



---

## LogikRenko Bar Type An Improvement to the Classic Renko

**U.S. Government Required Disclaimer** - *Forex, futures, stock, and options trading is not appropriate for everyone. There is a substantial risk of loss associated with trading these markets. Losses can and will occur. No system or methodology has ever been developed that can guarantee profits or ensure freedom from losses. No representation or implication is being made that using this methodology or system or the information in this letter will generate profits or ensure freedom from losses.*

## Contents

<i>Software User Agreement</i> .....	3
Overview .....	6
LogikRenko Wick – Displaying True Price Movement.....	8
LogikRenko Wick Benefits .....	9
Reversal Brick OpenToOpen vs. CloseToOpen Implementation.....	12
LogikRenko CloseToOpen Benefits.....	13
Reversal Brick Size – Traditional vs. LogikRenko .....	14
LogikRenko – IntraBar Brick Generation .....	16
Adaptive Brick Size Determination and Implementation.....	17
Current Bar Indicator application - Traditional vs. LogikRenko .....	18
Spurious Noise Filtering – Traditional vs. LogikRenko.....	20
Consistent Reference point choice – Traditional vs. LogikRenko .....	21
Traditional Renko Implementation.....	21
LogikRenko Implementation.....	23
LogikRenko – Customizable Options / Inputs .....	24
Input Examples.....	25

## *Software User Agreement*

IMPORTANT: PLEASE READ THIS AGREEMENT CAREFULLY BEFORE CONTINUING WITH THE INSTALLATION PROCESS OF THE SOFTWARE ("THE SOFTWARE"). BY INSTALLING THE SOFTWARE, YOU AGREE TO BE BOUND BY THE TERMS OF THIS AGREEMENT. IF YOU DO NOT AGREE TO THE TERMS OF THIS AGREEMENT, DO NOT INSTALL THE SOFTWARE.

1. License. If you have purchased a license to the Software, PureLogikTrading grants you a non-exclusive right and license to use the Software and the related documentation (the "Documentation") as set forth in this Agreement. You may use the Software on a single computer. You may also use the Software on a second (portable or home) computer so long as only one copy is used at a time. You may make a single copy of the Software for backup and archival purposes only provided that any copy must contain all proprietary notices included in the original. You may use the Documentation to assist in your use of the Software. If you download the Software without physical delivery of Documentation, you may review the Documentation online, but you may not make further copies of the Documentation. You own the media on which the Software is recorded, but not the Software itself or any copy of it. This license is not a sale of the original or any copy of the Software.
2. Copying, Transferring or Modifying Software. The Software contains copyrighted material, trade secrets and other proprietary intellectual property. You may not permit concurrent use of the Software unless each user has an applicable license. You may not permit other individuals to use the Software except under the terms listed above. For the purposes of protecting PureLogikTrading trade secrets, you may not decompile, disassemble, reverse-engineer or otherwise display the Software in human-readable form. You may not modify, translate, rent, lease, distribute or lend the Software, and you may not sell to others the right to use the Software on your computer. You may not remove any proprietary notices or labels on the Software. You may not copy, transfer, transmit, sublicense or assign this license or the Software except as expressly permitted in this Agreement.
3. Term. If you have purchased a license to the Software, this Agreement and license are effective from the time you accept the terms of this Agreement until this Agreement is terminated. You may terminate this Agreement at any time by destroying all copies of the Software. This Agreement will terminate immediately and without further notice if you fail to comply with any provision of this Agreement. All obligations of confidentiality and restrictions on use, and all other provisions that may reasonably be interpreted to survive termination of this Agreement, will survive termination of this Agreement for any reason. Upon termination, you agree to destroy all copies of the Software. If you have obtained an evaluation copy of the Software, and have not purchased a license to the Software, this Agreement and license are effective for a period of thirty (30) days from the date of installation of the Software.
4. Warranty Disclaimers. THE SOFTWARE AND THE DOCUMENTATION ARE PROVIDED "AS IS" AND WITHOUT WARRANTIES OF ANY KIND, EXPRESS, STATUTORY OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND

PERFORMANCE OF THE SOFTWARE AND THE DOCUMENTATION IS WITH YOU. SHOULD THE SOFTWARE OR THE DOCUMENTATION PROVE DEFECTIVE, YOU (AND NOT PURELOGIKTRADING) ASSUME THE ENTIRE COST OF ALL NECESSARY SERVICING OR REPAIR.

PURELOGIKTRADING DOES NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE SOFTWARE WILL MEET YOUR REQUIREMENTS OR OPERATE IN THE COMBINATION THAT YOU MAY SELECT FOR USE, THAT THE OPERATION OF THE SOFTWARE WILL BE UNINTERRUPTED OR ERROR FREE, OR THAT DEFECTS IN THE SOFTWARE WILL BE CORRECTED. NO ORAL OR WRITTEN STATEMENT BY PURELOGIKTRADING OR BY A REPRESENTATIVE OF PURELOGIKTRADING SHALL CREATE A WARRANTY OR INCREASE THE SCOPE OF THIS WARRANTY.

Notwithstanding the above, you may have certain warranty rights which vary from state to state and which cannot be disclaimed by contract. Any warranties that by law survive the foregoing disclaimers shall terminate ninety (90) days from the date you received the Software as shown by your receipt. Some states do not allow limitations on how long an implied warranty lasts, so the foregoing limitation may not apply to you if prohibited by law.

5. Limitation of Liability. YOUR SOLE REMEDIES AND PURELOGIKTRADING' ENTIRE LIABILITY ARE SET FORTH ABOVE. IN NO EVENT WILL PURELOGIKTRADING BE LIABLE FOR DIRECT, INDIRECT, INCIDENTAL OR CONSEQUENTIAL DAMAGES RESULTING FROM THE USE OF THE SOFTWARE, THE INABILITY TO USE THE SOFTWARE, OR ANY DEFECT IN THE SOFTWARE, INCLUDING ANY LOST PROFITS, EVEN IF THEY HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. IN NO EVENT WILL PURELOGIKTRADING' TOTAL LIABILITY TO YOU FOR ALL DAMAGES, LOSSES, AND CAUSES OF ACTION (WHETHER IN CONTRACT, TORT, INCLUDING NEGLIGENCE, OR OTHERWISE) EXCEED THE AMOUNT YOU PAID FOR THIS PRODUCT. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF RELIEF, INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATION OR EXCLUSION MAY NOT APPLY TO YOU.

You agree that PureLogikTrading will not be liable for defense or indemnity with respect to any claim against you by any third party arising from your possession or use of the Software or the Documentation.

6. Export Control Laws. You agree to comply with all laws, rules and regulations applicable to the export of the Software or the Documentation. Specifically, you shall not export, re-export or transship the Software or the Documentation, or the direct product thereof, in violation of any United States laws and regulations which may from time to time be applicable. None of the Software or underlying information or technology may be downloaded or otherwise exported or re-exported (i) into any country to which the U.S. has embargoed goods; or (ii) to anyone on the U.S. Treasury Department's list of Specially Designated Nationals or the U.S. Commerce Department's Table of Denial Order. By downloading or using the Software, you are agreeing to the foregoing and you are representing and warranting that you are not located in, under the control of, or a national or resident of any such country or on any such list.
7. Government Restricted Rights. The Software has been developed at private expense and is "commercial computer software" or "restricted computer

software" within the meaning of the FARs, the DFARs, and any other similar regulations relating to government acquisition of computer software. Nothing contained herein will be deemed to grant any government agency any license or other rights greater than are mandated by statute or regulation for commercial computer software developed entirely at private expense.

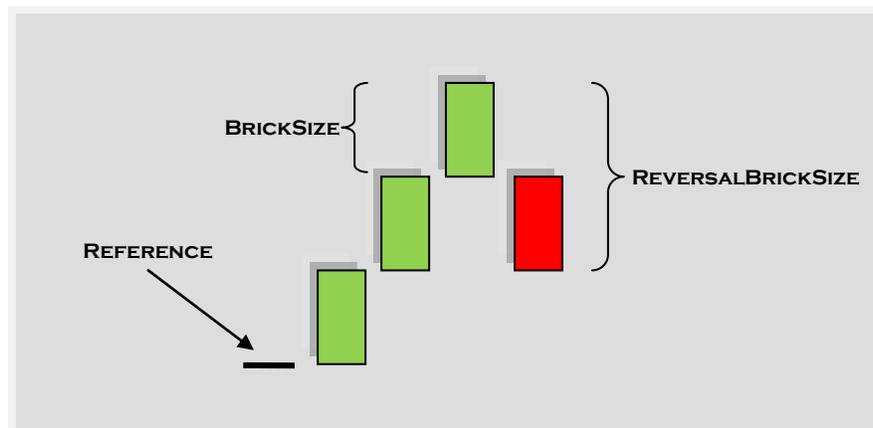
8. Entire Agreement. This Agreement is the complete agreement between PureLogikTrading and you and supersedes all prior agreements, oral or written, with respect to the subject matter hereof. If you have any questions concerning this Agreement, you may write to PureLogikTrading, Customer Service, [purelogiktrading@gmail.com](mailto:purelogiktrading@gmail.com).

# LogikRenko Data Model / Charting Method

## Overview

A Charting Method is the visual representation of a data model, describes how data is represented and accessed. There are many different methods used to exploit different aspects of the data they are applied to. Some of the more traditional charting methods include time, ticks, and volume. While these charting methods are primarily based on time, the Renko charting method excludes time and volume.

The Renko charting method is thought to have acquired its name from *renga*, which is the Japanese word for “brick.” Similar to Kagi and Three-Line-Break charts, a Renko chart plots the first price tick of a data set as its starting reference point. If the price moves up by the predefined amount, an up brick is created. If the price moves down by the predefined amount, a down brick is created. This predefined amount is referred to as the *Brick Size*. Traditionally, the bricks are always equal in size. After this first brick is created, the following brick is created in the same direction if the price moves the *BrickSize* in that direction. However, the price is required to move 2 times the *BrickSize* from the close to create an opposite direction brick. This “reversal amount” multiple is referred to as the *ReversalBrickSize*. The anatomy of this is described below.



**Figure 1 (Basic Renko)**

As show above, the reference starts the position of all future Bricks. The price of the current brick is always compared to the price of the last brick. The importance of this reference point will be examined further on. The chart is then constructed by placing a brick in the next column once the price has surpassed the top or bottom of the previous brick by the *BrickSize* or *ReversalBrickSize* amount. The reversal brick Open price is displayed at the previous bar Open price, although this is not the “true” Open price. This will be explored in more detail later.

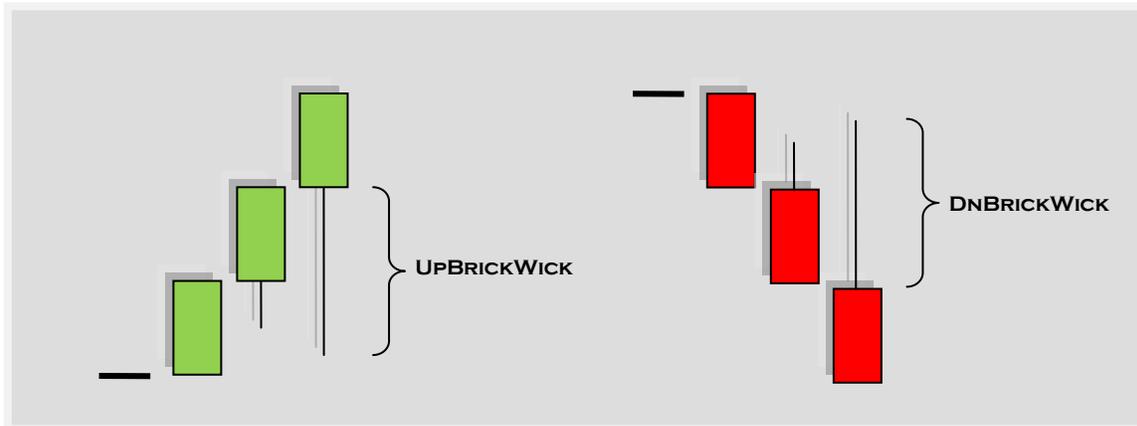
Basic trend reversals are signaled with the emergence of a new green or red brick. A new green brick indicates the beginning of a new uptrend. A new red brick indicates the beginning of a new downtrend. This brick by brick design reflects price movement alone, without time and

volume considerations. It isolates the underlying price trend by filtering out the minor price changes.

While filtering minor price fluctuations is beneficial to determine the underlying trend, it is simultaneously detrimental by eliminating important price information. The LogikRenko can depict the true High / Low with a candle chart like *wick* to reintroduce this important price information, while maintaining the benefit of isolating the underlying price trend.

## LogikRenko Wick – Displaying True Price Movement

Renko charts drop all price movements smaller than the *Brick Size*. A smaller *Brick Size* denotes a higher data granularity and therefore a higher number of bricks. Conversely, a larger *Brick Size* denotes lower data granularity and therefore a lower number of bricks. Although the Renko chart will display the High / Low at the Open / Close, the price may move beyond these points.



**Figure 2 (LogikRenko Wick)**

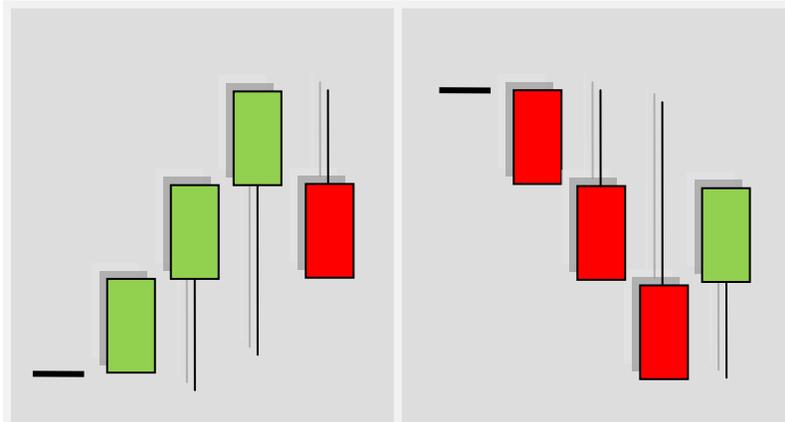
An up brick may have a Low that is as much as twice the *ReversalBrickSize* – 1 Tick in length; a down brick may have a High that is twice the *ReversalBrickSize* – 1 Tick in length. This display method depicts the true price movement in the formation of a new brick.

The LogikRenko is configurable to display either the conventional Brick without the Wick, or display the Brick with the Wick. This is defined later in this paper.

## LogikRenko Wick Benefits

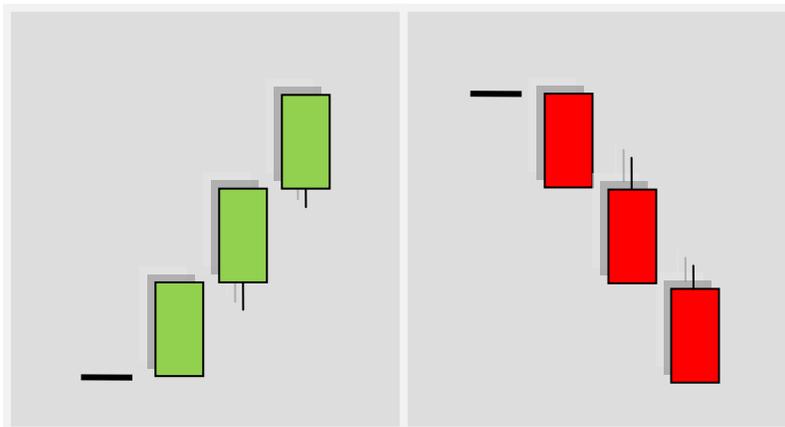
The LogikRenko *wick* display can be helpful in the following aspects (illustrated below)

- Large wick-to-brick ratio illustrates strong indecision / possible reversal



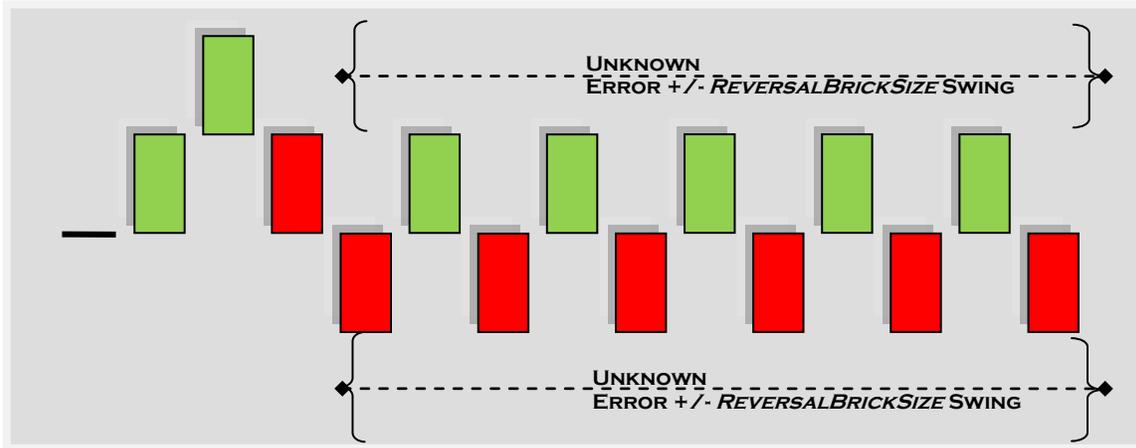
**Figure 3 (Large wick-to-brick) Figure 4 (Large wick-to-brick)**

- Small wick-to-brick ratio illustrates little to no indecision / less likelihood of reversal



**Figure 5 (Small wick-to-brick) Figure 6 (Small wick-to-brick)**

- Congestion areas can be analyzed more efficiently with the true price
  - Renko Congestion lacks true price – indeterminable Support / Resistance region



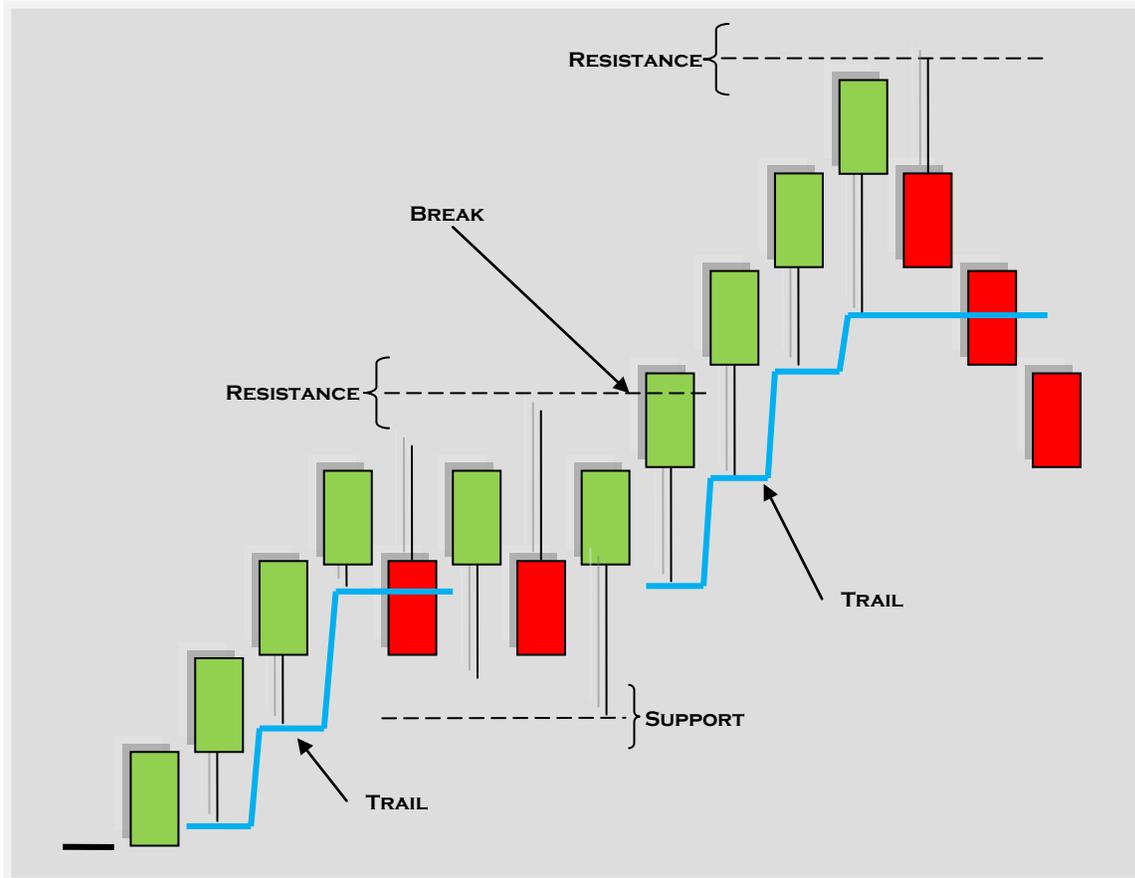
**Figure 7 (Renko Congestion)**

- Congestion areas can be analyzed more efficiently with the true price
  - LogikRenko Congestion includes true price –this provides an identifiable Support / Resistance region or channel



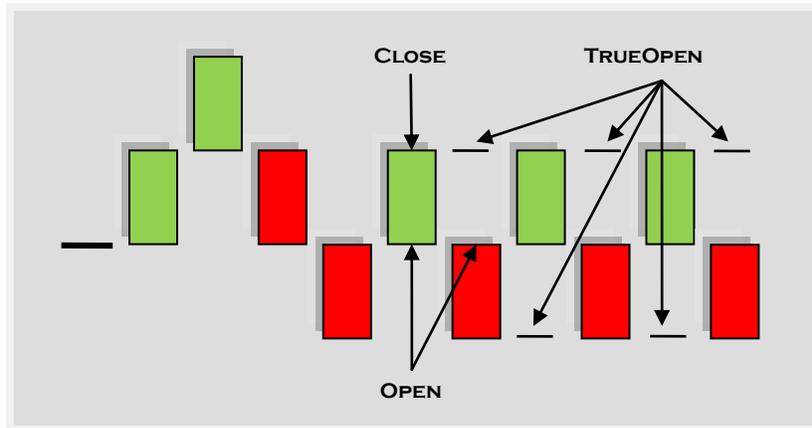
**Figure 8 (LogikRenko Congestion)**

- A wick endpoint can be utilized to determine an adaptive / dynamic trail more easily
- A wick endpoint can be utilized to determine Support / Resistance areas more easily



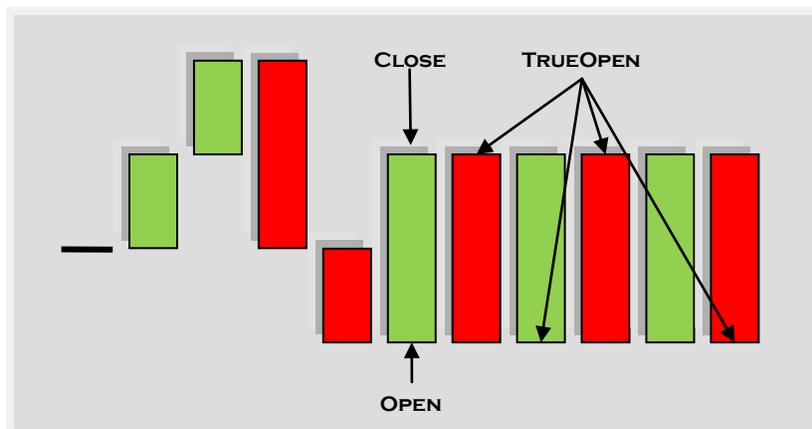
## Reversal Brick OpenToOpen vs. CloseToOpen Implementation

As described earlier, the Traditional Renko will create a reversal brick if the current tick price is two times the *BrickSize* from the Close of the last brick in the opposite direction. The Open price of the reversal brick is traditionally the same as the Open of the previous brick. This is not the TrueOpen price of a reversal brick. Let's define this as the *OpenToOpen* reversal implementation and is denoted below;



When a reversal brick is created, the Open of the reversal bar should point at the previous brick Close. To properly depict this, the *CloseToOpen* implementation sets the Open of the reversal brick to the Close of the previous brick. Although this may not be as visually appealing to some, it represents the **TrueOpen** of the reversal, as well as properly performs a *buy / sell at next bar open* strategy order during a strategy back-test execution. This *CloseToOpen* reversal implementation is denoted below;

Note: The two patterns above and below are identical data with different depictions to enable the user to pick the most appropriate for his/her trading needs.



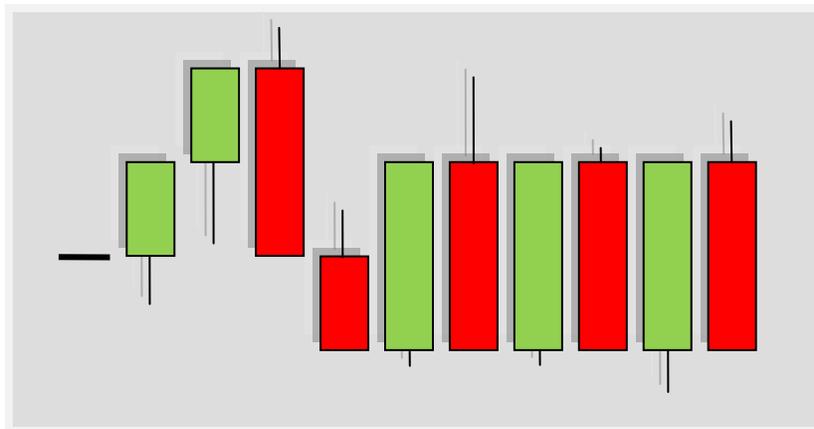
The LogikRenko features the option to use either the *OpenToOpen* implementation, or the *CloseToOpen* implementation to depict a reversal brick.

## LogikRenko CloseToOpen Benefits

The LogikRenko *CloseToOpen* implementation can be beneficial in the following respects

- Clearly identifying strong support
- Clearly identifying strong resistance
- Clearly identifying double tops, as they can display in this manner
- Clearly identifying double bottoms, as they can display in this manner

The LogikRenko offers the ability to utilize the *CloseToOpen* implementation a step further. Enabling the *Wick* and the *CloseToOpen* option, the LogikRenko will expose the visual benefit of displaying the true reversal Open brick price, while simultaneously depicting the true High / Low. This will overlay important price information on the true reversal open brick price, granting the benefit of both implementations. Additionally, wick size is a better representation of the benefits described previously because it doesn't include the *BrickSize* inherent with the *OpenToOpen* implementation. This combination of the *CloseToOpen* reversal implementation and the *Wick* implementation is denoted below;



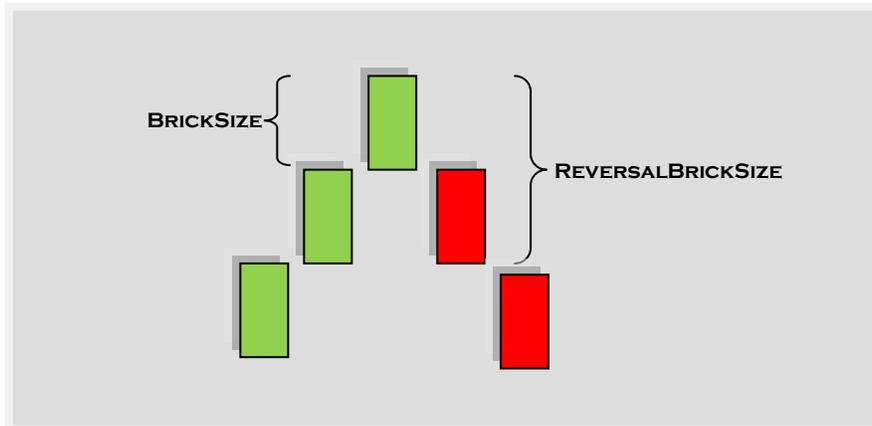
The LogikRenko *CloseToOpen + Wick* implementation *can* be helpful in the following respects

- Clearly identifying strong support with
- Clearly identifying strong resistance
- Clearly identifying double tops, as they can display in this manner
- Clearly identifying double bottoms, as they can display in this manner
- Large wick-to-brick ratio illustrates stronger indecision than using wicks alone
- Congestion areas can be identified and analyzed more efficiently

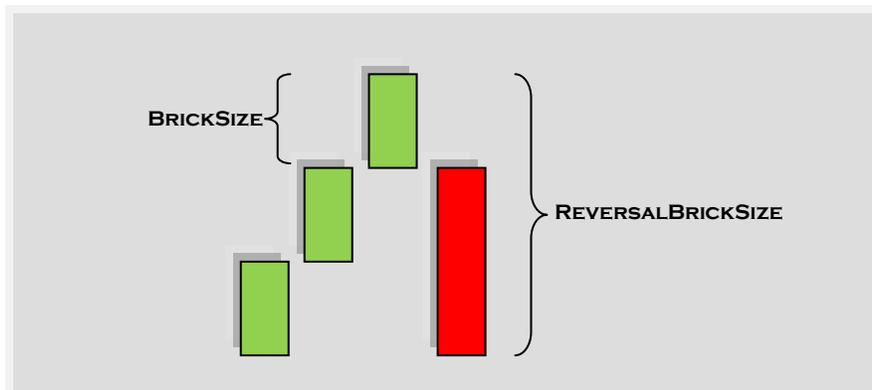
The LogikRenko is configurable to display either the *CloseToOpen* or the *OpenToOpen* Reversal Brick display method. This is defined later in this paper.

## Reversal Brick Size – Traditional vs. LogikRenko

As described earlier, the price of the current tick is always compared to the price of the *close* price of the last brick. A new brick is created in the same direction of the current brick if the tick price moves the *BrickSize* amount in the same direction. However, if the price move (2) times, or the *ReversalBrickSize* amount of the *BrickSize* from the close of the previous brick, an opposite direction (reversal) brick is created. This is illustrated below has the *ReversalBrickSize* set to 2.



A powerful modification allows the trader to more clearly and accurately depict the trend direction and to minimize spurious reversal whipsaws, by allowing for the increase in the *ReversalBrickSize*. This is illustrated below and has the *ReversalBrickSize* set to 3.



The LogikRenko is configurable to use a Reversal Brick Size multiple ranging from 1 to 9. This is defined later in this paper.

The following page displays the GBPUSD on 12/17/2009 EST. The *BrickSize* is 15 pips, with increasing *ReversalBrickSize* on each chart. You will notice immediately that this adaptation to true price changes significantly changes the price bar formation and even the chart interpretation.



**Figure 5 - ReversalBrickSize = 2**



**Figure 6 - ReversalBrickSize = 3**



**Figure 7 - ReversalBrickSize = 4**

## LogikRenko – IntraBar Brick Generation

The traditional Renko algorithm will update on a chart when a new brick is formed by either the current tick price surpassing the top or bottom of the previous brick by the *BrickSize* or *ReversalBrickSize* amount. This End-of-Bar implementation hides all tick by tick price movement, as well as the current tick price of the underlying instrument.

The LogikRenko implementation gives the user the option to enable intrabar brick generation. The algorithm described above to create a new brick remains the same, but the brick is updated visually. The current tick price of the underlying instrument is always displayed, as well as highest / lowest movement within the *BrickSize* / *ReversalBrickSize* permitted range in the form of the wick.

When IntraBar Brick Generation is enabled, the Reversal Brick Display is always set to *CloseToOpen*. As described in the “*Reversal Brick OpenToOpen vs. CloseToOpen Implementation*” section, the TrueOpen price must be used in order to properly draw an intrabar brick. Additionally, as described in the “*LogikRenko Wick – Displaying True Price Movement*” section, the true price movement is also used in order to properly draw an intrabar brick.

## Adaptive Brick Size Determination and Implementation

The LogikRenko has the ability to dynamically determine the appropriate brick size to best adapt to market conditions. The unique algorithm constantly monitors market data on a tick by tick basis to self tune the brick size throughout the day. At the beginning of each new day the computed brick size is automatically implemented on the chart.

In order to enable this feature, the input brick size must be set to zero.

Note :: When this feature is enabled, the first two days loaded on the chart will not be indicative of the proper brick size. The algorithm needs two days of data to adapt itself to the market.

## Current Bar Indicator application - Traditional vs. LogikRenko

A flaw with the traditional Renko and its *OpenToOpen* calculation style is that although the current bar is actually completed, the platform doesn't recognize its completion until the next bars open exists. This means that any indicator applied to the chart will be a bar behind, unless intrabar generation is enabled for the indicator itself. Although this may not be CPU intensive for some indicators, an applied complex indicator or group of indicators will drastically affect the performance of the platform and computer.

The LogikRenko allows any applied indicator to compute as if intrabar generation was enabled on that indicator, while actually only computing once at the end of the bar. This is done by either enabling intrabar brick generation for the LogikRenko chart itself, or by using the *CloseToOpen* method of displaying the Open price of a brick.

When using the Intrabar Brick Generation method or the *CloseToOpen* method, the Open of the next bar is pinned on the chart for the next bar. This allows the platform to believe the bar is completed, thus allowing the applied indicator to reference the true last completed brick. A screenshots below compare the Traditional Renko vs. LogikRenko indicator application.



Figure 8 - Traditional Renko Indicator EOB



Figure 9 - LogikRenko EOB

## Spurious Noise Filtering – Traditional vs. LogikRenko

Depending on the real-time tick filter algorithm implemented by the platform, a spurious tick price may pass through from a broker's data stream. Since the Renko chart algorithm uses pure tick data as it is received from the broker, this noise may create bricks which are inconsistent with the true in a whip saw manner. Although the effect on the price is minimal since the chart price is dependent on the latest tick, any indicator applied to the chart will be useless since this price data is filtered into it. This effect increases as the input size of the brick decreases.

The LogikRenko implements a real-time tick filter independent of any other filter. The algorithm is designed to eliminate this whip saw effect without removing important price information. Spurious ticks are detected, classified, and disregarded.

## Consistent Reference point choice – Traditional vs. LogikRenko

### Traditional Renko Implementation

Renko charts are non-perturbative. Perturbation theory describes how the response of a known system changes in response to small changes in its parameters. In this application, the parameter is the *Reference*. This means that depending on the *Reference* point, the brick anatomy on any particular day may be different. A small change in *Reference* may incur a rather large change on the chart. The proper selection of the *Reference* is critical, and must be done on a day by day basis for chart consistency. Below is an analysis of the Traditional *Reference* selection

A Traditional Renko chart will use the first tick of the dataset as the *Reference*. Assume the following test dataset.

	Chart A	Chart B
Days Loaded	<b>6</b>	<b>5</b>
BrickSize	<b>100 Ticks</b>	<b>100 Ticks</b>
Reference	<b>1.64762</b>	<b>1.64556</b>

This test consists of two charts, A and B with a different number of days loaded. Since each chart has a different number of days loaded, the *Reference* for each chart is different. Let's quantify this difference in terms of the *BrickSize* selected.

Using the *mantissa* of the reference price, we compute the Offset Affect ( $O_a$ ).

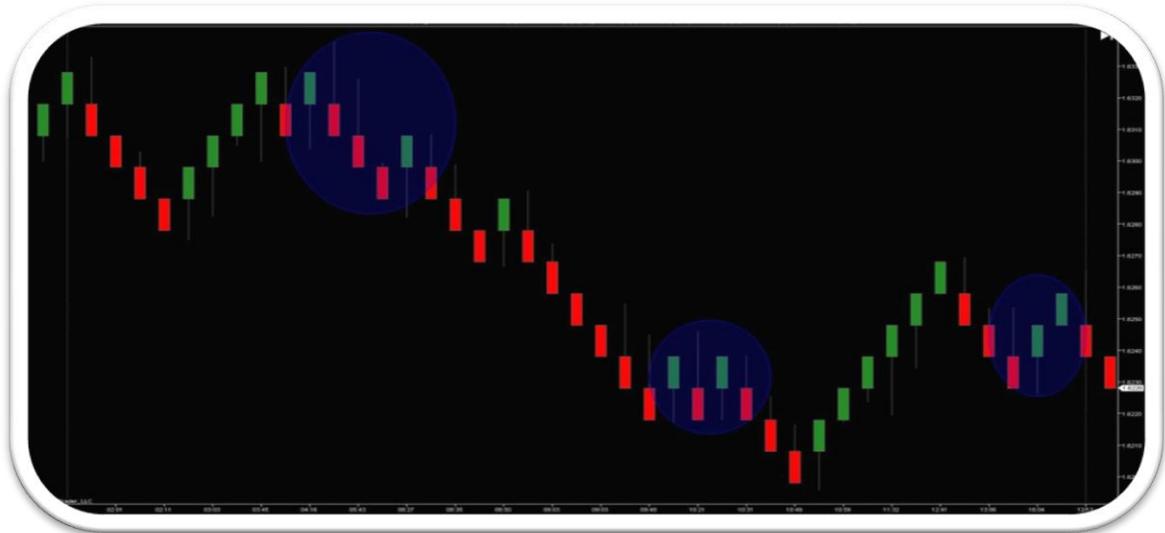
$$= (|man(ref1) \% BrickSize - man(ref2) \% BrickSize|) \quad (1)$$

$$= (|0.64762 \% 100 - 0.64456 \% 100|) \quad (2)$$

$$= 0.00006 \quad (3)$$

$$O_a = 0.6 \text{ Pips} \quad (4)$$

The following two charts illustrate the dataset used in the example. This dataset consists of one day; GBPUSD 12/13/2009. The appropriate areas of concern are highlighted.



**Figure 10 (5 Days)**



**Figure 11 (6 Days)**

The effect of an  $O_a = 0.6$  pips is significant enough to alter the anatomy of the day. Creating a strategy on such loosely defined data is very difficult, if not impossible.

## LogikRenko Implementation

Rather than arbitrarily selecting the first tick of a dataset, the LogikRenko takes a different approach to selecting the *Reference*.

First, in order to determine the best approximation *Reference* the focus is at a time when the market is settled. The market is at its most settled state when the contributing markets are closed; at 12:00AM EST. An initial solution is to use the first new tick after 12:00AM, but again arbitrary. Basing this *Reference* on time alone will alleviate the issue on a single computer, but it may persist on other computers since their clocks differ due to hardware clock drift. This observation alone exploits the non-perturbative effect we are trying to obviate.

Why should time alone be used to determine the *Reference* when the Renko chart in general doesn't utilize time? The LogikRenko uses a unique approach to determine the best *Reference* after 12:00AM. This best approximation is used to minimize the non-perturbative effect of the Traditional Renko, thereby creating a consistent dataset for analysis.

The chart below illustrates the LogikRenko *Reference* selection algorithm. This dataset consists of one day; GBPUSD 12/13/2009. Loading more or less data has perturbative affect on any one day after the first 12:00AM algorithm *Reference* point. The appropriate area of concern is highlighted

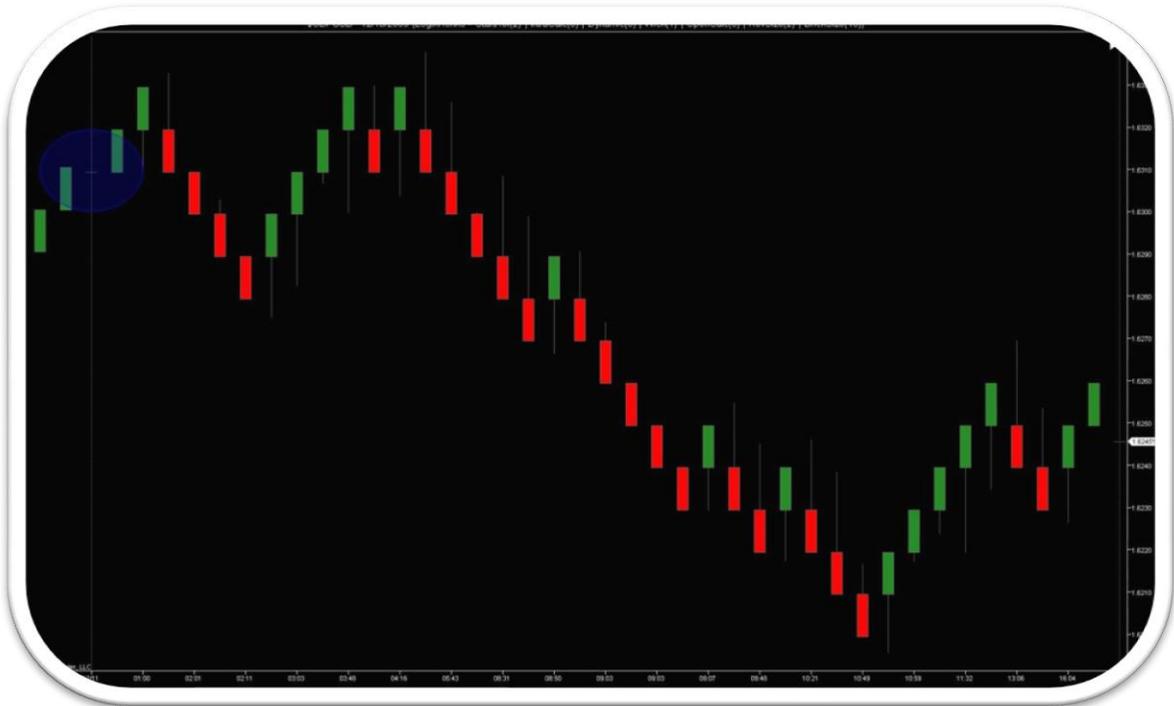


Figure 12 (variable days loaded)

## LogikRenko – Customizable Options / Inputs

The LogikRenko is unlike any other data type in that multiple options may be controlled through a single input string. This is done through a technique called masking. The following attributes of the LogikRenko are customizable:

- Reversal Brick Size (1 – 9)
- TickSize input (0 – 999)
- Wick Display On/Off
- Reversal Brick OpenToOpen / CloseToOpen
- IntraBar Brick Generation On/Off

The **Interval Setting** input field is always made up of six (6) integer values. Each place represents a specific definable input field as described above. These values are represented below

Field	Reversal Brick Size	BrickSize (Ticks)	Wick Display	Reversal Brick Display	IntraBar Generation
Valid Range	(1 – 9)	01-999	0 Off 1 On	0 OpenToOpen 1 CloseToOpen	0 Off 1 On

**All input fields are one (1) integer value with the exception of the Brick Size. The Brick Size input must always be two (2) integer values. Some examples are given below.**

Using the table above we can create any Interval Setting input to configure which options we want to use. If an illegal (less than 6 integer values) Interval Setting is used, a default value is automatically used. The Interval Setting is displayed in its simplified individual field representation at the top of the chart. The following page shows some example configurations with the accompanying Interval Settings.

## Input Examples

Configuration	Integer Value	NT Interval Settings
<ul style="list-style-type: none"> <li>➤ Reversal Brick Size 2</li> <li>➤ Brick Size 5 Pips</li> <li>➤ Wick Display On</li> <li>➤ Reversal Brick Display OpenToOpen</li> <li>➤ IntraBar Generation Off</li> </ul>	<ul style="list-style-type: none"> <li>➤ 2</li> <li>➤ <b>005</b></li> <li>➤ 1</li> <li>➤ 0</li> <li>➤ 0</li> </ul>	205100
<ul style="list-style-type: none"> <li>➤ Reversal Brick Size 2</li> <li>➤ Brick Size 10 Pips</li> <li>➤ Wick Display Off</li> <li>➤ Reversal Brick Display CloseToOpen</li> <li>➤ IntraBar Generation Off</li> </ul>	<ul style="list-style-type: none"> <li>➤ 2</li> <li>➤ <b>010</b></li> <li>➤ 0</li> <li>➤ 1</li> <li>➤ 0</li> </ul>	210010
<ul style="list-style-type: none"> <li>➤ Reversal Brick Size 2</li> <li>➤ Brick Size 10 Pips</li> <li>➤ Wick Display On</li> <li>➤ Reversal Brick Display CloseToOpen</li> <li>➤ IntraBar Generation On</li> </ul>	<ul style="list-style-type: none"> <li>➤ 2</li> <li>➤ <b>010</b></li> <li>➤ 1</li> <li>➤ 1</li> <li>➤ 1</li> </ul>	210111
<p><b>Adaptive BrickSize</b></p> <ul style="list-style-type: none"> <li>➤ Reversal Brick Size 2</li> <li>➤ Brick Size auto detect</li> <li>➤ Wick Display On</li> <li>➤ Reversal Brick Display CloseToOpen</li> <li>➤ IntraBar Generation On</li> </ul>	<ul style="list-style-type: none"> <li>➤ 2</li> <li>➤ <b>000</b></li> <li>➤ 1</li> <li>➤ 1</li> <li>➤ 1</li> </ul>	200111

Note: When IntraBar Brick Generation is enabled, the following inputs are always as listed below regardless of their input values.

- Wick Display → On
- Reversal Brick Display → CloseToOpen