ADAPTIVE LEARNING IN CONTINUOUS-TIME: TECHNIQUES AND THEORY

by

CHANDLER REED LESTER

A DISSERTATION

Presented to the Department of Economics
and the University of Oregon Division of Graduate Studies
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy

June 2021

DISSERTATION APPROVAL PAGE

Student: Chandler Reed Lester

Title: Adaptive Learning in Continuous-Time: Techniques and Theory

This dissertation has been accepted and approved in partial fulfillment of the requirements for the Doctor of Philosophy degree in the Department of Economics by:

Bruce McGough	Chairperson
George Evans	Core Member
David Evans	Core Member
Peter Ralph	Institutional Representative

and

Andrew Karduna	Interim Vice Provost for Graduate Studies

Original approval signatures are on file with the University of Oregon Division of Graduate Studies.

Degree awarded June 2021

DISSERTATION ABSTRACT

Chandler Reed Lester

Doctor of Philosophy

Department of Economics

June 2021

Title: Adaptive Learning in Continuous-Time: Techniques and Theory

How we model individual's expectations and predictions in economic models plays an essential role in economic outcomes. We can assume that individuals are well informed and developed nuanced views on the economy, meaning they understand and have detailed knowledge of economic parameters and economic models, or we can suppose individuals are observant and develop perceptions of the economy and make decisions based on available data.

One method of including this level of realistic behavior in economic models is adaptive learning. In adaptive learning models, agents use simple forecasting rules to make predictions about future values of economic variables or the state of the economy. The work presented in this dissertation builds a framework for examining these dynamics in a high-frequency setting. It is important to extend these behavioral modeling techniques to this setting because increasing data are available at higher frequencies. This work combines existing continuous-time modeling techniques with emerging research from economics to develop modelings in which an agent can respond to high-frequency information.

This dissertation demonstrates that complex high-frequency learning is possible and has potential benefits and improvements over discrete-time counterparts. The

dominant theme of this work is defining and mathematically developing a framework for examining bounded rationality in continuous-time models. In chapter two, basic exogenous adaptive rules are explored in a simple Ramsey Model setting. Chapter three introduces shadow-price learning and more complicated endogenous learning rules, including a derivation of continuous-time recursive least squares and the definition of a continuous-time mapping between an agent's perceptions and actuality. Chapter four builds on the dynamics defined in chapter three by applying them to a linearized Real Business cycle model. We find that the continuous-time learning dynamics offer some improvements to the volatility of predictions.

CURRICULUM VITAE

NAME OF AUTHOR: Chandler Reed Lester

GRADUATE AND UNDERGRADUATE SCHOOLS ATTENDED:

University of Oregon, Eugene, OR
Florida State University. Tallahassee, FL

DEGREES AWARDED:

Doctor of Philosophy, Economics, 2021, University of Oregon
Master of Science, Economics, 2017, University of Oregon
Bachelor of Science, 2016, Applied & Computational Mathematics
Bachelor of Science, 2016, Economics and Statistics

AREAS OF SPECIAL INTEREST:

Macroeconomics
Applied Econometrics

GRANTS, AWARDS, AND HONORS:

Dissertation Research Fellowship, University of Oregon, 2020-2021
Graduate Teaching Fellowship, University of Oregon, 2016-2021
Kleinsorge Summer Research Fellowship, University of Oregon, 2019
Promising Scholar Award, University of Oregon, 2016-2018

ACKNOWLEDGMENTS

I thank my committee members-- Bruce McGough, George Evans, David Evans, and Peter Ralph--for their guidance throughout the preparation of this manuscript. I am thankful to each of my committee members for contributing to my research and knowledge throughout my tenure at Oregon. Additionally, I am grateful for Jeremy Piger's mentorship and support throughout this program. I would also like to thank my fellow graduate students in the economics and mathematics departments for helping me prepare my dissertation fellowship application; without your help, I would not have been able to receive this award.

Most importantly, I would like to thank my partner Greyson as well as my friends Jenni, Youssef, and Cory for providing love and support during this entire process.

To Greyson, Jenni, Youssef, & Cory

TABLE OF CONTENTS

Chapter Page

Chapter Page

LIST OF FIGURES

Figure Page

Figure Page

LIST OF TABLES

CHAPTER I

INTRODUCTION

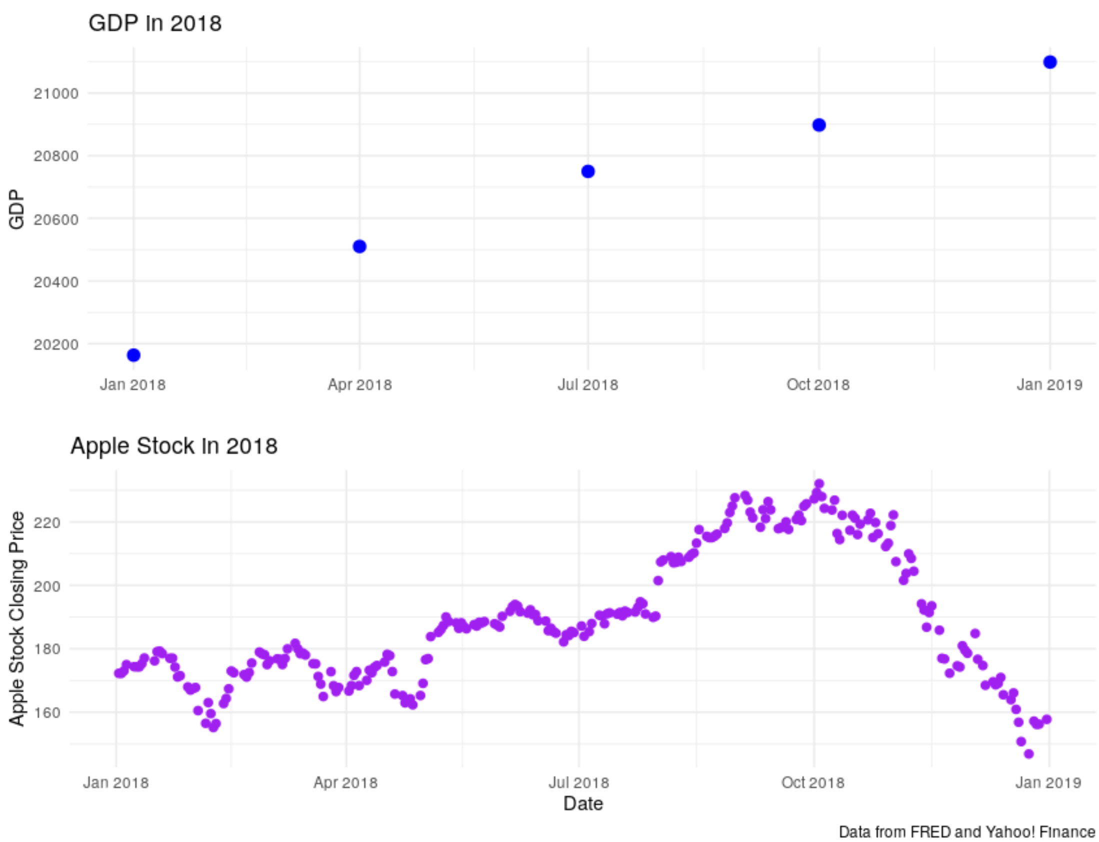
The 2008 financial crisis and ensuing Great Recession permanently altered the global economy and how most people think about their finances. Why did financial markets crash? How did the lending behavior of a few banks lead to a global slowdown? And, perhaps most importantly, what can we learn to help us prevent future similar catastrophes?

We seek to answer these questions through developing more sophisticated models of the economy that capture the financial system's influence over individual decision making; just as previous economic tools were developed in response to crises of demand or production, we now require new tools to anticipate crises caused by financial frictions and failing capital markets. Our research combines work in macroeconomic theory, finance, and behavioral modeling to deliver these tools.

I.1 Continuous-Time Macroeconomics

A decade ago, most economists were not prepared for the housing market to crash. Since, unlike previous economic crises that were caused by issues with demand or production, the 2008 crisis occurred because of financial frictions and failing assets. Often economists do not consider financial assets or regulations; instead, specialists in finance study these. This oversight meant that economists scrambled to understand the origins of recession and how to fix it. One solution that immediately stood out was combining macroeconomic models with financial models and data. Merging these two fields proved difficult since economic and financial data are collected at different frequencies. Financial data, especially stock market data, are collected almost down to the second. Macroeconomic data are often difficult to measure; for example, Gross Domestic Product (GDP) is estimated quarterly and would be costly to measure at a higher frequency.

Figure I.1
A year worth of economic data versus a year of financial data



Data frequency measurements impact the models used in finance and economics. Economists rely on discrete-time models with distinct time periods. In finance, continuous-time models are most common. Time measurements impact many modeling aspects. They alter the way variables change over time, as well as how individuals make decisions and respond to change.

To incorporate financial choices into macroeconomic models, economists need to re-evaluate how we model decision making by altering traditional economic models and standard modeling techniques for continuous-time. The continuous-time macroeconomics literature is rapidly growing and includes many influential papers such as Kaplan et al. (2018),Ahn et al. (2018), and Brunnermeier and Sannikov (2014). Other key papers in the literature include Achdou et al. (2014), which highlights that the continuous-time setting yields more detailed distribution information than discrete-time this contribution is vital since many individuals are interested in information about the distribution of economic variables, such as wealth.

I.2 Behavioral Modeling

We focus on adjusting a behavioral modeling technique called adaptive learning to continuous-time. Adaptive learning—often called learning—is a technique wherein decision-makers estimate model parameters as if they do not know them, but have access to related data. In a simple example, a sock company might want to gather information to price and sell socks optimally. The firm may want to set prices based on expectations of future prices. Since they do not know future prices, they employ an analyst. The analyst runs a simple linear regression using available price data and gives the firm an estimate of what prices might be in the future. Now the company can set a price for their socks. They can also ask the analyst to re-estimate prices later and update the price.

Before the adaptive learning literature emerged, economic models used rational expectations, which assumes that decision-makers understand theoretical models correctly—the decision-makers know the value of all parameters in the model. While rational expectations is a convenient modeling assumption, it is unlikely that individuals in the real world have this level of knowledge (Bray, 1982). Additionally, some rational expectation models do not align with outcomes that appear in the real world.

Some economic situations have multiple outcomes; rational expectations models would likely discover a single result. However, individuals in learning models usually learn both outcomes. By analyzing the stability conditions in the learning dynamics, economists can discover if both outcomes are stable or if one is more likely than the other (Evans and Honkapohja, 2001). Furthermore, learning can allow policymakers to better understand the role of expectations in regard to economic outcomes and how individuals adjust to new policies (Mitra et al., 2019). Often in economics, theoretical models focus too heavily on math and impracticable assumptions about human decision making. Re-examining a model using learning techniques adds more plausibility because these methods add realistic behavior.

I.3 Dissertation Outline

The second chapter of this dissertation takes a close look at a typical class of solution methods frequently used in continuous-time macroeconomics—viscosity solutions. In this chapter, a stylized learning rule is applied to information gathered by an agent who solves for their steady-state equilibrium with misspecified parameters. There two types of stylized learning rules presented in this chapter the first assumes that agents observe the true value of key parameters and gradually update their estimates over time and second is a real-time learning rule in which agents take in data with noise and use this information to update their parameter estimates. In this chapter, the agents are still learning in a discrete-time setting despite existing in a continuous-time economy. This means that agents learn using a slightly re-weighted version of continuous-time adaptive learning rules.

Building on this, the third chapter formally derives both a continuous-time recursive least squares algorithm and a continuous-time optimal linear regulator problem. This is done so that learning can be examined with additional feedback in the continuous-time setting. Using this new solution method and new learning algorithm, we are able to define bounded rationality in this setting. Furthermore, we are able to numerically demonstrate the convergence and stability of shadow-price learning algorithms in this setting.

Lastly, the fourth chapter looks to extend continuous-time learning to additional models. We first examine linearization in this setting using a simple real business cycle model. In this work, we find that when our learning techniques are applied to the real business cycle framework not only does our model converge to rational expectations equilibrium but we are also able to better match second moments from the data. There are many ways to extend the work in this dissertation, including linearizing additional models or extending techniques beyond the linear-quadratic setting. The end of this dissertation outlines these possibilities.

CHAPTER II

ADAPTIVE LEARNING IN A CONTINUOUS-TIME SETTING: REPRESENTATIVE AGENT EXERCISES

II.1 Introduction

Macroeconomic modeling in stochastic continuous-time has become increasingly popular, as solution methods for optimization problems in this setting have been introduced to economics literature. Solutions to optimization problems in this setting take the same form as fluid dynamic problems common in applied mathematics, and it has taken some time for the mathematical solution techniques to become more prevalent in economics. The appeal for economic modeling in this framework comes from several key features of this setting, not just the availability of simple solution methods. Systems in continuous time can be summarized using sparse matrices that are simple to evaluate and use in calculations, leading to fast algorithms that use minimal computational time. This is an attractive feature that allows for complex problems with multiple layers of heterogeneity that can be easily solved. Solutions in this system also yield more detailed and easily computed probability density functions than discrete-time solution methods.

These distributional advantages come from close-ties between the stochastic processes used to summarize the evolution of key variables in these models and their probability density functions. Stochastic processes are defined according to the distribution of the random variables they represent, and optimization problems that depend on these processes inherit some of this distributional dependency (this will be described in more detail in section 2 of this paper). For instance, Gaussian processes, such as the integral of Brownian motion, have a joint normal distribution for all of the variables they define. Poisson point processes are similarly defined using a Poisson distribution. Using these processes that are defined by continuous probability density functions allows researchers to carefully inspect the evolution of the distribution of variables, such as wealth, with little computational burden.

Evaluating these distributions in discrete time is more difficult since probability density functions in this setting are often point masses that truncate the tail-ends of the distribution. Going forward, discrete methods are going to become less favorable as policy becomes more distribution-oriented. Already, the distribution of wealth and assets is becoming a popular topic when it comes to policy goals. Using the traditional discrete methods, central banks and other policymakers will be unable to properly evaluate the effects of their potential actions on distributions of wealth or assets. Since continuous-time modeling has distributional and computational advantages, this modeling framework will become more attractive, and modifying modeling techniques for continuous-time models will be necessary. In this paper, we will take the first steps in examining adaptive learning methods in a stochastic continuous-time framework.

Many macroeconomic models in both continuous and discrete-time depend on agents' expectations. Thus far, continuous-time modeling has depended solely on rational expectations. Using rational expectations limits the model by creating strong assumptions about the agents' information set; rational expectations imply that the agent knows the correct underlying model and that they will respond optimally to the actions of others. These assumptions are unlikely to hold in the real world, as agents may not correctly specify a forecast for the model, and they may not understand the actions of others. Therefore, using a different form of expectations that allows for agents' to make mistakes may be closer to reality. This motivates the use of adaptive learning, a technique that allows for agents to misspecify models and to update their misspecification once they gain more information.

Currently, adaptive learning has been widely implemented in discrete-time modeling; however, it has not been used in continuous-time models. There are two main reasons for this; most economists still use discrete-time models, and learning is more challenging to intuit in continuous-time. As continuous-time modeling becomes more popular, we will want to able to utilize a powerful tool, like adaptive learning, in this setting. The main goals of this paper are to make continuous-time modeling seem more intuitive and less niche to economists and to implement basic adaptive learning techniques in continuous time intuitively.

Sections 2-4 of this paper map out continuous techniques and literature to make these methods more tractable to economists that focus on discrete modeling. Section 2 gives some mathematical background so that the terminology and motivations of continuous-time literature make sense to the reader. The next section provides a literature review that spans a large portion of the economics continuous-time literature and offers more background on adaptive learning. Despite not being widely popular, continuous-time literature spans several decades, has many significant contributions, and includes a large number of papers by notable economists. The fourth section of this paper explores the mathematical relationship between a variety of discrete and continuous-time models. This section should provide a clear link between these models and make continuous-time modeling more intuitive to those who use discrete-time models.

Section 5 begins the task of implementing adaptive learning techniques in continuous-time. A key part of this section is the methodology for finding steady-state solutions in continuous-time. Although the solution methods for stochastic continuous time models currently used in economics have only recently been introduced to the literature, interest in this class of models has existed in the field for a long time. Exploration of the Ramsey model in stochastic continuous-time has been previously studied, notably by Merton (1975), Mirrlees (1966), and Mirman (1973). There are several different methods for implementing a stochastic process in this modeling framework. Some such as Merton (1975) have introduced a stochastic process for capital accumulation. Others, such as Achdou et al. (2014), have used stochastic processes to model productivity. In this paper, we will look at modeling the changes in technological progress and capital as stochastic processes.

This implementation is more intuitive for several different reasons. First, capital accumulation, in part, depends on technological progress; thus, if technological progress can change according to this type of process, capital accumulation with inherently depends on this process as well. Additionally, technological progress is a variable that, in the real world, often seems to change and improve continually. Therefore, it is reasonable to assume that variables like capital stock evolve continuously as they depend on variables we may model continuously, such as technological progress. We can observe technological progress growing over time, so agents are likely to forecast a positive mean and an upward trend. In practice, though, we often are unsure of what sectors or improvements will happen over time, and technological progress is almost constantly evolving. Technological progress is something that most believe is continually improving because of open-source software and near-constant technological improvements in modern productivity.

Before further discussing the work in this paper, it worth reiterating the benefits that come from continuous solution methods. Continuous-time models have unique solutions that can be found using a portable and straightforward algorithm, and these models only need a few weak boundary conditions to obtain unique solutions. Additionally, these solution methods are computationally faster than discrete methods. This means solving complex economic models with heterogeneity can be done with fewer boundary conditions and in less time. A simple description of an algorithm to solve for a steady-state solution in this setting is as follows. First, we discretize the state spaces in our model. This allows us to maintain the continuous-time setting while giving us discrete state spaces to use in a finite difference algorithm. We then implement a finite difference scheme until we get a stable, steady-state estimate of our value function. Despite the discretization, this solution method is different and faster than most discrete methods. Because in this setting, we can summarize the evolution of our system in large sparse matrices.

We can then take advantage of this discretization to implement traditional discrete learning algorithms in continuous-time. The main difference will be the agent's observation over a given time period. When altering adapting learning algorithms for continuous-time, it is tempting to use discrete-methods, since the solution methods for continuous-time problems are discretized. However, this discretization is only over state-spaces so, we must be careful to maintain continuity in our time-dimension. This will be important in section 5 when we examine adaptive learning methods in stochastic continuous-time models.

In this paper we work to accomplish this through two different methods, one method uses supposes that an agent uses a misspecified process to solve for their steady state and then at discrete time periods gain more information and resolves the continuous model. The other method supposes that an agent uses ordinary least squares to create a forecast of model parameters and then at updates this forecast, using recursive least squares, over intervals of time. The first method demonstrates that continuous-models respond in a predictable manner when presented with misspecification and an exogenous updating rule, and the second provides an intuitive way for adapting learning techniques to continuous models. As we proceed with learning in continuous-time, it will be essential to picture our time periods as disjoint intervals of time. Thus, in our forecasting model, each forecasting period contains several observations from our continuous stochastic process. In future work, this could be a key feature of continuous-time learning.

In sections below, we develop two key results that serve as a primer to learning in continuous-time. First, continuous-time models and discrete-time models are somewhat comparable mathematically. This can be seen in section 4, which derives discrete models with an unknown time step (Δt) and then limits these models to their continuous-time counterparts. Our second result is that basic models in this setting respond in an expected fashion to new information, through an exogenous and more discrete updating rule and a more continuous forecasting method. Together these results reveal that further studies on adaptive learning in continuous-time may be promising.

The rest of the paper precedes as follows. The next section gives a basic mathematical background for modeling in this framework. Section three discusses the literature relevant to stochastic continuous-time modeling and adaptive learning techniques in economics. Section four derives the representative agent model in discrete and continuous time Section five describes the exogenous learning rule model and provides the numerical results of this exercise, and section 6 concludes.

II.2 Mathematical Background

Continuous-time optimization problems in economics have a simple general form, and the continuous-time analog of the Bellman equation, the Hamilton-Jacobi-Bellman, can be intuitively derived from the discrete model (Dixit, 1992). Suppose we have a simple Ramsey model where agents maximize their expected utility per unit time over time t

$$\mathbb{E}_0 \sum_{t=0}^{\infty} \sum_{n=0}^{\lfloor \frac{1}{\Delta t} \rfloor} e^{-\rho(t+n\Delta t)} u(c_{t+n\Delta t}) \Delta t, \tag{II.1}$$

where capital evolves according to the following stochastic differential equation

$$\Delta k_{t+\Delta t} = a(k_t, c_t)\Delta t + b(k_t, c_t)\Delta W_t, \tag{II.2}$$

where ΔW_t is the increment of the Wiener process and the maximum value of n, $\lfloor \frac{1}{\Delta t} \rfloor$, limits value of n to integer values. This floor function will be equal to one when $\Delta t = 1$. Note that as $\Delta t \to 1$ equation (II.1) limits to the typical discrete utility maximization problem with a constant discount factor. The Wiener process can be written as $\varepsilon\sqrt{\Delta t}$, where $\varepsilon \sim N(0, 1)$. Thus, we can calculate the expectation and variance of ΔW_t

$$\mathbb{E}[\Delta W_t] = 0 \quad \text{and} \quad \mathbb{E}[(\Delta W_t)^2] = \Delta t.$$

The Bellman equation for this system can then be written as follows,

$$V(k, t) = \max_{c} \; u(c)\Delta t + e^{-\rho\Delta t}\mathbb{E}[V(k + \Delta k, t + \Delta t)] \tag{II.3}$$

in this setting the value function can be thought of as: the value of capital today is equal to the gain from the utility of consumption over one interval of time (Δt) plus expected discounted value the agent receives at $t + \Delta t$. The utility function in (II.3) is multiplied by the length of our time period as we care about the benefits that will accrue in that first period relative to its size (Dorfman, 1969). Since our value function is defined recursively, this expectation captures all future value of capital over time. To get the desired continuous-time value function, we can transform this discrete version (Dixit, 1992). First, using the power series expansion of $e^{-\rho\Delta t}$ we rewrite this problem. [1]

$$\rho \Delta t V(k,t) = \max_c \; u(c)\Delta t + (1-\rho\Delta t)\mathbb{E}[V(k+\Delta k, t+\Delta t) - V(k,t)] \qquad \text{(II.4)}$$

Next we have to use stochastic calculus to find the value of this expectation. In stochastic calculus, we need to apply Itô's lemma to properly take the derivative of a function that depends on a stochastic process. This is necessary because these processes are continuous everywhere, but due to their volatile nature, they are nowhere differentiable.

Suppose, for a moment, that we are in a continuous setting with the following diffusion process,

$$dX_t = \mu dt + \sigma dW_t$$

in this setting μ is a drift term, σ is a variance term, and dW_t is the increment of a Wiener process. If we have a function $f(X_t, t)$ that depends on X_t and time t, we cannot take its derivative using traditional methods since X_t is nowhere differentiable. Instead, we must use Itô's lemma; this will yield

$$df(X_t,t) = \frac{\partial f}{\partial t}dt + \frac{\partial f}{\partial x}\cdot dX_t + \frac{1}{2}\frac{\partial^2 f}{\partial x^2}\cdot (dX_t)^2 + \mathcal{O}(dt^{\frac{3}{2}}).$$

Note, the application of Itô's lemma is essentially just a Taylor expansion of the series using particular assumptions about the stochastic nature of the system. A key assumption of stochastic calculus is at work in the equation above, we assume that all terms with dt^n where $n \geq \frac{3}{2}$ are approximately zero. This will lead to the cancellation of several terms in the expansion of dX_t^2 and all of terms in $\mathcal{O}(dt^{\frac{3}{2}})$. After expanding

[1] The power series expansion of $e^{-\rho\Delta t} = 1 - \rho\Delta t + \rho\Delta t^2 + \mathcal{O}(\Delta t^3)$. One of the common assumptions of stochastic calculus is that terms with including Δt to the power of 3/2 or higher will be approximately zero in the limit. Thus, we will approximate $e^{-\rho\Delta t}$ as $1 - \rho\Delta t$.

terms and rearranging the equation, we will be left with,

$$df(X_t,t) = \left(\frac{\partial f}{\partial t} + \frac{\partial f}{\partial x}\cdot\mu + \frac{1}{2}\frac{\partial^2 f}{\partial x^2}\cdot\sigma^2\right)dt + \frac{\partial f}{\partial x}\cdot\sigma dW_t.$$

Now, if we take the expectation of this the last term will drop out since $\mathbb{E}[dW_t] = 0$. Thus, we will have

$$\mathbb{E}[df(X_t,t)] = \left(\frac{\partial f}{\partial t} + \frac{\partial f}{\partial x}\cdot\mu + \frac{1}{2}\frac{\partial^2 f}{\partial x^2}\cdot\sigma^2\right)dt.$$

Following a similar set of steps, we can look at the difference in our value function over time. Approximating dV as $V(k+\Delta k, t+\Delta t) - V(k,t)$ we can write this as

$$V(k+\Delta k, t+\Delta t) - V(k,t) = V_t(k,t)\Delta t + V_k(k,t)(\Delta k) + \frac{1}{2}V_{kk}(k,t)(\Delta k)^2,$$

here we have already dropped out most terms with t^n where $n \geq \frac{3}{2}$. Carrying through the expectation will the give us the original term from our Bellman equation on the left hand side.

$$\mathbb{E}[V(k+\Delta k, t+\Delta t) - V(k,t)] = V_t(k,t)\Delta t + V_k(k,t)a(k,c)\Delta t + \frac{1}{2}V_{kk}(k,t)b(k,c)^2\Delta t,$$

the $a(k,c)$ and $b(k,c)$ terms come from the original equation for our capital accumulation process given by equation (II.2). Plugging our expectation term into our value function in (II.4) we get,

$$\rho\Delta t V(k,t) = \max_c \ u(c)\Delta t + (1-\rho\Delta t)\big(V_t(k,t) + V_k(k,t)a(k,c) + \frac{1}{2}V_{kk}(k,t)b(k,c)^2\big)\Delta t.$$

Then if we divide by Δt and take the limit as $\Delta t \to 0$ we get the standard HJB

$$\rho V(k,t) = \max_c \ u(c) + V_t(k,t) + V_k(k,t)a(k,c) + \frac{1}{2}V_{kk}(k,t)b(k,c)^2.$$

This HJB equation represents a solution to the given continuous-time maximization problem,

$$\max_{c_t} \int_{t=0}^{\infty} e^{-\rho t} u(c_t) dt.$$

Often, when we are concerned with infinite-horizon problems the $V_t(k,t)$ term will be left out of the HJB. This term is assumed to be zero in infinite horizon problems because as our time dimension becomes infinitely large changes in our value function over (the already infinitely small) increments of time become negligible.

Additionally, in this setting, we might care about the distribution of our state variable k, $g(k,t)$. This distribution is particularly of interest in a setting with heterogeneous agents because heterogeneity and idiosyncratic shocks will impact the evolution of this distribution over time. We can find this distribution using the Kolmogorov Forward Equation (KF), sometimes called the Fokker-Planck Equation. Given an initial distribution $g_0(k)$ the distribution $g(k,t)$ satisfies,

$$\frac{\partial g(k,t)}{\partial t} = -\frac{\partial}{\partial k}[a(k,c)g(k,t)] + \frac{1}{2}\frac{\partial^2}{\partial k^2}[b(k,c)^2 g(k,t)].$$

If a stationary distribution for $g(k)$ exists, it satisfies the ordinary differential equation (ODE)

$$0 = -\frac{\partial}{\partial k}[a(k,c)g(k)] + \frac{1}{2}\frac{\partial^2}{\partial k^2}[b(k,c)^2 g(k)].$$

In a model with multiple agents, the KF equation is one of the key equations that describe the system. In an Aiyagari model, for instance, the KF will determine prices and market clearing, since market clearing is dependent on the distribution of the agents and their preferences. The KF equation is an essential feature in stochastic continuous-time literature; however, it is not used in the representative agent setting present in the rest of this paper. For more information on the derivation and key concepts of the KF equation, see the appendix.

Another way to view the KF equation is as a continuous-time analog to the multiplication of transition matrices (in a Markovian setting). The time-dependent version of this equation describes the evolution of the probability density function of the key variables under the influence of deterministic and random forces found in diffusion processes. The continuous probability distributions that come from our KF equation are one of the most attractive features of continuous-time modeling. Since often with modern policies, we care most about the distribution of goods, wealth, or assets.

Now that we have explored both the HJB and KF equations, it is important to note that the HJB equation is closely related to the maximized Hamiltonian, this is easily shown. First, if we have the system defined in this section with $b(k, c) = 0$ our Hamiltonian is

$$\mathcal{H}(k_t, c_t, \lambda_t) = u(c_t) + \lambda_t a(k_t, c_t),$$

while our HJB equation is

$$\rho V(k) = \max_c \ u(c) + V'(k) a(k_t, c_t).$$

Connecting the two we see $\lambda_t = V'(k)$, i.e. the shadow price of k is equivalent to the marginal value of k. Thus we can rewrite the HJB as

$$\rho V(k) = \max_c \ H(k, c, V'(k))$$

where,

$$H(k, V'(k)) = u(c) + V'(k) a(k, c).$$

The HJB and KF equations, though compact and simple in appearance, can be used to solve complex economic and financial problems. Closed-form solutions to these problems are often impossible to calculate by hand, but with new computational developments finding solutions to these systems has become more plausible, and these solution methods show some advantages to long-popular discrete models.

Not only is continuous-time modeling is more intuitive, but it also provides more information about the distribution of parameters with convenience. This comes from the KF equation that summarizes the distribution of parameters, using the distribution from this equation, researchers can analyze the distribution of a variable over time or after a shock. The distribution that solves the KF equation can also be used for estimating model parameters and can provide a likelihood estimator for the model. Additionally, the algorithms for solving continuous-time systems are fast due to the sparsity of the matrices that determine the evolution of the system.

These modern advances have made continuous-time modeling more attractive to economists since solutions to these systems can now be found without a large number of assumptions. Though these continuous-time problems did not have simple solution methods until more recently, many researchers have explored modeling in a stochastic continuous-time setting.

II.3 Literature

This paper works to develop learning techniques in stochastic continuous time. Therefore we blend two distinct kinds of literature, stochastic continuous-time modeling and adaptive learning. In this section, we will first review the stochastic continuous-time literature. Research on these models in economics has been sparse but spread widely throughout time. For a deeper understanding of this setting and on why it is becoming more relevant today, a historical overview of these modeling techniques is necessary. Learning literature, on the other hand, has been consistently studied for a long time. There is a wealth of knowledge on this topic, and we only examine a small part of this literature that is relevant to our work.

II.3.1 Stochastic Continuous-Time Literature

The stochastic continuous-time setting has become increasingly popular in macroeconomic modeling. Interest in this framework first arose in the early 1970s with financial economic models. These early works include Merton (1969), Merton (1971), and Black and Scholes (1973). In financial economics casting models in continuous time is particularly intuitive as many financial variables evolve, such as stock prices, can be observed on very small intervals; making their prices virtually a continuous variable instead of a discrete one.

Some early works in continuous time financial models include Black and Scholes (1973), Eaton (1981), Merton (1971), Merton (1969), and Mirrlees (1971). These papers set up continuous-time models and solve them as rigorously as possible without the aid of modern computational techniques, often by using comparative statics. This is done because the system of partial differentials that describes equilibrium in this class of models is often unsolvable unless specific forms for the value function are assumed. Due to these identification issues, most of the papers mentioned focus on solving for the distributional steady-state.

Black and Scholes (1973) develops a method for determining fair prices for European call options. Unlike many economic models, Black and Scholes (1973) can assume several boundary conditions and functional forms that aid in solving their key partial differential equations. These boundary conditions and functional forms are such that the HJB can be written in the same form as a standard heat equation. Once the HJB problem is in this format, it is easy to solve for the equilibrium using Fourier transformations. Most optimization problems in this setting cannot be solved for explicitly like the Black-Scholes problem. Part of the reason why this is possible to solve the Black-Scholes model is that it is explicitly defined for European call options that can only be called at the end of their lifespan.

Eaton (1981) explores the effects of fiscal policies on the composition of portfolios and the accumulation of capital. This model defines the net output, government expenditure, and tax revenue as stochastic processes. All of these processes depend on aggregate capital stock, which allows the government in this model to tax the random component of capital income at a different rate than the deterministic part and defines government expenditure to depend differently on the deterministic and random parts of capital. After setting up this model, the author uses comparative statics and some simplifying assumptions to conclude that fiscal policy changes impact the average yield and riskiness of capital relative to government debt.

Robert Merton has several papers from this period that develop models in stochastic continuous-time. Merton (1969) develops a model for optimal portfolio selection where returns on assets generate the agents' income. Merton (1971) further examines this problem and uses explicit forms for the utility function to derive optimal consumption and portfolio rules. This paper also uses comparative statics to examine the response of these rules to certain parameter changes, a popular technique during this time. Merton (1975) examines standard economic growth models in this setting. The model discusses in Merton (1975) is a one-sector neoclassical growth model where the size of the labor force evolves according to a stochastic process. The paper then takes the neoclassical growth model and expands it into a stochastic Ramsey problem. Merton (1975) is one of the first publications that use more traditional economic modeling in this stochastic continuous-time setting. Another paper that implements traditional economic models is Brock and Mirman (1972).

Brock and Mirman (1972) differs from these other papers because, in this model, a solvable steady state exists. This growth model is unique due to the linearity of the consumption function. This allows for the steady-state of the stochastic model to be equal to the steady-state of the non-stochastic model. Due to the tractability of this model, the Brock-Mirman model is one of the most common stochastic continuous models used before the introduction of more advanced computational methods.

Dixit (1989) models firm entry and exit decisions where output price follows a geometric Brownian motion, this model is solved by simplifying the system of PDEs into a simpler system of ordinary differentials. This produces a solution that consists of trigger prices for firm entry and exit. Prices in between the entry trigger and the exit trigger price produce “hysteresis,” which appears in the model even with small sunk costs.

During the late 1990s and early 2000s a number of books were published on continuous time models in financial economics; these include Merton (1992), Dixit (1992), Dixit and Pindyck (1994), and Stokey (2009). The publication of these works formalized the use of continuous-time models, particularly in finance. Merton (1992) contains several of Merton’s papers mentioned earlier in this literature review and is directed at finance graduate students. Dixit and Pindyck (1994) is also targeted at finance graduate students and contains some of the most intuitive derivations of the HJB equation out of all economics and finance literature. Dixit (1992) includes intuitive mathematical introductions and focuses on how to implement boundary conditions in the stochastic continuous-time setting. Stokey (2009), differs from the other books on stochastic continuous-time modeling. This book focuses more on the mathematical background and measure theory that is necessary for a deeper understanding of this material. The main contribution of this work is the understanding that continuous-time modeling better captures the dynamics of inaction and boundaries that are rarely binding. This setting’s ability to capture inaction and boundary conditions is the reason why stochastic continuous-time modeling has become so popular in financial economics.

With the availability of better computational methods, more econometric papers have been written on stochastic continuous-time models. Hansen and Scheinkman (1995) derive moment conditions for estimating and testing continuous-time Markov models using discrete-time data. Aït-Sahalia has several economics and finance papers published throughout the 1990s and early 2000s on econometric tests for diffusion processes. Aït-Sahalia (2002) constructs a maximum-likelihood approach to estimating parameters in discretely sampled diffusion models. Aït-Sahalia (2004) furthers the methods from Aït-Sahalia (2002) and constructs an approach to estimating parameters in these models when the sampling intervals are not uniform. Posch (2009) solves continuous time dynamic stochastic general equilibrium models with jumps and shows how the continuous-time setting can make it simpler to estimate the likelihood function. This paper solves the model by introducing several simplifying assumptions and confirming the results with Monte Carlo estimates.

Most stochastic continuous-time modeling in the early 2000s used assumptions about the form of the value function or by imposing multiple boundary conditions. Financial economists such as Sannikov extended stochastic continuous-time modeling to a microeconomic setting. In Sannikov (2008) and DeMarzo and Sannikov (2006) a principal-agent setting is developed in continuous-time. Solutions to these principal agents are found by implementing several boundary constraints, which at the time of their publication was an innovative technique. This technique opened up the door for more publications in the stochastic continuous-time setting.

Hansen et al. (2006) takes a more theoretical approach to stochastic continuous-time modeling and explores model misspecification in this setting. Duffie and Epstein (1992) develops a stochastic differential formulation of recursive utility. Gabaix (2009) has a section on continuous-time approaches to power laws. In this paper, the size of an economic unit (cities or firms) is modeled as a stochastic process that can hit reflective boundaries at some points. Using this process, one can use the KF equation to describe the evolution of this distribution using power laws a unique solution to this system can be found.

Before 2015 economists were not widely implementing computational methods to find solutions to this class of optimization problems. However, Forsyth and Labahn (2007), a computational finance paper, studies numerical methods for solving HJB equations in finance. This paper finds that discretizing the HJB and solving it numerically will converge to the viscosity solution. The viscosity solution is the same solution that economists began focusing on around 2015. Viscosity solutions are continuous and differentiable solutions to the HJB that are in most cases unique. Forsyth and Labahn (2007) also analyzes Newton-type iterations schemes and finds that these also solve the HJB equation, another result economists realized later.

With the rise of heterogeneity in macroeconomics, economic models have developed new more complexity. Discrete-time models can capture rich heterogeneity; however, these methods are time-consuming and cannot provide the same level information about the distribution of key variables as continuous-time models. Many recent papers focus on developing and implementing these algorithms to solve these new richer models.

Achdou et al. (2014) uses tools from applied mathematics to solve the HJB equation. The algorithm outlined in the paper uses finite difference methods to solve for an approximate solution to the HJB. This approximate solution, called the viscosity solution, assumes that the value function is differentiable on its entire domain. Viscosity solutions are unique, given that several weak conditions hold. In Achdou et al. (2014), this method allows the authors to find both steady-state and time-dependent solutions for their models. Other papers such as Kaplan et al. (2018), Achdou et al. (2020), and Parra-Alvarez et al. (king) implement the same techniques. This paper uses the steady-state solution methods presented in Achdou et al. (2014) in the exogenous learning rule model.

A key issue with the solution methods presented in Achdou et al. (2014) is that the time-dependent solutions cannot be used in conjunction with random aggregate shocks. Ahn et al. (2018) uses the foundation developed in Achdou et al. (2020) to create a more complicated algorithm for analyzing models with heterogeneous agents that are subject to shocks. This algorithm calculates the steady-state versions of the HJB and KF equations and then linearizes the system around that steady-state without aggregate shocks. Linearization around this steady-state involves using a first-order Taylor expansion since this system has a large number of variables the derivatives needed for this Taylor expansion cannot be taken by hand and must be calculated using automatic differentiation.

After the system is linearized, it can be easily solved, and the Schur decomposition of the coefficient matrix can be used to check for stable roots. Using this algorithm, one can look at impulse response functions and the effects of shocks on a continuous model. The algorithm in this paper is an important innovation as previous solution methods prevented researchers from analyzing random macroeconomic shocks. Being unable to analyze these types of shocks was a drawback of stochastic continuous-time modeling in macroeconomics. Now that a simple portable algorithm for analyzing these types of models exists, the stochastic continuous-time setting is likely to become increasingly popular among researchers in theoretical macroeconomics.

The representative agent model outlined in this paper will use the same approach as Ahn et al. (2018) to solve the model and to implement learning in this framework.

II.3.2 Learning Literature

The motivation of this paper is to develop adaptive learning techniques in the stochastic continuous-time setting. Adaptive learning is a statistical approach that overcomes the strict model assumptions implied by traditional rational expectations. In learning models, agents use statistical techniques to estimate model parameters and update their expectations of parameters and other values over time. Most learning papers involve direct feedback from the agents' estimates through a special mapping called a T-map. This paper relies on exogenous learning rules that appear similar to simple econometric learning as described in Evans and Honkapohja (2001); however, the algorithms in section five do not have this feedback rule. Instead, the learning in this paper comes from simple information drops, and all new information is used to update parameter estimates directly.

This method of learning is more similar to the early works in this literature. For instance, Bray (1982) looks at a more simple version of updating estimates via OLS. Some of the models explored in this paper do not look directly at feedback rules and instead focus on seeing if agents can get rational expectations estimates of parameters when presented with additional information. The agents do this by implementing OLS each period with updated information. This paper found that under some assumptions, the OLS learning converged to the rational expectations equilibrium's values. Also, learning in this paper focuses on learning parameters in a steady-state setting. Similar environments have explored previously work, notable steady learning, as mentioned in Evans and Honkapohja (2009).

There does exist some literature similar to stochastic continuous-time adaptive learning in asset pricing literature. Veronesi (2019) examines a Bayesian learning rule in an asset pricing model with heterogeneous risk preferences. Some other papers, such as Bhamra and Uppal (2014), also discuss implementing a similar learning rule. The work in these papers is distinctly different than what we will proceed with, since the focus of these works is finding parameters based on distributions.

In this paper, one of the main focuses in our learning section is adapting misspecification. There has been some work on this within asset pricing literature, notably Hansen and Sargent (2019b) and Hansen and Sargent (2019a). These papers look at misspecification within models and also look at an agent's choice between several well-defined models. All of these asset pricing models are cast in stochastic continuous-time. This is done to exploit the convenient properties of Brownian motion and continuous likelihood functions.

II.4 A Representative Agent Model

In this section, we develop discrete and continuous models in deterministic and stochastic settings to better understand the connections between discrete and continuous models. This is done with a few simple Ramsey models. We first develop the discrete and continuous models separately before comparing them carefully. A critical feature of the discrete methods is the inclusion of time increments Δt, which allows us to compare our discrete and continuous models. The use of Δt in the following sections is based on previous work by Dorfman (1969). Doing this allows us to understand the similarities of discrete and continuous-time systems better, and creates a discrete setting to develop a benchmark for how learning should impact a system with infinitely small time intervals. This section of the paper proceeds by developing deterministic and stochastic versions of the model in discrete and continuous-time. Next, we will compare these models and show how they are related as increments of time get infinitely small. In both of the cases outlined below, the discrete model limits to the continuous version.

II.4.1 A Deterministic Model

Before worrying about systems with stochasticity, we first outline a simple Ramsey model in a deterministic setting. First, we will describe the discrete case and the continuous case separately. Then, we will compare the two models.

The Discrete-Time Deterministic Model

Discrete-time models in economic often assume that $\Delta t = 1$. This assumption makes models less notationally bulky. However, in doing so, information is lost, the utility functions used in economics are utility *per unit time*, and our discrete discount factor is dependent on units of time as well. The model outlined below considers these units of time, and carefully examines the optimality conditions with this Δt term.

Before describing the model, it is important to understand the discount factor's dependence on time. The discount factor β is defined as the discount rate per unit of time and can be written as a function of the increment of time $\beta(\Delta t)$. Furthermore, $\lim_{\Delta t \to 0} [\beta(\Delta t)]^t = e^{-\rho t}$.

Using this discount factor, we can proceed with our model. A representative agent in this setting will maximize utility per unit time according to

$$\max_{c_t} \mathbb{E}_0 \sum_{t=0}^{\infty} \sum_{n=0}^{\lfloor \frac{1}{\Delta t} \rfloor} \beta^{t+n\Delta t} u(c_{t+n\Delta t}) \Delta t, \tag{II.5}$$

here $\lfloor \frac{1}{\Delta t} \rfloor$ limits n to integer values, since $\Delta t \leq 1$. In this setting, capital evolves according to the following process[2]

$$k_{t+\Delta t} = (e^{z_t} f(k_t) - \delta k_t - c_t)\Delta t + k_t \tag{II.6}$$

where $f(k_t) = k_t^{\alpha}$.[3] In this deterministic setting we will have the following process for the evolution of productivity z_t, [4]

$$z_{t+\Delta t} = (1 - \eta \Delta t) z_t \tag{II.8}$$

[2]Setting

$$\dot{k} = \frac{k_{t+\Delta t} - k_t}{\Delta t} = e^{z_t} f(k_t) - \delta k_t - c_t$$

(II.6) is the typical equation for the evolution of capital in a discrete Ramsey model

[3]In this discrete model if we normalize $\Delta t = 1$, (II.6) is the standard equation for capital accumulation.

$$k_{t+1} = e^{z_t} f(k_t) + (1 - \delta) k_t - c_t$$

[4]In a stochastic setting productivity z_t will evolve according to the following AR(1) process. This process was derived from the standard Ornstein-Uhlenbeck process in (II.30) using the Euler-Maruyama method

$$z_{t+\Delta t} = (1 - \eta \Delta t) z_t + \sigma \epsilon_t \sqrt{\Delta t} \tag{II.7}$$

Here $\epsilon_t \sim N(0, 1)$.

This model is closely related to the stochastic continuous model outlined later in this section. We can note that Δt becomes dt in the limit, using this the equations (II.5)-(II.8) will be equivalent to the ones used for the continuous deterministic model.

Optimization problems in this setting can take several different forms. First, we can write out the Lagrangian.

$$\mathscr{L}(z_0, c_0, \lambda_0) = \mathbb{E}_0 \sum_{t=0}^{\infty} \sum_{n=0}^{\lfloor \frac{1}{\Delta t} \rfloor} \beta^{t+n\Delta t} \{u(c_{t+n\Delta t})\Delta t$$
$$+ \lambda_{t+n\Delta t}[k_{t+n\Delta t} + (e^{z_t} f(k_{t+n\Delta t}) - \delta k_{t+n\Delta t} - c_{t+n\Delta t})\Delta t - k_{t+(n+1)\Delta t}]\}$$

In this setting our first order conditions will be the following,

$$\frac{\partial \mathscr{L}}{\partial c_t} = \frac{\partial u}{\partial c}\Delta t - \lambda_t \Delta t = 0 \tag{II.9}$$
$$\frac{\partial \mathscr{L}}{\partial k_{t+\Delta t}} = \beta^{t+\Delta t}\mathbb{E}_t \lambda_{t+\Delta t}\big[\big(e^{z_t} f'(k_{t+\Delta t}) - \delta\big)\Delta t + 1\big] - \beta^t \lambda_t = 0 \tag{II.10}$$
$$\frac{\partial \mathscr{L}}{\partial \lambda_t} = k_t + (e^{z_t} f(k_t) - \delta k_t - c_t)\Delta t - k_{t+\Delta t} = 0 \tag{II.11}$$

where we have supposed that $\lambda_{t+\Delta t} = \lambda_t + \dot{\lambda}\Delta t$, where $\dot{\lambda}$ is that rate at which it will change over our interval of time. Since this setting is deterministic, we can now drop the expectation term. Then we can rewrite (II.10).

$$\frac{\partial \mathscr{L}}{\partial k_{t+\Delta t}} = \lambda_t[e^{z_t} f'(k_{t+\Delta t}) - \delta] = -\lambda_t \ln \beta - \dot{\lambda} \tag{II.12}$$

For a full derivation of (II.12) see the appendix.

We can look at this problem from a Hamiltonian framework. In this setting the current value Hamiltonian is,

$$J(k_t, \mu_{t+\Delta t}, c_t, t, t+\Delta t) = u(c_t) + \mu_{t+\Delta t}(e^{z_t} f(k_t) - \delta k_t - c_t) + \gamma_{t+\Delta t}\big(-\eta z_t \Delta t\big) \tag{II.13}$$

with the transversality condition

$$\lim_{t\to\infty} \beta^t \mu_t k_t \leq 0. \tag{II.14}$$

The first order conditions for this system are given by the following equations,

$$\frac{\partial J}{\partial c_t} = u'(c_t) - \mu_{t+\Delta t} = 0 \tag{II.15}$$

$$\frac{\partial J}{\partial k_t} = \mu_{t+\Delta t}(e^{z_t} f'(k_t) - \delta) = -\frac{\mu_{t+\Delta t} - \mu_t}{\Delta t} - \ln(\beta)\mu_{t+\Delta t} \tag{II.16}$$

in this setting $\frac{\mu_{t+\Delta t}-\mu_t}{\Delta t} = \dot{\mu}_t$. At a glance (II.16), looks very similar to (II.12). Using (II.15) and (II.16) we can get the typical first order conditions for a Hamiltonian system,

$$\big(e^{z_t} f'(k_t) - \delta + \ln(\beta)\big) = -\frac{u''(c_t)}{u'(c_t)}\dot{c}_t.$$

This is similar to the continuous time version; however, the multiplier in this case is incremented forward one unit of time, and our discrete discount rate causes our first order conditions to include a $\ln(\beta)$ term. As the increment of time approaches zero, the discrete Hamiltonian outlined here will be equivalent to the continuous Hamiltonian described in the following section.

We can also write a discrete Bellman equation for this system

$$V(z_t, k_t) = \max_{c_t}\ u(c_t)\Delta t + \beta^{\Delta t}[V(z_{t+\Delta t}, k_{t+\Delta t})]. \tag{II.17}$$

This setting will have similar first conditions. First we can take the first order condition of this system with respect to c_t

$$\frac{\partial u}{\partial c_t}\Delta t + \beta^{\Delta t}\frac{\partial}{\partial c_t}V(z_{t+\Delta t}, k_{t+\Delta t}, t+\Delta t) = 0. \tag{II.18}$$

In this case $\frac{\partial}{\partial c_t}V(k_{t+\Delta t}, z_{t+\Delta t}, t+\Delta t) = \frac{\partial V(\cdot)}{\partial k_{t+\Delta t}}\frac{\partial k_{t+\Delta t}}{\partial c_t}$. Thus, we will have

$$\frac{\partial u}{\partial c_t}\Delta t = \beta^{\Delta t}\frac{\partial V(\cdot)}{\partial k_{t+\Delta t}}\frac{\partial k_{t+\Delta t}}{\partial c_t}\Delta t.$$

Simplifying and denoting the marginal value of capital at time t as $\mu_t = \frac{\partial}{\partial k}V(k,t)$ we will have the following equation

$$\frac{\partial u}{\partial c_t} = \beta^{\Delta t}\mu_{t+\Delta t}\frac{\partial k_{t+\Delta t}}{\partial c_t},$$

this is equivalent to (II.15). Taking the first-order condition with respect to k will then yield,

$$\mu_t = [1 + \big(e^{z_t}f'(k_t) - \delta\big)\Delta t](\mu_{t+\Delta t})\beta^{\Delta t}$$

Simplifying this will give us (II.16) from our discrete Hamiltonian. Examining this, we can see that the value function and Hamiltonian are closely related as in Dorfman (1969).

The Continuous-Time Deterministic Model

The continuous-time version of this model can be described according to the following equations. Our agent will maximize expected utility according to the following equation, here $e^{-\rho t}$ will be the continuous time equivalent of the discrete discount factor β^t,

$$\max_{c_t}\ \mathbb{E}_0\int_{t=0}^{\infty} e^{-\rho t}u(c_t)dt. \tag{II.19}$$

This is setting capital will evolve according to the following process

$$dk_t = (e^{z_t}f(k_t) - \delta k_t - c_t)dt, \tag{II.20}$$

where the production function is the same as before. Productivity will evolve according to

$$dz_t = -\eta z_t dt, \tag{II.21}$$

the continuous time analog to the discrete process in the previous section.

In this setting, the current value Hamiltonian can be rewritten as follows,

$$H(k_t, z_t, c_t, \gamma_t, \mu_t, t) = u(c_t) + \mu_t(e^{z_t} f(k_t) - \delta k_t - c_t) - \gamma_t(\eta z_t).$$

It is clear that $H(\cdot) = \lim_{\Delta t \to 0} J(\cdot)$, thus this directly related to our discrete time problem. The following equations will give the first-order conditions for this system.

$$\frac{\partial H}{\partial k_t} = \mu_t(e^{z_t} f'(k_t) - \delta) = -\frac{d\mu_t}{dt} + \rho\mu_t \tag{II.22}$$

$$\frac{\partial H}{\partial c_t} = u'(c_t) - \mu_t = 0 \tag{II.23}$$

The transversality condition in continuous time can be written as follows.

$$\lim_{t \to \infty} e^{-\rho t} \mu_t k_t \leq 0$$

Together the first order conditions (II.22) and (II.23) imply,

$$u'(c_t)(e^{z_t} f'(k_t) - \delta - \rho) = -\frac{d\mu_t}{dt}$$

We can also write a HJB for this system, since we are in a continuous time setting.

$$\rho V(k, z) = \max_c \; u(c) + \partial_k V(k, z)(e^z f(k) - \delta k - c) - \partial_z V(k, z)(\eta z) \tag{II.24}$$

Setting μ_t in the current value Hamiltonian equal to $\partial_k V(k, z)$, and γ_t equal to

$\partial_z V(k,z)$ we can rewrite this again.

$$\rho V(k,z) = \max_{c} \; H\big(k, z, c, \partial_k V(k,z), \partial_z V(k,z)\big) \tag{II.25}$$

Comparing the Deterministic Models

For a clear comparison of the discrete and continuous time models outline in this section, we can examine the discrete Bellman equation (II.17) as $\Delta t \to 0$. First, we can take an approximation of $V(k_{t+\Delta t}, z_{t+\Delta t})$, in a method similar to Dorfman (1969).

$$V(k_{t+\Delta t}, z_{t+\Delta t}) = V(k_t, z_t) + \partial_k V(k_t, z_t)(k_{t+\Delta t} - k_t) + \partial_z V(k_t, z_t)(z_{t+\Delta t} - z_t) + \mathcal{O}\big(\Delta t^{\frac{3}{2}}\big)$$

All other partials and cross partial derivatives will be in the $\mathcal{O}$ term. These terms will all be approximately zero in the limit as $\Delta t \to 0$. Next, we will approximate $\beta^{\Delta t} \approx e^{-\rho \Delta t} \approx (1 - \rho\Delta t)$. Using these two approximations we can rewrite (II.17) as follows.

$$V(k_t, z_t) = \max_{c_t} \; u(c_t) + (1 - \rho\Delta t)[V(k_t, z_t) + \partial_k V(k_t, z_t)(k_{t+\Delta t} - k_t) + \partial_z V(k_t, z_t)(z_{t+\Delta t} - z_t)]$$

Simplifying and substituting in for the changes in k and z, this will yield

$$\rho V(k,z) = \max_{c} \; u(c) + \partial_k V(k,z)(e^z f(k) - \delta k - c) - \partial_z V(k,z)(\eta z). \tag{II.26}$$

This is the same equation as the HJB derived earlier in this section (II.24). In the deterministic system, comparing the HJB and the Bellman equation is more simple since we do not need to worry about expectation terms. This is because the deterministic version of this model does not have uncertainty, adding in a continuous-time version of our process in (II.7) will give us a more complicated optimization problem.

II.4.2 A Stochastic Model

Now, we build a stochastic model in discrete and continuous time. Adding in stochasticity will yield more complex models and additional terms in the HJB equation. These stochastic models are more common in literature and are closely related to the Ramsey models used later in this paper.

The Discrete-Time Stochastic Model

This model will be a stochastic version of the discrete-time model defined previously. In this setting, agents will maximize utility according to (II.5), and capital will evolve according to (II.6) with the same Cobb-Douglas production function. The main difference between this model and the previous deterministic model is that z_t evolves according to the following AR(1) process,

$$z_{t+\Delta t} = (1 - \eta \Delta t) z_t + \sigma \epsilon_t \sqrt{\Delta t}, \tag{II.27}$$

where $\epsilon_t \sim N(0,1)$. This model is closely related to the continuous stochastic model outlined later in this section.

Optimization problems in this setting can take several different forms. The current value Hamilton for this problem is,

$$J(k_t, \mu_{t+\Delta t}, c_t, t, t+\Delta t) = u(c_t) + \mu_{t+\Delta t}(e^{z_t} f(k_t) - \delta k_t - c_t) + \gamma_{t+\Delta t}\big(-\eta z_t + \sigma \epsilon_t \sqrt{\Delta t}\big). \tag{II.28}$$

The transversality condition will be the same as in the discrete deterministic model (II.14). Despite the presence of an additional term, the first-order conditions for this system will be the same as the ones from the discrete deterministic model. Also, as the increment of time approaches zero, the discrete Hamiltonian outlined here will be equivalent to the continuous Hamiltonian described in the following section.

We can also write the discrete Bellman equation for this system

$$V(k_t, z_t) = \max_{c_t} \; u(c_t)\Delta t + \beta^{\Delta t}\mathbb{E}[V(z_{t+\Delta t}, k_{t+\Delta t})]. \tag{II.29}$$

This setting will have similar first conditions to the discrete model previously studied.

The Continuous-Time Stochastic Model

One of the key differences between the continuous-time model in this section and the one previously outlined is the process for productivity. Productivity in the continuous-time setting will evolve according to the following Ornstein-Uhlenbeck process, the continuous-time analog of (II.27).

$$dz_t = -\eta z_t dt + \sigma dW_t \tag{II.30}$$

Where dW_t is the increment of the Wiener process.

Equilibrium in the continuous-time setting is given by the following equations. First, equilibrium will depend on the HJB equation (II.31), the continuous-time analog of the Bellman equation. We can first write this equation in a form similar to (II.17).

$$V(k, z) = \max_{c_t} \; u(c_t) + e^{-\rho t}\mathbb{E}[V(k', z')]$$

The expectation term in this model will differ from the expectations in (II.17). This is because the Wiener process in (II.30) is continuous but is nowhere differentiable, making it impossible to treat this expectation like a standard Riemann integral. Using stochastic calculus to solve for this expectation will yield the following HJB equation.

$$\rho V(k,z) = \max_c \; u(c) + \partial_k V(k,z)(e^{z_t} f(k) - \delta k - c_t) + \partial_z V(k,z)(-\eta z_t) + \frac{1}{2}\partial_{zz} V(k,z)\sigma^2 \tag{II.31}$$

Taking the first order condition with respect to consumption for the HJB equation will give us (II.32).

$$u'(c_t) = \partial_k V(k,z)$$

This is analogous to (II.9) in the discrete model or (II.23) in the deterministic continuous model. The term setting the μ_t from the continuous time current value Hamiltonian (II.13) equal to $\partial_k V(k,z)$ we can rewrite the HJB.

$$\rho V(k,z) = \max_c \; H\big(k, c, z, \partial_k V(k,z), \partial_z V(k,z)\big) + \frac{1}{2}\partial_{zz} V(k,z)\sigma^2 \tag{II.32}$$

This equation links the HJBs of our stochastic and non-stochastic continuous time models.

Comparing the Stochastic Models

Furthermore, we can compare the discrete and continuous stochastic models we have outlined thus far. If we take the discrete Bellman in (II.29), we can recast it and make it more similar to (II.32). First, we can take an approximation of $V(k_{t+\Delta t}, z_{t+\Delta t})$, in a method similar to Dorfman (1969).

$$V(k_{t+\Delta t}, z_{t+\Delta t}) = V(k_t, z_t) + \partial_k V(k_t, z_t) dk + \partial_z V(k_t, z_t) dz + \frac{1}{2}\partial_{zz} V(k_t, z_t) dz^2 + \mathcal{O}(\Delta t^{\frac{3}{2}})$$

All other partials and cross partial derivatives will be in the $\mathcal{O}$ term. These terms will all be approximately zero in the limit as $\Delta t \to 0$. Next, we will approximate $\beta^{\Delta t} \approx e^{-\rho \Delta t} \approx (1 - \rho \Delta t)$. Using these two approximations we can rewrite (II.17) as follows.

$$
\begin{aligned}
V(k_t, z_t) &= \max_{c_t}\ u(c_t) + (1 - \rho\Delta t)[V(k_{t+\Delta t}, z_{t+\Delta t})] \\
&= V(k_t, z_t) + \partial_k V(k_t, z_t)dk + \partial_z V(k_t, z_t)dz + \frac{1}{2}\partial_{zz} V(k_t, z_t)dz^2
\end{aligned}
$$

Simplifying and taking the limit as $\Delta t \to 0$ we wil be left with the following equation.

$$
\rho V(k, z) = \max_{c}\ u(c) + \partial_k V(k, z)(e^{z_t} f(k) - \delta k - c) - \partial_z V(k, z)(\eta z_t) + \frac{1}{2}\partial_{zz} V(k, z)\sigma^2
$$

Using this derivation we have gotten the stochastic HJB in equation (II.31). Thus, we have connected our discrete and continuous models in both deterministic and continuous settings.

II.4.3 Results

We have built four closely related models in this section and shown how discrete-time models limit to their continuous-time counterparts. With the correct setup, discrete-time models will be the same in the limit as the continuous-time models. The model comparisons in this section have demonstrated clear connections between discrete and continuous models. These connections are especially evident in the deterministic version of our models; however, with the use of stochastic calculus, they are easily seen.

Furthermore, in this section, we have recast discrete-time models so that they contain the increment of time, Δt. This alone is a contribution to current literature as few economists examine models where $\Delta t = 1$. Within this class of models where Δt is built into the model, one could explore and compare many models with different values for Δt.

II.5 Adaptive Learning Rules

Now that we have developed our modeling framework for this paper, we will move on to examining representative agent exercises in learning. The first group of exercises will focus on an "stylized" learning rule. In this setting, we build models where our agents have a misperception of the true underlying parameters. Then our agents receive information dumps where they get some insight into the correct model parameters. Here agents are trying to update their parameters to make optimal steady-state decisions. Thus, our system is not time-dependent. Agents recalculate the model for many periods, but those periods do not correspond to time periods in our model.

The following section explores three different models. The first examines the stylized learning rule when the unknown model parameter is part of the exogenous stochastic process. Next, the stylized learning rule is applied to a model with mis-specification in an endogenous stochastic process for the evolution of capital stock. Lastly, we modify the model with a stochastic process for productivity and implement a real-time updating rule that utilizes recursive least squares, a more meaningful and realistic approach.

II.5.1 Learning the Process for Productivity

There is a representative agent that makes consumption choices c and has capital stock k. The state of the economy depends on the flow of capital stock. The agent has standard preferences over utility flows based on capital discounted at rate $\rho \geq 0$. This can be written as the following equation:

$$\mathbb{E}_0 \int_{t=0}^{\infty} e^{-\rho t} u(c_t) dt \tag{II.33}$$

Here consumption, $c_t \geq 0$ for all periods. The agent's capital stock will evolve according to the following stochastic process used in Achdou et al. (2014).

$$dk_t = \big(z_t k_t^{\alpha} - \delta k_t - c_t\big)dt \tag{II.34}$$

This is the continuous time analog of the typical equation for the evolution of capital stock. The production function used in this section is Cobb-Douglas, $f(k_t) = k_t^{\alpha}$. Technological progress z_t will evolve according to the following equation

$$d\log(z_t) = -\theta \log(z_t)dt + \sigma dW_t. \tag{II.35}$$

This is a logged version of an Ornstein-Uhlenbeck process, this means that z_t will follow a stationary continuous process that is analogous to an AR(1) process. This can be rewritten in terms of z_t,

$$dz_t = \bigg(-\theta\log(z_t) + \frac{\sigma^2}{2}\bigg)z_t dt + \sigma z_t dW_t. \tag{II.36}$$

In this form we can more clearly see the drift for this process will be, $\big(-\theta\log(z_t) + \frac{\sigma^2}{2}\big)z_t$, and the variance term will be, σz_t.

The utility function used throughout this project will have constant relative risk aversion (CRRA),

$$u(c_t) = \frac{c_t^{1-\gamma} - 1}{1-\gamma}$$

here the CRRA parameter will be γ and $\gamma > 0$.

Stationary Equilibrium

A stationary equilibrium in this setting is given by the following equations. Our HJB for this problem is

$$\rho V(k,z) = \max_c \; u(c) + \partial_k V(k,z) \cdot \big(z_t f(k) - \delta k - c\big) + \\ + \partial_z V(k,z) \cdot \bigg(- \theta \log(z) + \frac{\sigma^2}{2}\bigg) z + \partial_{zz} V(k,z) \cdot \frac{1}{2}\sigma^2 z^2.$$

The derivation of this HJB can be found in the appendix along with a description of the algorithm used to solve this value function problem.

The agents in this simple model hold an incorrect belief about the diffusion process for technological progress. In this setting with exogenous learning, they predict that the diffusion process is given by the equation below,

$$d \log(z_t) = -\theta_g \log(z_t) dt + \sigma_g dW_t$$

There are two parameters that the agent misspecifies in this setting, σ and θ. These misspecifications could be modeled in several different ways, but in this section, we have selected misspecified values of θ and σ that move the drift of the z_t in the same direction. The results from other specifications are shown in the appendix. In the results presented in this section, the agent initially believes that θ is larger than the actual value and that σ is smaller than the true value. Specifically, in period one, $\theta_g = 0.25$ while $\theta = 0.105$ and $\sigma_g^2 = 0.008$ when $\sigma^2 = 0.015$.

To test how a learning process could evolve in this environment, we first introduce an exogenous learning process. Since the process is exogenous, the agents will repeatedly solve the steady-state of the HJB with different amounts of information at each period. In each one of these periods, there is a chance that the agents will have the opportunity to gain more information in the form of noisy observations of the true parameter values. In this model, these noisy observations will be of the form,

$$\tilde{\theta}_i = \theta + \epsilon_{i,\theta}, \quad \epsilon_\theta \sim N(0, 0.1) \tag{II.37}$$

$$\tilde{\sigma}_i^2 = \sigma^2 + \epsilon_{i,\sigma}, \quad \epsilon_\sigma \sim N(0, 0.01) \tag{II.38}$$

The information will be given to an agent based on a draw from a standard Bernoulli distribution, and the agents will update their estimate of both parameters using the following equations

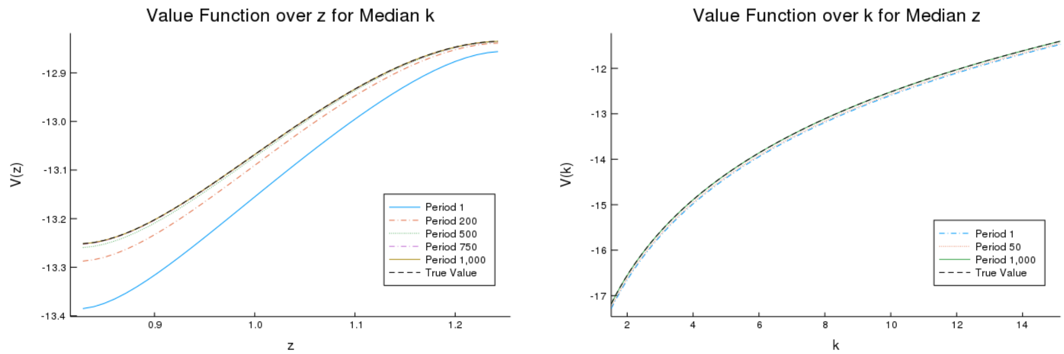
$$\theta_{g,i+1} = \theta_{g,i} + 0.01(\tilde{\theta}_i - \theta_{g,i}),$$
$$\sigma^2_{g,i+1} = \sigma^2_{g,i} + 0.01(\tilde{\sigma}_i^2 - \sigma^2_{g,i}).$$

In this problem θ and σ are the true values of the parameters, and i is an index for the updating period. Parameters are updated using the algorithm above, and then used to calculate the steady-state of our system; this steady-state algorithm is described in the appendix (Achdou et al., 2014).

Productivity Process Results

Below are the convergence results for the stylized learning rule in this setting. The following figure displays the value function over z and k. Looking at the convergence in the value function over z for a median value of k, we can see apparent convergence, here our value function starts flat and develops the correct slope and curvature as our updating procedure continues. However, after $1,000$ periods, we are still some distance from the true value function. Convergence over k for a median value of z is less impressive. In this case, there is appropriate convergence. However, the difference between the misspecification and the true value is much smaller than in the z dimension.

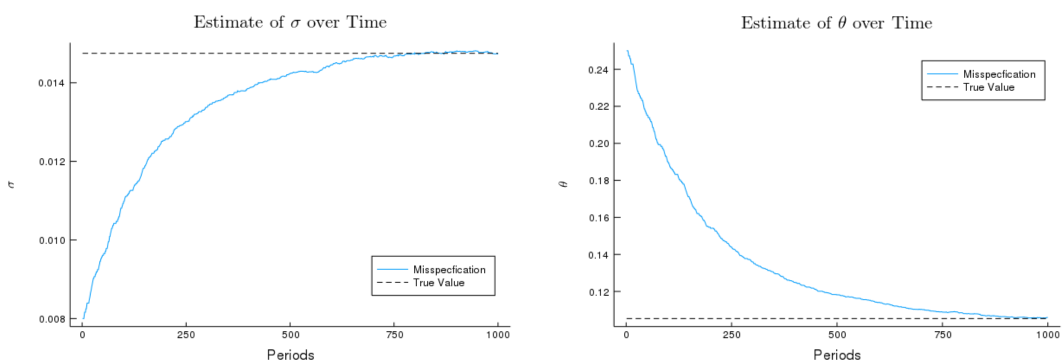
Figure II.1



(a) (b)

The misspecified parameters, θ and σ, converge as we would thought. Below is a graph of the values of σ and θ at each period, including those in which the system does not update.

Figure II.2



(a) (b)

This exercise displays the type of convergence we would have predicted.. Thus, we expect that learning rules would perform in a predictable manner in a stochastic continuous-time setting.

II.5.2 Learning the Process for Capital

After examining the stylized learning rule's impacts on a model with a misspecified exogenous process, we investigate a model with a misspecified endogenous process. In this model, we have a diffusion process that summarizes the evolution of capital stock. Misspecification in this diffusion process impacts optimal savings and, therefore, the optimal consumption choice in the model. Thus, an incorrect specification of this process directly impacts our equilibrium choices. Furthermore, a poor consumption choice directly impacts the drift term in our diffusion process.

In our endogenous process model, there is a representative agent that makes consumption choices c and has capital stock k. The state of the economy depends on the flow of capital stock. The agent has standard preferences over utility flows based on capital discounted at rate $\rho \geq 0$. This can be written as the following equation:

$$\mathbb{E}_0 \int_{t=0}^{\infty} e^{-\rho t} u(c_t) dt$$

Here consumption, $c_t \geq 0$ for all periods. The agent's capital stock will evolve according to the following stochastic process used in Merton (1975). This change has been made so that we can model learning with stochastic process capital. The earlier specification where our stochasticity came from z_t is more common in the literature. In this setting, capital will follow the stochastic process

$$dk_t = \big(f(k_t) - (\delta + n - \sigma^2)k_t - c_t\big)dt + \sigma k_t dW_t.$$

Here n measures the growth of the work force and dW_t is the increment of a Wiener process. In this setting, $f(k_t) - (\delta + \sigma^2)k_t - c_t$ summarizes the drift of capital and σk_t describes the variance.

Stationary Equilibrium

Stationary equilibrium in this setting will be given by several equations. The HJB for this problem will be

$$\rho V(k) = \max_c \; u(c) + V'(k) \cdot \big(f(k) - (\delta + n - \sigma^2)k - c\big) + \frac{1}{2} V''(k) \cdot (\sigma k).$$

The derivation of the HJB can be found in the appendix. This system will be defined on $(\bar{k}, \infty)$ where $\bar{k}$ is the value of capital at which the agent would consume nothing.

The agents in this simple model hold an incorrect belief about the diffusion process for capital stock. In this setting with exogenous learning they predict that the diffusion process is given by equation (II.39).

$$dk_t = \big(f(k_t) - (\delta + n - \sigma_g^2)k_t - c_t\big)dt + \sigma_g k_t dW_t$$

In this model the agent believes that the parameter σ is smaller than it should be, $\sigma_g < \sigma$. Specifically, $\sigma_g = 0.02$ when the true value $\sigma = 0.5$. With this misspecification, the agent believes the drift is larger than it should be *and* the variance is smaller than the true variance of the process. Other misspecifications for this process were examined; these results are in the appendix.

To test how a learning process could evolve in this environment, we first introduce a stylized learning process. Since the information gain is exogenous, the agents will repeatedly solve the steady-state of the HJB with different amounts of information at each period. In each one of these periods, there is a chance that the agents will have a chance to gain more information in the form of a noisy observation of the true parameter estimate. The noisy parameter estimate will take the form,

$$\tilde{\sigma}_i = \sigma + \epsilon_i, \quad \epsilon_i \sim N(0, 0.1). \tag{II.39}$$

The information will be given to the agent based on draw from a standard Bernoulli distribution and the agents will update their estimate of σ_g according to

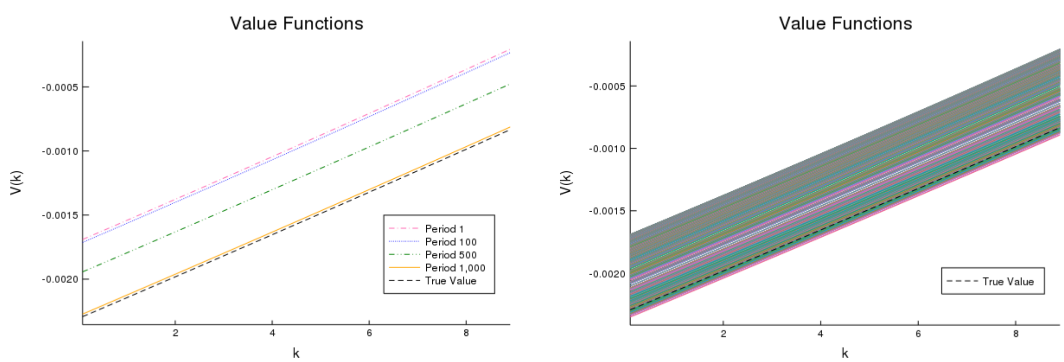
$$\sigma_{g,i+1} = \sigma_{g,i} + 0.01(\tilde{\sigma}_i - \sigma_{g,i}).$$

Here i is the index for the updating period and this updating process will continue for 1,000 periods.

Capital Process Results

Below are several results, the figures on the left show all the output from all 1,000 iterations of the endogenous learning algorithm. Figures on the right display select output from different periods of the iteration.

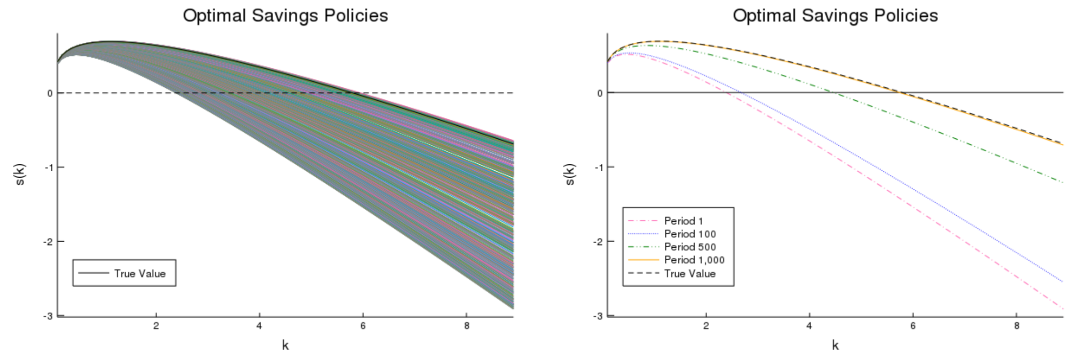
Figure II.3



(a) **(b)**

When the agent uses the learning rule, the value function converges to the true estimate over time. In this setting, convergence is slow, and even after 1,000 periods, the value function is still a small distance from the true value. Convergence is equally slow for some measures, such as savings.

Figure II.4

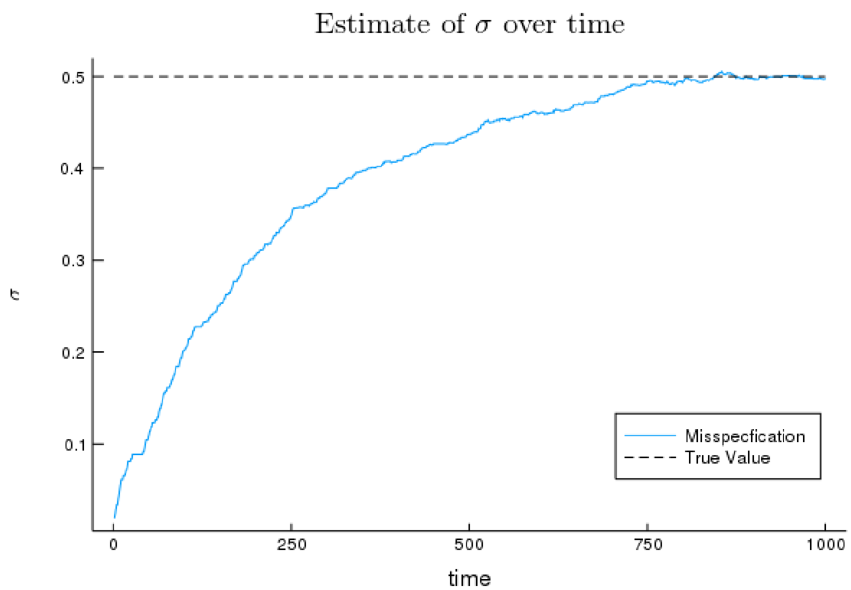


(a) **(b)**

From these figures, we can see that the savings policies appear to converge more quickly to the true policy than the value functions converge to the true steady-state estimates. This is likely due to the fact that optimal savings policies don't depend as strongly on the parameter σ. While σ does impact the calculations of the savings policies, it is only one part of the savings decision. This parameter impacts the value function more directly since it will affect the evolution of the system and the algorithm's choice of implementing a forward difference or backward difference for calculating the derivative of the value function.

Our prediction of σ converges in an expected way. We can see this in the graph below, which verifies that our updating rule works as expected. After 1,000 iterations, the guess for σ is 0.005 away from the true parameter value. This is why our value functions and optimal savings policies have not completely converged to their true values.

Figure II.5



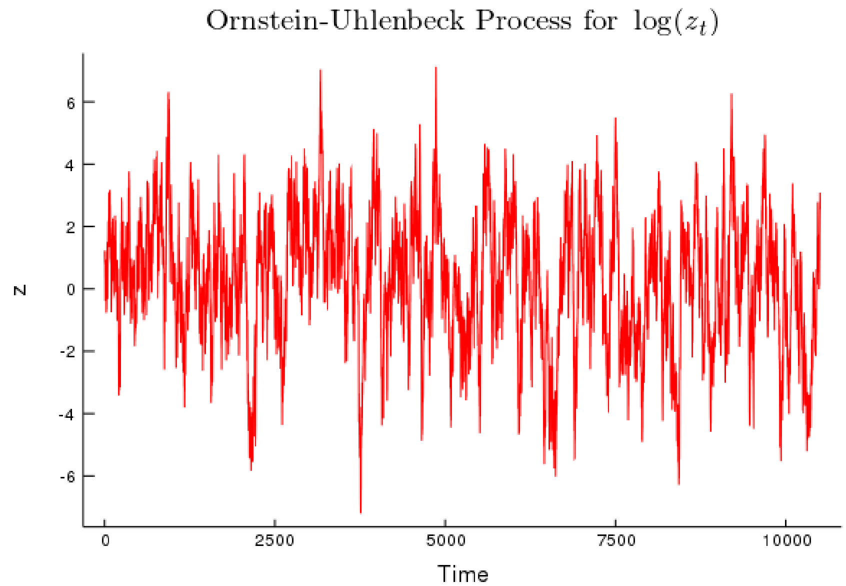
II.5.3 Learning Using Real-Time Updating

In this next section, we will explore a modified model with a stochastic process for productivity. In this model, agents will observe the process over time and update their parameter estimates based on these observations. Agents will maximize utility according to

$$\mathbb{E}_0 \int_{t=0}^{\infty} e^{-\rho t} u(c_t) dt.$$

Here productivity will evolve according to the same process as before, (II.35). Production will still be a standard Cobb-Douglas function used in previous sections. This means that $\log(z_t)$ is evolving according to an Ornstein-Uhlenbeck process, the continuous-time analog of an AR(1) process. Defining the process for z_t this way avoids negative values for z_t. Looking more closely at the $\log(z_t)$ process we have figure II.6.

Figure II.6



This process does have negative values, but the process for z_t will not.

Real Time Updating of Parameter Estimates

The HJB for this stochastic Ramsey model will be

$$\rho V(k,z) = \max_c \; u(c) + \partial_k V(k,z) \cdot \big(zf(k) - \delta k - c\big) + \\ \partial_z V(k,z) \cdot \left(-\theta \log(z) + \frac{\sigma^2}{2z_t} \right) + \partial_{zz} V(k,z) \cdot \frac{1}{2}\sigma^2 z^2,$$

in this setting our parameter for σ will be set equal to one. Setting $\sigma = 1$ will not only simplify our updating problem, but it will also allow for a more intuitive connection between our Ornstein-Uhlenbeck process and an AR(1) process.

In this model, agents believe that the stochastic process for productivity evolves according to

$$d\log(z_t) = -\theta_g \log(z_t)dt + dW_t.$$

Where θ_g is the agent's forecast for the process's parameter θ. Before the agents in this model begin trying to solve their value function problem, they look at the first

100 observations of the process and use ordinary least squares (OLS) to predict a value for θ and a possible constant.

In this setting, the agent can use OLS to predict an initial value for θ_g, since the process for $\log(z_t)$ can be rewritten as a discrete AR(1) process using the Euler-Maruyama method. Applying this method the AR(1) process for $\log(z_t)$ will be,

$$\log(z_{t+\Delta t}) = (1 - \theta_g \Delta t) \log(z_t) + \varepsilon_t \sqrt{\Delta t}$$

for simplicity we will assume that the agent estimates these parameters as if Δt is observable.

Next, they use the finite difference algorithm described in the appendix. They implement this algorithm 10,000 times, each time they observe several additional values of the productivity process. Therefore, in this setting, we should think of the updating periods as independent intervals of time that each contains several observations. Next, using recursive least squares (RLS), the agent updates their parameter estimates. This RLS formula is given by,

$$R_{g,t+1} = R_{g,t} + \gamma_t (xx' \Delta t - R_{g,t})$$
$$\phi_{g,t+1} = \phi_{g,t} + \gamma_t R_{g,t+1}^{-1} \cdot x(y - x' \phi_{g,t}) \Delta t$$

here all variables with a g subscripts represent the agent's forecast x and y are matrices that contain value of x_t and y_t for all points between $t-1$ to t and t to $t+1$ respectively. The number of points in each of these intervals will depend on dt. In the results below, the agent observes 5 points of the process in each updating period. This means that after 100 periods, the agent has 500 new points on which to base their estimates. This has been done in order to maintain continuity in the time dimension. Additionally,

x_t and $\phi_{g,t}$ are defined as

$$x_t = \begin{bmatrix} 1 \\ \log(z_t) \end{bmatrix}, \quad \phi_{g,t} = \begin{bmatrix} c_{g,t} \\ 1 - \theta_{g,t}\Delta t \end{bmatrix},$$

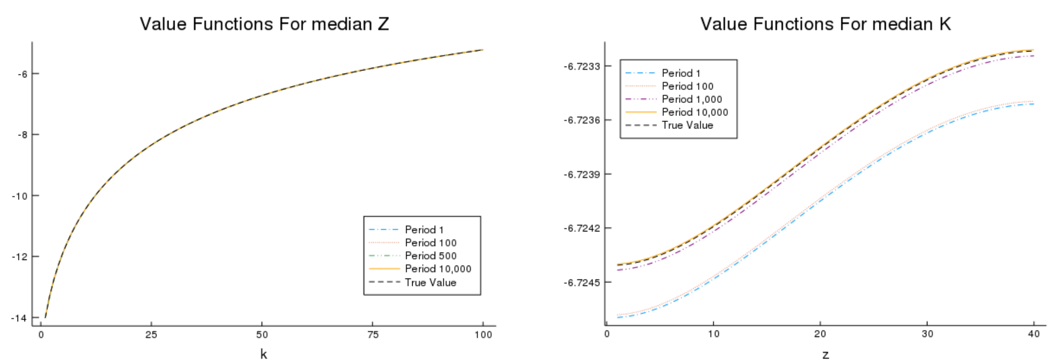
where $c_{g,t}$ is our estimate for a constant in the model. The agent uses this formula to update parameter estimates and then reruns the finite difference algorithm; this is done $10,000$ times.

Real Time Updating Results

Some of the results from the forecasting model resemble the results from previous sections. In this setting, value functions converge quickly in the k dimension and more slowly in the z dimension. This is in line with the results from before and makes sense as the misspecification is for the process that governs z.

First, we will look at results for an algorithm where the gain $\gamma_t = \frac{1}{t}$. This means that the agent discounts the information in each updating period by $\frac{1}{t}$. Here t represents the updating period that the agent is in.

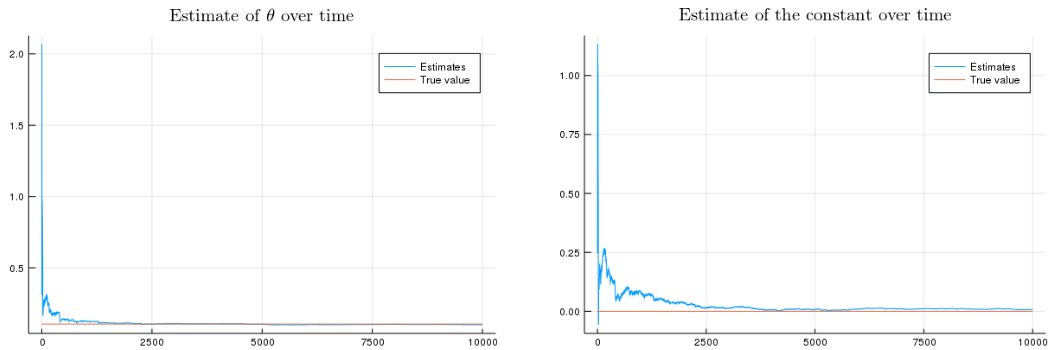
Figure II.7



(a) **(b)**

We can take a closer look at convergence in this setting by examining our parameter estimates over time.

Figure II.8

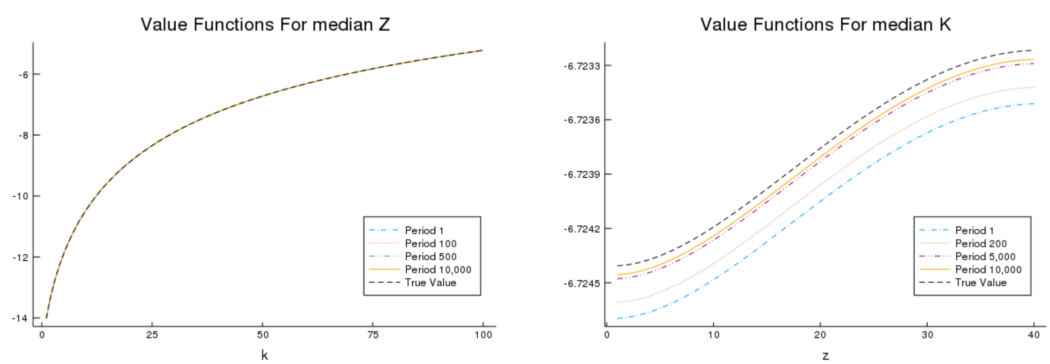


(a) **(b)**

Looking at the results above we can see that convergence in this setting is fast. Despite starting from incorrect parameter values, θ, and the constants are close to their true parameter values after 200 periods.

We can also examine this real-time updating rule with a constant gain. Here we set the gain $\gamma_t = 0.01$ for all time periods. The value functions converge similarly to the decreasing gain case, as seen below.

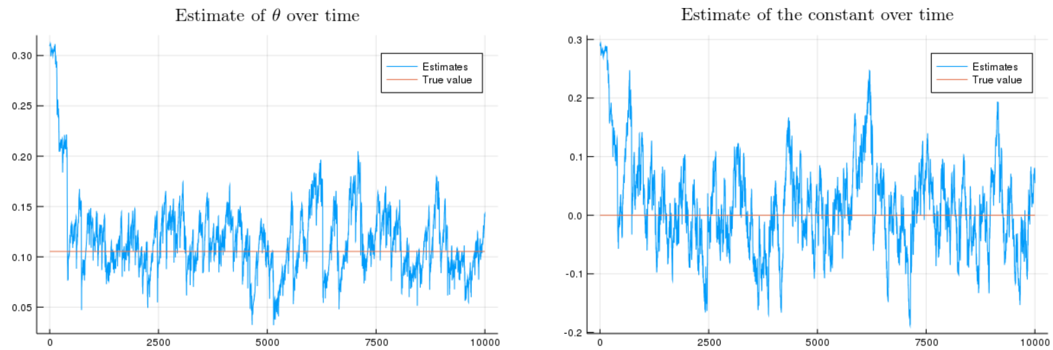
Figure II.9



(a) **(b)**

We can again examine the convergence of θ_g and the estimate for the constant over time.

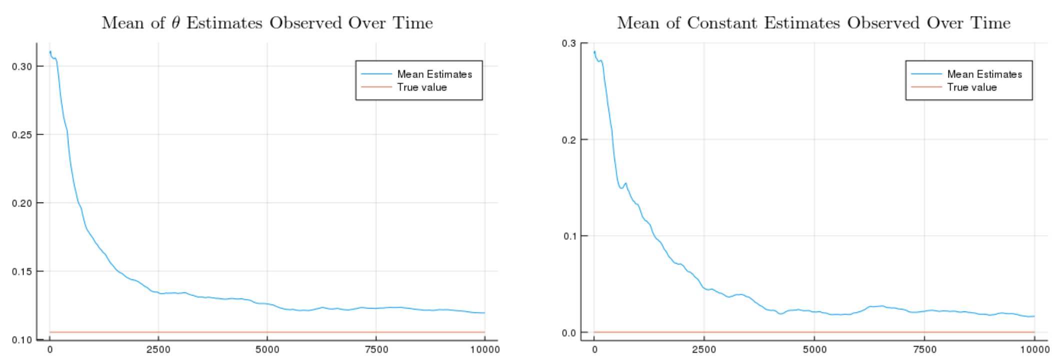
Figure II.10



(a) **(b)**

Since, we are using a constant gain algorithm there is noise in our parameter estimates even after many periods. Constant gain algorithms place equal emphasis on all observed points from the Ornstein-Uhlenbeck process, since this is a noisy process we will see our estimates trend about the correct parameter value instead of directly to the correct value. Due to this, it is helpful to examine the mean estimates of θ and the constant over time.

Figure II.11



(a) **(b)**

Here the mean estimates of θ and the constant are approaching the true parameter values.

II.5.4 Summary

Our exogenous learning rules perform well in the stochastic continuous-time steady state calculations. This is encouraging because it means that we can expect some of the familiar results from discrete-time learning to carry over in our continuous setting. Although the results in this section are not particularly stunning, there are several extensions to this simple learning rule that may yield more interesting results. Looking at this exogenous learning rule in a heterogeneous agent setting may allow for more feedback through the system KF equation, thus yielding less predictable results. A heterogeneous agent model creates this additional feedback through internal pricing frictions that do not exist in our representative agent model.

The performance of the forecasting rule demonstrates that using adaptive learning techniques over intervals of time works well. This method may be beneficial for future work, as it provides a clear link between discrete RLS methods and the continuous-time framework. Despite using different methodologies, it appears that the forecasting rule in section 5.3 and the exogenous learning rule in section 5.1 have similar convergence results, this is an interesting result that may be due to the model similarities in these sections.

II.6 Conclusion

This paper serves a primer on continuous-time modeling and adapting discrete adaptive learning methods to continuous-time. The mathematical results in section 4 link discrete models to continuous-time counterparts. Section 5 contains some basic results for a simple learning method applied to continuous-time models. Using the results of this paper, we can conclude that the continuous-time framework is comparable to discrete-time and that learning algorithms can be adapted and form well in this setting. Future extensions to work could include implementing a continuous-time version of recursive least squares to simple continuous-models and creating a learning algorithm with more feedback in a representative agent model. There remains much to do in order to modify adaptive learning techniques to a continuous-time setting properly.

CHAPTER III
BOUNDEDLY RATIONAL DECISION MAKING IN CONTINUOUS-TIME

III.1 Introduction

The macroeconomics toolkit has significantly expanded in recent years due to increased access to computational power and interdisciplinary research. One promising modeling framework emerging from this development is stochastic continuous-time modeling. Continuous-time models have existed in economics literature for over thirty years, becoming popular during the period Black and Scholes (1973) was first published. During this time economists published papers using the continuous-time framework including, Brock and Mirman (1972), Merton (1969, 1975), and Mirrlees (1971). However, many of these works could only examine specific aspects of models, such as the steady-state distribution of key parameters, as economists did not have techniques for solving the systems of partial differential equations that represent most continuous-time models. Now, with methods drawn from the field of applied mathematics, it has become feasible to solve more continuous-time macroeconomic models.

Continuous-time macroeconomic models have become increasingly popular for two distinct reasons. First, the field of finance has long favored continuous-time modeling, thus building macroeconomic models in continuous-time allows economists to include financial frictions as in Brunnermeier and Sannikov (2014). Second, as we previously mentioned, solutions to many macroeconomic models can now be easily found—because of better computers and new solution methods—and these solutions often include detailed distributional information. Several works that take advantage of this property are Ahn et al. (2018), Achdou et al. (2020), Kaplan et al. (2018) and Gabaix et al. (2016). As this class of models becomes popular, economists must redevelop traditional macroeconomic modeling techniques to create richer models in this continuous-time framework. This paper modifies adaptive learning techniques for use with continuous-time economies.

Currently, the continuous-time macroeconomic literature consists primarily of models that depend on rational expectations. Rational expectations is a standard modeling technique where agents within economics are assumed to understand theoretical models correctly—the agents know the value of all parameters in the model and understand the distribution of any unobserved processes. It is improbable that individuals in the real world have this level of knowledge about the economy. However, individuals can likely perceive the world around them and gradually adjust their expectations based on their observations—adaptive learning takes this approach.

Allowing for adaptive learning, as opposed to rational expectations, in macroeconomic models avoids allowing agents to have unrealistic amounts of information about the system by instead allowing them to gather information on the economy over time slowly. This technique was developed initially in Bray (1982) and been further refined in more recent work Evans and Honkapohja (2001). Adaptive learning is an attractive modeling tool since rational expectations often make to many strict assumptions about agents' knowledge of parameter values and the distribution of parameters.

Additionally, adaptive learning models often converge to a rational expectations equilibrium over time; however, if a model has two rational expectations equilibria, an adaptive learning model may only converge to one—the equilibria learned by these agents would then be stable under adaptive learning whereas the other equilibria would not. Therefore adaptive learning techniques are beneficial when economists want to examine the stability or particular outcomes.

Despite this, rational expectations is a standard model assumption and the emerging continuous-time literature centers on rational expectations models—some continuous-time asset pricing models use Bayesian methods, for instance, Hansen and Sargent (2019a) and Hansen and Sargent (2019b). However, these methods require agents' to have prior belief over the distribution of parameters another strong assumption. We instead concentrate on an adaptive learning technique called shadow-price learning, or SP-learning, outlined in Evans and McGough (2018). Under SP-learning agents view their optimization problem as a two-period problem.

During the first period (today), they use a forecast of their shadow-price to form the best possible choices for today, given those choices' impacts on tomorrow (the second period). Hence this learning mechanism focuses on an agent's ability to generate optimal forecasts and the agent's ability to make optimal decisions with the forecasted information, an issue discussed in (Marimon and Sunder, 1993, 1994; Hommes, 2011). In continuous-time, this problem is very similar; however, instead of having today and tomorrow, the agents examine the trade-off between choices using the change in parameters over time—in other words—the continuous-time version of SP-learning examines derivatives of variables with respect to time.

We develop a tractable setting for SP-learning by building a continuous-time linear-quadratic (LQ) framework. The LQ environment aids the study of adaptive learning techniques due to the linearity of first-order conditions, generality, and certainty equivalence in this framework. In economics, the LQ framework is useful for approximations of complex economies since these models can contain lots of information. There is wide-ranging literature on discrete-time economic optimal linear regulator problems that includes several works on optimal policies such as Benigno and Woodford (2004) and Benigno and Woodford (2006), as well as a wealth of papers on techniques and developing the LQ framework in economics, Kendrick (2005), Amman and Kendrick (1999), and Benigno and Woodford (2012). Because of the richness of this framework and the sparse usage of continuous-time LQ problems in economics, further exploration of this technique is necessary.

Although continuous-time LQ problems are not common in economics, some economists have examined this type of modeling framework. Hansen and Sargent (1991) develops a framework for continuous-time LQ problems. Several chapters of this book examine various models and the identification of parameters in this setting. The LQ framework we build in this paper differs from Hansen and Sargent (1991), as it does not use solution methods based on the Lagrangian. Instead, we take a value function approach. Value function methods are conventional in the discrete-time economics literature, and many continuous-time problems in other fields feature similar solution methods.

We build this framework by outlining a basic discrete LQ problem and then describing a similar continuous-time problem, using a value function approach for both settings. We work through both types of problems, so those familiar with only the discrete case can more easily see the parallels between these two settings. After setting up the LQ problems, we look at solution methods for the resulting algebraic Riccati equations (AREs). Though there are many methods for solving AREs, we concentrate on iterative Newtonian methods, as in Kleinman (1968), as this method better complements the adaptive learning environment in later sections. Also, discrete-time LQ systems commonly use iterative methods (Hansen and Sargent, 2013).

After developing a continuous-time LQ framework, we can then examine continuous-time adaptive learning rules. Before reworking discrete-time adaptive learning rules into continuous-time rules, we need to consider several important items. First, does an agent have "continuous" observations of continuous variables, or do they have discrete observations? If these observations are discrete, are they taken at specific points in time or over intervals, and does the spacing of these points or intervals matter?

We take a simplified approach, drawing from empirical economics and finance literature. Bergstrom (1993), a general survey of continuous-time econometric methods, highlights that continuous-time systems can be measured accurately with exact discrete-time equivalents that take time-interval lengths into account, a conclusion initially drawn from Phillips (1959) and discussed further in Bergstrom (1984). In finance, Kellerhals (2001) uses discrete-time data to measure continuous-time financial systems while carefully implementing exact discrete-time models as in the economics literature. Additional work on this topic includes Aït-Sahalia (2010), which examines the maximum likelihood estimation of continuous model parameters using discrete data points. All of these works find that it is possible to measure continuous-time systems with discrete data.

When using learning algorithms to forecast an agent's perception of the model, we implement the exact discrete-time method since—despite the model parameters evolving continuously—as it is most likely that agents observe the data discretely but at fine intervals. The agents observe data as it becomes available, and they observe all data points. Concentrating on this approach for the agent's sampling of the data allows for more direct tie-ins with typical discrete learning methods. Extensions to this work may include observation intervals that vary from the data generating process's time intervals and data that arrive at random intervals.

The contributions of this work are two-fold. First, to create a modeling framework in which we can develop adaptive learning techniques, we construct a novel continuous-time LQ framework. We outline this framework and discuss it in detail in sections III.2 and III.3. Continuous-time optimal linear regulator problems similar to those outlined in this paper do exist in other disciplines; however, problems outside of economics do not usually include key features such as stochasticity and discounting. Second, we use this new LQ framework to develop continuous-time shadow-price learning in section III.4. Also, we demonstrate parallels between the discrete and continuous models and derive a continuous-time version of recursive least squares (RLS). The bulk of this is done in section III.2.2 and section III.4.

The paper precedes as follows. Section III.2 builds a simple LQ problem without interaction terms or stochasticity. This section also examines iterative solution methods with a univariate test case and convergence of the discrete test case to the continuous one under small time increments. Section III.3 studies a more complicated univariate model with stochasticity as well as this model's solutions, the convergence results with the equivalent discrete-time model. Preliminary results for a simple learning algorithm and the convergence of a discrete-time learning rule to the continuous solution are discussed in section III.4. We evaluate a simple economic model in section III.5; the model used is a simple Robin Crusoe economy as in Evans and McGough (2018). Section III.6 concludes.

III.2 The Optimal Linear Regulator Problem

Before examining a continuous-time LQ problem, we start with a review of a generic deterministic discrete case and focus on defining recursive solutions for this class of problems. In the LQ framework, we examine a value function problem where our objective function is quadratic with respect to our state and choice variables. The state variables are commonly denoted as x_t, here x_t takes the form of an $(n \times 1)$ vector and contains variables that evolve based on past states and past choices. In an economic setting x_t might include variables like capital or productivity. Our choice variables, u_t, are represented by an $(m \times 1)$ vector. These choice variables reflect decisions made by our agent and they can impact future states. A deterministic linear-quadratic problem can be expressed according to the following equations (Ljungqvist and Sargent, 2012),

$$V(x_0) = \max_{u} \; -\mathbb{E}\sum_{t=0}^{\infty}\beta^t\{x_t'Rx_t + u_t'Qu_t\} \tag{III.1}$$

where x_t evolves according to

$$x_{t+1} = Ax_t + Bu_t. \tag{III.2}$$

Here A and R are $(n \times n)$ matrices that summarize how x_t influences future states and our objective function, respectively. For our purposes, x_t always includes a constant; however, the constant is not necessary (Hansen and Sargent, 2013). Similarly, B and Q are $(m \times m)$ matrices that summarize how u_t influences future states and the objective function. Using equations (III.1) and (III.2), we can write the Bellman system as,

$$V(x_t) = \max_{u} \; \{-x_t'Rx_t - u_t'Qu_t + \beta\mathbb{E}V(x_{t+1})\}. \tag{III.3}$$

To solve the Bellman in the LQ framework we use a guess-and-verify approach, positing that $V(x_t) = -x_t'Px_t$, where P is a positive semi-definite matrix (Hansen and Sargent, 2013). Based on the initial posit of the value function's form and the evolution of that state variables we can measure expected future values as, $\mathbb{E}V(x_{t+1}) = -\mathbb{E}(x_{t+1}'Px_{t+1}) = -(Ax_t + Bu_t)'P(Ax_t + Bu_t)$. Substituting these expressions for $V(x_t)$ and $V(x_{t+1})$ into (III.3) yields,

$$-x'Px = \max_{u}\{-x'Rx - u'Qu - \beta(Ax + Bu)'P(Ax + Bu)\}.$$

To create a recursive solution for this system we need to further simplify this expression by eliminating u and x. If we look at the first order condition with respect to u, we get an equation that allows us to expresses choices, u, based solely on model parameters and our states, x.

$$u = -\beta(Q + \beta B'PB)^{-1}(B'PA)x = -Fx$$

using this expression for u, often called a policy function, we can now eliminate u and x from equation (III.3) and write a recursive solution for P using our Riccati equation

$$P_{j+1} = R + \beta A'P_jA - \beta^2 A'P_jB(Q + \beta B'P_jB)^{-1}B'P_jA \tag{III.4}$$

where j denotes the iterations. By implementing this recursive solution method, not only can we find the solution to the discrete-time ARE, but we can start understanding how an agent might update an initial estimate of the value function. Equation (III.4) provides a solution for our value function problem only when certain conditions are met, in this paper we focus on the stability conditions for the continuous-time case; for a treatment of the discrete-time case see Hansen and Sargent (2013), Lewis (1986), or Anderson and Moore (2007).

III.2.1 The Continuous-Time Optimal Linear Regulator

The continuous-time version of this problem is solved with a similar approach. We now examine the continuous-time optimal linear regulator problem using a system similar to—but not the same as—the one in the previous section. The vectors x_t and u_t maintain the same dimensions and continue to represent our state and control variables, respectively. Matrices B, R, and Q also remain the same as before. The matrix A is altered; it maintains its $(n \times n)$ dimensions but not contains different values since we now measure the evolution of our state variables in changes in levels. We assume that A is symmetric to simplify arithmetic for this problem.[1] In the

[1]For a version of this problem that does not assume A is symmetric, please see the appendix.

continuous-time setting, the maximization problem is written as follows,

$$V(x_0) = \max_u \; -\mathbb{E}\int_{t=0}^{\infty} e^{-\rho t}\{x_t' R x_t + u_t' Q u_t\}dt \tag{III.5}$$

where x_t evolves according to,

$$dx_t = Ax_t dt + Bu_t dt \tag{III.6}$$

here our discount factor takes the form of an exponential, $e^{-\rho t}$. Equation (III.6) is a standard expression of a continuous-time deterministic process, in continuous-time the levels of variables over time do not summarize their evolution—instead the changes in a variable describe how it grows over time Dixit (1992).

The continuous value function problem takes a distinct form called the Hamilton-Jacobi-Bellman (HJB). HJBs differ from discrete-time Bellman problems in how they apply discounting and handle expectations; however, they are still closely related to Bellman systems. To demonstrate the close connection between discrete-time and continuous-time value function problems, we show how to derive the HJB from a Bellman equation. First, we write down our problem discretely using the power series expansion of $e^{-\rho\Delta}$, $(1-\rho\Delta)$, as a representation of our discount over a period of time (Dixit, 1992). Here Δ represent the increments of the time periods.

$$V(x_t) = \max_u \{-x_t' R x_t \Delta - u_t' Q u_t \Delta + (1-\rho\Delta)\mathbb{E}[V(x_{t+\Delta})]\}.$$

Expectations in this setting are found by applying Itô's lemma, i.e. by measuring the expected change in the value function $V_x(x)$ weighted by the expected in change in x. Thus, as $\Delta \to 0$ our expectational term $\mathbb{E}[V(x_{t+\Delta})] = V_x(x)$. After simplifying the system and taking the limit as $\Delta \to 0$, the HJB becomes

$$\rho V(x) = \max_u \biggl(-x'Rx - u'Qu + V_x(x)\frac{dx_t}{dt} \biggr) \tag{III.7}$$

Now applying the same guess-and-verify approach as before, we posit that $V(x) = -x'Px$. Using this value function we can rewrite the HJB in (III.7),

$$\begin{aligned} -\rho x'Px &= \max_u \biggl(-x'Rx - u'Qu - 2x'P\frac{dx_t}{dt} \biggr) \\ &= \max_u \{-x'Rx - u'Qu - 2x'P(Ax + Bu)\} \end{aligned} \tag{III.8}$$

Again, we our goal is to create a recursive iterative solution method for finding P. Therefore, we need to eliminate u and x from the system. This is accomplished by taking the first order condition with respect to u,

$$u = -Q^{-1}B'Px = -\tilde{F}x. \tag{III.9}$$

This equation is our policy function for u in the continuous-time system. Note that the policy for u is not the same as the discrete case policy. We should expect the policies for the discrete and continuous-time cases to differ, since expectations[2] and discounting between discrete and continuous-time varies.

Utilizing our policy function we remove u and then x from the HJB equation giving us our Riccati equation,

$$R + 2PA - PBQ^{-1}B'P - \rho P = 0. \tag{III.10}$$

Getting the continuous-time system into a final recursive form can be done with two different methods. Both methods begin with the Lyapunov equation for our optimal

[2]In discrete-time, $\mathbb{E}[V(x_{t+1}] = \mathbb{E}(x_{t+1}Px_{t+1}) = (Ax_t + Bu_t)P(Ax_t + Bu_t)$. While in continuous-time expectations depend on Itô's lemma, $\mathbb{E}[V(x_{t+\Delta})] = V_x(x)\frac{dx_t}{dt} = 2x'P(Ax + Bu)$.

linear regulator problem,

$$2\tilde{A}_i' P_i = -(R + \tilde{F}_i' Q^{-1} \tilde{F}_i).$$

Here, $\tilde{A}_i = A - \frac{1}{2} I\rho - B\tilde{F}_i$, $\tilde{F}_i = Q^{-1} B' P_{i-1}$, and i indexes each iteration. The first method we explore involves subtracting, $2\tilde{A}_i' P_{i-1}$ from both sides giving us,

$$2\tilde{A}_i'(P_i - P_{i-1}) = -2\tilde{A}_i' P_{i-1} - \tilde{F}_i' Q^{-1} \tilde{F}_i + R. \tag{III.11}$$

We can then rewrite this as,

$$P_i = P_{i-1} - (2\tilde{A}_i')^{-1}(2\tilde{A}_i' P_{i-1} - \tilde{F}_i' Q^{-1} \tilde{F}_i + R) \tag{III.12}$$

the main benefit of this method is that it clearly demonstrates how past values P_{i-1} are altered over recursions. Alternatively we can use the second method which is more easily mathematically derived,

$$P_i = -(2\tilde{A}_i')^{-1}(\tilde{F}_i' Q^{-1} \tilde{F}_i + R). \tag{III.13}$$

With these recursive algorithms we can now solve the individual's value function problem. These algorithms also provide insight into how an initial posit of the value function matrix P is updated over time, this system of revising estimates of P will be crucial to the learning dynamics we introduce in later sections. To ensure solutions to (III.12) and (III.13) are asymptotically stable and exist, several conditions must be met (Lewis, 1986; Anderson and Moore, 2007; Evans and McGough, 2018).

LQ.1 The matrix R is symmetric positive semi-definite and thus can be decomposed in $R = DD'$ by rank-decomposition, and the matrix Q is symmetric positive definite.

LQ.2 The matrix pair (A,B) is *stabilizable*—there exists a matrix $\tilde{F}$ such that $A - B\tilde{F}$ is stable, meaning the eigenvalues of $A - B\tilde{F}$ have modulus less than one.

LQ.3 The pair (A,D) is *detectable*—if y is a non-zero eigenvector of A associated with eigenvector μ then $D'y = 0$ only if $|\mu| < 0$. Detectability implies that the feedback control will plausibly stabilize any unstable trajectories.

The conditions outlined in LQ.1-LQ.3 are standard in optimal linear regulator literature and are necessary for stable solutions in both discrete and continuous time-invariant problems. LQ.1 can be interpreted as a condition on the concavity of the system, making sure that the system is bounded above. Additionally, LQ.2 ensures that the value function $V(x)$ does not become infinitely negative by guaranteeing that it is possible to find a policy F that drives the state x to zero.

Theorem 1. *If the conditions outlined in LQ.1-LQ.3 are true, then the continuous-time algebraic Riccati equation has a unique positive semi-definite solution P*

For a proof of theorem 1 see Lewis (1986).

Now that we have examined both discrete and continuous-time linear-quadratic problems and their solutions, we must compare the two and relate them to one another. In the following section, we recast the discrete model so that it depends on discrete-time increments Δ and examining its convergence to the continuous-time problem as $\Delta \to 0$.

III.2.2 Convergence of the Discrete Case to the Continuous Case

The discrete and continuous LQ problems outlined in the previous sections had different Riccati equations because these systems have several differences that cause these equations to evolve dissimilarly. In this section, we rewrite the discrete problem and demonstrate that as time intervals become increasingly small, the discrete Riccati equation solution converges to the continuous solution.

Theorem 2. *The discrete-time system outlined in* (III.1) *and* (III.2) *can be transformed so that its solutions converge to the continuous-time solutions outlined in* (III.5) *and* (III.6).

Proof. To begin, we start with the typical continuous-time system given by equations (III.5) and (III.6). To discretize this system, we rewrite (III.5) as a summation over time periods that increment over integers and an integral over individual time increments, Δ.

$$-\mathbb{E}\sum_{k=0}^{\infty}\int_{t=\Delta k}^{\Delta(k+1)}\{e^{-\rho t}(x_t'Rx_t+u_t'Qu_t)\}dt = -\mathbb{E}\sum_{k=0}^{\infty}\int_{\Delta k}^{\Delta(k+1)}\{e^{-\rho t}f(x_t,u_t,t)\}dt \tag{III.14}$$

For convenience the boundaries on the integral will be changed from $(\Delta k, \Delta(k+1))$ to $(0,\Delta)$, thus $f(x_t,u_t,t)$ must be transformed to $f(x_{t+s},u_{t+s},t+s)$ and integrated over ds. Using a Taylor approximation, the function becomes,

$$\begin{aligned}f(x_{\Delta k+s},u_{\Delta k+s},\Delta k+s) =& x_{\Delta k}'Rx_{\Delta k}+u_{\Delta k}'Qu_{\Delta k}+2x_{\Delta k}'R(x_{\Delta k+s}-x_{\Delta k})+2u_{\Delta k}'Q(u_{\Delta k+s}-u_{\Delta k})\\ &+R(x_{\Delta k+s}-x_{\Delta k})^2+Q(u_{\Delta k+s}-u_{\Delta k})^2.\end{aligned}$$

This function can be further simplified since $x_{\Delta k+s}-x_s = (Ax_{\Delta k}+Bu_{\Delta k})s$ and $u_{\Delta k+s}-u_t=\dot{u}s$ where $\dot{u}$ is a smooth function that summarizes that change in u over an increment of time. Using these substitutions only a few terms in the function will remain—as $s^2\approx 0$ in the continuous-time limit,

$$f(x_{\Delta k+s},u_{\Delta k+s},\Delta k+s) = x_{\Delta k}'Rx_{\Delta k}+u_{\Delta k}'Qu_{\Delta k}+2x_{\Delta k}'R(Ax_{\Delta k}+Bu_{\Delta k})s+2u_{\Delta k}'Q\dot{u}s$$

plugging this into (III.14) yields,

$$-\mathbb{E}\sum_{k=0}^{\infty}\int_{s=0}^{\Delta} e^{-\rho(\Delta k+s)}\{x'_{\Delta k}Rx_{\Delta k}+u'_{\Delta k}Qu_{\Delta k}+2x'_{\Delta k}R(Ax_{\Delta k}+Bu_{\Delta k})s+2u'_{\Delta k}Q\dot{u}s\}ds$$

Focusing on the inter integral,

$$\int_{s=0}^{\Delta} e^{-\rho(\Delta k+s)}\{x'_{\Delta k}Rx_{\Delta k}+u'_{\Delta k}Qu_{\Delta k}+2x_{\Delta k}R(Ax_{\Delta k}+Bu_{\Delta k})s+2u'_{\Delta k}Q\dot{u}s\}ds$$
$$=-\frac{1}{\rho}e^{-\rho\Delta k}[e^{-\rho\Delta}-1](x'_{\Delta k}Rx_{\Delta k}+u'_{\Delta k}Qu_{\Delta k}).$$

Plugging this result[3] back into the main summation term and replacing k with t while setting $\hat{x}_t = x_\Delta$, $\hat{u}_t = u_\Delta$, and $\hat{\rho} = \rho\Delta$ yields,

$$-\mathbb{E}\sum_{t=0}^{\infty}\frac{1}{\hat{\rho}}e^{-\hat{\rho}t}[1-e^{-\hat{\rho}}](\hat{x}'_tR\hat{x}_t+\hat{u}'_tQ\hat{u}_t)\Delta \tag{III.15}$$

to get this into the typical discrete LQ format, as in (III.1), β, R, and Q must be appropriately transformed. The discount factor β becomes $\beta(\Delta) = e^{-\hat{\rho}}$, R is now $R(\Delta) = \frac{1}{\rho}(1-e^{-\hat{\rho}})R$, and $Q(\Delta) = \frac{1}{\rho}(1-e^{-\hat{\rho}})Q$.

Lastly, the equation for the evolution of the state variables must be transformed by applying the Euler-Maruyama method to equation (III.6) yielding,

$$x_{\Delta(t+1)} = (I + A\Delta)x_{\Delta t} + B\Delta u_{\Delta t} \tag{III.16}$$

where I is an $(n \times n)$ identity matrix. Thus the transformed coefficients are $A(\Delta) = (I + A\Delta)$ and $B(\Delta) = B\Delta$. □

We have now built a discrete version of the model that now takes increments of time Δ into account. As $\Delta \to 0$, this system becomes our continuous-time version

[3]The term $\int_0^\Delta e^{-\rho(\Delta k+s)}\{2x_{\Delta k}R(Ax_{\Delta k}+Bu_{\Delta k})s+2u'_{\Delta k}Q\dot{u}s\}ds$ goes to zero after implementing integration by parts and then using the power series expansion of $e^{-\rho(\Delta)}$, $(1-\rho\Delta)$.

of the model, which will have a slightly different numeric solution than the discrete version of the model due to continuous-time discounting methods and constantly evolving states. We turn to demonstrate that, numerically, the discrete version of the model that utilizes time periods Δ does converge to the continuous-time solutions as Δ becomes increasingly small.

A Numerical Illustration

Now that we have shown all of the necessary variable transformations, we can examine the convergence of the transformed discrete-time system to the continuous-time system. As shown in figure III.1 after decreasing Δ from 1.0 to 0.001 the transformed discrete-time system converges to the same solution as the continuous-time system.

Figure III.1

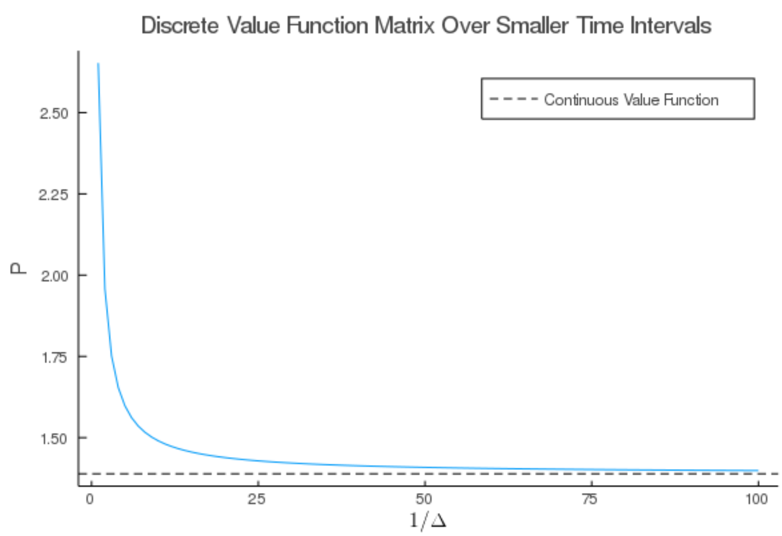


Figure III.1 displays the unique tie between the discrete-time LQ solutions and the continuous-time version. Thus far, our analysis has focused on deterministic LQ problems. To applied adaptive learning techniques properly, we need to add stochasticity to our problem; this is our main focus in the following section.

III.3 A Model with Stochasticity

Thus far, the models explored were deterministic, meaning that our states evolved according to a known process that did not involve randomness. We now recast our state variables so that they evolve according to a stochastic process. Thus specific state values are impacted by a random normally distributed shock each period. Furthermore we include interaction terms between x and u, these are summarized by the $(n \times m)$ matrix W. Our stochastic optimal linear regulator problem takes the following form,

$$V(x_0) = \max_u \; -\mathbb{E}\int_{t=0}^{\infty} e^{-\rho t}\{x_t' R x_t + u_t' Q u_t + 2x_t' W u_t\}dt. \tag{III.17}$$

Where the state of the system, x_t, evolves according to a continuous-time stochastic process

$$dx_t = Ax_t dt + Bu_t dt + CdZ_t \tag{III.18}$$

here dZ_t is the increment of the Wiener process[4] and A is again assumed to be symmetric. As before x_t is an $(n \times 1)$ vector of state variables and u_t is a $(m \times 1)$ vector of control variables.

The HJB for this problem can be found using the same approach implemented in section III.2. In the stochastic case our HJB is,

$$\rho V(x) = \max_u \; -x'Rx - u'Qu - 2x'Wu + \frac{1}{dt}\mathbb{E}\left(V_x(x)dx_t + \frac{1}{2}V_{xx}(x)(dx_t)^2\right). \tag{III.19}$$

Note that unlike the HJB in (III.8), this HJB equation has an additional term that comes from applying Itô's lemma to the stochastic process for dx_t (Dixit, 1992). This additional term changes the proposed $V(x)$ (Hansen and Sargent, 2013). When using

[4] The increment of the Wiener process can be approximated as $dZ_t = \varepsilon_t\sqrt{dt}$ where $\varepsilon_t \sim \mathrm{N}(0,1)$. Thus, $\mathbb{E}[dZ_t] = 0$ and $\mathbb{E}[(dZ_t)^2] = dt$

the guess-and-verify method for the stochastic problem our initial posit is,

$$V(x) = -x'Px - \xi$$

where P is a positive semi-definite matrix and ξ is a constant that does not depend on our state or control variables. Substituting the proposed value function for $V(x)$ in (III.19) yields,

$$\rho x'Px + \rho\xi = \max_{u}\{x'Rx + u'Qu + 2x'Wu + 2x'P(Ax + Bu) + P(CC')\}. \quad \text{(III.20)}$$

As before, our goal is to create a recursive solution method for finding the matrix P. To accomplish this, we must eliminate u and x from equation (III.20). The policy function for u is almost the same as before; however, it now includes the interaction terms in W,

$$u = -(Q')^{-1}(W + PB)'x = -Fx.$$

Using this policy function to remove u and x from (III.20) produces,

$$\begin{aligned} \rho P &= R + F'QF - 2WF + 2A'P - 2PBF \\ \rho\xi &= PCC'. \end{aligned}$$

Our continuous-time system of equations is similar to the discrete stochastic case discussed in Hansen and Sargent (2013) in that the matrix C that multiplies the Wiener process dZ_t does not impact P; instead, it affects ρ. The matrix P is independent of the stochasticity in this problem, a beneficial outcome since we can now solve the more complex stochastic problem by finding the solution to the more simple deterministic version. Steady-state solutions for this type of system can be found recursively like in section III.2.1 using the following recursive scheme,

$$P_i = -(2\tilde{A}_i')^{-1}(\tilde{F}_i'Q^{-1}\tilde{F}_i + R - 2W\tilde{F}_i) \tag{III.21}$$

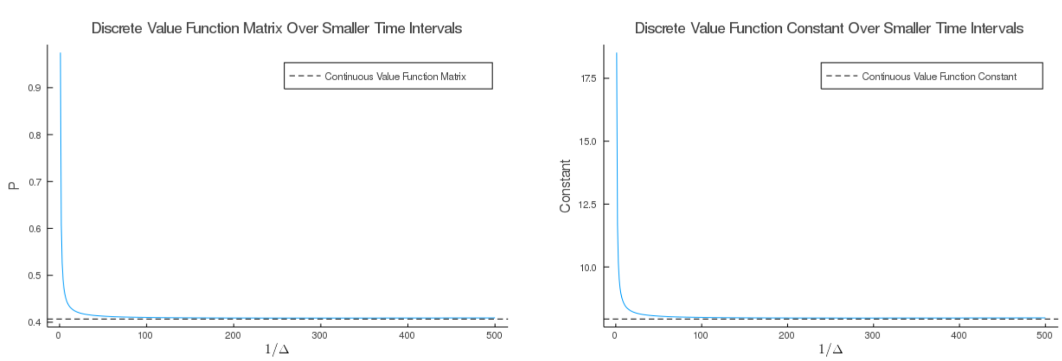
$$\xi_i = \rho^{-1}\text{trace}(P_{i-1}CC'), \tag{III.22}$$

where $\tilde{A}_i = (A - B\tilde{F}_i - .5\rho)$ and $\tilde{F}_i = (Q')^{-1}(W + P_{i-1}B)'$. These equations will provide a positive semi-definite solution for the matrix P and a solution for the constant ξ as long as the conditions outlined in LQ.1-LQ.3 hold.

III.3.1 Convergence in the Complex Case

Before moving on, it is worth noting that under transformations similar to those in section III.2.2 a discrete version of this system converges to the continuous model we described in the previous section. The necessary transformations are β becomes $\beta(\Delta) = e^{-\hat{\rho}}$, R is now $R(\Delta) = \frac{1}{\rho}(1 - e^{-\hat{\rho}})R$, $Q(\Delta) = \frac{1}{\rho}(1 - e^{-\hat{\rho}})Q$, $W(\Delta) = \frac{1}{\rho}(1 - e^{-\hat{\rho}})W$, $A(\Delta) = (I + A\Delta)$ and $B(\Delta) = B\Delta$, and $C(\Delta) = C\sqrt{\Delta}$ where $\hat{\rho} = \rho\Delta$. To test convergence for this model, we used the same univariate case as in section III.2 with $W = 1.0$ and $C = 1.0$. The rate of convergence for the matrix P in the complex case is similar to the rate of convergence in the simple case considered earlier.

Figure III.2



(a) **(b)**

In figure III.2a the transformed discrete system's value function, or P matrix, converges to the continuous system's value function, and in figure III.2b the value function's constant term ξ converges to the continuous-time system's constant. Figure III.2 demonstrates that even in the more complicated model, the discrete-time system's solutions can limit to the continuous-time solutions.

III.4 Learning Dynamics

The primary goal of this work is to capture an agent's behavior under bounded rationality in a basic continuous-time setting. To fulfill this objective, we need a continuous-time updating rule to describe how agents take-in information and adaptive learning dynamics define how agents' choices and forecasts impact their future observations. In this section, we outline both a continuous-time updating rule and adaptive learning dynamics. Modeling the agent's ability to update forecasts is done using a continuous-time analog to recursive least squares (RLS) (Lewis et al., 2007), we derive our version of continuous-time RLS using the continuous-time Kalman filter. Adaptive learning dynamics used in this paper follow shadow-price learning techniques from Evans and McGough (2018).

III.4.1 Continuous-Time Recursive Least Squares

Recursive algorithms are used to estimate parameters and states in a wide variety of models. However, as stated in Ljung and Söderström (1983), "There is only one recursive identification method. It contains some design variables to be chosen by the user." While this statement is not valid for all models, we can use the same general algorithm for a wide variety of linear regression and state-space models. This relationship between recursive algorithms has been often noted for the Kalman filter and LQ problems as in Ljungqvist and Sargent (2012); however, we explore this relationship with two other standard recursive algorithms in economics—recursive least squares and the Kalman filter.

Connections between the Kalman filter and RLS are well understood in economics research and have been cited in Branch and Evans (2006) and Sargent (1999). Exploiting the likeness of these two algorithms, we derive the RLS algorithm from the Kalman filter. We first explore the connection between the Kalman filter and RLS in discrete-time to better understand their linkage before examining both these systems in continuous-time. Direct connections between discrete and continuous-time recursive algorithms have been noted in Ljung (1977) and Lewis et al. (2007). These relationships prove helpful when we turn to examine continuous-time algorithms.

The recursive least squares algorithm used in adaptive learning literature is not more conceptually complex than weighted least squares. We derive RLS as a simple weighted least squares algorithm. The main difference between RLS and weight least squares is that our RLS algorithm is designed to update and account for new information each period. Instead of having our agent re-run their estimation scheme each period RLS has built-in updating methods that take into account the agent's original estimation and the updated information. As with most least squares methods our problem begins with a simple linear regression,

$$y_t = \theta' x_t + e_t$$

where $e_t \sim \mathrm{N}(0,1)$. Here our agent can estimate the model parameters, θ, by choosing an estimator that minimizes the model's errors. We select a generic least-squares method that allows for the possibility of weights,

$$V_N(\theta) = \frac{1}{N} \sum_{t=1}^{N} \alpha_t [y_t - \theta' x_t]^2 \tag{III.23}$$

where N is the number of observations in the data and α_t is a weighting vector that may depend on time. The weighting vector α_t is indirectly related to the gain sequence in adaptive learning literature, it is one of two parameters that determines whether or not our system has constant gain (all data points are evenly weighted) or decreasing gain (as more data is accumulated the data are gradually given less weight). The optimal method of setting α_t depends on the variance of e_t. For simplicity we set $\alpha_t = 1$, i.e. we assume $e_t \sim N(0, 1)$. Implementing this least-squares method we can derive a common form of RLS that uses decreasing gain,

$$\begin{aligned}\hat{\theta}_t &= \hat{\theta}_{t-1} + \frac{1}{t}\mathcal{R}_t^{-1}x_t[y_t - \hat{\theta}'_{t-1}x_t], \\ \mathcal{R}_t &= \mathcal{R}_{t-1} + \frac{1}{t}[x_t x'_t - \mathcal{R}_{t-1}]\end{aligned}$$

This recursive algorithm estimates coefficients based on observations and estimates of the second moment $\mathcal{R}_t$. The avoid the matrix inversion in the system above we can instead use $\mathcal{P}_t = (t \cdot R_t)^{-1}$.

$$\begin{aligned}\mathcal{P}_t &= [\mathcal{P}_{t-1}^{-1} + x_t x'_t]^{-1} \\ &= \mathcal{P}_{t-1} - \frac{\mathcal{P}_{t-1}x_t x'_t \mathcal{P}_{t-1}}{1 + x'_t \mathcal{P}_{t-1}x_t}.\end{aligned}$$

Thus our system will become,

$$\hat{\theta}_t = \hat{\theta}_{t-1} + L_t[y_t - \hat{\theta}'_t x_t], \tag{III.24}$$

$$L_t = \frac{\mathcal{P}_{t-1}x_t}{1 + x'_t \mathcal{P}_{t-1}x_t}, \tag{III.25}$$

$$\mathcal{P}_t = \mathcal{P}_{t-1} - \frac{\mathcal{P}_{t-1}x_t x'_t \mathcal{P}_{t-1}}{1 + x'_t \mathcal{P}_{t-1}x_t}. \tag{III.26}$$

The method of deriving RLS examined thus far is not ideal. While it does intuitively connect the least-squares framework to our agent's recursive updating scheme, it is distant from the behavioral perspective from which we want to examine forecasting. Re-approaching this algorithm from a filtering viewpoint allows us to separate two key parts of developing forecasts: one, how do individuals observe information, and two, how do they use this information to develop forecasts.

Now, we re-derive recursive least squares using Kalman filter, a recursive algorithm used for tracking unobservable states. Suppose we have the following state-space model,

$$\text{Transition Equation: } x_{t+1} = A_t x_t + w_t, \tag{III.27}$$

$$\text{Measurement Equation: } y_t = \theta_t' x_t + e_t \tag{III.28}$$

Where $\{w_t\} \sim N(0, R_t)$ and $\{e_t\} \sim N(0, r_t)$, r_t and R_t may be defined as constants. The Kalman filter is a valuable method for examining our agent's behavior and beliefs via parameters r_t and R_t. As previously mentioned, our agent can weigh observations one of two ways, they can either give more weight to the first few observations and decrease weights to data points observed at later dates or give all observations equal weighting. For the first method, decreasing gain, we select $R_t = 0$ and $r_t = 1$, meaning the agent believes there is no noise behind the process for x_t and the errors for equation (III.28) are from an i.i.d white noise process. A constant gain system requires $R_t = \frac{\gamma}{1-\gamma}\mathcal{P}_t$ and $r_t = (1 - \gamma)$ where $\gamma \in (0, 1)$ is our "constant." Under constant gain, the agent believes their forecasts to be subject to some error and that the states they are trying to predict, x_t, are stochastic. Under constant gain, learning forecasts oscillate about equilibrium and are expected to respond to shocks in all periods equally.

A general Kalman Filter, that allows for the possibility of either type of gain, can be described by the following equations

$$x_{t+1} = A_t x_t + K_t[y_t - \theta_t' x_t], \tag{III.29}$$

$$K_t = \frac{A_t \mathcal{P}_t \theta_t'}{r_t + \theta_t \mathcal{P}_t \theta_t'}, \tag{III.30}$$

$$\mathcal{P}_{t+1} = A_t \mathcal{P}_t A_t' + R_t - A_t \mathcal{P}_t \theta_t' [r_t + \theta_t \mathcal{P}_t \theta_t']^{-1} \theta_t \mathcal{P}_t \theta_t'. \tag{III.31}$$

Note the parallels between this and the system in (III.26). We can imagine these as the same algorithm. If we re-imagine the state-space model we used to derive the recursive least squares algorithm as,

$$\text{Transition Equation: } \theta_{t+1} = \theta_t + \nu_t \tag{III.32}$$

$$\text{Measurement Equation: } y_t = \theta_t' x_t + e_t \tag{III.33}$$

where $\nu_t \sim N(0, R_t)$ and $e_t \sim N(0, r_t)$, the Kalman filter will become our RLS system from (III.24)-(III.26) when $R_t = 0$ and $r_t = 1$. This particular RLS system will have a decreasing gain. The transition equation in (III.32) is now the transition equation for model parameters θ_t instead of data x_t, as shown in (III.32) the parameters in this setting are constant over time. The measurement equation in (III.33) is essentially the same as the measurement equation in (III.28); however, now there is uncertainty about the parameters θ_t as apposed to the data x_t. The decreasing gain Kalman filter for the system described in (III.32) and (III.33) yields,

$$\hat{\theta}_{t+1} = \hat{\theta}_t + K_t[y_t - x_t'\theta_t], \tag{III.34}$$

$$K_t = \frac{\mathcal{P}_t x_t}{1 + x_t'\mathcal{P}_t x_t}, \tag{III.35}$$

$$\mathcal{P}_{t+1} = \mathcal{P}_t - \mathcal{P}_t x_t[1 + x_t'\mathcal{P}_t x_t]^{-1} x_t'\mathcal{P}_t. \tag{III.36}$$

As we can see this is equivalent to the system in (III.24)-(III.26) with $K_t = L_t$, decreasing gain values for r_t and R_t, and some modified timing conventions. Thus, we can see the connection between the Kalman filter and RLS.

Constant gain RLS, which we did not derive earlier, is more easily defined from the Kalman filter since it requires the agent to believe they are estimating a stochastic state. For constant gain RLS our Kalman filter derivation method yields

$$\begin{aligned}
\hat{\theta}_{t+1} &= \hat{\theta}_t + K_t[y_t - x_t'\theta_t], \\
K_t &= \frac{\mathcal{P}_t x_t}{(1-\gamma) + x_t'\mathcal{P}_t x_t}, \\
\mathcal{P}_{t+1} &= \frac{1}{1-\gamma}\mathcal{P}_t - \mathcal{P}_t x_t[(1-\gamma) + x_t'\mathcal{P}_t x_t]^{-1} x_t'\mathcal{P}_t.
\end{aligned}$$

While this RLS algorithm is very similar to the decreasing gain case, it will not generate the same results, although both may converge to the same equilibrium.

For our purposes, we need a version of RLS that assumes measurements are continuous functions of time. While not widely used, the continuous-time Kalman filter is commonly implemented in some engineering and applied mathematics fields. A continuous-time analog of RLS called the continuous-time recursive least squares filter does exist; however, as discussed, we would like an approach that allows us to derive algorithms for decreasing *and* constant gain.

In this section, we derive the continuous-time Kalman filter using methods from Lewis et al. (2007). First, we modify (III.27)-(III.28) to depend on increments of time (Δ) and recast our state transition matrix,

$$\begin{aligned}
x_{t+1} &= (I + A_t\Delta)x_t + w_t \\
y_t &= \theta_t x_t + e_t
\end{aligned}$$

here the covariance matrix for $\{w_t\}$ is $R_t\Delta$ and the covariance matrix for $\{e_t\}$ is r_t/Δ. First, we examine what happens to the Kalman gain in (III.30) as $\Delta \to 0$. Our Kalman gain becomes,

$$K_t = \frac{(I + A_t\Delta)\mathcal{P}_t\theta_t'}{(r_t/\Delta) + \theta_t P_t \theta_t'}$$

or

$$\frac{1}{\Delta}K_t = \frac{(I + A_t\Delta)\mathcal{P}_t\theta_t'}{r_t + \theta_t\mathcal{P}_t\theta_t'\Delta}$$

and taking the limit of this as $\Delta \to 0$ yields,

$$\lim_{\Delta \to 0} \frac{1}{\Delta} K_t = \mathcal{P}_t \theta_t' r_t^{-1}. \tag{III.37}$$

This is our continuous-time Kalman gain. Next, we examine (III.31),

$$\mathcal{P}_{t+\Delta} = (I + A_t\Delta)\mathcal{P}_t(I + A_t\Delta)' + R_t\Delta - (I + A_t\Delta)\mathcal{P}_t\theta_t'[(r_t/\Delta) + \theta_t\mathcal{P}_t\theta_t']^{-1}\theta_t\mathcal{P}_t(I + A_t\Delta)'.$$

Eliminating and terms and dividing by Δ yields,

$$\frac{1}{\Delta}\mathcal{P}_{t+\Delta} = \frac{1}{\Delta}\mathcal{P}_t + A_t\mathcal{P}_t + \mathcal{P}_tA_t' + R_t - (I + A_t\Delta)\mathcal{P}_t\theta_t'[r_t + \theta_t\mathcal{P}_t\theta_t'\Delta]^{-1}\theta_t\mathcal{P}_t(I + A_t\Delta)'.$$

Then, taking the limit as $\Delta \to 0$,

$$\lim_{\Delta \to 0} \frac{1}{\Delta}\big(\mathcal{P}_{t+\Delta} - \mathcal{P}_t\big) = \frac{d\mathcal{P}_t}{dt} = A_t\mathcal{P}_t + \mathcal{P}_tA_t' + R_t - \mathcal{P}_t\theta_t'[r_t]^{-1}\theta_t\mathcal{P}_t$$

this equation is our continuous-time covariance updating equation.

Last, we derive the estimate updating equation. In this setting (III.29) will become,

$$\hat{x}_{t+\Delta} = (I + A_t\Delta)\hat{x}_t + K_t[y_t - \theta_t\hat{x}_t]$$

diving this by Δ will give us,

$$\frac{1}{\Delta}(\hat{x}_{t+\Delta} - \hat{x}_t) = A_t\hat{z}_t + \frac{K_t}{\Delta}[y_t - \theta_t\hat{x}_t].$$

Now, we can take the limit as $\Delta \to 0$ and use equation (III.37),

$$\frac{d\hat{x}_t}{dt} = A_t\hat{x}_t + \mathcal{P}_t\theta_t'r_t^{-1}[y_t - \theta_t'\hat{x}_t]$$

this will be our systems estimate updating equation.

Thus our continuous-time Kalman filter for the system can be described by the following equations.

$$\begin{aligned}\frac{d\mathcal{P}_t}{dt} &= \theta_t'\mathcal{P}_t + \mathcal{P}_t A_t' + R_t - \mathcal{P}_t\theta_t' r_t^{-1}\theta_t'\mathcal{P}_t \\ K &= \mathcal{P}\theta_t' r_t^{-1} \\ \frac{d\hat{x}_t}{dt} &= A_t\hat{x}_t + K[y_t - \theta_t'\hat{x}_t]\end{aligned}$$

Our corresponding transition and measurement equations for this filter are

$$\begin{aligned}\frac{dx_t}{dt} &= Ax_t + w_t \\ y_t &= \theta' x_t + v_t\end{aligned}$$

Here w_t and v_t are error terms and $w \sim N(0, R_t)$ and $v \sim N(0, r_t)$.

Since we have established how to derive the continuous-time Kalman filter and the Kalman filter's connections to recursive least squares, we exploit these connections to create a continuous version of RLS. We can rewrite our state-space model in (III.32)-(III.33) as,

$$\begin{aligned}\frac{d\theta_t}{dt} &= \nu_t \\ y_t &= \theta_t' x_t + e_t\end{aligned}$$

Now, $\nu_t \sim N(0, R_t)$ and variance for e_t is r_t, our RLS system will be

$$\frac{d\mathcal{P}_t}{dt} = -\mathcal{P}_t x_t' r_t^{-1} x_t \mathcal{P}_t + R_t \tag{III.38}$$

$$K = \mathcal{P}_t x_t' r_t^{-1} \tag{III.39}$$

$$\frac{d\hat{\theta}_t}{dt} = K[y_t - \hat{\theta}_t' x_t]. \tag{III.40}$$

The constant gain version of the algorithm used later on this paper can be represented as

$$\begin{aligned}
\frac{d\mathcal{P}_t}{dt} &= \frac{1}{1-\gamma}(-\mathcal{P}_t x_t' x_t \mathcal{P}_t + \gamma \mathcal{P}_t) \\
K &= \frac{1}{1-\gamma}\mathcal{P}_t x_t' \\
\frac{d\hat{\theta}_t}{dt} &= K[y_t - \hat{\theta}_t' x_t].
\end{aligned}$$

For the decreasing gain version of the algorithm simply use $r_t = 1$ and $R_t = 0$ in equations (III.38)-(III.40). A more direct definition of continuous-time RLS that stems from discrete RLS is included in the appendix.

We now have a continuous-time updating rule that will govern how our agents take in information and updater their estimates of key model parameters. To complete our adaptive learning model, we need one more item, adaptive learning dynamics, that reflect how an agent's estimates and perceptions impact the economy and the future states the agent observes. Our approach to modeling these dynamics is shadow-price learning. In the following section, we expand upon what shadow-price learning means and define our adaptive learning model.

III.4.2 Adaptive Learning Rules in Continuous-Time

Before we can start analyzing and implementing adaptive learning in basic macroeconomic models, we need to develop our actual learning dynamics. Thus far, we have created a rich environment that will facilitate learning and an updating algorithm that will allow our agent to utilize the information they obtain; however, we still need to connect the agent's forecasts and choices to their impact on the agent's perceptions of the future. First, we review the continuous-time LQ problem described in section III.3. Our agent seeks to maximize the value of a quadratic objective function by selecting a sequence of optimal choices u_t.

$$V(x_0) = \max_u \; -\mathbb{E}\int_{t=0}^{\infty} e^{-\rho t}\{x_t' R x_t + u_t' Q u_t + 2x_t' W u_t\}dt.$$

Where the state of the system, x_t, evolves according to a continuous-time stochastic process

$$dx_t = Ax_t dt + Bu_t dt + CdZ_t$$

In the adaptive learning model agents gain information about a data generating process for x_t and use this information to update their predictions of parameters and optimal choices in turn their decisions will impact the states that they observe. Agent's modify their optimal choices in this setting using shadow-price parameters, in economics these parameters function as future prices for objects that may not traditionally have prices—i.e. capital or investment. The agent will update their estimates of the system's transition matrix, A, and the shadow price parameters which we will denote as H $(H = -2P)$ using the continuous analog of recursive least squares. Estimated values of A and H will then impact the agent's policy decision and the shadow prices they observe next period. Our use of H impacts our policy function, changing it to

$$u = -\frac{1}{2}(Q')^{-1}(2W - HB)'x = -F^{SP}(H, B)x. \qquad \text{(III.41)}$$

To differentiate between this version of the continuous-time policy function and the version define earlier we label the shadow-price version, F^{SP}, and specify that it is a function of shadow-prices, H.

Before delving into the adaptive learning model and the specifics of our adaptive learning dynamics, we preview the interactions between our LQ model, continuous-time RLS, and the adaptive learning methodology, and we develop later in this section. Below is our adaptive learning algorithm that determines our model outcomes, please note that we have formatted the learning algorithm in terms of changes in levels as opposed to time derivatives to more closely fit the formatting of stochastic processes in macroeconomic literature.

$$\begin{aligned}
dx_t &= Ax_t dt + Bu_t dt + CdZ_t \\
d\mathcal{P}_t &= \frac{1}{1-\gamma_t}(\gamma_t \mathcal{P}_t - \mathcal{P}_t x_t x_t' \mathcal{P}_t)dt \\
dH_t' &= \frac{1}{1-\gamma_t}\mathcal{P}_t x_t (\lambda_t - H_t x_t)' dt \\
dA_t' &= \frac{1}{1-\gamma_t}\mathcal{P}_t x_t (dx_t - Bu_t dt - A_t x_t dt)' \\
u_t &= -F^{SP}(H_t, B)x_t = -\frac{1}{2}(Q')^{-1}(2W - H_t' B)' x_t \\
\lambda_t &= T^{SP}(H_t, A_t, B)x_t \\
\gamma_t &= \kappa(t+N)^{-\nu}.
\end{aligned} \tag{III.42}$$

Here $\mathcal{P}_t$ is the covariance matrix for x_t and γ_t is the gain sequence that measures the response of estimates to forecast errors. For simplicity, we assume that the gain is constant—$\nu = 0$ and $\kappa = 0.01$. Additionally, $F^{SP}(H_t, B)$ is the policy under shadow price learning and $T^{SP}(H_t, A_t, B)$ is the T-map—a link between agent's perception and the actual system, we will describe both functions as well as the link between H and P in the following section.

Continuous-time Policies and the T-map

Previously we focused on solving optimal linear regulator problems using recursive methods, meaning that given an approximation to the solution $V_k(x)$ a new approximation $V_{k+1}(x)$ can be obtained. Note that here k is not a measure of time but an index representing iterations. This approach conveniently lends itself to learning algorithms as the first approximation $V_k(x)$ can be viewed as the perceived value function, using $V_k(x)$ one can then compute the induced value function $V_{k+1}(x)$. For the following derivation we utilize P to represent the perceived value function matrix and $V^P(x)$ to represent the induced value function that results from the agent's initial estimation of P

$$\rho V^P(x) = \max_u \{-x'Rx - u'Qu - 2x'Wu - 2x'P(Ax + Bu) - P(CC')\} \qquad \text{(III.43)}$$

Agent's need to select u in order to solve the value function problem in (III.43). The unique optimal control decision for perceptions P is given by,

$$u = -F(P)x = -(Q')^{-1}(W + PB)'x.$$

We first examine the deterministic case for this problem, where $C = 0$. Recall from earlier that the solution for our deterministic problem yields the solution for the stochastic case. In this setting, the induced value function is defined as $V^P(x) = -x'T(P)x$, $T(P)$ is a function that maps the agent's perception or initial estimate of P to the resulting updated value function $V^P(x)$. The mapping function $T(P)$, more formally called the T-map, for this problem is

$$T(P) = (2\tilde{A}')^{-1}(F'Q^{-1}F + R - 2WF) \qquad \text{(III.44)}$$

here $\tilde{A} = A - \frac{1}{2}I\rho - BF$. Note the right-hand-side of $T(P)$ is similar to the Riccati equation (III.21). Based on it's similarity to the Riccati equation and the underlying iterative solution methods we can conclude that the fixed point of this T-map identifies the solution to the agent's optimal control problem. In the stochastic case where $C \neq 0$ our T-map is given by,

$$T^{\varepsilon}(\tilde{P}) = \tilde{P} - \rho^{-1}\text{trace}(\tilde{P}CC')$$

where $T(\tilde{P}) = \tilde{P}$. Optimally decision making in this setting is determined by the fixed pint of $T^{\varepsilon}(P_{\varepsilon}^*)$, P_{ε}^*. The fixed point of the stochastic system is directly related to the solution for the deterministic case, P^*, by the following equation

$$P_{\varepsilon}^* = P^* - \rho^{-1}\text{trace}(P^* C C').$$

Thus, the solution to the deterministic problem yields the solution to the stochastic problem. This aligns with the rational expectations problems discussed earlier in this work.

III.4.3 Shadow Price Learning

The learning dynamics outlined thus far have made strong assumptions about the agent's knowledge of the value function. In the problem outlined in (III.43), an agent understands that the value function is quadratic in x, knows how to solve for the matrix P by iterating on the Riccati equation, and knows parameters A and B. In the following section, we modify these assumptions. As opposed to assuming the agent knows A and B, we assume that the agent does know B, indicating they understand how their control decisions impact the state. However, the agent is not assumed to know the parameters of the state-contingent transition dynamics. Meaning they must estimate A. Additionally, the agent in the following problem is not assumed to know how to solve the programming problem. Instead, they use a simple forecasting model to estimate the value of the state tomorrow—the shadow price of the state. The agent then uses this estimate and an estimate of the transition equation to determine the best control response for today.

We now outline a learning framework in which the agent forms expectations of future shadow prices. The boundedly optimal behavior modeled in this section is shadowing price learning or SP-learning (Evans and McGough, 2018). Under SP-learning, the agent believes that the shadow price, λ, is linear in x. Thus they can forecast the shadow price as,

$$\lambda_t = Hx_t + \mu_t \tag{III.45}$$

where μ_t is some error term. Using this perceived law of motion (PLM), we can create a T-map for the agent's perceptions using our HJB equation. we first estimate that,

$$\mathbb{E}[V_x(x)] = \lambda^e = Hx$$

where λ^e is the updated estimate of λ. Plugging this into the HJB for our stochastic LQ problem we get,

$$\rho V(x) = \max_u \{-x'Rx - u'Qu - 2x'Wu + (Hx)'(Ax + Bu) + \frac{1}{2}(H'CC')\}.$$

In this new setting our policy function will depend on H and B,

$$u = -\frac{1}{2}(Q^{-1})'(2W - H'B)'x = -F^{SP}(H, B)x \tag{III.46}$$

this is the same policy function mentioned earlier in this section. Next, to get the mapping from the PLM to the actual law of motion (ALM) we use the envelope condition,

$$\rho V_x(x) = \rho\lambda^e = -2x'R - 2u'W + 2x'A'H + u'B'H. \tag{III.47}$$

We can rewrite (III.47) to clearly define expected shadow-prices λ^e,

$$\lambda^e = \rho^{-1}\{-2x'R - 2u'W + 2x'A'H + u'B'H\}$$

or

$$\begin{aligned}\lambda^e &= T^{SP}(H, A, B)x \\ &= \rho^{-1}\big(-2R + 2H'A - (H'B - 2W)F^{SP}(H, B)\big)x.\end{aligned} \tag{III.48}$$

This is the T-map we use to model the agent's boundedly rational behavior. The fixed points of this mapping correspond to equilibrium values of shadow-prices, H. In terms of the shadow-price learning algorithm, the T-map provides feedback for the agent's choices and allows them to update to more optimal choices as they gain experience and information.

Stability of shadow-price learning dynamics

The stability of the T-map is essential to learning dynamics. If the fixed points of our T-map are not stable, it is possible that our agent will not reach an equilibrium or that they will deviate from the desired rational expectations equilibrium. The following conjecture provides conditions that should insure T-map stability in both the discrete and continuous-time cases,

Conjecture 1. *Assuming that LQ.1-LQ.3 hold, there exists an $n \times n$ solution P^* to the Riccati equation given any symmetric positive definite initial matrix P_0 (Evans and McGough, 2018). Therefore $T^m(P_0) \to P^*$ as $m \to \infty$ and*

1. *$T(P^*) = P^*$—the solution P^* is a fixed point of the T-map.*
2. *$DT_v(\text{vec}(P^*))$ is stable—has eigenvalues less than one.*
3. *P^* is the unique fixed point of T among the class of $n \times n$, symmetric positive semi-definite matrices.*

Thus, if the Riccati equation has asymptotically stable solutions, the T-map for the system is stable. Conjecture 1 is proved to be true in the discrete-time setting in Evans and McGough (2018). Based on numerical and analytical results, it is conjectured to hold true in the continuous-time setting as well.

Next, we will examine the solutions and stability of the learning system using $A = 0.0$, $R = Q = B = 1.0$, $W = C = 0$, and $\rho = 0.05$. Our T-map (III.48) can be rewritten as a function of H using these values. This function $T(H)$ has two fixed points. One at $\tilde{H} \approx 2.880$ and a second solution at $H^* \approx -2.778$. This second solution is consistent with the solutions for P from both the continuous iterative scheme and the icare function since $H = -2P$. Directly comparing the solution for P from the iterative schemes and $-\frac{1}{2}H^*$ there is a difference of 2.220×10^{-16}.

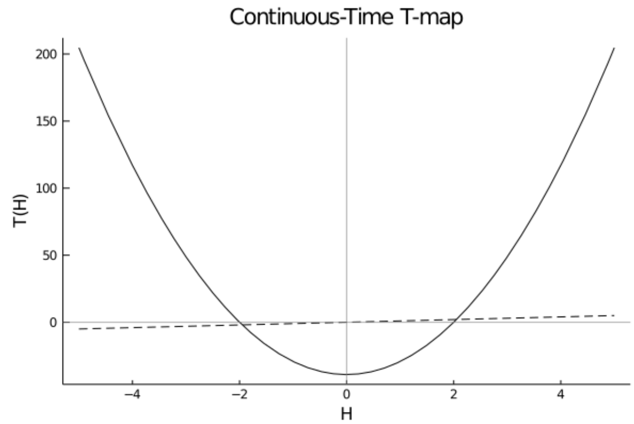
The solution H^* is stable, based on stability conditions for the Riccati and the T-map. For the continuous-time Riccati equation to be stable, $A + BF^{SP}(H^*, B)$ must have eigenvalues with real parts less than one, and our T-map must satisfy the condition that $DT^{SP}(H^*, A_t, B)$ has eigenvalues with real parts less than one. H^* meets these stability conditions as,

$$A + BF^{SP}(H^*, B) = -0.975, \quad DT^{SP}(H^*, A, B) = -39.012.$$

However, the unstable solution $\tilde{H}$ does not meet these criteria as

$$A + BF^{SP}(\tilde{H}, B) = 1.025, \quad DT^{SP}(\tilde{H}, A, B) = 41.012.$$

Figure III.3
T-map

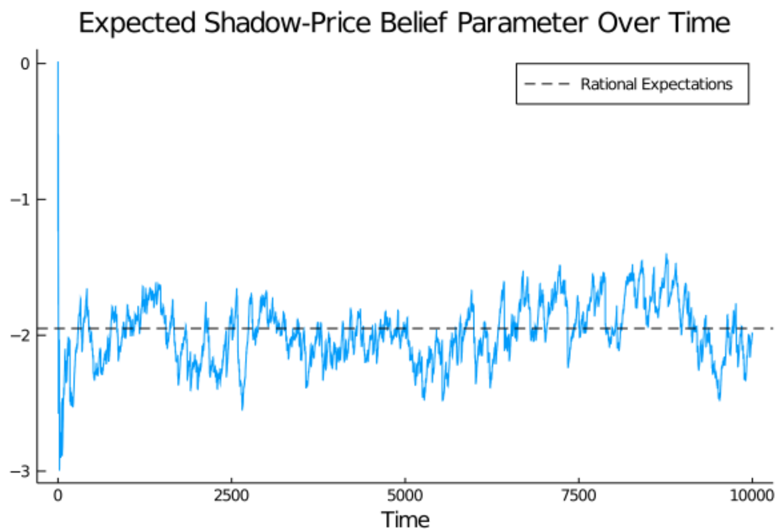


Now that we have examined adaptive learning dynamics and derived a continuous-time version of RLS, we can examine the convergence of the learning algorithm outlined in (III.42).

Continuous-Time Learning Results

Using the learning dynamics we have already developed, we examine how the agent in the univariate learning model estimates the shadow-price parameter H. As shown below in figure III.4, when using an approximation of the length of the time increment ($dt \approx 0.01$) and constant gain ($\gamma = 0.01$) the method outlined in (III.42) will converge to the rational expectations equilibrium.

Figure III.4
Univariate Continuous-Time SP-Learning

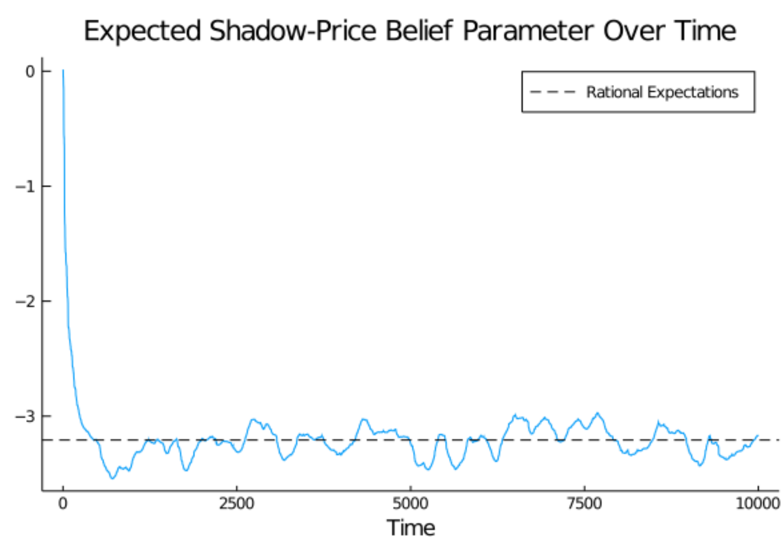


Though our result is simple, it is encouraging that our adaptive learning displays convergence to rational expectations equilibrium. One would expect and hope that a simple stochastic model would display the behavior exhibited in III.4. For better reference, we compare our results to a discrete-time system where an agent's bounded rational behavior can be modeled by the following equations (Evans and McGough, 2018),

$$
\begin{aligned}
x_t &= Ax_{t-1} + Bu_{t-1}dt + C\varepsilon_t \\
\mathcal{R}_t &= \mathcal{R}_{t-1} + \gamma_t(x_t x_t' - \mathcal{R}_{t-1}) \\
H_t' &= H_{t-1} + \gamma_t \mathcal{R}_{t-1}^{-1} x_{t-1}(\lambda_{t-1} - H_{t-1}x_{t-1})' \\
A_t' &= A_{t-1} + \gamma_t \mathcal{R}_{t-1}^{-1} x_{t-1}(x_t - Bu_{t-1} - A_{t-1}x_{t-1})' \\
u_t &= -F^{SPD}(H_t, A_t, B)x_t \\
&= (2Q - \beta B'HB)^{-1}(\beta B'HA_t - 2W')x_t \\
\lambda_t &= T^{SPD}(H_t, A_t, B)x_t \\
&= \Big(-2R - 2WF^{SPD}(H_t, A_t, B) + \beta A_t'H\big(A_t + BF^{SPD}(H_t, A_t, B)\big)\Big)x_t \\
\gamma_t &= \kappa(t+N)^{-\nu}.
\end{aligned}
\tag{III.49}
$$

Here $(t \cdot \mathcal{R}_t)^{-1} = \mathcal{P}_t$, this does impact the model besides requiring the use of matrix inversion. Using the equivalent parameter values from our univariate continuous-time case this system has comparable convergence results,

Figure III.5



both models convergence to rational expectations equilibria; however, by construction the continuous-time case updates more frequently and experiences more rapid changes over time. In an economic model, this could lead to second-moments varying between continuous and discrete models, depending on the setting and how the models are calibrated. This could lead to better-fitting second moments from our continuous-time model.

Our results thus far are encouraging. In the simplest case, our continuous-time learning algorithm converges to rational expectations equilibrium and performs comparably to a well-tested discrete-time algorithm. In advance of moving to a more complicated and economically motivated LQ problem, we exploit our simple univariate test case to inspect whether our discrete learning algorithm can converge to continuous-time rational expectation equilibrium.

Convergence in the Context of Learning

In section III.3, we showed that our discrete-time system's solution for the value function matrix P can converge to the continuous-time solution under certain transformations. Similarly, we will show that the discrete learning rule outlined in equation (III.49) with $\gamma_t = (0.01)\Delta$ converges to the continuous-time expected shadow price parameter when Δ is sufficiently small.

Figure III.6 shows how the discrete learning rule responds under the transformations in section III.3 with select values of Δ.[5] In figure III.6 the modified discrete learning rule gradually gets closer to the continuous-time rational expectations solution as Δ gets increasingly small.

[5] The learning iterations in figure III.6 have been re-scaled for easier representation. Each iteration is equivalent to a discrete time period $t = 1, 2, \ldots 10,000$ that contains Δ^{-1} observations. Meaning that for $\Delta = 1/4$ this graph is displaying the results from $40,000$ iterations

Figure III.6
Univariate Discrete-Time SP-Learning

III.5 A Robinson Crusoe Economy

Now that we have developed the modeling framework for continuous-time LQ problems and examined basic learning rules in this setting, we can examine a slightly more involved model.

We begin with a simple Robinson Crusoe economy as in, Evans and McGough (2018). The representative agent in this model maximizes a quadratic objective function that depends on their consumption decisions, preferences, and resources

$$\max_{c_t} -\mathbb{E}\int_{t=0}^{\infty} e^{-\rho t}\big((c_t - b_t)^2 + \phi l_t^2\big) \tag{III.50}$$

where the economy is subject to,

$$\begin{aligned} y_t &= A_1 s_t \\ ds_t &= (y_t - c_t - s_t)dt + dZ_t \\ s_t &= l_t \\ b_t &= b^* \end{aligned} \tag{III.51}$$

as before dZ_t is the increment of the Wiener process. The model we have outlined in (III.50) and (III.51) is a version of the discrete Robinson Crusoe (RC) model used in Evans and McGough (2018).

The agent in our setting has only one consumable good, fruit, and only one means of production, growing trees from seeds of the fruit. Thus, income y_t can be thought of as fruit, and consumption c_t as consumption of that fruit and its seeds. The production of the fruit comes from planting seeds, s_t. The change in the number of seeds over time depends on growing conditions—represented by the increment of the Wiener process dZ_t—and leftovers from consumption. In this one-person economy, work is burdensome and causes disutility for the worker ($\phi > 0$). Lastly, b_t is a bliss point represented by the constant b^*.

We have simplified this model to maintain similarities between a continuous and discrete case. For instance, we do not have a possible time lag in production—in this model, young trees and old trees produce the same amount. Additionally, the bliss point is non-stochastic, and there are no productivity shocks; instead, production only depends on the availability of seeds.

To analyze this model in our LQ environment, we need to transform this system into the format from (III.17) and (III.18). We set our state vector as $x_t = (1, s_t)'$ and the vector of control variables to be $u_t = c_t$. Our states evolve according to,

$$dx_t = Ax_t dt + Bu_t dt + CdZ_t.$$

The matrices A, B, and C are defined as

$$A = \begin{bmatrix} 0 & 0 \\ 0 & A_1 - 1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ -1 \end{bmatrix}, \quad C = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

The final objects necessary for transforming our RC model into an easy to analyze LQ problem are the R, Q, and W matrices. Given the already quadratic nature of the agent's objective function we can find via inspection that,

$$R = \begin{bmatrix} b^{*2} & 0 \\ 0 & \phi \end{bmatrix}, \quad Q = 1.0, \quad W = \begin{bmatrix} -b^* \\ 0 \end{bmatrix}.$$

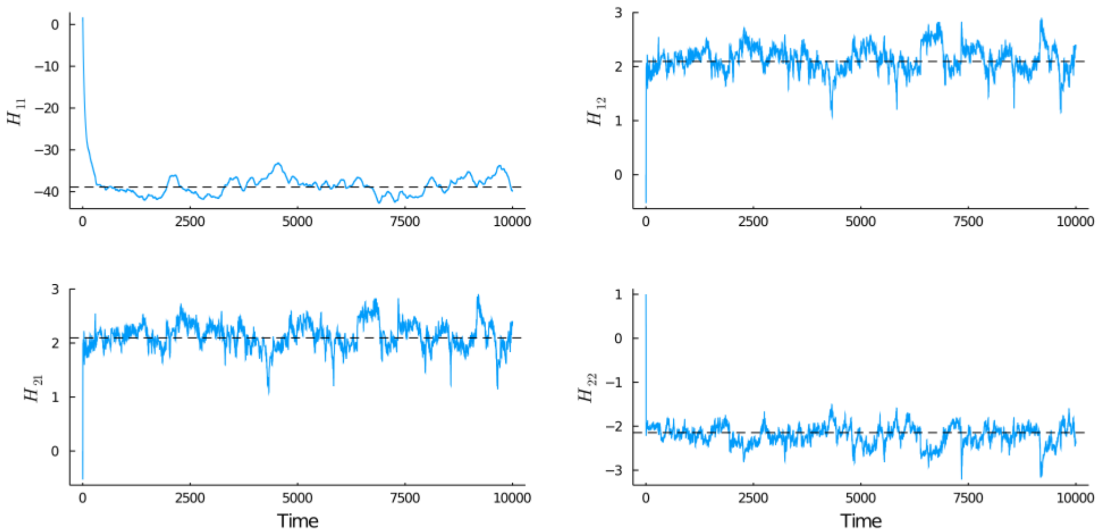
Using these matrices and parameter values we can now calculate the rational expectations equilibrium for this system and implement our adaptive learning model.

III.5.1 Learning in the Continuous RC Model

In this setting, it is likely that our agent does not know the parameters of the production function, or the value of an additional tree tomorrow. However, the agent can use the system outlined in (III.42) to forecast these unknown values. As the agent gains more information they can update their parameter estimates using (III.42); the matrices B, C, R, Q, and W; and initial values for A_t, H_t, $\mathcal{P}_t$, and λ_t.

Under the learning rules described in (III.42), the agent learns parameters for the matrix H and the matrix A (in this case, both are a 2×2 matrix). To generate data for this model, an approximation for dt was necessary. For the following results, we used $dt \approx \Delta = 1/100$. Additionally, we used a constant gain term where $\kappa = 0.01$, and $\nu = 0$.

Figure III.7
Expected Shadow-Price Parameters, The Continuous Case

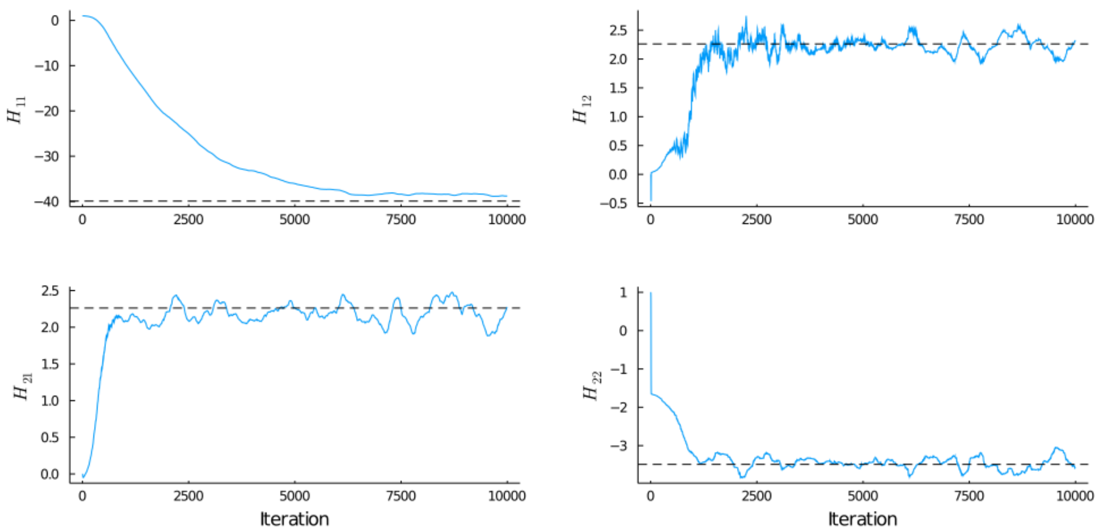


As shown in figure III.7, an agent with boundedly rational behavior modeled by (III.42) will be able to generate an accurate estimate of the steady-state shadow price parameters. In figure III.7 we plot $10,000$ discrete time periods, in the continuous-time case with $dt = 0.01$ this means we have included $1,000,000$ learning iterations or updates of the shadow-price parameters.

III.5.2 Learning in the Discrete RC Model

A discrete version of this model with, as outlined in Evans and McGough (2018), converges similarly with the same constant gain parameter. Below we have plotted $10,000$ discrete periods to make it easy to compare the convergence of this system to the continuous system in section III.5.1.

Figure III.8
Expected Shadow-Price Parameters, The Discrete Case



The agent in our discrete-time shadow-price learning model displays similar behavior to our continuous-time agent. Both agents converge to rational expectations equilibrium, and both estimations oscillate about their respective equilibrium. One interesting outcome in this model is that the continuous-time shadow-price value corresponding to our constant converges more quickly in our continuous-time model. Additionally, analysis on continuous-time learning techniques may provide insight into why this occurs; however, there is no intuitive explanation.

III.6 Conclusion

As continuous-time macroeconomic literature expands, it is necessary to modify and re-evaluate discrete modeling techniques in this framework. Adaptive learning mechanisms are particularly essential to modify as they relax the strong assumption of rational expectations—the belief that agents forecast optimally. The shadow-price learning technique outlined in the previous sections goes beyond easing rational expectations, as it also examines the optimality of an agent's decisions as they optimize according to their forecasts. Since agents in this setting use available information to forecast their shadow-prices and then make control decisions based on their forecasts (Evans and McGough, 2018).

It was beneficial to develop a continuous-time linear-quadratic framework for macroeconomic models to implement shadow price learning in a continuous-time environment efficiently. Other disciplines, such as engineering, frequently use continuous-time linear quadratic methods (Vrabie et al., 2009; Lewis, 1986). However, very few examples of economic models in this framework exist (Hansen and Sargent, 1991). After building this general framework, we examined convergence results and equilibrium stability in this class of models.

Within this continuous-time LQ framework, we implemented a continuous analog to recursive least squares and analyzed a continuous-time T-map. This system yielded results that suggest an agent can learn to optimize decisions in both simple univariate cases and with more sophisticated models. This paper serves as a basic template for continuous-time shadow-price learning. Our main result is simply that shadow-price learning can be done in continuous-time through the framework we have defined.

The basic tools provided in this chapter lay the groundwork for many potential applications and explorations of adaptive learning methods in continuous-time macroeconomic models. Our RLS algorithm creates a baseline for updating rules in a continuous-time setting, which is necessary for nearly all learning models. The continuous-time LQ framework implemented in this chapter is restrictive since most macroeconomic models are not linear-quadratic. However, our LQ setting provides a basis from which a well-sized class of models can be explored and allows us to begin exploring the underlying dynamics and differences that occur in continuous-time settings.

CHAPTER IV

BOUNDED RATIONALITY IN MACROECONOMIC MODELS: A CONTINUOUS-TIME APPROACH

IV.1 Introduction

Macroeconomic models often assume that both changes in the economy and agent's decisions occur at quarterly intervals, since data are most often available at that frequency. This approximation is bound to generate a loss of precision; since individuals make decisions about their employment, consumption, and investment at higher frequencies—arguably every day—despite less frequent economic data on these measures. In the economy, factors such as productivity and technology also change at a high frequency since computing power and innovations change rapidly. While discrete-time models provide useful insight into the economy, parameters that evolve quarterly and quarterly decision making can produce less accurate measures of volatility in real business cycle models (Aadland, 2001). One way to easily capture these high-frequency changes is continuous-time modeling, which assumes that the economic system is constantly evolving. Thus, building economic models in continuous-time provides an attractive alternative to discrete-time modeling.

We use a continuous-time real business cycle model combined with continuous-time adaptive learning dynamics, which allow our agent to improve their forecasts of key parameters and their optimal choices at high frequencies, to show that volatility of parameter estimates can be improved using high-frequency information. We demonstrate that the continuous-time model has less volatile parameter estimates as the agent's forecasts near rational expectations equilibrium (REE). Additionally, when examining these models near REE, the second moments of the continuous-time model came closer to matching relative moments from economic data than the model's discrete version.

We chose the continuous-time setting not just because of its ability to include high-frequency data and dynamics easily but also because it has a few key advantages over discrete-time and has recently gained popularity in macroeconomics. This class of models had been studied and examined in the past; however, continuous-time models did not gain the same prevalence as discrete-time modeling in economics due to their more complicated solution methods (Merton, 1971; Mirman, 1973; Mirrlees, 1971). With increased computing power and more interdisciplinary research from applied mathematics and engineering, continuous-time macroeconomic models can now be easily solved even if they are involved. There are several different solution methods for these models ranging from viscosity solutions as in Kaplan et al. (2018), Achdou et al. (2020), and Ahn et al. (2018) to martingale methods as in Brunnermeier and Sannikov (2014).

As this literature enters the mainstream, it is necessary to modify macroeconomic modeling tools standard in discrete-time research. Thus far, continuous-time macroeconomic literature has focused almost exclusively on rational expectations, a modeling assumption wherein the agent knows key model parameters' values and distributions. We aim to extend an alternative to rational expectations, adaptive learning, to continuous-time literature. Adaptive learning models allow the agent to misspecify parameters and then—using data or knowledge that becomes available over time—update their estimates of these parameters. One complication with extending this technique is time-dependency in continuous-time models. For instance, the viscosity solution method and the martingale method both require the system to be either independent of time or if the system is time-dependent, it must be solved working backward from the steady-state (i.e., $t = \infty$). Neither of these methods creates an ideal environment for learning; solving the system from the end of time backward does not facilitate the agent's observation of new data. Additionally, the solution methods for continuous-time systems that do not depend on time lack the necessary feedback mechanisms for learning.

The insufficiency of feedback and observability in these methods necessitates the re-examination of continuous-time macroeconomic problems in a new environment. In this work and previous work, we have examined a linear-quadratic (LQ) framework that though independent of time, allows for the feedback necessary for agent-level adaptive learning. There are extensive studies of discrete LQ environments in economics and other fields, as outlined in Kendrick (2005). One of the LQ setting's key features is that the agent maximizes an objective function with a quadratic form, leading to linear first-order conditions. However, most economic models are non-linear and do not fit into the traditional LQ format. Several papers, including Benigno and Woodford (2004, 2006, 2012), use discrete-time linearization techniques to recast non-linear models into the LQ setting. Benigno and Woodford (2012) examines various linearization frameworks and how to ensure accurate linearization, the LQ methods implemented in this paper carefully follow the dynamic programming approach outlined Benigno and Woodford (2012) and Hansen and Sargent (2013).

With few exceptions (Hansen and Sargent, 1991), the continuous-time LQ environment has been under-explored in the economic literature, despite its promise for building tractable and complex economic models. The field of computational finance has a considerable number of works on the continuous-time LQ environment, including Forsyth and Labahn (2007), Wang and Forsyth (2010), Huang et al. (2012), and Xie et al. (2008). In these papers, the optimization problems have a finite horizon, making these LQ settings distinct from the one we will outline in this paper. Additionally, some studies implement learning dynamics in linear optimal regulator problems; for instance, Vrabie et al. (2007) and Wang and Zhou (2019) focus on reinforcement learning in an LQ environment.

Recasting non-linear models into the LQ setting has a few key advantages. The LQ framework allows for the inclusion of many economic variables in a compact model, allowing economists to study complex economies with ease. Additionally, solving LQ problems tends to be less computationally intensive than solution methods for complex non-LQ economies. These advantages are particularly relevant in the context of rational expectations equilibrium, solving the REE of the models outlined in the following sections takes mere seconds using the LQ solution methods. This setting's solution method also does not depend on sparse grids or complicated differentiation schemes. The most important advantage of the LQ-setting, concerning adaptive learning, is that LQ methods contain important feedback mechanisms that allow us to understand the decisions an agent makes based on their observations; this is especially important in our shadow-price learning setting.

We aim to not only create a continuous-time setting where an agent learns how to forecast parameter values accurately; we construct a framework in which an agent learns to forecast and make decisions optimally. An adaptive learning technique that accomplishes both of these goals is shadow-price learning. Shadow-price learning, or SP-learning, assumes the agent uses observations of state variables to understand how the states evolve and future shadow prices. Using these estimates, the agent modifies their behavior using updated shadow prices and the state transition dynamics through the LQ framework's built-in feedback mechanism. Since the state variables' evolution depends on the agent's choices, the agent's behavior influences the states they observe. Eventually, after gaining enough information, the agent in our SP-learning environment learns how to make decisions optimally and how to forecast future state values.

SP-learning allows us to examine better our agent's ability to learn to forecast and make decisions in our economy. The agent in our setting does not know the conditional distribution of key variables and faces uncertainty in our stochastic environment. It has been shown that discrete-time SP-learning can converge asymptotically to fully optimal decision-making in Evans and McGough (2018); we demonstrate that those same results hold in the continuous-time version of a real business cycle (RBC) model. We also compare the results of the continuous-time SP-learners to their discrete-time counterparts. Other works have explored various adaptive learning dynamics in RBC models, including Branch and McGough (2011), Eusepi and Preston (2011), and Mitra et al. (2013); this paper builds on this literature by re-examining learning in a continuous-time real-business cycle model.

We also explore data frequency dynamics in the continuous-time version of the model after inspecting the relationship between the discrete and continuous-time versions of the model and learning outcomes in these settings. Though often overlooked in macroeconomic models, data frequency impacts real-world decisions and macroeconomic outcomes. The importance of data frequency in estimating continuous-time financial models via maximum-likelihood methods has previously been studied in Aït-Sahalia (2010), which examines model estimation based on exact discrete-time estimates that take time-interval length into account. Here we approach this problem using learning algorithms that rely on recursive least squares instead of the maximum likelihood approach.

As part of this exercise, we relax the assumption of continuous updating to better match empirical reality. Our approach assumes that the agent views the time and the economic changes as continuous occurrences and estimates a continuous-time version of our RBC model. Because real-world agents take in information at discrete time intervals and then, in turn, use this information to update their parameter estimates. Some additional considerations have been made regarding data observation.

In particular, we examine how observing continuous processes at different frequencies impacts agents' responses and how information asymmetries can influence economic outcomes by comparing outcomes in an RBC model under learning with varying data collection frequencies and examining a version of the model wherein the agent collects data at varying frequencies. This question of how data availability can impact economic agents is of increasing importance since data today is available at increasingly higher frequencies. While quarterly data will likely be the most common frequency in macroeconomic data for some time to come, as macroeconomists move to include more micro-data and even big-data in macroeconomic analysis, we must consider how data frequency can impact our models.

Our work accomplishes several tasks; first, we demonstrate that the continuous-time learning algorithm does converge to rational expectations equilibrium. Then we closely contrast the outcomes of discrete and continuous-time learning models. Our comparison highlights the varying outcomes between these models, particularly the differences between the volatility of estimates and convergence rates in this setting. Additionally, we explore the linearization of simple macroeconomic models in continuous-time. There is sparse literature on this topic; some linearization of continuous-time macroeconomic models has been researched in other settings (Ahn et al., 2018). We also build on the work done in Evans and McGough (2018) and demonstrate that SP-learning can be modified for a continuous-time setting. Lastly, we examine how data collection can impact the agent's decisions in our model's continuous-time version.

This paper proceeds as follows, section IV.2 outlines a simple real business cycle model in continuous-time and describes the SP-learning algorithm that the continuous-time agent uses to estimate parameters. A discrete-time version of this model is included in the appendix. After separately examining the discrete and continuous-time algorithms, we compare the rational expectations equilibrium of both settings and the learning outcomes in section IV.3. In this section, we compare the second moments of the discrete and continuous-models to the data; the continuous-time version of the model slightly outperforms the discrete version when examining standard deviations of key variables relative to the standard deviation of output. The fourth section examines the impact of data frequency on continuous-time models under learning. The final section concludes.

IV.2 A Real Business Cycle Model—An LQ Approach

The framework used throughout this paper is that of a standard real business cycle model. We select this framework because our baseline model's simplicity allows us to add complex dynamics more easily. To efficiently use common SP-learning methods defined in Evans and McGough (2018), we need our RBC model to fit into a linear quadratic format. Accomplishing this involves linearizing our model objective function and recasting it into a quadratic form. The purpose of utilizing the LQ framework is to generate a model that can be solved recursively with clear and well defined connection between our agent's perceptions, or initial prediction for the value function, and the rational expectations equilibrium value. The continuous-time real business cycle model has a few key differences from a familiar discrete model. Our objective function maintains a similar form; it employs an isoelastic utility function that depends on labor and consumption. However, our discount factor is represented by an exponential function. Additionally, the processes that describe the evolution of capital and government spending now follow Brownian motions. Our household maximizes the following objective function over consumption and labor input,

$$V(k_0, \tilde{z}_0) = \max_{c_t, k_t, h_t} \mathbb{E} \int_{t=0}^{\infty} e^{-\rho t} \left\{ \frac{c_t^{1-\sigma}}{1-\sigma} - \chi \frac{h_t^{1+\varphi}}{1+\varphi} \right\} \tag{IV.1}$$

subject to the following conditions on consumption, productivity, and capital,

$$c_t + i_t = Ak_t^{\alpha}(e^{\tilde{z}_t}h_t)^{1-\alpha} \tag{IV.2}$$

$$d\tilde{z}_t = -\theta_{\tilde{z}}\tilde{z}_t dt + \sigma_{\tilde{z}} dZ_t \tag{IV.3}$$

$$dk_t = (-\delta k_t + i_t)dt. \tag{IV.4}$$

In equation (IV.3) $\tilde{z}$ represents the logarithm of productivity and dZ_t is the increment of the Wiener process[1]. Firms in this economy maximize profits, using a Cobb-Douglas production function, $f(k_t, \tilde{z}_t) = k_t^{\alpha}(e^{\tilde{z}_t}h_t)^{1-\alpha}$. Under this production function the equilibrium rental rate on capital is $r_t = \alpha A k_t^{\alpha-1}(e^{\tilde{z}_t}h_t)^{1-\alpha}$ and the equilibrium wage is $w_t = (1-\alpha)Ak_t^{\alpha}(e^{\tilde{z}_t}h_t)^{-\alpha}e^{\tilde{z}_t}$.

It is standard to take a dynamic programming approach to find the system's steady-state. Our value function problem takes the form of a Hamilton-Jacobi-Bellman (HJB) equation—the continuous-time analog of a Bellman equation. The HJB for the household's problem takes the following form,

$$\rho V(k_t, \tilde{z}_t) = \max_{c_t}\left\{\frac{c_t^{1-\sigma}}{1-\sigma} - \chi\frac{h_t^{1+\varphi}}{1+\varphi}\right\} + V_k(-\delta k_t + Ak_t^{\alpha}(e^{\tilde{z}_t}h_t)^{1-\alpha} - c_t) - \theta V_{\tilde{z}}\tilde{z}_t + \frac{1}{2}V_{\tilde{z}\tilde{z}}\sigma_{\varepsilon}^2$$

the terms V_k, V_z, and V_{zz} all represent partial derivatives of the value function $V(k, z)$ these terms are functions of k and z. The main difference between the HJB and a Bellman equation is how expectations are handled in continuous-time. Deriving expectations of the future value function requires using Itô's lemma since our state variables' evolution depends on continuous-time stochastic processes. Using the HJB we can find the non-stochastic steady state values for our parameters by analyzing

[1]One method of approximating dZ_t, is setting $dZ_t = \varepsilon_t\sqrt{dt}$ where $\varepsilon_t \sim N(0, 1)$ (Dixit, 1992). Thus the increments of the Wiener process are independent and Gaussian.

this system's first order conditions

$$\rho V_k = V_k(\alpha A k^{\alpha-1}(e^{\tilde{z}}h)^{1-\alpha} - \delta)$$
$$c^{-\sigma} = V_k$$
$$\chi h^{\varphi} = V_k(1-\alpha)Ak^{\alpha}(e^{\tilde{z}}h)^{-\alpha}$$

in this setting V_k is analogous to the shadow-price of capital as it measures the estimated value of a unit of capital. With our first order conditions defined, a numerical optimizer can be used to find the non-stochastic steady-state for our household's problem. Knowing the non-stochastic steady-state values of key parameters allows us to linearize our model about this point and simplifies the eventual LQ system we build in this work. After finding the system's non-stochastic steady state, we re-examine the planner's problem. First, we eliminate consumption, c_t, from our objective function by re-writing it as a function of capital, labor, investment, and productivity. This allows us to recast our maximization problem so that it only depends on state and control variables,

$$V(x_0) = \max_{x_t, u_t} \mathbb{E} \int_{t=0}^{\infty} e^{-\rho t} r(x_t, u_t).$$

where,

$$r(x_t, u_t) = \frac{1}{1-\sigma}[Ak_t^{\alpha}(e^{\tilde{z}_t}h_t)^{1-\alpha} - i_t]^{1-\sigma} - \chi \frac{h_t^{1+\varphi}}{1+\varphi}$$

The vectors x_t and u_t contain the state and control variables for the system, $x_t = (1, k_t, \tilde{z}_t)'$ and $u_t = (h_t, i_t)'$. Now that the maximization problem is in terms of the state and control vectors, we use a second-order linear approximation of $r(x, u)$ about the non-stochastic steady state to recast the maximization problem into a linear-quadratic format.

The second-order Taylor expansion about the steady-state—where $\bar{x}$ and $\bar{u}$ are the steady-state values of x and u is standard and the same in continuous and discrete-time,

$$
\begin{aligned}
r(x,u) &= r(\bar{x},\bar{u}) + (x-\bar{x})' r_x(\bar{x},\bar{u}) + (u-\bar{u})' r_u(\bar{x},\bar{u}) \\
&\quad + \frac{1}{2}(x-\bar{x})' r_{xx}(\bar{x},\bar{u})(x-\bar{x}) + \frac{1}{2}(u-\bar{u})' r_{uu}(\bar{x},\bar{u})(u-\bar{u}) \\
&\quad + (x-\bar{x})' r_{xu}(\bar{x},\bar{u})(u-\bar{u})
\end{aligned}
$$

automatic differentiation can be used to compute the partial derivatives of $r(x,u)$. Once this is complete the problem is easily reformatting into a linear quadratic problem. This system does not gain any terms from Itô's lemma since the Taylor expansion is about a single point, instead of a stochastic process.

The maximization problem can now be put into a standard LQ representation. Our objective function now depends on several matrices, R is a 3×3 matrix that summarizes how our states impact the optimization problem directly, Q is a 2×2 matrix that describes how choice variables affect the system, and W is a 3×2 matrix that captures indirect effects (terms that involve both x and u). Below is the continuous-time LQ representation of our RBC model,

$$
V(x_0) = \max_{u_t} \; -\mathbb{E} \int_{t=0}^{\infty} e^{-\rho t} (\hat{x}_t' R \hat{x}_t + \hat{u}_t' Q \hat{u}_t + 2\hat{x}_t' W \hat{u}_t)
$$

where the state variables evolve according to

$$
d\hat{x}_t = A\hat{x}_t + B\hat{u}_t + C dZ_t.
$$

This problem is linearize about the steady-state, thus $\hat{x}_t = x_t - \bar{x}$ and $\hat{u} = u_t - \bar{u}$. The matrices R, Q, and W are equivalent to the following,

$$
\underset{3\times 3}{R} = \begin{bmatrix} r(\bar{x},\bar{u}) & \frac{1}{2} r_x(\bar{x},\bar{u}) \\ \frac{1}{2} r_x(\bar{x},\bar{u}) & \frac{1}{2} r_{xx}(\bar{x},\bar{u}) \end{bmatrix} \quad \underset{2\times 2}{Q} = \begin{bmatrix} \frac{1}{2} r_{uu}(\bar{x},\bar{u}) \end{bmatrix} \quad \underset{3\times 2}{W} = \begin{bmatrix} r_u(\bar{x},\bar{u}) \\ r_{xu}(\bar{x},\bar{u}) \end{bmatrix}
$$

The matrices R, Q, and W are the same for both the discrete and continuous-time version of our model. Although the matrices that summarize our objective function remain the same between these two settings, the matrices that describe the evolution of our state variables are not the exactly alike. In the continuous-time setting our matrix A is noticeable different from what we might expect from a discrete version of the model. This is because in continuous-time our system depends on changes in the state variables not on levels of the state variables at particular moments of time. The matrices that describe the evolution of our states are defined as follows,

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -\delta & 0 \\ 0 & 0 & -\theta \end{bmatrix} \quad B = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \quad C = \begin{bmatrix} 0 \\ 0 \\ \sigma_{\tilde{z}} \end{bmatrix}$$

this difference occurs because in the discrete version of our model we are measuring the level of $\hat{x}_t$ whereas in the continuous version we are calculating changes over increments of time.

To solve the value function problem we utilize a "guess-and-verify" approach by positing that the value function takes the form $V(x_t) = -x_t'Px_t - \xi$, where P is a positive semi-definite matrix. We then solve for P by substituting our supposed value function into the HJB equation

$$\rho x'Px + \rho\xi = \max_u \{x'Rx + u'Qu + 2x'Wu + 2x'P(Ax + Bu) + P(CC')\}. \quad \text{(IV.5)}$$

As previously mentioned, one of the advantages of the LQ setting is its neat recursive solution methods. To implement this method we need to eliminate x and u from equation (IV.5), this can be achieved by finding the system's policy function (a function that defines choices u based on states and model parameters). Using this system's first order conditions with respect to u we can define this system's policy function,

$$u = -(Q')^{-1}(W + PB)'x = -\tilde{F}x. \tag{IV.6}$$

Combining the policy function in equation (IV.6) and the system in equation (IV.5) allows us to eliminate both u and x from the system. With the state-independent version of our value function problem we formulate a recursive algorithm that solves for the value function matrix P (Anderson and Moore, 2007; Vrabie et al., 2007),

$$P_i = -(2\tilde{A}_i')^{-1}(\tilde{F}_i'Q^{-1}\tilde{F}_i + R - 2W\tilde{F}_i) \tag{IV.7}$$

$$\xi_i = \rho^{-1}\text{trace}(P_{i-1}CC') \tag{IV.8}$$

where $\tilde{A}_i = (A - B\tilde{F}_i - .5\rho)$, $\tilde{F}_i = (Q')^{-1}(W + P_{i-1}B)'$, i represents iterations of the recursive algorithm, and P_0 is set exogenously. Additionally, note that this system is formulated under the assumption that A is symmetric.

Several conditions must be met to ensure solutions to the algorithm are asymptotically stable and exist (Lewis, 1986; Anderson and Moore, 2007; Evans and McGough, 2018).

LQ.1 The matrix R is symmetric positive semi-definite and can be decomposed in $R = DD'$ by rank-decomposition, and the matrix Q is symmetric positive definite.

LQ.2 The matrix pair (A,B) is *stabilizable*—there exists a matrix $\tilde{F}$ such that $A - BF$ is stable, meaning the eigenvalues of $A - B\tilde{F}$ have modulus less than one.

LQ.3 The pair (A,D) is *detectable*—if y is a non-zero eigenvector of A associated with eigenvector μ then $D'y = 0$ only if $|\mu| < 0$. Detectability implies that the feedback control will plausibly stabilize any unstable trajectories.

The continuous-time recursive algorithm will have a unique solution provided that the conditions in LQ.1-LQ.3 hold true for this system's R, Q, A, and B matrices and the continuous-time policy function $\tilde{F}$.[2] Conveniently, the conditions outlined in LQ.1-LQ.3 also apply to the discrete-time version of this system; the only difference being that the discrete problem has a different policy function F. Now we turn to adding adaptive learning dynamics to our linearized RBC model.

IV.2.1 Shadow-Price Learning in the Continuous-Time RBC Model

The recursive solution method for our linearized model has a clear linkage between perceptions and actuality, which can be used to establish learning dynamics in this setting (Evans and McGough, 2018). Focusing on equation (IV.7), we see a relationship between our agent's initial perception, P_{i-1}, and updated calculations of their value function matrix P_i. In this setting we define the agent's perceived value function as $V^P(x) = -x'T(P)x$ where $T(P)$ is our T-map, the formal link between perceptions and actuality in learning models.

The T-map, $T(P)$ is matrix function that maps an initial perception of shadow-prices, P, to the updated shadow-prices generated by our agent's choices. Our T-map's fixed point, $T(P^*) - P^* = 0$, is our learning model's equilibrium point, given that certain stability conditions hold. As shown by our derivation of the recursive algorithm in (IV.7) and (IV.8), the stochasticity of our system does not impact the solution to our value function problem. The solution for P is not impacted by the stochastic term C. Knowing this, we begin our explanation of the learning algorithm by focusing on our problem's non-stochastic version. The agent's perceived value function in the continuous-time non-stochastic setting is,

$$\rho V^P(x) = \max_u \{-x'Rx - u'Qu - 2x'Wu - 2x'P(Ax + Bu)\}. \qquad \text{(IV.9)}$$

[2]For a proof of this result, see Lewis (1986).

The unique optimal control decision for perceptions P is given by,

$$u = -\tilde{F}(P)x = -(Q')^{-1}(W + PB)'x.$$

Our policy function is then substituted into equation (IV.9) to find the T-map for our system,

$$T(P) = (2\tilde{A}')^{-1}(\tilde{F}'Q^{-1}\tilde{F} + R - 2W\tilde{F}) \tag{IV.10}$$

here $\tilde{A} = A - \frac{1}{2}I\rho - B\tilde{F}$ and we again assume that $\tilde{A}$ is symmetric. The T-map above describes the mapping between perceptions and reality in a model without stochasticity. The unique fixed point, P^* of this mapping, is the solution to our value function problem. This result has been proved in discrete-time and has been analytically demonstrated to hold for continuous-time models (Evans and McGough, 2018; Lester, 2020). As in the discrete-time case, the non-stochastic case mapping will yield the same fixed point as the T-map for the stochastic version of the system.

Thus far, the continuous-time learning dynamics assume that our agent knows information about the value function's quadratic nature and the values of the state transition dynamics. These assumptions are strict, it is unlikely an average person would understand the form of their utility function let alone assume that it was quadratic in nature. Instead it is more likely they estimate the system's shadow-prices using a simple linear forecasting rule. Equation (IV.11) represents this simple linear forecasting model, where the agent predicts shadow prices μ_t using state values,

$$\mu_t = Hx_t + \varepsilon_t^{\mu}. \tag{IV.11}$$

The matrix H is the shadow-price parameter matrix as we soon show it is directly related to our value function matrix P, in fact $H = -2P$ at rational expectations equilibrium. This forecasting rule can then be used to estimate the shadow-price parameters for our state variables, x.

$$\mathbb{E}[V_x(x)] = \mu^e = Hx$$

where μ^e is the updated estimate of μ. We use this forecasting rule to estimate the future expected utility in our HJB equation,

$$\rho V(x) = \max_u \{-x'Rx - u'Qu - 2x'Wu + (Hx)'(Ax + Bu) + \frac{1}{2}(H'CC')\}.$$

Our modified HJB equation provides insight into how our agent selects optimal choice variables under their forecast of shadow-price parameters. Again we use the policy function to eliminate x and u from our system, to create a compact recursive solution method. We find our learning agent's policy function using the first-order conditions of the HJB,

$$u = -\frac{1}{2}(Q^{-1})'(2W - H'B)x = -\tilde{F}^{SP}(H, B)x.$$

Then to get the mapping from the PLM to the actual law of motion (ALM) we use the envelope condition,

$$\rho\mathbb{E}[V_x(x)] = \rho\mu^e = -2x'R - 2u'W + 2x'A'H + u'B'H. \tag{IV.12}$$

we can rewrite (IV.12) as,

$$\begin{aligned}\mu^e &= \rho^{-1}\{-2x'R - 2u'W + 2x'A'H + u'B'H\} \\ &= \rho^{-1}\big(-2R + 2H'A - (H'B - 2W)\tilde{F}^{SP}(H, B)\big)x \\ &= T^{SP}(H, A, B)x.\end{aligned} \tag{IV.13}$$

The T-map in equation (IV.13) will define the mapping between the agent's PLM in equation (IV.11) and the actual law of motion (ALM) of the system. Our T-map allows us to model the boundedly rational behavior of an agent in this model, using a continuous-time analog to recursive least squares (RLS) that has been derived using the parallels between RLS and the Kalman filter (Lewis et al., 2007; Ljung and Söderström, 1983; Huarng and Yeh, 1992).

A brief discussion of recursive least squares methods is necessary before we define our SP-learning algorithm. To create a functional SP-learning algorithm, we need to define how the agent updates forecasts in the continuous-time setting. In discrete-time, this forecasting updating rule takes the form of RLS, an adaptive algorithm that allows an agent to update their parameter estimates as they acquire additional information. We begin in a discrete setting with a simple linear regression model,

$$y_t = \theta' x_t + \varepsilon_t.$$

For this example y_t is a vector that contains our dependent variable, x_t is a matrix of independent variables (the information that agents' receive), θ is a vector of coefficients, and ε_t our error term, which is assumed to be a normally distributed white-noise process. The recursive least squares algorithm's objective is to update parameter estimates as new data points are observed by minimizing a weighted function of the summed of squared errors. In discrete-time this objective function takes a familiar form,

$$\phi_N(\theta) = \frac{1}{N}\sum_{t=1}^{N} \alpha_t [y_t - \theta' x_t]^2.$$

since this is a weighted least squares problem, α_t is a vector of weights set by the modeler. This vector of weights is related to the gain parameter present in most adaptive learning algorithms. Using this estimator we arrive at a simple recursive algorithm for estimates of the vector of parameters θ_t and the second moment of the data x_t,

$$\hat{\theta}_t = \hat{\theta}_{t-1} + \gamma_t \mathcal{R}_t^{-1} x_{t-1}[y_t - \hat{\theta}'_{t-1} x_t], \tag{IV.14}$$
$$\mathcal{R}_t = \mathcal{R}_{t-1} + \gamma_t [x_t x'_t - \mathcal{R}_{t-1}],$$

the parameter γ_t is the aforementioned gain parameter. The RLS algorithm allows for the agent to use an initial estimate of the coefficient matrix and second moment matrix, $\mathcal{R}_t$, and then update their estimates as they acquire additional information.

RLS takes a similar form in the continuous-time setting; however, our algorithm becomes a system of stochastic differential equations. We begin with a stochastic differential equation instead of the linear regression model,

$$dy_t = \theta' x_t dt + dZ_t$$

the term dZ_t represent the increment of the Wiener process as we've described previously. The RLS estimator now takes the form of

$$\phi_N(\theta) = \frac{1}{N} \int_{\tau=1}^{N} \alpha_\tau [dy_\tau - \theta' x_\tau d\tau]^2.$$

The continuous-time version of RLS is then found using parallels between recursive least squares and other filtering methods (Sastry and Bodson, 1989). We use a constant gain algorithm in this work, thus below is a version of RLS where γ_t is set as a constant. Implementing this version of RLS means that individuals put equal weight on all observations and expect some noise in their parameter estimates,

$$d\hat{\theta}_t = \frac{1}{1-\gamma_t} \mathcal{P}_t x_t [dy_t - \hat{\theta}'_{t-1} x_t dt], \tag{IV.15}$$
$$d\mathcal{P}_t = \frac{1}{1-\gamma_t} [\gamma_t \mathcal{P}_t - \mathcal{P}_t x_t x'_t \mathcal{P}_t] dt.$$

It is most common in continuous-time literature to use the matrix $\mathcal{P}_t$, the covariance matrix, to avoid matrix inversion. The "recursive least squares filter" as it often called in engineering literature, is strikingly similar to the system in (IV.14). By observation one can see that (IV.15) is essential the derivative of the system in (IV.15) with respect to time. For a full derivation of the continuous-time RLS system, please see the appendix or Goodwin and Mayne (1987).

With an established background in Shadow-Price learning dynamics and continuous-time recursive least squares, we can now outline an algorithm that models an agent's bounded rationality in our framework. In this system, the agent's policy function $\tilde{F}$ impacts the choices they make, and the future states they observe. Thus, our learning algorithm includes updates to the state variable impacted by the agent's choices and subsequent updates to agent's choice and forecasts based on the current state of the economy.

$$
\begin{aligned}
dx_t &= Ax_t dt + Bu_t dt + CdZ_t \\
d\mathcal{P}_t &= \frac{1}{1-\gamma_t}(\gamma_t \mathcal{P}_t - \mathcal{P}_t x_t x_t' \mathcal{P}_t)dt \\
dH_t' &= \frac{1}{1-\gamma_t}\mathcal{P}_t x_t (\lambda_t - H_t x_t)' dt \\
dA_t' &= \frac{1}{1-\gamma_t}\mathcal{P}_t x_t (dx_t - Bu_t dt - A_t x_t dt)' \\
u_t &= -F^{SP}(H_t, B)x_t = -\frac{1}{2}(Q')^{-1}(2W - H_t B)' x_t \\
\lambda_t &= T^{SP}(H_t, A_t, B)x_t \\
\gamma_t &= \kappa(t+N)^{-\nu}.
\end{aligned}
\tag{IV.16}
$$

In this algorithm $\mathcal{P}_t$ is the covariance matrix, unlike the discrete algorithm that uses $\mathcal{R}_t$, an approximation for the second moment, $\mathcal{P}_t$ can tend toward zero, something we need to be careful of in our setting (Sastry and Bodson, 1989). The use of $\mathcal{P}_t$ reduces the computational burden of taking the matrix inverse and is more in-line with the continuous-time Kalman filter notation. The gain parameter, γ_t, will again be assume to be constant with $\kappa = 0.01$ and $\nu = 0$.

Continuous-Time Learning Results

Now that we have defined an agent's bounded rationality in this setting, we can examine our learning algorithm's convergence. Before we can examine the dynamics of the learning model, the model parameters must be set. The continuous-time model was selected to align with parameters from discrete-time literature. To select appropriate values for some of the parameters, such as the discount factor, we consulted Kaplan et al. (2018). For the continuous-time SP-learning algorithm, it is necessary to approximate the time-step dt. We selected 1/100. Since the discrete-time version of the model is calibrated based on quarterly data, $dt = 1/100$ indicates that our agent updates parameters at least once a day.[3] The final parameters, $\sigma_{\tilde{z}}$ and $\theta_{\tilde{z}}$ were set in accordance with discrete time literature. The process for $\tilde{z}_t$ defined in equation (IV.3) is the continuous-time analog to an AR(1) process, thus there exist many comparisons of the two. Basing our estimates off a discrete model with an auto-regressive term of 0.895 and white-noise term with a standard deviation of 0.01, the parameters of the continuous-time model are set to $\theta_{\tilde{z}} = 0.105$ and $\sigma_{\tilde{z}} = 0.01$.[4] Table IV.1 summarizes the parameter values for the continuous-time model.

[3] Approximately 1.09 times a day. Assuming 91 days in a quarter.

[4] With our naive estimation approach, the limiting distributions of the discrete and continuous-time models have approximately the same variance (Posch et al., 2011). For the discrete-time case $\text{Var}(x) = \frac{\sigma_{\tilde{z}}^2}{1-\theta_{\tilde{z}}^2} = \frac{(0.01)^2}{1-0.895^2} \approx 0.0005$. While in the continuous-time setting $\text{Var}(x) = \frac{\sigma_{\tilde{z}}^2}{2\theta_{\tilde{z}}} = \frac{(0.01)^2}{2\cdot 0.105} \approx 0.0005$.

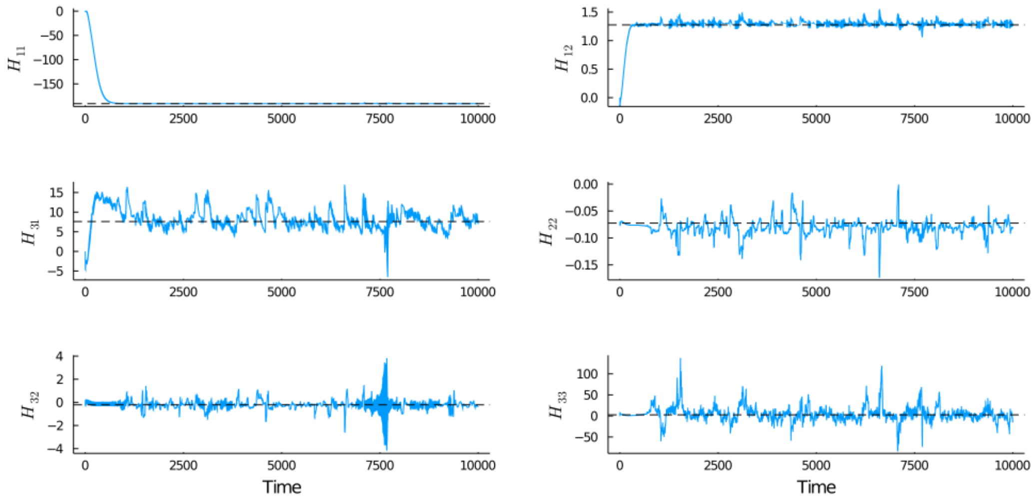
Table IV.1
Continuous-Time Parameter Values

Description		Parameter Value
A	Total Factor Productivity	1.0
ρ	Discount factor	0.01
σ	Intertemporal elasticity of subst.	1.0 (log utility)
φ	Frisch elasticity of labor supply	-1.0 (log utility)
χ	Disutility of labor	1.75
α	Capital share	1/3
δ	Depreciation rate	0.025
$\theta_{\tilde{z}}$	Drift parameter for tech.	0.105
dt	Approximation of time-step	1/100
$\sigma_{\tilde{z}}$	Standard deviation for tech.	0.01

After finalizing key parameter values, we focused on the initial values for the learning algorithm. The misspecification used in this setting varied from the discrete-time version. Here A and H were set to small negative constants times identity matrices. Initial values for x_0 and u_0 were, again, set near steady-state values. The second-moment matrix $\mathcal{P}$ was initialized based on initial values of x_0. Misspecification in the continuous-time was set to ensure stability with the continuous-time T-map and policy function. While the SP-learning algorithm's initialization does not need to be near the REE, it is best if the initial policy is stable, meaning the T-map's derivative has eigenvalues within the unit root. Additionally, the agent in this setting understands the basic structure of the transition matrix A and does not use the constant in estimating parameters; instead, they estimate the technology and capital processes' parameters separately.

Simulations of the model were run for the equivalent $10,000$ discrete-time periods so the agents were able to update their forecasts over $(100 \times 10,000)$ iterations, since $dt = 1/100$. Examining figure IV.1 we see that in the continuous-time model the agent's estimates converge quickly and fluctuate around their REE values. At the end of $10,000$ periods the agent in continuous-time model forecasts a shadow-price parameter matrix that is a distance of 11.25 from REE according to the matrix norm

Figure IV.1
Convergence of Shadow-Price Parameters



measurement. The agent also updated their estimates of the state transition matrix A over these 50,000 periods. The distance between the agent's estimate of A and the true transition matrix, measured using matrix norms, is 0.012 after only 10,000 periods.

Thus far, we have demonstrated that the continuous-time real business cycle model converges to REE under our SP-learning algorithm. Next, we compare these models' learning outcomes to understand the differences between bounded rationality in these settings. The following section of this paper examines parameter values and their distances from REE values, the volatility of these models' estimations, and the second moments of key variables, as is common in real business cycle literature.

IV.3 Comparing Discrete and Continuous-Time Systems

We now compare the REE values to the learning models' outcomes, with the models initialized "far-away" from the rational expectations equilibrium. As measured by state and choice variable values, the economic outcomes of the discrete and continuous-time models closely match REE values after 50,000 periods; however, the continuous-time model comes closer to reaching the REE for shadow-price parameters. Additionally, the continuous-time model's shadow-price parameter estimates display less volatility than the estimates for the discrete-time model, implying that the continuous-time learning estimates exhibit more stability than their discrete counterparts.

For some reference, the rational expectations equilibrium values of the discrete and continuous-time shadow-price parameter matrices are given in equations (IV.17) and (IV.18). Solutions for the steady-state values of key variables and the value function matrix H are similar for the discrete and continuous-time versions of the real-business cycle model.

$$H^*_{\text{Discrete}} = \begin{bmatrix} -190.764 & 1.29026 & 7.67191 \\ 1.29026 & -0.0753887 & -0.210606 \\ 7.67191 & -0.210606 & 2.73081 \end{bmatrix} \tag{IV.17}$$

and

$$H^*_{\text{Continuous}} = \begin{bmatrix} -190.642 & 1.2759 & 7.6087 \\ 1.2759 & -0.0724069 & -0.212827 \\ 7.6087 & -0.212827 & 2.64364 \end{bmatrix} \tag{IV.18}$$

The fact that the continuous-time matrix is so close to the discrete-time version (the matrix norm of the discrete solution minus the continuous one is 0.386) solidifies that these matrices are the equivalent solutions to their respective problems.

There are some minor computational gains when solving for the rational expectations equilibrium in continuous-time. The discrete version of the recursive LQ algorithm presented which is presented in the appendix converges in 0.009413 seconds and 1,333 iterations for the model with government spending. While the continuous version, from section IV.2, converges in 0.000316 seconds and 11 iterations.[5] Additionally, the discrete-time algorithm used 10.764 MiB of memory, while the continuous-time version only required 0.2849 MiB. The continuous-time LQ algorithm's speedier convergence is not observable by the programmer in this instance but could have serious impacts on a more complex economy with more than four state variables.

IV.3.1 Comparing Learning Outcomes

Comparing learning outcomes between the discrete and continuous-time models is difficult since there are many factors to consider, such as the distance between REE and the initial specifications and how the initial covariance/second-moment matrix is set. Since the learning algorithms both implement constant gain, the most accurate method of comparing learning outcomes in both models is to examine the learning parameters over the last $1,000$ periods of the learning iterations.

For a better comparison between the discrete and continuous-time cases, we have only included points from the continuous models that occurred at the end of each discrete period, so the continuous-time mean values and standard deviations are calculated using the same observation size as the discrete-case. Without this sampling scheme, the continuous-time standard deviations would still be almost the same any change in these values occurred at the third (or higher) decimal place. Standard deviations of state and choice variables are included in the table, in parentheses underestimated parameter values.

In our shadow-price learning algorithm, the agent forecasts two key objects, the state transitions matrix, and their shadow prices; these state-transition dynamics and shadow-prices impact the system's evolution via the choices our agent makes regarding investment and hours worked. We first examine the impact of learning on key parameters' values, such as investment and capital, before more closely examining

[5] Run-times were calculated using the instructions in Julia documentation. This requires compiling functions beforehand for accurate measurements.

shadow-price parameters. Table IV.2 lists the REE equilibrium values of economic variables for the continuous-time and discrete-time models as well as the averaged learning outcomes over the last 1,000 periods.

Table IV.2
Steady State Values and Learning Outcomes

	Discrete		Continuous	
Variable	**REE Value**	**Learning**	**REE Value**	**Learning**
Labor	0.333	0.333	0.333	0.333
Investment	0.244	0.243	0.245	0.245
Capital	9.749	9.758	9.797	9.805
Consumption	0.783	0.783	0.784	0.785
Wages	2.054	2.055	2.057	2.059
Rental Rate on Capital	0.035	0.035	0.035	0.035

Although the differences between the discrete-time and continuous-time steady state values are similar they highlight a few key differences between the systems. In the continuous-time system, steady-state wages are slightly higher, as is an investment. This is likely necessary to help offset continuous-time discounting. Learning outcomes between these two models are similar; however, the discrete version of our learning model appears to underestimate the level of capital. This likely comes from the shadow-price parameter estimates as these impact the agent's investment choices, which in turn impact capital accumulation.

Next, we examine the shadow-price parameters. The matrix norm between the agent's forecast of H and the REE was 2.35. in the continuous case and 2.42 in the discrete case. In the discrete case, the matrix norm between the initial guess H_0 and the true value was 192, and in the continuous version, that same measure was 191.

To analyze the difference between shadow-price forecasts in the continuous and discrete model, we again examine the last 1,000 periods of both learning algorithms. Table IV.3 contains the average learning outcome over the last 1,000 periods, the standard deviation of the parameter over the last 1,000 periods, and the rational expectations equilibrium values.

Table IV.3
Shadow-Price Parameter Outcomes

	Learning Outcome		REE Value	
Variable	**Discrete**	**Continuous**	**Discrete**	**Continuous**
Constant	-189.909	-190.564	-190.764	-190.642
	(0.026)	(0.0018)		
Capital	-0.077	-0.075	-0.075	-0.072
	(0.0002)	(0.0000)		
Productivity	2.544	2.548	2.731	2.644
	(4.64)	(0.004)		

Overall the continuous-time version of the model has more accurate measures of shadow-price parameter values and lower standard deviations for these parameter estimates.

IV.3.2 Comparing the Accuracy of the Models' Second Moments

After examining the parameter estimates under SP-learning dynamics, a few questions arise about the impact of continuous-time on the model's second moments. In real business cycle literature, it is common to examine the theoretical model's second moments and compare them to economic data (Plosser, 1989; Hansen and Wright, 1992; Romer, 1996). In this exercise, we compare the outcomes of the discrete and continuous-time learning models to second moments from data that have been detrended using the HP-filter.

Economic data from 1960-2019 on GDP, consumption, investment, wages, and hours worked was collected using the FRED database. Then using the HP-filter and logarithmic transformation, we detrended the data. We simulated the same model used in the previous sections to compare the second moments between the data, discrete, and continuous-time systems. The calibration of our model was changed the stochastic process for technology in this version has an auto-correlation term of 0.99 in the discrete case and 0.01 for the continuous-time case. In both instances, the standard deviation of the white-noise process was also set to 0.01. In the continuous-time setting time intervals, dt are approximated as 1/100. This approximation of dt means that the agent updates their estimates about once a day since the discrete model is calibrated using quarterly data.

Each model's economy was simulated for 240 periods (the same number of periods present in the data). We ran these simulations one thousand times for the discrete and continuous-time models with learning dynamics and applied the logarithmic transformation and HP-filter to these 1,000 series. We report standard deviations and correlations averaged over all 1,000 simulations in table IV.4. Since the variables we measured are primarily flow variables, the continuous-time model's points were aggregated by integrating information to compare with the discrete model. Table IV.4 displays the standard deviations of values from the data and the theoretical models, along with the correlations between key variables and output.

Table IV.4
Second Moments and Autocorrelations of Key Economic Variables

	Standard Deviation*			**Correlation w. Output**		
Variable	**Data**	**Discrete**	**Cont.**	**Data**	**Discrete**	**Cont.**
Output	1.43%	1.30%	1.06%	1.00	1.00	1.00
Consumption	0.510	0.471	0.515	0.748	0.971	0.773
Investment	2.880	2.815	2.879	0.799	0.989	0.972
Hours	0.646	0.365	0.471	0.650	0.982	0.854
Wage	0.660	0.645	0.646	0.172	0.994	0.925

*standard deviations for variables other than output are measured relative to output

The continuous-time version of the model matches the relative second moments from the data slightly better for consumption, investment, and hours worked. While the continuous version of the model still overestimates economic variables' procyclicality, it does so by less than the discrete version.

IV.4 Learning and Data Frequency

Now that we have outlined methods and results for continuous-time learners, we examine outcomes when an agent takes in information over larger intervals. In this section, the economy that the agent participates in is continuous, and state variables update continuously as well; however, the agent is only capable of taking in observations at lower frequencies. This setting parallels the real-world where we may believe that economic factors like productivity or even GDP are continually updating. However, due to our limited ability to take in information and data availability restrictions, we cannot constantly update our estimates of these parameters. Our state variables evolve according to a continuous-time process we approximate as updating daily—$dt \approx 1/100$. We examine three different agents in this setting. The first updates information weekly, the second bi-weekly, and the third every day. An essential aspect of our agent's forecasts is that they understand that they are approximating a higher frequency process, i.e., the weekly updater understands that they are using weekly data and includes that information in their estimations.

IV.4.1 Learning under Varying Data Frequencies

As previously mentioned, the agents in this section exist in an economy where variables are continuously evolving. They maximize their utility subject to the continuous-time RBC model in section IV.2. However, the agents in this setting do not continuously update their parameters. Instead, they only observe data at specific time intervals, and they know they are approximating a continuous-time system using this discrete data. Considering this, they use Δ—the time step of their discrete observations—in their forecasts to approximate dt. The learning algorithm implemented by these agents is similar to the continuous-time algorithm, with a few key changes.

$$
\begin{aligned}
dx_t &= Ax_t dt + Bu_t dt + CdZ_t \\
\Delta \mathcal{P}_t &= \frac{1}{1-\gamma_t}(\gamma_t \mathcal{P}_t - \mathcal{P}_t x_t x_t' \mathcal{P}_t)\Delta \\
\Delta H_t' &= \frac{1}{1-\gamma_t}\mathcal{P}_t x_t(\lambda_t - H_t x_t)'\Delta \\
\Delta A_t' &= \frac{1}{1-\gamma_t}\mathcal{P}_t x_t(x_t - Bu_t\Delta - A_t x_t \Delta - x_{t-\Delta})' \qquad \text{(IV.19)} \\
u_t &= -F^{SP}(H_t, B)x_t = -\frac{1}{2}(Q')^{-1}(2W - HB)'x_t \\
\lambda_t &= T^{SP}(H_t, A_t, B)x_t \\
\gamma_t &= \kappa(t+N)^{-\nu}.
\end{aligned}
$$

The state variables for this system still evolve continuously but now they are observed at distinct periods of time. Meaning the agent will observe, $x_1, x_{1+\Delta}, x_{1+2\Delta} + \ldots x_\tau$ where τ represents the end period of the model.

We examine three different agents that observe data at the three varying frequencies in this system. For ease, we assume that our state variables evolve almost daily and approximate $dt = 1/100$. This is consistent for all agents in this section and is the same approximation of dt used in the previous sections. For these versions of the model, the same parameterization from IV.2 is recycled; however, the three models explored in this section have an additional parameter Δ. The new parameter Δ represents the intervals at which agents take in additional information, whereas dt is the actual time interval for the data generating process.

We use three specifications, one where $\Delta = 1/25$ for individuals that update their estimates every four days, or about twice a week, another with $\Delta = 1/50$ to represent weekly up-daters, and the last has $\Delta = 1/100$ meaning that the agent observes every point in the true data generating process. In all three of these cases, agents updated thier shadow-price forecasts over 10,000 discrete-time periods (in this case, over 10,000 quarters). Initially, this exercise examines the differences in learning dynamics over varying time intervals. However, learning outcomes are nearly identical in all three cases– likely because we did not constrain the number of learning iterations and gave each type of agent 10,000 periods of data. The only major difference between these specifications was run-time. Measurements for matrix norms and the standard deviation of matrix norms were measured using the mean matrix norms and standard deviation of the matrix norm over the last $1,000$ discrete-time periods.

Table IV.5
Continuous-Time Learning Results under Varying Data Frequencies

dt	Δ	**Matrix Norm**	**Norm Std.**	**Run Time**
$dt = 1/364$	$\Delta = 1/364$	12.86	15.37	187
	$\Delta = 1/91$	13.26	15.56	58
	$\Delta = 1/52$	13.34	15.49	19
	$\Delta = 1/26$	13.07	15.12	12
dt=1/100	$\Delta = 1/100$	13.28	16.96	36
	$\Delta = 1/50$	13.25	16.88	12
	$\Delta = 1/25$	13.46	17.13	7

Table IV.5 demonstrates that one the short comings of the continuous-time learning algorithm, long run-times, can be minimized by implementing different sampling frequencies. This table also includes extra specifications using $dt = 1/364$ to provide additional evidence on how sampling frequencies and smaller approximations for dt can reduce computational time.

IV.5 Conclusion

Rational expectations is a powerful modeling tool that allows economists to compute equilibrium outcomes efficiently. However, as we look to micro-foundations, that assumption of rational expectations is far too strict. It is unlikely that individuals understand the evolution and distribution of productivity or the capital stock. It is also unlikely that they understand how to fully optimize when making decisions.

The adaptive learning literature has relaxed both of these assumptions; in this paper, we relax a third assumption: that agents make decisions intermittently at fixed time intervals. Previous literature has approached optimization and forecasting as a discrete problem. We introduce a continuous-time shadow-price learning algorithm that converges to the rational expectations equilibrium without imposing unrealistic assumptions.

Not only does this result match the point estimates in the continuous-time rational expectations model, it improves the estimates volatility when compared against discrete-time models and economic data. This result supports the outcomes of continuous-time rational expectations models while demonstrating that convergence in the continuous-time setting is not the same as convergence in the discrete-time case. Our continuous-time model displays less volatile shadow-price parameter estimates and smoother convergence (measured using matrix norms) of the shadow-price parameter matrix to REE values. This decreased volatility demonstrates that when agents gain more information more rapidly and have the ability to update their forecasts more frequently, they will make smaller, less reactionary updates to their predictions and choices.

Furthermore, we demonstrate that the continuous-time version of the model can provide improvements when matching the data's second moments. Since the continuous-time model more closely matches the data and displays less volatile convergence to the REE values of shadow-price parameters, we can conclude that our continuous-time model captures important dynamics that the discrete version of our model does not.

Though helpful in demonstrating the continuous-time framework, continually updating is unlikely for agents and computationally burdensome for modelers. In a refinement exercise, we introduce an alternative sampling method that allows the continuous-time agent to sample data observed at high frequency and update their forecasts less frequently. This alternative sampling method results in faster computational time and similar parameter estimates.

Since the continuous-time shadow-price learning algorithms presented in this work converge to REE, it would be simple to conclude that we should model agents as fully rational and fully optimizing or as discrete decision-makers. However, in reality, agents are not infinitely lived, and they may experience structural changes that will cause them to re-evaluate their decisions. Additionally, there are key differences between convergence in continuous and discrete settings. These dissimilarities show that the agent more gradually converges to REE in continuous-time and makes less volatile choices as they near convergence. It seems that continuous-time agents make more stable decisions and smaller updates to their choices. The proposed framework improves on two desirability properties of agent optimization models: predictive precision and assumption parsimony.

Continuous-time adaptive learning literature is limited, and there is much work to be done on this topic. We intend to explore extensions to this work, including further improvements to the shadow-price learning algorithm. Improvements to our basic shadow-price learning algorithms likely exist; unlike the discrete version of the algorithm, our problem is a system of differential equations with no matrix inverse necessary. Therefore, we could attempt to simplify our problem using matrix algebra. We also would like to apply this method to a wide range of macroeconomic and financial models. For instance, many portfolio selection problems are already in the LQ format; thus, our shadow-price learning framework could be easily extended to these models. Additionally, we would like to find applications for continuous-time adaptive learning algorithms to economic models outside of the linear-quadratic format.

CHAPTER V

DISSERTATION CONCLUSION

In this dissertation, we examine how to incorporate adaptive learning dynamics in continuous-time macroeconomic models. There are several motivations for extending these techniques to continuous-time frameworks. Firstly, we aim to provide an alternative to rational expectations—a modeling technique wherein individuals have a complex understanding of the economy—in continuous-time models, thus far, the literature relies on this strong modeling assumption. However, in our adaptive learning framework, individuals don't necessarily understand economic parameters or the economic model they interact with; instead, they use observations from the world around them to update their estimations and perceptions.

Capturing this level of realistic agent-level behavior not only adds credibility to our model assumptions. It also aids us in exploring key features of models that are important to our economy and policymakers. For instance, if policies change or follow a particular rule, policymakers want to understand how quickly individuals react to these changes and how fast they adapt their economic expectations. Additionally, they would want to understand if people will respond to specific changes in rational or predictable ways. Adaptive learning provides insight into how individuals react to new information and economic changes; however, the literature thus far focuses on discrete-time modeling, which often assumes that decision-makers update their predictions and gather information at quarterly or at most monthly intervals. In this dissertation, we relax the assumption that data is only available over these longer discrete intervals. We allow individuals to access high-frequency information and enable them to make decisions at higher frequencies.

Chapter II reviews continuous-time and adaptive learning literature, examines the connections between continuous-time and discrete-time models and begins exploring methods of incorporating adaptive learning rules in continuous-time frameworks. This initial work adds salient information about estimation and data frequency to existing literature and lays the groundwork for the rest of this dissertation. The third chapter delves into a major issue with continuous-time economic models—solution methods accessible in continuous-time macroeconomics do not allow for feedback because they rely on agents' knowledge of the future. To add endogenous learning rules to continuous-time models, we build a linear-quadratic framework in continuous-time by drawing upon engineering literature. Linear-quadratic methods are standard in many scientific fields because, in these models, individuals observe the world around them and respond based on their observations, making this framework ideal for learning. Combining this framework with a continuous-time analog to recursive least squares, we find that it is possible to incorporate bounded rationality in continuous-time economic models.

We conclude with chapter IV; this chapter applies the shadow-price learning environment developed in chapter III to a classic real business cycle model. To accomplish this, we must linearize the RBC model and carefully implement our SP-learning algorithm. In addition to extending the methods outlined in chapter III, chapter IV analyzes the continuous-time learning model's ability to match the data and overall accuracy when compared to a discrete version of the learning model. We find that the continuous-time model better matches relative second moments from economic data. Lastly, we explore the implications of sampling frequencies in our continuous-time model. We find that decreasing the sampling frequency in our continuous-time SP-learning model can decrease computational time while maintaining results. However, it seems that estimates are more likely to possess increased accuracy and lower volatility when sampling is more frequent.

This dissertation begins a distinct body of work that has many potential extensions. With this research, we have developed rich and complex methods of modeling expectations in continuous-time macroeconomic models. The shadow-price learning framework outlined in this dissertation could readily be applied to various LQ models, which are present in economic and finance literature, or simple linearized models. Additionally, the methods outlined in this work can be improved upon using various sampling schemes or streamlining the continuous-time recursive least squares algorithm. However, much work is necessary to incorporate these dynamics in sophisticated mainstream models with complicated economic dynamics or multiple agents. Ultimately, this work aims to create a baseline from which continuous-time heterogeneous agent models could utilize adaptive learning techniques. Such work would be appealing to policymakers as heterogeneous agent models provide insight into how resources are distributed amongst individuals from varying backgrounds, something that is relevant today.

APPENDIX A
THE KOLMOGOROV FORWARD EQUATION

The derivation of the KF equation is not always intuitive. Dixit (1992) gives one of the clearest derivations of the KF equation targeted at economists. In this next section, we will present this derivation and compare the KF equation to discrete distributional methods. The Kolmogorov forward and backward equations govern the more general dynamics of stochastic processes. Suppose we are in a discrete system at a point $(x_1, t_1 + \Delta t)$ there two ways we could have gotten to this point. First, we could have previously been at $(x_1 - \Delta h, t_1)$ before moving forward in the x direction. Alternatively, we may have been located at $(x_1 + \Delta h, t_1)$ and then moved back in the x direction. Using this information we can write the probability of of being at (x_1, t_1) as,

$$\Pi(x_1, t_1 + \Delta t) = p\Pi(x_1 - \Delta h, t_1) + q\Pi(x_1 + \Delta h, t_1)$$

in this equation p is the probability of moving forward in the x direction and $q = 1-p$ is the probability of moving backward. For a Brownian motion, $dx = \mu dt + \sigma dW_t$, $p = \frac{1}{2}[1 + \frac{\mu}{\sigma^2}\Delta h]$ and $q = \frac{1}{2}[1 - \frac{\mu}{\sigma^2}\Delta h]$. Using a Taylor expansion, our previous expression will become,

$$\begin{aligned}\Pi(x_1, t_1) + \Pi_t(x_1, t_1)\Delta t + \mathcal{O}(\Delta t) = &\frac{1}{2}[1 + \frac{\mu}{\sigma^2}\Delta h]\big(\Pi(x_1, t_1) - \Pi_x(x_1, t_1)\Delta h \\ &+ \frac{1}{2}\Pi_{xx}(x_1, t_1)(\Delta h)^2 + \mathcal{O}(\Delta h)^2\big) \\ &+ \frac{1}{2}[1 - \frac{\mu}{\sigma^2}\Delta h]\big(\Pi(x_1, t_1) \\ &- \Pi_x(x_1, t_1)\Delta h + \frac{1}{2}\Pi_{xx}(x_1, t_1)(\Delta h)^2 + \mathcal{O}(\Delta h)^2\big).\end{aligned}$$

As $\Delta t \to 0$ this equation will become

$$\Pi_t(x_1, t_1) = \frac{1}{2}\sigma^2 \Pi_{xx}(x_1, t_1) - \mu \Pi_x(x_1, t_1),$$

our standard KF equation. This derivation is less intuitive and not as straight forward for other stochastic processes.

A.1 Deriving Equation (12)

First simplifying the original equation (II.10) we get,

$$\beta^{\Delta t} \lambda_{t+\Delta t}[\big(f'(k_{t+\Delta t}) - \delta\big)\Delta t + 1] = \lambda_t$$

Then we can set $\lambda_{t+\Delta t} = \lambda_t + \dot{\lambda}\Delta t$ and use the expansion $\beta^{\Delta t} = 1 + \Delta t \ln \beta$

$$[1 + \Delta t \ln \beta][\lambda_t + \dot{\lambda}\Delta t][\big(f'(k_{t+\Delta t}) - \delta\big)\Delta t + 1] = \lambda_t$$

Foiling this out yields the following.

$$\begin{aligned}
&[\lambda_t \Delta t \ln(\beta) + \lambda_t + \dot{\lambda}\Delta t + \dot{\lambda}(\Delta t)^2 \ln(\beta)][f'(k_{t+\Delta t}) - \delta]\Delta t \\
&= \lambda_t - \lambda_t - \lambda_t \Delta t \ln(\beta) - \dot{\lambda}\Delta t - \dot{\lambda}(\Delta t)^2 \ln(\beta)
\end{aligned}$$

Diving through by Δt and then assuming any remaining terms with Δt are negligible we will get equation (II.12).

$$\lambda_t[f'(k_{t+\Delta t}) - \delta] = -\lambda_t \ln \beta - \dot{\lambda}$$

A.2 Steady State Algorithm for solving the HJB

The steady-state algorithm used in section 5 of this paper comes from Achdou et al. (2014) and is one of the more simple solution methods in this setting.

For a simple Ramsey model as described in section 5, the algorithm proceeds as follows,

1. Compute $\partial_k V(\cdot)$ for all k
2. Compute the value of consumption from $c_i = (u')^{-1}[\partial_k V(\cdot)]$
3. Implement an upwind scheme to find "correct" $\partial_k V(\cdot)$
4. Using the coefficients determined by the upwind scheme create a transition matrix for this system
5. Solve the following system of non-linear equations

$$\rho V^{n+1} + \frac{V^{n+1} - V^n}{\Delta} = u(V) + A^n V^{n+1}$$

6. Iterate until $V^{n+1} - V^n \approx 0$

For the most part, the algorithm described above is a typical finite difference scheme. The main difference between this algorithm and what is often used for value function iteration is the upwind scheme. The upwind scheme described in this paper selects a forward difference when we experience positive drift, i.e., positive savings, in our variable of interest, a backward difference if this drift term is negative, and selects a steady-state value if we see no drift. In this scheme, we continue our difference algorithms for $(n+1)$ iterations until we are no longer significantly updating our value functions.

Now, we will describe the upwind scheme in more detail. For the algorithm described above, we need to approximate three different derivatives, the backward and forward difference of the first derivative of the value function and the second derivative for the value function.

The forward difference will be given by,

$$\frac{V_{i+1} - V_i}{\Delta k}$$

and the backward difference will be

$$\frac{V_i - V_{i-1}}{\Delta k}.$$

The second derivative will be approximated by

$$\frac{V_{i+1} - 2V_i + V_{i-1}}{(\Delta k)^2},$$

where i represents the point in the k grid-space. When the drift of the state variable is positive, the upwind scheme will choose the forward difference, and when it is negative, the upwind scheme will select the backward differences. If neither of these conditions holds, then the upwind scheme will select a steady-state value.

There are several different ways to explain the upwind scheme. We can think of it as a method for consistent estimation in this setting. In this setting, we need our finite difference scheme to take the dynamics of our system into consideration.

Suppose we have the following HJB,

$$\rho V(k,z) = \max_c \; u(c) + \partial_k V(k,z)(f(k) - \delta k - c) - \partial_z V(k,z)(\eta z_t) + \frac{1}{2}\partial_{zz} V(k,z)\sigma^2$$

in order to approximate the derivatives of our values functions, we need to consider the flow of k and z. For z this is simple since the sign of $-\eta z_t$ will be the same for all positive values of z_t, we can use the backward difference at all points. This works as long as our z-grid contains only positive points. (The $\log(z_t)$ processes from earlier in this paper was used to help ensure we could use only one differencing method).

Since our values of k cannot be similarly limited, especially since they rely on c, we need to use an upwind scheme in order to approximate the derivative in this dimension. Suppose we at one specific point in the k-dimension, and we are unsure about the shape and differentiability of our value function. However, we do know that the drift of the k process has a positive drift at that value of k_i or, in other terms, the savings function at k_i is positive. As discussed in Achdou et al. (2014), we can then what matters most is how our value function changes when capital increases by a small amount. Conversely, if savings are negative, we want to measure how the value function changes when capital decreases by a small amount. This is our motivation for using the upwind scheme. This numerical approximation technique will take the forward difference when savings is positive and the backward difference when savings is negative.

It is worth noting that in fluid dynamics literature, the upwind scheme is defined differently. In these works, the upwind scheme takes the forward difference when drift is negative and the backward difference when the drift is positive. This difference emerges because these systems of partial differentials are solved forward in time, whereas in this setting, we in effect solving our system of equation backward in time. In the problems outlined in this paper, we are solving for the steady-state of our system. Hence, we are effectively at $t = \infty$, meaning that our solution techniques can be thought of as working backward in time.

APPENDIX B

ALTERNATIVE SPECIFICATIONS

This section of the appendix outlines several different initial specifications that could have been used for the models in section 5 of this paper.

B.1 Learning the Process for Productivity

In section 5, the agents specified that θ was larger than its true value, 0.105 and σ was smaller than its true value, 0.015. Now, in the following sections, we will look at various misspecifications of these parameters and the convergence results. Below, is a table of the various initial values we examine in sections D.1.1-D.1.7.

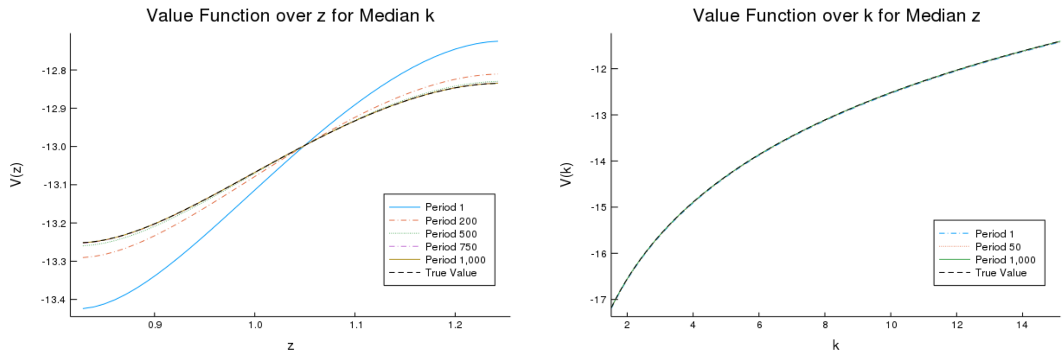
Table B.1
Initial values for σ and θ

Specification	θ_g	σ_g^2
Section 5	0.25	0.008
B.1.1	0.08	0.008
B.1.2	-0.11	0.008
B.1.3	2.0	0.008
B.1.4	0.25	0.8
B.1.5	0.25	1.5
B.1.6	-0.11	1.5
B.1.7	-0.11	0.8

B.1.1

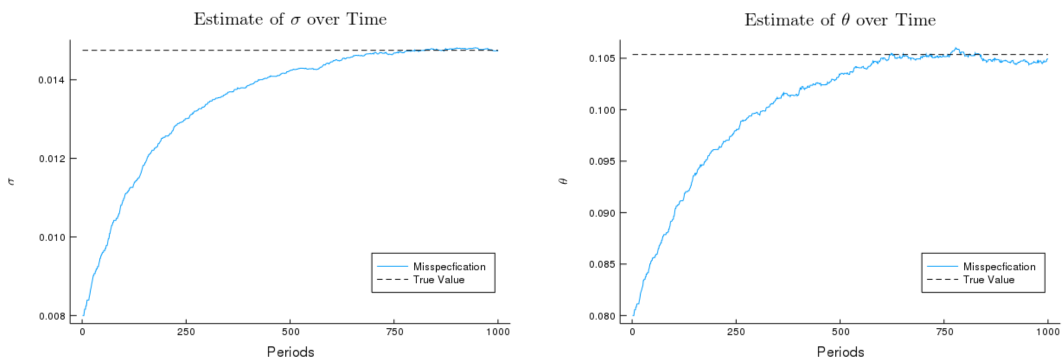
We first examine what would happen to this model if θ was set to be smaller and the correct sign and if σ was also a smaller value. In this section the initial value for θ_g is 0.08 and the initial value for σ_g^2 is 0.008.

Figure B.1



(a) **(b)**

Figure B.2



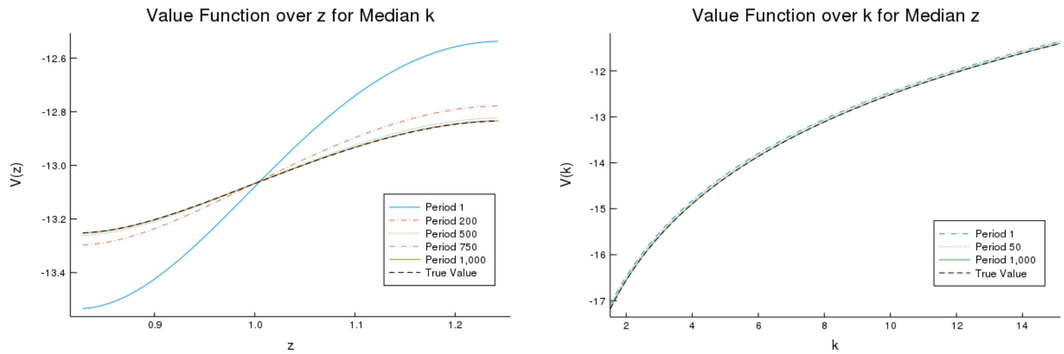
(a) **(b)**

The key differences in this specification are in the value function convergence. In this setting the slope of the value function in the z dimension changes significantly as the parameters update over time.

B.1.2

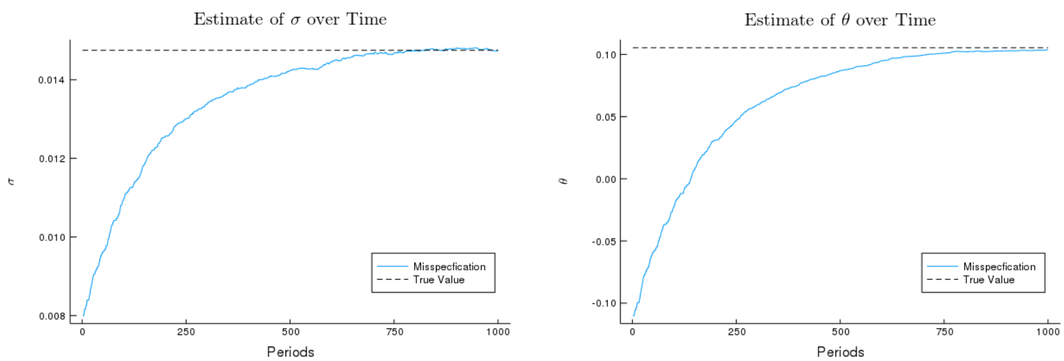
Next, we examined convergence when the initial θ value was set to a negative value and left the value for σ smaller than the true value. In this section the initial value for θ_g is -0.11 and the initial value for σ_g^2 is 0.008.

Figure B.3



(a) **(b)**

Figure B.4



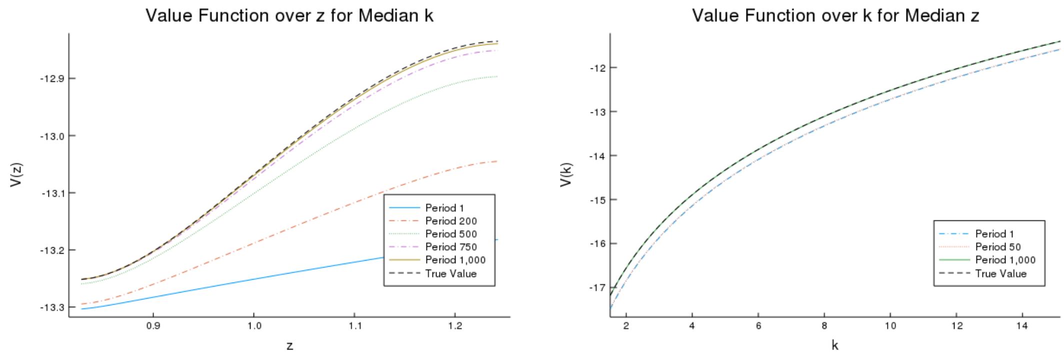
(a) **(b)**

These results were similar to the previous specification's graphs.

B.1.3

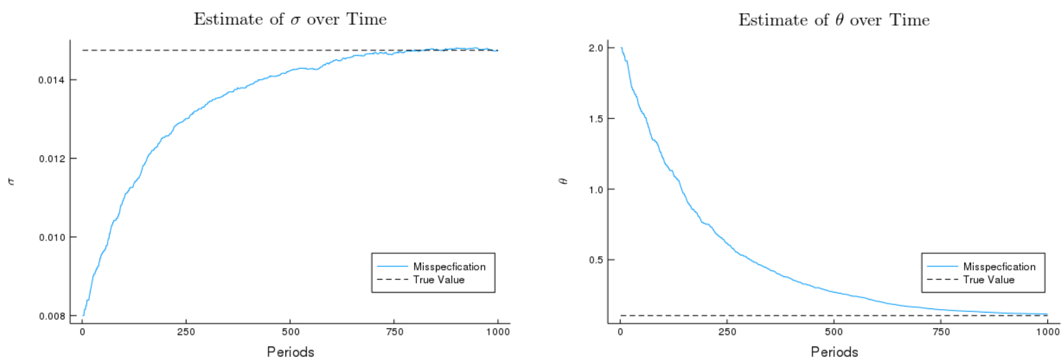
The last value tested for θ was a much larger positive value, again σ was initialized with a value smaller than the true parameter value. In this section the initial value for θ_g is 2.0 and the initial value for σ_g^2 is 0.008.

Figure B.5



(a) **(b)**

Figure B.6

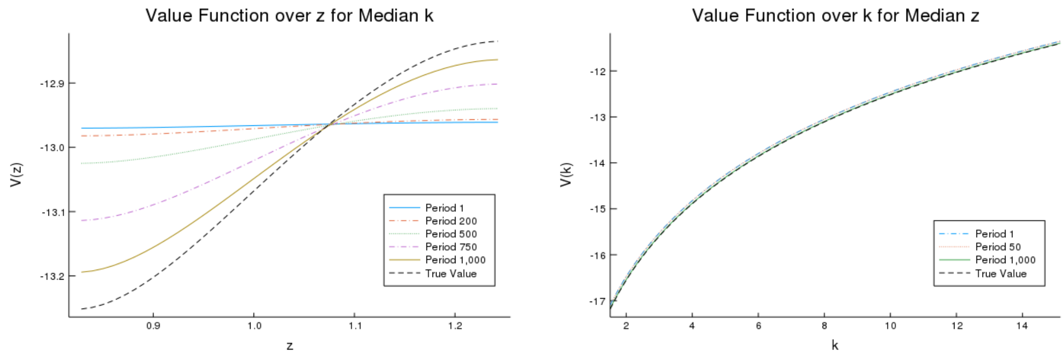


(a) **(b)**

B.1.4

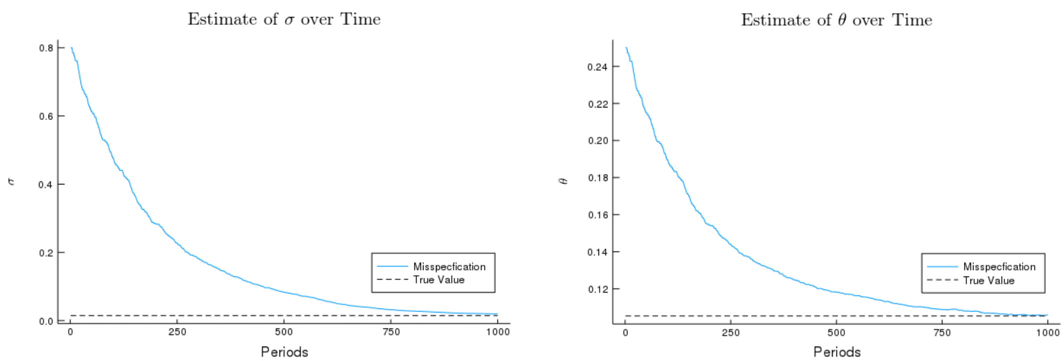
Next, different values for σ were explored. In the results below σ was set to be much higher than the original value but still less than one and θ was set to a larger value. Here the initial value for θ_g is 0.25 and the initial value for σ_g^2 is 0.8.

Figure B.7



(a) **(b)**

Figure B.8

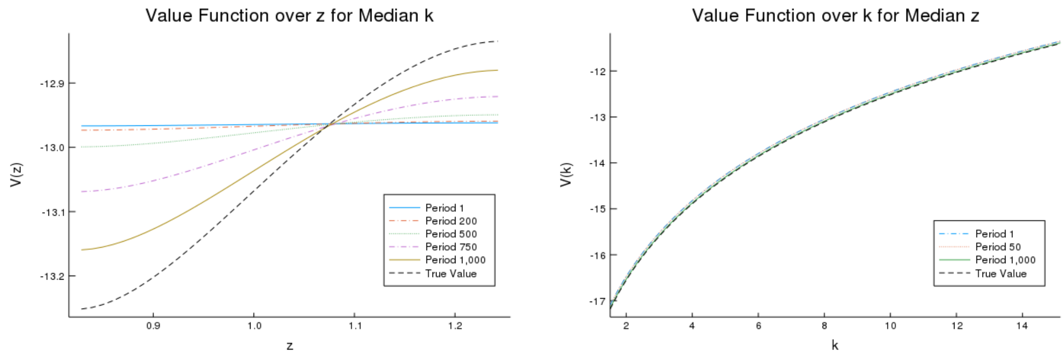


(a) **(b)**

B.1.5

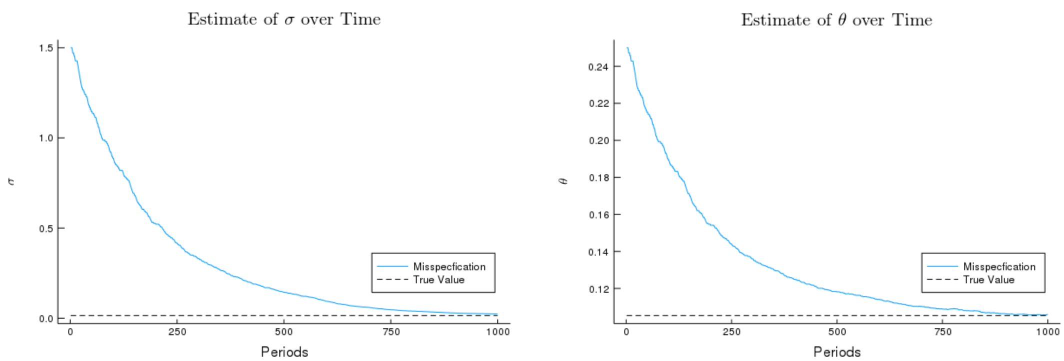
The same algorithm was run with a θ value that was much larger than the true value. In this section the initial value for θ_g is 0.25 and the initial value for σ_g^2 is 1.5.

Figure B.9



(a) **(b)**

Figure B.10

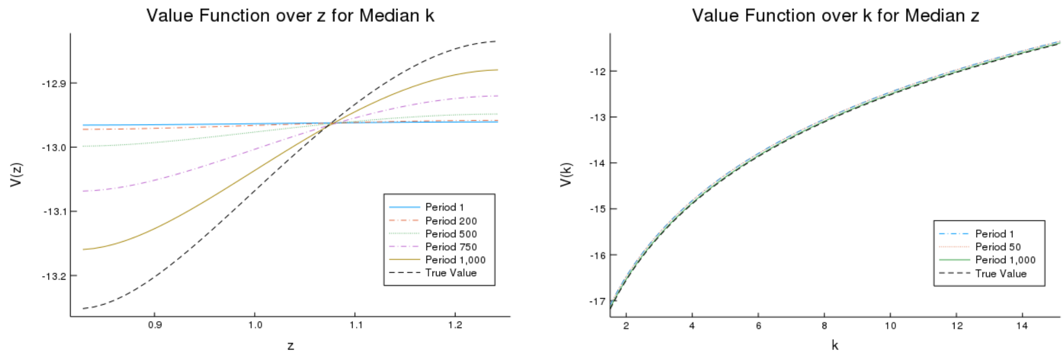


(a) **(b)**

B.1.6

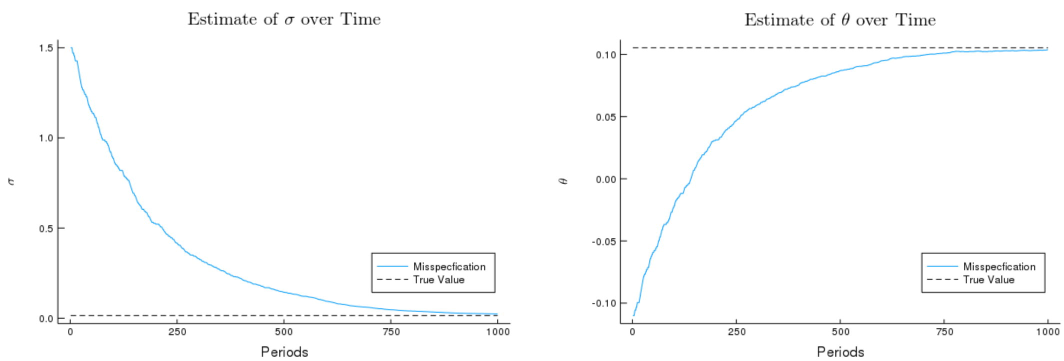
These initial values for σ were then run again with a small negative value for θ. In this section the initial value for θ_g is -0.11 and the initial value for σ_g^2 is 1.5.

Figure B.11



(a) **(b)**

Figure B.12

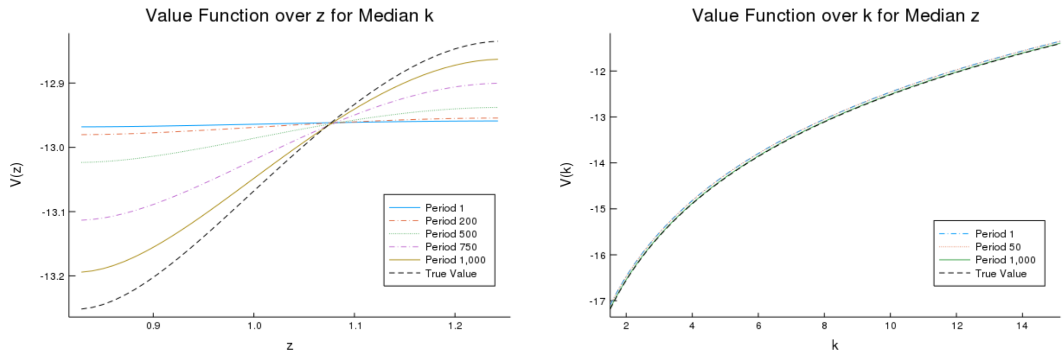


(a) **(b)**

B.1.7

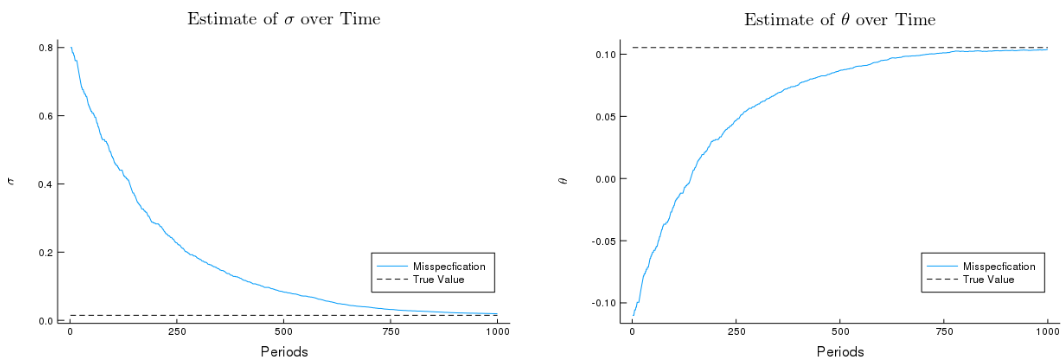
Last, we examine what would happen to this model if θ was set to be small and negative and if σ was a large value. In this section the initial value for θ_g is -0.11 and the initial value for σ_g^2 is 0.8.

Figure B.13



(a) (b)

Figure B.14



(a) (b)

B.2 Learning the Process for Capital

Section five examined converge when σ_g was set to a lower initial value than the true parameter value, 0.5. In the following section we will explore different initial values for σ_g with varying signs and magnitudes. Below is a table of the initial values we will examine.

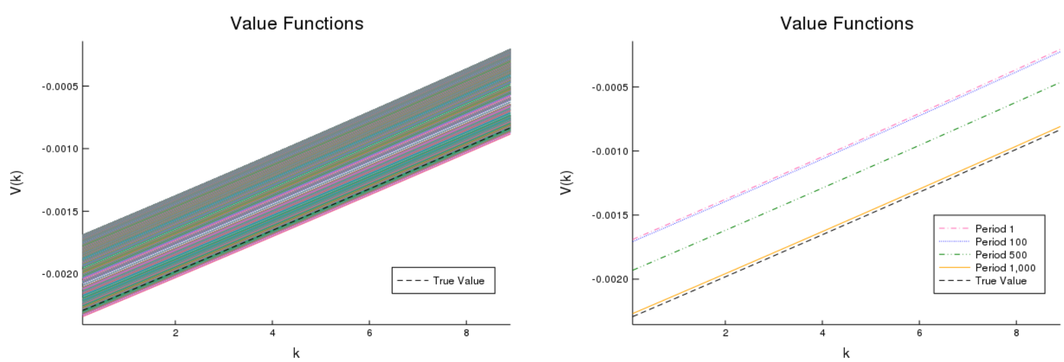
Table B.2
Initial values for σ

Specification	σ_g
Section 5	0.02
B.2.1	-0.02
B.2.2	8.0
B.2.3	-4.0

B.2.1

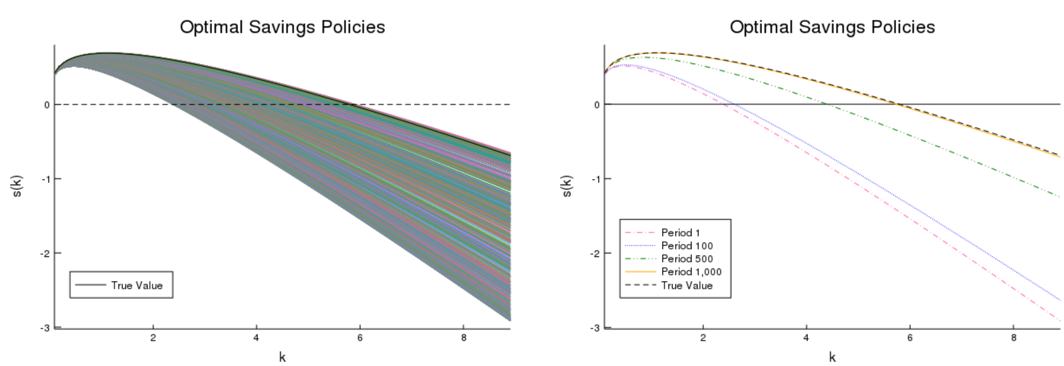
In our first alternative misspecification we look at an initial value of σ_g that is the same magnitude as the correct value, but the incorrect sign. In this section $\sigma_g = -0.02$.

Figure B.15



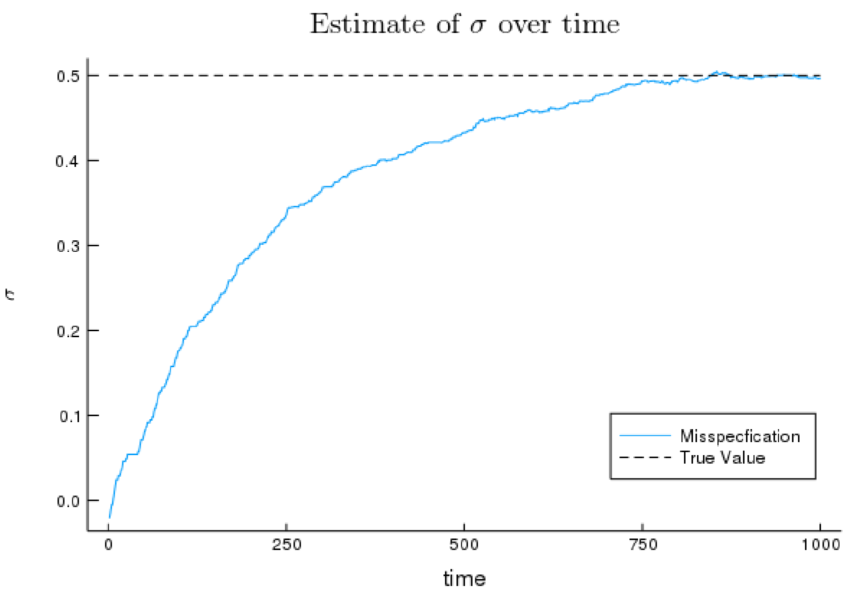
(a) (b)

Figure B.16



(a) (b)

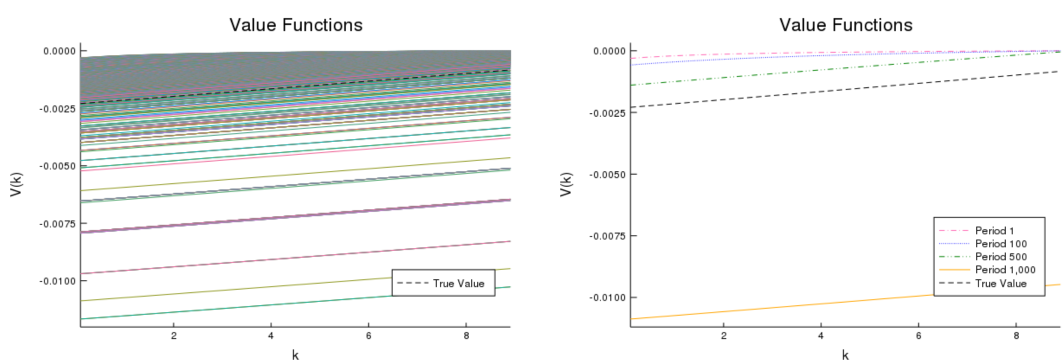
Figure B.17



B.2.2

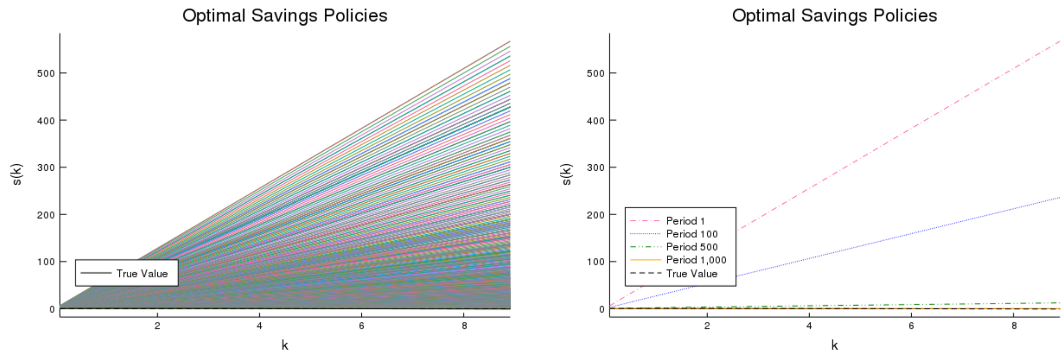
Next, we examine what would happen if the agents initial specification were much larger than the true value. Here the initial σ_g is 8.0.

Figure B.18



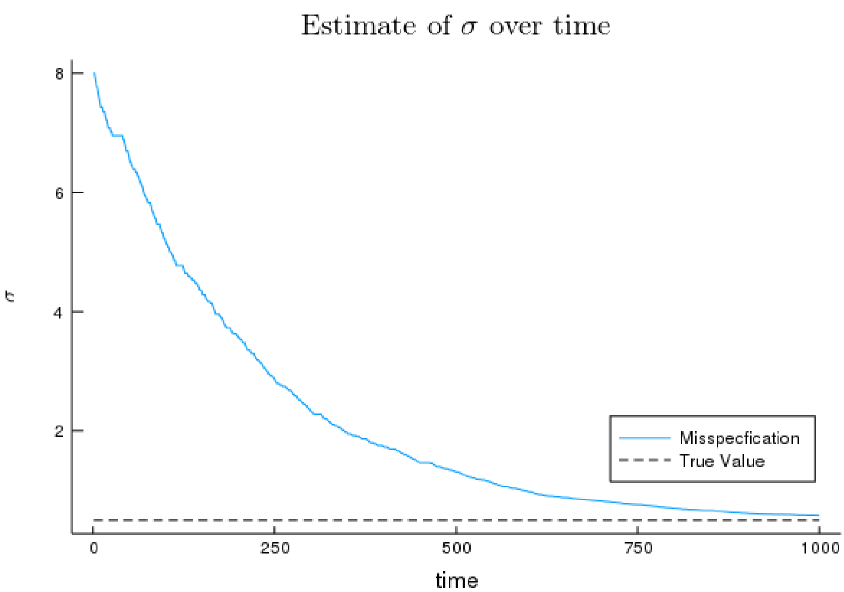
(a) **(b)**

Figure B.19



(a) **(b)**

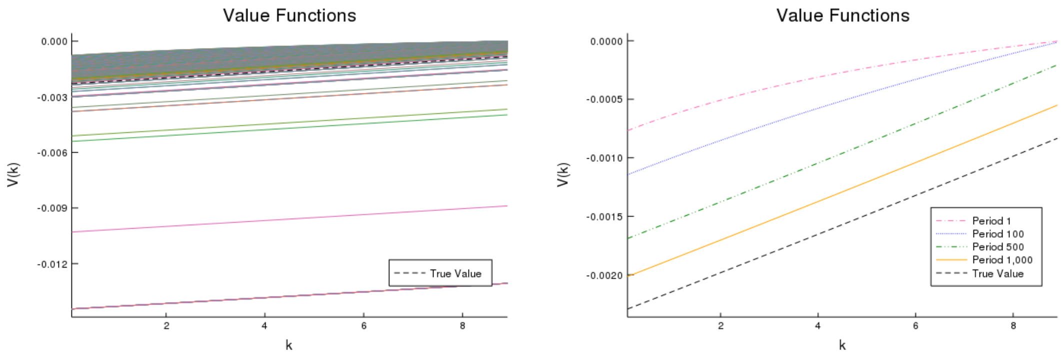
Figure B.20



B.2.3

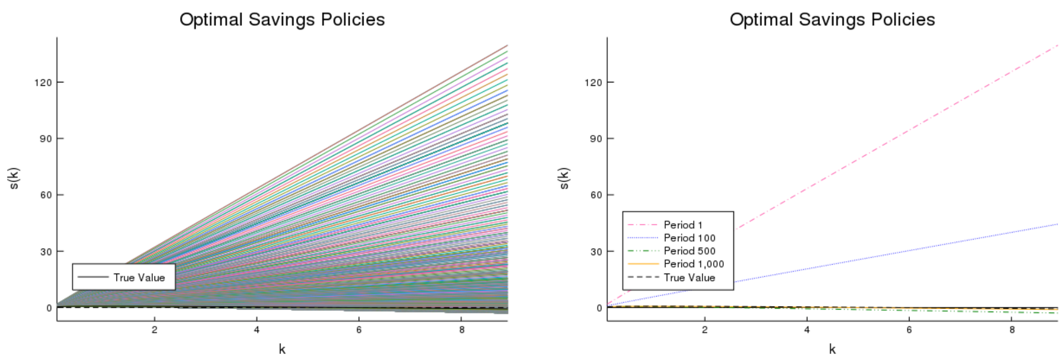
Last, we set the initial value for σ so that it is negative and has a large magnitude, $\sigma_g = -4.0$.

Figure B.21



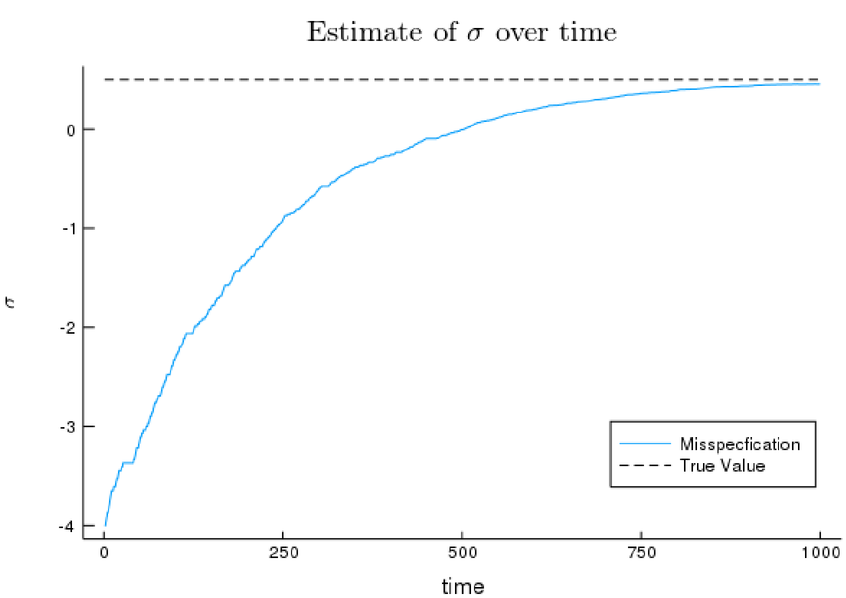
(a) (b)

Figure B.22



(a) (b)

Figure B.23



APPENDIX C
ALGEBRAIC RICCATI EQUATION SOLUTIONS

To verify the convergence of (III.4), (III.12), and (III.13) a simple univariate system was tested. In this test case, $A = 0$, $B = 1$, $R = 2$, $Q = 1$, $\beta = .95$, and $\rho = -\ln \beta$ (for consistency between the continuous and discrete discount rates). Below, is a table comparing the results of the iterative methods to output from MATLAB's built-in functions for solving AREs, icare for continuous systems and idare for discrete ones.

Table C.1
Iterative Scheme Results

Iterative Scheme	Iterative Solution	MATLAB Solution	Difference
Equation (III.4)	2.0000	2.0004	4.1670e-04
Equation (III.12)	1.3887	1.3894	6.3507e-04
Equation (III.13)	1.3887	1.3894	6.3507e-04

As table C.1 shows the results from the iterative schemes are fairly close to the standard MATLAB solutions.[1] Additionally, (III.12) and (III.13) output identical solutions in our simple case and should be able to be used interchangeably.

[1]The iterative solutions were found using julia not MATLAB. This may contribute to the difference between the iterative solutions and MATLAB functions as julia and MATLAB round differently.

APPENDIX D
OLRP WITH FEWER SYMMETRY ASSUMPTIONS

Here we outline a continuous-time optimal linear regulator problem without symmetry assumptions. In this section we revisit the continuous-time problem in section III.3 and relax the assumption that the matrix A is symmetric. In this setting the an agent faces the following optimization problem,

$$V(x_0) = \max -\mathbb{E}\int_{t=0}^{\infty} e^{-\rho t}\{x_t' R x_t + u_t' Q u_t + 2x_t' W u_t\}dt. \tag{D.1}$$

Where the state of the system, x_t, evolves according to,

$$dx_t = Ax_t dt + Bu_t dt + CdW_t \tag{D.2}$$

here dW_t is the increment of the Wiener process. The HJB for this problem can be found similarly to (III.8). For this system, the HJB will be,

$$\rho V(x) = \max_u \; -x'Rx - u'Qu - 2x'Wu + \mathbb{E}\bigg(V_x(x)\dot{x} + \frac{1}{2}V_{xx}(x)\dot{x}^2\bigg). \tag{D.3}$$

In this setting the value function takes the form (Hansen and Sargent, 2013),

$$V(x) = -x'Px - \xi$$

where ξ does not depend on the state or control variables. Plugging the proposed value function into (D.3) yields,

$$\rho x'Px + \rho\xi = x'Rx + u'Qu + 2x'Wu + x'P(Ax+Bu)(Ax+Bu)'Px + P(CC'). \quad \text{(D.4)}$$

This yields the following policy for u,

$$u = -(Q')^{-1}(W + PB)'x = -Fx. \quad \text{(D.5)}$$

Now, plugging this policy into (D.4) and rewriting the result in a general form produces,

$$\rho P = R + F'QF - 2WF + PA + A'P - PBF - F'B'P \quad \text{(D.6)}$$

$$\rho\xi = PCC'. \quad \text{(D.7)}$$

This is similar to the discrete stochastic case discussed in Hansen and Sargent (2013). The steady-state solution for this system can be found similarly to the system in section III.2.1 using the following iterative scheme

$$P_i = -(I_n \otimes \tilde{A}' + \tilde{A}' \otimes I_n)^{-1}\text{vec}(\tilde{F}_i'Q^{-1}\tilde{F}_i + R - 2W\tilde{F}_i)$$

$$\xi_i = \rho^{-1}trace(P_{i-1}CC'),$$

where $\tilde{A}_i = (A - B\tilde{F}_i - .5\rho)$ and $\tilde{F}_i = (Q')^{-1}(W + P_{i-1}B)'$.

APPENDIX E

AN ADDITIONAL DERIVATION OF CONTINUOUS-TIME RLS

We can also derive RLS more rigorously starting from a discretized version of the model. The discretized version of our model with an undetermined time step Δ is,

$$
\begin{aligned}
\theta_{t+\Delta} &= \theta_t \\
y_t &= \theta_t' x_t + e_t
\end{aligned}
$$

Where, the covariance matrix for $e_t \sim N(0, \frac{1}{\Delta})$ as in Lewis et al. (2007). First, we can examine the gain term in (III.25). Writing (III.25) in this setting we'll have,

$$
\begin{aligned}
L_t &= \mathcal{P}_{t-\Delta} x_t [(1/\Delta) + x_t \mathcal{P}_{t-\Delta} x_t']^{-1} \\
&= \mathcal{P}_{t-\Delta} x_t \Delta [1 + x_t \mathcal{P}_{t-\Delta} x_t' \Delta]^{-1}.
\end{aligned}
$$

Dividing through by Δ and then taking the limit as $\Delta \to 0$ we get,

$$K = \lim_{\Delta \to 0} \frac{1}{\Delta} L_t = \mathcal{P}_t x_t \tag{E.1}$$

Next, if we look at (III.26) we can rewrite this equation as,

$$
\begin{aligned}
\mathcal{P}_t - \mathcal{P}_{t-\Delta} &= -\mathcal{P}_{t-\Delta} x_t x_t' \mathcal{P}_{t-\Delta} [(1/\Delta) + x_t \mathcal{P}_{t-\Delta} x_t']^{-1} \\
&= -\mathcal{P}_{t-\Delta} x_t x_t' \mathcal{P}_{t-\Delta} \Delta [1 + x_t \mathcal{P}_{t-\Delta} x_t' \Delta]^{-1}.
\end{aligned}
$$

Dividing through by Δ and taking the limit as $\Delta \rightarrow 0$,

$$\frac{d\mathcal{P}_t}{dt} = -\mathcal{P}_t x_t x_t' \mathcal{P}_t = -K x_t' \mathcal{P}_t.$$

Last, we can derive the continuous-time estimate updating equation (III.24). Rewriting this equation and diving through by Δ yields,

$$\frac{1}{\Delta}\left(\hat{\theta}_t - \hat{\theta}_{t-\Delta}\right) = \frac{1}{\Delta} L_t [y_t - \hat{\theta}'_{t-\Delta} x_t].$$

Limiting this as $\Delta \rightarrow 0$ we get,

$$\frac{d\hat{\theta}_t}{dt} = K[y_t - \hat{\theta}'_t x_t]. \tag{E.2}$$

These equations we have just derived are the same as the Kalman filter equations in (III.38)-(III.40).

APPENDIX F

THE DISCRETE-TIME MODEL

We examine decision making under bounded rationality using a real business cycle model with taxation on wages and capital. In this RBC model households maximize their utility according to the following function of consumption and labor,

$$V(k_0, z_0) = \max_{c_t, k_{t+1}, h_t} \mathbb{E}_t \sum_{t=0}^{\infty} \beta^t \left\{ \frac{c_t^{1-\sigma}}{1-\sigma} - \chi \frac{h_t^{1+\varphi}}{1+\varphi} \right\}. \tag{F.1}$$

This maximization problem is subject to the following constraints on consumption and capital accumulation

$$c_t + k_{t+1} = Ak_t^{\alpha}(z_t h_t)^{1-\alpha} - \delta k_t \tag{F.2}$$

$$k_{t+1} = (1-\delta)k_t + i_t. \tag{F.3}$$

Firms in this economy seek to maximize profits according to their costs and production capabilities with a Cobb-Douglas production function, $f(k_t, z_t h_t) = Ak_t^{\alpha}(z_t h_t)^{1-\alpha}$. Productivity, z_t, evolves according to

$$\log(z_t) = \theta_z \log(z_{t-1}) + \varepsilon_t^z. \tag{F.4}$$

and $\varepsilon_t^z \sim N(0, \sigma_z^2)$. The LQ format necessary for implementing SP-learning in our social planner's problem must be linearized about the steady state, thus we must first find the non-stochastic steady state of the system. We use these steady state values to build the LQ version of the model by recasting the objective function to depend solely on state and choice variables than then re-writing this new objective function as a second degree Taylor expansion about the system's steady state (Ljungqvist and Sargent, 2012).

To use the LQ framework we want need the RBC model in the following form

$$V(x_0) = \max_{x_t, u_t} \mathbb{E}_t \sum_{t}^{\infty} \beta^t r(x_t, u_t).$$

where u_t is a vector of the agent's choice variables and x_t is a vector of state variables. These state variables evolve according to the following process,

$$x_{t+1} = Ax_t + Bu_t + \varepsilon_t$$

Reformatting the problem is accomplished using the modified equation for consumption. Using this, the objective function depends solely on capital, labor, investment, and technology

$$r(x_t, u_t) = \frac{1}{1-\sigma}[Ak_t^{\alpha}(z_t h_t)^{1-\alpha} - i_t - g_t]^{1-\sigma} - \chi \frac{h_t^{1+\varphi}}{1+\varphi}.$$

The vectors x_t and u_t contain the state and control variables for the system respectively—$x_t = (1, k_t, \log(z_t), g_t)'$ and $u_t = (h_t, i_t)'$. Now that our maximization problem is rewritten to depend on x_t and u_t, we use a second order linear approximation of $r(x_t, u_t)$ about the non-stochastic steady state to reformat the maximization problem.

The second-order Taylor expansion about the steady-state where $\bar{x}$ and $\bar{u}$ are the steady-state values of x and u, can be found using automatic differentiation to compute the partial derivatives of $r(x, u)$. Once this is complete the problem is easily reformatted into a linear-quadratic optimization problem,

$$V(x_0) = \max_{u_t} \; -\mathbb{E}_t \sum_{t=0}^{\infty} \beta^t (\hat{x}_t' R \hat{x}_t + \hat{u}_t' Q \hat{u}_t + 2\hat{x}_t' W \hat{u}_t)$$

where the state variables evolve according to

$$\hat{x}_{t+1} = A\hat{x}_t + B\hat{u}_t + C\varepsilon_t$$

here $\hat{x}_t = x_t - \bar{x}$ and $\hat{u} = u_t - \bar{u}$.

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & (1-\delta) & 0 \\ 0 & 0 & \theta_z \end{bmatrix} \quad B = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \quad C = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

The matrices that define the objective function—R, Q and $W-$ will be the same as before. These matrices combined with the matrices that define the state variables' evolution—A, B, and C—can solve the value function problem for the system above

$$V(\hat{x}_t, \hat{u}_t) = -\hat{x}_t' R x_t - \hat{u}_t' Q \hat{u}_t - 2\hat{x}_t' W \hat{u}_t + \beta \mathbb{E}_t V(x_{t+1}, u_{t+1}).$$

To get a closed-form solution to this problem we posit that the value function takes the form $V(x_t) = -\hat{x}_t P \hat{x}_t - \xi$, where P is a positive semi-definite matrix that summarizes the evolution of value function Hansen and Sargent (2013). Thus, we can rewrite the problem above as

$$-\hat{x}_t P \hat{x}_t - \xi = -\hat{x}_t' R \hat{x}_t - \hat{u}_t' Q \hat{u}_t - 2\hat{x}_t' W \hat{u}_t - \beta(A\hat{x}_t + B\hat{u}_t)' P (A\hat{x}_t + B\hat{u}_t) - \beta \text{trace}(PCC') - \beta\xi.$$

To simplify this system we eliminate $\hat{u}$ by taking the first-order condition with respect to $\hat{u}$, this yields our policy function

$$\hat{u}_t = -(Q + \beta B' P B)^{-1} (\beta B' P B + W') \hat{x}_t = -F\hat{x}_t$$

Next, using a well-established algorithm we can use the matrices above to calculate the matrix P that summarizes the evolution of the value function. In this stochastic discrete-time setting this algorithm will take the form,

$$P_{j+1} = R + \beta A' P_j A - (\beta A' P_j B + W')(Q + \beta B' P_j B)^{-1}(\beta B' P_j A + W) \tag{F.5}$$

$$\xi_{j+1} = \beta(1-\beta)^{-1}\text{trace}(P_{j+1}CC') \tag{F.6}$$

the subscript j represents iterations of the recursive solution method and P_0 is set exogenously.

F.0.1 Shadow-Price Learning in the Discrete Model

The iterative solution method outlined in the previous section, provides more information about the system than simply the solution. In the recursive algorithm outlined in (F.5) and (F.6) an initial guess or perception of the equilibrium in these equations maps to an updated perception of the value function matrix.

We can describe this mapping between perceptions and actuality using an adaptive learning tool called the T-map. The T-map is constructed by examining the link between agents' perceptions and the updated value function that results from these perceptions. The T-map is derived by examining the induced value functions for perception, $V^P(x) = -xT(P)x$. For the discrete non-stochastic case ($C = 0$) the value function induced by a perceived matrix P is

$$V^P(x) = \max_u -(x'Rx + u'Qu + 2x'Wu) - \beta(Ax + Bu)'P(Ax + Bu).$$

Once we characterize agent's control decision we can then describe the T-map, $T(P)$. In the discrete setting the control decision will take the following form,

$$F(P) = (Q + \beta B'PB)^{-1}(\beta B'PA + W')$$

using this we can rewrite the induced value function for perceptions as

$$T(P) = R + \beta A'PA - (\beta A'PB + W)'(Q + \beta B'PB)^{-1}(\beta B'PA + W').$$

This is the mapping between agent's perceptions and actuality in this model. The fixed point of the T-map is the unique steady-state solution for our system (Evans and McGough, 2018).

We, as in the continuous-time case, impose a linear forecasting rule for μ_t since long the optimal path $\mu_t^* = -2P^* x_t$. For additional simplification we assume that the agent forecasts a matrix H instead of $-2P^*$; thus

$$\mu_t = Hx_t + \varepsilon_t^\mu \tag{F.7}$$

This forecasting rule is what our agent believes at time t, the rule acts as a perceived law of motion (PLM). Our agent wants to develop a forecast of future prices using this linear relationship and their beliefs about transition matrix for the state variables A,

$$\mathbb{E}_{t+1}\mu_{t+1} = H\mathbb{E}_{t+1}(x_{t+1}) = H(\tilde{A}x_t + Bu_t)$$

in this forecast $\tilde{A}$ represents the agent's estimation of A. When the agent uses this estimate in their decision making they will estimate the following policy rule and shadow-price parameters

$$u = (2Q - \beta B'HB)^{-1}(\beta B'H\tilde{A} - 2W') = F^{SPD}(H, \tilde{A}, B)x \tag{F.8}$$

and

$$\mu = \Big(-2R - 2WF^{SPD}(H, \tilde{A}, B) + \beta \tilde{A}' H \big(\tilde{A} + BF^{SPD}(H, \tilde{A}, B) \big) \Big) x \qquad \text{(F.9)}$$
$$= T^{SPD}(H, \tilde{A}, B)$$

Equation (F.9) defines the T-map for our learning rule, this maps the agent's perceived law of motion to the actual law of motion for the system. In our models the agent takes in more information over time using new data. The basic forecasting model the agent implement is,

$$x_{t+1} = A_t x_t + Bu_t + \epsilon_t^x$$
$$\mu_t = H_t x_t + \epsilon_t^\mu$$

where ϵ_t^μ and ϵ_t^μ are error terms. The agent updates their estimates of A_t and H_t using this new information. Below is a dynamic system describing how the agent estimates A_t and H_t, and how this estimations evolve over time under bounded rationality

(Evans and McGough, 2018).

$$
\begin{aligned}
x_t &= Ax_{t-1} + Bu_{t-1} + C\varepsilon_t \\
\mathcal{R}_t &= \mathcal{R}_{t-1} + \gamma_t(x_t x_t' - \mathcal{R}_{t-1}) \\
H_t' &= H_{t-1} + \gamma_t \mathcal{R}_{t-1}^{-1} x_{t-1}(\lambda_{t-1} - H_{t-1}x_{t-1})' \\
A_t' &= A_{t-1} + \gamma_t \mathcal{R}_{t-1}^{-1} x_{t-1}(x_t - Bu_{t-1} - A_{t-1}x_{t-1})' \\
u_t &= -F^{SPD}(H_t, A_t, B)x_t \\
&= (2Q - \beta B'HB)^{-1}(\beta B'HA_t - 2W')x_t \\
\mu_t &= T^{SPD}(H_t, A_t, B)x_t \\
&= \Big(-2R - 2WF^{SPD}(H_t, A_t, B) + \beta A_t' H\big(A_t + BF^{SPD}(H_t, A_t, B)\big)\Big)x_t \\
\gamma_t &= \kappa(t+N)^{-\nu}.
\end{aligned}
\tag{F.10}
$$

Here $\mathcal{R}_t$ is a measurement for the second moment of the state variable observations x_t and γ_t is a standard gain sequence. For our purposes we will use a constant gain thus $\kappa = 0.01$ and $\nu = 0$.

Learning Results

The algorithm in (F.10) was applied to a misspecified version of the RBC model outlined in the beginning of this section and a simplified version of the RBC model without government spending or taxation. For both misspecifications, the initial H and A matrices were set as identity matrices, and R was set to fifty times an identity matrix. The initial x and u observations were set near their steady-state values, despite being in deviation from steady-state form. In the discrete-time RBC model we used typical parameter values for the many of the model parameters, the parameter χ was set so that the portion of hours worked in the non-stochastic steady state was 33% (Hansen, 1985). Below is a table summarizing our parameter values.

Table F.1
Discrete-Time Parameter Values

Description		Value
A	Total Factor Productivity	1.0
β	Discount factor	0.99
σ	Intertemporal elasticity of subst.	1.0 (log utility)
φ	Frisch elasticity of labor supply	-1.0 (log utility)
χ	Disutility of labor	1.75
α	Capital share	1/3
δ	Depreciation rate	0.025
θ	Drift parameter for tech.	0.895
σ_z	Standard Deviation for tech.	0.01

The agent in this setting understands the basic structure of the transition matrix A and does not use the constant in estimating parameters, instead they estimate coefficients for the processes governing technology and capital using only relevant data. Similar results can be achieved when the agent uses the full set of regressors. Since we use constant gain, the agent's forecast of these parameters oscillates around their rational expectations equilibrium (REE) value, since the agent places equal weight on the information gained from all observations.

The simple model without government spending was run for $50,000$ discrete time periods, at the end of which subtracting the shadow-price parameter matrix from its REE counterpart results in matrix with a norm of 2.42.

REFERENCES CITED

Aadland, D. (2001). High Frequency Real Business Cycles. *Journal of Monetary Economics*, 48(2):271 – 292.

Achdou, Y., Buera, F., Lasry, J., Lions, P., and Moll, B. (2014). Partial Differential Equation Models in Macroeconomics. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 372(2028).

Achdou, Y., Han, J., Lasry, J., Lions, P., and Moll, B. (2020). Income and Wealth Distribution in Macroeconomics: A Continuous-Time Approach. *Review of Economic Studies*.

Ahn, S., Kaplan, G., Moll, B., Winberry, T., and Wolf, C. (2018). When Inequality Matters for Macro and Macro Matters for Inequality. *NBER Macroeconomics Annual*, 32(1).

Aït-Sahalia, Y. (2002). Maximum Likelihood Estimation of Discretely Sampled Diffusions: A Closed-Form Approximation Approach. *Econometrica*, 70:202–262.

Aït-Sahalia, Y. (2004). Disentangling Diffusion from Jumps. *Journal of Financial Economics*, 74:487–528.

Aït-Sahalia, Y. (2010). Estimating Continuous-Time Models with Discretely Sampled Data. *Advances in Economics and Econometrics, Theory and Applications, Ninth World Congress*.

Amman, H. M. and Kendrick, D. A. (1999). Should Macroeconomic Policy Makers Consider Parameter Covariances? *Computational Economics*, 14(3):263–267.

Anderson, B. D. and Moore, J. B. (2007). *Optimal Control Linear Quadratic Methods*. Dover.

Benigno, P. and Woodford, M. (2004). Optimal Monetary and Fiscal Policy: A Linear-Quadratic Approach. In *NBER Macroeconomics Annual 2003, Volume 18*, NBER Chapters, pages 271–364. National Bureau of Economic Research, Inc.

Benigno, P. and Woodford, M. (2006). Optimal Taxation in an RBC Model: A Linear-Quadratic Approach. *Journal of Economic Dynamics and Control*, 30(9-10):1445–1489.

Benigno, P. and Woodford, M. (2012). Linear-Quadratic Approximation of Optimal Policy Problems. *Journal of Economic Theory*, 147(1):1–42.

Bergstrom, A. (1984). Continuous Time Stochastic Models and Issues of Aggregation over Time. In Griliches, Z. and Intriligator, M., editors, *Handbook of Economics*, volume 2, chapter 2, pages 1145–1212. Amsterdam: Elsveier.

Bergstrom, A. (1993). Survey of Continuous Time Econometrics. In Barnett, William A. and Giancarlo, Gandolfo and Hillinger, Clauda, editor, *Dynamic Disequilibrium Modeling: Theory and Applications*, chapter 1, pages 3–25. Cambridge University Press.

Bhamra, H. and Uppal, R. (2014). Asset Prices with Heterogeneity in Prices. *The Review of Financial Studies*.

Black, F. and Scholes, M. (1973). Pricing of Options and Corporate Liabilities. *Journal of Political Economy*, 81(3):637–654.

Branch, W. and Evans, G. (2006). A Simple Recursive Forecasting Model. *Economics Letters*, 91(2):158.

Branch, W. and McGough, B. (2011). Business Cycle Amplification with Heterogeneous Expectations. *Economic Theory*, 47(2):395–421.

Bray, M. (1982). Learning, Estimation, and the Stability of Rational Expectations. *Journal of Economic Theory*, 26:318–339.

Brock, W. and Mirman, L. (1972). Optimal Economic Growth and Uncertainty: The Discounted Case. *Journal of Economic Theory*, 4(3):479–513.

Brunnermeier, M. K. and Sannikov, Y. (2014). A Macroeconomic Model with a Financial Sector. *American Economic Review*, 104(2):379–421.

DeMarzo, P. and Sannikov, Y. (2006). Optimal Security Design and Dynamic Capital Structure in a Continuous-Time Agency Model. *Journal of Finance*, 61(6):2681–2724.

Dixit, A. (1989). Entry and Exit Decisions under Uncertainty. *Journal of Political Economy*, 97(3):620–638.

Dixit, A. (1992). The Art of Smooth Pasting. STICERD - Theoretical Economics Paper Series, Suntory and Toyota International Centres for Economics and Related Disciplines, LSE.

Dixit, A. and Pindyck, R. (1994). *Investment under Uncertainty*. Princeton University Press.

Dorfman, R. (1969). An Economic Interpretation of Optimal Control Theory. *The American Economic Review*, 59(7):817 – 831.

Duffie, D. and Epstein, L. (1992). Stochastic Differential Utility. *Econometrica*, 60(2):353–394.

Eaton, J. (1981). Fiscal Policy, Inflation and the Accumulation of Risky Capital. *Review of Economic Studies*, 48(3):435–445.

Eusepi, S. and Preston, B. (2011). Expectations, Learning, and Business Cycle Fluctuations. *American Economic Review*, 101(6):2844–72.

Evans, G. and Honkapohja, S. (2009). Learning and Macroeconomics. *Annual Review of Economics*.

Evans, G. and McGough, B. (2018). Learning to Optimize. *Draft*.

Evans, G. W. and Honkapohja, S. (2001). *Learning and Expectations in Macroeconomics*. Princeton University Press.

Forsyth, P. and Labahn, G. (2007). Numerical Methods of Controlled Hamilton-Jacobi-Bellman PDEs in Finance. *Journal of Computational Finance*, 11(2):1–44.

Gabaix, X. (2009). Power Laws in Economics and Finance. *Annual Review of Economics*, 1:255–293.

Gabaix, X., Lasry, J., Lions, P., Moll, B., and Qu, Z. (2016). The Dynamics of Inequality. *Econometrica*, 84(6):2071–2111.

Goodwin, G. and Mayne, D. (1987). A Parameter Estimation Perspective of Continuous Time Model Reference Adaptive Control. *Automatica*, 23(1):57–70.

Hansen, G. (1985). Indivisible Labor and the Business Cycle. *Journal of Monetary Economics*, 16(3):309–327.

Hansen, G. and Wright, R. (1992). The Labor Market in Real Business Cycle Theory. *Quarterly Review*, 16(Spr):2–12.

Hansen, L. and Sargent, T. (1991). *Rational Expectations Econometrics*. Underground classics in economics. Westview Press. Includes bibliographical references (pages 283-293).

Hansen, L. and Sargent, T. (2013). *Recursive Models of Dynamic Linear Economies*. Princeton University Press.

Hansen, L. and Sargent, T. (2019a). Macroeconomic Uncertainty Prices when Beliefs are Tenuous. *NBER working paper*.

Hansen, L. and Sargent, T. (2019b). Structured Uncertainty and Model Misspecification. *University of Chicago, Becker Friedman Institute for Economics Working Paper No. 2018-77*.

Hansen, L., Sargent, T., Turmuhambetova, G., and Williams, N. (2006). Robust Control and Model Misspecification. *Journal of Economic Theory*, 128(1):45–90.

Hansen, L. and Scheinkman, J. (1995). Back to the Future - Generating Moment Implications for Continuous-Time Markov Processes. *Econometrica*, 63:767–804.

Hommes, C. (2011). The Heterogeneous Expectations Hypothesis: Some Evidence from the Lab. *Journal of Economic Dynamics and Control*, 35(1):1–24.

Huang, M., Caines, P. E., and Malhame, R. P. (2012). Social Optima in Mean Field LQG Control: Centralized and Decentralized Strategies. *IEEE Transactions on Automatic Control*, 57(7):1736–1751.

Huarng, K. and Yeh, C. (1992). Continuous-time recursive least-squares algorithms. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 39(10):741–745.

Kaplan, G., Moll, B., and Violante, G. (2018). Monetary Policy According to HANK. *American Economic Review*, 108(3):697–743.

Kellerhals, P. B. (2001). *Financial Pricing Models in Continuous Time and Kalman Filtering*, volume 506. Springer Science & Business Media.

Kendrick, D. A. (2005). Stochastic Control for Economic Models: Past, Present and the Paths Ahead. *Journal of Economic Dynamics and Control*, 29(1-2):3–30.

Kleinman, D. L. (1968). On an Iterative Technique for Riccati Equation Computations. *IEE Transactions on Automatic Control*, 13(1):114–115.

Lester, C. (2020). Boundedly rational decision making in continuous-time. *Working.*

Lewis, F. L. (1986). *Optimal Control*. Wiley-Interscience.

Lewis, F. L., Xie, L., and Popa, D. (2007). *Optimal and Robust Estimation: With an Introduction to Stochastic Control Theory*. CRC Press, Second edition.

Ljung, L. (1977). Analysis of Recursive Stochastic Algorithms. *IEEE Transactions on Automatic Control*, 22(4):551–575.

Ljung, L. and Söderström, T. (1983). *Theory and Practice of Recursive Identification*. MIT Press.

Ljungqvist, L. and Sargent, T. J. (2012). *Recursive Macroeconomic Theory*. MIT Press.

Marimon, R. and Sunder, S. (1993). Indeterminacy of Equilibria in a Hyperinflationary World: Experimental Evidence. *Econometrica*, 61(5):1073–1107.

Marimon, R. and Sunder, S. (1994). Expectations and Learning under Alternative Monetary Regimes: An Experimental Approach. *Economic Theory*, 4(1):131–162.

Merton, R. (1969). Lifetime Portfolio Selection Under Uncertainty - Continuous-Time Case. *Review of Economics and Statistics*, 51(3):247–257.

Merton, R. (1971). Optimum Consumption and Portfolio Rules in a Continuous-Time Model. *Journal of Economic Theory*, 3(4):373–413.

Merton, R. (1975). An Asymptotic Theory of Growth Under Uncertainty. *Review of Economic Studies*, 42(3):375–393.

Merton, R. (1992). *Continuous-Time Finance*. Macroeconomics and Finance Series. Wiley.

Mirman, L. (1973). Steady State Behavior of a Class of One Sector Growth Models with Uncertain Technology. *Journal of Economic Theory*, 6(3):219–242.

Mirrlees, J. (1966). Optimum Accumulation Under Uncertainty. *Econometrica*, 34(5S):26–&.

Mirrlees, J. (1971). *Optimum Accumulation Under Uncertainty: the Case of Stationary Returns to Investment*, pages 36–50. Palgrave Macmillan UK, London.

Mitra, K., Evans, G., and Honkapohja, S. (2013). Policy Change and Learning in the RBC model. *Journal of Economic Dynamics and Control*, 37(10):1947–1971.

Mitra, K., Evans, G. W., and Honkapohja, S. (2019). Fiscal Policy Multipliers in an RBC Model with Learning. *Macroeconomic Dynamics*, 23(1):240–283.

Parra-Alvarez, J., Posch, O., and Wang, M. (Working). Identification and Estimation of Heterogeneous Agent Models: A Likelihood Approach. *CREATES Research Paper*.

Phillips, A. W. (1959). The Estimation of Parameters in Systems of Stochastic Differential Equations. *Biometrika*, 46(1-2):67–76.

Plosser, C. I. (1989). Understanding Real Business Cycles. *Journal of Economic Perspectives*, 3(3):51–77.

Posch, O. (2009). Structural Estimation of Jump-Diffusion Processes in Macroeconomics. *Journal of Econometrics*, 153(2):196 – 210.

Posch, O., Rubio-Ramírez, J. F., and Fernández-Villaverde, J. (2011). Solving the New Keynesian Model in Continuous Time. Technical report.

Romer, D. (1996). *Advanced Macroeconomics*. Advanced Series in Economics. McGraw-Hill Companies.

Sannikov, Y. (2008). A Continuous-Time Version of the Principal-Agent Problem. *Review of Economic Studies*, 75:957–984.

Sargent, T. (1999). *The Conquest of American Inflation*. Princeton University Press.

Sastry, S. and Bodson, M. (1989). *Adaptive Control: Stability, Convergence, and Robustness*. Prentice-Hall, Inc., USA.

Stokey, N. (2009). *The Economics of Inaction: Stochastic Control Models with Fixed Costs*. Princeton University Press.

Veronesi, P. (2019). Heterogeneous Households under Uncertainty. Working Paper 25448, National Bureau of Economic Research.

Vrabie, D., Abu-Khalaf, M., Lewis, F. L., and Wang, Y. (2007). Continuous-Time ADP for Linear Systems with Partially Unknown Dynamics. In *2007 IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning*, pages 247–253.

Vrabie, D., Pastravanu, O. C., Abu-Khalaf, M., and Lewis, F. L. (2009). Adaptive Optimal Control for Continuous-Time Linear Systems Based on Policy Iteration. *Automatica*, 45:477–484.

Wang, H. and Zhou, X. Y. (2019). Continuous-Time Mean-Variance Portfolio Selection: A Reinforcement Learning Framework.

Wang, J. and Forsyth, P. (2010). Numerical Solution of the Hamilton-Jacobi-Bellman Formulation for Continuous Time Mean Variance Asset Allocation. *Journal of Economic Dynamics and Control*, 34(2):207 – 230.

Xie, S., Li, Z., and Wang, S. (2008). Continuous-time Portfolio Selection with Liability: Mean-variance model and stochastic LQ approach. *Insurance: Mathematics and Economics*, 42(3):943 – 953.