

Kalman Velocity Trend

Expert Advisor Documentation

PLATFORM

MetaTrader 5 (MT5)

TYPE

Trend Following (Pullback)

TIMEFRAME

H1 – H4 (configurable)

WEBSITE

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

Kalman Velocity Trend is a trend-pullback Expert Advisor built on a two-state, constant-velocity **Kalman filter**. Rather than relying on a moving average — which only estimates a smoothed price level and lags badly when you try to read its slope — the filter tracks two hidden states of the market simultaneously:

- **Level** — the filter's best estimate of "fair" price, stripped of tick noise.
- **Velocity** — the estimated price change per bar; a directly-estimated, noise-reduced measure of trend slope.

The velocity state is where the edge lies. Because it is estimated jointly with the level (not computed after the fact from a lagging average), it turns earlier than an MA slope would, and its sign and magnitude form a clean, self-scaling gauge of the prevailing regime. The EA only trades in the direction that velocity points, and only enters on a **pullback that reclaims the filtered level** — buying dips in uptrends and selling rallies in downtrends instead of chasing extended moves.

Process noise and measurement noise are auto-scaled to each symbol's own volatility by expressing them as fractions of **ATR**. That means the same handful of parameters behave sensibly on any pair or timeframe, with no per-symbol tuning of raw variances. The strategy runs on whatever single timeframe you attach it to.

Why a Kalman filter instead of a moving average? An MA gives you one number — a smoothed level — and you have to infer direction by differencing it, which adds lag and noise. The Kalman filter estimates direction (velocity) as a first-class state, updated optimally each bar from the newest close. You get a smoother *and* more responsive read of the trend at the same time.

How It Works

The filter model

One filter step corresponds to one closed bar. The state vector is $x = [\text{level} , \text{velocity}]$, and the model assumes constant velocity between bars:

```
predict:  level' = level + velocity
          velocity' = velocity
measure:  z = Close      (the close observes level only)
```

Each bar the filter runs a standard predict/update cycle:

- **Predict** — project the state forward using the constant-velocity transition $F = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$, then inflate the covariance by the process-noise variance Q .
- **Update** — take the just-closed bar's `Close` as the measurement z , compute the innovation $y = z - \text{level}'$ and the Kalman gain, and correct both the level and velocity estimates.

Both noise terms are auto-scaled to volatility: $Q = (\text{ProcNoiseFrac} \times \text{ATR})^2$ and $R = (\text{MeasNoiseFrac} \times \text{ATR})^2$. A larger **ProcNoiseFrac** makes the filter trust the model less and react faster; a larger **MeasNoiseFrac** makes it trust each individual close less and smooth harder.

Entry logic

Signals are distinct and fully symmetric for long and short. Two conditions must line up on the close of a bar:

- **Trend gate** — the estimated $|\text{velocity}|$ must exceed $\text{VelThreshFrac} \times \text{ATR}$. In other words the drift has to be a meaningful fraction of one bar's noise before any trade is allowed. The *sign* of velocity decides the only side that can be taken: positive → up regime (longs only), negative → down regime (shorts only).
- **Pullback reclaim** — in an up regime, price must have dipped *below* the filtered level on the prior bar and then *closed back above* it. This is a pullback into the trend that has resumed. The short side is the mirror image: price pokes above the filtered level, then a bar closes back below it.

A **long** fires when the up regime and an up-reclaim coincide; a **short** fires when the down regime and a down-reclaim coincide.

Long setup, step by step

1. Filtered velocity is clearly positive and above the gate → up regime confirmed.
2. Price pulls back and one or more bars close below the Kalman level line.
3. A subsequent bar closes back above the level line → pullback reclaim.
4. The EA buys at Ask with an ATR stop and target. The short side is exactly reversed.

Position management

The EA holds at most **one position per magic number** at a time. When a fresh signal appears in the opposite direction to an open trade, that trade is **closed first (reversal)** and the new position is opened. If a same-direction position is already open, no duplicate is added.

Exit logic — stop-loss & take-profit

Every position is protected by ATR-based brackets set at entry, using the ATR of the just-closed bar:

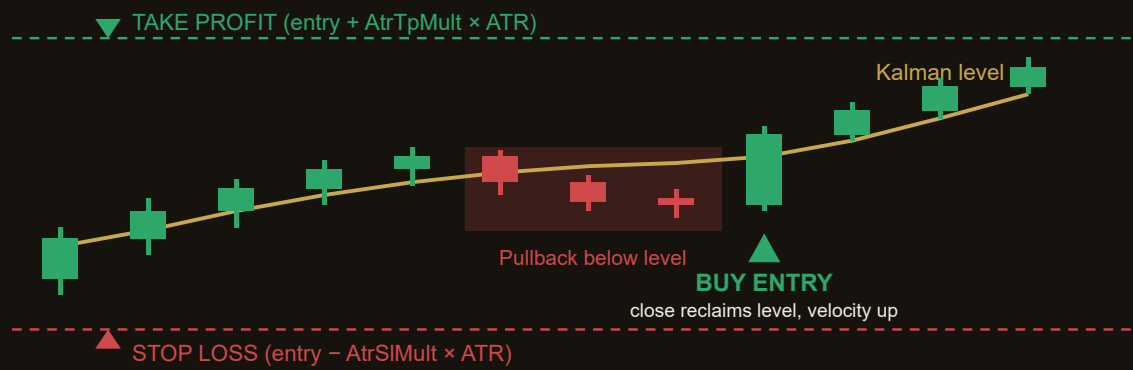
- **Long:** Stop-loss = entry - $\text{AtrSlMult} \times \text{ATR}$ | Take-profit = entry + $\text{AtrTpMult} \times \text{ATR}$
- **Short:** Stop-loss = entry + $\text{AtrSlMult} \times \text{ATR}$ | Take-profit = entry - $\text{AtrTpMult} \times \text{ATR}$

With the defaults ($\text{AtrSlMult} = 2.0$, $\text{AtrTpMult} = 3.0$) each trade risks two ATR to make three, a 1.5 : 1 reward-to-risk profile. A trade can also be exited early by the reversal mechanism if the opposite signal arrives before either bracket is hit. Sizing is a fixed lot (Lots).

New-bar processing. All logic runs once per completed bar — the EA detects a new bar and evaluates on the candle that just closed (shift 1), never on the still-forming candle. This avoids intrabar repainting and keeps live behaviour consistent with backtests.

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.

Parameters

Parameter	Default	Description
ProcNoiseFrac	0.10	Process-noise sigma as a fraction of ATR. Larger values make the filter trust the constant-velocity model less and react faster to price (more responsive velocity). Range 0.02–0.50, step 0.02.
MeasNoiseFrac	1.00	Measurement-noise sigma as a fraction of ATR. Larger values make the filter trust each individual close less and smooth harder (steadier, slower velocity). Range 0.25–3.00, step 0.25.
VelThreshFrac	0.05	Trend gate: the required <code> velocity </code> as a fraction of ATR before any trade is allowed. Higher values demand a stronger drift and filter out chop. Range 0.00–0.30, step 0.01.
AtrPeriod	14	ATR length used for volatility-based noise scaling, the trend gate, and stop / target distances. Integer, range 7–30, step 1.
AtrSlMult	2.0	Stop-loss distance as a multiple of ATR: $SL = \text{entry} -/+ \text{AtrSlMult} \times ATR$. Range 1.0–5.0, step 0.5.
AtrTpMult	3.0	Take-profit distance as a multiple of ATR: $TP = \text{entry} +/- \text{AtrTpMult} \times ATR$. Range 1.0–8.0, step 0.5.
Lots	0.10	Fixed lot size for each new position. Range 0.01–1.0, step 0.05.
Magic	4130	Magic number identifying this EA's positions. Give each chart / instance a unique value so trades are managed independently.

Recommended Settings

The strategy is intentionally built from a few, broadly-ranged parameters to resist curve-fitting, and its noise terms self-scale to volatility — so the defaults are a sound starting point across most instruments and timeframes.

- **Symbols:** Major FX pairs with clean trends and reasonable spread (e.g. EURUSD, USDJPY, GBPUSD, AUDUSD) work well; the ATR-scaling also makes it portable to indices and metals.
- **Timeframe:** H1 to H4 give the filter enough structure to separate trend from noise while keeping trade frequency practical. Lower timeframes are usable but expect more noise and cost sensitivity.
- **Responsiveness:** To make velocity turn faster, raise `ProcNoiseFrac` and/or lower `MeasNoiseFrac`. For steadier, slower signals, do the reverse.

- **Selectivity:** Raise `VelThreshFrac` toward 0.10–0.15 to trade only stronger trends and reduce whipsaw in ranging conditions.

Tip: Because the SL/TP distances scale with ATR, the risk profile stays consistent as volatility changes. Adjust `AtrSlMult` and `AtrTpMult` together to keep a reward-to-risk ratio you are comfortable with (defaults give 1.5 : 1). Always validate any change with a walk-forward test before going live.

How to Install on MetaTrader 5

- 1 Copy `KalmanVelocityTrend.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

Before live trading: Run the EA in the Strategy Tester and on a demo account first. Confirm the filter has warmed up (it needs at least `AtrPeriod + 3` completed bars before it can trade) and that spread, commission, and slippage on your broker are compatible with the ATR-based targets.

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.