Boundedly Rational Decision Making in Continuous-Time

Chandler Lester

Preliminary

Link to Latest Version

Abstract

Continuous-time macroeconomic literature has grown remarkably in recent years. As work on continuous-time models becomes more prevalent, macroeconomists need to adapt essential discrete-time methods to continuous-time. In this work, we focus on modifying adapting learning techniques to continuous-time settings to provide a continuous-time alternative to rational expectations. Examining the adaptive learning toolkit in continuous-time proves difficult since continuous-time value function problems are traditionally solved backward from the end of time. Thus as a first step in developing adaptive learning dynamics in continuous-time, we apply shadow-price learning (SP-learning) techniques, wherein agents learn to forecast future states and shadow prices. The SP-learning environment allows for continuous-time learning since agents are learning to make optimal decisions at the system's steady-state, therefore our setting does not run into time dependency issues. To use this framework efficiently, we first need a tractable continuous-time linear-quadratic (LQ) environment to study how the agent's forecast of future states and choices their outcomes and observations. Hence, our contributions are two-fold we build a continuous-time LQ framework for solving value function problems using iterative methods and implement adaptive learning techniques in this new setting.

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 2 and 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 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 2.2 and section 4.

The paper precedes as follows. Section 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 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 4. We evaluate a simple economic model in section 5; the model used is a simple Robin Crusoe economy as in Evans and McGough (2018). Section 6 concludes.

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),

$$V(x_0) = \max_u \; -\mathbb{E}\sum_{t=0}^{\infty} \beta^t \{x_t' R x_t + u_t' Q u_t\} \tag{1}$$

where x_t evolves according to

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

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 (1) and (2), we can write the Bellman system as,

$$V(x_t) = \max_u \; \{-x_t' R x_t - u_t' Q u_t + \beta \mathbb{E} V(x_{t+1})\}. \tag{3}$$

To solve the Bellman in the LQ framework we use a guess-and-verify approach, positing that $V(x_t) = -x_t' P x_t$, 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 (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 (3) and write a recursive solution for P using our Riccati equation

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

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 (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).

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

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

where x_t evolves according to,

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

here our discount factor takes the form of an exponential, $e^{-\rho t}$. Equation (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

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

the limit as $\Delta \to 0$, the HJB becomes

$$\rho V(x) = \max_u \left(-x'Rx - u'Qu + V_x(x)\frac{dx_t}{dt} \right) \tag{7}$$

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 (7),

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

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,

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

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,

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

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 linear regulator problem,

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

[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, $\mathbb{E}[V(x_{t+\Delta})] = V_x(x)\frac{dx_t}{dt} = 2x'P(Ax + Bu)$.

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,

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

We can then rewrite this as,

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

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,

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

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 (12) and (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 $\Delta \to 0$.

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

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

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

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 (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,

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

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,

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

to get this into the typical discrete LQ format, as in (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 (6) yielding,

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

where I is an (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 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.

2.2.1 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 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.

[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 $e^{-\rho(\Delta)}$, $(1-\rho\Delta)$.

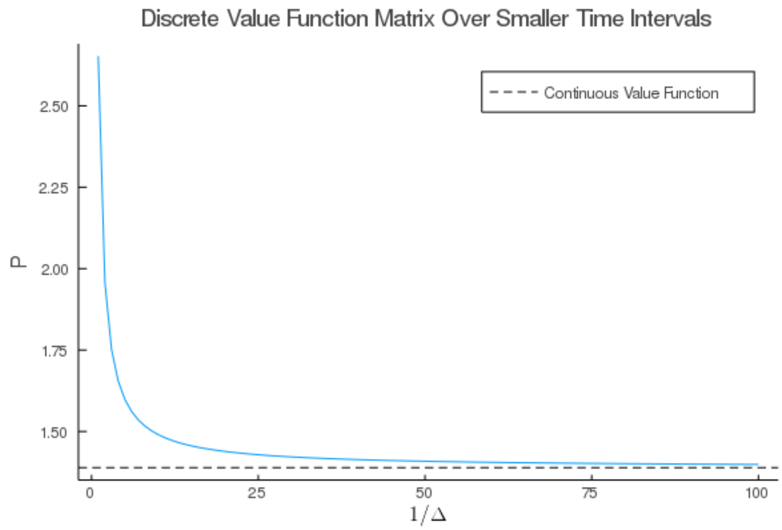


Figure 1

Figure 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.

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,

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

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

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

here dZ_t is the increment of the Wiener process[4] and A is again assumed to be symmetric. As before x_t is a $(n \times 1)$ 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 2. In the stochastic case our HJB is,

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

Note that unlike the HJB in (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 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 (19) yields,

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

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 (20). The policy function for u is

[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 $\mathbb{E}[(dZ_t)^2] = dt$

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 (20) produces,

$$\rho P = R + F'QF - 2WF + 2A'P - 2PBF$$
$$\rho\xi = PCC'.$$

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 2.1 using the following recursive scheme,

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

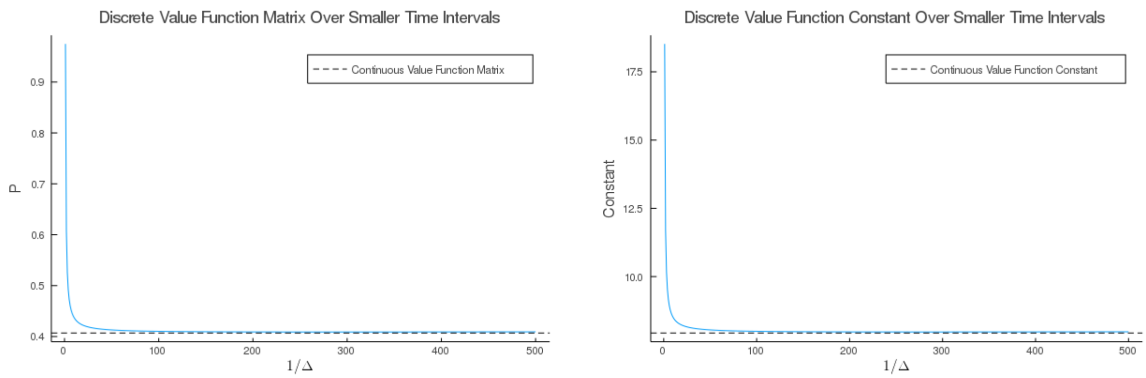
$$\xi_i = \rho^{-1}\text{trace}(P_{i-1}CC'), \tag{22}$$

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.

3.1 Convergence in the Complex Case

Before moving on, it is worth noting that under transformations similar to those in section 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 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.



(a) (b)

Figure 2

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

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).

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,

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

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,

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

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

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

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,

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

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

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 (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

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

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

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

Note the parallels between this and the system in (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,

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

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

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 (24)-(26) when $R_t = 0$ and $r_t = 1$. This particular RLS system will have a decreasing gain. The transition equation in (32) is now the transition equation for model parameters θ_t instead of data x_t, as shown in (32) the parameters in this setting are constant over time. The measurement equation in (33) is essentially the same as the measurement equation in (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 (32) and (33) yields,

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

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

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

As we can see this is equivalent to the system in (24)-(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 (27)-(28) to depend on increments of time (Δ) and recast our state transition matrix,

$$x_{t+1} = (I + A_t\Delta)x_t + w_t$$

$$y_t = \theta_t x_t + e_t$$

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 (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,

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

This is our continuous-time Kalman gain. Next, we examine (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 (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 (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 (32)-(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

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

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

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

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

$$\frac{d\mathcal{P}_t}{dt} = \frac{1}{1-\gamma}(-\mathcal{P}_t x_t' x_t \mathcal{P}_t + \gamma \mathcal{P}_t)$$

$$K = \frac{1}{1-\gamma}\mathcal{P}_t x_t'$$

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

For the decreasing gain version of the algorithm simply use $r_t = 1$ and $R_t = 0$ in equations (38)-(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.

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 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

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

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.

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

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.

4.2.1 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

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

Agent's need to select u in order to solve the value function problem in (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

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

here $\tilde{A} = A - \frac{1}{2}I\rho - BF$. Note the right-hand-side of $T(P)$ is similar to the Riccati equation (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

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

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.

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 (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,

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

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,

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

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,

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

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

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

or

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

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.

4.3.1 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 (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

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

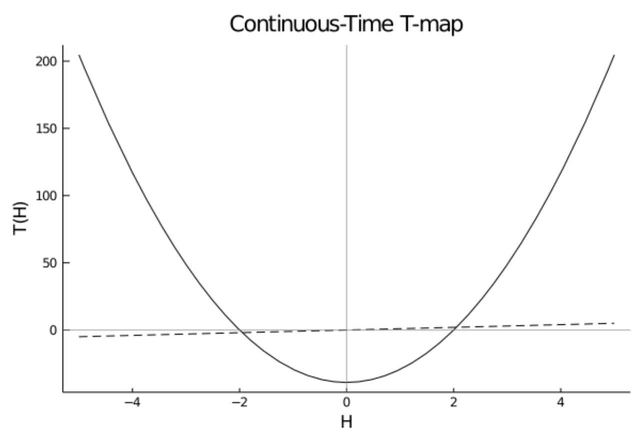


Figure 3: T-map

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 (42).

4.3.2 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 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 (42) will converge to the rational expectations equilibrium.

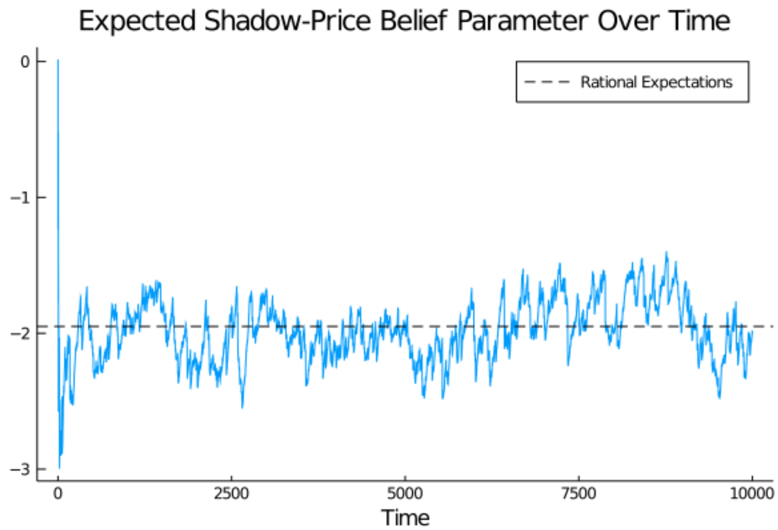


Figure 4: Univariate Continuous-Time SP-Learning

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 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),

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

$$\gamma_t = \kappa(t+N)^{-\nu}.$$

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,

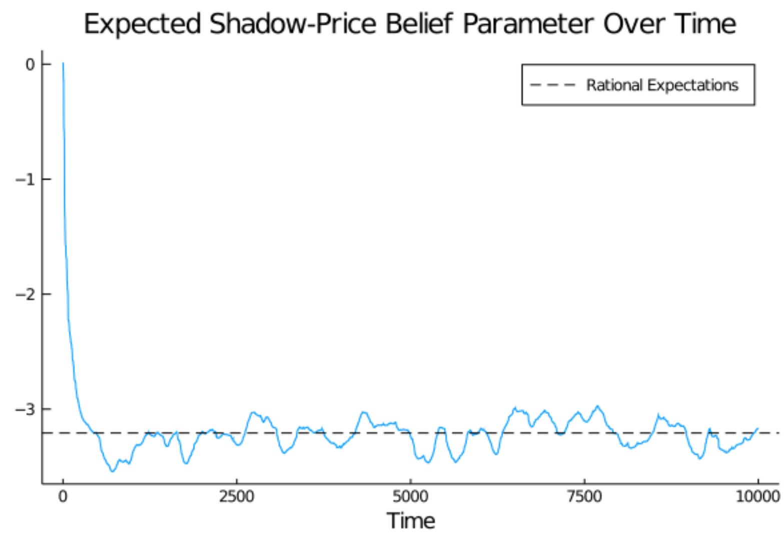


Figure 5

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.

4.3.3 Convergence in the Context of Learning

In section 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 (49) with $\gamma_t = (0.01)\Delta$ converges to the continuous-time expected shadow price parameter when Δ is sufficiently small.

Figure 6 shows how the discrete learning rule responds under the transformations in section 3 with select values of Δ.[5]

Figure 6

In figure 6 the modified discrete learning rule gradually gets closer to the continuous-time rational expectations solution as Δ gets increasingly small.

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.

[5]The learning iterations in figure 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

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

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

where the economy is subject to,

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

as before dZ_t is the increment of the Wiener process. The model we have outlined in (50) and (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 (17) and (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.

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 (42) to forecast these unknown values. As the agent gains more information they can update their parameter estimates using (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 (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 $\nu = 0$.

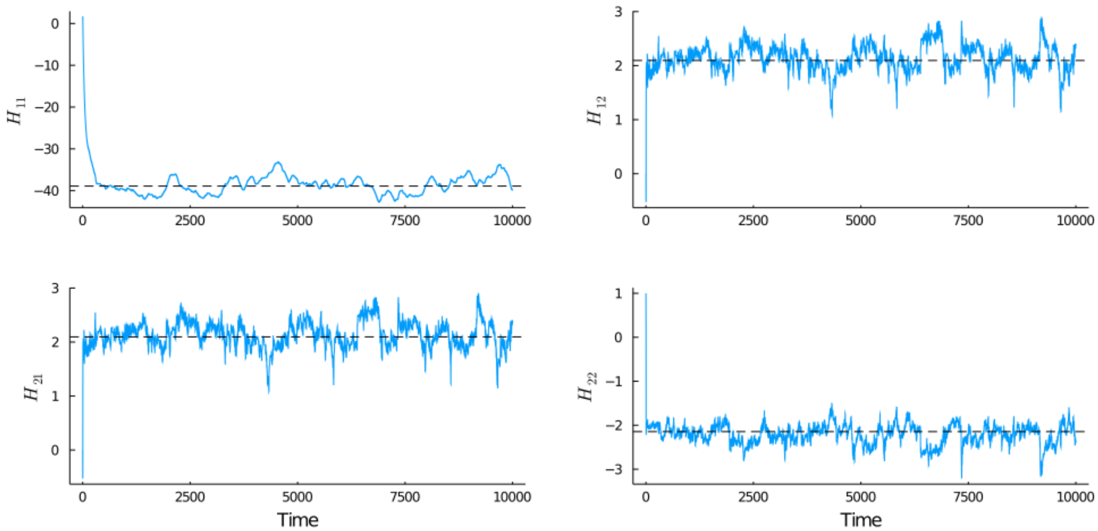


Figure 7: Expected Shadow-Price Parameters, The Continuous Case

As shown in figure 7, an agent with boundedly rational behavior modeled by (42) will be able to generate an accurate estimate of the steady-state shadow price parameters. In figure 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.

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 5.1.

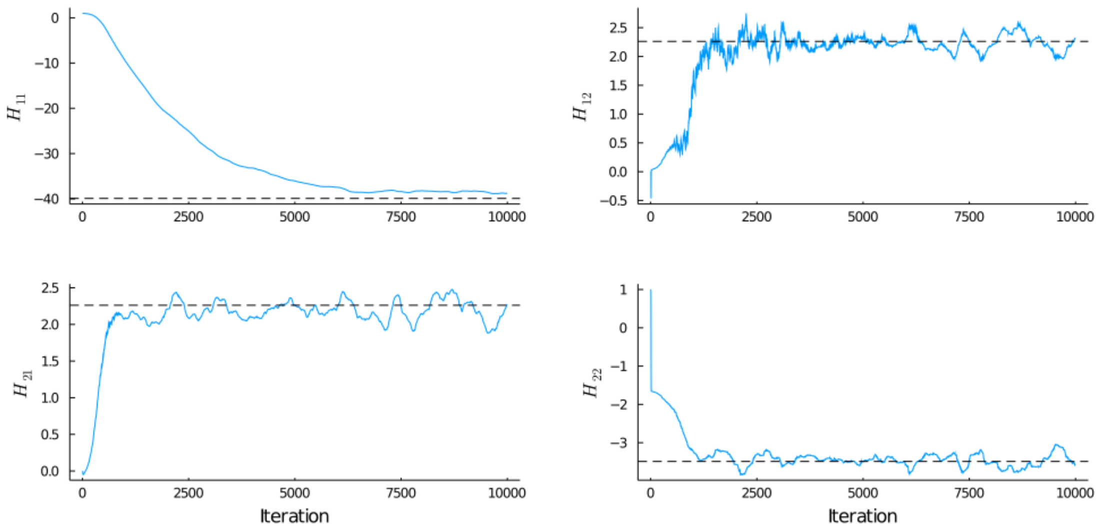


Figure 8: Expected Shadow-Price Parameters, The Discrete Case

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.

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.

A Algebraic Riccati Equation Solutions

To verify the convergence of (4), (12), and (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.

Solution Comparisons

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

Table 1: Iterative Scheme Results

As table 1 shows the results from the iterative schemes are fairly close to the standard MATLAB solutions.[6] Additionally, (12) and (13) output identical solutions in our simple case and should be able to be used interchangeably.

B 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 3 and relax the assumption that the matrix A is symmetric. In this setting the an agent faces the following optimization problem,

$$V(x_0) = \max -\mathbb{E}\int_{t=0}^{\infty} e^{-\rho t}\{x_t' R x_t + u_t' Q u_t + 2x_t' W u_t\}dt. \tag{52}$$

[6]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.

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

$$dx_t = Ax_t dt + Bu_t dt + CdW_t \tag{53}$$

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

$$\rho V(x) = \max_u \; - x'Rx - u'Qu - 2x'Wu + \mathbb{E}\bigg(V_x(x)\dot{x} + \frac{1}{2}V_{xx}(x)\dot{x}^2 \bigg). \tag{54}$$

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 (54) yields,

$$\rho x'Px + \rho\xi = x'Rx + u'Qu + 2x'Wu + x'P(Ax + Bu)(Ax + Bu)'Px + P(CC'). \tag{55}$$

This yields the following policy for u,

$$u = -(Q')^{-1}(W + PB)'x = -Fx. \tag{56}$$

Now, plugging this policy into (55) and rewriting the result in a general form produces,

$$\rho P = R + F'QF - 2WF + PA + A'P - PBF - F'B'P \tag{57}$$

$$\rho\xi = PCC'. \tag{58}$$

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

C 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,

$$\theta_{t+\Delta} = \theta_t$$
$$y_t = \theta_t' x_t + e_t$$

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 (25). Writing (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,

$$K = \lim_{\Delta \to 0} \frac{1}{\Delta} L_t = \mathcal{P}_t x_t \tag{59}$$

Next, if we look at (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 $\Delta \to 0$,

$$\frac{d\mathcal{P}_t}{dt} = -\mathcal{P}_t x_t x_t' \mathcal{P}_t = -K x_t' \mathcal{P}_t. \tag{60}$$

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

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

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

$$\frac{d\hat{\theta}_t}{dt} = K[y_t - \hat{\theta}_t' x_t]. \tag{61}$$

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

References

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. (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.

Black, F. and Scholes, M. (1973). Pricing of Options and Corporate Liabilities. *Journal of Political Economy*, 81(3):637–654.

Branch, W. A. and Evans, G. W. (2006). A simple recursive forecasting model.(report). *Economics Letters*, 91(2):158.

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.

Dixit, A. (1992). The Art of Smooth Pasting. STICERD - Theoretical Economics Paper Series, Suntory and Toyota International Centres for Economics and Related Disciplines, LSE.

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.

Gabaix, X., Lasry, Jand Lions, P., Moll, B., and Qu, Z. (2016). The Dynamics of Inequality. *Econometrica*, 84(6):2071–2111.

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.*

Hommes, C. (2011). The Heterogeneous Expectations Hypothesis: Some Evidence from the Lab. *Journal of Economic Dynamics and Control*, 35(1):1–24.

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.

Kendrick, D. (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.

Lewis, F., Xie, L., and Popa, D. (2007). *Optimal and Robust Estimation: With an Introduction to Stochastic Control Theory*. CRC Press, Second edition.

Lewis, F. L. (1986). *Optimal Control*. Wiley-Interscience.

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. (1975). An Asymptotic Theory of Growth Under Uncertainty. *The Review of Economic Studies*, 42(3):375–393.

Mirrlees, J. (1971). *Optimum Accumulation Under Uncertainty: the Case of Stationary Returns to Investment*, pages 36–50. Palgrave Macmillan UK, London.

Phillips, A. W. (1959). The Estimation of Parameters in Systems of Stochastic Differential Equations. *Biometrika*, 46(1-2):67–76.

Sargent, T. J. (1999). *The Conquest of American Inflation*. Princeton University Press.

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.