massachusetts institute of technology — artificial intelligence laboratory

An Electronic Market-Maker

Nicholas Tung Chan and Christian Shelton

AI Memo 2001-005 April 17, 2001
CBCL Memo 195

Abstract

This paper presents an adaptive learning model for market-making under the reinforcement learning framework. Reinforcement learning is a learning technique in which agents aim to maximize the long-term accumulated rewards. No knowledge of the market environment, such as the order arrival or price process, is assumed. Instead, the agent learns from real-time market experience and develops explicit market-making strategies, achieving multiple objectives including the maximizing of profits and minimization of the bid-ask spread. The simulation results show initial success in bringing learning techniques to building market-making algorithms.

This report describes research done within the Center for Biological and Computational Learning in the Department of Brain and Cognitive Sciences and in the Artificial Intelligence Laboratory at the Massachusetts Institute of Technology. This research was sponsored by grants from: Office of Naval Research under contract No. N00014-93-1-3085, Office of Naval Research (DARPA) under contract No. N00014-00-1-0907, National Science Foundation (ITR) under contract No. IIS-0085836, National Science Foundation (KDI) under contract No. DMS-9872936, and National Science Foundation under contract No. IIS-9800032 This research was partially funded by the Center for e-Business (MIT). Additional support was provided by: Central Research Institute of Electric Power Industry, Eastman Kodak Company, DaimlerChrysler AG, Compaq, Honda R&D Co., Ltd., Komatsu Ltd., Merrill-Lynch, NEC Fund, Nippon Telegraph & Telephone, Siemens Corporate Research, Inc., and The Whitaker Foundation.

1 Introduction

Many theoretical market-making models are developed in the context of stochastic dynamic programming. Bid and ask prices are dynamically determined to maximize some long term objectives such as expected profits or expected utility of profits. Models in this category include those of Ho & Stoll (1981), O'Hara & Oldfiled (1986) and Glosten & Milgrom (1985). The main limitation of these models is that specific properties of the underlying processes (price process and order arrival process) have to be assumed in order to obtain a closed-form characterization of strategies.

This paper presents an adaptive learning model for market-making using reinforcement learning under a simulated environment. Reinforcement learning can be considered as a model-free approximation of dynamic programming. The knowledge of the underlying processes is not assumed but learned from experience. The goal of the paper is to model the market-making problem in a reinforcement learning framework, explicitly develop market-making strategies, and discuss their performance. In the basic model, where the market-maker quotes a single price, we are able to determine the optimum strategies analytically and show that reinforcement algorithms successfully converge to these strategies. The major challenges of the problem are that the environment state is only partially observable and reward signals may not be available at each time step. The basic model is then extended to allow the market-maker to quote bid and ask prices. While the market-maker affects only the direction of a price in the basic model, it has to consider both the direction of the prices as well as the size of the bid-ask spreads in the extended model. The reinforcement algorithm converges to correct policies and effectively control the trade-off between profit and market quality in terms of the spread.

This paper starts with an overview of several important theoretical market-making models and an introduction of the reinforcement learning framework in Section 2. Section 3 establishes a reinforcement learning market-making model. Section 4 presents a basic simulation model of a market with asymmetric information where strategies are studied analytically and through the use of reinforcement learning. Section 5 extends the basic model to incorporate additional actions, states, and objectives for more realistic market environments.

2 Background

2.1 Market-making Models

The understanding of the price formation process in security markets has been one of the focal points of the market microstructure literature. There are two main approaches to the market-making problem. One focuses on the uncertainties of an order flow and the inventory holding risk of a market-maker. In a typical inventory-based model, the market-maker sets the price to balance demand and supply in the market while actively controlling its inventory holdings. The second approach attempts to explain the price setting dynamics employing the role of information. In information-based models, the market-maker faces traders with superior information. The market-maker makes inferences from the orders and sets the quotes. This informational disadvantage is reflected in the bid-ask spread.

Garman (1976) describes a model in which there is a single, monopolistic, and risk neutral market-maker who sets prices, receives all orders, and clears trades. The dealer's objective is to maximize expected profit per unit time. Failure of the market-maker arises when the it runs out of either inventory or cash. Arrivals of buy and sell orders are characterized by two independent Poisson processes whose arrival rates depend on the market-maker's quotes. Essentially the collective activity of the traders is modeled as a stochastic flow of orders. The solution to the problem resembles that of the Gambler's ruin problem. Garman studied several inventory-independent strategies that lead to either a *sure* failure or a possible failure. The conditions to avoid a sure failure imply a positive bid-ask spread. Garman concluded that a market-maker *must* relate its inventory to the price-setting strategy in order to avoid failure. Amihud & Mendelson (1980) extends Garman's model by studying the role of inventory. The problem is solved in a dynamic programming framework with inventory as the state variable. The optimal policy is a pair of bid and ask prices, both as decreasing functions of the inventory position. The model also implies that the spread is positive, and the market-maker has a preferred level of inventory. Ho & Stoll (1981) studies the optimal behavior of a single dealer who is faced with a stochastic demand and return risk of his own portfolio. As in Garman (1976), orders are represented by price-dependent stochastic processes. However, instead of maximizing expected profit, the dealer maximizes the ex-

pected utility of terminal wealth which depends on trading profit and returns to other components in its portfolio. Consequently dealer's risks play a significant role in its price-setting strategy. One important implication of this model is that the spread can be decomposed into two components: a risk neutral spread that maximizes the expected profits for a set of given demand functions and a risk premium that depends on the transaction size and return variance of the stock. Ho & Stoll (1983) is a multiple-dealer version of Ho & Stoll (1981). The price-dependent stochastic order flow mechanism is common in the above studies. All preceding studies only allow market orders traded in the market. O'Hara & Oldfiled (1986) attempts to incorporate more realistic features of real markets into its analysis. The paper studies a dynamic pricing policy of a risk-averse market-maker who receives both limit and market orders and faces uncertainty in the inventory valuation. The optimal pricing strategy takes into account the nature of the limit and market orders as well as inventory risk.

Inventory-based models focus on the role of order flow uncertainty and inventory risk in the determination of the bid-ask spread. The information-based approach suggests that the bid-ask spread could be a purely informational phenomenon irrespective of inventory risk. Glosten & Milgrom (1985) studies the market-making problem in a market with asymmetric information. In the Glosten-Milgrom model some traders have superior (insider) information and others do not. Traders consider their information and submit orders to the market sequentially. The specialist, which does not have any information advantage, sets his prices, conditioning on all his available information such that the expected profit on any trade is zero. Specifically, the specialist sets its prices equaled the conditional expectation of the stock value given past transactions. Its main finding is that in the presence of insiders, a positive bid-ask spread would exist even when the market-maker is risk-neutral and make zero expected profit.

Most of these studies have developed conditions for optimality but provided no explicit price adjustment policies. For example, in Amihud & Mendelson (1980), bid and ask prices are shown to relate to inventory but the exact dependence is unavailable. Some analyses do provide functional forms of the bid/ask prices (such as O'Hara & Oldfiled (1986)) but the practical applications of the results are limited due to stringent assumptions made in the models. The reinforcement learning models developed in this paper make few assumptions about the market environment and yield explicit price setting

strategies.

2.2 Reinforcement Learning

Reinforcement learning is a computational approach in which agents learn their strategies through trial-and-error in a dynamic interactive environment. It is different from supervised learning in which examples or learning targets are provided to the learner from an external supervisor.[1] In a typical reinforcement learning problems the learner is not told which actions to take. Rather, it has to find out which actions yield the highest reward through experience. More interestingly, actions taken by an agent affect not only the immediate reward to the agent but also the next state in the environment, and therefore subsequent rewards. In a nutshell, a reinforcement learner interacts with a its environment by adaptively choosing its actions in order to achieve some long-term objectives. Kaelbling & Moore (1996) and Sutton & Barto (1998) provide excellent surveys of reinforcement learning. Bertsekas & Tsitsiklis (1996) covers the subject in the context of dynamic programming.

Markov decision processes (MDPs) are the most common model for reinforcement learning. The MDP model of the environment consists of (1) a discrete set of *states* S, (2) a discrete set of *actions* the agent can take A, (3) a set of real-valued *rewards* R or reinforcement signals, (4) a starting probability distribution over S, (5) a transition probability distribution $p(s'|s,a)$, the probability of a state transition to s' from s when the agent takes action a, and (6) a reward probability distribution $p(r|s,a)$, the probability of issuing reward r from state s when the agent takes action a.

The MDP environment proceeds in discrete time steps. The state of the world for the first time step is drawn according to the starting probability distribution. Thereafter, the agent observes the current state of the environment and selects an action. That action and the current state of the world determine a probability distribution over the state of the world at the next time step (the transition probability distribution). Additionally, they determine a probability distribution over the reward issued to the agent (the reward probability distribution). The next state and a reward are chosen according to these

[1] Bishop (1995) gives a good introduction to supervised learning. See also Vapnik (1995),Vapnik (1998), and Evgeniou, M. & Poggio (2000).

distributions and the process repeats for the next time step.

The dynamics of the system are completely determined except for the action selection (or policy) of the agent. The goal of the agent is to find the policy that maximizes its long-term accumulated rewards, or *return*. The sequence of rewards after time step t is denoted as $r_t, r_{t+1}, r_{t+2}, \ldots$; the return at the time t, R_t, can be defined as a function of these rewards, for example,

$$R_t = r_t + r_{t+1} + \ldots + r_T;$$

or if rewards are to be discounted by a discount rate γ, $0 \leq \gamma \leq 1$:

$$R_t = r_t + \gamma r_{t+1} + \ldots + \gamma^{T-1} r_T,$$

where T is the final time step of a naturally related sequence of the agent-environment interaction, or an *episode*.[2]

Because the environment is Markovian with respect to the state (*i.e.* the probability of the next state conditioned on the current state and action is independent of the past), the optimal policy for the agent is deterministic and a function solely of the current state.[3] For reasons of exploration (explained later), it is useful to consider stochastic policies as well. Thus the policy is represented by $\pi(s,a)$, the probability of picking action a when the world is in state s.

Fixing the agent's policy converts the MDP into a Markov chain. The goal of the agent then becomes to maximize $E_\pi[R_t]$ with respect to π where E_π stands for the expectation over the Markov chain induced by policy π. This expectation can be broken up based on the state to aid in its maximization:

$$\begin{aligned} V^\pi(s) &= E_\pi[R_t | s_t = s], \\ Q^\pi(s,a) &= E_\pi[R_t | s_t = s, a_t = a], \end{aligned}$$

[2]These definitions and algorithms also extend to the non-episodic, or infinite-time, problems. However, for simplicity this paper will concentrate on the episodic case.

[3]For episodic tasks for which the stopping time is not fully determined by the state, the optimal policy may also need to depend on the time index. Nevertheless, this paper will consider only reactive policies or policies which only depend on the current state.

These quantities are known as value functions. The first is the expected return of following policy π out of state *s*. The second is the expected return of executing action *a* out of state *s* and thereafter following policy π.

There are two primary methods for estimating these value functions. The first is by Monte Carlo sampling. The agent executes policy π for one or more episodes and uses the resulting trajectories (the histories of states, actions, and rewards) to estimate the value function for π. The second is by temporal difference (TD) updates like SARSA (Sutton (1996)). TD algorithms make use of the fact that $V^{\pi}(s)$ is related to $V^{\pi}(s')$ by the transition probabilities between the two states (from which the agent can sample) and the expected rewards from state *s* (from which the agent can also sample). These algorithms use dynamic-programming-style updates to estimate the value function:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \,. \tag{1}$$

α is the learning rate that dictates how rapidly the information propagates.[4] Other popular TD methods include Q-learning (Watkins (1989), Watkins & Dayan (1992)) and TD(λ) (Watkins (1989), Jaakkola, Jordan & Singh (1994)). Sutton & Barto (1998) gives a more complete description of Monte Carlo and TD methods (and their relationship).

Once the value function for a policy is estimated, a new and improved policy can be generated by a policy improvement step. In this step a new policy π_{k+1} is constructed from the old policy π_k in a greedy fashion:

$$\pi_{k+1}(s) = \arg\max_a Q^{\pi_k}(s, a). \tag{2}$$

Due to the Markovian property of the environment, the new policy is guaranteed to be no worse than the old policy. In particular it is guaranteed to be no worse at every state individually: $Q^{\pi_{k+1}}(s, \pi_{k+1}(s)) \geq Q^{\pi_k}(s, \pi_k(a))$.[5] Additionally, the sequence of policies will converge to the optimal policy provided

[4] The smaller the α the slower the propagation, but the more accurate the values being propagated.

[5] See p. 95 Sutton & Barto (1998)

sufficient exploration (*i.e.* that the policies explore every action from every state infinitely often in the limit as the sequence grows arbitrarily long). To insure this, it is sufficient to not exactly follow the greedy policy of Equation 2 but instead choose a random action ε of the time and otherwise choose the greedy action. This ε-greedy policy takes the form

$$\pi_{k+1}(s,a) = \begin{cases} 1-\varepsilon & \text{if } a = \arg\max_{a'} Q^{\pi_k}(s,a'), \\ \frac{\varepsilon}{|A|-1} & \text{otherwise.} \end{cases} \tag{3}$$

An alternative to the greedy policy improvement algorithm is to use an actor-critic algorithm. In this method, the value functions are estimated using a TD update as before. However, instead of jumping immediately to the greedy policy, the algorithm adjusts the policy towards the greedy policy by some small step size. Usually (and in this paper), the policy is represented by a Boltzmann distribution:

$$\pi_t(s,a) = \Pr[a_t = a | s_t = s] = \frac{\exp(w_{(s,a)})}{\sum_{a' \in A} \exp(w_{s,a'})} \tag{4}$$

where $w_{(s,a)}$ is a weight parameter of π corresponding to action a in state s. The weights can be adjusted to produce any stochastic policy which can have some advantages (discussed in the next section).

All three approaches are considered in this paper: a Monte Carlo method, SARSA (a temporal difference method) and an actor-critic method. Each has certain advantages. The Monte Carlo technique can more easily deal with long delays between an action and its associated reward than SARSA. However, it does not make as efficient use of the MDP structure as SARSA does. Therefore, SARSA does better when rewards are presented immediately whereas Monte Carlo methods do better with long delays.

Actor-critic has its own advantage in that it can find explicitly stochastic policies. For MDPs this may not seem to be as much an advantage. However, for most practical applications, the world does not exactly fit the MDP model. In particular, the MDP model assumes that the agent can observe the true state of the environment. However in cases like market-marking that is not the case. While the agent can observe certain aspects (or statistics) of the world, other information (such as the information or beliefs

of the other traders) is hidden. If that hidden information can affect the state transition probabilities, the model then becomes a partially observable Markov decision process (POMDP). In POMDPs, the ideal policy can be stochastic (or alternatively depend on all prior observations which is prohibitively large in this case). Jaakkola, Singh & Jordan (1995) discusses the POMDP case in greater details.

While none of these three methods are guaranteed to converge to the ideal policy for a POMDP model (as they are for the MDP model), in practice they have been shown to work well even in the presence of hidden information. Which method is most applicable depends on the problem.

3 A Reinforcement Learning Model of Market-making

The market-making problem can be conveniently modeled in the framework of reinforcement learning. In the following market-making problems, an episode can be considered as a trading day. Note that the duration of an episode does not need to be fixed. An episode can last an arbitrary number of time steps and conclude when the certain task is accomplished. The market is a dynamic and interactive environment where investors submit their orders given the bid and ask prices (or quotes) from the market-maker. The market-maker in turn sets the quotes in response to the flow of orders. The job of the market-maker is to observe the order flow, the change of its portfolio, and its execution of orders and set quotes in order to maximize some long-term rewards that depend on the its objectives (*e.g.* profit maximization and inventory risk minimization).

3.1 Environment States

The environment state includes market variables that are used to characterize different scenarios in the market. These are variables that are observed by the market-maker from the order flow, its portfolio, the trades and quotes in the market, as well as other market variables:

- Inventory of the market-maker — amount of inventory-holding by the market-maker.
- Order imbalance — excess demand or supply in the market. This can be defined as the share difference between buy and sell market or limit orders received within a period of time.

- Market quality measures — size of the bid-ask spread, price continuity (the amount of transaction-to-transaction price change), depth of a market (the amount of price change given a number of shares being executed), time-to-fill of a limit order, etc.

- Others — Other characteristics of the order flow, information on the limit order book, origin of an order or identity of the trader, market indices, prices of stocks in the same industry group, price volatility, trading volume, time till market close, etc.

In this paper, we focus on three fundamental state variables: inventory, order imbalance and market quality. The state vector is defined as

$$\mathbf{s}_t = (INV_t,\ IMB_t,\ QLT_t)\ ,$$

where INV_t, IMB_t and QLT_t denote the inventory level, the order imbalance, and market quality measures respectively. The market-maker's inventory level is its current holding of the stock. A short position is represented by a negative value and a long position by a positive value. Order imbalance can be defined in many ways. One possibility is to define it as the sum of the buy order sizes minus the sum of the sell order sizes during a certain period of time. A negative value indicates an excess supply and a positive value indicates an excess demand in the market. The order imbalance measures the total order imbalance during a certain period of time, for example, during the last five minutes or from the last change of market-maker's quotes to the current time. Market qualities measure quantities including the bid-ask spread and price continuity (the amount of price change in a subsequent of trades). The values of INV_t, IMB_t and QLT_t are mapped into discrete values: $INV_t \in \{-M_{inv}, ..., -1, 0, 1, ..., M_{inv}\}$, $IMB_t \in \{-M_{imb}, ..., -1, 0, 1, ..., M_{imb}\}$, and $QLT_t \in \{-M_{QLT}, ..., 1, 0, 1, ..., M_{QLT}\}$. For example, a value of $-M_{inv}$ corresponds to the highest possible short position, -1 corresponds to the smallest short position and 0 represents an even position. Order imbalance and market quality measures are defined similarly.

3.2 Market-maker's actions

Given the states of the market, the market-maker reacts by adjusting the quotes, trading with incoming public orders, etc.. Permissible actions by the market-maker include the following:

- Change the bid price
- Change the ask price
- Set the bid size
- Set the ask size
- Others — Buy or sell, provide price improvement (provide better prices than the current market quotes).

The models in this paper focus on the determination of the bid and ask prices and assume fixed bid and ask sizes (*e.g.* one share). The action vector is defined as

$$\mathbf{a}_t = (\Delta BID_t, \Delta ASK_t) \ ,$$

where $\Delta BID_t = BID_t - BID_{t-1}$ and $\Delta ASK_t = ASK_t - ASK_{t-1}$, representing the change in bid and ask prices respectively. All values are discrete: $\Delta BID_t \in \{-M_{\Delta BID}, ..., 0, ..., M_{\Delta BID}\}$ and $\Delta ASK_t \in \{-M_{\Delta ASK}, ..., 0, ..., M_{\Delta ASK}\}$, where $M_{\Delta BID}$ and $M_{\Delta ASK}$ are the maximum allowable changes for the bid and ask prices respectively.

3.3 Reward

The reward signal is the agent's driving force to attain the optimal strategy. This signal is determined by the agent's objectives. Possible reward signals (and their corresponding objectives) include

- Change in profit (maximization of profit)
- Change in inventory level (minimization of inventory risk)

- Current market quality measures (maximization of market qualities)

The reward at each time step depends on the change of profit, the change of inventory, and the market quality measures at the current time step. The reward can be defined as some aggregate function of individual reward components. In its simplest form, assuming risk neutrality of the market-maker, the aggregate reward can be written as a linear combination of individual reward signals:

$$r_t = w_{pro}\Delta PRO_t + w_{inv}\Delta INV_t + w_{qlt}QLT_t \ , \tag{5}$$

where w_{pro}, w_{inv} and w_{qlt} are the parameters controlling the trade-off between profit, inventory risk and market quality; $\Delta PRO_t = PRO_t - PRO_{t-1}$, $\Delta INV_t = INV_t - INV_{t-1}$ and QLT_t are the change of profit, the change of inventory, and market quality measure respectively at time t. Note that the market-maker is interested in optimizing the *end-of-day* profit and inventory, but not the instantaneous profit and inventory. However, it is the market quality measures at each time step with which the market-maker is concerned in order to uphold the execution quality for all transactions. Recall that the agent intends to maximize the total amount of rewards it receives. The total reward for an episode with T time steps is

$$\begin{aligned} R_T &= \sum_{t=1}^{T} r_t \\ &= w_{pro}PRO_T + w_{inv}INV_T + w_{qlt}\sum_{t=1}^{T} QLT_t . \end{aligned}$$

Here the market-maker is assumed to start with zero profit and inventory: $PRO_0 = 0$ and $INV_0 = 0$.

The market-maker can observe the variables INV_t and QLT_t at each time t, but not necessarily PRO_t. In most cases, the "true" value or a fair price of a stock may not be known to the market-maker. Using the prices set by the market-maker to compute the reward could incorrectly value the stock. Furthermore the valuation could induce the market-maker to raise the price whenever it has a long position and lower the price whenever it has a short position, so that the value of its position can be maximized. Without a fair value of the stock, calculating the reward as in Equation 5 is not feasible. In these cases, some proxies of the fair price can be considered. For example, in a market

with multiple market-makers, other dealers' quotes and execution prices can reasonably reflect the fair value of the stock. Similarly, the fair price may also be reflected in the limit prices from the incoming limit orders. Lastly, the opening and closing prices can be used to estimate the fair price. This approach is motivated by how the market is opened and closed at the NYSE. The NYSE specialists do not open or close the market at prices solely based on their discretion. Instead, they act as auctioneers to set prices that balance demand and supply at these moments. Consequently these prices represent the most informative prices given all information available at that particular time.

In the context of the reinforcement learning algorithm, the total reward for an episode is calculated as the difference between the the end-of-day and the beginning-of-day profit:

$$R_T = PRO_T - PRO_0 = PRO_T.$$

Unfortunately, the profit reward at each time step is still unavailable. One remedy is to assume zero reward at each $t < T$ and distribute all total reward to at $t = T$. An alternative approach is to assign the episodic average reward $r_t = R_T/T$ to each time step.

For this paper two approaches in setting the reward are considered. In the first case, we assume that the reward can be calculated as a function of the true price at each time step. However, the true price is still *not* observable as a state variable. In the second case, we only reveal the true price at the end of a training episode at which point the total return can be calculated.

4 The Basic Model

Having developed a framework for the market-maker, the next step is to create a market environment in which the reinforcement learner can acquire experience. The goal here is to develop a simple model that adequately simulates the strategy of a trading crowd given the quotes of a market-maker. Information-based models focusing on information asymmetry provide the basis for our basic model. In a typical information-based model, there is a group of informed traders or insiders who have superior information about the true value of the stock and a group of uninformed traders who possess only public information.

The insiders buy whenever the market-maker's prices are too low and sell whenever they are too high given their private information; the uninformed simply trade randomly for liquidity needs. A single market-maker is at the center of trading in the market. It posts the bid and ask prices at which all trades transact. Due to the informational disadvantage, the market-maker always loses to the insiders while he breaks even with the uninformed.

4.1 Market Structure

To further illustrate this idea of asymmetric information among different traders, consider the following case. A single security is traded in the market. There are three types of participants: a monopolistic market-maker, insiders, and uninformed traders. The market-maker sets *one* price, p^m, at which the next arriving trader has the option to either buy or sell *one* share. In other words, it is assumed that the bid price equals the ask price. Traders trade only with market orders. All orders are executed by the market-maker and there are no crossings of orders among traders. After the execution of an order, the market-maker can adjust its quotes given its knowledge of past transactions. In particular it focuses on the order imbalance in the market in determining the new quotes. To further simplify the problem, it is assumed that the stock position is liquidated into cash immediately after a transaction. Hence inventory risk is not a concern for the market-maker. This is a continuous market in which the market-maker executes the orders the moment when they arrive.

For simplicity, events in the market occur at discrete time steps. In particular, events are modeled as independent Poisson processes. These events include the change of the security's true price and the arrival of informed and uninformed orders.

There exists a true price p^* for the security. The idea is that there is an exogenous process that completely determines the value of the stock. The true price is to be distinguished from the market price, which is determined by the interaction between the market-maker and the traders. The price p^* follows a Poisson jump process. In particular, it makes discrete jumps, upward or downward with a probability λ_p at each time step. The size of the discrete jump is a constant 1. The true price, p^*, is given to the insiders but not known to the public or the market-maker.

The insider and uninformed traders arrive at the market with a probability of λ_i and $2\lambda_u$ respectively.[6] Insiders are the only ones who observe the true price of the security. They can be considered as investors who acquire superior information through research and analysis. They compare the true price with market-maker's price and will buy (sell) one share if the true price is lower (higher) than the market-maker's price, and will submit no orders otherwise. Uninformed traders will place orders to buy and sell a security randomly. The uninformed merely re-adjust their portfolios to meet liquidity needs, which is not modeled in the market. Hence they simply submit buy or sell orders of one share randomly with equal probabilities λ_u.

All independent Poisson processes are combined together to form a new Poisson process. Furthermore, it is assumed that there is one arrival of an event at each time step. Hence, at any particular time step, the probability of a change in the true price is $2\lambda_p$, that of an arrival of an insider is λ_i, and that of an arrival of an uninformed trader is $2\lambda_u$. Since there is a guaranteed arrival of an event, all probabilities sum up to one: $2\lambda_p + 2\lambda_u + \lambda_i = 1$.

This market model resembles the information-based model, such as Glosten & Milgrom (1985), in which information asymmetry plays a major role in the interaction between the market-maker and the traders. The Glosten and Milgrom model studies a market-maker that sets bid and ask prices to earn zero expected profit given available information, while this model examines the quote-adjusting strategies of a market-maker that maximize sample average profit over multiple episodes, given order imbalance information. This model also shares similarities with the work of Garman (1976) and Amihud & Mendelson (1980) where traders submit price-dependent orders and the market-making problem is modeled as discrete Markov processes. But instead of inventory, here the order imbalance is used to characterize the state.

4.2 Strategies and Expected Profit

For this basic model, it is possible to compute the ideal strategies. We do this first, before presenting the reinforcement learning results for the basic model.

[6] Buy and sell orders from the uninformed traders arrive at a probability of λ_u respectively.

Closed-form characterization of an optimal market-making strategy in such a stochastic environment can be difficult. However, if one restricts one's attention to order imbalance in the market, it is obvious that any optimum strategy for a market-maker must involve the raising (lowering) of price when facing positive (negative) order imbalance, or excess demand (supply) in the market. Due to the insiders, the order imbalance on average would be positive if the market-maker's quoted price is lower than the true price, zero if both are equal, and negative if the quoted price is higher than the true price.

We now must define order imbalance. We will define it as the total excess demand since the last change of quote by the market-maker. Suppose there are x buy orders and y sell orders of one share at the current quoted price; the order imbalance is $x-y$. One viable strategy is to raise or lower the quoted price by 1 whenever the order imbalance becomes positive or negative. Let us denote this as Strategy 1. Note that under Strategy 1, order imbalance can be -1, 0 and 1. To study the performance of Strategy 1, one can model the problem as a discrete Markov process.[7] First we denote $\Delta p = p^m - p^*$ as the deviation of market-maker's price from the true price, and IMB as the order imbalance. A Markov chain describing the problem is shown in Figure 1. Suppose $\Delta p = 0$, p^* may jump to p^*+1 or p^*-1 with a probability of λ_p (due to the true price process); at the same time, p may be adjusted to $p+1$ or $p-1$ with a probability λ_u (due to the arrival of uninformed traders and the market-maker's policy). Whenever $p \neq p^*$ or $\Delta p \neq 0$, p will move toward p^* at a faster rate than it will move away from p^*. In particular, p always moves toward p^* at a rate of $\lambda_u + \lambda_i$, and moves away from p^* at a rate of λ_u. The restoring force of the market-maker's price to the true price is introduced by the informed trader, who observes the true price. In fact, it is the presence of the informed trader that ensures the existence of the steady-state equilibrium of the Markov chain.

Let q_k be the steady-state probability that the Markov chain is in the state where $\Delta p = k$. By symmetry of the problem, we observe that

$$q_k = q_{-k}, \quad \text{for } k = 1, 2, \ldots \tag{6}$$

Focus on all $k > 0$ and consider the transition between the states $\Delta p = k$ and $\Delta p = k+1$. One can relate

[7] Lutostanski (1982) studies a similar problem.

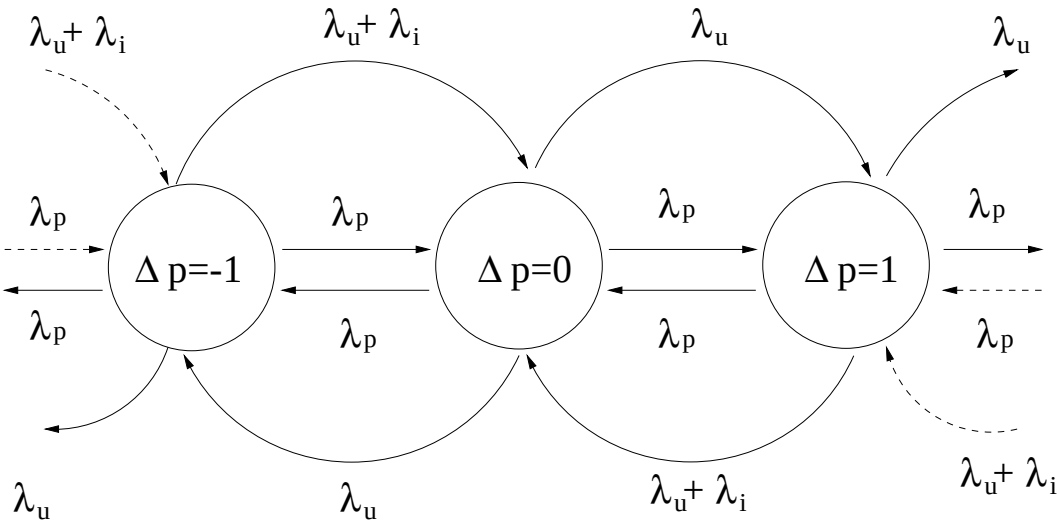

Figure 1: The Markov chain describing Strategy 1, imbalance threshold $M_{imb} = 1$, in the basic model.

the steady-state probabilities as

$$\begin{aligned} q_{k+1}(\lambda_p + \lambda_u + \lambda_i) &= q_k(\lambda_p + \lambda_u) \\ q_{k+1} &= q_k \frac{(\lambda_p + \lambda_u)}{(\lambda_p + \lambda_u + \lambda_i)} \quad \text{for } k = 0, 1, 2, ... \end{aligned} \tag{7}$$

because a transition from $\Delta p = k$ to $\Delta p = k+1$ is equally likely as a transition from $\Delta p = k+1$ to $\Delta p = k$ at the steady state. By expanding from Equation 8 and considering Equation 6, the steady-state probability q_k can be written as

$$q_k = q_0 \left(\frac{\lambda_p + \lambda_u}{\lambda_p + \lambda_u + \lambda_i} \right)^{|k|}, \forall\, k \neq 0.$$

All steady-state probabilities sum up to one

$$\begin{aligned} \sum_{k=-\infty}^{\infty} q_k &= 1, \\ q_0 + 2\sum_{i=1}^{\infty} q_k &= 1, \\ q_0 &= \frac{\lambda_i}{2\lambda_p + 2\lambda_u + \lambda_i}\ . \end{aligned} \tag{8}$$

With the steady-state probabilities, one can calculate the expected profit of the strategy. Note that at the state $\Delta p = k$, the expected profit is $-\lambda_i|k|$ due to the informed traders. Hence, the expected profit can be written as

$$\begin{aligned} EP &= \sum_{k=-\infty}^{\infty} -q_k\lambda_i|k| \\ &= -2\sum_{k=1}^{\infty} q_k\lambda_i|k| \\ &= -2q_0\lambda_i\sum_{k=1}^{\infty} k\left(\frac{\lambda_p+\lambda_u}{\lambda_p+\lambda_u+\lambda_i}\right)^k \\ &= \frac{-2(\lambda_p+\lambda_u)(\lambda_p+\lambda_u+\lambda_i)}{(2\lambda_p+2\lambda_u+\lambda_i)} \quad . \end{aligned} \tag{9}$$

The expected profit measures the average profit accrued by the market-maker per unit time. The expected profit is negative because the market-maker breaks even in all uninformed trades while it always loses in informed trades.

By simple differentiation of the expected profit, we find that EP goes down with λ_p, the rate of price jumps, holding λ_u and λ_i constant. The expected profit also decreases with λ_i and λ_u respectively, holding the other λ's constant. However, it is important to point out that $2\lambda_p+2\lambda_u+\lambda_i = 1$ since there is a guaranteed arrival of a price jump, an informed or uninformed trade at each time period. Hence changing the value of one λ while holding others constant is impossible. Let us express λ_p and λ_u in terms of λ_i: $\lambda_p = \alpha_p\lambda_i$ and $\lambda_u = \alpha_u\lambda_i$. Now the expected profit can be written as:

$$EP = \frac{-2(\alpha_p+\alpha_u)(\alpha_p+\alpha_u+1)}{(2\alpha_p+2\alpha_u+1)^2} \quad .$$

Differentiating the expression gives

$$\frac{\partial EP}{\partial \alpha_p} = \frac{\partial EP}{\partial \alpha_u} = \frac{-2}{(2\alpha_p+2\alpha_u+1)^3} < 0.$$

The expected profit increases with the relative arrival rates of price jumps and uninformed trades.

To compensate for the losses, the market-maker can charge a fee for each transaction. This would

relate the expected profit to the bid-ask spread of the market-maker. It is important to notice that the strategy of the informed would be different if a fee of x unit is charged. In particular, if a fee of x units is charged, the informed will buy only if the $p^* - p^m > x$ and sell only if $p^m - p^* > x$. If the market-maker charges the same fee for buy and sell orders, the sum of the fees is the spread. Let us denote the fee as a half of the spread, $SP/2$. The market-maker will gain $SP/2$ on each uninformed trade, and $|\Delta p| - SP/2$ (given that $|\Delta p| - SP/2 > 0$) on each informed trade. If the spread is constrained to be less than 2, then the informed traders' strategy does not change, and we can use the same Markov chain as before. Given SP and invoking symmetry, the expected profit can be written as

$$EP = \lambda_u SP - 2\lambda_i \sum_{k \geq SP/2}^{\infty} (k - SP/2) q_k.$$

If the market-maker is restricted to making zero profit, one can solve the previous Equation for the corresponding spread. Specifically, if $(1-\lambda_i)(1-2\lambda_i) < 4\lambda_u$, the zero expected profit spread is

$$SP_{EP=0} = \frac{1-\lambda_i}{2\lambda_u + \lambda_i(1-\lambda_i)} < 2. \tag{10}$$

Although inventory plays no role in the market-making strategy, the symmetry of the problem implies a zero expected inventory position for the market-maker.

Strategy 1 reacts to the market whenever there is an order imbalance. Obviously this strategy may be too sensitive to the uninformed trades, which are considered noise in the market, and therefore would not perform well in high noise markets. This motivates the study of alternative strategies. Instead of adjusting the price when $IMB = 1$ or $IMB = -1$, the market-maker can wait until the absolute value of imbalance reaches a threshold M_{imb}. In particular, the market-maker raises the price by 1 unit when $IMB = M_{imb}$, or lowers the price by 1 unit when $IMB = -M_{imb}$ and resets $IMB = 0$ after that. The threshold equals 1 for Strategy 1. All these strategies can be studied in the same framework of Markov models. Figure 2 depicts the Markov chain that represents strategies with $M_{imb} = 2$. Each state is now specified by two state variables Δp and IMB. For example, at the state $(\Delta p = 1, IMB = -1)$, a sell order (a probability of $\lambda_u + \lambda_i$) would move the system to $(\Delta p = 0,\ IMB = 0)$; a buy order (a probability of

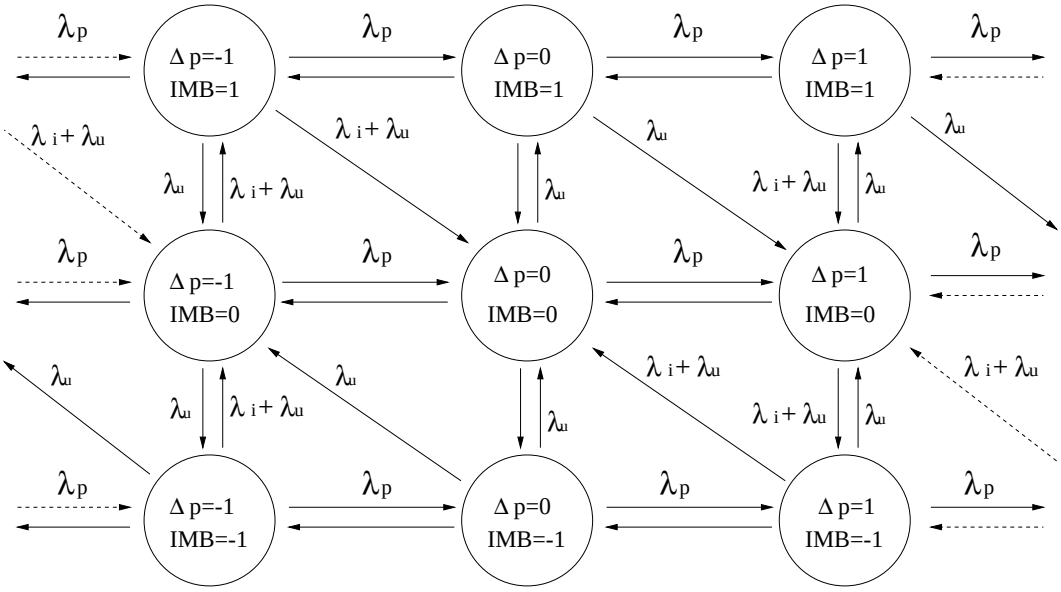

Figure 2: The Markov chain describing Strategy 2, with the imbalance threshold $M_{imb} = 2$ in the basic model.

λ_u) would move the system to $(\Delta p = 1,\ IMB = 0)$; a price jump (a probability of λ_u) would move the system to either $(\Delta p = 0,\ IMB = -1)$ or $(\Delta p = 2,\ IMB = -1)$.

Intuitively, strategies with higher M_{imb} would perform better in noisier (larger λ_u) markets. Let us introduce two additional strategies: strategies with $M_{imb} = 2$ and $M_{imb} = 3$ and denote them as Strategies 2 and 3 respectively. The expected profit provides a criterion to choose among the strategies. Unfortunately analytical characterization of the expected profit for Strategies 2 and 3 is mathematically challenging. Instead of seeking explicit solutions in these cases, Monte Carlo simulations are used to compute the expected profits for these cases. To compare among the strategies, we set α_p to a constant and vary α_u and obtain the results in Figure 3. The expected profit for Strategy 1 decreases with the noise level whereas the expected profit for Strategies 2 and 3 increases with the noise level. Among the three strategies, we observe that Strategy 1 has the highest EP for $\alpha_u < 0.3$, Strategy 2 has the highest EP for $0.3 < \alpha_u < 1.1$ and Strategy 3 has the highest EP for $\alpha_u > 1.1$.

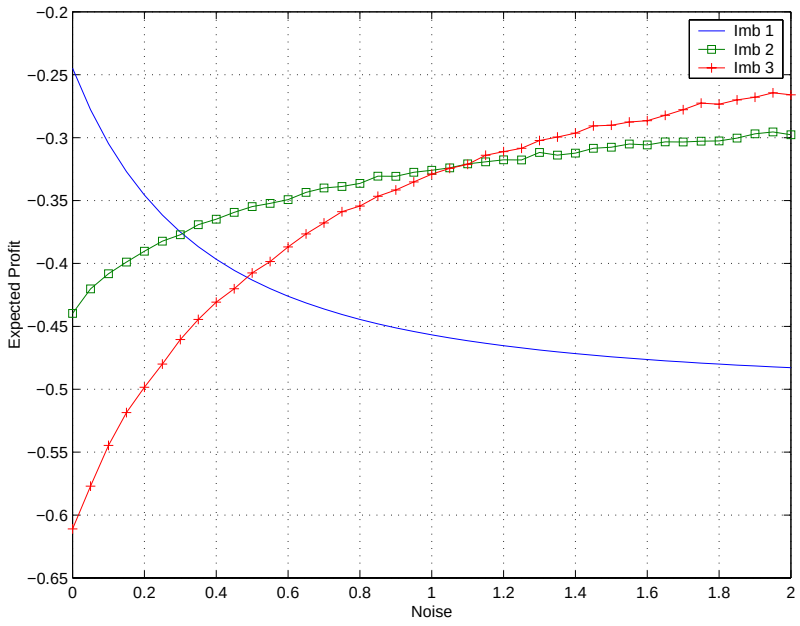

Figure 3: Expected profit for Strategies 1, 2, and 3 in the basic model.

		ΔP		
		-1	0	1
	-3	-0.86	-0.86	-0.86
	-2	-1.13	-1.41	-1.62
	-1	**-1.16**	-1.70	-2.61
IMB	0	-1.62	**-0.74**	-1.62
	1	-2.63	-1.78	**-1.13**
	2	-1.67	-1.43	-1.15
	3	-0.91	-0.91	-0.91

(a) Strategy 1

		ΔP		
		-1	0	1
	-3	-1.95	-1.96	-1.96
	-2	**-1.97**	-2.41	-3.23
	-1	-2.18	**-2.03**	-2.72
IMB	0	-2.33	**-1.79**	-2.35
	1	-2.76	**-2.05**	-2.21
	2	-3.20	-2.37	**-2.01**
	3	-1.90	-1.89	-1.89

(b) Strategy 2

		ΔP		
		-1	0	1
	-3	**-10.10**	-19.13	-19.09
	-2	-18.09	**-10.67**	-18.50
	-1	-12.18	**-10.41**	-12.82
IMB	0	-11.13	**-9.92**	-11.56
	1	-11.99	**-9.76**	-12.11
	2	-18.57	**-10.32**	-18.14
	3	-19.17	-19.22	**-10.54**

(c) Strategy 3

Figure 4: Examples of Q-functions for Strategies 1, 2 and 3. The bold values are the maximums for each row showing the resulting greedy policy.

4.3 Market-making with Reinforcement learning Algorithms

Our goal is to model an optimal market-making strategy in the reinforcement learning framework presented in Section 3. In this particular problem, the main focus is on whether reinforcement learning algorithms can choose the optimum strategy in terms of expected profit given the amount of noise in the market, α_u. Noise is introduced to the market by the uninformed traders who arrive at the market with a probability $\lambda_u = \alpha_u \lambda_i$.

For the basic model, we use the Monte Carlo and SARSA algorithms. Both build a value function $Q^\pi(s,a)$ and employ an ε-greedy policy with respect to this value function. When the algorithm reaches equilibrium, π is the ε-greedy policy of its own Q-function. The order imbalance $IMB \in \{-3,-2,\ldots,2,3\}$ is the only state variable. Since market-maker quotes only one price, the set of actions is represented by $\Delta p^m \in \{-1,0,1\}$. Although the learning algorithms have the ability to represent many different policies (essentially any mapping from imbalance to price changes), in practice they converge to one of the three strategies as described in the previous section. Figure 4 shows three typical Q-functions and their implied policies after SARSA has found an equilibrium. Take Strategy 2 as an example, it adjusts price only when *IMB* reaches 2 or -2:

Yet, this seemingly simple problem has two important complications from a reinforcement learning point-of-view. First the environment state is only partially observable. The agent observes the order imbalance but not the true price or the price discrepancy Δp. This leads to the violation of the Markov property. The whole history of observed imbalance now becomes relevant in the agent's decision making. For instance, it is more likely that the quoted price is too low when observing positive imbalance in two consecutive time steps than in just one time step. Formally, $\Pr[\Delta p | IMB_t, IMB_{t-1}, \ldots, IMB_0] \neq \Pr[\Delta p | IMB_t]$. Nevertheless the order imbalance, a noisy signal of the true price, provides information about the hidden state variable Δp. Our model simply treats *IMB* as the state of the environment. However, convergence of deterministic temporal difference methods are not guaranteed for non-Markovian problems. Oscillation from one policy to another may occur. Deterministic policies such as those produced by the Monte Carlo method and SARSA may still yield reasonable results. Stochastic policies, which will be studied in the extended model, may offer some improvement in partially observable

environments.

Second, since the true price is unobservable, it is infeasible to give a reward to the market-maker at each time step. As mentioned in Section 3.3, two possible remedies are considered. In the first approach, it is assumed that the true price is available for the calculation of the reward, but not as a state variable. Recall that the market-maker's inventory is liquidated at each step. The reward at time t is therefore the change of profit for the time step

$$r_t = \Delta PRO_t = \begin{cases} p_t^* - p_t^m & \text{for a buy order} \\ p_t^m - p_t^* & \text{for a sell order} \end{cases} \tag{11}$$

Alternatively, no reward is available during the episode, but only one final reward is given to the agent at the end of the episode. In this case, we choose to apply the Monte Carlo method and assign the end-of-episode profit per unit time, PRO_T/T, to *all* actions during the episode. Specifically, the reward can be written as

$$r_t = \frac{1}{T}\sum_{\tau=1}^{T} \Delta PRO_t. \tag{12}$$

Table 1 shows the options used for each of the experiments in this paper. The first two experiments are conducted using the basic model of this section, whereas the rest are conducted using the extended model of the next section that incorporates a bid-ask spread. Each experiment consists of 15 (10 for the extended model) separate sub-experiments, one for each of 15 (10) different noise levels. Each sub-experiment was repeated for 1000 different learning sessions. Each learning session ran for 2000 (1000 for the extended model) episodes each of 250 time steps.

4.4 Simulation Results

In the experiments, the primary focus is whether the market-making algorithm converges to the optimum strategy that maximizes the expected profit. In addition, the performance of the agent is studied in terms of profit and inventory at the end of an episode, PRO_T and INV_T, and average absolute price

Experiment Number	Model	Learning Method	State(s) s_t	Actions a_t	Reward r_t
1	basic	SARSA	IMB_t	$\Delta P \in A$	ΔPRO_t
2	basic	Monte Carlo	IMB_t	$\Delta P \in A$	PRO_T/T
3	extended	actor-critic	(IMB_t, QLT_t)	$\Delta BID_t \in A$ $\Delta ASK_t \in A$	$w_{pro}\Delta PRO_t + w_{qlt}QLT_t$
3a	extended	SARSA	(IMB_t, QLT_t)	$\Delta BID_t \in A$ $\Delta ASK_t \in A$	$w_{pro}\Delta PRO_t + w_{qlt}QLT_t$
4	extended	actor-critic	(IMB_t, QLT_t)	$\Delta BID_t \in A$ $\Delta ASK_t \in A$	$-\lvert\Delta PRO_t\rvert$

Table 1: Details of the experiments for the basic and extended models.

deviation for the entire episode, $\overline{\Delta_p} = \frac{1}{T}\sum_{t=1}^{T}|p_t^m - p_t^*|$. The agent's end-of-period profit is expected to improve with each training episode, though remain negative. Its inventory should be close to zero. The average absolute price deviation measures how closely the agent estimates the true price. Figure 5 shows a typical realization of Experiment 1 in episodes 25, 100, 200 and 500. One can observe that the market-maker's price tracks the true price more closely as time progresses. Figures 6a and 6b show the realized end-of-period profit and inventory of the market-maker and their corresponding theoretical values. The profit, inventory and price deviation results all indicate that the algorithm converges at approximately episodes 500.

With the knowledge of the instantaneous reward as a function of the true price, the SARSA method successfully determines the best strategy under moderate noise level in the market. Figure 7 shows the overall results from Experiment 1. The algorithm converges to Strategy 1, 2, or 3, depending on the noise level. For each value of α_u, the percentages of the sub-experiments converging to strategies 1, 2 and 3 are calculated. One important observation is that the algorithm does not always converge to the same strategy, especially under high noise circumstances and around points of policies transitions. The agent's policy depends on its estimates of the Q-values, which are the expected returns of an action given a state. Noisier observations result in estimates with higher variability, which in turn transforms

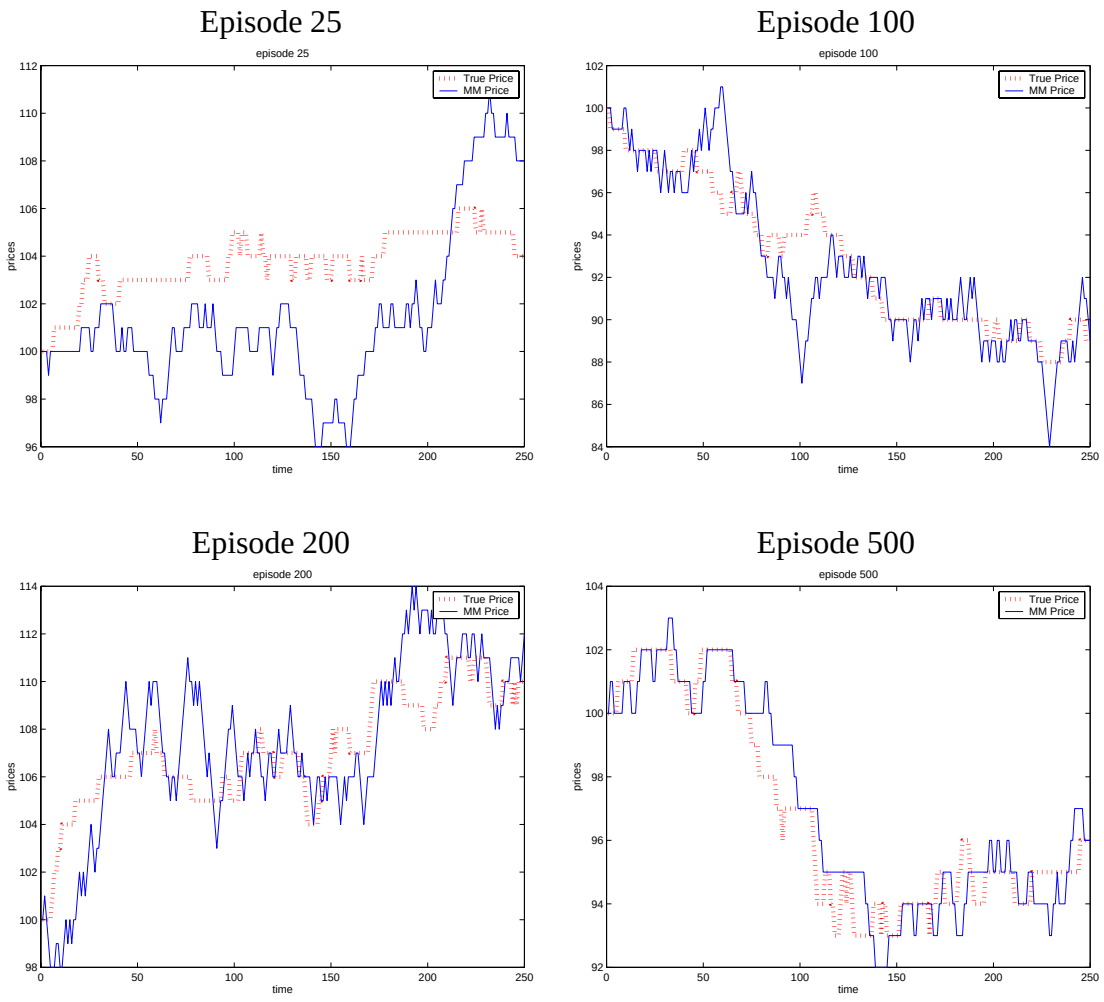

Figure 5: Episodes 25, 100, 200 and 500 in a typical realization of Experiment 1. The market-maker's price is shown in the solid line while the true price in dotted line. The maker's price traces the true price more closely over time.

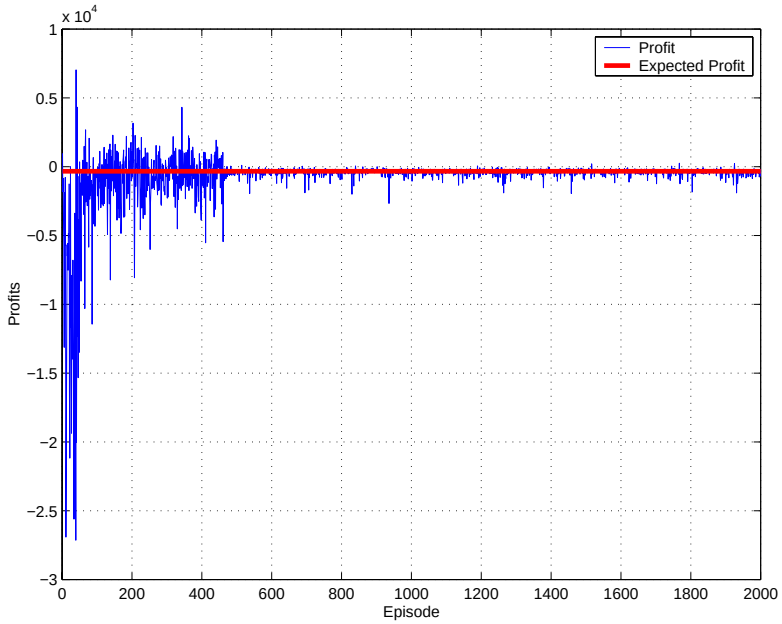

Figure 6a: End-of-episode profit and the corresponding theoretical value of the market-maker in Experiment 1 for a typical run with $\lambda_u = 0.25\lambda_i$. The algorithm converges around episode 500 when realized profit goes to its theoretical value.

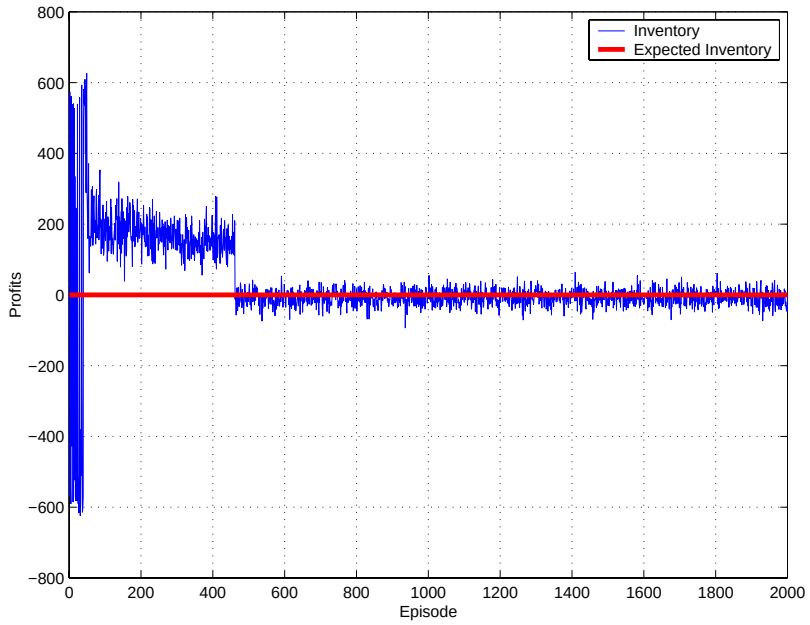

Figure 6b: End-of-episode Inventory and the corresponding theoretical value of the market-maker in Experiment 1 for a typical run with $\lambda_u = 0.25\lambda_i$. The algorithm converges around episode 500 when realized inventory goes to zero.

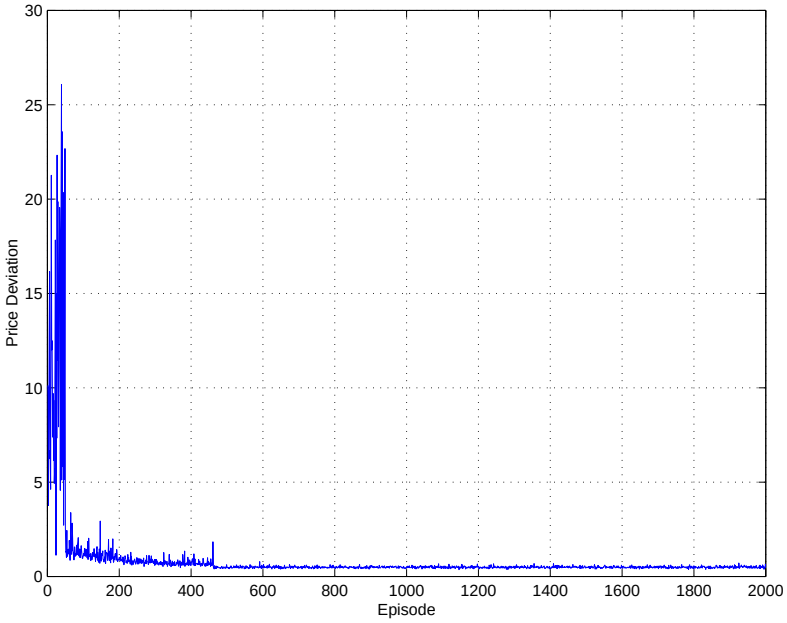

Figure 6c: Average absolute price deviation of the market-maker's quotation price from the true price in Experiment 1 for a typical run with $\lambda_u = 0.25\lambda_i$. The algorithm converges around episode 500 when the price deviation settles to its minimum.

into the variability in the choice of the optimum policy. Noise naturally arising in fully observable environments is handled well by SARSA and Monte Carlo algorithms. However, the mismatch between fully observable modeling assumption and the partially observable world can cause variability in the estimates which the algorithms do not handle as well. This is responsible for the problems seen at the transition points.

The results show that the reinforcement learning algorithm is more likely to converge to Strategy 1 for small values of α ($\alpha < 0.25$) and Strategy 2 for higher values of α ($0.35 < \alpha < 1.00$). There are abrupt and significant points of change at $\alpha \simeq 0.30$ and $\alpha \simeq 1.00$ where the algorithm switches from one strategy to another. These findings are consistent with the theoretical predictions based on the comparison of the expected profits for the strategies (Figure 3). When the noise level α exceeds the level of 1.0, the algorithm converges to Strategies 2 and 3 with an approximate likelihood of 80 and 20 percent respectively. According to the theoretical prediction, Strategy 3 would dominate the other two strategies when $\alpha_u > 1.1$. Unfortunately, the simulation fails to demonstrate this change of strategy. This is partially due to the inaccuracy in estimating the Q-function with the increasing amount of noise

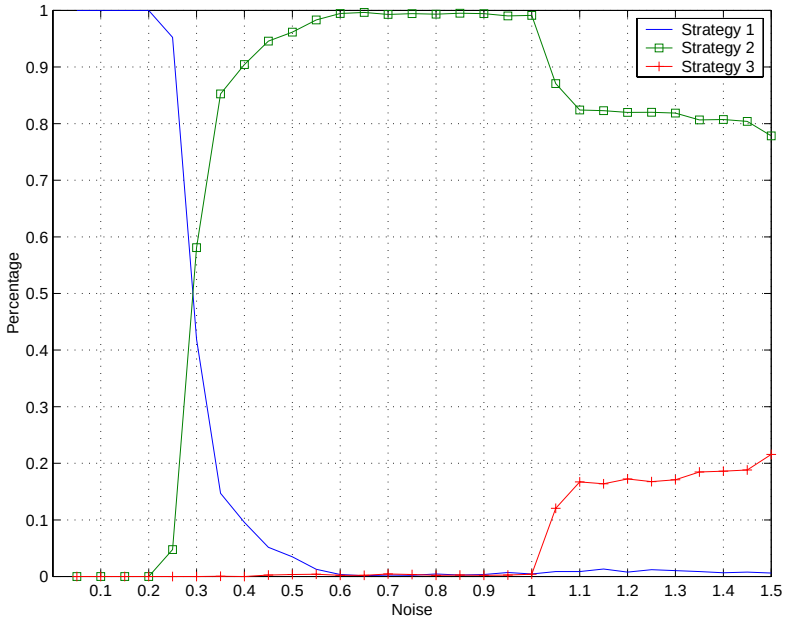

Figure 7: Percentages of SARSA simulations converges to Strategies 1, 2 and 3 in Experiment 1.

in the market. Furthermore, the convergence to Strategy 3 is intrinsically more difficult than that to Strategies 1 and 2. In order to recommend Strategy 3, the algorithm has to first decide to maintain the price for $IMB \leq 2$, effectively rejecting Strategies 1 and 2, and then estimate the relevant Q-values for $IMB = 3$; more exploration of the state space is necessary to evaluate Strategy 3.

What if no reward is given to the agent during the course of an episode? Experiment 2 is the same as Experiment 1 except for the differences in the learning method and the way reward is calculated. Even without the knowledge of the precise reward at each time step, the Monte Carlo algorithm still manages to shed some light on the choice of the optimum strategy. Figure 8 presents the percentages of the strategies chosen for different noise levels. The algorithm is more likely to choose Strategy 1 for small values of α_u ($\alpha_u < 0.30$), Strategy 2 for moderate values of α_u ($0.30 < \alpha_u < 1$), and Strategy 3 for large values of α_u ($\alpha_u > 1$). This finding to some extent agrees with what the theory predicts.

Information on how much each action contributes to the total return is missing, unlike in the case of the SARSA method where the value of an action is more immediately realized. This is known as the *credit assignment problem*, first discussed by Minsky (1963). Even without the knowledge of the contribution of individual actions, the Monte Carlo method still works. This is because, on average,

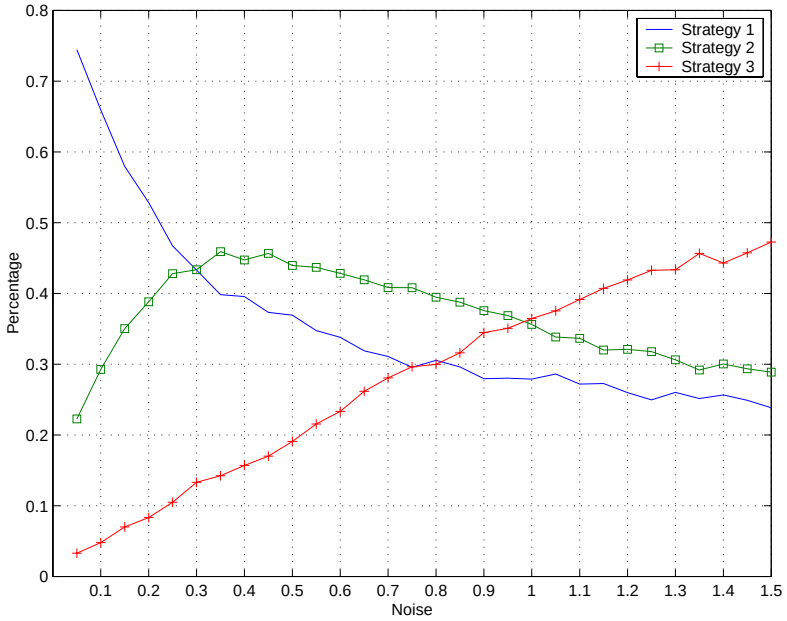

Figure 8: Percentages of the simulations of the Monte-Carlo method converges to Strategies 1, 2 and 3 in Experiment 2.

"correct actions" yield more reward and episodes with more "correct actions" consequently gather higher total return. But the missing reward information on individual action results in a higher variance in the estimation of values functions.

5 The Extended Model

The previous section demonstrates how reinforcement learning algorithms can be applied to market-making problems and successfully converge to optimum strategies under different circumstances. Although the basic model is useful because the experimental and theoretical results can be compared, one major limitation of the basic model is the equality of the bid and ask prices. Without the bid-ask spread the market-maker suffers a loss from the market due to the information disadvantage. A natural extension of the basic model is to let the market-maker quote bid and ask prices. This section studies a reinforcement learning strategy of the market-maker that balances the conflicting objectives of maximizing profit and market quality. Computer experiments demonstrate that the market-making agent successfully tracks the true price using the its bid and ask prices, and controls its average spread in a

continuous scale.

To incorporate bid and ask prices to the model, the set of actions is augmented to include the change of bid price and the change ask price:

$$(\Delta BID_t, \Delta ASK_t) \in A \times A,$$

where $A = \{-1, 0, 1\}$. Altogether there are nine possible actions. Now, to characterize a market scenario, the set of states should also include the spread, a measure of market quality. Specifically, the state vector becomes

$$\mathbf{s}_t = (IMB_t, SP_t),$$

where $IMB_t \in \{-1, 0, 1\}$ is the order imbalance and $SP_t = ASK_t - BID_t \in \{1, 2, 3, 4\}$ is the spread at time t. The spread also enters the reward function for the control of market quality maintained by a market-making algorithm. Recall that a market-maker may have multiple objectives. In the basic model, the market-maker only aims at maximizing profit. With spread added to the model, the market-maker would also need to consider the quality of market it provides. To balance between the two objectives, consider the following reward function that linearly combines the measures of profit and spread:

$$r_t = w_{pro}(\Delta PRO_t) + w_{qlt} SP_t,$$

where the reward for profit now depends on the side of the order:

$$\Delta PRO_t = \begin{cases} ASK_t - p_t^* & \text{for a buy order} \\ p_t^* - BID_t & \text{for a sell order} \end{cases} \tag{13}$$

As for the reinforcement learning technique, an actor-critic method as described in Section 2.2 is used for the extended model. This algorithm allows the agent to expressly pick stochastic policies, which is important for two reasons. First, stochastic policies allow real-valued average spreads and profits. Essentially, this gives the agent more control over the fine-tuning of the trade-off between profit

and market quality. For example, a policy which maintains a spread of 1 and 2 with equal probability of $1/2$ would lead to an average spread of 1.5. Since the spread is intimately related to the profit (as shown as Section 4.2), the agent also indirectly controls the profit. Second, stochastic policies are particularly efficient in problems with partially observable states. This extended model pushes the partial observability of the environment much further.

The market-making agent should aim to set its bid and ask prices such that they enclose the observed true price: $BID_t \leq p_t^* \leq ASK_t$. Under this condition, the market-maker will gain from any trades (those of the uninformed traders) submitted to the market.

Three computer experiments are conducted for the extended model. In Experiments 3 and 3a, the market-maker simultaneously maximizes profit and market-quality. The weight w_{pro} is fixed but w_{qlt} is varied to demonstrate how spread can be fine-tuned. Experiment 3 applies the actor-critic method that yields stochastic policies; Experiment 3a considers the SARSA method that yields deterministic policies. It is interesting to compare the performance of the two approaches under partially observable environments.

Experiment 4 studies how one can directly control the profit by incorporating a target profit ΔPRO^* into the reward function:

$$r'_t = |\Delta PRO_t - \Delta PRO^*|.$$

The target profit ΔPRO^* is the desired average profit per unit time. Experiment 4 studies the particular case when $\Delta PRO^* = 0$. The resulting spread is the zero profit spread for the market-maker.

5.1 Simulation Results

As in the basic model, the performance of the market-maker is measured with variables including profit and average absolute price deviation. The end-of-episode profit PRO_T measures how much the market-maker makes in an episode:

$$PRO_T = \sum_{t=1}^{T} \Delta PRO_t,$$

where ΔPRO_t is defined in Equation 13. The average absolute price deviation for an episode is calculated by considering both bid and ask prices:

$$\overline{\Delta_p} = \sum_{t=1}^{T} |BID_t - p_t^*| + |ASK_t - p_t^*| \ .$$

The episodic average spread for an episode is calculated as the average of the spread over time.

$$\overline{SP} = \frac{1}{T} \sum_{t=1}^{T} SP_t .$$

Figure 9 presents a typical run of Experiment 4. The accuracy in tracking the true price improves over the episodes. Figures 10a to 10d show the end-of-episode profit and inventory, average spread, and average absolute price deviation for a run of Experiment 3. The figures indicate that the algorithm converges approximately at episode 500.

To demonstrate the results of an actor-critic method, Figure 11 graphically depicts the details of a typical stochastic policy found in Experiment 3. The figure shows the probability distribution of actions in all twelve possible situations specified by the state vector (IMB, SP). For each situation, the probabilities of the nine possible actions are shown as a grid of squares. The areas of the squares represent the probabilities of pairs of bid/ask actions under the policy. The bid/ask actions have been transformed into changes of the mid-quote, $(\Delta ASK + \Delta BID)/2$, and the changes of the spread, $\Delta ASK - \Delta BID$, for easier interpretation of the figure.

The policy adjusts the prices for two objectives: to react to the order imbalance and to control the spread. It behaves correctly by reducing, maintaining, and raising bid/ask prices under negative, zero, and positive imbalance respectively, for cases of $SP = 1, 2, 3$. For the case when $SP = 4$, order imbalance is ignored (*i.e.* the adjustment of the mid-quote is not biased towards any direction). On the other hand, the policy tends to increase the spread for $SP = 1$, maintain or slightly increase the spread for $SP = 2$, and decrease the spread for $SP = 3,\ 4$. The mean and median spread resulting from this policy are both approximately 2.7.

By varying the spread parameter w_{qlt}, we can control the spread of the policy learned by either

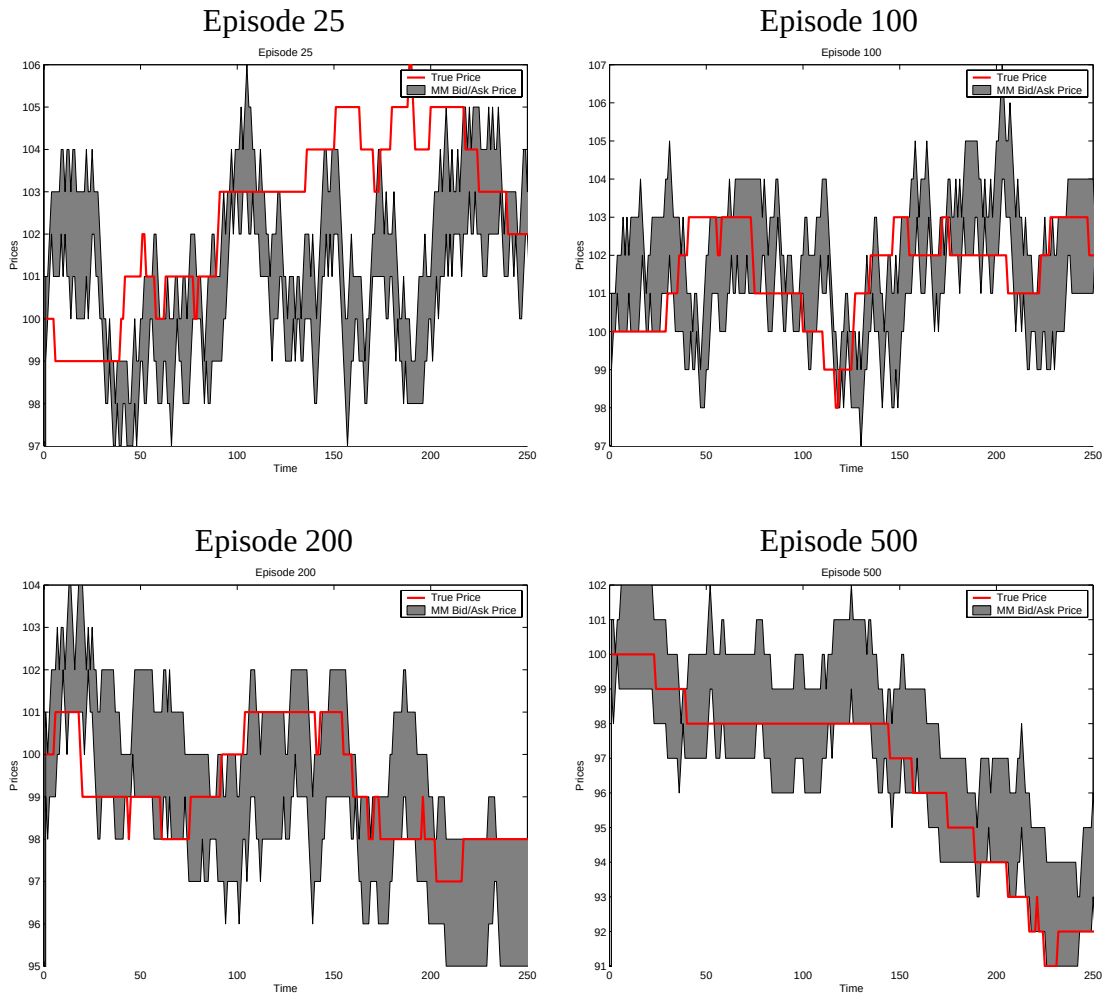

Figure 9: Episodes 25, 100, 200 and 500 in a typical realization of Experiment 3 with $w_{qlt} = 0.1$. The bid and ask prices are shown in the shaded area, and the true price in the single solid line. The algorithm shows improvement in tracing the true price with the bid and ask prices over time.

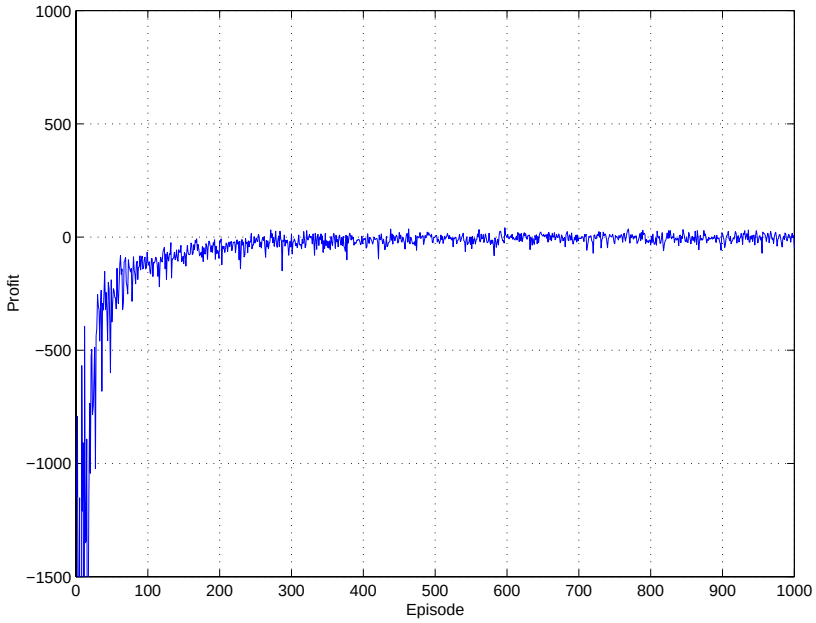

Figure 10a: End-of-episode profit, PRO_T, of a typical epoch of Experiment 3 with $w_{qlt} = 0.1$.

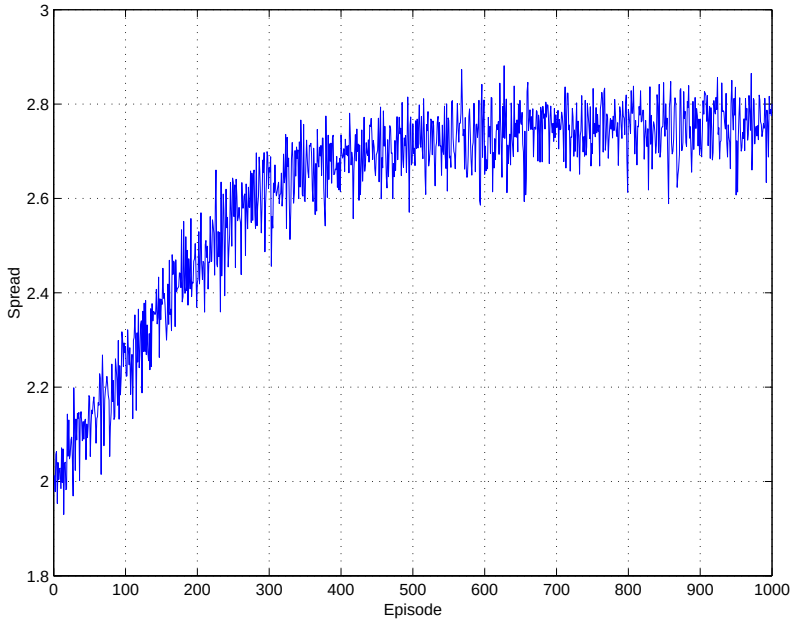

Figure 10b: Episodic average spread, $\overline{SP}$, of a typical epoch of Experiment 3 with $w_{qlt} = 0.1$.

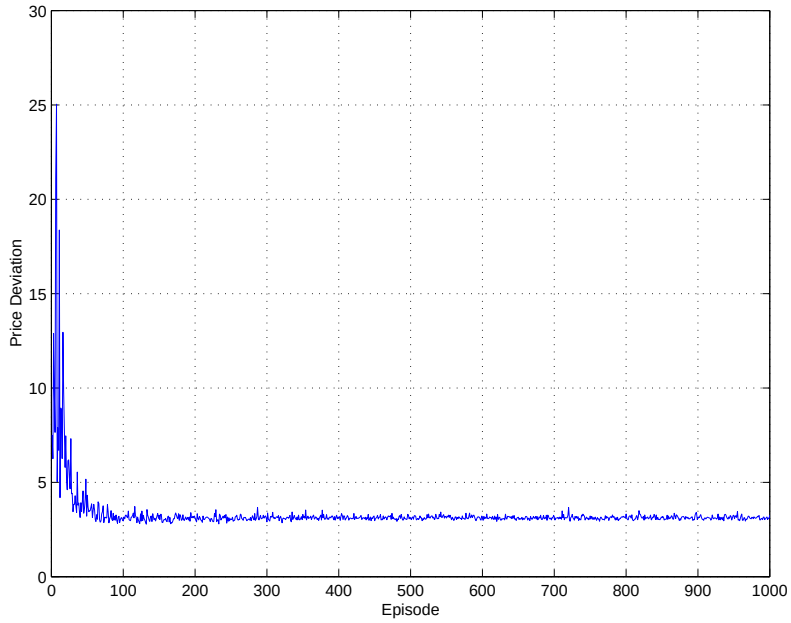

Figure 10c: Episodic average absolute price deviation, $\overline{\Delta_p}$, of a typical epoch of Experiment 3 with $w_{qlt} = 0.1$.

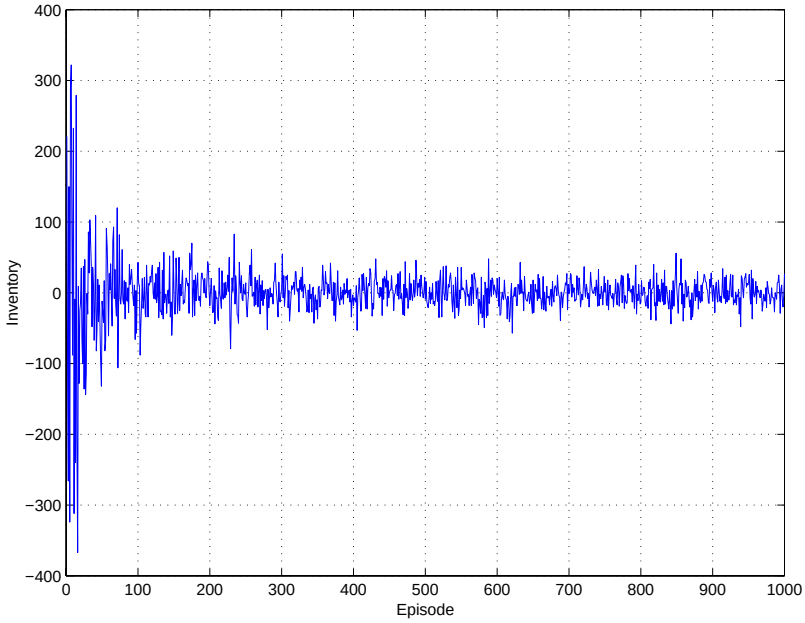

Figure 10d: End-of-episode inventory of a typical epoch of Experiment 3 with $w_{qlt} = 0.1$.

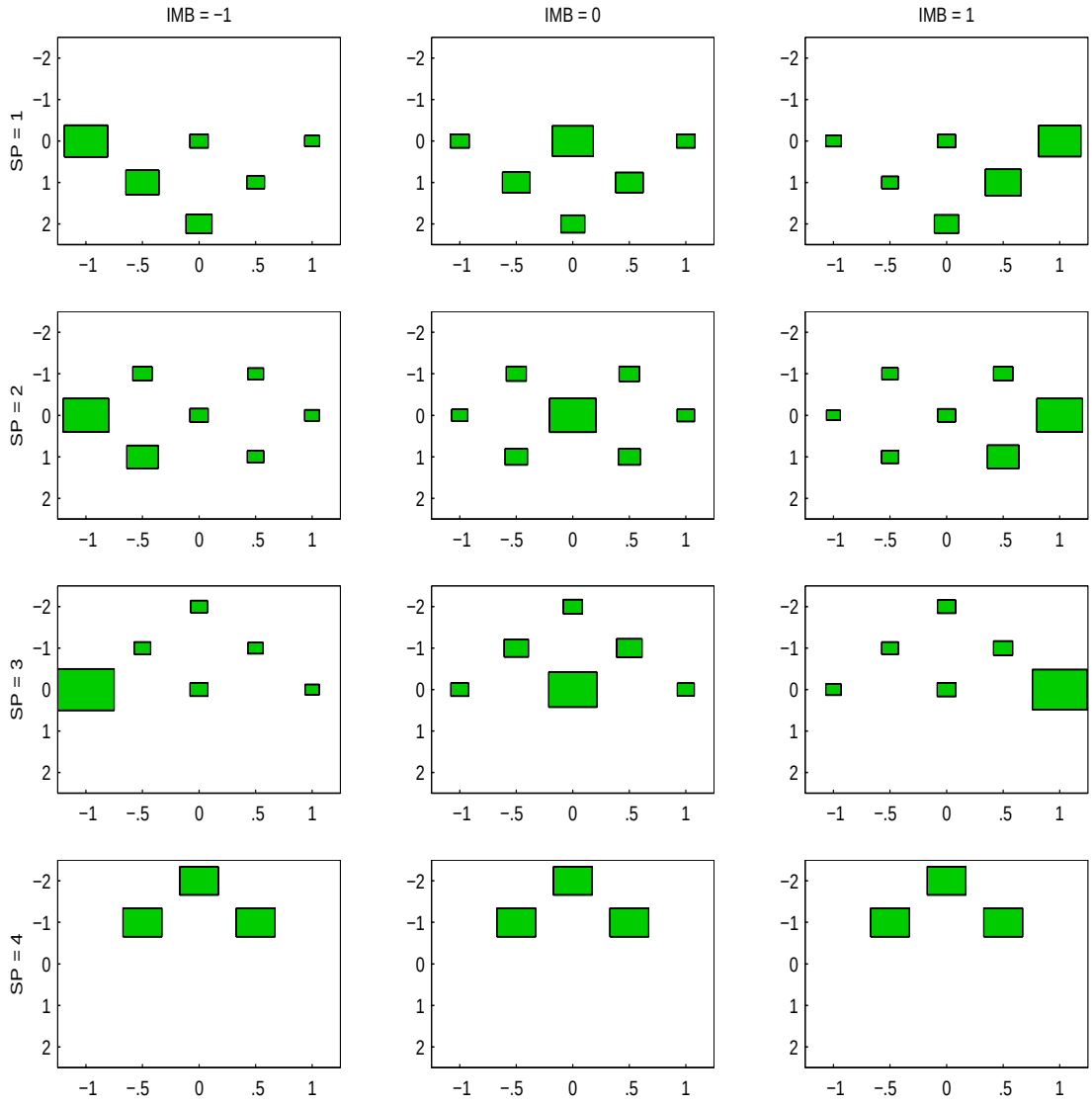

Figure 11: Conditional probability distribution of actions given imbalance and spread in a typical epoch of Experiment 3 with $w_{qlt} = 0.1$. Each probability distribution is depicted as a grid of squares whose areas represent the actual probability of pairs of bid/ask actions. In each panel, the change of the mid-quote and the change of the spread are shown on x-axis and y-axis respectively. For example, the panel at the third row and first column shows the conditional probability $\Pr(\mathbf{a} = \mathbf{a}' | IMB = -1, SP = 3)$. The action $\mathbf{a}' = (\Delta BID = -1, \Delta ASK = -1)$, which is equivalent to a change of mid-quote of -1 and a change of spread of 0, has the highest probability among all actions. In general, areas that appear in the upper (lower) portion of the panel represent a tendency to reduce (raise) the spread; areas that appear to the left (right) of the panel represent a tendency to decrease (increase) the mid-quote price.

SARSA or actor-critic. The spread, as shown in Figure 12, decreases with an increasing value of w_{qlt} in Experiments 3 and 3a. For each w_{qlt}, the mean, median and deciles of the average episodic spread are shown. The variance of the average spread is due to the stochastic nature of the algorithm, randomness in the order flow and true price process, and the imperfect state information. Comparing the results of Experiments 3 and 3a, we notice that stochastic policies yield a much lower variance for the resulting spread than deterministic policies do. As we expect, stochastic policies are better able to control partially observable environments.

Figure 13 presents the relationship between spread and profit in Experiment 3. Profit increases with spread as is expected. The results also indicate that to make a zero profit, the market-maker must maintain a spread approximately between 2.8 and 2.9.

In Experiment 4, the algorithm successfully enforces a zero profit in the market. The mean, median and standard error of profit are -0.48, 2.00 and 2.00 respectively, while the mean and median of spread are 2.83 and 2.84 respectively. This result agrees with the results from Experiment 3. The empirical distributions of profit and spread are shown in Figures 14a and 14b.

6 Conclusions

This paper presents an adaptive learning model for market-making in a reinforcement learning framework. We develop explicit market-making strategies, achieving multiple objectives under a simulated environment.

In the basic model, where the market-maker quotes a single price, we are able to determine the optimum strategies analytically and show that the reinforcement algorithms successfully converge to these strategies. In the SARSA experiment, for example, given the reward at each time step, a significant percentage of the epochs converges to the optimum strategies under moderate noise environments. It is also important to point out that the algorithm does not always converge to a single strategy, primarily due to the partial observability of the problem.

The basic model is then extended to allow the market-maker to quote bid and ask prices. While

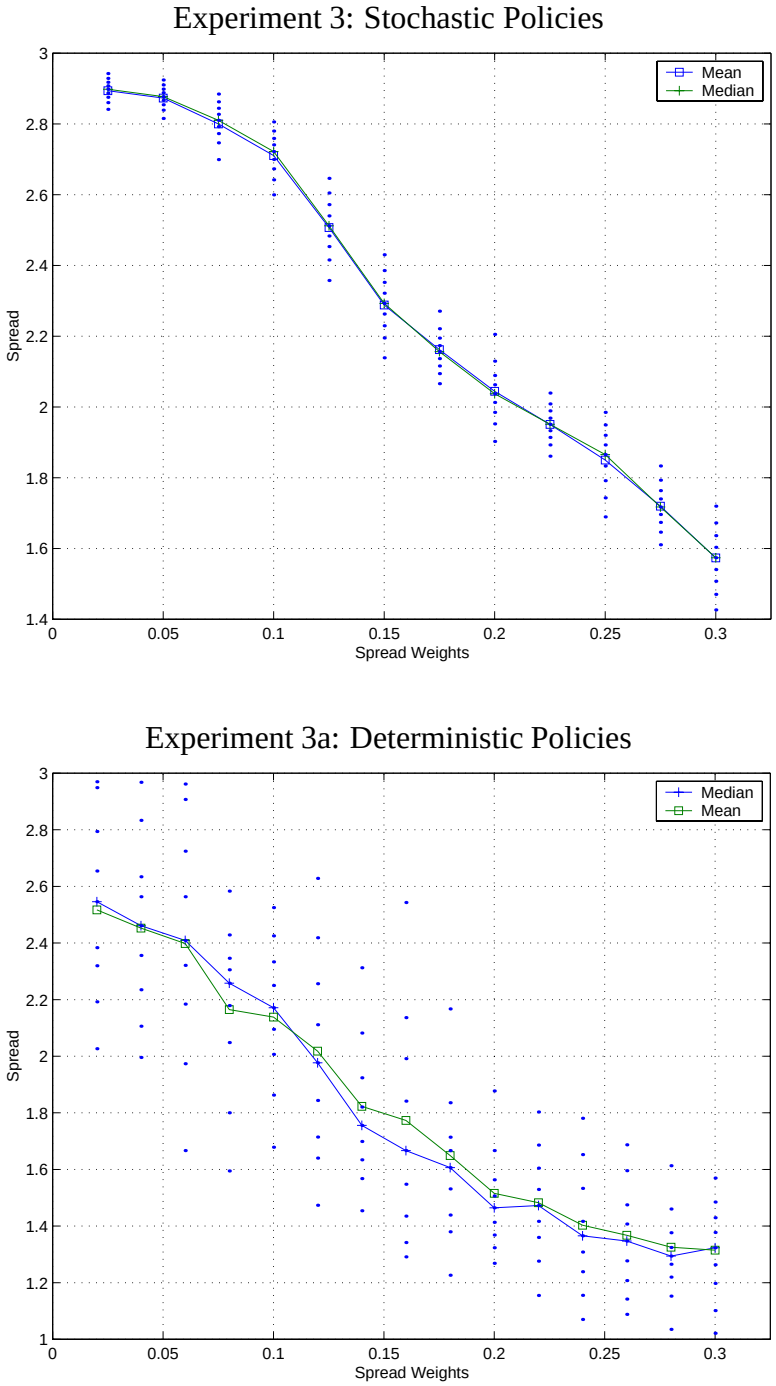

Figure 12: Spread weight versus episodic average spread in Experiment 3 and 3a. The deciles, median and mean of average episodic spread, $\overline{SP}$, of all the episodes over all epochs, are shown for different values of w_{qlt}. For both experiments, the spread decreases with the weight parameter, but the variance of the spread is much lower for the actor-critic method that yields stochastic policies.

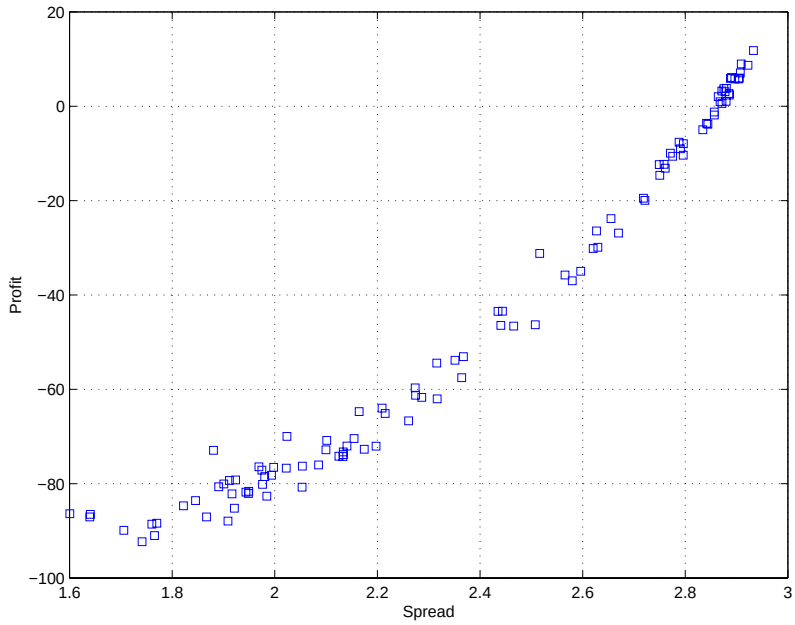

Figure 13: Episodic average spread versus end-of-period profit in Experiment 3. The figure presents the average $\overline{SP}$ versus the average PRO_T over all episode of an epoch. The profit goes up with the spread.

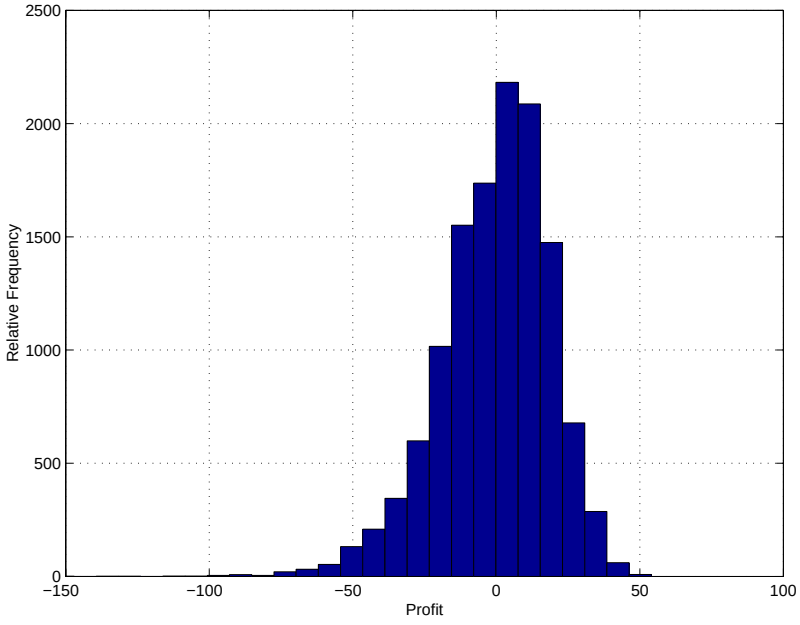

Figure 14a: Empirical distribution of end-of-episode profit, PRO_T, in Experiment 4. The mean, median and standard error of the profit is -0.48, 2.00 and 19.36 respectively.

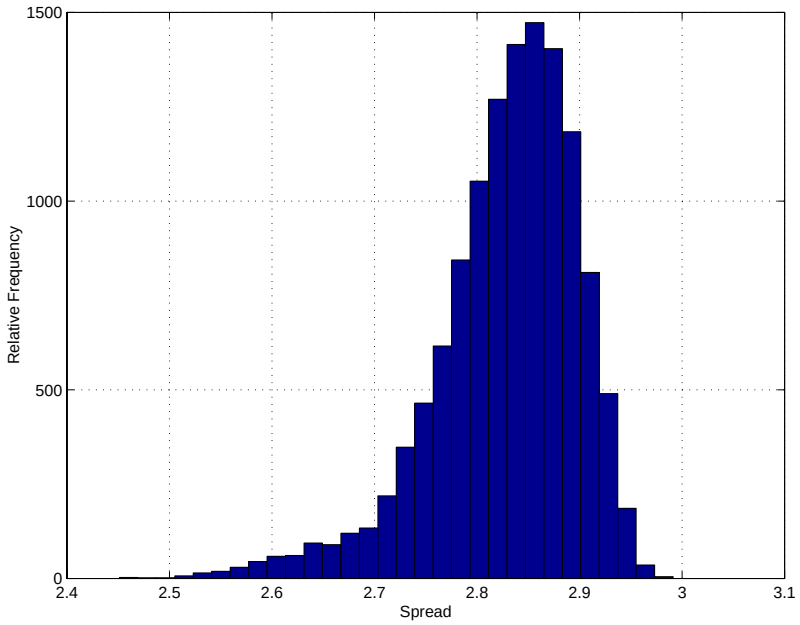

Figure 14b: Empirical distribution of average episodic spread, $\overline{SP}$ in Experiment 4. The mean, median and standard error of the spread is 2.83, 2.84 and 0.07 respectively.

the market-maker only controls the direction of the price in the basic model, it has to consider both the direction of the price and the size of the bid-ask spread in the extended model. The actor-critic algorithm generates stochastic policies that correctly adjust bid/ask prices with respect to order imbalance and effectively control the trade-off between the profit and the spread. Furthermore, the stochastic policies are shown to out-perform deterministic policies in achieving a lower variance of the resulting spread.

Reinforcement learning assumes no knowledge of the underlying market environment. This means that it can be applied to market situations for which no explicit model is available. We have shown initial success in bringing learning techniques to building market-making algorithms in a simple simulated market. We believe that it is ideal to use the agent-based approach to address some of the challenging problems in the study of market microstructure. Future extensions of this study may include the setup of more realistic and complex market environments, the introduction of additional objectives to the market-making model, and the refinement of the learning techniques to deal with issues such as continuous state variables.

References

Amihud, Y. & Mendelson, H. (1980), 'Dealership market: Market-making with inventory', *Journal of Financial Economics* **8**, 31–53.

Bertsekas, D. P. & Tsitsiklis, J. N. (1996), *Neural-Dynamic Programming*, Athena Scientific, Belmont, MA.

Bishop, C. M. (1995), *Neural Networks for Pattern Recognition*, Clarendon Press, Oxford.

Evgeniou, T., M., P. & Poggio, T. (2000), 'Regularization networks and support vector machines', *Advances in Computational Mathematics* **13**(1), 1–50.

Garman, M. (1976), 'Market microstructure', *Journal of Financial Economics* .

Glosten, L. R. & Milgrom, P. R. (1985), 'Bid, ask and transaction prices in a specialist market with heterogeneously informed traders', *Journal of Financial Economics* **14**, 71–100.

Ho, T. & Stoll, H. R. (1981), 'Optimal dealer pricing under transactions and return uncertainty', *Journal of Financial Economics* .

Ho, T. & Stoll, H. R. (1983), 'The dynamics of dealer markets under competition', *Journal of Finance* .

Jaakkola, T., Jordan, M. I. & Singh, S. P. (1994), 'On the convergence of stochastic iterative dynamic programming algorithms', *Neural Computation* **6**, 1185–1201.

Jaakkola, T., Singh, S. P. & Jordan, M. I. (1995), Reinforcement learning algorithm for partially observable markov decision problems, *in* G. Tesauro, D. S. Touretzky & T. K. Leen, eds, 'Advances in Neural Information Processing Systems', MIT Press, Cambridge, MA, pp. 345–352.

Kaelbling, L. & Moore, A. (1996), 'Reinforcement learning: A survey', *Journal of Artificial Intelligence Research* pp. 237–285.

Lutostanski, J. (1982), Liquidity and Market-making, PhD thesis, Massachusetts Institute of Technology.

Minsky, M. L. (1963), Steps towards artificial intelligence, *in* E. A. Feigenbaum & J. Feldman, eds, 'Computers and Thought', McGraw-Hill, pp. 8–30.

O'Hara, M. & Oldfiled, G. (1986), 'The microeconomics of market making', *Journal of Financial and Quantitative Analysis* .

Sutton, R. & Barto, A. (1998), *Reinforcement Learning – an Introduction*, The MIT Press, Cambridge, MA.

Sutton, R. S. (1996), Generalization in reinforcement learning: Successful examples using sparse coarse coding, *in* D. S. Touretzky, M. C. Mozer & M. E. Hasselmo, eds, 'Advances in Neural Information Processing Systems: Proceesings of the 1995 Conference', MIT Press, Cambridge, MA, pp. 1038–1044.

Vapnik, V. (1995), *The Nature of Statistical Learning Theory*, Springer.

Vapnik, V. (1998), *Statistical Learning Theory*, Wiley.

Watkins, C. J. C. H. (1989), Learning from Delayed Rewards, PhD thesis, Cambridge University.

Watkins, C. J. C. H. & Dayan, P. (1992), 'Q-learning', *Machine Learning* **8**, 279–292.