Dynamic Harmonic Regression

Yichen Dong

# Abstract

In this paper we take a look at a method of forecasting seasonal data called Dynamic Harmonic Regression. We will pull mainly from (Young, Pedregal, Tych 1999) for the theoretical framework behind DHR. We will then look at (Trapero, Kourentzes, Martin 2015) for comparisons between DHR and other methods such as Seasonal Naïve and Exponential smoothing in a state space framework. Lastly, we will look at using CAPTAIN toolbox in MATLAB in order to model seasonal data using DHR.

# Introduction

Harmonic Regression is a class of methods that are used to model data with periodic elements. This could be yearly sales data or daily electricity usage. Such elements are likely to have a seasonal or cyclical component that is hard to capture with a simple GLM (generalized linear model). “Harmonic” in this sense designates the sinusoidal properties of the data, which is a term taken from physics.

There are many practical reasons for why we would want to practice harmonic regression. One good example is the short-term solar irradiation forecasting paper by Trapero, Kourentzes, and Martin (2015) [1]. In their paper, they model solar irradiation in the short term (1-24h). The benefit to this is that an error in the forecasted solar irradiation can lead to high costs. This is because electric companies and transmission systems operators frequently need estimates of electricity production and demand for the next 24 hours.

Another good example is mentioned in Hyndman, Koehler, Snyder, Grose [2]. The problem that they bring up in their paper is the need for a model to handle supply chain management, where increases in demand needs to be met with an increase in inventory. This needs accurate forecasting, since goods often have a manufacturing lead time of a few weeks to a few months. While the coronavirus is anything but seasonal, we can see what happens when supply does not meet demand in the case of masks.

One last example is in (Zavala and Messina, 2016) [3] in which they talk about using DHR to forecast wind power generation. They found that they could use this for a short-term forecasting timescale of a few seconds to a few hours.

## Harmonic Regression Background

One of the oldest and most well known techniques for extracting signals from harmonic data is the X-11 and X-12 ARIMA methods. The most current version is X-13ARIMA-SEATS, which can be downloaded here: <https://www.census.gov/srd/www/x13as/x13down_unix.html>. This software is used by the U.S. Census Bureau to adjust for seasonal impacts in the data.

The core concept behind these algorithms is the term ARIMA, or “Autoregressive integrated moving average”. [4] “Autoregressive” means that the variable is regressed on a transformed lagged version of itself. This means that the model will capture historical information about the variable to predict its current value. The “Moving Average” indicates that the normal error term in a GLM is replaced with a linear combination of error terms that have occurred in the past and the present. “Integrated” means that the data points are replaced with the difference between the value at present and the value at present-lagged time.

Most of the regression for harmonic series depends on the decomposition of the data. [5] What this means is that the data is split into several components- usually the trend component, the cyclical component, the seasonal component, and the “noise” component at time T. Ways to “deseasonalize” the data include the Fourier seires, high order polynomial models, cosine function models, and Gaussian models.

We will not spend too much time on methods other than DHR- we should just be aware that they exist. One of the papers that we will discuss compares the performance of the DHR model against other algorithms.

## Dynamic Harmonic Regression

Dynamic Harmonic Regression is formulated within a stochastic State Space setting- it is specifically a formulation of the “Unobserved Component” (UC) models. DHR uses the methodologies of recursive estimation, specifically the Kalman Filter, and Fixed Interval Smoothing. Kalman Filters have good applications to estimating time variable parameters, as well as being able to be extended to forecasting, backcasting, smoothing, and signal extraction. Fixed Interval Smoothing can be used for optimal signal extraction, smoothing, and interpolation over gaps in the data.

For people like me who don’t intuitively know what a Kalman Filter is, the Wikipedia article provides a good background [6]. In essence, it’s a two-part model. The first part is a “prediction” phase of the model, where the Kalman filter gives its best estimate of the current state variables given the last estimate that it has “observed” so far. Once it is able to observe the next outcome, it then uses that outcome to adjust its previous prediction of the state variables. This process continues recursively until the end of the data is reached.

According to the Young paper [7], the DHR Model can be formulated as a special case of the univariate UC model:

“captures the influence of a vector of exogenous variables ”, and is a stochastic perturbation model. is usually defined as a normally distributed Gaussian set with a mean of zero and variance

Usually, the DHR model is simplified down to just these terms:

The seasonal term and the cyclical terms are very similar, but defined slightly differently. The seasonal term is

Where a, b, are all stochastic TVPs, ω is the harmonic frequencies associated with the seasonality, and f are the frequencies associated with the cyclical component. We can consider the trend component “T” as an intercept parameter for Dynamic Harmonic Regression.

We can define as a two-dimensional stochastic state vector where is the changing level and is the changing slope of the TVP. We can then describe the stochastic process

Where (The 1 comes from the intercept T) and

in this case is the white noise term.

We can then aggregate these matrices and combine them with our original DHR equation to get

For n = 2R, F is a n x n diagonal block with blocks from the matrices defined above, G is defined similarly, and is a 1xn vector that relates the scalar observation to the state variables in the state equation, and is a n dimensional vector containing the zero mean and the white noise input vectors .

## Process for estimation of the Time Variable Parameters of DHR

Having defined the DHR equations, we can now use Kalman filters and Fixed Interval Smoothing in order to estimate the TVPs.

We have for the Forward pass filtering equations:

Prediction:

Correction:

And for the backward pass smoothing equation (operating from the end of the sample set to the beginning):

With .

is the n x n Noise Variance Ratio, and the n x n matrix are both defined as follows:

here means the error covariance matrix associated with the state estimates, and the NVR matrix is assumed to be diagonal.

For the Forward Pass Kalman filter portion of the problem, we can think of (1) as the prediction of the next point given the current point and the state transition matrix, which contains our best estimates of where the point would go to at time t+1. In addition to just calculating the state estimate, the covariance matrix P is also estimated in (2). This covariance can be thought of as a measure of the estimated accuracy of the state estimate. In the correction part of the algorithm, we can think of (3) as the updated prediction after we have “observed” the next point. Since the observation has a measure of inaccuracy within it due to the noise, we use it in order to update our prediction of the current point in order to capture what might be the “true” observation. This is done using our prior covariance matrix and our matrix. This matrix was previously defined as a vector that relates the observation to the state variables. Basically, the maps the observation to our trend, cyclical, and seasonal components, and the covariance matrix determines how much that will change our prior predicted variable. After the state prediction is corrected, we again use and our prior covariance matrix to determine what our posterior covariance matrix should be.

The Backward pass smoothing equations part is a little less easy to understand. According to Young et. al., it is based on “combining Bryson and Ho’s (1969) recursion for the Lagrange multipliers with the state update recursion of Norton (1986). We can think of “N” as the final time index, and the as the prediction of x given the observations from time t to N. can be thought in a similar manner, except it is the prediction of the covariance matrix instead. I’m honestly not too sure what to make of . It might be a smoother term for the noise variance in the model and data?

If it helps, the dynamic harmonic regression algorithm could be thought of as an example of a forward-backward algorithm, as described on Wikipedia [8]. The three main steps are to 1) compute forward probabilities, 2) compute backward probabilities, and 3) compute smoothed values. This fits pretty nicely with what the DHR model is doing.

## Optimization of the NVR matrix Q for DHR

Since our forwards-backwards smoothing equation needs estimates for the NVR matrix Q, we look at a method of obtaining those estimates. The most common method of optimization is optimization using Maximum Likelihood. However, Young criticizes this method, because convergence to the optimum could be slow.

Young’s method for optimizing the hyperparameters is based on spectrum analysis in the frequency domain based on Fourier transforms. A Fourier series “is a periodic function composed of harmonically related sinusoids, combined by a weighted summation” [9]. Based on this, he goes on to note that the frequency response is

This can then be extended to the case . In this case, if we model the TVP of the sine and cosine terms as an IRW (an Integrated Random Walk, where the previously mentioned F and G components ) with equal variance (), we would get

Young posits that the pseudo spectrum of the full DHR model can be represented by

Where R is the number of different frequency components in the model.

Based on this definition, we would need to estimate the best value of the variance parameters . Young initially talks about minimizing a least squares object function J:

With , as the empirical spectrum, and having T distinct frequencies. However, a non-linear logarithmic objective function leads to improved runtimes and estimation results:

This objective function does require non-linear optimization; however, the result from the previous linear optimization can be used as a good starting point.

## The DHR Algorithm

1. Estimate an AR(n) spectrum of the observation process and its associated residual variance .
2. Optimize the Linear Least Squares estimate for the NvR parameter

1. Optimize the non-linear least squares estimate for NVR:
2. Use the NVR estimates from step 3 as the Q matrix for the forward pass Kalman filter and the backward pass (FIS) smoothed estimates of the components of the DHR model.

## Comparison of Optimization techniques for DHR

Young had compared the different techniques to optimize the NVR parameters on the Airplane Passengers dataset (which we will be walking through in the example to follow). He found that the non-linear method converged much faster than the constrained linear Maximum Likelihood method. He found that the unconstrained ML optimization does not converge at all if all the NVRs are unconstrained. Even constraining a couple of the NVRs (Young constrained the harmonic components of periods 4 and 2.4 months by trial and error), it still takes much longer than the other methods to converge.

Besides the algorithm speed, Young also found that the performance of the model improved in the non-linear case vs. the constrained ML case, based on improved innovations variance and some statistical tests. However, performance was slightly worse than the “unconstrained” ML case- however, it should be noted that this was a time intensive effort to get the ML to converge.

## Comparison of Dynamic Harmonic Regression with other Harmonic Methods

In the previously mentioned example about solar irradiation, they used DHR in order to complete the task of forecasting for the next 1-24 hours. They also compared it against other models that might be used for harmonic regression.

The first comparison that they made was against a “Persistence model”. This is a very simple model that says the next observation is equal to the current observation, with the next observation being adjusted each time with the current observation’s error. So if the previous observation was 4, the current is 5, the next will be 6 (depending on the alpha assigned to the error). The equation is as follow:

This can also be extended to a longer timeframe, so that the observation can be made equal to the observation 24 hours prior instead of just one hour prior. This will be called the seasonal Naïve model, and will be used as the base of comparison for all the other models.

The second one that they compared is “Exponential smoothing in a state space framework”. This accounts for trend and seasonality as well. This is also discussed in the Hyndman et. al paper (2000). The math is a little bit complicated to explain in the small space here, but the reader is welcome to read that paper.

The last one is the ARIMA model that we had talked about previously. They write the model as

Where is an observable time series (solar irradiance) and is a white noise process having mean zero and variance . The backward shift operator is denoted by . The non-seasonal Autoregressive and Moving Average operators are defined by ϕ(B) and θ(B) polynomials of order p and q respectively. D denotes the order of differencing that is required to make the time series stationary:

In the paper, they compared these four modeling techniques on a sample of solar irradiation time series data. The data had solar irradiation measurements every minute, and consists of 26,280 observations aggregated into hourly segments.

The method they used to compared the models was to compare the rMBE (relative Mean Biased Error) and the rRMSE (relative Root Mean Squared Error).

And

Where , ,

The authors of the article found that the simple Naïve method performed the worst, with the highest bias and squared error out of all the models. This makes intuitive sense, since the Naïve model would not capture any of the seasonality effects.

Interestingly, ARIMA also performed pretty poorly, with high bias for both of the variables modeled (Global Horizontal Irradiance and Direct Horizontal Irradiance). It also exhibited high square root error for DNI. ETS (the exponential smoothing method) performed better than the simple Naïve, but it could not perform better than the seasonal Naïve model. The DHR model was the one that performed better on both GHI and DNI measures. Of course, DHR is not perfect, showing higher rMSE for GHI than the ETS or S.Naive model did, but that does not prevent DHR from being a really good tool in the seasonal prediction toolbox.

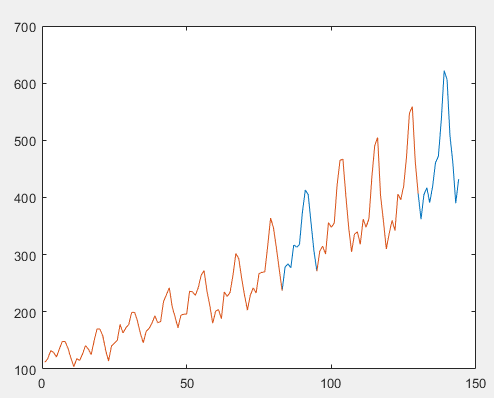
In addition, Young states that after adding the cyclical component of the DHR model, it was the only model to capture the longer 51-month trend.

## DHR in MATLAB

DHR in MATLAB is surprisingly easy to perform using the Captain’s Toolbox [10]. After installing the Captain’s Toolbox and pointing it to the correct paths, you can type in “captdemo” to get a pretty good overview of all the functions that it has. For our purposes, we will be looking at the “TVPMOD demos” → “DHR air passengers” demo.

Since it is pretty simple to install, I will not spend too much time on it. However, for those of you who don’t have MATLAB or who don’t want to install Captain’s Toolbox just yet, I can give a brief overview of exactly what DHR in Captain’s Toolbox does.

Our sample dataset is the count of passengers (in thousands) per month from 1949-1960. We can see a clear seasonal trend in the data on top of the linear trend:

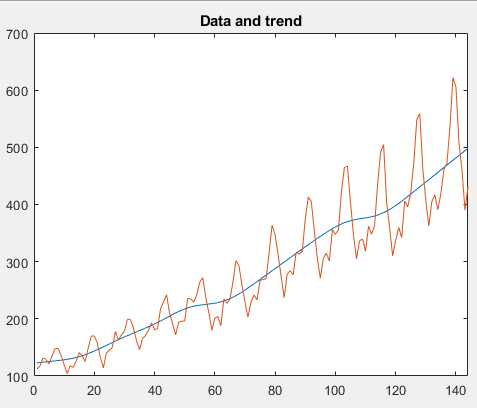


The blue lines are where they replaced the values with Null values. The software writers did this because they wanted to show the accuracy of the DHR method on interpolation as well as extrapolation.

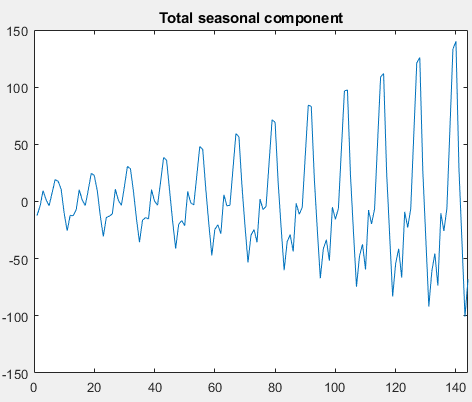
The authors then created a trend component and 5 harmonics, initialized TVP (Time Variable Parameters) to 1, and specified the order of the AR spectrum as 16.

The authors then estimated the model hyper-parameters with a very simple command: nvr=dhropt(y, P, TVP, nar). P is the trend component with the 5 harmonics and nar as the AR spectrum.

The next command returns the components of the DHR model using the newly optimized hyperparameters: [fit, fitse, trend, trendse, comp]=dhr(y, P, TVP, nvr); You can see the trend appears to capture the underlying linear trend pretty well:



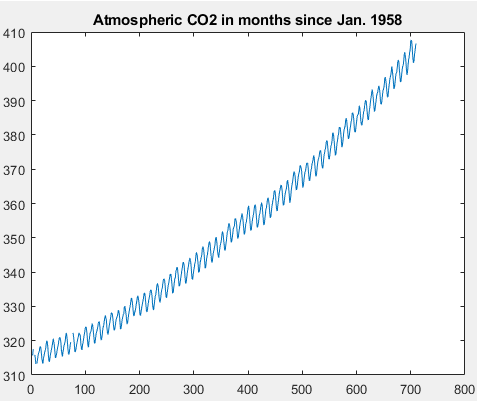
You can also see the seasonal component “comp”:



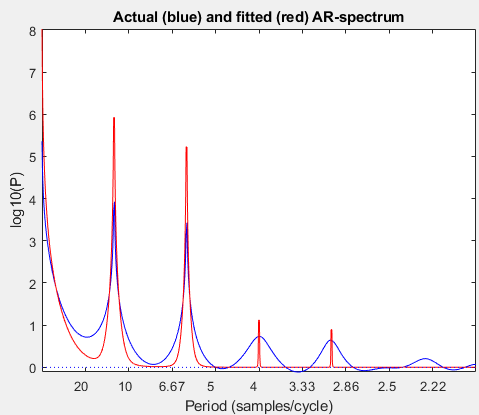
The model pretty accurately interpolates and extrapolates the data. As you might expect from an example someone put into the demo for their software package. This example can also be seen online [11].

## Analysis in MATLAB

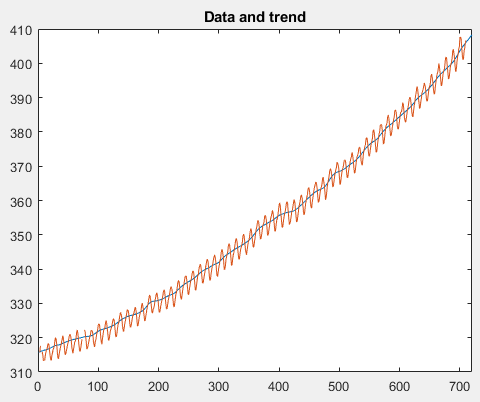
The first dataset that I will look at is the “Carbon Dioxide levels in Atmosphere”, found here: <https://www.kaggle.com/ucsandiego/carbon-dioxide/data>. This contains monthly atmospheric Carbon Dioxide data as measured from the Mauna Loa Observatory. It contains 720 observations, from 1958 to 2017. As we can see, the data seems to have a highly seasonal pattern to it, as well as a clear linearly increasing trend.



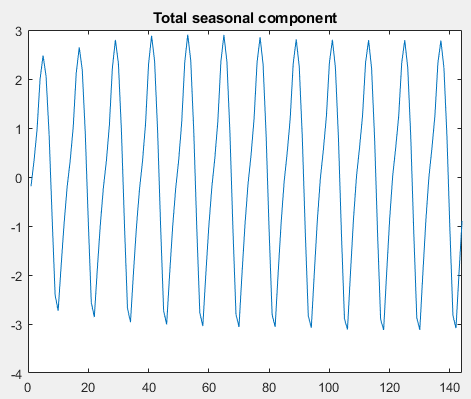
At first, I will do a test run with most of the data intact. The only missing values will be the values that are already missing from the dataset. It seems like using the same parameters as the example dataset, we obtain a pretty good fit on the AR-spectrum. (P= [0 12 6 4 3 2.4], nar =16, and using an IRW model type for TVP).



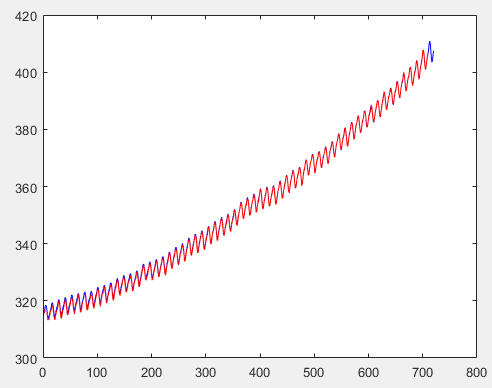
The linear trend seems to be captured pretty well:



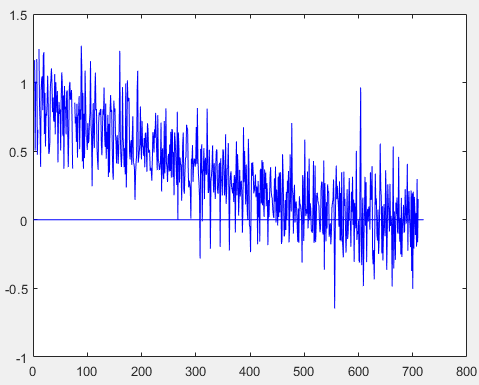
In contrast to the demo, the seasonal component of this data does not seem to be increasing:



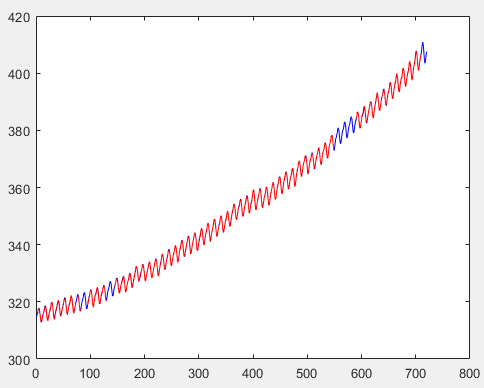
We can see that the fitted data is almost right on top of the actuals



There does seem to be some trend in the residuals, but overall the residuals are all pretty small:



If we make some of the data missing as in the demo, we see that the interpolation seems to perform really well:



## DHR in other software environments

Unfortunately, I was not able to find a lot of information on completing DHR in other programming environments. I had found no available packages at all for Python, and although I did find one called “dhReg” for R [12], it appears to me that it is not exactly the same thing that Young had discussed. The package seems to be based on Hyndman’s definition of Dynamic Harmonic Regression [13], which involves “combining Fourier terms for capturing seasonality with ARIMA errors”. Perhaps I am misunderstanding what they are saying, but it does not appear to use the forward-backward algorithm that we had discussed.

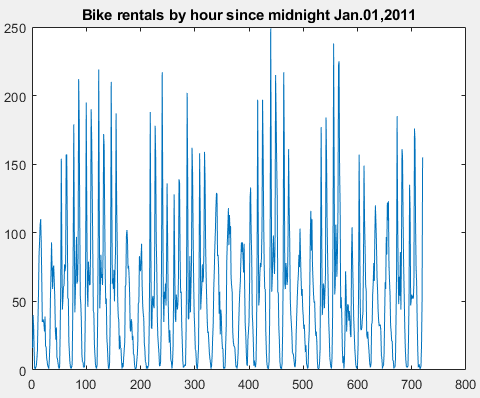
In addition, I don’t believe that the source code for CAPTAIN’s toolbox is available, which makes it harder to try to replicate it in another language. Maybe one day the author of CAPTAIN will port it over to another language, but for not, it seems the only way to perform Young’s version of DHR is through MATLAB.

## Conclusions

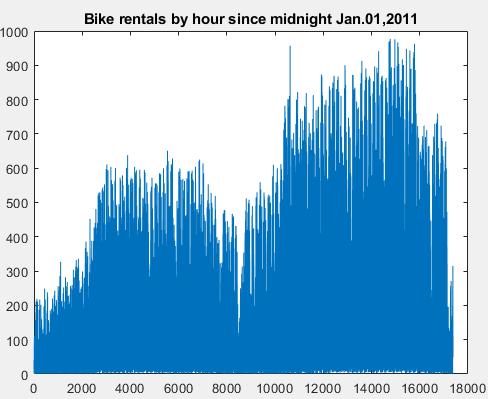
Dynamic Harmonic Regression appears to be a powerful tool for forecasting data with seasonality. It can accurately capture hourly, daily, or yearly trends. It can be used in a large variety of fields, such as solar irradiation and wind generation, or carbon dioxide levels, or air passengers. In addition, its implementation in MATLAB is relatively simple to use. I believe that one thing holding it back from more widespread usage is that it is not implemented in Python or R. In addition, it also might not predict incredibly well for data with a lot of noise.

## Appendix

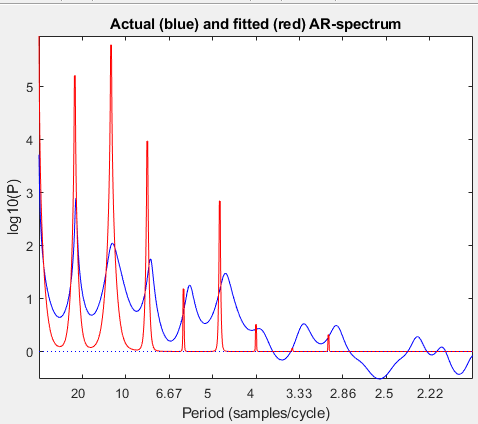
I am including an appendix here to show that either the DHR does not work well on all datasets, or that I do not understand enough about DHR to make it work for this particular dataset. The dataset in question here is the “Bike Sharing in Washington D.C. Dataset”, found at <https://www.kaggle.com/marklvl/bike-sharing-dataset>. This includes two datasets—one of hourly bike rentals, and one of daily bike rentals. If we look at the first 720 hours (first month) for the hourly data, we can see that there definitely is a trend going on, where bike rentals drop off to 0 around 3am and pick back up again. There are also less bike rentals on the weekends compared to the weekdays.



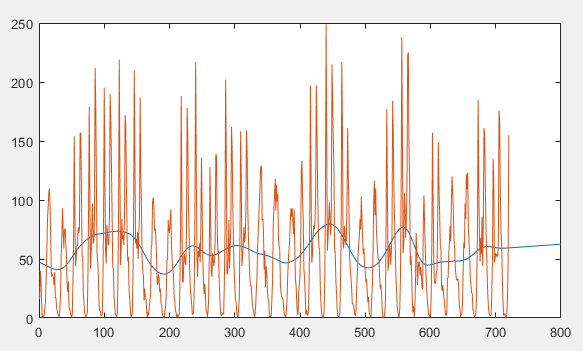
(For those of you wondering what the whole dataset looks like, it’s pretty illegible):



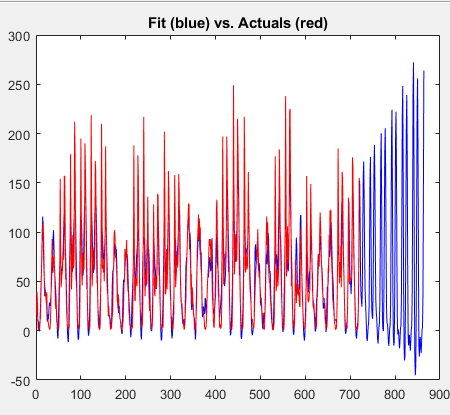
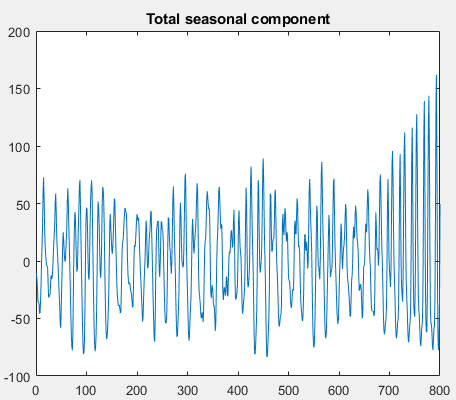
This was the best that I could do in terms of fitting the AR-spectrum:



We can see when we plot the trend lines that this data probably isn’t suited to DHR (or I am not good enough to make it suit DHR):



The fit doesn’t look too bad for the interpolation regions, but any extrapolation seems crazy:



It seems like the seasonal effect just increases significantly during the extrapolation regions. I know that Trapero had modeled hourly solar irradiation successfully, but perhaps the bike data is too noisy or the trends not clear enough to read.

# Bibliography

|  |  |
| --- | --- |
| [1] | N. K. A. M. Juan R. Trapero, "Short-term solar irradiation forecasting based on Dynamic Harmonic Regression," *Energy,* 2015. |
| [2] | R. J. Hyndman, A. B. Koehler, R. D. Snyder and S. Grose, "A State Space framework for automatic forecasting using exponential smoothing methods," [Online]. Available: https://www.monash.edu/business/econometrics-and-business-statistics/research/publications/ebs/a\_state\_space\_framework\_for\_automatic\_forecasting\_using\_exponential\_smoothing\_methods.pdf. |
| [3] | A. J. Zavala and A. R. Messina, "Dynamic Harmonic Regression Approach to Wind Power Generation Forecasting," in *PES Transmission & Distribution Conference and Exposition*, Morelia, Mexico, 2016. |
| [4] | "Autoregressive integrated moving average," [Online]. Available: https://en.wikipedia.org/wiki/Autoregressive\_integrated\_moving\_average. |
| [5] | "Decomposition of Time Series," [Online]. Available: https://en.wikipedia.org/wiki/Decomposition\_of\_time\_series. |
| [6] | "Kalman Filter," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Kalman\_filter. [Accessed 27 April 2020]. |
| [7] | D. J. P. W. T. Peter C. Young, "Dynamic Harmonic Regression," *Journal of Forecasting,* 1999. |
| [8] | Wikipedia, "Forward Backward Algorithm," [Online]. Available: https://en.wikipedia.org/wiki/Forward%E2%80%93backward\_algorithm. [Accessed 9 May 2020]. |
| [9] | "Fourier Series," [Online]. Available: https://en.wikipedia.org/wiki/Fourier\_series. |
| [10] | C. Taylor, "The CAPTAIN TOOLBOX for System Identification, Time Series Analysis, Forecasting and Control". |
| [11] | T. C, "Analysis of air passenger series," 23 02 2011. [Online]. Available: http://www.es.lancs.ac.uk/cres/captain/dhrdemo.html. |
| [12] | P. Gaikwad, "dhReg," [Online]. Available: https://cran.r-project.org/web/packages/dhReg/dhReg.pdf. |
| [13] | "Dynamic Harmonic Regression," 01 04 2020. [Online]. Available: https://otexts.com/fpp2/dhr.html. |
| [14] | U. S. Diego, "Carbon Dioxide Levels in Atmosphere," Kaggle, [Online]. Available: https://www.kaggle.com/ucsandiego/carbon-dioxide/data. [Accessed 11 05 2020]. |
| [15] | M. Kaghazgarian, "Bike Sharing in Washington D.C. Dataset," Kaggle, [Online]. Available: https://www.kaggle.com/marklvl/bike-sharing-dataset. [Accessed 11 05 2020]. |