Prediction of Hidden Liquidity in the Limit Order Book of GLOBEX Futures

Hugh L. Christensen and Robert Woodmansey

Hugh L. Christensen
is a researcher in the Engineering Department at the University of Cambridge in Cambridge, UK.
hlc54@cam.ac.uk

Robert Woodmansey
is the chief executive of Onix Solutions in London, UK.
robert.woodmansey@onixs.biz

Modern securities exchanges have the concept of the open *limit order book* (LOB), where any market participant can see all the orders in the market. *Hidden orders* are a variation on this theme, where certain orders are not visible in the LOB. The LOB can be considered a store of participants' future intentions. The ability to hide information in this store is detrimental to traders, who use that information to decide upon their future actions. In contrast, the owner of the hidden information benefits by removing the informational disadvantage associated with having their intentions publicly known and avoiding being exploited by subsequent participants with more timely information.

Hidden volumes in the LOB are of great interest to both *liquidity providers* (market makers) and *liquidity takers*. Liquidity providers can use knowledge of hidden volumes both to generate alpha and manage risk. In terms of risk management, hidden orders can result in a liquidity provider's limit order being "picked off" when the liquidity provider did not intend to execute that volume or was unaware of some new information that had come to the market. If the liquidity provider knew of the hidden order, he or she might choose to withdraw his or her volume from the LOB. A liquidity provider's alpha generation can benefit from knowledge of a hidden order in the LOB, either directly or indirectly. First, there are a number of strategies for directly exploiting the hidden volume: for example, "front running," a type of hidden order called an *iceberg order*. An iceberg order is a limit order where only a fraction of the total order size is shown in the LOB at any one time (the *peak*), with the remainder of volume hidden. If the iceberg is an order to sell, the strategy is to short the market to the estimated size of the iceberg, causing the price to decrease. When the volume of the iceberg has been hit and the price has decreased, the position is closed at a profit (Durbin [2010]). Second, and indirectly, some alpha generation strategies use the information content of the LOB to forecast future returns, and hidden orders lead to a false picture of the LOB, giving an erroneous prediction of future returns (Cont et al. [2010]). Liquidity takers are interested in using hidden orders to reduce their exposure risk by minimizing the announcement of their intention to trade, thus decreasing their market impact. In this way, informed traders can conceal the fact that they know useful information from the rest of the market (De Winne and D'hondt [2007]).

The motivation behind this article is to present an online algorithm that can be used to detect iceberg orders for the major derivatives exchanges. The emphasis of the article is on the applicability of the research for use both in academia and in industry.

This article is structured as follows: First, we introduce the market participants to clarify why hidden volume is of relevance. In the next section, an overview of the exchanges is given and the data used in the paper presented. We then review the LOB, the different sorts of hidden volume are introduced, and we carry out a literature review on detecting icebergs. For the article's main contribution, we present an algorithm for detecting iceberg-hidden volume on GLOBEX and outline a low-latency computational implementation of the algorithm. The performance of the algorithm on LOB data is then simulated and results presented. Next, examples of how the algorithm can be used by the investment community are given. Finally, we draw conclusions and make suggestions for further work.

MARKET PARTICIPANTS

The application of quantitative approaches to trading is now a well-established field; however, the majority of participants do not have the ability to apply quantitative methods at a micro-structure level. This is still limited to a small subgroup of ultra-high-frequency traders. In the futures market, this subgroup is made up of the Principal Traders Group.[1] Unlike many quantitative hedge funds, this group is largely self-financed, and as such has a different outlook on risks and regulations. The main aim of this group is to participate in "low-risk" trading, which is limited to market-making and arbitrage activities. Given that these activities take place at very high frequency and result in short holding periods, the participants are required to trade very large volumes to generate their required returns. One such company, RSJ,[2] makes public its trading volumes for EUREX, CME GLOBEX, and NYSE Liffe derivatives exchanges, stating that their total monthly trading volume "exceeds 20 million lots." The data equate to 1.1%, 3.8%, and 4.7%, respectively, as a percentage of the total electronic volume traded on these exchanges in 2010 and 2011. These percentages are similar to figures released by the Scandinavian equities exchange, NASDAQ OMX, in 2011, which show that 10 high-frequency firms alone are responsible for 16% of the exchange's volume (Cave [2011]). Given the dependence of these firms on executing large volumes, detecting hidden volume is of particular economic relevance, as this is volume which could potentially be traded against, leading to increased profits.

DERIVATIVES EXCHANGES AND DATA

Algorithmic traders tend to be interested only in the most liquid, vanilla securities. To this end, we begin by considering the four leading Western derivatives exchanges by volume traded: CME GLOBEX, ICE, EUREX, and NYSE Liffe. These exchanges trade two main products: futures and options. While the volumes are similar in both products, there are higher quoting rates and fewer trades in options relative to the futures (Bank for International Settlement [2010]). The mechanisms by which electronic trading platforms operate tend to vary widely between exchanges, and we summarize the relevant points from these four exchanges in Exhibit 1. Due to their dominant market position, we proceed considering just CME GLOBEX, while noting that the contents of this article are relevant to the other exchanges with hidden volume. Descriptive statistics associated with GLOBEX and high-frequency trading are shown in Exhibit 2.

In the case of GLOBEX, options do not support implied pricing or, in some cases, iceberg orders,[3] and for this reason this article looks only at futures. Specifically, we consider the e-mini S&P 500 (ES) future for the 258 trading days of 2011. We use the CME datamine product, which is a recording of the real-time GLink session, is fully public, and can be accessed via the exchange (Chicago Mercantile Exchange [2012]). Unlike on many other exchanges, platforms, and dark pools, data

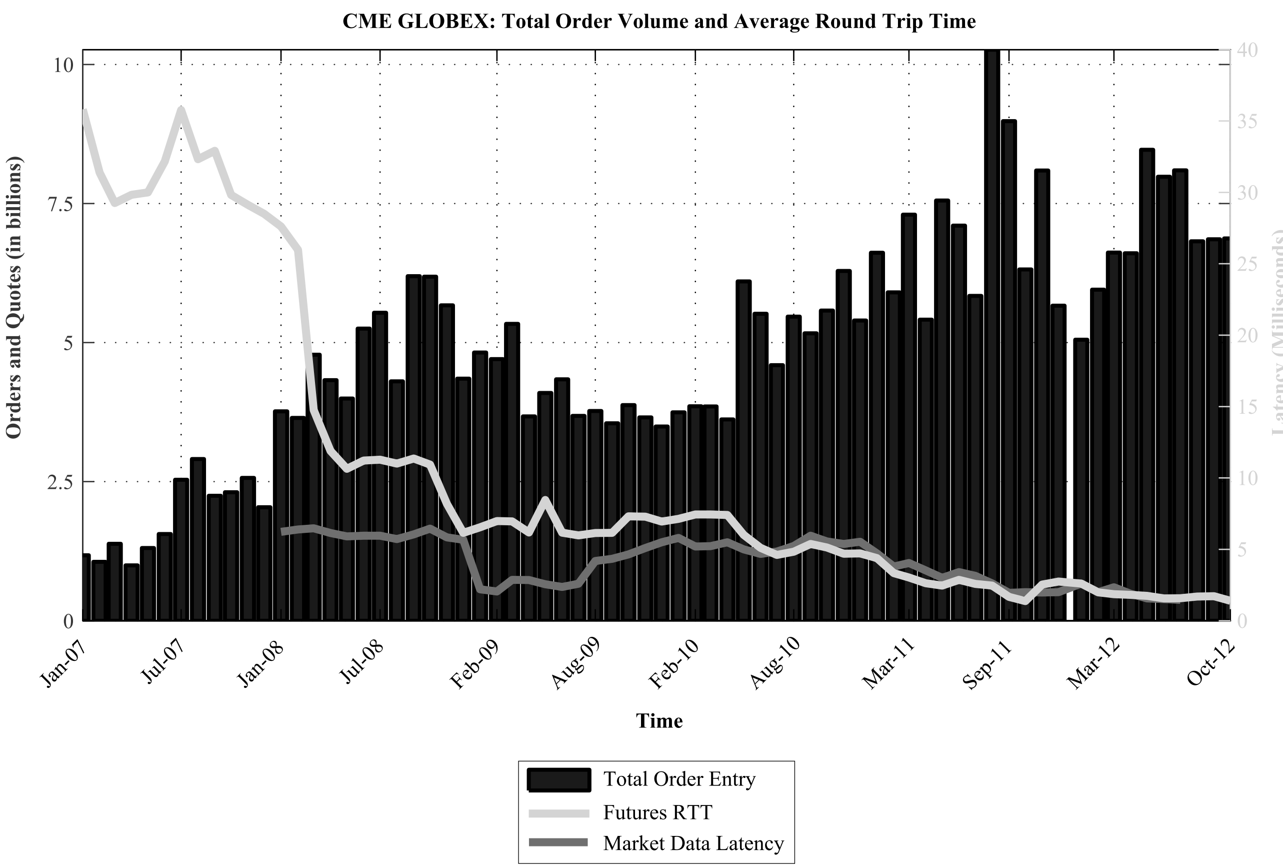
EXHIBIT 1
Derivatives Exchanges

	EUREX	CME	ICE	NYSE Liffe
Iceberg orders	✗	✓	✓	✗
Implied orders	✓	✓	✓	✓
Second-generation implieds	✓	✓	✓	✗
All implieds broadcast	✗	✗	✗	✓

E X H I B I T 2

CME GLOBEX Total Order Volume and Average Round Trip Time (RTT)

RTT is measured from the time the ilink gateway begins to process a message, processing by the match engine and subsequent outbound acknowledgement by the ilink gateway. Futures RTT and market data latency can be seen to be decreasing while total order volume is increasing.



access at CME is transparent and no parties have access to any special data that is restricted in any way, nor can "extra information" be purchased or subscribed to. Historical data were pre-processed from their raw FIX FAST structure into a structure suitable for analysis in MATLAB. This structure is shown in Exhibit 3 and constitutes one row in an ASCII file. For each day, for each futures contract, the data is parsed into a separate ASCII file. The final class of data used in this article is *static data*, which comprise system parameters that vary on a future-by-future or contract-by-contract basis, such as tick-size, implied quoting functionality, number of price levels in the LOB, and so on. (Chicago Mercantile Exchange [2013b]).

LOB Rebuild

Rebuilding the LOB is the process of taking the broadcast data and regenerating the multi-dimensional LOB. Fields 1-9 of Exhibit 3 are required for the basic rebuild of the LOB, and fields 7 and 10-15 are required to carry out trade matching. For GLOBEX, the LOB is reset weekly with the last data sent on Friday and the LOB being blank at Sunday start-up.

The deterministic rebuild process for GLOBEX is shown in Algorithm 1, as per Christensen et al. [2013]. The three dimensions of the LOB are side $\widetilde{S}$ (bid/ask), class (price/size/number of orders) and M price levels, such that $m = 1, \ldots, M$. dV is a vector size $1 \times T$ of the

EXHIBIT 3
GLOBEX Data Structure

#	FIX Tag	Name	Example Values	Description
1	1023	MDPriceLevel	$1, 2, \ldots, 10$	Price level of the LOB
2	52	SendingTime	HH:MM:SS.FFF	Time of message transmission in UTC
3	269 & 270	BidPrice	1342.75	Bid price
4	269 & 271	BidSize	1	Bid size
5	269 & 270	AskPrice	1343.25	Ask price
6	269 & 271	AskSize	9	Ask size
7	279	MDUpdateAction	0, 1, 2	Type of market data update action. 0=new, 1=change, 2=delete.
8	346	NumberOfOrders	500	Number of orders in the LOB at that price level. Supported for explicit orders only.
9	276	QuoteCondition	K	K=implied, absent=explicit.
10	269 & 270	TradePrice	1343.00	Trade price
11	269 & 271	TradeSize	3	Trade size
12	277	TradeCondition	E,1	E=opening trade, 1=trade causes no price or volume change in the LOB. Broadcast price and volume are meaningless.
13	5797	AggressorSide	1, 2	Indicates which side is aggressor of the trade. 1=buy (trade volume comes out of the ask side of the LOB), 2=sell (trade volume comes out of the bid side of the LOB), absent=no aggressor (as per pre-open). Not sent in an implied spread or outright trade message.
14	336	TradingSessionId	0, 1, 2	Market state identifier. 0=pre-opening, 1=opening, 2=continuous trading.
15	5799	MatchEventIndicator	{SP H1-F1;H1;F1}	The legs of a strategy trade. Partially parsed from the FIX message structure.

changes in volume at each time step. $dV < 0$ relates to either trades, size-reducing order modifications, or order cancellations.

THE STRUCTURE OF THE GLOBEX LIMIT ORDER BOOK

On GLOBEX, two types of futures contract exist, each with their own LOB: single-leg (outright contracts) and multi-leg (strategy contracts). An example of a double-leg intra-product contract is a calendar spread. Additionally, on GLOBEX, two types of order exist: explicit orders and implied orders. Explicit orders are orders entered into the LOB by market participants. Implied orders are orders entered into the LOB by the trading system itself as a result of no-arbitrage arguments between single-leg contracts and multi-leg contracts. Implied orders exist to increase the market's liquidity and shift the high liquidity found at the front of the forward curve back down toward the less liquid contracts (Overdahl [2011], Blank [2007]). As implied pricing depends on an active futures curve, implied functionality is present on a futures-specific basis (CME [2013a]). For example, there is no implied functionality in the equity index or FX sectors, but there is in the interest rates and agricultural sectors.

The LOB on GLOBEX is a *combination* of two "sub-books"—an explicit order book and an implied order book. The implied book is limited to being up to two levels deep, while the explicit book is limited to being up to 10 levels deep, both on a security-specific basis. In the broadcast exchange data feed, implied and explicit orders can be distinguished. The process by which the two sub-books are combined is based on the premise that the explicit sub-book has higher priority than the implied sub-book. In the case of ES, there is no implied pricing, so the LOB is equal to the explicit sub-book.

The process by which the LOB is constructed has consequences for later analysis of the LOB. The simplest approach to rebuilding the LOB is a purely *deterministic* implementation of exchange rules. On GLOBEX, this rebuild approach results in volumes being aggregated at each price level, or the L2H model. Another possible approach to rebuilding is a *probabilistic* approach whereby the unaggregated volumes are inferred, or the L3 model (Christensen et al. [2013]). By unaggregating volumes, an extra dimension is introduced into the LOB, giving side, class, price level, and size level. The ability to see the detail of individual orders is especially beneficial to market makers. For example, volume allocation from pro-rata match engines is conditional on the relative sizes of the individual orders at a price level, so a liquidity provider can maximize an allocation by knowing the L3 structure (Janecek and Kabrhel [2007]). The hidden volume considered in this article can be thought of as occupying a new dimension, a hidden level, or the L2H model. L2H shows the hidden volume at the price level and the aggregated visible volume, with the exception of the peak of the hidden volume, which is unaggregated and visible.

Alg.1 GLOBEX LOB Rebuild Algorithm. $[LOB,\ dV] = \text{Rebuild(FIX)}$

$LOB(side, class, priceLevel) \leftarrow 0$ {Initialize the empty LOB}
$dV \leftarrow 0$ {Initialize the vector. $dV = dV_1, \ldots, dV_T$}
for $t = 1 : T$ **do**
  {Only process orders, not trades}
  if order **then**
    {Extract price, volume, side, price level and number of orders}
    $[P, V, \tilde{S}, m, n] = FIX_t$
    if (MDUpdateAction == 0) **then**
      Insert a new price level
      $dV_t = V$
      for $i = m : M - 1$ **do**
        {Shift existing levels down by 1}
        $LOB(\tilde{S}, 1, i+1) = LOB(\tilde{S}, 1, i)$
        $LOB(\tilde{S}, 2, i+1) = LOB(\tilde{S}, 2, i)$
        $LOB(\tilde{S}, 3, i+1) = LOB(\tilde{S}, 3, i)$
      end for
      $LOB(\tilde{S}, 1, m) = P$ {Assign price}
      $LOB(\tilde{S}, 2, m) = V$ {Assign size}
      $LOB(\tilde{S}, 3, m) = n$ {Assign nu. of orders}
    else if (MDUpdateAction == 1) **then**
      Change an existing price level
      $dV_t = V - LOB(\tilde{S}, 2, m)$
      $LOB(\tilde{S}, 1, m) = P$ {Assign price}
      $LOB(\tilde{S}, 2, m) = V$ {Assign size}
      $LOB(\tilde{S}, 3, m) = n$ {Assign nu. of orders}
    else if (MDUpdateAction == 2) **then**
      Delete an existing price level
      $dV_t = -V$
      for $i = m : M - 1$ **do**
        {Shift existing levels up by 1}
        $LOB(\tilde{S}, 1, i) = LOB(\tilde{S}, 1, i+1)$
        $LOB(\tilde{S}, 2, i) = LOB(\tilde{S}, 2, i+1)$
        $LOB(\tilde{S}, 3, i) = LOB(\tilde{S}, 3, i+1)$
      end for
      $LOB(\tilde{S}, 1, M) = 0$ {Nullify last price level}
      $LOB(\tilde{S}, 2, M) = 0$
      $LOB(\tilde{S}, 3, M) = 0$
    end if
  end if
end for

These three representations of the same LOB are shown in Exhibit 4. The L2 view shows the aggregated volume at the price level, as broadcast by GLOBEX. The L3 view shows the inferred unaggregated volume at the price level. The L2H view shows the aggregated volume with an iceberg order present. In this example, the peak of the iceberg is equal to 25. In all three views the *visible* volume is equal to 50 lots. In the L2 and L3 views the *realizable* volume is equal to 50 lots, however in the L2H view the realizable volume is equal to 175 lots due to the presence of the hidden volume (one visible peak at 25 lots and six hidden peaks at 25 lots). The order at bottom of the bar with S = 25 has time-priority, however in the L2H case, all subsequent tranches of the iceberg order have time priority over the S = 15 order.

HIDDEN LIQUIDITY

There are three reasons why the LOB may not display the "true" liquidity state of the market: *iceberg* orders, *unbroadcast* orders, and *phantom* orders. All these order types cause the *realizable* volume of the LOB to differ from the displayed volume.

Iceberg Orders

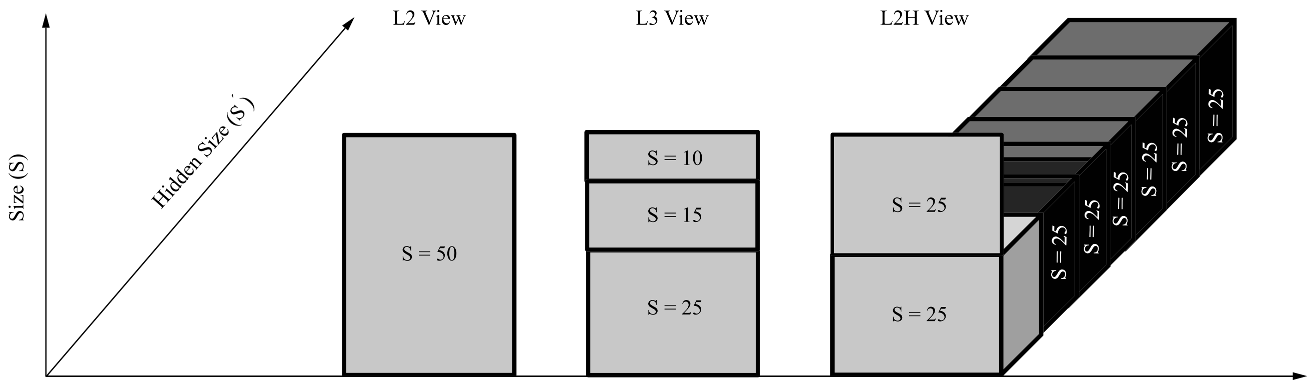
Iceberg orders are an order type supported by GLOBEX. An iceberg order is a special type of limit order, where in addition to a price, side, and size, the user is required to specify a *max show* value. Max show is the upper size limit of the fraction of the total order size that is shown to the market, while the remainder of the order volume remains hidden. When the displayed quantity has been filled, another portion *less than or equal to* the displaced quantity is then displayed, with the time priority of the *initial* peak and the remaining hidden volume reduced by the peak size. This is notably different from a trader constructing his or her own iceberg order system, as a sequential series of limit orders would not have the time priority of the first order. Hence iceberg orders are particularly important for latency, sensitive markets where there is a time component to the matching algorithm, such as the equity index and FX sectors (Chicago Mercantile Exchange [2013b]).

Iceberg orders are a way of limiting information flow, their prime reason for existing being to facilitate large trade execution. This is done by preventing market makers from noticing the large incoming order and changing the price in anticipation of it, thereby reducing the market impact of the trade. However, iceberg orders are controversial. They diminish the benefits of transparent, order-driven markets, including price efficiency, low costs of market monitoring, and less information asymmetries (Madhavan [2000]). If the iceberg allows a trader to avoid the informational disadvantage associated

EXHIBIT 4

Three Views of the LOB, Each Showing One Price Level

In all three views, the *visible* volume is equal to 50 lots. In the L2H view the *Realizable* volume is equal to 175 lots.



with limit orders over market orders, why are icebergs not always used? The answer might be that liquidity suppliers sometimes wish to influence the LOB in some way to their advantage. Approaches such as destabilizing the LOB, "moving the market," and phantom orders are legal gray areas.

GLOBEX refuses to quantify any statistics relating to iceberg orders, but has said they are "popular" and that due to icebergs, the "true liquidity in CME Group markets is generally much superior to displayed liquidity" CME [2011].

Unbroadcast Orders

Some types of implied orders *are* eligible to be filled but are *not* broadcast in the market data feed CME [2013b]. This is significant because it means that there is liquidity in the LOB which cannot be seen, meaning there is the potential for a fill at a price level in the LOB where no order was seen to sit. These unbroadcast orders occur only in the contracts with implied pricing and then only in the first two price levels of the LOB. While the implied book is just two levels deep, it has the potential to have a far greater update rate.

Phantom Orders

A phantom order is one to which a trader is not committed or indeed which a trader does not intend ever to execute (Burghardt et al. [2006]). Phantom orders are generally considered bad for the market, as they give the impression of more liquidity than there actually is. These orders can be hit only by being able to act faster than the trader who placed them, and so are termed "negative liquidity." Phantom orders are characterized by rapid-fire submission followed by cancellation, with quoting rates of up to 20KHz being observed in highly illiquid securities (Hunsader [2010]). Borkovec defines phantom volume as limit orders added and canceled from the LOB in a period of less than two seconds and finds that this is 10% of all orders for equities data from NYSE ARCA in 2005 Borkovec et al. [2012]. GLOBEX limits this behavior by regulating the message-to-volume ratio submitted by a particular trader, where a message includes an order, modification, or cancel. This ratio ranges from 4 (for ES) to 60 (for less liquid futures). The messaging policy is an aggregated average over a 24-hour window, and if a trader exceeds the ratio, he or she is fined (Chicago Mercantile Exchange [2013b]). The reasons for submitting phantom orders are unclear but may include manipulating/destabilizing the LOB or detecting hidden liquidity.

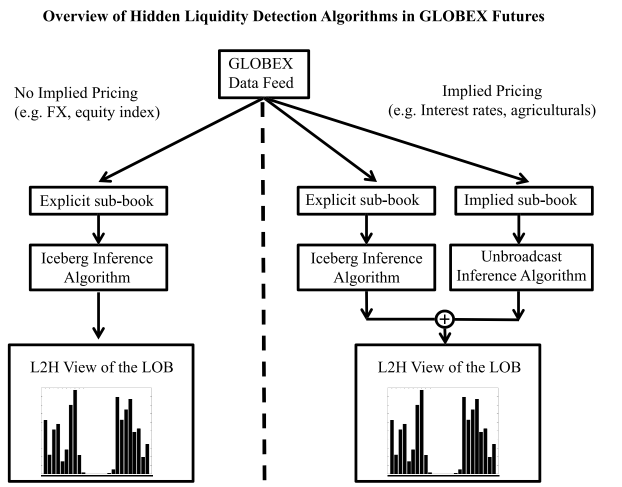
Summary of Hidden Volumes

Phantom orders are not *truly* realizable volume and so are not considered further. Realizable hidden volume on GLOBEX futures takes two forms—iceberg orders and unbroadcast orders. In order to detect all the hidden volume in the LOB of GLOBEX futures, the approach outlined in Exhibit 5 can be applied. As the LOB is built by combining the sub-books of implied and explicit

E X H I B I T 5

Schematic of Prediction Algorithms for GLOBEX Hidden Volume

Conditional on implied pricing, iceberg and unbroadcast algorithms are run independently or in parallel.



orders, the iceberg algorithm can be applied to the explicit sub-book to detect all the hidden liquidity present in futures which do not support implied orders. For futures which do support implied pricing, the iceberg algorithm needs to be run in parallel with an unbroadcast algorithm, which detects the hidden volume in the implied book. In this article we just consider the hidden volume resulting from iceberg orders and plan to publish research on the unbroadcast hidden volume in a subsequent article.

LITERATURE REVIEW OF ICEBERG DETECTION

An extensive literature on detection of iceberg orders exists, which falls into the three categories of active algorithms, model-based algorithms, and frequentist algorithms. These categories of detection algorithm are now reviewed.

Active Algorithms

An active algorithm seeks to detect iceberg orders by "pinging" the LOB with orders that the participant never intends to be filled. In Hasbrouck and Saar [2001] the authors note that hidden orders constitute 3% of all submitted limit orders but account for 12% of all executions, while more than 25% of all limit orders submitted are canceled within two seconds of submission. They propose that these fleeting orders are likely used by aggressive traders searching for hidden orders. For example, a buyer might submit an order priced just short of the ask quote, hoping to trade against any hidden sell orders. In this view, a fleeting limit order represents a liquidity demander, rather than a supplier.

Durbin [2010] suggests detecting icebergs by use of fill or kill (FOK) limit orders. On GLOBEX, FOK are canceled if not immediately filled for the specified minimum quantity at the specified price or better. By submitting small FOK orders over a range of prices, the presence of hidden orders can be detected by whether the order is filled or not.

Model-Based Algorithms

Both De Winne and D'hondt [2007] and Bessembinder et al. [2009] use regression models on 2002–2003 Euronext equities data. This data set allows hidden depth to be directly observed. De Winne finds that more than 45% of the depth at the top five levels of the LOB is hidden and that iceberg order size is six times greater than a normal order. Bessembinder finds that 18% of incoming orders include some hidden size, 44% of order volume is hidden, and hidden orders are more common in illiquid issues and for large trade sizes and when order arrival rates are low.

Hautsch and Huang [2009] build a Bayesian model with a Bernoulli likelihood function using logit multiple regression, where the probability of an order being executed with hidden liquidity can be predicted by eight predictors, including distance from the mid price, the size of the spread, and the lagged return.

Avellaneda et al. [2011] try to match the empirically observed LOB mechanics by including an implied hidden volume term in a stochastic diffusion model. The approach allows the implied hidden liquidity of different securities to be compared. For four securities on three exchanges, the results show differences of more than 220%.

Esser and Mönch [2007] consider the case for exchanges where the iceberg peak loses time priority after each execution and generate a stochastic model for the optimal peak size of an iceberg order. The authors model price by a jump-diffusion process, and when the diffusion process hits an iceberg order, a jump in price occurs, leading to a probabilistic model for the peak size. They conclude that 8% of the LOB volume is hidden.

Fleming and Mizrach [2009] consider U.S. Treasury data from the inter-dealer platform BrokerTec, using a model for the LOB which incorporates hidden volume Moinas [2006]. The authors observe that the percentage of executed hidden volume is low, at 2%, and that this low figure masks the fact that there is usually no hidden depth, but when there is hidden depth, it is substantial. The authors also observe that the pattern of hidden depth differs from that of visible depth, having the largest volume at the first price level for most maturities, while the visible volume is greatest a few levels out.

Frequentist Algorithms

In Burghardt et al. [2006], the authors consider the concept of "sweep to fill," whereby a large trader clears out all volume from the LOB. The authors compare the sweep to fill average price with the observed VWAP market impact and note that the impact of VWAP tends to be smaller than sweep to fill measures would suggest, meaning that the LOB must be more liquid than it seems, with impact factors differing between 4% and 10%, depending on the order size. The authors postulate that this extra liquidity exists in the form of iceberg orders in the LOB.

Borkovec [2012] takes an approach that is based on estimating the true liquidity environment by generating joint probability distributions of intraday volume profiles and various predictors (for example, spread, volatility, and depth). Hidden volumes are found by trying to match trades to quotes. Hidden orders are found to be 53% larger than the visible orders.

Frey and Sandas [2009] develop an empirical frequentist approach for detecting hidden volume using LOB data, which relies on the fact that the peak size is constant and that the time stamp for resupply is the same as the time stamp for the executed volume. On XETRA, time priority is *lost* between successive peaks, so each peak goes to the back of the queue. The authors apply their approach to XETRA, DAX 30 equities data from 2004 and find that iceberg orders make up 9% of all orders and 16% of all executed volume, and that the average number of tranches is five. In terms of size, iceberg orders are on average 12 to 20 times larger than visible limit orders and have a peak size that is 2.5 times larger than visible limit orders.

Summary of Literature Review

From the literature review, we find a lack of algorithms which could be applied online for hidden volume prediction on GLOBEX. The literature review is summarized in Exhibit 6.

PREDICTING ICEBERGS ON GLOBEX

In this section we present our model for detecting and predicting iceberg orders on GLOBEX. In summary, initially an iceberg order is indistinguishable from a limit order, with the same price and size. But over time, the execution of displayed peaks and subsequent display of new peaks allow market participants to learn about its existence and predict its size with increasing accuracy.

EXHIBIT 6
Literature Review on Modeling Iceberg Orders

Reference	Exchange/ Platform	Data	Period	Volume Hidden (%)
Burghardt et al. [2006]	GLOBEX	ES future	Jan–Apr 2006	4-10
Avellaneda et al. [2011]	NASDAQ, NYSE, BATS	4 ETF's/equities	Jan 2010	Relative
Bessembinder et al. [2009]	Euronext Paris	100 equities	Apr 2003	44
Esser and Mönch [2007]	XETRA	1 equity	Jan–Mar 2002	8
Moro et al. [2009] Vaglica et al. [2008]	SETS, SIBE	97 equities	Jan 2001–Dec 2004	52
Tuttle [2005]	NASDAQ	97 equities	4 weeks 2001, 2002	25
De Winne and D'hondt [2007]	Euronext Paris	CAC 40, equities	Oct–Dec 2002	45
Borkovec et al. [2012]	NYSE ARCA	329 equities	Jun–Aug 2005	3
Frey and Sandas [2009]	XETRA	DAX 30, equities	Jan–Mar 2004	16
Fleming and Mizrach [2009]	ICAP BrokerTec	US Treasuries	Jan 2001–Feb 2006	2
Hasbrouck and Saar [2001]	Island ECN	300 equities	Oct–Dec 1999	12
Hautsch and Huang [2009]	NASDAQ	7 equities	Jan 2009	14
Yao [2012]	NASDAQ	2,390 equities	Jan 2010–Nov 2011	16

Iceberg Mechanics on GLOBEX

The mechanics of how GLOBEX icebergs operate are now presented. On GLOBEX, when a trader enters an iceberg order, he or she is required to specify side $\widetilde{S}$, limit price P, the total order size V, and max show Ψ (a fraction of the total size).[4] A limit price means that the order must be filled at a price *at least as good* as the specified price. In the data feed, this means that an aggressor order that fills a resting limit order need *not* show a trade price equal to the resting limit order price. Depending on the side of the LOB, there will be a greater (less) than or equal to condition $\gtreqless$. When V is exactly divisible by Ψ, the size of the iceberg peak ψ is equal to the max show size. When V is not exactly divisible by Ψ, the final tranche is equal to the remainder. When the iceberg order enters the LOB, it displays only a portion of the order to the marketplace (the peak), while the rest of the iceberg remains hidden. When the displayed quantity has been filled, another peak is then displayed, with the time priority of the *initial* peak and the remaining hidden volume reduced by the peak size. While GLOBEX *retains* time priority between successive peaks, this is not the case for all exchanges; for example, XETRA loses time priority between peaks. Due to this conservation of time priority, icebergs are most popular on markets that have a time component in their match algorithm (such as the FIFO algorithm used by ES), as opposed to markets that do not have a time component (for example, the pro-rata algorithm used by eurodollar). When pro-rata matching does occur, the match algorithm considers only ψ and not V in making its allocation. Iceberg order execution is shown in Algorithm 2.

In the broadcast data feed, there is no *simple* way to distinguish an iceberg order from any other order. The only way to know if an order is part of an iceberg is if you had placed the order yourself or had access to private exchange information. However, a key feature of the data structure allows us to *infer* the presence of iceberg orders in the broadcast data feed. This feature is that LOB update messages are broadcast *post* trades happening, albeit with a time lag due to system latency.

The mechanics associated with broadcast messages at the point of iceberg order execution are at the center of this article. In the simplest case, when the peak of an iceberg order is filled by a trade, three messages are seen in the data feed: first, a trade message, with associated size; second, an LOB update message, from which the

Alg. 2 GLOBEX Iceberg Execution Algorithm

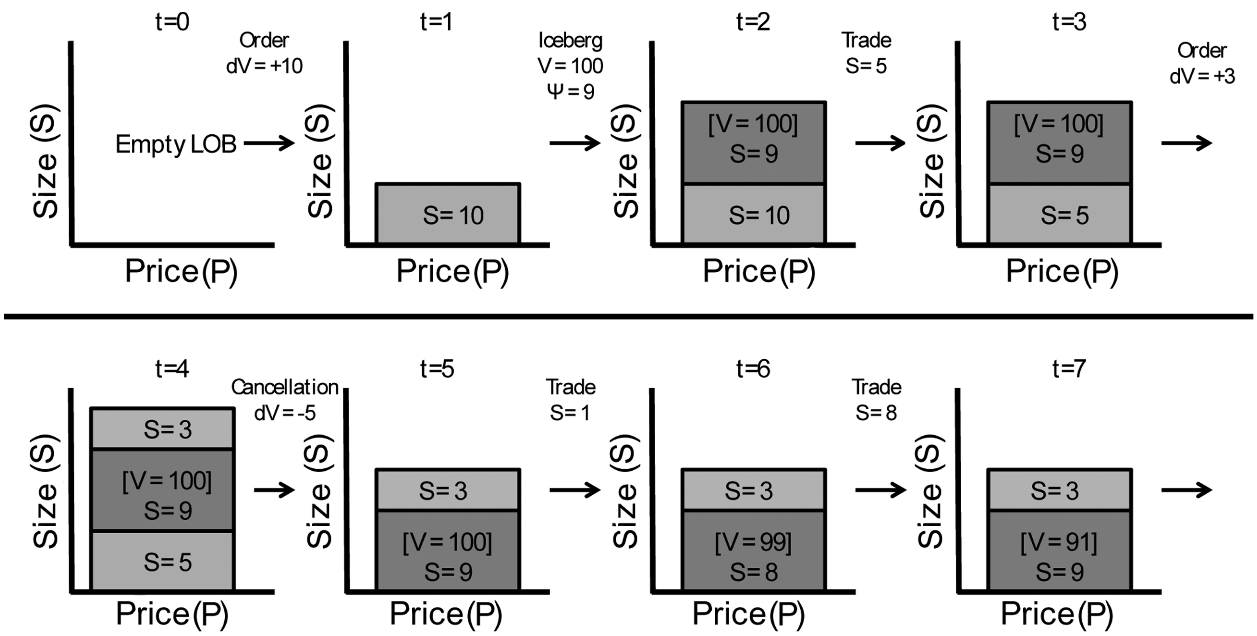
```
Inputs(V, Ψ)          {Total order size, max show}
N = ⌈V/Ψ⌉             {Number of tranches}
for n = 1 to N do
    if (n == N) then
        ψ = V         {Assign the peak size}
    else
        ψ = Ψ
    end if
    v_n = ψ           {Place the order into the LOB}
    while v_n > 0 do
        v_n ← trade   {Match volume from trade(s)}
    end while
    V = V − ψ         {Reduce the hidden volume by the peak}
end for
```

decrease in volume dV can be inferred; third, an LOB update message replenishing the peak.

These mechanics are now illustrated with an example, as shown in Exhibit 7. An iceberg order is specified with a total size $V = 100$ and a max show $\Psi = 9$. The first 11 tranches of the iceberg are of size $\psi = 9$ and the final tranche is size $\bar{\psi} = 1$, so $N = 12$ (where N is the number of tranches). At time step $t = 6$, a trade message for $S = 8$ is seen in the data feed. This results in two further messages being sent by GLOBEX, firstly, a LOB update message $dV = -8$, and secondly a peak replenish message $dV = 9$. The first of these messages is the trade volume being removed from the LOB. The second of these messages is the next tranche of the iceberg order being placed into the LOB. It this peak replenish message time dt after a trade which allows the iceberg order to be detected. The rules associated with what messages are broadcast become more complex than in Exhibit 7 when the trade size is greater than Ψ. In summary, in the event of a large trade, not all the peak replenish messages are broadcast. For example, if there is an iceberg order to buy ($V = 100$, $\Psi = 10$) currently showing $\psi = 10$, and there is a sell side aggressor order of size 30, there will be three trade messages (3×10 lots), but *no* peak replenish message, because the displayed peak quantity will remain the same, $\psi = 10$. In a second example, if there is an iceberg order to buy ($V = 100$; $\Psi = 10$) currently showing $\psi = 10$, and there is a sell side aggressor order of size 32, there will be four trade messages (3×10 lots, 1×2 lots), and a peak replenish message of size 8, which is the displayed peak quantity

Exhibit 7
Schematic of Iceberg Order Mechanics.

A simplified LOB consisting of just one price level is shown progressing through time. At each step a FIX message is applied to the LOB. The bottom of stack has the highest time priority. Normal limit orders are shown in light greys, iceberg order in dark grey. Square brackets around *V* mean the value cannot be seen in the broadcast data.



$\psi = 8$. These two examples can be explained by the fact that the GLOBEX incremental LOB management rules send out update messages only when the external characteristics of the LOB change. This means that the presence of iceberg orders can result in the LOB volume not changing after trades have been executed.[5]

As for any limit order, iceberg orders can be modified after submission. If Ψ is modified by being increased in size after being submitted, then the order currently shown retains the old max show, and once that has been filled, thereafter shows the new value, while the time priority of the iceberg is maintained. However, if the modification is a decrease in Ψ, then the iceberg order loses its time priority.

Trade Size Descriptive Statistics

The variations in the message broadcast rules mean that our approach needs to be conditional on trade size. In Exhibit 8, descriptive statistics for trade size in the front month ES contract for the period January 1, 2011, through December 31, 2011, are presented. For this period, the mean number of trades per day was 0.5 million, while the mean volume per day was 2.2 million lots. Trade size approximately follows an exponential distribution, with most trades being small and a few being very large. This prior distribution of trade sizes leads to a way of classifying a trade as "normal" or "large," based on the 99th percentile. If an order at time $t - 1$ meets the conditions to be a viable iceberg order, and at time t a large trade occurs, then the large trade may lead to more complex iceberg mechanics.

Exhibit 8
ES Trade Size Statistics (lots)

Statistic	Value	Statistic	Value
Mean	5	Maximum	1,069
Mode	1	95th Percentile	21
Median	2	97.5th Percentile	34
Minimum	1	99th Percentile	53

Prediction Algorithm Overview

Our algorithm has two phases, a learning phase and an online inference phase. While the learning phase *could* be online, we set it to be offline for reasons of simplicity and computational latency.

1. Offline learning. The algorithm carries out a forward pass to identify likely icebergs. The resulting identified orders are then used to generate the distributional parameter $\mathbf{\Theta}$. This is done ex post, using the previous H days of historical data.
2. Online inference. This phase uses the output $\mathbf{\Theta}$ from the learning phase during a forward pass. The output $\mathbf{\Gamma}$ is an online estimate for the existence of icebergs.

The forward pass algorithm is so called as it proceeds forwards in time.

Variables Used in the Algorithm

In this section, the variables used in the algorithm are defined. The latent variables in the problem are max show $\mathbf{\Psi}$ and total order size V. The set of K icebergs is denoted by $\mathbf{\Phi} = \{\mathbf{\Phi}_1, \ldots, \mathbf{\Phi}_k, \ldots, \mathbf{\Phi}_K\}$, where K is the total number of iceberg orders seen in one trading day. $\mathbf{\Phi}_k$ denotes a single iceberg order such that $\mathbf{\Phi}_k = \{\phi_{k,1}, \ldots, \phi_{k,n}, \ldots, \phi_{k,N}\}$. N is the number of tranches within a single iceberg order. The variables are summarized in Exhibit 9.

The inference stage outputs the variables shown in Exhibit 10. The variable $\mathbf{\Gamma}$ dynamically updates over time and can be examined at any point in time to see the parameters of the estimated icebergs in the LOB. $\mathbf{\Gamma}$ is the final output of the algorithm.

Detection Algorithm

In this section, the logic behind how the algorithm works is described. The algorithm applies a pattern recognition approach which seeks out a certain combination of events that allows an iceberg order to be identified and tracked, and thus predicted.

The only way an iceberg order can be distinguished from a normal limit order is when a *trade* causes the peak

EXHIBIT 9
Learning Phase Variables

Variable	Description
k	The number of iceberg orders seen in one trading day for a given security. $k = \{1, \ldots, K\}$.
Φ_k	Unique identifier for this iceberg order. $\Phi_k = \left\{\tilde{S}, P, \Psi, \alpha\right\}$.
$\tilde{S}$	The side of the LOB. Is the iceberg a buy or sell order?
m	Price level in the LOB. $m = 1, \ldots, M$.
P	Limit price of the iceberg order.
α	Is the iceberg currently active or not? 0=false, 1=true.
ω	If the iceberg is inactive, was it cancelled or filled? 0=cancelled, 1=filled.
Ψ	Max show. The upper size limit of the iceberg peak.
$\phi_{k,n}$	The n^{th} tranche of iceberg order Φ_k. $\phi_{k,n} = \{\psi, t, \lambda, \theta\}$.
n	The number of tranches of an iceberg order. $n = \{1, \ldots, N\}$.
ψ	Peak size of tranche. The final tranche peak size is a special case and denoted by $\bar{\psi}$.
t	Time of tranche (HH:MM:SS.FFF). $t = 1, \ldots, T$.
λ	Time between the trade and the publication of the refresh message. In milliseconds.
θ	Time between tranches. $\theta = \phi_{k,n}(t) - \phi_{k,n-1}(t)$. In milliseconds.

EXHIBIT 10
Inference Phase Outputs

Variable	Description
$\hat{V}_k$	The estimated total size of the k^{th} iceberg (in lots).
$\hat{\Psi}_k$	The set of peak sizes of the k^{th} iceberg (in lots). $\hat{\Psi} = \left\{\hat{\psi}_1, \ldots, \hat{\psi}_n, \ldots, \hat{\bar{\psi}}_N\right\}$.
$\mathbf{\Gamma}$	The set of the inference phase outputs. $\mathbf{\Gamma} = \left\{\hat{V}, \hat{\Psi}\right\}$.

of an iceberg order to be filled, and then GLOBEX automatically refreshes the peak in the LOB. This automatic refresh is indistinguishable from a new order submission. The *initial* detection signal is defined as an order placed after a trade that satisfies a list of conditions:

- The order must be submitted within dt of the trade occurring, due to internal latency of the GLOBEX system. dt is set equal to 300 milliseconds. If dt is too large, it is likely that non-iceberg orders will fall inside the window. If too small, iceberg orders may be missed.
- The trade price must satisfy the limit order price.
- The change in order size must be positive (i.e., not a size reducing modification or a cancellation).
- An implied order can never be an iceberg.
- As the trade volume must have been taken from the first price level $m = 1$, the order must add volume back to the first price level.
- The side of the trade must be equal to the side of the order update. For example, when the aggressor trade is a buy, the trade volume comes out of the ask side of the LOB.

The inference is that an order that meets these criteria might be the peak of an iceberg. This methodology cannot detect iceberg orders in the depth levels, as by definition trades occur only at the first price level $m = 1$. An exception to this is when an iceberg has been partially executed and the price moves away, leaving the iceberg in a depth position in the LOB.

The next step of the algorithm is to check whether this is the first tranche of a new iceberg, or an additional tranche of an existing iceberg. This is complicated by the ability of the refresh message to change in size, conditional on the trade size. Four cases can occur:

- If the trade size is less than Ψ, the refresh order size must be greater than or equal to the trade size.
- If the trade size is equal to Ψ, or multiples thereof, then no refresh order is seen.
- If the trade size is greater than Ψ, then the refresh order size is equal to the modulus after division. If a large trade occurs *and* the refresh message is the first peak $\phi_{k,1}$ seen for that iceberg, then a special case occurs. It is assumed that the iceberg order peak refresh message is not equal to the max show $\psi_1 \neq \Psi$, as would be the case for a normal trade size, as the large trade size will be greater than the max show. Instead, it is assumed that $\psi_1 \leq \Psi$. This is incorporated by not setting the max show for the iceberg equal to the first peak seen.
- The final tranche of the iceberg order is a special case, as it may be less than or equal to the max show, $\bar{\psi} \leq \Psi$. This occurs when V is not exactly divisible by Ψ.

The binary indicator variable α denotes whether an iceberg is currently active by tracking the cumulative volume traded at each specific side/price (i.e. Φ_k) since the last tranche was seen. If the cumulative volume at Φ_k exceeds the max show (plus dt to allow for system latency) and no refresh message has appeared, $\alpha = 0$. This allows the algorithm to detect multiple iceberg orders with the same price/side/max show combinations. The approach is made possible by the fact that on GLOBEX, time priority is retained between tranches. If the activity test comes back with $\alpha = 0$, there are three possibilities:

- The suspected iceberg was not really an iceberg to begin with (a false positive). This is defined as: if when $\alpha = 0$, $\phi_{k,n}$ has $n = 1$, then Φ_k is removed from Φ, i.e., if only one tranche of a possible iceberg order was seen and then the iceberg was not seen again, we say it was coincidence that an order was submitted in time window dt that met assorted criteria. If Φ_k was not deleted from Φ, then the parameter set would be distorted by the presence of non-iceberg limit orders.
- The iceberg order was canceled. As with any limit order, cancellation of an iceberg order can occur at any time before being fully filled. The cancellation message is *seen* to apply to just the displayed tranche, $|dV| \leq \Psi$; however, it actually *applies* to the remaining volume of the iceberg. Search the time between $\phi_{k,n}$ and the cumulative volume exceeding Ψ at the price/side of interest. Set $\omega = 0$.
- The iceberg order was filled in its entirety. If neither of the other possibilities are true, this is taken to be the case. Set $\omega = 1$.

Initial detection of an iceberg order occurs by first looking for an order within a window dt and with certain size constraints. The subsequent tranches are further constrained by P, $\widetilde{S}$, and Φ; however, there is still the possibility that n sequential orders could have been seen which meet the classification requirements. The probability that $\phi_{k,1}$ is not really an iceberg order is reasonably high, while the probability that $\phi_{k,2}$ is not really an iceberg order is lower, and so on. The probability of sequential false detections is *mutually independent.* This compounding of probabilities is shown in Equation (1)

$$p(\Phi_k) = 1 - \left(\prod_{n}^{N} p(\phi_{k,n}) \right) \approx \left(1 - e^{-(\beta \times n)}\right), \ \beta > 0 \tag{1}$$

The detection algorithm assumes that for Φ_k, once $n > 1$, the order is an iceberg. Hence, our criteria of $n > 1$ is an approximation. This is shown graphically in Exhibit 11. Subplot one shows the probability of detection conditional on the number of tranches seen, according to Equation (1). Subplot two shows this probability according to Algorithm 6. Both examples use an iceberg with ten tranches.

A summary of the forward pass is shown in Algorithm 6. This algorithm is run in offline (learning) and online (inference) modes. The variable "data" is the data structure from Exhibit 3, and dV is the change in volume output by Algorithm 1. Only for online mode is $\mathbf{\Theta}$ required.

We are now at the stage where we have generated $\mathbf{\Phi}$. In the next two sections, the details of generating $\mathbf{\Theta}$ from learning and $\mathbf{\Gamma}$ from inference are presented.

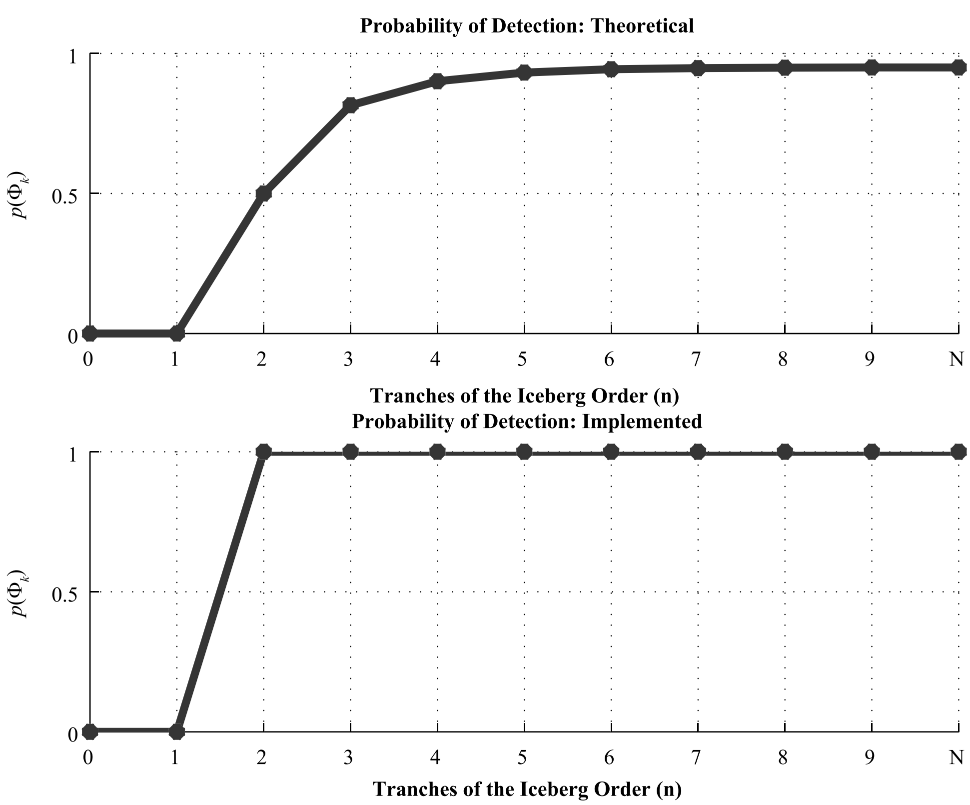
Learning Phase: Generating Θ

In this section, we describe the process of generating $\mathbf{\Theta}$ from $\mathbf{\Phi}$. Our main interest in iceberg orders is being able to *predict* the existence of hidden volume. To do this, information on the prior behavior of iceberg orders is used. It is hypothesized that participants

Exhibit 11

Graphical Representation of the Probability of Detection

$p(\Phi_k)$ is the probability that the order Φ_k is an iceberg. $p(\Phi_k) = 1$ [0] means it is [is not] an iceberg.



who submit icebergs do not do so randomly, but display repetitive behavior when they submit these orders, particularly in terms of the ratio $V/\mathbf{\Psi}$. The learning phase captures these statistically significant relationships from historical data, allowing prediction of the remaining iceberg order once it has been detected.

In order to carry out learning, we have to solve a *multidimensional mass estimation* problem (Hastie et al. [2001]). In our problem, the dimensions are max show $\mathbf{\Psi}$ and total order size V. As both V and $\mathbf{\Psi}$ are discrete integers (as the minimum size that can be traded is one lot), we are interested in finding the bivariate *probability mass function*. A probability mass function is a function that describes a *discrete* probability distribution. Estimating probability mass functions with discrete variables can be straightforward. As there is only a finite number of values, the simple relative frequency of occurrence can be found. For the case of a bivariate joint distribution, a separate density could be found for each value of the first variable, while holding the second variable constant. However, this approach is practically awkward if the number of levels for the discrete variable is large compared to the number of samples. Moreover, the joint distribution problem has us estimating completely separate distributions for the second variable for every value of the first variable, without any sharing of information between them. A better solution is to *smooth* those distributions toward each other, allowing for interpolation over sparse (missing) data, while taking into account any nearby observations, or lack of them. Many methodologies exist that allow for density estimation including splines, wavelets, Fourier series, and parametric approaches. We opt to use nonparametric kernel estimation Scott [1992]. While kernels for discrete variables do exist, for example Aitchison and Aitken [1976]; Rajagopalan and Lall [1995], we use the *Gaussian kernel* on the basis of simplicity, and then, at the point of sampling the distribution, round the output to the nearest integer Haddad and Akansu [1991]. Carrying out this estimation on a large data set is computationally demanding, though as this is done off-line it does not affect trading.

Learning is carried out before the start of each trading day, using the previous H days worth of data, excluding any icebergs $\mathbf{\Phi}_k$ that were not filled $\omega = 0$. The learning phase is summarized in Algorithm 3. Bivariate Gaussian kernel density estimation is shown in Algorithm 4, where n is the number of observations in each vector and α is a Gaussian confidence limit. The output distribution $\mathbf{\Theta}$ is shown in Exhibit 12.

Alg. 3 Iceberg Off-line Learning Algorithm. $\mathbf{\Theta}$ = offline_learning($\mathbf{\Phi}$)

```
[V, Ψ] ← 0 {Initialize empty vector}
for h = 1 : H do
    Consider iceberg orders from past H days
    for k = 1 : K do
        Consider the K iceberg orders that day
        if Φ_k(ω) == 1 then
            Consider iceberg orders which were filled
            [V, Ψ] = extract(Φ_k)
        end if
    end for
end for
Θ = KDE(V, Ψ)
```

Alg. 4 Bivariate Gaussian Kernel Density Estimation. $f(x, y) = \text{KDE}(x, y)$

$$\sigma_x = \frac{1}{\alpha n^{\frac{1}{6}}} \text{median}\,(|x - \hat{x}|)$$

$$\sigma_y = \frac{1}{\alpha n^{\frac{1}{6}}} \text{median}\,(|y - \hat{y}|)$$

$$\hat{f}(x, y) = \frac{1}{n} \sum \mathcal{N}(x; \mu_x, \sigma_x^2)\mathcal{N}(y; \mu_y, \sigma_y^2)$$

Inference Phase: Generating $\mathbf{\Gamma}$

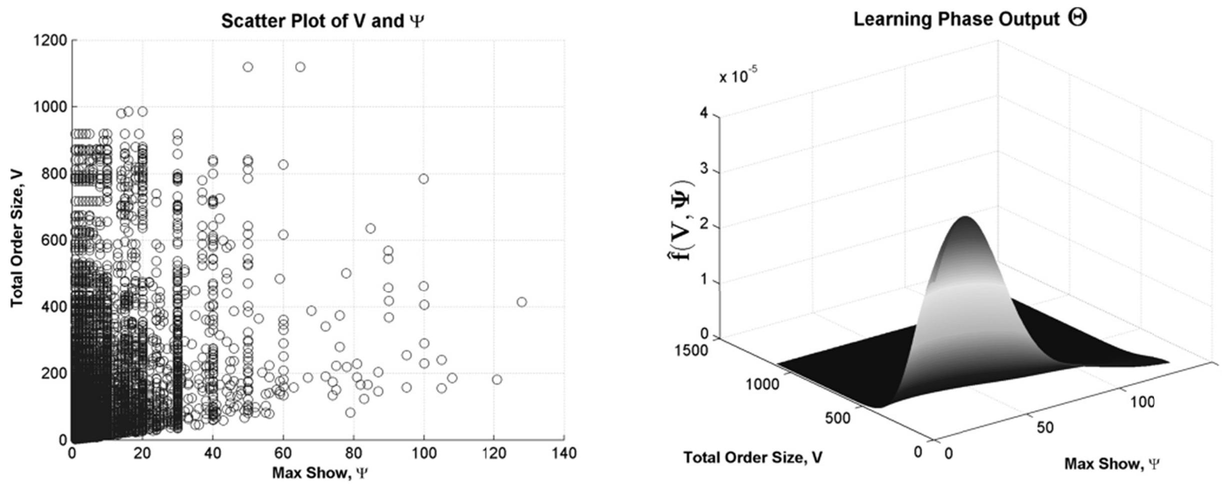
In this section, we describe the process of evaluating $\mathbf{\Theta}$ to give the prediction $\mathbf{\Gamma}$. For each iceberg $\mathbf{\Phi}_k$ in the LOB, an update rule is used to update the estimated remaining volume, given the volume seen so far. This is done by a five-step process:

1. We wish to evaluate $\mathbf{\Theta}$ for $\mathbf{\Phi}_k(\mathbf{\Psi})$, the marginal distribution. This has the effect of reducing the dimensionality of the distribution by 1, such that $f = \mathcal{G}(V)$ where $\mathcal{G}$ is some non-linear function that captures the relationship.
2. We are interested in finding the most likely value for $\hat{V}$ for this order, given a known max show $\mathbf{\Psi}$ and the knowledge that some of the volume of the iceberg order has already been filled. The filled volume is denoted by $V_{0:n}$ and the unfilled volume by $V_{n+1:N}$, so $V = V_{0:n} + V_{n+1:N}$. Hence the search space is constrained in this manner.

EXHIBIT 12

The Output from the Learning Phase, Θ from One Trading Day

It is this distribution which is evaluated during the inference phase. The distribution is generated by nonparametric kernel mass estimation, which acts to smooth the data set and separate the latent signal from the background noise. Data shown is for ES, January 10–15, 2011.



3. $\hat{V}_k$ is found by maximizing the distribution $p(\hat{V} \mid \Psi, V_{0:n})$ which is equal to maximizing the function $f = \mathcal{G}(V_{n+1:N})$.
4. Knowing $\hat{V}$ allows us to calculate the volume of the subsequent tranches, such that $\hat{V}_k = \sum_{n=1}^{N-1} \psi_n + \bar{\psi}_N$, giving $\hat{\psi}_k$, such that $\Gamma = \{\hat{V}, \hat{\Psi}\}$.
5. The forecast is updated on one of two events:

- every time a new tranche is received, $V_{0:n}$ changes, and so the maximization is reevaluated, or
- if $\Phi_k(\alpha)$ changes to become inactive.

This is a discrete empirical probabilistic update scheme, where the prior distribution is estimated from the data. The conditional probability is updated in light of the volume filled so far $\mathcal{G}(V_{n+1:N})$. The scheme is summarized in Algorithm 5 An example output shown in Exhibit 13. In this example $\Psi = 10$, $V_{1:n} = 20$, so only the distribution ≥ 21 is searched and volume which has already been executed is excluded from the search. The result of the search is $\hat{V}_k = 52$, giving $\hat{\Psi}_k = \{\psi_{1,..5} = 10, \bar{\psi}_6 = 2\}$.

COMPUTATIONAL IMPLEMENTATION

A full discussion of the details of the computational implementation is beyond the scope of this article; however, we wish to give the reader a flavor of the issues involved. The CME matching engines are physically located at the CME data center (the "colo") in Aurora, Illinois.[6] At this data center, market participants can buy "rack space" to connect to GLOBEX via GLink, the order routing interface. What hardware is inserted into the rack space is up to the individual participant.

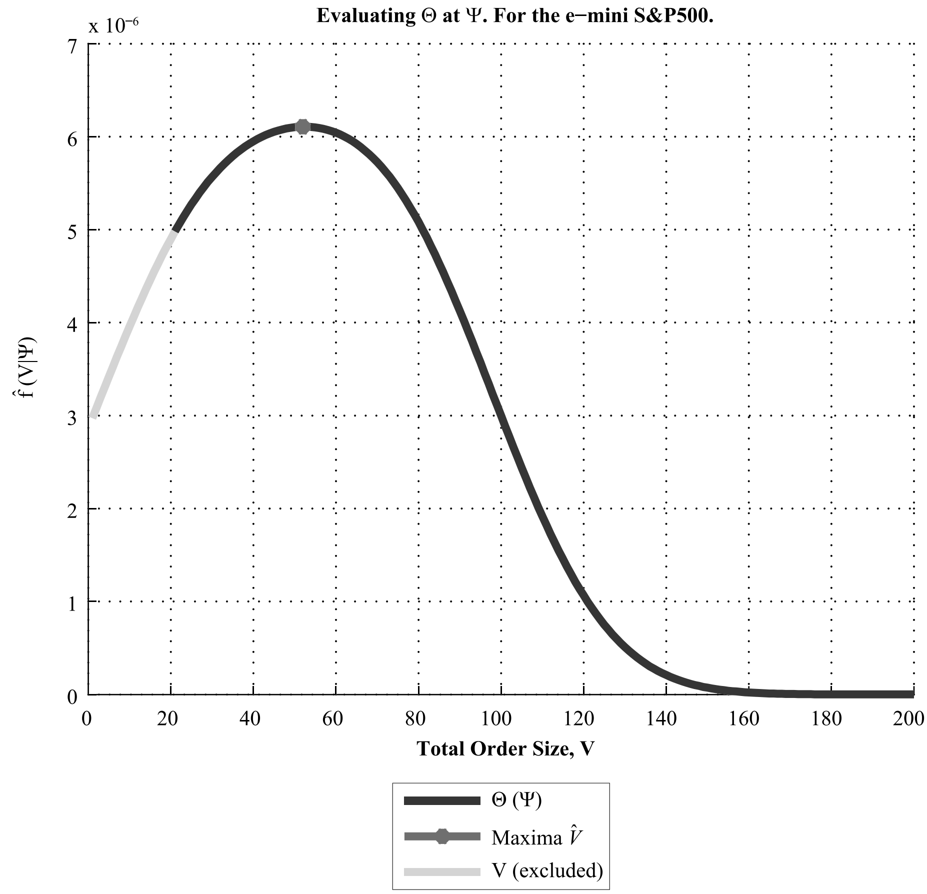
Computation of the algorithm can be split into two distinct phases: offline learning and online inference. Offline learning does not need to happen at the colo, where resources are expensive, but can take place at any server farm anywhere in the world. The learning phase is run as a daily batch process, and each day before the markets open, the latest parameter set is uploaded to the colo servers for use during inference.

Alg. 5 Iceberg On-line Inference Algorithm. Γ = online_inference(Φ, Θ)

1: **for** $k = 1 : K$ **do**
2: $\quad \mathcal{G} = \Theta(\Psi)$ {Evaluate Θ for $\Phi_k(\Psi)$}
3: $\quad \hat{V} = max(\mathcal{G}(V_{n+1:N}))$ {Maximize constrained $\mathcal{G}$}
4: $\quad \Gamma_k = \{\hat{V}, \hat{\Psi}\}$
5: **end for**

Exhibit 13

Inference by Evaluating the Distribution Θ. The Distribution is Reduced in Dimensionality to 2D as Ψ Is Known. $\hat{V}$ Is then Found by Maximizing the Constrained Distribution.



Data Storage

Before any learning can take place, the raw FIX/FAST data must be processed, as per Exhibit 3, and added to a data depository. On GLOBEX, we estimate that there are approximately 70 futures over seven asset classes which trade on average once a minute and can be deemed "liquid." The data storage requirements from these are large but manageable, with the unprocessed, compressed data for the 70 futures at 12 GB per day and the processed uncompressed data at 36 GB per day. A cost-efficient way of managing such data is a cloud-based solution, which includes backups: for example, the Amazon Simple Storage Service (S3) file system.[7]

Learning

Once data processing is done, learning is then carried out in batch mode using the last H days worth of data to generate the following day's parameters. For each futures contract, the parameter $\boldsymbol{\Theta}$ is learned. To run the learning phase on these 70 futures within a 24-hour window, we estimate needing a server farm consisting of 25 nodes, each node being a quad core machine with 16GB RAM. Again, a cloud solution allows a high degree of flexibility with low costs: for example, the Amazon Elastic Compute Cloud (EC2).

Batch learning requires the use of a parallelized distributed framework, such as Condor. Condor is open-source software which optimally allocates CPU from the cluster of worker nodes to process jobs as and when the nodes become free (Thain et al. [2005]). While, in theory, learning could be done with MATLAB, MATLAB would need to be licensed on every worker node in the cluster, and so for this reason Java executables are suggested.

Inference

The online inference code is run by hardware sitting in the rack space at the colo and is latency sensitive. Even during peak market events (approximately 15,000 orders/second), GLOBEX market data is disseminated *externally* within 15 milliseconds of being generated. This market data then takes 5 microseconds from the GLOBEX server to a colo rack, while a subsequent order then takes 5 microseconds from a colo rack to the FALCON match engine. The inference software must be quick enough to operate in this time space if exploitation of detected icebergs is to be achieved.

A Java-based architecture running on Unix blade servers (one per asset class) is suggested to carry out the online LOB rebuilds and subsequent inference. One possible development solution would be to use *content-addressable memory* (CAM). CAM is a special type of computer memory used in certain very high-speed searching applications. JavaSpaces are a Java implementation of this memory paradigm for parallel computing. JavaSpace is shared memory, which, in addition to simple object caching, can use a "master-worker" software pattern, where the master hands out jobs to generic workers Setzkorn and Paton [2004]. This pattern has analogies with the distributed Condor framework used in the learning phase; however, rather than being across different machines as in the case of Condor, we are suggesting implementing JavaSpaces on a single server with multiple cores. This framework negates the need for complex scheduling algorithms, while at the same time achieving low latency. The hidden order problem is particularly well suited to such parallelization, as the individual LOBs are independent and can each be treated as a Java object.

Summary of Implementation

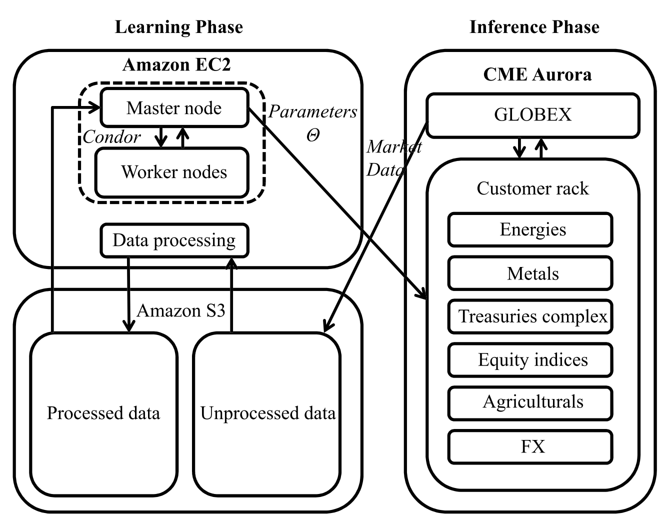
The barriers to entry for this type of algorithmic trading are high: specifically, costs associated with rack space, server farms, hardware, and the specialized development expertise required to implement the research. A schematic of the implementation is shown in Exhibit 14. The schematic shows one physical location, the CME colo data center in Aurora, IL, where the GLOBEX matching engines are based and the inference phase of our algorithm occurs and two cloud locations, where data storage, processing and subsequent learning occurs. The server farm is shown as being Amazon EC2 and the persistent storage as Amazon S3, both popular solutions with financial developers. Learning at the server farm takes place on a daily basis and then pretrading the latest set of parameters are uploaded to the colo for the inference algorithm to use during that trading day.

SIMULATION AND RESULTS

As the values of the latent variables V and Ψ are never known, the *true* performance of the algorithm cannot be directly tested. There are two exceptions to this, however:

1. Using a generative model to create synthetic data where some of the orders are iceberg orders.
2. Testing in a "live" setting by submitting iceberg orders using a small amount of capital, and then immediately closing the positions.

EXHIBIT 14
Computational Implementation of the Hidden Liquidity Detection Algorithm



Due to space and monetary constraints, neither of these approaches is implemented. The algorithm is evaluated on real data in the following section.

Real Data

The results from running our algorithm on the ES future are now presented. While the ES future does have strategy contracts, these are limited to calender spreads, and quoting and trading activity in them is extremely limited, to the extent that they can be ignored. Nearly 100% of activity is in the front month outright contract, so only these orders are considered. The mean number of limit orders per day over the period from January 1, 2011, to December 31, 2011, was 4.4 million. The mean number of iceberg orders per day, $\bar{K}$, was seen to be 0.14 million, meaning that 3.2% of all limit orders are iceberg orders.

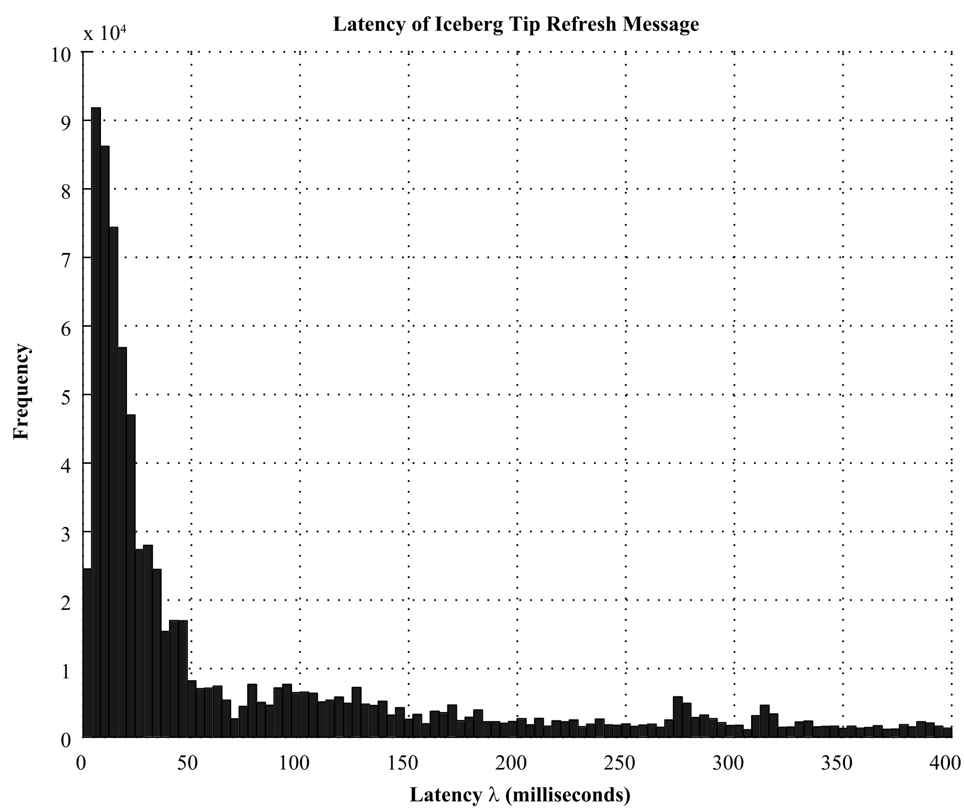
From the iceberg orders detected by Algorithm 6, the latency of inserting the iceberg refresh messages can be inspected. The insertion time delay is due to computational latency. The latency is not constant due to the varying load in the trading engine at any point in time. By setting $dt = \bar{\lambda} + 3\sigma_\lambda$ in the algorithm, unnecessary searching for refresh messages can be minimized. The distribution of this latency is shown in Exhibit 15. The data shows the refresh messages are broadcast according to an exponential distribution with a mean μ_λ of 76 milliseconds and standard deviation σ_λ of 98 milliseconds. This allows us to set dt = 369 in Algorithm 6 so that orders are searched up to 369 milliseconds after trade messages.

The joint distribution of total order volumes and peak sizes is shown in Exhibit 12. Related to this distribution is the number of tranches of iceberg orders $N = \hat{V}/\Psi$, as shown in Exhibit 16. It is noteworthy that

EXHIBIT 15

The Distribution of the Latency of Iceberg Refresh Messages λ

When a trade fills the tranche of an iceberg order, a new tranche is inserted into the LOB by GLOBEX.



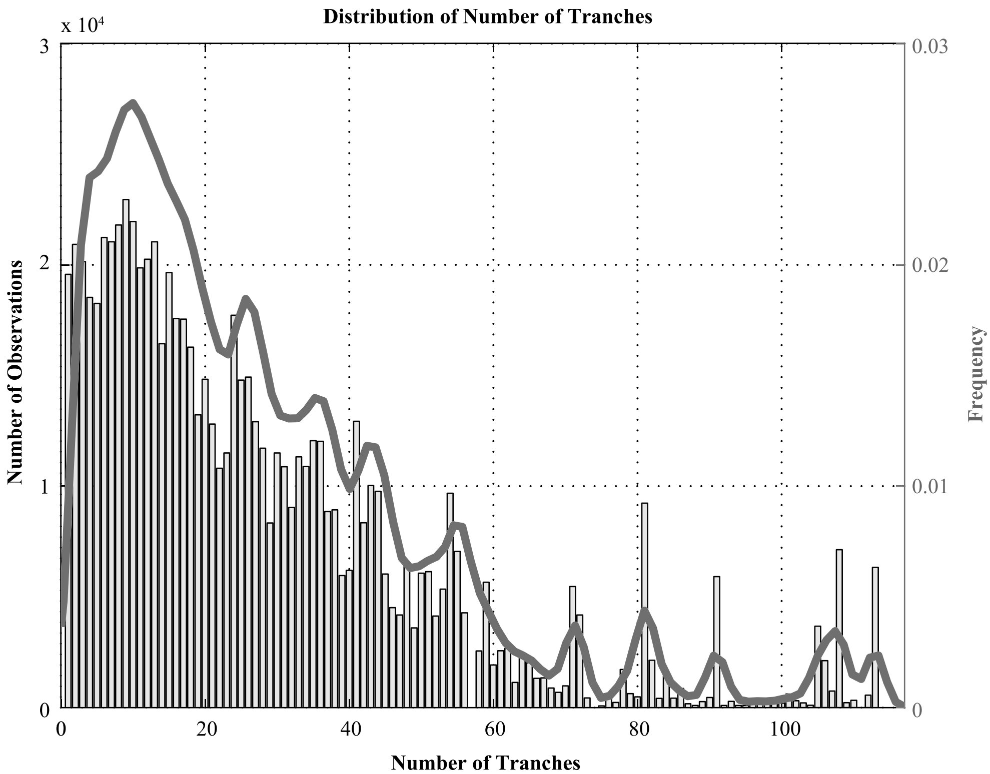
Alg.6 Iceberg Detection Forward Algorithm. $[\mathbf{\Gamma}, \mathbf{\Theta}] = \text{IDFA}(data, dV, dt, \mathbf{\Theta})$

```
for t = 1 : T do
  if trade then
    [tPrice, tSide, tSize] = trade_t {Extract trade price, side and size}
    for i = t : t + dt do
      if order then
        [P, S̃, m] = order_i {Extract order price, side and price level}
        if (S̃ == ask) then
          pCond = P ≤ tPrice {The limit price on the ask. The lowest price at which a seller is willing to
          sell.}
        else if (S̃ == bid) then
          pCond = tPrice ≤ P
        end if
        if ((m == 1) & (pCond) & (S̃ == tSide) & (dV_t > 0) & (isImplied ≠ true)) then
          ψ = dV_t
          if Φ_k(largeTrade) then
            if ψ > Φ_k(Ψ) then
              Φ_k(Ψ) = ψ
              Φ_k(largeTrade) = 0
              sizeCond = 1
            else
              if tSize > largeOrder then
                sizeCond = 1
              else
                sizeCond = 0
              end if
            end if
          else
            if tSize == 1 then
              expectedRefresh = Φ_k(Ψ)
              sizeCond = (expectedRefresh == ψ)
            else
              if tSize ≥ ψ then
                expectedRefresh = rem(tSize, Φ_k(Ψ))
              else if tSize < ψ then
                expectedRefresh = Φ_k(Ψ) − tSize
              end if
              sizeCond = (expectedRefresh ≤ ψ)
            end if
          end if
          if sizeCond then
            Φ_k = testIfActive(Φ_k)
            if ((exists(Φ_k)) & (Φ_k(α) == 1)) then
              {Φ_k exists and is active}
              λ = i − t
              θ = i − φ_{k,n}(t)
              φ_{k,n+1} = {ψ, t, λ, θ} {Add a new tranche to an existing iceberg order}
            else if ((exists(Φ_k)) & (Φ_k(α) =!1)) then
              {Φ_k might have existed and is no longer active.}
              {How many tranches of Φ_k have been seen so far?}
              if (n == 1) then
                {Φ_k turned out not to be an iceberg order. Delete it.}
                Φ_k = [ ]
              else if (n > 1) then
                {Φ_k existed and is no longer active.}
                {Search for cancelled orders at t > φ_{n,k}(t) which match Φ_k parameters.}
                if (Cancelled) then
                  Φ_k(ω) = 0
                else
                  {Φ_k was filled}
                  Φ_k(ω) = 1
                end if
              end if
            else
              {This is a new iceberg order}
              Ψ = ψ
              Φ_k = {S̃, P, Ψ, α}
              λ = i − t
              φ_{k+1,n} = {ψ, t, λ}
            end if
            if (Inference phase) then
              Γ = online_inference(Φ, Θ)          {Use Θ from the learning phase. Return Γ.}
            end if
          end if
        end if
      end if
    end for
  end if
  if rem(tradeSize, Φ_k(Ψ)) == 0 then
    {An existing iceberg active order has a peak size equal to integer multiples of the the max show.}
    φ_{k,n+1} = {ψ, t, λ, θ} {Add a new tranche to an existing iceberg order. Set Φ_k(largeTrade) = 1}
  end if
end for
if (Learning phase) then
  {Run at the end of each trading day. offline_learning loads the previous H trading days worth of Φ and
  concatenates with the current version.}
  Θ = offline_learning(Φ)          {Return Θ.}
end if
```

EXHIBIT 16

Distribution of the Number of Tranches of an Iceberg Order

The kernel density plot shows visible spikes around round numbers, suggesting human bias.



local maxima can be seen in the distribution at multiples of 5 and 10, suggesting a human bias in selecting these numbers. In particular, the following values of N are popular: 10, 25, 40, 55, 70, 80, 90, and 110. This is in agreement with earlier studies where the icebergs are known from private data, which show that values of $N = 5$ and $N = 10$ account for 17% and 37%, respectively, of all the icebergs in the sample (Esser and Mönch [2007]). The mean value of the distribution is 29, with the minimum and maximum values of 2 and 113 respectively. Our mean is notably larger than the the findings of Frey, who finds that on average an iceberg consists of seven tranches, of which five are executed and the remaining two canceled (Frey and Sandas [2009]). By definition the algorithm does not consider an order with only one tranche to be an iceberg order, introducing a known bias to the results.

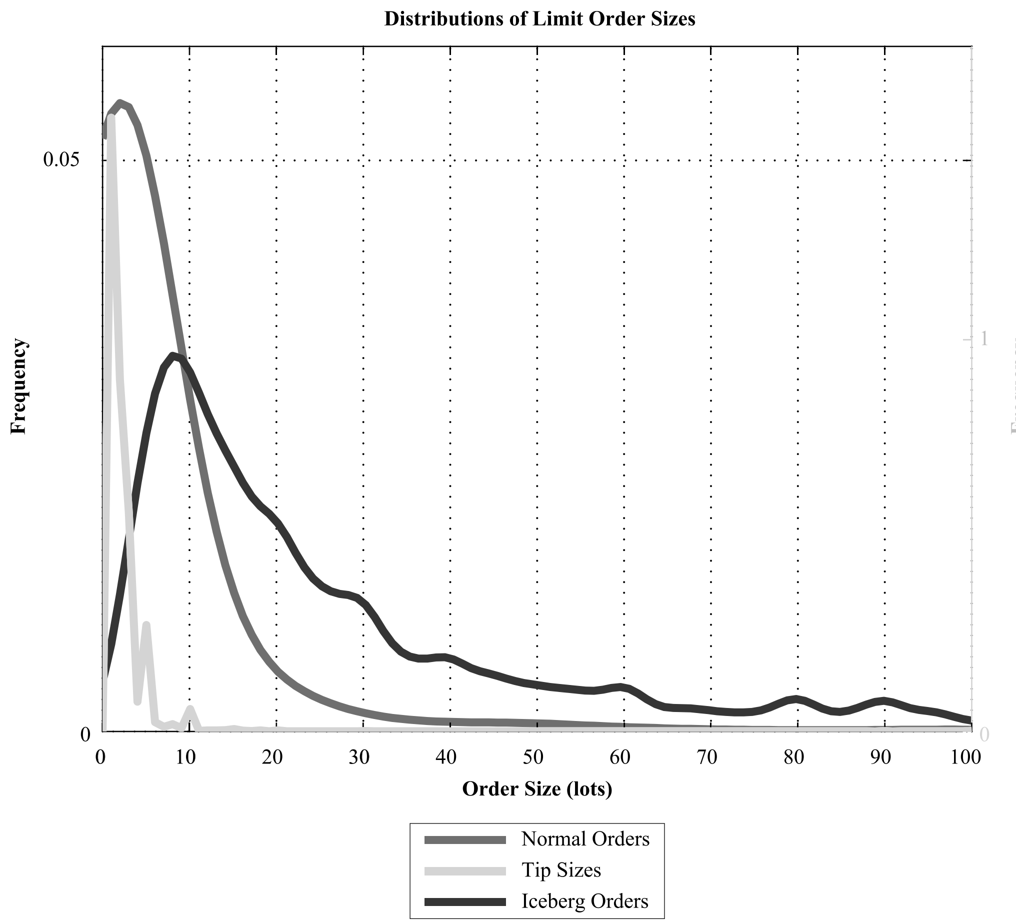
Distributions of total order size for normal limit orders, total order size for iceberg orders, and peak size for iceberg orders are shown in Exhibit 17. It can be seen that the mean iceberg order is 12 times bigger than the mean limit order (5 lots versus 62 lots), while the peak size is approximately equal to the mean limit order size. This is in close agreement with the findings of Frey, who finds that iceberg orders are 16 times bigger than average limit orders and that the peak size of these limit orders is 2.5 times bigger than the average limit order. This statistically significant increase in size suggests that the requirement to use iceberg orders comes from wishing to minimize market impact. The large size of iceberg orders is also good reason for liquidity providers to attempt to detect to detect this hidden volume.

The cumulative traded volume is shown against the cumulative traded volume from iceberg orders for a single trading day in Exhibit 18. It can be seen that iceberg orders are filled as per the trading profile of the regular market. When the whole data set is used, iceberg orders account for 12% of the total volume executed.

The number of orders present at each price is broadcast by GLOBEX. In Exhibit 19, this figure is

Exhibit 17
Distribution of Order Sizes

Kernel density estimates are shown for three classes of order–non-iceberg orders, iceberg orders and iceberg peak sizes.



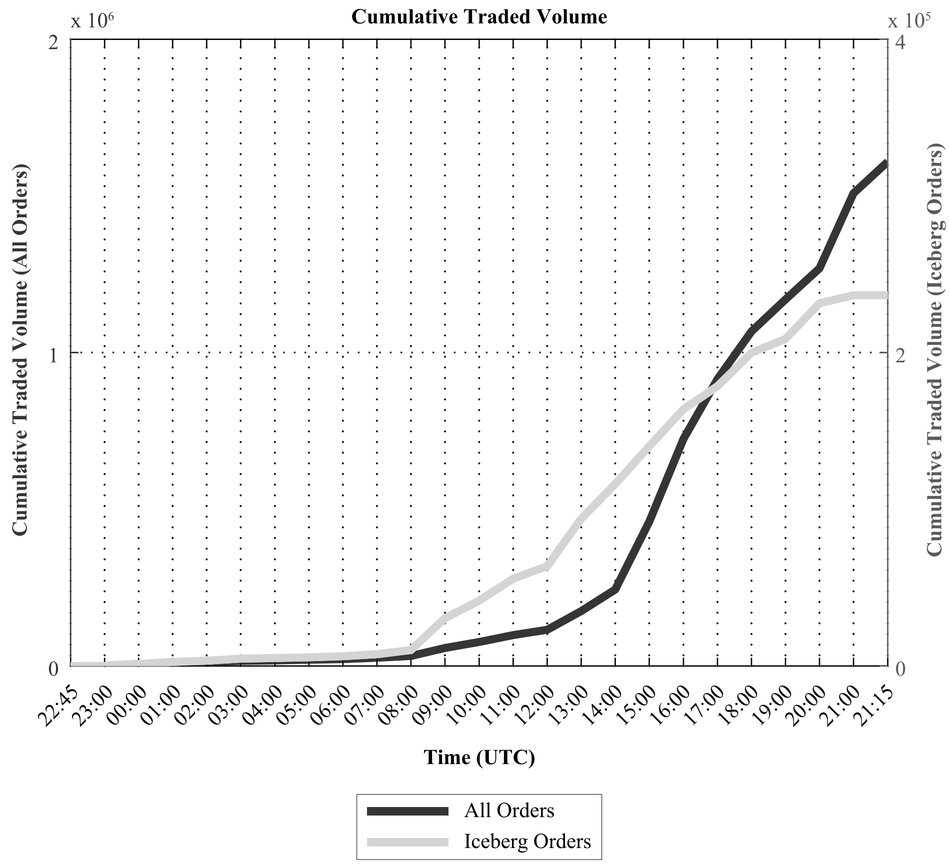
compared against the number of iceberg orders at each price level for a single trading day. Data is bucketed into one-hour samples and the number of orders averaged. As iceberg orders in depth levels can only ever be detected if the price moves away from the inside price level, it is likely that the number of iceberg orders is underestimated for the depth price levels. The results show that on this trading day, iceberg orders comprise of 1.7% of limit orders by number and 5.6% of limit orders by volume, with nearly 100% of these occurring at the best price level. Both figures are below the data set averages of 3.2% and 9% respectively.

The predictive ability of Algorithm 5 is inspected by tracking the error term in the prediction of the total iceberg volume $\hat{V}$ over the course of an iceberg order being filled. Three experiments are carried out, each of which looks at $\hat{V}$ being tracked for each iceberg on the trading day of January 13-14, 2011, for ESH1, using $\boldsymbol{\Theta}$ generated from the previous trading day. Each experiment uses a different way of generating $\hat{V}_{k,n}$, allowing the performance of our approach to be benchmarked.

1. The iceberg total volume $\hat{V}_{k,n}$ is generated by the online inference Algorithm 5.
2. The iceberg total volume is randomly selected from a uniform distribution. Constraints are $\hat{V}_{k,n}$ larger than the current volume and smaller than the largest iceberg seen.
3. The iceberg total volume is equal to the current volume. The iceberg is finished or canceled.

EXHIBIT 18
Cumulative Traded Volume from Displayed and Hidden Orders

The intraday profile of hidden order fills appear visually different from that of displayed volume.



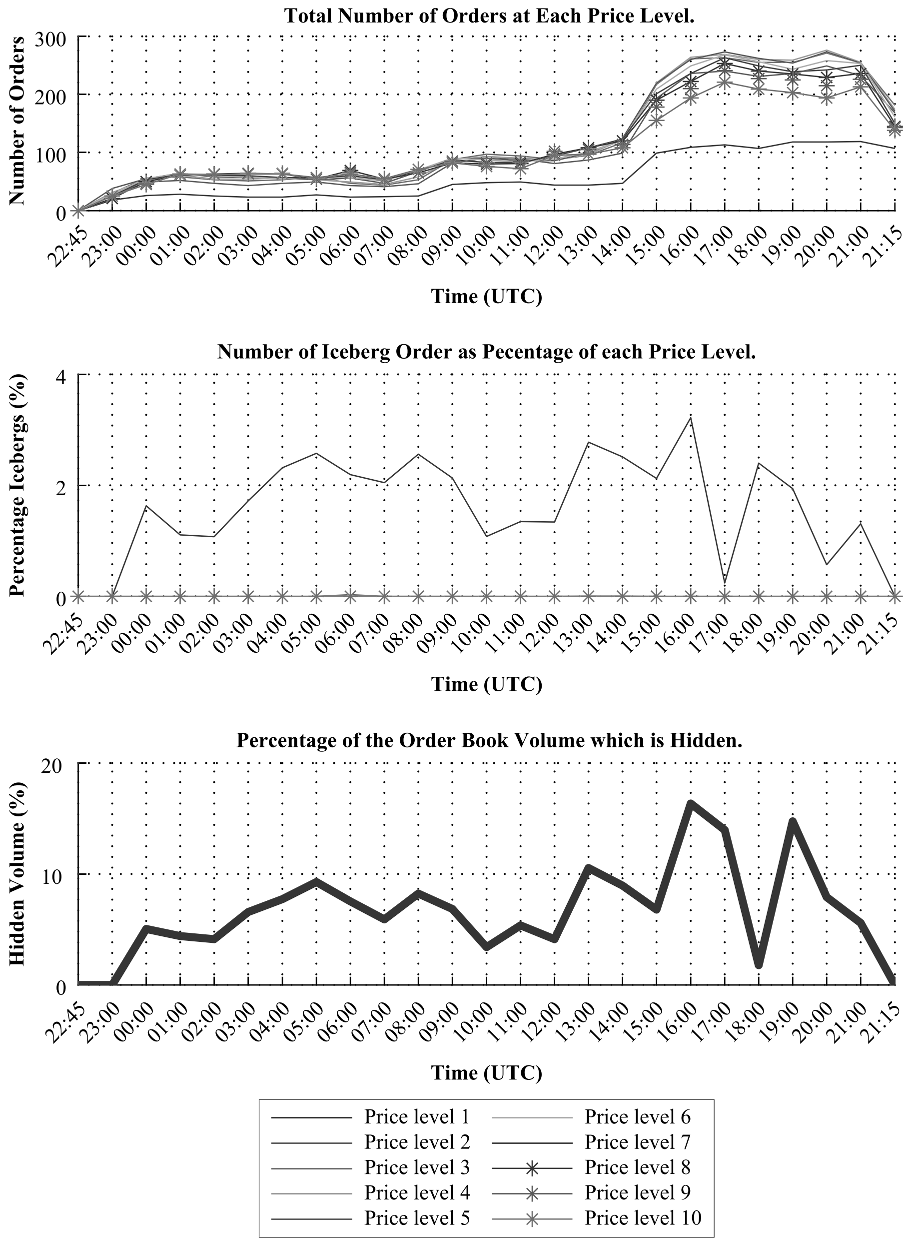
The results are expressed as root-mean-square deviation (RMSD) *prediction errors* by comparing $\hat{V}_{k,n}$ to the realized iceberg order volume $V_{\mathrm{k,R}}$ using $\mathrm{RMSD}_k = \frac{\sqrt{\sum_n^N (V_{\mathrm{k,R}} - \hat{V}_{k,n})}}{N}$. For each experiment, the progression of each iceberg order is normalized by linear interpolation, so that each iceberg order has an equal number of tranches, $N = 10$. Bootstrapping is a resampling methodology that allows a sampling distribution to be estimated from limited data. Bootstrapping enables the differences between the performance of the three experiments to be inspected for statistical significance. Using bootstrapping, for each experiment, the mean μ of the population of the means and standard deviation σ of the means are calculated. The results are shown in Exhibit 20 and Exhibit 21. Exhibit 20 shows Experiment I is seen to have the lowest μ and σ, suggesting that Algorithm 5 gives the best predictions. Experiment II is the worst performing, as expected, with both experiment I and III showing several factors of improvement. While the performance of experiment I and III is similar, the difference can be seen to statistically significant at the 95% confidence level. Exhibit 21 shows the sampling distribution of the means of each experiment, along with the mean of means and the 2.5% tail confidence intervals. For the case of experiments I and III which gave similar results, the distribution of means can be seen to be statistically significant as the mass of the distributions do not overlap.

Using ESH1 data for 13-14 January 2011 Exhibit 22 inspects the degree of sequential updates. A flat curve means that the first prediction was accurate, while a steep or non-linear curve means that the estimated volume

EXHIBIT 19

Number of Iceberg Orders at Price Levels

Price levels 1-10 are shown for the ESH1 future. Bid and ask are combined. January 13–14, 2011.



changes dramatically throughout the trading day. The information from this gradient $d\hat{V}/dn$ is captured by the RMSD. In subplot one, all the trajectories are shown, with some have a flat gradient and a few being non-linear and having a steep gradient. In subplot two, the mean curve is shown and this can be seen to be reasonably flat, suggesting that once an iceberg has been detected the predicted volume is accurate.

Exhibit 20
Quality of Iceberg Predictions

Experiment	μRMSD	σ_μ
I	23.57	0.017
II	1,502.34	0.144
III	24.61	0.018

Applications for the Investment Community

Hidden liquidity detection and prediction has a range of applications, from smart order routing (Almgren and Harts [2008]) to front-running (Harris [1996]). In this section, we present specific examples of how our algorithm can be used for liquidity providers and for liquidity takers.

Market makers can use a weighted bid-ask ratio as a proxy for supply and demand (see, for example, Kim et al. [2007]). An example of such a proxy is the distance of a quote from the mid, weighted by its order size, $\boldsymbol{r} = \frac{|mid-bid|\times bidSize}{|mid-ask|\times askSize}$. The presence of hidden volume can cause this proxy to be wrongly estimated. Exhibit 23 shows the ratio of this proxy using hidden volume, by Algorithm 5, to the displayed volume. ESH1 data for 13–14 January 2011 was used. Only the best price levels of the LOB were used in the calculation. The ratio of the metrics was calculated as a percentage error $\boldsymbol{r} = \frac{|\boldsymbol{r}_I - \boldsymbol{r}_E|}{\boldsymbol{r}_I}$ where $\boldsymbol{r}_I$ is the ratio including the iceberg volume and

Exhibit 21
Bootstrap Sampling of RMSDs from Iceberg Predictions

For each of the three experiments, this graphic shows the sampling distribution of the population of the means.

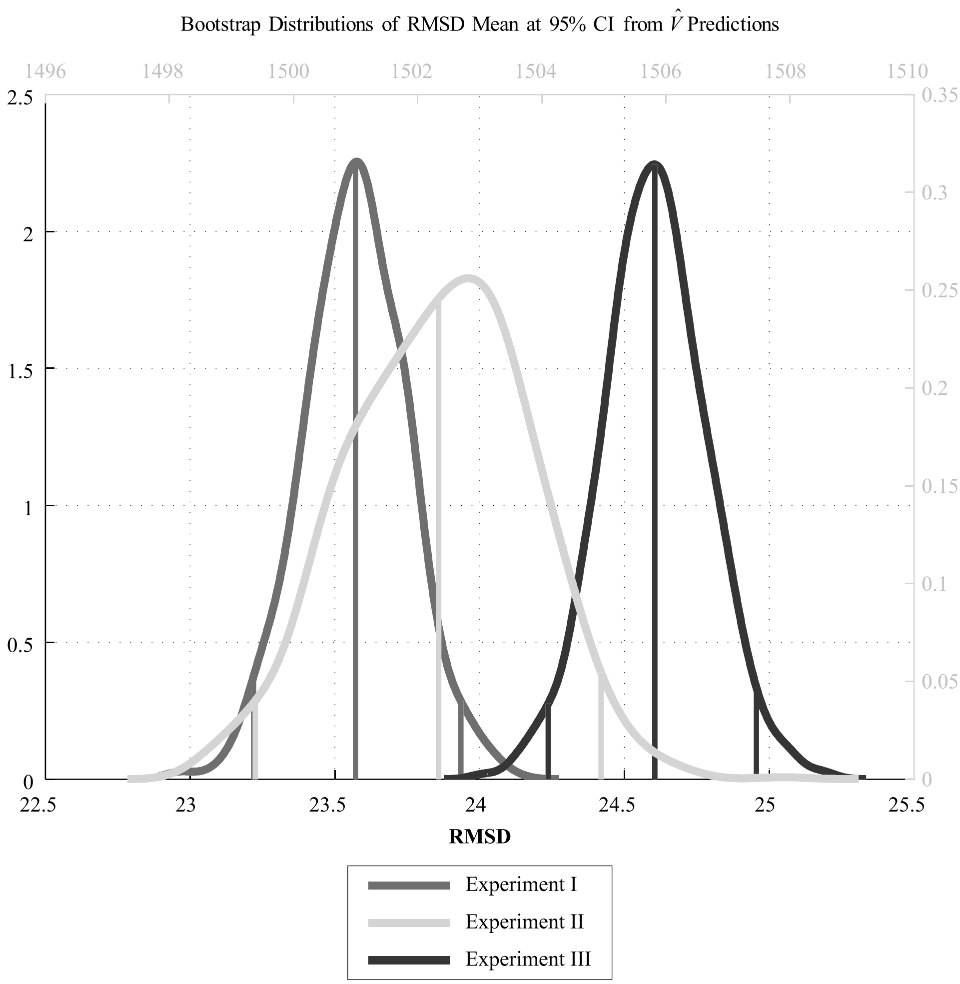


EXHIBIT 22
Quality of Iceberg Predictions from Experiment I

For all the iceberg orders on one trading day, the value of $\hat{V}$ was tracked and normalized to $N = 10$.

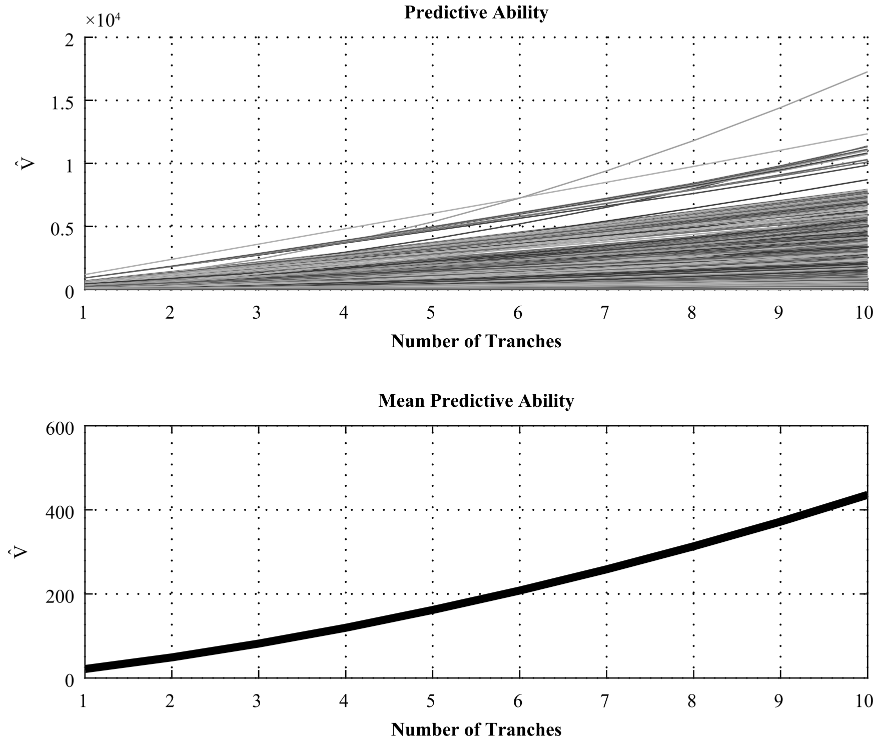
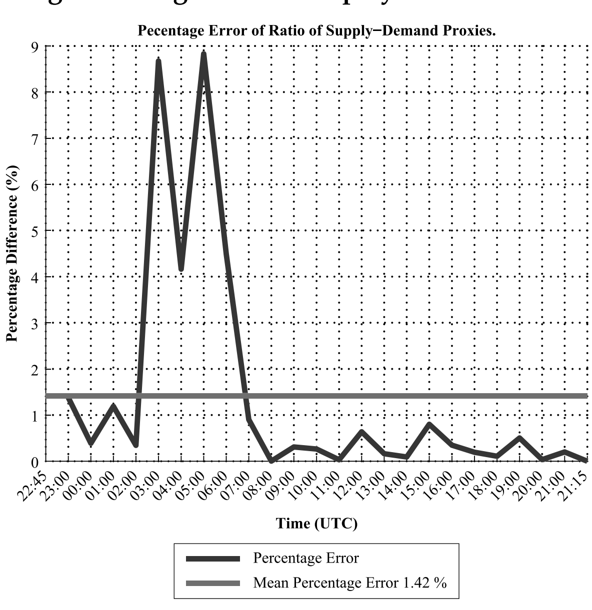


EXHIBIT 23
Time-Varying Percentage Difference for a Weighted Bid-Ask Metric, Using the LOB Volume Predicted by the Algorithm Against the Displayed LOB Volume



$\boldsymbol{r}_E$ is the ratio excluding the iceberg volume. The results show that an error greater than 1% exists when hidden volume is considered, which could be enough to affect liquidity provision based on $\boldsymbol{r}$.

In a second liquidity supplier example, an order could be placed on the opposite side of the LOB to the detected iceberg in size equal to $\hat{V}$, to try and gain the bid-ask spread. The time until $\hat{V}$ is filled could be estimated using an order arrival rate model (see, for example, Easley et al. [2008], Wolff [1982]). The higher the arrival rate, the shorter the waiting time. The practical benefit of such a prediction would be to let the user know how long he had left to "act" on the iceberg information. In a time-priority market such as ES, joining the bottom of the queue on the other side of the LOB from Φ_k could mean that the time delay causes the iceberg to be missed and hence the liquidity supplier would be "penalized" via the exchanges quote-to-trade ratio. Such volume "targeting" strategies would increase market makers' executed volume in proportion to the volume in the LOB.

A liquidity taker uses iceberg orders to minimize market impact. When an iceberg order submitted by a liquidity taker is detected by the market, the market will stop filling it and move the price away, causing the liquidity taker to incur slippage, behavior that can be explained by the market realizing that the order imbalance has changed (Esser and Mönch [2007]). It is suggested that market impact resulting from iceberg orders being detected on GLOBEX can be minimized by canceling an iceberg after the second tranche of the iceberg order has been filled. This strategy would make the prediction algorithm presented in this article redundant, because the first tranche of an iceberg is used for the initial detection, and only after the second tranche is the order positively confirmed as an iceberg, as per Exhibit 11. By applying this strategy to a series of iceberg orders, the disadvantages associated with detection could be avoided.

CONCLUSIONS AND FURTHER WORK

Conclusions

Iceberg orders on GLOBEX are detectable and predictable. The presented approach can be used by liquidity suppliers to exploit iceberg orders, while liquidity takers can use it to submit iceberg orders in a way that inferring the true hidden volume is most difficult. The average iceberg order for the ES has total size 62 with a peak size of 5, meaning that its order size is 12 times larger than the average order size. In our ES data sample, iceberg order peaks account for 3% of all limit orders submitted by number, and iceberg orders match against 12% of all traded volume. Iceberg orders constitute 9% by volume and 1.5% by number of all resting orders in the LOB.

The gaming behavior presented in this article could be countered by GLOBEX supporting an optional FIX tag to allow randomization of certain parts of the iceberg order, such as $\mathbf{\Psi}$, while constraining $V = \sum_n^N \psi_n$. Not to support such functionality may be unfair to less sophisticated investors, who may not realize the potential consequences of using iceberg orders. Having markets which are fair and transparent is central to the integrity of the global financial system.

Further Work

By applying the algorithm across the seven major asset sectors (equity index, STIRs, bonds, FX, agricultural, energies, metals) traded on GLOBEX, iceberg orders may be seen to play a more important role in some sectors than others it is speculated that this might be related to the matching algorithm used in the sector, specifically if that algorithm has a time component.

An additional modification to the learning step presented in this article could be to carry out estimation of $\mathbf{\Theta}$ conditional on various factors which could affect either V or $\mathbf{\Psi}$: for example, the distance between the limit price and the mid-price, the size of the bid-ask spread, or market volatility.

The empirical model presented in this article can be reformulated as a probabilistic Bayesian model, and we plan to publish on such a model in the future. By defining the state variables as the peak size and the total size, the process can be represented as a Markov chain with some deterministic transition probabilities.

A variety of patterns exists in the LOB. Some of these patterns are generated by system effects, such as order types and latencies, and some patterns are generated by repetitive human behavior, such as order sizes and support levels. An exciting area of future work will be further automating pattern recognition in the LOB using expectation maximization algorithms, such as Baum-Welch, to learn latent structure (Baum et al. [1970]).

ENDNOTES

We would like to thank www.cmegroup.com for allowing us to use their data in this research. We would also like to thank www.onixs.biz for allowing us to use their C# FAST decoder.

[1]www. futuresindustry.org/ptg/membership.asp

[2]www.rsj.com/en/algorithmic-trading/current-volumes

[3]Only interest rate options do not support icebergs on GLOBEX.

[4]FIX tag 210

[5]When FIX tag 277=1, trade volume is also not removed from the book.

[6]www.cmegroup.com/colo

[7]http://aws.amazon.com

REFERENCES

Aitchison, J., and C. Aitken. "Multivariate Binary Discrimination by the Kernel Method." *Biometrika*, Vol. 63, No. 3 (1976), pp. 413-420.

Almgren, R., and B. Harts. "A Dynamic Algorithm for Smart Order Routing." White paper, StreamBase, 2008.

Avellaneda, M., J. Reed, and S. Stoikov. "Forecasting Prices from Level-I Quotes in the Presence of Hidden Liquidity." *Algorithmic Finance*, Vol. 1, No. 1 (2011), pp. 35-43.

Bank for International Settlement. "*Triennial Central Bank Survey of Foreign Exchange and Derivatives Market Activity in 2010*." 2010.

Baum, L., T. Petrie, G. Soules, and N. Weiss. "A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains." *The Annals of Mathematical Statistics*, Vol. 41, No. 1 (1970), pp. 164-171.

Bessembinder, H., M. Panayides, and K. Venkataraman. "Hidden Liquidity: An Analysis of Order Exposure Strategies in Electronic Stock Markets." *Journal of Financial Economics*, Vol. 94, No. 3 (2009), pp. 361-383.

Blank, J. "Implied Trading in Energy Futures." *The Journal of Trading*, Vol. 2, No. 3 (2007), pp. 45-48.

Borkovec, M. "Algorithmic Trading System and Method." U.S. Patent 8,140,416, 2012.

Burghardt, G., J. Hanweck, and L. Lei. "Measuring Market Impact and Liquidity." *The Journal of Trading*, Vol. 1, No. 4 (2006), pp. 70-84.

Cave, T. "Europe's Top 10 High-Frequency Kingmakers (in Scandinavia, at Least)." Dow Jones, October 2011.

Christensen, H., R. Turner, S. Hill, and S. Godsill. "Rebuilding the Limit Order Book: Sequential Bayesian Inference on Hidden States." *Quantitative Finance* (2013), submitted.

Chicago Mercantile Exchange. First Quarter Liquidity Monitor. www.cmegroup.com/education/files/Liquidity-Monitor-Q1-2011.pdf, 2011.

——. CME Data Mine. Market Data Platform FIX/FAST. Core Functionality. Version 3.2, 2012.

——. GCC Product Reference Sheet. www.cmegroup.com/confluence/display/EPICSANDBOX/GCC+Product+Reference+Sheet, 2013a.

——. Product Specific Information. www.cmegroup.com/confluence/display/EPICSANDBOX/Product+Specific+Information, 2013b.

Cont, R., S. Stoikov, and R. Talreja. "A Stochastic Model for Order Book Dynamics." *Operations Research*, Vol. 58, No. 3 (2010), pp. 549-563.

De Winne, R., and C. D'hondt. "Hide-and-Seek in the Market: Placing and Detecting Hidden Orders." *Review of Finance*, Vol. 11, No. 4 (2007), pp. 663-692.

Durbin, M. *All About High Frequency Trading*, 1st ed. McGraw-Hill Professional, 2010.

Easley, D., R. Engle, M. O'Hara, and L. Wu. "Time-Varying Arrival Rates of Informed and Uninformed Trades." *Journal of Financial Econometrics*, Vol. 6, No. 2 (2008), pp. 171-207.

Esser, A., and B. Mönch. "The Navigation of an Iceberg: The Optimal Use of Hidden Orders." *Finance Research Letters*, Vol. 4, No. 2 (2007), pp. 68-81.

Fleming, M., and B. Mizrach. "The Microstructure of a U.S. Treasury ECN: The Brokertec Platform." Federal Reserve Bank of New York, 2009.

Frey, S., and P. Sandas. "The Impact of Iceberg Orders in Limit Order Books." American Finance Association San Francisco Meetings Papers, 2009.

Haddad, R., and A. Akansu. "A Class of Fast Gaussian Binomial Filters for Speech and Image Processing." *Signal Processing, IEEE Transactions*, Vol. 39, No. 3 (1991), pp. 723-727.

Harris, L."Does a Large Minimum Price Variation Encourage Order Exposure?" New York Stock Exchange, 1996.

Hasbrouck, J., and G. Saar. "Limit Orders and Volatility in a Hybrid Market: The Island ECN." New York University, Working Paper, No. FIN-01-025, 2001.

Hastie, T., R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. Springer, 2001.

Hautsch, N., and R. Huang. "A Statistical Model for Detecting Hidden Liquidity." Quantitative Products Laboratory, Berlin, 2009.

Hunsader, S. "Analysis of the Flash Crash." NANEX, 2010.

Janecek, K., and M. Kabrhel. "Matching Algorithms of International Exchanges." 2007.

Kim, A., J.D. Farmer, and A. Lo. "Market Making by Learning Liquidity Imbalance." MIT, 2007.

Madhavan, A. "Market Microstructure: A Survey." *Journal of Financial Markets*, Vol. 3, No. 3 (2000), pp. 205-258.

Moinas, S. "Hidden Limit Orders and Liquidity in Limit Order Markets." Toulouse Business School, 2006.

Moro, E., J. Vicente, L. Moyano, A. Gerig, J. Farmer, G. Vaglica, F. Lillo, and R. Mantegna. "Market Impact and Trading Profile of Hidden Orders in Stock Markets." *Physical Review E*, Vol. 80, No 6 (2009), pp. 0661021-0661028.

Overdahl, J. "Implied Matching Functionality in Futures Markets." *Futures Industry* (November 2011), pp. 37-41.

Rajagopalan, B., and U. Lall. "A Kernel Estimator for Discrete Distributions." *Journal of Nonparametric Statistics*, Vol. 4, No. 4 (1995), pp. 409-426.

Scott, D. *Multivariate Density Estimation*. Wiley, 1992.

Setzkorn, C., and R. Paton. "Javaspaces: An Affordable Technology for the Simple Implementation of Reusable Parallel Evolutionary Algorithms." In *Knowledge Exploration in Life Science Informatics*, edited by W. Dubitzky, Springer, 2004, pp. 151-160.

Thain, D., T. Tannenbaum, and M. Livny. "Distributed Computing in Practice: The Condor Experience." *Concurrency and Computation: Practice and Experience*, 17 (2005), pp. 323-356.

Tuttle, L. "Hidden Orders, Trading Costs and Information." 2005.

Vaglica, G., F. Lillo, E. Moro, and R. Mantegna. "Scaling Laws of Strategic Behavior and Size Heterogeneity in Agent Dynamics." *Physical Review E*, Vol. 77, No. 3 (2008).

Wolff, R. "Poisson Arrivals See Time Averages." *Operations Research*, Vol. 30, No. 2 (1982), pp. 223-231.

Yao, C. "Hidden Agendas: A Study of the Impact of Concealed Orders." University of Illinois, 2012.

Erratum: Prediction of Hidden Liquidity in the Limit Order Book of GLOBEX Futures

Hugh L. Christensen[a,*], Robert Woodmansey[b]

[a]*Signal Processing and Communications Laboratory, Engineering Department, Cambridge University, CB2 1PZ, UK*
[b]*Onix Solutions, Alpha House, 100 Borough High Street, London, SE1 1LB, UK*

In our recent article we describe an iceberg prediction algorithm for limit order books (LOB) on CME GLOBEX (Christensen and Woodmansey, 2013). In the article successive peaks of an iceberg orders are described as retaining time priority. This is incorrect. Successive peaks of an iceberg order do *not* retain time priority. Each time a peak is filled, the refresh peak is inserted into the LOB at the back of the queue. CME Group have updated their public documentation to reflect this (CME, 2013).

These mechanics are now illustrated with an example, as shown in a revised Exhibit 7. An iceberg order is specified with a total size $V = 100$ and a max show $\Psi = 9$. The first 11 tranches of the iceberg are of size $\psi = 9$ and the final tranche is size $\bar{\psi} = 1$, so $N = 12$ (where N is the number of tranches). At time $t = 2$ the first peak of the iceberg can be seen in the LOB, behind a normal limit order for $S = 10$ (S is size). At this point in time, the hidden volume in the LOB is $V_H = 91$. At time step $t = 6$, a trade message for $S = 8$ is seen in the data feed. This results in two further messages being sent by GLOBEX, firstly, a LOB update message $dV = -8$, and secondly a peak replenish message $dV = 9$. The first of these messages is the trade volume being removed from the LOB. The second of these messages is the next tranche of the iceberg order being placed into the LOB *behind* the normal limit order of $S = 3$ (i.e. the normal limit order has higher time priority than the peak replenish order). It this peak replenish message time dt after a trade which allows the iceberg order to be detected.

As time-priority is lost between each peak, in essence an iceberg order is equivalent to a series of sequential limit orders entered by the trader. In reality, given the time it takes for a fill to be relayed back to the customer, iceberg orders are significantly quicker to be inserted into the LOB and also more convenient.

How does this change in system mechanics affect the prediction algorithm? The loss of time priority between tranches does *not* significantly affect the prediction algorithm. In the algorithm, orders which enter the LOB within dt seconds of a trade are considered viable candidates for peak refresh orders. We had initially reported that these peak refresh orders went to the front of the queue (having retained time priority). It is now known they go to the back of the queue having lost time priority. The only change to the algorithm requires a modification to the way in which the cumulative volume is tracked (page 79). Originally $\alpha = 0$ is set when the tracked cumulative volume exceeds the max show, plus dt to allow for system latency effects. Setting $\alpha = 0$ means that the iceberg order is no longer active (i.e. it has either been filled or cancelled). Now, in light of the loss of time priority between peaks, the α is set to zero when the tracked cumulative volume exceeds the max show plus the sum observable volume at the price level plus dt.

Does this change to the algorithm affect the presented results? Having re-run the algorithm using the updated methodology, the answer is no, hardly at all. The inside price level of ES contains significantly less volume than the depth price levels. For the year 2011, the volume of the inside price level (averaged over bid and ask) is approximately 750 lots. The differences between the old and new implementation of the algorithm were inspected. It was seen that the previous choice of dt, used to allow for system latency, was essentially acting as a buffer term. This meant significantly more volume was received than was present in the max show Ψ. Searching over this extended period allowed us to detect the refresh peak entering the LOB. In other words, by setting

*Corresponding author.
Email addresses: `hlc54@cam.ac.uk` (Hugh L. Christensen), `robert.woodmansey@onixs.biz` (Robert Woodmansey)

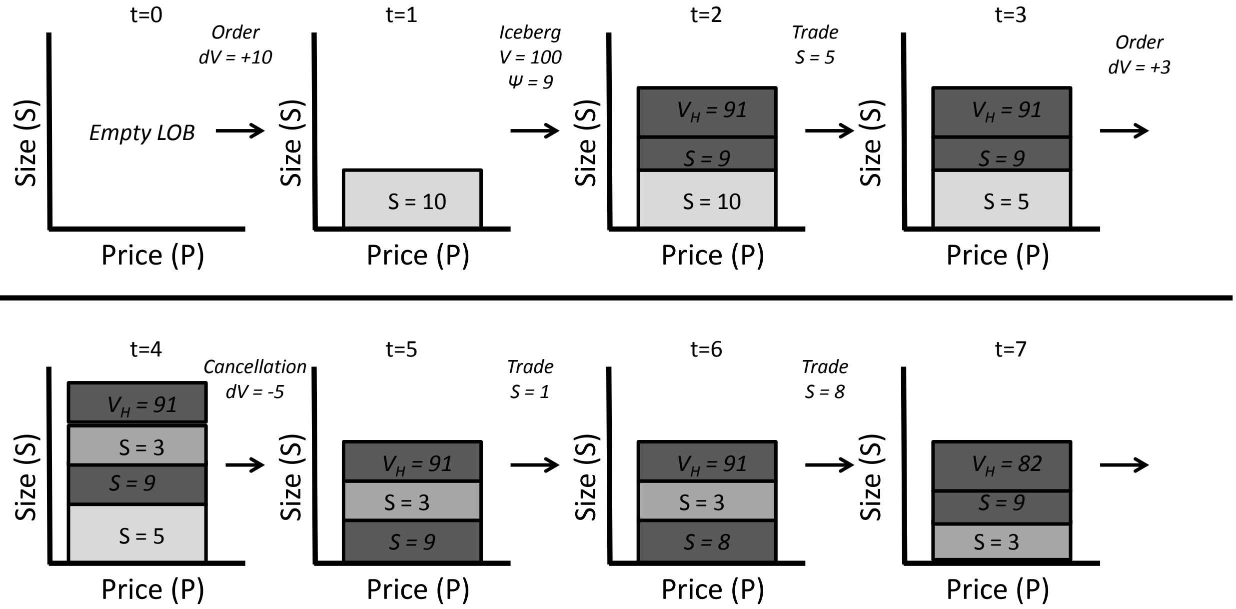


Exhibit 7: Schematic of iceberg order mechanics. A simplified LOB consisting of just one price level is shown progressing through time. At each step a FIX message is applied to the LOB. The bottom of stack has the highest time priority. Normal limit orders are shown in light greys, iceberg order in dark grey. V_H denotes hidden volume which can *not* be seen in the LOB.

dt to be large, it had the same effect as summing volume over the price level. This is in agreement with Exhibit 15, the distribution of the latency of iceberg refresh messages, which shows a maximum at 8 milliseconds and a tail extending out to 400 milliseconds. The maximum corresponds to either high number of lots per second being traded, or to when the price level contained little volume. The long tail corresponds to the opposite case, when it took some time to trade through the whole price level and the new peak replenish message appear.

References

Christensen, Hugh, R.Woodmansey. "Prediction of Hidden Liquidity in the Limit Order Book of GLOBEX Futures." *The Journal of Trading*, 8 (3) (1976), pp. 68–95.

Chicago Mecantile Exchange. "Order Qualifiers." `www.cmegroup.com/confluence/display/EPICSANDBOX/Order+Qualifiers`.