The Stata Journal (2018)
**18**, Number 3, pp. 618–662

# Theory and practice of total-factor productivity estimation: The control function approach using Stata

Gabriele Rovigatti
University of Chicago
Booth School of Business
Chicago, IL
gabriele.rovigatti@gmail.com

Vincenzo Mollisi
Free University of Bozen–Bolzano
Faculty of Economics and Management
Bolzano, Italy
vincenzo.mollisi@gmail.com

**Abstract.** Alongside instrumental-variables and fixed-effects approaches, the control function approach is the most widely used in production function estimation. Olley and Pakes (1996, *Econometrica* 64: 1263–1297), Levinsohn and Petrin (2003, *Review of Economic Studies* 70: 317–341), and Ackerberg, Caves, and Frazer (2015, *Econometrica* 83: 2411–2451) have all contributed to the field by proposing two-step estimation procedures, whereas Wooldridge (2009, *Economics Letters* 104: 112–114) showed how to perform a consistent estimation within a single-step generalized method of moments framework. In this article, we propose a new estimator based on Wooldridge's estimation procedure, using dynamic panel instruments à la Blundell and Bond (1998, *Journal of Econometrics* 87: 115–143), and we evaluate its performance by using Monte Carlo simulations. We also present the new command `prodest` for production function estimation, and we show its main features and strengths in a comparative analysis with other community-contributed commands. Finally, we provide evidence of the numerical challenges faced when using the Olley–Pakes and Levinsohn–Petrin estimators with the Ackerberg–Caves–Frazer correction in empirical applications, and we document how the generalized method of moments estimates vary depending on the optimizer or starting points used.

**Keywords:** st0537, prodest, production functions, productivity, MrEst, dynamic panel GMM

## 1 Introduction

The correct estimation of total-factor productivity is a fundamental issue in applied economics and is the main topic of several seminal articles. When subject to productivity shocks, firms respond by expanding their level of output and by demanding more input; negative shocks, on the other hand, lead to a decline in both output and demand for input. The positive correlation between the observable input levels and the unobservable productivity shocks is a source of bias in ordinary least squares (OLS) when estimating total-factor productivity. Various methods have been proposed to tackle this simultaneity issue, and according to their approaches, it is possible to group them into three families: fixed-effects (FE), instrumental-variables (IV), and control function approaches.

 

In the latter group, Olley and Pakes (1996) are the first to propose a two-step procedure aimed at overcoming the endogeneity: they use the investment level to proxy for productivity. Their approach has been refined by Levinsohn and Petrin (2003) and Ackerberg, Caves, and Frazer (2015). Wooldridge (2009) proposes a novel estimation setting, showing how to obtain the Levinsohn–Petrin (LP) estimator within a system generalized method of moments (GMM) econometric framework, which can be estimated in a single step, and showing the appropriate moment conditions.

All the mentioned models rely on a crucial assumption that underlies the dynamic profit-maximization problem faced by the firm at each period $t$: the idiosyncratic shock to productivity at time $t$ (that is, $\xi_t$) does not affect the choice of the level of state variables, which is taken at $t-b$,[1] but only that of free variables. Therefore, $\xi_t$ is uncorrelated to the contemporaneous value of the state and to all the lagged values of the free and state variables, and all of these are valid instruments for parameter identification. However, adding lags to the system reduces sample dimension and decreases available information.

In this article, we propose a modification to the Wooldridge (WRDG) estimator based on a matrix of dynamic panel instruments. Such an approach makes it possible to increase the moment restrictions without losing information, which is a highly desirable feature when dealing with "large $N$, small $T$" panel datasets that are so frequent in the related literature. We show that our estimator, which we will call MrEst, performs better than Wooldridge's on simulated data with a small number of periods, increases the sample size in overidentified models, and produces more stable results.

We then introduce a new command, `prodest`, that implements all the above listed methodologies. We present the command syntax, describe all options, and perform a brief comparison with existing community-contributed commands. Eventually, we focus on Ackerberg–Caves–Frazer (ACF) methodology, and using their data-generating process (DGP), we show how such nonlinear problems' solutions are extremely dependent on the choice of optimization starting points.

The remainder of the article is structured as follows: in section 2, we review all control function approaches, list their weaknesses, and provide a general overview of the state of the art; in section 2.6, we introduce our new estimator, MrEst, with a comprehensive presentation of its key characteristics; in section 3, we introduce `prodest`, its main features, and practical examples of its usage; in section 4, we comparatively describe the command and present evidence on ACF dependence on starting points; and in section 5, we conclude.

## 2 Control function approach

In this section, we provide a brief but complete overview of the most common techniques using a control function approach for production function estimation. For the remainder of the article, consider Cobb and Douglas (1928) technology for firm $i$ at time $t$:

1. Where $b > 0$ can take different values depending on state-variable dynamics.

$$y_{it} = \alpha + \mathbf{w}_{it}\boldsymbol{\beta} + \mathbf{x}_{it}\boldsymbol{\gamma} + \omega_{it} + \varepsilon_{it} \tag{1}$$

where $y_{it}$ is the log gross or the value-added output, $\mathbf{w}_{it}$ is a $1 \times J$ vector of log free-variables, and $\mathbf{x}_{it}$ is a $1 \times K$ vector of log state-variables. The random component $\omega_{it}$ is the unobservable productivity or technical efficiency, and $\varepsilon_{it}$ is an idiosyncratic output shock distributed as white noise. We assume with the Olley–Pakes (OP) and LP methods that productivity evolves according to a first-order Markov process:

$$\omega_{it} = E(\omega_{it}\,|\Omega_{it-1}) + \xi_{it} = E(\omega_{it}\,|\omega_{it-1}) + \xi_{it} = g(\omega_{it-1}) + \xi_{it} \tag{2}$$

where $\Omega_{it-1}$ is the information set at $t-1$ and $\xi_{it}$ is the productivity shock, assumed to be uncorrelated with productivity $\omega_t$ and with state variables $\mathbf{x}_{it}$.

## 2.1 Olley–Pakes method

Olley and Pakes (1996) were the first to propose a consistent two-step estimation procedure for (1). Their key idea is to exploit firm investment levels as a proxy variable for $\omega_{it}$. They prove their estimates of productivity to be consistent under several assumptions on top of those mentioned above:

A.1 $i_{it} = f(\mathbf{x}_{it}, \omega_{it})$ is the investment policy function, invertible in $\omega_{it}$. Moreover, $i_{it}$ is monotonically increasing in $\omega_{it}$.

A.2 The state variables (typically capital) evolve according to the investment policy function $i_{it}$, which is decided at time $t-1$.

A.3 The free variables $\mathbf{w}_{it}$ (typically labor inputs and intermediate materials) are nondynamic, in the sense that their choice at $t$ does not impact future profits, and are chosen at time $t$ after the firm realizes productivity shock.

Hence, given A.1 and A.2, the investment $i_{it}$ is orthogonal to the state variable in $t$ such that $E(i_{it}|\mathbf{x}_{it}) = 0$ and can be inverted, yielding the following proxy for productivity:

$$\omega_{it} = f^{-1}(i_{it}, \mathbf{x}_{it}) = h(i_{it}, \mathbf{x}_{it}) \tag{3}$$

This is an unknown function of observable variables. Plugging (3) into (1), we obtain

$$y_{it} = \alpha + \mathbf{w}_{it}\boldsymbol{\beta} + \mathbf{x}_{it}\boldsymbol{\gamma} + h(i_{it}, \mathbf{x}_{it}) + \varepsilon_{it} = \alpha + \mathbf{w}_{it}\boldsymbol{\beta} + \Phi_{it}(i_{it}, \mathbf{x}_{it}) + \varepsilon_{it} \tag{4}$$

where we define $\Phi_{it}(i_{it}, \mathbf{x}_{it}) = \mathbf{x}_{it}\boldsymbol{\gamma} + h(i_{it}, \mathbf{x}_{it}) = \mathbf{x}_{it}\boldsymbol{\gamma} + \omega_{it}$. Equation (4) is a partially linear model identified only in the free-variable vector $\mathbf{w}_{it}$, and can be nonparametrically estimated approximating $\Phi_{it}(i_{it}, \mathbf{x}_{it})$ by an $n$th-degree polynomial $\widehat{\Phi}$ or by a local linear regression (first stage). This yields a consistent estimate of the free variables' parameters $\widehat{\boldsymbol{\beta}}$. Then, using (2), it becomes possible to estimate $\boldsymbol{\gamma}$ by rewriting the model for $y_{it} - \mathbf{w}_{it}\widehat{\boldsymbol{\beta}}$ conditional on $\mathbf{x}_{it}$:

$$\begin{aligned} y_{it} - \mathbf{w}_{it}\widehat{\boldsymbol{\beta}} &= \alpha_0 + \mathbf{x}_{it}\boldsymbol{\gamma} + \omega_{it} + \varepsilon_{it} \\ &= \alpha_0 + \mathbf{x}_{it}\boldsymbol{\gamma} + E(\omega_{it}|\,\omega_{it-1}) + \xi_{it} + \varepsilon_{it} \\ &= \alpha_0 + \mathbf{x}_{it}\boldsymbol{\gamma} + g(\omega_{it-1}) + e_{it} \end{aligned} \tag{5}$$

where $e_{it} = \xi_{it} + \varepsilon_{it}$. Given $\widehat{\omega}_{it} = \widehat{\Phi}_{it} - \mathbf{x}_{it}\boldsymbol{\gamma}$, (5) becomes

$$y_{it} - \mathbf{w}_{it}\widehat{\boldsymbol{\beta}} = \alpha_0 + \mathbf{x}_{it}\boldsymbol{\gamma} + g(\widehat{\Phi}_{it-1} - \mathbf{x}_{it-1}\boldsymbol{\gamma}) + e_{it} \tag{6}$$

where the function $g(\cdot)$ can be left unspecified and estimated nonparametrically. Alternatively, if we assume $g(\cdot)$ to follow a random walk, we can restate (6) as

$$y_{it} - \mathbf{w}_{it}\widehat{\boldsymbol{\beta}} = \alpha_0 + (\mathbf{x}_{it} - \mathbf{x}_{it-1})\,\boldsymbol{\gamma} + \widehat{\Phi}_{it-1} + e_{it} \tag{7}$$

and

$$e_{it} = y_{it} - \mathbf{w}_{it}\widehat{\boldsymbol{\beta}} - \alpha_0 - \mathbf{x}_{it}\boldsymbol{\gamma}^* - g(\widehat{\Phi}_{it-1} - \mathbf{x}_{it-1}\boldsymbol{\gamma}^*)$$

at the true $\boldsymbol{\gamma}^*$ values.

Equation (7) suggests an immediate approach to the estimation. In fact, residuals $e_{it}$ can be used to build a GMM estimator exploiting the moment conditions $E(e_{it}x_{it}^k) = 0$, $\forall k$ (second stage),[2] where $x^k$ are the single elements of vector $\mathbf{x}$. The $\boldsymbol{\gamma}^*$ vector is the vector of parameters that minimizes the criterion function:

$$\boldsymbol{\gamma}^* = \text{argmax}\left\{ -\sum_k \left( \sum_i \sum_t e_{it} x_{it}^k \right)^2 \right\}$$

In their seminal article, Olley and Pakes (1996) discuss potential selection bias due to the nonrandomness in firms dropping out of the sample. More specifically, less productive firms could be forced out of the market exactly because of their low levels of productivity, thus leaving only the most productive firms in the sample. Olley and Pakes assume that a firm continues to operate provided that its productivity level exceeds the lower bound, that is, $\chi_{it} = 1 \iff \omega_{it} \geq \underline{\omega}_{it}$, where $\chi_{it}$ is a survival binary variable and $\underline{\omega}_{it}$ is an industry-specific exit-triggering threshold (see Hopenhayn [1992] and Melitz [2003]). Hence, they propose a third step in estimation to account for that: (6) is expressed conditionally not only on the state variable but also on $\chi_{it}$—that is, productivity is a function of its past values and the survival indicator variable:

2. Alternatively, estimation of the second stage can be done with (7) using nonlinear least squares because $e_{it}$ is a combination of pure errors.

$$y_{it} - \mathbf{w}_{it}\widehat{\boldsymbol{\beta}} = \alpha_0 + \mathbf{x}_{it}\boldsymbol{\gamma} + E(\omega_{it}|\,\omega_{it-1},\,\chi_{it}) + e_{it}$$

The bias correction proposed by Olley and Pakes (1996) consists of adding to (7) an estimate of the conditional probability of remaining active in the market, that is, $\widehat{\Pr}_{it+1} \equiv \Pr\left(\chi_{it+1} = 1|\mathbf{x}_{it}\right)$. Thus,

$$y_{it} - \mathbf{w}_{it}\widehat{\boldsymbol{\beta}} = \alpha_0 + \mathbf{x}_{it}\boldsymbol{\gamma} + g(\widehat{\Phi}_{it-1} - \mathbf{x}_{it-1}\boldsymbol{\gamma},\,\widehat{\Pr}_{it}) + e_{it}$$

where $\widehat{\Pr}_{it}$ is the fitted surviving probability—typically estimated through a discrete choice model on a polynomial of the state-variable vector $\mathbf{x}_{it}$ and the investment.

## 2.2 Levinsohn–Petrin method

The OP approach has a major drawback in empirical applications that limits its range of applications: real firm-level data have many zeros in investment, preventing estimation in practice. This is because of common industrial practices that violate the monotonicity assumption A.1: investments are not decided at each point in time but are postponed for a few years before being made all at once. Levinsohn and Petrin (2003) propose to overcome this issue by exploiting intermediate input levels as a proxy variable for $\omega_{it}$. As in the OP approach, the LP methodology is based on assumptions:

B.1 Firms observe their productivity shock and adjust their optimal level of intermediate inputs (materials) according to the demand function $m(\omega_{it},\,\mathbf{x}_{it})$.

B.2 $m_{it} = f(\mathbf{x}_{it},\,\omega_{it})$ is the intermediate input function, invertible in $\omega_{it}$. Moreover, $m_{it}$ is monotonically increasing in $\omega_{it}$.

B.3 The state variables (typically capital) evolve according to the investment policy function $i()$, which is decided at time $t-1$.

B.4 The free variables $\mathbf{w}_{it}$ (typically labor inputs and intermediate materials) are nondynamic, in the sense that their choice at $t$ does not impact future profits, and are chosen at time $t$ after the firm realizes productivity shock.

Under the set of assumptions B.1–B.4, intermediate input demand is orthogonal to the set of state variables in $t$ such that $E(m_{it}|\mathbf{x}_{it}) = 0$ and $m_{it}$ can be inverted, yielding the following technical efficiency proxy:

$$\omega_{it} = h(m_{it}, \mathbf{x}_{it}) \tag{8}$$

which is an unknown function of observable variables. Plugging (8) into (1) and distinguishing the intermediate input variable from the free variables, we obtain

$$\begin{aligned} y_{it} &= \alpha + \mathbf{w}_{it}\boldsymbol{\beta} + \mathbf{x}_{it}\boldsymbol{\gamma} + \delta m_{it} + h(m_{it}, \mathbf{x}_{it}) + e_{it} \\ &= \alpha + \mathbf{w}_{it}\boldsymbol{\beta} + \Phi_{it}(m_{it}, \mathbf{x}_{it}) + e_{it} \end{aligned} \tag{9}$$

where $e_{it} = \xi_{it} + \varepsilon_{it}$.

Equation (9) is a partially linear model identified only in the free variable vector but not in the proxy variable, $m_{it}$. Similar to OP, (9) can be nonparametrically estimated, approximating $\Phi_{it}(m_{it}, \mathbf{x}_{it})$ by an $n$th-degree polynomial or by local linear regression (first stage). At the true values [$\boldsymbol{\gamma}^*$, $\delta$*], we can define the residual function $e_{it}$ as the following:

$$e_{it} = y_{it} - \mathbf{w}_{it}\widehat{\boldsymbol{\beta}} - \mathbf{x}_{it}\boldsymbol{\gamma}^* - \delta^* m_{it} - g\left(\widehat{\Phi}_{it-1} - \mathbf{x}_{it-1}\boldsymbol{\gamma}^* - \delta^* m_{it}\right)$$

However, $e_{it}$ is no longer a combination of pure errors. The intermediate input variable is correlated with the error term given the firms' response to the technology efficiency shock $\xi_{it}$. Thus, nonlinear least squares would provide inconsistent estimates, and relying on a GMM estimator is mandatory. The GMM estimator might be constructed by exploiting the residuals $e_{it}$ and the set of moment conditions $E(e_{it} z_{it}^k) = 0, \forall k$, where $k$ is the index of the instrument vector $\mathbf{z} = [\mathbf{x}_{it}, m_{it-1}]$,

$$\begin{bmatrix} \boldsymbol{\gamma}^* \\ \delta^* \end{bmatrix} = \text{argmax} \left\{ -\sum_k \left( \sum_i \sum_t e_{it} z_{it}^k \right)^2 \right\}$$

consistently estimates the set of parameters [$\boldsymbol{\gamma}$, $\delta$]$^\intercal$.

## 2.3 Ackerberg–Caves–Frazer correction

Both OP and LP assume that firms are able to instantly adjust some inputs at no cost when subject to productivity shocks. However, Ackerberg, Caves, and Frazer (2015) and Bond and Söderbom (2005) remark that the labor coefficient can be consistently estimated in the first stage only if the free variables show variability independently of the proxy variable. If this is not the case, their coefficients would be perfectly collinear in the first-stage estimation and hence not identifiable. In particular, in the LP setting, labor and intermediate inputs are assumed to be allocated simultaneously at time $t$. This implies that labor and materials are both chosen as a function of productivity and state variables $\mathbf{x}_{it}$:

$$\begin{aligned} m_{it} &= m(\omega_{it}, \mathbf{x}_{it}) \\ l_{it} &= l(\omega_{it}, \mathbf{x}_{it}) \end{aligned}$$

Using the monotonicity condition (B.2), the ACF approach provides the following results:

$$l_{it} = l\left\{h(m_{it}, \mathbf{x}_{it}),\, \mathbf{x}_{it}\right\}$$

Hence, a collinearity issue arises in estimating the first stage, where the labor appears both as a free variable and in the nonparametric polynomial approximation $\widehat{\Phi}_{it}$. In the same fashion, the collinearity issue affects the OP estimator. Ackerberg, Caves, and Frazer (2015) propose an alternative approach based on the following assumptions:

C.1 $p_{it} = f(\mathbf{x}_{it},\, l_{it},\, \omega_{it})$ is the proxy variable policy function, invertible in $\omega_{it}$. Moreover, $p_{it}$ is monotonically increasing in $\omega_{it}$.

C.2 The state variables are decided at time $t - b$.

C.3 The labor input, $l_{it}$, is chosen at time $t - \zeta$, where $0 < \zeta < b$. The free variables, $\mathbf{w}_{it}$, are chosen at time $t$ when the firm productivity shock is realized.

C.4 The production function is value added in the sense that the intermediate input $m_{it}$ does not enter the production function to be estimated.

Assumption C.4 is needed because Bond and Söderbom (2005) have shown that, under the scalar unobservable assumptions of ACF, a gross output production function is not identified without imposing further restrictions of the model; see paragraph 4.1 of Ackerberg, Caves, and Frazer (2015) for the details. Under the set of assumptions C.1–C.3, the first-stage estimation is meant to remove the shock $\varepsilon_{it}$ from the output $y_{it}$. In particular, the policy function can be inverted and plugged into (1), yielding

$$y_{it} = \Phi_{it}(p_{it}, \mathbf{x}_{it},\, \mathbf{w}_{it},\, l_{it}) + \varepsilon_{it}$$

where $\Phi_{it}(p_{it}, \mathbf{x}_{it},\, \mathbf{w}_{it},\, l_{it}) = \mathbf{x}_{it}\boldsymbol{\gamma} + \mathbf{w}_{it}\boldsymbol{\beta} + \mu l_{it} + h(p_{it}, \mathbf{x}_{it},\, \mathbf{w}_{it},\, l_{it})$. Once $\widehat{\Phi}_{it}$ is recovered, for any candidate vector $[\boldsymbol{\gamma}^*,\, \boldsymbol{\beta}^*,\, \mu^*]$, it is possible to obtain the residuals

$$\widehat{\omega}_{it} = \widehat{\Phi}_{it} - \mathbf{x}_{it}\boldsymbol{\gamma} - \mathbf{w}_{it}\boldsymbol{\beta} - \mu l_{it}$$

and, exploiting the Markov chain assumption $\omega_{it} = E(\omega_{it}\,|\omega_{it-1}) + \xi_{it} = g(\omega_{it-1}) + \xi_{it}$, obtain the residuals $\xi_{it}$. These, combined with the set of moment conditions $E(\xi_{it} z_{it}^k) = 0$, $\forall k$, where $k$ is the index of the instrument vector $\mathbf{z} = [\mathbf{x}_{it},\, m_{it-1},\, l_{it-1}]$, lead to the GMM criterion function (second stage):

$$\begin{bmatrix} \boldsymbol{\gamma}^* \\ \boldsymbol{\beta}^* \\ \delta^* \end{bmatrix} = \text{argmax}\left\{ -\sum_k \left( \sum_i \sum_t \xi_{it} z_{it}^k \right)^2 \right\}$$

## 2.4 Wooldridge estimation

Wooldridge (2009) proposes to address the OP and LP problems by replacing the two-step estimation procedure with a GMM setup as in Wooldridge (1996). In particular, he shows how to write the relevant moment restrictions in terms of two equations that have the same dependent variable ($y_{it}$) but are characterized by a different set of instruments. This approach has useful features with respect to previously proposed estimation routines:

- It overcomes the potential identification issue highlighted by Ackerberg, Caves, and Frazer (2015) in the first stage.
- Robust standard errors are easily obtained, accounting for both serial correlation and heteroskedasticity.[3]

In the first stages of both OP and LP, the estimation of the parameters is addressed under the assumption that

$$E(\varepsilon_{it}|\omega_{it-1}, \mathbf{w}_{it}, \mathbf{x}_{it}, m_{it}, \mathbf{w}_{it-1}, \mathbf{x}_{it-1}, m_{it-1}, \ldots, \mathbf{w}_{i1}, \mathbf{x}_{i1}, m_{i1}) = 0 \tag{10}$$

without imposing any functional form on the control function $\omega_{it} = h(.,.)$. The second-stage assumption exploits the Markovian nature of productivity and the assumed orthogonality between productivity shocks and current values of the state variables, as well as between productivity shocks and past realizations of the free variables and the intermediate inputs. Following LP and rewriting (2), it states

$$\begin{aligned} E(\omega_{it}|\mathbf{x}_{it}, \mathbf{w}_{it-1}, \mathbf{x}_{it-1}, m_{it-1}, \ldots, \mathbf{w}_{i1}, \mathbf{x}_{i1}, m_{i1}) &= E(\omega_{it}\,|\omega_{it-1}) \\ &= f\{h(\mathbf{x}_{it-1}, m_{it-1})\} \end{aligned} \tag{11}$$

where, as for $h(.,.)$, no functional form is imposed on $f(\cdot)$. Assumptions (10) and (11) directly lead to the formulation of the following two equations:

$$y_{it} = \alpha + \mathbf{w}_{it}\boldsymbol{\beta} + \mathbf{x}_{it}\boldsymbol{\gamma} + h(\mathbf{x}_{it}, m_{it}) + v_{it} \tag{12}$$

$$y_{it} = \alpha + \mathbf{w}_{it}\boldsymbol{\beta} + \mathbf{x}_{it}\boldsymbol{\gamma} + f\{h(\mathbf{x}_{it-1}, m_{it-1})\} + \eta_{it} \tag{13}$$

where $\eta_{it} = \xi_{it} + v_{it}$.

In the estimation, the approach is to deal with the unknown functional forms using $n$th-degree polynomials in $\mathbf{x}_{it}$ and $m_{it}$,[4] where the limiting case with $\mathbf{x}_{it}$ and $m_{it}$ (that is, $n = 1$) entering linearly should always be allowed. In particular, if we assume that

3. Levinsohn and Petrin (2003) and Olley and Pakes (1996) recommend instead to bootstrap the standard errors of their estimators, as is usual in two-step estimation procedures.
4. Levinsohn and Petrin (2003) suggest to use third-degree polynomials. However, the higher the degree, the better the result.

$$h(\mathbf{x}_{it},\, m_{it}) = \lambda_0 + \mathbf{k}(\mathbf{x}_{it}, m_{it})\lambda$$

where $k(.,.)$ is a $1 \times Q$ collection of functions,

$$f(h) = \delta_0 + \delta_1 h + \delta_2 h^2 + \cdots + \delta_G h^G$$

then it implies

$$\begin{aligned} f(\omega_{it}) = \delta_0 &+ \delta_1\{\mathbf{k}(\mathbf{x}_{it-1}, m_{it-1})\lambda_1\} + \delta_2\{\mathbf{k}(\mathbf{x}_{it-1}, m_{it-1})\lambda_1\}^2 + \cdots \\ &+ \delta_G\{\mathbf{k}(\mathbf{x}_{it-1}, m_{it-1})\lambda_1\}^G \end{aligned}$$

For the sake of simplicity, consider the case with $G = 1$ and $\delta_1 = 1$.[5] A simple substitution in (12) and (13) yields

$$y_{it} = \zeta + \mathbf{w}_{it}\boldsymbol{\beta} + \mathbf{x}_{it}\boldsymbol{\gamma} + \mathbf{k}(\mathbf{x}_{it}, m_{it})\lambda_1 + v_{it} \tag{14}$$

$$y_{it} = \theta + \mathbf{w}_{it}\boldsymbol{\beta} + \mathbf{x}_{it}\boldsymbol{\gamma} + \mathbf{k}(\mathbf{x}_{it-1}, m_{it-1})\lambda_1 + \eta_{it} \tag{15}$$

where $\zeta$ and $\theta$ are the new constant parameters obtained through aggregation of all the constant terms.

Under the assumptions of $G = 1$ and $\delta_1 = 1$, the system GMM has linear moments. The choice of instruments for (14) and (15) is straightforward and reflects the orthogonality conditions listed above: in particular, we define $\mathbf{z}_{it1} = [1, \mathbf{x}_{it}, \mathbf{w}_{it}, \mathbf{k}(\mathbf{x}_{it}, m_{it})]$, $\mathbf{z}_{it2} = [1, \mathbf{x}_{it}, \mathbf{w}_{it-1}, \mathbf{k}(\mathbf{x}_{it-1}, m_{it-1})]$, and $\mathbf{Z}_{it} = \begin{pmatrix} \mathbf{z}_{it1} \\ \mathbf{z}_{it2} \end{pmatrix}$.

For each $t > 1$, the usual GMM with IV setup applies, and the moment conditions are derived from the residual functions

$$\mathbf{r}_{it}(\theta) = \begin{bmatrix} r_{it1}(\theta) \\ r_{it2}(\theta) \end{bmatrix} = \begin{bmatrix} y_{it} - \zeta - \mathbf{w}_{it}\boldsymbol{\beta} - \mathbf{x}_{it}\boldsymbol{\gamma} - \mathbf{k}(\mathbf{x}_{it}, m_{it})\lambda_1 \\ y_{it} - \theta - \mathbf{w}_{it}\boldsymbol{\beta} - \mathbf{x}_{it}\boldsymbol{\gamma} - \mathbf{k}(\mathbf{x}_{it-1}, m_{it-1})\lambda_1 \end{bmatrix}$$

and $E\{\mathbf{Z}'_{it}\mathbf{r}_{it}(\theta)\} = 0$.

In this leading case, the estimation is particularly straightforward, because the whole system boils down to a linear estimation problem. Following Wooldridge (2009), we can rewrite the system as $\mathbf{y}_{it} = \mathbf{X}_{it}\boldsymbol{\theta} + \mathbf{r}_{it}$, where $\mathbf{y}_{it}$ is a vector containing $y_{it}$ twice (stacked), $\boldsymbol{\theta}$ is the vector of parameters of interest, $\mathbf{r}_{it}$ is defined as above, and

5. This is the case whose estimation is implemented in `prodest`.

$$\mathbf{X}_{it} = \begin{bmatrix} 1 & 0 & \mathbf{w}_{it} & \mathbf{x}_{it} & \mathbf{k}(\mathbf{x}_{it}, m_{it}) \\ 0 & 1 & \mathbf{w}_{it} & \mathbf{x}_{it} & \mathbf{k}(\mathbf{x}_{it-1}, m_{it-1}) \end{bmatrix}$$

Using $\mathbf{Z}_{it}$ as above yields consistent estimates.

## 2.5 IV estimation of ACF: The Robinson estimator

The WRDG estimator implemented in `prodest` collapses into the Robinson (1988) semi-parametric estimator for the ACF case. If we agree with the input timing in ACF, the first equation would be unable to identify any of the parameters. However, identification is achieved by estimating semiparametrically (15) only.

Following Wooldridge (2009), (13) requires the orthogonality condition

$$E(\eta_{it}|\mathbf{x}_{it}, \mathbf{w}_{it-1}, \mathbf{x}_{it-1}, m_{it-1}, \dots, \mathbf{w}_{i1}, \mathbf{x}_{i1}, m_{i1}) = 0 \qquad t = 2, \dots, T$$

to be consistently estimated. Given that, within the ACF framework it is possible to estimate $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ by an IV version of Robinson's (1988) estimator,[6] with $\mathbf{x}_{it}, \mathbf{x}_{it-1}$, and $m_{it}$ used as included instruments and with $\mathbf{w}_{it-1}$ instrumenting the endogenous $\mathbf{w}_{it}$.

## 2.6 MrEst: Introducing dynamic panel instruments

As Wooldridge (2009) suggests, previous lags are valid instruments in the above GMM estimation framework, but using them can be costly in terms of sample size because each additional lag implies the loss of $n$ observations during the estimation. Because most datasets in the literature have a relatively modest number of observations per panel, this may be problematic. In particular, it could be detrimental in combination with the use of investments as proxy variable (OP), which already leads to a reduced sample in the estimation.

To tackle this issue, we propose using dynamic panel-data instruments à la Blundell and Bond (1998) within the Wooldridge (2009) framework outlined above.

As before, for each $t > 1$ define a $2(T-1)$ residual function vector as

$$\mathbf{r}_i(\theta) = \begin{bmatrix} y_{i2} - \zeta - \mathbf{w}_{i2}\boldsymbol{\beta} - \mathbf{x}_{i2}\boldsymbol{\gamma} - k(\mathbf{x}_{i2}, \mathbf{m}_{i2})\lambda_1 \\ y_{i2} - \theta - \mathbf{w}_{i2}\boldsymbol{\beta} - \mathbf{x}_{i1}\boldsymbol{\gamma} - k(\mathbf{x}_{i1}, \mathbf{m}_{i1})\lambda_1 \\ \dots \\ \dots \\ y_{iT} - \zeta - \mathbf{w}_{iT}\boldsymbol{\beta} - \mathbf{x}_{iT}\boldsymbol{\gamma} - k(\mathbf{x}_{iT}, \mathbf{m}_{iT})\lambda_1 \\ y_{iT} - \theta - \mathbf{w}_{iT}\boldsymbol{\beta} - \mathbf{x}_{iT}\boldsymbol{\gamma} - k(\mathbf{x}_{iT-1}, \mathbf{m}_{iT-1})\lambda_1 \end{bmatrix}$$

6. In the same ACF setting, (12) does not identify $\boldsymbol{\beta}$ even under the orthogonality condition in (10).

For each panel $i$, we define $t-b$, the last available lag (that is, when $b = 1$ at $t = 2$, $b = 2$ at $t = 3$, and $b = T - 1$ at $t = T$). Then let $\mathbf{Z}_i$ denote the dynamic panel instrument matrix for each panel (we suppress the subscript $i$ to avoid an abuse of notation):

$$\mathbf{Z} = \begin{bmatrix} \mathbf{z}_2' & \mathbf{z}_3' & \cdots & \mathbf{z}_T' & 0 & 0 & 0 & 0 \\ 0 & 0 & \cdots & 0 & \widetilde{\mathbf{z}}_3' & 0 & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & \widetilde{\mathbf{z}}_4' & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \widetilde{\mathbf{z}}_T' \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

where component $\widetilde{\mathbf{z}}_t$ is a vector of dimension $1 \times b$ consisting of $\mathbf{z}_{t-1}, \ldots, \mathbf{z}_{t-b}$.

As usual, GMM conditions are defined as

$$E\left\{\mathbf{Z}_i \mathbf{r}_i(\theta)\right\} = 0 \tag{16}$$

Using dynamic panel-data instruments in this setting strengthens the robustness and the efficiency of estimates. Indeed, using (16) allows one to maximize the number of restrictions and enhances parameter identification. In section 4, we report the results of several Monte Carlo simulations to show how our proposed method, which we call MrEst, performs better than WRDG in many applications because of its increased precision.

# 3 Production function estimation using Stata

## 3.1 The prodest command

### Syntax

`prodest` *depvar* [ *if* ] [ *in* ], `method(`*method*`) free(`*varlist*`) proxy(`*varlist*`) state(`*varlist*`)` [ `control(`*varlist*`) endogenous(`*varlist*`) acf valueadded attrition init(`*string*`) translog` *optoptions* `overidentification id(`*varname*`) t(`*varname*`) reps(#) poly(#) seed(#) fsresiduals(`*newvar*`) gmm level(#)` ]

### Options

`method(`*method*`)` specifies the estimation method. *method* may be one of the following: `op` (Olley–Pakes), `lp` (Levinsohn–Petrin), `wrdg` (Wooldridge), `rob` (Wooldridge–Robinson), or `mr` (Mollisi–Rovigatti). `method()` is required.

`free(`*varlist*`)` specifies the free variables, which are ln(labor) in OP, LP, and ACF. `free()` is required.

`proxy(`*varlist*`)` specifies the proxy variables, which are ln(investments) in OP and ln(intermediate inputs) in LP and ACF. `proxy()` is required.

`state(`*varlist*`)` specifies the state variables, which are ln(capital) in OP, LP, and ACF. `state()` is required.

`control(`*varlist*`)` specifies the control variables to be included.

`endogenous(`*varlist*`)` specifies the endogenous variables to be included.

`acf` applies the ACF correction.

`valueadded` indicates that *depvar* is the value to be added to output. The default is to add gross output.

`attrition` corrects for attrition (that is, a firm's exit) in the data.

`init(`*string*`)` specifies the initial starting points for the optimization routine.

`translog` specifies to use a translog production function for estimation. This option is available only with option `acf`.

*optoptions* control the optimization process. Available options are the following:

`optimizer(`*opttype*`)` specifies the optimization technique, where *opttype* may be one of the following: `nm` (Nelder–Mead), `nr` (modified Newton–Raphson), `dfp` (Davidon–Fletcher–Powell), `bfgs` (Broyden–Fletcher–Goldfarb–Shanno), `gn` (Gauss–Newton), or `bhhh` (Berndt–Hall–Hall–Hausman). `nm` and `bhhh` are available only with options `method(op)` and `method(lp)`; `gn` is available only with options `method(wrdg)` and `method(mr)`.

`maxiter(`#`)` specifies the maximum number of iterations. The default is `maxiter(10000)`.

`evaluator(`*string*`)` specifies the evaluator type according to Mata function `optimize_init_evaluatortype()`. The default is `evaluator(d0)`, unless option `optimizer(bhhh)` is specified, in which case the default is `evaluator(gf0)`.

`tolerance(`#`)` sets the tolerance in the optimization algorithm according to Mata function `optimize_init_conv_nrtol()`. The default is `tolerance(e-05)`.

`overidentification` specifies to include the lagged polynomial in state and proxy variables among instruments. This option is available only with options `method(wrdg)` and `method(mr)`.

`id(`*varname*`)` specifies the *panelvar* to which the unit belongs. The user may either specify `id()` or use `xtset` *panelvar timevar* before using the `prodest` command. See [XT] **xtset**.

`t(`*varname*`)` specifies the *timevar* of the observation. The user may either specify `t()` or use `xtset` *panelvar timevar* before using the `prodest` command. See [XT] **xtset**.

`reps(`#`)` specifies the number of bootstrap repetitions. The default is `reps(5)`.

`poly(#)` specifies the degree of polynomial approximations for the first stage. The default is `poly(3)`.

`seed(#)` specifies the seed to be set before estimation. The default is `seed(12345)`.

`fsresiduals(`*newvar*`)` stores the first-stage residuals in *newvar*. This option is available only with options `method(lp)` and `method(op)`.

`gmm` indicates to use the Stata `gmm` command to perform the estimation instead of the Stata `ivregress` command. This option is available only with option `method(wrdg)`.

`level(#)` specifies the confidence level $\alpha$. The default is `level(95)`.

## 3.2 Stored results

`prodest` stores the following in `e()`:

Scalars

| | |
|---|---|
| `e(N)` | number of observations |
| `e(N_g)` | number of panel IDs |
| `e(tmin)` | minimum number of periods |
| `e(tmean)` | average number of periods |
| `e(tmax)` | maximum number of periods |

Macros

| | |
|---|---|
| `e(cmd)` | `prodest` |
| `e(depvar)` | name of dependent variable |
| `e(free)` | free variables |
| `e(state)` | state variables |
| `e(proxy)` | proxy variables |
| `e(control)` | control variables |
| `e(endogenous)` | endogenous variables |
| `e(technique)` | optimization technique |
| `e(idvar)` | ID variable |
| `e(timevar)` | time variable |
| `e(method)` | estimation method |
| `e(model)` | value-added or gross output model |
| `e(correction)` | correction (`acf`) |
| `e(hans_j)` | Hansen's $J$ (`method(wrdg)`) |
| `e(hans_p)` | Hansen's $J$ $p$-value |
| `e(waldT)` | Wald test on constant returns to scale |
| `e(waldP)` | Wald test $p$-value |

Matrices

| | |
|---|---|
| `e(b)` | coefficient vector |
| `e(V)` | variance–covariance matrix of the estimators |

Functions

| | |
|---|---|
| `e(sample)` | estimation sample |

## 3.3 The predict command

### Syntax

`predict` [ *newvar* ] [ *if* ] [ *in* ] [ `, residuals exponential parameters omega` ]

### Options

`residuals` calculates the residuals from the log production function (1).

`exponential` calculates the exponential from the residuals of the log production function (1).

`parameters` calculates the estimated input elasticities. For the Cobb–Douglas production function, they are the estimated parameters. For the translog production function (17), they are $\overline{\boldsymbol{\beta}}_w^{\text{translog}} = \{\sum_{i=1}^{N}\sum_{t=1}^{T}(\widehat{\boldsymbol{\beta}}_w + 2\widehat{\boldsymbol{\beta}}_{ww}\mathbf{w}_{it} + \widehat{\boldsymbol{\beta}}_{wx}\mathbf{x}_{it})\}/N \times T$ for the free variables and $\overline{\boldsymbol{\beta}}_x^{\text{translog}} = \{\sum_{i=1}^{N}\sum_{t=1}^{T}(\widehat{\boldsymbol{\beta}}_x + 2\widehat{\boldsymbol{\beta}}_{xx}\mathbf{x}_{it} + \widehat{\boldsymbol{\beta}}_{wx}\mathbf{w}_{it})\}/N \times T$ for the state variables.

`omega` calculates the predicted values of omega [that is, $\widehat{\phi}_{it} - f(w_{it}, k_{it}, \widehat{\beta})$]. Available only after `prodest, fsresiduals()`.

## 3.4 Example

In the following example, we show the use of `prodest`. Interested readers will find that the syntax is similar to other community-contributed commands for production function estimation, namely, `opreg` (Yasar, Raciborski, and Poi 2008) for OP estimation, `levpet` (Petrin, Poi, and Levinsohn 2004) for LP, and `acfest` (Manjón and Mañez 2016) for ACF.[7] Our command, `prodest`, is able to fit all models, adds new methodologies, is faster in many applications (see tables 1 and 2 for a comparison), allows the user to customize the optimization processes (which is desirable mostly with ACF applications), and uses GMM optimization instead of nonlinear least squares in OP estimation.

7. We will use these three community-contributed commands in this article for results comparison with our new command, `prodest`.

We test `prodest` on a dataset of Chilean firms from 1995–2013.[8] Once the data are uploaded, type

```
. use chile_analysis
. prodest va, method(lp) free(skilled unskilled) proxy(water ele) state(k)
> poly(3) valueadded reps(50)
.........10.........20.........30.........40.........50
lp productivity estimator                       Cobb-Douglas PF
Dependent variable: value added                 Number of obs      =     91598
Group variable (id): ID                         Number of groups   =     17956
Time variable (t): ANIO
                                                Obs per group: min =         1
                                                               avg =       5.1
                                                               max =        14
------------------------------------------------------------------------------
          va |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
    skilled |   .2925325   .0054422    53.75   0.000     .2818659    .3031991
  unskilled |   .1793727   .0047613    37.67   0.000     .1700407    .1887048
           k |   .1457723   .0045962    31.72   0.000     .1367639    .1547807
------------------------------------------------------------------------------
Wald test on Constant returns to scale: Chi2 = 2079.42
                                        p = (0.00)
. prodest va, method(wrdg) free(skilled unskilled) proxy(water ele) state(k)
> poly(3) valueadded overidentification
wrdg productivity estimator                     Cobb-Douglas PF
Dependent variable: value added                 Number of obs      =     69376
Group variable (id): ID                         Number of groups   =     17956
Time variable (t): ANIO
                                                Obs per group: min =         1
                                                               avg =       5.1
                                                               max =        14
------------------------------------------------------------------------------
          va |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
    skilled |   .3373717   .0040003    84.34   0.000     .3295312    .3452122
  unskilled |   .2407734   .0033602    71.65   0.000     .2341875    .2473593
           k |   .1418322   .0025391    55.86   0.000     .1368557    .1468087
------------------------------------------------------------------------------
Wald test on Constant returns to scale: Chi2 = 1939.51
                                        p = (0.00)
Hansen´s J statistic for overidentification = 4592.56
                                        p = (0.00)
```

The output table is similar to that of most Stata panel commands[9] in that it indicates the panel and *timevar*, the dependent variable, the methodology used, and the number of observations and groups.

8. It is a well-known and broadly used dataset in the related literature. See Petrin, Poi, and Levinsohn (2004) among others.
9. In programming the command and output, we were inspired by the community-contributed command `levpet` (Petrin, Poi, and Levinsohn 2004) because of its user-friendly structure and its clear output interpretation.

# 4 Methods

## 4.1 OP, LP, and WRDG methods compared with prodest

In table 1, we compare the OP method across various commands. Columns 1 and 2 refer to OLS and FE methodologies as benchmarks. Columns 3–6 report results of the OP methodology; columns 3 and 4 do not account for attrition, while columns 5 and 6 do. The `levpet` command with `investment` as a proxy variable (column 3) shows the same point estimates as `prodest` (column 4). Because neither of those models accounts for attrition, we also show results of the `opreg` command (column 5) and `prodest` with the `attrition` option (column 6). In both cases, there are no statistical differences between the models' estimates.[10] Our command proves to be faster than any other available.

Table 1. The OP method: Chilean dataset value added

| | OLS | FE | levpet | prodest | opreg | prodest_exit |
|---|---|---|---|---|---|---|
| main | | | | | | |
| $\beta_k$ | 0.116*** | 0.0828*** | 0.402*** | 0.402*** | 0.408*** | 0.398*** |
| | (0.00127) | (0.00126) | (0.00916) | (0.00642) | (0.00929) | (0.00911) |
| $\beta_{sk}$ | 0.668*** | 0.458*** | 0.313*** | 0.313*** | 0.313*** | 0.313*** |
| | (0.00317) | (0.00341) | (0.00742) | (0.00621) | (0.00742) | (0.00621) |
| $\beta_{unsk}$ | 0.436*** | 0.339*** | 0.224*** | 0.224*** | 0.224*** | 0.224*** |
| | (0.00266) | (0.00283) | (0.00665) | (0.00556) | (0.00665) | (0.00556) |
| time | 0.0640 | 0.979 | 84.84 | 55.54 | 297.2 | 201.0 |
| N | 91598 | 91598 | 60253 | 60253 | 60253 | 60253 |

NOTE: Column 1 reports results of a linear regression of log output (value added) on free and state variables. Column 2 reports added individual fixed effects. Column 3 reports results of the `levpet` command with `investment` as proxy variable (`levpet va, free(skilled unskilled) capital(k) proxy(inv) reps(50) valueadded`). Columns 4 and 6 report `prodest` results without and with `attrition`, respectively (`prodest va, method(op) free(skilled unskilled) state(k) proxy(inv) valueadded reps(50)` [`attrition`]). Column 5 reports results of the `opreg` command (`opreg va, exit(exit) free(skilled unskilled) proxy(inv) state(k)`). Standard errors in parentheses.
*** denotes $p < 0.01$; ** denotes $p < 0.05$; * denotes $p < 0.1$.

In table 2, we compare the LP and WRDG methods across various commands. Again, columns 1 and 2 contain results from OLS and FE models as benchmarks. Column 3 reports results of the `levpet` command. And columns 4–6 report results from various implementations of `prodest`, namely, `method(lp)` (column 4), `method(lp)` with `attrition` (column 5), and `method(wrdg)` (column 6). Levinsohn and Petrin (2003) argue that OLS overestimates the parameters of free variables; this is the case in our application as well.[11] `prodest` (column 4) shows the same point estimates as `levpet` (column 3), but it runs faster. Attrition does not significantly affect results. The WRDG method (column 6) yields results consistent with the other commands but with smaller standard errors.

10. This suggests that attrition is a relatively rare phenomenon in these markets.
11. The bias on state-variable parameters, instead, depends on the correlation between inputs and productivity shocks; thus, there exists no prior on it.

Table 2. The LP and WRDG methods: Chilean dataset value added

| | OLS | FE | levpet | prodest | prodest_exit | Wooldridge |
|---|---|---|---|---|---|---|
| $\beta_k$ | 0.116*** | 0.0828*** | 0.146*** | 0.146*** | 0.147*** | 0.135*** |
| | (0.00127) | (0.00126) | (0.00448) | (0.00460) | (0.00482) | (0.00157) |
| $\beta_{sk}$ | 0.668*** | 0.458*** | 0.293*** | 0.293*** | 0.293*** | 0.358*** |
| | (0.00317) | (0.00341) | (0.00721) | (0.00544) | (0.00544) | (0.00449) |
| $\beta_{unsk}$ | 0.436*** | 0.339*** | 0.179*** | 0.179*** | 0.179*** | 0.210*** |
| | (0.00266) | (0.00283) | (0.00653) | (0.00476) | (0.00476) | (0.00342) |
| time | 0.0640 | 0.633 | 153.1 | 110.9 | 206.1 | 3.729 |
| N | 91598 | 91598 | 91598 | 91598 | 91598 | 69376 |

NOTE: Column 1 reports results of a linear regression of log output (value added) on free and state variables. Column 2 reports added individual fixed effects. Column 3 reports results of the `levpet` command with `investment` as proxy variable (`levpet va, free(skilled unskilled) capital(k) proxy(water ele) reps(50) valueadded`). Columns 4 and 5 report `prodest` results without and with `attrition`, respectively (`prodest va, method(lp) free(skilled unskilled) state(k) proxy(water ele) valueadded reps(50) [attrition]`). Column 6 reports results of `prodest` using the WRDG method and a second-degree polynomial (`prodest va, method(wrdg) free(skilled unskilled) state(k) proxy(water ele) poly(2) valueadded`). Standard errors in parentheses.
*** denotes $p < 0.01$; ** denotes $p < 0.05$; * denotes $p < 0.1$.

## 4.2 ACF: Simulations and discussion

Ackerberg, Caves, and Frazer (2015) propose a correction to the OP and LP methodologies. Their article presents i) a new approach, ii) an application to real data, and iii) estimates using simulated data. These have been generated according to three distinct DGPs:

DGP1. With serially correlated wages and labor input decisions set at $t - \delta$

DGP2. With an optimization error in labor

DGP3. With the elements from DGP1 and DGP2 at once

On top of that, they simulate data with four different measurement errors in the intermediate input ($\sigma_m^2 = 0$, 0.1, 0.2, and 0.5).

In tables 3 and 4, we compare `prodest` with the `acfest` command (Manjón and Mañez 2016). More specifically, we run various models on the Chilean dataset and then on a simulated dataset (DGP3) to test several features of both commands.

We first used the Chilean dataset to generate the results reported in table 3. `prodest` (columns 3, 6, and 9) yields more stable and "plausible" results. All estimates $\widehat{\beta}_g^{\text{prodest}} \in [0, 1], g = [sk, unsk, k]$ and (with 50 cluster bootstrap repetitions) standard errors are smaller than `acfest`'s, on average. Timing shows mixed evidence: `prodest` is faster in the gross output, third-degree application but is slower in the value-added output, second- and third-degree applications.

Table 3. The ACF comparison: Chilean dataset

| | GO | | | VA - II | | | VA | | |
|---|---|---|---|---|---|---|---|---|---|
| | LP | acfest | prodest | LP | acfest | prodest | LP | acfest | prodest |
| $\beta_{sk}$ | 0.268*** | 1.991*** | 0.422*** | 0.322*** | −0.147*** | 0.695*** | 0.309*** | −0.212*** | 0.695*** |
| | (0.007) | (0.392) | (0.007) | (0.006) | (0.050) | (0.005) | (0.006) | (0.048) | (0.004) |
| $\beta_{unsk}$ | 0.160*** | −0.528*** | 0.272*** | 0.210*** | −0.089** | 0.460*** | 0.192*** | −0.161*** | 0.461*** |
| | (0.005) | (0.156) | (0.006) | (0.005) | (0.041) | (0.004) | (0.005) | (0.040) | (0.002) |
| $\beta_k$ | 0.073*** | 0.069*** | 0.040*** | 0.139*** | 0.252*** | 0.061*** | 0.143*** | 0.269*** | 0.058*** |
| | (0.003) | (0.011) | (0.003) | (0.005) | (0.009) | (0.004) | (0.005) | (0.009) | (0.005) |
| time | 144 | 741 | 425 | 88 | 265 | 310 | 92 | 261 | 324 |
| N | 93,191 | 71,369 | 93,191 | 91,598 | 70,238 | 91,598 | 91,598 | 70,238 | 91,598 |

NOTE: In columns 1–3, the dependent variable is log(gross output) (GO); in columns 4–9, the dependent variable is log(value added) (VA). Value-added models have been fit with second-degree polynomials (columns 4–6) and third-degree polynomials (columns 7–9). Columns 1, 4, and 7 report the benchmark LP estimates (`prodest go|va, free(skilled unskilled) proxy(ele) state(k) reps(50) method(lp) [va [poly(2)]]`). Columns 2, 5, and 8 report results obtained using the `acfest` command with 50 bootstrap repetitions (`acfest go|va, free(skilled unskilled) proxy(ele) state(k) nbs(50) robust [va [second]]`). Columns 3, 6, and 9 refer to the same models fit with `prodest` (`prodest go|va, free(skilled unskilled) proxy(ele) state(k) acf reps(50) [va [poly(2)]]`). Standard errors in parentheses.
*** denotes $p < 0.01$; ** denotes $p < 0.05$; * denotes $p < 0.1$.

Table 4 shows various value-added models implemented on a DGP3 simulated dataset, with $\beta_k^\star = 0.4$, $\beta_l^\star = 0.6$, and 1,000 firms observed 10 times. Though slower, `prodest` performs better than `acfest` in both the second- and the third-degree polynomial versions. In all cases, the Newton–Raphson (NR) algorithm overcomes the Davidon–Fletcher–Powell (DFP) and Nelder–Mead (NM) algorithms in terms of model mean squared error.

Table 4. The ACF comparison: DGP3 dataset

| | VA - II | | | | | VA | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | LP | acfest | prodest | prodest-DFP | prodest-NR | LP | acfest | prodest | prodest-DFP | prodest-NR |
| $\beta_l$ | 0.477*** | 1.007*** | 0.597*** | 0.610*** | 0.609*** | 0.477*** | 1.007*** | 0.597*** | 0.611*** | 0.600*** |
| | (0.003) | (0.005) | (0.014) | (0.009) | (0.006) | (0.003) | (0.005) | (0.014) | (0.008) | (0.009) |
| $\beta_k$ | 0.564*** | −0.007 | 0.385*** | 0.460*** | 0.444*** | 0.564*** | −0.007 | 0.385*** | 0.462*** | 0.405*** |
| | (0.014) | (0.007) | (0.017) | (0.021) | (0.021) | (0.014) | (0.007) | (0.017) | (0.022) | (0.016) |
| time | 8 | 13 | 19 | 17 | 28 | 8 | 14 | 20 | 19 | 27 |
| N | 10,000 | 9,000 | 10,000 | 10,000 | 10,000 | 10,000 | 9,000 | 10,000 | 10,000 | 10,000 |

NOTE: Columns 1–5 report results of value-added estimation with a second-degree polynomial; columns 6–10 report results of value-added estimation with a third-degree polynomial. Columns 1 and 6 report the LP benchmark results (`prodest lny, free(lnl) proxy(lnm) state(lnk) method(lp) va reps(50) [poly(2)]`). Columns 2 and 7 report `acfest`'s benchmark results (`acfest y, free(l) proxy(m) state(k) va nbs(50) [second] robust`). Columns 3–5 and 8–10 report `prodest` models fit with NM, DFP, and NR optimization algorithms, respectively, (`prodest lny, free(lnl) proxy(lnm) state(lnk) acf va reps(50) [poly(2)] [optimizer(dfp|nr)] init(".5,.5")`). Standard errors in parentheses.
*** denotes $p < 0.01$; ** denotes $p < 0.05$; * denotes $p < 0.1$.

Table 5 replicates table I in Ackerberg, Caves, and Frazer (2015). For each DGP and measurement-error couple, we present the estimated parameters using ACF-corrected and LP parameters. A Monte Carlo simulation of the estimates is performed with 1,000 repetitions, and results reported are coefficient averages with standard deviations across the replications. The true values of $\beta_l$ and $\beta_k$ are 0.6 and 0.4, respectively. ACF persistently performs better than LP, even if the routine takes longer to complete.

Table 5. ACF and LP: Monte Carlo simulations

| | ACF | | | | LP | | | |
|---|---|---|---|---|---|---|---|---|
| | $\beta_l$ | | $\beta_k$ | | $\beta_l$ | | $\beta_k$ | |
| Meas. error | Coeff. | St. Dev. | Coeff. | St. Dev. | Coeff. | St. Dev. | Coeff. | St. Dev. |
| *DGP1: Serially correlated wages and labor set at time $t-b$* | | | | | | | | |
| 0.0 | 0.609 | 0.015 | 0.415 | 0.024 | −0.000 | 0.005 | 1.089 | 0.030 |
| 0.1 | 0.594 | 0.019 | 0.425 | 0.020 | 0.676 | 0.009 | 0.364 | 0.012 |
| 0.2 | 0.634 | 0.024 | 0.399 | 0.017 | 0.788 | 0.007 | 0.241 | 0.010 |
| 0.5 | 0.670 | 0.011 | 0.356 | 0.013 | 0.875 | 0.005 | 0.170 | 0.126 |
| *DGP2: Optimization error in labor* | | | | | | | | |
| 0.0 | 0.619 | 0.022 | 0.424 | 0.024 | 0.600 | 0.003 | 0.399 | 0.013 |
| 0.1 | 0.610 | 0.016 | 0.404 | 0.019 | 0.753 | 0.004 | 0.255 | 0.009 |
| 0.2 | 0.612 | 0.018 | 0.397 | 0.021 | 0.807 | 0.004 | 0.202 | 0.011 |
| 0.5 | 0.651 | 0.019 | 0.374 | 0.016 | 0.863 | 0.003 | 0.312 | 0.304 |
| *DGP3: DGP1 plus DGP2* | | | | | | | | |
| 0.0 | 0.620 | 0.022 | 0.478 | 0.052 | 0.473 | 0.004 | 0.570 | 0.016 |
| 0.1 | 0.610 | 0.019 | 0.440 | 0.022 | 0.634 | 0.005 | 0.412 | 0.012 |
| 0.2 | 0.606 | 0.011 | 0.428 | 0.017 | 0.700 | 0.005 | 0.344 | 0.012 |
| 0.5 | 0.615 | 0.013 | 0.414 | 0.019 | 0.777 | 0.005 | 0.877 | 0.605 |

NOTE: We have 1,000 replications, and standard deviations have been calculated among 1,000 replications. In each replication, the models are fit with `prodest lnva, method(lp) free(lnl) proxy(lnm) state(lnk) poly(3) valueadded reps(50) [acf]`. True values of parameters are $\beta_l = 0.6$ and $\beta_k = 0.4$. $\rho$ is set at 0.7, and we used the NR optimizer.

In table 6, we report the same results in terms of "Bias" and mean squared error ("MSE"). In all but one model (that is, DGP2, no measurement error), ACF shows an MSE that is persistently an order of magnitude smaller than that of LP.

Table 6. ACF and LP bias and MSE: Monte Carlo simulations

| | ACF | | | LP | | |
|---|---|---|---|---|---|---|
| Meas. error | $\text{Bias}_l$ | $\text{Bias}_k$ | MSE | $\text{Bias}_l$ | $\text{Bias}_k$ | MSE |
| | *DGP1: Serially correlated wages and labor set at time $t-b$* | | | | | |
| 0.0 | 0.009 | 0.015 | 0.001 | −0.600 | 0.689 | 0.418 |
| 0.1 | −0.006 | 0.025 | 0.001 | 0.076 | −0.036 | 0.004 |
| 0.2 | 0.034 | −0.001 | 0.001 | 0.188 | −0.159 | 0.030 |
| 0.5 | 0.070 | −0.044 | 0.004 | 0.275 | −0.230 | 0.072 |
| | *DGP2: Optimization error in labor* | | | | | |
| 0.0 | 0.019 | 0.024 | 0.001 | −0.000 | −0.001 | 0.000 |
| 0.1 | 0.010 | 0.004 | 0.000 | 0.153 | −0.145 | 0.022 |
| 0.2 | 0.012 | −0.003 | 0.000 | 0.207 | −0.198 | 0.041 |
| 0.5 | 0.051 | −0.026 | 0.002 | 0.263 | −0.088 | 0.085 |
| | *DGP3: DGP1 plus DGP2* | | | | | |
| 0.0 | 0.020 | 0.078 | 0.005 | −0.127 | 0.170 | 0.023 |
| 0.1 | 0.010 | 0.040 | 0.001 | 0.034 | 0.012 | 0.001 |
| 0.2 | 0.006 | 0.028 | 0.001 | 0.100 | −0.056 | 0.007 |
| 0.5 | 0.015 | 0.014 | 0.000 | 0.177 | 0.477 | 0.312 |

NOTE: We have 1,000 replications. In each replication, the models are fit with `prodest lnva, method(lp) free(lnl) proxy(lnm) state(lnk) poly(3) valueadded reps(50) [acf]`. True values of parameters are $\beta_l = 0.6$ and $\beta_k = 0.4$. $\rho$ is set at 0.7, and we used the NR optimizer. The definition of $\text{Bias}_{ig} = \widehat{\beta}_{ig} - \beta^{\star}_{ig}$, where $g \in [l, k]$ stands for state and free variables, and $i \in [1, \ldots, 1000]$ identifies each Monte Carlo replication. $\text{Bias}_l$ and $\text{Bias}_k$ are averaged across replications. MSE is defined as the average across replications of $\text{MSE}_i = (\text{Bias}^2_{il} + \text{Bias}^2_{ik})/2$.

Results are extremely robust to various specifications of the AR(1) parameter of productivity (that is, $\rho$) and optimizers across simulations; see the *Appendix*.

In empirical applications, however, ACF methodology shows serious limitations. Table 7 reports bias and MSE estimates obtained using a DGP3 simulated dataset with different starting points for the optimization routine with the NR optimizer. The starting points $[\beta^0_{lnl}, \beta^0_{lnk}]$ in column 1 are $[0.1, (1-0.1)]$; the starting points in column 2 are $[0.2, (1-0.2)]$; and so on up to $[0.9, (1-0.9)]$. In column 6 (in boldface), the starting points are fixed at the true values. It is immediately noticeable how results differ dramatically when starting points depart from the true values.[12] This is true in particular for extreme values of the starting points (columns 1 and 9)[13], when the model yields to estimates greater than 1 or to estimates that are nonsignificant or basically 0.

12. Apart from the optimization starting points, the data and the commands are the same: `prodest lny, method(lp) free(lnl) proxy(lnm) state(lnk) poly(3) valueadded reps(50) acf init("'starting_points'")`.
13. In our examples, we fixed the sum of starting points to 1. The bias is even worse when starting from larger values (in absolute terms).

Table 7. ACF (DGP3) bias and MSE: Monte Carlo simulations

Panel a: Meas. err. 0

| | (1) | (2) | (3) | (4) | (5) | (6) | (7) | (8) | (9) |
|---|---|---|---|---|---|---|---|---|---|
| $\text{Bias}_l$ | 0.277 | 0.030 | 0.041 | 0.020 | 0.007 | **0.004** | 0.016 | 0.156 | 0.397 |
| | (0.177) | (0.057) | (0.042) | (0.061) | (0.006) | **(0.004)** | (0.017) | (0.069) | (0.009) |
| $\text{Bias}_k$ | 0.301 | 0.089 | 0.113 | 0.053 | 0.020 | **0.026** | 0.085 | 0.925 | 0.382 |
| | (0.184) | (0.141) | (0.103) | (0.063) | (0.016) | **(0.020)** | (0.085) | (0.930) | (0.023) |
| MSE | 0.116 | 0.016 | 0.013 | 0.005 | 0.000 | **0.001** | 0.008 | 0.875 | 0.152 |
| | (0.096) | (0.109) | (0.048) | (0.025) | (0.001) | **(0.001)** | (0.013) | (1.863) | (0.012) |

Panel b: Meas. err. 0.1

| | (1) | (2) | (3) | (4) | (5) | (6) | (7) | (8) | (9) |
|---|---|---|---|---|---|---|---|---|---|
| $\text{Bias}_l$ | 0.053 | 0.073 | 0.034 | 0.011 | 0.010 | **0.004** | 0.010 | 0.106 | 0.409 |
| | (0.088) | (0.178) | (0.091) | (0.024) | (0.028) | **(0.004)** | (0.006) | (0.108) | (0.053) |
| $\text{Bias}_k$ | 0.146 | 0.194 | 0.056 | 0.038 | 0.030 | **0.029** | 0.091 | 0.251 | 0.480 |
| | (0.249) | (0.334) | (0.092) | (0.026) | (0.026) | **(0.015)** | (0.069) | (0.501) | (0.093) |
| MSE | 0.047 | 0.093 | 0.011 | 0.001 | 0.001 | **0.001** | 0.007 | 0.168 | 0.205 |
| | (0.170) | (0.354) | (0.036) | (0.008) | (0.009) | **(0.001)** | (0.007) | (0.924) | (0.075) |

Panel c: Meas. err. 0.2

| | (1) | (2) | (3) | (4) | (5) | (6) | (7) | (8) | (9) |
|---|---|---|---|---|---|---|---|---|---|
| $\text{Bias}_l$ | 0.054 | 0.032 | 0.036 | 0.009 | 0.019 | **0.006** | 0.011 | 0.140 | 0.329 |
| | (0.108) | (0.101) | (0.073) | (0.020) | (0.057) | **(0.005)** | (0.007) | (0.115) | (0.009) |
| $\text{Bias}_k$ | 0.115 | 0.068 | 0.103 | 0.028 | 0.036 | **0.027** | 0.070 | 0.246 | 0.332 |
| | (0.276) | (0.152) | (0.166) | (0.023) | (0.052) | **(0.013)** | (0.054) | (0.533) | (0.016) |
| MSE | 0.052 | 0.019 | 0.022 | 0.001 | 0.004 | **0.000** | 0.004 | 0.189 | 0.109 |
| | (0.440) | (0.156) | (0.067) | (0.006) | (0.018) | **(0.000)** | (0.004) | (0.756) | (0.009) |

Panel d: Meas. err. 0.5

| | (1) | (2) | (3) | (4) | (5) | (6) | (7) | (8) | (9) |
|---|---|---|---|---|---|---|---|---|---|
| $\text{Bias}_l$ | 0.052 | 0.052 | 0.033 | 0.024 | 0.040 | **0.014** | 0.021 | 0.229 | 0.296 |
| | (0.097) | (0.116) | (0.048) | (0.035) | (0.076) | **(0.009)** | (0.015) | (0.057) | (0.006) |
| $\text{Bias}_k$ | 0.057 | 0.057 | 0.046 | 0.034 | 0.038 | **0.019** | 0.043 | 0.215 | 0.285 |
| | (0.114) | (0.129) | (0.120) | (0.056) | (0.068) | **(0.012)** | (0.045) | (0.269) | (0.012) |
| MSE | 0.014 | 0.018 | 0.010 | 0.003 | 0.007 | **0.000** | 0.002 | 0.087 | 0.085 |
| | (0.105) | (0.158) | (0.107) | (0.012) | (0.021) | **(0.000)** | (0.003) | (0.176) | (0.005) |

NOTE: We have 1,000 replications. All results are averaged across 1,000 Monte Carlo simulations. In columns 1–9, we report the bias in ACF estimates with different optimization starting points from $[0.1, (1 - 0.1)]$ to $[0.9, (1 - 0.9)]$ and the relative MSE. Column 6 (in boldface) contains the results when the starting points equal the true values. Standard errors in parentheses.

In figure 1, we provide a visual of the bias of ACF estimates. The $y$ axis contains the average bias. We define the bias as $|E(\widehat{\theta}) - \theta^{\star}|$. The $x$ axis contains the distance of the starting point from the true parameter value, that is, $\theta^{\circ} - \theta^{\star}$. Both the bias and its standard deviation reach their minimum at $\theta^{\circ} = \theta^{\star}$ (that is, when the starting point is at the true value); however, the bias pattern differs depending on whether the starting point is above or below the true $\theta$. Lower starting points lead to very noisy but not very biased results, while when $\theta^{\circ} > \theta^{\star}$, the bias increases and is statistically significant.

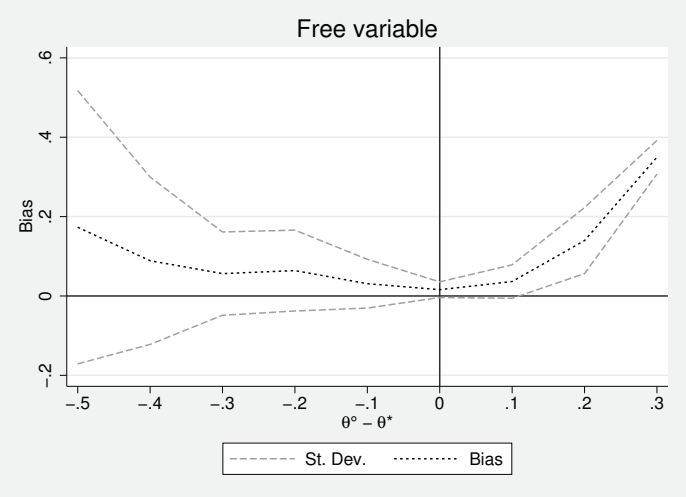


Figure 1. Bias of ACF estimates with respect to starting points: Free variable

In figure 2, we show a heat map of the MSE of 6,400 models fit with an NR optimizer and starting points in a $\pm 0.4$ range around the true parameter value (step $= 0.01$). The MSE is defined as $\sum_j (\widehat{\beta}_j - \beta_j^{\star})^2$, with $j = [k, l]$. More specifically, we use a simulated dataset (DGP3, measurement error $= 0.2$), and for each pair $[\theta_l^{\circ}, \theta_k^{\circ}]$, $\theta_l^{\circ} \in [0.2, 1]$ and $\theta_k^{\circ} \in [0, 0.8]$, we type the command `prodest lny, method(lp) free(lnl) proxy(lnm) state(lnk) poly(3) valueadded acf init("`$\theta_l^{\circ}$`,`$\theta_k^{\circ}$`")`.

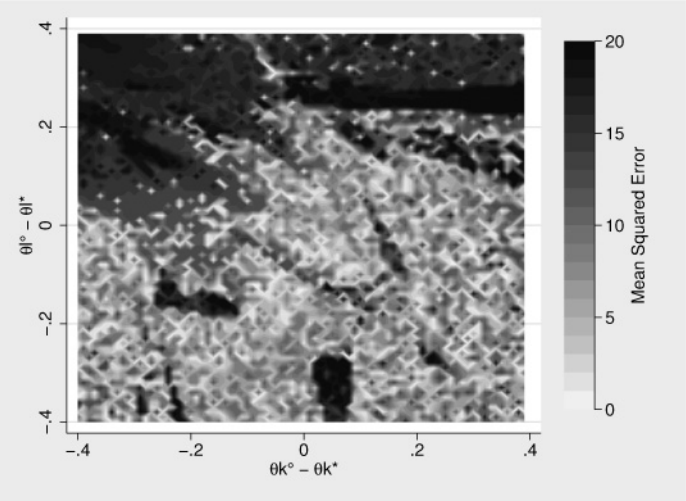


Figure 2. MSE and starting points in ACF estimates

The plot highlights the limitations of ACF methodology with respect to optimization procedures. It is easy to see how the MSE increases as the starting point for the free variable outweighs the true value. Apart from this element, however, no clear pattern emerges: we see a patchwork of high MSE regions throughout the starting points' domain and see single models yielding correct values within regions of severely biased estimates. In turn, this often indicates that the optimization procedures (NR in the present case) are trapped in local maximum.

In the spirit of Knittel and Metaxoglou (2014), who show how often numerical convergence in nonlinear GMM models leads to severely biased results,[14] in the *Appendix*, we have repeated all the above exercises with different choices of optimizers. More specifically, all results hold with NR (table A.2), Broyden–Fletcher–Goldfarb–Shanno (table A.3), DFP (table A.4), and Berndt–Hall–Hall–Hausman (table A.5) optimizers.

## 4.3 Translog production function—ACF

First introduced by Kmenta (1967), the translog production function has been proposed as a feasible approximation of constant elasticity of substitution production functions through a second-order Taylor expansion. Unlike the Cobb–Douglas production function, the translog does not require the assumption of smooth substitution between production factors.

14. In the article, they deal with models à la Berry, Levinsohn, and Pakes (1995).

In this section, we briefly describe the models fit with the `translog` option of ACF methods in `prodest`.[15] The translog production function is

$$y_{it} = \mathbf{w}_{it}\boldsymbol{\beta}_w + \mathbf{x}_{it}\boldsymbol{\beta}_x + \mathbf{w}_{it}^2\boldsymbol{\beta}_{ww} + \mathbf{x}_{it}^2\boldsymbol{\beta}_{xx} + \mathbf{w}_{it}\mathbf{x}_{it}\boldsymbol{\beta}_{wx} + \omega_{it} + \epsilon_{it} \tag{17}$$

The first-stage equation, which is the same in the value-added and gross output cases, turns into

$$\begin{aligned} y_{it} &= \mathbf{w}_{it}\boldsymbol{\beta}_w + \mathbf{x}_{it}\boldsymbol{\beta}_x + \mathbf{w}_{it}^2\boldsymbol{\beta}_{ww} + \mathbf{x}_{it}^2\boldsymbol{\beta}_{xx} + \mathbf{w}_{it}\mathbf{x}_{it}\boldsymbol{\beta}_{wx} + h(p_{it}, \mathbf{x}_{it}, \mathbf{w}_{it}) + \epsilon_{it} \\ &= \Phi(p_{it}, \mathbf{x}_{it}, \mathbf{w}_{it}) + \epsilon_{it} \end{aligned}$$

Once we have obtained $\widehat{\Phi}$, the estimation of productivity terms $\theta = (\boldsymbol{\beta}_w, \boldsymbol{\beta}_x, \boldsymbol{\beta}_{ww}, \boldsymbol{\beta}_{xx}, \boldsymbol{\beta}_{wx})$ in (17) follows as in the usual ACF case.[16] The interpretation of translog parameters, however, differs from that of the Cobb–Douglas ACF. Indeed, the elasticities for free and state variables are given by

$$\begin{aligned} \overline{\boldsymbol{\beta}}_{\text{translog}} &= \frac{\sum_{i=1}^{N}\sum_{t=1}^{T}(\widehat{\boldsymbol{\beta}}_w + 2\widehat{\boldsymbol{\beta}}_{ww}\mathbf{w}_{it} + \widehat{\boldsymbol{\beta}}_{wx}\mathbf{x}_{it})}{N \times T} \\ \overline{\boldsymbol{\gamma}}_{\text{translog}} &= \frac{\sum_{i=1}^{N}\sum_{t=1}^{T}(\widehat{\boldsymbol{\beta}}_x + 2\widehat{\boldsymbol{\beta}}_{xx}\mathbf{x}_{it} + \widehat{\boldsymbol{\beta}}_{wx}\mathbf{w}_{it})}{N \times T} \end{aligned}$$

Table 8 reports results of Cobb–Douglas (odd columns) and translog (even columns) production functions for both value-added (VA) and gross output (GO) models. The number of translog parameters is obviously higher, and increasing the number of parameters implies that GMM optimization takes longer to complete.

15. See De Loecker and Warzynski (2012) and Gandhi, Navarro, and Rivers (2011) for a review of the models presented here.
16. Exploiting that $\widehat{\omega}_{it}(\theta) = \widehat{\Phi}_{it} - \mathbf{w}_{it}\boldsymbol{\beta}_w - \mathbf{x}_{it}\boldsymbol{\beta}_x - \mathbf{w}_{it}^2\boldsymbol{\beta}_{ww} - \mathbf{x}_{it}^2\boldsymbol{\beta}_{xx} - \mathbf{w}_{it}\mathbf{x}_{it}\boldsymbol{\beta}_{wx}$, it is possible to regress it on its past values to recover the productivity shocks $\xi_{it}$ and then proceed with the GMM estimation.

Table 8. Translog production function comparison: Chilean dataset

| | VA | | GO | |
|---|---|---|---|---|
| | Cobb–Douglas | Translog | Cobb–Douglas | Translog |
| $\beta_{sk}$ | 0.617*** | 0.512*** | 0.515*** | 0.816*** |
| | (0.001) | (0.000) | (0.000) | (0.001) |
| $\beta_{unsk}$ | 0.603*** | 0.525*** | 0.523*** | 0.779*** |
| | (0.001) | (0.000) | (0.000) | (0.000) |
| $\beta_{k}$ | 0.053*** | 0.726*** | 0.008*** | 0.573*** |
| | (0.003) | (0.000) | (0.003) | (0.000) |
| $\beta_{water}$ | | | 0.724*** | 0.569*** |
| | | | (0.001) | (0.000) |
| $\beta^2_{sk}$ | | 0.639*** | | 0.603*** |
| | | (0.000) | | (0.000) |
| $\beta_{sk,unsk}$ | | 0.542*** | | 0.540*** |
| | | (0.000) | | (0.000) |
| $\beta_{sk,k}$ | | 0.548*** | | 0.529*** |
| | | (0.000) | | (0.000) |
| $\beta^2_{unsk}$ | | 0.499*** | | 0.815*** |
| | | (0.000) | | (0.001) |
| $\beta_{unsk,k}$ | | 0.492*** | | 0.433*** |
| | | (0.000) | | (0.000) |
| $\beta^2_{k}$ | | −0.302*** | | −0.472*** |
| | | (0.001) | | (0.002) |
| $\beta_{sk,water}$ | | | | 0.713*** |
| | | | | (0.000) |
| $\beta_{unsk,water}$ | | | | 0.551*** |
| | | | | (0.000) |
| $\beta_{k,water}$ | | | | 0.134*** |
| | | | | (0.001) |
| $\beta^2_{water}$ | | | | 0.441*** |
| | | | | (0.000) |
| time | 41 | 190 | 60 | 278 |
| N | 91,598 | 91,598 | 93,191 | 93,191 |
| # Pars | 3 | 9 | 4 | 14 |

NOTE: All models have been fit with `prodest va|go, free(skilled unskilled) proxy(water) state(k) acf [reps(5) init(".5,.5,.5")]`, where `reps()` and `init()` were specified with `va`.
Standard errors in parentheses.
*** denotes $p < 0.01$; ** denotes $p < 0.05$; * denotes $p < 0.1$.

## 4.4 The endogenous() option

The `endogenous()` option in `prodest` allows the user to specify one or more variables that endogenously affect the dynamics of productivity $\omega$. In particular, if any variable $a_{it}$ has an effect on productivity level at time $t$, the law of motion (2) should read $\omega_{it} = g(\omega_{it-1}, a_{it-1}) + \xi_{it}$ (that is, $\omega_{it}$ follows a first-order Markov chain process and $g(\cdot)$ is a nonparametric function of $\omega_{it-1}$ and $a_{it-1}$). This model is able to capture productivity changes conditional on the level of the endogenous variables; that is, it accounts for firms updating their expectation of the productivity level and adjusting their investment based on the optimal level of the endogenous variable.

Models with endogenous variables have been implemented by De Loecker (2007), who use lagged export quotas as drivers of productivity dynamics, and Doraszelski and Jaumandreu (2013), who account for R&D expenditure in estimating $\omega$. In addition, Konings and Vanormelingen (2015) evaluate the impact of workforce training on both output and productivity. Using a subsample of their data, we report in table 9 the results of using the `endogenous()` option on both LP and ACF models.

Table 9. OLS, LP, and ACF models with and without the `endogenous()` option

| | OLS | | LP | | ACF | |
|---|---|---|---|---|---|---|
| | | | Plain | End | Plain | End |
| Labour | 0.617*** | 0.509*** | 0.615*** | 0.615*** | 0.668*** | 0.693*** |
| | (0.00907) | (0.0197) | (0.0178) | (0.0178) | (0.0190) | (0.0126) |
| Capital | 0.191*** | 0.104*** | 0.0969*** | 0.0989*** | 0.135*** | 0.125*** |
| | (0.00484) | (0.0103) | (0.0259) | (0.0265) | (0.0299) | (0.0281) |
| Training | 0.181*** | 0.0405 | 0.0917** | 0.110*** | 0.202*** | 0.192*** |
| | (0.0535) | (0.0313) | (0.0383) | (0.0426) | (0.0586) | (0.0534) |
| N | 2651 | 2651 | 2651 | 2651 | 2651 | 2651 |
| FE | | ✓ | | | | |

NOTE: In columns labeled "Plain", models are reported without the endogenous variable; in columns labeled "End", the endogenous variables are added. The model fit is `prodest y if e(sample), method(lp) free(Labour) state(Capital Training) proxy(Materials) valueadded optimizer(dfp) reps(50)` [`acf`] [`endogenous(lagTraining)`].
Standard errors in parentheses.
*** denotes $p < 0.01$; ** denotes $p < 0.05$; * denotes $p < 0.1$.

## 4.5 WRDG and MrEst

Introducing dynamic panel instruments could be useful in the estimation of "large $N$, small $T$" panel datasets. The overidentification helps improve estimation fit by increasing lags and moment conditions. Table 10 reports the estimation results of LP, WRDG, and MrEst with 2, 3, and 4 lags on subsets of the Chilean data. In particular, we report the $\widehat{\beta}_{sk}$, $\widehat{\beta}_{unsk}$, and $\widehat{\beta}_{k}$,[17] estimated with LP and averaged across 61 industrial sectors (CIIU2) with various sample sizes (panel a). We define $\text{Bias}_j = \widehat{\beta}_j - \beta_j^{lp}, \forall j \in [sk, unsk, k]$ and $\text{MSE} = E(\text{Bias}_j^2)$ (that is, we test WRDG and MrEst performance with respect to the benchmark LP method) and report their average values across sectors.

Table 10. WRDG and MrEst (various DGP)

Panel a: LP

| | $\widehat{\beta}_{sk}$ | $\widehat{\beta}_{unsk}$ | $\widehat{\beta}_{k}$ | MSE |
|---|---|---|---|---|
| Levinsohn–Petrin | 0.303 | 0.228 | 0.039 | 0.000 |
| | (0.121) | (0.086) | (0.045) | (0.000) |

Panel b: WRDG and MrEst: Bias + MSE

| | $\text{Bias}_{sk}$ | $\text{Bias}_{unsk}$ | $\text{Bias}_{k}$ | MSE |
|---|---|---|---|---|
| Wooldridge | −0.007 | −0.014 | −0.003 | 0.002 |
| | (0.050) | (0.041) | (0.021) | (0.002) |
| MrEst - 2 lags | −0.025 | −0.017 | −0.004 | 0.001 |
| | (0.042) | (0.036) | (0.016) | (0.002) |
| MrEst - 3 lags | −0.025 | −0.014 | −0.002 | 0.001 |
| | (0.043) | (0.034) | (0.013) | (0.002) |
| MrEst - 4 lags | −0.026 | −0.014 | −0.004 | 0.001 |
| | (0.044) | (0.033) | (0.012) | (0.002) |

NOTE: In panel a, we report the average $\widehat{\beta}$ value of LP estimation on 60 subsets (that is, industry sectors, according to the CIIU2 variable) of Chilean firm-level data. These are the benchmark values: we define $\text{Bias}_j = \widehat{\beta}_j - \beta_j^{lp}$, $\forall j \in [sk, unsk, k]$ and $\text{MSE} = E(\text{Bias}_j^2)$. In panel b, we report the average bias and the MSE, with their standard deviations, of WRDG and MrEst models. Standard errors in parentheses.

17. This refers to skilled labor, unskilled labor, and capital, respectively.

MrEst consistently performs better than WRDG in terms of mean squared error—even if not always in terms of bias. MrEst models are particularly time intensive, but the computational time does not increase dramatically with the number of lags required for each dynamic panel instrument.

In table 11, we report the MSE of MrEst with 2, 3, or 4 lags on simulated data (DGP2, no measurement error) as $N$ increases with $T$ fixed (panel a) and as $T$ increases with $N$ fixed (panel b). Increasing the sample size leads to lower MSE, and adding lags increases estimate precision, as expected. Increasing the time dimension while keeping $N$ fixed, however, does not appear to have clear effects on the MSE.

Table 11. MrEst: MSE with simulated data (DGP2)

Panel a: $N \to \infty$, fixed $T = 10$

| | (1) | (2) | (3) | (4) | (5) | (6) |
|---|---|---|---|---|---|---|
| MrEst - 2 lags | 0.184 | 0.111 | 0.067 | 0.026 | 0.000 | 0.005 |
| | (0.000) | (0.000) | (0.000) | (0.000) | (0.000) | (0.000) |
| MrEst - 3 lags | 0.179 | 0.106 | 0.064 | 0.025 | 0.000 | 0.004 |
| | (0.000) | (0.000) | (0.000) | (0.000) | (0.000) | (0.000) |
| MrEst - 4 lags | 0.175 | 0.099 | 0.062 | 0.025 | 0.000 | 0.004 |
| | (0.000) | (0.000) | (0.000) | (0.000) | (0.000) | (0.000) |
| N | 1500 | 3000 | 5000 | 6500 | 8000 | 10000 |

Panel b: increasing $T$ to 10, fixed $N$

| | (1) | (2) | (3) | (4) | (5) | (6) |
|---|---|---|---|---|---|---|
| MrEst - 2 lags | 0.032 | 0.080 | 0.070 | 0.040 | 0.052 | 0.067 |
| | (0.000) | (0.000) | (0.000) | (0.000) | (0.000) | (0.000) |
| MrEst - 3 lags | 0.029 | 0.076 | 0.067 | 0.037 | 0.049 | 0.064 |
| | (0.000) | (0.000) | (0.000) | (0.000) | (0.000) | (0.000) |
| MrEst - 4 lags | 0.028 | 0.075 | 0.065 | 0.036 | 0.048 | 0.062 |
| | (0.000) | (0.000) | (0.000) | (0.000) | (0.000) | (0.000) |
| N | 2500 | 3000 | 3500 | 4000 | 4500 | 5000 |

NOTE: Standard errors in parentheses.

## 5 Conclusions

In the literature, there are three main approaches to production function estimation, namely, instrumental-variables, fixed-effects, and control function approaches. Olley and Pakes (1996), Levinsohn and Petrin (2003), and Ackerberg, Caves, and Frazer (2015) all decisively contributed to the control function approach by developing widely used methodologies. Wooldridge (2009) showed how to implement the control function approach in a system GMM framework. We build a new estimator, MrEst, based on his results, adding dynamic panel instruments to improve efficiency and gain predictive power. Our estimator proves to be consistent and to perform better than the WRDG estimator as the number of individuals increases.

We provide evidence that nonlinear GMM models in general, and OP and LP models with ACF correction in particular, must be handled with care in empirical applications. In our Monte Carlo simulations (based on the ACF DGP), results change dramatically depending on the starting points passed to the optimization algorithm; this result is robust to different choices of optimizer, model, and sample.

We introduced a new Stata command, `prodest`, aimed at implementing all the methods listed above in a user-friendly and effective way. It performs well in comparison with other community-contributed commands and introduces new estimation methods. It features several options for expert users aimed at controlling optimization procedures, model specification, and results handling.

## 6 Acknowledgments

Special thanks to Francesco Decarolis, who supervised the whole project, providing expertise and support. We also thank Daniel Ackerberg, Federico Belotti, Sara Calligaris, Leonardo Giuffrida, Andrea Pozzi, Lorenzo Rovigatti, and several `prodest` users for comments that improved the command and the article. We thank an anonymous referee for precious insights on command implementation. Any errors are our own.

## 7 References

Ackerberg, D. A., K. Caves, and G. Frazer. 2015. Identification properties of recent production function estimators. *Econometrica* 83: 2411–2451.

Berry, S., J. Levinsohn, and A. Pakes. 1995. Automobile prices in market equilibrium. *Econometrica* 63: 841–890.

Blundell, R., and S. Bond. 1998. Initial conditions and moment restrictions in dynamic panel data models. *Journal of Econometrics* 87: 115–143.

Bond, S., and M. Söderbom. 2005. Adjustment costs and the identification of Cobb Douglas production functions. Working Paper WP05/04, Institute for Fiscal Studies. http://www.ifs.org.uk/wps/wp0504.pdf.

Cobb, C. W., and P. H. Douglas. 1928. A theory of production. *American Economic Review* 18 (Suppl. 1): 139–165.

De Loecker, J. 2007. Do exports generate higher productivity? Evidence from Slovenia. *Journal of International Economics* 73: 69–98.

De Loecker, J., and F. Warzynski. 2012. Markups and firm-level export status. *American Economic Review* 102: 2437–2471.

Doraszelski, U., and J. Jaumandreu. 2013. R&D and productivity: Estimating endogenous productivity. *Review of Economic Studies* 80: 1338–1383.

Gandhi, A., S. Navarro, and D. Rivers. 2011. On the identification of production functions: How heterogeneous is productivity? Report, Winsconsin–Madison.

Hopenhayn, H. A. 1992. Entry, exit, and firm dynamics in long run equilibrium. *Econometrica* 60: 1127–1150.

Kmenta, J. 1967. On estimation of the CES production function. *International Economic Review* 8: 180–189.

Knittel, C. R., and K. Metaxoglou. 2014. Estimation of random-coefficient demand models: Two empiricists' perspective. *Review of Economics and Statistics* 96: 34–59.

Konings, J., and S. Vanormelingen. 2015. The impact of training on productivity and wages: Firm-level evidence. *Review of Economics and Statistics* 97: 485–497.

Levinsohn, J., and A. Petrin. 2003. Estimating production functions using inputs to control for unobservables. *Review of Economic Studies* 70: 317–341.

Manjón, M., and J. Mañez. 2016. Production function estimation in Stata using the Ackerberg–Caves–Frazer method. *Stata Journal* 16: 900–916.

Melitz, M. J. 2003. The impact of trade on intra-industry reallocations and aggregate industry productivity. *Econometrica* 71: 1695–1725.

Olley, G. S., and A. Pakes. 1996. The dynamics of productivity in the telecommunications equipment industry. *Econometrica* 64: 1263–1297.

Petrin, A., B. P. Poi, and J. Levinsohn. 2004. Production function estimation in Stata using inputs to control for unobservables. *Stata Journal* 4: 113–123.

Robinson, P. M. 1988. Root-n-consistent semiparametric regression. *Econometrica* 56: 931–954.

Wooldridge, J. M. 1996. Estimating systems of equations with different instruments for different equations. *Journal of Econometrics* 74: 387–405.

———. 2009. On estimating firm-level production functions using proxy variables to control for unobservables. *Economics Letters* 104: 112–114.

Yasar, M., R. Raciborski, and B. Poi. 2008. Production function estimation in Stata using the Olley and Pakes method. *Stata Journal* 8: 221–231.

### About the authors

Gabriele Rovigatti graduated in economics and finance at Tor Vergata University in Rome, Italy, and is a post-doc visiting researcher at Chicago Booth University and research scholar at the Einaudi Institute for Economics and Finance in Rome, Italy.

Vincenzo Mollisi is a PhD candidate in law, economics, and institutions at Tor Vergata University in Rome, Italy, and is a visiting researcher at the Free University of Bozen–Bolzano.

# A Appendix

Table A.1. ACF and LP, $\rho = 0.5$

Panel a: Meas. err. 0

| | DGP1 | | DGP2 | | DGP3 | |
|---|---|---|---|---|---|---|
| | ACF | LP | ACF | LP | ACF | LP |
| $\beta_l$ | 0.613*** | 0.00129 | 0.630*** | 0.600*** | 0.597*** | 0.437*** |
| | (0.0189) | (0.00438) | (0.0140) | (0.00268) | (0.00905) | (0.00288) |
| $\beta_k$ | 0.391*** | 1.022*** | 0.384*** | 0.401*** | 0.402*** | 0.576*** |
| | (0.0175) | (0.0181) | (0.0110) | (0.00784) | (0.0129) | (0.0107) |
| time | 50.44 | 8.574 | 35.98 | 7.949 | 47.78 | 7.878 |

Panel b: Meas. err. 0.1

| | ACF | LP | ACF | LP | ACF | LP |
|---|---|---|---|---|---|---|
| $\beta_l$ | 0.611*** | 0.549*** | 0.641*** | 0.750*** | 0.605*** | 0.584*** |
| | (0.0360) | (0.00566) | (0.0213) | (0.00241) | (0.00848) | (0.00322) |
| $\beta_k$ | 0.405*** | 0.470*** | 0.368*** | 0.253*** | 0.403*** | 0.434*** |
| | (0.0226) | (0.0103) | (0.0155) | (0.00474) | (0.0123) | (0.00871) |
| time | 51.26 | 8.478 | 34.20 | 7.862 | 50.59 | 8.444 |

Panel c: Meas. err. 0.2

| | ACF | LP | ACF | LP | ACF | LP |
|---|---|---|---|---|---|---|
| $\beta_l$ | 0.662*** | 0.698*** | 0.649*** | 0.805*** | 0.615*** | 0.654*** |
| | (0.0340) | (0.00632) | (0.0166) | (0.00242) | (0.0193) | (0.00379) |
| $\beta_k$ | 0.368*** | 0.317*** | 0.358*** | 0.199*** | 0.403*** | 0.364*** |
| | (0.0237) | (0.00923) | (0.0147) | (0.00503) | (0.0151) | (0.00807) |
| time | 37.41 | 8.449 | 23.65 | 7.806 | 48.69 | 9.494 |

Panel d: Meas. err. 0.5

| | ACF | LP | ACF | LP | ACF | LP |
|---|---|---|---|---|---|---|
| $\beta_l$ | 0.668*** | 0.821*** | 0.659*** | 0.862*** | 0.624*** | 0.734*** |
| | (0.0300) | (0.00534) | (0.0182) | (0.00230) | (0.0270) | (0.00391) |
| $\beta_k$ | 0.358*** | 0.189*** | 0.354*** | 0.140*** | 0.402*** | 0.282*** |
| | (0.0246) | (0.00781) | (0.0138) | (0.00506) | (0.0193) | (0.00764) |
| time | 29.56 | 8.222 | 20.90 | 11.28 | 49.68 | 12.06 |
| N | 10000 | 10000 | 10000 | 10000 | 10000 | 10000 |

NOTE: Models have been fit with `prodest lnva, method(lp) free(lnl) proxy(lnm) state(lnk) poly(3) valueadded reps(50) [acf]`. Standard errors in parentheses.

Table A.2. ACF (`nr`), various starting points, DGP3

Panel a: Meas. err. 0

| | (1) | (2) | (3) | (4) | (5) | (6) | (7) | (8) | (9) |
|---|---|---|---|---|---|---|---|---|---|
| $\beta_l$ | 0.994*** | 0.600*** | 0.626*** | 0.615*** | 0.604*** | 0.598*** | 0.632*** | 0.790*** | 1.001*** |
| | (0.185) | (0.0995) | (0.0407) | (0.0145) | (0.00824) | (0.00537) | (0.0273) | (0.0761) | (0.00951) |
| $\beta_k$ | 0.0303 | 0.409*** | 0.475*** | 0.462*** | 0.419*** | 0.397*** | 0.484*** | 1.252 | 0.0101 |
| | (0.246) | (0.105) | (0.0695) | (0.0328) | (0.0166) | (0.0203) | (0.0907) | (0.882) | (0.0259) |
| time | 30.36 | 28.21 | 29.43 | 23.86 | 20.59 | 13.22 | 27.22 | 10560.1 | 34.70 |

Panel b: Meas. err. 0.1

| | (1) | (2) | (3) | (4) | (5) | (6) | (7) | (8) | (9) |
|---|---|---|---|---|---|---|---|---|---|
| $\beta_l$ | 0.610*** | 0.615*** | 0.634*** | 0.628*** | 0.602*** | 0.609*** | 0.623*** | 0.618*** | 1.085*** |
| | (0.131) | (0.108) | (0.105) | (0.00839) | (0.137) | (0.00438) | (0.00722) | (0.110) | (0.0578) |
| $\beta_k$ | 0.416 | 0.427* | 0.433*** | 0.431*** | 0.396*** | 0.415*** | 0.236*** | 0.424 | −0.213** |
| | (0.280) | (0.251) | (0.128) | (0.0178) | (0.144) | (0.0126) | (0.0833) | (0.410) | (0.0965) |
| time | 34.08 | 27.94 | 25.14 | 19.31 | 23.16 | 13.52 | 24.34 | 3806.3 | 22.84 |

Panel c: Meas. err. 0.2

| | (1) | (2) | (3) | (4) | (5) | (6) | (7) | (8) | (9) |
|---|---|---|---|---|---|---|---|---|---|
| $\beta_l$ | 0.613 | 0.660*** | 0.615*** | 0.606*** | 0.602*** | 0.614*** | 0.607*** | 0.619*** | 0.937*** |
| | (0.405) | (0.0772) | (0.0854) | (0.00877) | (0.124) | (0.00705) | (0.00847) | (0.109) | (0.0139) |
| $\beta_k$ | 0.412 | 0.423 | 0.414** | 0.400*** | 0.396*** | 0.414*** | 0.409*** | 0.418 | 0.0591** |
| | (0.516) | (0.312) | (0.200) | (0.0150) | (0.127) | (0.0141) | (0.0698) | (0.468) | (0.0292) |
| time | 37.69 | 35.37 | 26.94 | 20.41 | 20.56 | 13.00 | 22.31 | 31.04 | 17.94 |

Panel d: Meas. err. 0.5

| | (1) | (2) | (3) | (4) | (5) | (6) | (7) | (8) | (9) |
|---|---|---|---|---|---|---|---|---|---|
| $\beta_l$ | 0.841*** | 0.690*** | 0.620*** | 0.614*** | 0.606*** | 0.616*** | 0.624*** | 0.656*** | 0.897*** |
| | (0.0709) | (0.135) | (0.0254) | (0.0215) | (0.0541) | (0.00961) | (0.0266) | (0.0618) | (0.00678) |
| $\beta_k$ | 0.261*** | 0.385*** | 0.417*** | 0.420*** | 0.421*** | 0.419*** | 0.413*** | 0.400*** | 0.114*** |
| | (0.0852) | (0.114) | (0.0534) | (0.0197) | (0.0517) | (0.0134) | (0.0690) | (0.115) | (0.0120) |
| time | 38.64 | 32.04 | 34.67 | 30.98 | 19.41 | 14.46 | 19.91 | 23.44 | 13.76 |

NOTE: In columns 1–9, we report ACF estimates with different optimization starting points from $[0.1, (1 - 0.1)]$ to $[0.9, (1 - 0.9)]$, respectively. Standard errors in parentheses.

Table A.3. ACF (`bfgs`), various starting points, DGP3

Panel a: Meas. err. 0

| | (1) | (2) | (3) | (4) | (5) | (6) | (7) | (8) | (9) |
|---|---|---|---|---|---|---|---|---|---|
| $\beta_l$ | 0.665*** | 0.598*** | 0.600*** | 0.600*** | 0.600*** | 0.599*** | 0.594*** | 0.792*** | 1.007*** |
| | (0.0476) | (0.110) | (0.0174) | (0.0197) | (0.00912) | (0.00752) | (0.00606) | (0.0914) | (0.00570) |
| $\beta_k$ | −0.311 | 0.394** | 0.406*** | 0.414*** | 0.408*** | 0.400*** | 0.324*** | 1.146 | −0.00699 |
| | (0.447) | (0.165) | (0.0430) | (0.0435) | (0.0335) | (0.00142) | (0.0119) | (0.884) | (0.00801) |
| time | 22.88 | 12.97 | 16.27 | 17.24 | 14.71 | 7.172 | 8.781 | 79.84 | 16.59 |

Panel b: Meas. err. 0.1

| | (1) | (2) | (3) | (4) | (5) | (6) | (7) | (8) | (9) |
|---|---|---|---|---|---|---|---|---|---|
| $\beta_l$ | 0.559 | 0.590*** | 0.723*** | 0.610*** | 0.610*** | 0.604*** | 0.607*** | 0.605*** | 0.948*** |
| | (0.346) | (0.0618) | (0.0734) | (0.0350) | (0.0109) | (0.00685) | (0.00843) | (0.110) | (0.00535) |
| $\beta_k$ | 0.503 | 0.362*** | −0.277 | 0.417*** | 0.416*** | 0.401*** | 0.411*** | 0.417* | 0.0589*** |
| | (0.404) | (0.140) | (0.211) | (0.0895) | (0.0123) | (0.00679) | (0.0367) | (0.241) | (0.00941) |
| time | 23.16 | 22.93 | 15.86 | 13.59 | 12.50 | 7.902 | 11.38 | 30.11 | 13.93 |

Panel c: Meas. err. 0.2

| | (1) | (2) | (3) | (4) | (5) | (6) | (7) | (8) | (9) |
|---|---|---|---|---|---|---|---|---|---|
| $\beta_l$ | 0.929*** | 0.512 | 0.674 | 0.673*** | 0.618*** | 0.610*** | 0.601*** | 0.859*** | 0.929*** |
| | (0.0868) | (0.663) | (0.695) | (0.0521) | (0.0108) | (0.0110) | (0.00790) | (0.111) | (0.00591) |
| $\beta_k$ | 0.0791 | 0.494 | 0.120 | 0.122 | 0.418*** | 0.403*** | 0.339*** | 1.197* | 0.0788*** |
| | (0.119) | (0.719) | (0.756) | (0.171) | (0.0205) | (0.00972) | (0.0313) | (0.624) | (0.0150) |
| time | 22.76 | 20.81 | 19.04 | 16.11 | 10.17 | 8.177 | 10.62 | 76.07 | 11.27 |

Panel d: Meas. err. 0.5

| | (1) | (2) | (3) | (4) | (5) | (6) | (7) | (8) | (9) |
|---|---|---|---|---|---|---|---|---|---|
| $\beta_l$ | 0.647* | 0.617** | 0.618 | 0.623*** | 0.617*** | 0.606*** | 0.623*** | 0.894*** | 0.902*** |
| | (0.365) | (0.293) | (0.424) | (0.0243) | (0.0189) | (0.0110) | (0.0139) | (0.0671) | (0.00398) |
| $\beta_k$ | 0.413 | 0.418 | 0.418 | 0.420*** | 0.417*** | 0.422*** | 0.396*** | 0.407** | 0.101*** |
| | (0.402) | (0.321) | (0.473) | (0.0640) | (0.0194) | (0.0143) | (0.0368) | (0.196) | (0.00477) |
| time | 21.56 | 22.25 | 19.90 | 18.09 | 14.39 | 8.183 | 12.51 | 11.83 | 7.139 |

NOTE: In columns 1–9, we report ACF estimates with different optimization starting points from $[0.1, (1 - 0.1)]$ to $[0.9, (1 - 0.9)]$, respectively. Standard errors in parentheses.

Table A.4. ACF (`dfp`), various starting points, DGP3

Panel a: Meas. err. 0

| | (1) | (2) | (3) | (4) | (5) | (6) | (7) | (8) | (9) |
|---|---|---|---|---|---|---|---|---|---|
| $\beta_l$ | 0.664*** | 0.598*** | 0.630*** | 0.635*** | 0.611*** | 0.599*** | 0.594*** | 0.769*** | 1.007*** |
| | (0.0762) | (0.110) | (0.0185) | (0.0216) | (0.00945) | (0.00752) | (0.00611) | (0.0822) | (0.00793) |
| $\beta_k$ | −0.315 | 0.394** | 0.495*** | 0.510*** | 0.462*** | 0.400*** | 0.324*** | 0.814 | −0.00803 |
| | (0.261) | (0.175) | (0.0458) | (0.0490) | (0.0234) | (0.00142) | (0.00481) | (0.886) | (0.0134) |
| time | 16.97 | 14.85 | 17.03 | 22.17 | 12.24 | 7.139 | 8.752 | 46.04 | 28.89 |

Panel b: Meas. err. 0.1

| | (1) | (2) | (3) | (4) | (5) | (6) | (7) | (8) | (9) |
|---|---|---|---|---|---|---|---|---|---|
| $\beta_l$ | 0.552*** | 0.535*** | 0.711*** | 0.610*** | 0.610*** | 0.604*** | 0.608*** | 0.869*** | 0.947*** |
| | (0.0871) | (0.0788) | (0.0622) | (0.0363) | (0.00843) | (0.00702) | (0.00716) | (0.0901) | (0.00880) |
| $\beta_k$ | 0.141 | 0.171 | −0.141 | 0.417*** | 0.416*** | 0.401*** | 0.415*** | 0.365 | 0.0618*** |
| | (0.188) | (0.244) | (0.197) | (0.104) | (0.0151) | (0.00590) | (0.0163) | (0.543) | (0.0190) |
| time | 28.71 | 25.08 | 17.55 | 13.05 | 12.79 | 7.769 | 10.20 | 30.31 | 21.88 |

Panel c: Meas. err. 0.2

| | (1) | (2) | (3) | (4) | (5) | (6) | (7) | (8) | (9) |
|---|---|---|---|---|---|---|---|---|---|
| $\beta_l$ | 0.616 | 0.520** | 0.909 | 0.673*** | 0.618*** | 0.610*** | 0.601*** | 0.856*** | 0.920*** |
| | (0.380) | (0.229) | (0.616) | (0.0591) | (0.0122) | (0.0109) | (0.00893) | (0.0760) | (0.00482) |
| $\beta_k$ | 0.411 | 0.459 | 0.105 | 0.122 | 0.418*** | 0.403*** | 0.339*** | 0.641* | 0.105*** |
| | (0.438) | (0.305) | (0.669) | (0.180) | (0.0209) | (0.00935) | (0.0213) | (0.380) | (0.0125) |
| time | 34.62 | 29.45 | 25.71 | 16.44 | 10.98 | 8.100 | 10.16 | 36.71 | 10.82 |

Panel d: Meas. err. 0.5

| | (1) | (2) | (3) | (4) | (5) | (6) | (7) | (8) | (9) |
|---|---|---|---|---|---|---|---|---|---|
| $\beta_l$ | 0.846*** | 0.666*** | 0.907*** | 0.651*** | 0.625*** | 0.606*** | 0.592*** | 0.893*** | 0.902*** |
| | (0.247) | (0.124) | (0.0493) | (0.0455) | (0.0364) | (0.0110) | (0.0139) | (0.0358) | (0.00348) |
| $\beta_k$ | 0.255 | 0.406* | 0.0785 | 0.408*** | 0.412*** | 0.422*** | 0.358*** | 0.415*** | 0.101*** |
| | (0.271) | (0.222) | (0.114) | (0.118) | (0.0566) | (0.0142) | (0.0306) | (0.0624) | (0.00185) |
| time | 42.35 | 60.63 | 46.65 | 37.31 | 21.72 | 8.193 | 10.39 | 10.66 | 7.095 |

NOTE: In columns 1–9, we report ACF estimates with different optimization starting points from $[0.1, (1 - 0.1)]$ to $[0.9, (1 - 0.9)]$, respectively. Standard errors in parentheses.

Table A.5. ACF (`bhhh`), various starting points, DGP3

Panel a: Meas. err. 0

| | (1) | (2) | (3) | (4) | (5) | (6) | (7) | (8) | (9) |
|---|---|---|---|---|---|---|---|---|---|
| $\beta_l$ | 0.619*** | 0.617*** | 0.594*** | 0.673*** | 0.669*** | 0.599*** | 0.644*** | 0.624*** | 0.971*** |
| | (0.122) | (0.0636) | (0.00662) | (0.0290) | (0.0375) | (0.00470) | (0.0204) | (0.188) | (0.00993) |
| $\beta_k$ | 0.469 | 0.129 | 0.380*** | −0.506** | −0.628 | 0.403*** | 0.502*** | 0.0690 | 0.171*** |
| | (0.305) | (0.402) | (0.0262) | (0.229) | (0.683) | (0.0182) | (0.0467) | (0.173) | (0.0493) |
| time | 55778.1 | 36491.7 | 16269.2 | 81819.9 | 97641.4 | 15458.1 | 14277.6 | 30696.4 | 14450.4 |

Panel b: Meas. err. 0.1

| | (1) | (2) | (3) | (4) | (5) | (6) | (7) | (8) | (9) |
|---|---|---|---|---|---|---|---|---|---|
| $\beta_l$ | 0.125 | 0.617** | 0.830*** | 0.588*** | 0.620*** | 0.603*** | 0.605*** | 0.920*** | 0.920*** |
| | (0.271) | (0.258) | (0.154) | (0.0296) | (0.0387) | (0.00507) | (0.0135) | (0.0389) | (0.00833) |
| $\beta_k$ | 0.866*** | 0.428** | 0.532*** | 0.394*** | 0.239 | 0.408*** | 0.405*** | 0.159*** | 0.158*** |
| | (0.232) | (0.180) | (0.201) | (0.0903) | (0.209) | (0.0123) | (0.0164) | (0.0547) | (0.0288) |
| time | 17636.8 | 21812.7 | 27509.9 | 17593.2 | 17792.3 | 13932.1 | 14142.6 | 15666.7 | 14428.2 |

Panel c: Meas. err. 0.2

| | (1) | (2) | (3) | (4) | (5) | (6) | (7) | (8) | (9) |
|---|---|---|---|---|---|---|---|---|---|
| $\beta_l$ | 0.0901 | 0.926*** | 0.317 | 0.626*** | 0.573*** | 0.604*** | 0.613*** | 0.859*** | 0.912*** |
| | (0.289) | (0.214) | (0.202) | (0.0800) | (0.0233) | (0.00748) | (0.0124) | (0.0568) | (0.00586) |
| $\beta_k$ | 0.885*** | 0.0875 | 0.690*** | 0.419*** | 0.407*** | 0.408*** | 0.413*** | 1.237*** | 0.133*** |
| | (0.258) | (0.178) | (0.207) | (0.0928) | (0.0460) | (0.0137) | (0.0125) | (0.284) | (0.0187) |
| time | 16243.4 | 17217.9 | 21202.8 | 17543.8 | 15956.4 | 14076.1 | 14036.5 | 25479.9 | 16253.9 |

Panel d: Meas. err. 0.5

| | (1) | (2) | (3) | (4) | (5) | (6) | (7) | (8) | (9) |
|---|---|---|---|---|---|---|---|---|---|
| $\beta_l$ | 0.0873 | 0.530*** | 0.296** | 0.408*** | 0.581*** | 0.637*** | 0.662*** | 0.854*** | 0.901*** |
| | (0.210) | (0.168) | (0.134) | (0.103) | (0.0395) | (0.0129) | (0.0365) | (0.0668) | (0.00294) |
| $\beta_k$ | 0.887*** | 0.460*** | 0.696*** | 0.596*** | 0.429*** | 0.411*** | 0.401*** | 0.235*** | 0.102*** |
| | (0.205) | (0.146) | (0.115) | (0.118) | (0.0309) | (0.00879) | (0.0168) | (0.0729) | (0.00546) |
| time | 15207.0 | 14494.8 | 14516.6 | 17186.1 | 14295.1 | 16183.7 | 14129.8 | 20289.1 | 68515.0 |

NOTE: In columns 1–9, we report ACF estimates with different optimization starting points from $[0.1, (1 - 0.1)]$ to $[0.9, (1 - 0.9)]$, respectively. Standard errors in parentheses.

Table A.6. ACF and LP (`nr`), $\rho = 0.6$

| | ACF | | | | LP | | | |
|---|---|---|---|---|---|---|---|---|
| | $\beta_l$ | | $\beta_k$ | | $\beta_l$ | | $\beta_k$ | |
| Meas. error | Coeff. | St. Dev. | Coeff. | St. Dev. | Coeff. | St. Dev. | Coeff. | St. Dev. |
| *DGP1: Serially correlated wages and labor set at time $t-b$* | | | | | | | | |
| 0.0 | 0.598 | 0.012 | 0.396 | 0.021 | −0.000 | 0.005 | 1.048 | 0.025 |
| 0.1 | 0.611 | 0.031 | 0.417 | 0.018 | 0.608 | 0.009 | 0.420 | 0.012 |
| 0.2 | 0.637 | 0.031 | 0.396 | 0.021 | 0.737 | 0.008 | 0.283 | 0.010 |
| 0.5 | 0.668 | 0.017 | 0.358 | 0.014 | 0.845 | 0.006 | 0.168 | 0.008 |
| *DGP2: Optimization error in labor* | | | | | | | | |
| 0.0 | 0.603 | 0.015 | 0.404 | 0.017 | 0.600 | 0.003 | 0.400 | 0.011 |
| 0.1 | 0.608 | 0.019 | 0.397 | 0.020 | 0.750 | 0.004 | 0.254 | 0.007 |
| 0.2 | 0.633 | 0.024 | 0.384 | 0.018 | 0.805 | 0.004 | 0.200 | 0.007 |
| 0.5 | 0.657 | 0.018 | 0.361 | 0.015 | 0.862 | 0.003 | 0.145 | 0.025 |
| *DGP3: DGP1 plus DGP2* | | | | | | | | |
| 0.0 | 0.599 | 0.014 | 0.422 | 0.029 | 0.452 | 0.004 | 0.573 | 0.015 |
| 0.1 | 0.597 | 0.012 | 0.420 | 0.014 | 0.603 | 0.005 | 0.424 | 0.011 |
| 0.2 | 0.598 | 0.011 | 0.419 | 0.016 | 0.671 | 0.005 | 0.355 | 0.011 |
| 0.5 | 0.620 | 0.019 | 0.407 | 0.016 | 0.752 | 0.005 | 0.271 | 0.032 |

NOTE: Models have been fit with `prodest lnva, method(lp) free(lnl) proxy(lnm) state(lnk) poly(3) valueadded reps(50) [acf]`.

Table A.7. ACF and LP (`dfp`), $\rho = 0.6$

| | ACF | | | | LP | | | |
|---|---|---|---|---|---|---|---|---|
| | $\beta_l$ | | $\beta_k$ | | $\beta_l$ | | $\beta_k$ | |
| Meas. error | Coeff. | St. Dev. | Coeff. | St. Dev. | Coeff. | St. Dev. | Coeff. | St. Dev. |
| *DGP1: Serially correlated wages and labor set at time $t-b$* | | | | | | | | |
| 0.0 | 0.603 | 0.012 | 0.402 | 0.015 | −0.000 | 0.005 | 1.047 | 0.025 |
| 0.1 | 0.638 | 0.014 | 0.409 | 0.015 | 0.607 | 0.010 | 0.420 | 0.012 |
| 0.2 | 0.654 | 0.005 | 0.388 | 0.011 | 0.737 | 0.008 | 0.283 | 0.010 |
| 0.5 | 0.658 | 0.003 | 0.366 | 0.008 | 0.845 | 0.006 | 0.168 | 0.009 |
| *DGP2: Optimization error in labor* | | | | | | | | |
| 0.0 | 0.593 | 0.013 | 0.379 | 0.008 | 0.600 | 0.003 | 0.400 | 0.011 |
| 0.1 | 0.619 | 0.017 | 0.398 | 0.017 | 0.751 | 0.004 | 0.253 | 0.007 |
| 0.2 | 0.647 | 0.006 | 0.380 | 0.013 | 0.805 | 0.004 | 0.200 | 0.007 |
| 0.5 | 0.656 | 0.004 | 0.362 | 0.008 | 0.862 | 0.003 | 0.145 | 0.024 |
| *DGP3: DGP1 plus DGP2* | | | | | | | | |
| 0.0 | 0.585 | 0.008 | 0.373 | 0.008 | 0.452 | 0.004 | 0.573 | 0.015 |
| 0.1 | 0.587 | 0.011 | 0.403 | 0.017 | 0.604 | 0.005 | 0.424 | 0.011 |
| 0.2 | 0.594 | 0.013 | 0.416 | 0.014 | 0.672 | 0.005 | 0.355 | 0.011 |
| 0.5 | 0.628 | 0.016 | 0.406 | 0.016 | 0.753 | 0.005 | 0.276 | 0.073 |

NOTE: Models have been fit with `prodest lnva, method(lp) free(lnl) proxy(lnm) state(lnk) poly(3) optimizer(dfp) valueadded reps(50) [acf]`.

Table A.8. ACF and LP (`dfp`), $\rho = 0.7$

| | ACF | | | | LP | | | |
|---|---|---|---|---|---|---|---|---|
| | $\beta_l$ | | $\beta_k$ | | $\beta_l$ | | $\beta_k$ | |
| Meas. error | Coeff. | St. Dev. | Coeff. | St. Dev. | Coeff. | St. Dev. | Coeff. | St. Dev. |
| *DGP1: Serially correlated wages and labor set at time $t-b$* | | | | | | | | |
| 0.0 | 0.594 | 0.010 | 0.373 | 0.005 | −0.000 | 0.005 | 1.087 | 0.029 |
| 0.1 | 0.590 | 0.022 | 0.422 | 0.013 | 0.676 | 0.009 | 0.363 | 0.012 |
| 0.2 | 0.634 | 0.016 | 0.400 | 0.017 | 0.788 | 0.007 | 0.240 | 0.010 |
| 0.5 | 0.656 | 0.003 | 0.364 | 0.009 | 0.875 | 0.005 | 0.260 | 0.239 |
| *DGP2: Optimization error in labor* | | | | | | | | |
| 0.0 | 0.596 | 0.009 | 0.362 | 0.004 | 0.600 | 0.003 | 0.400 | 0.012 |
| 0.1 | 0.598 | 0.013 | 0.378 | 0.011 | 0.753 | 0.004 | 0.256 | 0.008 |
| 0.2 | 0.606 | 0.019 | 0.393 | 0.015 | 0.807 | 0.004 | 0.203 | 0.010 |
| 0.5 | 0.649 | 0.012 | 0.377 | 0.016 | 0.863 | 0.003 | 0.370 | 0.320 |
| *DGP3: DGP1 plus DGP2* | | | | | | | | |
| 0.0 | 0.589 | 0.006 | 0.360 | 0.002 | 0.473 | 0.003 | 0.571 | 0.016 |
| 0.1 | 0.588 | 0.007 | 0.370 | 0.010 | 0.634 | 0.005 | 0.413 | 0.012 |
| 0.2 | 0.589 | 0.010 | 0.380 | 0.015 | 0.701 | 0.005 | 0.344 | 0.012 |
| 0.5 | 0.598 | 0.015 | 0.393 | 0.016 | 0.777 | 0.005 | 1.444 | 0.123 |

NOTE: Models have been fit with `prodest lnva, method(lp) free(lnl) proxy(lnm) state(lnk) poly(3) optimizer(dfp) valueadded reps(50) [acf]`.

Table A.9. ACF and LP (`bfgs`), $\rho = 0.6$

| | ACF | | | | LP | | | |
|---|---|---|---|---|---|---|---|---|
| | $\beta_l$ | | $\beta_k$ | | $\beta_l$ | | $\beta_k$ | |
| Meas. error | Coeff. | St. Dev. | Coeff. | St. Dev. | Coeff. | St. Dev. | Coeff. | St. Dev. |
| *DGP1: Serially correlated wages and labor set at time $t-b$* | | | | | | | | |
| 0.0 | 0.603 | 0.012 | 0.401 | 0.015 | −0.000 | 0.005 | 1.047 | 0.025 |
| 0.1 | 0.631 | 0.022 | 0.410 | 0.015 | 0.607 | 0.009 | 0.420 | 0.012 |
| 0.2 | 0.652 | 0.011 | 0.388 | 0.012 | 0.737 | 0.008 | 0.283 | 0.010 |
| 0.5 | 0.658 | 0.004 | 0.366 | 0.008 | 0.845 | 0.006 | 0.168 | 0.008 |
| *DGP2: Optimization error in labor* | | | | | | | | |
| 0.0 | 0.594 | 0.013 | 0.379 | 0.010 | 0.600 | 0.003 | 0.399 | 0.011 |
| 0.1 | 0.619 | 0.016 | 0.398 | 0.017 | 0.751 | 0.004 | 0.253 | 0.007 |
| 0.2 | 0.647 | 0.007 | 0.380 | 0.013 | 0.805 | 0.004 | 0.200 | 0.007 |
| 0.5 | 0.655 | 0.004 | 0.362 | 0.008 | 0.862 | 0.003 | 0.145 | 0.029 |
| *DGP3: DGP1 plus DGP2* | | | | | | | | |
| 0.0 | 0.586 | 0.007 | 0.374 | 0.011 | 0.452 | 0.004 | 0.572 | 0.014 |
| 0.1 | 0.589 | 0.009 | 0.405 | 0.019 | 0.604 | 0.005 | 0.424 | 0.010 |
| 0.2 | 0.596 | 0.010 | 0.417 | 0.015 | 0.672 | 0.005 | 0.355 | 0.010 |
| 0.5 | 0.627 | 0.016 | 0.406 | 0.015 | 0.753 | 0.005 | 0.275 | 0.072 |

NOTE: Models have been fit with `prodest lnva, method(lp) free(lnl) proxy(lnm) state(lnk) poly(3) optimizer(bfgs) valueadded reps(50)` [`acf`].

Table A.10. ACF and LP (`bfgs`), $\rho = 0.7$

| | ACF | | | | LP | | | |
|---|---|---|---|---|---|---|---|---|
| | $\beta_l$ | | $\beta_k$ | | $\beta_l$ | | $\beta_k$ | |
| Meas. error | Coeff. | St. Dev. | Coeff. | St. Dev. | Coeff. | St. Dev. | Coeff. | St. Dev. |
| *DGP1: Serially correlated wages and labor set at time $t-b$* | | | | | | | | |
| 0.0 | 0.594 | 0.010 | 0.373 | 0.006 | 0.000 | 0.005 | 1.088 | 0.029 |
| 0.1 | 0.591 | 0.020 | 0.422 | 0.014 | 0.677 | 0.009 | 0.363 | 0.012 |
| 0.2 | 0.633 | 0.017 | 0.401 | 0.017 | 0.788 | 0.007 | 0.240 | 0.010 |
| 0.5 | 0.656 | 0.003 | 0.365 | 0.010 | 0.875 | 0.005 | 0.258 | 0.237 |
| *DGP2: Optimization error in labor* | | | | | | | | |
| 0.0 | 0.596 | 0.010 | 0.362 | 0.005 | 0.600 | 0.003 | 0.400 | 0.013 |
| 0.1 | 0.599 | 0.013 | 0.378 | 0.012 | 0.753 | 0.004 | 0.256 | 0.008 |
| 0.2 | 0.606 | 0.019 | 0.393 | 0.015 | 0.807 | 0.004 | 0.203 | 0.010 |
| 0.5 | 0.650 | 0.011 | 0.376 | 0.015 | 0.863 | 0.003 | 0.374 | 0.321 |
| *DGP3: DGP1 plus DGP2* | | | | | | | | |
| 0.0 | 0.590 | 0.006 | 0.360 | 0.002 | 0.473 | 0.004 | 0.570 | 0.016 |
| 0.1 | 0.589 | 0.008 | 0.374 | 0.019 | 0.634 | 0.005 | 0.412 | 0.012 |
| 0.2 | 0.594 | 0.011 | 0.393 | 0.028 | 0.701 | 0.005 | 0.344 | 0.033 |
| 0.5 | 0.603 | 0.015 | 0.398 | 0.019 | 0.777 | 0.005 | 1.449 | 0.127 |

NOTE: Models have been fit with `prodest lnva, method(lp) free(lnl) proxy(lnm) state(lnk) poly(3) optimizer(bfgs) valueadded reps(50) [acf]`.

Table A.11. ACF and LP (`bhhh`), $\rho = 0.7$

| | ACF | | | | LP | | | |
|---|---|---|---|---|---|---|---|---|
| | $\beta_l$ | | $\beta_k$ | | $\beta_l$ | | $\beta_k$ | |
| Meas. error | Coeff. | St. Dev. | Coeff. | St. Dev. | Coeff. | St. Dev. | Coeff. | St. Dev. |
| *DGP1: Serially correlated wages and labor set at time $t-b$* | | | | | | | | |
| 0.0 | 0.611 | 0.009 | 0.414 | 0.018 | 0.003 | 0.004 | 1.099 | 0.021 |
| 0.1 | 0.579 | 0.023 | 0.424 | 0.023 | 0.683 | 0.011 | 0.360 | 0.015 |
| 0.2 | 0.643 | 0.018 | 0.399 | 0.011 | 0.792 | 0.008 | 0.238 | 0.009 |
| 0.5 | 0.669 | 0.007 | 0.360 | 0.004 | 0.877 | 0.007 | 0.142 | 0.009 |
| *DGP2: Optimization error in labor* | | | | | | | | |
| 0.0 | 0.607 | 0.014 | 0.416 | 0.025 | 0.600 | 0.003 | 0.392 | 0.015 |
| 0.1 | 0.608 | 0.013 | 0.407 | 0.016 | 0.755 | 0.004 | 0.248 | 0.009 |
| 0.2 | 0.602 | 0.019 | 0.403 | 0.017 | 0.809 | 0.003 | 0.197 | 0.009 |
| 0.5 | 0.642 | 0.025 | 0.376 | 0.026 | 0.866 | 0.003 | 0.713 | 0.066 |
| *DGP3: DGP1 plus DGP2* | | | | | | | | |
| 0.0 | 0.617 | 0.014 | 0.476 | 0.033 | 0.475 | 0.004 | 0.571 | 0.019 |
| 0.1 | 0.612 | 0.006 | 0.443 | 0.011 | 0.634 | 0.005 | 0.415 | 0.013 |
| 0.2 | 0.606 | 0.008 | 0.423 | 0.019 | 0.702 | 0.004 | 0.344 | 0.009 |
| 0.5 | 0.610 | 0.017 | 0.413 | 0.016 | 0.775 | 0.006 | 1.462 | 0.048 |

NOTE: Models have been fit with `prodest lnva, method(lp) free(lnl) proxy(lnm) state(lnk) poly(3) optimizer(bhhh) valueadded reps(50)` [`acf`].

Table A.12. ACF (`nr`): Monte Carlo and DGP2

Panel a: Meas. err. 0

| | (1) | (2) | (3) | (4) | (5) | (6) | (7) | (8) | (9) |
|---|---|---|---|---|---|---|---|---|---|
| $\beta_l$ | 0.656 | 0.636 | 0.641 | 0.609 | 0.630 | 0.601 | 0.636 | 0.644 | 0.981 |
| | (0.086) | (0.079) | (0.107) | (0.014) | (0.096) | (0.008) | (0.051) | (0.019) | (0.065) |
| $\beta_k$ | 0.384 | 0.336 | 0.345 | 0.416 | 0.387 | 0.401 | 0.433 | 0.175 | 0.261 |
| | (0.189) | (0.164) | (0.140) | (0.021) | (0.095) | (0.018) | (0.031) | (0.076) | (0.622) |

Panel b: Meas. err. 0.1

| | (1) | (2) | (3) | (4) | (5) | (6) | (7) | (8) | (9) |
|---|---|---|---|---|---|---|---|---|---|
| $\beta_l$ | 0.628 | 0.620 | 0.615 | 0.611 | 0.628 | 0.607 | 0.613 | 0.695 | 0.957 |
| | (0.111) | (0.084) | (0.071) | (0.041) | (0.086) | (0.012) | (0.018) | (0.031) | (0.032) |
| $\beta_k$ | 0.377 | 0.378 | 0.382 | 0.376 | 0.384 | 0.403 | 0.382 | 0.149 | 0.148 |
| | (0.124) | (0.110) | (0.104) | (0.081) | (0.086) | (0.017) | (0.049) | (0.074) | (0.134) |

Panel c: Meas. err. 0.2

| | (1) | (2) | (3) | (4) | (5) | (6) | (7) | (8) | (9) |
|---|---|---|---|---|---|---|---|---|---|
| $\beta_l$ | 0.553 | 0.571 | 0.583 | 0.590 | 0.624 | 0.613 | 0.620 | 0.715 | 0.943 |
| | (0.189) | (0.140) | (0.103) | (0.064) | (0.062) | (0.014) | (0.015) | (0.027) | (0.016) |
| $\beta_k$ | 0.442 | 0.424 | 0.421 | 0.412 | 0.389 | 0.400 | 0.356 | 0.155 | 0.139 |
| | (0.181) | (0.142) | (0.121) | (0.064) | (0.060) | (0.018) | (0.050) | (0.059) | (0.084) |

Panel d: Meas. err. 0.5

| | (1) | (2) | (3) | (4) | (5) | (6) | (7) | (8) | (9) |
|---|---|---|---|---|---|---|---|---|---|
| $\beta_l$ | 0.307 | 0.382 | 0.447 | 0.521 | 0.610 | 0.628 | 0.675 | 0.731 | 0.927 |
| | (0.239) | (0.196) | (0.145) | (0.099) | (0.057) | (0.020) | (0.042) | (0.034) | (0.022) |
| $\beta_k$ | 0.674 | 0.603 | 0.544 | 0.477 | 0.404 | 0.389 | 0.344 | 0.181 | 0.128 |
| | (0.224) | (0.183) | (0.132) | (0.086) | (0.046) | (0.022) | (0.040) | (0.055) | (0.093) |

NOTE: In columns 1–9, we report ACF estimates with different optimization starting points from $[0.1, (1 - 0.1)]$ to $[0.9, (1 - 0.9)]$, respectively. Standard errors in parentheses.

Table A.13. ACF (`dfp`): Monte Carlo and DGP3

Panel a: Meas. err. 0

| | (1) | (2) | (3) | (4) | (5) | (6) | (7) | (8) | (9) |
|---|---|---|---|---|---|---|---|---|---|
| $\beta_l$ | 0.595 | 0.626 | 0.622 | 0.620 | 0.607 | 0.592 | 0.589 | 0.739 | 1.006 |
| | (0.116) | (0.101) | (0.024) | (0.019) | (0.009) | (0.006) | (0.005) | (0.088) | (0.007) |
| $\beta_k$ | 0.108 | 0.401 | 0.487 | 0.487 | 0.466 | 0.401 | 0.325 | 0.942 | −0.00786 |
| | (0.324) | (0.158) | (0.053) | (0.049) | (0.022) | (0.001) | (0.006) | (0.739) | (0.013) |

Panel b: Meas. err. 0.1

| | (1) | (2) | (3) | (4) | (5) | (6) | (7) | (8) | (9) |
|---|---|---|---|---|---|---|---|---|---|
| $\beta_l$ | 0.645 | 0.657 | 0.642 | 0.610 | 0.613 | 0.591 | 0.590 | 0.904 | 0.943 |
| | (0.099) | (0.226) | (0.210) | (0.042) | (0.010) | (0.009) | (0.008) | (0.112) | (0.009) |
| $\beta_k$ | 0.315 | 0.241 | 0.226 | 0.425 | 0.447 | 0.405 | 0.343 | 0.234 | 0.0694 |
| | (0.224) | (0.322) | (0.293) | (0.059) | (0.015) | (0.004) | (0.021) | (0.303) | (0.021) |

Panel c: Meas. err. 0.2

| | (1) | (2) | (3) | (4) | (5) | (6) | (7) | (8) | (9) |
|---|---|---|---|---|---|---|---|---|---|
| $\beta_l$ | 0.651 | 0.676 | 0.672 | 0.623 | 0.607 | 0.593 | 0.593 | 0.868 | 0.917 |
| | (0.121) | (0.274) | (0.256) | (0.053) | (0.013) | (0.012) | (0.009) | (0.038) | (0.004) |
| $\beta_k$ | 0.319 | 0.281 | 0.241 | 0.320 | 0.428 | 0.412 | 0.347 | 0.304 | 0.102 |
| | (0.186) | (0.325) | (0.319) | (0.162) | (0.026) | (0.009) | (0.021) | (0.160) | (0.007) |

Panel d: Meas. err. 0.5

| | (1) | (2) | (3) | (4) | (5) | (6) | (7) | (8) | (9) |
|---|---|---|---|---|---|---|---|---|---|
| $\beta_l$ | 0.610 | 0.660 | 0.656 | 0.632 | 0.608 | 0.606 | 0.605 | 0.841 | 0.901 |
| | (0.308) | (0.215) | (0.097) | (0.050) | (0.023) | (0.013) | (0.015) | (0.035) | (0.003) |
| $\beta_k$ | 0.373 | 0.324 | 0.321 | 0.351 | 0.413 | 0.419 | 0.353 | 0.305 | 0.101 |
| | (0.327) | (0.241) | (0.141) | (0.114) | (0.028) | (0.013) | (0.029) | (0.088) | (0.003) |

NOTE: In columns 1–9, we report ACF estimates with different optimization starting points from $[0.1, (1 - 0.1)]$ to $[0.9, (1 - 0.9)]$, respectively. Standard errors in parentheses.