Q-Learning and Algorithmic Market Making: Loss-free, Collusive, or Competitive Prices?

Antonio Guarino* Philippe Jehiel† James Symons-Hicks‡

October 26, 2026

Abstract

We study whether algorithmic market makers using Q-learning choose competitive or supra-competitive prices in a quote-driven asset market. We show, analytically and through simulations, that the outcome depends on the algorithm's design. Basic Q-learning leads to supra-competitive prices that differ from collusive (monopoly) prices and arise because algorithms tend to abandon loss-generating actions. Q-learning with counterfactual updating produces competitive prices. Crucially, when market makers choose between these algorithms, adoption of counterfactual updating is always an equilibrium and often the unique one, thus leading to competitive pricing.

*Guarino: Department of Economics, University College London and CEPR (email: a.guarino@ucl.ac.uk).

†Jehiel: PSE and University College London (email: jehiel@enpc.fr).

‡Symons-Hicks: University of California, Berkeley (email: james.symons-hicks@berkeley.edu).

We are grateful to the editor, the referees, Antonio Cabrales, David Easley, Xavier Gabaix, Piero Gottardi, Tuomas Sandholm, colleagues and students at LSE, PSE, and UCL, and participants in several conferences and seminars for useful comments. We are also grateful to Jing Hang, Xiaotong Wu and Yihan Xu for outstanding research assistance. We acknowledge the use of the UCL Myriad High Performance Computing (Myriad@UCL) and the LSE Fabian High Performance Computing facilities, and associated support services, in the completion of this work.

1 Introduction

Does algorithmic market making lead to supra-competitive prices, and, if so, should such prices be interpreted as evidence of collusion? This issue is the focus of the paper, and is of particular importance given the increasing prevalence of algorithms in modern financial markets. Our purpose is not to study specific algorithms used in practice. Rather, we aim to provide analytically tractable benchmarks that shed light on the main forces underlying how algorithms operate and how financial firms compete in their choice of algorithms. An advantage of our approach is that it allows us to provide formal results, rather than relying solely on simulations.

Our study is based on a simplified version of the canonical Glosten and Milgrom (1985) model of a quote-driven asset market. The novelty is that market makers, rather than choosing prices to maximize expected profits, delegate their pricing decisions to Q-learning algorithms. While standard analyses of financial markets typically assume that the model is known (indeed, that it is common knowledge), reinforcement-learning algorithms are model-free. The algorithms do not know the model and, instead, learn through a simple process of choosing actions and observing the resulting rewards.

We start our analysis with a baseline Q-learning algorithm. Each Algorithmic Market Maker (AMM) attaches a Q-value to every possible ask and bid. These values represent the perceived profitability of the quotes. The algorithm chooses the quotes with the highest Q-values, except when it experiments by selecting prices randomly. After observing its realized payoff, it updates the Q-values of the prices it chose, leaving those of all other prices unchanged. The algorithm, therefore, learns only from the outcomes of its own choices, without using information about prevailing market quotes or the relationship between prices and the probability of trading.

Our baseline simulations, with a constant learning rate and exponentially declining experimentation, show that this learning rule leads to supra-competitive pricing. In particular, all AMMs eventually choose asks weakly above the highest possible asset value, v_H, and bids weakly below the lowest possible value, v_L. At these prices, informed traders cannot earn positive profits from their information. Profitable market making survives only to the extent that noise traders remain willing to trade at such extreme quotes; without that willingness, the outcome would be a market breakdown.

To understand this finding, we study the case in which Q-learning operates without experimentation. We establish a loss-free pricing result: with a constant learning rate, no experimentation, and a mild assumption on initial Q-values, all AMMs eventually choose asks weakly above v_H and bids weakly below v_L, as in our baseline simulations. The mechanism is simple.

An ask below v_H may be profitable on average, but selling at that ask when the asset value is high produces a loss. With a constant learning rate, a sufficiently long sequence of unfavorable outcomes can push its Q-value below zero. Once this happens, the algorithm abandons the ask because a loss-free alternative has a non-negative Q-value. Without experimentation, the abandoned ask is never revisited and its assessment is never corrected. Thus, the algorithm avoids the possibility of losses rather than learning to balance losses against gains.

We establish these results under two different assumptions about noise traders: in the first, they are price-inelastic (as is often assumed in empirical market microstructure), while in the second, they exhibit some price elasticity. We obtain exactly the same loss-free pricing result in both cases, despite the fact that, in the first case, the monopoly ask price is higher than the loss-free ask, whereas in the second it is lower. This provides further evidence that the tendency of Q-learning algorithms to set supra-competitive prices has nothing to do with AMMs colluding to maximize joint profits.

Having established that Q-learning AMMs are governed by the possibility of losses, rather than by the trade-off between margins and trading volume, we establish a second result that makes the limitations of basic Q-learning particularly clear. We let a single AMM compete against a market maker that simply posts a fixed ask and a fixed bid. These fixed quotes are supra-competitive but lie inside the loss-free thresholds, so the AMM could earn positive expected profits by undercutting them. Nevertheless, we prove that the algorithm eventually quotes a higher ask and a lower bid, ceases to trade, and leaves all trading profits to its competitor. The failure to compete, therefore, does not even require strategic interaction between two learning algorithms. An AMM that is outplayed by such a simple pricing strategy appears poorly suited to market making. This is why we interpret the baseline results primarily as evidence of the limitations of this learning rule, rather than as a general prediction that algorithmic market making produces supra-competitive prices.

A fundamental limitation of basic Q-learning is that it makes very limited use of available information: the algorithm learns only from the realized payoffs associated with its chosen bid and ask. Financial firms, however, invest substantial resources in developing and improving their trading algorithms, and it seems highly implausible that they would ignore readily available market information. To this end, we modify our baseline Q-learning algorithm by drawing on a well-established idea in economics and game theory: learning from counterfactual outcomes (see, e.g., Camerer and Ho, 1999). Our objective is to investigate what happens when algorithms make use of even a minimal amount of market information and economic understanding.

An AMM that sells at a given ask can infer that it would also have sold at a lower ask. An AMM that observes another market maker selling can infer that it could have captured the trade by offering a strictly lower price. Similar reasoning applies to bids. We incorporate these observations into a counterfactual-updating version of Q-learning. The algorithms observe market outcomes and prevailing quotes, as would be observed in actual financial markets, and update not only the prices they chose, but also other prices for which the observed outcome supports an inference. Importantly, they do not need to know the model's primitives or the demand function, but simply that demand slopes downwards and supply upwards.

This modest change produces a sharp contrast with the baseline results. Under standard conditions on declining learning rates and a sufficiently large weight on counterfactual feedback, we prove convergence to competitive prices in the limit as the price grid becomes arbitrarily fine. Our analysis draws on stochastic approximation methods (see Benaïm, 1999), and the learning-rate conditions are the familiar Robbins–Monro conditions used in Q-learning convergence results. Our simulations produce prices at or close to the competitive benchmark. Counterfactual updating allows the algorithms to learn about profitable undercutting opportunities without first posting the corresponding quotes themselves. Under these learning-rate conditions, payoff noise is averaged out, so the relevant comparison is between expected profits: undercutting a supra-competitive quote attracts additional order flow, whereas setting an ask below the competitive ask or a bid above the competitive bid generates losses in expectation. The same algorithm also learns to undercut a fixed-price market maker and capture the trading opportunities that baseline Q-learning fails to exploit.

The existence of a competitive algorithm does not by itself settle the question of market outcomes. Financial firms choose which algorithms to deploy, and they might prefer designs that sustain supra-competitive profits. We, therefore, study an algorithm-choice game in which firms choose between baseline and counterfactual-updating Q-learning. We show that an outcome in which all firms adopt counterfactual updating is always an equilibrium in our theoretical benchmark. In many cases, counterfactual updating is a dominant strategy, and the competitive equilibrium is unique. Thus, even when firms would earn higher profits collectively from loss-free pricing, competition in algorithm choice can lead them to adopt designs that produce competitive prices.

Counterfactual updating can be viewed as an economic response to the limitations of basic Q-learning. As a final contribution of the paper, we also examine a model-free response drawn from the machine-learning literature: Deep Q-Networks (DQN), as in Mnih et al. (2015). DQN improves upon the baseline Q-learning algorithm by replacing Q-tables with a neural-network

approximation of Q-values and introducing experience replay, which reuses batches of observations stored in a buffer. By averaging over past realized payoffs, experience replay reduces the influence of individual recent outcomes and moves learning towards comparisons of average profitability. With sufficiently large buffers and batches, prices move closer to competitive levels, although not as close as under counterfactual updating. We find that experience replay, rather than the network architecture, creates this movement towards competitive prices. These findings provide further evidence that the outcomes of algorithmic market making depend crucially on algorithm design.

Our work is related to that of Colliard et al. (2025), who also study Q-learning algorithmic market makers in a market with adverse selection. Using simulations, they show that Q-learning algorithms fail to converge to competitive prices and attribute this failure to limited experimentation and noisy feedback about the profitability of undercutting. We prove analytically that supra-competitive prices arise from the algorithm's tendency to abandon quotes that can generate losses, even when they are profitable on average. We also prove that the same mechanism prevents the algorithm from competing successfully even against a market maker using the trivial strategy of setting fixed quotes. We interpret these findings as evidence that basic Q-learning is poorly suited to market making, rather than as a compelling explanation for supra-competitive prices in real financial markets. As discussed, we also study counterfactual-updating Q-learning and examine how changing the algorithm's use of information affects both pricing and firms' incentives to adopt it. This analysis of counterfactual updating and algorithm choice is absent from Colliard et al. (2025), but is central to the comparison with Asker et al. (2024), who study algorithmic pricing in an oligopoly setting. Asker et al. (2024), like us, also show that the adoption of a different algorithm—which they call synchronous updating and which is similar to our counterfactual updating—produces competitive prices. The difference between their results and ours may at first appear merely technical—stemming from deterministic rather than stochastic payoffs—but it is actually substantial. In their oligopoly setting, adopting asynchronous updating, which is similar to our baseline Q-learning and sustains supra-competitive prices, is a dominant strategy. In our financial market, by contrast, adopting counterfactual-updating Q-learning is typically the dominant strategy, leading to competitive prices. More broadly, our analysis contributes to the literature on algorithmic pricing and collusion, including Calvano et al. (2020), while isolating a mechanism that does not rely on history-dependent pricing strategies. It also complements work on algorithmic trading under asymmetric information, such as Dou et al. (2025), by focusing on market makers rather than informed traders. Cartea et al. (2022) study the role of tick size in an algorithmic market without private information; our

focus is instead on the interaction between stochastic trading profits, learning rules, and algorithm choice. These distinctions matter for interpreting supra-competitive outcomes: similar price patterns can arise from different mechanisms and need not have the same implications for competition or policy.

The remainder of the paper is organized as follows. Sections 2–5 describe the market, the competitive and monopoly benchmarks, and the baseline Q-learning algorithm. Section 6 presents the loss-free results. Sections 7 and 8 introduce counterfactual updating and establish its pricing implications. Section 9 studies competition in algorithm choice. Section 10 explains DQN, discusses the broader implications of our analysis and concludes by outlining directions for future research.

2 The Model

We consider a sequential trade model based on the seminal paper of Glosten and Milgrom (1985). Following Easley et al. (1997) and Cipriani and Guarino (2014), we consider a multiple-day version of the model. A single risky asset is traded in a specialist market over multiple days by informed and noise traders. The novelty of our model is that the specialist (also known as a market maker) sets ask and bid prices using a Q-learning algorithm. Given our purposes, rather than considering the entire sequence of trades in each day, we will specialize the analysis to the first trading time of each day.

2.1 The Asset

There is one risky asset, and its fundamental value in day d ($d = 1, ..., D$) is denoted by V^d. The asset value does not change during the day, but can change from one day to the next. Each day, with probability δ, the value of the asset, V^d, is equal to v_H and, with probability $1 - \delta$, is equal to v_L, where $v_H > v_L \geq 0$ and $\delta v_H + (1 - \delta)v_L = \overline{v}$.[1] At the end of the trading day, the value of the asset is known to all market participants.

[1]Note that v_H and v_L are the realizations of the random variable V^d. Throughout the text, we will denote random variables with capital letters and their realizations with lowercase letters. Note also that we use both subscripts and superscripts. A superscript should not be interpreted as an exponent, unless it is obvious from the context.

2.2 The Market

The asset is exchanged in a specialist market. Its price is set by market makers who interact with a sequence of traders. Each trading day consists of T trading times, also referred to as trading periods. As we said, for our purposes, we will consider only one period; this is equivalent to setting $T = 1$.

Each day, a trader is randomly chosen to act and can buy, sell, or decide not to trade. Each trade consists of the exchange of one unit of the asset for cash. The trader's action space is, therefore, $\{-1, 0, 1\}$, where 1 and -1 are interpreted as the trader buying one unit and selling one unit of the asset, respectively. We denote the action of the trader in day d by X^d.

2.2.1 The Market Makers

Each day d, N market makers, indexed by $i = 1, ..., N$, set the prices at which a trader can buy or sell the asset.

We denote market maker i's ask price (the price at which a trader can buy) by $a_i^d \in \mathcal{A} \equiv [\underline{\alpha}, \overline{\alpha}]$ and, similarly, the bid price (the price at which a trader can sell) by $b_i^d \in \mathcal{B} \equiv [\underline{\beta}, \overline{\beta}]$. Traders trade at the best ask and bid prices, denoted as follows:

$$a^d = \min\{a_1^d, ..., a_N^d\} \quad \text{and} \quad b^d = \max\{b_1^d, ..., b_N^d\}. \tag{1}$$

In the case that there are multiple market makers quoting the best price, and a trader wants to trade at that price, one market maker is randomly chosen, with equal probability, from the set of market makers quoting that price. We denote the market maker with the prevailing ask and bid quotes by i_{ask}^* and i_{bid}^*. Each day, market maker i's trade is equal to

$$\Phi_i^d = \begin{cases} 1 & \text{if } X^d = -1 \text{ and } i = i_{bid}^*, \\ -1 & \text{if } X^d = 1 \text{ and } i = i_{ask}^*, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Market maker i's profit on day d is equal to

$$\pi_i^d = (a_i^d - v^d)\mathbf{1}_{\{\Phi_i^d = -1\}} + (v^d - b_i^d)\mathbf{1}_{\{\Phi_i^d = 1\}}, \tag{3}$$

where $\mathbf{1}_{\{.=.\}}$ is the indicator function.

2.2.2 The Traders

There is an uncountable number of traders. Each trader is chosen to take an action only once. Traders are of two types: informed and noise. The trader's own type is private information. On any day d, an informed trader is chosen to trade with probability μ and a noise trader with probability $1-\mu$, with $\mu \in (0,1)$. Informed traders have private information on the value of the asset. In particular, they receive a perfectly informative private signal on the asset value; that is, they know the realization of V^d.

An informed trader's profit on day d is defined as

$$u^d = (v^d - a^d)\mathbf{1}_{\{X^d=1\}} + (b^d - v^d)\mathbf{1}_{\{X^d=-1\}}. \tag{4}$$

An informed trader chooses X^d to maximize expected profits; that is, they are risk neutral. Since a trader knows the asset value, they find it optimal to buy whenever $a^d \leq v^d$, and sell whenever $b^d \geq v^d$. They choose not to trade when $b^d < v^d < a^d$.[2]

As for noise traders, they trade due to heterogeneous private valuations of the asset. We consider two cases. In the first case, noise traders are price inelastic: they buy with probability $\frac{\eta}{2}$, sell with probability $\frac{\eta}{2}$, and do not trade with probability $1-\eta$ (with $0 < \eta \leq 1$). This is equivalent to noise traders having either a private value larger than $\overline{\alpha}$ or less than $\underline{\beta}$, both with probability $\frac{\eta}{2}$. In the second case, noise traders are price elastic: their valuation of the asset is drawn from a uniform distribution with support $[v_L - \xi, v_H + \xi]$, with $0 \leq \xi \leq v_L$. Therefore, they buy when their valuation of the asset is larger than the ask price, sell when it is less than the bid, and do not trade when it is strictly in between the bid and the ask.

The case of inelastic noise traders is often used in the empirical market microstructure literature to estimate the probability of informed trading and other features of the order flow (see, e.g., Easley et al., 1997, 2012). The case of elastic noise traders with a uniform distribution of private values is proposed by Glosten (2009) and Glosten and Putnins (2016) to study the welfare cost of informed trading, and used, for instance, by Cipriani and Guarino (2014) in their structural estimation of the effects of a financial transaction tax. Glosten and Putnins (2016) prove that these private values can be the result of risk aversion and trading hedge reasons.

[2] We are implicitly assuming that $a^d > b^d$, which is typically the case. If, instead, that were not the case, a trader may find it profitable both to buy *and* to sell the asset. In such a case, the informed trader chooses the action that earns the highest profit. Also, note that we are assuming that the trader trades in the case of indifference. Alternative tie-breaking rules would not alter the analysis.

3 Competitive and Monopoly Pricing

Before discussing algorithmic market making, it is worth discussing the competitive equilibrium price, as well as the price when market makers maximize their joint profits. We will refer to the latter case as the monopoly price.

Market makers face an adverse selection problem since a trader deciding to trade may be doing so because they know the value of the asset. In a perfectly competitive market, market makers make zero profits in expected value. Therefore, the prevailing quotes are equal to the expected value of the asset, conditional on trade at the quoted prices:

$$a^d = E(V^d | X^d = 1, a^d, b^d), \tag{5}$$

$$b^d = E(V^d | X^d = -1, a^d, b^d). \tag{6}$$

In the case of inelastic noise traders, solving Equation 5 and Equation 6, we obtain:

$$a^d = \frac{(1-\mu)\eta E[V^d] + 2\mu\delta v_H}{2\mu\delta + (1-\mu)\eta} \equiv a^{C_I}, \tag{7}$$

$$b^d = \frac{(1-\mu)\eta E[V^d] + 2\mu(1-\delta) v_L}{2\mu(1-\delta) + (1-\mu)\eta} \equiv b^{C_I}. \tag{8}$$

In the case of elastic noise traders, the fixed-point conditions (5) and (6) yield quadratic equations in a^d and b^d. Solving equation (5), we obtain the equilibrium ask as the smaller root; similarly, solving equation (6), we obtain the equilibrium bid as the larger root:[3]

$$a^d = \frac{v_H + \xi + \bar{v} + \kappa_a - \sqrt{(v_H + \xi + \bar{v} + \kappa_a)^2 - 4\left[(v_H + \xi)\,\bar{v} + \kappa_a v_H\right]}}{2} \equiv a^{C_E}, \tag{9}$$

$$b^d = \frac{v_L - \xi + \bar{v} - \kappa_b + \sqrt{(v_L - \xi + \bar{v} - \kappa_b)^2 - 4\left[(v_L - \xi)\,\bar{v} - \kappa_b v_L\right]}}{2} \equiv b^{C_E}, \tag{10}$$

where $\kappa_a \equiv \frac{\mu\delta(v_H - v_L + 2\xi)}{1-\mu}$ and $\kappa_b \equiv \frac{\mu(1-\delta)(v_H - v_L + 2\xi)}{1-\mu}$.

Let us now move to the case in which market makers collude to maximize joint profits. In the case of price inelastic noise traders, market makers set an ask price $a^d = \overline{\alpha}$ and a bid price $b^d = \underline{\beta}$. The reason is that, at these prices, informed traders do not trade, whereas noise traders trade with the same probability as at any other price. Therefore, there is no trade-off:

[3] In general, the ask is the minimum solution for Equation 5 and the bid is the maximum solution for Equation 6. See, e.g., Cipriani and Guarino (2008) for the formal argument showing the uniqueness of ask and bid.

posting the highest possible ask and the lowest possible bid maximizes joint profits. Clearly, the specific prices $\overline{\alpha}$ and $\underline{\beta}$ are only due to our assumptions that the market makers choose prices in a bounded interval and that noise traders are willing to trade at any possible price.

In the case of price elasticity, instead, there is a trade-off, since, e.g., increasing the ask price lowers the probability of selling to a noise trader. The maximization of the joint profits leads to the following result:

$$a^d = \begin{cases} \frac{(1+\delta-\mu)v_H+(1-\delta-\mu)v_L+(1-\mu+2\mu\delta)\xi}{2(1-\mu)} & \text{if } \xi \leq \frac{(v_H-v_L)(1-\delta-\mu)}{1-\mu+2\mu\delta}, \\ v_H & \text{if } \frac{(v_H-v_L)(1-\delta-\mu)}{1-\mu+2\mu\delta} < \xi \leq (1-\delta)(v_H-v_L), \\ \frac{(1+\delta)v_H+(1-\delta)v_L+\xi}{2} & \text{if } \xi > (1-\delta)(v_H-v_L), \end{cases} \tag{11}$$

$$b^d = \begin{cases} \frac{(\delta-\mu)v_H+(2-\delta-\mu)v_L-(1+\mu-2\mu\delta)\xi}{2(1-\mu)} & \text{if } \xi \leq \frac{(v_H-v_L)(\delta-\mu)}{1+\mu-2\mu\delta}, \\ v_L & \text{if } \frac{(v_H-v_L)(\delta-\mu)}{1+\mu-2\mu\delta} < \xi \leq \delta(v_H-v_L), \\ \frac{\delta v_H+(2-\delta)v_L-\xi}{2} & \text{if } \xi > \delta(v_H-v_L). \end{cases} \tag{12}$$

Note that, for a sufficiently low value of ξ, the ask price is strictly *below* v_H, and, similarly, the bid price is strictly *above* v_L. Price elasticity gives an incentive to market makers to lower their ask prices (increase their bid prices) to increase their probability of trading with a noise trader.

4 Algorithmic Market Making

We now consider the case in which the function of market making is delegated to machines that set the ask and bid prices according to a Q-learning algorithm (Watkins, 1989). While in the analysis of the competitive equilibrium, as standard, we have assumed that the model is common knowledge and that rational market makers and rational informed traders maximize their objective function on the basis of all available information, the starting point of algorithmic market making is that the machines have limited knowledge of the environment in which they are operating. The machines are used because they can exploit large data to learn the best pricing strategy, with minimal informational input.

The N Algorithmic Market Makers (AMMs) operating in our financial market set the prices

based on a Q-learning algorithm, an example of off-policy reinforcement learning. The term "off-policy" refers to the fact that the algorithm uses a behavioral policy different from the optimal one. In particular, we consider an *ε-greedy* policy: each day, AMM_i chooses the greedy action (i.e., chooses the price with the highest perceived payoff) with probability $1-\varepsilon_i^d$ and the explorative action (i.e., chooses a random price from its action set) with the complementary probability, ε_i^d. Later, in Section 10, we will also consider a different policy.

We consider the case in which the algorithms have very limited knowledge of the environment and set their prices without conditioning on any specific variable. This is often referred to as "stateless Q-learning," although some authors prefer to say that the state space is a singleton.

On each day d, each market maker i chooses the ask price a_i^d in the discrete space $\mathbf{A} = \{\alpha_1, \alpha_2, ..., \alpha_J\}$, where $\alpha_1 < \overline{v}$ and $\alpha_J > v_H$. Similarly, they choose the bid price b_i^d in the discrete space $\mathbf{B} = \{\beta_1, \beta_2, ..., \beta_J\}$, where $\beta_1 < v_L$ and $\beta_J > \overline{v}$. Each market maker attaches a particular value, called a "Q-value," $q_i^d(\alpha_j)$ to each possible ask price, and, similarly, a value $\hat{q}_i^d(\beta_j)$ to each possible bid price. These values represent the perceived reward for choosing a given price. On the first day, $d = 1$, the Q-vectors q_i^1 and $\hat{q}_i^1$ take random values.

On each following day, the Q-vectors are updated on the basis of the payoffs that the market maker receives that day. In particular, the Q-learning rule is as follows. If AMM_i chooses ask $a_i^d = \alpha_j$ and bid price $b_i^d = \beta_k$, then, following any market order (buy, sell, or no trade), the Q-values for these particular quotes are updated as follows:

$$q_i^{d+1}(\alpha_j) = \lambda_i^d \left(\left(\alpha_j - v^d \right) \mathbf{1}_{\{\Phi_i^d = -1\}} \right) + \left(1 - \lambda_i^d \right) q_i^d(\alpha_j), \tag{13}$$

$$\hat{q}_i^{d+1}(\beta_k) = \lambda_i^d \left(\left(v^d - \beta_k \right) \mathbf{1}_{\{\Phi_i^d = 1\}} \right) + \left(1 - \lambda_i^d \right) \hat{q}_i^d(\beta_k), \tag{14}$$

where $\lambda_i^d \in [0, 1]$ is the weight that market maker i places on the realized payoff on that day (the "learning rate").

For all other (i.e., unchosen) ask prices α_l ($l \neq j$) and bid prices β_m ($m \neq k$), the Q-values are unchanged:

$$q_i^{d+1}(\alpha_l) = q_i^d(\alpha_l), \tag{15}$$

$$\hat{q}_i^{d+1}(\beta_m) = \hat{q}_i^d(\beta_m). \tag{16}$$

It is important to note that, in this formulation, a chosen ask or bid is updated independently of whether a trader buys, sells or does not trade. An alternative formulation could consist in "decoupling" the updating, that is, updating the Q-value for, e.g., the chosen ask only when

the trader buys or does not trade, but not when they are on the other side of the market, that is, when they sell.[4] All the results we present in Sections 6–8 refer to the main formulation, but also hold under this alternative formulation. In Section 9, we will discuss when decoupling the updating may make a difference.

As we said, on each day, AMM_i chooses to "exploit" or "explore", with probability $1 - \varepsilon_i^d$ and ε_i^d, respectively. The value of ε_i^d is given by

$$\varepsilon_i^d = c_i + (1 - c_i)e^{-\gamma_i d}, \tag{17}$$

where c_i is a constant (weakly between 0 and 1) and γ_i is a parameter (weakly larger than 0) that controls the rate at which this probability converges to the constant. If a market maker chooses to exploit on a given day, they do so for both the ask and bid prices; the same is true if they choose to explore. The market maker exploits the greedy action by choosing the ask price α_j associated with the highest value of q_i^d and the bid price β_k associated with the highest value of $\hat{q}_i^d$. That is, the greedy ask and bid prices of AMM_i on day d are given by

$$a_i^{d*} = \arg\max_{\alpha_j} q_i^d(\alpha_j), \tag{18}$$

$$b_i^{d*} = \arg\max_{\beta_k} \hat{q}_i^d(\beta_k). \tag{19}$$

When, instead, the AMM explores, α_j and β_k are drawn independently from uniform distributions over $\mathbf{A}$ and $\mathbf{B}$, respectively.

Note that, in our analysis, we will also consider a variant of this model, with only one AMM ($N = 1$) and one "human" market maker. Nevertheless, the general analysis is for $N \geq 2$ AMMs that compete among themselves.

5 Implementation of the Baseline Case

Let us now implement our baseline financial market model with AMMs through simulations. We run each simulation for 500,000 days, to give plenty of time for price convergence. Of course, taken literally, 500,000 days is an unrealistically long period. We use this terminology mainly for exposition. Day d can be thought of as any unit of time after which the asset value changes, and the process of trading restarts, so that the market maker faces exactly the same situation,

[4] The updating rules for the "decoupled" updating are described in Appendix C.11.

that is, they face the same pricing problem.[5] We repeat the simulations 1,000 times and then present average results over these repetitions.

For our baseline simulation, we specialize the analysis as follows (for the reader's convenience, we summarize all parameter values in Table 1).

First, we set the two asset values to be $v_H = 102$ and $v_L = 98$, and the probability of these two values to be $\delta = 0.5$. This implies that the mean value of the asset, $\overline{v}$, is equal to 100. Moreover, for the case of price inelastic noise traders, we set $\frac{\eta}{2} = \frac{1}{2}$, that is, we assume that a noise trader buys or sells with probability 0.5 and never decides not to trade. For this case, we assume that the probability of an informed trader trading is $\mu = 0.3$. In the price-elastic case, we assume that $\xi = 0.25$ and set $\mu \approx 0.239$ so that the competitive equilibrium prices are the same in the two cases, for ease of comparison.[6]

Second, we consider the case in which there are $N = 2$ AMMs. They both choose ask prices in $\mathbf{A} = \{\alpha_1, \alpha_2, ..., \alpha_J\}$, where, $\alpha_1 = 99.5$ and $\alpha_J = 103$. Furthermore, we assume that the other ask prices are evenly spaced between these two values, with a tick size of 0.025. Note that $\mathbf{A}$ satisfies the conditions indicated in Section 4, that is, $\alpha_1 < \overline{v}$ and $\alpha_J > v_H$. Similarly, they choose bid prices in $\mathbf{B} = \{\beta_1, \beta_2, ..., \beta_J\}$, where, $\beta_1 = 97$ and $\beta_J = 100.5$, and the other bid prices are evenly spaced between these two values, with a tick size of 0.025. Overall, market makers choose among 141 ask and 141 bid prices, that is, $J = 141$.

Third, as far as the Q-learning algorithm is concerned, we assume that both market makers use the same learning rate $\lambda_i^d = \lambda = 0.1$, constant over all days. They explore with probability $\varepsilon_i^d = \exp(-0.00004d)$, that is, $c_i = 0$ and $\gamma_i = 0.00004$. Finally, we assume that the initial Q-values, $q_i^1(\alpha_j)$ and $\hat{q}_i^1(\beta_j)$, for each AMM are independently drawn from the same uniform distribution on $[5, 8]$.

A few comments are in order. The values for the asset and for the traders are chosen for illustration, and do not play a particular role. The two sets for ask and bid prices for the AMMs are chosen as if the AMMs know that the asset value in the previous day is 100 and might have gone up or down, but by not more than 3. It is of course important that these sets include the competitive equilibrium prices, since we are interested in understanding whether the prices set by the AMMs converge to these prices. Given our parameter values, if the market makers could choose in a continuous space, the competitive equilibrium bid and ask prices, as computed in

[5] For instance, Easley et al. (2012) advocate the use of volume time rather than clock time. In their framework, the process of trading "restarts" every time a "volume bucket" is filled.

[6] In Appendix C.6, we also consider the price-elastic case in which $\mu = 0.3$.

Table 1: Baseline Simulation

Parameter	Value	Description
The Simulation		
K	1,000	Number of simulation repetitions
D	500,000	Number of trading days
The Asset		
v_H	102	High value of the asset
v_L	98	Low value of the asset
δ	0.5	Probability of the high asset value
The Traders		
μ	0.3/0.239	Informed trading mass (inelastic/elastic)
$\frac{\eta}{2}$	$\frac{1}{2}$	Price-inelastic noise trader buy/sell probability
$[v_L - \xi, v_H + \xi]$	[97.75, 102.25]	Price-elastic noise trader's valuation support
The AMMs		
N	2	Number of AMMs
λ_i^d	0.1	Learning rate
γ_i	0.00004	Exploration decay rate
c_i	0	Minimum exploration rate
J	141	Number of discrete prices

expressions 7 and 8 (as well as in expressions 9 and 10), would be 99.4 and 100.6, respectively.[7] Because of the discrete space, with a tick size of 0.025, the price inelastic case yields three pure-strategy-equilibrium bid prices (99.35, 99.375, and 99.4) and ask prices (100.6, 100.625, and 100.65). In the price elastic case, there are two pure-strategy-equilibrium bid prices (99.375 and 99.4) and ask prices (100.6 and 100.625).[8] As for the collusive (monopolistic) ask and bid prices, in the case of price-inelastic noise traders, they are equal to 103 and 97, respectively, as explained in Section 3; for the price-elastic case, they are equal to 101.48 and 98.52, as computed in expressions 11 and 12. The value of the exploration rate can be more easily interpreted by

[7]Recall that we modified the probability of an informed trader trading in the case with price-elastic noise traders exactly to have the same competitive equilibrium prices.

[8]The first equilibrium ask price corresponds to the zero-profit condition. The second equilibrium occurs one tick above it: at that price, a unilateral undercut by one tick would lead to the zero-profit ask, and is, therefore, not profitable. In the price-inelastic case, there is also a third equilibrium because the no-undercutting condition binds: by undercutting by one tick, a market maker would capture the whole demand but earn exactly the same expected profit as when sharing the market at the higher ask. This equality does not hold in the price-elastic case, where undercutting is profitable and the third price is, therefore, not an equilibrium. Analogous considerations apply to the equilibrium bids. We refer the reader to Appendix B for the details.

looking at the number of times the algorithm explores each possible price. As $D \to \infty$, the number of times the AMM is expected to explore is

$$\sum_{d=1}^{\infty} e^{-\gamma_i d} = \frac{e^{-\gamma_i}}{1 - e^{-\gamma_i}},$$

which, for $\gamma_i = 0.00004$, is equal to 25,000; this means that, in expectation, each price is randomly visited approximately 177 times. The learning rate of 0.1 means that the Q-value for a particular price is updated rather substantially whenever the price is chosen by the AMM. Finally, the initial Q-values are set at least equal to 5, which is the maximum profit an AMM can obtain, e.g., by selling an asset worth 98 for 103. Because of these high initial Q-values, all actions will be played initially as Q-values are revised downwards, thus preventing the algorithm from becoming stuck at certain prices. This feature will play a role in our results, and we will discuss it in the next section.

6 Results for the Baseline Case

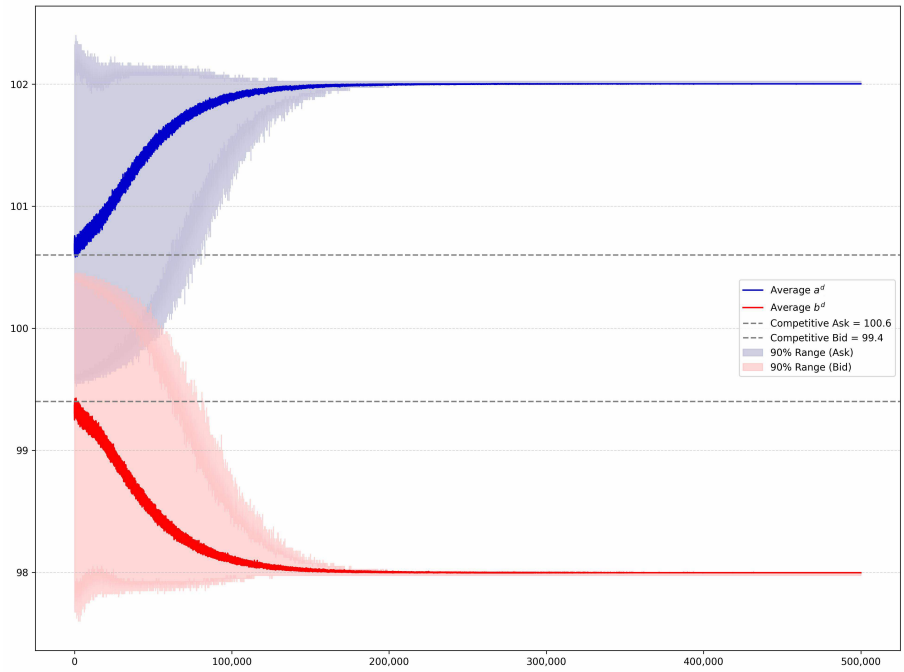
We now illustrate the results for the baseline case. Figure 1 shows the mean of the best ask prices and of the best bid prices (the market prices) across our 1,000 repetitions over the 500,000 days for both cases of price-inelastic and price-elastic noise traders.[9] The mean ask price tends towards v_H and, similarly, the mean bid price tends towards v_L. The convergence is reached after approximately 200,000 days. The graphs also report the 90% ranges for the best ask and bid prices across the simulations, in lighter blue and lighter red. In the first days, the ranges are large, as the probability of exploration is near one; as the exploration falls, the AMMs settle on ask prices around 102 and bid prices around 98. Appendix Figures C1 and C2 show how the distribution of prices across the simulations changes over time. After 250,000 days, all 1,000 repetitions have converged to ask prices weakly above v_H and bid prices weakly below v_L; in most cases, prices have converged to exactly v_H and v_L. By the final trading day, in 964 (983) of the 1,000 simulations with price-inelastic (price-elastic) noise traders, the best ask submitted by the AMMs is 102, that is, v_H, and, in the remaining cases, it is above 102. Similarly, in 974 (977) of the 1,000 simulations for the price-inelastic (price-elastic) noise traders case, the best bid is equal to 98, that is, v_L; in the remaining simulations, it is strictly lower. When the market ask (bid) is weakly higher than v_H (respectively, lower than v_L), informed traders no longer

[9] As we explained in Section 5, the word "day" is used in the model for exposition and should not be taken literally.

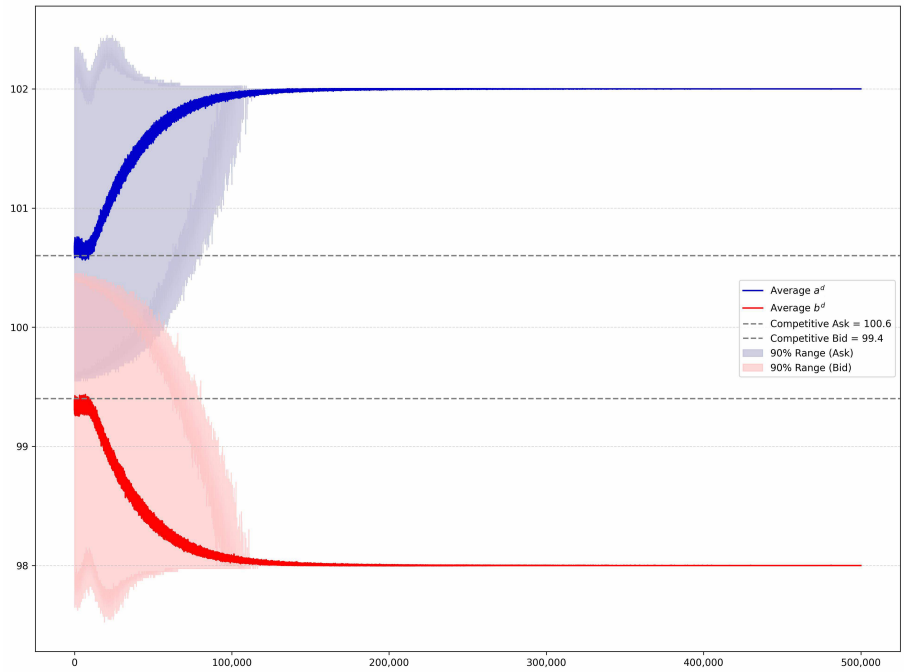
have an incentive to trade. Profitable trade from the perspective of the AMMs occurs only because we have assumed that noise traders keep trading even at the high and low realizations of the asset value (all noise traders in the case in which they are price inelastic, or at least for some valuations in the case of price elasticity). If we had assumed that noise traders do not trade when the ask and bid prices reach the highest and lowest possible asset valuation, the quotes set by the AMMs would imply a market breakdown. Note also that, by setting the same price, the two AMMs get an equal share of the market. If an AMM undercut the other, they would achieve a much larger volume of trade and higher profits in expectation.

The reason the AMMs do not undercut one another is that the Q-values associated with prices in $(98, 102)$ are all strictly negative by the end of the 500,000 days, as we show in Appendix Figures C3 and C4. Given that the probability of exploration is approximately zero in the final day, actions in the range $(98, 102)$ with negative Q-values are not picked as the greedy action because there exist actions for which the expected Q-value is, at a minimum, zero. This also occurs far before the final day. When the probability of exploration becomes sufficiently low (e.g., ε_i^d is approximately 0.03% after 200,000 days), then almost all actions chosen are the greedy actions, which, as we have seen, are asks at least equal to v_H and bids not higher than v_L. Appendix Figures C3 and C4 also show that the ask prices sufficiently above 102 and the bid prices sufficiently below 98 all have a Q-value of zero, suggesting some downward updating for asks above 102 and bids below 98. This occurs because, if one AMM has settled on an ask price of 102, then the other AMM setting a price above this would earn a payoff of zero.

So far, we have established that the AMMs' prices converge to supracompetitive levels. It is important to remark that they set exactly *the same prices*, independently of whether noise traders are price elastic or price inelastic, and that, *in neither case*, do they set the monopoly prices. As we explained in Section 5, in the price-inelastic case, the ask (bid) price that maximizes the joint profits would be the highest (lowest) admissible level, that is, 103 (97) for our parameter choices. At these prices, each AMM would earn an expected profit of 2.1, whereas by setting the ask and bid at 102 and 98 respectively, each obtains a profit of 1.4, that is, only 66.6% of the monopoly profit. In the price-elastic case, at the joint profit-maximizing ask and bid prices of 101.48 and 98.52, each AMM would obtain a profit of 0.26; their actual profit is only 64.8% of this amount. While these specific values depend on our parametric assumption, one lesson is clear: the supracompetitive prices set by the AMMs are not the result of any form of tacit collusion aimed at setting monopoly prices and maximizing joint profits: they must indeed be set for a completely different reason.



(a) Inelastic Noise Traders



(b) Elastic Noise Traders

The graph plots the mean market quotes over days, averaging across the 1,000 repetitions. The graph refers to the baseline Q-learning setup with (a) price-inelastic and (b) price-elastic noise traders. The mean ask (bid) is the dark blue (red) line; the blue and red shaded bands represent the 90% range across the simulations for the ask and bid, respectively. The dashed lines plot the competitive ask and bid prices, as discussed in Section 3. In Panel (a), the mean market ask and bid quotes in the final day are 102 and 98, respectively. In Panel (b), the mean market ask and bid quotes in the final day are 102 and 98, respectively.

Figure 1: Baseline - Mean Market Quotes Over Days

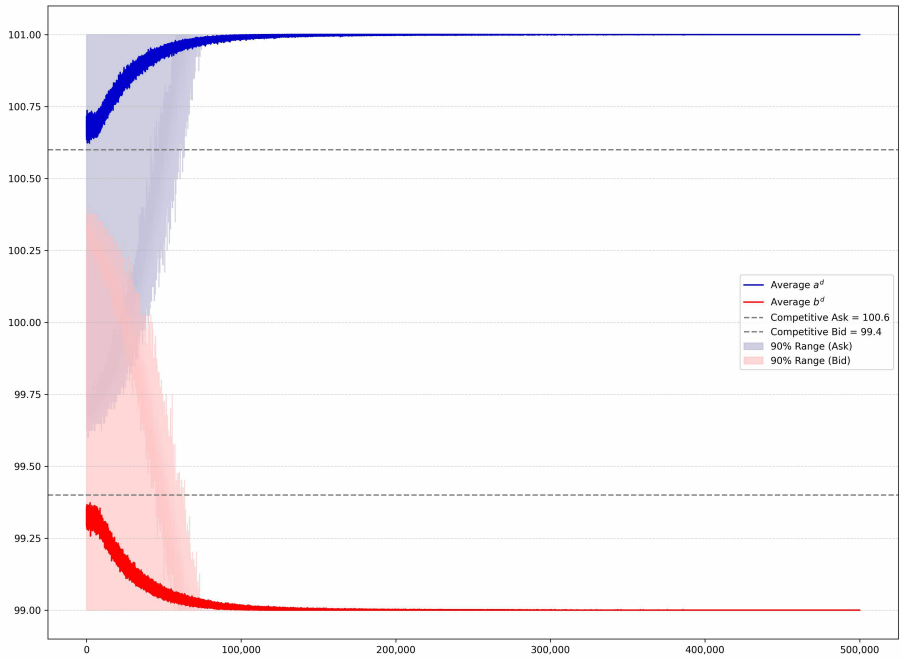
6.1 Playing Against a Fixed-price Market Maker

The bid-ask spread set by the AMMs seems implausibly large, to the point of eliminating any incentive to trade on private information. Each AMM could increase its profits consistently by undercutting the other, but it does not learn to do so. We now ask whether this result is due to the strategic interaction between the two AMMs. Specifically, we ask whether a single AMM would at least be able to learn to undercut a "human" market maker who adopts the very simple strategy of setting a fixed ask lower than the asset's high value (v_H) (and higher than the competitive equilibrium ask) and a fixed bid higher than the asset's low value (v_L) (and lower than the competitive equilibrium bid). Specifically, we set a fixed ask price of 101 and a fixed bid price of 99; these prices are away from the competitive equilibrium prices, and also away from the possible asset value realizations.

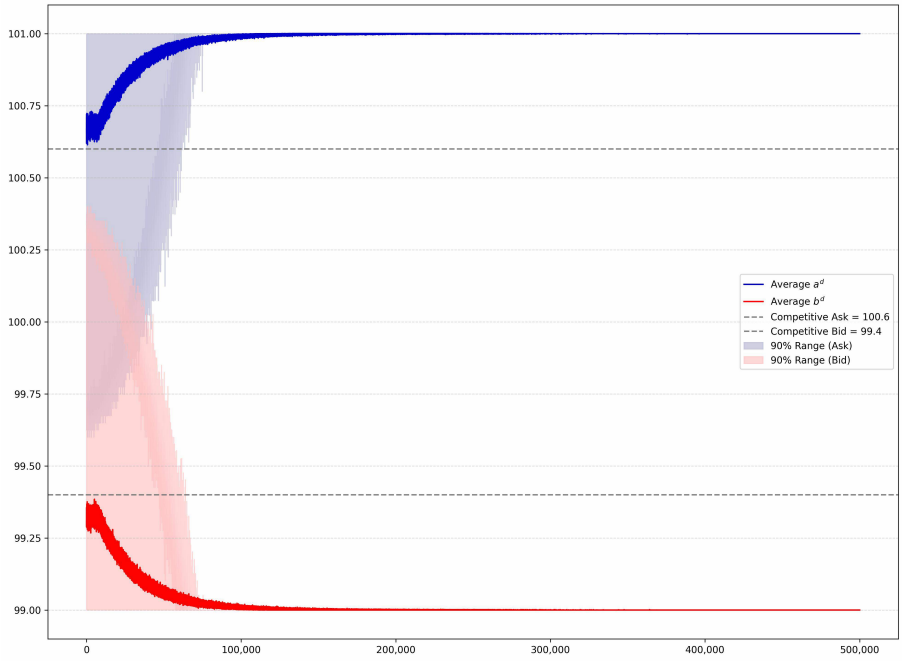
Figure 2 is the analogue of Figure 1 for this case. The market quotes, after some initial experimentation, are the two fixed prices. Appendix Figures C5 and C6 show that the AMM sets quotes that are strictly less competitive than the fixed-price market maker, with the distribution of ask quotes having support strictly above 101 and that of bid quotes having support strictly below 99. Rather than undercutting these prices by one tick, which would be the best response, the AMM chooses less competitive quotes and, therefore, does not trade.

Appendix Figures C7 and C8 show the reason for this. All ask prices weakly lower than 101 and all bid prices weakly larger than 99 are attached a negative Q-value. Ask prices strictly larger than 101 and bid prices strictly lower than 99, instead, have a Q-value of 0, which is intuitive, since every time the AMM chooses them (by experimentation or exploitation), they do not trade and, hence, receive a payoff of zero.

Overall, through simulations, we have obtained two unsettling results: first, AMMs using Q-learning do not learn to undercut, which implies an implausibly large bid-ask spread when they compete with each other; second, even more surprisingly, an AMM would not be able to outplay a human market maker who would simply set ask and bid prices at a fixed level, far from the competitive prices. By undercutting these fixed prices, the AMM would earn a large profit in expected value, but they are not able to do so. Appendix Figures C9 and C10 make this point in a stark way by comparing the cumulative profits made by the AMM and those they would have made by undercutting the human market maker by one tick size. The AMM makes some initial losses in the first days and then sets prices that are strictly less competitive; at this point, the AMM earns zero profits in each day.



(a) Inelastic Noise Traders



(b) Elastic Noise Traders

The graph plots the mean market quotes over days, averaging across the 1,000 repetitions. The graph refers to the baseline Q-learning setup with (a) price-inelastic and (b) price-elastic noise traders, against a dealer setting a fixed ask (101) and a fixed bid (99). The mean ask (bid) is the dark blue (red) line; the blue and red shaded bands represent the 90% range across the simulations for the ask and bid, respectively. The dashed lines plot the competitive ask and bid prices, as discussed in Section 3. In Panel (a), the mean market ask and bid quotes in the final day are 101 and 99, respectively. In Panel (b), the mean market ask and bid quotes in the final day are 101 and 99, respectively.

Figure 2: Baseline Q-learning AMM against a fixed-price market maker - Mean Market Quotes Over Days

6.2 The Loss-free Pricing Result

While these simulation results are, at first glance, unsettling and surprising, they have a simple explanation. We now present a formal result stating that, in the absence of experimentation, the AMMs will eventually settle down on ask prices no smaller than v_H and on bid prices no larger than v_L whenever at least one ask above v_H and one bid below v_L have weakly positive initial Q-values. Note that this is a very minimal assumption given that, for very large asks and very low bids, an AMM is sure not to make a loss.

Proposition 1. *Suppose that $\lambda_i^d = \lambda$, and $\varepsilon_i^d = 0$ for every day d and every AMM$_i$. Moreover, suppose that, for every AMM$_i$, for at least one $\alpha_j \geq v_H$ and at least one $\beta_k \leq v_L$, the initial Q-values, $q_i^1(\alpha_j)$ and $\hat{q}_i^1(\beta_k)$, are weakly positive. Then, almost surely, there exists a $\overline{d} > 0$ such that, for every day $d > \overline{d}$, it holds that $a_i^d \geq v_H$ and $b_i^d \leq v_L$ for every AMM$_i$.*

Proof. We show the result for ask prices. The same reasoning can be used to show the result for bid prices.

First, observe that if, for $\alpha_j \geq v_H$, $q_i^1(\alpha_j) \geq 0$, then $q_i^d(\alpha_j) \geq 0$ for all d, since the payoff from quoting α_j (used in the updating of the Q-values) is never negative when $\alpha_j \geq v_H$.

Next, suppose, by contradiction, that there are infinitely many days d such that $a_i^d < v_H$ for at least one AMM$_i$. Define α^* as follows:

$$\alpha^* = \arg\min(\alpha_k \text{ such that } a_i^d = \alpha_k \text{ for infinitely many } d \text{ for at least one AMM}_i).$$

Given our hypothesis (by contradiction), $\alpha^* < v_H$.

Now, observe that the Q-value $q_i^d(\alpha^*)$ is no larger than the maximum of the initial Q-value $q_i^1(\alpha^*)$ and of $\alpha^* - v_L$, since the latter is the largest payoff that can be obtained with the ask price α^*, no matter what d is.

Next, observe that, for d large enough, whenever the ask price α^* is chosen by AMM$_i$, there is a positive probability that AMM$_i$ trades (because this is the best ask price proposed by all AMMs) and that the resulting payoff is $\alpha^* - v_H$ (whenever the asset has high value).

Given that $\alpha^* - v_H < 0$ and that $\lambda_i^d = \lambda$ for all d, there must exist k^* such that

$$(1-\lambda)^{k^*} \max(q_i^1(\alpha^*), \alpha^* - v_L) + (1 - (1-\lambda)^{k^*})(\alpha^* - v_H) < 0.$$

After k^* consecutive such draws, which eventually must occur with probability 1, we have that,

almost surely, $q_i^{\overline{d}}(\alpha^*) < 0$ for some finite $\overline{d}$.

Given that $q_i^d(\alpha_j) \geq 0$ for all d, the inequality holds in particular for all $d \geq \overline{d}$. Therefore, it cannot be that $a_i^d = \alpha^*$ for any $d \geq \overline{d}$, since for $d = d^*$, AMM$_i$ cannot choose α^* as $q_i^{d^*}(\alpha_j) \geq 0 > q_i^{d^*}(\alpha^*)$. For $d > d^*$, AMM$_i$ cannot choose α^* as the updating rule ensures that, for all $d > d^*$, $q_i^d(\alpha_j) \geq 0 > q_i^d(\alpha^*) = q_i^{d^*}(\alpha^*)$. This leads to the contradiction that α^* is not chosen by AMM$_i$ infinitely many times. Q. E. D. □

The logic of the proof is simple. For any ask price α^* strictly lower than v_H, there is a positive probability that the market maker choosing it sells at this price (if this market maker has the lowest ask price) and the value of the asset is high, resulting in a strictly negative payoff $\alpha^* - v_H$. After enough consecutive such events, which occur with probability 1 in the infinite sequence of days, this would lead $q_i^{d^*}(\alpha^*)$ to become negative, which, in turn, would lead this market maker to never consider this ask price again when there is no experimentation (because the Q-values for at least one $\alpha_j \geq v_H$ are always weakly positive).[10]

Our result is related to the *maxmin* result obtained by Sarin and Vahid (1999) in an individual learning environment in which different alternatives yield stochastic rewards. Considering a learning model like ours (with a constant weight on the realized immediate reward in the updating of Q-values), they show that, eventually, the decision maker chooses an alternative with the highest minimum payoff among the chosen alternatives. Note that, in our model with one AMM and one market maker setting a fixed price, the result by Sarin and Vahid (1999) indeed provides the insight that the AMM would never learn to undercut (because the undercutting would not be best from this *maxmin* perspective), resulting in all trade being performed by the fixed-price market maker. We now state this result formally:

Proposition 2. *Consider the case of one AMM (denoted AMM$_1$) that competes with a market maker choosing a fixed ask price $\overline{a}^d = \overline{\alpha} < v_H$ and a fixed bid price $\overline{b}^d = \overline{\beta} > v_L$ for all days d. Suppose that $\lambda_1^d = \lambda$, $\varepsilon_1^d = 0$ for all days d, and that the initial Q-values $q_1^1(\alpha_j)$ and $\hat{q}_1^1(\beta_k)$ are weakly positive for at least one ask price $\alpha_j \geq v_H$ and at least one bid price $\beta_k \leq v_L$. Then, almost surely, there exists some $\overline{d} > 0$ such that, for all days $d > \overline{d}$, $a_1^d > \overline{\alpha}$ and $b_1^d < \overline{\beta}$. Asymptotically, almost surely, the AMM will not trade.*

Proof. First, note that the minimum payoff for all asks $\alpha_j > \overline{\alpha}$ and all bids $\beta_k < \overline{\beta}$ is zero.

[10] Our assumption that at least one ask price above v_H has a weakly positive initial Q-value rules out initial assessments so pessimistic that all loss-free asks are permanently excluded from consideration.

In contrast, all asks weakly lower than $\overline{\alpha}$ and all bids weakly higher than $\overline{\beta}$ have a negative minimum payoff (in the case of equality, there is a positive probability the AMM will buy an asset with value v_L or sell an asset with value v_H). Then, observe that Proposition 1 and Corollary 1 of Sarin and Vahid (1999) immediately imply that asymptotically, for $d > \overline{d}$, almost surely, $a_1^d > \overline{\alpha}$ and $b_1^d < \overline{\beta}$. Given these ask and bid prices, asymptotically, almost surely, any trade occurs with the market maker setting a fixed price. Q. E. D. □

In our simulation with a fixed-price market maker, we still observe this result even if we allow for some exponentially decaying experimentation.[11]

In our Proposition 1, we cannot use the result of Sarin and Vahid (1999) because of the interaction among N learning players. But, similarly to Sarin and Vahid (1999), we obtain that algorithmic learning leads the market makers to choose loss-free prices in the absence of experimentation. This result would extend to the case in which the absence of experimentation holds in the limit, that is, if there is no experimentation from a time t onward.[12] Furthermore, the simulation with exponentially decaying experimentation presented above leads to observations very much in line with those obtained theoretically without experimentation. In Section 10, we go beyond this case and simulate the model with various experimentation rates and learning weights, as well as with logit exploration. We observe convergence to loss-free prices for a large range of parameters. Persistent experimentation (with a fixed experimentation probability bounded away from zero) and smaller updating weights lead to ask prices that are, on average, lower than the loss-free threshold, but still very far from the competitive equilibrium and with many AMMs still settling within the loss-free region in a high percentage of the 1,000 simulations; thus, the qualitative insight is unchanged.

It is worth noting that, while Proposition 1 rules out ask and bid prices in (v_L, v_H) in the long run, we mostly observe ask prices close to v_H and bid prices close to v_L in the final days of the simulations. All ask (bid) prices weakly above v_H (respectively, below v_L) have the *maxmin* property; hence, they can all arise according to the logic used to establish our result. Nevertheless, it would be possible to refine our result and show that, most of the time, AMMs choose v_H (respectively, v_L) as the ask (respectively, bid) price, while some stochasticity

[11] Sarin and Vahid (1999) obtain their result by making an assumption of "optimism" for all initial Q-values (but see their footnote 10). In the proposition, we make the weaker assumption that at least one initial Q-value has to be weakly positive, which is very natural in our context. In the simulations, we assumed that all initial Q-values are "optimistic". In other simulations, we use lower, but positive, initial Q-values, but this does not alter the results (see Appendix C.8).

[12] We refer the reader to Appendix C.7 for more details and simulation results.

persists in the limit.[13] The reason for the stochasticity is simple. To see it, let us focus on the ask prices, and suppose that there exists a $\bar{d}$ such that the ask price $a^d = \min\{a_1^d, ..., a_N^d\}$ is equal to α_j for all $d > \bar{d}$. Clearly, given Proposition 1, $\alpha_j \geq v_H$; let us consider, in particular, $\alpha_j > v_H$. Any AMM$_i$ who does not sell (either because they have chosen an ask strictly higher than α_j or because they have chosen α_j but, by randomness, another AMM with the same ask is selected to trade, or because there is no buy order) keeps receiving a payoff of zero. Therefore, for their chosen ask α_k, $q_i^d(\alpha_k)$ becomes closer to zero. After a sufficiently long sequence of such events, $q_i^d(\alpha_k)$ becomes smaller than the Q-value for another ask price, say α_l ($l \neq j$). AMM$_i$ then switches to α_l. Finally, note that if $\alpha_l > \alpha_j$, AMM$_i$ continues not to trade, hence also $q_i^d(\alpha_l)$ becomes closer to zero. All Q-values for ask prices weakly larger than α_j will eventually become smaller than the Q-value for some smaller ask price, which contradicts that $a^d = \alpha_j$ for all $d > \bar{d}$.

While this argument shows why Q-learning cannot converge to a single ask price strictly larger than v_H, it also unveils why, in most cases, the ask price is exactly equal to v_H. When an AMM chooses a higher ask price, they receive a zero payoff, and the logic just illustrated applies. Intuitively, among the ask prices weakly larger than v_H (the only ones not ruled out by Proposition 1), the ask price $a_i^d = v_H$ is the most frequently chosen, asymptotically, because it is the price at which it is more likely that an AMM will sell. It is not the ask price chosen indefinitely because it is possible that all AMMs earn a series of zero profits (e.g., either because there is trade on the other side of the market, or because the value of the asset is v_H) which causes $q_i^d(v_H)$ to fall towards zero, leading to $q_i^d(\alpha_j) > q_i^d(v_H)$, for some $\alpha_j > v_H$. When this occurs for all AMMs, they will temporarily switch to a higher ask price. Note that this argument is different from the usual undercutting argument, in which the comparison between the payoffs of two prices leads to choosing the lower one. With Q-learning, for an AMM, a higher ask price leads to no trade and, hence, to a decreasing Q-value, which eventually leads the AMM to switch from that ask.[14]

[13] In our argument, we assume that the grid of ask (bid) prices includes v_H (v_L). Furthermore, we assume that at least v_H (v_L) and another ask (bid) price strictly above v_H (below v_L) have a strictly positive initial Q-value.

[14] As a final note, it is worth observing that this logic does not hold in the case in which there is only one AMM competing with a market maker setting fixed ask and bid prices. In this case, there is no reason to expect that the AMM will mainly choose an ask close to the fixed ask price (101 in our example). On the contrary, the Q-values for all the ask prices above 101 (the fixed ask price) will eventually reach the same level and all these ask prices will keep being chosen. Accordingly, our simulations show that, in the last day of trading, the distribution of all ask prices chosen by the AMM as the greedy action is approximately uniform, above 101 (see Appendix Figures C5 and C6).

7 Counterfactual Updating

In a standard theoretical analysis of financial markets, we typically assume that the model is known by all players (in fact, even common knowledge among them). The advantage of Q-learning pricing algorithms is that we can dispense with such an assumption: in our stateless Q-learning, the AMMs only know the range in which to find the best quotes, their own payoffs, and nothing else. While a complete knowledge of the model is a heroic assumption, we can, however, endow AMMs with at least an elementary understanding of a financial market. It seems highly plausible that a market maker should understand that a trader willing to buy the asset at a specific ask price would also buy it at any lower price. Similarly, they should understand that a trader willing to sell the asset at a specific bid price would also be willing to sell it at any greater price. We follow this line of reasoning and let AMMs update not only the Q-values for the quotes they chose, but also for the other quotes for which an obvious inference could be made. In addition to this simple understanding of how markets work, we endow the AMMs with minimal information, that of the market quotes (i.e., the best ask and the best bid), an uncontroversial hypothesis for any market maker model.[15]

As an example, consider the case of two AMMs, and suppose a trader bought the asset from AMM_1 at the price of $a^d = \alpha_j$. AMM_1 knows that the trader would have also bought had they chosen any other price lower than α_j. Hence, in our new algorithm, they also update the Q-values for all ask prices lower than α_j. They do not update Q-values for ask prices greater than α_j since they do not know for sure whether they would have been able to sell at those prices. What about AMM_2 who had chosen an ask greater than α_j? Observing the trade at $a^d = \alpha_j$, they know that they could have undercut their competitor, while they would not have sold for any ask greater than α_j. For an ask price of α_j, they would have sold with probability 50%. Therefore, AMM_2 updates the Q-values for all asks: for any ask strictly greater than α_j, the Q-values are updated with a payoff of zero; for any ask weakly lower than α_j, the Q-values are updated taking into account that the AMM would have sold (with probability 50% in the case of a tie at α_j).

One could observe that a realized profit may be more relevant than a hypothetical one (i.e., in the counterfactual case) when updating the Q-values. For instance, the AMM could be unsure of whether, in the case they had chosen a lower ask, the other AMM would have also

[15] We could assume that AMMs are also aware of their competitors' ask and bid quotes (e.g., through the order book), but we do not use this assumption here.

done the same. We can accommodate this variant by discounting the counterfactual payoff by a factor ρ ($0 < \rho \leq 1$).

Let us now illustrate the general case of N AMMs in detail. The following possibilities can occur:

1. **AMM$_i$ quotes the best ask $a_i^d = \alpha_j = a^d$ and sells the asset**. AMM$_i$ updates the Q-values for α_j, the action chosen, on the basis of the *realized* payoff, and the Q-values for all asks lower than α_j using the payoffs they would have received by selling at those prices (the *counterfactual* payoff), discounted by a factor ρ. AMM$_i$ does not update the Q-value for any ask greater than α_j, since they do not know whether they would have been able to sell.

   Formally, the Q-values are updated as follows:

$$q_i^{d+1}(\alpha_j) = \lambda_i^d(\alpha_j - v^d) + (1 - \lambda_i^d)q_i^d(\alpha_j), \tag{20}$$

$$q_i^{d+1}(\alpha_l) = \lambda_i^d\rho(\alpha_l - v^d) + (1 - \lambda_i^d\rho)q_i^d(\alpha_l), \qquad \text{for all } \alpha_l < \alpha_j, \tag{21}$$

$$q_i^{d+1}(\alpha_k) = q_i^d(\alpha_k), \qquad \text{for all } \alpha_k > \alpha_j. \tag{22}$$

2. **AMM$_i$ quotes the ask $a_i^d = \alpha_j = a^d$ and another AMM sells the asset**. As AMM$_i$ knows that the other AMM also quoted α_j, they infer that they would have earned zero at any counterfactual price strictly above α_j; therefore, the Q-values are updated using the counterfactual updating weight $\lambda_i^d\rho$. The realized action of α_j is also updated using the realized profit of zero, with the usual weight λ_i^d. For ask prices strictly lower than α_j, the AMM knows that they would have undercut the market price and sold the asset; thus, the Q-values are updated with the counterfactual payoff:

$$q_i^{d+1}(\alpha_j) = (1 - \lambda_i^d)q_i^d(\alpha_j), \tag{23}$$

$$q_i^{d+1}(\alpha_l) = \lambda_i^d\rho(\alpha_l - v^d) + (1 - \lambda_i^d\rho)q_i^d(\alpha_l), \qquad \text{for all } \alpha_l < \alpha_j, \tag{24}$$

$$q_i^{d+1}(\alpha_k) = (1 - \lambda_i^d\rho)q_i^d(\alpha_k), \qquad \text{for all } \alpha_k > \alpha_j. \tag{25}$$

3. **AMM$_i$ quotes the ask $a_i^d > \alpha_j = a^d$ and another AMM sells the asset**. AMM$_i$ updates the Q-value associated with the chosen price, a_i^d, with a payoff of zero using weight λ_i^d. For all other prices strictly above α_j, the Q-values are also updated using the counterfactual payoff of zero, with weight $\lambda_i^d\rho$. The Q-value associated with α_j is updated with the *counterfactual* payoff, discounted by a factor $\frac{1}{M}$ to reflect the probability with

which the AMM would have traded. We assume that M is chosen so that the Q-values are updated as conservatively as possible (that is, with the smallest change from the current Q-value). In particular, the AMM computes the counterfactual payoff for all $M \in \{2, ..., N\}$; then, if all counterfactual profits are above (below) the current Q-value, it chooses the smallest (largest) counterfactual profit; in the case that the possible profits surround the current Q-value, the counterfactual payoff is set to be the current Q-value such that it is not updated. For any $\alpha_l < \alpha_j$, they would have undercut and their Q-values are updated using the counterfactual payoffs.

$$q_i^{d+1}(a_i^d) = (1 - \lambda_i^d) q_i^d(a_i^d), \tag{26}$$

$$q_i^{d+1}(\alpha_j) = \lambda_i^d \rho \frac{1}{M}(\alpha_j - v^d) + (1 - \lambda_i^d \rho) q_i^d(\alpha_j), \tag{27}$$

$$q_i^{d+1}(\alpha_k) = (1 - \lambda_i^d \rho) q_i^d(\alpha_k), \qquad \text{for all } \alpha_k > \alpha_j, \alpha_k \neq a_i^d, \tag{28}$$

$$q_i^{d+1}(\alpha_l) = \lambda_i^d \rho(\alpha_l - v^d) + (1 - \lambda_i^d \rho) q_i^d(\alpha_l), \qquad \text{for all } \alpha_l < \alpha_j. \tag{29}$$

4. **AMM$_i$ quotes the ask $a_i^d = \alpha_j = a^d$ and there is no buy**. Note that this can occur either because there is no trade or because there is a sell order. AMM$_i$ updates α_j with the realized profit of zero; for all $\alpha_k > \alpha_j$, the AMM updates with a counterfactual profit of zero as the trader would still not have bought. For any $\alpha_l < \alpha_j$, AMM$_i$ is unsure whether they would have been able to sell and, thus, the Q-values for these prices are not updated:

$$q_i^{d+1}(\alpha_j) = (1 - \lambda_i^d) q_i^d(\alpha_j), \tag{30}$$

$$q_i^{d+1}(\alpha_k) = (1 - \lambda_i^d \rho) q_i^d(\alpha_k), \qquad \text{for all } \alpha_k > \alpha_j, \tag{31}$$

$$q_i^{d+1}(\alpha_l) = q_i^d(\alpha_l), \qquad \text{for all } \alpha_l < \alpha_j. \tag{32}$$

5. **AMM$_i$ quotes the ask $a_i^d > \alpha_j = a^d$ and there is no buy**. Again, this can occur either because there is no trade or because there is a sell order. AMM$_i$ can update all prices weakly larger than α_j using zero profits, applying the counterfactual weight of $\lambda_i^d \rho$ to all prices except a_i^d, which is updated with weight λ_i^d. Again, the AMM does not know if they would have sold by undercutting the market quote and so does not update the

Q-value for any price below α_j.

$$q_i^{d+1}(a_i^d) = (1 - \lambda_i^d) q_i^d(a_i^d), \tag{33}$$

$$q_i^{d+1}(\alpha_k) = (1 - \lambda_i^d \rho) q_i^d(\alpha_k), \qquad \text{for all } \alpha_k \geq \alpha_j, \alpha_k \neq a_i^d, \tag{34}$$

$$q_i^{d+1}(\alpha_l) = q_i^d(\alpha_l), \qquad \text{for all } \alpha_l < \alpha_j. \tag{35}$$

For the bid, the cases are analogous; just note that higher bid prices are preferred by the traders, since they are the prices at which they sell to the market maker. We present the formulas in Appendix A.

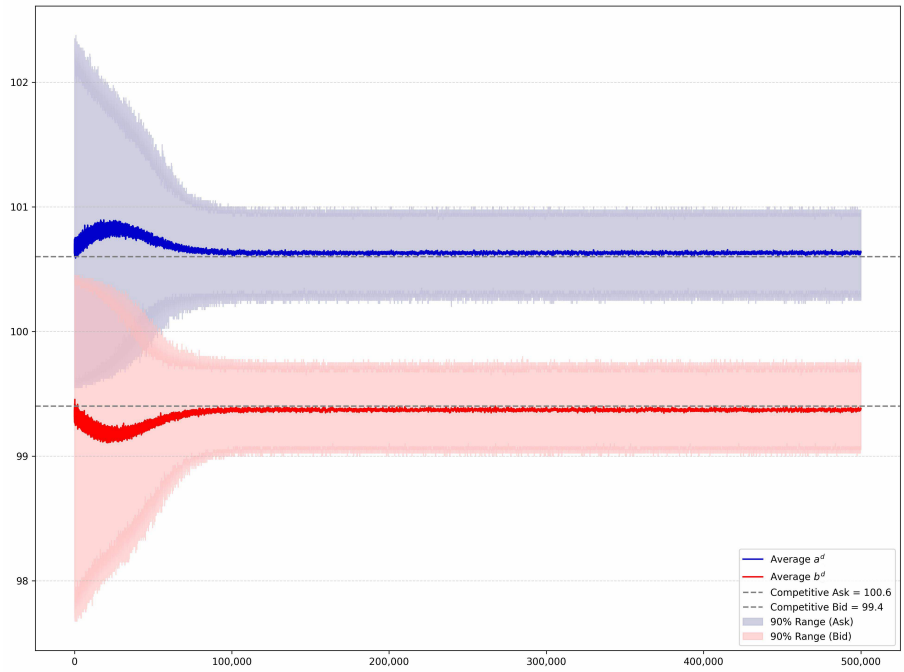
We are not the first to suggest that a model of reinforcement learning can (and, perhaps, should) incorporate some understanding of the economy or of the game that is being played. Models of learning with counterfactual updating have, indeed, been studied in game theory. Camerer and Ho (1999) propose a model of "Experience-Weighted Attraction" (EWA) learning which encompasses both reinforcement learning and belief learning (fictitious play). Our learning model with counterfactual updating is a special case of the EWA learning, and so is the individual decision making model of Sarin and Vahid (1999) previously discussed. Note that Camerer and Ho (1999) allow also for the counterfactual updating to have a lower weight (captured by their parameter δ). More recently, counterfactual updating has been studied by Asker et al. (2024) in their work on algorithmic pricing in an oligopoly. We will discuss the differences between our work and Asker et al. (2024) in detail in Section 9.

8 Results for Counterfactual Updating

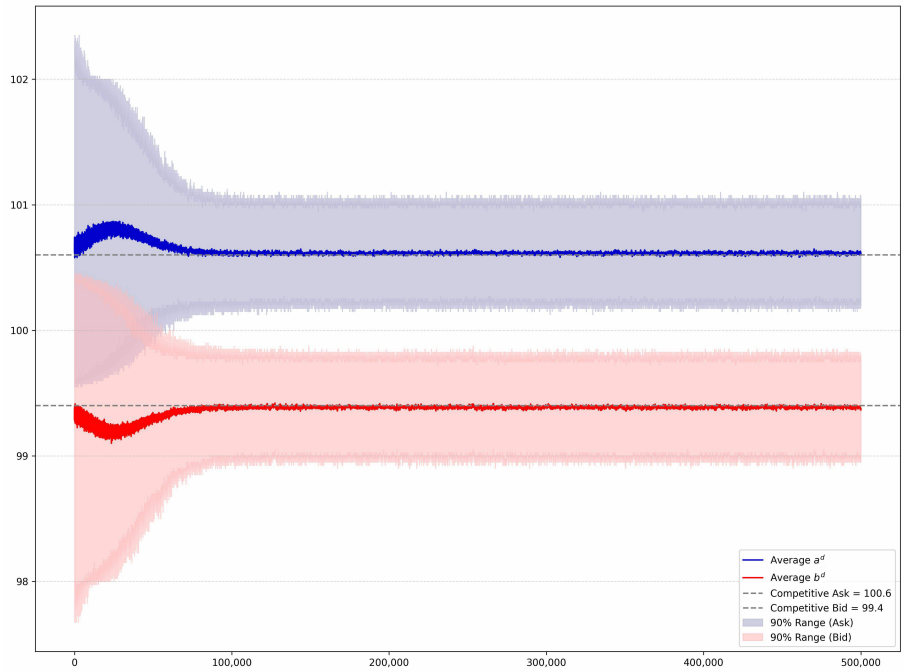
Figure 3 shows the mean market ask and bid prices over days when two Q-learning algorithms employ counterfactual updating. The figure refers to simulations in which the parameter values are identical to those listed in Table 1 (and, for exposition, we have set $\rho = 0.6$). Prices converge towards the competitive equilibrium in both cases of price-inelastic and price-elastic noise traders. In the final days, in all simulations, ask and bid are either at the competitive levels or very close to them (see Appendix Figures C11 and C12).

8.1 The Competitive Equilibrium Pricing Result

The contrast between these simulation results and those we obtained for the baseline case could not be stronger. Letting the AMMs have a minimal understanding of the market is enough to



(a) Inelastic Noise Traders



(b) Elastic Noise Traders

The graph plots the mean market quotes over days, averaging across the 1,000 repetitions. The graph refers to the counterfactual-updating Q-learning setup with (a) price-inelastic and (b) price-elastic noise traders. The mean ask (bid) is the dark blue (red) line; the blue and red shaded bands represent the 90% range across the simulations for the ask and bid, respectively. The dashed lines plot the competitive ask and bid prices, as discussed in Section 3. In Panel (a), the mean market ask and bid quotes in the final day are 100.64 and 99.38, respectively. In Panel (b), the mean market ask and bid quotes in the final day are 100.61 and 99.37, respectively.

Figure 3: Counterfactual Updating - Mean Market Quotes Over Days

move from loss-free pricing to competitive pricing.

To understand what is at the root of these simulation findings, we now provide a formal

result. To obtain a sharp theoretical result, we assume that the learning rates λ_i^d in the updating rule satisfy the Robbins-Monro conditions, $\sum_d \lambda_i^d = \infty$ and $\sum_d \left(\lambda_i^d\right)^2 < \infty$, which will allow us to rely on continuous time approximation to analyze the long-run dynamics of the system (see Benaïm (1999) for an exposition of the mathematical background establishing such connections). The reader familiar with Q-learning will notice that these conditions are the same as those used in the classic convergence theorem of Watkins and Dayan (1992). They guarantee that decision makers (the AMMs in our context) continue to learn (the learning rate decays slowly), but the learning rate is not too fast (the decay is not too slow). Similarly to the theoretical result in the baseline Q-learning model presented in Section 6, we also assume that there is no experimentation, although experimentation would not alter our conclusions much. Finally, we assume that any ask and bid prices in the continuum can be considered, even if similar conclusions would hold for the case of the finite ask and bid grid that we have employed in our simulations.[16]

With these premises, we are now ready to state that, under a minimal condition on the initial Q-values, similar to that we used for Proposition 1, there is asymptotic convergence to the competitive equilibrium ask and bid prices, a^C and b^C:

Proposition 3. *Suppose that, for every AMM_i, the sequences of λ_i^d satisfy $\sum_d \lambda_i^d = \infty$ and $\sum_d \left(\lambda_i^d\right)^2 < \infty$, and, moreover, that $\varepsilon_i^d = 0$ for all d; assume that $\rho > 0.5$ and suppose that, for every AMM_i, for at least one $\alpha_j \geq a^C$ and at least one $\beta_k \leq b^C$, the initial Q-values, $q_i^1(\alpha_j)$ and $\hat{q}_i^1(\beta_k)$, are weakly positive. Then, asymptotically, in the limit of a continuous grid of ask and bid prices, every AMM_i quotes the competitive ask price a^C and bid price b^C almost surely.*

Proof. We prove the proposition for the case of price-inelastic noise traders and, for exposition, we focus on the ask price (for the bid price one can follow identical arguments). The proof for the case of price-elastic noise traders differs only in the form of the differential equation presented below.

As a preliminary note, we find it convenient to change the notation and, instead of d (for days), we use t (for time), which is more common in continuous-time analysis. Moreover, to simplify the exposition, we will consider the case of $\lambda_i^t = \lambda^t$ for every i.

Given the conditions for λ^t, we can employ a continuous-time approximation of the discrete-time dynamics of Q-values for the ask prices. More precisely, we use Propositions 4.1 and

[16] The qualification is that, even theoretically, with a discrete grid, one or two tick sizes above the competitive equilibrium can also be an equilibrium price, as observed in Section 5.

4.2, along with the Limit Set Theorem (Theorem 5.7), of Benaïm (1999) to ensure that the asymptotics of the discrete-time dynamics are well approximated by the continuous-time motion that we now describe.

Calling $\alpha^{\min}$ the prevailing ask price at time t (recall that this is the minimum ask quoted by all AMMs at that time), the continuous-time approximation of the motion of Q-values for any ask price α strictly below $\alpha^{\min}$ can be written as

$$\frac{dq_i^t}{dt}(\alpha) = \rho[(1-\mu)\frac{\eta}{2} + \mu\delta][\alpha - a^C] - q_i^t(\alpha). \tag{36}$$

This expression can be understood as follows. The Q-values of ask prices below $\min(v_H, \alpha^{\min})$ can be updated with the counterfactual formula. When the ask price is $\alpha^{\min}$, the expected payoff obtained with any ask price strictly below $\min\{v_H, \alpha^{\min}\}$ is $[(1-\mu)\frac{\eta}{2} + \mu\delta][\alpha - a^C]$ since trade on this side occurs with probability $[(1-\mu)\frac{\eta}{2} + \mu\delta]$ and, when it occurs, it results in an expected payoff of $\alpha - a^C$ (recall that informed traders buy if $v = v_H$, and recall, also from our updating formulas that, when trade is on the other side, or there is no trade, the payoff is zero); moreover, given that $\alpha < \alpha^{\min}$, it was not chosen at time t, hence the payoff is updated with the ρ discount.

In the same spirit, for an AMM_i quoting $\alpha^{\min}$, we have:

$$\frac{dq_i^t}{dt}(\alpha^{\min}) = \frac{1}{M}[(1-\mu)\frac{\eta}{2} + \mu\delta][\alpha^{\min} - a^C] - q_i^t(\alpha^{\min}) \tag{37}$$

where M is the number of AMMs quoting $\alpha^{\min}$ at time t. The difference with the previous expression is that there is no ρ discount in the update, since the ask price $\alpha^{\min}$ is chosen by AMM_i and, in the case several (i.e., M) AMMs choose $\alpha^{\min}$, AMMs are selected to be the seller with equal probability.

Now, define α_{inf} as the infimum of the market ask prices that can be quoted asymptotically and let α_{inf}^- be the next ask price below α_{inf} in the grid (assumed to be close to the continuous grid).

We first show that α_{inf} cannot strictly exceed a^C (in the limit of the fine grid). This follows because

$$\frac{dq_i^t}{dt}(\alpha_{\text{inf}}^-) - \frac{dq_i^t}{dt}(\alpha_{\text{inf}}) > 0$$

for every AMM_i whenever $\alpha_{\text{inf}} > a^C$ and $\rho > 0.5$ and the minimum ask price is α_{inf}. (When the minimum ask price is different from α_{inf}^-, $\frac{dq_i^t}{dt}(\alpha_{\text{inf}}^-) - \frac{dq_i^t}{dt}(\alpha_{\text{inf}})$ gets arbitrarily close to zero in

the limit of the continuous grid.) This implies that eventually, for some $\bar{t}$ and for every AMM$_i$, we must have that, for all $t > \bar{t}$, $q_i^t(\alpha_{\text{inf}}^-) > q_i^t(\alpha_{\text{inf}})$, implying that α_{inf} is not chosen infinitely many times by AMM$_i$.

We next observe that we cannot have $\alpha_{\text{inf}} < a^C$ as this would imply that eventually, for some $\bar{t}$ and every AMM$_i$ and $t > \bar{t}$, $q_i^t(\alpha_{\text{inf}}) < 0$, making it impossible that α_{inf} is chosen infinitely many times by AMM$_i$. We conclude that $\alpha_{\text{inf}} = a^C$.

Consider an arbitrary accumulation point α^* of the sequences of ask prices a_i^t for the various AMM$_i$. We must have that $\alpha^* \geq \alpha_{\text{inf}} = a^C$ as we have just shown. But $\alpha^* > a^C$ can be ruled out as, by the same argument as the one used above, we would have that, for some $\bar{t}$ and for all AMM$_i$, it must be that, for all $t > \bar{t}$, $q_i^t(\alpha^{*-}) > q_i^t(\alpha^*)$, implying that α^* is not chosen infinitely many times by any AMM$_i$. Q. E. D. □

At an intuitive level, when the ask on some given day d is $\alpha^{\min}$, any AMM can assess that with a counterfactual ask price slightly lower than $\alpha^{\min}$, they would get $\alpha^{\min} - V^d$ in the case of a sale and zero otherwise. This gives rise to a stochastic payoff, but the conditions on λ_i^d stated in the proposition ensure that, in the limit (as d grows large), one can reason with the induced mean of the reward. That is, using a continuous-time approximation, after some re-writing and after noting that counterfactual payoffs are discounted by the ρ factor, one gets (36). Similarly, for the on-path ask price $\alpha^{\min}$, one gets (37). For $\rho > 0.5$, these expressions ensure that the Q-value of an ask price slightly lower than $\alpha^{\min}$ must grow more quickly than the Q-value of $\alpha^{\min}$ whenever $\alpha^{\min}$ lies strictly above the competitive ask price a^C.[17] Therefore, starting from ask prices above the competitive one, AMMs will eventually drift their choice of ask prices toward the competitive one through a gradual undercutting process. The system will not go beyond the competitive ask price because, at a lower ask price, AMMs would be making losses in expectation. This overall dynamic leads to the competitive pricing that we described in Section 3. Note that the argument works even in the absence of any experimentation (whereas in the classic one-agent problem, Q-learning requires experimentation to obtain the optimal policy - see Watkins and Dayan, 1992). Intuitively, at ask (bid) prices higher (lower) than the competitive equilibrium, the counterfactual updating offers enough information that experimentation is not needed.

[17] The sufficient condition $\rho > 0.5$ comes from the observation that, independently of the number N of AMMs, an AMM could be quoting the best price with only one other AMM; by undercutting, the AMM would *double* their payoff. They would more than double it in other cases.

The logic of our proof can also be applied to the case of one AMM competing against a market maker who uses a fixed ask (higher than the competitive ask price) and a fixed bid (lower than the competitive bid price).

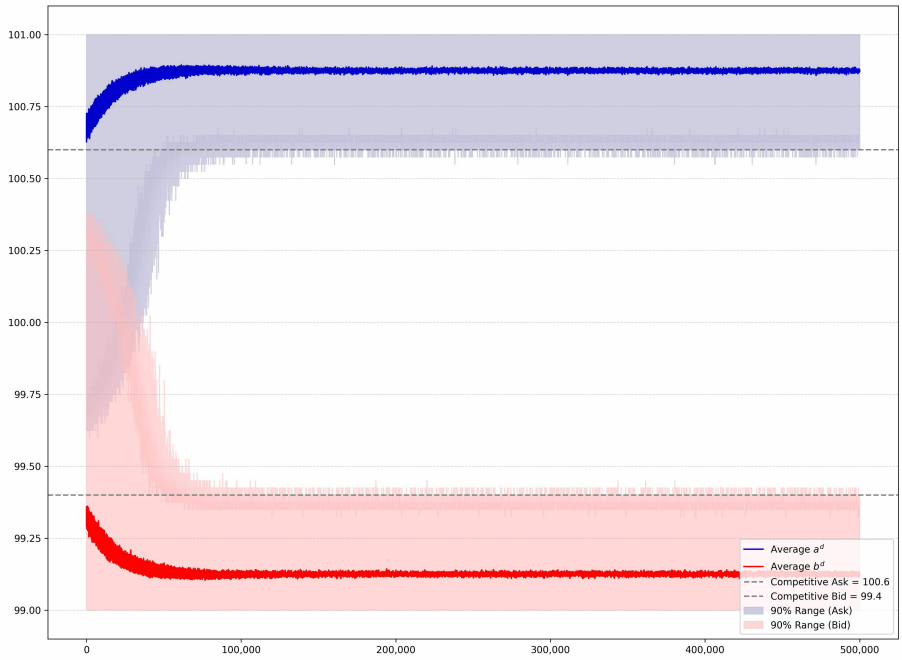
Proposition 4. *Consider the case of one AMM (denoted AMM_1) who competes with a market maker choosing a fixed ask price $\overline{\alpha}$ that is at least two ticks above a^C and a fixed bid price $\overline{\beta}$ that is at least two ticks below b^C in the respective finite grids of ask and bid prices. Suppose that the sequences of λ_1^d satisfy $\sum_d \lambda_1^d = \infty$ and $\sum_d \left(\lambda_1^d\right)^2 < \infty$, and, moreover, that $\varepsilon_1^d = 0$ for all d; assume that $\rho > 0.5$ and suppose that, for AMM_1, for at least one $\alpha_j \geq a^C$ and at least one $\beta_k \leq b^C$, the initial Q-values, $q_1^1(\alpha_j)$ and $\hat{q}_1^1(\beta_k)$, are weakly positive. Then, asymptotically, almost surely, AMM_1 quotes an ask price that is strictly below $\overline{\alpha}$ and a bid price that is strictly above $\overline{\beta}$, and, as a result, AMM_1 does all the trading.*

Proof. For AMM_1, one can obtain the same expressions (36) and (37) in the proof of the previous proposition. From these expressions, note that for any ask α satisfying $a^C < \alpha < \overline{\alpha}$, $q_i^t(\alpha)$ grows more quickly than the Q-value of any other ask price weakly larger than $\overline{\alpha}$, thereby leading, eventually, AMM_1 to choose an ask price strictly lower than $\overline{\alpha}$. Q. E. D. □

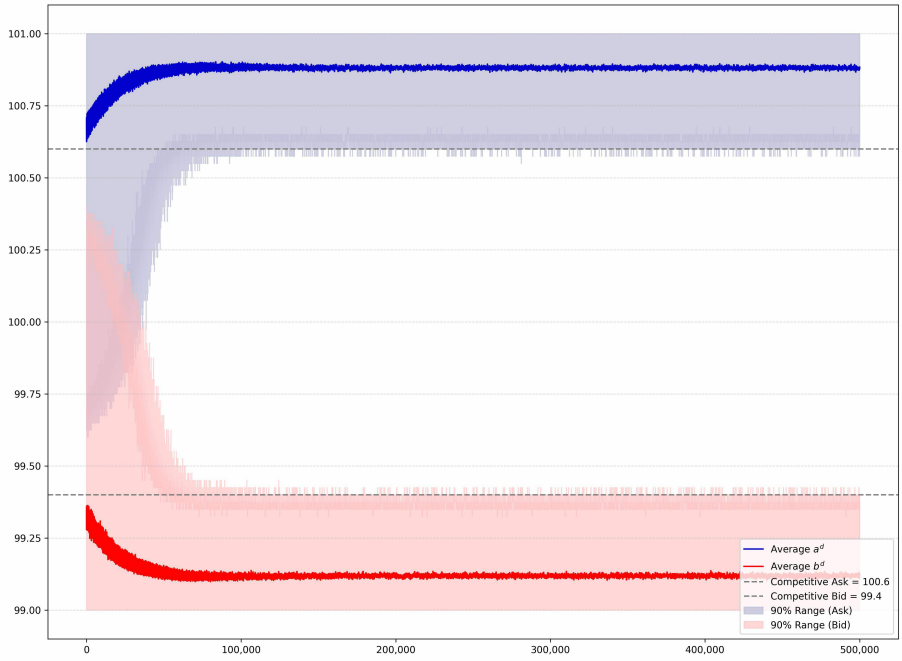
With counterfactual updating, Q-learning produces the result that someone using an algorithmic market maker would like to see: when facing another market maker using a trivial pricing strategy (fixed bid and ask prices), the AMM undercuts the competitor and earns all trading profits. Figure 4 shows the results, with experimentation, for a single AMM competing against a fixed-price market maker.[18]

Given the stark contrast in results between the baseline Q-learning (Propositions 1 and 2 and related simulations) and the counterfactual-updating Q-learning (Propositions 3 and 4 and related simulations), we would like to re-emphasize the mechanisms that lead to them. For exposition, let us focus on the ask prices. Suppose that, at a particular trading time, the ask with the highest Q-value for an AMM is in between the competitive equilibrium ask and the highest value of the asset. In expectation, this ask price is profitable. In the baseline Q-learning with a fixed learning rate, despite the value for this ask being positive in expectation, the corresponding Q-value will eventually become negative and, at that point, the AMM will not choose it again (forever, if there is no experimentation). The reason the Q-value will eventually become negative is that, almost surely, there will be a sufficiently long sequence of bad draws

[18] See Appendix C.4 for additional results and details.



(a) Inelastic Noise Traders



(b) Elastic Noise Traders

The graph plots the mean market quotes over days, averaging across the 1,000 repetitions. The graph refers to the counterfactual-updating Q-learning setup with (a) price-inelastic and (b) price-elastic noise traders, against a dealer setting a fixed ask (101) and a fixed bid (99). The mean ask (bid) is the dark blue (red) line; the blue and red shaded bands represent the 90% range across the simulations for the ask and bid, respectively. The dashed lines plot the competitive ask and bid prices, as discussed in Section 3. In Panel (a), the mean market ask and bid quotes in the final day are 100.87 and 99.12, respectively. In Panel (b), the mean market ask and bid quotes in the final day are 100.88 and 99.13, respectively. The parameter ρ is set equal to 0.6.

Figure 4: Counterfactual-Updating Q-learning AMM against a fixed-price market maker - Mean Market Quotes Over Days

in which the AMM will make negative profits, which will push the Q-value to the negative region. Thus, although profitable in expectation, this ask price cannot be chosen in the long

run. Observe that the logic of this result holds independently of the size of the learning rate, even if, with a smaller learning rate, a longer sequence of bad draws may be needed to make the Q-value cross into the negative region.

Consider now the case of the counterfactual-updating Q-learning. Here, there is a "race" among Q-values, since more than one Q-value is updated every time. In expectation, the Q-values for smaller asks (but higher than the competitive equilibrium one) will grow faster because the AMM choosing them is the one that manages to trade and to earn a profit. This eventually leads to the competitive equilibrium (or, in the individual decision-making case, to an ask below the fixed price), by the standard logic of undercutting. While our formal results are obtained for sequences of learning rates satisfying the Robbins-Monro conditions (which allow for crisper results using stochastic approximation results), qualitatively similar results are obtained in the case of constant but small learning rates.[19] Unlike in baseline Q-learning, even with small, constant learning rates, if the chosen asks remained above a threshold strictly higher than the competitive ask, the Q-value of each ask strictly below that threshold would tend to remain close to its corresponding expected payoff in the long run. This is because the Q-values of those asks continue to be updated using counterfactual payoffs as new market outcomes are observed. Our simulations confirm this and also show that the results we obtain analytically extend to other parameter values and, in particular, to the case of experimentation. In Appendix C.5, we show the results when the learning rates respect the Robbins-Monro conditions and there is experimentation.

9 Choosing an Algorithm

So far, we have presented the results in the case in which prices are set by N baseline Q-learning algorithms or by N Q-learning algorithms with counterfactual updating. In practice, financial firms choose the algorithms with which to price assets. In our setup, they could independently adopt the baseline Q-learning algorithm (from now on, B-AMM), or the more sophisticated Q-learning with counterfactual updating (from now on, CF-AMM). The question we want to answer in this section is which algorithm emerges as the equilibrium choice.

Let us consider the theoretical benchmarks discussed in our propositions, with no experimentation, and, to start with, let us assume that there are only two firms in the market. If

[19] The only qualification is that in this case stochasticity persists in the limit.

both firms choose a B-AMM, they would both enjoy a profit due to supra-competitive, loss-free prices. If both choose the CF-AMM, they would receive zero profits, since the price converges to the competitive equilibrium. On the other hand, if one firm adopts the B-AMM and the other the CF-AMM, the former is completely priced out, whereas the latter obtains the entire market. The reason is that, for a given ask (bid) price set by the CF-AMM, the B-AMM will eventually settle on a strictly larger ask (lower bid) given that these are loss-free. Figure 5 presents this strategic interaction as a normal-form game. It is immediately clear that both firms adopting the CF-AMM is always an equilibrium of the game. The only question is whether this is the only equilibrium, in the case in which the game is similar to a Prisoner's Dilemma, or there exists another equilibrium in pure strategies, in the case the game is akin to a coordination game.[20] This crucially depends on the comparison between the payoffs G and $\frac{\pi_{LF}}{2}$ in the matrix in Figure 5a. Figure 5b illustrates the payoffs of a firm choosing one algorithm in competition with $N-1$ other financial firms, for any $N \geq 2$. The same logic applies, with the observation that when all firms choose the B-AMM, they all receive a fraction $\frac{1}{N}$ of the profits earned with loss-free prices.

	B-AMM	*CF-AMM*
B-AMM	$\frac{\pi_{LF}}{2}, \frac{\pi_{LF}}{2}$	$0, G$
CF-AMM	$G, 0$	$0, 0$

(a) Two AMMs

	All B-AMM	*At least one CF-AMM*
B-AMM	$\frac{\pi_{LF}}{N}$	0
CF-AMM	G	0

(b) N AMMs

Panel (a) represents the competition of two firms adopting a B-AMM or a CF-AMM as a normal-form game. The payoffs refer to the theoretical benchmarks without experimentation, as illustrated in Propositions 1 and 3. If both firms choose a B-AMM, they share the profits π_{LF} earned by the loss-free prices (Proposition 1). If they both choose the CF-AMM, they receive zero profits, as in a competitive equilibrium (Proposition 3). If only one firm chooses the B-AMM, it will receive zero profits, since it will set strictly less competitive prices compared to the other firm, and will not trade. The other firm will capture the entire order flow and receive a positive profit G. Panel (b) shows the payoffs for a firm choosing between a B-AMM and a CF-AMM when competing with any number of other firms. The payoffs depend only on whether all other firms choose the B-AMM or at least one chooses the CF-AMM. We report the payoffs here only for the row player.

Figure 5: Algorithm-Choice Game: Theoretical Benchmark

To get more insights, we simulated the competition between AMMs for different parameter specifications. Figure 6 presents the payoff matrix for these simulations. The payoffs are the average ones obtained asymptotically; that is, the prices are those from the final trading day, averaged across the 1,000 simulations.

[20] In the first case, the game is different from the classical Prisoner's Dilemma only in that one strategy is weakly and not strictly dominated.

	B	CF
B	**0.70, 0.70**	0, 0.41
CF	0.41, 0	**0.02, 0.02**

(a) Inelastic: Baseline

	B	CF
B	0.57, 0.57	0, 0.59
CF	0.59, 0	**0.02, 0.02**

(b) Inelastic: $c_i = 0.05$

	B	CF
B	0.70, 0.70	0, 1.26
CF	1.26, 0	**0.14, 0.14**

(c) Inelastic: Decoupled

	B	CF
B	0.08, 0.08	0, 0.20
CF	0.20, 0	**0.01, 0.01**

(d) Elastic: Baseline

	B	CF
B	0.10, 0.10	0, 0.24
CF	0.24, 0	**0.01, 0.01**

(e) Elastic: $c_i = 0.05$

	B	CF
B	0.08, 0.08	0, 0.23
CF	0.23, 0	**0.04, 0.04**

(f) Elastic: Decoupled

The figure shows the normal-form games of two firms adopting a B-AMM or a CF-AMM for different cases. Payoffs are rounded to two decimals; bold entries are pure-strategy Nash equilibria. The two "Baseline" cases use the parameterizations for price-inelastic and price-elastic noise traders adopted in Section 5. The $c_i = 0.05$ cases differ from them only in that there is a floor on the exploration rate equal to 5 percent. The "Decoupled" cases differ from them only in that the ask-side/bid-side is decoupled, as discussed in Section 4.

Figure 6: Algorithm-Choice Game: Different Cases for $N = 2$ Firms

Figures 6a and 6d refer to the price-inelastic noise traders and price-elastic noise traders cases and parameterizations used in Section 5. In the price-inelastic case, the game has two equilibria in pure strategies: both firms choose either the B-AMM or both the CF-AMM. In the price-elastic case, instead, choosing the CF-AMM is the unique equilibrium. Note that, in this equilibrium, firms obtain a slightly positive profit, simply due to the discreteness of the action space.

The difference between the two cases depends on two factors: the profit $\frac{\pi_{LF}}{2}$ earned by the two B-AMMs, and the profit G that the CF-AMM earns when competing against a B-AMM. In the inelastic case, $\frac{\pi_{LF}}{2}$ is large because the loss-free ask price is very high (although not as high as the monopoly price, as noted in Section 6); the same consideration applies to the bid. At the same time, the CF-AMM is unable to price the asset sufficiently high to obtain at least $\frac{\pi_{LF}}{2}$ by capturing the entire market. In the price-elastic case, the profit $\frac{\pi_{LF}}{2}$ is substantially lower because the loss-free ask and bid prices are further away from the monopoly ones (i.e., at those prices, the probability that a noise trader trades is much lower).

While the change in $\frac{\pi_{LF}}{2}$ as a function of price elasticity is easy to understand, the value of G in the two cases deserves more comments. In the price-inelastic case, if the CF-AMM were able to set an ask (bid) price smaller (larger) than v_H (v_L) by just one tick, it would obtain the entire market and earn almost π_{LF}. In the price-elastic case, it could actually set monopoly ask and bid prices (since the loss-free ask and bid prices are more extreme) and obtain monopoly profits, strictly larger than π_{LF}. This does not happen in our simulations. The CF-AMM operates on the basis of a partial knowledge of the market. Once it settles on, e.g., a particular ask price α and earns positive profits, it may be unable to choose a higher ask. At the same

time, if the B-AMM settles on an ask price larger than α, it may remain there, since this is now loss-free. Clearly, more experimentation can help to avoid such a result. This is what we observe in Figure 6b, where there is a floor of 5 percent on the experimentation rate (Figure 6e is the price-elastic counterpart). In this case, the CF-AMM is able to earn a larger profit, and there is a unique equilibrium. Fundamentally, whether the (B-AMM, B-AMM) equilibrium survives or not depends on how effective the CF-AMM is in choosing prices not too close to the competitive ones, since, loosely speaking, the B-AMM's "best response" is to choose (close to) loss-free prices. As we explained in Section 4, we have implemented a version of the Q-learning in which the Q-values for both bid and ask are updated after each market order: for instance, if an AMM sells at a particular ask price, it updates the Q-value for this ask, and it also updates the Q-value for the posted bid, given that it did not buy at that bid (hence, updating the Q-value with a zero, as with a no-trade). In other words, even when a trader is, e.g., a buyer, the AMMs update the bids too. This pushes the Q-values for the chosen bids towards zero. In the case of a CF-AMM, it pushes the Q-values for all bids lower than the posted one towards zero, by virtue of the updating process. As we discussed in Section 4, a variant consists in updating the Q-values for the asks (bids) only when an AMM sells (buys) or there is no trade. This essentially decouples the update of the Q-values and is based on the idea that one should not change the Q-values of the ask prices when there was an order from a seller. While, as mentioned before, the chosen variant of the Q-learning is not relevant for the results described in the previous sections, it does matter for the current analysis. Figures 6c and 6f present the payoff matrix for the same cases in Figures 6a and 6d, respectively, except that the updating of the Q-tables is decoupled. The CF-AMM is now more effective in setting the prices to undercut the B-AMM, with the result that there is a unique equilibrium.

The previous analysis can easily be extended to larger values of N. Figure 7 presents the results of simulations with $N = 2, 3$, or 5 firms. The only difference with Cases (a) and (d) of Figure 6, for price-inelastic and price-elastic noise traders, respectively, is that the CF-AMM adopts a learning rate $\lambda_i^d = \min\{0.1, \frac{1}{0.0001d}\}$ in agreement with the Robbins-Monro conditions. Similarly to Figure 6, for $N = 2$, there are two equilibria. This remains true for $N = 3$, but not for $N = 5$. While the payoff for a single CF-AMM competing with all B-AMMs remains constant and equal to 0.330 for any N, the payoff from coordinating on the B-AMM is decreasing in N. Given our parameters, $N = 5$ is the smallest number of firms such that the unique equilibrium consists of all firms choosing a CF-AMM. The more general lesson is that, even in cases in which there are two equilibria for $N = 2$, the equilibrium in which all firms choose the B-AMM may not exist for a larger number of competitors.

	B	CF
B	**0.70**	0
CF	0.33	**0.06**

(a) $N = 2$, Inelastic

	All B-AMM	*At least one CF-AMM*
B	**0.47**	0
CF	0.33	$\mathbf{0}^+$

(b) $N = 3$, Inelastic

	All B-AMM	*At least one CF-AMM*
B	0.28	0
CF	0.33	$\mathbf{0}^+$

(c) $N = 5$, Inelastic

	B	CF
B	0.08	0
CF	0.20	**0.05**

(d) $N = 2$, Elastic

	All B-AMM	*At least one CF-AMM*
B	0.06	0
CF	0.20	$\mathbf{0}^+$

(e) $N = 3$, Elastic

	All B-AMM	*At least one CF-AMM*
B	0.03	0
CF	0.20	$\mathbf{0}^+$

(f) $N = 5$, Elastic

The figure shows the payoffs of a firm choosing a B-AMM or a CF-AMM when competing with $N-1$ other firms, for $N \in \{2, 3, 5\}$. The payoffs depend on whether all other $N-1$ firms choose the B-AMM or at least one chooses the CF-AMM. Payoffs are rounded to two decimals; bold entries are pure-strategy Nash equilibria. Entries for the cases in which at least one of the $N-1$ firms adopts the CF updating are reported as 0^+ for conciseness: in the simulations, they are always negligible but not always exactly zero, as they slightly vary with the number of CF-AMMs (e.g., for $N = 3$ and inelastic noise traders, as in panel (b), if there is only one other CF-AMM, the payoff from choosing the CF-AMM is 0.06, whereas if all firms choose the CF-AMM, the payoff is 0.003). In the simulations, the learning rate for a B-AMM is constant (as in Proposition 1) and equal to 0.10; for a CF-AMM the learning rate is equal to $\min\{0.1, \frac{1}{0.0001d}\}$, hence satisfying the Robbins-Monro conditions, as in Proposition 3. The other parameters are identical to those of Section 5. Due to a slower convergence in the case with the λ satisfying the Robbins-Monro conditions and $N = 5$ CF-AMMs, we extend this simulation to $D = 2{,}000{,}000$.

Figure 7: Algorithm-Choice Game: N Firms

In the conclusions of their work, Asker et al. (2024) briefly discuss the choice of algorithms in their oligopoly model (see their Figure 7). Their results are in stark contrast with ours. Indeed, in their case, adopting the "asynchronous algorithm," equivalent to our B-AMM, rather than the "synchronous algorithm" (similar to our CF-AMM), is the dominant strategy.[21] This means that, in their oligopoly setting, supra-competitive prices arise even though firms are competing in algorithms. At a technical level, the difference between their oligopoly model and our financial-market model is that, in their setup, payoffs are deterministic, whereas in ours they are stochastic. This technical difference, however, has substantial conceptual implications.[22] To appreciate them, first note, from their Proposition 1, that their asynchronous algorithm can converge to many rest points. If two firms settle on a profitable price, the algorithm will fail to move to a more profitable one since it does not update the Q-values for other prices. In our case, the dynamics are very different: because the asset value is stochastic, at prices that are not loss free, the B-AMM will eventually receive a large enough number of negative

[21] In their example in Figure 7, this strategy is weakly dominant and any reasonable selection criterion would choose both firms adopting the asynchronous algorithm as the equilibrium; in other examples, it is strictly dominant (as they observe in their Footnote 50).

[22] In the Glosten and Milgrom (1985) model, the stochasticity of the asset values, coupled with the assumption of traders being informed about these random asset values implies adverse selection for market makers. As we discuss, the root of the difference between our results and those of Asker et al. (2024) is the stochasticity itself, and not adverse selection. Indeed, all our results would keep holding in a market with no informed traders.

profits that will induce it to move to a loss-free price. This explains the difference between the many convergence points in Asker et al. (2024) and our loss-free result. Second, by the same argument, note that in their Figure 7, the "synchronous algorithm" (similar to our CF-AMM) earns the same profit (close to the competitive equilibrium one) independently of the competing algorithm.[23] In our case, instead, the CF-AMM captures the entire market when facing a B-AMM and, therefore, earns higher and higher profits the closer the selected prices are to the monopoly prices. To understand this point, suppose that, in the oligopoly model of Asker et al. (2024), the synchronous algorithm chooses a supra-competitive price. If the other, asynchronous, algorithm chooses an even higher price, the resulting profit will be zero. The Q-value for this price will, therefore, be revised downwards, until the algorithm chooses the same price as the synchronous one (or a lower one). Eventually, the algorithms will settle on the competitive price. Intuitively, the synchronous algorithm helps the asynchronous one to learn the existence of other supra-competitive prices and the asynchronous algorithm can lower its price to match the other without risk (given the deterministic payoff), thus generating the same outcome as in Bertrand competition. This does not happen in our case. When the CF-AMM settles on, e.g., a supra-competitive ask price below v_H, the B-AMM will settle on a higher ask, since it is loss-free. No algorithm will select a new price: the B-AMM will not select a lower price because lower asks imply possible losses; the CF-AMM does not change its price because the partial knowledge of the market does not allow it to discover higher and more profitable ask prices. One could consider even more sophisticated algorithms that are more effective in setting prices, but the conclusion of our work is that even the possible choice of an algorithm that uses a very partial understanding of the market and very limited information leads to competitive prices, in stark contrast with the previous literature.

10 Discussion and Conclusions

The baseline Q-learning algorithm presented in Sections 4 and 5 leads to some unsettling results, as discussed in Section 6. In particular, Proposition 1 establishes a loss-free result in the absence of experimentation. We obtain the same result in simulations with an exponentially declining

[23] The synchronous algorithm in Asker et al. (2024) is similar but not identical to our CF-AMM algorithm since it is assumed to know the demand function perfectly and to observe competitors' prices. We only assume that the CF-AMM has a very basic understanding of the market and knows the market quotes. Asker et al. (2024) also briefly examine, through simulations, a variant labeled "synchronous updating using downward demand." This variant requires less knowledge of the market but still differs from ours in that it updates Q-values using bounds on counterfactual payoffs.

exploration rate and a constant learning rate. The main conclusion is that the supra-competitive prices to which the AMMs converge arise from the algorithms' tendency to abandon prices that generate losses. They are not the result of collusion and do not maximize monopoly profits.

In addition to the cases presented in the previous sections, we have simulated the model with many other learning rates and exploration probabilities. Prices nevertheless remain supra-competitive, and we observe convergence to loss-free prices for a wide range of parameters. Persistent experimentation, with an exploration probability bounded away from zero, and smaller updating weights lead to asks that are, on average, below the loss-free threshold, but still far above the competitive ask. While, in many cases, the average market asks (bids) are below (above) the loss-free threshold, we observe that, across the simulations, there is a sizable fraction of the 1,000 simulations in which the algorithms still choose loss-free prices. These changes to the algorithm's parameters lead to greater variation in both price paths and final price levels, but the qualitative insight remains unchanged. We also simulate cases in which the ε-greedy policy is replaced with soft-max (or logit) exploration (see, e.g., Sutton and Barto, 2018), again obtaining supra-competitive prices. Table 2 summarizes these results; Appendix C.10 provides the full analysis and results.

Overall, the very large spread generated by baseline Q-learning—so large that, in some cases, it completely crowds out informed trading—casts doubt on whether this algorithm is representative of those deployed in the financial industry. These doubts are reinforced by Proposition 2, which shows that baseline Q-learning is outplayed even by the simple strategy of setting a fixed ask and a fixed bid. This analysis suggests to us that baseline Q-learning is a poor and inefficient way of pricing assets, rather than a convincing explanation for supra-competitive prices in financial markets.

Financial institutions devote enormous resources to developing their own algorithmic pricing and trading strategies. The baseline Q-learning algorithm uses only data on the payoffs generated by chosen prices. In Section 7, we modify the algorithm to use some basic information: the prevailing market quotes. As we prove analytically in Proposition 3 and show in simulations, providing this publicly available information, together with an elementary understanding of the market—that traders prefer to buy at lower prices and sell at higher prices—produces very different results: under counterfactual-updating Q-learning, bid and ask prices converge to competitive equilibrium levels. Furthermore, as we show in Proposition 4, this algorithm is able to undercut a market maker setting fixed prices.

Despite these results, one might argue that financial firms have an incentive to use algo-

Table 2: Comparative Statics – Final Mean Market Quotes

Parameter	Mean Ask	Mean Bid
Learning Rate		
$\lambda_i^d = 0.001$	102.28	97.71
$\lambda_i^d = 0.01$	101.76	98.24
$\lambda_i^d = 0.05$	102.01	97.99
$\lambda_i^d = 0.1$ (*)	102.00	98.00
Minimum Exploration Probability		
$c_i = 0$ (*)	102.00	98.00
$c_i = 0.01$	101.94	98.06
$c_i = 0.05$	101.77	98.24
$c_i = 0.10$	101.59	98.40
$c_i = 0.20$	101.37	98.65
Exploration Decay Rate		
$\gamma_i = 1 \times 10^{-4}$	102.00	98.00
$\gamma_i = 4 \times 10^{-5}$ (*)	102.00	98.00
$\gamma_i = 2 \times 10^{-5}$	102.00	98.00
$\gamma_i = 1 \times 10^{-5}$	102.00	98.00
$\gamma_i = 5 \times 10^{-6}$	101.95	98.05
Soft-max Exploration		
$\tau = 20$	101.82	98.20

The table reports the mean market ask and bid quotes on the final day ($d = 1{,}000{,}000$), averaged across 1,000 repetitions, for the baseline Q-learning setup. In each row, the indicated parameter is changed relative to the baseline parameterization (Table 1). All other parameters remain unchanged, except that all simulations run for $D = 1{,}000{,}000$ trading days. In the soft-max row, the ε-greedy policy is replaced with logit exploration with parameter τ. The asterisk (*) denotes the baseline parameterization considered in Section 6.

rithms that produce supra-competitive prices, and that these are, therefore, the prices we should expect. However, this conclusion does not follow. The choice of an algorithm is a strategic decision. In Section 9, we show that the algorithm-choice game often has a unique equilibrium in which firms adopt counterfactual-updating Q-learning (and that this outcome is always an equilibrium).

One can think of counterfactual updating as the "economic solution" to the inefficiency of basic Q-learning. The computer science literature has also recognized the limitations of basic Q-learning and proposed ways to improve its performance. A notable contribution in this area is the Deep Q-Network (DQN; Mnih et al., 2015).

Relative to our baseline Q-learning algorithm, DQN introduces two main innovations relevant to our setting: (1) function approximation through a neural network, and (2) experience

replay using random mini-batches drawn from a replay buffer. The neural network replaces the Q-table: rather than maintaining a separate table entry for each action, the algorithm uses a parameterized function to approximate Q-values. The parameters are updated by gradient descent to reduce the discrepancy between predicted Q-values and observed payoffs.[24] Experience replay, in turn, stores past observations in a buffer. Rather than learning only from the most recent observation, the algorithm updates Q-values using a random sample (a "batch") of stored observations. This allows each observation to be used more than once and reduces the influence of individual, recent payoff realizations. Larger batches average over more observations, making each update less sensitive to a single sampled payoff.

Experience replay is particularly relevant in our setting. By learning from multiple realized payoffs associated with the same price, rather than from a single realization, it makes the estimated value of that price (i.e., its Q-value) reflect its average profitability more closely. In theory, experience replay by itself does not change the loss-free result stated in Proposition 1. It does not eliminate the possibility that a sequence of unfavorable observations pushes the Q-values of loss-generating prices below those of loss-free alternatives.[25] Averaging over past observations, however, coupled with experimentation may help the process stay away from the loss-free region. By contrast, the network structure appears more important in problems with a richer state space than in our stateless setting. Our simulations support these intuitions. In the interest of space, we provide details of the DQN implementation in Appendix D and report here only the main findings for the case of price-inelastic noise traders.

Table 3 reports the average final ask and bid prices under our baseline market parameterization for the case of price-inelastic noise traders. Each row corresponds to a combination of batch size b and buffer size B. The first pair of price columns reports results for tabular Q-learning with experience replay; the second pair reports results for the full-fledged DQN. We consider batch sizes $b = 32, 64, 128, 256$, typically used in the literature, and a substantially larger batch of $b = 1024$. We also consider different buffer sizes. A larger buffer spreads the sample over a longer history and can reduce the correlation between sampled payoffs, but it may include older and less relevant observations.[26]

[24] Another feature of DQN is the target network. This has no role in our setting because the learning target is simply the realized payoff, with no continuation-value term.

[25] A sufficiently long unfavorable sequence can push the Q-value below zero while ensuring that no positive-payoff observations for that ask remain in the buffer, regardless of its initial contents.

[26] Our environment is not stationary because an AMM's payoff also depends on the choices of the other AMMs. Payoffs earned early in the interaction may, therefore, be less representative of a price's current profitability.

Table 3: Final ask and bid prices by batch and buffer size

		Q-learning + Experience Replay		DQN	
B	b	Ask	Bid	Ask	Bid
1	1	102.00	98.00	101.84	98.10
500	1	101.98	98.02	101.84	97.99
	32	101.64	98.39	101.45	98.53
1,000	32	101.58	98.43	101.39	98.65
	128	101.39	98.63	101.34	98.63
5,000	128	101.18	98.84	101.07	98.95
50,000	128	100.81	99.21	100.88	99.09
	256	100.85	99.15	100.88	99.12
	1024	100.83	99.18	100.84	99.16
500,000	64	100.83	99.18	100.83	99.18
	128	100.83	99.17	100.82	99.18
	256	100.82	99.18	100.81	99.18
	1024	100.80	99.20	100.81	99.19

The table reports the average ask and bid prices across 1,000 simulations on the final day ($d = 500{,}000$). B is the buffer size and b is the batch size. The case $b = B = 1$ is the specification without experience replay, so the tabular Q-learning results in that row correspond to the baseline in Section 6. When $B = 500{,}000$, the buffer holds the full history. Tabular Q-learning uses $\lambda = 0.1$; DQN uses stochastic gradient descent with a learning rate of 0.01. Otherwise, the tabular Q-learning setup uses the parameterization described in Section 5, and the DQN setup is described in Appendix D.

Replacing the Q-value tables with a neural network without experience replay does not, by itself, change our results: the DQN converges towards the loss-free prices of Section 6, with a final average ask (bid) price of 101.84 (98.10) and around 40% of simulations ending with the algorithms quoting loss-free prices (see Appendix D.3). Experience replay, by contrast, moves prices away from the loss-free benchmark. For large values of B and b, DQN brings prices closer to the competitive equilibrium (recall that the competitive equilibrium ask is equal to 100.6 and the bid is equal to 99.4) than baseline Q-learning does, although not as close as counterfactual updating does (average ask equal to 100.64 and average bid equal to 99.38). The same holds when we keep the Q-value tables and only add experience replay: at matched buffer and batch sizes, tabular Q-learning with experience replay and the DQN settle at similar prices, approaching but never reaching the competitive level. It seems, therefore, that experience replay, rather than the network architecture, drives the departure from loss-free pricing.

In conclusion, our results caution against treating the outcomes of the baseline Q-learning algorithm as a general prediction for algorithmic market making. In our setting, supra-competitive

prices arise from the algorithm's tendency to avoid losses, not from collusion. Improving how algorithms use information produces markedly different outcomes, while competition in algorithm choice often makes counterfactual-updating Q-learning the unique equilibrium. The central issue is, therefore, not simply whether prices are set by algorithms, but how those algorithms learn and which designs firms have an incentive to adopt.

We deliberately use a simple, stateless setup to make these mechanisms transparent. This is important both for understanding market outcomes and for assessing their regulatory and policy implications. Algorithmic market makers may use much richer information and condition their choices on variables such as inventories (see, e.g., Ganesh et al., 2019 and the papers reviewed in Bai et al., 2024). Nevertheless, we believe that the mechanisms identified here are useful for understanding more complex environments.

A natural extension would replace our one-trader-per-day setup with sequences of trades, allowing quotes to depend on the evolving trading history. The resulting growth in the number of Q-values would create computational challenges, making approaches that update several values at once potentially useful. One possibility is to use the already mentioned DQN (Mnih et al., 2015). Another possibility is to use Coarse Q-learning, which groups alternatives into similarity classes, as explored in decision problems by Jehiel and Satpathy (2025).

References

Asker, J., Fershtman, C., & Pakes, A. (2024). The Impact of Artificial Intelligence Design on Pricing. *Journal of Economics & Management Strategy*, *33*(2), 276–304.

Bai, Y., Gao, Y., Wan, R., Zhang, S., & Song, R. (2024, November). A Review of Reinforcement Learning in Financial Applications [arXiv:2411.12746 [q-fin]].

Benaïm, M. (1999). Dynamics of stochastic approximation algorithms. In J. Azéma, M. Émery, M. Ledoux, & M. Yor (Eds.), *Séminaire de Probabilités XXXIII* (pp. 1–68). Springer.

Calvano, E., Calzolari, G., Denicolò, V., & Pastorello, S. (2020). Artificial Intelligence, Algorithmic Pricing, and Collusion. *American Economic Review*, *110*(10), 3267–3297.

Camerer, C., & Ho, T.-H. (1999). Experience-weighted Attraction Learning in Normal Form Games. *Econometrica*, *67*(4), 827–874.

Cartea, Á., Chang, P., & Penalva, J. (2022, May). Algorithmic Collusion in Electronic Markets: The Impact of Tick Size.

Cipriani, M., & Guarino, A. (2008). Herd Behavior and Contagion in Financial Markets. *The B.E. Journal of Theoretical Economics*, *8*(1).

Cipriani, M., & Guarino, A. (2014). Estimating a Structural Model of Herd Behavior in Financial Markets. *The American Economic Review*, *104*(1), 224–251.

Colliard, J.-E., Foucault, T., & Lovo, S. (2025). Algorithmic Pricing and Liquidity in Securities Markets.

Dou, W. W., Goldstein, I., & Ji, Y. (2025, July). AI-Powered Trading, Algorithmic Collusion, and Price Efficiency.

Easley, D., Kiefer, N. M., & O'Hara, M. (1997). One Day in the Life of a Very Common Stock. *The Review of Financial Studies*, *10*(3), 805–835.

Easley, D., López de Prado, M. M., & O'Hara, M. (2012). Flow Toxicity and Liquidity in a High-frequency World. *The Review of Financial Studies*, *25*(5), 1457–1493.

Ganesh, S., Vadori, N., Xu, M., Zheng, H., Reddy, P., & Veloso, M. (2019). Multi-Agent Simulation for Pricing and Hedging in a Dealer Market.

Glosten, L. R. (2009). Information and learning in markets: The impact of market microstructure. *Journal of Economic Literature*, *47*(3), 802–804.

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*(1), 71–100.

Glosten, L. R., & Putnins, T. J. (2016). Welfare costs of informed trade. *Working Paper*, 16–58.

Jehiel, P., & Satpathy, A. (2025). Learning to be Indifferent in Complex Decisions: A Coarse Payoff-Assessment Model.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529–533.

Mookherjee, D., & Sopher, B. (1997). Learning and Decision Costs in Experimental Constant Sum Games. *Games and Economic Behavior*, *19*(1), 97–132.

Sarin, R., & Vahid, F. (1999). Payoff Assessments without Probabilities: A Simple Dynamic Model of Choice. *Games and Economic Behavior*, *28*(2), 294–309.

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. MIT Press.

Watkins, C. J. C. H. (1989). *Learning from Delayed Rewards* [Doctoral dissertation].

Watkins, C. J. C. H., & Dayan, P. (1992). Q-learning. *Machine Learning*, *8*(3), 279–292.

A Counterfactual Updating - Bid-side Updating

Here, we repeat the explanation of the counterfactual updating for the bid side.

1. **AMM$_i$ quotes the best bid $b_i^d = \beta_j = b^d$ and buys the asset.** The AMM$_i$ updates the Q-values for β_j on the basis of the *realized* payoff and the Q-values for all the bids greater than β_j for the payoffs that they would have received by buying at those prices. AMM$_i$ does not update the Q-values for any bid less than β_j since they do not know whether they would have been able to buy. Formally, the Q-values are updated as follows:

$$\hat{q}_i^{d+1}(\beta_j) = \lambda_i^d(v^d - \beta_j) + (1 - \lambda_i^d)\hat{q}_i^d(\beta_j) \tag{38}$$

$$\hat{q}_i^{d+1}(\beta_k) = \lambda_i^d\rho(v^d - \beta_k) + (1 - \lambda_i^d\rho)\hat{q}_i^d(\beta_k), \qquad \text{for } \beta_k > \beta_j \tag{39}$$

$$\hat{q}_i^{d+1}(\beta_l) = \hat{q}_i^d(\beta_l), \qquad \text{for } \beta_l < \beta_j \tag{40}$$

2. **AMM$_i$ quotes the bid $b_i^d = \beta_j = b^d$ and another AMM buys the asset.** Since the AMM is aware of their own bid price and of the market bid at which the transaction occurred, they update the Q-values for all the bids as follows: the chosen bid price is updated using a zero profit and weight λ_i^d; for any bid strictly less than β_j, the Q-value is updated with a payoff of zero, using the counterfactual updating weight $\lambda_i^d\rho$; for all prices above β_j, the AMM would have set the highest price and bought the asset, and so, the Q-values are updated using the counterfactual profit.

$$\hat{q}_i^{d+1}(\beta_j) = (1 - \lambda_i^d)\hat{q}_i^d(\beta_j), \tag{41}$$

$$\hat{q}_i^{d+1}(\beta_l) = (1 - \lambda_i^d\rho)\hat{q}_i^d(\beta_l), \qquad \text{for all } \beta_l < \beta_j, \tag{42}$$

$$\hat{q}_i^{d+1}(\beta_k) = \lambda_i^d\rho(v^d - \beta_k) + (1 - \lambda_i^d\rho)\hat{q}_i^d(\beta_k), \qquad \text{for all } \beta_k > \beta_j. \tag{43}$$

3. **AMM$_i$ quotes the bid $b_i^d < \beta_j = b^d$ and another AMM buys the asset.** AMM$_i$ updates the Q-value associated with b_i^d using a profit of zero and a weight of λ_i^d; all bids below β_j, excluding b_i^d, are updated using a zero profit but with the counterfactual weight $\lambda_i^d\rho$. For all bids strictly above β_j, AMM$_i$ would have quoted a higher price and bought the asset, so we update the Q-values associated with these prices using the counterfactual payoffs. In updating β_j, we follow the same convention as described in Section 7 for the ask prices. That is, the counterfactual payoffs are discounted by $\frac{1}{M}$, where M is chosen so that the Q-values are updated as conservatively as possible (that is, with the smallest

change in the Q-value).

$$\hat{q}_i^{d+1}(b_i^d) = (1-\lambda_i^d)\hat{q}_i^d(b_i^d), \tag{44}$$

$$\hat{q}_i^{d+1}(\beta_j) = \lambda_i^d \rho \frac{1}{M}(v^d - \beta_j) + (1-\lambda_i^d\rho)\hat{q}_i^d(\beta_j), \tag{45}$$

$$\hat{q}_i^{d+1}(\beta_l) = (1-\lambda_i^d\rho)\hat{q}_i^d(\beta_l), \qquad \text{for all } \beta_l < \beta_j, \beta_l \neq b_i^d, \tag{46}$$

$$\hat{q}_i^{d+1}(\beta_k) = \lambda_i^d\rho(v^d - \beta_k) + (1-\lambda_i^d\rho)\hat{q}_i^d(\beta_k), \qquad \text{for all } \beta_k > \beta_j. \tag{47}$$

4. **AMM$_i$ quotes the bid $b_i^d = \beta_j = b^d$ and there is no sell.** Note that this can occur because there is no trade or the trader buys. AMM$_i$ updates the Q-values for all prices that are smaller than or equal to β_j since the payoff would have been zero, using weight λ_i^d for the chosen b_i^d and $\lambda_i^d\rho$ for the counterfactual prices. AMM$_i$ does not update the Q-values for higher bids, since they do not know whether setting a higher bid would have led to a buy.

$$\hat{q}_i^{d+1}(\beta_l) = (1-\lambda_i^d\rho)\hat{q}_i^d(\beta_l), \qquad \text{for } \beta_l < \beta_j, \tag{48}$$

$$\hat{q}_i^{d+1}(\beta_j) = (1-\lambda_i^d)\hat{q}_i^d(\beta_j), \tag{49}$$

$$\hat{q}_i^{d+1}(\beta_k) = \hat{q}_i^d(\beta_k), \qquad \text{for } \beta_k > \beta_j. \tag{50}$$

5. **AMM$_i$ quotes $b_i^d < \beta_j = b^d$ and there is no sell.** Again, this can occur because there is no trade or the trader buys. All prices smaller than or equal to β_j are updated with a zero payoff using weight $\lambda_i^d\rho$, except for b_i^d which is updated with weight λ_i^d. They do not know whether they would have bought by setting a price higher than β_j and so these Q-values are not updated.

$$\hat{q}_i^{d+1}(\beta_l) = (1-\lambda_i^d\rho)\hat{q}_i^d(\beta_l), \qquad \text{for } \beta_l \leq \beta_j, \beta_l \neq b_i^d, \tag{51}$$

$$\hat{q}_i^{d+1}(b_i^d) = (1-\lambda_i^d)\hat{q}_i^d(b_i^d), \tag{52}$$

$$\hat{q}_i^{d+1}(\beta_k) = \hat{q}_i^d(\beta_k), \qquad \text{for } \beta_k > \beta_j. \tag{53}$$

B Nash Equilibrium Pricing for $N = 2$ Market Makers

We now present the discrete Nash-equilibrium pricing outcomes with $N = 2$ market makers, given the setup described in Section 5. We present the payoff matrices for price-inelastic noise traders (with $\mu = 0.3$) in Tables B1 and B2, and for price-elastic noise traders (with $\mu \approx 0.24$) in Tables B3 and B4. Each table reports only a subset of the full price grid, namely the five prices closest to the competitive benchmark. Rows correspond to the quotes of market maker 1 and columns to the quotes of market maker 2, and each cell reports the expected payoffs. The blue cells highlight the Nash equilibria of the game.

Table B1: Expected Payoffs for Ask Quotes – Price-Inelastic Noise Traders

	100.575	**100.600**	**100.625**	**100.650**	**100.675**
100.575	$-0.0063, -0.0063$	$-0.0125, 0.0000$	$-0.0125, 0.0000$	$-0.0125, 0.0000$	$-0.0125, 0.0000$
100.600	$0.0000, -0.0125$	$\mathbf{0.0000, 0.0000}$	$0.0000, 0.0000$	$0.0000, 0.0000$	$0.0000, 0.0000$
100.625	$0.0000, -0.0125$	$0.0000, 0.0000$	$\mathbf{0.0062, 0.0062}$	$0.0125, 0.0000$	$0.0125, 0.0000$
100.650	$0.0000, -0.0125$	$0.0000, 0.0000$	$0.0000, 0.0125$	$\mathbf{0.0125, 0.0125}$	$0.0250, 0.0000$
100.675	$0.0000, -0.0125$	$0.0000, 0.0000$	$0.0000, 0.0125$	$0.0000, 0.0250$	$0.0187, 0.0187$

Table B2: Expected Payoffs for Bid Quotes – Price-Inelastic Noise Traders

	99.425	**99.400**	**99.375**	**99.350**	**99.325**
99.425	$-0.0063, -0.0063$	$-0.0125, 0.0000$	$-0.0125, 0.0000$	$-0.0125, 0.0000$	$-0.0125, 0.0000$
99.400	$0.0000, -0.0125$	$\mathbf{0.0000, 0.0000}$	$0.0000, 0.0000$	$0.0000, 0.0000$	$0.0000, 0.0000$
99.375	$0.0000, -0.0125$	$0.0000, 0.0000$	$\mathbf{0.0062, 0.0062}$	$0.0125, 0.0000$	$0.0125, 0.0000$
99.350	$0.0000, -0.0125$	$0.0000, 0.0000$	$0.0000, 0.0125$	$\mathbf{0.0125, 0.0125}$	$0.0250, 0.0000$
99.325	$0.0000, -0.0125$	$0.0000, 0.0000$	$0.0000, 0.0125$	$0.0000, 0.0250$	$0.0187, 0.0187$

Table B3: Expected Payoffs for Ask Quotes – Price-Elastic Noise Traders

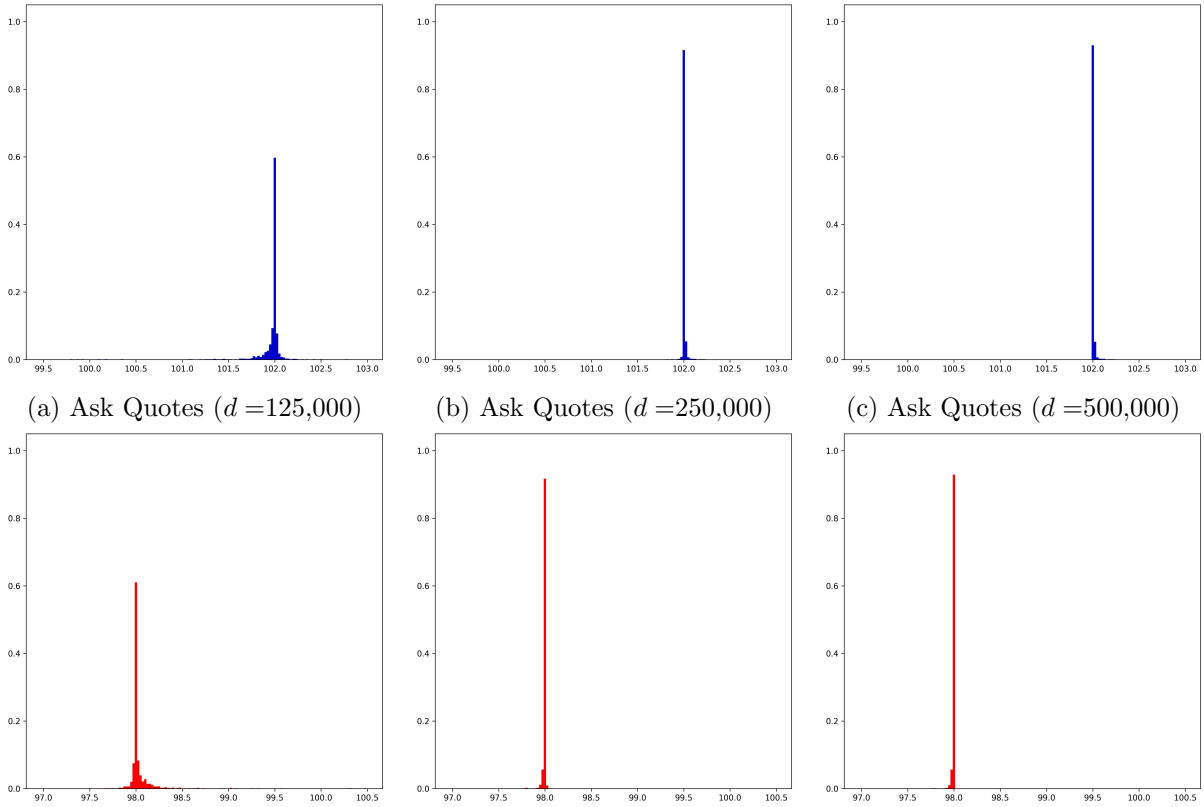
	100.575	**100.600**	**100.625**	**100.650**	**100.675**
100.575	$-0.0038, -0.0038$	$-0.0075, 0.0000$	$-0.0075, 0.0000$	$-0.0075, 0.0000$	$-0.0075, 0.0000$
100.600	$0.0000, -0.0075$	$\mathbf{0.0000, 0.0000}$	$0.0000, 0.0000$	$0.0000, 0.0000$	$0.0000, 0.0000$
100.625	$0.0000, -0.0075$	$0.0000, 0.0000$	$\mathbf{0.0037, 0.0037}$	$0.0073, 0.0000$	$0.0073, 0.0000$
100.650	$0.0000, -0.0075$	$0.0000, 0.0000$	$0.0000, 0.0073$	$0.0072, 0.0072$	$0.0144, 0.0000$
100.675	$0.0000, -0.0075$	$0.0000, 0.0000$	$0.0000, 0.0073$	$0.0000, 0.0144$	$0.0107, 0.0107$

Table B4: Expected Payoffs for Bid Quotes – Price-Elastic Noise Traders

	99.425	**99.400**	**99.375**	**99.350**	**99.325**
99.425	$-0.0038, -0.0038$	$-0.0075, 0.0000$	$-0.0075, 0.0000$	$-0.0075, 0.0000$	$-0.0075, 0.0000$
99.400	$0.0000, -0.0075$	$\mathbf{0.0000, 0.0000}$	$0.0000, 0.0000$	$0.0000, 0.0000$	$0.0000, 0.0000$
99.375	$0.0000, -0.0075$	$0.0000, 0.0000$	$\mathbf{0.0037, 0.0037}$	$0.0073, 0.0000$	$0.0073, 0.0000$
99.350	$0.0000, -0.0075$	$0.0000, 0.0000$	$0.0000, 0.0073$	$0.0072, 0.0072$	$0.0144, 0.0000$
99.325	$0.0000, -0.0075$	$0.0000, 0.0000$	$0.0000, 0.0073$	$0.0000, 0.0144$	$0.0107, 0.0107$

C Additional Results

C.1 Additional Figures for Baseline Algorithm with $N = 2$ AMMs

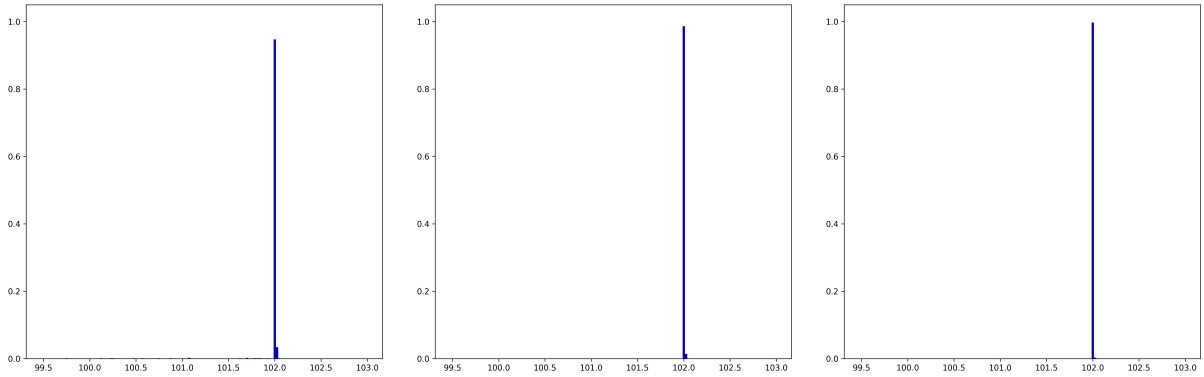


(a) Ask Quotes (d =125,000) (b) Ask Quotes (d =250,000) (c) Ask Quotes (d =500,000)

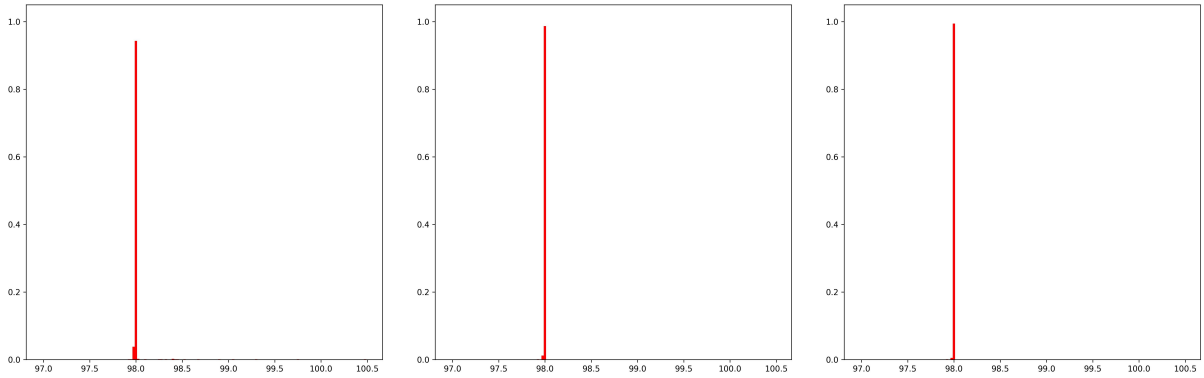
(d) Bid Quotes (d =125,000) (e) Bid Quotes (d =250,000) (f) Bid Quotes (d =500,000)

The graph plots the distributions of market ask and bid prices across the 1,000 repetitions at different points in the simulation: days 125,000, 250,000, and 500,000. The graph refers to the baseline Q-learning setup with price-inelastic noise traders.

Figure C1: Baseline (Price-Inelastic Noise Traders) – Distributions of Market Ask and Bid Quotes over Time



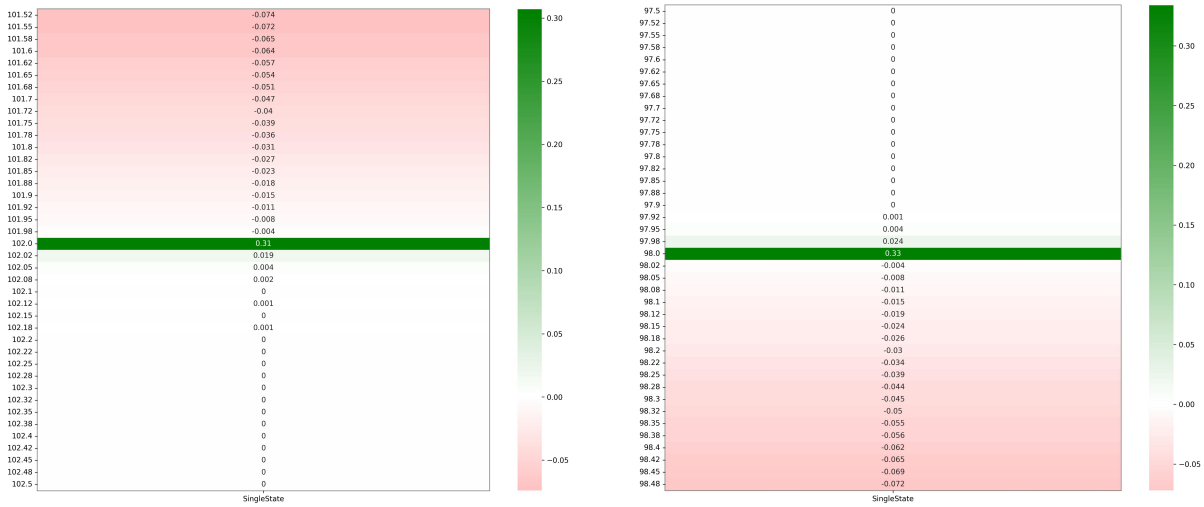
(a) Ask Quotes (d =125,000) (b) Ask Quotes (d =250,000) (c) Ask Quotes (d =500,000)



(d) Bid Quotes (d =125,000) (e) Bid Quotes (d =250,000) (f) Bid Quotes (d =500,000)

The graph plots the distributions of market ask and bid prices across the 1,000 repetitions at different points in the simulation: days 125,000, 250,000, and 500,000. The graph refers to the baseline Q-learning setup with price-elastic noise traders.

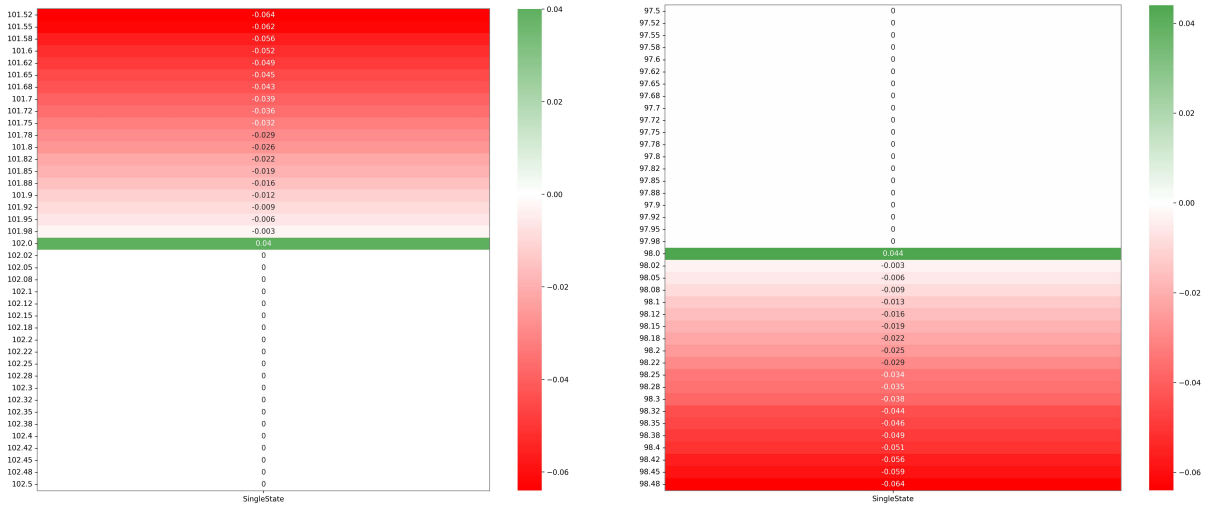
Figure C2: Baseline (Price-Elastic Noise Traders) – Distributions of Market Ask and Bid Quotes over Time



(a) Ask Q-Table (b) Bid Q-Table

The graph plots the mean ask and bid Q-tables at the end of the simulations, averaging across the 1,000 repetitions. It shows the Q-tables for only a subset of prices, centered around the final mean prices. The graph refers to the baseline Q-learning setup with price-inelastic noise traders.

Figure C3: Baseline (Price-Inelastic Noise Traders) - Final Mean Q-Tables

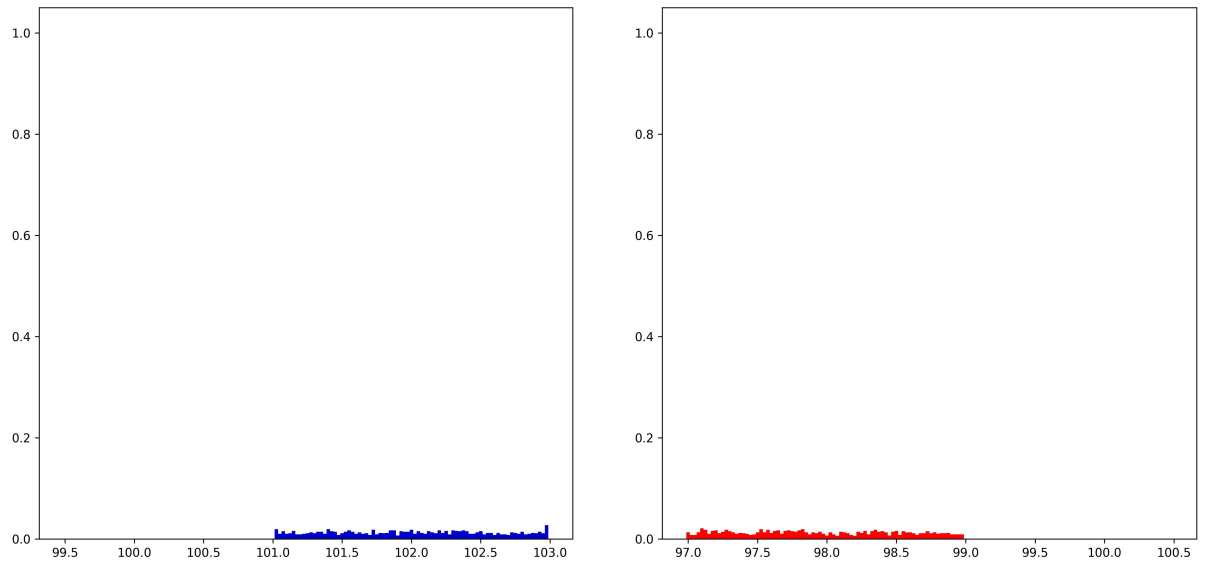


(a) Ask Q-Table (b) Bid Q-Table

The graph plots the mean ask and bid Q-tables at the end of the simulations, averaging across the 1,000 repetitions. It shows the Q-tables for only a subset of prices, centered around the final mean prices. The graph refers to the baseline Q-learning setup with price-elastic noise traders.

Figure C4: Baseline (Price-Elastic Noise Traders) - Final Mean Q-Tables

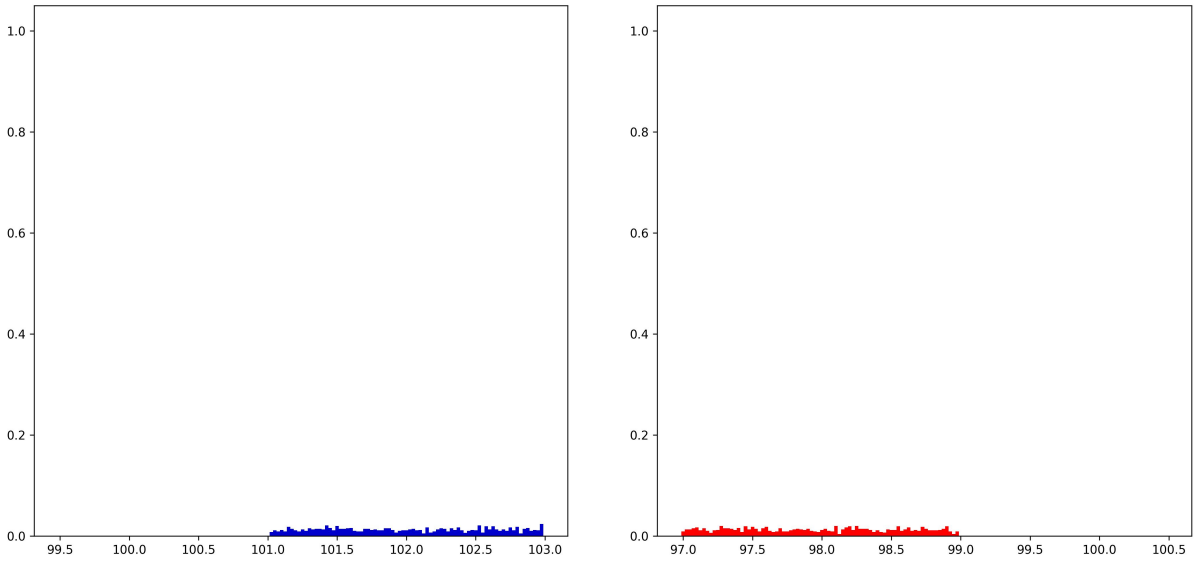
C.2 Additional Figures for Baseline Algorithm vs. Fixed-Price Dealer



(a) Ask (b) Bid

The graph shows the distribution of the ask and bid quotes set by the AMM in the final day, across the 1,000 repetitions. The graph refers to the baseline Q-learning setup with price-inelastic noise traders.

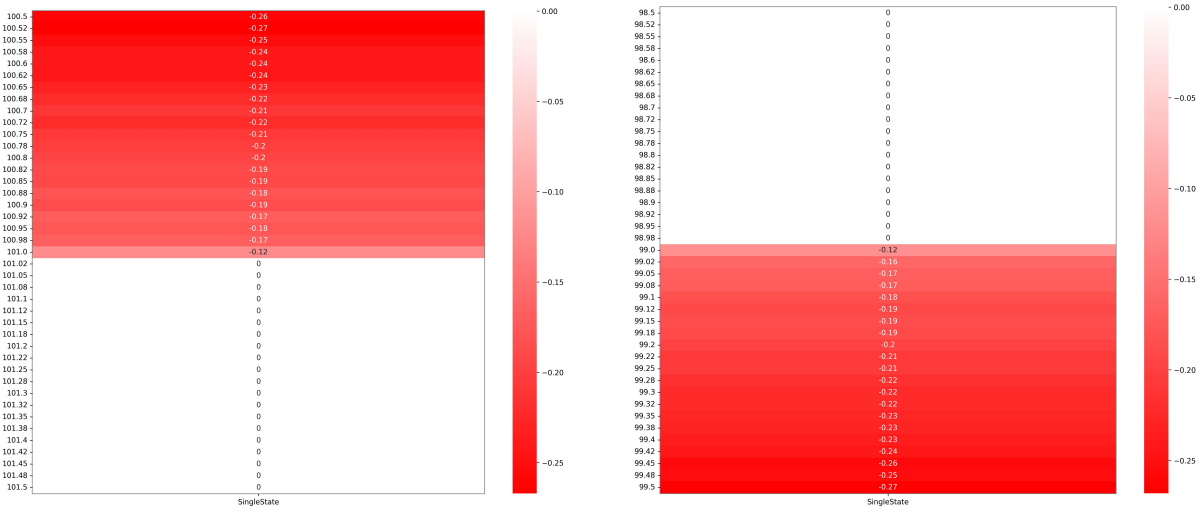
Figure C5: Fixed-price Market Maker (Price-Inelastic Noise Traders) - Distribution of AMM Quotes in $d = D = 500,000$



(a) Ask (b) Bid

The graph shows the distribution of the ask and bid quotes set by the AMM in the final day, across the 1,000 repetitions. The graph refers to the baseline Q-learning setup with price-elastic noise traders.

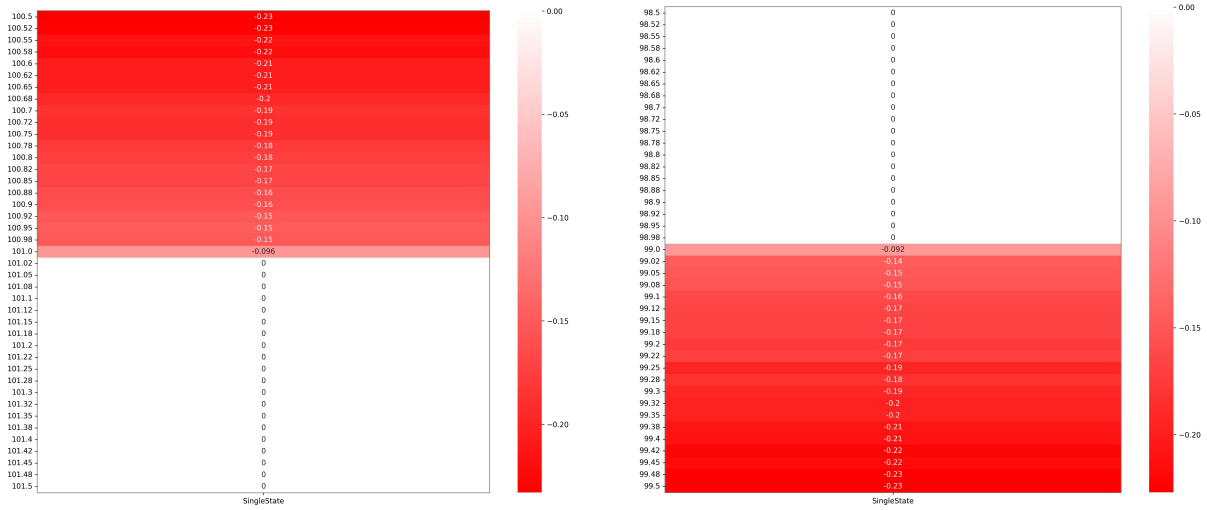
Figure C6: Fixed-price Market Maker (Price-Elastic Noise Traders) - Distribution of AMM Quotes in $d = D = 500,000$



(a) Ask Q-Table (b) Bid Q-Table

The graph plots the mean ask and bid Q-tables at the end of the simulations, averaging across the 1,000 repetitions. It shows the Q-tables for only a subset of prices, centered around the final mean prices. The graph refers to the baseline Q-learning setup with price-inelastic noise traders.

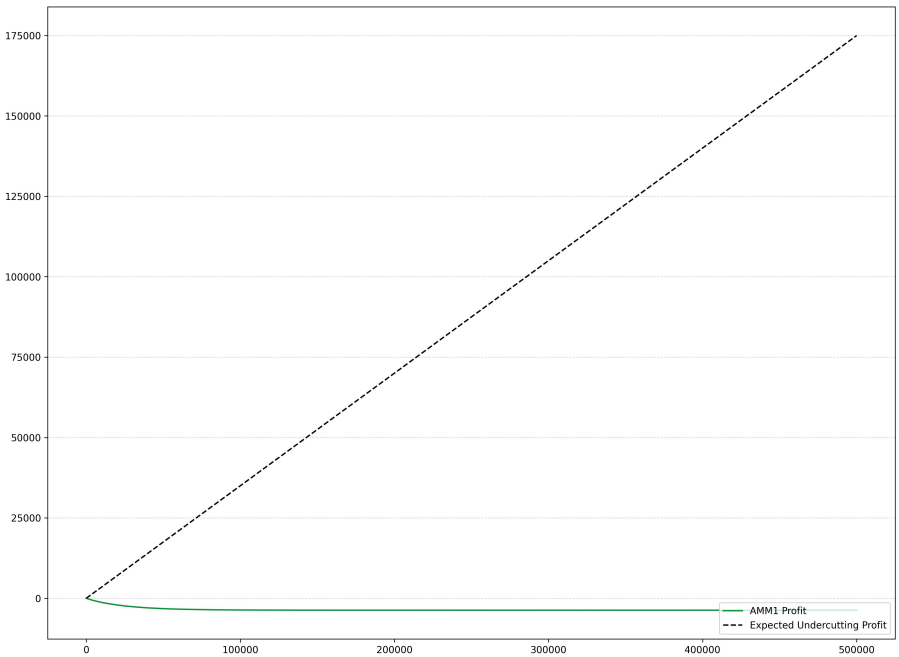
Figure C7: Fixed-price Market Maker (Price-Inelastic Noise Traders) - Final Mean Q-Tables



(a) Ask Q-Table (b) Bid Q-Table

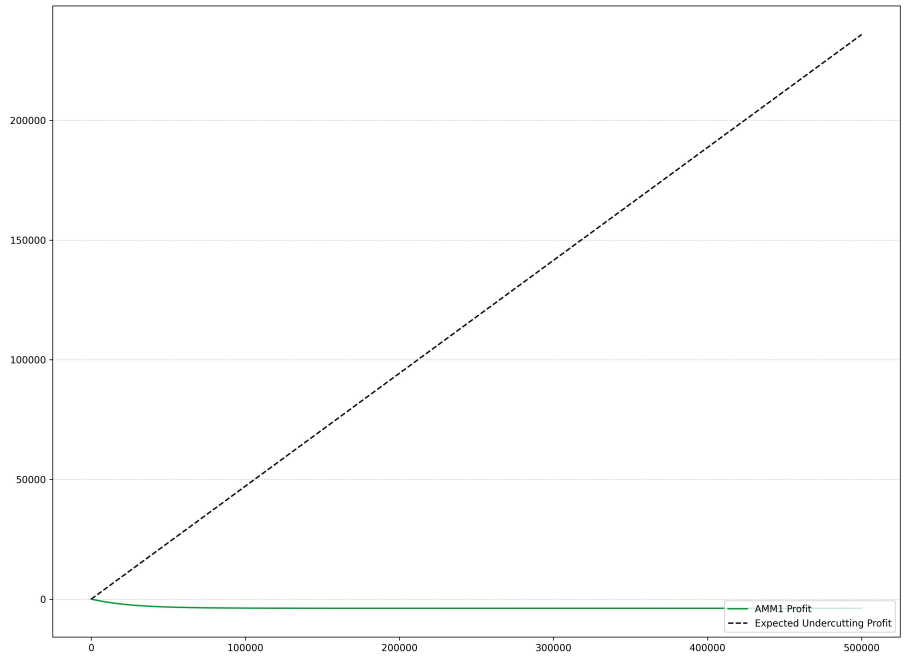
The graph plots the mean ask and bid Q-tables at the end of the simulations, averaging across the 1,000 repetitions. It shows the Q-tables for only a subset of prices, centered around the final mean prices. The graph refers to the baseline Q-learning setup with price-elastic noise traders.

Figure C8: Fixed-price Market Maker (Price-Elastic Noise Traders) - Final Mean Q-Tables



The graph shows the mean of the cumulative profits across the 1,000 repetitions. The graph refers to the baseline Q-learning setup with price-inelastic noise traders. The green line plots the cumulative profit between day 1 and day d, where the daily profit is computed as in Equation 3. The dashed line represents the expected cumulative profit from undercutting the fixed price market maker by one tick, i.e., continually setting an ask of 100.975 and a bid of 99.025.

Figure C9: Fixed-price Market Maker (Price-Inelastic Noise Traders) - Realized vs Undercutting Cumulative Profits

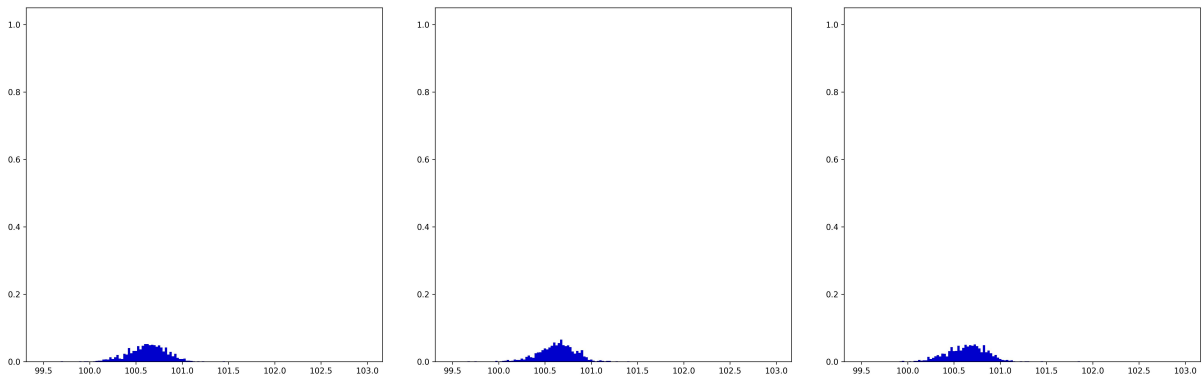


The graph shows the mean of the cumulative profits across the 1,000 repetitions. The graph refers to the baseline Q-learning setup with price-elastic noise traders. The green line plots the cumulative profit between day 1 and day d, where the daily profit is computed as in Equation 3. The dashed line represents the expected cumulative profit from undercutting the fixed price market maker by one tick, i.e., continually setting an ask of 100.95 and a bid of 99.05.

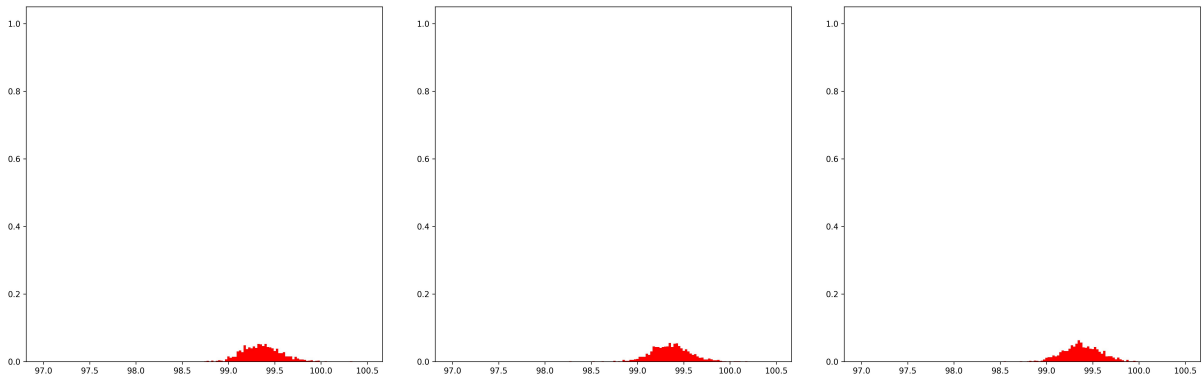
Figure C10: Fixed-price Market Maker (Price-Elastic Noise Traders) - Realized vs Undercutting Cumulative Profits

C.3 Additional Figures for Counterfactual-Updating Algorithm with $N = 2$ AMMs

In this subsection, we present the additional results relating to the counterfactual-updating algorithm, using the parameters corresponding to those used for Figure 3 in Section 8.



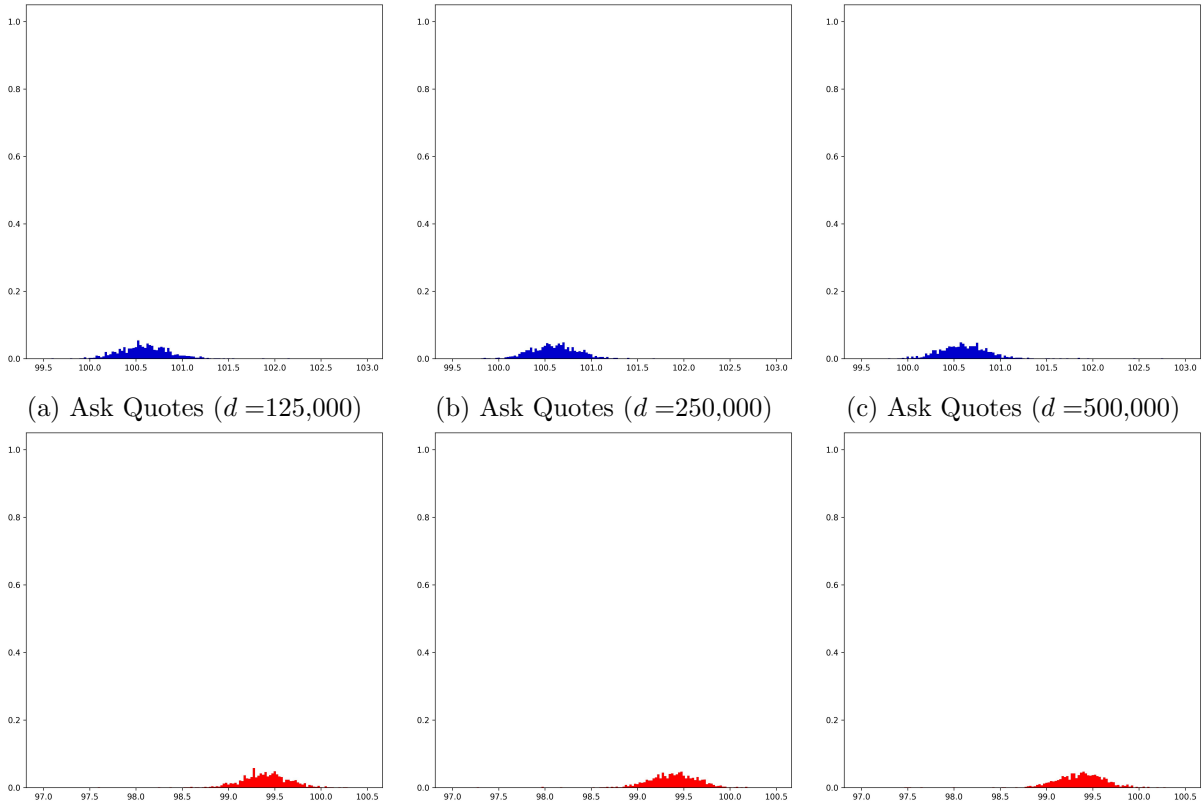
(a) Ask Quotes (d =125,000) (b) Ask Quotes (d =250,000) (c) Ask Quotes (d =500,000)



(d) Bid Quotes (d =125,000) (e) Bid Quotes (d =250,000) (f) Bid Quotes (d =500,000)

The graph plots the distributions of market ask and bid prices across the 1,000 repetitions at different points in the simulation: days 125,000, 250,000, and 500,000. The graph refers to the counterfactual-updating Q-learning setup with price-inelastic noise traders.

Figure C11: Counterfactual Updating (Price-Inelastic Noise Traders) – Distributions of Market Ask and Bid Quotes over Time



(a) Ask Quotes (d =125,000) (b) Ask Quotes (d =250,000) (c) Ask Quotes (d =500,000)

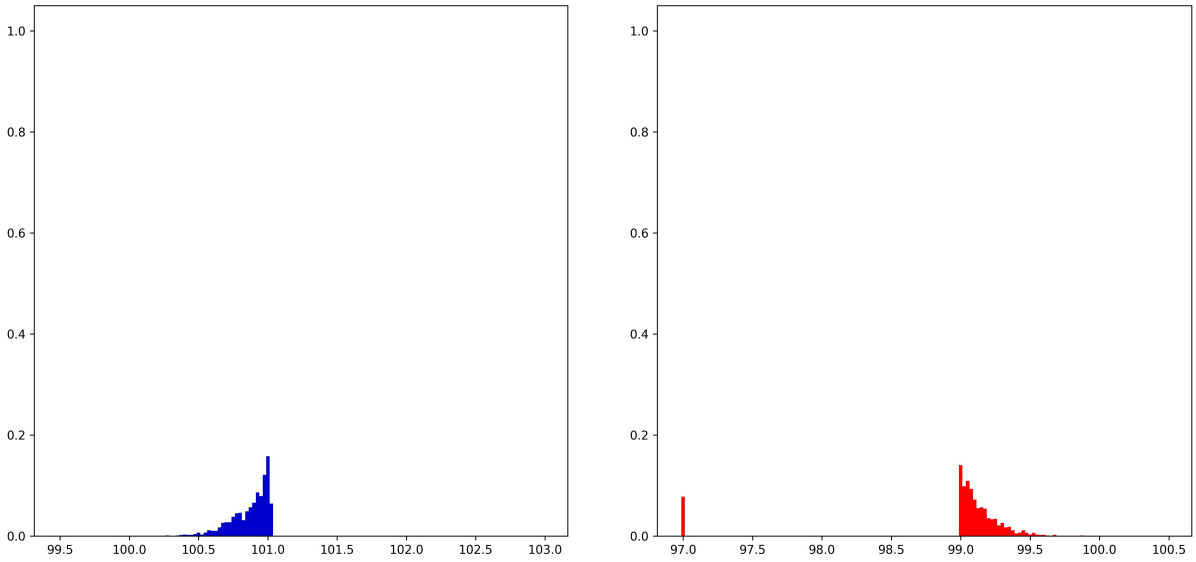
(d) Bid Quotes (d =125,000) (e) Bid Quotes (d =250,000) (f) Bid Quotes (d =500,000)

The graph plots the distributions of market ask and bid prices across the 1,000 repetitions at different points in the simulation: days 125,000, 250,000, and 500,000. The graph refers to the counterfactual-updating Q-learning setup with price-elastic noise traders.

Figure C12: Counterfactual Updating (Price-Elastic Noise Traders) – Distributions of Market Ask and Bid Quotes over Time

C.4 Additional Figures for Counterfactual-Updating Algorithm vs. Fixed-Price Dealer

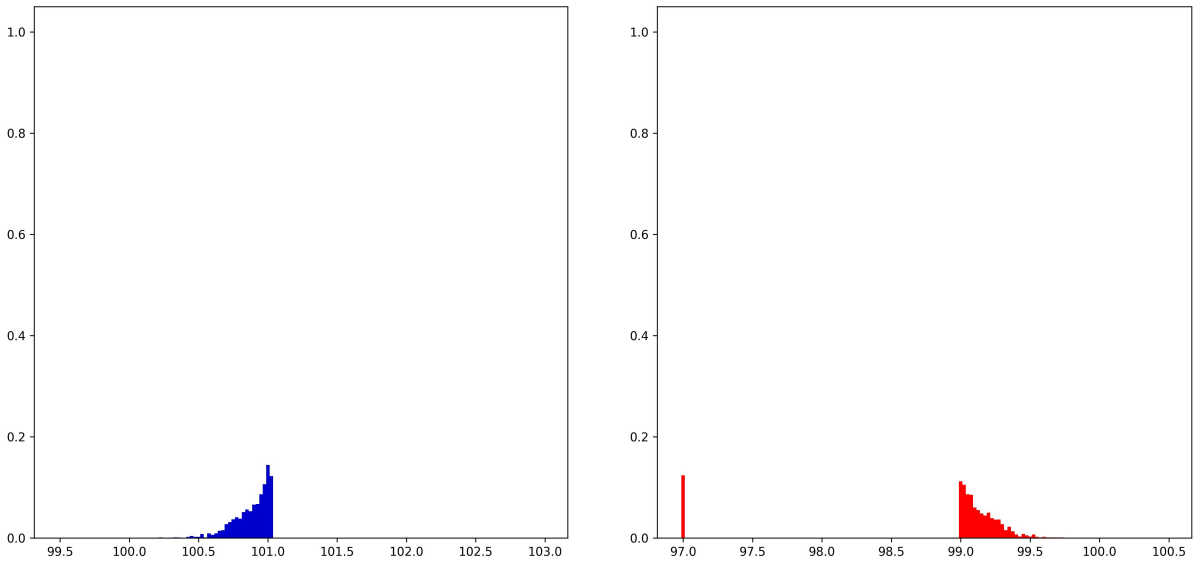
In this subsection, we consider how a single AMM, now using counterfactual updating, responds to a "human" market maker quoting fixed ask and bid prices of 101 and 99, respectively.



(a) Ask (b) Bid

The graph shows the distribution of the ask and bid quotes set by the AMM in the final day, across the 1,000 repetitions. The graph refers to the counterfactual-updating Q-learning setup with price-inelastic noise traders.

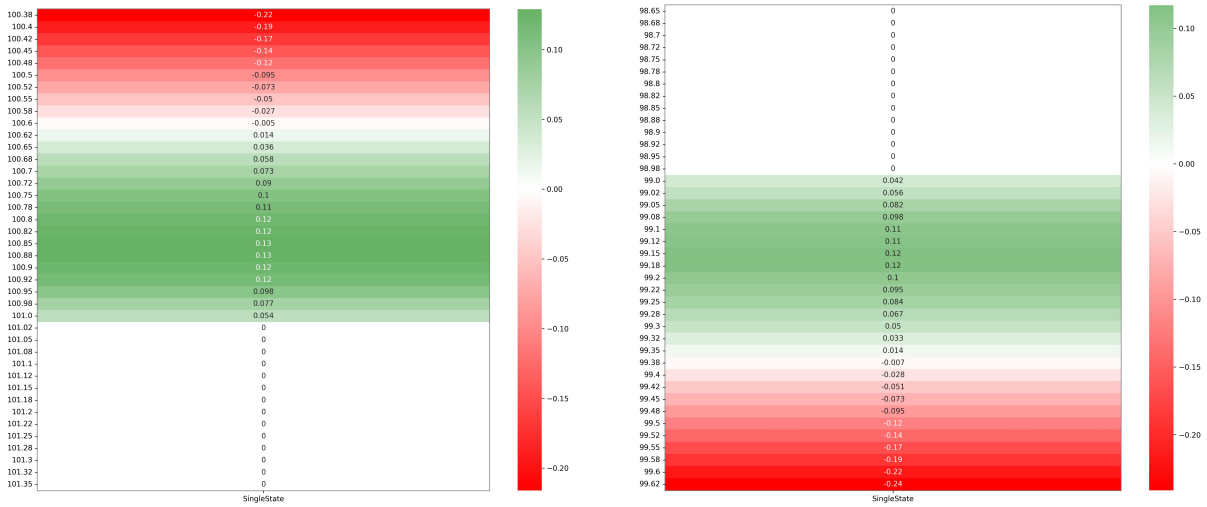
Figure C13: Counterfactual Updating, Fixed-price Market Maker (Price-Inelastic Noise Traders) - Distribution of AMM Quotes in $d = D = 500,000$



(a) Ask (b) Bid

The graph shows the distribution of the ask and bid quotes set by the AMM in the final day, across the 1,000 repetitions. The graph refers to the counterfactual-updating Q-learning setup with price-elastic noise traders.

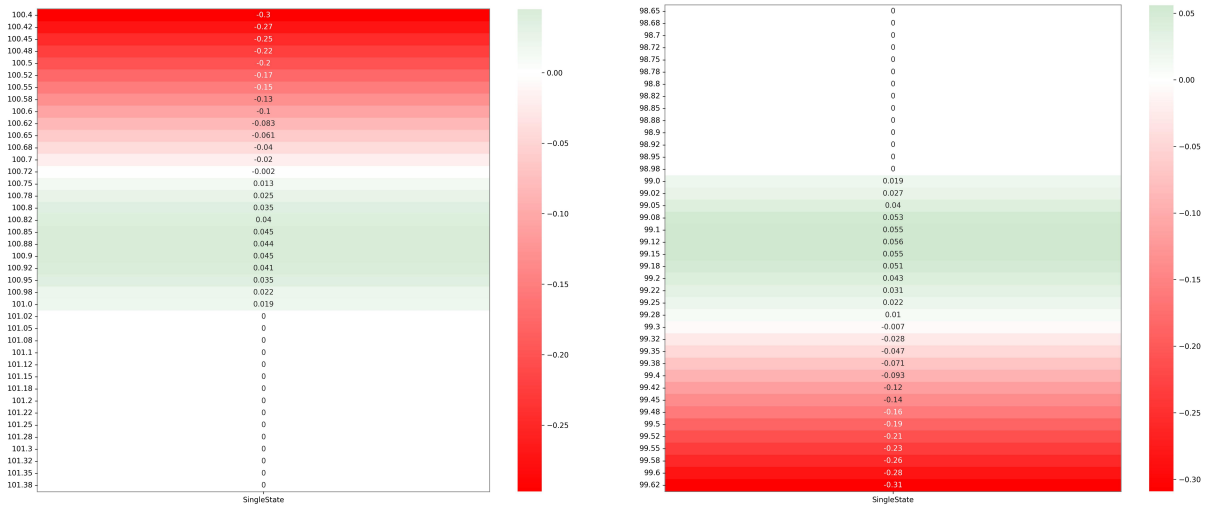
Figure C14: Counterfactual Updating, Fixed-price Market Maker (Price-Elastic Noise Traders) - Distribution of AMM Quotes in $d = D = 500,000$



(a) Ask Q-Table (b) Bid Q-Table

The graph plots the mean ask and bid Q-tables at the end of the simulations, averaging across the 1,000 repetitions. It shows the Q-tables for only a subset of prices, centered around the final mean prices. The graph refers to the counterfactual-updating Q-learning setup with price-inelastic noise traders.

Figure C15: Counterfactual Updating, Fixed-price Market Maker (Price-Inelastic Noise Traders) - Final Mean Q-Tables



(a) Ask Q-Table (b) Bid Q-Table

The graph plots the mean ask and bid Q-tables at the end of the simulations, averaging across the 1,000 repetitions. It shows the Q-tables for only a subset of prices, centered around the final mean prices. The graph refers to the counterfactual-updating Q-learning setup with price-elastic noise traders.

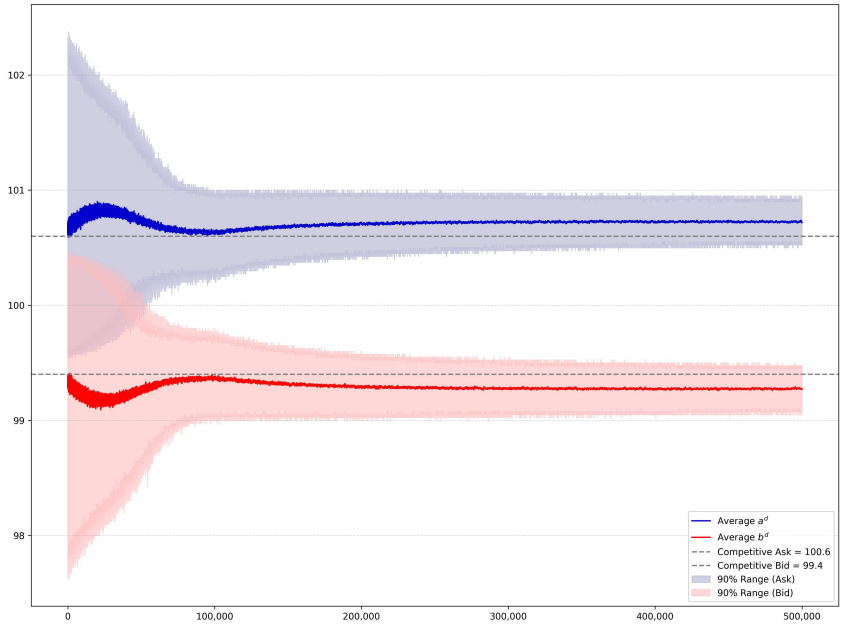
Figure C16: Counterfactual Updating, Fixed-price Market Maker (Price-Elastic Noise Traders) - Final Mean Q-Tables

C.5 Counterfactual-Updating Algorithm with Robbins-Monro Weights

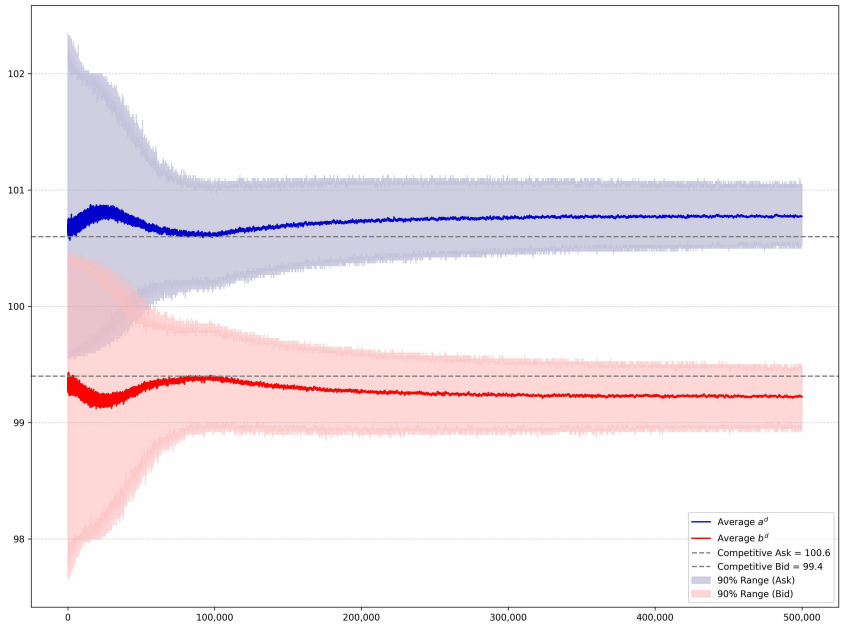
In this subsection, we repeat the simulations in Section 8 but we replace the constant learning rate with one that satisfies the Robbins-Monro conditions. Specifically, we set

$$\lambda_i^d = \min\left\{\frac{1}{0.0001d}, 0.1\right\}. \tag{54}$$

Appendix Figure C17 shows the average market ask and bid quotes over days using counterfactual updating with Robbins-Monro learning rates, for price-inelastic and price-elastic noise traders.



(a) Inelastic Noise Traders



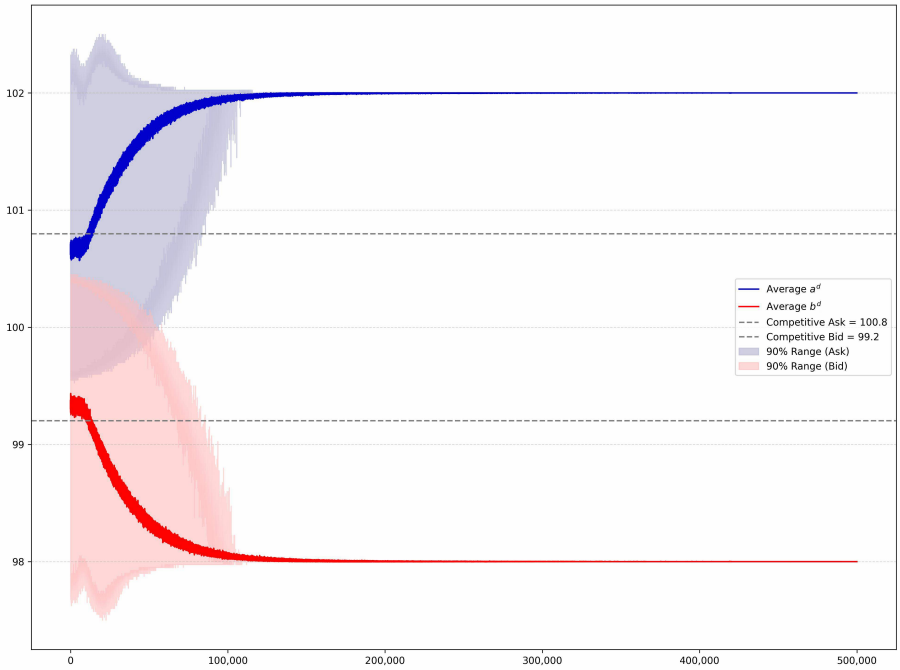
(b) Elastic Noise Traders

The graph plots the mean market quotes over days, averaging across the 1,000 repetitions. The graph refers to the counterfactual-updating Q-learning setup with (a) price-inelastic and (b) price-elastic noise traders. The mean ask (bid) is the dark blue (red) line; the blue and red shaded bands represent the 90% range across the simulations for the ask and bid, respectively. The dashed lines plot the competitive ask and bid prices, as discussed in Section 3. In Panel (a), the mean market ask and bid quotes in the final day are 100.72 and 99.28, respectively. In Panel (b), the mean market ask and bid quotes in the final day are 100.77 and 99.22, respectively. The form of λ_i^d is as given in expression 54.

Figure C17: Counterfactual Updating with Robbins-Monro Updating Weights - Mean Market Quotes Over Days

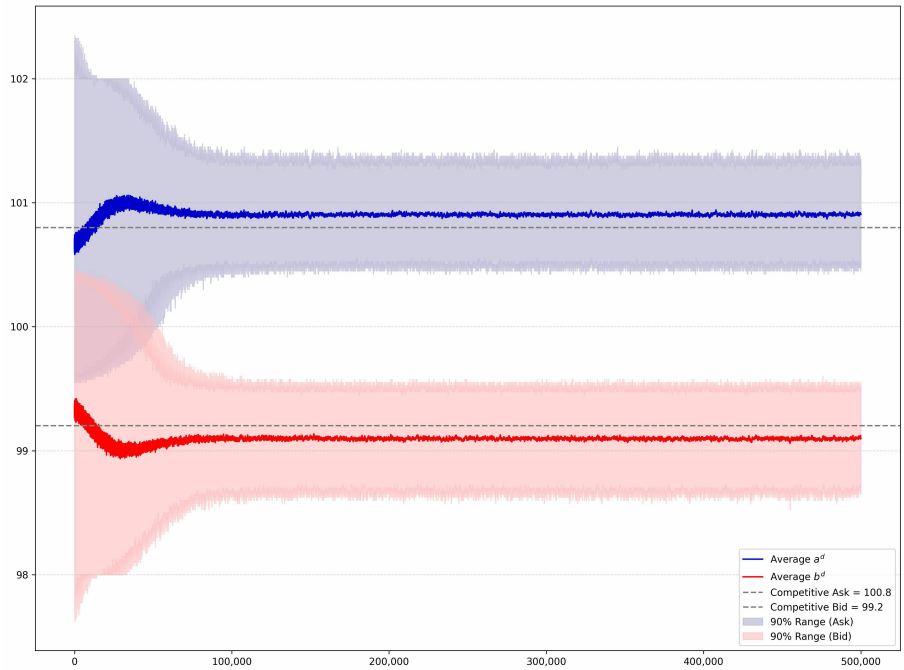
C.6 Price-Elastic Noise Traders with $\mu = 0.3$

As discussed in Section 5, our price-elastic baseline sets $\mu \approx 0.239$ so as to match the competitive equilibrium prices of the price-inelastic case. For completeness, we repeat here our main simulations under price-elastic noise traders with $\mu = 0.3$, for the baseline and counterfactual-updating algorithms, both with two competing AMMs and against a fixed-price dealer; results are qualitatively unchanged (Appendix Figures C18–C21).



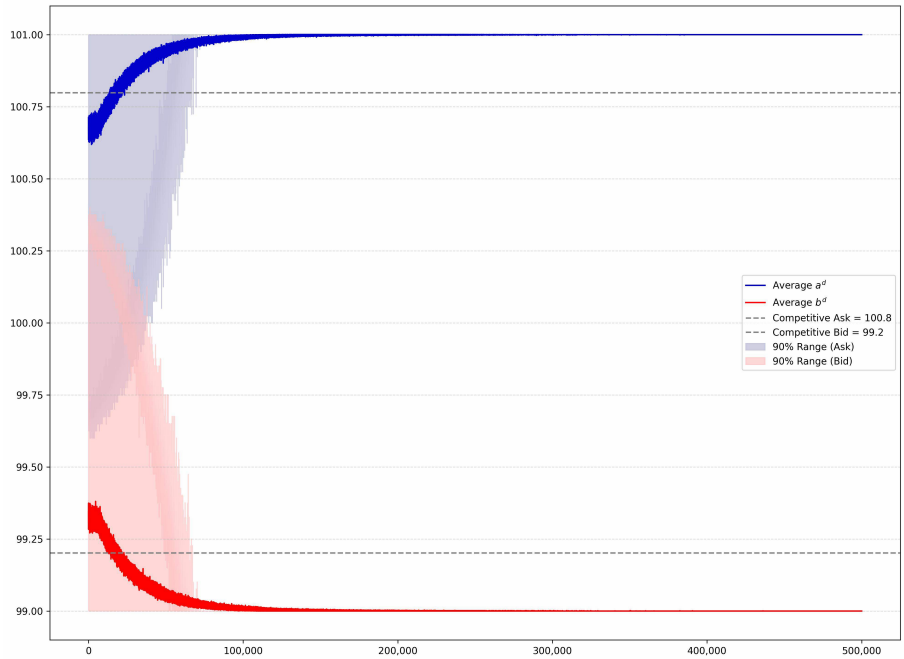
The graph plots the mean market quotes over days, averaging across the 1,000 repetitions. The graph refers to the baseline Q-learning setup with price-elastic noise traders. The mean ask (bid) is the dark blue (red) line; the blue and red shaded bands represent the 90% range across the simulations for the ask and bid, respectively. The dashed lines plot the competitive ask and bid prices, as discussed in Section 3. The mean market ask and bid quotes in the final day are 102.0 and 98.0, respectively.

Figure C18: Baseline (Price-Elastic Noise Traders, $\mu = 0.3$) – Mean Market Ask and Bid Quotes over Time



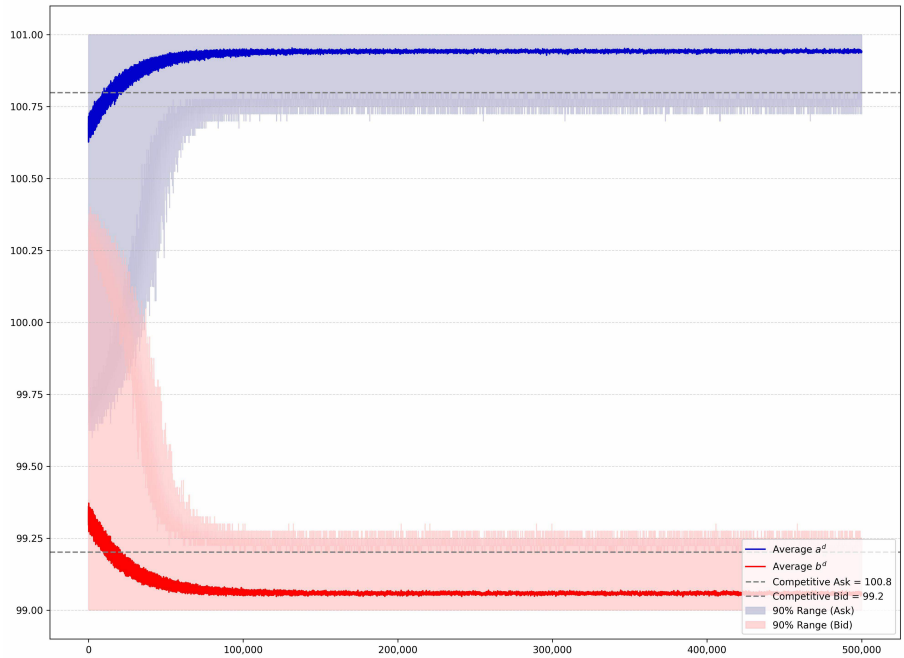
The graph plots the mean market quotes over days, averaging across the 1,000 repetitions. The graph refers to the counterfactual-updating Q-learning setup with price-elastic noise traders. The mean ask (bid) is the dark blue (red) line; the blue and red shaded bands represent the 90% range across the simulations for the ask and bid, respectively. The dashed lines plot the competitive ask and bid prices, as discussed in Section 3. The mean market ask and bid quotes in the final day are 100.91 and 99.1, respectively.

Figure C19: Counterfactual Updating (Price-Elastic Noise Traders, $\mu = 0.3$) – Mean Market Ask and Bid Quotes over Time



The graph plots the mean market quotes over days, averaging across the 1,000 repetitions. The graph refers to the baseline Q-learning setup competing against a fixed-price market maker, with price-elastic noise traders. The mean ask (bid) is the dark blue (red) line; the blue and red shaded bands represent the 90% range across the simulations for the ask and bid, respectively. The dashed lines plot the competitive ask and bid prices, as discussed in Section 3. The mean market ask and bid quotes in the final day are 101.0 and 99.0, respectively.

Figure C20: Baseline vs Fixed-price Market Maker (Price-Elastic Noise Traders, $\mu = 0.3$) – Mean Market Ask and Bid Quotes over Time

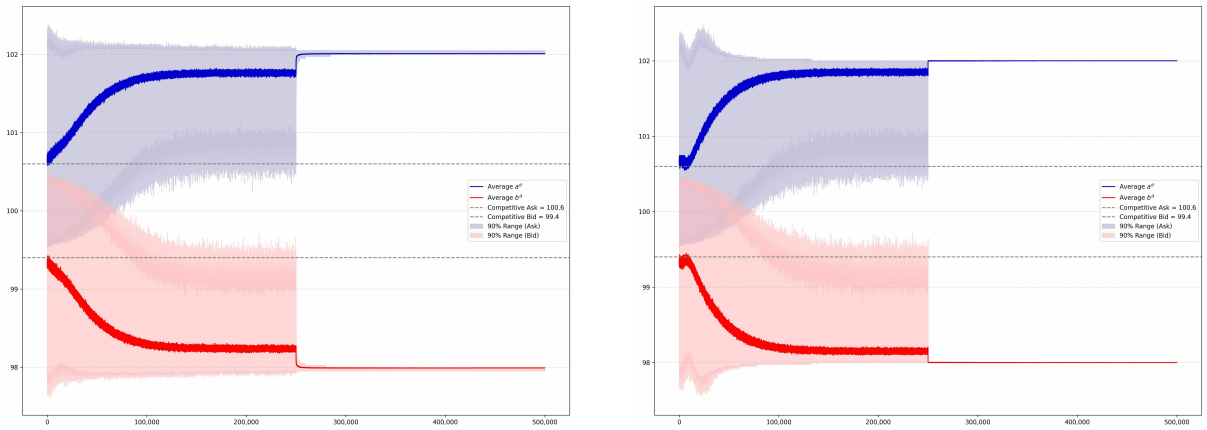


The graph plots the mean market quotes over days, averaging across the 1,000 repetitions. The graph refers to the counterfactual-updating Q-learning setup competing against a fixed-price market maker, with price-elastic noise traders. The mean ask (bid) is the dark blue (red) line; the blue and red shaded bands represent the 90% range across the simulations for the ask and bid, respectively. The dashed lines plot the competitive ask and bid prices, as discussed in Section 3. The mean market ask and bid quotes in the final day are 100.94 and 99.06, respectively.

Figure C21: Counterfactual Updating vs Fixed-price Market Maker (Price-Elastic Noise Traders, $\mu = 0.3$) – Mean Market Ask and Bid Quotes over Time

C.7 Baseline Algorithm – Temporary Minimum Exploration Floors

We now consider the case in which $c_i = 0.05$ for $d \leq \tilde{d}$ and zero afterwards. We present the results for $\tilde{d} = 250{,}000$, for both price-inelastic and price-elastic noise traders. Figure C22 plots the mean market ask and bid quotes over time under this exploration floor, with Panel (a) (Appendix Figure C22a) reporting the case of price-inelastic noise traders and Panel (b) (Appendix Figure C22b) the case of price-elastic noise traders.



(a) Price-inelastic noise traders

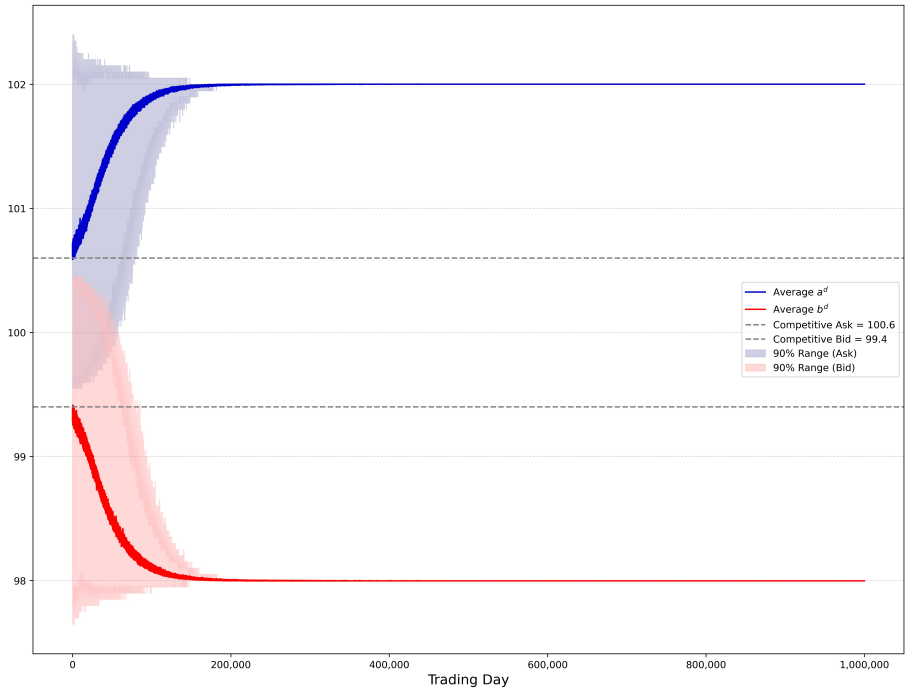
(b) Price-elastic noise traders

The graphs plot the mean market ask and bid quotes over days, averaging across the 1,000 repetitions. The mean ask (bid) is the dark blue (red) line; the blue and red shaded bands represent the 90% range across the simulations for the ask and bid, respectively. The dashed lines plot the competitive ask and bid prices. The graphs refer to the baseline Q-learning setup with an exploration floor of 0.05 for 250,000 days; that is, $c_i = 0.05$ if $d \leq 250{,}000$ and $c_i = 0$ otherwise. Panel (a) refers to price-inelastic noise traders, panel (b) to price-elastic noise traders. In Panel (a), the mean market ask and bid quotes in the final day are 102.01 and 97.99, respectively. In Panel (b), the mean market ask and bid quotes in the final day are 102.00 and 98.00, respectively.

Figure C22: Baseline (Exploration Floor – 250,000 Days) – Mean Market Ask and Bid Quotes over Time

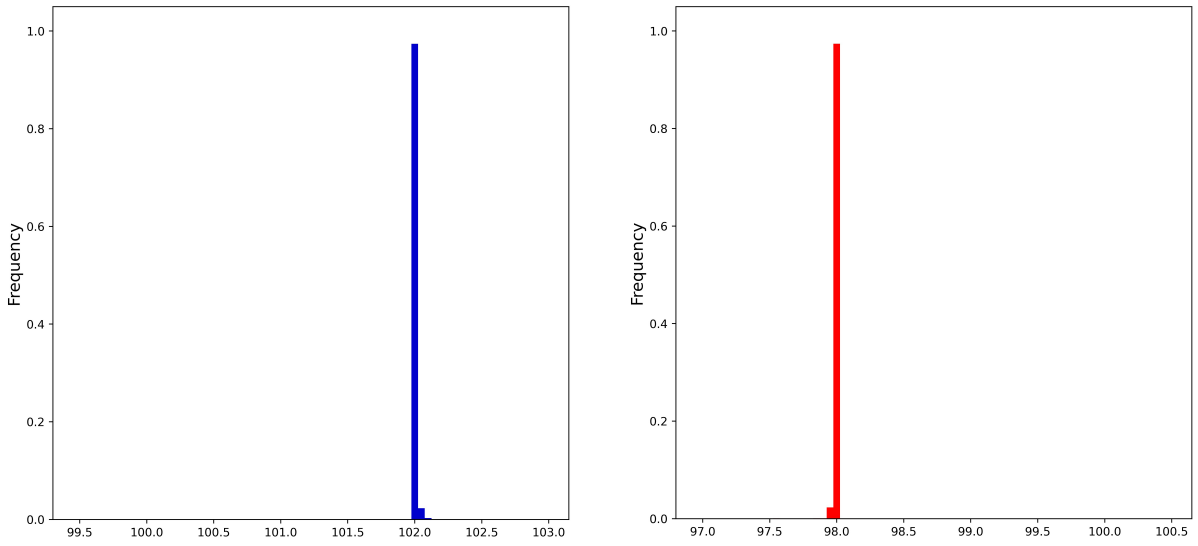
C.8 Lower Initial Q-values

In Section 5, we considered an "optimistic" distribution of initial Q-values, i.e., the initial Q-values were drawn from a uniform distribution over $[5, 8]$. For this section, we initialize using positive but lower initial Q-values, specifically, they are drawn from a uniform distribution over $[0.01, 0.05]$. We focus only on the case of price-inelastic noise traders. We run these simulations for $D = 1,000,000$ days. All other parameters are as listed in Table 1. We find no significant difference between the results below and those discussed in Section 6. Appendix Figures C23 and C24 report, respectively, the mean market quotes over time and the distribution of final-day quotes.



The graph plots the mean market quotes over days, averaging across the 1,000 repetitions. The graph refers to the baseline Q-learning setup. The mean ask (bid) is the dark blue (red) line; the blue and red shaded bands represent the 90% range across the simulations for the ask and bid, respectively. The dashed lines plot the competitive ask and bid prices, as discussed in Section 3. The mean market ask and bid quotes in the final day are 102 and 98, respectively.

Figure C23: Lower Initial Q-values - Mean Market Quotes Over Days



(a) Ask (b) Bid

The graph shows the distributions of best ask and bid quotes across the 1,000 repetitions on the final day. The graph refers to the baseline Q-learning setup.

Figure C24: Lower Initial Q-values - Distribution of AMM Quotes in $d = D = 1,000,000$

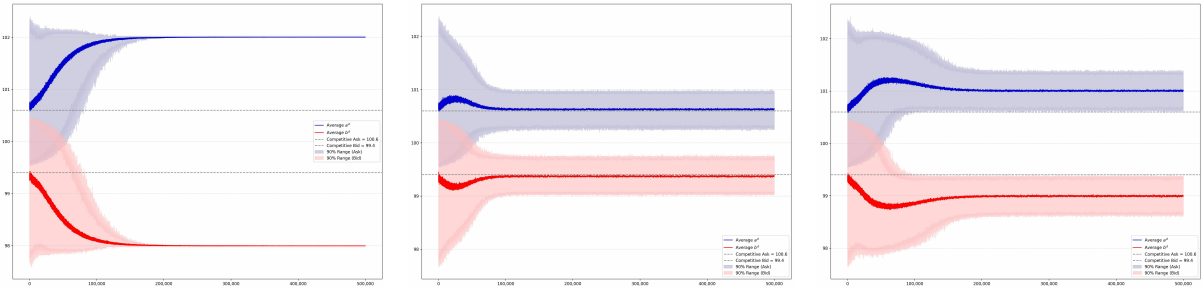
C.9 Simulations Underlying the Algorithm-Choice Game

In this subsection, we report the simulations behind the payoff matrices of Section 9 (Figures 6 and 7). Each payoff is computed from the mean quotes in the final trading day, averaged

across the 1,000 repetitions. For $N = 2$, every case requires three simulations: two B-AMMs, two CF-AMMs, and one CF-AMM competing against one B-AMM. In the latter, the B-AMM is priced out, so the market quotes are those of the CF-AMM; the graphs plot the quotes of the CF-AMM. For $N \geq 3$, we report one simulation per composition of the market (e.g., BBC denotes two B-AMMs and one CF-AMM); in every composition with at least one CF-AMM, the graphs plot the quotes of a CF-AMM.

The cases with $c_i = 0.05$ (Figures 6b and 6e) impose the exploration floor on all AMMs over the entire simulation; that is, $c_i = 0.05$ for all $d \leq D = 500{,}000$. This differs from Appendix C.7, where the floor is removed after $\tilde{d} = 250{,}000$ days. The decoupled cases (Figures 6c and 6f) are reported in Appendix C.11: Appendix Figures C46 and C49 for two B-AMMs, Appendix Figures C52 and C55 for two CF-AMMs, and Appendix Figures C58 and C59 for one CF-AMM against one B-AMM.

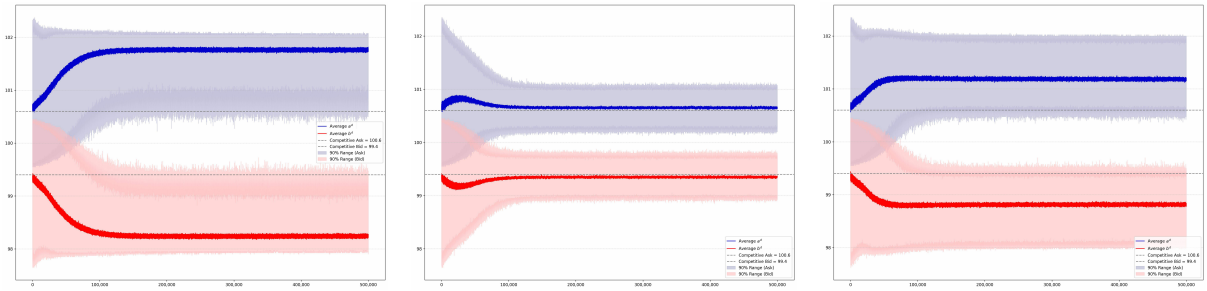
C.9.1 Two Firms, Constant Learning Rate



(a) (B-AMM, B-AMM) (b) (CF-AMM, CF-AMM) (c) (CF-AMM, B-AMM)

The graphs plot the mean quotes over days, averaging across the 1,000 repetitions, for price-inelastic noise traders and the parameterization of Section 5. The mean ask (bid) is the dark blue (red) line; the blue and red shaded bands represent the 90% range across the simulations for the ask and bid, respectively. The dashed lines plot the competitive ask and bid prices. Panel (c) plots the quotes of the CF-AMM. The mean ask and bid quotes in the final day are 102.00 and 98.00 in Panel (a), 100.64 and 99.38 in Panel (b), and 101.01 and 98.99 in Panel (c). These figures refer to the payoff matrix in Figure 6a.

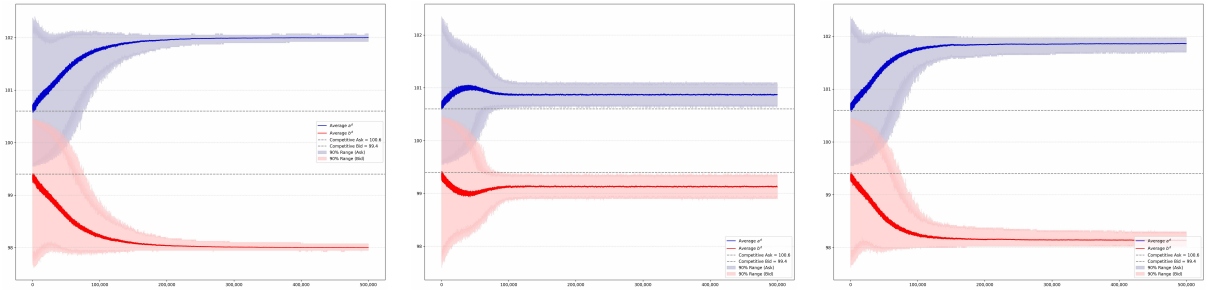
Figure C25: Algorithm-Choice Game, Inelastic: Baseline – Mean Quotes over Time



(a) (B-AMM, B-AMM) (b) (CF-AMM, CF-AMM) (c) (CF-AMM, B-AMM)

The graphs plot the mean quotes over days, averaging across the 1,000 repetitions, for price-inelastic noise traders with an exploration floor of 0.05 over the entire simulation; that is, $c_i = 0.05$ for all $d \leq 500{,}000$. The mean ask (bid) is the dark blue (red) line; the blue and red shaded bands represent the 90% range across the simulations for the ask and bid, respectively. The dashed lines plot the competitive ask and bid prices. Panel (c) plots the quotes of the CF-AMM. The mean ask and bid quotes in the final day are 101.75 and 98.27 in Panel (a), 100.65 and 99.36 in Panel (b), and 101.19 and 98.81 in Panel (c). These figures refer to the payoff matrix in Figure 6b.

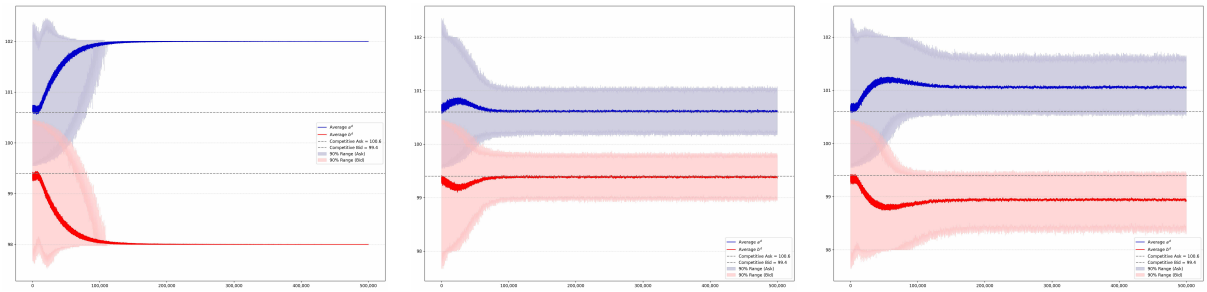
Figure C26: Algorithm-Choice Game, Inelastic: $c_i = 0.05$ – Mean Quotes over Time



(a) (B-AMM, B-AMM) (b) (CF-AMM, CF-AMM) (c) (CF-AMM, B-AMM)

The graphs plot the mean quotes over days, averaging across the 1,000 repetitions, for price-inelastic noise traders with decoupled updating of the ask and bid Q-values (Appendix C.11). The mean ask (bid) is the dark blue (red) line; the blue and red shaded bands represent the 90% range across the simulations for the ask and bid, respectively. The dashed lines plot the competitive ask and bid prices. Panel (c) plots the quotes of the CF-AMM. The mean ask and bid quotes in the final day are 102.00 and 98.00 in Panel (a), 100.87 and 99.12 in Panel (b), and 101.86 and 98.14 in Panel (c). These figures refer to the payoff matrix in Figure 6c.

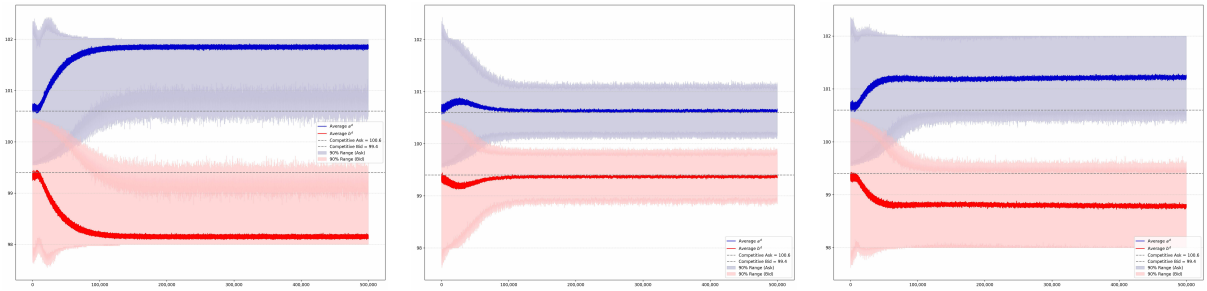
Figure C27: Algorithm-Choice Game, Inelastic: Decoupled – Mean Quotes over Time



(a) (B-AMM, B-AMM) (b) (CF-AMM, CF-AMM) (c) (CF-AMM, B-AMM)

The graphs plot the mean quotes over days, averaging across the 1,000 repetitions, for price-elastic noise traders and the parameterization of Section 5. The mean ask (bid) is the dark blue (red) line; the blue and red shaded bands represent the 90% range across the simulations for the ask and bid, respectively. The dashed lines plot the competitive ask and bid prices. Panel (c) plots the quotes of the CF-AMM. The mean ask and bid quotes in the final day are 102.00 and 98.00 in Panel (a), 100.61 and 99.37 in Panel (b), and 101.06 and 98.94 in Panel (c). These figures refer to the payoff matrix in Figure 6d.

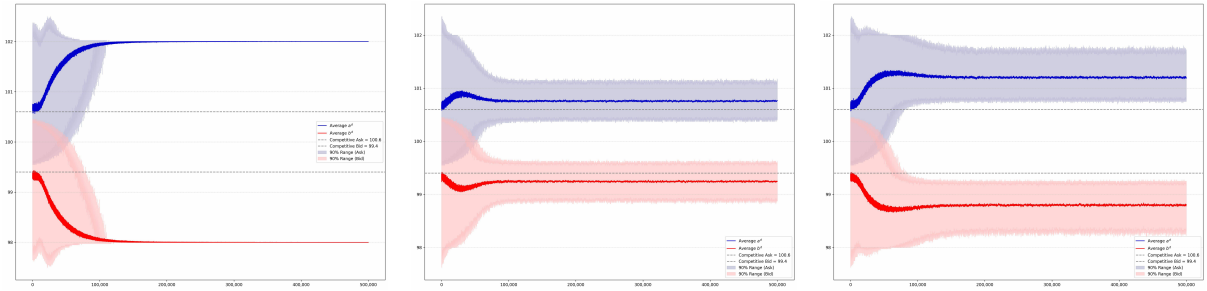
Figure C28: Algorithm-Choice Game, Elastic: Baseline – Mean Quotes over Time



(a) (B-AMM, B-AMM) (b) (CF-AMM, CF-AMM) (c) (CF-AMM, B-AMM)

The graphs plot the mean quotes over days, averaging across the 1,000 repetitions, for price-elastic noise traders with an exploration floor of 0.05 over the entire simulation; that is, $c_i = 0.05$ for all $d \leq 500{,}000$. The mean ask (bid) is the dark blue (red) line; the blue and red shaded bands represent the 90% range across the simulations for the ask and bid, respectively. The dashed lines plot the competitive ask and bid prices. Panel (c) plots the quotes of the CF-AMM. The mean ask and bid quotes in the final day are 101.88 and 98.12 in Panel (a), 100.64 and 99.36 in Panel (b), and 101.22 and 98.78 in Panel (c). These figures refer to the payoff matrix in Figure 6e.

Figure C29: Algorithm-Choice Game, Elastic: $c_i = 0.05$ – Mean Quotes over Time



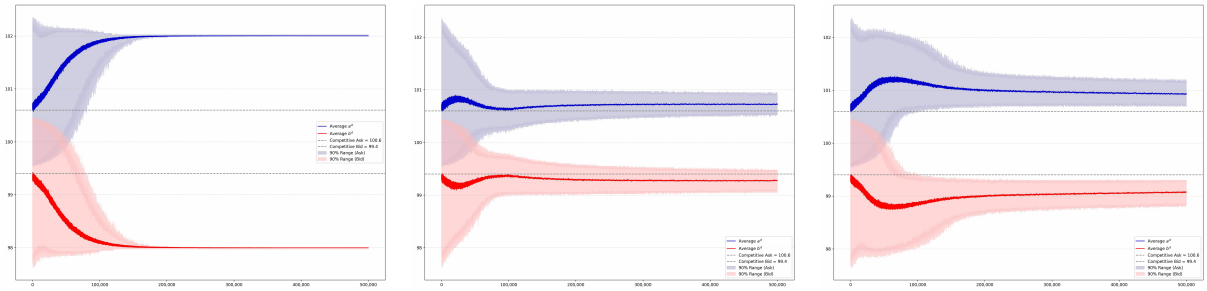
(a) (B-AMM, B-AMM) (b) (CF-AMM, CF-AMM) (c) (CF-AMM, B-AMM)

The graphs plot the mean quotes over days, averaging across the 1,000 repetitions, for price-elastic noise traders with decoupled updating of the ask and bid Q-values (Appendix C.11). The mean ask (bid) is the dark blue (red) line; the blue and red shaded bands represent the 90% range across the simulations for the ask and bid, respectively. The dashed lines plot the competitive ask and bid prices. Panel (c) plots the quotes of the CF-AMM. The mean ask and bid quotes in the final day are 102.00 and 98.00 in Panel (a), 100.77 and 99.25 in Panel (b), and 101.20 and 98.81 in Panel (c). These figures refer to the payoff matrix in Figure 6f.

Figure C30: Algorithm-Choice Game, Elastic: Decoupled – Mean Quotes over Time

C.9.2 N Firms, Robbins-Monro Learning Rate for the CF-AMM

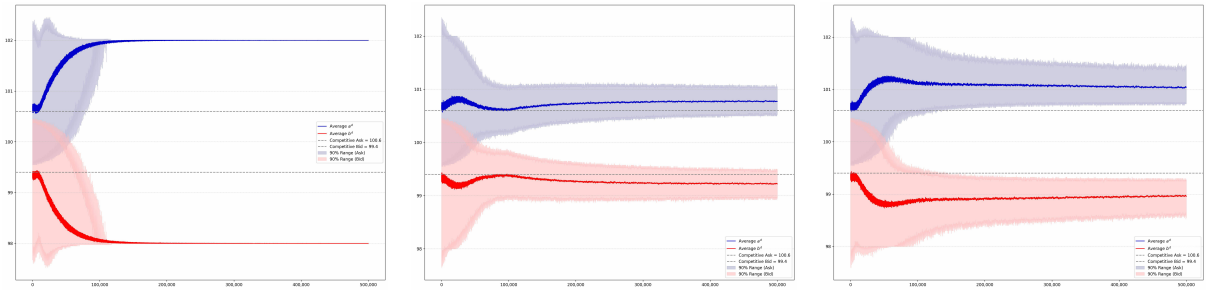
In the simulations behind Figure 7, the learning rate of a B-AMM is constant and equal to 0.10, whereas the learning rate of a CF-AMM is $\lambda_i^d = \min\left\{0.1, \frac{1}{0.0001d}\right\}$. For $N = 5$, we report the compositions entering Figure 7; the simulation with five CF-AMMs is extended to $D = 2{,}000{,}000$ days.



(a) (B-AMM, B-AMM) (b) (CF-AMM, CF-AMM) (c) (CF-AMM, B-AMM)

The graphs plot the mean quotes over days, averaging across the 1,000 repetitions, for price-inelastic noise traders and $N = 2$. The mean ask (bid) is the dark blue (red) line; the blue and red shaded bands represent the 90% range across the simulations for the ask and bid, respectively. The dashed lines plot the competitive ask and bid prices. Panel (c) plots the quotes of the CF-AMM. The mean ask and bid quotes in the final day are 102.00 and 98.00 in Panel (a), 100.72 and 99.28 in Panel (b), and 100.94 and 99.08 in Panel (c).

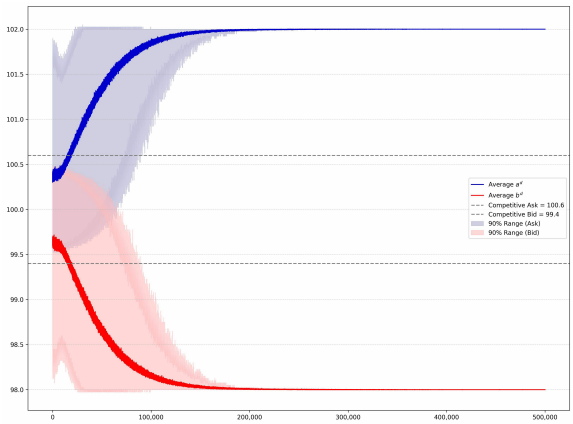
Figure C31: Algorithm-Choice Game, $N = 2$, Inelastic – Mean Quotes over Time



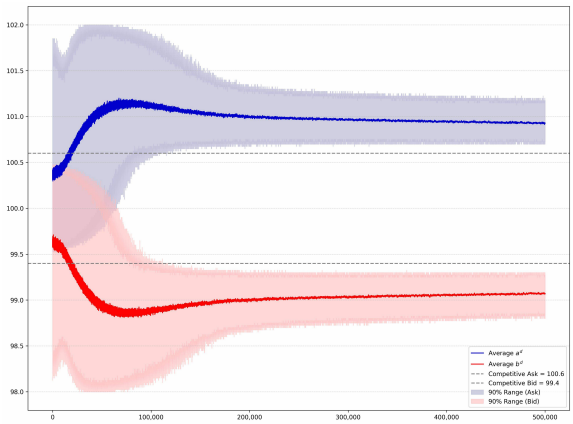
(a) (B-AMM, B-AMM) (b) (CF-AMM, CF-AMM) (c) (CF-AMM, B-AMM)

The graphs plot the mean quotes over days, averaging across the 1,000 repetitions, for price-elastic noise traders and $N = 2$. The mean ask (bid) is the dark blue (red) line; the blue and red shaded bands represent the 90% range across the simulations for the ask and bid, respectively. The dashed lines plot the competitive ask and bid prices. Panel (c) plots the quotes of the CF-AMM. The mean ask and bid quotes in the final day are 102.00 and 98.00 in Panel (a), 100.77 and 99.22 in Panel (b), and 101.04 and 98.96 in Panel (c).

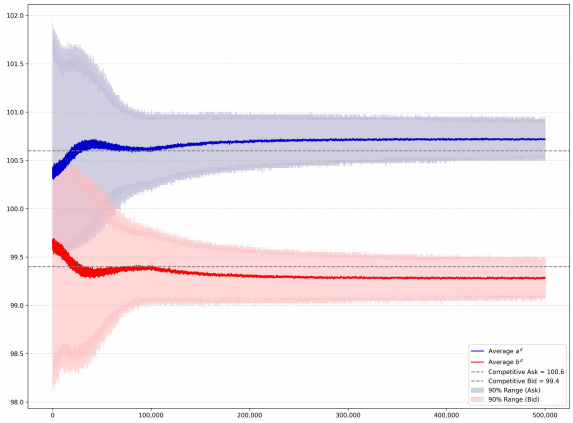
Figure C32: Algorithm-Choice Game, $N = 2$, Elastic – Mean Quotes over Time



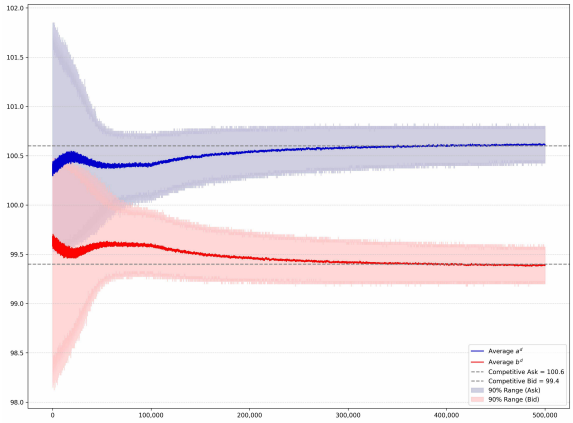
(a) BBB



(b) BBC



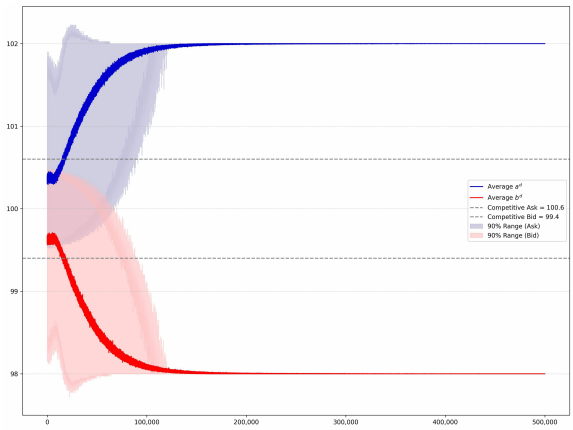
(c) BCC



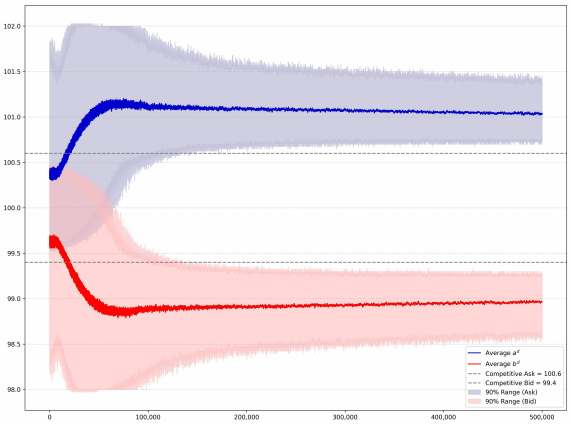
(d) CCC

The graphs plot the mean quotes over days, averaging across the 1,000 repetitions, for price-inelastic noise traders and $N = 3$. B (C) denotes a B-AMM (CF-AMM). The mean ask (bid) is the dark blue (red) line; the blue and red shaded bands represent the 90% range across the simulations for the ask and bid, respectively. The dashed lines plot the competitive ask and bid prices. Panels (b)–(d) plot the quotes of a CF-AMM. The mean ask and bid quotes in the final day are 102.00 and 98.00 in Panel (a), 100.93 and 99.07 in Panel (b), 100.72 and 99.28 in Panel (c), and 100.61 and 99.39 in Panel (d).

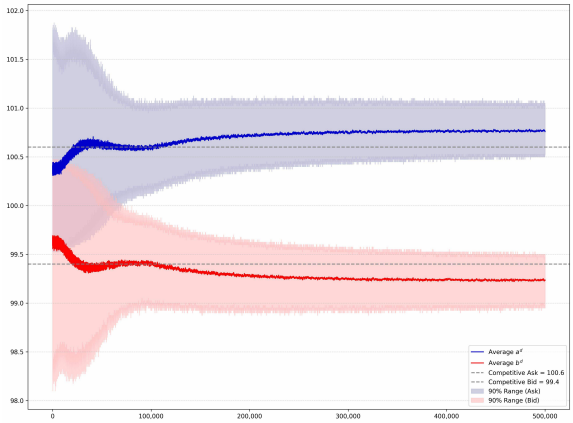
Figure C33: Algorithm-Choice Game, $N = 3$, Inelastic – Mean Quotes over Time



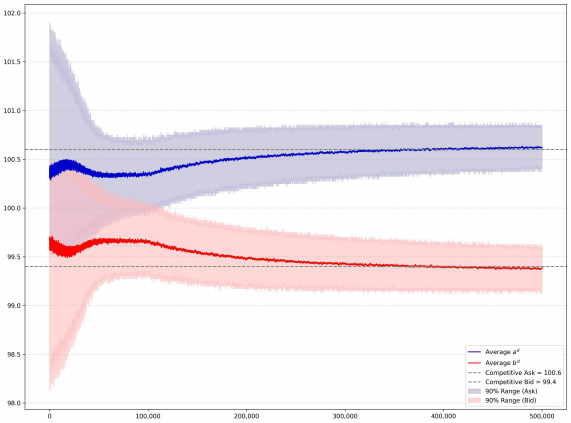
(a) BBB



(b) BBC



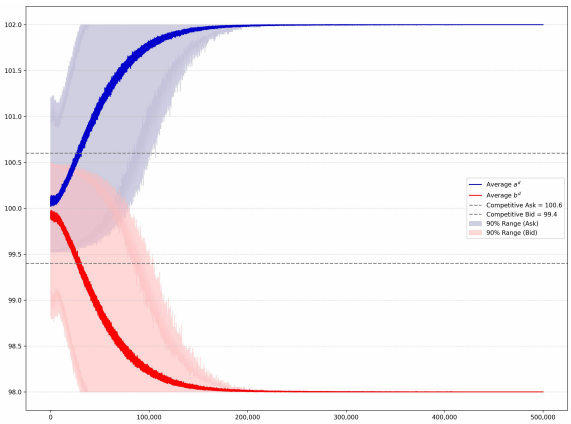
(c) BCC



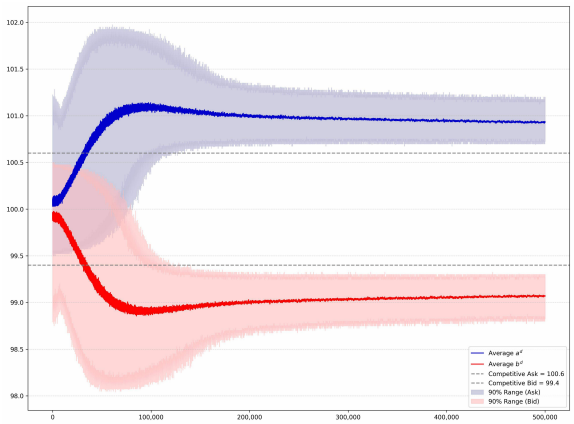
(d) CCC

The graphs plot the mean quotes over days, averaging across the 1,000 repetitions, for price-elastic noise traders and $N = 3$. B (C) denotes a B-AMM (CF-AMM). The mean ask (bid) is the dark blue (red) line; the blue and red shaded bands represent the 90% range across the simulations for the ask and bid, respectively. The dashed lines plot the competitive ask and bid prices. Panels (b)–(d) plot the quotes of a CF-AMM. The mean ask and bid quotes in the final day are 102.00 and 98.00 in Panel (a), 101.04 and 98.96 in Panel (b), 100.76 and 99.23 in Panel (c), and 100.62 and 99.38 in Panel (d).

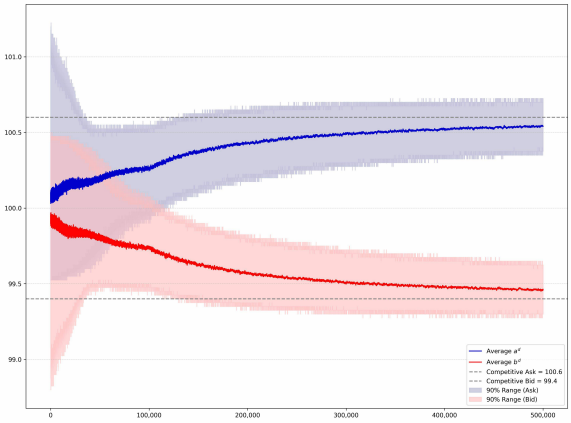
Figure C34: Algorithm-Choice Game, $N = 3$, Elastic – Mean Quotes over Time



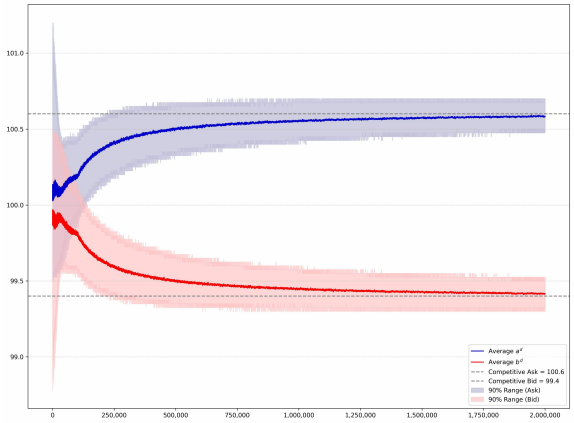
(a) BBBBB



(b) BBBBC



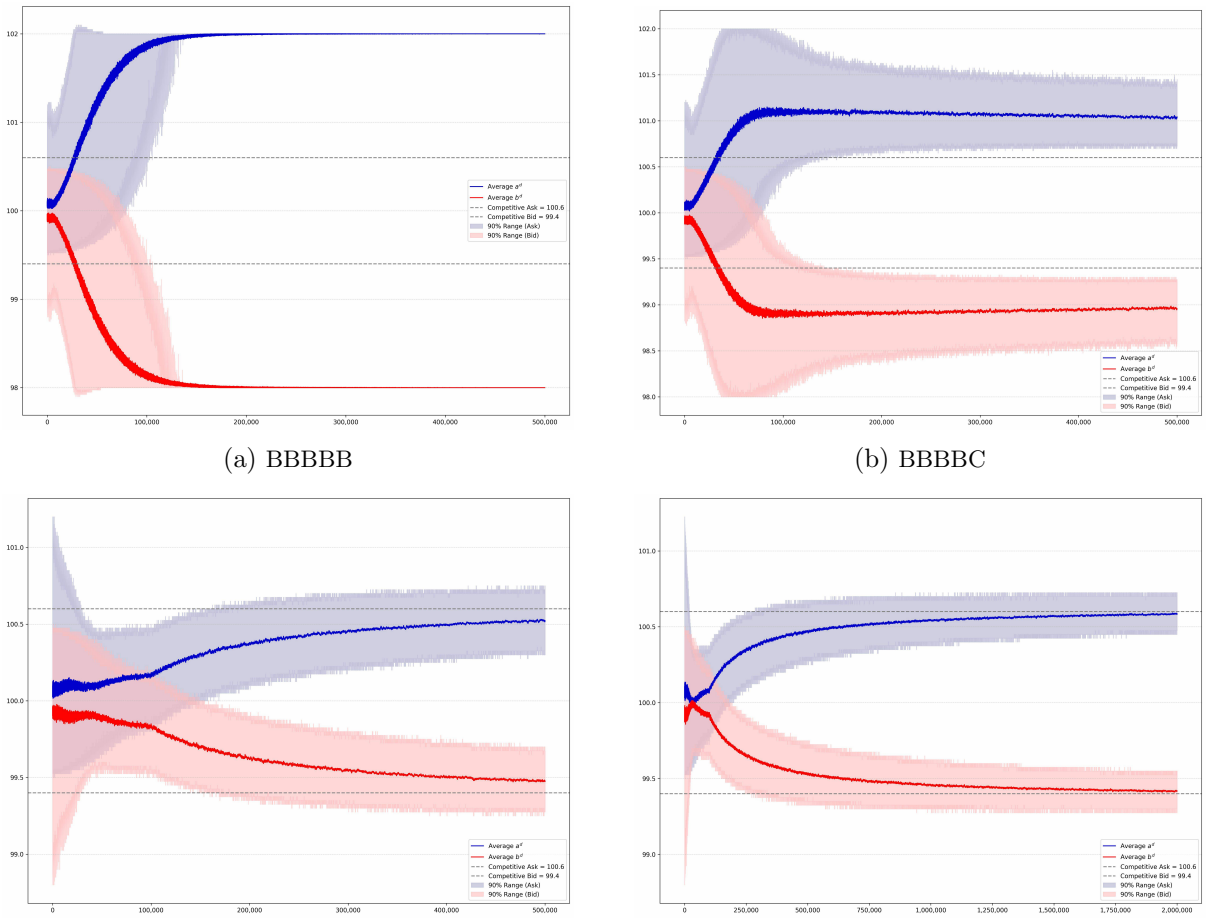
(c) BCCCC



(d) CCCCC ($D = 2{,}000{,}000$)

The graphs plot the mean quotes over days, averaging across the 1,000 repetitions, for price-inelastic noise traders and $N = 5$. B (C) denotes a B-AMM (CF-AMM). The mean ask (bid) is the dark blue (red) line; the blue and red shaded bands represent the 90% range across the simulations for the ask and bid, respectively. The dashed lines plot the competitive ask and bid prices. Panels (b)–(d) plot the quotes of a CF-AMM. The simulation in Panel (d) runs for $D = 2{,}000{,}000$ days. The mean ask and bid quotes in the final day are 102.00 and 98.00 in Panel (a), 100.93 and 99.07 in Panel (b), 100.54 and 99.46 in Panel (c), and 100.58 and 99.42 in Panel (d).

Figure C35: Algorithm-Choice Game, $N = 5$, Inelastic – Mean Quotes over Time



(a) BBBBB (b) BBBBC

(c) BCCCC (d) CCCCC ($D = 2{,}000{,}000$)

The graphs plot the mean quotes over days, averaging across the 1,000 repetitions, for price-elastic noise traders and $N = 5$. B (C) denotes a B-AMM (CF-AMM). The mean ask (bid) is the dark blue (red) line; the blue and red shaded bands represent the 90% range across the simulations for the ask and bid, respectively. The dashed lines plot the competitive ask and bid prices. Panels (b)–(d) plot the quotes of a CF-AMM. The simulation in Panel (d) runs for $D = 2{,}000{,}000$ days. The mean ask and bid quotes in the final day are 102.00 and 98.00 in Panel (a), 101.04 and 98.96 in Panel (b), 100.52 and 99.48 in Panel (c), and 100.59 and 99.42 in Panel (d).

Figure C36: Algorithm-Choice Game, $N = 5$, Elastic – Mean Quotes over Time

C.10 Varying the Hyperparameters

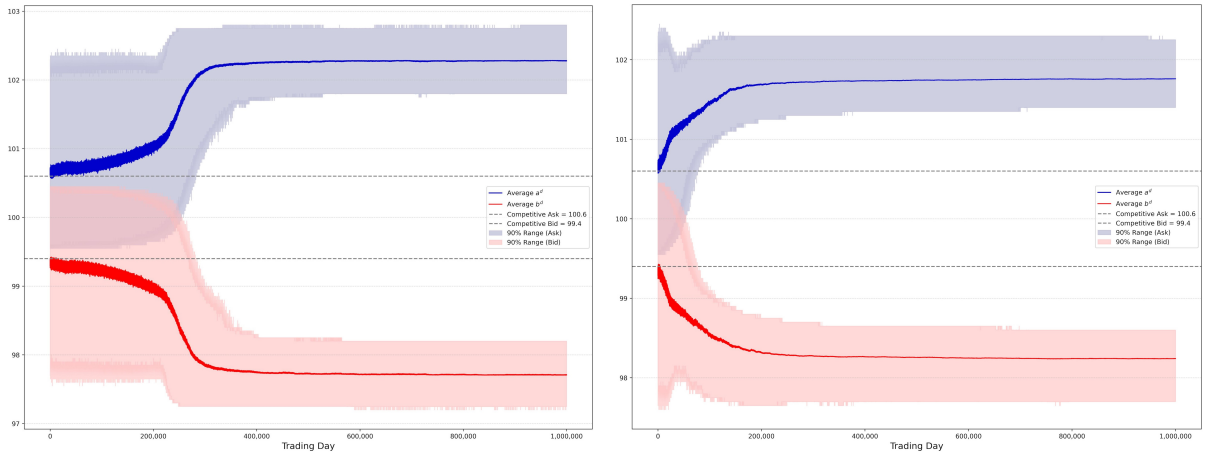
The theoretical loss-free pricing result that we stated in Section 6 is obtained in the absence of experimentation and asymptotically, as the number of days converges to infinity. Our simulation results for the baseline case, on the other hand, show that the same result occurs even in the presence of experimentation and in finite time. It is worth studying what happens when we consider alternative parameterizations to the baseline case. One possibility is to let AMMs react to market feedback in a different way: this means that they update the Q-values on the basis of a different value of λ_i^d (set equal to 0.1 in the baseline case); another is to let them have different exploration dynamics, due to different values of the parameters γ_i and c_i (set equal to 0.00004 and 0, respectively, in the baseline case), or to a different functional form of ε_i^d. Here, we only report the main results for these alternative specifications in terms of the mean market prices

over days and the distribution of market quotes in the final day for the case with price-inelastic noise traders. We simulate each of these specifications for $D = 1,000,000$ days. In the interest of space, we present the distributions only for the ask side.

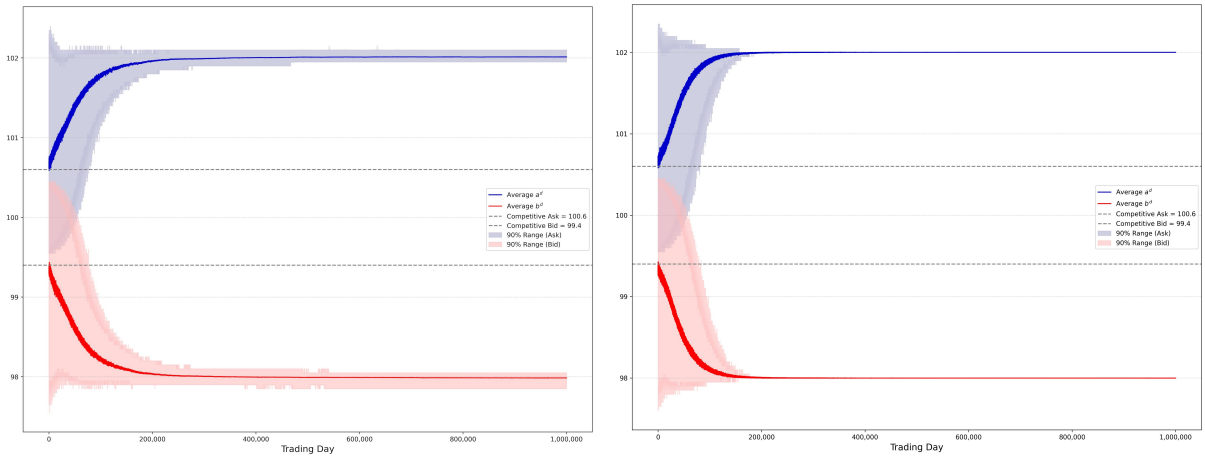
C.10.1 Varying the Learning Rate

As a first exercise, we set $\lambda_i^d = 0.01$ for both AMMs (i.e., $i = 1, 2$), leaving all other parameters unchanged from the baseline case. AMMs now give much less weight to any feedback they receive from the market. The resulting ask and bid prices are shown in Appendix Figure C37b, the analog of Figure 1. The final mean ask is 101.76, lower than v_H (102), although not by much, and still substantially above the competitive equilibrium ask. Similarly, the final mean bid is 98.24, higher than v_L (98), but still far from the competitive bid.

In the other panels of Appendix Figure C37, we consider alternative λ_i values. We find that $\lambda_i^d = 0.05$ produces a very similar result to the baseline of $\lambda_i^d = 0.1$ (reported in Panel d). Interestingly, a similar result is also observed for $\lambda_i^d = 0.001$ (Appendix Figure C37a), in this case due to insufficient learning. Appendix Figure C38 shows the corresponding distributions of the market ask quotes in the final day.



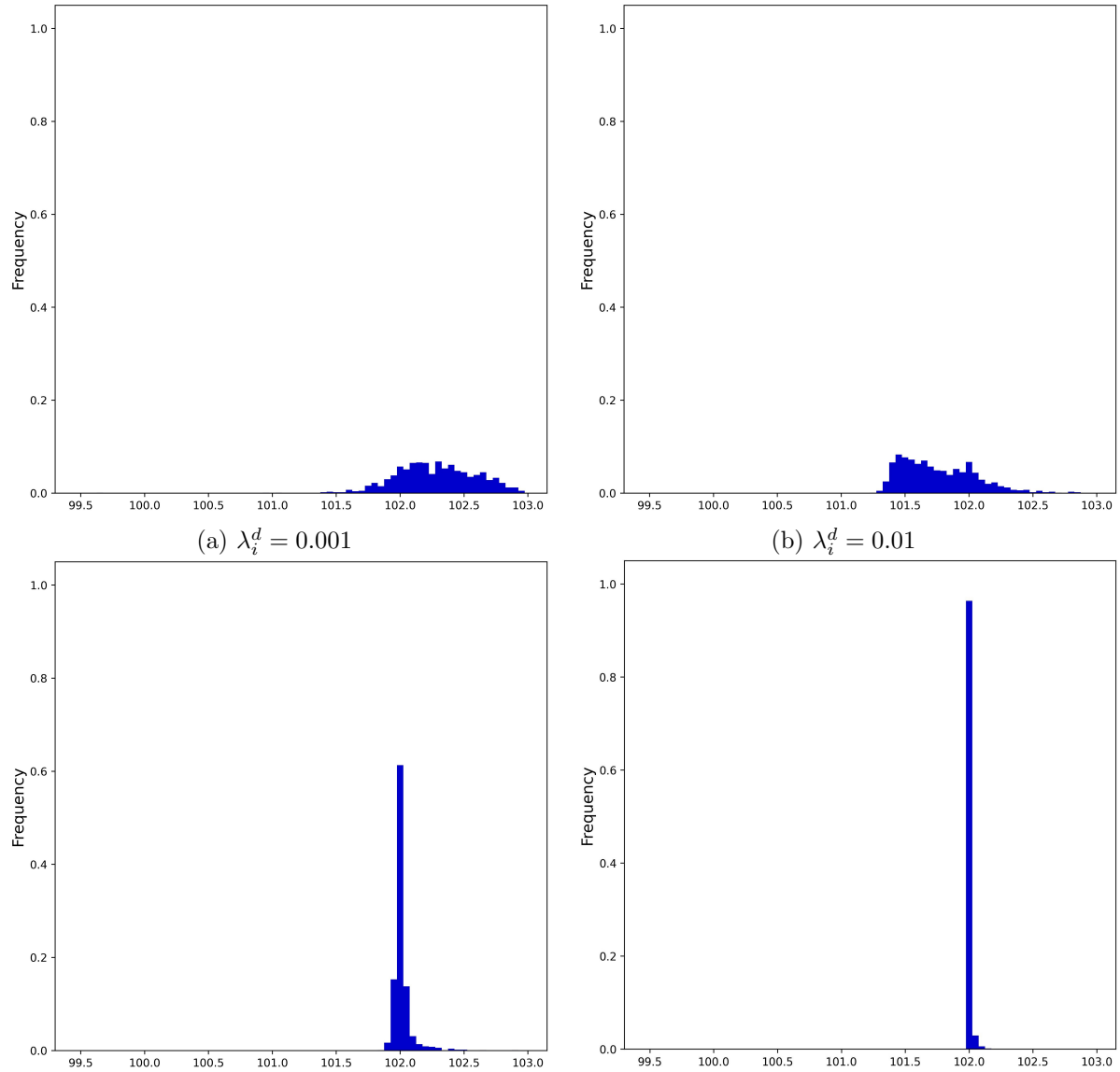
(a) $\lambda_i^d = 0.001$ (b) $\lambda_i^d = 0.01$



(c) $\lambda_i^d = 0.05$ (d) $\lambda_i^d = 0.1(*)$

The graph plots the mean market quotes over days, averaging across the 1,000 repetitions. The graph refers to the baseline Q-learning setup with price-inelastic noise traders and (a) $\lambda_i^d = 0.001$, (b) $\lambda_i^d = 0.01$, (c) $\lambda_i^d = 0.05$, and (d) $\lambda_i^d = 0.1$. The mean ask (bid) is the dark blue (red) line; the blue and red shaded bands represent the 90% range across the simulations for the ask and bid, respectively. The dashed lines plot the competitive ask and bid prices, as discussed in Section 3. The "(*)" refers to the baseline parameterization that we considered in Section 6. The mean market ask and bid quotes in the final day are 102.28 and 97.71 in Panel (a), 101.76 and 98.24 in Panel (b), 102.01 and 97.99 in Panel (c), and 102 and 98 in Panel (d).

Figure C37: λ_i^d Comparative Statics - Mean Market Quotes Over Days



(a) $\lambda_i^d = 0.001$ (b) $\lambda_i^d = 0.01$

(c) $\lambda_i^d = 0.05$ (d) $\lambda_i^d = 0.1(*)$

Each panel shows the distribution of the market ask quotes across the 1,000 repetitions in the final day, for a given value of λ_i^d. The graph refers to the baseline Q-learning setup with price-inelastic noise traders. The "(*)" refers to the baseline parameterization that we considered in Section 6.

Figure C38: λ_i^d Comparative Statics - Distribution of Market Ask Prices in Day d = 1,000,000

To understand these results, it is worth going back to the intuitions for Proposition 1. The loss-free pricing result is obtained because, in infinite horizon, with probability 1, an ask strictly lower than v_H is associated with a sufficiently large number of negative payoffs that its Q-value becomes negative, and, thus, it can no longer be the greedy action. In the absence of experimentation, that ask is never chosen again. With relatively large λ_i^d, as in our baseline simulation, just a few negative payoffs are enough to make the Q-value of an ask price below v_H negative. When $\lambda_i^d = 0.01$ rather than 0.1, the number of negative experiences needed to bring the Q-value below zero becomes much larger, since each experience is weighted much less in the

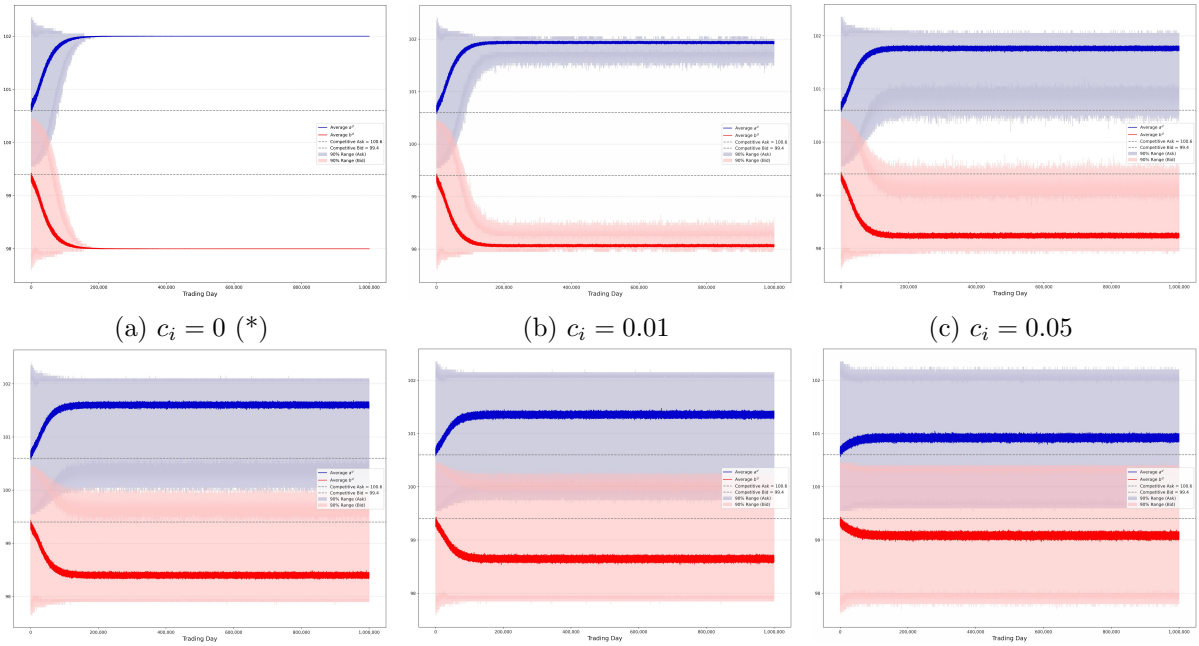
updating process.[27] As our simulations reveal, we were able to move away from the loss-free pricing result with this lower λ_i^d.[28] When λ_i^d is too small, however, the algorithm does not react strongly enough to market feedback and the baseline result occurs again.

C.10.2 Varying the Minimum Exploration Probability

Rather than changing the learning rate, λ_i^d, let us now see what happens if we change the frequency with which the AMMs explore rather than exploit. As a first exercise, we set $c_i = 0.05$, which means that, in each day, there is at least a 5% probability of exploration. As Appendix Figure C39c shows, the mean ask is below the loss-free price, although still far from the competitive price. Exploration leads to a lower average ask, although the change is not very large. Part of this may look mechanical, since prices (including lower ask prices) are chosen by exploration without being optimal. However, there is a more important mechanism at work: intuitively, without exploration, once the Q-value for an ask price above the competitive equilibrium becomes negative, that price will not be used again (since it cannot be the greedy action) and the AMM is unable to learn that it is profitable (in expectation). With a constant experimentation probability, the price will be used again and, eventually, the Q-value can become positive. With a sequence of one million days, AMMs learn that at least some prices are profitable, although not those too close to the competitive equilibrium. Indeed, for asks and bids closer to the competitive equilibrium, a negative Q-value requires more frequent random exploration before becoming positive again, since the expected profits from these prices are smaller than those from more extreme prices. Another notable feature in Appendix Figure C39 is that, even at the end of one million days, there is a great deal of heterogeneity in prices, as shown by the thicker mean price lines (greater variation) and wider ranges across simulations for both ask and bid when $c_i > 0$. This is the cost of experimentation, since all prices keep being tried. Appendix Figure C40 shows the corresponding distributions of the market ask quotes in the final day.

[27] For example, suppose that the Q-value for an ask price of 101 is 0.2: when $\lambda_i^d = 0.1$, it takes two losses (the loss in our given example is -1) to reach a negative Q-value; in contrast, for $\lambda_i^d = 0.01$, the number of required losses is 19.

[28] In other words, across the simulations, there are sequences of one million days in which the Q-values of ask prices below v_H are positive in the last days.

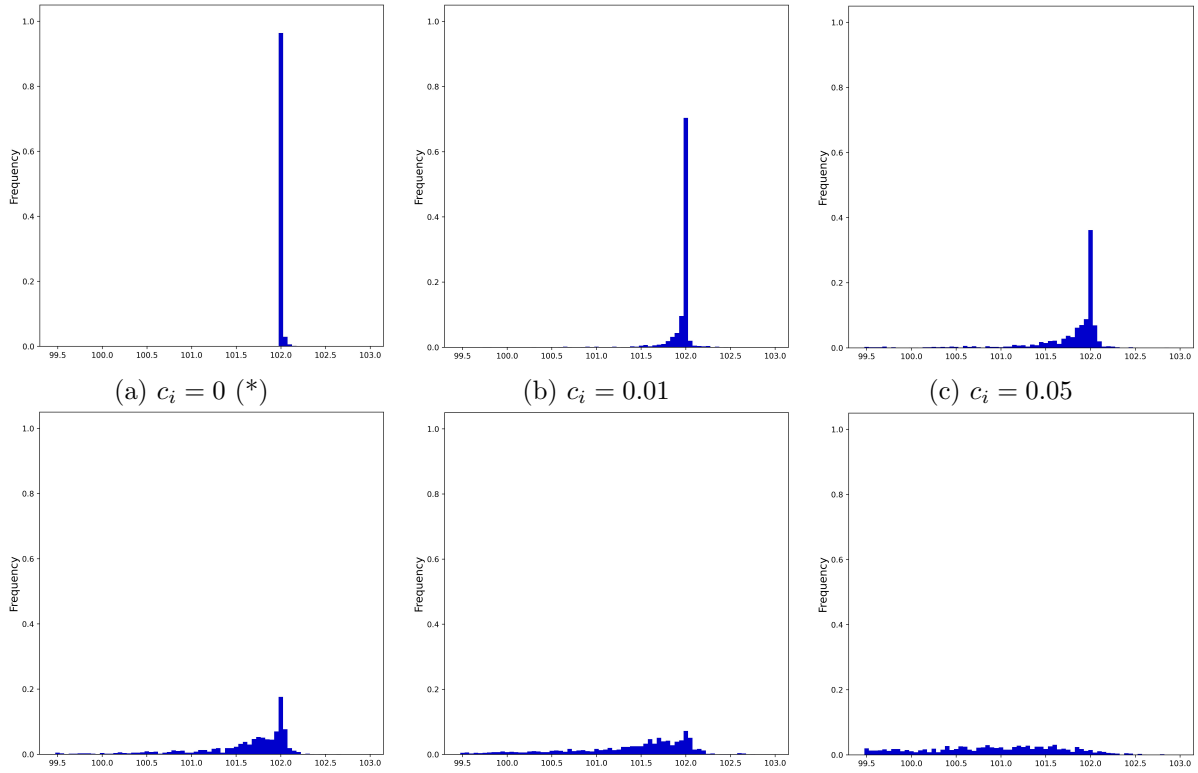


(a) $c_i = 0$ (*) (b) $c_i = 0.01$ (c) $c_i = 0.05$

(d) $c_i = 0.10$ (e) $c_i = 0.20$ (f) $c_i = 0.50$

The graph plots the mean market quotes over days, averaging across the 1,000 repetitions. The graph refers to the baseline Q-learning setup with price-inelastic noise traders and (a) $c_i = 0$, (b) $c_i = 0.01$, (c) $c_i = 0.05$, (d) $c_i = 0.10$, (e) $c_i = 0.20$, and (f) $c_i = 0.50$. The mean ask (bid) is the dark blue (red) line; the blue and red shaded bands represent the 90% range across the simulations for the ask and bid, respectively. The dashed lines plot the competitive ask and bid prices, as discussed in Section 3. The "(*)" refers to the baseline parameterization that we considered in Section 6. The mean market ask and bid quotes in the final day are 102 and 98 in Panel (a), 101.94 and 98.06 in Panel (b), 101.77 and 98.24 in Panel (c), 101.59 and 98.40 in Panel (d), 101.37 and 98.65 in Panel (e), and 100.92 and 99.07 in Panel (f).

Figure C39: c_i Comparative Statics - Mean Market Quotes Over Days



(a) $c_i = 0$ (*) (b) $c_i = 0.01$ (c) $c_i = 0.05$

(d) $c_i = 0.10$ (e) $c_i = 0.20$ (f) $c_i = 0.50$

Each panel shows the distribution of the market ask quotes across the 1,000 repetitions in the final day, for a given value of c_i. The graph refers to the baseline Q-learning setup with price-inelastic noise traders. The "(*)" refers to the baseline parameterization that we considered in Section 6.

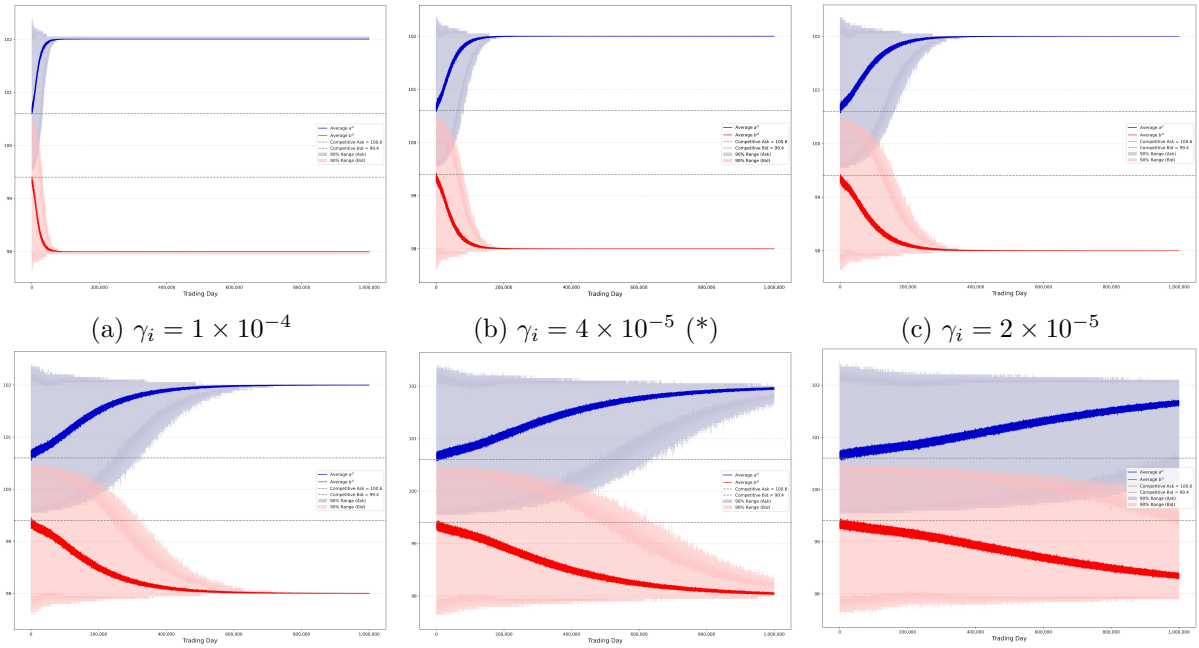
Figure C40: c_i Comparative Statics - Distribution of Market Ask Prices in Day $d = 1{,}000{,}000$

C.10.3 Varying the Exploration Decay Rate

An alternative to considering different lower bounds on the level of exploration (c_i) is to consider different values for the exploration decay parameter, γ_i. We have done this exercise for different values of γ_i:

$$\gamma_i = \{1 \times 10^{-4}, \quad 4 \times 10^{-5}, \quad 2 \times 10^{-5}, \quad 1 \times 10^{-5}, \quad 5 \times 10^{-6}, \quad 2.5 \times 10^{-6}\}.$$

Broadly speaking, these different parameter values do not produce significantly different results. Appendix Figure C41 shows the market quotes over days, with each subplot representing a different value of γ_i. For a larger γ_i, we see that the simulations reach the result at an accelerated speed; as we decrease γ_i towards zero, we see that the time taken for the simulations to reach ask prices of 102 and bid prices of 98 increases. At the smallest value we consider ($\gamma_i = 2.5 \times 10^{-6}$), we see that, whilst the final ask and bid quotes are 101.68 and 98.33, respectively, no convergence has been reached and the prices are still trending upwards. Appendix Figure C42 shows the corresponding distributions of the market ask quotes in the final day.

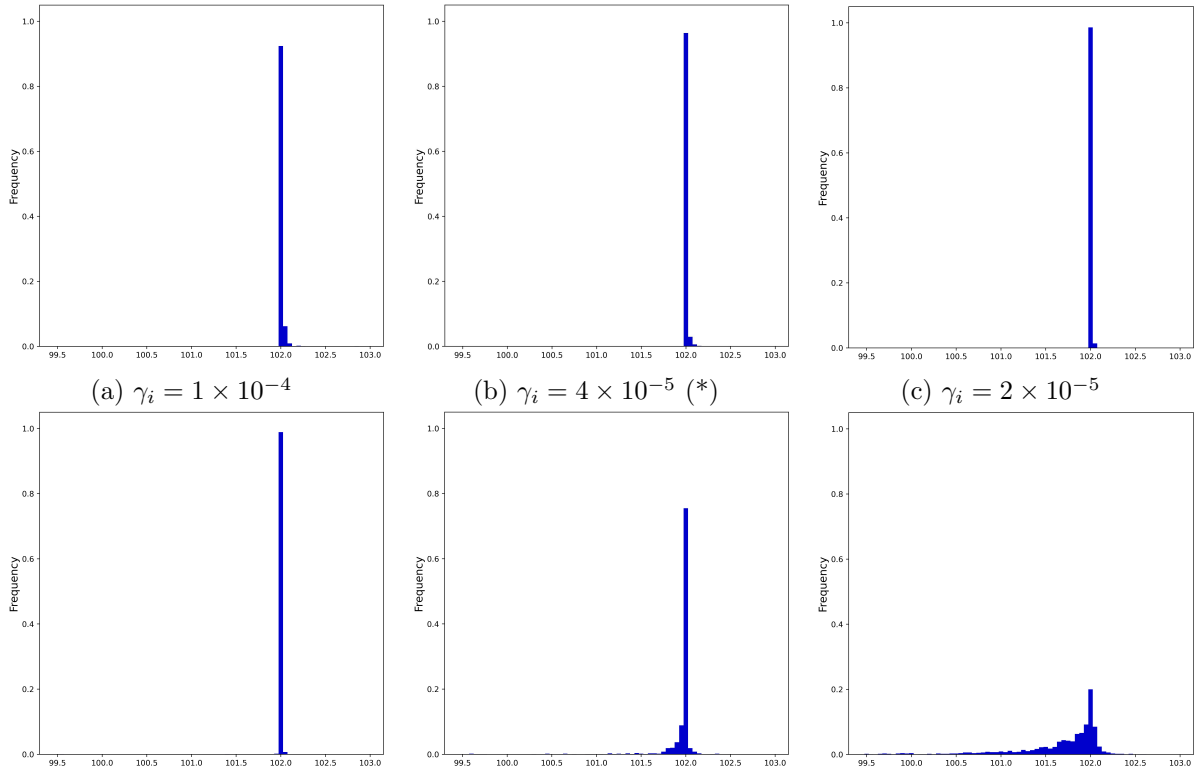


(a) $\gamma_i = 1 \times 10^{-4}$ (b) $\gamma_i = 4 \times 10^{-5}$ (*) (c) $\gamma_i = 2 \times 10^{-5}$

(d) $\gamma_i = 1 \times 10^{-5}$ (e) $\gamma_i = 5 \times 10^{-6}$ (f) $\gamma_i = 2.5 \times 10^{-6}$

The graph plots the mean market quotes over days, averaging across the 1,000 repetitions. The graph refers to the baseline Q-learning setup with price-inelastic noise traders and (a) $\gamma_i = 1 \times 10^{-4}$, (b) $\gamma_i = 4 \times 10^{-5}$, (c) $\gamma_i = 2 \times 10^{-5}$, (d) $\gamma_i = 1 \times 10^{-5}$, (e) $\gamma_i = 5 \times 10^{-6}$, and (f) $\gamma_i = 2.5 \times 10^{-6}$. The mean ask (bid) is the dark blue (red) line; the blue and red shaded bands represent the 90% range across the simulations for the ask and bid, respectively. The dashed lines plot the competitive ask and bid prices, as discussed in Section 3. The "(*)" refers to the baseline parameterization that we considered in Section 6. The mean market ask and bid quotes in the final day are 102 and 98 in Panels (a) through (d), 101.95 and 98.05 in Panel (e), and 101.68 and 98.33 in Panel (f).

Figure C41: γ_i Comparative Statics - Mean Market Quotes Over Days



(a) $\gamma_i = 1 \times 10^{-4}$ (b) $\gamma_i = 4 \times 10^{-5}$ (*) (c) $\gamma_i = 2 \times 10^{-5}$

(d) $\gamma_i = 1 \times 10^{-5}$ (e) $\gamma_i = 5 \times 10^{-6}$ (f) $\gamma_i = 2.5 \times 10^{-6}$

Each panel shows the distribution of the market ask quotes across the 1,000 repetitions in the final day, for a given value of γ_i. The graph refers to the baseline Q-learning setup with price-inelastic noise traders. The "(*)" refers to the baseline parameterization that we considered in Section 6.

Figure C42: γ_i Comparative Statics - Distribution of Market Ask Prices in Day $d = 1{,}000{,}000$

C.10.4 Soft-max Exploration

The ε-greedy exploration is only one of the possible exploration strategies considered in machine learning. Another common exploration method is *logit exploration*, also known in the machine learning literature as "Boltzmann exploration" or "soft-max exploration" (see, e.g., Sutton and Barto, 2018).[29]

Rather than having the best action chosen with probability $1 - \varepsilon_i^d$ (in case of exploitation) and all actions chosen uniformly with probability ε_i^d (in case of experimentation), with logit exploration more promising actions are explored with higher probabilities. In our context, this means that AMMs choose asks and bids with higher Q-values with a higher probability. Specifically, on day d, AMM_i chooses the ask α_j with the following probability:

$$\Pr(a_i^d = \alpha_j) = \frac{e^{\tau Q_i(\alpha_j)}}{\sum_{\alpha_l \in \mathbf{A}} e^{\tau Q_i(\alpha_l)}},$$

[29] Logit exploration has also been used in learning models in Game Theory, see, e.g., Mookherjee and Sopher (1997) and Camerer and Ho (1999).

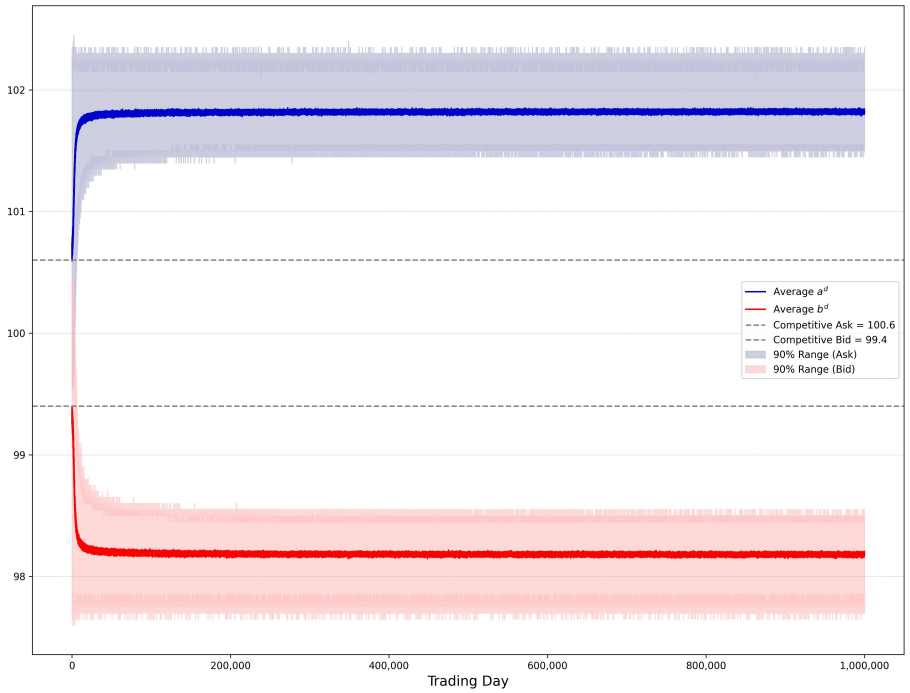
and, similarly, chooses the bid β_j with the following probability:

$$\Pr(b_i^d = \beta_j) = \frac{e^{\tau Q_i(\beta_j)}}{\sum_{\beta_l \in \mathbf{B}} e^{\tau Q_i(\beta_l)}},$$

where the parameter τ controls the extent of randomness: when $\tau = 0$, then this method chooses all asks (bids) uniformly, and when τ goes to ∞, then this method chooses the greedy ask (bid) with probability 1. For intermediate values, this method offers a smoother way of choosing different actions than ε-greedy exploration. The soft-max exploration functions differently from the ε-greedy in that zero is no longer a "cut-off" point for the Q-values. Previously, when a Q-value dropped below zero and exploration was near zero, this action would not be chosen again, even if the chosen alternative was close in value, but just positive. Now, if two Q-values are close but one is slightly positive and the other slightly negative, then the probabilities of these actions being chosen are quite similar. As a result, actions with potential losses can still be chosen through the soft-max exploration.

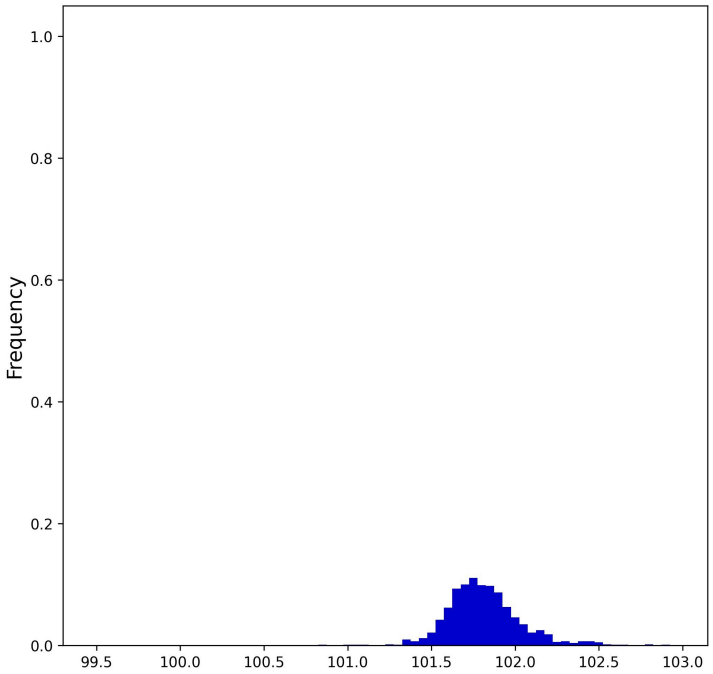
We now present some results for the illustrative case of $\tau = 20$. In the final day, we see (Appendix Figure C43) that the mean ask and bid prices are 101.82 and 98.20, within the $(98, 102)$ interval but still away from equilibrium prices. Appendix Figure C44 shows the distribution of the market ask quotes in the final day.

In contrast to the Q-tables displayed in Appendix Figure C3, we see that the AMMs attach positive Q-values to ask (bid) prices below v_H (respectively, above v_L) even in the last trading day (see Appendix Figure C45).



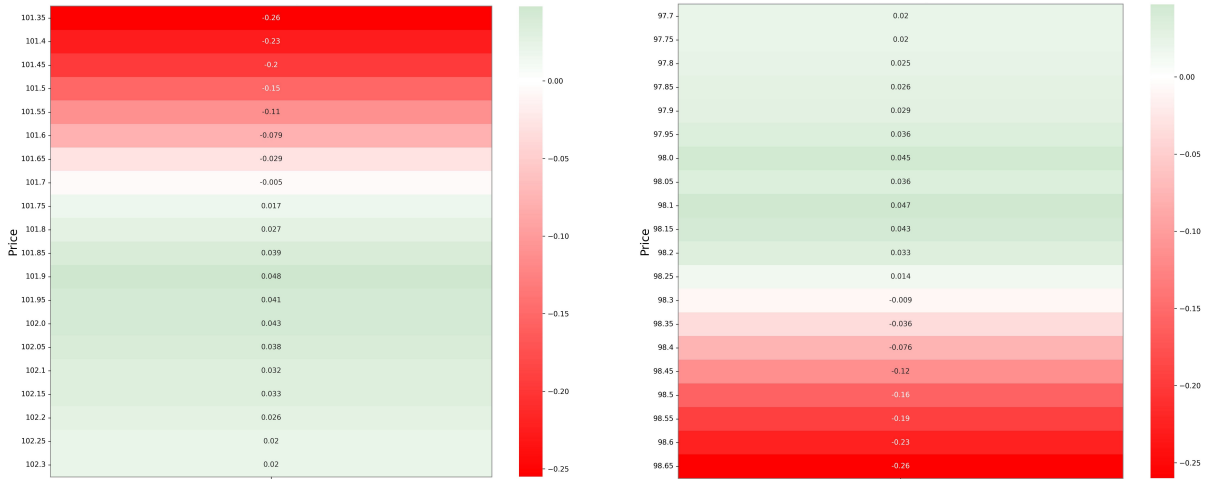
The graph plots the mean market quotes over days, averaging across the 1,000 repetitions. The graph refers to the baseline Q-learning setup. The mean ask (bid) is the dark blue (red) line; the blue and red shaded bands represent the 90% range across the simulations for the ask and bid, respectively. The dashed lines plot the competitive ask and bid prices, as discussed in Section 3. The mean market ask and bid quotes in the final day are 101.82 and 98.20, respectively.

Figure C43: Soft-max Exploration - Mean Market Quotes Over Days



The graph shows the distribution of the market ask quotes across the 1,000 repetitions in the final day. The graph refers to the baseline Q-learning setup with soft-max exploration ($\tau = 20$).

Figure C44: Soft-max Exploration - Distribution of Market Ask Prices in Day d = 1,000,000



(a) Ask Q-Table (b) Bid Q-Table

The graph plots the mean ask and bid Q-tables at the end of the simulations, averaging across the 1,000 repetitions. It shows the Q-tables for only a subset of prices, centered around the final mean prices. The graph refers to the baseline Q-learning setup.

Figure C45: Soft-max Exploration - Final Mean Q-Tables

C.11 Decoupled Updating

In this subsection, we consider a variation of the updating rule in which the Q-values on one side of the market are not updated when the trade occurs on the other side. That is, the ask Q-values are only updated if the trader buys the asset or does not trade; the bid Q-values are only updated if the trader sells the asset or does not trade. As noted in Section 4, the results of Sections 6–8 also hold under this alternative formulation; the simulations below support this claim. Appendix C.11.1 presents the baseline Q-learning algorithm (Appendix Figures C46–C48 for price-inelastic and Appendix Figures C49–C51 for price-elastic noise traders), Appendix C.11.2 presents counterfactual updating (Appendix Figures C52–C54 for price-inelastic and Appendix Figures C55–C57 for price-elastic noise traders), and Appendix C.11.3 presents one counterfactual-updating AMM competing against one baseline AMM (Appendix Figures C58 and C59). Decoupling does matter for the algorithm-choice game of Section 9, as shown in Figures 6c and 6f.

C.11.1 Baseline

If market maker i chooses ask $a_i^d = \alpha_j$ and bid price $b_i^d = \beta_k$, then these are updated as follows:

$$q_i^{d+1}(\alpha_j) = \lambda_i^d \left(\left(\alpha_j - v^d \right) \mathbf{1}_{\{\Phi_i^d = -1\}} \right) + \left(1 - \lambda_i^d \right) q_i^d(\alpha_j), \qquad \text{if } X^d \in \{0, 1\}, \tag{55}$$

$$q_i^{d+1}(\alpha_j) = q_i^d(\alpha_j), \qquad \text{if } X^d = -1, \tag{56}$$

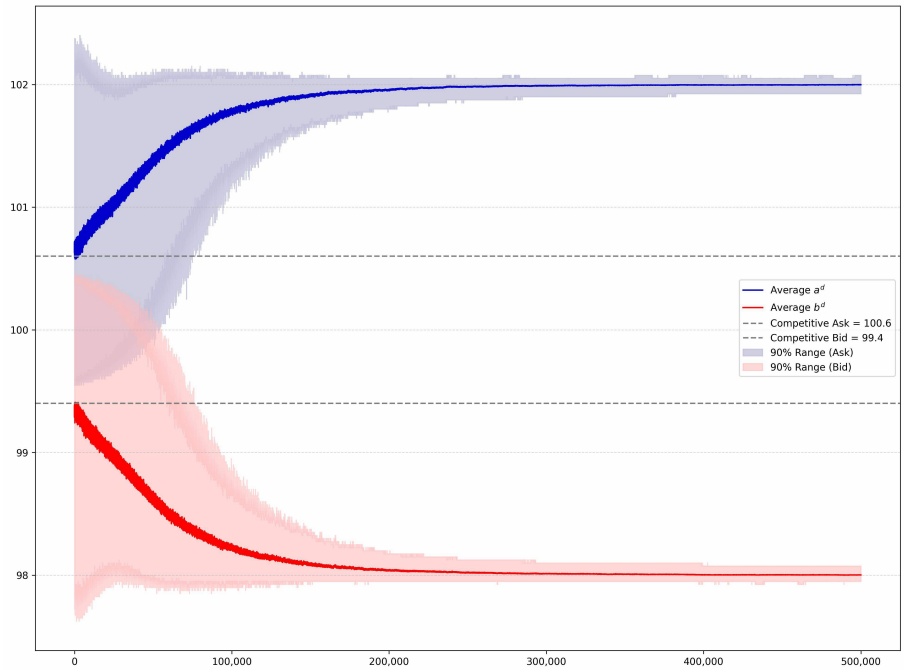
$$\hat{q}_i^{d+1}(\beta_k) = \lambda_i^d \left(\left(v^d - \beta_k \right) \mathbf{1}_{\{\Phi_i^d = 1\}} \right) + \left(1 - \lambda_i^d \right) \hat{q}_i^d(\beta_k), \qquad \text{if } X^d \in \{-1, 0\}, \tag{57}$$

$$\hat{q}_i^{d+1}(\beta_k) = \hat{q}_i^d(\beta_k), \qquad \text{if } X^d = 1. \tag{58}$$

For all other (i.e., non-chosen) ask prices α_l $(l \neq j)$ and bid prices β_m $(m \neq k)$, the Q-values are unchanged:

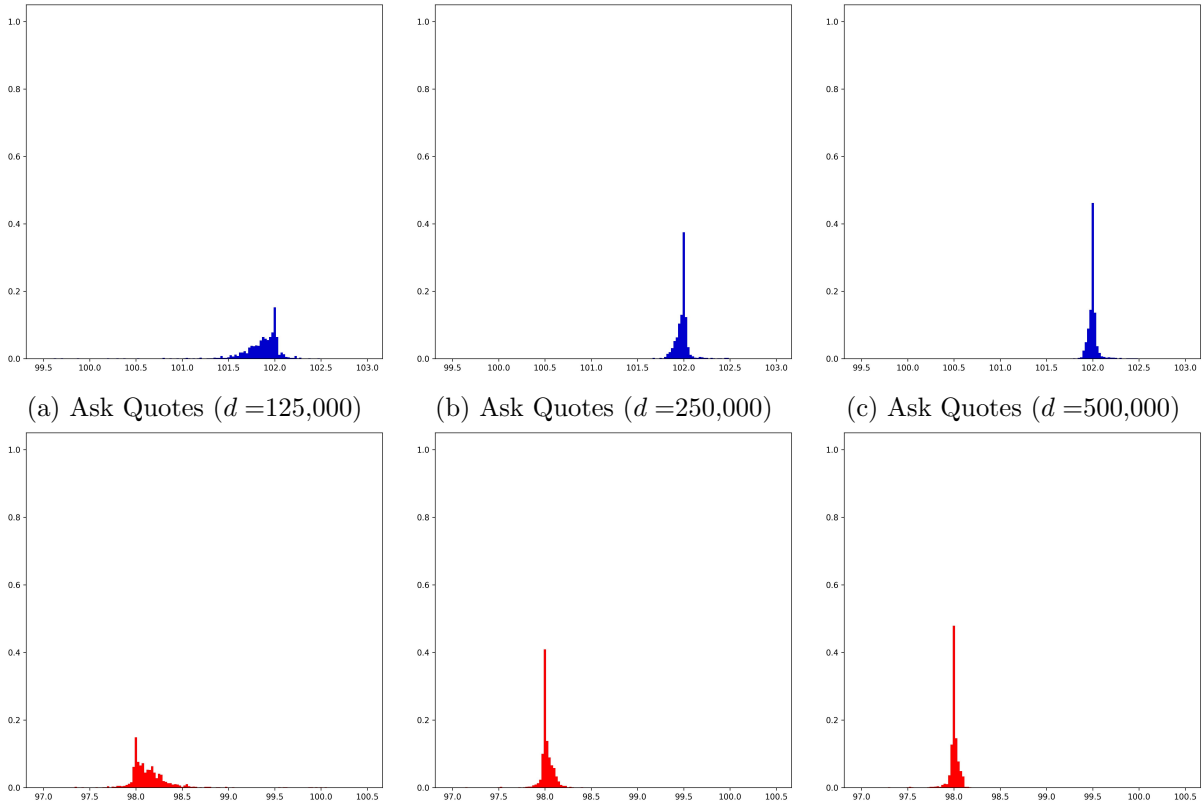
$$q_i^{d+1}(\alpha_l) = q_i^d(\alpha_l), \tag{59}$$

$$\hat{q}_i^{d+1}(\beta_m) = \hat{q}_i^d(\beta_m). \tag{60}$$



The graph plots the mean market quotes over days, averaging across the 1,000 repetitions. The graph refers to the baseline Q-learning setup with price-inelastic noise traders and decoupled updating ($\lambda = 0.1$, $c = 0$). The mean ask (bid) is the dark blue (red) line; the blue and red shaded bands represent the 90% range across the simulations for the ask and bid, respectively. The dashed lines plot the competitive ask and bid prices, as discussed in Section 3. The mean market ask and bid quotes in the final day are 102.0 and 98.0, respectively.

Figure C46: Decoupled Updating (Inelastic), Baseline - Mean Market Quotes Over Days

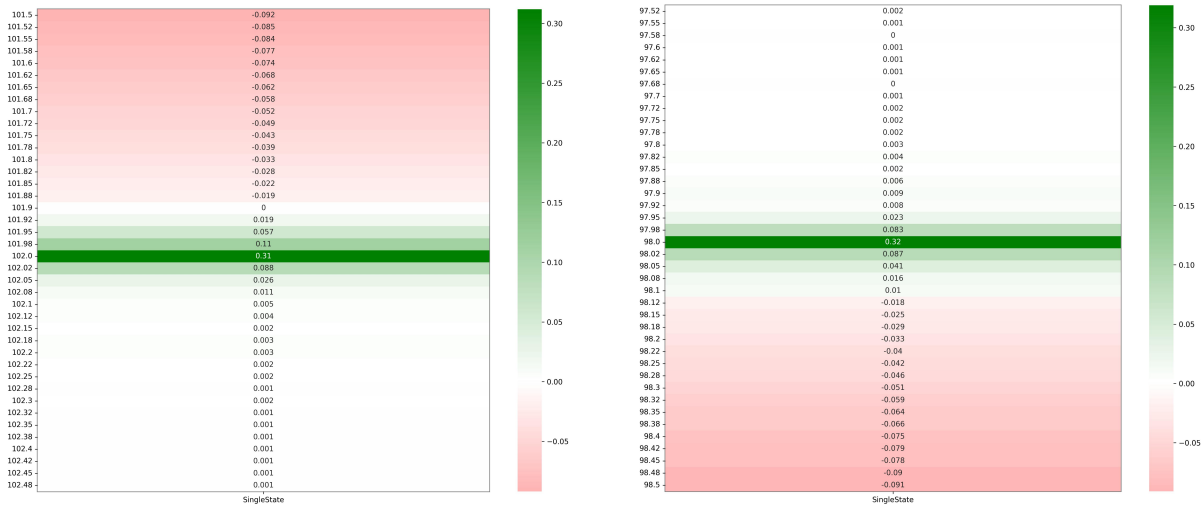


(a) Ask Quotes (d =125,000) (b) Ask Quotes (d =250,000) (c) Ask Quotes (d =500,000)

(d) Bid Quotes (d =125,000) (e) Bid Quotes (d =250,000) (f) Bid Quotes (d =500,000)

The graph plots the distributions of market ask and bid prices across the 1,000 repetitions at different points in the simulation: days 125,000, 250,000, and 500,000. The graph refers to the baseline Q-learning setup with price-inelastic noise traders and decoupled updating ($\lambda = 0.1$, $c = 0$).

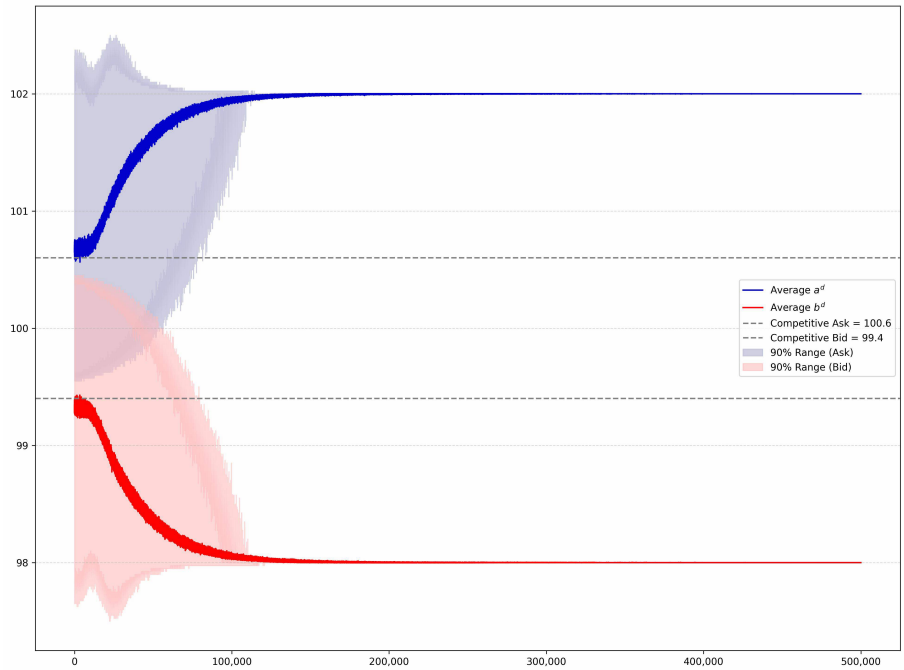
Figure C47: Decoupled Updating (Inelastic), Baseline – Distributions of Market Ask and Bid Quotes over Time



(a) Ask Q-Table (b) Bid Q-Table

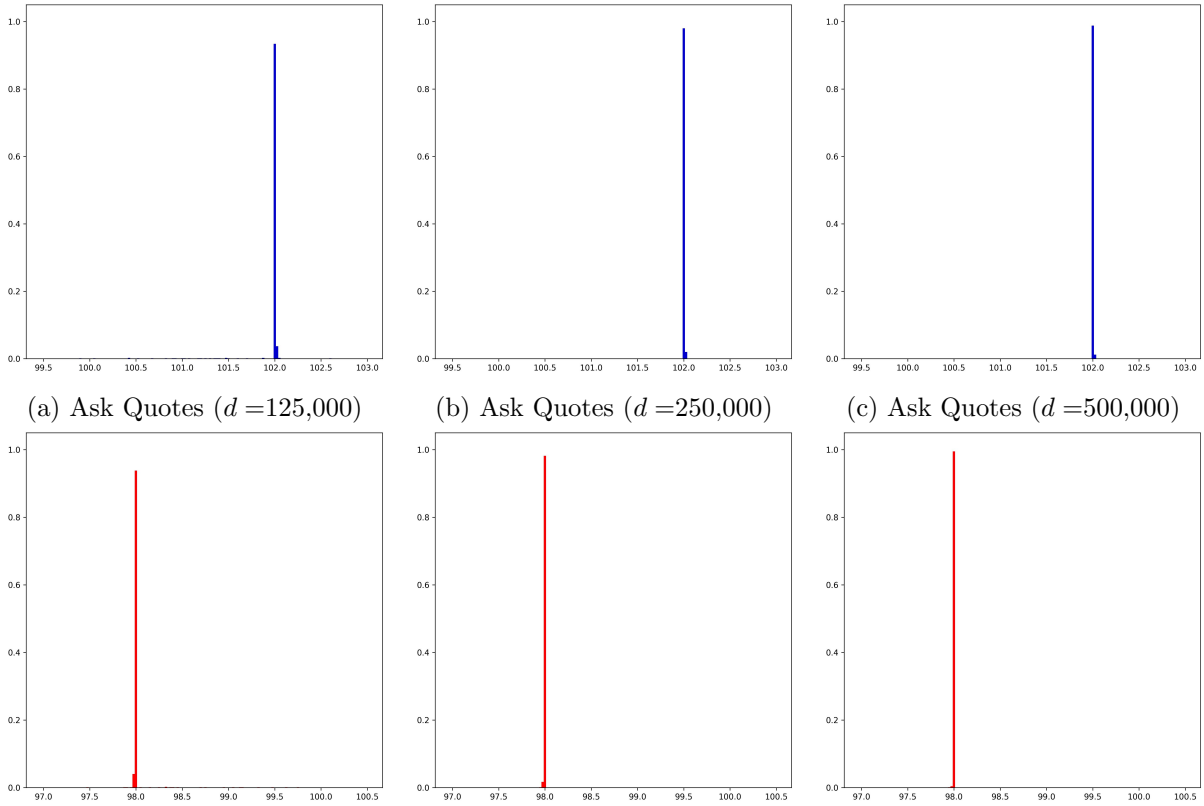
The graph plots the mean ask and bid Q-tables at the end of the simulations, averaging across the 1,000 repetitions. It shows the Q-tables for only a subset of prices, centered around the final mean prices. The graph refers to the baseline Q-learning setup with price-inelastic noise traders and decoupled updating ($\lambda = 0.1$, $c = 0$).

Figure C48: Decoupled Updating (Inelastic), Baseline - Final Mean Q-Tables



The graph plots the mean market quotes over days, averaging across the 1,000 repetitions. The graph refers to the baseline Q-learning setup with price-elastic noise traders and decoupled updating ($\lambda = 0.1$, $c = 0$). The mean ask (bid) is the dark blue (red) line; the blue and red shaded bands represent the 90% range across the simulations for the ask and bid, respectively. The dashed lines plot the competitive ask and bid prices, as discussed in Section 3. The mean market ask and bid quotes in the final day are 102.0 and 98.0, respectively.

Figure C49: Decoupled Updating (Elastic), Baseline - Mean Market Quotes Over Days

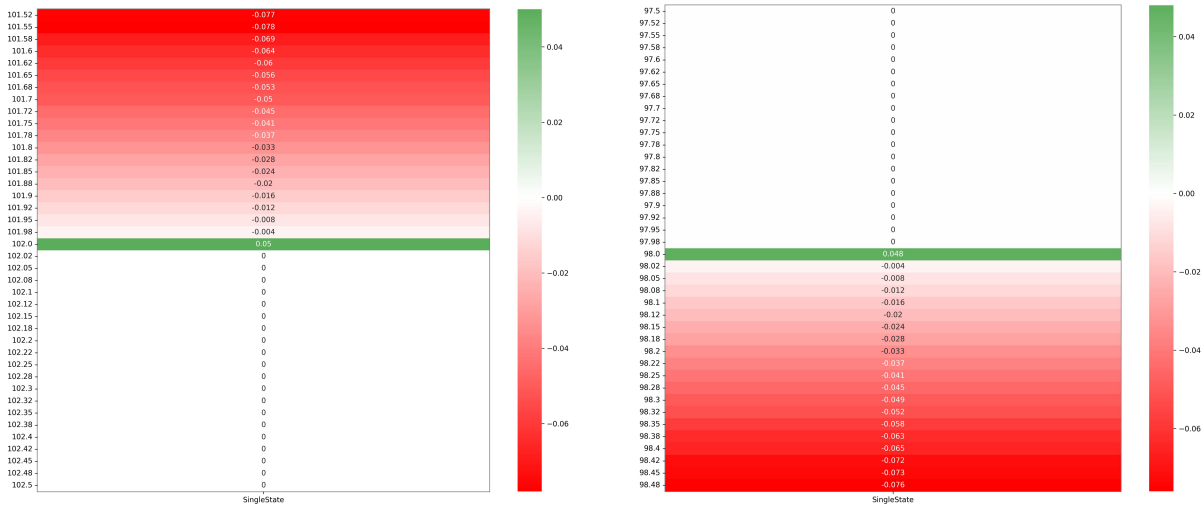


(a) Ask Quotes (d =125,000) (b) Ask Quotes (d =250,000) (c) Ask Quotes (d =500,000)

(d) Bid Quotes (d =125,000) (e) Bid Quotes (d =250,000) (f) Bid Quotes (d =500,000)

The graph plots the distributions of market ask and bid prices across the 1,000 repetitions at different points in the simulation: days 125,000, 250,000, and 500,000. The graph refers to the baseline Q-learning setup with price-elastic noise traders and decoupled updating ($\lambda = 0.1$, $c = 0$).

Figure C50: Decoupled Updating (Elastic), Baseline – Distributions of Market Ask and Bid Quotes over Time



(a) Ask Q-Table (b) Bid Q-Table

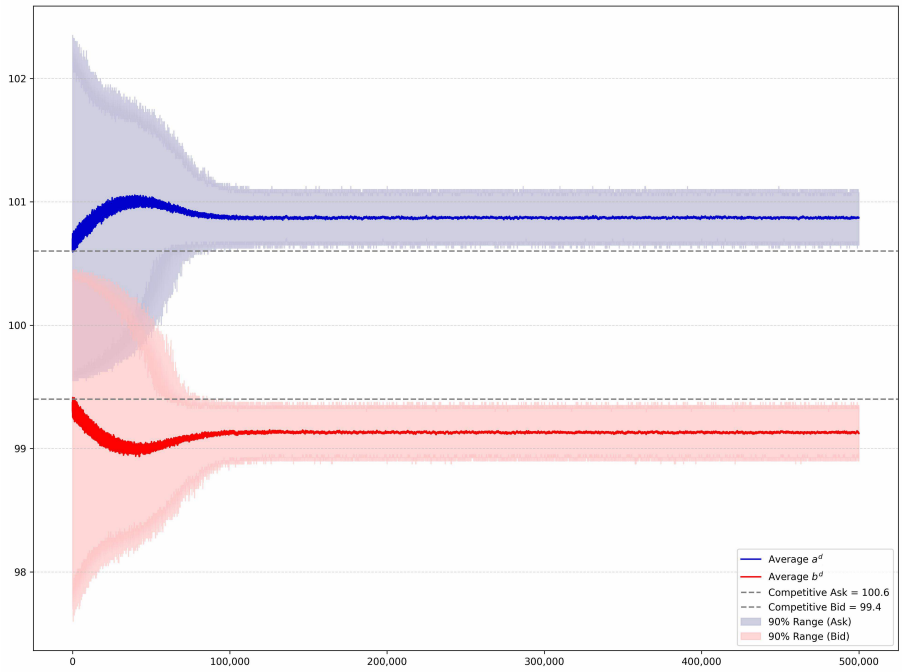
The graph plots the mean ask and bid Q-tables at the end of the simulations, averaging across the 1,000 repetitions. It shows the Q-tables for only a subset of prices, centered around the final mean prices. The graph refers to the baseline Q-learning setup with price-elastic noise traders and decoupled updating ($\lambda = 0.1$, $c = 0$).

Figure C51: Decoupled Updating (Elastic), Baseline - Final Mean Q-Tables

C.11.2 Counterfactual Updating

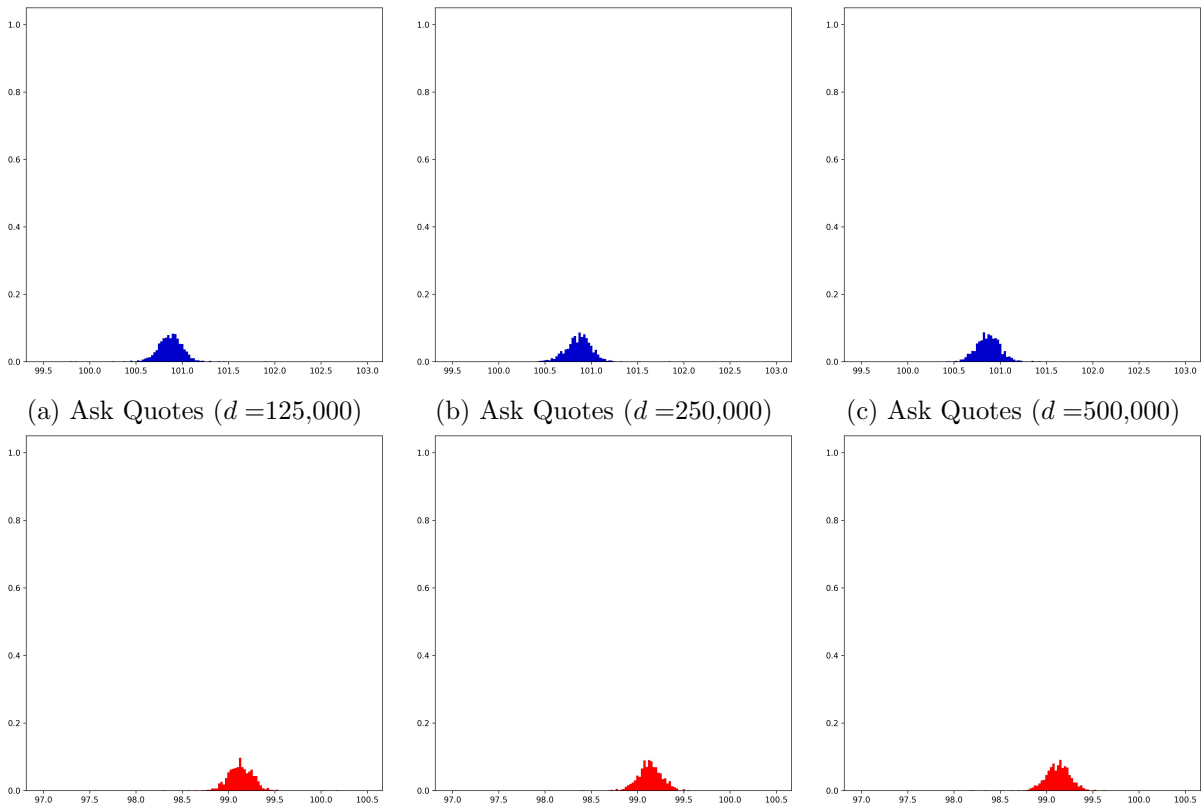
The counterfactual updating in the decoupled version follows the same logic as in Section 7, subject to the updating conditions described in Appendix C.11.1. The ask Q-values are only updated when $X^d \in \{0, 1\}$ and the bid Q-values are only updated when $X^d \in \{-1, 0\}$; when the trade is on the other side of the market, the Q-values on that side are left unchanged.

Below we present the results for the counterfactual weighting of $\rho = 0.6$, as in Section 8, with $\lambda_i = 0.1$, $c_i = 0$, and $D = 500{,}000$, for both price-inelastic and price-elastic noise traders.



The graph plots the mean market quotes over days, averaging across the 1,000 repetitions. The graph refers to the counterfactual-updating Q-learning setup with price-inelastic noise traders and decoupled updating ($\lambda = 0.1$, $c = 0$, $\rho = 0.6$). The mean ask (bid) is the dark blue (red) line; the blue and red shaded bands represent the 90% range across the simulations for the ask and bid, respectively. The dashed lines plot the competitive ask and bid prices, as discussed in Section 3. The mean market ask and bid quotes in the final day are 100.87 and 99.12, respectively.

Figure C52: Decoupled Updating (Inelastic), Counterfactual Updating - Mean Market Quotes Over Days

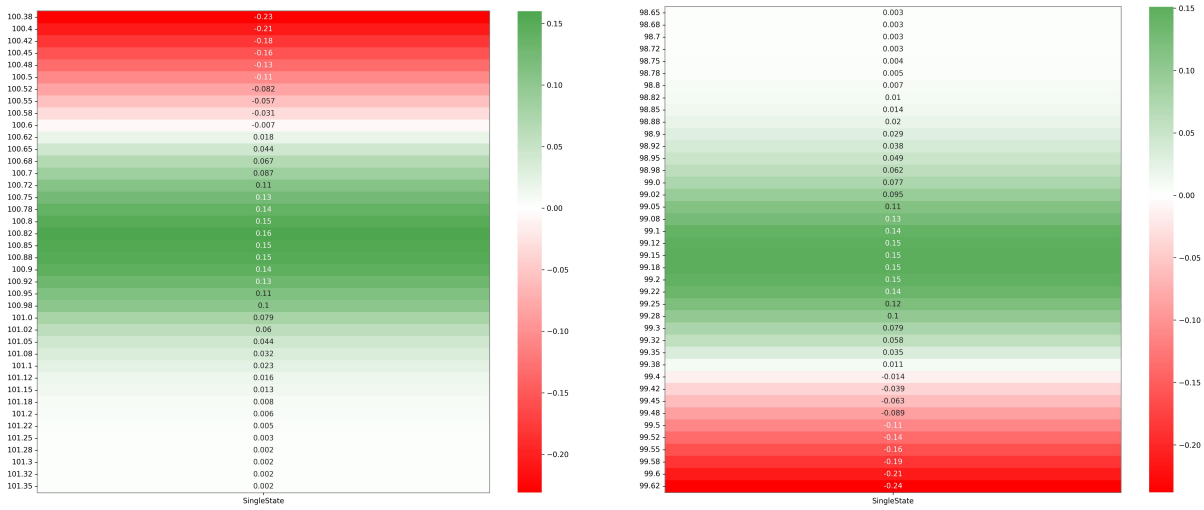


(a) Ask Quotes (d =125,000) (b) Ask Quotes (d =250,000) (c) Ask Quotes (d =500,000)

(d) Bid Quotes (d =125,000) (e) Bid Quotes (d =250,000) (f) Bid Quotes (d =500,000)

The graph plots the distributions of market ask and bid prices across the 1,000 repetitions at different points in the simulation: days 125,000, 250,000, and 500,000. The graph refers to the counterfactual-updating Q-learning setup with price-inelastic noise traders and decoupled updating ($\lambda = 0.1$, $c = 0$, $\rho = 0.6$).

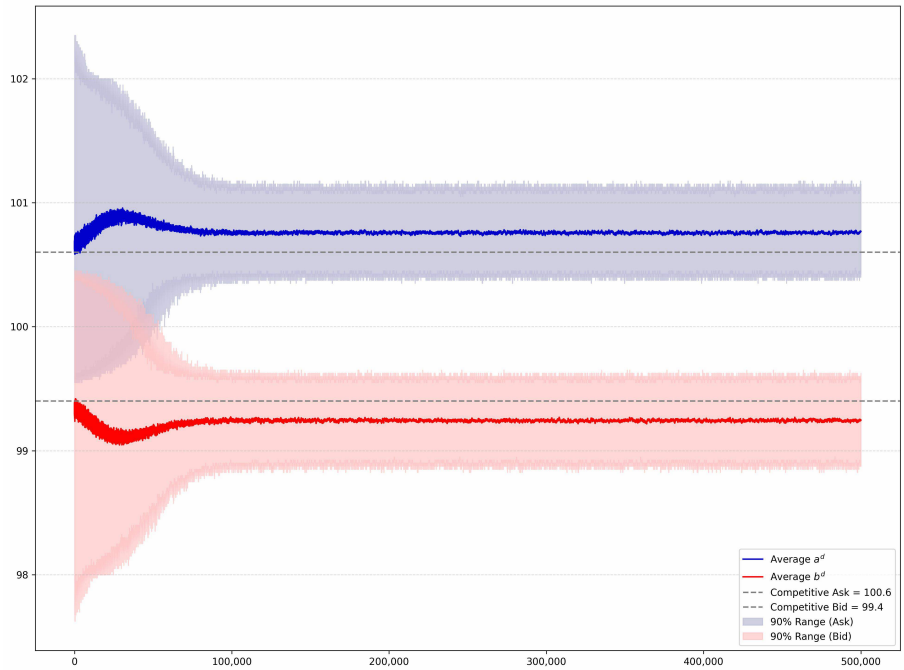
Figure C53: Decoupled Updating (Inelastic), Counterfactual Updating – Distributions of Market Ask and Bid Quotes over Time



(a) Ask Q-Table (b) Bid Q-Table

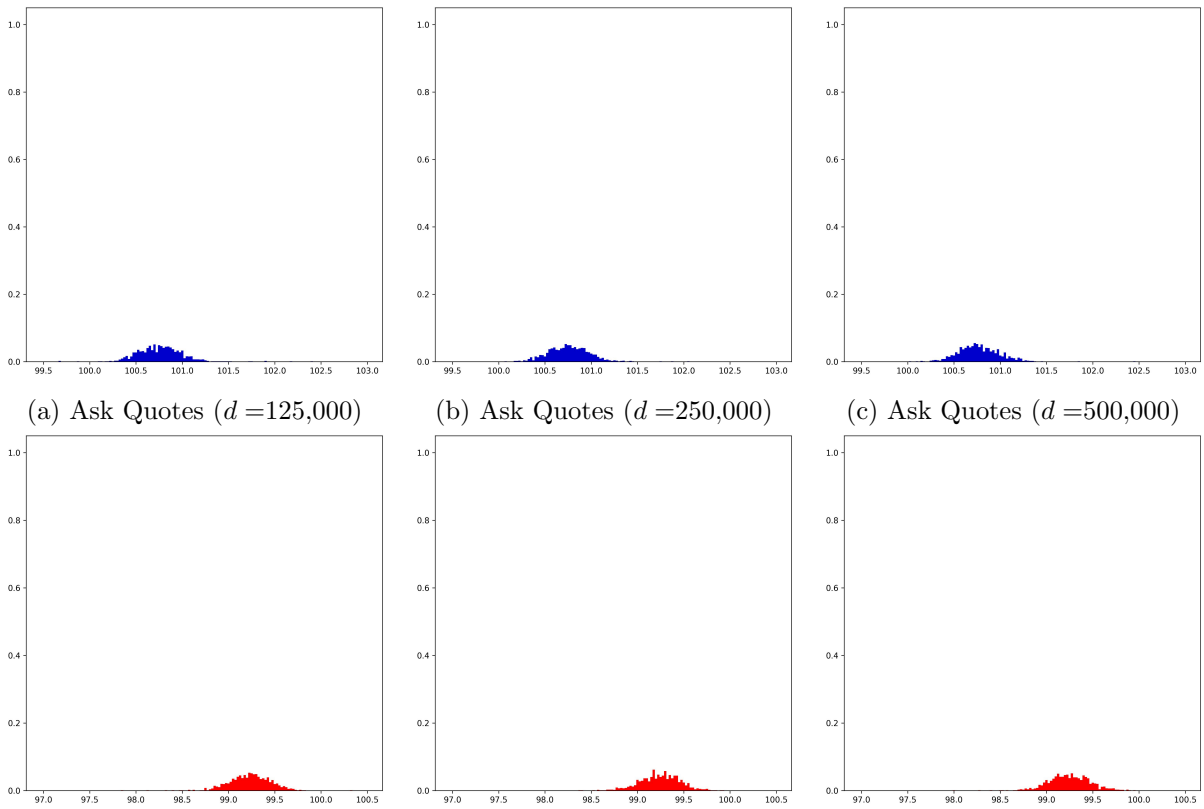
The graph plots the mean ask and bid Q-tables at the end of the simulations, averaging across the 1,000 repetitions. It shows the Q-tables for only a subset of prices, centered around the final mean prices. The graph refers to the counterfactual-updating Q-learning setup with price-inelastic noise traders and decoupled updating ($\lambda = 0.1$, $c = 0$, $\rho = 0.6$).

Figure C54: Decoupled Updating (Inelastic), Counterfactual Updating - Final Mean Q-Tables



The graph plots the mean market quotes over days, averaging across the 1,000 repetitions. The graph refers to the counterfactual-updating Q-learning setup with price-elastic noise traders and decoupled updating ($\lambda = 0.1$, $c = 0$, $\rho = 0.6$). The mean ask (bid) is the dark blue (red) line; the blue and red shaded bands represent the 90% range across the simulations for the ask and bid, respectively. The dashed lines plot the competitive ask and bid prices, as discussed in Section 3. The mean market ask and bid quotes in the final day are 100.77 and 99.25, respectively.

Figure C55: Decoupled Updating (Elastic), Counterfactual Updating - Mean Market Quotes Over Days



(a) Ask Quotes (d =125,000) (b) Ask Quotes (d =250,000) (c) Ask Quotes (d =500,000)

(d) Bid Quotes (d =125,000) (e) Bid Quotes (d =250,000) (f) Bid Quotes (d =500,000)

The graph plots the distributions of market ask and bid prices across the 1,000 repetitions at different points in the simulation: days 125,000, 250,000, and 500,000. The graph refers to the counterfactual-updating Q-learning setup with price-elastic noise traders and decoupled updating ($\lambda = 0.1$, $c = 0$, $\rho = 0.6$).

Figure C56: Decoupled Updating (Elastic), Counterfactual Updating – Distributions of Market Ask and Bid Quotes over Time



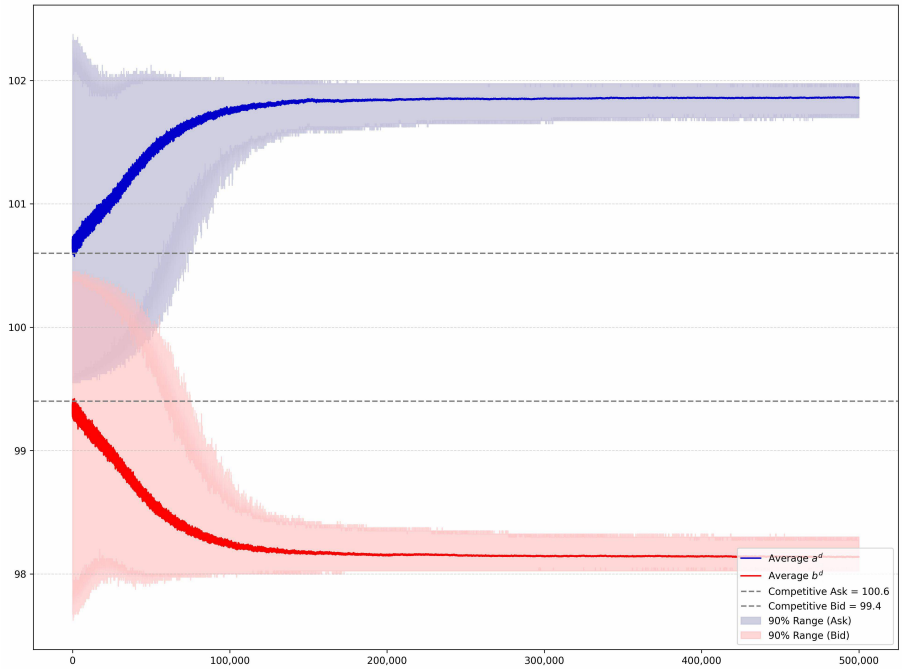
(a) Ask Q-Table (b) Bid Q-Table

The graph plots the mean ask and bid Q-tables at the end of the simulations, averaging across the 1,000 repetitions. It shows the Q-tables for only a subset of prices, centered around the final mean prices. The graph refers to the counterfactual-updating Q-learning setup with price-elastic noise traders and decoupled updating ($\lambda = 0.1$, $c = 0$, $\rho = 0.6$).

Figure C57: Decoupled Updating (Elastic), Counterfactual Updating - Final Mean Q-Tables

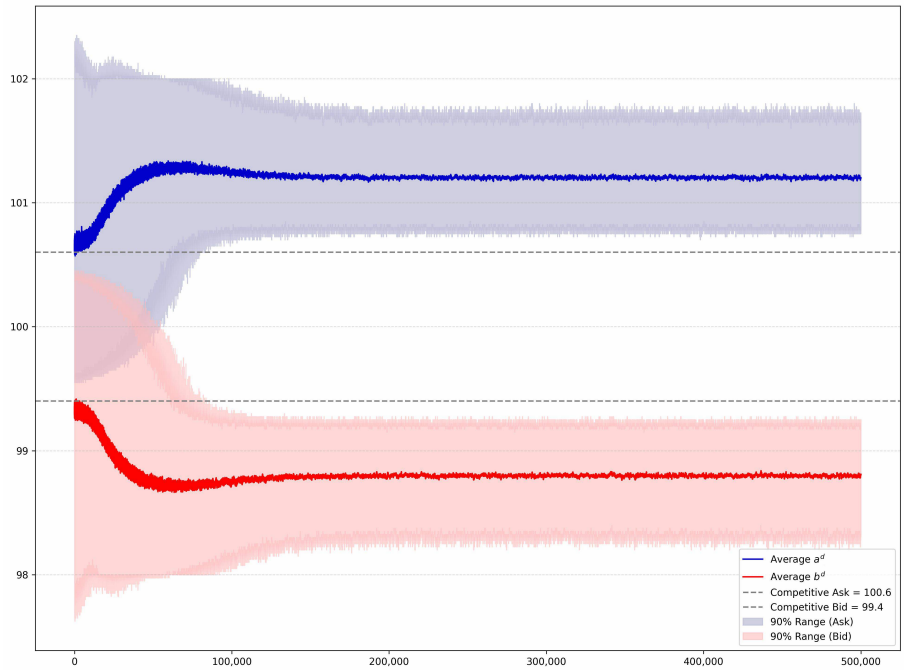
C.11.3 Baseline vs. Counterfactual Updating

We now consider one AMM using counterfactual updating competing against one AMM using the baseline algorithm, both with decoupled updating ($\lambda_i = 0.1$, $c_i = 0$, $\rho = 0.6$, $D = 500{,}000$). These are the simulations behind the off-diagonal entries of Figures 6c and 6f.



The graph refers to one counterfactual-updating AMM (AMM_1) competing against one baseline AMM (AMM_2), with price-inelastic noise traders and decoupled updating ($\lambda = 0.1$, $c = 0$, $\rho = 0.6$). It plots the mean market quotes over days, averaging across the 1,000 repetitions; the mean ask (bid) is the dark blue (red) line, and the blue and red shaded bands represent the 90% range across the simulations. The mean ask and bid quotes in the final day are 101.86 and 98.14, respectively.

Figure C58: Decoupled Updating (Inelastic), Baseline vs. Counterfactual Updating – Mean Quotes over Time



The graph refers to one counterfactual-updating AMM (AMM_1) competing against one baseline AMM (AMM_2), with price-elastic noise traders and decoupled updating ($\lambda = 0.1$, $c = 0$, $\rho = 0.6$). It plots the mean market quotes over days, averaging across the 1,000 repetitions; the mean ask (bid) is the dark blue (red) line, and the blue and red shaded bands represent the 90% range across the simulations. The mean ask and bid quotes in the final day are 101.20 and 98.81, respectively.

Figure C59: Decoupled Updating (Elastic), Baseline vs. Counterfactual Updating – Mean Quotes over Time

D Deep Q-Networks: Architecture, Implementation, and Additional Results

This appendix details the Deep Q-Network (DQN) implementation discussed in Section 10, as well as the experience-replay extension of the tabular Q-learning.

D.1 From Q-table to Q-network

Each market maker i is equipped with two networks, one for the ask side and one for the bid side, mirroring the two Q-vectors q_i^d and $\hat{q}_i^d$ of the tabular specification. Because the environment is stateless, each network receives a constant input $x = 1$ and outputs a vector of $J = 141$ Q-values, one for each grid point $\alpha_j \in \mathbf{A}$ (respectively $\beta_k \in \mathbf{B}$). Writing θ for the network's weights and biases and $\sigma(\cdot) = \max(\cdot, 0)$ for the ReLU activation, the ask network computes, following Mnih et al. (2015),

$$h^{(1)} = \sigma(W_1 x + b_1)\,, \qquad h^{(2)} = \sigma\Big(W_2 h^{(1)} + b_2\Big)\,, \qquad q_i^d(\cdot\,;\theta) = W_3 h^{(2)} + b_3, \tag{61}$$

with $W_1 \in \mathbb{R}^{128\times 1}$, $W_2 \in \mathbb{R}^{128\times 128}$, and $W_3 \in \mathbb{R}^{J\times 128}$; the bid network $\hat{q}_i^d(\cdot\,;\theta)$ is defined analogously. The j-th entry of the output vector, $q_i^d(\alpha_j;\theta)$, replaces the tabular entry $q_i^d(\alpha_j)$: the AMM still selects the greedy ask (bid) as $\arg\max_{\alpha_j} q_i^d(\alpha_j;\theta)$ ($\arg\max_{\beta_k} \hat{q}_i^d(\beta_k;\theta)$) and explores with the same ε-greedy schedule as in the tabular case.

After realizing profit π on the chosen ask α_j, the tabular rule (Equation (13)) moves the corresponding entry directly, $q_i^{d+1}(\alpha_j) = \lambda_i^d \pi + (1-\lambda_i^d) q_i^d(\alpha_j)$, leaving every other entry unchanged. The network instead takes one gradient-descent step, with a fixed step size κ, to move its prediction for α_j toward a target y, minimizing the squared loss

$$\mathcal{L}(\theta) = \big(q_i^d(\alpha_j;\theta) - y\big)^2, \qquad \theta \leftarrow \theta - \kappa\, \nabla_\theta \mathcal{L}(\theta). \tag{62}$$

The general Q-learning target carries a continuation value, $y = \pi + \gamma_Q \max_{\alpha_l} q_i^d(\alpha_l;\theta)$; because the environment is single-period ($\gamma_Q = 0$), this reduces to $y = \pi$, the realized profit itself, exactly as on the right-hand side of the tabular rule.

Unlike the tabular update, which moves only the entry $q_i^d(\alpha_j)$, a single gradient step moves every output of the network, since all outputs depend on the shared parameters θ. Because each update moves more of the Q-function at once, we set the network's step size to be $\kappa = 0.01$, below the tabular learning rate λ_i^d, so that the network is not more reactive to a single realized

profit than the tabular rule.

We initialize all network weights and biases by drawing uniformly from $[0, 0.1]$; since the input $x = 1$ is positive, every layer's output is then positive, so the network starts with strictly positive Q-values throughout, consistent with our tabular initialization convention.[30]

D.2 Experience replay

The update in Equation (62) learns only from the most recent observation. The second innovation of DQN, experience replay (Mnih et al., 2015), instead stores past observations and learns from a random sample of them. For each market maker and each side of the market, we keep a buffer containing the B most recent pairs of posted price and realized profit, (α_j, π), including the zero profits from days without a trade; once the buffer is full, each new observation replaces the oldest one. Every day, after the new observation is added, the market maker draws a batch of b observations from the buffer, uniformly and without replacement, $\{(\alpha_{j_m}, \pi_m)\}_{m=1}^{b}$, and this draw replaces the single-observation update above. The network takes one gradient step on the loss averaged over the batch,

$$\mathcal{L}(\theta) = \frac{1}{b} \sum_{m=1}^{b} \big(q_i^d(\alpha_{j_m}; \theta) - \pi_m\big)^2, \qquad \theta \leftarrow \theta - \kappa \, \nabla_\theta \mathcal{L}(\theta). \tag{63}$$

With $b = B = 1$, the buffer contains only today's observation and Equation (63) collapses to Equation (62); this is the like-for-like specification used whenever we compare the network architecture alone with the tabular rule.

Experience replay with a Q-table. To separate the effect of experience replay from that of the network, we also apply the same replay mechanism to the tabular Q-learning of Section 4, with the same buffer and the same sampling. A batch may contain several observations associated with the same price and, in general, observations associated with several different prices. Let n_j denote the number of sampled observations at price α_j in the batch and $\bar{\pi}_j$ their average realized profit. Every price that appears in the batch is updated toward its sample-average

[30] The default initialization commonly used in the DQN literature ("Kaiming initialization") draws weights from a range centered on zero, which can assign a negative initial value to prices for which negative profit is not economically possible, such as an ask above v_H. We instead follow the same positive-initialization convention that we use for the tabular Q-values.

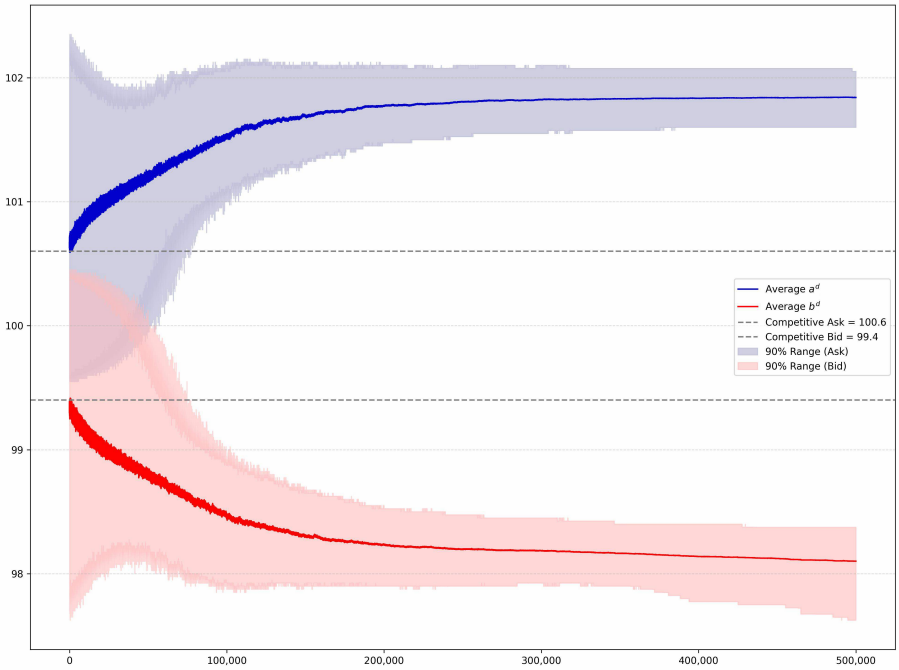
profit,

$$q_i^{d+1}(\alpha_j) = w_j\,\bar{\pi}_j + (1 - w_j)\,q_i^d(\alpha_j), \qquad w_j = \lambda_i^d\,\frac{n_j}{b}, \tag{64}$$

while prices that do not appear in the batch are left unchanged; the bid side is updated analogously. The weight w_j scales the learning rate by the share of the batch that α_j accounts for. This mirrors the DQN update, where the batch loss in Equation (63) accounts for the number of times a price appears within a sample. Since $\sum_j n_j = b$, the weights sum to λ_i^d, so the total weight a market maker places on new information in each update is the same as under the standard rule, regardless of how many distinct prices the batch contains. With $b = B = 1$, Equation (64) reduces to the tabular rule of Equation (13).

D.3 Additional Results: DQN and Experience Replay

Appendix Figure D1 shows the evolution of the mean market quotes when the DQN is used without experience replay ($b = B = 1$). As in the tabular baseline of Section 6, the quotes settle close to the loss-free prices, with final-day mean ask and bid quotes of 101.84 and 98.10, respectively.



The graph plots the mean market quotes over days, averaging across the 1,000 repetitions. The graph refers to the DQN setup described above, with $b = B = 1$ (i.e., without experience replay). The mean ask (bid) is the dark blue (red) line; the blue and red shaded bands represent the 90% range across the simulations for the ask and bid, respectively. The dashed lines plot the competitive ask and bid prices, as discussed in Section 3. The mean market ask and bid quotes in the final day are 101.84 and 98.10, respectively.

Figure D1: DQN Architecture without Experience Replay – Mean Market Ask and Bid Quotes over Time