KINETIC COMPONENT ANALYSIS

Marcos López de Prado
Guggenheim Partners
Lawrence Berkeley National Laboratory
lopezdeprado@lbl.gov

Riccardo Rebonato
EDHEC
Mathematical Institute, University of Oxford
Riccardo.Rebonato@edhec.edu

First version: February 19, 2014
This version: June 5, 2016

Journal of Investing, Forthcoming, 2016

We are grateful to Tony Anagnostakis (Moore Capital), David H. Bailey (Lawrence Berkeley National Laboratory), José Blanco (UBS), Jonathan M. Borwein (University of Newcastle), Sid Browne (Credit Suisse), Peter Carr (Morgan Stanley, NYU), Marco Dion (J.P. Morgan), David Easley (Cornell University), Matthew D. Foreman (University of California, Irvine), Jeffrey S. Lange (Guggenheim Partners), Attilio Meucci (KKR, NYU) and Luis Viceira (HBS).

KINETIC COMPONENT ANALYSIS

ABSTRACT

We introduce *Kinetic Component Analysis* (KCA), a state-space application that extracts the signal from a series of noisy measurements by applying a Kalman Filter on a Taylor expansion of a stochastic process. We show that KCA presents several advantages compared to other popular noise-reduction methods such as Fast Fourier Transform (FFT) or Locally Weighted Scatterplot Smoothing (LOWESS): First, KCA provides band estimates in addition to point estimates. Second, KCA further decomposes the signal in terms of three hidden components, which can be intuitively associated with position, velocity and acceleration. Third, KCA is more robust in forecasting applications. Fourth, KCA is a forward-looking state-space approach, resilient to structural changes. We believe that this type of decomposition is particularly useful in the analysis of trend-following, momentum and mean-reversion of financial prices.

An instrument exhibits financial inertia when its price acceleration is not significantly greater than zero for long periods of time. Our empirical analysis of 19 of the most liquid futures worldwide confirms the presence of strong inertia across all asset classes. We also argue that KCA can be useful to market makers, liquidity providers and faders for the calculation of their trading ranges.

Keywords: Kinetic Component Analysis, Time Series, Principal Component Analysis, LOWESS, Fourier Analysis, Kalman Filter.

JEL Classification: G0, G1, G2, G15, G24, E44.
AMS Classification: 91G10, 91G60, 91G70, 62C, 60E.

1. INTRODUCTION

Financial analysts and portfolio managers often describe price actions using classical mechanics terms such as momentum, oscillation, acceleration, resistance, mean-reversion, etc. Financial academics have researched and found evidence of these phenomena. In doing so, analysts, portfolio managers and academics evoke well-defined physical concepts. Since investors seem to find this analogy useful in characterizing the state of a market, we propose the formalization of this analysis.

In this paper we introduce *Kinetic Component Analysis* (KCA), a state-space application that extracts the signal from a series of noisy measurements by applying a Kalman Filter on a Taylor expansion of a stochastic process. KCA presents the additional advantage that the signal is further decomposed in terms of three hidden components. These components can be intuitively associated with position, velocity and acceleration, hence KCA's name.

The idea of decomposing financial measurements into hidden components has found many applications. Principal Component Analysis (PCA) is used to model the dynamics of yield curves, by decomposing a series of rates changes into parallel shift, slope and curvature changes. Similarly, in this paper we show how to decompose a change in prices (or rates, volatilities, etc.) into position, velocity and acceleration. We believe that this new type of decomposition is particularly useful to the analysis of trend-following, momentum and mean-reverting financial prices.

The rest of the paper is divided as follows: Section 2 discusses the literature and describes our goals and contributions. Section 3 presents our KCA method. Section 4 compares KCA with FFT. Section 5 compares KCA with LOWESS. Section 6 illustrates some of KCA's applications. Section 7 lists our conclusions.

2. LITERATURE REVIEW AND CONTRIBUTIONS

2.1. FINANCIAL MOMENTUM

De Bondt and Thaler [1985, 1987] was one of the first papers to argue that stock prices mean-revert as a result of investors' overreaction to new information. Their findings spurred a heated controversy, with some studies claiming that such mean-reversion is not a consequence of overreaction but the result of systematic risk of contrarian portfolios, size effects, liquidity effects and many other potential explanations. We refer the interested reader to Chan [1988], Ball and Kothari [1989], Zarowin [1990], Chopra et al. [1992], Jegadeesh [1990, 1992], Lehmann [1990], Lo and MacKinlay [1990], to cite only a few opinions on this subject.

In contrast, there are a myriad of papers that argue that stock prices exhibit momentum and trend-following features, see Grinblatt and Titman [1989, 1991], Jegadeesh and Titman [1993, 1999, 2002], Jarrow et al. [2003]. These views are not necessarily contradictive, since mean-reversion and momentum may occur at different timeframes, however authors associate their claims with a wide range of timeframes.

To add confusion, there is no agreement in the financial literature regarding how to define and measure these effects. Many studies define price momentum as a difference or a ratio between two average prices computed over different time windows. For example, Jegadeesh and Titman [1993] reports that Value Line defines a stock price's momentum as the ratio of the 10-week average price divided by the 52-week average price. Other studies measure momentum in terms of serial-correlation over various horizons (Moskowitz et al. [2010]). The range of values investigated for these variables seems rather arbitrary, and conclusions are likely to be sensitive to choosing different parameter combinations. This of course also raises the problem of backtest overfitting, or how representative are these conclusions out-of-sample (see Bailey et al. [2014a, 2014b] on this important issue).

2.2. FINANCIAL SIGNAL PROCESSING

It is known that mathematicians and physicists have applied signal processing to betting problems for at least 50 years. Perhaps the earliest example is Claude Shannon, known as the "father of information theory," who applied this knowledge to make a fortune in Las Vegas' casinos. Ed Thorp and John L. Kelly (the discoverer of the Kelly Criterion for bet sizing), also successfully used information theory principles to beating casinos (Poundstone [2005]).

In the context of investments, signal processing has also proven extremely successful. Elwyn Berlekamp, Kelly's research assistant from 1960 to 1962, wrote the algorithms used by Axcom's Medallion Fund. This fund is now managed by Renaissance Technologies (also known as RenTech), and has returned in excess of 30% average annual return over the last 20 years (Berlekamp [2005]). RenTech has been considered one of the most successful hedge funds in history (Lux [2000]). Although hedge funds are extremely secretive regarding their quantitative techniques, we may deduce the relevance of signal processing to RenTech by noting that the firm is currently managed by Peter Brown and Robert Mercer. These two scientists were recruited by RenTech because of the expertise in signal processing that they developed while working at IBM. In the words of Nick Patterson, a Renaissance researcher: *"I realized that there are some deep technical links between speech recognition is done and some good ways of predicting the markets*" (Patterson and Strasburg [2010]).

Although it is evident that some of the most successful investors have used signal processing techniques, the number of publications dedicated to financial applications of signal processing is relatively small. The IEEE Signal Processing Society did not call for a Special Issue on Signal Processing Methods in Finance and Electronic Trading until as late as 2011. The issue was eventually published in August of 2012 (Volume 6, Number 4 of the IEEE Journal of Selected Topics in Signal Processing), and it dealt with a variety of issues such as the robust estimation of correlation matrices, an evaluation of the CAPM model, log-volatility estimation, modelling of transaction costs, Extreme Learning Machine, etc.

2.3. CONTRIBUTIONS

In this paper we make a number of contributions to the literature of financial signal processing. First, we introduce the KCA decomposition of a series of noisy measurements into three classical mechanics components: Position, Velocity and Acceleration. We show how a Kalman filter specified on the Taylor expansion of the series produces good point and confidence interval estimates. While engineers have used Kalman filters to track the movement of physical objects, we believe that this is the first study to use these techniques to decompose the dynamics of financial prices and analyze trend information.

Second, we introduce an algorithm that extracts the subsample of most significant FFT frequencies while controlling for overfitting. Third, we show that KCA delivers better and more insightful results than FFT, even when in the FFT-ideal scenario of fitting noisy measurements from a periodic signal.

Fourth, we explain how KCA can be applied to determining the levels at which market makers should provide liquidity in order to avoid adverse selection, or what constitutes a similar application, at what levels prices could be faded.

3. OUR MODEL

We would like to decompose a financial instrument's price dynamics in terms of three instantaneous components: Position, velocity and acceleration. These components are not directly observable, and measurements incorporate noise. We rely on a Kalman filter to model a state-space system of equations, where the matrices that characterize the system are derived from a Taylor expansion of the price dynamics.

3.1. KINETIC DECOMPOSITION BY TAYLOR SERIES

Let's denote the observable price, or rate, or yield on which we focus by $P(t)$. We assume that the observed price is made up of a continuous and twice-differentiable latent component, p (t) — which for reasons we explain below we refer to as the 'fundamental component' —, and a source of noise — for instance, Brownian shocks, $h(t)$:

$$P(t) = p(t) + h(t) \qquad (1)$$

So, the observed price function, $P(t)$, inherits the non-differentiability from the noise component, $h(t)$. We can, but need not, give the following interpretation to the latent fundamental component, $p(t)$: it is output of a (smooth) mapping from the 'fundamental' economic variables to the smooth component of the price. To give a stylized but illuminating example, let $P_i(t)$ be the observed i-maturity yields at time *t*. Let's place ourselves in a world where central banks, which determine the short rate, follow strictly the Taylor rule. Let's also assume, for the sake of argument, that the behavior of the economic variables (inflation, output, etc.) that enter the Taylor rule follow such processes that the resulting behavior of the short rate is mean-reverting (CIR or Vasicek-like). Then, in this universe, the yields of i-maturity bonds are a function of the reversion speed, reversion level and volatility of this process. In this stylized universe, the function $p(t)$ would then be the result of the smooth mapping (via the Taylor rule and the Vasicek/CIR-like 'duration' terms) from the macroeconomic variables (inflation, output, etc.) to smooth component of the observed prices, $P(t)$. With our procedure we endeavor to recover the latent function $p(t)$. We stress that our procedure does not rely on the actual existence of the mapping outlined above; nor does it rely on the correctness of the structural model, or on our ability to determine its parameters. We only offer an interpretation of our target latent function to aid intuition and facilitate an assessment of the plausibility of the results.

Consider a price function in continuous time, where $p(t)$ represents the expected value of price at time *t*. We assume that $p(t)$ is twice differentiable with respect to time. The underlying function may represent prices, log prices, rates, implied volatility, or any other form of asset value quotation. Its Taylor series up to a second degree is, centered around t_0:

$$p(t) = p(t_0) + \left.\frac{\partial p(t)}{\partial t}\right|_{t=t_0} (t - t_0) + \frac{1}{2}\left.\frac{\partial^2 p(t)}{\partial t^2}\right|_{t=t_0} (t - t_0)^2 + \sum_{n=3}^{\infty}\frac{1}{n!}\left.\frac{\partial^n p(t)}{\partial t^n}\right|_{t=t_0} (t - t_0)^n \tag{2}$$

For a t value sufficiently close to t_0, we can assume that $\sum_{n=3}^{\infty}\frac{1}{n!}\left.\frac{\partial^n p(t)}{\partial t^n}\right|_{t=t_0} (t - t_0)^n \approx 0$, hence

$$p(t) \approx p(t_0) + \left.\frac{\partial p(t)}{\partial t}\right|_{t=t_0} (t - t_0) + \frac{1}{2}\left.\frac{\partial^2 p(t)}{\partial t^2}\right|_{t=t_0} (t - t_0)^2 \tag{3}$$

Let us simplify this notation by renaming $v(t_0) = \left.\frac{\partial p(t)}{\partial t}\right|_{t=t_0}$ (analogous to p's *velocity*) and $a(t_0) = \left.\frac{\partial^2 p(t)}{\partial t^2}\right|_{t=t_0}$ (analogous to p's *acceleration*). We can discretize this function by setting $h = t - t_0$, and sampling values at that fixed frequency. Then, Eq. (3) can be represented in discrete time as:

$$p_s \approx p_{s-1} + v_{s-1}h + \frac{1}{2}a_{s-1}h^2 \tag{4}$$

with $s = 1, \dots, S$ samples. This means that p_s can be approximated by p_{s-1}, with an adjustment v_{s-1} linear in the time step h and another linear adjustment $\frac{1}{2}a_{s-1}$ quadratic in h. The approximation improves as h becomes smaller. Likewise, variable v_s can be approximated through a Taylor series up to the first degree, since $a(t_0) = \left.\frac{\partial^2 p(t)}{\partial t^2}\right|_{t=t_0} = \left.\frac{\partial v(t)}{\partial t}\right|_{t=t_0}$. The result is a discrete dynamic system characterized as:

$$\begin{gathered} p_s \approx p_{s-1} + v_{s-1}h + \frac{1}{2}a_{s-1}h^2 \\ v_s \approx v_{s-1} + a_{s-1}h \\ a_s \approx a_{s-1} \end{gathered} \tag{5}$$

Eq. (5) can be expanded to derivatives of any degree. In other words, there is no need to assume that acceleration is parsimonious ($a_s \approx a_{s-1}$).

3.2. KALMAN REPRESENTATION OF THE KINETIC DECOMPOSITION

The system in Eq. (5) can be represented as a Kalman filter of the form:

$$\begin{gathered} x_s = Ax_{s-1} + Bu_{s-1} + w_s \\ z_s = Hx_s + \varepsilon_s \end{gathered} \tag{6}$$

where $x \in \mathfrak{R}^n$ represents the system's state, and $z \in \mathfrak{R}^m$ represents the measurement. Of course, the state is unobservable, hence the state-space characterization. The transition from state x_{s-1} to state x_s is modeled through matrix *A*, of order *nxn*. Random vector w_s represents the process noise and is assumed to be distributed $w_s \sim N(0, Q)$, where Q is the process noise covariance matrix, of order *nxn*. Random vector z_s is the measurement, which results from a linear transformation of the state x_s, determined by *mxn* matrix *H*. Random vector ε_s is the measurement noise, which is assumed to be distributed $\varepsilon_s \sim N(0, R)$, where R is the measurement noise covariance, of order *mxm*.

Matrix *B* has order *nxl*, and it relates the control input $u_{s-1} \in \mathfrak{R}^l$ to the state x_s. We discuss it in a separate paragraph because this component is optional. Systems where the observer has no possibility to influence the states do not require this component. However, if the observer's actions can influence the states of the system, this *B* matrix would specify how the observer reacts to the system's feedback. For example, this would be an interesting feature in the case of a large trader adjusting her aggressiveness in response to increased market impact. Another example would be how an energy company adjusts supply to respond to price oscillations, thus having some control on the prices.

We need to specify *H*, *A*, *Q* and *R* to incorporate the features in our approach:

- Matrix *H* has order *mxn*. In Section 3.1, $x_s \in \mathfrak{R}^3$, because our system is characterized by three variables, $\{p_s, v_s, a_s\}$. However observation $z_s \in \mathfrak{R}$, hence $(m, n) = (1,3)$. In our model, observation z_s can only be mapped to the first component of the state x_s. From this, we deduce that $H = [1 \quad 0 \quad 0]$.
- Matrix *A* has order *nxn*, which in our framework translates to 3x3. We can deduce its elements from Eq. (5), $A = \begin{bmatrix} 1 & h & \frac{1}{2}h^2 \\ 0 & 1 & h \\ 0 & 0 & 1 \end{bmatrix}$.
- Matrix *Q* has order *nxn*, which in our framework translates to 3x3. Depending on the observations, the state components may or may not exhibit any cross-correlation. An EM algorithm can be used to determine the optimal value of Q. A possible seed could take the form $Q = qI_n$, where I_n is an identity matrix of order *nxn*, and q is real positive value. The greater q, the more likely we are to overfit, because we are indicating to the model that a greater proportion of the noise comes from the states rather than the measurements. So the researcher should try alternative values of q until she settles for one that delivers an output consistent with her knowledge of the system. In general, lower values of q should be preferred, in order to mitigate overfitting.
- Matrix *R* has order *mxm*, which in our framework translates to 1x1, a real positive number. This value is the easiest to estimate by EM, since z_s are directly observable.

Once the problem has been fully characterized, we can apply standard Kalman Filter estimation algorithms to solve it. We refer the reader to Welch and Bishop [2006] and Brown and Hwang [1997] for a description of such algorithms. Appendix 1 provides the Python code that implements the KCA specification, generates estimates and delivers forecasts.

4. KCA vs. FFT

The Fast Fourier Transform (FFT) is an algorithm that transforms a function of time into a function of frequency. It accomplishes that by approximating general functions as linear combinations of periodic functions. FFT is applied to functions in a similar way that PCA is applied to vector spaces. Analogously to the way PCA is used to extract the principal components of a vector space, we can use FFT to extract the principal frequencies that best describe the signal in a noisy measurement.

One necessary caution with Fourier analysis is that it can extract noise as easily as it extracts signal. At the limit, if we use all the frequencies, we will perfectly fit noise.[1] We must therefore come up with a criterion that determines what frequencies are truly relevant, and halts the extraction once additional model complexity does not provide significant additional explanatory power. Appendix 2 provides the Python implementation of one such procedure which we have devised for preventing FFT overfitting. At every iteration our algorithm scans all unused frequencies, looking for the one that delivers the greatest decrease on the residual's Ljung-Box statistic. Our algorithm stops when the probability associated with the Ljung-Box statistic exceeds a given threshold (e.g., 5%), or the Ljung-Box decreases by less than a threshold, hence indicating the further model complexity is unwarranted.

FFT has been considered *"one of the most important numerical algorithms of our lifetime"* (Strang [1994]). Since FFT is the standard signal extraction method in mathematics, science and engineering, it would be useful to evaluate KCA's performance against FFT. In order to demonstrate KCA's capabilities, we have chosen a time series where FFT is expected to perform well. Extracting a periodic signal from noisy measurements is a particularly trivial task for FFT, since it relies on periodic functions for fitting the data. Appendix 3 details the Python code that generates a sinusoidal sample, to which we have added Gaussian white noise. Figure 1 plots the states estimated by KCA, which are very close to the actual states used to generate the observations.

[FIGURE 1 HERE]

Figure 2 shows that FFT extracts a signal very similar to the first state component extracted by KCA. FFT signal extraction was halted once additional frequencies could not reduce the Ljung-Box statistic by more than 5%, thus preventing overfitting. Note FFT's departure around the edges of the series. This situation is called Gibbs' phenomenon, and it is the consequence of attempting to fit piecewise continuously differentiable periodic functions around a jump discontinuity, and in particular around the beginning and end of the series. KCA is not affected by Gibbs' phenomenon. However, we can appreciate that KCA fits the right edge better than the left edge. The reason is, Kalman filters' estimates converge to the true values asymptotically, and it takes a few observations at the beginning of the series for that convergence to take place (a situation known as *burn-in period* in Bayesian filtering). This is not usually a problem, because researchers are almost always more interested on the right edge, for the purpose of forecasting.

[FIGURE 2 HERE]

[1] A visual demonstration of FFT's overfitting is available at: http://youtu.be/D8e3FcySitY

One important advantage of KCA over FFT is that the former provides estimates of the means as well as the standard deviations of the hidden states. Figure 3 plots the states estimates as well as the 2 standard deviation confidence intervals.

[FIGURE 3 HERE]

In summary, KCA presents three advantages over FFT: i) KCA provides point as well as confidence interval estimates of the signal's position, while FFT only provides a point estimate. ii) beyond the position state, KCA also reveals information regarding the velocity and acceleration of the series (with confidence bands for the three of them). iii) KCA's extracted signal is closer to the true signal at the extremes of the series, because KCA does not exhibit the Gibbs phenomenon. This third advantage is critical, because a researcher is typically interested in extrapolating or forecasting a signal, which requires that the most recent estimates are the most accurate.

5. KCA vs. LOWESS

Locally Weighted Scatterplot Smoothing (LOWESS) is another popular method used to deal with noisy measurements. LOWESS fits weighted linear regressions to localized subsets of the data in order to build a function that filters noise point by point (Cleveland [1979]). Figure 4 plots the result of fitting several LOWESS functions to the same observations used in Section 4. When LOWESS is fit on local regressions that employ 50% of the data, the fit barely resembles the signal. A LOWESS function that uses 25% of the data gives a result similar to FFT's. A LOWESS function of 10% is very close to KCA's estimate. As the fraction of data is reduced, the LOWESS function fits the signal more closely, but unfortunately it also becomes more unstable. We can appreciate the onset of that phenomenon on the LOWESS(0.1) function, which exhibits several irregularities or bumps.

[FIGURE 4 HERE]

Figure 5 provides an example of a 20 step forward forecast performed by the KCA algorithm. KCA presents several advantages compared to LOWESS: i) LOWESS is not equipped to generate forecasts, because each estimate is highly dependent on the neighboring sample (before and after the observation). KCA incorporates a Kalman filter, thus inheriting the features of state-space signal processing methods. In particular, KCA is a forward-looking method that is robust to structural changes. ii) LOWESS requires a subsample to generate every single estimate, which makes it computationally intensive. Not only KCA can produce accurate forecasts several steps forward, but it can be updated online. This means that the last known state of the system is all KCA needs to forecast the next. iii) Like in the case of FFT, LOWESS does not decompose the signal into the three kinetic components, nor provides confidence intervals for those estimates.

[FIGURE 5 HERE]

6. SOME APPLICATIONS

6.1. FINANCIAL INERTIA

As we mentioned in Section 2, the financial literature has debated for decades whether financial momentum exists. We have called this section "Financial Inertia" rather than "Financial Momentum" because we believe that the former is what the academic literature actually meant. In plain English, *inertia* is the tendency of an object to keep moving in a straight line at a constant speed. The principle of inertia is postulated in Newton's First Law of Motion. In contrast, *momentum* is the product of mass and velocity. Newton's Second Law tells us that it takes the same force to deviate an object at double speed with half mass or at half speed with double mass. Momentum requires the definition of mass and velocity, while inertia is merely the observation that an object's velocity remains unaltered unless a force acts upon it.

It is not the goal of this section to settle a long-standing controversy, but to demonstrate the use of KCA. We have studied the price dynamics of some of the most liquid investments across all asset classes. By applying KCA on price series, we can extract estimates of price acceleration. An instrument exhibits financial inertia when its price acceleration is not significantly greater than zero for long periods of time.

Table 1 summarizes our data. Our source is level 1 tick data recorded by TickWrite. For each instrument we used the front contract, rolled forward by volume. Tick series were grouped in volume buckets, at an average of 1 bucket per day. KCA was then applied on the series of volume weighted average prices (VWAP) computed on each bucket.

[TABLE 1 HERE]

Table 2 lists the key inertia statistics per contract. *Mean_Accel* is the average value of the acceleration. *Std_Accel* is the standard deviation of the estimated acceleration values. *Inertia* is the proportion of acceleration estimates that did not exceed a 95% confidence bound centered around zero. This value can be interpreted as the proportion of market activity associated with insignificant acceleration. The greater the inertia, the greater was the amount of activity (measured as transacted volume) that occurred under a relatively unchanged price speed.

[TABLE 2 HERE]

Inertia results are generally high, with an average value of 0.8942 and a standard deviation of 0.067. Corn (CN) registers the lowest reading, at 0.7615, and Euro FX (EC) the highest, at 1. Out of the 19 instruments studied, 9 exhibit an inertia greater than 0.9: Dollar Index (DX), Euro FX (EC), Natural Gas (NG), E-mini S&P500 (ES), XX (Eurostoxx 50), E-mini Dow-Jones (YM), Eurodollar (ED), T-Note 5 years (FV) and T-Note 2 years (TU).

An intuitive result is given by futures on treasury notes, where the inertia is lower as the duration increases. As short term interest rates are anchored by the Federal Reserve's policy, inertia gradually increases as we move away from the yield curve's front end: From 0.9986 in the case of Eurodollar (ED), all the way to 0.8588 in the case of T-Bond 30 years (US).

In terms of average inertia per asset class, the highest average values seem to be associated with currencies (0.9540, although we only count with two examples) and rates/fixed income (0.9298). These are followed by equity indices (0.8963) and commodities (0.8425).

Much has been published in recent financial outlets regarding the poor performance experienced by momentum or trend following funds. This contrasts with the above results, as they evidence strong inertia (the basis for the profitability of momentum funds). One possible explanation is the application of flawed trading rules to the monetization of existing momentum opportunities. A popular trading rule is called "crossing moving averages momentum", and it consists in taking a long position on product X whenever the moving averages of sample sizes m and n satisfy the condition $\bar{x}_m > \bar{x}_n$, where $m < n$. A short position would be triggered by $\bar{x}_m < \bar{x}_n$. Since the sample size determines a limited number of parameter combinations that *m* and *n* can adopt, it is relatively easy to determine the pair (*m*,*n*) that maximizes the backtest's performance. This sort of backtest overfitting has been shown to lead to negative performance in the presence of memory effects (Bailey et al. [2014a, 2014b]). Furthermore, let us not forget that "crossing moving averages" is an adaptive trading rule, in the sense that its estimates are purely historical and do not attempt to anticipate future behavior. That is not the case of KCA, which relies on Bayesian learning to update forward-looking priors.

For these two reasons (overfitted and adaptive trading rules), it is not unreasonable to think that momentum funds can generate substantial losses even in the presence of strong momentum. A better approach may have been to invest in momentum opportunities by applying forward-looking trading rules that are not so easily overfitted.

6.2. MICROSTRUCTURAL NOISE

Beyond the traditional study of financial momentum, KCA can be used in a variety of contexts. One such application is the modeling of price dynamics under microstructural noise. With the advent of High Frequency Trading, a large percentage of quotes are generated with no intention of actual trading. Also, trades often lead to positions that last barely a few seconds or even less. This microstructural noise makes it difficult to determine towards what levels prices are trending, and at what speed.

Suppose that a market maker targets to hold inventory for a maximum period h, in chronological or volume time (see Easley et al. [2012b] for the difference). In Section 4 we demonstrated how KCA provides estimated means and confidence intervals for our three components. We can use these estimates to determine the levels at which market makers can provide liquidity. In particular, let $\underline{p}_s$, $\underline{v}_s$, $\underline{a}_s$ be the lower bound estimates for position, velocity and acceleration, and $\overline{p}_s$, $\overline{v}_s$, $\overline{a}_s$ the respective upper bound estimates associated with a significance level α. For example, $\underline{p}_s = p_s + Z_\alpha \sigma_{p,s}$ and $\overline{p}_s = p_s - Z_\alpha \sigma_{p,s}$, with analogous expressions for the other components. Then, we can apply the Taylor expansion in Section 3.1 to determine a confidence interval for p_{s+1}, at which liquidity can be provided

$$\begin{aligned} \underline{p}_{s+1} &\approx p_s + Z_\alpha \sigma_{p,s} + \left(v_s + Z_\alpha \sigma_{v,s}\right)h + \frac{1}{2}\left(a_s + Z_\alpha \sigma_{a,s}\right)h^2 \\ \overline{p}_{s+1} &\approx p_s - Z_\alpha \sigma_{p,s} + \left(v_s - Z_\alpha \sigma_{v,s}\right)h + \frac{1}{2}\left(a_s - Z_\alpha \sigma_{a,s}\right)h^2 \end{aligned} \tag{7}$$

where $Z_\alpha < 0$ for $\alpha < \frac{1}{2}$, hence the signs of the Z_α factors. The same approach is valid for traders who wish to fade market overreactions. The advantage of following this method compared to popular heuristics such as Bollinger Bands or range trading is that KCA is forward looking, while the former methods are backwards looking. This means that KCA responds much quicker to structural breaks, while Bollinger Bands will not respond for the duration of the sample length used to estimate historical means and standard deviations. KCA's forward looking feature is due to the Bayesian Inference approach implicit in the Kalman filter.

7. CONCLUSIONS

Principal Component Analysis (PCA) derives what are the orthogonal components that explain most of the variance in a vector space. In a functional space, mathematicians, scientists and engineers have used the Fast Fourier Transform (FFT) to extract the principal frequencies that characterize a signal. These are valuable techniques that apply linear algebra and functional analysis to decompose a series into hidden components.

In this paper we have introduced a new technique, called *Kinetic Component Analysis* (KCA), that computes a Kalman filter on a Taylor expansion of a series of noisy measurements. As a result, our approach belongs to the family of state-space, signal processing methods. The series is decomposed into classical mechanic components, position, velocity and acceleration.

Several features make KCA preferable over other popular noise reduction procedures, such as FFT or LOWESS. First, KCA provides point as well as confidence interval estimates of the signal's position. Second, beyond the position state, KCA also reveals information regarding the velocity and acceleration of the series (with confidence bands for the three of them). Third, KCA does not exhibit the Gibbs phenomenon. Fourth, KCA is forward-looking and resilient to structural changes. Fifth, KCA is updated online.

An instrument exhibits financial inertia when its price acceleration is not significantly greater than zero for long periods of time. Our empirical analysis of 19 of the most liquid futures worldwide confirms the presence of strong inertia across all asset classes. Beyond providing robust estimates of financial momentum, KCA can be useful by market makers, liquidity providers and faders to determine their trading ranges.

APPENDICES

A.1. PYTHON IMPLEMENTATION OF KCA

Snippet 1 shows the implementation of KCA in Python. Dependencies consist of two libraries: *numpy* and *pykalman*. All Python libraries used in this paper are included in Enthought's Canopy distribution.

```
# by MLdP on 02/22/2014 <lopezdeprado@lbl.gov>
# Kinetic Component Analysis
import numpy as np
from pykalman import KalmanFilter
#------------------------------------------------------------------------------
def fitKCA(t,z,q,fwd=0):
    '''
    Inputs:
        t: Iterable with time indices
        z: Iterable with measurements
        q: Scalar that multiplies the seed states covariance
        fwd: number of steps to forecast (optional, default=0)
    Output:
        x[0]: smoothed state means of position velocity and acceleration
        x[1]: smoothed state covar of position velocity and acceleration
    Dependencies: numpy, pykalman
    '''
    #1) Set up matrices A,H and a seed for Q
    h=(t[-1]-t[0])/t.shape[0]
    A=np.array([[1,h,.5*h**2],
                [0,1,h],
                [0,0,1]])
    Q=q*np.eye(A.shape[0])
    #2) Apply the filter
    kf=KalmanFilter(transition_matrices=A,transition_covariance=Q)
    #3) EM estimates
    kf=kf.em(z)
    #4) Smooth
    x_mean,x_covar=kf.smooth(z)
    #5) Forecast
    for fwd_ in range(fwd):
        x_mean_,x_covar_=kf.filter_update(filtered_state_mean=x_mean[-1], \
            filtered_state_covariance=x_covar[-1])
        x_mean=np.append(x_mean,x_mean_.reshape(1,-1),axis=0)
        x_covar_=np.expand_dims(x_covar_,axis=0)
        x_covar=np.append(x_covar,x_covar_,axis=0)
    #6) Std series
    x_std=(x_covar[:,0,0]**.5).reshape(-1,1)
    for i in range(1,x_covar.shape[1]):
        x_std_=x_covar[:,i,i]**.5
        x_std=np.append(x_std,x_std_.reshape(-1,1),axis=1)
    return x_mean,x_std,x_covar
```

Snippet 1 – KCA implementation

Function fitKCA has three arguments: *t*, *z*, and *q*. Numpy array *t* conveys the index of observations. Numpy array *z* passes the observations. Scalar *q* provides a seed value for initializing the EM estimation of the states covariance. Positive integer *fwd* determines the number of steps forward to be forecasted. This is an optional argument, with default value 0.

A.2. FFT SIGNAL EXTRACTION WITH FREQUENCY SELECTION

Snippet 2 shows the implementation a FFT signal extraction with frequency selection in Python. It relies on two popular Python libraries: *numpy* and *statsmodels*. The arguments are *series* and *minAlpha*. *series* is a numpy array containing the observation. *minAlpha* is an optional variable, with a default value of *None*. When a value is passed for *minAlpha*, this algorithm will select frequencies until one of two conditions is verified: 1) The Ljung-Box statistic is statistically significant beyond a probability *minAlpha*, or 2) the Ljung-Box statistic has decreased by less than a proportion *minAlpha*, indicating that additional model complexity is not justified in terms of greater explanatory power.

```
# by MLdP on 02/20/2014 <lopezdeprado@lbl.gov>
# FFT signal extraction with frequency selection
import numpy as np
import statsmodels.stats.diagnostic as sm3
#-----------------------------------------------------------
def selectFFT(series,minAlpha=None):
    # Implements a forward algorithm for selecting FFT frequencies
    #1) Initialize variables
    series_=series
    fftRes=np.fft.fft(series_,axis=0)
    fftRes={i:j[0] for i,j in zip(range(fftRes.shape[0]),fftRes)}
    fftOpt=np.zeros(series_.shape,dtype=complex)
    lags,crit=int(12*(series_.shape[0]/100.)**.25),None
    #2) Search forward
    while True:
        key,critOld=None,crit
        for key_ in fftRes.keys():
            fftOpt[key_,0]=fftRes[key_]
            series__=np.fft.ifft(fftOpt,axis=0)
            series__=np.real(series__)
            crit_=sm3.acorr_ljungbox(series_-series__,lags=lags) # test for the max # lags
            crit_=crit_[0][-1],crit_[1][-1]
            if crit==None or crit_[0]<crit[0]:crit,key=crit_,key_
            fftOpt[key_,0]=0
        if key!=None:
            fftOpt[key,0]=fftRes[key]
            del fftRes[key]
        else:break
        if minAlpha!=None:
            if crit[1]>minAlpha:break
            if critOld!=None and crit[0]/critOld[0]>1-minAlpha:break
    series_=np.fft.ifft(fftOpt,axis=0)
    series_=np.real(series_)
    out={'series':series_,'fft':fftOpt,'res':fftRes,'crit':crit}
    return out
```

Snippet 2 – FFT implementation

The output object *out* is a dictionary containing:

- *'series'*: The extracted signal, as a double type numpy array.
- *'fft'*: The selected frequencies, as a complex type numpy array.
- *'res'*: A dictionary containing the unused frequencies. Frequencies not listed in this dictionary have been selected for signal extraction.
- *'crit'*: The value of the Ljung-Box stat associated with the extracted signal.

A.3. KCA vs. FFT

Snippet 3 relies on the standard library numpy as well as the algorithms introduced in Appendices 1 and 2. Matplotlib is used for plotting the results. Function *vsFFT* carries out the following calculations. First, it generates a series of noisy measurements of a periodic function, which are stored in a numpy array *z*. Second, we apply KCA on *z*, obtaining estimates of the means and confidence intervals of the hidden states associated with position, velocity and acceleration. Second, we select the most relevant FFT frequencies that minimize the Ljung-Box statistic on the sample's residuals. KCA and FFT's results are then plotted for comparison.

```
# by MLdP on 02/20/2014 <lopezdeprado@lbl.gov>
# Kinetic Component Analysis of a periodic function
import numpy as np,matplotlib.pyplot as pp,kca
from selectFFT import selectFFT
mainPath='../../'
#---------------------------------------------------------
def getPeriodic(periods,nobs,scale,seed=0):
    t=np.linspace(0,np.pi*periods/2.,nobs)
    rnd=np.random.RandomState(seed)
    signal=np.sin(t)
    z=signal+scale*rnd.randn(nobs)
    return t,signal,z
#---------------------------------------------------------
def vsFFT():
    #1) Set parameters
    nobs,periods=300,10
    #2) Get Periodic noisy measurements
    t,signal,z=getPeriodic(periods,nobs,scale=.5)
    #3) Fit KCA
    x_point,x_bands=kca.fitKCA(t,z,q=.001)[:2]
    #4) Plot KCA's point estimates
    color=['b','g','r']
    pp.plot(t,z,marker='x',linestyle='',label='measurements')
    pp.plot(t,x_point[:,0],marker='o',linestyle='-',label='position', \
        color=color[0])
    pp.plot(t,x_point[:,1],marker='o',linestyle='-',label='velocity', \
        color=color[1])
    pp.plot(t,x_point[:,2],marker='o',linestyle='-',label='acceleration', \
        color=color[2])
    pp.legend(loc='lower left',prop={'size':8})
    pp.savefig(mainPath+'Data/test/Figure1.png')
    #5) Plot KCA's confidence intervals (2 std)
    for i in range(x_bands.shape[1]):
        pp.plot(t,x_point[:,i]-2*x_bands[:,i],linestyle='-',color=color[i])
```

```
        pp.plot(t,x_point[:,i]+2*x_bands[:,i],linestyle='-',color=color[i])
    pp.legend(loc='lower left',prop={'size':8})
    pp.savefig(mainPath+'Data/test/Figure2.png')
    pp.clf();pp.close() # reset pylab
    #6) Plot comparison with FFT
    fft=selectFFT(z.reshape(-1,1),minAlpha=.05)
    pp.plot(t,signal,marker='x',linestyle='',label='Signal')
    pp.plot(t,x_point[:,0],marker='o',linestyle='-',label='KCA position')
    pp.plot(t,fft['series'],marker='o',linestyle='-',label='FFT position')
    pp.legend(loc='lower left',prop={'size':8})
    pp.savefig(mainPath+'Data/test/Figure3.png')
    return
```

Snippet 3 – KCA vs. FFT

A.4. KCA vs. LOWESS

Snippet 4 relies on the standard libraries numpy, statsmodels, as well as the algorithm introduced in Appendix 1. Matplotlib is used for plotting the results. We have omitted function *getPeriodic* for brevity, as that was listed in Snippet 3.

Function *vsLOWESS* estimates a LOWESS on the same sample used in Appendix 3. The fraction of the sample used by each local regression is determined by *frac*. We have tried three alternative values: .5, .25 and .1. KCA and LOWESS' results are then plotted for comparison.

```
# by MLdP on 02/20/2014 <lopezdeprado@lbl.gov>
# Kinetic Component Analysis of a periodic function
import numpy as np,matplotlib.pyplot as pp,kca
import statsmodels.nonparametric.smoothers_lowess as sml
mainPath='../../'
#---------------------------------------------------------
def vsLOWESS():
    # by MLdP on 02/24/2014 <lopezdeprado@lbl.gov>
    # Kinetic Component Analysis of a periodic function
    #1) Set parameters
    nobs,periods,frac=300,10,[.5,.25,.1]
    #2) Get Periodic noisy measurements
    t,signal,z=getPeriodic(periods,nobs,scale=.5)
    #3) Fit KCA
    x_point,x_bands=kca.fitKCA(t,z,q=.001)[:2]
    #4) Plot comparison with LOWESS
    pp.plot(t,z,marker='o',linestyle='',label='measurements')
    pp.plot(t,signal,marker='x',linestyle='',label='Signal')
    pp.plot(t,x_point[:,0],marker='o',linestyle='-',label='KCA position')
    for frac_ in frac:
        lowess=sml.lowess(z.flatten(),range(z.shape[0]),frac=frac_)[:,1].reshape(-1,1)
        pp.plot(t,lowess,marker='o',linestyle='-',label='LOWESS('+str(frac_)+')')
    pp.legend(loc='lower left',prop={'size':8})
    pp.savefig(mainPath+'Data/test/Figure4.png')
    return
```

Snippet 4 – KCA vs. LOWESS

TABLES

SYMBOL	DESCRIPTION	EXCHANGE	CLASS	CURRENCY	START	END
DX	Dollar index ICE	ICE	Currency	USD	1/1/2007	8/1/2012
EC	Euro FX	CME	Currency	USD	1/1/2007	8/1/2012
DA	Dax futures	Eurex	Equity	EUR	1/1/2007	8/1/2012
DJ	DJIA Futures	CBOT	Equity	USD	1/1/2007	8/1/2012
ES	S&P 500 E-mini	CME	Equity	USD	1/1/2007	8/1/2012
NQ	Nasdaq 100	CME	Equity	USD	1/1/2007	8/1/2012
XX	EURO STOXX 50	Eurex	Equity	EUR	1/1/2007	8/1/2012
YM	Dow Jones E-mini	CBOT	Equity	USD	1/1/2007	8/1/2012
ED	Eurodollar	CME	Interest rates	USD	1/1/2007	8/1/2012
TU	T-Note 2 yr	CBOT	Interest rates	USD	1/1/2007	8/1/2012
FV	T-Note 5 yr	CBOT	Interest rates	USD	1/1/2007	8/1/2012
TY	T-Note 10 yr	CBOT	Interest rates	USD	1/1/2007	8/1/2012
US	T-Bond 30 yr	CBOT	Interest rates	USD	1/1/2007	8/1/2012
CL	Light Crude NYMEX	NYMEX	Energy	USD	1/1/2007	8/1/2012
NG	Natural Gas	NYMEX	Energy	USD	1/1/2007	8/1/2012
CN	Corn	CBOT	Grain	USD	1/1/2007	8/1/2012
LH	Lean Hogs	CME	Meal	USD	1/1/2007	8/1/2012
GC	Gold Comex	COMEX	Metal	USD	1/1/2007	8/1/2012
CT	Cotton #2	ICE	Softs	USD	1/1/2007	8/1/2012

Table 1 – Futures contracts used in our empirical analysis

SYMBOL	DESCRIPTION	MEAN_ACCEL	STD_ACCEL	INERTIA
DX	Dollar index ICE	-6.35E-05	4.27E-02	0.9080
EC	Euro FX	5.06E-06	1.43E-03	1.0000
DA	Dax futures	-4.25E-01	6.94E+00	0.8627
DJ	DJIA Futures	-7.71E-01	1.25E+01	0.8926
ES	S&P 500 E-mini	-8.91E-02	1.33E+00	0.9148
NQ	Nasdaq 100	-1.21E-01	1.97E+00	0.8630
XX	EURO STOXX 50	-2.51E-01	3.92E+00	0.9224
YM	Dow Jones E-mini	-7.80E-01	1.20E+01	0.9224
ED	Eurodollar	2.51E-05	7.41E-03	0.9986
TU	T-Note 2 yr	-3.94E-05	1.19E-02	0.9875
FV	T-Note 5 yr	-2.85E-04	2.81E-02	0.9223
TY	T-Note 10 yr	-5.95E-04	3.76E-02	0.8815
US	T-Bond 30 yr	-1.32E-03	5.38E-02	0.8588
CL	Light Crude NYMEX	-2.63E-04	8.60E-02	0.8162
NG	Natural Gas	1.05E-05	1.60E-02	0.9754
CN	Corn	-2.48E-02	7.43E-01	0.7615
LH	Lean Hogs	-2.39E-03	9.43E-02	0.8182
GC	Gold Comex	-3.30E-02	6.72E-01	0.8143
CT	Cotton #2	-3.17E-04	9.40E-02	0.8694

Table 2 – Inertia per futures contract

FIGURES

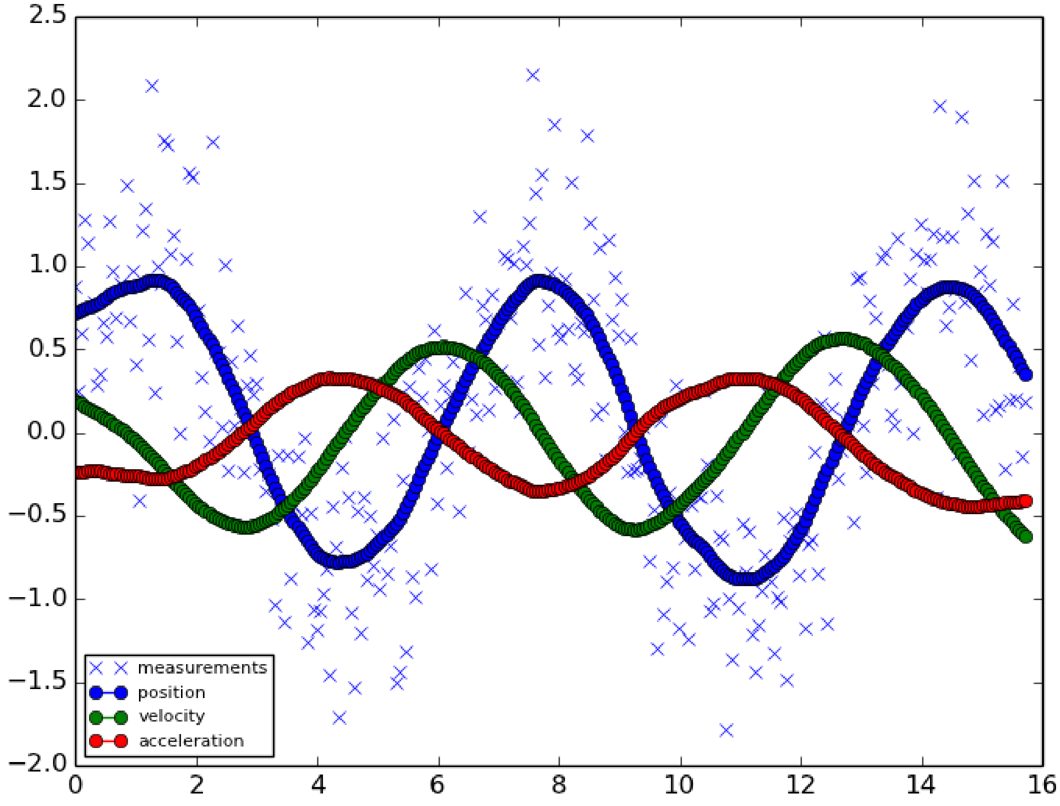

Figure 1 – KCA's estimated mean states

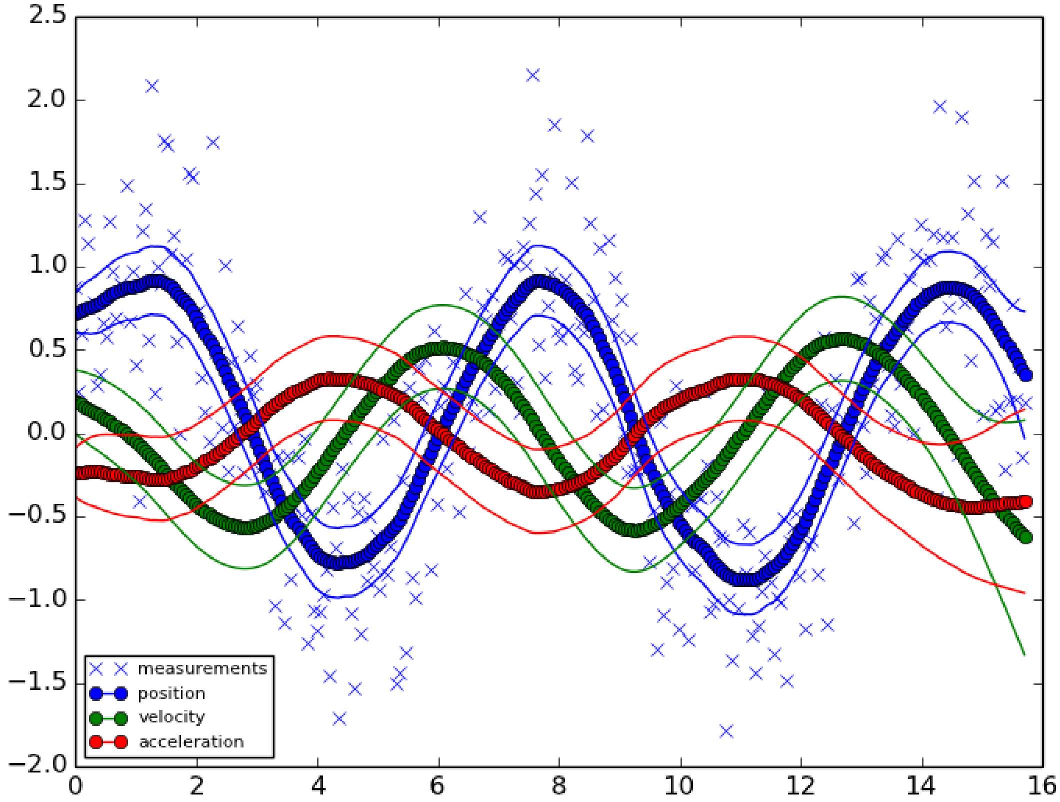

Figure 2 – KCA's means states with confidence intervals

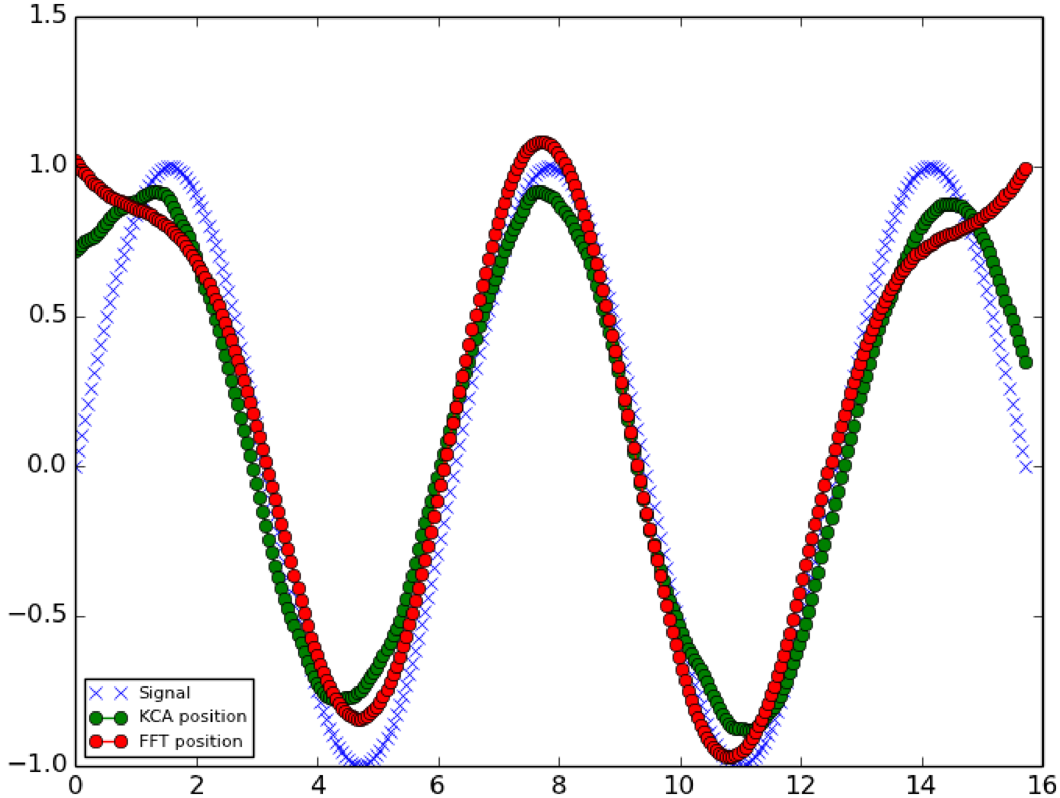

Figure 3 - KCA vs. FFT

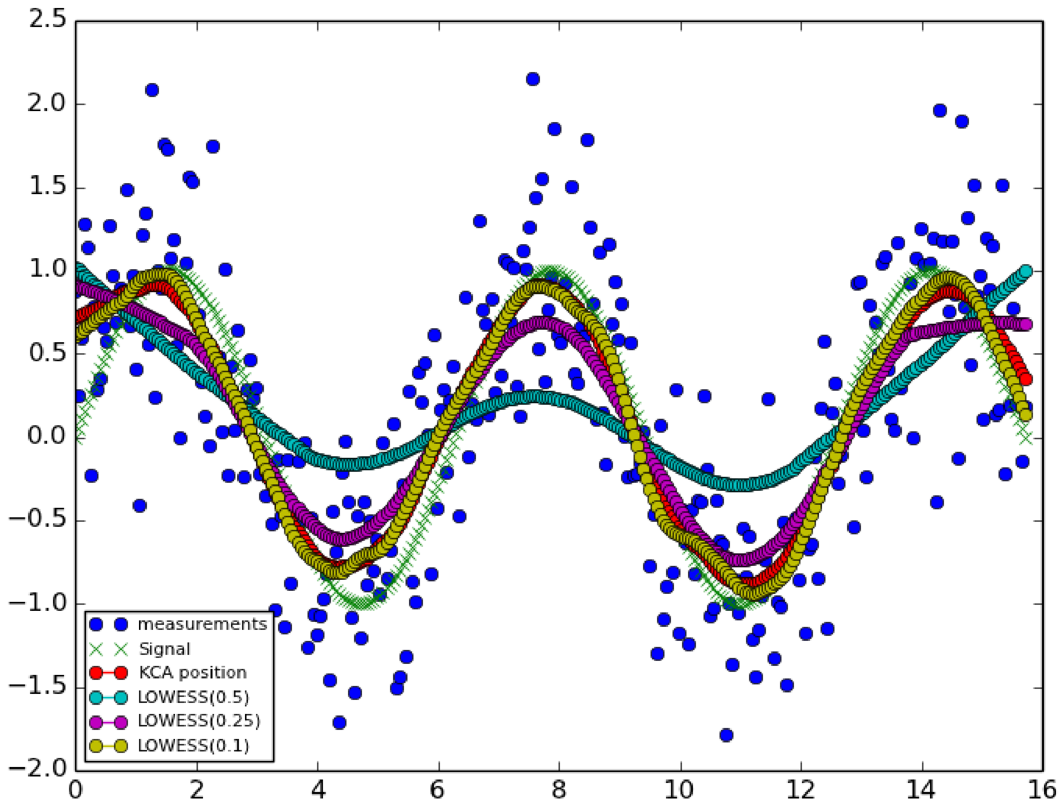

Figure 4 - KCA vs. LOWESS

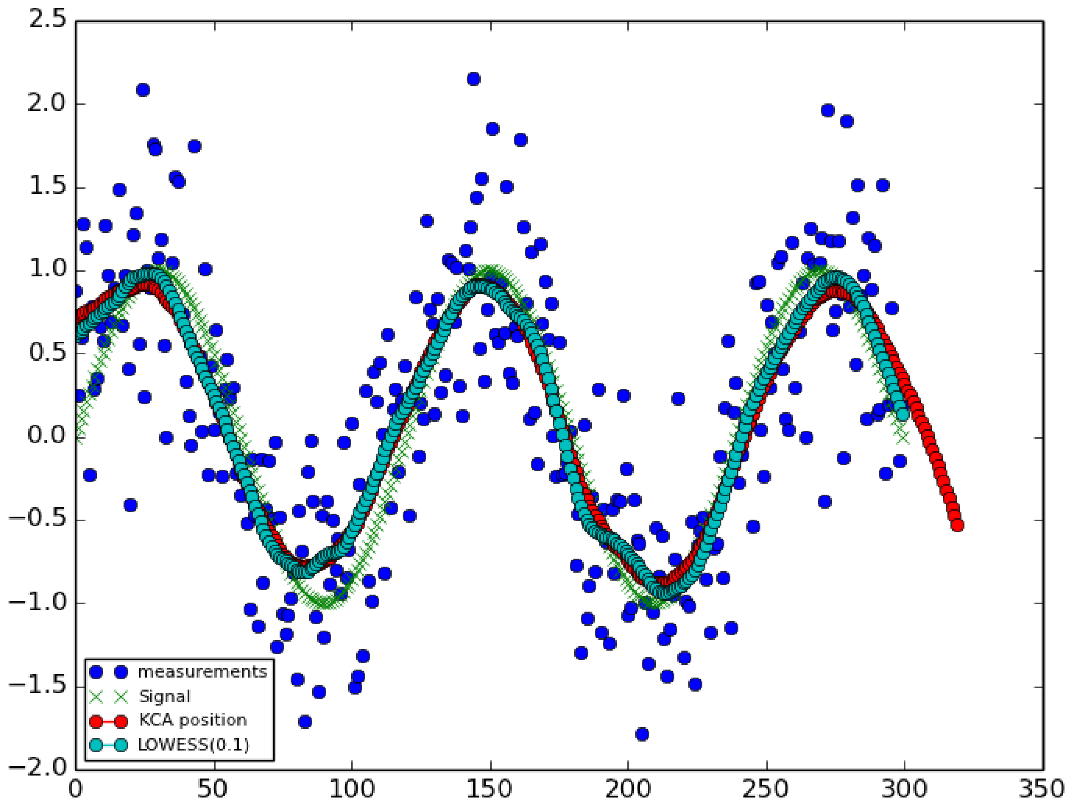

Figure 5 - KCA forecasts, 20 steps forward

REFERENCES

- Bailey, D., J. Borwein, M. López de Prado and J. Zhu (2014a): "Pseudo-Mathematics and Financial Charlatanism: The Effects of Backtest Overfitting on Out-Of-Sample Performance." *Notices of the American Mathematical Society*, 61(5), May. Available at: http://ssrn.com/abstract=2308659
- Bailey, D., J. Borwein, M. López de Prado and J. Zhu (2014b): "The Probability of Backtest Overfitting." Working paper. Available at: http://ssrn.com/abstract=2326253
- Ball, R. and S. Kothari (1989): "Nonstationary expected returns: Implications for tests of market efficiency and serial correlation in returns." *Journal of Financial Economics* 25, pp. 51-74.
- Brown, R. and P. Hwang (1997): *Introduction to Random Signals and Applied Kalman Filtering*. Wiley, 3rd Edition.
- Chan, K. (1988): "On the contrarian investment strategy." *Journal of Business* 61, pp. 147-163.
- Choi, J. (2014): "Physical approach to price momentum and its application to momentum strategy." *Physica A: Statistical Mechanics and its Applications* 415, pp. 61-72.
- Chopra, N., J. Lakonishok, and J. Ritter (1992): "Measuring abnormal returns: Do stocks overreact?" *Journal of Financial Economics*, 31, pp. 235-268.
- Cleveland, W. (1979): "Robust Locally Weighted Regression and Smoothing Scatterplots." *Journal of the American Statistical Association*, 74(368), pp. 829-836.
- De Bondt, W. and R. Thaler (1985): "Does the stock market overreact?" *Journal of Finance* 40, pp.793-805.
- De Bondt, W. and R. Thaler (1987): "Further evidence of investor overreaction and stock market seasonality." *Journal of Finance* 42, pp. 557-581.
- Easley, D., R. F. Engle, M. O'Hara and L. Wu (2008): "Time-Varying Arrival Rates of Informed and Uninformed Traders." *Journal of Financial Econometrics*, 6(2), pp. 171-207.
- Easley, D., M. López de Prado and M. O'Hara, (2011): "The Microstructure of the Flash Crash." *Journal of Portfolio Management*, 37(2), Winter, 118-128.
- Easley, D., M. López de Prado and M. O'Hara (2012a): "Flow Toxicity and Liquidity in a High Frequency World." *Review of Financial Studies*, 25(5), 1547-1493. Available at http://ssrn.com/abstract=1695596
- Easley, D., M. López de Prado and M. O'Hara (2012b): "The Volume Clock: Insights into the High Frequency Paradigm." *Journal of Portfolio Management*, 39(1). Available at http://ssrn.com/abstract=2034858
- Easley, D., M. López de Prado and M. O'Hara (2012c): "Optimal Execution Horizon." *Mathematical Finance*, forthcoming. Available at http://ssrn.com/abstract=2038387
- Easley, D., M. López de Prado and M. O'Hara (2012d): "Bulk Volume Classification." Working paper. Available at http://ssrn.com/abstract=2038387
- Grinblatt, M. and S. Titman (1989): "Mutual fund performance: An analysis of quarterly portfolio holdings." *Journal of Business*, 62, pp. 394-415.

- Grinblatt, M. and S. Titman (1991): "Performance measurement without benchmarks: An examination of mutual fund returns." Working paper, University of California at Los Angeles.
- Jarrow, R. and S. Hogan, T. Melvyn and M. Warachka (2003): "Testing Market Efficiency Using Statistical Arbitrage with Applications to Momentum and Value Strategies." Working paper, SSRN.
- Jegadeesh, N. (1990): "Evidence of predictable behavior of security returns." *Journal of Finance* 45, pp. 881-898.
- Jegadeesh, N. (1992): "Does market risk really explain the size effect?" *Journal of Financial and Quantitative Analysis,* 10, pp. 337-35l.
- Jegadeesh, N., and S. Titman (1993): "Returns to Buying Winners and Selling Losers: Implications for Stock Market Efficiency." *Journal of Finance*, 48(1), pp. 65-91.
- Jegadeesh, N., and S. Titman (1999): "Profitability of Momentum Strategies: An Evaluation of Alternative Explanations." NBER Working paper #7159.
- Jegadeesh, N., and S. Titman (2001): "Momentum." Working paper, SSRN.
- Lehmann, B. (1990): "Fads, martingales and market efficiency." *Quarterly Journal of Economics,* 105, pp. 1-28.
- Lo, A. and C. MacKinlay (1990): "When are contrarian profits due to stock market overreaction?" Review of Financial Studies, 3, pp. 175-205.
- Lux, H. (2000): "The Secret World of Jim Simons." Institutional Investor Magazine, November.
- Moskowitz, T., Y. Ooi and Pedersen, L. (2012): "Time series momentum." *Journal of Financial Economics*, 104, pp. 228-250.
- Patterson, S. and J. Strasburg (2010): "Pioneering Fund Stages Second Act." *Wall Street Journal*, 03/16/2010.
- Poundstone, W. (2005): *Fortune's Formula: The Untold Story of the Scientific Betting System that Beat the Casinos and Wall Street*. Hill and Wang. 1st Edition.
- Strang, G. (1994): "Wavelets." *American Scientist*, 82(3), May-June, p. 253.
- Welch and Bishop (2006): "An Introduction to the Kalman Filter." Technical Report 95-041. Department of Computer Science, University of North Carolina at Chapel Hill.
- Zarowin, P. (1990): "Size, seasonality, and stock market overreaction." *Journal of Financial and Quantitative Analysis,* 25, pp. 113-125.