Algobot
Maximize Your Profits

Adaptive Transition Matrix Bias

Expert Advisor Documentation

PLATFORM	TYPE	TIMEFRAME	WEBSITE
MetaTrader 5 (MT5)	Adaptive Statistical Bias (Markov)	M15 / H1 (Intraday)	www.algobot.live

> ⚠ **Important Disclaimer** This document is for educational and informational purposes only. It does not constitute financial or investment advice. Trading forex, CFDs, and other leveraged instruments involves substantial risk of loss and is not suitable for all investors. Past backtest performance does not guarantee future results. Never trade with capital you cannot afford to lose.

Overview

Adaptive Transition Matrix Bias is an original, first-principles strategy that models the market as a stream of discrete symbols rather than a series of prices. Every completed bar is reduced to one of three states — `UP`, `DOWN`, or `FLAT` — and the Expert Advisor learns a self-updating **3×3 Markov transition matrix** that measures how likely each state is to be followed by another. From the row belonging to the current state it reads a one-step directional forecast for the *next* bar and trades it — but only when the forecast is statistically significant.

The EA uses **no indicator overlays**, no chart patterns, and no support/resistance, SMC, ICT, or Wyckoff methodology. The Average True Range (ATR) appears solely as a neutral volatility yardstick — it sets the width of the "no-effort" dead-zone and scales the stop, target, and trailing distances. It is never itself a trading signal.

> **Core hypothesis.** If prices were a pure random walk, the symbol stream would be memoryless and the next state would be independent of the current one. This strategy bets that real order flow leaves a *short-memory, time-varying fingerprint* in that stream: the conditional distribution of the next symbol departs from the memoryless baseline, and that departure drifts as the market regime changes. Capturing that conditional asymmetry — while it is statistically real — is the entire edge.

How It Works

1. Symbolizing each bar

Each completed bar is classified by comparing its signed body against a volatility dead-zone. The dead-zone width is a fraction of ATR (parameter `FlatZoneAtr`), so a bar only counts as directional when its net effort clears the ambient noise:

```
body  = Close - Open              (signed body of the bar)
dead  = FlatZoneAtr * ATR         (volatility dead-zone = "no meaningful effort")

state = UP    if  body >  dead
        DOWN  if  body < -dead
        FLAT  otherwise
```

2. The self-learning transition matrix

The EA keeps a 3×3 matrix `C` of **recency-weighted** transition counts, where `C[s][s']` is the weighted number of times state `s` was followed by state `s'`. On every new bar the whole matrix is decayed by the forgetting factor `Lambda` and the newest transition is registered. This exponential forgetting lets the estimate track the current regime and discard stale structure (effective memory ≈ `1 / (1 - Lambda)` bars):

```
every new bar:   C  <-  Lambda * C          (exponential forgetting / decay)
                 C[prev][cur] += 1          (register the newest transition)
```

3. Reading the directional edge

From the row of the *current* state `s` the EA reads the maximum-likelihood one-step forecast for the next bar. The **edge** `e` is simply the probability of an UP transition minus the probability of a DOWN transition:

```
n   = C[s][DOWN] + C[s][FLAT] + C[s][UP]     (effective transitions seen out of s)
pU  = C[s][UP]   / n
pD  = C[s][DOWN] / n
e   = pU - pD                in [-1, 1]     (directional edge for the NEXT bar)
```

4. The self-adapting significance gate

Because `e` is a difference of two sample proportions, it carries sampling noise. The EA estimates that noise and only acts when the edge clears `SignificanceZ` standard errors *and* at least `MinTransitions` observations have accumulated:

```
Var(e) = ( pU + pD - e^2 ) / n
SE     = sqrt( Var(e) )
band   = SignificanceZ * SE

LONG   when  e >  band     (chain out of s is biased UP)
SHORT  when  e < -band     (chain out of s is biased DOWN)
```

This band is fully self-adapting. With only a few observations out of the current state, a *large* edge is demanded before trading; once many consistent transitions accumulate, a small but persistent edge is enough. There is no fixed price threshold anywhere — the decision rescales itself to the weight of evidence.

5. Entry logic

- The current bar's state indexes the forecast row, and the EA enters at the **open of the next bar** (the live close of the current bar), so there is **no look-ahead** — only completed bars are ever used.
- **Long** when `e > band` (the chain out of the current state is biased UP).
- **Short** when `e < -band` (the chain out of the current state is biased DOWN).
- **One position per Magic** number at a time — no pyramiding.

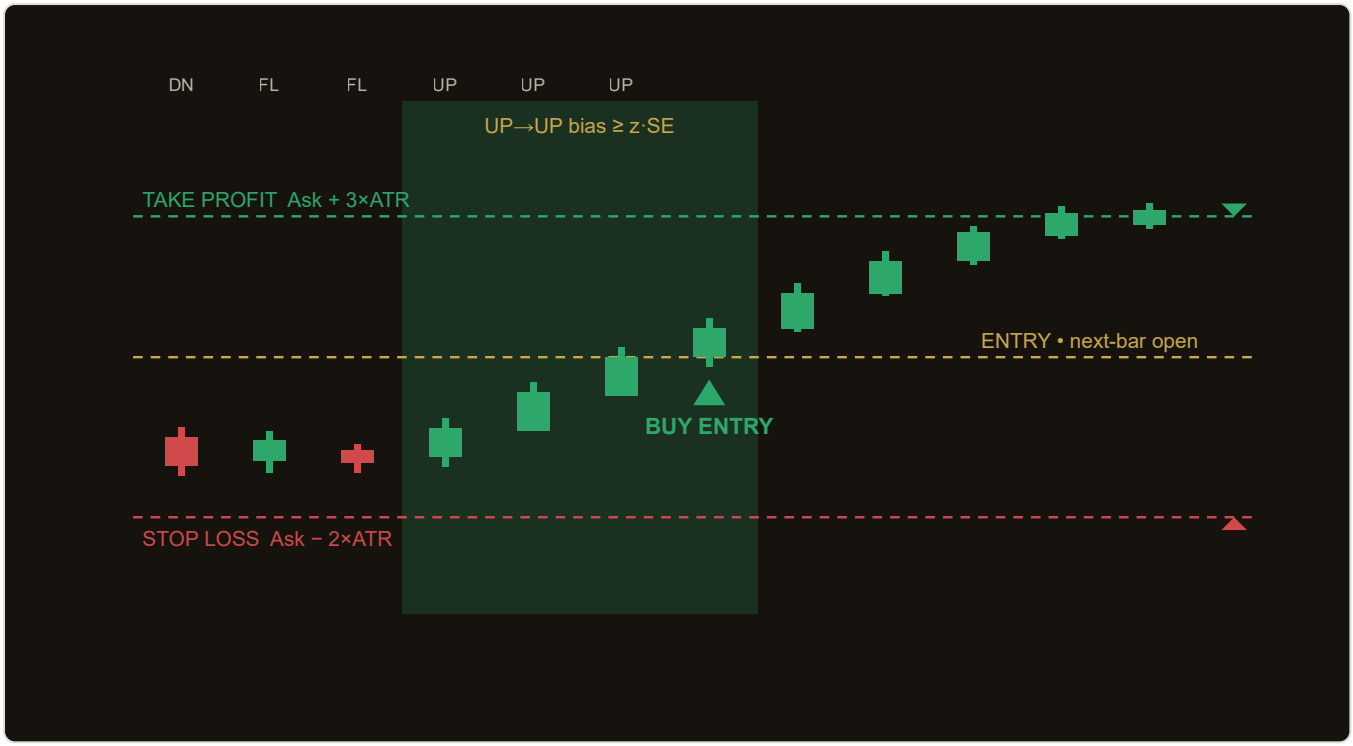
6. Exit logic — stop, target, trailing & flip

- **Stop loss / take profit:** both are ATR-scaled. A long is protected at `Ask − StopAtrMult×ATR` with a target at `Ask + TargetAtrMult×ATR` (mirrored for shorts).
- **Trailing stop:** an ATR-based trailing stop (`TrailAtrMult`) that only ever ratchets in the trade's favour and is managed on *every tick* against live price. Set `TrailAtrMult = 0` to disable it.
- **Adaptive flip-exit:** if a significant edge reverses against an open trade — a long while the chain flips DOWN, or a short while it flips UP — the position is closed immediately. This regime-change exit acts independently of the fixed stop.

> **Causality guarantee.** Learning and decisions happen once per completed bar, and entries use the just-closed bar to forecast the bar that is only now opening. No forming-bar data ever influences a decision, so the logic is strictly causal in both the C# and MQL5 implementations.

Strategy in Action

The illustration below shows an example of how the strategy identifies a setup and triggers its entry and exit. This is a simplified, illustrative example for educational purposes — not real market data.



Illustrative example only. Actual market behaviour varies.

Reading the illustration

After two choppy `FLAT` bars the market prints three consecutive `UP` bars. The recency-weighted matrix now shows a strong `UP→UP` transition whose edge clears the `z·SE` significance band (shaded zone). The EA goes **long at the open of the next bar**, places the stop `2×ATR` below and the target `3×ATR` above, and trails the stop upward as price advances toward take profit.

Parameters

Parameter	Default	Description
Lambda	`0.97`	Forgetting factor for the transition matrix. Effective memory ≈ 1/(1−Lambda) bars. Range 0.90–0.995, step 0.005. Higher = longer memory, slower to adapt.
FlatZoneAtr	`0.10`	Dead-zone width as a fraction of ATR. A bar is FLAT unless its body exceeds this. Range 0.00–0.50, step 0.02. Higher = fewer directional states.
SignificanceZ	`1.50`	Width of the self-adapting significance band in standard errors. Range 0.50–3.00, step 0.10. Higher = stricter, fewer trades.
MinTransitions	`6`	Minimum effective transitions (n) out of the current state before any trade is allowed. Range 3–20, step 1.
AtrPeriod	`14`	ATR look-back period used for the dead-zone and all risk distances. Range 5–40, step 1.
StopAtrMult	`2.0`	Stop-loss distance as a multiple of ATR. Range 0.5–5.0, step 0.1.
TargetAtrMult	`3.0`	Take-profit distance as a multiple of ATR. Range 0.5–8.0, step 0.1.
TrailAtrMult	`2.0`	Trailing-stop distance as a multiple of ATR. Set to 0 to disable trailing. Range 0.0–6.0, step 0.1.
Lots	`0.10`	Fixed trade volume in lots. Range 0.01–1.0, step 0.01.
Magic	`4207`	Magic number identifying this EA's positions. Use a unique value per chart to keep trades isolated.

Recommended Settings

The EA is timeframe-agnostic — it locks to no fixed period and runs on whatever chart it is attached to. A liquid FX major such as **EUR/USD** on an intraday timeframe (**M15** or **H1**) is the recommended proving ground, where the symbol stream is dense enough for the transition matrix to accumulate meaningful statistics.

SUGGESTED STARTING POINT

- **Symbol / Timeframe:** EUR/USD, M15 or H1
- **Lambda:** 0.97 (about 33 bars of effective memory) — balances responsiveness and stability
- **SignificanceZ:** 1.50 with **MinTransitions** 6 — a moderate quality filter
- **Risk:** StopAtrMult 2.0, TargetAtrMult 3.0 (a 1.5:1 reward-to-risk), TrailAtrMult 2.0

Tuning tips. To trade more selectively (fewer, higher-conviction entries), raise `SignificanceZ` and `MinTransitions`. To adapt faster to changing regimes, lower `Lambda` — but note this shortens memory and can make the edge estimate noisier. Always re-run the Strategy Tester after any change and validate on out-of-sample data.

Always backtest first. The default parameters are a balanced starting point, not an optimized configuration. Run the MT5 Strategy Tester across a representative history and, ideally, walk-forward on unseen data before considering any live deployment. Optimizing too aggressively on one period risks curve-fitting.

How to Install on MetaTrader 5

1. Copy `AdaptiveTransitionMatrixBias.ex5` to your MT5 `MQL5\Experts\` folder
2. Restart MetaTrader 5 and refresh the Navigator panel
3. Drag the EA onto a chart matching the recommended symbol and timeframe
4. Configure the input parameters and click **OK**
5. Enable **Algo Trading** in the MT5 toolbar

Risk Warning

Trading foreign exchange, CFDs, and other leveraged financial instruments involves substantial risk of loss and is not suitable for all investors. The strategies and tools described in this document are provided for **educational purposes only** and do not constitute financial advice, investment recommendations, or solicitation to trade. Always consult a qualified financial adviser before making trading decisions. Past backtest performance is not indicative of future results.