Intelligent Prediction

Randal Allen, Sam Vasta, John Volpi
Lone Star Analysis
Addison, TX
rallen@lone-star.com, svasta@lone-star.com, jvolpi@lone-star.com

ABSTRACT

The Intelligent Predictor is a node available to the user of Analytics OS (AOS) where a filter/prediction method is selected and applied to sensor data to provide predictive analytics and subsequently prescriptive analytics. This paper discusses the research associated with different filtering and predicting methods and their development. Details of each method are provided in the Methods section of the paper. The data used and the criteria applied to rank the performance of each method appear in the Analysis section where the results are presented. Finally, our recommended methods and suggestions for future work are provided in the Conclusion section.

ABOUT THE AUTHORS

Randal Allen has over 20 years of industry experience and has been with Lone Star Analysis since 2006. As Chief Scientist, he is responsible for applied research and technology development. He is a certified modeling and simulation professional (NTSA). He is the co-author of "Simulation of Dynamic Systems with MATLAB and Simulink," 2nd ed. He is also an Associate Fellow of the American Institute of Aeronautics and Astronautics (AIAA). He holds a Ph.D. in Mechanical Engineering from the University of Central Florida, an Engineer's Degree in Aeronautical and Astronautical Engineering from Stanford University, an M.S. in Applied Mathematics and a B.S. in Engineering Physics, both from the University of Illinois (Urbana-Champaign). He also serves as an Adjunct Professor in the Mechanical and Aerospace Engineering (MAE) department at UCF.

Sam Vasta is a Software Engineer and has been with Lone Star Analysis since 2015. He is responsible for the implementation and development of statistical analysis algorithms and tools, with a focus on user interface design and time series forecasting techniques. He holds a B.S. in Computer Science from the University of Texas at Dallas

John Volpi is the Chief Technology Officer (CTO) of Lone Star Analysis and has served in this role since 2004. He is responsible for all technical activities, intellectual property evaluation, and process development. He began his professional career at Texas Instruments as a theoretical systems analyst and evolved into Systems Engineering, where he was named a Senior Member of the Technical Staff. He attended the Illinois Institute of Technology (B.S. Physics) and Michigan State University (M.S. Physics). He also attended graduate classes at Southern Methodist University and the University of Southern California. He has over 30 patents awarded or pending. He has been a member of the IEEE for over 30 years and was named a Senior Member for his work in wireless technologies and his efforts in the development of Intellectual Property. In 2012, John was awarded the Tech Titans Award for Corporate CTO by the DFW Metroplex Technology Business Counsel out of a pool of 4,000 firms.

Intelligent Prediction

Randal Allen, Sam Vasta, John Volpi
Lone Star Analysis
Addison, TX
rallen@lone-star.com, svasta@lone-star.com, jvolpi@lone-star.com

INTRODUCTION

The Intelligent Predictor[1] capability of Analytics OS (AOS) allows the user to implement different sensor data filtering and prediction methods. Our intention is to contrast and compare these methods. However, since the filtering algorithm is consistent for all methods, the methods differ only in their predictive steps.

AOS is a software platform that provides predictive and prescriptive analytics at the edge of the network or in centralized locations based on the business outcomes the customer is looking to achieve. In particular, it is the Intelligent Predictor node which furnishes the predictive analytics. A single Intelligent Predictor node automatically provides predictions at 10%, 50%, and 90% confidence, collectively called a "prediction interval." The points in a prediction interval may be interpreted as "there is an X% chance the data will cross the threshold on or before the predicted date." Individual points within the prediction interval are only provided when the prediction is within the notification time window.

In addition to the prediction interval, the Intelligent Predictor Node also provides a "Gross Estimation" metric. This metric represents a rough estimate of when the threshold might be crossed. Unlike the prediction interval, the Intelligent Predictor will always output a Gross Estimation metric, assuming the Intelligent Predictor has enough information.

This paper serves as a summary of the research that went into developing the Intelligent Predictor node. In the next section, we introduce each filter and prediction method, including those which weren't worthy of full analysis. In the following section, we discuss the data used for analysis, the criteria by which the methods were rated, and the results of our findings. Finally, we conclude with recommendations about when a particular prediction method might be selected and future work to be considered.

METHODS

For each method under consideration, the sensor data is first filtered and then subsequently used for prediction purposes. In this section, we describe the filtering approach followed by a description of each prediction method.

Filtering

The sensor data filter is common across each prediction method, i.e. a one-state Kalman filter (Kalman 1960) with the option to select the sensitivity level: low, medium, or high. For example, if the data is slowly varying from a temperature sensor (thermistor), the user might select a high level of sensitivity to track the signal (see Figure 1 – left panel). As an alternate example, if the data is rapidly changing from a vibration sensor (accelerometer), the user might select a low level of sensitivity to filter the data and extract the signal from the noise (see Figure 1 – right panel).

[1] For more information about the Intelligent Predictor, please see the AOS Release Notes v5.5 available from Lone Star Analysis upon request.

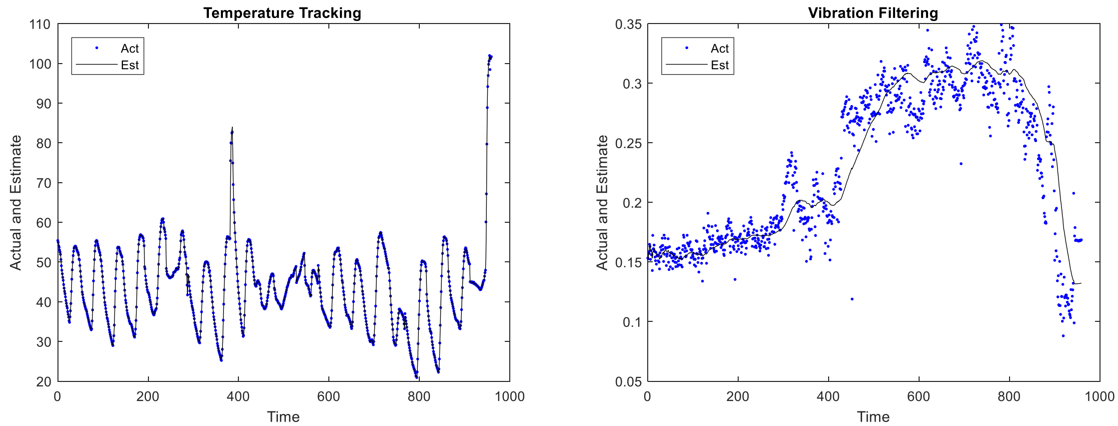

Figure 1 – Temperature data with high sensitivity tracking and vibration data with low sensitivity filtering

Prediction

The prediction methods under consideration are: (1) Linear Regression, (2) Error Trend Seasonality (ETS), and (3) a two-state Kalman filter. Other methods that were considered, but ultimately abandoned were a peak detector, the extended Kalman filter (EKF), and the unscented Kalman filter (UKF).

For Linear Regression, outputs of the one-state Kalman filtered sensor data are used to find the best linear fit over multiple time periods. To obtain useful estimates, G.E.P. Box (Box et al. 2016) recommends using at least fifty observations. Therefore, we begin by using fifty Kalman filter samples to evaluate the performance of linear regression. We also include 100, 200, and 400 samples for an extended analysis of this method.

If the one-state Kalman filtered sensor data has trend, seasonal, and random components, then exponential smoothing extracts the trend and seasonality. ETS is a weighted average of past observations with weights decaying exponentially. While there are different variations on this theme, we elected to implement the additive algorithm, that is, the trend component and the seasonal component are simply added to the error signal. Incidentally, one of the variants is the multiplicative algorithm. However, both the additive and multiplicative algorithms give the same point forecasts but different prediction intervals.

Rounding out the prediction methods is the two-state Kalman filter. In this case, the raw sensor data is not filtered by the one-state Kalman filter. Instead, we implement the two-state to filter and use the second state – which is the derivative or slope of the sensor data – to form the direction by which predictions are forecasted from the first state.

Regarding methods which were initially considered, but subsequently discarded, we have (1) a peak detector, (2) the Extended Kalman Filter (EKF), and (3) the Unscented Kalman Filter (UKF). We have more to say about why we discarded these methods in the Analysis section of the paper.

In the Conclusions section, we discuss the Particle Filter as a candidate for future consideration. We like its implementation as a filter because it makes no assumptions about the process. In fact, we know the Kalman filter is the optimal linear filter. But, if the process is nonlinear, then the particle filter is superior. While the UKF is a balance between the Kalman filter and the particle filter, we still have no knowledge, a priori, of possible nonlinearities. Another attractive feature of the particle filter is its estimation error converges to zero as the number of particles increases. Of course, as in any engineering situation, there's a trade-off. In this case it is the computational effort. Still, with the speed of modern computing, this is less of an impact.

ANALYSIS

In this section, we discuss the data used to evaluate each filter/prediction method, the criteria by which the performance of each filter/prediction was measured, and present the results of our findings.

Data

Data used to perform the analysis was gathered from a live, operating Horizontal Pump System (HPS). We tested the filters/predictors using a total of eighteen data sets: eight occurred during a twenty-day period in November of 2015, and the remaining ten occurred during a two-month period in May and June of 2016. For all data sets, measurements were taken every hour so that one forecasting period represents one hour of elapsed time. The data measures various components of the HPS including bearing and winding temperatures in the motor, pump vibration and suction pressure, overall system health, and other attributes of the system.

Each data set includes some amount of noise which may vary with time. The different sources of data provide a mixture of different characteristics such as seasonality, trends, impulses, and randomness. For example, temperature data was affected by the day/night (diurnal) cycle which creates a (short) seasonal characteristic. Vibration data, however, was not affected by the day/night cycle and is not seasonal, but does contain a significant portion of randomness.

Criteria

Three main criteria were used to test and compare the different prediction strategies. An optimal prediction strategy will minimize all three criteria. These criteria include *false alarms*, *missed predictions*, and *impulse recovery*.

False Alarms

When a prediction strategy produces a forecast, which exceeds the threshold value, it will send an alarm. A false alarm occurs if the observed measurements do not exceed that threshold value within the forecasted timespan. False alarms are misleading and may cause complacency. Therefore, a prediction strategy should produce as few false alarms as possible.

Missed Predictions

A missed prediction occurs when an observed measurement exceeds the threshold value, but no forecast was produced which predicted the exception. Any forecast which predicted the exception within twelve periods leading up to the exception was not considered because such a short forecast is not useful. A prediction strategy should produce as few missed predictions as possible.

Impulse Recovery

A large impulse in measured data will negatively affect predictions for some number of periods after the impulse while the prediction method adapts to, or forgets the impulse. This criterion measures the average number of periods a prediction strategy requires to return to the pre-impulse level, within a 5% margin.

The impulse recovery test uses computer-generated input data. The input data begins with 1000 samples from a random variable, followed by a large impulse which lasts for ten periods, and ends with samples from the original random variable until a 95% recovery is achieved. This test is repeated thirty times, each time using a different standard deviation for the random variable, and a different impulse magnitude.

Results

False Alarm Results

Analysis of the *false alarm* criteria used eight unique forecast lengths ranging from 24 to 1440 periods, or one to sixty days. For all prediction strategies, the frequency of false alarms dropped as the forecast length increased. By 840 periods (thirty-five days) the Two-State Kalman, ETS, Linear (50 History), and Linear (100 History) strategies, each with a *Low* filter sensitivity setting, all achieved 99% or higher average alarm correctness across all eighteen data sets.

Table 1 – False Alarms Performance Summary

Prediction Strategy	Filter Sensitivity	Forecast Length Periods (Days)	Average % False Alarms	
			Low Noise	High Noise
ETS	High	24 (1)	26.96%	6.01%
Two-State Kalman	High	24 (1)	55.59%	69.84%
Linear (50 History)	High	24 (1)	26.81%	27.07%
Linear (100 History)	High	24 (1)	7.74%	6.22%
Linear (200 History)	High	24 (1)	1.62%	28.48%
Linear (400 History)	High	24 (1)	3.51%	9.30%
ETS	Medium	24 (1)	27.19%	22.71%
Two-State Kalman	Medium	24 (1)	57.86%	48.55%
Linear (50 History)	Medium	24 (1)	29.52%	27.63%
Linear (100 History)	Medium	24 (1)	6.69%	6.22%
Linear (200 History)	Medium	24 (1)	1.47%	29.47%
Linear (400 History)	Medium	24 (1)	3.45%	8.92%
ETS	Low	24 (1)	5.64%	7.49%
Two-State Kalman	Low	24 (1)	17.28%	22.00%
Linear (50 History)	Low	24 (1)	14.56%	30.64%
Linear (100 History)	Low	24 (1)	3.40%	27.00%
Linear (200 History)	Low	24 (1)	7.11%	48.33%
Linear (400 History)	Low	24 (1)	4.93%	9.81%
ETS	High	336 (14)	8.10%	8.37%
Two-State Kalman	High	336 (14)	11.94%	22.85%
Linear (50 History)	High	336 (14)	9.35%	3.68%
Linear (100 History)	High	336 (14)	5.59%	6.94%
Linear (200 History)	High	336 (14)	10.02%	18.36%
Linear (400 History)	High	336 (14)	12.39%	12.05%
ETS	Medium	336 (14)	8.71%	2.90%
Two-State Kalman	Medium	336 (14)	12.45%	20.40%
Linear (50 History)	Medium	336 (14)	10.02%	3.93%
Linear (100 History)	Medium	336 (14)	7.99%	7.26%
Linear (200 History)	Medium	336 (14)	28.15%	19.38%
Linear (400 History)	Medium	336 (14)	30.03%	12.16%
ETS	Low	336 (14)	9.06%	3.01%
Two-State Kalman	Low	336 (14)	13.53%	4.73%
Linear (50 History)	Low	336 (14)	7.70%	7.32%
Linear (100 History)	Low	336 (14)	5.55%	10.14%
Linear (200 History)	Low	336 (14)	10.61%	21.08%
Linear (400 History)	Low	336 (14)	14.29%	32.50%

The results in Table 1 compare the average false alarm percentage of each prediction strategy with different combinations of filter sensitivity and forecast length settings. Forecast lengths of 24 periods (one day) and 336 periods (fourteen days) are included which represent short and long forecast length performance, respectively. Note: fourteen days is the baseline for prescriptive analytics and one day is used as a sanity check to be sure filters and predictions are behaving intuitively. The *Average % False Alarms* columns show the probability that an alarm produced by the corresponding prediction/filter sensitivity combination was a false alarm. The *Low Noise* column represents results from data sets which contain a low amount of noise, such as temperature data. The *High Noise* column shows results from data sets which contain a high amount of noise, such as vibration data. For this metric, a lower percentage is better. The best performing prediction strategy/filter combination has been highlighted for each forecast length.

Tests on data with low noise favor prediction strategies which are less reactive such as Linear (200 History) and Linear (400 History). High noise tests favor prediction strategies which are more reactive such as ETS, Linear (50 History), and Linear (100 History). Although the Two-State Kalman strategy is reactive, it might be considered too reactive in scenarios which have high noise or a *Medium* or *High* filter sensitivity. This highly reactive behavior causes the Two-State Kalman to produce many more alarms than other strategies.

Note that the performance of the Two-State Kalman strategy is affected more by the filter sensitivity setting than are the other prediction strategies. Linear regression acts as an additional smoothing step in the Linear (50/100/200/400 History) strategies, and the ETS strategy employs additional smoothing techniques which also act as an additional filter. The Two-State Kalman strategy does not further smooth the data, so the filter sensitivity setting has a larger impact on the performance of this strategy.

Missed Predictions Results

Analysis of the *missed predictions* criteria used the same eight forecast lengths as were used to analyze the false alarms criteria. Again, these forecast lengths ranged from 24 to 1440 periods, or one to sixty days. By only 540 periods, the ETS and Two-State Kalman strategies achieved 94% or higher average prediction correctness across all data sets. By 840 periods, every prediction/filter sensitivity combination achieved 92% or higher average prediction correctness across all data sets.

Table 2 – Missed Predictions Performance Summary

Prediction Strategy	**Filter Sensitivity**	**Forecast Length** Periods (Days)	**Average % Missed Predictions**	
			Low Noise	High Noise
ETS	High	24 (1)	18.92%	43.19%
Two-State Kalman	High	24 (1)	0.00%	0.00%
Linear (50 History)	High	24 (1)	35.74%	54.34%
Linear (100 History)	High	24 (1)	47.61%	62.67%
Linear (200 History)	High	24 (1)	63.73%	73.40%
Linear (400 History)	High	24 (1)	81.66%	86.32%
ETS	Medium	24 (1)	16.67%	42.94%
Two-State Kalman	Medium	24 (1)	0.00%	0.00%
Linear (50 History)	Medium	24 (1)	33.81%	54.81%
Linear (100 History)	Medium	24 (1)	47.61%	62.67%
Linear (200 History)	Medium	24 (1)	62.32%	73.40%
Linear (400 History)	Medium	24 (1)	81.66%	86.32%
ETS	Low	24 (1)	61.17%	62.05%
Two-State Kalman	Low	24 (1)	24.96%	13.89%
Linear (50 History)	Low	24 (1)	38.98%	46.30%
Linear (100 History)	Low	24 (1)	51.77%	60.37%
Linear (200 History)	Low	24 (1)	75.16%	73.40%
Linear (400 History)	Low	24 (1)	81.88%	86.32%
ETS	High	336 (14)	0.00%	5.56%
Two-State Kalman	High	336 (14)	0.00%	0.00%
Linear (50 History)	High	336 (14)	0.00%	2.78%
Linear (100 History)	High	336 (14)	0.00%	5.56%
Linear (200 History)	High	336 (14)	33.33%	5.56%
Linear (400 History)	High	336 (14)	37.93%	30.79%
ETS	Medium	336 (14)	0.00%	2.78%
Two-State Kalman	Medium	336 (14)	0.00%	0.00%
Linear (50 History)	Medium	336 (14)	0.00%	2.78%
Linear (100 History)	Medium	336 (14)	0.00%	5.56%
Linear (200 History)	Medium	336 (14)	16.67%	5.56%
Linear (400 History)	Medium	336 (14)	21.26%	30.79%
ETS	Low	336 (14)	0.00%	5.56%
Two-State Kalman	Low	336 (14)	0.00%	0.00%
Linear (50 History)	Low	336 (14)	0.00%	2.78%
Linear (100 History)	Low	336 (14)	0.00%	2.78%
Linear (200 History)	Low	336 (14)	33.33%	5.56%
Linear (400 History)	Low	336 (14)	37.45%	42.86%

Table 2 compares each filter/prediction strategiy over 24 periods (one day) and 336 periods (fourteen days) in detail.

The *Average % Missed Predictions* columns show the probability that, given the data has crossed a critical threshold, the corresponding prediction/filter sensitivity combination failed to predict the event. A lower percentage indicates better performance for this metric. The best performing strategy/filter sensitivity combination has been highlighted in each forecast length. Many prediction strategy/filter sensitivity combinations performed perfectly in some cases. This does not guarantee that these prediction strategies will perform perfectly in similar scenarios.

In each forecast length scenario, the Linear (50 History) with a medium sensitivity filter outperforms the other Linear prediction/filter sensitivity combinations. In general, the Linear (200 History) and Linear (400 History) strategies performed notably worse than other strategies. This is because the longer history size attenuates any quick changes in the data's trend. The linear regression strategies cannot adapt quickly enough to make a valid prediction.

In each scenario (low noise/high noise), the Two-State Kalman strategy with a *High* or *Medium* filter sensitivity correctly predicted every event. This is likely because the higher filter sensitivities remove a smaller amount of noise. The resulting filtered data will still contain enough noise to cause rapid changes, which results in the Two-State Kalman prediction strategy forecasting an imminent critical event.

The properties of the Linear (400 History) strategy which resulted in favorable performance during the *false alarms* test simultaneously resulted in poor performance in the *missed predictions* test. Additionally, the properties of the Two-State Kalman/High sensitivity combination which resulted in poor performance in the *false alarms* test simultaneously resulted in favorable performance in the *missed predictions* test. For these reasons, it is important to consider both criteria together when evaluating a filter/prediction sensitivity combination.

Impulse Recovery Results

The results in Table 4 show the average recovery time for each prediction strategy over thirty impulse recovery tests. The *Prediction Strategy* and *Filter Sensitivity* columns describe which combination of prediction and filter strategy were used, respectively. The *Periods (95% Recovery)* column describes how many periods after an impulse each prediction/filter combination required before reaching a 95% recovery. Lower values are considered better, and the lowest valued row in each *Filter Sensitivity* group has been highlighted.

Table 4 – Impulse Recovery Performance Summary

Prediction Strategy	Filter Sensitivity	Periods (95% Recovery)
ETS	High	52
Two-State Kalman	High	26
Linear (50 History)	High	51
Linear (100 History)	High	64
Linear (200 History)	High	119
Linear (400 History)	High	248
ETS	Medium	51
Two-State Kalman	Medium	43
Linear (50 History)	Medium	70
Linear (100 History)	Medium	65
Linear (200 History)	Medium	119
Linear (400 History)	Medium	248
ETS	Low	67
Two-State Kalman	Low	61
Linear (50 History)	Low	134
Linear (100 History)	Low	128
Linear (200 History)	Low	127
Linear (400 History)	Low	246

In these tests, the Two-State Kalman Filter strategy performs best for each filter sensitivity setting, and recovers faster with lower sensitivity settings. This is because a prediction strategy can only begin recovery once the data it receives returns to pre-impulse values. The lower sensitivity settings produce smoother output and so they will take more time to return to the pre-impulse levels.

The ETS, Linear (200 History), and Linear (400 History) prediction strategies appear to recover at around the same rate regardless of the filter sensitivity. In the case of ETS, this is most likely a result of the prediction strategy further smoothing the data after the Kalman filter. The data was smoothed enough from the prediction strategy that any smoothing the Kalman filter provided had little effect on the final recovery time.

Similarly, the Linear (200 History) and Linear (400 History) have such large history sizes that linear regression created a smoothing effect similar to ETS. This would also explain why the Linear (50 History) and Linear (100 History) were affected more by the filter sensitivity.

Omissions

The objective of implementing the peak detector was to take the linear regression algorithm and improve its performance with seasonal (regular cycle lengths) and cyclic (irregular cycle lengths) data. The peak detector performs well when the data in a monotonically increasing sinusoid. However, the peak detector performs poorly in almost every other scenario – it either creates very steep-sloped predictions or stops changing its prediction for large periods of time which makes it appear to be broken. Another issue is that the peak detector has more parameters than our other strategies making the setup cumbersome. Furthermore, the results are difficult to interpret.

The objective of implementing the Extended Kalman Filter (EKF) and/or the Unscented Kalman Filter (UKF) was to improve upon the linear Kalman filter to handle nonlinearities. However, we were led to discard both methods because we don't know the process a priori, therefore we have no information by which to model any nonlinearities.

One method still under consideration is the particle filter. The reader will find a discussion of this in the Future Work portion of the following section.

Finally, we would be remiss to omit any discussion of artificial neural nets (ANNs). One reason we don't consider ANNs is because we believe ANNs to be a black box (e.g. hidden layers). At Lone Star Analysis, our philosophy is to present customers with a glass box so they can see exactly what is going on inside our tools. Technical reasons to dismiss ANNs are because the guidance for architectural implementation varies too widely. For example, too few neurons lead to under-fitting; whereas too many neurons lead to over-fitting. The best approach is to try different architectures with a different number of layers and different numbers of neurons to discover which works best for each individual case by numerical investigation. Alas, there is no generalization to this approach.

CONCLUSIONS

Recommendations

As mentioned previously, it is important to consider all criteria when evaluating a filter/prediction (and sensitivity) combination.

We've shown that Linear Regression techniques perform well when it is desired to omit false alarms from low noise (temperature-like) data, while results were mixed between Linear Regression and ETS in the case of high noise (vibration-like) data.

To avoid missed predictions, the Two-State Kalman filter performed the best over short period predictions and performed the best along with other strategies, i.e., ETS and Linear Regression, over long period (14-day) predictions

To recover quickly from sensor impulse signals, the Two-State Kalman filter performed the best overall.

These results appear to be independent of filter sensitivity setting (low, medium, or high). Therefore, in general, the user could use Linear regression to avoid false alarms, and use the Two-State Kalman Filter to avoid missed predictions and to recover from sensor impulses quickly.

Future Work

Earlier in this paper, we mentioned the particle filter. We have implemented it and tested it with favorable filtering qualities. We will investigate replacing the one-state Kalman filter with the particle filter for sensor data filtering. This would alleviate the need for the user to tune the Kalman for low, medium, or high sensitivity. Additionally, there is a possibility of using the particle filter for prediction. This is an area of research and development for us.

REFERENCES

Box, G.E.P. et al. (2016). Time Series Analysis Forecasting and Control, 5th ed., Wiley
Gelb, A. (1974). Applied Optimal Estimation, Technical Staff, The Analytic Sciences Corporation
Hyndman, R.J. and Athanasopoulos, G. (2014). Forecasting: Principles and Practice, otexts.com
Jategaonkar, R. (2006). Flight Vehicle System Identification: A Time Domain Methodology, AIAA
Kalman, R.E. (1960). A New Approach to Linear Filtering and Prediction Problems
Simon, D. (2006). Optimal State Estimation: Kalman, H-infinity, and Nonlinear Approaches, Wiley
Zarchan, P. (2005). Fundamentals of Kalman filtering: A Practical Approach, 2nd ed., AIAA