Mehdi Raza Khorasani

Lab 07 – Applications of DSP in Finance

1. Objectives

In this lab, we will use simple digital signal processing techniques on time-series data from financial markets. We will build a simple trading algorithm, based on processed data, to try and generate outperformance from the broader market. If your algorithm can outperform a broader market, it is referred to as having an 'alpha'.

2. Background Review

Stocks, or *shares* of a company, represent partial ownership of the company, which gives shareholders the voting rights as well as a claim on the earnings of the company. *Stock markets* are where individual and institutional investors come together to buy and sell shares of companies at a public venue. Nowadays, these exchanges exist as electronic marketplaces. Share prices are set by supply and demand in the market as buyers and sellers place orders.

A *stock index* (such as KSE100, S&P500) represents a weighted sum of the share prices of a set of companies. Typically, the weight assigned to each company's share is proportional to its market capitalization (the product of share price and total number of shares). For example, KSE 100 represents the weighted sum of the share price of top 100 companies listed at the Pakistan Stock Exchange (previously, known as Karachi Stock Exchange). Similarly, S&P500 represents the weighted sum of the share price of 500 leading publicly-traded companies in the US.

Besides purchasing stocks/shares of a single company, there are various other investment vehicles/instruments available in advanced stock markets. An interesting investment option is to buy a unit of the *Exchange Traded Fund (ETF)*. There are many types of ETFs. However, in a simple form, buying a share/unit of an *index ETF* (corresponding to a stock index) can allow you to obtain ownership (albeit a very very tiny one) in all the companies included in the underlying stock index. For example, we will work with the SPDR S&P 500 ETF (known by its ticker symbol 'SPY'). In simple words, you can buy a share of SPY, just like the share of an individual company, however, it will give you ownership (again a very very tiny one) in the profits of all the S&P 500 companies.

As an investor in a stock or ETF, you can pursue various trading strategies that dictate when you buy or sell these shares over a time-period. However, the "quality" of these trading strategies can be determined by comparing your investment value against the passive strategy of just buying certain number of shares/units and holding them for a long time, say 10 years.

3. Setting up the Environment

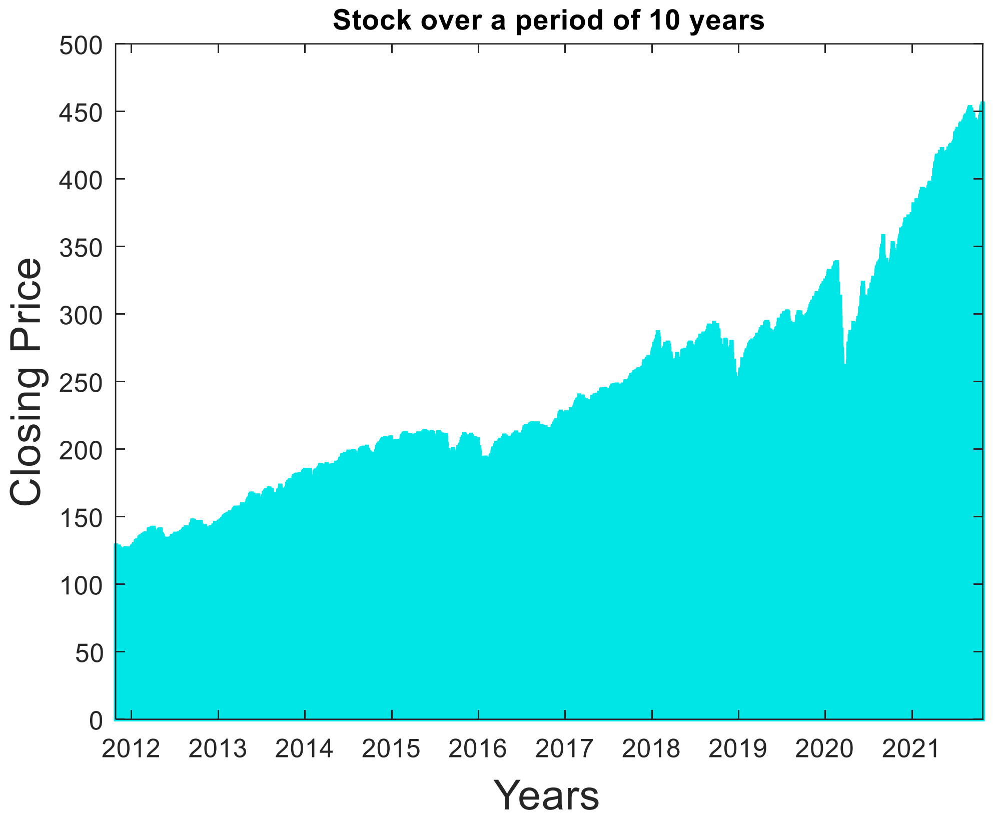
We can use Yahoo Finance (https://www.finance.yahoo.com) to download the daily closing price of the ETF SPY for last 10 years, by following these steps:

- Go to Yahoo Finance (https://www.finance.yahoo.com)
- Enter the quote 'SPY' into the search field.
- Select a quote in the search results to view it.
- Click Historical Data.
- Select a time period (10 years), data to show, and frequency.
- Click Apply.
- To use the data offline, click Download.

Through these steps, we will get the 'SPY.csv' file that can be opened in Microsoft Excel. Once we are able to view the data in an Excel sheet, we can also import it into MATLAB through the *readtable( )* command.

Task 1:

Write a script that imports the 'SPY.csv' file into MATLAB and plots the daily closing price of SPY for the last 10 years. Compare your plot with the chart for closing price of SPY on yahoo finance for the last 10 years. Your chart should look pretty much identical to the one you see online.



The script is as follows:

```
history = readtable('SPY.csv');
history.Properties.VariableDescriptions
bar(history.Date, history.Close, 'FaceColor',[0 .5 .5],'EdgeColor',[0 .9
.9],'LineWidth',1.5)
xlabel('Years', 'fontsize', 16)
ylabel('Closing Price', 'fontsize', 16)
title('Stock over a period of 10 years')
```

You are now all set up to experiment with the time-series, closing price dataset for SPY.

4. Designing Trading Strategies

When designing a trading strategy, remember, beating S&P (generating better returns than just buying and holding SPY ETF) over a time period is not easy. Hundreds of thousands of dedicated portfolio managers have tried it for decades with very limited success. However, this exercise is to get you acclimatized with some basic digital signal processing tools that can be used to design trading strategies.

Our strategy is called a *momentum strategy*. Momentum strategy is based on the differential of two filters. One a long-term filter, and another, a short-term filter. Long-term filter is used to define the average price over a much longer time frame. Short-term filter is designed to indicate when the trend is diverging from the long-term price[1]. The long-term filter is generally averaging over 200+ days, while the short-term filter is averaging over 20+ days. When the difference between the short and long-term filter is positive, we say that the market is in an uptrend. When the difference is negative, (i.e. the value of short-term filter is lower than the long-term one) we say that the market is in a downtrend trend.

For a start, we can devise a loss-prevention strategy. Which can be as follows:

"When the SPY is in downtrend above a certain trigger threshold, sell your holdings and go to cash. When the SPY is in an uptrend again, buy it. The idea is that you exit out of the market when it is going down, while come back again into the market when it is going up. So, ride the highs and ditch the falls!"

For the strategy outlined above, we need to design long-term and short-term filters. A simple moving-average FIR filter is a great start. Let's say we need to build a 300-day moving-average filter. The following MATLAB command can create coefficients for this filter:

```
>> b1 = ones(1,300)./300
```

This command will create coefficients for a 300-point moving average filter, such that the sum of all coefficients is equal to one. We can perform filtering with this filter on a time series x, using:

```
>> y_filtered1 = filter(b1,1,x)
```

[1] To read more about it, you can look up Moving-Average Convergence Divergence (MACD), which is a popular convergence/divergence indicator for stocks.

In this case, the time series ‘x’ is the one obtained by downloading the daily closing price data of any stock/ETF (SPY in our case). The 1 in the filter command indicates that it is an FIR filter, which has no recursive coefficients. The result of filtering the time-series of closing prices for SPY with this long-term filter is another time-series. Let’s call this y_filtered1. Similarly, a short-term filter (as outlined in the strategy) can be designed by using 50 taps in FIR filter instead of 300. The required normalization is the same:

```
>> b2 = ones(1,50)./50
```

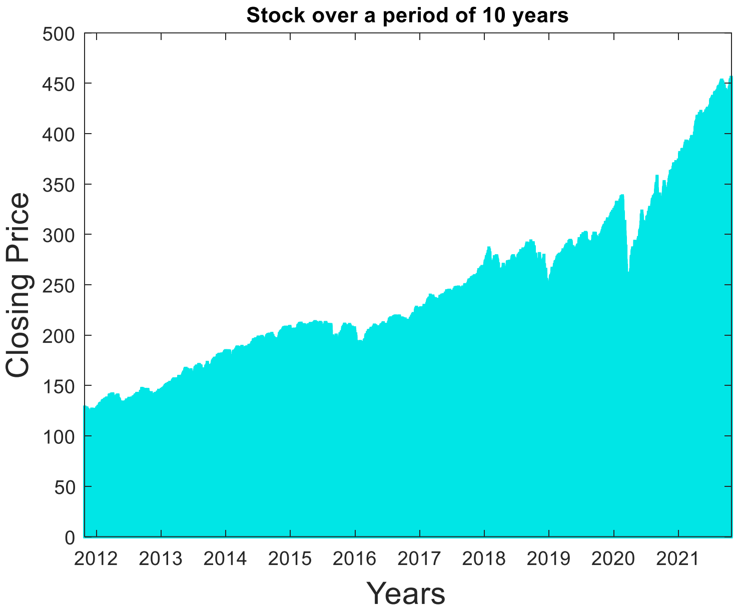
Now, another time-series can be generated by using this short-term filter. Let’s call that time series y_filtered2.

```
>> y_filtered2 = filter(b2,1,x)
```

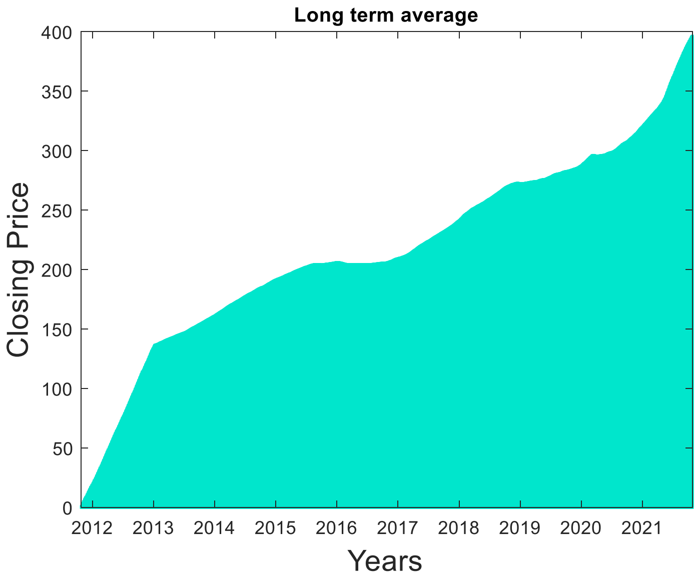
Task 2:

Create a single plot that shows

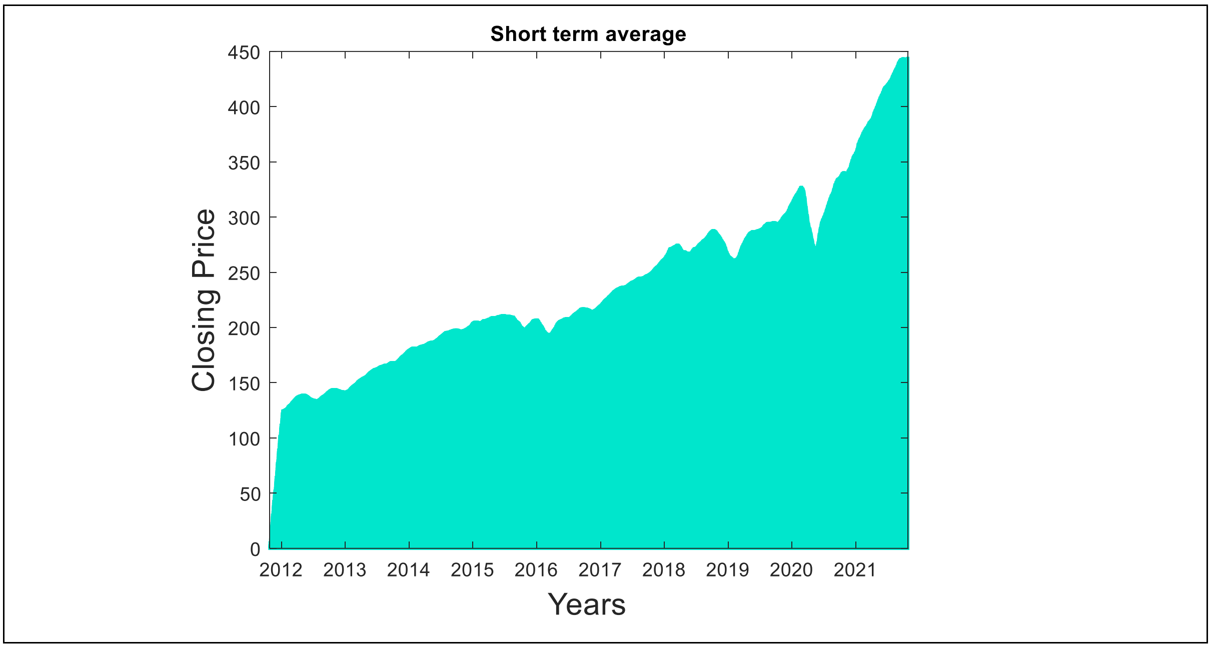
i. The closing price of SPY for the last 10 years.



ii. The output of the long-term filter (300 taps) applied to the time-series of the closing price of the SPY for last 10 years.

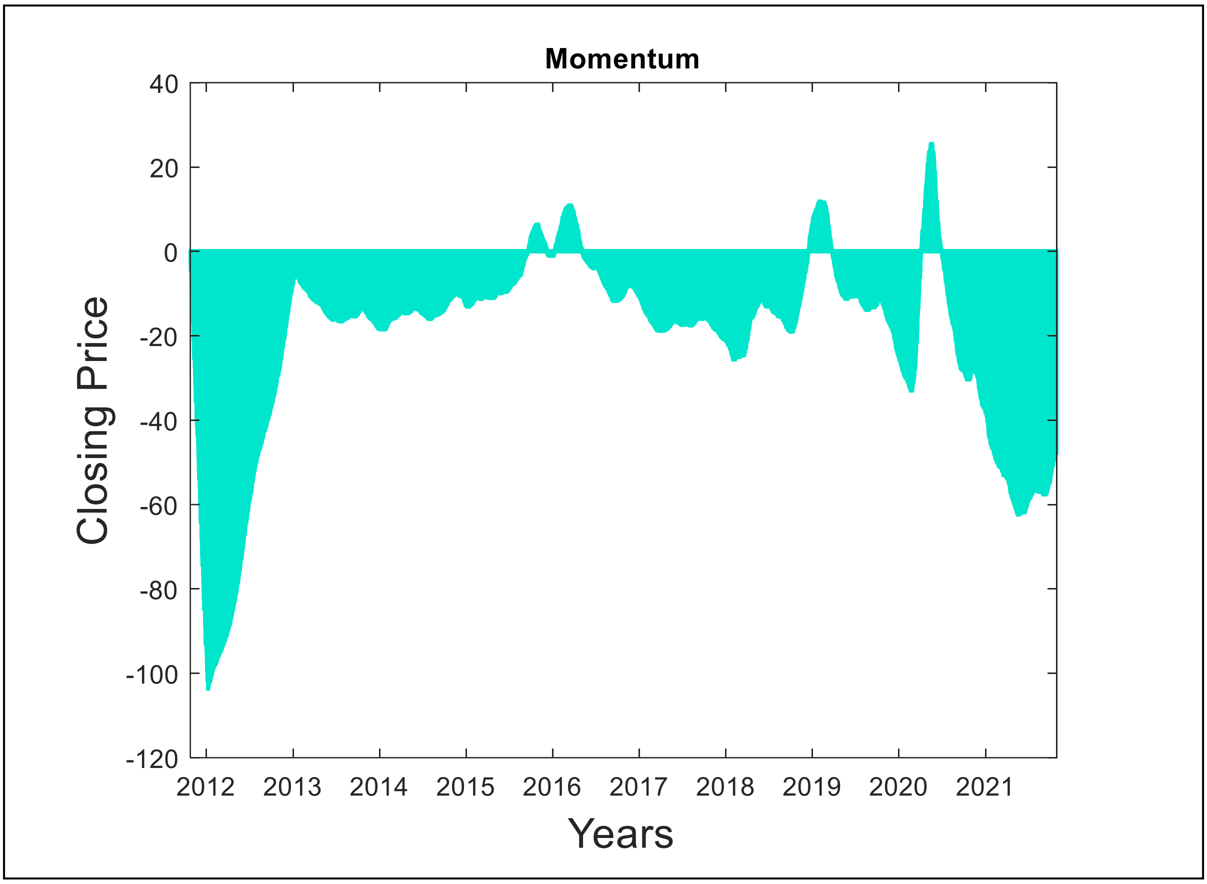


iii. The output of the short-term filter (50 taps) applied to the time-series of the closing price of the SPY for last 10 years.



Now by subtracting y_filtered1 (output of long-term filter) from y_filtered2 (output of short-term filter), we can create a momentum signal. If the momentum signal is positive, we can buy the stock, and if the momentum signal is negative, we can sell the stock. In our simulated environment, buying and selling is simple and it works as following:

Buying just means that we mark the start date of our buying and from then on, we keep tracking the value of stock as the value of our investment. Selling (and going to cash) at a certain point just means that we stop tracking the value of the stock as the value of our investment and instead retain our investment value as fixed (in the form of cash), until the next buy signal.



5. Measuring a Trading Strategy

Let's say we want to test our trading strategy on the data for SPY from last 10 years. At this point, we have three time-series available to use:

1. Closing prices of SPY
2. Output of the long-term filter applied to closing prices of SPY
3. Output of the short-term filter applied to closing prices of SPY

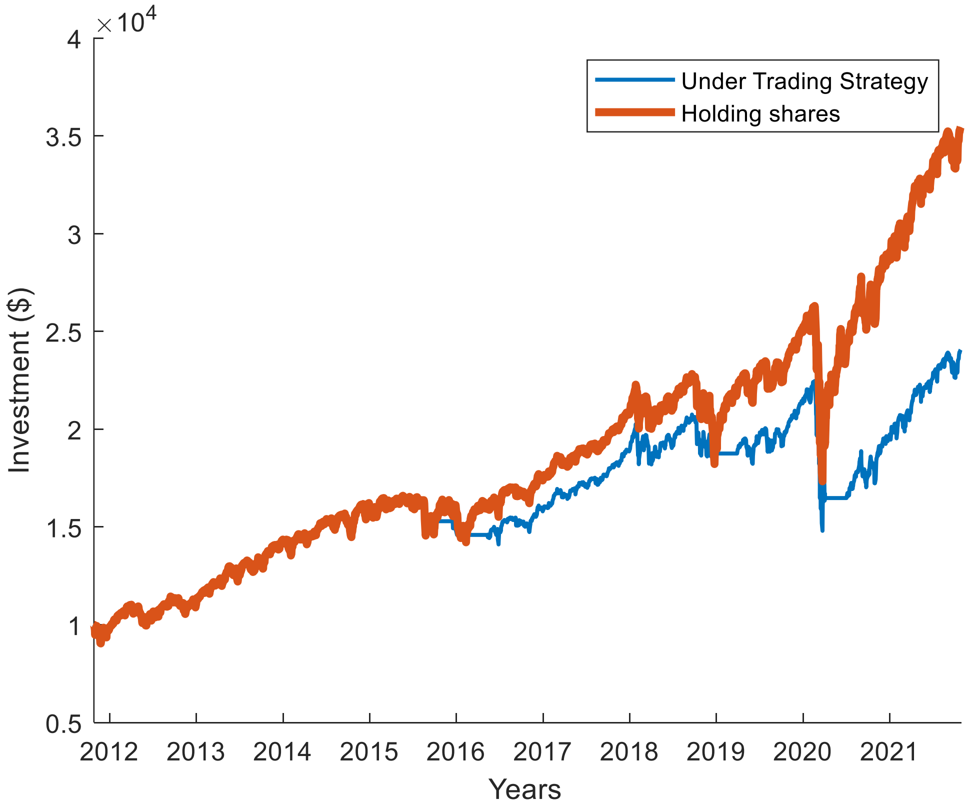
We can write a MATLAB script that loops through the time-series data (for 10 years) mentioned above. At the start of 10 years, we invest (more strictly, we simulate the investment) the nominal sum of 10,000 dollars. Then, we move through the time-series data either staying invested in the stock (in which case the invested money rises or falls with the market) or exiting from the stock into cash (in which case the invested money stays at whatever the value was when we exited to cash). The decision to stay invested or exit into cash is based on the comparison of output of long-term and short-term filters at each point. Once, we are in cash, we can stay in cash until we get a buy signal (when the short-term filter value goes above the long-term filter value). At that point, we move back into SPY shares again by buying SPY shares with the cash we have at that point. We continue this process until today (the end of time-series data) and see whether the value of our investment was greater or less than just buying SPY shares at the start of 10 years and holding it till today.

Task 3:

Write a MATLAB script, implementing the measurement strategy outlined above. As an output of this script, create a single plot that shows the following:

i. Value of the $10,000 initial investment over the period of 10 year under our trading strategy for SPY.

ii. Value of the $10,000 initial investment over the period of 10 years if we just hold the SPY shares for the whole period of 10 years.

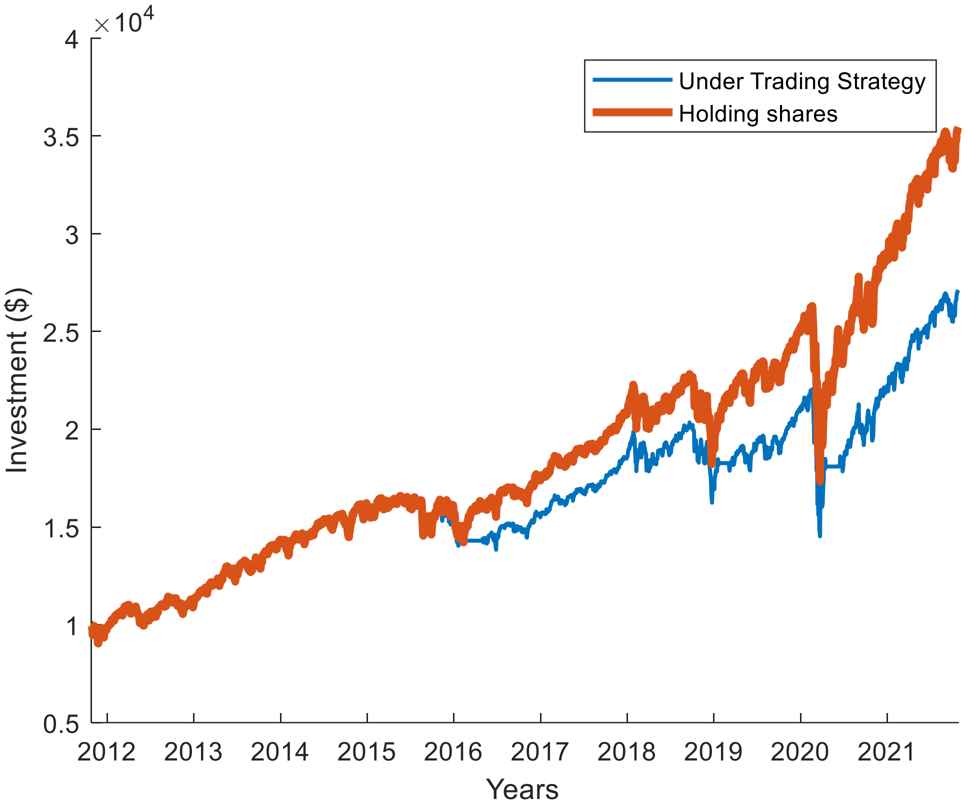


Task 4:

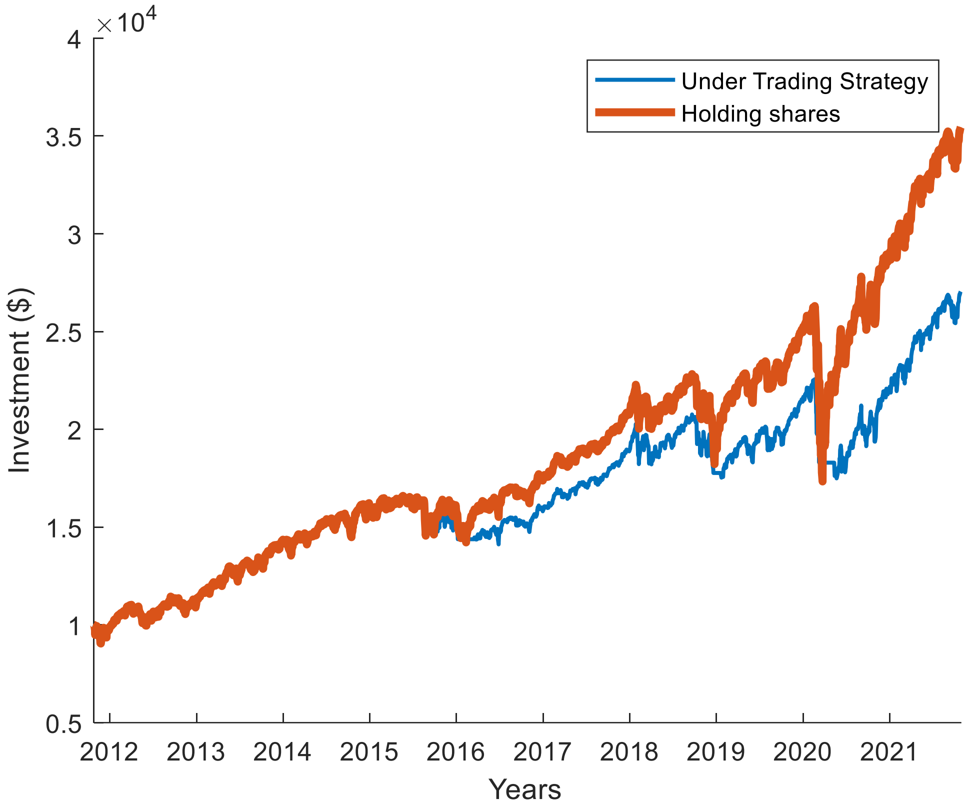
Repeat Task 3 while using 2 different combinations of the following:

- Number of taps in the long-term filter
- Number of taps in the short-term filter
- The difference between the output of long-term and short-term filters that we use as a trigger to generate the buy and sell signal.

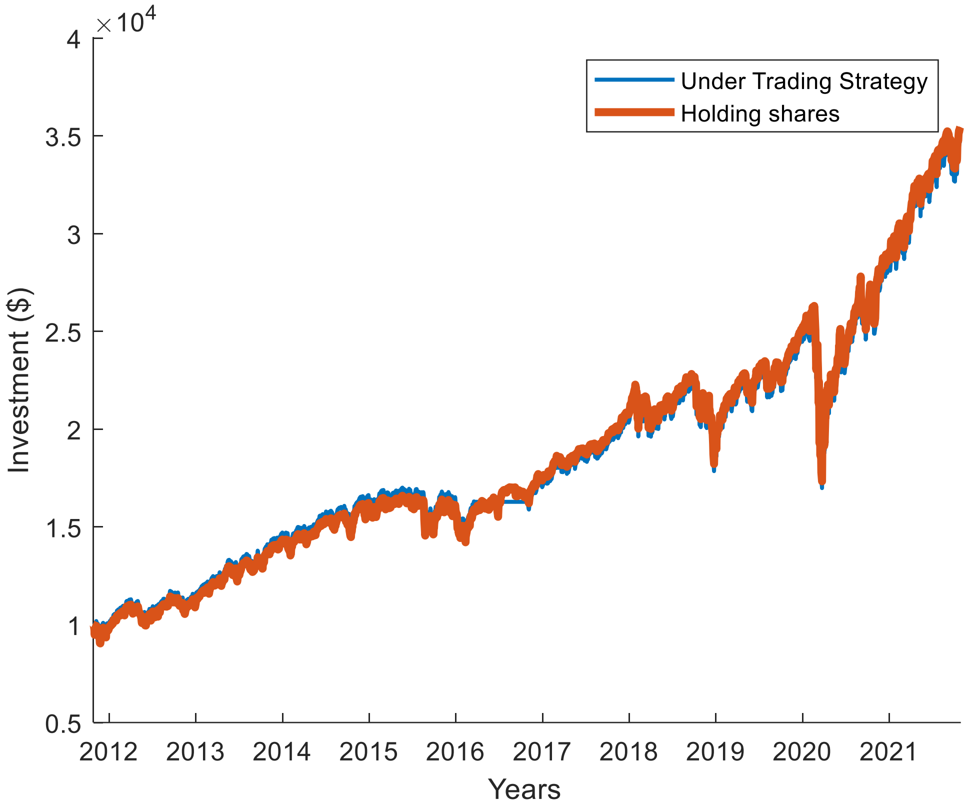
Short term filter: 50
Long term filter: 500



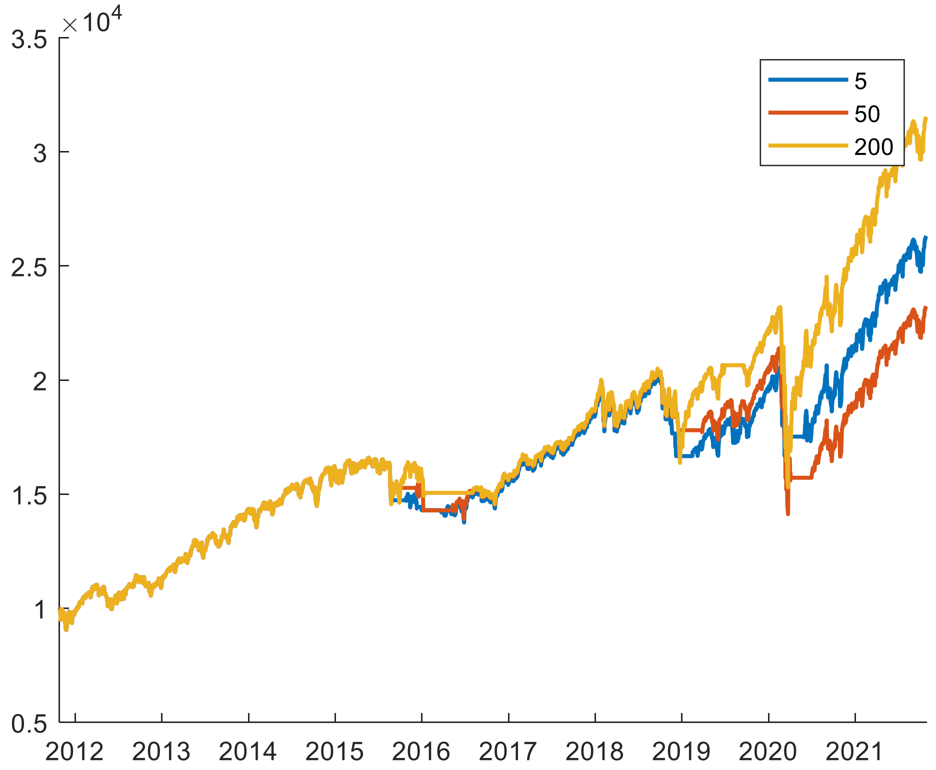
Short term filter: 5
Long term filter: 500



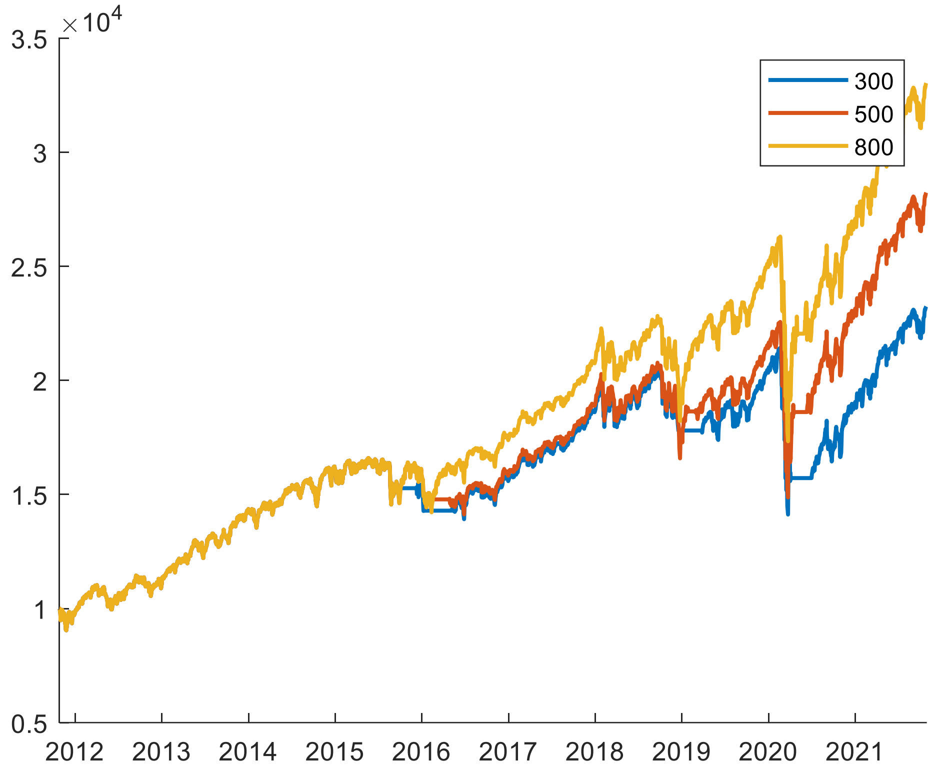
Short term filter: 200
Long term filter: 500



Comparison by increasing short averaging filter values from 5, 50, 200

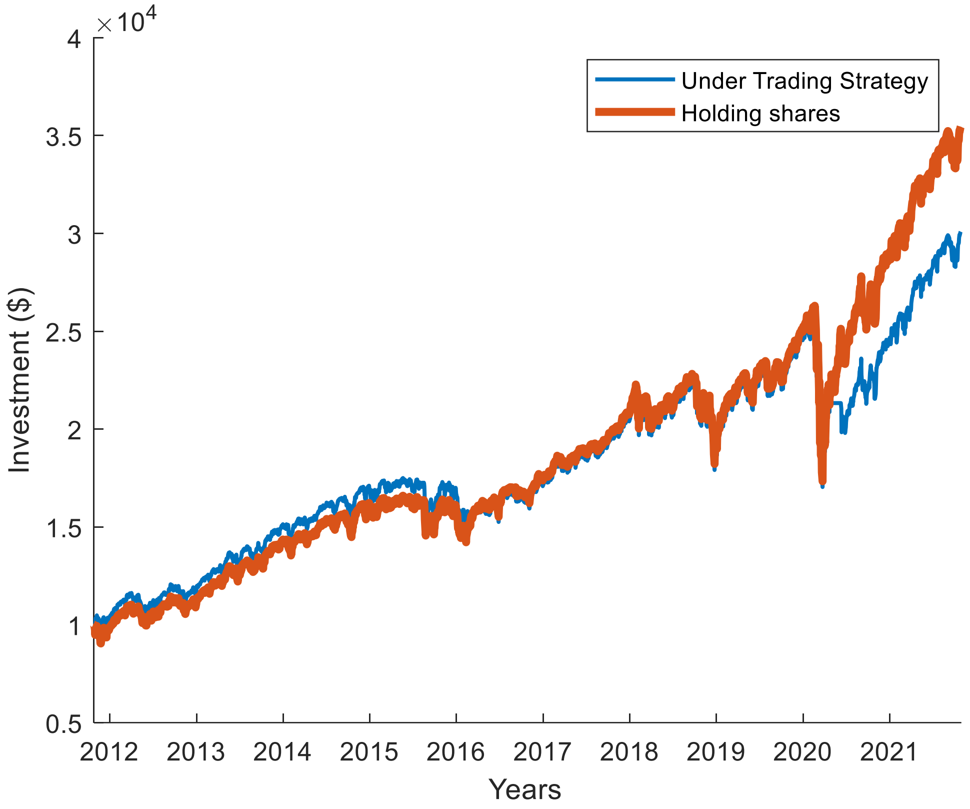


Comparison of strategy by increasing the long term filter points = [300, 500, 800]



All graphs indicate that as we are increasing the averaging points, the investment value is being increased and is reaching the ***hold on investment*** value.

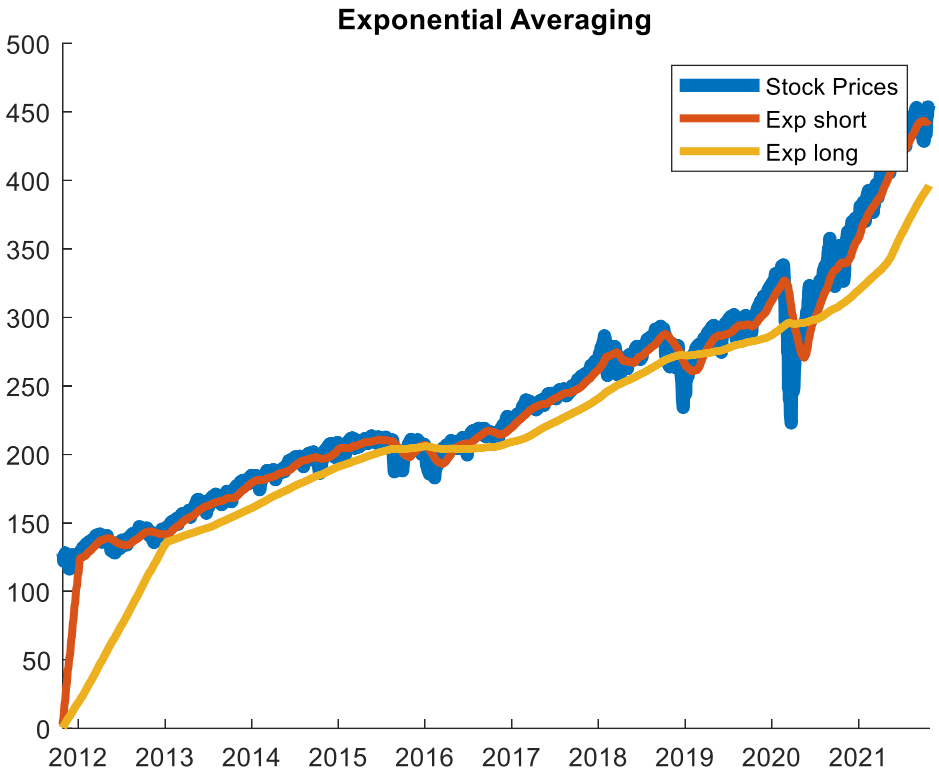
If we use a threshold value of 8 i.e. when momentum crosses 8, we buy/sell. Then the plot is as follows:

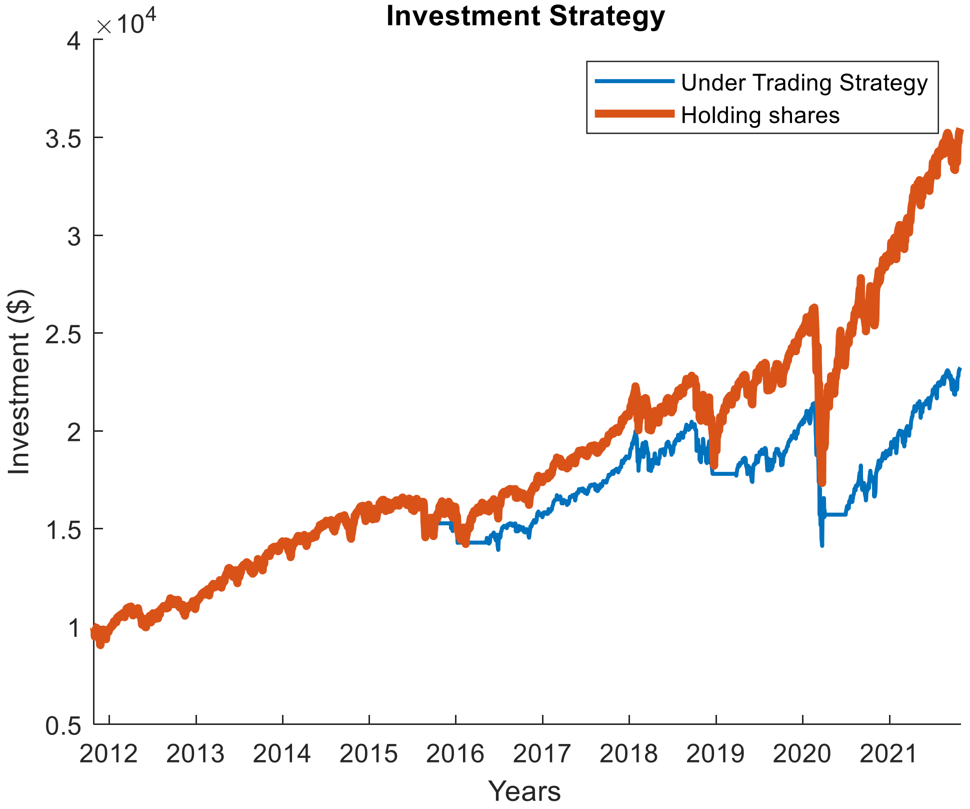


It can be observed, that the strategy is now performing better as compared to when threshold was zero.

6. Post Lab Activity (Optional)

Instead of moving-average filters, try exponential moving-average filters. Basically, while filtering, instead of giving equal weights to all the past samples, you give decaying weights to past samples. This makes the current samples more important. See if converting the filters to exponential moving average helps you with your profits.





Sadly, it doesn't help much. We will be better off holding shares and not doing anything.