Materials 14 - Maybe a last attempt to get rid of the overshooting

Laura Gáti

January 24, 2020

Overview

1 **Model summary** 1

2 **Regime-switching** 2

3 **Projection facility: checking `eig(phi)` when ϕ isn't square?** 4

4 **Endogenous states don't evolve as they should** 4

5 **Reference plots** 6

1 Model summary

$$x_t = -\sigma i_t + \hat{\mathbb{E}}_t \sum_{T=t}^{\infty} \beta^{T-t}\big((1-\beta)x_{T+1} - \sigma(\beta i_{T+1} - \pi_{T+1}) + \sigma r_T^n\big) \tag{1}$$

$$\pi_t = \kappa x_t + \hat{\mathbb{E}}_t \sum_{T=t}^{\infty} (\alpha\beta)^{T-t}\big(\kappa\alpha\beta x_{T+1} + (1-\alpha)\beta\pi_{T+1} + u_T\big) \tag{2}$$

$$i_t = \psi_\pi \pi_t + \psi_x x_t + \bar{i}_t \tag{3}$$

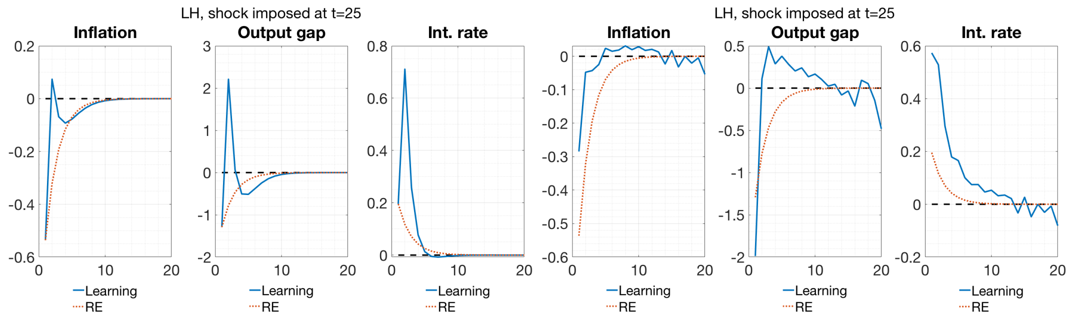
$$\hat{\mathbb{E}}_t z_{t+h} = \bar{z}_{t-1} + b h_x^{h-1} s_t \quad \forall h \geq 1 \qquad b = g_x\ h_x \qquad \text{PLM} \tag{4}$$

$$\bar{z}_t = \bar{z}_{t-1} + k_t^{-1} \underbrace{\big(z_t - (\bar{z}_{t-1} + b s_{t-1})\big)}_{\text{fcst error using (4)}} \tag{5}$$

(Vector learning. For scalar learning, $\bar{z} = \begin{pmatrix} \bar{\pi} & 0 & 0 \end{pmatrix}'$. I'm also not writing the case where the slope b is also learned.)

$$k_t = \begin{cases} k_{t-1} + 1 & \text{for decreasing gain learning} \\ \bar{g}^{-1} & \text{for constant gain learning.} \end{cases} \tag{6}$$

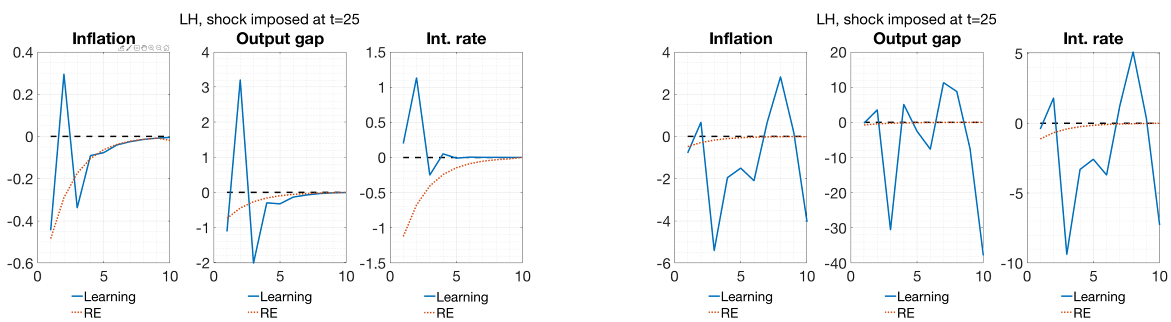
Figure 1: Reference: baseline model



(a) Learning constant only (b) Learning slope and constant

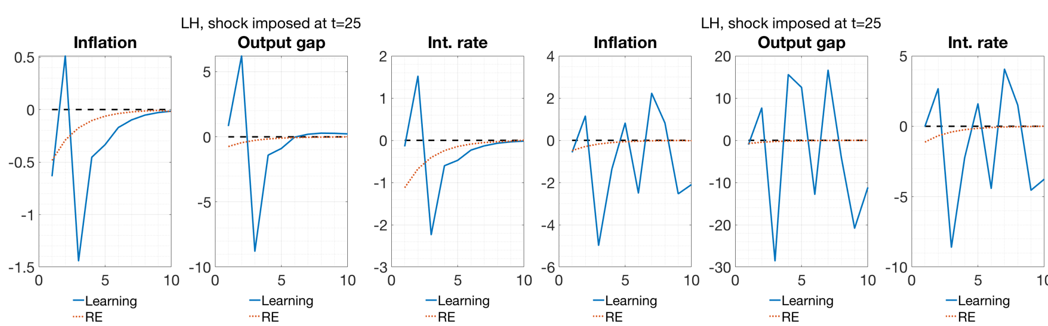
2 Regime-switching

Figure 2: Markov-switching Taylor rule, baseline, learning initialized at active state



(a) Learning constant only (b) Learning slope and constant

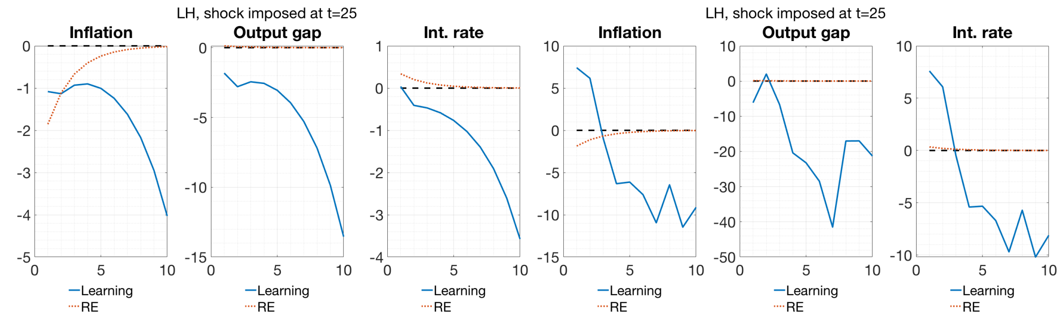
Figure 3: Markov-switching Taylor rule, baseline, learning initialized at passive state



(a) Learning constant only (b) Learning slope and constant

- Different initialization of learning doesn't make a whole lot of difference.
- It just changes where you start, but doesn't fundamentally affect dynamics.

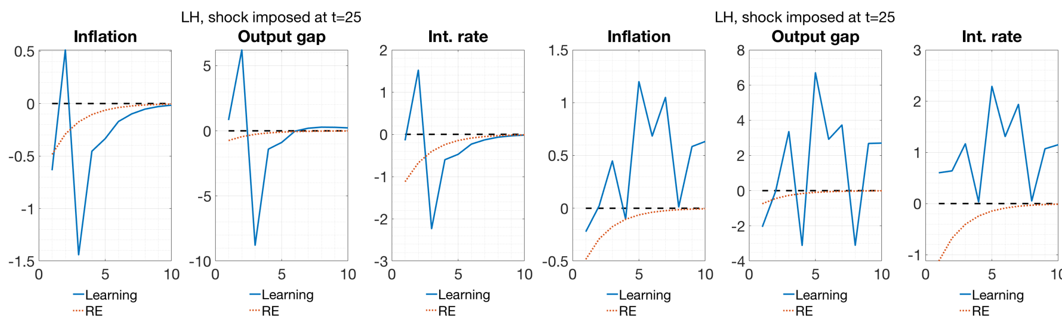
Figure 4: Markov-switching Taylor rule, baseline, learning initialized at passive state, conditional on passive regime only



(a) Learning constant only

(b) Learning slope and constant

Figure 5: Markov-switching Taylor rule, baseline, learning initialized at passive state, conditional on active regime only



(a) Learning constant only

(b) Learning slope and constant

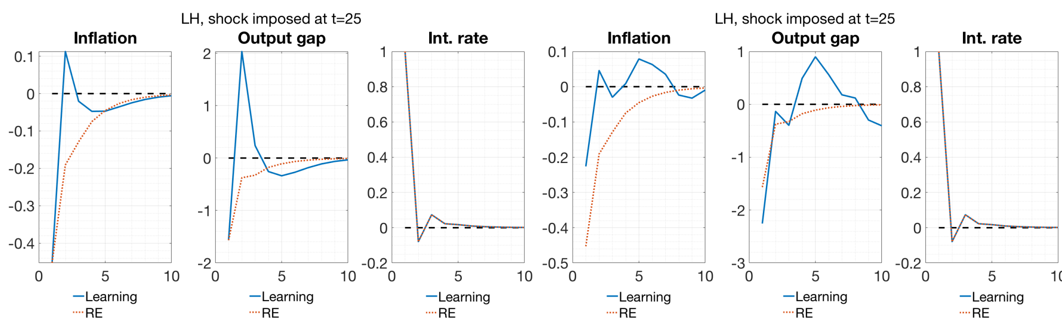
- I'm surprised that the all-passive state is unstable. I've checked and it's not E-stable: the difference in the learning matrix ϕ grows over time, even with decreasing gain learning.
- The all-active is very volatile.

3 Projection facility: checking `eig(phi)` when ϕ isn't square?

What I do now is I check `eig(R)` because that is always square, and when ϕ explodes, usually R does too. Of course I can't do this for learning the constant only, but according to my experience, that's where the projection facility is least likely to ever be needed. Of course, this doesn't always work - for interest rate smoothing, it doesn't.

4 Endogenous states don't evolve as they should

Figure 6: Lagged inflation in TR, "suboptimal forecasters" info assumption



(a) Learning constant only (b) Learning slope and constant

Let an e superscript denote the position of the endogenous state in the jump and state transition matrices. (So it will be a different number for the two.) What I want to do is to replace h_x^e with $\hat{g}_x^e$. So let me call the thus created state transition matrix H:

$$H = \begin{bmatrix} h_x & \mathbf{0} \\ \hat{g}_x^e & \end{bmatrix} \tag{7}$$

There are several issues with this. Ideally I want to move the states forward as:

$$s_{t+1} = Hs_t \tag{8}$$

But that doesn't take into account that:

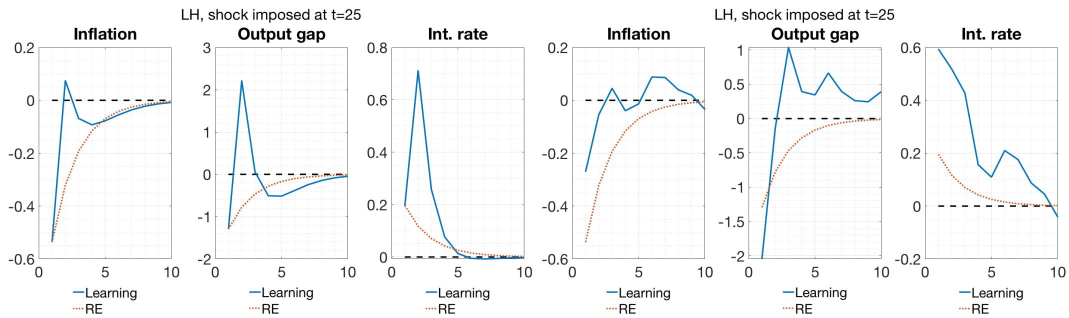
- Agents' estimate of g_x is misspecified: ϕ has one higher row dimension than g_x. For this reason, what I'm trying is to replace the last element in s_{t+1} with $a^e + b^e s_t$.

- But b doesn't correspond to $\hat{g}_x$ either; it corresponds to $\hat{g}_x h_x$. Ok, so take bh_x^{-1}. But h_x^{-1} doesn't exist. So I take the pseudo inverse. But now I'm very far from where I want to be. I can also force h_x^{-1} to exist by simply assuming that no exogenous process is iid.

- Another thing I try is simply to set $s^e_{t+1} = y^e_t$. But for some reason that's not giving me the same thing. I'm relying on this one for now because this is the only one that results in the simulated endogenous state equaling the respective jump. (The figures correspond to this version.)

5 Reference plots

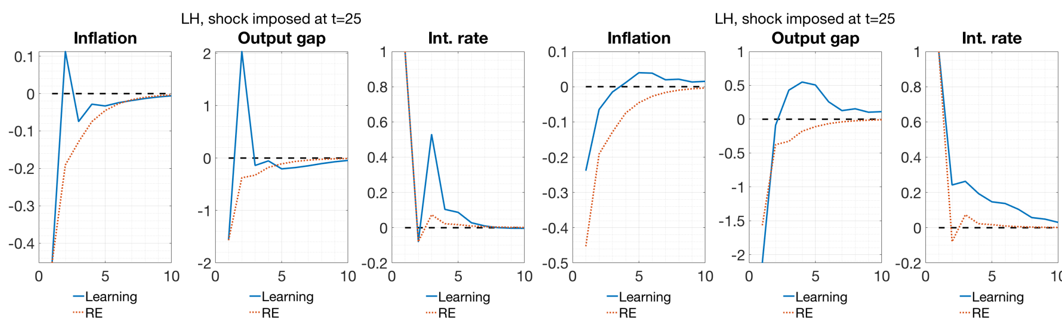
Figure 7: Baseline



(a) Learning constant only

(b) Learning slope and constant

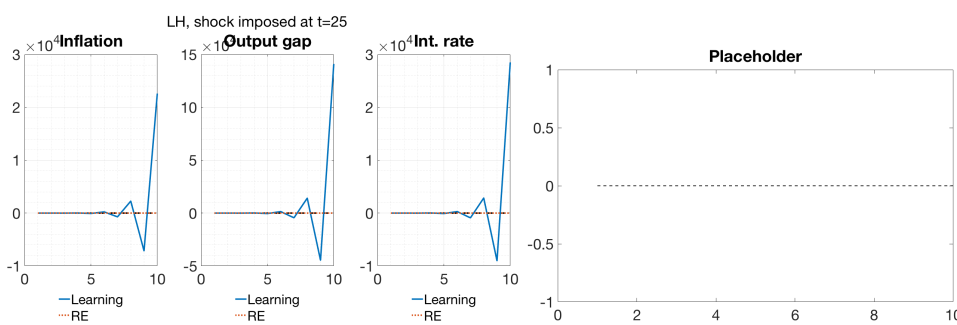
Figure 8: Lagged inflation in Taylor rule, "suboptimal forecasters" info assumption



(a) Learning constant only

(b) Learning slope and constant

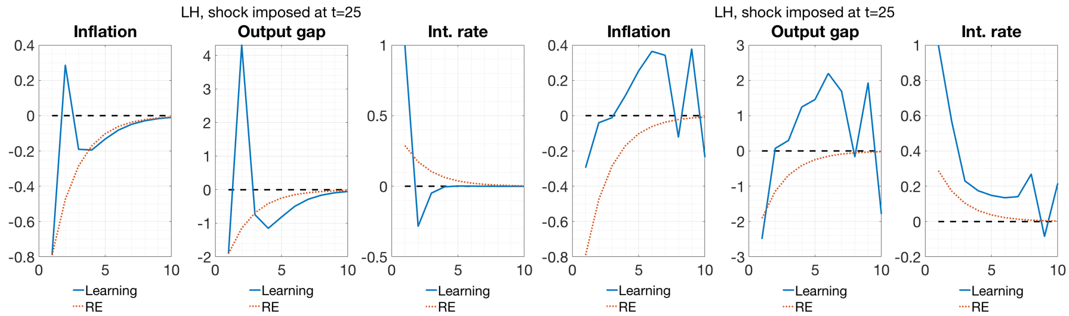
Figure 9: Interest rate smoothing, "suboptimal forecasters" info assumption



(a) Learning constant only

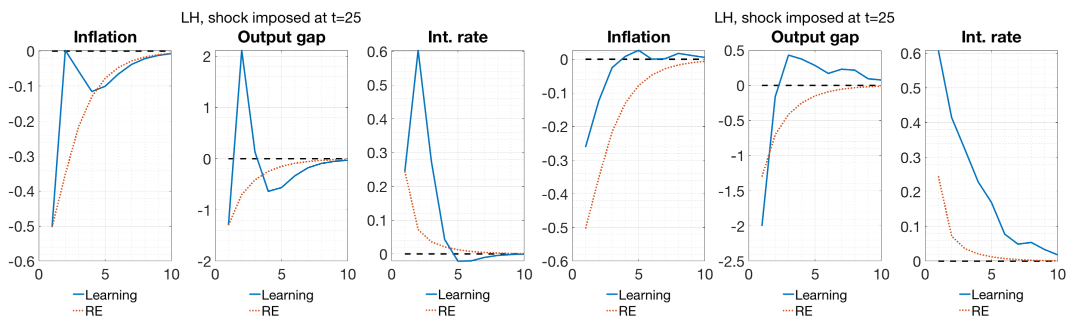
(b) Learning slope and constant - no projection facility in the world can make this not explode

Figure 10: Expected inflation in Taylor rule



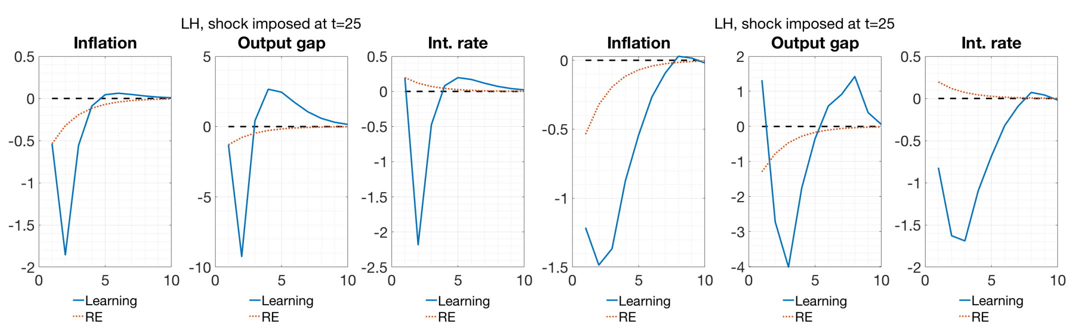
(a) Learning constant only (b) Learning slope and constant

Figure 11: Indexation in NKPC, "suboptimal forecasters" info assumption



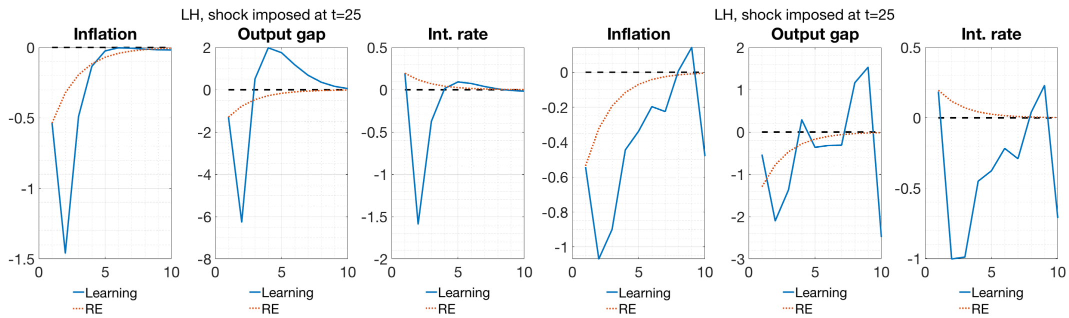
(a) Learning constant only (b) Learning slope and constant

Figure 12: Learn Taylor rule



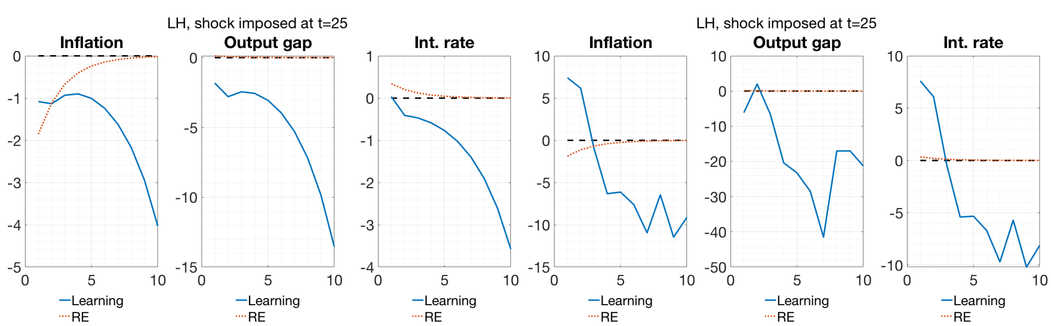
(a) Learning constant only (b) Learning slope and constant

Figure 13: Learn h_x



(a) Learning constant only (b) Learning slope and constant

Figure 14: Markov-switching Taylor rule, conditional on passive regime only, learning initialized at passive regime



(a) Learning constant only (b) Learning slope and constant