ETF MARKET MAKING

Jonathas Diogenes Castello Branco (CID: 01461378)

Department of Mathematics
Imperial College London
London SW7 2AZ
United Kingdom

Thesis submitted as part of the requirements for the award of the MSc in Mathematics and Finance, Imperial College London, 2017-2018

Declaration

I hereby declare that this thesis, titled *ETF Market Making*, and the work presented in it are my own unless otherwise stated. I confirm that:

- This work was done wholly for a MSc in Mathematics and Finance degree at this University.
- No part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Jonathas Diogenes Castello Branco
11 September 2018

Signature

_____

Date

_____

Acknowledgements

I would like to thank my supervisor Dr Thomas Cass, Professor of Mathematics at Imperial College London, for his readiness to help and support, as well as for his valuable suggestions. I would also like to thank the experts who were involved in the validation survey for this research project: Douglas Machado Vieira, PhD candidate in Mathematics at Imperial College London, who helped me review and provide important suggestions along my research in his area of domain; Oswaldo Luiz do Valle Costa, PhD in Electrical Engineering at Imperial College, who was my supervisor for my MBA dissertation, and Dorival Leao Pinto Junior, PhD in Electrical Engineering at Unicamp, who both helped guide my learning process; and Christopher Brocklehurst and all his colleagues from Barclays for opening their doors and listening to my ideas and for their precious comments on my work. Without their passionate participation and input, the validation of this thesis could not have been successfully conducted.

ETF Market Making

Jonathas Diogenes Castello Branco

September 2018

Abstract

ETF Markets are characterized by two important facts: firstly, they are risk-fungible, and hence they can be hedged, most likely in diverse ways; secondly, they enjoy the creation/redemption mechanism, causing the total number of ETFs issued in the market to float freely, when additional liquidity is provided by the primary market authorized participants. Market makers need to quote bid-offers on ETFs for their clients both competitively and profitably. On quote-driven markets, they must estimate the market impact in their cost of hedging the ETF, while on order-driven markets, they must determine the price and size of their bid-offers in accordance with the NAV of the ETF and the slippage costs of their hedging. In this work, we propose a jump-diffusion model of the limit order book taking in consideration fundamental aspects of ETF markets and the existence of market-neutral market makers. Then, we provide a combined stochastic and impulse control problem formulation on the ETF market maker problem, and derive the *Hamilton-Jacobi-Bellman Quasi Variational Inequalities* that arise from it. Finally, we evidence the inviability of traditional numerical solutions, and propose sub-optimal approximate solutions by using *Reinforcement Learning* techniques, analyzing their properties with respect to the desired qualities of day-to-day market making solutions.

Last revision: 2018/09/11 01:38:45 Z

Table of Contents

1 Introduction

1.1 Market Making

Making a market is the activity by which dealers engage by quoting bids and offers on financial instruments, with the intent of making profit by offering liquidity to the markets. Such dealer is also known as a 'market maker', or 'liquidity provider'. By contrast, the market participants that transact with dealers, either by hitting the bids or lifting the offers quoted by the market maker, are called 'liquidity takers', or simply 'traders'. Nowadays, agents can act in both capacities, and the distinction basically is the need for immediacy in the execution.

When making market only on a non-replicable financial product (like a single stock), a liquidity provider (dealer) in general has no alternative but to hold an inventory in the course of its dealings with clients. Academic literature assumes that market makers expect to be compensated by holding such risk in their own trading books, and that the bid-offer spread is such compensation. The reality is that such agents may not wait for their portfolios liquidation passively in order to realize their P&L, eventually becoming active by sending marketable orders to control their risks according to their risk management guidelines. Also, the volatile nature of the financial prices makes the bid-offer spread so small compared to the price fluctuations that in practice the spread does not play a relevant role for a dealer that is making an order-driven market. In a quote-driven market, the bid-offer spread will be significantly more than that of an order-driven market due to the bigger sizes of the transactions, thus justifying the claims in the literature.

While fundamentally all financial products have a quote-driven market, many also enjoy order-driven markets at exchanges (and other automated execution venues), where trading is generally anonymous but for smaller quantities and require order execution strategies, while trading desks of investment banks typically act as dealers by providing quotes on basically any financial product to their institutional clients, but generally without anonymity and only for larger deals.

In this regard, on order-driven markets, the liquidity provision is a dynamic process with an embedded price discovery mechanism, as larger orders need to be split up into smaller ones and be sent for execution against the limit order book, which takes time. On the other hand, in a traditional quote-driven market, the price is determined by the competition amongst the dealers (clients know all dealers quotes, but the dealers can't see other dealer quotes), and the execution is guaranteed at the quoted price (for firm quotes), entailing a more static process – quotes may be updated but they are generally good for a longer time-horizon.

Both types of markets are complementary to each other, and transactions in the former may lead to transactions on the later. Dealers on a quote-driven market may choose to liq-

uidate their exposure on the order-driven market or look for another agent that is interested in the exposure. Likewise, exposure accumulated by an order-driven market maker may be liquidated directly against a dealer quote. Thus, market liquidity cannot be seen as a segregated pool, but as a set of interconnected liquidity pools. So, in this sense, we consider the price for a trade to be a hidden state variable which can be only be observed in an exchange or by requesting quotes from diverse dealers.

1.2 Exchange-Traded Funds

Exchanged Traded funds were conceived by Nathan Most following the crash of 1987, drawing from his experience as commodities trader. Since then, the ETF market has grown spectacularly, moved by a continuous trend away from actively managed funds towards passive funds. Compared to Mutual Funds, ETFs are not only simpler to get in and out (they can be traded electronically at Exchanges), but they are also cheaper and more tax efficient.

As the name suggests, ETFs can be bought and sold at exchanges, and not directly from/to the fund. Capital flows in and out of such funds through an *Authorized Participant* (AP), by what is known as the *Creation/Redemption Process*. When demand and offer do not match properly at the secondary markets (exchanges), the AP uses the creation/redemption mechanism in the primary-market in order to providing additional liquidity and bring equilibrium to the system. Thus, if there are more buyers than sellers, the AP can issue brand new ETF shares in the primary market and sell them in the secondary market. On the other hand, if there are more sellers than buyers, the AP is also able to buy those ETF shares in the secondary market and redeem them directly from the ETF issuer in the primary market. In this role, the AP acts simultaneously as a market maker and arbitrageur. Competition between these dealers guarantee that the spread charged for these services are competitive and in line with market interest in the funds.

ETFs can be constituted of any asset class. Although widely popular in the equities market, with SPY being the most famous with $229B in AUM (the largest for an ETF), there are ETFs of all kinds: GLD is the largest fund that invest in physical gold, with more than $29B in gold bars at vaults in London, and AGG has more than $56B invested in US investment grade bonds[1]. Some ETFs may be composed of derivatives in order to provide synthetic or leveraged exposure to other risks. Risk-wise, ETFs display the same properties as their underlying constituents: it makes sense to treat fixed income ETFs are a fixed income portfolio and leveraged ETFs as derivatives.

Regardless of asset class or risk profile, ETFs have a *Net Asset Value* (NAV), which is a linear combination of the prices according to the fund composition, calculated with official

[1] As of September 2018 based on information from ETF.com (2018)

end-of-day (EOD) prices. If calculated intraday with the latest available prices, the NAV is called *Intraday Indicative Value* (IIV), but *Intraday NAV* (INAV) is also a popular term. The *Premium-Discount* for an ETF is simply the difference between the trading price and its intraday NAV value. If positive, the ETF is said to be trading at *premium*, whereas if negative, it is said to be trading at *discount*.

In this work, we focus on ETFs that do not make use of any derivatives for their composition and that can be usually created or redeemed by physical settlement, i.e., by delivering or taking delivery of the underlying ETF composition. For these types of ETFs, creation/redemption fees are paid by the AP to the ETF issuer. These fees are driven by offer and demand. In ETFs where there are more buyers than sellers, the APs will end up short the ETF and will eventually have to create new ETFs, thus pushing the creation fee up. On the other hand, when there are more sellers than buyers, the APs will end up long the ETF and will have to redeem some of their position, driving the redemption fee up.

Other factors also affect the final costs to the AP. Any taxes due to the transfer of ownership will play an important role, as is the case of UK stamp taxes and Brazilian IOF. Exchange trading costs, cash funding and ETF lending/borrow rates also influence the final costs for the AP, which influences the Premium-Discount actually observed in the markets.

1.2.1 Hedging and Replication

Hedging an ETF position may involve trading the underlying fund composition as best as possible and/or performing some sort of proxy-hedging, by trading other assets in some rational fashion[2].

A perfect replication is possible for dealers who are authorized participants and that have access to both the ETF market and the underlying market, but only during the period of time both markets are open. This is the case for country ETFs, like EWU[3] and EWZ[4]. Some ETFs never ever have their underlying markets totally open simultaneously, as is the case of EEM[5], and many dealers also do not have access to all such markets. This phenomenon is studied in Levy & Lieberman (2013), who concluded: *"Our findings suggest a structural difference between synchronized and non-synchronized trading hours. While during synchronized trading hours ETF prices are mostly driven by their NAV returns, during non-synchronized trading hours the S&P 500 index has a dominant effect"*. This an effect of the market electing the most liquid broad-market index future available as the go-to proxy instrument to hedge their

[2]The dealer may design such portfolio to optimize a combination of aspects like accessibility, carry costs, transaction costs, liquidity, etc.

[3]iShares MSCI United Kingdom ETF

[4]iShares MSCI Brazil ETF

[5]iShares MSCI Emerging Markets ETF

ETF positions.

Various other reasons can be given to explain the impossibility of perfect replication:

(i) Some ETFs have thousands of underlying constituents, making it unpractical to send thousands of orders[6]

(ii) The underlying basket for some ETFs (notably Corporate Bond ETFs like LQD and HYG) may not be liquid enough, so perfect replication may be extremely hard or impossible, or way more costly than expected

(iii) The dealer may not be able to short sell the underlying constituents, or may be restricted from buying too many shares of specific companies

1.3 Liquidity provision on ETFs

1.3.1 Market-neutral Market Making

Let us consider the case of a market maker who is an authorized participant in an ETF and with market access to the underlying constituents, which we assume trade on the same currency and timezone as the ETF. In such perfectly replicable ETF, a *Market-Neutral Market Maker* will try to be market-neutral as much as possible, avoiding to the greatest extent possible to have any risk exposure in either the ETF or the underlying basket by immediately hedging perfectly its exposure. It will bid the ETF according to the price at which can immediately sell the underlying basket, which is the Basket bid. Accordingly, it will offer the ETF in accordance with the Basket offer. The liquidity it can post will depend precisely on the liquidity available on-screen for the underlying basket. Larger quotes will have to have higher depth in the limit order book. If only such type of agent is posting liquidity on the ETF limit order book, then the shape of the ETF order book is proportional to the shape of the aggregated order book for the underlying basket. In this sense, every *ETF arbitrageur* is also a market-neutral market maker.

If perfect hedging is not possible, a market-neutral market maker will always immediately hedge its portfolio as to have exposure to the broad market indices as close to zero as possible. However, the choice of proxy hedge is subjective, hence the ETF market in most cases is flexible in which bids and quotes the market makers will most likely to post.

1.3.2 Risky Market Making

An ETF market maker that makes a market in various ETFs and that needs to keep inventory (due to differences in the timezone between the ETF market and the constituents

[6]Trading on too many instruments simultaneously may overload both the dealer and the exchange systems, causing the slippage to increase significantly

markets) will inevitably have to incur in some sort of exposure, and consider if it should relax its necessity to be market-neutral. This way, it will have to consider, after trading with a client, if it should hedge or not such transaction. If hedging, it will need to consider the impact of the hedge on its own inventory, as it may help(or not) unwind it, or may help improve (or not) the *marked to market* (MtM) value of its inventory. If not hedging, the trader will book the trade and carry the risk on its own inventory, while it looks for flow on the opposite side. A relaxation or impracticality of the market-neutral premise represents risk, and should be considered as part of the decision making process of the MM, affecting the pricing of its quotes. Constant hedging brings the cost up, leading to less executions in a competitive market. In this sense, we claim that market makers, and in special ETF dealers, must choose their risk-aversion adequately in order to operate.

1.4 Practicalities

There are some practical difficulties in making ETF markets beyond the usual issues faced by market makers in general, from which we mention some:

(i) Proxy hedging is necessary for many ETFs

(ii) Trading systems are not always flexible enough to handle the ETF diversity

(iii) Decisions must be constantly made regarding the creation/redemption in order to optimize carry costs or fulfill delivery

(iv) Creation/Redemption process may be disrupted by various events

(v) Currency hedging is necessary for many International ETFs

2 The Model

Garman (1976) is considered the inaugural work on market microstructure, where it is introduced the analysis of stochastic demand and supply of securities to study the equilibrium price of those same securities.

In market microstructure, a central limit order book model (CLOB) is a mathematical model for order-driven markets that aims to explain how stochastic offer (via limit orders) and demand (via market orders) for a financial asset interact to give rise to executions (trades), at a very "microscopic scale", thus the term "market microstructure". Based on assumptions on the nature of the offer and demand, and the dynamics of this "model" market, results can be drawn, either analyzing hypothesis or producing optimal behavior for agents. Clearly, the actual markets do not work exactly like these mathematical models, and any theoretical results are to be taken and interpreted in this context, giving practitioners some intuition about cause-effect relationships and ideas on how to approach problems in actual markets.

This section will firstly discuss previous market making models before introducing our proposed model. Then we will carry a series of modeling arguments specifically related to ETFs, and finally derive the wealth dynamics that will be used in the next section.

2.1 Early models and related work

2.1.1 Ho & Stoll (1981)

Stoll (1978) introduced the first formal mathematical framework to study the role of dealers and dealers services, helping shape policy and regulatory discussions regarding the intermediary role in financial markets. According to Stoll, dealers are providers of immediacy, and Stoll (1978) is among the first to study the cost of such services, which is broken up into three components: (1) inventory costs, (2) information costs and (3) transaction costs. The inventory costs, as described by Stoll, is related to the costs (risks) that the dealer faces by holding a sub-optimal portfolio in *"order to accomodate the desires of investors to buy or sell a stock in which the dealer specializes"* (Stoll 1978, p.1134, part I) .

Taking such inspiration from Garman (1976) and Stoll (1978), Ho & Stoll (1981, 1980, 1983) focused on the dealer's inventory problem under various aspects, which Ho & Stoll (1981) deserves special attention for being the first concerned about finding a pricing strategy for a risk-averse dealer problem that is concerned about the suboptimal portfolio, using dynamic programming and stochastic optimal control theory. Ho & Stoll use Poisson jump process to model the evolution of the dealers position, i.e. it indirectly uses jump processes to model investors demand for assets. Our work makes this explicit, by understanding that demand for assets, (or immediacy) is represented by arrival of market orders to a limit order

book, hence we embed the market orders arrivals by adopting jumps directly into the asset model.

In Ho & Stoll (1981), the dealer cumulative sales and purchases are modeled by two stochastic jump processes, q_a and q_b, with intensities λ_a and λ_b and constant jump size Q, i.e. $q_a \in \{0, Q\}$ represents the total number of shares sold and $q_b \in \{0, Q\}$ represents the total number of shares bought, and dq_a and dq_b are the number of shares per transaction, with transactions expected to have constant intensity throughout the period. A pricing strategy is posed in terms of calculating what is called the *"price of immediacy"* a and b, the distance of the dealer quotes to its own opinion of the stock price p, and λ_a and λ_b are decreasing functions of a and b respectively, so the further the dealer's quote is away, less likely an execution will be. With p, a and b in hands, the dealer is expected to quote bids at $p - b$ and offers at $p + a$ using passive limit orders. Next, Ho & Stoll (1981) models the dealer's wealth W using 3 components, namely (1) the cash process F, (2) the inventory process I and (3) the base wealth Y. The dealer is then assumed to maximize his expected utility of his wealth at a terminal time T, upon which it is supposed to liquidate its inventory without any transaction costs. This can be also understood as a portfolio optimization problem with single consumption at final time. The solution of the problem is to find a strategy for calculating a and b at each time step until T.

Two major problems in Ho & Stoll analysis: the price p for the stock is assumed to be constant, and while the author claims that modeling price uncertainty into the inventory process (capital gains are modeled as dividends) accounts for all uncertainty, O'Hara & Oldfield (1986) strongly disagrees and arrives at distinct conclusions when the price is assumed to vary itself. Another major problem is the very unrealistic assumption that the intensities λ_a and λ_b are linear functions of a and b. Both of these problems are tackled in Avellaneda & Stoikov (2008).

2.1.2 Avellaneda & Stoikov (2008)

Avellaneda & Stoikov (2008) is an improvement on Ho & Stoll (1981). Firstly, the authors assume the stock price process is stochastic diffusion instead of a constant price. Their choice of an arithmetic brownian motion[7] $dS_t = \sigma dW_t$ is mostly intended to improve the tractability of the problem under an exponential utility function. They also recognize that a geometric brownian motion $dS_t = \sigma S_t dW_t$ can also be as tractable under a mean-variance analysis, but they argue that this would not change the essence of their results.

Secondly, the execution intensities $\lambda_t^b = \Lambda(\delta^b)$ and $\lambda_t^a = \Lambda(\delta^a)$ are exponentially decaying functions of $\delta^b = S_t - S_t^b$ and $\delta^a = S_t^a - S_t$, the distance of the quoted prices to the reference prices S_t. The processes $(N_t^b)_t$ and $(N_t^a)_t$ which drive the purchase and sales are assumed

[7] i.e. a Bachelier model

independent, and the executions are always assumed to be for a single unit instead of Q shares as in Ho & Stoll (1981).

2.1.3 Veraart (2010)

A major characteristic of the market making model of Veraart (2010) is that the dealer is allowed to cross the spread and actively trade against other participants quotes. The dealer trades are thus either "liquidity adding" (the ones initiated by incoming market orders) or "liquidity removing" (the market orders initiated by the dealer). In contrast to Ho & Stoll (1981) and Avellaneda & Stoikov (2008), the quantity of assets bought and sold from the liquidity adding activity are modeled by a diffusion process approximation of a Poisson process, because the authors argue that at large intensities, a Poisson process is normally distributed and converges to a Brownian motion. The drift and diffusion coefficients are functions of the intensity function Λ, closely in the same spirit of Ho & Stoll (1981) and Avellaneda & Stoikov (2008), but since only a numerical solution is proposed, the discretization of the control space allows them to use a "lookup table" for Λ. The liquidity removing trades are modeled as impulse controls, and thus the dealer's inventory problem posed is a *combined stochastic and impulse control problem*, for which the authors propose a Markov chain that locally approximates the wealth dynamics by matching the first two moments, to be calibrated (trained) by policy iteration.

2.2 Basic Framework

In the mathematical model we propose, the market maker (liquidity provider) is allowed to interact with the ETF market exclusively by sending limit orders[8], and traders (liquidity takers) are assumed to use only market orders[9], 2.4. Additionally, in this work we permit the dealer to hedge any or all portion of his inventory by sending market orders on the hedging market, which we describe in the next section, so it can behave as either traditional market makers or as arbitrageurs.

The main idea of our work is to use a jump-diffusion process to model our basic asset (in our case, an exchange-traded fund, but the principle can be applied to any asset), where the same pure jump process dN_t drives both the price jumps dJ_t and our dealer's limit order executions dQ_t. This is more realistic than the Avellaneda & Stoikov (2008) model, because when a limit order that is not at the best bid offer gets executed, the price actually jumps, since a market order wipes out the on-screen liquidity until it gets fully filled, and this adversely affects existing portfolio. Thus the Wiener process models the liquidity provider activity (limit

[8] Non-marketable limit orders, which do not cross the bid-offer spread

[9] Or marketable limit-orders, which we assume to have the same practical effect

orders) while the jump process model the liquidity taker activity (market orders)[10].

Let $(\Omega, \mathcal{F}, (\mathcal{F}_t)_{t\geq 0}, \mathbb{P})$ be the filtered probability space under which $W = (W_t)_{t\geq 0} = [W_t^S, W_t^P]'_{t\geq 0}$ is a 2-dimensional Wiener process under the natural filtration $\mathcal{F}_t$, and N

$$N = (N_t)_{t\geq 0} = [N_t^a, N_t^b, N_t^{ap}, N_t^{bp}]'_{t\geq 0}$$

is a 4-dimensional Poisson process also under the natural filtration $\mathcal{F}_t$. We denote by

(i) $\{S_t : dS_t = S_{t^-}(\sigma dW_t^S - dJ_t^b + dJ_t^a)\}$ the reference price process for the ETF, i.e. at which the ETF is trading and can be closely approximated as the last trade price available or, in absence of any trade, the last trade price for the underlying basket adjusted by the premium-discount (see equation 1).

(ii) $\{P_t : dP_t = S_{t^-}(\theta(\mathfrak{p} - P_{t^-}/S_{t^-})dt + \kappa dW_t^P + dJ_t^a - dJ_t^b + \mathbb{1}_{P_{t^-}<<\mathfrak{p}} dJ_t^{ap} - \mathbb{1}_{P_{t^-}>>\mathfrak{p}} dJ_t^{bp})\}$ the premium-discount process for the ETF, modeled by a modified Ornstein-Uhlenbeck process, whose dynamics will be explained in the next section

(iii) $\{J_t^a : dJ_t^a = r_t^a dN_t^a\}$ and $\{J_t^b : dJ_t^b = r_t^b dN_t^b\}$ the compounded poisson processes representing the positive and negative return jumps

(iv) $\{J_t^{ap} : dJ_t^{ap} = r_t^{ap} dN_t^{ap}\}$ and $\{J_t^{bp} : dJ_t^{bp} = r_t^{bp} dN_t^{bp}\}$ the compounded poisson processes representing the positive and negative return jumps caused by ETF arbitrageurs

(v) $\{Q_t^b : dQ_t^b = \eta_t^b dN_t^b = \eta(r_t^b, \delta_t^b) dN_t^b\}$ and $\{Q_t^a : dQ_t^a = \eta_t^a dN_t^a = \eta(r_t^a, \delta_t^a) dN_t^a\}$ the compounded poisson processes representing the quantity of ETF shares bought and sold

(vi) $\eta(r_t, \delta_t) \in [0, Q]$ the quantity filled for a limit order of size Q as a function of the instantaneous jump return r_t and the distance δ_t of our limit order to the reference price S_t. This function is non decreasing with respect to r_t/δ_t.

(vii) $\{q_t : q_t = Q_t^b - Q_t^a\}$ our dealer's inventory process, i.e, the number of ETF shares representing the dealer current inventory.

(viii) λ_b and λ_a the intensities of the poisson processes N^b and N^a, which we assume are independent.

(ix) λ_{bp} and λ_{ap} the intensities of the poisson processes N^{bp} and N^{ap}, which we assume are independent.

(x) r_t^b and r_t^a the size of the price return jumps, assumed independent and under the same distribution family (with possibly distinct parametrization).

[10] In reality, it is totally fine for the price to move around without any trade.

(xi) r_t^{bp} and r_t^{ap} the size of the premium jumps caused by ETF arbitrage, assumed independent and under the same distribution family (with possibly distinct parametrization)

We work on a continuous limit order book, thus we assume the market maker can continuously post two limit orders for one share according to two stochastic processes $S_t^b = S_t(1 - \delta_t^b)$ and $S_t^a = S_t(1 + \delta_t^b)$, the bids and offers, where δ_t^b and δ_t^a are the bid-offer spread our dealer wants to capture. Then we define the stochastic control $u = u(t, X_t) = (\delta_t^b(X_t), \delta_t^a(X_t))$ as an $\mathcal{F}_t$-predictable markovian process, i.e., it depends on the system history only through the latest available state X_t of the system. No cost is assumed for sending limit orders, and no tick-size concerns are made here.

2.3 Premium-Discount

2.3.1 Conversion levels

At conversion levels, market-neutral market makers can substantially increase the liquidity of the ETF at the expense of market impact to the underlying basket market, causing the premium-discounts to be bounded within the conversion levels. These levels change only as a result of structural changes to the markets, like changes to transaction taxes (UK stamp tax, Brazil IOF) and conversion fees, hence we assume the creation and redemption levels remain static for our problem.

Thus, a buy (sell) order on the ETF will probably cause the premium to increase (decrease), but only up to the conversion levels. In the unusual event that either creating or redeeming, or both, not being possible, then this argument does not apply, and in that case what is observed is that the spread between ETF and the underlying opens up. We do not consider this risk in our work.

Bounded Levy process The mean-reversion nature of ETF premiums-discounts initially suggests using a Ornstein-Uhlenbeck process

$$dP_t = S_{t^-}(\theta(\mathfrak{p} - P_{t^-}/S_{t^-})dt + \kappa dW_t^P)$$

but this alone is not enough for ETF premiums to remain bounded within the conversion levels. If we denote by P_{cheap} and P_{rich} the levels ($-\infty < P_{cheap} < P_{rich} < \infty$) which trigger ETF arbitrage, we can then model the premium by adding two additional jump processes, similarly as done by Hilliard (2014):

$$dP_t = S_{t^-}(\theta(\mathfrak{p} - P_{t^-}/S_{t^-})dt + \kappa dW_t^P + dJ_t^a - dJ_t^b + \mathbb{1}_{P_t<P_{cheap}} dJ_t^{ap} - \mathbb{1}_{P_t>P_{rich}} dJ_t^{bp})$$

where J^{ap} and J^{bp} are compounded Poisson processes representing the ETF arbitrageurs activity. The values P_{cheap} and P_{rich} varies per ETF and the degree of competitiveness. This

modeling choice means the ETF returns and premiums are mutually excited processes, which is clearly the case for co-integrated assets. The intensity $\lambda_{bp}(t)$ of J^{bp} can be made constant or an increasing linear function of P_t, and the intensity $\lambda_a(t)$ of J^{ap} can also be made constant or an decreasing linear function of P_t. The amplitude can be considered exponentially distributed with constant parameter, as we assume all arbitrageurs would use a sequence of similarly sized market orders.

Additionally, the instantaneous correlation ϱ_t between the two Wiener processes can be made constant and calibrated from data.

2.3.2 Jump amplitudes

The previous section was concerned about the influence from S_t to P_t. Our interest now is the way back: how the jump amplitudes r_t^a and r_t^b behave near the conversion levels, i.e., how the premium affects the liquidity on the ETF. We assume that our dealer's limit orders presence (or activity) in the market does not affect or influence the intensity of the market orders arrival rate, i.e., dN_t^b and dN_t^a are independent of our dealer's choice of controls (δ_t^b, δ_t^a)[11].

Conditional distribution of jumps When trading at conversion levels, the additional liquidity brought by ETF arbitrageurs will lower the jump amplitudes r_t^a, when near or above creation level, and r_t^b, when near or below redemption level. This means the jump amplitudes distribution is conditional to the current state.

Hanson & Westman (2002) suggests using a log-uniform distribution to model the jump amplitudes for equities. In that case, they would follow a mixture distribution of log-uniforms where the weights are determined by P_t and the arbitrage levels P_{cheap} and P_{rich}, so that when P_t reaches these levels, the distribution is more weighed on lower returns.

A better and possibly more elegant alternative is assume that r_t^a and r_t^b both follow exponential distributions. This means that the aggregated jump process $dJ_t^a - dJ_t^b$ would follow an asymmetric Laplace distribution, and the ETF log-returns as we modeled would follow what is called a Normal-Laplace distribution, as described by Arnold et al. (2006):

$$r_t^a \sim \boldsymbol{Exp}(\Lambda^a)$$
$$r_t^b \sim \boldsymbol{Exp}(\Lambda^b)$$

A nice property of this approach, beyond being a better fit to data, is that exponential distributions are closed under scaling by a positive factor, so if we multiply either r_t^b or r_t^a by a positive number, the result will still follow an exponential distribution. This means we can use

[11] Although a large limit order at a competitive price may cause some liquidity takers to accelerate their execution rate, but we do not consider this in our work

a simple linear scaling to explain the conditional distribution and keep the same distribution family:

$$r_t^a \sim r^a(P_t)\boldsymbol{Exp}(\Lambda) = \boldsymbol{Exp}(\Lambda/r^a(P_t)) = \boldsymbol{Exp}(\Lambda/(1 - r\mathbb{1}_{P_t > P_{high}}))$$
$$r_t^b \sim r^b(P_t)\boldsymbol{Exp}(\Lambda) = \boldsymbol{Exp}(\Lambda/r^b(P_t)) = \boldsymbol{Exp}(\Lambda/(1 - r\mathbb{1}_{P_t < P_{low}}))$$

for some positive constants Λ and $0 < r < 1$, which can be calibrated by analyzing the distribution of the jump amplitudes of ETF returns conditioned on the regime of P_t with respect to P_{cheap} and P_{rich}.

Further improvements An even more sophisticated improvement that can be made and which we will not pursue in this work is that the dealer's quoting activity play a cushioning role to the impact of market orders. To illustrate this, recall that r_t^a is a positive random variable describing the positive jump in the ETF returns caused by an incoming market buy order. Obviously if we are quoting an offer, r_t^a will assume smaller values due to the cushioning effect, so the closer we are quoting to the reference price (smaller δ_t^a) and the bigger our offer size, the smaller the impact r_t^a will be. Similar reasoning follows for r_t^b with respect to our bid. Thus if $r_t^a > \delta_t^a$ or if $r_t^b > \delta_t^b$, then our dealer's quotes presence in the market will cushion the impact of incoming market orders, as an increasing function of $\eta_t^a = \eta(r_t^a, \delta_t^a)$ and $\eta_t^b = \eta(r_t^b, \delta_t^b)$, i.e., r_t^a and r_t^b follow a distribution that depends on δ_t^a and δ_t^b, respectively.

2.4 Hedging and Replication

Recent research (e.g. Lai & Lim (2003)) have suggested the use of singular perturbation and impulse control techniques market microstructure, thus pointing towards event-based and non-smooth policies for algorithmic trading[12]. In this spirit, it is clear that singular and impulse control approaches are highly appropriate for finance, with results enjoying greater applicability. In particular, Bruder & Pham (2009) uses impulse control with execution delay to understand the importance of latency and timely execution. It has also particular interesting applicability to option pricing, because it can account for discretization aspects (e.g. option gamma monetization, transaction costs, market impact) while automatically prescribing a (super)hedging strategy. Many other interesting applications of impulse control in finance can be found in Korn (1999).

Veraart (2010), Guilbaud & Pham (2011) and Guilbaud & Pham (2012) all go in this direction, and encouraged by their work, we propose the use of impulse control for the hedging activity of our ETF market maker. This approach not only allow us to easily account for intervention costs, but also to model the repercussion of the intervention back into the system,

[12]The event-based software development paradigm called "reactive programming" is extensively used in algorithmic trading, and thus thinking about impulse control is very relatable to practitioners.

e.g. hedging activity may negatively affect the remaining portfolio value (selling part of a long portfolio will depress the mark-to-market value of the remaining).

Impulse Control Singular and impulse controls both prescribe an intervention (perturbation) to the system at specific moments in time, in contrast to traditional deterministic or stochastic control, where the control is continuously active in the system. The main distinction between them is that in singular control, the intervention times are deterministic, while in impulse control, such intervention times are stopping times. An example of a singular control problem is a executing a TWAP order (time-weighted average price), where the intervention times are regularly spaced in time during the period of the execution, or the end of day liquidation of the inventory of a market maker. The delta hedging of an options portfolio is an impulse control problem, because trading the underlying is only necessary if the underlying market price moves, so the impulse is conditioned to a "trading rule", which in mathematical finance jargon is called a "stopping time".

In our work, we consider only impulse controls and focus on the perfect replication case, where the dealer is an authorized participant in the ETF in which it is providing liquidity and that has access to the underlying market, which we assume trades on the same currency as of the ETF, hence no foreign exchange risk. Let us denote by

(i) $0 < \tau_1 < \tau_2 < \cdots < \tau_i < \cdots$ a sequence of stopping times representing the impulse (hedging) times

(ii) $\xi_1, \xi_2, \cdots, \xi_i, \cdots$ a sequence of non-zero real valued random variables, where each ξ_i is $\mathcal{F}_{\tau_i}$-adapted, representing the hedging quantity in ETF shares (positive for buying, negative for selling)

(iii) $v_i = (\tau_i, \xi_i)$ the i-th impulse control, to be decided and instantaneously executed by the dealer at τ_i, when ξ_i underlying shares will be traded in the underlying market

(iv) $v = (v_1, v_2, \cdots, v_i, \cdots)$ is the impulse control policy, i.e. the hedging strategy to be followed

(v) $h_t = \sum_{\tau_i <= t} \xi_i$ the position in the underlying at time t

All hedging is performed by trading the underlying, whose reference price B_t is given by the simple defining relation of premium

$$P_t = S_t - B_t \Rightarrow B_t = S_t - P_t \tag{1}$$

where the premium here is considered in price unit instead of return unit.

By using the underlying portfolio to hedge, we are indirectly trading the premium, because being long (resp. short) on S_t and hedging that by selling (resp. buying) $S_t - P_t$ means replacing a risk exposure to S_t to a risk exposure on P_t. In other words, an ETF arbitrageur is essentially a market maker of the premium-discount.

Additional constraints are necessary to prevent the dealer's hedge from increasing an exposure, or from reverting the exposure (ex.: dealer long 100 ETF should not be allowed to sell more than 100 shares of the underlying):

$$\begin{cases} 0 \geq \xi_i \geq -q_{\tau_i} & \text{if } q_{\tau_i} > 0 \\ 0 \leq \xi_i \leq q_{\tau_i} & \text{if } q_{\tau_i} < 0 \\ 0 = \xi_i & \text{if } q_{\tau_i} = 0 \end{cases} \tag{2}$$

Optimal stopping Another question that can be formulated is how to define the optimal stopping time T. Di Graziano (2014) and Leung & Zhang (2017) study trailing stops in optimal trading. This is an interesting area of research and further study would have to be done on the optimal calibration of stopping rules, like the trailing stop, to data.

Proxy hedging The premium-discount is just a market conceptual construct to study the cointegration between exchange-traded funds and their underlying baskets, and it does not exist in the real world, therefore does not play any role in proxy-hedging, which is necessary when perfect replication is not possible or not practical. Typical approaches under imperfect replication involve the use of:

(a) a market-broad index futures contract like ES futures[13]

(b) a portfolio of futures contracts reasonably correlated to the ETF

(c) an optimized basket restricted to a reduced subset of ETF constituents

(d) another more liquid and closely related ETF

The interested reader will find a more comprehensive coverage on the usage of futures for proxy-hedging in (Alexander 2008, c.III.2), as well as in the blog post *Proxy / Cross Hedging* (2011)[14].

A future work can cover the case of ETF market making under imperfect replication, by abolishing the existence of a market for the premium P_t and instead considering the existence of a portfolio of assets with some dependence structure related to S_t.

[13] E-mini S&P 500 futures trade on CME and are among the most liquid futures available

[14] No author information available

2.5 Wealth dynamics

The wealth process can be described by a function of the cash Y_t, price S_t, premium P_t and inventories Q_t and H_t:

$$v(y, s, p, q, h) = y + qs + h(s - p) \tag{3}$$

thus we only need dynamical description of these processes.

2.5.1 Cash process

We denote by $(Y_t^{(u)})_{t\geq 0}$ the u-controlled process representing the amount of cash of the ETF market maker, which has the following dynamics in absence of any impulse control:

$$dY_t^{(u)} = S_t^a dQ_t^a - S_t^b dQ_t^b = S_t[\eta_t^a(1+\delta_t^a)dN_t^a - \eta_t^b(1-\delta_t^b)dN_t^b]$$

which is the amount of cash obtained from the market making activity during the instant from t to $t + dt$. When hedging is allowed, the impulses take effect only at τ_i and thus we have the w-controlled process $(Y_t^{(w)})_{t\geq 0}$

$$\begin{aligned} Y_t^{(w)} &= Y_t^{(u)} && \forall t \neq \tau_i \quad i = 1, 2, \cdots \\ Y_{\tau_i}^{(w)} &= Y_{\tau_i}^{(u)} - B_{\tau_i}\xi_i - |\xi_i| S_{\tau_i}\chi_{\tau_i} && \forall i = 1, 2, \cdots \end{aligned}$$

where $w = (u; v)$ is the combined stochastic and impulse control, and the term $B_{\tau_i}\xi_i - |\xi_i|S_{\tau_i}\chi_t$ refers to the amount of cash made or lost on the hedging activity. Note that hedging is done by crossing the spread on the hedging portfolio, so the dealer will sell at the bid and buy at the offer. We assume the proportional cost χ_t already includes the bid-offer spread of the underlying and is given in terms of ETF returns, hence $S_{\tau_i}\chi_{\tau_i}$ is the actual dollar cost of trading one share of the underlying portfolio.

2.5.2 Inventory process

Let us denote by $(Q_t^{(u)})_{t\geq 0}$ the u-controlled ETF inventory process, representing the quantity of ETF shares in the dealers inventory, whose mark-to-market value is defined as $Q_t S_t$ and whose dynamics are simply described by

$$\begin{aligned} Q_t &= Q_t^b - Q_t^a \\ dQ_t &= \eta_t^b dN_t^b - \eta_t^a dN_t^a \\ &= \eta(r_t^b, \delta_t^b)dN_t^b - \eta(r_t^a, \delta_t^a)dN_t^a \end{aligned}$$

We also denote by $(H_t^{(v)})_{t\geq 0}$ the v-controlled underlying inventory process, whose mark-to-market value is $H_t B_t = H_t(S_t - P_t)$, is simply described by netting all hedging activity from

the impulse control v:

$$H_t^{(v)} = \sum_{\tau_i \leq t} \xi_i$$

$$dH_{\tau_i}^{(v)} = \xi_i$$

3 Optimal Control of Jump-Diffusions

Control theory has seen a large number of successful applications in many areas of engineering and finance. Controllable dynamical systems are those which humans can apply control, like a force or energy, to exert influence on the trajectory, performance or costs of the said system. Driving a car is an example of a controllable dynamic system - the driver is able to control its trajectory and even optimize its efficiency by the proper use of the available controls like the wheel, gears and brakes. Financial systems can also be considered controllable systems - the act of sending limit or market orders is the investor's control, whose objective is to maximize his wealth or minimize his risk, given specific criteria regarding the wealth's trajectory. All the works so far mentioned in the field of optimal execution and market making are classical applications of stochastic control theory, a branch of control theory dedicated to the study and control of stochastic processes. Some of the first works on stochastic control of financial systems are Samuelson (1969), Merton (1969, 1971), which extend Markowitz (1952) modern portfolio theory to a multi-period setting.

The market maker's problem is essentially a control problem, where the dealer must make decisions at each time t regarding the price and quantity of its bid and offer quotes. Regarding passage of time, in this work $t = 0$ is considered the start of the first day, $t = 1^-$ is the end of the first day, $t = 1$ is the start of the second day, and so on. In practice, we would fix our final time $T = 1$, and solve the market maker problem from market open from one day to the next. From 0 until T our parameters remain constants, similarly to Ho & Stoll (1981), Avellaneda & Stoikov (2008) and Mudchanatongsuk et al. (2008). Thus at the beginning of every day, we must estimate the model parameters and solve our optimal control problem to find the optimal trading strategy, which would be considered optimal[15] for that day only. Alternatively, we could consider the problem on a higher frequency[16], and recalibrate every hour, for example.

We also consider that the dealer is allowed to trade only under strict risk constraints, so if any of the risk limits (constraints) is met, then all trading must stop. Mathematically, this is represented by a stopping time $\tau_{\mathcal{S}} = \inf\{t : X_t \notin \mathcal{S}\}$ where $\mathcal{S}$ is the set of all states that respect the risk limits. We also define $\mathcal{X}$ as the whole state space, thus $\mathcal{S} \subset \mathcal{X}$ and $\mathcal{X} \setminus \mathcal{S}$ is the set of all restricted states.

3.1 Dynamic Programming

The term Dynamic Programming was coined by Bellman (1957) to refer to the mathematical optimization "theory of multi-stage decision processes"

[15] Such strategy is optimal conditioned on the assumptions and the quality of the parameter estimations.

[16] Ideally we should pick a time window for which our estimation procedure is reliable enough.

(Bellman 1957, p.viii). This theory aims to solve problems that display what is called an "optimality substructure", i.e., those that can be broken up in two smaller subproblems whose optimal solutions can be composed to form an optimal solution to the bigger problem. Such class of problems are thereby referred to as *Dynamic Programs*. For Bellman, a *policy* is a sequential decision making rule, i.e., a function that prescribes an action to take given the state of the problem at a specific moment in time. An *optimal policy* is then the best course of action for a decision maker to achieve the desired objectives. If a problem displays the "optimality substructure", then the optimal policy is said to follow the *"Principle of Optimality"*, which is stated below:

Principle of Optimality *An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision (Bellman 1957, chapter 3, page 83).*

Consider the following discrete-time dynamic system

$$x_{t+1} = f_t(x_t, u_t, w_t)$$

where x_t is a controllable process, u_t is a control (or policy), w_t is some random disturbance and f_t is some *state transition function*. The *value function* $V^{(u)}(x_t, t)$ is defined as the value of the state x_t according to the policy $u = u(x_t, t)$, i.e., the total performance (or reward) of following such policy starting at the state x_t from the current time t until some future time T.

$$V^{(u)}(x_t, t) = \mathbb{E}_t \left[\sum_{i=t,\cdots,T-1} r_i(x_i, u_i, w_i) + r_T(x_T, w_T) \right]$$

where the functions r_i are called performance or reward functions. The controller, or agent, wants to find an optimal policy $u^* = \{u_t^*; t = 0, 1, \cdots, T\}$, i.e., one that maximizes the value function V:

$$V^{(u^*)}(x_t, t) = \max_u V^{(u)}(x_t, t)$$

The function $V^* = V^{(u^*)}$ is said to be the *optimal value function* of the dynamic program. The principle of optimality can be expressed in recursive form by what is called the Bellman equation:

$$V^{(u^*)}(x_t, t) = \max_{u_t} \mathbb{E}_t \left[r_t(x_t, u_t, w_t) + V^{(u^*)}(x_{t+1}, t+1) \right] \tag{4}$$

The optimal control u_t for this last problem, followed by the optimal policy for the rest of the problem (from $t+1$ to T) is essentially the optimal policy for the whole problem. Consecutive applications of this recursive principle means we can break up the problem into tail subproblems whose optimal solutions are all part of the original problem. The same principle

is also applicable to infinite horizon problems, when T is infinity, assuming that the future total reward series $\sum_{i=t,\cdots,T} r_i(x_i, u_i, w_i)$ converges when $T \to \infty$.

3.1.1 Hamilton-Jacobi-Bellman equations

The Hamilton-Jacobi-Bellman PDE equations are essentially a continuous-time version of the Bellman equation (4). Here we provide an intuitive and informal derivation of the HJB PDE equations similar to one presented in (Prigent 2007, section 6.1.3).

Finite-horizon case Let's first consider the following continuous-time version of the finite-horizon dynamic program:

$$V^{u^*}(x,t) = \max_{u \in \mathcal{U}} \mathbb{E}\left[\int_t^T dR(X_s, u_s, s) + r_f(X_T, T) | X_t = x\right] \tag{$\mathcal{P}1$}$$

where dR is the instantaneous reward (or running performance) function, r_f is the final reward function and X is a jump-diffusion process controlled by the stochastic control u. No impulse control is considered at this moment.

If we apply the stochastic Bellman's principle of optimality to $\mathcal{P}1$, we obtain the following expression:

$$V^{u^*}(x_t,t) = \max_{u \in \mathcal{U}(t,t+\epsilon]} \mathbb{E}_t\left[\int_t^{t+\epsilon} dR(X_s, u_s, s) + V^{u^*}(X_{t+\epsilon}, t+\epsilon) | X_t = x_t\right]$$

for any $\epsilon \in (0, T-t]$. The integral term can be rewritten as $R_{t+\epsilon} - R_t$ as follows

$$V^{u^*}(x_t,t) = \max_{u \in \mathcal{U}(t,t+\epsilon]} \mathbb{E}_t\left[R_{t+\epsilon} - R_t + V^{u^*}(X_{t+\epsilon}, t+\epsilon) | X_t = x_t\right]$$

Then, by moving the left term $V^{u^*}(x_t,t)$ inside the expectation and dividing by ϵ:

$$0 = \max_{u \in \mathcal{U}(t,t+\epsilon]} \mathbb{E}_t\left[\frac{1}{\epsilon}\left(R_{t+\epsilon} - R_t + V(X_{t+\epsilon}, t+\epsilon) - V(x_t,t)\right) | X_t = x_t\right] \tag{5}$$

Taking the limit of the above when $\epsilon \to 0^+$:

$$\lim_{\epsilon \downarrow 0} \mathbb{E}_t\left[\frac{1}{\epsilon}\left(R(X_{t+\epsilon}, t+\epsilon) - R(X_t,t)\right) | X_t = x_t\right] = \mathcal{A}R(x_t,t)$$
$$\lim_{\epsilon \downarrow 0} \mathbb{E}_t\left[\frac{1}{\epsilon}\left(V(X_{t+\epsilon}, t+\epsilon) - V(x_t,t)\right) | X_t = x_t\right] = \mathcal{A}V(x_t,t)$$

where $\mathcal{A}$ is the infinitesimal generator of the process X, also known as the Dynkin operator. Applying those results back into equation (5) while taking the same limit $\epsilon \to 0^+$ yields:

$$0 = \max_{u_t}\left[\mathcal{A}V(x_t,t) + \mathcal{A}R(x_t,t)\right] \tag{6}$$

which is the HJB equation, and must be satisfied for all combinations of t and x_t, subject to the final boundary condition $V(x,T) = r_f(x,T)$. Observe here that R is given, so we must solve such equation for V.

Infinite-horizon case Let now consider the discounted infinite-horizon version of the same problem:

$$V^{u^*}(x,t) = \max_{u\in\mathcal{U}} \mathbb{E}\left[\int_t^\infty e^{-\rho(s-t)} dR(X_s, u_s, s) | X_t = x\right] \tag{$\mathcal{P}2$}$$

where we are only given the reward process R and a constant discount rate ρ. We can again apply the principle of optimality and have the following Bellman equation:

$$V^{u^*}(x,t) = \max_{u\in\mathcal{U}} \mathbb{E}\left[\int_t^{t+\epsilon} e^{-\rho(s-t)} dR(X_s, u_s, s) + e^{-\rho\epsilon} V^{u^*}(X_{t+\epsilon}, t+\epsilon) | X_t = x\right] \tag{7}$$

By applying Ito formula on the function $f(R,t) = e^{-\rho t}R$ (which ends up being an application of integration by parts since $e^{-\rho(s-t)}$ is a deterministic function), we can establish

$$\int_t^{t+\epsilon} e^{-\rho(s-t)} dR(X_s, u_s, s) = e^{-\rho\epsilon} R_{t+\epsilon} - R_t + \int_t^{t+\epsilon} \rho e^{-\rho(s-t)} R_s ds$$

Substituting that back into the Bellman equation (7) and subtracting V^{u^*} from both sides, we have

$$\begin{aligned} 0 = \max_{u\in\mathcal{U}} \mathbb{E}\Bigg[& \int_t^{t+\epsilon} \rho e^{-\rho(s-t)} R_s ds + e^{-\rho\epsilon} R_{t+\epsilon} - R_t \\ & + e^{-\rho\epsilon} V^{u^*}(X_{t+\epsilon}, t+\epsilon) - V^{u^*}(x,t) | X_t = x\Bigg] \end{aligned} \tag{8}$$

Multiplying that by $e^{\rho\epsilon}/\epsilon$ and taking the limit as $\epsilon \to 0^+$, we obtain

$$\lim_{\epsilon\downarrow 0} \mathbb{E}_t\left[\frac{1}{\epsilon}\left(\int_t^{t+\epsilon} \rho e^{-\rho(s-t)} R_s ds\right) | X_t = x_t\right] = \rho R$$

$$\lim_{\epsilon\downarrow 0} \mathbb{E}_t\left[\frac{1}{\epsilon}\left(R(X_{t+\epsilon}, t+\epsilon) - R(X_t,t) + (1-e^{\rho\epsilon}) R(X_t,t)\right) | X_t = x_t\right] = \mathcal{A}R - \rho R$$

$$\lim_{\epsilon\downarrow 0} \mathbb{E}_t\left[\frac{1}{\epsilon}\left(V(X_{t+\epsilon}, t+\epsilon) - V(x_t,t) + (1-e^{\rho\epsilon}) V(X_t,t)\right) | X_t = x_t\right] = \mathcal{A}V - \rho V$$

Putting everything together back into (8), we finally get

$$0 = \max_{u_t} \left[\mathcal{A}R(x_t,t) + \mathcal{A}V(x_t,t) - \rho V(X_t,t)\right]$$

which is the HJB equation for the infinite horizon case and must hold for all t and x_t, but now without any boundary condition. For a more complete and formal statement and proof of the HJB equations for Optimal Control of Jump-Diffusions, we refer to (Øksendal & Sulem 2007, p.46-47 thm 3.1)

3.1.2 Hamilton-Jacobi-Bellman quasi-variational inequalities

Impulse control problems can also be approached by dynamic programming, but this does not lead to an HJB equation anymore - instead we obtain what are called HJB quasi-variational inequalities. Quasi-Variational Inequality Problems (QVI) are a generic class of problems for

which, given a function $F : \mathbb{R}^n \to \mathbb{R}^n$ and a point to set mapping $K : \mathbb{R}^n \to \mathbb{R}^n$ with closed convex images, the objective is to find for a point $x^* \in K(x^*)$ such that

$$F(x^*) \cdot (x - x^*) \geq 0 \quad \forall x \in K(x^*) \tag{9}$$

Traditional convex optimization problems can be written as variational inequality problems (special case of QVI where the mapping K is constant). For example, if the objective is to minimize a given convex objective function $F \in \mathcal{C}^2(\mathbb{R}^n)$ over a convex set $K \subset \mathbb{R}^n$, then the problem is find x^* for the first-order condition (here expressed as a variational inequality):

$$\nabla F(x^*) \cdot (x - x^*) \geq 0 \quad \forall x \in K$$

Quasi-Variational Inequalities can also be defined on function spaces embedded with norm and inner products (a Hilbert space). In this case, F would be a functional (function of functions) and K would be a closed convex set of functionals, and the objective would be to find a function x^* that solves (9). This is precisely the case for the HJB quasi-variational inequalities, introduced by Bensoussan & Lions, where the objective is to find the optimal value function V and its related optimal control function for the given HJBQVI. In the case of impulse control, since the problem is dynamic, we must solve one HJBQVI for each time t. A complete coverage of impulse control and related HJBQVI are Bensoussan & Lions (1984), Øksendal & Sulem (2007). On the general topic of QVI, we refer to Antipin et al. (2018), Facchinei et al. (2014).

Like in section 3.1.1, the principle of optimality allow us to restrict on looking for optimal control for the period $(t, t + \epsilon]$ of the combined stochastic and impulse control problem, which entails finding a combination of optimal stochastic control u and optimal impulse control v.

At each instant $\mathfrak{t} \in (t, t + \epsilon]$ we have the choice of applying or not the impulse impulse $(\mathfrak{t}, \xi)$, and so we must make the optimal choice by checking if intervening will lead us to a better value-to-go $V^{(w)}(X_{t+\epsilon}, t + \epsilon)$ or not. The intervention operator $\mathcal{M}$ on V is defined such that $\mathcal{M}V(x, \mathfrak{t})$ is the best resulting value possible of V by applying a non-zero impulse ξ at time $\mathfrak{t}$ to the system X. The value $\mathcal{M}V(x, \mathfrak{t}) - V(x, \mathfrak{t})$ then must be compared to what would happen if we just applied the optimal control u^*. If $\mathcal{M}V - V > \mathcal{A}R + \mathcal{A}V - \rho V$ for the optimal u^*, then applying impulse is optimal, otherwise, we are better off by not intervening until the next instant time $\mathfrak{t} + dt$, falling back to the no impulse case where the HJB equation must hold. Thus, when $\epsilon \downarrow 0$, the last section equations become:

$$0 = \max_{w_t \in \mathcal{W}} \left[\sup_{u \in \mathcal{U}} \{\mathcal{A}R + \mathcal{A}V - \rho V\}, \mathcal{M}V - V \right] \tag{10}$$

which are known as the *Hamilton-Jacobi-Bellman Quasi-Variational Inequalities*, and must be satisfied for all x and t. In the finite-horizon case, $\rho = 0$ and the boundary condition $V(\cdot, T) = J(\cdot, T)$ must be satisfied. The intervention operator $\mathcal{M}$ is defined as

$$\mathcal{M}V(x) = \sup\{V(\Gamma(x, \xi)) + K(x, \xi); \xi \in \Xi \text{ and } \Gamma(x, \xi) \in \mathcal{S}\} \tag{11}$$

where $\Gamma : \mathcal{X} \times \Xi \to \mathcal{X}$ is a function that maps a state x and a corresponding impulse ξ to a new state resultant from the application of such impulse, and K is the utility benefit function (reward) of applying the impulse ξ while on state x.

3.2 Problem formulation

We now formulate the ETF Market maker problem as a combined stochastic and impulse control problem. We start defining the controlled jump-diffusion state process $X_t \in \mathcal{X} = \mathbb{R}^5$

$$X_t = X_t^{(w)} = \begin{bmatrix} Y_t^{(w)} & S_t & P_t & Q_t & H_t \end{bmatrix}'$$

which depends on the combined stochastic and impulse control process $w = (u; v) = w_t(\omega) : [0, \infty) \times \Omega \to \mathcal{W}$:

$$w_t = (\begin{bmatrix} \delta_t^b & \delta_t^a \end{bmatrix}'; (\tau_i, \xi_i)_{i=1,2,\cdots})$$

where $\mathcal{W} = \mathcal{U} \times \mathcal{V}$ is the set of admissible combined controls, and transaction costs are considered constant $\chi_t = \chi$. The state dynamics can be described by

$$dX_t = dX_t^{(u)} = \begin{bmatrix} dY_t^{(u)} & dS_t & dP_t & dQ_t & dH_t \end{bmatrix}'$$

which can also be expressed in the following way

$$dX_t = \mu(X_t, w_t, t)dt + \sigma(X_t, w_t, t)dW_t + \gamma(X_t, w_t, t)dN_t$$

where

(i) dW_t describes the uncorrelated 2-dimensional Wiener process $(W_t)_{t \geq 0}$

(ii) dN_t describes the 4-dimensional Poisson process $(N_t)_{t \geq 0}$

(iii) $\mu(X_t, w_t, t) : \mathbb{R}^5 \times \mathcal{W} \times \mathbb{R}^+ \to \mathbb{R}^5$ is the drift function

$$\begin{bmatrix} 0 \\ 0 \\ \theta(\mathfrak{p} S_{t^-} - P_{t^-}) \\ 0 \\ 0 \end{bmatrix} \tag{12}$$

(iv) $\sigma(X_t, w_t, t) : \mathbb{R}^5 \times \mathcal{W} \times \mathbb{R}^+ \to \mathbb{R}^{5 \times 2}$ is the diffusion function

$$\begin{bmatrix} 0 & 0 \\ S_{t^-}\sigma\sqrt{1-\varrho^2} & S_{t^-}\sigma\varrho \\ 0 & S_{t^-}\kappa \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{13}$$

(v) $\gamma(X_t, w_t, t) : \mathbb{R}^5 \times \mathcal{W} \times \mathbb{R}^+ \to \mathbb{R}^{5\times 4}$ is the jump amplitude function

$$\begin{bmatrix} S_{t^-}\eta_t^a(1+\delta_t^a) & -S_{t^-}\eta_t^b(1-\delta_t^b) & 0 & 0 \\ S_{t^-}r_t^a & -S_{t^-}r_t^b & 0 & 0 \\ S_{t^-}r_t^a & -S_{t^-}r_t^b & \mathbb{1}^{cheap}S_{t^-}r_t^{ap} & -\mathbb{1}^{rich}S_{t^-}r_t^{bp} \\ -\eta_t^a & \eta_t^b & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{14}$$

We will denote by μ_i the i-th component of μ, and likewise for σ and γ, therefore μ_1 would be the drift function in the dY_t expression we obtained earlier, μ_2 would be the drift function in the dS_t expression, and so on.

If an impulse is made, then the state is impacted by the following function Γ

$$\Gamma(X_t, \xi) = X_t + \begin{bmatrix} -(S_t - P_t)\xi - |\xi|S_t\chi_t \\ 0 \\ 0 \\ 0 \\ \xi \end{bmatrix} \tag{15}$$

which implies that the hedging does not affect neither the ETF price or the premium.

While many formulations to the problem are possible, here we illustrate two classes of control problems: finite horizon, where the objective is to optimize the total reward (or cost) over a finite time interval, and infinite horizon ones, where the objective is to optimize the total future reward (or cost).

Finite horizon problem In the finite horizon formulation, our ETF dealer's objective is to maximize the conditional expectation of a risk-averse utility functional $U[X, w, T](x_t, t)$, risk constrained to the state space $\mathcal{S}$, evaluated at the final time T:

$$V^{(w^*)}(x_t, t) = \max_{w\in\mathcal{W}_{(t,T]}} \mathbb{E}_t\left[U[X, w, T](x_t, t)|X_t = x_t\right] \tag{$\mathcal{PF}$}$$
$$x_t = \begin{bmatrix} y & s & p & q & h \end{bmatrix}'$$

where $Y_t = y$ is the initial wealth, $S_t = s$ is the initial ETF price, $P_t = p$ is the initial premium-discount, $q_t = q$ is our initial position in the ETF, $H_t = h$ is the initial position in the underlying and t is the initial time. The function V^{w^*}, the optimal value function, is the solution to the dynamic programming problem stated above, from time t until T. The optimal control w^* is

$$w^* = \operatorname*{argmax}_{w\in\mathcal{W}(t,T]} \mathbb{E}_t\left[U[X, w, T](x_t, t)\right]$$

i.e., $V^{(w^*)}(x, t) = \mathbb{E}_t\, U[X, w^*, T](x, t)$ for all $x \in \mathcal{S}$ and $t \in [0, T]$.

Infinite Horizon problem One can argue that an ETF market maker desires be making ETF markets forever (or at least without a set deadline). Hence, it makes sense to formulate the infinite horizon dealer's problem, where the objective is to maximize the conditional expectation of the present value of all future rewards:

$$V^{(w^*)}(x_t,t) = \max_{w\in\mathcal{W}_{(t,T]}} \mathbb{E}_t\left[\int_t^\infty e^{-\rho(s-t)} dR(X_s,w_s,s)|X_t=x_t\right] \qquad (\mathcal{PI})$$
$$x_t = \begin{bmatrix} y & s & p & q & h \end{bmatrix}'$$

where dR stands for the dynamics of the reward process $(R_t)_{t\geq 0}$ is defined by $R_t = R(X_t,w_t,t) = v(X_t) - \psi(X_t)$, the wealth[17] value $v(X_t)$ penalized by the risk $\psi(X_t)$ incurred on the state X_t. If seen this way, the reward function R coincides with the utility function U.

3.2.1 HJBQVI equations

We now proceed to derive the exact expression for the HJBQVI equations for both formulations of the ETF dealer's problem, as they both share the same mathematical 'ingredients'. First we state the infinitesimal generator formula, considering constant intensities for the Poisson processes.

Proposition 1. *The infinitesimal generator $\mathcal{A}V$ for the optimal value function $V^{(u^*)}$ where $u^* = u^*(x)$ is a markov control is given by*

$$\mathcal{A}V(x) = \sum_{i=1}^5 \mu_i(x,u(x))\frac{\partial V}{\partial x_i}(x) + \frac{1}{2}\sum_{i,j=1}^5 (\sigma\sigma')_{ij}(x,u(x))\frac{\partial^2 V}{\partial x_i \partial x_j}(x)$$
$$+ \sum_{j\in\{a,b,ap,bp\}} \left\{\mathbb{E}\left(V(x+\gamma^{(j)}(x,u(x))) - V(x)\right)\right\}\lambda_j$$

where $\gamma^{(j)}$ is the column of γ relative to the Poisson process N^j (i.e. for $j=a$ (alt. b, ap, bp) the jump-amplitude function relative to dN_t^a (alt. $dN_t^b, dN_t^{ap}, dN_t^{bp}$)

Proof. Application of theorem 1.22 page 11 from Øksendal & Sulem (2007) □

Based on equation (13), we calculate $\sigma\sigma'$:

$$(\sigma\sigma')(X_t,u_t,t) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & S_{t^-}^2\sigma^2 & S_{t^-}^2\varrho\sigma\kappa & 0 & 0 \\ 0 & S_{t^-}^2\varrho\sigma\kappa & S_{t^-}^2\kappa^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

[17] Recall wealth function definition (3)

and applying the proposition 1 (with equations (12), and (14)) we have the expression for $\mathcal{A}V(x) = \mathcal{A}V(y, s, p, q)$

$$\begin{aligned}\mathcal{A}V(y,s,p,q,h,t) = V_t + \theta(\mathfrak{p}s - p)V_p + \frac{1}{2}\left\{s^2\sigma^2 V_{ss} + s^2\kappa^2 V_{pp} + 2s^2\varrho\sigma\kappa V_{sp}\right\} \\ + \lambda_a\,\mathbb{E}[\Delta_a V] + \lambda_b\,\mathbb{E}[\Delta_b V] + \lambda_{ap}\,\mathbb{E}[\Delta_{ap} V] + \lambda_{bp}\,\mathbb{E}[\Delta_{bp} V]\end{aligned} \tag{16}$$

where $\mathfrak{p}$ is the fair value for the premium, σ now is the constant volatility parameter and $\Delta_j V$ stands for the change in V cause by a jump dN^j:

$$\begin{aligned}\Delta_a V &= V(y + s\eta_t^a(1+\delta_t^a), s + sr_t^a, p + sr_t^a, q - \eta_t^a, h) - V(y,s,p,q,h)\\ \Delta_b V &= V(y - s\eta_t^b(1-\delta_t^b), s - sr_t^b, p - sr_t^b, q + \eta_t^b, h) - V(y,s,p,q,h)\\ \Delta_{ap} V &= V(y, s, p + \mathbb{1}^{cheap} sr_t^{ap}, q, h) - V(y,s,p,q,h)\\ \Delta_{bp} V &= V(y, s, p - \mathbb{1}^{rich} sr_t^{bp}, q, h) - V(y,s,p,q,h)\end{aligned}$$

where $\mathbb{1}^{cheap} = \mathbb{1}_{P_t < P_{cheap}}$ and $\mathbb{1}^{rich} = \mathbb{1}_{P_t > P_{rich}}$ are measurable functions of the state X_t, and $\eta_t^b = \eta(r_t^b, \delta_t^b)$ and $\eta_t^a = \eta(r_t^a, \delta_t^a)$ are "controlled" random variables (functions of a random variable and the control u).

Now we use equation (6) to find the final expression for the HJB equations of the problem $(\mathcal{PF})$:

$$\begin{aligned}0 \geq V_t + \theta(\mathfrak{p}s - p)V_p + \frac{1}{2}\left\{s^2\sigma^2 V_{ss} + s^2\kappa^2 V_{pp} + 2s^2\varrho\sigma\kappa V_{sp}\right\} \\ + \lambda_a \max_{u\in\mathcal{U}} \mathbb{E}[\Delta_a V] + \lambda_b \max_{u\in\mathcal{U}} \mathbb{E}[\Delta_b V] + \lambda_{ap}\,\mathbb{E}[\Delta_{ap} V] + \lambda_{bp}\,\mathbb{E}[\Delta_{bp} V]\end{aligned}$$

with the boundary condition $V(X_T, u_T, T) = U[X, u, T](X_T, T)$, equality holding if no impulse is allowed. The expression for $\mathcal{M}V$ based on equation (11) and Γ in (15) is:

$$\mathcal{M}V(x) = \sup\{V(\Gamma(x,\xi)); \xi \in \Xi \text{ and } \Gamma(x,\xi) \in \mathcal{S}\}$$

where we are restricted to impulses that bring the new state to the valid states set S.

HJB quasi-variational inequalities for $(\mathcal{PF})$ By applying (10) we get the final expression for the quasi-variational inequalities for the finite horizon problem

$$\begin{aligned}0 = \max_{w_t \mathcal{W}} \Bigg[V_t + \theta(\mathfrak{p}s - p)V_p + \frac{1}{2}\left\{s^2\sigma^2 V_{ss} + s^2\kappa^2 V_{pp} + 2s^2\varrho\sigma\kappa V_{sp}\right\} \\ + \lambda_a \max_{u\in\mathcal{U}} \mathbb{E}[\Delta_a V] + \lambda_b \max_{u\in\mathcal{U}} \mathbb{E}[\Delta_b V] \\ + \lambda_{ap}\,\mathbb{E}[\Delta_{ap} V] + \lambda_{bp}\,\mathbb{E}[\Delta_{bp} V], \\ \max_{\xi\in\Xi;\Gamma(x,\xi)\in\mathcal{S}} (V(\Gamma(x,\xi)) - V) \Bigg]\end{aligned}$$

where Ξ must satisfy conditions (2) and $V(\cdot, T) = U(\cdot, T)$.

HJB quasi-variational inequalities for $(\mathcal{PI})$ The HJB equation for the infinite horizon problem can be derived by assuming that the reward process R coincides with the process generated by the utility function U evaluated at each instant in time, i.e.,

$$\int_t^T dR(X_s, w_s, s) = U[X, w, T](x_t, t) - U[X, w, t](x_t, t)$$

Thus, the HJBQVI for the infinite horizon problem can be expressed as

$$0 = \max_{w_t \in \mathcal{W}} \left[\sup_{u_t \in \mathcal{U}} [\mathcal{A}U + \mathcal{A}V - \rho V], \max_{\xi \in \Xi; \Gamma(x,\xi) \in \mathcal{S}} (V(\Gamma(x, \xi)) - V) \right]$$

where $\mathcal{A}V$ is defined in (16), and $\mathcal{A}U$ similarly defined but for a given utility function U.

3.3 Problem Extensions

Intraday liquidity patterns It is typical for equity markets, as demonstrated in Brooks et al. (2003) for NYSE stocks and Ivanov (2017) for 18 ETFs, to exhibit a J-shaped pattern for the spread and the volume. In addition, Iwatsubo et al. (2017) also investigates intraday liquidity for commodities markets, identifying relevant influences from exchanges that trade in different timezones. International and Global ETFs also display this same characteristic, suggesting that we should pay special attention to this aspect.

Hence, in the case we want study the problem where the spread χ_t of the underlying is not constant, the problem can be trivially extended by adding χ_t to the state $X_t \in \mathcal{X} = \mathbb{R}^6$:

$$X_t = X_t^{(w)} = \begin{bmatrix} Y_t^{(w)} & S_t & P_t & Q_t & H_t & \chi_t \end{bmatrix}'$$
$$x_t = \begin{bmatrix} y & s & p & q & h & \chi_t \end{bmatrix}'$$

The transaction cost χ_t can be considered a stochastic process itself (which can be estimated by online filtering) or be considered a deterministic function of time (possibly in $\mathcal{C}^{1,2}$, piecewise continuous or even a table lookup). By solving the optimal control under different scenarios for χ_t, it is then possible to understand how different intraday patterns for liquidity affects our dealer.

Cost of carry and Funding The infinite horizon problem can be further enhanced to account for the cost of carry by considering checkpoints at $t = 1, 2, \cdots$ when the trader must pay or receive the discounted carry for the next (or previous) day. The optimization would be let to run until $\tau_{\mathcal{S}}$, so as long as the dealer does not breach any risk limit, it will be allowed to trade. Either a stochastic or a deterministic model for the carry cost and funding could be used.

Additionally, the dealer could be allowed to use singular controls at each of those checkpoints $t = 1, 2, 3, \cdots$ to reduce its exposure to both the ETF and the underlying in order to avoid carry a position that is hard to finance.

If a futures contract is available on the same index of the ETF or a very similar one, an impulse control for hedging using futures could improve the carry while simultaneously hedging the dealers position.

Choice of utility In economics, the *utility function* $U(.)$ models the investor preferences towards the reward-risk tradeoff. A risk-neutral dealer will only care a about maximizing the expected value, so in that case $U[X, u, T](x, t) = v(x)$, indifferent to any risk resultant from any remaining portfolio at time T, thus not displaying any of the usual asymmetrical risk-aversion to losses. On the extreme spectrum, an infinitely (or extremely) risk averse dealer will behave like a market-neutral market maker, and will try its best to immediately realize any arbitrage opportunity without incurring in inventory risk.

The literature is diverse regarding the possible choices available for the utility functional U:

(a) Mean variance utility (as in Ho & Stoll (1981))

$$U[X, w, T](x, t) = v(Y_T^{(w)}, S_t, P_t, Q_t, H_t) - \frac{\lambda}{2} Var(v(Y_T^{(w)}, S_t, P_t, Q_t, H_t))$$

(b) Running quadratic utility (Veraart (2010))

$$U[X, w, \infty](x, t) = \int_0^\infty e^{-b\,s}(v(X_s) - \frac{\lambda}{2} \sum_{j \in \{1,4,5\}} \sigma_j^2(X_s, w, s))ds$$

(c) Exponential utility (as in Avellaneda & Stoikov (2008))

$$U[X, w, T](x, t) = -exp(-\gamma v(X_T^{(w)}))$$

(d) Power utility[18] (as in Mudchanatongsuk et al. (2008))

$$U[X, w, T](x, t) = \frac{1}{\gamma}(v(X_T^{(w)}))^\gamma$$

In sight of recent changes to the Fundamental Review of the Trading Book (FRTB) and its mandatory use of the Conditional Value-At-Risk (CVaR) for risk management of trading desks, we could imagine a Coherent Risk measure utility

$$U[X, w, T](x, t) = v(X_T^{(w)}) - \lambda ES_{T-t,\alpha}(v(X_T^{(w)}))$$

where $ES_{T-t,\alpha}$ is the Expected Shortfall, or Conditional Value-at-Risk, over a period of the time horizon from t to T at $\alpha\%$ level, i.e. the average loss in the worst $\alpha\%$ of the cases from t to T. In this direction, Miller & Yang (2015) investigates Optimal Control under coherent risk measures on portfolio optimization under CVaR constraints, providing an interesting alternative to the previously cited utility functions, which is more sensible for real-world trading applications, as also hinted by Bertsimas et al. (2004) and Gundel & Weber (2008).

[18] This function can only be applied to positive wealth processes

4 Algorithmic solutions

4.1 Backward induction

A traditional way to solve HJB PDEs is by backward induction, whereby we discretize the time and state spaces and solve the HJB PDEs numerically going backwards in time from T to current t. Infinite horizon problems can also be approached this way if we take a large enough T. The finite differences numerical scheme we suggest here is largely based on Guilbaud & Pham (2012) and Azimzadeh (2017). First let us assume a uniform discretization of the time interval $[0, T]$ into intervals of $\Delta t = T/N$:

$$\mathbb{T}_N = \{t_i = i\Delta t; i = 0, 1, \cdots, N\}$$

In the hedging problem, each component of the state space $\mathcal{X} = \mathbb{R}^5$ must be discretized:

(i) $\mathtt{S} = \{s_i = s + i\Delta s; i = -N^S, \cdots, -1, 0, 1, \cdots, N^S\}$ where Δs is the ETF tick size, and the bounds s_min and s_max cover at least 3 sigmas in price change for the given time horizon $[0, T]$.

(ii) $\mathtt{P} = \{p_i = p + i\Delta s; i = -N^P, \cdots, -1, 0, 1, \cdots, N^P\}$ where the bounds $p_{min} = p_{-N^P}$ and $p_{max} = p_{+N^P}$ for the ETF premium-discount cover the conversion levels

(iii) $\mathtt{Q} = \{q_i = q + i\Delta q; i = -N^Q, \cdots, -1, 0, 1, \cdots, N^Q\}$ where $\Delta q = 1$ is the minimum tradable lot size, and $q_{min} = q_{-N^Q}$ and $q_{max} = q_{+N^Q}$ are risk-limit bounds approved by the dealer's risk department

(iv) $\mathtt{H} = \{h_i = h + i\Delta q; i = -N^H, \cdots, -1, 0, 1, \cdots, N^H\}$ with bounds $h_{min} = h_{-N^H}$ and $h_{max} = h_{N^H}$ large enough as to be able to fully hedge the ETF position

(v) $\mathtt{Y} = \{y_i = y + i(\Delta s \Delta q); i = -N^Y, \cdots, -1, 0, 1, \cdots, N^Y\}$ is the discretization of the cash amount

(vi) $\mathtt{R} = \{r_i = i\Delta r; i = -N^R, \cdots, -1, 0, 1, \cdots, N^R\}$ is a sensible discretization of the random variables r_t^a, r_t^b, r_t^{ap} and r_t^{bp}, so Δr must be about the return generated by a move in 1 tick in the price

(vii) $\mathtt{U} = \{\delta_i^n = i\Delta s; i = -N^U, \cdots, -1, 0, 1, \cdots, N^U\}$ is the discretization of the stochastic control space

(viii) $\Xi = \{\xi_i = i\Delta q; i = 0, 1, \cdots, N^\Xi\}$ is the discretization of the impulse space representing the hedging quantity, which must respect any imposed maximum order size for a market order

(ix) $\mathtt{X} = \mathtt{Y} \times \mathtt{S} \times \mathtt{P} \times \mathtt{Q} \times \mathtt{H}$ is the whole discretized state space

(x) I is the index space for all elements in X, i.e., it is composed of all combinations of indexes $(i_y, i_s, i_p, i_q, i_h)$ such that $x_i = (y_{i_y}, s_{i_s}, p_{i_p}, q_{i_q}, h_{i_h})$

Define $V_i^n = V(x_i, t_n)$ the discrete value for the optimal value function, and by $V^n = \{V_i^n; i \in \mathtt{I}\}$ the discrete values for the value function at time t_n. The partial derivatives V_t, V_p, V_{ss} and V_{pp} are approximated using forward and central differences methods:

$$\begin{aligned}
\frac{\partial V}{\partial t}(x_i, t_j) &\approx \frac{V(x_i, t_{j+1}) - V(x_i, t_j)}{\Delta t} = \frac{V_i^{j+1} - V_i^j}{\Delta t} \\
\frac{\partial V}{\partial p}(x_i, t_j) &\approx \frac{V(x_{i+\mathbf{1}_p}, t_j) - V(x_i, t_j)}{\Delta p} \\
\frac{\partial^2 V}{\partial s^2}(x_i, t_j) &\approx \frac{V(x_{i+\mathbf{1}_s}, t_j) - 2V(x_i, t_j) + V(x_{i-\mathbf{1}_s}, t_j)}{\Delta s^2} \\
\frac{\partial^2 V}{\partial p^2}(x_i, t_j) &\approx \frac{V(x_{i+\mathbf{1}_p}, t_j) - 2V(x_i, t_j) + V(x_{i-\mathbf{1}_p}, t_j)}{\Delta s^2}
\end{aligned}$$

where $i \in \mathtt{I}$ and $\mathbf{1}_s = (0,1,0,0,0)$ and $\mathbf{1}_p = (0,0,1,0,0)$ are the unit vectors in I corresponding to the S and P directions. We thus define operators $\mathcal{D}_t, \mathcal{D}_p, \mathcal{D}_{ss}$ and $\mathcal{D}_{pp}$

$$\begin{aligned}
\mathcal{D}_t V^n &= \left\{ \frac{V_i^{n+1} - V_i^n}{\Delta t}; i \in \mathtt{I} \right\} \\
\mathcal{D}_p V^n &= \left\{ \frac{V_{i+\mathbf{1}_p}^n - V_i^n}{\Delta p}; i \in \mathtt{I} \right\} \\
\mathcal{D}_{ss} V^n &= \left\{ \frac{V_{i+\mathbf{1}_s}^n - 2V_i^n + V_{i-\mathbf{1}_s}^n}{\Delta s^2}; i \in \mathtt{I} \right\} \\
\mathcal{D}_{pp} V^n &= \left\{ \frac{V_{i+\mathbf{1}_p}^n - 2V_i^n + V_{i-\mathbf{1}_p}^n}{\Delta s^2}; i \in \mathtt{I} \right\}
\end{aligned}$$

Since we discretized the jump amplitude space as a finite set R, the expected values in $\nabla_k V$ for $k \in \{a, b, ap, bp\}$ can be calculated by weighted average of V:

$$\begin{aligned}
\nabla_a V(x_i, u_i, t_n) = \nabla_a V_i^n(\delta_i^{a,n}) &= \sum_{r \in \mathtt{R}} \mathbb{P}(r^a = r; x_i) V_{i+\mathbf{a},\mathbf{r},\delta_\mathbf{i}^{\mathbf{a},\mathbf{n}}}^n - V_i^n \\
\nabla_b V(x_i, u_i, t_n) = \nabla_b V_i^n(\delta_i^{b,n}) &= \sum_{r \in \mathtt{R}} \mathbb{P}(r^b = r; x_i) V_{i+\mathbf{b},\mathbf{r},\delta_\mathbf{i}^{\mathbf{b},\mathbf{n}}}^n - V_i^n \\
\nabla_a V(x_i, u_i, t_n) = \nabla_a V_i^n &= \sum_{r \in \mathtt{R}} \mathbb{P}(r^{ap} = r; x_i) V_{i+\mathbf{ap},\mathbf{r}}^n - V_i^n \\
\nabla_b V(x_i, u_i, t_n) = \nabla_b V_i^n &= \sum_{r \in \mathtt{R}} \mathbb{P}(r^{bp} = r; x_i) V_{i+\mathbf{bp},\mathbf{r}}^n - V_i^n
\end{aligned}$$

where $V_{i+\mathbf{k},\mathbf{r},\delta}^n$ is a multivariate interpolation approximation to V in the case the compounded process N^k jumps by r given the stochastic control δ. Jumps on the state variable will not necessarily make the index i land on the index space I, so we need to resort to an approximation. The probability function is conditional on the state x_i (because of principle 2.3.2) but it is deterministic and can be calculated easily.

The above are ingredients to approximate the HJB term of the HJBQVI by means of the

HJB equation operator $\mathcal{L}$

$$\begin{aligned}\mathcal{L}V^n = \{\mathcal{L}V_i^n = V_i^n + \mathcal{D}_t V_i^n + \frac{1}{2}s^2\sigma^2\mathcal{D}_{ss}V_i^n + \frac{1}{2}s^2\kappa^2\mathcal{D}_{pp}V_i^n \\ + \lambda_a \max_{\delta_i^{a,n}} \nabla_a V_i^n + \lambda_b \max_{\delta_i^{b,n}} \nabla_b V_i^n \\ + \lambda_{ap}\nabla_{ap}V_i^n + \lambda_{bp}\nabla_{bp}V_i^n; i \in \mathtt{I}\}\end{aligned}$$

while the intervention operator $\mathcal{M}$ can be approximated as

$$\mathcal{M}V^n = \{\mathcal{M}V_i^n = \max_{\xi\in\Xi^n} V^n_{\Gamma(i,\xi)}; i \in \mathtt{I}\}$$

where $V^n_{\Gamma(i,\xi)}$ is a multivariate interpolation approximation to V if an impulse ξ is applied when in state x_i, since it is not guaranteed that $\Gamma(i,\xi)$ will land on the index space (h will, but y may not).

An important remark to be made is is that we assume no extrapolation on all these operators, so if index is out of bounds, we use the nearest neighbor, e.g. $V_{(i_y,i_s,i_p,i_q,i_h)} = V_{(\hat{i}_y,\hat{i}_s,\hat{i}_p,\hat{i}_q,\hat{i}_h)}$ where $\hat{i}_k = -N_k \vee (i_k \wedge N_k)$ and so on. Lastly, we can define the operator $\mathcal{K}$ as

$$\mathcal{K}V^n = \{\max(\mathcal{L}V_i^n, \mathcal{M}V_i^n); i \in \mathtt{I}\}$$

and the backward induction algorithm is simply an application of the $\mathcal{K}$ operator backwards, which is the proposed algorithm by Guilbaud & Pham (2012).

Algorithm 1 Backward induction

for all $i \in \mathtt{I}$ **do**
  $V_i^N \leftarrow U(x_i, t_N)$ ▷ Initialization using the final condition
for $t = N-1, N-2, \cdots, 0$ **do**
  $V^t \leftarrow \mathcal{K}V^{t+1}$

As we can see, the index space has $\#\mathtt{I} = (2N^Y+1)(2N^S+1)(2N^P+1)(2N^Q+1)(2N^H+1)$ elements, while the time space have N elements. For each item in the time and state space grid, we must search through the control space $\mathtt{U}$ and through the impulse space Ξ. This makes the number of operations at least $2^6\mathcal{O}(NN^YN^SN^PN^QN^H(N^U + N^\Xi))$. If any of the model parameters needs to be updated, a new recalculation is necessary. As we can see, the time complexity of this algorithm escalates quickly the larger T is. For example, if our time range is 5 minutes, and $\Delta s = \Delta p = 0.01$, $\Delta q = \Delta h = 1$, $\Delta t = 0.0001$ (1 basis point), we estimate that $\#\mathtt{S} = \#\mathtt{P} = 50$, and if the dealer can hold up to 100 shares, then $\#\mathtt{Q} = \#\mathtt{H} = 201$. If $N = 300$ (so Δt is 1 second), we end up with more than 30 billion possible states. As T increase, the number of possibilities for the final price S also increases, evidencing the curse of dimensionality of the problem.

4.2 Reinforcement Learning

The concepts and methods of *Dynamic Programming* (DP) and *Reinforcement Learning* (RL) have striking similarities, but according to Buşoniu et al. (2010), the main difference lies in the reliance on models: DP precludes a model for the state dynamics, hence it is considered a model-based approach, while RL does not require any kind of model, thus it is considered a model-free / data-based approach. Although this is a very important distinction, in the rest of this work, we will consider Approximate Dynamic Programming (ADP) as model-based RL, employing the term *Reinforcement Learning* in the broader sense when referring to both model-free and model-based approaches.

The objective of using *Approximate Dynamic Programming* to our problem is to obtain a near-optimal approximative solution to a dynamic program instead of analytically or numerically solving the HJBQVI system. This approach was specifically conceived to handle the curse of dimensionality, and it has been successful in obtaining solutions that are very close to optimality. In general, when discussing model-based and model-free Reinforcement Learning methods, we will be exclusively interested in those that are able to handle large state spaces.

Powell (2009) and Powell (2014) are good introductory articles on ADP. In the later, the author compares different approximative methods by applying them to a problem with known analytical solution and benchmarking those suboptimal approximations against the known optimal solution.

Reinforcement learning problems distinguish themselves from the traditional *Supervised learning* (SL) problems in that it is not always clear what label should be assign to the training data. In supervised learning, problems are be solved by training a prediction model with labeled data, so that the model can extrapolate and predict a label for a new data point. The dynamical structure of problems like weather prediction and algorithmic trading prevent us from easily identifying labels: we would have to wait until the label is known is order to train our model. In reinforcement learning, instead of labeling data, we provide a reward mechanism that facilitates the learning process, and the model is trained not with examples, but by incentives. Hence, RL methods are particularly suited to dynamic optimization problems like high-frequency trading, with the added bonus that they improve with experience, contrary to supervised learning methods.

The proposed use of Markov chain in Veraart (2010) is an application of model-based RL to the market maker problem. A more recent study (Spooner et al. 2018) focuses on analyzing a model-free RL method called *Temporal Differences* (TD) to market making, presenting interesting results. The problem of model-based approaches for trading is that they are only as good as the model, failing in the real world when the model assumptions are not verified, while the success of model-free approaches relies on the fact that they are data-based: *data is*

a better proxy to reality. However, as we will discuss later, we consider model-based techniques relevant as *training bootstrap*.

4.2.1 Preliminaries

Before we proceed, we must establish a few definitions and notations. In general we use π to denote a deterministic policy, which is a function $\pi : \mathcal{X} \to \mathfrak{A}$ that maps a state x_t from the state space $\mathcal{X}$ into an action a_t from the action space $\mathfrak{A}$.

In Reinforcement Learning problems, instead of dealing with labeled datasets of the form $\{(x_i, z_i); i = 0, 1, \cdots, n\}$ common in Supervised Learning, we usually deal time-series datasets of the form $\{(x_1, a_1, r_1), (x_2, a_2, r_2), \cdots, (x_T, a_T, r_T)\}$, which are called *episodes*, *trajectories* or just *state-action-reward sequences*. For the market making problem we defined, these episodes can be simulated by applying a given policy to the state dynamics of the controlled process we define in our problem formulation (section 3.2). This form of representing the data is natural in algorithmic trading, market making and other financial applications.

Based on (4), in a discrete-time setting, the *Bellman operator*[19] $\mathcal{T}^\pi$ for the policy π on a function $V : \mathcal{X} \to \mathbb{R}$ is defined as

$$\mathcal{T}^\pi V(x_t) = \mathbb{E}\left[R(x_t, \pi(x_t), x_{t+1}) + \gamma V(x_{t+1}) | \mathcal{F}_t\right]$$

where $R(x_t, \pi(x_t), , x_{t+1})$ is the reward (or cost) of applying the policy π while in state x_t, γ is a discounting factor and $\mathcal{F}_t$ is the filtration generated by the discrete-time controlled state process $(x_t)_{t=1,2,\dots}$. Extending this concept, the *Optimality operator*[20] $\mathcal{T}^*$ on the same function V is defined as

$$\mathcal{T}^* V(x_t) = \max_\pi \mathbb{E}\left[R(x_t, \pi(x_t), x_{t+1}) + \gamma V(x_{t+1}) | \mathcal{F}_t\right]$$

Both $\mathcal{T}^\pi$ and $\mathcal{T}^*$ are called *Backup operators* (Williams & Baird 1993). We can then redefine the *value function* V^π of the policy π as the fixed-point of the Bellman operator $\mathcal{T}^\pi$:

$$\mathcal{T}^\pi V^\pi = V^\pi$$

and the *optimal value function* V^* as the fixed-point of the optimality operator $\mathcal{T}^*$

$$\mathcal{T}^* V^* = V^*$$

By definition, the *Bellman residual* (or *Bellman error*) $\mathcal{T}^\pi V - V$ for the value function V^π and the *Optimal Bellman residual* $\mathcal{T}^* V^* - V^*$ for the optimal value function V^* are both 0.

[19] Some authors also call it *Bellman evaluation operator*, since it is used only for policy evaluation

[20] Some authors also call it *Dynamic programming operator* (De Farias & Van Roy 2000)

If we remove the expectation operator from these Bellman operators, we have what are called *Sampled Bellman operators* $\hat{\mathcal{T}}^\pi$ and $\hat{\mathcal{T}}^*$:

$$\hat{\mathcal{T}}^\pi V(x_t) = R(x_t, \pi(x_t), x_{t+1}) + \gamma V(x_{t+1})$$
$$\hat{\mathcal{T}}^* V(x_t) = \max_\pi R(x_t, \pi(x_t), x_{t+1}) + \gamma V(x_{t+1})$$

where x_{t+1} is sampled either based on the state dynamics, or from some episode.

The *Greedy Policy Operator* $\mathcal{E}$ of a value function V is defined as

$$\mathcal{E}V(x_t) = \operatorname*{argmax}_\pi \mathbb{E}\left[R(x_t, \pi(x_t), x_{t+1}) + \gamma V(x_{t+1}) | \mathcal{F}_t\right]$$

and a policy π is said to be *greedy* for a given value function V if $\pi(x_t) = \mathcal{E}V(x_t)$ for all states $x_t \in \mathcal{X}$. The *optimal policy* π^* is then the greedy policy for the optimal value function V^*, i.e., $\pi^* = \mathcal{E}V^*$.

Value functions can also be expressed as *Action-Value functions*, which many authors also refer to as *Q-functions*. The action-value function $Q^\pi : \mathcal{X} \times \mathfrak{A} \to \mathbb{R}$ maps a state x_t and action a_t to the value resultant from immediately taking action a_t and following the policy π onwards

$$\begin{aligned} Q^\pi(x_t, a_t) &= \mathbb{E}_t\left[R(x_t, a_t, x_{t+1}) + \gamma V^\pi(x_{t+1}) | \mathcal{F}_t\right] \\ &= \mathbb{E}\left[R(x_t, a_t, x_{t+1}) + \gamma Q^\pi(x_{t+1}, \pi(x_{t+1})) | \mathcal{F}_t\right] \end{aligned}$$

and the optimal action-value function $Q^* : \mathcal{X} \times \mathfrak{A} \to \mathbb{R}$ is defined by $Q^* = Q^{\pi^*}$. All the operators previously defined for value functions can also be applied for action-value functions

$$\begin{aligned} \mathcal{T}^\pi Q(x_t, a_t) &= \mathbb{E}\left[R(x_t, a_t, x_{t+1}) + \gamma Q(x_{t+1}, \pi(x_{t+1})) | \mathcal{F}_t\right] \\ \mathcal{T}^* Q(x_t, a_t) &= \mathbb{E}\left[R(x_t, a_t, x_{t+1}) + \gamma \max_a Q(x_{t+1}, a) | \mathcal{F}_t\right] \\ \hat{\mathcal{T}}^\pi Q(x_t, a_t) &= R(x_t, a_t, x_{t+1}) + \gamma\, \mathbb{E}\left[Q(x_{t+1}, \pi(x_{t+1}))\right] \\ \hat{\mathcal{T}}^* Q(x_t, a_t) &= R(x_t, a_t, x_{t+1}) + \gamma \max_a Q(x_{t+1}, a) \end{aligned}$$

and the greedy policy operator $\mathcal{E}$ of action-value functions Q is thus defined as

$$\mathcal{E}Q(x_t) = \operatorname*{argmax}_\pi Q(x_t, \pi(a))$$

4.2.2 Value Iteration

The method called *Approximate Value Iteration* tries to approximate the value function V (can be either V^π or V^*) by recursively defining a sequence of functions $(V_i)_{i=0,1,\dots}$ with the Backup operator $\mathcal{T}$ (either T^π or T^*)

$$V_i = \mathcal{T} V_{i-1}$$

where $V_0 = \bar{V}$ for some given function $\bar{V}$, until convergence, i.e., the sequence of Bellman residuals $\mathcal{T}V_i - V_i$ reaches an established error threshold $\epsilon > 0$ as $i \to \infty$. This method is

said to be a *bootstrapping* method because the choice of the function $\bar{V}$ is arbitrary. Since the Bellman operator is a contraction, the performance of the greedy policy $\mathcal{E}V_{i+1}$ is guaranteed to be better than the greedy policy $\mathcal{E}V_i$, and the sequence of greedy policies converges, i.e., $\lim_{i\to\infty} \mathcal{E}V_i = \mathcal{E}V$.

Since it needs to calculate an expectation for every step in the recursion, this method tends to be computationally slow, only working well if the initial $\bar{V}$ is a good approximation for all $x_t \in \mathcal{X}$. If we are able to find good approximations for subsets of $\mathcal{X}$, we can also compose them into a decision tree and obtain such V_0.

The sequence of greedy policies generated by this method are also known as *look-ahead* policies, and $\mathcal{E}V_i$ is said to be an $(i+1)$-step look-ahead policy. In a variant of the value iteration, instead of electing an arbitrary $\bar{V}$ and performing a recursive computation until convergence, the number of iterations i is fixed, and some intelligent choice of approximation function $\bar{V}$ is used.

4.2.3 Rollout

The *rollout method* (Bertsekas et al. 1997, Bertsekas 2013) is a subclass of the value iteration (some authors also relate it to policy improvement) method where function $\bar{V}$ is chosen to be the value of some sub-optimal policy that can be calculated analytically or by simulation. Such sub-optimal policy is called the *heuristic policy*. For the reasons already discussed, the greedy policy $\mathcal{E}V_i = \mathcal{E}(\mathcal{T}^*)^i\bar{V}$ is guaranteed to outperform the heuristic policy used to calculate $\bar{V}$.

For the infinite-horizon problem, a possible heuristic policy is to quote a constant spread for δ^a and δ^b and never hedge ($\xi = 0$). The existence of a model for the ETF limit order book then allow us to try to find an analytical formula for the value of such heuristic policy, or just use Monte Carlo simulation. Another example of heuristic policy that can be employed is the immediate portfolio liquidation policy. In that case, the value of such trivial policy is just the portfolio value minus the liquidation costs, which is a well researched topic (Almgren & Chriss 2001, Bouchaud 2009).

4.2.4 Fitted Value Iteration

Instead of adopting a heuristic policy and performing a rollout, we can instead approximate the optimal value function V^* by making a sensible choice for the bootstrap function $\bar{V}$ between members of some parametric family of functions $\mathfrak{V}$.

Under this parametric approach, a functional architecture $\bar{V}[\theta;\phi]$ is proposed, where $\bar{V}$ is a functional, $\theta \in \mathbb{R}^n$ is a n-dimensional parameter and $\phi : \mathcal{X} \to \mathbb{R}^m$ is a vector valued function whose components $\{\phi_i; i = 1, \cdots, m\}$, called *features*, define a set of basis functions. These

three elements span the space of functions $\mathfrak{V}$, under which the desired $\hat{V}$ is optimal for the training sample. The linear architecture

$$\bar{V}[\theta;\phi](X_t) = \sum_{i=1,\cdots,n} \theta_i \phi_i(X_t) \tag{17}$$

is basically a linear combination of the (possibly non-linear) basis functions ϕ_i, being very popular among practitioners of RL. Another area of study that has gained popularity is *Neuro-dynamic programming* (NDP), which studies the use of neural networks as approximation architectures in ADP (Bertsekas & Tsitsiklis 1995). If such NN are multiple layered, then we say such architecture is a *Deep Neural Network* (DNN) and the study of RL methods restricted to NN and DNN architectures is the subject of *Deep Reinforcement Learning*.

The numerous ways to calibrate $\bar{V}$ to data can be grouped in two classes of RL methods (Geist & Pietquin 2010). *Projection methods*[21] focus on trying to project the sequence V_i of value iteration functions onto the parametric function space $\mathfrak{V}$, generating a sequence of project value functions $\bar{V}_i \in \mathfrak{V}$ defined by $\bar{V}_i = \Pi V_i$ as to minimize $\|\Pi V_i - V_i\|^{\mathfrak{n}}$ for some norm $\mathfrak{n}$. *Gradient methods* applies gradient descent to generate a sequence of value functions $\bar{V}_i$ without ever leaving the function space $\mathfrak{V}$, avoiding the need for projections, while also minimizing the objective $\|\mathcal{T}V_i - V_i\|^{\mathfrak{n}}$ under the norm $\mathfrak{n}$. Since the value functions $\bar{V}_i$ are parameterized by θ_i, the sequence $(\bar{V}_i)_{i=0,1,2\cdots}$ is defined by the parameter sequence $(\theta_i)_{i=0,1,\cdots}$ under the update rule $\theta_{i+1} = \theta_i + \alpha \nabla_{\theta_i} \bar{V}_i$ for some learning rate α and initial θ_0.

For both approaches, the objective is to minimize the Bellman residual $\mathcal{T}^*\bar{V} - \bar{V}$ directly, or the sampled Bellman residual $\hat{\mathcal{T}}^*\bar{V} - \bar{V}$, which is also called *Temporal Difference error*. Methods that minimize the Bellman residual are called *Residual* methods, while methods that minimize the sampled Bellman residual are called *Temporal Differences* (TD) methods. By definition, TD methods require the ability to sample data, whereas Residual methods can but do not necessarily require sampling.

Projected Residuals Let us apply the projection methods to minimize the Bellman residual. Assuming the linear architecture (17) with some chosen feature function ϕ and some initial value for θ_0, a sequence of approximate value functions $(\bar{V}_i)_{i=0,1,\cdots}$ is defined by

$$\bar{V}_i = \bar{V}[\theta_i;\phi]$$

Under the L^2 norm, we define $\bar{V}_{i+1}$ as the function in $\mathfrak{V}$ that best approximates $\mathcal{T}^*\bar{V}_i$ by minimizing $(\mathcal{T}^*\bar{V}_i - \bar{V}_{i+1})^2$, i.e., by performing an ordinary least squares linear regression

$$\theta_{i+1} = \operatorname*{argmin}_{\theta} \sum_{x \in \tilde{\mathcal{X}}} \|\mathcal{T}^*\bar{V}_i(x) - \bar{V}[\theta;\phi](x)\|^2 \tag{18}$$

[21] Geist & Pietquin (2010) uses the term *Projected Fixed-Point methods*

over a sample $\tilde{\mathcal{X}}$ of the state space $\mathcal{X}$. Once a reasonably good parameter vector θ_k is reached, the greedy policy $\mathcal{E}\bar{V}[\theta_k;\phi]$ is used. Regularization techniques can be applied to the regression problem (18) to avoid over-fitting. A particular class of L^1 regularization, LASSO, is also able to perform 'feature selection', thus reducing the number of basis functions and helping us achieve the right balance of model complexity. In particular, Geist & Scherrer (2012) studies L^1 penalized Projected Bellman residual methods. We refer to Farahmand (2011) as an excellent work regarding the application of regularization methods to reinforcement learning.

Algorithm 2 Projected Residuals

1: $\bar{V}_i \equiv \bar{V}[\theta_i;\phi]$
2: $\mathcal{T}^*\bar{V}_i(x_t) = \max_u \mathbb{E}_t\left[R(x_t,u) + \gamma\bar{V}_i(x_t)|x_t\right]$ ▷
3: Initialize θ_0
4: **for all** $i = 0, 1, \cdots, n$ or until convergence **do**
5: $\quad\theta_{i+1} \leftarrow \mathrm{argmin}_\theta \sum_{x\in\tilde{\mathcal{X}}} \|\mathcal{T}^*\bar{V}_i(x) - \bar{V}[\theta;\phi](x)\|^2$

Temporal Differences In *Temporal Difference* (TD) methods (Sutton 1988), we try to minimize the TD error $\hat{\mathcal{T}}\bar{V}_i - \bar{V}_i$, instead of the Bellman residual $\mathcal{T}^*\bar{V}_i - \bar{V}_i$. This minimization can be performed by gradient methods, as in the original publication of Sutton (1988), or by projection methods, as in Bradtke & Barto (1996) where least squares are used in similar fashion to (18).

Under this method, a sequence of value predictors $(\bar{V}_i)_{i=0,1,\cdots}$ are defined by iterations of predict-measure-update steps. During the $k+1$-th iteration, $\bar{V}_k$ is used to predict the value of each state of the state-action-reward sequence (x_1,u_1,r_1), (x_2,u_2,r_2), $\cdots$. After this prediction step is performed, the measure step reviews the prediction quality of the predictor $\bar{V}_k$, and based on such measurement, the update is performed, generating a new value predictor $\bar{V}_{k+1}$.

The measurement step is the most interesting aspect of the TD method. As time passes, every state transition from x_t into x_{t+1} contributes with more information, and for that episode, the complete information is known only at T. Information is represented as the set $I_t = \{r_1, \cdots, r_{t-1}\}$. As t increases from 1 to T, the value predictor $\bar{V}_k$ can be improved for all previous $x_{t'<t}$: while at time t the prediction for $V(x_t)$ is $\bar{V}_k(x_t)$, at time $t+1$ the prediction for $V(x_t)$ would have been $r_t + \gamma\bar{V}_k(x_{t+1})$. The information delta $\Delta I_{t+1} = I_{t+1} - I_t = \{r_t\}$ is used to update the predictor $\bar{V}_k$ by minimizing the Temporal Difference error for the states $x_{t'<t}$.

The objective is to minimize the total TD error for the episode

$$\sum_t \|\hat{\mathcal{T}}\bar{V}_k - \bar{V}_k\|^2 = \sum_{t=1,\cdots,T} \|r_t + \gamma\bar{V}_k(x_{t+1}) - \bar{V}_k(x_t)\|^2$$

The parameter θ_{k+1} for the next predictor $\bar{V}_{k+1}$ is initialized with θ_k and then updated by

Table 1: Value Predictions and TD errors

State	**Prediction given** $I_{t'}$	**Prediction given** $I_{t'+1}$	**TD error**
x_1	$\bar{V}_k(x_1)$	$r_1 + \gamma\bar{V}_k(x_2)$	$r_1 + \gamma\bar{V}_k(x_2) - \bar{V}_k(x_1)$
x_2	$\bar{V}_k(x_2)$	$r_2 + \gamma\bar{V}_k(x_3)$	$r_2 + \gamma\bar{V}_k(x_3) - \bar{V}_k(x_2)$
$\cdots$	$\cdots$	$\cdots$	$\cdots$
x_t	$\bar{V}_k(x_t)$	$r_t + \gamma\bar{V}_k(x_{t+1})$	$r_t + \gamma\bar{V}_k(x_{t+1}) - \bar{V}_k(x_t)$

gradient descent

$$\theta_{k+1} \leftarrow \theta_{k+1} + \alpha(r_t + \gamma\bar{V}_k(x_{t+1}) - \bar{V}_k(x_t)) \sum_{t'=1,\cdots,t} \nabla_\theta \bar{V}_k(x_{t'}) \tag{19}$$

where α is called the *learning rate*. The update rule (19) defines the $TD(1)$ algorithm. The $TD(\lambda)$ family, for $0 \leq \lambda \leq 1$, is the same algorithm but with a different update rule:

$$\theta_{k+1} \leftarrow \theta_{k+1} + \alpha(r_t + \gamma\bar{V}_k(x_{t+1}) - \bar{V}_k(x_t)) \sum_{t'=1,\cdots,t} \lambda^{t-t'} \nabla_\theta \bar{V}_k(x_{t'})$$

where the λ parameter defines how further in the past we update our previous predictions.

TD methods are not only very attractive but also very successful: they can be implemented iteratively, are computationally very efficient, with good convergence and can perform both *offline* and online learning in various degrees. During *Offline learning*, a new parameter θ_{k+1} is defined only after processing a whole episode. In online learning, new parameters θ_{k+1} are generated much more frequently, possibly after each time step, without waiting for the end of the episode. This is a very important aspect for algorithmic trading and market making: the possibility to perform offline training with historical market data[22] before the opening of the trading session, and continue with online learning while executing the trades during the day.

[22]TD methods can also be applied in model-based RL by performing offline training on simulated data

Algorithm 3 Model-based offline TD(λ)

$\bar{V}_i \equiv \bar{V}[\theta_i; \phi]$

$\hat{\mathcal{T}}\bar{V}_i(x_t, e_i) \equiv \max_u \left[R(x_t, u) + \gamma \bar{V}_i(x_{t+1})|e_i\right]$ ▷ x_{t+1} is draw from the episode e_i

Initialize θ_0

for all $i = 0, 1, \cdots, n$ or until convergence **do**

  Define policy $\pi_i \equiv \mathcal{E}\bar{V}_i$

  Generate episode e_i by simulating the state dynamics under the policy π_i

  Initialize $\theta_{i+1} \leftarrow \theta_i$

  Initialize $\Delta V \leftarrow 0$

  for all $t = 0, 1, \cdots, T$ **do**

    $\Delta V \leftarrow \lambda \Delta V + \nabla_{\theta_i} \bar{V}_i(x_t)$

    $\theta_{i+1} \leftarrow \theta_{i+1} + \alpha(\hat{\mathcal{T}}\bar{V}_i(x_t, e_i) - \bar{V}_i(x_t))\Delta V$

Final policy is $\pi^* \equiv \mathcal{E}\bar{V}_{n^*}$ ▷ θ_{n^*} is the converged parameter

4.2.5 Approximate Policy Iteration

So far the methods we have discussed approximate only the optimal value function V^*, and are called *'critic-only'* methods. Methods that approximate the optimal policy function π^* directly are called *'actor'* methods, where 'actor' term is used to refer to the *Policy Function Approximation* (PFA). Actor methods are computationally more efficient than calculating the various expectations required by the VFA policy $\mathcal{E}\bar{V}_{n^*}$, making them ideal for latency-sensitive high-frequency market making.

There is an important distinction in the nature of the policy. In critic-only methods, the policies $\mathcal{E}\bar{V}_i$ are always deterministic, while with actor methods, we can search for policies in the wider class of stochastic policies, whose outputs are probability distributions on the action space.

A stochastic policy $\pi : \mathcal{X} \times \mathfrak{A} \to [0, 1]$ maps a state x and an action u to a conditional probability, i.e., $\pi(x, u) = \mathbb{P}(u_t = u|x_t = x)$. In *Approximate Policy Iteration* (Williams 1992, Sutton et al. 1999, Perkins & Precup 2003), a sequence of stochastic policies $(\pi_i)_{i=0,1,\cdots}$ are drawn from a parametric family of functions $\mathfrak{P}$ by iterations of *policy evaluation* and *policy improvement* steps, where $\pi_0 = \bar{\pi}[\vartheta_0] \in \mathfrak{P}$ is chosen arbitrarily. In practice, this sequence of parametric functions is represented by the sequence of parameters $(\vartheta_i)_{i=0,1,\cdots}$, and it is expected that this sequence converges to some ϑ_{n^*} for which the policy $\pi_{n^*} = \bar{\pi}[\vartheta_{n^*}]$ is considered optimal.

The most interesting aspect of policy iteration is the policy evaluation procedure. While in VFA methods we were interested in the optimal value $V^*(x_t)$ of a state x_t, here the interest is in the value of a policy π_i,

$$V(\pi_i) = \mathbb{E}\left[V^{\pi_i}(x)\right] \tag{20}$$

the average total reward of the policy regardless of circumstances (initial state).

In the policy improvement step, the objective is to define a new improved policy π_{i+1} based on an existing policy π_i such that $V(\pi_{i+1}) \geq V(\pi_i)$. The *policy gradient* methods perform this by gradient ascent:

$$\vartheta_{i+1} \leftarrow \vartheta_i + \alpha \nabla_\vartheta V(\pi_i)$$

where α is the learning rate. The challenge remains on how to calculate the gradient $\nabla_\vartheta V(\pi_i)$. Williams (1992) proposes an actor-only algorithm named REINFORCE, which is also called *Monte Carlo Policy Gradient* (Sutton & Barto 2018, chapter 13, pages 326-329). This algorithm essentially simulates an episode e_i for each policy π_i

$$\begin{aligned} e_i = \{&(x_{i,1}, u_{i,1}, r_{i,1}), \\ &(x_{i,2}, u_{i,2}, r_{i,2}), \\ &\cdots, \\ &(x_{i,T}, u_{i,T}, r_{i,T})\} \end{aligned}$$

to perform an approximative policy evaluation step called *episodic policy evaluation*:

$$V(\pi_i) \approx V(e_i) = \sum_t \gamma^{t-1} r_{i,t}$$

The policy update step is then realized using the Policy Gradient Theorem (Sutton et al. 1999), restricted to the episode in question. We skip the details about the derivation of the update rule and present it below in the algorithm 4.

Algorithm 4 Model-based offline REINFORCE

1: $\pi_i(x) \equiv \bar{\pi}[\vartheta_i](x)$
2: $V(e_i, t) \equiv \sum_{k \geq t} \gamma^{k-1} r_{j,k}^{\pi_i}$
3: Initialize ϑ_0
4: **for all** $i = 0, 1, \cdots, n$ or until convergence **do**
5: Generate episode e_i under the policy π_i
6: Initialize $\vartheta_{i+1} \leftarrow \vartheta_i$
7: **for all** $t = 1, \cdots, T$ **do**
8: $\vartheta_{i+1} \leftarrow \vartheta_{i+1} + \alpha V(e_i, t) \nabla_\vartheta \log \pi_i(x_{i,t}, u_{i,t}))$
9: Final policy: $\pi^* \equiv \pi_{n^*}$ ▷ ϑ_{n^*} is the converged parameter

4.2.6 Actor-critic methods

The slow convergence of Monte Carlo Policy gradient methods and the high variance of their predictions point us toward *actor-critic methods*, which, in essence, allow us to approximate a policy function with the help of an approximation to the value function. The 'critic'

helps the 'actor' in finding the policy gradients that minimize the variance of the prediction error (Grondman et al. 2012), while achieving faster convergence. The convergence of actor-critic policy gradient methods to a locally optimal policy is proved by the Policy Iteration with Function Approximation Theorem in Sutton et al. (1999). The exact policy gradient is unknown, but the approximated value function can give us a rough estimate of the policy gradient, hence actor-critic methods are also known *Approximate Policy Gradient* methods. The policy update rule is also provided by Sutton et al. (1999) in the Policy Gradient with Function Approximation Theorem. It should be noted that the critic reduces the variance at the cost of introducing bias because the approximated policy gradient tends to be biased.

Bertsekas & Ioffe (1996) approximates the value function by the TD method while performing approximate policy iteration, a form of *TD-based Policy Iteration*, unifying two successful approaches.

Deterministic Policy Gradient We can also learn deterministic policies by applying the Deterministic Policy Gradient Theorem from Silver et al. (2014). This is an important aspect for high-frequency market making, as we would like to avoid changing quotes constantly. Deterministic policies map states into actions directly, hence are computationally more efficient than stochastic policies. The *Deterministic Policy Gradients* (DPG) method (Silver et al. 2014) is an actor-critic method.

In order to evaluate a deterministic policy, we must observe that for any episode e generated by a policy π, the distribution of the initial state x_1 is independent of the policy π, but the distribution of the resulting states depend on both π and on x_1. Thus, the value of an episode that follows a deterministic policy π can be expressed as

$$\sum_{t=1}^{T} \gamma^{t-1} R(x_t, \pi(x_t), x_{t+1})$$

which is a random variable dependent on the initial state x_1, the policy π and the dynamics of the controlled state process $(x_t)_{t=1,2,\dots}$. Hence, the expected value of an episode that follows a policy π given the initial state x_1 is

$$V(\pi, x_1) = \sum_{t=1}^{T} \gamma^{t-1} \, \mathbb{E}\left[R(x_t, \pi(x_t), x_{t+1})\right] \tag{21}$$

The value of a policy π can then be defined as the expected value of all episodes that follow π, i.e. regardless of circumstances, as already noted in (20).

$$V(\pi) = \mathbb{E}\left[V(\pi, x)\right] \tag{22}$$

If we denote $\mathbb{P}_t^{\pi}(x \to (x', x''))$ the probability density of observing the state transition $x_t = x' \to x_{t+1} = x''$ in an episode following the policy π given an initial $x_1 = x$, then we can express

(21) as

$$V(\pi, x) = \sum_{t=1}^{T} \gamma^{t-1} \int_{(x',x'')\in\mathcal{X}\times\mathcal{X}} \mathbb{P}_t^{\pi}(x \to (x', x''))R(x', \pi(x'), x'')$$
$$= \int_{(x',x'')\in\mathcal{X}\times\mathcal{X}} \sum_{t=1}^{T} \gamma^{t-1}\mathbb{P}_t^{\pi}(x \to (x', x''))R(x', \pi(x'), x'')$$

which by replacing $\sum_{t=1}^{T} \gamma^{t-1}\mathbb{P}_t^{\pi}(x \to (x', x''))$ for $d^{\pi}(x, x', x'')$ gives

$$V(\pi, x) = \int_{(x',x'')\in\mathcal{X}\times\mathcal{X}} d^{\pi}(x, x', x'')R(x', \pi(x'), x'')$$

From this last results and (22), the value of a policy $V(\pi)$ can be expressed by

$$V(\pi) = \int_{(x',x'')\in\mathcal{X}\times\mathcal{X}} d^{\pi}(x', x'')R(x', \pi(x'), x'')$$
$$= \mathbb{E}_{(x',x'')\sim d^{\pi}}\left[R(x', \pi(x'), x'')\right]$$

where $d^{\pi}(x', x'') = \mathbb{P}(x_1 = x)d^{\pi}(x, x', x'')$ is called the (improper) *discounted state distribution* under π (Silver et al. 2014).

Given this policy evaluation procedure, the Deterministic Policy Gradient Theorem (Silver et al. 2014) states the following

Theorem 1. Deterministic Policy Gradient Theorem

Assume $\nabla_{\vartheta}\pi(x)$ and $\nabla_a Q^{\pi}(x, a)$ exists for all states $x \in \mathcal{X}$ and actions $a \in \mathfrak{A}$. Then the gradient of the objective $V(\pi)$ is

$$\nabla_{\vartheta} V(\pi) = \mathbb{E}_{x\sim d^{\pi}}\left[\nabla_{\vartheta}\pi(x)\nabla_a Q^{\pi}(x, a)|_{a=\pi(x)}\right]$$

We can then perform policy iteration by defining a sequence of policy parameters $(\vartheta_i)_{i=0,1,\dots}$ while performing gradient ascent on $V(\pi_i)$ using the above theorem, with an action-value function approximation for Q^{π_i} by the TD(λ) method. We present below two versions of this algorithm: in the online version, there is little exploratory search in the action space: the episodes are generated by the *learning policy* (the one we are learning), while in the offline version, the episodes are generated by a *behavior policy* that guarantees enough exploration of the action space, which can be just a stochastic policy which randomly selects an action (maybe by adding noise to the learning policy). The choice between these two versions is a specific case of a dilemma known as *exploration vs exploitation tradeoff*. Offline learning tends to explore and gather more data before improving the decision making, while online learning tends to make the best decision given the current data, at the risk of getting stuck at locally optimal policy.

Algorithm 5 Off-Policy Deterministic Actor-Critic

$\bar{Q}_t \equiv \bar{Q}[\theta_t; \phi]$
$\hat{\mathcal{T}}_{\text{off}}\bar{Q}_t(x_t, e_i) \equiv R(x_t, a_t, x_{t+1}) + \gamma\bar{Q}_t(x_{t+1}, \pi_t(x_{t+1}))$ ▷ For off-policy learning
Initialize θ_0 and ϑ_0
for all $i = 0, 1, \cdots, n$ or until convergence **do**
    Start new episode e_i under ψ ▷ ψ is called a *behavior policy*
    Initialize $\Delta Q \leftarrow 0$
    for all $t = 0, 1, \cdots, T$ **do**
        Grow episode e_i under ψ
        $\Delta Q \leftarrow \lambda\Delta Q + \nabla_{\theta_t}\bar{Q}_t(x_t, a_t)$
        $\theta_{t+1} \leftarrow \theta_t + \alpha_\theta(\hat{\mathcal{T}}_{\text{off}}\bar{Q}_t(x_t, e_i) - \bar{Q}_t(x_t, a_t))\Delta Q$ ▷ TD update for Critic
        $\vartheta_{t+1} \leftarrow \vartheta_t + \alpha_\vartheta\nabla_\vartheta\pi_t(x_t)\nabla_a\bar{Q}_t(x_t, a_t)|_{a=\pi_t(x_t)}$ ▷ Policy Gradient ascent
    $\theta_0 \leftarrow \theta_T$
    $\vartheta_0 \leftarrow \vartheta_T$

Algorithm 6 On-Policy Deterministic Actor-Critic

$\bar{Q}_t \equiv \bar{Q}[\theta_t; \phi]$
$\hat{\mathcal{T}}_{\text{on}}\bar{Q}_t(x_t, e_i) \equiv R(x_t, a_t, x_{t+1}) + \gamma\bar{Q}_t(x_{t+1}, a_{t+1})$ ▷ For on-policy learning
Initialize θ_0 and ϑ_0
for all $i = 0, 1, \cdots, n$ or until convergence **do**
    Start new episode e_i
    Initialize $\Delta Q \leftarrow 0$
    for all $t = 0, 1, \cdots, T$ **do**
        Grow episode e_i under π_t
        $\Delta Q \leftarrow \lambda\Delta Q + \nabla_{\theta_t}\bar{Q}_t(x_t, a_t)$
        $\theta_{t+1} \leftarrow \theta_t + \alpha_\theta(\hat{\mathcal{T}}_{\text{on}}\bar{Q}_t(x_t, e_i) - \bar{Q}_t(x_t, a_t))\Delta Q$ ▷ TD update for Critic
        $\vartheta_{t+1} \leftarrow \vartheta_t + \alpha_\vartheta\nabla_\vartheta\pi_t(x_t)\nabla_a\bar{Q}_t(x_t, a_t)|_{a=\pi_t(x_t)}$ ▷ Policy Gradient ascent
    $\theta_0 \leftarrow \theta_T$
    $\vartheta_0 \leftarrow \vartheta_T$

4.2.7 Actor-critic algorithm under HJBQVI conditions

We should note that the HJBQVI conditions can be considered a continuous time version of the Bellman residual, and thus we could devise a residual-based actor-critic method to approximate a near-optimal policy while approximating a solution to V^* using the HJBQVI conditions (10) instead of the Bellman equations. We can illustrate this method using the

following linear architectures

$$\begin{aligned}\bar{\delta}^a[\theta^a](x) &= \theta^a \cdot \phi^a(x)\\ \bar{\delta}^b[\theta^b](x) &= \theta^b \cdot \phi^b(x)\\ \bar{\xi}[\theta^\xi](x) &= \theta^\xi \cdot \phi^\xi(x)\\ \bar{V}[\theta^v](x) &= \theta^v \cdot \phi^v(x)\end{aligned} \tag{23}$$

which will be used as our approximations. It is worth noting that the $\bar{\xi}[\theta^\xi](x)$ may return 0, which indicates no intervention is to be made. We also define $(\theta_i^a)_{i\in\mathbb{N}}$, $(\theta_i^b)_{i\in\mathbb{N}}$, $(\theta_i^\xi)_{i\in\mathbb{N}}$ and $(\theta_i^v)_{i\in\mathbb{N}}$ the sequence of parameters for $\bar{\delta}^a$, $\bar{\delta}^b$, $\bar{\xi}$ and $\bar{V}$ respectively, and for short notation, this implies a sequence of approximations $(\bar{\delta}_i^a \equiv \bar{\delta}^a[\theta_i^a])_{i\in\mathbb{N}}$, $(\bar{\delta}_i^b \equiv \bar{\delta}^b[\theta_i^b])_{i\in\mathbb{N}}$, $(\bar{\xi}_i \equiv \bar{\xi}[\theta_i^\xi])_{i\in\mathbb{N}}$ and $(\bar{V}_i \equiv \bar{V}[\theta_i^v])_{i\in\mathbb{N}}$. The outline of our algorithm starts with initializing θ_0^a, θ_0^b, θ_0^ξ and θ_0^v, and defining the rest of the sequence recursively by iterating a two-phase procedure. In a first phase, we learn new policies by updating θ_i^a, θ_i^b and θ_i^ξ using θ_i^v. On the following phase, we learn a new value function by updating θ_{i+1}^v using the θ_i^a, θ_i^b, θ_i^ξ and θ_i^v learned on the previous phase.

The main difference now is regarding the use of the HJBQVI conditions instead of the Bellman equations. For the problem at hand, what really matters is to learn a policy, not the value function. The HJBQVI conditions help up establish estimates for the controls that can be used to form a policy estimate:

$$\begin{aligned}\hat{u}_i(x) &\equiv \begin{bmatrix}\hat{\delta}_i^a(x)\\ \hat{\delta}_i^b(x)\end{bmatrix} \equiv \operatorname*{argmax}_u \mathcal{L}\bar{V}_i(x,u)\\ \hat{\xi}_i(x) &\equiv \operatorname*{argmax}_\xi \mathcal{M}\bar{V}_i(x,\xi)\end{aligned}$$

where the operators $\mathcal{L}$ and $\mathcal{M}$ are defined as follows

$$\begin{aligned}\mathcal{L}\bar{V}_i(x,u) &= \mathcal{A}R(x,u) + \mathcal{A}\bar{V}_i(x,u) - \rho\bar{V}_i(x)\\ \mathcal{M}\bar{V}_i(x,\xi) &= \bar{V}_i(\Gamma(x,\xi)) - \bar{V}_i(x)\end{aligned}$$

Then the first phase resumes to learning new policies using those estimates

$$\begin{aligned}\theta_i^a &= \operatorname*{argmin}_{\theta^a} \sum_{x\in\mathcal{X}^*} \|\hat{\delta}_i^a(x) - \bar{\delta}^a[\theta^a](x)\|^2\\ \theta_i^b &= \operatorname*{argmin}_{\theta^b} \sum_{x\in\mathcal{X}^*} \|\hat{\delta}_i^b(x) - \bar{\delta}^b[\theta^b](x)\|^2\\ \theta_i^\xi &= \operatorname*{argmin}_{\theta^\xi} \sum_{x\in\mathcal{X}^*} \|\hat{\xi}_i(x) - \bar{\xi}[\theta^\xi](x)\|^2\end{aligned}$$

which can be solved by projection or by gradient methods by sampling $\mathcal{X}^*$ from $\mathcal{X}$, with the similar considerations as discussed in the previous sections. The second phase defines an estimate for V again using the HJBQVI conditions, which is a continuous time version of the

Bellman equation

$$\hat{V}_i(x) \equiv \bar{V}_i(x) + \max\left[\mathcal{L}\bar{V}_i(x, \bar{u}_i(x)), \mathcal{M}\bar{V}_i(x, \bar{\xi}_i(x))\right]$$

where $\bar{u}_i(x) \equiv \begin{bmatrix} \bar{\delta}_i^a(x) \\ \bar{\delta}_i^b(x) \end{bmatrix}$, and learning a new value function

$$\theta_{i+1}^v = \underset{\theta^v}{\operatorname{argmin}} \sum_{x \in \mathcal{X}^*} \|\hat{V}_i(x) - \bar{V}[\theta^v](x)\|^2$$

.

Proposition 2. *The estimates $\hat{u}$ and $\hat{\xi}$ are equivalent to a continuous-time look-ahead policy with value function approximation $\bar{V}$.*

Proof. We prove this by simplifying the expressions for $\hat{u}$ and $\hat{\xi}$ as follows

$$\begin{aligned}
\hat{\delta}_i^a(x) &= \underset{\delta^a}{\operatorname{argmax}}\, \mathcal{L}\bar{V}_i \\
&= \underset{\delta^a}{\operatorname{argmax}}\, \mathcal{A}R(x, \begin{bmatrix} \delta^a \\ \delta^b \end{bmatrix}) + \mathcal{A}\bar{V}_i(x, \begin{bmatrix} \delta^a \\ \delta^b \end{bmatrix}) - \rho\bar{V}_i(x) \\
&= \underset{\delta^a}{\operatorname{argmax}}\, \mathcal{A}R(x, \begin{bmatrix} \delta^a \\ \delta^b \end{bmatrix}) + \mathcal{A}\bar{V}_i(x, \begin{bmatrix} \delta^a \\ \delta^b \end{bmatrix}) \\
&= \underset{\delta^a}{\operatorname{argmax}}\, \lambda_a(\mathbb{E}\,\Delta_a R + \mathbb{E}\,\Delta_a\bar{V}_i) + \lambda_b(\mathbb{E}\,\Delta_b R + \mathbb{E}\,\Delta_b\bar{V}_i) \\
&= \underset{\delta^a}{\operatorname{argmax}}\, \lambda_a(\mathbb{E}\,\Delta_a R + \mathbb{E}\,\Delta_a\bar{V}_i) \\
&= \underset{\delta^a}{\operatorname{argmax}}\, \mathbb{E}\,\Delta_a R + \mathbb{E}\,\Delta_a\bar{V}_i \\
&= \underset{\delta^a}{\operatorname{argmax}}\, \mathbb{E}\left[R(x + \Delta_a(x, \delta^a, r^a)) - R(x)\right] + \mathbb{E}\left[\bar{V}_i(x + \Delta_a(x, \delta^a, r^a)) - \bar{V}_i(x)\right] \\
&= \underset{\delta^a}{\operatorname{argmax}}\, \mathbb{E}\, R(x + \Delta_a(x, \delta^a, r^a)) + \mathbb{E}\,\bar{V}_i(x + \Delta_a(x, \delta^a, r^a)) \\
\hat{\delta}_i^b(x) &= \underset{\delta^b}{\operatorname{argmax}}\, \mathbb{E}\, R(x + \Delta_b(x, \delta^b, r^b)) + \mathbb{E}\,\bar{V}_i(x + \Delta_b(x, \delta^b, r^b))
\end{aligned}$$

where we just eliminated the terms that do not depend on δ^a or δ^b and

$$\Delta_a(x, \delta^a, r^a) \equiv \gamma^{(a)} = \begin{bmatrix} s\mathbb{1}_{r^a \geq \delta^a}(1 + \delta^a) \\ sr^a \\ sr^a \\ -\mathbb{1}_{r^a \geq \delta^a} \\ 0 \end{bmatrix}$$

$$\Delta_b(x, \delta^b, r^b) \equiv \gamma^{(b)} = \begin{bmatrix} -s\mathbb{1}_{r^b \geq \delta^b}(1 - \delta^a) \\ -sr^b \\ -sr^b \\ \mathbb{1}_{r^b \geq \delta^b} \\ 0 \end{bmatrix}$$

are the impact of a jump in N^a and N^b to the state x. These expressions for $\hat{\delta}^a$ and $\hat{\delta}^b$ means they are infinitesimal-step look-ahead policies with value approximation $\bar{V}$. We can also simplify $\hat{\xi}_i(x)$ as follows

$$\begin{aligned}\hat{\xi}_i(x) &= \operatorname*{argmax}_{\xi} \mathcal{M}\bar{V}_i(x,\xi) \\ &= \operatorname*{argmax}_{\xi} \bar{V}_i(\Gamma(x,\xi)) - \bar{V}_i(x) \\ &= \operatorname*{argmax}_{\xi} \bar{V}_i(\Gamma(x,\xi))\end{aligned}$$

and since the intervention is instantaneous, no reward is immediately gained, and thus $\hat{\xi}$ is basically a policy based the value function approximation $\bar{V}$. □

The estimate $\hat{\xi}$ is deterministic and can be solved by simple search over the space Ξ or by gradient methods if the architecture $\bar{V}$ is made of differentiable functions and by the fact that Γ is differentiable in ξ.

Algorithm 7 Continuous time actor-critic

1: $\hat{\delta}_i^a(x) \equiv \operatorname{argmax}_{\delta^a} \mathbb{E}\, R(x + \Delta_a(x,\delta^a,r^a)) + \mathbb{E}\, \bar{V}_i(x + \Delta_a(x,\delta^a,r^a))$
2: $\hat{\delta}_i^b(x) \equiv \operatorname{argmax}_{\delta^b} \mathbb{E}\, R(x + \Delta_b(x,\delta^b,r^b)) + \mathbb{E}\, \bar{V}_i(x + \Delta_b(x,\delta^b,r^b))$
3: $\hat{\xi}_i(x) \equiv \operatorname{argmax}_{\xi} \bar{V}_i(\Gamma(x,\xi))$
4: $\hat{V}_i(x) \equiv \bar{V}_i(x) + \max\left[\mathcal{L}\bar{V}_i(x,\bar{u}_i(x)), \mathcal{M}\bar{V}_i(x,\bar{\xi}_i(x))\right]$
5: Initialize θ_0^v and θ_0^w
6: **for all** $i = 0, 1, \cdots, n$ or until convergence **do**
7: $\quad \theta_i^a \leftarrow \operatorname{argmin}_{\theta^a} \sum_{x\in\mathcal{X}^*} \|\hat{\delta}_i^a(x) - \bar{\delta}^a[\theta^a](x)\|^2$
8: $\quad \theta_i^b \leftarrow \operatorname{argmin}_{\theta^b} \sum_{x\in\mathcal{X}^*} \|\hat{\delta}_i^b(x) - \bar{\delta}^b[\theta^b](x)\|^2$
9: $\quad \theta_i^\xi \leftarrow \operatorname{argmin}_{\theta^\xi} \sum_{x\in\mathcal{X}^*} \|\hat{\xi}_i(x) - \bar{\xi}[\theta^\xi](x)\|^2$
10: $\quad \theta_{i+1}^v \leftarrow \operatorname{argmin}_{\theta^v} \sum_{x\in\mathcal{X}^*} \|\hat{V}_i(x) - \bar{V}[\theta^v](x)\|^2$

4.3 Convergence

While the traditional numerical schemes to solve the HJBQVI have nice convergence properties (Azimzadeh et al. 2017, Azimzadeh 2017) , that cannot be taken for granted regarding function approximation approaches. Boyan & Moore (1995) raises concerns about the lack of convergence guarantees for general value approximations, showing that such methods do not automatically inherit the convergence properties of table lookup RL methods. The authors classify convergence in the following four categories

(i) *Good convergence*: The function approximations converges to the correct solution.

(ii) *Lucky convergence*: The approximation converges to the wrong value function in absolute terms, but the relative value between the states are correct and thus the implied greedy policy is near-optimal.

(iii) *Bad convergence*: The approximation converges to the wrong solution, and the implied policy is also wrong.

(iv) *Divergence*: The approximation never converges.

and study the convergence of various approximation architectures by solving simple problems where the value function is known analytically or numerically. Boyan & Moore then propose the *Grow-Support algorithm* which is claimed to be robust and convergent. Although computationally expensive for stochastic problems (our case) due to the need to perform simulations, its usage of *support states* can improve the convergence of other algorithms. They define *support* as the set of states for which the optimal value is known, starting with a set of goal states. For the market maker infinite-horizon problem, we can, for example, establish goal states like $\begin{bmatrix} y & s & 0 & 0 & 0 \end{bmatrix}^T$ and, although we cannot say their absolute value (because we are actually trying to learn them), we can assume that the value of that state is time-independent and can be defined as y pounds plus the value of the null state $\begin{bmatrix} 0 & s & 0 & 0 & 0 \end{bmatrix}^T$, thus making sure that the relative value of our support are correct according to the wealth utility function. In *Grow-Support*, the support grows by adding more states using a *rollout* procedure.

Perhaps the most relevant convergence result on VFA is for the Temporal Differences methods, as demonstrated in Sutton (1988) and Gordon (1995). Gordon (1996) proves convergence of VFA for architectures that are non-expansion in the max-norm. Another relevant work is Lizotte (2011), which introduces a regularization method called *Expansion-Constrained Ordinary Least Squares*, guaranteeing converge for linear approximation architectures. Regarding convergence of Policy Gradient methods, the major results are Sutton et al. (1999) and Silver et al. (2014), which guarantee convergence to locally optimal policies.

Zang et al. (2010) introduces a new method called *Expanding Value Function Approximation (EVFA)* with probabilistic convergence guarantees and also proposes a human interaction scheme called *training regimen*, which allows a human to guide the learning process of the RL algorithm. This concept is highly important for training and supervision of high-frequency trading AI algorithms, which we argue must not be allowed to trade without proper *human supervision and coaching*.

In this aspect, we propose the use the model discussed in section 2 for the ETF limit order book as a type of *training regimen*. Although the jump-diffusion model we proposed is far from being the true representation of reality, it is a starting point that helps an ETF market maker by speeding up the learning of their RL algorithms, which is important for trading desks,

where not even humans are given much room for mistakes in their learning processes. Once the RL algorithm offline training is done with the assistance of the theoretical model (which we call *training bootstrap*), it is then allowed to trade in the markets with strict limits so it can perform online learning, with convergence to the near-optimal robust policies is guided on a day-by-day basis by human traders.

4.4 Robustness

The proposed algorithms are not useful if the approximated functions do not meet a minimum threshold of quality demanded from robust market making systems. Any pricing model, regardless of how of its implementation, must exhibit the following necessary conditions:

(i) *Policy validity*: Policy functions must respect financially responsible bound constraints on their outputs. It makes no sense to trade infinite number of shares or even traded above established limits. If hedging is necessary, the impulse $\xi(x)$ must decrease risk, and not increase or revert it. The quoting spreads $\delta^a(x)$ and $\delta^b(x)$ must not be negative, since continuously cross the spread leads to bankruptcy.

(ii) *Policy feasibility*: Policy functions with discretized outputs improves the feasibility of the policy by the trading engines responsible for carrying out the execution of the policy. On all trading exchanges, order prices are multiples of a tick size. A policy $\delta^a(x)$ that suggests quoting 0.015 away from the reference price when the tick size is 0.01 forces the execution layer to decide between quoting 0.01 or 0.02, when that is not supposed to be its responsibility. Actually, the difference between 0.01 and 0.02 is huge when the ETF price is small: one cent for a \$1 asset is actually 1%! Hence we believe classification policies are more robust. However, we believe value functions should remain real valued, since we must be able to compare the relative value between a huge number of states, and this is done by picking an utility function.

(iii) *Policy stability*: We require that tiny perturbations in the state do not cause the value or policy function outputs to change radically. Many states are similar, and the presence of noise in the state should not render fundamentally distinct policies. Policy stability is fundamental: any instability or oscillatory behavior in the signals $\delta^a(x)$, $\delta^b(x)$ and $\xi(x)$ will not only generate excessive number of messages, but also pose serious doubt about the correctness of the prices, increasing the risk of erroneous and unnecessary trading.

(iv) *Monotonicity in wealth utility*: The approximated value function must be monotone in the utility of the wealth. This means, for example, that the state $[y = 100, s = 100, p = 0, q = 0, h = 0]$ is logically preferable to $[y = 0, s = 100, p = 0, q = 0, h = 0]$, as the former is

precisely the later with an additional \$100 in cash. We refer to wealth utility instead of just wealth because the preference between states like $[y = 100, s = 100, p = 0, q = 0, h = 0]$ and $[y = 0, s = 100, p = 0, q = 1, h = 0]$ are mostly determined by the chosen utility function. If no alpha signal is available, a risk-averse investor would prefer \$100 in cash to an ETF position marked to market at \$100. However, if an alpha signal indicates a strong probability of appreciation to ETF prices, even a risk-averse investor would prefer to be long the ETF.

(v) *Meaningful utility*: The utility function greatly determines how the policy will behave, and hence it needs to make sense to the dealer responsible for the algorithm. A robust choice of utility will lead to superior and desirable policies, and this means not choosing an utility just because of nice mathematical tractability, but because it actually reflects the inteded *risk-reward preferences*. We also understand that the *risk-reward preference* may not only be time dependent but also a function of variables like covariance matrices[23], funding and carry rates and alphas, hence the robust calibration of the parameters is an equally important matter.

The suggested use of classification policies trivially solves the first two conditions above. Ensuring the *monotonicity in the wealth utility* also helps satisfying the *policy stability*, and we believe the former to be a necessary although not a sufficient condition for the later. We postulate *monotonicity in wealth utility* as a very important guideline to achieve robust policies because it is essentially a necessary and sufficient condition to obtain near-optimal *relative value* as opposed to the *absolute value* approximation already mentioned so far. For a good policy, what is important is knowing the relative value between two distinct states, not their absolute value, because then the policy is able to suggest actions that maximize the probability of ending up in superior states, in relative terms. By being monotonic in the wealth utility, our value function approximation is able to learn the relative value between states from the wealth utility function, thus respecting the chosen *risk-reward preferences*.

Nevertheless, a range of other approaches are also available to improve *policy stability*, and one does not necessarily exclude the others. In (Zheng et al. 2016), a technique called *stability training* is proposed to improve the robustness of image classification, and which can be applied to our classification policies as well. *Stability training* intuition resides on the idea that if x' is a small *perturbation* of the state x, then we also desire that $\delta(x')$ and $\delta(x)$ be close enough. The authors propose an auxiliary *stability objective* function that depends on a metric on the policy output space, thus forcing the classification of close states to be close as well. In the market maker problem, we suggest using the metric implied by the wealth utility function for the purposes of formulating this stability objective.

[23] Copulas may also be used to model a non-linear dependence structure

Fundamentally, *stability training* reinforces local stability, but not local consistency, i.e., the state x' might be marginally better than x, but the policy might still recommend an action that will mostly likely end up in x than the marginally better x'. In this respect, the *monotonicity in the wealth utility* help us achieve *global consistency*, because it is able to enforce the *risk-reward preferences* not only locally but also globally, for example, between widely 'distant states'.

4.4.1 Monotonicity in wealth utility

The problem of calibrating a value function while enforcing *monotonicity in the wealth utility* for a given architecture defined on the basis functions ϕ can be described as

$$\theta^v = \underset{\theta}{\operatorname{argmin}} \sum_x \|\hat{V}(x) - \bar{V}[\theta, \phi](x)\|^2 \tag{24}$$

subject to the following variational inequality constraint

$$d\bar{V}(x) dU(x) \geq 0 \qquad \text{for all } x \in \mathcal{X} \tag{25}$$

where $\hat{V}$ is some estimate we want to calibrate $\bar{V}$ to, and $d\bar{V}(x)$ and $dU(x)$ are the total differentials of the approximation architecture $\bar{V}$ and the wealth utility function U respectively

$$\begin{aligned} d\bar{V}(x) &= \frac{\partial \bar{V}}{\partial y} dy + \frac{\partial \bar{V}}{\partial s} ds + \frac{\partial \bar{V}}{\partial p} dp + \frac{\partial \bar{V}}{\partial q} dq + \frac{\partial \bar{V}}{\partial h} dh \\ dU(x) &= \frac{\partial U}{\partial y} dy + \frac{\partial U}{\partial s} ds + \frac{\partial U}{\partial p} dp + \frac{\partial U}{\partial q} dq + \frac{\partial U}{\partial h} dh \end{aligned} \tag{26}$$

where we assume $\bar{V}$ and U are differentiable and continuous in each of the state variables y, s, p, q and h.[24]

The idea of such constraint is that we want the total derivative of U and $\bar{V}$ to have the same sign when evaluated at the same state x: if we move the state x is any fixed direction, then *monotonicity in the wealth* means that if U increases (decreases) in that direction, then $\bar{V}$ must also increase (decrease) in that same direction.

The above problem can be reduced to a *PDE-constrained non-linear program* formulation. In order to see this, if we multiply expressions (26), then the variational inequality condition (25) can be expressed as follows:

$$dx^T \begin{bmatrix} \frac{\partial \bar{V}}{\partial y}\frac{\partial U}{\partial y} & \frac{\partial \bar{V}}{\partial y}\frac{\partial U}{\partial s} & \frac{\partial \bar{V}}{\partial y}\frac{\partial U}{\partial p} & \frac{\partial \bar{V}}{\partial y}\frac{\partial U}{\partial q} & \frac{\partial \bar{V}}{\partial y}\frac{\partial U}{\partial h} \\ \frac{\partial \bar{V}}{\partial s}\frac{\partial U}{\partial y} & \frac{\partial \bar{V}}{\partial s}\frac{\partial U}{\partial s} & \frac{\partial \bar{V}}{\partial s}\frac{\partial U}{\partial p} & \frac{\partial \bar{V}}{\partial s}\frac{\partial U}{\partial q} & \frac{\partial \bar{V}}{\partial s}\frac{\partial U}{\partial h} \\ \frac{\partial \bar{V}}{\partial p}\frac{\partial U}{\partial y} & \frac{\partial \bar{V}}{\partial p}\frac{\partial U}{\partial s} & \frac{\partial \bar{V}}{\partial p}\frac{\partial U}{\partial p} & \frac{\partial \bar{V}}{\partial p}\frac{\partial U}{\partial q} & \frac{\partial \bar{V}}{\partial p}\frac{\partial U}{\partial h} \\ \frac{\partial \bar{V}}{\partial q}\frac{\partial U}{\partial y} & \frac{\partial \bar{V}}{\partial q}\frac{\partial U}{\partial s} & \frac{\partial \bar{V}}{\partial q}\frac{\partial U}{\partial p} & \frac{\partial \bar{V}}{\partial q}\frac{\partial U}{\partial q} & \frac{\partial \bar{V}}{\partial q}\frac{\partial U}{\partial h} \\ \frac{\partial \bar{V}}{\partial h}\frac{\partial U}{\partial y} & \frac{\partial \bar{V}}{\partial h}\frac{\partial U}{\partial s} & \frac{\partial \bar{V}}{\partial h}\frac{\partial U}{\partial p} & \frac{\partial \bar{V}}{\partial h}\frac{\partial U}{\partial q} & \frac{\partial \bar{V}}{\partial h}\frac{\partial U}{\partial h} \end{bmatrix} dx \geq 0 \tag{27}$$

[24] Choosing a non-differentiable or discontinuous function as one of the basis functions in ϕ implies that we no longer can propose this form of strong monotonicity,and must resort to a more complex and/or weaker form of monotonicity conditions

for all vectors of differentials $dx = \begin{bmatrix} dy & ds & dp & dq & dh \end{bmatrix}^T$, i.e., the above matrix must be positive semi-definite.

Note, for example, that on a change in the y direction only (varying the cash while fixing the rest of the state variables), we have $dx = \begin{bmatrix} dy & 0 & 0 & 0 & 0 \end{bmatrix}^T$ and the positive semi-definite condition resumes to $\frac{\partial \bar{V}}{\partial y}\frac{\partial U}{\partial y} \geq 0$. Thus by varying each state variable while fixing the others, we consequentially obtain the following elementary first order conditions:

$$\frac{\partial \bar{V}}{\partial x_i}\frac{\partial U}{\partial x_i} \geq 0$$

for $i = 1, 2, \cdots, 5$, where x_i represents the i-th state variable. Now if we vary, for example, the state variables x_i and x_j while fixing the rest, the positive definite condition become second order cross conditions:

$$\frac{\partial \bar{V}}{\partial x_i}\frac{\partial U}{\partial x_i} + \frac{\partial \bar{V}}{\partial x_j}\frac{\partial U}{\partial x_j} \pm \left(\frac{\partial \bar{V}}{\partial x_i}\frac{\partial U}{\partial x_j} + \frac{\partial \bar{V}}{\partial x_j}\frac{\partial U}{\partial x_i}\right) \geq 0$$

where the $\pm$ comes from the fact that dx_i and dx_j may have opposite signs. Further constraints can be obtained in this fashion, and a weaker form of monotonicity can also be formulated by choosing only a subset of all those conditions derived from the positive semi-definiteness of the matrix in (27).

To perform this optimization, we use the *First-order Augmented Lagrangian* method, by changing the objective (24) as follows:

$$\theta^v = \underset{\theta}{\operatorname{argmin}} \sum_x \|\hat{V}_i(x) - \bar{V}[\theta](x)\|^2 - \lambda^x \cdot (f(\theta, x) - z) + \frac{c_i}{2}\|f(\theta, x) - z\|^2 \tag{28}$$

where $f(\theta^v, x) \geq 0$ are the PDE-constrains derived from the monotonicity conditions, λ^x is the vector of *Lagrange multipliers* (also known as *adjoint state*), $\frac{c_i}{2}\|f(\theta, x) - z\|^2$ is a regularization term and $z \geq 0$ are some slack variables. It is valid to note that the conditions must hold for every point x in the sample X^*. The method works by fixing the *adjoint state* λ and performing the minimization by gradient descent using the gradient of the *Lagrangian Augmented* objective (28). Once the minimum θ^v is found, then the adjoin is updated:

$$\lambda^x_{i+1} \leftarrow \lambda^x_i - c_i f(\theta^{v*}, x)$$

and the process is repeated until convergence. For more on subject of *PDE-constrained non-linear optimization*, we suggest Bartholomew-Biggs (2005), Diwekar (2008), Ito & Kunisch (2008), Haber & Hanson (2007), Hinze (2009) and De Los Reyes (2015).

5 Preliminary results

Given the diversity of approaches and the fact that training models is a time-consuming task, we cannot present here a full analysis of all algorithms discussed or presented. We instead provide a qualitative analysis derived from preliminary results and pitfalls found while trying to implement them. A more complete analysis is deserved, and , and practitioners applying reinforcement learning to market making should focus on the following aspects:

(i) *Convergence stability*: Run each of the algorithms a number of times under different initial guesses for θ_0^v and verify if they all converge to the same solution, or if there is any divergence. Compare the solutions of those convergent algorithms.

(ii) *Convergence speed*: Analyze how quickly the algorithms converge, on average.

(iii) *Simulation analysis*: Simulate the expected total reward for each learned policy by running them against Monte Carlo simulations of the limit order book model proposed.

(iv) *Backtesting analysis*: Perform cross validation by backtesting the policies. Replaying historical tick data not only helps to identify behavior around news or major economic events, but also helps understanding in which circumstances each policy is the best.

(v) *Qualitative analysis*: Investigate the learned the policies under a few basic scenarios for which rational judgments can be made.

In our implementations, we have used a risk-neutral utility function, $U(x) = v(x)$ in order to make qualitative judgments on the various approaches. We made use of the Julia language (Bezanson et al. 2017) alongside the packages ForwardDiff (Revels et al. 2016) and Flux (Innes 2018), which proved incredible flexible and productive. ForwardDiff provides automatic differentiation, which helped us derive gradients and jacobians without having to resort to finite element methods. Flux is a Machine Learning framework entirely developed in Julia, which although on its early stages, it already offers impressive functionality.

For quick results, the value iteration methods performed better, as it is easy to train only one model, when compared to actor-critic models, where we need to train both actor and critic. Value functions approximations had about the same qualitative characteristics of Action-Value function approximations, but the former is faster to train: the extra degree of freedom of the action-value function $Q(x_t, a_t)$ makes their training slower. Between the VFA methods, the one that rendered the best results was the TD(λ) methods - their convergence is proved by Sutton (1988). The TD method proved quick to detect trends in the price and the greedy policies generated take this opportunity to generated P&L.

Also, gradient methods provided better convergence and stable results, when compared to projection methods. Most of the projection methods we implemented and tested failed to convergence, with some convergence when using supports.

With respect to Policy iteration, they are more sophisticated but slower to train. Stochastic policy gradient hardly complies with our requirement for policy stability. Deterministic Policy Gradients are simple to implement, but the policy produced by linear architectures generated negative quoting spreads δ^a and δ^b, thus not complying with policy validity requirements. We tried to use Deep Neural Networks for both actor and critic with the *Deep Deterministic Policy Gradients* (DDPG) method (Lillicrap et al. 2016) using activator functions that constrain δ^a and δ^b between 1 and 20 basis points and ξ between -5 and 5 shares, but the model proved extremely slow to train in the commodity hardware available, and the policy parameters moved very slowly. The first models generated this way produced a 'cheating' policy by quoting 20 basis points (the farther possible from the reference price, thus effectively not adding liquidity) while hedging constantly according to the detected trend. Since algorithmic trading problems are highly constrained, we might consider constrained policy gradient methods as in Achiam et al. (2017) and Geibel & Wysotzki (2005), Geibel (2006). It is also worth to investigate designing reinforcements that generate market making policies that actually add liquidity instead of just speculative policies.

One thing that we note here is that, in practice, the utility function U, which coincides with the reward function R, are also constantly evolving - investor preferences are directly affected by parameters like alphas, volatilities. They must be calibrated simultaneously alongside the value function.

Example of Bad Convergence The following figures illustrate the bid and offer proposed for different configurations of constant wealth positions, per quantity in the ETF q and in the underlying h. As we can see, bad convergence leads to clearly wrong policies.

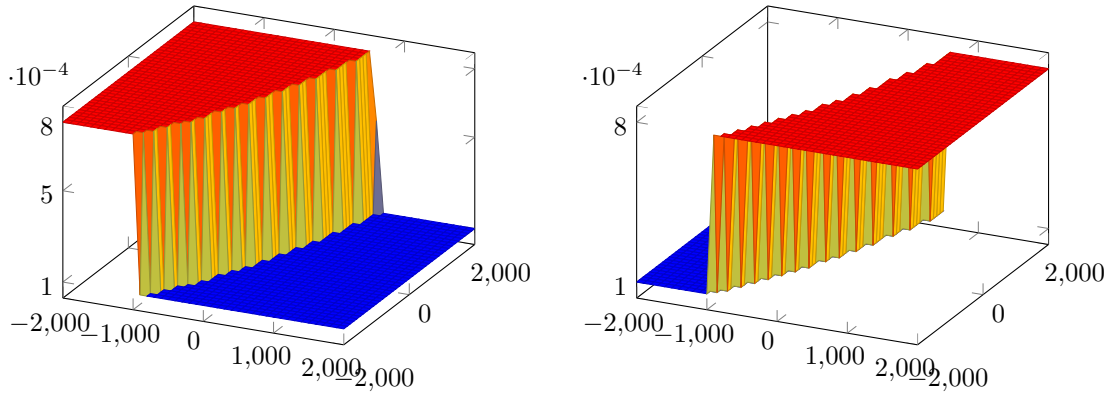

Figure 1: Bid-Offer spreads from policy implied by value function approximation (bad convergence)

The resulting value function display tendency to favor extreme positions, clearly displaying lack of equilibrium - greedy policies derived from such policy would actual rogue trading algorithms.

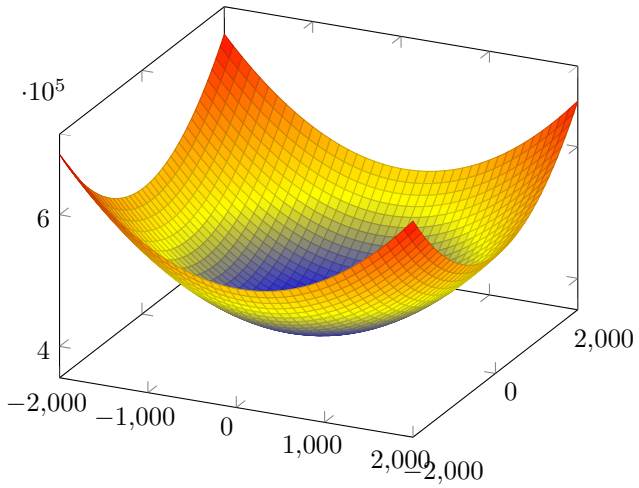

Figure 2: Value function approximation for constant wealth (bad convergence)

Example of Good Convergence The following figures display the value of a null position and a hedged position, as the learning progresses, under TD(0.3). Over the long term, the value tends to roughly the same values, regardless of initial conditions, and any differences would account for a preference between being hedging (by crossing the spread on the underlying) or liquidating (unwinding the position by trading the ETF itself).

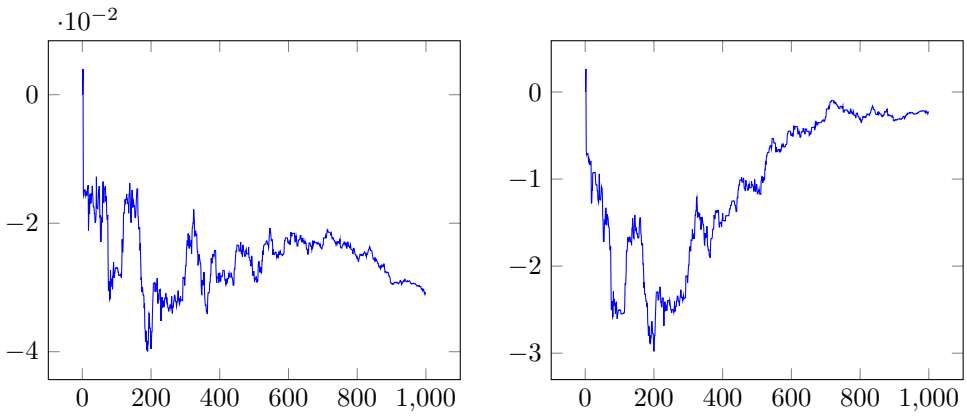

Figure 3: Convergence of the value V^* for the null position [0,100,0,0,0] and a hedged position [0,100,0,2000,-2000]

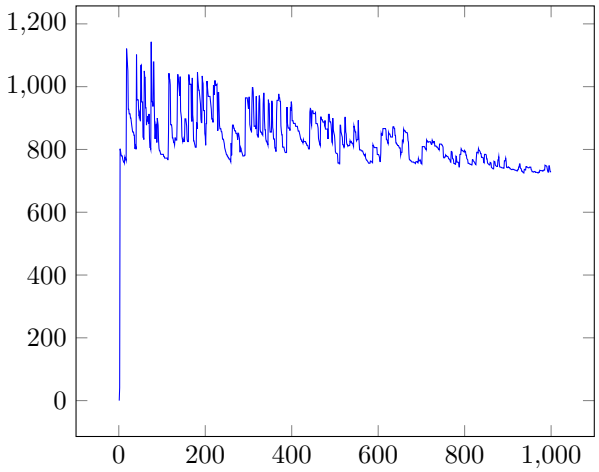

Figure 4: Convergence of the value V^* for $1000 null position [1000,100,0,0,0]

Next, we can observe that the TD(0.3) policies recommend a constant spread of 1 basis point for both the bid and the offer, while opportunistically hedging any downside in order to maximize profits.

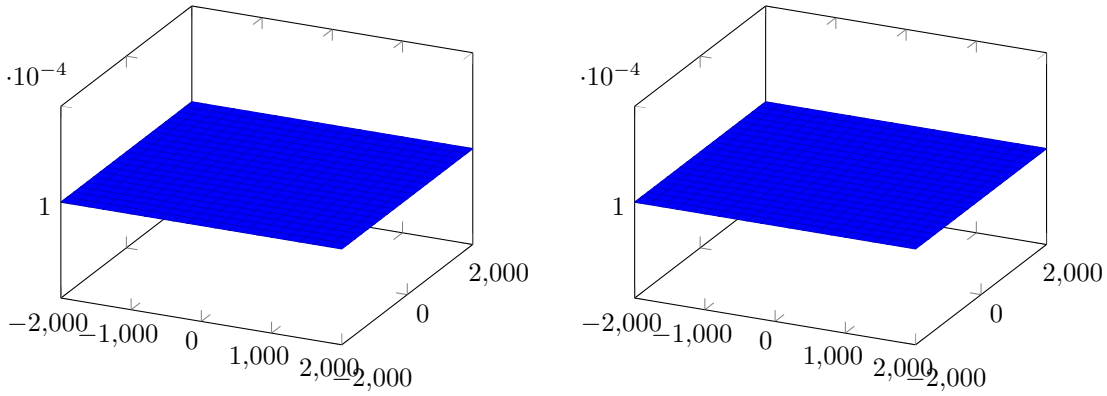

Figure 5: Bid-Offer spreads from TD(0.3) policy for constant wealth case

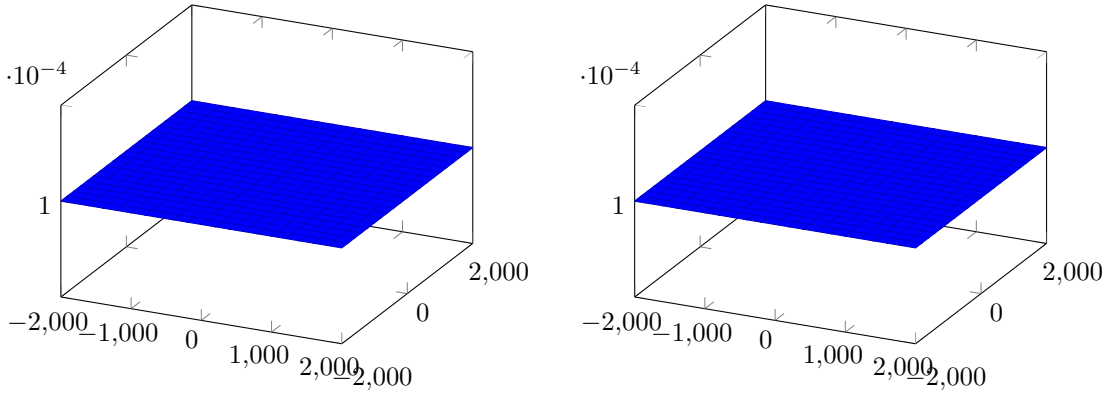

Figure 6: Bid-Offer spreads from TD(0.3) policy for zero cash case

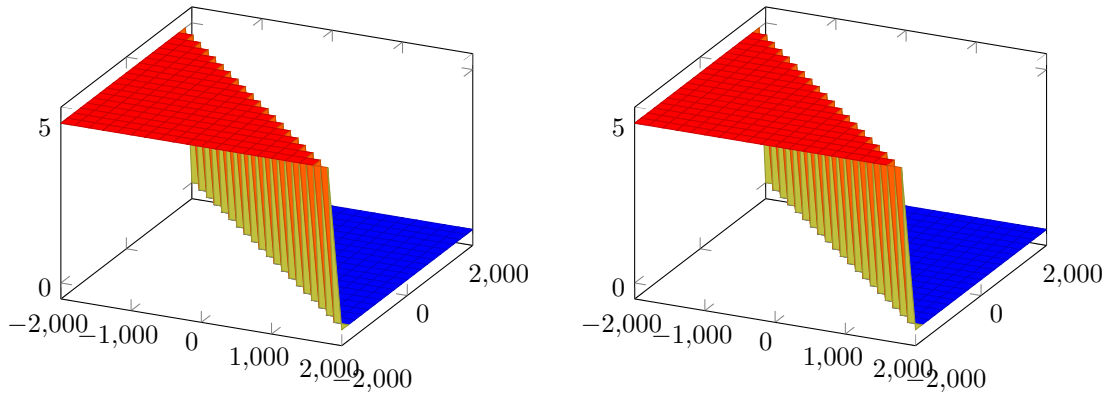

Figure 7: Hedging in TD(0.3) policy for constant wealth and the zero cash cases

Conclusion

The goal of this thesis was to provide the reader with the step-by-step thought process around the complexity of market making problems grounded on a solid mathematical and computational framework, and how to solve them using Reinforcement Learning methods. In parallel, we tried to foment a formal discussion on mathematical models for ETFs, usually treated pragmatically but in a somewhat limited way by the industry, while inviting academics to contemplate other surprisingly interesting financial products. Reinforcement Learning has been a very solid area of research with many decades of published works, which are now seeing successful practical applications in diverse areas, thanks in large scale to the continuous increasing of computational power and the availability of massive datasets. There many promises but also many pitfalls due to the very particular nature of algorithmic trading problems. Understanding convergence and robustness of the obtained policies are essential to the successful applicability of RL in high frequency trading and electronic market making. We hope that the arguments and directions that we provided are able to help paving the way for future applications in finance.

References

Achiam, J., Held, D., Tamar, A. & Abbeel, P. (2017), Constrained Policy Optimization, *in* 'arXiv:1705.10528 [cs]', Sydney. arXiv: 1705.10528.
URL: *http://arxiv.org/abs/1705.10528*

Alexander, C. (2008), *Market Risk Analysis*, Vol. III, John Wiley & Sons.

Almgren, R. & Chriss, N. (2001), 'Optimal execution of portfolio transactions', *The Journal of Risk* **3**(2), 5–39.
URL: *http://www.risk.net/journal-of-risk/technical-paper/2161150/optimal-execution-portfolio-transactions*

Antipin, A. S., Jaćimović, M. & Mijajlović, N. (2018), 'Extragradient method for solving quasivariational inequalities', *Optimization* **67**(1), 103–112.
URL: *https://www.tandfonline.com/doi/full/10.1080/02331934.2017.1384477*

Arnold, B. C., Balakrishnan, N., Castillo, E. & Sarabia, J.-M., eds (2006), *Advances in distribution theory, order statistics, and inference*, Statistics for industry and technology, Birkhäuser, Boston. OCLC: ocm70249068.

Avellaneda, M. & Stoikov, S. (2008), 'High-frequency trading in a limit order book', *Quantitative Finance* **8**(3), 217–224.
URL: *http://www.tandfonline.com/doi/abs/10.1080/14697680701381228*

Azimzadeh, P. (2017), 'Impulse Control in Finance: Numerical Methods and Viscosity Solutions', *arXiv:1712.01647 [math]* . arXiv: 1712.01647.
URL: *http://arxiv.org/abs/1712.01647*

Azimzadeh, P., Bayraktar, E. & Labahn, G. (2017), 'Convergence of implicit schemes for Hamilton-Jacobi-Bellman quasi-variational inequalities', *arXiv:1705.02922 [math]* . arXiv: 1705.02922.
URL: *http://arxiv.org/abs/1705.02922*

Bartholomew-Biggs, M. C. (2005), *Nonlinear optimization with financial applications*, Kluwer, Boston. OCLC: ocm58522886.

Bellman, R. (1957), *Dynamic Programming*, Princenton University Press, New Jersey.

Bensoussan, A. & Lions, J. L. (1984), *Impulse Control and Quasi-Variational Inequalities*, 1st edn, Gauthier-Villars, Paris.

Bertsekas, D. P. (2013), Rollout Algorithms for Discrete Optimization: A Survey, *in* P. M. Pardalos, D.-Z. Du & R. L. Graham, eds, 'Handbook of Combinatorial Optimization', Springer New York, New York, NY, pp. 2989–3013.
URL: *http://link.springer.com/10.1007/978-1-4419-7997-1_8*

Bertsekas, D. P. & Ioffe, S. (1996), 'Temporal Differences-Based Policy Iteration and Applications in Neuro-Dynamic Programming'.
URL: *http://www.mit.edu/ dimitrib/Tempdif.pdf*

Bertsekas, D. P., Tsitsiklis, J. N. & Wu, C. (1997), 'Rollout Algorithms for Combinatorial Optimization', *Journal of Heuristics* p. 18.

Bertsekas, D. & Tsitsiklis, J. (1995), Neuro-dynamic programming: an overview, Vol. 1, IEEE, pp. 560–564.
URL: *http://ieeexplore.ieee.org/document/478953/*

Bertsimas, D., Lauprete, G. J. & Samarov, A. (2004), 'Shortfall as a risk measure: properties, optimization and applications', *Journal of Economic Dynamics and Control* **28**(7), 1353–1381.
URL: *http://linkinghub.elsevier.com/retrieve/pii/S016518890300109X*

Bezanson, J., Edelman, A., Karpinski, S. & Shah, V. (2017), 'Julia: A Fresh Approach to Numerical Computing', *SIAM Review* **59**(1), 65–98.
URL: *https://doi.org/10.1137/141000671*

Bouchaud, J. P. (2009), 'Price Impact', *arXiv:0903.2428 [q-fin]* . arXiv: 0903.2428.
URL: *http://arxiv.org/abs/0903.2428*

Boyan, J. A. & Moore, A. W. (1995), 'Generalization in Reinforcement Learning: Safely Approximating the Value Function', *Advances in Neural Information Processing Systems 7* pp. 369–376.

Bradtke, S. J. & Barto, A. G. (1996), 'Linear Least-Squares Algorithms for Temporal Difference Learning', *Machine Learning* **22**, 33–57.

Brooks, C., Hinich, M. J. & Patterson, D. M. (2003), 'Intraday Patterns in the Returns, Bid-ask Spreads, adn Trading Volume of Stocks Traded on the New York Stock Exchange'.
URL: *http://www.la.utexas.edu/hinich/files/Economics/Signal-NYSE.pdf*

Bruder, B. & Pham, H. (2009), 'Impulse control problem on finite horizon with execution delay', *Stochastic Processes and their Applications* **119**(5), 1436–1469.
URL: *http://linkinghub.elsevier.com/retrieve/pii/S0304414908001208*

Buşoniu, L., De Schutter, B. & Babuška, R. (2010), Approximate Dynamic Programming and Reinforcement Learning, *in* J. Kacprzyk, R. Babuška & F. C. A. Groen, eds, 'Interactive Collaborative Information Systems', Vol. 281, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 3–44.
URL: *http://link.springer.com/10.1007/978-3-642-11688-9_1*

De Farias, D. P. & Van Roy, B. (2000), 'On the Existence of Fixed Points for Approximate Value Iteration and Temporal-Difference Learning', *Journal of Optimization Theory and Applications* **105**(3), 589–608.
URL: *http://link.springer.com/10.1023/A:1004641123405*

De Los Reyes, J. C. (2015), *Numerical PDE-constrained optimization*, SpringerBriefs in Optimization, Springer-Verlag, Cham. OCLC: 923520912.

Di Graziano, G. (2014), 'Optimal Trading Stops and Algorithmic Trading', *SSRN Electronic Journal* .
URL: *http://www.ssrn.com/abstract=2381830*

Diwekar, U. (2008), *Introduction to applied optimization*, number 22 *in* 'Springer optimization and its applications', 2nd edn, Springer, New York. OCLC: 604380439.

ETF.com (2018), 'ETF.com: Find the Right ETF - Tools, Ratings, News'.
URL: *https://www.etf.com/*

Facchinei, F., Kanzow, C. & Sagratella, S. (2014), 'Solving quasi-variational inequalities via their KKT conditions', *Mathematical Programming* **144**(1-2), 369–412.
URL: *http://link.springer.com/10.1007/s10107-013-0637-0*

Farahmand, A.-m. (2011), 'Regularization in Reinforcement Learning'.

Garman, M. B. (1976), 'Market microstructure', *Journal of Financial Economics* **3**(3).
URL: *https://www.sciencedirect.com/science/article/pii/0304405X76900064*

Geibel, P. (2006), Reinforcement Learning for MDPs with Constraints, *in* D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, J. Fürnkranz, T. Scheffer & M. Spiliopoulou, eds, 'Machine Learning: ECML 2006', Vol. 4212, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 646–653.
URL: *http://link.springer.com/10.1007/11871842_63*

Geibel, P. & Wysotzki, F. (2005), 'Risk-Sensitive Reinforcement Learning Applied to Control under Constraints', *Journal of Artificial Intelligence Research* **24**, 81–108. arXiv: 1109.2147.
URL: *http://arxiv.org/abs/1109.2147*

Geist, M. & Pietquin, O. (2010), A Brief Survey of Parametric Value Function Approximation, Technical report, Supelec.

Geist, M. & Scherrer, B. (2012), L^1-Penalized Projected Bellman Residual, *in* S. Sanner & M. Hutter, eds, 'Recent Advances in Reinforcement Learning', Springer Berlin Heidelberg, pp. 89–101.

Gordon, G. J. (1995), 'Stable Function Approximation in Dynamic Programming'.
URL: *http://www.cs.cmu.edu/ ggordon/ml95-stable-dp.pdf*

Gordon, G. J. (1996), 'Stable Fitted Reinforcement Learning', *Advances in Neural Information Processing Systems 8* pp. 1052–1058.

Grondman, I., Busoniu, L., Lopes, G. A. D. & Babuska, R. (2012), 'A Survey of Actor-Critic Reinforcement Learning: Standard and Natural Policy Gradients', *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* **42**(6), 1291–1307.
URL: *http://ieeexplore.ieee.org/document/6392457/*

Guilbaud, F. & Pham, H. (2011), 'Optimal High Frequency Trading with limit and market orders', *arXiv:1106.5040 [cs, math, q-fin]* . arXiv: 1106.5040.
URL: *http://arxiv.org/abs/1106.5040*

Guilbaud, F. & Pham, H. (2012), 'Optimal High Frequency Trading in a Pro-Rata Microstructure with Predictive Information', *arXiv:1205.3051 [q-fin]* . arXiv: 1205.3051.
URL: *http://arxiv.org/abs/1205.3051*

Gundel, A. & Weber, S. (2008), 'Utility maximization under a shortfall risk constraint', *Journal of Mathematical Economics* **44**(11), 1126–1151.
URL: *http://linkinghub.elsevier.com/retrieve/pii/S0304406808000086*

Haber, E. & Hanson, L. (2007), 'Model Problems in PDE-Constrained Optimization'.

Hanson, F. B. & Westman, J. J. (2002), 'Jump-Diffusion Stock Return Models in Finance: Stochastic Process Density with Uniform-Jump Amplitude'.
URL: *http://homepages.math.uic.edu/ hanson/pub/MTNS2002/mtns02fmt-cdpaper.pdf*

Hilliard, J. (2014), 'Premiums and discounts in ETFs: An analysis of the arbitrage mechanism in domestic and international funds', *Global Finance Journal* **25**(2), 90–107.
URL: *http://linkinghub.elsevier.com/retrieve/pii/S1044028314000167*

Hinze, M., ed. (2009), *Optimization with PDE constraints*, number 23 *in* 'Mathematical modelling', Springer, Dordrecht. OCLC: 255277417.

Ho, T. S. Y. & Stoll, H. R. (1983), 'The Dynamics of Dealer Markets Under Competition', *The Journal of Finance* **38**(4), 1053.
URL: *https://www.jstor.org/stable/2328011?origin=crossref*

Ho, T. & Stoll, H. R. (1980), 'On Dealer Markets Under Competition', *The Journal of Finance* **35**(2), 259.
URL: *https://www.jstor.org/stable/2327382?origin=crossref*

Ho, T. & Stoll, H. R. (1981), 'Optimal dealer pricing under transactions and return uncertainty', *Journal of Financial Economics* **9**(1), 47–73.
URL: *http://linkinghub.elsevier.com/retrieve/pii/0304405X81900209*

Innes, M. (2018), 'Flux: Elegant Machine Learning with Julia', *Journal of Open Source Software* .

Ito, K. & Kunisch, K. (2008), *Lagrange Multiplier Approach to Variational Problems and Applications*, Advances in Design and Control, SIAM, Philadelphia.

Ivanov, S. (2017), 'Comparative Analysis of ETF and Common STock Intraday Bid-Ask Spread Behavior', *Economics Bulletin* **37**(2), 723–732.
URL: *http://www.accessecon.com/Pubs/EB/2017/Volume37/EB-17-V37-I2-P66.pdf*

Iwatsubo, K., Watkins, C. & Xu, T. (2017), 'Intraday Seasonality in Efficiency, Liquidity, Volatility and Volume: Platinum and Gold Futures in Tokyo and New York', *SSRN Electronic Journal* .
URL: *https://www.ssrn.com/abstract=3021533*

Korn, R. (1999), 'Some applications of impulse control in mathematical finance', *Mathematical Methods of Operations Research* pp. 493–518.

Lai, T. L. & Lim, T. W. (2003), 'Singular Stochastic Control in Optimal Investment and Hedging in the Presence of Transaction Costs', *Lecture Notes-Monograph Series* **41**, 209–228.
URL: *http://www.jstor.org/stable/4356216*

Leung, T. & Zhang, H. (2017), 'Optimal Trading with a Trailing Stop', *arXiv:1701.03960 [q-fin]* . arXiv: 1701.03960.
URL: *http://arxiv.org/abs/1701.03960*

Levy, A. & Lieberman, O. (2013), 'Overreaction of country ETFs to US market returns: Intraday vs. daily horizons and the role of synchronized trading', *Journal of Banking & Finance* **37**(5), 1412–1421.
URL: *http://linkinghub.elsevier.com/retrieve/pii/S0378426612000866*

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D. & Wierstra, D. (2016), 'Continuous control with deep reinforcement learning', *International Conference on Learning Representations* . arXiv: 1509.02971.
URL: *http://arxiv.org/abs/1509.02971*

Lizotte, D. J. (2011), 'Convergent Fitted Value Iteration with Linear Function Approximation', *Advances in Neural Information Processing Systems 24* pp. 2537–2545.

Markowitz, H. (1952), 'Portfolio Selection', *The Journal of Finance* **7**(1), 77–91.

Merton, R. C. (1969), 'Lifetime Portfolio Selection under Uncertainty: The Continuous-Time Case', *The Review of Economics and Statistics* **51**(3), 247.
URL: *https://www.jstor.org/stable/1926560?origin=crossref*

Merton, R. C. (1971), 'Optimum consumption and portfolio rules in a continuous-time model', *Journal of Economic Theory* **3**(4), 373–413.
URL: *http://linkinghub.elsevier.com/retrieve/pii/002205317190038X*

Miller, C. W. & Yang, I. (2015), 'Optimal Control of Conditional Value-at-Risk in Continuous Time', *arXiv:1512.05015 [cs, math, q-fin]* . arXiv: 1512.05015.
URL: *http://arxiv.org/abs/1512.05015*

Mudchanatongsuk, S., Primbs, J. A. & Wong, W. (2008), Optimal pairs trading: A stochastic control approach, IEEE, pp. 1035–1039.
URL: *http://ieeexplore.ieee.org/document/4586628/*

O'Hara, M. & Oldfield, G. S. (1986), 'The Microeconomics of Market Making', *The Journal of Financial and Quantitative Analysis* **21**(4), 361.
URL: *https://www.jstor.org/stable/2330686?origin=crossref*

Perkins, T. J. & Precup, D. (2003), A Convergent Form of Approximate Policy Iteration, *in* S. Becker, S. Thrun & K. Obermayer, eds, 'Advances in Neural Information Processing Systems 15', MIT Press, pp. 1627–1634.
URL: *http://papers.nips.cc/paper/2143-a-convergent-form-of-approximate-policy-iteration.pdf*

Powell, W. B. (2009), 'What you should know about approximate dynamic programming', *Naval Research Logistics* **56**(3), 239–249.
URL: *http://doi.wiley.com/10.1002/nav.20347*

Powell, W. B. (2014), Clearing the Jungle of Stochastic Optimization, *in* A. M. Newman, J. Leung, J. C. Smith & H. J. Greenberg, eds, 'Bridging Data and Decisions', INFORMS, pp. 109–

137.
URL: *http://pubsonline.informs.org/doi/abs/10.1287/educ.2014.0128*

Prigent, J.-L. (2007), *Portfolio optimization and performance analysis*, Chapman & Hall/CRC, Boca Raton, Fla. OCLC: 938370221.

Proxy / Cross Hedging (2011).
URL: *https://quantivity.wordpress.com/2011/10/02/proxy-cross-hedging/*

Revels, J., Lubin, M. & Papamarkou, T. (2016), 'Forward-Mode Automatic Differentiation in Julia', *arXiv:1607.07892 [cs.MS]* .
URL: *https://arxiv.org/abs/1607.07892*

Samuelson, P. A. (1969), 'Lifetime Portfolio Selection By Dynamic Stochastic Programming', *The Review of Economics and Statistics* **51**(3), 239.
URL: *https://www.jstor.org/stable/1926559?origin=crossref*

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D. & Riedmiller, M. (2014), Deterministic Policy Gradient Algorithms, *in* 'Proceedings of the 31st International Conference on Machine learning - ICML '14', Vol. 32, Beijing, pp. 387–395.

Spooner, T., Fearnley, J., Savani, R. & Koukorinis, A. (2018), 'Market Making via Reinforcement Learning', *Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems* p. 11.

Stoll, H. R. (1978), 'The Supply of Dealer Services in Securities Markets', *The Journal of Finance* **33**(4), 1133–1151.
URL: *http://www.jstor.org/stable/2326945*

Sutton, R. S. (1988), 'Learning to Predict by the Methods of Temporal Differences', *Machine Learning* **3**, 9–44.

Sutton, R. S. & Barto, A. G. (2018), *Reinforcement learning: an introduction*, Adaptive computation and machine learning, 2nd edition edn, MIT Press, Cambridge, Mass.

Sutton, R. S., McAllester, D. A., Singh, S. P. & Mansour, Y. (1999), 'Policy Gradient Methods for Reinforcement Learning with Function Approximation', *Advances in Neural Information Processing Systems 12* pp. 1057–1063.

Veraart, L. A. M. (2010), 'Optimal Market Making in the Foreign Exchange Market', *Applied Mathematical Finance* **17**(4), 359–372.
URL: *http://www.tandfonline.com/doi/abs/10.1080/13504860903387588*

Williams, R. J. (1992), 'Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning', *Machine Learning* **8**, 229–256.

Williams, R. J. & Baird, L. (1993), Tight Performance Bounds on Greedy Policies Based on Imperfect Value Functions, Technical Report NU-CCS-93-14.
URL: *http://leemon.com/papers/1993wb2.pdf*

Zang, P., Irani, A. J., Zhou, P., Jr, C. L. I. & Thomaz, A. L. (2010), 'Using Training Regimens to Teach Expanding Function Approximators', *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems* .

Zheng, S., Song, Y., Leung, T. & Goodfellow, I. (2016), 'Improving the Robustness of Deep Neural Networks via Stability Training', *arXiv:1604.04326 [cs]* . arXiv: 1604.04326.
URL: *http://arxiv.org/abs/1604.04326*

Øksendal, B. K. & Sulem, A. (2007), *Applied stochastic control of jump diffusions*, Universitext, 2. ed edn, Springer, Berlin. OCLC: 255486987.