Imperial College London

MATH70127 - Algorithmic and High-Frequency Trading

Imperial College London

Department of Mathematics

Coursework

Authors:
Max Dedieu (02289032)

Lecturer:
Kevin Webster
Johannes Muhle-Karbe

May 9, 2023

Contents

1 Introduction

The world of finance has evolved significantly in the past few decades, and algorithmic trading has become an integral part of it. Algorithmic trading refers to the use of computer programs to execute trades automatically, without any human intervention. One of the critical aspects of algorithmic trading is understanding the impact of a trade on the market's price. In this coursework, we will focus on price impact modeling for stock trading and algorithmic trading in general. We will explore various models to estimate the impact of trades on stock prices and how to use these models to generate alpha signals for optimal trading strategies.

The coursework is structured as follows: First, we will introduce the concept of volume curves and VWAP schedules, which are crucial for understanding the objective of modelling price impact. Then, we will delve into the various price impact models and discuss how to fit them to empirical data. Next, we will discuss how to remove the impact of trades from the price and explore various techniques to generate alpha signals. We finally will use these alpha signals to deduct the best trading strategy and observe its performance through back-tests. All the code used to generated the results can be found here here.

2 Preliminary work

2.1 Volume curves

In this first part, we will use the binned databases. This databases, comprised of several ".csv" files, contains aggregated data from trades on 50 stocks listed on NYSE, spanning over the course of a year, with a granularity of 10 seconds.
We first wish to compute the volume curve, which is defined as bellow :

$$V_t = \mathbf{E}[\sum_{s>t} v_s|\mathcal{F}_t]$$

For each stock/date pair, we compute the volume curve V_t, using the average traded volume in the last month.
To do so, we put each month's data into matrix format, and apply pandas transformation, using *pandas*'function *pivot*. We can plot the result several stock/date pairs, as seen in 1.

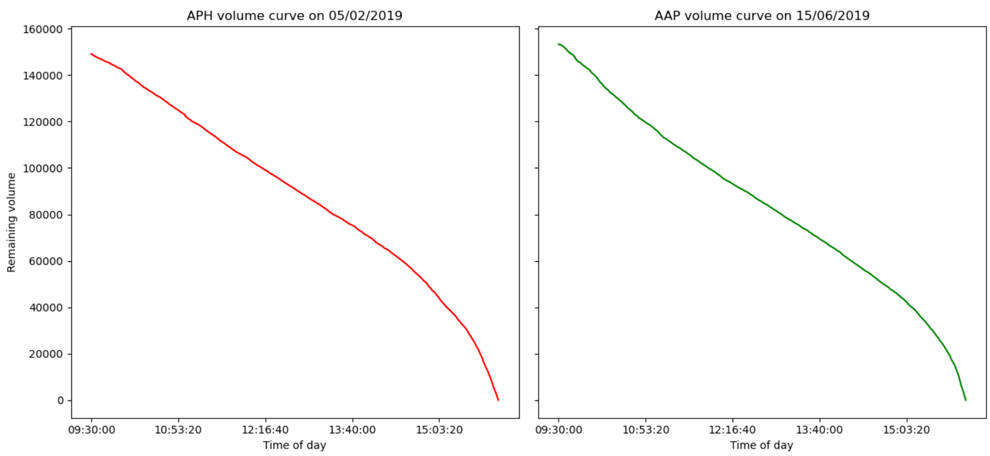

Figure 1: Volume curves of 2 stock/date pairs computed with a 1-month rolling average

The curve have more-or-less the same shape. We notice a distinct acceleration of trading toward the end of day, which is observed in practice.

2.2 VWAP schedules

VWAP, or "Volume-Weighted Average Price" is an execution technique used in algorithmic trading to diminish the impact of a trade in the market. The base concept is to "hide" the order amongst the natural flow of the market within a day. To execute this, we divide the large order Q in chunk of smaller volume, and execute the partial orders q_t during the day. A VWAP schedule determines the volume of the orders at each time-step t.
The first schedule is a "naive" schedule, meaning we introduce look-ahead bias (we know what will be traded during the day). v_t represents the volume traded at time t during that day.

$$q_t^{\text{naive}}/Q = \frac{|v_t|}{\sum |v_s|}$$

To make the model more realistic, we introduce slippage :

$$q_t^{\text{slippage}}/Q = \frac{|v_{t-\delta_t}|}{\sum |v_s|}$$

Finally, to remove the look-ahead bias, we use the monthly-volume curve instead of the future information.

$$q_t^{\text{realistic}} = \min(\frac{|v_{t-\delta_t}|}{V_t}Q, Q - \sum_{s<t} q_s^{\text{realistic}})$$

We can compare the different VWAP schedules by testing them on a percentage of total daily volume Q on a specific stock/date pair. We plot the execution progression as percentage of execution target Q, as seen on figure 2

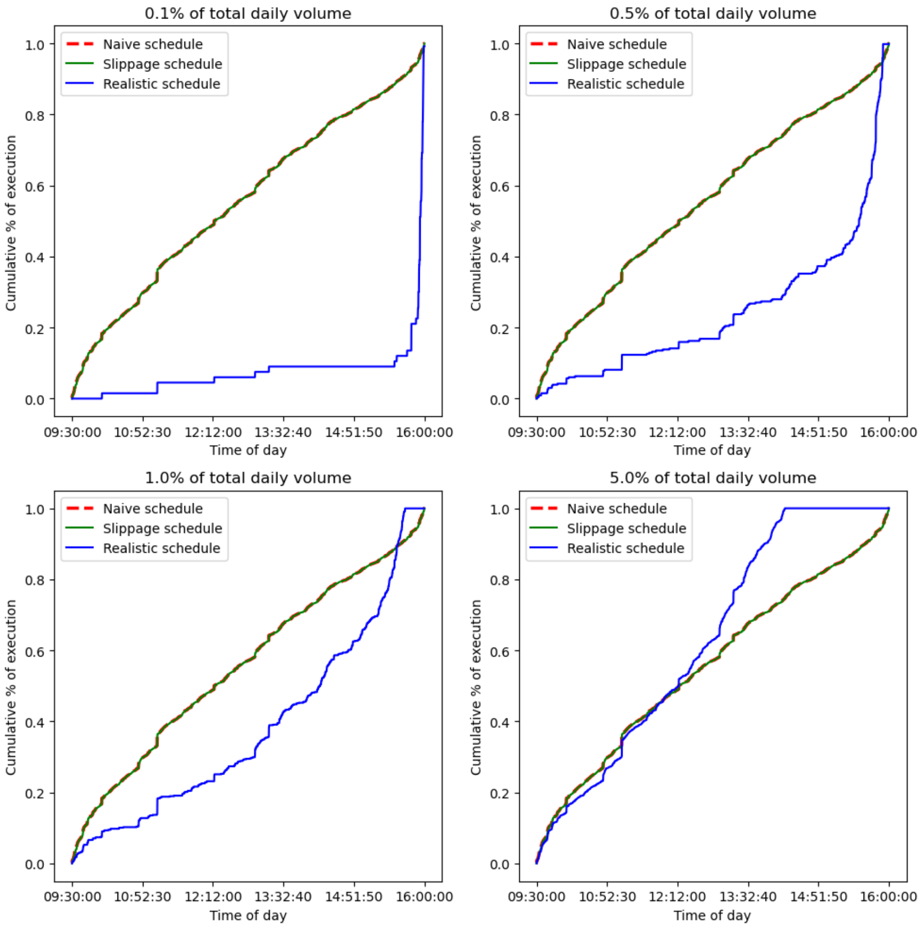

Figure 2: Different VWAP schedule for different percentage of total volume executed of AAPL on 05/02/2019

We notice the VWAP schedule "naive" and "slippage" are independent of Q, while the realistic one takes into account Q, the greater Q is, the faster it will execute the order during the day. To give some idea of the size of Q, 0.1% of AAPL total daily traded volume is around 425,500 USD, for this particular trading day.

3 The price impact models

3.1 Fitting the models

In this coursework, we will study two price impact models. Let $(I_t)_{t\geq 0}$ be the impact caused by trades $(q_t)_{t\geq 0}$, we can derive two differential equations seen as EMAs :

$$I_{t+\delta t} - I_t = -\beta I_t \delta t + \lambda\sigma \frac{q_t}{ADV} \tag{1}$$

$$I_{t+\delta t} - I_t = -\beta I_t \delta t + \lambda\sigma \operatorname{sign}(q_t)\sqrt{\frac{q_t}{ADV}} \tag{2}$$

Where σ is the stock volatility, ADV is the average daily volume, δt is an infinitesimal increase in time and β, λ are model parameter. β can be seen as a mean reverting speed, or duration of impact, and λ as the impact strength. As such, β is linked to the half-life $h_{\frac{\cdot}{2}}$ of the impact by this relation :

$$h_{\frac{\cdot}{2}} = \frac{\log(2)}{\beta}$$

This is a very helpful relation since we can select a target half-life, and remove an unknown from 1 and 2. The only unknown left is λ. These two models are very similar, they just differ by the "kernel" function we apply to the unit-less variable $\frac{q_t}{ADV}$, this translates well in code, where we can create a unique function for both models, and pass the kernel as a parameter.
To fit the model, we select a target half-life $h_{\frac{\cdot}{2}}$, and set $\lambda = 1$. This way, we have no unknown in 1 and 2. We compute the impact of historical trades observed during the training period, using the equations. We then have a series of impact $(I_t(\beta, 1))_{t\geq 0}$. We then select an explanation period h, and we compute the change in price and impact :

$$\Delta_h P_t = \frac{P_{t+h} - P_t}{P_t} \qquad \Delta_h I_t(\beta, 1) = \frac{I_{t+h}(\beta, 1) - I_t(\beta, 1)}{I_t(\beta, 1)} \tag{3}$$

To find the values λ that fit to each stock, we compute the coefficient of the regression :

$$\Delta_h P_t = \lambda \Delta_h I_t(\beta, 1) + \alpha + \epsilon_t \tag{4}$$

Where α is the intercept term and the $(\epsilon_t)_{t\geq 0}$ are gaussian i.i.d .
This gives us the coefficient of impact λ for each stock. To model the linear regression, we use sample data (called "in-sample data" or "is"), and test data, to assess the performance of the model (called "out-of-sample data" or "oos"). A key point to assess the performance of a linear regression model is the $\mathbf{R}^2$ value. This value is defined as below :

$$\mathbf{R}^2 = 1 - \frac{\sum_{i=0}^{N}(y_i - \hat{y_i})^2}{\sum_{i=0}^{N}(y_i - \bar{y})^2}$$

The closest this value is to 1, the more the model explains the data. If this value is 0, it means the model doesn't explain the data at all. We create a function to compute the lambda on the "is" data and evaluate the $\mathbf{R}^2$ on the "oos" date, from the equation 4.
We can now view our problem has a 2-input function that takes in β, h and gives the λ and "oos" $\mathbf{R}^2$ for each stock. Since we have many stocks in our data-set, we assess the quality of a (β, h) pair

by plotting the "oos" $\mathbf{R}^2$ distribution across stocks. This is what we have below, for the linear and non-linear kernel models, on figure 3.

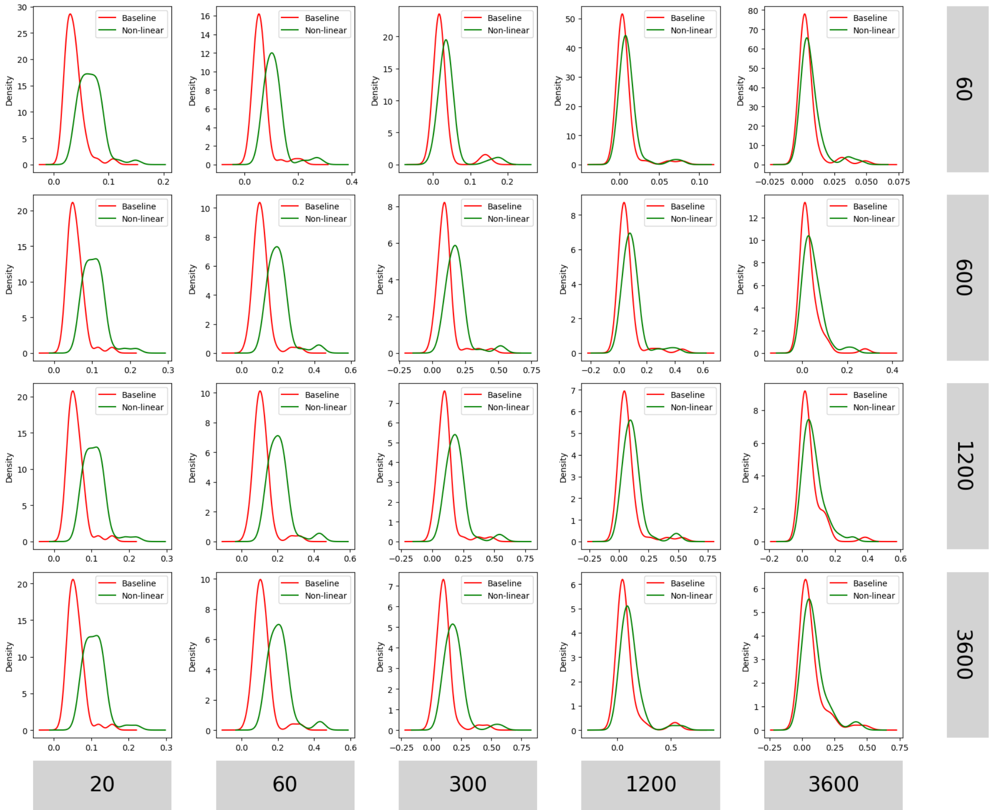

Figure 3: "oos" $\mathbf{R}^2$ Distribution across stocks for different values of $h_{\frac{.}{2}}$ (rows) and h (columns), both in seconds, for the linear and non-linear models

As we can see, the distribution of $\mathbf{R}^2$ are greatly affected by the choice of parameters, we find that $h_{\frac{.}{2}} = 1$ hour and $h = 5$ minutes create the most satisfying distribution of $\mathbf{R}^2$. We do not wish to find the exact best parameters for our data because this will cause over-fitting and create unsatisfactory results on other data-sets.
We can also compute the "is" and "oos" $\mathbf{R}^2$ for different pairs of month (usually the training month comes just before the testing month), to assess the robustness of our linear regressions. We then have a series of λ_{stock} for each data-set pair. We can create the statistic $S_{\lambda_{\text{stock}}} = \frac{\bar{\lambda}_{\text{stock}}}{\sigma_{\lambda_{\text{stock}}}}$ where $\sigma_{\lambda_{\text{stock}}}$ is the standard-deviation of the series of λ_{stock}. Usually, under the assumptions of normality, we assume a value $S_{\lambda_{\text{stock}}} \geq 2$ means the results of the regression is statistically significant (at a 95% confidence interval). We plot this result below in figure 4

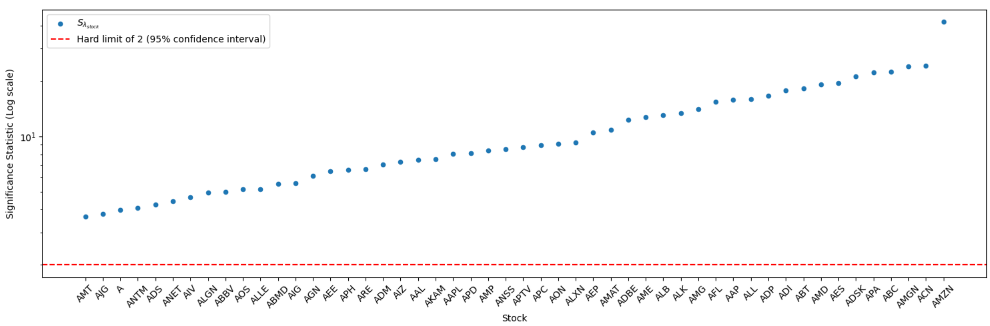

Figure 4: $S_{\lambda_{\text{stock}}}$ statistic for all the stock in the data-set, where $h_{\frac{.}{2}} = 1$ hour,$h = 5$ minutes

As we can see, the results of the regression pass the statistical significance test.

3.2 Removing the impact

With the part above, we have computed the prices impact coefficient for each stock, using the equations 1 and 2, we can compute the impacts of the trades Q_t^{hist} we observed during a trading day, and simulate the "impact-adjusted" price of the stock S_t, meaning the price if those trade did not happen. We proceed using below formula :

$$S_t = P_t^{\text{hist}} - I_t(Q_t^{\text{hist}})$$

We create a piece of code to generate the "impact-adjusted" price data-set going from a model type ("baseline" or "non-linear"), a half-life $h_{\frac{1}{2}}$ and an explanation period h. We see a sample on figure 5 below, for a particular trading day and stock pair.

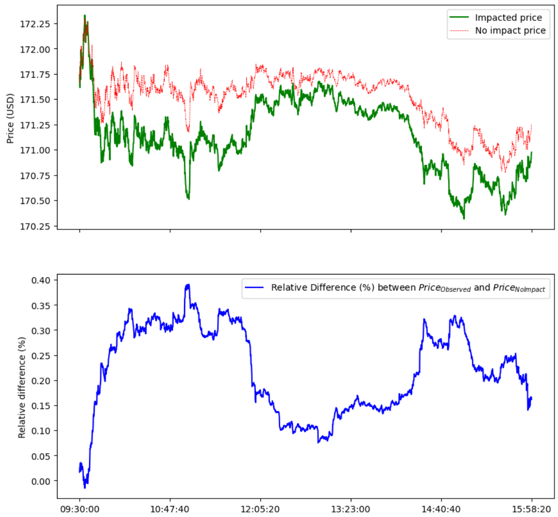

Figure 5: Impact adjusted price of "AAPL" stock on 2019-02-14 using the baseline model $h_{\frac{1}{2}} = 1$ hour and $h = 1$ minute

Computing the "impact-adjusted" price allows use to back-test trading strategies on historical data. Taking a trading strategy x that yields trades $(Q_t^{\text{X}})_{t \geq 0}$, we can compute the impact of this strategy using below formula :

$$P_t^X = P_t^{\text{hist}} - I_t(Q_t^{\text{hist}}) + I_t(Q_t^X)$$

This constitute a back-test method, since we can then assess the performance of our trading strategy. In this fashion, we reproduce some trading strategies to execute a fixed percentage of total daily volume of a particular (stock/date) pair. We will study :

- The realistic VWAP seen in part II
- The TWAP execution algorithm which sends uniform orders "1/N" during the day at each time-step
- A random trading strategy that sends buy order during the day for the set total amount. The cumulative volume follows roughly an exponential function, with random hits

We assess the impact of these three strategies in the below figure 6

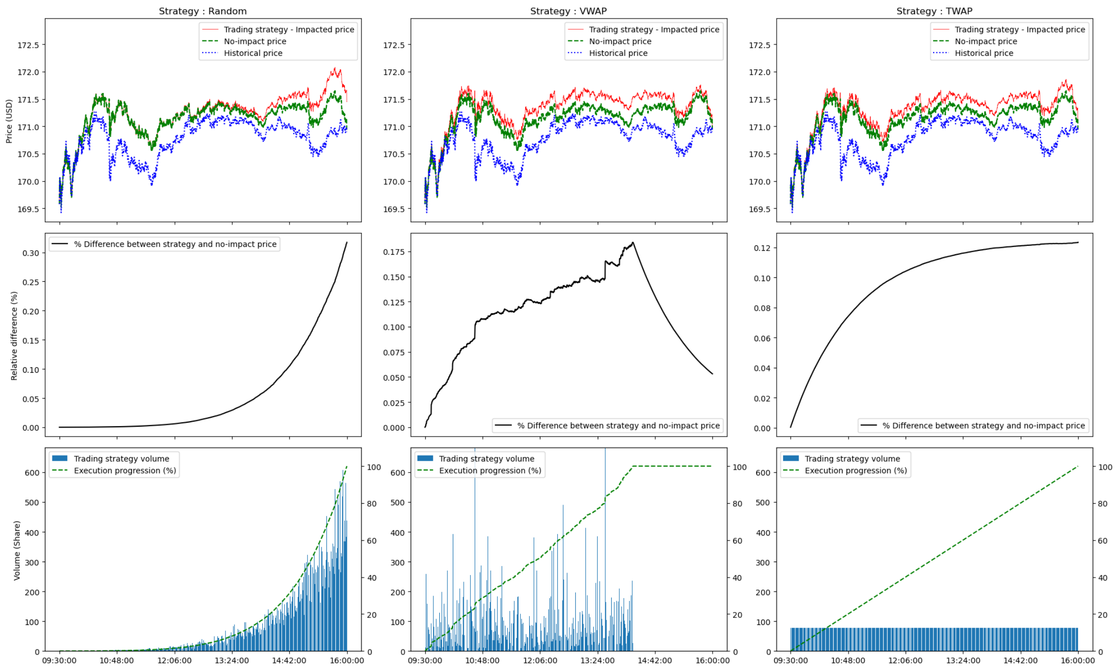

Figure 6: Different trading strategies for "AAPL" stock on 2019-02-14, total execution size is 0.1% of total volume ($\simeq 31,000,000 USD$)

As we can see, the first, pseudo-random, trading strategy has an exponential impact, this is because the trades don't have time to be dissipated in the total volume traded in the market, since they are growing exponentially in size. It is the strategy with the most impact. The realistic VWAP sends trades until it has executed the target amount, and from there the impact decays exponentially. It has the least impact out of the three strategies because it successfully "hides" the trade in the market when the volume is high. The last strategy is a naive TWAP, which executes a fixed sized every time step, the cumulative traded volume is linear. Its impact is not great, but not terrible.

4 Alpha signals

4.1 Synthetic alphas

An alpha signal is an opportunity of future gain. We can use these signal to send trades in the market and exploit it, but we have to take into account the impact of those trades so that it does not eat into our profit and constitute a loss. Using notation h as a forward look-up period, We define the forward return r_t^h and synthetic alpha α_t^h as below (where W is a Brownian motion and $x, y \in \mathbf{R}^2$).

$$r_t^h = \frac{P_{t+h} - P_t}{P_t} \tag{5}$$

$$\alpha_t^h = \frac{x(P_{t+h} - P_t) + y(W_{t+h} - W_t)}{P_t} = x r_t^h + \frac{y(W_{t+h} - W_t)}{P_t} \tag{6}$$

We wish to find specific values for x and y so that $\mathrm{E}[r_t^h|\alpha_t^h] = \alpha_t^h$ and that $Corr(\alpha_t^h, r_t^h) = \rho$. The computation is in the annex, it yields :

$$x = \rho^2 \quad \text{and} \quad y = \rho\sqrt{1-\rho^2}\sqrt{\frac{\mathrm{V}[r_t^h]}{\mathrm{E}[hP_t^{-2}]}} \tag{7}$$

Equation 6 and 7 allow us to compute our own synthetic alpha signal and control its fit to the actual forward return of a price series. Choosing a fixed (stock/date) pair, we can observe the synthetic alpha and forward returns for different values of ρ on figure 7.

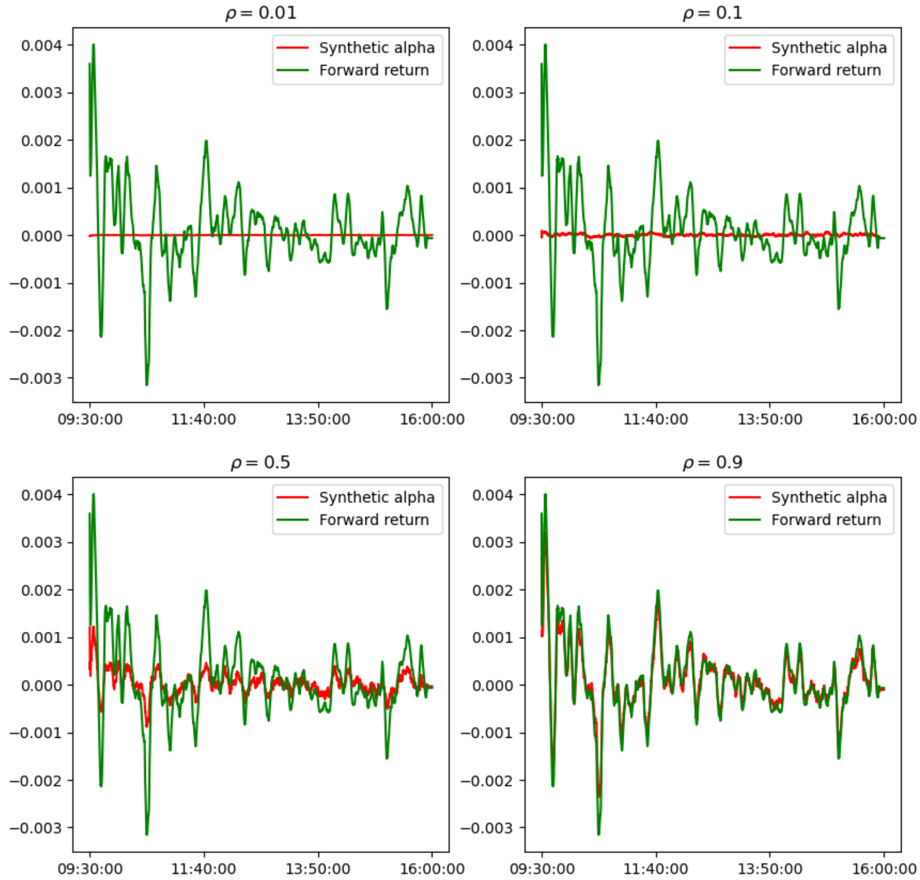

Figure 7: Smoothed synthetic alpha and forward return for $h = 1$minute and different values of ρ

The signals are smoothed with an E.M.A to make it more visible, without loss of information. We can see that the greater the ρ, the better fit between the synthetic alphas and the forward return. We expect a greater P&L if we use a greater value of ρ to simulate an alpha signal.
We can challenge the generation of the synthetic alphas and its supposed correlation with the forward returns. After generating the synthetic alphas with a target ρ for a set (stock/date) pair, we can compute the correlation of each serie with the associated forward return. We get a correlation per (stock/date) pair, and we can then plot it on a histogram, and see how the distribution spreads around the target ρ. This is observed in figure 8

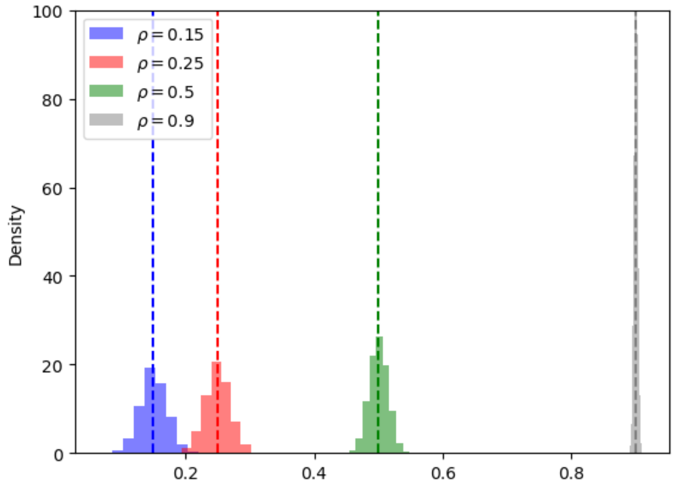

Figure 8: Histograms of correlation between synthetic alphas and forward returns for a few target correlations ρ. The histogram are empirical data, and the vertical lines are the corresponding target ρ

As we can see, the distribution are centered around the target ρ, which means the generation of the synthetic alphas is correct.

4.2 Optimal trading strategy

We have created synthetic alpha signals using a correlation parameter ρ and look-ahead period h, we now wish to implement the best trading strategy to profit from these signal. The aim of our trading strategy will be to negate the impact of the trades so that we lose as little as possible in trading impact.

We can pair the synthetic alpha signal created earlier with the impact-adjusted price to create a baseline for our trading strategy. This is what we observe on figure 9

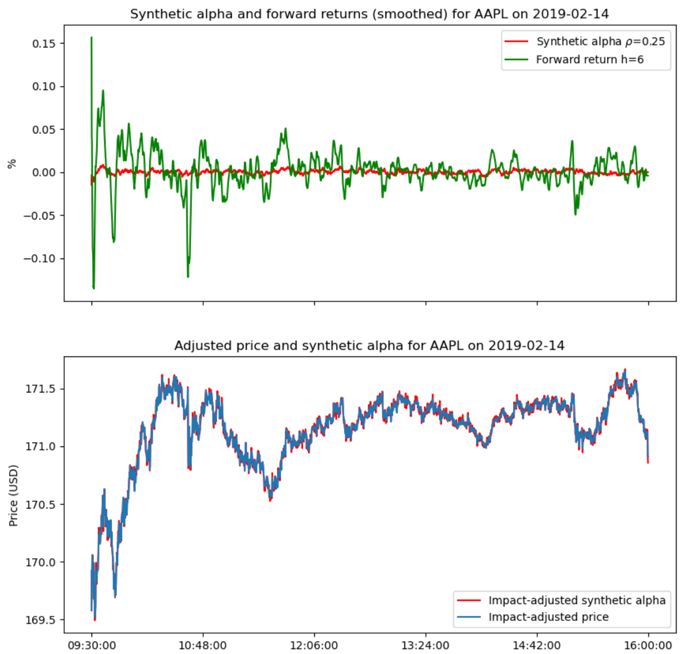

Figure 9: Synthetic alpha with associated forward return (top) and impact-adjusted price with alpha signal (bottom), $h = 1$ minute and $\rho = 0.25$

We have a synthetic alpha signal, we know that we pay roughly half our alpha in impact, we can use below equation to compute the target impact from the alpha :

$$I_t^* = \frac{1}{2}(\alpha_t - \beta^{-1}\mu_t) \qquad \mu_t = \frac{d}{dt}\alpha_t \tag{8}$$

We call the signal μ_t the alpha decay, it is the first order derivative of the signal and helps us determine for how long we can enjoy the high returns predicted by the alpha signal. We understand that computing the derivative of the α signal requires some smoothness, that we do not have in practice. We will apply a smoothing function (rolling mean of exponential moving average with short span) to be able to compute the derivative. We can see the results on figure 10

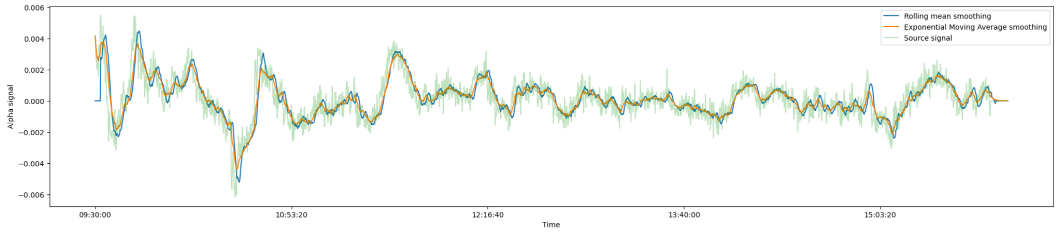

Figure 10: Alpha signal smoothing using different a rolling mean scheme (span = 10 periods, shifted by 5 period) and a exponential moving average (decay factor = 0.3)

We then define :

$$v_t = \sum_{s=t-\Delta_t}^{t} |q_s| \qquad \lambda_t = \frac{\lambda_{\text{scaled}}}{\sqrt{v_t}}$$

Where λ_{scaled} is the coefficient λ from the non-linear model, scaled using the stock ADV and volatility. This ensures the homogeneity of the equations, if α's dimension is a price (in USD). We then reverse the baseline S.D.E from 1 to get an expression of the optimal trades from the impact :

$$Q_t^* = I_t \frac{1}{\lambda_t} + \beta \sum_{s \le t} I_s \frac{1}{\lambda_s} \delta_t \tag{9}$$

We obtain from equation 9 the optimal trades to execute from the target impact, obtained from the alpha signal and the alpha decay signal. This methodology is taken from the "dynamic liquidity" price-impact model.

Knowing the optimal trading strategies, and the impact adjusted prices for a specific (stock/date) pair, we can compute the P&L of the strategy.

We can either use the accounting of fundamental P&L. The main difference between the two is that the accounting P&L uses the book's price to compute the values of the position, whereas the fundamental P&L uses the adjusted price to compute the positions. The fundamental P&L is model-dependant since we need to use an impact model to compute the adjusted price. However, it is fundamentally less biased, since it disregard the impact of acquiring the position in its valuation. The simple example is taking a market participant trading in an non-liquid market. By buying shares, the price will be driven up, and thus the position of the trader will be over-valued. When the trader will want to unload his position, the selling price will be lower than the acquisition price because of the lack of liquidity, and the "theoretical" gain of the trader will disappear.

We thus use the fundamental P&L since we have a price-impact model. We can compute the realised P&L $\Pi_t^{(r)}$ and the expected P&L $\Pi_t^{(e)}$, based on the alpha signal and optimal impact computed in equation 8. The formulas for these two values are :

$$\Pi_t^{(r)} = S_t \int_0^t dQ_s - \int_0^t P_s dQ_s = \int_0^t (S_t - S_s - I_s^*) dQ_s \tag{10}$$

$$\Pi_t^{(e)} = \mathbf{E}\left[\int_0^t (\alpha_s - I_s^*) dQ_s\right] \tag{11}$$

The realised P&L $\Pi_t^{(r)}$ evaluates the value of the positions at time $t = t$, against the cost of acquiring them. The expected P&L evaluates what we can expect the gain to be from the α signal and the optimal impact alone. We expect this value to be higher than the first because it is only what we can hope to capture at best (no slippage, no transactions costs, etc ...). We compile the results of the trading strategy below (figure 11 and 12).

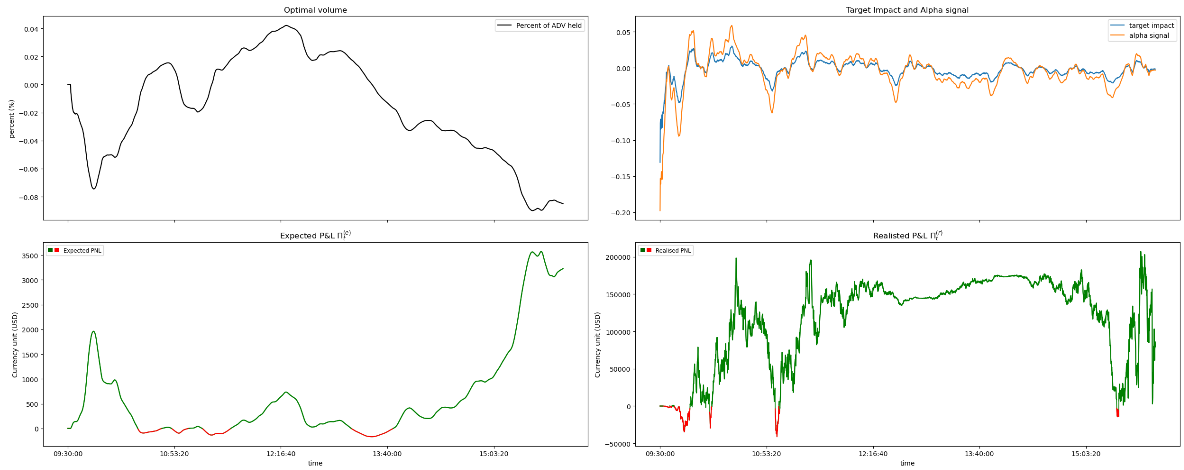

Figure 11: Optimal impact, trades and P&Ls for the AAPL on 2019-12-02, synthetic alpha generated with $\rho = 0.99$

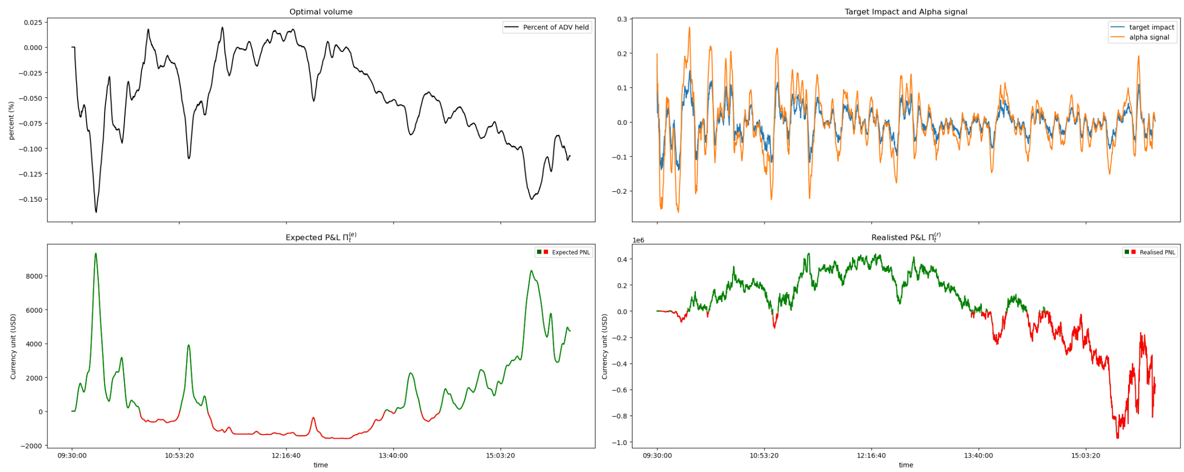

Figure 12: Optimal impact, trades and P&Ls for the AAPL on 2019-12-02, synthetic alpha generated with $\rho = 0.15$

As we can see, the strategies are hugely relying on the quality of the synthetic alpha signal, and its correlation to the forward return. The more correlated our signal is, the greater the P&Ls. Moreover, the signals are not smoothed with the same parameters, since a lower correlation implies lower variation amplitudes (the same smoothing would dampen the signal too much). Additionally, we observe that the optimal trades are for a relatively small percentage of the daily volume traded for the stock.

We can additionally compute performance of the different trading strategies, using standard portfolio management metric such as Sharpe ratio, maximum drawdown, final P&L and such. We compile the results in below table 1.

Correlation ρ	Sharpe Ratio	Maximum Drawdown (USD)	Final realised P&L (USD)
0.01	-0.1065	-199.7	-633001
0.10	0.1467	-73.91	-627113
0.25	0.2256	-120.70	-530701
0.50	-0.0957	-1100.15	-157017
0.75	-0.2000	-53.56	333444
0.99	0.5881	-26.41	1010675

Table 1: Optimal trading strategy performance with various values of correlation ρ for AAPL on 2019-02-12

As we can see, the data indicates a trend toward better metrics with a higher correlation ρ. The data shows some level of noise however, but this is to be expected since the synthetic alpha signal is the realisation of a Brownian motion.
In an effort to make the model and simulation more realistic, we can add trading cost to the P&L computation. The trader pays half the spread in transaction cost each time he makes a trade, since he crosses the order book. To reflect this, we use the column "spread" in the binned data. The new formulas for the P&Ls are as below, using C_t as the spread at time $t = t$.

$$\Pi_t^{(e,spread)} = \mathbf{E}\left[\int_0^t (\alpha_s - I_s^*)dQ_s - \int_0^t \frac{C_s}{2} d|Q_s|\right] \tag{12}$$

$$\Pi_t^{(r,spread)} = \int_0^t (S_t - S_s - I_s^*)dQ_s - \int_0^t \frac{C_s}{2} d|Q_s| \tag{13}$$

We can then observe the impact of the added transaction cost to the back-test by plotting the expected and realised P&Ls with and without the transaction costs. We observe this result in below figure 13

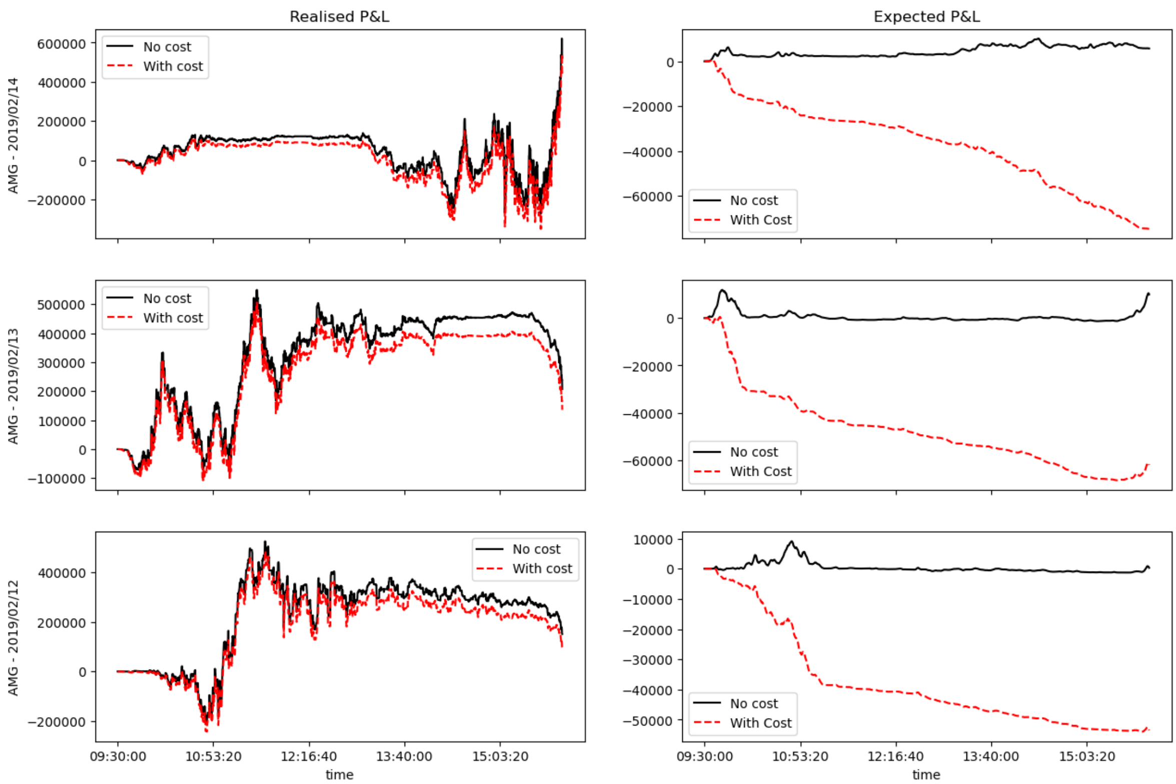

Figure 13: Realised and expected P&Ls for the optimal trading strategy taking into account transaction cost, for stock AMG in 2/2019

As we can see, the transaction cost shift the P&Ls down, as expected. Here the average spread for the stock in the data-set is small (5bps on average). We can however expect some days where these values are much greater, in period of market stress/crash for instance. To simulate this, we can impact the historical spread with a random multiplicands ν_t. We draw the multiplicands ν_t from a shifted log-normal distribution, to ensure they are always greater than 1, and round them to the immediate smaller integer. We choose the shape/intensity parameter s as the "hit"-intensity parameter, the greater it is the greater the multiplicands will be. The p.d.f of the log-normal distribution is given below :

$$\frac{1}{xs\sqrt{2\pi}} \exp\left(-\frac{[\ln(x)-\mu]^2}{2s^2}\right)$$

We can control the shape parameter s, or "intensity" to obtain different distribution, as seen on figure 14

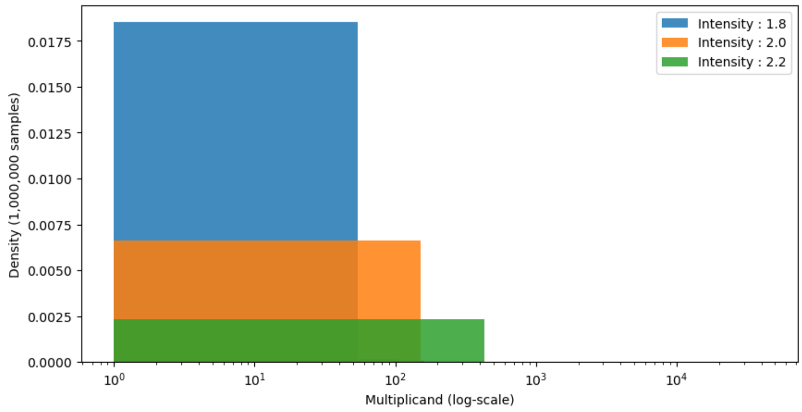

Figure 14: Distribution of the custom log normal distribution with different intensity/shape parameters for 1,000,000 samples

Finally, we can plot a few scenarios with random hit on the spread, with various level of stress, this is shown in figure 15.

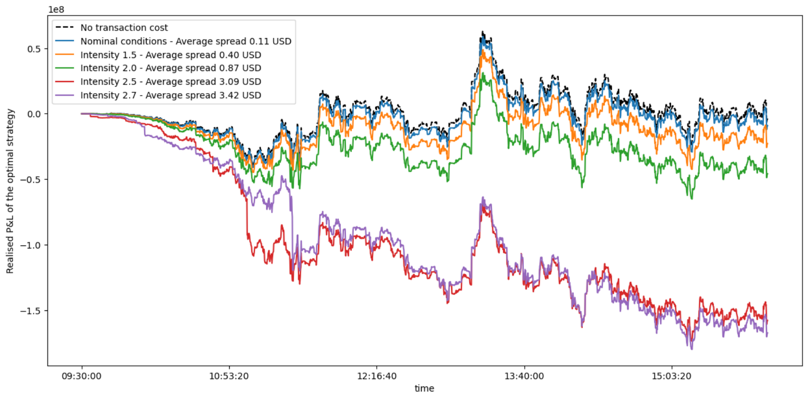

Figure 15: Realised P&L with "hit" transaction cost for various levels of stress, for stock "ALGN" on 2019/05/31

The greater the stress on the market, the worse the strategy will perform using this model. This is to be expected, we understand that this is a crucial part of creating a model, since unprepared trading algorithms have in the past made very costly mistakes to financial firms in periods of market turmoil.

5 Conclusion

To summarize, this coursework allowed us to us real-life data to fit and work on price-impact models. The data was composed of a year of binned-trade data of 50 American stocks for a year. We have worked with different price impact models, fitting the parameters to maximize the fit on "in-sample" and "out-of-sample" data. We then evaluated the adjusted-price for each (stock/date) pair, with the objective to back-test trading strategies. We implemented the standard TWAP algorithm, as well as a realistic VWAP, in order to asses the impact of those execution algorithm on a real day of trading.
Lastly, we used synthetic alpha signals, correlated with the forward return of a stock, to obtain the optimal impact and optimal trades for this stock/date pair. We did this by reverting the SDEs, and implementing the dynamic liquidity model. We compiled the results of this "optimal trading" strategies in straight-forward charts, and evaluated its P&Ls. We also studied the impact of the correlation between the synthetic alpha signal and the forward returns in the performance of the trading strategies. If we were to expand on this work, we would try to implement more sophisticated models, like Bouchaud or A.F.S. We could also create signals overnight, to create a persistent portfolio and asses its performance over multiple days/weeks.

6 Annex

We have :

$$r_t^h = \frac{P_{t+h} - P_t}{P_t}$$

and

$$\alpha_t^h = \frac{x(P_{t+h} - P_t) + y(W_{t+h} - W_t)}{P_t} = x r_t^h + \frac{y(W_{t+h} - W_t)}{P_t}$$

We express the variance of the signal α_t^h using its expression.

$$\mathrm{V}[\alpha_t^h] = x^2 \mathrm{V}(r_t^h) + y^2 h \mathrm{E}[P_t^{-2}]$$

We thus have :

$$Cov(\alpha_t^h, r_t^h) = Cov(x.r_t^h, r_t^h) = x\mathrm{V}[r_t^h]$$

$$Corr(\alpha_t^h, r_t^h) = \frac{x\mathrm{V}[r_t^h]}{\sqrt{x^2\mathrm{V}[r_t^h] + y^2\mathrm{E}[P_t^{-2}]}\sqrt{\mathrm{V}[r_t^h]}} = \frac{x\sqrt{\mathrm{V}[r_t^h]}}{\sqrt{x^2\mathrm{V}[r_t^h] + y^2\mathrm{E}[P_t^{-2}]h}}$$

We now wish to find specific values for x and y so that $\mathrm{E}[r_t^h|\alpha_t^h] = \alpha_t^h$ (1) and that $Corr(\alpha_t^h, r_t^h) = \rho$ (2). We have :

$$\mathrm{E}[r_t^h|\alpha_t^h] = \rho\alpha_t^h \frac{\sqrt{\mathrm{V}[r_t^h]}}{\sqrt{x^2\mathrm{V}[r_t^h] + y^2 h\mathrm{E}[P_t^{-2}]}}$$

We thus have from (1) :

$$\rho\alpha_t^h \frac{\sqrt{\mathrm{E}[r_t^h]}}{\sqrt{x^2\mathrm{V}[r_t^h] + y^2 h\mathrm{E}[P_t^{-2}]}} = \alpha_t^h$$

We have from (2) :

$$\rho = \frac{x\sqrt{\mathrm{V}[r_t^h]}}{\sqrt{x^2\mathrm{V}[r_t^h] + y^2 h\mathrm{E}[P_t^{-2}]}}$$