APPLICATION OF THE SCHWARTZ-SMITH MODEL (2000) IN COPPER DERIVATIVES PRICING

by

Xiaoyu Fu
Bachelor of Science, Finance and Math, Penn State University

and

Zheng Peng
Bachelor of Business Administration, Accounting, University of Regina

PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE IN FINANCE

In the Master of Science in Finance Program
of the
Faculty
of
Business Administration

SIMON FRASER UNIVERSITY

Fall 2018

Approval

Name: **Xiaoyu Fu, Zheng Peng**

Degree: **Master of Science in Finance**

Title of Project: **Application of Schwartz-Smith Model (2000) in Copper Derivatives Pricing**

Supervisory Committee:

Dr. Eduardo Schwartz
Senior Supervisor
Professor, Faculty of Business Administration

Dr. Victor Song
Second Reader
Lecturer, Faculty of Business Administration

Date Approved:

Abstract

In this paper, we explore the use of Schwartz and Smith two-factor model in copper pricing. We used both Copper future data from LME and Analyst Forecast data from Bloomberg (LME) and World Bank as input to generate futures curve and spot curve. The Schwartz- Smith model incorporates the long-term equilibrium prices that commodity price will approach in the long-term and short-term mean reversion characteristic of commodity prices. To estimate the state variables and model parameters, Kalman filter technique was used to update the state variables through iteration and Maximum likelihood approximation to compute the term structure, since Kalman filter is able to estimate model's parameters when the model relies on non-observable data. This model is able to explain the copper's term structure in an intuitive way. We begin by describing the input data in section 2 and explaining the short-term and long-term model in section 3. In section 4, we discuss the estimation process using the Kalman filter and, in section 5 we describe the empirical result by applying the model to Copper futures and forecast data. In section6, we offer the concluding remarks.

Keywords: Copper futures; Analysts' forecasts; Two-factor model; Kalman filter

Acknowledgements

We would like to express our great gratitude to our supervisor Eduardo Schwartz for helping us understanding the model and providing valuable comments and inspiring encouragement all the way in writing this project. We also would like to thank Phil Goddard for helping us with Matlab coding of the model. We would also like to thank our second grader Victor Song for supporting us and providing great comments and feedback.

Table of Contents

List of Figures

List of Tables

1: Introduction

In this paper, we use the Schwartz and Smith (2000) Model to fit the Copper prices and examine how well the model fits the data.

For the reasons why we choose to analyze copper, first, its spot pricing is a good indicator of worldwide capital construction and its futures prices are a good indicator of expectations of future capital construction. Second, it is dense in terms of value per weight and is easy to transport worldwide with little concern for environmental damage, as happens with crude oil. Thus, it is not subject to regional supply bottlenecks, as we see with intercontinental natural gas or the current Brent-WTI crude oil spread. Third, its consumption is not subject to seasonal fluctuations, as occurs for natural gas or electricity. Last but not least, it has a liquid forward market with deliveries well into the future, unlike iron ore or steel.

Before the Schwartz and Smith two factors model, studies on commodity stochastic model assumed the commodity prices followed a "random walk" described by geometric Brownian motion. However, this model allows mean-reversion in short-term prices and uncertainty in the equilibrium level to which prices revert to be incorporate into the model, making it more intuitive and easy to understand. Moreover, this model facilitates risk analysis, because it provides volatility estimates of the mean-reverting and long-run mean factor. And the model is useful for real options models that estimate the value of investment opportunities and provide criteria for starting, delaying, expanding and abandoning projects.

We begin by describing the input data in section 2 and explaining the short-term and long-term model in section 3. In section 4, we discuss the estimation process using the Kalman filter and, in section 5 we describe the empirical result by applying the model to Copper futures and forecast data. In section6, we offer the concluding remarks.

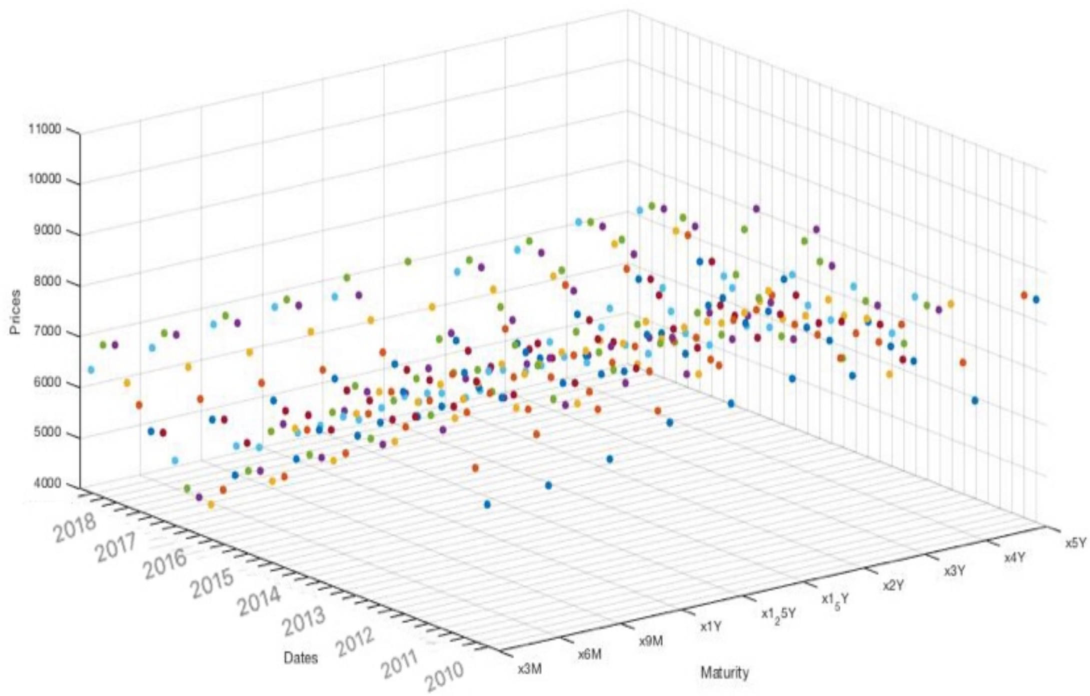
2: Data

2.1 Analyst's Forecast Data

Analysts' price forecast data are obtained from Bloomberg and World Bank. The Bloomberg analyst forecast provides forecasts of Copper prices up to five years with 9 maturities. We were able to get quarterly forecast data and in each observation quarters there are many forecasts provided by analysts from different banks, and we took the median of the available forecasts. The analyst forecasts can be viewed as a proxy of real commodity prices so in this paper it is used to construct the spot curve. However, the analyst forecast data is more noisy, since the source of data is not very consistent, for example, the number of forecasts available in each period is different, and analysts sometimes have huge disagreement on future prices. But it is the best data we can obtain to analyze the market's view on Copper.

The forecasts by World Bank and IMF (not included in this paper) have longer-term forecast up to over 15 years. However, they don't have data in similar maturity in the future and don't have data in every quarter. Since our code requires data to have all maturities in each observation and need to have observations every quarter, we have a to make some assumptions in the input data and fill the unavailable data. In the data table 4, cells in red are either calculated using the average of t-1 and t+1 data or filled in by using forecast by World Bank. We then obtain a matrix of the mix of forecast from different sources, the y-axis is date (quarterly data starting from Jan. 2010 to Oct. 2018) the x-axis is maturity (3 months, 6 months, 9 months, 1 year, 1.25 years, 1.5 years, 2 years, 3 years and 4 years) and all our numbers is in unit of USD/tonne, Shown in Figure 1.

Figure 1. Analysts' Forecasts Observations

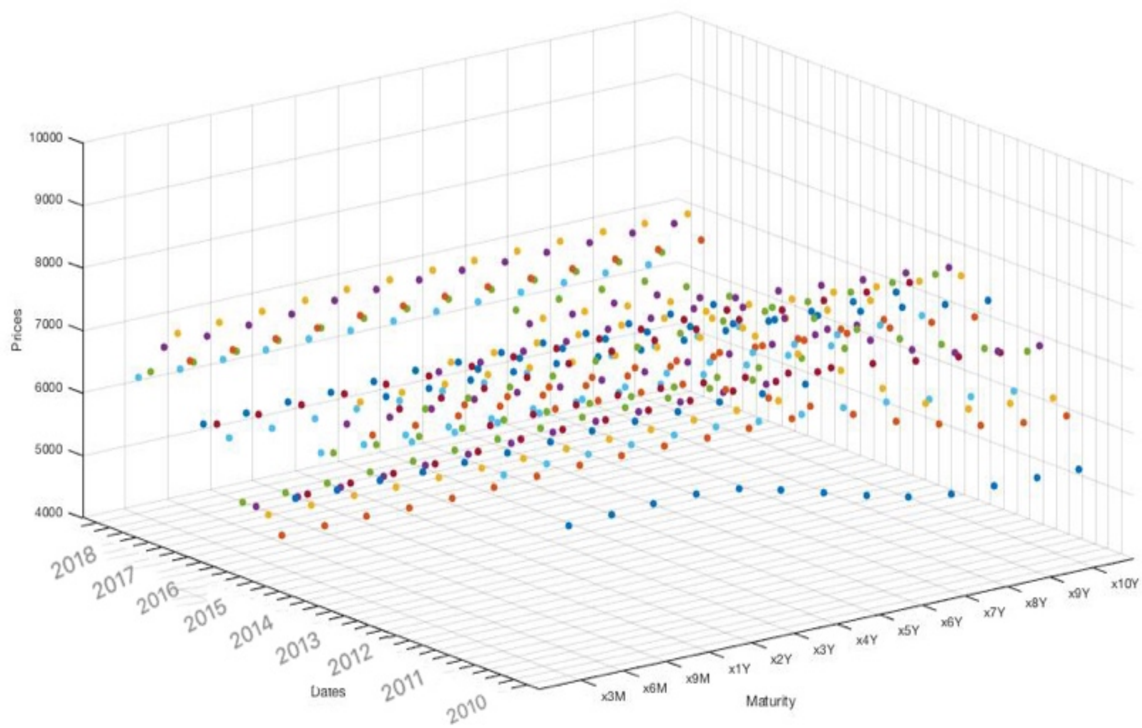


2.2 Copper Futures Data

Copper Futures Data is obtained from the London Metal Exchange. We took quarterly futures with maturities of 3 months, 6 months, 9 months and every year up to 10 years.

Futures data in more frequent than Forecast data and with data of different future maturities available in every observation. So, we did not make any modification for the futures contracts. Our data for Copper Futures is composed of 34 quarterly data from Jul. 2010 to Oct. 2018, with a unit of USD/tonne, shown in Figure 2.

Figure 2. LME Futures Observations



3: The Schwartz and Smith Two-Factor Model

This section provides a description of the short-term/long-term model by Schwartz and Smith (2000). We introduce the structure and properties of this model and the distribution for future spot prices in Section 3.1. And then in Section 3.2, we describe the risk-neutral version of this model, which was used to derive closed-form expressions for prices of futures and other commodity-related derivatives

3.1 The Short-Term/Long-Term Model

In the previous stochastic models for commodity prices, prices are expected to grow at some constant rate with the variance in future spot prices increasing in proportion to time. For most commodities, however, it seems that there is a mean reversion in prices and uncertainty about the equilibrium price to which prices revert. Considering these two effects, Schwartz and Smith developed a simple two-factor model of commodity prices that allows mean-reversion in short-term prices and uncertainty in the equilibrium level to which prices revert. Although neither of these two factors is directly observable, they can be estimated from spot and futures prices. The differences between the prices for the short-term and long-term contracts provide information about short-term variations in prices. And movements in prices for long-term futures contracts provide information about the equilibrium price level.

Specifically, the spot price of a commodity at time t (St) is constituted by two stochastic factors, which are the short-term deviation from this equilibrium price (χ_t) and the long-term equilibrium price (ξ_t). And the sum of these two factors is the logarithm of the spot price.

$$\ln(\mathrm{St}) = \chi_t + \xi_t \tag{2.1.1}$$

The short-term deviations are expected to revert to zero following an Ornstein-Uhlenbeck process, reflecting short-term changes in prices resulting from unusual weather or a supply disruption. The mean-reversion coefficient (κ) represents the rate at which the short-term deviations revert towards zero.

$$d\chi_t = -\kappa\chi_t dt + \sigma_\chi dz_\chi \quad (2.1.2)$$

And the long-term equilibrium price is assumed to follow geometric Brownian motion with drift (μ_ξ), reflecting expectations of the exhaustion of existing supply, improvement of the technology for the production, inflation, and political effects.

$$d\xi_t = \mu_\xi dt + \sigma_\xi dz_\xi \quad (2.1.3)$$

The dz_χ and dz_ξ are correlated increments of standard Brownian motion processes with $dz_\chi dz_\xi = \rho_{\chi\xi} dt$. And Schwartz and Smith (2000) also pointed out that this model is equivalent to the stochastic convenience yield model of Gibson and Schwartz (1990), but with the difference that changes in short-term futures prices are interpreted as short-term price variations rather than changes in the instantaneous convenience yield.

On the one hand, the process for short-term deviation allows for changes in the spot price which are not expected to persist in the long run and specifies the way in which these short-run deviations from the equilibrium price are expected to disappear. On the other hand, the process for equilibrium price level separates the short-term/long-term model from the class of pure mean-reversion models. And it also allows for the possibility that long-run changes in the spot price. Therefore, this model allows for mean-reversion in short-term prices and uncertainty in, and evolution of, the equilibrium price, a model structure that is in line with the inherent uncertainty of equilibrium prices and the apparent mean-reversion in prices for most commodities at the time (the year 2000).

Based on the structure of this model, Schwartz and Smith derived the distributions for future spot prices. Given the initial values of the two factors (χ_0 and ξ_0) and based the Equations 2.1.2 and 2.1.3, χ_t and ξ_t were found to be jointly normally distributed with mean vector and covariance matrix:

$$E[(\chi_t, \xi_t)] = [e^{-\kappa t}\chi_0, \xi_0 + \mu_\xi t] \tag{2.1.4}$$

and

$$Cov[(\chi_t, \xi_t)] = \begin{bmatrix} (1 - e^{-2\kappa t})\frac{\sigma_\chi^2}{2\kappa} & (1 - e^{-\kappa t})\frac{\rho_{\chi\xi}\sigma_\chi\sigma_\xi}{\kappa} \\ (1 - e^{-\kappa t})\frac{\rho_{\chi\xi}\sigma_\chi\sigma_\xi}{\kappa} & \sigma_\xi^2 t \end{bmatrix} \tag{2.1.5}$$

And from on the Equations 2.1.4 and 2.1.5, the log of the future spot prices is then normally distributed, and the spot price is then log-normally distributed, with which are:

$$\mathrm{E}[\ln(S_t)] = e^{-\kappa t}\chi_0 + \xi_0 + \mu_\xi t \tag{2.1.6}$$

$$\mathrm{Var}[\ln(S_t)] = (1 - e^{-2\kappa t})\frac{\sigma_\chi^2}{2\kappa} + \sigma_\xi^2 t + 2(1 - e^{-\kappa t})\frac{\rho_{\chi\xi}\sigma_\chi\sigma_\xi}{\kappa} \tag{2.1.7}$$

$$\mathrm{E}[S_t] = \exp\left(\mathrm{E}[\ln(S_t)] + \tfrac{1}{2}\mathrm{Var}[\ln(S_t)]\right) \tag{2.1.8}$$

or

$$\begin{aligned} \ln(\mathrm{E}[S_t]) = {} & \mathrm{E}[\ln(S_t)] + \frac{1}{2}\mathrm{Var}[\ln(S_t)] \\ & = e^{-\kappa t}\chi_0 + \xi_0 + \mu_\xi t \\ & + \tfrac{1}{2}((1 - e^{-2\kappa t})\tfrac{\sigma_\chi^2}{2\kappa} + \sigma_\xi^2 t + 2(1 - e^{-\kappa t})\tfrac{\rho_{\chi\xi}\sigma_\chi\sigma_\xi}{\kappa}) \end{aligned} \tag{2.1.9}$$

Table 1. Model Parameter Description

Short-Term/Long-Term Model Parameter		
Symbol	Description	Definition in Terms of Stochastic Convenience Yield Model
κ	Short-term mean-reversion rate	κ
σ_χ	Short-term volatility	σ_2/κ
dz_χ	Short-term process increments	dz_2
μ_ξ	Equilibrium drift rate	$(\mu - \alpha - \frac{1}{2}\sigma_1^2)$
σ_ξ	Equilibrium volatility	$(\sigma_1^2 + \sigma_2^2/\kappa^2 - 2\rho\sigma_1\sigma_2/\kappa)^{1/2}$
dz_ξ	Equilibrium process increments	$(\sigma_1 dz_1 - (\sigma_2/\kappa)dz_2)(\sigma_1^2 + \sigma_2^2/\kappa^2 - 2\rho\sigma_1\sigma_2/\kappa)^{-1/2}$
$\rho_{\xi\chi}$	Correlation in increments	$(\rho\sigma_1 - \sigma_2/\kappa)(\sigma_1^2 + \sigma_2^2/\kappa^2 - 2\rho\sigma_1\sigma_2/\kappa)^{-1/2}$
λ_χ	Short-term risk premium	λ/κ
λ_ξ	Equilibrium risk premium	$\mu - r - \lambda/\kappa$

3.2 Risk-Neutral Processes and Valuation

To value future contracts and European options on these futures by using the two-factor model, Schwartz and Smith developed a risk-neutral version shown by Equations 2.2.1 and 2.2.2 below.

$$\mathrm{d}\chi_t = (-\kappa\chi_t - \lambda_\chi)dt + \sigma_\chi dz_\chi^* \quad (2.2.1)$$

and

$$\mathrm{d}\xi_t = (\mu_\xi - \lambda_\xi)dt + \sigma_\xi dz_\xi^* \quad (2.2.2)$$

where the dz_χ^* and dz_ξ^* are correlated increments of standard Brownian motion processes with $dz_\chi^* dz_\xi^* = \rho_{\chi\xi} dt$.

Noticeably, there are three major differences between the short-term/long-term model and the risk-neutral version. First, two risk premium parameters (λ_χ *and* λ_ξ) are introduced to the risk-neutral paradigm, and they take the form of adjustments to the drift of the stochastic processes. Second, the short-term deviations are assumed to follow an Ornstein-Uhlenbeck process reverting to $\frac{-\lambda_\chi}{\kappa}$, rather than zero. Third, the long-term equilibrium price is assumed to follow geometric Brownian motion with drift $\mu_\xi^* = (\mu_\xi - \lambda_\xi)$, instead of μ_ξ.

Therefore, Schwartz and Smith found that, under these risk-adjusted processes, χ_t *and* ξ_t were found to be jointly normally distributed with mean vector and covariance matrix:

$$E^*[(\chi_t, \xi_t)] = [e^{-\kappa t}\chi_0 - (1 - e^{-\kappa t})\frac{-\lambda_\chi}{\kappa}, \xi_0 + \mu_\xi^* t] \quad (2.2.3)$$

and

$$Cov^*[(\chi_t, \xi_t)] = Cov[(\chi_t, \xi_t)] \quad (2.2.4)$$

Then, the log of the future spot price under risk-adjusted valuation paradigm is normally distributed with:

$$E^*[\ln(S_t)] = e^{-\kappa t}\chi_0 + \xi_0 - (1 - e^{-\kappa t})\frac{-\lambda_\chi}{\kappa} + \mu_\xi^* t \quad (2.2.5)$$

and

$$Var^*[\ln(S_t)] = \mathrm{Var}[\ln(S_t)] \quad (2.2.6)$$

Comparing the Equation 2.1.6 and 2.2.5, it is easy to find that risk premiums reduce the log of the expected spot price by $(1 - e^{-\kappa t})\frac{-\lambda_\chi}{\kappa} + \lambda_\xi \mathrm{t}$.

And in the risk-neutral valuation framework, futures prices are equal to the expected future spot prices. Thus, the relationship between futures prices and expected future spot prices can be expressed as

$$\ln\left(F_{T,0}\right) = \ln(E^*[S_T]) = E^*[\ln(S_T)] + \frac{1}{2}Var^*[\ln(S_T)]$$
$$= e^{-\kappa t}\chi_0 + \xi_0 + A(T) \quad (2.2.7)$$

Where

$$A(T) = \mu_\xi^* T - (1 - e^{-\kappa T})\frac{-\lambda_\chi}{\kappa} + \frac{1}{2}((1 - e^{-2\kappa T})\frac{\sigma_\chi^2}{2\kappa} + \sigma_\xi^2 T + 2(1 - e^{-\kappa T})\frac{\rho_{\chi\xi}\sigma_\chi\sigma_\xi}{\kappa})$$

From the Equation 2.2.7, $F_{T,0}$, denoting the current market price for a futures contract with time T until maturity, depends on the model parameters, the short-term deviations (χ_t), the equilibrium price level (ξ_t) and the maturity T. Thus, one can value futures contracts for any given T (including those that are no futures contracts trading) and generate the term structure for the futures prices with the short-term/long-term model if a set of model parameters and initial values of the two factors are given. However, the model's parameters are unknown. Moreover, the short-term deviation and the equilibrium price level are not directly observable. To deal with these two problems, the Kalman filter is introduced to do estimations for both parameters and state variables, which will be described in Sections 4.

4: Kalman Filter in Finance

As mentioned in the previous section, a Kalman filter can be applied to the estimation of a model's parameters, when the model relies on non-observable variables. In finance, for examples, there are term structure models of interest rates, term structure models of commodity prices, and the capital asset pricing model of market portfolios. Additionally, the Kalman filter is also an effective method to problems with a large volume of information as it is very fast. Lastly, the filter provides a set of optimal parameters when the model is associated with an optimization procedure. In this section, we begin with briefly reviewing the Kalman filter in Section 4.1 and then discuss its use in the Schwartz and Smith (2000) Model within Section 4.2.

4.1 Introduction to the Kalman Filter

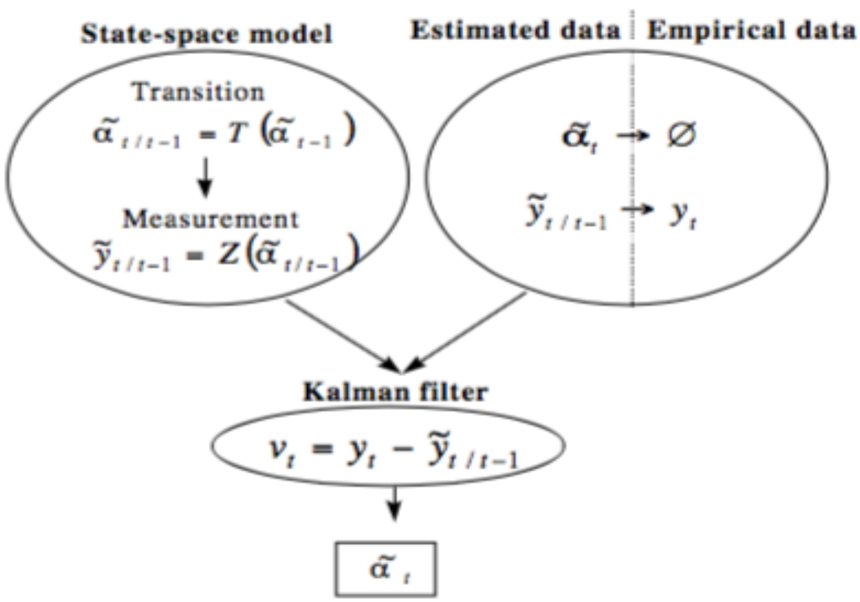
Under this sub-section, we first introduce the basic principle of the Kalman filter and problems that can be solved by it. Then we describe two forms of Kalman filter, which are simple and extended filters. And lastly, we discuss how to estimate model parameters using this tool.

The basic principle of the Kalman filter is the use of a temporal series of observable variables to reconstitute the value of the non-observable variables. The requirement of this method, first of all, is a state-space model, which is characterized by a transition equation and a measurement equation. And then a three-step iteration process begins once a model expressed on a state-space form.

Figure 3. represents the kind of problem a Kalman filter can resolve. The only information for non-observable variables ($\tilde{\alpha}$) that a model relies on is the transition equation,

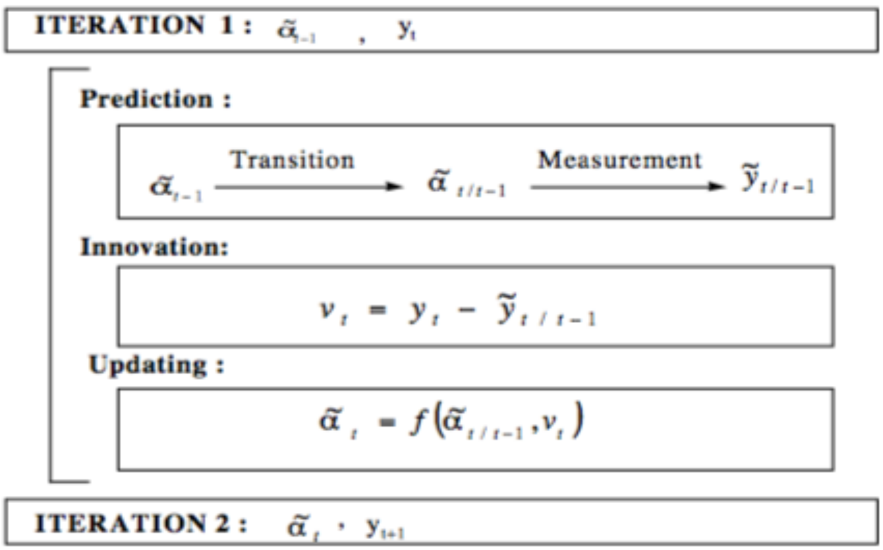
describing their dynamic. This equation gives predicted values of $\tilde{\alpha}$ at time t, conditionally to their values at time (t-1). Based on the calculation of $\tilde{\alpha}$, the measurement equation can determine the measure ($\tilde{y}$) at time t. And the differences, at time t, between the measure $\tilde{y}$ and the observable data (y) refer to the innovation (v), which represents some new information. Finally, this innovation is used to update the value of $\tilde{\alpha}$ at time t.

Figure 3. Basic Principle of Kalman Filter



In a word, there is one iteration for each observation date t: the Kalman filter first calculates values of $\tilde{\alpha}$ given their values at time (t-1), and then updates when some new information arrives. As shown in Figure 4, three phases are included in each iteration. During the prediction phase, the first step, the transition equation, and measurement equation give the estimated values of non-observables ($\tilde{\alpha}_{t/(t-1)}$) and measurement ($\tilde{y}_{t/(t-1)}$) at time t. And the second step, or innovation phase, calculates the innovation ($v_t = y_t - \tilde{y}_{t/(t-1)}$). And finally, conditionally to the information given by v_t, the updating phase re-estimates the values of non-observable variables that are computed in the prediction phase. Then the set of updated values for non-observable variables ($\tilde{\alpha}_t$) is used in the next iteration.

Figure 4. Three Steps of Iteration



Noticeably, there are two remarks in this figure. First, to estimate the values of $\tilde{\alpha}_t$ in the prediction phase, one must know the values of $\tilde{\alpha}_{t-1}$. Second, there are only two elements used to reconstitute temporal series for non-observable variables ($\tilde{\alpha}$), which are the transition equation and the innovation. And because there is an updating phased at each iteration, the volume of information used in very low, explaining the reason why the Kalman filter is a very fast method.

Then comes to the two versions of Kalman filter. When the transition and measurement equations are linear, the simple Kalman filter can be employed, which is the most frequently used version of the Kalman filter. However, when the model is non-linear, the extended Kalman filter can be used. As it is generally impossible to obtain an optimal estimator for the non-observable variables in a non-linear condition, the extended Kalman filter introduces an approximation in the estimation and leads to the linearization of the model.

For the parameter estimation, an initial vector of parameters is first used to compute all innovations of the given time period and the logarithms of the likelihood function for the innovations. Then the iterative procedure makes a search for the parameter's vector that maximizes the likelihood function and minimizes the innovations. And the optimal set of parameters is used to reconstitute the non-observable variables.

4.2 Application in the Short-Term/Long-Term Model

As indicated in Section 2, the state variables (χ_t and ξ_t) in the short-term/long-term model cannot be observed directly and must be estimated from the spot and/or futures prices. Meanwhile, Schwartz and Smith (2000) stated that there are two cases. First, if both short- and long-maturity futures contracts are traded, changes in the long-maturity futures prices give information about changes in the equilibrium price and changes in the differences in the short- and long-term futures prices give information about the short-term deviations. Second, if there are no traded long-maturity futures contracts, we may have to estimate the levels of the state variables and treat them probabilistically. And estimates in both cases can be generated by Kalman filter. Moreover, as mentioned in Section 3.1, the Kalman filter can also calculate the likelihood of observing a particular data series given a particular set of model parameters. And then find the optimal set of using maximum likelihood techniques.

Before discussing how the Kalman filter can be applied in the short-term/long-term model for estimating the parameters and two non-observable variables, it is necessary to show how this model can be transformed into a state-space form, which is a prerequisite for using Kalman filter. As the two non-observable factors are assumed to be state, we can derive the transition and measurement equations for the short-term/long-term model as:

$$x_t = c + Gx_{t-1} + \omega_t,\ \mathrm{t} = 1,2,\dots,n_T \qquad (3.2.1)$$

$$y_t = d_t + F_t' x_t + v_t,\ t = 1,2,\dots,n_T \qquad (3.2.2)$$

Where

$x_t = \begin{bmatrix} \chi_t \\ \xi_t \end{bmatrix}$, a 2 x 1 vector of state variables; c = $\begin{bmatrix} 0 \\ \mu_\xi \Delta t \end{bmatrix}$, a 2x1 vector;

G = $\begin{bmatrix} e^{-k\Delta t} & 0 \\ 0 & 1 \end{bmatrix}$, a 2 x 2 matrix; Δt =length of each time steps;

n_T = number of time periods in the data set; n = number of future contracts;

ω_t is a 2x1 vector of serially uncorrelated, normally distributed disturbances with E[ω_t] = 0, and Var[ω_t] = W = Cov[$(\chi_{\Delta t}, \xi_{\Delta t})$];

$y_t = \begin{bmatrix} lnF_{T1} \\ \vdots \\ lnF_{Tn} \end{bmatrix}$, a n x1 vector of observed log future prices with maturities $T_1, T_2, \dots, T_n$;

$y_t = \begin{bmatrix} A(T_1) \\ \vdots \\ A(T_n) \end{bmatrix}$, a n x 1 vector; $F_t' = \begin{bmatrix} e^{-\kappa T_1} & 1 \\ \vdots & \vdots \\ e^{-\kappa T_n} & 1 \end{bmatrix}$, a n x 2 matrix;

v_t is a n x 1 vector of serially uncorrelated, normally distributed disturbances with E[v_t] = 0, and Cov[v_t] = V.

In the transition equation, the matrix G and vector c specify how the 'true' and non-observable state vector (x_t) is expected to evolve from a one-time step to another. And in the measurement equation, the matrix F_t and vector d_t map the state vector into the measurement domain, which allows the estimated system states at time t to be transformed into a prediction for the measurement observation at time t. The residuals from this measurement predictions, denoted as v_t, are measurement errors and can be interpreted as errors in the reporting of prices, or errors in model's fit to observed prices. For simplicity, Schwartz and Smith (2000) assumed that the covariance matrix of measurement errors (V) is diagonal. And v_t and w_t are assumed to be independent of each other and uncorrelated with the initial state at all time periods.

To estimate model parameters, we suppose that non-observable variables and errors are normally distributed and compute the logarithm of the likelihood function for the innovation v_t at each iteration and for a given vector of parameters:

$$\log l\,(\mathrm{t}) = -\left(\frac{n}{2}\right) \times \ln(2\pi) - \frac{1}{2}\ln(dV_t) - \frac{1}{2} v_t' \times V_t^{-1} \times v_t \qquad (3.2.3)$$

And the iterative procedure makes a search for a vector of optimal parameters that maximize the likelihood function and minimizes the innovations.

5: Empirical Results

This section presents the results from calibrating the two-factor Schwartz and Smith Model to construct the futures curve (F-model) and Expected Spot curve (A-model).

Parameter values obtained using the Kalman filter for both models are reported in Appendix A, Table 2. Futures contract errors for F-model and Analyst Forecasts errors for A-model are computed and presented in Appendix A, Table 3. Figure 5, 6,7 and 8, from Appendix B, are graphs of the term structure of Futures curve with maturities of three months, two years, four years and ten years. Figure 9, 10 and 11 are graphs of the expected spot price term structure with maturities of three months, two years and four years.

By analyzing the model fit (Table3), we can see that F-model can better fit the data than A-model with a low mean absolute error. This is also shown in the model term structure figures in Appendix B. Also we can see that the model fit gets worse as we go further to the longer maturity. In figure 7(F-model with 4 years maturity), 8 (F-model with 10 years maturity) and 11(A-model with 4 years maturity), we can observe that the models were unable to fit the data very well.

It is hard to draw any economic reason for the negative correlation parameter for the future contract. This might be the model fails to filter out whether the change in price was due to change in the equilibrium price to the short-term deviation. In the paper (Goodwin & Larsson), the authors notice that as the periods for observation shorten, the correlation starts tends to be estimated to minus 1, and in shorter time frame there are little different in the movements in the long and short-term prices.

6: Conclusion

In this article, we applied the Schwartz and Smith Two-Factor model in copper derivative pricing. We were able to see that the Schwartz and Smith two-factor model was able to provide an intuitive explanation of the movement in Copper pricing.

By examining both the F-model and the A-model, we see that F-model has a better fit to the observation than the A-model since the Analyst forecast are more noisy than the F-model.

Appendices

Appendix A

Table 2. Parameter Estimations

Maximum-Likelihood Parameter Estimates

Parameter	Description	Futures Data (F-Model)			Analysts' Data (A-Model)		
			Estimation	Standard Error		Estimation	Standard Error
κ	Short-term mean-reversion rate		0.03	0.0029		1.2106	0.1119
σ_χ	Short-term volatility		0.5145	0.0163		0.1927	0.0359
λ_χ	Short-term risk premium		0.0488	0.0105		0	1
μ_ξ	Equilibrium drift rate		0.0069	0.0591		-0.0323	0.1008
σ_ξ	Equilibrium volatility		0.3878	0.0127		0.2003	0.0282
μ_ξ^*	Equilibrium risk-neutral drift rate		0.0035	0.0064		0	1
$\rho_{\chi\xi}$	Correlation in increments		-0.9264	0.0198		-0.628	0.0917
	Standard deviation(s) of error for measurement equation	Contract Matuity			Contract Matuity		
s_1		3 mo.	0.0142	0.0001	3 mo.	0.0201	0.0002
s_2		6 mo.	0.0103	0	6 mo.	0.013	0.0001
s_3		9 mo.	0.007	0	9 mo.	0.014	0.0001
s_4		1 yr.	0.0045	0	1 yr.	0.0194	0.0001
s_5		2 yr.	0.0025	0	1.25 yr.	0.0216	0.0002
s_6		3 yr.	0.0024	0	1.5 yr.	0.0255	0.0002
s_7		4 yr.	0.0024	0	2 yr.	0.0296	0.0003
s_8		5 yr.	0.0022	0	3 yr.	0.0818	0.0015
s_9		6 yr.	0.0031	0	4 yr.	0.1094	0.0028
s_{10}		7 yr.	0.005	0			
s_{11}		8 yr.	0.007	0			
s_{12}		9 yr.	0.014	0			
s_{13}		10 yr.	0.0225	0.0001			

Table 3. Model Fit: Mean Absolute Error for F-model and A-model for Each Maturity

Errors in the Model Fit to the Observations

	Futures Data (F-Mode)				Analysts' Data (A-Model)		
Contract maturity	Mean Error	S.D. for Error	Mean Absolute Error	Contract maturity	Mean Error	S.D. for Error	Mean Absolute Error
3 mo.	-0.0046	0.0137	0.0106	3 mo.	0.0018	0.0151	0.012
6 mo.	-0.0032	0.0097	0.0076	6 mo.	-0.0032	0.01	0.0082
9 mo.	-0.0018	0.0066	0.0053	9 mo.	-0.0027	0.0121	0.0094
1 yr.	-0.0005	0.0041	0.0033	1 yr.	0.0032	0.0178	0.0132
2 yr.	0.0009	0.002	0.0015	1.25 yr.	0.0073	0.0183	0.0144
3 yr.	0.0005	0.0022	0.0017	1.5 yr.	0.0099	0.0214	0.017
4 yr,	-0.0001	0.0022	0.0017	2 yr.	-0.008	0.0261	0.0209
5 yr.	-0.0002	0.002	0.0015	3 yr.	-0.0073	0.0814	0.054
6 yr.	-0.0006	0.0025	0.0016	4 yr.	-0.0164	0.1088	0.0826
7 yr.	-0.0007	0.0047	0.0033				
8 yr.	0.0007	0.0068	0.0053				
9 yr.	0.0035	0.0132	0.0097				
10 yr.	0.0067	0.0213	0.0146				

Table 4. Futures Prices Data

LME Copper Futures Prices from 2010 to 2015 (USD/Metrix Tone)

Date/Maturity	3M	6M	9M	1Y	2Y	3Y	4Y	5Y	6Y	7Y	8Y	9Y	10Y
Jul-10	6405.75	6429.00	6448.00	6459.50	6397.00	6220.00	6023.00	5834.00	5664.00	5554.00	5524.00	5512.00	5500.00
Oct-10	8177.50	8175.50	8154.50	8126.50	7897.50	7610.50	7320.50	7049.50	6784.50	6583.50	6406.50	6319.50	6259.50
Jan-11	9562.00	9515.00	9454.00	9381.00	8961.00	8511.00	8076.00	7661.00	7231.00	6831.00	6601.00	6541.00	6481.00
Apr-11	9387.50	9404.50	9410.00	9403.00	9204.00	8894.00	8579.00	8261.00	7931.00	7626.00	7416.00	7290.00	7230.00
Jul-11	9541.75	9551.00	9553.00	9535.00	9299.00	8999.00	8674.00	8348.00	8018.00	7718.00	7470.00	7254.00	7060.00
Oct-11	6816.00	6833.50	6845.00	6855.00	6846.00	6826.00	6780.00	6740.50	6685.00	6616.00	6529.00	6415.00	6349.00
Jan-12	7539.00	7551.75	7562.00	7566.00	7526.00	7456.00	7386.00	7316.00	7246.00	7158.00	7068.00	6978.00	6888.00
Apr-12	8362.50	8370.50	8378.00	8380.00	8339.50	8276.50	8182.00	8081.00	8003.00	7915.00	7825.00	7735.00	7642.00
Jul-12	7700.00	7694.50	7695.75	7696.50	7684.50	7660.00	7626.00	7586.00	7541.00	7485.00	7425.00	7365.00	7305.00
Oct-12	8298.00	8300.00	8302.50	8304.00	8289.00	8260.50	8220.00	8173.00	8122.00	8063.00	8003.00	7943.00	7883.00
Jan-13	8067.00	8087.25	8102.50	8113.00	8143.00	8167.00	8190.00	8210.00	8170.00	8110.00	8050.00	7990.00	7930.00
Apr-13	7403.00	7427.25	7452.00	7476.00	7564.50	7641.00	7708.50	7757.00	7757.00	7757.00	7757.00	7757.00	7757.00
Jul-13	6793.50	6789.00	6795.50	6812.00	6873.50	6926.00	6980.00	7011.00	7011.00	7011.00	7011.00	7011.00	7011.00
Oct-13	7240.25	7262.50	7281.00	7298.00	7359.50	7405.50	7442.75	7459.50	7465.25	7466.00	7466.00	7466.00	7466.00
Jan-14	7354.00	7340.25	7328.00	7316.00	7281.00	7246.00	7206.00	7212.00	7217.00	7217.00	7217.00	7217.00	7217.00
Apr-14	6678.00	6677.00	6680.00	6684.25	6698.00	6705.75	6707.00	6708.50	6712.00	6712.00	6712.00	6712.00	6712.00
Jul-14	7127.50	7121.00	7113.50	7104.00	7062.00	7012.50	6962.50	6941.00	6943.00	6943.00	6943.00	6943.00	6943.00
Oct-14	6678.50	6653.50	6639.50	6630.75	6587.25	6535.25	6491.50	6468.50	6462.50	6462.50	6462.50	6462.50	6462.50
Jan-15	6154.00	6124.00	6112.00	6102.00	6072.00	6048.00	6033.00	6003.00	6003.00	6003.00	6003.00	6003.00	6003.00
Apr-15	6071.50	6055.00	6046.00	6041.50	6027.00	6012.00	5988.50	5973.00	5973.00	5973.00	5973.00	5973.00	5973.00
Jul-15	5339.00	5348.50	5359.50	5372.25	5420.25	5449.00	5464.75	5472.25	5472.25	5472.25	5472.25	5472.25	5472.25
Oct-15	5181.00	5169.50	5165.50	5163.75	5160.75	5169.00	5188.00	5208.00	5211.00	5211.00	5211.00	5211.00	5211.00
Jan-16	4525.50	4520.75	4517.00	4514.00	4511.00	4534.00	4564.00	4581.00	4581.00	4581.00	4581.00	4581.00	4581.00
Apr-16	4777.00	4769.75	4764.75	4763.25	4766.25	4766.25	4768.50	4776.25	4776.25	4776.25	4776.25	4776.25	4776.25
Jul-16	4816.25	4826.25	4833.00	4841.25	4867.50	4894.50	4924.00	4949.50	4969.50	4979.00	4979.00	4979.00	4979.00
Oct-16	4798.50	4811.00	4823.25	4832.50	4860.75	4888.75	4918.00	4941.00	4961.00	4965.50	4965.50	4965.50	4965.50
Jan-17	5754.25	5765.25	5773.50	5776.00	5761.00	5746.00	5736.00	5736.00	5736.00	5736.00	5736.00	5736.00	5736.00
Apr-17	5894.75	5909.25	5920.25	5925.50	5939.25	5930.25	5915.25	5915.25	5915.25	5915.25	5915.25	5915.25	5915.25
Jul-17	5825.75	5850.00	5867.00	5883.50	5912.00	5922.50	5919.00	5919.00	5919.00	5919.00	5919.00	5919.00	5919.00
Oct-17	6752.00	6789.00	6815.50	6832.50	6872.50	6882.00	6880.00	6880.00	6880.00	6880.00	6880.00	6880.00	6880.00
Jan-18	7116.25	7149.75	7172.75	7190.00	7226.50	7231.50	7229.50	7229.50	7229.50	7229.50	7229.50	7229.50	7229.50
Apr-18	6812.75	6845.50	6876.50	6902.50	6964.50	6986.00	6986.00	6986.00	6986.00	6986.00	6986.00	6986.00	6986.00
Jul-18	6341.50	6358.50	6380.00	6398.50	6437.50	6452.50	6454.50	6454.50	6454.50	6454.50	6454.50	6454.50	6454.50
Oct-18	6173.00	6167.00	6170.50	6175.00	6184.50	6183.00	6183.00	6183.00	6183.00	6183.00	6183.00	6183.00	6183.00

LME Copper Futures Prices from 2010 to 2015 Grouped by Maturity Bucket

Maturity Bucket (Years)	Mean Price ($/Metric Tone)	Price S.D.	Mean Maturity (Years)	Min.Price ($/Metric Tone)	Max.Price ($/Metric Tone)	No. of Observations
0~1	6812.73	1322.32	0.50	4517.00	9562.00	102.00
1~5	6741.35	1191.76	3.00	4511.00	9535.00	204.00
6~10	6514.07	962.20	8.00	4581.00	8170.00	204.00
Total	6670.41	1146.99	4.35	4511.00	9562.00	510.00

Table 5. Analysts' Forecasts Data (Bloomberg)

LME Copper Analysts' Price Forecasts from 2010 to 2015 (USD/Metrix Tone)

Date/ Maturity	3M	6M	9M	1Y	1.25Y	1.5Y	2Y	3Y	4Y
Jan-10	6213.50	6250.00	6517.00	6900.00	6300.00	6650.00	7000.00	6834.00	6500.00
Apr-10	6944.00	6700.00	6850.00	7082.50	7267.50	7453.00	7533.87	7716.00	6668.86
Jul-10	6800.00	6900.00	7165.00	7617.50	7716.00	7932.50	7716.00	7716.00	6668.86
Oct-10	7450.00	7825.00	8050.00	7750.00	8350.00	8150.00	7950.00	8350.00	7330.22
Jan-11	8700.00	8914.00	9100.00	9400.00	9139.50	9479.00	9010.90	7273.22	8378.00
Apr-11	9900.00	9900.00	10200.00	10650.00	10925.00	11000.00	10325.00	8542.72	8189.00
Jul-11	9700.00	10050.00	10500.00	10850.00	11000.00	10500.00	10000.00	8542.72	8189.00
Oct-11	8550.00	8700.00	9000.00	9400.00	9075.00	9259.00	9231.50	8731.44	8250.00
Jan-12	8000.00	8400.00	8900.00	8708.00	9279.50	8818.00	8380.00	7957.88	6916.81
Apr-12	8400.00	8500.00	8625.00	8300.00	8420.00	8710.00	8338.50	7497.50	7082.50
Jul-12	8300.00	8377.56	8029.00	8200.00	8100.00	8000.00	8282.85	7500.00	7082.50
Oct-12	8200.00	8064.50	8212.50	8150.00	8050.00	8100.00	8125.00	7800.70	7469.00
Jan-13	8100.00	8300.00	8000.00	7960.50	7921.00	7800.50	7500.00	7250.00	6980.00
Apr-13	8150.00	8000.00	7900.00	7716.17	7716.17	7608.09	7500.00	7193.50	6500.35
Jul-13	7450.00	7575.00	7275.25	7500.00	7400.00	7328.02	7225.99	7287.62	7163.00
Oct-13	7100.00	7030.00	7000.00	6956.50	6917.16	6945.20	7056.50	6990.00	7000.00
Jan-14	6983.34	6875.00	6906.50	6816.67	6826.94	6772.27	6859.89	7200.00	6835.65
Apr-14	6900.00	6962.00	6944.00	6800.00	6844.00	6875.00	7022.54	7152.00	6809.50
Jul-14	6829.50	6867.00	6817.00	6930.00	7020.50	7197.00	6950.00	7013.00	6834.00
Oct-14	6834.00	6833.00	6900.00	6900.00	6941.50	6816.50	6903.00	7121.00	7055.00
Jan-15	6600.00	6700.00	6650.00	6950.00	7200.00	7209.00	7075.00	7038.00	7054.78
Apr-15	6184.50	6338.00	6436.95	6434.00	6619.40	6619.30	6576.32	6538.50	6389.00
Jul-15	6200.00	6470.00	6487.50	6600.00	6581.80	6500.00	6538.64	6538.50	6500.00
Oct-15	5475.00	5500.00	5700.00	5800.00	6100.00	6393.00	5800.00	6350.00	6777.00
Jan-16	5075.00	5071.00	5247.00	5500.00	5512.00	5291.00	5382.00	5500.00	5452.00
Apr-16	4700.00	4900.00	5006.00	5112.50	5300.00	5238.50	5050.50	5544.00	5085.50
Jul-16	4750.00	5000.00	5000.00	4980.00	4975.00	5340.50	4926.00	5088.00	5000.00
Oct-16	4825.00	4900.00	4960.35	5000.00	5181.00	5368.25	4960.35	5270.50	5534.50
Jan-17	5291.00	5300.00	5300.00	5291.00	5363.00	5300.00	5515.00	6000.00	6500.00
Apr-17	5750.00	5735.00	5647.22	5771.00	5900.00	5987.50	5750.00	6048.94	6601.50
Jul-17	5679.55	5650.00	5741.01	5900.00	5925.00	6100.00	5774.82	6094.00	6500.00
Oct-17	6100.00	5950.00	6000.00	6000.00	6050.00	6012.50	5975.00	6556.93	6600.00
Jan-18	6450.00	6500.00	6500.00	6650.00	6600.00	6600.00	6675.00	6656.90	7000.00
Apr-18	7100.00	7050.00	7000.00	7062.50	7000.00	7000.00	7000.00	7025.00	7250.00
Jul-18	7030.00	6982.50	7062.50	7100.00	7250.00	7300.00	7063.00	7150.00	7250.00
Oct-18	6435.00	6600.00	6800.00	6850.00	6783.33	6754.17	6725.00	6894.68	7165.00

LME Copper Analysts' Price Forecasts from 2010 to 2015 Grouped by Maturity Bucket

Maturity Bucket (Years)	Mean Price ($/Metric Tone)	Price S.D	Mean Maturity (Years)	Min.Price ($/Metric Tone)	Max.Price ($/Metric Tone)	No. of Observations
0~1	6993.04	1341.54	0.50	4700.00	10500.00	108.00
1~5	7091.53	1229.95	2.13	4926.00	11000.00	216.00
Total	7058.70	1267.01	1.58	4700.00	11000.00	324.00

Table 6. Analysts' Forecasts Data (World Bank)

World Banks Copper Analysts' Price Forecasts from 2000 to 2018
(USD/Metrix Tone)

Date/Maturity	3M	6M	9M	1Y	2Y	3Y	4Y	5Y	6Y	7Y	8Y	9Y	10Y	11Y	12Y	13Y	14Y	15Y
Jan-00				1800.00	1900.00	2000.00		2200.00					2400.00					
Apr-00				1800.00	1900.00	2000.00		2200.00					2400.00					
Nov-00				1808.00	1899.00	1891.00		1894.00					1867.00					
Oct-01					1681.00	1784.00		1880.00					1829.00					1783.00
Nov-02	1602.00			1661.00		1834.00					1799.00					1730.00		
Jul-03		1650.00			1800.00	1900.00				2000.00					2050.00			
Feb-04			2400.00		2200.00	2000.00				2000.00					2050.00			
Sep-04	2725.00			2400.00	2000.00				2000.00					2050.00				
Jan-05				2850.00	2400.00	2100.00			2000.00					2050.00				
Feb-06			4300.00		3500.00	2800.00		2400.00					2300.00					
Jul-07		7000.00			6000.00	4500.00	3500.00					2850.00						
Sep-07	7200.00				6500.00	5500.00	4500.00					3275.00					3400.00	
May-08		7600.00			7000.00	6000.00					3600.00					3825.00		
Jan-09				3700.00	4000.00	4200.00					4525.00					4650.00		
Jan-10				5800.00	6169.00	5364.00			4052.00					4031.00				
Jan-11				9000.00	8500.00	8000.00		6000.00					5700.00					
Jul-11		9250.00		9500.00	8500.00	8000.00	7500.00	7000.00	6500.00	6000.00	5750.00	5500.00						
Jan-12				8500.00	9000.00	8000.00	7000.00	6500.00	6000.00	5500.00	5750.00	6000.00					6500.00	
Jan-13				7800.00	7400.00	7000.00	6980.00	6960.00	6939.00	6919.00	6899.00					6800.00		
Apr-14		6471.00	6471.00		6314.00	6211.00	6128.00	6033.00	5935.00	5836.00	5736.00	5539.00	5442.00	5347.00				
Apr-15	6050.00			6118.00	6187.00	6257.00	6328.00	6399.00	6471.00	6544.00	6618.00	6800.00						
Oct-15	5900.00			5984.00	6070.00	6157.00	6245.00	6334.00	6425.00	6516.00	6610.00	6704.00						
Oct-16	4367.00			4490.00	4619.00	4752.00	4889.00					5614.00						
Oct-17	6050.00			6118.00	6187.00	6257.00	6328.00	6399.00	6471.00	6544.00	6618.00	6800.00					7000.00	
Apr-18			7043.00	6923.00	6824.00	6727.00	66627.00	6529.00	6432.00	6337.00					5871.00			
Oct-18	6732.00			6642.00	6572.00	6503.00	6431.00	6359.00	6289.00	6219.00					5871.00			

Appendix B

Approximate for Different Maturities

Figure 5. Futures Price Observations for an Approximate Maturity of Three-Month and the Corresponding F-Model Prices

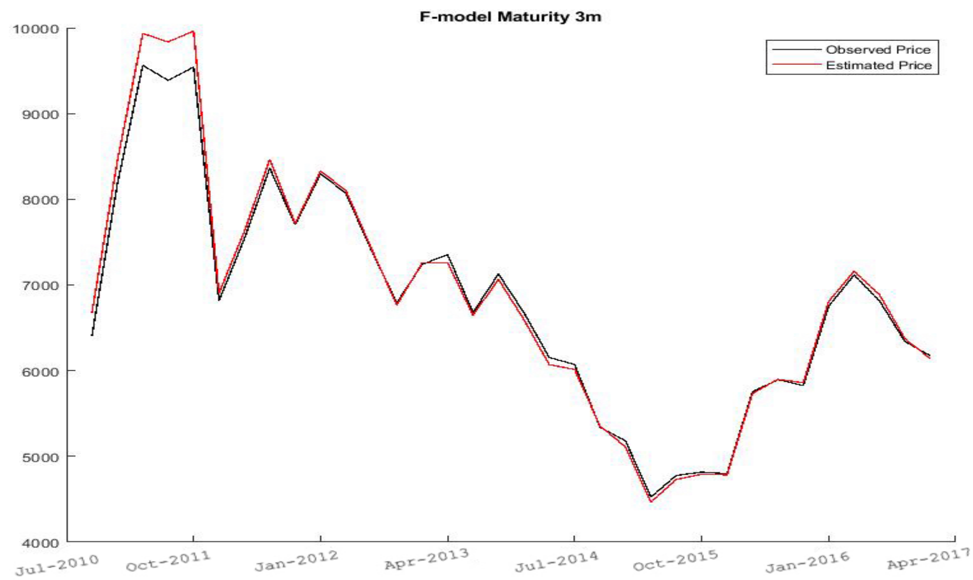


Figure 6. Futures Price Observations for an Approximate Maturity of Two-Year and the Corresponding F-Model Prices

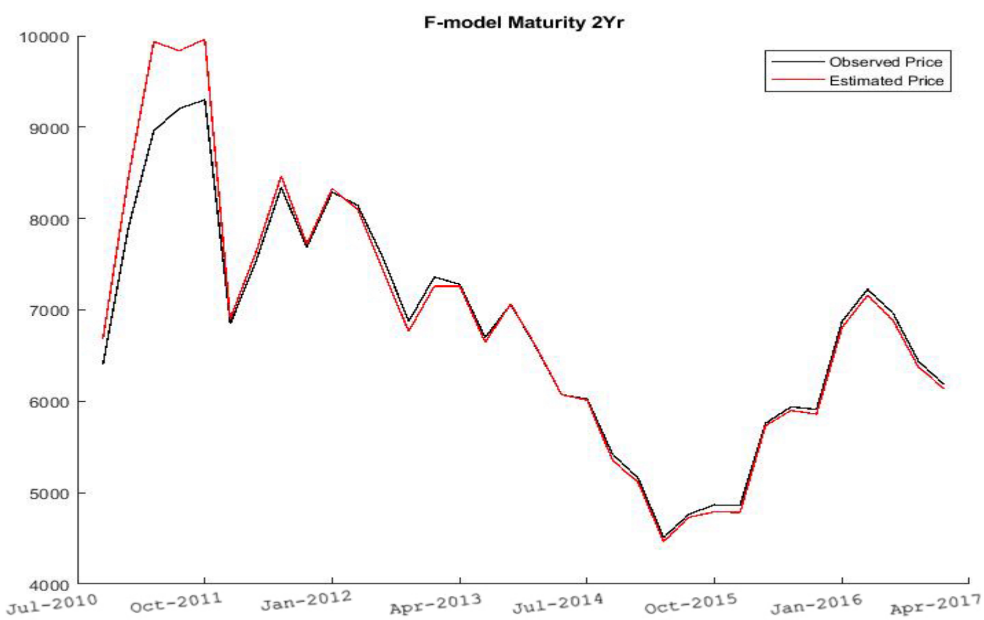


Figure 7. Futures Price Observations for an Approximate Maturity of Four-Year and the Corresponding F-Model Prices

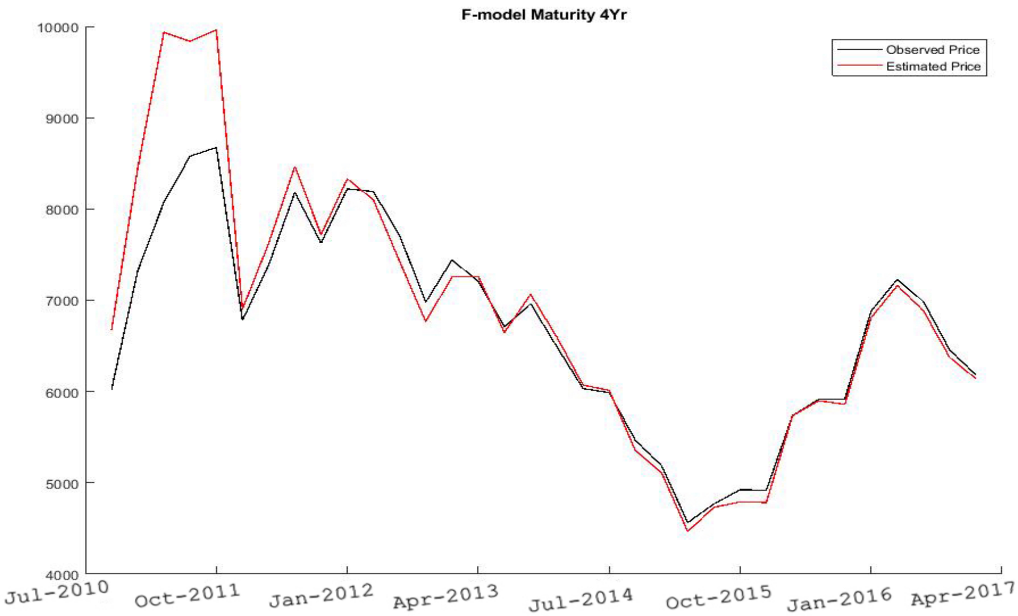


Figure 8. Futures Price Observations for an Approximate Maturity of Ten-Year and the Corresponding F-Model Prices

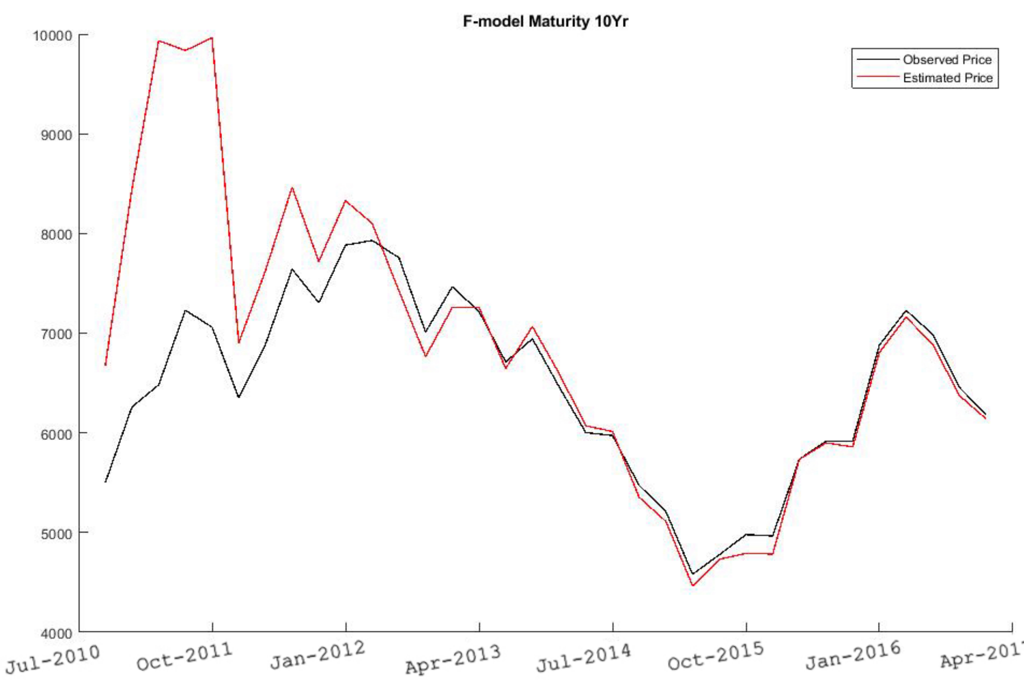


Figure 9. Analysts' Forecast Observations for an Approximate Maturity of Three Month and the Corresponding A-Model Prices

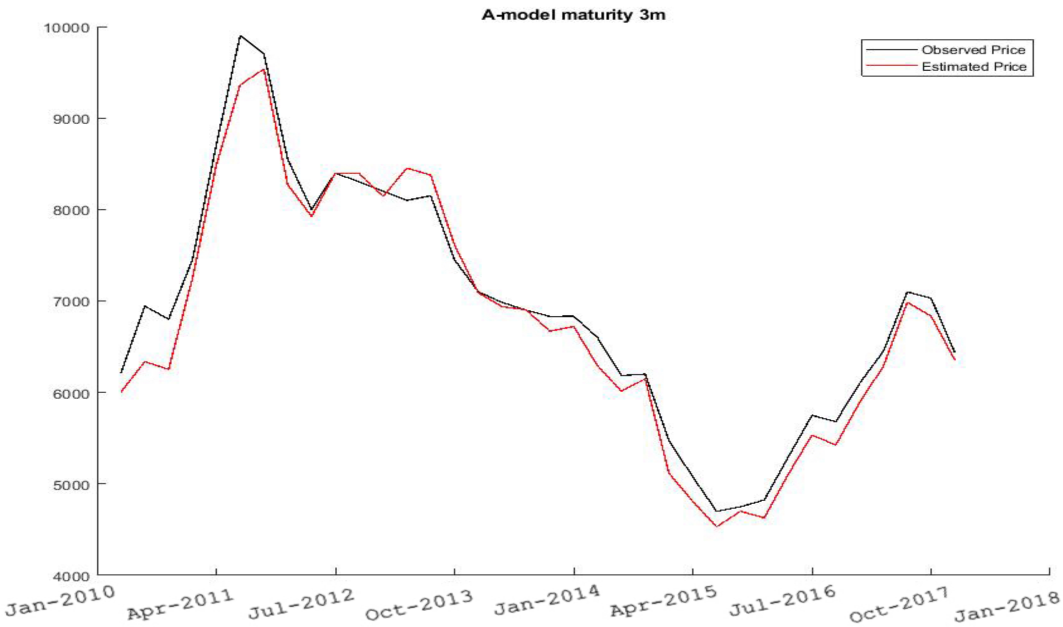


Figure 10. Analysts' Forecast Observations for an Approximate Maturity of Two-Year and the Corresponding A-Model Prices

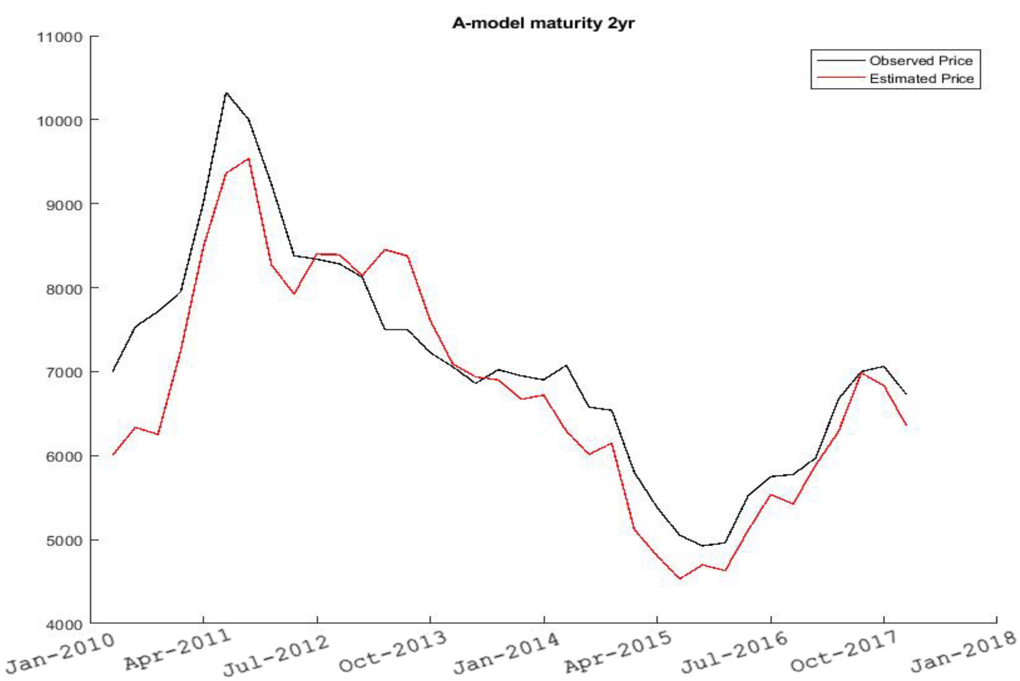
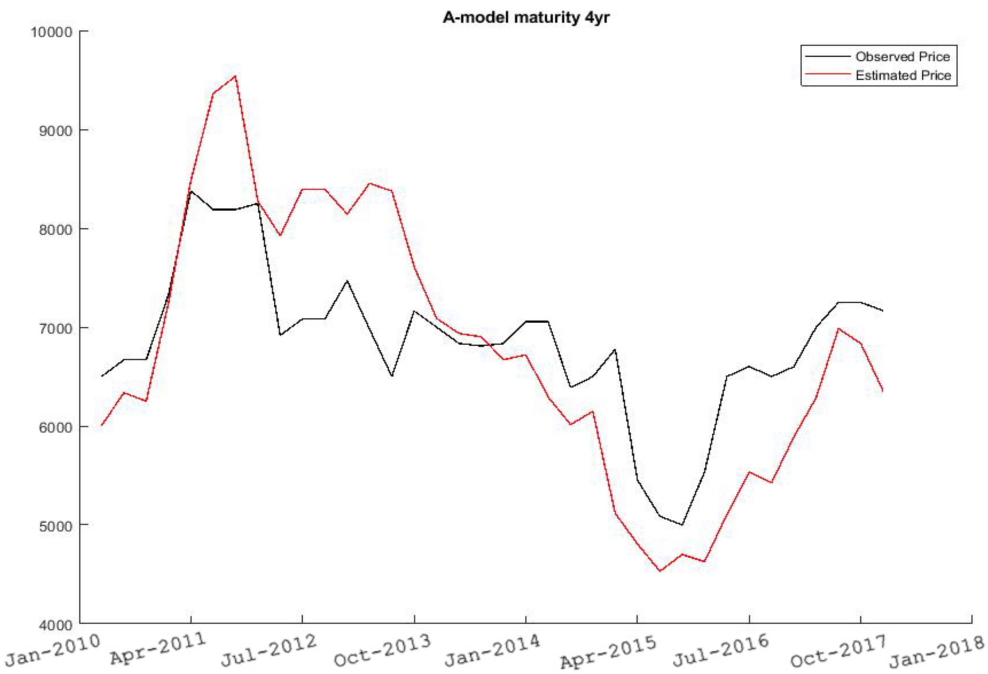


Figure 11. Analysts' Forecast Observations for an Approximate Maturity of Four-Year and the Corresponding A-Model Prices



Appendix C

Code for Futures Data

```
function log_L = Kalman_Estimation(y, psi, matur, dt, a0, P0, N, nobs,
locked_parameters)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Extracting initial parameter values from initial psi
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
k = psi(1,1);
sigmax = psi(2,1);
lambdax = psi(3,1);
mu = psi(4,1);
sigmae = psi(5,1);
rnmu = psi(6,1);
pxe = psi(7,1);
if sum(locked_parameters) == 0
    k = psi(1,1);
    sigmax = psi(2,1);
    lambdax = psi(3,1);
    mu = psi(4,1);
    sigmae = psi(5,1);
    rnmu = psi(6,1);
    pxe = psi(7,1);

    s = zeros(1, size(psi,1)-7);
    for i = 1:size(s,2)
        s(1, i) = psi(i+7,1);
    end
end

if sum(locked_parameters) ~= 0
    s = zeros(1, size(psi,1)-7+size(locked_parameters,1));
    j = 1;
    for i = 1:size(s,2)
        if all(abs(i-(locked_parameters))) == 1
             s(1, i) = psi(7+j,1);
             j = j+1;
        end
    end
end

% m = Number of state variables (number of rows in a0)
m = size(a0,1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% THE TRANSITION EQUATION
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% S&S NOTATION: x(t)=c+G*x(t-1)+w(t)        w~N(0,W)    Equation (14)
% NEW NOTATION: a(t)=c+T*a(t-1)+R(t)*n(t)   n~N(0,Q)
% c is a {m x 1} Vector
% T is a {m x m} Matrix
c=[0;mu*dt];
T=[exp(-k*dt),0;0,1];
% Defining Q = var[n(t)] and R
xx=(1-exp(-2*k*dt))*(sigmax)^2/(2*k);
xy=(1-exp(-k*dt))*pxe*sigmax*sigmae/k;
```

```
yx=(1-exp(-k*dt))*pxe*sigmax*sigmae/k;
yy=(sigmae)^2*dt;
Q=[xx,xy;yx,yy];
R=eye(size(Q,1));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% THE MEASUREMENT EQUATION
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% S&S NOTATION: y(t)=d(t)+F(t)'x(t)+v(t)    v~N(0,V) Equation (15)
% NEW NOTATION: y(t)=d(t)+Z(t)a(t)+e(t)     e~N(0,H)
% d is a {N x 1} Vector
% Z is a {N x m} Matrix
    for i=1:N
        p1=(1-exp(-2*k*matur(i)))*(sigmax)^2/(2*k);
        p2=(sigmae)^2*matur(i);
        p3=2*(1-exp(-k*matur(i)))*pxe*sigmax*sigmae/k;
        d(i,1)=rnmu*matur(i)-(1-exp(-k*matur(i)))*lambdax/k+.5*(p1+p2+p3);
        Z(i,1)=exp(-k*matur(i));
        Z(i,2)=1;
    end
% Measurment errors Var-Cov Matrix: Cov[e(t)]=H
H=diag(s);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% RUNNING THE KALMAN FILTER
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Creating placeholder vectors/matrices for variables to be stored in
global save_vt save_att save_dFtt_1 save_vFv save_vtt save_Ptt_1 save_Ftt_1
save_Ptt
save_ytt_1 = zeros(nobs,N);
save_vtt = zeros(nobs,N);
save_vt    = zeros(nobs,N);
save_att_1 = zeros(nobs,m);
save_att   = zeros(nobs,m);
save_Ptt_1 = zeros(nobs,m*m);
save_Ptt   = zeros(nobs,m*m);
save_Ftt_1 = zeros(nobs,N*N);
save_dFtt_1 = zeros(nobs,1);
save_vFv    = zeros(nobs,1);
%save_log_Lt   = zeros(nobs,1);
Ptt = P0;
att = a0;
% Running the kalman filter for t = 1,...,nobs
    for t = 1:nobs
        Ptt_1   = T*Ptt*T'+R*Q*R';
        Ftt_1   = Z*Ptt_1*Z'+H;
        dFtt_1  = det(Ftt_1);

        att_1   = T*att + c;
        yt      = y(t,:)';
        ytt_1   = Z*att_1+d;
        vt      = yt-ytt_1;
        att = att_1 + Ptt_1*Z'*inv(Ftt_1)*(vt);
        Ptt = Ptt_1 - Ptt_1*Z'*inv(Ftt_1)*Z*Ptt_1;

        ytt = Z*att+d;
        vtt  = yt-ytt;

        save_vtt(t,:) = vtt';
        save_vt(t,:)    = (vt)';
```

```
        save_att(t,:)   = att';
        save_Ptt_1(t,:) = [Ptt_1(1,1), Ptt_1(1,2), Ptt_1(2,1), Ptt_1(2,2)];
        save_Ptt(t,:)   = [Ptt(1,1), Ptt(1,2), Ptt(2,1), Ptt(2,2)];

        save_dFtt_1(t,:)= dFtt_1;
        save_vFv(t,:)   = vt'*inv(Ftt_1)*vt;

    end

logL = -(N*nobs/2)*log(2*pi)-0.5*sum(log(save_dFtt_1))-0.5*sum(save_vFv);

log_L = -logL;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This Matlab Script estimates the parameters of the model presented in
Schwartz-Smith
% (2000) paper(Short-Term Variations and Long-Term Dynamics in Commodity
Prices).
% NOTE: it can take up to 10 minutes for the estimation to complete.
%
% Code originally produced by Dominice Goodwin (May 2013) to conduct the
empirical study in
% master thesis D. Goodwin (2013), Xiaoyu Fu and stella modify the code to
for Final Project paper:
% (http://www.lunduniversity.lu.se/o.o.i.s?id=24965&postid=3809118)
%
% Contact: xfa17@sfu.ca zpa9@sfu.ca
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 format short; % Spot data in first column. All price in log.
which_model = 1;
% [1 = Schwartz-Smith (2000) Model on the approximately the same Crude Oil
% data as used in this article.]
if which_model == 1 % Schwartz-Smith (2000) on crude oil data

    %%% INPUT SETTINGS %%%
    data = LMEFuturesS1{:,:};                              % Specify which variable
that contains data for estimation (Column1 = Spot, Column2 = Future(Shortest
Maturity)...)
    include_spot_in_estimation = 1 ;          % [0 = No, 1 = Yes (Include the
first column of Spot data in estimation)]
    Num_Contracts = 13;                       % # of future contracts in data
to use
    matur = [3/12,6/12,9/12,1,2,3,4,5,6,7,8,9,10];    % Maturities of included
contracts
    frequency = 1;                           % [1 = all observations in data
variables are considered, 2 = every second observation is considered, ...]
(This data is weekly .. so frequency = 1 -> weekly frequency.
```

```
    dt = 90/360;                                % Time step size (Since weekly
data) to get parameters on per year basis.
    start_obs = 1;                             % Start at first observation in
data.
    end_obs = 34;                             % End at last observation in data.

% The standard errors are obtained from the hessian. However, since the model
estimates the parameters
% so that the one or a couple of futures contracts are matched with close to
zero measurement errors,
% leading to that the measurement error covariance matrix (usually) is
positive semi-defined.
% --> Matlab error: Warning: Matrix is close to singular or badly scaled.
Results may be inaccurate.
% To be able to invert the hessian and obtain standard errors the following
% ad hoc approach can be used:
%  - Once it is known which of the future contracts is matched with close to
zero measurement errors
% the estimation can be redone with the corresponding elements in measurement
error covariance matrix
% restricted to zero and thus excluded from the estimation. In this way
measurement error covariance matrix
% is positive defined and invertible.
    locked_parameters = 0;                     % [ 0 = No parameter locked, 1 to
... = Forces a measurement error parameter to be Zero]
                                               % OBS: This data requiers
locked_parameters = 0;

    %%% SELECT INITIAL VALUES %%%
    k       = 1.49;                               % NOTE: These initial values
have to be changed manually in order to find a Global Maximum Log-Likelihood
Score
    sigmax  = 0.286;
    lambdax = 0.157;
    mu      = -0.0125;
    sigmae  = 0.145;
    rnmu    = 0.0115;
    pxe     = 0.3;
    s_guess = 0.005;
    initial_statevector = [0;3.1307];       % Initial state vector
m(t)=E[xt;et]
    initial_dist = [0.01,0.01;0.01,0.01];   % Initial covariance matrix for
the state variables C(t)=cov[xt,et]
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% ADJUSTING DATA ACCORDING TO INPUTS %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
data_SelectedPeriod = data(start_obs:end_obs,1:end);
num_obs = size(data_SelectedPeriod,1);
if frequency ~= 1
    new_num_obs = floor((num_obs-1)/frequency);
    data_SelectedPeriod_SelectedFrequency =
zeros(new_num_obs,size(data_SelectedPeriod,2));
    data_SelectedPeriod_SelectedFrequency(1,:) = data_SelectedPeriod(1,:);
    for t = 1:new_num_obs
        data_SelectedPeriod_SelectedFrequency(t+1,:) =
data_SelectedPeriod((t*frequency)+1,:);
    end
```

```
else
    data_SelectedPeriod_SelectedFrequency = data_SelectedPeriod;
end
St = data_SelectedPeriod_SelectedFrequency(1:end,1);
if include_spot_in_estimation == 1
    y  = data_SelectedPeriod_SelectedFrequency(1:end,1:Num_Contracts);
else
    y  = data_SelectedPeriod_SelectedFrequency(1:end,2:Num_Contracts+1);
end
% y is a {nobs x N} Matrix, N = number of future contracts, nobs = number of
observations
nobs = size(y,1);
N    = size(y,2);
num_locked_parameters = size(locked_parameters,1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Optimizing the parameters with the Kalman filter & MLE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Placeholders & Variable def.
global save_att save_vtt save_vt  save_dFtt_1 save_vFv save_Ptt_1 save_Ftt_1
save_Ptt
lnL_scores = zeros(3,1);
boundary = Inf;
% Running the estimation for The S&S 2 factor model and two benchmark
% models (The GBM model and the Ornstein-Uhlenbeck model).
for model = 1 % [1 = The S&S 2 factor model]
    if model == 1 % The S&S 2 factor model
        if sum(locked_parameters) == 0

            psi = zeros(7+N,1);
            psi(1:7,1) = [k, sigmax, lambdax, mu, sigmae, rnmu, pxe]';
            psi(8:end,1) = s_guess;

            lb = zeros(7+N,1);
            lb(1:7,1) = [0, 0, -boundary, -boundary, 0, -boundary, -1]';
            lb(8:end,1) = 0.0000001;

            ub = zeros(7+N,1);
            ub(1:7,1) = [boundary, boundary, boundary, boundary, boundary,
boundary, 1]';
            ub(8:end,1) = boundary;
        else
            psi = zeros(7+N-num_locked_parameters,1);
            psi(1:7,1) = [k, sigmax, lambdax, mu, sigmae, rnmu, pxe]';
            psi(8:end,1) = s_guess;

            lb = zeros(7+N-num_locked_parameters,1);
            lb(1:7,1) = [0, 0, -boundary, -boundary, 0, -boundary, -1]';
            lb(8:end,1) = 0.0000001;

            ub = zeros(7+N-num_locked_parameters,1);
            ub(1:7,1) = [boundary, boundary, boundary, boundary, boundary,
boundary, 1]';
            ub(8:end,1) = boundary;
        end
        a0 = initial_statevector;
        P0 = initial_dist;
    end
```

```
    % Running estimation
    options = optimset('Algorithm','interior-point','Display','off');
%interior-point active-set
    MaxlnL_Kalman = @(psi) Kalman_Estimation(y, psi, matur, dt, a0, P0, N,
nobs, locked_parameters);
    [psi_optimized, log_L,exitflag,output,lambda,grad,hessian] =
fmincon(MaxlnL_Kalman, psi, [], [],[], [], lb, ub, [], options);
    % Saving estimation output
    lnL_scores(model,1) = -log_L;
    if model == 1
        ss_att = save_att;
        ss_vtt = save_vtt;
        ss_vt = save_vt;
        ss_dFtt_1 = save_dFtt_1;
        ss_vFv = save_vFv;
        ss_Ptt_1 = save_Ptt_1;
        ss_Ftt_1 = save_Ftt_1;
        ss_Ptt = save_Ptt;

        if sum(locked_parameters) == 0
             ss_psi_estimate =
[psi_optimized(1:7,1);sqrt(psi_optimized(8:end,1))];
             ss_SE = sqrt(diag(inv(hessian)));
        else
            prel_SE = sqrt(diag(inv(hessian)));
            prel_ss_psi_estimate =
zeros(size(psi,1)+size(locked_parameters,1),1);
            ss_SE = zeros(size(psi,1)+size(locked_parameters,1),1);
            j = 1;
            for i = 1:size(prel_ss_psi_estimate,1)
                 if all(abs(i-(locked_parameters+7))) == 1
                     prel_ss_psi_estimate(i,1) = psi_optimized(j,1);
                     ss_SE(i,1) = prel_SE(j,1);
                     j = j+1;
                 else
                     prel_ss_psi_estimate(i,1) = 0;
                     ss_SE(i,1) = 0;
                 end
            end
            ss_psi_estimate =
[prel_ss_psi_estimate(1:7,1);sqrt(prel_ss_psi_estimate(8:end,1))];
         end
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calculating/outputing key statistics
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Output
ss_psi_estimate
ss_SE

% S&S Model fit
ss_Mean_Error = mean(ss_vtt)'
ss_Std_of_Error = std(ss_vtt)'
ss_MAE = mean(abs(ss_vtt))'

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Outputing Graph
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(1);
set(figure(1), 'Position', [100 100 400 1000])
hold on
plot(exp(St),'k','linewidth',1);
plot(exp(ss_att(:,1)+ss_att(:,2)),'r','linewidth',1);
plot(exp(ss_att(:,2)),'b','linewidth',1);
h = legend('Observed Price','Estimated Price','Equilibrium Price');
title('Schwartz-Smith 2-factor model')
hold off
```

Code for Analysts' Forecast Data

```
function log_L = Kalman_Estimation_Real(y, psi, matur, dt, a0, P0, N, nobs,
locked_parameters)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Extracting initial parameter values from initial psi
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
k = psi(1,1);
sigmax = psi(2,1);
lambdax = psi(3,1);
mu = psi(4,1);
sigmae = psi(5,1);
rnmu = psi(6,1);
pxe = psi(7,1);
if sum(locked_parameters) == 0
    k = psi(1,1);
    sigmax = psi(2,1);
    lambdax = psi(3,1);
    mu = psi(4,1);
    sigmae = psi(5,1);
    rnmu = psi(6,1);
    pxe = psi(7,1);

    s = zeros(1, size(psi,1)-7);
    for i = 1:size(s,2)
        s(1, i) = psi(i+7,1);
    end
end

if sum(locked_parameters) ~= 0
    s = zeros(1, size(psi,1)-7+size(locked_parameters,1));
    j = 1;
    for i = 1:size(s,2)
        if all(abs(i-(locked_parameters))) == 1
             s(1, i) = psi(7+j,1);
             j = j+1;
        end
    end
end

% m = Number of state variables (number of rows in a0)
m = size(a0,1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% THE TRANSITION EQUATION
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% S&S NOTATION: x(t)=c+G*x(t-1)+w(t)        w~N(0,W)    Equation (14)
% NEW NOTATION: a(t)=c+T*a(t-1)+R(t)*n(t)   n~N(0,Q)
% c is a {m x 1} Vector
% T is a {m x m} Matrix
c=[0;mu*dt];
T=[exp(-k*dt),0;0,1];
% Defining Q = var[n(t)] and R
xx=(1-exp(-2*k*dt))*(sigmax)^2/(2*k);
xy=(1-exp(-k*dt))*pxe*sigmax*sigmae/k;
yx=(1-exp(-k*dt))*pxe*sigmax*sigmae/k;
yy=(sigmae)^2*dt;
```

```
Q=[xx,xy;yx,yy];
R=eye(size(Q,1)); %R is a (2*2) indentity matrix with rows and column equals
to the number of rows of Q
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% THE MEASUREMENT EQUATION
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% S&S NOTATION: y(t)=d(t)+F(t)'x(t)+v(t)    v~N(0,V) Equation (15)
% NEW NOTATION: y(t)=d(t)+Z(t)a(t)+e(t)     e~N(0,H)
% d is a {N x 1} Vector
% Z is a {N x m} Matrix
    for i=1:N
        p1=(1-exp(-2*k*matur(i)))*(sigmax)^2/(2*k);
        p2=(sigmae)^2*matur(i);
        p3=2*(1-exp(-k*matur(i)))*pxe*sigmax*sigmae/k;
        d(i,1)=mu*matur(i)+.5*(p1+p2+p3);
        Z(i,1)=exp(-k*matur(i));
        Z(i,2)=1;
    end
% Measurment errors Var-Cov Matrix: Cov[e(t)]=H
H=diag(s);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% RUNNING THE KALMAN FILTER
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Creating placeholder vectors/matrices for variables to be stored in
global save_vt save_att save_dFtt_1 save_vFv save_vtt save_Ptt_1 save_Ftt_1
save_Ptt
save_ytt_1 = zeros(nobs,N);
save_vtt = zeros(nobs,N);
save_vt    = zeros(nobs,N);
save_att_1 = zeros(nobs,m);
save_att   = zeros(nobs,m);
save_Ptt_1 = zeros(nobs,m*m);
save_Ptt   = zeros(nobs,m*m);
save_Ftt_1 = zeros(nobs,N*N);
save_dFtt_1 = zeros(nobs,1);
save_vFv    = zeros(nobs,1);
%save_log_Lt   = zeros(nobs,1);
Ptt = P0;
att = a0;
% Running the kalman filter for t = 1,...,nobs
    for t = 1:nobs
        Ptt_1    = T*Ptt*T'+R*Q*R';
        Ftt_1    = Z*Ptt_1*Z'+H;
        dFtt_1   = det(Ftt_1);

        att_1    = T*att + c;
        yt       = y(t,:)';
        ytt_1    = Z*att_1+d;
        vt       = yt-ytt_1;
        att = att_1 + Ptt_1*Z'*inv(Ftt_1)*(vt);
        Ptt = Ptt_1 - Ptt_1*Z'*inv(Ftt_1)*Z*Ptt_1;

        ytt = Z*att+d;
        vtt  = yt-ytt;
        % save_ytt_1(t,:) = ytt_1';
        save_vtt(t,:) = vtt';
        save_vt(t,:)    = (vt)';
        % save_att_1(t,:) = att_1';
```

```
        save_att(t,:)   = att';
        save_Ptt_1(t,:) = [Ptt_1(1,1), Ptt_1(1,2), Ptt_1(2,1), Ptt_1(2,2)];
        save_Ptt(t,:)   = [Ptt(1,1), Ptt(1,2), Ptt(2,1), Ptt(2,2)];

        save_dFtt_1(t,:)= dFtt_1;
        save_vFv(t,:)   = vt'*inv(Ftt_1)*vt;

    end

logL = -(N*nobs/2)*log(2*pi)-0.5*sum(log(save_dFtt_1))-0.5*sum(save_vFv);

log_L = -logL;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This Matlab Script estimates the parameters of the model presented in
Schwartz-Smith
% (2000) paper(Short-Term Variations and Long-Term Dynamics in Commodity
Prices).
% NOTE: it can take up to 10 minutes for the estimation to complete, depend
% on amount of data you use for this code
%
%
% Originally produced by Dominice Goodwin (May 2013) to conduct the empirical
study in
% master thesis,modify by Xiaoyu Fu and Zheng Peng to conduct research on
% using Analyst forecast for real distribution of expected sopt price
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
format short; % Spot data in first column. All prices in log.
which_model = 1;
% [1 = Schwartz-Smith (2000) Model on the approximately the same Crude Oil
% data as used in this article is extracted from the file AnalystForecast,
% we first imported the data in Commend window as table, then we run this
% code.
if which_model == 1 % Schwartz-Smith (2000) on crude oil data

    %%% INPUT SETTINGS %%%
    data = AnalystForecast{:,:};            % Specify which variable that
contains data for estimation (Column1 = Future(Shortest
Maturity)...Future(Longest Maturity))
    include_spot_in_estimation = 1;         % [0 = No, 1 = Yes (Include the
first column of Spot data in estimation)]
    Num_Contracts = 9;                      % # of future contracts of
different maturity
    matur = [3/12,6/12,9/12,1,1.25,1.5,2,3,4];   % Maturities of included
contracts
    frequency = 1;                          % [1 = all observations in data
variables are considered, 2 = every second observation is considered, ...]
(This data is weekly .. so frequency = 1 -> weekly frequency.
    dt = 90/360;                             % Time step size (Since weekly
data) to get parameters on per year basis.
    start_obs = 1;                          % Start at first observation in
data.
    end_obs = 36;                          % End at last observation in data.

% The standard errors are obtained from the hessian. However, since the model
estimates the parameters
```

```
% so that the one or a couple of futures contracts are matched with close to
zero measurement errors,
% leading to that the measurement error covariance matrix (usually) is
positive semi-defined.
% --> Matlab error: Warning: Matrix is close to singular or badly scaled.
Results may be inaccurate.
% To be able to invert the hessian and obtain standard errors the following
% ad hoc approach can be used:
%  - Once it is known which of the future contracts is matched with close to
zero measurement errors
% the estimation can be redone with the corresponding elements in measurement
error covariance matrix
% restricted to zero and thus excluded from the estimation. In this way
measurement error covariance matrix
% is positive defined and invertible.
    locked_parameters = 0;                  % [ 0 = No parameter locked, 1 to
... = Forces a measurement error parameter to be Zero]
                                            % OBS: This data requiers
locked_parameters = 4;

    %%% SELECT INITIAL VALUES %%%
    k       = 1.48;                            % NOTE: These initial values
have to be changed manually in order to find a Global Maximum Log-Likelihood
Score
    sigmax  = 0.286;                          % NOTE: For this paper we used
the parameter from the Schwartz-Smith (2000) Model
    lambdax = 0;
    mu      = -0.0125;
    sigmae  = 0.145;
    rnmu    = 0;
    pxe     = 0.3;
    s_guess = 0.005;
    initial_statevector = [0;3.1307];       % Initial state vector
m(t)=E[xt;et]
    initial_dist = [0.01,0.01;0.01,0.01];   % Initial covariance matrix for
the state variables C(t)=cov[xt,et]
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% ADJUSTING DATA ACCORDING TO INPUTS %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
data_SelectedPeriod = data(start_obs:end_obs,1:end);
num_obs = size(data_SelectedPeriod,1);
if frequency ~= 1
    new_num_obs = floor((num_obs-1)/frequency);
    data_SelectedPeriod_SelectedFrequency =
zeros(new_num_obs,size(data_SelectedPeriod,2));
    data_SelectedPeriod_SelectedFrequency(1,:) = data_SelectedPeriod(1,:);
    for t = 1:new_num_obs
        data_SelectedPeriod_SelectedFrequency(t+1,:) =
data_SelectedPeriod((t*frequency)+1,:);
    end
else
    data_SelectedPeriod_SelectedFrequency = data_SelectedPeriod;
end
St = data_SelectedPeriod_SelectedFrequency(1:end,1);
if include_spot_in_estimation == 1
    y  = data_SelectedPeriod_SelectedFrequency(1:end,1:Num_Contracts);
else
```

```
    y  = data_SelectedPeriod_SelectedFrequency(1:end,2:Num_Contracts+1);
end
% y is a {nobs x N} Matrix, N = number of future contracts, nobs = number of
observations
nobs = size(y,1); %nobs is the number of rows of y
N    = size(y,2); %N is the number of column of y
num_locked_parameters = size(locked_parameters,1); %number of parameters that
is locked
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Optimizing the parameters with the Kalman filter & MLE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Placeholders & Variable def.
global save_att save_vtt save_vt  save_dFtt_1 save_vFv save_Ptt_1 save_Ftt_1
save_Ptt
lnL_scores = zeros(3,1);
boundary = Inf;
% Running the estimation for The S&S 2 factor model and two benchmark
% models (The GBM model and the Ornstein-Uhlenbeck model).
for model = 1 % [1 = The S&S 2 factor model, 2 = The GBM modell, 3 = The
Ornstein-Uhlenbeck model.]
    if model == 1 % The S&S 2 factor model
        if sum(locked_parameters) == 0

            psi = zeros(7+N,1);
            psi(1:7,1) = [k, sigmax, lambdax, mu, sigmae, rnmu, pxe]';
            psi(8:end,1) = s_guess;

            lb = zeros(7+N,1);
            lb(1:7,1) = [0, 0, -boundary, -boundary, 0, -boundary, -1]';
            lb(8:end,1) = 0.0000001;

            ub = zeros(7+N,1);
            ub(1:7,1) = [boundary, boundary, boundary, boundary, boundary,
boundary, 1]';
            ub(8:end,1) = boundary;
        else
            psi = zeros(7+N-num_locked_parameters,1);
            psi(1:7,1) = [k, sigmax, lambdax, mu, sigmae, rnmu, pxe]';
            psi(8:end,1) = s_guess;

            lb = zeros(7+N-num_locked_parameters,1);
            lb(1:7,1) = [0, 0, -boundary, -boundary, 0, -boundary, -1]';
            lb(8:end,1) = 0.0000001;

            ub = zeros(7+N-num_locked_parameters,1);
            ub(1:7,1) = [boundary, boundary, boundary, boundary, boundary,
boundary, 1]';
            ub(8:end,1) = boundary;
        end
        a0 = initial_statevector;
        P0 = initial_dist;
    end

    % Running estimation
    options = optimset('Algorithm','interior-point','Display','off');
%interior-point active-set
    MaxlnL_Kalman = @(psi) Kalman_Estimation_Real(y, psi, matur, dt, a0, P0,
N, nobs, locked_parameters);
```

```
    [psi_optimized, log_L,exitflag,output,lambda,grad,hessian] =
fmincon(MaxlnL_Kalman, psi, [], [],[], [], lb, ub, [], options);
    % Saving estimation output
    lnL_scores(model,1) = -log_L;
    if model == 1
        ss_att = save_att;
        ss_vtt = save_vtt;
        ss_vt = save_vt;
        ss_dFtt_1 = save_dFtt_1;
        ss_vFv = save_vFv;
        ss_Ptt_1 = save_Ptt_1;
        ss_Ftt_1 = save_Ftt_1;
        ss_Ptt = save_Ptt;

        if sum(locked_parameters) == 0
             ss_psi_estimate =
[psi_optimized(1:7,1);sqrt(psi_optimized(8:end,1))];
             ss_SE = sqrt(diag(inv(hessian)));
        else
            prel_SE = sqrt(diag(inv(hessian)));
            prel_ss_psi_estimate =
zeros(size(psi,1)+size(locked_parameters,1),1);
            ss_SE = zeros(size(psi,1)+size(locked_parameters,1),1);
            j = 1;
            for i = 1:size(prel_ss_psi_estimate,1)
                 if all(abs(i-(locked_parameters+7))) == 1
                     prel_ss_psi_estimate(i,1) = psi_optimized(j,1);
                     ss_SE(i,1) = prel_SE(j,1);
                     j = j+1;
                 else
                     prel_ss_psi_estimate(i,1) = 0;
                     ss_SE(i,1) = 0;
                 end
            end
            ss_psi_estimate =
[prel_ss_psi_estimate(1:7,1);sqrt(prel_ss_psi_estimate(8:end,1))];
         end
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calculating/outputing key statistics
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Output
ss_psi_estimate
ss_SE

% S&S Model fit
ss_Mean_Error = mean(ss_vtt)'
ss_Std_of_Error = std(ss_vtt)'
ss_MAE = mean(abs(ss_vtt))'

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Outputing Graph
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(1);
set(figure(1), 'Position', [100 100 400 1000])
hold on
plot(exp(St),'k','linewidth',1);
```

```
plot(exp(ss_att(:,1)+ss_att(:,2)),'r','linewidth',1);
%plot(exp(ss_att(:,2)),'b','linewidth',1);
h = legend('Observed Price','Estimated Price');
title('Schwartz-Smith 2-factor model')
hold off
```

Bibliography

Works Cited

Schwartz, Eduardo, and James E. Smith. "Short-Term Variations and Long-Term Dynamics in Commodity Prices." Management Science, vol. 46, no. 7, 2000, pp. 893–911., doi:10.1287/mnsc.46.7.893.12034.

Cortazar, Gonzalo, et al. "Commodity Price Forecasts, Futures Prices and Pricing Models." 2016, doi:10.3386/w22991.

Lautier, Delphine. "The Kalman Filter in Finance: An Application to Term Structure Models of Commodity Prices and a Comparison between the Simple and the Extended Filters." Jan. 2002.

Delphine Lautier, and Alain Galli. "Simple and Extended Kalman Filters: an Application to Term Structures of Commodity Prices." Applied Financial Economics, vol. 14, no. 13, 2004, pp. 963–973., doi:10.1080/0960310042000233629.

Larsson, Karl, and Rikard Green. "Schwartz-Smith Two Factor Model in the Copper Market: Before and After the New Market Dynamics." Master Thesis of School of Economics, Lund University, May 2013.

Saha, Saikat. "Bayesian Calibration of the Schwartz-Smith Model Adapted to the Energy Market." 2014 IEEE Workshop on Statistical Signal Processing (SSP), 2014, doi:10.1109/ssp.2014.6884687.

Sick, Gordon, and Mark Cassano. "Forward Copper Price Models A Kalman Filter Analysis." 9 July 2012.

Websites Reviewed

Becker, Alex. "Online Kalman Filter Tutorial." Kalman Filter, kalmanfilter.net/.

Consulting, Goddard. Option Pricing - Finite Difference Methods,www.goddardconsulting.ca/control-system-design.html#Kalman.

Consulting, Goddard. "An Introduction to the Kalman Filter." Option Pricing - Finite Difference Methods, www.goddardconsulting.ca/kalman-filter.html.