Optimal dynamic asset allocation using a non-linear discrete time liquidity driven microstructure market model and extended Kalman filtering

Anders Stromberg

Lund University
Department of Economics

Supervisor: Associate Professor, Hossein Asgharian

Autumn 2005

Mailing address:
Nationalekonomiska Institutionen
Box 7082
SE-220 07 LUND
Sweden

Table of Contents

Page

List of Figures

Figure Page

Abstract

This paper use a discrete time microstructure model which considers excess demand and market liquidity as two unobservable state variables as determinants whether the market is overvalued or undervalued. The model expresses the variation of conditional variance of price, were the amplitude of the price changes is dependent on the liquidity of the market. It is shown that the filtered process of hidden excess demand and its liquidity is meaningful to apply in an asset allocation strategy and is efficient in terms of producing a residual money gain compared to a passive strategy.

Optimal dynamic asset allocation using a non-linear discrete time liquidity driven microstructure market model and extended Kalman filtering

I. Introduction

A basic assumption within economics is that demand and supply of a good affect its price. Excess demand drives the price up and excess supply makes it fall. Goods may also be financial products.

The current most acceptable theorem of price movements in financial markets is that they are random walks with predictive errors close to white noise and this is due to the Markov properties of financial time series. Hence, much of financial theory is based on this assumption.

However, an electronification of market places combined with increased collaboration between the financial industry and the academic world, have changed the financial model framework in the last twenty years or so. The theories have been extended to look closer on the departure from a random walk and many researchers and practitioners, by modelling the prices underlying behavior through market microstructure analysis (c.f. (19)), have found evidence supporting a seemingly natural and frequent non-random behavior in different financial markets (c.f. (15)).

The quest for answers and the desire to technically explore those "inefficiencies" have introduced more sophisticated methods into the financial markets and scholars from other disciplines, mainly mathematicians and physicists, have therefore shown an increasing interest in research on financial dynamics. Lacking traditional economic schooling, they have contributed with fresh input on how to model financial time series that is not limited by the previous contributions in this field.

In several articles and conference proceedings, (8), (20), (21), (22), (23), and (24) the physicists Mitsunori Iino, Tohru Ozaki, and Hui Peng worked with deriving an alternative way to look at a price process. Their work was inspired by Jean-Philippe Bouchaud and

Rama Cont's model of excess demand (2), which in turn used some insights made by Albert Kyle (13). They assumes there exists a diffusion of information between informed markets participants and less informed participants which causes trends in financial markets that may exists at different degrees.

If there exists an underlying demand and supply, it should be possible to model its dynamics in relation to the whole price movement. One could use techniques developed for the control theory community to determine its current "clean" state in every sample. If the assumption is not valid, the extracted process's predictive power should resemble a random walk process, with residuals close to white noise.

As the results presented by Peng et. al. are conflicting with a traditional viewpoint of an efficient market it is interesting to apply their model on different data, in order to achieve similar results. Hence, the purpose of this paper is to replicate studies made by Peng and Ozaki (21), (22). That is, to find a hidden process of excess demand in financial price data that is meaningful to use in a dynamic asset allocation control strategy.

This paper will mainly follow Peng et. al.'s article in The European Physical Journal B (22), in which a discrete time version of their stochastic volatility model is presented. The main purpose, is to a greater extent explaining how the model is derived preceded by explaining the underlying theory and cause of action. Another purpose, is to present alternatives to some parts of the analyzed model, as well as, to test and analyze the model with a different data set.

II. Theoretical framework

2.1 Dynamic systems

A system is defined as a process in which an input signal is converted into an output signal. Many existing physical processes, such as the movement of a vehicle, can be approximately modelled by a linear control system for which an external influence, i.e. an input signal variable u, controls the output signal variable y of the system.[1]

A linear control system is a system were two exogenous inputs simultaneously affect the system but do not interact within the system. That is, input u_1 produces output y_1 and input u_2 produces output y_2. Such system is therefore characterized by the principle of supposition which is that input $u_1 + u_2$ produces the output $y_1 + y_2$. Systems that not have these feature are called nonlinear.

The interaction of inputs with the internal components of a system is constituted by the state variable x. The state of a system can be described as an aggregation of the most relevant information that, both necessarily and sufficiently, describing the dynamics of it; i.e. vector x contains all the information about the present state of the system. Its complexity is determined by the dimension of the state vector. Therefore, from a state variable it is possible to determine the future dynamics of a system, given that all future inputs are known. The dynamics are either represented as a difference equation (in discrete time) or as a differential equation (in continuos time).

In a model, the dynamics are either explicitly known or not. A known dynamic is often referred to as a "white box model" which contains an explicit description of the inner workings of the system. An unknown, is referred to as a "black box model" and is one in which we can observe input and output but not the internal structure of the system.

Often we cannot measure x directly. Instead, we observe y_k at discrete time steps, t_k, which is a function of the unobservable x_k (subscript k is a shorthand for notation t_k). The purpose of the state analysis is therefore to infer the relevant properties of the

[1] Engineers and economists tend to use the term signal differently. Engineers look at it as a transmitter, while economists looks at it as a receiver, which observes a signal and then uses it to estimate some other variable. (4)

state x_k from a knowledge of the observations $y_1 \dots y_n$. This analysis is mainly done by computational algorithms which are based on recursion; in which a function calls itself.

In practise, a common way of conducting such analysis is to describe the input-output relation in a so called state space form. In the linear dynamic case, were the signals are assumed related by a linear system, the state space model in discrete time of the control system is constituted by two linear difference equations,

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \text{ =: State Transition Equation,} \\ y_k &= Cx_k + Du_k \text{ =: Observation Equation.} \end{aligned} \tag{1}$$

A, B, C, and D are all matrices and determines the relationship between the state, input, and output variables, i.e. the internal structure or dynamics of the system. The first equation is called the State Transition Equation, System Equation, or State Equation and describes the behavior of the state variable, x, in time. Using an Itô interpretation, this is a (first order) Markov process. The second equation is called the Observation Equation and describes how the state variable is mapped into the observed variables y. The two equations constitutes a state space model.

Definition 1 *$\{y_k\}$ has a state-space representation if there exists a state-space model for $\{y_k\}$ given by the State Transition Equation and the Observation Equation.*

A major advantage of representing a dynamic system in state space form is its Markov properties. The rationale for this is the following; a system is said to be completely state observable or full-state observable if every state x_k can be determined or reconstructed from the observation of y_k over a finite time interval. Through the standard Markov assumption (i.e. the probability of the next state is conditioned on only the current state, and action is independent of the past) the last observation (state) summarizes the information of all previous states and thus reveal the true state of the world. This property is also rational in the sense of model reduction. Therefore, a state is analogous to principal component in multivariate statistics.(17)

Another major advantages with the state space form is that it allows for unobserved variables to be incorporated and estimated along with an observable model. Estimation within a state space representation has therefore two aspects:

- measuring the underlying unobserved state which summarizes a process,
- estimation of unknown parameters (maximum likelihood).

2.1.1 Noise is introduced. To be able to control a system, we need the true current state of the system, i.e. the clean variable x. Without any type of noise in a system, control is fairly easy. However, a common problem is when we measure y, the observation is often corrupted by several disturbances. Therefore, it is more realistic to introduce noise terms in our state space equation which describes the nature of this disturbances and this leads us to a stochastic state-space model. The linear stochastic state space model in discrete time is given by the equations:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + w_k, \\ y_k &= Cx_k + Du_k + n_k, \end{aligned} \tag{2}$$

where w_k is called the system or process noise which drives the system and n_k represents the observation or measure noise. In order to identify the unobservable state vector x, given the knowledge that there is noise, we need a method of obtaining an estimator that gives an accurate estimate for the true state out of the noisy observation data.

In control theory, a so called filter is usually used to remove noise from a signal which is corrupted by the measuring process itself. Given our model (2), filtering is defined as:

Definition 2 *Filtering applies to the problem of estimating the state of a discrete time controlled process that is represented by a linear stochastic difference equation,* $x_{k+1} = Ax_k + Bu_k + w_k$ *with measurement* y_k *that is* $y_k = Cx_k + Du_k + n_k$*. In a state space representation, filtering is the same as estimating a clean state vector.*

Intuitively, we need an estimator that is efficient; that has an expected value equal to the state and which has the smallest possible error variance.

2.1.1.1 Kalman filter for linear Gaussian problems. Rudolph E. Kalman introduced his version of such efficient filter for the control theory community in 1960 (11). He proposed a recursive algorithm which makes optimal use of observable variables with Gaussian errors, y, to continuously update the best current state estimate of unobservable, but correlated variable, x. This was possible as its complete distribution is characterized by its mean and variance.

Hence, the Kalman algorithm assumes that, at any time k, the noise terms - process and observation - are independent zero-mean (i.e. does not change the average intensity level) white Gaussian noise processes with certain covariance properties. If $w = \Omega$, and $n = \Gamma$, in model (2), these properties could be described as $\begin{pmatrix} \Omega \\ \Gamma \end{pmatrix} = \begin{pmatrix} Cov_1 & Cov_{12} \\ Cov_{12}^T & Cov_2 \end{pmatrix}$, $1 = \Omega$, $2 = \Gamma$; which are uncorrelated with x_k and u_k. Noise covariance matrices of Q and R are defined by:

- Process noise covariance, $Q \equiv \mathbb{E}\left(\Omega\Omega^T\right)$ and
- measurement noise covariance $R \equiv \mathbb{E}\left(\Gamma\Gamma^T\right)$.

Therefore, we can write our model (2) as;

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + \Omega, \\ y_k &= Cx_k + Du_k + \Gamma, \end{aligned} \tag{3}$$

with $(\Omega) \sim N(0, Q)$ as the process noise and $(\Gamma) \sim N(0, R)$ as the measurement noise, where $y_k \in R^m$ models the noise in the measurement.

Ω is assumed to have a zero mean and does therefore not influence the estimate of the mean equation x_{k+1}. Neither, we assume observation function to have a control, hence D drops out of the model. Again, y_k is the process measurement vector and C is a measurement matrix which relates state vector x_k to measurement y_k.

Defining $\hat{x}_k^- \in R^n$ as our á priori state estimate and $\hat{x}_k \in R^n$ as our á posteriori state estimate, then estimate errors are expressed as $e_k^- \equiv x_k - \hat{x}_k^-$, and $e_k \equiv x_k - \hat{x}_k$. Associated covariance matrices are defined as:

$$S_k^- = \mathbb{E}\left(\left(x_k - \hat{x}_k^-\right) \quad \left(x_k - \hat{x}_k^-\right)^T \right) = \mathbb{E}\left(e_k^- \quad e_k^{-T} \right) \tag{4}$$

for the á priori estimate error covariance, and

$$S_k = \mathbb{E}\left(\left(x_k - \hat{x}_k\right) \quad \left(x_k - \hat{x}_k\right)^T \right) = \mathbb{E}\left(e_k \quad e_k^T \right) \tag{5}$$

for the á posteriori estimate error covariance.

By creating a linear combination of an á priori estimate with a weighted difference between the actual measurement y_k and a measurement prediction, $C\hat{x}_k$, we can obtain the posteriori state estimate.

$$\hat{x}_k = \hat{x}_k^- + K\left(y_k - C\hat{x}_k^-\right) \tag{6}$$

We call the difference $\left(y_k - C\hat{x}_k^-\right)$ the measurement innovation or the measurement residual, $\hat{\Gamma}_k$, and it is our estimate for measurement noise Γ, with covariance

$$\hat{V}_k = \mathbb{E}\left(\hat{\Gamma}_k \quad \hat{\Gamma}_k^T \right), \tag{7}$$

and is a residual of zero mean. (7) can also be formulated in our state space model as

$$\hat{V}_k = CS_k^- C^T + R. \tag{8}$$

The matrix K in (6) is called the Kalman gain. It is a blending or gain factor for the new state estimate, chosen to minimize the á posteriori error covariance S_k. Intuitively, it is a factor that tells the trustfulness of our prediction. Practically, by substituting $\hat{x}_k = \hat{x}_k^- + K\left(y_k - C\hat{x}_k^-\right)$ into our definition of e_k and latter into our á posteriori estimate error covariance. Hence, it follows that

$$S_k = \mathbb{E}\left(x_k - \hat{x}_k^- + K\left(y_k - C\hat{x}_k^-\right) \quad \left(x_k - \hat{x}_k^- + K\left(y_k - C\hat{x}_k^-\right)\right)^T \right). \tag{9}$$

Our optimization criterion is the sum of the variances of the á posteriori estimation error, therefore we take the derivative of the trace with respect to K, set it equal to zero and solve for K. The result of this measurement weighting can be represented as

$$\begin{aligned} K_k &= AS_k^- C^T \left(\hat{V}_k\right)^{-1} \\ &= \frac{AS_k^- C^T}{CS_k^- C^T + R}. \end{aligned} \tag{10}$$

A intuitive way of thinking about K is that when the measurement error covariance R approaches zero, the observable measurement taken, y_k, is trusted more, meanwhile the predicted one, $C\hat{x}_k^-$, is trusted less. However, as the á priori estimate error covariance S_k^- goes towards zero, the actual measurement, y_k, is trusted less while the predicted measurement, $C\hat{x}_k^-$, is trusted more. In short, a large K gives the observation lot of creditability whereas a small K gives the measurement y small credibility when computing the next $\hat{x}_k$.

Our prediction, the á priori estimate error covariance, can alternatively be formulated in our state space as

$$S_k^- = AS_{k-1}A^T + Q$$

and a derived version of the á posteriori estimate covariance matrix, see e.g. (6) can be presented as,

$$S_k = (I - KC)\, S_k^- . \tag{11}$$

Next step in the filter operation is to update and correct the measurement. We do this by updating the state estimate with the measurement y_k to obtain á posteriori state estimate, equation (6), and update the error covariance (11).

2.1.1.2 Kalman Filter equations. Hence, the filter equations, in the discrete case, can be summarized in three steps (five bullets), predict, weighting, and correction.

- Time update or prediction of the á priori state ahead, i.e. state estimate of conditional mean. Ω has zero mean, therefore it is omitted.

$$\hat{x}_k = A\hat{x}_{k-1}^{-} + Bu_{k-1} \tag{12}$$

- Project the error covariance, the conditional á priori state covariance ahead.

$$S_k^{-} = AS_{k-1}A^T + Q. \tag{13}$$

- Measurement weighting by compute the Kalman gain

$$K_k = AS_k^{-}C^T \left(CS_k^{-}C^T + R\right)^{-1}. \tag{14}$$

- Measurement update, correct the á posteriori estimates with noise measurement of the process y_k. This gives us an á posteriori state estimate $\hat{x}_k$,

$$\hat{x}_k = A\hat{x}_k^{-} + K\left(y_k - C\hat{x}_k^{-}\right). \tag{15}$$

- Update the error covariance to obtain an á posteriori error covariance S_k,

$$S_k = (I - KC)\, S_k^{-}. \tag{16}$$

This constitutes the Kalman filter. It needs initial estimates of $\hat{x}_{k-1}$ and S_{k-1}^{-} in order to start, as the process is recursively updated by using previous á posteriori estimate to propagate the new á priori estimate. The equations assume zero mean errors and that all errors and noises are normally distributed.

2.1.2 Extended Kalman Filter for nonlinear Gaussian problems. However, if the model is nonlinear, the Kalman filter has to be extended through a linearization procedure. The linearization is necessary in order to analytically propagate the Gaussian random variable (GRV) representation. The resulting filter is referred to as the Extended Kalman Filter (EKF).

In the basic framework for an EKF we estimate, as previous, a state of a discrete-time system. However, the system is now represented by a nonlinear dynamic model, f and h, which are assumed known (i.e. computable). Our nonlinear dynamic system in state space representation becomes:

$$\begin{aligned} x_{k+1} &= f\left(x_k, u_k\right) + w_k, \\ y_k &= h\left(x_k, z\right) + n_k. \end{aligned} \tag{17}$$

$f\left(x_k, u_k\right)$ is a nonlinear transition matrix, i.e. known system dynamics and $h\left(x_k, z\right)$ denotes a nonlinear measurement matrix, i.e. an observation function. h is nonlinear map which is parameterized by the known parameter vector z.

To perform the EKF, the nonlinear dynamics are, firstly, approximated by a first order linearization of the state space representation, using partial derivatives of f and h, at each time instant around the most recent state estimate. Secondly, the standard Kalman filter equations are applied on the linearization. These two steps constitute the EKF.

III. Ozaki and Peng's model

Bouchaud and Cont (2) use a phenomenological approach to describe market dynamics in a financial market by using a simple nonlinear Langevin equation. This approach is often used in order to describe many physical phenomena. They model financial asset price dynamics by deriving a microstructure model that focuses on the phenomenon of demand and supply on a time scale where a collective behavior of the market and impact on prices can be represented by a few stochastic dynamic equations. On this time scale, researchers have found regularities which are common between markets with different microstructures, i.e. where they have found price variations to be strongly non-Gaussian c.f. (15).

Iino and Ozaki (8) and Peng et. al. (22) provide a concise notation for Bouchaud's and Cont's model. Let P_t denote the financial asset price at time t. Let λ, $\lambda \neq 0$, denote the inverse of market depth or market liquidity[1], and let us call ϕ_t^+ the demand and ϕ_t^- the supply. Demand and supply is the instantaneous size of the demand and supply order-flow innovation. We now define ϕ_t as the excess demand for the financial asset at time t (i.e. the difference between demand and supply), with $\phi_t \equiv \phi_t^+ - \phi_t^-$. Bouchaud's and Cont's model can then be written as

$$dP_t = \lambda \phi_t dt. \tag{18}$$

We see that the financial asset price process in (18) is driven by the excess demand and that the magnitude in price changes is determined by the market liquidity, i.e. the instantaneous excess demand required to move the financial asset price by one unit. This simple model works in accordance with the postulates of the most fundamental of all economic models, the demand-price model; in an over-valued market, i.e. when $\phi_t > 0$, the asset price will tend to be pushed up and in an under-valued market, i.e. when $\phi_t < 0$, it will tend to be pushed down. In a market with high liquidity, i.e. when $\frac{1}{\lambda}$ is low, only a small price change will be necessary to absorb a given excess demand, and in a market

[1] Some researchers propose a much wider definition of market liquidity, e.g. Fisher Black defines a liquid market as one which is almost infinitely tight, which is not infinitely deep and which is resilient enough so that prices eventially tend towards their underlying value.(13)

with low liquidity, i.e. when $\frac{1}{\lambda}$ is high, large price change will be needed to absorb a given excess demand.

Certainly a simple model, alas ϕ_t and λ are often immeasurable variables and thus equation (18) cannot be used for practical modelling in many markets but only as an intellectual exercise.

Based on this equation, Iino and Ozaki (8) propose a system of stochastic differential equations where λ and ϕ_t are treated as unobservable state variables. This made it possible for the authors to model these two variables in relation to a price process; thus made (18) useful for practical purposes.

Let $W_{i,t}$, $i = 1, \ldots, 3$, denote mutually independent Wiener-processes, and α_1, α_2, β_1, β_2, γ_1, and γ_2 denote constant model parameters. The model of Bouchaud and Cont can then be written as

$$\begin{aligned} dP_t &= \lambda_t \phi_t dt + \lambda_t dW_{1,t}, \\ d\phi_t &= (\alpha_1 + \beta_1 \phi_t)dt + \gamma_1 dW_{2,t}, \\ d\log\lambda_t &= (\alpha_2 + \beta_2 \log\lambda_t)dt + \gamma_2 dW_{3,t}. \end{aligned} \tag{19}$$

The first equation in (19) describes the financial asset price process. It is notable that the conditional expected value and conditional variance of the financial asset price process is given by

$$\mathbb{E}_t(dP_t) = \mathbb{E}_t(\lambda_t \phi_t dt + \lambda_t dW_{1,t}) = \mathbb{E}_t(\lambda_t \phi_t dt) + \mathbb{E}_t(\lambda_t dW_{1,t}) = \lambda_t \phi_t dt,$$

respectively,

$$Var_t(dP_t) = V_t(\lambda_t \phi_t dt + \lambda_t dW_{1,t}) = V_t(\lambda_t dW_{1,t}) = \lambda_t^2.$$

This tells us that (19) is the same as (18), with a noise added and letting λ become time variant.

The second equation model the process of the immeasurable hidden excess demand variable ϕ_t. It is notable that, with the first R.H.S. term, Iino and Ozaki have modelled a mean reversion effect or stationary process around α_1.

The third equation in (19) models the dynamics of liquidity, which together with the first line resemble an EGARCH which is a model for heteroscedastic volatility. Its property, as such, catches asymmetries in financial time series, c.f. (25) and (18). Consequently, the third equation tells the first's its volatility dynamics.

According the authors, model (19) offers a better representation of the internal characteristics of a financial price varying process. They find that the information offered by the hidden variables posses better stability than the market trend information obtained for the mere prediction of a price process.

It is notable that they model an underlying process that directly affects and explains, the change of a price. This contrasts to a traditional model framework were most financial stochastic processes focus on modelling the price dynamics in itself and its conditional variance.

3.1 A discrete time microstructure model

However, when Iino and Ozaki tested the model (19) with a local linearization technique they found that it was very sensitive and computationally heavy, hence inappropriate for some practical applications. Therefore, Hui Peng, Tohru Ozaki, and Valerie Haggan-Ozaki in (21), and (22), used an alternative version which discretize (19) using an Euler-Maruyama scheme.

Consider a Langevin equation on the form

$$dx = f(x) + g(x)\xi(t),$$

for which the stationary distribution is unknown. Solving such a system often uses time discretization and numerical solution, where we let the system find its "equilibrium" and reconstruct the stationary distribution from the numerical trajectories.

When $\xi(t)$ is assumed to be Gaussian white noise, the Euler-Maruyama scheme can be used to solve this system. In this scheme, the numerical trajectory is generated by

$$x_{n+1} = x_n + hf(x_n) + \sqrt{h}g(x_n)\zeta_n,$$

where h is the time step and ζ_n is a discrete Gaussian white noise. (12)

With $h = 1$, $f = \mathbf{A}$, and $g = \eta$ in the system (19) and letting the conditional variance λ_t^2 be directly modeled of the price P_t, Peng et. al. obtain the following discrete time model:

$$\begin{aligned} P_k &= P_{k-1}\lambda_{k-1}\phi_{k-1} + \gamma_3\lambda_{k-1}\xi_{1,k}, \\ \phi_k &= \alpha_1 + (1+\beta_1)\phi_{k-1} + \gamma_1\xi_{2,k}, \\ \log\lambda_k^2 &= \alpha_2 + (1+\beta_2)\log\lambda_{k-1}^2 + \gamma_2\xi_{3,k}, \end{aligned} \tag{20}$$

where $\xi_{i,k} \in N(0,1)$, $i = 1, \ldots, 3$, are independent Gaussian white noise processes. The authors also add a new constant parameter, γ_3, to the discretized system which describes the relation between P_k, λ_{k-1} and ϕ_{k-1}.

According to Peng et al. the resulting model (20) produces similar results to (19), while offering more flexibility and stability with a considerable reduction in computational burden.

In order to estimate the derived discrete time microstructure model using the Kalman filtering and maximum likelihood; a state space model of the parameters of the non-observable state variables has to be built.

To construct a signal equation that fulfills the observability condition within the state space theory (as mentioned above), Peng et al. square the first equation in (19), and ignoring the higher order of terms according to rules of Itô calculus, i.e.

$$(dP_t)^2 \approx (\lambda_t\phi_t dt + \lambda_t dW_{1,t})^2 = (\lambda_t\phi_t dt)^2 + 2\lambda_t\phi_t dt\lambda_t dW_{1,t} + (\lambda_t dW_{1,t})^2,$$

which together with the simplified rules of squares for Itô-calculus, i.e. $(dW)^2 = dt$ and $dWdt = (dt)^2 = 0$, gives us

$$(dP_t)^2 \approx \lambda_t^2 dt. \tag{21}$$

A Euler-Maruyama scheme applied to (21) and by taking the logarithms of it, we get

$$\log(P_{k+1} - P_k)^2 \approx \log \lambda_k^2. \tag{22}$$

Hence, Peng et al. regard P_k and $\log(P_{k+1} - P_k)^2$ as two separable observation variables and applies (22) to construct their model in state space form which fulfills the observability condition.

If $\boldsymbol{\theta}$ is a vector which includes all the constant parameters in (20), and P'_k denotes the systems-true, unobservable, asset price at time k, which together with

$$\begin{aligned}
\mathbf{X}_k &= \begin{pmatrix} P'_k \\ \phi_k \\ \log \lambda_k^2 \end{pmatrix}, \mathbf{A}(\mathbf{X}_{k-1}|\boldsymbol{\theta}) = \begin{pmatrix} 1 & \lambda_{k-1} & 0 \\ \frac{\alpha_1}{P'_{k-1}} & 1+\beta_1 & 0 \\ \frac{\alpha_2}{P'_{k-1}} & 0 & 1+\beta_2 \end{pmatrix}, \\
\Omega_k &= \begin{pmatrix} \gamma_3 \lambda_{k-1} \xi_{1,k} \\ \gamma_1 \xi_{2,k} \\ \gamma_2 \xi_{3,k} \end{pmatrix},
\end{aligned}$$

and

$$\mathbb{E}\left(\Omega_k \Omega_k^T\right) = \begin{pmatrix} \gamma_3^2 \lambda_{k-1}^2 & 0 & 0 \\ 0 & \gamma_1^2 & 0 \\ 0 & 0 & \gamma_2^2 \end{pmatrix} =: \mathbf{Q}_k,$$

form the state transition equation of model (20) in state space. This could be condensed to;

$$\mathbf{X}_k = \mathbf{A}(\mathbf{X}_{k-1}|\boldsymbol{\theta})\mathbf{X}_{k-1} + \Omega_k, \tag{23}$$

where $\Omega_k \in N_1(\mathbf{0}_1, \mathbf{Q}_k)$.

The observation function for model (20) is in state space formulated as;

$$\begin{aligned}
\mathbf{Y}_k &= \begin{pmatrix} P_k \\ \log(P_k - P_{k-1})^2 \end{pmatrix}, \mathbf{C}(\mathbf{X}_k|\boldsymbol{\theta}) = \begin{pmatrix} 1 & 0 & 0 \\ \frac{\delta}{P_k} & 0 & 1 \end{pmatrix}, \\
\Gamma_k &= \begin{pmatrix} \epsilon_{1,k} \\ \epsilon_{2,k} \end{pmatrix},
\end{aligned}$$

with covariance matrix

$$\mathbb{E}\left(\Gamma_k \Gamma_k^T\right) = \begin{pmatrix} \epsilon_1^2 & 0 \\ 0 & \epsilon_2^2 \end{pmatrix} =: \mathbf{R}_{\boldsymbol{k}}.$$

Hence, in a more condensed form, expressed as

$$\mathbf{Y}_k = \mathbf{C}(\mathbf{X}_k|\boldsymbol{\theta})\mathbf{X}_k + \Gamma_{\boldsymbol{k}}, \tag{24}$$

where $\Gamma_{\boldsymbol{k}} \in N_2(\mathbf{0}_2, \mathbf{R}_{\boldsymbol{k}})$.

If we expand (24) we get

$$\mathbf{Y}_k = \begin{pmatrix} P_k \\ \log(P_k - P_{k-1})^2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ \frac{\delta}{P_k'} & 0 & 1 \end{pmatrix} \begin{pmatrix} P_k' \\ \phi_k \\ \log \lambda_k^2 \end{pmatrix} + \begin{pmatrix} \epsilon_{1,k} \\ \epsilon_{2,k} \end{pmatrix},$$

i.e.

$$\begin{aligned}
P_k &= P_k' + \epsilon_{1,k}, \qquad (25) \\
\log(P_t - P_{k-1})^2 &= \delta + \log \lambda_k^2 + \epsilon_{2,k}.
\end{aligned}$$

Peng et al consider $\epsilon_{i,k}$ $i = 1, 2$, together with δ as constants, with the latter as an adjustment for the bias in the variable transformation stated in (22). The model developed above is accordingly simpler in its presentation (and function) than the previous model, thus easier to generalize.

3.2 Model estimation

Peng et al. use $\mathbf{A}(\hat{\mathbf{X}}_{k|k}|\boldsymbol{\theta})$ and $\mathbf{C}(\hat{\mathbf{X}}_{k|k}|\boldsymbol{\theta})$ to approximate $\mathbf{A}(\mathbf{X}_k|\boldsymbol{\theta})$ and $\mathbf{C}(\mathbf{X}_k|\boldsymbol{\theta})$**,** and according the filter approach, described above, $\hat{\mathbf{X}}_k$ is calculated through an extended Kalman filtering scheme, starting with the prediction.

- $\hat{\mathbf{X}}_{k|k-1}$ is the conditional mean of X_k given $Y^{k-1} = (Y_1, \ldots, Y_{k-1})$. It is the matrix corresponding to our previous mentioned $\hat{x}_k^-$ in our description of the Kalman filter. $\hat{\mathbf{X}}_{k|k}$ is our posteriori state estimate at time k, hence $\hat{\mathbf{X}}_{k-1|k-1}$ is the same for $k-1$.

$$\hat{\mathbf{X}}_{k|k-1} = \mathbb{E}\left\{X_k|Y^{k-1}\right\} = A\left(\hat{\mathbf{X}}_{k-1|k-1}|\boldsymbol{\theta}\right)\hat{\mathbf{X}}_{k-1|k-1}. \tag{26}$$

$\hat{\Gamma}_k$ is the measurement residual (or innovation) between observation Y_k and predicted $\hat{Y}_k = C\left(\hat{\mathbf{X}}_{k-1|k-1}|\boldsymbol{\theta}\right)\hat{\mathbf{X}}_{k|k-1}$,

$$\hat{\Gamma}_k = Y_k - C\left(\hat{\mathbf{X}}_{k-1|k-1}|\boldsymbol{\theta}\right)\hat{\mathbf{X}}_{k|k-1}. \tag{27}$$

- S_k^- is the á priori conditional variance of X_k given $Y^{k-1} = (Y_1, \ldots, Y_{k-1})$. It project the error covariance ahead,

$$\begin{aligned} S_k^- &= \mathbb{E}\left\{\left(X_k - \hat{\mathbf{X}}_{k|k-1}\right)\left(X_k - \hat{\mathbf{X}}_{k|k-1}\right)^T\right\} \\ &= A\left(\hat{\mathbf{X}}_{k-1|k-1}|\boldsymbol{\theta}\right)S_{k-1}A\left(\hat{\mathbf{X}}_{k-1|k-1}|\boldsymbol{\theta}\right)^T + Q_k. \end{aligned} \tag{28}$$

- Then comes the weighting part of the Kalman filter equations. First we need the measurement or residual á posteriori covariance $\hat{\mathbf{V}}$,

$$\begin{aligned} \hat{\mathbf{V}} &= \mathbb{E}\left\{\ \hat{\Gamma}_k \quad \hat{\Gamma}_k^T\ \right\} \\ &= C\left(\hat{\mathbf{X}}_{k-1|k-1}|\boldsymbol{\theta}\right)S_k^- C\left(\hat{\mathbf{X}}_{k-1|k-1}|\boldsymbol{\theta}\right)^T + R_{k,} \end{aligned} \tag{29}$$

to be used when we doing measurement weighting by compute the Kalman gain, K_k,

$$K_k = S_k^- C\left(\hat{\mathbf{X}}_{k-1|k-1}|\boldsymbol{\theta}\right)^T\hat{\mathbf{V}}^{-1}. \tag{30}$$

- The measurement update or correction first starts with the update of conditional mean. Peng et al. uses following equation to update the state estimation with measurement Y^k,

$$\hat{\mathbf{X}}_{k|k} = \mathbb{E}\left\{X_k|Y^k\right\} = \hat{\mathbf{X}}_{k|k-1} + K_k\hat{\Gamma}_k. \tag{31}$$

- The last part, the update of the á posteriori state error conditional covariance S_k,

$$\begin{aligned} S_k &= \mathbb{E}\left\{\left(X_k - \hat{\mathbf{X}}_{k|k}\right)\left(X_k - \hat{\mathbf{X}}_{k|k}\right)^T\right\} \\ &= \left[I - K_k C\left(\hat{\mathbf{X}}_{k-1|k-1}|\boldsymbol{\theta}\right)\right] S_k^-. \end{aligned} \tag{32}$$

3.2.1 Parameter estimation. In order to determine the parameters $\boldsymbol{\theta}$ and the initial state $\mathbf{X}_{k|k-1} = \mathbb{E}\left\{X_k|Y_{k-1},\ldots,Y_1\right\}$ in the model, Peng et al. minimizing a log-likelihood function derived from equation (27).

Consider the probability density function for the normal distribution with mean μ and standard deviation σ,

$$f\left(x;\mu,\sigma\right) = \frac{1}{\sigma\sqrt{2\pi}}e^{-(x-\mu)^2/2\sigma^2}. \tag{33}$$

Assume Γ_k is two dimensional and normally distributed with covariance matrix $\hat{\mathbf{V}}$. The conditional distribution for Γ_k, (33) can then be written as

$$\Pr\left(\Gamma_k|Y_{k-1},\ldots,Y_1,\boldsymbol{\theta}\right) = \frac{1}{2\pi\sqrt{\left|\hat{\mathbf{V}}_k\right|}}e^{-\left(\frac{1}{2}\Gamma_k^T\hat{\mathbf{V}}_k^{-1}\Gamma_k\right)}. \tag{34}$$

Taking the $(-2)\log$ of the expression (34), a log likelihood model is obtained, where N is the number of data-points;

$$\begin{aligned} & (-2)\log\Pr\left(Y_{k-1},\ldots,Y_1|\boldsymbol{\theta}\right) \\ = & \sum_{k=1}^{N}(-2)\log\Pr\left(Y_k|Y_{k-1},\ldots,Y_1,\boldsymbol{\theta}\right) \\ = & \sum_{k=1}^{N}(-2)\log\Pr\left(\Gamma_k|Y_{k-1},\ldots,Y_1,\boldsymbol{\theta}\right) \\ = & \sum_{k=1}^{N}\left\{\log\left|\hat{\mathbf{V}}_k\right|+\hat{\Gamma}_k\left(\boldsymbol{\theta}\right)_k^T\hat{\mathbf{V}}_k\left(\boldsymbol{\theta}\right)^{-1}\hat{\Gamma}_k\left(\boldsymbol{\theta}\right)\right\}+2N\log 2\pi. \end{aligned} \tag{35}$$

To obtain the optimal parameters, $\boldsymbol{\theta}^*$, from the estimation of innovations and its covariance, can be done by minimizing (35) as,

$$\boldsymbol{\theta}^*=\arg\min_{\boldsymbol{\theta}}\sum_{k=1}^{N}\left\{\log\left|\hat{\mathbf{V}}_k\left(\boldsymbol{\theta}\right)\right|+\hat{\Gamma}_k\left(\boldsymbol{\theta}\right)_k^T\hat{\mathbf{V}}_k\left(\boldsymbol{\theta}\right)^{-1}\hat{\Gamma}_k\left(\boldsymbol{\theta}\right)\right\}+2N\log 2\pi. \tag{36}$$

The initial conditions $X_{0|0}$ and V_0 and the system observation noise variance R_k are also regarded as parameters by Peng et. al., to be estimated and are included in the parameter vector $\boldsymbol{\theta}$**.**

IV. Numerical study

4.1 Introduction

To test the validity of the derived process of excess demand and the accompanying proposed model, a numerical study was undertaken. The financial time series chosen, were the foreign exchange rate of the Danish and the Swedish Krona, both against the US Dollar during the periods 13/09/1989 - 13/05/1997 (2000 price observations), and 01/06/2005 - 31/03/2005 (1000 price observations). The exchange rate data was obtained from a Bloomberg Professional terminal. The observation data analyzed was calculated as $P_k = 100\left(\log(FX_k)\right)$, where FX_k is the foreign exchange daily closing spot price of DKK/USD and SEK/USD, i.e. at discrete time steps k. The log transformation was applied to stabilize the variance, c.f. (1).

We divide the two P_k series into two series each, in which the first part constitute the training part which is used to estimate the parameters in (36), and the threshold parameters for the strategy evaluation below, see equation (42). These *trained* parameters are then used to retrieve the filtered excess demand variable and its liquidity, which are then evaluated in the asset allocation model (43), with the trained threshold parameters.

To get a picture of the model's performance, the strategy is evaluated against a passive portfolio, were the model's trained parameters are used in the second, testing, part of the data, see figures 1, and 2.

As can be seen, the two chosen series differs in characteristics. The DKK/USD series constitutes of several local trends whereas the SEK/USD series constitutes of a major trend. It is interesting to study these two series as both have recognizable trends but differ in type. This could answer the question if a predetermined trend pattern are decisive for the proposed model's performance.

4.2 Analyzing the first differences in the series

To determine possible structures in the data, we take the first difference and examine the innovations. We are looking for a white noise process to explain our return residuals and determine if such is present. White noise is defined by:

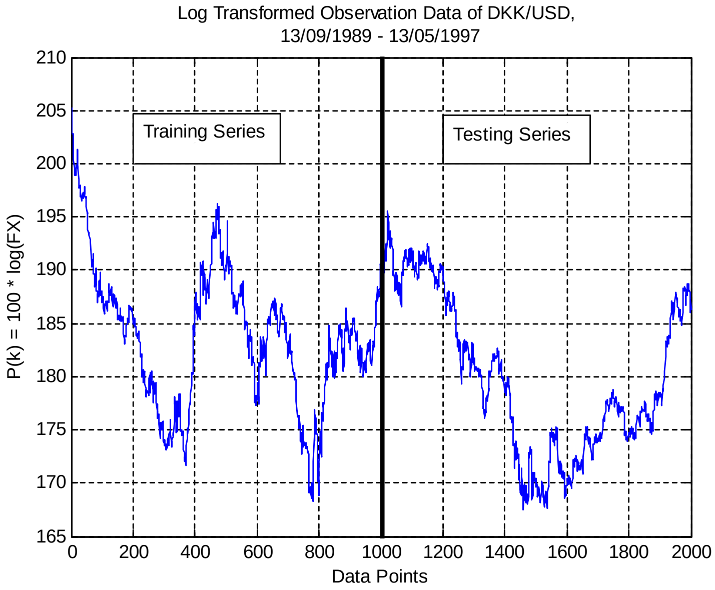

Figure 1 Levels in Observation Data DKK/USD

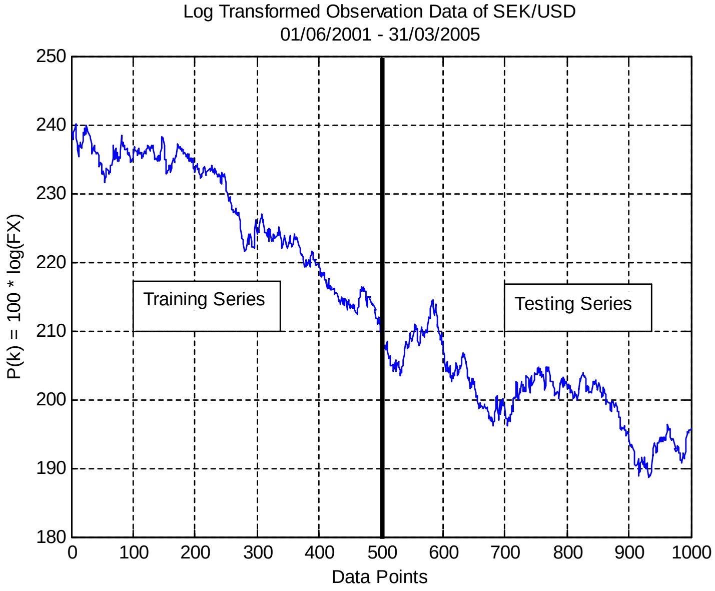

Figure 2 Levels in Observation Data SEK/USD

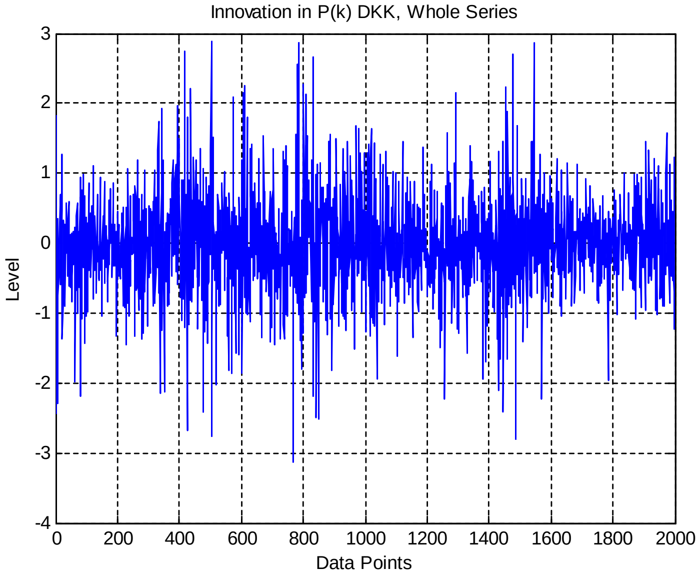

Figure 3 First difference in the $P_{k,DKK}$ data

Definition 3 *For a given time series $\{P_k\}$ determine a general function f such that $\{\varepsilon_t\}$ defined by $f(\{P_k\}) = \varepsilon_t$ is a strict white noise; a sequence of independent random variables.*

I.e. white noise is a sequence of uncorrelated, identical distributed random variables. In discrete time white noise is sometimes referred to as a completely uncorrelated process or a pure random process.(17)

The white noise process is important to identify as it gives validity to our model (that is Markov). By visual inspection of figures 3, and 4, we can see that the innovations resemble white noise very much, with a slight tendency of volatility clustering or volatility pooling within it. This is easier to see in the SEK/USD series as it less dense in observation than the DKK/USD series. The clustering effect is due to heteroscedastic innovations that possess "autocorrelation" in which large changes tend to follow each other and small changes vice versa. In other words, the current level of volatility tends to be positively correlated with its level during the immediately preceding periods. This means that previous period's disturbances can be a part of constructing a volatility forecast in the next period.

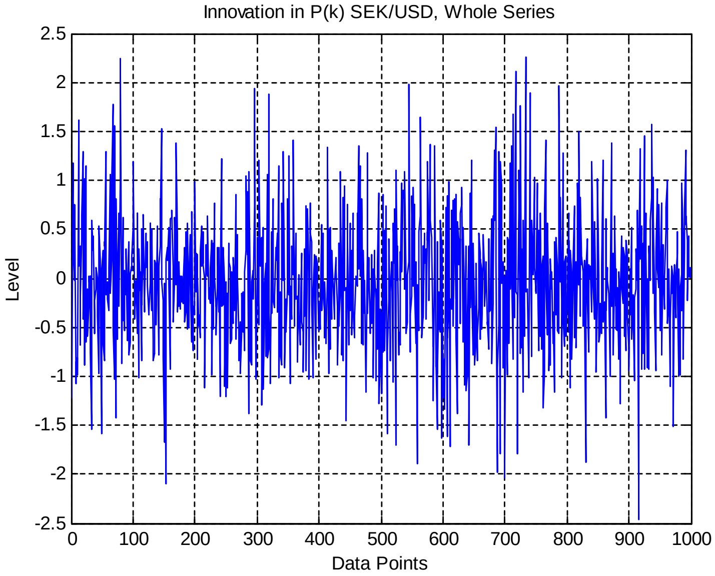

Figure 4 First difference in the $P_{k,SEK}$ data

The clustering effect is commonly known in financial time series data and its explanation lies much in the series' leptokurtic distribution, also known as a fat tail distribution. To determine the existence of such a distribution, we plot the two series in a density plot and try to fit a normal, as well as an alternative, distribution to them.

From figure 5, we can see that the differentiated $P_{k,DKK}$ series has a slightly skewed normal mean, which is -0.00888715 with kurtosis 5.1946 (of normal 3) and skewness of 0.0356. We might suspect the distribution is not normal. Defining N as number of observations, S as skewness, and K as kurtosis the Jarque-Bera (JB) test $\left(JB = \frac{N}{6}\left(S^2 + \frac{(K-3)^2}{4}\right)\right)$, and the Lilliefors test (Lillie) for goodness-of-fit to a normal distribution, reject the null hypothesis of a normal distribution. After fitting the data to a t-distribution instead, we conclude the differences in the DKK/USD data is better represented by a t-distribution with mean -0.0125172, variance 0.490076 and shape parameter of 4.25215 degree of freedom.

Looking at the differentiated $P_{k,SEK}$ series, figure 6 has a slightly skewed normal mean of -0.0431931 with kurtosis 3.5776 (of normal 3), and skewness of 0.1529. Variance is 0.452162. We could suspect this better represent a normal distribution than the $P_{k,DKK}$

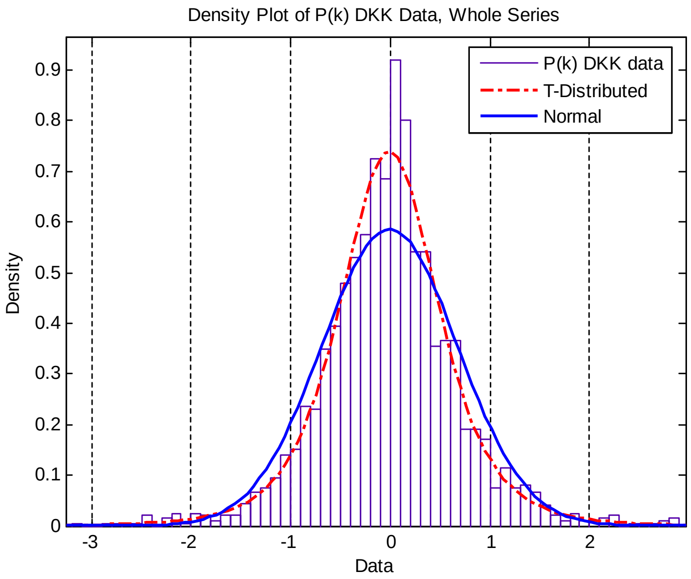

Figure 5 Density plot of the differenced $P_{k,DKK}$ series

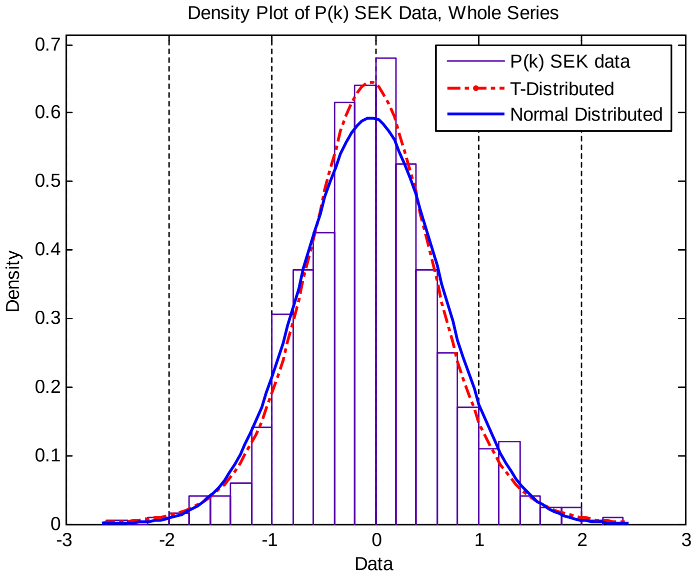

Figure 6 Density plot of the differenced $P_{k,SEK}$ series

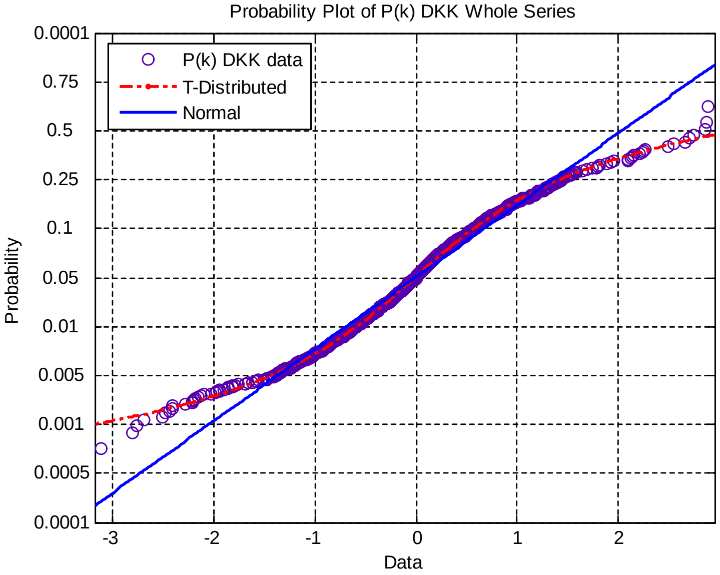

Figure 7 Normality Plot of the differenced $P_{k,DKK}$ series

series but the JB and Lillie test reject our null hypotheses again. After fitting the data to a t-distribution instead we conclude the differences in the SEK/USD data is better represented by a t-distribution with mean -0.0508788, variance 0.604005 and shape parameter of 10.1697 degree of freedom.

An alternative way of looking at the studied distributions is to use the Probability-Probability plots (also known as P-P plots or its normal variant, Norm-Plots). The P-P plot is used to see if the given data follows some specified distribution. Given a normal distribution, the ordered residuals should form a linear relation. Deviation from the linear cumulative distribution function can be explained by an another probability density function. Visually, the plots 7, and 8 also conclude what was analyzed above in the probability density function. We find the leptokurtic characteristics of excess kurtosis and (slight) fat tails is better explained by a t-distribution than a Gaussian.

As a stochastic process may exhibit degrees of serial correlation from an observation to the next, i.e. be able to use its volatility structure to predict future events, it is useful to determine the existence of such structure in the studied series, to conclude if it is white noise data or not. The Box-Pierce/Ljung-Box Q-statistics for residual serial correlation up

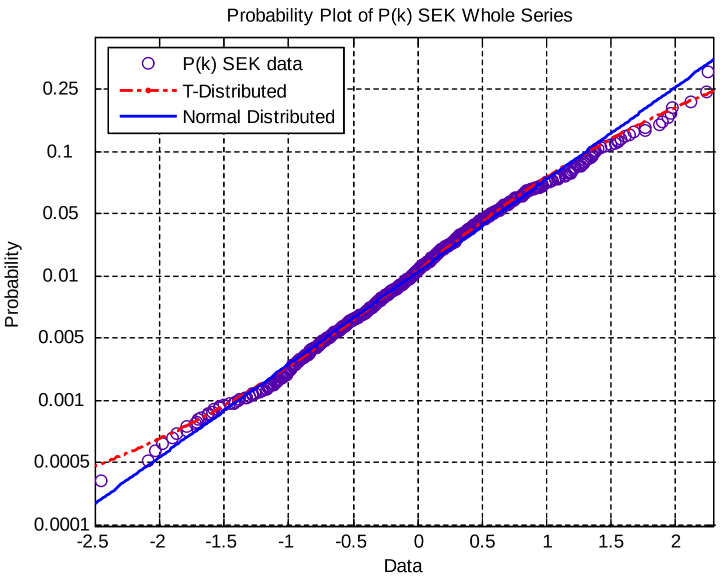

Figure 8 Normality Plot of the differenced $P_{k,SEK}$ series

to a specified order (see (16) for details and their p-values (14)) are often used as a tool to determine this.

If the τ_k is the k:th autocorrelation and N is the number of observations, the Q-statistics at lag k is a test statistic for the null hypothesis that there is no autocorrelation up to order k. It is computed as: $Q_{LB} = N(N+2)\sum_{k=1}^{K}\left(\frac{\tau_k^2}{N-k}\right)$. Since the autocorrelated values can also be autocorrelated, the Q test should be used a preliminary identification tool but provide an indication of the broad correlation structure within the series.(3)

In figure 9, we can see the sample autocorrelation function (ACF) with confidence bounds of 95% for the $P_{k,DKK}$ series. There are tendencies for serial correlation on lag 6 as that lag is lying outside the lower confidence bound, as indicated in the figure. Using the (E-Views) produced Q-statistics probability on 95% confidence level (see table below), it cannot reject the null hypothesis of no serial correlation up to lag 6 on the 5% confidence level. Therefore, it is concluded that the first difference of the $P_{k,DKK}$ series is white noise.

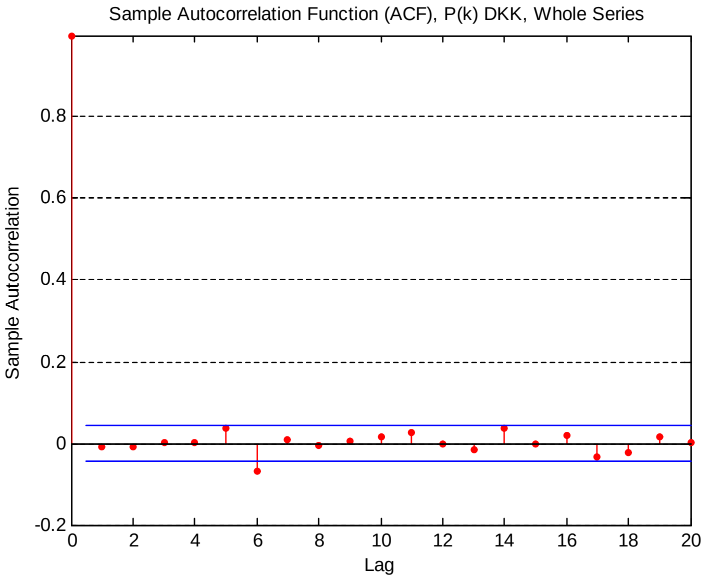

Figure 9 ACF for the differenced $P_{k,DKK}$ series, with confidence bounds.

DKK	Lag	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	Q-Stat	0.1979	0.3351	0.3438	0.3485	2.9759	12.331	12.436	12.511	12.572	13.185	14.587	14.589	15.037	17.82	17.824	18.524	20.843	22.041	22.469	22.469
	Prob	0.656	0.846	0.952	0.986	0.704	0.055	0.087	0.13	0.183	0.214	0.202	0.265	0.305	0.215	0.272	0.294	0.233	0.23	0.262	0.316

In figure 10, we can see the same function as above for the $P_{k,SEK}$ series and the function value is lying inside the upper and lower standard deviation confidence bounds, based on the assumption that all autocorrelations are zero beyond lag zero. Therefore, it is concluded that the first difference of the $P_{k,SEK}$ series also is white noise.

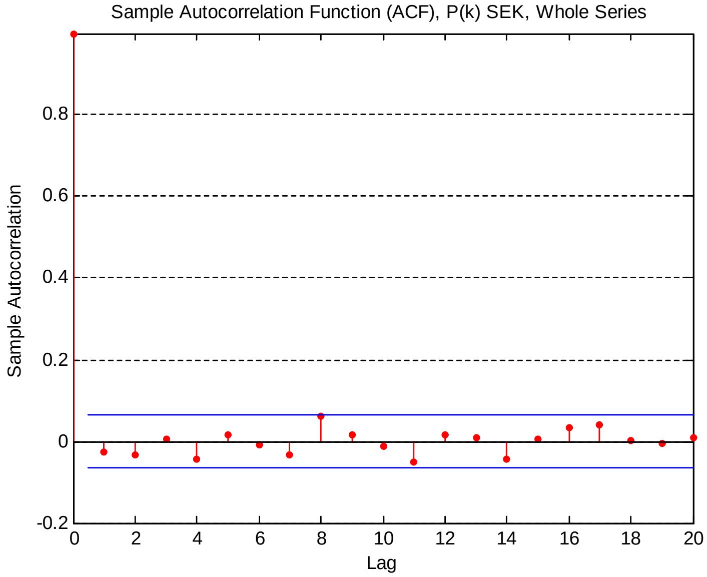

Figure 10 ACF for the differenced $P_{k,SEK}$ series, with confidence bounds.

SEK	Lag	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	Q-Stat	0.7368	1.7415	1.7813	3.7506	4.0114	4.0914	5.2403	9.1371	9.4002	9.5966	12.29	12.496	12.58	14.591	14.608	15.817	17.573	17.579	17.617	17.673
	Prob	0.391	0.419	0.619	0.441	0.548	0.664	0.631	0.331	0.401	0.477	0.342	0.407	0.481	0.407	0.48	0.466	0.416	0.484	0.548	0.609

In both series we can see a pattern of volatility clustering or heteroscedastisity which explains the excess kurtosis or fat tail effect displayed; which is typically observed in financial data. We cannot reject the null hypotheses of zero autocorrelation, hence we can conclude both series are good for our modelling purposes.

4.3 Analyzing the performance of the proposed model

We now divide the two series into their two sub-parts, training data and testing data. In order to deal with zero logarithms in our observation equation $\mathbf{Y}_k = \begin{pmatrix} P_k & \log(P_k - P_{k-1})^2 \end{pmatrix}^T$, we use an approximation suggested by Fuller (5) and also used by Peng et al. Assume σ_p^2

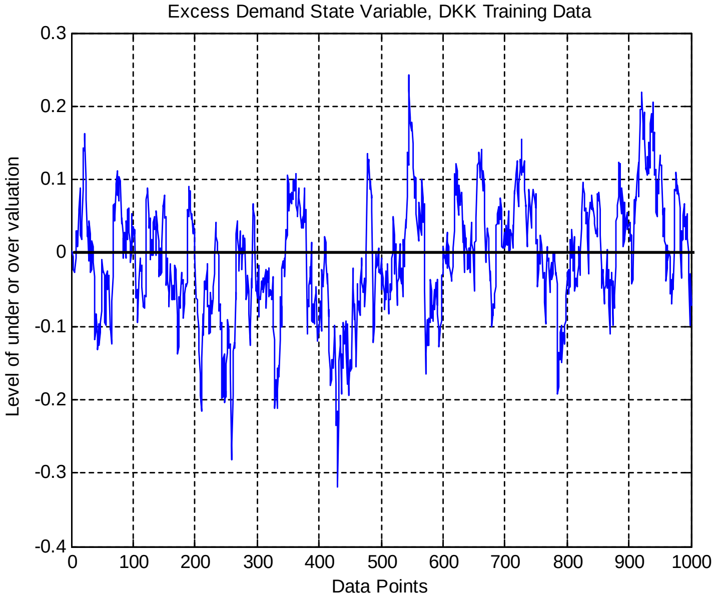

Figure 11 The estimated $\hat{\phi}_{k|k}$ for our $P_{k,DKK}$ training data.

is the sample variance of $P_k - P_{k-1}$ and η is a constant, (Peng et al. (22), (24) set this to 0.2) the following proposed model is used:

$$\log\left(P_k - P_{k-1}\right)^2 \approx \log\left[\left(P_k - P_{k-1}\right)^2 + \eta\sigma_p^2\right] - \frac{\eta\sigma_p^2}{\left(P_k - P_{k-1}\right)^2 + \eta\sigma_p^2}. \tag{37}$$

As mentioned, we are using the first half of the data to train our model. In order to optimize the model's parameters, Matlab's fminsearch function is used to minimize our function (36).

Inspecting our estimated excess demand $\hat{\phi}_{k|k}$ for the $P_{k,DKK}$ and the $P_{k,SEK}$ series (figures 11, and 12) we can see the mean revering process of $\hat{\phi}_{k|k}$ approximately around zero explaining when the asset is considered under or overvalued. Its dynamics are better explained by a smooth process than purely a random series.

We also plot the process of the estimated liquidity parameters $\hat{\lambda}_{k|k}$. Together with the estimated $\hat{\phi}_{k|k}$ the model express the variation of conditional variance of price by the change of market liquidity. Therefore, the liquidity parameter is equally important for the model to work.

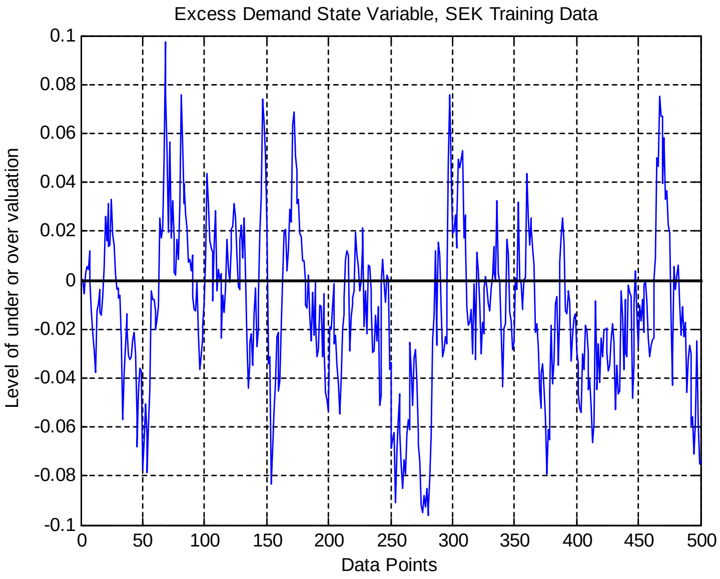

Figure 12 The estimated $\hat{\phi}_{k|k}$ for our $P_{k,SEK}$ training data.

The patterns differs between the series, reflecting the variation in the unobservable market liquidity. The high level of the first data points in the $\hat{\lambda}_{k,SEK}$ series is due to which start value we are using. In this case we let it stay in for illustrative purpose and it does not affect our study too much - we can see the smoothing effect of the Fuller approximation as the calculation base of the conditional variance increases.

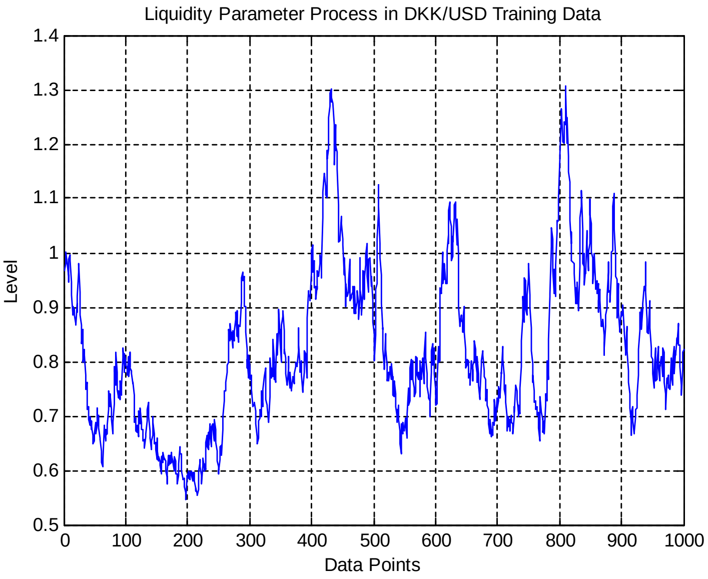

Figure 13 The estimated $\hat{\lambda}_{k|k}$ for our $P_{k,DKK}$ training data.

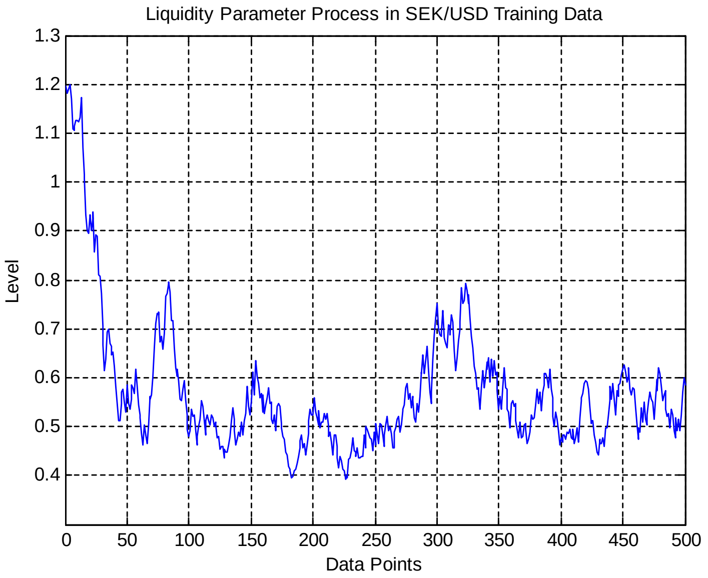

Figure 14 The estimated $\hat{\lambda}_{k|k}$ for our $P_{k,SEK}$ training data.

Applying fminsearch on formula (36), we obtain following estimated parameters and initial conditions for the training part:

$$\begin{array}{c} \text{DKK training Parameters} \\ \begin{bmatrix} \alpha_1 & -0.00016627 & \gamma_1 & 0.06365 \\ \alpha_2 & -0.0018725 & \gamma_2 & 0.081369 \\ \beta_1 & -0.044753 & \gamma_3 & 0.89422 \\ \beta_2 & -0.008597 & \delta & -1.0121 \\ \varepsilon_1 & 0.0010512 & \varepsilon_2 & 1.2653 \end{bmatrix} \end{array}, \tag{38}$$

$$\begin{aligned} V_0 &= \begin{pmatrix} 2.8457e-009 & 0 & 0 \\ 0 & 1.955e-009 & 0 \\ 0 & 0 & 6.0482e-010 \end{pmatrix}, \\ X_{0|0} &= \begin{pmatrix} 204.01 \\ -0.0002164 \\ -0.05687 \end{pmatrix}. \end{aligned} \tag{39}$$

Contrary to Peng et al. (22), this study have chosen not to estimate P_0. Instead we are using the first value in the P_k training series, for the testing part we are using the last P_k in the training series. This is because it lets us estimate fewer parameters. For the $P_{k,DKK}$ training series the start value is $P_0 = 204.01$, and for the $P_{k,SEK}$ series it is

233.04, as seen in the $X_{0|0}$ matrices (39), and (41).

$$\begin{aligned}
&\text{SEK Training Parameters}\\
&\begin{bmatrix}
\alpha_1 & 8.4481e-005 & \gamma_1 & 0.067794\\
\alpha_2 & 0.0064737 & \gamma_2 & 0.10046\\
\beta_1 & -0.10214 & \gamma_3 & 1.1392\\
\beta_2 & -0.00085975 & \delta & -0.42383\\
\varepsilon_1 & 0.00023767 & \varepsilon_2 & 1.2894
\end{bmatrix}, \qquad (40)
\end{aligned}$$

$$\begin{aligned}
V_0 &= \begin{pmatrix} 9.9553e-010 & 0 & 0\\ 0 & 1.354e-009 & 0\\ 0 & 0 & 1.2407e-009 \end{pmatrix},\\
X_{0|0} &= \begin{pmatrix} 233.04\\ -0.00032235\\ 0.33192 \end{pmatrix}. \qquad (41)
\end{aligned}$$

Although this reduction of parameters, it was found hard to find good initial values for the estimation process. As there are 15 parameters to be estimated, there are reasons to assume there exists numerous of local maximum (minimum) within which, the model converges. Also, that the model is sensitive to our starting values, as it within certain ranges did not converge at all. Consequently, the model is very sensitive and as the numerical study's performance is much due to the calibration of the initial guesses, we would expect its performance to be different when using other start values than presented above.

4.3.1 Dynamic Asset allocation strategy in-sample. To see whether the proposed model, using excess demand, is valid or not, we develop, like Peng et al. an asset allocation strategy for illustrative and testing purposes. We want to determine on which levels of the estimated $\hat{\phi}_{k|k}$ we want to buy or sell, i.e. by using thresholds. We determine a strategy which have the following properties;

$$\tau(\hat{\phi}_{k|k}) = if \begin{cases} \hat{\phi}_{k|k} > \tau_1, & \text{keep Asset 1 100\%} \\ \tau_2 < \hat{\phi}_{k|k} \leq \tau_1, & \text{keep Asset 1 80\%, Asset 2 20\%} \\ -\tau_3 < \hat{\phi}_{k|k} \leq \tau_{2,} & \text{keep Asset 1 50\%, Asset 2 50\%} \\ -\tau_4 < \hat{\phi}_{k|k} \leq \tau_{3,} & \text{keep Asset 1 20\%, Asset 2 80\%} \\ \hat{\phi}_{k|k} \leq -\tau_{4};_{3,} & \text{keep Asset 2 100\%} \end{cases} \tag{42}$$

Intuitively, this means that if the estimated $\hat{\phi}_{k|k}$ are within certain levels, it tells us to keep a certain proportion of our assets 1 and 2. The studied assets are DKK/USD, and SEK/USD. Our initial investment, A_0, is 100 USD.

In order to optimize the τ_i, $i = 1, \ldots, 4$ threshold parameters, we are using a asset valuation function proposed by Peng et al. (22), (24). In this, initial asset A_0, A_k assets at time k, and μ is a weighting factor a model can be built where the first part describing the requirements for final assets, and a second part representing fluctuations in asset during the test period,

$$J(\tau) = -A_N(\tau) + \frac{\mu}{N}\sum_{k=1}^{N}\left[A_k(\tau) - \left(A_0 + \frac{k}{N}\left(A_N(\tau) - A_0\right)\right)\right]^2. \tag{43}$$

Thus, the function represent that it is not only the largest final portfolio value that is considered but also the robustness of the allocation strategy. This is because a trader may make bad decisions based of the strategy.(24)

Function (43) is optimized with respect to following constraint:

$$\tau^* = \arg\min_{\tau} J(\tau), s.t.\tau_1 > 0, \tau_2 > 0, \tau_3 > 0, \tau_4 > 0, \tau_1 > \tau_2, \tau_3 < \tau_4. \tag{44}$$

However, this optimization problem can be rewritten as a standard minimization problem with constraints $\tau^* = \arg\min_{\tau} J(\tau)$ s.t. $\tau_{\min} \leq \tau \leq \tau_{\max}$, $A\tau \leq -b$ $A = \begin{pmatrix} -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{pmatrix}$, $b = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$, where the elements of $\tau_{\min}$ and b is very small positive numbers (e.g. 10^{-10}

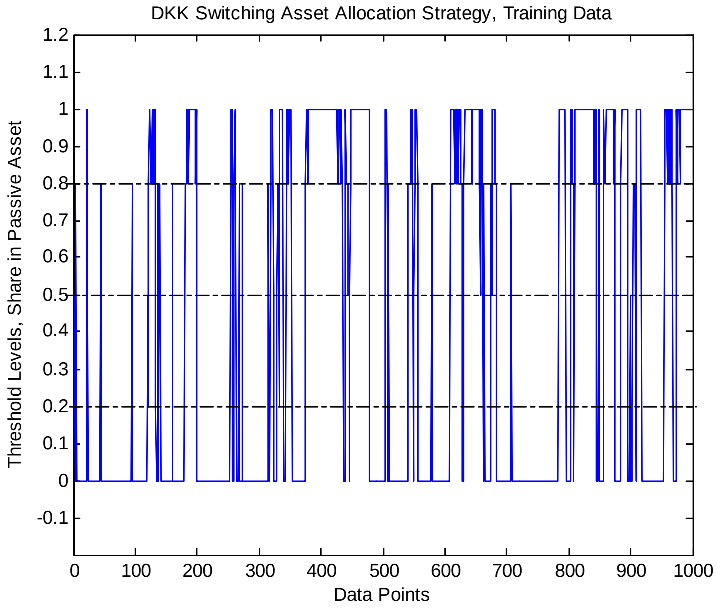

Figure 15 The DKK switching strategy for training data

) and $\tau_{\max}$ is given by the varying along the estimated process for $\hat{\phi}_{k|k}$. This latter minimization problem is solved by Matlabs's fmincon function, that is constraint minimization.

Alternatively, one could use trial and error to find the best threshold parameters. A more educated guess in this process is to produce a histogram of the $\hat{\phi}_{k|k}$ variable, to see how it is distributed around a mean. This study have chosen this method in conjuction with the fminsearch function where the initial guesses are in line with the threshold conditions in (44).

After optimizing function (43), we receive the following τ^* for $\hat{\phi}_{k|k,DKK}$ and $\hat{\phi}_{k|k,SEK}$. N denotes the sample size of the data, parameter μ is set to 1 and initial investment is 100.

	τ_1^*	τ_2^*	τ_3^*	τ_4^*	N	μ	A_0
$DKK\ \tau^*$	0.033726	0.004378	0.0007057	0.001356	1000	1	100
$SEK\ \tau^*$	0.00807	0.0001	$3e-005$	$3.1e-005$	500	1	100

(45)

This results in the following switching strategies for our training series, see figures 15, and 16.

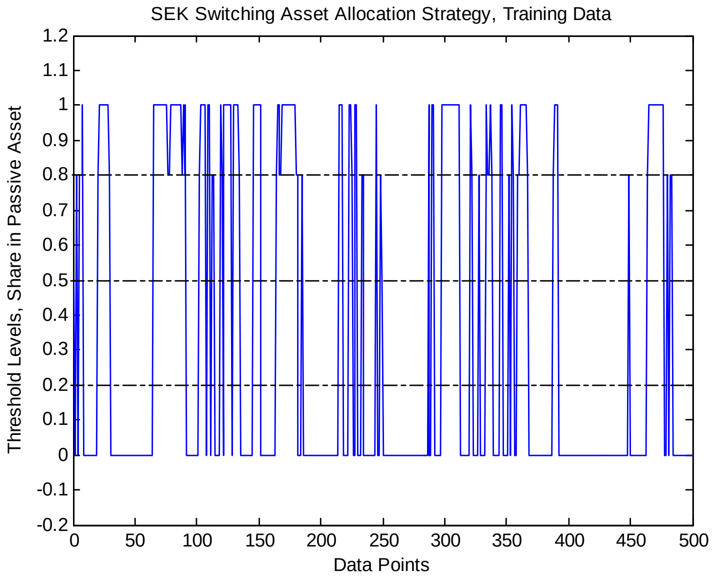

Figure 16 The SEK switching strategy for training data

The switching strategy tells which proportions that are allocated in the two assets at any given time. It is ranging from 0 to 100% investing in DKK or SEK, the rest in USD.Using the proposed strategy versus a passive strategy, we can now plot the result of the training series. The total value of the portfolio is calculated in respective currency, as well as the initial starting sum, 100 USD. The passive portfolio does not exhibit any allocation control and only reflects the current exchange rate at each time instance. It is against this passive portfolio we can measure the relative performance of our allocation strategy.

From figures 17, and 18 we can see that the Kalman filtered hidden excess demand variable in combination with the allocation strategy. It smooths the underlying price process into line segments of different lengths and tends to avoid sharp price drops most of the time. This is since the inner working of our Kalman filter, our Kalman gain K, gives a low credibility to large deviations, thus realizes more on the predicted measurement $C\hat{x}_k^-$. The deviations slowly trains the covariance matrix which produces new levels in the gain matrix. Consequently, the filtered series produces a smoother curve than the passive portfolio.

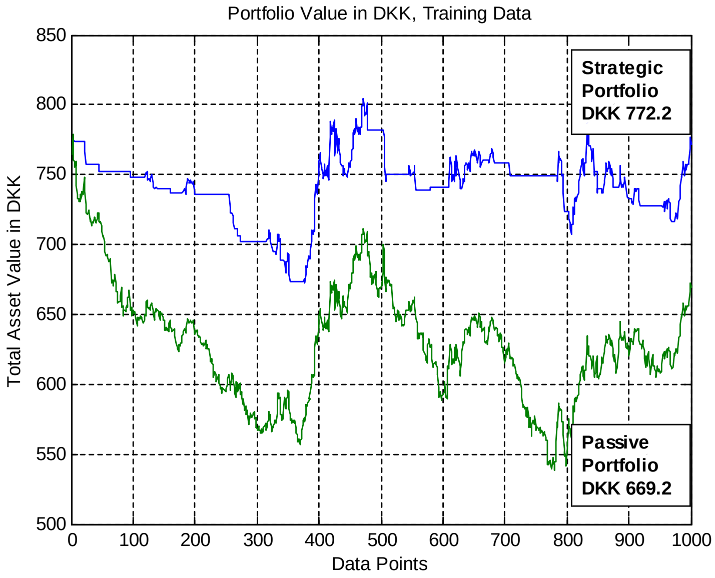

Figure 17 Portfolio value, allocation strategy versus passive strategy for $P_{k,DKK}$ training series

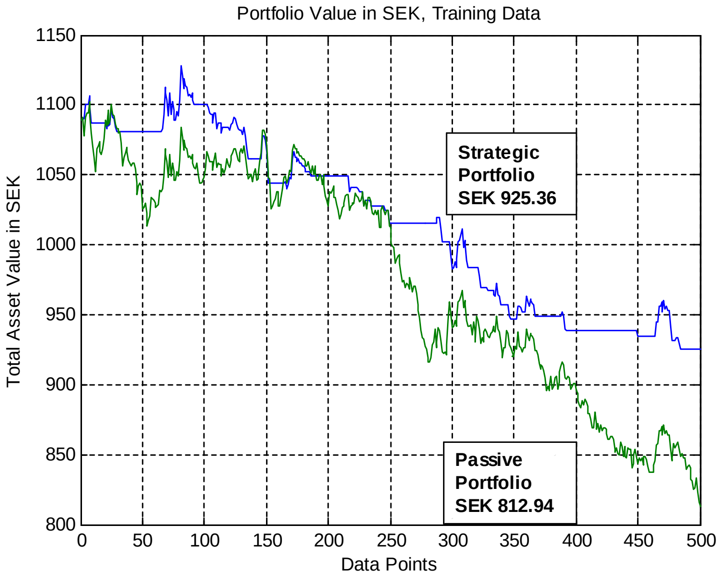

Figure 18 Portfolio value, allocation strategy versus passive strategy for $P_{k,SEK}$ training series

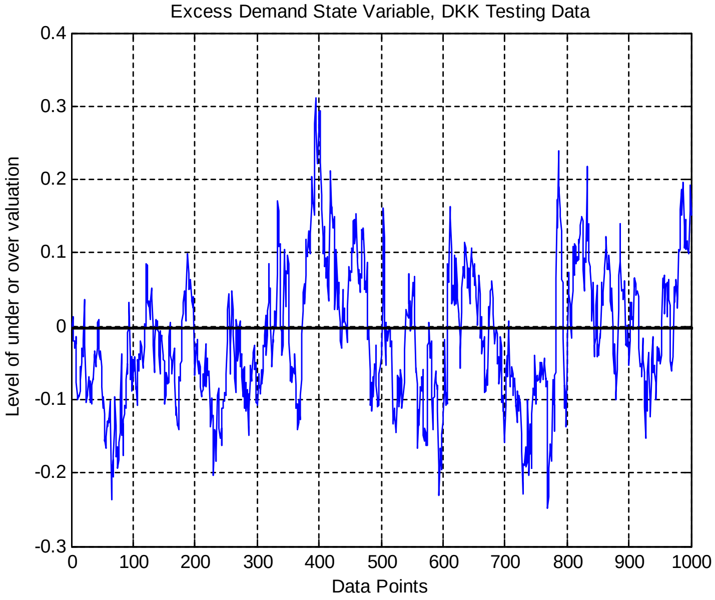

Figure 19 Estimated $\hat{\phi}_{k|k}$ for $P_{k,DKK}$ testing series

4.3.2 Dynamic Asset allocation strategy out-of-sample. The analysis above is dealing with in-the-sample data and it may be that the performance of the model is worse using out-of-sample data. To test this, we are using our estimated parameters and start values in (39), and (41) together with our estimated threshold parameters in (45). Given these, we recursively produce a new series of $\hat{\phi}_{k|k}$, and $\hat{\lambda}_{K|k}$ from the testing data to be evaluated in the asset allocation strategy (42).

As we can see in figures 19, 20, 21, 22, 23, 24, 25, and 26 our results are consistent with the one obtained for our training data. For the $P_{k,DKK}$ series, our strategy produces DKK 689.57 versus passive strategy of 644.25 Danish Krona. The discrete time asset allocation strategy also beat the passive strategy for the $P_{k,SEK}$ series, with 770.18 Swedish Krona versus SEK 707.46.

From the figures, we can see that the only miss-performing in the proposed model is between observations 100 and 200 for the $P_{k,DKK}$ testing data, were the model cannot beat the passive strategy. Assumedly, this result could be enhanced by fine-tuning the starting values before we estimate parameters for our training data (hence, get better estimates for the testing part).

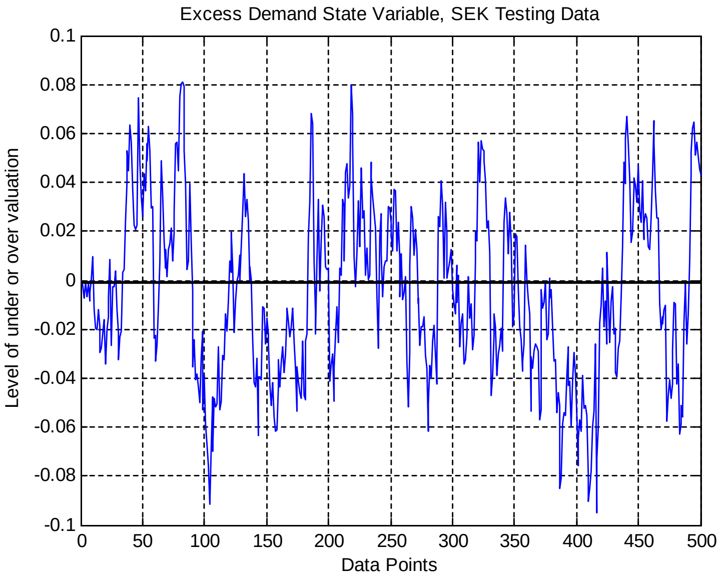

Figure 20 Estimated $\hat{\phi}_{k|k}$ for $P_{k,SEK}$ testing series

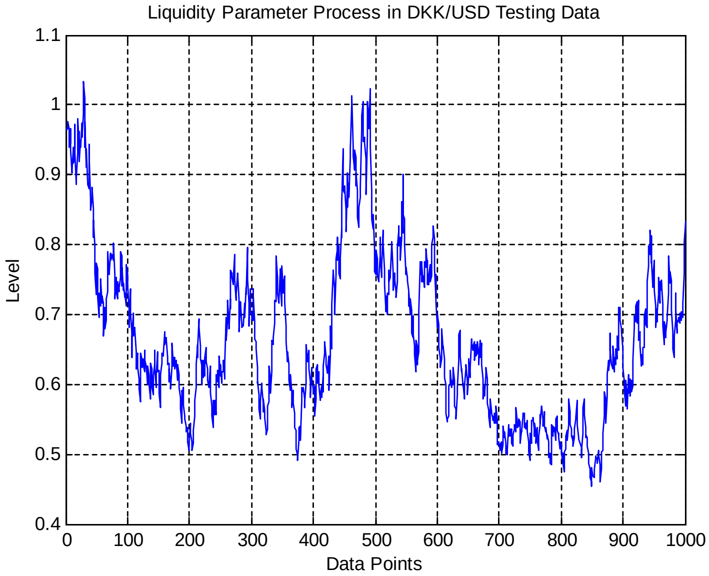

Figure 21 Estimated $\hat{\lambda}_{k|k}$ for $P_{k,DKK}$ testing series

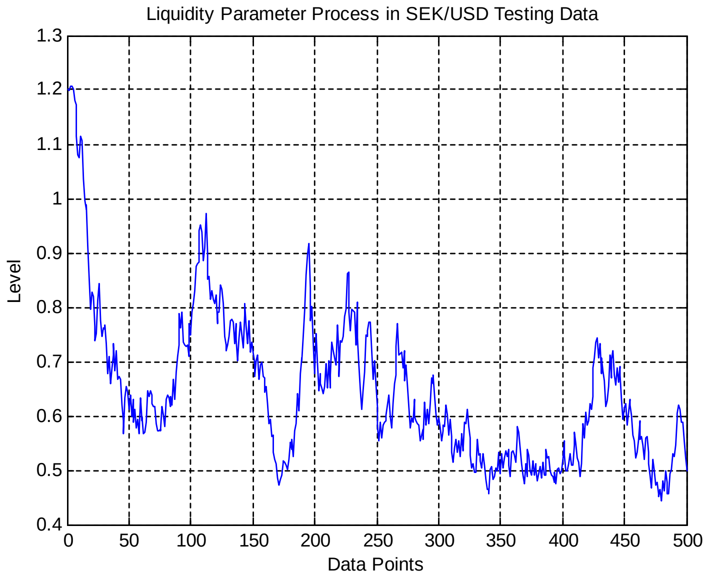

Figure 22 Estimated $\hat{\lambda}_{k|k}$ for $P_{k,SEK}$ testing series

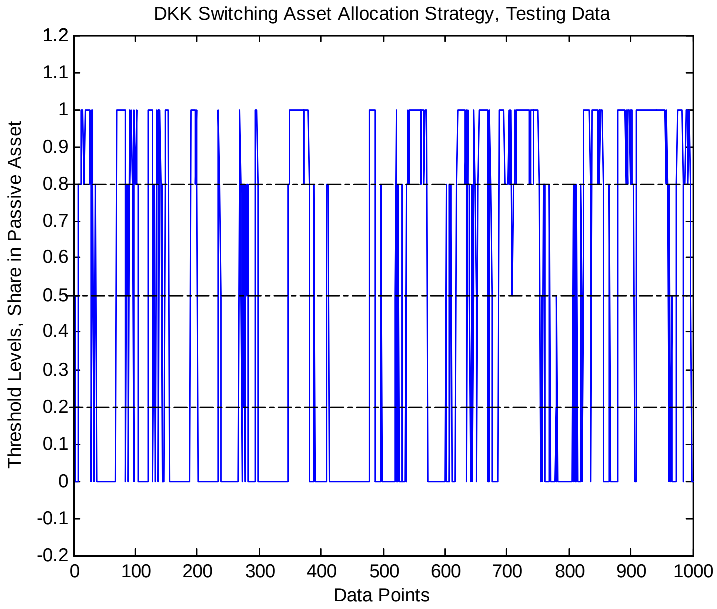

Figure 23 The DKK switching strategy for testing data

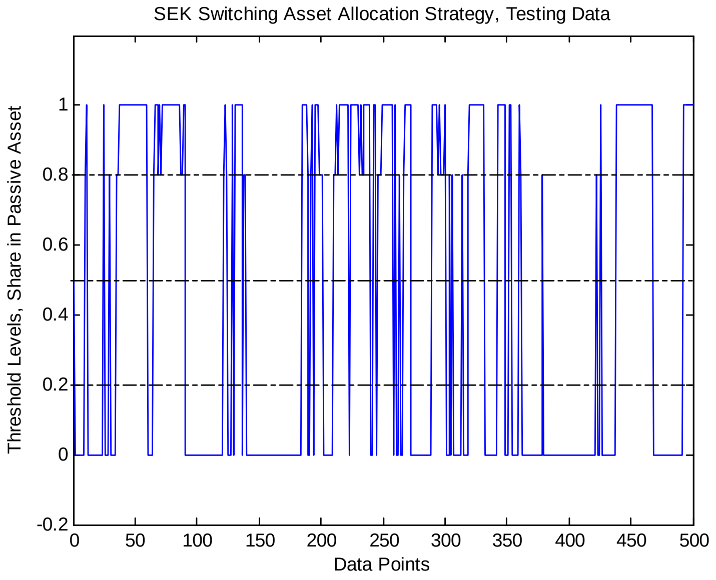

Figure 24 The SEK switching strategy for testing data

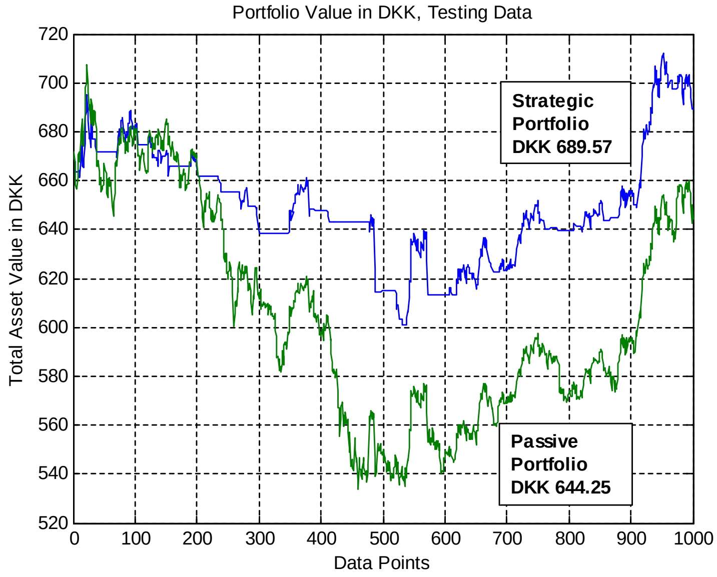

Figure 25 Portfolio values, allocation strategy versus passive strategy for $P_{k,DKK}$ testing series

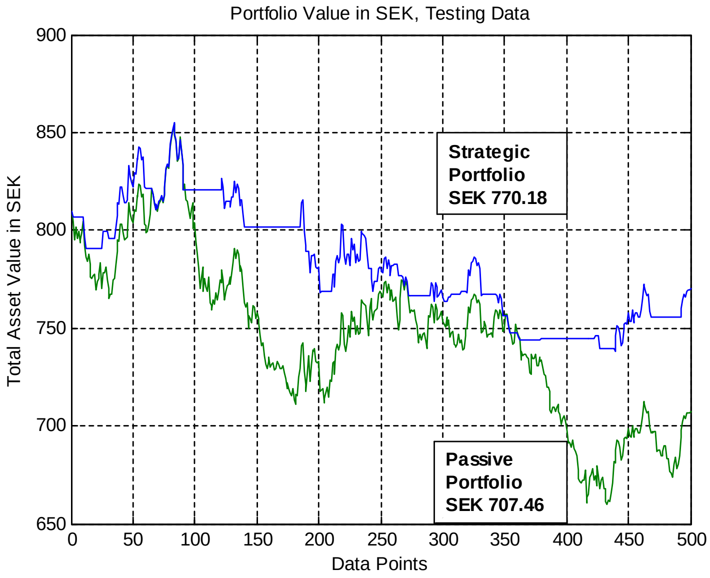

Figure 26 Portfolio values, allocation strategy versus passive strategy for $P_{k,SEK}$ testing series

V. Conclusions

In this paper we have studied a discrete time microstructure model, proposed by Ozaki and Peng (21), (22). This model suggests the use of excess demand ϕ and the inverse of market liquidity λ as two unobservable state variables as marker, determining whether the market is overvalued or undervalued. The model expresses the variation of conditional variance of price, were the amplitude of the price changes is dependent on the liquidity of the market that is allowed to vary over time and has effects on the conditional mean, as well as the conditional variance. We have shown that using this process of hidden excess demand and its liquidity is meaningful to base an asset allocation strategy which is efficient in terms of producing a residual money gain compared to a passive strategy.

To study these hidden state variables we have used state space analysis in combination with extended Kalman filtering and by this an interesting alternative analysis to traditional time series analysis have emerged, although it shows similarities to existing methods of model reduction and EGARCH. However, the analysis should foremost be put in contrast to a model framework were most financial stochastic processes traditionally focus on modelling the price dynamics in itself and its conditional variance.

The study has also shown that the viewpoint to consider the existence of a diffusion of information between informed markets participants and the less informed participants is fruitful. This diffusion causes trends in markets (in financial time series) which may exist at different degrees. By filtering its process, i.e. extracting the underlying trend information, we obtain a process with which we can use in a dynamic asset allocation model. By choosing sample data with two different kind of broad trend characteristics, one consisting of several smaller, and one consisting of mainly a directional trend, our model does not make any visible difference between them. This implies that the broad trend information does not need to have a certain profile in order to extract a useful process of hidden state variables.

Given the analysis we can, in accordance with Peng and Ozaki, see that the analyzed model have desirable properties both from a theoretical as well as a empirical viewpoint.

The drawbacks of the model, are that it is very sensitive to its initial values and much of its performance is due to which initial guesses one make before running the optimization. This is because there are 15 parameters to be estimated and by this, is assumed that there exist numerous of local minima and maxima, hence it is hard to determine where a global such exists, or if the model will reach convergence at all.

Also, our study did not take transaction costs into account. Such costs could have significant impact on the model's performance, all due to how frequent the switching pattern is and the magnitude of the costs (such as bid/offer spread). However, we mentioned that such tests were undertaken by Peng et. al. (24) suggesting this did not have any significant negative impact.

5.1 Proposal for further research

There are several ways in which the model could perform better. One such way is, instead of using the Extended Kalman Filter (EKF) is to use an Unscented Kalman Filter (UKF). This is as the EKF is only an approximation to the optimal nonlinear estimation. According to Wan and van der Merwe (7), UKF addresses many of the approximation issues in the EKF with a better performance, given the same level of computational complexity.

The major difference between EKF and UKF is how the state distribution is measured and propagated through the dynamics of a system. In the case of EKF, the distribution is propagated analytically by a first order linearization of the nonlinear system using derivatives. Such an approach estimates the first moment well but is poor in higher moments, as it can introduce large errors in the true posteriori mean and covariance of the transformed Gaussian Random Variable (GRV). This may lead to suboptimal performance and sometimes divergence of the filter. Instead, the UKF uses sample points, called sigma points, to approximate the GRV. By choosing a minimal set of them carefully it is possible to capture the true mean and covariance of the GRV. Therefore, UKF should capture posteriori mean and covariance accurately to a second order (Taylor series expansion) for any nonlinearity. This also leads to a more accurate recursive estimation.

A second improvement of the model is related to the estimation of parameters. There is a strong interdependence between our state estimation and model estimation where the state estimation needs a observable model to be estimated and model estimation needs a clean state to modeling the dynamics.

The EKF framework can be used for estimating a model's parameters (or weights) from clean data. This involves learning the nonlinear map $h(\cdot)$ which is now parameterized by the set of unknown parameters; the vector z_k. This approach is sometimes referred to as system identification, machine learning, or just modeling. The corresponding state space representation for this problem is formulated as:

$$\begin{aligned} z_{k+1} &= I \cdot z_k + w_k, \\ y_k &= h(x_k, z_k) + n_k, \end{aligned} \tag{46}$$

where z_k corresponds to a stationary process with identity state transition matrix, driven by process noise w_k. The output y_k corresponds to a nonlinear observation on z_k.

However, when the input clean state x_k is unobserved/hidden (or is not clean enough) and we can only observe a noisy signal y_k it is hard to do predictions, as we preferably need both. A solution to this problem is denoted as the dual estimation approach. Within an EKF environment, dual estimation means that two EKFs are run separately and concurrently. At every time step, one of the EKFs uses the other EKF's current map parameter estimate / weight estimate $\hat{w}_k$ for its own state estimate $\hat{x}_k$, while the other EKF uses its state estimate for a current weight estimate. This procedure can either be done iterative (off-line) where block of data is used, or sequential (on-line) were both EKFs uses new data point to update the signal estimate or the model estimate. Hence, the dual EKF estimation problem refers to an simultaneous estimation of both the states of the discrete-time nonlinear dynamic system and its parameters, given only noisy observations.

In a discrete time nonlinear dynamic state space environment the problem looks like;

$$\begin{aligned} x_{k+1} &= f\left(x_k, u_k\right) + v_k, \\ z_{k+1} &= z_k + v_k, \\ y_k &= h\left(x_k, w_k\right) + n_k. \end{aligned} \tag{47}$$

Finally, in order to enhance the measurement of the proposed model, we suggest its performance should be tested in a more traditional framework. This entails the calculations of different kind of information criteria, such as the use of mean squared errors (MSE), log-likelihood values, but also using mean pricing errors (MPE), root mean squared errors (RMSE) as used by Javaheri et. al. in (10). The model's value should be compared with existing models, such as EGARCH(P,Q) and GJR(P,Q) which are both (asymmetric) models developed to handle negative correlation between volatility and returns. They also take respect to the sign and the magnitude of the innovation noise term.

However, for the study in question, one should be aware of that there generally are no asymmetries in foreign exchange rates (9) and therefore models as EGARCH(P,Q) and GJR(P,Q) are probably not efficient in comparison?

Appendix A. Matlab programmes

This appendix present the Matlab programming of the model. It constitutes of two programmes and two functions in total. Below is example code for one of the studied series.

```
% ********** START FILTER PROGRAM *******************************
% Empty the memory
clear all
clc
% Define variables z and pk as global
global z
global pk
% Load the file pk.mat
load pk.mat
% Determine z as the first 1000 observations
z=pk(1:1000);
% Define parameters K and r. K equals length of z, i.e. in this case
% 1000 and r equals logarithmic reuturn of z(k) and Z(k-1).
K = length(z);
r = z(2:end)-z(1:end-1);
% We define needed matrices in three dimensions, row, column and lenght and
% fill these with empty values.
I =eye(3);
lambda = ones(K+1,1);
phi = zeros(K+1,1);
A = zeros(3,3,K+1);
C = zeros(2,3,K+1);
post_X = zeros(3,1,K+1);
Y = zeros(2,1,K+1);
S_pri = zeros(3,3,K+1);
KG = zeros(3,2,K+1);
```

```
% ********** START (GUESS) VALUES ***********************************************
% Define the start value on the parameters
a1 = 0.00008267;
a2 = 0.005910;
b1 = 0.9417-1;
b2 = 0.9917-1;
g1 = 0.05442;
g2 = 0.09417;
g3 = 1.001;
delta = -2.253;
e1 = 0.0003169;
e2 = 2.513;
v1 = 0.879*10^-9; v2=1.035*10^-9; v3=0.875*10^-9;
phi0=-0.000166;
x03=-1.4312;
% **********************************************************************
% Change these according: insample - first value of testdata, outsample - last value
% of test data
P0=204.01;
p=[P0; z]; %I.e. we start with P0 and fill the rest with the value in z.
% Define them as parameters
parameter =[a1 a2 b1 b2 g1 g2 g3 delta e1 e2 v1 v2 v3 phi0 x03]';
% ***********************************************************************
options = optimset('LargeScale','on','Display','iter',
'TolFun',10^(-1),'TolX',10^(-1),'MaxFunEvals',10000,'MaxIter',10000)
[x1,fval,exitflag,output]= fminsearch('logl_EKF',parameter, options);

% ****************FUNCTION logl_EKF**START*******************************
function lnl=logl_EKF(initial);
global z
global pk
K = length(z);
r = z(2:end)-z(1:end-1);
a1 = initial(1);
a2 = initial(2);
b1 = initial(3);
b2 = initial(4);
g1 = initial(5);
g2 = initial(6);
g3 = initial(7);
delta = initial(8);
e1 = initial(9);
e2 = initial(10);
v1 = initial(11); v2=initial(12); v3=initial(13);
phi0=initial(14);
x03=initial(15);
P0=233.04;
% We define needed matrices
I =eye(3);
lambda = ones(K+1,1);
phi = ones(K+1,1);
A = zeros(3,3,K+1);
C = zeros(2,3,K+1);
post_X = zeros(3,1,K+1);
Y = zeros(2,1,K+1);
S_pri = zeros(3,3,K+1);
KG = zeros(3,2,K+1);
Rk=[e1^2 0 ; 0 e2^2];
lambda0 = sqrt(exp(x03));
p=[P0; z];
```

```
% On the surface of the depth - the starting values - , the matrices should
% have the following properties.
% Functions within the matrices are created.
lambda0 = sqrt(exp(x03));
A(:,:,1) = [1 lambda0 0 ; a1/P0 1+b1 0 ; a2/P0 0 1+b2];
C(:,:,1) = [1 0 0 ; delta/P0 0 1];
Rk=[e1^2 0 ; 0 e2^2];
lambda(1,1)= lambda0;
post_X(:,:,1)= [P0 ; phi0 ; x03];
S_post(:,:,1) =[v1 0 0 ; 0 v2 0 ; 0 0 v3];
% For the depth >= 2 to end, the matrics has the following properties.
% A loop is created.
for k = 2:K+1;
% A priori state estimate of state X given information up to k-1
% is created by A matrix times posteriori X.
% This is the time update or prediction
pri_X(:,:,k)=A(:,:,k-1)*post_X(:,:,k-1);
% The observation function is a vector of observation and its variance
% we use the Fuller (1996) approximation with constant 0.2 and sample
% variance in order to avoid the problem with zero logarithms.
% Y(:,:,k)= [p(k) ; log((p(k)-p(k-1))^2+0.2*varr(k))-(0.2*varr(k))
/((p(k)-p(k-1))^2+0.2*varr(k))];
 Y(:,:,k)= [p(k) ; log((p(k)-p(k-1))^2+0.2*var(r))-(0.2*var(r))
/((p(k)-p(k-1))^2+0.2*var(r))];
% The measurement residual, the innovation Gamma, is measured by
% our observation subtracted by our apriori estimate for observation Y.
% This is also part of the prediction.
Gam(:,:,k)=Y(:,:,k)-C(:,:,k-1)*pri_X(:,:,k);
% Q is our variance-covariance matrix of Omega in our state equation,
% i.e. our process noise covariance.
```

```
Q(:,:,k) = [g3^2*(lambda(k-1,1))^2 0 0; 0 g1^2 0; 0 0 g2^2];
% Given above we now project the conditional state covariance
% ahead. This is done by using A matrix and posteriori state
% covariance + using the process noise covariance.
S_pri(:,:,k)=A(:,:,k-1)*S_post(:,:,k-1)*A(:,:,k-1)’+Q(:,:,k); %Sk
% In the weighting part of the Kalman filter equations we first have to
% determine the measurement residual posteriori covariance. This is
% determined by matrix C, apriori state covariance, and measurement
% covariance R.
V(:,:,k)=C(:,:,k-1)*S_pri(:,:,k)*C(:,:,k-1)’+ Rk; ; %Psi(k)
% The measurement weighting is determined by the Kalman Gain (KG) and
% is calculated using apriori state covariance, matrix C and the
% inverse of the measurement residual posteriori covariance.
KG(:,:,k)=S_pri(:,:,k)*C(:,:,k-1)’*inv(V(:,:,k));
% Updating the posteriori error covariance is determind by Kalman Gain,
% C and apriori error covariance.
S_post(:,:,k)=(I-KG(:,:,k)*C(:,:,k-1))*S_pri(:,:,k); % Vk
% Hence, a posteriori state estimate X is determined by apriori state
% estimate corrected by Kalman Gained innovation.
post_X(:,:,k)=pri_X(:,:,k)+KG(:,:,k)*Gam(:,:,k);
% We now extract our state estimate of the price from the
% posteriori state matrix, it is the first value in the vector at k.
phat(k,1)=post_X(1,:,k);
% And extract our state estimate of excess demand from the
% posteriori state matrix, it is the second value in the vector at k.
phi(k,1)=post_X(2,:,k);
% And finally, extract our state estimate of the price variance.
% It is the third value in the vector at k.
lambda(k,1)=sqrt(exp(post_X(3,:,k)));
% At last we can now fill matrices A and C with information.
```

```
C(:,:,k) = [1 0 0; delta/phat(k,1) 0 1];
A(:,:,k) = [1 lambda(k,1) 0 ; a1/phat(k,1) 1+b1 0 ; a2/phat(k,1) 0 1+b2];
for k = 2:K
lnl_k(k,1) = log(det(V(:,:,k))) + Gam(:,:,k)'* inv(V(:,:,k))*Gam(:,:,k);
end
lnl  =  sum(lnl_k) + 2*K*log(2*pi) ;
%****************FUNCTION logl_EKF**END**************************************
a1 = x1(1);
a2 = x1(2);
b1 = x1(3);
b2 = x1(4);
g1 = x1(5);
g2 = x1(6);
g3 = x1(7);
delta = x1(8);
e1 = x1(9);
e2 = x1(10);
v1 = x1(11); v2=x1(12); v3=x1(13);
phi0=x1(14); x03=x1(15);
% ***********************************************************************
A(:,:,1) = [1 lambda0 0 ; a1/P0 1+b1 0 ; a2/P0 0 1+b2];
C(:,:,1) = [1 0 0 ; delta/P0 0 1];
Rk=[e1^2 0 ; 0 e2^2];
lambda(1,1)= lambda0;
post_X(:,:,1)= [P0 ; phi0 ; x03];
S_post(:,:,1) =[v1 0 0 ; 0 v2 0 ; 0 0 v3];
for k = 2:K+1 ;
pri_X(:,:,k)=A(:,:,k-1)*post_X(:,:,k-1);
/((p(k)-p(k-1))^2+0.2*varr(k))];
Y(:,:,k)= [p(k) ; log((p(k)-p(k-1))^2+0.2*var(r))-(0.2*var(r))
```

```
/((p(k)-p(k-1))^2+0.2*var(r))];
Gam(:,:,k)=Y(:,:,k)-C(:,:,k-1)*pri_X(:,:,k);
Q(:,:,k) = [g3^2*(lambda(k-1,1))^2 0 0; 0 g1^2 0; 0 0 g2^2];
S_pri(:,:,k)=A(:,:,k-1)*S_post(:,:,k-1)*A(:,:,k-1)'+Q(:,:,k);
V(:,:,k)=C(:,:,k-1)*S_pri(:,:,k)*C(:,:,k-1)'+ Rk;   ;
KG(:,:,k)=S_pri(:,:,k)*C(:,:,k-1)'*inv(V(:,:,k));
S_post(:,:,k)=(I-KG(:,:,k)*C(:,:,k-1))*S_pri(:,:,k);
post_X(:,:,k)=pri_X(:,:,k)+KG(:,:,k)*Gam(:,:,k);
phat(k,1)=post_X(1,:,k);
phi(k,1)=post_X(2,:,k);
lambda(k,1)=sqrt(exp(post_X(3,:,k)));
C(:,:,k) = [1 0 0; delta/phat(k,1) 0 1];
A(:,:,k) = [1 lambda(k,1) 0 ; a1/phat(k,1) 1+b1 0 ; a2/phat(k,1) 0 1+b2];
end
save ('phi.mat','phi');
save ('x1.mat','x1');
save('Kalman_opti');
SUBPLOT(4,1,1), plot (phi, 'DisplayName', 'phi', 'YDataSource',
'phi'); figure(gcf)
SUBPLOT(4,1,2), plot (lambda, 'DisplayName', 'lambda',
'YDataSource', 'lambda'); figure(gcf)
strateveal
% ************END FILTER PROGRAM*******************************************

%****************PROGRAM strateveal**START*********************************
clear all
clc
load phi.mat;
load kurs.mat;
spot=kurs(1:500);
```

```
K = length(spot);
bet=100;
tau=ones(4,1);
tau0=[0.00807 0.0001 0.00003 0.000031]’;
% tau0=[0.005504 0.003382 0.003555 0.005937]’;
pos_1_1 = zeros(1,K)’; % Defines dimension of pos_1_1
pos_1_2 = zeros(1,K)’; % Defines dimension of pos_1_2
pos_2_1 = zeros(1,K)’; % Defines dimension of pos_2_1
pos_2_2 = zeros(1,K)’; % Defines dimension of pos_2_2
totpos_2 = zeros(1,K)’; % Defines dimension of totpos_2
netch_1 = zeros(1,K)’; % Defines dimension of netch_1
netch_2 = zeros(1,K)’; % Defines dimension of netch_2
gpos_1 = zeros(1,K)’; % Defines dimension of gpos_1
port_1 = zeros(1,K)’; % Defines dimension of port_1
port_2 = zeros(1,K)’; % Defines dimension of port_2
A=[-1 1 0 0 ;0 0 1 -1];
b=[-10^(-10);-10^(-10)];
% lb = [10^(-10) 10^(-10) 10^(-10) 10^(-10)]; % Lower bounds x > 0
% ub = [phi(k) phi(k) phi(k) phi(k)];         % Upper bounds
options = optimset(’LargeScale’,’off’,’Display’,’iter’,
’TolFun’,10^(-2),’TolCon’,10^(-2),’MaxFunEvals’,10000)
[x2,fval,exitflag,output]= fminsearch(’taumax’,tau0, options);
% [x,fval,exitflag,output] = fmincon(’taumax’,tau0,A,b,[],[],[],[],[],options);
% [x,fval,exitflag,output] = fmincon(’taumax’,tau0,A,b,[],[],lb,ub,[],options);

%****************FUNCTION TAUMAX**START**************************************
function taumin = taumax(tau);
load phi.mat;
load kurs.mat;
spot=kurs(1:500);
```

```
K = length(spot);
bet=100;
mu=1;
for k = 1:K
    if phi(k)>tau(1)
        prop_1(k)=1;
    elseif (tau(2) < phi(k))& (phi(k) <= tau(1))
        prop_1(k)=0.8;
    elseif (-tau(3) < phi(k))& (phi(k) <= tau(2))
        prop_1(k)=0.5;
    elseif (-tau(4) < phi(k))& (phi(k) <= -tau(3))
        prop_1(k)=0.2;
    elseif phi(k) <= -tau(4)
        prop_1(k)=0;
    else prop_1(k)=999999
    end;
end;
for k = 1:1
        pos_1_1(1)=bet;
        pos_2_1(1)=0;
        totpos_2(1)=(pos_1_1(1)*spot(1))+pos_2_1(1);
        gpos_1(1)=(prop_1(1)*totpos_2(1))/spot(1);
        netch_1(1)=gpos_1(1)-pos_1_1(1);
        netch_2(1)=-netch_1(1)*spot(1);
        pos_2_2(1)=pos_2_1(1)+netch_2(1);
        pos_1_2(1)=pos_1_1(1)+netch_1(1);
        port_2(1)=pos_2_2(1)+pos_1_2(1)*spot(1);
        port_1(1)=port_2(1)/spot(1);
end;
for k = 2:K
```

```
        pos_1_1(k)=pos_1_2(k-1);
        pos_2_1(k)=pos_2_2(k-1);
        totpos_2(k)=(pos_1_1(k)*spot(k))+pos_2_1(k);
        gpos_1(k)=(prop_1(k)*totpos_2(k))/spot(k);
        netch_1(k)=gpos_1(k)-pos_1_1(k);
        netch_2(k)=-netch_1(k)*spot(k);
        pos_2_2(k)=pos_2_1(k)+netch_2(k);
        pos_1_2(k)=pos_1_1(k)+netch_1(k);
        port_2(k)=pos_2_2(k)+pos_1_2(k)*spot(k);
        port_1(k)=port_2(k)/spot(k);
end;
for k = 1:K
   sumport_2 = sum(port_2(k)-(bet+k/K*(port_2(K)-bet)))^2;
end;
taumin(1,1) = -port_2(K)+mu/K*sumport_2;
%***************FUNCTION TAUMAX**END************************************
tau(1) = x2(1);
tau(2) = x2(2);
tau(3) = x2(3);
tau(4) = x2(4);
for k = 1:K
    if phi(k)>tau(1)
        prop_1(k)=1;
    elseif (tau(2) < phi(k))& (phi(k) <= tau(1))
        prop_1(k)=0.8;
    elseif (-tau(3) < phi(k))& (phi(k) <= tau(2))
        prop_1(k)=0.5;
    elseif (-tau(4) < phi(k))& (phi(k) <= -tau(3))
        prop_1(k)=0.2;
    elseif phi(k) <= -tau(4)
```

```
        prop_1(k)=0;
    else prop_1(k)=999999
    end;
end;
prop_2 = 1 - prop_1;
for k = 1:1
        pos_1_1(1)=bet;
        pos_2_1(1)=0;
        totpos_2(1)=(pos_1_1(1)*spot(1))+pos_2_1(1);
        gpos_1(1)=(prop_1(1)*totpos_2(1))/spot(1);
        netch_1(1)=gpos_1(1)-pos_1_1(1);
        netch_2(1)=-netch_1(1)*spot(1);
        pos_2_2(1)=pos_2_1(1)+netch_2(1);
        pos_1_2(1)=pos_1_1(1)+netch_1(1);
        port_2(1)=pos_2_2(1)+pos_1_2(1)*spot(1);
        port_1(1)=port_2(1)/spot(1);
end;
for k = 2:K
        pos_1_1(k)=pos_1_2(k-1);
        pos_2_1(k)=pos_2_2(k-1);
        totpos_2(k)=(pos_1_1(k)*spot(k))+pos_2_1(k);
        gpos_1(k)=(prop_1(k)*totpos_2(k))/spot(k);
        netch_1(k)=gpos_1(k)-pos_1_1(k);
        netch_2(k)=-netch_1(k)*spot(k);
        pos_2_2(k)=pos_2_1(k)+netch_2(k);
        pos_1_2(k)=pos_1_1(k)+netch_1(k);
        port_2(k)=pos_2_2(k)+pos_1_2(k)*spot(k);
        port_1(k)=port_2(k)/spot(k);
end;
passp = zeros(1,k)'; % Defines dimension of passp
```

```
   for k = 1:K
    passp(k) = 100*spot(k);
end;
save (’x2.mat’,’x2’);
save(’Tau_opti’);
SUBPLOT(4,1,3), plot (port_2, ’DisplayName’, ’port_2’,
’YDataSource’, ’port_2’); hold all; plot (passp, ’DisplayName’,
’passp’, ’YDataSource’, ’passp’); hold off; figure(gcf)
SUBPLOT(4,1,4), plot (prop_1, ’DisplayName’, ’prop_1’,
’YDataSource’, ’prop_1’); figure(gcf)
%****************PROGRAM strateveal**END**************************************
%*****************************************************************************
```

Bibliography

1. Bland, Martin. *An Introduction to Medical Statistics* (2 Edition). Oxford Medical Publications, Oxford : Oxford University Press, 1995.

2. Bouchaud, J.P. and R. Cont. "A Langevin approach to stock market fluctuations and crashes," *The European Physical Journal B*, *6*:543–550 (1998).

3. Box, G. P., et al. *Time Series Analysis: Forecasting and Control.* (3 Edition). Upper Saddle River, New Jersey: Prentice-Hall, 1994.

4. Frittelli, M., et al. *Stochastic methods in finance: lectures given at the C.I.M.E.-E.M.S. Summer School, held in Bressanone/Brixen, Italy, July 6-12, 2003*. Berlin; New York: Springer, 2004.

5. Fuller, W.A. *Introduction to Time Series*. New York: John Wiley, 1996.

6. Hamilton, James D. *Time series analysis*. Princeton, N.J.: Princeton Univ. Press, 1994.

7. Haykin, Simon S. *Kalman filtering and neural networks*. Adaptive and learning systems for signal processing, communications, and control, New York: Wiley, 2001.

8. Iino, M. and T. Ozaki. *A nonlinear model for financial dynamics*. Technical Report, The Institute of Statistical Mathematics, February 4 2000.

9. Inc, The MathWorks. *Garch Toolbox User's Guide, Version* (2 Edition). Natick, MA: The MathWorks, Inc., 2002.

10. Javaheri, Alireza, et al. "Filtering in Finance," *Wilmott Magazine*, (5) (May 2003).

11. Kalman, R. E. "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering, Transactions ASMA*, *Series D*(82):35–45 (1960).

12. Kloeden, Peter E. and Eckhard Platen. *Numerical solution of stochastic differential equations*. Applications of mathematics; 23, Berlin; New York: Springer-Verlag, 1992.

13. Kyle, Albert S. "Continuous Auctions and Insider Trading," *Econometrica*, *53*(6):1315–35 (1985).

14. Ljung, G. M. and G. E. P. Box. "On a Measure of a Lack of Fit in Time Series Models," *Biometrika*, *65*:297–303 (1978).

15. Lo, Andrew W. and A. Craig MacKinlay. *A non-random walk down Wall Street*. Princeton: Princeton University Press, 1999.

16. Lütkepohl, Helmut. *Introduction to Multiple Time Series Analysis*. Berlin ; New York : Springer-Verlag, 1993.

17. Madsen, H., et al. *Statistics in Finance*. Lyngby: IMM, DTU, 2002.

18. Nelson, Daniel B. "Conditional Heteroskedasticity in Asset Returns: A New Approach." *ARCH: Selected readings* edited by Robert F. Engle, 87–113, Advanced Texts in Econometrics. Oxford and New York: Oxford University Press, 1995.

19. O'Hara, Maureen. *Market microstructure theory*. Cambridge, Mass.; Oxford: Blackwell, 1995.

20. Ozaki, T., et al. "Use of Stochastic Differential Equation Models in Financial Time Series Analysis: Monitoring and Control of Currencies in Exchange Markets." *US-Japan Time Series Conferences*. 2001.

21. Peng, Hui, et al. "A Discrete Microstructure Model Based Modelling and Control Method for Financial Markets." *Conference on Control Applications*. Glasgow, Scotland, U.K.: 2002 IEEE International, 2002.

22. Peng, Hui, et al. "Modeling and asset allocation for financial markets based on discrete time microstructure model," *The European Physical Journal B*, *31*:285–293 (2003).

23. Peng, Hui, et al. "Modeling and Control for Foreign Exchange Based on a Countinous Time Stochastic Microstructure Model." *Conference on Decision and Control*, edited by 41st IEEE. Las Vegas, Nevada USA: IEEE, 2002.

24. Peng, Hui, et al. "Modelling and asset allocation for financial markets based on a stochastic volatility microstructure model," *International Journal of Systems Science*, *36*:315–327 (2005).

25. Verbeek, Marno. *A guide to modern econometrics* (2nd Edition). John Wiley and Sons, 2004.