Bollinger Bands Mean Reversion Grid

luximil

February 2021

Disclaimer

The strategy described in this document and its related program are intended for educational purposes only. Nothing in this document should be consider investment advice. Moreover, the program created to test the strategy should not be used to trade real money. Trading entails risks, and you could lose all your stake. Consult with a professional to assess your risk profile before trading or investing any money.

Contents

1 Abstract

The main goal of the strategy is to trade the price's reversion to a moving average when it surpasses one of the outer bands of the Bollinger Bands indicator. To do that a set of positions is entered, and they are closed on the price reaching an intermediate level between the outer band and the moving average.

The strategy is intended to trade FX pairs constituted by currencies from countries/economies that are are tightly related to each other or that are driven by the same or very similar factors and risk drivers. Some examples are the Euro and the Swiss Franc, the Australian Dollar and the New Zealand Dollar or the Euro and the British Pound.

2 Limitations

Before starting the development of a cTrader robot to test the strategy, we already acknowledge the limitations of the strategy. We know that prices are not normally distributed, and that returns distributions have fat tails. This reduces the effectiveness of betting on mean reversion based on the difference between the current price and its moving average with respect to standard deviations.

We also recognise the limitation of not being able to trade the moving average. In an ideal environment, we would enter a position for the moving average in the opposite direction to the position entered for the current price. Those 2 position would then be closed on convergence between the current price and its moving average. Because we cannot trade the moving average, it is possible that mean reversion occur only because the moving average moves to close the gap between it and the current price. In that case, our position would close flat or even in a loss, although the price converged to its moving average.

These two facts have to be taken into account when optimising the strategy. Moreover, this strategy will work best when prices move laterally and the moving average stays relatively flat. This optimal market regime is the one we would like to deploy this strategy at. For trending markets, especially the ones with high one-sided volatility, this strategy should be put to rest.

3 Alpha Model

On an entry signal, a set of positions (grid) is entered, such that the distance between the current price and the moving average is divided into equal parts. For each price level between the current price and the moving average a positions is entered with that price level as take-profit.

A long grid is entered when,

- the current price crosses below the lower Bollinger Band, and the parameter *Enter Positions after Confirmation* is deactivated.
- the current candle closes below the lower Bollinger Band, and the parameter *Enter Positions after Confirmation* is activated.

A short grid is entered when,

- the current price crosses above the upper Bollinger Band, and the parameter *Enter Positions after Confirmation* is deactivated.
- the current candle closes above the upper Bollinger Band, and the parameter *Enter Positions after Confirmation* is activated.

All positions are entered with a fixed initial stop-loss set in pips.

Positions are closed when the price reach their stop-loss or take-profit level.

4 Risk Management

To control risk, several features have been included.

4.1 Transaction Costs

When entering a grid, only the positions with take-profit levels that would result in a net profit when reached are entered. Here the fixed round lot commissions and the current spread are considered. The current spread is used as best approximation for the spread at the moment of closing, so as to simplify the technical implementation of the position closing on the price reaching the take-profit level.

Moreover, a parameter (*Min. Net Initial TP in Pips*) has been implemented, so that only positions with a take-profit level that implies a net profit greater than the value of the parameter are entered. This is to avoid entering positions that do not correspond to the risk/reward profile of the investor. For example, to avoid entering positions with a net initial take-profit in pips lower than the initial stop-loss in pips.

4.2 Position Sizing

The position size is estimated using Kelly's formula separately for each level. The Kelly ratio for grid position level i is given by the following

$$K_i := \min\left\{W_i - \frac{(1-W_i)}{R_i}, K_{max}\right\}$$

$$\text{where } W_i := \frac{\text{Number of level } i \text{ closed trades with a net win}}{\text{Number of level } i \text{ trades closed with a net loss}}$$

$$R_i := \frac{\text{Total net profit of level } i \text{ closed trades with a net win}}{|\text{Total net profit of level } i \text{ closed trades with a net loss}|}$$

Positions closed with net profit equal 0 are considered as trades closed with a net loss.

In case the Kelly ratio cannot be computed or is less or equal to 0, K_i is set to $K_{default}$.

To avoid using K_i values that are too extreme for some investors, the K_i is capped by a maximum value K_{max}. Positions at level i are then entered with position size vol_i defined as

$$vol_i := \max\{K_i * AFM, vol_{min}\}$$

where AFM is the current account free margin and vol_{min} the minimum volume in units that can be traded for the instrument.

4.3 Dynamic Stop-loss

A variable stop-loss setting feature is introduced to update the stop-loss of the still opened positions in a grid when positions at lower levels are closed after reaching their take-profit level.

When the first position of a grid is closed, the stop-loss of the remaining positions in the grid is updated to net break-even, such that the fixed round lot commission and spread are taken into consideration. Here the current spread at the closing of the first positions is used as best approximation for the future spread so as to simplify the implementation.

After the first position of a grid is closed, the stop-loss of the remaining positions in the grid is updated on closing of the subsequent positions at their take-profit level. The new stop-loss for the remaining positions is, in this case, set to the take-profit level of the positions closed before the last closed one.

4.4 Dynamic Take-profit

Because the moving average can move against the direction of the grids entered using this strategy, a dynamic take-profit feature is implemented. This allows to close positions on updated price levels based on the current moving average and outer band levels.

On every value change of the moving average or outer bands, the take-profit levels are computed again as described in section 3.

5 Parameters

The default parameters are not set to any economically reasoned values. The default values for the indicator are set to the default values provided by the cTrader platform when adding the Bollinger Bands indicator to a chart. The other default values are set to random values, which however allow the bot to work correctly. For example, the *Risk Management - Max. Account Risk % per Grid* parameter is set to a value greater or equal the *Risk Management - Initial Account Risk %* parameter value.

5.1 Robot Id

This parameter is used for the bot to identify the positions it has created, and that it has to manage. The value of this parameter must be unique for each instance of the bot to avoid interference between different bots.

5.2 Bollinger Bands - Source

This parameters allows the user to choose what price time-series to use for the Bollinger Bands computation.

5.3 Bollinger Bands - Periods

This parameter sets the number of look-back periods (candles) used for the Bollinger Bands computation.

5.4 Bollinger Bands - Standard Deviations

This parameter sets the standard deviations multiplier used for the Bollinger Bands outer bands computation.

5.5 Bollinger Bands - MA Type

This parameter sets the moving average type to use for the Bollinger Bands computation.

5.6 Bollinger Bands - Shift

This parameter allows to use shifted Bollinger Bands values for trading. When set to 0, the latest Bollinger Bands values are used to find entry signals. When set to a value $x > 0$, the Bollinger Bands values computed x bars ago are used to find entry signals in the current bar.

5.7 Grid - Level Count

This parameter sets the amount of levels that each grid should contain. When an entry signal is given, grids are entered with so many positions as specified by this parameter (assuming that their corresponding take-profit levels can result in a net profit. See subsection 4.1)

5.8 Risk Management - Round Lot Commission

This parameter is used to consider the fixed transactions costs per lot.

5.9 Risk Management - Initial Account Risk %

This parameter sets $K_{default}$ in subsection 4.2.

5.10 Risk Management - Max. Account Risk % per Grid

This parameter divided by the number of grid levels sets K_{max} in subsection 4.2.

5.11 Risk Management - Initial SL in Pips

This parameter sets the initial stop-loss level in pips distance from the current price for positions on their execution.

5.12 Risk Management - Min. Net Initial TP in Pips

This parameter sets the minimum potential net profit in pips for entering a positions. See subsection 4.1.

5.13 Risk Management - Enter Positions after Confirmation

This parameter limits the bot to enter grids when a bar closes outside the outer bands of the Bollinger Bands. See section 3.

5.14 Risk Management - Activate Dynamic SL

This parameter activates the variable stop-loss feature described in subsection 4.3.

5.15 Risk Management - Activate Dynamic TP

This parameter activates the variable take-profit feature described in subsection 4.4.

6 Optimisation

For the optimisation process, we decided to select a period of 1 year to test different parameter sets, and find the best ones. The optimal parameter sets are then tested "out-of-sample" in the period starting from the end of the 1-year period used to find them and ending at end of the last month for which data is available.

We used the genetic optimisation method that is available on the cTrader platform, even though we acknowledge that genetic optimisation methods tend to generate optimal parameters sets that are very similar. The initial account size was set to 100000 €, the round lot commission per million to 30€, and tick data was used for the optimisation as well as for the subsequent backtests.

The first idea was to optimise the Sharpe ratio. However, this led to optimal parameter sets that generated a very reduced number of trades and extreme Sharpe ratios (even above 10). In a second attempt, the optimisation criteria were extended, and parameter sets were optimised regarding the Sharpe ratio and the total number of trades. However, this still led to extreme overfitting. Optimal parameter sets were found that resulted in Sharpe ratios still above 10 and only a handful of total trades. Therefore, a custom *GetFitness* method was added to the bot, such that the target function to optimise was redefined. Because this bot is meant to trade mean-reversion, the *GetFitness* method was constructed to optimise the parameters such that the resulting returns mimic the typical distribution of a mean-reversion-strategy. That typical distribution is characterised by having a relatively low but positive expected value, and the mass of the distribution very concentrated around the expect value. The *GetFitness* method is defined as follows:

$$Fitness := \ln\left(\text{Total Number of Trades}\right) * \frac{\text{Number of Winning Trades}}{\text{Number of Losing Trades}} * \frac{\mu_{\text{geo}}}{\sigma_{\text{geo}}}$$

where ln is the natural logarithm, μ_{geo} the return's geometric mean and σ_{geo} the standard deviation of returns with respect to their geometric mean μ_{geo}.

Because of the limited computational resources available to the author, the algorithm optimisation is reduced to a single FX pair in 1 time-frame. The chosen FX pair is EURCHF in 1-hour timeframe.

6.1 EURCHF H1

The 1-year period starting date was chosen as the first day, one week after the SNB lifted the EURCHF floor: 1st February 2015 to 1st February 2016. The reason for this is to optimise the parameters in a period of time without such a strong central bank intervention, more similar to the current environment. Moreover, we would like to have an extended period of time outside the optimisation period in order to have a more statistically significant out-of-sample backtest with optimal parameters afterwards. Therefore, the starting date was chosen neither later nor earlier.

The following parameters were optimised.

Parameter	**Min. Value**	**Max.Value**	**Step**
BB - Periods	6	30	1
BB - Standard Deviations	1.5	3	0.5
BB - MA Type	Simple, Exponential		
Grid - Level Count	1	5	1
RM - Initial SL in Pips	10	100	10
RM - Min. Net Initial TP in Pips	10	200	10
RM - Enter Positions after Conf.	Yes, No		
RM - Activate Dynamic SL	Yes, No		

Table 1: Optimised Parameters

The other parameters were left fixed for the optimisation process. The round lot commission parameter was set to match the value as in the optimisation feature. The account risk parameters were also fixed. Moreover, the export results feature was deactivated.

Parameter	**Value**
BB - Source	Close
BB - Shift	0
RM - Round Lot Commission	3
RM - Initial Account Risk %	5
RM - Max. Account Risk % per Grid	30
RM - Activate Dynamic TP	Yes

Table 2: Fixed Parameters

We saved the 20 parameter sets with the highest fitness. You can find these parameter sets in the folder *Optimal Parameters*
EURCHF H1
in the repository. The parameter sets file names match their fitness in the optimisation sample according to the *GetFitness* method defined above.

6.1.1 Out-of-Sample Results

Again, because of the limited computational resources available to the author, we randomly selected a single parameter set out of the 20 best to backtest it from 2nd February 2016 to 28th February 2021. The selected parameter set was the third best, i.e. the one named *Fitness 0.58.cbotset*. The results of the backtest are presented below.

From the equity chart, we can see that the account balance was volatile during the backtesting period.

In the 5 years and one month time that lasted the backtesting period, the strategy achieved a low net profit of 520.90 €, which amounts to roughly a 0.5% net return. This is a very low return, and the fact that the account balance was so volatile during the backtesting period suggests that increasing leverage is not

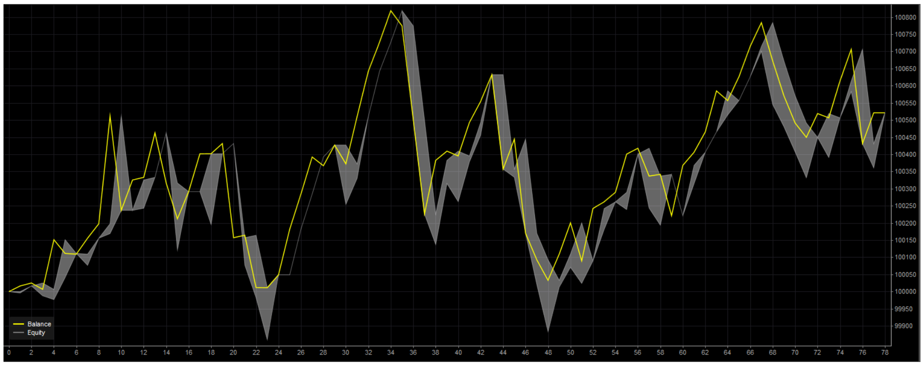

Figure 1: Equity Chart

Net Profit	Starting Capital	Ending Balance	Ending Equity
520.90	100 000.00	100 520.90	100 520.90

Table 3: cTrader Backtesting Account Results

a possibility to increase the net return.

Summary	All Trades	Long Trades	Short Trades
Net Profit	520.90	323.27	197.63
Profit Factor	1.16	1.16	1.15
Commission	-134.10	-77.70	-56.40
Max. Balance Drawdown	0.78%	0.97%	0.42%
Max. Equity Drawdown	0.93	-	-
Total Trades	77	44	33
Winning Trades	48	27	21
Max. Cons. Winning Trades	5	5	5
Largest Winning Trade	314.35	156.41	314.35
Losing Trades	29	17	12
Max. Cons. Losing Trades	4	3	3
Largest Losing Trade	-275.47	-275.47	-274.38
Average Trade	6.76	7.35	5.99
Sharpe Ratio	0.06	0.06	0.05
Sortino Ratio	0.07	0.07	0.07

Table 4: cTrader Trade Statistics

The trade statistics do not show any remarkable differences between long and short trades, which is in line with our expectations. The Sharpe ratio as well as the Sortino ratio are very low, which supports our idea that increasing leverage is not a suitable method to increase net returns.

We have just presented backtesting result's statistics related to absolute returns. Now we analyse relative return's statistics.

Min.	**Max.**	**Mean**	**Geo. Mean**	**St. Dev.**	**Fitness**
-0.918233%	1.047833%	0.023957%	0.006748%	0.409135%	0.044911

Table 5: Return's Statistics

The returns produced by the selected optimal parameter set has a fitness statistic significantly lower than its in-sample fitness (in the optimisation period). Although the standard deviation appears to be low, the arithmetic as well as the geometric mean are even considerably lower.

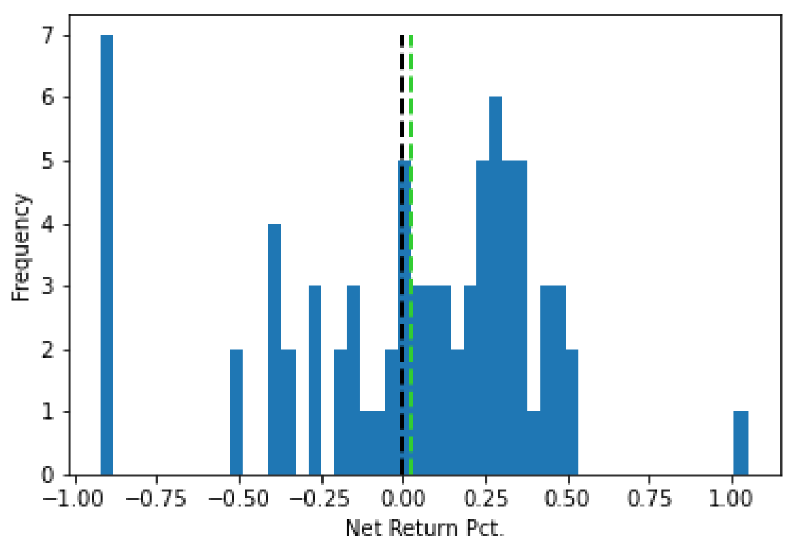

Figure 2: Backtesting Trades Return's Distribution

The dashed black lines marks the 0% return level and the dashed green line the arithmetic mean return. The arithmetic mean return is very near 0. The return's distribution has a peak around -0.9%, probably caused by positions that get stopped out before their stop-loss can be moved. A second peak is observed around the 0.3% level. The return's distribution fails to match our objective of a distribution with low variance with a peak around its expected value. At this point, we acknowledge, that the peak on the left hand side of the distribution is normal, given that it is caused by position that failed to turn a profit or even break-even. This is something difficult to avoid other than by increasing the accuracy of the entry signals.

We can conclude that the optimal parameter analysed is not one that we would consider for trading this strategy. The out-of-sample results do not match the expectations of the in-sample results at all. The parameter set can be considered an overfitted one. We encourage others to try optimising the strategy in a different period and/or test other in-sample-optimal parameter sets.

7 Further Development

7.1 Trend Filter

Because it is impossible to trade the MA, as explained in 2, this bot performs best when the MA remains at a stable level. That means, that this bot will preform best when markets are not trending. Therefore, adding some filter to identify trending market regimes and limit trading during them is a potential improvement for the strategy. This could increase the accuracy of the entry signals.

7.2 Kurtosis Optimisation

In the custom optimisation function, we used the standard deviation of returns as measure for how spread the returns are around their geometric mean. In order to find optimal parameter sets that generate returns with positive mean and very condensed around a distribution peak at its mean, we could instead compute the kurtosis of returns. Kurtosis "punishes" outliers that lie "far" from the mean stronger than standard deviation.