Teaching an Artificial Central Bank to Conduct Monetary Policy

Preliminary draft

Albert Flak*

April 13, 2025

Abstract

This study explores the potential of reinforcement learning to enhance monetary policy design. As an example, I simulate a calibrated, stylised, non-linear economy wherein an artificial policymaker is tasked with setting interest rates to keep inflation close to its goal as well as minimising output gaps and interest rate volatility. The artificial policymaker successfully learns a non-linear decision rule that outperforms benchmark Taylor rules, even under partial information and without knowing the DGP of the economy. The paper illustrates the entire development process, including construction of the simulation, algorithm implementation, policy evaluation and interpretation, to showcase the flexibility and usefulness of the method.

JEL classification: E37, E52, E58

Keywords: Monetary policy, reinforcement learning, Taylor rule

*University of St. Gallen, Department of Economics, Switzerland. Email: albert.flak@unisg.ch.

1 Introduction

Simple and robust policy rules have long been one of the cornerstones of monetary policy, providing central banks with straightforward guidelines on responding to variations in inflation, output and other macroeconomic variables. Their enduring utility is evident in both academic research and practical policy-making. However, when central banks design policy, they face an imperfectly understood economy, uncertainties about their models, non-linearities, partial observability of economic variables, measurement noise and many other practical challenges. Traditional methods to derive policy rules are often difficult or not flexible enough to apply when the policymaker wants to fully reflect these challenging aspects of real-world economies.

This paper proposes a novel approach to deriving robust monetary policy rules using reinforcement learning (RL), a branch of artificial intelligence that has shown remarkable success in solving complex decision-making problems in various fields, from robotics to game theory. RL is potentially well-suited for studying monetary policymaking in economies characterized by non-linearities, high dimensionality, and partial observability, where traditional optimization tools may fall short. The approach does not require precise knowledge of the economy's transition dynamics. Beyond its ability to derive linear policy ("Taylor") rules, reinforcement learning can search the space of non-linear decision rules. Instead of assuming a particular functional form, I represent the central bank's decision rule as a deep neural network capable of approximating any non-linear function.

I construct an *artificial monetary policymaker* – a central bank *digital twin* – which is tasked with learning to make interest rate decisions *from scratch* inside of a non-linear economic simulation. Replicating an extreme version of real-world uncertainty, the artificial monetary policymaker (AP) faces an economy it does not "understand" – it has no knowledge of the dynamic system generating macroeconomic data. The AP has to rely solely on experience created through the simulated trial-and-error interaction with the economy that provides performance feedback about its three main goals – keeping inflation close to its goal, eliminating output gaps and smoothing interest rate changes. Based on the experience and feedback, the digital twin of the central bank successively improves its policy.

After constructing an economic simulator and implementing a state-of-the-art RL algo-

rithm, I demonstrate that the artificial policymaker can learn useful "Taylor rules" (TR) from scratch without understanding how the economy works. I show that in a purely linear simulation, the AP learns to act in accordance with a variant of the Taylor (1993) rule. Introducing non-linearities, I show that AP is able to approximate an optimal non-linear rule. Finally, I study the properties of the AP's interest rate setting strategy in a rich simulation calibrated with US data that features the zero lower bound and a non-linear trade-off between inflation and output gaps that results from variation in asymmetric inflation expectations anchoring.

I evaluate the trained policy and analyse its properties. In the benchmark stylised economy, the AP can outperform standard Taylor rules substantially, with expected losses being reduced by $10 - 50\%$ depending on the exact setting. The trained policy appears a better predictor of historical FED policy than the original Taylor (1993) rule, having a correlation coefficient with historical FED decisions of 0.87 compared to 0.77 using real-time data. Moreover, the AP can outperform Taylor rules even when it only observes the current level of inflation and the output gap, i.e. when the natural rate of interest r_t^* is fully unobservable, demonstrating how recognising non-linearities might be more important for performance than incorporating all information. AP's behaviour displays several unusual characteristics. For example, the AP tends to keep interest rates higher on average than the Taylor (1993) rule, as it retains the economy slightly under its potential, perhaps as a form of insurance against sudden inflationary outbreaks. The AP also tends to move more hastily towards the zero lower bound (ZLB) and remains longer at the ZLB than a TR benchmark.

The paper is structured as follows. Section 2 puts my contribution into the context of the relevant literature on monetary policy rules. Section 3 introduces the reinforcement learning framework, explaining how it can be adapted to the central bank's decision problem. Section 4 presents a stylised economic simulator calibrated to US data, which serves as the training environment for the artificial central bank. Section 5 discusses the results, comparing the performance of the RL-based policy to standard Taylor rules and analysing the impact of non-linearities and partial observability. Section 6 concludes with a discussion of the implications of my findings and directions for future research. This paper shall motivate and inspire future research by outlining the potential of RL.

2 Literature

Studying optimal monetary policy has a long tradition in macroeconomics. This paper relates particularly to the stream of literature emphasising monetary policy feedback *rules*, going back at least to the seminal work of Taylor (1993).[1] Since John Taylor's contributions, researchers and monetary institutions around the world have analysed the properties and usefulness of many "Taylor-type" rules, such as the balanced–approach rule (Taylor, 1999), the inertial rule (English, Nelson, & Sack, 2003) or the first-difference rule (Orphanides & Williams, 2002). These rules rely on signals about macroeconomic variables, most prominently inflation, output gaps or past interest rates, to provide a policy recommendation for the central bank's instrument. While I consider rules based on precisely these signals, I do not assume any specific functional form for them. Importantly, the Taylor-type rule examined in this paper can be non-linear.

Whereas the focus of the related literature has been chiefly on positive (as in the original work by Taylor (1993)) or normative (as in Woodford (2001)) aspects of Taylor-type rules, this paper's contribution is mainly *methodological*. I present reinforcement learning as a flexible method to derive Taylor-type rules based on an approximate global solution of the central bank's control problem in a broad range of macroeconomic models that allow for simulating trajectories of economic variables subject to different rules. To the best of my knowledge, the only existing paper using RL to derive monetary policy rules is Hinterlang and Taenzer (2024). My work is comparable, but I differ by relying on a different algorithm, focusing on an interpretable form of non-linearity, fully non-linear Taylor-type rules and a partially observable environment. Inspired by the ideas of Taylor (1993, p. 208), I imagine a future where the policy rule of a trained artificial central bank will be one of the inputs to central bank decision-making.

More broadly, I contribute with a fresh perspective on the use of methods from (recursive) control theory for macroeconomic policy design. RL is an emerging control method which, thanks to its flexibility and track record in applied game theory, could potentially alleviate parts of the critique found in Lucas (1976) and Kydland and Prescott (1977). For example, in Kydland and Prescott (1977), the authors "conclude that there is *no* way

[1] Earlier modern contributions include Friedman (1959, Ch. 4)'s rule or McCallum (1988)'s rule. I only examine the case of an inflation-targeting central bank with an interest rate instrument.

control theory can be made applicable to economic planning when expectations are rational".[2] Contrary to standard engineering control methods, RL has seen particular success in playing *games* against expert humans and artificial learning opponents, e.g. AlphaZero for the games of chess, shogi and go (Silver et al., 2018). RL has been applied in settings where properties of the controlled dynamic system change subject to other players, for example, teams of artificial agents playing hide-and-seek in Baker et al. (2020), teams of artificial agents playing Dota 2 against professional gaming teams in OpenAI et al. (2019), or designing controllers for autonomous driving in cities by Kendall et al. (2019). In these scenarios, RL has learnt useful policies even though the controller interacts with strategising human players where its current decisions depend on expectations about others' policies and where policies can alter the expectations of other agents. The key to addressing the critiques by Lucas (1976) and Kydland and Prescott (1977) is to examine ways in which more "model-consistent" behaviour of economic agents can be simulated while training the artificial central bank. Simulating economies with rational expectation agents is a major area for future research, and I will sketch potential first steps in this direction.

The artificial central bank trained in my paper faces an economy characterised by non-linear dynamics and partial observability. The latter connects my work to the literature studying monetary policy rules under real-world uncertainty, such as measurement noise and data revisions (Orphanides, 2001, 2003b), unobservability of "natural" variables (Orphanides & Williams, 2002) as well as learning (Orphanides & Williams, 2008) and robustness to model uncertainty (Orphanides & Williams, 2007). In all these works, the central bank's information set was either incomplete or inaccurate. I focus on a central bank which cannot observe important state variables, such as the natural real rate r^*. Simulating measurement noise is an interesting area for further research. The artificial central bank faces fundamental uncertainty in that during both training and policy execution, it never "learns" about the equations and the parametrisation of the underlying model of the economy but merely deduces a satisfactory policy based on learning to associate economic states and actions with experienced losses. The robustness of the policy to model misspecification is examined, although further research is needed. More broadly, the emphasis on the usefulness of RL in non-linear settings naturally connects this work with

[2] As highlighted in Ljungqvist and Sargent (2018, Ch. 19), recursive control theory has found its way back to macroeconomic policy design since Kydland and Prescott (1977).

research analysing non-linear economies. Out of these, the zero-lower bound (Fernández-Villaverde, Gordon, Guerrón-Quintana, & Rubio-Ramírez, 2015) and a non-linear Phillips curve (Bunn et al., 2024; Karadi, Nakov, Barrau, Pasten, & Thaler, 2024) are important features of my economic simulator.

Reinforcement learning has been used only scarcely in macroeconomic research to date. Notable exceptions are Hinterlang and Taenzer (2024) in the context of optimal monetary policy and Chen, Joseph, Kumhof, Pan, and Zhou (2023), where a representative household learns to choose consumption, labour supply and savings in a DSGE model. For reviews of RL applications in economics, see Atashbar and Shi (2022) and Charpentier, Élie, and Remlinger (2023).

3 Reinforcement Learning

This section aims to introduce reinforcement learning to a macroeconomist audience and show how the method can be adapted to address the problem of optimal monetary policy in a wide variety of settings. The goal is to write a generic decision problem faced by the central bank that can be easily understood by macroeconomists, engineers, and computer scientists working on reinforcement learning alike. A significant contribution is to showcase the set of monetary decision problems where reinforcement learning could be more suitable than traditional methods while also exposing its weak points. I shall focus heavily on the *practical application* of RL to the decision problem of the central bank. A reader interested in the underlying theory is referred to the textbooks of Woodford (2003), Bertsekas (2023) and Sutton and Barto (2018).

3.1 Optimal Monetary Policy as a Dynamic Programming Problem

The monetary authority faces the problem of finding an optimal *instrument rule* $\mu^*(\cdot)$ from a set of admissible policies M that maps from the set of observations $\mathbf{o}_t \in \mathcal{O}$ to action $a_t \in \mathcal{A}$ and maximises the sum of its expected rewards, or equivalently, minimises the sum of its expected losses:

$$\min_{\mu \in M: \mathcal{O} \to \mathcal{A}} \mathbb{E}_0 \left[\sum_{t=1}^{\infty} \beta^{t-1} L(\mathbf{s}_t, a_t) \right] \tag{1}$$

s.t. Laws of motion: $$\mathbf{s}_{t+1} = F(\mathbf{s}_t, a_t, \mathbf{w}_{t+1}) \tag{2}$$

Policy (Taylor rule): $$a_t = \mu(\mathbf{o}_t) \tag{3}$$

The period loss of the policymaker, $L(\mathbf{s}_t, a_t)$ is assumed to be expressible in terms of the vector of state variables of the economy $\mathbf{s}_t$ while a_t (for action) denotes the policy instrument.[3] For now, I assume the DGP is stationary and Markovian; thus, the policy μ of interest will also be stationary and can be thought of as a variant of the "Taylor rule". A standard period loss could feature inflation deviations from the policymaker's goal, interest rate variation or an appropriately defined output gap. Furthermore, I assume that the data-generating process of the economy can be described by a function $F(\mathbf{s}_t, a_t, \mathbf{w}_{t+1})$ where $\mathbf{s}_t$ is a vector of only *predetermined* variables. This limits the focus onto models of the economy where expectations of future states are expressible in terms of realisations of past variables such that no variables in $\mathbf{s}_t$ can "jump" as a result of adjusting the policy $\mu(\cdot)$. Thus, I currently do not consider models where agents in the economy have rational expectations.[4] However, models with agents' learning (such as adaptive expectations) or static and backward-looking expectations are covered. The DGP of the economy is allowed to be non-linear, and the number of states can be large. The vector $\mathbf{w}_{t+1}$ stands for a vector of iid, serially uncorrelated random disturbances.[5] Finally, the policy rule of the central bank is assumed to react to a vector of observable variables $\mathbf{o}_t$, which may not fully overlap with the vector of the states of the economy $\mathbf{s}_t$. When the state of the economy is fully observable, $\mathbf{o}_t = \mathbf{s}_t$.[6]

[3] In a microfounded model, L could be represented directly by households' welfare. In contrast with linear-quadratic approaches, reinforcement learning allows minimisation with almost any non-linear loss function.

[4] In economies with non-predetermined variables, it is necessary to assume a policy rule and determine consistent expectations about the indefinite future. Only then can the current state of the economy be determined. Therefore, monetary policy decision problems where all agents have rational expectations are not naturally recursive and dynamic programming cannot be readily used. Currently, this paper considers only naturally recursive settings. There are approaches to recursify the decision problem under RE by augmenting the state, such as the promised value approach (Hills, Nakata, & Sunakawa, 2021) or the saddle-point method (Marcet & Marimon, 2019). Future research shall examine how to introduce RE.

[5] An example could be an iid normal demand shock occurring *after* the central bank set interest rates i_t, such that the reward the policymaker receives as a result of the decision i_t is stochastic even if the policymaker observes all time t variables.

[6] For ease of exposition, I might occasionally use $\mathbf{o}_t$ and $\mathbf{s}_t$ interchangeably, i.e. as-if assuming $\mathbf{o}_t = \mathbf{s}_t$.

To facilitate consistency in notation with RL literature, I shall denote $R(\mathbf{s_t}, a_t)$ the period *reward* function of the monetary authority, defined as the negative loss: $R(\mathbf{s_t}, a_t) \equiv -L(\mathbf{s}_t, a_t)$. The problem of the policymaker can then be written *recursively* in terms of a standard Bellman equation

$$V_t(\boldsymbol{s}_t) = \max_{\mu_t \in M} \{\mathbb{E}_t \left[R(\boldsymbol{s}_t, a_t, \mathbf{w}_{t+1}) + \beta V_{t+1}(\boldsymbol{s}_{t+1})\right]\} \tag{4}$$

where V denotes the value function. To solve this decision problem in well-behaved problems, I could use the many numerical tools from *exact* dynamic programming (EDP), e.g., value function iteration (VFI). However, implementing these numerical tools would require discretising the decision problem. In particular, I would have to simplify the DGP of the economy and use the knowledge of the reward function to find a finite state stochastic process $\mathbb{P}(s_{t+1} = s', r_{t+1} = r | s_t = s, a_t = a)$ to enable numerical evaluation of the expectation operator in equation (4). For the example of VFI, I would then use $\mathbb{P}$ to iterate

$$V_{k+1}(s) \leftarrow \max_a \sum_{s',r} \mathbb{P}(s', r \mid s, a) \left[r + \beta V_k(s')\right], \tag{5}$$

until convergence. Once the value function is found, the optimal (deterministic) policy can be recovered from $\mu^*(s) = \arg\max_a \sum_{s',r} \mathbb{P}(s', r \mid s, a) \left[r + \beta V^*(s')\right]$.

This procedure, and EDP in general, relies on a precise knowledge of the transition dynamics $\mathbb{P}(\cdot)$. In many situations it might be unfeasible or unpractical to obtain a closed-form expression for $\mathbb{P}(\cdot)$, or it might be prohibitive or inaccurate to numerically approximate these dynamics ((Sutton & Barto, 2018, Ch. 8)).[7] Reinforcement learning (RL) methods demonstrate how to solve decision problems *without* having access to these precise transition dynamics.[8]

RL methods extend to partially observable Markov decision processes (POMDPs), and the framework developed in the subsequent sections extends to the case where the policymaker receives a noisy or incomplete signal about the state of the economy.

[7] Sutton and Barto (2018) distinguish between a *distribution model*, which fully specifies the environment's probabilistic dynamics, and a *sample model*, which only provides observed transitions without an explicit probability distribution. Although both models can be used to sample trajectories, obtaining a sample model is typically much easier in practice. This advantage means that RL methods can work effectively even when the full dynamics are unknown, enabling researchers or central bank practitioners to explore and evaluate decision rules across a wider, more flexible range of scenarios.

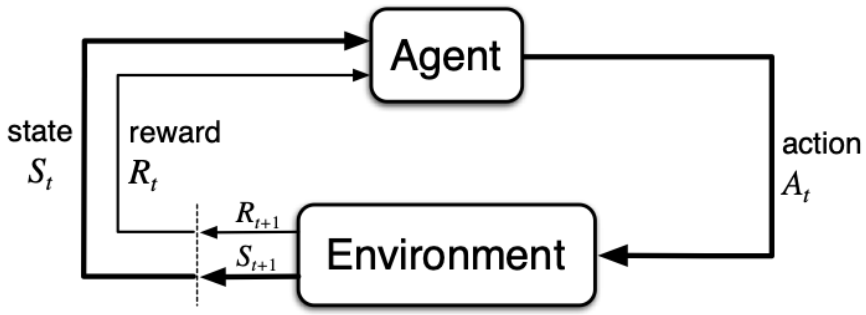
[8] Another practical problem with solving decision problems such as those in equations (1)–(4) exactly is the *curse of dimensionality*. Even if I had access to the precise dynamics of the economy, the problem's computational complexity increases exponentially in the number of state and choice variables. Addressing this weak point is the domain of *deep* reinforcement learning, which will be introduced later.

Instead, RL relies on sampling trajectories $\{s_0, a_0, r_1, s_1, a_1, r_2, \dots\}$, which can be done purely based on simulating the DGP, $F(\cdot)$, and using a pre-specified reward function $R(\cdot)$. These two functions can be jointly thought of as defining the "rules of the game", whereas knowing $\mathbb{P}(\cdot)$ is more demanding as it entails calculating exact transition probabilities for any possible state given any possible policy function. An agent using reinforcement learning creates its own data – the sampled trajectories – to learn and improve its approximations to either the value function $\widehat{V}(s)$ or the policy $\widehat{\mu}(s)$, or both. These sampled trajectories can come from a real-world interaction or an interaction in a simulator.[9]

3.2 Setup: The Main Building Blocks

I propose constructing an *economic simulator* based on an estimated macroeconomic model to enable this interaction for the decision problem of the monetary authority. Section 3.2.1 outlines a generic economic simulator – the "environment" – with a specific instance of an estimated minimalistic simulator provided in Section 4. Afterwards, the focus turns towards the possible representations of the policymaker – the "agent" – in Section 3.2.2. The subsequent Section 3.2.3 then describes how these two parts – the economic simulator and the policymaker – interact to generate trajectories of the economy. The details on how the system improves the policy – the RL algorithm – are then summarised in Section 3.3.

Figure 1: The Artificial Policymaker (Agent) Interacting with the Economy (Environment)



[9] Clearly, using real-world interactions is not feasible for RL in macroeconomics. Many decision problems, to which RL was already successfully applied, such as in robotics, autonomous vehicles, healthcare or energy management, were trained using simulated data. Using real-world interactions is often risky, costly, and time-consuming, so creating a simulation is the only viable alternative.

3.2.1 An Economic Simulator ("Environment")

The economic simulator can be considered a "playground" for the (artificial) policymaker, allowing it to try out different policies, collect experiences and stress-test the chosen strategy. As such, it is essential that the simulator can realistically capture how the economy reacts to its actions, i.e. that the simulator provides accurate feedback when asking "what-if" questions. For example, what happens to inflation and output over the business cycle when the central bank adjusts short-term interest rates more strongly to variation in unemployment? Answering "what-if" questions is a key task of macroeconomic models, and therefore, these are a good starting point to inspire an economic simulator.

Any model will be a simplification of the "true" DGP of an economy denoted by $F(\cdot)$. A researcher interested in finding well-performing monetary policy rules will have access to a simplified model $\widehat{F}(\cdot) \approx F(\cdot)$ pinning down the economic dynamics. There are at least two common forms $\widehat{F}(\cdot)$ can take. First, the (difference) equations in $\widehat{F}(\cdot)$ could be structural in the sense of being based on a microfounded model, particularly on the first-order equations describing the optimizing behaviour of agents in the economy. A well-known example would be the (calibrated) model of Galí (2015, Chapter 3). Secondly, $\widehat{F}(\cdot)$ could be an estimated macroeconometric model, such as an (S)VAR. A large-scale example of such a model used for policy analysis and forecasting would be the FED's FRB/US model (Brayton & Tinsley, 1996). In both cases, $\widehat{F}(\cdot)$ is allowed to be non-linear and highly-dimensional, which is precisely the situation where applying standard optimization tools would likely become impractical. The economic simulator will consist of all equations from $\widehat{F}(\cdot)$, *excepting* the central bank's reaction function, which I shall denote $\widehat{F}_{sim}(\cdot)$ for a simulator.

While both options to construct a simulator are interesting in their own right, in the present work, I focus on the second case where the $\widehat{F}_{sim}(\cdot)$ is based upon an estimated macroeconometric model. This choice comes with its weaknesses, particularly related to the critique of Lucas (1976). The parameters of $\widehat{F}_{sim}(\cdot)$ are assumed fixed, regardless of the currently active policy. The paper shall serve as a first attempt at implementing RL, and disregarding the Lucas critique at this point simplifies the problem considerably. This shall be relaxed in future work.[10]

[10]The reader should note that $\widehat{F}_{sim}(\cdot)$ could include equations that describe agents' learning about the

More specifically, my stylised simulator shall resemble the core structure of many macroeconomic models and consist of a set of difference equations $\widehat{F}_{sim}(\cdot) \equiv (f^S\ f^D\ f^E\ f^U)'$ describing aspects such as supply, demand, expectations and shocks:

$$\text{Supply: } \pi_t = f^S(\pi_{t-1}, \widehat{E}_t[\pi_{t+1}], x_t, u_t, \dots) \tag{6}$$

$$\text{Demand: } x_t = f^D(x_{t-1}, i_{t-1}, r_t^*, \widehat{E}_t[\pi_{t+1}], u_t, \dots) \tag{7}$$

$$\text{Expectations: } \widehat{E}_t[\pi_{t+1}] = f^E(\Omega_t, u_t, \dots) \tag{8}$$

$$\text{Disturbances: } u_t = f^U(u_{t-1}, \dots) + \varepsilon_t \tag{9}$$

where π denotes inflation, x is a measure of the output gap, r^* denotes the natural rate of interest, and the disturbances u can be thought of as a vector associated with demand, supply and expectation shocks $\varepsilon \sim \mathcal{N}(0, \sigma)$. Lastly, Ω is the information set available to agents in the economy when forming expectations. Section 4 proposes a minimalistic example of equations (6) – (9) and calibration.

3.2.2 An Artificial Central Bank ("Agent")

Our agent is the central bank tasked with setting short-term interest rates in the economy. I am interested in finding feedback rules of the form proposed by Taylor (1993) and extended by others such as Reifschneider and Williams (2000) and Orphanides (2003a). Policymakers at the FED commonly consult these policy rules to support their decision-making.[11] This paper deviates from previous work by allowing these policy rules to become non-linear.

Take as an example a version of the feedback rule proposed by Taylor (1993) denoted $\mu^{TR}(\cdot)$:

$$\mu^{TR}(\cdot) : i_t = r^* + \pi^* + \phi_\pi(\pi_t - \pi^*) + \phi_x x_t \tag{10}$$

where π^* is the inflation goal of the central bank, π_t is a measure of current inflation and x_t an estimate of the output gap. Taylor (1993) proposed $\phi_\pi = 1.5$ and $\phi_x = 0.5$.

currently active policy by re-estimating their prediction models, such as with VAR or adaptive expectations. For the case of the simulator being based on a microfounded model with full information rational expectations (FIRE), it is not immediately clear how to isolate the central bank's reaction function from the rest of the model. Showing how $\widehat{F}_{sim}(\cdot)$ can be implemented while allowing for FIRE is an important area for future research.

[11] See https://www.federalreserve.gov/monetarypolicy/policy-rules-and-how-policymakers-use-them.htm and FOMC historical materials, particularly Tealbooks A: Monetary Policy Strategies.

In my notation, using this rule implies that the central bank observes the state of the economy with $\mathbf{o}_t := \{\pi_t, x_t; r^*, \pi^*\}$. As many authors have shown (see Taylor and Williams (2010) for a good overview), policy rules of this form do work remarkably well in a broad range of models. Furthermore, simple rules offer practical advantages, mainly related to accountability, ease of commitment and communication. Nevertheless, there are numerous reasons why a researcher or a policymaker might be interested in allowing $\mu(\cdot)$ to become non-linear.

First, monetary policy strategies followed by central banks retain an important judgmental component, highlighted by authors such as Svensson and Tetlow (2005). The practical impossibility of a simple rule to capture the complexities of real-world practitioners is perhaps best summarised by a statement from the FED chairman:

"I am unable to think of any critical, complex human activity that could be safely reduced to a simple summary equation."

– Remarks of Jerome Powell at the Forecasters Club of New York Luncheon; February 22, 2017

If a researcher or practitioner is interested in a more accurate positive description of central bank decision-making, allowing for non-linear policy rule seems essential. The FED might follow a Taylor rule augmented with judgmental components, introducing non-linearities.

Secondly, there are many reasons to believe that non-linearities in the economy are of significant importance for the successful conduct of monetary policy. For example, the zero lower bound on interest rates or other occasionally binding constraints render the optimal policy non-linear (Adam & Billi, 2006, 2007; Fernández-Villaverde et al., 2015; Orphanides & Wieland, 2000). The same goes for the evidence on non-linear Phillips curves (Dolado, María-Dolores, & Naveira, 2005; Schaling, 2004). If the economy is subject to non-normal shocks, this can also lead to non-linear optimal policy (Swanson, 2006). The mere presence of model parameter uncertainty can render the optimal monetary policy non-linear (Tillmann, 2011; Wieland, 2000). Therefore, researchers interested in normative questions might also wish to search the space of non-linear feedback rules.

I shall represent the instrument rule of the monetary authority with an artificial neural network (ANN) parametrised by a vector θ. When the RL agent uses neural networks as function approximators, the methods are known as *deep* reinforcement learning (DRL).

ANNs are able to approximate any continuous function to an arbitrary degree of accuracy (Cybenko, 1989; Hornik, Stinchcombe, & White, 1989; Leshno, Lin, Pinkus, & Schocken, 1993). The agent will learn a parametrised *stochastic* policy $\widehat{\mu}(a_t|\mathbf{s}_t;\theta)$ that denotes a probability density function (PDF) of selecting an action a_t conditional on observing the state of the economy $\mathbf{s}_t$ where θ represents the weights and biases of the ANN.[12]

More specifically, the agent trains by sampling actions based on $a_t \sim \mathcal{N}\left(\eta(\mathbf{s}_t;\theta), \sigma_\mu^2\right)$ where $\eta(\mathbf{s}_t;\theta)$ is the policy ANN being updated, and σ_μ^2 a (learnable) parameter that adjusts the degree of randomness in actions also called "exploration". See Appendix A for the architecture of the policy ANN. Once the training has been concluded – for example, after a satisfactory performance is observed — the as-if deterministic policy is equal to the mean of $\mathcal{N}\left(\eta(\mathbf{s}_t;\theta), \sigma_\mu^2\right)$, i.e.

$$a_t = \widehat{\mu}(\mathbf{s}_t;\theta) = \eta(\mathbf{s}_t;\theta) \tag{11}$$

Allowing for stochastic policies during training is important for multiple reasons, but the key one is exploration. The stochasticity of the policy motivates the agent to try out different strategies – different interest rates for an observed state. Once the training is completed, the policy can be made as-if deterministic, fully "exploiting" the most likely action a_t for a state of the economy $\mathbf{s}_t$. I analyse such trained as-if deterministic policies in Section 5. The goal is to find such a function that well approximates the optimal policy, i.e. $\widehat{\mu}(\mathbf{s}_t;\theta) \approx \mu^*(\mathbf{s}_t)$ in the sense of performing similarly well in minimising the (artificial) policymaker's loss.[13]

Within the realm of DRL algorithms, I shall rely on a *policy gradient* and *actor-critic* method introduced in Section 3.3 below called *proximal policy optimisation* (PPO). Although the agent focuses on directly optimising the parametrised policy (the "actor"), it is also concerned with approximating a "critic", which is the value function of the decision problem. The presence of a critic reduces the variance of updates to θ and helps to

[12] Note that I write $\widehat{\mu}(a_t|\mathbf{s}_t;\theta)$ for a stochastic policy and $\widehat{\mu}(\mathbf{s}_t;\theta)$ for a deterministic policy. The difference is that, whereas the former is a PDF that needs to be sampled to determine the action, the latter directly determines the action. For example, $\widehat{\mu}(a_t|\mathbf{s}_t;\theta) = \mathcal{N}(a_t|\eta(\mathbf{s}_t;\theta), \sigma_\mu^2)$ samples $a_t \sim \mathcal{N}(\eta(\mathbf{s}_t;\theta), \sigma_\mu^2)$ with $\eta(\mathbf{s}_t;\theta)$ denoting the mean of the stochastic policy at state s_t. For the deterministic policy, $a_t = \widehat{\mu}(\mathbf{s}_t;\theta)$.

[13] This is a subtle point. In a non-linear and otherwise complex world, one should not expect that there is a unique, approximately optimal policy. Rather, there might be multiple strategies that lead to a satisfactory reward. There is no reason why the policy uncovered by the artificial agent $\widehat{\mu}(\mathbf{s}_t;\theta)$ shall recommend similar actions than an optimal policy $\mu^*(\mathbf{s}_t)$ would, *beyond the fact that the policy of the artificial agent should yield rewards comparable to the optimal policy.*

stabilise learning. As the function approximator of the critic, I shall use a *separate* neural network.[14]

3.2.3 Collecting Experience: The Interaction Between Agent and Environment

Once the artificial economy and the central bank are assembled, I can let them interact with each other in a joint artificial system. The main goal of this interaction is to generate data from which the agent can learn. The data, or rather the "experience", comes in the form of sampled trajectories (histories) of the economy. Denote one such trajectory τ_i, i.e. $\tau_i = \{s_0^{(i)}, a_0^{(i)}, r_1^{(i)}, s_1^{(i)}, a_1^{(i)}, r_2^{(i)}, \ldots, r_T^{(i)}\}$. The simulation is stopped, at the latest, after some pre-specified number of periods T. In implementation, this is true even though the agent optimises an infinite-horizon decision problem.[15] I shall call one such sampled trajectory an *episode*. Depending on the properties of the simulator, the episode can also be terminated early if the choices of the central bank have led to a "natural" terminal state, for example, when the policy leads to unstable dynamics in the simulator, resulting in hyper-deflation or hyper-inflation.[16]

I rely on a *model-free* RL method, which means that the agent is *not* learning a representation for the transition probabilities $\mathbb{P}(s', r|s, a)$ of the economy. There are both advantages and disadvantages associated with this choice. Model-free learning situates the agent in a world of extreme uncertainty – from the artificial central bank's perspective, the economy becomes a black box. Forcing the artificial central bank to interact with a black box is more scalable to complex simulations. It might also benefit the policy's robustness as the policy does not directly rely on a (potentially misspecified) model of transition dynamics. Relying on model-free learning also simplifies the algorithm and the learning procedure. However, using model-based RL could be helpful, mainly if the researcher is interested in finding a representation for the transition probabilities and understanding how the agent "plans" when deciding, which can help with the interpretability

[14] The architecture of this network does not have to be shared nor related to the policy network architecture. In many decision problems, value functions tend to be more complicated than policy functions. Therefore, the network architecture of the critic might be the more complex one of the two networks involved.

[15] Choosing T high enough does approximate an infinite-horizon decision problem. In my implementation, a period constitutes a quarter and $T = 500$, i.e. an episode lasts 125 years. Note that even for minor discounting with $\beta = 0.99$, $\beta^T \approx 0.007$.

[16] Early termination is additionally penalised, and the researcher must choose the degree appropriately.

Algorithm 1 An Episode (Example)

Require: The agent uses the currently best policy $\widehat{\mu}(a_t \mid s_t, \theta)$ ▷ On-Policy algorithm

Initialise the episode with $\mathbf{s}_0$ ▷ State at $t = 0$ may be randomised
$t \leftarrow 0$, truncated $\leftarrow$ **False**, terminated $\leftarrow$ **False**
while not (terminated **or** truncated) **do** ▷ Loop until episode ends
    Draw economic shocks w_{t+1}
    Sample an action $a_t \sim \widehat{\mu}(a_t \mid s_t, \theta)$
    Simulate through environment to get the next state $\mathbf{s}_{t+1}$
    Calculate reward r_{t+1}
    $t \leftarrow t + 1$
    if $t = T$ **then** ▷ T: max length of episode (hyperparameter)
        truncated $\leftarrow$ **True**
    end if
    if termination condition met **then**
        terminated $\leftarrow$ **True**
        $r_{t+1} \leftarrow r_{t+1} -$ penalty
    end if
end while

of the policy.

3.3 Training Procedure

3.3.1 Policy Gradient Methods

This section examines how the information contained in the sampled trajectories can help the agent improve the performance of his policy function. While I rely on a substantially more complex method introduced in the next section, the reader might wish to get a high-level intuitive understanding of the agent's learning. I explain the intuition on the example of the REINFORCE algorithm introduced by Williams (1992).

The REINFORCE algorithm is a foundational *policy gradient method*. The goal is to *directly optimise*[17] a parametrised policy $\widehat{\mu}(a_t|\mathbf{s}_t;\theta)$ that selects action a_t at time t given the state of the economy $\mathbf{s}_t$ parametrised with θ. An agent learning with REINFORCE uses the *experienced* discounted sum of rewards G_t for each time-step t, together with

[17] Optimising the policy directly, as opposed to optimising the value or action-value function, makes sense for at least two reasons here. First, I am mainly interested in the policy function of the decision problem. Secondly, the (near-optimal) policy is likely a much simpler object than the state-action or the state value function. Thirdly, these methods have stronger convergence guarantees ((Sutton & Barto, 2018, p. 324)). However, the reader should note that there is little to no evidence of local or *global* convergence of DRL algorithms, particularly for continuous action/state spaces and partially observable environments. Nevertheless, DRL seems to work well in many practical real-world problems. See, for example, Agarwal, Kakade, Lee, and Mahajan (2021) for a discussion of convergence properties.

his "memory" of taking action a_t in state s_t, to nudge the likelihood of choosing different actions in that state towards those actions that resulted in higher rewards.

Algorithm 2 REINFORCE: Monte-Carlo Policy-Gradient Control

Require: Input a parametrised policy $\widehat{\mu}(a|\mathbf{s},\theta)$; define learning step size $\alpha > 0$; Initialise the policy parameter $\theta \in \mathbb{R}^d$ where d is the number of components of θ

1: **while** True **do** ▷ Looping as long as necessary for satisfactory policy
2: Generate an episode $s_0, a_0, r_1, \ldots, s_{T-1}, a_{T-1}, r_T$, following $\widehat{\mu}(\cdot|\mathbf{s};\theta)$
3: **for** each step of the episode $t = 0, 1, \ldots, T-1$ **do**
4: Compute the return: $G_t \leftarrow \sum_{k=t+1}^{T} \beta^{k-t-1} r_k$
5: **end for**
6: Update the policy parameters: $\theta \leftarrow \theta + \alpha \sum_{t=0}^{T-1} \beta^t G_t \nabla_\theta \log \widehat{\mu}(a_t|\mathbf{s}_t,\theta)$
7: **end while**

The reason why policy gradient methods work is the *policy gradient theorem* proven by Sutton, McAllester, Singh, and Mansour (1999). Intuitively, this theorem shows that a gradient improving a *performance measure* of the policy can be approximated based purely on the data from the sampled trajectories. Specifically, define performance of a policy μ_θ as the value of that policy given a starting state s_0:

$$J(\theta) \equiv V^{\mu_\theta}(\mathbf{s}_0) = \mathbb{E}_{\mu_\theta}[G_t \mid S_t = \mathbf{s}_0] = \mathbb{E}_{\mu_\theta}\left[\sum_{k=0}^{\infty} \beta^k R_{t+k+1} \mid S_t = \mathbf{s}_0\right] \tag{12}$$

The policy gradient theorem shows that the gradient of equation (12) is proportional to

$$\nabla J(\theta) \propto \sum_{\mathbf{s}} d^{\mu}(\mathbf{s}_t) \sum_{a} \left(Q^{\mu}(\mathbf{s}_t, a_t) - b(\mathbf{s}_t)\right) \nabla \mu(a_t \mid \mathbf{s}_t, \theta) \tag{13}$$

where d^μ is the stationary distribution of the DGP under policy μ, $Q^\mu(\mathbf{s}_t, a_t)$ is the action-value function[18] defined as $Q^\mu(\mathbf{s}_t, a_t) = \mathbb{E}_\mu\left[r_{t+1} + \beta r_{t+2} + \ldots | \mathbf{s}_t, a_t\right]$ and $b(\mathbf{s}_t)$ is an appropriately chosen *baseline function* which reduces the variance of the expected value of the update

$$\theta \leftarrow \theta + \alpha \widehat{\nabla J(\theta)} \tag{14}$$

with a stepsize α. The update can be performed with stochastic gradient ascent once it is specified exactly how either $J(\theta)$ or $\widehat{\nabla J(\theta)}$ should be estimated based on the sampled trajectories. For example, in REINFORCE, $b(s_t) = 0$ and G_t is an unbiased estimate of $Q^\mu(s_t, a_t)$ with the update in (14) being performed on line 6 of Algorithm 2.

[18] In contrast with the value function, the action-value function gives the value of being in a state *after* an action has been chosen.

3.3.2 Proximal Policy Optimization

The REINFORCE algorithm has been substantially improved by subsequent research. One of the state-of-the-art policy gradient methods is the algorithm I shall use to optimise the policy of the artificial central bank – *proximal policy optimization* (PPO). Besides relying on the policy gradient theorem, PPO is also an *actor-critic method*, which means it also works with an estimate of a critic – the value function. Actor-critic methods use an estimate of the value function as the baseline in equation (13). PPO is a versatile, state-of-the-art DRL algorithm capable of handling both continuous and discrete action spaces. It can also be equipped with recurrent neural networks as function approximators, which could be useful in future applications with partial state observability or non-Markovian simulations. PPO is widely used in practice, for example, as an algorithm of default choice by OpenAI.[19] The algorithm has a proven track record of handling diverse tasks and excelling specifically in standard engineering optimal control problems.[20]

Instead of estimating the policy gradient directly via equation (13), PPO updates the policy based on a *surrogate objective function*. The objective is constructed such that its gradients approximate $\nabla J(\theta)$ and improve sample efficiency, variance of the updates and overall stability of the training. The algorithm was introduced by Schulman, Wolski, Dhariwal, Radford, and Klimov (2017) and builds upon ideas from trust region policy optimization (TRPO) in Schulman, Levine, Moritz, Jordan, and Abbeel (2017). The surrogate objective maximised by PPO is

$$J^{\text{PPO}}(\theta, \varphi) = \mathbb{E}[J^{\text{CLIP}}(\theta)] - c_1(V_\varphi(s) - V_{\text{target}})^2 + c_2 H(s, \mu_\theta(\cdot)) \tag{15}$$

where c_1 puts weight on how well the approximated value function predicts the data and c_2 weighs the entropy bonus of the policy H, which encourages exploration if performance improvements stagnate. Finally,

$$J^{\text{CLIP}}(\theta) = \mathbb{E}\left[\min\left(r(\theta)\widehat{A}_{\theta_{\text{old}}}(s,a),\ \text{clip}(r(\theta), 1-\epsilon, 1+\epsilon)\widehat{A}_{\theta_{\text{old}}}(s,a)\right)\right] \tag{16}$$

[19] See https://openai.com/index/openai-baselines-ppo/. PPO is regarded for its relative ease of implementation, stability of learning and insensitivity to hyperparameters.

[20] Hinterlang and Taenzer (2024) relied on a different algorithm, the deep deterministic policy gradient (DDPG). In future work, it will be interesting to benchmark different algorithms against each other for monetary policy tasks.

with ϵ adjusting how much the new policy is allowed to differ from the old by clipping $r(\theta)$ in between $1-\epsilon$ and $1+\epsilon$, and the probability ratio between the old and new policies $r(\theta)$ given by

$$r(\theta) = \frac{\mu_\theta(a_t \mid \mathbf{s}_t)}{\mu_{\theta_{\text{old}}}(a_t \mid \mathbf{s}_t)} \tag{17}$$

The key information that is estimated from the sampled trajectories is the *advantage* $\widehat{A}^\mu(\mathbf{s}_t, a_t)$. Intuitively, the advantage tells the agent how much better (in expectation) he will do if he chooses action a in state s relative to his expected sum of rewards from following the current policy, i.e. relative to the value of being in state s.[21] The gradient of J^{CLIP} is then able to nudge the policy in the appropriate direction to increase the performance. The estimate $\widehat{A}^\mu(s_t, a_t)$ is computed based on the *generalised advantage estimate* (GAE):[22]

$$\widehat{A}_t = \sum_{k=0}^{\infty} (\beta\lambda)^l \delta_{t+k} \tag{18}$$

where

$$\delta_t = r_t + \beta \widehat{V}_\varphi(\mathbf{s}_{t+1}) - \widehat{V}_\varphi(\mathbf{s}_t) \tag{19}$$

where the agent represents the critic $\widehat{V}(s_t; \varphi)$ by an ANN parametrised with φ. Equation (19) is called *temporal-difference error* (TD-error) in RL literature. λ controls the bias-variance trade-off of GAE with lower values lowering variance but increasing bias as the estimates become more myopic. The estimate of the critic gets continuously updated to minimise the mean squared errors of its value predictions in a manner resembling standard supervised learning with

$$\varphi_{k+1} = \arg\min_{\varphi} \left\| V_\varphi(\mathbf{s}_t) - \hat{R}_t \right\|^2_{\mathcal{D}_j}. \tag{20}$$

where $\mathcal{D}_j$ is the set of collected trajectories at updating iteration j and $\widehat{R}_t$ the *rewards-to-go*

$$\widehat{R}_t = \sum_{k=0}^{\infty} \beta^k r_{t+k} \tag{21}$$

Although this procedure might appear somewhat too complex, the upside is its flexibility

[21] For mathematical context, note that $A^\mu(\mathbf{s}_t, a_t) \equiv Q^\mu(\mathbf{s}_t, a_t) - V^\mu(\mathbf{s}_t)$ for each time-step t of each episode for the currently active policy μ where $V^\mu(\mathbf{s}_t) = \mathbb{E}_\mu\left[r_{t+1} + \beta r_{t+2} + \ldots | s_t\right]$ represents the value function under the currently active policy μ and $Q^\mu(\mathbf{s}_t, a_t) = \mathbb{E}_\mu\left[r_{t+1} + \beta r_{t+2} + \ldots | \mathbf{s}_t, a_t\right]$ is the corresponding action-value function.

[22] See Schulman, Moritz, Levine, Jordan, and Abbeel (2018) for an introduction into GAE.

and scalability to the most demanding optimisation problems. Even though the economic setup of this paper could be addressed with a substantially simpler DRL method (such as REINFORCE), I aim to use a state-of-the-art method. The performance of simpler algorithms could be analysed in an extension. I rely on the PPO implementation in Stable Baselines 3 (SB3). SB3 is an open-source Python package by an international team of researchers focused on transparent and careful replication of the papers underlying the original RL algorithms (Raffin et al., 2021).[23]

3.3.3 Hyperparameters

One of the advantages of using PPO is its relative ease of tuning. I demonstrate this by relying almost entirely on the default parametrisation of the SB3 implementation of PPO.[24] The only minor adjustment I introduce is to use a learning scheduler that dynamically adjusts the learning rate of the gradient descent α. I use cosine annealing with warm restarts (Loshchilov & Hutter, 2017). Learning schedulers are commonly employed to help with the multimodality of the loss function (avoiding local minima) and improve the convergence rate. I also set $c_2 = 0.01$ to encourage exploration. During training, the policies are evaluated once 90% of the envisioned interactions (total timesteps) are completed, i.e. after 90% of T interactions, I begin to evaluate the policy. The evaluation is performed roughly once every 25th training episode is completed, and each evaluation round simulates the environment for 1250 episodes. I keep the best-performing policy based on the rewards collected over the entire evaluation round.

[23] As an aside, note how the DRL algorithms solve the *curse of dimensionality* thanks to their reliance on deep neural networks. It is well known that the computational complexity of standard dynamic programming methods increases exponentially in the number of states and actions. This leads to bottlenecks related to the number of numerical operations needed and the computer memory involved. In contrast, DRL eliminates the computational complexity associated with memory. Notice that PPO is not storing the experienced trajectories, nor is it storing a one-to-one (table) mapping of states to values or states to actions. Instead, the only information stored across the entirety of the training consists of the two vectors of parameters, θ and φ, that describe the actor and critic functions. Furthermore, the numerical complexity depends on the length and number of episodes that need to be simulated until a policy performing well enough is found. For highly-dimensional problems, DRL might involve substantially lower amounts of floating-point operations than are needed for the exact solution to the problem. The important bottleneck becomes rather the quickness with which the simulator $\widehat{F}_{\text{sim}}(\cdot)$ can be sampled. The reader interested in the ability of neural nets to "solve" the curse of dimensionality in economic problems is referred to Fernández-Villaverde, Nuño, and Perla (2024).

[24] See https://stable-baselines3.readthedocs.io/en/master/modules/ppo.html for the default parametrisation.

Algorithm 3 PPO-Clip (Stable Baselines 3 Implementation)

Require: Initial policy and value function parameters θ_0, φ_0, entropy coefficient c_{ent}, value loss coefficient c_{vf}, clipping parameter ϵ, learning rate α, hyperparameters λ and β.

1: **for** $j = 0, 1, 2, \ldots$ **do**

2: Collect trajectories $\mathcal{D}_j = \{\tau_i\}$ by rolling out the policy μ_{θ_k} in the environment.

3: Compute rewards-to-go $\hat{R}_t$ (as above)

4: Compute advantage estimates $\hat{A}_t$ using GAE

5: Compute the PPO loss components:

- **Policy Loss**:

$$L_{\text{policy}}(\theta) = -\frac{1}{|\mathcal{D}_j|T} \sum_{\tau \in \mathcal{D}_j} \sum_{t=0}^{T} \min\left(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\right)$$

where $r_t(\theta) = \frac{\mu_\theta(a_t|\mathbf{s}_t)}{\mu_{\theta_j}(a_t|\mathbf{s}_t)}$ is the probability ratio with $\mu_{\theta_j}(a_t|\mathbf{s}_t)$ being the "old" policy used to collect the trajectories $\mathcal{D}_j$. The clipping ensures the ratio stays between $1 \pm \epsilon$.

- **Value Loss**:

$$L_{\text{value}}(\varphi) = \frac{1}{|\mathcal{D}_k|T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \left(V_\varphi(\mathbf{s}_t) - \hat{R}_t\right)^2$$

- **Entropy Regularization**:

$$L_{\text{entropy}}(\theta) = -\frac{1}{|\mathcal{D}_j|T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \text{Entropy}(\mu_\theta(\mathbf{s}_t))$$

6: Combine the losses:

$$L^{\text{PPO}}(\theta, \varphi) = L_{\text{policy}}(\theta) + c_{\text{vf}} L_{\text{value}}(\varphi) + c_{\text{ent}} L_{\text{entropy}}(\theta)$$

7: Compute gradients $\nabla_{\theta,\phi} L^{\text{PPO}}(\theta, \phi)$.

8: Update parameters:

$$(\theta_{j+1}, \varphi_{j+1}) \leftarrow (\theta_j, \varphi_j) - \alpha \nabla_{\theta,\varphi} L^{\text{PPO}}(\theta, \varphi)$$

9: **end for**

Table 1: Parametrisation of the PPO Benchmark (Stable Baselines 3 Implementation)

Hyperparameter	**Value**
Total timesteps (max interactions)	$10e^7$
Episode Length (T)	500
Activation Function	Tanh
Batch Size	64
Number of Epochs	10
Stochastic Gradient (Optimizer)	Adam
β	0.99
VF Coefficient (c_{vf})	0.5
Entropy Coefficient (c_{ent})	0.01
Clip Range (ϵ)	0.2
Clip Range VF	None
Initial Learning Rate (α)	$1e^{-4}$
Final Learning Rate	$5e^{-6}$
Learning Rate Scheduler	Cosine annealing w/ warm restarts
GAE Lambda	0.95
Max Gradient Norm	0.5
Network Architecture	2 hidden layers $[64, 64]$
Normalize Advantage	True
Use State-Dependent Exploration	False

4 A Minimalistic Economy

Next, I shall focus on building an economic simulator from which the artificial central bank can learn. I draw inspiration from modern microfounded models (Galí, 2015; Woodford, 2003) and seminal papers on optimal monetary policy under uncertainty (Orphanides, 2003a; Orphanides & Williams, 2007). I shall first construct a very tractable baseline linear case for the simulator and subsequently introduce non-linearities. The simulator's core is chosen so that it is easily extendible and matches the structure of textbook models. For the core of the simulator I rely on the "IS-LM-PC" structure as taught to undergraduates (for an example, see Blanchard (2020, Ch. 9)) where I replace the "LM" curve with a "Taylor rule" (TR) curve representing the artificial agent learning to conduct monetary policy. The "IS-TR-PC" is perhaps the most tractable model which could imitate the dynamics of larger linearised DSGE models, albeit without rational expectations.

I consider my "IS-TR-PC" simulator as a natural starting point for training an artificial agent at monetary policy.[25] Explicit microfoundations for the simulator are left for future

[25] Analogously, the "IS-LM-PC" model is perhaps the first macroeconomic model that undergraduate

work. The key aim for the present paper is clarity while ensuring somewhat "realistic" trajectories and policy trade-offs can be simulated. The simulator should introduce a non-trivial trade-off between three commonly highlighted goals of monetary policy, i.e. minimising inflation and output gap deviations as well as interest rate variability. Lastly, there should be room for time-varying natural rates, inertia in inflation and output gaps, persistent disturbances and the zero lower bound.

4.1 The Simulator: Main Building Blocks

There are several dimensions along which the textbook "IS-TR-PC" model needs to be adjusted so that a more realistic simulation emerges. Importantly, the textbook largely abstracts from the timing. I adopt the timing structure of Orphanides and Williams (2008), closely resembling the one found in microfounded New Keynesian models.[26] Furthermore, the IS-TR-PC simulator introduces additional persistence terms for output gap and inflation, allowing for a better match with macroeconomic data. The demand (IS) segment of the economy is given by

$$x_t = \gamma_0^{(d)} + \gamma_1^{(d)} x_{t-1} + \gamma_2^{(d)}(i_{t-1} - \pi_{t+1}^e - r_t^*) + u_t^{(d)} \tag{22}$$

$$u_t^{(d)} = \rho^{(d)} u_{t-1}^{(d)} + \varepsilon_t^{(d)} \tag{23}$$

where x_t is a measure of the output gap in the economy, r_t^* is a time-varying natural real rate of interest, and $u_t^{(d)}$ is an AR(1) demand disturbance. The length of the period t is a quarter. Particular attention should be paid to the exact meaning of i_{t-1} and π_{t+1}^e. The interest rate i_{t-1} is the average short-term federal funds rate *affecting the economy during quarter* t, set at the end of quarter $t-1$ (or equivalently at the beginning of quarter t).[27]

students encounter. I find it natural to think of my artificial agent as an uninitiated "student" of monetary policy.

[26] The timing is important for various reasons. Different assumptions could lead to different calibrations and dynamics. The timing also has implications for the monetary policy transmission mechanism and expectations. A different alternative would be to adopt a particularly tractable timing structure inspired by Orphanides and Williams (2004) that would allow me to simulate an economy with rational expectations. However, I stick to the timing that would be most recognisable to a macroeconomist.

[27] This is merely a notational convenience. Alternatively, one could write i_t while maintaining the assumption that the decision is based on information available until and including $t-1$, but I found this more confusing. Note that i_t in (22) cannot be determined from information about period t because no endogenous variable from period t is yet determined when the central bank would set i_t. While there are ways around this, for example, to use expected interest rates i_t^e in (22) or letting the central bank use expectations of time t variables while deciding interest rates, this would substantially complicate both the notation and the simulator.

Analogously, π^e_{t+1} are the inflation expectations about $t+1$ inflation formed at the end of quarter $t-1$ (or equivalently at the beginning of quarter t) with information about $t-1$ available.[28] The coefficients γ and ρ are to be calibrated and $\gamma_2^{(d)} < 0$. The timing of the demand function should capture the lag of monetary policy effects with interest rate decisions in t only affecting the output gap in period $t+1$. Real interest rates are given by the "Fisher equation" $r_t = i_{t-1} - \pi^e_{t+1}$. Monetary policy transmits via real interest rate gaps $(r_t - r^*_t)$ into the output gaps with a factor $\gamma_2^{(d)}$. The output gaps are persistent with factor $\gamma_1^{(d)}$, which helps to match empirical patterns. The supply (PC) segment of the economy is given by

$$\pi_t = \gamma_0^{(s)} + \gamma_1^{(s)}\pi^e_{t+1} + \gamma_2^{(s)}\pi_{t-1} + \gamma_3^{(s)}x_t + u_t^{(s)} \tag{24}$$

$$u_t^{(s)} = \rho^{(s)}u_{t-1}^{(s)} + \varepsilon_t^{(s)} \tag{25}$$

where π_t stands for a measure of the growth of price level between $t-1$ and t and $u_t^{(s)}$ is an AR(1) supply disturbance.[29]

The natural rate of interest r^*_t fluctuates unpredictably in the sense that it follows a random walk process

$$r^*_{t+1} = r^*_t + \varepsilon_t^{(r^*)} \tag{26}$$

which captures the essence of standard econometric models of natural rates, for example, Holston, Laubach, and Williams (2017). In the main case analysed in Section 5, the central bank also faces a zero lower bound, making $i_t \geq 0$ an occasionally binding constraint of the decision problem.

The central bank faces a non-trivial trade-off between its different goals of minimising inflation and output deviations as well as interest rate variability, with its period losses being

$$L(\mathbf{s_t}, i_t) = \lambda_\pi(\pi_t - \pi^*)^2 + \lambda_x x_t^2 + \lambda_i(i_t - i_{t-1})^2 \tag{27}$$

where λ_π, λ_x and λ_i are the weights on the respective goals.

[28] This matches real-world informational constraints. For example, the first public estimate of GDP (deflator) in quarter $t-1$ is available during the first month of quarter t.

[29] Note that if one assumes $r^*_t = r^*$, $\rho^{(d)} = \rho^{(s)} = 0$, $\gamma_1^{(d)} = \gamma_1^{(s)} = 0$, the simulator closely resembles the core ideas of the "IS-LM-PC" textbook model where demand and supply shocks shift the respective curves.

Lastly, an important building block of the simulator is the process of determining inflation expectations π^e_{t+1}. As a starting point, consider inflation expectations being formed by a weighted average of the central bank's inflation goal and households extrapolating past inflation:

$$\pi^e_{t+1} = (1-\vartheta)\pi^* + \vartheta\pi_{t-1} + u^{(e)}_t \tag{28}$$

$$u^{(e)}_t = \rho^{(e)} u^{(e)}_{t-1} + \varepsilon^{(e)}_t \tag{29}$$

where π^* is the inflation goal of the central bank and the parameter $1-\vartheta$ determines how well the expectations are anchored to the inflation goal. $u^{(e)}_t$ is an AR(1) disturbance term which affects the economy as an additional supply shock. The less anchored inflation expectations are ($\vartheta \uparrow$), the more persistent will be inflation – something the central bank wishes to avoid.

Relying on backwards-looking expectations is, without a doubt, a vast oversimplification of reality. The simulator could be enriched in various ways to incorporate features of adaptive expectations or even full information rational expectations. My first steps are sketched in Section 6.1. At this stage, I opted for a minimalistic simulator to focus on introducing the proposed approach and examining its viability for finding practical interest rate decision rules.

To summarise, the simulator provides the central bank (the agent) with feedback in every period. Specifically, after taking an action i_t the economy transitions to a new state $\mathbf{s}_{t+1}$ and the agent receives "reward" $\hat{R}_t = -\left[\lambda_\pi(\pi_{t+1}-\pi^*)^2 + \lambda_x x^2_{t+1} + \lambda_i(i_t - i_{t-1})^2\right]$ in the form of its negative period loss based on economic outcomes in the simulator. More granularly, an episode initialises with state variables $\mathbf{s}_0 = \{\pi_0, x_0, i_{-1}, r^*_0, u^{(d)}_0, u^{(s)}_0\}$. The simulator provides information on a subset of states $\mathbf{o}_0 \subseteq \mathbf{s}_0$ to the agent (the central bank) who needs to make a decision i_0.[30] *At the same time* households form inflation expectations π^e_1, i.e. neither the households nor the central bank observe each others' "choice" of i_0 and π^e_1. Lastly, shocks $\varepsilon^{(d)}_1$, $\varepsilon^{(s)}_1$ realise and the economy moves into the next period, i.e. $\mathbf{s}_1 = \{\pi_1, x_1, i_0, r^*_1, u^{(d)}_1, u^{(s)}_1\}$. Based on $\mathbf{s}_1$ and the action i_0, the agent receives feedback – a reward for its performance $\hat{R}_1$.

[30] I also experimented with providing the agent observations of state variables from a more distant past.

4.2 Introducing Non-linearities

As a next step, I shall introduce non-linearities into the simulator. The reader should view this as *one example* illustrating how policymaking in non-linear economies could be analysed thanks to RL. Notice that so far, given equations (22)–(29), the decision problem of the central bank is linear-quadratic.[31] If it remained so, the central bank could use optimal control theory – solve the linear-quadratic-Gaussian (LQG) problem – to derive the optimal (linear) decision rule given by a linear-quadratic regulator (LQR).[32] There would be no need for reinforcement learning in such a case, although it could still be used.[33] Therefore, to evaluate the potential usefulness of RL, I have to create non-linear dynamics.

As a highly illustrative and intuitive example, I chose to introduce non-linearity into the dynamic process for inflation expectations. Particularly, I shall make households' inflation expectations depend dynamically on the central bank's track record in keeping inflation at its goal. This adjustment will allow for richer dynamics, such as expectations becoming "unanchored", increasing inflation persistence due to a poor track record on stabilising inflation around π^*. More broadly, this paper is geared towards empowering and inspiring researchers who might want to understand how a non-linearity of their own interest impacts decision rules of economic agents.[34]

In the first step, the parameter ϑ, which determines the "degree of extrapolation" in households' inflation forecasts, shall become time-varying and dependent on how well the central bank has managed to stabilise the inflation at its goal ($|\pi_t - \pi^*|$). The better it performed, the more anchored shall expectations become. One way to model this "feedback

[31] Assuming no zero-lower bound, the data generating process of the economy was, so far, linear and the loss function is quadratic. For a theoretical overview of how to apply optimal control to LQ approximations of non-linear settings, see Benigno and Woodford (2012).

[32] This also assumes that the central bank could fully observe the state of the economy. In a partially observable case, LQR with a Kalman filter would be another option. For an example of applying LQR for optimal monetary policy in a partially observable setting, see Orphanides (2003a).

[33] Appendix B.1 examines the case of reinforcement learning in a linear simulation.

[34] Another example of an interesting non-linearity would be to focus solely on the presence of the zero lower bound and its impact on the "optimal" Taylor rule. Hinterlang and Taenzer (2024) have analysed this case in both linear and non-linear simulators and have derived piecewise linear Taylor-type rules. In contrast, I focus on learning a fully non-linear Taylor-type rule.

effect" is

$$\pi^e_{t+1} = (1 - \vartheta_{t-1})\pi^* + \vartheta_{t-1}\pi_{t-1} + u^{(e)}_t \tag{30}$$

$$\vartheta_{t-1} = 1 - e^{-k|\pi_{t-1} - \pi^*|} \tag{31}$$

where k is a parameter to be calibrated. Equation (31) implies that the degree of unanchordness of the expectations ϑ grows exponentially in the central bank's deviations from its goal. The equation also ensures proper bounds with $\vartheta_t \in [0, 1)$. Intuitively, the more unanchored expectations are, the stronger the central bank needs to account for inflation developments. For a particularly insightful and tractable case, I can analytically describe the optimal *non-linear* Taylor-type policy.

PROPOSITION .1 (Optimality of a ϑ-Dependent Taylor Rule) *Consider the economy described by equations (22) – (27) and (30) – (31). Suppose that* $\gamma^{(d)}_0 = \gamma^{(d)}_1 = \gamma^{(s)}_0 = \gamma^{(s)}_2 = \rho^{(d)} = \rho^{(s)} = \rho^{(e)} = 0$, $1 - \gamma^{(s)}_1 = \gamma^{(s)}_2$, *the natural rate is constant at* $r^*_t = r^*$ *and* $\{\vartheta_t, \pi_t\}$ *are perfectly observable. Further, suppose that the central bank's loss function is restricted to deviations of inflation from its target, i.e.* $\lambda_x = \lambda_i = 0$. *Then,*

$$i_t = r^* + \pi^* + \vartheta_t \left(1 - \frac{1}{\gamma^{(d)}_2 \gamma^{(s)}_3}\right)(\pi_t - \pi^*). \tag{32}$$

will generate the optimal sequence of interest rate decisions and solve (1) – (3).

Proof. In Appendix B.1 ■

Corollary .1 (Properties of the Simulated Economy under a Taylor-type Rule) *Consider the economy described by equations (22) – (27), (30) – (31), and parametrised as in Proposition .1. Following a Taylor-type rule of the form* $i_t = r^* + \pi^* + \phi_{\pi,t}(\pi_t - \pi^*)$ *leads to inflation evolving as an AR(1) process, i.e.* $\pi_{t+1} = (1 - \rho_{\pi,t})\pi^* + \rho_{\pi,t}\pi_t + error_{t+1}$ *with its persistence* $\rho_{\pi,t}$ *given by*

$$\rho_{\pi,t} = \vartheta_t\big(1 - \gamma^{(d)}_2 \gamma^{(s)}_3\big) + \gamma^{(d)}_2 \gamma^{(s)}_3 \phi_{\pi,t}$$

When the central bank follows (32), the inflation persistence will be eliminated, and the inflation goal will be achieved in expectations, i.e.

$$\rho_{\pi,t} = 0 \;\; and \;\; \mathbb{E}[\pi_{t+1} \mid \pi_t] = \mathbb{E}[\pi_{t+1}] = \pi^* \;\; \forall\, t$$

Proposition .1 shows how the degree of extrapolation in households' inflation forecasts ϑ_t directly affects the (optimal) strength of response to inflation deviations $\phi^*_{\pi,t} = \vartheta_t \left(1 - \frac{1}{\gamma_2^{(d)} \gamma_3^{(s)}}\right)$. Subsequently, Corollary 1 illustrates how a Taylor-type policy impacts inflation persistence. Whenever ϕ_π is chosen suboptimally, $\rho_{\pi,t} \neq 0$ *and* $\mathbb{E}_t[\pi_{t+1}] \neq \pi^*$.[35] These results highlight how and why the artificial central bank should care about the degree of extrapolation in agents' forecasts ϑ_t and about inflation persistence $\rho_{\pi,t}$, where the latter is directly tied to the track record of the central bank. More persistent inflation leads to additional pressure on the central bank to react vigorously, increasing interest rate variability.

Besides being an interpretable form of non-linearity, this specification of expectations also has the convenient side-effect that it reduces the incentive for the central bank to create an inflationary bias, which is of concern because households' inflation expectations are backwards-looking. The feedback effect from the track record to expectations should help to discourage the artificial central bank from learning to systematically exploit the Phillips curve relationship in the simulator, which is a key policymaking lesson going back to at least Friedman (1968) and Lucas (1976). Intuitively, suppose the artificial central bank tries to exploit the Phillips curve trade-off by pushing inflation above the goal. In that case, this comes with the additional risk of making inflation more persistent. If the central bank systematically does not deliver on its promise, hyperinflation occurs. Because of the "threat" of hyperinflation, a central bank that cares at least somewhat about its inflation and interest rate variability goals (λ_π, $\lambda_i > 0$) will likely find it undesirable to systematically deviate from its inflation goal.[36] In the simplified case from Proposition .1, the persistence of inflation is directly tied to the central bank's track record via ϑ_t.

In a second step, I adjust the simulator such that the artificial policymaker may more realistically "experience" the zero lower bound. Without further adjustments, for example, after a large negative demand shock, the simulated economy could easily drift into a

[35] For more details on this result, see the proof in Appendix.

[36] Depending on the exact parametrisation of the simulator, the central bank's policy can be passive when inflation is close to its announced goal. Therefore, it can still learn to "exploit" the PC relationship and create an inflationary bias as long as it remains small, perhaps as an insurance against the zero lower bound. However, either as a result of shocks or its attempt at generating positive output gaps, the central bank will have to react to inflationary pressures unless it wants inflation to become explosive. Essentially, the economy switches from stable to unstable regimes depending on the current value of ϑ_t and the parametrisation of the central bank reaction function. The variability of the inflation persistence has a more profound implication; *any* Taylor rule in the usual sense (with time-invariant parametrisation) can no longer be optimal.

deflationary spiral if the policymaker cannot set negative interest rates.[37] This would not match the recent zero lower bound (ZLB) experience well, where severe negative output gaps were coupled with positive inflation and inflation expectations. For example, during the Great Recession, a decline in US GDP by about 10% has only been associated with an inflation decline of about 1.5% (Christiano, Eichenbaum, & Trabandt, 2015). Naturally, there could be other reasons why the economy (inflation, output gaps) remains stable even with a binding zero lower bound than well-anchored or biased expectations. For example, policymakers could use fiscal interventions and monetary policy alternatives such as QE and forward guidance.[38] However, I rely on a non-linearity in expectations to "stabilise" the simulator at the ZLB.

Specifically, I use the empirical observation that surveyed agents (households and firms) tend to *not* expect deflations (Banerjee & Mehrotra, 2023), i.e. the expectations stay relatively well anchored (ϑ_t low) even with a minor deflation. One way to achieve this is to transform equation (31) into a threshold function that takes on the value of $\overline{k}$ for $\pi_t > \pi^*$ and $\underline{k}$ for $\pi_t < \pi^*$. As long as $|\underline{k}| < \overline{k}$, expectations will be more firmly anchored with negative than with positive deviations from the inflation goal. Data from the Surveys of Professional Forecasters suggest this could be the case. Lastly, I introduce an upper bound $\bar{\vartheta}$ for ϑ_t. Inflation expectations are fully described by

$$\begin{aligned}
\pi^e_{t+1} &= (1-\vartheta_{t-1})\pi^* + \vartheta_{t-1}\pi_{t-1} + u^{(e)}_t \\
\vartheta_{t-1} &= \begin{cases} \overline{\vartheta}\left[1 - e^{-\overline{k}(\pi_{t-1}-\pi^*)}\right], & \text{if } \pi_{t-1} > \pi^*, \\ \overline{\vartheta}\left[1 - e^{-\underline{k}(\pi_{t-1}-\pi^*)}\right], & \text{if } \pi_{t-1} \leq \pi^*. \end{cases} \\
u^{(e)}_t &= \rho^{(e)} u^{(e)}_{t-1} + \varepsilon^{(e)}_t
\end{aligned} \tag{33}$$

The equations (22)–(27) and (33) complete the simulator. Accounting for the non-linearity

[37] The artificial policymaker might find the economy drifting into a deflationary spiral when the ZLB is binding because of the variability in the real natural interest rates, large standard deviations and persistence of supply/demand shocks. In my initial experiments, the artificial bank learned to insure against deflationary spirals by creating an inflationary bias. Such a policy would be unrealistic, although it is an interesting finding which highlights the rationale behind raising the inflation target that has been proposed in the literature (e.g., Ball, 2014; Blanchard, Dell'ariccia, & Mauro, 2010).

[38] Reasons not related to policy were also proposed. For example, the "missing deflation puzzle" could be explained with real wage rigidities, non-linear Phillips curve relationship (Harding, Lindé, & Trabandt, 2022) and anchored inflation expectations (Bernanke, 2010). The artificial policymaker does not have the policy-related tools, and I do not model these non-policy mechanisms here. However, exploring how to introduce these realistic aspects and policy choices into a richer and more realistic simulator would be an interesting area for future research.

is clearly important for a well-performing policy. From the perspective of the artificial policymaker learning with model-free RL, this simulator is a "black box". To some degree, this perspective seems natural; even an experienced researcher might struggle to derive a sufficiently well-performing policy in this *sample* model. The natural next step is calibration, which should ensure that the trajectories the simulator generates are roughly comparable to real-world time series.

4.3 Calibration

I use quarterly US data gathered in December 2024 to calibrate the simulator. The dataset spans Q1 1969 until Q3 2024, the longest time frame for which all the necessary variables are available. I do not remove any observations or otherwise pre-process the time series. I use the annualised quarterly percentage change of the GDP deflator as a comprehensive measure of inflation π_t. The inflation expectations are taken from the Survey of Professional Forecasters (SoPF) maintained by the Federal Reserve Bank of Philadelphia. I rely on surveys of professional forecasters instead of households and firms to ensure the expectations are explicitly concerned with the GDP deflator. The inflation expectations π^e_{t+1} come from the survey conducted in quarter t. At the time the survey is conducted, professional forecasters have access to the "advance" estimates of GDP for $t-1$. The interest rate i_t is measured as the quarterly average of the FED funds rate. For measures of the output gap x_t and the real natural interest rate r^*_t, I use estimates of Holston et al. (2017) (HLW) available from the New York FED.

My goal is to bring the moments of the simulator close to the data while remaining agnostic about the "true" model of the economy. I note that the calibration procedure is coarse and shall be reexamined in future versions of the paper.[39] First, I estimate the demand and supply equations, i.e. equations (22) and (24), line-by-line with OLS. The coefficients obtained are directly used to calibrate the simulator.[40] The calibrated values are roughly in line with the estimates found in published work featuring similar reduced-form economies such as in Orphanides and Williams (2008) or Orphanides (2003b). A

[39] For example, the regression coefficients are biased. Nevertheless, the calibration yields values that generate reasonable moments, underlined by the fact that the artificial central bank can learn a Taylor rule, the recommendations of which correlate well with actual FED policy in Section 5.

[40] The intercept in the demand equation is only weakly significant (at the 10% level) and is therefore omitted from the simulator. The intercept in the supply equation is not significant.

Durbin-Watson test is performed on the demand and supply residuals $\varepsilon_t^{(s)}$ and $\varepsilon_t^{(d)}$, in both cases failing to reject the null of no autocorrelation.[41] Therefore, these shocks are simulated as white noise. The variance of the demand and supply shocks $\hat{\sigma}_u^{(d)}$, $\hat{\sigma}_u^{(s)}$ is calibrated by the variance of the estimated residuals. For the residuals in the random walk of r^*, i.e. equation (26), I use the standard deviation of $r_t^* - r_{t-1}^*$ to simulate the shocks $\varepsilon_t^{(r^*)}$. For the non-linear part of the simulator, i.e. the system for inflation expectations described in equation (33), I first estimate $\overline{k}$, $\underline{k}$ and $\overline{\vartheta}$ jointly using non-linear least squares.[42] The disturbance $u_t^{(e)}$ is found to be autocorrelated, and therefore $\rho^{(e)}$ is calibrated based on this remaining autocorrelation in inflation expectations.[43] The full calibrated system is

$$x_t = \underset{(0.02)}{0.92}\, x_{t-1} - \underset{(0.2)}{0.10}(i_{t-1} - \pi_{t+1}^e - r_t^*) + \varepsilon_t^{(d)}\,, \qquad \hat{\sigma}_u^{(d)} = 0.60 \tag{34}$$

$$\pi_t = \underset{(0.07)}{0.67}\,\pi_{t+1}^e + \underset{(0.06)}{0.39}\,\pi_{t-1} + \underset{(0.03)}{0.25}\,x_t + \varepsilon_t^{(s)}\,, \qquad \hat{\sigma}_u^{(s)} = 1.00 \tag{35}$$

$$\pi_{t+1}^e = (1-\vartheta_t)\pi^* + \vartheta_t \pi_{t-1} + u_t^{(e)} \tag{36}$$

$$\vartheta_t = \begin{cases} \underset{(0.02)}{0.77}\left[1 - e^{-24.70(\pi_{t-1}-2)}\right], & \text{if } \pi_{t-1} > 2, \\ \underset{(0.02)}{0.77}\left[1 - e^{0.12(\pi_{t-1}-2)}\right], & \text{if } \pi_{t-1} \leq 2. \end{cases} \tag{37}$$

$$u_t^{(e)} = \underset{(0.05)}{0.60}\, u_{t-1}^{(e)} + \varepsilon_t^{(e)}\,, \qquad \hat{\sigma}_u^{(e)} = 0.75 \tag{38}$$

$$r_t^* = r_{t-1}^* + \varepsilon_t^{(r^*)}\,, \qquad \hat{\sigma}_\varepsilon^{(r^*)} = 0.18 \tag{39}$$

For the calibration, I rely entirely on post-hoc data. While I am aware of the necessity to carefully consider real-time data availability in the design of practical monetary policy (Orphanides, 2001), the simulator should represent the "true" data generating process of the economy.[44] Figure 3 plots the calibrated system of inflation expectations with the blue

[41] For the demand and supply equations, the p-values are 0.11 and 0.35 respectively. The lack of serial correlation in the demand and supply disturbance terms could, perhaps, be ascribed to the inertia in the inflation and the output gap as well as to the variation in r_t^*.

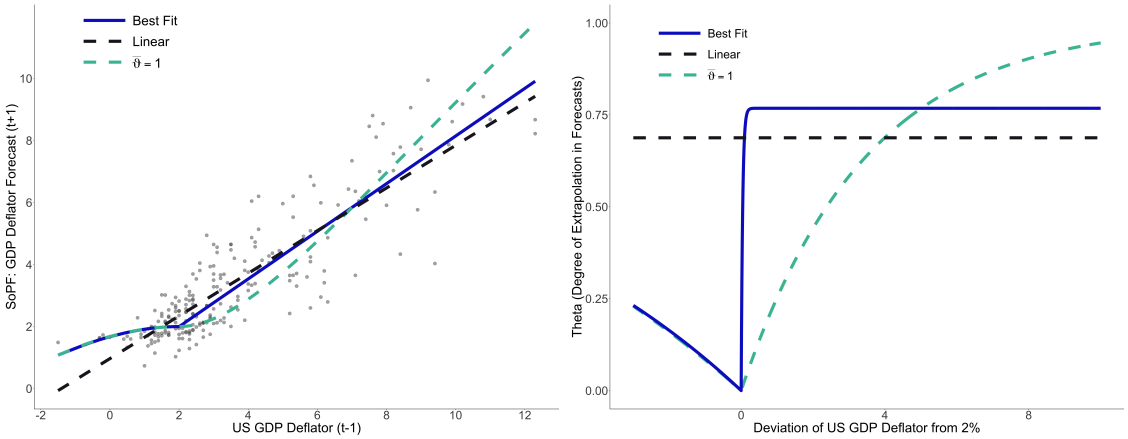
[42] I experimented with estimating π^* together with the three other parameters of the inflation expectations system. Since the coefficient was found to be 2.1%, I imposed $\pi^* = 2\%$ for simplicity. Additionally, estimating $\overline{\vartheta}$ individually for $\pi_{t-1} > \pi^*$ and $\pi_{t-1} \leq \pi^*$ did not substantially decrease the sum of squared residuals. Therefore, the simulator uses a single parameter for $\overline{\vartheta}$.

[43] To further improve the stability of the simulator at the zero lower bound. I simulate the heteroskedasticity of the white noise expectation shocks $\varepsilon_t^{(e)}$ by segmenting $\hat{\sigma}_u^{(e)}$ into four regions (equivalent amount of observations) depending on π_{t-1}. The error variance is substantially lower with low values of inflation. The estimated standard error ranges between 0.36 for the lowest and 1.12 for the highest inflation segment.

[44] Another reason for relying purely on post-hoc data when simulating the economy is the minimal availability of real-time estimates of r_t^* and x_t^*. For example, real-time estimates of r_t^* by HLW only go

lines showing the baseline (best fit) calibration described above. In the "best fit" (i.e., the residual sum of squared minimising) calibration, the expectation process essentially "jumps" from an anchored regime to a (somewhat) unanchored regime at 2% inflation. The non-linearity visibly improves the fit on the lower end in Figure 3a. However, the upper portion is well approximated by a linear curve.[45] For comparison, a "linear fit" is also plotted with the black dashed line representing the calibrated equation (28). Finally, the green dashed line highlights a case where $\overline{\vartheta}$ is not estimated but assumed $\overline{\vartheta} = 1$. This case leads to a more strongly (and smoothly) non-linear simulation, with inflation expectations gradually unanchoring with larger deviations from the inflation goal. Both the linear and the $\overline{\vartheta} = 1$ simulators serve as natural benchmarks in the subsequent section.

Figure 3: Inflation Expectations



(a) Expectation Formation: Data and Models

(b) ϑ for Different Levels of Inflation

5 Results

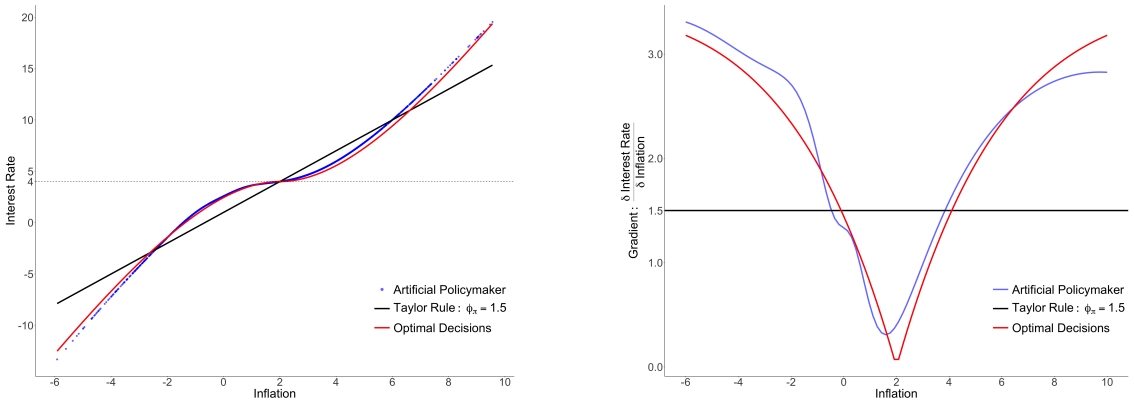
The results examine the outcomes of the artificial policymaker's learning. Before moving towards the more complex setting, it should pay off to analyse a case where the theoretically optimal decisions of the policymaker are known analytically.[46] This serves as a check of whether the artificial policymaker is learning as expected.

back to 2015Q4. An interesting extension of my current work would be to simulate central bank exposure to real-time measurement noise instead of assuming the observed data are "correct".

[45] Overall, the improvement in RSS (MAE) is only about 1% (3%) over the linear benchmark. It is important to keep in mind that the non-linearity does, nevertheless, have major implications for the optimal policy. The goal here is to showcase the method; improving RSS is only of secondary importance.

[46] Appendix B.1 examines the case of learning in the linear simulator.

Figure 6: Decisions Compared: Artificial Policymaker (RL), Taylor Rule, and Optimal



(a) Policy Functions

(b) Gradient (Slope) of Policy Functions

5.1 Simplified Non-linear Simulator

As the most tractable non-linear settings, I begin with the simplified simulator described in Proposition .1 in Section 4.2, for which the optimal policy is known. Figure 6a plots the optimal policy against the artificial policymaker's (AP) decisions and a comparable Taylor rule specification.[47] The AP decisions are plotted as points; these are 3000 random draws of actions observed in the simulation at different inflation levels after concluding learning and evaluation.[48] You can see that AP's decisions closely approximate optimal ones, although not perfectly. Bear in mind that the AP had no knowledge whatsoever about the DGP of the economy; it knew neither the functional forms nor the parameters. Still, the AP was able to learn a useful policy. Contrast the optimal policy to the black line plotting a standard Taylor rule specification using the "correct" intercept $(r^* + \pi^*)$ and a slope of 1.5.

Figures 6a and 6b show that the AP was able to recognise that around 2% inflation monetary policy can remain fairly passive, but once inflation gets out of close proximity of the goal (more than $\approx 6\%$ or less than $\approx -2\%$), the policy needs to react more forcefully. The mathematical reason for this goes back to the optimal coefficient ϕ_π^* in (32), which for this parametrisation varies from 3.5 (in both inflation limits) to 0 (at $\pi_t = \pi^*$). For a standard Taylor rule, this will be a constant ($\phi_\pi = 1.5$) depicted by the black horizontal

[47] The Taylor rule used here is $i_t = r^* + \pi^* + \phi_\pi(\pi_t - \pi^*)$.

[48] More specifically, I train the AP in monetary policy during 10 million interactions, where the last million is also used to evaluate the policy every 50 thousand interactions. I keep the policy that performed best out of these 20 policy evaluation trials (each lasting 1250 episodes).

line. Intuitively, the central bank needs to react more forcefully the farther away from 2% inflation is. When inflation is exactly at the goal, the optimal decision is to set $i_t = r^* + \pi^*$ depicted in the dashed horizontal line where all policy functions cross.

Table 3: Losses Comparison in the Simplified Non-linear Simulator

	Optimal	**Artificial Policymaker**	**Taylor Rule:** $\phi_\pi = 1.5$
$\pi - \pi^*$ (RMSE; pp.)	2.47 (100%)	2.48 (100.2%)	2.54 (102.7%)

NOTE: The root mean squared error losses are computed as average period losses (without discounting) over an episode lasting one million interactions (N). The RMSE is given by $\sqrt{\sum_{t=1}^{N}(\pi_t - \pi^*)^2/N}$ and can be interpreted as an estimate of the variance of inflation deviations for the policy. For interpretability, the numbers for relative RMSE are given *relative* to the optimum. pp. = per period.

Table 3 summarises losses experienced by a central bank acting according to the optimal policy, the policy of the AP and a Taylor rule. AP comes very close to the optimal loss, with TR yielding on average a 2.7% larger deviation from the goal per period. The differences relative to the optimal policy may appear small across the board – this is caused mainly by the lack of autocorrelated disturbances and a well-chosen Taylor rule specification. Experiments conducted in the subsequent sections have shown that the difference between the performance of AP relative to standard Taylor rules becomes more substantial with more complex policy trade-offs and richer dynamics introduced into the simulation.

5.2 Full Non-linear Simulator: Performance of the Artificial Policymaker

I now proceed to the main simulation calibrated in Section 4.3. Relative to the case just discussed, the artificial policymaker now faces a situation where it aims to achieve all three of its main goals – keeping inflation at its goal, minimising the output gap and smoothing interest rates. Moreover, the central bank faces an occasionally binding zero lower bound (ZLB) set at 0%. It also faces a time-varying natural rate of interest (r^*) and autocorrelated disturbances. The central bank attempts to minimise:

$$\min_{\mu:\mathcal{O}\to\mathcal{A}} \mathcal{L} \equiv \mathbb{E}_{t_0}\left\{\sum_{t=t_0+1}^{\infty} \beta^{t-t_0}\left[\lambda_\pi(\pi_t - \pi^*)^2 + \lambda_x x_t^2 + \lambda_i(i_t - i_{t-1})\right]\right\}$$

$$\text{s.t. } EQ\,(34) - (39)\,,\; i_t \geq 0\,(ZLB)$$

where the artificial policymaker receives feedback in the form of period rewards (negative of the period losses) while learning in the simulator. Since each of these three loss terms has different scales, it is not clear what the relative weight of each one should be if one wants to obtain a reasonable policy. I selected the weight such that the original Taylor (1993) rule, when acted upon in the simulator, would generate approximately a third of its loss from each of these three components. The corresponding values are $\lambda_\pi = 1$, $\lambda_x = 4$ and $\lambda_i = 2$.[49] The subsequent subsections document the performance of the artificial policymaker from various interesting angles, starting with the impact of having access to more or less information during training. I first analyse the performance impact of observing successively fewer state variables and then experiment with whether enlarging AP's "memory" leads to improved policy. Thereafter, AP's performance is benchmarked to Taylor rules. Lastly, I attempt to better understand the performance differential between TR and AP by quantifying the impact the introduced economic non-linearity can have on performance.

5.2.1 The Impact of Observability

Beyond the question of just how well the artificial policymaker (AP) will be able to perform in the simulator, I am interested in how this performance varies when the artificial FED has to rely on various information sets. There are multiple reasons for studying various informational settings. First, by forcing the AP to rely solely on r_t^*, inflation and output gap information, I can get a comparably fair estimate of the difference between the performance of RL and the standard versions of Taylor rules. Second, relying on even less information might force the policy to be more robust to simulator misspecification (reduce overfitting). Third, partial observability is an essential feature of real-world policy (Orphanides, 2001, 2003b) and a significant reason behind the usefulness of Taylor-type rules (Taylor & Williams, 2010). For example, output gaps are difficult to measure for central banks, particularly in real-time (Barbarino, Berge, & Stella, 2024). Relying on estimates of natural rates can lead to non-robust policymaking (Orphanides & Williams, 2002). More broadly, I wish to demonstrate the method's potential in finding useful poli-

[49] A further note on practical implementation is in order. Each episode is initialised with $\pi_0 = 2\%$, $x_0 = 0$, $r_0^* = 4\%$, $i_{-1} = 6\%$ and all disturbances set to zero.

cies in the realistic case of partially observable settings.[50] Therefore, I shall focus my results on the performance of AP that only relies on observing past inflation, output gap and interest rates. Having to rely on incomplete state information will make the task of the artificial central bank profoundly challenging.

Initially, I trained the AP while observing the full state of the economic environment. I call this the "full information" case, as the AP observes $\mathbf{o}_t = \mathbf{s}_t = \{\pi_t, x_t, i_{t-1}, r_t^*, u_t^{(e)}\}$ each period. Afterwards, I successively eliminate $u_t^{(e)}$, r_t^* and x_t, and for each case train new policies (from scratch) based on the reduced set of observations.[51] By reducing the observation space, I can study the performance impact of each piece of information and get a sense of the relative value of information for the central bank. The results from this experiment are summarised in Table 4.

Table 4: Full Simulator: Artificial Policymaker (AP) with Different Observations

Loss: AP	**Full Information** ($\mathbf{o}_t = \mathbf{s}_t$)	$\mathbf{o}_t^{(\mathbf{I})}$	$\mathbf{o}_t^{(\mathbf{II})}$	$\mathbf{o}_t^{(\mathbf{III})}$
Inflation $-\ \pi^*$	45% (2.63)	43% (2.61)	44% (2.83)	32% (3.14)
Output Gap	36% (0.98)	40% (1.00)	32% (1.13)	52% (2.02)
Δ Interest	19% (1.31)	18% (1.27)	24% (1.62)	17% (1.77)
Total	1.00	1.01	1.21	2.03

NOTE: The table shows the contributions of the three losses to the total loss of the central bank acting in the full simulator according to policies based on different observation sets to learn and execute the policy. The full information case shows the performance when the AP observes all state variables of the simulator: $s_t = \{\pi_t, x_t, i_{t-1}, r_t^*, u_t^{(e)}\}$. The subsequent columns list performances for successively sparser information sets: $\mathbf{o}_t^{(\mathrm{I})} = \{\pi_t, x_t, i_{t-1}, r_t^*\}$, $\mathbf{o}_t^{(\mathrm{II})} = \{\pi_t, x_t, i_{t-1}\}$, $\mathbf{o}_t^{(\mathrm{III})} = \{\pi_t, i_{t-1}\}$. The parentheses tabulate standard deviations of the period losses, i.e. the standard deviation of inflation from its goal, of the output gap and the interest rate. The benchmark (Total = 1.00) is the fully observable case. A total value of 1.21 means that the policy generated a 21% larger total loss than the benchmark. See the Note under Table 5 and 6 for further information.

The full information case could be considered the approximately best performance the AP can achieve given the current setup, i.e. using the same neural network architecture, algorithm, etc. Somewhat surprisingly, losing the information about the latest level of the expectations disturbance $u_t^{(e)}$ does not seem to make a meaningful impact, as the performance of $\mathbf{o}_t^{(\mathrm{I})}$ and full information case is almost equivalent.[52] However, once I remove the

[50] Another interesting avenue for future research is to simulate noisy observations about economic states.

[51] Note that eliminating i_{t-1} from the set of observations would make it impossible for the agent to learn interest rate smoothing. In contrast, learning to minimise the output gap is not necessarily entirely dependent on observing x_t, at least to the degree to which inflation is contemporaneously correlated with output gaps inside the simulator. A distinct alternative for teaching interest rate smoothing to the AP would be to rely on recurrent neural networks for the policy architecture. This would be an interesting avenue for future research.

[52] Note that minor differences in performance might be due purely to chance because of the stochastic

natural interest rate r_t^* from the set of observed variables, performance deteriorates by around 20%. Notably, the standard deviation of interest rates rises by almost 25% to 1.62, meaning the agent has to resort to a more activist monetary policy to achieve its other two goals of inflation and output gap stabilisation. Nevertheless, their standard deviations also increase by about 7.5% and 13%, respectively. Lastly, losing all information on output gaps severely deteriorates policy performance, with the total expected loss being about twice as high as under the full information setting. This underscores the importance of accounting for some measure of output gaps (or at the very least, a better proxy of output gaps than inflation) in a Taylor rule, substantially more so than accounting for measures of r^*. However, a central bank that worries less about economic slack ($\lambda_x \downarrow$), might experience only moderately higher losses under a $\mathbf{o}_t^{(\text{III})}$ policy, since the standard deviations of inflation and interest rates increase only by about 11% and 9% respectively. Unsurprisingly, most of the additional loss comes from AP's failure to stabilise the output gap with such a limited policy.

5.2.2 The Impact of Memory

Initially, I hypothesised that enlarging the "memory" of the AP could further improve the policy. For example, I could let the AP observe more lags of inflation, output gap and past interest rates as this might encode further useful information about the state of the economy, perhaps making up for the loss of information by not observing r_t^* and $u_t^{(e)}$. Somewhat surprisingly, I find that enlarging the observation space of the artificial agent in this way did mostly *not* improve the policy. The only meaningful improvement in the less informed policies (cases $\mathbf{o}_t^{(\text{II})}$ and $\mathbf{o}_t^{(\text{III})}$) I noticed was when the agent was allowed to observe the previous *two* interest rate decisions, instead of just one. This is the case $\mathbf{o}_t^{(\text{II};1,1,2)}$ in Table 5 below where the corresponding observations are $\{\pi_t, x_t, i_{t-1}, i_{t-2}\}$. Knowing the previous two interest rate decisions further helped the agent to better learn to smooth interest rates, improving the overall loss by further 7% vs. not observing i_{t-2}, i.e. the $\mathbf{o}_t^{(\text{II})}$ benchmark.[53] The standard deviation of interest rates decreased to a low of

nature of the initialisation and learning.

[53] I have also experimented with the AP observing the *change* of interest rates and/or making the *change* in interest rates the decision variable. To my surprise, this did *not* improve the policy performance. Particularly, observing only the *change* of interest rates and never the actual level resulted in AP not learning to smooth interest rates well.

1.38, which is still high relative to the standard deviation of the historical FED interest rate changes.[54] As the best-performing policy among the policies relying on minimal information, $\mathbf{o}_t^{(\text{II};1,1,2)}$ is further analysed below in Sections 5.3 and 5.4.

Table 5: Full Simulator: Artificial Policymaker with Larger Memory

Loss: AP	$\mathbf{o}_t^{(\mathbf{II})}$	$\mathbf{o}_t^{(\mathbf{II};1,1,2)}$	$\mathbf{o}_t^{(\mathbf{II};1,1,3)}$	$\mathbf{o}_t^{(\mathbf{II};2,2,2)}$	$\mathbf{o}_t^{(\mathbf{III})}$	$\mathbf{o}_t^{(\mathbf{III};1,\cdot,3)}$
Inflation $-\ \pi^*$	44% (2.83)	43% (2.73)	44% (2.75)	49% (2.90)	32% (3.14)	29% (3.08)
Output Gap	32% (1.13)	39% (1.09)	35% (1.07)	29% (1.07)	52% (2.02)	50% (2.06)
Δ Interest	24% (1.62)	19% (1.38)	21% (1.47)	22% (1.51)	17% (1.77)	21% (2.01)
Total	1.00	0.93	0.93	0.96	1.00	1.05

NOTE: The columns compare policy performances where the AP relied on different information sets to learn and execute the policy. Particularly, the information sets of $\mathbf{o}_t^{(\mathbf{II})}$ and $\mathbf{o}_t^{(\mathbf{III})}$ are enlarged with higher lags. The default for $\mathbf{o}_t^{(\text{II})}$ is the latest inflation, output gap and interest rate observation, i.e. $\mathbf{o}_t^{(\text{II};1,1,1)}$. For example, $\mathbf{o}_t^{(\mathbf{II};1,1,2)}$ adds an additional lag of interest rate. $\mathbf{o}_t^{(\mathbf{III};1,\cdot,3)}$ observes $\pi_t, i_{t-1}, i_{t-2}, i_{t-3}$. See Notes under Table 4 and 6 for further information.

Endowing the AP with earlier observations of past inflation and the output gap in $\mathbf{o}_t^{(\text{II};2,2,2)}$ does not seem to further improve the policy. Neither does knowing the level of past *three* interest rate levels in $\mathbf{o}_t^{(\mathbf{II};1,1,3)}$ and $\mathbf{o}_t^{(\mathbf{III};1,\cdot,3)}$. The performance of the policy might even deteriorate when further lags are introduced into the observation space as can be seen with $\mathbf{o}_t^{(\text{II};2,2,2)}$ and $\mathbf{o}_t^{(\mathbf{III};1,3)}$. I hypothesise that increasing the complexity of the observation space might make it more difficult to learn a successful policy. In future work, it would be interesting to examine how performance would evolve if the neural network underlying the policy function was made recursive.

5.2.3 Horse Race: Taylor Rules

The natural benchmark against which AP needs to perform well to justify using RL for such tasks are established Taylor rules, of which I analyse three. The first one is what I refer to as "the original" Taylor rule, being the specification proposed by Taylor (1993). When the ZLB is relevant, this is implemented as:

$$\mu^{TR;\,ZLB} : i_t = \max\left(0, r_t^* + \pi^* + 1.5(\pi_t - \pi^*) + 0.5x_t\right) \tag{40}$$

[54] Examining quarterly FED Funds Rate over the entire period available in the FRED database (Q3 1954 – Q4 2024), the standard deviation of the changes in FED funds was 0.85. Looking at a more recent history (Q1 1990 – Q4 2024) the standard deviations are merely 0.45. FED's June 2016 Tealbook B: "Optimal Control and the Loss Function" analysed different loss function weightings. Surprisingly, out of the four weight specifications analysed there, λ_i is never weighted more strongly than with $\lambda_i = 1$. Note that I am already optimising with $\lambda_i = 2$ here.

Another important benchmark shall be a Taylor rule (TR) using an imperfectly measured natural real rate $\widehat{r^*}$. When the actual FED consults a Taylor rule, it has to rely on a measurement of r_t^*, which can be noisy. This case is implemented as:

$$\mu^{TR;r^*noise} : i_t = \max\left(0, \widehat{r_t^*} + \pi^* + 1.5(\pi_t - \pi^*) + 0.5x_t\right) \tag{41}$$

where the measurement disturbance process for $r_t^* - \widehat{r_t^*}$ is simulated as in Orphanides and Williams (2002).[55] Lastly, I consider an inertial Taylor rule, which dampens the adjustments of interest rates over time relative to the responses of standard Taylor rules. The inertial TR is also one of the key simple rules FED is looking at while evaluating policy.[56]

$$\mu^{TR;inertial} : i_t = \max\left(0, 0.85i_{t-1} + 0.15\left(r_t^* + \pi^* + 1.5(\pi_t - \pi^*) + x_t\right)\right) \tag{42}$$

where the coefficients are chosen as in the FED's Tealbooks. Notice that for this last rule, output gaps are weighted more strongly. Another reason to include the inertial TR is that the previous two Taylor Rules have no notion of interest rate smoothing, which is one of the policymakers' goals, making the comparison somewhat unfair.

Table 6 summarises the performance results of the three Taylor rule alternatives (equations (40)–(42)) and the AP decision rule acting based on observations of inflation, output gap and the previous interest rate level, i.e. $\mathbf{o}_t^{\mathrm{II}}$. Interestingly, the artificial policymaker performs about 11% better than the original TR, *even without relying on the level of* r_t^* to decide. The AP achieves this thanks to a lower interest rate and output gap variability, with inflation deviations being somewhat larger on average.[57]

[55] The process for measurement disturbance follows $r_t^* - \widehat{r_t^*} = \rho^{(\widehat{r^*})}(r_{t-1}^* - \widehat{r_{t-1}^*}) + \varepsilon_t^{(\widehat{r^*})}$. Orphanides and Williams (2002) estimate this process to be highly persistent with $\rho^{(\widehat{r^*})} = 0.98$ and the standard deviation of the error at 1.96. I use these estimates to generate the mismeasurements when evaluating μ^{r^*noise}. Notice that lower persistence and standard deviations would significantly increase the performance of this Taylor rule.

[56] See, for example, FED's "Policy Rules and How Policymakers Use Them" at https://www.federalreserve.gov/monetarypolicy/policy-rules-and-how-policymakers-use-them.htm or simple rules in more recent Tealbooks A (e.g. December 2019 Meeting in Federal Reserve (2019)).

[57] The performance differential is likely caused by at least three things. First, the non-linearity of the simulated economy might be important for the performance, even more so than the level of r_t^*. Letting the AP observe r_t^* would increase the performance differential to about $0.89/1.21 \approx 0.74$, or 26%. Second, the original Taylor rule is not chosen to minimise interest rate variation. However, even disregarding losses from the interest rate variation, the AP still somewhat outperforms the original TR by about 0.5%. Third, and perhaps most importantly, the coefficients of the benchmark Taylor rule are *not* optimised for the simulator. However, running a time-consuming brute force search over a coarse grid of ϕ_π and ϕ_x

Table 6: Full Simulator: Comparing Taylor Rule(s) with Artificial Policymaker

Loss	**Taylor Rule**	**TR (r^* noise)**	**Inertial TR**	**Artificial Policymaker**
Inflation $-\ \pi^*$	36% (2.63)	36% (2.86)	49% (3.53)	44% (2.83)
Output Gap	33% (1.28)	36% (1.49)	47% (1.84)	32% (1.12)
Δ Interest	31% (1.97)	28% (2.01)	4% (0.77)	24% (1.62)
Total	1.00	1.17	1.29	0.89

NOTE: The "total" row compares the total loss from each of the rules to the loss obtained from the "original" Taylor rule (TR). For example, a relative loss of 0.89 means the agent acting according to the corresponding rule achieved a 11% lower total loss than the original TR. The results are based on simulating 2000 episodes of 500 periods each, i.e. at most one million interactions. The artificial policymaker (AP) acts while observing $\mathbf{o}_t^{(\text{II})} = \{\pi_t, x_t, i_{t-1}\}$. The individual loss terms are rescaled such that the three loss contributions for the case of the "original" Taylor rule (column 2) are roughly a third (33%) each. Specifically, $\lambda_\pi = 1$, $\lambda_x = 4$ and $\lambda_i = 2$. The tabulated losses are *not* discounted.

Furthermore, acting according to the original Taylor rule while accounting for a realistic degree in r^* measurement noise adds another 17% to the loss of the benchmark policy, making the losses on average $1.17/0.89 \approx 31\%$ higher than those achieved by the AP. While this additional noise does not substantially impact interest rate variability, the output gap and inflation deviations have higher variance. The inertial TR performs even worse on both the inflation and the output gap variability but minimises interest rate variability.[58] Overall, performance differentials in the range of $10\% - 60\%$ in lower total expected period losses between TR (r^* noise) and AP are observed. These striking differences highlight the potential of exploring non-linear TR given (more) realistic economic simulators.

5.2.4 The Impact of the Degree of Non-linearity

Although I have now established how well AP's actions perform during the simulation, it is still puzzling *why* the AP is able to outperform TR, especially in situations where the AP does *not* observe r_t^*. For example, it could be that accounting for the introduced non-linearity is relatively more important than fully observing the state of the economy. To better understand just how important the introduced non-linearity is, I compare the AP/TR loss differential in both a linear and a more non-linear simulation. Table 7 summarises results for three distinct parametrisations of inflation expectations. Results from Table 6 above are contrasted to the case of a fully linear simulation and the case where

revealed that the chosen coefficients are close to the best-performing ones. The AP still outperformed the best-performing TR by about 6% at 0.94 relative loss.

[58] As targetting interest rate variability becomes more important ($\lambda_i \uparrow$), the inertial TR might become preferable to the other TRs.

household expectations are more strongly non-linear. The three cases of expectation formation are contrasted in Figure 3a above. In the linear case, the equation (30) describing inflation expectations is kept linear with constant ϑ and estimated with OLS, corresponding to the black dashed line. In the relatively more non-linear case, expectations are modelled by the green dashed line estimated assuming $\bar{\vartheta} = 1$.

Table 7: Full Simulator: Best Fit vs. Linear vs. Exponential ($\bar{\vartheta} = 1$)

Loss: Table 6	**Taylor Rule**	**TR (r^* noise)**	**Inertial TR**	**Artif. Policymaker**
Inflation $-\ \pi^*$	36% (2.63)	36% (2.86)	49% (3.53)	44% (2.83)
Output Gap	33% (1.28)	36% (1.49)	47% (1.84)	32% (1.12)
Δ Interest	31% (1.97)	28% (2.01)	4% (0.77)	24% (1.62)
Total	1.00	1.17	1.29	0.89
Loss: Linear	**Taylor Rule**	**TR (r^* noise)**	**Inertial TR**	**Artif. Policymaker**
Inflation $-\ \pi^*$	33% (2.64)	33% (2.89)	45% (3.59)	49% (2.98)
Output Gap	31% (1.28)	35% (1.46)	51% (1.90)	30% (1.08)
Δ Interest	36% (1.97)	32% (2.01)	4% (0.81)	22% (1.53)
Total	1.00	1.18	1.34	1.00
Loss: $\bar{\vartheta} = 1$	**Taylor Rule**	**TR (r^* noise)**	**Inertial TR**	**Artif. Policymaker**
Inflation $-\ \pi^*$	54% (5.48)	61% (7.87)	–	19% (2.28)
Output Gap	32% (2.08)	31% (2.73)	–	36% (1.48)
Δ Interest	14% (2.05)	8% (2.17)	–	45% (2.55)
Total	1.00	1.67	–	0.51

In the linear case, AP performs on par with the original Taylor rule – although it tends to generate more considerable inflation losses accompanied by smaller output gap and interest rate variability losses. If the TR coefficients were optimised, a linear TR would easily outperform the AP, which cannot observe r_t^*. In contrast, when the importance of non-linearities increases ($\bar{\vartheta} = 1$), the AP markedly outperforms the original TR. The inertial TR fails to stabilise the economy for this scenario, with as many as 34% episodes ending in either an inflationary or a deflationary spiral. This result shows that the importance of non-linearities can be more decisive than a severe lack of information, such as not observing (and not accounting for) a critical variable such as the natural interest rate.

5.3 Interpreting a Neural-Network-Based Taylor Rule

Now that I have examined the performance of the AP relative to different Taylor rules and compared AP policies forced to rely on various information sets, the next goal is to try to

better understand the AP's decisions, mainly how they differ from the TR benchmarks. Interpreting a neural network is more challenging than a simple TR, which is a clear disadvantage of the proposed method. However, I wish to demonstrate to the reader that the neural network policy can nevertheless be usefully interpreted and that the AP's policy does not exhibit what could be considered unreasonable behaviour. I shall focus on analysing the policy denoted $\mathbf{o}_t^{(\mathrm{II};\,1,1,2)}$, as this is the best-performing policy among the policies *not* relying on observations of r_t^*, making it particularly intriguing.

Figure 7: Example of an Episode (Quarters 0 to 75)

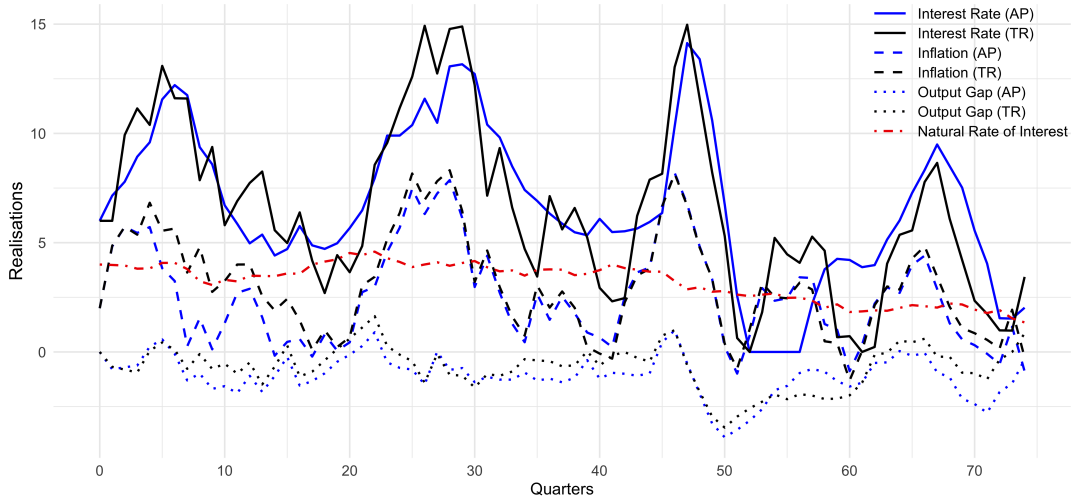


Figure 7 plots the first 75 quarters of an example episode chosen for its pronounced inflation outbreaks, which have occurred as a result of adverse supply shocks (high realisations of $u_t^{(e)}$ and $\varepsilon_t^{(s)}$). Both policymakers – TR and AP – have been exposed to equivalent shocks. Overall, the trajectory of AP's decisions is comparable to TR's decisions, their correlation over this time window being 0.84. Nevertheless, there are several differences in the decisions and the associated economic outcomes. AP's decisions appear to lead to lower interest rates during inflation peaks but higher interest rates during more "normal" times. The key difference seems to be in output gap outcomes. AP's output gap trajectory hardly ever reaches positive territory and is almost always below that of TR. I hypothesise that the AP has learnt to keep the output gap slightly lower, perhaps as insurance against inflation outbreaks.[59]

However, the AP is not willing to tolerate output gaps that are too negative. Once the

[59] See Figure E1 for a more extreme manifestation of this "insurance".

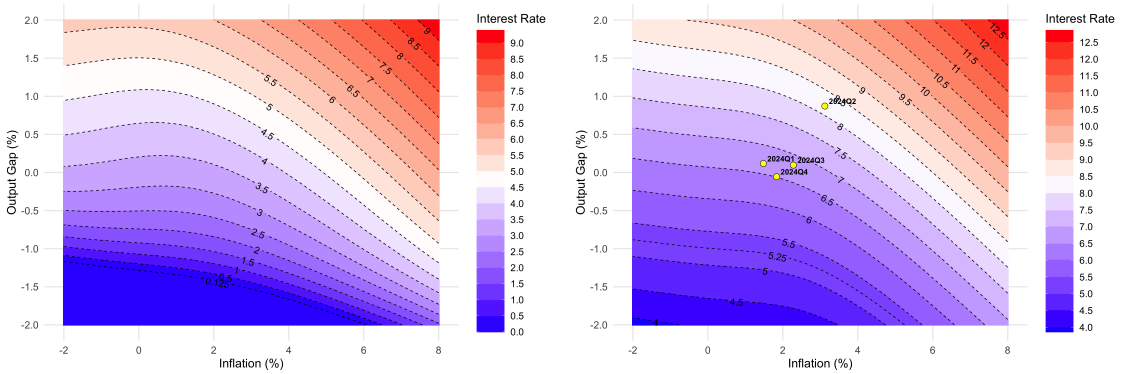
output gap becomes unusually negative (around Q50), AP rushes to the ZLB and keeps the interest rates there until the output gap recovers sufficiently (Q55-Q60). During such output "crises" in the simulator, AP rushes to the ZLB and sticks to it until recovery. Over these 70 quarters, AP's decisions generated lower mean inflation (2.6% instead of 3.0%) at the expense of a more negative output gap (−1.1% instead of −0.7%) with only slightly higher interest rates average (6.7% instead of 6.6%). Unsurprisingly, AP's decisions are much smoother than those chosen by the TR (standard deviation of 1.46 instead of 2.23).

One way to interpret AP's interest rate choices more systematically is proposed in Figure 8. These "heatmaps" plot AP's interest rate choices over a grid of inflation and output gap observations while fixing past interest rates. Figure 8a fixes past policy at the ZLB and Figure 8b at the level of 5.25%, the latter being FED's policy from Q4-2023 until Q4-2024. The heatmaps plot policy contours for increments of 0.5%. For example, a contour showing the level of 5% shows at which levels of inflation and output gap would the AP choose to readjust the policy rate to exactly 5% given the past two interest rate decisions have been 0% (left) or 5.25% (right). Beyond seeing what levels of interest rates the AP would choose in various situations, I can also interpret the slope of the contour lines. The flatter the contours, the (relatively) more important the variable on the y-axis (output gap) for the decisions of the artificial policymaker. For example, you can see that for lower levels of inflation, the output gap is relatively more important in determining policy than for higher levels of inflation, where the contour lines take on a more pronounced downward slope. Overall, however, the contour lines are substantially flatter than for the original Taylor (1993) rule (c.f. Figure E2), highlighting that output gaps play a relatively more prominent role in AP's decisions. This flatness is clearly visible when the AP is at the ZLB. For example, when the output gap is strongly negative at −2%, the artificial FED does *not* lift the policy rate until inflation is at least around 6%, well above its 2% target.[60]

What is perhaps striking is how thin the regions are and, therefore, how aggressive the suggested readjustment in interest rates might turn out to be. For example, you can see four policy recommendations for Q1-Q4 2024 in Figure 8b depicted by points based on real-time data. The AP recommends rate readjusting the FED Funds Rate by about 1.5%

[60]Major crises where the output gap remains below −2.0 for at least a year are ongoing in about 8% of the periods, and thus are relatively sporadic.

Figure 8: AP's Decisions at Different Levels of Inflation and Output Gap (Interest Rates Fixed)



(a) Policy "Heatmap" at the ZLB

(b) Policy "Heatmap" at 5.25% Interest Rates

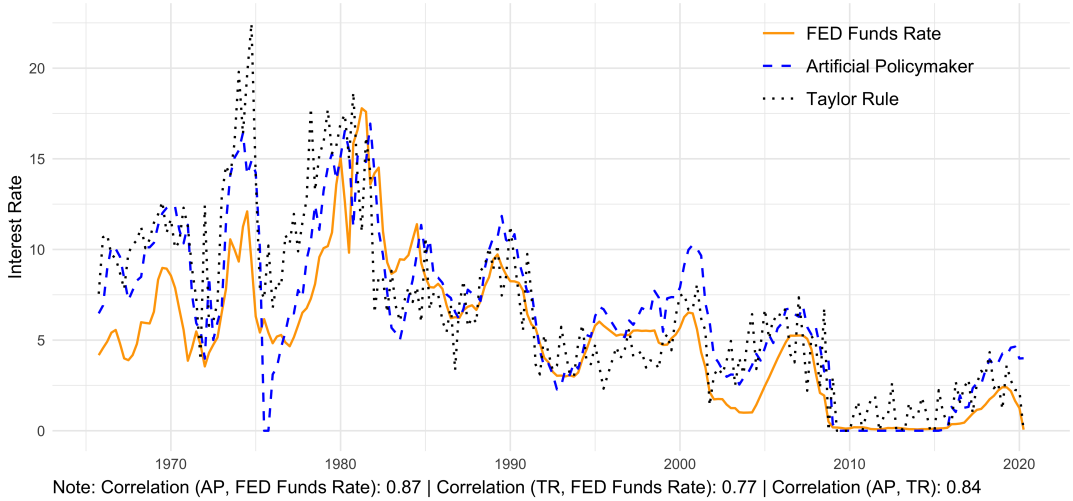
during this period. Adjusting the policy rate by more than 0.5% seems vastly inappropriate given the FED's past policy choices, but these values merely suggest a hypothetical degree of misalignment between the actually chosen and AP's suggested level of interest rates. The heatmaps are contingent upon the moments of actual economic variables being comparable to those of the simulator *while always following AP's recommendations*, an assumption which cannot be satisfied for real-world data. Calibrating AP to weigh interest rate smoothing more strongly would widen the regions, particularly those around the latest interest rate level, making AP's suggestions more appropriate for informing FED's decision-making processes.

5.4 Horse Race: Historical Data and the FED

To further evaluate the reasonableness of AP's decisions, I assemble real-time data on inflation and output gaps together with the path of interest rates chosen by the FED.[61] I let the artificial FED recommend interest rate decisions using this real-time data on the US economy. Figure 9 plots these recommendations from 1965 until 2020 against actual FED decisions and the Taylor (1993) rule.

[61] The data on inflation stems from the Real Time Dataset for Macroeconomists available from Philadelphia's FED. Analogous to model estimation, I use the Q/Q (annualised) growth rate of the GDP/GNP deflator. The "first" estimates for quarter t are used (corresponding to BEA's Advance estimates) to inform the interest rate active during quarter $t+1$. For the real-time output gap estimates, I use the database constructed by Barbarino et al. (2024), as the real-time HLW estimates are only available starting 2015. From the estimates in Barbarino et al. (2024), I take the series with the most similar moments (mean, standard deviation) to the moments experienced by the AP in the simulator. This is the series for an unobserved components (UC) model accounting for inflation, which they refer to as UC (GDP/π).

Figure 9: US Data: FED Actions vs. Taylor Rule vs. Artificial Policymaker



The correlation of FED policy with AP's suggestions is substantial (0.87) and higher than that of the Taylor (1993) rule (0.77).[62] The corresponding root mean squared errors are 2.5 for the AP and 3.3 for the TR. This result may be surprising since the real-world economy likely differs substantially from my calibrated simulation. Moreover, AP's suggestions are also highly correlated with TR (0.84). Part of the higher correlation between AP and FED's policy can be explained by a better fit during the ZLB period of the first half of the 2010s. AP is also a good predictor of actual FED policy during the late 1980s and early 1990s, similar to what initially triggered the interest in Taylor (1993). If a better match between the AP's and the actual FED decisions is desired, the weight of interest rates for its losses λ_i could be increased.[63]

There are, however, interesting differences between the suggestions of AP and FED policy. During the late 1990s and the turn of the century, AP would have suggested substantially higher interest rates, compared to *both* the FED policy *and* the Taylor rule. These higher interest rate recommendations are most likely the result of relatively high output gaps in the run-up to the dot-com bubble and the AP being particularly wary of an overheating economy, even in the face of low inflation. However, AP suggests higher

[62] This is true for all output gap estimates in Barbarino et al. (2024) *which have comparable moments* to those experienced by AP in the simulator. When using one of the output gap series with markedly different moments, I get lower correlations than those obtained with the TR policy, suggesting a degree of AP overfitting to the simulator. Making the decisions proposed by AP less dependent on the moments (mean, standard deviation, min, max) of observed variables is an important area for further research.

[63] Alternatively, the other weights could also be readjusted – but this resulted in only marginally higher correlations.

interest rates for the *entirety* of the 2000s. Regarding the lift-off of the ZLB, the first quarter where AP suggests doing so is Q3 of 2015, with Q4 2016 being the first quarter where AP has suggested a ZLB lift-off for two consecutive quarters. Q4 2016 is also *exactly* the second quarter after the FED historically exited the ZLB. However, AP would have wanted to see a more vigorous and faster lift-off, persisting well until the start of the Covid pandemic. Overall, AP's suggestions seem reasonable, at least in the sense of being comparable to standard Taylor rules and correlated with actual FED actions.

6 Discussion and Outlook

While the project has already yielded many insights, numerous issues and derivative research questions could not yet be addressed in the present volume. The text has raised some of these points, such as the need to build microfoundations to show how microfounded simulators can be used with reinforcement learning, robustness of the policy, greater realism and better estimation of the simulator, benchmarking different RL algorithms, and more. I believe two of these avenues for future research are particularly important.

First, the issue of how to incorporate Lucas (1976) critique more firmly into the simulation, and ultimately how to simulate an economy with full information rational expectations agents. The subsequent Section 6.1 sketches my initial steps towards this goal. Second, the issue of designing more realistic simulators. Section 6.2 highlights my initial work in simulating a large-scale macroeconometric model of the FED, the FRB-US, as a particularly realistic learning environment for the artificial policymaker. Lastly, I realise the importance of further studying and improving the robustness of the trained AI-based Taylor rule. Many proven tools could be used to enhance the robustness of the policy, such as *domain randomisation* or *adversarial learning*. I briefly outline how these could be used in Section 6.3.

6.1 Alternatives in Expectation Formation

The current version of the simulator relies on backwards-looking expectations of inflation π_t^e. Although the expectations are thanks to ϑ_t – to some degree – adaptive, they neither react systematically to the policy function chosen by the artificial agent nor do they "learn" from observing the economic dynamics. The researcher can easily incorporate

"smarter" expectations by directly modelling households' learning in the simulator. A straightforward way to implement this is to postulate a parametrised expectations model $\pi^e_{t+1} = f^E(\Omega_t; \xi_t)$ where Ω_t is the information set of the households at the beginning of period t and ξ_t is a vector of *learnable* parameters.[64]

In my initial experiments, I allowed the households to utilise and re-estimate a linear regression model. Specifically, $f^E(\cdot)$ is a linear regression where the households re-estimate ζ_t at each t during the simulation based on $\pi_{t-1:t_0}, x_{t-1:t_0}, i_{t-1:t_0} \in \Omega_t$. The households use the most recent lag of the variables, i.e. $\pi_{t-1}, x_{t-1}, i_{t-1}$ to forecast π_{t+1}, and the model is initialised with historical US data. Naturally, $f^E(\cdot)$ could also be any other forecasting model, such as a VAR or an artificial neural network. It could even be a second reinforcement learning agent who gets rewarded by forecasting inflation well. The artificial policymaker can readily be embedded in settings where other agents are learning about his policy. Doing so would move the focus towards implementing *multi-agent* reinforcement learning, which is a promising area of RL research behind many recent breakthroughs such as Google Deepmind's Alphastar (Vinyals et al., 2019) or OpenAI's Five (OpenAI et al., 2019).

To incorporate economic agents with full information rational expectations, the researcher would have to provide information about the currently active monetary policy function to other agents in the simulator. This could likely be done by using the currently active policy μ_θ of the AP to calculate $\mathbb{E}_t[\pi^e_{t+1}]$ consistently with all dynamic equations of the simulator. While this seems straightforward for linear systems of difference equations, implementing this for non-linear systems, such as by using the first-order conditions of a large-scale DSGE, is possibly time-consuming and complex. Future research should determine how this could be done and move towards a unifying "manual" of using a non-linearised DSGE as a training simulator for the artificial monetary policymaker.

[64] Notice this is different to readjusting expectations based on a fixed rule such as

$$\pi^e_t = \pi^e_{t-1} + \lambda\left(\pi_{t-1} - \pi^e_{t-1}\right)$$

which is common in *adaptive expectations* literature. In such case, neither the parameter λ is being learnt, nor is the updating rule itself readjusted if it does not serve well. Such static forecasting rules can be readily introduced into the simulation, such as by replacing equation (28).

6.2 Alternatives in Simulators

A readily available source of realistic, easy-to-simulate economic environments can be found with macroeconometric models used by central banks worldwide. For example, the FED relies on a large-scale non-linear model of the US economy – FRB/US (Brayton, Laubach, & Reifschneider, 2014; Brayton & Tinsley, 1996) – for policy analysis and scenario simulation when preparing supporting materials (e.g., Tealbooks) for FOMC meetings. The computer code and data are available on FED's homepage. I adjusted the codebase to allow me to iteratively generate economic trajectories based on the interest rate decisions of the RL agent. I train the agent similarly to the AP presented in Section 5.[65] Initial results have confirmed that the trained RL agent can outperform actual FED decisions in their model. Figure 16 and the corresponding table plots the decisions of the "artificial FED" and shows the loss breakdown. The artificial FED has learnt to minimise inflation variation very well but varies interest rates too strongly. Further analysis and likely reweighing of the goals would be necessary to learn a more realistic policy function.

6.3 Robustness

Future research shall focus on testing and improving the robustness of AP's decision rule. The robustness could be tested, for example, by measuring AP's performance losses when varying the parametrisation of the simulator. Alternatively, one could embed the proposed Taylor rule into existing (microfounded) models.[66]

In terms of improving robustness during training, one approach proposed in the RL literature is *domain randomisation*. Examples of the method can be found in Peng, Andrychowicz, Zaremba, and Abbeel (2018) and Vuong, Vikram, Su, Gao, and Chris-

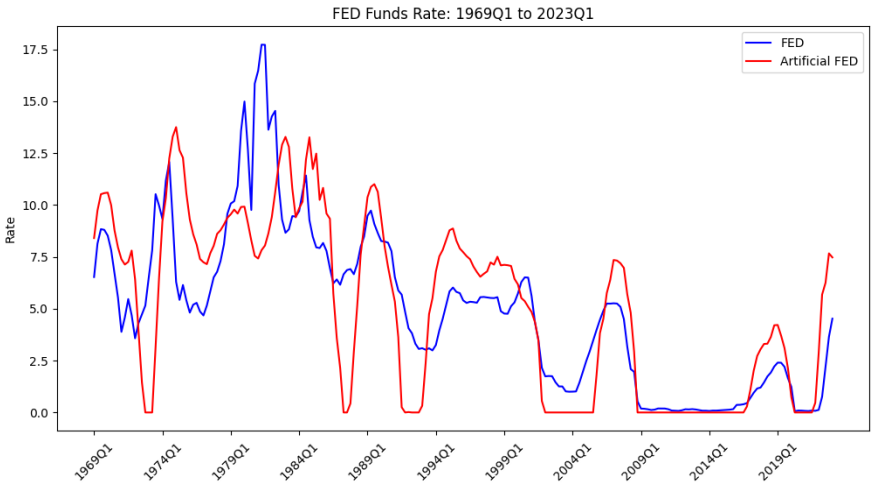
[65] The optimisation uses the exact loss function as in FED's Tealbooks, namely

$$L_t = \sum_{\tau=0}^{T} \beta^{\tau} \Big\{ \lambda_{\pi} \big(\pi_{t+\tau}^{PCE} - \pi^{LR}\big)^2 \; + \; \lambda_{u,t+\tau} \left(\text{ugap}_{t+\tau}\right)^2 \; + \; \lambda_R \big(R_{t+\tau} - R_{t+\tau-1}\big)^2 \Big\}$$

with equal weights. See FED Tealbooks for further details. I run FRB/US under the assumption of VAR expectations. The artificial FED was trained on a high-performance computing server. The result comes from a coarse experiment, and the performance could likely be substantially improved with further optimisation.

[66] Hinterlang and Taenzer (2024) have done a similar exercise with the trained *linear* policy rules using the Macroeconomic Model Data Base designed for DSGE model comparison by Wieland, Cwik, Müller, Schmidt, and Wolters (2012) and Wieland, Afanasyeva, Kuete, and Yoo (2016). Making such comparisons with non-linear rules is more demanding as this would require higher-order approximations of dynamics around the steady state in these DSGE models.

Figure 16: The Artificial FED in the FRB/US Model



Loss	**Taylor Rule**	**Artificial FED**
Inflation $-\ \pi^*$	63% (2.17)	52% (1.95)
Unemployment Gap	29% (1.47)	33% (1.57)
Δ Interest	8% (0.76)	15% (1.05)
Total	1.00	0.95

tensen (2019). Domain randomisation would entail defining a prior belief distribution about the parameters of the simulator and then sampling these prior beliefs when initialising an episode. In this way, the artificial policymaker would be trained in a "variety" of simulators instead of a specific one, hoping that once the agent is deployed, the dynamics of the actual economic system overlap with some trained-on examples. In the RL literature, particularly in robotics, this is also referred to as "closing" the *sim-to-real gap*.

Think of a specific parametrisation of the simulator as a model m and contrast it to the actual "true parametrisation" m_{real} of the economy. Instead of training the AP's policy parameters θ to maximise a performance measure such as done in equations (12)–(14), I could define a prior p_γ over the simulator's parameters γ, draw a model $m \sim p_\gamma$ (particular parametrisation of the simulator), and choose θ to maximise an expected performance measure over this prior distribution of simulators, i.e.

$$\theta^* = \arg\max_{\theta} \mathop{\mathbb{E}}_{m \sim p_\gamma} \left[J^{(m)}(\theta) \right]$$

In this way, the policymaker's uncertainty about the simulator's parametrisation can be baked into the trained policy function.

Another approach that the RL literature uses to enhance the robustness of the trained agents is the generative adversarial method, with an example found in Pinto, Davidson, Sukthankar, and Gupta (2017). The idea is to train a second agent (*the adversary*) who is actively trying to diminish the rewards of the policymaker. These adversarial attacks are separate actions in the economic simulation and can disturb the policymaker's observations or perturb the simulator's parametrisation. Effectively, the adversary is rewarded for forcing the artificial policymaker to experience unfavourable conditions, and by implication, the AP would need to focus more on learning to act in these bad states of the economy.

7 Conclusion

This study demonstrates that reinforcement learning can be a powerful tool for deriving monetary policy rules in non-linear macroeconomic models. I show that an artificial policymaker trained within a simulated economy can successfully approximate optimal policy decisions and outperform traditional Taylor rules, even without knowledge of the economy's data-generating process. A key advantage of the RL-based approach is its flexibility in tackling complex non-linear and high-dimensional decision problems as well as deriving strategies for partially observable settings. The paper showcases an entire development process, from crafting a simulated economy and choosing and implementing an appropriate RL algorithm to analysing and interpreting its results. It should serve as an inspiration for future projects.

Looking ahead, integrating more realistic expectation formation mechanisms and testing the policy's robustness across different economic environments are promising directions for future research. Applying RL to non-linearised large-scale DSGE models could further demonstrate its practical usefulness. Overall, this paper underscores the potential of RL to enhance monetary policy design, providing policymakers with a very flexible data-driven approach to navigating complex and uncertain economic dynamics.

References

Adam, K., & Billi, R. M. (2006). Optimal monetary policy under commitment with a zero bound on nominal interest rates. *Journal of Money, Credit and Banking*, *38*(7), 1877–1905.

Adam, K., & Billi, R. M. (2007). Discretionary monetary policy and the zero lower bound on nominal interest rates. *Journal of Monetary Economics*, *54*(3), 728-752.

Agarwal, A., Kakade, S. M., Lee, J. D., & Mahajan, G. (2021). On the theory of policy gradient methods: optimality, approximation, and distribution shift. *J. Mach. Learn. Res.*, *22*(1).

Atashbar, T., & Shi, R. A. (2022). *Deep Reinforcement Learning: Emerging Trends in Macroeconomics and Future Prospects* (IMF Working Papers No. 2022/259). International Monetary Fund.

Baker, B., Kanitscheider, I., Markov, T., Wu, Y., Powell, G., McGrew, B., & Mordatch, I. (2020). *Emergent tool use from multi-agent autocurricula.*

Ball, L. (2014). *The case for a long-run inflation target of four percent* (IMF Working Papers No. 2014/092). International Monetary Fund.

Banerjee, R., & Mehrotra, A. (2023). Unanticipated and Backward-Looking: Deflations and the Behavior of Inflation Expectations. *International Journal of Central Banking*, *19*(4), 41-83.

Barbarino, A., Berge, T. J., & Stella, A. (2024). The stability and economic relevance of output gap estimates. *Journal of Applied Econometrics*, *39*(6), 1065-1081.

Batista, Q., Coleman, C., Furusawa, Y., Hu, S., Lunagariya, S., Lyon, S., ... Zhang, H. (2024). Quantecon.py: A community based python library for quantitative economics. *Journal of Open Source Software*, *9*(93), 5585.

Benigno, P., & Woodford, M. (2012). Linear-quadratic approximation of optimal policy problems. *Journal of Economic Theory*, *147*(1), 1-42.

Bernanke, B. S. (2010, August 27). *The economic outlook and monetary policy.* https://www.federalreserve.gov/newsevents/speech/bernanke20100827a.htm. (Speech delivered at the Federal Reserve Bank of Kansas City Economic Symposium, Jackson Hole, Wyoming)

Bertsekas, D. (2023). *A course in reinforcement learning.* Athena Scientific.

Blanchard, O. (2020). *Macroeconomics Global Edition.* Pearson Deutschland.

Blanchard, O., Dell'ariccia, G., & Mauro, P. (2010). Rethinking macroeconomic policy. *Journal of Money, Credit and Banking*, *42*(s1), 199-215.

Brayton, F., Laubach, T., & Reifschneider, D. (2014). *The frb/us model: A tool for macroeconomic policy analysis* (FEDS Notes). Washington: Board of Governors of the Federal Reserve System. (Published on April 03, 2014)

Brayton, F., & Tinsley, P. A. (1996). *A guide to FRB/US: a macroeconomic model of the United States* (Finance and Economics Discussion Series No. 96-42). Board of Governors of the Federal Reserve System (U.S.).

Bunn, P., Anayi, L., Bloom, N., Mizen, P., Thwaites, G., & Yotzov, I. (2024). *How curvy is the phillips curve?* (NBER Working Papers No. 33234). National Bureau of Economic Research, Inc.

Charpentier, A., Élie, R., & Remlinger, C. (2023). Reinforcement Learning in Economics and Finance. *Computational Economics*, *62*(1), 425–462.

Chen, M., Joseph, A., Kumhof, M., Pan, X., & Zhou, X. (2023). *Deep reinforcement learning in a monetary model* (Papers). arXiv.org.

Christiano, L. J., Eichenbaum, M. S., & Trabandt, M. (2015). Understanding the great recession. *American Economic Journal: Macroeconomics*, *7*(1), 110–67.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, *2*(4), 303–314.

Dolado, J. J., María-Dolores, R., & Naveira, M. (2005). Are monetary-policy reaction functions asymmetric? the role of nonlinearity in the phillips curve. *European Economic Review*, *49*(2), 485-503.

Dubey, S. R., Singh, S. K., & Chaudhuri, B. B. (2022). Activation functions in deep learning: A comprehensive survey and benchmark. *Neurocomputing*, *503*, 92-108.

English, W. B., Nelson, W. R., & Sack, B. P. (2003). Interpreting the significance of the lagged interest rate in estimated monetary policy rules. *Contributions in Macroeconomics*, *3*(1).

Federal Reserve. (2019, November). *Tealbook a* (Tech. Rep.). Federal Reserve. (Published November 26, 2019. Available at https://www.federalreserve.gov/monetarypolicy/files/FOMC20191211tealbooka20191126.pdf)

Fernández-Villaverde, J., Gordon, G., Guerrón-Quintana, P., & Rubio-Ramírez, J. F. (2015). Nonlinear adventures at the zero lower bound. *Journal of Economic Dynamics and Control*, *57*, 182-204.

Fernández-Villaverde, J., Nuño, G., & Perla, J. (2024). *Taming the curse of dimensionality: Quantitative economics with deep learning* (Working Paper No. 33117). National Bureau of Economic Research.

Friedman, M. (1959). *A program for monetary stability*. Fordham University Press.

Friedman, M. (1968). The role of monetary policy. *The American Economic Review*, *58*(1).

Galí, J. (2015). *Monetary policy, inflation, and the business cycle*. Princeton University Press.

Hansen, L. P., & Sargent, T. J. (2011). *Robustness*. Princeton: Princeton University Press.

Harding, M., Lindé, J., & Trabandt, M. (2022). Resolving the missing deflation puzzle. *Journal of Monetary Economics*, *126*, 15-34.

Hills, T., Nakata, T., & Sunakawa, T. (2021). A promised value approach to optimal monetary policy. *Oxford Bulletin of Economics and Statistics*, *83*(1), 176-198.

Hinterlang, N., & Taenzer, A. M. (2024). *Optimal monetary policy using reinforcement learning* (Working Paper). SSRN Electronic Journal.

Holston, K., Laubach, T., & Williams, J. C. (2017). Measuring the natural rate of interest: International trends and determinants. *Journal of International Economics*, *108*, S59-S75. (39th Annual NBER International Seminar on Macroeconomics)

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*(5), 359-366.

Karadi, P., Nakov, A., Barrau, G. N., Pasten, E., & Thaler, D. (2024). *Strike while the iron is hot: optimal monetary policy with a nonlinear phillips curve* (BIS Working Papers No. 1203). Bank for International Settlements.

Kendall, A., Hawke, J., Janz, D., Mazur, P., Reda, D., Allen, J.-M., ... Shah, A. (2019). Learning to drive in a day. In *2019 international conference on robotics and automation (icra)* (p. 8248-8254).

Kydland, F. E., & Prescott, E. C. (1977). Rules rather than discretion: The inconsistency of optimal plans. *Journal of Political Economy*, *85*(3), 473–491.

Leshno, M., Lin, V. Y., Pinkus, A., & Schocken, S. (1993). Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, *6*(6), 861-867.

Ljungqvist, L., & Sargent, T. J. (2018). *Recursive macroeconomic theory*. MIT press.

Loshchilov, I., & Hutter, F. (2017). *Sgdr: Stochastic gradient descent with warm restarts.*

Lucas, J., Robert E. (1976). Econometric policy evaluation: A critique. In K. Brunner & A. H. Meltzer (Eds.), *Carnegie-rochester conference series on public policy* (Vol. Vol. 1, p. 19-46). North-Holland.

Marcet, A., & Marimon, R. (2019). Recursive contracts. *Econometrica*, *87*(5), 1589-1631.

McCallum, B. T. (1988). Robustness properties of a rule for monetary policy. *Carnegie-Rochester Conference Series on Public Policy*, *29*, 173-203.

OpenAI, Berner, C., Brockman, G., Chan, B., Cheung, V., Debiak, P., ... Zhang, S. (2019). *Dota 2 with large scale deep reinforcement learning.*

Orphanides, A. (2001). Monetary policy rules based on real-time data. *American Economic Review*, *91*(4), 964–985.

Orphanides, A. (2003a). Historical monetary policy analysis and the taylor rule. *Journal of Monetary Economics*, *50*(5), 983-1022.

Orphanides, A. (2003b). Monetary policy evaluation with noisy information. *Journal of Monetary Economics*, *50*(3), 605-631. (Swiss National Bank/Study Center Gerzensee Conference on Monetary Policy under Incomplete Information)

Orphanides, A., & Wieland, V. (2000). Efficient monetary policy design near price stability. *Journal of the Japanese and International Economies*, *14*(4), 327-365.

Orphanides, A., & Williams, J. C. (2002). Robust monetary policy rules with unknown natural rates. *Brookings Papers on Economic Activity*, *2002*(2), 63–118.

Orphanides, A., & Williams, J. C. (2004). Imperfect Knowledge, Inflation Expectations, and Monetary Policy. In *The Inflation-Targeting Debate.* National Bureau of Economic Research, Inc.

Orphanides, A., & Williams, J. C. (2007). Robust monetary policy with imperfect knowledge. *Journal of Monetary Economics*, *54*(5), 1406-1435. (Carnegie-Rochester Conference Series on Public Policy: Issues in Current Monetary Policy Analysis November 10-11, 2006)

Orphanides, A., & Williams, J. C. (2008). Learning, expectations formation, and the pitfalls of optimal control monetary policy. *Journal of Monetary Economics*, *55*, S80-S96. (Contributions to Macroeconomics in Honor of John Taylor)

Peng, X. B., Andrychowicz, M., Zaremba, W., & Abbeel, P. (2018). Sim-to-real transfer of robotic control with dynamics randomization. In *2018 ieee international conference on robotics and automation (icra)* (p. 1–8). IEEE Press.

Pinto, L., Davidson, J., Sukthankar, R., & Gupta, A. (2017). Robust adversarial reinforcement learning. In *Proceedings of the 34th international conference on machine learning - volume 70* (p. 2817–2826). JMLR.org.

Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., & Dormann, N. (2021). Stable-baselines3: reliable reinforcement learning implementations. *J. Mach. Learn. Res.*, *22*(1).

Reifschneider, D., & Williams, J. C. (2000). Three lessons for monetary policy in a low-inflation era. *Journal of Money, Credit and Banking*, *32*(4), 936–966.

Schaling, E. (2004). The nonlinear phillips curve and inflation forecast targeting: Symmetric versus asymmetric monetary policy rules. *Journal of Money, Credit and Banking*, *36*(3), 361–386.

Schulman, J., Levine, S., Moritz, P., Jordan, M. I., & Abbeel, P. (2017). *Trust region policy optimization.*

Schulman, J., Moritz, P., Levine, S., Jordan, M., & Abbeel, P. (2018). *High-dimensional continuous control using generalized advantage estimation.*

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). *Proximal policy optimization algorithms.*

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., ... Hassabis, D. (2018). A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, *362*(6419), 1140-1144.

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction.* MIT press.

Sutton, R. S., McAllester, D., Singh, S., & Mansour, Y. (1999). Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, *12*.

Svensson, L. E., & Tetlow, R. J. (2005). Optimal Policy Projections. *International Journal of Central Banking*, *1*(3).

Swanson, E. T. (2006). Optimal nonlinear policy: signal extraction with a non-normal prior. *Journal of Economic Dynamics and Control*, *30*(2), 185-203.

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

Taylor, J. B. (1999). A Historical Analysis of Monetary Policy Rules. In *Monetary Policy Rules* (p. 319-348). National Bureau of Economic Research, Inc.

Taylor, J. B., & Williams, J. C. (2010). Chapter 15 - simple and robust rules for monetary policy. In B. M. Friedman & M. Woodford (Eds.), (Vol. 3, p. 829-859). Elsevier.

Tillmann, P. (2011). Parameter uncertainty and nonlinear monetary policy rules. *Macroeconomic Dynamics*, *15*(2), 184–200.

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., ... Silver, D. (2019). Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, *575*(7782), 350–354. doi: 10.1038/s41586-019-1724-z

Vuong, Q., Vikram, S., Su, H., Gao, S., & Christensen, H. I. (2019). *How to pick the domain randomization parameters for sim-to-real transfer of reinforcement learning policies?*

Wieland, V. (2000). Monetary policy, parameter uncertainty and optimal learning. *Journal of Monetary Economics*, *46*(1), 199-228.

Wieland, V., Afanasyeva, E., Kuete, M., & Yoo, J. (2016). Chapter 15 - New Methods for Macro-Financial Model Comparison and Policy Analysis. In J. B. Taylor & H. Uhlig (Eds.), *Handbook of Macroeconomics* (Vol. 2, pp. 1241–1319). Elsevier.

Wieland, V., Cwik, T., Müller, G. J., Schmidt, S., & Wolters, M. (2012). A new comparative approach to macroeconomic modeling and policy analysis. *Journal of Economic Behavior & Organization*, *83*(3), 523–541.

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, *8*, 229–256.

Woodford, M. (2001). The taylor rule and optimal monetary policy. *The American Economic Review*, *91*(2), 232–237.

Woodford, M. (2003). *Foundations of a theory of monetary policy.* Princeton University Press.

Appendix A: Taylor Rules as Artificial Neural Networks

A.1 A Non-Linear Policy Rule

The neural network representing the policy function of the artificial central bank in the main text consists of an input layer, hidden layers, and an output layer.

- **Input Layer**: The input to the network is a vector $\mathbf{o}_t = \{o_{t,1}, o_{t,2}, \ldots, o_{t,n}\}$, representing the observed economic variables (inflation π_t, output gap x_t, etc.) at time t.

- **Hidden Layers**: Denote each hidden layer by an index l (e.g., $l = 1$ for the first hidden layer), with $\mathbf{h}^{(l)}$ representing the output of layer l computed based on input from previous layer $\left(\mathbf{h}^{(l-1)}\right)$. The input vector (or layer) $\mathbf{h}^{(0)} = \mathbf{o}_t$ is fed directly into the first hidden layer. I assume a fully connected architecture, i.e. each neuron in the hidden layer is connected to each neuron in the previous layer, with connections parametrised by weights $\mathbf{w}_j$ and bias b. A given neuron j in layer l outputs:

$$h_j^{(l)} = \zeta^{(l)} \left(\sum_i w_{ji}^{(l)} \, h_i^{(l-1)} + b_j^{(l)} \right)$$

  where $w_{ji}^{(l)}$ is the weight for the connection between neuron i in layer $l-1$ and neuron j in layer l, $b_j^{(l)}$ is the bias term for neuron j in layer l and $\zeta(\cdot)$ is an activation function.

- **Output Layer**: The output layer determines the policy action a_t, based on the activations from the final hidden layer.[67] Since PPO works with a stochastic policy, the output layer returns the mean parameter of the distribution of the policy.[68] Assuming the last hidden layer is L, I get:

$$\eta(\mathbf{s}_t; \theta) = \zeta^{(o)} \left(\sum_i w_i^{\eta} h_i^{(L)} + b^{\eta} \right)$$

[67] The reader should note that the methods do extend to multivariate action spaces, while I keep to a single decision variable for simplicity.

[68] As a default, the learning algorithm I am using assumes standard deviation of the policy as independent of the state and, therefore, the policy network does not output it. However, the standard deviation is still a part of the parametrisation of the policy θ and will be updated during the training. Alternatively, one can assume state-dependent variance and make standard deviation an explicit output of the policy network.

where w_i^η is the weights connecting the i-th neuron of the last hidden layer to the mean and b^η is the bias for the mean of the distribution.

The activation function $\zeta(\cdot)$ I use is the hyperbolic tangent (Tanh) function is

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

and maps any real number into the range [-1,1]. Commonly used activation functions are the (Leaky) Rectified Linear Unit (ReLU), Hyperbolic Tangent (Tanh), Sigmoid, Softplus and many others. For the hidden layers, I mostly rely on Tanh, as this has yielded best results in the simplified non-linear simulation. The output layers are kept linear. See Dubey, Singh, and Chaudhuri (2022) for a comprehensive overview of activation functions.

A.2 The Original Taylor (1993) Rule as an ANN

Consider the Taylor (1993) rule:

$$\mu^{TR}(\cdot) : i_t = r^* + \pi^* + \phi_\pi(\pi_t - \pi^*) + \phi_x x_t$$

with $r^* + \pi^* = 4\%$, $\phi_\pi = 1.5$ and $\phi_x = 0.5$. It is straightforward to represent this rule with an ANN. In terms of the notation from A.1 and the main text, the vector of observed variables is $\mathbf{o}_t = \{\pi_t, x_t\}$. Moreover, there will be no hidden layers, i.e. $L = 0$. The activation function of the output layer $\zeta^{(o)}$ will be the identity function, i.e. $\zeta^{(o)}$ will not transform the data. Furthermore, $\mathbf{h}^{(L)} = \mathbf{h}^{(0)}$ and thus $\mathbf{h}^{(L)}$ are the raw input observations $\mathbf{o}_t$. b^η would then be estimating the intercept which in (10) is $r^* + (1 - \phi_\pi)\pi^*$ and $\{w_1^\eta, w_2^\eta\} = \{\phi_\pi, \phi_x\}$. Note that you can represent any other common Taylor in this way and let an RL algorithm search for the weights and biases in a simulator.

Appendix B: The Optimal Decision Rule

This section examines how to analytically derive the optimal monetary policy for a particularly tractable parametrisation of the simulator. The first part of the section proves the Proposition .1 from the main text. The optimal solution to the decision problem (the theoretically best performing "Taylor-type rule") allowed me to compare the performance

of the artificial policymaker with the theoretically best-performing policy in Section 5.1. The second part analyses in greater detail the uniqueness of the solution.

For completeness, the third part of this section also outlines the LQR method, which demonstrates how to derive the optimal decision rule in the broad class of linear-quadratic-Gaussian (LQG) settings. These insights are used in the subsequent appendix section, which studies how the artificial agent's policy converges to the theoretically optimal one during learning in an LQG setting.

B.1 Proof of Proposition .1

Consider an economy given by (22)–(29) and assume that $\rho^{(d)} = \rho^{(s)} = \gamma_0^{(d)} = \gamma_0^{(s)} = 0$, the real natural rate is time invariant ($r_t^* = r^*$), $1 - \gamma_1^{(s)} = \gamma_2^{(s)}$, and that the zero lower bound is not binding ($i_t \in \mathbb{R}$). Further, assume that the central bank aims squarely at inflation stabilisation ($\lambda_\pi = 1$) and disregards other goals ($\lambda_x = \lambda_i = 0$). The central bank decides the optimal policy at time $t = t_0$ by choosing a sequence of interest rates $\{i_t\}_{t=t_0}^{\infty}$ to minimise a loss given constraints

$$\begin{aligned}
\min_{\{i_t\}_{t=t_0}^{\infty}} \quad & \mathcal{L} \equiv \mathbb{E}_{t_0}\left[\sum_{t=t_0+1}^{\infty} \beta^{t-t_0}(\pi_t - \pi^*)^2\right] \\
\text{s.t.} \quad & x_t = \gamma_1^{(d)} x_{t-1} + \gamma_2^{(d)}(i_{t-1} - \pi_{t+1}^e - r^*) + \varepsilon_t^{(d)} \\
& \pi_t = \gamma_1^{(s)} \pi_{t+1}^e + \gamma_2^{(s)} \pi_{t-1} + \gamma_3^{(s)} x_t + \varepsilon_t^{(s)} \\
& \pi_{t+1}^e = (1 - \vartheta_{t-1})\pi^* + \vartheta_{t-1}\pi_{t-1} + \varepsilon_t^{(e)} \\
& \vartheta_{t-1} = 1 - e^{-k|\pi_{t-1} - \pi^*|}
\end{aligned} \tag{B1}$$

The proof is composed of the following parts. First, I derive the first-order condition of the loss minimisation associated with the decision problem and derive two separate conditions that, when fulfilled by an appropriate sequence of decisions, ensure the FOC holds. Second, I guess a decision rule akin to Taylor (1993) and verify that this policy generates a sequence of decisions that fulfil the two conditions. Third, I argue that the minimisation problem is strictly convex in the control variable, such that fulfilling the FOC is the necessary *and* sufficient condition for the optimality of the proposed policy.

First, the FOC of the loss function with respect to the choice variable at time t_0,

treating $\{i_{t_1}, i_{t_2}, \dots\}$ as given, is:

$$\begin{aligned}
\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}i_{t_0}} &= \frac{\mathrm{d}\mathbb{E}_{t_0}\left[(\pi_{t_1}-\pi^*)^2 + \beta(\pi_{t_2}-\pi^*)^2 + \cdots\right]}{\mathrm{d}i_{t_0}} \\
&= \mathbb{E}_{t_0}\left[(\pi_{t_1}-\pi^*)\frac{\mathrm{d}\pi_{t_1}}{\mathrm{d}i_{t_0}} + \beta(\pi_{t_2}-\pi^*)\frac{\mathrm{d}\pi_{t_2}}{\mathrm{d}i_{t_0}} + \cdots\right] \overset{!}{=} 0 \\
&= \mathbb{E}_{t_0}\left\{\begin{array}{l}
\left[\pi_{t_1}-\pi^*\right]\dfrac{\partial\pi_{t_1}}{\partial x_{t_1}}\dfrac{\mathrm{d}x_{t_1}}{\mathrm{d}i_{t_0}} \\
+\beta\left[\pi_{t_2}-\pi^*\right]\left[\dfrac{\partial\pi_{t_2}}{\partial\pi^e_{t_3}}\dfrac{\mathrm{d}\pi^e_{t_3}}{\mathrm{d}i_{t_0}} + \dfrac{\partial\pi_{t_2}}{\partial\pi_{t_1}}\dfrac{\mathrm{d}x_{t_1}}{\mathrm{d}i_{t_0}}\right. \\
\qquad\left. + \dfrac{\partial\pi_{t_2}}{\partial x_{t_2}}\left(\dfrac{\partial x_{t_2}}{\partial x_{t_1}}\dfrac{\mathrm{d}x_{t_1}}{\mathrm{d}i_{t_0}} + \dfrac{\partial x_{t_2}}{\partial\pi^e_{t_3}}\dfrac{\mathrm{d}\pi^e_{t_3}}{\mathrm{d}i_{t_0}}\right)\right] + \cdots
\end{array}\right\} \overset{!}{=} 0
\end{aligned}$$

where the "$\cdots$" stand for the (discounted) terms associated with π_t where $t > t_2$. Note that adjusting i_{t_0} has an almost trivial effect on π_{t_1} (Term 1) but a complex effect on π_{t_2} and inflation further into the future. Specifically, i_{t_0} transmits into π_{t_2} over four distinct channels. First and second via its impact on inflation expectations $\pi^e_{t_3}$ and from there directly to π_{t_2} (Term 2) or via x_{t_2} and onwards to π_{t_2} (Term 5). Third via its impact on π_{t_1} (Term 3) which is a direct consequence of the initial effect on x_{t_1}, and finally via its impact on x_{t_2} (Term 4) via the initial effect on x_{t_1}. These feedback chains grow increasingly complex for inflation in $t > t_2$ and are omitted for tractability. For economically meaningful calibrations where $0 < \gamma_3^{(s)} < 1$ and $-1 < \gamma_2^{(d)} < 0$, these feedback effects from i_{t_0} into distant inflation realisations successively die out. Examining the individual differential terms in the FOC, I get:

$$\begin{aligned}
&\text{Term 1:} && \frac{\partial\pi_{t_1}}{\partial x_{t_1}}\frac{\mathrm{d}x_{t_1}}{\mathrm{d}i_{t_0}} = \gamma_3^{(s)}\,\gamma_2^{(d)}, \\
&\text{Term 2:} && \frac{\partial\pi_{t_2}}{\partial\pi^e_{t_3}}\frac{\mathrm{d}\pi^e_{t_3}}{\mathrm{d}i_{t_0}} = \gamma_1^{(s)}\,\gamma_3^{(s)}\,\gamma_2^{(d)}\Big[\vartheta_{t_1} + (\pi_{t_1}-\pi^*)\,k\,e^{-k|\pi_{t_1}-\pi^*|}\,\operatorname{sgn}\big(\pi_{t_1}-\pi^*\big)\Big], \\
&\text{Term 3:} && \frac{\partial\pi_{t_2}}{\partial\pi_{t_1}}\frac{\mathrm{d}x_{t_1}}{\mathrm{d}i_{t_0}} = \gamma_2^{(s)}\,\gamma_3^{(s)}\,\gamma_2^{(d)}, \\
&\text{Term 4:} && \frac{\partial\pi_{t_2}}{\partial x_{t_2}}\frac{\partial x_{t_2}}{\partial x_{t_1}}\frac{\mathrm{d}x_{t_1}}{\mathrm{d}i_{t_0}} = \gamma_1^{(d)}\,\gamma_3^{(s)}\,\gamma_2^{(d)}, \\
&\text{Term 5:} && \frac{\partial\pi_{t_2}}{\partial x_{t_2}}\frac{\partial x_{t_2}}{\partial\pi^e_{t_3}}\frac{\mathrm{d}\pi^e_{t_3}}{\mathrm{d}i_{t_0}} = -\Big(\gamma_3^{(s)}\Big)^2\Big(\gamma_2^{(d)}\Big)^2\Big[\vartheta_{t_1} + (\pi_{t_1}-\pi^*)\,k\,e^{-k|\pi_{t_1}-\pi^*|}\,\operatorname{sgn}\big(\pi_{t_1}-\pi^*\big)\Big].
\end{aligned} \tag{B2}$$

where

$$\operatorname{sgn}(x) = \begin{cases} -1, & \text{if } x < 0, \\ 1, & \text{if } x > 0. \end{cases}$$

The first-order condition can be rewritten as

$$\begin{aligned} \frac{\mathrm{d}\mathcal{L}}{\mathrm{d}i_{t_0}} = \mathbb{E}_{t_0}\left[\pi_{t_1} - \pi^*\right] \times \text{Term 1} + \beta\,\mathbb{E}_{t_0}\left[\pi_{t_2} - \pi^*\right] \times \text{Term 3} + \beta\,\mathbb{E}_{t_0}\left[\pi_{t_2} - \pi^*\right] \times \text{Term 4} \\ + \beta\,\mathbb{E}_{t_0}\left[(\pi_{t_2} - \pi^*) \times \text{Term 2}\right] + \beta\,\mathbb{E}_{t_0}\left[(\pi_{t_2} - \pi^*) \times \text{Term 5}\right] + \cdots \end{aligned}$$

If the solution ensures that the inflation deviations are serially uncorrelated, i.e. that

$$\mathbb{E}_{t_0}\Big[(\pi_{t+1} - \pi^*)(\pi_{t-k} - \pi^*)\Big] = 0, \quad \forall\, t, k \text{ where } t - k > t_0,\ k \geq 0, \tag{B3}$$

then I can factor out the multiplication inside the expectation operators associated with Term 2 and Term 5 into two distinct expectation terms and rewrite the FOC as

$$\begin{aligned} \frac{\mathrm{d}\mathcal{L}}{\mathrm{d}i_{t_0}} = \mathbb{E}_{t_0}\left[\pi_{t_1} - \pi^*\right] \times \text{Term 1} + \beta\,\mathbb{E}_{t_0}\left[\pi_{t_2} - \pi^*\right] \times \text{Term 3} + \beta\,\mathbb{E}_{t_0}\left[\pi_{t_2} - \pi^*\right] \times \text{Term 4} \\ + \beta\,\mathbb{E}_{t_0}\left[(\pi_{t_2} - \pi^*)\right] \times \mathbb{E}_{t_0}\left[\text{Term 2}\right] + \beta\,\mathbb{E}_{t_0}\left[(\pi_{t_2} - \pi^*)\right] \times \mathbb{E}_{t_0}\left[\text{Term 5}\right] + \cdots \end{aligned}$$

If the solution also ensures that in expectations, inflation deviations are zero, i.e.

$$\mathbb{E}_{t_0}\Big[(\pi_t - \pi^*)\Big] = 0 \quad \forall\, t > t_0 \tag{B4}$$

it immediately follows that

$$\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}i_{t_0}} = 0$$

and the first-order condition of the decision problem is fulfilled. I guess that an interest rate sequence generated by a Taylor-type decision rule can ensure that inflation satisfies conditions (B3) and (B4). Consider the following Taylor-type decision rule μ:

$$\mu(\cdot) : i_t = r^* + \pi^* + \phi_{\pi,t}(\pi_t - \pi^*) + \phi_{x,t} x_t \tag{B5}$$

where the coefficients $\phi_{\pi,t}$ and $\phi_{x,t}$ are to be determined. Now, verify that this indeed satisfies the two conditions. Plugging (B5) into the constraints of the decision problem in

(B1), the law of motion for inflation becomes

$$\begin{aligned}\pi_{t+1} = \pi^* &+ \gamma_2^{(s)}\pi_t - (1-\gamma_1^{(s)})\pi^* \\ &+ \left[\vartheta_t(\gamma_1^{(s)} - \gamma_3^{(s)}\gamma_2^{(d)}) + \gamma_3^{(s)}\gamma_2^{(d)}\phi_{\pi,t}\right](\pi_t - \pi^*) \\ &+ \gamma_3^{(s)}\left[\gamma_1^{(d)} + \gamma_2^{(d)}\phi_{x,t}\right]x_t \\ &+ (\gamma_1^{(s)} - \gamma_3^{(s)}\gamma_2^{(d)})\varepsilon_{t+1}^{(e)} + \gamma_3^{(s)}\varepsilon_{t+1}^{(d)} + \varepsilon_{t+1}^{(s)}\end{aligned}$$

Using $1-\gamma_1^{(s)} = \gamma_2^{(s)}$ and eliminating $\gamma_2^{(s)}$ from the system, I get

$$\begin{aligned}\pi_{t+1} = \pi^* &+ \left[1 - \gamma_1^{(s)} + \vartheta_t(\gamma_1^{(s)} - \gamma_3^{(s)}\gamma_2^{(d)}) + \gamma_3^{(s)}\gamma_2^{(d)}\phi_{\pi,t}\right](\pi_t - \pi^*) \\ &+ \gamma_3^{(s)}\left[\gamma_1^{(d)} + \gamma_2^{(d)}\phi_{x,t}\right]x_t \\ &+ (\gamma_1^{(s)} - \gamma_3^{(s)}\gamma_2^{(d)})\varepsilon_{t+1}^{(e)} + \gamma_3^{(s)}\varepsilon_{t+1}^{(d)} + \varepsilon_{t+1}^{(s)}\end{aligned} \tag{B6}$$

Rewrite equation (B6) to get

$$\mathbb{E}_{t_0}\left[\pi_{t+1} - \pi^*\right] = \left[1-\gamma_1^{(s)}+\vartheta_t(\gamma_1^{(s)}-\gamma_3^{(s)}\gamma_2^{(d)})+\gamma_3^{(s)}\gamma_2^{(d)}\phi_{\pi,t}\right](\pi_t-\pi^*)+\gamma_3^{(s)}\left[\gamma_1^{(d)} + \gamma_2^{(d)}\phi_{x,t}\right]x_t \tag{B7}$$

Based on (B7), I can directly see the optimal coefficients $\phi_{\pi,t}^*$ and $\phi_{x,t}^*$ which ensure that condition (B4) holds, namely

$$\phi_{\pi,t}^* = \vartheta_t - \frac{1-\gamma_1^{(s)}(1-\vartheta_t)}{\gamma_3^{(s)}\gamma_2^{(d)}} \tag{B8}$$

$$\phi_{x,t}^* = -\frac{\gamma_1^{(d)}}{\gamma_2^{(d)}} \tag{B9}$$

Furthermore, plugging the coefficients (B8) and (B9) into the DGP for inflation in equation (B6), I see that the DGP for inflation deviations fulfils the condition (B3) and the inflation deviations evolve as a sum of white noise error terms

$$\pi_{t+1} - \pi^* = (\gamma_1^{(s)} - \gamma_3^{(s)}\gamma_2^{(d)})\,\varepsilon_{t+1}^{(e)} + \gamma_3^{(s)}\varepsilon_{t+1}^{(d)} + \varepsilon_{t+1}^{(s)}$$

Thus, both optimality conditions (B3) and (B4) hold, and the proposed policy fulfils the necessary condition to be a solution to the decision problem. The proposed Taylor-type rule fully "neutralises" the effect of the control variable (i_t) on inflation deviations ($\pi_t - \pi^*$) and transforms the DGP for inflation into a linear sum of error terms. Note that

the Proposition .1 assumed $\gamma_1^{(d)} = \gamma_2^{(s)} = 0$.

I can now derive the second order condition of the optimisation problem, i.e. $\frac{\mathrm{d}^2\mathcal{L}}{\mathrm{d}i_{t_0}^2}$. To see how to do this, rearrange the FOC as

$$\begin{aligned}\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}i_{t_0}} &= \mathbb{E}_{t_0}\left[(\pi_{t_1} - \pi^*)\frac{\mathrm{d}\pi_{t_1}}{\mathrm{d}i_{t_0}} + \beta(\pi_{t_2} - \pi^*)\frac{\mathrm{d}\pi_{t_2}}{\mathrm{d}i_{t_0}} + \cdots\right] \\ &= \mathbb{E}_{t_0}\left[\pi_{t_1}\frac{\mathrm{d}\pi_{t_1}}{\mathrm{d}i_{t_0}} - \pi^*\frac{\mathrm{d}\pi_{t_1}}{\mathrm{d}i_{t_0}} + \beta\pi_{t_2}\frac{\mathrm{d}\pi_{t_2}}{\mathrm{d}i_{t_0}} - \beta\pi^*\frac{\mathrm{d}\pi_{t_2}}{\mathrm{d}i_{t_0}} + \cdots\right]\end{aligned}$$

Taking the second derivative yields

$$\frac{\mathrm{d}^2\mathcal{L}}{\mathrm{d}i_{t_0}^2} = \mathbb{E}_{t_0}\left[\left(\frac{\mathrm{d}\pi_{t_1}}{\mathrm{d}i_{t_0}}\right)^2 + (\pi_{t_1} - \pi^*)\frac{\mathrm{d}^2\pi_{t_1}}{\mathrm{d}i_{t_0}^2} + \beta\left(\frac{\mathrm{d}\pi_{t_2}}{\mathrm{d}i_{t_0}}\right)^2 + \beta(\pi_{t_2} - \pi^*)\frac{\mathrm{d}^2\pi_{t_2}}{\mathrm{d}i_{t_0}^2} + \cdots\right]$$

And now note that three of the five terms in (B2) collapse to zero when differentiated with respect to i_{t_0}. Since $\frac{\mathrm{d}\pi_{t_1}}{\mathrm{d}i_{t_0}} = \text{Term 1}$ and $\frac{\mathrm{d}\pi_{t_2}}{\mathrm{d}i_{t_0}} = \text{Term 2} + \text{Term 3} + \text{Term 4} + \text{Term 5}$, I know that $\frac{\mathrm{d}^2\pi_{t_1}}{\mathrm{d}i_{t_0}^2} = 0$ and $\frac{\mathrm{d}^2\pi_{t_2}}{\mathrm{d}i_{t_0}^2} = \frac{\mathrm{dTerm\ 2}}{\mathrm{d}i_{t_0}} + \frac{\mathrm{dTerm\ 5}}{\mathrm{d}i_{t_0}}$. [69] Therefore, I am left with

$$\frac{\mathrm{d}^2\mathcal{L}}{\mathrm{d}i_{t_0}^2} = \mathbb{E}_{t_0}\left[\left(\frac{\mathrm{d}\pi_{t_1}}{\mathrm{d}i_{t_0}}\right)^2 + \beta\left(\frac{\mathrm{d}\pi_{t_2}}{\mathrm{d}i_{t_0}}\right)^2 + \beta(\pi_{t_2} - \pi^*)\left(\frac{\mathrm{dTerm\ 2}}{\mathrm{d}i_{t_0}} + \frac{\mathrm{dTerm\ 5}}{\mathrm{d}i_{t_0}}\right) + \cdots\right] \quad \text{(B10)}$$

I can show that the derivatives of the two non-linear terms are

$$\begin{aligned}\frac{\mathrm{dTerm\ 2}}{\mathrm{d}i_{t_0}} &= \gamma_1^{(s)}\left(\gamma_3^{(s)}\right)^2\left(\gamma_2^{(d)}\right)^2 k\, e^{-k\left|\pi_{t_1} - \pi^*\right|}\left[2\ \mathrm{sgn}\left(\pi_{t_1} - \pi^*\right) - k\ (\pi_{t_1} - \pi^*)\right] \\ \frac{\mathrm{dTerm\ 5}}{\mathrm{d}i_{t_0}} &= -\left(\gamma_3^{(s)}\right)^3\left(\gamma_2^{(d)}\right)^3 k\, e^{-k\left|\pi_{t_1} - \pi^*\right|}\left[2\ \mathrm{sgn}\left(\pi_{t_1} - \pi^*\right) - k\ (\pi_{t_1} - \pi^*)\right]\end{aligned}$$

Note that both non-linear terms are functions of past inflation and that I am interested in the global convexity of the *expected* losses. And so it can be proven that for *any* policy that achieves no inflation bias (property (B4)) and serially decorrelates inflation (property

[69] Derivatives $\frac{\mathrm{d}^2\pi_t}{\mathrm{d}i_{t_0}^2}$ where $t > t_2$ will not equal zero, because interest rates propagate non-linearly through expectations into $\pi_{t_3}^e$ and onward. However, the terms in these derivatives will be multiplied by higher exponents of γ_3

(B3)), the expected loss function is strictly convex because

$$\begin{aligned}\frac{\mathrm{d}^2\mathcal{L}}{\mathrm{d}i_{t_0}^2} &= \left(\frac{\mathrm{d}\pi_{t_1}}{\mathrm{d}i_{t_0}}\right)^2 + \beta\left(\frac{\mathrm{d}\pi_{t_2}}{\mathrm{d}i_{t_0}}\right)^2 + \beta\mathbb{E}_{t_0}\left[(\pi_{t_2}-\pi^*)\right]\mathbb{E}_{t_0}\left[\left(\frac{\mathrm{dTerm\ 2}}{\mathrm{d}i_{t_0}} + \frac{\mathrm{dTerm\ 5}}{\mathrm{d}i_{t_0}}\right)\right] + \mathbb{E}_{t_0}\left[\cdots\right] \\ &= \left(\frac{\mathrm{d}\pi_{t_1}}{\mathrm{d}i_{t_0}}\right)^2 + \beta\left(\frac{\mathrm{d}\pi_{t_2}}{\mathrm{d}i_{t_0}}\right)^2 + \cdots\end{aligned}$$

where all terms in the infinite sum are squared and, hence, strictly positive. Therefore, for the subset of policies that achieve no inflation bias and no serial correlation in inflation, the sequence of interest rate decisions generated by the proposed policy rule is the unique solution to the decision problem. The FOC is a necessary and sufficient condition for the optimality of my solution. ■

B.2 Further Results on the Uniqueness of the Optimal Decisions

For some parameter values, it can be shown that the second derivative of the expected loss is strictly convex for a wide range of inflation deviations from the steady state. For those parametrisations, the sequence of decisions generated by the proposed decision rule is unique in a large region around π^*, even when considering policies that do *not* achieve properties (B3) and (B4) in equilibrium. This can be shown by evaluating (B10) for different parameter values. Define the second derivative of the expected loss function as $D(x_1, x_2)$, i.e. $D(x_1, x_2) \equiv \frac{\mathrm{d}^2\mathcal{L}}{\mathrm{d}i_{t_0}^2}$

$$D(x_1,x_2) = \mathbb{E}_{t_0}\left\{A^2 \;+\; \beta\left[B(x_1)\right]^2 \;+\; \beta\, x_2\left[A^2\Big(\gamma_1^{(s)} - A\Big)F'(x_1)\right] + \cdots\right\}$$

where

$$\begin{aligned} x_1 &= \pi_{t_1} - \pi^*, \\ x_2 &= \pi_{t_2} - \pi^*, \\ A &= \gamma_3^{(s)}\,\gamma_2^{(d)}, \\ C_1 &= \gamma_1^{(s)}, \qquad C_2 = \gamma_2^{(s)} + \gamma_1^{(d)}, \\ F(x_1) &= 1 - e^{-k|x_1|} + k\,|x_1|\,e^{-k|x_1|}, \\ F'(x_1) &= k\,e^{-k|x_1|}\Big(2 - k\,|x_1|\Big)\ \mathrm{sgn}(x_1), \\ B(x_1) &= A\,C_2 + A\Big(C_1 - A\Big)\,F(x_1). \end{aligned}$$

For example for the reasonable parametrisation $\gamma_1^{(s)} = 0.5$, $\gamma_2^{(s)} = 1 - \gamma_1^{(s)} = 0.5$, $\gamma_3^{(s)} = 0.25$, $\gamma_1^{(d)} = 0.9$, $\gamma_2^{(d)} = -0.1$, $k = 0.15$, $\beta = 0.9$, I can show that $D(x_1, x_2)$ is positive for any combination of x_1, x_2 where $x_1 \in [-10, 10]$ and $x_2 \in [-10, 10]$.

B.3 LQR: The General Linear Quadratic Case

For cases where $\lambda_x \neq 0$ or $\lambda_i \neq 0$, the decision problem becomes more complex. The central bank will need to trade off both intra- and inter-temporally between achieving inflation at its goal, closing the output gap and smoothing interest rates. For example, keeping the DGP of expected inflation at its goal will be associated with additional costs, as this requires both interest rate variation and output gaps. To solve for the optimal Taylor rule coefficients in a more general case, a researcher can instead use the theory behind Linear Quadratic Gaussian (LQR) regulators.[70] However, one must abandon the non-linear specification of ϑ. Specifically, I assume that $\vartheta_t = \vartheta$ is a constant.

The optimal control rule can be written in matrix notation as $u_t = Fy_t$, where the matrix F contains the Taylor rule coefficients.[71] In our case, y_t will be a (3x1) matrix (two state variables and one to represent the constants). u_t is a scalar (the interest rate). By implication F will be (1x3), giving me ϕ_π, ϕ_x and the constant of the Taylor rule. I can write the loss function in general form as:

$$\mathbb{E}_{t_0} \left\{ \sum_{t=t_0+1}^{\infty} \beta^{t-t_0} \left(y_t' R y_t + u_t' Q u_t + 2 u_t' N x_t \right) \right\} \tag{B11}$$

[70] For more context, see Hansen and Sargent (2011).

[71] Note that u_t is often used to denote the control variable (here: interest rates) in optimal control literature. Previously, I have used u_t to denote AR(1) disturbances, but since I only consider white noise shocks in this appendix, u_t is taken to denote generic control variable(s) in the LQR problem. For the state variables, I use y_t so as not to confuse this with the output gap x_t.

where

$$
\begin{aligned}
y_t &= \begin{pmatrix} \pi_t - \pi^* \\ x_t \\ 1 \end{pmatrix} \\
u &= i_t \\
R &= \begin{pmatrix} \lambda_\pi & 0 & 0 \\ 0 & \lambda_x & 0 \\ 0 & 0 & 0 \end{pmatrix} \\
Q &= \lambda_i \\
N &= 0
\end{aligned}
$$

corresponds to the setup of the linear simulator. Furthermore, I know that

$$F = (Q + \beta B'PB)^{-1}\beta B'PA \tag{B12}$$

where the matrix P is the solution to the Riccati equation:

$$P = R - (\beta B'PA + N)'(Q + \beta B'PB)^{-1}(\beta B'PA + N) \tag{B13}$$

where the matrices A, B, C come from the system's dynamics written as $y_{t+1} = Ay_t + Bu_t + Cw_{t+1}$. For the linear simulator above, I get

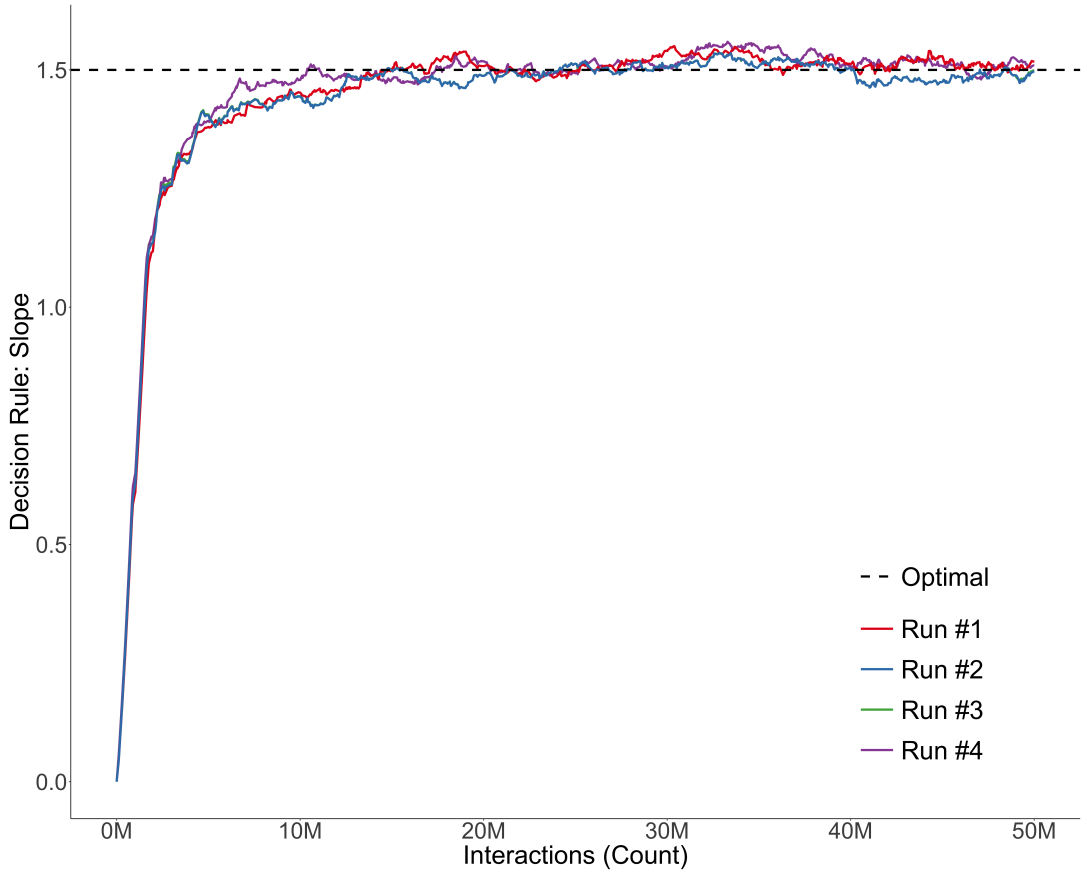
$$
\begin{aligned}
A &= \begin{pmatrix} \gamma_1^{(s)}\vartheta + \gamma_2^{(s)} - \gamma_3^{(s)}\gamma_2^{(d)}\vartheta & \gamma_3^{(s)}\gamma_1^{(d)} & \left(\gamma_1^{(s)} + \gamma_2^{(s)} - \gamma_3^{(s)}\gamma_2^{(d)} - 1\right)\pi^* - \gamma_3^{(s)}\gamma_2^{(d)}r^* + \gamma_0^{(s)} + \gamma_3^{(s)}\gamma_0^{(d)} \\ -\gamma_2^{(d)}\vartheta & \gamma_1^{(d)} & -\gamma_2^{(d)}(r^* + \pi^*) + \gamma_0^{(d)} \\ 0 & 0 & 1 \end{pmatrix} \\
B &= \begin{pmatrix} \gamma_3^{(s)}\gamma_2^{(d)} \\ \gamma_2^{(d)} \\ 0 \end{pmatrix} \\
C &= \begin{pmatrix} \sqrt{\left(\gamma_3^{(s)}\right)^2 \sigma^2_{\varepsilon^{(d)}} + \sigma^2_{\varepsilon^{(s)}}} \\ \sigma_{\varepsilon^{(d)}} \\ 0 \end{pmatrix}
\end{aligned}
$$

I can calculate P and F, which gives the optimal ϕ_π^*, ϕ_x^* and a constant.[72] The relationship between the value function and the matrix P is $V = y'Py$.

Appendix C: Learning Inside the Linear Simulator

I can now look at a particularly tractable case to examine how the artificial policymaker learns. I assume $\left\{\gamma_0^{(s)}, \gamma_1^{(s)}, \gamma_2^{(s)}, \gamma_3^{(s)}, \gamma_0^{(d)}, \gamma_1^{(d)}, \gamma_2^{(d)}, \vartheta, r^*, \pi^*\right\} = \{0, 1, 0, 1, 0, 0, -0.5, 0.5, 2, 2\}$. This leads to the optimal Taylor rule coefficients being $\{\phi_\pi^*, \phi_x^*, \text{constant}^*\} = \{1.5, 0, 4\}$. I let the neural network for the artificial central bank's policy be single-layer linear with identity activation to match this structure. I simplify the value function neural net to two hidden layers of 4 nodes, each with the Tanh activation function. The artificial policymaker observes only period t inflation π_t when deciding interest rates. All other hyperparameters are kept as in the baseline setting described in the main text.

Figure C1: Learning: Weight of the Policy Network



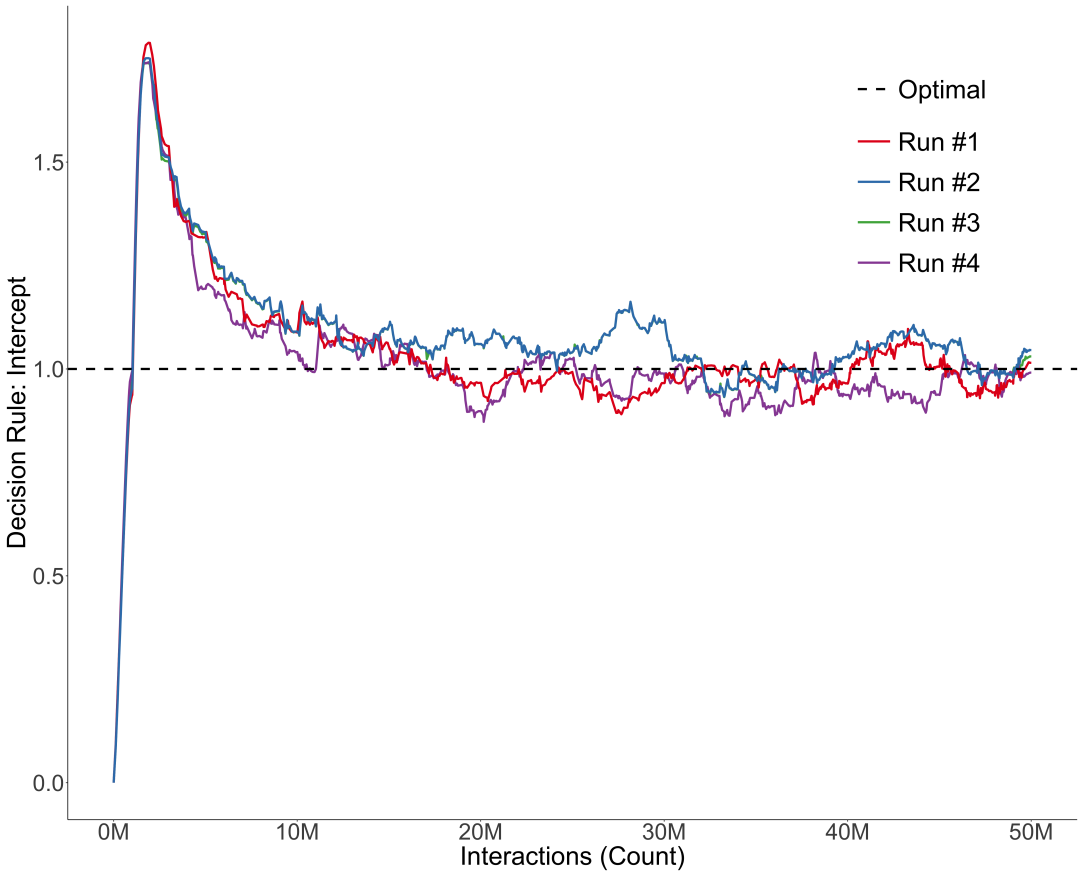
[72] I use the implementation from the QuantEcon Python package for this. See Batista et al. (2024).

Figure C1 plots the trajectory of the single "weight" parameter of the policy network over the total interaction count in millions (the number of times a decision has been made in the simulator). This weight corresponds directly to ϕ_π. In the optimum, this value should converge to $\phi_\pi^* = 1.5$. Analogously, Figure C2 plots this trajectory for the bias parameter of the policy network, corresponding to the intercept in the Taylor rule. In the optimum, this value should converge to $r^* + (1 - \phi_\pi^*)\pi^* = 1$ because $i_t = r^* + \pi^* + \phi_\pi^*(\pi_t - \pi^*) = r^* - (1 + \phi_\pi^*) + \phi_\pi^*\pi_t = \text{bias} + \text{weight} \cdot \pi_t$. Both figures show that the values approach the optimal ones already after an initial period of approx. 10 million decisions (interactions) within the simulator. Although this seems like a large number, simulating 10 million interactions takes only around 20 minutes when running in parallel on nine cores of the Apple M1 Max processor. Because of the stochasticity in generating experience as well as in learning (gradient descent), the learning trajectories of different runs can differ substantially. It is not clear whether and how quickly the algorithm will converge to the true solution; in fact, the agent keeps learning forever and, even after a large number of interactions, has not *settled* on the true value. This behaviour is expected, and therefore, in a straightforward linear case, the standard methods like LQR are far more suitable. Nevertheless, this exercise helps to see the learning in action and test whether the algorithm behaves as desired.

Perhaps, unsurprisingly, initialising the weights and biases of the neural network can be very important for convergence. When initialised closer to the true values, the network approximates the true values with as little as ≈ 1 million interactions (≈ 2 minutes). This can be seen in run #5 in Figure C3, where the intercept and slope have been initialised at 0.5 and 1, respectively. Run #1 is shown for comparison. Somewhat counter-intuitively, run #6, which relies on initialisation farther away from true values (3,3 instead of 0,0), converges much quicker than run #1. I hypothesise that this is because higher values of intercept and slope are more likely to stabilise inflation in the simulator, making it less prone to end in hyper-inflations or deflations, which stabilises the learning.

Before training the agent, the researcher should always consider whether the setup

Figure C2: Learning: Bias of the Policy Network

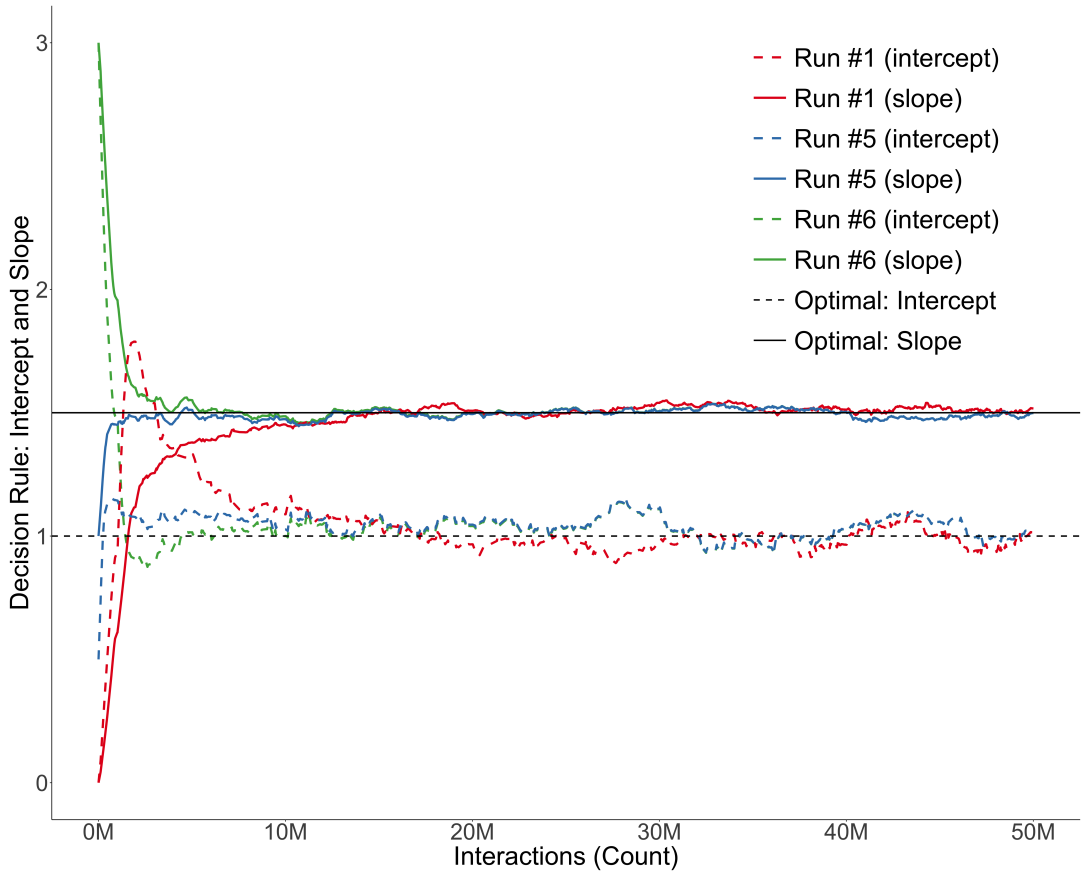


enables the artificial agent to *explore* a well-performing policy.[73] I demonstrate this with a somewhat extreme example. Say I initialise the weight and bias far from the true value and do not carefully consider how the agent *explores* the environment. The agent might then end up never converging to the true values. Consider the example shown in Figure C4, where the slope and intercept were both initialised with the negative value of -1. [74]

[73] This is a subtle point, yet fundamental questions to consider are: does the artificial policymaker have enough tools to perform well (example: what is the highest level of interest rates it is allowed to set in the simulator and is this enough to counter inflation?) and will the algorithm let the policymaker try that potentially well-performing policy? (For example, when exploring, does the artificial policymaker ever set the interest rates to the level that would be dictated by an appropriate benchmark policy?)

[74] There are many candidate reasons for observing a lack of convergence, or even divergence after an initial convergence, of an agent. One possibility is an inappropriately chosen learning rate. For example, the learning rate might be too high for the policy to settle in a (perhaps local) optimum. It might also be too low such that convergence takes too long or settles in a local instead of global optimum. Another possibility is that the neural networks used are too simple or even too complex for the environment (simulator). The researcher also needs to consider whether the agent explores different policies sufficiently. Uncovering the reason for the lack of convergence to a well-enough performing policy can be a time-consuming process where a lot of different nuts and bolts of the algorithm might need to be tweaked. However, for the problem of the central bank analysed here, once I made sure *a well-performing policy can be explored by the artificial policymaker*, the algorithm always converged to a practical policy.

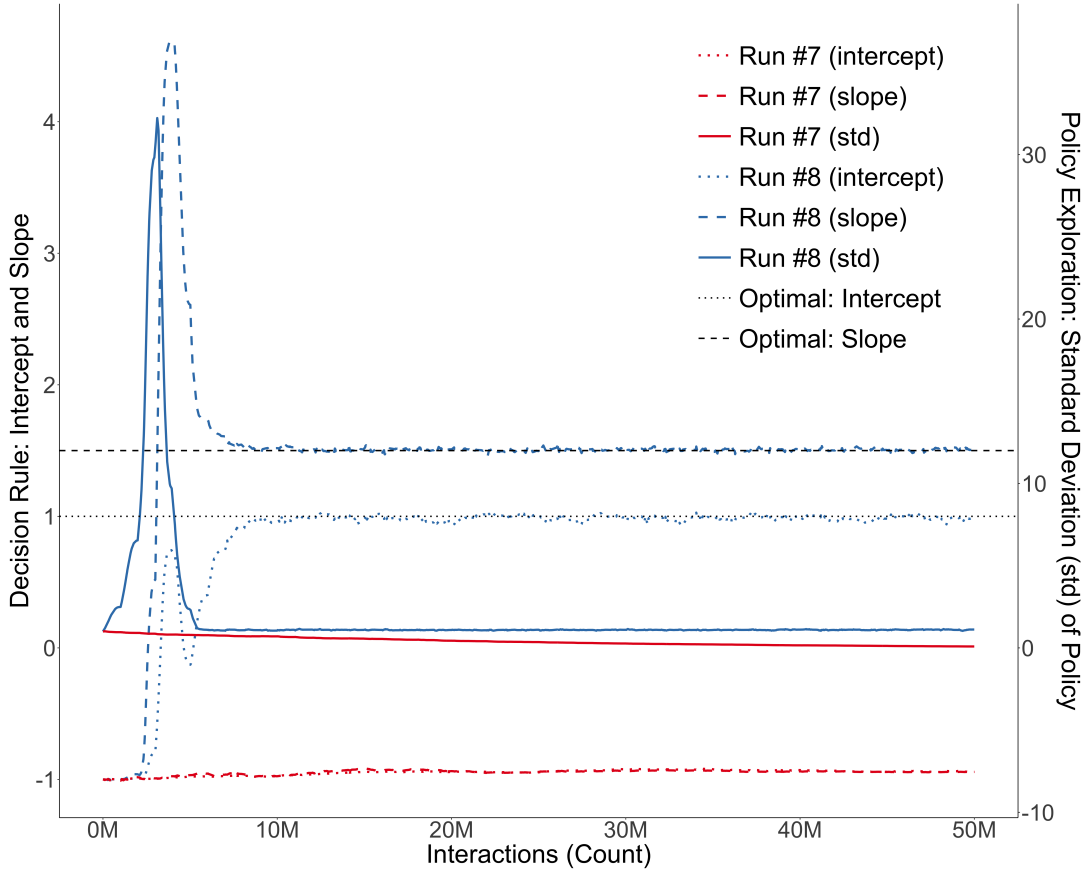
Figure C3: Convergence for Different Initialisations



In run #7, even after completing 50 million interactions, the agent has not learnt a sensible policy with the parametrisation still stuck close to the initial values. Why? Because the artificial agent does *not even try* a policy which would substantially improve its rewards. With negative initialisation, the policymaker is choosing negative interest rates during the episode, which quickly leads to an inflationary spiral that would require even more positive interest rates, which the agent does *not* try. One possibility to solve this problem is to force the agent to explore more by increasing the weight of entropy in the PPO objective (parameter c_2 in (15), which is set to 0.01 in the main text). This will enable the artificial policymaker to endogenously increase the standard deviation of its policy and explore a wider range of interest rates in the simulator. Compare the solid line for run #7, which slowly decreases towards zero from its initial value of 1, with the solid line for run #8, where the standard deviation of the policy increases substantially over training, which eventually leads to the discovery of a well-performing policy. While the agent in run #7 has grown increasingly sure the policy cannot be improved and stopped exploring, the agent in run #8 eventually "realised" he needs to try a wider range of interest rates to be

successful.[75]

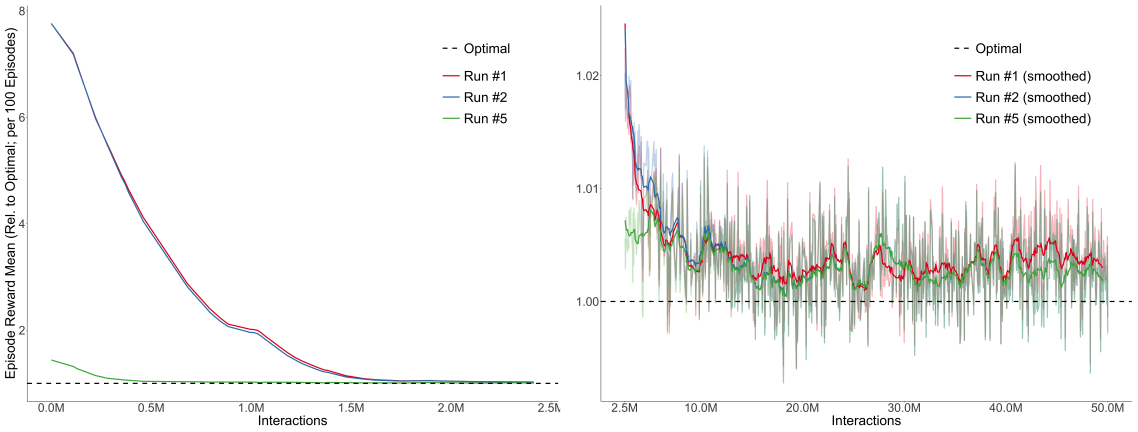
Figure C4: Policy Exploration is Crucial for Successful Learning



Lastly, examine Figure C5. It shows the convergence of three different runs in terms of average loss from 100 episodes over the training relative to the theoretical optimum. The policy quickly becomes "good-enough" but never entirely matches the optimum. After 2.5 million interactions, the loss is in the area of being 2% worse for RL than the optimal policy. Afterwards, loss starts to oscillate around $0.1\% - 0.5\%$ penalty vs. optimum. Although it might appear that RL sometimes performs better than the optimal policy (non-smoothed lines below 1.0), this is an artefact of the stochasticity of the environment (100 episodes are not enough to average out the impact of "good" shocks on the performance).

[75] Besides tweaking the c_2 hyperparameter in the surrogate loss of the PPO algorithm, there is a simpler solution which could prevent such lack of convergence and stabilise learning. It is common (and suggested by Stable Baselines 3) to normalise the action space to [-1,1] and then rescale these actions inside of the simulator. This should ensure that, initially, all actions can be tried out with high probability as the initial standard deviation of the policy is equal to 1. To further improve the stability of the learning, observations and even rewards are usually normalised as well. I follow this practice in the main text.

Figure C5: Average Episode Losses over Learning Relative to Optimum



When deciding whether to use RL as proposed in this paper, it is important to consider the following practical issues. First, although the algorithm usually approaches a "good-enough" policy quickly (here: 0.5% larger than optimal loss after 20 minutes of runtime[76]), the policymaker never stops learning, and the values begin to oscillate in the close vicinity of the optimal values. Therefore, it is important to realise that this process does not uncover the *exact* true optimal coefficients. Instead, the policymaker finds and then constantly re-updates a sub-optimal policy, which should nevertheless be performing satisfactorily. Secondly, in settings where the researcher knows the true optimal policy analytically, or when the optimal policy can be easily calculated using numeric methods with guaranteed and fast convergence, the RL approach will be inferior. Thirdly, if one does not choose appropriate values for the hyperparameters, does not initialise or does not choose the architecture of the neural networks well, the process might entirely fail as Run #7 in Figure C4 demonstrates. The learning process might also fail by pure chance as when stuck in a local optimum. However, the economies I simulate in this paper seem straightforward enough, such that I have not encountered catastrophic convergence issues with standard hyperparameter values and neural network architectures. Fourthly, when evaluating the policy during training to determine which policies to keep and which to discard, the researcher needs to make sure the policy is not performing well at random

[76]This approximate time-frame *scales* to all the settings examined in this paper (more arguments in the Taylor rule, non-linearity of the simulator, etc.). I do not run more than 10 million iterations for the results reported in the main text. Simulating a non-linear economy is not significantly more computationally demanding than the linear case examined here.

(such as because of many randomly drawn "good" shocks) but that it performs well when the randomness of the simulator is appropriately accounted for. Therefore, the researcher needs to evaluate a candidate policy for a sufficient number of episodes in the simulator when deciding the best-performing policy. This can be time-consuming, especially if small changes in policy parameters lead to considerable performance changes. Lastly, because of the substantial randomness involved in the learning process and simulations, exactly replicating a learning trajectory is challenging.

Appendix D: Empirical Fit of the Full Non-linear Simulator

Figure D1: Inflation (Actual vs. Predicted). Adjusted $R^2 = 0.84$.

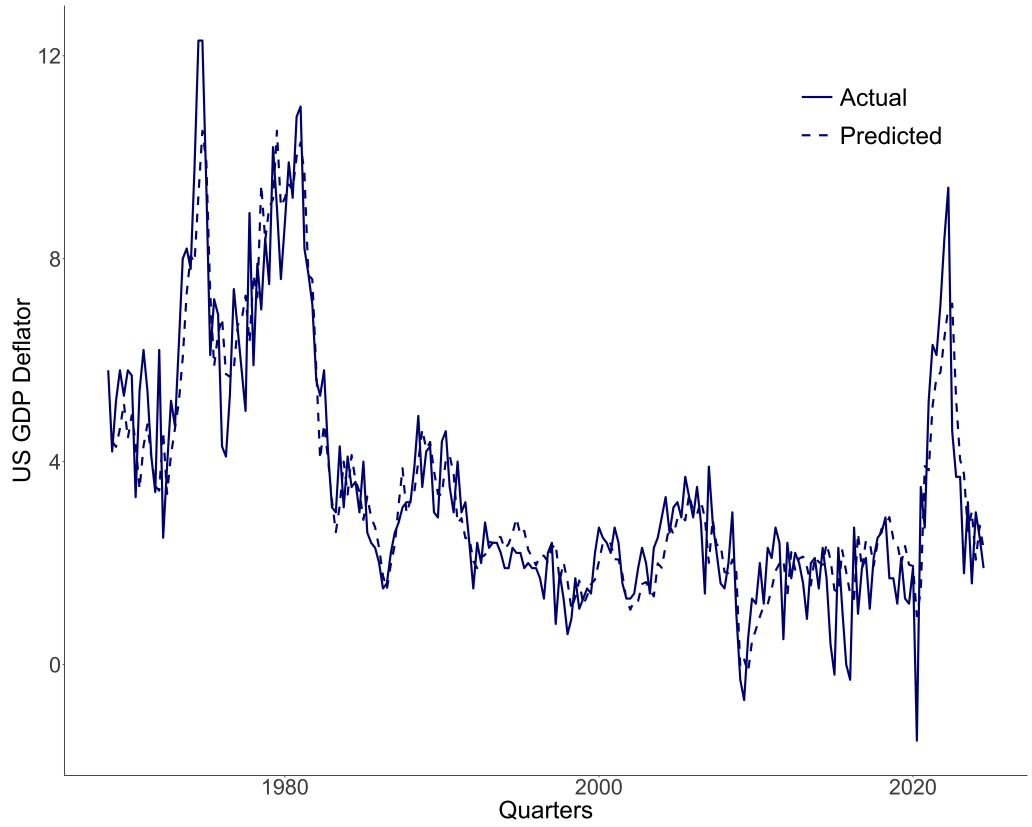
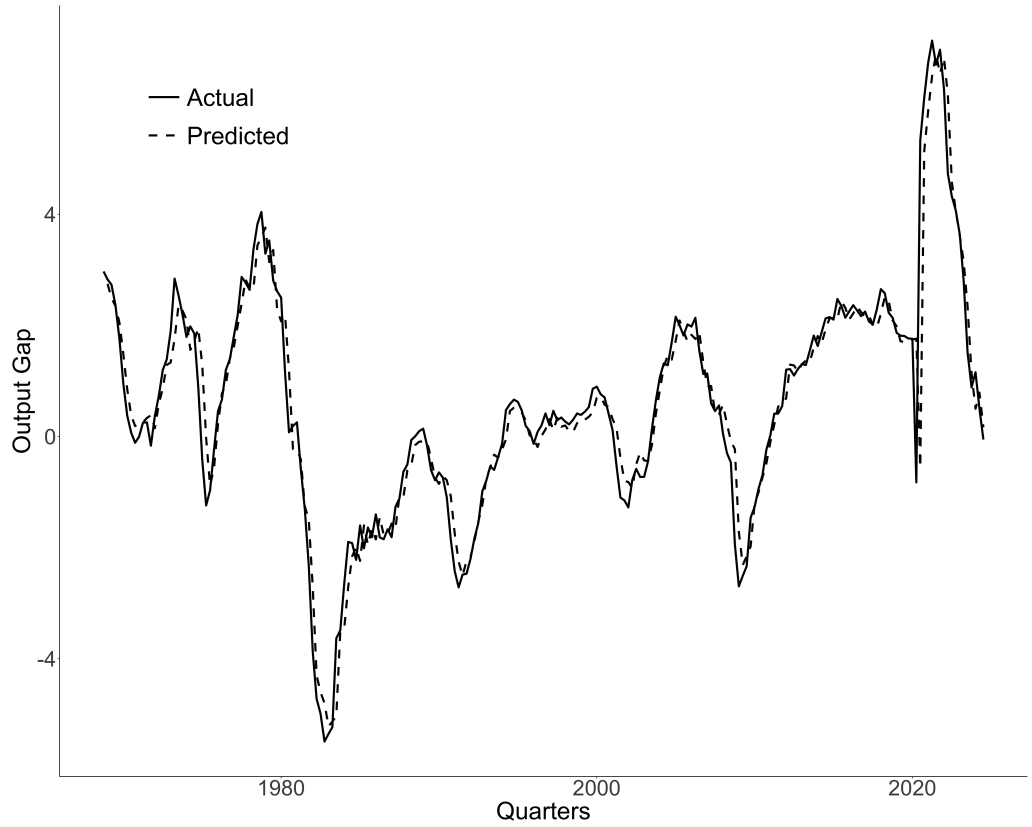


Figure D2: Output Gap (Actual vs. Predicted). Adjusted $R^2 = 0.92$.



Appendix E: Auxiliary Figures

Figure E1: Example of an Episode (Quarter 0 to 75)

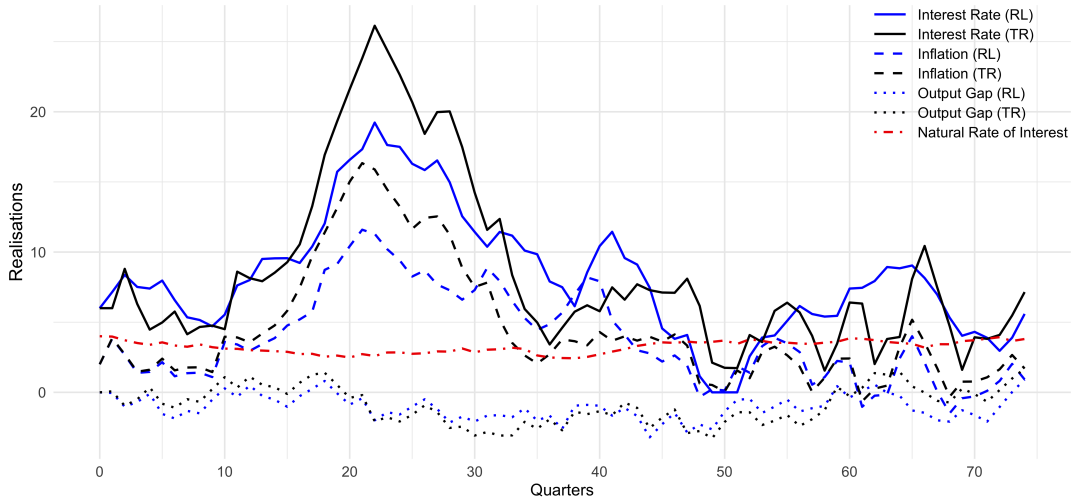


Figure E2: Policy Heatmap (Taylor (1993) rule)