**Stochastic Oscillator Algorithm**

Quant:

Desk: Jerry Cao, Louis Gabison

Backtester:

Implementer:

**Introduction**:

Relative strength strategies use the characteristics of the rest of the market or asset class to analyze a single stock—and we will first implement stochastic oscillator. This strategy is designed to correspond with the over/under valuation of an asset - usually a high stochastic reading would indicate a sell, and vice versa. We have created a property self.cutoff=(0.2, 0.8), where if %K drops below the first value we buy, and if %K goes above the second value we sell.

The stochastic oscillator value is calculated as follows:

Text

Description automatically generated

**Inputs:**

Here’s an example input data dictionary to the algorithm:

stock\_data\_day = {

"Ticker": "APPL",

"Opening Price": 100,

"Closing Price": 200,

"Lowest Price": 50,

"Highest Price": 300,

"Volume": 100000000

}

stocksdata = {

"APPL": stock\_data\_day,

"FISV": stock\_data\_day,

#......#

}

* For now, we are only using daily stock data, namely “opening price”, “closing price”, “lowest price”, and “highest price” since calculating L14 and H14 every tick might not bring substantial improvement in performance despite requiring much more computation.

**Outputs**:

* Upon calling “update\_all(self, new\_stocksdata)”, a dictionary is returned where key is the stock ticker. The value associated with each key (ticker) is a list of orders (list of “BUY” and “SELL”)

**Backtesting:**

Pending