อ่านกระดาน
Reading the Book

Part 11: MM-3 ลงมือจริง — จาก quote สู่ระบบรันได้

Order Flow & Microstructure สำหรับรายย่อย · องก์ IV Market Making

MM loop, microprice, post-only, inventory limit + kill switch และกับดัก backtest 3 ข้อ

Part 11 — MM-3 ลงมือจริง + กับดัก backtest

🎯 **อ่านจบ Part นี้คุณจะ**

- เห็นโครง **MM loop** ที่รันได้จริง: รับ L2 + trade stream → คำนวณ → ตั้ง/ยกเลิก quote วน
- รู้จัก **microprice** — ราคาอ้างอิงที่ดีกว่า mid เพราะถ่วงน้ำหนักด้วยความหนาแน่นสองฝั่ง
- ใช้ **post-only** เพื่อการันตีเป็น maker และตั้ง **inventory limit + kill switch** กันพอร์ตระเบิด
- รู้ทัน **กับดัก backtest 3 ข้อ** ที่ทำให้กราฟสวยกลายเป็นขาดทุนจริง: fill assumption, latency, market impact

backtest คือกระจกห้องลองเสื้อ

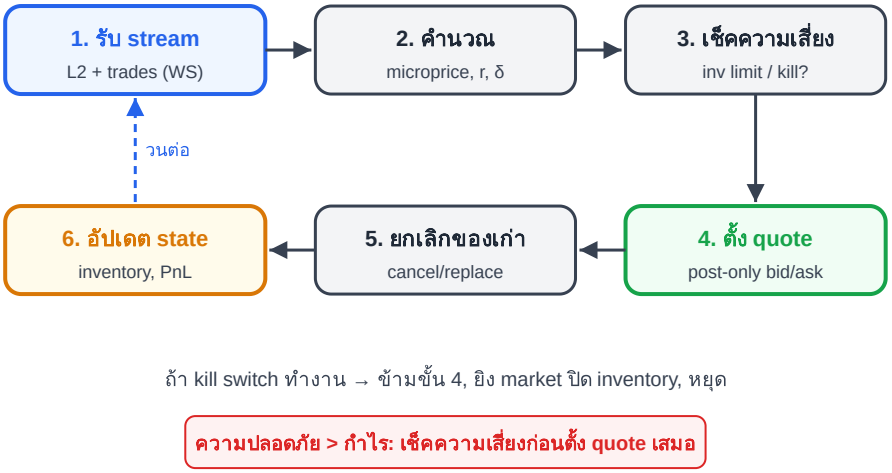
เคยลองเสื้อในห้องลองที่ห้างไหม? — กระจกเอียงนิด ไฟวอร์มนิด ทุกตัวดูดีไปหมด พอซื้อกลับบ้านมาส่องกระจกตัวเองถึงรู้ว่า "มันไม่เหมือนในห้องลองเลย"

Backtest ของกลยุทธ์ MM ก็คือกระจกห้องลองเสื้อ — มันถูกออกแบบ (โดยตั้งใจหรือไม่ตั้งใจ) ให้ผลลัพธ์ดูสวยกว่าความจริง เพราะมันตั้งสมมติฐานเข้าข้างตัวเองหลายอย่าง โดยเฉพาะเรื่อง "เราได้ fill ทุก quote ที่ราคาแตะ" ซึ่งในตลาดจริง **ไม่จริงเลย**

Part นี้แบ่งสองครึ่ง: ครึ่งแรกประกอบ "ร่างกาย" ของ MM ที่รันได้จริง ครึ่งหลังคือ "กระจกเงาความจริง" — สามกับดัก backtest ที่ต้องรู้ก่อนเอาเงินจริงไปเสี่ยง

MM loop: ร่างกายของระบบ

ระบบ MM ที่รันได้จริงคือลูปที่หมุนตลอดเวลา รับข้อมูลสองสาย (L2 order book + trade stream ผ่าน WebSocket) แล้วตัดสินใจตั้ง/ยกเลิก quote:



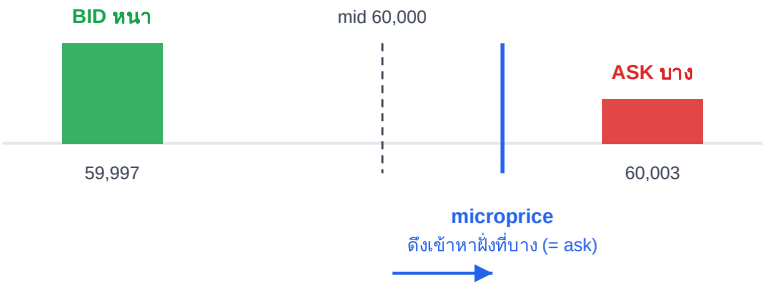
ภาพ 11.1 · MM loop — รับ stream → คำนวณ → เช็คความเสี่ยง → ตั้ง/ยกเลิก quote → วน

Microprice: ราคาอ้างอิงที่ดีกว่า mid

mid = (bid+ask)/2 มองสองฝั่งเท่ากันเสมอ แต่ถ้าฝั่ง bid หนากว่าฝั่ง ask มาก ๆ ราคา "จริง ๆ" มีแนวโน้มถูก**ดึงเข้าหาฝั่งที่บาง (= ask) microprice** แก้จุดนี้ด้วยการถ่วงน้ำหนักด้วยปริมาณฝั่งตรงข้าม:

```
microprice = (p_bid · V_ask + p_ask · V_bid) / (V_bid + V_ask)

สังเกต: ฝั่งที่ "หนากว่า" จะดึง microprice เข้าหาฝั่งที่บาง (ฝั่งตรงข้าม)
V_bid > V_ask (แรงซื้อหนา) → microprice ดึงเข้าหาฝั่งที่บาง (= ask) → มีแนวโน้มขึ้นเชิงสถิติระยะสั้น
ใช้ microprice แทน s ในสูตร reservation price ของ Part 10 ได้
```



ภาพ 11.1b · microprice ดึงเข้าหาฝั่งที่บาง (= ask) เมื่อ bid หนากว่า ask

Post-only, inventory limit, kill switch

เครื่องมือ	ทำอะไร	ทำไมขาดไม่ได้
post-only	limit ที่ยกเลิกอัตโนมัติถ้าจะกลายเป็น taker	การันตีเป็น maker เสมอ → ได้ rebate / ไม่จ่าย taker fee + spread โดยบังเอิญ
inventory limit	หยุดตั้ง quote ฝั่งที่ทำให้ |q| เกินเพดาน	กัน inventory วิ่งหนีฝั่งเดียวจนเกินที่รับความเสี่ยงไหว
kill switch	ปิด inventory ทันที + หยุดบอท เมื่อขาดทุน/ผันผวนเกินกำหนด	กันหายนะตอนตลาดเกิดเหตุผิดปกติ (flash crash, ข่าวแรง)

```
# pseudo-code: ก้อนความปลอดภัยใน MM loop
if abs(inventory) >= INV_LIMIT:
    skip_quote_on_growing_side() # ไม่เติมฝั่งที่ทำให้ q โตขึ้น
if pnl_drawdown > MAX_DD or vol > VOL_CAP:
    market_close_all(); halt() # kill switch
else:
    place_order(bid, post_only=True)
    place_order(ask, post_only=True)
```

กับดัก backtest 3 ข้อ (หัวใจของ Part)

นี่คือ "กระจกเงาความจริง" — สามเหตุผลที่กราฟ backtest สวยมักโกหก:

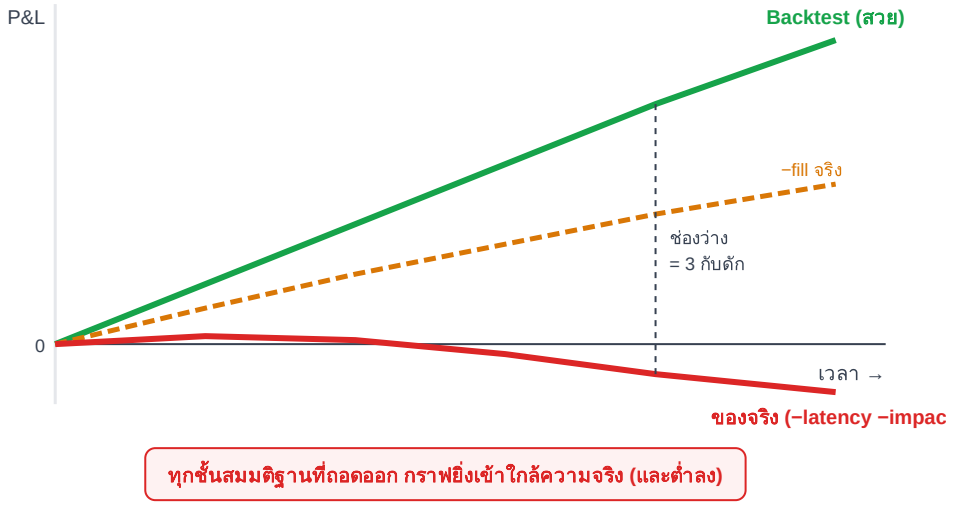
⚠️ กับดัก 1 — Fill assumption (อันตรายที่สุด)

Backtest ส่วนใหญ่สมมติว่า "**ถ้าราคาแตะ quote ของเรา = เราได้ fill**" — ผิดมหันต์! ในความจริงคุณอยู่ **ท้ายคิว** (Part 1) ราคาต้องแตะ *แล้วเทรดทะลุคิวข้างหน้าคุณหมด* คุณถึงได้ fill และร้ายกว่านั้น: quote ที่ได้ fill ง่ายมักเป็นจังหวะ **adverse selection** (ราคากำลังวิ่งสวนคุณ) backtest ที่ assume fill ฟรีจึงนับกำไรที่ไม่มีจริง และมองข้ามการขาดทุนที่มีจริง

⚠️ กับดัก 2 — Latency (ความหน่วง)

Backtest มองเห็นข้อมูลและตอบสนอง "ทันที 0 วินาที" แต่ของจริงมี latency ทั้งขารับข้อมูลและขายิงคำสั่ง กว่าคุณจะเห็น microprice ขยับแล้วยกเลิก quote เก่า ตลาดเคลื่อนไปแล้ว — **quote เก่าที่ยังค้างอยู่คือ quote ที่โดน "เก็บ" ในจังหวะที่แย่ที่สุด** รายย่อยที่แข่ง latency กับ HFT คือเอาจักรยานไปแข่งกับรถแข่ง

> ⚠️ **กับดัก 3 — Market impact (เราคือส่วนหนึ่งของตลาด)**
>
> Backtest รันบนข้อมูลในอดีต **ที่ไม่มีตัวเราอยู่ในนั้น** มันสมมติว่า order ของเราไม่กระทบกระดานเลย — แต่ของจริง quote ของเราเปลี่ยน OBI, เบียดคิวคนอื่น, และถ้าเราต้องยิง market ปิด inventory เราจะ walk the book ดันราคาสวนตัวเอง (Part 2) **ยิ่งไม้ใหญ่ ยิ่ง impact มาก กำไรในกระดาษหายไปกับ impact ที่ backtest มองไม่เห็น**



ภาพ 11.2 · backtest สวย vs ความจริง — fill assumption + latency + market impact กินกำไรหมด

บันได backtest → paper → live

อย่ากระโดดจาก backtest ไป live ตรง ๆ ให้ไต่บันได เพื่อให้ความจริงค่อย ๆ เผยตัว:

- **Backtest** — เร็ว ถูก แต่โกหกได้ (3 กับดัก) ใช้คัดไอเดียที่แย่ชัด ๆ ออกเท่านั้น
- **Paper / forward test** — รันสด ๆ บนตลาดจริงด้วยเงินปลอม เห็น latency จริง เห็นว่า quote ได้ fill จริงไหม
- **Live ไม้เล็กสุด** — เงินจริงจำนวนน้อยมาก วัด **adverse fill rate (AFR)** และ PnL_net หลังหัก fee จริง ค่อยขยายถ้ารอด

```
microprice = (p_bid·V_ask + p_ask·V_bid)/(V_bid+V_ask)
PnL_net = PnL_spread + q·Δp + rebate − taker_fee_on_closes − slippage
AFR = adverse_fills / total_fills (ยิ่งสูง = โดน adverse selection เยอะ)
```

📑 งานวิจัยรองรับ

ความสำคัญของ queue position และ fill ที่ไม่ใช่ "แตะแล้วได้" สะท้อนในงานสาย queue dynamics เช่น **Lu & Abergel (2018)** และ *Importance of Order Sizes* (arXiv 2405.18594) ส่วนการต่อยอด A-S ให้รับมือ latency/inventory จริงมีใน *Resolving Latency & Inventory Risk in MM* (arXiv 2505.12465) และงาน RL บน LOB ที่ไม่นิ่ง (arXiv 2509.12456) — ทั้งหมดตอกย้ำว่า **ช่องว่างระหว่าง backtest กับ live คือ fill realism, latency และ impact** ไม่ใช่ความฉลาดของสูตร

🔗 เชื่อมโยงเล่มอื่น

microprice ใช้ OBI/depth จาก **Part 5** และ adverse selection จาก **Part 6** โดยตรง · PnL_net คือ payoff ที่หักต้นทุนจริงครบทุกชั้น ตรงกับวินัยของ **Payoff Mastery (pm)** ที่ย้ำว่าต้องคิดต้นทุนการเข้า-ออกจริง · ส่วน effective price ตอนปิด inventory (walk the book) คือต้นทุนเดียวกับที่เล่ม **arb** ใช้คำนวณว่าโอกาส arb คุ้มจริงไหม

✍️ แบบฝึก (ทำได้ด้วยข้อมูลคริปโตฟรี)

เป้าหมาย: "จับโกหก" ตัวเองให้ได้ก่อนเสียเงินจริง

1. ดู pseudo-code ลูปด้านล่าง หา **fill assumption ที่อันตราย** แล้วอธิบายว่าทำไมมันทำให้ backtest สวยเกินจริง
2. แก้ฟังก์ชัน fill ให้สมจริงขึ้น: fill ได้ก็ต่อเมื่อ volume ที่เทรดผ่านราคานั้น > คิวที่อยู่ข้างหน้าเรา
3. ใส่ latency จำลอง: หน่วงการเห็นข้อมูลและการยกเลิก quote สัก 100–300 ms แล้ววัด AFR ใหม่
4. เทียบ P&L_net 3 เวอร์ชัน: (ก) fill ฟรี (ข) fill ตามคิว (ค) fill ตามคิว + latency — ดูว่ากราฟต่ำลงทีละชั้นจริงไหม

```
# pseudo-code มีกับดัก — หา fill assumption ที่ผิด
for tick in backtest_data:
    bid, ask = quote(microprice(tick), inventory, ...)
    if tick.low <= bid: # <-- กับดัก! แตะ = fill?
        inventory += 1; cash -= bid
    if tick.high >= ask: # <-- กับดักเดียวกัน
        inventory -= 1; cash += ask
# ของจริง: ต้องเช็คคิวข้างหน้า + latency + ไม่นับ fill ที่ราคาเพิ่งทะลุผ่านเฉย ๆ
```

ต่อ Part 12: Grid Trading กับกระดาน — ยกระดับ grid ด้วย OBI/OFI และ VPIN กันกริดแตก →