Materials 41 - Need large forecast errors for identification

Laura Gáti

August 19, 2020

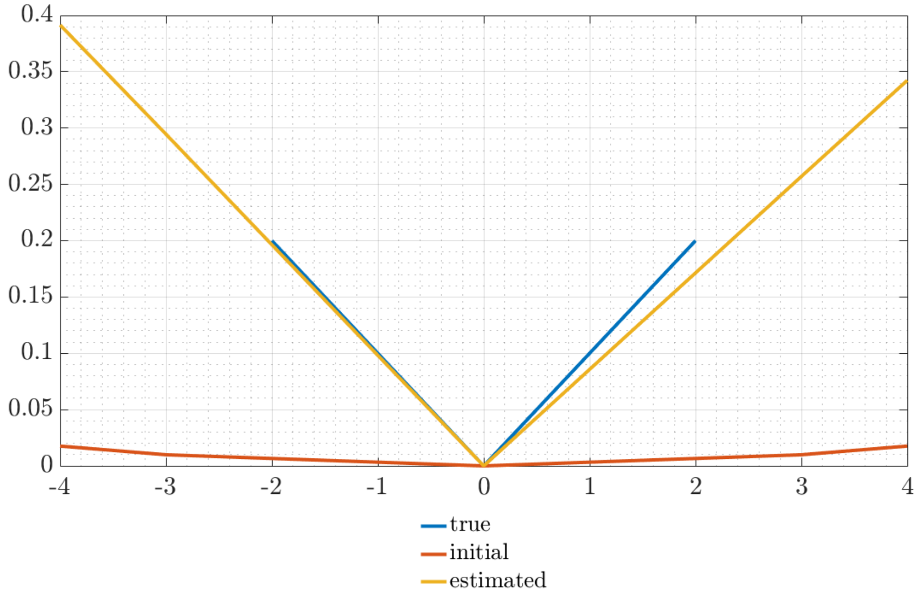
Overview

1 What I need for identification

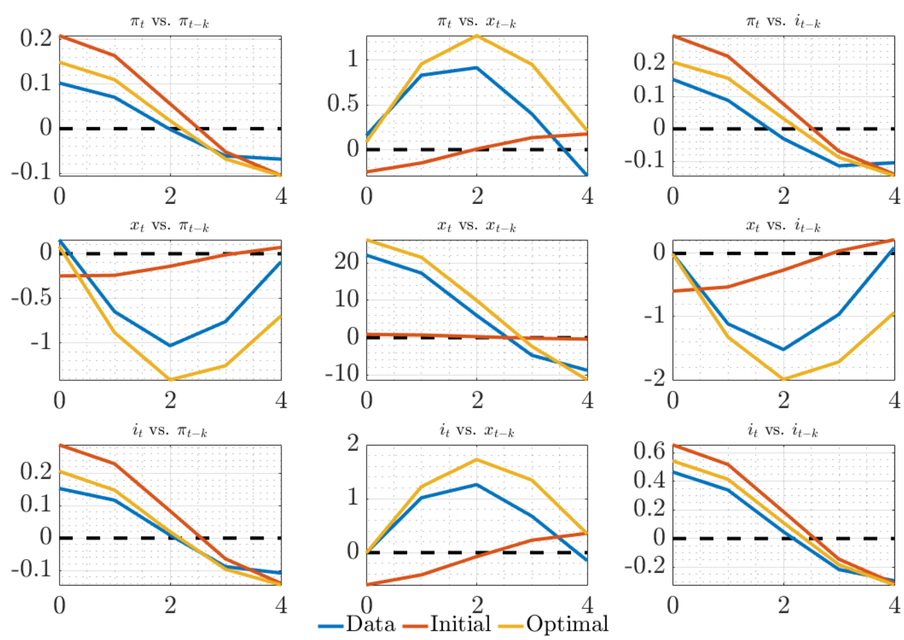
- Need approximating coefficients to pertain to large forecast errors (αs out in the edges)
- Need those large forecast errors to occur in the sample
  - scaled up the "true" αs
  - could play around with variance of shocks, σ_i^2

2 Only α corresponding to large forecast errors are identified - Figure 15 from Materials 40

Figure 1: Not using 1-step ahead forecasts of inflation, estimate mean moments once, imposing convexity with weight 100K, w/o measurement error, truth with $nfe = 5, fe \in (-2, 2)$, gridpoints $= [-4, -3, 0, 3, 4]$ with 0 at 0 imposed with weight 1000, true parameters scaled up by 4



(a) Estimated parameters

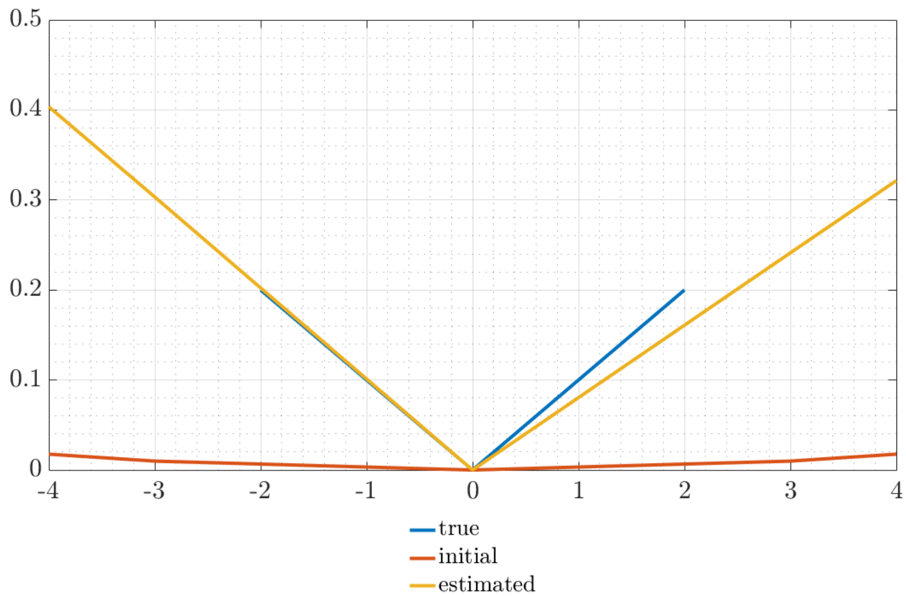


(b) Autocovariogram

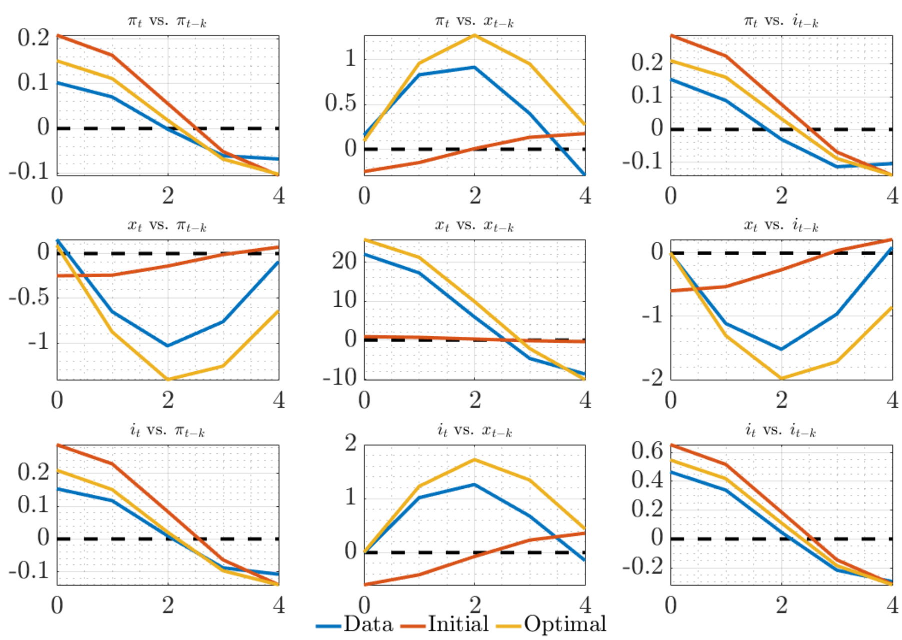
3 GMM weighting matrix mystery solved: I didn't scale the weight on the convexity moments

3.1 Taking square root of elements of W

Figure 2: Not using 1-step ahead forecasts of inflation, estimate mean moments once, imposing convexity with weight 100K, w/o measurement error, truth with $nfe = 5, fe \in (-2, 2)$, gridpoints = $[-4, -3, 0, 3, 4]$ with 0 at 0 imposed with weight 1000, true parameters scaled up by 4, taking square root of elements of W

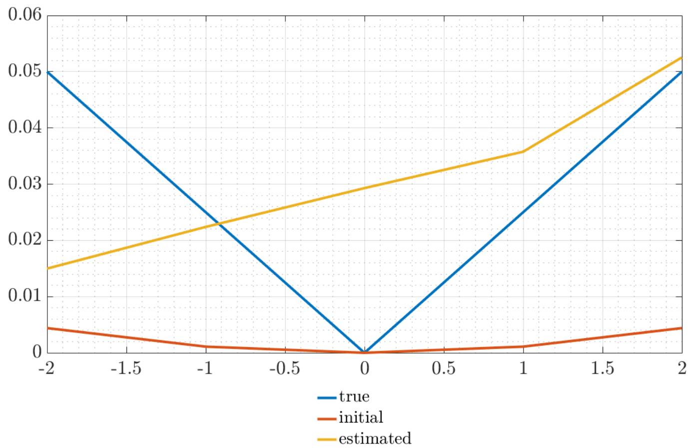


(a) Estimated parameters

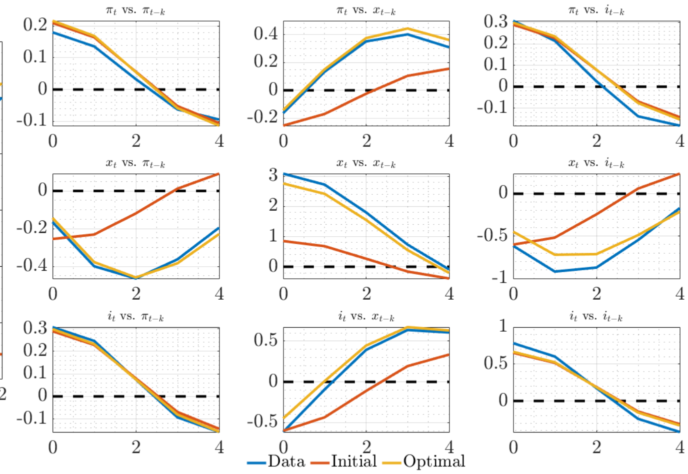


(b) Autocovariogram

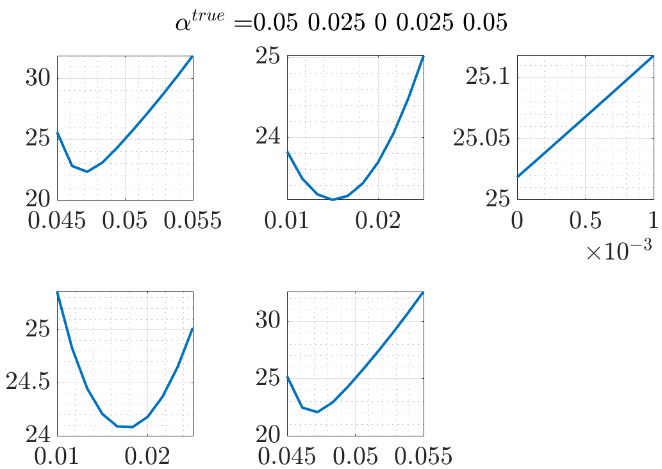
Figure 3: Not using 1-step ahead forecasts of inflation, estimate mean moments once, imposing convexity with weight 100K, w/o measurement error, truth with $nfe = 5, fe \in (-2, 2)$, taking square root of elements of W (This is to be compared with the default Nsimulations figure, and then to replace it as default.)



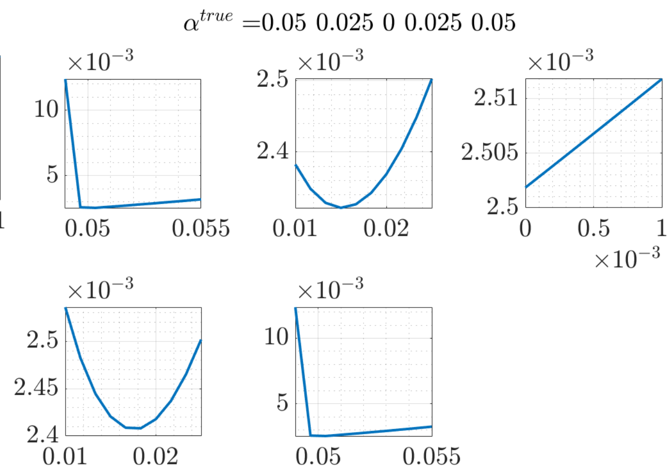
(a) Estimated parameters



(b) Autocovariogram



(c) Loss function



(d) Loss function, scale W

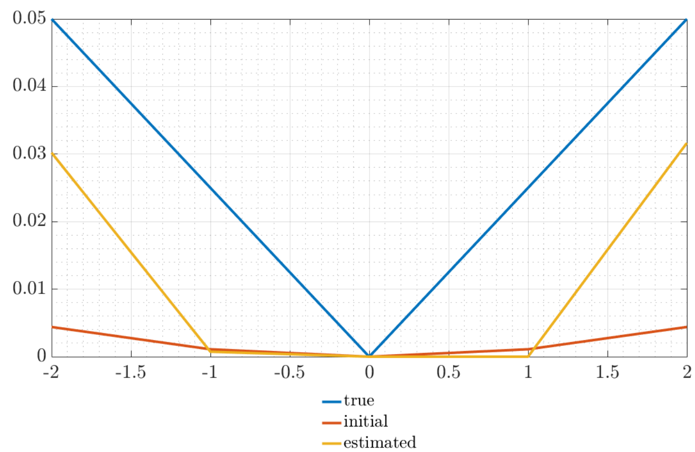
- Rescaling is still doing the same thing to the loss as before: it pushes $\alpha_{1,5}$ up to the true value and makes the loss function nonsmooth. Why?

$\rightarrow$ I've got why! It's the convexity restriction whose weight stays constant when I rescale, so effectively, the convexity moments become relatively more important when I rescale!

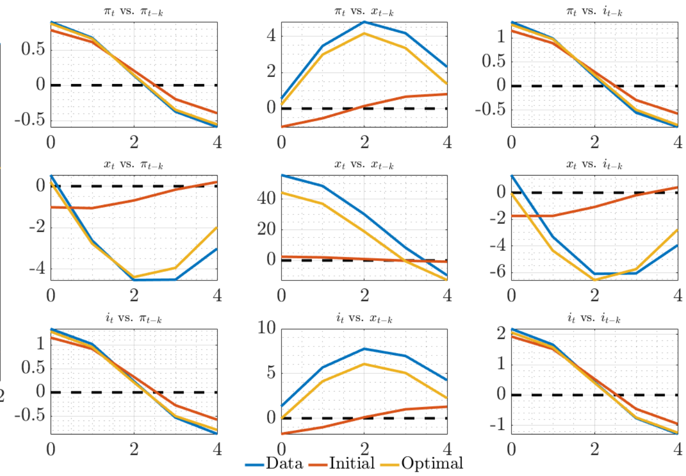
- How do I know? I've plotted the losses w/ and w/o rescaling when setting the weight on the convexity moment to zero and I see absolutely no change on the shape of the loss function!

4 Increasing variance of shocks

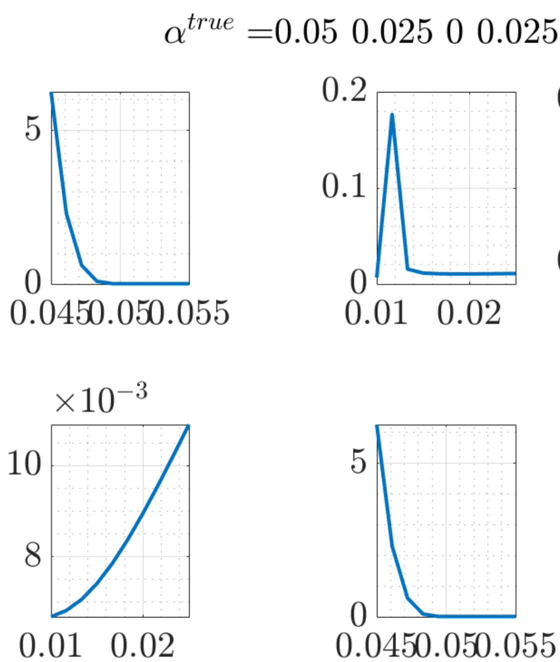
Figure 4: Not using 1-step ahead forecasts of inflation, estimate mean moments once, imposing convexity with weight 100K, truth with $nfe = 5, fe \in (-2, 2)$, taking square root of elements of W, $\sigma_u = 2$, ridge regression with $\lambda = 0.001$ for data generation and estimation.



(a) Estimated parameters

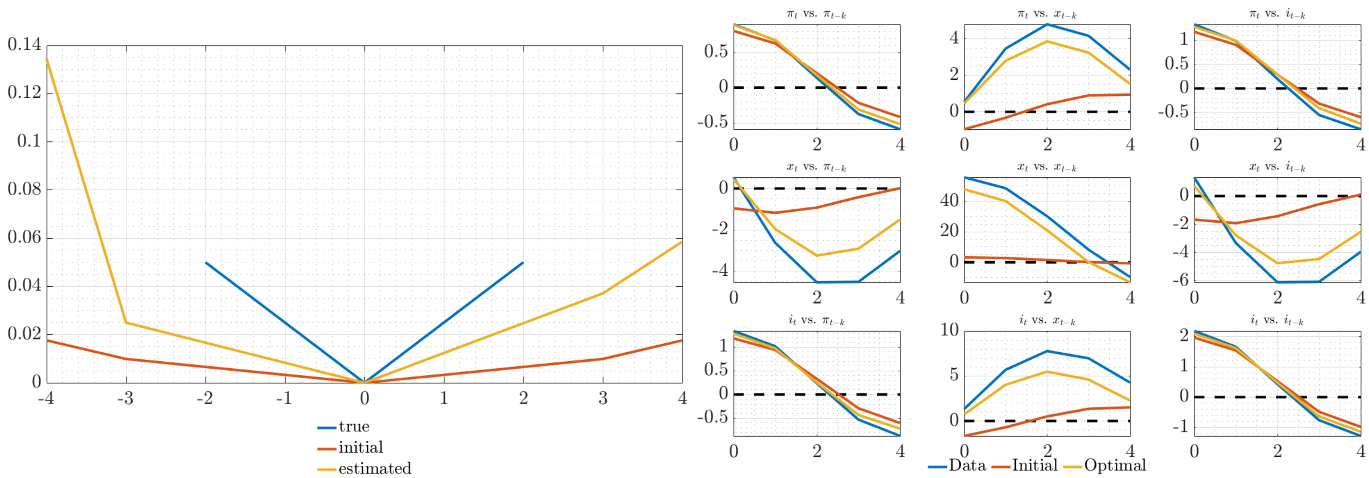


(b) Autocovariogram



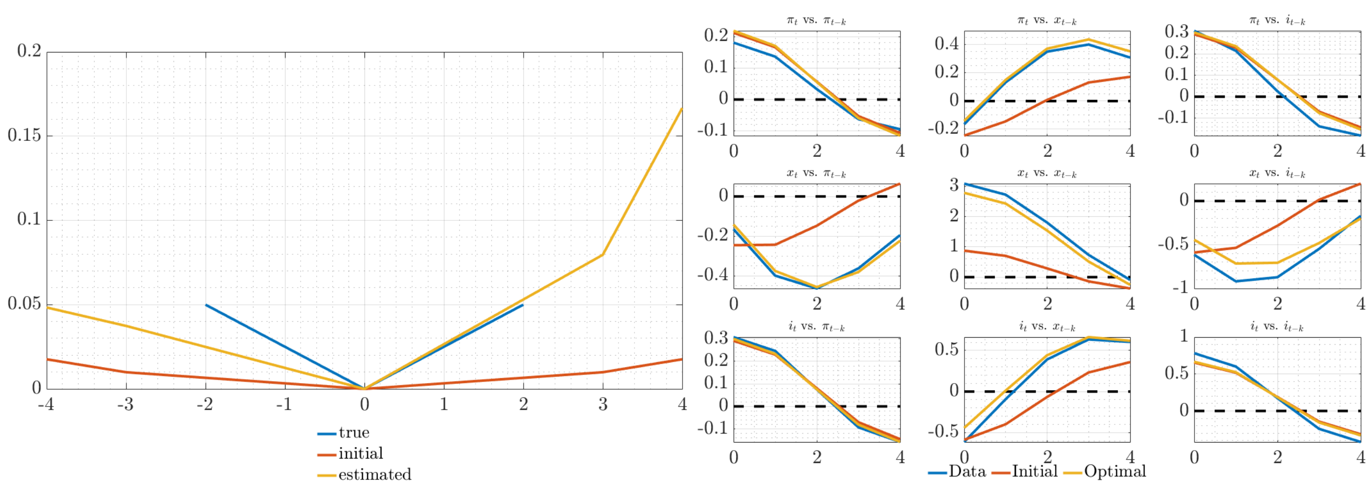
(c) Loss function

Figure 5: Not using 1-step ahead forecasts of inflation, estimate mean moments once, imposing convexity with weight 100K, truth with $nfe = 5, fe \in (-2, 2)$, taking square root of elements of W, $\sigma_u = 2$, ridge regression with $\lambda = 0.001$ for data generation and estimation, gridpoints $= [-4, -3, 0, 3, 4]$ with 0 at 0 imposed with weight 1000



(a) Estimated parameters (b) Autocovariogram

Figure 6: Not using 1-step ahead forecasts of inflation, estimate mean moments once, imposing convexity with weight 100K, truth with $nfe = 5, fe \in (-2, 2)$, taking square root of elements of W, gridpoints $= [-4, -3, 0, 3, 4]$ with 0 at 0 imposed with weight 1000

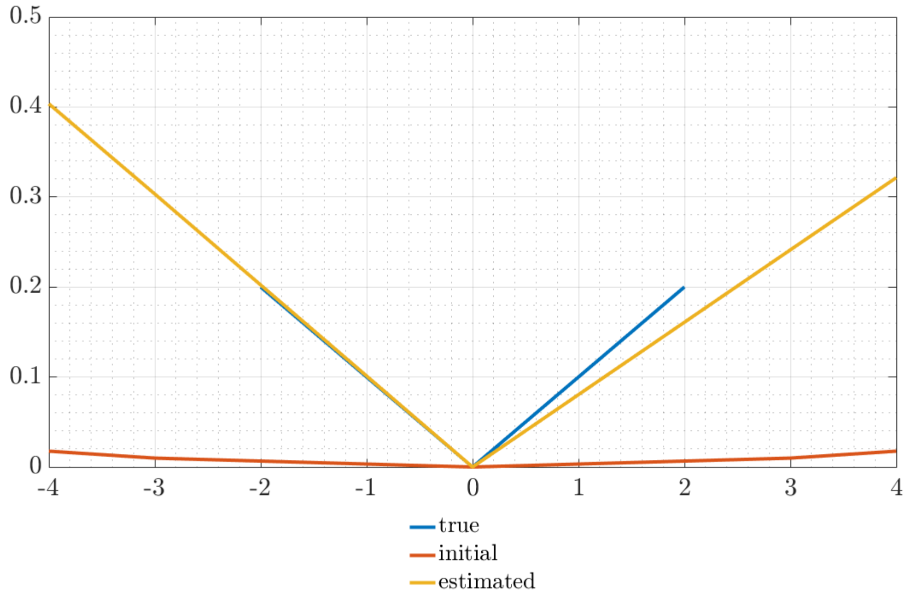


(a) Estimated parameters (b) Autocovariogram

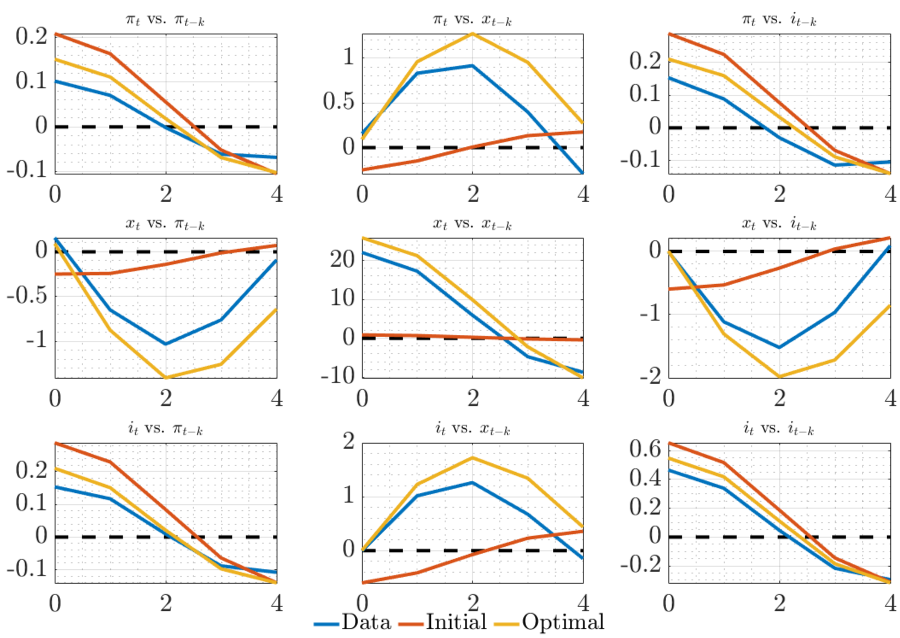
5 Trying to estimate shock variances too

There are three shocks: natural rate shock, cost-push shock and monetary policy shock. I'm fixing the truth to $\sigma_j = 1, \quad j =, r, u, i$, and using $\alpha^{true} = [0.2, 0.1, 0, 0.1, 0.2]$ (the original "truth" scaled up by a factor of 4).

Figure 7: Reference Fig: Not using 1-step ahead forecasts of inflation, estimate mean moments once, taking square root of elements of W, imposing convexity with weight 100K, with 0 at 0 imposed with weight 1000, gridpoints $= [-4, -3, 0, 3, 4]$ $\alpha^{true} = [0.2, 0.1, 0, 0.1, 0.2]$ at fe=[-2,1,0,1,2].

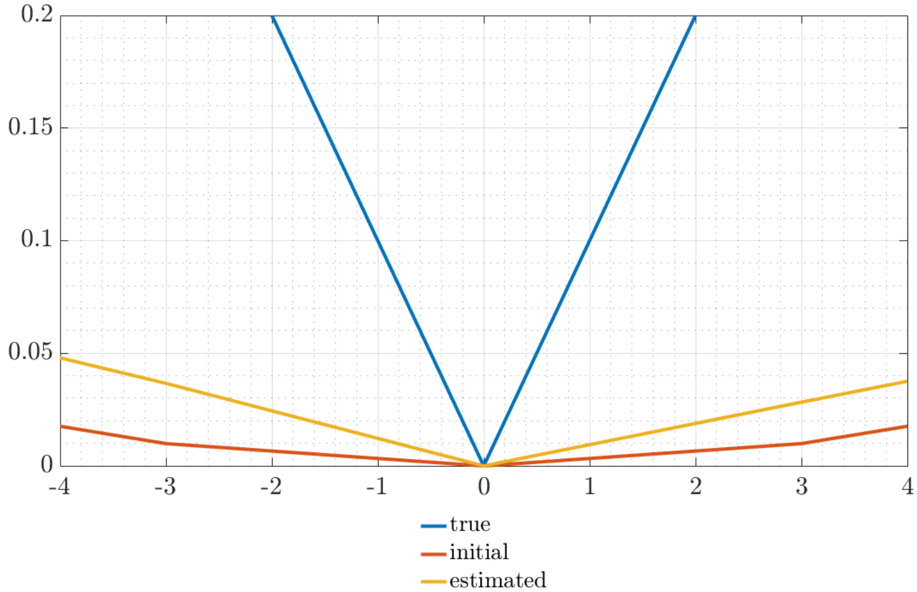


(a) Estimated parameters

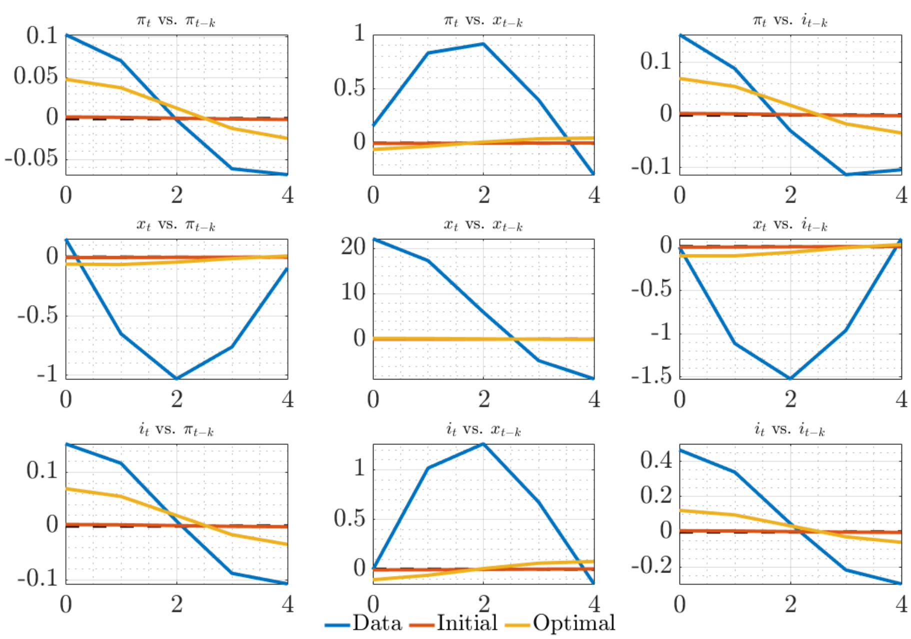


(b) Autocovariogram

Figure 8: Not using 1-step ahead forecasts of inflation, estimate mean moments once, taking square root of elements of W, imposing convexity with weight 100K, with 0 at 0 imposed with weight 1000, gridpoints = $[-4, -3, 0, 3, 4]$ $\alpha^{true} = [0.2, 0.1, 0, 0.1, 0.2]$ at fe=[-2,1,0,1,2], estimating σ_j too



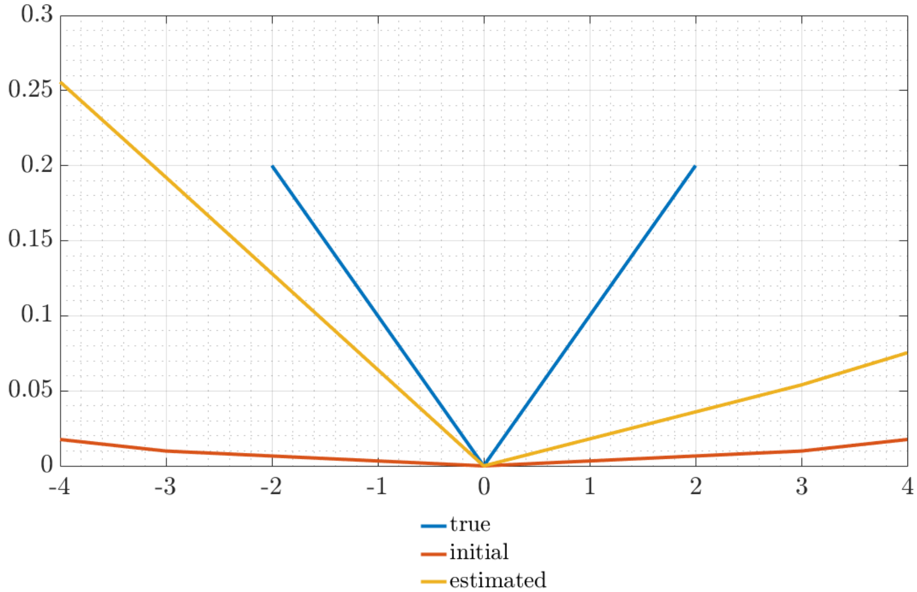
(a) Estimated parameters, $\hat{\sigma_j} = 0.1\ \forall j$ (= initial value)



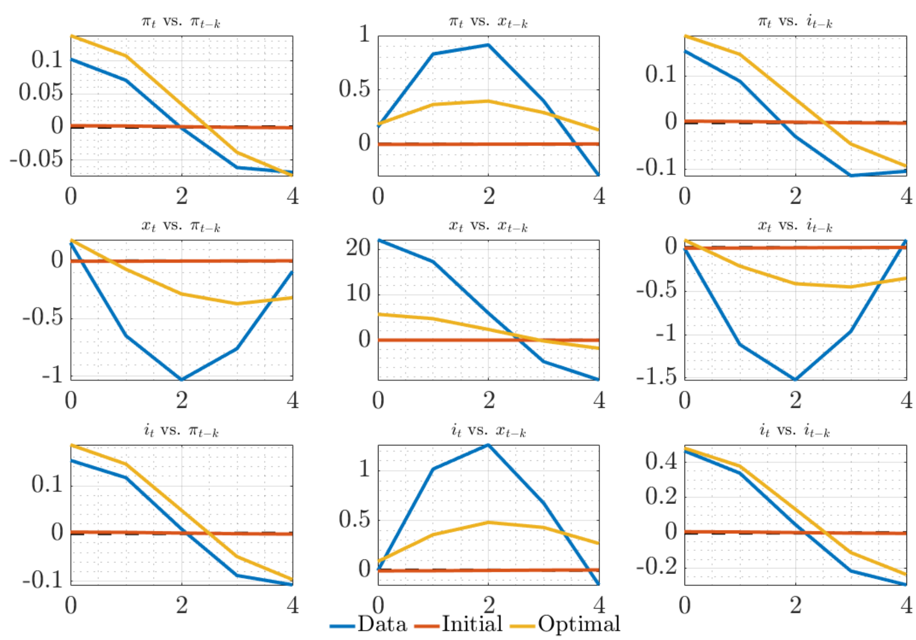
(b) Autocovariogram

Stopped prematurely! The $\hat{\sigma}_j \approx 0.4$. Increasing `MaxFunEvals` 3 times, and still iterates out. Let's try 100 times the default finite difference step size.

Figure 9: Not using 1-step ahead forecasts of inflation, estimate mean moments once, taking square root of elements of W, imposing convexity with weight 100K, with 0 at 0 imposed with weight 1000, gridpoints $= [-4, -3, 0, 3, 4]$ $\alpha^{true} = [0.2, 0.1, 0, 0.1, 0.2]$ at fe=[-2,1,0,1,2], estimating σ_j too, $h \equiv$ finite difference step size = 100K$\times$ default



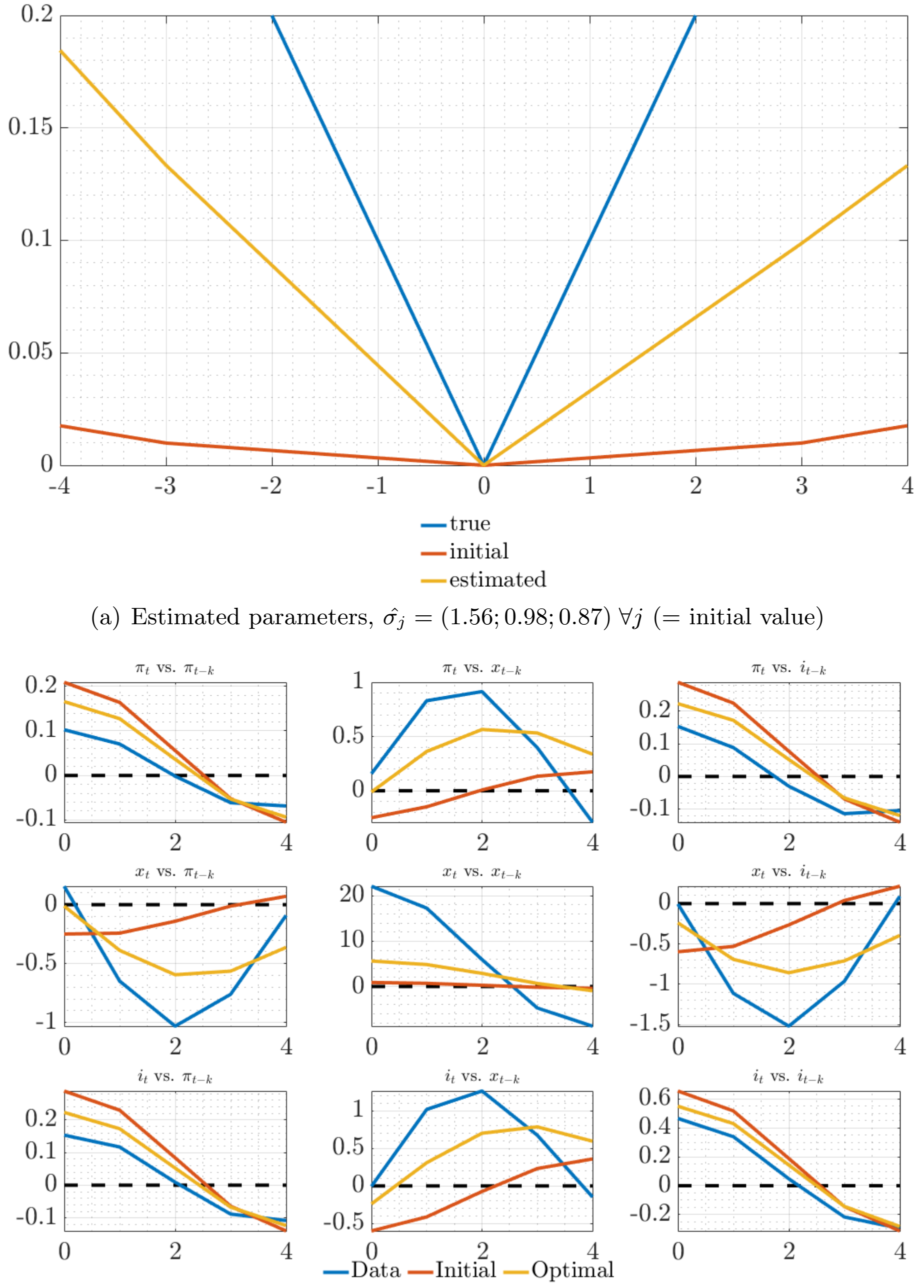
(a) Estimated parameters, $\hat{\sigma_j} = (3.53; 0.94; 0.75)\ \forall j$ (= initial value)



(b) Autocovariogram

Improvement doesn't continue if I raise the finite difference step size to 1M$\times$default. I then get no movement from initial α_0, and I get $\hat{\sigma_j} = (3.94; 1; 0.64)$. I also tried having h = 1M$\times$default for σ_j, but default for the αs. Surprisingly, that doesn't do as well as having a uniformly larger stepsize. The reverse does almost as well as a uniformly larger h.

Figure 10: Not using 1-step ahead forecasts of inflation, estimate mean moments once, taking square root of elements of W, imposing convexity with weight 100K, with 0 at 0 imposed with weight 1000, gridpoints $= [-4, -3, 0, 3, 4]$ $\alpha^{true} = [0.2, 0.1, 0, 0.1, 0.2]$ at fe=[-2,1,0,1,2], estimating σ_j too, $h \equiv$ finite difference step size = 100K$\times$ default, initialize σ_j at truth, 1

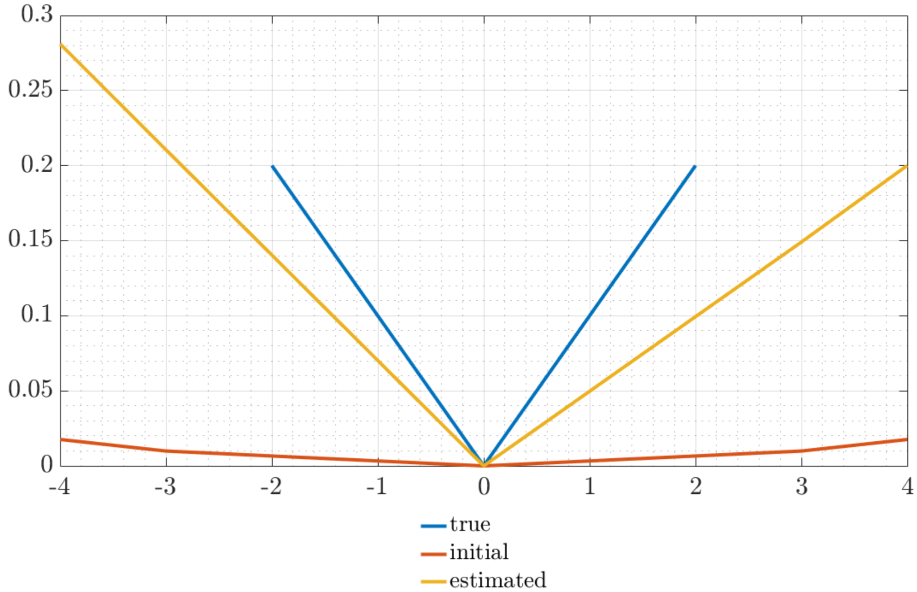


(a) Estimated parameters, $\hat{\sigma_j} = (1.56; 0.98; 0.87)\ \forall j$ (= initial value)

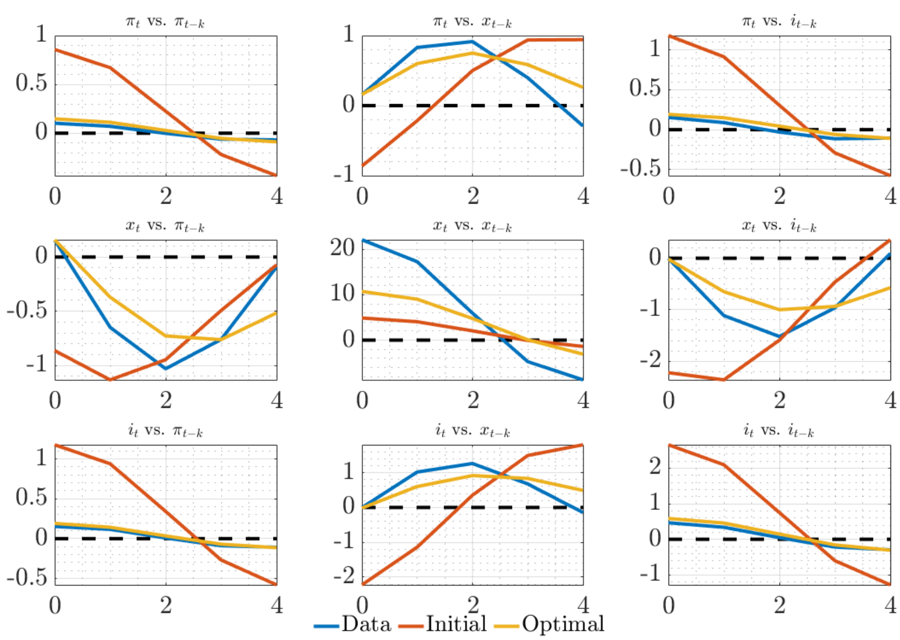
(b) Autocovariogram

Still did not converge.

Figure 11: Not using 1-step ahead forecasts of inflation, estimate mean moments once, taking square root of elements of W, imposing convexity with weight 100K, with 0 at 0 imposed with weight 1000, gridpoints $= [-4, -3, 0, 3, 4]$ $\alpha^{true} = [0.2, 0.1, 0, 0.1, 0.2]$ at fe=[-2,1,0,1,2], estimating σ_j too, $h \equiv$ finite difference step size = 100K$\times$ default, initialize σ_j at 2



(a) Estimated parameters, $\hat{\sigma_j} = (2.87; 1.14; 0.83)$ $\forall j$ (= initial value)



(b) Autocovariogram

A Model summary

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

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

$$i_t = \psi_\pi \pi_t + \psi_x x_t + \bar{i}_t \qquad \text{(if imposed)} \tag{A.3}$$

$$\text{PLM:} \qquad \hat{\mathbb{E}}_t z_{t+h} = a_{t-1} + b h_x^{h-1} s_t \quad \forall h \geq 1 \qquad b = g_x\, h_x \tag{A.4}$$

$$\text{Updating:} \qquad a_t = a_{t-1} + k_t^{-1}\big(z_t - (a_{t-1} + b s_{t-1})\big) \tag{A.5}$$

$$\text{Anchoring function:} \qquad k_t^{-1} = \rho_k k_{t-1}^{-1} + \gamma_k f e_{t-1}^2 \tag{A.6}$$

$$\text{Forecast error:} \qquad f e_{t-1} = z_t - (a_{t-1} + b s_{t-1}) \tag{A.7}$$

$$\text{LH expectations:} \qquad f_a(t) = \frac{1}{1-\alpha\beta} a_{t-1} + b(\mathbb{I}_{nx} - \alpha\beta h)^{-1} s_t \qquad f_b(t) = \frac{1}{1-\beta} a_{t-1} + b(\mathbb{I}_{nx} - \beta h)^{-1} s_t \tag{A.8}$$

This notation captures vector learning (z learned) for intercept only. For scalar learning, $a_t = \begin{pmatrix} \bar{\pi}_t & 0 & 0 \end{pmatrix}'$ and b_1 designates the first row of b. The observables (π, x) are determined as:

$$x_t = -\sigma i_t + \begin{bmatrix} \sigma & 1-\beta & -\sigma\beta \end{bmatrix} f_b + \sigma \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} (\mathbb{I}_{nx} - \beta h_x)^{-1} s_t \tag{A.9}$$

$$\pi_t = \kappa x_t + \begin{bmatrix} (1-\alpha)\beta & \kappa\alpha\beta & 0 \end{bmatrix} f_a + \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} (\mathbb{I}_{nx} - \alpha\beta h_x)^{-1} s_t \tag{A.10}$$

B Target criterion

The target criterion in the simplified model (scalar learning of inflation intercept only, $k_t^{-1} = \mathbf{g}(f e_{t-1})$):

$$\pi_t = -\frac{\lambda_x}{\kappa}\Bigg\{ x_t - \frac{(1-\alpha)\beta}{1-\alpha\beta}\Big(k_t^{-1} + ((\pi_t - \bar{\pi}_{t-1} - b_1 s_{t-1}))\mathbf{g}_\pi(t)\Big) \\ \Big(\mathbb{E}_t \sum_{i=1}^{\infty} x_{t+i} \prod_{j=0}^{i-1} (1 - k_{t+1+j}^{-1} - (\pi_{t+1+j} - \bar{\pi}_{t+j} - b_1 s_{t+j})\mathbf{g}_{\bar{\pi}}(t+j))\Big)\Bigg\} \tag{B.1}$$

where I'm using the notation that $\prod_{j=0}^{0} \equiv 1$. For interpretation purposes, let me rewrite this as follows:

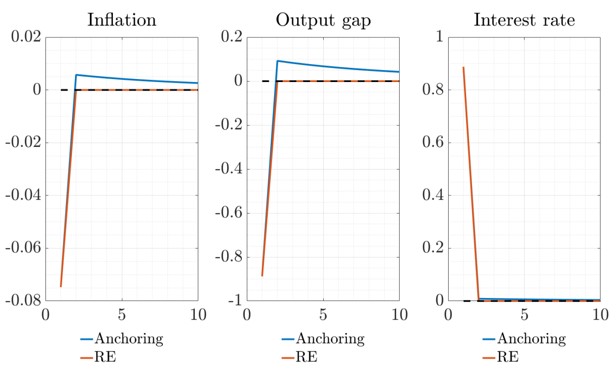
$$\pi_t = -\frac{\lambda_x}{\kappa} x_t + \frac{\lambda_x}{\kappa}\frac{(1-\alpha)\beta}{1-\alpha\beta}\Big(k_t^{-1} + f e_{t|t-1}^{eve}\mathbf{g}_\pi(t)\Big)\mathbb{E}_t \sum_{i=1}^{\infty} x_{t+i} \\ -\frac{\lambda_x}{\kappa}\frac{(1-\alpha)\beta}{1-\alpha\beta}\Big(k_t^{-1} + f e_{t|t-1}^{eve}\mathbf{g}_\pi(t)\Big)\Big(\mathbb{E}_t \sum_{i=1}^{\infty} x_{t+i} \prod_{j=0}^{i-1} (k_{t+1+j}^{-1} + f e_{t+1+j|t+j}^{eve})\mathbf{g}_{\bar{\pi}}(t+j)\Big) \tag{B.2}$$

Interpretation: tradeoffs from discretion in RE + effect of current level and change of the gain on future tradeoffs + effect of future expected levels and changes of the gain on future tradeoffs

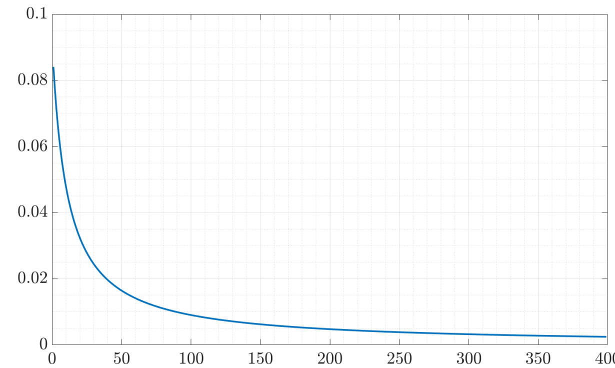
C Impulse responses to iid monpol shocks across a wide range of learning models

$T = 400, N = 100, n_{drop} = 5$, shock imposed at $t = 25$, calibration as above, Taylor rule assumed to be known, PLM = learn constant only, of inflation only.

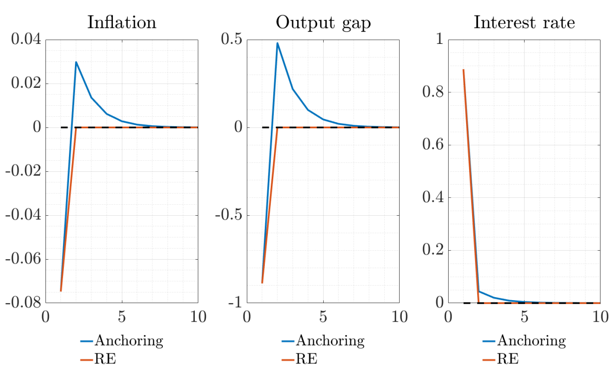
Figure 12: IRFs and gain history (sample means)



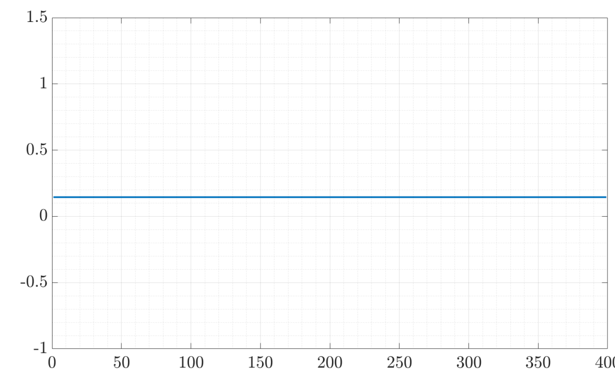
(a) Decreasing gain learning



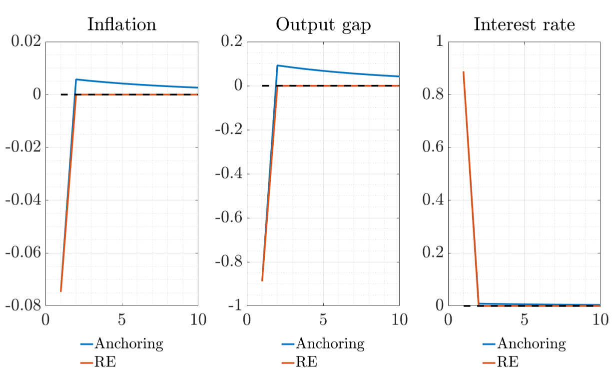
(b) Mean gain



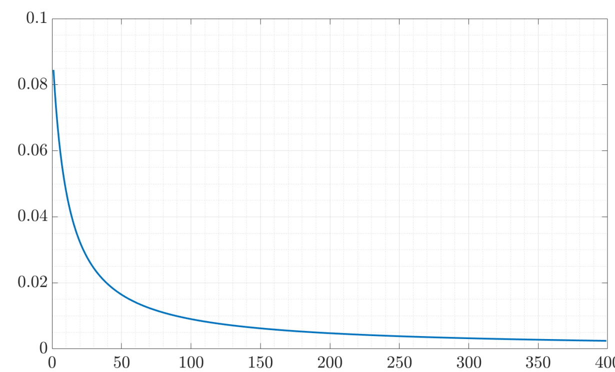
(c) Constant gain learning



(d) Mean gain

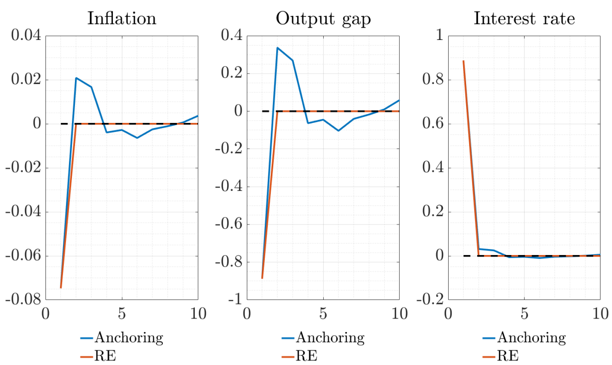


(e) CEMP criterion (vector)

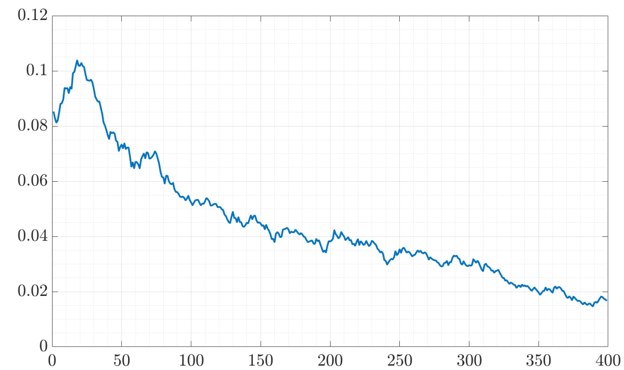


(f) Mean gain

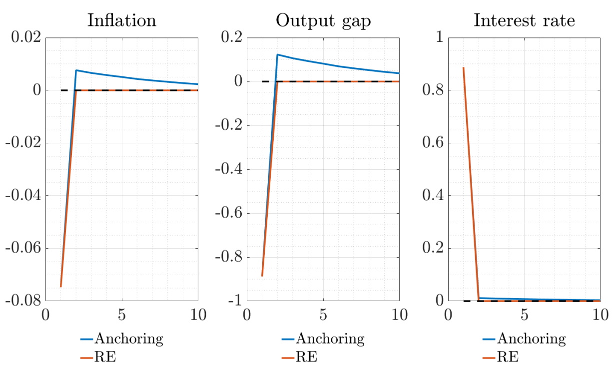
Figure 13: IRFs and gain history (sample means), continued



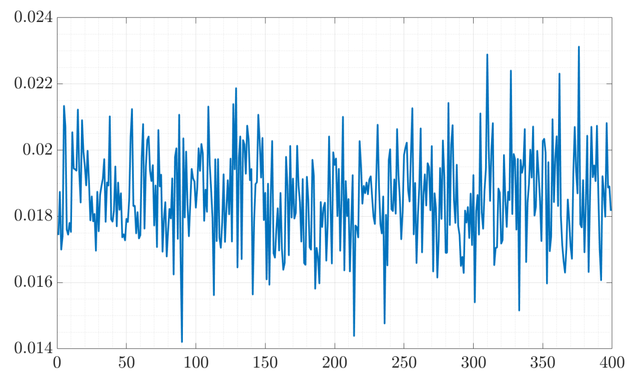
(a) CUSUM criterion (vector)



(b) Mean gain



(c) Smooth criterion, approximated, using $\alpha^{true} = (0.05; 0.025; 0; 0.025; 0.05)$, on $fe \in (-2, 2)$.



(d) Mean gain