UNIVERSIDADE DE SÃO PAULO
FACULDADE DE ECONOMIA, ADMINISTRAÇÃO E CONTABILIDADE
DEPARTAMENTO DE ECONOMIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ECONOMIA

APRENDIZADO EM MACROECONOMIA DSGE

LEARNING IN DSGE MACROECONOMICS

Igor Velecico

Orientador: Pedro Garcia Duarte

SÃO PAULO

2013

IGOR VELECICO

LEARNING IN DSGE MACROECONOMICS

Tese apresentada ao Departamento de Economia da Faculdade de Economia, Administração e Contabilidade da Universidade de São Paulo como requisito para a obtenção do título de Doutor em Ciências.

Área de concentração: Teoria Econômica

Orientador: Prof. Dr. Pedro Garcia Duarte

Versão corrigida

SÃO PAULO

2013

FICHA CATALOGRÁFICA

Elaborada pela Seção de Processamento Técnico do SBD/FEA/USP

Velecico, Igor
Learning in DSGE macroeconomics / Igor Velecico. -- São Paulo, 2013.
139 p.

Tese (Doutorado) – Universidade de São Paulo, 2013.
Orientador: Pedro Garcia Duarte.

1. Macroeconomia 2. DSGE 3. Expectativas racionais I. Universidade de São Paulo. Faculdade de Economia, Administração e Contabilidade. II. Título.

CDD – 339

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my advisor Prof. Pedro Garcia Duarte for the continuous support of my research, for his patience, motivation and knowledge. I would like to thank Prof. Márcio Nakane, Prof. Fábio Kanczuk and Prof. Mauro Rodrigues for their encouragement, comments and ideas.

My sincere thanks also goes to Prof. Francesco Bianchi for the helpful discussions and ideas, and for receiving me as a visiting scholar in Duke University in the first semester of 2012, sponsored by CAPES - Foundation Coordination for the Improvement of Higher Education Personnel, of the Ministry of Education, Brazil.

I also want to thank my friends from University of São Paulo, for the stimulating discussions, and the University's staff, who helped me a lot during these years.

Last but not the least; I would like to thank my family: Boris, Olga, Alexandre, Marina and Lívia.

RESUMO

Nesta tese analisamos os instrumentos de aprendizado (Learning) aplicados a uma variedade de modelos macroeconômicos. Em nosso primeiro capítulo, apresentamos e discutimos as vantagens e limitações de se estimar modelos dinâmicos e estocásticos de equilíbrio geral (DSGE) acrescidos de um mecanismo de aprendizado, ou seja, abandonando-se a hipótese de expectativas racionais, tão cara a estes modelos. Em primeiro lugar, mostramos como esse mecanismo pode ser introduzido nesses modelos, começando pela discussão de onde e como o operador de expectativas racionais é substituído pelo operador de aprendizado. Em seguida apresentamos configurações alternativas em relação ao conjunto de informações disponível aos agentes dentro do mecanismo de aprendizado, que afeta diretamente a dinâmica do modelo final a ser estimado. Por fim, estimamos três modelos usando nosso mecanismo de aprendizado, aplicando-o a dados artificiais e reais para a economia brasileira.

No segundo capítulo, mostramos algebricamente as limitações do mecanismo de aprendizado em modelos DSGE e propomos dois métodos mais flexíveis para lidar com a instabilidade dos parâmetros nos dados. O primeiro desses métodos é intimamente ligado à literatura de DSGE-VAR, e que chamamos de Learning DSGE-VAR, enquanto o segundo método, que se afasta ainda mais do modelo DSGE, ao qual chamamos de LMSV.

No terceiro capítulo, provemos evidências de que os ganhos supostamente moderados de nosso modelo de aprendizado apresentados nos dois primeiros capítulos têm mais a ver com a natureza dos modelos estimados do que com o método de aprendizado utilizado. Para tal, simulamos dois grupos de dados usando uma estrutura econômica que varia no tempo, semelhante àquela estudada no primeiro capítulo, e estimamos os modelos utilizando diferentes mecanismos de aprendizado. Por fim, fornecemos evidências empíricas de aprendizado em modelos de forma reduzida para projetar inflação, taxas de juros e hiato do produto para a economia brasileira, através de modelos *ad-hoc* comumente utilizado por econometristas.

ABSTRACT

In this thesis we analyze learning mechanisms applied to a variety of macroeconomic models. In the first chapter, we present and discuss the advantages and limitations of estimating Dynamic Stochastic General Equilibrium (DSGE) models added with learning, thus suppressing the central assumption of rational expectations. First, we introduce the reader to the issue of how learning can be inserted in those models, starting from the discussion of where and how the rational expectations operator is substituted by the learning mechanism. We then present several additional learning setups related to the information set available to agents considered by the literature, which affect directly the dynamics of the final model. Last, we estimate three different models to assess the advantages of learning in our artificially generated data and real data for Brazil.

In the second chapter, we algebraically show the limitations of learning and propose two flexible methods to deal with the parameter instability in data. The first of these methods is closely related to the DSGE-VAR methodology, which we call Learning DSGE-VAR, and the second, which departs even further from the DSGE model, which we call Learning Minimum State Variable, or LMSV.

Finally, in the third chapter we provide evidence that the supposedly moderate improvements found in the previous chapters have more to do with the nature of the model at hand than to the learning method itself. To do so, we simulate problems using a time-varying structure similar to the one presented in chapter 1 and evaluate the likelihood improvements with different learning mechanisms. We then provide empirical evidence on learning in reduced form models to forecast inflation, interest rates and output gap for the Brazilian economy, using ad-hoc reduced form models commonly used by practitioners.

Contents

List of Tables

List of Figures

Introduction

Expectations play a central role in modern macroeconomic models. The current benchmark, given by the rational expectations hypothesis, assumes that agents know the exact structure of the economy, its structural parameters, the distribution of the shocks and the "fact" that other agents' expectations are also rational. This can be seen as an extreme assumption, specially from the point of view of an economist who is building a model: economists know that models are tools that help in the process of understanding a more complex reality.

Motivated by having a less restrictive hypothesis about expectations formation, economists have searched and tested alternative approaches to relaxing rational expectations. Different economists did so for different objectives. Woodford (2010), for instance, assesses how optimal monetary policy changes in a forward-looking model if the central bank assumes that private sector expectations are not exactly model-consistent (something which is tantamount to rational expectations), but close to, which is called near-rational expectations. The policymaker needs to choose a robustly optimal policy, given the uncertainty surrounding private-sector expectations, in the sense that this optimal policy does not result in an outcome that is too bad (similar to Hansen & Sargent (2008) robustness methodology, cited below). Under linear optimal policies, the main conclusion is that *"just as in the rational expectations analysis, commitment is important for optimal policy. The distortions predicted to result from discretionary policymaking become even more severe when the central bank allows for the possibility of near-rational expectations, so that the importance of commitment is increased"*.

Hansen and Sargent's (2008) book "Robustness" develops a methodology to assess robustness of a policy: for a proposed policy rule, the planner finds the worst outcome (using a specific loss function) that each possible misspecified model under a set of models can produce (hence, the planner has in mind a set of models, but he is uncertain of whether one of those models is correct). The planner then wants to find a policy that minimizes the maximum expected loss (worst outcome), using a min-max strategy. As Söderlind (2003) argues, *"the aim of robust control is to design a policy that will work reasonably well even if the approximating model does not coincide with the true model, as opposed to* *a policy that is optimal if they do coincide but possibly disastrous if they don't".*

On the macro/finance front, Brunnermeier & Paker (2005) develop a framework called "optimal expectations" in which agents care about current utility flow and expected future utility flows. More optimistic agents derive more utility from being optimistic (overestimating the true probability of positive outcomes) and less utility from distorted decisions (for the same reason). It is shown that for a consumption-saving problem with stochastic income and time-separable, quadratic utility function, the gains related to this "anticipated" utility are of first-order, while the losses related to the distorted allocations are of second-order. Thus the optimal expectations, in the sense that they maximize utility (which is now given by current utility flow and expected future utility flows), are the ones in which beliefs tend towards a little optimism, producing a higher utility in comparison to the rational expectations equilibrium.

Another departure from rational expectations is Sims' theory of "rational inattention" (1998, 2003), in which agents have a limited capability of processing all the (vast) information available (or there is a cost related to it) which makes attention scarce, so that agents have to deal with the quantitative problem of allocating it optimally. To quantify information flow, Sims uses the concept of entropy, a simple scalar measure of uncertainty.[1] Agents optimise the allocation of attention comparing its benefits (the reduction of uncertainty) to the cost of allocating attention to a specific variable. One of Sims' motivations is that existing models imply that (i) most kind of disturbances to the economy should result in instant movements in prices (which are denominated "classical" models) or (ii) most kind of disturbances to the economy should produce instant movements in output (which are denominated "Keynesian" models). As Sims argues, *"none of them, unfortunately, meets the criterion of being based on explicit recognition of the importance of information-processing"*, and *"the best of these models ought to be regarded as proxies for better, harder-to-construct models that would recognize the implications of limited information-processing capacity".*

[1]For instance, the entropy of a N-multivariate normally distributed random vector Y is given by:

$$H(Y) = \frac{1}{2}\log_2\left[(2\pi e)^N \det \Omega_Y\right]$$

where Ω_Y is the covariance matrix of Y. The entropy is then proportional to the determinant of the covariance matrix.

Another form of relaxing rational expectations is the learning approach of Evans & Honkapohja (2001). It assumes that agents act like econometricians, using a limited information set to forecast the variables of interest. These forecasts affect the dynamics of the economic model, which is then given by time-varying parameters. The objectives of using learning models can be theoretical, as in Rantala (2003), in which learning is used as an equilibrium selection mechanism (between one with low-inflation and one with high-inflation), and Mitra & Evans (2013), which are interested in assessing the analytical E-stability conditions for the stochastic Ramsey model under two different types of learning horizons; or objectives can be empirical, as in Milani (2006), Vilagi (2008), Slobodyan and Wouters (2009, 2012) and Gaus (2013). Empirical objectives often differ: for instance, Milani's (2006) main goal is to assess if learning is a possible substitute for other frictions in Dynamic Stochastic General Equilibrium (DSGE) models, while Slobodyan and Wouters (2009) are more interested in the improvement of fit and change in dynamics of a medium-scale DSGE learning model. A common feature to both objectives, however, as Evans and Honkapohja (2001, p. 12) argue, is the recognition that:

> *"In empirical work economists, who postulate rational expectations, do not themselves know the parameter values and must estimate them econometrically. It appears more natural to assume that the agents in the economy face the same limitations on knowledge about the economy. This suggests that a more plausible view of rationality is that the agents act like statisticians or econometricians when doing the forecasting about the future state of the economy. This insight is the starting point of the adaptive learning approach to modeling expectations formation"*

The list of alternative approaches to rational expectations goes on, with different researchers aiming different objectives, but all not fully satisfied with the assumption of rational expectations or trying to assess to what extent their results are valid if rational expectations is relaxed. For instance, Woodford (2013) reviews some alternative approaches to rational expectations (learning included) and how they deal with issues such as Ricardian equivalence, determinacy of equilibrium and the trade-off between inflation and output gap stabilization, among others, which indicates that these alternative approaches have been receiving increasing attention in the past few years.

In this thesis, we will follow the learning approach of Evans & Honkapohja (2001). We have several objectives: our first goal is to help organizing the discussions of these alternative learning setups in DSGE models, which is fragmentally present in the literature. Our second goal is to assess how learning models are able to improve the fit of DSGE models, relative to their rational expectations counterparts. We do so by estimating a larger set of learning specifications in comparison to what has been done in the empirical literature so far, which focused on a very limited set of specifications. As we will show, the set of specifications is quite large, and we will help filling that gap, which is our third goal. We estimate those learning specifications to three different DSGE models: a univariate model, from which we can derive analytical conclusions easily; a basic New Keynesian model, which is the standard textbook model taught in graduate programs; and a medium-size model, inspired in the work of Smets & Wouters (2003), which have become the benchmark DSGE model in the literature, with several frictions introduced to produce a better fit to real data.

Our fourth goal is to assess how the theoretical structure of the model changes our results. While in chapter 1 we deal with DSGE models, in chapter 2 we gradually fexiblize its structure, first estimating a (Learning) DSGE-VAR and then a (Learning) Minimum State Variable model. In chapter 3, we finally break the link with DSGE models and estimate reduced-form models. Hence, from the point of view of the theoretical structure of the models, chapters one to three compose a "gradient", from the most structural models of chapter 1 to the reduced-form models of chapter 3. This is done since economists and praticitioners often use a wide range of models in empirical work. Last, our fifth goal is the construction of a set of MATLAB routines to perform the estimation of this larger set of learning DSGE models, so that their performance can be compared easierly.

Our main results show that learning is a powerful tool for improving the fit of dynamic models. Since it is able to encompass rational expectations models as special cases, all three chapters presented improvements in the fit of some models, when learning replaced rational expectations. This lends credibility to arguments that favor learning as an empirical modeling strategy. We also show that the theoretical structure of DSGE models impose some restrictions to the law of motion of the economy, therefore limiting the improvement of fit. In chapter 3, we show some evidences that learning can improve the fit of reduced-form models by a larger extent (than DSGE models), and that the kind of information produced by the learning mechanism is so useful that pratictioners would be better off if they estimate learning models before proceeding to rational expectations.

This thesis is organized as follows: chapter 1 will mainly focus on the discussion of Learning DSGE models. In the second chapter, we provide an algebraic understanding of the limitations of learning in DSGE models and propose the two flexible methods mentioned earlier to deal with the parameter instability in data. Finally, in the third chapter we provide evidence that the supposedly moderate improvements found in the previous chapters have more to do with the nature of the model at hand than to the learning method itself, reinforcing learning as a relevant empirical mechanism to improve the fit of such models. In this chapter we also estimate some reduced-form models for Brazil, showing evidence that the improvement of fit can be much larger in this type of model (in comparison to the previous models linked to the DSGE). Chapter 4 concludes.

1 The learning approach

1.1 Introduction and motivation

The main goal of this first chapter is to present and discuss the advantages and limitations of estimating DSGE models added with a learning mechanism. First, we introduce the reader on how learning can be inserted in modern macroeconomic models, starting from the discussion of where and how the rational expectations operator is substituted by the learning mechanism.

Next, we present several additional learning setups considered by the literature, each defined by the information set available to agents. The learning mechanism impacts directly the dynamics of the model, which is then estimated. The learning structure is represented by a single extra parameter, called *gain*. The estimation of positive gains means that the learning mechanism is able to increase the fit (relative to the rational expectations specification) of the final model. We briefly introduce our MATLAB program used to run simulations and to estimate alternative models.

Last, we estimate three different models to assess the advantages of learning: an univariate model, using only simulated data; a three-equation basic New Keynesian model, using both simulated and Brazilian real data; and a simpler version of a medium-scale benchmark DSGE model, with 12 endogenous variables and 9 observables, also using Brazilian real data. The strategy here is first to test our learning mechanism in a very simple model and gradually increase its complexity toward a benchmark DSGE model widely used in the literature. The results show that for all models, there is at least one learning scheme that improves the fit of the model, but only to a certain extent.

Our first chapter is organized as follows: section 1 gives the reader an introduction to the learning methodology applied to DSGE models. In section 1.1 we explain how Learning DSGE Models (henceforth LDSGE) can add flexibility to improve some empirical results comparatively to standard, rational expectation DSGE models; in sections 1.2 and 1.3 we give an overview of the related literature and where our research stands. Section 2 then dives into some of the possible specifications of a LDSGE, and the differences among them. Section 3 estimates two models using the learning methodology and clarifies how well can LDSGE adapt to problems in simulated data. Section 4 estimates the complex LDSGE model using Brazilian real data, and section 5 concludes.

1.1.1 DSGE models and learning

DSGE models have become very popular since the the work of Christiano, Eichenbaum & Evans (2005) and Smets and Wouters (2003, 2007), specially in the literature of policymaking and among central banks. By adding a rich set of mechanical model frictions such as Calvo sticky prices and wages, adjustment costs to investment, habit formation, among others, and using Bayesian econometrics, these models have become a powerful tool for both policymakers and practitioners. While the former were mainly interested in the evaluation of alternative scenarios, possible only under structural, policy-invariant parameters,[2] the latter's concerns were more related on how well could these highly parametrized models match real data.[3]

Once Smets and Wouters (2007) (henceforth SW-07) proposed a model that was able to both match the data well while estimating structural parameters, which had economic interpretations,[4] research on DSGE models experienced a huge boost. New models and frictions were created and tested, giving birth to a vast literature of models telling different possible stories, which with the help of Bayesian econometrics could be compared on equal ground.[5] Central Banks economic teams also started to research and create their own DSGE models: the European Central Bank's NAWM (New Area-Wide Model), the Brazilian's SAMBA (Stochastic Analytical Model with a Bayesian Approach) and the US' SIGMA are just but a few examples on how fast these models were able to reach policymakers.

In fact, this transition was surprisingly fast, since it usually takes much more than a few years to transplant certain methods or models from academic researchers to Cen-

[2] See Lucas (1976) critique.

[3] Obviously, policymakers are also concerned on how reliable are forecasts produced by the model, but often not as much as practitioners.

[4] See Fernandez-Villaverde & Rubio-Ramirez (2007) critique on how structural are these structural parameters.

[5] In opposition to classical econometrics, which is based on a comparison of a "true" model (null hypothesis) against an alternative one. Bayesian econometrics treats all models as equally false, and the data helps on choosing which one is best. See Sims (2008) for more details on how research on DSGE models is and should be pursued.

tral Banks.[6] One of the reasons that might have helped is the fact that, even though economists often disagree about which shocks are really structural and which ones are consistent with microeconomic evidence, as in Chari, Kehoe & McGrattan (2009), there was a convergence in terms of methodology towards dynamic stochastic general equilibrium models.

Altough SW-07 is seen as a benchmark medium-size model, research is far from done. As Sims (2008) argues, "*the Smets and Wouters example ought to lead not to a convergence toward one type of model, but to a proliferation of models that fit well, allowing us to compare theoretical approaches with a new level of discipline from the data*".

Learning can be seen as one of these models, one alternative story to explain economic data. The goals for using this approach, however, are not always the same. For simplicity, we divide these goals below into theoretical and source of inertia and fit.

Theoretical The assumption of Rational Expectations involves the assumption that agents know (i) the exact structure of the economy, (ii) its structural parameters, (iii) the distribution of the shocks and (iv) that other agents are also rational. From the point of view of an economist who is building a model, this can be seen as an extreme assumption. Economists know that models are tools that help in the process of understanding a more complex reality. We know that all models are false,[7] and therefore *we know that we don't know* (i) the exact structure of the economy, (ii) its structural parameters or (iii) the distribution of the shocks. Hence, it seems implausibly strong that agents in our economic model should know more than the economic modeler himself. In this sense, learning can provide a more plausible way to model rationality: agents act like econometricians, using a limited information set.

One can then ask whether rational expectations is plausible, or whether it is a strong assumption that should not be taken for granted without testing. One can also ask if a particular rational expectations equilibrium (REE) is learneable by agents, a principle called E-Stability, which is closely related to the stability conditions of the REE itself. This principle was developed by Evans & Honkapohja (2001) and can be also found in

[6] An interested reader will find more detail in Sims (2008). For more details on the usage of DSGE models by Central Banks, see Tovar (2009).

[7] Lucas (1980, pgs 697,709-10)

Mitra & Evans (2013) as a way of producing evidence *in favor of* rational expectations (RE), since in their model all learning paths leads to the (unique) REE, i.e. it is learneable by agents.

In fact, this early literature of adaptive learning is mainly theoretical and focused on the convergence of models to the RE equilibrium. We recommend Evans and Honkapohja (2001) for a review of this debate. But there are other theoretical considerations that can also be addressed. For example, learning can be used as a selection device when there are multiple equilibria (low and high inflation) in a macro model, as in Rantala (2003).

Source of inertia and fit Good economic policy relies strongly on at least two things: knowing the limitations of the model, and trusting, with the appropriate and usual caution, in its forecasts. Until the work of Christiano, Eichenbaum & Evans (2005), monetary models were not able to generate the inertia observed in real data, in terms of dynamic responses of inflation and output to shocks, as shown in Gali (2008, Ch. 1).

Christiano, Eichenbaum & Evans's work, combined with SW-07, changed the focus of the literature to studying and comparing various types of stories, each containing different frictions, that were able to match this inertia. However, as models became larger, the complexity to understand them increased too. DSGE models now range from the simple 3 equation basic New Keynesian (NK) model to large models containing dozens of equations. Even medium-scale models like SW-07 require from economics analysts a strong background in mathematics to be fully understood. This higher complexity is there to (i) provide better counterfactuals,[8] given estimated structural parameters, and (ii) to produce enough frictions to match the inertia present in data.

Learning, in this case, can be viewed as a tool of relaxing the rigid time-invariant state-space into a time-varying one, which has the capability to match the data better than RE models. Slobodyan and Wouters (2012) showed that learning can increase the fit of SW-07 model, while still estimating structural parameters robust to relaxing the rational expectations assumption.

[8] In order to produce trustworthy counterfactuals, parameters must be policy invariant and forecasts acceptable. This requires the construction of a more complex set of problems faced by agents and firms (for example, Calvo pricing), which must be now solved before estimating the model.

This relaxation mechanism, however, should not be seen as one trying to insert time-varying dynamics or just more lags into the model to generate fit. In our view, the insertion of learning is somewhere between time varying parameters and Christiano, Eichenbaum & Evans (2005)-type frictions, but closer to the latter. Expectations are formed by updating reduced-form regressions, but they affect model dynamics only through the expectations operator, and not necessarily by simply adding lags or time-varying parameters to all equations. This impact on model dynamics is closer to Calvo pricing than to time-varying parameters. As will be clearer in chapter 2, learning approximates the DSGE model to a time-varying parameters framework only to a certain extent.

There are at least two possible interesting questions related to learning: one, which can be answered, is how robust are DSGE frictions in a framework with more flexible dynamics. Milani (2006) estimated a 3 equation DSGE model with learning and found out that learning was able to generate enough intertia to diminish the importance of frictions such as habit formation and inflation indexation. The second, which cannot be answered by learning, is when these frictions are not robust to its presence, what is the best way to proceed: to insist on using those frictions in a rational expectations model, to migrate to a model with learning without those frictions, or to find new frictions under rational expectations that are robust (under RE)?[9]

Also important is the following question: if learning is a possible substitute source of inertia it could be used to allow simpler models to fit the data equally well as bigger[10] ones, without the cost of focusing on too many frictions/equations. This is important specially for practitioners, who usually want their models to be as simple as possible, with reliable forecasts and policy-invariant parameters.

This thesis was originally motivated by both reasons (theoretical and inertia/fit), but we will discuss here only the latter. The main reasons are that (i) the theoretical discussion of learning is already well understood[11] and there is little, if any, possible theoretically appealing contribution left, (ii) the empirical discussion of learning models is still in its

[9]The idea here is that a model inserted with learning can diminish the importance of other frictions, but it does not necessarily perform better than all other frictions. Hence, it is up to the user to decide how to proceed: to estimate the model with learning or to search for alternative models that are robust to it.

[10]Here, we refer to models with higher mathematical complexity and/or more parameters, designed to explain the same data as the simpler models mentioned before.

[11]As seen in Evans and Honkapohja (2001).

initial steps, with much to be explored. To illustrate our point, we briefly discuss why learning can adapt to real data in a more flexible way the usual rational expectations models.

Any DSGE model can be expressed as:

$$F(y_{t-1}, y_t, y_{t+1}, w_t, \varepsilon_t) = 0 \tag{1.1}$$

where y_t is a vector of endogenous variables, w_t is a vector of exogenous variables, and ε_t is a vector of innovations. F is a function that relates the variables at hand, which is (often) given by the first order conditions of the DSGE model. After a first order approximation,[12] the linearized DSGE model can be cast as:

$$A_0 y_{t-1} + A_1 y_t + A_2 E_t y_{t+1} + A_3 w_t + A_4 \varepsilon_t = 0 \tag{1.2}$$

The matrices A_0, A_1, A_2, A_3 and A_4 are functions of structural parameters collected in a vector θ, but we ommit this dependence for cleariness. E_t is the rational expectations operator, which tells us that expectations are model consistent.[13] We can then, for a given θ, solve this system[14] and cast it in a state-space format, which can be expressed as:

$$\alpha_t + \mu + T\alpha_{t-1} + R\xi_t = 0 \tag{1.3}$$

$$Y_t^{obs} - H\alpha_t = 0 \tag{1.4}$$

where α_t collects state variables, ξ_t collects innovations, and matrices μ, T, R are functions of θ. The matrix H is a selection matrix which links state variables to the variables observed by the economist. We then use Bayesian econometrics and numerical algorithms to find a vector $\hat{\theta}$ that maximizes the fit of this state-space to the data.

Learning, in contrast to rational expectations, relies on the understanding that agents have limited information and therefore use some reduced-form model to guess the evolu-

[12] We will work with first order approximations of DSGE models. For a discussion of higher order approximations, see Woodford (2003).

[13] By model consistent we mean that expectations are given by the solution implied by the model at hand.

[14] If it can be solved in the first place, obviously. We will discuss this topic later.

tion of the variables they have access to. This became known as *econometric* learning or *adaptive* learning, and will be the approach we will discuss in this thesis. Suppose the economic model is given by:[15]

$$A_0 y_{t-1} + A_1 y_t + A_2 \widehat{E}_t y_{t+1} + A_3 w_t + A_4 \varepsilon_{1,t} = 0 \tag{1.5}$$

where $\widehat{E}_t$ denotes a non-rational expectations operator related to the underlying reduced-form model that agents use. For now, we suppose this reduced-form model is given by:

$$\widehat{E}_t y_{t+1} = a_t + b_t y_t + c_t w_t \tag{1.6}$$

where a_t, b_t and c_t are reduced-form parameters that somehow evolve over time. By substituting the latter equation into (1.5), we get:

$$A_0 y_{t-1} + A_1 y_t + A_2 (a_t + b_t y_t + c_t w_t) + A_3 w_t + A_4 \varepsilon_{1,t} = 0 \tag{1.7}$$

By defining some auxiliary matrices,[16] we can write the model as a state-space system given by:

$$\alpha_t + \mu_t + T_t \alpha_{t-1} + R_t \xi_t \;=\; 0 \tag{1.8}$$

$$Y_t^{obs} - H\alpha_t \;=\; 0 \tag{1.9}$$

which is similar to the RE's, but now matrices μ_t, T_t and R_t are all indexed to time (since they are functions of a_t, b_t and c_t), and end up producing a time-varying state-space system. This adds flexibility to learning models to dominate rational expectations models in terms of fit, and it is one of the reasons why learning is an attractive empirical tool for modelers.

In fact, this new time-varying state-space has two different channels through which it improves the fit of the model to the data: one is related to the addition of a reduced-form model (with no time varying parameters), which can account for misspecifications inside the original DSGE model[17] $-$ if the Taylor rule equation somehow does not have inflation

[15] The next sections will distinguish and clarify between different ways of (i) introducing learning in a DSGE model and (ii) choosing a specific reduced form model.

[16] See section 2.

[17] The typical mispecification problem here is the absence of a relevant variable from an equation.

on the right-hand side, for example. Since the reduced-form equations allow for richer dynamics (bringing the DSGE model somehow closer to an unrestricted VAR), parameter estimation becomes less cumbersome, hence improving fit.

The second channel is related to the time-variation of the parameters of this reduced-form model, which can account for mispecifications but also structural breaks, and time-varying parameters in data. Our research is not interested in answering which of these channels is more important to improve fit, leaving the answer for future research.

Now that we understood the main differences between rational expectations and learning models, we can take a step back and start from the beginning: in the next two sections we give an overview on how learning can be inserted in a DSGE model, and where our research is located.

1.1.2 Literature and our research

The learning literature applied to DSGE models is still on its initial stages. To help illustrate where our research is located, table 1.1 shows some of the alternative ways of building a learning DSGE model.[18] This involves several choices. First, one has to choose between learning horizons (that will be explained in the next section) and the order of approximation of the usually non-linear DSGE model. So far, most of the literature has used first order approximations and Infinite Horizon (IH) or Euler Equation (EE) learning.

Then one has to select agents' reduced-form model, which is called the *Perceived Law of Motion* (PLM) and can be organized in four categories: Minimal State Variables with or without constants (MSVc and MSV, respectively) and Vector Autoregressive, again with or without constants (VARc and VAR). The timing of this perceived law of motion can also be changed, allowing agents to observe or not $time - t$ exogenous variables at time t.

Last, one needs to choose an algorithm to update agents' beliefs (constant gain, stochastic gradient, Kalman filter and endogenous gain are the most common choices), and how the initial beliefs are formed (fixed at some vector, consistent to the rational expectations equilibrium, or chosen optimally).

Table 1.1: Some of the setups for a Learning model

Horizon	**Approx**	**PLM**	**Timing**	**Algorithm**	**Initial beliefs**
IH	1^{st} order	MSVc	t	Constant gain	Fixed
EE	2^{nd} order	MSV	t-1	Stochastic gradient	RE consistent
FH-EE	Above 2^{nd}	VARc		Kalman Filter	Optimal
FH-IH		VAR		Endogenous gain	

Unitl now there are no papers applying Finite Horizon (FH) learning to real data. On the IH front, Milani (2007) estimates Preston's model expanded with habit formation and price indexation as the main sources of inertia (a 3 equation DSGE model for output, inflation and interest rates). There is a learning mechanism in which agents form expectations using a VAR model. As in Preston, learning is introduced into model's primitives, generating an infinite horizon framework. One of the findings is that the learning mechanism make the other mechanical frictions useless; learning is a possible source of inertia, in substitution of mechanical ones.

Eusepi and Preston (2010) use an IH learning model with similar size than Milani's, but with a slightly different information set: agents observe only variables contained on their own maximization problem (their own consumption, constraints and exogenous aggregate variables). All agents have the same model (a common hypothesis in this literature), so aggregation is straightforward. Their goal is not to see if structural parameters are robust to learning, but to understand how its insertion changes model dynamics. They calibrate the parameters to match some empirical facts, and choose a small gain (we will enter in more detail later). Their main conclusion is that learning is able to amplify the propagation mechanism of disturbances relative to the rational expectations economy, also improving the characterization of the second moments of the data.

On the Euler Equation front, Vilagi (2008) estimates some learning DSGE models for Hungary (three models, two of them with 5 equations and one with 3 equations), and finds out that the models with adaptive learning perform better than the ones under rational expectations. However, learning does not act as a substitute for mechanical sources of inertia, even though the persistence associated with them became lower.

[18] We will cover all these paths in detail in the next section.

Slobodyan and Wouters (2009) (henceforth SLW-09) assess if Milani's (2007) result is robust in a more realistic model, where more frictions (and a bigger model) allow for a better fit of the data. They work with Euler Equation learning, so agents only need to forecast variables one period ahead, and not their path in the entire infinite future as in Milani's model. This inhibits a clean comparison of the model-size importance between SLW-09 and Milani, since the learning horizons are different and therefore the final model dynamics themselves are different. Their main findings are: (i) a medium scale DSGE model learning is useful but it does not make other frictions useless; (ii) both initial beliefs and learning play an important role to improve the fit in the model.

As we can see there are many options and thus a lot of choices when one wants to introduce learning in a DSGE model. Most of the literature used only some variations of the estimation procedure. If we consider a first order Euler Equation model (such as we do throughout our chapters), we can build at least 96 different models (the combination of 4 Perceived Law of Motion, 2 Perceived Law of Motion timings, 3 types of initial beliefs and 4 types of algorithms). In fact, if we deal with bigger models, this number can be much larger, since we can use different subsets of observables variables in the Perceived Law of Motion (under VAR learning).

Of course, our gol is not to estimate all possible methods. The restrictions we impose are: (i) first order approximations, (ii) Euler Equation learning, (iii) don't use endogenous gain or Kalman Filter learning, only constant gain or stochastic gradient, and (iv) rational expectations consistent initial beliefs. We end up with 16 possible models to use to characterize our data.[19]

Most of the literature focused on applying some narrow subset of these estimation procedures, and very often only to real or simulated data, not both. Our goal here is to compare how these different dynamics can deal with a set of different problems in a simulated environment and in real data. To do that, we simulate data with time-variation in structural parameters, giving, in theory, space for time-varying models to outperform RE. We then go to real data to understand the fit improvement given the learning mechanisms.

[19]Our MATLAB code can estimate 48 of all 96 models described above. In addition to what we estimate here, our code can account for (i) fixed initial beliefs and (ii) optimised beliefs.

1.2 Learning in DSGE models

In this section, we will enter into more detail about the possibilities of inserting learning in a DSGE model. As it was briefly showed in table 1.1, there are 5 decisions that must be made after choosing to work with a (linear) DSGE model, which we group into three categories: (i) the learning horizon, which is how learning enters the DSGE system of equations and affects its dynamics; (ii) agents' information set, which is given by the Perceived Law of Motion and the timing associated with its variables; (iii) initial beliefs and a learning algorithm, which gives the law of motion of the reduced-form parameters of the Perceived Law of Motion. Thus this section is divided into three subsections.

1.2.1 The main setup: learning horizon

The first of many decisions that one needs to make when inserting learning in a DSGE model is when the rational expectations operator is supposed to be substituted by the non-rational one. Here we briefly review three ways that the literature has dealt with this issue and the discussions in favor of each approach.

The first way of introducing learning in an infinite horizon macro model is due to Honkapohja, Mitra and Evans (2002). They depart from the linearized Euler equation for consumption of the Ramsey-Cass-Koopmans' model under rational expectations, given by:

$$c_t = E_t c_{t+1} + \varphi E_t r_{t+1} \tag{1.10}$$

where consumption at time t depends on the expected future consumption and expected real interest rates. Here E_t is the rational expectations operator and φ combines some structural parameters of the economy. What they do is to simply substitute E_t by $\hat{E}_t$, where the latter represents a non-rational expectations operator. Equation (1.10) then becomes:

$$c_t = \hat{E}_t c_{t+1} + \varphi \hat{E}_t r_{t+1} \tag{1.11}$$

As this method consists on substituting the rational expectations operator for a non rational expectations operator directly into the linearized system of equations, this approach is called Euler-Equation learning.

This method was criticized by Preston (2005), who argues that if learning is introduced directly on the primitives of the model, i.e. at agents' level, the optimal decision rules would require that agents build infinite-horizon expectations for the aggregate variables. Consumption's optimal choice, in that case, would be given by:

$$c_t^i = E_t^i c_{t+1}^i + \varphi r_t^i \tag{1.12}$$

Combining this optimal choice with the problem's intertemporal budget constraint we get:

$$\hat{E}_t^i \sum_{s=0}^{\infty} \beta^{t+s} c_{t+s}^i = w_t^i + \hat{E}_t^i \sum_{s=0}^{\infty} \beta^{t+s} y_{t+s}, \tag{1.13}$$

Thus, it can be shown that agent's i consumption in this case is given by:

$$c_t^i = (1-\beta) w_t^i + \hat{E}_t^i \sum_{s=0}^{\infty} \beta^{t+s} \left[(1-\beta) y_{t+s} - \varphi r_{t+s} \right] \tag{1.14}$$

which is now function of some initial wealth w_t^i and *the path of* expected output (y_t) and interest rates for an infinite horizon.

The main problem, identified by Preston, is that the EE approach *"transforms the decision problem from one in which households forecast state variables beyond their control to one in which they forecast their own future consumption decisions using an arbitrary statistical rule and as a results form expectations with respect to a probability distribution that is not implied by the decision problem"*. As a consequence, Preston argues that EE learning (i) fails to produce optimal alocations of consumption over time; (ii) does not respect intertemporal budget constraint and (iii) is inconsistent with the model micro-foundations. Honkapohja & Mitra (2011) later answered to that critique, showing that "*the intertemporal accounting consistency holds in an ex post sense along the sequence of temporary equilibria under Euler equation learning*", and that "*the usual system based on Euler equations with subjective expectations can be obtained from Preston's approach and is, therefore, a valid way of analyzing learning dynamics under incomplete knowledge*".

In paralel to this debate, Brach, Evans & McGough (2010) introduced the Finite Horizon learning (FH), in which two new approaches are developed: (i) the N-step EE learning and (ii) the *optimal* N-step EE learning. The first one consists of the simple iteration of the Euler Equation N steps ahead,[20] resulting in the following decision rule:

$$c_t = E_t c_{t+N} + \varphi E_t \sum_{s=0}^{N} r_{t+1} \tag{1.15}$$

where consumption on t depends on the expectations of medium-term consumption (c_{t+N}) and the interest rate *path* from t to $t+N$.

In their second approach (*optimal* N-step learning), the budget constraint is first iterated N periods ahead, and then combined with agent i Euler equation, also iterated N periods ahead, resulting in:

$$c_t^i = \phi_1(N) s_{t-1}^i + \phi_2(N) s_{t+N}^i + \sum_{n=1}^{N} \phi_3(N,n) \hat{E}_t^i r_{t+n} + \phi_4(N) \sum_{n=1}^{N} \beta^n \hat{E}_t^i w_{t+n} \tag{1.16}$$

where the functions ϕ_j collects structural parameters as φ did in our EE example, and s_t denotes savings at period t. As emphasized by the authors, this consumption rule will converge to IH learning as N goes to infinity. For this reason, this latter rule is viewed as the FH counterpart to IH learning, while the first is viewed as the finite horizon counterpart to EE learning.

The choice of which learning approach to adopt is not trivial, and depends on what questions the researcher is addressing. In favor of FH learning, at least three arguments must be considered: first, it is not intuitive that agents should make infinite-horizon forecasts in a framework that assumes limited rationality; second, even if agents could make infinite-horizon forecasts, it is not necessarily true that this strategy will produce the best consumption allocation over time, since when agents are deciding how much to consume today, a large weight is given to long-horizon forecasts, even though agents know that they are learning and their decision rules can (and probably will) change[21] in the future; third, in fact most real agents work with a finite horizon, and when they are challenged to forecast variables in the long-run, this is done with great caution and uncertainty.

[20] This is the finite-horizon counterpart to the EE approach. The autors also show in their paper that this rule will not converge to IH solution when N goes to infinity.

[21] As the authors emphasize, *"this reasoning suggests that agents may do best with finite horizon models that look further ahead than one period, but do not attempt to forecast beyond some suitable finite horizon"*.

As Branch, Evans & McGough argue, *"On the one hand, Euler-equation learning has been offered as a simple behavioral rule providing a boundedly rational justification for examining adaptive learning within one-step ahead reduced-form systems. On the other hand, the principal alternative proposal has been to assume that agents solve their infinite-horizon dynamic optimization problem each period, using current estimates of the forecasting model to form expectations infinitely far into the future. In contrast, introspection and common sense suggests that boundedly rational decision-making is usually based on a finite horizon, the length of which depends on many factors"*

IH learning, on the other hand, deals with expectations in a more formal and rigorous way, since they are inserted in the primitives of the model. However, this leads to a consumption rule (and any other equations which involve expectations) which can become very complicated, specially in larger models, blurring economic intuition and making analysis a lot more difficult.

We use EE learning in this thesis. In our view there are good reasons to do so: in the first place, the main benefit of EE approach (in contrast to both IH and FH) is that model's expectations are kept in a simple and very tractable form, which does not require the iteration of budget constraints or any other equations of the original model. In this sense, EE learning can be promptly applied to any linearized system of equations with expectations, and the evolution of the reduced-form coefficients of agents' model can be understood in a straightforward way.[22] In the second place, because the literature on learning DSGE models is very recent, there are still some gaps to be filled in how well these models can adapt to different problems that could be present in real data. Thus we aim to fill some of these gaps in understanding how the most simple and tractable form of learning can adapt to these problems. As will be clearer in section 2, as we have already mentioned, even though it is the simplest way of inserting expectations in a macro model, there is a rich set of different settings on how one can control agents' information and therefore the final model dynamics.[23]

[22] This argument will be clearer when we present the settings that can be changed in our code.

[23] Another intuitive argument for EE against FH is that the higher volatility of model's expectations under FH (given by equation (1.15), which involves mid-term forecasts) could be translated into a lower estimated gain, which would bring EE and FH final dynamics close to each another. We do not test this argument here, leaving it for future research.

1.2.2 Agents' information set: Perceived Law of Motion

We will work with a general linearized DSGE model given by:

$$
\begin{aligned}
A_0 y_{t-1} + A_1 y_t + A_2 E_t y_{t+1} + A_3 w_t + A_4 \varepsilon_{1,t} &= 0 && (1.17)\\
w_t + B_0 w_{t-1} + B_1 \varepsilon_{2,t} &= 0, && (1.18)
\end{aligned}
$$

which is the notation used in our MATLAB code. As it is already known, the rational expectations solution to this DSGE model is given by:

$$
\alpha_t + \mu + T\alpha_{t-1} + R\xi_t = 0 \tag{1.19}
$$

where α_t is a vector of states, ξ_t are innovations, and matrices μ, T and R are functions of structural parameters θ, present in matrices A_0 to B_1. The demonstration is done in Appendix B.

We now turn to the problem of finding a solution for a model like this with learning. We will split our analysis into four information sets: the first one is the *time t information set* and includes y_t and w_t as explanatory variables in the *Perceived Law of Motion*. The second one is the *time t-1 information set*, which includes y_{t-1} and w_{t-1} instead. The third one is the mixed time information set, which includes y_{t-1} and w_t as explanatory variables. Last, we present VAR Learning, which includes only observable variables in the PLM.

Time t information set Our model is given by:

$$
\begin{aligned}
A_0 y_{t-1} + A_1 y_t + A_2 \widehat{E}_t y_{t+1} + A_3 w_t + A_4 \varepsilon_{1,t} &= 0 && (1.20)\\
w_t + B_0 w_{t-1} + B_1 \varepsilon_{2,t} &= 0, && (1.21)
\end{aligned}
$$

where expectations of the endogenous variables ($\widehat{E}_t$) are not rational, but given by the following reduced-form model, called *Perceived Law of Motion* (PLM):

$$
y_t = a_{t-1} + b_{t-1} y_{t-1} + c_{t-1} w_t, \tag{1.22}
$$

which states that agents form expectations of y_t using its own past values and present values of w_t. It follows from (1.22) that:

$$y_{t+1} = a_t + b_t y_t + c_t w_{t+1} \tag{1.23}$$

We now take (non-rational) expectations at time t of (1.23) and obtain:

$$\widehat{E}_t y_{t+1} = a_t + b_t y_t - c_t B_0 w_t \tag{1.24}$$

where we used (1.21) since expectations include only variables up to time t.[24] To avoid simultaneity between y_t and b_t, our coefficients a_t,b_t,c_t are obtained[25] by running an OLS regression of y_{t-1} on a constant, y_{t-2} and w_{t-1}. We then insert this PLM (1.24) into the economic model (1.20), we get the *Actual Law of Motion* (ALM):

$$\begin{aligned} A_0 y_{t-1} + A_1 y_t + A_2(a_t + b_t y_t - c_t B_0 w_t) + A_3 w_t + A_4 \varepsilon_{1,t} &= 0 && (1.25) \\ w_t + B_0 w_{t-1} + B_1 \varepsilon_{2,t} &= 0 && (1.26) \end{aligned}$$

We now define auxiliary matrices: $\widetilde{A} = (A_1 + A_2 b_t)^{-1}$, and $\widetilde{B} = (A_3 - c_t B_0)$. After rearranging terms and inserting (1.26) into (1.25), we get:

$$\begin{aligned} \widetilde{A}A_0 y_{t-1} + y_t + \widetilde{A}A_2 a_t - \widetilde{A}\widetilde{B}B_0 w_{t-1} - \widetilde{A}\widetilde{B}B_1 \varepsilon_{2,t} + \widetilde{A}A_4 \varepsilon_{1,t} &= 0 && (1.27) \\ w_t + B_0 w_{t-1} + B_1 \varepsilon_{2,t} &= 0. && (1.28) \end{aligned}$$

We want to write the system above as a state-space system. First, we write (1.27) and (1.28) as a vector:

$$\begin{vmatrix} y_t \\ w_t \end{vmatrix} + \begin{vmatrix} \widetilde{A}A_2 a_t \\ 0 \end{vmatrix} + \begin{vmatrix} \widetilde{A}A_0 & -\widetilde{A}\widetilde{B}B_0 \\ 0 & B_0 \end{vmatrix} \begin{vmatrix} y_{t-1} \\ w_{t-1} \end{vmatrix} + \begin{vmatrix} \widetilde{A}A_4 & -\widetilde{A}\widetilde{B}B_1 \\ 0 & B_1 \end{vmatrix} \begin{vmatrix} \varepsilon_{1,t} \\ \varepsilon_{2,t} \end{vmatrix} = 0, \tag{1.29}$$

[24]Note that by using (1.21) we allow agents to know the behavior of the exogenous variables.
[25]We will explain this topic later.

and, defining the auxiliary matrices

$$\begin{aligned}
\alpha_t &= \begin{vmatrix} y_t \\ w_t \end{vmatrix}, \mu_t = \begin{vmatrix} \widetilde{A} A_2 a_t \\ 0 \end{vmatrix}, \xi_t = \begin{vmatrix} \varepsilon_{1,t} \\ \varepsilon_{2,t} \end{vmatrix}, \\
T_t &= \begin{vmatrix} \widetilde{A} A_0 & -\widetilde{A}\widetilde{B} B_0 \\ 0 & B_0 \end{vmatrix}, R_t = \begin{vmatrix} \widetilde{A} A_4 & -\widetilde{A}\widetilde{B} B_1 \\ 0 & B_1 \end{vmatrix},
\end{aligned}$$

and an auxiliary selection matrix H (which contains only zeros and ones), we can write the following state-space system:[26]

$$\alpha_t + \mu_t + T_t \alpha_{t-1} + R_t \xi_t = 0 \tag{1.30}$$

$$Y_t^{obs} - H\alpha_t = 0 \tag{1.31}$$

Time t-1 information set Now we allow agents to use only time t-1 variables; y_{t-1} and w_{t-1}. The PLM is now given by:

$$\begin{aligned}
y_t &= a_{t-1} + b_{t-1} y_{t-1} + c_{t-1} w_{t-1} \\
y_{t+1} &= a_t + b_t y_t + c_t w_t && (1.32) \\
y_{t+1} &= a_t + b_t(a_{t-1} + b_{t-1} y_{t-1} + c_{t-1} w_{t-1}) - c_t B_0 w_{t-1} && (1.33) \\
\widehat{E}_t y_{t+1} &= a_t + b_t a_{t-1} + b_{t-1} b_t y_{t-1} + (b_t c_{t-1} - c_t B_0) w_{t-1}, && (1.34)
\end{aligned}$$

which is different from the *time t* Perceived Law of Motion (1.24). Thus, when we insert this equation into the economic model, we will get a new Actual Law of Motion:

$$A_0 y_{t-1} + A_1 y_t + A_2[a_t + b_t a_{t-1} + b_{t-1} b_t y_{t-1} + (b_t c_{t-1} - c_t B_0) w_{t-1}] + A_3 w_t + A_4 \varepsilon_{1,t} = 0 \tag{1.35}$$

$$w_t + B_0 w_{t-1} + B_1 \varepsilon_{2,t} = 0 \tag{1.36}$$

[26] The usual (but equivalent) notation of a state space system is:

$$\begin{aligned}
Y_t^{obs} &= H\alpha_t \\
\alpha_t &= \mu_t + T_t \alpha_{t-1} + R_t \xi_t
\end{aligned}$$

After rearranging terms and using again auxiliary matrices $\widetilde{A} = A^{-1}$, $\widetilde{B} = A_0 + A_2 b_{t-1} b_t$ and $\widetilde{C} = [A_2(b_t c_{t-1} - c_t B_0) + A_3]$, we get:

$$\begin{vmatrix} y_t \\ w_t \end{vmatrix} + \begin{vmatrix} \widetilde{A}A_2(a_t + b_t a_{t-1}) \\ 0 \end{vmatrix} + \begin{vmatrix} \widetilde{A}\widetilde{B} & -\widetilde{A}\widetilde{C} \\ 0 & B_0 \end{vmatrix} \begin{vmatrix} y_{t-1} \\ w_{t-1} \end{vmatrix} + \begin{vmatrix} \widetilde{A}A_4 & 0 \\ 0 & B_1 \end{vmatrix} \begin{vmatrix} \varepsilon_{1,t} \\ \varepsilon_{2,t} \end{vmatrix} = 0, \quad (1.37)$$

The system above is equivalent to (1.29) and is almost in state-space notation; we just need (again) to specify observables with a selection matrix H and redefine previous auxiliary matrices:

$$\begin{aligned} \alpha_t &= \begin{vmatrix} y_t \\ w_t \end{vmatrix}, \dot{\mu}_t = \begin{vmatrix} \widetilde{A}A_2(a_t + b_t a_{t-1}) \\ 0 \end{vmatrix}, \xi_t = \begin{vmatrix} \varepsilon_{1,t} \\ \varepsilon_{2,t} \end{vmatrix}, \\ \dot{T}_t &= \begin{vmatrix} \widetilde{A}\widetilde{B} & -\widetilde{A}\widetilde{C} \\ 0 & B_0 \end{vmatrix}, \dot{R}_t = \begin{vmatrix} \widetilde{A}A_4 & 0 \\ 0 & B_1 \end{vmatrix}. \end{aligned}$$

Once defined, we can write the state-space system as before:

$$\begin{aligned} \alpha_t + \dot{\mu}_t + \dot{T}_t \alpha_{t-1} + \dot{R}_t \xi_t &= 0 \quad (1.38) \\ Y_t^{obs} - H\alpha_t &= 0, \quad (1.39) \end{aligned}$$

Note, however, that this system is different from its *time t* counterpart. This difference does not come from the reduced-form coefficients, since a_t,b_t,c_t continue to come from a regression of y_{t-1}on y_{t-2} and w_{t-1} in both *time t* and *time t-1* information sets; the difference lies in the agents' Perceived Law of Motion, which implies a different Actual Law of Motion.

Mixed time information set This information set allow agents to observe *time t* exogenous variables, but only up to *time t-1* endogenous ones. The PLM is now given by:

$$
\begin{aligned}
y_t &= a_{t-1} + b_{t-1} y_{t-1} + c_{t-1} w_t \\
y_{t+1} &= a_t + b_t y_t + c_t w_{t+1} && (1.40) \\
y_{t+1} &= a_t + b_t (a_{t-1} + b_{t-1} y_{t-1} + c_{t-1} w_t) - c_t B_0 w_t && (1.41) \\
\widehat{E}_t y_{t+1} &= a_t + b_t a_{t-1} + b_{t-1} b_t y_{t-1} + (b_t c_{t-1} - c_t B_0) w_t, && (1.42)
\end{aligned}
$$

which is different from both *time t* and *time t-1* PLM's. Again, inserting this equation into the economic model, we get another ALM:

$$
A_0 y_{t-1} + A_1 y_t + A_2 [a_t + b_t a_{t-1} + b_{t-1} b_t y_{t-1} + (b_t c_{t-1} - c_t B_0) w_t] + A_3 w_t + A_4 \varepsilon_{1,t} = 0 \quad (1.43)
$$

$$
w_t + B_0 w_{t-1} + B_1 \varepsilon_{2,t} = 0 \quad (1.44)
$$

After rearranging terms and using again auxiliary matrices $\widetilde{A} = A_1^{-1}$, $\widetilde{B} = A_0 + A_2 b_{t-1} b_t$ and $\widetilde{C} = [A_2(b_t c_{t-1} - c_t B_0) + A_3]$, we get:

$$
\begin{vmatrix} y_t \\ w_t \end{vmatrix} + \begin{vmatrix} \widetilde{A} A_2 (a_t + b_t a_{t-1}) \\ 0 \end{vmatrix} + \begin{vmatrix} \widetilde{A}\widetilde{B} & -\widetilde{A}\widetilde{C} B_0 \\ 0 & B_0 \end{vmatrix} \begin{vmatrix} y_{t-1} \\ w_{t-1} \end{vmatrix} + \begin{vmatrix} \widetilde{A} A_4 & -\widetilde{A}\widetilde{C} B_1 \\ 0 & B_1 \end{vmatrix} \begin{vmatrix} \varepsilon_{1,t} \\ \varepsilon_{2,t} \end{vmatrix} = 0, \quad (1.45)
$$

We repeat the previous steps and define auxiliary matrices:

$$
\begin{aligned}
\alpha_t &= \begin{vmatrix} y_t \\ w_t \end{vmatrix}, \ddot{\mu}_t = \begin{vmatrix} \widetilde{A} A_2 (a_t + b_t a_{t-1}) \\ 0 \end{vmatrix}, \xi_t = \begin{vmatrix} \varepsilon_{1,t} \\ \varepsilon_{2,t} \end{vmatrix}, \\
\ddot{T}_t &= \begin{vmatrix} \widetilde{A}\widetilde{B} & -\widetilde{A}\widetilde{C} B_0 \\ 0 & B_0 \end{vmatrix}, \ddot{R}_t = \begin{vmatrix} \widetilde{A} A_4 & -\widetilde{A}\widetilde{C} B_1 \\ 0 & B_1 \end{vmatrix},
\end{aligned}
$$

which allows us to write the state-space system:

$$\alpha_t + \ddot{\mu}_t + \ddot{T}_t \alpha_{t-1} + \ddot{R}_t \xi_t = 0 \quad (1.46)$$

$$Y_t^{obs} - H\alpha_t = 0, \quad (1.47)$$

This system is, as expected, different from both its *time t* and *time t-1* counterparts.

VAR Learning The Perceived Law of Motion described by (1.22) is known as the *Minimum State Variable* (MSV) solution of the system. In this solution, we allow agents to know the reduced-form of the structural model, but not its coefficients, for which they generate estimates. Note that estimation of (1.22) or its *time t-1* counterpart could (and probably would, for larger models) require agents to forecast non-observable variables, which requires the use of the Kalman Filter and, also important, the knowledge of *which* non-observables would be present in the model. For this reason, sometimes it makes more sense to restrict agents to a set of observable variables only. In this case, the Perceived Law of Motion takes the form:

$$y_t = a_{t-1} + b_{t-1} y_{t-1}^{obs} + c_{t-1} w_{t-j}^{obs}, \quad (1.48)$$

where only a subset of variables in y and w are observables and, again, we can set $j = 0$ or $j = 1$, restricting the avaiable information to the agents.

As $Y_t^{obs} = H\alpha_t$ and $w_t^{obs} = H_w w_t$ describes the observables variables in our model, this Perceived Law of Motion can be rewritten as:[27]

$$y_t = a_{t-1} + b_{t-1} H y_{t-1} + c_{t-1} H_w w_{t-j},$$

and then inserted into the economic model to get the Actual Law of Motion as before:

$$A_0 y_{t-1} + A_1 y_t + A_2[a_t + b_t H y_t + c_t H_w w_t] + A_3 w_t + A_4 \varepsilon_{1,t} = 0 \quad (1.49)$$

$$w_t + B_0 w_{t-1} + B_1 \varepsilon_{2,t} = 0, \quad (1.50)$$

with, again, defining auxiliary matrices:

[27] Some authors allow only for the observable endogenous variables to be present in the agents' PLM, so that $H_w = 0$.

$$\alpha_t = \begin{vmatrix} y_t \\ w_t \end{vmatrix}, \dddot{\mu}_t = \begin{vmatrix} \widetilde{A}A_2 a_t \\ 0 \end{vmatrix}, \xi_t = \begin{vmatrix} \varepsilon_{1,t} \\ \varepsilon_{2,t} \end{vmatrix},$$
$$\dddot{T}_t = \begin{vmatrix} \widetilde{A}A_0 & -\widetilde{A}\widetilde{C}B_0 \\ 0 & B_0 \end{vmatrix}, \dddot{R}_t = \begin{vmatrix} \widetilde{A}A_4 & -\widetilde{A}\widetilde{C}B_1 \\ 0 & B_1 \end{vmatrix},$$

with now $\widetilde{A} = (A_1 + A_2 b_t H)^{-1}$ and $\widetilde{C} = (A_2 c_t H_w + A_3)$.

Other types of learning We described here the main set of possible Perceived Laws of Motion. These are not, however, the only ones. The possible information could be, in principle,[28] huge; any under or overparametrization of the Perceived Law of Motion can be used to model agents' expectations, each of them leading to a different Actual Law of Motion and altering the system dynamics.

1.2.3 Initial beliefs and learning algorithm

Learning algorithms Suppose that our economy is given by (1.20) and (1.21) and our reduced-form model given by (1.22). If y_t and ε_{1t} are $n \times 1$ vectors of endogenous variables and shocks, and w_t is a $k \times 1$ vector of exogenous/predetermined variables, our matrices a_t, b_t and c_t have dimensions $n \times 1$, $n \times n$ and $n \times k$, respectively. We can therefore represent (1.22) as:

$$\begin{vmatrix} y_{1,t} \\ ... \\ y_{n,t} \end{vmatrix} = \begin{vmatrix} a_{1,t-1} \\ ... \\ a_{n,t-1} \end{vmatrix} + \begin{vmatrix} b_{11,t-1} & ... & b_{1n,t-1} \\ ... & ... & ... \\ b_{n1,t-1} & ... & b_{nn,t-1} \end{vmatrix} \begin{vmatrix} y_{1,t-1} \\ ... \\ y_{n,t-1} \end{vmatrix} +$$
$$\begin{vmatrix} c_{11,t-1} & ... & c_{1k,t-1} \\ ... & ... & ... \\ c_{n1,t-1} & ... & c_{nk,t-1} \end{vmatrix} \begin{vmatrix} w_{1,t} \\ ... \\ w_{k,t} \end{vmatrix} + \begin{vmatrix} \varepsilon_{11,t} \\ ... \\ \varepsilon_{1n,t} \end{vmatrix}, \quad (1.51)$$

or, in matrix notation, as:

$$y_t = \beta'_{t-1} Z_t + \varepsilon_{1,t}, \quad (1.52)$$

[28] Obviously, the amount of variables we can use to model expectations is limited by the sample size. This will be more important the more endogenous variables are present in the model, since each new exogenous variable included in the model is present in all equations.

where we define β'_{t-1} as a $n \times (1+n+k)$ matrix of coefficients and Z_t as a $(1+n+k) \times 1$ matrix of regressors.

$$\beta_{t-1} = \begin{vmatrix} a_{t-1} \\ b_{t-1} \\ c_{t-1} \end{vmatrix}, Z_t = \begin{vmatrix} I \\ y_{t-1} \\ w_t \end{vmatrix} \tag{1.53}$$

An updating algorithm relates the evolution of β_t over time. The *Recursive Least Squares* (RLS) algorithm is given by running (1.52) by least squares. In this case, our vector β_{t-1} is given by:

$$\beta_{t-1} = (\sum_{i=1}^{t-1} Z_i Z_i')^{-1} (\sum_{i=1}^{t-1} Z_i y_i'), \tag{1.54}$$

or, recursively, by:

$$\beta_t = \beta_{t-1} + t^{-1} R_t^{-1} Z_{t-1}(y_t - \beta'_{t-1} Z_{t-1}) \tag{1.55}$$

$$R_t = R_{t-1} + t^{-1}(Z_{t-1} Z'_{t-1} - R_{t-1}), \tag{1.56}$$

where R_t is the second-moment matrix of Z_t. Given initial values, β_0 and R_t, we can use data to update β_t and R_t over time.

There are alternate updating algorithms to equations (1.55) and (1.56). We can write these equations as:

$$\beta_t = \beta_{t-1} + g_t R_t^{-1} Z_{t-1}(y_t - \beta'_{t-1} Z_{t-1}) \tag{1.57}$$

$$R_t = R_{t-1} + g_t(Z_{t-1} Z'_{t-1} - R_{t-1}), \tag{1.58}$$

where g_t is a gain sequence, which determine how fast β_t can change over time in response to past prediction errors. The usual RLS has decreasing gain $g_t = t^{-1}$, while constant-gain RLS (CG-RLS) assumes that $g_t = g$, $0 < g < 1$, with g usually small. Stochastic Gradient Constant Gain (SG-CG) learning uses $g_t = gR_t$, $g > 0$, so that the second moment recursion (1.58) is not used in the update equation for the reduced-form coefficients (which is computationally less intensive and can prevent some numerical problems that leads the betas to diverge). See Evans & Honkapohja (2001), Slobodyan (2006) and Evans, Honkapohja & Williams (2008) for a more detailed survey of these and other Stochastic Recursive Algorithms.

Initial beliefs The algorithms described above require the specification of initial beliefs, which are given by matrices β_0 and R_0. These can be obtained in, at least, three different ways:

1) Arbitrarily fixed: this is the simplest way of initialising beliefs. The matrices can come, for example, from subsample regressions, setting zero (for β) and identity (for R) matrices, or any choice of interest.

2) RE consistent: initalialising the beliefs in a rational expectations consistent way requires us to solve the model *as if* rational expectations was in place, for a given vector of structural parameters. The actual law of motion induced by these parameters will have a unique correspondence with both matrices β and R, since the structural parameters involve both law-of-motion and covariance matrix parameters.

3) Optimal initial belifs: one can also initialise β and R optimally, in a likelihood sense. This means that the search for the posterior mode (or max) involves not only the initial set of structural parameters but also the initial beliefs matrices. Since the latter ones can have a much higher dimensionality, this process can impose practical time restrictions to one trying to estimate different models.

Slobodyan and Wouters (2009) showed that a relevant part of the posterior/likelihood improvement in some estimated learning DSGE models comes from estimating optimal initial beliefs. It is important, hence, to control for the type of initial beliefs when assessing the possible advantages of learning over rational expectations.

1.2.4 Putting it all together: our MATLAB code

We chose to build our own MATLAB code[29] since our analysis require a high degree of customization, not available yet in DYNARE. Slobodyan & Wouters (2009) did change DYNARE's code to estimate learning DSGE models, however their code can only estimate a subset of the cases we consider here. Also, their code relies on the the correctness of DYNARE's internal codes. In fact, even though DYNARE is a very powerful software, it has showed a lot of bug corrections since the version used by Slobodyan and Wouters.

[29]The set of codes is available at FEA-USP's website, under Professor Pedro Garcia Duarte personal space, or via direct link at: http://www.fea.usp.br/feaecon/download/LDM.zip. It is also available upon request at: igorvele at gmail dot com. More details are given in Appendix E.

DYNARE's main advantages are the ability to write down models as equations in a non-linear fashion (not in a linear system of difference equations, like ours) with a built-in linearization tool, and the computationally efficient code and calculations (for example, calculating the likelihood function using matrices and not loops), which perform tasks in an efficient way. On the other hand, these efficient calculations create a very complex set of codes, in which customization is difficult[30] – not exactly a black box, but a huge fixed cost for additional code customization – that cannot guarantee compatibility with new versions of DYNARE.[31] Our code relies on straightforward matrix algebra and loops that can be customized in a simpler way.

There are other DSGE learning softwares avaiable, such as those of Carceres-Poveda (2006) and of Gaus (2013), which are based on C or MATLAB. Again, their estimation softwares allow for only a subset of all our mechanisms, with the exception of Gaus's, that is configured to run endogenous gain (which our code does not run yet).

Here we summarize the steps taken by our code, which allows us to estimate learning DSGE models under the EE approach. We use seven major inputs:

(i) *in_model*: identifies the *m-file* containing the description of the model (its equations in matrix notation, its parameters, priors distributions and data source);

(ii) *in_data*: identifies the type of data being used (real or simulated);

(iii) *in_sims*: number of simulations (draws) on the Metropolis-Hastings steps;

(iv) *in_learning_type*: allows for Minimum State Variables and VAR learning, with or without constants;

(v) *in_ini_beliefs*: allows for three types of initial beliefs (1 - fixed at RE maximum: finds posterior maximum under rational expectations and uses it as initial beliefs; 2 - RE consistent: finds posterior max using initial beliefs consistent with each vector of parameters (under learning); 3 - Optimized: adds the vector of initial beliefs to the vector of structural parameters and searches for a global posterior maximum);

(vi) *in_learning_algo*: set the algorithm to Stochastic Gradient or Constant Gain (and Time-Varying Parameters in one special case);

(vii) *in_timing_PLM*: allows two PLM timing options (t and $t-1$)

[30] For instance, it is easier to customize models using loops instead of matrices, since all operations are straightforward and require little learning time.

[31] Since DYNARE is a separate project, there is no guarantee that the name of global variables and parameters (inside the code) will be the same from one version to another.

There are other settings that can be changed inside our code. For example, $search\_method$ allows one to change the posterior maximum searching algorithm to $fminsearch$, $fminunc$ or $csminwel$; $search\_RE$ enables the code to use the rational expectations posterior maximum as a guide to the learning posterior maximum; $search\_RE\_rob$ and $search\_robust$ sets the number of vectors to perform a more robust search for the posterior maximum; $search\_mcmc$ allows the code to search for the posterior maximum at MCMC steps (this is restricted, in our code, to the burn-in draws); other options include display options (such as showing the posterior maximum candidates, displaying warnings and penalties due to numerical instabilities) and others, such as maximum number of iterations for $fminsearch/fminunc/csminwel$, burn-in draws in MCMC, penalties for out-of-bond priors and so on.

Given these main seven major inputs, our code reads the specified model m-file, finds the posterior maximum for rational expectations and learning model and then starts the MCMC steps, drawing from a Random-Walk-Metropolis-Hastings algorithm. Our code automatically discards the first 30% (which can also be customized) draws to calculate the mean and standard deviation of the estimated parameters. In Appendix C, we give the reader the steps to calculate the likelihood of a time-varying state-space; appendix D reviews the basics of Bayesian estimation.

1.3 Estimations

In this section, we describe the first two models we will work with and how we expect our approach to capture problems that can be inserted into them.

1.3.1 Models and simulated data

We simulate data using two different models. The first one is given by:

$$y_t = \alpha y_{t-1} + \beta E_t y_{t+1} + Z_t \tag{1.59}$$

$$Z_t = \rho Z_{t-1} + \varepsilon_t \tag{1.60}$$

which is a model with one endogenous variable (y_t) that depends on its previous and expected future values and one exogenous variable (Z_t) that can be set to an *iid* disturbance if $\rho = 0$. The second model comes from Gali (2002), a New-Keynesian 3-equation macro model that describes the joint evolution of inflation (cpi), output gap (gdp) and nominal interest rates (ff). The model is given by:

$$cpi_t = \beta E_t cpi_{t+1} + \kappa gdp_t + \varepsilon_{1t} \quad (1.61)$$

$$gdp_t = E_t gdp_{t+1} - \sigma ff_t + \sigma E_t cpi_{t+1} + \varepsilon_{2t} \quad (1.62)$$

$$ff_t = \rho_i ff_t + \phi_1 E_t cpi_{t+1} + \phi_2 gdp_t + \varepsilon_{3t} \quad (1.63)$$

$$\varepsilon_{1t} = \rho_1 \varepsilon_{1t-1} + \xi_{1t} \quad (1.64)$$

$$\varepsilon_{2t} = \rho_2 \varepsilon_{2t-1} + \xi_{2t} \quad (1.65)$$

$$\varepsilon_{3t} = \rho_3 \varepsilon_{3t-1} + \xi_{3t} \quad (1.66)$$

where ε_{it} are exogenous variables that can be set to an iid disturbance if $\rho_1 = \rho_2 = \rho_3 = 0$.

These are our two benchmark models. Model 1 is one of the simplest rational expectations models that can be tested against learning, and will allow us to understand the contributions and limitations of our learning mechanism.[32] Model 2 (the basic New Keynesian model) involves a greater set of variables and could be thought as a small structural VAR describing the economy. We will simulate data for model 1 and 2 and will use real data for model 2.

Our goal here is to use the learning approach to understand how well our models fit the data. To do that, we first estimate the two models with artificial data simulated from these models with rational expectations. We are interested in introducing a structural break in the artificial data to see how learning dynamics can adapt to this situation. Thus, we follow two steps:

(1) Simulate rational expectations data: for a given vector of structural parameters (in model 1, $\alpha = 0.5$, $\beta = 0.3$, $\rho = 0$, $\sigma_{\varepsilon t} = 1$; in model 2, $\beta = 0.99, \kappa = 0.4$, $\sigma = 0.5$, $\phi_1 = 0.7$, $\phi_2 = 0.7$, $\rho_i = 0.5$, $\rho_1 = \rho_2 = 0.5$, $\rho_3 = 0.3$, $\sigma_{\xi_1} = 0.5$,$\sigma_{\xi_2} = 0.5$, $\sigma_{\xi_3} = 0.5$), we simulate 1000 data points for each model with rational expectations. Wtih this data we

[32] An even simpler model could be considered by setting $\alpha = \rho = 0$. However, in this case the *Minimum State Variable* solution would be $y_t = c$, where c is some constant. Testing this model against learning would be the equivalent to test a time varying constant structure to the model, which is too simple to be considered here. The idea of estimating an univariate model is to replicate the forward and backward looking structure of larger models in an univariate model.

consider the learning models, estimating the parameters and the gain parameter with our code, using different learning mechanisms.

(2) Simulate artificial rational expectations data: we consider here a structural break by simulating with rational expectations using a specific parameter vector for the first 500 observations and another vector for the last 500 for each of the two models. For model 1, we simulate the first 500 observations using $\alpha = 0.5$, $\beta = 0.3$, $\rho = 0$, $\sigma_{\varepsilon t} = 1$. For the last 500 observations, we set $\alpha = 0.6$. We do the same to model 2, with $\beta = 0.99, \kappa = 0.4$, $\sigma = 0.5$, $\phi_1 = 0.7$, $\phi_2 = 0.7$, $\rho_i = 0.5$, $\rho_1 = \rho_2 = 0.5$, $\rho_3 = 0.3$, $\sigma_{\xi_1} = 0.5$,$\sigma_{\xi_2} = 0.5$, $\sigma_{\xi_3} = 0.5$ for the first half and we change $\sigma = 0.3$ for the last 500 observations.

Step one is just a "check step" to make sure that our code is estimating parameters correctly and that the random numbers algorithm generated good data (which means that the parameters shall be estimated properly with the gain parameter being equal to zero).

Step two is central to our thesis: in a simulated environment, we are able to account more precisely the sources of contribution of the learning mechanism to the improvement of the posterior distribution of the parameters. Specifically, we are interested in answering (i) if giving agents less information (allowing them to have a time $t-1$ perceived law of motion) leads to an increase in the fit and (ii) is there a clear difference between stochastic gradient and constant gain updating algorithms?

1.3.2 A larger model

In addition to estimating those two models with learning using simulated data, we also estimate model 2 using real data (which will be described below). The goal is to understand how our learning scheme adapts to a real environment. We follow Christiano, Eichenbaum & Evans (2005) and Smets & Wouters (2003, 2007) in recognizing, however, that model 2 seems too stylized to describe the joint evolution of inflation, nominal interest rates and output gap. For that reason, we also estimate a larger model, which we introduce now.

The possible mispecification present in model 2 may favor learning, since residuals are much less likely to behave as *iid*. A larger model such as Smets & Wouters (2003), for instance, will likely do a better job describing real data than the stylized New Keynesian model and will likely produce a more competitive environment (with less mispecifications) to test our learning specifications.

We chose to estimate a simpler version of the original model (Smets and Wouters, 2003), which became a benchmark in the DSGE literature (a full detailed description of the model can be found in the original paper). The core structure of the model is given by the following set of equations:

Capital accumulation (1.67), labor demand (1.68) and goods market equilibrium (1.69) are given by:

$$
\begin{aligned}
K_t &= (1-\tau)K_{t-1} + \tau I_{t-1} &\quad (1.67)\\
L_t &= -W_t + (1+\varphi_c)R_t^K + K_{t-1} &\quad (1.68)\\
Y_t &= (1-\tau k_y - g_y)C_t + \tau k_y I_t + \epsilon_t^g &\quad (1.69)
\end{aligned}
$$

where the (usual) law of motion of the stock of capital (K_t) evolves over time through a depreciation rate τ and addition of new investment (I_t). Hours worked (L_t) are function of real wages (W_t), rental rate of capital (R_t^K) and the stock of capital at time $t-1$. Goods market equilibrium equation states that total output (Y_t) is the weighted sum of consumption (C_t), investment and a government expenditure shock ϵ_t^g; φ_c is the inverse elasticity of capital utilizarion cost, while k_y is the capital income share; g_y is the government expenditures shares in output.

Next we have the production function (1.70), Taylor rule (1.71), and the consumption equation (1.72).

$$Y_t = \phi\epsilon_t^a + \phi\alpha K_{t-1} + \phi\alpha\varphi_c R_t^K + \phi(1-\alpha)L_t \quad (1.70)$$

$$
\begin{aligned}
R_t = \rho R_{t-1} + (1-\rho)\left[\bar{\pi}_t + r_\pi(\pi_{t-1} - \bar{\pi}_t) + r_y(Y_t - Y_t^P)\right] + \\
r_{\Delta\pi}(\pi_t - \pi_{t-1}) + r_{\Delta y}\left[(Y_t - Y_t^P) - (Y_{t-1} - Y_{t-1}^P)\right] + \eta_t^R \quad (1.71)
\end{aligned}
$$

$$C_t = \frac{h}{1+h}C_{t-1} + \frac{1}{1+h}E_t C_{t+1} - \frac{1-h}{(1+h)\sigma_c}(R_t - E_t\pi_{t+1}) + \frac{1-h}{(1+h)\sigma_c}\epsilon_t^b \tag{1.72}$$

where the production function (1.70) relates output as a function of capital, the rental rate of capital, labor and a productivity shock ϵ_t^a. α is the capital output ratio, while ϕ denotes the production function fixed cost. φ_a is the inverse elasticity of investment adjusment cost.

The Taylor rule equation states that the interest rated (R_t) is a function of lagged interest rates (smooth parameter ρ), deviations of inflation from its target (π_t and $\bar{\pi}_t$, respectively), deviations of output from its potential (Y_t and Y_t^P), *changes* of inflation and output gap from its $t-1$ levels (through parameters $r_{\Delta\pi}$ and $r_{\Delta y}$, respectively), and a monetary policy shock, η_t^R.

Consumption is a function of lagged and expected consumption (weighted by an external habit parameter h), real *ex-ante* interest rates and a preference shock ϵ_t^b; σ_c denotes the relative risk aversion coefficient.

Last we have the investment (1.73), value of capital (1.74), inflation (1.75) and wage (1.76) equations.

$$I_t = \frac{1}{1+\beta}I_{t-1} + \frac{\beta}{1+\beta}E_t I_{t+1} + \frac{\varphi_a}{1+\beta}Q_t + \epsilon_t^i \tag{1.73}$$

$$\begin{aligned} Q_t = -(R_t - E_t\pi_{t+1}) + \frac{1-\tau}{1-\tau+\overline{RK}}E_t Q_{t+1} + \\ + \frac{\overline{RK}}{1-\tau+\overline{RK}}E_t R_{t+1}^K + \eta_t^Q \end{aligned} \tag{1.74}$$

$$\begin{aligned} \pi_t = \frac{\beta}{1+\beta\gamma_p}E_t\pi_{t+1} + \frac{\gamma_p}{1+\beta\gamma_p}\pi_{t-1} + \\ + \frac{\beta}{1+\beta\gamma_p}\frac{(1-\beta\xi_p)(1-\xi_p)}{\xi_p}\left[\alpha R_t^K + (1-\alpha)W_t - \epsilon_t^a\right] + \eta_t^p \end{aligned} \tag{1.75}$$

$$W_t = \frac{\beta}{1+\beta}E_t W_{t+1} + \frac{1}{1+\beta}W_{t-1} + \frac{\beta}{1+\beta}E_t\pi_{t+1} - \frac{1+\beta\gamma_w}{1+\beta}\pi_t + \frac{\gamma_w}{1+\beta}\pi_{t-1}$$
$$- \frac{\beta}{1+\beta}\frac{(1-\beta\xi_w)(1-\xi_w)}{(1+\frac{(1+\lambda_w)\sigma_L}{\lambda_w})\xi_p}\left[W_t - \sigma_L L_t - \frac{\sigma_c}{1-h}(C_t - hC_{t-1}) + \epsilon_t^L\right]$$
$$+ \eta_t^w \quad (1.76)$$

Investment is a function of its past and expected values, the value of capital (Q_t) at time t and a shock ϵ_t^i. The value of capital depends negatively on the real *ex-ante* interest rates, on its expected value at $t+1$, on the expected rental rate of capital and on an equity premium shock η_t^Q.

Inflation and wage equations are closely linked: inflation depends on its past and expected values (weighted by the price indexation parameter γ_p), on the rental rate of capital, real wages, productivity shock and an inflation shock η_t^p. The parameter ξ_p is the Calvo price stickiness.

Wages are a function of past and expected wages, past, present and expectated inflation, hours worked, deviations of consumption from its habitual level, the labor supply shock, and a shock η_t^w. ξ_w refers to the wage indexation parameter; σ_L is the inverse elasticity of labor supply and λ_w is the Calvo wage stickiness parameter.

The model is then completed by specifying the structure of exogenous variables ϵ and η, which are modeled as AR(1) processes.[33] For simplicity, we also assume that potential output is observable, which does not require the construction of a separate flexible system.[34] The model consists on 12 endogenous variables (Y, C, I, R, W, π, $\bar{\pi}$, L, R^K, Y^P, K, Q), 9 observables, except K, R^K and Q and a total of 42 parameters (19 structural, 22 related to exogenous processes and one gain parameter).

The model is estimated using seasonally adjusted quarterly data for Brazil, from 2002Q3 to 2012Q4. The short time span is mainly due to large structural changes in

[33] We also need to model the inflation target, which is both observable and time-varying for the Brazilian economy. We model its first difference as an exogenous *iid* process.

[34] The potential output Y_t^P is given by the level of output consistent with flexible prices. It is therefore required that a separate system must be built (setting all friction parameters to zero) to produce the potential output at each period. We chose not to do so here.

Brazilian economy (inflation targetting was only introduced in June 1999) and availability of reliable labor data (which starts in 2002). Data sources are IBGE (GDP, inflation, wages, working age population, hours worked and payroll) and BCB (interest rate and inflation target). Some parameters are calibrated (depreciation rate, output ratios) and the others are estimated.

As we argued before, this more complex model (with leads and lags) will have a richer solution of the model, with an improved dynamics (in relation to the 3-equation model New Keynesian model). Our learning mechanism can now be tested in a more realistic environment.

1.4 Main results

Table 1.2 summarizes the results[35] for the univariate model. The first column stands for the different specifications used in the estimation, from the (not numbered) benchmark rational expectations specification to the sixteen learning specifications, which relates to different Perceived Law of Motions types (second column), initial beliefs (third column), learning algorithm (fourth column, in which SG stands for Stochastic Gradient and CG for Constant Gain learning algorithm), Perceived Law of Motion timing (fifth column), Posterior distribution value at the maximum (sixth column), and the estimated gain parameter (last column).

Our main findings are: first, there is at least one learning setup[36] that is able to outperform rational expectations as it is the case for all models whose gain parameter (last column) is positive. Of all 16 specifications estimated, 14 performed better than rational expectations. The best performing model setup is given by MSV Perceived Law

[35] Detailed tables are available in Appendix A. Posteriors and (marginal) likelihoods values are evaluated at the parameter vector $\theta^{\max}$ that produces the best posterior. If we were comparing different models with different priors, our measure or marginal likelihood would be affected and could not be calculated as we do here. However, since we are dealing only with different specifications of learning under the same model (and the same priors), our measure of likelihood can be used without concerns.

[36] We will abreviate from now on. For the Perceived Law of Motion, MSV stands for Minimum State Variable; MSVc stands for Minimum State Variable with constant, while VAR and VARc are intuitive; for the learning algorithm, CG stands for constant gain and SG for stochastic gradient.

Table 1.2: Results for model 1, simulated data

Spec number	Learning Type	Initial beliefs	Learning algorithm	Timing	Posterior	gain
	RE				3261.90	
1	VAR	RE consis	SG	t	3251.84	0.0035
2	VAR	RE consis	SG	t-1	3251.84	0.0035
3	VAR	RE consis	CG	t	3251.60	0.0094
4	VAR	RE consis	CG	t-1	3251.60	0.0094
5	VAR c	RE consis	SG	t	3253.10	0.0032
6	VAR c	RE consis	SG	t-1	3253.10	0.0032
7	VAR c	RE consis	CG	t	3253.44	0.0078
8	VAR c	RE consis	CG	t-1	3253.44	0.0078
9	MSV	RE consis	SG	t	3261.90	-
10	MSV	RE consis	SG	t-1	3251.67	0.0037
11	MSV	RE consis	CG	t	3261.90	-
12	MSV	RE consis	CG	t-1	3251.09	0.0085
13	MSV c	RE consis	SG	t	3261.63	0.2000
14	MSV c	RE consis	SG	t-1	3252.98	0.0034
15	MSV c	RE consis	CG	t	3261.75	0.1293
16	MSV c	RE consis	CG	t-1	3252.92	0.0080

of Motion with timing $t-1$ and CG learning algorithm, 12^{th} specification; the best performing model showed a gain parameter[37] of 0.0085.

Second, there is no clear advantage of constant gain over stochastic gradient, with small differences of performance across models. This is not true, however, with the Perceived Law of Motion timing: time $t-1$ setup outperformed time t by a considerable extent under MSV learning, and showed no difference under VAR learning.

Also important is the difference of posterior between specifications 11 and 12. Both are using the same setup except from the its timing. The time t performed as well as the rational expectations estimation (with gain parameter equal to zero), while time $t-1$ performed better, with a positive estimated gain parameter. As we have seen, time t and time $t-1$ Perceived Laws of Motion result in different Actual Laws of Motion.

The estimations also pointed that giving agents *less* information helped improving the fit: all VAR models performed better than rational expectations and showed positive gain parameters. The best VAR specifications (3 and 4) showed only a slightly worse posterior than the best performing model.

[37] The weight of a data point t-k is given by $g(1-g)^{t-k}$, where g is the estimated gain parameter. The weight has a half-life of $P = -\frac{\ln(2)}{\ln(1-g)} = \frac{0.69}{g}$ periods. Therefore a gain parameter with value 0.01 will give half weight to an observation after 69 periods (approximately 17 years using quarterly data)

Improvements in the likelihood with zero estimated gain are indicatives that the reduced-form equations are also helping the fit. This is not the case here, since there is no mispecification in our simulated data. Consider, for instance, a mispecified model without time variation in parameters. The learning model could improve the fit by adding lags to the right hand side of the equation, fixing the mispecification. Since there is no time variation in parameters, one would expect the estimated gain parameter to be (near) zero. In the simulated example of model 1, there was no mispecification in the original model, hence models 9 and 11 were not able to improve the fit (with zero gain) by adding lags to the right hand side. This will be an issue in the second model using real data. First, however, we analyze the results for the simulated data of model 2, which is presented below.

Table 1.3: Results for model 2, simulated data

Spec number	Learning Type	Initial beliefs	Learning algorithm	timing	Posterior	gain
	RE				4504.59	
1	VAR	ReC*	SG	t	4647.22	0.0004
2	VAR	ReC*	SG	t-1	4646.91	0.0004
3	VAR	ReC*	CG	t	4639.52	0.0028
4	VAR	ReC*	CG	t-1	4595.29	0.0012
5	VAR c	ReC*	SG	t	4651.36	0.0004
6	VAR c	ReC*	SG	t-1	4651.36	0.0004
7	VAR c	ReC*	CG	t	4610.88	0.0011
8	VAR c	ReC*	CG	t-1	4644.02	0.0022
9	MSV	ReC*	SG	t	4504.59	-
10	MSV	ReC*	SG	t-1	4501.59	0.0008
11	MSV	ReC*	CG	t	4504.59	-
12	MSV	ReC*	CG	t-1	4501.22	0.0022
13	MSV c	ReC*	SG	t	4504.59	-
14	MSV c	ReC*	SG	t-1	4504.74	-
15	MSV c	ReC*	CG	t	4504.59	-
16	MSV c	ReC*	CG	t-1	4504.90	-

Model 2 showed new important results: models with VAR learning showed a worse performance than rational expectations. This is due to the most important difference between models 1 and 2, which is the value of the shocks' persistence parameters. In model 1, the persistence parameter ρ was very low (simulated with $\rho = 0$ and estimated in sample below 0.05), while model 2 simulated data was generated with persistence parameters 0.50, 0.40 and 0.30.

This inertia is not fully captured by the VAR learning mechanism, which does not model persistence in shocks. This result suggests that learning VAR models (at least as modeled here) will perform better when errors are close to *iid*, which means that the estimation of our larger model will in fact *favor* this type of learning mechanism.

Models with Minimum State Variable solution (MSV) in their Perceived Law of Motion, on the other hand, will not suffer from the same problem, since exogenous variables are taken as observables (therefore being capable of estimating correlations of shocks). This is precisely what we observe in our estimated model: the only specifications able to outperform rational expectations were given by (i) MSV Perceived Law of Motion with timing $t-1$ and constant gain learning algorithm, which was the best model, 12^{th} specification, and (ii) MSV Perceived Law of Motion with timing $t-1$ and stochastic gradient learning algorithm, second best model, 10^{th} specification.

As in model 1, the best learning algorithm was not unanimous: constant gain presented better results in four specifications (3^{rd}, 4^{th}, 7^{th}, 8^{th}, and the best performing model, 12^{th}); stochastic gradient was better in one specification (16^{th}) and there were two ties (9^{th} with 11^{th}, 13^{th} with 15^{th}), all of them under MSV learning. The first three specifications in which constant gain showed better results, however, performed worse than rational expectations.

The discussion of giving agents less information is now complemented by the fact that if agents are given too little information, specially with respect to the persistence of shocks, the resulting VAR learning model (as least as used so far in the literature) is very likely to perform poorly. This result can be easily corrected by standard time-series tools. For example, one can add more lags to the Perceived Law of Motion VAR, or add MA terms.

The lack of information led the VAR specifications to overestimate ϕ_1 by a large extent (see corresponding detailed tables at the end of the thesis): the estimated ϕ_1 ranged from 1.48 to 2.36, while its true value was 0.95. This is not surprising, since ϕ_1 is the response of interest rates to inflation and the inflation shock is the most persistent of the system.[38]

The estimated gain parameter was smaller than the one estimated in model 1: 0.0008 (against 0.0085). The posterior/likelihood improvement was also smaller, of only 3.4 points (4501.2 under specification 12, against 4504.6 of rational expectations).

[38] The variance parameter var1 (related to the inflation equation was also the most overestimated under VAR learning).

The results using Brazilian real data showed larger improvements, which are presented next.

Table 1.4: Results for model 2, real data

Spec number	Learning type	Initial beliefs	Learning algorithm	Timing	Posterior	gain
	RE				-508.13	
1	VAR	ReC*	SG	t	-434.36	-
2	VAR	ReC*	SG	t-1	-434.36	-
3	VAR	ReC*	CG	t	-326.87	-
4	VAR	ReC*	CG	t-1	-349.60	-
5	VAR c	ReC*	SG	t	-434.37	-
6	VAR c	ReC*	SG	t-1	-434.37	-
7	VAR c	ReC*	CG	t	-397.19	-
8	VAR c	ReC*	CG	t-1	-444.46	-
9	MSV	ReC*	SG	t	-508.63	0.1000
10	MSV	ReC*	SG	t-1	-508.29	0.0034
11	MSV	ReC*	CG	t	-507.35	-
12	MSV	ReC*	CG	t-1	-507.19	-
13	MSV c	ReC*	SG	t	-519.39	0.0118
14	MSV c	ReC*	SG	t-1	-522.26	0.0018
15	MSV c	ReC*	CG	t	-508.03	-
16	MSV c	ReC*	CG	t-1	-527.63	0.0030

The best performing specification (16^{th}) was MSV learning with constants, time $t-1$ Perceived Law of Motion timing and constant gain learning algorithm. It showed an improvement of 19.5 points, (from -508.1, rational expectations, to -527.6) which is equivalent to 464 points in comparison to the 3.4 points of simulated data.[39] The estimated gain parameter for the best specification was 0.003, higher than with simulated data (0.0008).

It is also important to notice that specifications 9 and 13 are different only by the presence of constants, which account for a large part in the posterior/likelihood improvement. This is in line with data for emerging countries, which are often subject to structural breaks. For instance, Brazil implemented inflation targetting regime in 1999, and the targets changed 7 times in the five subsequent years, stabilizing at 4.5% after 2005.

[39] Considering now that the lenght of the data is $T = 42$ instead of $T = 1000$, an equivalent rule of thumb comparison of posterior improvements would be $\frac{1000}{42}\frac{19.5}{1} = 464.3$, where we assume that the likelihood improvements is proportional to the sample size (which is now 1000).

Again, all VAR models performed worse than rational expectations, most likely in consequence of some persistence present in shocks. Only five specifications performed better than rational expectations: 9^{th}, 10^{th}, 13^{th}, 14^{th} and 16^{th}, which was the best performing.

Results for the third model (simpler version of Smets & Wouters) showed a modest improvement of fit relatively to the rational expectations specification.

Table 1.5: Results for model 3, real data

Spec number	Learning Type	Initial beliefs	Learning algorithm	Timing	Posterior	gain
	RE				-1302.91	
1	VAR	RE consis	SG	t	-1264.69	-
2	VAR	RE consis	SG	t-1	-1192.75	-
3	VAR	RE consis	CG	t	-1266.33	-
4	VAR	RE consis	CG	t-1	-1254.01	-
5	VAR c	RE consis	SG	t	-1168.65	-
6	VAR c	RE consis	SG	t-1	-1168.33	-
7	VAR c	RE consis	CG	t	-1253.21	-
8	VAR c	RE consis	CG	t-1	-1244.13	-
9	MSV	RE consis	SG	t	-1302.87	-
10	MSV	RE consis	SG	t-1	-1302.88	-
11	MSV	RE consis	CG	t	-1302.70	-
12	MSV	RE consis	CG	t-1	-1302.83	-
13	**MSV c**	**RE consis**	**SG**	**t**	**-1305.21**	**0.0055**
14	**MSV c**	**RE consis**	**SG**	**t-1**	**-1303.35**	**0.0041**
15	MSV c	RE consis	CG	t	-1302.74	-
16	MSV c	RE consis	CG	t-1	-1302.91	-

The improvement in posterior from the rational expectations to the best performing specification was about 2.3 points. The best fitting specification (11^{th} line) had the following setup: MSV Perceived Law of motion with constants and timing t; stochastic gradient learning algorithm. The estimated learning parameter was lower than model 1 but higher than model 2: 0.0055 in the best model and 0.0041 in the second best model (14^{th} specification, MSV Perceived Law of Motion, with constants, timing $t-1$, stochastic gradient learning algorithm).

Since the number of variables between models 3 and 2 is now different (9 observables for model 3 against 3 observables for model 2), likelihood/posterior comparison is more complicated to perform. Following our rule of thumb will tell us that the 2.3 points improvement in model 3 was about 0.8 point when comparing to model 2 with real data[40] and about 127 points when comparing to simulated data lenght of model 2 (where T=1000).[41]

As model 2 with real data, VAR models presented a worse performance than MSV learning; the average VAR model performed 76 points worse than rational expectations. We highlight that with a larger DSGE model, improvements in the posterior were more modest, as expected.

1.5 Conclusions

The learning mechanism helped fit the models considered here not only the artificial data with time-variation in the structural parameters but also of the Brazilian economy in the 2002-2012 period. The best Learning model improved the posterior by 10.8 points and 3.4 points, for the simulated data of models 1 and 2, respectively, relative to their rational expectations counterparts. Learning was able to improve the fit through two channels: (a) the new dynamics created by the addition of a reduced-form Perceived Law of Motion, which can account for mispecification problems; and (b) the time time variation of the parameters of these laws. Our artificial (non mispecified) data for models 1 and 2 showed improvements through channel (b), while real data (specially model 2) showed improvements coming from both channels.

Real data estimations suggested that learning was able to outperform rational expectations by a larger extent: using quarterly Brazilian data from 2002 to 2012, model 2 showed a posterior improvement of 19.5 points,[42] while model 3 showed an improvement of 2.3 points.[43] The intuitive idea of using a larger model in the horse race between learning mechanisms and rational expectations was supported by our data: the improvement of posterior due to the learning mechanism was *smaller* when using our version of Smets & Wouters model relative to the standard New Keynesian model of Gali (2002). This is an indicative that learning can act as a possible substitute of Calvo-type frictions, since

[40] Since the number of observables are three times bigger in model 3.

[41] This is just $\frac{464*0.8}{3} = 123.7$.

[42] When controlled for the sample size (to match $T = 1000$ of the simulated data) is approximately 464.

See discussion in the previous section.

[43] When controlled for the sample size and number of observables (to match $T = 1000$ and $N_{obsy} = 3$ of the simulated data) is approximately 123.7. See discussion above.

the improvements in smaller models was much larger than improvements in our Smets & Wouters model.

Except for model 1, which (by construction) had shocks very close to *iid*, VAR learning specifications presented a worse performance than MSV. The reason is that under VAR learning exogenous variables (persistent shocks) are not modeled, hence there is less persistence in the final model dynamics, diminishing the potential explanatory power of this type of reduced-form Perceived Law of Motion. Under MSV learning, this persistence is modeled, allowing the final model dynamics to fully match the persistence present in shocks. Hence the there is no loss of potential explanatory power under MSV learning. This result suggests that learning VAR models (at least as modeled in here) will perform better when persistence of errors are close to zero, which means that the estimation of our larger model (in the sense of having more lagged variables and frictions) may favor this type of learning mechanism.

All best performing learning models showed positive gains, which indicates that the *time variation* of reduced-form parameters and their effects through the Actual Law of Motion are important to account for the evolution of data. The results suggest that learning can improve the fit of DSGE models, specially if the model is potencially mispecified, as our estimation of models 2 and 3 using real data showed. This feature is present when a time $t-1$ specification shows a better posterior than its time t counterpart (specifications 14^{th} and 13^{th} for model 2, under real data, for instance), or when models with zero estimated gain showed a better fit than the rational expectations model, not present here. Results using simulated data (with only one occurance of structural break in data), however, showed a more modest improvement in fit. We will discuss in more details why this learning approach displayed a modest improvement in the next two chapters. Before that, we would like to point out some interesting topics for future research that were brought up by the analysis of this chapter.

A first issue would be to allow either for more lags of the endogenous variables or moving average terms (MA) in agents' reduced-form model. In this case, VAR models would perform as well as Minimum State Variable specifications. This would improve the fit of the VAR learning specifications of models 2 and 3, providing a cleaner comparison between the different types of learning. Second, one could consider performing a sensitivity anaysis under learning: this would require the estimation of all 16 specifications of model 3, each one of them setting one friction (structural) parameter to zero, in order to assess more precisely how much (for the same data/number of observables) the learning mechanism increases its improvement relative to the model with all frictions. This would give an interesting intuition on how well learning is able to compensate for each friction separatedly.

A third issue would be to estimate learning models with more specifications, for instance: (a) optimised or fixed beliefs (which will produce a total of 48 specifications), which in Slobodyan and Wouters (2009) played an important role for improving the fit of the model;[44] and (b) Kalman Filter and Endogenous gain learning, which will bring the number of specifications up to 96. Our MATLAB code is ready to implement only the first set of extensions (item (a)).

Fourth, it would be interesting to compare the posterior/likelihood improvement of our learning mechanism in models 1 and 2 (using simulated data) to the improvements coming from models specifically designed to that of problem, for instance, Markov Switching DSGE models, as described in Sims & Zha (2006), Bianchi (2009) and Liu and Mumtaz (2010). The goal is not to compare if the learning models perform better than Markov Switching (they will likely not, as we discuss in the next chapter), but to assess how much of the fit can be achieved just through our simple and flexible learning mechanism. It would also be interesting to compare if the Markov Switching model performs better than learning using our real data for Brazil.

Overall, the aim of this first chapter was to introduce the reader to the methods involved in estimating Learning DSGE models, and show how well it could improve the fit compared to the benchmark rational expectations model. Both DSGE and Learning DSGE literatures are very recent and empirical applications in those areas are growing fast. We consider the learning approach a flexible and interesting alternative to rational expectations, which can be added to any DSGE model. In the next chapters, we will introduce alternative, more flexible methods of estimating learning models, gradually breaking the link between Learning and DSGE models.

[44] In fact, most of the improvement of posterior/likelihood in Slobodyan and Wouters (2009) comes from optimised beliefs. We pursued a different path here, using only rational expectations consistent beliefs.

2 Limitations and additional methods of learning: Learning DSGE-VAR and Learning MSV models

As we saw in the previous chapter, we were able to improve the fit of estimated DSGE models when we introduce a learning mechanism. This happened both when we used simulated data for models 1 and 2 and as well as when we used real data for Brazil for models 2 and 3. In this chapter we explain more formally some important limitations of the learning mechanism in a DSGE model, and suggest two ways of overcoming or mitigating them.

The main contributions of this chapter are the algebraic understanding of the limitations of learning and the proposal of two flexible methods to deal with the parameter instability in the data. The first of these methods is closely related to the DSGE-VAR methodology, which we call Learning DSGE-VAR, or LDSGE-VAR, and the second, which almost completely breaks the link with the DSGE model, which we call Learning Minimum State Variable, or LMSV. We provide MATLAB codes for these methods, which are integrated into our main LDSGE code used and discussed in the preceding chapter.

This chapter is organized as follows: the next section presents the limitations of the learning approach in a numerical example from our univariate model. Section 2 presents in detail the two methods above mentioned which deal with these limitations. Section 3 presents some estimation setups and estimates some of the models presented in chapter 1 using the these two methods. We estimate models 1 and 2 using our simulated data and model 3 using real data. Section 4 summarizes our main findings, and section 5 concludes.

2.1 Introduction and Motivation

There is an algebraic reason why learning can improve only to a certain extent the fit of DSGE models. For simplicity, here we give a numerical example from our univariate model. Suppose the economy is given by:

$$y_t = \alpha y_{t-1} + \beta E_t y_{t+1} + \varepsilon_t \tag{2.1}$$

with ε_t being an *iid* disturbance. Solving the model under rational expectations will give us a law of motion of the type:

$$y_t = \phi_1 y_{t-1} + \phi_2 \epsilon_t \tag{2.2}$$

Suppose that α and β are respectively 0.5 and 0.3. The rational expectations solution of the model is given by $\phi_1 = 0.61$ and $\phi_2 = 1.23$.[45] Since this is the Actual Law of Motion of the economy, any regression made by agents in their Perceived Law of Motion will produce (estimate) reduced-form parameters close to that law. Suppose the Perceived Law of Motion is given by:

$$y_t = b_t y_{t-1} \tag{2.3}$$

where b_t evolve over time (and will fluctuate around 0.61, since errors of this regression are supposed to be *iid*). The time t learning solution to this model is given by:

$$\begin{aligned} y_t &= \alpha y_{t-1} + \beta b_t y_t + \varepsilon_t & (2.4)\\ (1-\beta b_t) y_t &= \alpha y_{t-1} + \varepsilon_t & (2.5)\\ y_t &= \frac{\alpha}{1-\beta b_t} y_{t-1} & (2.6) \end{aligned}$$

Suppose that $b_t = 0.61$ and $\beta = 0.3$. For (2.6) to match (2.2) it is necessary that $\alpha = 0.5$, which (as expected) is consistent with the proposed RE solution.

Now we repeat this procedure in a time-varying framework. Suppose that, for a subset of our sample (the half of the entire sample, for example), we have $\alpha = 0.6$, which is consistent with $\phi_1 = 0.78$. Now suppose that there are 3 possible scenarios when estimating this model with learning for the full sample: (i) $\alpha = 0.5$, $\beta = 0.3$; (ii) $\alpha = 0.6$,

[45] Both ϕ_1 and ϕ_2 are the solution to the Lyapunov equation which gives us the solution of (2.1). Generally, if agents' Perceived Law of Motion is given by $y_t = by_{t-1} + c\xi_t$, inserting this into (2.1) would produce:

$$\begin{aligned} y_t &= \alpha y_{t-1} + \beta b y_t + \varepsilon_t \\ (1-\beta b) y_t &= \alpha y_{t-1} + \varepsilon_t \\ y_t &= \frac{\alpha}{1-\beta b} y_{t-1} + \frac{1}{1-\beta b} \varepsilon_t \end{aligned}$$

For the Perceived Law of Motion to be valid it is necessary that $\frac{\alpha}{1-\beta b} = b$ and $\frac{1}{1-\beta b} = c$, which can be solved for b and c given numerical values for α and β. For $\alpha = 0.5$ and $\beta = 0.3$, we have $b - 0.3b^2 = 0.5$, with roots $b_1 = 2.72$ (unstable) and $b_2 = 0.61$. Solving for $\frac{1}{1-\beta b} = c$ gives us $c = 1.23$. Now set $\phi_1 = b$ and $\phi_2 = c$.

$\beta = 0.3$; and finally (iii) $\alpha = 0.55$, $\beta = 0.3$.[46]

Scenarios (i) to (iii) will only match the time-varying Actual Law of Motion of the economy ($\phi_1 = 0.61$ for the fist half and $\phi_1 = 0.78$ for the second half) if b_t varies over time. Scenario (i) would require that b_t fluctuates around 0.61 at the first half of the sample and then around 1.2 at the second half; scenario (ii) would require that b_t fluctuates around 0.06 for the first half and then around 0.78 at the second half. The third scenario would require that b_t fluctuates around 0.34 at the first half and around 0.99 at the second half.

It is clear from this simple numerical example that there could be large forecast errors in scenarios (i) and (iii), since agents' Perceived Law of Motion (in this simple case, expressed by b_t) is only capable of fluctuating around 0.61 for the first half and around 0.78 for the second half of the observations (since we generated data this way and errors are *iid*)

It is also clear that, even though our learning model is not able to replicate the Actual Law of Motion of the economy, it allows for some flexibility and may still do a better job than its rational expectations counterpart. This is precisely what happened in our simulated problems with models 1 and 2.

The possibility of not being able to replicate the Actual Law of Motion is our motivation to develop a more flexible strategy to deal with these problems in data. We present two methods to do so. The first method, called Learning DSGE-VAR, still maintain some connection with the DSGE model at hand, while the Learning MSV almost completely breaks this link and estimates a fully flexible model (as will be clearer soon). We present them now.

[46]These scenarios suppose that when α jumps from 0.5 to 0.6, its estimator $\hat{\alpha}$ for the full sample will be somewhere between these two numbers. We consider a "partial equilibrium" case in which β does not adjust to match the data, for the sake of exposition. When we applied our LDSGE to the simulated data, however, both $\hat{\alpha}$ and $\hat{\beta}$ adjusted to fit the data (the best fitting specification estimated $\hat{\alpha} = 0.49, \hat{\beta} = 0.43$), which will allow for more adjustment/time variation. In the best specification case, the parameter ϕ_1 can fluctuate around 0.65 in the first half to 0.72 in the second half of observations. This flexibility is quite large here, since there are no cross equation restrictions forcing the estimation of the time invariant parameter β close to its true value, and we are dealing with an univariate case. If, on the other hand, our parameter β is calibrated (or identified throught another equation), the estimator $\hat{\alpha}$ will produce the adjustment to maximize fit and will most likely be estimated between 0.5 and 0.6.

2.2 LDSGE-VAR and Learning MSV approaches

2.2.1 DSGE-VAR and Learning DSGE-VAR

Simply put, the solution of a DSGE model is a VAR added with unobserved variables and cross-equation restrictions. These cross-equation restrictions come from the solution of maximization problems that depends on (at best) policy-invariant structural parameters. As we discussed in chapter 1, over time macroeconomists realized that a rich set of frictions was needed to produce respectable DSGE models in terms of forecasts.

The DSGE-VAR methodology emerged as a bridge between the theoretical elegance of DSGE models and the empirical flexibility of VARs. As Del Negro and Schorfheide (2003) argue, the DSGE-VAR take the DSGE model restrictions seriously without imposing them dogmatically. It is indeed a recognition that these two models are limited tools that could produce better results working together.

The DSGE-VAR literature started with DeJong, Ingram and Whiteman (1993) and Ingram and Whiteman (1994), who showed that DSGE priors improved the forecasting performance of VAR models from one to four quarters ahead, in comparison with a pure random walk forecast.

Schorfheide (2004) then extended their research on the Bayesian front, showing how to translate inference from VAR to DSGE parameters, enabling them to construct impulse response functions. Del Negro & Schorfheide (2004) estimated a DSGE-VAR using a variation of the Smets & Wouters (2003) model for the Euro Area.

We will briefly review the DSGE-VAR calculations, which can be found in more details in Del Negro and Schorfheide (2004). We start from an economy given by a (linearized) DSGE model. Its solution is given by a state-space system:

$$y_t = \mu_y + H'S_t + w_t \tag{2.7}$$

$$S_t = TS_{t-1} + R\varepsilon_t \tag{2.8}$$

$$\varepsilon_t \sim N(0, \Sigma_\varepsilon) \tag{2.9}$$

$$w_t \sim N(0, \Sigma_w) \tag{2.10}$$

where (2.7) and (2.8) are, respectively, the measurement and the transition equations. y_t is a $N_{obsy} \times 1$ vector of observable variables, while S_t is a vector of states. w_t and ε_t are innovations. This state-space is then approximated by a VAR of order p:

$$y_t = \phi_0 + \sum_{j=1}^{p} \phi_j y_{t-j} + \xi_t \tag{2.11}$$

$$\xi_t \sim N(0, \Sigma_\xi) \tag{2.12}$$

where ϕ_0 and ϕ_j are matrices of coefficients and ξ_t is a vector of innovations. This VAR can then be rewritten in matrix form as:

$$y_t = \Phi X_t + \xi_t \tag{2.13}$$

$$\xi_t \sim N(0, \Sigma_\xi) \tag{2.14}$$

with

$$X_t = \begin{bmatrix} 1 & y'_{t-1} & y'_{t-2} & \ldots & y'_{t-p} \end{bmatrix} \tag{2.15}$$

$$\Phi = \begin{bmatrix} \phi_0 & \phi_1 & \phi_2 & \ldots & \phi_p \end{bmatrix} \tag{2.16}$$

The mapping from the DSGE(θ) model to the VAR(Φ) is given by the population-based regression of y_t on X_t. To do so, we define some auxiliary moment / covariance matrices:

$$\mu_y = E(y_t \mid \theta) \tag{2.17}$$

$$\Sigma_y^0 = E(y_t y'_t \mid \theta) = H'\Sigma_S H + \Sigma_w \tag{2.18}$$

$$\Sigma_y^j = E(y_t y'_{t-j} \mid \theta) = H' T^j H \text{ , for } j > 0 \tag{2.19}$$

$$\Sigma_S = T\Sigma_S T' + RR' \tag{2.20}$$

$$\begin{aligned}
\Gamma_{XX} &= E(X_t X_t' \mid \theta) && (2.21)\\
\Gamma_{yX} &= E(y_t X_t' \mid \theta) && (2.22)\\
\Gamma_{yy} &= E(y_t y_t' \mid \theta) = \Sigma_y^0 + \mu_y \mu_y' && (2.23)
\end{aligned}$$

The first equation is related to the first moment (mean) of y_t; equations (2.18) and (2.19) are the autocovariance functions of y_t; equation (2.20) is the second moment matrix of the endogenous variables S_t. The last three equations are the population covariance matrices of, respectively, $X_t X_t'$, $y_t X_t'$ and $y_t y_t'$.

Note that all moments are conditional on the structural parameter vector θ, which means that the moments will be a function of θ. The population regression of y_t on X_t will give us the reduced-form parameters mean and variance, which are also function of the DSGE model structural parameters:

$$\begin{aligned}
\Phi(\theta) &= \Gamma_{yX}\Gamma_{XX}^{-1} && (2.24)\\
\Sigma_\xi(\theta) &= \Gamma_{yy} - \Gamma_{yX}\Gamma_{yy}^{-1}\Gamma_{yX} && (2.25)
\end{aligned}$$

Both Φ and Σ_ξ are restriction functions that will be used to center the prior distribution of the VAR, conditional on θ and on an hyperparameter λ that measures the tightness of the DSGE restrictions over the DSGE-VAR priors. The priors of the DSGE-VAR(θ,λ) are given by:

$$\begin{aligned}
vec(\Phi \mid \theta,\lambda) &\sim N_{n(np+1)}\left(vec(\Phi(\theta)), (\lambda T)^{-1}\left[\Gamma_{XX}^{-1} \otimes \Sigma_\xi(\theta)\right]\right) && (2.26)\\
(\Sigma_\xi \mid \theta,\lambda) &\sim IW\left(\lambda T \Sigma_\xi(\theta), \lambda T - (np+1)\right) && (2.27)
\end{aligned}$$

which basically tells us that the prior for the mean, given a vector of structural parameters θ and a tightness parameter λ follows a multivariate normal distribution, with mean $vec(\Phi(\theta))$ and variance $(\lambda T)^{-1}\left[\Gamma_{XX}^{-1} \otimes \Sigma_\xi(\theta)\right]$. The prior for the variance, given θ and λ, is an Inverse Wishart (which is proper if $\lambda > (n(np+1))/T$).

It is also important to notice that, when $\lambda \to 0$, the covariance matrix of our mean parameters become infinite, which means that our priors are uninformative (therefore our final model will have almost nothing from the DSGE). When $\lambda \to \infty$, the DSGE will act as a very restrictive prior, such that the estimated model will be very close to the DSGE model itself.[47] Another interpretation of the tightness hyperparameter λ, as Christoffel (2013) argues, can be of augmenting our VAR model with λT virtual observations from the DSGE model.

The estimation or calibration of λ is a separate discussion in the DSGE-VAR literature. As a matter of fact, part of the success of the DSGE-VAR approach derived from showing that estimated lambdas $(\widehat{\lambda})$ were different from zero (or from its minimium value, such that the Inverse Wishart distribution mentioned above is proper), meaning that DSGE priors coming from simple DSGE models could improve the forecasting accuracy of the model at hand.

When comparing in-sample fit, however, it should be clear that lambda, if estimated, will be pushed towards zero, since VAR parameters are estimated such that in-sample fit is maximized. This is true for a VAR and is particularly more important in a time-varying framework[48] such as ours: since parameters are allowed to vary over time, the best fitting model will be one that penalizes time-variation the least, which means an estimated lambda close to zero. Our discussion will therefore be focused on the forecast improvement of the models for different values of the gain parameter, as will be seen below.

The first methodological contribution of this chapter is to develop a learning counterpart to the DSGE-VAR methodology, which we call Learning DSGE-VAR or LDSGE-VAR. It uses the main idea of the DSGE-VAR approach, which is to use a DSGE model to form priors to a (Bayesian) VAR describing the economy. However, the reduced-form parameters of the VAR are now allowed to vary over time, while still linked to the DSGE model through time.

[47] More precisely, the resulting model will become very close to the VAR approximation of the DSGE model, given the structural parameters.

[48] We will discuss this issue in more detail in the next section.

The way we build our learning mechanism allows the time-varying VAR to still maintain some connection to the DSGE model. The literature so far has worked either with flexible, time-varying Bayesian VARs such as Amisano & Serati (2004) and Primiceri (2005), with no connection at all to a DSGE model, or with time-invariant (B)VARs, with priors (if Bayesian) coming from a DSGE model, such as in Christoffel (2013) and Del Negro & Schorfheide (2006).

As already mentioned, our contribution is to help filling the gap between time-varying VARs and DSGE models, and we do so by estimating a Bayesian VAR with reduced-form parameters evolving over time in a learning framework. Priors to this Bayesian VAR come from a rational expectations DSGE model. The DSGE-VAR economy, which was given by (2.13) and (2.14), now is given by:

$$y_t = \Phi_t^L X_t + \xi_t \tag{2.28}$$

$$\xi_t \sim N(0, \Sigma_\xi) \tag{2.29}$$

where y_t is a vector of observable variables, Φ_t^L is a matrix of reduced-form coefficients, X_t is a matrix of regressors (which, since this is a VAR, are the endogenous variables lagged until p periods); ξ_t are disturbances, which we assume follow a Normal distribution with mean zero and covariance represented by Σ_ξ. The priors for the reduced-form coefficients come from the rational expectations DSGE(θ) model:

$$vec(\Phi_t^L \mid \theta, \lambda) \sim N_{n(np+1)}\left(vec(\Phi(\theta)), (\lambda T)^{-1}\left[\Gamma_{XX}^{-1} \otimes \Sigma_\xi(\theta)\right]\right) \tag{2.30}$$

$$(\Sigma_\xi \mid \theta, \lambda) \sim IW\left(\lambda T \Sigma_\xi(\theta), \lambda T - (np+1)\right) \tag{2.31}$$

which, as in the traditional DSGE-VAR methodology, states that mean coefficients have a prior given by a time-invariant normal distribution with mean $vec(\Phi(\theta))$ and also time-invariant variance $(\lambda T)^{-1}\left[\Gamma_{XX}^{-1} \otimes \Sigma_\xi(\theta)\right]$. The variance coefficients have a prior given by a time-invariant Inverse Wishart, just as before.

The reduced-form coefficients Φ_t^L, however, now are allowed to evolve over time, following a law of motion given by:

$$\Phi_t^{L'} = \Phi_{t-1}^{L'} + g_t R_t^{-1} X_{t-1}(y_{t-1} - \Phi_{t-1}^{L} X_{t-1}) \tag{2.32}$$

$$R_t = R_{t-1} + g_t(X_{t-1}X_{t-1}' - R_{t-1}), \tag{2.33}$$

where g_t is a sequence of gains ($g_t = g$ gives us a constant gain learning algorithm, while $g_t = g_t R_t$ gives us the stochastic gradient learning algorithm), and R_t is the second-moment matrix of the regressors (X_t). It is important to notice that if the gain parameter is set to zero, $\Phi_t = \Phi_{t-1} = \Phi$ and $R_t = R_{t-1} = R$. Our learning approach, therefore, encompasses the traditional DSGE-VAR as a special case where $g_t = 0$.

This is very similar to what we did in the first chapter, except for the priors. In the learning DSGE models of chapter 1, the priors were defined over the structural time-invariant parameters, so its numerical value could be calculated at any moment.[49] In contrast, in the LDSGE-VAR approach, priors are defined over the reduced time-varying parameters, and must be calculated at every observation. This is how the time-varying structure maintains its connection to the DSGE model. If the reduced-form parameters are driven too far away from the rational expectations induced by the DSGE model in order to improve the likelihood (and therefore the posterior), the prior function will penalize these deviations from the DSGE model.

There are two main objectives for using a LDSGE-VAR model. First, it is an alternative way of describing our data, that takes into account how DSGE models captures cross-correlations in real data and deals with problems such as model mispecification, time-variation, structural breaks, etc. This very same motivation was present in the development of the traditional DSGE-VAR methodology. The contributions of adding learning to that is that it encompasses the traditional DSGE-VAR as a special case, while still maintaining a link with the DSGE model restrictions, and it also allows for a more flexible environment which can improve forecasts further.

Second, our LDSGE-VAR model can be used as a detection mechanism for changes in the economy (understood by instability in the parameters governing the law of motion of

[49] Our code calculates the prior before the likelihood for the sake of saving processing time, given that out-of-bounds parameter draws have zero probability density function (pdf) and therefore negative infinite log pdf.

the economy), whose flexibility to adjust to time-varying laws of motion of the economy allows it to learn from new data as soon as it becomes available. Since we can calibrate our tightness hiperparameter λ to put little weight on the DSGE restrictions (which means a flat prior), our reduced-form parameters can move more freely, adjusting to the problems in data.

When formulating this methodology, two important questions emerged: (i) why not build priors using a learning DSGE model? and (ii) why priors are calculated for every observation and what is their role, since parameters could now vary freely over time?

The answer to (i) is not simple. In theory, we could form priors using a LDSGE, but the result would be that both our priors and LDSGE-VAR would lose some of its intuition: since our LDSGE produces a time-varying actual law of motion, we would end up with time-varying restriction functions (2.26) and (2.27). That means a different prior distribution for each observation, for a given vector of parameters, which doesn't make much sense as a prior in a Bayesian framework.

In practical terms, using time-varying priors would most certainly increase the fit of our model, since some of the job of describing the data would have already been done by the learning/time-varying structure of the DSGE model. Our approach is different. We will depart from a rational expectations DSGE model and leave the task of describing/forecasting the (potentially with time-varying parameters) data only to the LDSGE-VAR.

That leads us to our second question, (ii). In "regular" Bayesian econometrics, priors are calculated only once, and then added to the likelihood of the model. This could be a problem here, since theoretically there is nothing preventing our means (which are updated over time by our learning mechanism) from diverging or escaping from the bounds of their prior distributions.

To prevent that, we calculate our prior at each point in time, dividing the result by T (the number of periods) and adding them up to calculate the value of the prior. Calculated this way, our priors coming from the DSGE model will penalize large gain parameters that cause means to diverge and prevent them from escaping their bounds (or fluctuate over small probability areas defined by the DSGE model). Priors are, in essence, the memory of the DSGE model restrictions.

In order to estimate a LDSGE-VAR, we have to choose some important settings according to the question we are trying to answer. First, we have to decide which parameters we want to estimate (besides the ones contained in the reduced-form VAR). The standard DSGE-VAR(λ, θ, p)[50] allows us to estimate λ, p and θ jointly or not. We can, for instance, use the rational expectations DSGE model to estimate θ and use it as fixed for the rest of the exercise, or we can estimate it simultaneously with the other parameters.

One might not be necessarily interested in the best fitting DSGE-VAR model, but instead in the sensitivity of the likelihood/posterior to (i) λ, (ii) the gain parameter g and (iii) the relation between the estimated gain parameter and λ. Del Negro & Schorfheide (2006), for instance, computed a 3d surface plotting the marginal likelihood of the DSGE-VAR for different values of λ and the time span of their rolling regression,[51] in order to assess how important / strong are the restrictions coming from DSGE models over time.

Second, as previously mentioned, our VAR model is an approximation to the DSGE model, hence we need to specify the order of the VAR. Our MATLAB code is able to perform the estimation of VAR's up to lag $p_{\max} = 4$, which is the order chose because we are dealing with quarterly data.

Third, one has to choose a metric to assess the performance of the DSGE-VAR against the DSGE model. The DSGE-VAR is, by construction, more flexible in dealing with in-sample data, therefore the estimated λ will be driven to zero if the DSGE restrictions are too tight, which will be agravated if the Actual Law of Motion of the economy is time varying (as it will be the case in our simulated data). Our measure of performance will be the mean squarred forecasts errors from one to four periods ahead, which our MATLAB code is designed to compute.

Fourth, our learning mechanism model adds another important parameter to the estimation, the *gain* parameter g. Therefore our LDSGE-VAR is indexed by LDSGE-

[50] λ is the hyperparameter that measure the tightness of the DSGE model restrictions over the prior of the (L)DSGE-VAR; θ is the vector of structural parameters from the DSGE model; p in the order of the VAR which will approximate the DSGE model.

[51] They estimated the sensitivity of the likelihood to λ over rolling window regressions, therefore creating a 3d surface.

VAR(g, θ, λ, p). This adds another dimension to the first consideration (on which parameters should be estimated). One might also be interested in assessing the sensitivity of the gain parameter to the DSGE prior tightness λ, given θ. Since the hyperparameter λ only affects the (prior) variance of the DSGE-VAR reduced-form estimators, a larger λ should produce a smaller estimated gain, as seen in equations (2.30) and (2.31).

2.2.2 Learning MSV

The second method of this chapter, in contrast with the LDSGE and the LDSGE-VAR, almost completely breaks the link with the DSGE model parameters, aiming to estimate a fully flexible time-varying learning model compatible with the solution induced by the DSGE model. We call this method Learning MSV (Learning Minimum State Variable).

The main idea is to assess how much improvement in the marginal likelihood our LDSGE could have achieved if the limitations (as seen in the beginning of this chapter, in the numerical example of the univariate model) of the LDSGE methodology were not present.

Before doing so, we take a step back and present another restriction of the LDSGE framework. This restriction (rather than limitation) of the LDSGE methodology is that the Actual Law of Motion of the economy is the combination of the DSGE model solution and the Perceived Law of Motion. This means that the impact of time variation of the Perceived Law of Motion will be different across equations. For example, consider the following bivariate model:

$$\begin{vmatrix} X_{1,t} \\ X_{2,t} \end{vmatrix} = \begin{vmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \end{vmatrix} \begin{vmatrix} X_{1,t-1} \\ X_{2,t-1} \end{vmatrix} + \begin{vmatrix} \beta_{11} & 0 \\ 0 & 0 \end{vmatrix} \begin{vmatrix} X_{1,t+1} \\ X_{2,t+1} \end{vmatrix} + \begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix} \begin{vmatrix} \xi_{1,t} \\ \xi_{2,t} \end{vmatrix} \tag{2.34}$$

with the following Perceived Law of Motion (considered here as time t-1 for the sake of exposition):

$$\begin{vmatrix} X_{1,t+1} \\ X_{2,t+1} \end{vmatrix} = \begin{vmatrix} b_{11,t} & b_{12,t} \\ b_{21,t} & b_{22,t} \end{vmatrix} \begin{vmatrix} X_{1,t-1} \\ X_{2,t-1} \end{vmatrix} \tag{2.35}$$

Inserting the Perceived Law of Motion back into the model, we get:

$$\begin{vmatrix} X_{1,t} \\ X_{2,t} \end{vmatrix} = \begin{vmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \end{vmatrix} \begin{vmatrix} X_{1,t-1} \\ X_{2,t-1} \end{vmatrix} + \begin{vmatrix} \beta_{11} & 0 \\ 0 & 0 \end{vmatrix} \begin{vmatrix} b_{11,t} & b_{12,t} \\ b_{21,t} & b_{22,t} \end{vmatrix} \begin{vmatrix} X_{1,t-1} \\ X_{2,t-1} \end{vmatrix} + \begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix} \begin{vmatrix} \xi_{1,t} \\ \xi_{2,t} \end{vmatrix} \tag{2.36}$$

rearranging:

$$\begin{vmatrix} X_{1,t} \\ X_{2,t} \end{vmatrix} = \begin{vmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \end{vmatrix} \begin{vmatrix} X_{1,t-1} \\ X_{2,t-1} \end{vmatrix} + \begin{vmatrix} \beta_{11} b_{11,t} & \beta_{11} b_{12,t} \\ 0 & 0 \end{vmatrix} \begin{vmatrix} b_{11,t} & b_{12,t} \\ b_{21,t} & b_{22,t} \end{vmatrix} \begin{vmatrix} X_{1,t-1} \\ X_{2,t-1} \end{vmatrix} + \begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix} \begin{vmatrix} \xi_{1,t} \\ \xi_{2,t} \end{vmatrix} \tag{2.37}$$

and finally:

$$\begin{vmatrix} X_{1,t} \\ X_{2,t} \end{vmatrix} = \begin{vmatrix} \alpha_{11} + \beta_{11} b_{11,t} & \alpha_{12} + \beta_{11} b_{12,t} \\ \alpha_{21} & \alpha_{22} \end{vmatrix} \begin{vmatrix} X_{1,t-1} \\ X_{2,t-1} \end{vmatrix} + \begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix} \begin{vmatrix} \xi_{1,t} \\ \xi_{2,t} \end{vmatrix} \tag{2.38}$$

Note that the DSGE restrictions (equation for X_2 not having an expectations term) did not allow the final model dynamics (the Actual Law of Motion) to be fully time varying. This kind of restriction is model specific and we will not deal with that here. It should be clear, however, that the learning mechanism does not necessarily have an effect on all equations of the model: it will depend on how important expectations matter. Now we can go back to our LMSV methodology.

We suppose the economy is now given by the solution of the DSGE model, expressed in a state-space system:

$$y_t = H' S_t + w_t \tag{2.39}$$

$$S_t = T_t^{MSV} S_{t-1} + R\varepsilon_t \tag{2.40}$$

$$\varepsilon_t \sim N(0, \Sigma_\varepsilon) \tag{2.41}$$

$$w_t \sim N(0, \Sigma_w) \tag{2.42}$$

where (2.39) is the measurement equation, which links the vector of observable variables y_t to the states S_t through the selection matrix H; (2.40) is the transition equation; ε_t and w_t are *iid* innovations.

We now allow the transtition matrix T_t^{MSV} to evolve over time through a learning mechanism, which is given by:

$$
\begin{aligned}
T_t^{MSV\prime} &= T_{t-1}^{MSV\prime} + g_t(R_t^T)^{-1}S_{t-2}(S_{t-1} - T_{t-1}^{MSV}S_{t-2})' && (2.43) \\
R_t^{MSV} &= R_{t-1}^{MSV} + g_t(S_{t-2}S_{t-2}' - R_{t-1}^{MSV}), && (2.44)
\end{aligned}
$$

where g_t is a sequence of gains and R_t^{MSV} is the second moment matrix of the state variables S_t.

Note that the way information is used here is very similar to the MSV time t learning Perceived Law of Motion (as explained when we introduced different Perceived Laws of Motion in the first chapter): agents know the structure of the economy (number of variables, including non observables) but do not know the parameters. Forecast of time t variables uses information up to time $t-1$ (hence time t information set). Now, instead of inserting the Perceived Law of Motion into agents' expectations to achieve an Actual Law of Motion, we work directly with the Actual Law of Motion, allowing it to vary over time.

The initial conditions T_0^{MSV} and R_0^{MSV} could come, as in the LDSGE model, from optimised initial conditions, fixed at some vector, or from a DSGE or LDSGE models.[52] Note, however, that these initial conditions are not essential here to determine the likelihood of the system (unlike in the LDSGE model), since T_t^{MSV} and R_t^{MSV} are allowed to vary freely over time. Hence, any improvement coming from the initial observations would be limited and will not be focused here.

Our main goal is to compare this marginal likelihood to the ones produced under LDSGE. Our MATLAB code reads the parameter vector from the model m-file and performs a sensitivity analisys of the marginal likelihood to the gain parameter under our

[52] We find more accurate to call T_0^{MSV} and R_0^T initial conditions rather than initial beliefs, since beliefs are related to agents' expectations.

new flexible method.

In order to compare on equal ground results from the LDSGE model and the LMSV model, we only use two specifications from the LDSGE model. These specifications have the same information set than LMSV, therefore we are only comparing how this more flexible method is able to improve fit. We use the following specification: MSV learning, without constants, time t Perceived Law of Motion. We will end up with two specifications (one that uses stochastic gradient and other that uses constant gain as the learning algorithm). We chose the one that produces the best marginal likelihood and use it as our parameter vector in the sensitivity analysis.[53]

2.3 Estimation setup

We aim to test these two new methods using our artificial data of chapter 1, which contains a structural break. We will perform three estimations for each of the two methods mentioned earlier in this chapter (LDSGE-VAR and LMSV): one for each artificial data, from models 1 and 2, and one for the real data for model 3. Reminding us about the models, model 1 is an univariate model given by:

$$y_t = \alpha y_{t-1} + \beta E_t y_{t+1} + Z_t \tag{2.45}$$

$$Z_t = \rho Z_{t-1} + \varepsilon_t \tag{2.46}$$

where y_t is an endogenous variable and we have the first half of data simulated using $\alpha = 0.5$, $\beta = 0.3$ and $\rho = 0$ and the second using $\alpha = 0.6$, $\beta = 0.3$ and $\rho = 0$ Since ρ is equal to zero, a VAR(1) model will fully capture the dynamics of the model solution (which, under rational expectations, is given by $y_t = by_{t-1} + c\varepsilon_t$).

Model 2 is the already presented 3-equation New Keynesian model given by:

[53] Our MATLAB code also allows for the estimation of optimal initial conditions, but we do not perform it here, since we want to compare only marginal likelihoods of methods that differ on their updating scheme, not on initial conditions. Allowing for optimal initial conditions would likely improve the marginal likelihood further. This improvement would come mostly in the initial data points of the Kalman Filter.

$$cpi_t = \beta E_t cpi_{t+1} + \kappa gdp_t + \varepsilon_{1t} \quad (2.47)$$
$$gdp_t = E_t gdp_{t+1} - \sigma ff_t + \sigma E_t cpi_{t+1} + \varepsilon_{2t} \quad (2.48)$$
$$ff_t = \rho_i ff_{t-1} + \phi_1 E_t cpi_{t+1} + \phi_2 gdp_t + \varepsilon_{3t} \quad (2.49)$$
$$\varepsilon_{1t} = \rho_1 \varepsilon_{1t-1} + \xi_{1t} \quad (2.50)$$
$$\varepsilon_{2t} = \rho_2 \varepsilon_{2t-1} + \xi_{2t} \quad (2.51)$$
$$\varepsilon_{3t} = \rho_3 \varepsilon_{3t-1} + \xi_{3t} \quad (2.52)$$

where, as before, cpi stands for inflation, gdp for output gap and ff for the nominal interest rates. The first half of data was simulated using $\beta = 0.99, \kappa = 0.4$, $\sigma = 0.5$, $\phi_1 = 0.7$, $\phi_2 = 0.7$, $\rho_i = 0.5$, $\rho_1 = \rho_2 = 0.5$, $\rho_3 = 0.3$, $\sigma_{\xi_1} = 0.5$,$\sigma_{\xi_2} = 0.5$, $\sigma_{\xi_3} = 0.5$ and the second half we changed $\sigma = 0.3$. Model 3 is the simpler version of Smets & Wouters (2003) model already described in chapter 1.

We take the following steps to estimate each model:

LDSGE-VAR(g, θ, λ, p) models

(1) Estimate a rational expectations DSGE model and find its posterior maximum ($\theta^{\max}$);

(2) Calculate $\Phi(\theta^{\max})$ and $\Sigma_{\xi}(\theta^{\max})$, which are our prior functions, given λ;[54]

(3) Choose a value $\bar{p}$ for the order of our VAR approximation of the DSGE;

(4) Build grids for λ and g;

(5) Estimate the likelihood of the LDSGE-VAR($g, \theta^{\max}, \bar{p}, \lambda$), for the space given by **(4)**;

(6) Plot the posterior/likelihood curves for the grids mentioned above;

(7) Estimate the forecast improvements for our artificial data.

LMSV models

(1) Use the parameters of the best LDSGE specifications (using MSV Perceived Law of Motion learning, with time t Perceived Law of Motion) to generate initial conditions for T_0^{MSV} and R_0^{MSV};

[54] In the calculation of our priors, we only considered mean parameters in the evaluation of the prior, as they are the only ones with a structural break in our sample. We proceeded in good faith, assuming the result should not affect our results by a large extent.

(2) Plot marginal likelihoods for a grid of gains using initial conditions provided in (1) and compare them with the best marginal likelihood achieved under LDSGE;

(3) Specifically for model 1, we plot the time evolution of the Actual Law of Motion of the economy (specifically, the parameter related to y_{t-1}, which is supposed to vary from 0.61 to 0.78) for the LMSV and compare it with the evolution implied by the LDSGE model.

2.4 Main results

We present first the LDSGE-VAR results.[55] We are interested in assessing if the learning mechanism can improve the fit comparatively its non learning DSGE-VAR. We will plot likelihood differentials for all models, and forecasts improvements from one to four periods for our artificially generated data.

The figure below illustrates the sensitivity of the gain parameter g to the DSGE tightness parameter λ for the univariate model. As DSGE restrictions become lower (λ approximates zero), the learning mechanism is able to improve the likelihood of the LDSGE-VAR, given (θ, λ, p), by a larger extent. This is indicated by the "negative" likelihood improvements (which means better) for the sets of λ and gains considered here.

It is important to reinforce that this graph is comparing the likelihood of LDSGE-VAR models to their LDSGE-VAR zero gain counterparts (which is a DSGE-VAR), in order to assess the sensitivity of the learning gain parameter to the tightness of the DSGE priors. Our goal is not to compare the fit of the LDSGE-VAR to the rational expectations DSGE model but, for the sake of exposition, we plot the difference between the marginal likelihood of the LDSGE-VAR and the rational expectations specification for model 1 in figure (2.2). The middle (blue) area represents all the LDSGE-VAR$(g, \theta^{RE}_{MAX}, \lambda, p = 1)$ specifications that produce a better marginal likelihood than the rational expectations DSGE model. As can be seen, the learning mechanism is not able to compensate for excessive values of tightness from the DSGE model, hence the overall shape of these differences is convex, producing an unique maximum in the (λ, p) space.

[55] We fixed $y - axis$ to represent likelihood deltas, and switched x and $y - axis$ when necessary in order to produce a clearer figure.

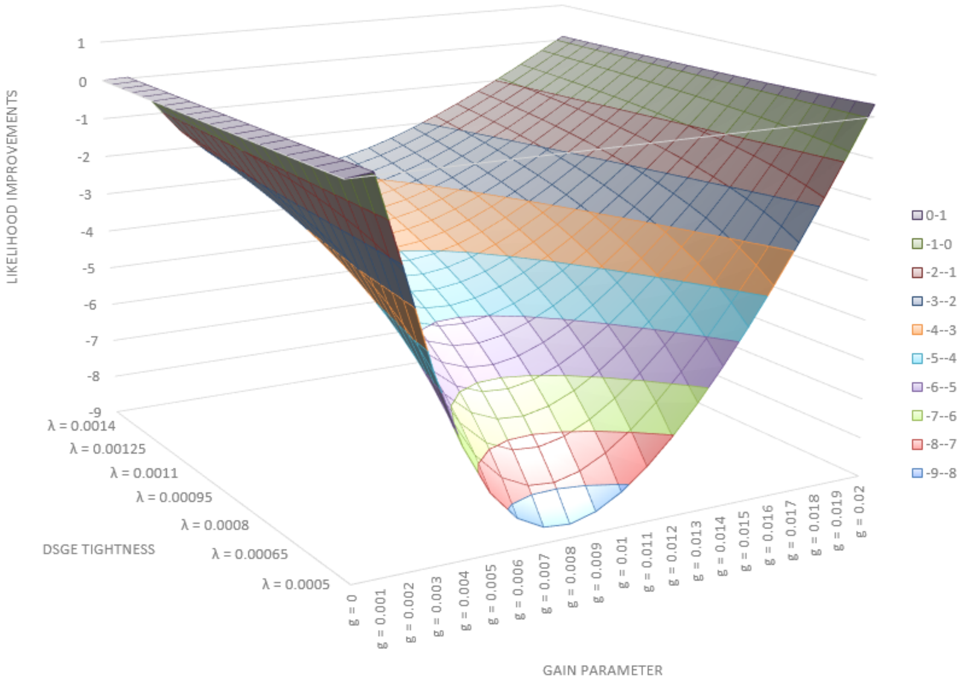

Figure 2.1: Likelihood improvements of LDSGE-VAR relative to the zero gain LDSGE-VAR, model 1

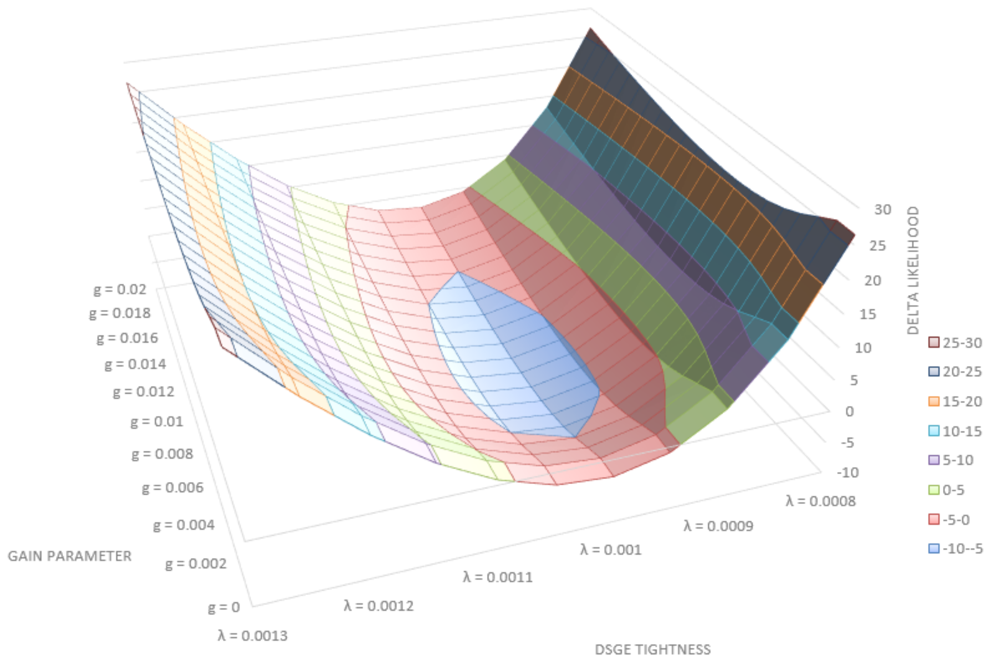

Figure 2.2: Likelihood improvements of LDSGE-VAR relative to the rational expectations DSGE model, model 1

Results for the basic New Keynesian model (model 2) and the simpler version of Smets & Wouters (model 3) followed the results presented under LMSV (which will be showed below). For model 2, the excessive volatility of parameters lead to a worse performance of fit, and no evidence of learning was found. The fit of the LDSGE-VAR was equal or worse for all combinations of tightness and gain parameters considered. Differently from the univariate model, we estimated a second order VAR to approximate the dynamics to the rational expectations DSGE model, since there is some persistence in the simulated shocks. The likelihood improvements of both models are presented below. The likelihood improvements for the Smets & Wouters model was close to the LMSV model: around 2 points. Important to note that, as the tightness of the DSGE restrictions increase (for model 3), the gain space that allows the likelihood to improve (relatively to the zero gain LDSGE-VAR) becomes larger, which means that the learning mechanism is able to (partially) compensate for the increase in tightness of the DSGE model.

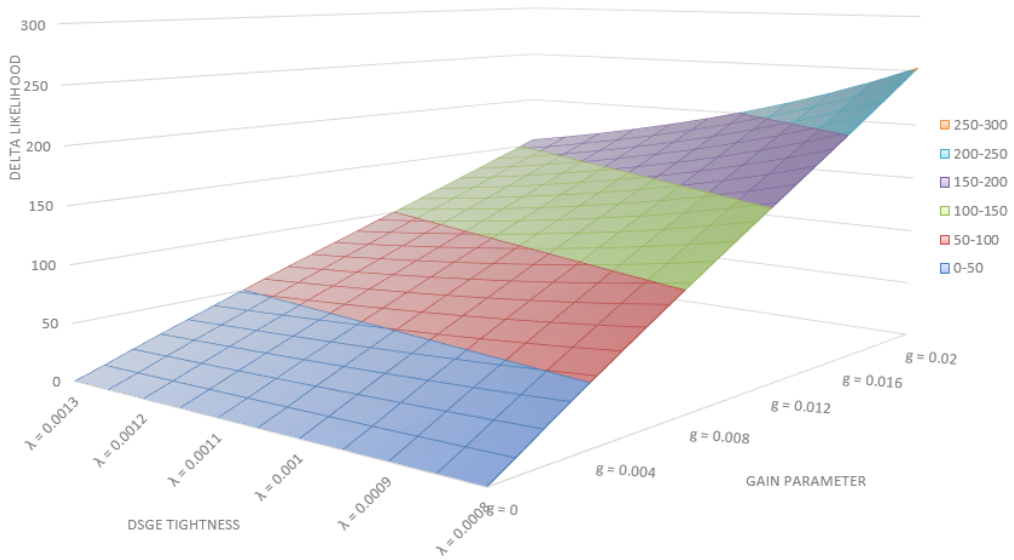

Figure 2.3: Likelihood improvements of LDSGE-VAR relative to the zero gain LDSGE-VAR, model 2

The forecast improvements are showed next for our artificial data. Our measure of improvement is given by, first, averaging mean squared errors (MSE) across equations from one to four periods ahead[56] for the LDSGE-VAR$(g, \theta^{RE}_{MAX}, \lambda^{best}, \bar{p})$ model, and comparing

[56]To forecast some variable X_{t+4} at time t, we use period t parameter values in order to estimate the *level* of variable X_{t+4}. Hence, if specification (a) produce the following sequence of errors: $[0.1; 0.1; 0.1; 0.1]$

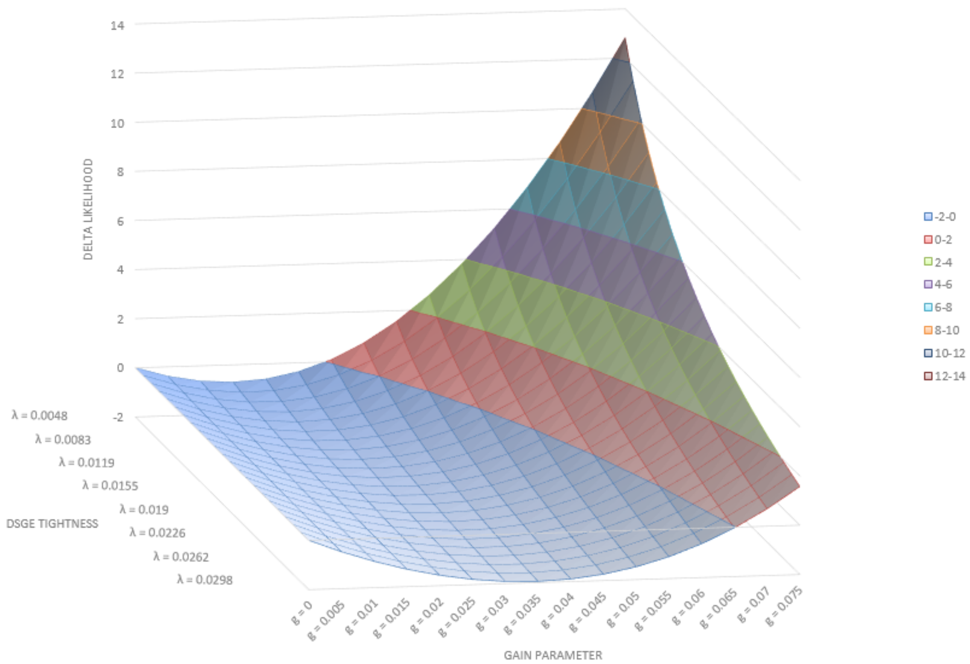

Figure 2.4: Likelihood improvements of LDSGE-VAR relative to the zero gain LDSGE-VAR, model 3

it to the zero gain LDSGE-VAR$(g = 0, \theta^{RE}_{MAX}, \lambda^{best}, \bar{p})$. There are forecast improvements for the univariate model up to two periods ahead, even though these improvements (as in the likelihood) are "supposedly" small (we will discuss this topic in more detail in chapter 3). Results for model 2 were consistent to the worse likelihood surface that was presented: all forecasting horizons considered show worse results for positive gain parameters.

Overall, results for the LDSGE-VAR model showed that there is room for improvement under such type of estimation method. The estimated tightness λ was pushed towards zero, confirming that the DSGE restrictions needed to be relaxed in order to match our artificial data, which had an structural break that allowed parameters under learning to vary over time. We also calculated in-sample forecast improvements for our artificial data, which showed results consistent to the improvements in their fit: model 1 showed better forecasts up to two periods ahead, and model 2 showed worse forecasts under positive gains for all forecast horizons considered.

and specification (b) produces the sequence $[0.1; -0.1; 0.1; -0.1]$, our measure of forecast accuracy will penalize specification (a) by a larger extent, since the level of the output variable will be different from specification (b), even though both produced the same mean squared errors.

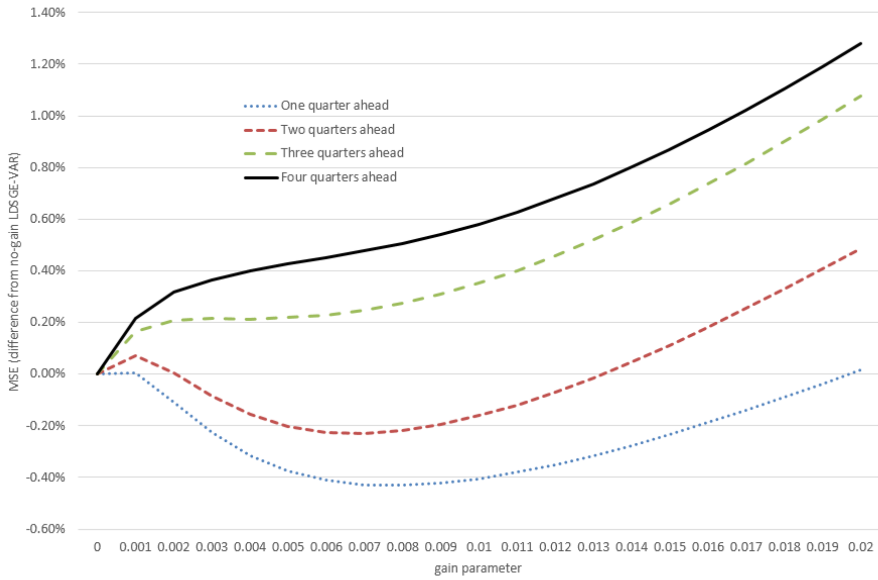

Figure 2.5: Forecast improvements for artificial data, univariate model

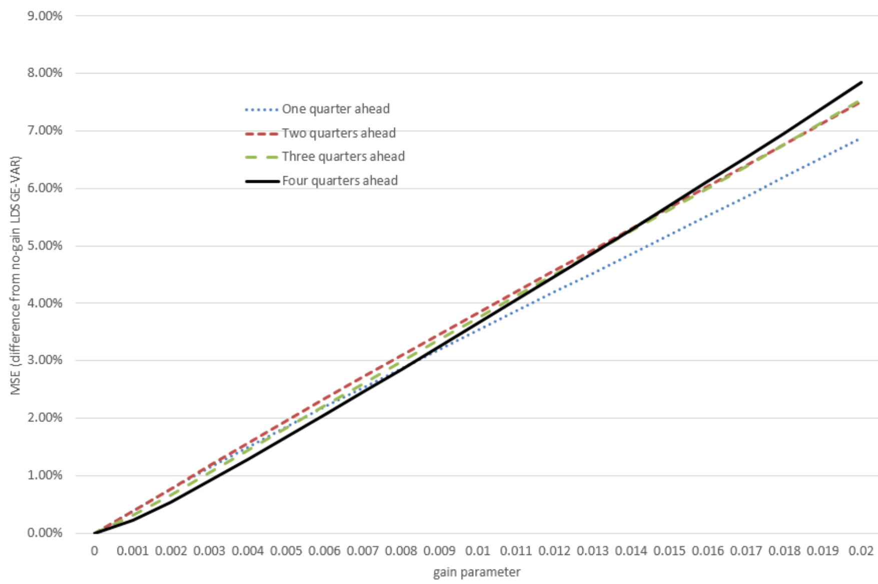

Figure 2.6: Forecast improvements for artificial data, basic New Keynesian model

We now present results for the LMSV method. The figure below illustrates the main result[57] for the univariate model. The vertical axis refers to the value of the likelihood of the models (reminding that the smaller the likelihood, the better the fit). The horizontal axis refers to the gain parameter of the LMSV models. Our benchmark is the best

[57]Detailed tables are available in Appendix A.

performing time t LDSGE model, but we decided to include the best time $t-1$ LDSGE as well (the red horizontal line).

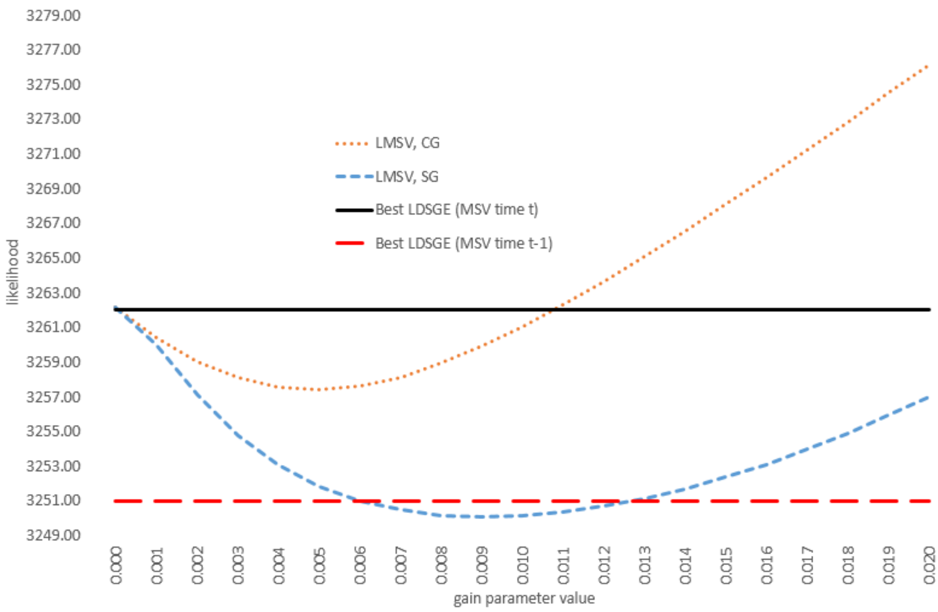

Figure 2.7: Likelihoods of LMSV for model 1

Our flexible learning MSV mechanism (LMSV) was able to perform better than its time t LDSGE counterpart by 12 (stochastic gradient, red dotted line) and 4.7 (constant gain, blue line) points. The best likelihood of the LMSV for the univariate model was achieved under stochastic gradient learning algorithm (3250.08), with an estimated gain of 0.009; constant gain algorithm showed a worse performance, with the best marginal likelihood around 3257.39, and an estimated gain around 0.005. This is compared with the best time t LDSGE model marginal likelihood, 3262.1. Detailed results can be found in the detailed tables in Appendix A. We also plotted the marginal likelihood of the best LDSGE model using time $t-1$ Perceived Law of Motion, which could improve LDSGE results by adding more flexibility to the Actual Law of Motion: the best marginal likelihood in this case was 3251.0. Comparing to that, only stochastic gradient was able to perform better, by 0.9 points.

As we noted before, the point estimations of both α and β (the first ranging from 0.47 to 0.50, the second from 0.39 to 0.45) contributed to make the time-varying system more flexible and to match more closely the true Law of Motion of the economy. This would produce a higher improvement in the likelihood than in the case where, for instance, β is calibrated or identified through another equation. For the sake of illustration, we reestimated the best global LDSGE specification (MSV learning, constant gain algorithm, Perceived Law of Motion with timing $t-1$, and marginal likelihood 3251.0) but now fixing $\beta = 0.30$. This estimation produced a marginal likelihood of 3253.2, 2.2 points worse than the original estimation (we also estimated the best time t best LDSGE model fixing $\beta = 0.30$, however since the estimated gain was zero (both in the original and in this reestimation), the difference between the marginal likelihoods was negligible).

Also important is the evolution in time of the estimated Actual Law of Motion of the economy. We display five paths in the figure below: the first relates to the theoretical path of the economy, if the variance of our artificial errors was zero. The second path is related to the LMSV model with best performance (stochastic gradient); third and fourth paths are related, respectively, to the best performing LDSGE specifications using time t and time $t-1$ Perceived Law of Motion timing. Since time t LDSGE showed zero gain, it has an equal transition matrix (Actual Law of Motion) than the pure rational expectations model.

Last, we plot the OLS estimates of the Actual Law of Motion (blue dashed line). This is done because our model has a quite large variance and we work with only one "realization of the stochastic process", as it is known in time series econometrics. It shows that our sample average coefficient is estimated to be around 0.58 for the first 500 observations and around 0.75 for the last 500 observations.

LMSV and LDSGE with timing $t-1$ showed a fairly different evolution of the economy, specially taking into account their "modest" difference in marginal likelihoods from each other, of 0.9 points. We will develop this topic more deeply in our third chapter. As for now, we note that, as expected, the restrictions of the LDSGE methodology prevents the transition matrix from going too far apart from the rational expectations equilibrium, which does not happens under LMSV.

The evolution of the transition matrix of the economy under learning clearly shows how learning is able to outperform rational expectations in dealing with our artificial data.

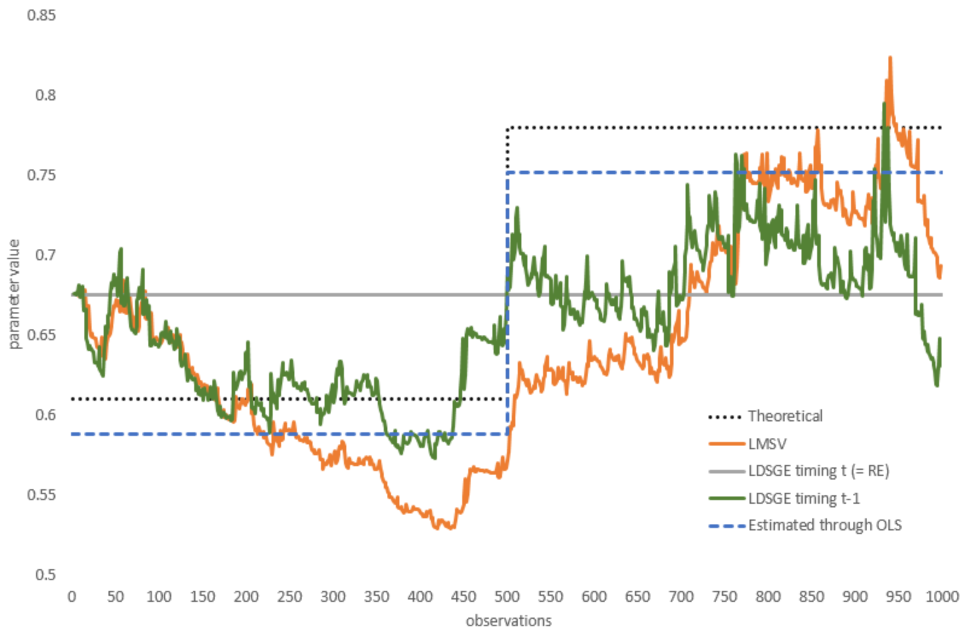

Figure 2.8: Evolution of lagged observable variable parameter, univariate model

Also important, if we chose to estimate our LDSGE and LMSV models using optimised initial beliefs, the marginal likelihood would be improved even further (since, as we can see, the initial conditions for both are given by the rational expectations time-invariant DSGE model).

Now let us move to a more realistic and complex model, the canonical New Keynesian model (model 2), and see how learning changes the model dynamics and performance. Results for model 2 showed that the learning mechanism has a cost in terms of excess volatility of parameters. When simulating data from model 2, we chose to add structural breaks only in one of the three equations of the model: the response of output for a drop in *ex-ante* real interest rates, measured by coefficient α, dropped from 0.5 to 0.3.

The learning mechanisms of chapter 1 did not detect evidences of learning for the economy (figure below), even though we generated our artificial data by changing α over time. The idea is that the volatility in the overall system did not compensate the fit in the output gap equation in terms of posterior/likelihood.

Since our gain parameter affects the reduced-form parameters on agents' Perceived Law of Motion equally, all reduced-form parameters are allowed to change over time and

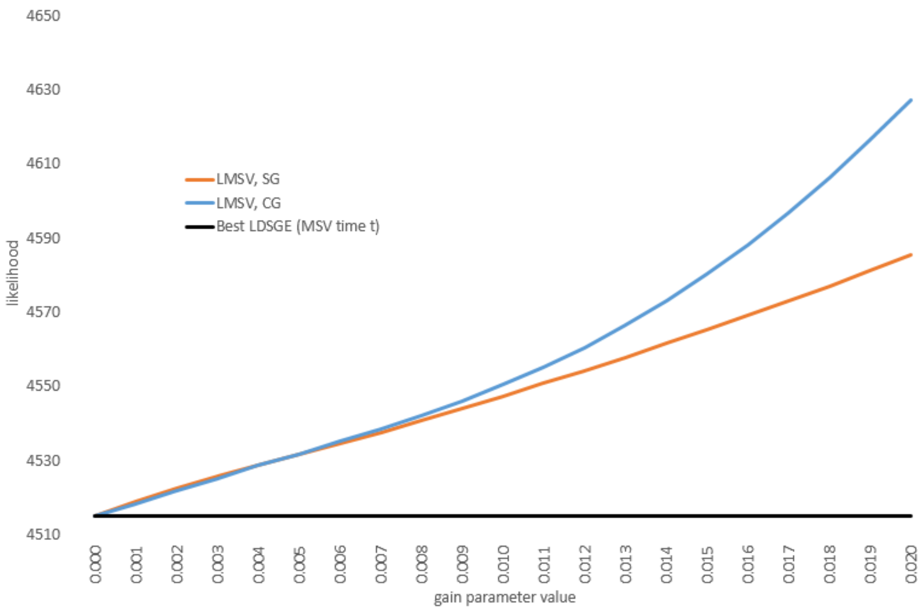

Figure 2.9: Likelihoods of LMSV for model 2

therefore the Actual Law of Motion as a whole[58] can fluctuate. This excess volatility can worsen the likelihood, since most of the data has a time invariant Data Generating Process.

This result was confirmed when the DSGE restrictions on how learning affect the economy were disabled through the LMSV methodology. This result is not surprising given that now there could be more volatility in the overall economy. Our LMSV learning model did not find sufficient evidences of learning in the simulated data which contains a structural break. The excess volatility induced by positive gains lead to large losses in the likelihood, as shown in the previous figure.

This result strengthens our learning models as a relevant mechanism to check and explain our data, since we can see more clearly a cost, in terms of likelihood coming from excess volatility, which penalizes our method. The estimated gain will be greater than zero only if the improvement in the likelihood/posterior function compensates the loss caused by the volatility of parameters.

For the third model, the more complex DSGE model inspired in Smets & Wouters (2003), our LMSV learning mechanism was able to improve the marginal likelihood by 1

[58] Given restrictions of the DSGE model, as explained above.

point, from -1494.3 (9^{th} specification) to -1495.3, under constant gain learning algorithm (which delivered an estimated gain of 0.042), and by 0.3 points, from -1494.3 to -1494.6 under stochastic gradient learning algorithm (which presented an estimated gain of 0.42).[59]

It is interesting to note that the flexibility of the LMSV mechanism produced a better marginal likelihood than LDSGE models without constants, but could not achieve the marginal likelihood of the LDSGE with constants (-1502.2, not shown in the figure, but avaiable in the Detailed tables section), which tells us that it is not the DSGE model restrictions that are unable to improve the marginal likelihood, but instead some structural break that is translated in time-varying constants (a feature that tends to be important in emerging economies): the best model with constants showed a marginal likelihood 7 points better than our LMSV.

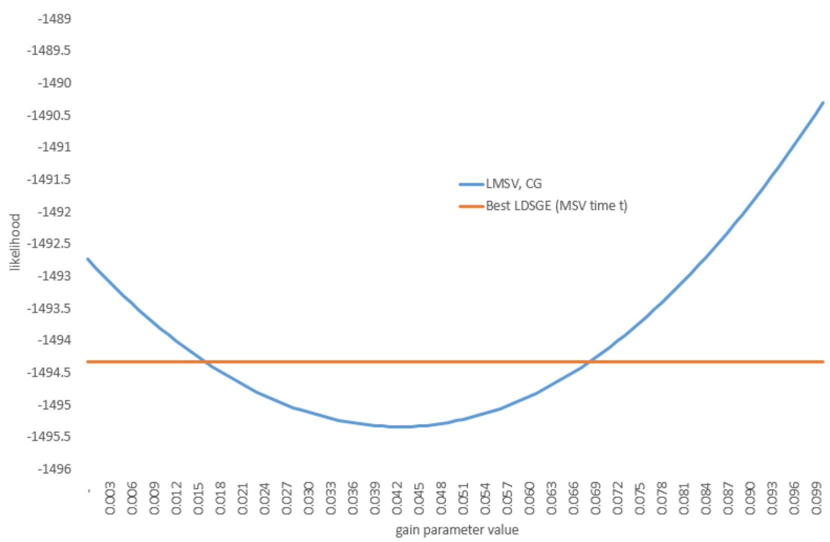

Figure 2.10: Likelihoods of LMSV for model 3, constant gain

Slobodyan and Wouters (2012), for instance, estimated a learning DSGE model similar to ours[60] and found an improvement of 0.2 points in the marginal likelihood while

[59] One known disadvantage of the stochastic gradient learning algorithm is that it is not scale invariant. If the scale of the variables is changed (for instance, multiplying every variable by 100), a stochastic gradient gain coefficient fixed at $g_{SG} = \bar{g}$ will produce larger updates in the recursive coefficients if the model is rerun under the new scaled variables. Unlike constant gain algorithm, in which the gain parameter is always estimated between 0 and 1, the stochastic gradient gain parameter can be greater than 1. Therefore, stochastic gradient and constant gain parameter values are not comparable.

[60] Also based in Smets and Wouters' model.

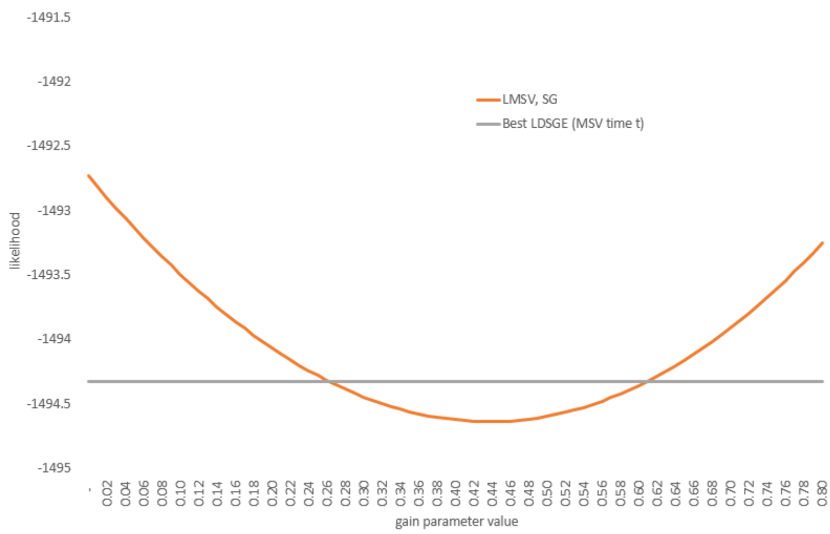

Figure 2.11: Likelihoods of LMSV for model 3, stochastic gradient

comparing their learning model (MSV time t Perceived Law of Motion and rational expectations consistent initial beliefs) to the rational expectations DSGE model. Adding constants to the Perceived Law of Motion did not produce relevant changes in their marginal likelihood. The authors found out that optimised beliefs were the most important source of improvement of the marginal likelihood, responsible for an improvement of 12.4 points.[61]

This suggests that our relative improvement of the marginal likelihood due to time variation is relevant compared to their result, specially when taking into account that their model covered US economy from 1966 to 2005, while our data ranged from 2002 to 2012 (hence, as their data sample was much larger, the a priori probability of their sample containing a structural break was larger).

We point out three topics not covered in our simulations but that may be interesting for future research: (i) to reproduce the estimation of LDSGE models with optimised initial beliefs, and estimate how further the marginal likelihood could be improved by them; (ii) to include constants in our LMSV model; and (iii) to perform the estimation of (ii) using optimised initial beliefs, in order the achieve the most flexible learning model

[61] Our measure of posterior and marginal likelihood has inverse signal than theirs, such that smaller values of posterior/likelihood for us is an improvement, and the opposite for the authors.

consistent with the structure of the DSGE model (added with constants).

2.5 Conclusions

We started this chapter explaining two limitations/restrictions of the standard Learning DSGE models. The first limitation, discussed in the numerical example at the beginning of this chapter, was the incapacity of the learning mechanism to fully match the true Actual Law of Motion of the economy, since this would require reduced-form parameters being estimated at values not compatible with information coming from the artificial data. The second limitation (which we prefer to call restriction, since it is model specific) was the fact that the learning mechanism would not produce time varying dynamics in all equations of the learning model, as we showed in the bivariate model given by (2.34). These limitations motivated us to pursue more flexible methods to improve fit and assess how well our LDSGE performed if its limitations were disabled. To do so, we introduced two new methods: Learning DSGE-VAR (LDSGE-VAR) and Learning MSV (LMSV), which gradually departed from the structural DSGE parameters estimation and focused in reproducing the dynamics of the artificial data.

LDSGE-VAR models were able to improve the fit of models 1 and 3, in comparison to the its corresponding DSGE-VAR (which we call zero gain LDSGE-VAR). The estimated surface of likelihoods for those models showed that, as the DSGE restrictions became tighter, the more important the gain mechanism became in order to compensate for this excessive tightness. In-sample forecast improvements for model 1 showed better forecasts up to two periods ahead, while model 2 showed worse forecasts for all positive gains considered.

LMSV models used the best time t parametrization from the LDSGE and disabled the DSGE restrictions, allowing for the transition matrix to vary freely. It was able to improve the likelihood of the univariate model by 12 and 4.7 points (under the same information set) and by 0.9 points (under the best LDSGE specification, with time $t-1$ Perceived Law of Motion). The time evolution of the transition matrix for this model showed how, for the same artificial data, our method was able to adjust more accurately to the "true" law of motion of the economy, in comparison to the LDSGE model, and how the rational expectations time-invariant DSGE model performed, estimating an "average" law of motion for the full sample. For the second model, LMSV performed worse than LDSGE and the rational expectations model for all positive gains. The excess volatility induced by learning was not sufficiently compensated by the improvement in forecasts. Results for the third model showed an improvement of 1 point, in comparison to its equivalent LDSGE specification, but it was worse than the best LDSGE specification, which had constants.

We gradually departed from the DSGE model in these two chapters: our first learning mechanism, in chapter 1, changed the way expectations were formed, and showed improvements in the fit in relation to the rational expectations model. In this second chapter, we presented the LDSGE-VAR method, which still maintained links with the original DSGE model, and the LMSV, which maintained the DSGE solution structure and initial conditions as the only memory of the DSGE model. In the next chapter, we will take a further step and fully break the link between learning and the DSGE solution structure. We will work with reduced-form observable variables models, pointing out why our "modest" results of likelihood improvements are due to the model at hand, and not to our learning approach.

3 Learning in reduced-form models and empirical evidences for Brazil

Some of the posterior/likelihood improvements presented in the last chapters may suggest that allowing for time variation in the Law of Motion of the economy would lead to small or moderate improvements in the fit of the model. This is important particularly for those looking for alternative methods to best describe the data at hand.

In this chapter we provide evidences that the supposedly small improvements found in the previous chapters have more to do with the nature of the model at hand than to the learning method itself. To do so, we simulate problems using a time-varying structure similar to the one presented in chapter 1 and evaluate the likelihood improvements with different learning mechanisms.[62]

We then provide empirical evidences of learning in reduced-form models to forecast inflation, interest rates and output for the Brazilian economy, using *ad-hoc* reduced-form models commonly used by pratictioners (even if they are inspired by new Keynesian models discussed in chapter 1).

This last chapter is organized as follows: section 1 presents the motivation and the methodology we will use in our reduced-form model simulations; section 2 presents the reduced-form model and the time-variation problems that are being simulated; section 3 presents the main results; section 4 estimates standard reduced-form models for inflation, output and interest rates for the Brazilian economy; section 5 concludes.

3.1 Motivation and methodology

As we saw in chapter 1, the improvement in the likelihood function for the univariate model was around 11 points. Given the large sample size ($T = 1000$), one might argue

[62] In this chapter we fully depart from the Bayesian perspective, going back to standard econometrics. Since we are now estimating data driven reduced-form models, there is no clear prior for these reduced form parameters.

that the improvement in the likelihood was rather small. We shall argue here that this "modest" improvement is more related to characteristics of the model than to the learning mechanism itself.

To argue so, we present a generic reduced-form learning model (LRF), given by:

$$Y_t = X_t \beta_t + \varepsilon_t \tag{3.1}$$

$$\varepsilon_t \sim N(0, \Sigma\varepsilon) \tag{3.2}$$

where Y_t is a vector and X_t is a matrix of observable variables, ε_t are reduced-form errors and β_t is a matrix containing (reduced-form) parameters that evolve over time by a learning mechanism, given by:

$$\beta_t = (\beta'_{t-1} + g_t R_t^{-1} X'_{t-1}(Y_{t-1} - X_{t-1}\beta_{t-1}))' \tag{3.3}$$

$$R_t = R_{t-1} + g_t(X'_{t-1}X_{t-1} - R_{t-1}), \tag{3.4}$$

where R_t is, as before, the matrix corresponding to the second moments of X_t. We study the cases in which gain sequences are given by a constant gain $g_t = g$ or by a stochastic gradient, with $g_t = gR_t$. Like in the previous chapters, initial conditions may come from subsample regressions, fixed values or from some structural model. Since parameters can vary freely over time, as in chapter 2 under LMSV, initial conditions will only affect the likelihood of the system in the initial data points, and will not be our main focus here.[63]

There are two main differences between LMSV and LRF methods (and their MATLAB codes). One advantage of LMSV is that it does allow for unobserved variables (using the Kalman Filter to generate the likelihood function) and hence can account for autorregressive or moving average (AR/MA) terms in a straightforward fashion. On the other hand, LMSV does that by solving a DSGE model, which is costly in terms of testing alternative reduced-form specifications.

Our LRF MATLAB code, on the other hand, requires only the specification of Y_t and X_t as inputs, making the task of estimating alternative reduced-form models faster

[63] Our MATLAB code allows for fixed or full sample OLS initial conditions.

(in comparison to LMSV codes, which would require alternative specifications of DSGE models). Autorregressive or moving average terms can be approximated by standard time series methods, such as adding lagged endogenous variables to the right hand side of the equation. Our MATLAB code default setting is to use the full sample OLS estimates to generate initial conditions for the parameters, but the user can provide specific matrices.

The likelihood function[64] of this system can then be evaluated by:

$$L_t = \log(2\pi) + \log(\det(\Sigma_\varepsilon)) + \varepsilon_t'(\Sigma_\varepsilon)^{-1}\varepsilon_t \tag{3.5}$$

The difference between this and the one calculated via Kalman Filter in (5.50) is that the latter allows for a time-varying second moment matrix F_t, which is a function of the second moments of filtered state variables and the transition matrix T_t. When the variance of state variables is low, the middle term of (5.50) becomes lower (improving the likelihood) and the right term become larger, penalizing residuals.

Our MATLAB code also allows for updating of the matrix Σ_ε over time. The idea is to *approximate* (if the update is enabled) the dynamics of the LRF likelihood to the one under LMSV. The evolution of Σ_{ε_t} in this case is given by:

$$\Sigma_{\varepsilon t} = \Sigma_{\varepsilon_{t-1}} + g_t(\varepsilon_{t-1}'\varepsilon_{t-1} - \Sigma_{\varepsilon_{t-1}}) \tag{3.6}$$

We will now describe our simulations.

3.2 Simulations

We will perform two groups of simulations using the reduced-form model presented in the previous section. The first group is given by an univariate model given by (3.1) and (3.2), with $\beta_t = 0.61$ for the first 500 observations and $\beta_t = 0.78$ for the last 500 observations. X_t is given by a random number generator with variance V_1 and V_2. This is done to replicate the time varying second moments of the law of motion used in the first chapter, which is given by:

[64] In fact, this is a transformation of the true likelihood function, which has some constants and the opposite sign. Hence the lower our likelihood, the best is the fit of the model.

$$
\begin{aligned}
y_t &= \beta y_{t-1} + \varepsilon_t & (3.7)\\
V(y_t) &= \beta^2 V(y_{t-1}) + V(\varepsilon_t) & (3.8)\\
V_y &= \beta^2 V_y + \Sigma_\varepsilon & (3.9)\\
V_y &= \frac{1}{1-\beta^2}\Sigma_\varepsilon & (3.10)
\end{aligned}
$$

where V_y denotes the unconditional variance of y_t. This model gives us $V_y = 1.59\Sigma_\varepsilon$ if $\beta = 0.61$ and $V_y = 2.55\Sigma_\varepsilon$ if $\beta = 0.78$. So, in order to fully replicate the time variation of the variance of the left-hand side variable, we simulate X_t with time-varying variance.

The second group is an univariate model with the same characteristics but now X_t is given by Y_{t-1}, which is the univariate model of chapter one. We do not need to apply any correction to the variance since now the law of motion is exactly the same as before.

For each group, we simulate $T = 1000$ observations (each half with a particular set of parameters, as explained above) and calculate three likelihoods: the first comes from the full sample OLS regression, which is the standard, reduced-form, time-invariant model. The second likelihood is also given by the OLS regression, but now splitting the sample in half and estimating parameters for each subsample. Since we generated data this way, this is the best expected possible likelihood that can be achieved by OLS, and represents our benchmark to assess how well our learning mechanism is able to match the data. The third likelihood is given by estimating the reduced-form model under our learning mechanism, for different values of the gain parameter, and with initial conditions coming from the full sample OLS estimation (which penalizes in some extent our learning model).

We do the simulations above for a grid of constants multiplying the variance of the simulated error, ranging from 0.1 to 1, in order to assess the relation between likelihood improvements and the variance of the error term. In order to eliminate specific sample results, we replicate the likelihood estimation procedure one thousand times for the first group and one and ten thousand times for the second group, and average the results.

3.3 Main results

Figures (3.1) and (3.2) below presents the likelihood estimates for the first group of simulations. As before, our measure of likelihood is such that the smaller its value, the better the fit. We plot the likelihood estimates as functions of a multiplicative constant k to the standard deviation of errors, ranging from 0.1 to 1 (hence, the simulated variance of errors ranged from 0.01 to 1 times the original variance).

There are several likelihood curves plotted for the first group: the two thick lines refer to the OLS regressions using the full sample and the split sample (as expected, splitting the sample produced a much better (lower) likelihood). Dashed lines refer to the learning mechanism using constant gain learning algorithm and updating of $\Sigma\varepsilon_t$ over time, for different values of the gain parameter.

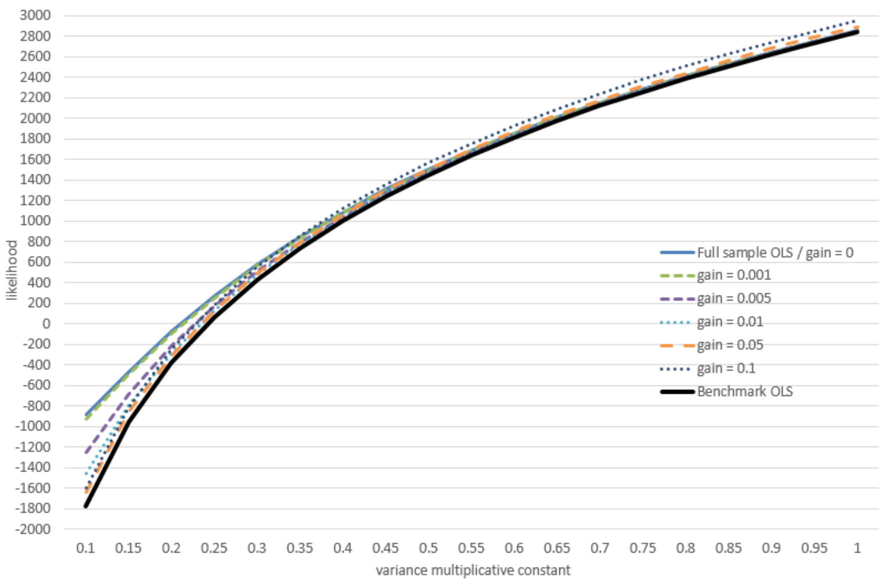

Figure 3.1: Average likelihood curves for the first group, one thousand simulations

The main result is that our learning mechanism was able to improve the likelihood by a considerable extent relatively to the OLS benchmark. For small variances, positive gains showed large improvements. When variances are higher, however, some of the larger gains produced worse likelihoods relatively to the full sample OLS estimates: the loss of likelihood from $gain = 0.05$ to $gain = 0.1$ for larger variances indicates that large gain

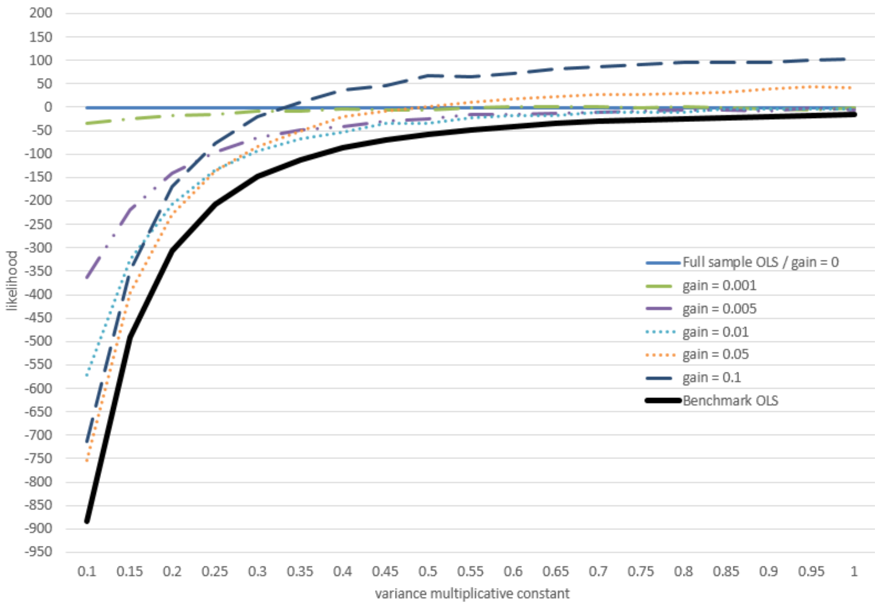

Figure 3.2: Average likelihood differentials from the full sample OLS estimation, one thousand simulations

parameters will worsen the likelihood via excess volatility of parameters (as we argued in the previous chapter).

Results for the second group of simulations (in which $X_t = Y_{t-1}$) are presented in the figures below. The benchmark model showed a very modest improvement in the likelihood: around 16 points, in comparison to the 900 points achieved when X_t was not the lagged endogenous variable (compare the benchmark OLS case of figures (3.4) and (3.2), for the smallest variance considered).

The magnitude of the difference of improvements between the two groups is remarkable. It confirms that the supposedly "small to moderate" improvements found in the last two chapters are mainly due to the structure of the model at hand, and not to our learning mechanism. The split sample OLS estimates provided an average improvement of 16 points, while learning models average improvements ranged from -2 to 9 points.

It is important to note, however, that the split OLS estimates must not be seen as the maximum possible improvement, but as the benchmark average result. Learning improvements for particular samples were able to surpass the fit of the split model. Also striking is that even averaging over one thousand simulations (each containing $T = 1000$ observations) the likelihood improvement curves are still not well behaved for the second

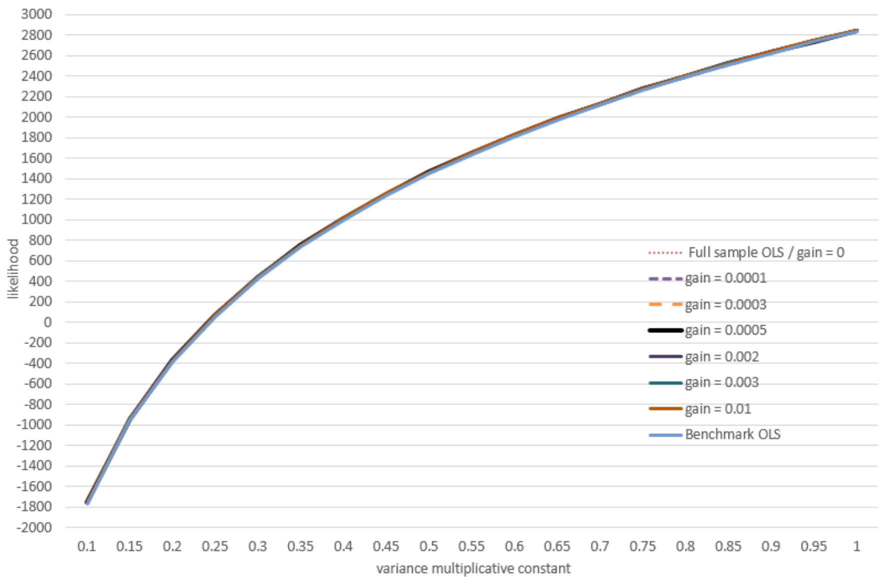

Figure 3.3: Average likelihood curves when X is the endogenous lagged variable, one thousand simulations

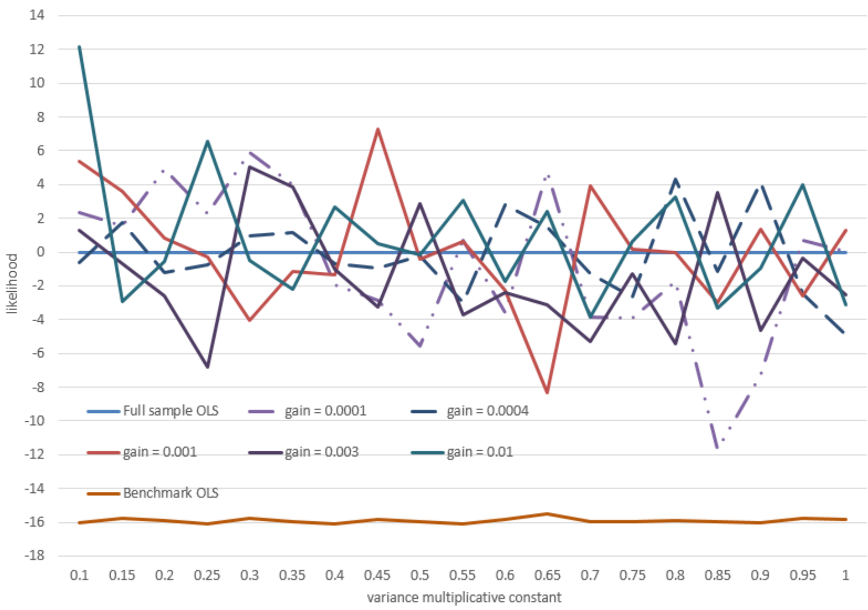

Figure 3.4: Average likelihood differentials from the full sample OLS when X is the endogenous lagged variable, one thousand simulations

group of simulations. In order to smooth results, we redid the same exercise averaging between ten thousand simulations. The results became a lot smoother, as we see in Figure (3.5).

Also, it is important to point out that (against intuition) improvements did not show a strong negative relation with variance of errors in the second group. Actually, a linear

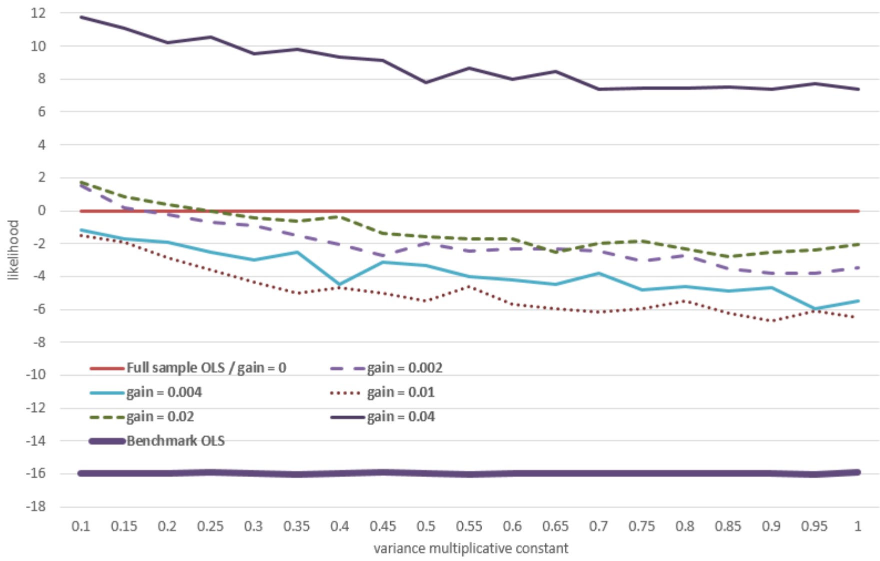

Figure 3.5: Average likelihood differentials from the full sample OLS when X is the endogenous lagged variable, ten thousand simulations

regression would suggest a slightly positive relation. So the idea that the large variance of errors of model 1 would reduce the likelihood improvements was not supported by the simulations.

The results from these simulated examples are evidences that a relevant part of the supposedly "small to moderate" likelihood improvements in the last two chapters had more to do with the type of model being estimated than with the learning mechanism we applied to the artificial data. This result reinforces learning as a relevant empirical mechanism to improve fit, and sheds some light into the expected likelihood improvements one might achieve under learning when estimating different types of models.

3.4 Some empirical evidences for Brazil

3.4.1 Equations and data

We aim to assess some LRF models empirical evidences. To do so, we estimate three reduced-form models for the Brazilian economy. We chose to use some *ad-hoc* models from praticitioners, acknowledging the fact that these models are part of a set of models that analysts use to forecast major macroeconomic variables, along side with DSGE models, (Bayesian) VARs and other calibrated models. For instance, some of the models used by the Brazilian Central Bank are: Stochastic Analytical Model with a Bayesian Approach

(SAMBA), a large scale DSGE model;[65] a semi-structural model designed to describe deviations of variables from their trends, as in Minella et al. (2013), and several small-scale reduced-form models, some of them described in the boxes of the Quarterly Inflation Report[66] (for instance, we highlight the Jun/2013 and Mar/2013 reports, which contain updates of some of those models).

It is common among private sector agents or pratictioners in general to also make use of several types of models (and try to replicate the ones used by the Central Bank), which often produce different elasticities and forecasts. Here, we use an example of one of those models. Hence, the results presented here have no intention of being general, but to provide evidences in favor of learning in some models.

The first equation is the output gap equation, which is given by:

$$GAP_t = \beta_0 + \beta_1 GAP_{t-1} + \beta_2(PRE_t - FOCUS_t - JRN_t) + \\ \beta_3 FISC_t + \beta_4 WGAP_t + \varepsilon_t^{gap} \quad (3.11)$$

where GAP_t is the level of output gap measured by the average of two alternative ways to calculate output gaps in Brazil (the first way can be found in Areosa (2009); the second is the standard HP filter); PRE_t is market interest rates one year ahead (PRE-DI 360), $FOCUS_t$ is the median market expectation of inflation 12 months ahead; JRN_t is the natural rate of interest, given by the maximum[67] of two values: 4% and the result of filtering $(PRE_t - FOCUS_t)$ using a standard HP filter; $FISC_t$ is primary surplus (discounted by exceptional[68] revenues) minus 1.5%;[69] $WGAP_t$ is the output gap for the world, given by filtering a measure of aggregate world output (weighted by PPP) using an HP filter; ε_t^{gap} is the error. The full sample OLS estimates are presented in Figure (3.6).

[65] See Minella et al (2011).

[66] Available at http://www.bcb.gov.br/?BOXESRELINF

[67] We do so since there is a large uncertainty of the level of natural interest rates in Brazil. We build our measure of natural interest rates as the minimum of the HP filtered *ex-ante* interest rates and 4%. This rule becomes binding at the end of the sample.

[68] We exclude from the fiscal surplus excess dividends, revenues from the sovereign fund and government concessions. This is due to produce a measure of fiscal surplus that has a cleaner impact on the output gap.

[69] This is done just to normalize the fiscal surplus around what is the value needed to stabilize the Debt / GDP ratio. It does not have an impact on estimates.

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	0.004294	0.001280	3.356196	0.0020
GAP(-1)	0.478444	0.072998	6.554167	0.0000
PRE(-2)-FOCUS(-2)-JRN(-2)	-0.238060	0.083521	-2.850294	0.0075
FISC-0.015	-0.708868	0.168702	-4.201899	0.0002
WGAP	0.652224	0.113902	5.726191	0.0000

Figure 3.6: OLS estimates for output gap equation

As can be seen, all the coefficients have the expected signal and have small p-values.

The second equation is the inflation equation (Phillips curve), given by:

$$\begin{aligned} \log(CPI_t/CPI_{t-4}) = \beta_0 + \beta_1 \log(CPI_{t-1}/CPI_{t-5}) + \\ \beta_2[\log(CRB_t/CPI_{t-4}) + \log(CRB_{t-1}/CRB_{t-5})]/2 + \\ \beta_3[\log(BRL_t/BRL_{t-4}) + \log(BRL_{t-1}/BRL_{t-5})]/2 + \\ \beta_4 GAPt + \beta_5 D2006q2 + \varepsilon_t^{CPI} \end{aligned} \quad (3.12)$$

in which CPI_t is the official price index for Brazil (IPCA), CRB_t^{US} is the commodity prices expressed in US dollars, BRL_t is the nominal exchange rate, $GAPt$ is the level of output gap presented earlier, $D2006q2$ is a dummy variable, and ε_t^{CPI} is the error term. The OLS regression estimates is provided below:

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	0.017470	0.004599	3.798413	0.0007
LOG(CPI(-1)/CPI(-5))	0.685727	0.079854	8.587287	0.0000
@MOVAV(LOG(CRBT/CRBT(-4)),2)	0.009629	0.009256	1.040307	0.3065
@MOVAV(LOG(BRL/BRL(-4)),2)	0.024009	0.012140	1.977602	0.0572
GAP	0.273440	0.077903	3.510006	0.0014
D2006Q2	-0.016891	0.006413	-2.633916	0.0132

Figure 3.7: OLS estimates for inflation equation

All estimated coefficients have the expected signal, but the coefficient for CRB prices has low precision.

The third and last equation is a simple version of Taylor rule, given by:

$$(SELICt - JRN_t - FOCUS_t) = \beta_0 + \beta_1(SELIC_{t-1} - JRN_{t-1} - FOCUS_{t-1}) + \beta_2 GAP_t + \beta_3(CPI_t^{12} - CPI_t^{TARG}) + \varepsilon_t^{SELIC} \quad (3.13)$$

where CPI_t^{TARG} is the inflation target for inflation, CPI_t^{12} is the inflation rate over the past year and ε_t^{SELIC} is the error term. The OLS regression is presented below.

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	-0.003615	0.001536	-2.353677	0.0249
SELIC(-1)-JRN(-1)-FOCUS(-1)	0.833350	0.082970	10.04405	0.0000
GAP	0.154115	0.076195	2.022630	0.0515
CPI12-CPI_TARGET	0.415405	0.115547	3.595105	0.0011

Figure 3.8: OLS estimates for interest rates equation

Coefficients have the correct expected signal and are significant (except from the output gap coefficient, which has a lower precision than the other coefficients).

Data sources are IBGE (IPCA inflation and output), Brazil's Central Bank (Selic rate, market interest rates (PRE-DI 360), inflation expectations and nominal exchange rates), Bloomberg (CRB) and IMF (World GDP). Output gap equation uses data from 2003Q3 to 2012Q4; inflation and Taylor rule equations uses data from 2004Q1 to 2012Q4 due to data availability.

3.4.2 Results

The following table summarizes our main findings.[70] Stochastic gradient learning was able to improve the likelihood of two of the three equations relatively to the benchmark OLS by a reasonable extent. Output gap equation showed an improvement of 20.4 points, from -267.5 to -287.9, while Taylor rule equation showed an improvement of 3.1 points, from -247.5 to -250.3. Constant gain learning was not able to improve the likelihood of any equations by a considerable extent: the estimated gain was virtually zero in all equations.

Table 3.1: Estimated likelihoods for the three reduced form equations for Brazil

	Likelihood Benchmark OLS	Likelihood / gain LRF, Constant gain	Likelihood / gain LRF, Stochastic gradient
Output gap Equation	-267.5	-267.4 / 0	-287.9 / 0.77
Inflation Equation	-265.4	-265.5 / 0	-265.4 / 0
Taylor rule Equation	-247.5	-247.5 / 0	-250.6 / 0.188

The fact that only the stochastic gradient learning was able to improve the likelihood of our reduced-form models is important. We argued in the second chapter that the stochastic gradient mechanism is not scale invariant, which may favor the updating of parameters referring to variables that are expressed in a larger scale. For instance, our largest explanatory variable (comparatively to all others in each equation) is the constant, which will receive a larger update from SG, comparatively to the other explanatory variables. Since the constant will receive most of the update, the cost in terms of volatility of parameters will also be adjusted downwards, since now the other parameters are closer to the benchmark OLS estimates. The four figures below show the evolution of the reduced-form parameters (as differentials from their OLS, time-invariant, full sample estimates, given by $\beta_t^Z - \beta_{OLS}^Z$ for a given explanatory variable Z) for the output gap and Taylor rule equations. Since the updating of constants is larger, we plotted two graphs for each equation: one containing the evolution of all parameters, and another containing the evolution of all parameters except the constant.

The evolution over time of the reduced-form parameters for the output gap suggest, first, that the Brazilian output (gap) became more correlated to the growth of the international economy after the 2009 crisis: the world output gap parameter rose in the 2008-2009 period and declined after 2010, but to a level higher that the OLS estimates and the 2004-2008 period. On the other hand, the autorregressive parameter for the output gap declined in the period, except during the 2008-2009 subprime crisis. The same occurred to the Fiscal surpluses, which became a little less important to explain output gap in our sample. Monetary policy, on the other hand, showed a little increase in its sensitivity to the output gap (since this coefficient is negative and became *more* negative).

[70] Detailed tables are available in Appendix A.

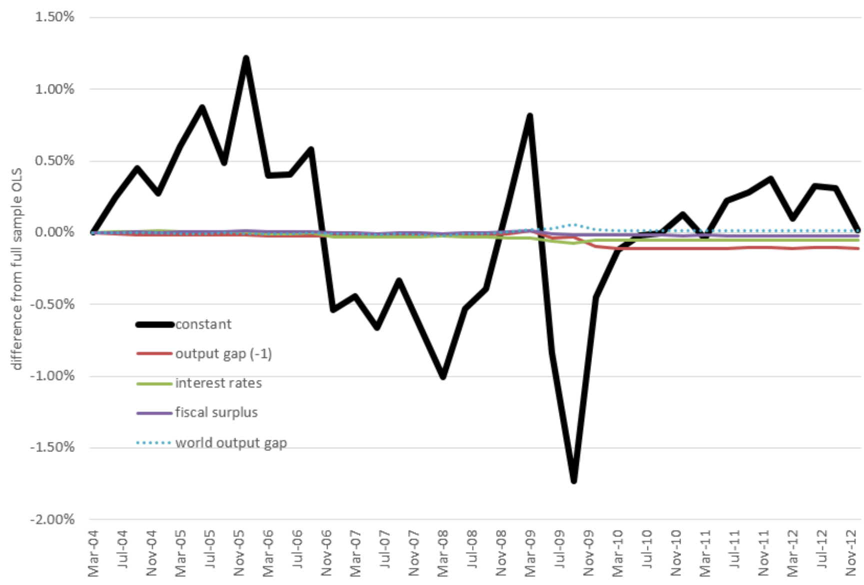

Figure 3.9: Evolution of the difference between learning parameters and benchmark OLS estimates: output gap equation

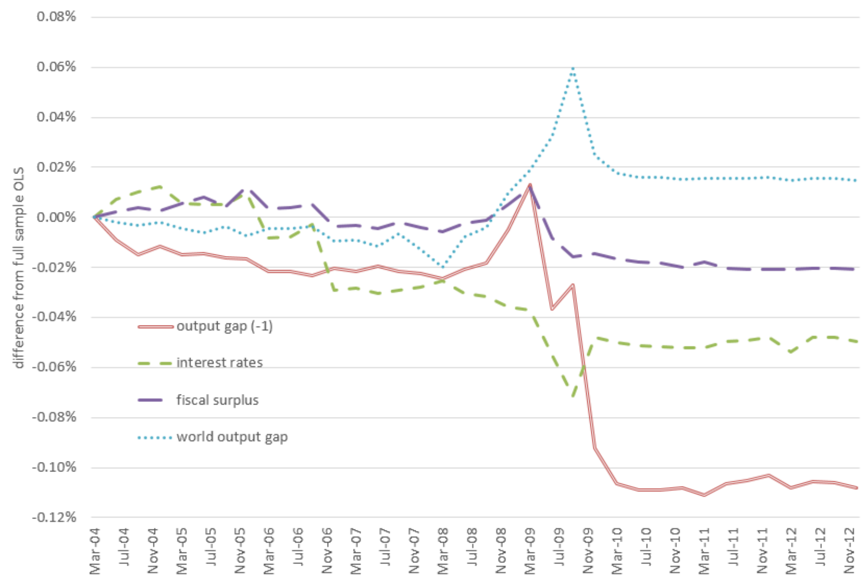

Figure 3.10: Evolution of the difference between learning parameters and benchmark OLS estimates: output gap equation (all parameters except constant)

The response of Central Bank to inflation and output also showed interesting results. Here, we are more interested in the end of our sample, the 2011-2012 period, when Brazil's monetary authority pushed interest rates down to 7.25% (only to be risen again a few months later, in 2013). The evolution of the constant parameter is remarkable: it dropped

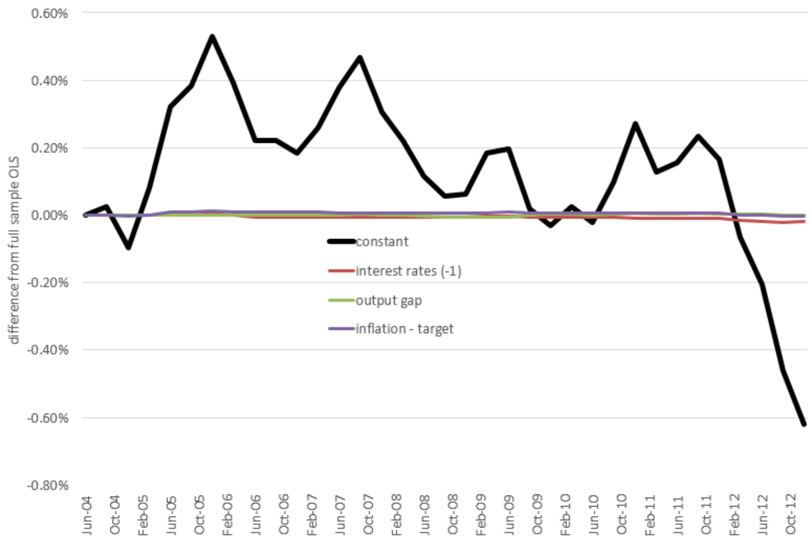

Figure 3.11: Evolution of the difference between learning parameters and benchmark OLS estimates: Taylor rule equation

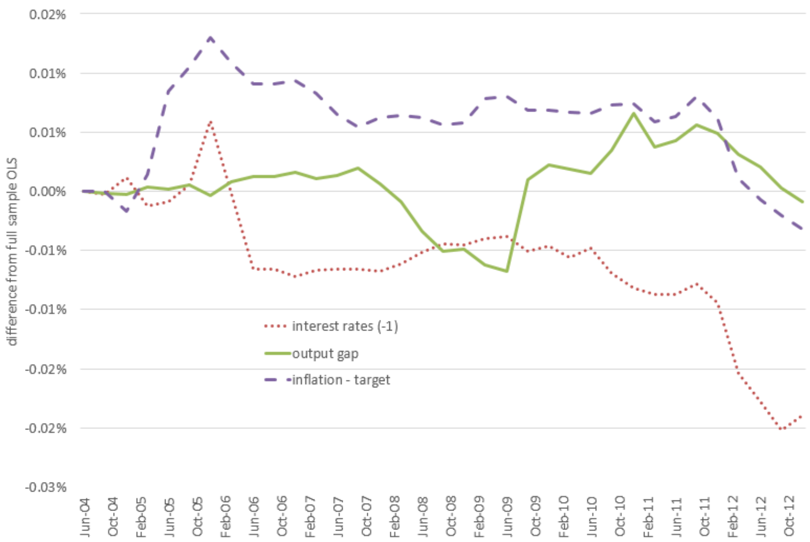

Figure 3.12: Evolution of the difference between learning parameters and benchmark OLS estimates: Taylor rule equation (all parameters except constant)

from a positive difference (in comparison to the OLS estimates) of 0.2% in mid-2011 to a negative difference of 0.6% by the end of 2012. The other coefficients also showed a decline in the 2011-2012 period, specially lagged interest rates and the response to deviations of inflation from its target. This is in line with the view that the Central Bank has become more lenient in the past years, tolerating an inflation rate consistently above the central target in the period.

It is important to reinforce that the results presented here are specific to the reduced-form models presented, and should be taken as evidences of how learning is able to improve the fit of our models, comparatively to the OLS full sample benchmark. Also important is the fact that the learning mechanism is capable of telling a (quantitative) story, describing the evolution of parameters over time. For instance, it may serve as a useful method for quantitatively describing how dovish or hawkish the Central Bank has become, and what were the most important channels that are affecting this behavior.

3.5 Conclusions

In this third chapter we had two main objectives. First, to provide evidences that the supposedly small improvements found in the previous chapters have more to do with the nature of the model at hand than to the learning method itself. To do so, we simulated artificial data using a time-varying structure similar to the one presented in chapter 1 and evaluated the likelihood improvements with different learning mechanisms.

Our simulations (the first group with an independent variable X_t at the right-hand side; the second group with $X_t = Y_{t-1}$) showed that the learning mechanism was able to improve the likelihood (in comparison to their OLS benchmarks) for both groups, but with a remarkable difference in magnitude. The first group of simulations, more closely related to reduced-form models, showed large improvements, while the second group, more related to a solution of a general DSGE model (in which the endogenous lagged variable is often an important explanatory variable), showed much smaller improvements. This result reinforces learning as a relevant empirical mechanism to improve fit, and sheds some light into the expected likelihood improvements one might achieve under learning when estimating different types of models.

The second objective was to apply our reduced-form learning model to assess the empirical relevance of learning in models to forecast inflation, interest rates and output gap for the Brazilian economy in the period 2002-2012. To do so, we inserted learning in *ad-hoc* reduced-form models commonly used by pratictioners. We found evidences of learning in output gap and Taylor rule equations under stochastic gradient learning algorithm: the output gap equation under learning showed an improvement of 20 points in the likelihood, while the Taylor rule equation showed an improvement under learning of 3 points.

Aside from improving the fit of our equations, our learning mechanism allows us to tell a quantitative story of how the importance of explanatory variables evolved over time. This information may be useful, for instance, to: (i) improve forecasts at the margin by using last available estimates of parameters (which is the same idea present in the Kalman Filter) and (ii) improve the description of data and serving as a guide for different specifications of the model (for example, detecting structural breaks in parameters).

4 Conclusions

This thesis focused on the understanding of learning mechanisms applied to a variety of macroeconomic models. The three chapters can be seen as a "gradient" of models, starting from modern structural DSGE models in chapter 1, gradually loosening the link to the microfoundations in chapter 2 and ending with reduced-form models used by praticioners in chapter 3. Hence, rather than three different chapters, there is a common line running thoughout the thesis, which is the relevance of learning mechanisms in improving the fit of estimated macroeconomic models.

Our main contributions were, first, the construction of a set of MATLAB routines that are able to estimate a large variety of Learning DSGE models, so that their performance can be compared easierly. Second, we hope to have helped organizing the debate around alternative types of learning mechanisms. The literature of learning in DSGE models is still on its initial stages and so far research was focused on a rather limited set of learning setups.

Third, we showed evidences of how learning could improve the fit of some types of models. In the first chapter, we showed some of the alternative ways to insert learning in a DSGE model, and the impact it produced in the DSGE dynamics. We simulated artificial data containing a structural break on a specific parameter for two models (one univariate, model 1, and one basic New Keynesian model, model 2), and estimated the models using rational expectations and learning (which we called Learning DSGE model, LDSGE). We also estimated the basic New Keynesian model and a simpler version of Smets & Wouters (2003) (model 3) using real data for Brazil, totaling four sets of data (two simulated with models 1 and 2, and two real, with models 2 and 3). Our learning specifications performed as good as rational expectations for the basic New Keynesian model under simulated data, and performed better than rational expectations for the two other models with real data.

Fourth, the supposedly moderate improvements found in chapter 1 motivated us to explain more deeply the restrictions and limitations of the learning mechanism. We showed two important issues that limited the LDSGE dynamics to fully match the Actual Law of Motion of the economy, and gradually departed from the DSGE restrictzions in order to approximate this true Law of Motion. We first presented a Learning DSGE-VAR model, in which the DSGE memory (the restriction functions of the DSGE-VAR) was still present through the priors generated by the rational expectations DSGE model. Then, we presented the Learning MSV model, in which the memory from the DSGE became even more faded: only the structure of the economy and the initial conditions for the parameters were the same as in the DSGE model. Both estimation procedures showed that learning is a relevant mechanism for improving fit.

Fifth, we then fully departed from the structural DSGE model and focused on reduced-form models in our third chapter, in order to assess if there was empirical evidence of learning for Brazilian data in models used by pratictioners. We used three reduced-form models to forecast output gap, inflation and interest rates for Brazil. The results, which are particular to those models, showed that learning was able to improve the fit by a considerable extent. Also, by updating reduced-form parameters at each period, the learning mechanism is able to tell a quantitative story of, for instance, how some parameter of interest is evolving over time, which may be a very intersting question for policymakers, pratictioners and economists in general.

In our view, the fact that learning is able to encompass rational expectations models as special cases (in which the gain parameter is zero) and the encouraging results presented in all three chapters should serve as arguments in favor of this type of modeling strategy. One might even reverse the common question of "why should we use those learning models?" to "why aren't we using them yet?".

We argue, however, that the learning methods presented here cannot answer to the question of what is the best (or right) way to proceed[71] after estimating a model in which the improvement of fit by the learning mechanism is relevant. This is clearer in DSGE models: if the main goal of inserting learning into the model is to flexibilize rational expectations, allowing agents to forecast variables using time-varying laws of motion, one might promptly proceed with the learning model. However, one might also be interested in answering if the the Calvo-type frictions present in the DSGE model are robust to a hypothesis different from rational expectations. If, in this case, data suggests that some friction is not robust under learning, the alternatives can be (i) proceed with the

[71] In the sense of which model shoud we use, whether with rational expectations or not.

model with the non-robust friction and learning or (ii) search for alternative frictions that are robust to learning. This can be important when one is trying to assess alternative scenarios (counterfactuals) under the requirement that some (or all) structural parameters be policy invariant. Even in reduced-form models, the time-variation of parameters might be an indicative that there could be additional external variables that should be included in the set of explanatory variables of the model. In any case, learning provides *more* information than rational expectations / time-invariant standard models.

5 Appendices

5.1 Appendix A: Detailed tables

Table 5.1: Detailed results of LDSGE estimation for model 1

Spec number	**0**	1	2	3	4	5	6	7	8
Learning Type	**RE**	VAR	VAR	VAR	VAR	VAR c	VAR c	VAR c	VAR c
Initial beliefs		ReC*	ReC*	ReC*	ReC*	ReC*	ReC*	ReC*	ReC*
Learning algorithm		SG	SG	CG	CG	SG	SG	CG	CG
timing		t	t-1	t	t-1	t	t-1	t	t-1
Posterior	**3261.90**	3251.8	3251.8	3251.6	3251.6	3253.1	3253.1	3253.4	3253.4
Marginal likelihood	**3262.07**	3251.5	3251.5	3251.3	3251.3	3252.9	3252.9	3253.3	3253.3
alfa	**0.50**	0.47	0.47	0.47	0.47	0.47	0.47	0.47	0.47
beta	**0.39**	0.45	0.45	0.45	0.45	0.44	0.44	0.42	0.42
rho	**0.03**	0.03	0.03	0.02	0.02	0.03	0.03	0.04	0.04
gain		0.0035	0.0035	0.0094	0.0094	0.0032	0.0032	0.0078	0.0078
Sig_eps	**0.81**	0.75	0.75	0.75	0.75	0.78	0.78	0.80	0.80

Table 5.2: Detailed results of LDSGE estimation for model 1 (cont)

Spec number	**0**	9	10	11	12	13	14	15	16
Learning Type	**RE**	MSV	MSV	MSV	MSV	MSV c	MSV c	MSV c	MSV c
Initial beliefs		ReC*	ReC*	ReC*	ReC*	ReC*	ReC*	ReC*	ReC*
Learning algorithm		SG	SG	CG	CG	SG	SG	CG	CG
timing		t	t-1	t	t-1	t	t-1	t	t-1
Posterior	**3261.90**	3261.9	3251.7	3261.9	3251.1	3261.6	3253.0	3261.7	3252.9
Marginal likelihood	**3262.07**	3262.1	3251.5	3262.1	3251.0	3261.8	3252.9	3261.9	3253.0
alfa	**0.50**	0.50	0.47	0.50	0.47	0.49	0.48	0.49	0.48
beta	**0.39**	0.39	0.45	0.39	0.45	0.40	0.43	0.40	0.42
rho	**0.03**	0.03	0.03	0.03	0.03	0.03	0.04	0.03	0.05
gain		-	0.0037	-	0.0085	0.2000	0.0034	0.1293	0.0080
Sig_eps	**0.81**	0.81	0.74	0.81	0.73	0.79	0.77	0.79	0.78

Table 5.3: Detailed results of LDSGE estimation for model 2, simulated data

Spec number	**0**	1	2	3	4	5	6	7	8
Learning Type	**RE**	VAR	VAR	VAR	VAR	VAR c	VAR c	VAR c	VAR c
Initial beliefs		ReC*	ReC*	ReC*	ReC*	ReC*	ReC*	ReC*	ReC*
Learning algorithm		SG	SG	CG	CG	SG	SG	CG	CG
timing		t	t-1	t	t-1	t	t-1	t	t-1
Posterior	**4504.59**	4647.2	4646.9	4639.5	4595.3	4651.4	4651.4	4610.9	4644.0
Marginal likelihood	**4515.01**	4636.4	4635.6	4637.9	4565.6	4639.2	4639.2	4578.9	4638.1
beta	**0.99**	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
kappa	**0.36**	0.35	0.35	0.43	0.42	0.35	0.35	0.41	0.45
sig	**0.20**	0.00	0.00	0.00	-0.01	0.00	0.00	-0.01	0.00
phi1	**0.95**	1.75	1.77	1.48	2.31	1.80	1.80	2.36	1.63
phi2	**0.54**	0.25	0.26	0.41	0.68	0.27	0.27	0.66	0.44
rho(i)	**0.50**	0.45	0.45	0.57	0.55	0.45	0.45	0.54	0.56
rho1	**0.52**	0.49	0.49	0.51	0.49	0.49	0.49	0.49	0.48
rho2	**0.48**	0.40	0.40	0.41	0.40	0.40	0.40	0.40	0.40
rho3	**0.30**	0.32	0.32	0.32	0.31	0.33	0.33	0.36	0.36
gain		0.0004	0.0004	0.0028	0.0012	0.0004	0.0004	0.0011	0.0022
var1	**0.46**	1.13	1.14	0.87	1.15	1.15	1.15	1.15	0.91
var2	**0.44**	0.98	0.98	0.70	0.95	0.99	0.99	0.96	0.76
var3	**0.48**	0.31	0.31	0.51	0.47	0.31	0.31	0.51	0.52

Table 5.4: Detailed results of LDSGE estimation for model 2, simulated data (cont)

Spec number	**0**	9	10	11	12	13	14	15	16
Learning Type	**RE**	MSV	MSV	MSV	MSV	MSV c	MSV c	MSV c	MSV c
Initial beliefs		ReC*	ReC*	ReC*	ReC*	ReC*	ReC*	ReC*	ReC*
Learning algorithm		SG	SG	CG	CG	SG	SG	CG	CG
timing		t	t-1	t	t-1	t	t-1	t	t-1
Posterior	**4504.59**	4504.6	4501.6	4504.6	4501.2	4504.6	4504.7	4504.6	4504.9
Marginal likelihood	**4515.01**	4515.0	4509.6	4515.0	4508.1	4515.0	4514.9	4515.0	4514.8
beta	**0.99**	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
kappa	**0.36**	0.36	0.38	0.36	0.39	0.36	0.37	0.36	0.37
sig	**0.20**	0.20	0.34	0.20	0.39	0.20	0.22	0.20	0.22
phi1	**0.95**	0.95	1.11	0.95	1.15	0.95	0.97	0.95	1.00
phi2	**0.54**	0.54	0.67	0.54	0.74	0.54	0.56	0.54	0.55
rho(i)	**0.50**	0.50	0.51	0.50	0.51	0.50	0.50	0.50	0.50
rho1	**0.52**	0.52	0.49	0.52	0.49	0.52	0.52	0.52	0.51
rho2	**0.48**	0.48	0.51	0.48	0.52	0.48	0.48	0.48	0.48
rho3	**0.30**	0.30	0.29	0.30	0.29	0.30	0.30	0.29	0.29
gain		-	0.0008	-	0.0022	-	-	-	-
var1	**0.46**	0.46	0.52	0.46	0.53	0.46	0.46	0.46	0.49
var2	**0.44**	0.44	0.51	0.44	0.55	0.44	0.44	0.44	0.45
var3	**0.48**	0.48	0.54	0.48	0.56	0.48	0.49	0.48	0.49

Table 5.5: Detailed results of LDSGE estimation for model 2, real data

Spec number	**0**	1	2	3	4	5	6	7	8
Learning Type	**RE**	VAR	VAR	VAR	VAR	VAR c	VAR c	VAR c	VAR c
Initial beliefs		ReC*	ReC*	ReC*	ReC*	ReC*	ReC*	ReC*	ReC*
Learning algorithm		SG	SG	CG	CG	SG	SG	CG	CG
timing		t	t-1	t	t-1	t	t-1	t	t-1
Posterior	**-508.13**	-434.4	-434.4	-326.9	-349.6	-434.4	-434.4	-397.2	-444.5
Marginal likelihood	**-563.80**	-502.5	-502.5	-390.6	-466.0	-502.5	-502.5	-493.0	-545.1
beta	**0.99**	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
kappa	**0.84**	0.84	0.84	0.82	1.31	0.84	0.84	1.11	0.87
sig	**0.04**	0.06	0.06	0.40	1.26	0.06	0.06	0.38	0.20
phi1	**1.53**	1.53	1.53	1.48	1.06	1.53	1.53	1.04	1.47
phi2	**0.75**	0.75	0.75	0.74	0.79	0.75	0.75	0.72	0.74
rho(i)	**1.01**	1.01	1.01	1.09	1.82	1.01	1.01	1.98	1.15
rho1	**-0.01**	-0.01	-0.01	0.00	0.42	-0.01	-0.01	0.09	0.00
rho2	**-0.04**	-0.05	-0.05	-0.05	0.52	-0.05	-0.05	0.25	-0.09
rho3	**0.84**	0.84	0.84	0.03	0.48	0.84	0.84	0.51	0.58
gain		-	-	-	-	-	-	-	-
var1	**0.04**	0.04	0.04	0.04	0.02	0.04	0.04	0.03	0.04
var2	**0.03**	0.03	0.03	0.03	0.02	0.03	0.03	0.02	0.02
var3	**0.04**	0.03	0.03	0.02	0.02	0.03	0.03	0.03	0.02

Table 5.6: Detailed results of LDSGE estimation for model 2, real data (cont)

Spec number	**0**	9	10	11	12	13	14	15	16
Learning Type	**RE**	MSV	MSV	MSV	MSV	MSV c	MSV c	MSV c	MSV c
Initial beliefs		ReC*	ReC*	ReC*	ReC*	ReC*	ReC*	ReC*	ReC*
Learning algorithm		SG	SG	CG	CG	SG	SG	CG	CG
timing		t	t-1	t	t-1	t	t-1	t	t-1
Posterior	**-508.13**	-508.6	-508.3	-507.3	-507.2	-519.4	-522.3	-508.0	-527.6
Marginal likelihood	**-563.80**	-563.6	-563.8	-563.8	-563.1	-571.6	-576.1	-564.2	-578.0
beta	**0.99**	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
kappa	**0.84**	0.81	0.82	0.84	0.83	0.91	0.93	0.84	0.96
sig	**0.04**	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
phi1	**1.53**	1.51	1.52	1.53	1.53	1.49	1.52	1.53	1.51
phi2	**0.75**	0.75	0.74	0.75	0.75	0.69	0.76	0.75	0.79
rho(i)	**1.01**	1.02	1.02	1.01	1.01	0.95	0.97	1.01	0.89
rho1	**-0.01**	-0.01	-0.01	-0.01	-0.01	0.00	0.00	-0.01	-0.01
rho2	**-0.04**	-0.04	-0.04	-0.04	-0.05	-0.04	-0.04	-0.05	-0.04
rho3	**0.84**	0.84	0.84	0.83	0.83	0.84	0.82	0.83	0.84
gain		0.1000	0.0034	-	-	0.0118	0.0018	-	0.0030
var1	**0.04**	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
var2	**0.03**	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03
var3	**0.04**	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04

Table 5.7: Detailed results of LDSGE estimation for model 3, real data

Model number	**0**	1	2	3	4	5	6	7	8
Learning Type	**RE**	VAR	VAR	VAR	VAR	VAR c	VAR c	VAR c	VAR c
Initial beliefs		ReC*	ReC*	ReC*	ReC*	ReC*	ReC*	ReC*	ReC*
Learning algorithm		SG	SG	CG	CG	SG	SG	CG	CG
timing		t	t-1	t	t-1	t	t-1	t	t-1
Posterior	**-1302.9**	-1264.7	-1192.7	-1266.3	-1254.0	-1168.6	-1168.3	-1253.2	-1244.1
Marginal likelihood	**-1493.4**	-1450.8	-1382.0	-1500.6	-1461.0	-1354.6	-1354.2	-1448.0	-1458.0
beta	**0.98**	0.98	0.98	0.98	0.97	0.98	0.98	0.98	0.98
tau	**0.03**	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03
alfa	**0.45**	0.45	0.45	0.46	0.46	0.45	0.45	0.46	0.46
psi	**7.61**	7.71	7.69	7.61	7.61	7.75	7.75	7.61	7.61
gamma_p	**0.91**	0.91	0.91	0.91	0.91	0.91	0.91	0.91	0.91
gamma_w	**0.54**	0.56	0.55	0.54	0.54	0.56	0.56	0.54	0.54
lambda_w	**0.50**	0.50	0.53	0.50	0.50	0.50	0.50	0.49	0.50
csi_p	**0.98**	0.97	0.97	0.98	0.98	0.97	0.97	0.98	0.98
csi_w	**0.00**	0.01	0.01	0.00	0.00	0.01	0.01	0.00	0.00
sig_L	**4.97**	5.18	5.02	4.97	4.98	5.18	5.18	4.98	4.98
sig_c	**2.08**	1.76	2.10	2.12	2.12	1.76	1.76	2.00	2.11
habit	**0.39**	0.82	0.78	0.70	0.76	0.82	0.82	0.78	0.76
phi_fix	**2.55**	2.51	2.54	2.55	2.55	2.01	2.01	2.55	2.55
phi_adj	**0.01**	0.05	0.05	0.02	0.05	0.05	0.05	0.04	0.02
r_pi	**1.50**	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50
r_y	**0.34**	0.31	0.31	0.34	0.34	0.31	0.31	0.33	0.34
r_dpi	**0.00**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
r_dy	**0.00**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
rho (taylor)	**0.90**	0.90	0.90	0.90	0.90	0.90	0.90	0.89	0.90
rho a	**0.25**	0.37	0.37	0.25	0.25	0.37	0.37	0.27	0.25
rho b	**-0.26**	-0.15	-0.15	-0.24	-0.24	-0.14	-0.14	-0.22	-0.24
rho i	**-0.73**	0.34	0.28	-0.68	0.08	0.34	0.34	0.17	-0.68
rho g	**0.40**	0.41	0.41	0.40	0.40	0.41	0.41	0.40	0.40
rho L	**0.56**	0.50	0.51	0.56	0.56	0.50	0.50	0.53	0.56
rho np	**-0.17**	0.38	0.39	-0.15	-0.10	0.38	0.38	0.17	-0.14
rho nq	**0.37**	0.40	0.41	0.37	0.37	0.40	0.40	0.37	0.37
rho nw	**0.43**	0.42	0.42	0.43	0.43	0.42	0.42	0.43	0.43
rho nr	**0.41**	0.41	0.41	0.41	0.41	0.41	0.41	0.41	0.41
rho nyp	**0.36**	0.36	0.36	0.36	0.36	0.36	0.36	0.36	0.36
rho pim	**0.10**	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10
gain		-	-	-	-	-	-	-	-
var1	**0.04**	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
var2	**0.19**	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
var3	**0.03**	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03
var4	**0.02**	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
var5	**0.04**	0.06	0.06	0.05	0.05	0.06	0.06	0.05	0.05
var6	**0.03**	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03
var7	**0.18**	0.19	0.16	0.19	0.18	0.19	0.19	0.19	0.19
var8	**0.15**	0.15	0.13	0.16	0.16	0.15	0.15	0.16	0.15
var9	**0.02**	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
var10	**0.02**	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
var11	**0.02**	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02

Table 5.8: Detailed results of LDSGE estimation for model 3, real data (cont)

Model number	**0**	9	10	11	12	13	14	15	16
Learning Type	**RE**	MSV	MSV	MSV	MSV	MSV c	MSV c	MSV c	MSV c
Initial beliefs		ReC*	ReC*	ReC*	ReC*	ReC*	ReC*	ReC*	ReC*
Learning algorithm		SG	SG	CG	CG	SG	SG	CG	CG
timing		t	t-1	t	t-1	t	t-1	t	t-1
Posterior	**-1302.9**	-1302.9	-1302.9	-1302.7	-1302.8	-1305.2	-1303.3	-1302.7	-1302.9
Marginal likelihood	**-1493.4**	-1494.3	-1492.1	-1492.8	-1492.5	-1502.2	-1493.9	-1493.2	-1493.3
beta	**0.98**	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98
tau	**0.03**	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03
alfa	**0.45**	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45
psi	**7.61**	7.61	7.61	7.61	7.61	7.54	7.62	7.61	7.61
gamma_p	**0.91**	0.91	0.91	0.91	0.91	0.92	0.91	0.91	0.91
gamma_w	**0.54**	0.54	0.54	0.54	0.54	0.55	0.54	0.54	0.54
lambda_w	**0.50**	0.50	0.50	0.50	0.50	0.53	0.50	0.50	0.50
csi_p	**0.98**	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98
csi_w	**0.00**	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00
sig_L	**4.97**	4.97	4.97	4.97	4.97	4.74	4.97	4.97	4.97
sig_c	**2.08**	2.08	2.08	2.08	2.08	1.04	2.13	2.08	2.09
habit	**0.39**	0.39	0.39	0.40	0.39	0.68	0.39	0.39	0.39
phi_fix	**2.55**	2.55	2.55	2.55	2.55	2.61	2.55	2.55	2.55
phi_adj	**0.01**	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
r_pi	**1.50**	1.50	1.50	1.50	1.50	1.49	1.50	1.50	1.50
r_y	**0.34**	0.34	0.34	0.34	0.34	0.37	0.34	0.34	0.34
r_dpi	**0.00**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
r_dy	**0.00**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
rho (taylor)	**0.90**	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90
rho a	**0.25**	0.25	0.25	0.25	0.25	0.33	0.24	0.25	0.25
rho b	**-0.26**	-0.26	-0.25	-0.26	-0.25	-0.41	-0.25	-0.25	-0.25
rho i	**-0.73**	-0.73	-0.73	-0.73	-0.73	-0.74	-0.73	-0.73	-0.73
rho g	**0.40**	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40
rho L	**0.56**	0.56	0.56	0.56	0.56	0.53	0.56	0.56	0.56
rho np	**-0.17**	-0.17	-0.17	-0.17	-0.17	0.03	-0.17	-0.17	-0.17
rho nq	**0.37**	0.37	0.37	0.37	0.37	0.37	0.37	0.37	0.37
rho nw	**0.43**	0.43	0.43	0.43	0.43	0.42	0.43	0.43	0.43
rho nr	**0.41**	0.41	0.41	0.41	0.41	0.40	0.41	0.41	0.41
rho nyp	**0.36**	0.36	0.36	0.36	0.36	0.35	0.36	0.36	0.36
rho pim	**0.10**	0.10	0.10	0.10	0.10	0.08	0.10	0.10	0.10
gain		-	-	-	-	0.0055	0.0041	-	-
var1	**0.04**	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
var2	**0.19**	0.19	0.19	0.19	0.19	0.24	0.19	0.19	0.19
var3	**0.03**	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03
var4	**0.02**	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
var5	**0.04**	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.04
var6	**0.03**	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03
var7	**0.18**	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18
var8	**0.15**	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15
var9	**0.02**	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
var10	**0.02**	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
var11	**0.02**	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02

Table 5.9: Priors for model 1

Parameter	**Distribution**	**Param1***	**Param2***
α	N	0.5	0.5
β	N	0.3	0.1
ρ	N	0.5	0.5
$gain$	$Unif$	0	0.2
$\Sigma\varepsilon$	IG	5.0	1.0

*For Normal distributions, Param1 stands for mean, Param2 stands for standar deviation; for Uniform distributions, param1 and param2 stands for the interval [param1;param2]; for Inverse Gamma distributions, Param1 stands for shape and Param2 stands for scale (the same for Beta distributions)

Table 5.10: Priors for model 2

Parameter	**Distribution**	**Param1***	**Param2***
β	N	0.99	0.001
κ	N	0.5	0.1
σ	N	0.3	0.5
ϕ_1	N	0.6	0.2
ϕ_2	N	0.6	0.2
ρ_i	N	0.5	0.2
ρ_1	N	0.7	0.4
ρ_2	N	0.5	0.4
ρ_3	N	0.3	0.4
$gain$	$Unif$	0	0.1
Σ_{ξ_1}	IG	3.0	1.0
$\Sigma_{\xi 2}$	IG	3.0	1.0
$\Sigma_{\xi 2}$	IG	3.0	1.0

*For Normal distributions, Param1 stands for mean, Param2 stands for standar deviation; for Uniform distributions, param1 and param2 stands for the interval [param1;param2]; for Inverse Gamma distributions, Param1 stands for shape and Param2 stands for scale (the same for Beta distributions)

Table 5.11: Priors for model 3

Parameter	**Distribution**	**Param1***	**Param2***
β	N	0.98	0.01
τ	N	0.025	0.001
α	N	0.45	0.01
φ_c	N	5.9	1.0
γp	β	2.0	2.0
γw	β	2.0	2.0
λw	β	2.0	2.0
ξp	N	0.5	0.4
ξw	N	0.3	0.4
σL	$Unif$	2.0	0.7
σC	IG	1.3	0.7
h	β	2.0	2.0
ϕ	N	1.5	4.0
φ_a	β	2.0	2.0
$r\pi$	N	1.5	0.5
ry	N	0.3	0.2
$r_{\Delta\pi}$	N	0	0.001
$r_{\Delta y}$	N	0	0.001
ρ	N	0.9	0.03

*For Normal distributions, Param1 stands for mean, Param2 stands for standar deviation; for Uniform distributions, param1 and param2 stands for the interval [param1;param2]; for Inverse Gamma distributions, Param1 stands for shape and Param2 stands for scale (the same for Beta distributions)

Table 5.12: Priors for model 3 (cont)

Parameter	**Distribution**	**Param1***	**Param2***
ρ_a	N	0.5	0.3
ρ_b	N	0.5	0.3
ρ_i	N	0.5	0.3
ρ_g	N	0.5	0.3
ρ_l	N	0.5	0.3
ρ_{np}	N	0.5	0.3
ρ_{nq}	N	0.5	0.3
ρ_{nw}	N	0.5	0.3
ρ_{nr}	N	0.5	0.3
ρ_{nYp}	N	0.5	0.3
$\rho_{\bar{\pi}}$	N	0.5	0.3
$gain$	$Unif$	0	0.1
Σ_a	IG	3.0	1.0
Σ_b	IG	3.0	1.0
Σ_i	IG	3.0	1.0
Σ_g	IG	3.0	1.0
Σ_l	IG	3.0	1.0
Σ_{np}	IG	3.0	1.0
Σ_{nq}	IG	3.0	1.0
Σ_{nw}	IG	3.0	1.0
Σ_{nr}	IG	3.0	1.0
Σ_{nYp}	IG	3.0	1.0
$\Sigma_{\bar{\pi}}$	IG	3.0	1.0

Table 5.13: Likelihoods for LDSGE and LMSV, model 1, simulated data

	Likelihoods			
gain (only LMSV)	LMSV SG	LMSV CG	Best LDSGE (MSV time t)	Best LDSGE (MSV time t-1)
0.000	3262.2	3262.2	3262.1	3251.0
0.001	3260.0	3260.4	3262.1	3251.0
0.002	3257.1	3259.1	3262.1	3251.0
0.003	3254.8	3258.1	3262.1	3251.0
0.004	3253.1	3257.5	3262.1	3251.0
0.005	3251.9	$\boxed{3257.4}$	3262.1	3251.0
0.006	3251.0	3257.6	3262.1	3251.0
0.007	3250.5	3258.1	3262.1	3251.0
0.008	3250.2	3258.9	3262.1	3251.0
0.009	$\boxed{3250.1}$	3259.9	3262.1	3251.0
0.010	3250.1	3261.0	3262.1	3251.0
0.011	3250.4	3262.3	3262.1	3251.0
0.012	3250.7	3263.6	3262.1	3251.0
0.013	3251.1	3265.1	3262.1	3251.0
0.014	3251.7	3266.6	3262.1	3251.0
0.015	3252.4	3268.1	3262.1	3251.0
0.016	3253.1	3269.6	3262.1	3251.0
0.017	3254.0	3271.2	3262.1	3251.0
0.018	3254.9	3272.9	3262.1	3251.0
0.019	3255.9	3274.5	3262.1	3251.0
0.020	3257.0	3276.2	3262.1	3251.0

Table 5.14: Likelihoods for LDSGE and LMSV, model 2, simulated data

	Likelihoods		
gain (only LMSV)	LMSV SG	LMSV CG	Best LDSGE (MSV time t)
0.000	4515.0	4515.0	4515.0
0.001	4519.1	4518.4	4515.0
0.002	4522.6	4521.9	4515.0
0.003	4525.8	4525.3	4515.0
0.004	4528.8	4528.6	4515.0
0.005	4531.7	4531.9	4515.0
0.006	4534.7	4535.1	4515.0
0.007	4537.7	4538.6	4515.0
0.008	4540.8	4542.2	4515.0
0.009	4544.0	4546.1	4515.0
0.010	4547.4	4550.5	4515.0
0.011	4550.8	4555.2	4515.0
0.012	4554.3	4560.5	4515.0
0.013	4557.9	4566.5	4515.0
0.014	4561.6	4573.0	4515.0
0.015	4565.4	4580.2	4515.0
0.016	4569.2	4588.2	4515.0
0.017	4573.1	4596.9	4515.0
0.018	4577.2	4606.3	4515.0
0.019	4581.3	4616.5	4515.0
0.020	4585.5	4627.4	4515.0

Table 5.15: Likelihoods for LDSGE and LMSV, model 3, real data, constant gain

Likelihoods

gain (only LMSV)	LMSV, CG	Best LDSGE (MSV time t)	gain (only LMSV)	LMSV, CG	Best LDSGE (MSV time t)	gain (only LMSV)	LMSV, CG	Best LDSGE (MSV time t)
0.000	-1492.7	-1494.3	0.033	-1495.211	-1494.330	0.067	-1494.419	-1494.330
0.001	-1492.9	-1494.3	0.034	-1495.237	-1494.330	0.068	-1494.343	-1494.330
0.002	-1493.0	-1494.3	0.035	-1495.260	-1494.330	0.069	-1494.264	-1494.330
0.003	-1493.1	-1494.3	0.036	-1495.279	-1494.330	0.070	-1494.181	-1494.330
0.004	-1493.2	-1494.3	0.037	-1495.296	-1494.330	0.071	-1494.096	-1494.330
0.005	-1493.3	-1494.3	0.038	-1495.311	-1494.330	0.072	-1494.008	-1494.330
0.006	-1493.4	-1494.3	0.039	-1495.322	-1494.330	0.073	-1493.916	-1494.330
0.007	-1493.5	-1494.3	0.040	-1495.330	-1494.330	0.074	-1493.822	-1494.330
0.008	-1493.6	-1494.3	0.041	-1495.335	-1494.330	0.075	-1493.724	-1494.330
0.009	-1493.7	-1494.3	0.042	-1495.337	-1494.330	0.076	-1493.624	-1494.330
0.010	-1493.8	-1494.3	0.043	-1495.336	-1494.330	0.077	-1493.520	-1494.330
0.011	-1493.9	-1494.3	0.044	-1495.333	-1494.330	0.078	-1493.413	-1494.330
0.012	-1494.0	-1494.3	0.045	-1495.326	-1494.330	0.079	-1493.303	-1494.330
0.013	-1494.1	-1494.3	0.046	-1495.317	-1494.330	0.080	-1493.191	-1494.330
0.014	-1494.2	-1494.3	0.047	-1495.304	-1494.330	0.081	-1493.075	-1494.330
0.015	-1494.3	-1494.3	0.048	-1495.288	-1494.330	0.082	-1492.956	-1494.330
0.016	-1494.3	-1494.3	0.049	-1495.270	-1494.330	0.083	-1492.834	-1494.330
0.017	-1494.4	-1494.3	0.050	-1495.248	-1494.330	0.084	-1492.709	-1494.330
0.018	-1494.5	-1494.3	0.051	-1495.224	-1494.330	0.085	-1492.581	-1494.330
0.019	-1494.5	-1494.3	0.052	-1495.196	-1494.330	0.086	-1492.450	-1494.330
0.020	-1494.6	-1494.3	0.053	-1495.165	-1494.330	0.087	-1492.316	-1494.330
0.021	-1494.7	-1494.3	0.054	-1495.132	-1494.330	0.088	-1492.179	-1494.330
0.022	-1494.7	-1494.3	0.055	-1495.095	-1494.330	0.089	-1492.039	-1494.330
0.023	-1494.8	-1494.3	0.056	-1495.055	-1494.330	0.090	-1491.896	-1494.330
0.024	-1494.8	-1494.3	0.057	-1495.013	-1494.330	0.091	-1491.750	-1494.330
0.025	-1494.9	-1494.3	0.058	-1494.967	-1494.330	0.092	-1491.601	-1494.330
0.026	-1494.9	-1494.3	0.059	-1494.918	-1494.330	0.093	-1491.449	-1494.330
0.027	-1495.0	-1494.3	0.060	-1494.867	-1494.330	0.094	-1491.294	-1494.330
0.028	-1495.0	-1494.3	0.061	-1494.812	-1494.330	0.095	-1491.136	-1494.330
0.029	-1495.1	-1494.3	0.062	-1494.754	-1494.330	0.096	-1490.975	-1494.330
0.030	-1495.1	-1494.3	0.063	-1494.693	-1494.330	0.097	-1490.811	-1494.330
0.031	-1495.2	-1494.3	0.064	-1494.629	-1494.330	0.098	-1490.644	-1494.330
0.032	-1495.2	-1494.3	0.065	-1494.562	-1494.330	0.099	-1490.475	-1494.330
0.033	-1495.2	-1494.3	0.066	-1494.492	-1494.330	0.100	-1490.302	-1494.330

Table 5.16: Likelihoods for LDSGE and LMSV, model 3, real data, stochastic gradient

Likelihoods

gain (only LMSV)	LMSV, SG	Best LDSGE (MSV time t)	gain (only LMSV)	LMSV, SG	Best LDSGE (MSV time t)	gain (only LMSV)	LMSV, SG	Best LDSGE (MSV time t)
0.000	-1492.7	-1494.3	0.280	-1494.385	-1494.330	0.570	-1494.453	-1494.330
0.010	-1492.8	-1494.3	0.290	-1494.416	-1494.330	0.580	-1494.424	-1494.330
0.020	-1492.9	-1494.3	0.300	-1494.445	-1494.330	0.590	-1494.393	-1494.330
0.030	-1493.0	-1494.3	0.310	-1494.472	-1494.330	0.600	-1494.360	-1494.330
0.040	-1493.1	-1494.3	0.320	-1494.497	-1494.330	0.610	-1494.324	-1494.330
0.050	-1493.1	-1494.3	0.330	-1494.520	-1494.330	0.620	-1494.287	-1494.330
0.060	-1493.2	-1494.3	0.340	-1494.541	-1494.330	0.630	-1494.247	-1494.330
0.070	-1493.3	-1494.3	0.350	-1494.560	-1494.330	0.640	-1494.206	-1494.330
0.080	-1493.4	-1494.3	0.360	-1494.577	-1494.330	0.650	-1494.162	-1494.330
0.090	-1493.4	-1494.3	0.370	-1494.592	-1494.330	0.660	-1494.116	-1494.330
0.100	-1493.5	-1494.3	0.380	-1494.605	-1494.330	0.670	-1494.068	-1494.330
0.110	-1493.6	-1494.3	0.390	-1494.615	-1494.330	0.680	-1494.018	-1494.330
0.120	-1493.6	-1494.3	0.400	-1494.624	-1494.330	0.690	-1493.966	-1494.330
0.130	-1493.7	-1494.3	0.410	-1494.631	-1494.330	0.700	-1493.912	-1494.330
0.140	-1493.7	-1494.3	0.420	-1494.635	-1494.330	0.710	-1493.855	-1494.330
0.150	-1493.8	-1494.3	0.430	$\boxed{-1494.638}$	-1494.330	0.720	-1493.797	-1494.330
0.160	-1493.9	-1494.3	0.440	-1494.638	-1494.330	0.730	-1493.736	-1494.330
0.170	-1493.9	-1494.3	0.450	-1494.636	-1494.330	0.740	-1493.674	-1494.330
0.180	-1494.0	-1494.3	0.460	-1494.633	-1494.330	0.750	-1493.609	-1494.330
0.190	-1494.0	-1494.3	0.470	-1494.627	-1494.330	0.760	-1493.542	-1494.330
0.200	-1494.1	-1494.3	0.480	-1494.619	-1494.330	0.770	-1493.473	-1494.330
0.210	-1494.1	-1494.3	0.490	-1494.609	-1494.330	0.780	-1493.402	-1494.330
0.220	-1494.2	-1494.3	0.500	-1494.597	-1494.330	0.790	-1493.330	-1494.330
0.230	-1494.2	-1494.3	0.510	-1494.583	-1494.330	0.800	-1493.254	-1494.330
0.240	-1494.2	-1494.3	0.520	-1494.566	-1494.330			
0.250	-1494.3	-1494.3	0.530	-1494.548	-1494.330			
0.260	-1494.3	-1494.3	0.540	-1494.527	-1494.330			
0.270	-1494.4	-1494.3	0.550	-1494.505	-1494.330			
0.280	-1494.4	-1494.3	0.560	-1494.480	-1494.330			

5.2 Appendix B: Solving DSGE models

In this section we briefly review how to solve a linear rational expectations model, based on Blanchard & Kahn (1980). Some alternative solutions are Anderson & Moore (1983, 1985), Sims (1996), King & Watson (2002), Uhlig (1999), among others. An interesting comparison of these solution techniques and their computational performance can be found in Anderson (2008). Consider an economic model given by:

$$A_0 E_t \begin{vmatrix} w_{t+1} \\ y_{t+1} \end{vmatrix} = A_1 \begin{vmatrix} w_t \\ y_t \end{vmatrix} + B_0 \begin{vmatrix} \varepsilon_{1,t+1} \\ \varepsilon_{2,t+1} \end{vmatrix}, \tag{5.1}$$

where y_t denotes a vector of endogenous variables, w_t is a vector of exogenous or predetermined variables and $\varepsilon_{i,t}$ are innovations[72]. Matrices A_0, A_1 and B_0 are functions of structural parameters of the economy (θ), but we suppress this dependence for now. We seek a solution to this system, which is given by two policy functions: one that relates the endogenous variables to the predetermineds, and other that describes the evolution of the predetermineds over time. If A_0 is invertible[73], then we can rewrite (5.1) as:

$$\begin{aligned} A_0^{-1} A_0 E_t \begin{vmatrix} w_{t+1} \\ y_{t+1} \end{vmatrix} &= A_0^{-1} A_1 \begin{vmatrix} w_t \\ y_t \end{vmatrix} + A_0^{-1} B_0 \begin{vmatrix} \varepsilon_{1,t+1} \\ \varepsilon_{2,t+1} \end{vmatrix} \\ E_t \begin{vmatrix} w_{t+1} \\ y_{t+1} \end{vmatrix} &= A \begin{vmatrix} w_t \\ y_t \end{vmatrix} + B \begin{vmatrix} \varepsilon_{1,t+1} \\ \varepsilon_{2,t+1} \end{vmatrix} \end{aligned} \tag{5.2}$$

Jordan decomposition allow us to write $A = PDP^{-1}$, where D is a diagonal matrix containing eigenvalues and P is a matrix with its associated eigenvectors. Premultiplying the system by P^{-1} and denoting:

$$P^{-1} = \begin{vmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{vmatrix}, D = \begin{vmatrix} D_1 & 0 \\ 0 & D_2 \end{vmatrix}, P^{-1}B = R = \begin{vmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{vmatrix},$$

[72] The difference between shocks and innovations is that the latter are assumed iid and therefore their expected value is always zero. Shocks, in the other hand, can have an autorregressive component. In our specification, the shocks would be put in the predetermined vector.

[73] If A_0 is not invertible we can use Klein's (2000) method of generalized Schur decomposition to solve the model. Here we assume that A_0 is invertible for the sake of exposition.

we can write (5.2) as:

$$\begin{aligned}
P^{-1}E_t \begin{vmatrix} w_{t+1} \\ y_{t+1} \end{vmatrix} &= DP^{-1} \begin{vmatrix} w_t \\ y_t \end{vmatrix} + P^{-1}B \begin{vmatrix} \varepsilon_{1,t+1} \\ \varepsilon_{2,t+1} \end{vmatrix} \\
\begin{vmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{vmatrix} \begin{vmatrix} w_{t+1} \\ y_{t+1} \end{vmatrix} &= \begin{vmatrix} D_1 & 0 \\ 0 & D_2 \end{vmatrix} \begin{vmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{vmatrix} \begin{vmatrix} w_t \\ y_t \end{vmatrix} + \begin{vmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{vmatrix} \begin{vmatrix} \varepsilon_{1,t+1} \\ \varepsilon_{2,t+1} \end{vmatrix}, \qquad (5.3)
\end{aligned}$$

where the expectations operator has been ommitted for convenience. Now we define two variables:

$$\widetilde{w}_{t+1} = P_{11}w_{t+1} + P_{12}y_{t+1} \qquad (5.4)$$

$$\widetilde{y}_{t+1} = P_{21}w_{t+1} + P_{22}y_{t+1} \qquad (5.5)$$

So we can rewrite the system above as:

$$\begin{vmatrix} \widetilde{w}_{t+1} \\ \widetilde{y}_{t+1} \end{vmatrix} = \begin{vmatrix} D_1 & 0 \\ 0 & D_2 \end{vmatrix} \begin{vmatrix} \widetilde{w}_t \\ \widetilde{y}_t \end{vmatrix} + \begin{vmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{vmatrix} \begin{vmatrix} \varepsilon_{1,t+1} \\ \varepsilon_{2,t+1} \end{vmatrix}, \qquad (5.6)$$

which is now decoupled in respect to "tilda" variables. If the eigenvalues in D_1 are stable and the eigenvalues of D_2 are unstable, the system satisfies the so-called Blanchard & Kahn conditions (the number of unstable eigenvalues is equal to the number of control/endogenous non-predetermined variables), and we can find its solution by backward and forward iteration, respectively.

Let's start with the unstable equation (5.6). Iterate both sides forward, take date t expectations, using the fact that $E_t\varepsilon_{2,t+j} = E_t\varepsilon_{1,t+j} = 0$ for $j > 0$:

$$\begin{aligned}
E_t\widetilde{y}_{t+2} &= (D_2)^2\widetilde{y}_t + E_t(D_2)(R_{11}\varepsilon_{1,t+2} + R_{12}\varepsilon_{2,t+2}) + E_t(R_{11}\varepsilon_{1,t+1} + R_{12}\varepsilon_{2,t+1}) \\
E_t\widetilde{y}_{t+3} &= (D_2)^3\widetilde{y}_t + E_t(D_2)^2(R_{11}\varepsilon_{1,t+3} + R_{12}\varepsilon_{2,t+3}) + 0 \\
&\ldots \\
E_t\widetilde{y}_{t+s} &= (D_2)^s\widetilde{y}_t \qquad (5.7)
\end{aligned}$$

As D_2 contains "explosive" eigenvalues, i.e., elements greater than one, this equality holds at all dates only if $\widetilde{y}_t = 0$, for all t. So, by the definition of $\widetilde{y}_t$, we have:

$$
\begin{aligned}
\widetilde{y}_{t+1} &= P_{21}w_{t+1} + P_{22}y_{t+1} \\
0 &= P_{21}w_{t+1} + P_{22}y_{t+1} \\
y_{t+1} &= -P_{22}^{-1}P_{21}w_{t+1} \qquad (5.8)
\end{aligned}
$$

We now turn to the stable sub-system, substituting (5.8) into (5.4) to get:

$$
\begin{aligned}
\widetilde{w}_{t+1} &= P_{11}w_{t+1} + P_{12}y_{t+1} \\
\widetilde{w}_{t+1} &= P_{11}w_{t+1} - P_{12}P_{22}^{-1}P_{21}w_{t+1} \\
\widetilde{w}_{t+1} &= (P_{11} - P_{12}P_{22}^{-1}P_{21})w_{t+1}, \qquad (5.9)
\end{aligned}
$$

and then into (5.6):

$$
\begin{aligned}
\widetilde{w}_{t+1} &= D_1\widetilde{w}_t + R_1\varepsilon_{1,t+1} \\
(P_{11} - P_{12}P_{22}^{-1}P_{21})w_{t+1} &= D_1(P_{11} - P_{12}P_{22}^{-1}P_{21})w_t + R_1\varepsilon_{1,t+1}, \qquad (5.10)
\end{aligned}
$$

which yields:

$$
\begin{aligned}
w_{t+1} = (P_{11} - P_{12}P_{22}^{-1}P_{21})^{-1}&D_1(P_{11} - P_{12}P_{22}^{-1}P_{21})w_t \\
&+ (P_{11} - P_{12}P_{22}^{-1}P_{21})^{-1}R_1\varepsilon_{1,t+1} \qquad (5.11)
\end{aligned}
$$

Equations (5.11) and (5.8) represent the solution to our system. It relates the endogenous variables to the predetermined, $y_t = f(w_t)$, and gives a law of motion for the predetermined, $w_t = g(w_{t-1}, \varepsilon_t)$.

The model presented below can be augmented to include lagged variables. Suppose that our economic model is now given by:

$$
\begin{vmatrix} A_{01} & A_{02} \\ A_{03} & A_{04} \end{vmatrix} E_t \begin{vmatrix} w_{t+1} \\ y_{t+1} \end{vmatrix} = \begin{vmatrix} A_{11} & A_{12} \\ A_{13} & A_{14} \end{vmatrix} \begin{vmatrix} w_t \\ y_t \end{vmatrix} + \begin{vmatrix} A_{21} & A_{22} \\ A_{23} & A_{24} \end{vmatrix} \begin{vmatrix} w_{t-1} \\ y_{t-1} \end{vmatrix} + \begin{vmatrix} B_{01} & B_{02} \\ B_{03} & B_{04} \end{vmatrix} \begin{vmatrix} \varepsilon_{1,t+1} \\ \varepsilon_{2,t+1} \end{vmatrix} \qquad (5.12)
$$

We can rewrite this system as:

$$\begin{vmatrix} A_{01} & 0 & A_{02} & 0 \\ 0 & I & 0 & 0 \\ A_{03} & 0 & A_{04} & 0 \\ 0 & 0 & 0 & I \end{vmatrix} E_t \begin{vmatrix} w_{t+1} \\ w_t \\ y_{t+1} \\ y_t \end{vmatrix} = \begin{vmatrix} A_{11} & A_{21} & A_{12} & A_{22} \\ I & 0 & 0 & 0 \\ A_{13} & A_{23} & A_{14} & A_{24} \\ 0 & 0 & I & 0 \end{vmatrix} \begin{vmatrix} w_t \\ w_{t-1} \\ y_t \\ y_{t-1} \end{vmatrix} + \begin{vmatrix} B_{01} & 0 & B_{02} & 0 \\ 0 & 0 & 0 & 0 \\ B_{03} & 0 & B_{04} & 0 \\ 0 & 0 & 0 & 0 \end{vmatrix} \begin{vmatrix} \varepsilon_{1,t+1} \\ 0 \\ \varepsilon_{2,t+1} \\ 0 \end{vmatrix}, \quad (5.13)$$

where we defined a new vector of variables containing w_{t+1}, w_t y_{t+1} and y_t, and added some zero and identity matrices of appropriate dimensions. This system can be represented as:

$$\overline{A_0} E_t \begin{vmatrix} \overline{w}_{t+1} \\ \overline{y}_{t+1} \end{vmatrix} = \overline{A_1} \begin{vmatrix} \overline{w}_t \\ \overline{y}_t \end{vmatrix} + \overline{B_0} \begin{vmatrix} \overline{\varepsilon}_{1,t+1} \\ \overline{\varepsilon}_{2,t+1} \end{vmatrix}, \quad (5.14)$$

which is similar to the one studied in (5.1).

When A0 is not invertible, we can use the generalized Schur decomposition (or QZ decomposition) to find a solution to the system. We rewrite our problem as:

$$A_0 E_t \begin{vmatrix} w_{t+1} \\ y_{t+1} \end{vmatrix} = A_1 \begin{vmatrix} w_t \\ y_t \end{vmatrix} + B_0 \begin{vmatrix} \varepsilon_{1,t+1} \\ \varepsilon_{2,t+1} \end{vmatrix} \quad (5.15)$$

The QZ decomposition allows us to write $A_0 = QSZ, A_1 = QTZ$, where Q and Z are unitary[74] matrices and S, T are upper triangular matrices. So we have:

$$QSZE_t \begin{vmatrix} w_{t+1} \\ y_{t+1} \end{vmatrix} = QTZ \begin{vmatrix} w_t \\ y_t \end{vmatrix} + B_0 \begin{vmatrix} \varepsilon_{1,t+1} \\ \varepsilon_{2,t+1} \end{vmatrix} \quad (5.16)$$

Premultiplying by Q', ommiting the expectations operator for convenience and denoting $Q'B_0 = R$, similar to the Blanchard-Kahn example, we get:

[74] A square matrix Q is unitary or orthogonal if $QQ' = Q'Q = I_Q$.

$$
\begin{aligned}
Q'QSZ\begin{vmatrix} w_{t+1} \\ y_{t+1} \end{vmatrix} &= Q'QTZ\begin{vmatrix} w_t \\ y_t \end{vmatrix} + Q'B_0\begin{vmatrix} \varepsilon_{1,t+1} \\ \varepsilon_{2,t+1} \end{vmatrix} \\
S\begin{vmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{vmatrix}\begin{vmatrix} w_{t+1} \\ y_{t+1} \end{vmatrix} &= T\begin{vmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{vmatrix}\begin{vmatrix} w_t \\ y_t \end{vmatrix} + \begin{vmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{vmatrix}\begin{vmatrix} \varepsilon_{1,t+1} \\ \varepsilon_{2,t+1} \end{vmatrix} \qquad (5.17)
\end{aligned}
$$

Denoting $\widetilde{w}_t = Z_{11}w_t + Z_{12}y_t$ and $\widetilde{y}_t = Z_{21}w_t + Z_{22}y_t$:

$$
\begin{aligned}
S\begin{vmatrix} \widetilde{w}_{t+1} \\ \widetilde{y}_{t+1} \end{vmatrix} &= T\begin{vmatrix} \widetilde{w}_t \\ \widetilde{y}_t \end{vmatrix} + \begin{vmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{vmatrix}\begin{vmatrix} \varepsilon_{1,t+1} \\ \varepsilon_{2,t+1} \end{vmatrix} \\
\begin{vmatrix} S_{11} & S_{12} \\ 0 & S_{22} \end{vmatrix}\begin{vmatrix} \widetilde{w}_{t+1} \\ \widetilde{y}_{t+1} \end{vmatrix} &= \begin{vmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{vmatrix}\begin{vmatrix} \widetilde{w}_t \\ \widetilde{y}_t \end{vmatrix} + \begin{vmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{vmatrix}\begin{vmatrix} \varepsilon_{1,t+1} \\ \varepsilon_{2,t+1} \end{vmatrix}, \qquad (5.18)
\end{aligned}
$$

which denotes two decoupled systems, one stable and other, given by:

$$
\begin{aligned}
S_{11}\widetilde{w}_{t+1} + S_{12}\widetilde{y}_{t+1} &= T_{11}\widetilde{w}_t + T_{12}\widetilde{y}_t + R_{11}\varepsilon_{1,t+1} + R_{12}\varepsilon_{2,t+1} \qquad (5.19) \\
S_{22}\widetilde{y}_{t+1} &= T_{22}\widetilde{y}_t + R_{21}\varepsilon_{2,t+1} + R_{22}\varepsilon_{2,t+1} \qquad (5.20)
\end{aligned}
$$

As before, the solution to the unstable system implies that $\widetilde{y}_t = 0$ for all t. So, our solution in terms of the original variables is given by:

$$
\begin{aligned}
\widetilde{y}_t &= 0 \\
0 &= Z_{21}w_t + Z_{22}y_t \\
y_t &= -Z_{22}^{-1}Z_{21}w_t, \qquad (5.21)
\end{aligned}
$$

and (denoting $R_{11}\varepsilon_{1,t+1} + R_{12}\varepsilon_{2,t+1} = v_{t+1}$):

$$\begin{aligned}
S_{11}\widetilde{w}_{t+1} + S_{12}\widetilde{y}_{t+1} &= T_{11}\widetilde{w}_t + T_{12}\widetilde{y}_t + R_{11}\varepsilon_{1,t+1} + R_{12}\varepsilon_{2,t+1} \\
S_{11}\widetilde{w}_{t+1} &= T_{11}\widetilde{w}_t + v_{t+1} \\
\widetilde{w}_{t+1} &= S_{11}^{-1}T_{11}\widetilde{w}_t + S_{11}^{-1}v_{t+1} \\
(Z_{11}w_{t+1} + Z_{12}y_{t+1}) &= S_{11}^{-1}T_{11}(Z_{11}w_t + Z_{12}y_t) + S_{11}^{-1}v_{t+1} \\
(Z_{11}w_{t+1} + Z_{12}(-Z_{22}^{-1}Z_{21}w_{t+1})) &= S_{11}^{-1}T_{11}(Z_{11}w_t + Z_{12}(-Z_{22}^{-1}Z_{21}w_t)) + S_{11}^{-1}v_{t+1} \\
(Z_{11} - Z_{12}Z_{22}^{-1}Z_{21})w_{t+1} &= S_{11}^{-1}T_{11}(Z_{11} - Z_{12}Z_{22}^{-1}Z_{21})w_t + S_{11}^{-1}v_{t+1},
\end{aligned}$$

which yields:

$$w_{t+1} = (Z_{11} - Z_{12}Z_{22}^{-1}Z_{21})^{-1}S_{11}^{-1}[T_{11}(Z_{11} - Z_{12}Z_{22}^{-1}Z_{21})w_t + v_{t+1}] \tag{5.22}$$

Equations (5.21) and (5.22) are the solution of our model, equivalent to (5.8) and (5.11). So, we are able to solve the system even with a non-invertible matrix A_0. After this step, we put this solution in a state-space format and proceed to (bayesian) estimation of its parameters:

$$\alpha_t + \mu + T\alpha_{t-1} + R\xi_t = 0 \tag{5.23}$$

$$Y_t^{obs} - H\alpha_t = 0 \tag{5.24}$$

where α_t is a vector of states, Y_t^{obs} is a vector of observables, H is a selection matrix which maps α_t into Y_t^{obs}, ξ_t are innovations, and matrices μ, T and R are functions of structural parameters θ. It is important to notice that this state-space representation is time-invariant, i.e., μ, T and R are not indexed by t. As will be discussed in the following section, this will be a crucial difference between learning and RE models.

5.3 Appendix C: Likelihood function

This section reviews the construction and evaluation of the likelihood function of a time-varying state-space model given by (1.30) and (1.31). For simplicity, we rewrite the system as:

$$\alpha_t = \mu_t + T_t\alpha_{t-1} + R_t\xi_t \tag{5.25}$$

$$Y_t^{obs} = H\alpha_t + \zeta_{2,t} \tag{5.26}$$

where we change signs of some matrices if necessary. Equation (5.26) is the measurement equation (added with measurement errors) and (5.25) is the transition equation. Also, we define $\xi_t \sim N(0, \Omega_T)$ and $\zeta_{2t} \sim N(0, \Omega_M)$. Note that μ_t, T_t and R_t are allowed to be time-varying. So, at each period, we will need to specify how these matrices evolve, in addition to the usual Kalman Filter steps.

What we want here is, given initial conditions and structural parameters, evaluate the likelihood function L_t using the (modified) Kalman Filter. So, we start by giving initial conditions to: (i) the state vector and its associated variance, (α_0, P_0); (ii) the reduced-form parameters, (β_0, R_0). We define:

$$\alpha_{1|0} = \alpha_0 \tag{5.27}$$

$$P_{1|0} = P_0 \tag{5.28}$$

$$\beta_0 = \beta_0 \tag{5.29}$$

$$R_0 = R_0 \tag{5.30}$$

With these in hand, we can start the usual Kalman filter steps, given by calculating the forecast error for Y_1^{obs} and some auxiliary matrices:

$$v_1 = Y_1^{obs} - H\alpha_{1|0}, \tag{5.31}$$

$$F_1 = HP_1H' + \Omega_M \tag{5.32}$$

$$M_1 = P_1H', \tag{5.33}$$

where the first equation gives the forecast error of the observables in period 1, given[75] Y_0, and the other equations produce auxiliary matrices. After this, Y_1^{obs} is observed and we proceed to the state vector filtering to produce $\alpha_{t|t}$.

$$\alpha_{1|1} = \alpha_{1|0} + M_1^{-1} F_1 v_1 \tag{5.34}$$

$$P_{1|1} = P_{1|0} - M_1^{-1} F_1 M_1', \tag{5.35}$$

The next step of the Kalman filter would be using equation (5.25) to forecast the state vector for $t = 2$, $\alpha_{2|1}$. However, to do so, it is required that we know the reduced-form parameters in time 2. It is exactly here where the usual evaluation of the likelihood function via the Kalman Filter is changed to accomodate for the time-varying parameters. We use a gain sequence g_t to update β_t and R_t over time:

$$\beta_2 = \beta_1 + g_t R_2^{-1} Z_1 (y_1 - \beta_1' Z_1) \tag{5.36}$$

$$R_2 = R_1 + g_t (Z_1 Z_1' - R_1), \tag{5.37}$$

We can now proceed to the prediction step, given by:

$$\alpha_2 = \alpha_{2|1} = \mu_2 + T_2 \alpha_{1|1} \tag{5.38}$$

$$P_2 = P_{2|1} = T_t P_{1|1} T_t' + R_1 \Omega_T R_1, \tag{5.39}$$

and to the filtering of observable variables:

$$Y_{1,f}^{obs} = H \alpha_{1|1} \tag{5.40}$$

This conclude the Kalman Filter equations for $t = 1$. To summarize our results, we

[75] Here, given Y_0 denotes that the underlying model to forecast Y_1 includes information until Y_t, with $t = 0$.

show the recursive equations for time $t \geq 2$:

$$
\begin{aligned}
v_t &= Y_t^{obs} - H\alpha_{t|t-1}, && (5.41)\\
F_t &= HP_{t|t-1}H' + \Omega_M && (5.42)\\
M_t &= P_t H' && (5.43)\\
\alpha_{t|t} &= \alpha_{t|t-1} + M_t^{-1}F_t v_t && (5.44)\\
P_{t|t} &= P_{t|t-1} - M_t^{-1}F_t M_t' && (5.45)\\
\beta_t &= \beta_{t-1} + g_t R_t^{-1} Z_{t-1}(y_{t-1} - \beta_{t-1}' Z_{t-1}) && (5.46)\\
R_t &= R_t + g_t(Z_{t-1}Z_{t-1}' - R_{t-1}) && (5.47)\\
\alpha_{t+1|t} &= \mu_t + T_t\alpha_{t|t} && (5.48)\\
P_{t+1|t} &= T_t P_{1|1} T_t' + R_t \Omega_T R_t && (5.49)
\end{aligned}
$$

To implement equations (5.41) to (5.49) numerically and evaluate the likelihood function, we need to fully specify: the matrices (H, T_t, R_t, Ω_M), which contain the strucutral parameters of the model; the initial conditions $(\alpha_0, P_0, \beta_0, R_0)$, and the gain sequence g_t. The log-likelihood function is then recursely given by:

$$L_t = \text{constant} + \log(\det(F_t)) + v_t' F_t^{-1} v_t \tag{5.50}$$

5.4 Appendix D: Bayesian estimation of DSGE models

We briefly review some of the required steps to estimate the model described by the state-space system (5.25) and (5.26). A good introduction of Bayesian econometrics can be found in Koop (2007) and Congdon (2006).

As usual, we state Bayes' rule:

$$P(A,B) = P(A|B)P(B), \tag{5.51}$$

so it's also valid that

$$P(B,A) = P(B|A)P(A), \tag{5.52}$$

and, by equating the former equations:

$$P(A|B) = \frac{P(A|B)P(B)}{P(B)} \tag{5.53}$$

In our case, $A = \phi$, the structural parameters, and $B = Y$, our data. So, we get:

$$P(\phi|B) = \frac{P(Y|\phi)P(\phi)}{P(Y)}, \tag{5.54}$$

The term given by $P(Y)$ describes the unconditional probability of our data, and can be seen as a proportionality constant. So, can write (5.54) as:

$$P(\phi|Y) \propto P(Y|\phi)P(\phi), \tag{5.55}$$

where the symbol $\propto$ indicates that the left-side term (called the posterior distribution) is proportional to the product of the two right-side terms, the likelihood $P(Y|\phi)$ and the prior $P(\phi)$.

The goal of bayesian econometrics is to find the posterior density of the parameter vector ϕ. Then, one can calculate usual statistics (mean, mode, standard deviation), make point forecasts and calculate bayesian confidence intervals (also called credible interval or region). This is done by (i) finding the vector $\phi_{\max}$ that maximizes the posterior density and (ii) simulating an abritrarely large quantity of draws from some candidate distribution that gives us the distribution of the structural parameters.

The first step can be done by "brute force", which involves simulating a sufficiently large quantity of parameter vectors and selecting the one that produces the maximal posterior density, or by a search algorithm, which finds the maximum using derivative-based methods, such as csminwel, developed by Sims.

The second step involves simulating from a candidate distribution that replicates the behavior of the unknown posterior density. A well know candidate comes from the Metropolis-Hastings Random Walk algorithm, which simulates from a (multivariate) Normal distribution with time-varying mean and variance. Once enough simulations are made (in general, hundreds of thousands), the initial simulations are excluded from the sample and the resulting draws make up the distribution of the parameters, from which we can obtain statistics (mode, mean, median, standard deviations, etc.).

5.5 Appendix E: Guide to our MATLAB program

Our software is a set of MATLAB routines designed to run learning models. The main two m-files are given by:

1) **LDM.m**, which is our main m-file and requires seven inputs: (1) in_model, the number of the model (each model is assigned to a folder "modelX" where X stands for the model number; (2) in_data, the data type (simulated or real data); (3) in_sims, the number of simulations in the MCMC steps; (4) $learning\_type$, the learning type (VAR, VARc, MSV, MSVc); (5) $ini\_beliefs$, the type of initial beliefs; (6) $learning\_algo$, the learning algorithm (SG, CG); (7) $timing\_PLM$, the PLM timing (t or $t-1$). Inside the m-file there is a clear description of the inputs and the values assigned to each option, as can be seen below:

```
function [xmax,postmax,marglike]=LDM(in_model,in_data,in_sims,
in_learning_type,in_ini_beliefs,in_learning_algo,in_timing_PLM)
%% global variables
global datab x0 db_sheet db_cell A0 A1 A2 A3 A4 B0 B1 Hw Hy Hc
omgt
global gain Nobsy Nobsw Nendog Nexog N priors aa2i bb2i cc2i rr2i
global learning_type learning_algo search_method search_maxiter
global T ridgec prior_penalty burn graphs_rf xmax_new
global xmax_new_global C mh_scale mh_output mh_final
global sv_m sv_d sv_s model_no search_robust ini_beliefs search_mcmc
global z0 priors_z Nparam Nparam_z search_display
global mh_draws mh_likelihood mh_prior mh_posterior
global mh_postratio mh_accepted mh_ratio mh_continue results
global step_draw search_RE search_RE_rob llh_RE
global search_displayf numeric_display search_LDM gain_sens
global timing_PLM
global disp_ltype disp_belief disp_lalgo disp_time
%% learning settings
% PLM default options
if nargin<3
```

```
in_data = 1; % 0=RD, 1=RE
in_sims = 0;
learning_type = 3; % 1=VAR, 2=VARc, 3=MSV, 4=MSVc
ini_beliefs = 2; % 1=fix@xmax, 2=RE consis, 3=optimized
learning_algo = 2; % 1=sg, 2=cg
timing_PLM = -1; % PLM time: 0=[t,t-1,t],-1=[t,t-1,t-1],MSV-only
elseif nargin<3
learning_type = 3; % 1=VAR, 2=VARc, 3=MSV, 4=MSVc
ini_beliefs = 2; % 1=fix@xmax, 2=RE consis, 3=optimized
learning_algo = 1; % 1=sg, 2=cg
timing_PLM = -1; % PLM time: 0=[t,t-1,t],-1=[t,t-1,t-1],MSV-only
else
learning_type = in_learning_type;
ini_beliefs = in_ini_beliefs;
learning_algo = in_learning_algo;
timing_PLM = in_timing_PLM;
end
% search options
search_method = 3; % 1=fmin, 2=fminunc, 3=csminwel
search_RE = 1; % uses RE to help search xmax
search_LDM = 1; % uses learning (0~only RE)
search_RE_rob = 1; % # of initial random vecs for RE postsearch
search_robust = 1; % # of initial random vecs for LDM postsearch
search_mcmc = 0; % 0=no post search in mcmc steps, 1=yes
% display options
search_display = 1; % 0=only display xmax, 1=display all steps
search_displayf = 'iter'; % 'off' or 'iter' to display search options
numeric_display = 0; % 0=dont display numerical penalties, 1=display
display_warning = 0; % 0=dont display numerical problems
% path/model settings
[sv_m,sv_d,sv_s] = LDM_aux_filename(in_data,in_model,in_sims);
model_no = ['model' sv_m]; addpath(['./' model_no '/']);
```

% loads initial model information and data

[~,x0]=feval(model_no,zeros(1000,1));

[A0,A1,A2,A3,A4,B0,B1,Hw,Hy,Hc,omgt,gain,Nparam,Nobsy, Nobsw,Nendog,Nexog,N,priors,db_cell,db_sheet,x0]=feval(model_no,x0);

datab = xlsread(num2str(['./' model_no '/' model_no '_' sv_d '.xls']), db_sheet,db_cell);

% other options

search_maxiter = 2000; % max iterations for search method

ridgec = 1e-6; % ridge criteria

prior_penalty = -1e100; % penalty for out-of-bonds prior

burn = 0.3; % burn-in MH

graphs_rf = 0; % reduced form graphs

llh_RE = 0; % calculates loglike using RE (do not change this!)

gain_sens = -0.1; % minimum improvement in posterior that gain dif 0

T = size(datab,1);

if display_warning==0

warning('off', 'MATLAB:nearlySingularMatrix');

warning('off', 'MATLAB:singularMatrix');

warning('off', 'MATLAB:illConditionedMatrix');

end

%% initial beliefs

[z0,priors_z,aa2i,bb2i,cc2i,rr2i]=LDM_2_beliefs(ini_beliefs,learning_type, x0,priors);

Nparam_z=size(z0,1);

%% display model settings

for disppp=1:1

if search_method==1

disp_algo='fminsearch algorithm';

elseif search_method==2

disp_algo='fminunc algorithm';

elseif search_method==3

disp_algo='csminwel algorithm';

```
end
if learning_type==4
disp_ltype='MSVc';
elseif learning_type==3
disp_ltype='MSV';
elseif learning_type==2
disp_ltype='VARc';
elseif learning_type==1
disp_ltype='VAR';
end
if ini_beliefs==3
disp_belief='fixed at RE xmax';
elseif ini_beliefs==2
disp_belief='RE consistent';
elseif ini_beliefs==1
disp_belief='optimized';
end
if learning_algo==4
disp_lalgo='Reverse Eng TVP/S';
elseif learning_algo==3
disp_lalgo='Kalman filter';
elseif learning_algo==2
disp_lalgo='Constant gain';
elseif learning_algo==1
disp_lalgo='Stochastic gradient';
end
if timing_PLM==0
disp_time='t';
elseif timing_PLM==-1
disp_time='t-1';
end
disp(' ')
```

```
disp('————————————————————————')
disp([' Initialializing search : ' disp_algo])
disp([' Learning type : ' disp_ltype])
disp([' Initial beliefs : ' disp_belief])
disp([' Learning algorithm : ' disp_lalgo])
disp([' PLM timing : ' disp_time])
disp(' ')
end
%% searching posterior max
[postmax,zmax,Cz,marglike]=LDM_1_robust(search_robust,priors_z,z0);
xmax=zmax;
C = Cz .* [ ones(Nparam,1);
zeros(Nparam_z-Nparam,1)];
%% performing RW-MH draws
step_draw=1;
if in_sims==0
disp('No draws selected. Exiting')
disp(' ')
return
else
[mh_scale,xmax_new,xmax_new_global,mh_output,mh_final,mh_draws,
mh_likelihood,mh_prior,mh_posterior,mh_postratio,mh_accepted,mh_ratio]=
LDM_4_draws(in_sims);
mh_continue=mh_draws(:,end);
%% results
results=LDM_5_results;
eval(['save ' './model' sv_m '/' 'output_ldm_' sv_d '_' sv_s 'k']);
end
end
```

For instance, if one wants to estimate model 1 (univariate model, $in\_model = 1$) under simulated data ($in\_data = 1$), with no draws from MCMC ($in\_sims = 0$), with MSV learning ($learning\_type = 3$), RE consistent initial beliefs ($ini\_beliefs = 2$), SG learning algorithm ($learning\_algo = 1$) and $t-1$ PLM timing ($timing\_PLM = -1$), the LDM function must be executed as LDM(1,1,0,3,2,1,-1). This sequence is hard to remember (and we did not attempt that) but it is very useful to run loops between different learning specifications.There are three outputs of our m-file: the parameter vector at the posterior maximum (xmax), the value of the posterior at the maximum (postmax) and the marginal likelihood (marglike). A simple loop can be run for simulated data using the following code:

```
in_model=x1;
output=zeros(x2+6,4*2*2)'; % 4 learning type, 2 algo, 2 timing
in_data =1; in_sims =0; m=1; in_ini_beliefs=2;
for j=1:4 % learning type (var,varc,msv,msvc)
for k=[1 2]; % learning algo (sg,cg,tvp)
for l=[0 -1]; % timing (t, t-1)
[xmax,postmax,marglike]=LDM(in_model,in_data,0,j,2,k,l);
output(m,:)=[j,in_ini_beliefs,k,l,postmax,marglike,xmax'];
m=m+1;
end;end;end
disp(output)
```

Where **x1** stands for the number of the model and **x2** stands for the number of parameters that are estimated (in order to create the output row for each specification, that will contain (i) the specification description (learning type, initial beliefs, learning algorithm, PLM timing), a 1×4 vector, the (scalar) value of the posterior maximum, the (scalar) value of the marginal likelihood, and the $1 \times \mathbf{x2}$ vector of parameters. This simple loop runs 16 learning specifications, and create a matrix $16 \times (6 + \mathbf{x2})$ with the results. This matrix is then transposed and (after translating the inputs back to the learning specification) gives us the detailed tables of Appendix A.

If one wants to run a particular specification, however, it is possible to do so by simply running:

```
[xmax,postmax,marglike]=LDM(1,1,0,3,2,1,-1)
```

2) **modelx.m**, which is the model m-file that describes all the information about the model: the number of parameters, endogenous and exogenous variables, the database excel sheet and cells in which the data is located (each model must have its files located at a subdirectory[76] named modelx), the priors and the equations of the model (in matrix form). As an example, we show our simple 3 equation new Keynesian model m-file (which is named model3 in our computer, but it is model 2 in our thesis):

```
function [A0,A1,A2,A3,A4,B0,B1,Hw,Hy,Hc,omgt,gain,Nparam,Nobsy,
Nobsw,Nendog,Nexog,N,priors,db_cell,db_sheet,x0]=model3(param)
% MODEL INSTRUCTIONS:
% 1) write model settings (number of param, etc)
% 2) write parameter list in order: ordinary, gain, variance matrix
% 3) write matrices A0-A4,B0-B1, with observables in top
% A0.y(t-1) + A1.Y(t) + A2.E_t[y(t+1)] + A3.w(t) + A4 = 0
% w(t) + B0.w(t-1) + B1.eps2(t) = 0
% settings
Nparam = 13; % number of parameters
Nendog = 3; % number of endogenous variables
Nexog = 3; % number of exogenous variables
N = Nendog+Nexog; % number of states
Nobsy = 03; % observable endogenous variables
Nobsw = 00; % observable exogenous variables
priors=zeros(Nparam,4); % vector

db_cell='A2:C1000'; % database cells
db_sheet='Sheet1'; % database sheet
% parameters' names % priors: 1=N, 2=beta, 3=unif, 4=invG
beta = param(1,1); priors(1,1:3) =[1 0.99 .001];
kappa = param(2,1); priors(2,1:3) =[1 0.5 0.20];
sig = param(3,1); priors(3,1:3) =[1 0.3 0.20];
```

[76] All LDM functions must be located in the same folder. Files for each model (which include m-file with model description and the Excel spreadsheet containing the data) must be put in a subdirectory under the LDM files named "modelx" where x is the number of the model. The m-file name must be "modelx.m", while the xls files must be "modelx_RD.xls" for real data and "modelx_RE.xls" for simulated data.

```
phi1 = param(4,1); priors(4,1:3) =[1 0.6 0.20];
phi2 = param(5,1); priors(5,1:3) =[1 0.6 0.20];
rhoi = param(6,1); priors(6,1:3) =[1 0.5 0.20];
rho1 = param(7,1); priors(7,1:3) =[1 0.7 0.40];
rho2 = param(8,1); priors(8,1:3) =[1 0.5 0.40];
rho3 = param(9,1); priors(9,1:3) =[1 0.3 0.40];
gain = param(10,1); priors(10,1:3)=[3 0.0 0.10];
omgt(1,1) = param(11,1); priors(11,1:3)=[4 3.0 1.00];
omgt(2,2) = param(12,1); priors(12,1:3)=[4 3.0 1.00];
omgt(3,3) = param(13,1); priors(13,1:3)=[4 3.0 1.00];

% true/initial vector
x0 = [0.99 0.5 0.5 0.7 0.7 0.5 0.5 0.5 0.3 0.01 0.5 0.5 0.5]';

% matrices:

A0= [0 0 0;
0 0 0;
0 0 -rhoi];

A1=[1 -kappa 0;
0 1 sig;
0 0 1];

A2=[-beta 0 0;
-sig -1 0;
-phi1 -phi2 0];

A3=[-1 0 0;
0 -1 0;
0 0 -1];

A4=[0;
0;
0];
B0=[-rho1 0 0;
```

```
0 -rho2 0;
0 0 -rho3];
B1=[-1 0 0;
0 -1 0;
0 0 -1];
Hc=[0;0;0];
Hy=[ 1 0 0;
0 1 0;
0 0 1];
Hw=zeros(3,3);
end
```

Aside from these files, which are the backbone of estimating the learning specifications of this thesis, there are several auxiliary m-files that perform different tasks, such as evaluating the likelihood (*LDM_aux_loglike*) and prior (*LDM_aux_prior*) functions, search for the posterior maximum in a standard (*LDM_3_postsearch*) or robust (*LDM_1_robust*) way, generating initial beliefs (*LDM_2_beliefs*), perform more draws from the MCMC steps (*LDM_continue*) and so on. More details are available at igorvele at gmail dot com.

6 References

AMISANO, G.; SERATI, M. Unemployment persistence in Italy. An econometric analysis with multivariate time varying parameter models. Cattaneo University, Working paper, n.21, apr/2003.

ANDERSON, G.; MOORE, G. A linear algebraic procedure for solving linear perfect foresight models. Economics Letters, v.17, n.3, p.247-252, 1985.

ANDERSON, G. Solving Linear Rational Expectations Models: A Horse Race. Computational Economics, Society, v.31, n.2, p.95-113, mar/2008.

BIANCHI, F. Regime Switches, Agents' Beliefs, and Post-World War II U.S. Society for Economic Dynamics, Working paper, n.198, 2009.

BLANCHARD, O.J.; KAHN, C. M. The Solution of Linear Difference Models under Rational Expectations. Econometrica, v.48, n.5, p.1305-11, jul/1980.

BRANCH, W.; EVANS, G. W.; MCGOUGH, B. Finite Horizon Learning. Centre for Dynamic Macroeconomic Analysis, Working paper n.201204, 2012.

BRANCH, W. A.; EVANS, G. W. A simple recursive forecasting model. Economics Letters, v. 91, n.2, p.158-166, may/2006.

BRUNNERMEIER, M.K.; PARKER, J.A. Optimal Expectations, American Economic Review, American Economic Association, vol. 95(4), pages 1092-1118, sep/2005.

CARCELES-POVEDA, E.; GIANNITSAROU, C. Adaptive learning in practice. Journal of Economic Dynamics and Control, v.31, n.8, p.2659-2697, aug/2007.

CASTRO, M. R.; GOUVEA, S. N.; MINELLA, A.; SANTOS, R. C.; SOUZA-SOBRINHO, N. F. SAMBA: Stochastic Analytical Model with a Bayesian Approach. Central Bank of Brazil, Working Papers, n.239, 2011

CHARI, V.V.; KEHOE, P.J.; MCGRATTAN, E.R. New Keynesian Models: Not Yet Useful for Policy Analysis. American Economic Journal: Macroeconomics, American Economic Association, v.1, n.1, p.242-66, jan/2009.

CHRISTIANO, L. J.; EICHENBAUM, M.; EVANS, C. L. Nominal Rigidities and the Dynamic Effects of a Shock to Monetary Policy. Journal of Political Economy: University of Chicago Press, v. 113, n.1, p.1-45, feb/2005.

CONGDON, P. Bayesian Statistical Modelling, Wiley, 2006, 2nd edition.

FERNANDEZ-VILLAVERDE, J.; RUBIO-RAMIREZ, J.F. How Structural Are Structural Parameters?, National Bureau of Economic Research, Working paper, v.22, p.83-137, 2008.

EUSEPI, S.; PRESTON, B. Expectations, Learning, and Business Cycle Fluctuations. American Economic Review, v.101, n.6, p. 2844-72, oct/2011.

EVANS, G.W.; HONKAPOHJA, S. Learning and Expectations In: Microeconomics.Princeton University, 2001.

EVANS, G. W.; HONKAPOHJA, S. Adaptive learning and monetary policy design. Proceedings, Federal Reserve Bank of Cleveland, Working paper, p.1045-1084, 2003.

EVANS, G. W.; MITRA, K..E-stability in the stochastic Ramsey model. Economics Letters, v.118, n.2, p. 407- 410, 2013.

GALI, J. Introduction to Monetary Policy, Inflation, and the Business Cycle: An Introduction to the New Keynesian Framework. Princeton University Press, 2008.

GAUS, E. Time-Varying Parameters and Endogenous Learning Algorithms. Ursinus College, Working Papers, n.13-02, mar/2013.

JAKAB, Z. M., VILAGI, B. An estimated DSGE model of the Hungarian economy. Magyar Nemzeti Bank, Working paper, n.2008/9, 2008.

KOOP, G.; POIRIER, D. J.; TOBIAS, J. L. Bayesian Econometric Methods. Cambridge University Press, 2007.

HONKAPOHJA, S.; MITRA, K.; EVANS, G. W. Notes on Agents' Behavioral Rules Under Adaptive Learning and Studies of Monetary Policy. Centre for Dynamic Macroeconomic Analysis, Working Paper, n.201102, 2011.

HANSEN, L.P.; SARGENT, T.J. Robustness, Princeton: Princeton University Press, 2008.

INGRAM, B. F.; WHITEMAN, C. H. Supplanting the "Minnesota" prior: Forecasting macroeconomic time series using real business cycle model priors. Journal of Monetary Economics, v.34, n.3, p.497-510, dec/1994.

LIU, P.; MUMTAZ, H. Evolving macroeconomic dynamics in a small open economy: an estimated Markov-switching DSGE model for the United Kingdom. Journal of Money, Credit and Banking, v.43, n.7, p.1443-1474, oct/2011.

LUCAS, R. E. Jr. Methods and Problems in Busines Cycle Theory. Journal of Money, Credit and Banking, v. 12, n. 4, parte 2, p.696-715, nov/1980.

LUCAS, R. E. Jr. Econometric Policy Evaluation: A Critique. in: K. Brunner and

A. Meltzer (eds.), The Phillips Curve and Labor Markets, Carnegie-Rochester Conference Series on Public Policy, v. 1, p. 19-46, 1976

KING, R. G.; WATSON, M. W. System Reduction and Solution Algorithms for Singular Linear Difference Systems under Rational Expectations. Computational Economics, Society, v.20, n.1-2, p.57-86, oct/2002.

MILANI, F. A Bayesian DSGE Model with Infinite-Horizon Learning: Do "Mechanical" Sources of Persistence Become Superfluous?. International Journal of Central Banking, v.2, n.3, p.87-106, sep/2006.

MILANI, F. Expectations, learning and macroeconomic persistence. Journal of Monetary Economics, v.54, n.7, p.2065-2082, oct/2007.

MINELLA, A.; SOUZA-SOBRINHO, N. F. Monetary policy channels in Brazil through the lens of a semi-structural model. Economic Modelling, v.30, p.405-419, 2013.

NEGRO, M. D.; SCHORFHEIDE F. Priors from General Equilibrium Models for VARS. International Economic Review, v.45, n.2, p.643-673, aug/2004.

NEGRO, M. D.; SCHORFHEIDE F. A DSGE-VAR for the Euro Area. Society for Economic Dynamics, Working paper, n.43, 2004.

NEGRO, M. D.; SCHORFHEIDE F.; SMETS, F.; WOUTERS, R. On the fit and forecasting performance of New-Keynesian models. European Central Bank, Working paper, n.491, jun/2005.

NEGRO, M. D.; SCHORFHEIDE F. How good is what you've got? DSGE-VAR as a toolkit for evaluating DSGE models. Economic Review, n. Q2, p. 21-37, 2006.

ORPHANIDES, A.; WILLIAMS, J.C. Monetary Policy with Imperfect Knowledge, unpublished, Federal Reserve Board, aug/2005.

PRIMICERI, G. E. Time Varying Structural Vector Autoregressions and Monetary Policy. Review of Economic Studies, v.72, n.3, p. 821-852, 2005.

PRESTON, B. Learning about Monetary Policy Rules when Long-Horizon Expectations Matter. International Journal of Central Banking, v.1, n.2, p.81-126, sep/2005.

RANTALA, A. Adaptive Learning and Multiple Equilibria in a Natural Rate Monetary Model with Unemployment Persistence. Bank of Finland, Working Paper n. 30/2003, nov/2003.

SARGENT, T.J. Evolution and Intelligent Design, American Economic Review, American Economic Association, vol. 98(1), pages 5-37, mar/2008.

SIMS, C. A. Stickiness. Carnegie-Rochester Conference Series on Public Policy, vol. 49, pages 317-356, 1998.

SIMS, C. A.; ZHA, T. Bayesian Methods for Dynamic Multivariate Models. International Economic Review, v.39, n.4, p. 949-68, nov/1998.

SIMS, C. A. Implications of Rational Inattention. Journal of Monetary Economics, vol. 50(3), pages 665-690, 2003.

SIMS, C. A.; ZHA, T. Were There Regime Switches in U.S. Monetary Policy?. American Economic Review, v. 96, n.1, p.54-81, mar/2006.

SIMS, C. A. Improving monetary policy models. Journal of Economic Dynamics and Control, v.32, n.8, p.2460-2475, aug/2008.

SIMS, C. A. Rational Inattention and Monetary Economics, Handbook of Monetary Economics, edition 1, volume 3, chapter 4, pages 155-181, 2010.

SLOBODYAN, S.;WOUTERS, R. Estimating a medium–scale DSGE model with expectations based on small forecasting models. Society for Economic Dynamics, Working paper, n.654, 2009.

SLOBODYAN, S.; WOUTERS, R. Learning in an estimated medium-scale DSGE model. Journal of Economic Dynamics and Control, v.36, n.1, p.26-46, 2012.

SMETS, F.; WOUTERS, R. An Estimated Dynamic Stochastic General Equilibrium Model of the Euro Area. Journal of the European Economic Association, v.1, n.5, p. 1123-1175, sep/2003.

SMETS, F.; WOUTERS, R. Shocks and Frictions in US Business Cycles: A Bayesian DSGE Approach. American Economic Review, v. 97, n.3, p. 586-606, jun/2007.

TOVAR, C. E. DSGE Models and Central Banks. Economics: The Open-Access, Open-Assessment E-Journal, v.3, n.16, p.1-31, 2009.

WARNE, A.; COENEN, G.; CHRISTOFFEL, K. Predictive likelihood comparisons with DSGE and DSGE-VAR models. European Central Bank, Working paper, n.1536, 2013.

WOODFORD, M. Interest and Prices: Foundations of a Theory of Monetary Policy, Princeton University Press, 2003.

WOODFORD, M. Robustly Optimal Monetary Policy with Near-Rational Expectations, American Economic Review, American Economic Association, vol. 100(1), pages 274-303, mar/2010.

WOODFORD, M. Macroeconomic Analysis without the Rational Expectations Hypothesis. NBER Working Papers 19368, National Bureau of Economic Research, Inc., 2013.