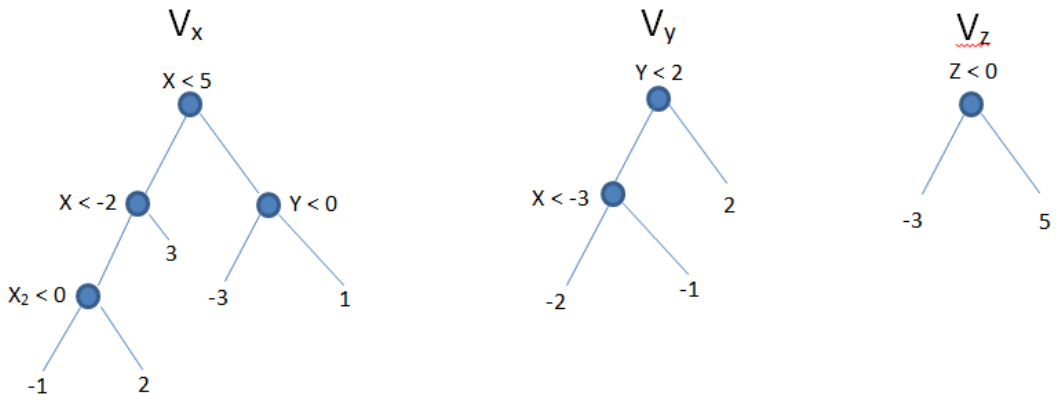

Figure 1: Sample Regression Tree

Knowing the estimated changes in x, y, and z allows us to estimate the new position $(x_{n+1}, y_{n+1}, z_{n+1})$.

3. Oscillators

Systems that oscillate are a particularly interesting application for RTFF. There are numerous applications of oscillators such as electrical systems, populations dynamics and neurons in the brain. Regression trees do an excellent job of identifying the point of oscillation as well as the amplitude and direction of these oscillation. We will show that the oscillator state can be forecasted easily from regression trees.

3.1 Simple Oscillator

We start by presenting an example of a simple oscillator, that is one with a single stable state.

$$x_{i+1} = x_i + V(x_i) + N(0, 0.1) \tag{4}$$

$$V(x) = \begin{cases} 4 & x < 1 \\ -1 & x \geq 1 \end{cases} \tag{5}$$

where $N(0, 0.1)$ represents Gaussian noise with standard deviation 0.1. We start with an arbitrary initial guess and then, using equations (4) and (4) we generate a time series of 1005 observations. The first 1000 observations are used to generate the regression tree and the remaining 5 observations are used for testing. A time series plot of this simple oscillator is shown in figure 2.

The regression tree from this example (not shown) is quite simple, there is just one split at approximately 1. Using the information from time series observation 1000, we follow the regression tree until we come to the leaf. The leaf is then our prediction of vx. As one can see from figure 3 we see that the forecasts provided by this method are quite accurate.

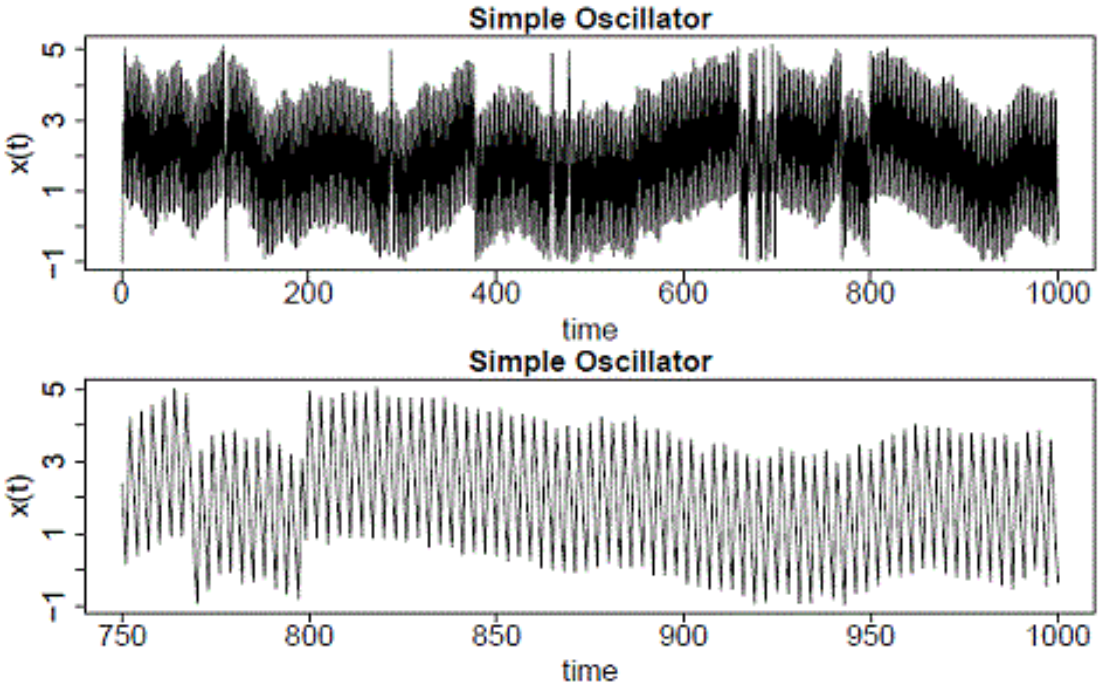

Figure 2: Simple Oscillator

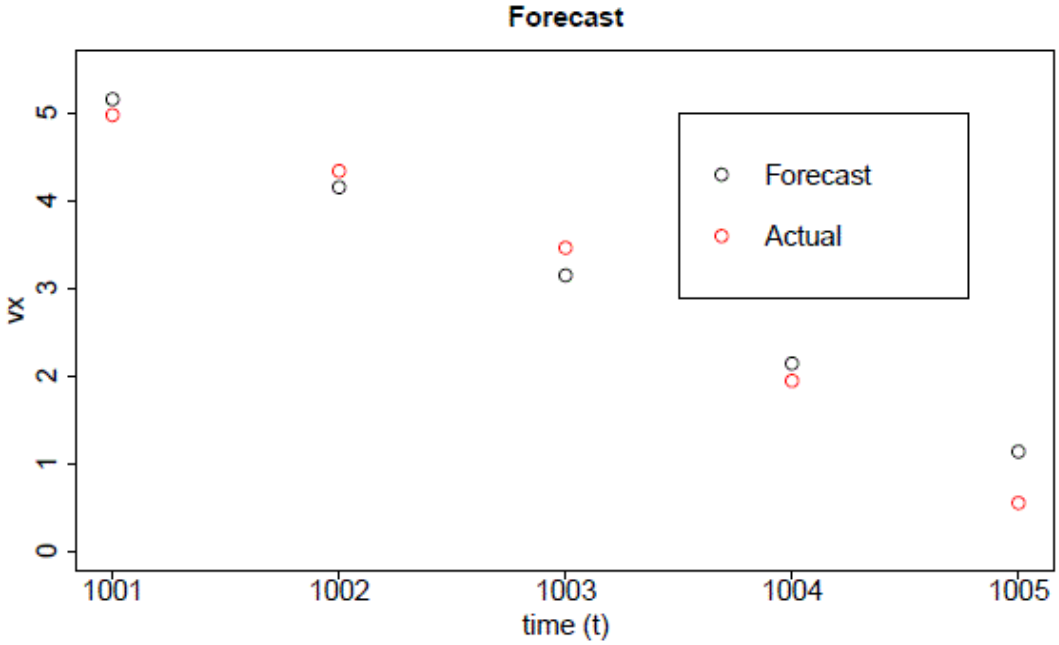

Figure 3: Forecast for Simple Oscillator

3.2 Complex Oscillator

We now provide a more complicated example. For this example, we have a 3 oscillator system where x and y oscillators are slaved to a third (z). The x oscillator is affected by x, z and x lag 2. The y oscillator is affected by y, z and time (t). Thus, for this example we show that this method is able to identify non-stationarities in the time series. A simple block diagram for this system is shown in figure 4.

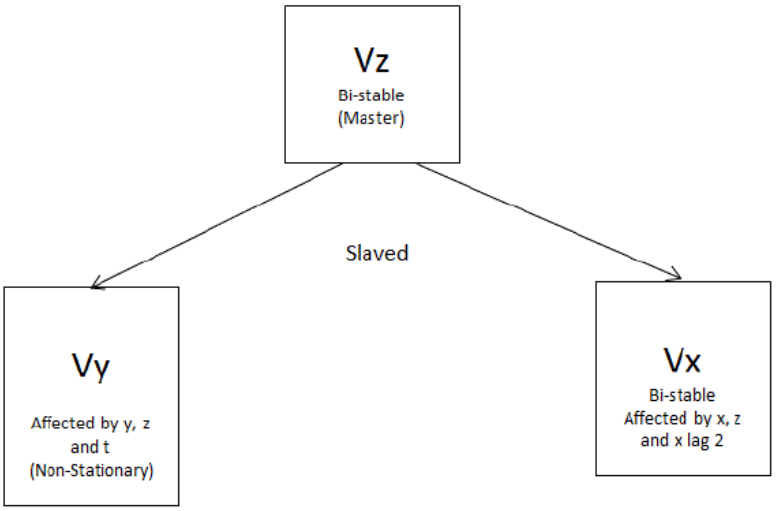

Figure 4: Complex Oscillator

Again, we generate 1005 observations for this oscillating system and reserve observations 1001-1005 for testing. In figure 5 we show a time series plot of this system for all 3 variables. By inspection of figure 5 it is clear that forecasting from this system would be difficult.

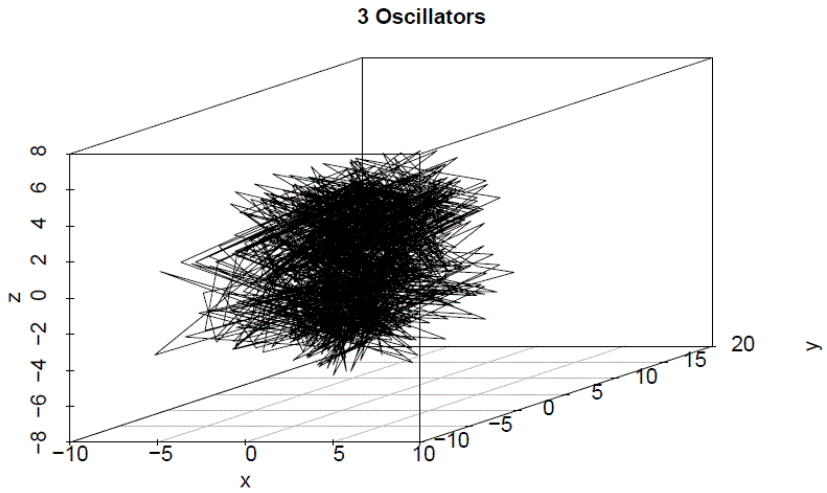

Figure 5: Time Series Complex Oscillator (3D plot)

In figure 6 we can show that individual time series for x, y and z. Since $y(t)$ is non-stationary, we see a change in the time series characteristics at $t = 500$. From the times series plot of $x(t)$, we see that occasionally there is a precipitous drop. This large drop is based on the lagged relationship two time steps back. Again, we create regression trees, one for each variable x, y and z.

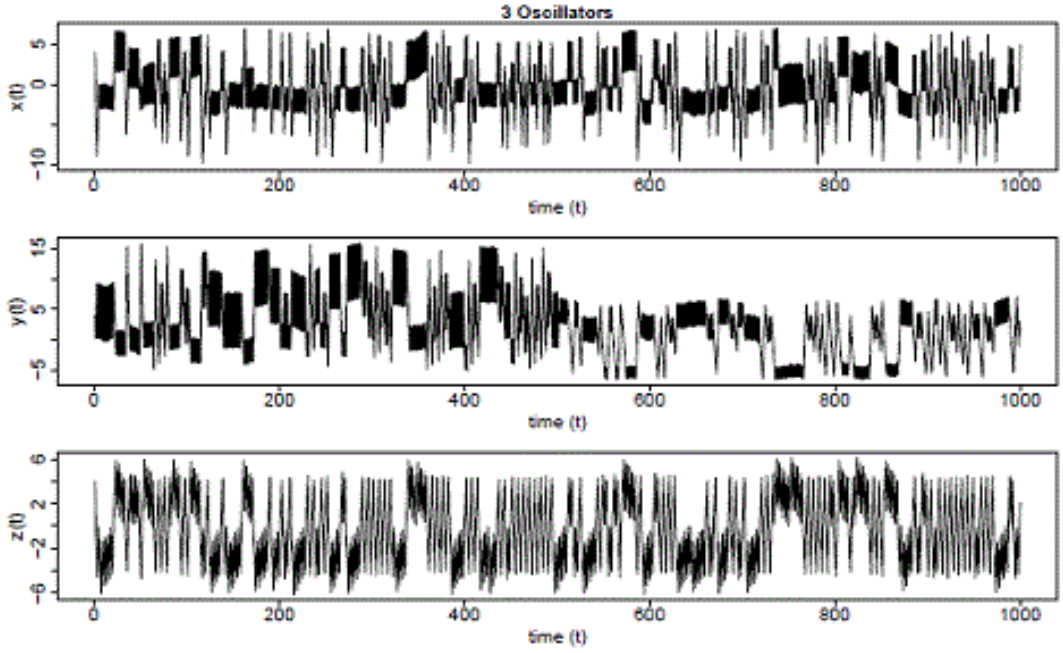

Figure 6: Time Series Complex Oscillator (Separate Variables)

Using the current point we follow each regression tree in order to find the change in each coordinate and from there were determine the new position. We only show the regression tree for vx in figure 7. By inspection of figure 7 we see that one of the decision splits was made on x-lag. Thus, it illustrates the point that any number of useful forecasting variables may be included in your regression tree.

Finally, in figure 8 we show the forecast values and the actual values on a 3 dimensional plot for $t = 1001 - 1005$. It is interesting to note the oscillating nature of this system. The even numbered values are on the right of the plot and the odd values the left of the plot. Forecasting by this method is again quite accurate.

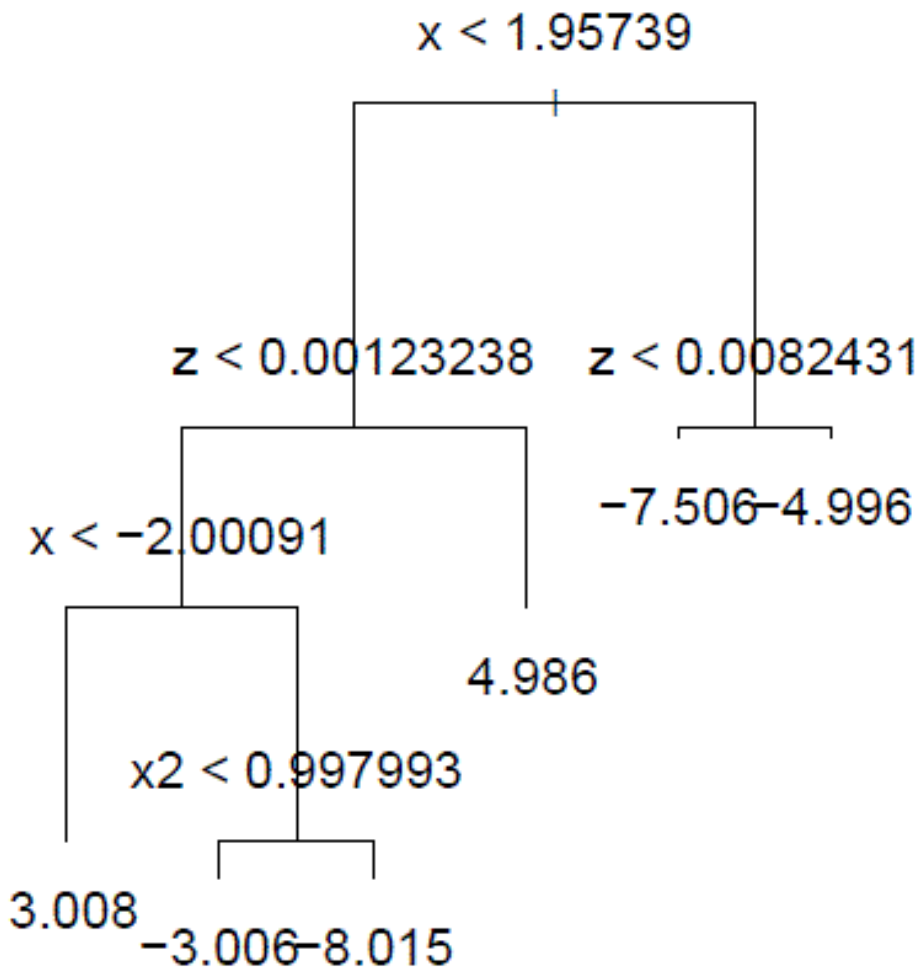

Figure 7: Regression Tree for Vx (Complex Oscillator)

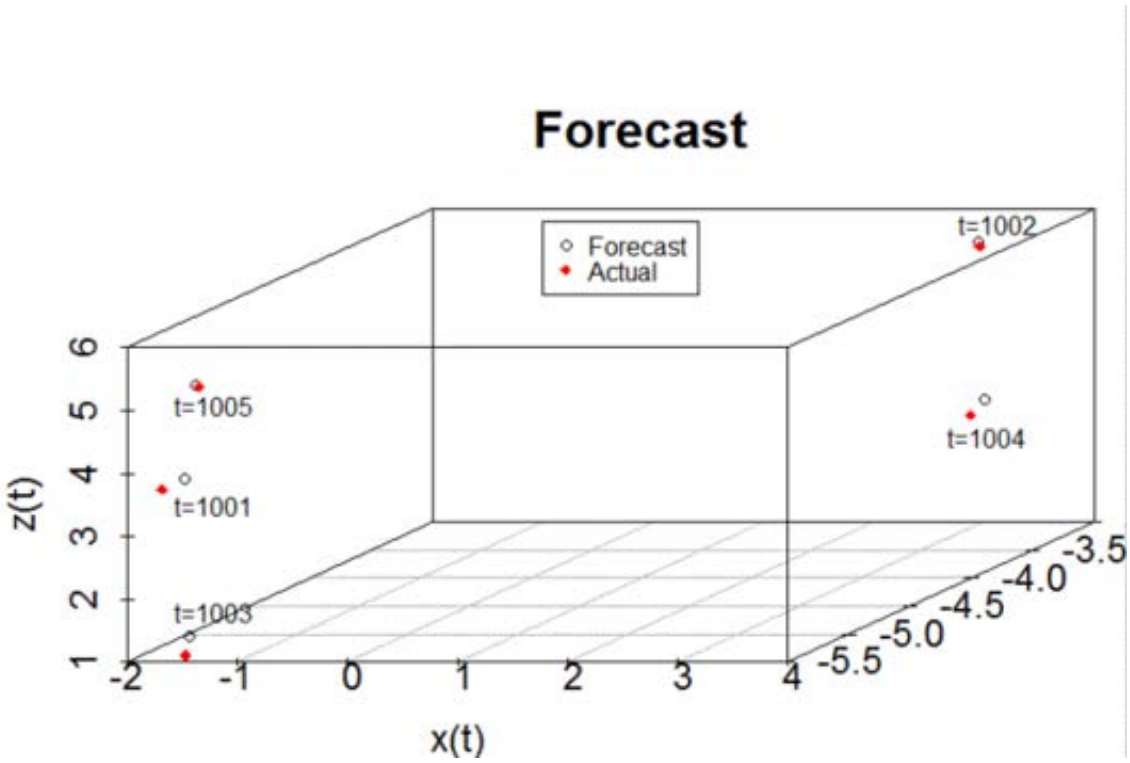

Figure 8: Forecast for Complex Oscillator

4. Final Remarks

This paper shows outlines a method of forecasting that is somewhat different than traditional forecasting methods. First, we forecast the change in position as opposed to the position itself. This is essentially the guiding principle of flow field forecasting in that previous levels and change are predictive of future levels and changes. Although forecasting change in observation is equivalent to forecasting the next (future) observation, this paradigm shift emphasizes that the underlying generative process is a set of stochastic differential equations [8].

Another contribution of this paper is that RTFF is able to handle a very large set of potential predictor variables. Although we did not illustrate the computational efficiency of this method, classification and regression tree (CART) software is very fast and capable of handling literally hundreds of potential predictor variables. If a variable is included in the model that is not a useful predictor, CART will not use this variable as a splitting variable.

Finally, we have shown that RTFF has a very useful application of predicting systems that oscillate. Suppose we have a system which is known to oscillate. this system may have various interactions between oscillators and the oscillators themselves may have multiple states. By building a regression tree from data generated from this oscillating system, we can these time dependent interactions and the oscillating states. Our immediate goal is to see if we can accurately identify the parameters of an oscillating system. We plan to report these findings elsewhere.

5. Acknowledgment

This publication [article] results from research supported by the Naval Postgraduate School Assistance Grant N00244-16-1-16 awarded by the NAVSUP Fleet Logistics Center San Diego (NAVSUP FLC San Diego). The views expressed in written materials or publications and/or made by speakers, moderators, and presenters, do not necessarily reflect the official policies, or organizations imply endorsements by the U.S. Government.

References

[1] Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. When is "nearest neighbor" meaningful? In *Lecture Notes in Computer Science*, volume 1540, pages 217–235. Springer, January 1999.

[2] G.E.P. Box, G.M. Jenkins, and G.C. Reinsel. *Time Series Analysis: Forecasting and Control*. John Wiley & Sons, Hoboken, NJ, 4th edition, 2008.

[3] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

[4] R.G. Brown. *Statistical Forecasting for Inventory Control*. McGraw-Hill, New York, 1959.

[5] R.G. Brown. *Smoothing, Forecasting, and Prediction*. Prentice Hall, Englewood Cliffs, NJ, 1963.

[6] K. Caudle, M. Frey, P. Fleming, and N. Brubaker. Next generation flow filed forecasting. In *JSM Proceedings*, Statistical Learning Section, pages 365–375, 2015.

[7] Michael R. Frey and Kyle A. Caudle. Introducing flow field forecasting. In *Proceedings of the 10th International Conference on Machine Learning and Applications*, IEEE, pages 395–400, 2011.

[8] Michael R. Frey and Kyle A. Caudle. Flow field forecasting for univariate time series. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 6:506–518, 2013.

[9] S. Haykin. *Neural Networks and Learning Machines*. Pearson, New York, 3rd edition, 2009.

[10] C.E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, 2006.