EXPERT OPINION

The importance of maintaining a time-series database of quotes, orders and trade executions

By Stuart Farr, President of Deltix

Stuart Farr

There has been significant growth in the use of algos for executing forex orders. There has also been a proliferation of execution venues (banks, ECNs, non-bank trading). As such, forex traders face a bewildering set of choices of where and how to execute. Given the dynamic nature of the forex market, any choices need to be constantly re-evaluated and changed objectively.

The essential component required for such on-going decision making is the ability to faithfully and dynamically record the full depth of the order book for each liquidity venue to which the trading entity is connected. With multiple liquidity providers (LPs), this is the proverbial "drinking from the firehose" which requires the ability for the trading system to ingest quote updates and trades at rates measured in hundreds of thousands per second. Time-stamping is critical. This should be at maximum intervals of one millisecond but preferably microsecond. Timestamps should include the time when the "message" was sent, time received and time processed. As data moves from one server to another via cross-connects and other methods it is important to record the timestamps of data arrival at each location. Clearly, the ability to have synchronized clocks is required.

A key aspect here is that this quote recording system is plugged into the trading firm's production trading infrastructure. As such, the latencies and infrastructure implicit in the trading firm's particular set-up are baked into the historical time-series thus recorded. Further, the liquidity specific to the trading firm can be used for subsequent analysis: a critical point given the customer-specific nature of much forex liquidity. The resultant intertwining of orders, executions, and order books from multiple LPs with microsecond granularity is a different order of precision than that typically provided by legacy EMS and OMS.

Once this recording system is part of the trading infrastructure, a time-series of trading-firm specific quotes and trades is automatically created and maintained.

WHERE TO EXECUTE

The question where to execute has two aspects:

1. within the existing set of LPs used by a trading firm,
2. other LPs not (currently) used.

The second of these requires a thorough understanding of the business model of the LP's being considered and the nature of their liquidity preferably as evidenced by sample historical data. With such historical data, analysis similar to that required for (1) can be carried out. Where to execute is preferably undertaken in real-time by a smart order routing (SOR) algo. SOR algos determine venue selection typically by looking at the best bid (or offer) simultaneously provided by each of the connected LPs and, as necessary, liquidity provided by each LP in lower levels of each order book. Of course, depending on the time of day, currency pair, order size, required aggressiveness etc, the SOR algo will send child orders to multiple LPs and may use multiple levels of liquidity. The intelligent SOR algo accounts for the historical profile of order fills and rejections. Even if a venue offers the current best price, it may be more risky to route the flow to this venue if historically this location has a high rejection rate. Thus, it is essential to SOR operation to have access to the time-series of market data, orders and actual executions in order to calculate fill and rejection profiles for each liquidity venue. The current choice of the SOR algo should be continually evaluated. This is done by back-testing candidate SOR algos (including different parameterization of the "same" algo) also against the trading firm's own time-series database.

HOW TO EXECUTE

The question how to execute clearly is a function of a set of business requirements such as required aggressiveness and market impact. Whether to use market orders, limit orders (static or pegged) or execution algos is again a process of formulating such candidate "execution methods" and back-testing these execution methods against the time-series database. This execution analysis provides the best execution method over the historical period tested. This approach also enables a library of execution methods to be available for use for given order requirements. Whilst this is a deterministic approach which can be operated systematically, the weakness is the reliance on historical market data. This deficiency can be mitigated by continually back testing these execution methods against the most recent market data and changing the algos (or parameterization thereof) and/or algo selection as required. This is particularly important for intraday trading strategies in which profit per trade is usually low and so achieving "good" execution is simply essential. "Good" is defined by minimizing the loss of potential profit on each trade and will very vary with each strategy.

For orders executed algorithmically, ongoing execution analysis is essential. Outside of keeping track of the performance of any broker execution algos a firm might be using, a trader needs to know how she is performing in respect of the chosen benchmark. This ongoing analysis will provide comfort or alert to unacceptable changes in execution quality in respect of over/underperformance. For example, for a given period, do all orders achieve similar out performance relative to the chosen benchmark: on each day of the week, for all order sizes, for all currencies? The answer is likely to be 'no' to at least one of these and so provide opportunities for improvement in either algo selection, algo parameterization or both.

ADAPTIVE ALGO'S

The holy grail, of course, is having fully adaptive algos: that is, those that change their behaviour in real-time in response to real-time market data. Ironically, such real-time feedback loops are part and parcel of systematic trading but it is still a relatively new concept in execution algos used in discretionary trading. One way to institute real-time modification of execution algos is to provide the (human) trader the ability to dynamically change attributes. In that way, ideas for improving execution generated by the research team can be implemented, manually. As comfort and acceptance is achieved, these real-time adjustments to the running algo can be implemented automatically.

In all cases, implementing continual systematic algo selection is both enabled and facilitated by maintaining a time-series database of quotes, orders and executions as a fully integrated component of the production trading infrastructure.