Increments

## Overview

The program is to have active and inactive periods through the week. When active, it is to open trades based off either a fixed Buy / Sell instruction, or comparison between a moving average and the current mid-price.

The take profit, stop loss, and stake per point are all determined by a value called the “increment” which cross references a string of comma separated values given by the user. For example if the increment = 3, the third value in each string will be used for the next trade.

Once a trade is opened, the program is to track the market mid-price, closing the trade when it goes beyond the take profit or stop loss.

On close various totals are updated, including the increment, which depending on inputs will increase, decrease, or stay the same.

I would like the source code on completion of the project.

## Glossary

Program variables are underlined. User input variables are in **bold**.

### Internal values

|  |  |  |
| --- | --- | --- |
| ***Name*** | ***Range of values*** | ***Description*** |
| Session | open / closed | A value recording if new trades are allowed to be opened |
| Session balance | -100000.0 - +100000.0 | A cumulative total of points won / lost in this session, affected by the stake |
| Increment | -1000.0 - +1000.0 | A decimal that is modified when a trade exits |
| Rounded Increment | -1000 - +1000 | The Increment rounded to the nearest integer, this number will say what the next trades stake, Take profit and Stop loss will be |
| Moving average | -100000.0 - +100000.0 | A simple, smoothed, linear weighted, EMA, DEMA or TEMA value |
| MA Upper boundary | -100000.0 - +100000.0 | The Moving average + **MA Differential** |
| MA Lower boundary | -100000.0 - +100000.0 | The Moving average - **MA Differential** |
| Trade | active / inactive | A value recording if a trade is already open |
| Trade value | -100000.0 - +100000.0 | The [pips won/lost] \* [stake] of the most recent trade |

### User inputs (in desired order)

|  |  |  |  |
| --- | --- | --- | --- |
| ***Name*** | ***Possible values*** | ***Description*** | ***Desired default*** |
| **Trade direction** | Buy / Sell / Converge / Diverge | An instruction to buy, sell, buy/sell toward the Moving average, or buy/sell away from the Moving average | Buy |
| **Stake Sequence** | A string e.g. “0.01,0.02,0.04,0.08,0.08,0.08” | A series of comma separated values, in order from first to last, denoting the stake per point of a trade | 0.01,0.02,0.04,0.08,0.16,0.32,0.64 |
| **TP Sequence** | A string e.g. “200,200,200,300,300,400” | A series of comma separated values, in order from first to last, denoting the Take profit of a trade | 500,500,500,500,500,500,500 |
| **SL Sequence** | A string e.g. “100,100,100,200,200,300” | A series of comma separated values, in order from first to last, denoting the Stop loss of a trade | 500,500,500,500,500,500,500 |
| **Maximum increment** | 1 – 100 | An integer which, if the Rounded increment exceeds, causes the **End of sequence action** to trigger | 5 |
| **End of sequence action** | Restart / Hold / Exit | An instruction on what to do when the Rounded increment becomes greater than the **Maximum increment** | Restart |
| **MA Type** | Simple / EMA / DEMA / TEMA / Linear weighted / Smoothed | The method to calculate the Moving average. Definitions for these types can be found in investopedia.com, and are available as indicators included in the base MLQ5 software | Simple |
| **MA Length** | 1 – 1000 | How many candles of the current chart are to be used in calculating the Moving average | 50 |
| **MA differential** | 0 – 10000 | A distance in pips to be added to the Moving average defining the MA Upper and MA Lower boundary | 0 |
| **Differential qualifier** | Beyond / Within | Whether trades can only be opened when the current price is outside the MA Upper boundary and MA Lower boundary, or between them | Beyond |
| **Win increment multiplier** | -100.0 – +100.0 | When a trade exits profitably, multiply the increment by this number | 1 |
| **Win increment addition** | -100.0 – +100.0 | When a trade exits profitably, add this to the increment number | 0 |
| **Loss increment multiplier** | -100.0 – +100.0 | When a trade exits in loss, multiply the increment by this number | 1 |
| **Loss increment addition** | -100.0 – +100.0 | When a trade exits in loss, add this to the increment number | 1 |
| **Open trades on** | Tick / Candle | An instruction to check to open new trades each tick or on the open of a new candle on the chart the program is attached to | Tick |
| **Max spread** | 0 – 1000 | An integer saying the maximum acceptable spread beyond which opening new trades is to be halted | 30 |
| **Max session balance** | 0 – 100000 | A number of points in profit which, if exceeded, causes the close of the session | 100000 |
| **Min session balance** | -100000 – 0 | A number of points in loss which, if exceeded, causes the close of the session | 100000 |
| **Daily session open time** | 0 – 23 | An hour each day when the Session should be set to open. | 0 |
| **Daily session close time** | 0 – 24 | An hour each day when the Session should be set to closed. Please note though 24:00 is not a time, it is intentionally included so that it is possible for the session not to be closed in this way. | 24 |
| **Trade on Monday** | Yes / No | Whether sessions can be opened on Mondays | Yes |
| **Trade on Tuesday** | Yes / No | Whether sessions can be opened on Tuesdays | Yes |
| **Trade on Wednesday** | Yes / No | Whether sessions can be opened on Wednesdays | Yes |
| **Trade on Thursday** | Yes / No | Whether sessions can be opened on Thursdays | Yes |
| **Trade on Friday** | Yes / No | Whether sessions can be opened on Fridays | Yes |
| **Initial session status** | Open / closed | Whether on initialisation, the Session should be set as open (allowing trading immediately) or closed. | closed |

## Key points of function

I am open to different ways of achieving the following, But the following is the key requirements of functioning. Proposed system variables are Underlined. User inputs are in **Bold**.

### Session management

|  |  |
| --- | --- |
| 1.1 | When the program is first loaded onto a chart, if **Initial session status** is set as “open”, the Session must be set to open so the program is able to look for valid entries immediately after initialisation. |
| 1.2 | When the hour of the day = **Daily session open time**, a new Session is opened if the day is valid as per **Trade on [Mon/Tue/Wed/Thur/Fri].** |
| 1.3 | When the hour of the day = **Daily session close time**, an open Session is closed irrespective of if the day is valid as per **Trade on [Mon/Tue/Wed/Thur/Fri].** |
| 1.4 | If the **Daily session close time** = the **Daily session open time**, the Session is closed then immediately reopened, refreshing the Session balance and resetting the Increment. |
| 1.5 | Daily session open and close functions happen independently of the time frame of the chart the program is attached to. So if the chart is H8, and the open time is 4am, the program will begin looking to open trades at 4am, not 8am. |
| 1.6 | When a Session is opened / refreshed, the Session balance and Increment must be reset. |
| 1.7 | If the **Daily session close time** = 24, a number that cannot be reached, then once a Session is opened it cannot be closed by the **Daily session close time**, and this is intentional. |
| 1.8 | The close of a Session does not close an active trade. Trades must exit via Take, Stop, or manual cancellation. |

### Calculating the Moving average

|  |  |
| --- | --- |
| 2.1 | The Moving average, MA Upper, and MA Lower boundaries should be updated when looking to open a trade. |
| 2.2 | For efficiencies sake, there is no need to update them while a trade is active, while the session is closed, or multiple times per candle. |
| 2.3 | The Moving average uses the time period and price data of the chart the program is attached to. |
| 2.4 | The Moving average always uses the Close values of candles. |
| 2.5 | Whatever kind of Moving average is selected,  the MA Upper boundary = the Moving average + **MA Differential** (in points)  the MA Lower boundary = the Moving average - **MA Differential** (in points) |

### Checking to opening a trade

|  |  |
| --- | --- |
| 3.1 | If the Session is open and there is no trade active, the program is to check to open a trade  If **Open trades on** = “Tick” then this must be checked every tick  If **Open trades on** = “Candle” then this must be checked at the start of every candle, including the candle the Session opens on.  If both criteria below are met, a new trade should be opened immediately / as soon as possible |
| 3.2 | If the spread is above **Max spread**, no trades should be opened (but can be closed). |
| 3.3 | If **Differential qualifier** = “Beyond”, then the current mid-price must be equal to or above the MA Upper boundary, *or,* equal to or below the MA Lower boundary to be valid. Therefore if the **MA Differential** = 0, the program will always be able to open. |
| 3.4 | If **Differential qualifier** = “Within”, then for the current price to be valid, the current mid-price must be below the MA Upper boundary, *and,* above the MA Lower boundary to be valid. Therefore if the **MA Differential** = 0, the program will never be able to open. |

### Opening a trade

|  |  |
| --- | --- |
| 4.1 | Whether the trade is to buy or sell depends on **Trade direction**:  If “Buy”, make a buy order  If “Sell”, make a sell order  If “Converge”, buy/sell toward the Moving average:   * So if the Moving average is below the current mid-price, sell. * If the Moving average is above the current mid-price, buy. * In the unlikely event the current mid-point = the Moving average, then do not open a trade.   If “Diverge”, buy/sell away from the Moving average:   * So if the Moving average is below the current mid-price, buy. * If the Moving average is above the current mid-price, sell. * As before, if the current mid-price = the Moving average, then do not open a trade. |
| 4.2 | The stake per point for the trade depends on the Increment rounded to the nearest integer and the **Stake Sequence**. The rounded Increment is the position of the value in the **Stake Sequence** to use.  This is to be read from left to right, with the first comma separated value as 1.  So for example if the **Stake Sequence** = “0.01,0.02,0.04,0.08,0.16,0.32” and the Increment = 2.5, this rounds to 3, so the stake should be the third value in the sequence (0.04). |
| 4.3 | The Take profit for the trade depends on the Increment rounded to the nearest integer.  This time using the **TP Sequence**, the rounded Increment is the position of the value in the **TP Sequence** to use.  All values are pips to be added to the current mid-price when buying, and subtracted from the mid-price when selling (as you would expect).  Again reading from left to right, with the first comma separated value as 1.  So for example if the **TP Sequence** = “500,500,500,400,300,200” and the Increment = 2.5, this rounds to 3, and the third value (500) should be used. |
| 4.4 | The Stop loss for the trade is exactly the same as the Take profit, except it of course compares the Increment with the **SL Sequence**, and the pips value are to be subtracted from the mid-price when buying, and added when selling. |
| 4.5 | The program should only be able to have one trade at a time, so if one is successfully opened, the program must not open any others until it has closed it. |

### Closing a trade

|  |  |
| --- | --- |
| 5.1 | While a trade is active, the program should look to close it each time the market mid-point price moves (though it may be easier to just check every tick).  This should carry on until the trade exits, and should not be stopped even if the Session closes or the spread exceeds **Max spread**. |
| 5.2 | As soon as the current mid-price meets or moves beyond either the Take profit price or the Stop loss price, the trade is to be closed. |
| 5.3 | Once done, the program will have won or lost a certain number of pips, this number is to be multiplied by the stake, and then added to the Session balance. If this causes the Session balance to be greater than **Max session balance** or lower than **Min session balance**, the Session should immediately be closed. |
| 5.4 | If the trade closed in profit the Increment must first be multiplied by the **Win increment multiplier**, then have the **Win increment addition** added to it. |
| 5.5 | If the trade closed in loss the Increment must first be multiplied by the **Loss increment multiplier**, then have the **Loss increment addition** added to it. |
| 5.6 | After the Increment has been modified by the outcome, if the closest integer it would round to is 0 or less, then the Increment must be set to 1 instead.  If when rounded, the closest integer would be greater than the **Maximum increment**, the **End of sequence action** must now be triggered. |
| 5.7 | When the **End of Sequence action** is triggered:  If it is “Repeat”, set the Increment to 1.  If it is “Hold”, set the Increment to the **Maximum increment**.  If it is “Exit”, close the Session. |

## Clarifications

* The “Maximum increment” will always be equal to, or less than, the number of values in the three sequences.
* If a trade is closed manually, the program should perform the same process as if it had closed via Take or Stop.
* Any mention of price in this document refers to the market mid-price, not the actual buy / sell prices.
* Trades can close any time, but can only be opened when the session is open.