TraderServe Briefing Note

Played Out: Why market replay fails as a solution for MiFID II algorithm testing – Extended Version with example

Nicholas Hallam and Nick Idelson, TraderServe Limited, 12-16 Laystall Street, London, EC1R 4PF,
e: research@traderserve.com, w: www.traderserve.com, t: 02071935717.

Under MiFID II a new obligation is being placed on investment firms using European venues: they must test any algorithm used for trading, on any of a vast range of instruments, to ensure that it **"does not contribute to disorderly trading conditions"** and that it "continues to work effectively in stressed market conditions"[1]. Furthermore, trading venues "shall require their members to certify that the algorithms they deploy have been tested to avoid contributing to or creating disorderly trading conditions prior to the deployment or substantial update of a trading algorithm or trading strategy and explain the means used for that testing"[2]. ESMA's Final Report on the MiFID II Consultation offered some insight into the regulatory intention of this testing: it should "recreate real market conditions to ensure the well-functioning of algorithms under changing circumstances"[3]. It is apparent from these quotes that the type of testing envisaged is a far cry from the sort of profitability testing routinely performed by investment firms, but the problem goes deeper than this: **standard back-testing with historical data replayed to the algorithm is a fundamentally unsuitable approach to assessing that algorithm's propensity to cause or contribute to market disorder.** We argue this in the following paragraphs.

Problems with Market Replay

Market replay servers deliver historical data tick by tick to a running algorithm, allowing estimation of when and at what price a trade would have been initiated and when and at what price it would have been exited and, thereby, enabling calculation in simulation of an estimated profitability of the algorithm. The limitations of the method are well understood. In real markets, when an algorithm trades into an order book, it interacts with the market in question and it does so in three ways: first, by placing a passive order at a particular level it inhibits price movement through that level; secondly, by consuming liquidity from the market it can help move the price though current levels; and, thirdly, the placing and cancelling of its orders can stimulate the activity of other market participants including other algorithms. With market replay none of these effects is present. Nevertheless, with these provisos and perhaps making an allowance for the impact of the orders, the results may be of value for assessing the potential profitability of a strategy. However, where the stability of the algorithm is in question and where the disposition of the algorithm to cause or contribute to market disorder is the key issue, the limitations of market replay are fundamental: **the reaction of the market to the algorithm's activity is precisely what matters**.

---

[1] *MiFID II EC Delegated Regulation 19 July 2016 RTS6 supplementing 2014/65/EU specifying the organisational requirements of investment firms engaged in algorithmic trading* **Art 5, 4 (d)**
http://ec.europa.eu/finance/securities/docs/isd/mifid/rts/160719-rts-6_en.pdf

[2] *MiFID II EC Delegated Regulation 14 July 2016 RTS7 supplementing 2014/65/EU specifying organisational requirements of trading venues* **Art 10, 1**
http://ec.europa.eu/finance/securities/docs/isd/mifid/rts/160714-rts-7_en.pdf

[3] *ESMA Final Report Draft Regulatory and Implementing Technical Standards MiFID II/MiFIR* - **3.2.33**
https://www.esma.europa.eu/sites/default/files/library/2015/11/2015-esma-1464_-_final_report_-_draft_rts_and_its_on_mifid_ii_and_mifir.pdf

A Realistic, Responsive, Reactive Solution

TraderServe's AlgoGuard avoids these shortcomings. An emulator algorithm uses key statistical properties of each instrument's historical market data to send orders and cancels into a simulated exchange to create and maintain an artificial market with realistic microstructure. When a tested algorithm trading into this emulated market consumes liquidity, the volume is removed exactly as happens in a real market. If all available liquidity at the best bid or offer is consumed, the market moves naturally to a new level. In addition to these price changes which are a simple consequence of allowing the algorithm to interact with the emulated market in the same order book, the platform supports more subtle reactions through allowing interplay between the tested algorithm and other running algorithms. In the Appendix we show an example of algorithmic interaction resulting in a mini-flash crash.

For multi-market algorithms AlgoGuard links the simulated markets together by algos, in much the same way as happens in the real markets. **The propensity of the algorithm to contribute to disorder through interaction with other algos can be properly investigated with effective pass/fail measurement of disorder provocation**. In particular the behaviour of the algorithm can be examined in circumstances where the normal correlations between markets break down – i.e. those most likely to cause problems in respect of the algorithm's stability. None of this is possible with market replay as the relationship between the various markets cannot be any different from what it was historically regardless of the activity of the algorithm under test.

Replay-Based Solutions now being developed Despite Regulatory Awareness that Replay Technology is Unfit for Purpose

The limitations of using market replay technology to test algos for avoiding disorderly trading were evident both to regulators and respondents to the consultation process as these remarks from the Final Report make clear: "The main criticism received by ESMA was the difficulty to reproduce real market conditions in a non-live environment, where the simultaneous interaction with other relevant market players is a prerequisite. These respondents viewed such technical limitations significant enough to render such an exercise ineffectual as designing scenarios as close to market situations as possible would be fundamentally unachievable…"[4].

This being so, it is surprising to learn that a number of providers are now preparing replay-based solutions to the new regulatory requirement. One pervasive idea is that by replaying data from problem days when markets were stressed or exhibited disorderly trading conditions it will be possible to detect whether a given algorithm would have contributed to disorder. But this overlooks the arguments of our earlier paragraph: in the real markets, and in any simulation suitable for detecting the propensity of an algorithm to contribute to market disorder, the interaction of algorithms is key. This is not just a theoretical point: **analysis of both the 2010 S&P minis flash crash[5] and the 2014 US Treasuries flash crash[6] show market disorder emerging from the interaction of a number of different algorithms operating on different timescales**. Assessing what

[4] *ESMA Final Report Draft Regulatory and Implementing Technical Standards MiFID II/MiFIR* - **3.2.36**
https://www.esma.europa.eu/sites/default/files/library/2015/11/2015-esma-1464_-_final_report_-_draft_rts_and_its_on_mifid_ii_and_mifir.pdf

[5] *Briefing notes principally on the 6th May 2010 Flash Crash and its Implications for Non-Live Testing*
http://www.traderserve.com/pdf/TS-briefingnote0n6thMay2010FlashCrashandNon-LiveTesting.pdf

[6] *Oct 15, 2014 Flash Crash in the US Treasury Markets & its implications for Non-Live Testing*
http://www.traderserve.com/pdf/TraderServe-BriefingNoteon15oct2014TreasuriesFlashCrash-20160126.pdf

contribution an algorithm might make to market disorder requires allowing it to interact with other algorithms. This cannot be achieved in static historical replay.

Additional Problem for Replay: How to test Shutdown

There is a further reason why a replay-based approach cannot meet the MiFID II requirements. RTS 6, Article 14.3 says this: "An investment firm shall ensure that its trading algorithm or trading system can be shut down in accordance with its business continuity arrangements without creating disorderly trading conditions." How is replay going to help here? A major threat to market order from shutting down an algorithm flows from the cancellation of its resting orders. If a large amount of volume close to the top of the order book is removed all at once this can destabilise the market. Exactly such a mechanism has long been used by spoofing algorithms to commit market abuse. Or if a very large number of orders are pulled simultaneously this can induce significant latency and desynchronization of markets, a known type of disorderly trading condition. Neither of these is testable in a replay scenario as the orders are never actually entered into an order book with orders representing the replayed market.

Conclusions

Properly testing an algorithm's stability needs an environment in which the algorithm can place orders and cancels into a responsive market which has realistic microstructure and is capable of producing the sort of stressed conditions that often prove problematic to poorly designed or reckless algorithms. The platform must also facilitate the interplay between algorithms from which disorder can emerge[7]. In practice this means that tested algorithms must interact in the same order book as orders reproducing the microstructure of a real market and those generated by additional algorithms whose purpose is to antagonise the algorithm under test. That this is not easy to achieve is recognised in the financial industry, which is why ESMA reports that some respondents to the MiFID II consultation, presumably familiar with market replay servers, opined that the technology was "fundamentally unachievable"[8]. Nevertheless, such a testing technology has been accessible and, indeed, in production at a major exchange for a number of years and now, in the form of AlgoGuard[9], is available to all firms engaged in algorithmic trading.

Appendix: Flash Crash Roulette in action : Interacting Algorithms Causing Market Disorder

The four charts below show market activity in the AlgoGuard market on RDSA (London Stock Exchange) on 4 separate runs emulating from the same 3 minutes of historical data. The mid-price between the bid and offer is shown as a white line against the left axis; volume activity is shown in coloured bars against the right axis.

---

[7] *Trading Algorithms, Disorderly Markets and Non-Live Testing A study of emergent behaviours supporting the case for non-live testing regulations*
http://www.traderserve.com/download.php?file=publicdomainresearch/Trading%20Algorithms-Disorderly%20Markets-Non-Live%20Testing-20141202-researchpaper_final.pdf

[8] *ESMA Final Report Draft Regulatory and Implementing Technical Standards MiFID II/MiFIR* - **3.2.36**
https://www.esma.europa.eu/sites/default/files/library/2015/11/2015-esma-1464_-_final_report_-_draft_rts_and_its_on_mifid_ii_and_mifir.pdf

Fig 1 below shows a simple emulation of an RDSA orderbook with no additional algorithms running. The volumes shown (in dark blue) are of all orders of the emulator on buy and sell sides.

Side	Orders Placed	Orders Cancelled	Cancel Ratio	Shares Placed	Shares Filled	Shares Cancelled	Shares Filled	Shares Cancelled	Max Shares Ordered Per Sec	Max Shares Cancelled Per Sec
Buy	615	616	1.002	514,906	24,380	352,248	4.7%	68.4%	173,085	47,805
Sell	749	861	1.150	701,031	24,380	515,405	3.5%	73.5%	177,332	42,556
Total	1,364	1,477	1.083	1,215,937	48,760	867,653	4.0%	71.4%	350,417	90,361

Fig 1 Simple emulation of RDSA order book

In Fig 2 below, a poorly designed sell execution algo trades on four occasions with little effect on the price compared with the first chart (though the price troughs are slightly deepened).

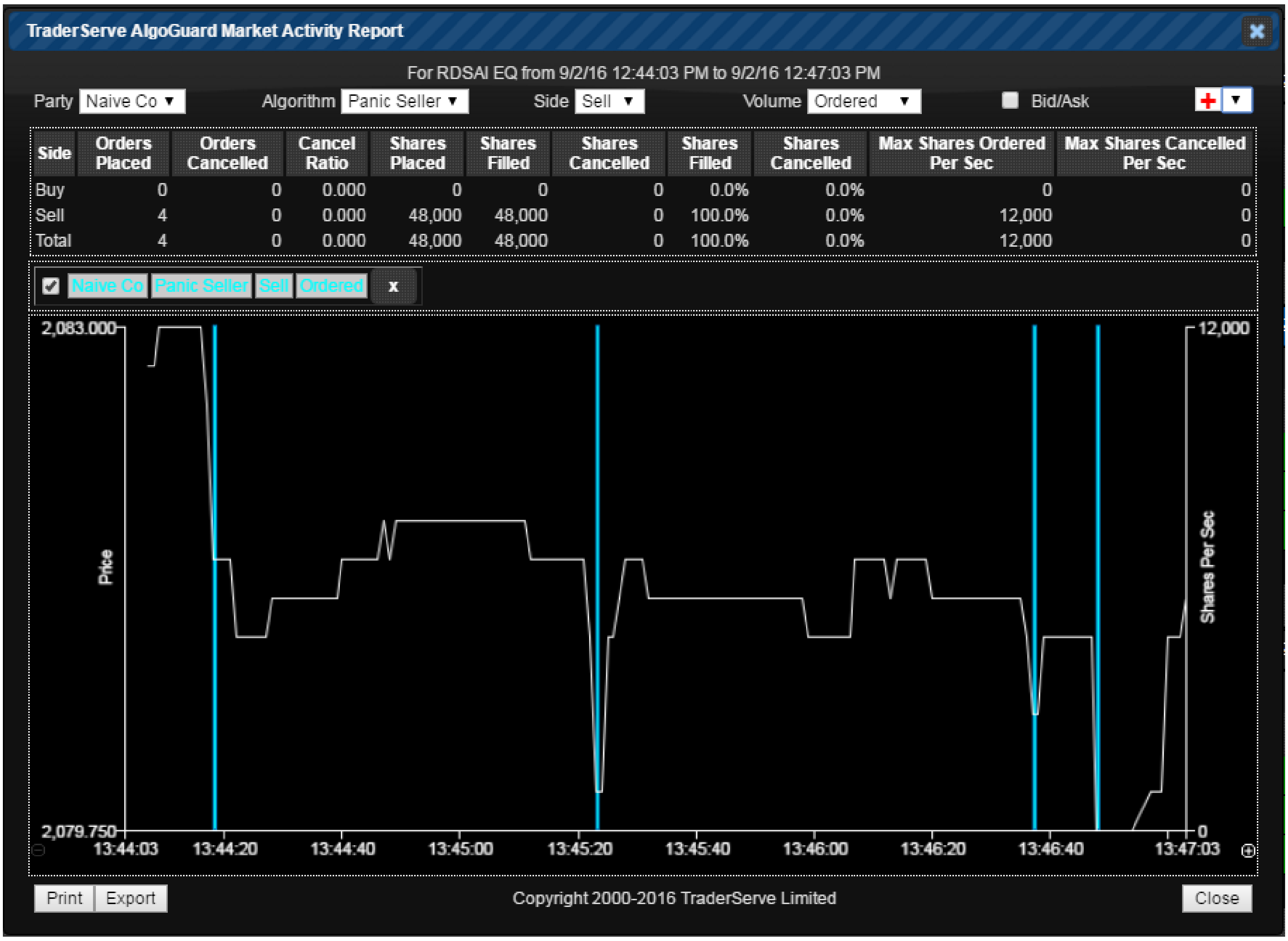

Fig 2 RDSA order book emulation with poorly designed sell execution algorithm

In Fig 3 below a predatory momentum ignition algorithm is turned on with the emulator: this sells through multiple levels on the offer when the volume on those levels is sufficiently low, placing an equal sized order on the offer just inside the new level. The conditions triggering a trade occur on 5 occasions, shown as yellow sell volumes in the chart. Again there is little influence on price.

Side	Orders Placed	Orders Cancelled	Cancel Ratio	Shares Placed	Shares Filled	Shares Cancelled	Shares Filled	Shares Cancelled	Max Shares Ordered Per Sec	Max Shares Cancelled Per Sec
Buy	5	0	0.000	69,519	37,295	0	53.6%	0.0%	18,467	0
Sell	5	0	0.000	69,519	69,519	0	100.0%	0.0%	18,467	0
Total	10	0	0.000	139,038	106,814	0	76.8%	0.0%	36,934	0

Fig 3 RDSA order book emulation with predatory momentum ignition algorithm

In Fig 4 below both the execution strategy and the momentum ignition algo are turned on together with the emulator. The effect is dramatic: each provokes additional activity in the other resulting in 23 sell trades from the execution strategy and 23 sell trades from the momentum ignition algo: the execution strategy is sensitive to sudden drops precipitated by the momentum ignition algo which in turn finds the execution strategy exposing new low volume levels on the offer causing it to trigger again. Through this interdependency, very large selling pressure is felt on the market, which falls 2.5% in three minutes.

This illustrates the way "emergent market disorder" can result from the interactions between algorithms. This sort of dangerous interdependence cannot be revealed from testing against replayed historical data.

Fig 4 RDSA order book emulation with sell execution algorithm and momentum ignition algorithm