

Knowing the Forecasts of Others

Thomas J. Sargent and John Stachurski

April 20, 2021

1 Contents

- Introduction 2
- The Setting 3
- Tactics 4
- Equilibrium conditions 5
- Equilibrium with θ_t stochastic but observed at t 7
- Equilibrium with one signal on θ_t 7
- Equilibrium with two noisy signals on θ_t 10
- Notes on History of the Problem 11

In addition to what's in Anaconda, this lecture will need the following libraries:

```
In [1]: !pip install --upgrade quantecon
!conda install -y -c plotly plotly plotly-orca
```

2 Introduction

Robert E. Lucas, Jr. [5], Kenneth Kasa [3], and Robert Townsend [9] showed that giving decision makers incentives to infer persistent hidden state variables from equilibrium prices and quantities can elongate and amplify impulse responses to aggregate shocks in business cycle models.

Townsend [9] noted that such incentives can naturally induce decision makers to want to forecast the forecast of others.

This theme has been pursued and extended in analyses in which decision makers' imperfect information forces them into pursuing an infinite recursion of forming beliefs about the beliefs of other (e.g., [1]).

Lucas [5] side stepped having decision makers forecast the forecasts of other decision makers by assuming that they simply pool their information before forecasting.

A **pooling equilibrium** like Lucas's plays a prominent role in this lecture.

Because he didn't assume such pooling, [9] confronted the forecasting the forecasts of others problem.

To formulate the problem recursively required that Townsend define decision maker's **state** vector.

Townsend concluded that his original model required an intractable infinite dimensional state space.

Therefore, he constructed a more manageable approximating model in which the hidden Markov component of the demand shock is revealed to all firms after a fixed and finite number of periods.

In this lecture, as yet another instance of the theme that **finding the state is an art**, we show how to formulate Townsend's original model in terms of a low-dimensional state space.

By doing so, we show that Townsend's model shares equilibrium prices and quantities with those that prevail in a pooling equilibrium.

That finding emerged from a line of research about Townsend's model that culminated in [?] that built on [4].

However, rather than deploying the [4] machinery here, we shall rely instead on a sneaky **guess-and-verify** tactic.

- We compute a pooling equilibrium and represent it as an instance of a linear state-space system provided by the Python class `quantecon.LinearStateSpace`.
- Leaving the state-transition equation for the pooling equilibrium unaltered, we alter the observation vector for a firm to what it is in Townsend's original model. So rather than directly observing the signal received by firms in the other industry, a firm sees the equilibrium price of the good produced by the other industry.
- We compute a population linear least squares regression of the noisy signal that firms in the other industry receive in a pooling equilibrium on time t information that a firm receives in Townsend's original model. The R^2 in this regression equals 1. That verifies that a firm's information set in Townsend's original model equals its information set in a pooling equilibrium. Therefore, equilibrium prices and quantities in Townsend's original model equal those in a pooling equilibrium.

2.1 A Sequence of Models

We proceed by describing a sequence of models of two industries that are linked in a single way: shocks to the demand curves for their products have a common component.

The models are simplified versions of Townsend's [9].

Townsend's is a model of a rational expectations equilibrium in which firms confront the problem **forecasting the forecasts of others**.

In Townsend's model, firms condition their forecasts on observed endogenous variables whose equilibrium laws of motion are determined by their own forecasting functions.

We start with model components that we shall progressively assemble in ways that can help us to appreciate the structure of a **pooling equilibrium** that ultimately concerns us.

While keeping other aspects of the model the same, we shall study consequences of alternative assumptions about what decision makers observe.

Technically, this lecture deploys concepts and tools that appear in [First Look at Kalman Filter](#) and [Rational Expectations Equilibrium](#).

3 The Setting

We cast all variables in terms of deviations from means.

Therefore, we omit constants from inverse demand curves and other functions.

Firms in each of two industries $i = 1, 2$ use a single factor of production, capital k_t^i , to produce output of a single good, y_t^i .

Firms bear quadratic costs of adjusting their capital stocks.

A representative firm in industry i has production function $y_t^i = f k_t^i$, $f > 0$, acts as a price taker with respect to output price P_t^i , and maximizes

$$E_0^i \sum_{t=0}^{\infty} \beta^t \{P_t^i f k_t^i - .5h(k_{t+1}^i - k_t^i)^2\}, \quad h > 0. \quad (1)$$

Demand in industry i is described by the inverse demand curve

$$P_t^i = -bY_t^i + \theta_t + \epsilon_t^i, \quad b > 0, \quad (2)$$

where P_t^i is the price of good i at t , $Y_t^i = fK_t^i$ is output in market i , θ_t is a persistent component of a demand shock that is common across the two industries, and ϵ_t^i is an industry specific component of the demand shock that is i.i.d. and whose time t marginal distribution is $\mathcal{N}(0, \sigma_\epsilon^2)$.

We assume that θ_t is governed by

$$\theta_{t+1} = \rho\theta_t + v_t \quad (3)$$

where $\{v_t\}$ is an i.i.d. sequence of Gaussian shocks each with mean zero and variance σ_v^2 .

To simplify notation, we'll study a special case of the model by setting $h = f = 1$.

The presence of costs of adjusting their capital stocks imparts to firms an incentives to forecast the price of the good that they sell.

Throughout, we use the **rational expectations** equilibrium concept presented in this lecture [Rational Expectations Equilibrium](#).

We let capital letters denote market wide objects and lower case letters denote objects chosen by a representative firm.

In each industry, a competitive equilibrium prevails.

To rationalize the big K , little k connection, we can think of there being a continua of each type of firm, each indexed by $\omega \in [0, 1]$ with $K^i = \int_0^1 k^i(\omega) d\omega$.

In equilibrium, $k_t^i = K_t^i$, but as usual we must distinguish between k_t^i and K_t^i when we pose the firm's optimization problem.

4 Tactics

We shall compute equilibrium laws of motion for capital in industry i under a sequence of assumptions about what a representative firm observes.

Successive members of this sequence make a representative firm's information more and more obscure.

We begin with the most information, then gradually withdraw information in a way that approaches and eventually reaches the information structure that that we are ultimately interested in.

Thus, we shall compute equilibria under the following alternative information structures:

- **Perfect foresight:** future values of θ_t, ϵ_t^i are observed in industry i .
- **Observed but stochastic θ_t :** $\{\theta_t, \epsilon_t^i\}$ are realizations from a stochastic process; current and past values of each are observed at time t but future values are not.
- **One noise-ridden observation on θ_t :** Values of $\{\theta_t, \epsilon_t^i\}$ separately are never observed. However, at time t , a history w^t of a scalar noise-ridden observations on θ_t is observed at time t .
- **Two noise-ridden observations on θ_t :** Values of $\{\theta_t, \epsilon_t^i\}$ separately are never observed. However, at time t , a history w^t of *two* noise-ridden observations on θ_t is observed at time t .

Successive computations build one on another.

We proceed by first finding an equilibrium under perfect foresight.

To compute an equilibrium with θ_t observed, we use a *certainty equivalence principle* to justify modifying the perfect foresight equilibrium by replacing future values of $\theta_s, \epsilon_s^i, s \geq t$ with mathematical expectations conditioned on θ_t .

This provides the equilibrium when θ_t is observed at t but future θ_{t+j} and ϵ_{t+j}^i are not observed.

To find an equilibrium when only a history w_t of a single noise ridden observations on θ_t is observed, we again apply a certainty equivalence principle and replace future values of the random variables $\theta_s, \epsilon_s^i, s \geq t$ with their mathematical expectations conditioned on w^t .

To find an equilibrium when only a history w_t of a *two* noisy signal on θ_t is observed, we replace future values of the random variables $\theta_s, \epsilon_s^i, s \geq t$ with their mathematical expectations conditioned on history w^t .

We call the equilibrium with two noise-ridden observations on θ_t a **pooling equilibrium**.

- It corresponds to an arrangement in which at the beginning of each period firms in industries 1 and 2 somehow get together and share information about current values of their noisy signals on θ .

We want ultimately to compare outcomes in a pooling equilibrium with an equilibrium under the following alternative information structure for a firm in industry i that interested [9]:

- **Firm i 's noise-ridden signal on θ_t and the price in industry $-i$,** a firm in industry i observes a history w^t of *one* noise-ridden signal on θ_t and a history of industry $-i$'s price is observed.

With this information structure, the representative firm i sees the price as well as the aggregate state variable Y_t^i in its own industry.

That allows it to infer the total demand shock $\theta_t + \epsilon_t^i$.

However, at time t , the firm sees only P_t^{-i} and does not see Y_t^{-i} , so that firm i does not directly observe $\theta_t + \epsilon_t^{-i}$.

Nevertheless, it will turn out that equilibrium prices and quantities in this equilibrium equal their counterparts in a pooling equilibrium because firms in industry i are able to infer the noisy signal about the demand shock received by firms in industry $-i$.

We shall eventually verify this assertion by using a guess and verify tactic. Section ??

5 Equilibrium conditions

It is convenient to solve the firm's problem without uncertainty by forming the Lagrangian:

$$J = \sum_{t=0}^{\infty} \beta^t \{ P_t^i k_t^i - .5(\mu_t^i)^2 + \phi_t^i [k_t^i + \mu_t^i - k_{t+1}^i] \}$$

where $\{\phi_t^i\}$ is a sequence of Lagrange multipliers on the transition law for k_{t+1}^i . First order conditions for the nonstochastic problem are

$$\begin{aligned} \phi_t^i &= \beta \phi_{t+1}^i + \beta P_{t+1}^i \\ \mu_t^i &= \phi_t^i. \end{aligned} \tag{4}$$

Substituting the demand function (2) for P_t^i , imposing the condition that the representative firm is representative ($k_t^i = K_t^i$), and using the definition below of g_t^i , the Euler equation (4), lagged by one period, can be expressed as $-bk_t^i + \theta_t + \epsilon_t^i + (k_{t+1}^i - k_t^i) - g_t^i = 0$ or

$$k_{t+1}^i = (b+1)k_t^i - \theta_t - \epsilon_t^i + g_t^i \tag{5}$$

where we define g_t^i by

$$g_t^i = \beta^{-1}(k_t^i - k_{t-1}^i) \tag{6}$$

We can write Euler equation (4) as:

$$g_t^i = P_t^i + \beta g_{t+1}^i \tag{7}$$

In addition, we have the law of motion for θ_t , (3), and the demand equation (2).

In summary, with perfect foresight, equilibrium conditions for industry i include the following system of difference equations:

$$\begin{aligned} k_{t+1}^i &= (1+b)k_t^i - \epsilon_t^i - \theta_t + g_t^i \\ \theta_{t+1} &= \rho \theta_t + v_t \\ g_{t+1}^i &= \beta^{-1}(g_t^i - P_t^i) \\ P_t^i &= -bk_t^i + \epsilon_t^i + \theta_t \end{aligned} \tag{8}$$

Without perfect foresight, the same system prevails except that the following equation replaces the third equation of (8):

$$g_{t+1,t}^i = \beta^{-1}(g_t^i - P_t^i)$$

where $x_{t+1,t}$ denotes the mathematical expectation of x_{t+1} conditional on information at time t .

5.1 Equilibrium under perfect foresight

Our first step is to compute the equilibrium law of motion for k_t^i under perfect foresight.

Let L be the lag operator. Section ??

Equations (7) and (5) imply the second order difference equation in k_t^i : Section ??

$$[(L^{-1} - (1 + b))(1 - \beta L^{-1}) + b] k_t^i = \beta L^{-1} \epsilon_t^i + \beta L^{-1} \theta_t. \quad (9)$$

Factor the polynomial in L on the left side as:

$$-\beta[L^{-2} - (\beta^{-1} + (1 + b))L^{-1} + \beta^{-1}] = \tilde{\lambda}^{-1}(L^{-1} - \tilde{\lambda})(1 - \tilde{\lambda}\beta L^{-1})$$

where $|\tilde{\lambda}| < 1$ is the smaller root and λ is the larger root of $(\lambda - 1)(\lambda - 1/\beta) = b\lambda$.

Therefore, (9) can be expressed as

$$\tilde{\lambda}^{-1}(L^{-1} - \tilde{\lambda})(1 - \tilde{\lambda}\beta L^{-1})k_t^i = \beta L^{-1} \epsilon_t^i + \beta L^{-1} \theta_t.$$

Solving the stable root backwards and the unstable root forwards gives

$$k_{t+1}^i = \tilde{\lambda} k_t^i + \frac{\tilde{\lambda}\beta}{1 - \tilde{\lambda}\beta L^{-1}}(\epsilon_{t+1}^i + \theta_{t+1}).$$

Recall that we have already set $k^i = K^i$ at the appropriate point in the argument (i.e., *after* having derived the first-order necessary conditions for a representative firm in industry i).

Thus, under perfect foresight the equilibrium capital stock in industry i satisfies

$$k_{t+1}^i = \tilde{\lambda} k_t^i + \sum_{j=1}^{\infty} (\tilde{\lambda}\beta)^j (\epsilon_{t+j}^i + \theta_{t+j}). \quad (10)$$

Next, we shall investigate consequences of replacing future values of $(\epsilon_{t+j}^i + \theta_{t+j})$ in equation (10) with alternative forecasting schemes.

In particular, we shall compute equilibrium laws of motion for capital under alternative assumptions about the information available to decision makers in market i .

6 Equilibrium with θ_t stochastic but observed at t

If future θ 's are unknown at t , it is appropriate to replace all random variables on the right side of (10) with their conditional expectations based on the information available to decision makers in market i .

For now, we assume that this information set $I_t^p = [\theta^t \ \epsilon^{it}]$, where z^t represents the infinite history of variable z_s up to time t .

Later we shall give firms less information.

To obtain an appropriate counterpart to (10) under our current assumption about information, we apply a certainty equivalence principle.

In particular, it is appropriate to take (10) and replace each term $(\epsilon_{t+j}^i + \theta_{t+j})$ on the right side with $E[(\epsilon_{t+j}^i + \theta_{t+j})|\theta^t]$.

After using (3) and the i.i.d. assumption about $\{\epsilon_t^i\}$, this gives

$$k_{t+1}^i = \tilde{\lambda}k_t^i + \frac{\tilde{\lambda}\beta\rho}{1 - \tilde{\lambda}\beta\rho}\theta_t$$

or

$$k_{t+1}^i = \tilde{\lambda}k_t^i + \frac{\rho}{\lambda - \rho}\theta_t \quad (11)$$

where $\lambda \equiv (\beta\tilde{\lambda})^{-1}$.

For future purposes, it is useful to represent the equilibrium $\{k_t^i\}_t$ process recursively as

$$\begin{aligned} k_{t+1}^i &= \tilde{\lambda}k_t^i + \frac{1}{\lambda - \rho}\hat{\theta}_{t+1} \\ \hat{\theta}_{t+1} &= \rho\theta_t \\ \theta_{t+1} &= \rho\theta_t + v_t. \end{aligned} \quad (12)$$

6.1 Filtering

6.1.1 One noisy signal

We get closer to a model that we ultimately want to study by now assuming that firms in market i do not observe θ_t , but instead observe a history w^t of noisy signals at time t .

In particular, assume that

$$\begin{aligned} w_t &= \theta_t + e_t \\ \theta_{t+1} &= \rho\theta_t + v_t \end{aligned} \quad (13)$$

where e_t and v_t are mutually independent i.i.d. Gaussian shock processes with means of zero and variances σ_e^2 and σ_v^2 , respectively.

Define

$$\hat{\theta}_{t+1} = E(\theta_{t+1}|w^t)$$

where $w^t = [w_t, w_{t-1}, \dots, w_0]$ denotes the history of the w_s process up to and including t .

Associated with the state-space representation (13) is the *innovations representation*

$$\begin{aligned}\hat{\theta}_{t+1} &= \rho\hat{\theta}_t + ka_t \\ w_t &= \hat{\theta}_t + a_t\end{aligned}\tag{14}$$

where $a_t \equiv w_t - E(w_t|w^{t-1})$ is the *innovations* process in w_t and the Kalman gain k is

$$k = \frac{\rho p}{p + \sigma_e^2}\tag{15}$$

and where p satisfies the Riccati equation

$$p = \sigma_v^2 + \frac{p\rho^2\sigma_e^2}{\sigma_e^2 + p}.\tag{16}$$

6.1.2 θ -reconstruction error:

Define the state *reconstruction error* $\tilde{\theta}_t$ by

$$\tilde{\theta}_t = \theta_t - \hat{\theta}_t.$$

Then $p = E\tilde{\theta}_t^2$.

Equations (13) and (14) imply

$$\tilde{\theta}_{t+1} = (\rho - k)\tilde{\theta}_t + v_t - ke_t.\tag{17}$$

Now notice that we can express $\hat{\theta}_{t+1}$ as

$$\hat{\theta}_{t+1} = [\rho\theta_t + v_t] + [ke_t - (\rho - k)\tilde{\theta}_t - v_t],\tag{18}$$

where the first term in braces equals θ_{t+1} and the second term in braces equals $-\tilde{\theta}_{t+1}$.

We can express (11) as

$$k_{t+1}^i = \tilde{\lambda}k_t^i + \frac{1}{\lambda - \rho}E\theta_{t+1}|\theta^t.\tag{19}$$

An application of a certainty equivalence principle asserts that when only w^t is observed, the appropriate solution is found by replacing the information set θ^t with w^t in (19).

Making this substitution and using (18) leads to

$$k_{t+1}^i = \tilde{\lambda}k_t^i + \frac{\rho}{\lambda - \rho}\theta_t + \frac{k}{\lambda - \rho}e_t - \frac{\rho - k}{\lambda - \rho}\tilde{\theta}_t.\tag{20}$$

Simplifying equation (18), we also have

$$\hat{\theta}_{t+1} = \rho\theta_t + ke_t - (\rho - k)\tilde{\theta}_t. \quad (21)$$

Equations (20), (21) describe the equilibrium when w^t is observed.

Relative to (11), the equilibrium acquires a new state variable, namely, the θ -reconstruction error, $\tilde{\theta}_t$.

For future purposes, by using (15), it is useful to write (20) as

$$k_{t+1}^i = \tilde{\lambda}k_t^i + \frac{\rho}{\lambda - \rho}\theta_t + \frac{1}{\lambda - \rho} \frac{p\rho}{p + \sigma_e^2} e_t - \frac{1}{\lambda - \rho} \frac{\rho\sigma_e^2}{p + \sigma_e^2} \tilde{\theta}_t \quad (22)$$

In summary, when decision makers in market i observe a noisy signal w_t on θ_t at t , we can represent an equilibrium law of motion for k_t^i as

$$\begin{aligned} k_{t+1}^i &= \tilde{\lambda}k_t^i + \frac{1}{\lambda - \rho} \hat{\theta}_{t+1} \\ \hat{\theta}_{t+1} &= \rho\theta_t + \frac{p\rho}{p + \sigma_e^2} e_t - \frac{\rho\sigma_e^2}{p + \sigma_e^2} \tilde{\theta}_t \\ \tilde{\theta}_{t+1} &= \frac{\rho\sigma_e^2}{p + \sigma_e^2} \tilde{\theta}_t - \frac{p\rho}{p + \sigma_e^2} e_t + v_t \\ \theta_{t+1} &= \rho\theta_t + v_t. \end{aligned} \quad (23)$$

6.2 Two noisy signals

We now construct a **pooling equilibrium** by assuming that a firm in industry i receives a vector w_t of *two* noisy signals on θ_t :

$$\begin{aligned} \theta_{t+1} &= \rho\theta_t + v_t \\ w_t &= \begin{bmatrix} 1 \\ 1 \end{bmatrix} \theta_t + \begin{bmatrix} e_{1t} \\ e_{2t} \end{bmatrix} (0) \end{aligned}$$

To justify that we are constructing is a **pooling equilibrium** we can assume that

$$\begin{bmatrix} e_{1t} \\ e_{2t} \end{bmatrix} = \begin{bmatrix} \epsilon_t^1 \\ \epsilon_t^2 \end{bmatrix}$$

so that a firm in industry i observes the noisy signals on that θ_t presented to firms in both industries i and $-i$.

The appropriate innovations representation becomes

$$\begin{aligned} \hat{\theta}_{t+1} &= \rho\hat{\theta}_t + ka_t \\ w_t &= \begin{bmatrix} 1 \\ 1 \end{bmatrix} \hat{\theta}_t + a_t \end{aligned} \quad (24)$$

where $a_t \equiv w_t - E[w_t|w^{t-1}]$ is a (2×1) vector of innovations in w_t and k is now a (1×2) vector of Kalman gains.

Formulas for the Kalman filter imply that

$$k = \frac{\rho p}{2p + \sigma_e^2} [1 \quad 1] \quad (25)$$

where $p = E\tilde{\theta}_t\tilde{\theta}_t^T$ now satisfies the Riccati equation

$$p = \sigma_v^2 + \frac{p\rho^2\sigma_e^2}{2p + \sigma_e^2}. \quad (26)$$

Thus, when a representative firm in industry i observes *two* noisy signals on θ_t , we can express the equilibrium law of motion for capital recursively as

$$\begin{aligned} k_{t+1}^i &= \tilde{\lambda}k_t^i + \frac{1}{\lambda - \rho}\hat{\theta}_{t+1} \\ \hat{\theta}_{t+1} &= \rho\theta_t + \frac{\rho p}{2p + \sigma_e^2}(e_{1t} + e_{2t}) - \frac{\rho\sigma_e^2}{2p + \sigma_e^2}\tilde{\theta}_t \\ \tilde{\theta}_{t+1} &= \frac{\rho\sigma_e^2}{2p + \sigma_e^2}\tilde{\theta}_t - \frac{p\rho}{2p + \sigma_e^2}(e_{1t} + e_{2t}) + v_t \\ \theta_{t+1} &= \rho\theta_t + v_t. \end{aligned} \quad (27)$$

Below, by using a guess-and-verify tactic, we shall show that outcomes in this **pooling equilibrium** equal those in an equilibrium under the alternative information structure that interested [9]. Section ??

7 Guess-and-verify tactic

As a preliminary step we shall take our recursive representation (23) of an equilibrium in industry i with one noisy signal on θ_t and perform the following steps:

- Compute λ and $\tilde{\lambda}$ by posing a root-finding problem and then solving it using `numpy.roots`
- Compute p by forming the appropriate discrete Riccati equation and then solving it using `quantecon.solve_discrete_riccati`
- Add a *measurement equation* for $P_t^i = bk_t^i + \theta_t + e_t$, $\theta_t + e_t$, and e_t to system (23). Write the resulting system in state-space form and encode it using `quantecon.LinearStateSpace`
- Use methods of the `quantecon.LinearStateSpace` to compute impulse response functions of k_t^i with respect to shocks v_t, e_t .

After analyzing the one-noisy-signal structure in this way, by making appropriate modifications we shall analyze the two-noisy-signal structure.

We proceed to analyze first the one-noisy-signal structure and then the two-noisy-signal structure.

8 Equilibrium with one signal on θ_t

8.1 Step 1: Solve for $\tilde{\lambda}$ and λ

1. Cast $(\lambda - 1) \left(\lambda - \frac{1}{\beta} \right) = b\lambda$ as $p(\lambda) = 0$ where p is a polynomial function of λ .
2. Use `numpy.roots` to solve for the roots of p
3. Verify $\lambda \approx \frac{1}{\beta\tilde{\lambda}}$

Note that $p(\lambda) = \lambda^2 - \left(1 + b + \frac{1}{\beta}\right)\lambda + \frac{1}{\beta}$.

8.2 Step 2: Solve for p

1. Cast $p = \sigma_v^2 + \frac{p\rho^2\sigma_e^2}{2p+\sigma_e^2}$ as a discrete matrix Riccati equation.
2. Use `quantecon.solve_discrete_riccati` to solve for p
3. Verify $p \approx \sigma_v^2 + \frac{p\rho^2\sigma_e^2}{2p+\sigma_e^2}$

Note that:

$$\begin{aligned} A &= \begin{bmatrix} \rho \end{bmatrix} \\ B &= \begin{bmatrix} \sqrt{2} \end{bmatrix} \\ R &= \begin{bmatrix} \sigma_e^2 \end{bmatrix} \\ Q &= \begin{bmatrix} \sigma_v^2 \end{bmatrix} \\ N &= \begin{bmatrix} 0 \end{bmatrix} \end{aligned}$$

8.3 Step 3: Represent the system using `quantecon.LinearStateSpace`

We use the following representation for constructing the `quantecon.LinearStateSpace` instance.

$$\underbrace{\begin{bmatrix} e_{t+1} \\ k_{t+1}^i \\ \tilde{\theta}_{t+1} \\ P_{t+1} \\ \theta_{t+1} \\ v_{t+1} \end{bmatrix}}_{x_{t+1}} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{\kappa}{\lambda-\rho} & \tilde{\lambda} & \frac{-1-\kappa\sigma_e^2}{\lambda-\rho} & 0 & \frac{\rho}{\lambda-\rho} & 0 \\ -\kappa & 0 & \frac{\kappa\sigma_e^2}{p} & 0 & 0 & 1 \\ \frac{b\kappa}{\lambda-\rho} & b\tilde{\lambda} & \frac{-b-\kappa\sigma_e^2}{\lambda-\rho} & 0 & \frac{b\rho}{\lambda-\rho} + \rho & 1 \\ 0 & 0 & 0 & 0 & \rho & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_A \underbrace{\begin{bmatrix} e_t \\ k_t^i \\ \tilde{\theta}_t \\ P_t \\ \theta_t \\ v_t \end{bmatrix}}_{x_t} + \underbrace{\begin{bmatrix} \sigma_e & 0 \\ 0 & 0 \\ 0 & 0 \\ \sigma_e & 0 \\ 0 & 0 \\ 0 & \sigma_v \end{bmatrix}}_C \begin{bmatrix} z_{1,t+1} \\ z_{2,t+1} \end{bmatrix}$$

$$\underbrace{\begin{bmatrix} P_t \\ e_t + \theta_t \\ e_t \end{bmatrix}}_{y_t} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_G \underbrace{\begin{bmatrix} e_t \\ k_t^i \\ \tilde{\theta}_t \\ P_t \\ \theta_t \\ v_t \end{bmatrix}}_{x_t} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}}_H w_{t+1}$$

$$\begin{bmatrix} z_{1,t+1} \\ z_{2,t+1} \\ w_{t+1} \end{bmatrix} \sim \mathcal{N}(0, I)$$

$$= \text{p}_{p+\sigma_e^2}(0)$$

This representation includes extraneous variables such as P_t in the state vector.

We formulate things in this way because it allows us easily to compute covariances of these variables with other components of the state vector (step 5 above) by using the `stationary_distributions` method of the `LinearStateSpace` class.

```
In [2]: import numpy as np
import quantecon as qe
from plotly.subplots import make_subplots
import plotly.graph_objects as go
import plotly.express as px
import plotly.offline as pyo
from statsmodels.regression.linear_model import OLS
from IPython.display import display, Latex, Image

pyo.init_notebook_mode(connected=True)

/home/ubuntu/anaconda3/lib/python3.7/site-packages/numba/np/ufunc/parallel.py:
↳355:
NumbaWarning: The TBB threading layer requires TBB version 2019.5 or later i.e.,
TBB_INTERFACE_VERSION >= 11005. Found TBB_INTERFACE_VERSION = 11004. The TBB
↳threading
layer is disabled.
warnings.warn(problem)
```

```
In [3]: β = 0.9 # Discount factor
ρ = 0.8 # Persistence parameter for the hidden state
b = 1.5 # Demand curve parameter
σ_v = 0.5 # Standard deviation of shock to ϑ_t
σ_e = 0.6 # Standard deviation of shocks to w_t
```

```
In [4]: # Compute λ
poly = np.array([1, -(1 + β + b) / β, 1 / β])
roots_poly = np.roots(poly)
λ_tilde = roots_poly.min()
λ = roots_poly.max()
```

```
In [5]: # Verify that λ = (βλ_tilde) ^ (-1)
tol = 1e-12
np.max(np.abs(λ - 1 / (β * λ_tilde))) < tol
```

Out[5]: True

```
In [6]: A_ricc = np.array([[ρ]])
B_ricc = np.array([[1.]])
R_ricc = np.array([[σ_e ** 2]])
```

```

Q_ricc = np.array([[σ_v ** 2]])
N_ricc = np.zeros((1, 1))
p = qe.solve_discrete_riccati(A_ricc, B_ricc, Q_ricc, R_ricc, N_ricc).item()

p_one = p # Save for comparison later

```

```

In [7]: # Verify that  $p = \sigma_v^2 + p * \rho^2 - (\rho * p)^2 / (p + \sigma_e^2)$ 
tol = 1e-12
np.abs(p - (σ_v ** 2 + p * ρ ** 2 - (ρ * p) ** 2 / (p + σ_e ** 2))) < tol

```

Out[7]: True

```

In [8]: x = ρ * p / (p + σ_e ** 2)
x_prod = x * σ_e ** 2 / p

x_one = x # Save for comparison later

A_lss = np.array([[0., 0., 0., 0., 0., 0.],
                  [x / (λ - ρ), λ_tilde, -x_prod / (λ - ρ), 0., ρ / (λ - ρ), 0.],
                  [-x, 0., x_prod, 0., 0., 1.],
                  [b * x / (λ - ρ), b * λ_tilde, -b * x_prod / (λ - ρ), 0., b
↪ * ρ / (λ -
                  ρ) + ρ, 1.],
                  [0., 0., 0., 0., ρ, 1.],
                  [0., 0., 0., 0., 0., 0.]])

C_lss = np.array([[σ_e, 0.],
                  [0., 0.],
                  [0., 0.],
                  [σ_e, 0.],
                  [0., 0.],
                  [0., σ_v]])

G_lss = np.array([[0., 0., 0., 1., 0., 0.],
                  [1., 0., 0., 0., 1., 0.],
                  [1., 0., 0., 0., 0., 0.]])

```

```

In [9]: mu_0 = np.array([0., 0., 0., 0., 0., 0.])

lss = qe.LinearStateSpace(A_lss, C_lss, G_lss, mu_0=mu_0)

```

```

In [10]: ts_length = 100_000
x, y = lss.simulate(ts_length, random_state=1)

```

```

In [11]: # Verify that two ways of computing  $P_t$  match
np.max(np.abs(np.array([[1., b, 0., 0., 1., 0.]]) @ x - x[3])) < 1e-12

```

Out[11]: True

8.4 Step 4: Compute impulse response functions

To compute impulse response functions of k_t^i , we use the `impulse_response` method of the `quantecon.LinearStateSpace` class and plot the result.

```
In [12]: xcoef, ycoef = lss.impulse_response(j=21)
         data = np.array([xcoef])[0, :, 1, :]

         fig = go.Figure(data=go.Scatter(y=data[:-1, 0], name=r'$e_{t+1}$'))
         fig.add_trace(go.Scatter(y=data[1:, 1], name=r'$v_{t+1}$'))
         fig.update_layout(title=r'Impulse Response Function',
                           xaxis_title='Time',
                           yaxis_title=r'$k^{i}_{t}$')

         fig1 = fig
         # Export to PNG file
         Image(fig1.to_image(format="png"))
         # fig1.show() will provide interactive plot when running
         # notebook locally
```

Out[12]:

8.5 Step 5: Compute stationary covariance matrices and population regressions

We compute stationary covariance matrices by calling the `stationary_distributions` method of the `quantecon.LinearStateSpace` class.

By appropriately decomposing the covariance matrix of the state vector, we obtain ingredients of some population regression coefficients.

$$\Sigma_x = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}$$

where Σ_{11} is the covariance matrix of dependent variables and Σ_{22} is the covariance matrix of independent variables.

Regression coefficients are $\beta = \Sigma_{21}\Sigma_{22}^{-1}$.

To verify an instance of a law of large numbers computation, we construct a long simulation of the state vector and for the resulting sample compute the ordinary least-squares estimator of β that we shall compare to the corresponding population regression coefficients.

```
In [13]: _, _, Sigma_x, Sigma_y, Sigma_yx = lss.stationary_distributions()

         Sigma_11 = Sigma_x[0, 0]
         Sigma_12 = Sigma_x[0, 1:4]
         Sigma_21 = Sigma_x[1:4, 0]
         Sigma_22 = Sigma_x[1:4, 1:4]

         reg_coeffs = Sigma_12 @ np.linalg.inv(Sigma_22)

         print('Regression coefficients (e_t on k_t, P_t, \tilde{\theta}_t)')
         print('-----')
         print(r'k_t:', reg_coeffs[0])
         print(r'\tilde{\theta}_t:', reg_coeffs[1])
         print(r'P_t:', reg_coeffs[2])
```

```
         Regression coefficients (e_t on k_t, P_t, \tilde{\theta}_t)
         -----
         k_t: -3.2755568452197705
```

```
\tilde{\theta}_t: -0.964946117047546
P_t: 0.9649461170475461
```

```
In [14]: # Compute R squared
R_squared = reg_coeffs @ \Sigma_x[1:4, 1:4] @ reg_coeffs / \Sigma_x[0, 0]
R_squared
```

```
Out[14]: 0.9649461170475461
```

```
In [15]: # Verify that the computed coefficients are close to least squares
estimates
model = OLS(x[0], x[1:4].T)
reg_res = model.fit()
np.max(np.abs(reg_coeffs - reg_res.params)) < 1e-2
```

```
Out[15]: True
```

```
In [16]: # Verify that R_squared matches least squares estimate
np.abs(reg_res.rsquared - R_squared) < 1e-2
```

```
Out[16]: True
```

```
In [17]: # Verify that \theta_t + e_t can be recovered
model = OLS(y[1], x[1:4].T)
reg_res = model.fit()
np.abs(reg_res.rsquared - 1.) < 1e-6
```

```
Out[17]: True
```

9 Equilibrium with two noisy signals on θ_t

Steps 1, 4, and 5 are identical to those for the one-noisy-signal structure.

Step 2 requires only a straightforward modification.

For step 3, we use construct the following state-space representation so that we can get our hands on all of the random processes that we require in order to compute a regression of the noisy signal about θ from the other industry that a firm receives directly in a pooling equilibrium on the information that a firm receives in Townsend's original model.

For this purpose, we include equilibrium goods prices from both industries appear in the state vector:

$$\underbrace{\begin{bmatrix} e_{1,t+1} \\ e_{2,t+1} \\ k_{t+1}^i \\ \tilde{\theta}_{t+1} \\ P_{t+1}^1 \\ P_{t+1}^2 \\ \theta_{t+1} \\ v_{t+1} \end{bmatrix}}_{x_{t+1}} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{\kappa}{\lambda-\rho} & \frac{\kappa}{\lambda-\rho} & \tilde{\lambda} & \frac{-1}{\lambda-\rho} \frac{\kappa\sigma_e^2}{p} & 0 & 0 & \frac{\rho}{\lambda-\rho} & 0 \\ -\kappa & -\kappa & 0 & \frac{\kappa\sigma_e^2}{p} & 0 & 0 & 0 & 1 \\ \frac{b\kappa}{\lambda-\rho} & \frac{b\kappa}{\lambda-\rho} & b\tilde{\lambda} & \frac{-b}{\lambda-\rho} \frac{\kappa\sigma_e^2}{p} & 0 & 0 & \frac{b\rho}{\lambda-\rho} + \rho & 1 \\ \frac{b\kappa}{\lambda-\rho} & \frac{b\kappa}{\lambda-\rho} & b\tilde{\lambda} & \frac{-b}{\lambda-\rho} \frac{\kappa\sigma_e^2}{p} & 0 & 0 & \frac{b\rho}{\lambda-\rho} + \rho & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & \rho & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_A \underbrace{\begin{bmatrix} e_{1,t} \\ e_{2,t} \\ k_t^i \\ \tilde{\theta}_t \\ P_t^1 \\ P_t^2 \\ \theta_t \\ v_t \end{bmatrix}}_{x_t} + \underbrace{\begin{bmatrix} \sigma_e & 0 & 0 \\ 0 & \sigma_e & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \sigma_e & 0 & 0 \\ 0 & \sigma_e & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \sigma_v \end{bmatrix}}_C \begin{bmatrix} z_{1,t+1} \\ z_{2,t+1} \\ z_{3,t+1} \end{bmatrix}$$

$$\underbrace{\begin{bmatrix} P_t^1 \\ P_t^2 \\ e_{1,t} + \theta_t \\ e_{2,t} + \theta_t \\ e_{1,t} \\ e_{2,t} \end{bmatrix}}_{y_t} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_G \underbrace{\begin{bmatrix} e_{1,t} \\ e_{2,t} \\ k_t^i \\ \tilde{\theta}_t \\ P_t^1 \\ P_t^2 \\ \theta_t \\ v_t \end{bmatrix}}_{x_t} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_H w_{t+1}$$

$$\begin{bmatrix} z_{1,t+1} \\ z_{2,t+1} \\ z_{3,t+1} \\ w_{t+1} \end{bmatrix} \sim \mathcal{N}(0, I)$$

$$= \mathcal{P}_{2p+\sigma_e^2}(0)$$

```
In [18]: A_ricc = np.array([[rho]])
B_ricc = np.array([[np.sqrt(2)]])
R_ricc = np.array([[sigma_e ** 2]])
Q_ricc = np.array([[sigma_v ** 2]])
N_ricc = np.zeros((1, 1))
p = qe.solve_discrete_riccatti(A_ricc, B_ricc, Q_ricc, R_ricc, N_ricc).
↪item()
```

```
p_two = p # Save for comparison later
```

```
In [19]: # Verify that p = sigma_v^2 + (p_e^2 sigma_e^2) / (2p + sigma_e^2)
tol = 1e-12
np.abs(p - (sigma_v ** 2 + p * rho ** 2 * sigma_e ** 2 / (2 * p + sigma_e ** 2))) < tol
```

```
Out[19]: True
```

```
In [20]: x = rho * p / (2 * p + sigma_e ** 2)
x_prod = x * sigma_e ** 2 / p

x_two = x # Save for comparison later

A_lss = np.array([[0., 0., 0., 0., 0., 0., 0., 0.],
                  [0., 0., 0., 0., 0., 0., 0., 0.],
                  [x / (lambda - rho), x / (lambda - rho), lambda_tilde, -x_prod / (lambda - rho), 0.,
↪0., rho / (lambda -
rho), 0.],
                  [-x, -x, 0., x_prod, 0., 0., 0., 1.],
                  [b * x / (lambda - rho), b * x / (lambda - rho), b * lambda_tilde, -b * x_prod
↪/ (lambda - rho),
0., 0., b * rho / (lambda - rho) + rho, 1.],
                  [b * x / (lambda - rho), b * x / (lambda - rho), b * lambda_tilde, -b * x_prod
↪/ (lambda - rho),
0., 0., b * rho / (lambda - rho) + rho, 1.],
                  [0., 0., 0., 0., 0., 0., rho, 1.],
                  [0., 0., 0., 0., 0., 0., 0., 0.]])

C_lss = np.array([[sigma_e, 0., 0.],
                  [0., sigma_e, 0.]])
```



```

[0., 0., 0.],
[0., 0., 0.],
[σ_e, 0., 0.],
[0., σ_e, 0.],
[0., 0., 0.],
[0., 0., σ_v]])

```

```

G_lss = np.array([[0., 0., 0., 0., 1., 0., 0., 0.],
                 [0., 0, 0, 0., 0., 1., 0., 0.],
                 [1., 0., 0., 0., 0., 0., 1., 0.],
                 [0., 1., 0., 0., 0., 0., 0., 1.],
                 [1., 0., 0., 0., 0., 0., 0., 0.],
                 [0., 1., 0., 0., 0., 0., 0., 0.]])

```

```
In [21]: mu_0 = np.array([0., 0., 0., 0., 0., 0., 0., 0.])
```

```
lss = qe.LinearStateSpace(A_lss, C_lss, G_lss, mu_0=mu_0)
```

```
In [22]: ts_length = 100_000
```

```
x, y = lss.simulate(ts_length, random_state=1)
```

```
In [23]: xcoef, ycoef = lss.impulse_response(j=20)
```

```
In [24]: data = np.array([xcoef])[0, :, 2, :]
```

```

fig = go.Figure(data=go.Scatter(y=data[:-1, 0], name=r'$e_{1,t+1}$'))
fig.add_trace(go.Scatter(y=data[:-1, 1], name=r'$e_{2,t+1}$'))
fig.add_trace(go.Scatter(y=data[1:, 2], name=r'$v_{t+1}$'))
fig.update_layout(title=r'Impulse Response Function',
                  xaxis_title='Time',
                  yaxis_title=r'$k^{i}_{t}$')

fig2=fig
# Export to PNG file
Image(fig2.to_image(format="png"))
# fig2.show() will provide interactive plot when running
# notebook locally

```

Out[24]:

```
In [25]: _, _, Σ_x, Σ_y, Σ_yx = lss.stationary_distributions()
```

```

Σ_11 = Σ_x[1, 1]
Σ_12 = Σ_x[1, 2:5]
Σ_21 = Σ_x[2:5, 1]
Σ_22 = Σ_x[2:5, 2:5]

```

```
reg_coeffs = Σ_12 @ np.linalg.inv(Σ_22)
```

```

print('Regression coefficients (e_{2,t} on k_t, P^{1}_t,
↪ \tilde{\theta}_t)')
print('-----')
print(r'k_t:', reg_coeffs[0])
print(r'\tilde{\theta}_t:', reg_coeffs[1])
print(r'P_t:', reg_coeffs[2])

```

```

Regression coefficients (e_{2,t} on k_t, P^{1}_t, \tilde{\theta}_t)
-----
k_t: 0.0
\tilde{\theta}_t: 0.0
P_t: 0.0

```

```

In [26]: # Compute R squared
R_squared = reg_coeffs @ \Sigma_x[2:5, 2:5] @ reg_coeffs / \Sigma_x[1, 1]
R_squared

```

Out[26]: 0.0

```

In [27]: # Verify that the computed coefficients are close to least squares
↳ estimates
model = OLS(x[1], x[2:5].T)
reg_res = model.fit()
np.max(np.abs(reg_coeffs - reg_res.params)) < 1e-2

```

Out[27]: True

```

In [28]: # Verify that R_squared matches least squares estimate
np.abs(reg_res.rsquared - R_squared) < 1e-2

```

Out[28]: True

```

In [29]: _, _, \Sigma_x, \Sigma_y, \Sigma_yx = lss.stationary_distributions()

\Sigma_11 = \Sigma_x[1, 1]
\Sigma_12 = \Sigma_x[1, 2:6]
\Sigma_21 = \Sigma_x[2:6, 1]
\Sigma_22 = \Sigma_x[2:6, 2:6]

```

```

reg_coeffs = \Sigma_12 @ np.linalg.inv(\Sigma_22)

```

```

print('Regression coefficients (e_{2,t} on k_t, P^{1}_t, P^{2}_t,
↳ \tilde{\theta}_t)')
print('-----')
print(r'k_t:', reg_coeffs[0])
print(r'\tilde{\theta}_t:', reg_coeffs[1])
print(r'P^{1}_t:', reg_coeffs[2])
print(r'P^{2}_t:', reg_coeffs[3])

```

```

Regression coefficients (e_{2,t} on k_t, P^{1}_t, P^{2}_t, \tilde{\theta}_t)
-----
k_t: -3.1373589171035654
\tilde{\theta}_t: -0.924234396744368
P^{1}_t: -0.037882801627815835
P^{2}_t: 0.9621171983721839

```

```

In [30]: # Compute R squared
R_squared = reg_coeffs @ \Sigma_x[2:6, 2:6] @ reg_coeffs / \Sigma_x[1, 1]
R_squared

```

Out[30]: 0.9621171983721838

9.1 Key step

Now we come to the key step of verifying that equilibrium outcomes for prices and quantities are identical in the pooling equilibrium and Townsend's original model.

We accomplish this by compute a population linear least squares regression of the noisy signal that firms in the other industry receive in a pooling equilibrium on time t information that a firm receives in Townsend's original model.

Let's compute the regression and stare at the R^2 :

```
In [31]: # Verify that  $\theta_t + e^2_t$  can be recovered
        #  $\theta_t + e^2_t$  on  $k^i_t, P^1_t, P^2_t, \tilde{\theta}_t$ 

        model = OLS(y[1], x[2:6].T)
        reg_res = model.fit()
        np.abs(reg_res.rsquared - 1.) < 1e-6
```

Out[31]: True

```
In [32]: reg_res.rsquared
```

Out[32]: 1.0

The R^2 in this regression equals 1.

That verifies that a firm's information set in Townsend's original model equals its information set in a pooling equilibrium.

Therefore, equilibrium prices and quantities in Townsend's original model equal those in a pooling equilibrium.

10 Comparison of the two signal structures

It is enlightening side by side to plot impulse response functions for capital in an industry for the two information noisy-signal information structures.

Please remember that the two-signal structure corresponds to the **pooling equilibrium** and also **Townsend's original model**.

```
In [33]: fig_comb = go.Figure(data=[*fig1.data,
                                   *fig2.update_traces(xaxis='x2',
                                                       yaxis='y2').data]).set_subplots(1, 2,
                                   subplot_titles=("One noisy-signal structure", "Two noisy-signal
↵structure"),
                                   horizontal_spacing=0.1,
                                   shared_yaxes=True)
        # Export to PNG file
        Image(fig_comb.to_image(format="png"))
        # fig_comb.show() will provide interactive plot when running
        # notebook locally
```

Out [33]:

The graphs above show that

- the response of k_t^i to shocks v_t to the hidden Markov demand state θ_t process is **larger** in the two-noisy=signal structure
- the response of k_t^i to idiosyncratic *own-market* noise-shocks e_t is **smaller** in the two-noisy-signal structure

Taken together, these findings in turn can be shown to imply that time series correlations and coherences between outputs in the two industries are higher in the two-noisy-signals or **pooling** model.

The enhanced influence of the shocks v_t to the hidden Markov demand state θ_t process that emerges from the two-noisy-signal model relative to the one-noisy-signal model is a symptom of a lower equilibrium hidden-state reconstruction error variance in the two-signal model:

```
In [34]: display(Latex('$\\textbf{Reconstruction error variances}$'))
         display(Latex(f'One-noise structure: ${round(p_one, 6)}$'))
         display(Latex(f'Two-noise structure: ${round(p_two, 6)}$'))
```

Reconstruction error variances

One-noise structure: 0.36618

Two-noise structure: 0.324062

Kalman gains for the two structures are

```
In [35]: display(Latex('$\\textbf{Kalman Gains}$'))
         display(Latex(f'One noisy-signal structure: ${round(x_one, 6)}$'))
         display(Latex(f'Two noisy-signals structure: ${round(x_two, 6)}$'))
```

Kalman Gains

One noisy-signal structure: 0.403404

Two noisy-signals structure: 0.25716

11 Notes on History of the Problem

To truncate what he saw as an intractable, infinite dimensional state space, Townsend constructed an approximating model in which the common hidden Markov demand shock is revealed to all firms after a fixed number of periods.

Thus,

- Townsend wanted to assume that at time t firms in industry i observe $k_t^i, Y_t^i, P_t^i, (P^{-i})^t$, where $(P^{-i})^t$ is the history of prices in the other market up to time t .
- Because that turned out to be too challenging, Townsend made an alternative assumption that eased his calculations: that after a large number S of periods, firms in industry i observe the hidden Markov component of the demand shock θ_{t-S} .

Townsend argued that the more manageable model could do a good job of approximating the intractable model in which the Markov component of the demand shock remains unobserved for ever.

By applying technical machinery of [4], [?] showed that there is a recursive representation of the equilibrium of the perpetually and symmetrically uninformed model formulated but not completely solved in section 8 of [9].

A reader of [?] will notice that their representation of the equilibrium of Townsend's model exactly matches that of the **pooling equilibrium** presented here.

We have structured our notation in this lecture to facilitate comparison of the **pooling equilibrium** constructed here with the equilibrium of Townsend's model reported in [?].

The computational method of [?] is recursive: it enlists the Kalman filter and invariant subspace methods for solving systems of Euler equations Section ?? .

As [8], [3], and [7] also found, the equilibrium is fully revealing: observed prices tell participants in industry i all of the information held by participants in market $-i$ ($-i$ means not i).

This means that higher-order beliefs play no role: seeing equilibrium prices in effect lets decision makers pool their information sets Section ?? .

The disappearance of higher order beliefs means that decision makers in this model do not really face a problem of forecasting the forecasts of others.

They know those forecasts because they are the same as their own.

11.1 Further historical remarks

[7] proposed a way to compute an equilibrium without making Townsend's approximation.

Extending the reasoning of [muth1960], Sargent noticed that it is possible to summarize the relevant history with a low dimensional object, namely, a small number of current and lagged forecasting errors.

Positing an equilibrium in a space of perceived laws of motion for endogenous variables that takes the form of a vector autoregressive, moving average, Sargent described an equilibrium as a fixed point of a mapping from the perceived law of motion to the actual law of motion of that form.

Sargent worked in the time domain and had to guess and verify the appropriate orders of the autoregressive and moving average pieces of the equilibrium representation.

By working in the frequency domain [3] showed how to discover the appropriate orders of the autoregressive and moving average parts, and also how to compute an equilibrium.

The [?] recursive computational method, which stays in the time domain, also discovered appropriate orders of the autoregressive and moving average pieces.

In addition, by displaying equilibrium representations in the form of [4], [?] showed how the moving average piece is linked to the innovation process of the hidden persistent component of the demand shock.

That scalar innovation process is the additional state variable contributed by the problem of extracting a signal from equilibrium prices that decision makers face in Townsend's model.

Footnotes

[1] [?] verified this assertion using a different tactic, namely, by constructing analytic formulas an equilibrium under the incomplete information structure and confirming that they match the pooling equilibrium formulas derived here.

[2] See [2] for an account of invariant subspace methods.

[3] See [2] for a discussion of the information assumptions needed to create a situation in which higher order beliefs appear in equilibrium decision rules. The way to read our findings in light of [1] is that Townsend's section 8 model has too few sources of random shocks relative to sources of signals to permit higher order beliefs to play a role.

[4] See [6], especially chapters IX and XIV, for the principles that guide solving some roots backwards and others forwards.

[5] As noted [6], this difference equation is the Euler equation for the planning problem of maximizing the discounted sum of consumer plus producer surplus.

[6] [?] verify the same claim by applying machinery of [4].

References

- [1] Franklin Allen, Stephen Morris, and Hyun Song Shin. Beauty contests, bubbles, and iterated expectations in asset markets. *mimeo*, 2002.
- [2] Evan Anderson, Lars Peter Hansen, Ellen R. McGrattan, and Thomas J. Sargent. Mechanics of forming and estimating dynamic linear economies. In Hans M. Amman, David A. Kendrick, and John Rust, editors, *Handbook of computational economics*, pages 171–252. Elsevier Science, North-Holland, 1996.
- [3] Kenneth Kasa. Forecasting the forecasts of others in the frequency domain. *Review of Economic Dynamics*, 3:726–756, 2000.
- [4] Joseph Pearlman, David Currie, and Paul Levine. Rational Expectations Models with Private Information. *Economic Modelling*, 3(2):90–105, 1986.
- [5] Jr. Robert E. Lucas. An equilibrium model of the business cycle. *Journal of Political Economy*, 83:1113–1144, 1975.
- [6] Thomas J Sargent. *Macroeconomic Theory*. Academic Press, New York, 2nd edition, 1987.
- [7] Thomas J. Sargent. Equilibrium with signal extraction from endogenous variables. *Journal of Economic Dynamics and Control*, 15:245–273, 1991.
- [8] Kenneth J. Singleton. Asset prices in a time-series model with disparately informed competitive traders. In William A. Barnett and Kenneth J. Singleton, editors, *New Approaches to Monetary Economics*. Cambridge University Press, 1987.
- [9] Robert M. Townsend. Forecasting the forecasts of others. *Journal of Political Economy*, 91:546–588, 1983.