# MODELING ADAPTIVE ECONOMIC AGENTS WITH PID CONTROLLERS

by

Ernesto Carrella
A Dissertation
Submitted to the
Graduate Faculty
of
George Mason University
In Partial fulfillment of
The Requirements for the Degree
of
Doctor of Philosophy
Computational Social Science

Committee:

______________________________ Chair of Committee

______________________________

______________________________

______________________________

______________________________ Director of Graduate Studies

______________________________ Director,
Krasnow Institute for Advanced Study

Date: __________________________ Summer 2015
George Mason University
Fairfax, VA

Modeling Adaptive Economic Agents with PID Controllers

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at George Mason University

By

Ernesto Carrella
Master of Science
University of Illinois at Urbana Champaign, 2010
Bachelor of Science
Chinese University of Hong Kong, 2008

Director: , Professor
Department of Computational Social Science

Summer 2015
George Mason University
Fairfax, VA

# Dedication

I dedicate this dissertation to my wife, my son and the rest of my family.

## Acknowledgments

I would first like to thank my committee for midwifing this dissertation: Rob Axtell, Andrew Crooks, Tim Gulden, Peter Woolf and Peter Howitt.

I'd also like to thank the housing market modelling group for letting me work on the project as it was such a great learning experience: Doyne Farmer, John Geanakoplos, Matthew Hendrey, Nathan Palmer, Chun-Yi Yang, Jonathan Goldstein, Benjamin Conlee and Philip Kalikman.

I'd like to thank George Mason University for providing me with the Presidential Fellowship and 3 years of stipend as a Research Assistant where I benefited greatly by being mentored by Matthew Hendrey, Andrew Crooks and Maksim Tsvetovat.

Finally I'd like to thank the rest of the Computational Social Science department, surely most of the merit (but none of the blame) of this dissertation goes to what we lazily discussed when trying to avoid actual work; in particular I'd like to thank Karen Underwood, Joseph Harrison, Sarah Wise, Omar Guerrero, David Masad, Maksim Tsvetovat, Johnny Nelson, Clarence Dillon, Chenna Cotla, Cristina Metgher, Chris Rouly and Claudio Cioffi.

# Table of Contents

# List of Tables

# List of Figures

## Abstract

MODELING ADAPTIVE ECONOMIC AGENTS WITH PID CONTROLLERS

Ernesto Carrella, PhD

George Mason University, 2015

Dissertation Director:

I provide here a counterpoint to the rational agents that dominate economics: rather than adding rigidities and information limits on an otherwise classical feed-forward agent, I build a new feed-back agent that achieves equilibrium without knowledge of the model or the market it is in. I collect here three essays that share this cybernetic approach. They focus on microeconomics, supply chains and macroeconomics respectively.

The first essay focuses on the agent itself and how it matches demand and supply, how it maximizes profits and drives production and how it competes against other agents. In the second essay I place the same agents in a supply chain and show how price rigidities emerge naturally from the lack of information and the resulting delays in coordinating downstream and upstream production. In the final essay I place these agents in a full macroeconomic model and show how changing the relative speed of adjustment of production targets to prices generate completely different disequilibrium dynamics.

# Chapter 1: General Introduction

## 1.1 Motivation

My aim for this work is rather ambitious. I provide a new methodology for the analysis of markets. The end result is a rather simple cybernetic model of the world where firms are competing controllers reacting and learning from prices they set. The starting point, however, was my inability to use standard economic analysis to examine the outcome of markets where agents have no information. Any practitioner trying to build models with less than ultra-rational agents must be ready to manipulate extremely complex mathematical objects saddled with many ad-hoc assumptions [Colander et al., 2008]. It is an approach that cannot be easily generalized across the domain and is therefore destined to be relegated to very specific case studies.

The three essays that follow are my answer. I give up the rational all-knowing fully learned agent and start over from the opposite direction. My agents are amateurs. They don't understand the market they trade in. But they try until it looks as if they do know what they are doing. These models are agent-based simulations and sidestep the usual mathematical difficulties associated with limited rationality. Simple trial-and-error agents are trivial to implement computationally but their interactions are complex enough to warrant this full dissertation.

Why use simple agents in the first place? Underlying the rational agent there is I believe a fundamental assumption of scale. An economic model necessarily reduces and simplifies the economy it portrays. This way the economy is optimizable by the rational agent within it. However, to apply the model to the real world we must assume that the learning abilities of the agent scale up as quickly as the economy complicates. This is an assumption I am not ready to make.

## 1.2 What has been achieved

I contribute to Economics both in theory and practice. The theoretical contribution is the cybernetic agent itself. The agent is simple to code, independent of market structure and can operate simultaneously in multiple markets; this makes it easy to plug into other agent-based models. Whether it is helpful to use this agent in more complicated markets is a question asked in chapter 2 and answered by the two chapters that follow it. In chapter 4 I plug my agent in a macroeconomy and it works "out of the box", no modification is required. When the agent has to be modified, as in chapter 3, the modification itself is significant: in this case price stickiness.

A second result of my models is to highlight the importance of dynamics to equilibrium rather than the equilibrium itself. Every single market I simulated in my dissertation has a single equilibrium point. For all the markets I simulated the standard way of studying them would be to assume that the agents are already in equilibrium. Instead what matters is how to get to equilibrium: in chapter 3 agents are unable to steady their supply chains without price rigidities, while in chapter 4 the speed of adjustment influences the depth of a recession.

My third contribution is more empirical. In section 3.9 I first explain how to fit price and error time series to find the parameters of a PI controller that would generate them. I then show how the European Central Bank rates and wholesale prices of consumables in Spain can be seen as if generated by a PI controller. That central bank rates can be simulated by a PI controller is not surprising, after all the Taylor rule is a rudimentary feedback control (see section 4.2) [Hawkins et al., 2014], but it's important that other markets and agents can be similarly simulated. This vindicates my approach to modeling: economists model central banking as feedback process because they recognize how hard it is to predict the effects of changing interest rates; my approach has been to generalize this lack of knowledge to all sectors and all agents within an economy forcing them to proceed by trial and error feedback like a central bank would.

## 1.3 Organization of the Dissertation

I develop my disequilibrium model in this dissertation over three separate essays. The first essay simplifies too much. Agents don't use efficiently the little information they have. The best way to understand what is missing is to focus on what prices do in the economy. In the first essay prices allocate endowment efficiently: agents with the highest demand consume the scarce resource produced and the cheaper workers get hired first. But prices in the first essay don't direct production as efficiently. The Zero-Knowledge firm spends most of its effort discovering the correct prices of inputs and outputs but then doesn't use them to drive production. The Zero-Knowledge firm in the first essay only uses past profit to choose how many workers to hire. This works only in a monopolistic setting; in a competitive market the first model produces too much noise.

The second essay solves all the technical hurdles of the first essay only to discover how less efficient controllers work better when taking into consideration the interaction among agents. Agents use marginal benefits and costs together with price slopes learning to drive production. It results in quick and effective decisions and much more realistic purely competitive markets. But when these efficient agents are placed in a supply chain it turns out that being too fast, too aggressive and too efficient results in noise and confusion being created and spread throughout the model. The main result of the second essay is to show how going slowly is more effective for the system as a whole.

The third essay is an extension of the second. In the second essay I show how agents in a supply chain can perform better when they slow down price adjustment. Slow price adjustment is however only one way to alter the agent's efficiency. There is in fact a spectrum of choices between the relative speed with which agents change prices compared to how quickly they change production. In the third essay then I compare the economy of a world where agents change price fast and labor slowly against a world where they adjust prices slowly and labor fast. This difference in relative speed has no effect when studying partial equilibrium but in a full macroeconomic model it gives rise to different dynamics with different winners and losers.

I coded this dissertation twice: once in Java and once in Dart. There are three reasons for that. First, I wanted to validate my results and having two completely separate codebases that output the same results is the most reassuring way to do so. Second, I wanted to circumvent some constraints I coded in the original Java model. Prices and quantities in the Java version can only be natural numbers, for example, and changing it would have required so much refactoring that coding it again from the beginning was easier. Finally, I wanted the ability to deploy my model on webpages and allow users to interact with it as quickly as possible and Dart allows it in a way Java used to before the new security measures.

This dissertation is open source[1]. The very dedicated reader will notice how the codebases are larger and more ambitious than the dissertation itself. I coded inventory controllers and feed-forward controllers, I coded a geographical monopolistic competition model and a 2-region agglomeration model. None is featured here. While technically interesting and more advanced those elements were not strictly needed to build the baseline economic model I wanted; they were all cut to keep the economics clear and concise.

The end result of this dissertation is a simple disequilibrium model. Agents eventually act rationally but only do so after groping laboriously for the equilibrium. I believe it represents an optimal base on which to build more models and I plan on using these agents as a starting point for my research in years to come.

## 1.4 Why Cybernetics through PI(D) Controllers

I built a cybernetic model, that is a world where agents are represented by closed-loop controllers fighting one another. But it isn't obvious why I would choose PID controllers in particular as a way to model trial and error. Neither [Tustin, 1957] nor [Lange, 1970] mention PID controllers in their discussion of cybernetic economies, preferring the more general description of economic "servos".

The main reason is the argument put forward in [Bennett, 1993] (also cited in [Hawkins

[1] The java code is available at `https://github.com/CarrKnight/MacroIIDiscrete`, the dart code is available at `https://github.com/CarrKnight/lancaster`

et al., 2014]) is that “in the absence of any knowledge of the process to be controlled, the PID controller is the best form of controller”. This matches very well with the purpose of my agents. Rather than thinking cybernetically top-down with the Gosplan being the well-informed well-meaning ultimate controller I wanted bottom up equilibrium coming with no centralized information. PID controllers are ideal for this.

Except for a few cases, I used the simplified PI controller rather than the PID. The derivative part of the controller is useful to fine-tune its operations as it tends to speed up the action of the controller. However my model would be weak if it depended too much on the tuning of the controller: if agents' trial and error worked only for very specific kinds of experimentation it wouldn't be as Zero-Knowledge as claimed. Fortunately, PID controllers are often badly tuned (80% of them are poorly tuned in industrial applications according to [Van Overschee et al., 1997]). This is cause of concern in the process control literature as the private sector doesn't use the sophisticated tuning rules that have been developed over the years. To me though it is a cause of modest celebration as it shows how PID controllers are resilient to the worst handicap of all: unsophisticated users like me.

## 1.5 Alternatives

In section 2.2 I categorize economic agents by the market process they generate and place my agents in the "endogenous-disequilibrium" group. In section 4.2 I categorize economic agents on a spectrum that goes from complete feedback to complete feed-forward and place my agents in the complete feedback group. Mine are not the only agents in the "endogenous-disequilibrium-feedback" class.

The "Probe and Adjust" agents [Kimbrough and Murphy, 2008] are the most similar to mine as both are trial-and-error adapters. The difference in the way they adapt is small but has large consequences. My agents have a sense of direction when they adapt: if they lower the price and that improves their situation, they will lower the price again; Probe and Adjust agents instead are pure hill-climbers, they probe a random neighboring state regardless of previous history. Probe and Adjust agents are then more useful in models where "direction" policies make less sense as in solving Sender-Receiver signaling game [Skyrms, 2012] or producing auction bids as step-wise supply functions [Kimbrough and Murphy, 2013]. My agents instead are easier to put in hierarchy (see section 2.5) and can work together to clear multiple markets (see chapter 3).

The other difference between Zero-Knowledge and Probe and Adjust is that my agents do not assume a particular market structure. Probe and Adjust agents interact in a Cournot game: they propose a quantity and the game tells them the resulting price. My agents figure out both price and quantities. The downside of having to discover prices together is that it creates noise in competitive markets (as small changes in price result in large sales effect ), the advantage is that the agents work regardless of market structure. Zero-Intelligence Plus [Cliff et al., 1997] is another framework for trial and error that shares the same dependency to market structure as the Probe and Adjust, as discussed in section 2.2.

These agents dealt with unknown markets by assuming them unknowable and resort to simple trial and error. A different modelling strategy is to have agents learn about their environment over time. A common approach is to use genetic algorithms, see [Arifovic, 1994] for a one-market example and [Gintis, 2006] for a multiple markets one. A genetic algorithm can be interpreted as a form of population wide trial-and-error. Many agents try different strategies and then the successful ones are copied. The main weakness of this approach is information inconsistency: agents follow blindly their genes having no information of the outside world except when the genetic algorithm runs at which point every agent suddenly knows everything about everyone else and can rank his genes with the competition.

Individual learning agents, like the Gjerstad and Dickhaut method [Gjerstad and Dickhaut, 1998], are usually tied to learning a very specific model and are therefore inflexible. If they are placed in a different market or if their prior is wrong as in [Kirman, 1975] then they fail to learn. An exception is reinforcement learning agents as in [Kimbrough and Lu, 2005]. The main problem of reinforcement learning is that it generates opaque agents whose strategies are brittle to changes in market conditions. They are also hard to implement which I think explains why they are rare when building simple agents.

# Chapter 2: Zero-Knowledge Traders [1]

## 2.1 Introduction

Agents are coordinated by the prices they set. Markets clear when prices balance correctly all possible economic information. Yet individual agents need to set these prices with only some of that information. I present a model where agents endogenously discover and set market-clearing prices using none of that information.

I give agents simple decision rules that allow them, with no knowledge of demand, supply or market structure, to solve for both competitive and monopolist prices and quantity. After a brief literature review in section 2.2, I explain how agents trade in section 2.3 and 2.4 of this chapter. I then expand them to make the agents also produce and maximize profits in section 2.5 and 2.6 of this chapter.

Agents using these rules are pure tinkerers. They adapt not by learning the real model of the world but by assuming such a model is unknowable and then proceeding by trial and error. Tinkering keeps these rules general-purpose.

I believe this methodology useful for two reasons. First, I provide a ready-made set of decision rules that can be used in almost any other agent-based model. There is no market structure or auctioneer feeding the prices to the agent. I believe them perfect as a baseline to compare to more nuanced decision rules.

Second, I provide a "rationality floor": the minimum information and rationality needed for markets to work, like the Zero-Intelligence project [Gode and Sunder, 1993]. Unlike Zero-Intelligence, my rules apply to both trading and production and do not depend on the very strict statistical assumptions that doomed Zero-Intelligence [Cliff et al., 1997].

[1] This chapter was published in the Journal of Artificial Societies and Social Simulation, Volume 17, Issue 3, Page 4

The lack of knowledge assumed in this model is extreme to the point of caricature. This is by design. Firstly because the fewer informational assumptions I make, the easier it is to plug-in these rules in other models. Secondly because I can test the robustness of traditional partial equilibrium analysis to a complete violation of standard rationality assumptions.

As [Mäki, 2008], but see also [Nowak et al., 2011], I claim that the fundamental contribution of any model is to isolate causal mechanisms in a complicated world. Here the mechanism allows firms to maximize profits and price goods correctly just by monitoring the difference between what they produce and what they sell.

In a more realistic model, firms would have more information and intelligence but they would need to solve a higher dimensional problem trying to manage not just production and prices but also customer satisfaction, labor relations, geography, social networks and so on, mixing all causal mechanisms in a single incomprehensible cacophony of parameters. That model would resemble reality better but it wouldn't be more useful.

## 2.2 Literature Review

I can categorize market processes along two axes. First whether the price vector is provided exogenously or discovered endogenously. Second whether the process allows trades to occur in disequilibrium before the equilibrium price is found.

*Exogenous-equilibrium*: the modeler solves for the market clearing prices and assumes they are known to the agents. This requires agents to be as rational, informed and computationally capable as the modeler who created them. Unfortunately the computational ability assumed is very high: even when equilibrium prices are known to exist, the utility is linear and the goods are indivisible, approximating equilibrium prices is NP-hard [Deng et al., 2002]. More generally exchange equilibrium is a PPAD(Polynomial Parity Arguments on Directed graphs)-complete problem [Papadimitriou, 1994].

*Exogenous-disequilibrium*: the modeler imposes a market formula that changes prices in reaction to some aggregate variables like excess demand or productivity. Scarf's paper on the subject[Scarf, 1960] is instructive in both explaining the idea and giving examples where an equilibrium exist but this methodology fails to find it. When finding an equilibrium is not an explicit goal, agent-based models use this methodology: for example the wage-setting algorithm in [Dosi et al., 2010].

*Endogenous-equilibrium*: the agents play a game against one another (for example a Bertrand competition) and choose the Nash equilibrium price and quantity. This still requires agents both to have enough information about their competitors to feed into their best response function and high computational ability: finding Nash-equilibria is also PPAD-complete [Chen and Deng, 2006].

*Endogenous-disequilibrium*: agents interact and trade between themselves without waiting or solving for equilibrium. The oldest market process model, the Walras's tatonnement, involved independent agents exchanging tickets at disequilibrium prices (see chapter 3 of [Currie and Steedman, 1990]). Agents traded tickets rather than goods because trading goods at disequilibrium creates wealth effects and path dependencies that invalidate welfare theorems; see [Jaffe, 1967] for a discussion about Walras, see [Foley, 2010] for a modern treatment on welfare theorems under disequilibrium. Modern disequilibrium models usually don't assume welfare theorems hold. I catalog these models by the market structure used.

In a strictly bilateral market, agents are randomly matched and barter with one another. There is no single market price but many trade prices. The pricing strategy depends on the matching and bartering functions used. If agents can compare their profitability with the rest of the population, like in [Gintis, 2007], the prices offered by each agent can be driven by evolutionary methods. If an agent only knows the characteristic of whom it is matched to, like in [Axtell, 2005], market clears by letting every beneficial barter occur between all trader pairs. Results can be driven by matching rather than bartering, as in [Howitt and Clower, 2000], where fixed-price shops are built endogenously and agents have to search for the right shops to exchange goods.

A more general market structure is the continuous double auctions with multiple buyers and sellers. My model belongs to this category. Here the two main behavior algorithms are: Zero Intelligence Plus [Cliff, 1997] and the Gjerstad and Dickhaut method [Gjerstad and Dickhaut, 1998]. They represent the two opposite views on adaptation: tinkering and learning. Zero Intelligence Plus traders tinker with their markup according to the previous auction results while Gjerstad and Dickhaut auctioneers first learn a probabilistic profit function and then maximize it.

My algorithm is simpler. Like Zero Intelligence Plus, I set prices by tinkering over previous errors. Unlike Zero Intelligence Plus, I use no auction-specific information and so my algorithm is market-structure independent. Moreover my algorithm can be expanded to direct production and maximize profits rather than just trade. The tinkering and adjustment is simulated through the use of Proportional Integral Derivative (PID) controllers.

While control theory is a staple of macroeconomics and PID controllers the simplest and commonest of controls, to the best of my knowledge I am the first to use PID control in economics. The closest paper to my approach is [Ortega and Lin, 2004] where a PID controller is suggested for inventory control, equalizing new buy orders to warehouse depletion. That is not an economic model as it doesn't deal with prices or markets. In the spirit of [Bagnall and Toft, 2006], I judge my algorithm by testing it in a series of markets where the economic theory identifies a clear optimum.

## 2.3 Zero-Knowledge Sellers

The seller is tasked to sell 100 units of a good every day. It has no information on demand or competition and no opportunity to learn. All the agent can do is set a sale price and wait. If at the end of the day it has sold too much, it will raise the price tomorrow. If it has sold too little, it will lower the price. This is an elementary control problem. The seller has a daily target of 100 sales and wants to attract exactly 100 customers a day. The seller has no power over customers themselves and so it needs to manipulate another variable (sale price) to affect the number of customers attracted. The seller doesn't know what the relationship between sale price and customers attracted is and so proceeds by trial and error. The trial and error algorithm used by sellers in this paper is a simple PID controller.

Given target $y^*$ (target sales) and process variable $y$ (today's number of customers), the daily error is:

$$e_t = y_t^* - y_t \tag{2.1}$$

Define $u_t$ as the policy (sale price). The seller manipulates the policy in order to reduce the error. The true relationship between policy and error is unknown, so the seller follows the general rule: "increase the policy when the error is positive, decrease it when the error is negative", which is the definition of negative-feedback control [Åström and Hägglund, 2006].

The PID controller manipulates the policy as follows:

$$u_{t+1} = ae_t + b\int_0^t e_\tau d\tau + c\frac{de_t}{dt} \tag{2.2}$$

Intuitively the policy is a function of the current error (proportional), all observed errors (integral) and the change from the earlier errors (derivative). In discrete time models (as the simulations in this paper) the equivalent formula is:

$$u_{t+1} = ae_t + b\sum_{i=0}^{t} e_i + c(e_t - e_{t-1} \tag{2.3}$$

There are four reasons as to why PID controllers are a good choice to simulate agents' trial and error. First, PID controllers assume no available information. Agents using PID rules act only on the outcome of previous choices.

Second, PID controllers assume no knowledge on how the world works. The PID formula contains no hint on how policy affects the error: there are no demand or supply functions. The PID formula leads agents to tinker and adapt without ever knowing or learning the "true" model.

Third, PID controllers make no assumptions on what the target should be. The target in the PID formula is completely exogenous. The controllers work regardless of how the target is chosen or how often it is changed.

Finally, PID controllers can complement other rules through feed-forwarding. Feed-forwarding refers to using PID controllers on the residuals of other rules. For example, take a more nuanced seller choosing its sale price by estimating a market demand function from data. This estimation would provide an approximate prediction of demand given the sale price. Still we could improve this approximation by adding a PID controller to adjust the sale price by setting as error the discrepancy between predicted and actual demand.

For PID controllers to work, four assumptions need to be made on the market in which they are employed. First, PIDs work by trial and error so the market structure must allow agents to experiment. This means that policies (prices) must be flexible. The stickier the policies, the slower the agent is at zeroing the error.

Second, PID controllers work better when even small changes in policy have some effects on the error. For Zero-Knowledge sellers the equivalent assumption is facing a continuous demand function. This does not mean that discontinuities automatically invalidate PID control and in fact all the computational examples in this chapter have discrete and discontinuous demands. But PID performance degrades with discontinuities resulting in more overshooting and slower approach to the equilibrium prices.

Thirdly, PID controllers implicitly assume a downward sloping demand: lower prices increase sales, higher prices decrease them. Zero-Knowledge sellers would fail to price Giffen goods.

Finally, targets must be achievable. For example, finding the price to sell to exactly $n$ agents in a world with infinitely elastic demand is impossible. The target "exactly $n$ sales" is unreachable: the error will oscillate between $n$ and infinity, never reaching zero. Section 2.5.2 of this chapter deals with how to set targets endogenously.

## 2.4 Zero-Knowledge Sellers Example

### 2.4.1 Mathematical Example

It is possible to show the workings of a Zero-Knowledge seller without software. Take a seller facing the unknown demand curve and tasked to sell 5 units of good every day. This Zero-Knowledge seller uses a PID controller with the parameters 0.01 for the proportional error, 0.15 for the integral and 0 for the derivative. Table 2.1 tracks the trial and error process of the seller as it discovers the right price (19) and sells the right number of goods.

Table 2.1: Non-Computational Example of a Zero-Knowledge Seller

| Day | $e_t$ | $\sum_{i=0}^{t} e_t$ | Price ($u_t$) | Quantity to sell ($y_t$) | Customers Attracted |
|---|---|---|---|---|---|
| 1 | · | · | 0 | 5 | 100 |
| 2 | 95 | 95 | 15.2 | 5 | 24 |
| 3 | 19 | 114 | 17.290 | 5 | 13.550 |
| 4 | 8.55 | 112.55 | 18.468 | 5 | 7.660 |
| 5 | 2.660 | 125.210 | 18.808 | 5 | 5.960 |
| 6 | 0.960 | 126.170 | 18.935 | 5 | 5.325 |
| 7 | 0.325 | 126.494 | 18.977 | 5 | 5.113 |
| 8 | 0.113 | 126.607 | 18.992 | 5 | 5.039 |
| 9 | 0.039 | 126.646 | 18.997 | 5 | 5.005 |

### 2.4.2 Computational Example

A seller receives daily 4 units of a good to sell. There is a fixed daily demand made up of 10 buyers. The first is willing to pay \$90 or less for one good, the second \$80 and so on. The demand repeats itself every day. Agents trade over an order book: the seller sets its price and all crossing quotes are cleared (while supplies last). The trading price is always the one set by the seller. Prices can only be natural numbers. The demand-supply schedule is shown in figure 2.1.

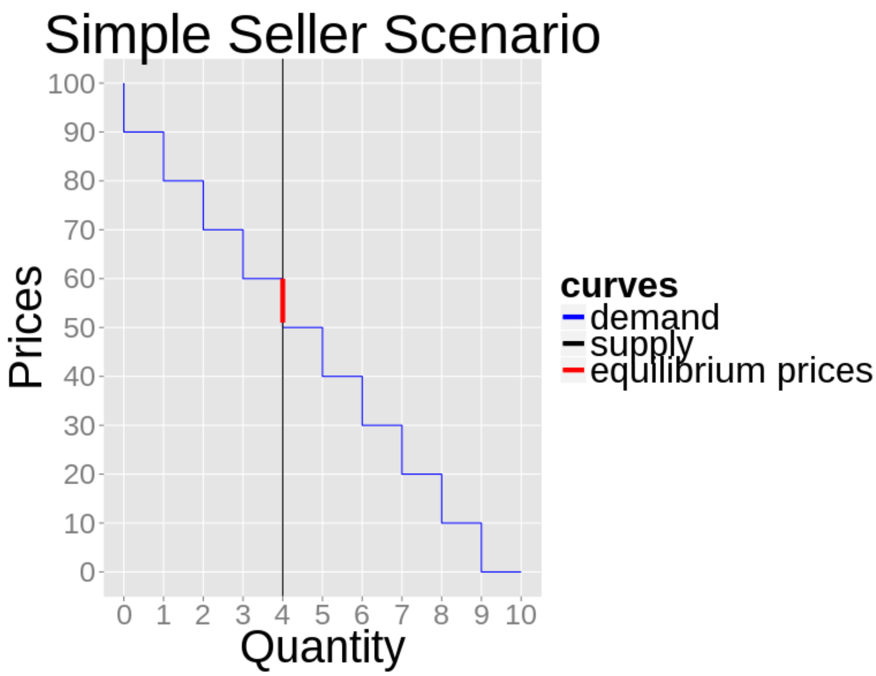


Figure 2.1: The example's daily market demand and supply

The seller starts by charging a random price and then adjusts it daily through its PID. The seller target is to sell all its inventory. Unsold goods accumulate. The seller knows only how many customers it attracted at the end of the day. There is no competition.

With this setup, any price between \$51 and \$60 (both included) will sell the 4 goods to the 4 top-paying customers. Figure 2.2 and figure 2.3 show the market closing prices of two sample runs. In both cases the seller selects the "right" price: \$51.

Notice how, when the initial price is too high as in figure 2.3, the adjustment initially undershoots. Undershooting is caused by the firm trying to dispose leftover inventory from

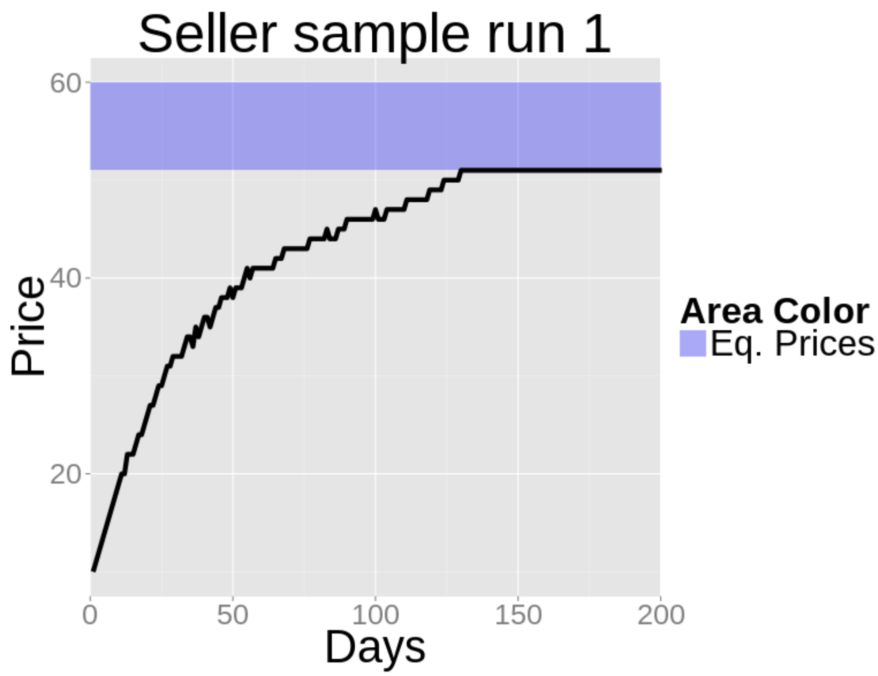


Figure 2.2: The closing prices of a Zero-Knowledge seller sample run when the initial random price is below the equilibrium

previous days; that is, while undershooting, the firm is trying to sell its usual 4 daily goods plus what has not been sold before.

### 2.4.3 PID Parameter Sweep

The PID equation depends on three parameters. Parameter $a$ for the proportional error, $b$ for the integrative error and $c$ for the derivative one. In the previous example the parameters were $a = 0.25$, $b = 0.25$ and $c = .0001$. Here I vary the parameters in turn to show their effects on sellers' behavior.

In figure 2.4 the $a$ parameter varies. An increase in $a$ makes the PID more responsive to today's error. This does not results in a faster approach to true prices but only a more jagged price curve.

In figure 2.5 the $b$ parameter varies. An increase in $b$ makes the PID more responsive to

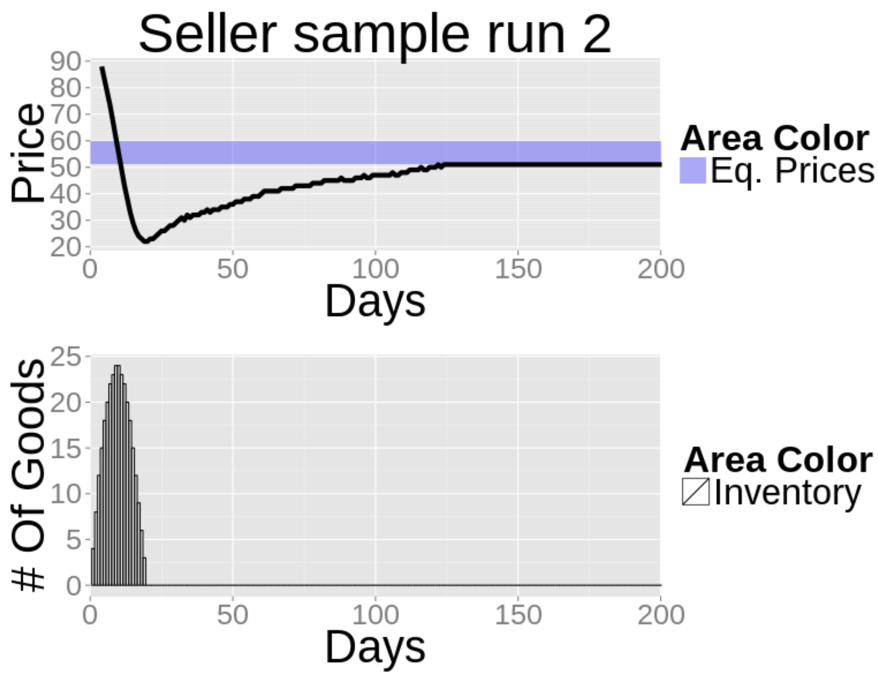


Figure 2.3: The closing prices and inventory of a Zero-Knowledge seller sample run when the initial random price is above the equilibrium

the cumulative sum of errors. This results in a faster approach to the true prices but it can cause fluctuation and overshooting.

Changing the $c$ parameter (even increasing it by 100 times) has almost no effect in this model. The derivative part of the PID becomes important to smooth overshooting which isn't a real issue to Zero-Knowledge sellers because their baseline parameters are very small.

### 2.4.4 Computational Example with Demand Shifts

Agents using PID controllers adapt rather than learn. This keeps them working when market conditions change. Here I replicate the Zero-Knowledge computational example of the previous sections but after 500 days 10 more buyers enter the market. These buyers have a higher demand: the first willing to pay \$190, the second \$180 and so on. The "right" price, after the shock, moves from \$51 to \$151.

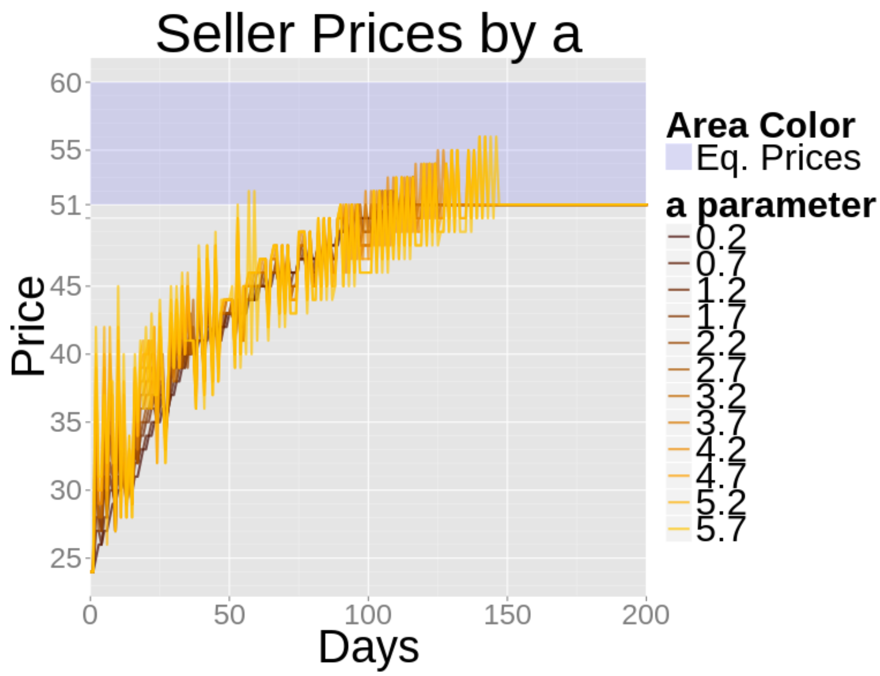


Figure 2.4: The effects of varying the $a$ parameter of a Zero-Knowledge seller

Figure 2.6 shows the sale prices of a Zero-Knowledge seller. The seller quickly finds the new price. Notice here that nothing was changed in the seller algorithm. The PID was not told that the demand had shifted. There is no "structural break" detection. Simply the PID reacts to a changing y (number of customers) by increasing prices to hit the old target.


Figure 2.5: The effects of varying the $b$ parameter of a Zero-Knowledge seller

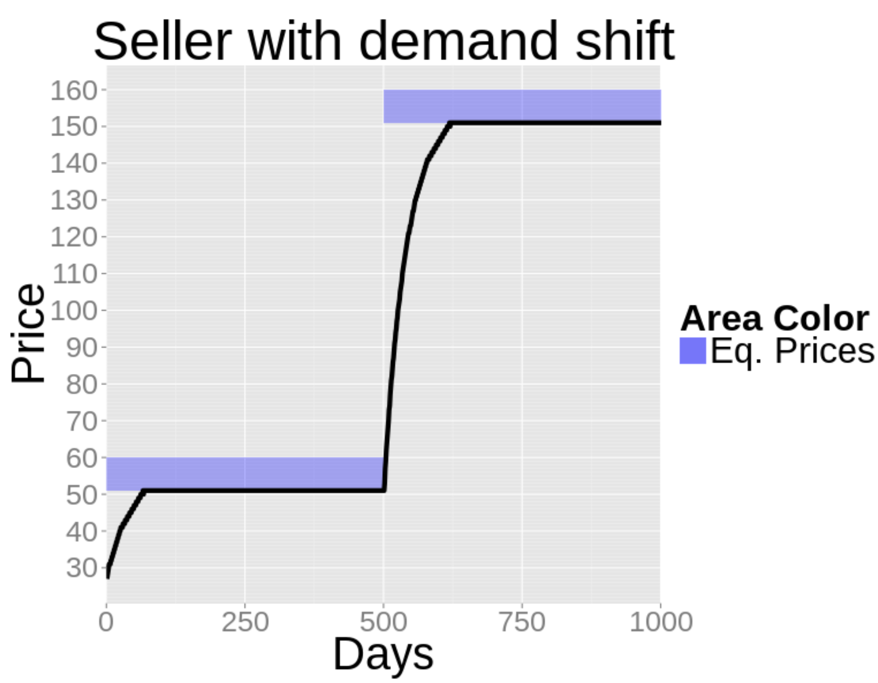


Figure 2.6: The sale prices of a Zero-Knowledge seller dealing with a demand shock after 500 days

## 2.5 Zero-Knowledge Firms

A firm is tasked to maximize its profits by producing and selling its output daily. It has no information on customer demand, labor supply or competition. The firm only knows its own production function. The firm has to decide daily and concurrently the sale price of its goods, the wage of its workers and its production quotas. The problem faced by the firm is harder for two reasons: firstly, it has to trade in multiple markets at the same time and secondly, it is a producer, not a passive receiver of endowment.

Zero-Knowledge firms maximize their profits by dividing the problem into sub-components and solving each separately. There are two equivalent ways to understand this division: by variables or by time as in figure 2.7.

Dividing the profit maximization problem by variables means recognizing that the firm has two kinds of variables to set:

- **Targets**: how much to produce, how much input to buy, how much output to sell, how many workers to hire. See figure 2.9
- **Policies**: how much to offer for inputs, how much to ask for outputs, what wages to offer. See figure 2.8

Rather than setting them all together at once, we proceed in turn. We manipulate policies in order to achieve targets, and we set targets in order to maximize the profit function.

The process of profit maximization of the Firm then is split in two classes of operations:

- Control: change policies to achieve targets
- Maximization: change targets to achieve the objective.

The alternative and equivalent way to subdivide the profit maximization process is by focusing on time. Here I take Hicks's [Leijonhufvud, 1984] division of time in economics between long run (both capital and labor are variable), short run (labor is variable) and market days (production is fixed and unchangeable). Control is the process of managing Hicksian market days: buying, hiring and selling assuming production can't be changed.

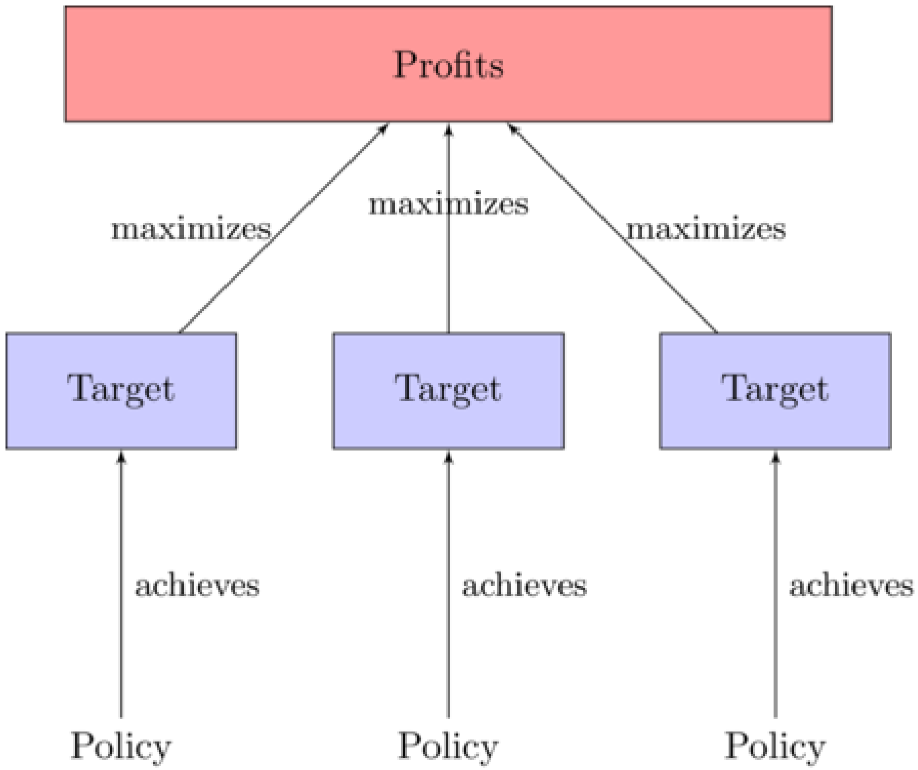


Figure 2.7: The sale prices of a Zero-Knowledge seller dealing with a demand shock after 500 days

Maximization is the process of managing the short run: changing production rate to maximize profits.

The two processes integrate as a feedback loop. The maximization process sets production targets for the controls. The controls, given time, discover the price associated with those targets. The maximization then uses the discovered prices to adjust to new targets and the loop restarts. This is a trial and error alternative to proper backward induction. Backward induction requires the firm to try every possible target, discover the prices associated with each and then choose the target that maximizes profits. Backward induction is exhaustive learning, while the maximization used by the Zero-Knowledge firm is just tinkering.

An example of how the two processes relate in time is shown in figure 2.10. Control

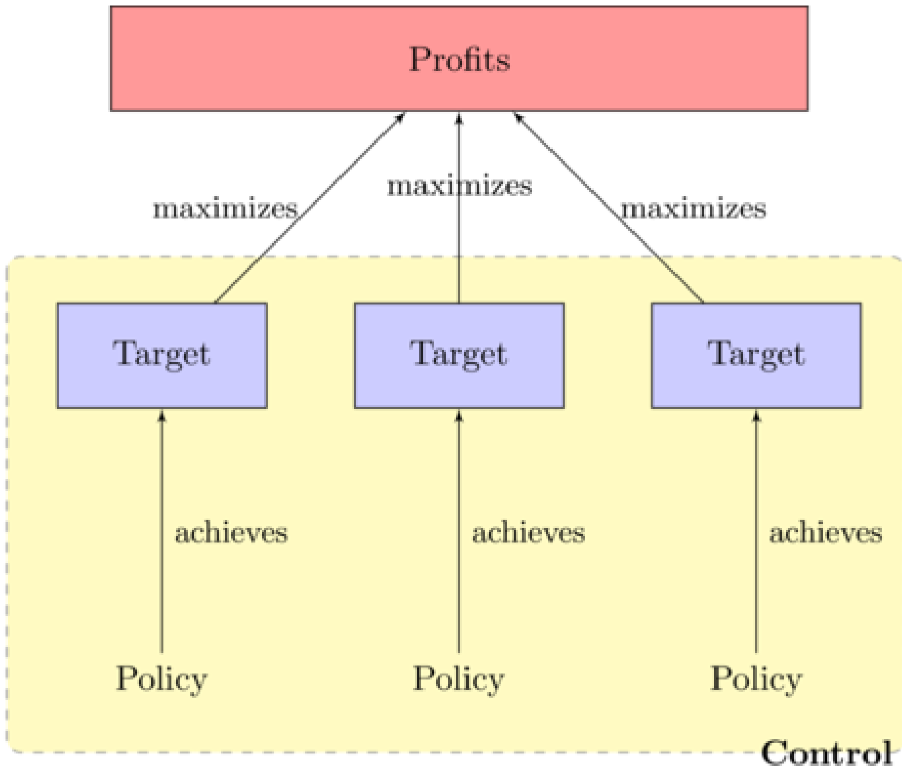


Figure 2.8: Define control as the process of changing policies to achieve targets

happens every day while maximization occurs less frequently to give control time to discover the right prices. In this example the firm revises its production every 3 days. This frequency is arbitrary, and in fact when and how one temporal phase ends and another begins has always been a weakness of Hicks's temporal model [Currie and Steedman, 1990]. I will show in the Section 2.5.2 how to avoid this arbitrariness and link the frequency of maximization with the results from the control process.

### 2.5.1 Control

Control is the process of manipulating policies to achieve targets. In section 2.3 I used a PID controller to solve a univariate problem: manipulate one policy to achieve one target. The Zero-Knowledge firm problem is multivariate as it needs to manipulate the prices for

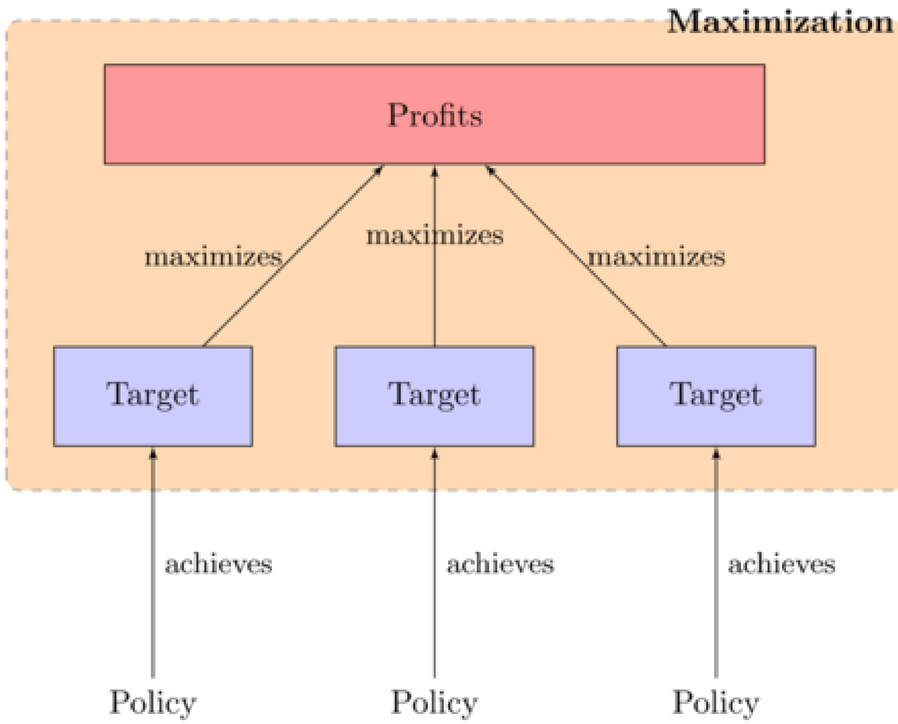


Figure 2.9: Define maximization as the process of setting target to maximize profits

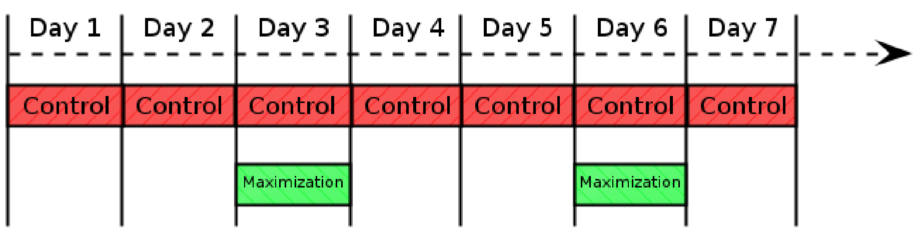


Figure 2.10: An example of how control and maximization processes occur over time. In this case a firm arbitrarily revise its production quota every 3 days, hence maximization on day 3 and 6. The firm needs to buy inputs and sell output every day, hence control every day of the week

all outputs and all inputs.

I solve this multivariate control problem by splitting it into multiple independent univariate control problems. The Zero-Knowledge firm is composed of many Zero-Knowledge traders each achieving a single target with their own PID controller. I call each of these traders a firm's department. This structure is appropriate for object-oriented programming through simple object composition, see figure 2.11.

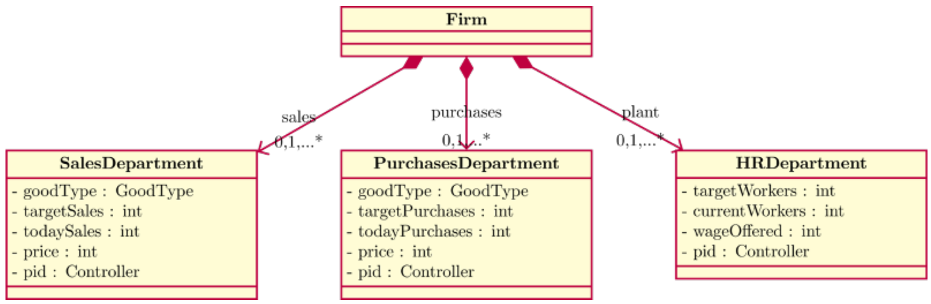


Figure 2.11: UML Diagram of a generic firm

In section 2.3 I showed how the PID controller solves the seller problem. Table 2.2 expands the PID methodology to buying and hiring.

Table 2.2: PID variables for each firm department

| Component | Variable $y$ | Target $y^*$ | Policy $u_t$ |
|---|---|---|---|
| Purchases | # of goods purchased | # of input needed | Price offered |
| Sales | # of customers | # of output produced | Price demanded |
| Human Resources | # of workers | Target # of workers | Wage offered |

If the firm produces more than one kind of goods, then it will have more than one sales department, each focusing on one kind of output.

I chose here to have departments targeting and dealing with flows rather than stocks thus reducing the need for inventory management and so making PID controllers simpler to use and less sensitive to the parameters I set. Focusing on stocks is not impossible, but it is harder and requires more tuning of the $a$ and $b$ parameters to keep the same level of accuracy [Smith and Corripio, 2005].

### 2.5.2 Maximization

Maximization is the process of finding the targets that maximize profits. From section 2.5.1 I know that the firm uses its controls to discover the prices (and therefore the profits) associated with a specific target. The maximization process involves adjusting targets given the information discovered by the control.

For control problems, I used the PID algorithm to adjust policies. I can't use a PID to adjust targets since the error (which in this case would be the distance from maximum profits) is unknown (the firm doesn't know what the maximum profits are). Therefore I use a more rudimentary adjustment algorithm.

I proceed in three steps. First, I simplify the maximization problem from multivariate (many targets to set together) to univariate. Second, I show the adjustment algorithm used to choose this target over time. Third, I define how much time the maximization algorithm should give to controls to discover the prices associated with each target.

Mathematically I want to maximize the profit function $\Pi(\cdot)$ by setting the vector targets $y^*$ :

$$\max_{y_1^*, y_2^*, \ldots, y_n^*} \Pi(y_1^*, y_2^*, \ldots, y_n^*) \tag{2.4}$$

This is a multivariate maximization where each variable is the target of an independent control process. I want to avoid having to explore the whole combination space to find the right target vector, so I am going to condition the targets among themselves to reduce this maximization to a single variable. The main target the firm sets is the number of workers to hire $L$; this is equivalent to setting the daily production quota $f(L)$. I then set sales targets equal to production (sell everything you make) and buying targets equal to daily inputs (buy everything you need). The maximization problem becomes:

$$\max_{L} \Pi(L) \tag{2.5}$$

I use algorithms 1 and 2 to adjust targets after observing profits. Both algorithms are simple hill-climbers to show that no special maximization is required. Both algorithms use little memory, choosing the new workers' target based only on the present and the previous one.

**Algorithm 1** Simple One-Shot Hill Climber maximizer

```
L ← 0                              ▷ Start by having no workers
loop
    oldProfits ← Π(L)              ▷ remember the current profits
    L ← L + 1                      ▷ Increase worker size
    wait                           ▷ Wait for controls to adapt
    if Π(L) < oldProfits then
        L ← L − 1                  ▷ Step back one, this is our final maximum
        break
    end if
end loop
```

**Algorithm 2** Forever Hill Climbing maximizer

```
1: L ← 0                              ▷ Start by having no workers
2: d ← 1                              ▷ We start with positive direction
3: loop
4:     oldProfits ← Π(L)              ▷ Remember the current profits
5:     L ← L + d                      ▷ Tweak the worker force
6:     wait                           ▷ Wait for controls to adapt
7:     if Π(L) < oldProfits then
8:         d ← −d                     ▷ Continue in the opposite direction
9:     end if
10: end loop
```

Line 5 in algorithm 1 and line 6 in algorithm 2 expects the command "wait". This is because controls need time to change policies to achieve the new targets. The wait time can be arbitrary (e.g. one week, one month), but I found it more natural to make it conditional on control achieving targets (e.g. a week after all targets have been achieved). Conditional wait time has the advantage of heterogeneity so that different firms with different controls can use the same maximization algorithm at different frequencies. It is also how I endogenously connect the Hicksian "market days" and "short run" that is the relative speed with which agents change prices and change production targets.

Like with control, having Zero-Knowledge has drawbacks. There are two major drawbacks with this maximization procedure: an economic problem and a practical one.

Trial and error maximization is economically inefficient. Until the profit maximizing targets are found, the firm spends time either under or over-producing. This performance can influence the decision and profitability of suppliers, clients and competitors which are also groping for the right targets. In a Zero-Knowledge setting one agents mistakes can have externalities through the rest of the system.

This maximization is also susceptible to noise due to competition. Both algorithm 1 and 2 are hill-climbers: they compare today's profits with the previous profits. Implicitly I am assuming that if I were to revert back to the old target I would earn the old profit. This stops being true when competitors are concurrently changing their targets. The maximization algorithm thinks it is maximizing $\Pi(L)$ but it is actually maximizing $\Pi(L_i, L_{-i})$ with no control or knowledge of opponents' workforce $L_{-i}$. Each agent decision shifts everybody else's profit function. As a setup it is similar to the "Moving Peaks Benchmark" problem [Blackwell and Branke, 2006] except that peaks are shifted endogenously by each agent rather than by stochastic shocks.

In spite of this I show in the competitive example that the resulting noise is manageable. It stops agents from approaching any steady state, but it does not stop them from approaching equilibrium prices.

## 2.6 Zero-Knowledge Firm Examples

### 2.6.1 Mathematical Example

In this example I use no software. Prices and quantities are continuous. A Zero-Knowledge firm hires workers from a market with daily labor supply $L = 2w$, it has daily production function $q = L$, and faces the daily demand function $q = 100 - 5p$. The firm is composed of two departments, a HR department hiring workers and a sales department selling goods. The departments act daily, in parallel and independently. The firm maximizes arbitrarily every 10 days using algorithm 1.

For the first 10 days, the target number of workers is 1. Table2.3 shows the HR PID process.

Table 2.3: Non-Computational Example of an HR department in a Zero-Knowledge Firm

| Day | HR's $e_t$ | HR's $\sum_{i=1}^{t} e_t$ | Wages $u_t$ | Workers to Hire $y_t^*$ | Workers Hired' $y_t$ | Daily Production |
|---|---|---|---|---|---|---|
| 1 | - | - | 0 | 1 | 0 | 0 |
| 2 | 1 | 1 | .250 | 1 | 5 | .5 |
| 3 | .5 | 1.5 | .325 | 1 | .650 | .650 |
| 4 | .350 | 1.850 | .388 | 1 | .775 | .775 |
| 5 | .225 | 2.075 | .426 | 1 | .853 | .853 |
| 6 | .148 | 2.223 | .452 | 1 | .904 | .904 |
| 7 | .096 | 2.319 | .469 | 1 | .937 | .937 |
| 8 | .063 | 2.382 | .479 | 1 | .959 | .959 |
| 9 | .041 | 2.423 | .487 | 1 | .973 | .973 |
| 10 | .027 | 2.450 | .491 | 1 | .982 | .982 |

At the same time the sales department is using its own PID controller to sell products. The target sales is equal to daily production (which is driven by the HR department) plus leftover inventory. For this example I force initial sale price to be 20.

Table 2.4: Non-Computational Example of a Sales department in a Zero-Knowledge Firm

| Day | Sales's $e_t$ | Sales's $\sum_{i=1}^{t} e_t$ | Sale Price $u_t$ | Daily Production | Goods to sell $y_t^*$ | Customers Attracted $y_t$ |
|---|---|---|---|---|---|---|
| 1 | · | · | 20 | 0 | 0 | 0 |
| 2 | 0 | 0 | 20 | .5 | .5 | 0 |
| 3 | -.5 | -.5 | 19.875 | .650 | 1.150 | .626 |
| 4 | -.525 | -1.025 | 19.769 | .775 | 1.3 | 1.156 |
| 5 | -.144 | -1.169 | 19.759 | .853 | .996 | 1.205 |
| 6 | .208 | -.960 | 19.818 | .904 | .904 | .908 |
| 7 | .004 | -.956 | 19.809 | .937 | .937 | .955 |
| 8 | .018 | -.938 | 19.813 | .959 | .959 | .934 |
| 9 | -.025 | -.963 | 19.806 | .973 | .998 | .970 |
| 10 | -.029 | -.992 | 19.800 | .982 | 1.011 | .999 |

At the end of Day 10 the maximization algorithm is called and compares profits with 0 workers (which is 0) against the profits with 1 worker target. The firm paid .491 in wages to .982 workers, for a total cost of .482; the firm produced .982 goods sold at 19.8 a unit for a total revenue of 19.443. The firm's daily profits then are 18.961. Because increasing workers increased the profits (from 0 to 18.961) the maximization algorithm sets the new worker target to be 2. 2 is set as target to the HR department from Day 11, restarting the loop.

The two departments are linked only through production: HR gathers the input, the sales department sells its production. In this particular example, and in the computational examples that follow, HR and production have priority over sales and always happen before. This is not an important assumption, if the sales department acts first the process is identical except that sales actions are delayed by one day (what happened in day 3 will happen in day 4 and so on). In chapter 3 the order is reversed but as section 3.4.1 shows, the results are the same.

### 2.6.2 A Monopolist Example

There is a single firm with two departments: a sales department and an HR department. There is a fixed daily demand for goods as shown in figure 2.12. The demand is step-wise and discrete. The firm also faces a step-wise discrete supply curve made up of individual workers' reservation wages. The firm must pay a single wage to all employees which explains why the marginal cost curve is steeper than the wage curve (the second worker has reservation wage $16, but hiring him requires raising the first worker wage by $1, hence the marginal cost is $17).

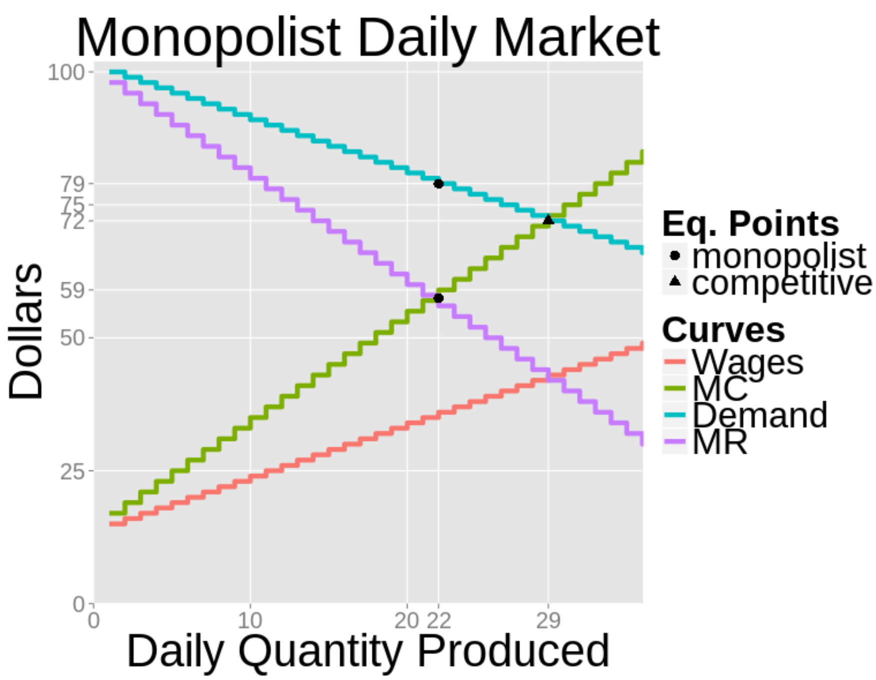


Figure 2.12: The daily demand faced by the monopolist, the wage curve and the resulting marginal cost curve

Production is constant returns: each worker produces 1 unit of good every day. There is no capital, no fixed costs and no other inputs. Market is an order book. Everybody places limit orders and crossing orders are automatically filled. The trading price is always the price quoted by the seller. Prices and quantities are always natural numbers. A rational monopolist maximizes profits by hiring 22 workers. The rational monopolist price is $79.

The Zero-Knowledge firm has none of this information. The firm has no knowledge of being a monopolist either. Initially the wage offered is set to 0, the sale price is set to 100. The maximization used is algorithm 1. The maximization wait time is endogenous: 3 weeks after the labor targets have been filled by the HR department.

The firm's daily production and sale price in a sample run are shown in the figures 2.13 and 2.14. The Zero-Knowledge firm acts rationally in spite of no knowledge, uncoordinated departments and rudimentary maximization.

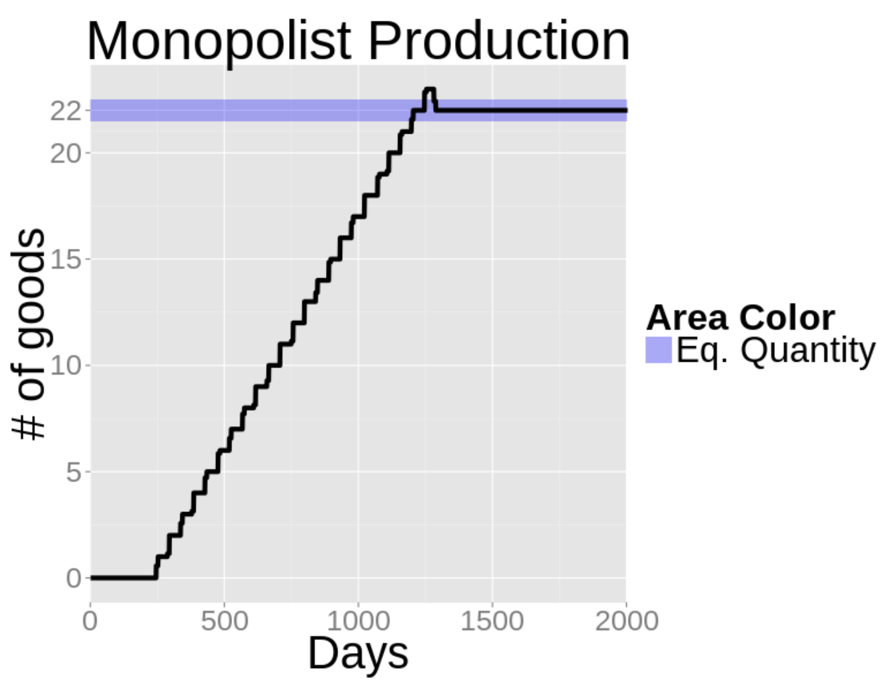


Figure 2.13: Daily production in a sample run with a single firm

Notice in figure 2.14 the same temporary undershooting as in the Zero-Knowledge seller

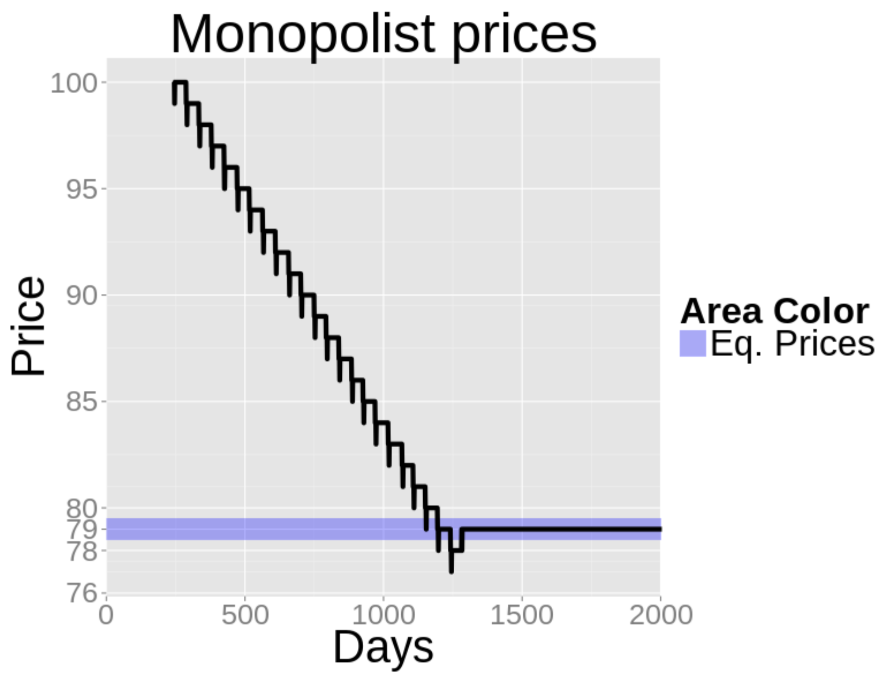


Figure 2.14: Daily production in a sample run with a single firm

example; this undershooting has a different cause: the sales department PID has no foreknowledge of different changes in worker targets. In a way the sales department is continually surprised by changes in production and its PID controller has to catch up. It is the cost of using completely reactive control and total departmental independence.

The results are only slightly different if I use algorithm 2. In this case the firm forever oscillates between hiring 21, 22 and 23 workers ad libitum.

### 2.6.3 A Competitive Example

I replicate the market of the section 2.6.2 and add competition. In this example there are 5 firms in the market. Nothing changes in the internal structure of the firm. The firms have no knowledge of having competitors. Each firm follows algorithm 2 to maximize. The competitive equilibrium price would be \$72 and the equilibrium daily production would be 29.

Figure 2.15 and figure 2.16 show a sample run. Unlike the monopolist case, the results are more noisy and do not stabilize. Both the quantity traded and the prices orbit around the equilibrium values, but they never settle.

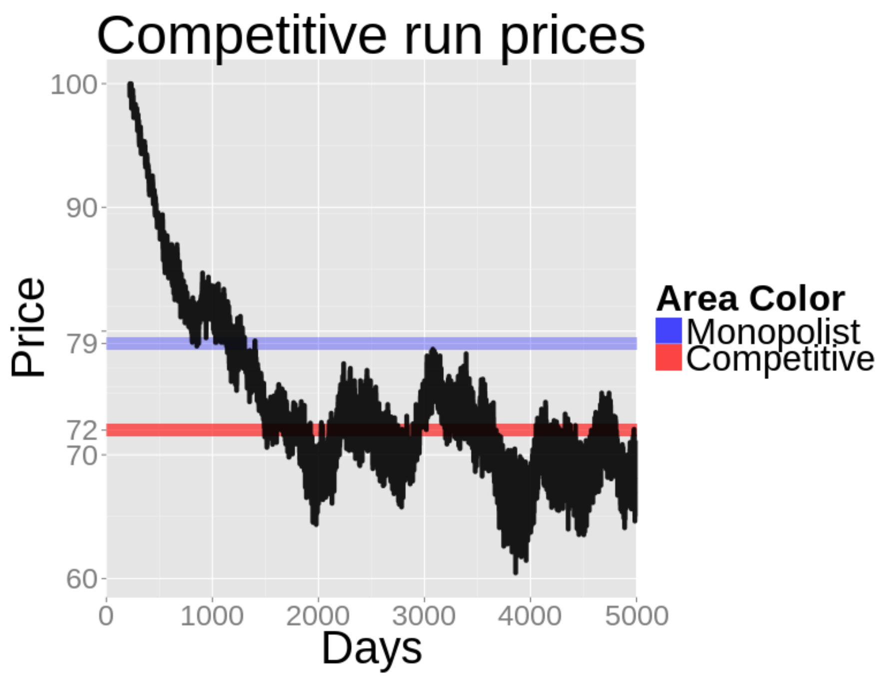


Figure 2.15: Daily prices in a sample run with 5 firms

I run the competitive model 5000 times changing only the random seed. I stop each simulation after 5000 days and record final price and quantity. Figure 2.17 shows the distribution of results. While dispersed, all observations cluster around the market demand function. This shows how with competitive noise, control keeps performing well in keeping production and price linked even when the maximization fails to find the profit maximization quantity. If I focus on prices alone, as in figure 2.18 I can see that almost all the simulations with competition have prices lower than than the monopoly setup.

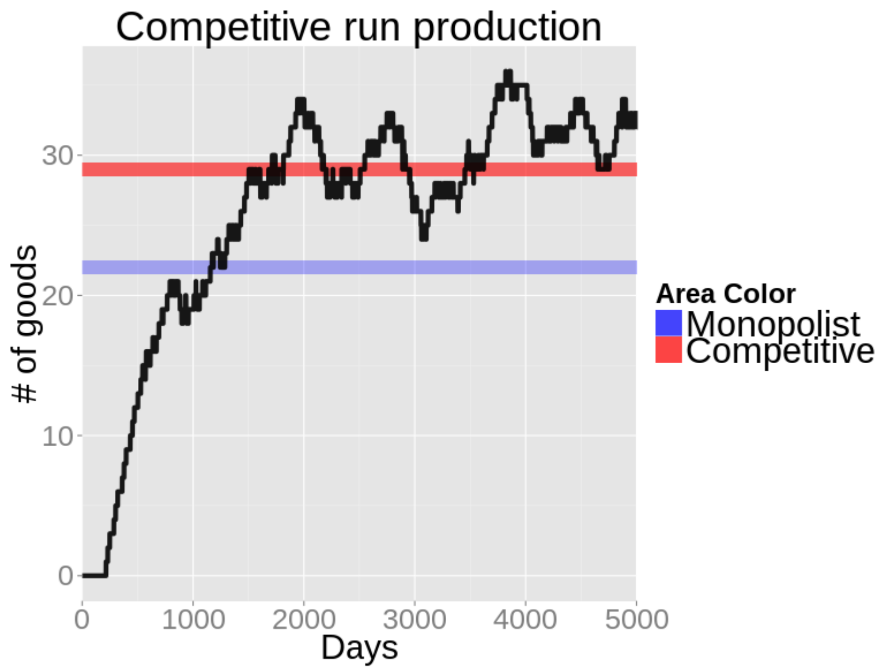


Figure 2.16: Daily production in a sample run with 5 firms

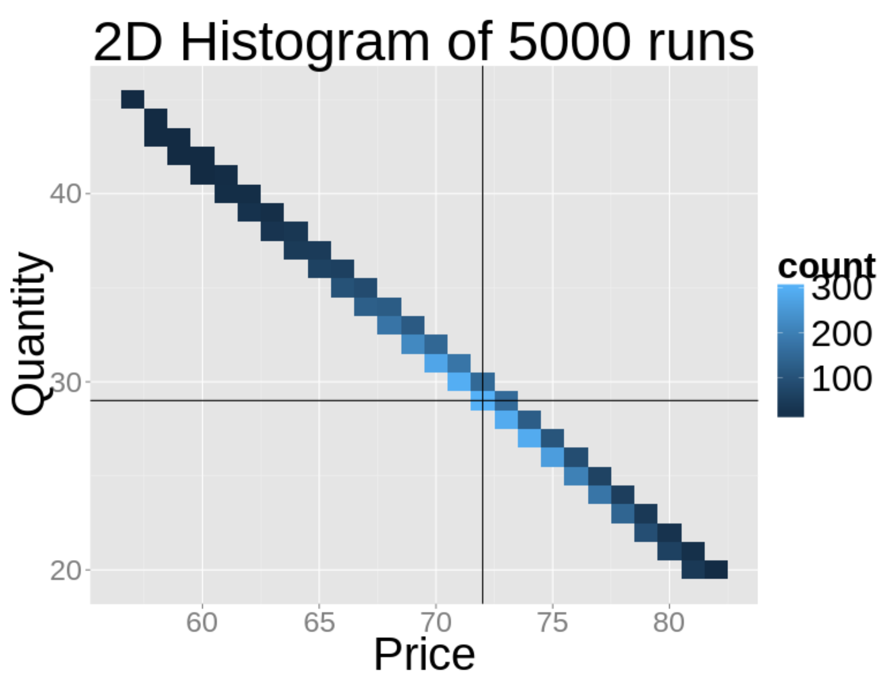


Figure 2.17: The 2D histogram of price-quantity results of 5000 sample runs of the competitive scenario

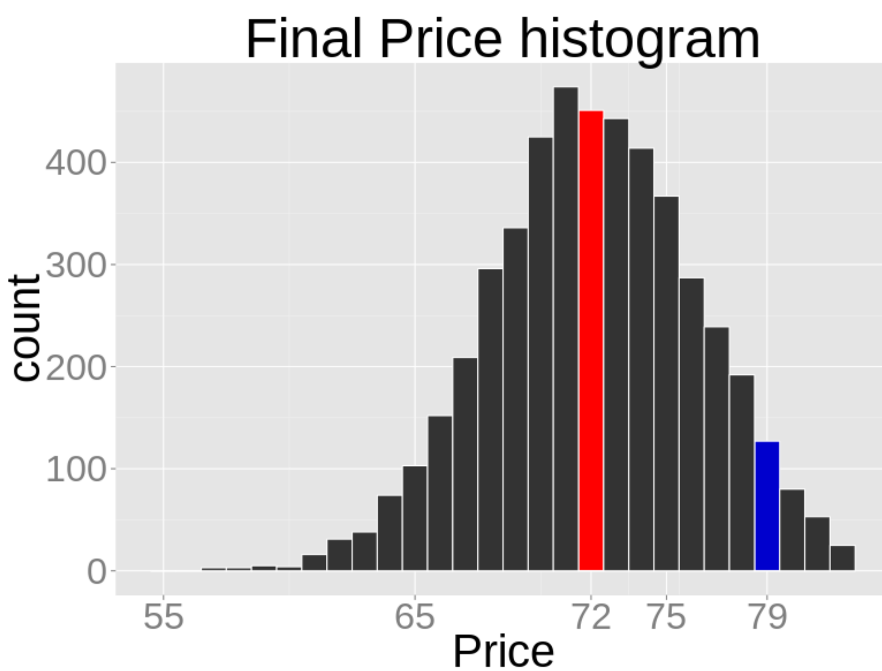


Figure 2.18: The histogram of prices from 5000 competitive runs. The red bar represents the theoretical competitive prices, the blue bar the theoretical monopoly prices

## 2.7 Conclusion

The Zero-Knowledge agents I built solve the monopolist problem in simple markets perfectly but are inaccurate in perfectly competitive markets. This is in part because the hill-climbing algorithms 1 and 2 react to past profits and prices rather than current ones. In chapter 3 and in particular section 3.4.1 I reduce the competitive noise of these simulations with agents using marginal profit analysis and current PID prices.

There are three assumptions that power the Zero-Knowledge traders that I think should be addressed. First, I have decided that firm's trial and error is done over prices. Thanks to economics surveys by [Blinder, 1998] and [Fabiani, Silvia et al., 2006] I know that price flexibility is uncommon. Prices are more like targets, changing perhaps three times a year. In chapter 3, I give a microfoundation to why prices might move more slowly and in chapter 4 I show the macro-economic effect of using prices as targets and quantities as policies.

Second, while Zero-Knowledge firms were created to show how agents can bootstrap correct behavior without looking at prices, there is no reason to assume agents are so autistic. Benchmarking is common-place in any industry. A more realistic model would use more feed-forwarding and, more importantly, more nuanced optimization.

Third, I assumed that demand reacts immediately to changes in prices. This is in line with usual economic assumptions but it has the additional advantage of avoiding the complicated design of controllers that deal with delays between policy changes and results.

In spite of its simplicity, agents with this behavior can provide a simple baseline on which to build other economic agent based models.

# Chapter 3: Sticky Prices Microfoundations in a Agent Based Supply Chain

## 3.1 Introduction

### 3.1.1 Motivation

As economists we see price flexibility as efficient and stickiness as an inferior compromise imposed by adjustment costs. I build a simple model with no adjustment costs where price stickiness is not only superior but necessary to achieve equilibrium. The model runs on two assumptions: delays to adapt to price changes and bounded rationality.

Consider an economy in disequilibrium where some goods are overproduced and others overconsumed. Prices have to change to signal agents to reallocate resources. What I focus on is how much time passes between the price changing and the agents reacting to it. The delay between a price changing and it having effect exists only with boundedly rational agents. All-knowing agents would predict any disequilibrium and adapt preemptively.

Agents in this paper are instead trial and error price-makers. When placed in a one-sector economy with immediately reacting demand, agents quickly find equilibrium price and quantity. When placed in a two-sector supply chain, prices spiral out of control. This is because firms downstream need time to adapt to a change in price upstream. This delay feeds into the trial and error of the upstream firms fooling them into thinking that prices are inelastic. To counter this inelasticity, upstream firms try ever larger price changes eventually overshooting and undershooting out of control. Sticky prices restore equilibrium by giving the time to agents to see their actions' full effect.

It is common to assume that prices can't be away from equilibrium for long. Traders would notice shortages or gluts and react by adjusting prices. In undergraduate textbooks this thinking is taken as an explanation itself of what equilibrium is, see for example page 82 of Mankiw's microeconomics textbook [Mankiw et al., 2011], and more in general this is taken as a license to ignore disequilibrium altogether and assume market-clearing prices [Conlisk, 1996]. I instead make explicit this adjustment process and show how it works well in simple markets but not in supply chains. With price-stickiness equilibrium emerges in supply chains as well. Price-stickiness here is not a poor substitute of total flexibility, it is necessary for agents to deal with a slowly adapting world.

### 3.1.2 Research Contribution

Here I tie together two separate academic literatures. The first is the "bullwhip effect": the large swings in prices we observe in supply chains that cannot be explained by changes in final demand [Baganha and Cohen, 1998]. The second is "sticky prices": the slowness in changing prices that we observe in macro-economics in spite of evident changes in the final demand [Klenow and Malin, 2010].

The operation research solution to bullwhip effects in supply chains is to active management by centralizing information [Chen et al., 2000]. Taking the whole economy as an input-output table [Leontief, 1966] each economic sector is linked to the others in a supply-chain like structure that generates similar bullwhip effects. However these sectors are so large and the agents so dispersed that it's impossible to manage this structure by centralizing information. I show then that prices can coordinate the supply-chain with no information sharing but only as long as the prices are sticky.

While my chapter generates and explains bull-whip effects, the main thrust is on how to fix them when information is not available or cannot be processed. For this reason I use simple agents with very limited rationality. It allows me to show how sticky prices are the method to coordinate supply chains and have them reach equilibrium. Sticky prices require little rationality and little information and are therefore perfect to manage the general supply chain that is the entire economy.

### 3.1.3 Roadmap

This chapter is split into three main parts. The first part goes from the literature review in section 3.2 to section 3.4.1. In it I summarize and expand the Zero-Knowledge methodology. First in section 3.3, I show how agents can price their output through trial and error. I then show how when there is a delay between price setting and demand adjusting to it, the trial and error rule oscillates away from equilibrium. Finally I show how price stickiness can recover the equilibrium. In section 3.4.1, I add production: agents can set their own production targets in order to maximize profits. Through marginal analysis agents are capable of reaching both monopolist and competitive equilibria.

The second part, section 3.5-3.6 deals with supply chains. In section 3.5, I plug the Zero-Knowledge traders into a supply chain. Because of the way the production targets downstream are set, the firms upstream face delayed demands. Again prices oscillate away from equilibrium and again price stickiness can be used to recover the equilibrium. In section 3.6 I go through various market structures for the supply chain and show how the results are robust to changes in market power.

The third part goes from section 3.7 to the conclusion. In this part of the chapter I cast off some of the assumptions I made in the previous sections. I show how removing those assumptions creates noisier results but the overall outcome is the same: supply chains keep on achieving equilibrium given sticky prices. I believe these sections are important as a robustness check on the main results of the previous sections. In section 3.7 I let agents discover on their own if they are in a competitive or monopolist market. In section 3.8 I let agents set their own price stickiness. Finally in section 3.9 I show how one should structure empirical work around Zero-Knowledge traders.

The source code is available[1] on an open-source MIT license. This Zero-Knowledge framework is described more completely in [Carrella, 2014]. The simulation is coded in Java and uses the MASON toolkit [Luke et al., 2003].

[1] https://github.com/CarrKnight/MacroIIDiscrete

## 3.2 Literature Review

The ideas here are thoroughly unoriginal. I apply the principle that "in a world of price makers, rather than auctioneers and price takers, it takes time and resources to change prices", [Okun, 1981] description of Hicksian fixprice, on agents that are "goal-oriented feedback mechanisms with learning" which is [Pickering, 1995] definition of cybernetics.

My chapter shares some similarities with the Beer-Distribution game [Sterman, 1995]. Both deal with supply-chains, both have agents that act by feedback and both result in noise and disequilibrium. But the similarities end there. The fundamental difference is that my model has prices. In the beer distribution game, each node of the supply chain is powerless to influence the number of orders it receives, my firms can adjust their sale prices to throttle their customers' demand. Other components of the beer-distribution game are missing: here there is no exogenous shock to demand, no delay in clearing orders, no anchoring or "wrongful mental simulation"[Sterman, 1989]. The focus is entirely on prices and disequilibrium.

this chapter's result also shares some superficial similarity with section 4.4 of "Information Distortion in a Supply Chain" [Lee et al., 2004]. In it the authors describe the strategy of "Every Day Low Price" where manufacturers reduce the frequency of discounts and promotions in order to stabilize supply chains. In both papers therefore rigid prices alleviates bullwhip effects. The causal mechanisms of the two papers are very different however. In their paper low price variation is a counter-measure to forward and strategic buying downstream that are incentivized by discounts and promotions. It is fundamentally a micro-economic result of managing other parties expectations. My chapter has no forward or strategic buying and focuses instead on the steady prices coordinating low rationality agents in a low information environment.

The most extensive empirical review on sticky prices is [Klenow and Malin, 2010]. The firm interviews by [Blinder, 1998] and [Fabiani, Silvia et al., 2006] are both literature surveys on price stickiness microfoundations and empirical tests of which theory firms find more credible. Both highlight the importance of returning customers' goodwill [Okun, 1981], coordination between firms [Clower, 1965] and long term contracts.

My chapter is most similar to [Blanchard, 1982]. In both papers, price inertia is due to the desynchronization between firms in a supply-chain. The results are similar but the causality is reversed. There production adapts instantaneously but prices are set slowly and asynchronously which generates inertia within the supply chain. Here there is production inertia in the supply chain so that while prices can be set quickly it is counterproductive to do so.

In sticky-information models [Mankiw and Reis, 2002] the lack of knowledge alone can cause price-stickiness as information about shocks is expensive. The difference between this chapter and the sticky-information literature is on how information gathering is modeled. In the sticky-information literature information gathering is a separate activity from trading: in the original Mankiw's paper a random proportion of firms receive information each day. In this chapter information gathering is a by-product of trading. Firms set prices and see how many customers they attract; firms can only discover the demand by experimentation. Sticky prices become necessary when there is a large delay between running the experiment (setting a price) and seeing its result (changes in behavior along the supply-line).

## 3.3 Zero-Knowledge Traders and Delays

In this section I introduce the Zero-Knowledge traders and explain how they find the correct prices when selling a fixed daily inflow of goods. This trial and error mechanism is used for the rest of the chapter. I then break the Zero-Knowledge pricing by adding arbitrary delays to how quickly the demand adapts to changes in prices. I finally show how sticky prices can solve the delays.

### 3.3.1 Trial and error pricing is effective when customers react immediately

Zero-Knowledge traders price their goods in a feedback loop. Every day a trader receives $q^*$ goods to sell. In the morning, the trader sets sale price $p$ and during the day it attracts

$q$ paying customers. If at the end of the day there are fewer customers than goods to sell, the trader will lower tomorrow's price. Defining the daily error as

$$e_t = q_t - q_t^* \tag{3.1a}$$

$$e_t = \text{Outflow} - \text{Inflow} \tag{3.1b}$$

$$e_t = \text{Netflow} \tag{3.1c}$$

The trader adjusts tomorrow prices through a PI(Proportional Integrative) controller rule:

$$p_{t+1} = ae_t + b\sum_{i=1}^{t} e_i + p_0 \tag{3.2}$$

Where $p_0$ is the initial random offset.

Start with a simple example: a price-maker agent receives 50 units of good each day to sell, that is $q_t^* = 50, \forall t$. It faces a fixed but unknown daily demand curve $q_t = 101 - p_t$. Imagine that the agent starts with random initial price $p_0 = 80$. The first day the trader attracts $q_0 = 21$ customers and since its target was 50 sales its first error is $e_0 = -29$. The agent then plugs this error in the PI controller formula:

$$p_1 = a * (-29) + b * (-29) + 80$$

Assuming $a = b = .1$ the agent sets $p_1 = 74.2$. In this chapter I allow only natural prices, so that $p_1$ is rounded to 74. The next day the seller attracts $q_1 = 27$. This generates the error $e_1 = -23$ which can be plugged in the PI controller to generate $p_2$:

$$p_2 = a * (-23) + b(-29 - 23) + 80$$

Figure 3.1 shows the simulated path generated by these parameters. The agent quickly finds the correct price.

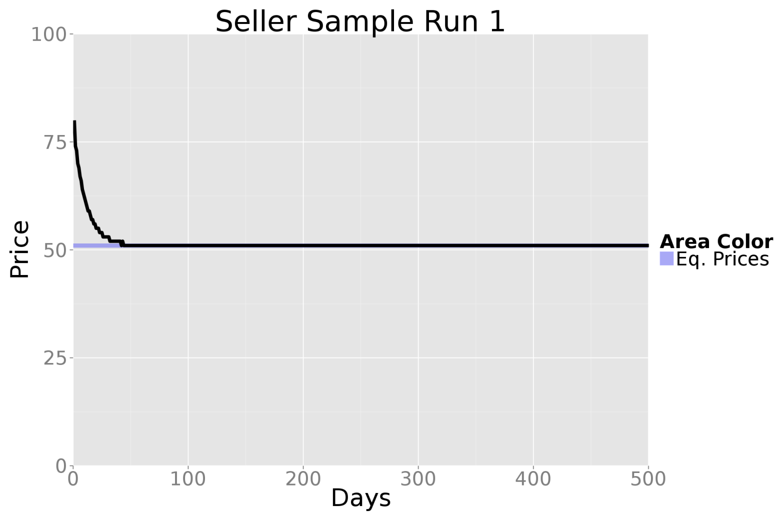


Figure 3.1: A sample run of a trader iteratively finding the correct prices when having 50 units of goods to sell and facing the daily linear demand $q_t = 101 - p_t$. This trader is using a PID controller with parameters $a = b = .1$.

### 3.3.2 Trial and error pricing fails if there is a long delay between setting a new price and it having effect

PID controllers simulate naïve trial and error pricing. As with all experimentation, PIDs work better when trial results are informative and unambiguous. A simple way to mislead the agents is to add a time delay $\delta$ between a price $p$ being set and the quantity demanded $q$ adjusting to it.

Take a delayed demand curve, that is the quantity demanded at time $t$ is a function of the price at time $\delta$ days before:

$$q_t = f(p_{t-\delta}) \tag{3.3}$$

This delay is completely arbitrary and exogenous, I add it here to expose a weakness of adapting prices through PI controllers. A more endogenous source of delay is introduced in section 3.5.

Delays mean that even when the trader guesses the right price it takes $\delta$ days to yield the right quantity. This fools the trader into thinking the correct price is wrong and changing it. Depending on $\delta$, the delay can slow down the approach to real prices as in figure 3.2 or prevent it entirely as in figure 3.3.

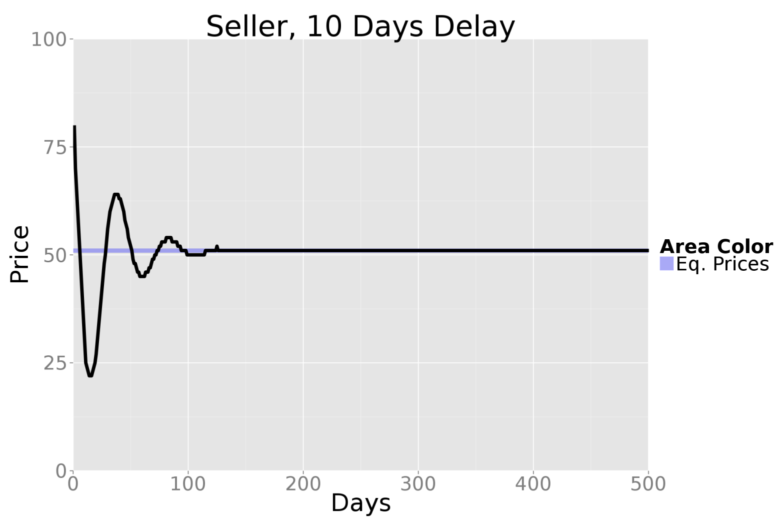


Figure 3.2: The same trader of Figure 3.1 now faces demand $q_t = 101 - p_{t-10}$. The trader takes longer to find the right price.

### 3.3.3 Sticky prices are a solution to the price delay

The simplest way to deal with delays is to slow down the trial and error loop accordingly. If it takes a week for prices to have an effect, the trader can change prices every week rather than every day. Effectively, sticky prices. An agent using sticky prices continues to follow the same PI formula as equation 3.2 but does not activate it every day. Define the stickiness of an agent as $s$ days, the agent activates its PI controller to change prices each day with

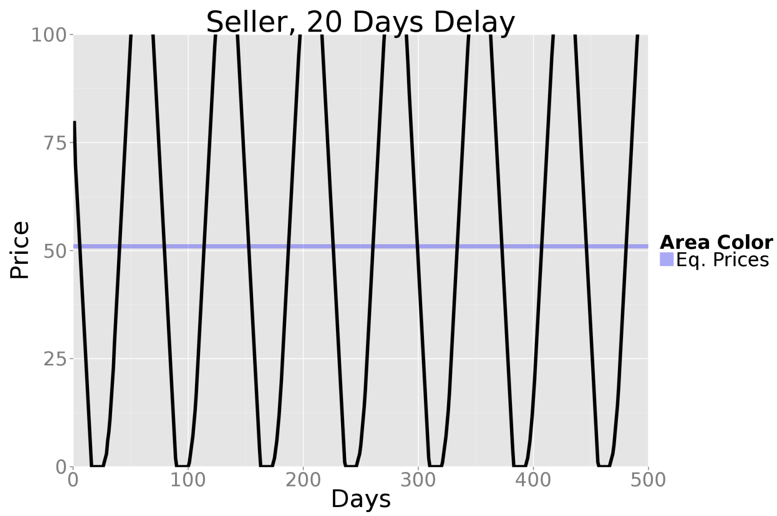


Figure 3.3: The same trader of figure 3.1 now faces demand $q_t = 101 - p_{t-20}$. The trader never finds the right price.

probability $\frac{1}{s}$. In other words $s$ is how many days on average pass between one change of price and another. This stochasticity is necessary to avoid creating spurious artifacts in simulations with multiple agents as they would otherwise proceed in lock-step.

Notice the significance of activating the PI controller only some days. The PI controller is there for the agent to adapt prices when they are perceived as wrong, that is when there is a non-zero error. By not activating the PI controller at any given day I am forcing the agent to maintain "wrong" prices. Why would it ever pay for an agent to keep an obviously wrong price? The answer is that the price is perceived as wrong today but it might not be wrong in the future. If the demand contains a delay, keeping prices constant allows the true demand $q$ associated with current price $p$ to emerge.

As shown in figure 3.4 adding stickiness $s = 20$ to an agent's pricing is enough to get back to equilibrium. Alternatively the trader can keep changing prices every day by small amounts so that the demand has time to catch up. This would mean using the PI equation 3.2 with small $a$ and $b$. This also works as shown in figure 3.5

I defined price stickiness $s$ as the average number of days the agent waits before activating its PI controller to change prices. Define timidity $z$ as the number dividing the baseline

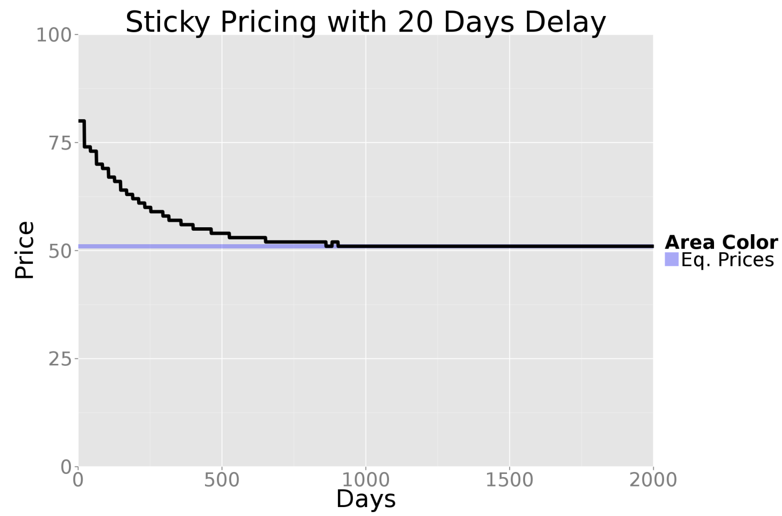


Figure 3.4: The same setting of figure 3.3 but this time the trader adjusts her prices only every 20 days. Notice that the time between one price change and the next is irregular, this is because there is a fixed $\frac{1}{s}$ chance of activating the PI controller each. The result is the same approach as figure 3.1 but in a longer time frame.

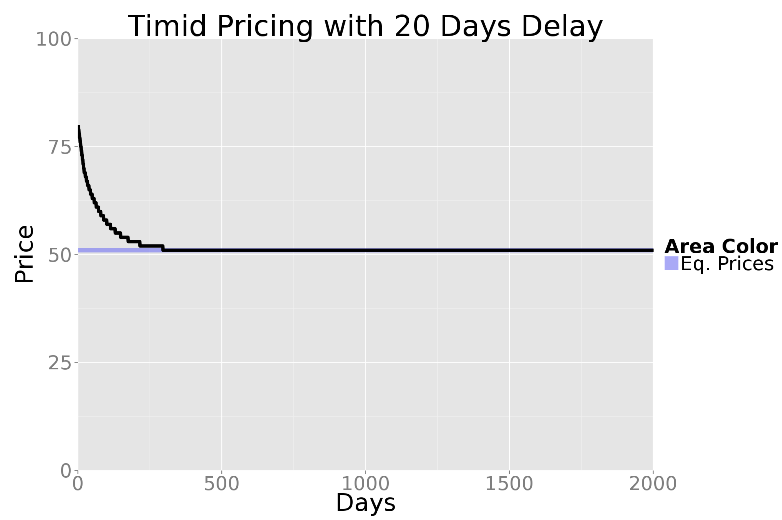


Figure 3.5: The same setting of Figure 3.3 but the PID controller has $a = b = .01$, 10 times smaller than the original.

PI parameters (so a timidity of 10 with a baseline of .1 means that the PI parameters

$a = b = \frac{1}{10} = .01$). This means changing the PI formula in equation 3.2 to:

$$p_{t+1} = \begin{cases} p_t, & \text{with probability } 1 - \frac{1}{s} \\ \frac{a}{z} e_t + \frac{b}{z} \sum_{i=1}^{t} e_i + p_0, & \text{with probability } \frac{1}{s} \end{cases} \tag{3.4}$$

Fix the demand delay $\delta$ to 50 (this is in order to better show the interplay between stickiness and timidity). Figure 3.6 shows which combinations of timidity and stickiness achieve correct prices over 5 experimental runs. Define the daily distance from the correct price as:

$$\sum_{t=1}^{T} (p_t - p^e)^2 \tag{3.5}$$

Using the demand $q_t = 101 - p_{t-50}$ the equilibrium price is $p^e = 51$. Figure 3.7 shows for which combination the distance is minimized. That is which combination of timidity and stickiness achieves the fastest convergence to correct prices.

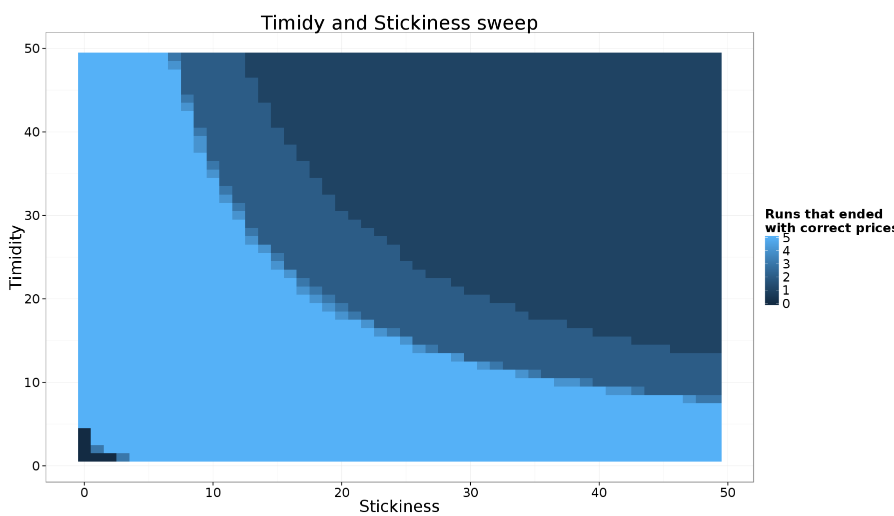


Figure 3.6: Run the model 5 times for 15000 market days with fixed PID parameters and speed but different initial prices. Controllers that are too fast or too slow fail in at least some cases. Demand delay is 50 days

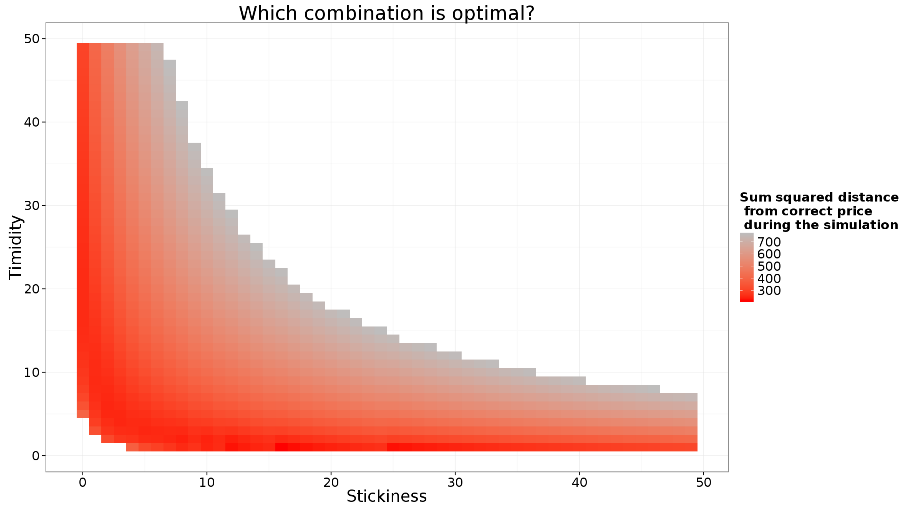


Figure 3.7: Average sum squared distance over 5 simulations run. The minimum distance is achieved by stickiness of 16 and timidity of 1. Only the successful combinations from figure 3.6 are considered. Demand delay is 50 days

Agents working by trial and error benefit from acting slowly and timidly whenever price changes take time to have an effect. Since there are no menu costs, the agents are indifferent between adjusting the price often but timidly or seldom but aggressively. The weakness of this section is that the delay is arbitrarily fixed and exogenous. The delay will become endogenous as supply chains are introduced in section 3.5.

### 3.3.4 We can reduce knowledge further with a minimum inventory buffer

One weakness of the error in equation 3.1 is that it assumes netflow can go negative. If the seller manages to sell all its stock, she must estimate how many more goods she would have been able to sell at that price. This is unrealistic. An alternative is to hold a minimum inventory to count excess customers.

The advantage of having an inventory is that we can change the error we feed in the controller from

$$e_t = \text{Outflow} - \text{Inflow} \tag{3.6}$$

to just

$$e_t = -\Delta\text{Inventory} \tag{3.7}$$

The disadvantage is that the seller must build up an inventory.

Remember that in this section the seller receives a fixed amount of goods $q^*$ every day. She then needs a strategy to stock up a sufficient level of inventory. Define $i^*$ as the minimum buffer inventory level. The PID controller targets 0 sales as long as actual inventory is below $i^*$ and targets $\Delta\text{Inventory} = 0$ otherwise. Excluding the initial days of stocking up the dynamics of this controller are exactly the same as those shown above. Traders will use inventory buffers for the rest of the chapter.

### 3.3.5 Smoothing prices through moving averages does not solve demand delays

I showed above how demand delays cause prices to oscillate as in figure 3.3 and then how stickiness and timidity can overcome such issues. Because the price oscillation is very regular in the example shown, it might seem possible to avoid dealing with stickiness and instead smooth the PI controller's prices through a moving average. This unfortunately does not work: moving averages do not solve oscillations caused by demand delays.

Notice first that there are two variables we can smooth. Either smooth the error $e_t$ to feed into the PI controller or the policy $p_t$ that comes out of it. In control theory, these are called respectively "process variable filtering" and "controller output filtering". Filtering $e_t$ through an arithmetic moving average makes the delay problem worse. This is because averaging out $e_t$ with previous values make the PI controller deal with a delayed version of its input. The demand delay is what fooled the PI controller into oscillating its prices in the first place and smoothing $e_t$ increases the delay faced by the PI controller and therefore the amplitude of its price oscillations.

Understanding why smoothing the PI output $p_t$ also has no positive effects help explain why stickiness and timidity work instead. Demand delays mean that the error $e_t$ fed into the controller does not accurately reflect the effect of the price $p_t$ on the demand. Over time the errors $e_t$ accumulate in the integrative part of the controller, that is $b\sum_{i=0}^{t} e_t$. It is the progressive increasing and then unwinding of the integrative part that causes prices to oscillate. Stickiness fixes this by only updating the integral part every $s$ days, timidity fixes this by dividing the effect of the integrative part of the controller. Smoothing out the PI output instead has no effect on the integrative controller; it slows down the output of the controller but not the accumulation of errors in its integrative part. The PI controller has a sluggish output but all this does is to give more time to the integrative part of the controller to increase and eventually this amplifies the price oscillations.

Figure 3.8 shows the prices generated by a PI controller filtering either its input or output by a 20-day moving average when facing a demand delayed by $\delta = 20$ days. The prices generated are not better and the oscillations are deeper, regardless of the filter used. Stickiness and timidity work better than filtering.

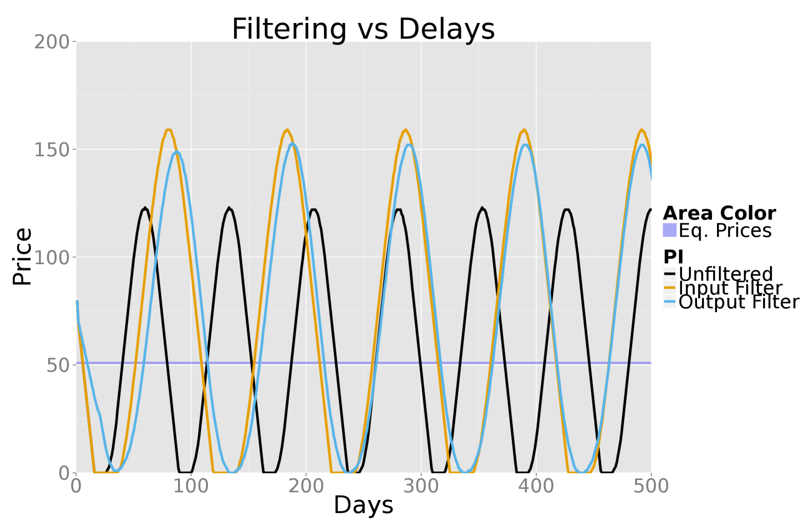


Figure 3.8: I compare here the effect of adding a 20-days moving average filter to the input $e_t$ or the output $p_t$ of a PI controller facing a 20 days delayed demand. The filters do not improve the controller output and in fact increase the amplitude of the price oscillations

## 3.4 Firms and Production

In this section I expand the Zero-Knowledge methodology in two directions. First I replace the previous section's exogenous fixed daily inflow with endogenous production targets. Second I allow the agent to act in multiple markets at the same time, hiring workers while selling output. Assuming the agent knows whether it is in a monopolist or competitive market the agents reach the correct production levels and prices.

### 3.4.1 Independent PID controls coupled with simple marginal analysis can simulate one-sector competitive and monopolist markets

Agents in this section, Zero-Knowledge firms, produce their own sale goods. In parallel these agents have to hire workers, buy inputs, and price their output. They are price-makers when selling or setting wages and price-takers when buying other inputs. Each price is set by an independent PID controller as in section 3.3.1.

Production is linear with respect to workers hired $L_t$:

$$F(L_t) = L_t \tag{3.8}$$

The firm has to decide how many workers to hire. The simplest way to do so is to raise production as long as:

$$\text{Marginal Benefit} > \text{Marginal Cost} \tag{3.9}$$

More precisely: a firm producing one type of good priced $p_t$ and consuming labor as only input with unit wage $w_t$ maximizes the following profit function:

$$\Pi_t = p_t y_t - w_t y_t \tag{3.10}$$

Where $p_t$ and $w_t$ are themselves function of production $q_t$ so that maximum profits are achieved when:

$$p + q\frac{\partial p}{\partial q} = w + q\frac{\partial w}{\partial q} \tag{3.11}$$

Now define $\mu_p = q\frac{\partial p}{\partial q}$ the price impact of increasing production, that is by how much sale price goes down when production goes up by one unit. Similarly define $\mu_w = q\frac{\partial w}{\partial q}$ as the wage impact of increasing production, that is by how much wages need to increase in order to hire enough workers to produce one more unit of good. So that at any point in time we want to set the production target such that:

$$p_t + \mu^p = w_t + \mu^w \tag{3.12}$$

This is a decision rule for production targets that is based on daily prices $p_t$ and $w_t$ which the PI controllers discover.

As shown in section 3.3.1 it takes some time for PI controllers to find the correct prices. Because marginal benefits and costs are computed with PI prices, production decisions should be taken infrequently to give the controllers time to be correct. In particular as long as $p_t + \mu^p > w_t + \mu^w$ increase production targets by 1 unit of output or lower it by 1 unit of output when $p_t + \mu^p < w_t + \mu^w$

Define $T$ as the decision period: how often, in days, the firm checks whether to change production. It is set to 20 for all simulations. Much like with stickiness $s$, $T$ is also stochastic: there is a fixed chance of $\frac{1}{T}$ each day of choosing a new production target.

The firm must also know what the price impacts $\mu$ are. I will deal with their endogenous discovery in section 3.7. Until then I'll simply assume they are known. To a competitive firm, price impacts are always 0 . To a monopolist, price impacts equal the demand and supply slopes.

Take a firm facing the daily demand function: $q = 102-p$, with daily production function $q = f(L) = L$ and wage curve $w = 14 + L^2$. A firm acting as a monopolist would maximize profits by producing 22 units a day and selling them at \$80.

Figure 3.9 shows a sample run of a Zero-Knowledge monopolist firm. Notice first that the monopolist starts producing after 250 days. This is the time it takes for the PI controller

[2] The demand and supply parameters here and in the following sections are chosen exclusively so that the equilibrium is in natural numbers

setting wages to find the right wage for 1 worker (15$). Notice also that there is no noise once reaching equilibrium.

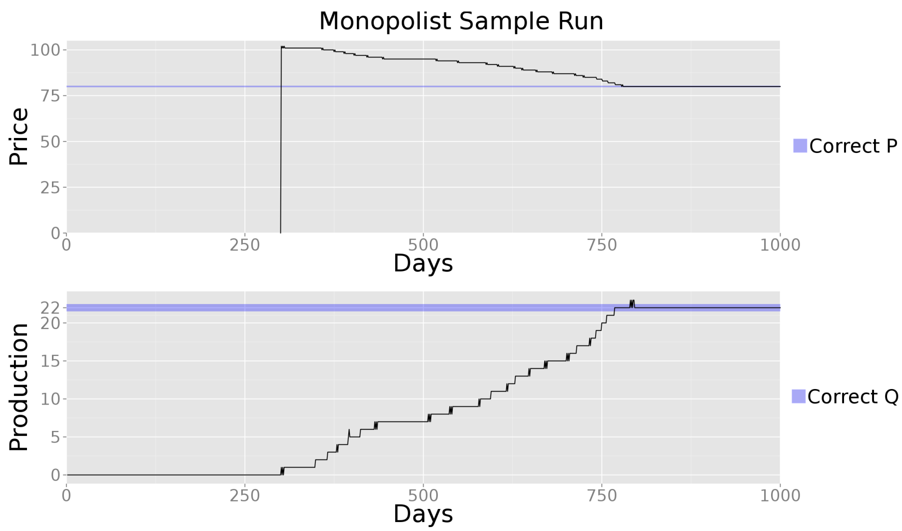


Figure 3.9: A sample run with a monopolist firm. It reaches the correct price and quantity

In perfect competition, equilibrium is a daily total production of 44 units sold at $58. Figure 3.10 shows a sample run with 5 competitors. Multiple agents create noisy results due to coordination failure. While each firm might see an increase in production as profitable, the demand is not enough when all firms increase production at the same time.

Notice that the number of firms competing is not important for the correct result. If I run the model with a single firm with zero price impacts I will get a noiseless perfect competitive solution. Perfect competition is a state of mind. At least until section 3.7.

### 3.4.2 Competitive markets micro-structure is more confusing than its aggregate equilibrium suggests

Competitive scenarios with multiple agents, as in figure 3.10, achieve quasi-equilibrium: prices and quantity hover around the equilibrium levels. However the firms that compose

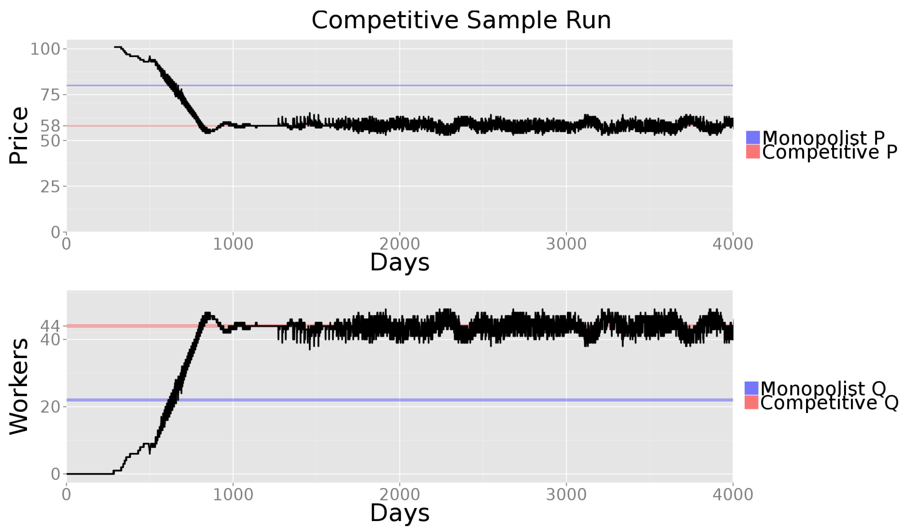


Figure 3.10: A sample run with a 5 competitive firm. There is noise but centered around the correct price and quantity.

this aggregate equilibrium are subject to more chaotic dynamics. Each day there is usually a single price-leading firm that sells to all the customers while the other firms sell nothing. This monopoly lasts only for a day and a different price-leading firm caters to all the customers the day after. This is in spite of the fact that each firm targets only a fraction of the total production.

More specifically each firm uses its PI controller to set their price each day. One of the firm will by necessity price their goods slightly below the others. Because of perfect competition that firm will attract all the customers available. The firm will sell more than its targets burning through its buffer inventory. The day after the firm that sold too much will increase its price (as dictated by its PI controller) while all the other firms will lower theirs since they didn't hit their target. A new firm then takes the pricing lead and the cycle starts over.

At its core, this is an information asymmetry problem. Firms know nothing of other firms while customers know everything about them. So that when a firm drops its price the customers react immediately while competitors can only do so after their consumer base has vanished. It is a dynamic that is enabled by the existence of inventories firms can dip into.

With no inventories the price-leading firms would only be able to supply a limited quantity of goods and multiple daily prices would emerge.

More generally perfect competition is the hardest market structure to model through trial and error pricing. The more market power a firm has, the more informative trial and error pricing is. Small changes in prices for a monopolist generates small changes in quantity demanded, but small changes in prices in a perfect competitive market sometimes generate large swings in demand and sometimes no change at all. But I believe it is important to show that even in the worst case scenario for trial and error Zero-Knowledge traders still get to the equilibrium successfully albeit only in an aggregate sense.

For the rest of the chapter I will continue using multiple agents within the same competitive market in spite of the noise and the fact that they could be replaced by a single agent forced to act competitively, that is with price impacts $\mu = 0$. This is because when I allow agents to learn price impacts in section 3.7 there need to be more than one for them to act competitively. Adding multiple agents only from that section on would generate two kinds of noises at once: competition and learning and that would make comparison between learning and non-learning agents impossible.

## 3.5 Supply Chains

I place the Zero-Knowledge firms of the previous section in a supply chain. Because it takes time for firms downstream to change production targets, firms upstream face a delayed demand similar to section 3.3. Much like that section, the trial and error pricing creates oscillations and fails to reach equilibrium. Much like that section, price-stickiness solves such issues.

### 3.5.1 Zero-knowledge firms in a supply chain create endogenous delays that break the model

Take a supply chain made of two sectors: wood and furniture. There is a final daily demand for furniture which is exogenous and fixed at:

$$q_F = 102 - p_F \tag{3.13}$$

Daily production of one unit of furniture requires one worker and consumes one unit of wood:

$$q_F = \min(L_F, q_W) \tag{3.14}$$

Daily production of one unit of wood requires one worker.

$$q_W = L_W \tag{3.15}$$

Each sector has its own independent linear labor supply:

$$w_W = L_W$$

$$w_F = L_F$$

Helpfully, there are infinite trees waiting to be cut.

In this section further assume that the wood sector is monopolized while the furniture market is competitive. I go through all the market-structure permutations in section 3.6.

Solving for the market equilibrium yields the following:

$$q_F = q_W = 17 \tag{3.16a}$$

$$w_W = w_F = 17 \tag{3.16b}$$

$$p_W = 68 \tag{3.16c}$$

$$p_F = 85 \tag{3.16d}$$

The theoretical demand for wood from the furniture sector is:

$$p_W = 102 - 2q_W \tag{3.17}$$

What I will do next is the following: I will find the best PI parameters that would deal the demand in equation 3.17 if it were a single independent market. I will then show how such parameters do not reach equilibrium in a supply-chain. Finally I will show that adding timidity and sticky prices solve the oscillations and achieve the market equilibrium.

Figure 3.11 shows the parameter sweep for the optimal PI controller of a monopolist facing the undelayed demand function 3.17; for each parameter I show the average simulate $log_1 0$ sum squared errors. The parameters with the lowest error are $a = 0$ and $b = 2$. This makes sense since 2 is the slope of the demand curve.

I have shown that $a = 0, b = 2$ are the optimal PI parameters for the wood producer when facing the fixed demand in equation 3.17. Now I let the same PI controller face the same wood demand but this time it is generated by a downstream market sectors made up of other Zero-Knowledge firms. The effects are shown in figure 3.12. Far from being optimal neither production nor prices ever reach equilibrium. Instead prices oscillate especially wood prices.

The parameters that were optimal when facing an exogenous demand prove too aggressive when the same demand is made up of other Zero-Knowledge firms. The issue is delay.

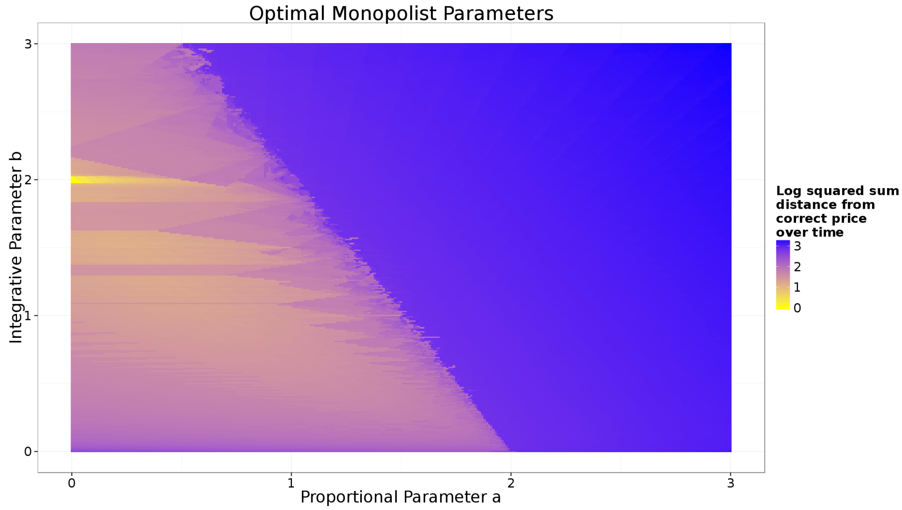


Figure 3.11: The average squared distance from correct prices when a monopolist faces demand $p = 102 - 2q$ and labor supply $w = L$. Each cell represent a pair of parameters used by the monopolist's sales PID controller. The optimal parameter pair is predictably $a = 0, b = 2$ reflecting the underlying demand.

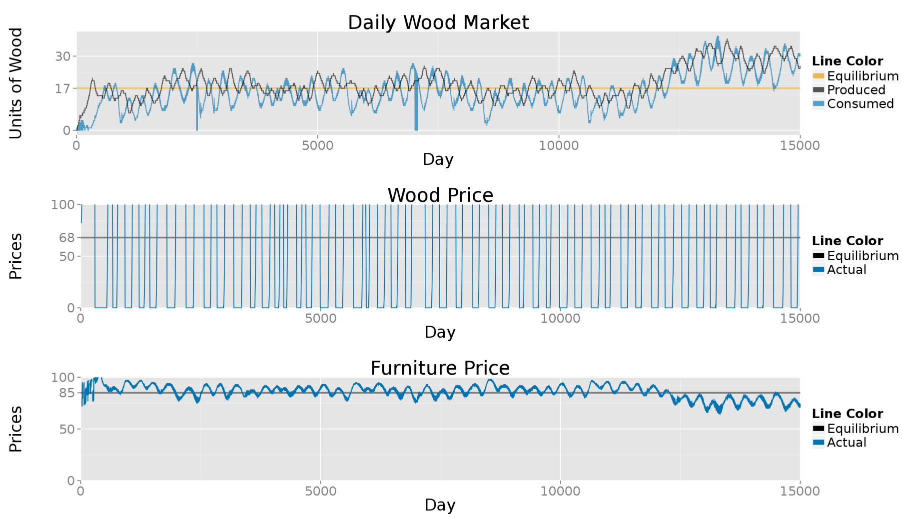


Figure 3.12: A sample run of the supply chain model using the PID parameters that were optimal when the demand was immediately reacting.

It takes time for furniture producers to notice new wood prices and change their production. By the time furniture consumers adjust, the producer has changed sale price again. These delays downstream feed into the upstream trial and error loop. The wood price swings from

0 to over 100 because the wood monopolist's netflow reacts slowly to changes in prices.

To show what causes the price swings table 3.1 shows the decisions made by the PI controller setting the prices of the wood producer from day 1437 to 1445. The root cause is the difference between what is produced and what is consumed. The wood producer is producing 15 units of wood a day but only sells 8 or 9. In fact the wood producer is targeting an even higher production of 16 because the prices are still very high. The PI controller has to find a price such that it can sell all 15 units of good; as a matter of fact that is impossible to do so in a short amount of time because the production decisions downstream respond to prices too slowly . Notice that the PI controller here is just $p_{t+1} = 2\sum_{i=0}^{t} e_i$ so the value of the first column ($p_t$) is just twice the value on the last column ($\sum e_i$).

Table 3.1: The Price and Production Decisions of the Wood Producers in Figure 3.12 from Day 1437 to 1445

| Day | $p_t$ | Target Production | Production | Sales | $e_t$ | $\sum e_i$ |
|---|---|---|---|---|---|---|
| 1437 | 94 | 16 | 15 | 10 | -5 | 47 |
| 1438 | 82 | 16 | 15 | 9 | -6 | 41 |
| 1439 | 70 | 16 | 15 | 9 | -6 | 35 |
| 1440 | 58 | 16 | 15 | 9 | -6 | 29 |
| 1441 | 46 | 16 | 15 | 9 | -6 | 23 |
| 1442 | 34 | 16 | 15 | 9 | -6 | 17 |
| 1443 | 22 | 16 | 16 | 9 | -7 | 10 |
| 1444 | 6 | 16 | 16 | 8 | -8 | 2 |
| 1445 | 0 | 16 | 16 | 8 | 8 | -6 |

The PI controller is aggressively cutting prices but these have very little effect in the short run. Eventually low prices do increase demand downstream and decrease production upstream but by then the prices are at a level too low (0$ in fact) and demand outstrips supply (with inventories being bought instead). This mismatch between the speed of production adjustment and price adjustment is why equilibrium is not achieved. It is clear that prices need to adapt more slowly.

Figure 3.13 shows a second example, this time with timidity $z = 10$. This is the same as lowering the $b$ parameter from 2 to 0.2. The equilibrium is not achieved and the oscillations are still present. I will now show how adding price stickiness improves the dynamics.

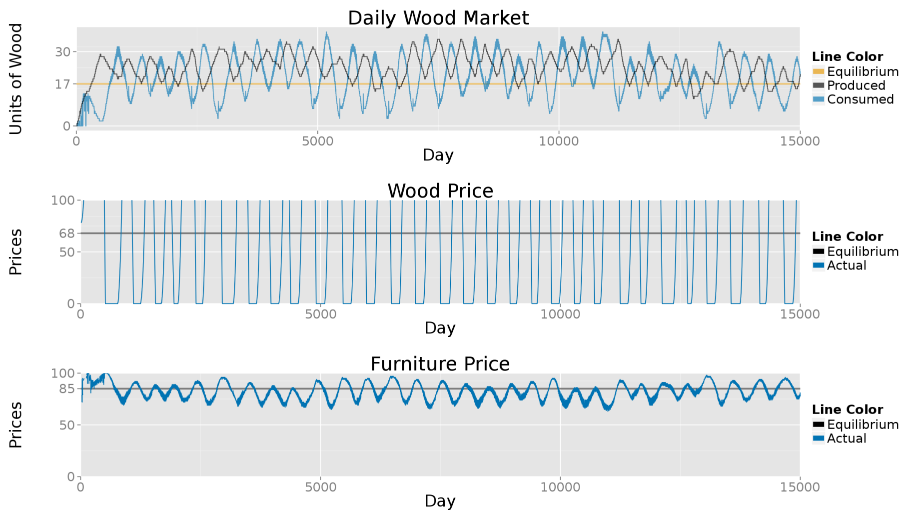


Figure 3.13: A sample run of the supply chain model using the PID parameters that are 10 times more timid than the optimal.

### 3.5.2 Adding price-stickiness to the upstream firm restores equilibrium

Take the supply chain example in figure 3.12 and add both a timidity $z = 10$ and a price stickiness of $s = 50$ days to the PI controller setting wood prices. As figure 3.14 shows, these parameters are enough to fix the supply chain: production and prices in both sectors are the correct ones and remain in equilibrium. This is because now prices changes slowly enough for production to fully adapt to them.

Sticky prices eliminate bullwhip effects the same way they dealt with arbitrary demand delays in section 3.3.3. The difference is that in section 3.3.3 the delays were exogenous as I added them just to show how to deal with them. In this section instead there is a demand delay but it is caused by the interaction between upstream and downstream firms.

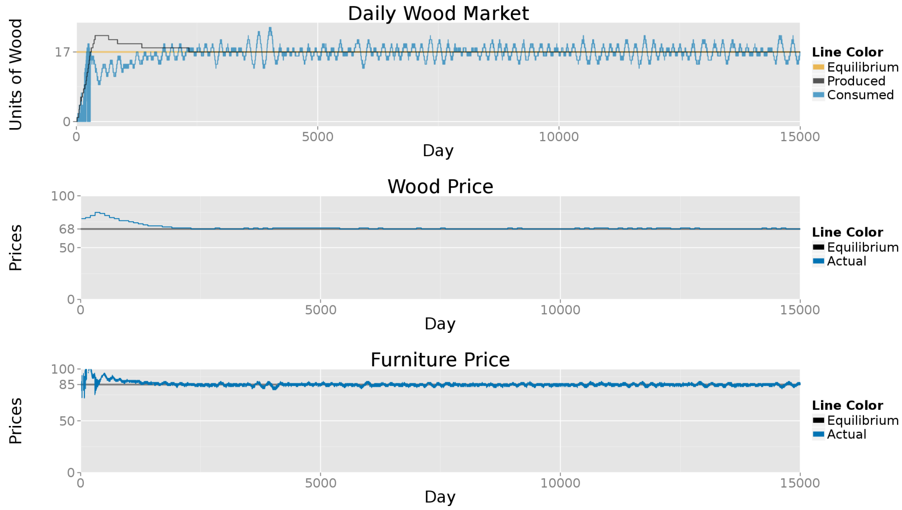


Figure 3.14: A sample run of the supply chain model where the wood producer uses sticky prices.

There are two endogenous sources of delays in the supply chain. The first delay is the time it takes between a firm making the decision to change its production quota and the controllers adapting to it by finding new wages and prices. The second delay source is the decision period $T$ of the furniture producers (how quickly they change production targets given current prices) which delays their response to change in prices of the wood supplier. The larger $T$ downstream the higher the upstream price stickiness $s$ need to be to balance. Figure 3.15 shows the relationship between the stickiness $s$ and $T$.

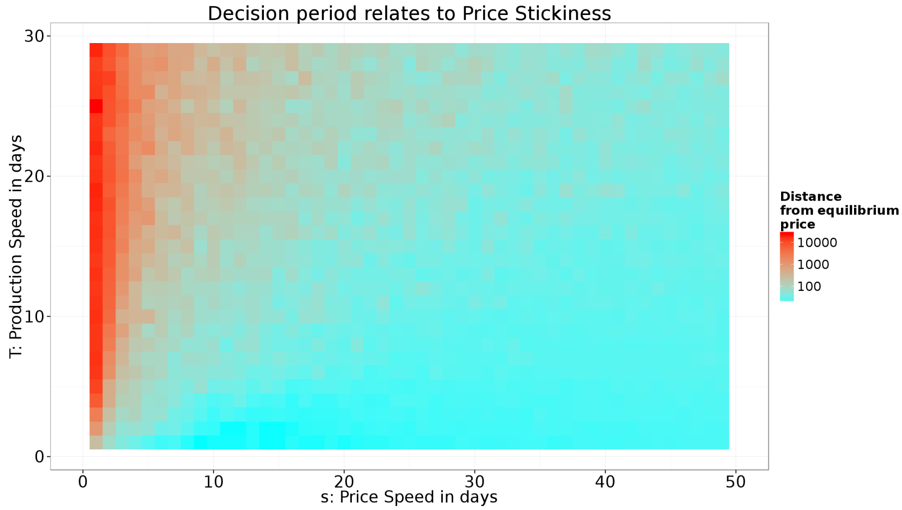


Figure 3.15: The decision period $T$(in days) of firms is an important source of delays in the system; the higher $T$ the higher the price stickiness needs to be in order to balance it. Each tile represents a 5-runs average squared distance from the correct price over the whole run.

## 3.6 Market Structure

In this section I go through the market structure permutations to show how the Zero-Knowledge firms can adapt to it and reach equilibrium. All these simulations use the same price stickiness as in the previous section.

In equations 3.16 I expressed the solution where the wood sector is a monopolist while the furniture sector is competitive. If the wood sector is competitive while the furniture is monopolistic the equilibrium is:

$$q_F = q_W = 17 \tag{3.18a}$$

$$w_W = w_F = 17 \tag{3.18b}$$

$$p_W = 17 \tag{3.18c}$$

$$p_F = 85 \tag{3.18d}$$

If both sectors are competitive the no-profit equilibrium should be:

$$q_F = q_W = 34 \tag{3.19a}$$

$$w_W = w_F = 34 \tag{3.19b}$$

$$p_W = 34 \tag{3.19c}$$

$$p_F = 68 \tag{3.19d}$$

I run 100 simulations for each market structure (competitive means 5 firms in the same sector). Each simulation runs for 15000 market days. Firm have inventory buffer of 100, regardless of market structure. All input producers use sticky prices (50 days each price change), regardless of market structure.

In general the model behaves as predicted by theory. Figure 3.16 shows the distribution of input prices at the end of the simulation; figure 3.17 shows the output prices; figure 3.18 shows the quantity produced.


Figure 3.16: The price of wood (first sector) for 300 simulated runs, 100 for each market structure. The vertical dashed lines represent the theoretical equilibrium. Each datum in the histogram is the average price of the last 500 days of simulation.

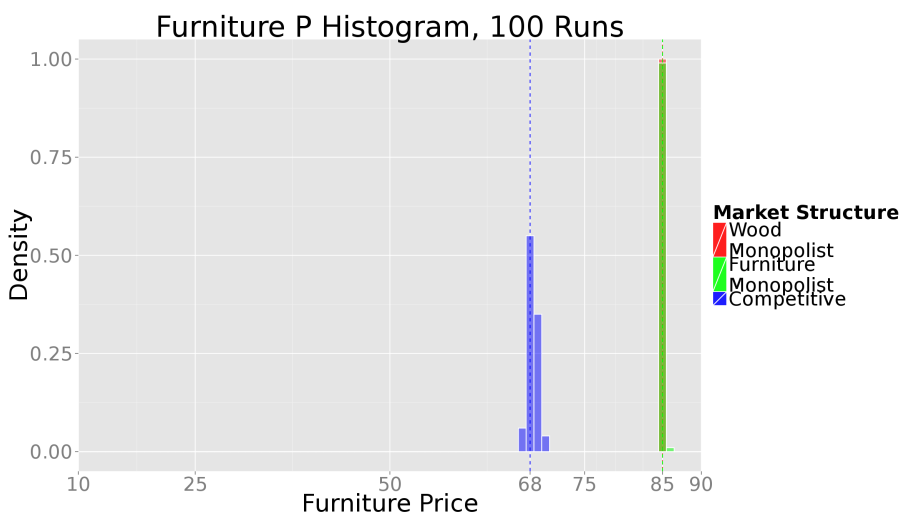


Figure 3.17: The price of furniture (second sector) for 300 simulated runs, 100 for each market structure. The vertical dashed lines represent the theoretical equilibrium. Each datum in the histogram is the average price of the last 500 days of simulation.

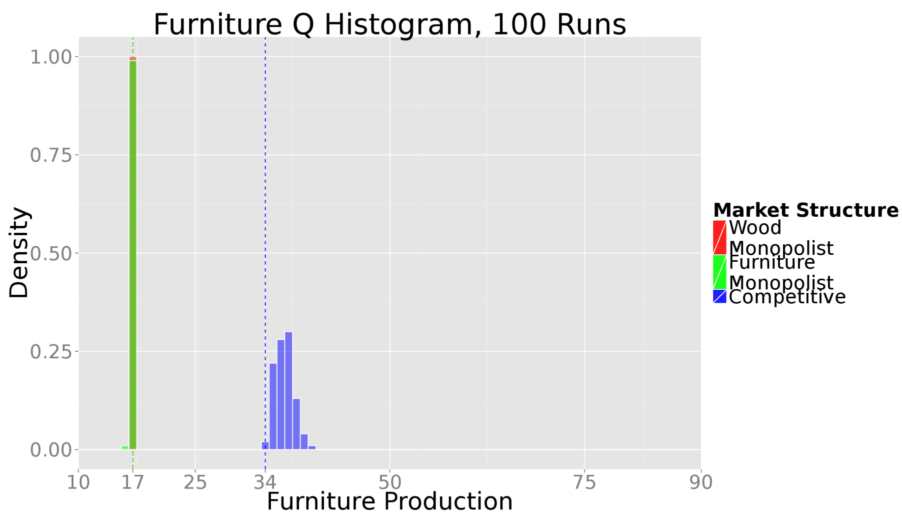


Figure 3.18: The units of furniture produced daily at the end of 300 simulated runs, 100 for each market structure. The vertical dashed lines represent the theoretical equilibrium. Each datum in the histogram is the average daily production over the last 500 days of simulation.

## 3.7 Learning Price Impacts

The results from section 3.4 depends on the firms knowing whether they are in a monopolist or competitive market. In this section I remove this assumption by allowing firms to learn on their own the price impacts they face. Learning produces noise compared to the results in the previous section, but the equilibrium quantities and prices are comparable. The only exogenous constraint remaining in this section is that upstream firms use sticky prices.

### 3.7.1 Regressing workers on price works well in a one-sector economy

A Zero-Knowledge firm should be able to learn its own market power. In previous sections the price impacts were given, now firms discover them.

The firm takes the price generated by its PID controls and regresses it against number of workers. The regression identifies how much increasing production changes prices. Zero-Knowledge firms use two regressions side by side. First, they fit one-step error correction regression model [Banerjee et al., 1993]:

$$\Delta p_t = \beta_0 + \beta_1 \Delta L_t + \beta_2 p_{t-1} + \beta_3 L_{t-1} + \epsilon \tag{3.20}$$

Where $p$ is price and $L$ are workers hired. The firm identifies the long run relationship between the two variables and use it as approximate price impact:

$$\mu^p = -\frac{\beta_3}{\beta_2} \tag{3.21}$$

The second regression is the linear model:

$$p_t = \gamma_0 + \gamma_1 L + \epsilon \tag{3.22}$$

Where the price impact discovered is:

$$\mu^p = \gamma_1 \tag{3.23}$$

Each day the Zero-Knowledge firm selects the regression that better predicts today's price. If a firm trades in multiple markets (for example selling furniture, buying wood and hiring workers) then it has multiple regression pairs, each focusing on predicting one price (output price, input price, wages). For the input markets where the firm is a price-taker the paid price is used in lieu of the PID one.

Because the PID controls generate one observation each day in each market it makes sense to implement the regressions by a Recursive Least Squares filter[Welch and Bishop, 1995]. Take as example Equation 3.20. It has four parameters: $\vec{\beta} = (\beta_0, \beta_1, \beta_2, \beta_3)$. Each day the firm observes the price offered, the labor hired and their lags $y_t = \Delta p_t, \vec{x}_t = (1, \Delta L_t, p_{t-1}, L_{t-1})$. The current estimation of $\vec{\beta}$ is $\hat{\vec{\beta}}_{t-1} = \left(\hat{\beta}_{0t-1}, \hat{\beta}_{1t-1}, \hat{\beta}_{2t-1}, \hat{\beta}_{3t-1}\right)$. Update it with the new observation in four steps:

$$\vec{k} = \mathbf{P}_{t-1}\vec{x}^T \left(\vec{x}P_{t-1}\vec{x}^T + 1\right)^{-1} \quad \text{Constructing the Kalman gain} \tag{3.24a}$$

$$\epsilon_t = y_t - x\hat{\vec{\beta}}_{t-1} \quad \text{Finding the prediction error} \tag{3.24b}$$

$$\hat{\vec{\beta}}_t = \hat{\vec{\beta}}_{t-1} + \vec{k}\epsilon_t \quad \text{Updating predictor given error} \tag{3.24c}$$

$$\mathbf{P}_t = (I - \vec{k}\vec{x}_t)\mathbf{P}_{t-1} \quad \text{Updating covariance matrix} \tag{3.24d}$$

Where $\mathbf{P}_t$ is the $4 \times 4$ covariance matrix. Functionally $\mathbf{P}_0$ is a Bayesian prior which I set at $10^4 I$ for all simulations.

Figure 3.19 and figure 3.20 show the results of running 100 simulations with a learning monopolist and 100 simulations with 5 learning competitors. Firms learn their market power correctly, firms quickly learn whether they are monopolists or not and if they are they learn the correct slopes.

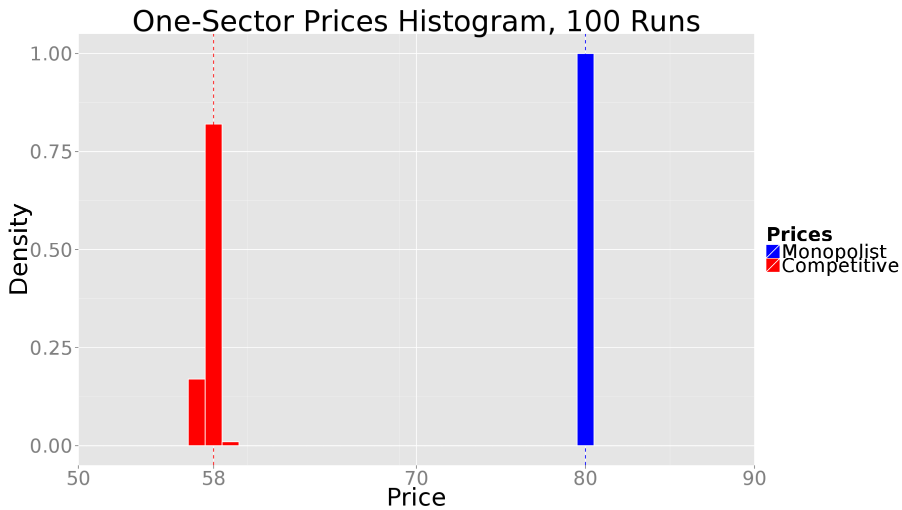


Figure 3.19: The histogram of prices from running 100 monopolist and 100 competitive (5 firms) scenarios. All firms need to learn the price and wages impact. All firms target inventory (100 units of output). Each observation in the histogram is the average of the last 500 days' prices of that particular simulation.

### 3.7.2 Learning in a supply-chain is harder and less effective

Learning is far more problematic in a supply-chain. Firstly, if there is a delay $\delta$ between setting a price $p_t$ and it affecting quantity traded, the Zero-Knowledge firm should regress $p_{t-\delta}$ over $L_t$. But this is impossible as the delay is unknown. Secondly, because of stickiness, the firm often sets prices $p_t$ that are not the market clearing ones. Because learning works by regressing paired $p_t$ and $L_t$, the results are often useless. Figures 3.21, 3.22, 3.23 show the results of 100 simulations for each market structure. All are using buffer inventory and sticky prices. The results are far more dispersed, usually because one of the recursive least squares filters failed to learn the correct slope.

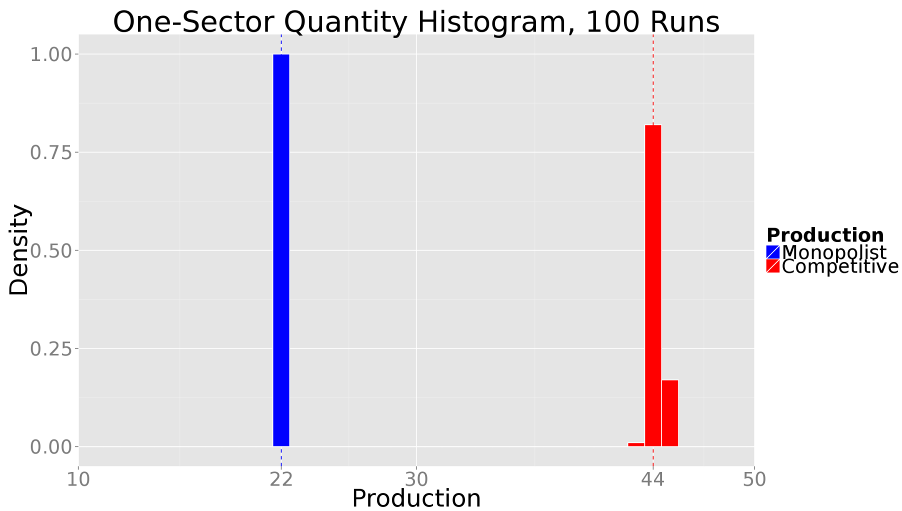


Figure 3.20: The histogram of simulated production obtained by running 100 monopolist and 100 competitive (5 firms) scenarios. All firms need to learn the price and wages impact. All firms have a buffer inventory (100 units of output). Each observation in the histogram is the average of the last 500 days' production of that particular simulation


Figure 3.21: The price of wood (first sector) for 300 simulated runs, 100 for each market structure. The dashed vertical lines represent the theoretical equilibrium. Each datum in the histogram is the average price of the last 500 days of simulation.

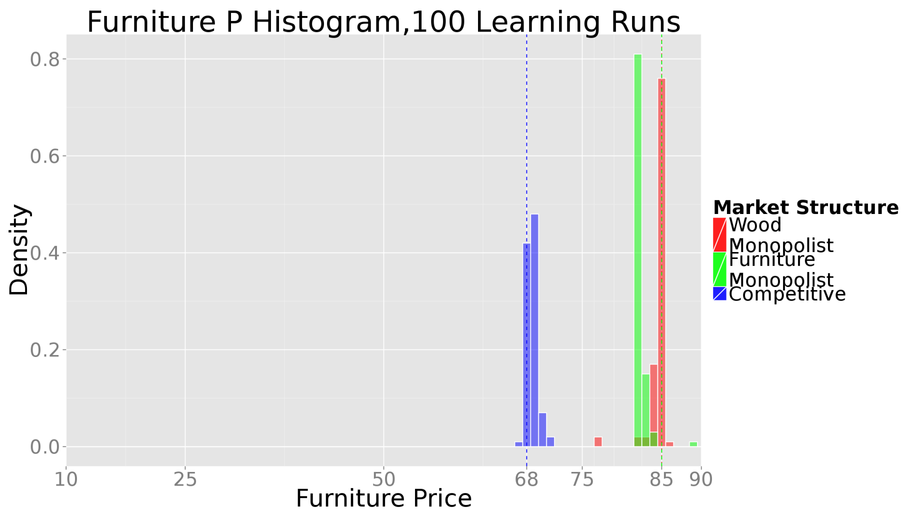


Figure 3.22: The price of furniture (second sector) for 300 simulated runs, 100 for each market structure. The dashed vertical lines represent the theoretical equilibrium. Each datum in the histogram is the average price of the last 500 days of simulation.

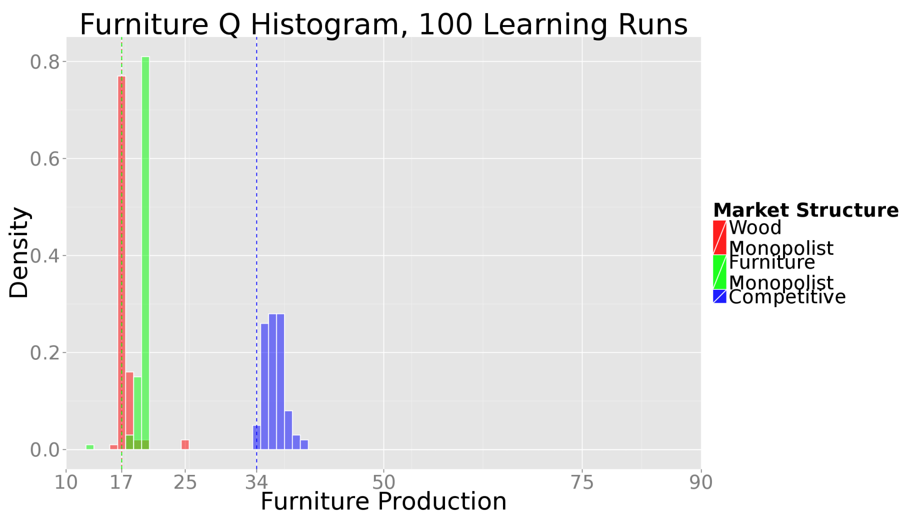


Figure 3.23: The units of furniture produced daily at the end of 300 simulated runs, 100 for each market structure. The dashed vertical lines represent the theoretical equilibrium. Each datum in the histogram is the average daily production over the last 500 days of simulation.

### 3.7.3 Circular causality and passivity are the main learning weaknesses

The error-correcting model assumes that labor determines prices. That is true as the PID control reacts to increased production by lowering prices. But production also responds to prices as firms fire workers when sale prices fall. Circular causality is exacerbated by the regression itself since the slope found is part of the profit maximization function linking the two variables.

The root cause of this confusion is that agents are passive learners when it comes to price impacts. Zero-Knowledge firms observe long time series of prices and production and try to make sense of them. What these firms never do is willfully experiment with wrong prices. Firms never try to double prices to see the effect on demand or increase production beyond the optimal level to test their estimated labor supply slope. Agents are, in meta-heuristics term, greedy. They always exploit current knowledge and never explore. This I believe is in line with how learning is usually modeled in economics [Evans and Honkapohja, 2009] but it is probably an assumption that should be dropped in future work.

## 3.8 Learning stickiness

In this section I remove a further assumption on firms behavior. I introduced stickiness $s$ in section 3.3.3 but I always set it exogenously: firms would either have stickiness or not. In this section $s$ emerges endogenously by providing the firm with a way of setting it on its own.

The firm needs to change the stickiness parameter of its PID controller while it is in use. This is the domain of adaptive control [Landau et al., 2011]. In this case too Zero-Knowledge firms act by trial and error.

The first step involves defining a performance metric to judge controllers and their parameters. Here I use the integral time absolute error (ITAE) performance index [Shinners, 1998]:

$$\sum_{i=t-M}^{t} i\,|\hat{e}_i| \tag{3.25}$$

The lower the ITAE the more precise the controller. $M$ is the time horizon and the error $e_t$ is the PID error as defined in equation 3.1. The performance index simply states that

PIDs are better parametrized if the system is on target and being on target in the long run matters more than in the short run.

In this chapter I only focus on the stickiness parameter: how many days pass between each adjustment by the PID controller. I modify this parameter by simple hill-climbing [Luke, 2009]. Zero-Knowledge firms set a stickiness and test it for $M = 100$ days. If it has better performance than the previous stickiness then we keep it otherwise we revert back to the previous one. This process loops forever.

A sample run where the firm tunes its stickiness is in figure 3.24. The firm starts tuning its stickiness after 1000 days, it changes stickiness in steps of 5. In this run there is a first abortive attempt at sticky prices at around day 2000. The experiments fail because prices get sticky while out of equilibrium which cause poor performance. After reaching equilibrium stickiness stops mattering which results in the stickiness parameter bouncing between 20 and 25 days.

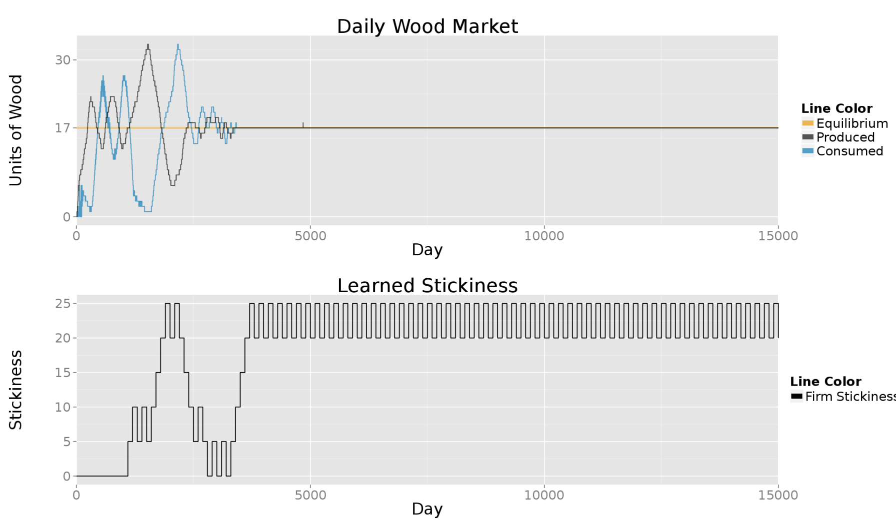


Figure 3.24: A sample run where the firm starts selling with a non-sticky PI controller $b = .2$. We start tuning after 1000 days. Time horizon $M$ is 100 days.

The tuning process could be improved by making learning forward-looking. This is generally called indirect adaptive control. The idea is to fit process data to a statistical model and then estimate performance by treating the fitted model as the real one. The issue is usually to find the right statistical model. See chapter 12 of [Landau et al., 2011] for a primer on the field. Hill-climbing bypasses that by experimenting directly over the real system.

## 3.9 Fitting Zero-Knowledge Traders to Data

In this section I show how to fit PI controllers to data-sets and how to use my model empirically. While PI controllers aren't used directly to price goods in the economy we might want to estimate what are the best PI parameters that simulate empirical pricing. We need two time series, the price set by the controller and the error the controller reacted to. The standard technique to fit a PI controller to data is to turn it into a velocity form [Åström and Hägglund, 1995] where the PI formula becomes:

$$\Delta u_t = \alpha e_t + \beta e_{t-1} + \gamma e_{t-2} \tag{3.26}$$

Which can then be fitted through ordinary least squares (OLS).

I am not pursuing that strategy here because it is brittle : OLS fails once I add windup stop or any other modification to the general PI controller. I fit the PI parameters by simulation instead. I start with a random vector of PI parameters $a, b$, generate the simulated policy time series $\tilde{p_t}$ given the error time series $e_t$. I then compare the simulated time series with the real policy time series $p_t$ and record the absolute simulated error:

$$\epsilon = \sum_{i=1}^{T} |\tilde{p_i} - p_i| \tag{3.27}$$

This gives us a mapping $(a, b) \rightarrow \epsilon_{a,b}$ which we can plug in any optimizing routine to find what parameters $(a, b)$ minimize the absolute simulated error $\epsilon$

A simple example would be fitting the European Central Bank(ECB) deposit rate as if set by a PI controller trying to keep unemployment rate at 8%. The error time series $e_t$ is then difference between unemployment rate and 8% while the policy time series $p_t$ is the deposit rate. The result is shown in figure 3.25. The PI controller has a strong proportional component and a weak integral. This matches the Taylor rule approach which is fundamentally P only.

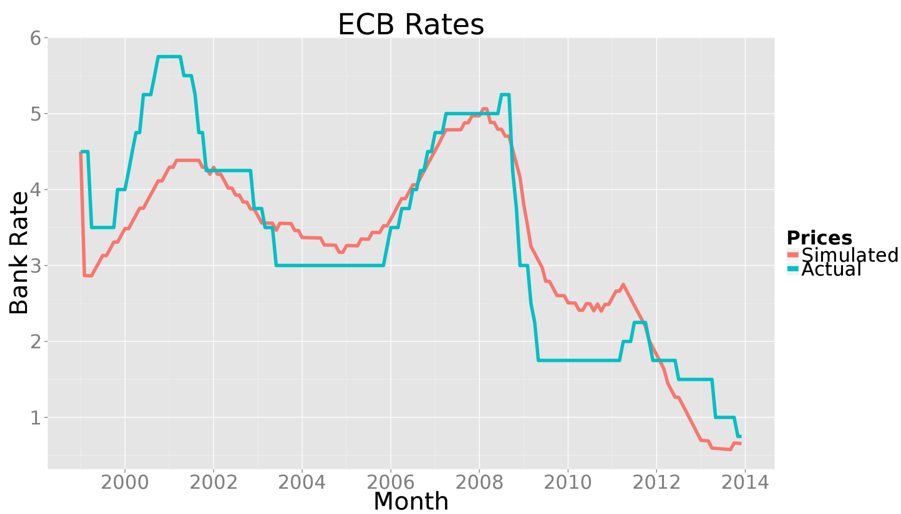


Figure 3.25: A comparison between European Central Bank(ECB) rates and the rate simulated by a PI controller targeting unemployment at 8%. The parameters are $a = 0.94$ and $b = 0.0005$

In reality central banks target both unemployment and inflation at the same time. [Hawkins et al., 2014] fits a PID controller targeting both (using potential output gap rather than unemployment) to the US central bank and compares the fit favorably with traditional Taylor rules estimations. In general, the main advantage of studying central banks is that there is a clear idea of what the target and the error are.

For the fit to be informative the data needs to be of high quality. If the time density is too coarse or we use aggregate market data the fit will be meaningless. As an example, imagine a PI controller setting rents in the United States. Take as error time series $e_t$ the monthly rental vacancy rate and as price series $p_t$ the real rent rate (more precisely the urban rent Consumer Price Index(CPI) divided by the general CPI). I show the fit in figure 3.26

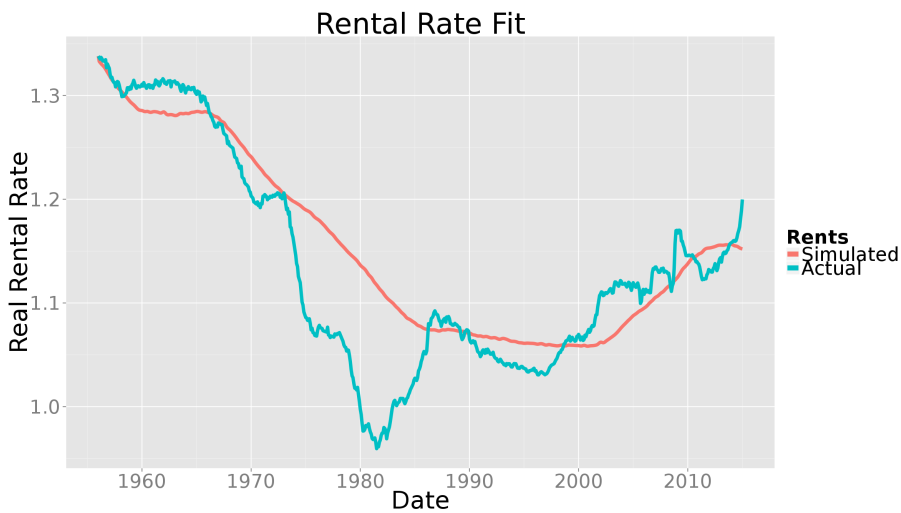


Figure 3.26: The real urban rental rate in the US and the closest PI output when targeting 8% vacancy rate. Notice that the PI parameters are $a = -0.0013$ and $b = -0.00038$, that is rent goes up when vacancy rate goes up

While the fit might look acceptable the PI parameters are negative. This means that the best PI fit has rent go up when vacancy are high. That is clearly meaningless. I replicated the circular causality problem of section 3.7.3 where over large periods of time (in this case months) two processes occur: price declines when demand is low but eventually production also declines because of low prices. For the housing market while it is possible that rents decline while vacancies are high it is also true there will be less houses on the market while rents are low. The overall correlation between monthly rent and monthly vacancies could then be of either sign.

What is needed is firm-level, high frequency data. A dataset that is firm-level but is not high frequency nor has enough observation is the free supermarket data from [Aguirregabiria,

Table 3.2: 10 best PI Fits for the Wholesale Price

| dart | aParam | bParam | margin |
|---|---|---|---|
| SIRO MARIA 400 GRS. | -0.000708 | 0.005561 | 34.68966 |
| AGUA SOLARES 250 C.C PAK 6 | 0.000865 | 0.000000 | 74.72056 |
| PERA HERO ALMIBAR 450 GRS. | -0.011320 | 0.022337 | 90.79018 |
| VINO OLARRA OTOAL 87 ROSADO | -0.002209 | 0.014678 | 93.63590 |
| BAYETA VILEDA SUELOS | -0.001862 | 0.002559 | 96.65143 |
| PAN LU DE PUEBLO 24 REBANAD. | -0.010047 | 0.010692 | 115.45256 |
| LENTEJA ASTURIANA PARDINA KG | -0.001683 | 0.003155 | 120.79976 |
| PIMI. VELA PICO LTA. 175 GRS | -0.000475 | 0.002262 | 121.79440 |
| FONTANEDA EL COLEGIAL 800 GR | -0.023865 | 0.039007 | 124.33759 |
| BOLSA BASURA 25 UND. RF.1007 | 0.000013 | 0.000000 | 126.79908 |

1999]. There are 529 goods, each with 29 observations, one per month. The problem with non central banks data is to figure out what the PI target and the errors are. This is particularly complicated in this data-set because of the presence of inventories, returns and the lack of information on the manufacturers themselves. For this data-set, I use as PI policy $p_t$ the wholesale price, and as PI error the difference between orders placed by retailers to the wholesaler and the orders placed by the wholesaler to the manufacturers. This is sub-optimal because it ignores customer returns and inventory targets, but it is a simple proxy for what must be the sales targets.

Table 3.2 shows the 10 best PI fits. In some cases, the P parameter is negative, but it is always the case $a$ is smaller than the $b$ parameter so that the PI controller never operates in "reverse" . The median simulated error $\epsilon$ is 416.48 Pt, so wrong by about 14 Pt a month. An example of the successful fit ($\epsilon \approx 200$) is shown in figure 3.27.

The main weakness of my estimations is that the simulated estimated error $\epsilon$ is computed over training data. A better approach, especially when comparing different model fits, would be to compute $\epsilon$ by cross-validation or training data. This was not feasible for these examples because the testing data of the European Central Bank would have been the rates during the crisis which are set more aggressively than the training data would suggest while the wholesales data is too small to afford it being cut into training and testing.

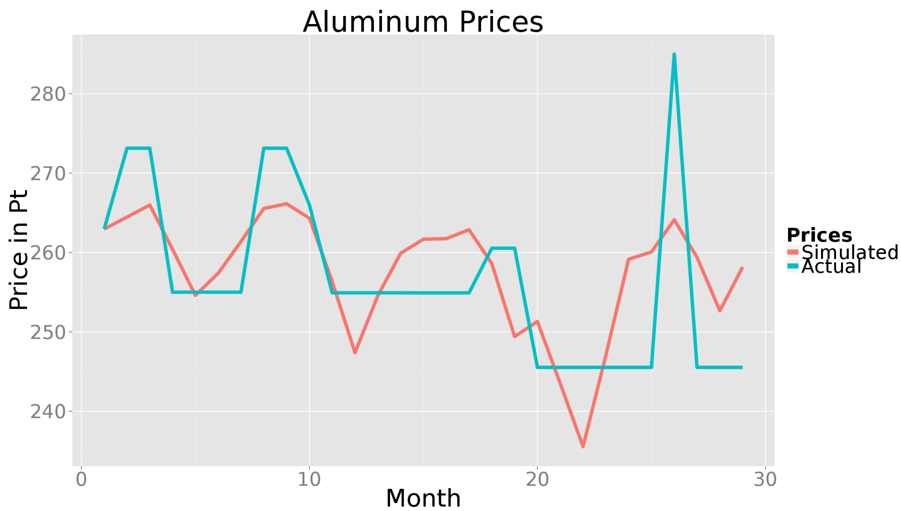


Figure 3.27: The comparison between PI controller and actual wholesale prices of 30 meters of Reynolds aluminum foil

## 3.10 Conclusion

In this chapter I showed how trial and error pricing creates bullwhip effects and how sticky prices can fix them. This allowed agents that are too simple to centralize information to coordinate over market prices. I also showed how the result is robust to market structure and knowledge assumptions about price impacts $\mu$ and price stickiness $s$. I believe this chapter represents an example of how focusing on interactions and agent-based models can provide new answers and hypotheses to old questions. It is a methodology that allows for expressing and examining time and trading rules' minutiae easily.

There are two paths I can take with this model. The first is improving its overall realism, probably by adding more feed-forward elements. Agents here are purposefully simple as a way to show how little is required for markets to coordinate. A better agent would use more data: here agents weren't allowed to even look at competitors' prices. If they use more data artifacts such as those described in section 3.4.2 would disappear as firms would be on the same level as their consumers. A better agent would also be able to auto-tune its PI parameters during the simulation. I did tune stickiness $s$ in section 3.4.2 but the process could be generalized to $a$ and $b$ as well.

The second path I can take with this model is to use it as it is with different market structures. I have shown that the Zero-Knowledge firm works well in a monopolist environment because trial and error is at its most informative when the agent is alone. Viceversa I have shown that in a competitive environment trial and error is at its least informative because of the noise generated by the competition and the ease of the customer base to switch providers. I think the Zero-Knowledge agent would thrive in a point between these two extremes. I believe that monopolistic competition, a market where each agent has only partial market power and information is too dispersed for game theory to apply, is the obvious next step.

# Chapter 4: A Cybernetic Model of Macroeconomic Disequilibrium

## 4.1 Introduction

I extend here the Zero-Knowledge traders methodology to macroeconomics. By modeling the economy as a process that agents try to control, I can study the effect that higher flexibility and adjustment speed have on the economy as a whole. More flexibility is not always beneficial as it can aggravate the disequilibrium and the social costs associated with a recession.

I use my model to study reforming the labor market during a recession. In Europe labor market reforms were touted both before [Siebert, 1997] and after [Bertola, 2014] the economic crisis as a way to boost economic growth. The point of reforming the labor market is to increase labor mobility and therefore productivity. This chapter challenges the notion that increasing either mobility or productivity is beneficial during a recession. I show how increasing mobility actually deepens the recession and increases production undershooting when it is increased at the start of a drop in demand.

I base my model on Leijonhufvud's "Keynes and the Keynesians" where the difference between Keynesian and Marshallian economics is how agents adapt to disequilibrium [Leijonhufvud, 1972]. To Leijonhufvud, Marshallian agents react to mismatches in demand by first adjusting prices and only later changing production. Keynesian agents instead react first by adjusting quantities and only later prices. I implement this idea and show how these differences have no effect in microeconomics but do so in macroeconomics.

This is not the first attempt to model the microfoundations of "Keynes and the Keynesians". Leijonhufvud himself cited the search models in "Information Costs, Pricing and Unemployment" [Alchian, 1969] and Clower's false trades [Clower, 1965] as a way to achieve his vision. This chapter follows in the false trades tradition of allowing exchanges at wrong prices but I provide a simple trial and error agent that corrects itself over time. It is the dynamics generated by this trial and error pricing that differentiate Keynesian disequilibrium from the Marshallian one.

## 4.2 Literature Review

I classify agents in macroeconomics on a spectrum that goes from complete feedback to complete feed-forward. Feedback agents are reactive, they manipulate control variables by inferring over time what their effect is on the other model variables. Feed-forward agents know perfectly the model and set all the control variables at the beginning of the simulation after having solved for the optimal path.

Modern economics focuses mostly on feed-forward agents. The Ramsey-Cass-Koopmans model [Ramsey, 1928][Cass, 1965][Koopmans, 1963] is an example of a pure feed-forward agent. In this model the agent is omniscient and chooses the saving rate for any instant of its infinite life by optimizing utility given the lifetime budget constraint. This omniscience is a fundamental driver of the model as it explains, for example, why permanent taxes do not crowd out investments while temporary taxes do [Romer, 2011].

In Prescott's Real Business Cycle growth model [Prescott, 1986] the agent is an imperfect feed-forward control. In this model there are auto-regressive random technological shocks that cannot be predicted ahead of time. The agent then has a large feed-forward element that finds the optimal distribution of control strategies to implement and a small feedback process to choose the control strategies from this distribution as the random shocks occur.

Feed-forwarding optimization with feedback adaptation to uncertainty remains the standard macroeconomic approach to this day. The more uncertainty a model has, the larger the feedback element of the agent is but the focus is always on feed-forwarding. Learning models as in [Evans and Honkapohja, 2009] are emblematic of this: agents don't have model knowledge but rather than managing this uncertainty they employ feedback econometrics to learn the model just so that they can then use the usual feed-forward control strategies on what they learned.

The only agent in economics that is still pure feedback is the central bank: the Taylor rule [Taylor, 1993] is a simple feedback and adaptive rule to set interest rates. It is in fact a simplified PI controller [Hawkins et al., 2014]. Real economists allow simulated economists some slack in assuming not only that they face uncertainty but that they are never able to reduce it by learning the full model.

The modern focus on feed-forwarding is surely a reaction to the feedback oriented methodology that preceded it. The Keynesian IS-LM [Modigliani, 1944] were almost pure feedback models. Consumption would be a fixed proportion of income, workers would be reacting the same fixed way to changes in prices and therefore could be fooled over and over again into generating a Philips curve [Heijdra and Van der Ploeg, 2002]. Explicitly cybernetic models shared this top-down fixed feedback approach [Tustin, 1957] [Phillips, 2000] [Cochrane and Graham, 1976] [Lange, 1970]. Leijonhufvud called it the "Keynesian Revolution that didn't come off" [Aoki and Leijonhufvud, 1976] but the approach survives in the field of system dynamics [Sterman and Sterman, 2000].

I use pure feedback agents but differently from how they were used. Rather than assuming fixed mechanical connections I have agents use feedback to adapt over time to shocks and disequilibrium.

## 4.3 Microeconomics

### 4.3.1 Marshallian Agents

This is a brief summary of the Zero-Knowledge trader methodology. The unit of time is a "market day" as in Hicks [Leijonhufvud, 1984] . Take a simple market for one type of good. Each market day, a firm produces $y_t^s$ units of good, consumers buy $y_t^d$ units at price $p_t$. The Marshallian firm is a price-maker that takes production as given and changes $p_t$ every day in order to make production equal demand that is:

$$y^s = y^d \tag{4.1}$$

The agent has no knowledge of market demand and how his own price $p_t$ affects it. It knows only that higher prices imply lower demand. It proceeds then by trial and error: it sets a price $p_t$ and computes the error $e_t = y^s - y^d$ and uses it to set $p_{t+1}$. I simulate the trial and error process by a PI controller:

$$p_{t+1} = \alpha e_t + \beta \sum_{i=0}^{t} e_i \tag{4.2}$$

By manipulating $p_t$ the Marshallian agent changes demand $y^d$ until it equals supply $y^s$. Within each market day the Marshallian agent treats its own good supply as given but over time it can use the price it discovers to guide production. At the end of each day there is a small fixed probability (in this simulation $\frac{1}{20}$) to change supply $y^s$ by adjusting labor hired. The decision is simple marginal optimization: increase production while Marginal Benefit > Marginal Costs and viceversa. The firm again adjusts by trial and error and with a separate PI controller whose error is $e_t = \frac{\text{Marginal Benefit}}{\text{Marginal Cost}} - 1$.

In the previous chapters I went through more complicated scenarios with multiple firms and markets, monopoly power and learning. But this minimal setup is enough for this chapter. Here a firm has only two degrees of freedom, price set $p$ and labor hired $L$. Each set by an independent PI controller.

### 4.3.2 Keynesian Agents

Keynesian firms function exactly as Marshallian ones except that they reverse the speed and the error of the two PI controllers. A Keynesian firm changes $L$ every day trying to make $y^s = y^d$ and changes $p$ with the a small fixed probability trying to make Marginal Benefit =

Marginal Cost.

Functionally the Keynesian firm tries to match supply and demand within a market day by changing $y^s$ directly rather than changing $p$ and therefore $y^d$ as the Marshallian one.

### 4.3.3 In a partial-equilibrium scenario Keynesian and Marshallian agents perform equally

There is one firm in the economy. It faces the exogenous daily linear demand

$$y_t^d = 100 - p_t \tag{4.3}$$

One person hired produces one unit of good a day:

$$y_t^s = L_t \tag{4.4}$$

There is infinite labor supply at $w = \$50$. The perfect competitive solution is $L = 50, y = 50$

I run 1000 simulations for a Marshallian and a Keynesian firm each setting their PI parameters $\alpha, \beta \in [0.05, 0.2]$ and random initial price and labor $p_0, L_0 \in [1, 100]$. Firms in all simulations always find the equilibrium as shown in figure 4.1.

Define equilibrium day as the simulation day when the firm produces within 0.5 units of the equilibrium production. Figure 4.2 compares the equilibrium day distribution of Keynesian and Marshallian firms. A two-sided Kolmogorov-Smirnoff test fails to reject that the two samples come from the same distribution (p-value is 0.263) In a partial equilibrium microeconomic scenario Keynesian and Marshallian firms performs equally well and at equal speed.

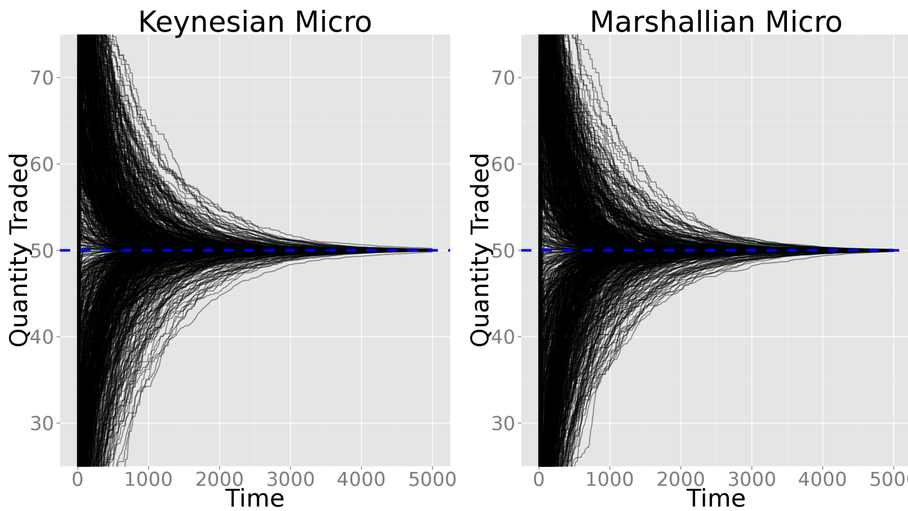


Figure 4.1: The path of $y$ traded for a 1000 Keynesian and Marshallian simulations. Regardless of initial conditions and PI parameters, the runs all reach equilibrium.

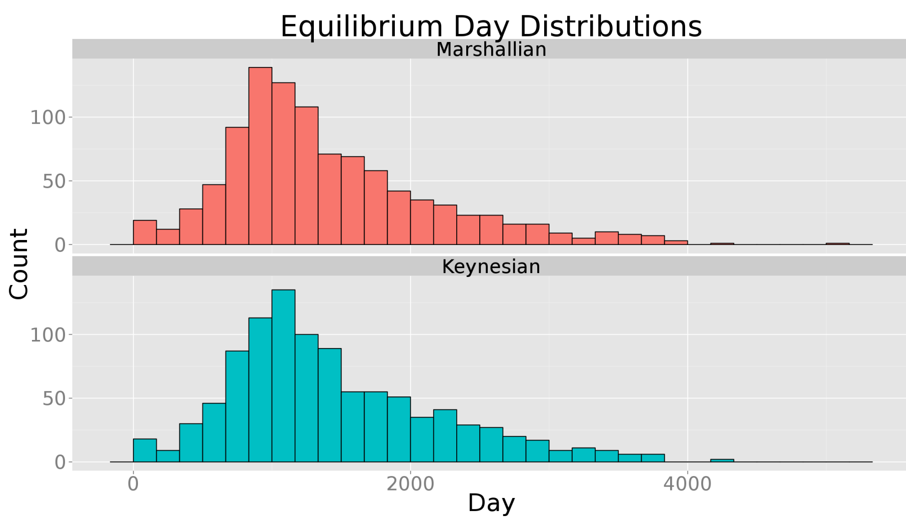


Figure 4.2: The empirical distribution of equilibrium time for the Keynesian and Marshallian microeconomic simulations. There is no difference between them.

## 4.4 Macroeconomics

### 4.4.1 Both Marshallian and Keynesian firms are able to reach equilibrium in a simple macro model

Here I present a minimal macroeconomic model and show how Marshallian and Keynesian dynamics diverge. The main reason they do is that Keynesian adjustment has side effects. Keynesian firms manipulate labor directly; in microeconomics that meant changing only the good supply but in macroeconomics the good demand is equal to labor income so that hiring and firing workers moves the demand as well. Marshallian firms instead manipulate prices moving the demand without affecting supply.

There is a single firm in the world. It is programmed to act as in perfect competition and targets Marginal Benefits=Marginal Costs. It produces a single good with daily production function:

$$Y^S = a\sqrt{L} - b \tag{4.5}$$

It has access to an infinite supply of labor $L$ at $w = 1$.

The demand for the output is equal to the real wages paid:

$$Y^D = \frac{L}{p} \tag{4.6}$$

Unsold output spoils, unused labor income is never saved. This market has the following unique equilibrium:

$$L = \frac{4b^2}{a^2} \tag{4.7}$$

$$p = \frac{2\sqrt{L}}{a} \tag{4.8}$$

$$y = -b \tag{4.9}$$

When $a = 0.5$ and $b = 1$ the solution is:

$$L = 16 \quad (4.10)$$

$$p = 16 \quad (4.11)$$

$$y = 1 \quad (4.12)$$

The computer simulation proceeds just like the previous microeconomic section except that demand here is endogenous and equal to wages paid Two sample runs are shown in the figure 4.3.

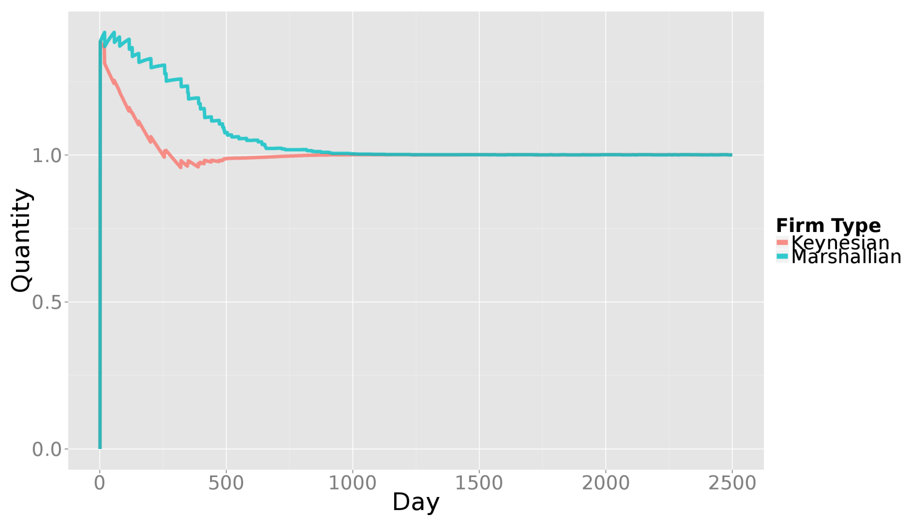


Figure 4.3: Two sample runs of the economy $Y$ with a Keynesian and a Marshallian firm.

I run 100 simulations each for Keynesian and Marshallian firms, where the $p$ and $i$ parameters of the controllers are random $\sim U[0.05, 0.2]$. Both Keynesian and Marshallian firms are always able to achieve equilibrium.

### 4.4.2 Keynesian and Marshallian firms generate very different dynamics when reacting to a demand shock

As initial conditions matter, rather than studying the dynamics toward equilibrium ab ovo, I first let the model reach equilibrium then subject it to a demand shock and see how the firms differ in adapting to it. I run the same simulation as before, but after 10,000 days the output demand is shocked by $s$\$:

$$Y = \frac{L}{p} - s \tag{4.13}$$

When $s = 0.2$ the new equilibrium becomes:

$$L = 10.24 \tag{4.14}$$

$$p = 12.8 \tag{4.15}$$

$$Y = 0.6 \tag{4.16}$$

Figure 4.4 shows the difference in adjustment dynamics between Keynesian and Marshallian firms. Marshallian firms react to the sudden drop in demand by lowering price so that quantity traded briefly recovers after the shock. Eventually though the lower prices feed into the profit maximization PI which cuts production towards the new equilibrium. Keynesian firms instead react to the drop in demand by immediately firing workers. While firing workers lowers supply it also decreases demand because unemployed workers don't consume. The Keynesian firm can't change supply without changing demand as well.

Keynesian firms reach the new equilibrium faster. Define equilibrium time as after how many days the output settles within 0.05 of equilibrium. Average equilibrium time is 570.2 days for a Keynesian firm and 808.37 days for a Marshallian one (which is a statistically significant difference). Moreover Keynesian firms tend to stay closer to equilibrium overall.

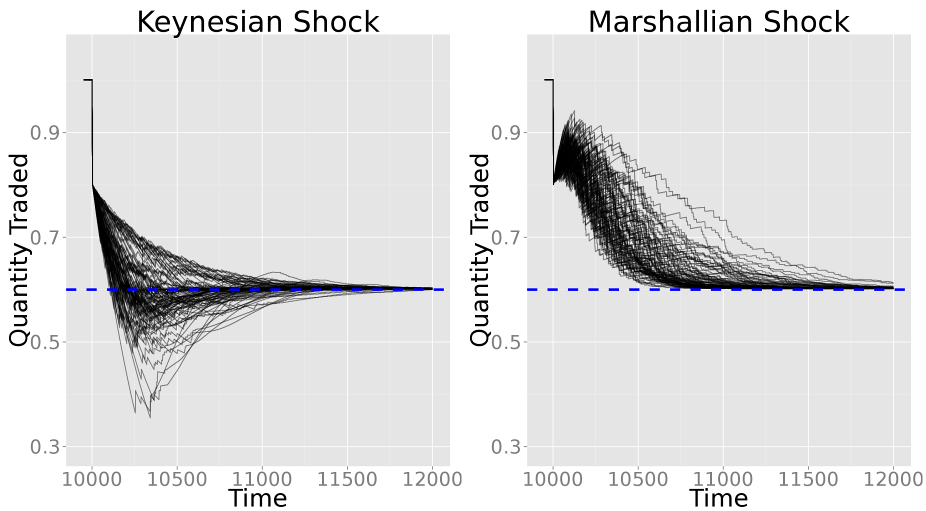


Figure 4.4: A comparison between the adjustment dynamics after a demand shock of Keynesian and Marshallian firms. The Keynesian runs often undershoot and have larger output contractions than the same Marshallian firms in spite of the pre-shock and after-shock equilibria being the same

To see this define deviation of output $y$ from equilibrium $y^*$ as:

$$\log(t) * (y_t - y^*)^2 \tag{4.17}$$

Then the average deviation for Keynesian economy is 4.076 while it is 20.971 in the Marshallian economy. Figure 4.5 shows the difference. On the other hand output drops 10% or more below the new equilibrium in 29 Keynesian runs out of 100 . Marshallian firms never undershoot.

Keynesian adjustment is less efficient and creates larger social losses in spite of reaching equilibrium faster. In figure 4.6 I compare firm profits and labor income during disequilibrium versus what they would be if the adjustment was immediate. Labor income is higher in the Marshallian world (on average 2010.957$ per run compared to 1024.894$ in the Keynesian world). This is because the disequilibrium involves firing unnecessary workers and the longer it takes the more the workers benefit from the disequilibrium. What is less obvious is that the Marshallian firm is also better off than the Keynesian one as figure 4.6 shows.

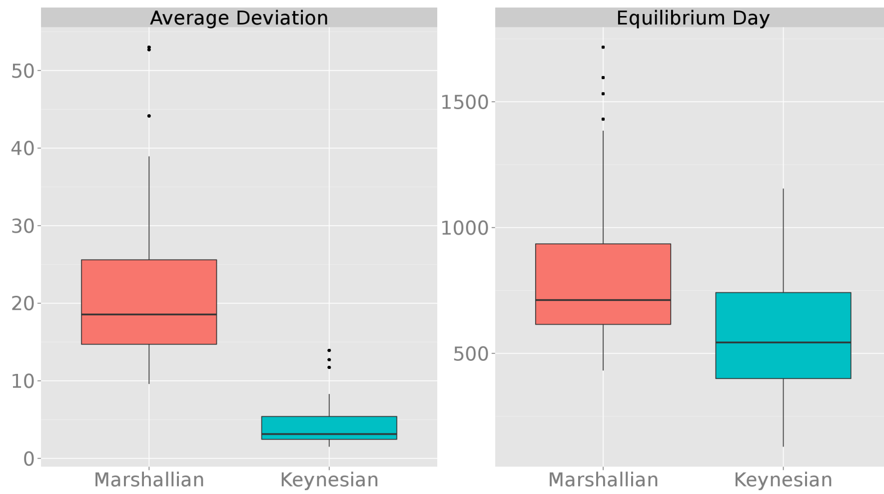


Figure 4.5: Box-plot comparison of deviation and equilibrium day between Keynesian and Marshallian macro

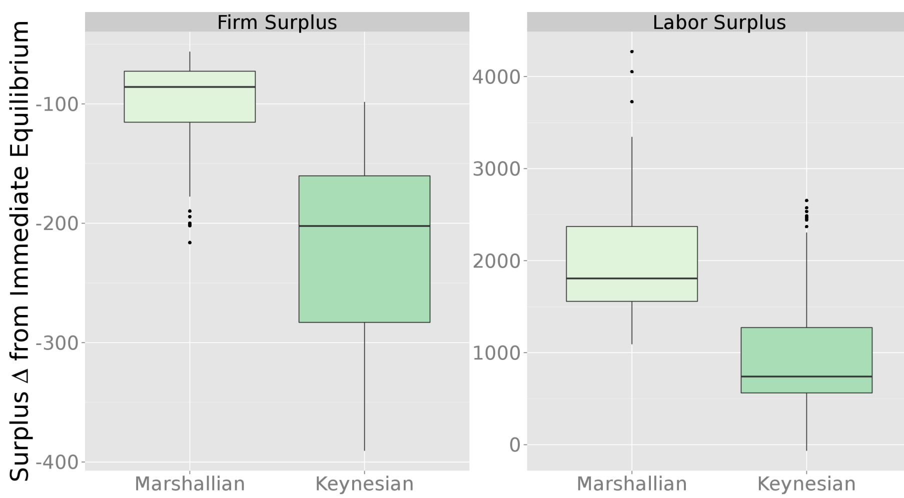


Figure 4.6: The difference in surpluses between Marshallian and Keynesian firms. The surplus is measured as a difference in $ (or wage units) compared to what it would be if it moved immediately to the new equilibrium

The reason Marshallian firms can over-produce for longer and still make consistently less losses than Keynesian firms is that Marshallian disequilibrium dynamics are less wasteful. To see this focus on market day equilibria, each day the difference between what is produced and what is sold is wasted. Figure 4.7 shows the daily waste and in particular how it is larger with Keynesian firms. Keynesian firms over-produce and waste because of their inability to match demand to supply quickly as any cut in production cuts demand as well. Marshallian firm takes longer to get to the new equilibrium but proceeds over a more efficient path where demand and supply match most of the time. Keynesian firms get to equilibrium faster but demand and supply never match until the equilibrium is reached.

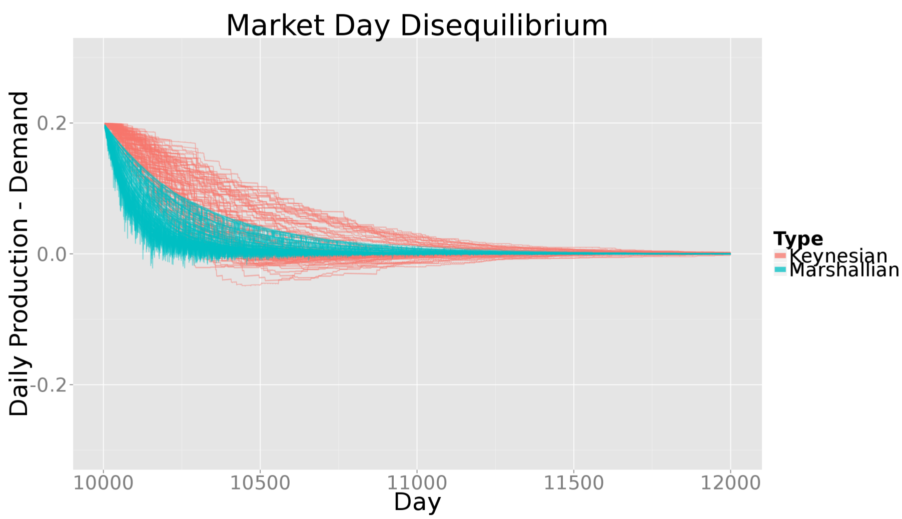


Figure 4.7: The difference between what is produced and what is sold each day, regardless of what the profit-maximizing equilibrium is. The larger the deviation from 0 the more the waste.

Overproduction is the signal that pushes Keynesian firms quickly to the new equilibrium, but it is a wasteful and expensive signal that costs more to society than the slower Marshallian alternative.

## 4.5 Labor Reforms and the Zero-Knowledge Agents

### 4.5.1 Increasing labor flexibility during a recession makes it worse

In this section I model the world as Keynesian. I do so because price rigidities are a well established empirical fact [Klenow and Malin, 2010]. It is also advantageous to model labor market reforms and speed in the Keynesian world since the PID controlling production targets (and therefore labor) is not sticky.

I model labor flexibility in two ways. First, increasing flexibility may mean faster hiring and firing. I can replicate this in the model by increasing the parameters of the PI controlling the workforce so that it adjusts more aggressively. Alternatively increasing flexibility may mean increasing the productivity of labor. I can replicate this in the model by increasing the $a$ parameter of the production function.

Assume the world is Keynesian. Assume the same shock to demand as the previous section. Here I simulate what happens if concurrent to the demand shock there is a flexibility shock to the firm where its PI parameters double. I compare the same simulation with the same random seed with and without the flexibility shock. Notice that the economic equilibria has not changed, the difference can only be in dynamics.

Figure 4.8 shows the effect of increasing flexibility together with the demand shock. Higher flexibility results in higher chance of overshooting, 88 runs out of 100 have output dropping more than 10% below equilibrium (compared to 29 without flexibility shock). Moreover in 10 runs the overshooting is so severe that the run ends on $Y = 0$ (which is a steady state) and never reaches the equilibrium. Figure 4.9 shows that the deviation from equilibrium is higher with higher flexibility (because of the severity of the overshooting) while there is no statistical significant difference in equilibrium time.

Figure 4.10 shows how labor surplus is lower when there is a flexibility shock. Overshooting is so severe that on average the labor surplus is negative. Note first that labor surplus was positive in the previous section: because the new equilibrium requires fewer workers and the agent takes time to get to the new equilibrium point some workers that should have

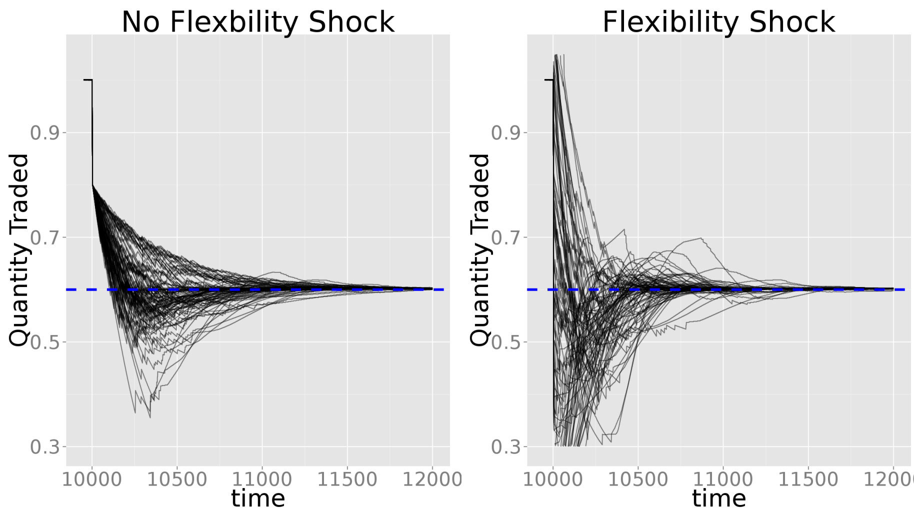


Figure 4.8: 100 Keynesian runs as in figure 4.4 and the same runs where concurrent to the demand shock we double the PI labor parameters. Overshooting becomes more likely and deeper. 10 runs fail to reach equilibrium when their flexibility is increased

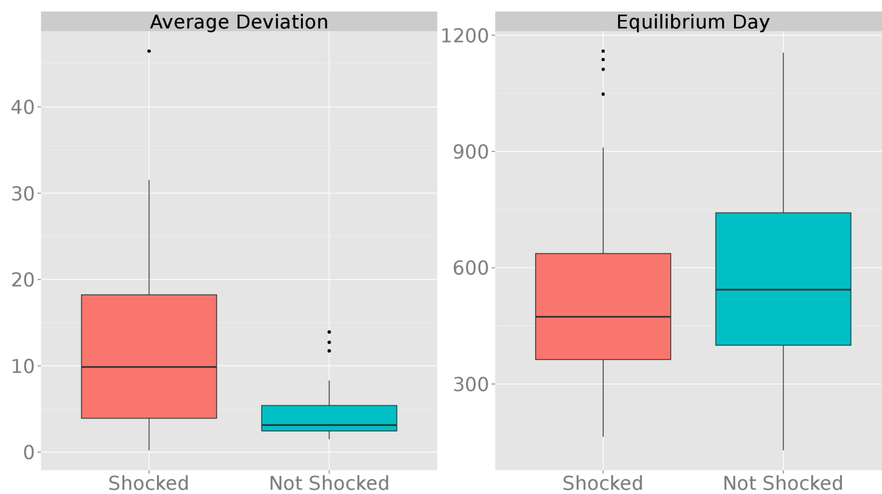


Figure 4.9: Comparison between the Keynesian equilibrium metrics with and without flexibility shock.

been fired instantly profited from the disequilibrium. Higher flexibility fire workers faster, which reduces benefits from disequilibrium and when overshooting it fires too many so that labor overall is hurt by the disequilibrium rather than profiting from it. Firm surplus is higher with more flexibility; the difference in means is statistically significant.

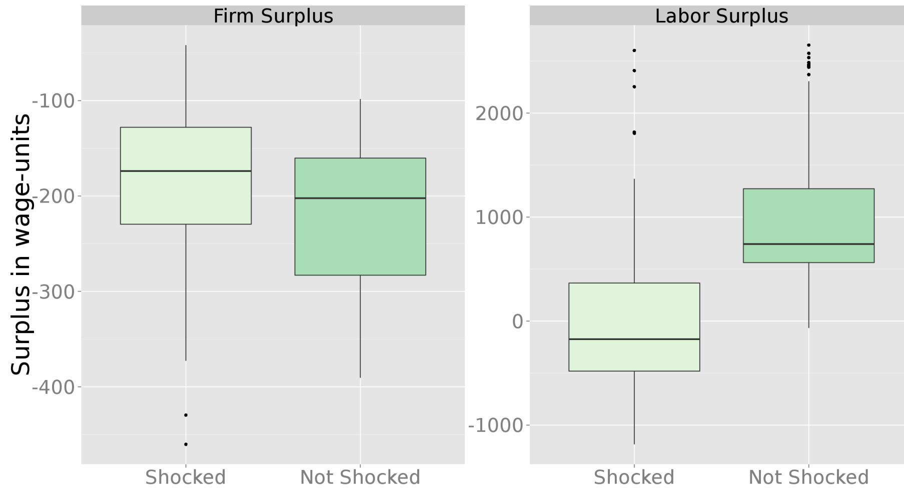


Figure 4.10: Box-plot of surplus differences between runs with and without flexibility shock. The values are \$ (or equivalently wage-units) differences between surpluses and what would the surplus be if the system immediately moved to the new equilibrium

More generally, what the right labor flexibility is in terms of speed is a tuning problem. What we want are the controller parameters that move the economy to the new equilibrium as fast as possible while minimizing overshooting. This is an empirical question and the answer depends on the kind of original equilibrium, production function, shock and every other parameter. It is not the case that more flexibility and speed always make for a better economy.

Turn to flexibility as an alias for productivity, assume again a Keynesian world. Concurrent with a demand shock the productivity $a$ increases from 0.5 to 0.6. This changes the equilibrium $L$ and $p$ but not optimal output $Y$:

$$L = 7.11 \tag{4.18}$$

$$p = 8.88 \tag{4.19}$$

$$Y = 0.6 \tag{4.20}$$

Again I run 100 simulations with and without productivity shock, keeping fixed random seeds for comparison. In this case the only change is in the new equilibrium conditions, PI controllers are invariate. Figure 4.11 compares the two dynamics.

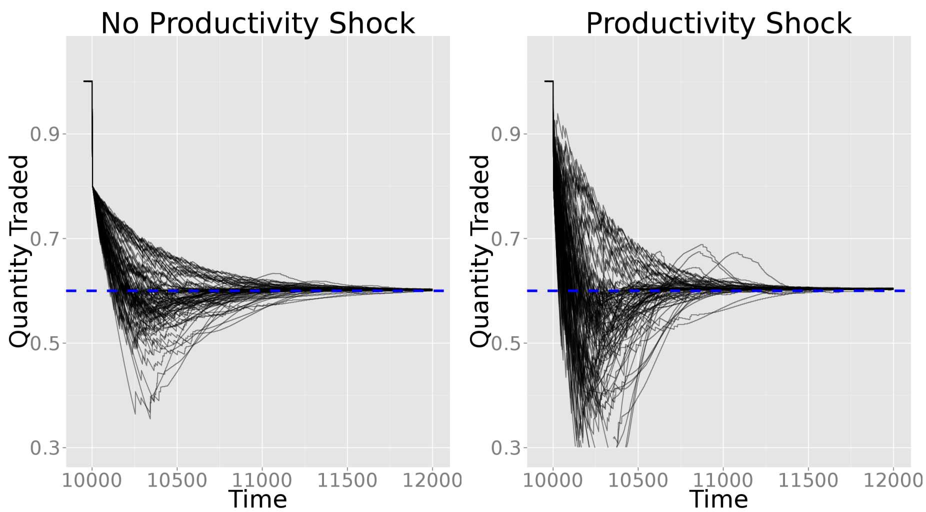


Figure 4.11: The dynamics of 100 Keynesian simulations with paired random seeds and how they deal with demand shock with and without productivity shock.

Increasing productivity makes the approach to equilibrium worse as shown in figure 4.12. More runs undershoot, 68 out of 100, and output deviation from equilibrium is higher with no improvement in equilibrium time. As shown in figure 4.13 there are no meaningful improvements in disequilibrium surplus for either firms or labor although it is hard to judge the overall effect because the equilibrium values the two sets of runs are compared to are different.

While improving productivity is always a good long term policy there is no validation from this model that raising it makes disequilibrium any better.

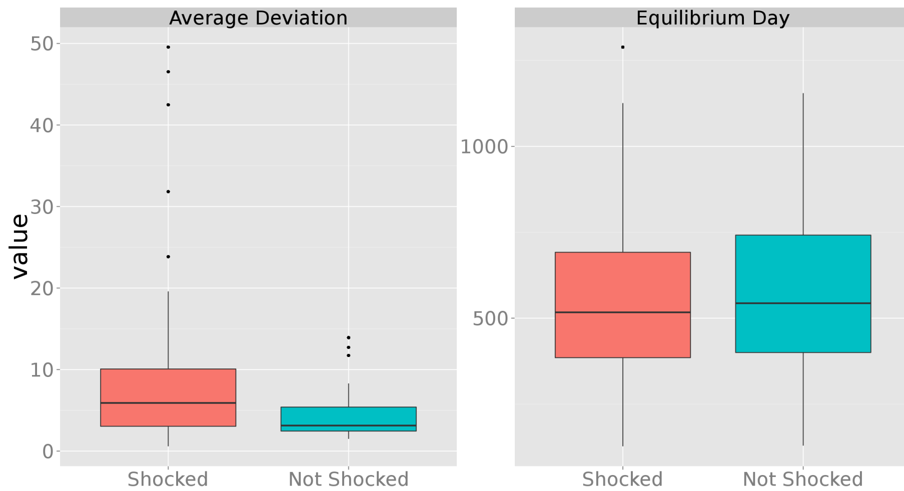


Figure 4.12: Comparison between the Keynesian equilibrium metrics with and without productivity shock.

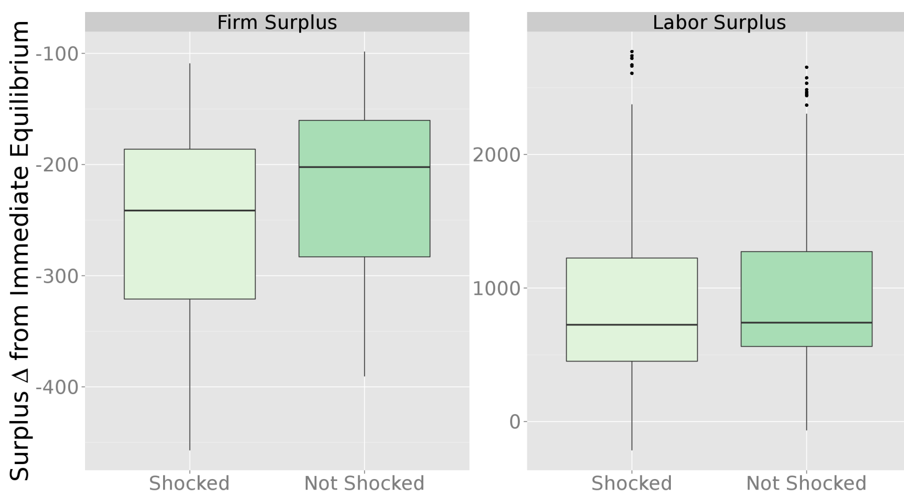


Figure 4.13: Box-plot of surplus differences between runs with and without productivity shock. Notice that the productivity shock changes the equilibrium $p$ and $l$ so that the two classes of surpluses are compared to two different optimal points

## 4.6 Conclusion

To highlight disequilibrium dynamics this model was made very simple. Some of the assumptions present should be removed in future work. The first large assumption I made is infinite fixed wage labor supply. Previous chapters did not assume this. I did so here in order to simplify the decision process of the firm; in this model the firm only sets one price (output) and one production target. Had I added wages it would have made it impossible to compare Marshallian and Keynesian dynamics since there would be two prices to set concurrently. In that circumstance Marshallian firms would be quicker since they would set $p$ and $w$ quickly and target $L$ slowly while Keynesian would have to set $L$ quickly while $p$ and $w$ slowly.

One could argue that we can still use fixed wages around the equilibrium and salvage the shock comparisons in section 4.4.2 by either assuming efficiency wages or some form of downward rigid wages as Modigliani's IS-LM [Modigliani, 1944]. But these would have to be micro-founded rather than just assumed.

The second large assumption is the lack of utility micro-foundations. If the consumer has a lexicographic utility where it prefers a world with no waste (that is demand equals supply) and splits ties according to the world that produces the most, then the simulation would be utility maximizing. But this is non-standard utility formulation and general results must not depend on these. I also gave no explanation for the demand shock.

The third limitation of the chapter is the lack of agents. Previous papers on the methodology had multiple firms competing with one another but here there is a single firm taking all the decisions. This was primarily to avoid any noise in the simulation except those caused by demand shocks. The same weakness is present regarding consumers and workers. A single force supplies labor and consumes wages; there are no distribution effects and no asymmetric cost to unemployment. All these assumptions are, I believe, minor. They are employed to remove noise from the model and further highlight the difference between Keynesian and Marshallian firms.

I believe this paper's result are timely. I show how increases in productivity and labor flexibility during a recession while improving the final economic equilibrium worsen the path the economy takes towards it. This kind of results can only be achieved through agent-based economics and simulation. This kind of results can only be achieved by focusing on disequilibrium. And these results are needed to chart a complete policy response to economic crises.

# Chapter 5: Conclusion

## 5.1 Research Contributions

I presented three essays where simple cybernetic agents find market equilibria without any need for knowledge commonly assumed in economics. Chapter 2 has two main results. The first result of this dissertation is then one of robustness. Economics' basic market results are robust to lack of knowledge and sophisticated rationality. The robustness I find is weaker than the original Zero-Intelligence paper because my agents do not act randomly, but the Zero-Intelligence agents only work under strict statistical assumptions that do not matter here [Cliff et al., 1997].

The second result of chapter 2 is the methodology itself: simple agents that work regardless of market structure. Chapter 3 and 4 are a proof of this result. The agents are placed in a supply-chain first and in a macroeconomic model later and in both of these scenarios they achieve the equilibrium predicted for more rational and more informed agents. Replicating equilibrium is somewhat in line with other "simple" agents like "Probe and Adjust" [Kimbrough and Murphy, 2008] but being able to plug these agents in other markets and still have them working is unique and in my opinion makes this methodology useful for other computational social science applications.

Chapter 3 combines two relatively separate literatures: one is the operations management's "bullwhip effect" literature [Lee et al., 2004] and the other is the macroeconomics' quest for microfounded price stickiness [Klenow and Malin, 2010]. My agents generate bullwhip effects when placed in a supply chain, but this result would only be a minor variation on Sterman's beer distribution game [Sterman, 1995]. My objective was to have agents deal with bullwhip effects without the ability to coordinate or to disseminate better supply-chain information which are the standard approaches of operations management. If we are not dealing narrowly with firms in a supply chain but rather sectors in an input-output table, information sharing becomes impossible due to the size and disconnect between any single firm. Zero-Knowledge agents represent this impossibility of gaining supply-chain information and the way they deal with bullwhip effects is by slowing down the changes in prices. The way the economy as a whole deals with bullwhip effects is sticky prices.

Chapter 4 is an attempt to model more precisely the political discourse surrounding the European labor reforms being now implemented. It is common to refer to the reforms as painful in the short term but beneficial in the long run. Underlying these claims is an idea of disequilibrium as the economy moves from one regime to the other. I believe then that Zero-Knowledge methodology is suited to analyze these claims as it models disequilibrium dynamics directly. Moreover the reforms aim to increase labor flexibility and this itself as a disequilibrium concept as it implies a difference in adjustment speed for firms within a disequilibrium phase. My model shows how faster adjustment can be deleterious after considering the macroeconomic feedback to a firm's adjustments.

## 5.2 Limitations

One major limitation of this dissertation is the lack of a unified sensitivity analysis of the PI parameters. This is because the "best" PI parameters are a function of the simulation parameters themselves (the demand curve, the endowment, the production function and so on). The true issue is that PI performance degrades non-linearly: for large intervals increasing $a$ and $b$ simply means increasing convergence speed, but then increasing slightly more causes runaway oscillations. See figure 5.1: when the PI parameters go from .0001 to .5 all controllers reach equilibrium in due time. But a controller with parameters $a = b = .6$ fails to reach equilibrium and instead oscillates around it.

There are two avenues to solve this directly, both unpalatable. First I could just keep PI parameters small and accept slow convergence speed. The problem with this solution is that the speed of one agent changes the control problem of the others, as the supply chain dynamics in chapter 3 showed. A second solution would be to adapt any self-tuning

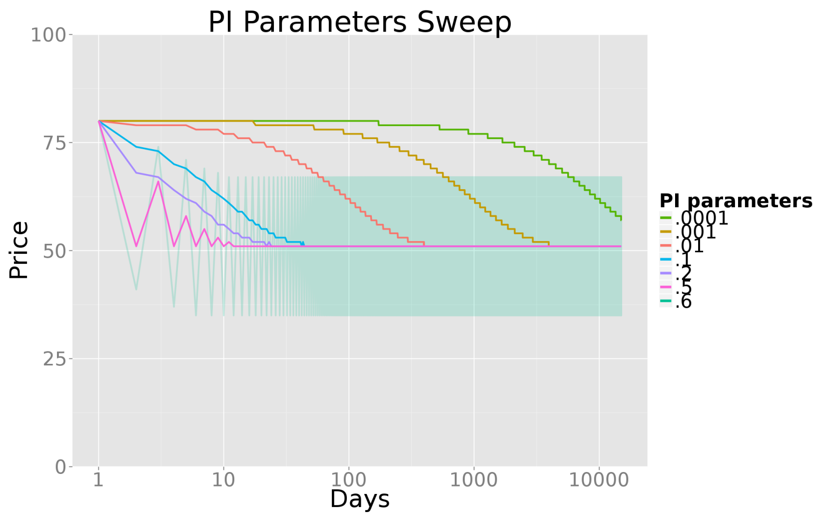


Figure 5.1: Parameter sweep for a PI controller facing demand $y_t = 101 - p_t$ and looking to sell 50 units of goods every day. All curves where $a, b \leq .5$ reach equilibrium, while $a = b = .6$ oscillates out of control. Notice that the time scale is in in $\log_{10}$ scale

algorithm so that traders would adjust their PI parameters on the fly. I did model an example of self-tuning agent in section 3.8 where the upstream firm would experiment with different stickiness values. The main issue with this solution is that tuning is a form of optimization: you are optimizing the adaptation parameters to deal with the current state of the model. Tune too much and you lose PI controller's ability to adapt to model shocks.

A better solution, I believe, would be to recognize that PI parameters, and adaptation speed more generally, are social elements themselves. PI parameters should be modified through innovation and adoption models where more successful firms see their parameters emulated. Done correctly this would achieve the same results of a more complicated self-tuning optimization but allow for enough noise and mutation to change parameters quickly when the model variables are shocked.

A second weakness was my focus on perfect monopoly and perfect competitive scenarios. I chose them because I was trying to replicate the canonical undergraduate economics curriculum. When it comes to agents discovering the economy over time however, perfect competition is the hardest scenario to model. This is because agents have very limited freedom in experimenting with prices, they either capture the whole market or nothing at all. The monopolist market by contrast is the easiest to work with when knowledge is scarce because experimentation has clear and consistent results. It is in fact in imperfect competition that this model will prove its value. Cases of small imperfect competition, oligopolies and such, are well described by game theory. But when the market increases in size and the pricing power of firms is small but significant game theory becomes harder to use. In these "monopolistic competition" models trial and error pricing will shine.

An important lesson from this dissertation was the way agent-based models ought to be distributed. The code, both the in Java and in Dart, has always been open and available to reviewers. But I now believe sharing code is a necessary but not sufficient way to receive the right kind of feedback from the modeling community. Simple, user-friendly GUIs that can run the model quickly and allow even cursory sensitivity analysis on model parameters generate better observations than expecting everyone to look at the code directly.

## 5.3 Future Directions

There are three immediate tasks to complete after this dissertation. First, I need to find the "imperfect competitive" niche where PI controllers will thrive. Too much competition and the PI controller has no room for trial and error, too little and the assumption of no knowledge becomes tenuous. "Monopolistic competitive" markets dominate modern economics in trade, macroeconomics, geography and growth [Chang, 2011], so the potential contribution is large and the cost of adapting Zero-Knowledge traders to them should be small.

Second, I need to tune PI parameters endogenously. I mentioned in the previous section how this will be solved by either engineering some form of self-tuning or by modeling social imitation and adoption. What is missing is an economic scenario that highlights and enables tuning. All the examples in these essays have an optimal steady state. It is impossible, and pointless, to tune a controller that sits in a steady state. The solution will then be to generate an economic scenario where the equilibrium either doesn't exist or continuously changes and study in it the tuning performance of my options.

Third, I need to generalize the Zero-Knowledge trader to use more information. Currently my agents proceed by feed-back only. They ought to have feed-forward elements in their decision making, when necessary. Just as it is hard to believe that agents possess infinite knowledge about the whole economy it is as hard to believe that they possess none.

There are also two more general and long-term tasks I plan to deal with. First I need to integrate the split between my micro-economic and macro-economic models. The chapters share a common agent behavior but they aren't subsets of one another; the supply-chain of chapter 3 does not aggregate into the macro model of chapter 4. I believe the link will be in imperfect competitive markets, as it is in traditional Neo-Keynesian macroeconomics. I prioritized showing how the agents work just as well in macroeconomics but soon it will be time to rebuild the macro-economic scenario from the ground up.

The second long-term task is to focus on the stock costs of disequilibrium. Agents in my model focus exclusively on daily states: prices are set on daily netflow and production is decided on daily prices. In reality pricing an input or output poorly has long term effects for example changes in inventories or damages to the balance sheet. These costs are ignored by the Zero-Knowledge traders who simply look for the daily optimum with no memory and no concern over history. Eventually I want to move towards an agent that deals with stocks just as well.

To develop better tools, then, summarizes what is left to be done. Economics has plenty of tricks to deal with equilibrium but few stay useful in disequilibrium. The PID controller, the Kalman Filter, and Agent-Based Models proved better fit for this kind of analysis. However, the study of disequilibrium has just started and the toolbox for it is still mostly empty. What the toolbox will contain once filled it is impossible to know, but these PID adaptive agents will be in it.

## Bibliography

# Bibliography


[Aguirregabiria, 1999] Aguirregabiria, V. (1999). The dynamics of markups and inventories in retailing firms. *The Review of Economic Studies*, 66(2):275–308.

[Alchian, 1969] Alchian, A. A. (1969). Information costs, pricing, and resource unemployment. *Economic Inquiry*, 7(2):109–128.

[Aoki and Leijonhufvud, 1976] Aoki, M. and Leijonhufvud, A. (1976). Cybernetics and macroeconomics: A comment. *Economic Inquiry*, 14(2):251–258.

[Arifovic, 1994] Arifovic, J. (1994). Genetic algorithm learning and the cobweb model. *Journal of Economic Dynamics and Control*, 18(1):3–28.

[Åström and Hägglund, 1995] Åström, K. J. and Hägglund, T. (1995). *Pid Controllers*. International Society for Measurement and Control.

[Axtell, 2005] Axtell, R. (2005). The Complexity of Exchange*. *The Economic Journal*, 115(504):F193–F210.

[Baganha and Cohen, 1998] Baganha, M. P. and Cohen, M. A. (1998). The stabilizing effect of inventory in supply chains. *Operations Research*, 46(3-supplement-3):S72–S83.

[Bagnall and Toft, 2006] Bagnall, A. and Toft, I. (2006). Autonomous Adaptive Agents for Single Seller Sealed Bid Auctions. *Autonomous Agents and Multi-Agent Systems*, 12(3):259–292.

[Banerjee et al., 1993] Banerjee, A., Dolado, J. J., Galbraith, J. W., and Hendry, D. (1993). Co-integration, Error Correction, and the Econometric Analysis of Non-Stationary Data. OUP Catalogue, Oxford University Press.

[Bennett, 1993] Bennett, S. (1993). *A history of control engineering, 1930-1955*. Number 47. IET.

[Bertola, 2014] Bertola, G. (2014). Labor market policies and european crises. *IZA Journal of Labor Policy*, 3(1):1–11.

[Blackwell and Branke, 2006] Blackwell, T. and Branke, J. (2006). Multiswarms, exclusion, and anti-convergence in dynamic environments. *IEEE Transactions on Evolutionary Computation*, 10(4):459 –472.

[Blanchard, 1982] Blanchard, O. J. (1982). Price Asynchronization and Price Level Inertia. Working Paper 900, National Bureau of Economic Research.

[Blinder, 1998] Blinder, A. S. (1998). *Asking About Prices: A New Approach to Understanding Price Stickiness*. Russell Sage Foundation.

[Carrella, 2014] Carrella, E. (2014). Zero-Knowledge Traders. *Journal of Artificial Societies and Social Simulation*, 17(3):4.

[Cass, 1965] Cass, D. (1965). Optimum growth in an aggregative model of capital accumulation. *The Review of Economic Studies*, 32(3):233–240.

[Chang, 2011] Chang, W. W. (2011). Monopolistic competition and product diversity: Review and extension. SSRN Scholarly Paper ID 1736712, Social Science Research Network, Rochester, NY.

[Chen et al., 2000] Chen, F., Drezner, Z., Ryan, J. K., and Simchi-Levi, D. (2000). Quantifying the bullwhip effect in a simple supply chain: The impact of forecasting, lead times, and information. *Management science*, 46(3):436–443.

[Chen and Deng, 2006] Chen, X. and Deng, X. (2006). Settling the Complexity of Two-Player Nash Equilibrium. In *47th Annual IEEE Symposium on Foundations of Computer Science, 2006. FOCS '06*, pages 261–272.

[Cliff, 1997] Cliff, D. (1997). Minimal-Intelligence Agents for Bargaining Behaviors in Market-Based Environments. Technical report.

[Cliff et al., 1997] Cliff, D., Bruten, J., and Road, F. (1997). Zero is Not Enough: On The Lower Limit of Agent Intelligence for Continuous Double Auction Markets. *HP Laboratories Technical Report HPL*.

[Clower, 1965] Clower, R. W. (1965). The Keynesian counterrevolution: a theoretical appraisal. *The theory of interest rates*, 103:125.

[Cochrane and Graham, 1976] Cochrane, J. L. and Graham, J. A. (1976). Cybernetics and macroeconomics. *Economic Inquiry*, 14(2):241–250.

[Colander et al., 2008] Colander, D., Howitt, P., Kirman, A., Leijonhufvud, A., and Mehrling, P. (2008). Beyond DSGE models: Toward an empirically based macroeconomics. *The American Economic Review*, 98(2):236–240.

[Conlisk, 1996] Conlisk, J. (1996). Why Bounded Rationality? *Journal of Economic Literature*, 34(2):669–700.

[Currie and Steedman, 1990] Currie, M. and Steedman, I. (1990). *Wrestling with Time: Problems in Economic Theory*. University of Michigan Press.

[Deng et al., 2002] Deng, X., Papadimitriou, C., and Safra, S. (2002). On the Complexity of Equilibria. In *IN PROCEEDINGS OF THE 16TH ANNUAL SYMPOSIUM ON THEORETICAL ASPECTS OF COMPUTER SCIENCE*, pages 404–413.

[Dosi et al., 2010] Dosi, G., Fagiolo, G., and Roventini, A. (2010). Schumpeter meeting Keynes: A policy-friendly model of endogenous growth and business cycles. *Journal of Economic Dynamics and Control*, 34(9):1748–1767.

[Evans and Honkapohja, 2009] Evans, G. W. and Honkapohja, S. (2009). Learning and macroeconomics. *Annual Review of Economics*, 1(1):421–449.

[Fabiani, Silvia et al., 2006] Fabiani, Silvia, Druant, Martine, Hernando, Ignacio, Kwapil, Claudia, Landau, Bettina, Loupias, Claire, Martins, Fernando, Matha, Thomas, Sabbatini, Roberto, Stahl, Harald, and Stokman, Ad (2006). What Firms' Surveys Tell Us about Price-Setting Behavior in the Euro Area. *International Journal of Central Banking (IJCB)*.

[Foley, 2010] Foley, D. K. (2010). What's wrong with the fundamental existence and welfare theorems? 75(2):115–131.

[Gintis, 2006] Gintis, H. (2006). The emergence of a price system from decentralized bilateral exchange. *Contributions in Theoretical Economics*, 6(1):1–15.

[Gintis, 2007] Gintis, H. (2007). The Dynamics of General Equilibrium*. *The Economic Journal*, 117(523):1280–1309.

[Gjerstad and Dickhaut, 1998] Gjerstad, S. and Dickhaut, J. (1998). Price formation in double auctions. Technical report, CiteSeerX.

[Gode and Sunder, 1993] Gode, D. K. and Sunder, S. (1993). Allocative Efficiency of Markets with Zero-Intelligence Traders: Market as a Partial Substitute for Individual Rationality. *Journal of Political Economy*, 101(1):119–37.

[Hawkins et al., 2014] Hawkins, R. J., Speakes, J. K., and Hamilton, D. E. (2014). Monetary policy and PID control. *Journal of Economic Interaction and Coordination*, pages 1–15.

[Heijdra and Van der Ploeg, 2002] Heijdra, B. J. and Van der Ploeg, F. (2002). The foundations of modern macroeconomics.

[Howitt and Clower, 2000] Howitt, P. and Clower, R. (2000). The emergence of economic organization. *Journal of Economic Behavior & Organization*, 41(1):55–84.

[Jaffe, 1967] Jaffe, W. (1967). Walras' Theory of Tatonnement: A Critique of Recent Interpretations. 75(1):1–19.

[Kimbrough and Lu, 2005] Kimbrough, S. O. and Lu, M. (2005). Simple reinforcement learning agents: Pareto beats nash in an algorithmic game theory study. *Information Systems and e-Business Management*, 3(1):1–19.

[Kimbrough and Murphy, 2008] Kimbrough, S. O. and Murphy, F. H. (2008). Learning to collude tacitly on production levels by oligopolistic agents. *Computational Economics*, 33(1):47–78.

[Kimbrough and Murphy, 2013] Kimbrough, S. O. and Murphy, F. H. (2013). Strategic bidding of offer curves: An agent-based approach to exploring supply curve equilibria. *European Journal of Operational Research*, 229(1):165–178.

[Kirman, 1975] Kirman, A. P. (1975). Learning by firms about demand conditions. *Adaptive economic models*, pages 137–156.

[Klenow and Malin, 2010] Klenow, P. J. and Malin, B. A. (2010). Microeconomic Evidence on Price-Setting. Working Paper 15826, National Bureau of Economic Research.

[Koopmans, 1963] Koopmans, T. (1963). On the concept of optimal economic growth. Cowles Foundation Discussion Paper 163A, Cowles Foundation for Research in Economics, Yale University.

[Landau et al., 2011] Landau, I. D., Lozano, R., M'Saad, M., and Karimi, A. (2011). *Adaptive Control: Algorithms, Analysis and Applications*. Springer Science & Business Media.

[Lange, 1970] Lange, O. (1970). *Introduction to Economic Cybernetics*. Elsevier.

[Lee et al., 2004] Lee, H. L., Padmanabhan, V., and Whang, S. (2004). Information distortion in a supply chain: The bullwhip effect. *Management Science*, 50(12_supplement):1875–1886.

[Leijonhufvud, 1972] Leijonhufvud, A. (1972). *On Keynesian Economics and the Economics of Keynes: A Study in Monetary Theory*. Oxford University Press.

[Leijonhufvud, 1984] Leijonhufvud, A. (1984). Hicks on Time and Money. *Oxford Economic Papers*, 36:26–46. ArticleType: research-article / Issue Title: Supplement: Economic Theory and Hicksian Themes / Full publication date: Nov., 1984 / Copyright © 1984 Oxford University Press.

[Leontief, 1966] Leontief, W. (1966). *Input-Output Economics*. Oxford University Press, USA, 1st edition.

[Luke, 2009] Luke, S. (2009). *Essentials of metaheuristics : a set of undergraduate lecture notes*. Lulu, [S.l.].

[Luke et al., 2003] Luke, S., Balan, G. C., Panait, L., Cioffi-Revilla, C., and Paus, S. (2003). MASON: A Java multi-agent simulation library. In *Proceedings of Agent 2003 Conference on Challenges in Social Simulation*, volume 9.

[Mankiw et al., 2011] Mankiw, Mankiw, N. G., and Taylor, M. P. (2011). *Microeconomics*. Cengage Learning.

[Mankiw and Reis, 2002] Mankiw, N. G. and Reis, R. (2002). Sticky information versus sticky prices: a proposal to replace the New Keynesian Phillips curve. *The Quarterly Journal of Economics*, 117(4):1295–1328.

[Modigliani, 1944] Modigliani, F. (1944). Liquidity preference and the theory of interest and money. *Econometrica*, 12(1):45–88.

[Mäki, 2008] Mäki, U. (2008). Economics. In Curd, M. and Psilos, S., editors, *The Routledge Companion to Philosophy of Science*, pages 543–554. Routledge, London.

[Nowak et al., 2011] Nowak, A., Rychwalska, A., and Borkowski, W. (2011). Why Simulate? To Develop a Mental Model. *Journal of Artificial Societies and Social Simulation*, 16(3):12.

[Okun, 1981] Okun, A. M. (1981). *Prices and Quantities: A Macroeconomic Analysis*. Brookings Institution Press.

[Ortega and Lin, 2004] Ortega, M. and Lin, L. (2004). Control theory applications to the production–inventory problem: a review. *International Journal of Production Research*, 42(11):2303–2322.

[Papadimitriou, 1994] Papadimitriou, C. H. (1994). On the complexity of the parity argument and other inefficient proofs of existence. *Journal of Computer and System Sciences*, 48(3):498–532.

[Phillips, 2000] Phillips, A. W. (2000). Arnold tustin's the mechanism of economic systems: a review. In *A. W. H. Phillips: A Collected Works in Contemporary Perspective*. Cambridge University Press.

[Pickering, 1995] Pickering, A. (1995). Cyborg history and the World War II regime. *Perspectives on Science*, 3(1):1–48.

[Prescott, 1986] Prescott, E. C. (1986). Theory ahead of business-cycle measurement. *Carnegie-Rochester Conference Series on Public Policy*, 25:11–44.

[Ramsey, 1928] Ramsey, F. P. (1928). A mathematical theory of saving. *The Economic Journal*, 38(152):543–559.

[Romer, 2011] Romer, D. (2011). *Advanced macroeconomics*. McGraw-Hill/Irwin.

[Scarf, 1960] Scarf, H. (1960). Some Examples of Global Instability of the Competitive Equilibrium. *International Economic Review*, 1(3):157–172. ArticleType: research-article / Full publication date: Sep., 1960 / Copyright © 1960 Economics Department of the University of Pennsylvania.

[Shinners, 1998] Shinners, S. M. (1998). *Modern Control System Theory and Design*. John Wiley & Sons.

[Siebert, 1997] Siebert, H. (1997). Labor market rigidities: At the root of unemployment in europe. *The Journal of Economic Perspectives*, 11(3):37–54.

[Skyrms, 2012] Skyrms, B. (2012). LEARNING TO SIGNAL WITH PROBE AND ADJUST. *Episteme*, 9(Special Issue 02):139–150.

[Smith and Corripio, 2005] Smith, C. A. and Corripio, A. B. (2005). *Principles and Practices of Automatic Process Control*. Wiley, 3 edition.

[Sterman, 1995] Sterman, J. (1995). The beer distribution game. *Games and Exercises for Operations Management*, pages 101–112.

[Sterman and Sterman, 2000] Sterman, J. and Sterman, J. D. (2000). *Business Dynamics: Systems Thinking and Modeling for a Complex World*. McGraw-Hill/Irwin, Boston.

[Sterman, 1989] Sterman, J. D. (1989). Modeling Managerial Behavior: Misperceptions of Feedback in a Dynamic Decision Making Experiment. *Management Science*, 35(3):321–339.

[Taylor, 1993] Taylor, J. B. (1993). Discretion versus policy rules in practice. *Carnegie-Rochester Conference Series on Public Policy*, 39:195–214.

[Tustin, 1957] Tustin, A. (1957). *The mechanism of economic systems: an approach to the problem of economic stabilization from the point of view of control-system engineering.* Heinemann.

[Van Overschee et al., 1997] Van Overschee, P., Moons, C., Van Brempt, W., Vanvuchelen, P., and De Moor, B. (1997). RAPID: the end of heuristic PID tuning. *Preprints Proc. PID '00: IFAC Workshop*, 38:6–10.

[Welch and Bishop, 1995] Welch, G. and Bishop, G. (1995). *An introduction to the Kalman filter.*

[Åström and Hägglund, 2006] Åström, K. J. and Hägglund, T. (2006). *Advanced PID control.* ISA-The Instrumentation, Systems, and Automation Society.

## Curriculum Vitae

Ernesto Carrella is a Ph.D candidate in the department of Computational Social Science at George Mason University. He was employed there as a Research Assistant building agent-based models of the economy, the financial market and the housing market. He earned a M.Sc in Economic Policy from the University of Illinois at Urbana Champaign in 2010 where he specialized in Advanced Econometrics. He received his Bachelor in Economics from the Chinese University of Hong Kong in 2008.