A Matlab Toolbox to Solve Dynamic Multivariate Rational Inattention Problems in the LQG Framework

Jianjun Miao* Jieran Wu†

First version: October 2020
This version: July 2021

Abstract

We develop a Matlab toolbox to solve dynamic multivariate rational inattention problems in the LQG framework. The toolbox works for general LQG control and tracking problems. We describe Matlab codes to solve for the golden rule, the steady state, and the transition dynamics of the optimal information structure. We also describe Matlab codes to replicate numerical results in Miao, Wu, and Young (2021).

*Department of Economics, Boston University, 270 Bay State Road, Boston, MA 02215. Email: miaoj@bu.edu
†Academy of Financial Research and College of Economics, Zhejiang University, jw5ya@zju.edu.cn

This paper describes a Matlab toolbox to solve dynamic RI problems in the LQG framework studied by Miao, Wu, and Young (MWY henceforth) (2021). This toolbox works for general LQG control and tracking problems. We introduce Matlab codes to solve for the golden rule, the steady state, and the transition dynamics of the optimal information structure. Section 1 presents the RI problems. Section 2 describes our RI solvers. Section 3 describes Matlab codes to replicate results in MWY (2021) and also to solve some additional examples.

1 Multivariate RI Problems

Our toolbox solves two types of RI problems in the LQG framework. We consider the infinite-horizon setup. Our toolbox also works for finite-horizon problems with suitable modification. The first type is control problems.

Problem 1 *(LQG control problem under RI)*

$$\max_{\{u_t\},\{C_t\},\{V_t\}} \quad -\mathbb{E}\left[\sum_{t=0}^{\infty}\beta^t\left(x_t'Qx_t+u_t'Ru_t+2x_t'Su_t\right)\right] \; -\lambda \;\sum_{t=0}^{\infty}\beta^t I\left(x_t;s_t|s^{t-1}\right)$$

subject to

$$x_{t+1}=Ax_t+Bu_t+\epsilon_{t+1}, \tag{1}$$

$$s_t=C_tx_t+v_t,\;\; t\geq 0, \tag{2}$$

where $\beta\in(0,1)$, $s^t=\{s_0,...,s_t\}$, $s^{-1}=\varnothing$, x_0 *is Gaussian* $\mathcal{N}\left(\overline{x}_0,\Sigma_{-1}\right)$, $\Sigma_{-1}\succ 0$, ϵ_t *is a Gaussian white noise* $\mathcal{N}\left(0,W\right)$, *and* v_t *is a Gaussian white noise* $\mathcal{N}\left(0,V_t\right)$, *and* $I\left(x_t;s_t|s^{t-1}\right)$ *denotes the mutual information. Assume that* x_0, v_t, *and* ϵ_t *are independent. The control* u_t *is adapted to* s^t.

We apply the following three-step procedure.

Step 1. Compute the full information solution.

The value function under full information is given by

$$v^{FI}\left(x_t\right)=-x_t'Px_t-\frac{\beta}{1-\beta}\mathrm{tr}\left(WP\right),$$

where P satisfies the Riccati equation

$$P=Q+\beta A'PA-\left(\beta A'PB+S\right)\left(R+\beta B'PB\right)^{-1}\left(\beta B'PA+S'\right). \tag{3}$$

The optimal control is given by

$$u_t=-Fx_t, \tag{4}$$

where

$$F=\left(R+\beta B'PB\right)^{-1}\left(S'+\beta B'PA\right).$$

Step 2. Compute the solution under limited information with fixed information structure.

The optimal control is given by

$$u_t = -F\widehat{x}_t,$$

where $\widehat{x}_t \equiv \mathbb{E}\left[x_t|s^t\right]$. The negative of maximized utility is given by

$$\begin{aligned} & \mathbb{E}\left[\sum_{t=0}^{\infty}\beta^t\left(x_t'Qx_t + u_t'Ru_t + 2x_t'Su_t\right)\right] \\ = \; & \mathbb{E}\left[x_0'P_0x_0\right] + \sum_{t=0}^{\infty}\beta^{t+1}\text{tr}\left(WP\right) + \sum_{t=0}^{\infty}\beta^t\text{tr}\left(\Omega\Sigma\right), \end{aligned}$$

where

$$\Omega = F'(R + \beta B'PB)F \succeq 0. \tag{5}$$

The state under the optimal control satisfies the dynamics

$$x_{t+1} = Ax_t - BF\widehat{x}_t + \epsilon_{t+1}. \tag{6}$$

By the Kalman filter formula, $\widehat{x}_t$ follows the dynamics

$$\widehat{x}_t = \widehat{x}_{t|t-1} + \Sigma_{t|t-1}C_t'\left(C_t\Sigma_{t|t-1}C_t' + V_t\right)^{-1}\left(s_t - C_t\widehat{x}_{t|t-1}\right), \tag{7}$$

$$\widehat{x}_{t|t-1} = \left(A - BF\right)\widehat{x}_{t-1}, \tag{8}$$

where $\widehat{x}_{t|t-1} \equiv \mathbb{E}\left[x_t|s^{t-1}\right]$ with $\widehat{x}_{0|-1} = \bar{x}_0$ and $\Sigma_{t|t-1} \equiv \mathbb{E}\left[\left(x_t - \widehat{x}_{t|t-1}\right)\left(x_t - \widehat{x}_{t|t-1}\right)'|s^{t-1}\right]$ with $\Sigma_{0|-1} = \Sigma_{-1}$ exogenously given. Moreover,

$$\Sigma_{t+1|t} = A\Sigma_tA' + W, \tag{9}$$

$$\Sigma_t = \left(\Sigma_{t|t-1}^{-1} + \Phi_t\right)^{-1}, \tag{10}$$

for $t \geq 0$, where $\Sigma_t \equiv \mathbb{E}\left[\left(x_t - \widehat{x}_t\right)\left(x_t - \widehat{x}_t\right)'|s^t\right]$ denotes the posterior covariance matrix given s^t and Φ_t denotes the signal-to-noise ratio (SNR) defined by $\Phi_t = C_t'V_t^{-1}C_t \succeq 0$.

Step 3. Compute optimal information structure $\{C_t, V_t, \Sigma_t\}$.

First solve the following RI problem for $\{\Sigma_t\}_{t=0}^{\infty}$. Then recover (C_t, V_t) using

$$C_tV_t^{-1}C_t' = \Sigma_t^{-1} - \left(A\Sigma_tA + W\right)^{-1}.$$

Our toolbox focuses mainly on this step.

Problem 2 *(Optimal information structure for Problem 1)*

$$\min_{\{\Sigma_t\}_{t=0}^{\infty}} \sum_{t=0}^{\infty}\beta^t\left[\text{tr}\left(\Omega\Sigma_t\right) + \lambda I\left(x_t; s_t|s^{t-1}\right)\right]$$

subject to

$$I\left(x_t; s_t|s^{t-1}\right) = \frac{1}{2}\log\det\left(A\Sigma_{t-1}A' + W\right) - \frac{1}{2}\log\det\left(\Sigma_t\right),$$

$$I\left(x_0; s_0|s^{-1}\right) = \frac{1}{2}\log\det\left(\Sigma_{-1}\right) - \frac{1}{2}\log\det\left(\Sigma_0\right),$$

$$\Sigma_t \preceq A\Sigma_{t-1}A' + W, \tag{11}$$

$$\Sigma_0 \preceq \Sigma_{-1}, \tag{12}$$

for $t \geq 0$, *where* Ω *is given by (5).*

Our toolbox can also solve pure tracking problems. In independent work, Afrouzi and Yang (2019) develop a Julia toolbox to solve this type of problems.

Problem 3 *(Pure tracking problem under RI)*

$$\min_{\{z_t, C_t, V_t\}} \mathbb{E}\sum_{t=0}^{\infty}\beta^t\left[(y_t - z_t)'(y_t - z_t) + \lambda I\left(x_t; s_t|s^{t-1}\right)\right]$$

subject to (2) and

$$x_{t+1} = Ax_t + \eta_{t+1}, \quad y_t = Gx_t.$$

The optimal action is $z_t = \mathbb{E}\left[y_t|s^t\right] = G\mathbb{E}\left[x_t|s^t\right]$. Then the optimal information structure in the above tracking problem is the solution to Problem 2, where we set $\Omega = G'G$.

We impose the following assumption taken from Afrouzi and Yang (2019) to ensure that $(A\Sigma_t A + W)$ is invertible for $\Sigma_t \succ 0$ and that the dynamic RI problem is convex, so that the first-order conditions are necessary and sufficient for optimality.

Assumption 1 $W \succeq 0$ *and* $AA' + W \succ 0$.

Our toolbox works under this assumption.

2 RI Solvers

This section presents Matlab codes to solve Problems 1, 2, and 3 under Assumption 1. We also describe Matlab codes to characterize the signal structure and plot impulse response functions. The description of algorithms and theoretical foundation are presented in MWY (2021). Each algorithm has its own pros and cons in terms of computation speed, stability, convergence, and robustness. When using value function based methods, one needs to download CVX from the internet (http://cvxr.com/cvx/). CVX supports two free semidefinite programming (SDP) solvers, SeDuMi (Sturm (1999)) and SDPT3 (Toh, Todd, and Tutuncu (1999)), and a commercial SDP solver, Mosek, which is also free for academic users. These solvers use the interior point method. None of the solvers is perfect. We find that Mosek is the fastest and SDPT3 is the most reliable for our examples.

2.1 Golden Rule Solution

- RI_GR.m: Compute the golden rule solution for dynamic RI problems.

Problem 4 *(Golden-rule information structure)*

$$\min_{\Sigma \succ 0} \; (1-\beta)\,\mathrm{tr}\left(A'PA\Sigma\right) + \mathrm{tr}\left(\Omega\Sigma\right) + \frac{\lambda}{2}\left[\log\det\left(A\Sigma A' + W\right) - \log\det\left(\Sigma\right)\right]. \tag{13}$$

subject to $\Sigma \preceq A\Sigma A' + W$.

When $\beta = 1$, the golden-rule solution is the same as the steady state solution.

2.2 Value Function Based Methods

- RI_VFI.m: This code computes the steady state and transition dynamics of the optimal information structure for the pricing example in Sims (2011) using the basic VFI method.

- RI_SS_EC.m: Compute the steady state of dynamic RI problems using the modified VFI method based on the envelope condition.

- RI_TD_EC.m: Compute the transition dynamics for dynamic RI problems using the modified VFI method based on the envelope condition.

- RI_SS_SQ: Compute the steady state of dynamic RI problems using the modified VFI method based on a sequence of static RI models.

- RI_TD_SQ: Compute the transition dynamics for dynamic RI problems using the modified VFI method based on a sequence of static RI models.

- RI_Static.m: Solve static RI problems.

2.3 First-order Conditions Based Methods

- RI_SS_FOC.m: Compute the steady state of dynamic RI problems using the first-order conditions.

- RI_TD_FOC.m: Compute the transition dynamics for dynamic RI problems using first-order conditions.

2.4 Signal Structure and Impulse Response Function

- RI_IRF.m: Compute the impulse response function for dynamic RI problems where the innovation covariance matrix for the state dynamics is diagonal.

- RI_IRF2.m: Compute the impulse response function for a general innovation covariance matrix in the state dynamics.

- RI_SIG.m: Compute the optimal signal structure for dynamic RI problems using Corollary 1 of MWY (2021).

- RI_SIG1.m: Compute the signal structure for given prior and posterior covariance matrices using Proposition 1 of MWY (2021).

2.5 Utilities

- LQG.m: Solve the standard linear-quadratic-Gaussian control model under full information.

- svdc.m: Singular value decomposition in a compact form.

- armafit.m: Fit VARMA(p, q) representation to a set of points $\{(z, f(z))\}$.

- Varma2ma1.m: Approximate VARMA(p, q) by VMA(N) with large N.

- armaeval.m: Evaluate VARMA(p, q) representation.

3 Examples

This section lists codes to replicate all numerical examples in MWY (2021) and verify propositions in MWY (2021).

3.1 Sims (2011) Example

- Nonconverge.m: This code solves for the steady state information structure for the pricing example in Sims (2011) using (i) the modified VFI method based on a sequence of static RI problems, (ii) the modified VFI method using envelope condition, (iii) our FOC based method, and (iv) Afrouzi and Yang (2019) method using their code Solve_RI_Dynamics.m.

- SimsEx_rho.m: This code replicates numerical examples in Section 4 of MWY (2021) and replicates Figure 1.

- SimsEx_betav.m: This code replicates numerical examples in Section 4 of MWY (2021) and replicates Figure 2.

- SimsEx_beta.m: This code is called by SimsEx_betav.m.

- SimsEx_NewConstraint.m: This code computes the steady state information structure for the pricing example in Sims (2011) with an additional constraint on the entropy of the posterior covariance matrix, i.e.,

$$\log\det(\Sigma_t) \geq l \text{ for some } l. \tag{14}$$

The code uses the modified VFI method using the envelope condition and the modified VFI method based on a sequence of static RI problems. The two methods replicate the same solution in Online Appendix G.4 of MWY (2021). The method of Afrouzi and Yang (2019) does not apply to this problem.

- SimsEx_Plot_rho.m: This code calls SimsEx_IRF.m to plot impulse response functions for different values of ρ.
- SimsEx_IRF.m: This code computes the impulse response function and signal structure in the steady state for the Sims (2011) pricing example.
- SimsEx_TD.m: This code computes the transition dynamics of the pricing example in Sims (2011) starting from any initial prior covariance matrix. The code compares the first-order conditions based method with the modified VFI method based on a sequence of static RI problems.
- PriceCapacity.m: This code solves the golden-rule information structure (or the steady state with $\beta = 1$) for the pricing example in Sims (2011) with period-by-period capacity constraints.

3.2 Confirming Propositions in MWY (2021)

- VerifyLemma3.m: This code verifies the static generalized reverse water-filling solution in Lemma 3 of MWY (2021).
- VerifyProp.m: This code uses examples to numerically verify Propositions 4-6 in MWY (2021).
- UnivariateEx.m: This code verifies Proposition 7 in MWY (2021).

3.3 Equilibrium Pricing

No Strategic Complementarity

- PriceNSC1.m: Solve equilibrium based on the steady state information structure for the baseline parameter values.
- PriceSigmaNSC1.m: Derive comparative statics of volatilities, which calls PriceNSC1.m.
- PriceRhoNSC1.m: Derive comparative statics with respect to persistence, which calls PriceNSC1.m.

Strategic Complementarity

- PricingCostIRF1.m: Compute comparative statics with respect to the information cost parameter, which calls PricingGE1.m.

  This codes replicates Figure 3 of MWY (2021).

- PricingSCIRF1.m: Compute comparative statics with the degree of strategic complementarity, which calls PricingGE1.m.

  This codes replicates Figure 3 of MWY (2021).

- PricingSigmaIRF1.m: Compute comparative statics of volatilities, which calls PricingGE1.m.

  This codes replicates Figure 4 of MWY (2021).

- PricingRhoIRF1.m: Compute comparative statics with respect to persistence, which calls PricingGE1.m.

  This codes replicates Figure 5 of MWY (2021).

- PricingGE1.m: Compute the general equilibrium model of Maćkowiak and Wiederholt (2009) without the signal independence assumption for one set of parameter values.

3.4 Consumption

- ConsRI.m: This code replicates Figure 6 of MWY (2021). Call LQG.m to solve for the value function and the policy function under full information.

3.5 Investment

- FirmRI.m: This code computes the firm investment example and replicates Figure 7 of MWY (2021). Call the code LQG.m to solve for the value function and the policy function under full information.

3.6 Tracking Problem with General ARMA Processes

- ExampleMA.m: This code computes the steady state and transition dynamics information structure for a tracking problem with discounted information costs and with MA(2) process and compares with different solution methods. This code replicates the example in Online Appendix D of MWY (2021).

- ExampleARMA.m: This code computes the steady state and transition dynamics information structure for a tracking problem with discounted information costs and with ARMA(p, q) process ($p \leq q + 1$) and compares with different solution methods.

References

Afrouzi, Hassan, and Choongryul Yang, 2019, Dynamic Rational Inattention and the Phillips Curve, working paper, Columbia University.

CVX Research, Inc., 2011, CVX: Matlab Software for Disciplined Convex Programming, version 2.0. http://cvxr.com/cvx.

Miao, Jianjun, Jieran Wu, and Eric Young, 2021, Multivariate Rational Inattention, working paper, Boston University.

Sims, Christopher A., 2011, Rational Inattention and Monetary Economics, in *Handbook of Monetary Economics*, V. 3A, Benjamin M. Friedman and Michael Woodford (ed.), North-Holland.

Sturm, Jos F., 1999, Using SeDuMi 1.02, A MATLAB Toolbox for Optimization over Symmetric Cones. In *Optimization Methods and Software*, 11-12: 625-633. Special issue on Interior Point Methods (CD supplement with software).

Toh, Kim-Chuan, Michael J. Todd, and R.H. Tutuncu, 1999, SDPT3 — A Matlab Software Package for Semidefinite Programming, *Optimization Methods and Software* 11, 545–581.