Re-examination of Stock Trade Informativeness during the Financial Crisis:

An Empirical Study with High Frequency Data

Dalong Sun

Dissertation Supervisor:

Antonio Guarino

Dissertation submitted in part-fulfilment of the MSc in Economics

University College London

September 2014

Declaration

I, Dalong Sun, hereby declare that the work presented in this dissertation is my own original work. Where information has been derived from other sources, I confirm that this has been clearly and fully identified and acknowledged. No part of this dissertation contains material previously submitted to the examiners of this or any other university, or any material previously submitted for any other assessment.

Name: Dalong Sun

Date: 28th September, 2014

Classification

This piece of research is primarily:

□ an empirical/econometric study

Dissertation Library

I give permission for a pdf electronic copy of this dissertation to be made available to future students by way of the Dissertation Library, should the Department of Economics choose to use it in this way.

Yes

Abstract

Based on approaches developed by Hasbrouck[1] to measure asymmetric information from stock trades and quotes, the present work utilises a structural vector autoregressive (SVAR) system to model the interplay of high-frequency price and transaction series of stocks listed on the NASDAQ exchange with samples spanning from pre- to post-2008 financial crisis. Motivated by high frequency and the crisis-coverage features of the data, the current work attempts to re-evaluate, refine and extend empirical findings in Hasbrouck's studies. Results consistent with and results contrary to prior research are found in individual stock, cross-sectional and intraday analysis. Novel patterns not documented by previous studies are revealed in a cross-period analysis. In a cross-sectional analysis based on market value subsamples, trades for smaller firms are found to convey more private information in both an absolute and a relative sense. From an intraday perspective, it appears that trades have stronger impacts on asset prices at the beginning half-hour of trading not only in absolute but also in relative terms, refining the findings of previous research. An analysis of changes in trade informativeness from pre- to post- crisis periods reveals novel patterns that trades are more informative in the crisis week in absolute terms but less informative relative to the total amount of public information.

Keywords: Microstructure; Asymmetric Information; SVAR; Persistence; Financial Crisis

[1] Hasbrouck (1991a) (1991b).

Acknowledgements

Here, I intend to express the deepest appreciation to my supervisor, Professor Antonio Guarino, who is always willing to spend his precious time to offer me instrumental advice. Without his guidance and persistent help, this dissertation would not have been possible.

I would like to thank all my family and my friends for moral support, and also to all of those who supported me in any aspect over the time of completing this dissertation. Special thanks to Mr. Carl Baumgarten, a proprietary trader, who explained to me the actual market practices of trading activities and provided useful feedback on my thesis draft; my colleagues Dongwoo Kim, Liphang Poh, and Munaj Ahmed for their insightful suggestions on improving this work. I am also thankful to Dr. Lu Han at University of Cambridge and Mr. Xuan Wang at Bank of England for their extremely useful advice. All remaining errors are mine.

Contents

Figures and Tables

1. Introduction

The feature of informational efficiency of financial markets is claimed by the widely known efficient market hypothesis (EMH), a topic on which practitioners and researchers have been debating for long. Briefly, the hypothesis contains three forms: weak, semi-strong and strong.

i. In weak form efficiency, all the past information has been reflected in the current asset prices; hence, any efforts, such as technical analysis, that attempt to predict future asset prices and generate abnormal profits (on a risk adjusted basis) by finding historical price patterns are in vain.
ii. Semi-strong form efficiency claims that not only past but also current public information are instantly and completely absorbed in the current asset prices. Activities trying to obtain excess returns by analysing publically available information are to no avail.
iii. In strong form efficiency it is implied that all the information – both public and private – is priced in by the market and any attempt to outperform the market would be fruitless.

The message that EMH delivers is that no one, including the informed agent, is able to possess superior information and beat the rational market. More importantly, asset prices – and the associated bid and ask price quotes – adjust to newly released information promptly and fully and are not influenced by trades. Accordingly, no valuable information can be acquired by investigating trade-related data. However, in a typical real world situation markets are fraught with various frictions, imperfections and irrationalities, and market participants are heterogeneously informed regarding the value of assets and firms. The EMH fails in such circumstances.

As opposed to the EMH, Stiglitz and Grossman (1980) proposed a model in which the information in the market of informed agents does not fully reflect all information. As a consequence, individuals who invest time to uncover information not currently priced into the market can produce excess returns. By definition of the EMH this would not be possible. In this setup Stiglitz and Grossman have shown that competitive markets will collapse if efficient market hypothesis holds while information is not costless. As a result, costless information is not only a sufficient but also a necessary condition for all the information – both public and private – to be fully factored into asset prices. This is because competitive markets and price systems are of significance only if it is not costless to gather information. They concluded that there is a 'fundamental conflict' between the incentive to obtain information and the degree of efficiency in disseminating information by markets.

In reality, financial market participants usually have different levels of information in terms of firm's strategies, performance, opportunities, threats and financial health, etc., which determine or affect the value of a firm and its stock. In other words, agents are asymmetrically informed. For instance, comparing an institutional investor with an individual investor, the institutional investor may have more insights into a given firm's value and thus be able to make a better prediction on the firm's future stock price movements. This is because the institutional investor has more capital and human resources to invest in research on the firm. A veteran trader tends to be better informed than a beginner because the veteran has richer information sources, networks and experiences. Nevertheless, information as such are usually kept private by the information owner and not publically available. The full and timely reflection of that information by the market occurs rarely. On the one hand it is therefore imperative to develop methods hardly observed asymmetric information from variables observable via the market, such as security prices and transactions. On the other hand, it is crucial to examine how informational asymmetry will affect the level and formation process of asset prices.

Inspired by the pioneering work by Bagehot[2], who pointed out the key role played by the market maker[3] in the securities transactions, a large body of research in market microstructure devoted to analysis the role information asymmetry plays in forming asset prices. Important theoretical frameworks include those developed in Glosten and Milgrom (1985), Kyle (1985) and Easley and O'Hara (1987). Some empirical studies built

[2] Bagehot (1971).

[3] The market maker is the exchange specialist for listed securities and the over-the counter dealer for unlisted securities.

on those theories are Chiang and Venkatesh (1988), Glosten and Harris (1988), Hasbrouck (1988) and Hasbrouck (1991a,b)[4].

Based on approaches developed in Hasbrouck (1991a,b) to extract asymmetric information from stock trades and quotes, the present work utilises a structural vector autoregressive (SVAR) system to model the interplay of high-frequency price and transaction series of stocks listed on NASDAQ exchange. Availability of recent data which reflects evolving market fundamentals and advancement of transaction reporting technology enables a re-evaluation of previous research.

The contributions of this work are threefold. Firstly, a number of findings regarding stated in prior articles based on obsolete data sets are confirmed. Results from VAR estimation confirm the decent performance of SVAR system in measuring the impact of trades on asset prices[5]. Persistent and positive price impact of trades are captured in estimation for individual stocks. From an analysis based on market value subsamples, it is confirmed that trades from smaller firms deliver more private information. That trades contain more information at beginning of trading is also found in an intraday analysis.

Secondly, some counter and/or refining results have been reported. The impact of trade innovation[6] on prices is found to be a convex function of trade innovation size, compared to the concavity documented in the prior work[7]. This indicates an amplifying effect of trade size under current market practice. Quote changes tend to positively Granger-cause[8] trades for larger cap stocks, contrary to the negative causality in the past study. Besides, trades taking place in the first half-hour of trading are found to be more informative both in absolute terms and relative to total public information, but the prior paper claimed that trades are more informative only in an absolute sense.

Thirdly, novel findings are recorded from an analysis of cross-period[9] patterns. The two most pronounced results are: a. trades are more informative in the crisis week in

[4] More detailed discussion of literature is presented in the next section.

[5] Also known as the 'price impact of trades' or simply 'price impact'.

[6] As is stated in Hasbrouck (1991a), this is the 'unexpected component of the trade, relative to an expectation formed from linear projection on the trade and quote revision history'.

[7] Hasbrouck (1991b).

[8] Briefly, if variable x Granger-causes y, it indicates that past values of x help predict y.

[9] The 'cross-periods' in this case means time intervals span from 3 periods prior to 2008 financial crisis to 3 periods after the crisis week of 15-19 September 2008.

an absolute sense but less informative in relative terms; b. it seems that there is a lagged adjustment in the market's assessment of and its corresponding responses to the crisis.

This remaining part of my paper is organised as follows. Section 2 reviews studies that measure information content from stock trades and quotes. Section 3 reviews the theoretical models and vector autoregressive system. Section 4 elaborates data source and cleansing practice. Section 5 discusses empirical results and findings. Section 6 concludes.

2. Literature Review

The aim of this review is to highlight contributions relevant to the current work and may not be exhaustive. Research considering the impact of information asymmetries on the corporate issues such as capital structure and board structure, on the price formation and on the behaviour of market participants, are discussed.

The significance of information asymmetry

The pioneering work on the asymmetrical information was developed by Akerlof (1970), where the famous example of the (second hand) automobiles market was used to explain how the issues of adverse selection would arise with asymmetrically informed car sellers and buyers. He argued that bad quality cars tend to 'drive out' good quality ones. This results from the fact that car sellers prefer selling bad cars to buyers since good and bad cars sharing the same appearance are sold at identical prices to buyers who are less informed of the machine's quality than are the sellers. This shows how informational asymmetry leads to adverse selection problems, which may cause business to suffer.

Some recent research emphasises the importance of information at the corporate level. Bharath et al. (2009) employed measures of adverse selection developed by prior researchers to propose a new index of information asymmetry. They found that informational asymmetry has impacted corporate capital structures for U.S. firms between 1973 and 2002. More specifically, the proportion of a firm's debt financing is a positive function of the firm's adverse selection costs. Companies belonging to the highest adverse selection cost quartile have 30% more capital funded through debt financing than companies in the lowest adverse selection quartile. In addition, they claim that approaches to gauge information asymmetry in market microstructure literature are developed to investigate adverse selection problems between informed traders and the rest of the market (uninformed traders). Because firm's management can be regarded as a sub group of informed traders, who in turn are considered as sub group of all traders, measures designed in market microstructure studies can be utilised as a (imperfect) proxy for financial markets' assessment (or belief) on firms' adverse selection costs that result from superior information possessed by insiders. It is those adverse selection costs that have an ultimate impact on the firm's financing costs. Ferreira et al. (2011) proposed an adverse selection model to examine the relation between the informativeness of stock prices and the degree of independence of the corporate board. They concluded that stock price informativeness plays a role in the organisation's design. The more informative the stock price is, the lower the degree of corporate board independence, and the simpler and smaller the board structure required.

Measuring asymmetric information

With the central notion that agents are informed asymmetrically, a large body of research in the field of market microstructure is dedicated to study how the asset price is formed in the financial transactions and the influence of information asymmetries on the formation of asset price and the behaviour of different market participants, in particular traders and market makers.

Bagehot (1971) was the first to point out the key role played by the market maker[10] in the securities transactions. Market makers transact with two major groups of traders, namely the information-driven traders who possess superior non-public information about the future value of the asset; and the liquidity-driven traders who merely convert cash into securities or securities into cash. The key is that the market maker and liquidity-driven traders are considered as equally uninformed relative to information-driven traders; hence, it is claimed that the market maker always loses to the informed traders. It is further pointed out that the market maker must profit from the liquidity traders who are willing to incur costs for liquidity purposes to recoup losses to information traders so that the market maker can stay in business. As a result, the bid-ask spread set by the market maker is modelled as a trade-off between expected gains from liquidity traders and expected losses to informed traders, given that the spread is related to market liquidity and flow of new information.

Inspired by Bagehot's work, Copeland and Galai (1983) formalised the model in which a market maker aims to maximise his profit by optimally setting a bid-ask spread that

[10] The market maker is the exchange specialist for listed securities and the over-the counter dealer for unlisted securities.

maximises the expected total income gained from liquidity traders net of expected losses to the well informed. The spread is determined by the strike prices of the straddle – a combined position of a call and a put option with identical strike prices. They showed that the bid-ask spread is positively correlated to the level of asset prices and volatility of returns while negatively related to trading volume and the degree of competition, etc..

Building on previous studies, Glosten and Milgrom (1985) developed a formal asymmetric information and sequential trading model to demonstrate how adverse selection would give rise to a discrepancy between the bid and ask prices. They pointed out that the transaction price itself is a source of private information delivered by those well informed through their trading activities; as a consequence, the bid-ask spread tends to narrow as more trades are executed. Kyle (1985) proposed a sequential auction model to measure the information contained in security prices, the value of special information to an informed trader, and the liquidity features of a market. Kyle showed that the private information held by an insider trader is progressively reflected in asset prices; however, the informed trader is able to make profits constantly at the market maker's expense as long as the uninformed noise trader trades actively in the market, which provides 'camouflage' to the insider and obscures the information received by the market maker.

While the previous papers have conducted in depth research on the effect of asymmetric information on the bid and ask quotes, Easley and O'Hara (1987) turns to analyse how asset prices are affected by the trade size, and how trade size relates to the issues of informational asymmetry. They argued that informed traders are more willing to trade than the uninformed ones as those informed possess superior information that enables them to better estimate an asset's true value. Consequently, informed traders tend to trade in large blocks at any price level, leading to an adverse selection problem. Furthermore, since this behaviour is not shared by uninformed traders, the possibility that a market maker is facing an informed trader increases with the quantity of trades. This in turn has an impact on the market maker's pricing strategy in such a way that the price facing a large trader takes into account the rising likelihood of 'information-based trade'. It is concluded that trade size has an impact on asset prices, and information effects do play a role in this price-size relationship.

As is stated by Hasbrouck[11], different methods are designed to measure the market maker's exposure to information-driven traders, and those methods are motivated by two major empirical predictions. One prediction states that the spread between the bid and ask prices is positively related to information asymmetry; thus, approaches can be developed to extract asymmetric information from the readily observable bid and ask quotes. Another prediction claims that the magnitude of the impact of a trade on security price can be measured and interpreted as a proxy for information asymmetry.

Chiang and Venkatesh (1988) introduced the 'concentration of insider holdings' as a proxy for the extent of informational asymmetry. They aimed to examine whether insider holdings affect the bid-ask spread posted by the market maker in order to find whether the small-firm effect is caused by information asymmetries. Employing bid-ask spread as a summary measure of informational asymmetry, the study found that the information trading cost incurred by market makers is positively affected by insider holdings. Glosten and Harris (1988) also utilised information of bid and ask prices but they decomposed the bid-ask spread into two components – a transient component and an adverse selection component – and analysed those components in an asymmetric information model. They argued that the adverse selection component of the spread tends to have a permanent rather than transitory impact on stock prices. The adverse selection component was found to be positive and statistically significant.

Hasbrouck (1988) developed a framework which is capable of modelling asymmetric information and inventory control effects simultaneously, whereas most of the prior studies on either asymmetric information or inventory control models evolved along their distinct lines respectively. Having analysed the interaction of trades and quote revisions mainly with a moving average specification of 200 lags, Hasbrouck managed to find some differentiating features of asymmetric information and inventory control effects. Firstly, he found a persistent impact of trade size on quote changes for stocks with high transaction volume, a result consistent with Easley and O'Hara (1987) that

[11] In Hasbrouck (1991a).

large trades reveal more asymmetric information. More importantly, Hasbrouck points out the key difference between inventory control effects and information effects in their influence on prices. The market maker's inventory control activity tends to have an insignificant and transient impact on quote changes, while the asymmetric information effects of trades have a significantly positive and persistent impact on quote changes.

Build on the prior research, Hasbrouck (1991a) pioneered a vector autoregression (VAR) system which is capable of capturing the interdependency of trades and quotes. As a result, either trade or quote series can be modelled against its own past and the lagged values of the other variable. The VAR system developed by Hasbrouck is unusual in the sense that quote changes depend on not only the past but also current values of trade volumes, but current trades cannot be affected by present quote changes. Such a design has effectively mimicked operations both in many theoretical models and in the actual markets, in which transactions are taking place based on prevailing quotes – the quotes that set in the last period. Once a transaction is executed, market makers revise the quotes thereafter. Another key feature of the work is that trade innovation rather than the trade variable is used to examine the price impact of trades in order to avoid misleading inferences caused by inventory control effects and other transient effects resulting from non-information imperfections[12]. Furthermore, the work also attempts to capture nonlinearities in the impact of trades on quotes by employing a more realistic but still tractable model in which quote revisions are linear in nonlinear transformations of the trade volume variable. The persistent impact of trade innovation on quotes is found to be a positive, increasing and concave function of trade innovation size, and more significant informational asymmetry is associated with smaller firms. Pointing to drawbacks of measuring information content by stated spreads and estimated price impact of trades, Hasbrouck (1991b) proposed two new summary measures of trade informativeness based on techniques of random walk decomposition and variance decomposition. It extended the Hasbrouck (1991a) results by concluding that trades convey more information for small firms, both in absolute and relative terms. Moreover, an intraday analysis discovered that trades tend to be more informative at the beginning

[12] For example, price discreteness, price pressure effect, order fragmentation and exchange-mandated price smoothing.

half-hour of trading in absolute terms, but less informative relative to total public information.

De Jong et al (1996) studied the intraday price impact of trading activities on the Paris Bourse – an electronic order-driven market – with transactions data covering 44 trading days. Several decomposition approaches are used to break price effects into transient and persistent components. They have shown that persistent price impacts estimated by a VAR system are twice as large as the impacts estimated by a one period structural model, and the structural model tends to underestimate the effects of trades on price.

3. Methodology

3.1 Theoretical Model

The present work follows the approach developed in Hasbrouck (1991b) in which novel measures of trade informativeness based on the combined use of random walk decomposition and variance decomposition approach were proposed. Results from the decomposition method are combined with a vector autoregression (VAR) model of transaction data discussed in Hasbrouck (1991a) to form an empirical specification for estimation purposes. Both papers focus on modelling the effect of trade innovation – the unanticipated component of the trade – on quote revisions. The aim is to mitigate errors induced by inventory control effects and other non-information imperfections such as price discreteness, order fragmentation and exchange-imposed price smoothing.

Before reviewing the model, it is worth mentioning the indexing conventions employed in Hasbrouck (1991b). The sequence of time index *t* can be defined either as a set of wall-clock times or as a set of 'event counters'. In order to moderate the problem of non-stationarity in variables resulting from the intraday variation of returns, the 'event counter' approach is used to define the index *t*. Therefore, index *t* used throughout the present work stands for 'event time' rather than 'natural time'.

Notations and event sequencing conventions used in this work also follow those in Hasbrouck (1991b). Specifically, trade variable x_t denotes the signed trade volume (positive for buy order and negative for sell order) and q_t represents the price variable which is the midpoint of the prevailing bid quote q_{t-1}^{b} and ask quote q_{t-1}^{a} at the beginning of time *t*. After a trade x_t has been executed at time *t*, nontrade public information updates and the market maker sets new bid q_t^b and ask q_t^a quotes. This makes it explicit that both public nontrade information and private information extracted from trades are incorporated in quote revisions. It should be noted that, under this convention, the prevailing quotes at the beginning of time *t* before trades have been executed are those posted in the preceding period (q_{t-1}). The volume x_t is treated as zero if no trade takes place between any quote changes.

A symmetry assumption is imposed on the quote midpoint:

$$\text{As } s \to T, E[(q_s^b + q_s^a)/2 - \mathcal{L}_T|\Phi_t] \to 0 \tag{1}$$

Where $E[\mathcal{L}_\tau|\Phi_t]$ is the expected terminal (T) value of a security at some future time s conditional on the public information set Φ_t (including the past history of trades and quotes plus q_t and x_t). This implies that, as time elapses, the transitory departures of quote midpoints from efficient prices will disappear and quotes are expected by rational agents to revert back to the true value of the security. Moreover, Hasbrouck (1991b) defines price variables in logarithms to facilitate the cross-firm comparisons as proportional measures to remove the size effects across firms. This means that the quote revision is not specified as the simple difference of quote midpoint from period *t-1* to *t* used in Hasbrouck (1991a) but as the log difference:

$$r_t = \log(q_t^b + q_t^a) - \log(q_{t-1}^b + q_{t-1}^a) = \log(q_t) - \log(q_{t-1}) = \log\left(\frac{q_t}{q_{t-1}}\right)$$

It is clear at this point why Hasbrouck called r_t a proportional measure. Moreover, it can be conveniently interpreted as (log) returns.

The quote midpoint is treated as the sum of two unobservable variables:

$$q_t = m_t + s_t \tag{2}$$

Where m_t is the efficient price which follows a martingale process (de Jone and Rindi, 2009, p.164):

$$m_t = m_{t-1} + w_t \tag{3}$$

Where w_t has zero mean, constant variance σ_w^2, and zero autocovariance. The innovation term w_t incorporates refinement in the public information set which also contains the relevant most recent trade. Suggested in Hasbrouck (1991a), w_t can be further regarded as a sum of two types of innovations:

$$w_t = zv_{2,t} + v_{1,t} \tag{4}$$

Where $v_{1,t}$ and $v_{2,t}$ both have zero mean and no serial correlation, and are mutually uncorrelated. $v_{1,t}$ stems from public nontrade information, and $v_{2,t}$ is the trade innovation in which the private information hides. Hasbrouck (1991a) states that the coefficient z can be interpreted as a measure of information asymmetry in this simple setup.

Two summary measures of trade informativeness are proposed in Hasbrouck (1991b):

$$\mathrm{Var}(E[w_t|x_t - Ex_t|\Phi_{t-1}]]) = \mathrm{Var}(E^*[w_t|v_{2,t}]]) \equiv \sigma^2_{w,x}$$

and

$$\mathrm{Var}(E[w_t|x_t - Ex_t|\Phi_{t-1}]])/\mathrm{Var}(w_t) \equiv \sigma^2_{w,x}/\sigma^2_w \equiv R^2_w \tag{5}$$

Where $x_t - E[x_t|\Phi_{t-1}]$ is the current trade innovation, an unexpected component of the current trade x_t that can intuitively be interpreted as the private information signal. $E[w_t|x_t - Ex_t|\Phi_{t-1}]])$ is the expected effect of private information conveyed by the trade on the efficient price innovation w_t. As a result, $\sigma^2_{w,x}$ is the variance of trade correlated part of the efficient price changes, which measures trade informativeness in absolute terms. σ^2_w is the total variance of efficient price changes representing the total public information; hence, R^2_w, the ratio of $\sigma^2_{w,x}$ to σ^2_w, serves as a relative measure.

3.2 Econometric Model

A bivariate VAR system is suggested in Hasbrouck (1991a, b):

$$x_t = c_1 r_{t-1} + c_2 r_{t-2} + \cdots + d_1 x_{t-1} + d_2 x_{t-2} + \cdots + v_{2,t}$$

$$r_t = a_1 r_{t-1} + a_2 r_{t-2} + \cdots + b_0 x_t + b_1 x_{t-1} + \cdots + v_{1,t} \tag{6}$$

Where $r_t = \log(q_t) - \log(q_{t-1})$ is the (log) quote revision.

The model used in this work is a VAR(5) system with four variables,$\{x^0_t, x_t, x^2_t, r_t\}$, and five lags; where x^0_t is a signed trade indicator variable with +1 for purchase and -1 for sale, and $x^2_t = sign(x_t)|x_t|^2$ is a signed quadratic term which tries to capture the nonlinear relationship between trades and quotes. As suggested by Hasbrouck (1991b), the x_t term in (6) can thus be replaced by the column vector $[x^0_t, x_t, x^2_t]^T$. Consequently, b_i are 1-by-3 coefficient vectors and d_i are 3-by-3 coefficient matrices.

Unlike the ordinary VAR model, this particular VAR system allows a contemporaneous impact from x_t to r_t, but not the other way round. This precisely reflects the fact that quote is revised only after the trading has taken place, a convention that characterises most actual markets. The resulting contemporaneous relationship allows the previous

VAR model to be rewritten as a structural VAR (SVAR) system where the contemporaneous effects are captured in a structural matrix A_0:

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ A_{041} & A_{042} & A_{043} & 1 \end{bmatrix}}_{A_0} \begin{bmatrix} x_t^0 \\ x_t \\ x_t^2 \\ r_t \end{bmatrix} = \underbrace{\begin{bmatrix} a_{id,j} & a_{td,j} & a_{sq,j} & a_{r,j} \\ b_{id,j} & b_{td,j} & b_{sq,j} & b_{r,j} \\ c_{id,j} & c_{td,j} & c_{sq,j} & c_{r,j} \\ d_{id,j} & d_{td,j} & d_{sq,j} & d_{r,j} \end{bmatrix}}_{A_j} \begin{bmatrix} x_{t-j}^0 \\ x_{t-j} \\ x_{t-j}^2 \\ r_{t-j} \end{bmatrix} + \underbrace{\begin{bmatrix} v_{id,t} \\ v_{td,t} \\ v_{sq,t} \\ v_{r,t} \end{bmatrix}}_{V},$$

$$j = 1,2, \dots 5. \quad (7)$$

Where the column vector V contains structural shocks[13] (or innovations) of each variable. 'a' denotes all the coefficients associated with equation for x_t^0, 'b' denotes all the coefficients associated with equation for x_t, 'c' for equation x_t^2 and 'd' for equation r_t; the subscript *'id' (indicator)* means that the coefficient corresponds to lagged x_{t-j}^0 terms, *'td' (trade)* for lagged x_{t-j} terms, *'sq' (squared)* for lagged x_{t-j}^2 and *'r' (revision)* for lagged r_{t-j}.

As can be seen from (7) matrix A_0 captures the contemporaneous impact of all trade related variables while guaranteeing that current trade variables are not affected by current r_t but only lagged quote revisions r_{t-j}.

At the last step, the structural VAR model needs to be inverted to its corresponding vector moving average (VMA) representation. If variables $\{x_t^0, x_t, x_t^2, r_t\}$ are jointly covariance stationary, then the VAR system is invertible according to Wold's theorem. If this VMA is invertible the system can be inverted back to the corresponding VAR representation. Ordinary least squares can then be employed to produce consistent parameter estimates (Hasbrouck 1991b). The VMA representation is:

$$\begin{bmatrix} x_t^0 \\ x_t \\ x_t^2 \\ r_t \end{bmatrix} = \begin{bmatrix} e_{id,j} & e_{td,j} & e_{sq,j} & e_{r,j} \\ f_{id,j} & f_{td,j} & f_{sq,j} & f_{r,j} \\ g_{id,j} & g_{td,j} & g_{sq,j} & g_{r,j} \\ h_{id,j} & h_{td,j} & h_{sq,j} & h_{r,j} \end{bmatrix} \begin{bmatrix} v_{id,t-j} \\ v_{td,t-j} \\ v_{sq,t-j} \\ v_{r,t-j} \end{bmatrix}, \quad j = 0,1,2, \dots..30. \qquad (8)$$

The subscripts *id, td, sq* and *r* have identical interpretation as for equation (7). But it should be noted that *j* starts from zero rather than 1 which represents the current innovations. The lag length of 30 for VMA follows the practice in Hasbrouck's research,

[13] Also known as the disturbance terms of the VAR model.

which justifies that coefficients associated with higher lag terms are too small in value to offer any meaningful impact. Based on the formulae provided by Hasbrouck, $\sigma_{w,x}^2$ and σ_w^2 in the present study is calculated as:

$$\sigma_{w,x}^2 = \left(\sum_{j=0}^{30}\left[h_{id,j}, h_{td,j}, h_{sq,j}\right]\begin{bmatrix} v_{id,t-j} \\ v_{td,t-j} \\ v_{sq,t-j} \end{bmatrix}\right)\Omega_{v,trade}\left(\sum_{j=0}^{30}\left[h_{id,j}, h_{td,j}, h_{sq,j}\right]\begin{bmatrix} v_{id,t-j} \\ v_{td,t-j} \\ v_{sq,t-j} \end{bmatrix}\right)^T$$

and

(9)[14]

$$\sigma_w^2 = \left(\sum_{j=0}^{30}\left[h_{id,j}, h_{td,j}, h_{sq,j}\right]\begin{bmatrix} v_{id,t-j} \\ v_{td,t-j} \\ v_{sq,t-j} \end{bmatrix}\right)\Omega_{v,trade}\left(\sum_{j=0}^{30}\left[h_{id,j}, h_{td,j}, h_{sq,j}\right]\begin{bmatrix} v_{id,t-j} \\ v_{td,t-j} \\ v_{sq,t-j} \end{bmatrix}\right)^T + \left(1 + \sum_{j=1}^{30} h_{r,0}\right)\sigma_{v,revision}^2$$

Where $\sigma_{v,revision}^2$ is the variance of (log) quote revision, a scalar; $\Omega_{v,trade}$ is the 3-by-3 covariance matrix of innovations of three trade variables. Both $\sigma_{v,revision}^2$ and $\Omega_{v,trade}$ can be computed when the model is calibrated with data.

It is worth mentioning that $\sum_{j=0}^{30}\left[h_{id,j}, h_{td,j}, h_{sq,j}\right]\begin{bmatrix} v_{id,t-j} \\ v_{td,t-j} \\ v_{sq,t-j} \end{bmatrix}$ is the multiplier used to calculate the total persistent impact of a given initial trade shock on quote revisions; the term $\left(\sum_{j=0}^{30}\left[h_{id,j}, h_{td,j}, h_{sq,j}\right]\begin{bmatrix} v_{id,0} \\ v_{td,0} \\ v_{sq,0} \end{bmatrix}\right)$ is precisely the impulse response function used in Hasbrouck (1991a) to estimate the permanent effect of an initial trade on the price. The intermediate steps required from equation (7) to (8) are presented in the appendix B[15].

[14] (9) refers to both equations.

[15] A short notes on model identification: the Sims-Bernanke 'recursive identification' scheme is used here to achieve the point identifications of structural parameters $A_{041}, A_{042},, A_{043}$, in the structural matrix A$_0$ and diagonal elements in the covariance matrix of structural shock vector V in (7). The method starts with estimating the reduced-form VAR corresponding to the SVAR model. Next, the relationship between coefficients of reduced-form VAR and those of SVAR model is exploited. If we denote coefficient matrix of the reduce-form VAR as ϕ_j, then $\phi_j = A_0^{-1}A_j$;hence $A_j = A_0\phi_j$.To compute the (orthogonalised) impulse response of the SVAR model to structural shocks, simply post multiply the VMA coefficients of the reduced-form VAR by A_0^{-1}, the inverse of the structural matrix. Applying Sims-Bernanke's method

4. Data Analysis

The results reported in this study are based on a sample of high frequency trading (HFT) data of stock quotes and trades recorded with a one-millisecond time resolution, covering the period from the beginning of 2008 to the end of 2009. However, the data does not cover all consecutive trading days and every month from 2008 to 2009. For each year, data of a full trading week (5 days) of the first month of each quarter is used based on the data availability. Only trading time data is used. More specifically, the data utilised covers 07-11 January, 07-11 April, 07-11 July and 15-19 September of the year 2008, and covers 13-17 April, 06-10 July and 05-09 October of the year 2009. The sample based on period 07-11 January 2008 is used for individual stock analysis (section 5.2), cross-sectional analysis (section 5.3) and intraday analysis (section 5.4). Results based on all seven sample periods will be discussed in the cross-period analysis (section 5.5).

A sample of 15 stocks listed on NASDAQ exchange is randomly selected from the database which contains 120 stocks. A stock picking program is created to facilitate the selection work based on random number generators. Stocks are categorised into market capitalisation groups of large, medium and small. For each group 5 stocks were selected that fit the relevant market capitalisation criteria. The abbreviations 'large cap', 'medium cap' and 'small cap' are used in the remaining work. A table summarising the name of each company, its associated stock ticker symbol and its corresponding market value group is provided in appendix A1.

4.1 Data Source

More specifically, the HTF database contains the following data fields for trade records:

1. Timestamp, expressed in milliseconds from midnight. For example, 9.30 is reported as 34200000;
2. Stock ticker symbol, which is an abbreviation of the company name and serves as an identifier for the listed company. For example, ticker symbol 'AMZN' is assigned to Amazon and 'GOOG' is assigned to Google;

leads to a system of 16 equations 7 unknowns to solve for A_0. Therefore, point identification is achieved. Detailed steps of solving for A_0 are provided in appendix B.

3. Shares, in a unit of 1 share; odd lots is also reported[16];
4. Buy/sell indicator; and
5. Transaction price.

Each trade record also contains information about the type of any liquidity provider or liquidity seeker, e.g. whether it is a HFT firm or not. This work is performed based on the exchange's knowledge of its clients; however, the information is not used as it is beyond the scope of the present study.

This work also uses quotes records of the following data fields contained in the database:

1. Timestamp, expressed in military time format to the millisecond. For instance, 9.30 is displayed as 093000000 and 15.52 would be 155200000;
2. Stock ticker symbol;
3. NASDAQ best bid quotes with 100 or more shares; and
4. NASDAQ best ask quotes with 100 or more shares.

Comparison with data in Hasbrouck (1991b) and motivation

Thanks to the advancement of computer hardware and information technology, the data employed in the current study is different both quantitatively and qualitatively from the data used by Hasbrouck. From the quantitative perspective, Hasbrouck's data covered covers 62 trading days in one-second time resolution with the total number of observations for each stock less than 5000 based on the total sample average. The data used in the this study contains on average more than 70000 observations for each 5 five trading days. Qualitatively, the data in this work is reported with a 2-decimal-place precision and spans from the pre financial crisis period to the post financial crisis period; while data in Hasbrouck's paper was reported with a 1/8 or ¼ point increment and covered a non-crisis period. Accordingly, two points can be made. Firstly, the non-information imperfection concern resulting from price discreteness mentioned by Hasbrouck is largely mitigated by using the new database. Secondly, an opportunity

[16] All the estimation results present in the rest of the work use a 100-share round lot unit for trade volumes. Trade volumes reported in mixed lots (e.g. 136 shares) are rounded to the unit of 100-share round lots.

arises for extending the previous research's coverage to cross period analysis. Since the database contains trades and quotes data of the crisis week of 15-19 September 2008 (known as 'crisis period' for conciseness), a 7-period analysis has been performed in the attempt to reveal some cross-period patterns of trade informativeness, with 3 periods ahead of the crisis period and 3 periods after the crisis period.

4.2 Data Cleansing

The progression of quotes and trades are sequenced based on the 'event timing' convention described by Hasbrouck (1991b). As a result, the index *t* serves as an 'event counter' instead of a natural time index. Put in another way, *'t'* represents transaction or event time. The first event is defined as the first quote and trade for the first trading day of the full trading week. The index *'t'* is increased by one each time a new quote or trade is recorded: firstly, the current database is relatively clean thanks to the automated trading system. Information about trades and quotes is reported in the correct sequence and 'reporting anomalies' is no longer a concern as no manual work is involved in the reporting process of quote revisions and transactions. Secondly, the time stamps gives the exact time of a quote revision or transaction. As a result, concerns relating to delayed reporting of transaction and the trade-quote re-sequencing are lessened here.

Nevertheless, the large volume of high frequency data also gives rise to some new problems. Firstly, in order to alleviate the issue of order fragmentation, trades that occurred at the same time and with the same direction (buy/sell) are combined as a single transaction. In the high frequency datasets, simultaneous trades occur more frequently than in the data set used in previous studies. Secondly, the amount of quotes data is on average 5-10 times larger than the trades data, e.g. more than 1000 quotes can be posted for some liquid stocks. But a large number of adjacent quotes are merely duplicates of each other. In this case, only the first quote is retained and other duplicates are removed in the sense that those duplicated quotes are not the quote changes used as the model inputs and only add much noise to the series. However, the trades combination and quote duplicates removal works are in turn complicated by differences in timestamp conventions. As is discussed in the previous section, the time stamps used in the trades report is documented in 'milliseconds-from-midnight' form whereas time stamps in the quotes report apply 'military time format'. In a timestamp format-matching practice, all the leading zeros in military time format are deleted, and 'milliseconds-from-midnight' timestamps are converted to military time format using a self-defined Matlab function[17].

An issue with determining the trade direction – using buy/sell indicator data vs. direction approximation (DA) approach

It is worth noting that the high frequency database contains information that allows the user to have an immediate judgement on the direction of the transaction, i.e. whether it is a purchase or a sale. More specifically, if the buy/sell indicator reports a 'B' which stands for 'buy', I add a plus sign '+' to trade volume x_t, and a minus sign '−' to x_t if an 'S' is reported.

Nonetheless, the direction approximation (DA) approach described in Hasbrouck (1991b) is still applied to estimate whether a recorded trade is a buy or a sell. In this approach, a trade is assumed to be a buy order if transaction price is higher than the prevailing quote midpoint and a positive sign'+' is assigned to the trade. A negative sign '−' is assigned when the transaction price is lower than the quote midpoint. When the transaction price equals the quote midpoint the direction is indeterminate and x_t is set to zero. The resulting signed series is compared with the signed trade series produced by real time buy/sell indicator data. The real time data-generated series can then evaluate the performance of DA-generated signed volumes.

4.3 Summary Results

Figure 1 below shows the 5-trading-day performance evaluation results of the direction approximation (DA) method used by Hasbrouck using Google stock (ticker symbol 'GOOG'), which serves as one of the three representative stocks for individual stock analysis in section 5.2. As mentioned above, results are based on the sample period 07-11 January 2008. 'PCOR' is the percent of corrected estimated signs in the sense that the estimation is accurate if the approximated sign matches with the real buy/sell indicator. 'PSIGN' indicates the percent of non-zero estimated signs. 'PB'/'PS' stands for

[17] Matlab codes are provided in the appendix.

percentage of buy/sell order respectively, with the blue line for the approximated results and red line for real data results. Overall, the DA approach has a decent performance over the full trading week sample. The correction rates are above 85% across all the trading days despite small fluctuations. The approximated buy/sell order proportion also follows real data closely. Results for another representative stock Modine Manufacturing (MOD) are shown in appendix A2. The results are similar to those for Google.

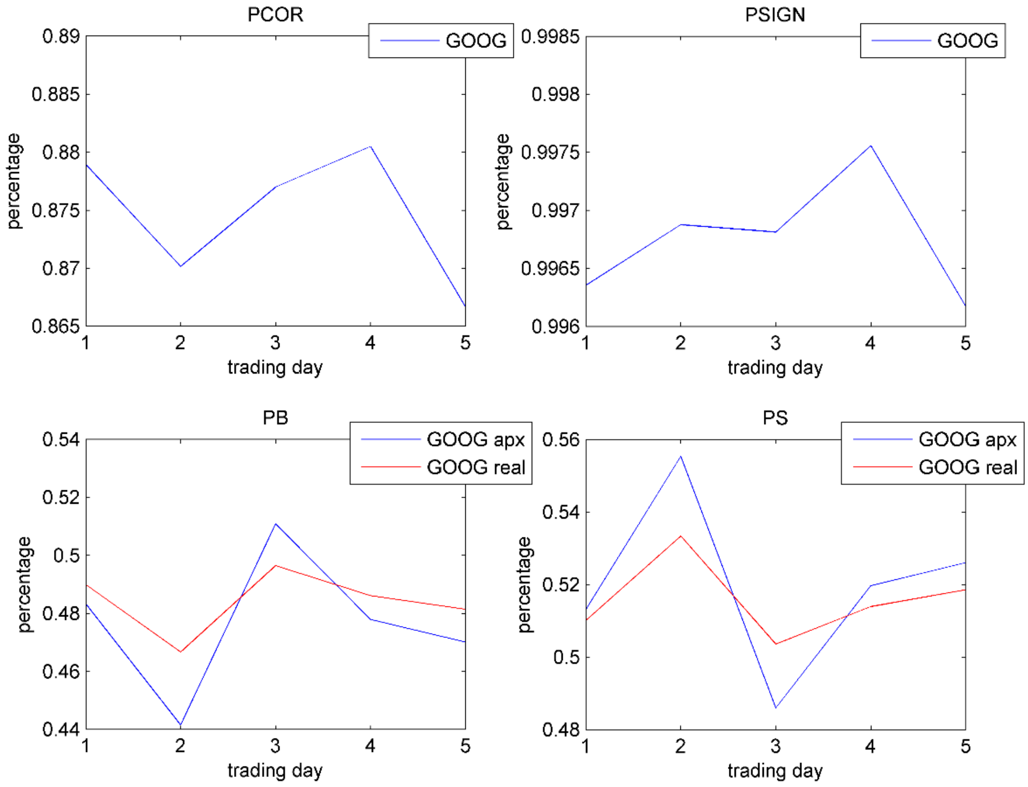


Figure 1 Performance Evaluation of DA Approach - Google

Appendix A3 displays the number of observations for trades and quotes before and after cleaning the data. A few points are worth mentioning:

1. Stocks with larger market capitalisation are traded more frequently;
2. A large discrepancy exists between number of trades done and of quotes posted; and
3. Discrepancies narrow substantially after cleaning the data due to the large number of quote duplicates.

5. Empirical Analysis and Results

5.1 Stationarity Tests and Descriptive Statistics

a. Stationarity Tests

Before putting time series data into the VAR model it is crucial to test for the stationarity of variable inputs. Non-stationary variables will cause the model to be erroneous[18] and results produced tend to bear little meaning. Following a 'confirmatory data analysis' approach (Brooks, 2008), three tests – namely Augmented Dicker Fuller (ADF) test, Philips and Perron (PP) test and Kwiatkowski, Philips, Schmidt and Shin (KPSS) test – are applied to examine the stationarity of trade and quote series.

The three stationarity tests are applied to both raw and cleansed series to compare their different time series properties. Test results are displayed in appendix A4 to conserve space. It can be seen from appendix A4.a and A4.b that the trade series is stationary both before and after data cleansing. However, the raw quote midpoints are not stationary while the cleansed (log) quote revisions are stationary. This also explains why quote revisions rather than midpoint prices are modelled.

b. Descriptive Statistics

The descriptive statistics for cleansed data of 15 stocks are presented in table 1. Companies have been categorised by their market values. In terms of trade series, the mean value is close to zero for all stocks due to the sign-adding practice. Nonetheless, adding positive/negative signs to buy/sell orders lets one observe the largest respective purchase and sale, which is not possible in the raw data series. Standard deviations and the largest buy/sell orders of large cap stocks are in general higher than those of medium and small cap stocks, but this is not necessarily true for medium cap relative to small cap stocks. The excess kurtosis[19] figures are high across firms, suggesting that trades data have fat tails and are not normally distributed, which is confirmed by the Jarque-Bera (JB) normality test. P-values of JB tests are smaller than 0.01; therefore we reject the null of a normal distribution.

[18] For example, spurious regression problem may be found if non-stationary data is used.

[19] Excess kurtosis = kurtosis – 3. Non-zero excess kurtosis indicate departure from normality.

Considering the quote revision (or log return) series, it can be seen that the mean return is of negligible size. However, it is not meaningless since those figures represent milliseconds returns which will be sizable if converted to annualised returns. The figures also offer a taste of the level of returns firms tend to generate by trading in this high frequency. Standard deviations have different patterns across market value subsamples. Returns for larger value stocks tend to be more stable as the standard deviation of return increases as the market capitalisation of stocks decrease. For small cap stocks, returns are fairly volatile relative to their means. The negative skewness across most firms indicates that negative returns are experienced more often than the positive ones, which is a common property of stock returns. The common large excess kurtosis figures suggest a heavy tail distribution for return series, meaning that large gains and losses are more likely to happen than indicated by a normal distribution. The p-value of JB tests point to the non-normality of stock return distributions.

Table 1 Descriptive Statistics for CLEANSED Data 07 – 11 January 2008

	LARGE CAP									
	Trades (Clean)					Quote Revisions (Clean)				
	AMZN	AMAT	DIS	GPS	GOOG	AMZN	AMAT	DIS	GPS	GOOG
Mean	0.45	4.40	16.80	-4.46	-0.10	-0.00003%	-0.00008%	-0.00006%	-0.00027%	0.00000%
Std. Dev	342.07	2,006.69	540.21	612.76	58.49	0.015%	0.034%	0.026%	0.047%	0.024%
Skewness	-85.00	0.65	-0.71	0.05	-2.87	-5.92	-2.04	1.30	-35.94	-0.07
Ex Kurtosis	19,916.09	165.68	62.05	111.23	807.02	554.41	74.64	145.59	4,331.73	8.30
Maximum	15,033	50,000	9,300	19,100	4,100	0.003	0.004	0.011	0.013	0.008
Minimum	-88,526	-78,170	-17,500	-20,918	-9,096	-0.015	-0.013	-0.007	-0.056	-0.012
P-Value (JB)	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
	MEDIUM CAP									
	Trades (Clean)					Quote Revisions (Clean)				
	CETV	FCN	LSTR	LPNT	CKH	CETV	FCN	LSTR	LPNT	CKH
Mean	0.68	0.13	0.17	-3.17	-0.26	-0.00003%	-0.00005%	0.00000%	-0.00021%	-0.00006%
Std. Dev	64.15	45.86	124.66	168.11	30.22	0.075%	0.087%	0.043%	0.030%	0.071%
Skewness	-14.97	9.35	9.01	0.08	-19.28	0.26	-0.35	0.01	-0.84	-0.45
Ex Kurtosis	1,303.87	982.44	576.44	34.26	1,897.90	54.57	275.22	36.86	91.25	102.39
Maximum	2,755	4,297	8,262	2,900	931	0.014	0.047	0.011	0.008	0.018
Minimum	-5,000	-1,900	-2,400	-2,800	-2,800	-0.013	-0.052	-0.009	-0.010	-0.018
P-Value (JB)	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
	SMALL CAP									
	Trades (Clean)					Quote Revisions (Clean)				
	BW	CDR	MOD	NXTM	PBH	BW	CDR	MOD	NXTM	PBH
Mean	1.01	7.48	0.70	-4.74	5.20	-0.00018%	-0.00024%	-0.00062%	0.00042%	-0.00040%
Std. Dev	42.29	144.24	73.04	200.74	162.80	0.152%	0.219%	0.279%	0.100%	0.325%
Skewness	35.83	3.08	3.18	11.16	3.48	-0.06	-1.07	-0.26	0.19	-0.16
Ex Kurtosis	3,618.85	71.73	158.01	854.24	58.71	10.63	54.17	23.73	13.82	23.17
Maximum	5,000	3,200	2,500	10,500	3,263	0.018	0.030	0.030	0.012	0.035
Minimum	-1,500	-2,000	-1,500	-6,849	-1,700	-0.018	-0.052	-0.047	-0.012	-0.041
P-Value (JB)	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001

5.2 SVAR and VMA Models for Two Representative Stocks

This section presents estimation results of the four-variable SVAR(5) model of $\{x_t^0, x_t, x_t^2, r_t\}$ for individual stock analysis. Two representative stocks are selected: Google (GOOG, representing large cap group) and Modine Manufacturing (MOD, representing small cap group). The purpose of choosing stocks from different market value groups is to form a comparative analysis. Stocks in medium cap group is not chosen since results of medium cap stocks resemble those of small cap stocks. More comprehensive cross-firm analysis is presented in the next section.

Moreover, both individual equation OLS regression and pooled OLS regression are performed. The pooled regression is designed to remedy high correlations among innovations of three trade variables. It turns out that the two approaches yield identical coefficient estimates. This confirms what is suggested in Hasbrouck (1991a) that OLS can consistently estimate the VAR systems if trade innovation and quote innovation have zero means, and are serially and mutually uncorrelated. The large sample size employed in current work also implies that coefficient estimates are close to true ones.

Table 2 shows coefficient estimates of the SVAR(5) model. The adjusted R^2 ($\bar{R}^2$) for r_t equation is 0.783[20]. This implies the superior performance of the four-variable structural VAR model, and a substantially lower $\bar{R}^2$ is expected if a bivariate VAR of $\{x_t^0, r_t\}$ is used, according to Hasbrouck (1991b)[21]. Three lines of important findings are worth elaborating.

Firstly, in the r_t equation, only 5 components are insignificant: the fourth lag of x_t^0, the second lag of x_t and the third to fifth lags of x_t^2. The most important set of coefficients, as suggested by Hasbrouck (1991a), are those of $x_t^0, \dots, x_{t-5}^0$. The coefficient of x_t^0, $d_{id,0}$, indicates that the log quote midpoint rises on average by 4.75*10^{-5} (0.00475%) immediately after execution of a buy order. At longer lags, coefficients are in general positive but decreasing in values. For x_t terms, the coefficient sum over five lags is positive. Besides, negative serial correlation in the quote revisions are implied by

[20] This figure for the small cap representative stock is 0.459, also a decent level.

[21] As per Hasbrouck (1991b), the four variable SVAR is capable of attributing considerably more of the efficient price variance to trades than if the simple bivariate SVAR is used.

negative lagged r_t coefficients $d_{r,1}$ to $d_{r,5}$. These three phenomena are consistent with Hasbrouck (1991a). Surprisingly, coefficients of x_t^2, x_{t-1}^2, x_{t-2}^2 are found to be significant but positive, suggesting that the impact of trades on prices is a convex function of trade size. These findings are opposite to those stated in Hasbrouck (1991a), in which negative coefficients for x_t^2 terms are found and a concave price impact is claimed. The convexity indicates an amplifying effect of trade size.

Secondly, in the x_t^0 equation, a strong positive autocorrelation in trades is found based on the positive lagged coefficients of x_t^0 terms, which means that there are no trade reversals at the first five lags. This result is inconsistent with the well documented inventory control mechanism but is in line with findings of Hasbrouck and Ho (1987) and Hasbrouck (1988) that buy orders is likely to be followed by buy orders and similarly for sell orders. The predominant short-run positive autocorrelation in trades is better explained by delayed adjustments to new information (Hasbrouck 1991a). In addition, significant lagged r_t coefficients imply that (log) quote revisions Granger-cause trades; therefore, past quote revisions help predict trades. However Granger causality is positive in this case, in contrast to negative causality reported by Hasbrouck. Results in the x_t equation share close pattern with results in the x_t^0 estimation.

Thirdly, compared to statistical insignificance of x_t^2 terms in x_t^0 and x_t^2 equations pointed out in Hasbrouck (1991a), the lagged x_t^2 coefficients are significant and negative in both x_t^0 and x_t^2 equations, which implies Granger causality running from the signed squared trade variable to both signed trade indicator variable x_t^0 and trade variable x_t. Additionally, in the x_t^2 equations, significant positive Granger causality is found from x_t^0 and x_t to x_t^2 based on positive lagged x_t^0 and lagged x_t coefficients, contrasting Hasbrouck's claim that x_t^0 and x_t are insignificant linear predictors of x_t^2. Besides, negative autocorrelation in x_t^2 is reported.

Estimation outputs for the small cap stock is presented in appendix A5. Strikingly, the estimation results of small cap stocks are different substantially from those of large cap stocks. Firstly, in the r_t equation, only x_t^0, x_{t-4}^0, x_{t-5}^0 are found to be significant. Although coefficient of x_t^0 is still positive, negative price impact of trade is suggested by negative coefficients of x_{t-4}^0 and x_{t-5}^0. All x_t and x_t^2 and insignificant. Secondly, Granger causality from quote revisions to trades is negative for the small cap stocks, contrary to the case of large-cap estimation. Thirdly, less significant Granger causality exists from x_t^0 and x_t to x_t^2, and x_t^2 terms are insignificant in x_t^0 and x_t^2 equations. Interestingly, results from small cap stocks coincide more with Hasbrouck's findings[22].[23]

[22] However, in the estimation for small cap representative stock, the impact of trades on prices is still a convex function of trade size.

[23] Coefficient estimates for the vector moving average (VMA) models inverted from SVAR are available upon request.

Table 2 Estimates of the multivariate SVAR(5) for GOOGLE (GOOG), 07-11 January 2008

x^0_t	Coeff.	x_t	Coeff.	x^2_t	Coeff.	r_t	Coeff.
						$\mathbf{d_{id,0}}$	4.75e-05***
							79.87
$a_{id,1}$	0.235***	$b_{id,1}$	0.322***	$c_{id,1}$	0.381***	$d_{id,1}$	3.04e-05***
	182.83		104.01		7.2		50.41
$a_{id,2}$	0.118***	$b_{id,2}$	0.156***	$c_{id,2}$	0.313***	$d_{id,2}$	4.65e-06***
	90.76		49.84		5.85		7.7
$a_{id,3}$	0.0543***	$b_{id,3}$	0.0604***	$c_{id,3}$	-0.334***	$d_{id,3}$	1.07e-05***
	41.64		19.2		6.22		17.75
$a_{id,4}$	0.0462***	$b_{id,4}$	0.062***	$c_{id,4}$	0.119**	$d_{id,4}$	3.11E-07
	35.52		19.78		2.22		0.52
$a_{id,5}$	0.0365***	$b_{id,5}$	0.0454***	$c_{id,5}$	0.12**	$d_{id,5}$	3.46e-06***
	28.41		14.65		2.27		5.8
						$\mathbf{d_{td,0}}$	1.38e-05***
							46.53
$a_{td,1}$	0.00114*	$b_{td,1}$	0.0593***	$c_{td,1}$	0.686***	$d_{td,1}$	-1.41e-06***
	1.78		38.41		26.04		4.74
$a_{td,2}$	-0.00164***	$b_{td,2}$	0.0243***	$c_{td,2}$	0.214***	$d_{td,2}$	-2.44E-07
	2.55		15.71		8.1		0.82
$a_{td,3}$	0.000497	$b_{td,3}$	0.0212***	$c_{td,3}$	0.383***	$d_{td,3}$	8.63e-07***
	0.78		13.69		14.5		2.9
$a_{td,4}$	0.000948	$b_{td,4}$	0.0131***	$c_{td,4}$	0.0975***	$d_{td,4}$	-6.84e-07**
	1.48		8.45		3.7		2.3
$a_{td,5}$	-0.00188***	$b_{td,5}$	0.00718***	$c_{td,5}$	0.154***	$d_{td,5}$	1.5e-06***
	2.94		4.65		5.84		5.04
						$\mathbf{d_{sq,0}}$	1.66e-07***
							13.06
$a_{sq,1}$	-0.000155***	$b_{sq,1}$	-0.00122***	$c_{sq,1}$	-0.00856***	$d_{sq,1}$	5.46e-08***
	5.63		18.36		7.56		4.29
$a_{sq,2}$	-6.39e-05**	$b_{sq,2}$	-0.000451***	$c_{sq,2}$	-0.00291***	$d_{sq,2}$	2.61e-08**
	2.32		6.8		2.57		2.05
$a_{sq,3}$	-1.37E-05	$b_{sq,3}$	-0.000226***	$c_{sq,3}$	0.00524***	$d_{sq,3}$	-9.34E-09
	0.5		3.4		4.63		0.73
$a_{sq,4}$	-6.21e-05**	$b_{sq,4}$	-0.000207***	$c_{sq,4}$	-0.00102	$d_{sq,4}$	1.93E-08
	2.26		3.13		0.9		1.51
$a_{sq,5}$	-4.59E-06	$b_{sq,5}$	-0.00016***	$c_{sq,5}$	-0.00287***	$d_{sq,5}$	-1.75E-09
	0.17		2.41		2.53		0.14
$a_{r,1}$	5.41***	$b_{r,1}$	-3.36	$c_{r,1}$	-42.2	$d_{r,1}$	-0.852***
	3		0.77		0.57		1019.75
$a_{r,2}$	57.7***	$b_{r,2}$	86.3***	$c_{r,2}$	236***	$d_{r,2}$	-0.189***
	24.39		15.11		2.42		172.15
$a_{r,3}$	62.8***	$b_{r,3}$	94.7***	$c_{r,3}$	278***	$d_{r,3}$	-0.209***
	26.57		16.62		2.86		191
$a_{r,4}$	51.5***	$b_{r,4}$	74.6***	$c_{r,4}$	192**	$d_{r,4}$	-0.0249***
	21.75		13.06		1.97		22.66
$a_{r,5}$	33.8***	$b_{r,5}$	57.3***	$c_{r,5}$	164**	$d_{r,5}$	-0.0678***
	18.79		13.2		2.22		81.11
$\bar{R}^2$	0.118	$\bar{R}^2$	0.066	$\bar{R}^2$	0.004	$\bar{R}^2$	0.783
$\hat{\sigma}^2_{id}$	0.056	$\hat{\sigma}^2_{td}$	0.324	$\hat{\sigma}^2_{sq}$	94.421	$\hat{\sigma}^2_{r}$	1.20E-08

*/**/*** denotes that the figure is significantly different from 0 under 10%/5% /1% significance level

5.3 Cross-sectional Analysis

The four-variable SVAR(5) model describe in Section 3 was estimated for each of 15 sample stocks. Results displayed in Table 3 are summary statistics for total sample and for each market value group. For convenience in presentation, statistics reported are sample averages instead of figures for individual stocks. All standard deviations are computed both on a per-event and on a per-hour basis. For each individual stock, the conversion of per-event standard deviations to per-hour ones can be achieved by: $\sigma/hour = (\sigma/event \times sample\ size)/(5\ days \times 6.5\ trading\ hours/day)$.

The 'spread' variable is known as the 'average proportional spread' and is calculated as a time weighted average of all intraday bid-ask spreads as suggested in Hasbrouck (1991b). On a total sample basis, the spread on average is 0.226 percent of the security price. This variable increases markedly with decreasing market capitalisation, which is in accordance with numerous prior studies. At first sight, this increasing pattern can be regarded as a tentative evidence that issues of information asymmetry are more severe – thus trading is more informative – for smaller firms. But such an interpretation may be inaccurate resulting from the fact that the adverse selection component of the spread is positively affected by the intensity of public information, the total return volatility (Hasbrouck, 1991b).

It is also tempting to use the cumulative impulse response (shown as 'Impulse' in table 3) to summarise the cross sectional pattern of trade informativeness. 'Impulse' represents the persistent impact of a trade innovation on the stock price given an initial trade shock, which is interpreted as the information content of a trade innovation net of transitory impact caused by non-information imperfections in Hasbrouck (1991a). To compute this figure, an initial trade shock of a 1000[24] share buy order is introduced in the system[25]. The total sample average impulse of 0.085 implies that a 1000 share raises the quote midpoint by around 0.085%. This number increases monotonically across

[24] 1000 share is chosen as a customary order size across all sample stocks; moreover, as discussed in section 4, the unit used for estimation work is a unit of 100-share round lot. As a result, 1000 share is expressed in 10 100-share round lots, i.e.,$x_t^0 = 1$, ., $x_t = 10$, ., $x_t^2 = 100$.

[25] More details in computing the impulse response function can be found in Hasbrouck (1991b, p.587).

market value subsamples (from left to right), meaning that trades of the same size contain more information for low-value firms, consistent with prior studies.

The problem with interpreting impulse figures as summary measures of trade's information content is that stocks of larger value firms are traded more frequently, which is manifested in different sample sizes across market value groups and in $\sigma_x/hour$ figures. $\sigma_x/hour$ measures trading intensity, and it is indeed considerably higher for large cap stocks.

Unlike in previous research, both $\sigma_w/hour$ and $\sigma_{w.x}/hour$ dysfunction in the current scenario. It was expected that both variables are higher for smaller firms, but current work offers exactly the opposite patterns. The answer to this unanticipated behaviour becomes clear after an investigation of the data: high frequency transactions occur much more frequently for large cap stocks relative to medium and small cap stocks, and sample size appears in the numerator of the formula for the 'per-event-to-per-hour' conversion. As is discussed in Section 3, variables defined on the per-event basis have much more stable behaviours; hence, I turn to use $\sigma_w/event$ and $\sigma_{w.x}/event$ to measure the trade informativeness for the high frequency data. σ_w stands for the standard deviation of change in the efficient price and serves as a proxy for total public information. $\sigma_{w.x}$ is the volatility of trade-correlated part of the efficient price changes and captures the portion of efficient price movements attributed to all trading activity in an absolute sense.

It can be seen from table 3 that both $\sigma_w/event$ and $\sigma_{w.x}/event$ increase with decreasing market capitalisations. $\sigma_w/event$ is 0.023% for large cap subsample and 0.078% for small cap subsample, which means that smaller companies have higher return volatilities. $\sigma_{w.x}/event$ is 0.004% for large firms and 0.023% for small firms. Furthermore, R_w^2 – the proportion private information conveyed by trades relative to total public information – is also higher for smaller firms[26]. These results confirm that trades of smaller firms tend to deliver more asymmetric information in both absolute and relative senses, in line with Hasbrouck (1991b).

[26] In this case, it is better to treat the medium and small cap firms as a group relative to the large cap firms.

Table 3 Cross-sectional summary statistics for 15 selected stocks, 07-11 January 2008

Cross-sectional summary statistics for 15 selected stocks, 07-11 January 2008

		Market value subsamples		
	Total	LARGE	MEDIUM	SMALL
	Sample	CAP	CAP	CAP
No. of firms	15	5	5	5
No. of obs	150584	369008	57008	25736
	[358153]	[597759]	[28152]	[27969]
Impulse (x100)	0.085	0.01	0.081	0.165
	[0.075]	[0.007]	[0.026]	[0.061]
σ_x /event	3.034	7.026	0.85	1.226
	[4.956]	[7.434]	[0.569]	[0.649]
σ_x/hour	6594.44	17901.546	1249.496	632.279
	[9841.262]	[9929.477]	[596.507]	[261.079]
σ_w/event (x100)	0.043	0.023	0.027	0.078
	[0.032]	[0.015]	[0.003]	[0.030]
σ_w/hour (x100)	66.175	100.724	48.761	49.04
	[61.945]	[95.191]	[28.796]	[36.079]
$\sigma_{w,x}$ /event (x100)	0.012	0.004	0.01	0.023
	[0.009]	[0.002]	[0.004]	[0.006]
$\sigma_{w,x}$ /hour (x100)	23.147	35.955	17.971	15.513
	[27.180]	[44.810]	[10.750]	[12.280]
R^2_w	0.117	0.093	0.153	0.106
	[0.092]	[0.124]	[0.095]	[0.054]
Spread (x100)	0.226	0.068	0.198	0.412
	[0.171]	[0.019]	[0.086]	[0.139]

Note: 1. Figures are average values for the total sample of 15 firms and subsamples based on market capitalisation categories. Stanadard deviations are given in square brackets.

2. Impulse = implied impact of a 100-share buy order on the log(quote midpoint)

3. σ_x = standard deviation of the innovation in x_t equation, a measure of trading intensity

4. σ_w = standard deviation of change in the efficient price

5. $\sigma_{w,x}$ = square root of the variance of trade-correlated part of the efficient price changes

6. R^2_w = the ratio of variance in trade-explained component of efficient price changes to the total variance of efficient price changes

7. Spread = time-weighted average spread proportional to stock price

5.4 Intraday Analysis

Intraday patterns were extensively researched by a number of academics. Admati and Pfleiderer (1988) predicted in theory that trade volumes and return volatilities tend be display 'U-shaped' pattern with concentrations at the beginning and end of a trading day. This prediction was subsequently confirmed by Hasbrouck (1991b). McInish and Wood (1989) claimed that a 'reverse J-shaped' pattern was found for intraday spreads. These findings are re-examined based on the current high frequency sample.

Following the convention used in Hasbrouck (1991b), the entire trading day is decomposed into three intervals: trading occurs in the first half-hour (9:30:00 A.M. to 10:00:00 A.M.), the interior period (10:00:01 A.M. to 3:30:00 P.M.), and trading occurs in the last half-hour (3:30:01 P.M. to 4:00:00 P.M).

One point is worth mentioning. In Hasbrouck's work, the intraday analysis was applied to a sample containing only large and medium cap stocks selected across all market value subsamples based on a minimum requirement for sample sizes. Out of this sample of large and medium cap stocks, 40 stocks are from the largest market cap subsample and the rest 6 stocks are from the second-largest market cap subsample. As a consequence, the analysis was biased towards large value stocks. Fortunately, the present study does not share this concern thanks to the high frequency data used.

Results are demonstrated in table 4. Firstly, both $\sigma_w/event$ and $\sigma_w/hour$ and higher in the first half-hour and the last half-hour periods than in the interior period, with the first half-hour figures higher than those for the last half-hour. This confirms the 'U-shaped' for return volatilities. Secondly, $\sigma_x/hour$ figures suggest that transaction volumes are higher in the beginning and end of trading, in line with Hasbrouck findings. Thirdly, larger spreads are found at the beginning but not the ending half-hour, confirming the 'reverse J-shaped' pattern stated by McInish and Wood. Fourthly, the Impulse figures that measures the ultimate log change in the quote midpoint resulting from the impact of a 1000-share purchase. This should be higher if the purchase occurred in the beginning half-hour, but it is not the case based on results in table 4.

Furthermore, $\sigma_{w,x}$[27] – the absolute measure of trade informativeness – is higher both at the beginning and at the end, but even more so at the beginning. Therefore, trades lead to more price changes at the first and the last half-hour relative to the midday periods, with the effect stronger at the first half trading hour. Last but not the least, R_w^2 is higher at the first half-hour, contrary to Hasbrouck's finding. Overall, results of $\sigma_{w,x}$ and R_w^2 together indicate trades are more informative at the start of a trading day both in absolute and in relative terms. That lower R_w^2 in the beginning half-hour found by Hasbrouck results probably from his sampling approach and limitation imposed by data availability[28], which fails to capture the more salient asymmetric information effects in smaller firms.

[27] Both on a per-event and in a per-hour basis.

[28] In Hasbrouck's work, sample size of smaller value stocks was not large enough for the analysis of intraday patterns.

Table 4 Summary intraday statistics for 15 selected stocks, 07-11 January 2008

	9:30 A.M. – 10:00 A.M.	10:00 A.M. – 3:30 P.M.	3:30 P.M. – 4:00 P.M.
No. of obs	13187	124145	13250
	[23386]	[306761]	[28928]
Impulse (x100)	-0.2	0.096	0.283
	[1.687]	[0.080]	[0.813]
σ_x /event	2.212	2.985	4.298
	[3.438]	[5.047]	[7.262]
σ_x /hour	9661.574	5503.737	12464.523
	[13952.104]	[7833.747]	[25558.298]
σ_w /event (x100)	0.111	0.039	0.089
	[0.104]	[0.023]	[0.064]
σ_w /hour (x100)	205.08	61.737	180.854
	[126.149]	[63.451]	[176.912]
$\sigma_{w,x}$ /event (x100)	0.044	0.012	0.03
	[0.061]	[0.010]	[0.029]
$\sigma_{w,x}$ /hour (x100)	69.925	21.414	58.087
	[49.749]	[25.864]	[80.081]
R^2_w	0.138	0.121	0.109
	[0.144]	[0.076]	[0.108]
Spread (x100)	0.825	0.187	0.165
	[0.991]	[0.125]	[0.115]

Note: 1. Figures are average values for the total sample of 15 firms and subsamples based on market capitalisation categories. Stanadard deviations are given in square brackets.

2. Impulse = implied impact of a 100-share buy order on the log(quote midpoint)

3. σ_x = standard deviation of the innovation in x_t equation, a measure of trading intensity

4. σ_w = standard deviation of change in the efficient price

5. $\sigma_{w,x}$ = square root of the variance of trade-correlated part of the efficient price changes

6. R^2_w = the ratio of variance in trade-explained component of efficient price changes to the total variance of efficient price changes

7. Spread = time-weighted average spread proportional to stock price

5.5 Cross-period Analysis – Pre, During and Post Crisis

As is mentioned in Section 4, a cross period analysis is presented in this section in the attempt to extend the research by Hasbrouck and numerous other academics which were performed before the 2008 financial crisis and in which data of high frequency was not utilised.

Seven different periods are covered in this analysis: 07-11 January (1), 07-11 April (2), 07-11 July (3) and 15-19 September (4) of the year 2008, and covers 13-17 April (5), 06-10 July (6) and 05-09 October (7) of the year 2009, with each period covering a full trading week of 5 days. The period of 15-19 September 2008 is denoted as the crisis period. Therefore, three periods are covered in both the pre-crisis and the post-crisis intervals. This setup should be able to capture changes in the trade informativeness as the markets approached the crisis and then left from the crisis.

5.5.1 Cross-sectional analysis

Figure 2 and 3 display the cross period plots for the total sample of 15 stocks based on total sample statistics in appendix A6. It should be noticed that there are two y-axes in each plot. This is because variables plotted on the same graph are in different magnitude or units. For example, in figure 2, the magnitude of $\sigma_w/hour$[29] is much higher than that of the rest of variables plotted on that graph, although all variables are expressed in percentage terms. In figure 3, the unit of variables is one instead of percentage, but the magnitude of No. of obs. is higher than that of $\sigma_x/hour$. Therefore, the plots are formatted in such a way that the first variable indicated by labels underneath the graphs has its y-axis on the left, and the rest variables have their y-axis on the right. This facilitates the readability of those graphs.

Excitingly, a number of clear patterns can be found in both figure 2 and 3. Firstly, $\sigma_x/hour$ in figure 2 peaks at crisis period and tends to fall for periods away from crisis. Hence, trade volumes are higher in the crisis period.

[29] Results for per-hour standard deviations resemble those for per-event standard deviations. But per-hour standard deviations are used as they show more pronounced patterns.

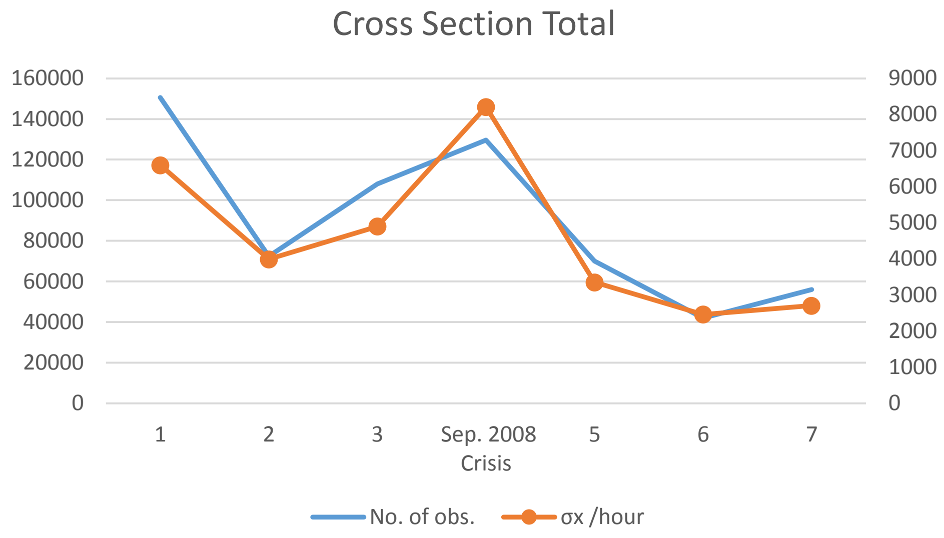


Figure 2 Cross-period summary plot for 15 stocks (A)

Regarding figure 3, spread widens as the market approaches the crisis, peaks in the first post-crisis period of 13-17 April 2009, and narrows as the market moves out of crisis. As the transaction return volatility is closely related to the spread behaviour (Hasbrouck, 1991b), this implies that the return volatility might also peak at the post-crisis period rather than in the crisis week. This seems to suggest that there is a lagged adjustment in the market's assessment of and its corresponding responses to the crisis.

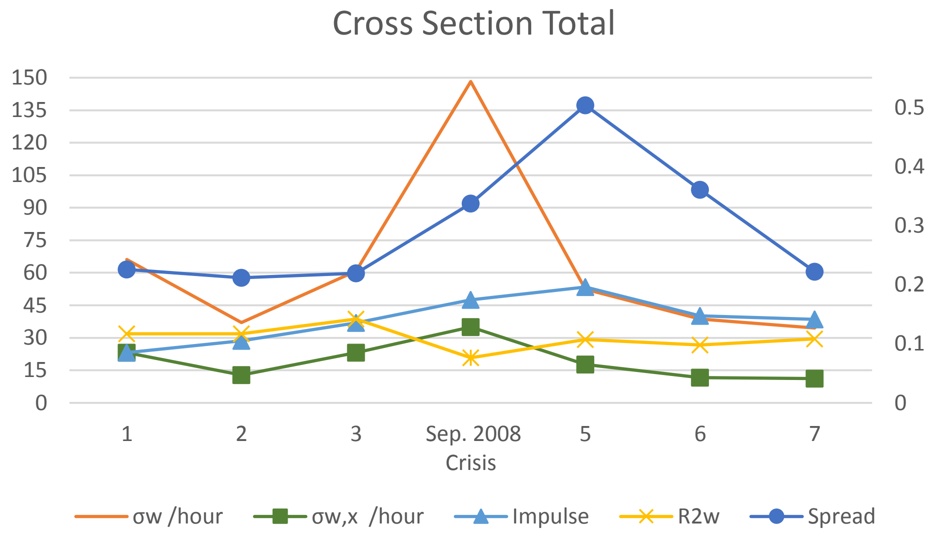


Figure 3 Cross-period summary plot for 15 stocks (B)

In terms of the three variables related to trade informativeness, patterns are also clear. The series of $\sigma_w/hour$ implies that volatility of the efficient price change tends to increase until it reaches its highest level at crisis week and falls afterwards. The series of

$\sigma_{w,x}/hour$ also has its maximum at the crisis week. But the combined force of these two variables lead to a trough for R_w^2 in the crisis period. One way of presenting this finding is that although more private information was revealed by trades in the crisis (in absolute terms), this is more than offset by larger amount of all other public information. This may be justified by the fact that correlations within and across asset classes tend to surge in the crisis and that market as a whole plummets, leading to an increase in the correlation between public and private information.

In terms of large, medium and small cap stocks, three primary plots are demonstrated in figure 4 to 6, three secondary plots are placed in appendix A7 (a), the corresponding statistics tables are displayed in appendix A7 (b). Some cross-firm findings are worth stating. Firstly, trading intensity suggested by $\sigma_x/hour$[30] peaks in the crisis week across all market value subsamples. Secondly, lagged summit of return volatility occurs only for small cap stocks. For large and medium value stocks, return volatility reaches the highest level at crisis. Thirdly, the phenomenon of highest level of $\sigma_w/hour$, highest level of $\sigma_{w,x}/hour$, and lowest level of R_w^2 in the crisis week prevails across all market value subsamples.

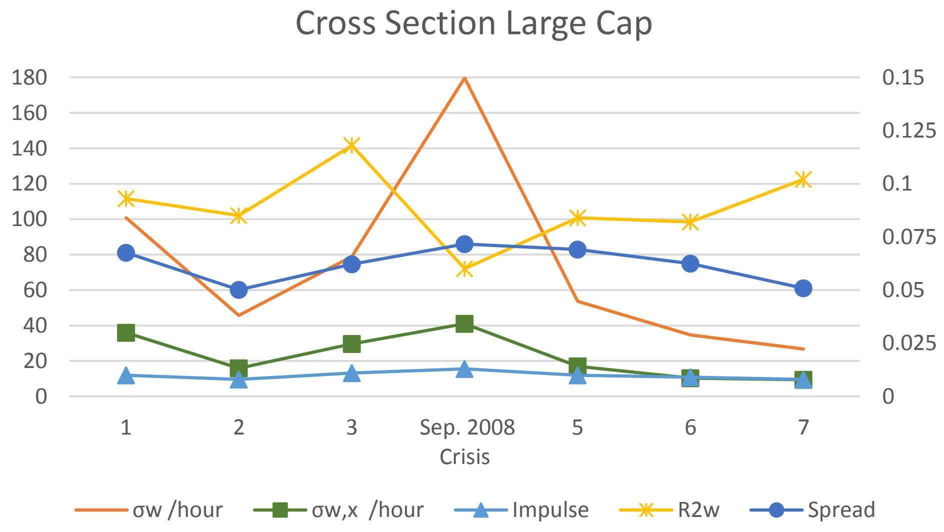


Figure 4 Cross-period summary plot for large cap stocks

[30] Plot for this variable is in appendix A7 (a). Other variables involved in the discussion are plotted in figure 4 to 6.

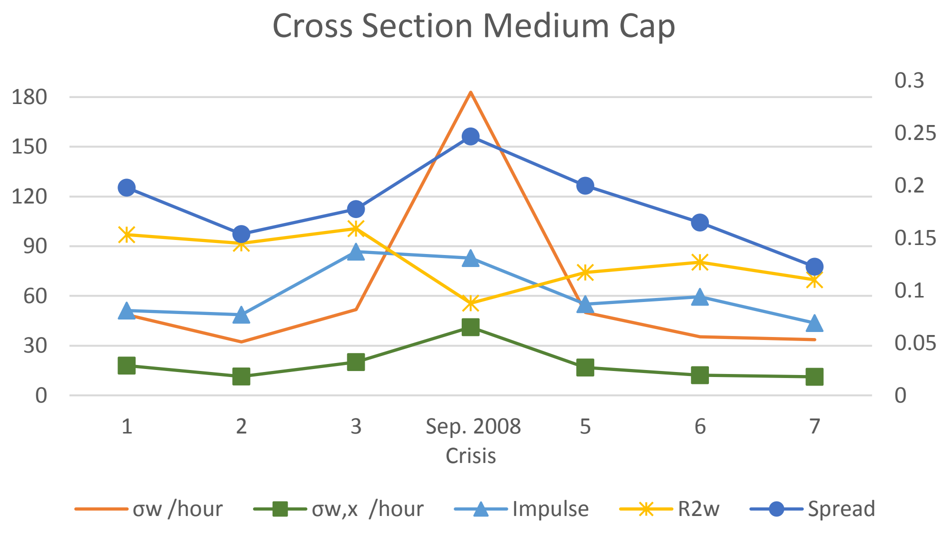


Figure 5 Cross-period summary plot for medium cap stocks

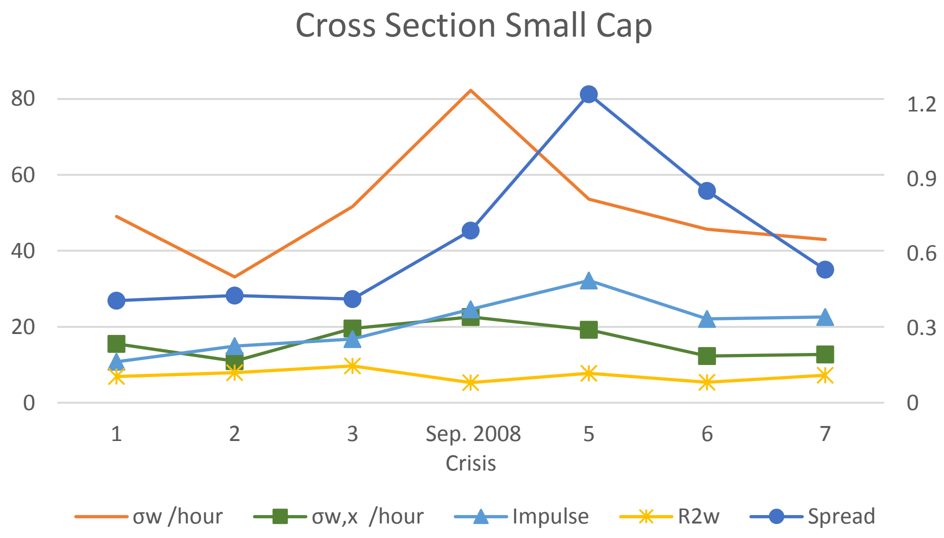


Figure 6 Cross-period summary plot for large cap stocks

5.5.2 Intraday analysis

For this part, three primary plots are displayed in figure 7 to 9, three secondary plots are shown in appendix A8 (a) and the statistics tables are provided in appendix A8 (b). For the reason of brevity, most findings in this section is stated from a descriptive viewpoint as results share similar interpretation as those in the last section 5.5.1:

1. $\sigma_x/hour$[31] hits the highest level at crisis across all intraday intervals;
2. Delayed peak of the spread exists for midday and the last half-hour of trading;
3. Lagged peak of impulse response occurs for all intraday sub-periods, a pattern not shared by cross sectional analysis 5.5.1;
4. That pattern of highest level of $\sigma_w/hour$, highest level of $\sigma_{w,x}/hour$, and lowest level of R_w^2 in the crisis week prevails across all intraday sub-periods, confirming the findings in section 5.5.1.

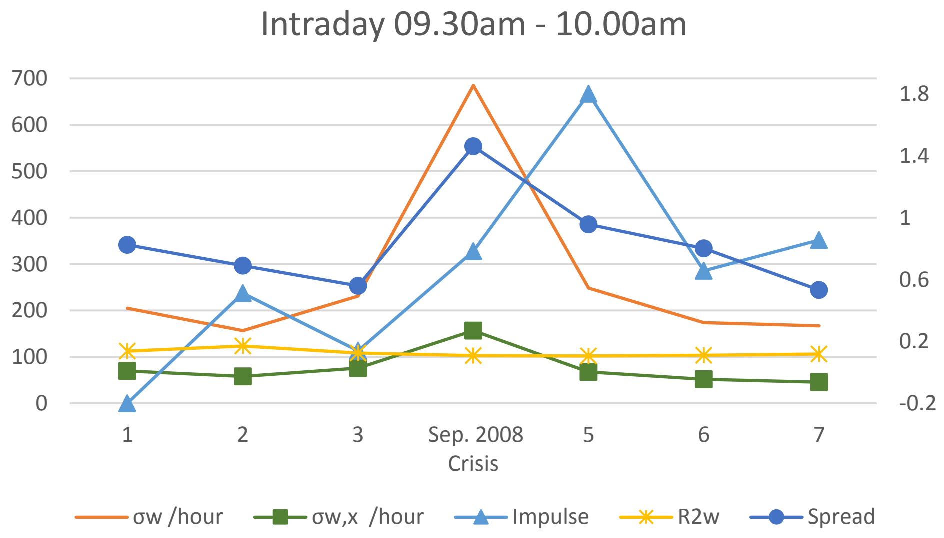


Figure 7 Cross-period summary plot for beginning of trading

[31] Plot for this variable is in appendix A8 (a). Other variables involved in the discussion are plotted in figure 7 to 9.

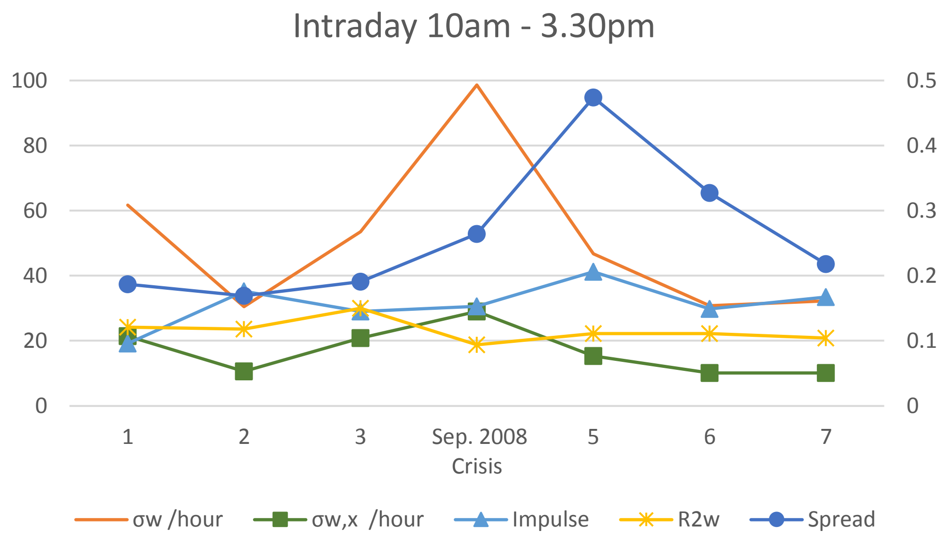


Figure 8 Cross-period summary plot for middle of the day

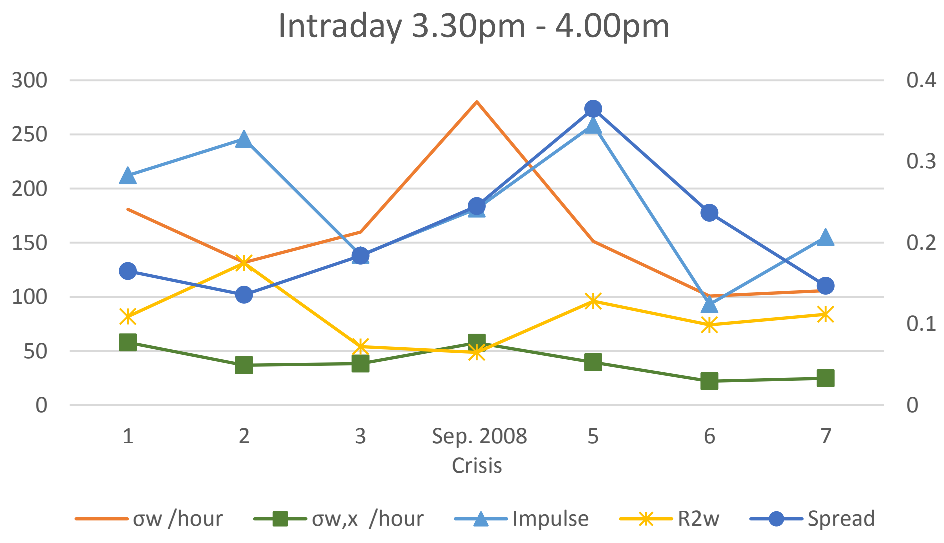


Figure 9 Cross-period summary plot for end of trading

6. Conclusion

The present study has employed a four-variable structural vector autoregression (SVAR) system modelling stock trade-quote interaction and several measures of trade informativeness developed by Hasbrouck (1991a,b) to extract asymmetric information content hidden in stock trades. Motivated by the availability and the crisis-coverage feature of high frequency data, the current work attempts to re-evaluate, refine and extend empirical findings in Hasbrouck's studies, in which results were generated based on out-of-date transaction data which behaves differently from data recorded under current market conditions and technologies.

This work contributes to the existing literature in three aspects. Firstly, it confirms some of the findings in previous research. Secondly, it raises counter results relative to previous studies. Thirdly, it extends past works by performing a cross-period analysis; novel patterns are discovered. Overall, results in the present research are consistent with the following conclusions.

1. The persistent impact of trade innovation – the unanticipated component of the trade – is a positive, increasing but convex function of trade innovation size;
2. Trades of smaller firms convey more asymmetric information in both absolute and relative sense;
3. Trades are more informative at the beginning of trading not only in absolute but also in relative terms;
4. Trades are more informative in absolute terms but less informative relative to total public information in the crisis week relative to other periods;
5. It seems that there is a lagged adjustment in the market's assessment of and its corresponding responses to the crisis.

Two cautions are worth raising. Firstly, the bootstrap approached used by Hasbrouck to correct for sample-size biases[32] in the estimation results is not applied in present study. Nonetheless, it is expected that this would not alter the nature of results in the

[32] There is a great difference in the transaction frequencies of different companies, which may lead to inaccurate inferences when the summary measures of trade informativeness are utilised for comparative and cross-sectional analyses (Hasbrouck 1991b).

current work as is shown by Hasbrouck (1991b). Secondly, a more complete picture of responses of trade informativeness to the changing market conditions throughout the crisis may be captured by bringing in more sample periods.

Since the current database contains information on different types of participating firms in security transactions[33], one further research direction is to test for the differences in asymmetric information generated by trading activities of high frequency trading firms and non-HFT firms.

[33] As discussed in section 4, the trade record also contains information about the type of any liquidity provider or liquidity seeker, e.g. whether the transaction participating party is a HFT firm or non-HFT firm.

7. References

Admati, A. & Pfleiderer, P. (1988) A theory of intraday patterns: Volume and price variability. *Review of Financial Studies.* 1, 3-40.

Bagehot, W. (pseudonym) (1971) The only game in town. *Financial Analysts Journal*. 8, 31-53.

Brogaard, J., Hendershott, T. & Riordan, R. (2014) High-Frequency Trading and Price Discovery. *Review of Financial Studies.* 27, 2267-2306.

Chiang, R. & Venkatesh, P. C. (1988) Insider holdings and perceptions of information asymmetry: A note. *Journal of Finance.* 43, 1041-1048.

Copeland, T. & Galai, D. (1983) Information effects on the bid-ask spread. *Journal of Finance*. 38, 1457-69.

de Jong, F., Nijman, T. & Roell, A. (1996) Price effects of trading and components of the bid-ask spread on the Paris Bourse. *Journal of Empirical Finance*. 3, 193-213.

Easley, D., & O'hara, M. (2004) Information and the cost of capital. *Journal of Finance.* 59, 1553-1583.

Easley, D., Kiefer, N. M. & O'Hara, M. (1997a) One day in the life of a very common stock. *Review of Financial Studies*. 10, 805-835.

Easley, D. & O'Hara, M. (1987) Price, trade size, and information in securities markets. *Journal of Financial Economics.* 18, 69-90.

Glosten, L. R. & Harris, L. E. (1988) Estimating the components of the bid ask spread. *Journal of Financial Economics.* 21, 123-142.

Glosten, L. R. & Milgrom, P. R. (1985) Bid, ask and transaction prices in a market-maker market with heterogeneously informed traders. *Journal of Financial Economics.* 14, 71-100.

Grossman J. S. & Stiglitz J. (1980) On the Impossibility of Informationally Efficient Markets. *American Economic Review.* 70, 393-408.

Hasbrouck, J. (1991a) Measuring the Information Content of Stock Trades. *Journal of Finance*. 46, 179-207.

Hasbrouck, J. (1991b) The summary informativeness of stocks trades: an econometric analysis. *Review of Financial Studies.* 4, 571-595.

Hasbrouck, J. (1988) Trades, quotes, inventories and information. *Journal of Financial Economics.* 22 229-252.

Hasbrouck, J. & Ho, S. Y. T. (1987) Order Arrival, Quote Behaviour and the Return Generating Process. *Journal of Finance.* 42, 1035-1048.

Kyle, A. S. (1985) Continuous Auctions and Insider Trading. *Econometrica.* 53, 1315-1336.

McInish, H. T. & Wood, A. R. (1992) An Analysis of Intraday Patterns in Bid/Ask Spreads for NYSE Stocks. *Journal of Finance*. 47, 753-764.

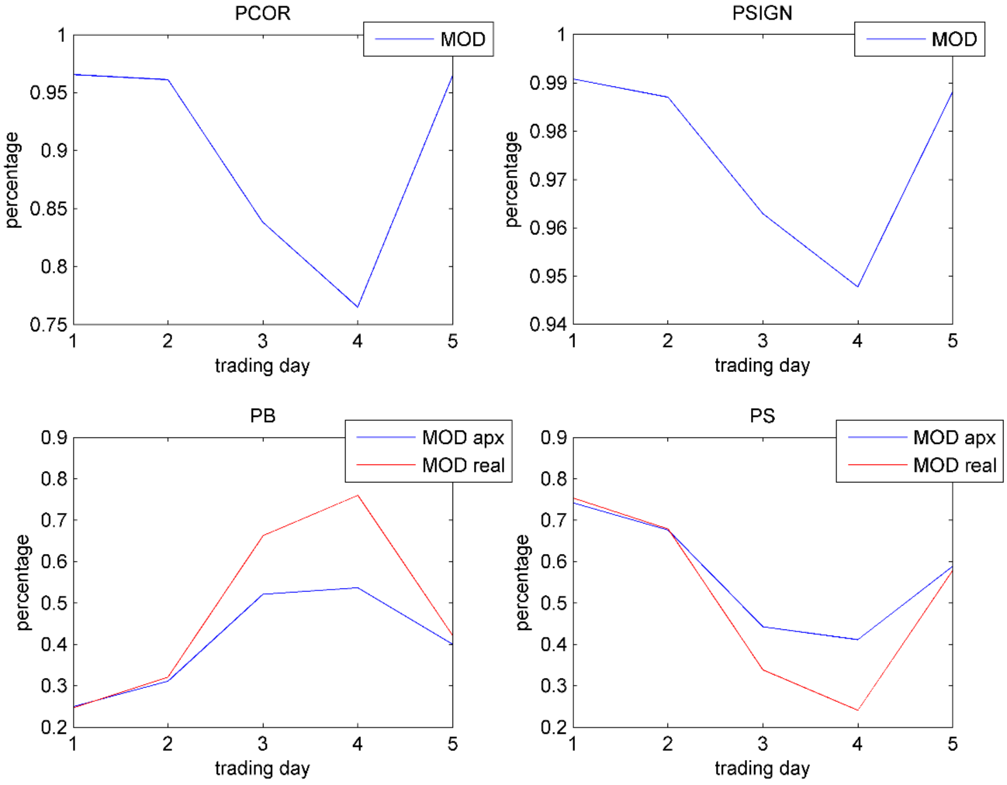
8. Appendices

Appendix A – Tables and Figures

A1 – Company Name, Ticker Symbol and Market Value Categories

Name	Symbol
LARGE CAP Sample:	
1. AMAZON COM INC	AMZN
2. APPLIED MATERIALS INC	AMAT
3. WALT DISNEY CO	DIS
4. GAP INC	GPS
5. GOOGLE INC	GOOG
MEDIUM CAP Sample:	
1. CENTRAL EUROPEAN MEDIA ENTERPRISES	CETV
2. FTI CONSULTING INC	FCN
3. LANDSTAR SYSTEM INC	LSTR
4. LIFEPOINT HOSPITALS INC	LPNT
5. SEACOR HOLDINGS INC	CKH
SMALL CAP Sample	
1. BRUSH ENGINEERED MATERIALS INC	BW
2. CEDAR SHOPPING CENTRES	CDR
3. MODINE MANUFACTURING CO	MOD
4. NXSTAGE MEDICAL INC	NXTM
5. PRESTIGE BRANDS HOLDINGS INC	PBH

A2 – Performance Evaluation Figure of DA Approach – Modine Manufacturing (MOD)



A3 – Number of Trade and Quote Observations – Raw & Cleansed Data

a. Raw Data

	LARGE CAP									
	Number of Trades (Raw)					Number of Quote Prices (Raw)				
	AMZN	AMAT	DIS	GPS	GOOG	AMZN	AMAT	DIS	GPS	GOOG
Day1	34,921	30,528	13,590	12,814	28,783	153,395	150,206	87,172	108,948	315,850
Day2	44,150	42,402	16,799	12,189	23,690	176,818	203,787	103,167	97,191	434,594
Day3	55,591	39,392	27,723	17,063	30,113	189,668	205,147	152,573	115,123	389,634
Day4	38,869	46,145	16,348	30,685	28,650	144,291	230,031	149,111	173,123	330,106
Day5	35,978	36,781	12,731	15,755	22,717	146,531	200,095	95,953	110,866	227,062
Total	209,509	195,248	87,191	88,506	133,953	810,703	989,266	587,976	605,251	1,697,246

	MEDIUM CAP									
	Number of Trades (Raw)					Number of Quote Prices (Raw)				
	CETV	FCN	LSTR	LPNT	CKH	CETV	FCN	LSTR	LPNT	CKH
Day1	995	1,973	5,676	3,142	459	8,916	22,003	28,493	16,817	6,972
Day2	1,204	1,273	3,509	2,240	306	13,502	20,796	27,328	12,808	4,144
Day3	1,209	1,557	3,786	1,860	304	21,681	20,932	29,767	22,829	9,919
Day4	2,015	1,162	4,880	4,448	322	50,831	27,972	23,931	20,411	18,008
Day5	1,411	939	3,120	3,255	323	19,027	30,061	19,583	18,860	9,762
Total	6,834	6,904	20,971	14,945	1,714	113,957	121,764	129,102	91,725	48,805

	SMALL CAP									
	Number of Trades (Raw)					Number of Quote Prices (Raw)				
	BW	CDR	MOD	NXTM	PBH	BW	CDR	MOD	NXTM	PBH
Day1	691	757	870	685	216	21,531	4,952	8,357	8,400	3,126
Day2	1,000	1,030	617	830	603	17,091	6,476	8,187	6,229	4,515
Day3	909	1,151	728	671	945	25,876	8,439	7,157	6,574	3,604
Day4	992	1,270	574	1,485	664	29,078	10,540	7,280	6,075	3,860
Day5	265	678	423	956	1,050	19,034	8,948	6,048	4,722	5,668
Total	3,857	4,886	3,212	4,627	3,478	112,610	39,355	37,029	32,000	20,773

Note :quote price = midpoint of quote bid and ask prices

b. Cleansed Data

	LARGE CAP									
	Number of Trades (Clean)					Number of Quote Revisions (Clean)				
	AMZN	AMAT	DIS	GPS	GOOG	AMZN	AMAT	DIS	GPS	GOOG
Day1	24,590	4,422	6,822	5,979	21,519	56,681	4,463	7,668	7,544	268,848
Day2	30,423	6,550	8,173	6,353	17,720	67,200	6,687	8,950	9,023	361,006
Day3	37,306	4,598	12,229	7,637	22,329	66,192	4,625	13,377	9,202	327,669
Day4	28,078	5,962	8,267	12,099	21,607	59,624	6,074	10,062	13,482	277,845
Day5	25,355	4,738	7,154	7,849	17,597	50,182	4,743	11,367	8,556	183,976
Total	145,752	26,270	42,645	39,917	100,772	299,879	26,592	51,424	47,807	1,419,344

	MEDIUM CAP									
	Number of Trades (Clean)					Number of Quote Revisions (Clean)				
	CETV	FCN	LSTR	LPNT	CKH	CETV	FCN	LSTR	LPNT	CKH
Day1	749	1,743	4,004	2,161	411	4,443	16,865	11,725	4,693	5,139
Day2	890	1,152	2,665	1,561	282	7,233	16,272	9,410	3,154	3,639
Day3	897	1,381	2,909	1,432	289	11,116	14,916	12,539	4,090	8,456
Day4	1,385	1,087	3,581	2,913	297	36,972	22,826	11,062	5,403	16,382
Day5	1,033	810	2,452	2,249	292	12,874	24,138	9,338	4,640	7,720
Total	4,954	6,173	15,611	10,316	1,571	72,638	95,017	54,074	21,980	41,336

	SMALL CAP									
	Number of Trades (Clean)					Number of Quote Revisions (Clean)				
	BW	CDR	MOD	NXTM	PBH	BW	CDR	MOD	NXTM	PBH
Day1	611	587	734	539	189	15,059	1,622	2,868	2,882	1,056
Day2	883	793	532	620	531	9,868	1,845	3,938	2,323	1,692
Day3	795	825	622	527	675	15,699	2,404	3,568	2,704	1,281
Day4	872	906	522	1,178	511	20,794	4,892	4,579	3,121	1,260
Day5	247	558	374	667	725	13,810	3,829	3,645	2,055	1,891
Total	3,408	3,669	2,784	3,531	2,631	75,230	14,592	18,598	13,085	7,180

Note :quote revision = log(quote price(t)) − log(quote price(t-1))

A4 – Stationarity Tests – Raw & Cleansed Data

a. Raw Data

Stationarity/Unit Root Test for Trading Time RAW Data 07 – 11 January 2008

					LARGE CAP					
	Trades (Raw)					Quote Prices (Raw)				
	AMZN	AMAT	DIS	GPS	GOOG	AMZN	AMAT	DIS	GPS	GOOG
ADF ⍰	-114.01**	-115.749**	-74.144**	-74.327**	-97.843**	-0.69	-2.593	-1.545	-0.376	-2.517
(Drift)	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.847]	[0.094]	[0.498]	[0.91]	[0.112]
ADF ⍰	-114.12**	-115.756**	-74.165**	-75.691**	-97.843**	-1.936	-3.451*	-1.292	-3.293	-2.61
(DT)	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.625]	[0.045]	[0.889]	[0.068]	[0.291]
PP⍰	-434.453**	-420.252**	-275.205**	-284.204**	-341.984**	-0.665	-2.596	-1.531	-0.331	-3.945**
(Drift)	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.853]	[0.094]	[0.504]	[0.918]	[0.002]
PP⍰	-434.487**	-420.253**	-275.211**	-284.68**	-341.983**	-1.869	-3.487*	-1.378	-3.58*	-4.132**
(DT)	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.658]	[0.041]	[0.867]	[0.032]	[0.006]
KPSS⍰	4.511**	1.105**	9.371**	32.492**	0.288	5423.431**	5527.605**	1601.262**	5286.583**	2900.099**
(Drift)	[0.01]	[0.01]	[0.01]	[0.01]	[0.1]	[0.01]	[0.01]	[0.01]	[0.01]	[0.01]
KPSS⍰	1.272**	0.757**	8.947**	2.416**	0.288**	462.372**	248.16**	936.063**	231.586**	1439.08**
(DT)	[0.01]	[0.01]	[0.01]	[0.01]	[0.01]	[0.01]	[0.01]	[0.01]	[0.01]	[0.01]
					MEDIUM CAP					
	Trades (Raw)					Quote Prices (Raw)				
	CETV	FCN	LSTR	LPNT	CKH	CETV	FCN	LSTR	LPNT	CKH
ADF ⍰	-21.638**	-21.6**	-36.739**	-30.109**	-9.949**	-1.856	-4.216**	-1.654	0.294	-4.503**
(Drift)	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.36]	[0.001]	[0.45]	[0.978]	[0.001]
ADF ⍰	-21.711**	-21.709**	-36.792**	-30.171**	-10.022**	-1.659	-4.215**	-1.7	-4.51**	-6.073**
(DT)	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.762]	[0.005]	[0.742]	[0.002]	[0.001]
PP⍰	-80.164**	-79.363**	-140.239**	-113.732**	-37.926**	-2.265	-5.73**	-1.576	0.293	-5.993**
(Drift)	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.184]	[0.001]	[0.484]	[0.978]	[0.001]
PP⍰	-80.187**	-79.388**	-140.255**	-113.749**	-37.936**	-2.296	-5.793**	-1.602	-4.241**	-8.982**
(DT)	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.447]	[0.001]	[0.79]	[0.004]	[0.001]
KPSS⍰	0.756**	0.742**	0.819**	1.696**	0.473*	392.815**	136.936**	191.873**	780.079**	244.511**
(Drift)	[0.01]	[0.01]	[0.01]	[0.01]	[0.048]	[0.01]	[0.01]	[0.01]	[0.01]	[0.01]
KPSS⍰	0.333**	0.246**	0.544**	0.805**	0.174*	178.889**	114.554**	189.666**	87.148**	39.435**
(DT)	[0.01]	[0.01]	[0.01]	[0.01]	[0.027]	[0.01]	[0.01]	[0.01]	[0.01]	[0.01]
					SMALL CAP					
	Trades (Raw)					Quote Prices (Raw)				
	BW	CDR	MOD	NXTM	PBH	BW	CDR	MOD	NXTM	PBH
ADF ⍰	-17.533**	-17.863**	-15.515**	-18.113**	-11.895**	-1.879	-3.976**	-0.574	-0.341	-2.499
(Drift)	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.35]	[0.002]	[0.873]	[0.916]	[0.116]
ADF ⍰	-17.774**	-17.9**	-15.988**	-18.253**	-12.746**	-1.769	-4.153**	-2.702	-2.503	-5.731**
(DT)	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.708]	[0.005]	[0.245]	[0.344]	[0.001]
PP⍰	-59.271**	-66.749**	-53.066**	-65.556**	-52.906**	-2.582	-6.58**	-1.808	-0.635	-3.892**
(Drift)	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.097]	[0.001]	[0.381]	[0.86]	[0.003]
PP⍰	-59.357**	-66.754**	-53.214**	-65.604**	-53.168**	-2.631	-6.951**	-4.855**	-2.536	-8.622**
(DT)	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.28]	[0.001]	[0.001]	[0.327]	[0.001]
KPSS⍰	0.954**	0.339	1.947**	0.722*	4.034**	459.931**	52.828**	232.469**	203.668**	133.425**
(Drift)	[0.01]	[0.1]	[0.01]	[0.012]	[0.01]	[0.01]	[0.01]	[0.01]	[0.01]	[0.01]
KPSS⍰	0.08	0.201*	0.138	0.157*	0.308**	209.037**	30.4**	20.336**	44.267**	5.193**
(DT)	[0.1]	[0.016]	[0.065]	[0.041]	[0.01]	[0.01]	[0.01]	[0.01]	[0.01]	[0.01]

Notes: 1. p-values are given in the square brackets

2. * and ** denote that the test is significant under 5% and 1% significance level respectively.

3. quote price = midpoint of quote bid and ask prices

4. DT stands for drift and trend

b. Cleansed Data

Stationarity/Unit Root Test for CLEANSED Data 07 – 11 January 2008

	LARGE CAP									
	Trades (Clean)					Quote Revisions (Clean)				
	AMZN	AMAT	DIS	GPS	GOOG	AMZN	AMAT	DIS	GPS	GOOG
ADF	-145.522**	-43.268**	-61.051**	-58.377**	-301.819**	-149.102**	-46.948**	-71.185**	-66.345**	-389.922**
(Drift)	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]
ADF	-145.561**	-43.279**	-61.111**	-58.397**	-301.82**	-149.102**	-46.948**	-71.189**	-66.347**	-389.922**
(DT)	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]
PP	-515.278**	-150.07**	-211.265**	-207.396**	-1038.246**	-685.331**	-180.931**	-298.473**	-262.642**	-5175.979**
(Drift)	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]
PP	-515.289**	-150.071**	-211.282**	-207.401**	-1038.246**	-685.33**	-180.929**	-298.481**	-262.642**	-5175.978**
(DT)	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]
KPSS	1.391**	0.342	1.156**	0.564*	0.246	0.094	0.02	0.164	0.082	0.018
(Drift)	[0.01]	[0.1]	[0.01]	[0.027]	[0.1]	[0.1]	[0.1]	[0.1]	[0.1]	[0.1]
KPSS	0.134	0.24**	0.164*	0.44**	0.173*	0.075	0.016	0.1	0.042	0.015
(DT)	[0.072]	[0.01]	[0.035]	[0.01]	[0.028]	[0.1]	[0.1]	[0.1]	[0.1]	[0.1]

	MEDIUM CAP									
	Trades (Clean)					Quote Revisions (Clean)				
	CETV	FCN	LSTR	LPNT	CKH	CETV	FCN	LSTR	LPNT	CKH
ADF	-71.271**	-81.256**	-58.638**	-36.834**	-54.383**	-97.854**	-120.917**	-67.072**	-43.398**	-86.855**
(Drift)	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]
ADF	-71.307**	-81.352**	-59.428**	-37.242**	-54.383**	-97.859**	-120.918**	-67.077**	-43.44**	-86.854**
(DT)	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]
PP	-246.929**	-272.84**	-208.386**	-134.989**	-177.694**	-679.024**	-736.203**	-352.967**	-205.562**	-489.343**
(Drift)	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]
PP	-246.937**	-272.862**	-208.591**	-135.099**	-177.692**	-679.045**	-736.207**	-352.97**	-205.62**	-489.341**
(DT)	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]
KPSS	0.837**	2.667**	13.823**	4.635**	0.088	0.083	0.037	0.128	0.381	0.009
(Drift)	[0.01]	[0.01]	[0.01]	[0.01]	[0.1]	[0.1]	[0.1]	[0.1]	[0.085]	[0.1]
KPSS	0.185*	0.527**	0.75**	0.562**	0.072	0.029	0.019	0.082	0.121	0.007
(DT)	[0.021]	[0.01]	[0.01]	[0.01]	[0.1]	[0.1]	[0.1]	[0.1]	[0.096]	[0.1]

	SMALL CAP									
	Trades (Clean)					Quote Revisions (Clean)				
	BW	CDR	MOD	NXTM	PBH	BW	CDR	MOD	NXTM	PBH
ADF	-74.535**	-29.384**	-29.882**	-29.428**	-20.169**	-102.716**	-45.339**	-55.285**	-36.358**	-35.122**
(Drift)	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]
ADF	-74.737**	-29.563**	-30.31**	-29.432**	-20.254**	-102.719**	-45.337**	-55.3**	-36.377**	-35.127**
(DT)	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]
PP	-253.553**	-112.528**	-129.345**	-110.477**	-78.226**	-728.344**	-243.117**	-409.675**	-198.82**	-200.204**
(Drift)	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]
PP	-253.611**	-112.581**	-129.492**	-110.475**	-78.246**	-728.36**	-243.113**	-409.778**	-198.88**	-200.204**
(DT)	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]	[0.001]
KPSS	4.244**	2.658**	7.555**	0.217	0.898**	0.106	0.051	0.074	0.12	0.012
(Drift)	[0.01]	[0.01]	[0.01]	[0.1]	[0.01]	[0.1]	[0.1]	[0.1]	[0.1]	[0.1]
KPSS	0.386**	0.665**	2.08**	0.173*	0.669**	0.034	0.02	0.018	0.032	0.01
(DT)	[0.01]	[0.01]	[0.01]	[0.028]	[0.01]	[0.1]	[0.1]	[0.1]	[0.1]	[0.1]

Notes: 1. p-values are given in the square brackets

2. * and ** denote that the test is significant under 5% and 1% significance level respectively.

3. quote revision = log(quote price(t)) − log(quote price(t-1))

4. DT stands for drift and trend

A5 – Estimation Outputs of SVAR(5) – Modine Manufacturing (MOD)

Estimates of the multivariate SVAR(5) for Modine Manufacturing Co. (MOD), 07-11 January 2008

x^0_t	Coeff.	x_t	Coeff.	x^2_t	Coeff.	r_t	Coeff.
						$\mathbf{d_{id,0}}$	0.000494***
							5.79
$a_{id,1}$	0.141***	$b_{id,1}$	0.238***	$c_{id,1}$	1.87***	$d_{id,1}$	9.53E-05
	9.84		8.16		6.66		1.11
$a_{id,2}$	0.31***	$b_{id,2}$	0.369***	$c_{id,2}$	0.836***	$d_{id,2}$	-0.000106
	21.61		12.68		2.97		1.22
$a_{id,3}$	-0.00761	$b_{id,3}$	-0.0398	$c_{id,3}$	-0.514*	$d_{id,3}$	1.53E-05
	0.52		1.35		1.8		0.18
$a_{id,4}$	0.152***	$b_{id,4}$	0.159***	$c_{id,4}$	0.243	$d_{id,4}$	-0.000167*
	10.62		5.48		0.87		1.95
$a_{id,5}$	0.0575***	$b_{id,5}$	0.0725***	$c_{id,5}$	0.227	$d_{id,5}$	-0.000162*
	4.01		2.5		0.81		1.89
						$\mathbf{d_{td,0}}$	0.000178***
							2.97
$a_{td,1}$	0.000584	$b_{td,1}$	-0.0269	$c_{td,1}$	-1.22***	$d_{td,1}$	2.57E-05
	0.06		1.31		6.18		0.43
$a_{td,2}$	-0.0205**	$b_{td,2}$	-0.00314	$c_{td,2}$	-0.147	$d_{td,2}$	1.26E-05
	2.03		0.15		0.74		0.21
$a_{td,3}$	0.0144	$b_{td,3}$	0.0381*	$c_{td,3}$	0.353*	$d_{td,3}$	7.91E-05
	1.42		1.86		1.78		1.31
$a_{td,4}$	-0.0132	$b_{td,4}$	0.0106	$c_{td,4}$	0.185	$d_{td,4}$	7.59E-05
	1.3		0.52		0.93		1.26
$a_{td,5}$	-0.0298***	$b_{td,5}$	-0.0306	$c_{td,5}$	-0.0468	$d_{td,5}$	9.53E-05
	2.95		1.49		0.24		1.58
						$\mathbf{d_{sq,0}}$	5.81E-06
							1.32
$a_{sq,1}$	8.94E-05	$b_{sq,1}$	0.00104	$c_{sq,1}$	0.0581***	$d_{sq,1}$	0.00
	0.12		0.69		4		0.32
$a_{sq,2}$	0.00136*	$b_{sq,2}$	0.000118	$c_{sq,2}$	0.00808	$d_{sq,2}$	2.04E-06
	1.83		0.08		0.56		0.46
$a_{sq,3}$	-0.000713	$b_{sq,3}$	-0.000981	$c_{sq,3}$	-0.0114	$d_{sq,3}$	-4.83E-06
	0.96		0.65		0.78		1.09
$a_{sq,4}$	0.000102	$b_{sq,4}$	-0.000624	$c_{sq,4}$	-0.0064	$d_{sq,4}$	-1.76E-06
	0.14		0.41		0.44		0.4
$a_{sq,5}$	0.00212***	$b_{sq,5}$	0.003**	$c_{sq,5}$	0.00791	$d_{sq,5}$	-6.40E-06
	2.86		2		0.55		1.45
$a_{r,1}$	-8.43***	$b_{r,1}$	-11.9***	$c_{r,1}$	-38.9	$d_{r,1}$	-0.741***
	6.9		4.81		1.63		101.81
$a_{r,2}$	-9.04***	$b_{r,2}$	-14***	$c_{r,2}$	-52.1*	$d_{r,2}$	-0.302***
	5.96		4.54		1.75		33.41
$a_{r,3}$	-5.24***	$b_{r,3}$	-8.74***	$c_{r,3}$	-30.2	$d_{r,3}$	-0.297***
	3.45		2.84		1.02		32.88
$a_{r,4}$	-4.16***	$b_{r,4}$	-5.65*	$c_{r,4}$	-12.8	$d_{r,4}$	-0.137***
	2.74		1.84		0.43		15.17
$a_{r,5}$	-1.67	$b_{r,5}$	-0.835	$c_{r,5}$	13.3	$d_{r,5}$	-0.112***
	1.37		0.34		0.56		15.4
$\bar{R}^2$	0.181	$\bar{R}^2$	0.085	$\bar{R}^2$	0.006	$\bar{R}^2$	0.459
$\hat{\sigma}^2_{id}$	0.119	$\hat{\sigma}^2_{td}$	0.489	$\hat{\sigma}^2_{sq}$	45.624	$\hat{\sigma}^2_{r}$	4.22E-06

*/**/*** denotes that the figure is significantly different from 0 under 10%/5% /1% significance level

A6 – Cross-period Summary Statistics Total Sample

Cross-period summary statistics for 15 stocks

	Pre Crisis Period			Crisis Period	Pre Crisis Period		
	07-11	07-11	07-11	15-19	13-17	06-10	05-09
	January 2008	April 2008	July 2008	September	April 2009	July 2009	October 2009
No. of firms	15			15			15
No. of obs.	150584	72421	108026	129635	70078	41968	56015
	[358153]	[144044]	[186677]	[157493]	[151001]	[63332]	[86277]
Impulse	0.085	0.105	0.135	0.174	0.196	0.147	0.141
(x100)	[0.075]	[0.121]	[0.126]	[0.210]	[0.258]	[0.162]	[0.204]
σ_x /event	3.034	3.095	2.393	2.321	2.734	2.841	3.333
	[4.956]	[4.848]	[3.625]	[3.405]	[3.256]	[4.052]	[6.663]
σ_x /hour	6594.44	3990.203	4900.721	8208.423	3354.269	2460.638	2707.011
	[9841.262]	[5260.015]	[6424.416]	[10741.609]	[4285.586]	[2981.806]	[3338.199]
σ_w /event	0.043	0.051	0.047	0.079	0.097	0.074	0.044
(x100)	[0.032]	[0.053]	[0.036]	[0.064]	[0.103]	[0.065]	[0.038]
σ_w /hour	66.175	37.044	60.781	148.283	52.435	38.599	34.503
(x100)	[61.945]	[23.113]	[33.734]	[100.121]	[29.495]	[18.894]	[19.064]
$\sigma_{w,x}$ /event	0.012	0.016	0.016	0.019	0.032	0.02	0.013
(x100)	[0.009]	[0.020]	[0.015]	[0.017]	[0.044]	[0.019]	[0.012]
$\sigma_{w,x}$ /hour	23.147	12.753	23.119	34.896	17.678	11.63	11.165
(x100)	[27.180]	[11.707]	[21.158]	[23.717]	[15.049]	[7.413]	[7.357]
R^2_w	0.117	0.117	0.142	0.076	0.107	0.098	0.108
	[0.092]	[0.087]	[0.093]	[0.049]	[0.069]	[0.065]	[0.074]
Spread	0.226	0.212	0.219	0.337	0.504	0.360	0.222
(x100)	[0.171]	[0.178]	[0.158]	[0.274]	[0.640]	[0.405]	[0.295]

Note: 1. Figures are average values for the total sample of 15 firms. Stanadard deviations are given in square brackets.

2. Impulse = implied impact of a 100-share buy order on the log(quote midpo

3. σ_x /hour = standard deviation of the innovation in x_t equation, a measure of trading intensity

4. σ_w /hour = standard deviation of change in the efficient price

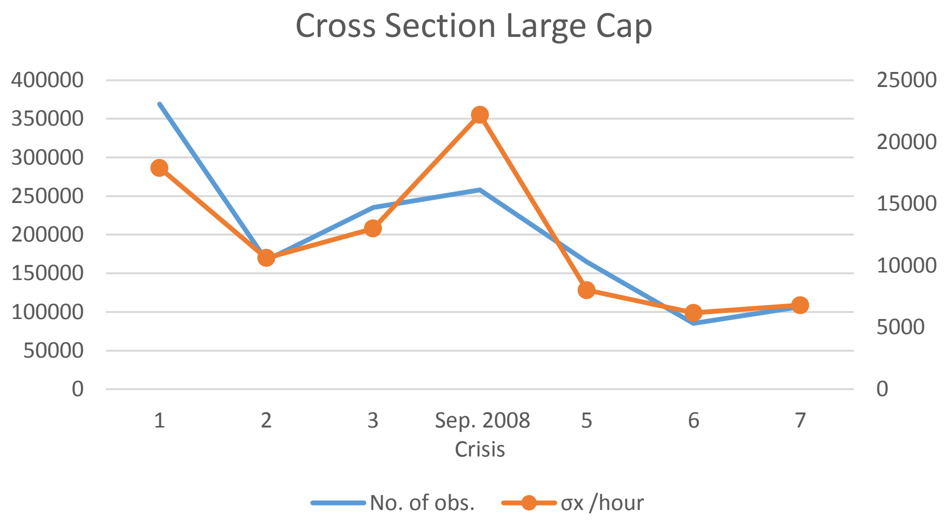
5. $\sigma_{w,x}$ /hour = square root of the variance of trade-correlated part of the efficient price changes

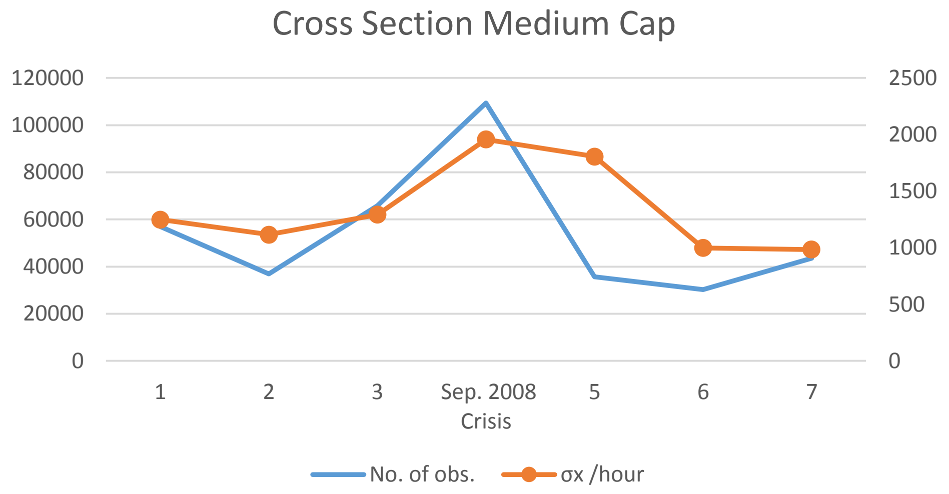
6. R^2_w = ratio of variance in trade-explained component of efficient price changes to total variance of efficient price changes

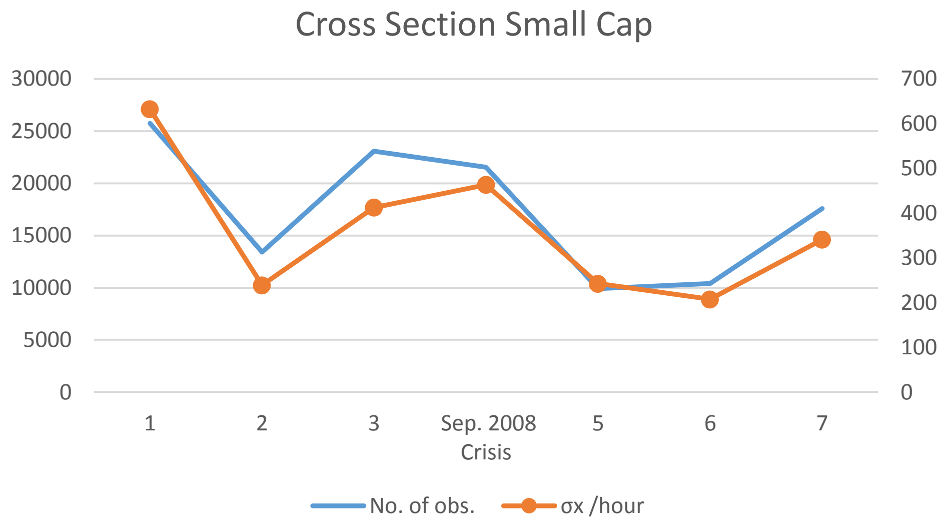
7. Spread = time-weighted average spread proportional to stock price

A7 Plots and Statistics Tables - LARGE, MEDIUM & SMALL CAP Stocks

A7 (a) - Plots







A7 (b) - LARGE CAP (Statistics)

Cross-period summary statistics for LARGE CAP stocks

	Pre Crisis Period			Crisis Period	Pre Crisis Period		
	07-11	07-11	07-11	15-19	13-17	06-10	05-09
	January 2008	April 2008	July 2008	September	April 2009	July 2009	October 2009
No. of firms	5			5			5
No. of obs.	369008	166989	235223	257971	164643	85253	106935
	[597759]	[234948]	[298885]	[221133]	[249961]	[100946]	[142880]
Impulse	0.01	0.008	0.011	0.013	0.01	0.009	0.008
(x100)	[0.007]	[0.008]	[0.011]	[0.008]	[0.009]	[0.007]	[0.006]
σ_x /event	7.026	7.381	5.626	5.167	5.317	6.44	8.482
	[7.434]	[6.839]	[5.097]	[4.959]	[4.731]	[5.663]	[10.260]
σ_x /hour	17901.546	10616.206	12996.723	22204.896	8014.626	6175.914	6795.047
	[9929.477]	[3716.631]	[4557.802]	[5879.599]	[4581.709]	[2141.326]	[2668.441]
σ_w /event	0.023	0.019	0.026	0.034	0.032	0.03	0.022
(x100)	[0.015]	[0.011]	[0.017]	[0.024]	[0.022]	[0.021]	[0.015]
σ_w /hour	100.724	45.641	78.959	179.751	53.592	34.742	26.777
(x100)	[95.191]	[36.317]	[46.281]	[73.537]	[35.503]	[15.960]	[15.760]
$\sigma_{w,x}$ /event	0.004	0.003	0.004	0.005	0.004	0.004	0.004
(x100)	[0.002]	[0.002]	[0.002]	[0.002]	[0.002]	[0.002]	[0.001]
$\sigma_{w,x}$ /hour	35.955	15.878	29.613	40.972	16.977	10.279	9.484
(x100)	[44.810]	[20.544]	[36.216]	[37.033]	[21.559]	[11.282]	[10.306]
R^2_w	0.093	0.085	0.118	0.06	0.084	0.082	0.102
	[0.124]	[0.110]	[0.155]	[0.061]	[0.108]	[0.103]	[0.113]
Spread	0.068	0.050	0.062	0.072	0.069	0.062	0.051
(x100)	[0.019]	[0.009]	[0.014]	[0.018]	[0.015]	[0.020]	[0.017]

Note: 1. Figures are average values for the total sample of 15 firms. Stanadard deviations are given in square brackets.

2. Impulse = implied impact of a 100-share buy order on the log(quote midpo

3. σ_x /hour = standard deviation of the innovation in x_t equation, a measure of trading intensity

4. σ_w /hour = standard deviation of change in the efficient price

5. $\sigma_{w,x}$ /hour = square root of the variance of trade-correlated part of the efficient price changes

6. R^2_w = ratio of variance in trade-explained component of efficient price changes to total variance of efficient price changes

7. Spread = time-weighted average spread proportional to stock price

A7 (b) - MEDIUM CAP (Statistics)

Cross-period summary statistics for MEDIUM CAP stocks

	Pre Crisis Period			Crisis Period	Pre Crisis Period		
	07-11	07-11	07-11	15-19	13-17	06-10	05-09
	January 2008	April 2008	July 2008	September	April 2009	July 2009	October 2009
No. of firms	5			5			5
No. of obs.	57008	36879	65794	109406	35685	30252	43530
	[28152]	[13584]	[28868]	[42687]	[8311]	[6429]	[16920]
Impulse	0.081	0.077	0.137	0.131	0.087	0.094	0.069
(x100)	[0.026]	[0.040]	[0.109]	[0.044]	[0.051]	[0.054]	[0.030]
σ_x /event	0.85	1.128	0.783	0.692	1.837	1.206	0.834
	[0.569]	[0.706]	[0.512]	[0.416]	[1.265]	[0.889]	[0.439]
σ_x /hour	1249.496	1115.648	1292.992	1957.137	1805.57	998.67	984.967
	[596.507]	[445.774]	[522.758]	[734.890]	[1012.460]	[505.327]	[437.684]
σ_w /event	0.027	0.028	0.027	0.051	0.048	0.041	0.026
(x100)	[0.003]	[0.007]	[0.007]	[0.018]	[0.028]	[0.022]	[0.006]
σ_w /hour	48.761	32.349	51.75	182.89	50.088	35.383	33.72
(x100)	[28.796]	[15.829]	[16.932]	[133.931]	[23.003]	[10.593]	[13.572]
$\sigma_{w,x}$ /event	0.01	0.01	0.011	0.013	0.016	0.014	0.009
(x100)	[0.004]	[0.002]	[0.003]	[0.004]	[0.011]	[0.008]	[0.003]
$\sigma_{w,x}$ /hour	17.971	11.368	20.189	41.085	16.861	12.246	11.26
(x100)	[10.750]	[4.151]	[7.439]	[7.969]	[8.829]	[3.803]	[6.152]
R^2_w	0.153	0.145	0.159	0.088	0.117	0.127	0.11
	[0.095]	[0.073]	[0.064]	[0.053]	[0.041]	[0.044]	[0.046]
Spread	0.198	0.154	0.178	0.247	0.200	0.165	0.123
(x100)	[0.086]	[0.069]	[0.050]	[0.074]	[0.106]	[0.065]	[0.065]

Note: 1. Figures are average values for the total sample of 15 firms. Stanadard deviations are given in square brackets.

2. Impulse = implied impact of a 100-share buy order on the log(quote midpo

3. σ_x /hour = standard deviation of the innovation in x_t equation, a measure of trading intensity

4. σ_w /hour = standard deviation of change in the efficient price

5. $\sigma_{w,x}$ /hour = square root of the variance of trade-correlated part of the efficient price changes

6. R^2_w = ratio of variance in trade-explained component of efficient price changes to total variance of efficient price changes

7. Spread = time-weighted average spread proportional to stock price

A7 (b) – SMALL CAP (Statistics)

Cross-period summary statistics for SMALL CAP stocks

	Pre Crisis Period			Crisis Period	Pre Crisis Period		
	07-11	07-11	07-11	15-19	13-17	06-10	05-09
	January 2008	April 2008	July 2008	September	April 2009	July 2009	October 2009
No. of firms	5			5			5
No. of obs.	25736	13395	23061	21528	9907	10400	17581
	[27969]	[11059]	[18054]	[19591]	[8519]	[6895]	[8357]
Impulse	0.165	0.229	0.257	0.377	0.492	0.338	0.346
(x100)	[0.061]	[0.132]	[0.078]	[0.258]	[0.253]	[0.126]	[0.252]
σ_x /event	1.226	0.775	0.769	1.104	1.047	0.878	0.685
	[0.649]	[0.672]	[0.373]	[0.725]	[0.474]	[0.490]	[0.457]
σ_x/hour	632.279	238.756	412.448	463.235	242.611	207.331	341.019
	[261.079]	[157.075]	[164.782]	[134.474]	[122.334]	[77.047]	[310.023]
σ_w/event	0.078	0.107	0.087	0.151	0.21	0.153	0.083
(x100)	[0.030]	[0.062]	[0.032]	[0.057]	[0.108]	[0.047]	[0.042]
σ_w/hour	49.04	33.142	51.634	82.208	53.626	45.671	43.014
(x100)	[36.079]	[12.697]	[30.574]	[59.557]	[35.283]	[28.080]	[25.953]
$\sigma_{w,x}$ /event	0.023	0.037	0.034	0.04	0.077	0.043	0.025
(x100)	[0.006]	[0.024]	[0.014]	[0.012]	[0.053]	[0.012]	[0.011]
$\sigma_{w,x}$ /hour	15.513	11.014	19.556	22.63	19.195	12.365	12.752
(x100)	[12.280]	[4.691]	[10.983]	[15.866]	[15.672]	[6.867]	[6.220]
R^2_w	0.106	0.122	0.149	0.081	0.119	0.083	0.111
	[0.054]	[0.082]	[0.025]	[0.036]	[0.048]	[0.026]	[0.065]
Spread	0.412	0.432	0.417	0.693	1.242	0.854	0.536
(x100)	[0.139]	[0.092]	[0.048]	[0.031]	[0.624]	[0.327]	[0.360]

Note: 1. Figures are average values for the total sample of 15 firms. Stanadard deviations are given in square brackets.

2. Impulse = implied impact of a 100-share buy order on the log(quote midpo

3. σ_x /hour = standard deviation of the innovation in x_t equation, a measure of trading intensity

4. σ_w /hour = standard deviation of change in the efficient price

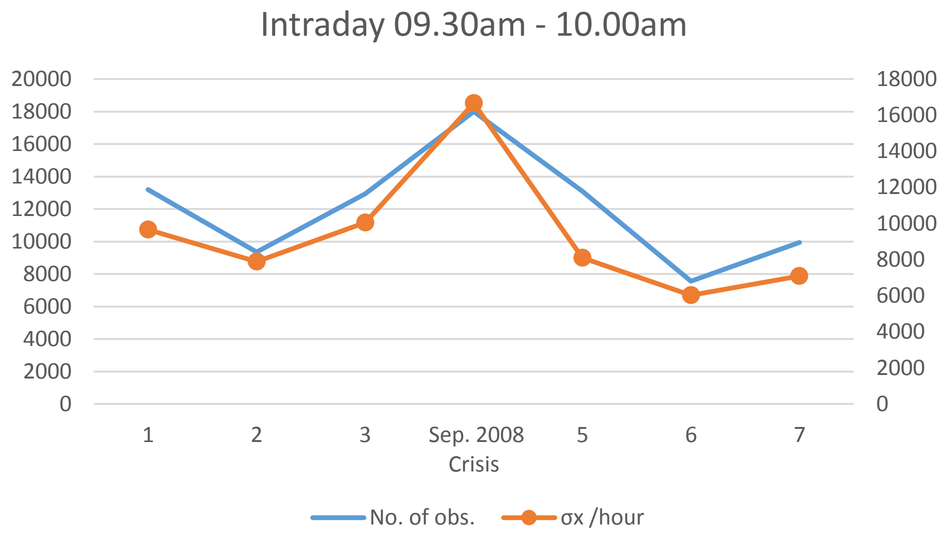
5. $\sigma_{w,x}$ /hour = square root of the variance of trade-correlated part of the efficient price changes

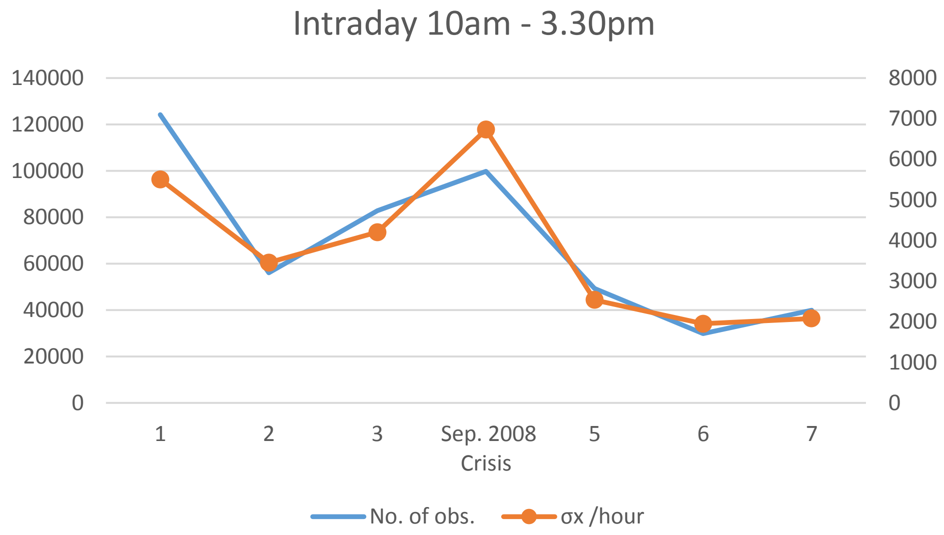
6. R^2_w = ratio of variance in trade-explained component of efficient price changes to total variance of efficient price changes

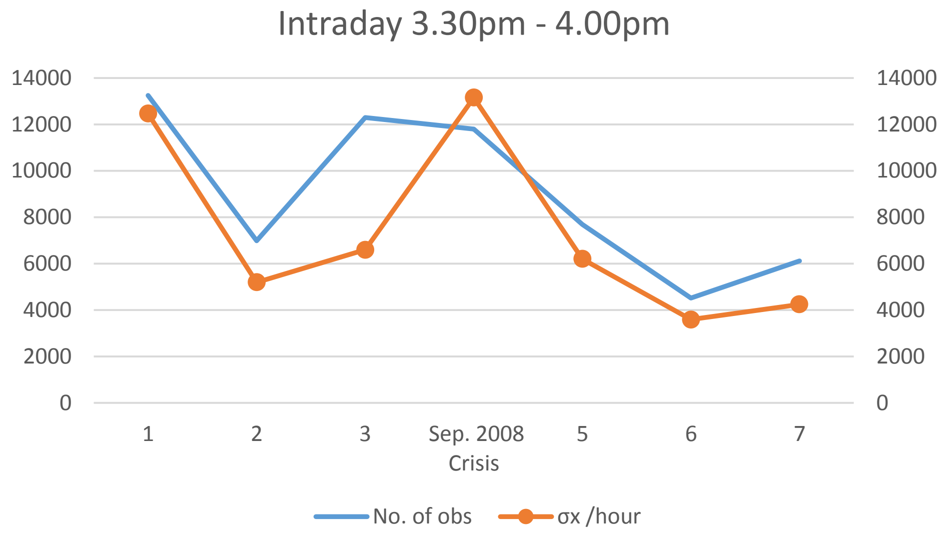
7. Spread = time-weighted average spread proportional to stock price

A8 Plots and Statistics Tables – Beginning, Middle & End of Day

A8 (a) - Plots







A8 (b) – Intraday Beginning (Statistics)

Cross-period summary statistics for 15 stocks 9.30 A.M. - 10.00 A.M.

	Pre Crisis Period			Crisis Period	Pre Crisis Period		
	07-11	07-11	07-11	15-19	13-17	06-10	05-09
	January 2008	April 2008	July 2008	September	April 2009	July 2009	October 2009
No. of firms							
No. of obs.	13187	9351	12936	18001	13103	7556	9949
	[23386]	[16221]	[18592]	[19586]	[28743]	[7950]	[14277]
Impulse	-0.2	0.512	0.14	0.783	1.801	0.657	0.856
(x100)	[1.687]	[1.277]	[1.964]	[1.543]	[4.151]	[1.108]	[2.047]
σ_x /event	2.212	2.191	1.926	1.834	2.028	2.139	2.721
	[3.438]	[3.860]	[3.205]	[2.956]	[2.188]	[3.215]	[6.256]
σ_x /hour	9661.574	7891.189	10058.088	16668.386	8104.308	6029.558	7091.909
	[13952.104]	[12725.591]	[15559.105]	[24698.683]	[11159.780]	[8029.672]	[10736.036]
σ_w /event	0.111	0.135	0.102	0.202	0.187	0.14	0.094
(x100)	[0.104]	[0.146]	[0.080]	[0.152]	[0.203]	[0.149]	[0.087]
σ_w /hour	205.08	156.382	231.132	685.108	248.249	173.918	167.346
(x100)	[126.149]	[67.792]	[118.781]	[543.511]	[140.800]	[106.323]	[91.623]
$\sigma_{w,x}$ /event	0.044	0.063	0.04	0.072	0.065	0.048	0.036
(x100)	[0.061]	[0.083]	[0.052]	[0.086]	[0.088]	[0.061]	[0.045]
$\sigma_{w,x}$ /hour	69.925	57.845	75.792	156.809	67.724	51.624	45.706
(x100)	[49.749]	[37.483]	[71.374]	[112.732]	[35.924]	[29.008]	[22.555]
R^2_w	0.138	0.17	0.125	0.109	0.106	0.11	0.119
	[0.144]	[0.171]	[0.129]	[0.125]	[0.078]	[0.077]	[0.128]
Spread	0.825	0.689	0.559	1.461	0.956	0.802	0.533
(x100)	[0.991]	[0.738]	[0.498]	[1.576]	[1.125]	[0.891]	[0.582]

Note: 1. Figures are average values for the total sample of 15 firms. Stanadard deviations are given in square brackets.

2. Impulse = implied impact of a 100-share buy order on the log(quote midpo

3. σ_x /hour = standard deviation of the innovation in x_t equation, a measure of trading intensity

4. σ_w /hour = standard deviation of change in the efficient price

5. $\sigma_{w,x}$ /hour = square root of the variance of trade-correlated part of the efficient price changes

6. R^2_w = ratio of variance in trade-explained component of efficient price changes to total variance of efficient price changes

7. Spread = time-weighted average spread proportional to stock price

A8 (b) – Intraday Middle (Statistics)

Cross-period summary statistics for 15 stocks 10.00 A.M. - 3.30 P.M.

	Pre Crisis Period			Crisis Period	Pre Crisis Period		
	07-11	07-11	07-11	15-19	13-17	06-10	05-09
	January 2008	April 2008	July 2008	September	April 2009	July 2009	October 2009
No. of firms							
No. of obs.	124145	56077	82798	99839.933	49280	29898	39951.733
	[306761]	[115610]	[147548]	[127944]	[106234]	[47839]	[63381]
Impulse	0.096	0.176	0.145	0.153	0.206	0.149	0.167
(x100)	[0.080]	[0.344]	[0.132]	[0.140]	[0.260]	[0.164]	[0.267]
σ_x /event	2.985	3.299	2.439	2.335	2.808	2.972	3.474
	[5.047]	[5.280]	[3.757]	[3.499]	[3.657]	[4.470]	[6.848]
σ_x /hour	5503.737	3453.546	4203.332	6728.142	2536.039	1949.064	2079.891
	[7833.747]	[4525.044]	[5453.003]	[8769.403]	[3265.525]	[2388.982]	[2532.725]
σ_w /event	0.039	0.044	0.044	0.066	0.097	0.072	0.046
(x100)	[0.023]	[0.041]	[0.032]	[0.063]	[0.105]	[0.065]	[0.036]
σ_w /hour	61.737	30.493	53.491	98.588	46.734	30.773	32.292
(x100)	[63.451]	[22.532]	[35.021]	[54.071]	[33.517]	[15.165]	[19.824]
$\sigma_{w,x}$ /event	0.012	0.014	0.016	0.017	0.034	0.021	0.013
(x100)	[0.010]	[0.020]	[0.016]	[0.016]	[0.045]	[0.021]	[0.010]
$\sigma_{w,x}$ /hour	21.414	10.546	20.842	29.012	15.266	10.09	10.052
(x100)	[25.864]	[10.725]	[20.122]	[22.315]	[12.417]	[6.818]	[7.046]
R^2_w	0.121	0.118	0.15	0.094	0.111	0.111	0.104
	[0.076]	[0.089]	[0.096]	[0.068]	[0.064]	[0.071]	[0.071]
Spread	0.187	0.169	0.191	0.264	0.474	0.327	0.218
(x100)	[0.125]	[0.139]	[0.131]	[0.207]	[0.604]	[0.361]	[0.282]

Note: 1. Figures are average values for the total sample of 15 firms. Stanadard deviations are given in square brackets.

2. Impulse = implied impact of a 100-share buy order on the log(quote midpo

3. σ_x /hour = standard deviation of the innovation in x_t equation, a measure of trading intensity

4. σ_w /hour = standard deviation of change in the efficient price

5. $\sigma_{w,x}$ /hour = square root of the variance of trade-correlated part of the efficient price changes

6. R^2_w = ratio of variance in trade-explained component of efficient price changes to total variance of efficient price changes

7. Spread = time-weighted average spread proportional to stock price

A8 (b) – Intraday End (Statistics)

Cross-period summary statistics for 15 stocks 3.30 P.M. - 4.00 P.M.

	Pre Crisis Period			Crisis Period	Pre Crisis Period		
	07-11	07-11	07-11	15-19	13-17	06-10	05-09
	January 2008	April 2008	July 2008	September	April 2009	July 2009	October 2009
No. of firms							
No. of obs	13250	6990	12291	11791.867	7693	4513	6112.2
	[28928]	[12446]	[20908]	[14614]	[16173]	[7913]	[9896]
Impulse	0.283	0.328	0.185	0.242	0.345	0.124	0.207
(x100)	[0.813]	[0.691]	[0.270]	[0.480]	[0.511]	[0.205]	[0.507]
σ_x /event	4.298	3.354	2.846	3.176	3.556	3.396	4.451
	[7.262]	[4.535]	[3.943]	[4.347]	[4.564]	[4.347]	[7.806]
σ_x/hour	12464.523	5196.731	6596.676	13154.726	6203.816	3589.333	4244.133
	[25558.298]	[7203.837]	[9454.503]	[17551.484]	[8431.951]	[4732.116]	[6274.471]
σ_w/event	0.089	0.143	0.085	0.131	0.182	0.154	0.115
(x100)	[0.064]	[0.153]	[0.056]	[0.120]	[0.212]	[0.136]	[0.080]
σ_w/hour	180.854	131.769	159.94	280.028	151.211	100.698	105.805
(x100)	[176.912]	[95.984]	[120.019]	[185.486]	[109.708]	[71.844]	[83.876]
$\sigma_{w,x}$ /event	0.03	0.073	0.024	0.037	0.07	0.049	0.038
(x100)	[0.029]	[0.135]	[0.023]	[0.047]	[0.105]	[0.074]	[0.033]
$\sigma_{w,x}$ /hour	58.087	37.045	38.378	57.529	39.77	22.226	24.825
(x100)	[80.081]	[34.398]	[28.105]	[37.001]	[33.618]	[17.444]	[18.657]
R^2_w	0.109	0.175	0.072	0.065	0.128	0.099	0.112
	[0.108]	[0.244]	[0.052]	[0.057]	[0.140]	[0.113]	[0.110]
Spread	0.165	0.136	0.184	0.245	0.365	0.237	0.147
(x100)	[0.115]	[0.095]	[0.135]	[0.188]	[0.460]	[0.261]	[0.134]

Note: 1. Figures are average values for the total sample of 15 firms. Stanadard deviations are given in square brackets.

2. Impulse = implied impact of a 100-share buy order on the log(quote midpo

3. σ_x /hour = standard deviation of the innovation in x_t equation, a measure of trading intensity

4. σ_w /hour = standard deviation of change in the efficient price

5. $\sigma_{w,x}$ /hour = square root of the variance of trade-correlated part of the efficient price changes

6. R^2_w = ratio of variance in trade-explained component of efficient price changes to total variance of efficient price changes

7. Spread = time-weighted average spread proportional to stock price

Appendix B – Steps Solving for Structural Matrix A₀

Start with $$\Omega_\epsilon = \begin{bmatrix} S_{11} & S_{21} & S_{31} & S_{41} \\ S_{21} & S_{22} & S_{32} & S_{42} \\ S_{31} & S_{32} & S_{33} & S_{43} \\ S_{41} & S_{42} & S_{43} & S_{44} \end{bmatrix},$$

Where Ω_ϵ is a symmetric positive(semi-) definite covariance matrix of shocks (or innovations) of the reduce-form Vector Autoregression (VAR) model.

Then define the structural matrix A_0 and covariance matrix of structural shocks of the structural VAR (SVAR) model Ω_v.

$$A_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ A_{041} & A_{042} & A_{043} & 1 \end{bmatrix}, \Omega_v = \begin{bmatrix} D_{11} & 0 & \dots & 0 \\ 0 & D_{22} & 0 & \dots \\ \dots & \dots & D_{33} & 0 \\ 0 & \dots & 0 & D_{44} \end{bmatrix},$$

Where D_{11} is the variance of innovation of signed trade volume x_t^0, D_{22} the variance of innovation of signed trade variable x_t, D_{33} the variance of innovation of signed squared trade variable x_t^2, D_{44} the variance of innovation of quote revision r_t.

Note that since $\phi_j = A_0^{-1}A_j$ as stated in the footnote 4, we have $A_j = A_0\phi_j$. But we also have $V_\varepsilon = A_0^{-1}V$, were V is the column vector of structural shocks in equation (7) in section 3, and V_ε is the vector of reduce-form VAR shocks. The corresponding relationship between the covariance matrix of reduced-form VAR shocks and structural shocks is shown below, which leads to systems of linear equations used to solve for the 7 unknowns: $A_{041}, A_{042}, A_{043}, D_{11}, D_{22}, D_{33}, D_{44}$.

$$\Omega_\epsilon = A_0^{-1}\Omega_v(A_0^{-1})^T$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ A_{041} & A_{042} & A_{043} & 1 \end{bmatrix} \begin{bmatrix} D_{11} & \dots & \dots & 0 \\ 0 & D_{22} & 0 & \dots \\ \dots & \dots & D_{33} & 0 \\ 0 & \dots & \dots & D_{44} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -A_{041} \\ 0 & 1 & 0 & -A_{042} \\ 0 & 0 & 1 & -A_{043} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} D_{11} & 0 & 0 & 0 \\ 0 & D_{22} & 0 & 0 \\ 0 & 0 & D_{33} & 0 \\ -D_{11}A_{041} & -D_{22}A_{042} & -D_{33}A_{043} & D_{44} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -A_{041} \\ 0 & 1 & 0 & -A_{042} \\ 0 & 0 & 1 & -A_{043} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} D_{11} & 0 & 0 & -A_{041} \\ 0 & D_{22} & 0 & -A_{042} \\ 0 & 0 & D_{33} & -A_{043} \\ -D_{11}A_{041} & -D_{22}A_{042} & -D_{33}A_{043} & D_{11}A_{041}^2 + D_{22}A_{042}^2 + D_{33}A_{043}^2 + D_{44} \end{bmatrix}$$

Solving for the systems gives the following solution sets, which can be used in Matlab to generate the coefficient estimates of SVAR model:

$$\begin{cases} D_{11} = S_{11} \\ D_{22} = S_{22} \\ D_{33} = S_{33} \\ D_{44} = S_{44} - S_{11}A_{041}^2 - S_{22}A_{042}^2 - S_{33}A_{043}^2 \end{cases}$$

$$\begin{cases} A_{041} = -\dfrac{S_{41}}{S_{11}} \\ A_{042} = -\dfrac{S_{42}}{D_{22}} \\ A_{043} = -\dfrac{S_{43}}{D_{33}} \end{cases}$$

Appendix C – Matlab Codes

C1 – DataCleanser

```
%1. The function converts milliseconds-from-midnight time stamp format
%   to military time format for trades data
%2. Note that the input needs to be in 'raw data' form
%3. The output is a column vector of converted time stamp

function MilitaryTimeConverted = TimeFormatConverter(trade_raw_data)

LengthTradeRaw = length(trade_raw_data);

milliseconds   = zeros(LengthTradeRaw-1,1);
%
for i = 2:LengthTradeRaw
    milliseconds(i-1,1) = trade_raw_data{i,1};
end

TimeInHour   = milliseconds/3600000;
HourInteger  = floor(TimeInHour);
HourDecimal  = TimeInHour - HourInteger;
TimeInMinute = HourDecimal*60;
MinuteInteger= floor(TimeInMinute);
MinuteDecimal= TimeInMinute - MinuteInteger;
Milliseconds = round(MinuteDecimal*60*1000);
MilitaryTimeConverted = HourInteger*10000000 + MinuteInteger*100000
+...
    + Milliseconds;

end


%1. This function removes leading zeros in military time string for
quotes data
%2. Note that the input needs to be in 'raw data' form
%3. The output is a column vector of transformed time stamp

function military_time = LeadingZeroRemover(quote_raw_data)

military_time  = zeros(length(quote_raw_data)-1,1);

for i = 2:length(quote_raw_data)
    military_time(i-1)  = str2double(quote_raw_data{i,2});
end

end


%1. This function checks if the two input variables have the same
signs, i.e.
%   both positive or both negative
%2. It returns 1 if the two input variables have the same sign,
%              0 if not

function y = IsSameSign(variable1, variable2)

if variable1>0 && variable2>0 || variable1<0 && variable2<0
```

```
    y = 1;
else
    y = 0;
end
end


%1. This function adds a sign to each trade, with '+' for purchase and
'-'
%   for sell
%2. Note that the input needs to be in 'raw data' form

function MilitaryTimeData = TradeSignAdder(trade_raw_data)

LengthTradeRaw   = length(trade_raw_data);

MilitaryTimeData = TimeFormatConverter(trade_raw_data);
LengthData       = numel(MilitaryTimeData);

for i = 2:LengthTradeRaw
    MilitaryTimeData(i-1,2) = trade_raw_data{i,3};
end

for i = 1:LengthData
    if strcmpi(trade_raw_data{i+1,4},'S') % string comparison
        MilitaryTimeData(i,2) = -MilitaryTimeData(i,2);
    end
end
end


%1. This function removes the trade and quote observations occurred
before
%   9.30am and after 4.00pm, at which market opens and closes
respectively
%2. If trades data enters the function, datatype = 0
%   If quotes data enters the function, datatype = 1
%3. Note that the input needs to be in 'raw data' form

function output = TradingTimeData(raw_data,datatype)

% for quotes data
if datatype == 1
    LengthQuoteRaw   = length(raw_data);
    MilitaryTimeData = LeadingZeroRemover(raw_data);
    LengthData       = numel(MilitaryTimeData);
    output           = zeros(LengthData,3);
    output(:,1)      = MilitaryTimeData;

    for i = 2:LengthQuoteRaw
        output(i-1,2) = raw_data{i,3}; % bid quote
        output(i-1,3) = raw_data{i,4}; % ask quote
    end

    for start = 1:LengthData
        if output(start,1) >= 93000000
            break % include the quote observation AT 9.30am by using
'>='
        end
    end

    for terminal = 1:LengthData
        if terminal ~= LengthData
            if output(terminal,1) >= 160000000
                output = output(start:terminal-1,:);
                break
            end
        else % the case when terminal runs to the END of the series
            if output(terminal,1) >= 160000000
                output = output(start:terminal-1,:);
            else % include the LAST observation if it occurs BEFORE
4.00pm
                output = output(start:terminal,:);
            end
        end
    end

    %for trades data
elseif datatype == 0
    MilitaryTimeData = TradeSignAdder(raw_data); % add sign to each
trade
    LengthData       = length(MilitaryTimeData);

    for start = 1:LengthData
        if MilitaryTimeData(start,1) > 93000000
            break % do NOT include the trade observation AT 9.30am
        end
    end

    for terminal = 1:LengthData
        if terminal ~= LengthData
            if MilitaryTimeData(terminal,1) >= 160000000
                output = MilitaryTimeData(start:terminal-1,:);
                break
            end
        else % the case when terminal runs to the END of the series
            if MilitaryTimeData(terminal,1) >= 160000000
                output = MilitaryTimeData(start:terminal-1,:);
            else % include the LAST observation if it occurs BEFORE
4.00pm
                output = MilitaryTimeData(start:terminal,:);
            end
        end
    end

else
    error('datatype should be either 1 or 0')

end

end


%1. This function cumulates trades that occur at the same time with
the
%   SAME signs as a single observation
%2. Note that the input needs to be in 'raw data' form

function TradeCombined = TradeCumulater(trade_raw_data)
```

```
TradeData            = TradingTimeData(trade_raw_data,0);
LengthData           = length(TradeData);

TradeCombined        = zeros(length(TradeData),2); %preallocation

TradeCombined(1,:) = TradeData(1,:);

j = 1;

for i = 2:LengthData
    if TradeData(i,1) == TradeData(i-1,1) && ...
            IsSameSign(TradeData(i,2),TradeData(i-1,2)) == 1
        TradeCombined(j,2) = TradeCombined(j,2) + TradeData(i,2);
        TradeCombined(j,1) = TradeData(i,1);
    else
        j = j + 1;
        TradeCombined(j,2) = TradeData(i,2);
        TradeCombined(j,1) = TradeData(i,1);
    end
end

TradeCombined(j+1:end,:) = []; %variable size reduction
end


%1. This Function removes the quote duplicates
%2. If, after a quote revision, the subsequent quotes remain unchanged
as
%   time goes by, the initial quote revision is kept and the rest
quote
%   duplicates are remove
%3. Note that the input needs to be in 'raw data' form

function DataNoDuplicates = QuoteDuplicatesRemover(quote_raw_data)

QuoteData               = TradingTimeData(quote_raw_data,1);
LengthData              = length(QuoteData);

DataNoDuplicates        = zeros(length(QuoteData),3); %preallocation

DataNoDuplicates(1,:) = QuoteData(1,:);

j = 2;

for i = 2:LengthData
    if QuoteData(i,2)~= QuoteData(i-1,2) ||...
            QuoteData(i,3) ~= QuoteData(i-1,3)

        DataNoDuplicates(j,:) = QuoteData(i,:);
        j = j + 1;
    end
end

DataNoDuplicates(j:end,:) = []; %variable size reduction
end
```

```
%1. This functions calculates the mid point of quoted bid and ask
spread
%2. Note that the input needs to be in 'raw data' form

function LogQuoteMidpoint = QuoteMidpointCalculator(quote_raw_data)

LogQuoteMidpoint     = QuoteDuplicatesRemover(quote_raw_data);

LogQuoteMidpoint(:,2)=
log(0.5*(LogQuoteMidpoint(:,2)+LogQuoteMidpoint(:,3)));

LogQuoteMidpoint(:,3)= [];
end


%1. This function performs the 'event count' based on the approach
stated
%   in Hasbrouck(1991b)
%2. Note that the input needs to be in 'raw data' form
%3. The each trade volume in the trade series is assigned either as a
buy
%   or sell based on REAL TIME data

function [TradeEventCount, QuoteEventCount] ...
    = EventCounter(trade_raw_data, quote_raw_data)

% cleanse the data applying the user defined functions
TradeCombined    = TradeCumulater(trade_raw_data);
LogQuoteMidpoint = QuoteMidpointCalculator(quote_raw_data);
LengthMidpoint   = length(LogQuoteMidpoint);
LengthTradeCom   = length(TradeCombined);

TradeEventCount  = zeros(LengthMidpoint,1);
counter          = 1;
i                = 1;

while i <= LengthMidpoint
    if LogQuoteMidpoint(i,1) == 93000000
        TradeEventCount(i) = 0;

        %if the program runs to the end of the trade series but there
are
        %still more entries in the quote series to be gone through,
then
        %set the remaining empty entries in TradeEventCount to be 0
    elseif counter > LengthTradeCom;
        TradeEventCount(i:end) = 0;

    elseif TradeCombined(counter,1) <= LogQuoteMidpoint(i,1)
        TradeEventCount(i) = TradeCombined(counter,2);
        counter = counter + 1;  % the counter moves 1 step forward
only if
        % we copy one trade volume data from TradeCombined to
TradeEventCount
    else
        TradeEventCount(i) = 0;
    end
    i = i + 1;
end
```

```
QuoteEventCount = LogQuoteMidpoint(:,2);

end


%1. This function combines five days trading data together to generate
a
%   complete and usable series ready to perform the VAR estimation
%2. Note that the input needs to be in 'raw data' form
%3. Type = 'real' if the output trade series is real-time-data signed
%        = 'approx' if the output trade series is approximated signed
%4. The output variables are also stored in a .mat file (in a struc
%   structure) in the current folder with user specified filename; use
''
%   as placeholder if the user does not save the outputs
%5. The following outputs can be generated if type = 'approx'
%   5.1. 'average' is 6-by-2 cell array that contains:
%   a. apcor  = average % of correct number of approximations across 5
trading days
%   b. apsign = average % of trades assigned as buys/sells across 5
trading days
%   c. apb    = average % of trades assigned as buys across 5 trading
days
%   d. aps    = average % of trades assigned as sells across 5 trading
days
%   e. arb    = average real % of buy trades across 5 trading days
%   e. arb    = average real % of sell trades across 5 trading days
%   5.2. 'individual' is 6-by-7 cell array that contains pcor, psign,
pb,
%   ps, rb, rs figures for each trading day
%6. If Save = 'large' the figure is stored in current holder with name
'figure_large.png'
%           = 'medium' the figure is stored in current holder with
name 'figure_medium.png'
%           = 'small' the figure is stored in current holder with name
'figure_small.png'
%7. legendname should be in string format
%8. SumStatsTrade,SumStatsQuote are summary statistics of the number
of
%   trades and quotes each trading day using cleansed data

function
[CombinedTradeData,CombinedQuoteData,SumStatsTrade,SumStatsQuote,...
    average,individual] ...
    = DataCombiner(trade_raw_data_day1, quote_raw_data_day1,...
    trade_raw_data_day2, quote_raw_data_day2,...
    trade_raw_data_day3, quote_raw_data_day3,...
    trade_raw_data_day4, quote_raw_data_day4,...
    trade_raw_data_day5,
quote_raw_data_day5,type,filename,legendname,Save)

% Case 1. for real data signed input
if strcmpi('real',type)

    % cleanse the data for each individual trading day
    [TradeEventCountDay1,QuoteEventCountDay1] ...
        = EventCounter(trade_raw_data_day1, quote_raw_data_day1);
    [TradeEventCountDay2,QuoteEventCountDay2] ...
        = EventCounter(trade_raw_data_day2, quote_raw_data_day2);
    [TradeEventCountDay3,QuoteEventCountDay3] ...
        = EventCounter(trade_raw_data_day3, quote_raw_data_day3);
    [TradeEventCountDay4,QuoteEventCountDay4] ...
        = EventCounter(trade_raw_data_day4, quote_raw_data_day4);
    [TradeEventCountDay5,QuoteEventCountDay5] ...
        = EventCounter(trade_raw_data_day5, quote_raw_data_day5);

    % Case 2. for approximated data signed input
elseif strcmpi('approx',type)
    % cleanse the data for each individual trading day
    [TradeEventCountDay1,QuoteEventCountDay1,pcor1,psign1,pb1,ps1] ...
        = EventCounterApprox(trade_raw_data_day1,
quote_raw_data_day1);
    [TradeEventCountDay2,QuoteEventCountDay2,pcor2,psign2,pb2,ps2] ...
        = EventCounterApprox(trade_raw_data_day2,
quote_raw_data_day2);
    [TradeEventCountDay3,QuoteEventCountDay3,pcor3,psign3,pb3,ps3] ...
        = EventCounterApprox(trade_raw_data_day3,
quote_raw_data_day3);
    [TradeEventCountDay4,QuoteEventCountDay4,pcor4,psign4,pb4,ps4] ...
        = EventCounterApprox(trade_raw_data_day4,
quote_raw_data_day4);
    [TradeEventCountDay5,QuoteEventCountDay5,pcor5,psign5,pb5,ps5] ...
        = EventCounterApprox(trade_raw_data_day5,
quote_raw_data_day5);

    % calculate the average pcor, psign, pb and ps value
    apcor  = (pcor1 + pcor2 + pcor3 + pcor4 + pcor5)/5;
    apsign = (psign1 + psign2 + psign3 + psign4 + psign5)/5;
    apb    = (pb1 + pb2 + pb3 + pb4 + pb5)/5;
    aps    = (ps1 + ps2 + ps3 + ps4 + ps5)/5;

    % calculate the actual percentage of buy or sell on each trading
data based
    % on real time data
    LengthTradeDay1 = length(trade_raw_data_day1);
    LengthTradeDay2 = length(trade_raw_data_day2);
    LengthTradeDay3 = length(trade_raw_data_day3);
    LengthTradeDay4 = length(trade_raw_data_day4);
    LengthTradeDay5 = length(trade_raw_data_day5);

    % day1
    countbuy  = 0;
    countsell = 0;
    for i = 2:LengthTradeDay1
        if strcmpi(trade_raw_data_day1{i,4},'B')
            countbuy = countbuy + 1;
        else
            countsell = countsell + 1;
        end
    end
    rb1  = countbuy/(LengthTradeDay1-1);
    rs1 = countsell/(LengthTradeDay1-1);
    % day2
    countbuy  = 0;
    countsell = 0;
    for i = 2:LengthTradeDay2
        if strcmpi(trade_raw_data_day2{i,4},'B')
            countbuy = countbuy + 1;
        else
            countsell = countsell + 1;
        end
    end
    rb2  = countbuy/(LengthTradeDay2-1);
    rs2 = countsell/(LengthTradeDay2-1);
    % day3
    countbuy  = 0;
    countsell = 0;
    for i = 2:LengthTradeDay3
        if strcmpi(trade_raw_data_day3{i,4},'B')
            countbuy = countbuy + 1;
        else
            countsell = countsell + 1;
        end
    end
    rb3  = countbuy/(LengthTradeDay3-1);
    rs3 = countsell/(LengthTradeDay3-1);
    % day4
    countbuy  = 0;
    countsell = 0;
    for i = 2:LengthTradeDay4
        if strcmpi(trade_raw_data_day4{i,4},'B')
            countbuy = countbuy + 1;
        else
            countsell = countsell + 1;
        end
    end
    rb4  = countbuy/(LengthTradeDay4-1);
    rs4 = countsell/(LengthTradeDay4-1);
    % day5
    countbuy  = 0;
    countsell = 0;
    for i = 2:LengthTradeDay5
        if strcmpi(trade_raw_data_day5{i,4},'B')
            countbuy = countbuy + 1;
        else
            countsell = countsell + 1;
        end
    end
    rb5  = countbuy/(LengthTradeDay5-1);
    rs5 = countsell/(LengthTradeDay5-1);
    % compute the average
    arb = (rb1+rb2+rb3+rb4+rb5)/5;
    ars = (rs1+rs2+rs3+rs4+rs5)/5;

    average      = cell(6,2);
    average(:,1) = {'apcor';'apsign';'apb';'aps';'arb';'ars'};
    average(:,2) = {apcor;apsign;apb;aps;arb;ars};

    individual          = cell(6,7);
    individual(2:6,1)     = {'day1';'day2';'day3';'day4';'day5'};
    individual(1,2:7)   = {'pcor','psign','pb','ps','rb','rs'};
    individual(2:6,2:7) = {pcor1,psign1,pb1,ps1,rb1,rs1;...
        pcor2,psign2,pb2,ps2,rb2,rs2;pcor3,psign3,pb3,ps3,rb3,rs3;...
        pcor4,psign4,pb4,ps4,rb4,rs4;pcor5,psign5,pb5,ps5,rb5,rs5};

    % Graphics:
    % plot five days pcor, psign, pb and ps
    plot_pcor  = [pcor1, pcor2, pcor3, pcor4, pcor5];
    plot_psign = [psign1, psign2, psign3, psign4, psign5];
    plot_pb    = [pb1, pb2, pb3, pb4, pb5];
```

```
    plot_ps    = [ps1, ps2, ps3, ps4, ps5];
    plot_actual_buy ...
        = [rb1,rb2,rb3,rb4,rb5];
    plot_actual_sell ...
        = [rs1,rs2,rs3,rs4,rs5];

    % define the position of legend on the screen (note that it may
differ in
    % different sized computer screen)
    PositionCell    = cell(4,1);
    PositionCell(1) = {[.433,.895,.02,.05]};
    PositionCell(2) = {[.866,.895,.02,.05]};
    PositionCell(3) = {[.433,.43,.02,.01]};
    PositionCell(4) = {[.866,.43,.02,.01]};

    figure
    subplot(2,2,1),plot(plot_pcor);
    title('PCOR'), xlabel('trading day'), ylabel('percentage')
    leg=legend(legendname);
    set(leg,'Position',PositionCell{1})

    subplot(2,2,2),plot(plot_psign);
    title('PSIGN'), xlabel('trading day'), ylabel('percentage')
    leg=legend(legendname);
    set(leg,'Position',PositionCell{2})

    % actual percentage of buy/sell are plotted against the
approximated ones
    subplot(2,2,3),plot(1:5,plot_pb,1:5,plot_actual_buy,'r');
    title('PB'), xlabel('trading day'), ylabel('percentage')
    leg=legend(strcat(legendname,' apx'),strcat(legendname,' real'));
    set(leg,'Position',PositionCell{3})

    subplot(2,2,4),plot(1:5,plot_ps,1:5,plot_actual_sell,'r');
    title('PS'), xlabel('trading day'), ylabel('percentage')
    leg=legend(strcat(legendname,' apx'),strcat(legendname,' real'));
    set(leg,'Position',PositionCell{4})

    % save the figure in .png format
    if strcmpi(Save,'') == 0
        if strcmpi(Save,'large')
            print -dpng -r500 figure_large
        elseif strcmpi(Save,'medium')
            print -dpng -r500 figure_medium
        elseif strcmpi(Save,'small')
            print -dpng -r500 figure_small
        end
    end

else
    error('type should be either real or approx')

end

% calculate the total length of the combined series
LengthCumsumTrade = cumsum([length(TradeEventCountDay1), ...
    length(TradeEventCountDay2), length(TradeEventCountDay3), ...
    length(TradeEventCountDay4), length(TradeEventCountDay5)]);
LengthCumsumQuote = cumsum([length(QuoteEventCountDay1), ...
    length(QuoteEventCountDay2), length(QuoteEventCountDay3), ...
    length(QuoteEventCountDay4), length(QuoteEventCountDay5)]);

% preallocation
CombinedTradeData = zeros(LengthCumsumTrade(end),1);
CombinedQuoteData = zeros(LengthCumsumQuote(end),1);

% trade series
CombinedTradeData(1:LengthCumsumTrade(1)) = TradeEventCountDay1;
CombinedTradeData(LengthCumsumTrade(1)+1:LengthCumsumTrade(2)) ...
    = TradeEventCountDay2;
CombinedTradeData(LengthCumsumTrade(2)+1:LengthCumsumTrade(3)) ...
    = TradeEventCountDay3;
CombinedTradeData(LengthCumsumTrade(3)+1:LengthCumsumTrade(4)) ...
    = TradeEventCountDay4;
CombinedTradeData(LengthCumsumTrade(4)+1:LengthCumsumTrade(5)) ...
    = TradeEventCountDay5;

% quote series
CombinedQuoteData(1:LengthCumsumQuote(1)) = QuoteEventCountDay1;
CombinedQuoteData(LengthCumsumQuote(1)+1:LengthCumsumQuote(2)) ...
    = QuoteEventCountDay2;
CombinedQuoteData(LengthCumsumQuote(2)+1:LengthCumsumQuote(3)) ...
    = QuoteEventCountDay3;
CombinedQuoteData(LengthCumsumQuote(3)+1:LengthCumsumQuote(4)) ...
    = QuoteEventCountDay4;
CombinedQuoteData(LengthCumsumQuote(4)+1:LengthCumsumQuote(5)) ...
    = QuoteEventCountDay5;

% remove the first observation in trade series
CombinedTradeData(1) = [];
% perform the first difference for log quote midpoint series
CombinedQuoteData    = CombinedQuoteData(2:end)-
CombinedQuoteData(1:end-1);

% summary statistics of the number of trades and quotes

% for trade
TradeEventCountDay1(TradeEventCountDay1==0) = []; % remove zero
entries
TradeEventCountDay2(TradeEventCountDay2==0) = [];
TradeEventCountDay3(TradeEventCountDay3==0) = [];
TradeEventCountDay4(TradeEventCountDay4==0) = [];
TradeEventCountDay5(TradeEventCountDay5==0) = [];

LT1 = numel(TradeEventCountDay1); % calculate the effective number of
trades
LT2 = numel(TradeEventCountDay2);
LT3 = numel(TradeEventCountDay3);
LT4 = numel(TradeEventCountDay4);
LT5 = numel(TradeEventCountDay5);

SumStatsTrade      = cell(6,2);
SumStatsTrade(:,1) = {'Day1';'Day2';'Day3';'Day4';'Day5';'Total'};
SumStatsTrade(:,2) = {LT1;LT2;LT3;LT4;LT5;sum([LT1;LT2;LT3;LT4;LT5])};

% for quote
QT1 = numel(QuoteEventCountDay1); % calculate the effective number of
trades
```

```
QT2 = numel(QuoteEventCountDay2);
QT3 = numel(QuoteEventCountDay3);
QT4 = numel(QuoteEventCountDay4);
QT5 = numel(QuoteEventCountDay5);

SumStatsQuote      = cell(6,2);
SumStatsQuote(:,1) = {'Day1';'Day2';'Day3';'Day4';'Day5';'Total'};
SumStatsQuote(:,2) = {QT1;QT2;QT3;QT4;QT5;sum([QT1;QT2;QT3;QT4;QT5])};


% save the combined series in a .mat file
if strcmpi(filename,'') == 0
    switch type
        case 'approx'

AllData=struct('Trade',CombinedTradeData,'QuoteRevision',CombinedQuote
Data,...
                'Average',{average},'Individual',{individual});
            save(filename,'-struct','AllData');
        case 'real'

AllData=struct('Trade',CombinedTradeData,'QuoteRevision',CombinedQuote
Data,...

'SumStatsTrade',{SumStatsTrade},'SumStatsQuote',{SumStatsQuote});
            save(filename,'-struct','AllData');
    end
end

end
```

C2 – Approximation of Trade Directions

```
%1. This function approximates the direction of a trade employing the
%   approach stated in Hasbrouck(1991a)(1991b)
%2. Note that the input needs to be in 'raw data' form
%3. pcor  = percentage of correct number of approximations
%   psign = percentage of trades assigned as either buys or sells
%   pb    = percentage of trades assigned as buys
%   pb    = percentage of trades assigned as sells

function
[y,pcor,psign,pb,ps]=TradeSignApprox(trade_raw_data,quote_raw_data)

LengthTradeRaw = length(trade_raw_data);
LengthQuoteRaw = length(quote_raw_data);

% Step 1. convert the time stamp format
TradeTimeConverted = TimeFormatConverter(trade_raw_data);
QuoteTimeConverted = LeadingZeroRemover(quote_raw_data);

% Step 2. change the time stamp in the raw data sets to make the time
stamp in
% trade and quote series consistent
for i = 2:LengthTradeRaw
    trade_raw_data{i,1} = TradeTimeConverted(i-1);
end

for j = 2:LengthQuoteRaw
    quote_raw_data{j,2} = QuoteTimeConverted(j-1);
end

% Step 3. removes the trade and quote observations occurred before
% 9.30am and after 4.00pm, at which market opens and closes
respectively

% for trade series
for start = 2:LengthTradeRaw
    if trade_raw_data{start,1} > 93000000
        break % do NOT include the trade observation AT 9.30am by
using '>'
    end
end

for terminal = 2:LengthTradeRaw
    if trade_raw_data{terminal,1} >= 160000000
        break
    end
end

if start == 2
    trade_raw_data(terminal:end,:) = [];
else
    trade_raw_data([2:start-1,terminal:end],:) = [];
end

% for quote series
for start = 2:LengthQuoteRaw
    if quote_raw_data{start,2} >= 93000000
        break % include the quote observation AT 9.30am by using '>='
    end
end

for terminal = 2:LengthQuoteRaw
    if quote_raw_data{terminal,2} >= 160000000
        break
    end
end

if start == 2
    quote_raw_data(terminal:end,:) = [];
else
    quote_raw_data([2:start-1,terminal:end],:) = [];
end

% Step 4. estimate the sign trade by trade by comparing the
transaction
% price in trade observation with the prevailing quote midpoint. If
the
% transaction price is above the prevailing quote midpoint the
associated
% trade is assumed to be a purchase and a positive sign is added to
the
% trade volume, while a negative sign is added if the transaction
price is
% below the prevailing quote midpoint. If the transaction price is
equal
% to the quote midpoint, the direction of trade is undetermined and
x_t is
% set to 0.
LengthTradeRaw = length(trade_raw_data);
LengthQuoteRaw = length(quote_raw_data);

a = 2; % trade series count
b = 2; % quote series count

while a <=LengthTradeRaw && b <= LengthQuoteRaw

    if a == 2 % scenario 1 - handle the initial trade observation
        while trade_raw_data{a,1} > quote_raw_data{b,2}
            b = b + 1;
        end

        if trade_raw_data{a,5}...
                < (0.5*(quote_raw_data{b-1,3}+quote_raw_data{b-1,4}))
            trade_raw_data{a,3} = -trade_raw_data{a,3};
        elseif trade_raw_data{a,5}...
                == (0.5*(quote_raw_data{b-1,3}+quote_raw_data{b-1,4}))
            trade_raw_data{a,3} = 0;
        end
        a = a + 1;

        % scenario 2 - the stopping process
        % the program enters the stopping process if either trade or
quote
        % series reach the end
    elseif a == LengthTradeRaw || b == LengthQuoteRaw
        if a == LengthTradeRaw% if trade series reach the end first
            while trade_raw_data{a,1} > quote_raw_data{b,2} &&...
                    b < LengthQuoteRaw
                b = b + 1;
            end

            if b ~= LengthQuoteRaw
                if trade_raw_data{a,5}...
                        < (0.5*(quote_raw_data{b-
1,3}+quote_raw_data{b-1,4}))
                    trade_raw_data{a,3} = -trade_raw_data{a,3};
                elseif trade_raw_data{a,5}...
                        == (0.5*(quote_raw_data{b-
1,3}+quote_raw_data{b-1,4}))
                    trade_raw_data{a,3} = 0;
                end
            else % when b == LengthQuoteRaw (reach the end of vector)
                if trade_raw_data{a,1} <= quote_raw_data{b,2}
                    if trade_raw_data{a,5}...
                            < (0.5*(quote_raw_data{b-
1,3}+quote_raw_data{b-1,4}))
                        trade_raw_data{a,3} = -trade_raw_data{a,3};
                    elseif trade_raw_data{a,5}...
                            == (0.5*(quote_raw_data{b-
1,3}+quote_raw_data{b-1,4}))
                        trade_raw_data{a,3} = 0;
                    end
                else
                    if trade_raw_data{a,5}...
                            <
(0.5*(quote_raw_data{b,3}+quote_raw_data{b,4}))
                        trade_raw_data{a,3} = -trade_raw_data{a,3};
                    elseif trade_raw_data{a,5}...
                            ==
(0.5*(quote_raw_data{b,3}+quote_raw_data{b,4}))
                        trade_raw_data{a,3} = 0;
                    end
                end
            end
            a = a + 1;
        elseif b == LengthQuoteRaw% if quote series reach the end
first
            while a <= LengthTradeRaw
                if trade_raw_data{a,5}...
                        <
(0.5*(quote_raw_data{b,3}+quote_raw_data{b,4}))
                    trade_raw_data{a,3} = -trade_raw_data{a,3};
                elseif trade_raw_data{a,5}...
                        ==
(0.5*(quote_raw_data{b,3}+quote_raw_data{b,4}))
                    trade_raw_data{a,3} = 0;
                end
                a = a + 1;
            end
        end

        % scenario 3 - cases where 2 adjacent trades happened at
DIFFERENT
        % time
    elseif trade_raw_data{a,1} ~= trade_raw_data{a-1,1}
        if trade_raw_data{a,1} <= quote_raw_data{b,2}
            if trade_raw_data{a,5}...
                    < (0.5*(quote_raw_data{b-1,3}+quote_raw_data{b-
1,4}))
```

```
                trade_raw_data{a,3} = -trade_raw_data{a,3};
            elseif trade_raw_data{a,5}...
                    == (0.5*(quote_raw_data{b-1,3}+quote_raw_data{b-
1,4}))
                trade_raw_data{a,3} = 0;
            end
            a = a + 1;
        end

        % KEY step: we lead 'b' one step forward ONLY IF the next
trade
        % occurs AFTER the then-selected quote
        if trade_raw_data{a,1} > quote_raw_data{b,2}
            b = b + 1;
        end

        % scenario 4 - cases where 2 adjacent trades happened at the
SAME
        % time
    elseif trade_raw_data{a,1} == trade_raw_data{a-1,1}
        if trade_raw_data{a,5}...
                < (0.5*(quote_raw_data{b-1,3}+quote_raw_data{b-1,4}))
            trade_raw_data{a,3} = -trade_raw_data{a,3};
        elseif trade_raw_data{a,5}...
                == (0.5*(quote_raw_data{b-1,3}+quote_raw_data{b-1,4}))
            trade_raw_data{a,3} = 0;
        end
        a = a + 1; % in this case we do NOT move b one step forward
    end
end

% Step 5. calculate pcor, psign, pb & ps

% for pcor
SignIndicator = 0;

for z = 2: LengthTradeRaw
    if trade_raw_data{z,3} > 0 && trade_raw_data{z,4} == 'B' ||...
            trade_raw_data{z,3} < 0 && trade_raw_data{z,4} == 'S'
        SignIndicator = SignIndicator + 1;
    end
end

pcor  = SignIndicator/LengthTradeRaw;

% for psign
SignIndicator = 0;

for z = 2: LengthTradeRaw
    if trade_raw_data{z,3} ~= 0
        SignIndicator = SignIndicator + 1;
    end
end

psign = SignIndicator/LengthTradeRaw;

% for pb
BuyIndicator = 0;
```

```
for z = 2: LengthTradeRaw
    if trade_raw_data{z,3} > 0
        BuyIndicator = BuyIndicator + 1;
    end
end

pb = BuyIndicator/LengthTradeRaw;

% for ps
SellIndicator = 0;

for z = 2: LengthTradeRaw
    if trade_raw_data{z,3} < 0
        SellIndicator = SellIndicator + 1;
    end
end

ps = SellIndicator/LengthTradeRaw;

% Step 6. generate the main output
y = zeros(LengthTradeRaw-1,2); % preallocation of output matrix

for count = 2:LengthTradeRaw
    y(count-1,1) = trade_raw_data{count,1};
    y(count-1,2) = trade_raw_data{count,3};
end

end
```

C3 – Structural Vector Autoregressive Model Estimation

```
%This function generates two transformed trade series using the
original
%combined trade data as input

function [SignedIndicator,SignedSquared] ...
    = TradeSeriesTransformer(combined_trade_data)

SignedIndicator = zeros(length(combined_trade_data),1);

% 1. create 'signed trade indicator variable':
%    the indicator = 1 if the trade volume is positive and -1 if
negative
for i = 1:length(combined_trade_data)
    if combined_trade_data(i) > 0
        SignedIndicator(i) = 1;
    elseif combined_trade_data(i) < 0
        SignedIndicator(i) = -1;
    else
        SignedIndicator(i) = 0;
    end
end

% 2. create 'signed squared trade variable
SignedSquared = SignedIndicator.*(combined_trade_data.^2);

end


%
[EstOutputCell,sumstatSVAR,meanCell,e_corrMatrix,beta_20by4,e_Matrix].
..
%     = VARtoSVAR(xt,rt,filename,writeOption,period,cap)
% convert x_t, to a 100-share ROUND LOTs unit before input it into the
function

function
[EstOutputCell,sumstatSVAR,meanCell,e_corrMatrix,beta_20by4,e_Matrix].
..
    = VARtoSVAR(xt,rt,filename,writeOption,period,cap)

xt = round(xt/100); % convert trade to a unit of 100-share round lot

% 1. calculate signed trade indicator X0 and signed squared trade
volume Xt^2
[x0,x2] = TradeSeriesTransformer(xt);

% 2. OLS regression equation-by-equation
T = numel(xt)-5; % Note: remember to subtract '5', otherwise there
will be
%                  dimension mismatch

Y_x0 = x0(6:end);
Y_xt = xt(6:end);
Y_x2 = x2(6:end);
Y_rt = rt(6:end);

% each series with 5 lags
```

```
Xi       =[x0(5:end-1),x0(4:end-2),x0(3:end-3),x0(2:end-4),x0(1:end-
5),...
    xt(5:end-1),xt(4:end-2),xt(3:end-3),xt(2:end-4),xt(1:end-5),...
    x2(5:end-1),x2(4:end-2),x2(3:end-3),x2(2:end-4),x2(1:end-5),...
    rt(5:end-1),rt(4:end-2),rt(3:end-3),rt(2:end-4),rt(1:end-5)];

beta_ols_x0 = lscov(Xi,Y_x0);
Y_hat_x0    = Xi*beta_ols_x0; % obtain the fitted line
e_x0        = Y_x0-Y_hat_x0; % obtain innovation

beta_ols_xt = lscov(Xi,Y_xt);
Y_hat_xt    = Xi*beta_ols_xt;
e_xt        = Y_xt-Y_hat_xt;

beta_ols_x2 = lscov(Xi,Y_x2);
Y_hat_x2    = Xi*beta_ols_x2;
e_x2        = Y_x2-Y_hat_x2;

beta_ols_rt = lscov(Xi,Y_rt);
Y_hat_rt    = Xi*beta_ols_rt;
e_rt        = Y_rt-Y_hat_rt;

e_Matrix= [e_x0,e_xt,e_x2,e_rt];
e_covMatrix = cov(e_Matrix);
e_corrMatrix= corrcoef(e_Matrix); %correlation matrix

% 3. Reshape coefficient into the VAR compatible n-by-n form
beta_20by4 = [beta_ols_x0,beta_ols_xt,beta_ols_x2,beta_ols_rt];
%
id   = 1;
td   = 6;
sq   = 11;
r    = 16;

phi1 =
[beta_20by4(id,1),beta_20by4(td,1),beta_20by4(sq,1),beta_20by4(r,1);..
.

beta_20by4(id,2),beta_20by4(td,2),beta_20by4(sq,2),beta_20by4(r,2);...

beta_20by4(id,3),beta_20by4(td,3),beta_20by4(sq,3),beta_20by4(r,3);...

beta_20by4(id,4),beta_20by4(td,4),beta_20by4(sq,4),beta_20by4(r,4)];
%
id   = id+1; % no 'loop' use for boosting speed
td   = td+1;
sq   = sq+1;
r    = r+1;

phi2 =
[beta_20by4(id,1),beta_20by4(td,1),beta_20by4(sq,1),beta_20by4(r,1);..
.

beta_20by4(id,2),beta_20by4(td,2),beta_20by4(sq,2),beta_20by4(r,2);...

beta_20by4(id,3),beta_20by4(td,3),beta_20by4(sq,3),beta_20by4(r,3);...

beta_20by4(id,4),beta_20by4(td,4),beta_20by4(sq,4),beta_20by4(r,4)];
%
id   = id+1;
```

```
td    = td+1;
sq    = sq+1;
r     = r+1;

phi3 =
[beta_20by4(id,1),beta_20by4(td,1),beta_20by4(sq,1),beta_20by4(r,1);..
.

beta_20by4(id,2),beta_20by4(td,2),beta_20by4(sq,2),beta_20by4(r,2);...

beta_20by4(id,3),beta_20by4(td,3),beta_20by4(sq,3),beta_20by4(r,3);...

beta_20by4(id,4),beta_20by4(td,4),beta_20by4(sq,4),beta_20by4(r,4)];
%
id    = id+1;
td    = td+1;
sq    = sq+1;
r     = r+1;

phi4 =
[beta_20by4(id,1),beta_20by4(td,1),beta_20by4(sq,1),beta_20by4(r,1);..
.

beta_20by4(id,2),beta_20by4(td,2),beta_20by4(sq,2),beta_20by4(r,2);...

beta_20by4(id,3),beta_20by4(td,3),beta_20by4(sq,3),beta_20by4(r,3);...

beta_20by4(id,4),beta_20by4(td,4),beta_20by4(sq,4),beta_20by4(r,4)];
%
id    = id+1;
td    = td+1;
sq    = sq+1;
r     = r+1;

phi5 =
[beta_20by4(id,1),beta_20by4(td,1),beta_20by4(sq,1),beta_20by4(r,1);..
.

beta_20by4(id,2),beta_20by4(td,2),beta_20by4(sq,2),beta_20by4(r,2);...

beta_20by4(id,3),beta_20by4(td,3),beta_20by4(sq,3),beta_20by4(r,3);...

beta_20by4(id,4),beta_20by4(td,4),beta_20by4(sq,4),beta_20by4(r,4)];

% 4. Solve a function in matlab to obtain the structural matrix A0 and
Omega(v)
D11 = e_covMatrix(1,1);
D22 = e_covMatrix(2,2);
D33 = e_covMatrix(3,3);
A041= -e_covMatrix(4,1)/D11;
A042= -e_covMatrix(4,2)/D22;
A043= -e_covMatrix(4,3)/D33;
D44 = e_covMatrix(4,4)-D11*(A041^2)-D22*(A042^2)-D33*(A043^2);
% L_e  = chol(e_covMatrix,'lower'); %alternative approach: cholesky
decomposition
% D11  = L_e(1,1)^2;
% D22  = L_e(2,1)^2+L_e(2,2)^2;
% D33  = L_e(3,1)^2+L_e(3,2)^2+L_e(3,3)^2;
% A041 = -L_e(4,1)/L_e(1,1);
% A042 = -(L_e(2,1)*L_e(4,1)+L_e(2,2)*L_e(4,2))/D22;
% A043 = -(L_e(3,1)*L_e(4,1)+L_e(3,2)*L_e(4,2)+L_e(3,3)*L_e(4,3))/D33;
% D44  = L_e(4,1)^2+L_e(4,2)^2+L_e(4,3)^2+L_e(4,4)^2-D11*A041^2-
D22*A042^2-D33*A043^2;
% %
A0          = [1,0,0,0;0,1,0,0;0,0,1,0;A041,A042,A043,1]; % define
structural matrix
invA0       = [1,0,0,0;0,1,0,0;0,0,1,0;-A041,-A042,-A043,1];
v_covMatrix = [D11,0,0,0;0,D22,0,0;0,0,D33,0;0,0,0,D44];

% 5. Use A0 to convert the VAR(5) coefficient estimates to SVAR(5)
ones
A1 = A0*phi1;
A2 = A0*phi2;
A3 = A0*phi3;
A4 = A0*phi4;
A5 = A0*phi5;

% 6. Reorganise SVAR(5) coefficient estimates to column-wise form for
each eq.
beta_x0              = zeros(20,1);
beta_x0([1,6,11,16]) = A1(1,:);
beta_x0([2,7,12,17]) = A2(1,:);
beta_x0([3,8,13,18]) = A3(1,:);
beta_x0([4,9,14,19]) = A4(1,:);
beta_x0([5,10,15,20])= A5(1,:);
% beta_x0              = RoundToDecimalPlace(beta_x0,7); %round to 3
decimal places

beta_xt              = zeros(20,1);
beta_xt([1,6,11,16]) = A1(2,:);
beta_xt([2,7,12,17]) = A2(2,:);
beta_xt([3,8,13,18]) = A3(2,:);
beta_xt([4,9,14,19]) = A4(2,:);
beta_xt([5,10,15,20])= A5(2,:);
% beta_xt              = RoundToDecimalPlace(beta_xt,7);

beta_x2              = zeros(20,1);
beta_x2([1,6,11,16]) = A1(3,:);
beta_x2([2,7,12,17]) = A2(3,:);
beta_x2([3,8,13,18]) = A3(3,:);
beta_x2([4,9,14,19]) = A4(3,:);
beta_x2([5,10,15,20])= A5(3,:);
% beta_x2              = RoundToDecimalPlace(beta_x2,7);

% beta rt has different size since it has contemporaneous x0,xt & x2
beta_rt              = zeros(23,1);
beta_rt([2,8,14,19]) = A1(4,:);
beta_rt([3,9,15,20]) = A2(4,:);
beta_rt([4,10,16,21])= A3(4,:);
beta_rt([5,11,17,22])= A4(4,:);
beta_rt([6,12,18,23])= A5(4,:);
beta_rt([1,7,13])    = [-A041,-A042,-A043]; % should change the sign
when move to RHS
% beta_rt              = RoundToDecimalPlace(beta_rt,10);

% 7. Use ?u to calculate std.error and t statistics and conduct t-stat
for
%    parameter estimates of SVAR(5)
v_var_vec = diag(v_covMatrix); % obtain the variance of innovations
v_var_x0  = v_var_vec(1);
```

```
v_var_xt  = v_var_vec(2);
v_var_x2  = v_var_vec(3);
v_var_rt  = v_var_vec(4);

% Xi_x is a T-by-20 matrix
Xi_x = [x0(5:end-1),x0(4:end-2),x0(3:end-3),x0(2:end-4),x0(1:end-5),...
    xt(5:end-1),xt(4:end-2),xt(3:end-3),xt(2:end-4),xt(1:end-5),...
    x2(5:end-1),x2(4:end-2),x2(3:end-3),x2(2:end-4),x2(1:end-5),...
    rt(5:end-1),rt(4:end-2),rt(3:end-3),rt(2:end-4),rt(1:end-5)];
XX_x = inv(Xi_x'*Xi_x);
XXdiag_x = diag(XX_x);

% Xi_r is a T-by-23 matrix
Xi_r = [x0(6:end),x0(5:end-1),x0(4:end-2),x0(3:end-3),x0(2:end-4),x0(1:end-5),...
    xt(6:end),xt(5:end-1),xt(4:end-2),xt(3:end-3),xt(2:end-4),xt(1:end-5),...
    x2(6:end),x2(5:end-1),x2(4:end-2),x2(3:end-3),x2(2:end-4),x2(1:end-5),...
    rt(5:end-1),rt(4:end-2),rt(3:end-3),rt(2:end-4),rt(1:end-5)];
XX_r = inv(Xi_r'*Xi_r);
XXdiag_r = diag(XX_r);

% t-test
cv = tinv([0.95;0.975;0.99],T); % obtain 3by1 critival value column vector
% for equation x0
std_error_x0= sqrt(v_var_x0*XXdiag_x);
t_test_x0   = abs(beta_x0./std_error_x0);
t_test_x0   = RoundToDecimalPlace(t_test_x0,2); %round to 2 decimal places
% tabulate the results
beta_x0_cell=cell(40,1);
j = 1;
for i = 1:20
    if t_test_x0(i) > cv(1) && t_test_x0(i) <= cv(2)
        beta_x0_cell(j)  = {strcat(num2str(beta_x0(i),'%.3g'),'*')};
        beta_x0_cell(j+1)  = {t_test_x0(i)};
    elseif t_test_x0(i) > cv(2) && t_test_x0(i) <= cv(3)
        beta_x0_cell(j)  = {strcat(num2str(beta_x0(i),'%.3g'),'**')};
        beta_x0_cell(j+1)  = {t_test_x0(i)};
    elseif t_test_x0(i) > cv(3)
        beta_x0_cell(j)  = {strcat(num2str(beta_x0(i),'%.3g'),'***')};
        beta_x0_cell(j+1)  = {t_test_x0(i)};
    else
        beta_x0_cell(j)  = {num2str(beta_x0(i),'%.3g')};
        beta_x0_cell(j+1)  = {t_test_x0(i)};
    end
    j = j+2;
end

% for equation xt
std_error_xt= sqrt(v_var_xt*XXdiag_x);
t_test_xt   = abs(beta_xt./std_error_xt);
t_test_xt   = RoundToDecimalPlace(t_test_xt,2);
% tabulate the results
beta_xt_cell=cell(40,1);
j = 1;
for i = 1:20
```

```
    if t_test_xt(i) > cv(1) && t_test_xt(i) <= cv(2)
        beta_xt_cell(j) = {strcat(num2str(beta_xt(i),'%.3g'),'*')};
        beta_xt_cell(j+1) = {t_test_xt(i)};
    elseif t_test_xt(i) > cv(2) && t_test_xt(i) <= cv(3)
        beta_xt_cell(j) = {strcat(num2str(beta_xt(i),'%.3g'),'**')};
        beta_xt_cell(j+1) = {t_test_xt(i)};
    elseif t_test_xt(i) > cv(3)
        beta_xt_cell(j) = {strcat(num2str(beta_xt(i),'%.3g'),'***')};
        beta_xt_cell(j+1) = {t_test_xt(i)};
    else
        beta_xt_cell(j) = {num2str(beta_xt(i),'%.3g')};
        beta_xt_cell(j+1) = {t_test_xt(i)};
    end
    j = j+2;
end

% for equation x2
std_error_x2= sqrt(v_var_x2*XXdiag_x);
t_test_x2   = abs(beta_x2./std_error_x2);
t_test_x2   = RoundToDecimalPlace(t_test_x2,2);
% tabulate the results
beta_x2_cell=cell(40,1);
j = 1;
for i = 1:20
   if t_test_x2(i) > cv(1) && t_test_x2(i) <= cv(2)
        beta_x2_cell(j) = {strcat(num2str(beta_x2(i),'%.3g'),'*')};
        beta_x2_cell(j+1) = {t_test_x2(i)};
    elseif t_test_x2(i) > cv(2) && t_test_x2(i) <= cv(3)
        beta_x2_cell(j) = {strcat(num2str(beta_x2(i),'%.3g'),'**')};
        beta_x2_cell(j+1) = {t_test_x2(i)};
    elseif t_test_x2(i) > cv(3)
        beta_x2_cell(j) = {strcat(num2str(beta_x2(i),'%.3g'),'***')};
        beta_x2_cell(j+1) = {t_test_x2(i)};
    else
        beta_x2_cell(j) = {num2str(beta_x2(i),'%.3g')};
        beta_x2_cell(j+1) = {t_test_x2(i)};
    end
    j = j+2;
end
% for equation rt
std_error_rt= sqrt(v_var_rt*XXdiag_r);
t_test_rt   = abs(beta_rt./std_error_rt);
t_test_rt   = RoundToDecimalPlace(t_test_rt,2);
beta_rt_cell=cell(46,1);
j = 1;
for i = 1:23
    if t_test_rt(i) > cv(1) && t_test_rt(i) <= cv(2)
        beta_rt_cell(j) = {strcat(num2str(beta_rt(i),'%.3g'),'*')};
        beta_rt_cell(j+1) = {t_test_rt(i)};
    elseif t_test_rt(i) > cv(2) && t_test_rt(i) <= cv(3)
        beta_rt_cell(j) = {strcat(num2str(beta_rt(i),'%.3g'),'**')};
        beta_rt_cell(j+1) = {t_test_rt(i)};
    elseif t_test_rt(i) > cv(3)
        beta_rt_cell(j) = {strcat(num2str(beta_rt(i),'%.3g'),'***')};
        beta_rt_cell(j+1) = {t_test_rt(i)};
    else
        beta_rt_cell(j) = {strcat(num2str(beta_rt(i),'%.3g'))};
        beta_rt_cell(j+1) = {t_test_rt(i)};
    end
    j = j+2;
end
```

```
% 8. Use matlab function vgxset to build a VAR(5) with KNOWN parameter
values
SpecAR = vgxset('AR',{phi1,phi2,phi3,phi4,phi5}); % set AR
specification

% 9. Use vgxma() to convert the VAR(5) to VMA(30)
SpecMA = vgxma(SpecAR,30);
MA     = SpecMA.MA;

% 10. Compute the response of the model to structural shock by POST
%     multiplying inv(A0) to each n-by-n VMA coefficient matrix
MA_structural    = cell(31,1);
MA_structural(1) = {invA0};
for i = 2:31
    MA_structural(i) = {MA{i-1}*invA0};
end

% 11. Compute the sigma(x)/hour, sigma(w)/hour, sigma(w,x)/hour, R2_w
and impulse
%     and convert them into /hour form by applying
%     ___× no.of obs./(no.of.trading.days×6.5hrs/day)
%     based on the formulae on Hasbrouck(1991b, p.577)

% sigma(x)/hour: standard deviation of innovation in xt equation
sigma_x    = sqrt(v_var_xt);
switch cap
    case 'Srt'
        sigma_x_hr = sigma_x*T/(5*0.5);
    case 'Mid'
        sigma_x_hr = sigma_x*T/(5*5.5);
    case 'End'
        sigma_x_hr = sigma_x*T/(5*0.5);
    otherwise
        sigma_x_hr = sigma_x*T/(5*6.5); % change to a per trading hour
basis
end

% sigma(w)/hour(×100): standard deviation of change in the efficient
price
Omega = v_covMatrix(1:3,1:3);

bstar_sum = zeros(1,3);
for i = 1:31
    bstar_sum = bstar_sum+MA_structural{i}(4,[1,2,3]);
end

astar_sum=0;
for i = 2:31
    astar_sum=astar_sum+MA_structural{i}(4,4);
end

sigmasq_w  = bstar_sum*Omega*bstar_sum'+v_var_rt*(1+astar_sum)^2;
sigma_w    = sqrt(sigmasq_w);
switch cap
    case 'Srt'
        sigma_w_hr = (sigma_w*T/(5*0.5))*100;
    case 'Mid'
        sigma_w_hr = (sigma_w*T/(5*5.5))*100;
    case 'End'
        sigma_w_hr = (sigma_w*T/(5*0.5))*100;
    otherwise
        sigma_w_hr = (sigma_w*T/(5*6.5))*100;
end

% sigma(w,x)/hour(×100): square root of the variance of trade-
correlated part of the
% efficient price changes
sigmasq_wx = bstar_sum*Omega*bstar_sum';
sigma_wx    = sqrt(sigmasq_wx);
switch cap
    case 'Srt'
        sigma_wx_hr= (sigma_wx*T/(5*0.5))*100;
    case 'Mid'
        sigma_wx_hr= (sigma_wx*T/(5*5.5))*100;
    case 'End'
        sigma_wx_hr= (sigma_wx*T/(5*0.5))*100;
    otherwise
        sigma_wx_hr= (sigma_wx*T/(5*6.5))*100;
end

% R2_w:  the ratio of variance in trade-explained component of
efficient
% price changes to the total variance of efficient price changes
R2_w = sigmasq_wx/sigmasq_w;

% impulse(×100): the persistent impact of a 1000-share buy order on
the log quote
% midpoint i.e. the impulse response function
% the units of xt are 100share round lots, so 1000 share=10 100-share
round lots
% i.e. x0 = 1, xt = 10, x2 = 100
impulse = (bstar_sum*[1;10;100])*100; %expressed in 100%

% 12. Tabulate all the results; also calculate and tabulate the mean,
%     correlation and autocorrelation of the innovation processes

% 12.1 mean vector in cell form
meanCell          = cell(5,2);
symbol            = strsplit(filename,'_');
meanCell(1,2)     = symbol(1);
meanCell(:,1)     = {'Mean','x0','xt','x2','rt'};
meanCell(2:5,2)  = {mean(x0),mean(xt),mean(x2),mean(rt)};

% 12.2 correlation matrix
v_corrMatrix= corrcov(v_covMatrix); % Convert covariance matrix to
correlation matrix

% 12.3 tabulate SVAR(5) estimation outputs
% adjusted R squared -> goodness-of-fit of the model
RSS_x0     = T*v_var_x0;
mean_x0    = mean(x0);
TSS_x0     = (x0-mean_x0)'*(x0-mean_x0);
AdjRsq_x0  = 1-((T-1)/(T-20))*(RSS_x0/TSS_x0);
AdjRsq_x0  = RoundToDecimalPlace(AdjRsq_x0,3);

RSS_xt     = T*v_var_xt;
mean_xt    = mean(xt);
TSS_xt     = (xt-mean_xt)'*(xt-mean_xt);
```

```
AdjRsq_xt = 1-((T-1)/(T-20))*(RSS_xt/TSS_xt);
AdjRsq_xt = RoundToDecimalPlace(AdjRsq_xt,3);

RSS_x2    = T*v_var_x2;
mean_x2   = mean(x2);
TSS_x2    = (x2-mean_x2)'*(x2-mean_x2);
AdjRsq_x2 = 1-((T-1)/(T-20))*(RSS_x2/TSS_x2);
AdjRsq_x2 = RoundToDecimalPlace(AdjRsq_x2,3);

RSS_rt    = T*v_var_rt;
mean_rt   = mean(rt);
TSS_rt    = (rt-mean_rt)'*(rt-mean_rt);
AdjRsq_rt = 1-((T-1)/(T-23))*(RSS_rt/TSS_rt);
AdjRsq_rt = RoundToDecimalPlace(AdjRsq_rt,3);

v_var_x0  = RoundToDecimalPlace(v_var_x0,3);
v_var_xt  = RoundToDecimalPlace(v_var_xt,3);
v_var_x2  = RoundToDecimalPlace(v_var_x2,3);
v_var_rt  = RoundToDecimalPlace(v_var_rt,10);

% estimation outputs table in cell form
EstOutputCell              = cell(49,4);
EstOutputCell(1,:) = {'x0','xt','x2','rt'};
EstOutputCell([4:13,16:25,28:37,38:47],1) = beta_x0_cell;
EstOutputCell([4:13,16:25,28:37,38:47],2) = beta_xt_cell;
EstOutputCell([4:13,16:25,28:37,38:47],3) = beta_x2_cell;
EstOutputCell([2:13,14:25,26:37,38:47],4) = beta_rt_cell;
EstOutputCell(48,:)={AdjRsq_x0,AdjRsq_xt,AdjRsq_x2,AdjRsq_rt};
EstOutputCell(49,:)={v_var_x0,v_var_xt,v_var_x2,num2str(v_var_rt,'%.3g
')};

% 12.4 tabulate summary statistics
sumstatSVAR         = cell(10,2);
sumstatSVAR(2:10,1) = {'No.of
obs';'Impulse';'sigma(x)/event';'sigma(x)/hour';...

'sigma(w)/event';'sigma(w)/hour';'sigma(w,x)/event';'sigma(w,x)/hour';
'R2_w'};
sumstatSVAR(1,2)    = symbol(1);
sumstatSVAR(2,2)    = {T+5}; % number of observation
sumstatSVAR(3,2)    = {impulse}; % impulse(×100)
sumstatSVAR(4,2)    = {sigma_x};  % sigma(x)/event
sumstatSVAR(5,2)    = {sigma_x_hr}; % sigma(x)/hour
sumstatSVAR(6,2)    = {sigma_w*100}; % sigma(w)/event(×100)
sumstatSVAR(7,2)    = {sigma_w_hr}; % sigma(w)/hour(×100)
sumstatSVAR(8,2)    = {sigma_wx*100}; % sigma(w,x)/event(×100)
sumstatSVAR(9,2)    = {sigma_wx_hr}; % sigma(w,x)/hour(×100)
sumstatSVAR(10,2)    = {R2_w}; % R2_w

% 13. data storage
AllData=struct('EstOutputCell',{EstOutputCell},'sumstatSVAR',{sumstatS
VAR},...
    'meanCell',{meanCell},'v_corrMatrix',{v_corrMatrix},...

'e_corrMatrix',{e_corrMatrix},'beta_20by4',{beta_20by4},'e_Matrix',{e_
Matrix});
save(filename,'-struct','AllData');

% 14. write in Excel
if strcmpi(writeOption,'write')
```

```
xlswrite(strcat('N:\MATLAB\Results\VAR
Estimation_',period,'.xlsx'),...
    EstOutputCell(2:end,1),strcat(symbol{1},'_VAREst'),'C4:C51');

xlswrite(strcat('N:\MATLAB\Results\VAR
Estimation_',period,'.xlsx'),...
    EstOutputCell(2:end,2),strcat(symbol{1},'_VAREst'),'E4:E51');

xlswrite(strcat('N:\MATLAB\Results\VAR
Estimation_',period,'.xlsx'),...
    EstOutputCell(2:end,3),strcat(symbol{1},'_VAREst'),'G4:G51');

xlswrite(strcat('N:\MATLAB\Results\VAR
Estimation_',period,'.xlsx'),...
    EstOutputCell(2:end,4),strcat(symbol{1},'_VAREst'),'I4:I51');
end

end
```

C4 – Intraday Analyser

```
% This function divides data of a trading day into 3 time intervals:
%       one: 9:30:00am - 10:00:00am;
%       two: 10:00:01am - 3:30:00pm;
%       three:3:30:01pm - 4:00:00pm

function
[T_CountSrt,Q_CountSrt,T_CountMid,Q_CountMid,T_CountEnd,Q_CountEnd]
...
    = EventCounterIntraday(trade_raw_data, quote_raw_data)

% cleanse the data applying the user defined functions
TradeCombined     = TradeCumulater(trade_raw_data);
LogQuoteMidpoint = QuoteMidpointCalculator(quote_raw_data);
LengthMidpoint    = length(LogQuoteMidpoint);
LengthTradeCom    = length(TradeCombined);

TradeEventCount   = zeros(LengthMidpoint,1);
counter           = 1;
i                 = 1;

while i <= LengthMidpoint
    if LogQuoteMidpoint(i,1) == 93000000
        TradeEventCount(i) = 0;

        %if the program runs to the end of the trade series but there
are
        %still more entries in the quote series to be gone through,
then
        %set the remaining empty entries in TradeEventCount to be 0
    elseif counter > LengthTradeCom;
        TradeEventCount(i:end) = 0;

    elseif TradeCombined(counter,1) <= LogQuoteMidpoint(i,1)
        TradeEventCount(i) = TradeCombined(counter,2);
        counter = counter + 1;  % the counter moves 1 step forward
only if
        % we copy one trade volume data from TradeCombined to
TradeEventCount
    else
        TradeEventCount(i) = 0;
    end
    i = i + 1;
end

for start = 1:LengthMidpoint
    if  LogQuoteMidpoint(start,1) > 100000000
        break
    end
end
T_CountSrt = TradeEventCount(1:start-1);
Q_CountSrt = LogQuoteMidpoint(1:start-1,2);

for mid = 1:LengthMidpoint
    if  LogQuoteMidpoint(mid,1) > 153000000
        break
    end
end
T_CountMid = TradeEventCount(start:mid-1);
```

```
Q_CountMid = LogQuoteMidpoint(start:mid-1,2);

T_CountEnd = TradeEventCount(mid:end);
Q_CountEnd = LogQuoteMidpoint(mid:end,2);

end


% This function combines 5 trading days data for all 3 intraday
intervals

function [Combined_TSrt,Combined_QSrt,Combined_TMid,...
    Combined_QMid,Combined_TEnd,Combined_QEnd] ...
    = DataCombinerIntraday(trade_raw_data_day1,
quote_raw_data_day1,...
    trade_raw_data_day2, quote_raw_data_day2,...
    trade_raw_data_day3, quote_raw_data_day3,...
    trade_raw_data_day4, quote_raw_data_day4,...
    trade_raw_data_day5, quote_raw_data_day5,filename)

% 1. cleanse the data for each individual trading day
[T_CountSrt1,Q_CountSrt1,T_CountMid1,Q_CountMid1,T_CountEnd1,Q_CountEn
d1] ...
    = EventCounterIntraday(trade_raw_data_day1, quote_raw_data_day1);
[T_CountSrt2,Q_CountSrt2,T_CountMid2,Q_CountMid2,T_CountEnd2,Q_CountEn
d2] ...
    = EventCounterIntraday(trade_raw_data_day2, quote_raw_data_day2);
[T_CountSrt3,Q_CountSrt3,T_CountMid3,Q_CountMid3,T_CountEnd3,Q_CountEn
d3] ...
    = EventCounterIntraday(trade_raw_data_day3, quote_raw_data_day3);
[T_CountSrt4,Q_CountSrt4,T_CountMid4,Q_CountMid4,T_CountEnd4,Q_CountEn
d4] ...
    = EventCounterIntraday(trade_raw_data_day4, quote_raw_data_day4);
[T_CountSrt5,Q_CountSrt5,T_CountMid5,Q_CountMid5,T_CountEnd5,Q_CountEn
d5] ...
    = EventCounterIntraday(trade_raw_data_day5, quote_raw_data_day5);

% 2. calculate the total length of the combined series
LengthCumsum_TSrt = cumsum([length(T_CountSrt1), ...
    length(T_CountSrt2), length(T_CountSrt3), ...
    length(T_CountSrt4), length(T_CountSrt5)]);
LengthCumsum_QSrt = cumsum([length(Q_CountSrt1), ...
    length(Q_CountSrt2), length(Q_CountSrt3), ...
    length(Q_CountSrt4), length(Q_CountSrt5)]);

LengthCumsum_TMid = cumsum([length(T_CountMid1), ...
    length(T_CountMid2), length(T_CountMid3), ...
    length(T_CountMid4), length(T_CountMid5)]);
LengthCumsum_QMid = cumsum([length(Q_CountMid1), ...
    length(Q_CountMid2), length(Q_CountMid3), ...
    length(Q_CountMid4), length(Q_CountMid5)]);

LengthCumsum_TEnd = cumsum([length(T_CountEnd1), ...
    length(T_CountEnd2), length(T_CountEnd3), ...
    length(T_CountEnd4), length(T_CountEnd5)]);
LengthCumsum_QEnd = cumsum([length(Q_CountEnd1), ...
    length(Q_CountEnd2), length(Q_CountEnd3), ...
    length(Q_CountEnd4), length(Q_CountEnd5)]);

% 3. preallocation
```

```
Combined_TSrt = zeros(LengthCumsum_TSrt(end),1);
Combined_QSrt = zeros(LengthCumsum_QSrt(end),1);

Combined_TMid = zeros(LengthCumsum_TMid(end),1);
Combined_QMid = zeros(LengthCumsum_QMid(end),1);

Combined_TEnd = zeros(LengthCumsum_TEnd(end),1);
Combined_QEnd = zeros(LengthCumsum_QEnd(end),1);


% 4. Data - start of day
% trade series
Combined_TSrt(1:LengthCumsum_TSrt(1)) = T_CountSrt1;
Combined_TSrt(LengthCumsum_TSrt(1)+1:LengthCumsum_TSrt(2)) ...
     = T_CountSrt2;
Combined_TSrt(LengthCumsum_TSrt(2)+1:LengthCumsum_TSrt(3)) ...
     = T_CountSrt3;
Combined_TSrt(LengthCumsum_TSrt(3)+1:LengthCumsum_TSrt(4)) ...
     = T_CountSrt4;
Combined_TSrt(LengthCumsum_TSrt(4)+1:LengthCumsum_TSrt(5)) ...
     = T_CountSrt5;
% quote series
Combined_QSrt(1:LengthCumsum_QSrt(1)) = Q_CountSrt1;
Combined_QSrt(LengthCumsum_QSrt(1)+1:LengthCumsum_QSrt(2)) ...
     = Q_CountSrt2;
Combined_QSrt(LengthCumsum_QSrt(2)+1:LengthCumsum_QSrt(3)) ...
     = Q_CountSrt3;
Combined_QSrt(LengthCumsum_QSrt(3)+1:LengthCumsum_QSrt(4)) ...
     = Q_CountSrt4;
Combined_QSrt(LengthCumsum_QSrt(4)+1:LengthCumsum_QSrt(5)) ...
     = Q_CountSrt5;
% remove the first observation in trade series
Combined_TSrt(1) = [];
% perform the first difference for log quote midpoint series
Combined_QSrt    = Combined_QSrt(2:end)-Combined_QSrt(1:end-1);

% 5. Data - middle of day
% trade series
Combined_TMid(1:LengthCumsum_TMid(1)) = T_CountMid1;
Combined_TMid(LengthCumsum_TMid(1)+1:LengthCumsum_TMid(2)) ...
     = T_CountMid2;
Combined_TMid(LengthCumsum_TMid(2)+1:LengthCumsum_TMid(3)) ...
     = T_CountMid3;
Combined_TMid(LengthCumsum_TMid(3)+1:LengthCumsum_TMid(4)) ...
     = T_CountMid4;
Combined_TMid(LengthCumsum_TMid(4)+1:LengthCumsum_TMid(5)) ...
     = T_CountMid5;
% quote series
Combined_QMid(1:LengthCumsum_QMid(1)) = Q_CountMid1;
Combined_QMid(LengthCumsum_QMid(1)+1:LengthCumsum_QMid(2)) ...
     = Q_CountMid2;
Combined_QMid(LengthCumsum_QMid(2)+1:LengthCumsum_QMid(3)) ...
     = Q_CountMid3;
Combined_QMid(LengthCumsum_QMid(3)+1:LengthCumsum_QMid(4)) ...
     = Q_CountMid4;
Combined_QMid(LengthCumsum_QMid(4)+1:LengthCumsum_QMid(5)) ...
     = Q_CountMid5;
% remove the first observation in trade series
Combined_TMid(1) = [];
% perform the first difference for log quote midpoint series
```

```
Combined_QMid     = Combined_QMid(2:end)-Combined_QMid(1:end-1);

% 6. Data - end of day
% trade series
Combined_TEnd(1:LengthCumsum_TEnd(1)) = T_CountEnd1;
Combined_TEnd(LengthCumsum_TEnd(1)+1:LengthCumsum_TEnd(2)) ...
    = T_CountEnd2;
Combined_TEnd(LengthCumsum_TEnd(2)+1:LengthCumsum_TEnd(3)) ...
    = T_CountEnd3;
Combined_TEnd(LengthCumsum_TEnd(3)+1:LengthCumsum_TEnd(4)) ...
    = T_CountEnd4;
Combined_TEnd(LengthCumsum_TEnd(4)+1:LengthCumsum_TEnd(5)) ...
    = T_CountEnd5;
% quote series
Combined_QEnd(1:LengthCumsum_QEnd(1)) = Q_CountEnd1;
Combined_QEnd(LengthCumsum_QEnd(1)+1:LengthCumsum_QEnd(2)) ...
    = Q_CountEnd2;
Combined_QEnd(LengthCumsum_QEnd(2)+1:LengthCumsum_QEnd(3)) ...
    = Q_CountEnd3;
Combined_QEnd(LengthCumsum_QEnd(3)+1:LengthCumsum_QEnd(4)) ...
    = Q_CountEnd4;
Combined_QEnd(LengthCumsum_QEnd(4)+1:LengthCumsum_QEnd(5)) ...
    = Q_CountEnd5;
% remove the first observation in trade series
Combined_TEnd(1) = [];
% perform the first difference for log quote Endpoint series
Combined_QEnd     = Combined_QEnd(2:end)-Combined_QEnd(1:end-1);

% 7. Save data
AllData=struct('TSrt',Combined_TSrt,'QSrt',Combined_QSrt,...
    'TMid',Combined_TMid,'QMid',Combined_QMid,...
    'TEnd',Combined_TEnd,'QEnd',Combined_QEnd);
save(strcat(filename,'_Intraday'),'-struct','AllData');

end
```

C5 – Time Weighted Average Spread

```
% This function calculates time-weighted average (TWA) of all intraday
spread
% for all sample stocks

function SpreadCell = TWAspread(...
    quote_raw_data_day1,quote_raw_data_day2,quote_raw_data_day3,...
    quote_raw_data_day4,quote_raw_data_day5, filename)

% for quote series
QuoteData1 = QuoteDuplicatesRemover(quote_raw_data_day1);
QuoteData2 = QuoteDuplicatesRemover(quote_raw_data_day2);
QuoteData3 = QuoteDuplicatesRemover(quote_raw_data_day3);
QuoteData4 = QuoteDuplicatesRemover(quote_raw_data_day4);
QuoteData5 = QuoteDuplicatesRemover(quote_raw_data_day5);

LengthQ1 = length(QuoteData1);
LengthQ2 = length(QuoteData2);
LengthQ3 = length(QuoteData3);
LengthQ4 = length(QuoteData4);
LengthQ5 = length(QuoteData5);

ElapsedTimeQ1 = zeros(LengthQ1,1);
ElapsedTimeQ2 = zeros(LengthQ2,1);
ElapsedTimeQ3 = zeros(LengthQ3,1);
ElapsedTimeQ4 = zeros(LengthQ4,1);
ElapsedTimeQ5 = zeros(LengthQ5,1);

ElapsedTimeQ1(end) = 0;
ElapsedTimeQ2(end) = 0;
ElapsedTimeQ3(end) = 0;
ElapsedTimeQ4(end) = 0;
ElapsedTimeQ5(end) = 0;

% compute the time difference between 2 adjacent quotes

for i = 2:LengthQ1
    ElapsedTimeQ1(i-1) = QuoteData1(i,1) - QuoteData1(i-1,1);
end

for i = 2:LengthQ2
    ElapsedTimeQ2(i-1) = QuoteData2(i,1) - QuoteData2(i-1,1);
end

for i = 2:LengthQ3
    ElapsedTimeQ3(i-1) = QuoteData3(i,1) - QuoteData3(i-1,1);
end

for i = 2:LengthQ4
    ElapsedTimeQ4(i-1) = QuoteData4(i,1) - QuoteData4(i-1,1);
end

for i = 2:LengthQ5
    ElapsedTimeQ5(i-1) = QuoteData5(i,1) - QuoteData5(i-1,1);
end

LengthCumsumTime =
cumsum([LengthQ1,LengthQ2,LengthQ3,LengthQ4,LengthQ5]);
```

```
CombinedQuoteTime = zeros(LengthCumsumTime(end),1);

CombinedQuoteTime(1:LengthCumsumTime(1)) = ElapsedTimeQ1;
CombinedQuoteTime(LengthCumsumTime(1)+1:LengthCumsumTime(2)) ...
    = ElapsedTimeQ2;
CombinedQuoteTime(LengthCumsumTime(2)+1:LengthCumsumTime(3)) ...
    = ElapsedTimeQ3;
CombinedQuoteTime(LengthCumsumTime(3)+1:LengthCumsumTime(4)) ...
    = ElapsedTimeQ4;
CombinedQuoteTime(LengthCumsumTime(4)+1:LengthCumsumTime(5)) ...
    = ElapsedTimeQ5;

TotalTime = sum(CombinedQuoteTime);
Weight    = CombinedQuoteTime/TotalTime;

BidAsk1 = QuoteData1(:,2:3);
BidAsk2 = QuoteData2(:,2:3);
BidAsk3 = QuoteData3(:,2:3);
BidAsk4 = QuoteData4(:,2:3);
BidAsk5 = QuoteData5(:,2:3);

LengthCumsumBidAsk =
cumsum([LengthQ1,LengthQ2,LengthQ3,LengthQ4,LengthQ5]);
CombinedBidAsk = zeros(LengthCumsumBidAsk(end),2);

CombinedBidAsk(1:LengthCumsumBidAsk(1),:) = BidAsk1;
CombinedBidAsk(LengthCumsumBidAsk(1)+1:LengthCumsumBidAsk(2),:) ...
    = BidAsk2;
CombinedBidAsk(LengthCumsumBidAsk(2)+1:LengthCumsumBidAsk(3),:) ...
    = BidAsk3;
CombinedBidAsk(LengthCumsumBidAsk(3)+1:LengthCumsumBidAsk(4),:) ...
    = BidAsk4;
CombinedBidAsk(LengthCumsumBidAsk(4)+1:LengthCumsumBidAsk(5),:) ...
    = BidAsk5;

QuoteMidpoint = 0.5*(CombinedBidAsk(:,1)+CombinedBidAsk(:,2));
Spread        = abs(CombinedBidAsk(:,2)-CombinedBidAsk(:,1));
% compute TWA proportional spread (expressed in 100%)
ProportionalSpread = sum(Weight.*(Spread./QuoteMidpoint))*100;

SpreadCell      = cell(2,2);
SpreadCell(2,1) = {'Spread'};
ticker          = strsplit(filename,'_');
SpreadCell(1,2) = ticker(1);
SpreadCell(2,2) = {ProportionalSpread};

% data storage
AllData=struct('Spread',{SpreadCell});
save(filename,'-struct','AllData');

end
```