LIETUVOS BANKAS

WORKING PAPER SERIES

No 6 / 2009

BUILDING AN ARTIFICIAL STOCK MARKET POPULATED BY REINFORCEMENT-LEARNING AGENTS

by Tomas Ramanauskas and Aleksandras Vytautas Rutkauskas

WORKING PAPER SERIES
No 6 / 2009

BUILDING AN ARTIFICIAL STOCK MARKET POPULATED BY REINFORCEMENT-LEARNING AGENTS

by Tomas Ramanauskas[1] and Aleksandras Vytautas Rutkauskas[2]

[1] Bank of Lithuania, Vilnius Gediminas Technical University

[2] Vilnius Gediminas Technical University

For correspondence: Economics Department, Bank of Lithuania
Totorių g. 4, LT-01121 Vilnius, Lithuania; e-mail: tramanauskas@lb.lt

Address
Totorių g. 4
LT-01121 Vilnius
Lithuania
Telephone +370 5 268 0132
Fax +370 5 212 4423

Internet
http://www.lb.lt

Statement of purpose
Working Papers describe research in progress by the author(s) and are published to stimulate discussion and critical comments.

The Series is managed by Economic Research Division of Economics Department

All Working Papers of the Series are refereed by internal and external experts Disclaimer

The views expressed are those of the author(s) and do not necessarily represent those of the Bank of Lithuania.

ISSN 2029-0446 (ONLINE)

Contents

Abstract 4

1. Introduction 5

2. Description of the ASM model 6
   2.1. Motivation 6
   2.2. General market setting and model's main building blocks 7
   2.3. Forecasting dividends 8
   2.4. Estimating fundamental the stock value and reservation prices 9
   2.5. Making individual trading decisions 10
   2.6. Learning and systemic adaptation in the model 14

3. Simulation results 17

4. Conclusions 20

References 22

APPENDIX 1. Reinforcement learning in the context of ASM modelling 23

APPENDIX 2. Parameter setting and experimental results 29

List of Bank of Lithuania Working Papers 35

Abstract

In this paper we propose an artificial stock market model based on interaction of heterogeneous agents whose forward-looking behaviour is driven by the reinforcement learning algorithm combined with some evolutionary selection mechanism. We use the model for the analysis of market self-regulation abilities, market efficiency and determinants of emergent properties of the financial market. Distinctive and novel features of the model include strong emphasis on the economic content of individual decision making, application of the Q-learning algorithm for driving individual behaviour, and rich market setup.

Keywords: Agent-based financial modelling, artificial stock market, complex dynamical system, emergent properties, market efficiency, agent heterogeneity, reinforcement learning.
JEL classification: G10, G11, G14.

Santrauka

Šiame straipsnyje pateikiamas dirbtinės akcijų rinkos modelis, pagrįstas heterogeninių agentų sąveika. Ateities galimybių vertinimu pasižyminčią agentų elgseną lemia skatinamojo mokymosi algoritmas, taikomas kartu su evoliucine agentų atranka. Modelis nėra tiesiogiai taikomas empirinei analizei, jis vertintinas kaip struktūrizuotos analizės pagrindas, tiriant rinkos savireguliacijos galimybes, rinkos efektyvumą bei kylančias rinkos savybes lemiančius veiksnius. Lyginant su daugeliu kitų dirbtinės akcijų rinkos modelių, šiame modelyje ekonominei individų elgsenai ir individualiai adaptacijai skiriamas gerokai didesnis dėmesys. Riboto racionalumo agentai šiame modelyje investicinius sprendimus grindžia ekonomine logika, t.y. vertindami tikėtinus diskontuotus pajamų srautus bei lygindami alternatyvių investicijų grąžas. Jie taip pat siekia tinkamai vertinti ateitį dideliu neapibrėžtumu pasižyminčioje aplinkoje bei atsižvelgia į kitų rinkos dalyvių veiksmų poveikį bendrai rinkos kainos dinamikai. Šis darbas yra vienas pirmųjų bandymų ekonominiu požiūriu įdomų skatinamojo mokymosi algoritmą (konkrečiau, Q-mokymąsi) dirbtinės akcijų rinkos modeliuose. Modelis taipogi pasižymi ganėtinai sofistikuota imitacinės rinkos struktūra.

1. Introduction

In this paper we develop an artificial stock market (ASM) model, which could be used to examine some emergent features of a complex system comprised of a large number of heterogeneous learning agents that interact in a detail-rich and realistically designed environment. This version of the model is not calibrated to empirical data, so at this stage the main aim of this research is to offer, implement and test some new ideas that could lay ground for a robust framework for analysis of financial market processes and their determinants. We believe that the model does offer an interesting framework for the structured analysis of market processes without abstracting from relevant and important features, such as an explicit trading process, regular dividend payouts, trading costs, agent heterogeneity, dissemination of experience, competitive behaviour, agent prevalence and forced exit, etc. Of course, some of these aspects have already been incorporated in existing agent-based financial models. However, the lack of the widely accepted fundament in this area of modelling necessitates the individual and largely independent approach, which is pursued in this study.

One of distinctive features of the proposed agent-based model is a strong emphasis on economic behaviour of individual agents. In the proposed model boundedly rational agents base their decisions on economic considerations, such as estimation of discounted earnings and comparison of returns on different investment strategies, and pursue forward-looking behaviour in highly uncertain environment. Agents' individual adaptation, intertemporal decision making and forward-looking behaviour in the multi-agent setting is governed by reinforcement learning technique borrowed from the field of machine learning. To our knowledge, this work is one of the first attempts to apply the reinforcement learning techniques in an ASM model. Also, this is apparently the first full-fledged artificial stock market model in the Lithuanian economic literature.

By conducting simulation experiments in this model, we aim to address some specific questions, such as market self-regulation abilities, the congruence between the market price of the stock and its fundamentals (the market efficiency issue), importance of intelligent individual behaviour and interaction at the population level for market efficiency and functioning, and relationship between stock prices and market liquidity. It should be stressed, however, that at this stage the model should largely be seen as a thought experiment that proposes to study financial market processes in the light of complex interaction of artificial agents acting an economically appealing way. Nevertheless, the proposed modelling approach serves as a basis for a refined and suitable for empirical analysis version of the model, which is developed in Ramanauskas (2009).

The paper is organised as follows. We provide a detailed description of model's main building blocks and basic internal processes in Section 2. Section 3 describes implemented simulation experiments and discusses results of model simulation in controlled environment. Section 4 concludes. The paper also contains two appendices. In Appendix 1, basic principles of reinforcement learning algorithm (more specifically, Q-learning) are presented. Description of model parameters, experimental settings and

selected simulation graphs are given in Appendix 2. Since the current paper is an integral part of our broader research effort, we do not provide a review of the related literature but rather refer an interested reader to Ramanauskas (2008) for a review of related ASM models.

2. Description of the ASM model

2.1. Motivation

The ASM research area is relatively new but there is a growing body of literature on the subject. There is a clear lack of the comprehensive literature review and classification of existing models. Some popular models and ASM modelling principles are presented in LeBaron (2006), Samanidou et al. (2007) have a review of some agent-based financial models, with the emphasis on econophysics. At the heart of ASM models lies interaction of heterogeneous agents, which leads to complex systemic behaviour and emergent systemic properties. There are two broad classes of ASM models, namely, models based on agents' hard-wired behavioural rules (see, e.g. Kim and Markowitz (1989), Sethi and Franke (1995), Lux (1995)) and models supporting systemic adaptation. The most prominent example of the latter category is the Santa Fe ASM model developed by Arthur et al. (1997); also see, e.g. Beltrati and Margarita (1992), Lettau (1997), LeBaron (2000), Tay and Linn (2001). See Ramanauskas (2009) for a general discussion about agent-based financial modelling and the abovementioned models.

An important caveat of many ASM models is that systemic adaptation often relies merely on evolutionary search algorithms. This means that systemic dynamics, e.g. trading and market price formation is generated by simply ensuring the sufficient variety of investment strategies and inducing some sort of evolutionary selection of strategies in favour to those that give highest utility to individuals. Such approach often downplays the importance of individual behaviour, which is often assumed to be driven by simplistic rules. Also, these algorithms generally do not support forward-looking behaviour except special cases, in which agents try to achieve myopic one-period optimisation. Unlike neoclassical financial theories, most existing agent-based models are not well-suited to model the intertemporal choice and hence miss a crucial aspect of financial decision making.

In our view, agents should exhibit economically interesting behaviour and retain elements of economic reasoning rather than constitute mere collections of behavioural rules Hence, the present ASM model does not fully abstract from many important features of real financial markets that are usually omitted both from standard financial models and other ASMs. For example, just like in the real world financial markets, agents in this ASM do not know the "true model" but try instead to adapt in the highly uncertain environment, they exhibit bounded rationality, non-myopic forward-looking behaviour, as well as diversity in experience and skill levels; the trading process is quite realistic and detailed; dividends are paid out in discrete time intervals and the importance of dividends

as a fundamental force driving stock prices is explicitly recognised. In this section we present the architecture of the artificial stock market in detail.

2.2. General market setting and model's main building blocks

The artificial stock market is populated by a large number of heterogeneous reinforcement-learning investors. Investors differ in their financial holdings, expectations regarding dividend prospects or fundamental stock value. This ensures diverse investor behaviour even though the basic principles governing experience accumulation are the same across population. We can summarise agents' basic behavioural principles as follows. All agents forecast an exogenously given, unknown dividend process and base their estimates of the fundamental stock value on dividend prospects. These estimates are intelligently adjusted to attain immediate reservation prices. Agents explore the environment and accumulate the experience with the aim of maximising long-term returns on their investment portfolios but there are no optimality guaranties against the background of high uncertainty and complex interaction of agents.

Figure 1. Main building blocks of the ASM model

Forming private forecasts of exogenously generated dividends Based on: • Exponential moving average • Adjustment as a result of reinforcement learning (agents seek to minimise forecast errors)
Making individual estimates of fundamental stock value and its reservation price Based on: • Discounted expected dividend flows • Adjustment as a result of reinforcement learning (agents seek to maximise portfolio returns)
Making individual trading decisions Based on: • Private estimates of fundamentals, • Maximisation of expected individual wealth at the end of a trading period • Publicly announced estimated probabilities of successful trades for given prices
Carrying out trades via the centralised exchange and collecting trading statistics Based on: • Double auction system • Simultaneous submission of trade orders and random queuing of individual orders
Learning to forecast dividends and learning about fundamental stock value Based on: • Standard Q-learning with linear gradient-descent approximation
Augmenting learning processes by specific interaction among agents (optional) Based on: • Successful strategy imitation • Evolutionary selection and resultant prevalence of successful investment strategies • Noise trading behaviour

As usual in financial market modelling, the modelled financial market is very simple. Only one, dividend-paying stock (stock index) is traded on the market. Dividends are generated by an exogenous stochastic process unknown to the agents, and they are paid out in regular intervals. The number of trading rounds between dividend payouts can be set arbitrarily, which enables interpretation of a trading round as a day, a week, a month, etc. Paid out dividends and funds needed for liquidity purposes are held in private bank accounts and earn constant interest rates, whereas liquidity exceeding some arbitrary threshold is simply removed from the system (e.g., consumed). Borrowing is not allowed. Initially agents are endowed with arbitrary stock and cash holdings, and subsequently in every trading round each of them may submit a limit order to buy or sell one unit of stock, provided, of course, that financial constraints are non-binding. Trading takes place via the centralised exchange.

For the ease of detailed model exposition, it is useful to break the model into a set of economically meaningful processes, though some of them are inter-related in complex ways. The general structure of the model is laid out in Figure 1. We will discuss these logical building blocks in the following subsections.

2.3. Forecasting dividends

Expected company earnings and dividend payouts are the main fundamental determinants of the intrinsic stock value. Even though in standard models based on the efficient market hypothesis corporate earnings and dividend dynamics are not forecasted explicitly, it is usually implicitly assumed that some market players do conduct fundamental analysis, which ultimately gets reflected in stock prices. Hence, the fundamental analysis of earnings perspectives does matter. It is only that some theories are willing to go so far as to assume that communication among market participants is efficient enough for most investors not to bother inquiring into companies' financial books.

Here we propose the view that in the uncertain environment investors (*i*) form their individual beliefs about the risk-neutral value of a risky stock as some basic value anchor, (*ii*) acknowledge that the market price of the stock may fluctuate about or systemically differ from individual risk-neutral fundamentals due to various factors, such as investors' risk preferences, animal spirits or heterogeneity of beliefs, and (*iii*) flexibly determine their individual reservation prices in the process of adaptive interaction with the environment. The inertia of beliefs about future prospects, as well as the entirety of individual incentives and reward structures then determine market's aggregate attitude toward risk and, consequently, result in episodes of market euphoria or pessimism.

We assume that all agents make their private forecasts of dividend dynamics. Dividend flows generated by an unknown, potentially non-stationary data generating process specified by a modeller. The only information, upon which agents can base their forecasts, is past realisation of dividends, and agents know nothing about stationarity of the data generating process. Hence, they are assumed to form adaptive expectations, augmented with the reinforcement learning calibration. We also allow for possibility to

improve a given agent's forecasting ability by probabilistic imitation of more successful individuals' behaviour (see Section 2.6 for more on this).

Agents start with determining basic reference points for their dividend forecasts. The exponentially weighted moving average (EWMA) of realised dividend payouts can be calculated as follows:

$$d_{i,y}^{EWMA} = \lambda_1 \cdot d_y + (1-\lambda_1) d_{i,y-1}^{EWMA}. \quad (1)$$

Here d_y denotes dividends paid out in period y (year) and λ_1 is the arbitrary smoothing factor (the same for all agents), which is a real number between 0 and 1. The subscript i on the averaged dividends in equation (1) to indicates that they vary across the population of agents. The differences arise due to different arbitrarily chosen initial values but over time, however, these exponential averages converge to each other. Also note that dividend payouts can be arbitrarily less frequent than stock trading rounds, e.g. if one trading period equals one month, dividends may be scheduled to be paid out every twelve periods and in equation (1) one time unit would be one year.

Exponential moving averages would clearly be unacceptable estimates of future dividends in a general case. Hence, their function in this model is twofold. First, they provide a basis for further "intelligent" refinement of dividend forecasts, i.e. these moving averages are multiplied by some adjustment factors calibrated in the process of the reinforcement learning. And second, forecasting dividends relative to their moving averages, as opposed to forecasting dividend levels directly, makes forecasting environment more stationary, which facilitates the reinforcement learning task.

The n-period dividend forecast is given by the following equation:

$$E(d_{i,y+n}) = d_{i,y}^{EWMA} \cdot a_{i,y}^{div}, \quad (2)$$

where $a_{i,y}$ is agent i's dividend adjustment factor. These adjustment factors are gradually changed as agents explore and exploit their accumulated experience, with the long-term aim to minimize squared forecast errors. The detailed description of the reinforcement learning procedure is provided in Section 2.6 and Appendix 1. Individual forecasts for periods $y + 1, \ldots, y + n$ formed in periods $y - n + 1, \ldots, y$, respectively, are stored in the program and used for determining individual estimates of the fundamental stock value.

2.4. Estimating fundamental the stock value and reservation prices

Quite similarly to the dividend forecasting procedure, agents' estimation of the intrinsic stock value is a two-stage process. It embraces formation of initial estimates of the fundamental value, based on discounted dividend flows, and ensuing intelligent adjustment grounded on agents' interaction with environment. We refer to this refined fundamental value as the reservation price.

The initial evaluation of the future dividend flows is a simple discounting exercise. To calculate the present value of expected dividend stream, the constant interest rate is used as the discount factor. Moreover, beyond the forecast horizon dividends are assumed to remain constant. Under these assumptions, individual estimates of the present value of expected dividend flows are

$$v_{i,y}^{fund} = d_y + E\left(\frac{d_{i,y+1}}{1+\bar{r}} + ... + \frac{d_{i,y+n}}{(1+\bar{r})^n} + \frac{d_{i,y+n}/\bar{r}}{(1+\bar{r})^{n+1}}\right), \quad (3)$$

where $\bar{r}$ is the constant interest rate. The last term in this equation is simply the discounted value of the infinite sum of steady financial inflows. These present value estimates are subject to further refinement.

To avoid excessive volatility of the estimates of the discounted value of dividend stream, they are again smoothed by calculating the exponentially weighted moving averages:

$$v_{i,y}^{EWMA} = \lambda_2 \cdot v_{i,y}^{fund} + (1-\lambda_2) v_{i,y-1}^{EWMA}. \quad (4)$$

The role of these averages is very similar to that of the averaged dividends in the dividend forecasting process, namely, to provide some background for the reinforcement learning procedure and (partially) stationarise the environment in which agents try to adapt.

The second stage in the estimation of the individual reservation prices of the stock is calibration based on the reinforcement learning procedure. For this we have to switch to the different time frame (in the base version of the model it is assumed that dividends are paid out annually, whereas agents can trade once per month). In a given trading round *t*, individual reservation prices $v_{i,t}^{reserve}$ are obtained from equation (4) by multiplying exponentially smoothed estimates of fundamental value by individual price adjustment factors, $a_{i,t}^{p}$:

$$v_{i,t}^{reserve} = v_{i,t}^{EWMA} \cdot a_{i,t}^{p}. \quad (5)$$

In this context the individual reservation price is understood as an agent's subjective assessment of the stock's intrinsic value that prompts immediate agent's response (to buy or sell the security).

2.5. Making individual trading decisions

Having formed their individual beliefs about the fundamental value of the stock price, agents have to make specific portfolio rebalancing decisions. In principle, they weigh their own assessment of the stock against market perceptions and make orders to buy (sell) one unit of the underpriced (overpriced) stock at the price that is expected to maximise their wealth at the end of the trading period. We give a more detailed description of these processes below.

The individual reservation price reflects what investors think the stock price should be worth. If the last period's average market price p_{t-1} is less than agent *i*'s reservation price today, it is willing to buy stock and pay at most $v_{i,t}^{reserve}$. Conversely, if the prevailing market price is higher than the agent's perceived fundamental, it is willing to sell it at $v_{i,t}^{reserve}$ or higher price. So its decision rule is like this:

If $v_{i,t}^{reserve} > p_{t-1}$ *and* $m_{i,t}^{0}$ *is sufficient* $\rightarrow$ *submit limit order to buy 1 share at price* $p_{i,t}^{q}$

if $v_{i,t}^{reserve} < p_{t-1}$ *and* $h_{i,t}^{0} > 0 \rightarrow$ *submit limit order to sell 1 share at price* $p_{i,t}^{q}$
otherwise, make no order.

Here $h_{i,t}^{0}$ and $m_{i,t}^{0}$ denote, respectively, agent *i*'s stock holdings (i.e. number of owned shares) and cash balance at the beginning of a trading round, $p_{i,t}^{q}$ is the quoted price to be determined below.

We would not expect real world investors to make orders to buy or sell the stock precisely at reservation prices because in that case they would miss potentially profitable asset allocation opportunities. The real world investor whose perception of the stock value considerably differs from the average market opinion is likely to take advantage of market liquidity and make an order to trade at a price close the prevailing market price rather than to his own reservation price. But what price would it be? There is no answer in the theory. The first obvious step, implemented in the model, is to allow limit orders, i.e. orders to trade the security at a specified *or* better price. Given the complexity of the agent interaction, the optimal pricing solution generally cannot be found. Thus we proceed in the following, intuitively appealing way: (*i*) we determine the possible price quote grid around the prevailing market price (i.e. determine tick sizes and possible price fluctuation bands), (*ii*) estimate aggregate supply and demand schedules, (*iii*) compute each individual's expected end-of-period wealth for every possible trading price and (*iv*) allow agents to make trading decisions that maximise their expected end-of-period wealth.

Agents, of course, aim at getting most favourable prices for their trades but they must take into account the fact that better bid or ask prices are generally associated with smaller probabilities of successful trades. The assumption that each agent is allowed to trade only one unit of stock in a given trading round has a very useful implication in this context – the probabilities of successful trades at all possible prices faced by a buyer and a seller can be loosely interpreted as the supply and demand schedules, respectively. So we further assume that these supply and demand schedules are estimated by the exchange institution from past trading data and constitute public knowledge.

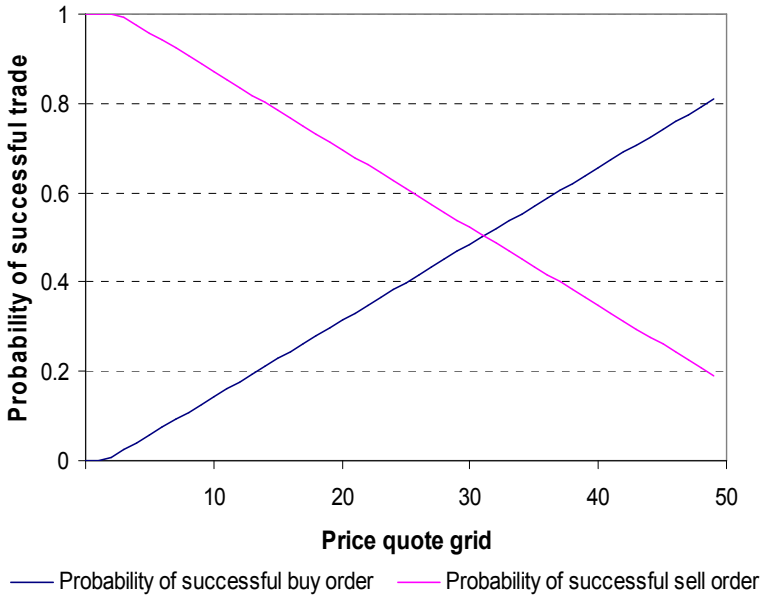
Estimated probabilities of successful trades at given (relative) price quotes are computed as follows. Simply put, these estimated probabilities should indicate chances of successful trading at prices that are "high" or "low" relative to the prevailing market price (i.e. last period's average price). So the probability of the successful trade for a given price quote (relative to the benchmark price) is calculated from the past trading rounds as a fraction of successfully filled buy (sell) orders out of all submitted orders to buy (sell) at that price. Unfortunately, due to computational constraints the number of agents and successful trades is not sufficiently high to obtain reliable estimated probabilities in this straightforward way. For this reason we employ the following three-step procedure:

i) estimates of probabilities of successful buy and sell orders for every price quote are smoothed over time by computing exponential moving averages;

ii) if there are no orders to buy or sell at a given price at time t, the exponential moving average estimates of successful trade probabilities are left unchanged from the t–1 period

iii) the scattered estimates are fitted to a simple cross-sectional regression line (with its values restricted to lie in the interval between 0 and 1) to ensure that the sets of successful trade probabilities retain meaningful economic properties.

Figure 2. Typical estimated demand and supply schedules in an upward-moving market



As a result, we get a nice upward-sloping line, which represents probabilities of successful buy orders for each possible price quote, and a downward-sloping line for the sell orders case. Figure 2 shows a typical example of estimated probabilities of successfully buying and selling one unit of stock at all possible prices (last period's average price set equal to 25 in this relative pricing grid). This particular example reflects an upward-trending market, in which agents reckon they have higher chances (estimated at around 60%) of selling the stock than buying it (estimated at around 40%) at the last period's average price.

At this stage agents have all the components needed to choose prices that give them highest expected wealth at the end of the trading round. First, agent i estimates its expected end-of-period stock holdings (i.e. the number of shares) for each possible price quote j:

$$E(h^1_{i,j,t}) = h^0_{i,t} + E(q_{i,j,t}) \cdot b_i \qquad \text{for all } j. \tag{6}$$

Here $E(q_{i,j,t})$ denotes of expected number of shares to be bought or sold by agent i at any quotable price j (as was explained above, these numbers lie in the closed interval between 0 and 1). The indicator variable b_i takes value of 1 if the agent is willing to buy the stock or -1 if it is willing to sell the stock.

Similarly, agent i's expected end-of-period cash holdings for each possible price quote j are

$$E(m^1_{i,j,t}) = m^0_{i,t} + E(q_{i,j,t}) \cdot x_{j,t} \cdot (-b_i - c) + E(h^1_{i,j,t}) \cdot E(d_{i,t}) \quad \text{for all } j. \tag{7}$$

Here $x_{j,t}$ denotes possible price quote j, c is the fractional trading cost and $E(d_{i,t})$ denotes the expected dividends, which are to be paid out following the trading round (this term equals zero in between the dividend payout periods). It is important to note here that the interest on spare cash funds is paid, as well as excess liquidity (cash holdings above some prespecified amount needed for trading) is taken away, at the beginning of the trading period. All of this is reflected in $m^0_{i,t}$. Dividends are paid out for those agents that hold stocks after the trading round, as can be seen from equation (7).

Finally, agent i's expected end-of-period stock holdings are valued at the individual reservation price and each agent calculates its expected end-of-period wealth for every possible price quote:

$$E(w^1_{i,j,t}) = E(h^1_{i,j,t}) \cdot v^{reserve}_{i,t} + E(m^1_{i,j,t}). \tag{8}$$

Hence, agent i's quoted price, p^q_i, is the price that is associated with the highest expected wealth at the end of the trading round:

$$p^q_i = \arg\max_{x_i} E(\vec{w}^1_{i,t}). \tag{9}$$

If several price quotes result in the same expected wealth, the agent chooses randomly among them. It is also important to note that in the process of the reinforcement learning, agents are occasionally forced to take exploratory actions. In those cases exploring agents choose prices from the quote grid in a random manner.

Market price determination and actual trading take place on the centralised stock exchange. The trading mechanism basically is the double auction system, in which both buyers and sellers contemporaneously submit their competitive orders to implement their trades. Agents are assumed to have no knowledge of individual market participants' submitted orders.

In this model the order book mechanism works as follows. Prior to a trading round, all agents' trade orders are queued randomly and then each of them undergoes the processing procedure. During this procedure, for an order that is being processed all earlier-queued orders are scanned in search for the most favourable matching (opposite) order. If such an order is found (a tie among several equally good orders is broken arbitrarily), the trade is executed at the average of the bid and ask price. Otherwise, the order remains open until it makes a match for other subsequently processed orders or until the end of the trading period, when it is closed as an unexecuted order. Following the trading round, all agents' cash and securities accounts are updated accordingly.

The centralised stock exchange also produces a number of trading statistics, both for analytical and computational purposes. These statistics include the market price, trading volumes and volatility measures. The market price in a given trading period is calculated as the average traded price. As was mentioned before, it is crucially important for making further trading decisions and it serves as the reference value in the subsequent trading round.

2.6. Learning and systemic adaptation in the model

Let us now turn to the learning process through which individual agents' pricing considerations, attitudes to risk and, more generally, goal-oriented behaviours are determined. Quite some learning methods are known, ranging from psychology-based models (stimulus-response, belief-based conscious learning, associative learning, etc.) to rationality-based methods (Beyesian, least-squares learning) to artificial intelligence approaches (evolutionary algorithms, replicator dynamics, neural nets, reinforcement learning). For an overview of popular learning algorithms see, e.g., Brenner (2006). As Brenner notes, virtually all of the learning models used in economic contexts are largely ad hoc, based only on introspection, common sense, artificial intelligence research or psychological findings.

We assume that agents' behaviour is driven by reinforcement learning since these learning algorithms borrowed from the machine learning literature seem to be conceptually suitable for modelling investor behaviour. Agents take actions in the uncertain environment and obtain immediate rewards associated with these (and possibly previous) actions. A specific learning algorithm allows agents to adjust their action policies in pursuit of highest long-term rewards. It is a very desirable feature of any financial model that agents strive for strategic, as opposed to myopic, behaviour. The reinforcement-learning agents do just that. On the other hand, it is the immense complexity of investors' interaction, both in real world financial markets and in the model, that dramatically limits agents' abilities to actually achieve optimal investment policies if not makes the optimal investment behaviour outright impossible.

In our model we use a popular reinforcement learning algorithm, also known as the Q-learning, which was initially proposed by Watkins (1989). It is the temporal difference learning based on the step-wise update (or back-up) of the action-value function and associated adjustment of behavioural policies (a more detailed exposition of basic Q-learning principles is given in Appendix 1). The principal back-up rule is closely related to Bellman optimality property and takes the following form:

$$Q(s_t,a_t) \leftarrow (1-\alpha) \underbrace{\cdot Q(s_t,a_t)}_{\text{Old estimate of } Q(s_t,a_t)} + \alpha \underbrace{(r_{t+1} + \gamma \max_a Q(s_{t+1},a))}_{\text{New estimate of } Q(s_t,a_t)}. \quad (10)$$

Here s_t denotes the state of environment, a_t is the action taken in period *t* and r_{t+1} is the immediate reward associated with action a_t (and possibly earlier actions). Parameter α is known as the learning rate and γ is the discount rate of future rewards. Function $Q(s_t,a_t)$ is usually referred to as the action-value function (or Q-function) and it basically shows the value of taking action a_t in state s_t under behavioural policy π. More specifically, the action-value function is the expected cumulative reward conditional on the current state, action and pursued behavioural policy.

However, the so-called "curse of dimensionality" implies that the straightforward implementation of the basic version of this algorithm is rarely possible in complicated environments. Following the standard practice, we apply the Q-learning algorithm with

gradient-descent approximation, which is briefly presented in Appendix 1. Here we only describe specific variables that are used in the Q-learning algorithm.

As was mentioned before, there are two instances of individual agent learning in the model: learning to forecast dividends and learning to adjust perceived fundamentals. In the dividend forecasting case agent i learns to adjust the dividend adjustment factor, $a_{i,t}^{div}$ (see equation (2)). In each state there are three possible actions – the agent can increase the dividend adjustment factor by a small proportion specified by the modeller, decrease it by the same amount or leave it unchanged.

Due to the complex nature of environment, the state of the world – as perceived by investor i – must be approximated, and it is described by a vector of so-called state features, $\vec{\phi}_s$ (see Figure A1.2 in Appendix 1). We choose four state features that are indicative of the reinforcement learner's "location" in the environment and summarize some properties of the dividend-generating process, which can provide basis for successful forecasting. These features include the size of the dividend adjustment factor, relative deviation of current dividend from its EWMA (compared to the standard deviation), the square of this deviation (to allow for nonlinear relation with forecasts) and the size of the current dividend relative to the EWMA.

The forecast decision is taken at time y and the actual dividend realisation is known at forecast horizon $y + n$. Then agent i gets the reward, which is the negative of the squared forecast error:

$$r_{i,y+n}^{d} = -\left(d_{y+n} - E_y(d_{i,y+n})\right)^2. \qquad (11)$$

Hence, the agent is punished for the forecasting errors. The learning process is augmented by modeller-imposed constraints on dividend forecasts. The forecast is not allowed to deviate by more than a prespecified threshold (e.g. 30%) from the current level of dividends. In that case, the agent gets extra-punishment and the dividend forecast is forced to be marginally closer to the current dividend level. Once the agent observes the resultant state, i.e. the actual dividend realisation, it updates its behavioural policy according to the Q-learning procedure.

In the case of the individual stock value estimation, agent i also can take one of three actions: fractionally increase or decrease the price adjustment factor, $a_{i,t}^{p}$ (see equation (5)), or leave it unchanged. Analogously to the dividend forecasting case, the four state features are the price adjustment factor, the stock price deviation from its exponential time-average (this difference is divided by the standard deviation), the square of this deviation and the current stock price divided by the weighted time-average.

The agent observes the state of the world and acts according to the pursued policy. After the trading round, the agent observes trading results and the resultant state of the world, which enables the agent to update its policies according to the usual Q-learning procedure. In this model, the basic immediate reward, $r_{i,t+1}^{p}$, is simply the log-return on the agent's portfolio:

$$r_{i,t+1}^{p} = \ln\left(h_{i,t}^{1} p_t + m_{i,t}^{1}(1 + \bar{r}^{monthly})\right) - \ln\left(h_{i,t}^{0} p_{t-1} + m_{i,t}^{0}\right). \qquad (12)$$

Recall that p_t denotes the market price following a trading round in time t and $\bar{r}^{monthly}$ is a one-period return on bank account. In order to ensure more efficient learning – just like in the case of dividend learning – constraints are imposed on the magnitude of price adjustment factors, and additional penalties are invoked if these constraints become binding.

The chosen specification of the reward function implies that the reinforcement-learning agents try to learn to organise their behaviour so that they maximise long-term returns on their investments. We could interpret agents in this model as professional fund managers that care about maximising clients' wealth, seek best long-term performance among peers and shun under-performance. They need not to be risk-averse, as is conventionally assumed about individual consumption-smoothing investors. Indeed, recent evidence from extremely turbulent financial markets shows that it might well quite the opposite – in some cases excessive risk-taking might generate superior performance for a prolonged period of time, which in turn generates solid growth in fee income during that time. In addition, it should be noted that in the model an agent's attitude toward risk is determined not only by its reward function but also by evolutionary selection and other systemic adaptation.

The model allows for optional alteration of agent behaviour via sharing private trading experience, competitive evolutionary selection and noise trading behaviour. These options help enhance realism of the artificial stock market and arguably augment the reinforcement learning procedure by removing clearly dominated trading policies implemented by individual agents and by strengthening competition among them.

In our model, dissemination of agents' experience is very stylised. At the end of each period agents are randomly matched in pairs. In every pair, agents' long-term performance measures, which are cumulative past rewards, are compared to each other. If the difference between matched agents' performance measures is sufficiently large (the threshold level is allowed to fluctuate randomly to reflect the random nature of knowledge dissemination), the worse-performing agent simply replicates the more successful agent's experience.

Evolutionary selection is another available option in the present ASM model. It assumes bankruptcy of worst-performing agents and their replacement with best-performers. So agents, whose performance relative to the benchmark (which is the average agents' performance) falls below a modeller-specified threshold, go bankrupt. Their place is taken over by best-performers, which then are forced to split so that the number of agents remains constant. This has a natural interpretation: inferior fund managers are forced out of the market as unsatisfied clients bring their wealth over to best-performing funds and the latter then have to split for regulatory or any other reasons. Successful agents are given substantial extra rewards in the event of the split, to encourage their performance.

Finally, the model allows for noise trading behaviour. Unlike in the evolutionary selection, the worst-performers are not replaced by most successful agents. Rather, they scrap their prior learning experience and, as a result, start learning from scratch.

3. Simulation results

Like the vast majority of other ASM models, the current model is based on a large number of parameters, and it is very difficult to calibrate the model to match empirical data. At this stage of model development we do not attempt to do that. Instead, we assign reasonable and, where possible, conventional values to the parameters and assume very simple forms of dividend-generating processes. This enables us to determine the approximate fundamental stock value dynamics and study how the market stock price, determined by the complex system of interacting heterogeneous agents, fares in relation to stock price fundamentals. Even though the model is not calibrated to the market data, model results can offer qualitative insights about market self-regulation, efficiency and other aspects of market functioning. In this section we examine these issues in more detail and report some of the more interesting simulation results.

The simulation procedure is implemented by performing batches of model runs. Each run consists of 20,000 trading rounds (about 1667 years). Batches of ten runs repeated under identical parameter settings are used to generate essential data and statistics that are in turn used for analysis and generalisation. In every run, the first 5,000 trading rounds – as the learning initiation phase – are excluded from the calculation of the descriptive statistics (presented in Table A2.3 in Appendix 2). The simulation concentrates on altering features of the reinforcement learning, interaction among agents and dividend-generating processes in an attempt to understand relative importance of intelligent individual behaviour, market setting and population-level changes for the aggregate market behaviour. Other model parameters are kept unchanged. Their values are provided in Table A2.1.

Dividends are assumed to fluctuate around an exponential trend and their volatility is proportional to the dividend level. The role of the trend is to necessitate the intelligent adjustment of dividend estimates, as forecasts based on exponentially-weighted moving averages would be clearly biased. Large dividend growth rates can only be sustained over relatively short time horizons, and hence in our very long-term model we have to choose very low dividend growth rates (e.g. 0.15 % per year). We also examine deterministic constant dividends, as a special case (see exact specifications of dividend generating processes in Table A2.2).

The primary question addressed in most ASM models is the market efficiency issue. Here efficiency is loosely interpreted as the congruence between the stock market price and its fundamentals. In the current setting it is not possible to know the right theoretical stock price, so we basically want to compare the market stock price with risk-neutral estimates of fundamentals.

Let us start with the examination of agents' ability to forecast dividends. Since dividends are driven by very simple data generating processes, it is not surprising that in the model version with enabled both reinforcement learning and evolutionary selection (Experiment 1 in Table A2.3) agents are able to form very precise forecasts. The average dividend forecast error for this model specification is -0.1%, while the average absolute forecast error again amounts to 0.4%. To assess the actual importance of the

reinforcement learning behaviour for dividend forecasting, simulation batches with disabled reinforcement learning are run (Experiment 3). In these runs agents neither learn to forecast dividends, nor try to optimise their portfolios, as their commensurate reinforcement rewards $r^{d}_{i,t+n}$ and $r^{p}_{i,t+1}$ are set to zero. In this case, the average forecast bias considerably increases to -0.8% and the average absolute errors stands at 1.4%. In this no-learning case the average percentage of agents hitting the modeller-imposed dividend forecast bounds increases significantly, as compared to the enabled learning case. In other words, learning agents are able to effectively form "reasonable" forecasts, while non-learning agents are simply forced to remain within prespecified boundaries but perform much worse, taken on individual basis. This leads us to a very natural conclusion that in the dividend forecasting process intelligent adaptation matters.

As the next step of our analysis we examine dynamics of the market price in relation to the fundamentals. In Experiment 1 fundamentals anchor the stock price dynamics to some extent, and the market price fluctuates in the vicinity of the perceived fundamental value The average percentage bias of market price from the fundamentals is low and stands at -1.6% (see Table A2.3). Nevertheless, the valuation errors are clearly autocorrelated – due to the market inertia and prevailing expectations, the stock price may be above or below risk-neutral fundamentals for extensive periods of time. For instance, runs of uninterrupted overvaluation stretch on average for 44 trading periods and an average length of undervaluation runs is 60 periods. By the same token, average market price deviations from the fundamental valuation are large relative to the price volatility. The enabled evolutionary selection option in the model ensures relatively even wealth distribution among agents and each trading period active agents (i.e. agents that have sufficient funds and/or stock holdings to trade constitute on average 89.7% of total population). Finally, the average fraction of agents whose adjusted fundamental valuations (reservation prices) fall out of modeller-imposed "reasonable" bounds is very low and stands on average at 0.1% of total population in a trading round.

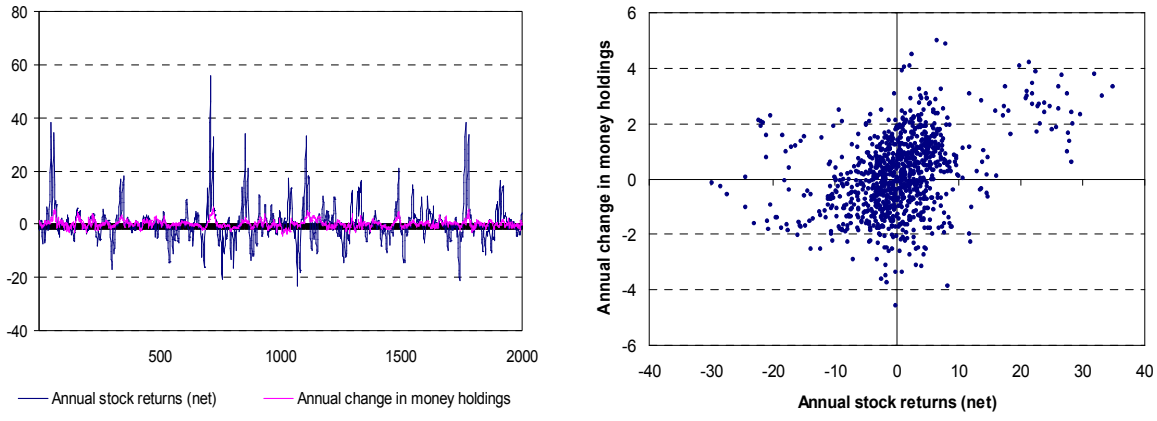
It turns out that the above results strongly depend on the evolutionary competition assumption. It suffices to disable the evolutionary selection (Experiment 2), and the average percentage stock price bias from the fundamentals boosts to 5.9% along with a dramatic increase in average overvaluation runs to 406. By the end of a simulation run the number of inactive agents per trading round increases to 70-80%, and wealth naturally concentrates in the hands of remaining 20-30% agents. There are some possible explanations to this overvaluation and wealth concentration. Such overvaluation can be to some extent associated with the model's feature that excess liquidity is simply taken away from the market, which means that the agents that tend to sell their stock holdings are more likely to "consume" their money and become inactive. In other words, those agents that highly value the stock tend to dominate in the market. Another interpretation is that worse-performing agents are simply driven out of the market. Moreover, a diminishing number of active participants and a smaller degree of competition allows agents to concert their portfolio rebalancing actions in such a way that the market price is driven up, which leads to larger unrealised returns and thereby stronger reinforcement for the remaining active players. These results make sense from the real world perspective.

The largest mass of investors want stock prices to be as high as possible (though possibly still compatible with fundamentals), and it is not in their direct interest to have prices that match fundamentals precisely.

We also perform simulations to examine market's self-regulation ability. In particular, we want to know whether economic forces are strong enough to bring the market to the true fundamentals if they systematically differ from average perceived fundamentals. For this purpose, we introduce and an arbitrary upward bias to the estimates of the fundamental value by adding an arbitrary term in equation (3). Then simulation runs are implemented for different model settings, with or without reinforcement learning. It turns out that the market is not able to find the true risk-neutral fundamentals. In the case of no-learning, stock prices tend to slowly grow larger than the perceived fundamentals. In the case of enabled reinforcement learning, agents tend to stick to the perceived fundamentals, and the market price fluctuates around them as a result.

The above results confirm the market self-regulation mechanism in this model is weak. We do not find evidence of agents adjusting their perceived fundamentals so that the market price gets in line with modeller-imposed fundamentals or, say, the usually assumed risk-averse behaviour. On the other hand, it is not surprising. Well known puzzles of empirical finance and recent mega-bubbles suggest that markets may not be tracking fundamentals so closely after all. It can be the case that markets exhibit so strong inertia that even fundamentally correct investment strategies pay out only in too distant future and may not be applied successfully or act as the market's self-regulating force. The obtained results suggest that (not necessarily objectively founded) market beliefs of what an asset is worth are a very important constituency of its market price.

Figure 3. Typical relationship between stock returns and liquidity in a constant dividend case



Last but not least, we want to examine the relationship between the market price fluctuations and the financial market liquidity. This experiment also helps to shed light on the reasons for a relatively loose connection between the market price and fundamentals. In this simulation run, the standard model version with reinforcement learning and evolutionary selection is used, while dividends are assumed to be deterministic and constant. It is notable that even in this environment market price fluctuations remain

significant and trading does not stop. The clue to understanding this excessive price volatility may be the positive relationship between market liquidity and the stock price. Since unnecessary liquidity at the individual level is removed from the system, overall liquidity fluctuates in a haphazard way. Increases in market liquidity bolster solvent demand for the stock and lifts its price. As can be seen from Figure 3, liquidity growth spikes are associated with strong price increases. The linear correlation between growth of money balances and stock price growth is found to be 0.32.

It should be noted that the latter experiment is devised so as to ensure that positive relationship between stock returns (with dividends included) and investors' cash holdings is not linked to fluctuations in dividend payouts, as they are assumed constant. This allows us to conclude that liquidity fluctuations affect the asset price in this case, and not vice versa. The evidence that market liquidity changes can move markets is very important for understanding the way liquidity crises, credit booms and busts (deleveraging), portfolio reallocations between asset classes and other exogenous factors may affect stock markets.

4. Conclusions

In this paper we developed an artificial stock market model based on the interaction of heterogeneous agents whose forward-looking behaviour is driven by the reinforcement learning algorithm combined with some evolutionary selection mechanism and economic reasoning. Other notable features of the model include knowledge dissemination and agents' competition for survival, detailed modelling of the trading process, explicit formation of dividend expectations and estimates of fundamental value, computation of individual reservation prices and best order prices, etc. Bearing in mind the uncertain nature of the model environment, mostly brought about by this same interaction, strategies followed by artificial agents seem to exhibit a good balance of economic rationale and optimisation attempts. Quite a strong emphasis on the model's economic content distinguishes this model from some other ASM models, which are most often based on evolutionary selection procedures and are sometimes criticised for lack of economic fundament.

Simulation results suggest that the market price of the stock in this model broadly reflects fundamentals but over- or under-valuation runs are sustained for prolonged periods. Both individual adaptive behaviour and the population level adaptation (evolutionary selection in particular) are essential for ensuring any efficiency of the market. However, market self-regulation ability is found to be weak. The institutional setting alone, such as the centralised exchange based on the double auction trading, cannot ensure effective market functioning. Even in the case of active adaptive learning, the market does not correct itself from erroneously perceived fundamentals if they are in the vicinity of actual fundamentals, which underscores the importance of market participants' beliefs for the market price dynamics. We also find a positive relationship between stock returns and changes in liquidity – there are indications that exogenous shocks to investors' cash holdings lead to strong changes in the market price of the stock.

Overall, this line of research seems promising. In our related research, we aim at developing a version of the model suitable for calibration to empirical data. This requires simplification of some processes in the model, taking steps to ensure more effective and robust learning, etc. The noteworthy implication of the proposed study is that similar modelling principles could be expanded and applied for modelling of other markets, such as markets for goods or labour. More generally, intelligent adaptive agents could form the basis of applied dynamic macroeconomic models.

References

Arthur W. B., Holland J., LeBaron B., Tayler P. 1997: Asset Pricing under Endogenous Expectations in an Artificial Stock Market. – In Arthur W. B., Durlauf S., Lane D. (Eds.) *The Economy as an Evolving Complex System II*, Addison-Wesley, Reading, MA, 15-44.

Beltratti A., Margarita S. 1992: Evolution of Trading Strategies among Heterogeneous Artificial Economic Agents. – In J. A. Meyer, H. L. Roitblat, S. W. Wilson (Eds.) *From Animals to Animats 2*, MIT Press, Cambridge MA.

Bertsekas D. P., Tsitsiklis J. N. 1996: *Neuro-Dynamic Programming*. Athena Scientific.

Brenner, T. 2006. Agent Learning Representation: Advice on Modelling Economic Learning. In L. Tesfatsion, K. L. Judd (Eds.) *Handbook of Computational Economics*, Vol. 2, 949–1011, Elsevier, Amsterdam.

Kaelbling L. P., Littman M.L., Moore A. W. 1996: Reinforcement Learning: A Survey. – *A Journal of Artificial Intelligence Research*, 4, 237-285.

Kim G., Markowitz H. 1989: Investment Rules, Margin, and Market Volatility. – *Journal of Portfolio Management*, 16, 45-52.

LeBaron B. 2000: Evolution and Time Horizons in an Agent Based Stock Market. Brandeis University Working paper, http://papers.ssrn.com/sol3/papers.cfm?abstract_id=218309.

LeBaron B. 2006: Agent-based Computational Finance. – In L. Tesfatsion, K. L. Judd (Eds.) *Handbook of Computational Economics*, Vol. 2, 1189-1233, Elsevier, Amsterdam.

Lettau M. 1997: Explaining the Facts with Adaptive Agents: The Case of Mutual Fund Flows. *Journal of Economic Dynamics and Control*, 21, 1117-1148.

Lux T. 1995: Herd Behaviour, Bubbles and Crashes. – *Economic Journal*, 105, 881-896.

Mitchell T. 1997: *Machine Learning*. McGraw Hill.

Ramanauskas T. 2009: Empirical Version of an Artificial Stock Market Model. *Monetary s studies* (forthcoming).

Ramanauskas T. 2009: Agent-Based Financial Modelling: A Promising Alternative to the Standard Representative-Agent Approach. Bank of Lithuania Working Paper No. 3.

Samanidou E., Zschischang E., Stauffer D., Lux T. 2007: Agent-based Models of Financial Markets. – *Reports on Progress in Physics*, 70, 409-450.

Sethi R., Franke R. 1995: Behavioural Heterogeneity under Evolutionary Pressure: Macroeconomic Implications of Costly Optimization. – *Economic Journal*, 105, 583-600.

Sutton R. S., Barto A. G. 1998: *Reinforcement Learning: An Introduction*. MIT Press.

Tay N. S. P., Linn S. C. 2001: Fuzzy Inductive Reasoning, Expectation Formation and the Behavior of Security Prices. – *Journal of Economic Dynamics and Control*, 25, 321-362.

Tesauro G., Kephart J. O. 2002: Pricing in Agent Economies Using Multi-Agent Q-Learning. – Autonomous Agents and Multi-Agent Systems, 5, 289-304.

Watkins C. J. C. H. 1989: *Learning from Delayed Rewards*. Ph.D Thesis, King's College.

APPENDIX 1. Reinforcement learning in the context of ASM modelling

Reinforcement learning addresses the question of how an autonomous agent that senses and acts in its environment can learn to choose optimal actions to achieve its goals (Mitchell 1997, p. 367). More specifically, by taking actions in an environment and obtaining associated rewards, a reinforcement-learning agent tries to find optimal policies, which maximise long-term rewards, and the process of improvement of agent policies is the central target for reinforcement learning methods. A good introduction to the reinforcement learning techniques may be found in Sutton and Barto (1998), Bertsekas and Tsitsiklis (1996) and Mitchell's (1997) books, and some broad overview of reinforcement learning models is given in Kaelbling, Littman and Moore (1996) survey. In this subsection we present briefly some basic principles of the reinforcement learning methodology with a special emphasis on Watkins' Q-learning algorithm, as it forms the basis of agent behaviour in our ASM model.

The iterative sequence of agent's interaction with environment is as follows. At time *t*, the agent observes environment state s_t and acts according to its action policy to produce action a_t. In the next time step it receives numerical reward signal r_{t+1} from the environment and observes new state s_{t+1}. Finally, it is ready to update its policies (if necessary) and take new action a_{t+1}. In the reinforcement learning problems it is also assumed that environment possesses the Markov property, i.e. all relevant information about possible future development of environment is encapsulated in the information about the current state and action. More formally,

$$\Pr\{s_{t+1}=s', r_{t+1}=r \mid s_t, a_t, r_t, s_{t-1}, a_{t-1}, r_{t-1}, ..., r_1, s_0, a_0\} = \\ = \Pr\{s_{t+1}=s', r_{t+1}=r \mid s_t, a_t\} \quad \text{(A1)}$$

If condition (A1) holds, such reinforcement learning task is called a Markov decision process. To completely specify the environment dynamics for a Markov decision process, it suffices to define state transition probabilities and expected rewards. State transition probabilities constitute a distribution of probabilities of each possible next state s', given any current state *s* and action *a*:

$$P^a_{ss'} = \Pr\{s_{t+1}=s' \mid s_t=s, a_t=a\}. \quad \text{(A2)}$$

Notably, in a general case, state transition probabilities are not known to the reinforcement-learning agent but can be inferred from interaction with environment. The expected next reward is

$$R^a_{ss'} = E(r_{t+1} \mid s_t=s, a_t=a, s_{t+1}=s'). \quad \text{(A3)}$$

As was mentioned above, learning is understood in this context as an attempt to find optimal policies. Here, a policy is defined as a mapping from each state *s* and action *a* to the probability $\pi(s,a)$ of taking action *a* when in state *s* (if a policy is deterministic, then it is simply a set of deterministic rules describing how to behave in each state). For the further elaboration of the reinforcement learning task, the notion of value functions

should be introduced. The state-value function for policy π is defined as the expected discounted cumulated reward conditional on state *s* and policy π :

$$V^{\pi}(s) = E_{\pi}\left(\sum_{k=0}^{\infty}\gamma^{k} r_{t+k+1} \mid s_t = s\right), \qquad \text{(A4)}$$

where E_{π} denotes the expectation given that the agent sticks to its policy π, and γ is a discounting parameter. It proves very useful to define also the value of taking action *a* in state *s* under policy π. The action-value function is given by

$$Q^{\pi}(s,a) = E_{\pi}\left(\sum_{k=0}^{\infty}\gamma^{k} r_{t+k+1} \mid s_t = s, a_t = a\right). \qquad \text{(A5)}$$

It is obvious that both value functions possess the Bellman property, i.e. they must be dynamically consistent. For instance, it follows from equation (A4) that

$$V^{\pi}(s) = E_{\pi}\left(r_{t+1} + \gamma V^{\pi}(s_{t+1}) \mid s_t = s\right). \qquad \text{(A6)}$$

Since condition (A6) holds for all value functions, it also holds for optimal value functions, i.e. those associated with optimal policies[1]. This leads directly to Bellman optimality equations for the state-value function

$$V^{*}(s) = \max_{a} E\left(r_{t+1} + \gamma V(s_{t+1}) \mid s_t = s, a_t = a\right) \qquad \text{for all } s \qquad \text{(A7)}$$

and for the action-value function

$$Q^{*}(s,a) = E\left(r_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a') \mid s_t = s, a_t = a\right) \qquad \text{for all } s \qquad \text{(A8)}$$

The most prominent feature of Bellman optimality equations is that they actually rearrange the multi-period optimisation problem into a problem consisting of a set of difference equations (one for each state). Notably, if value functions are known, it becomes very easy to find optimal policies. Equation (A7) implies that in any state *s* it suffices to take the greedy action (that is, concerned with only one period ahead) that maximises the expected sum of the immediate reward and the (discounted) next state-value[2]. It is even simpler if the problem is expressed in terms of known action-value functions – from equation (A8) it follows that action a' taken in state s_{t+1} will be optimal if it maximises the associated expected action-value function. To put differently, it is optimal to take actions that simply maximise each period's Q-function value (such actions are sometimes called Q-greedy actions).

The big question is, of course, how to find optimal value functions. One of the ways to do this is to apply dynamic programming, which also provides the foundation for reinforcement learning methods. The basic idea is to apply some iterative procedure aimed at evaluating current policies and gradually improving them until they converge to optimal policies. More specifically, the so-called generalised policy iteration consists of two interacting processes: (*i*) policy evaluation, which is the process of finding the value function for an arbitrary policy, and (*ii*) policy improvement, whereby policies are improved by making them greedy with respect to the current value function.

---

[1] Optimal policies are defined as policies that maximise state values V^{π} in all states.

[2] Notice that expectations are no longer conditioned on specific policies in equations (7) and (8).

The policy evaluation procedure uses Bellman equation (A6) as an update rule:

$$V_{k+1}(s) = E_\pi\left(r_{t+1} + \gamma V_k(s_{t+1}) \mid s_t = s\right), \tag{A9}$$

where V_k denotes the k-th approximation of the state-value function (V_0 is chosen arbitrarily). It can be shown that estimate V_k converges to true policy V^π as k converges to infinity. Each iteration is a sweep through all states – the value of every single state is backed up using equation (A9).

The policy improvement step is closely linked to Bellman optimality equation (A7). It can be shown that for every state s, the policy can be improved by taking action that maximises the immediate action value or, in other words, looks best in the short term (examining only one period ahead):

$$\pi^* = \arg\max_a E\left(r_{t+1} + \gamma V(s_{t+1}) \mid s_t = s, a_t = a\right). \tag{A10}$$

The two procedures, given in equations (A9) and (A10), are implemented alternately in each iteration, and the iterative process continues until state values and associated policies stabilise, which is when they become optimal. The problem with the dynamic programming is that in order to implement these back-up sweeps, state transition probabilities $P^a_{ss'}$ and expected rewards $R^a_{ss'}$ (see equations (A2) and (A3)) must be known, and it is very rarely the case in practice.

A natural way to overcome the problem of incomplete information is to use sample estimates instead of expectations. This is exactly what is done in two broad classes of reinforcement learning, namely, Monte Carlo methods and temporal difference models of learning. In the remainder of this section we present just one specific temporal difference learning method devised by Watkins (1989), also known as the Q-learning. This method's principal back-up rule is closely related to Bellman optimality equation (A8) and is of the following form:

$$Q(s_t, a_t) \leftarrow (1-\alpha) \underbrace{\cdot Q(s_t, a_t)}_{\text{Old estimate of } Q(s_t, a_t)} + \underbrace{\alpha(r_{t+1} + \gamma \max_a Q(s_{t+1}, a))}_{\text{New estimate of } Q(s_t, a_t)}. \tag{A11}$$

There are two differences from the dynamic programming update rule based on the Bellman optimality condition. First, as was already mentioned, the expectations operator is gone – the actual realised reward and actual action value from the look-up table are used instead of the expected reward and expected Q-value, respectively. Second, the Q-value in the look-up table is not directly replaced with its new estimate but is rather averaged with the previous estimate (which provides needed additional stability for the convergence to the correct Q function). The speed of learning, of course, depends on the learning parameter α – higher values of the learning parameter ensure faster learning. Higher values of α may be useful at the beginning of the learning process as the learning starts from arbitrary policies, or in nonstationary environment where the reinforcement-learning agent needs to adapt faster and more flexibly.

It was shown that under quite general conditions the update rule (A11) guarantees convergence of the action-value function to the optimal Q-function, provided all state-action pairs are visited infinitely many times. The latter condition is needed to avoid early

convergence to suboptimal policies. It requires that the learning agent continues to explore the environment by occasionally taking seemingly suboptimal values so as to ensure that all actions in all states are sufficiently explored. Hence, the Q-learning agent follows the Q-greedy policy most of the time but sometimes (e.g. with prespecified probability ε) takes an exploratory action, which may be completely random or oriented towards more efficient exploration. Such a behavioural policy is usually called ε-greedy.

Figure A1.1. Basic Q-learning algorithm

Initialise $Q(s,a)$, s arbitrarily
Repeat:
  Choose a using policy derived from Q (e.g. ε-greedy)
  Take action a, observe r, s'
  $Q(s,a) \leftarrow (1-\alpha)\cdot Q(s,a) + \alpha\left(r + \gamma \max_{a'} Q(s',a')\right)$
  $s \leftarrow s'$
until convergence is achieved or process is terminated

Source: adapted from Sutton and Barto (1998).

Having discussed the basic principles of the Q-learning agent's behaviour, now it is possible to describe its behaviour in the procedural form – see the pseudo-code in Figure A1.1. Unfortunately, this simple algorithm can be rarely applied in practice. The reason is that it requires representation of the Q-function as a table with one entry for each state-action pair. This is not possible if the state space is continuous. Even in discrete real-world problems – and especially in the problem of investment behaviour modelling – the size of the Q-table and the computational burden associated with back-up operations are basically unmanageable. This implies that usually it is impossible for the Q-learning agent to fully explore the state space and it is necessary to generalise its prior experience to unfamiliar, but qualitatively similar state-action pairs that are of interest. Such generalisation is also called structural credit assignment – another important feature of the reinforcement learning.

There are a number of readily available methods for experience generalisation. In our model we use the standard linear gradient-descent function approximation for the Q-function, which we now describe briefly.

The idea of the linear approximation procedure is to replace the representation of the Q-function as a look-up table with some linear function and iteratively update its parameters instead of updating Q-values for every single state. Hence, the estimate of the action value function is replaced by the following linear approximation:

$$Q_t(s,a) \approx \sum_{i=1}^{n} \Theta_t(i,a)\vec{\phi}_s(i), \qquad \text{for all } a. \tag{A12}$$

Here $\vec{\phi}_s$ is the $n \times 1$ vector of state features, i.e. arbitrarily chosen variables that reflect the distinctive features of a given state. Matrix Θ_t is the $n \times m$ parameter containing parameters associated with n state features for each of m possible actions. For more intuitive exposition it is convenient to work with column vectors of this matrix.

The gradient-descent methods seek to gradually adjust the current approximation of the Q-value toward its new estimate and the step size is proportional to the negative gradient of some measure of current deviation (e.g. mean squared error). More specifically, for a given action *a*, the parameter vector $\vec{\theta}_t$ can be updated as follows:

$$\vec{\theta}_{t+1} = \vec{\theta}_t - \frac{1}{2}\alpha\nabla_{\vec{\theta}_t}[v_t - Q_t(s_t,a_t)]^2, \qquad \text{for all } a, \tag{A13}$$

where v_t is the new approximation of the action-value function and serves as a training example for the parameter update, and $\nabla_{\vec{\theta}_t} f(\vec{\theta}_t)$ is the gradient of this example's squared error, i.e. the column vector of partial derivatives of function *f* with respect to elements of $\vec{\theta}_t$. By taking the derivatives in equation (A13), one gets

$$\vec{\theta}_{t+1} = \vec{\theta}_t + \alpha\left(v_t - Q_t(s_t,a_t)\right)\vec{\phi}_s, \qquad \text{for all } a. \tag{A14}$$

The new sample estimate of the action-value function, v_t, is obtained similarly to the basic Q-learning algorithm (see equations (A8) and (A11)). The parameter update equation (A14) thus becomes

$$\vec{\theta}_{t+1} = \vec{\theta}_t + \alpha\left(r_{t+1} + \gamma \max_a Q_t(s_{t+1},a_{t+1}) - Q_t(s_t,a_t)\right)\vec{\phi}_s, \qquad \text{for all } a. \tag{A15}$$

This equation forms the basis of the Q-learning algorithm, which is applied by artificial agents in our model when forming expectations about the intrinsic stock value. The detailed procedural form of the algorithm is given in Figure A1.2.

Figure A1.2. Gradient-descent function approximation Q-learning algorithm

Initialise Θ, ϕ_s, s, a arbitrarily
Repeat:
  Take action *a*, observe *r*, *s'*
  $\delta \leftarrow -\vec{\theta}_a^T \vec{\phi}_s$
  For all actions *a*:
    $Q(s',a) = \vec{\theta}_a^T \vec{\phi}_{s'}$
  $\delta \leftarrow \delta + r + \gamma \max_a Q(s',a)$
  $\vec{\theta}_a := \vec{\theta}_a + \alpha\delta\vec{\phi}_{s'}$
  With probability $1-\varepsilon$:
    $a \leftarrow \arg\max_a Q(s',a)$
  else:
    Choose a randomly
  $s \leftarrow s'$
until convergence is achieved or process is terminated

Source: Adapted from Sutton and Barto (1998).

The gradient-descent Q-learning is the so-called off-policy control method, as the value function backup procedure uses the highest Q-value of the resultant state, $\max_a Q(s',a)$, rather than the one associated with the current policy, $Q(s',a')$.

Unfortunately, convergence to the optimal solution or its vicinity is not guaranteed for the off-policy methods. Nevertheless Sutton and Barto (1998) suggest that it may be possible to guarantee convergence for the Q-learning algorithm when the Q-function estimation policy and the action policy are sufficiently close to each other, which is the case if the ε-greedy policy is followed. There is also evidence that these methods give good practical performance despite the lack of theoretical guarantees of convergence to optimal policies (Tesauro and Kephart, 2002).

APPENDIX 2. Parameter setting and experimental results

Table A2.1. Key parameter settings of the ASM model

General parameters	
Length of a simulation run (number of trading periods in a run)	20000
Number of simulation runs in a batch	10
Number of agents	100
Total number of shares	10000
Frequency of dividend payouts	Annual
Monthly discount rate	0.995
Annual interest rate on bank account	0.062
Liquidity ceiling (as a multiple of current stock price)	5
Trading	
Number of feasible price quotes in a trading period	50
Frequency of trading rounds	Monthly
Trade cost (as a fraction of trade value)	0.001
Learning	
Learning rate (alpha)	0.1
Exploration rate (epsilon)	0.1
Subjective discount parameter of reinforcement learning	0.995
Dividend forecasting horizon	5 years
Smoothing parameter in the EWMA of dividends, fundamental value	0.1
Dividend forecast constraint (as a fraction of current dividend)	± 0.3
Individual reservation price constraint (as a fraction of perceived fundamentals)	± 0.2
Action step size in the process of dividend learning (allowed percentage changes of the dividend adjustment factor)	-0.02; 0; 0.02
Action step size in the process of reservation price formation (allowed percentage changes of the price adjustment factor)	-0.02; 0; 0.02
Bankruptcy conditions in evolution procedure (and noise trading)	
Maximum number of bankruptcies in a trading round	3
Performance threshold (as a percentage of average performance)	0.7
Threshold for strategy imitation	
Average difference between two compared strategies (as percentage of the leading strategy)	0.2

Table A2.2. Specification of model experiment runs

Dividend generating process (Model 1)	$div_t = 25 \cdot 1.000125^t + 0.05 \cdot div_{t-1} \cdot \varepsilon_t$
Dividend generating process (Model 2)	$div_t = 25$

	Model	***Learning***	***Evolution***
Experiment 1	Model 1	ON	ON
Experiment 2	Model 1	ON	OFF
Experiment 3	Model 1	OFF	ON
Experiment 4	Model 2	ON	ON

Table A2.3. Basic descriptive statistics of simulation experiments

	Experiment			
	1	***2***	***3***	***4***
Dividend forecasting				
Average forecast bias, %	-0.1	-0.1	-0.8	0.0
Average absolute forecast error, %	0.4	0.4	1.4	0.1
Price dynamics relative to perceived fundamentals				
Average price bias from fundamentals, %	-1.6	5.9	7.6	-0.1
Average length of overvaluation runs	43.7	405.9	63.2	63.5
Average length of undervaluation runs	59.9	2.9	2.8	62.4
Upper semi-deviation (avg. overvaluation during a run above fundamentals), %	7.9	6.7	9.0	8.5
Lower semi-deviation (avg. undervaluation during a run below fundamentals), %	8.9	1.6	1.8	8.6
Average volatility (per trading round), %	2.9	2.2	3.6	2.8
Behavioural and budget constraints				
Average proportion of agents forming "unreasonable" dividend forecast (per forecasting round), %	0.0	0.0	5.0	0.0
Average number proportion of agents that have "unreasonable" reservation price (per trading round), %	0.1	0.4	3.3	0.1
Number of active agents, %	89.7	29.2	22.2	90.5
Adaptive adjustment				
Average dividend adjustment factor	1.0152	1.0162	0.9543	0.9979
Average price adjustment factor	0.9863	1.0044	0.9734	1.0022

Figure A2.1. Selected graphs of Experiment 1

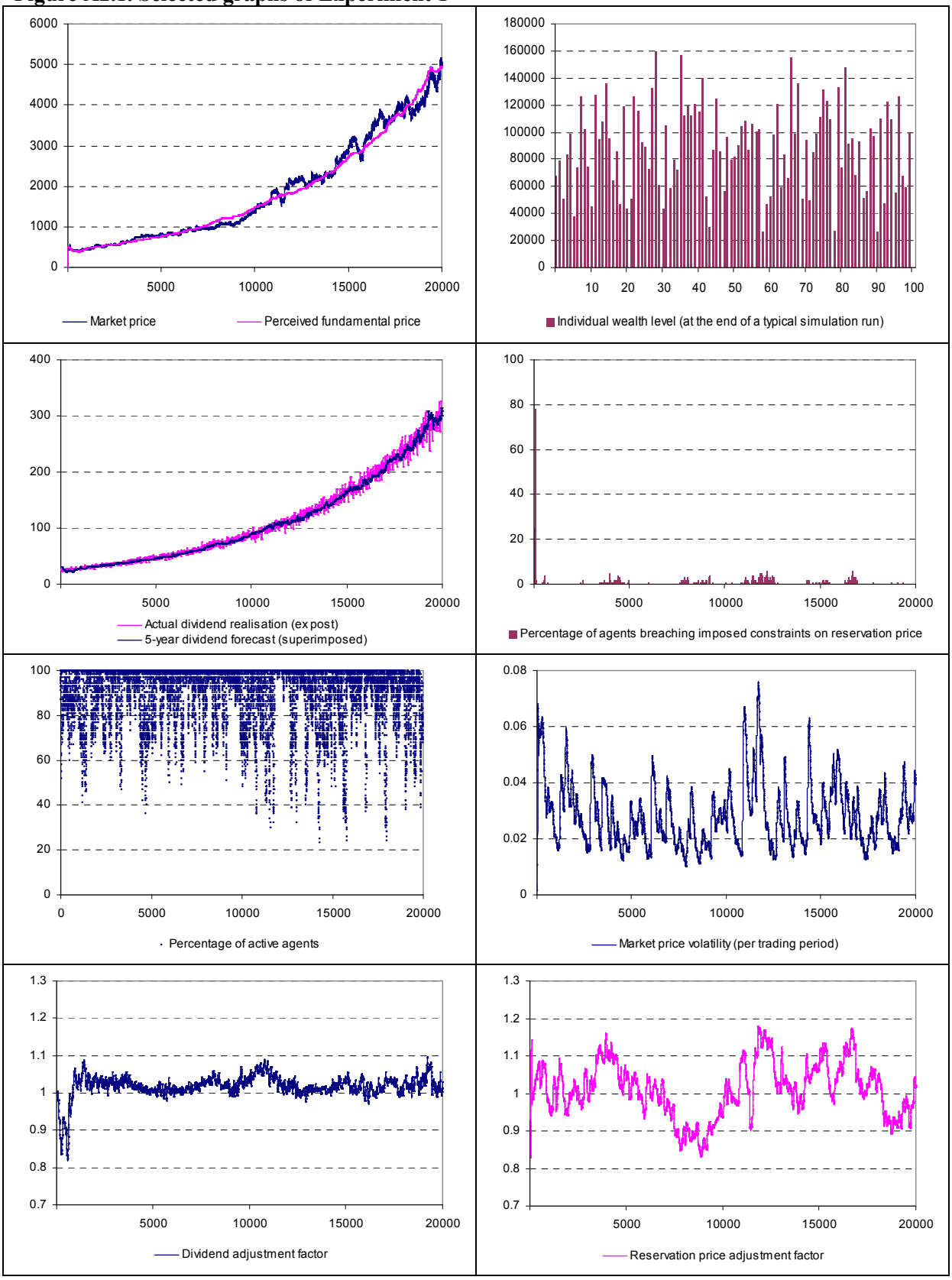


Figure A2.2. Selected graphs of Experiment 2

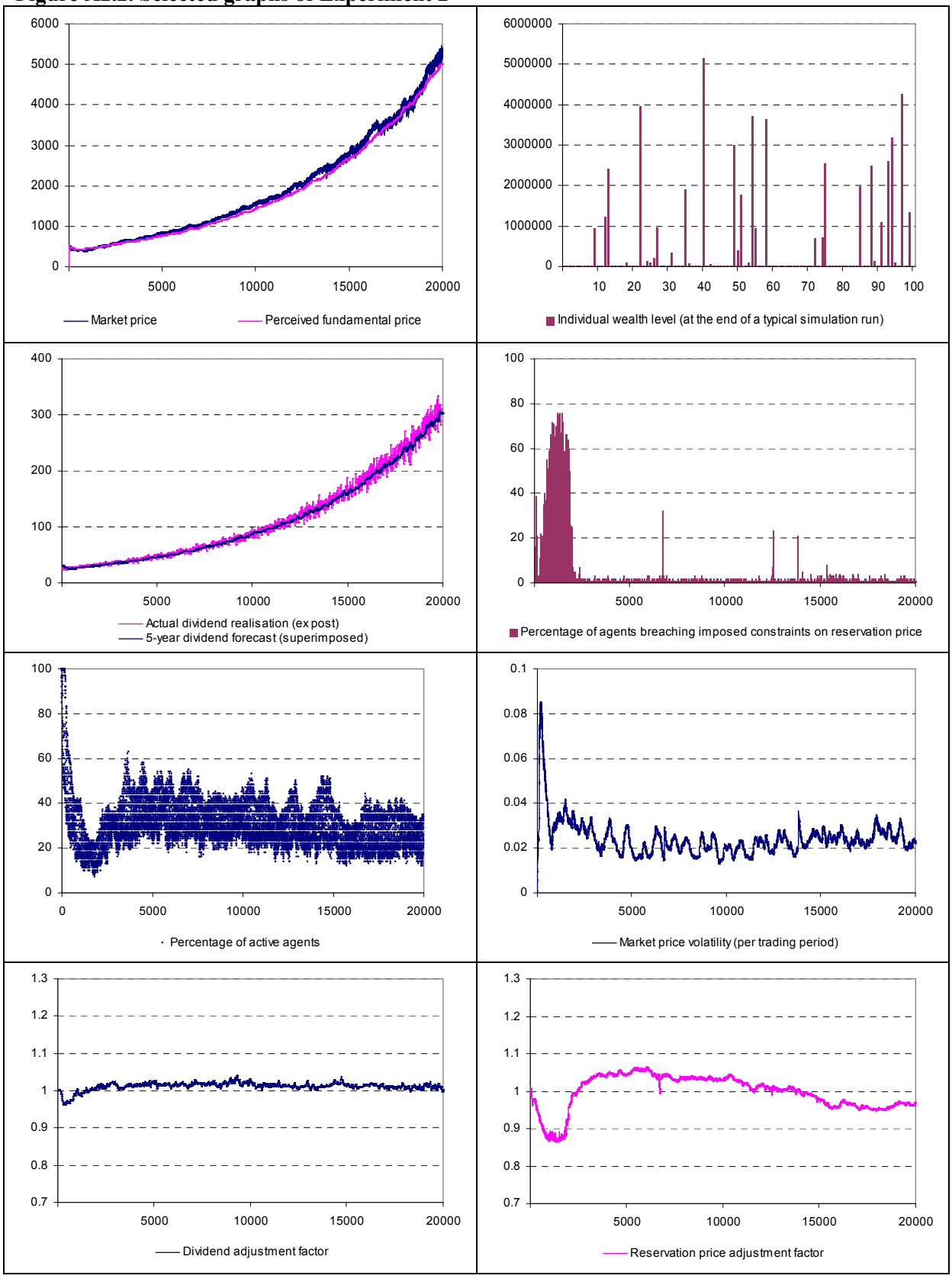


Figure A2.3. Selected graphs of Experiment 3

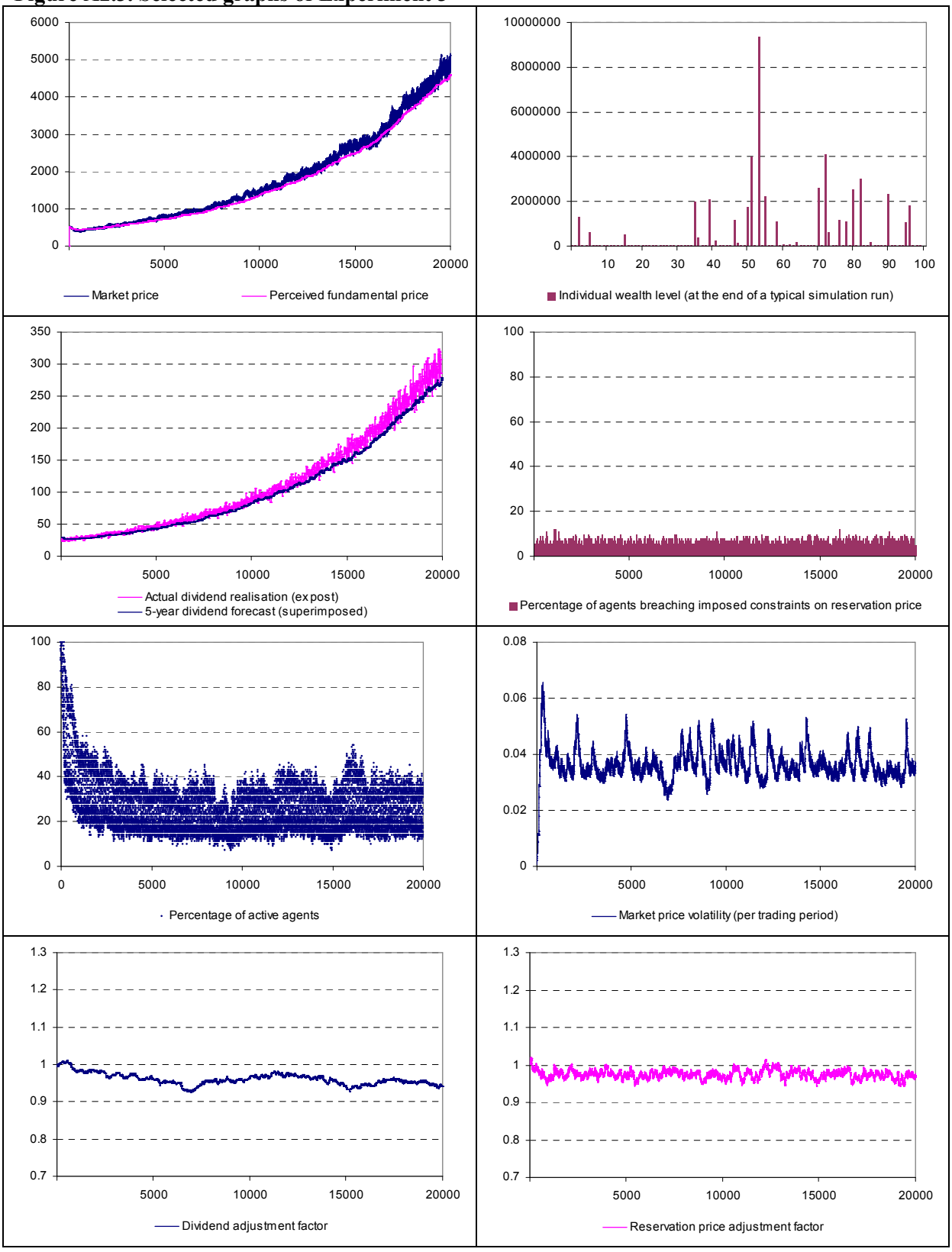
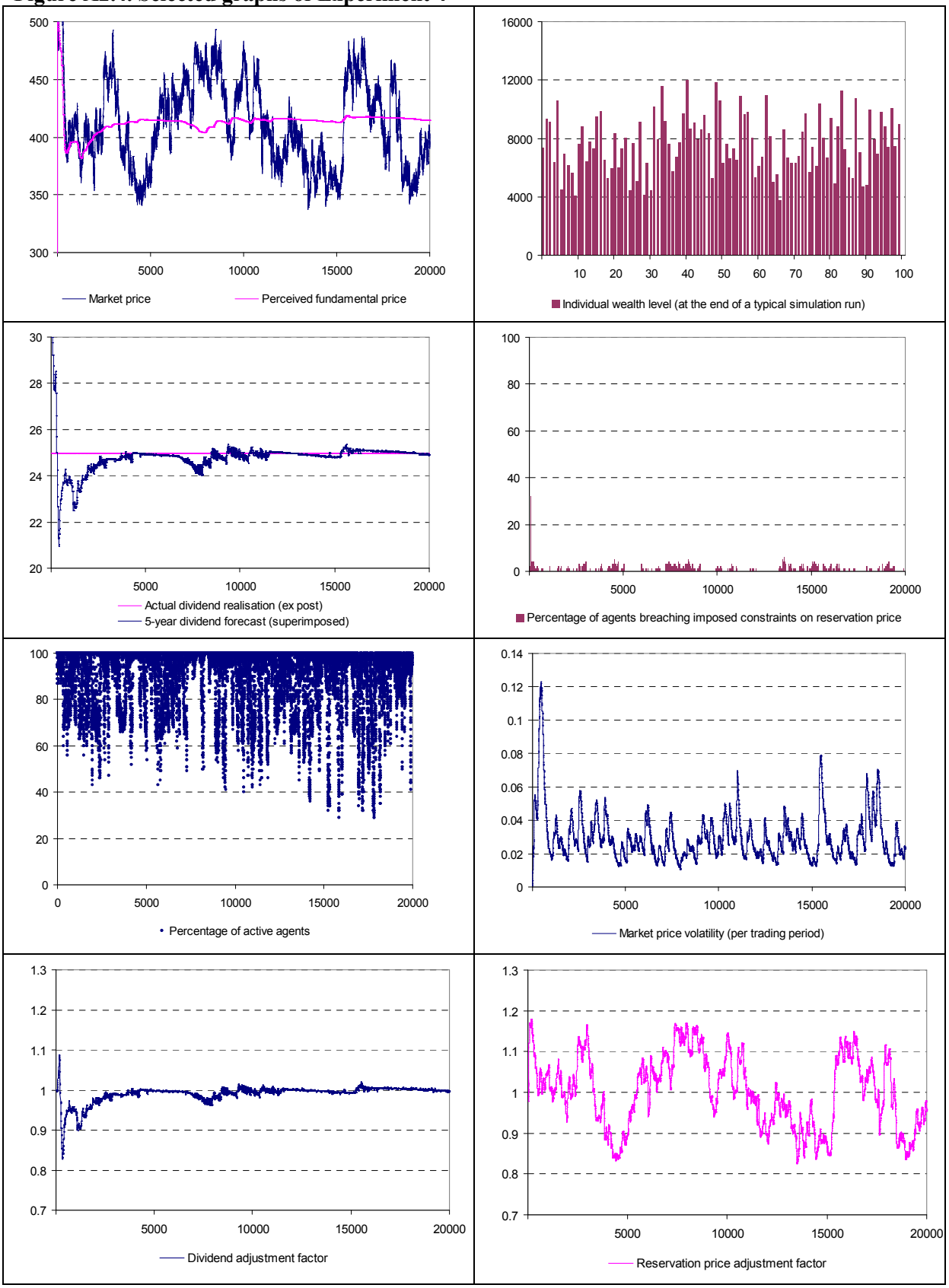


Figure A2.4. Selected graphs of Experiment 4



List of Bank of Lithuania Working Papers

No 6: "Building an Artificial Stock Market Populated by Reinforcement-Learning Agents" by Tomas Ramanauskas and Aleksandras Vytautas Rutkauskas, 2009.

No 5: "Estimation of the Euro Area Output Gap Using the NAWM" by Günter Coenen, Frank Smets and Igor Vetlov, 2009.

No 4: "The Effects of Fiscal Instruments on the Economy of Lithuania" by Sigitas Karpavičius, 2009.

No 3: "Agent-Based Financial Modelling: A Promising Alternative to the Standard Representative-Agent Approach" by Tomas Ramanauskas, 2009.

No 2: "Personal Income Tax Reform: Macroeconomic and Welfare Implications" by Sigitas Karpavičius and Igor Vetlov, 2008.

No 1: "Short-Term Forecasting of GDP Using Large Monthly Datasets: A Presudo Real-Time Forecast Evaluation Exercise" by G. Rünstler, K. Barhoumi, S. Benk, R. Cristadoro, A. Den Reijer, A. Jakaitienė, P. Jelonek, A. Rua, K. Ruth, C. Van Nieuwenhuyze, 2008.